SENTRY-LOGIC Framework: Detailed Modular Structure

1. Module Breakdown:

Module Name	Function	Input/Output Format	How it Contributes to Symbolic Audit Logging
Input Handler	Receives raw prompt and meta-tags.	Raw Prompt (String), Meta-tags (Dict) → Structured Input (Dict)	Parses raw input, identifies slots, and formats it for symbolic processing.
Symbol Mapper	Analyzes structured input.	Structured Input (Dict) → Symbolic Elements (List of Symbols)	Maps input to predefined symbolic representations (Δ, Ω, Λ, ⇄).
Log Writer	Creates audit logs.	Symbolic Elements (List of Symbols), LLM Response (String) → Log Entry (Structured)	Converts symbolic representation and LLM output into a structured log entry (e.g., JSON) with timestamps, context, and symbolic tags.
Alert Agent	Detects anomalous patterns.	Log Entry (Structured) → Alert (Optional, Structured)	Analyzes log entries for specific symbol sequences or unexpected patterns that indicate potential issues (e.g., policy violations, drift). Generates alerts if necessary.

2. Symbol Processing Flow:

The core symbolic elements are processed as follows:

1. **Input:** Raw prompt and meta-tags are received by the *Input Handler*.

2. **Structuring:** The *Input Handler* parses the raw input and structures it.

3. **Mapping:** The *Symbol Mapper* analyzes the structured input to identify:

    * **Δ (Context Shift):** If changes in topic, tone, or style are detected between the prompt and the response. Uses sentiment analysis, topic modeling, or similarity measures.

    * **Ω (Policy Trigger):** If specific keywords, phrases, or semantic patterns in the prompt or response match predefined policy rules. May involve regular expressions, keyword lists, or semantic similarity comparisons.

    * **Λ (Data Access):** If the LLM response contains references to external data sources (if metadata is available from the LLM API or can be inferred).

    * **⇄ (Semantic Rewrite):** If the meaning of the user's prompt is altered or distorted in the LLM's response. Uses semantic similarity calculations between the prompt and response.

4. **Fallback Methods:**

    * For **Δ**, if direct triggers are absent, semantic similarity between prompt and response is used to estimate context shifts. Changes in core terms, topics, or style indicate a potential context shift.

    * For **Ω**, if no policy information is directly available, keyword matching and semantic analysis of the LLM output is used to try to identify if common policy terms are present.

5. **Logging:** The *Log Writer* receives the extracted symbolic elements and the LLM response. It combines these into a structured log entry that includes:

    * Timestamp

    * Prompt

    * Meta-tags

    * LLM Response

    * List of detected symbols (Δ, Ω, Λ, ⇄)

3. Data Interfaces & Interactions:

The modules interact as follows:

* **Input Handler → Symbol Mapper:** The *Input Handler* provides the *Symbol Mapper* with a structured representation of the prompt and meta-tags.

* **Symbol Mapper → Log Writer:** The *Symbol Mapper* provides the *Log Writer* with a list of extracted symbolic elements (Δ, Ω, Λ, ⇄) representing the observed LLM behavior.

* **Log Writer -> Alert Agent (Optional):** The *Log Writer* passes the structured log entries (which contain the symbols) to the *Alert Agent* (if present).

* **Log Writer (and Input Handler/Output Proxy) → External Systems:** The *Log Writer* sends log entries to external logging/storage systems (e.g., Elasticsearch, cloud logging). If acting as a proxy, the *Input Handler* intercepts input, and the *Output Proxy* forwards the LLM response.

4. Deployment Model (Basic Version):

SENTRY-LOGIC can be deployed as a network proxy alongside the LLM:

```
[User/Application]
|
v (Prompt, Meta-tags)
+----------------+
| SENTRY-LOGIC   |
| (Network Proxy)  |
|   Input Handler|
|   Symbol Mapper |
|   Log Writer    |
+----------------+
|
v (Modified request, or direct pass-through)
+----------------+
|      LLM        |
+----------------+
|
v (LLM Response)
+----------------+
| SENTRY-LOGIC   |
| (Network Proxy)  |
|    Output Proxy |
|    Log Writer   |
+----------------+
|
v (LLM Response, Logs)
[User/Application and/or Logging/Storage]
```

In this model, SENTRY-LOGIC intercepts all requests and responses, performs the symbolic mapping and logging, and then forwards the traffic to the LLM (and back to the user/application).