Project Blueprint Outline: The Autonomous Literature Explorer

Document Purpose: To serve as the definitive technical and conceptual specification for the Autonomous Literature Explorer project. This document details the system's architecture, core mechanisms, operational logic, and staged implementation plan. It is intended for the project lead and the implementation team.

Part 1: Vision and Guiding Principles

- **1.1. Executive Summary: The Mission**
  - 1.1.1. Project Goal: To develop a fully autonomous agent for exploratory literature intelligence, capable of discovering, defining, and analyzing novel research questions.
  - 1.1.2. Core Capabilities: (a) Automated discovery of primary evidence, (b) High-confidence verification of existing Systematic Reviews (SRs), (c) Identification of specific literature gaps and contradictions.
  - 1.1.3. Key Differentiator: A disciplined, auditable process that combines the semantic reasoning of Large Language Models (LLMs) with the rigor of graph analytics and information retrieval.
- **1.2. The Core Problem: Beyond Procedural Search**
  - 1.2.1. Contrasting "Systematic Search" vs. "Exploratory Discovery": Defining the unstructured, heuristic-driven nature of forming new research questions.
  - 1.2.2. The Automation Challenge: Overcoming information overload, uncovering non-obvious connections, and formalizing the process of identifying "what isn't known."
- **1.3. Design Philosophy: The Foundational Pillars**
  - 1.3.1. **LLM-as-Policy, Code-as-Measurement:** The central architectural principle.
    - Defining the LLM's role (Gemma 3N): Strategic planning, semantic judgment, and policy selection.
    - Defining the code's role: Deterministic measurement, data retrieval, safety enforcement, and execution of the LLM's plan.
  - 1.3.2. **Fully Autonomous Operation:** Specification for a "headless" agent with no human-in-the-loop for the Proof-of-Concept (PoC) phase.
  - 1.3.3. **Resource-Aware Design:** Explicit constraints for efficient operation on modest local hardware (RTX 4050 6GB), prioritizing latency and VRAM management.
  - 1.3.4. **Reproducibility and Auditability:** Every decision, measurement, and state transition must be logged to ensure the agent's behavior is transparent and verifiable.

Part 2: System Architecture and Data Primitives

- **2.1. The Three-Layer Architectural Model**
  - 2.1.1. **Layer 1: Perception & Measurement (The Senses & The Lab)**
    - Responsibilities: Data acquisition, representation, and computation of quantitative signals.
    - Components: Hybrid Retrieval Engine, Graph Exploration Engine, Embedding Engine (Qwen3).
  - 2.1.2. **Layer 2: Cognition & Policy (The Brain - Gemma 3N)**
    - Responsibilities: Decision-making, planning, semantic judgment.
    - Components: Planner, PICO Gate, SR Verdict Judge, Identity Arbiter.
  - 2.1.3. **Layer 3: Memory & Governance (The Lab Notebook)**
    - Responsibilities: Persistence, state management, rule enforcement.
    - Components: The Database, The Agenda Service, The Orchestrator.
- **2.2. Core Data Primitives: The "Nouns" of the System**
  - 2.2.1. **Paper:** The atomic unit. Includes identifiers (DOI-first), metadata, and its multi-vector representation (Abstract + PICO Facets).
  - 2.2.2. **Graph:** A multi-layered network structure.
    - Layer A: Directed Citation Edges.
    - Layer B: Undirected Bibliographic Coupling Edges.
    - Layer C: Undirected Semantic K-Nearest Neighbor (KNN) Edges.
  - 2.2.3. **Focus Set:** The small "anchor" set of papers defining the current topic's center.
  - 2.2.4. **Sentinel Set (S):** The high-fidelity, evolving proxy for the "included studies" list. The foundation of trustworthiness.
  - 2.2.5. **Agenda Item & Epochs:** The persistent representation of a Research Question (RQ), designed to track its lifecycle over time without creating duplicates.
- **2.3. Research Question Identity: A Robust, Two-Key System**
  - 2.3.1. The Failure of Brittle Hashes: Why a simple string hash is inadequate.
  - 2.3.2. **The Semantic Key:** Using weighted cosine similarity of PICO facet vectors, with high weight on Intervention (I) and Comparator (C).
  - 2.3.3. **The Evidence Key:** Using Jaccard overlap of the Sentinel Sets of two RQs.
  - 2.3.4. **The LLM as Arbiter:** The process for Gemma to adjudicate borderline cases, labeling them same, broader, narrower, or different with a justification that names the differing facets.

Part 3: Deep Dive into Core Mechanisms

- **3.1. Hybrid Retrieval Engine**

    - 3.1.1. Lexical Retrieval (BM25): Explanation of the algorithm and the necessity of z-score normalization for combining scores.
    - 3.1.2. Dense Retrieval (Qwen3): The role of embeddings in capturing semantic meaning beyond keywords.
    - 3.1.3. The Hybrid Ranker: The formula for combining signals (cosine, z-score, recency, type boost, novelty) into a single priority score.
- **3.2. The Sentinel Set Lifecycle: From Nomination to Canonization**
    - 3.2.1. The Full Promotion Pipeline: A step-by-step breakdown.
        - Step A: Candidate Nomination (from Retrieval & Ripple).
        - Step B: Fast Metadata Filtering (Publication Type, Retraction Flags).
        - Step C: The LLM PICO Gate (The batched, binary relevance judgment for ambiguous abstracts).
        - Step D: Deterministic Ranking for Promotion (based on PICO fit, design, novelty).
        - Step E: Throttled Admission into the Sentinel Set.
    - 3.2.2. The Three Exhaustiveness Audits: How the system proves S is complete.
        - Audit 1: Discovery Curve Saturation Analysis.
        - Audit 2: Capture-Recapture Logic across retrieval channels.
        - Audit 3: The Systematic SR-Reference Audit to find and include "missing-from-S" primaries.
- **3.3. Ripple Explosion Control: The Multi-Gate Defense System**
    - 3.3.1. **The Semantic Gate:** Adaptive percentile-based similarity threshold.
    - 3.3.2. **The Structural Gate:** The concept of support_up and support_down based on overlap with the trusted set U.
    - 3.3.3. **The Innovation Budget:** The mechanism for preserving novelty by admitting a small quota of "bridge" papers that fail structural checks.
    - 3.3.4. **Hard Safeguards:** The role of degree caps and per-episode quotas.
    - 3.3.5. **Hub Attenuation:** The technique of down-weighting edges from high-degree nodes (e.g., guidelines) to prevent them from dominating graph-based metrics.
- **3.4. The SR Verdict Pipeline: From Evidence to Judgment**
    - 3.4.1. The Evidence Brief: A detailed breakdown of the structured, labeled data packet prepared for the LLM (fields: coverage, P/I/C/O_match, recency, guideline_like, missing_sentinels).
    - 3.4.2. The "Point-and-Quote" Discipline: The rule requiring Gemma to justify its verdict by quoting the labels from the brief, enforced by a validator.
    - 3.4.3. The Update Trigger ("Greediness"): The formula and policy for automatically triggering an Update-hunt mode based on SR age and missed evidence.
- **3.5. Key Supporting Technologies Explained**
    - 3.5.1. **Vector Indexing (HNSW on CPU):** A primer on Approximate Nearest Neighbor search, why HNSW is the state-of-the-art, and the strategic rationale for using a CPU implementation to conserve GPU VRAM.
    - 3.5.2. **Term Velocity:** A technical explanation of the calculation (time-series analysis of term frequency) and its application in Rapid Focus mode for identifying emerging research trends.
    - 3.5.3. **Bandit Auto-Tuning:** A simple explanation of multi-armed bandits as a method for online, lightweight optimization of system parameters (e.g., gate strictness) using Sentinels per Read as the payoff metric.

Part 4: The Agent's Operational Logic

- **4.1. The Episodic Loop: A Cycle of Perception, Planning, and Action**
    - 4.1.1. The Instrument Panel: A full specification of the JSON digest the LLM receives at the start of each episode.
    - 4.1.2. The Mode System: Defining each mode (Primary-scout, SR-hunt, Gap-scan, Contradiction-probe, Update-hunt, Rapid Focus) as a distinct goal-oriented policy.
    - 4.1.3. The Plan: The structure of the short-term action plan generated by Gemma, consisting of tool calls and quotas.
    - 4.1.4. The Execution Phase: How the Orchestrator executes the plan, enforces guardrails, and collects feedback.
- **4.2. Data Integrity and Persistence**
    - 4.2.1. The DOI-First Identifier Policy.
    - 4.2.2. Minimal Viable Strategies for Retraction and Preprint Handling in the PoC.
    - 4.2.3. High-Level Database Schema Outline.

Part 5: Proof-of-Concept (PoC) Strategy

- **5.1. Evaluation Methodology without Human Gold Standards**
    - 5.1.1. Using "Silver Standard" SRs for back-testing Sentinel recall.
    - 5.1.2. Self-Consistency and Stability Tests across different random seeds.
    - 5.1.3. Analysis of Learning Curves (e.g., acceptance_rate vs. shadow_recall).
- **5.2. The Staged PoC Implementation Plan**
    - A detailed list of ~11 independent PoCs, each with a clear Goal, Procedure, and Pass/Fail Metrics.
    - Examples: PoC for PICO Gate throughput, PoC for Sentinel Set saturation, PoC for Ripple Gate ablation testing, PoC for SR Verdict pipeline integrity, PoC for performance profiling on the target hardware.

Project Blueprint: The Autonomous Literature Explorer

Part 1: Vision and Guiding Principles

1.1. Executive Summary: The Mission

This document provides the definitive architectural and conceptual blueprint for the **Autonomous Literature Explorer**. This project's central mission is to develop a sophisticated, fully autonomous software agent capable of performing high-level exploratory intelligence within vast bodies of scientific literature, specifically targeting the biomedical domain. The system is designed to emulate and then augment the complex cognitive processes of a senior researcher navigating a new or evolving field of study.

The core capabilities of this agent are threefold, representing a complete cycle of scientific inquiry:

1. **Automated Discovery of Primary Evidence:** The agent will systematically identify and curate a high-fidelity collection of foundational studies (e.g., Randomized Controlled Trials, cohort studies) that form the evidentiary bedrock of a given research question. This curated collection, which we term the **Sentinel Set**, serves as the system's internal ground truth.
2. **High-Confidence Verification of Existing Systematic Reviews (SRs):** Before significant resources are committed to a new review, it is critical to determine if the question has already been comprehensively answered. Our agent will move beyond simple keyword searches for "systematic review." It will rigorously assess candidate SRs by checking if their cited evidence substantially overlaps with our independently-generated Sentinel Set, thereby verifying true topical and evidentiary coverage.
3. **Identification of Specific Literature Gaps and Contradictions:** The ultimate goal of exploratory research is discovery. The agent is designed to pinpoint precise, actionable opportunities for new research. It will identify **gaps** (e.g., untested drug comparators, unstudied patient populations, missing outcome measures) and **contradictions** (instances where studies with similar designs report conflicting results), then articulate these findings as well-formed, novel research hypotheses.

The key differentiator of this project lies in its disciplined and auditable synthesis of two powerful paradigms: the nuanced, semantic reasoning of modern Large Language Models (LLMs) and the structured, quantitative rigor of information retrieval, graph theory, and statistical analysis. The result is not merely a search tool, but an automated engine for scientific discovery.

1.2. The Core Problem: Beyond Procedural Search

The necessity of this project is rooted in a fundamental distinction between two modes of engaging with scientific literature.

- **Systematic Search:** This is a well-defined, procedural, and largely solved problem. It is analogous to following a detailed recipe. A researcher begins with a precise, fixed research question (often framed as PICO: Population, Intervention, Comparator, Outcome). They then construct a complex, exhaustive search query designed to retrieve *all* possible evidence pertaining to that specific question. This is a process of **verification**, aiming for maximum recall to confirm what is known.
- **Exploratory Discovery:** This is the creative, heuristic, and currently unstandardized process of *inventing the recipe*. It is the cognitive work a domain expert performs to identify a question worth asking in the first place. This process is inherently non-linear. It involves skimming recent publications, following citation trails backward (references) and forward (citers), noticing emergent patterns, sensing where the "heat" in a field is, and gradually converging on a research question that is not only interesting but also novel and feasible.

The automation of Exploratory Discovery presents a profound challenge. It is not simply a matter of scale; it is a matter of automating insight. The core difficulties include navigating overwhelming information volume, uncovering "unknown unknowns" (relevant papers one would not know to search for by name), and formalizing the intuitive "aha!" moment when a literature gap or contradiction becomes clear. Our system is designed to meet this challenge head-on by structuring and automating this very process of exploration.

1.3. Design Philosophy: The Foundational Pillars

The architecture of the Autonomous Literature Explorer is built upon four foundational pillars that ensure its intelligence, reliability, and practicality.

1. **LLM-as-Policy, Code-as-Measurement:** This is the system's central organizing principle. We strictly separate the roles of the AI and the traditional code.
    - **The LLM's Role (Policy and Judgment):** The LLM (Gemma 3N) acts as the strategic brain or the "captain of the ship." It is responsible for tasks requiring semantic understanding and reasoning. It interprets the state of the exploration, sets the goal for the next operational cycle, formulates search strategies, and makes final judgments on qualitative questions (e.g., "Does this abstract truly match our PICO?," "Does this SR's scope align with our findings?").
    - **The Code's Role (Measurement and Safety):** The deterministic codebase acts as the "ship's engine room and navigation instruments." It executes the LLM's commands precisely and reports back hard, quantitative data (e.g., similarity scores, term frequencies, graph centrality metrics, overlap statistics). It also enforces all safety guardrails, such as resource quotas and admission gates, ensuring the agent's exploratory behavior remains bounded, efficient, and predictable. The LLM decides *what* to do; the code determines *how* it gets done and measures the result.
2. **Fully Autonomous Operation:** For the initial and all core phases of this project, the system is designed to operate without any human intervention. There is no "human-in-the-loop" to correct course or provide feedback during a run. This is a deliberate design constraint, not a limitation. It forces us to build an agent with robust internal logic, sophisticated error-handling, and the capacity for self-correction based on the feedback from its own measurements. This ensures the final system is genuinely autonomous, not merely a human-assisted tool.
3. **Resource-Aware Design:** The system is explicitly engineered for high performance on modest, widely available local hardware (specifically, a laptop with an RTX 4050 6GB GPU and 32GB of RAM). This is not an afterthought but a primary design driver. Every architectural choice—from the selection of lightweight models like Gemma and Qwen, to the use of efficient algorithms like HNSW on the CPU for vector indexing, and the operational logic of batching and token-miserly prompts—is optimized to minimize VRAM usage and latency, making the tool accessible and practical.
4. **Reproducibility and Auditability:** For a scientific discovery tool to be trustworthy, its reasoning must be transparent. The system is designed to function like a meticulous researcher keeping a detailed lab notebook. Every decision, every measurement, and every state transition is logged in a structured, queryable format. This creates a complete audit trail for any given run, allowing us to reconstruct precisely why the agent pursued a certain path, how it arrived at a conclusion, and what evidence supported its final verdict.

Part 2: System Architecture and Data Primitives

This section outlines the concrete blueprint of the system, defining its main components and the essential data structures, or "primitives," that serve as the building blocks for all of its operations.

2.1. The Three-Layer Architectural Model

The system's architecture is modular, organized into three distinct layers, each with a clear set of responsibilities. This separation ensures that the system is robust, maintainable, and adaptable.

- **2.1.1. Layer 1: Perception & Measurement (The Senses & The Lab)**
  This is the system's interface with the external world of data and its internal engine for quantitative analysis. Its role is strictly to gather information and compute objective facts, without interpretation.
    - **Responsibilities:** Acquiring raw data from scientific literature databases, representing that data in a machine-readable format (vectors and graphs), and computing a suite of well-defined quantitative signals.
    - **Core Components:**
        - **Hybrid Retrieval Engine:** This component executes queries against external databases (e.g., PubMed). It employs both lexical (BM25) and semantic (vector-based) search methods to ensure comprehensive data acquisition.
        - **Graph Exploration Engine:** This component navigates the citation network by executing "ripple" operations—following links to both citing papers (downstream) and referenced papers (upstream).
        - **Embedding Engine (Qwen3):** This component uses a small, efficient embedding model to translate textual information into high-dimensional vectors, which underpin all semantic similarity calculations.
- **2.1.2. Layer 2: Cognition & Policy (The Brain - Gemma 3N)**
  This layer is the cognitive core of the agent, where reasoning, planning, and judgment occur. It is powered by the Gemma Large Language Model.
    - **Responsibilities:** Interpreting the measured state of the system, formulating strategic plans, and making nuanced judgments that require a deep understanding of natural language.
    - **Core Components:**
        - **Planner:** Selects the operational Mode (e.g., SR-hunt, Gap-scan) for the upcoming work cycle ("episode") and generates a specific, budgeted Plan of action.
        - **PICO Gate:** Performs a fast, binary relevance judgment on ambiguous papers to determine if they align with the current research question.
        - **SR Verdict Judge:** Renders the final, high-level conclusion regarding the existence and quality of a systematic review, based on a structured evidence brief.
        - **Identity Arbiter:** Resolves ambiguity between two similar but not identical research questions, deciding if they should be merged or treated as distinct.
- **2.1.3. Layer 3: Memory & Governance (The Lab Notebook)**
  This layer provides the system with persistence, statefulness, and the enforcement of operational rules. It is the bedrock of the system's auditability and long-term learning.

- **Responsibilities:** Storing all collected data and learned knowledge, managing the lifecycle of research questions, and executing the LLM's plans while rigorously enforcing resource quotas and safety guardrails.
- **Core Components:**
  - **The Database:** A structured persistence layer (e.g., DuckDB/Postgres) that stores all data primitives, from individual papers to entire episode logs.
  - **The Agenda Service:** A critical state management service that maintains the official registry of all researched questions, preventing duplicate work and tracking their progress over time.
  - **The Orchestrator:** The master controller code that translates the LLM's high-level Plan into concrete function calls to the Perception layer, ensuring all operational constraints (e.g., quotas, gates) are strictly followed.

2.2. Core Data Primitives: The "Nouns" of the System

These are the fundamental objects that the agent creates, analyzes, and manipulates.

- **2.2.1. Paper:** The atomic unit of literature. Each Paper object is a structured representation containing:
  - **Identifiers:** A canonical **DOI** (Digital Object Identifier) as the primary key, with a mapped **PMID** (PubMed ID) where applicable. A strict DOI-first policy is enforced.
  - **Metadata:** Standard bibliographic information (Title, Abstract, Authors, Year, Publication Venue, Publication Type).
  - **Multi-Vector Representation:** A set of numerical embeddings generated by the Qwen model. This includes a global **Abstract Vector** for overall topic similarity and five specific **Facet Vectors** corresponding to each element of the **PICO(S)** framework (Population, Intervention, Comparator, Outcome, Study Design). This faceted representation allows for highly targeted and weighted similarity comparisons.
- **2.2.2. Graph:** The system maintains a multi-modal network representation of the literature, where papers are nodes and relationships are typed edges.
  - **Layer A: Directed Citation Edges:** Represents the formal citation network (e.g., Paper A -> cites -> Paper B). This is the primary structure used for ripple exploration.
  - **Layer B: Undirected Bibliographic Coupling Edges:** A weighted edge connecting two papers that cite a high number of the same references. This is a strong indicator of shared intellectual foundation.
  - **Layer C: Undirected Semantic K-Nearest Neighbor (KNN) Edges:** A weighted edge connecting each paper to its 'k' most similar neighbors in the abstract vector space. This layer reveals thematic clusters independent of citation behavior.
- **2.2.3. Focus Set:** A small, dynamic list of approximately 20-50 papers that are considered the most central and representative "anchors" for the currently active Research Question. This set is used to compute a "topic centroid" in the vector space, which helps to keep all retrieval and ripple operations semantically aligned and prevent topic drift.
- **2.2.4. Sentinel Set (S):** This is arguably the most critical data structure for the system's trustworthiness. It is a high-fidelity, curated collection of **primary studies** (e.g., RCTs, cohort studies) that the agent has vetted and accepted as directly relevant to the PICO of a given Research Question. It serves as the system's evolving internal proxy for the final "list of

included studies" that would be produced by a comprehensive human-led systematic review. It is the ground-truth standard against which all candidate SRs are measured.

- **2.2.5. Agenda Item & Epochs:** An Agenda Item is the persistent, top-level object representing a single Research Question. It contains the canonical PICO definition, its associated facet vectors, pointers to its Focus and Sentinel sets, its current operational status (e.g., exploring, concluded), and a full log of all episodes undertaken in its investigation. To manage the evolution of science over time, the system uses **Epochs**. Instead of creating a new, duplicate Agenda Item when revisiting an old question, the agent re-opens the existing item and begins a new, timestamped epoch (e.g., 2025-Q3-Update), preserving the complete historical lineage of inquiry.

2.3. Research Question Identity: A Robust, Two-Key System

To effectively manage the Agenda and avoid redundant work, the system needs a sophisticated method for determining if a newly proposed research question is truly novel or simply a rephrasing of an existing one. A simple text hash is too brittle and will fail. Therefore, we employ a robust two-key identity system.

- **2.3.1. The Failure of Brittle Hashes:** A cryptographic hash of the PICO text is hyper-sensitive. Changing a single word ("heart failure" vs. "cardiac failure") would generate a completely different hash, incorrectly leading the system to believe it has discovered a new question. This approach is unworkable.
- **2.3.2. The Semantic Key (Soft Identity):** The first check for identity is based on meaning. We compute a **weighted cosine similarity** between the five PICO facet vectors of the new question and those of existing questions in the Agenda. The similarity function is weighted to reflect clinical and research importance, with the **Intervention (I)** and **Comparator (C)** facets receiving the highest weight. If this weighted similarity score surpasses a high threshold (e.g., >0.85), the questions are considered strong candidates for a merge.
- **2.3.3. The Evidence Key (Hard Identity):** Once a Sentinel Set S exists for an established question, the identity check becomes even more robust. We can compare the new question to the old one by calculating the **Jaccard overlap** of their respective Sentinel Sets. If two questions are investigated by analyzing largely the same set of primary studies, they are, for all practical purposes, the same question, regardless of minor wording differences.
- **2.3.4. The LLM as Arbiter:** When a new question is a "near-miss"—its semantic similarity is high but not definitive, or its semantic and evidence keys give conflicting signals—the ambiguity is resolved by the LLM. Gemma is presented with a compact brief comparing the two questions' facets and is tasked with assigning a discrete label: same, broader (e.g., new question covers more populations), narrower (e.g., new question has a more specific intervention), or different. The LLM must provide a one-sentence justification naming the specific facet that drove its decision. This judgment is then cached, ensuring the same pair is never arbitrated twice.

Part 3: Deep Dive into Core Mechanisms

This section provides a detailed technical and conceptual explanation of the primary mechanisms that enable the Autonomous Literature Explorer to function. We will dissect the engine of the system, from how it retrieves information to how it makes its most critical judgments.

3.1. Hybrid Retrieval Engine

The system's ability to discover relevant literature begins with a sophisticated retrieval engine that combines the strengths of classical and modern information retrieval techniques. This hybrid

approach is designed to maximize both precision (finding relevant documents) and recall (not missing relevant documents).

- **3.1.1. Lexical Retrieval (BM25): The Power of Keywords**
    - **Concept:** BM25 (Best Match 25) is a state-of-the-art keyword-based ranking function. It improves upon simpler term-frequency models by incorporating two key ideas:
        1. **Inverse Document Frequency (IDF):** It gives more weight to terms that are rare across the entire corpus of documents. The word "losartan" is more informative than the word "treatment," so it gets a higher score.
        2. **Term Frequency Saturation:** It recognizes that the relevance of a term does not increase linearly. The difference in relevance between a term appearing once vs. twice is significant, but the difference between it appearing 20 vs. 21 times is negligible. BM25 models this diminishing return.
    - **Normalization with z-scores:** The raw scores produced by BM25 are not directly comparable from one query to another. To solve this, we standardize the scores for all documents retrieved in a single operational "turn." We calculate the mean and standard deviation of the BM25 scores for the entire pool of candidates and convert each document's score into a **z-score**:
      z = (individual_score - mean_of_pool) / standard_deviation_of_pool
      This expresses each document's lexical relevance in terms of "how many standard deviations above or below the average" it is, creating a universal scale that can be safely combined with other metrics.
- **3.1.2. Dense Retrieval (Qwen3): Capturing Semantic Meaning**
    - **Concept:** This method overcomes the primary limitation of keyword search: its inability to understand synonyms, paraphrasing, and conceptual relationships. The process is as follows:
        1. The Qwen3 embedding model is used to convert a natural language query (e.g., "studies comparing SGLT2 inhibitors to sulfonylureas for cardiovascular mortality") into a high-dimensional vector.
        2. This query vector is then compared against the pre-computed abstract vectors of all documents in the database.
        3. The system retrieves the 'k' documents whose vectors are closest to the query vector, as measured by **cosine similarity**.
    - **Advantage:** This approach can find a paper that discusses "empagliflozin" and "glipizide" in relation to "major adverse cardiac events" even if the query used the broader class names and a different outcome term. It finds documents based on meaning, not just matching words.
- **3.1.3. The Hybrid Ranker: A Weighted, Multi-Factor Score**
  Neither lexical nor dense retrieval is sufficient on its own. The system fuses their signals, along with other heuristics, into a single priority score for each candidate paper. A typical linear combination formula is:
  score = w_1 * cosine_similarity_to_focus_centroid + w_2 * BM25_z_score + w_3 * recency_score + w_4 * type_boost + w_5 * novelty_score
    - cosine_similarity_to_focus_centroid: How semantically close is this paper to the established center of our current topic? This keeps the search focused.

- BM25_z_score: The standardized keyword relevance.
- recency_score: A normalized score where newer papers are weighted more heavily.
- type_boost: A small bonus given to certain publication types depending on the agent's current Mode (e.g., a boost for "RCT" when in Primary-scout mode, or for "Systematic Review" when in SR-hunt mode).
- novelty_score: A measure of how dissimilar the paper is to documents already accepted into the Focus or Sentinel sets, to encourage diversity.

The top-ranked papers from this hybrid scoring process are then passed to the next stage of screening.

3.2. The Sentinel Set Lifecycle: From Nomination to Canonization

The integrity of the entire system rests on the quality of the Sentinel Set (S). Its construction is a rigorous, iterative process designed to produce a high-fidelity proxy for a human-curated list of included studies.

- **3.2.1. The Full Promotion Pipeline:** Each paper considered for inclusion in S must pass through a multi-stage pipeline:
    1. **Step A: Candidate Nomination:** The pipeline is fed with high-scoring candidates from both the Hybrid Retrieval Engine and the Graph Ripple Engine.
    2. **Step B: Fast Metadata Filtering:** The first culling is done using cheap, deterministic checks on the paper's metadata. Any paper with a Publication Type of "Editorial," "Letter," or "Comment," or with a "Retracted Publication" flag, is immediately discarded.
    3. **Step C: The LLM PICO Gate:** This is a crucial step for handling ambiguous cases where metadata is insufficient. For these candidates, a request is sent to Gemma in a large batch. The model's only task is to return a tiny, structured JSON object for each paper with a binary relevance judgment: "is_primary_on_PICO": "yes|no". This leverages the LLM's language understanding for a precise task at a very low computational cost.
    4. **Step D: Deterministic Ranking for Promotion:** All candidates that pass the PICO gate are then ranked for final admission. This ranking is not based on retrieval scores but on a separate set of criteria tailored for building a strong evidence base: weighted PICO facet similarity (emphasizing I and C), preference for stronger study designs (RCT > cohort), and a novelty factor to ensure diversity within the set.
    5. **Step E: Throttled Admission:** To ensure the Sentinel Set grows in a stable and deliberate manner, only the top 'm' ranked candidates (e.g., 5-10) are formally admitted into S during any given episode. This prevents a single large batch of retrieved papers from suddenly and dramatically shifting the topic's evidentiary center.
- **3.2.2. The Three Exhaustiveness Audits: How the System Knows When to Stop**
  The agent does not stop searching for primaries based on a hunch. It ceases exploration only when three independent, quantitative audits signal that the Sentinel Set is functionally complete for the current epoch.
    1. **Audit 1: Discovery Curve Saturation:** The system plots the cumulative number of admitted sentinels against the cumulative number of candidates screened. As the exploration becomes more exhaustive, this curve will flatten. The agent declares saturation when the slope of this curve (the "yield" of new sentinels) falls below a predefined threshold for several consecutive episodes.

2. **Audit 2: Capture-Recapture Logic:** This technique, borrowed from ecology, provides an estimate of the size of an unseen population. The system treats retrieval and ripple as two independent "capture" methods. By analyzing the size of the sets found by each method (A and B) and the size of their overlap (A ∩ B), it can estimate the number of relevant papers missed by both. When this estimated number of "unseen" papers drops to a trivial level (e.g., 1-2), this audit passes.

3. **Audit 3: The Systematic SR-Reference Audit:** The agent actively weaponizes existing literature to find its own blind spots. It retrieves the reference lists of the top candidate SRs on the topic. It then programmatically checks which of those references are on-PICO primaries that are *not* currently in its own Sentinel Set. Any such "missing-from-S" papers are immediately prioritized for inclusion. This audit is complete only when this process yields no new, valid primaries to add.

When all three audits pass, the Sentinel Set is declared "canonized" for the current epoch, and the agent can proceed to a final SR verdict with high confidence.

3.3. Ripple Explosion Control: The Multi-Gate Defense System

Exploring the citation graph is essential for discovery but poses a significant risk of "combinatorial explosion," where the agent is flooded with thousands of tenuously related papers. To manage this, every candidate discovered via ripple is subjected to a strict, multi-layered gating system before it can be considered for inclusion.

- **3.3.1. The Semantic Gate:** The first and most important filter. A candidate paper is only considered if its abstract vector has a cosine similarity to the Focus Set's centroid that is above an **adaptive threshold**. This threshold is not a fixed number; it is dynamically set to a specific percentile (e.g., the 40th percentile) of the similarity scores of recently accepted papers. This allows the gate to be naturally stricter in dense, well-defined topics and more lenient in sparse, emerging fields.
- **3.3.2. The Structural Gate:** This gate ensures that a candidate has a meaningful connection to the body of evidence we already trust (U = S ∪ Focus Set). It uses two metrics:
    - **support_up:** What percentage of the candidate's own references are already in our trusted set U?
    - **support_down:** What percentage of our trusted set U has cited this candidate?
      A candidate must meet a minimum threshold on at least one of these support metrics to pass.
- **3.3.3. The Innovation Budget:** To counteract the conservative bias of the structural gate, a small percentage of the ripple admission quota (e.g., 10%) is reserved for "bridge" papers. These are papers that may have low structural support but exhibit other interesting properties, such as being a semantic outlier within a known cluster or having high betweenness centrality in the graph. This ensures the system remains open to novel, cross-disciplinary connections.
- **3.3.4. Hard Safeguards:** Two non-negotiable rules provide a final backstop:

1. **Degree Caps:** The agent will only process a maximum number of references or citers from any single source paper in one episode (e.g., max 100), regardless of how many it has.

2. **Per-Episode Quotas:** The total number of papers that can be admitted via ripple in a single episode is strictly limited.

- **3.3.5. Hub Attenuation:** To prevent highly-cited "celebrity papers" or guidelines from distorting graph-based metrics like PageRank, the weight of their outgoing edges in the graph is

programmatically down-weighted (e.g., by a factor of $1/\sqrt{\text{outdegree}}$). They remain visible but cannot dominate the structural analysis.

3.4. The SR Verdict Pipeline: From Evidence to Judgment

The system's final judgment on whether an SR exists is the culmination of its evidence-gathering process. It is a formal, two-step procedure.

- **3.4.1. The Evidence Brief:** Once the Sentinel Set is canonized, the Measurement layer compiles a structured, labeled data packet for each top candidate SR. This brief contains no free text, only discrete labels derived from quantitative analysis:
    - coverage: A label (very_high, high, medium, low, unknown) based on the bucketed percentage of the Sentinel Set S that is cited by the SR.
    - P/I/C/O_match: A label (strong, moderate, weak) for each PICO facet, based on the cosine similarity of their respective facet vectors.
    - recency: A label (current, recent, stale) based on the SR's publication year.
    - guideline_like: A boolean flag (yes or no).
    - missing_sentinels: A list of the DOIs of any sentinels the SR failed to cite.
- **3.4.2. The "Point-and-Quote" Discipline:** The Evidence Brief is passed to the Gemma LLM. The model's task is not to re-analyze but to synthesize these labels into a final verdict. Crucially, it must operate under the **"point-and-quote" discipline**. Its output must be a structured JSON object containing a decision (SR_exists, SR_partial, etc.) and a justification that **explicitly quotes the labels** from the brief it received (e.g., Justification: "Verdict based on coverage=high and I-match=strong."). A software validator enforces this rule, ensuring the LLM's reasoning is transparently and directly tied to the measured evidence.

3.5. Key Supporting Technologies Explained

- **3.5.1. Vector Indexing (HNSW on CPU):**
    - **The Challenge:** When you have hundreds of thousands or millions of vectors (one for each paper abstract), finding the nearest neighbors to a query vector by comparing it to every single other vector is computationally infeasible (an O(N) operation).
    - **The Solution: Approximate Nearest Neighbor (ANN):** ANN algorithms build a clever data structure that allows for dramatically faster searching by sacrificing a tiny amount of accuracy. **HNSW (Hierarchical Navigable Small World)** is the state-of-the-art algorithm for this. It organizes the vectors into a multi-layered graph, similar to a highway system with local roads. A search starts on the "highway" layer to quickly find the right general neighborhood, then navigates down to the "local roads" for a precise search in that small area.
    - **The Strategic Choice (CPU over GPU):** While HNSW can run on a GPU, we specify a CPU implementation (e.g., from the FAISS library). The reason is resource allocation. LLM inference is VRAM-hungry and is the primary task for the GPU. A CPU-based HNSW index is exceptionally fast for our scale (sub-millisecond latencies), fits comfortably within standard system RAM, and leaves the entire GPU VRAM budget available for Gemma, preventing resource contention and maximizing overall system throughput.
- **3.5.2. Term Velocity:**

- **The Concept:** A simple yet powerful metric for quantifying the "momentum" of a research concept. It measures the rate of change in the frequency of a term or n-gram's usage in the literature over a recent time window (e.g., the last 24 months).
- **The Calculation:** The system counts the occurrences of key terms in abstracts per quarter. It then fits a simple linear regression to these time-series data points. The slope of that line represents the term's velocity—a positive slope indicates a "hot" or emerging topic, while a negative slope indicates a fading one.
- **The Application:** This metric is primarily used in the Rapid Focus mode. When bootstrapping a new area, the agent can cluster recent papers and then rank the resulting clusters not just by size, but by the aggregate velocity of the terms within them. This allows it to automatically identify and prioritize the most dynamic and currently active areas of research for proposing new PICO questions.

- **3.5.3. Bandit Auto-Tuning:**
  - **The Challenge:** The system has several key parameters (e.g., the strictness of the semantic gate, the structural support threshold α). Manually tuning these for optimal performance is difficult and time-consuming.
  - **The Solution: A Simple Reinforcement Learning Approach:** A "multi-armed bandit" is a class of algorithms that solve the problem of which "lever to pull" to maximize a reward. In our case:
    - The "Arms" are a few discrete settings for a given parameter (e.g., Arm 1: gate=lenient, Arm 2: gate=normal, Arm 3: gate=strict).
    - The "Payoff" is a clearly defined metric of success. We use **New Sentinels per N Reads** (e.g., S/200) because it directly measures the efficiency of converting screening effort into high-quality evidence.
    - The "Algorithm" (e.g., ε-greedy): The agent tries each setting for an episode, observes the payoff, and over time, it learns to predominantly choose the setting that has historically yielded the best results, while still occasionally "exploring" the other options to ensure it can adapt if conditions change. This provides a lightweight, online method for auto-tuning the system's core behaviors without requiring any complex model training.

Part 4: The Agent's Operational Logic

This section describes the dynamic "verb" layer of the system—how the agent perceives its environment, makes decisions, and executes actions in a continuous, adaptive cycle.

4.1. The Episodic Loop: A Cycle of Perception, Planning, and Action

The agent's operation is not a single, linear pipeline but a series of discrete, iterative work cycles called **episodes**. Each episode is a self-contained loop of sensing, thinking, and acting, designed to make incremental progress towards a specific goal. This structure makes the agent's behavior manageable, measurable, and adaptable.

- **4.1.1. The Instrument Panel: The Agent's Worldview**
  At the beginning of every episode, the Cognition Layer (Gemma) does not access the entire database. Instead, it is presented with a compact, structured JSON object known as the Instrument Panel. This digest provides a complete, high-level summary of the current state of the investigation for the active Research Question. A full specification includes:

- **Identity:** The canonical one-line PICO and the rq_id.
- **Core State:**
  - focus_set_summary: A snapshot of the top 5 anchor papers (title, year, type).
  - sentinel_set_metrics: Current size of the Sentinel Set (S) and the number of new sentinels added in the previous episode.
- **SR Status:**
  - sr_verdict_status: The current verdict (exists, partial, no_sr, or pending).
  - best_sr_candidate: The DOI of the leading candidate SR.
  - best_sr_evidence_labels: The labeled evidence brief (coverage, PICO_match, etc.) for the leading candidate.
- **Performance Feedback:**
  - discovery_metrics: The acceptance_rate (relevant papers / screened papers) and novelty_percentile from the last episode.
  - shadow_recall: The percentage of new sentinels from the last episode that were found *only* by graph ripple (a key indicator of query weakness).
- **Operational Settings & Triggers:**
  - current_gate_labels: The current strictness settings (strict, normal, lenient) for the semantic and structural gates.
  - active_triggers: A list of flags for notable events, such as saturation_detected, contradiction_index_high, or new_citers_spike.
  - episode_quotas: The hard limits for the upcoming episode (e.g., max_nodes_to_admit, max_abstracts_to_read).

- **4.1.2. The Mode System: Goal-Oriented Policies**
  Based on its interpretation of the Instrument Panel, the LLM's first decision is to select an operational **Mode**. A mode is not a rigid script but a high-level policy that defines the primary objective for the episode. The available modes include:
  - Primary-scout: The main goal is to find and promote new primary studies to grow the Sentinel Set. This mode prioritizes discovery and recall.
  - SR-hunt: The main goal is to find and evaluate candidate Systematic Reviews against the current Sentinel Set to reach a coverage verdict.
  - Gap-scan: The main goal is to analyze the topology and metadata of the currently accepted papers to identify methodological, population, or comparator gaps.
  - Contradiction-probe: Triggered when conflicting evidence is detected, the goal is to specifically search for and analyze papers on both sides of a disputed claim to adjudicate the conflict.
  - Update-hunt: Triggered after an SR is found to be outdated, the goal is to find all relevant primary studies published *after* the SR's search window.
  - Rapid-Focus: The initial bootstrapping mode, designed to quickly generate a candidate PICO from a broad, recent slice of literature.

- **4.1.3. The Plan: The LLM's Actionable Commands**
  After selecting a mode, the LLM generates a **Plan**. This is a short, structured list of commands for the Orchestrator to execute. It is not free-form text but a precise sequence of tool calls with associated parameters and quotas. A typical plan might look like:

1. set_gates(semantic='normal', structural='lenient')
2. retrieve(query_type='hybrid', query_text='...', k=500)
3. ripple(source_nodes=['doi1', 'doi2'], direction='references', credits=50)
4. promote_to_sentinel(max_admissions=5)
5. evaluate_sr_candidates()

- **4.1.4. The Execution Phase: The Orchestrator's Role**
  The Memory & Governance Layer's **Orchestrator** takes the LLM's Plan and executes it. It is the system's "enforcer." It translates the plan's commands into calls to the Perception Layer's tools, rigorously applies all gating logic, and strictly enforces the specified quotas. If the retrieve call returns 10,000 results but the plan's k is 500, the Orchestrator truncates the results and logs this fact. At the end of the episode, it updates the database with all new papers, edges, and decisions, and prepares a new Instrument Panel for the next cycle. This closed loop of Perceive (Panel) -> Think (Plan) -> Act (Execute) allows the agent to learn and adapt its strategy with each turn.

4.2. Data Integrity and Persistence

To ensure the system's long-term reliability and the auditability of its findings, a strict and well-defined approach to data management is essential.

- **4.2.1. The DOI-First Identifier Policy**
  The **Digital Object Identifier (DOI)** is the canonical, immutable identifier for all scholarly works in the system.
  - **Ingestion Requirement:** Any paper entering the system *must* have a resolvable DOI. Records without a DOI are discarded at the source.
  - **PMID Mapping:** PubMed IDs (PMIDs) are treated as secondary identifiers. The system maintains a two-way cache mapping DOIs to PMIDs to facilitate interaction with PubMed's APIs, but the DOI remains the primary key in the database.
- **4.2.2. Minimal Viable Strategies for Retraction and Preprint Handling in the PoC**
  While complex data curation is deferred to later versions, two minimal integrity checks are essential for the PoC to ensure the quality of the Sentinel Set.
  - **Retraction Handling:** During the Fast Metadata Filtering stage of the sentinel promotion pipeline, any paper whose PubMed metadata includes the Publication Type "Retracted Publication" or whose title is explicitly prefixed with [Retracted] is immediately and permanently disqualified from promotion. No further checks are performed in the PoC.
  - **Preprint Handling:** To handle the most common form of study duplication, if a paper's metadata (from Crossref or PubMed) explicitly declares a relationship to another DOI (e.g., is-preprint-of, is-published-as), the system will flag the preprint as an alias and treat the peer-reviewed journal article as the canonical record for that study. No fuzzy matching or heuristic-based clustering is performed in the PoC.

- **4.2.3. High-Level Database Schema Outline**
  The system's memory is persisted in a relational database structured to support its operational logic. The core tables include:
    - papers: Stores the canonical record for each paper, keyed by DOI, with all associated metadata.
    - embeddings: Stores the pre-computed abstract and PICO facet vectors for each paper, versioned by the model used to generate them.
    - edges: A table storing all graph relationships (e.g., source_doi, target_doi, edge_type='cites').
    - agenda: The master table for all Research Questions, keyed by a unique rq_id, containing the canonical PICO and current status.
    - sentinels: A mapping table linking rq_id to the DOIs of the papers in its Sentinel Set.
    - episodes: A log of every operational cycle, storing the Instrument Panel received by the LLM, the Plan it generated, and the final measured outcomes.
    - decisions: A table that logs every high-stakes judgment made by the LLM, such as SR verdicts and identity arbitrations, along with the evidence brief that supported the decision.

Part 5: Proof-of-Concept (PoC) Strategy

This section outlines the staged, empirical strategy for validating the architecture and de-risking the project. The Proof-of-Concept (PoC) phase is not about building the entire system at once, but a series of targeted experiments designed to test the most critical and innovative components of the architecture in isolation and in concert. The entire strategy adheres to the project's core principles: fully autonomous operation and evaluation using objective, automatable metrics that do not require subjective human input or pre-labeled "gold standard" datasets.

5.1. Evaluation Methodology without Human Gold Standards

Trust in an autonomous system cannot be assumed; it must be earned through rigorous and transparent evaluation. Lacking access to panels of human experts to label data for us, we will employ a suite of robust, data-driven techniques to measure the agent's performance, stability, and intelligence.

- **5.1.1. Using "Silver Standard" Systematic Reviews for Back-Testing:** While perfect "gold standards" are rare, the scientific literature provides abundant "silver standards." For a given research topic, we can find a recently published, high-quality systematic review that publicly lists its final set of "included studies" (often in an appendix or supplementary file). This list of DOIs serves as an excellent, objective benchmark against which we can measure our agent's performance.
    - **Primary Metric: Sentinel Recall @ Episode k:** This quantifies the agent's discovery power. We measure the percentage of the silver standard's included studies that our agent has successfully found and promoted into its own Sentinel Set after a specific number of operational episodes (k). A high recall indicates the agent is effectively finding the same core evidence as human experts.
    - **Secondary Metric: Time-to-Coverage:** This quantifies the agent's efficiency. We measure the number of episodes (or total papers screened) required for the agent to reach a state where its Sentinel Set has a high overlap with the silver standard's list.

- **5.1.2. Self-Consistency and Stability Tests:** A reliable autonomous agent must be robust to minor variations in its starting conditions. To test this, we will execute runs on the same Research Question multiple times, each with a different random seed or a slightly different initial set of retrieved papers.
    - **Metric: Jaccard Similarity of Final Sentinel Sets ($J(S_1, S_2)$):** We will compare the final Sentinel Sets produced by these parallel runs. A high degree of overlap (a high Jaccard index) demonstrates that the agent's discovery process is convergent and stable, not chaotic or path-dependent.
- **5.1.3. Analysis of Internal Learning Curves:** Our agent is designed to learn and adapt its strategy over time. We can directly visualize and quantify this learning process by tracking key internal metrics throughout a run.
    - **Metric: Acceptance Rate vs. Shadow Recall:** We will plot two key performance indicators episode-by-episode. The acceptance rate (the percentage of screened papers deemed relevant) should trend upwards as the agent's queries improve. Concurrently, shadow recall (the percentage of new sentinels found *only* via graph ripple) should trend downwards, indicating that the queries are becoming more effective at capturing what was previously hidden in the citation network. This opposing trend is a clear signature of successful, adaptive learning.

5.2. The Staged PoC Implementation Plan

The following is a sequence of independent PoCs, each designed to test a specific architectural component or hypothesis.

- **PoC-0: Foundational Infrastructure & Data Flow**
    - **Goal:** To perform a "smoke test" of the absolute foundational layer of the system, ensuring data can be ingested, represented, and queried correctly before any complex logic is built upon it.
    - **Procedure:** A small corpus of ~50,000 papers will be ingested. The system will build the BM25 lexical index and the HNSW approximate-nearest-neighbor vector index. A single hybrid query will be executed.
    - **Pass/Fail Metrics:** The test passes if (a) the indices are built without error, (b) the DOI-to-PMID mapping cache is populated correctly, and (c) a hybrid query returns a ranked list of documents with valid scores within an acceptable latency threshold (e.g., <100ms for the vector search).
- **PoC-1: Retrieval + Ranking Quality**
    - **Goal:** To prove that the specified hybrid ranking formula provides a tangible benefit over its constituent parts.
    - **Procedure:** On a defined topic, run three separate retrieval passes: BM25 only, dense search only, and the full hybrid ranking. Compare the position of a known set of 5-10 relevant "seed" DOIs in each ranked list.
    - **Pass/Fail Metrics:** The test passes if the hit@k (the number of seed DOIs found in the top k results) for the hybrid ranker is superior to both baselines for a range of k values (e.g., k=50, 100, 200).
- **PoC-2: LLM PICO Gate Throughput & Accuracy**
    - **Goal:** To validate the core hypothesis that the Gemma LLM can function as a fast and accurate relevance classifier, a critical task for the Sentinel Set promotion pipeline.

- **Procedure:** A set of 500 abstracts, including many with known Publication Types from PubMed, will be processed through the batched PICO gate.
- **Pass/Fail Metrics:** The test passes if (a) the system achieves a high throughput (e.g., >50 abstracts/second on the target hardware), demonstrating its efficiency, and (b) the LLM's classification of study design shows high agreement (>85%) with the available PubMed metadata ("silver" precision).

- **PoC-3: Sentinel Growth and Exhaustiveness Audit Validation**
  - **Goal:** To demonstrate the end-to-end dynamic of the Sentinel Set's lifecycle—that it grows in a controlled manner and that the system can correctly identify when it has reached a state of practical completion.
  - **Procedure:** The agent will run on a single RQ for 5-7 full episodes. After each episode, the size of the Sentinel Set will be logged, and the SR-Reference Audit will be performed to find and queue any "missing-from-S" studies.
  - **Pass/Fail Metrics:** The test passes if (a) the "new sentinels per read" discovery curve visibly flattens over the episodes, indicating saturation, and (b) the number of new, valid primaries discovered by the SR-Reference Audit trends towards zero, confirming the audit's gap-closing function.
- **PoC-4: Ripple Gating Ablation Test**
  - **Goal:** To scientifically validate the effectiveness of our multi-layered ripple control system.
  - **Procedure:** This PoC is an ablation study. The agent will be run on the same RQ four times, each with a different gate configuration: (a) only hard degree caps, (b) semantic gate only, (c) semantic + structural gates, (d) all gates including the innovation budget.
  - **Pass/Fail Metrics:** The test passes if configuration (c) or (d) demonstrates the best overall efficiency, measured as the highest ratio of new sentinels per read while keeping the total number of admitted nodes within a bounded limit. This will provide quantitative proof of each gate's contribution.
- **PoC-5: The End-to-End SR Verdict Pipeline**
  - **Goal:** To test the integrity of the system's final judgment mechanism, ensuring it is evidence-based and auditable.
  - **Procedure:** A test case will be prepared with a pre-populated Sentinel Set (~15 DOIs) and 3-5 candidate SRs. The system will execute the full verdict pipeline: generating the labeled Evidence Brief for each SR and then passing it to the LLM for the final verdict.
  - **Pass/Fail Metrics:** The test passes if (a) the LLM's JSON output is always structurally valid and strictly adheres to the "point-and-quote" discipline (verified by the software validator), and (b) the final verdict is stable and consistent across repeated runs on the same input.
- **PoC-6: Update Trigger ("Greediness") Policy Test**
  - **Goal:** To verify the agent's ability to make a higher-level strategic shift based on its configured "greediness" policy.
  - **Procedure:** Using the setup from PoC-5, we will programmatically add two "new" sentinel studies published after the best SR's search date. We will then run the Update Trigger calculation under both a "strict" and a "greedy" policy setting.

- **Pass/Fail Metrics:** The test passes if the agent's next planned Mode correctly switches to Update-hunt under the "greedy" policy but remains concluded under the "strict" policy.

- **PoC-7: Rapid Focus and Term Velocity**
  - **Goal:** To demonstrate that our Term Velocity heuristic provides a tangible advantage in bootstrapping new research questions.
  - **Procedure:** The Rapid Focus mode will be run twice on a broad topic. The first run will seed the new RQ from a randomly selected cluster of recent papers. The second run will seed it from the cluster with the highest aggregate term velocity. We will then compare the efficiency of the first full discovery episode that follows each start.
  - **Pass/Fail Metrics:** The test passes if the start seeded by the high-velocity cluster leads to a measurably higher discovery yield (new sentinels per read) in the subsequent episode, proving the heuristic's value.
- **PoC-8: Identity & Epochs**
  - **Goal:** To validate that the agent's identity arbitration logic for RQs functions correctly.
  - **Procedure:** A set of 6 synthetic PICO definitions will be created, with controlled, subtle variations between pairs. The system will compute their facet similarities and pass them to the LLM for an identity judgment.
  - **Pass/Fail Metrics:** The test passes if the LLM's labels (same, broader, narrower) and justifications correctly align with the controlled edits (e.g., correctly identifying that a change in the Comparator facet makes two questions different).
- **PoC-9: Contradiction-Probe Micro-Test**
  - **Goal:** To demonstrate the functionality of the contradiction detection and adjudication mechanism in a controlled environment.
  - **Procedure:** A small set of ~12 primary studies with pre-defined, conflicting "claim stubs" will be ingested. The system will compute the contradiction_index, trigger the Contradiction-probe mode, and task the LLM with adjudicating the conflict.
  - **Pass/Fail Metrics:** The test passes if the agent correctly triggers the mode and the LLM's final adjudication accurately identifies the source of the conflict (e.g., "apparent contradiction due to different outcome measures") with a concise rationale.
- **PoC-10: Bandit Auto-Tuning**
  - **Goal:** To verify that the simple ε-greedy bandit can effectively optimize a key system parameter online.
  - **Procedure:** Over a 6-episode run, the agent will test three different settings for the semantic gate strictness. The first three episodes will explore each setting once. The subsequent three will predominantly use the setting that yielded the highest new sentinels per 200 reads, with a small chance of exploring others.
  - **Pass/Fail Metrics:** The test passes if the average discovery efficiency in the exploitation phase (episodes 4-6) is superior to the best single setting from the exploration phase, demonstrating that the bandit is successfully learning and adapting.
- **PoC-11: Performance and Resource Profiling**

- **Goal:** To establish a definitive performance and resource consumption baseline for the entire system operating on the target RTX 4050 hardware.
- **Procedure:** A standard 5-episode run will be executed on a single RQ, with detailed logging of all performance counters.
- **Pass/Fail Metrics:** This PoC does not have a binary pass/fail but produces a critical report card of the system's efficiency, documenting: wall-clock time per episode, LLM inference throughput (TTFT and tokens/sec), HNSW query latency, and peak CPU, RAM, and GPU VRAM usage. This data will confirm the practical feasibility of the architecture.