Agentic AI Extension for RAG Hiring System

Purpose

This document describes an **agentic AI** approach to address remaining limitations of the tool-augmented RAG hiring system: reactivity, shallow memory, limited adaptability, and narrow autonomy. The agentic design composes planner, persistent memory, tool adapters, and monitoring agents to create a safe, auditable, and semi-autonomous hiring assistant.

Recap: Problems to Solve

- **Not proactive:** system requires triggers; it does not plan or anticipate.
- **Limited continuous memory:** data stored but not synthesized into long-term policies or preferences.
- **Limited adaptability:** cannot alter strategy over time (e.g., channel mix, seniority targets).
- **Narrow autonomy:** tool actions are rule-based and brittle in ambiguous HR scenarios.

Agentic AI — Overview

Agentic AI = **autonomous (but controllable) agents** that can plan multi-step workflows, call tools, read/write memory, and request human approval when needed. Key properties introduced:

- **Planning**: generate multi-step plans (goals → subtasks → tool calls).
- **Persistent & semantic memory**: episodic + aggregated summaries stored in vector DB and structured DB.
- **Tool adapters**: robust connectors to LinkedIn, Mail, Calendar, HRM, Resume Parser.
- **Monitoring and rollback**: watch agents supervise actions and can roll back or escalate.
- **Human-in-the-loop gates**: configurable thresholds that require approval for risky actions.

System Architecture (Visual)

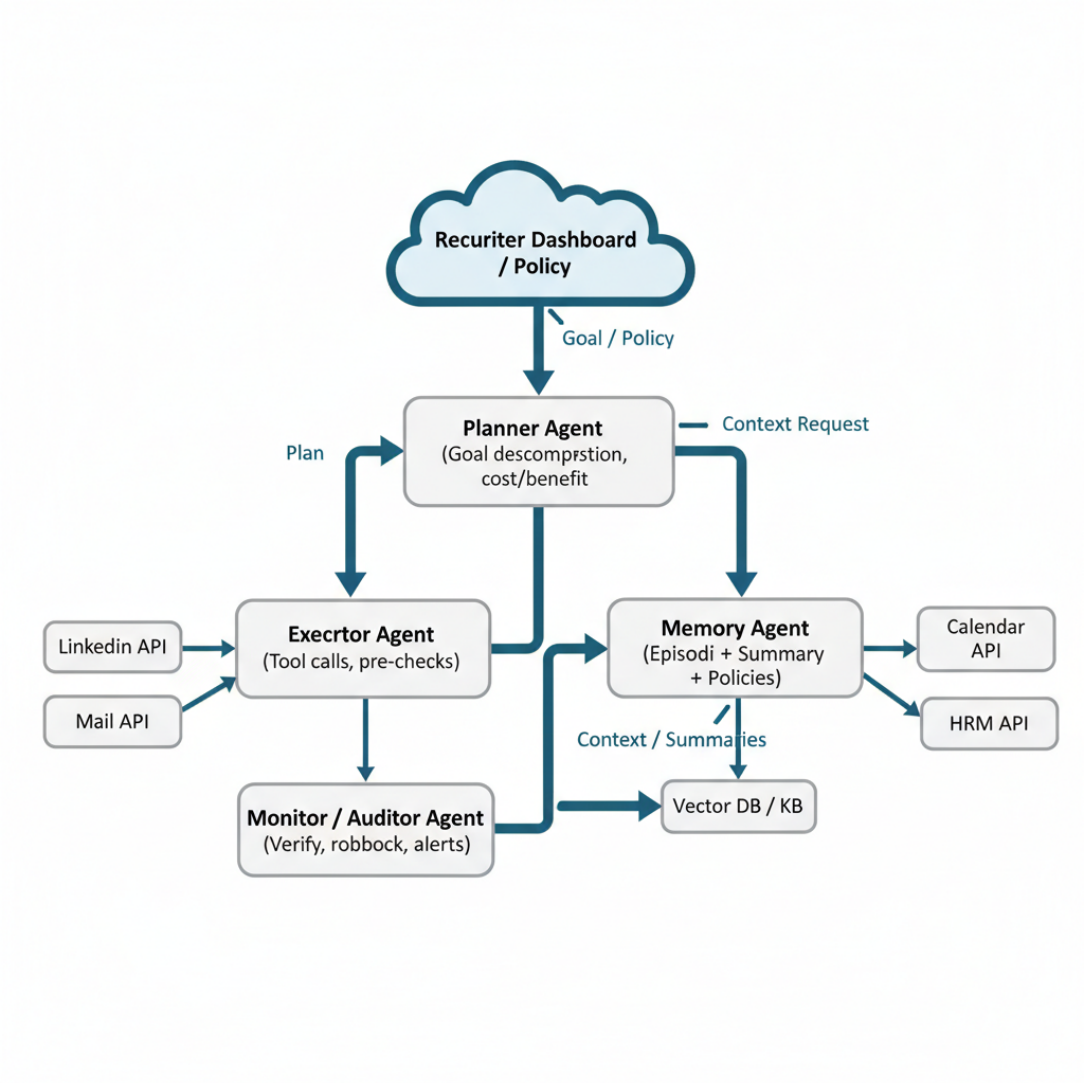

Figure 1: Agentic Approach

Agent Behaviors and Roles

Planner Agent

- Input: high-level goal (e.g., "Hire Backend Engineer in 60 days").
- Output: ordered plan with subtasks and required evidence (e.g., draft JD → post to LinkedIn → collect 50 applicants → shortlist top 10).
- Decides human approval points and risk level per action.

Executor Agent

- Performs deterministic tool calls using tool adapters.
- Validates outputs against preconditions (schema checks, quotas).
- Records action, tool response, and provenance in State DB.

Memory Agent

- Stores episodic logs (what actions were taken, timestamps, outcomes).
- Produces condensed persistent summaries (weekly hiring policy changes, channel performance).
- Exposes APIs for retrieval with relevance and recency filters for Planner.

Monitor / Auditor Agent

- Validates action outcomes, detects anomalies (e.g., many failed posts), and triggers rollbacks or human escalation.
- Tracks compliance, logs approvals, and maintains audit trails.

Agent Planning Loop (runtime)

1. **Goal received** by Planner (from recruiter or policy rule).
2. Planner **retrieves context** (Memory Agent + Vector DB) and generates a candidate plan.
3. Planner marks tasks as: `autonomous`, `pending_approval`, or `manual`.
4. Executor runs autonomous tasks (tool calls) and records outputs to State DB.
5. Monitor validates outputs; on anomaly, either retries, rolls back, or escalates.
6. Memory Agent updates episodic log and updates long-term summaries.
7. Planner re-plans if outcomes deviate from expectations.

Executor Contract (example)

- All tool calls must return structured responses with success/failure, error codes, and provenance. - On failure, return exact error and suggested remediation steps.

Safety, Governance & Human-in-the-loop

- **Approval policies:** Define which actions require explicit human approval (offers, salary overrides, public posts).
- **Explainability:** Planner and Memory must record rationale and evidence used for every decision.
- **Rate limits / quotas:** Enforce via Executor to avoid accidental spam or policy breaches.
- **Rollback hooks:** For critical tool actions, implement compensation logic (delete post, cancel invite).
- **Audit logging:** Persist prompts, plans, approvals, tool responses, and final outcomes in immutable logs.

Implementation Checklist (MVP Agentic)

1. Implement Memory Agent (Vector DB + summary pipeline).
2. Implement Planner Agent (LLM + planning schema enforcement).
3. Build Executor with tool adapters (LinkedIn, Mail, Calendar, HRM, Resume Parser).
4. Implement Monitor Agent with anomaly detection and rollback capabilities.
5. Add human approval UI and RBAC.
6. Add audit logging, encryption, and compliance checks.
7. Run controlled pilot with simulated traffic and predefined safety gates.

Example Use Cases Enabled

- **Proactive sourcing:** Planner schedules periodic re-posting or new channel experiments when applicant velocity drops.
- **Policy-aware offers:** Executor populates offer templates but requires HR approval for salary exceptions.
- **Continuous improvement:** Memory summaries surface that e.g., "LinkedIn yielded better senior candidates in Q3" and Planner adjusts channel allocations.

Conclusion

Agentic AI augments RAG by adding planning, persistent semantic memory, and safe tool execution. This approach moves the system from reactive advice + isolated actions to a coordinated, auditable, and semi-autonomous hiring assistant — while preserving human oversight for high-risk decisions.