RAG Flywheel Source-Priority and Curated Seed

RAG Flywheel Source-Priority and Curated Seed

MC Task: #103899
Status: Complete, Proveo-validated PASS
Date: 2026-06-18

Problem

The RAG cache (`~/system/databases/flywheel.db`) contained 75K+ entries, with 99.96% originating from youtube-learning sources. Only 38 entries had ever been reused (hit_count > 0).

Critical failure mode: Paraphrased ALAI-specific questions returned YouTube answers instead of curated ALAI facts. Example: A query about LightRAG VM location matched a YouTube entry at 0.731 similarity, while the correct curated fact scored 0.688 — below the global 0.70 threshold, so it was never served.

Fix: Dual-Threshold + Source-Priority Ranking

How It Works

The `rag-router.js query()` method now:

1. **Partitions cache matches** into curated vs non-curated sources
2. **Applies source-appropriate thresholds:**
   - Curated sources: **0.60** similarity threshold (configurable via `RAG_CURATED_THRESHOLD`)
   - Non-curated (YouTube): **0.70** threshold (existing `RAG_CACHE_THRESHOLD`)

3. **Source-priority selection:** If a curated source match exists above 0.60, it pre-empts higher-similarity non-curated matches

Environment Toggles

- `RAG_SOURCE_PRIORITY=true` (default) — Enable source-priority ranking
- `RAG_CURATED_THRESHOLD=0.60` (default) — Threshold for curated sources
- `RAG_CACHE_THRESHOLD=0.70` (default) — Threshold for non-curated sources

Implementation

Code location: `~/system/tools/rag-router.js`

- Lines 58-62: Constants defining thresholds and curated source list
- Lines 369-446: Source-priority partitioning and selection logic
- Lines 921-932: Extended `learn` CLI to accept `--source` flag

Curated Sources Taxonomy

Source Tag	Meaning	Threshold
`alai-curated`	Manually verified ALAI-specific facts (institutional knowledge)	0.60
`cli`	Manual entry via `rag-router learn` command	0.60
`capture`	Manual session capture	0.60
`session`	Session-extracted knowledge (manual)	0.60
`auto-local-raw`	Auto-indexed local model responses	0.60
`auto-local-enriched`	Auto-indexed knowledge-base-enriched responses	0.60
`manual`	Other manual curation	0.60
`youtube-learning*`	YouTube transcript index	0.70

Principle: Curated sources (human-verified or ALAI-domain-filtered) use a lower threshold (0.60) for higher recall. Generic/auto sources require stricter matching (0.70).

How to Seed Curated Knowledge

Use the `learn` CLI with the `--source` flag:

```
node ~/system/tools/rag-router.js learn "Question text" "Answer text" --source alai-curated
```

Guidance:

- Only seed **verified ALAI-specific facts** from authoritative sources:
  - `~/system/agents/specialist-mapping.json`
  - `~/.claude/CLAUDE.md`
  - `~/system/BUILD-BLUEPRINT.md`
  - Memory files in `~/.claude/projects/-Users-makinja/memory/`
  - BookStack documentation
- **Never invent facts** or seed generic knowledge (use YouTube sources for that)
- Keep answers specific, evidence-backed (paths, names, endpoints)
- Avoid hedging language ("generally", "typically") — curated facts should be definitive

Validation Results

Independent verification by Proveo: PASS all 6 acceptance criteria

AC	Description	Result
AC1	Curated paraphrase query returns alai-curated/cli source	PASS
AC2	YouTube-only topic still routes via YouTube (threshold intact)	PASS
AC3	9 alai-curated rows seeded with real ALAI content	PASS
AC4	YouTube count unchanged (~75K), no deletions	PASS
AC5	Curated match at 0.663 served (was blocked at 0.70 before)	PASS
AC6	Auto-loop plan doc exists (plan-only, no build)	PASS

Seeded Facts (IDs #414189–414197)

1. **LightRAG location:** Azure VM vm-alai-lightrag (20.240.61.67), access via az vm run-command
2. **FORGE Ollama endpoint:** 10.0.0.2:11434, primary models (qwen3-coder:30b, qwen3:32b, deepseek-r1:70b)
3. **ALAI Holding AS identity:** AI-driven dev agency, CEO Alem Basic, values, philosophy
4. **Specialist companies:** 7 companies (CodeCraft, Vizu, FlowForge, Proveo, Securion, AgentForge, Finverge, Skybound)
5. **John's role:** AI Director, orchestrator, delegates to specialists, does not build

6. **ZAKON NULA:** Tool-first enforcement, forbidden to answer from LLM memory
7. **Mission Control:** Database location, CLI commands
8. **Mehanik gate:** Pre-dispatch gate for H/BLOCKER tasks, verification steps
9. **CodeCraft:** Backend/architecture company, key specialists

Evidence: `/tmp/verify-103899/VALIDATION-REPORT.md`

Known Limitations

Shadow Log Misattribution (Low Severity)

Issue: The `shadow_log` table records `best_cache_id` as the globally highest-similarity candidate, not the actually-selected match when source-priority routing overrides raw similarity ranking.

Example: For a LightRAG query, shadow_log shows YouTube entry 359004 (similarity 0.723) but the actual response came from curated cli entry 414082 (similarity 0.663).

Impact: Routing correctness is **not affected**. Shadow log audit trails are misleading for source-priority queries. Analytics/auditability impaired.

Follow-on fix tracked separately (Low priority).

Auto-Loop Not Yet Built

The automatic flywheel indexing system (session extraction, LightRAG writeback) is **plan-only** in this MC. Implementation deferred to future work.

Plan document: `~/system/specs/rag-flywheel-auto-loop-plan.md`

The plan covers:

- Session extraction trigger (auto-extract Q&A pairs from completed sessions)
- Flywheel indexer daemon (`~/system/daemons/flywheel-indexer.js`)
- LightRAG writeback integration (push proven facts to graph)
- Quality gates (confidence assessment, deduplication)
- Phased rollout (Phase 1–3 pending)

References

- **Code:** `~/system/tools/rag-router.js`
- **Validation report:** `/tmp/verify-103899/VALIDATION-REPORT.md`
- **Build evidence:** `/tmp/evidence-103899/verification.md`

- **Auto-loop plan:** `~/system/specs/rag-flywheel-auto-loop-plan.md`
- **MC task:** #103899

Revision #1
Created 2026-06-18 14:00:15 UTC by John
Updated 2026-06-18 14:00:15 UTC by John