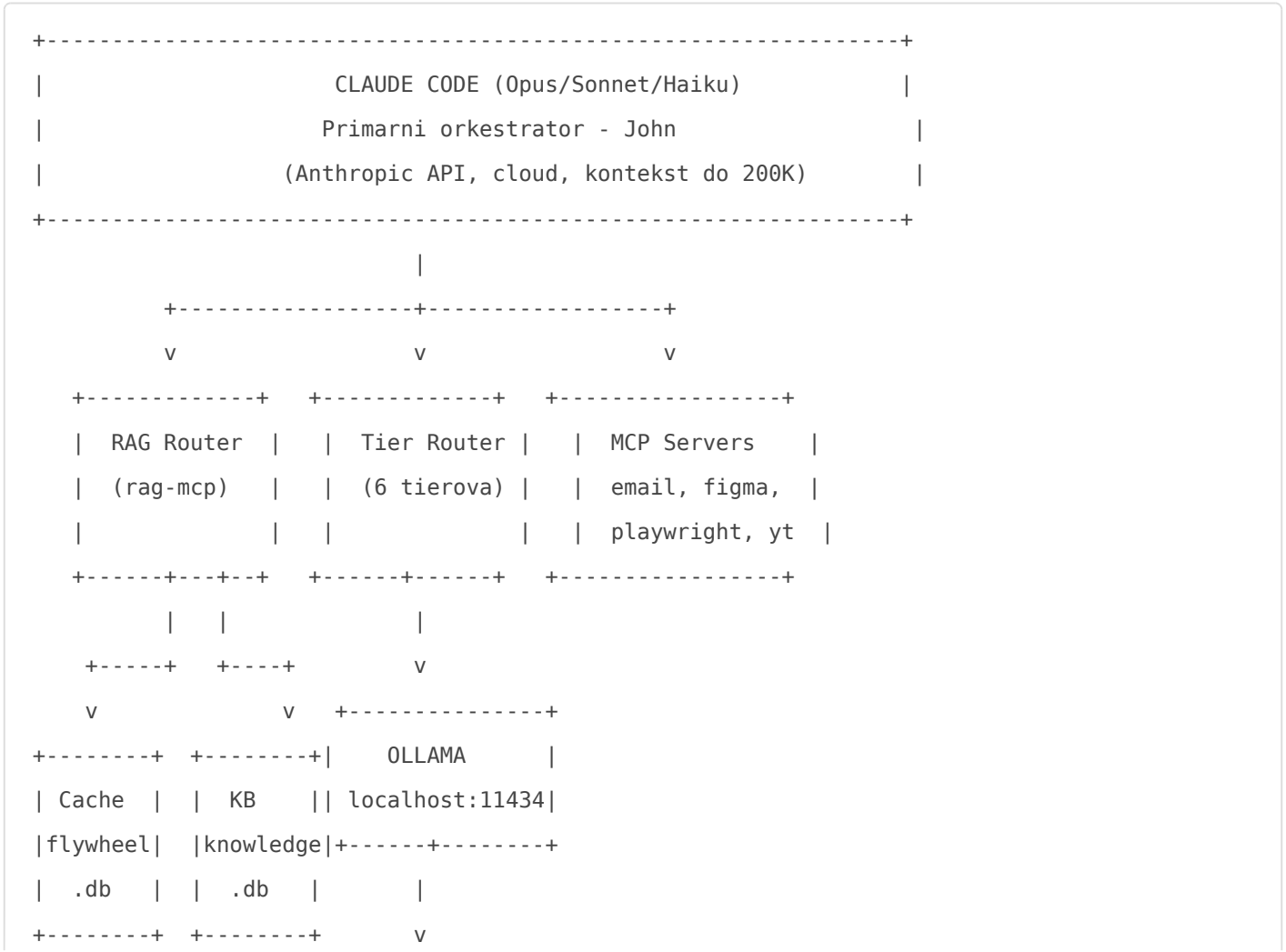
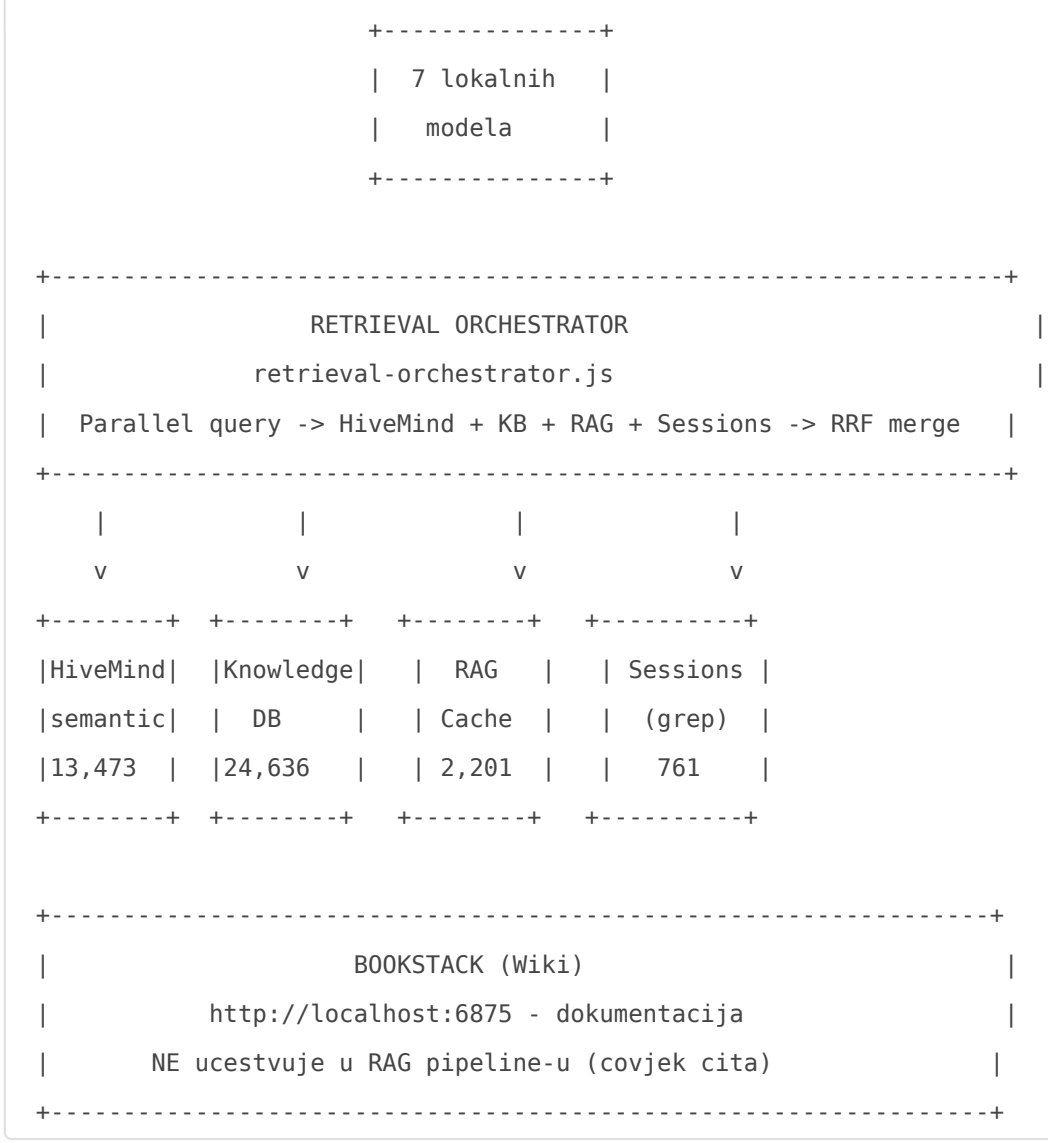
AI Model & RAG Architecture

AI Model & RAG Architecture

> Pregled svih AI modela i RAG (Retrieval-Augmented Generation) komponenti u ALAI sistemu. Datum: 2026-02-23. Izvor: verifikovan inventar iz filesystem-a i running servisa. Zadnji update: RAG System Upgrade (MC #1804) — unified embedding, HiveMind vector search, retrieval orchestrator, session archiver.

Pregled na jednoj stranici





1. Lokalni AI modeli (Ollama)

Server: `http://localhost:11434` **Hardware:** Mac Studio M3 Ultra, 96 GB RAM **LaunchAgent:** `homebrew.mxcl.ollama` **Config:** `~/system/config/ollama.json`

Instalirani modeli (ollama list, 2026-02-21)

Model	Velicina	Namjena	Status
`llama3.1:8b`	4.9 GB	Brzi classify/extract/filter (Tier 1)	AKTIVAN
`qwen2.5-coder:32b`	19 GB	Code review, debug, refaktor (Tier 2c)	AKTIVAN
`nomic-embed-text`	274 MB	Embeddings - 768-dim vektori za RAG	AKTIVAN

Model	Velicina	Namjena	Status
`alaiml-task-v1`	986 MB	Fine-tuned za MC task handling (Tier 2t)	AKTIVAN
`alaiml-tender-v1`	986 MB	Fine-tuned za tender analizu	AKTIVAN
`alaiml-email-v1`	986 MB	Fine-tuned za email klasifikaciju	AKTIVAN
`llama-guard3:8b`	4.9 GB	Content safety / guardrails	AKTIVAN

Konfigurirani ali NE instalirani

Model	Razlog	Napomena
`llama3.1:70b`	42 GB - ne stane uvijek u RAM	U config-u kao Tier 3 (complex reasoning)
`qwen2.5:72b`	47 GB - ne stane uvijek u RAM	U config-u kao Tier 2 (general)

Wrapper toolsi:

- `~/system/tools/ollama-engine.js` - HTTP wrapper za generate/classify
- `~/system/tools/ollama-tool-agent.js` - Multi-turn agent sa READ-ONLY toolsima
- `~/system/tools/agent-runner.js` - Agent lifecycle (identity -> state -> HiveMind -> Ollama -> save)

---

2. Tier Routing (Task -> Model dispatch)

File: `~/system/tools/tier-router.js` **Config:** `~/system/config/tier-routing.json`

Svaki AI request ide kroz routing koji odlucuje koji model procesira:

Tier	Engine	Model	Namjena
1	Ollama	llama3.1:8b	Trivijalno: classify, filter, extract
2	Ollama	qwen2.5:72b*	Medium: summarize, draft, analyze
2c	Ollama	qwen2.5-coder:32b	Code: review, debug, simple fix

Tier	Engine	Model	Namjena
2t	Ollama	alaiml-task-v1	Task-specific: MC task handling
3	Ollama	llama3.1:70b*	Complex reasoning (NO code execution)
4	Human Queue	-	Critical: multi-file, architecture, decisions

Tier 2 i 3 modeli nisu trenutno instalirani. Fallback na Tier 2c.

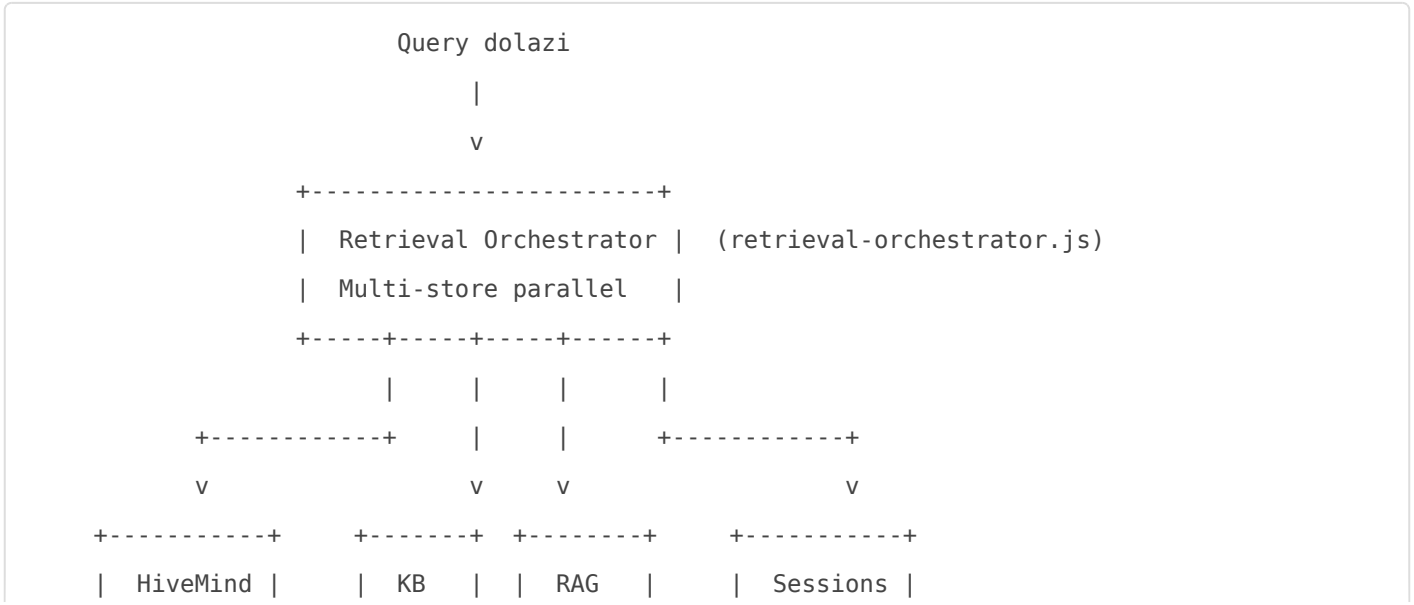
Routing logika

1. **Caller-based** - svaki daemon/agent ima fiksni tier:
   - email-agent, pipeline-watcher -> Tier 1
   - morning-routine, explore -> Tier 2
   - autowork-standard, validator -> Tier 2c
   - builder, interactive -> Tier 4 (human/Claude)
2. **Keyword fallback** - skenira task tekst za keyword match
3. **Default** - Tier 2

---

3. RAG System (Retrieval-Augmented Generation)

3.1 Arhitektura (v2, 2026-02-23)



```
    |  semantic |     | docs  |  | cache  |     |   grep    |
    |  13,473   |     |24,636 |  | 2,201  |     |    761    |
    +-----------+     +-------+  +--------+     +-----------+
           |               |          |
           +-------+-------+----------+
                   v
           +---------------+
           |  RRF Merge    |  Reciprocal Rank Fusion (k=60)
           |  Deduplicate  |
           +-------+-------+
                   |
                   v
            Top N results
```

Retrieval flow:

1. **Embed query** jednom (nomic-embed-text, 768-dim)
2. **Parallel query** svih 4 storea (HiveMind semantic, Knowledge DB, RAG Cache, Sessions grep)
3. **RRF Merge** — Reciprocal Rank Fusion kombinira rankings iz svih izvora
4. **Return** top N rezultata sa RRF score + source attribution

Inspirirano: Spring AI Modular RAG (RetrievalAugmentationAdvisor + MultiQueryExpander + ConcatenationDocumentJoiner)

3.2 Retrieval Orchestrator (NOVO, 2026-02-23)

File: `~/system/tools/retrieval-orchestrator.js` **MC Task:** #1804

Centralni entry-point za sav retrieval u sistemu. Umjesto rucnog "BookStack PRVO -> HiveMind -> etc", orchestrator automatski paralelno pretrazuje sve storee i vraca rankirane rezultate.

CLI:

```
node retrieval-orchestrator.js query "tema" [--limit N] [--verbose] [--stores s1,s2]
node retrieval-orchestrator.js stats
node retrieval-orchestrator.js stores
```

Module:

```
const { RetrievalOrchestrator } = require('./retrieval-orchestrator');
const ro = new RetrievalOrchestrator();
```

```
const { results, meta } = await ro.query('tema', { limit: 5 });
```

Stores:

Store	Tip pretrage	Entries	Izvor
`hivemind`	Cosine similarity + LIKE fallback	13,473	hivemind.db
`knowledge`	Cosine similarity (vector-db.js)	24,636	knowledge.db
`rag`	Cosine similarity na RAG cache	2,201	flywheel.db
`sessions`	Grep text search	761 fajlova	~/system/memory/sessions/

3.3 Vector Database

File: `~/system/tools/vector-db.js` **Tip:** SQLite + Float32Array BLOB kolone (custom implementacija) **Embedding model:** `nomic-embed-text` (768-dim, lokalni, via Ollama) **Nema:** ChromaDB, FAISS, Pinecone, Weaviate, pgvector — sve je custom SQLite

UNIFIED EMBEDDING (2026-02-23): Svi toolsi koriste ISTI model (`nomic-embed-text` via Ollama):

- `vector-db.js` — JS modul (originalni)
- `memory-indexer.py` — Python indexer (prepisani sa sentence-transformers)
- `hivemind.js` — HiveMind embeddings (novo)
- `session-archiver.js` — Session embeddings (novo)
- `rag-router.js` — RAG cache embeddings (originalni)

Prethodno: `memory-indexer.py` je koristio `all-MiniLM-L6-v2` (384-dim) — razliciti vektorski prostori, cosine similarity izmedju njih je besmislen. Fiksirano u MC #1804.

Mogucnosti:

- Semanticki search (cosine similarity)
- Hybrid search (SQL WHERE + vektor ranking)
- Kolekcije sa metadata kolonama
- Bulk insert sa batching-om (32 docs/batch)

3.4 Knowledge Base (Document Store)

File: `~/system/tools/knowledge-base.js` **DB:** `~/system/databases/knowledge.db`

Velicina (2026-02-23): 24,636 entries (13,558 dokumenata + 11,075 memory-file chunks + 3 session chunks)

Schema:

- `kb_docs` — metadata (source, title, tag, hash, chunk count)
- `documents` — vektor-indeksirani chunkovi (content, embedding BLOB, tag)

Tagovi:

Tag kategorija	Primjer tagova	Entries
`memory-file`	Svi ~/system/ MD fajlovi	11,075
Projekti	lumiscare, drop, drop-architecture	~8,000
Patterns	pattern-security, pattern-architecture	~500
System	agents, system, rules, organization	~900
Sessions	session	3+ (raste)

Dva indexera:

- `knowledge-base.js` — URL/file ingestion sa auto-chunking, tagging, dedup
- `memory-indexer.py` — ~/system/ MD file scanner, batch embedding, `tag='memory-file'`

3.5 RAG Flywheel (Cache + Ucenje)

File: `~/system/tools/rag-router.js` **DB:** `~/system/databases/flywheel.db` **MCP Server:** `~/system/tools/rag-mcp.js` -> registrovan u `~/.claude/mcp.json`

Flywheel metrike (live, 2026-02-23):

Metrika	Vrijednost
Total queries	886
Cache hit rate	61.1%
Local model rate	4.4%
External rate	34.5%
Cache size	2,201 entries
Cost saved queries	580

MCP Tools (dostupni iz Claude Code sesije):

- `mcp__rag__rag_query(query, task_type)` — Rutiraj upit kroz cache -> local -> external
- `mcp__rag__rag_learn(question, answer)` — Dodaj Q&A u cache
- `mcp__rag__rag_stats()` — Flywheel metrike

RAG Router flow (Progressive Enrichment):

1. **Cache search** — cosine similarity na rag_cache (threshold 0.75)
2. **Local RAW** — Ollama bez KB konteksta, confidence gate (0.75+)
3. **Local ENRICHED** — Ollama SA knowledge.db kontekstom
4. **External** — Flag za Claude Code

DB Schema (flywheel.db):

- `interactions` — svaki query logiran (model, routing, cost, latency)
- `rag_cache` — Q&A parovi sa embedding-om (query_embedding BLOB, response, hit_count, project_tag)
- `shadow_log` — routing odluke + top 3 similarity scores

3.6 Session Archiver (NOVO, 2026-02-23)

File: `~/system/tools/session-archiver.js` **LaunchAgent:** `com.john.session-archiver` (daily 03:00)

Upravlja lifecycleom session fajlova — cijenimo summary, cistimo raw transkripte.

Komande:

```
node session-archiver.js stats                    # Statistika
node session-archiver.js archive [--dry-run]      # Strip raw transkripata >14 dana
node session-archiver.js index [--limit N]        # Embeduj summarije u knowledge.db
node session-archiver.js cleanup [--dry-run]      # Archive + index (cron)
```

Stats (2026-02-23):

- 761 session fajlova, 688 sa raw transkriptom
- 21.5 MB total, 20 MB (93%) je raw transcript bulk
- ~20 MB estimated savings od archivinga

---

4. HiveMind (Shared Memory Bus + Semantic Search)

File: `~/system/agents/hivemind/hivemind.js` **DB:** `~/system/agents/hivemind/hivemind.db` **Tip:** SQLite — keyword search + **semantic vector search** (od 2026-02-23)

Live stats (2026-02-23):

Metrika	Vrijednost
Total intel entries	13,473

Metrika	Vrijednost
With embeddings	~13,473 (backfill u toku)
Memos	70+
Retencija	90 dana

Upgrade (MC #1804): HiveMind je dobio vektor search:

- `embedding BLOB` kolona dodana u `intel` tabelu
- Svaki novi `post` automatski embeduje poruku (best-effort, skip ako Ollama down)
- Tri nova search moda:

Komanda	Tip	Opis
`query "X"`	LIKE	Keyword match (originalni, backward compat)
`semantic-query "X"`	Cosine	Embedding similarity search (top 5000 recent)
`hybrid-query "X"`	LIKE + Cosine RRF	Reciprocal Rank Fusion merge
`backfill-embeddings`	Batch	Embeduje entries bez vektora (32/batch)

Schema:

- `intel` — agent poruke (agent, type, message, data, priority, **embedding BLOB**)
- `agents` — registrovani agenti (name, role, status)
- `subscriptions` — agent topic pretplate
- `memos` — key-value memorija (key, value, access_count)

Intel tipovi: discovery, alert, opportunity, update, request, response, learning, error

Retencija: 90 dana za intel, 7 dana za event fajlove

---

5. Claude API (Anthropic)

Primarni AI: Claude Code (Opus za sesije, Sonnet/Haiku za agente)

Direktna API integracija:

- `~/system/tools/comms-agent/claude-handler.ts` - Anthropic SDK wrapper za automatske odgovore
- `~/system/tools/comms-responder.js` - Komunikacijski agent

Nema OpenAI API integracija u sistemu.

6. MCP Serveri

Server	File	Namjena
`rag`	`~/system/tools/rag-mcp.js`	RAG query/learn/stats
`email`	`~/system/tools/email-mcp-bridge.js`	Email operacije (2 accounta)
`youtube-transcript`	`@fabriqa.ai/youtube-transcript-mcp`	YouTube transkripti
`playwright`	`@playwright/mcp`	Browser automatizacija
`figma`	`@anthropic-ai/figma-mcp`	Figma dizajn pristup

7. Fine-tuned modeli (ALAI ML)

Tri custom modela trenirani na internim podacima:

Model	Baza	Namjena	Velicina
`alaiml-task-v1`	llama3.1:8b (Modelfile)	MC task klasifikacija i handling	986 MB
`alaiml-tender-v1`	llama3.1:8b (Modelfile)	Tender analiza i filtriranje	986 MB
`alaiml-email-v1`	llama3.1:8b (Modelfile)	Email klasifikacija i triage	986 MB

Retrain daemon: `com.john.alaiml-retrain` (LaunchAgent)

8. AutoCoder (Python Agent Framework)

Path: `~/system/services/autocoder/` **Komponente:**

- `agent.py` - Glavni agent logic
- `agent_classifier.py` - Task klasifikacija
- `parallel_orchestrator.py` - Multi-agent orkestracija (53 KB)
- `mcp_server/` - MCP server

UI: LaunchAgent `com.john.autocoder-ui` (port 8888) **Status:** Instaliran, koristi se opcionalno kroz build mode.

9. Baze podataka (sve SQLite)

Baza	Velicina	Namjena	Ima vektore?	Entries
`knowledge.db`	~50 MB	Document store (KB + memory-file + sessions)	DA (BLOB 768-dim)	24,636
`flywheel.db`	~10 MB	RAG cache + interaction log + routing	DA (BLOB 768-dim)	2,201 cache + 886 interactions
`hivemind.db`	~30 MB	Agent memory bus + memos + semantic search	DA (BLOB 768-dim)	13,473
`mission-control.db`	~3 MB	Task management	NE	1,804+ tasks
`events.db`	~3 MB	Event bus	NE	—
`contacts.db`	~50 KB	Kontakti	NE	—
`invoices.db`	~40 KB	Fakture	NE	—

Unified embedding model (od 2026-02-23): Sve 3 vektor-baze koriste ISTI model (nomic-embed-text 768-dim via Ollama). Nema mismatch-a.

Nema eksternih vektor baza (ChromaDB, FAISS, Pinecone, Weaviate, Qdrant, pgvector).

---

10. Sto POSTOJI vs Sto NE POSTOJI

Postoji (verifikovano 2026-02-23)

- 7 lokalnih Ollama modela (ukljucujuci 3 fine-tuned)
- Unified embedding model (nomic-embed-text, 768-dim, lokalni) — ISTI za sve storee
- Custom vektor DB (SQLite + BLOB, cosine similarity)
- **Retrieval Orchestrator** — 4-store parallel search sa RRF merge (NOVO)
- RAG 3-tier routing sa flywheel cache-om (61.1% hit rate, 886 queries)
- Knowledge base: 24,636 entries (documents + memory files + sessions)
- **HiveMind semantic search** — cosine + hybrid + backfill (NOVO)
- **Session archiver** — cleanup + embedding + daily cron (NOVO)
- Tier router za task->model dispatch (6 tierova)
- 5 MCP servera (RAG, email, YouTube, Playwright, Figma)
- 3 ALAI fine-tuned modela
- Usage tracking za sve AI pozive

- Claude API integracija (comms-agent)

NE postoji

- Nema cloud vektor baza (ChromaDB, Pinecone, Weaviate...)
- Nema OpenAI API
- Nema LangChain / LlamaIndex / LanceDB (custom implementacija, zero external deps)
- Nema cloud embeddings (sve lokalno)
- Nema GraphRAG (prevelik effort za nas obim)
- Nema cross-encoder reranking (Ollama default dovoljan)
- llama3.1:70b i qwen2.5:72b konfigurirani ali NE instalirani
- BookStack NIJE dio RAG pipeline-a (samo human-readable wiki)

---

11. Arhitekturni princip

Cost-optimized hybrid: Cache prvo -> Lokalni modeli drugo -> Cloud API zadnji.

- Svi embeddings su lokalni (Ollama nomic-embed-text, 768-dim)
- Sav vektor storage je u SQLite BLOB kolonama (Float32Array)
- **Jedan embedding model** za cijeli sistem — nema mismatch-a
- Nema cloud zavisnosti za RAG
- Claude API se koristi samo za ono sto lokalni modeli ne mogu
- Fine-tuned modeli pokrivaju repetitivne domenske taskove (email, tender, MC tasks)
- **Retrieval orchestrator** objedinjuje sve storee u jedan poziv sa RRF merge

Tool-First Protocol (retrieval redoslijed)

```
BookStack (human wiki) -> RAG MCP (mcp__rag__rag_query) -> Manifest
-> HiveMind (semantic-query) -> Internet -> Azuriraj bazu
```

Za programski retrieval: `node retrieval-orchestrator.js query "tema"` — automatski paralelno pretrazuje sve.

---

12. Changelog

Datum	Promjena	MC Task
2026-02-23	RAG System Upgrade: unified embedding, HiveMind vector search, retrieval orchestrator, session archiver	#1804

Datum	Promjena	MC Task
2026-02-21	Initial document created — full system inventory	—

Revision #7
Created 2026-02-21 17:57:29 UTC by John
Updated 2026-07-26 20:02:13 UTC by John