ALAI AI System Architecture

Arhitektonska revizija ALAI AI sistema: kako je zamišljen, šta radi, virtuelne kompanije, event-sourcing, i gdje je smisao. Panel 2026-06-22 (Petter Graff lead). MC #104188.

00 — Verifikovani System Map (2026-06-22)

ALAI AI System — Verified System Map (input artifact)

Date: 2026-06-22 · **Compiled by:** John (tool-verified, not from memory) · **For:** Architect panel (Petter Graff lead)

This is the ground-truth snapshot the architect panel must reason over. Every line below was verified live this session (curl health, process list, DB reads, Slack API, log inspection).

1. Platform services (live status verified 2026-06-22)

Service	Domain	Status	Role
BookStack	docs.alai.no	200	Wiki / knowledge base
Planka	boards.alai.no	200	Kanban boards
Paperless	archive.alai.no	302 (up)	Document archive (invoices, contracts, attachments)
Documenso	sign.alai.no	302 (up)	E-signing
Grafana	grafana.alai.no	302 (up)	Monitoring
Vaultwarden	vault.alai.no	200	Secrets
Mission Control	mc.alai.no	200	Tasks (source of truth)
LightRAG	lightrag.alai.no	root 000 / /health 200	Knowledge graph (auth-gated)
Ollama	ollama.alai.no	200	Local LLM

2. Engine tools (non-skill/hook/agent)

- `mc.js` — Mission Control task engine (mc.db) — system core
- `discover.js` — unified discovery (tools/skills/agents/BookStack/RAG)
- Email suite — `email-inbox`, `email-reactor`, `email-to-task`, `email-to-ticket`, `email-to-contact`, `email-audit`, `email-briefing`
- `slack.js` / `slack-bot.js` — 19 channels (verified live)
- `planka-sync.js` / `planka-admin.js` — MC↔Planka bridge
- `bookstack-sync.js` / `bookstack-webhook-relay.js` / `bookstack-staleness.js`
- HiveMind (`hivemind-*.js`) — internal event bus + knowledge
- LightRAG tools (`rag-*`, `lightrag-auth-helper`)
- `fiken-*` (accounting), `documenso-webhook`, `contacts.js`
- Tool Shed (`tool-shed.js`) — running pid 1319, port 3050, 307 tools registered
- Library v3 (`library.js`) — 79 global skills, 12 companies, last sync today
- Dashboards (ceo / mc / health / tender / expansion); `cost-tracker`, `agent-manager`

3. Data flow (verified wiring)

```
MIGADU (1 acct, all domains; IMAP boxes: john/info/alai/dev/alem)
  → email-inbox.js (email-inbox.db) → classify (Ollama)
     ACTION → email-to-task.js → MISSION CONTROL (mc.db)   [dedup by message-id]
     INFO/SPAM → log only
     attachments → email-attachment-fetcher → PAPERLESS (archive.alai.no)
MISSION CONTROL ↔ PLANKA  (planka-sync.js bidirectional: createCard/moveCard/syncStatus)
MC/agent events → HiveMind event-bus → LightRAG (knowledge graph) + BookStack (docs)
SLACK = outbound notification layer (ceo-daily-digest / escalation-notify / cron-notify)
Fiken / Documenso / Vaultwarden = accounting / e-sign / secrets (on-demand)
```

4. Virtual companies (persona-agent model)

ALAI routes work to ~12 "virtual companies", each a persona-agent cluster (specialist-mapping.json):

- CodeCraft (architecture/backend) · Vizu (frontend/design) · FlowForge (devops/infra) · Proveo (QA) · Securion (security) · AgentForge (AI/ML/RAG) · Finverge (fintech) · Skybound (mobile/SaaS) · Lexicon (legal/docs) · Helixsupport (incident) · Proxima (marketing) · Resolver (cross-company systemic)
- John = AI Director / orchestrator (does NOT build); dispatches via MEHANIK gate → specialist → Proveo verify → MC done.

5. Event-sourcing surfaces (verified existing)

- `evidence-ledger` (SQLite, append; had a dup-insert bloat incident #102796 → dedup index)
- MC task history (`mc_history` — 542 rows this session window)
- HiveMind events (`hivemind.db`)
- Session checkpoints (auto-generated session summaries)
- NOTE: these are append-ish logs, NOT a formal event-sourced architecture (no single event store as source of truth, no projections/replay model). This is a key question for the panel.

6. Verified anomalies / smells (honest)

1. BookStack → Slack relay is DISABLED (`bookstack-webhook-relay.js`: "no Slack call made" — audit-log only).
2. Meta-agent cron daemon runs on schedule but produces 0 tasks / 0 action-items lately (idle loop).
3. Meta-skills pipeline (skill-creator/registry) exists but no new skills created since Jun 15.
4. HexaDB project just frozen (NO-GO) — a case of a bee/6×19 metaphor over-engineered into a DB with no real geo capability. Cautionary precedent: is more of the system metaphor-driven rather than need-driven?
5. mc.js `done` is gated by a per-session verdict file; research/advisory tasks need `--force` (friction).

01 — Ima li ovo smisla? (Executive Answer)

Ima li ovo smisla?

Da, ali ne onako kako je mjereno.

Sistem ima zdravu, zreliju jezgru nego što izgleda — `mc.js` je stvarni, dosljedni source-of-truth kroz koji prolaze svi tokovi; event-bus i transactional outbox su svjesno i tačno riješeni (to većina seniorskih timova zezne); a 12 "kompanija" je zapravo rutiranje koje radi. To je realan leverage koji je solo-CEO-u isporučio **Bilko, LumisCare, SnowIT, Phase 2** (9 PR-ova merged), migracije. To NIJE iluzija.

Problem nije da je sistem lažan — problem je da je **prestao mjeriti vrijednost i počeo mjeriti aktivnost.**

- 15.661 task · $11.735 Opus burn za 60 orkestracijskih poziva · 75–80% rada opslužuje sam sistem
- 23 "SEAL" taska čiji jedini deliverable je rečenica da sistem radi
- 27% backloga zaglavljeno (2.156 blocked + 2.018 paused)

Presuda u jednoj rečenici: motor je dobar i vrijedan, ali je narastao prsten tkiva oko motora koji troši gorivo da bi dokazao da motor radi. Ni odbaciti, ni tješiti — **odsjeći prsten, ostaviti motor.**

> Verdikt panela: Petter Graff (lead), Martin Kleppmann (event-sourcing), sentinel-architect (struktura), sentinel-BA (poslovni smisao). MC #104188.

02 — Kako je zamišljen vs šta stvarno radi

Vizija vs stvarnost

Vizija

"Build systems that build systems." John orkestrira → 12 virtuelnih kompanija grade → MEHANIK gate-uje → Proveo verifikuje → sve teče kroz jedan SSoT → znanje se akumulira u graf → sistem uči da poboljša sebe.

Stvarnost (verifikovano)

Tvrdnja vizije	Stvarnost
John orkestrira	✅ **Tačno** — mc.js je stvarni hub, svi flow-ovi kroz njega
Kompanije grade	⚠️ **Tačno, ali asimetrično** — ~25% rada revenue-bearing, ~75% sistem popravlja/dokazuje sebe
Sistem uči da poboljša sebe	❌ **Pukla** — meta-agent emituje `--route knowledge`, a to NIJE validna ruta (mc.js:2621) → svaki insert tiho pada → "0 taskova" je silent failure, ne miran queue

Najambiciozniji dio vizije (samo-učenje) je mrtav i niko nije primijetio jer je smrt tiha. 301 "APPLY-KNOWLEDGE" sesija + meta-agent + meta-skills su trebali biti petlja samopoboljšanja; meta-agent (kernel/meta-agent.js:156,179) baca grešku u tišini.

Razlika vizija↔stvarnost nije laž — to je **drift bez reconciliation-a.** Isti obrazac koji se vidi u podacima (10 izvora istine bez sloja koji ih miri) vidi se i u namjeri.

03 — Virtuelne kompanije: vrijednost vs teatar

Virtuelne kompanije

Vrijednost (zadržati)

Specijalizovano rutiranje je realno. "Backend Petteru, security Parisi, QA Angie" daje bolji prompt-context, bolju personu, bolji output. Kao dvanaest dobro naštelovanih system-promptova s jasnim granicama. **Vrijedi, ostaje.**

Teatar (smanjiti)

Ceremonija oko rutiranja: MEHANIK gate → dispatch → P2P peer → Proveo → done verdict file → postflight → memory writeback → BookStack → mesh receipt — **za firmu od jednog čovjeka bez kupca koji plaća za ceremoniju.** Svaki korak rođen iz stvarnog incidenta (pošteno), ali zbroj je proces-overhead koji bi imao smisla na 200-ljudi org sa compliance obavezama.

Koliko kompanija stvarno treba?

Aktivna isporuka ide kroz **~4–5**: CodeCraft (backend/arch), Vizu (frontend), FlowForge (devops), Proveo (QA), Securion (security).

Ostalih 7 (Finverge, Skybound, Lexicon, Helixsupport, Proxima, Resolver, AgentForge) drži kao **lazy-load persone — definicije, ne stalno-aktivni entiteti s ceremonijom.**

> Persona košta nula dok je ne pozoveš. Ceremonija košta tokene svaki put. **Reži ceremoniju, ne persone.**

04 — Event-sourcing: šta jeste, šta nije

Event-sourcing — razbijanje zablude

Šta JESTE (i zrelo je)

Pravi **event-driven messaging**: `events.db` sa correlation_id / causation_id / idempotency_key UNIQUE / retry state-machine, i **pravi transactional outbox** (outboxWrite unutar MC transakcije + relayOutbox). Svjesno rješenje dual-write problema koje 90% timova ne uradi kako treba. **Pohvala stoji.**

Šta NIJE

Ovo **NIJE event sourcing**:

- Source-of-truth je mutable `tasks` CRUD tabela
- `replay()` re-isporučuje side-efekte, NE rekonstruiše stanje iz događaja
- `task_history` (67k redova) je audit changelog, ne event log
- `causation_id` je **0% popunjen** — uzročni lanac postoji kao kolona ali je prazan
- 4 "event surface" (evidence_ledger / mc_history / hivemind / checkpoints) su nezavisni audit logovi VAN busa

Zabluda koju treba razbiti

"Trebamo li preći na pravi event-sourcing?" → NE. Task je obična state-mašina (open→started→ready→done). Full event-sourcing nad tim = čisti over-engineering — isti metafora-driven nagon koji je proizveo **HexaDB** (bee/6×19 metafora ugurana u DB bez geo-potrebe). Ne pravite drugi HexaDB od event-store-a.

Gdje JEDINO vrijedi formalizovati

Orkestracijske **odluke** — zašto je agent rutirao ovamo, šta je MEHANIK presudio, koji model, koji verdikt. Te odluke provući kroz **postojeći** bus sa popunjenim correlation_id + causation_id → deterministički replay "zašto je sistem ovo odlučio". Zlato za debug halucinacija i audit. **Ne diraj task-state, ulanči odluke.**

Najve?i data-rizik

Nekontrolisani dual-write na ~10 izvora istine bez reconciliation sloja. Konkretno **fire-and-forget MC→Planka** (greška se tiho guta) + ne-idempotentni append logovi. Lijek: provući SVE side-efekte (uklj. Planka) kroz postojeći outbox + periodični reconciliation job. **Imate napola — dovršite, ne gradite novo.**

05 — Tri arhitektonske odluke (ADR)

Tri preporu?ene odluke (prioritizovano)

ADR-1 — Event-store konsolidacija + HiveMind disk-bomba fix

Napor: M · Revenue: NE (ali sprečava outage koji blokira sve) · PRIORITET: URGENTNO

Jedina stvar s rokom diktiranim fizikom. Live masa = `~/Public/alai-system/rag/hivemind.db` = **195M (modifikovan danas 11:19), van backup-rotacije**, dok kanonski `system/databases/hivemind.db` = 0B + 4 stale symlinka pokazuju na prazan fajl. Topologija = ista bomba kao incident #102796 (disk-full koji je blokirao CEO login).

Odluka: (a) konsolidovati 8 event/knowledge baza na jedan kanonski put; (b) ukinuti 11 praznih (0B) baza + symlink-zbrku; (c) `wal_autocheckpoint` + WAL-nuke cron; (d) pravu lokaciju pod backup-rotaciju. **Odbrana od ponovljenog outage-a, ne poboljšanje.**

ADR-2 — Meta-agent route-enum fix ILI gašenje + Revenue Gate

Napor: S · Revenue: DA (mjerilo vrijednosti)

Meta-agent šalje `--route knowledge`, enum odbija, insert tiho pada → mrtav daemon koji se pretvara da je živ. **Odluka (biraj svjesno):** (a) dodaj `knowledge` u validRoutes i pusti petlju da radi, ILI (b) ugasi je potpuno. NE ostavljaj zombi.

Plus: uvedi **Revenue Gate metrik** — svaki self-serving infra-task mora referencirati klijent-isporuku koju omogućava, ili ide u "deferred" bucket. Zamrzni nove čisto-interne taskove dok omjer 25/75 ne krene. **Jedina stvar koja direktno napada "mjerimo aktivnost ne vrijednost".**

ADR-3 — Gate scar-tissue konsolidacija

Napor: M · Revenue: NE (smanjuje token-burn + kognitivni teret) · POSLIJE ADR-1/2

13+ distinct gate/verifier artefakata (alai-claim-gate, claim-verifier, mini-verifier, goal-verifier, evidence-ledger-gate, deploy-gate, gate-pre-claim, gate-pre-deploy, verifier-claim-fill…). Isti koncept "dokaži prije nego tvrdiš" implementiran 6 puta, svaki rođen iz incidenta, nijedan uklonjen — **arheologija straha.**

Odluka: jedan `verify(claim, evidence)` kontrakt s pluggable provjerama; ugasi duplikate iza njega. NE žuri — gate-ovi rade, samo ih je previše.

06 — Gdje je smisao (završni stav)

Gdje je smisao

Smisao postoji, i precizno se zna gdje: u onih **~25% rada koji je solo-CEO-u dao da isporuči ono što čovjek sam ne bi stigao.** Bilko, LumisCare, SnowIT, Phase 2 — čovjek koji nosi sve, koristeći sistem kao protezu kapaciteta. Legitiman, čak elegantan razlog da sistem postoji. Tu je smisao gust i stvaran.

Smisao NEMA — i tu budi nemilosrdno iskren — u onih **75% gdje je sistem počeo postojati radi sebe:** 23 SEAL-taska čiji deliverable je rečenica "sistem radi", mrtav meta-agent koji se pretvara da uči, šest verifikatora koji verifikuju da verifikacija radi, metafore (HexaDB) koje postaju kod bez potrebe. Nije zlonamjerno — to je **entropija ambicioznog sistema bez kupca koji plaća za njegovu ambiciju.**

Sistem je počeo graditi sisteme koji grade sisteme, i negdje na trećem nivou izgubio iz vida da na dnu lanca mora stajati neko ko plati.

Brutalna istina

Prihod = 0 nije problem sistema — to je presuda nad time gdje sistem troši energiju.

Motor koji vozi Bilko/LumisCare/SnowIT vrijedi svaki token. Prsten koji vozi sam sebe — ne.

Smisao se ne nalazi u tome da sistem bude veći ili pametniji o sebi. Smisao je u jednoj liniji:

> **Da li je task na drugom kraju lanca neko ko bi platio za rezultat.** Ako jeste — gradi. Ako je task da sistem dokaže da radi — to je trošak, ne smisao, i pripada deferred bucketu.

Sistem je dobar. Previše dobar prema sebi. Reži prsten, hrani motor, mjeri prihod ne aktivnost. Tu je smisao — i dovoljan je da se nastavi, ako se vrati sebi na zadatak.

ALAI AI System Architecture · Arhitektonski panel 2026-06-22 · MC #104188 · Petter Graff (lead), Kleppmann, sentinel-architect, sentinel-BA.

07 — Live-fire test (pravi kupac, 2026-06-22)

Live-fire test — pravi kupac kroz sistem (2026-06-22)

Knjiga je do ove tačke bila analiza. Ovo poglavlje je **empirijski dokaz** iz stvarnog kupca: CEO je dao realan lead — *Ćevabdžinica Specijal, Sarajevo, web prezentacija* — i pratili smo šta sistem STVARNO uradi.

Šta je sistem trebao pokrenuti

Sistem **ima** kompletan vođeni onboarding (`onboard-client` skill + `onboard-client.js`), 7 faza, svaka s gateom: First Contact → Discovery → **NDA** → Proposal → **Contract** → Project Setup → development.

Šta se STVARNO desilo (tool-verifikovano)

Faza	Ko	Stvarno
Intake	John ručno	bez onboard-client skilla
Security check	**NIKO**	Securion nije pozvan
Kontrakt	**NIKO**	rad počeo PRIJE ikakvog ugovora
Legal	**NIKO**	Lexicon nije pozvan
Planka	**0/6 sync**	verifikovano: Specijal taskovi nisu na boardu
Dizajn izbor	Vizu sam	CEO-u nije ponuđeno opcija za izbor
Gates (MEHANIK/prompt-forge/P2P/Proveo)	**preskočeno**	verifikaciju radio John očima

~90% uradio John ad-hoc. Jedina firma koja je stvarno radila = Vizu (dizajn/build, kvalitetno).

Presuda

Pipeline postoji, ali se ne pokreće sam — zavisi od toga da ga orkestrator ručno pozove. I nije ga pozvao.

Sposobnost je tu (skill, gateovi, firme); disciplina/ožičenje da se automatski okine — nije. To je doslovno "*sistem koji gradi dug, ne sisteme*": imamo gotov onboarding koji ne koristimo. Live-fire test potvrđuje teorijski verdikt iz poglavlja 01–06.

Šta je iz ovoga proizašlo

1. **Gate-enforced onboarding** — novi kupac okida `onboard-client` gate koji BLOKIRA (kao MEHANIK/PI2), ne oslanja se na pamćenje orkestratora. (MC fix-task otvoren.)
2. **Bonus nalaz:** email ingest pao (zadnji mail Jun 19, monitor exit=1) — MC #104211.

Pobo?na lekcija (anti-halucinacija u deliverable-u)

Prvi pokušaj sajta je sadržavao izmišljeno "od 1971" + kebab-stock fotke za ćevapdžinicu (CEO: "SCAM, FUJ"). Drugi pokušaj (Vizu, s tvrdim brifom): autentični sarajevski ćevapi (Wikimedia, lokalno), nula fabrikacije, premium izgled — John verifikovao screenshote očima. Pouka: nula izmišljanja u customer-facing artefaktu; ćevapi ≠ kebab.

Live-fire test 2026-06-22 · MC #104188 · podaci tool-verifikovani te sesije.