

Connectivity Diagram

2.1 — AI Factory Connectivity Diagram

Date: 2026-05-09 **Auditor:** sentinel-architect **Phase:** 2 — Synthesis from P1 inventory reports 1.1, 1.2, 1.3, 1.4 and P2 reports 2.2, 2.3 **Mode:** READ-ONLY. No mutations.

Section A — Control Plane Diagram

The diagram below shows the advertised flow from CEO input to task closure. Solid arrows are flows that actually work. Dotted red arrows are advertised edges that are broken or absent. Labels show the transport mechanism.

flowchart TD

CEO([CEO / Alem])

JOHN([John – Orchestrator\nClaude Code CLI session])

MH["/mehanik gate\n~/ .claude/agents/mehanik.md\n113 cleared tokens in /tmp"]

PF["/prompt-forge\n~/ .claude/skills/prompt-forge/"]

PIO["pi-orchestrator\n~/system/kernel/pi-orchestrator.js\nPID 75750 – MOCK MODE"]

SPEC["Specialist Agent\ne.g. petter-graff, angie-jones\n~/ .claude/agents/*.md"]

TOOL["Tool\n~/system/tools/ (250 live)"]

ART["Artifact\n(code / doc / spec / evidence file)"]

VERIFIER["Verifier / verify-fix-

loop\n~/ .claude/agents/verifier.md\n~/ .claude/skills/verify-fix-loop/"]

TPF["/task-postflight\n~/ .claude/skills/task-postflight/"]

MCD["mc.js done\n~/system/tools/mc.js"]

PROVEO["Proveo / Angie Jones\n~/ .claude/agents/angie-jones.md"]

HOOK["Hook Layer\n~/ .claude/hooks/ (12 active)"]

CEO -- "CLI conversation" --> JOHN

JOHN -- "CLI / Task dispatch" --> MH

MH -- "cleared token written to /tmp/mehanik-cleared-N\nBlueprint read enforced (PARTIALLY

```

- WARN scores pass)" --> PF
  PF -- "forged prompt → Task dispatch" --> PIO

  PIO -. "Task dispatch – mc.js write\nBROKEN: MOCK MODE\nalai-config-mock.json
loaded\nPlanka localhost:3100 not listening\n'No eligible tasks' every 30s" .-> SPEC

  SPEC -- "Tool calls\n(Read / Edit / Bash / Grep)" --> TOOL
  TOOL -- "Write / Edit" --> ART
  ART -- "mc.js ready write" --> HOOK
  HOOK -- "PreToolUse / PostToolUse\nexits 0 = pass, exits 2 = block" --> MCD

  MCD -. "ADVERTISED: auto-invokes verifier\nACTUAL: ABSENT\n0 hooks, daemons, or pi-orch
code\ncalls verify-fix-loop\n(source: 2.2)" .-> VERIFIER

  VERIFIER -. "ADVERTISED: auto-loop to fix-builder\nACTUAL: manual invocation only\nno
programmatic trigger" .-> SPEC

  MCD -- "mc.js ready → /task-postflight\n(manual invocation only for H tasks)" --> TPF
  TPF -- "Task dispatch – CLI" --> PROVEO
  PROVEO -- "AC checklist → verdict" --> TPF
  TPF -- "mc.js done (with evidence)" --> MCD

  MCD -. "ADVERTISED: pi-orchestrator consumes\n'done' events for next task\nACTUAL: MOCK
MODE – consuming nothing" .-> PIO

  style PIO fill:#ffcccc,stroke:#cc0000
  style VERIFIER fill:#ffcccc,stroke:#cc0000
  style MH fill:#ffffcc,stroke:#cccc00

```

Annotation notes:

- CEO → John: works. Standard CLI session.
- John → Mehanik: works. 113 cleared tokens confirm Mehanik runs regularly.
- Mehanik → prompt-forge → pi-orchestrator: the dispatch chain exists structurally. pi-orchestrator is alive (PID 75750) but in MOCK MODE — it reads mock config and never consumes real MC tasks.
- pi-orchestrator → Specialist: BROKEN because mock mode means pi-orchestrator never fires a Task dispatch to a real specialist.
- Specialist → Tool → Artifact: works when agents are dispatched by John manually (not via pi-orchestrator).
- Artifact → mc.js done (via hooks): works. The hook layer (12 active hooks) enforces gates on mc.js writes.

- mc.js done → verifier: ABSENT. No automated trigger. CEO is the de-facto verifier (source: 2.2).
- mc.js done → pi-orchestrator: BROKEN. Mock mode means pi-orchestrator does not react to task completions.

Section B — Data Plane Diagram

Shows all memory stores with their actual write paths (solid = live, dotted red = dead, dotted orange = partial/degraded).

flowchart LR

```
CC["Claude Code\n(built-in auto-memory)"]
```

```
MDFILES[".md auto-memory files\n~/\.claude/projects/-Users-makinja/memory/\n123 files - LIVE"]
```

```
H00K_LR["lightrag-auto-ingest.sh\nPostToolUse hook\nfires on Write/Edit to in-scope paths"]
```

```
LR["LightRAG\nlocalhost:9621\n999 docs indexed\npipeline_busy=true\nHEALTHY but DEGRADED"]
```

```
DISCOVER["discover.js\nhttps://lightrag.alai.no/query\n(external hostname – Caddy proxy)"]
```

```
IQ["ingest-queue.sqlite\n~/system/state/\n946 items FROZEN"]
```

```
RDW["rag-drain-worker\nPID 3640\nETIMEDOUT on Vaultwarden"]
```

```
RBA["rag-bookstack-adapter\nevery 5min – exit 256\nblocked by backpressure"]
```

```
RMCA["rag-mc-adapter\nevery 5min – exit 256\nblocked by backpressure"]
```

```
RFSEA["rag-fsevents-adapter\nWatchPaths – exit 1\nblocked by backpressure"]
```

```
BKS["BookStack\ndocs.alai.no"]
```

```
MCL0G["mc-task-outcomes.json\n~/system/logs/"]
```

```
MEM0["mem0 API\nlocalhost:9000\nHEALTHY – 0 active writers"]
```

```
QDR["Qdrant\nlocalhost:6333\n5 collections\n93,510 total vectors"]
```

```
MEM0J["mem0_john collection\n865 vectors – STALE"]
```

```
KNOW["knowledge collection\n31,274 vectors – STALE\nunknown origin"]
```

```
SESS["sessions collection\n929 vectors – unknown writer"]
```

```
HIVE_Q["hivemind collection\n60,442 vectors – LIVE"]
```

```
HIVEJS["hivemind.js CLI\ndual-write on post"]
```

```
HIVEDB["HiveDB SQLite\nhivemind.db\n17,551 intel rows – LIVE"]
```

```
CHROMA["Chroma\n~/\.claude-mem/chroma/\n6,584 embeddings\nno active writer or reader"]
```

```
FLYWHEEL["flywheel.db SQLite\n~/system/databases/\nLIVE – rag-router.js cache"]
```

```
RAG_ROUTER["rag-router.js\ncache → Ollama → external"]
```

```
CC -- "native write" --> MDFILES
```

```
MDFILES -- "PostToolUse trigger" --> HOOK_LR
```

```
HOOK_LR -- "curl POST localhost:9621" --> LR
```

```
LR -- "serves queries" --> DISCOVER
```

```
BKS -- "poll every 5min" --> RBA
```

```
MCLOG -- "tail" --> RMCA
```

```
RBA -- "enqueue" --> IQ
```

```
RMCA -- "enqueue" --> IQ
```

```
RFSEA -- "enqueue" --> IQ
```

```
IQ -- "drain attempt" --> RDW
```

```
RDW -. "DEADLOCKED\nVaultwarden ETIMEDOUT\nCF Access creds missing\n946 items queued, 0  
drained" .-> LR
```

```
HIVEJS -- "write" --> HIVEDB
```

```
HIVEJS -- "dual-write best-effort" --> HIVE_Q
```

```
HIVE_Q --> QDR
```

```
MEM0 --> QDR
```

```
QDR --> MEM0J
```

```
QDR --> KNOW
```

```
QDR --> SESS
```

```
QDR --> HIVE_Q
```

```
CC -. "INTENDED: POST localhost:9000/add\nACTUAL: ABSENT\n0 callers in  
hooks/tools/daemons" .-> MEM0
```

```
DISCOVER -. "INTENDED: query mem0 for personal facts\nACTUAL: ABSENT\ndiscover.js does not  
call localhost:9000" .-> MEM0
```

```
CHROMA -. "writer UNKNOWN\nreader UNKNOWN\n6584 embeddings orphaned" .-> CHROMA
```

```
RAG_ROUTER -- "learn" --> FLYWHEEL
```

```
RAG_ROUTER -- "query cache-hit" --> FLYWHEEL
```

```
style RDW fill:#ffcccc,stroke:#cc0000
```

```
style IQ fill:#ffcccc,stroke:#cc0000
```

```
style MEM0 fill:#fff0cc,stroke:#cc8800
```

```
style MEM0J fill:#ffcccc,stroke:#cc0000
```

```
style KNOW fill:#ffcccc,stroke:#cc0000
```

```
style CHROMA fill:#ffc000,stroke:#cc0000
```

```
style SESS fill:#fff0cc,stroke:#cc8800
```

Key findings:

- The LightRAG local write path (Claude Code → .md → hook → LightRAG) works but the queue-drain path (746+ items from bookstack, MC logs, fsevents) is completely deadlocked because `rag-drain-worker` cannot authenticate through Cloudflare Access (Vaultwarden ETIMEDOUT).
- mem0 is a ghost: server alive, 93K+ vectors in Qdrant, zero active writers, zero active readers through the API.
- Chroma is a full orphan: 6,584 embeddings from an unknown writer, no identified reader.
- The Qdrant `hivemind` collection (60K+ vectors) is live because `hivemind.js` writes to it directly, bypassing the mem0 API entirely — this is the only healthy Qdrant write path.

Section C — Agent / Persona / Chain Plane

```
flowchart TD
```

```
SMJ["specialist-mapping.json\n~/system/agents/specialist-mapping.json\n29 mapped agents\n9 registered companies\nSOURCE OF TRUTH (incomplete)"]
```

```
CLAUDE_AGENTS["~/claude/agents/\n66 .md files\nRUNTIME STORE\n(what Claude Code can dispatch)"]
```

```
DEFINITIONS["~/system/agents/definitions/\nBACKUP STORE\n48 synced + 8 definitions-only"]
```

```
SYNC["~/bin/agent-definitions-sync.sh\nMANUAL – not scheduled"]
```

```
PERSONAS["~/system/agents/personas/\n12 persona dirs"]
```

```
P_REAL["8 Routable Companies\nAgentForge, CodeCraft, Finverge\nFlowForge, Proveo, Securion\nSkybound, Vizu\n(partial mapping only)"]
```

```
P_PHANTOM["4 Phantom Companies\nAxiom, Datavera, Resolver, Lexicon\nFull persona dirs, CLAUDE.md, agents/\n0 entries in specialist-mapping.json\nDispatch path = NONE via John routing"]
```

```
CHAINS["~/system/agents/chains/\n35 .yaml files\nNO chain runner exists\nall DEAD as executable automation"]
```

```
MAPPED_OK["24 mapped agents\nreachable on disk\nCAN be dispatched"]
```

```
MAPPED_MISSING["7 mapped agents\nIN specialist-mapping.json\nMISSING from
~/claude/agents/\ndispatches SILENTLY FAIL\n(dorota-huizinga, hadi-hariri\njames-bach, lee-
robinson\nlisa-crispin, minion\nanthropicchief-architect=fully phantom)"]
```

```
UNMAPPED_CRITICAL["Critical unmapped agents\nIN ~/claude/agents/\nNOT in specialist-
mapping.json:\n- validator (44 skill refs)\n- distiller (21 chain refs)\n- mehanik (7 skill
refs)\n- evidence-verifier\n- baseline-comparator\n- dzevad-jahic (Lexicon)\n- planner
(phantom – in chains only)"]
```

```
UNMAPPED_ORPHAN["11 Orphan agents\nno chain/skill/daemon refs:\n0.md, dr-sarah-chen,
Explore\nhelixsupport, indy-dandev\nmaria-santos, meta-agent\nPlan, rag-builder\nredzo-
reviewer, thaer-sabri"]
```

```
SMJ --> CLAUDE_AGENTS
```

```
SMJ -. "7 mapped agents\nnot on disk = UNREACHABLE" .-> MAPPED_MISSING
```

```
CLAUDE_AGENTS --> MAPPED_OK
```

```
CLAUDE_AGENTS --> UNMAPPED_CRITICAL
```

```
CLAUDE_AGENTS --> UNMAPPED_ORPHAN
```

```
DEFINITIONS -- "manual sync\n(agent-definitions-sync.sh)" --> CLAUDE_AGENTS
```

```
SYNC -. "not scheduled\ndrift pressure continuous" .-> DEFINITIONS
```

```
PERSONAS --> P_REAL
```

```
PERSONAS --> P_PHANTOM
```

```
P_PHANTOM -. "no routing entry\ndirect session name-drop only\nundocumented and
unreliable" .-> CLAUDE_AGENTS
```

```
P_REAL --> SMJ
```

```
CHAINS -. "NO EXECUTOR\n35 YAML files are docs only\nSkills call agents inline\nnot via
chain runner" .-> CLAUDE_AGENTS
```

```
style MAPPED_MISSING fill:#ffcccc,stroke:#cc0000
```

```
style P_PHANTOM fill:#fff0cc,stroke:#cc8800
```

```
style CHAINS fill:#ffcccc,stroke:#cc0000
```

```
style UNMAPPED_CRITICAL fill:#fff0cc,stroke:#cc8800
```

Key findings:

- specialist-mapping.json covers only 29 of 66 agents (44%). The two highest-usage agents system-wide — `validator` (44 skill file refs) and `distiller` (21 chain refs) — are completely absent from the routing table.
- 7 agents are mapped (John thinks he can dispatch them) but physically missing from `~/claude/agents/`. Any dispatch attempt silently fails.

- 35 chain YAML files have no executor. They exist as documentation only — skills invoke agents inline and ignore chain files entirely.
 - 4 phantom companies (Axiom, Datavera, Resolver, Lexicon) have full organizational infrastructure on disk but are completely invisible to John's routing system.
-

Section D — The True Picture (CEO-readable, 60 seconds)

Plan vs. Reality

The architecture diagram on paper shows: CEO gives task → John gates it through Mehanik → pi-orchestrator dispatches specialists → work gets done → verifier autonomously checks it → mc.js closes the loop.

The actual flow is: CEO gives task → John manually dispatches a specialist in the current conversation → specialist builds → John manually verifies (or CEO does) → John manually calls mc.js done.

Every automatic layer between "task received" and "task closed" is either in mock mode, deadlocked, or simply absent.

The 3 Fattest Dead Edges

Dead Edge 1 — pi-orchestrator in MOCK MODE. The orchestration kernel (PID 75750) is alive and cycling every 30 seconds. It reads `alai-config-mock.json`. Planka/MC API at localhost:3100 is not listening. The kernel prints "No eligible tasks" and does nothing. Every task that should flow automatically through the factory instead requires John to manually dispatch via conversation. This is the single edge whose repair would convert the factory from "manual assembly" to "automated pipeline."

Dead Edge 2 — RAG drain-worker deadlocked (946 items queued, 0 drained). Three adapters (BookStack, MC logs, filesystem events) successfully enqueue documents into `ingest-queue.sqlite`. The drain-worker (PID 3640) picks them up and tries to POST to LightRAG through Cloudflare Access — but Vaultwarden times out, so CF credentials cannot be fetched. The entire 946-item queue has been frozen. Meanwhile, the fsevents adapter is watching for filesystem changes and trying to enqueue `lightrag-monitor` health files — creating a feedback loop where the monitoring system feeds into the broken pipeline it is monitoring. One credential fix (valid `/tmp/bw-session` + reachable Vaultwarden) unblocks all three adapters simultaneously.

Dead Edge 3 — Verifier auto-invocation ABSENT. The `verify-fix-loop` skill and its `verifier` + `fix-builder` agents are fully specified and internally correct. There is zero wiring to any automated trigger. No hook, no daemon, no pi-orchestrator code calls them. When `mc.js ready` fires, no verification agent is invoked. CEO is the de-facto quality gate for the entire factory. One wiring point in `/task-postflight` SKILL.md (Section 2b) would give autonomous verification for non-high-stakes tasks immediately, without new infrastructure.

The 3 Highest-Leverage Wire Fixes

Fix 1 — Restore pi-orchestrator real config (L fix, maximum leverage). Determine why `alai-config-mock.json` loads instead of real config. If Planka is intentionally offline, restore it or point the orchestrator at the real MC API endpoint. This single fix converts the factory from "John as human dispatcher" to "automated task routing." Impact: every other automation layer (specialist dispatch, postflight, cost tracking) becomes meaningful instead of idle.

Fix 2 — Fix rag-drain-worker CF credentials (S fix, unblocks 946-item queue). Ensure Vaultwarden is reachable and `/tmp/bw-session` is valid for the service token that holds the LightRAG CF Access credentials. This is estimated as a 30-minute fix (refresh session token + verify vault connectivity). Impact: 946 queued RAG items drain, BookStack sync resumes, MC outcome logging resumes, the circular monitoring feedback loop breaks.

Fix 3 — Wire verify-fix-loop into /task-postflight (M fix, eliminates CEO-as-verifier bottleneck). Add a Section 2b to `~/claude/skills/task-postflight/SKILL.md`: after Proveo passes AC checklist, dispatch `/verify-fix-loop` for `docs / system / refactor / polish` domain tasks (MAX_LOOPS=3, \$5 cap already defined in the skill). This requires no new infrastructure — the skill conversation context already supports Task dispatch. Impact: CEO is removed from the quality loop for the majority of non-high-stakes tasks.

Section E — Edge Inventory Table

#	From	To	Transport	Status	Evidence	Fix Size
1	CEO	John (orchestrator)	CLI conversation	LIVE	Observed every session	—
2	John	/mehanik gate	Task dispatch / CLI	LIVE	113 cleared tokens in /tmp	—
3	/mehanik gate	Blueprint read	Read tool call	PARTIAL	CB#2 enforced; WARN scores (65/80) pass; missing-MC-ID bypasses gate entirely (2.3)	S

#	From	To	Transport	Status	Evidence	Fix Size
4	/mehanic gate	/prompt-forge	CLI / Task dispatch	LIVE	Observed in token chain	—
5	/prompt-forge	pi-orchestrator	mc.js write / Task	PARTIAL	pi-orch alive but MOCK MODE (1.4)	L
6	pi-orchestrator	Specialist agent	Task dispatch	DEAD	MOCK MODE — "No eligible tasks" every 30s; Planka localhost:3100 not listening (1.4)	L
7	John (manual)	Specialist agent	Task dispatch (CLI)	LIVE	Observed — this is the actual dispatch path	—
8	Specialist agent	Tools (Read/Edit/Bash)	Tool API calls	LIVE	250 live tools verified (1.2)	—
9	Tools	Artifact (file/code)	Write / Edit	LIVE	Standard Claude Code behavior	—
10	Artifact	mc.js ready	mc.js write + hook	LIVE	mc-ready-gate.sh fires; 12 active hooks (2.2)	—
11	mc.js ready	verifier / verify-fix-loop	(absent)	DEAD	0 hooks, 0 daemons, 0 pi-orch code calls verify-fix-loop (2.2)	M
12	mc.js ready	/task-postflight	Manual CLI invocation	PARTIAL	H-tasks only; manual trigger; no auto-invocation (2.2)	M
13	/task-postflight	Proveo / Angie Jones	Task dispatch	LIVE	Skill dispatches angie-jones.md; present on disk (2.2)	—
14	Proveo	mc.js done	mc.js write	LIVE	AC checklist → done path works	—

#	From	To	Transport	Status	Evidence	Fix Size
15	mc.js done	pi-orchestrator (next task)	mc.js event / API	DEAD	MOCK MODE — pi-orch does not react to done events (1.4)	L
16	Claude Code built-in	.md memory files	Native write	LIVE	123 files, auto-written by Claude Code (1.1)	—
17	.md memory files	lightrag-auto-ingest.sh	PostToolUse hook trigger	LIVE	Hook fires on Write/Edit to in-scope paths (1.1)	—
18	lightrag-auto-ingest.sh	LightRAG localhost:9621	curl POST	LIVE	999 docs indexed; pipeline_busy=true (1.1)	—
19	discover.js	LightRAG (external)	HTTPS GET to lightrag.alai.no	LIVE	External hostname via Caddy proxy (1.1)	—
20	rag-bookstack-adapter	ingest-queue.sqlite	SQLite write	DEAD	Exit 256 — backpressure gate (946 > 500) from frozen drain-worker (1.4)	S
21	rag-mc-adapter	ingest-queue.sqlite	SQLite write	DEAD	Exit 256 — same backpressure cascade (1.4)	S
22	rag-fsevents-adapter	ingest-queue.sqlite	SQLite write / WatchPaths	DEAD	Exit 1 — blocked by backpressure; also feeding monitoring artifacts into queue (1.4)	S
23	rag-drain-worker	LightRAG (via CF Access)	HTTPS POST (authenticated)	DEAD	Vaultwarden ETIMEDOUT — CF credentials unavailable; 946 items queued, 0 drained (1.4)	S

#	From	To	Transport	Status	Evidence	Fix Size
24	Any tool/hook/daemon	mem0 API localhost:9000	HTTP POST	DEAD	0 callers found in all of ~/system/tools, ~/.claude/hooks, ~/system/daemons (1.1)	M
25	discover.js	mem0 API	HTTP GET	DEAD	discover.js does not query localhost:9000 (1.1)	M
26	mem0 API	Qdrant mem0_john collection	gRPC / HTTP	PARTIAL	Server healthy; mem0_john has 865 stale vectors; no active writer to keep them fresh (1.1)	M
27	hivemind.js	HiveDB SQLite	SQLite write	LIVE	17,551 intel rows; write path active (1.1)	—
28	hivemind.js	Qdrant hivemind collection	HTTP (qdrant-client)	LIVE	60,442 vectors; dual-write best-effort (1.1)	—
29	Chroma store	Any consumer	(unknown)	DEAD	6,584 embeddings, no traced writer or reader (1.1)	M
30	agent-definitions-sync.sh	~/.claude/agents/	file copy	PARTIAL	48 files synced; 8 definitions-only agents unreachable at runtime; sync not scheduled (1.3)	S

#	From	To	Transport	Status	Evidence	Fix Size
31	specialist-mapping.json	Dispatch routing	JSON lookup	PARTIAL	29/66 agents mapped; validator (44 refs) and distiller (21 refs) absent; 7 mapped agents missing from disk (1.3)	M
32	35 chain YAML files	chain runner / executor	(absent)	DEAD	No chain runner exists; skills call agents inline; chains are documentation only (1.3)	L
33	John routing	Axiom/Datavera/Resolver/Lexicon	discover.js lookup	DEAD	4 companies absent from specialist-mapping.json; routing impossible via normal path (1.3)	M
34	pi-orch-health monitor	pi-orchestrator health signal	shell script	DEAD	pi-orch-health.sh deleted; last verdict 2026-05-06 CRITICAL; dark since (1.4)	S
35	cost-daily-report daemon	daily cost visibility	shell script	DEAD	cost-daily-report.sh deleted; cost reporting dark since 2026-04-29 — 10 days (1.4)	S
36	mc-ready-gate.sh	Blueprint score enforcement	blueprint-check.js	PARTIAL	Check runs; WARN scores (65, 80) allow dispatch; threshold 90 is advisory only (2.3)	S

#	From	To	Transport	Status	Evidence	Fix Size
37	Mehanik	Session binding validation	token mehanik_session_id	DEAD	All 113 inspected tokens show mehanik_session_id: unknown; cross-session reuse possible (2.3)	S
38	b2-offsite-backup	B2 cloud storage	B2 API	DEAD	403 storage_capacity exceeded; nightly snapshots not landing (1.4)	S
39	litestream	B2 replication stream	B2 API	PARTIAL	Litestream PID alive; separate nightly job fails; live replication status uncertain (1.4)	S
40	slack-bot	Slack WebSocket	Socket Mode	PARTIAL	PID 18046 alive; last crash exit 1; 300min silent at audit time; reconnects on timeout (1.4)	S

Status key:

- LIVE — flow confirmed working by tool-verified evidence
- DEAD — flow confirmed broken or absent by tool-verified evidence
- PARTIAL — flow structurally exists but has gaps, bypass paths, or degraded throughput

Fix size:

- S — Small: under 4 hours, single-file or credential change
- M — Medium: 1-2 days, new wiring or multi-file coordination
- L — Large: 3+ days, architectural change or multi-system coordination

Summary Statistics

Category	Count
Total edges inventoried	40
LIVE	15
DEAD	15
PARTIAL	10
Edges repairable with S fix	10
Edges repairable with M fix	8
Edges repairable with L fix	3

The factory has a 37.5% live edge rate. The remaining 62.5% of advertised flows are either dead or degraded. The 3 L-fixes (pi-orchestrator mock mode, chain runner, verifier auto-invocation architecture) unblock the most downstream flows if resolved. The 10 S-fixes are individually cheap and collectively close significant operational blind spots (cost reporting, RAG drain, blueprint score enforcement, monitoring, B2 backup).

Revision #2

Created 2026-05-09 19:44:17 UTC by John

Updated 2026-06-14 20:02:52 UTC by John