Virtual Company Architecture — Overview & Board Evaluation

Overview

ALAI operates a **multi-company virtual organization** where 16 specialized AI agent teams handle different domains. Each company has its own CLAUDE.md instructions, agent configurations, and domain expertise. Companies communicate through tasks (Mission Control) and knowledge entries (HiveMind) — never directly.

Last evaluated: 2026-03-31 by architecture board (Petter Graff, Martin Kleppmann, Kelsey Hightower, Chip Huyen + Devil's Advocate).

Company Registry

Company	Type	Domain	Status
CodeCraft	Dev Shop	Backend, APIs, databases, full-stack, fintech	□□ Active
Vizu	Agency	Frontend, UI/UX, design, branding, components	□□ Active
Datavera	Product Co	Data engineering, analytics, ML pipelines, SQL	□□ Active
Skybound	Product Co	SaaS product development, multi-tenant systems	□□ Active
Proveo	Audit Firm	QA, testing, code review, validation (READ-ONLY)	□□ Active
Securion	Consultancy	Security audit, pentest, vulnerability scanning	□□ Active
FlowForge	Consultancy	DevOps, CI/CD, IaC, monitoring, deployment	□□ Active
HelixSupport	Consultancy	Production support, SLA, incidents, hotfixes	□□ Merge candidate → FlowForge
Lexicon	Consultancy	Legal docs, compliance (GDPR/PSD2), ADRs	□□ Active

Finverge	Consultancy	Fintech, payments, accounting, open banking	◫ Active
Skillforge	Consultancy	Runbooks, training, knowledge management	◫ Merge candidate → Lexicon
Proxima	Agency	Marketing, growth, SEO, content	◫ Merge candidate → Lexicon
AgentForge	AI Lab	AI/ML ops, RAG, embeddings, model ops, HiveMind	◫ Active
Axiom	Consultancy	Software architecture, system design, blueprints	◫ Active
Entra	Orchestration Hub	Undefined — needs definition or removal	◫ Review
Resolver	Meta-Ops	Cross-company diagnostics, systemic fixes	◫ Active

Communication Architecture

Layer 1: Task Routing (Synchronous)

PI Orchestrator classifies tasks by keywords and routes to the appropriate company via `~/system/config/domain-to-company.json`.

```
Task created → PI Orchestrator classifies (Tier 1-5) → keyword match → company assignment → agent execution
```

Layer 2: Pipeline Chain (Automatic Handoff)

Sequential quality gates managed by `pipeline-engine.js`:

```
BUILD (CodeCraft/Vizu) → REVIEW (Proveo) → SECURITY (Securion) → OPS (FlowForge) → DOCS (Lexicon)
  ↑                          |
  └── BUILD-FIX (max 2 cycles) ↵  If REVIEW fails
```

Layer 3: Cross-Company Event Bus (Asynchronous)

Managed by `cross-company-bus.js`. Scans HiveMind entries, applies routing rules from `cross-company-routes.json` (9 rules), creates inter-company tasks.

Board finding (2026-03-31): Bus was effectively dead — 1 task/day despite running every 6h. Root causes: agentPatterns didn't match actual HiveMind agent names, keyword matching too narrow. **Fixed same day.**

Layer 4: Resolver Meta-Daemon

Runs every 6h via `resolver-daemon.js`. Detects systemic patterns (3+ same failure = pattern), creates H-priority fix tasks.

Layer 5: Decision Log (NEW — 2026-03-31)

Structured, queryable decision log in `mission-control.db`. CLI: `node ~/system/tools/decision.js`. Supports log, query, list, history, supersede. Append-only audit trail with supersession chains.

Where Communication Lives

Store	Purpose	Location
Mission Control DB	Tasks, pipeline stages, task history, decisions	`~/system/databases/mission-control.db`
HiveMind DB	Knowledge entries, intel, memos (23K+ entries)	`~/system/databases/hivemind.db`
Events DB	System event log, event bus	`~/system/databases/events.db`
Slack	Notifications (ops, exec, alerts channels)	alai-talk.slack.com
Session Logs	Per-session summaries	`~/system/memory/sessions/`

Internal Company Structure

Each company follows a standard layout:

```
~/companies/<Name>/
├── CLAUDE.md       # Mission, expertise, rules, way of working
├── config.json     # Model selection, tier overrides, blueprints
├── agents/         # Agent configurations (lead, builder, reviewer)
├── state/          # Persistent state
```

```
└── skills/         # Company-specific skills
```

Every company has 3 standard agents:

1. **Lead** — Orchestrator: reads task specs, decomposes work, assigns phases
2. **Builder** — Implements work per blueprint (model: Sonnet)
3. **Reviewer** — Validates output, READ-ONLY (model: Sonnet or local Ollama)

Key Orchestration Files

File	Purpose
`~/system/kernel/pi-orchestrator.js`	Main daemon — task intake, classification, routing, execution, quality gates (3,953 lines)
`~/system/kernel/pipeline-engine.js`	BUILD→REVIEW→SECURITY automatic chain
`~/system/kernel/cross-company-bus.js`	Batch HiveMind scanner + event routing
`~/system/kernel/resolver-daemon.js`	Systemic issue detection (6h cron)
`~/system/config/domain-to-company.json`	Keyword → company routing map
`~/system/config/cross-company-routes.json`	9 inter-company event routing rules
`~/system/tools/decision.js`	Decision log CLI (log, query, history, supersede)

Board Evaluation — 2026-03-31

Panel

Petter Graff (System Architect) · Martin Kleppmann (Distributed Systems) · Kelsey Hightower (Orchestration) · Chip Huyen (AI Quality) · Devil's Advocate

Verdict

Structure is sound but underutilized at ~20% capacity. Fix existing infrastructure before adding new layers.

Key Findings

1. **Cross-company bus was dead** — agentPatterns didn't match real agent names. Fixed.
2. **getCompanyOverride bug** — returned string instead of object, tier overrides silently failed. Fixed.

3. **Skill-improver never fired** — dead `task.skill` condition. Fixed.
4. **QA-19 skipped ALL checks for automated tasks** — zero quality gating on pipeline. Fixed (retained checks 5, 6, 11, 12).
5. **No decision log** — session decisions evaporated. Fixed (decision.js).
6. **No quality scoring** — only pass/fail, no continuous signal. Planned (Phase 2).
7. **No observability per company** — throughput, first-pass rate, cycle time not tracked. Planned (Phase 3).
8. **82 LaunchAgent plists** — daemon sprawl, should consolidate to ~20. Planned.

Recommendations (Priority Order)

#	Action	Effort	Status
1	Fix 5 existing bugs	1.5h	✅ Done
2	Decision log (decisions table + CLI)	2h	✅ Done
3	Quality score column + basic scoring	2h	⬜ Planned
4	Observability DB + agent_spans	2h	⬜ Planned
5	MC Dashboard Company Health tab	2h	⬜ Planned
6	Daemon consolidation (82→~20)	4h	⬜ Planned
7	Company merge (16→10-12)	3h	⬜ CEO decision needed

Design Principles (Confirmed by Board)

- No direct company-to-company calls — all through MC tasks or HiveMind
- No real-time event bus needed — priority-triggered scan sufficient
- SQLite is the right choice for this scale — no Prometheus/Grafana/OTel locally
- INSERT is the telemetry pipeline, SQL is the query language
- Fewer companies, better utilized > more companies with overhead

---

Revision #2
Created 2026-03-31 11:09:28 UTC by John
Updated 2026-05-31 20:05:24 UTC by John