Agent System Improvements — Multi-Team Orchestration (2026-03-30)

Agent System Improvements — Multi-Team Orchestration (2026-03-30)

Source: IndyDevDan video "One Agent Is NOT ENOUGH: Agentic Coding BEYOND Claude Code"
Analysis: Plan agent + Devil's Advocate + Petter Graff (AI Architect) **Status:** IMPLEMENTED

Background

CEO reviewed IndyDevDan's multi-team agentic coding system (PI agent harness) and requested improvements to ALAI's virtual company/agent system. Video covers: three-tier architecture (Orchestrator → Leads → Workers), persistent mental models, domain write-locking, skills sharing, config-driven teams, multi-perspective consensus, and conversation logs.

Gap Analysis

What ALAI Already Does Better

- **14 specialized companies** vs 3 generic teams — more sophisticated domain pipeline
- **QA-19 quality gate** (19-point verification) — no equivalent in IndyDevDan's system
- **21 ZAKONs** (behavioral laws) with hook enforcement — codified governance
- **RAG/HiveMind infrastructure** (23K+ entries, Qdrant vector search) — shared knowledge base
- **Local AI tier routing** (Ollama ANVIL + FORGE) — multi-backend model routing

- **skill-improver.js** — skills improve from failures (NM i KI 2026 pattern)

Gaps Identified

Gap	Description	Impact
Lead delegation not enforced	Lead agents could write code (behavioral rule only)	HIGH
Write-lock not enforced	YAML allowed_paths existed but hooks didn't check them	HIGH
No session conversation logs	Agents couldn't see what other agents did	MEDIUM-HIGH
RAG-first was advisory only	Agents skipped RAG queries with no consequence	HIGH
HiveMind posts unstructured	Free-form text, no company/domain/pattern tags	MEDIUM

Rejected Proposal: Per-Agent Mental Models

Architect recommendation: Do NOT implement persistent expertise.md files per agent. Reason: creates 4th knowledge layer (HiveMind + Knowledge DB + RAG cache + expertise.md) that will diverge. Instead, enforce RAG-first as blocking and improve HiveMind post quality with structured tags. This gives 80% of the benefit with zero new infrastructure.

Implemented Changes

Faza 1: Lead YAML Constraints + Write-Lock Enforcement

Lead YAML updates (16/16 companies): All lead.yaml files now have:

```
constraints:
  allowed_paths:
    - "~/companies/<CompanyName>/**"
  forbidden_paths:
    - "~/projects/**"
    - "~/ALAI/products/**"
    - "~/system/**"
```

```
    - "~/.claude/**"
  write_locked: true
```

Leads can READ everything but only WRITE to their own company directory.

Write-Lock Script: `~/system/tools/agent-write-lock.py`

- Called by hook system on Write/Edit operations
- Reads agent identity from `/tmp/builder-session-active`
- Looks up agent's YAML constraints
- Blocks writes outside allowed_paths
- Pure stdlib Python (no PyYAML dependency)

Faza 2: Session Conversation Logs

Tool: `~/system/tools/session-workspace.sh` Commands:

- `create <mc-task-id>` — Creates shared session directory
- `report <mc-task-id> <agent-type> <company>` — Writes structured JSON report (stdin)
- `read <mc-task-id>` — Shows all agent reports as markdown
- `clean [--older-than 24h]` — Cleanup

Agent Integration:

- `builder.md` — Step 3: Read session reports on boot. Step 4c: Write report on completion.
- `validator.md` — Session Awareness section: Read builder reports before validation.

Report schema: agent_type, company, task_summary, files_written, key_decisions, blockers, verification.

Faza 3: RAG-First Blocking + Structured HiveMind Tags

RAG Enforcement Upgrade:

- `~/.claude/hooks/lib/rag_first_enforcer.py` upgraded with path classification
- `~/projects/**` and `~/ALAI/products/**` → **BLOCKING** (exit 2 = Write/Edit denied)
- `~/system/**` → advisory (warn only)
- `~/companies/**` → always allowed
- Config: `~/.claude/hooks/config/rag-enforcement.json`

Structured HiveMind Posts:

- `~/system/tools/hivemind-post-structured.sh`

- Format: `[TYPE] [Company] [domain] [Project] pattern: message`
- Data includes: company, domain, pattern, project as JSON metadata
- `builder.md` Step 5 updated to use structured format

Architecture Diagram

```
CEO (Alem)
  └── John (Orchestrator, Opus)
        ├── company.sh → Route to company
        ├── consensus-query.js → Multi-company consensus (planned)
        │
        ├── CodeCraft (Lead: delegate-only, write-locked)
        │     ├── Builder (Sonnet, allowed_paths enforced)
        │     └── Validator (Sonnet, read-only)
        │
        ├── Proveo (Lead: delegate-only, write-locked)
        │     ├── QA Builder
        │     └── Audit Validator
        │
        └── ... (14 companies total)

Shared Infrastructure:
  ├── HiveMind (23K entries, structured tags)
  ├── RAG-first hook (BLOCKING for project files)
  ├── Session workspace (/tmp/session-{task-id}/)
  └── Write-lock enforcement (agent-write-lock.py)
```

Files Changed

File	Change
~/companies/*/agents/lead.yaml (x16)	Added constraints, write_locked: true
~/system/tools/agent-write-lock.py	NEW — write-lock enforcement script
~/system/tools/session-workspace.sh	NEW — shared session log tool
~/system/tools/hivemind-post-structured.sh	NEW — structured HiveMind post helper
~/.claude/hooks/lib/rag_first_enforcer.py	Upgraded — blocking for project paths

File	Change
~/.claude/hooks/config/rag-enforcement.json	Updated — project_mode: blocking
~/.claude/agents/builder.md	Updated — session awareness + structured posts
~/.claude/agents/validator.md	Updated — session awareness

Future Work

- **Consensus Query Tool** — Build when concrete architectural decision needs multi-company synthesis
- **Per-company cost tracking** — Token attribution per company per task
- **Lead operationalization** — Lead YAMLs are structural stubs; define real orchestration logic when needed

---

Revision #2
Created 2026-03-30 21:43:44 UTC by John
Updated 2026-05-31 20:05:24 UTC by John