ALAI Orchestration Architecture — Virtual Companies + Pi Agent Pipeline

ALAI Orchestration Architecture — Virtual Companies + Pi Agent Pipeline

System Overview

ALAI koristi **16 virtualnih kompanija** kao specijalizirane izvršne jedinice. Svaka kompanija ima svoj domen, alate, skills i blueprinte. **Pi Agent** (Ollama na FORGE/ANVIL) orkestrira izvršavanje kroz DAG pipeline.

```
graph TB
    subgraph USER["□□Alem (CEO)"]
        MC["Mission Control<br/>mc.js add/start/done"]
    end

    subgraph ORCHESTRATION["□□Orchestration Layer"]
        PI["pi-orchestrator.js<br/>TaskIntake → Classifier → Router"]
        DR["durable-runner.js<br/>DAG + SQLite Persistence"]
        HTTP["orchestrator-http-server.js<br/>REST API :3052"]
    end

    subgraph PIAGENT["□□Pi Agent (Ollama)"]
        MODEL["ollama:orchestrator<br/>Modelfile + System Prompt"]
        WORKER["forge-worker.js<br/>Action Interpreter"]
```

```
    end

    subgraph ROUTING["□□Routing"]
        CLASSIFY["Semantic Classifier<br/>qwen2.5-coder:32b"]
        DOMAIN["domain-to-company.json<br/>Keyword → Company"]
        SKILL["skill-resolver.js<br/>Company → Skill Path"]
        MCP["mcp-resolver.js<br/>Company → MCP Tools"]
    end

    subgraph COMPANIES["□□Virtual Companies (16)"]
        subgraph BUILD["BUILD Companies"]
            CC["CodeCraft<br/>Backend, APIs, DB"]
            VZ["Vizu<br/>Frontend, UI/UX"]
            DV["Datavera<br/>Data, ML, RAG"]
            SB["Skybound<br/>SaaS, Cloud"]
            FV["Finverge<br/>Payments, Fintech"]
        end
        subgraph REVIEW["REVIEW Companies"]
            PV["Proveo<br/>QA, Testing"]
            SC["Securion<br/>Security Audit"]
        end
        subgraph OPS["OPS Companies"]
            FF["FlowForge<br/>DevOps, CI/CD"]
            HS["HelixSupport<br/>Incidents"]
        end
        subgraph SUPPORT["SUPPORT Companies"]
            LX["Lexicon<br/>Legal, Docs"]
            PX["Proxima<br/>Marketing"]
            SF["Skillforge<br/>Training"]
        end
        subgraph META["META Companies"]
            AX["Axiom<br/>Architecture"]
            EN["Entra<br/>Orchestration Hub"]
            AF["AgentForge<br/>AI/ML Platform"]
            RS["Resolver<br/>Cross-Company Meta"]
        end
    end

    subgraph EXECUTION["⚙□ Execution"]
```

```
    BP["blueprint-runner.js<br/>Phase Gates"]
    QA["qa-19.js<br/>19-Point Quality Gate"]
    HM["HiveMind<br/>Knowledge + Intel"]
    BUS["cross-company-bus.js<br/>Inter-Company Routing"]
end

subgraph INFRA["□□□Infrastructure"]
    ANVIL["ANVIL (Mac Studio M3 Ultra)<br/>96GB, Ollama, Docker, SQLite"]
    FORGE["FORGE (Pi)<br/>Ollama: deepseek-r1:70b, qwen3:32b"]
    AZURE["Azure VM<br/>BookStack, Vault, Grafana, Sign"]
end

%% Flow
MC -->|"task"| PI
PI -->|"classify"| CLASSIFY
CLASSIFY -->|"domain"| DOMAIN
DOMAIN -->|"route"| COMPANIES

PI -->|"load DAG"| DR
DR -->|"expose API"| HTTP
HTTP <-->|"poll/execute"| WORKER
WORKER <-->|"generate actions"| MODEL

DOMAIN -->|"resolve skills"| SKILL
DOMAIN -->|"resolve MCP"| MCP

CC -->|"blueprint"| BP
VZ -->|"blueprint"| BP
BP -->|"verify"| QA

PV -->|"findings"| BUS
SC -->|"findings"| BUS
BUS -->|"route fixes"| CC
BUS -->|"intel"| HM

RS -->|"systemic scan"| BUS

MODEL -.->|"inference"| ANVIL
MODEL -.->|"inference"| FORGE
```

```
    style USER fill:#e1f5fe
    style ORCHESTRATION fill:#f3e5f5
    style PIAGENT fill:#fff3e0
    style ROUTING fill:#e8f5e9
    style COMPANIES fill:#fce4ec
    style EXECUTION fill:#fff8e1
    style INFRA fill:#f5f5f5
```

Task Flow — End to End

```
sequenceDiagram
    participant A as Alem (CEO)
    participant MC as Mission Control
    participant PI as pi-orchestrator
    participant CL as Classifier (qwen)
    participant CO as Company (e.g. CodeCraft)
    participant BP as blueprint-runner
    participant QA as qa-19.js
    participant HM as HiveMind

    A->>MC: mc.js add "Build payment API"
    MC->>PI: Task #5432 ready
    PI->>CL: Classify: "payment API fintech"
    CL-->>PI: Domain: FINTECH → Finverge
    PI->>CO: Route to Finverge.lead
    CO->>BP: Load api-backend.yaml

    loop Each Phase
        BP->>CO: Execute phase (builder agent)
        CO-->>BP: Phase output
        BP->>BP: Check gates (file_exists, npm test)
    end

    BP->>QA: qa-19.js check #5432

    alt Score >= 15/19
        QA-->>BP: PASS
```

```
        BP->>MC: mc.js done #5432
        MC->>HM: Post completion intel
    else Score < 15/19
        QA-->>BP: FAIL
        BP->>CO: Retry (max 2x)
    end
```

Pi Agent Protocol

```
sequenceDiagram
    participant W as forge-worker.js
    participant O as Ollama:orchestrator
    participant H as HTTP Bridge :3052
    participant D as durable-runner.js

    W->>H: GET /pipelines/{id}/ready
    H->>D: dagReady(id)
    D-->>H: ["auth"]
    H-->>W: ready_tasks: ["auth"]

    W->>O: "Task: auth, no deps, ready"
    O-->>W: {"action":"dag-start","dag_id":"...","task":"auth"}
    W->>H: POST /tasks/auth/start
    H->>D: dagStart(id, "auth")

    W->>O: "Execute auth task"
    O-->>W: {"action":"execute","instructions":"..."}

    W->>H: POST /tasks/auth/complete
    H->>D: dagComplete(id, "auth")
    D-->>H: unblocked: ["api","frontend"]
```

Company Structure

```
graph LR
    subgraph COMPANY["~/companies/CodeCraft/"]
```

```
        CJ["company.json<br/>Schema v2, routing keywords"]
        CF["config.json<br/>Models, tier overrides"]
        CM["CLAUDE.md<br/>Company rules"]

        subgraph AGENTS["agents/"]
            L["lead.yaml"]
            B["builder.yaml"]
            R["reviewer.yaml"]
        end

        subgraph BLUEPRINTS["blueprints/"]
            API["api-backend.yaml"]
            NX["nextjs-app.yaml"]
        end

        subgraph SKILLS["skills/"]
            S1["api-design/SKILL.md"]
            S2["code-review/SKILL.md"]
        end

        subgraph CONFIG["config/"]
            MC2["mcp.json (overlay)"]
            TL["tools.json"]
        end
    end

    style COMPANY fill:#e3f2fd
```

Resolution Chain

```
graph TD
    TASK["Incoming Task"] --> R1{"skill-resolver.js"}
    R1 -->|"1. Company skill"| CS["~/companies/X/skills/"]
    R1 -->|"2. ENV fallback"| EF["ALAI_COMPANY env var"]
    R1 -->|"3. Global"| GS["~/.claude/skills/"]

    TASK --> R2{"mcp-resolver.js"}
    R2 -->|"Base"| GB["~/.claude/mcp.json"]
```

```
    R2 -->|"Overlay"| CO2["~/companies/X/config/mcp.json"]
    R2 -->|"Merge"| MR["add + remove + override"]

    TASK --> R3{"blueprint-runner.js"}
    R3 -->|"Company blueprint"| CB["~/companies/X/blueprints/"]
    R3 -->|"Inheritance"| IH["extends: api-backend"]
    R3 -->|"Global fallback"| GT["~/system/templates/"]
```

Model Tier Selection

```
graph LR
    T1["Tier 1<br/>llama3.1:8b<br/>ANVIL"] -->|"escalate"| T2["Tier 2<br/>qwen2.5-coder:32b<br/>ANVIL→FORGE"]
    T2 -->|"escalate"| T3["Tier 3<br/>qwen3:32b<br/>FORGE"]
    T3 -->|"escalate"| T4["Tier 4<br/>Claude Sonnet<br/>API"]
    T4 -->|"escalate"| T5["Tier 5<br/>Human Queue<br/>Alem"]

    style T1 fill:#c8e6c9
    style T2 fill:#fff9c4
    style T3 fill:#ffe0b2
    style T4 fill:#f8bbd0
    style T5 fill:#ef9a9a
```

Cross-Company Communication

```
graph TB
    SC["Securion<br/>finds XSS"] -->|"HiveMind post"| HM["HiveMind DB"]
    HM --> BUS["cross-company-bus.js<br/>Route scanner (6h cron)"]
    BUS -->|"fix in blueprint"| CC["CodeCraft"]
    BUS -->|"regression test"| PV["Proveo"]
    BUS -->|"systemic pattern?"| RS["Resolver<br/>(meta-ops)"]
    RS -->|"if pattern found"| ALL["All affected companies"]

    style RS fill:#ffcdd2
```

Key Numbers

Metric	Count
Virtual Companies	16
SQLite Databases	54+
Tools (~/system/tools/)	1,310
Skills (~/.claude/skills/)	80+
Active Daemons	27-33
Model Tiers	5 (local → cloud → human)
QA Gate Checks	19 per task
Blueprints	~30 across companies

Last updated: 2026-03-21 by John Published to BookStack: System Architecture shelf

---

Revision #3
Created 2026-03-21 19:03:37 UTC by John
Updated 2026-05-31 20:05:22 UTC by John