Eklavya Architecture Document

> ***Version**: 1.0 **Last Updated**: January 2026 **Status**: Pre-Implementation Review*

Table of Contents

1. Executive Summary
2. System Architecture
3. Component Overview
4. Technology Stack
5. Agent System
6. Data Architecture
7. Communication Patterns
8. Security Model
9. Scalability Considerations
10. Integration Points

---

1. Executive Summary

What is Eklavya?

Eklavya is an **autonomous agent orchestration platform** designed to run a software development business. It takes project requirements, generates plans, builds demos for client approval, and executes full project builds - all with minimal human intervention.

Core Value Proposition

```
┌─────────────────────────────────────────────────────────────────────────┐
│                         EKLAVYA VALUE FLOW                               │
├─────────────────────────────────────────────────────────────────────────┤
│                                                                          │
│   CLIENT           ADMIN              EKLAVYA              DELIVERABLE   │
│                                                                          │
│   "I need an      Reviews &          Autonomous           Working        │
│    app that..."   Approves           Development          Software       │
│        │              │                   │                    │         │
│        └──────────────┴───────────────────┴────────────────────┘         │
│                                                                          │
│   • You sleep, Eklavya works                                             │
│   • Demos as sales tools (win contracts before full build)               │
│   • Multiple projects simultaneously                                     │
│   • Smart notifications (only when truly needed)                         │
│                                                                          │
└─────────────────────────────────────────────────────────────────────────┘
```

Key Architectural Decisions

Decision	Choice	Why
Agent Runtime	Claude Code CLI	Battle-tested tools, sandboxing, proven reliability

Database	PostgreSQL 16	JSONB flexibility, LISTEN/NOTIFY for real-time events
Message Queue	Redis 7	Fast pub/sub, caching, rate limiting
Web Framework	Next.js 14	Full-stack, real-time WebSocket support
Container	Docker	Project isolation, reproducible environments
AI Provider	Anthropic Claude	Best coding capability, tool use support

---

2. System Architecture

High-Level Architecture

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                              EKLAVYA SYSTEM ARCHITECTURE
│
├─────────────────────────────────────────────────────────────────────────────┤
│
│
│   ┌─────────────────────────────────────────────────────────────────────┐
│
│   │                              WEB LAYER                                │
│
│   │  ┌───────────────┐ ┌───────────────┐ ┌───────────────┐                │
│
│   │  │  Admin Portal │ │  API Gateway  │ │   WebSocket   │                │
│
│   │  │  (Next.js)    │ │   (REST)      │ │   Server      │                │
│
│   │  └───────┬───────┘ └───────┬───────┘ └───────┬───────┘                │
│
│   └──────────┼─────────────────┼─────────────────┼────────────────────────┘
│
│              │                 │                 │
│
│   ┌──────────┴─────────────────┴─────────────────┴────────────────────────┐
│
│   │                           CORE SERVICES                                │
│
│   │                                                                        │
│
│   │  ┌────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐          │
│
│   │  │  Project   │ │   Agent    │ │  Message   │ │  Learning  │          │
│
│   │  │  Manager   │ │  Manager   │ │    Bus     │ │   Engine   │          │
│
│   │  └─────┬──────┘ └─────┬──────┘ └─────┬──────┘ └─────┬──────┘          │
│
│   │        │              │              │              │                  │
```

```
│
│   │  ┌──────┴──────┐  ┌──────┴──────┐  ┌──────┴──────┐  ┌──────┴──────┐          │
│
│   │  │  Checkpoint │  │    Demo     │  │ Notification│  │    Cost     │          │
│
│   │  │   Service   │  │   Builder   │  │   Service   │  │   Tracker   │          │
│
│   │  └─────────────┘  └─────────────┘  └─────────────┘  └─────────────┘          │
│
│   └──────────────────────────────────────────────────────────────────────────────┘
│
│
│
│   ┌──────────────────────────────────────────────────────────────────────────────┐
│
│   │                          AGENT LAYER                                          │
│
│   │                                                                               │
│
│   │  ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐    │  │
│   │  │Orchestr-│ │Architect│ │Developer│ │ Tester  │ │   QA    │ │   PM    │    │  │
│   │  │  ator   │ │         │ │  (1..N) │ │         │ │         │ │         │    │  │
│   │  └────┬────┘ └────┬────┘ └────┬────┘ └────┬────┘ └────┬────┘ └────┬────┘    │  │
│   │       │           │           │           │           │           │         │
│
│   │       └───────────┴───────────┴───────────┴───────────┴───────────┘         │
│
│   │                               │                                              │
│
│   │                    ┌──────────┴──────────┐                                   │
│
│   │                    │  Claude Code CLI    │                                   │
│
│   │                    │  (Agent Runtime)    │                                   │
│
│   │                    └─────────────────────┘                                   │
│
│   └──────────────────────────────────────────────────────────────────────────────┘
│
│
│
│   ┌──────────────────────────────────────────────────────────────────────────────┐
│
│   │                          DATA LAYER                                           │
│
│   │                                                                               │
│
│   │  ┌────────────────────────┐       ┌────────────────────────┐                 │
│
│   │  │     PostgreSQL 16      │       │        Redis 7         │                 │
│
│   │  │                        │       │                        │                 │
```

```
│
│   │  │  • Projects          │       │  • Message queues     │                │
│
│   │  │  • Agents            │       │  • Pub/Sub channels   │                │
│
│   │  │  • Tasks             │       │  • Rate limiting      │                │
│
│   │  │  • Checkpoints       │       │  • Session cache      │                │
│
│   │  │  • Learning data     │       │  • Real-time events   │                │
│
│   │  │  • Audit logs        │       │                       │                │
│
│   │  └─────────────────────────┘       └─────────────────────────┘                │
│
│   └────────────────────────────────────────────────────────────────────────────────┘
│
│
│
│   ┌────────────────────────────────────────────────────────────────────────────────┐
│
│   │                        PROJECT ISOLATION                                          │
│
│   │                                                                                   │
│
│   │  ┌──────────────────┐  ┌──────────────────┐  ┌──────────────────┐                │
│
│   │  │   Project A      │  │   Project B      │  │   Project C      │                │
│
│   │  │   (Container)    │  │   (Container)    │  │   (Container)    │                │
│
│   │  │                  │  │                  │  │                  │                │
│
│   │  │  /src            │  │  /src            │  │  /src            │                │
│
│   │  │  /.eklavya/      │  │  /.eklavya/      │  │  /.eklavya/      │                │
│
│   │  │  /agents/        │  │  /agents/        │  │  /agents/        │                │
│
│   │  └──────────────────┘  └──────────────────┘  └──────────────────┘                │
│
│   └────────────────────────────────────────────────────────────────────────────────┘
│
│
│
└───────────────────────────────────────────────────────────────────────────────────────┘
```

Request Flow

```
┌─────────────────────────────────────────────────────────────────────────────────────┐
│                            REQUEST FLOW DIAGRAM
```

```
│
├─────────────────────────────────────────────────────────────────────────┤
│
│
│                                                                        │
│  ┌────────┐   ┌────────┐   ┌────────┐   ┌────────┐   ┌────────┐        │
│  │ Admin  │   │  API   │   │Project │   │ Agent  │   │ Claude │        │
│  │ Portal │   │Gateway │   │Manager │   │Manager │   │  Code  │        │
│  └───┬────┘   └───┬────┘   └───┬────┘   └───┬────┘   └───┬────┘        │
│      │            │            │            │            │              │
│      │  1. Create │            │            │            │              │
│      │──Project──>│            │            │            │              │
│      │            │  2. Init   │            │            │              │
│      │            │───────────>│            │            │              │
│      │            │            │  3. Create │            │              │
│      │            │            │───────────>│            │              │
│      │            │            │  container │            │              │
│      │            │            │            │            │              │
│      │            │            │  4. Spawn  │            │              │
│      │            │            │───────────>│            │              │
│      │            │            │Orchestrator│            │              │
│      │            │            │            │  5. Launch │              │
│      │            │            │            │───────────>│              │
│      │            │            │            │  process   │              │
│      │            │            │            │            │              │
│      │            │            │            │<───────────│              │
│      │            │            │            │  6. Ready  │              │
│      │            │            │            │            │              │
│      │            │<───────────────────────────────────-─│              │
│      │            │      7. Status updates via WebSocket │              │
│      │            │            │            │            │              │
│      │<───────────│            │            │            │              │
│      │  8. Real-  │            │            │            │              │
│      │  time UI   │            │            │            │              │
│      │            │            │            │            │              │
│
│
└─────────────────────────────────────────────────────────────────────────┘
```

3. Component Overview

Core Services

Service	Responsibility	Key Functions
Project Manager	Project lifecycle management	Create, configure, archive projects
Agent Manager	Agent lifecycle control	Spawn, monitor, terminate agents
Message Bus	Inter-agent communication	Route messages, handle broadcasts
Learning Engine	Prompt optimization	Track outcomes, evolve prompts

Checkpoint Service	State persistence	Save/restore agent state
Demo Builder	Demo orchestration	Parallel demo builds, preview URLs
Notification Service	Smart alerts	Prioritize, route, escalate
Cost Tracker	Budget management	Track spend, enforce limits

Service Interactions

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                        SERVICE INTERACTION MAP                              │
├─────────────────────────────────────────────────────────────────────────────┤
│                                                                             │
│                            ┌─────────────────┐                              │
│                            │  Project Manager │                             │
│                            └────────┬────────┘                              │
│                                     │                                       │
│                     ┌───────────────┼───────────────┐                       │
│                     │               │               │                       │
│                     ▼               ▼               ▼                       │
│           ┌──────────────┐  ┌──────────────┐  ┌──────────────┐              │
│           │ Agent Manager │  │ Demo Builder │  │ Cost Tracker │             │
│           └───────┬──────┘  └───────┬──────┘  └──────────────┘              │
│                   │                 │                                       │
│          ┌────────┴────────┐        │                                       │
│          │                 │        │                                       │
│          ▼                 ▼        ▼                                       │
│   ┌─────────────┐   ┌─────────────┐                                         │
│   │ Message Bus │◄──│ Checkpoint  │                                         │
│   └──────┬──────┘   │  Service    │                                         │
│          │          └─────────────┘                                         │
│          │                                                                  │
│          ▼                                                                  │
│   ┌─────────────┐   ┌─────────────┐                                         │
│   │ Notification │  │  Learning   │                                         │
│   │   Service   │   │  Engine     │                                         │
│   └─────────────┘   └─────────────┘                                         │
│                                                                             │
└─────────────────────────────────────────────────────────────────────────────┘
```

4. Technology Stack

Runtime Environment

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                            TECHNOLOGY STACK                                 │
├─────────────────────────────────────────────────────────────────────────────┤
│                                                                             │
│  PRESENTATION LAYER                                                         │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  Next.js 14          │  React 18           │  TailwindCSS           │   │
│  │  (App Router)        │  (Server Components)│  (Styling)              │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  APPLICATION LAYER                                                          │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  Node.js 20+         │  TypeScript 5       │  Zod                   │   │
│  │  (Runtime)           │  (Type Safety)      │  (Validation)          │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  AGENT LAYER                                                                │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  Claude Code CLI     │  Anthropic SDK      │  Docker                │   │
│  │  (Agent Runtime)     │  (API Client)       │  (Isolation)           │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  DATA LAYER                                                                 │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  PostgreSQL 16       │  Redis 7            │  Drizzle ORM           │   │
│  │  (Primary DB)        │  (Cache/Queue)      │  (Type-safe queries)   │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  INFRASTRUCTURE                                                             │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  Docker Compose      │  GitHub Actions     │  Vercel (optional)     │   │
│  │  (Local Dev)         │  (CI/CD)            │  (Deployment)          │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
│  TESTING                                                                    │
│  ┌─────────────────────────────────────────────────────────────────────┐   │
│  │  Vitest              │  Playwright         │  MSW                   │   │
│  │  (Unit Tests)        │  (E2E Tests)        │  (API Mocking)         │   │
│  └─────────────────────────────────────────────────────────────────────┘   │
│                                                                             │
└─────────────────────────────────────────────────────────────────────────────┘
```

Why These Choices?

Technology	Alternatives Considered	Why We Chose This
Next.js 14	Remix, SvelteKit	Best ecosystem, server components, Vercel integration
PostgreSQL	MySQL, MongoDB	JSONB flexibility, LISTEN/NOTIFY, mature ecosystem
Redis	RabbitMQ, Kafka	Simple, fast, perfect for our scale
Claude Code	Custom runtime	Battle-tested, sandboxed, full tool support
Docker	VM, bare metal	Reproducible, isolated, industry standard
Drizzle	Prisma, TypeORM	Type-safe, lightweight, SQL-like syntax

---

5. Agent System

Agent Types and Roles

```
┌────────────────────────────────────────────────────────────────────────────┐
│                          AGENT HIERARCHY                                   │
├────────────────────────────────────────────────────────────────────────────┤
│                                                                            │
│                         ┌─────────────────────┐                            │
│                         │   ORCHESTRATOR      │                            │
│                         │                     │                            │
│                         │  • Project control  │                            │
│                         │  • Agent spawning   │                            │
│                         │  • Task routing     │                            │
│                         └──────────┬──────────┘                            │
│                                    │                                       │
│         ┌──────────────────────────┼──────────────────────────┐            │
│         │                          │                          │            │
│         ▼                          ▼                          ▼            │
│  ┌─────────────┐           ┌─────────────┐           ┌─────────────┐       │
│  │  ARCHITECT  │           │     PM      │           │  MENTOR     │       │
│  │             │           │             │           │             │       │
│  │  • Design   │           │  • Require- │           │  • Research │       │
│  │  • Breakdown│           │    ments    │           │  • Guidance │       │
│  │  • Standards│           │  • Accept   │           │  • Best     │       │
│  └──────┬──────┘           └─────────────┘           │    practice │       │
│         │                                            └─────────────┘       │
│         │                                                                  │
│         ▼                                                                  │
│  ┌─────────────────────────────────────────────────────────────────┐       │
│  │                    IMPLEMENTATION TEAM                          │       │
│  │                                                                 │       │
│  │  ┌───────────┐  ┌───────────┐  ┌───────────┐  ┌───────────┐     │       │
│  │  │ Developer │  │  Tester   │  │    QA     │  │   UAT     │     │       │
│  │  │  (1..N)   │  │           │  │           │  │           │     │       │
│  │  │           │  │  • Unit   │  │  • E2E    │  │  • User   │     │       │
│  │  │  • Code   │  │  • Integ- │  │  • Visual │  │    flows  │     │       │
│  │  │  • Debug  │  │    ration │  │  • Perf   │  │  • UX     │     │       │
│  │  └───────────┘  └───────────┘  └───────────┘  └───────────┘     │       │
│  └─────────────────────────────────────────────────────────────────┘       │
│                                                                            │
│  ┌─────────────────────────────────────────────────────────────────┐       │
│  │                       OPERATIONS                                │       │
│  │                                                                 │       │
│  │  ┌───────────┐  ┌───────────┐                                   │       │
│  │  │    SRE    │  │ Monitor   │                                   │       │
│  │  │           │  │           │                                   │       │
│  │  │  • CI/CD  │  │  • Health │                                   │       │
│  │  │  • Deploy │  │  • Alerts │                                   │       │
│  │  │  • Infra  │  │  • Logs   │                                   │       │
│  │  └───────────┘  └───────────┘                                   │       │
│  │                                                                 │       │
│  └──────────────────────────────────────────────────────────────┘       │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

Agent Permissions Matrix

Agent	File System	Git	Terminal	Browser	Spawn Agents	Network
Orchestrator	Read	Read	No	No	Yes	Internal
Architect	Read/Write (docs)	Read	No	No	No	Research
Developer	Full	Full	Full	No	No	Package mgr
Tester	Read/Write (tests)	Read	Test cmds	No	No	Test only
QA	Read	Read	Test cmds	Full	No	Test only
PM	Read/Write (docs)	Read	No	No	No	Research
UAT	Read	No	No	Full	No	Test only
SRE	Full	Full	Full	No	No	Full
Monitor	Read	Read	Diagnostic	No	No	Metrics
Mentor	Read	Read	No	Full	No	Research

Agent Lifecycle

```
┌──────────────────────────────────────────────────────────────────────────┐
│                          AGENT STATE MACHINE                              │
├──────────────────────────────────────────────────────────────────────────┤
│                                                                           │
│                                                                           │
│                              ┌──────────┐                                 │
│                              │  SPAWN   │                                 │
│                              └────┬─────┘                                 │
│                                   │                                       │
│                                   ▼                                       │
│                              ┌──────────┐                                 │
│                              │   INIT   │                                 │
│                              └────┬─────┘                                 │
│                                   │                                       │
│                                   ▼                                       │
│                              ┌──────────┐                                 │
│                       ┌─────>│  READY   │<─────┐                          │
│                       │      └────┬─────┘      │                          │
│                       │           │            │                          │
│                       │           ▼            │                          │
│                       │      ┌──────────┐      │                            │
│                       │      │ WORKING  ├──────┤                            │
│                       │      └────┬─────┘      │                            │
│                       │           │            │                            │
│              ┌────────┼───────────┼────────────┼────────────┐               │
│              │        │           │            │            │               │
│              ▼        ▼           ▼            ▼            ▼               │
│         ┌────────┐┌────────┐┌──────────┐┌──────────┐┌──────────┐           │
│         │COMPLETE││PAUSED  ││CHECKPOINT││ BLOCKED  ││ FAILED   │           │
│         └────────┘└───┬────┘└────┬─────┘└────┬─────┘└────┬─────┘           │
│                       │          │           │           │                  │
│                       └──────────┴───────────┘           │                  │
│                                  │                       │                  │
│                                  ▼                       ▼                  │
│                              (Resume)              ┌──────────┐            │
│                                                    │TERMINATED│            │
│                                                    └──────────┘            │
│                                                                            │
└────────────────────────────────────────────────────────────────────────────┘
```

6. Data Architecture

Database Schema Overview

```
┌────────────────────────────────────────────────────────────────────────────┐
│                        DATABASE SCHEMA (SIMPLIFIED)                        │
├────────────────────────────────────────────────────────────────────────────┤
│                                                                            │
│  ┌──────────────────────────────────────────────────────────────────────┐  │
│  │                              PROJECTS                                │  │
│  ├──────────────────────────────────────────────────────────────────────┤  │
│  │  id              UUID PRIMARY KEY                                    │  │
│  │  name            VARCHAR(255)                                        │  │
│  │  client_name     VARCHAR(255)                                        │  │
│  │  description     TEXT                                                │  │
│  │  status          ENUM (planning, demo, building, complete, paused)   │  │
│  │  budget_limit    DECIMAL                                             │  │
│  │  budget_spent    DECIMAL                                             │  │
│  │  created_at      TIMESTAMP                                           │  │
│  │  config          JSONB                                               │  │
│  └────────────────────────────────────┬─────────────────────────────────┘  │
│                                       │                                    │
│             ┌─────────────────────────┼────────────────────────┐           │
│             │                         │                        │           │
│             ▼                         ▼                        ▼           │
│  ┌────────────────────┐  ┌────────────────────┐  ┌────────────────────┐    │
│  │       AGENTS       │  │       TASKS        │  │       DEMOS        │    │
│  ├────────────────────┤  ├────────────────────┤  ├────────────────────┤    │
│  │  id                │  │  id                │  │  id                │    │
│  │  project_id   FK   │  │  project_id   FK   │  │  project_id   FK   │    │
│  │  agent_type        │  │  agent_id     FK   │  │  demo_number       │    │
│  │  status           │  │  title            │  │  status           │     │
│  │  pid              │  │  status           │  │  preview_url      │     │
│  │  working_dir      │  │  priority         │  │  feedback         │     │
│  │  tokens_used      │  │  result           │  │  created_at       │     │
│  │  last_heartbeat   │  │  created_at       │  │  approved_at      │     │
│  └───────────────────┘  └───────────────────┘  └───────────────────┘     │
│              │                                                             │
│              ▼                                                             │
│  ┌───────────────────┐  ┌───────────────────┐  ┌───────────────────┐     │
│  │    CHECKPOINTS    │  │      MESSAGES     │  │  PROMPT_VERSIONS  │     │
│  ├───────────────────┤  ├───────────────────┤  ├───────────────────┤     │
│  │  id               │  │  id               │  │  id               │     │
│  │  agent_id     FK  │  │  project_id   FK  │  │  agent_type       │     │
│  │  state        JSONB│  │  from_agent   FK  │  │  version          │     │
│  │  file_snapshot    │  │  to_agent     FK  │  │  prompt_text      │     │
│  │  resume_instr     │  │  message_type     │  │  status           │     │
│  │  created_at       │  │  payload      JSONB│  │  success_rate     │     │
│  └───────────────────┘  └───────────────────┘  └───────────────────┘     │
│                                                                            │
│  ┌────────────────────────────────────────────────────────────────────┐   │
│  │                          AUDIT_LOGS                                  │    │
│  ├────────────────────────────────────────────────────────────────────┤   │
│  │  id            │  project_id  │  agent_id  │  action  │  details     │   │
│  │  timestamp     │  (nullable)  │  (nullable)│  VARCHAR │  JSONB       │   │
│  └────────────────────────────────────────────────────────────────────┘   │
│                                                                            │
└──────────────────────────────────────────────────────────────────────────┘
```

Data Flow

```
┌──────────────────────────────────────────────────────────────────────────┐
│                            DATA FLOW DIAGRAM                               │
├──────────────────────────────────────────────────────────────────────────┤
│                                                                            │
│   USER INPUT                                                               │
│       │                                                                    │
│       ▼                                                                    │
│   ┌──────────────┐      Write       ┌──────────────┐                      │
│   │  API Layer   │ ───────────────>│  PostgreSQL  │                      │
│   └──────────────┘                  └──────────────┘                      │
│          │                                │                                │
│          │ Notify                         │ LISTEN/NOTIFY                  │
│          ▼                                ▼                                │
│   ┌──────────────┐     Pub/Sub      ┌──────────────┐                      │
│   │    Redis     │ <──────────────>│   Services   │                      │
│   └──────────────┘                  └──────────────┘                      │
│          │                                │                                │
│          │ Subscribe                      │ Spawn                          │
│          ▼                                ▼                                │
│   ┌──────────────┐                  ┌──────────────┐                      │
│   │   Agents     │ <──────────────>│ Agent Files  │                      │
│   └──────┬───────┘    Read/Write    └──────────────┘                         │
│          │                                                                    │
│          │ Events                                                             │
│          ▼                                                                    │
│   ┌──────────────┐      Push        ┌──────────────┐                         │
│   │  WebSocket   │ ─────────────────>│  Admin UI    │                         │
│   └──────────────┘                  └──────────────┘                         │
│                                                                               │
└──────────────────────────────────────────────────────────────────────────────┘
```

---

7. Communication Patterns

Inter-Agent Messaging

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                       MESSAGE QUEUE ARCHITECTURE                              │
├──────────────────────────────────────────────────────────────────────────────┤
│                                                                               │
│   REDIS CHANNELS (per project):                                               │
│                                                                               │
│   eklavya:{projectId}:orchestrator     ← Orchestrator inbox                   │
│   eklavya:{projectId}:{agentId}        ← Specific agent inbox                 │
│   eklavya:{projectId}:broadcast        ← All agents in project                │
│                                                                               │
│   MESSAGE TYPES:                                                              │
│                                                                               │
│   ┌─────────────────┬──────────────────────────────────────────────────────┐  │
│   │  TASK_ASSIGN    │  Orchestrator assigns work to agent                   │  │
│   │  TASK_COMPLETE  │  Agent reports successful completion                  │  │
│   │  TASK_FAILED    │  Agent reports failure with error                     │  │
│   │  TASK_BLOCKED   │  Agent needs help (escalate to admin)                 │  │
│   │  TASK_PROGRESS  │  Agent reports progress update                        │  │
│   │  MENTOR_REQUEST │  Agent asks Mentor for guidance                       │  │
│   │  MENTOR_REPLY   │  Mentor provides guidance                             │  │
│   │  CHECKPOINT     │  Agent saved checkpoint                               │  │
│   │  HEARTBEAT      │  Agent is alive                                       │  │
│   └─────────────────┴──────────────────────────────────────────────────────┘  │
│                                                                               │
│   EXAMPLE FLOW:                                                               │
│                                                                               │
│   ┌────────────┐              ┌────────────┐              ┌──────────┐        │
│   │Orchestrator│              │   Redis    │              │ Developer│        │
│   └─────┬──────┘              └─────┬──────┘              └────┬─────┘        │
│         │                           │                          │              │
│         │  TASK_ASSIGN              │                          │              │
│         │ ─────────────────────────>│                          │              │
│         │                           │  (notify)                │              │
│         │                           │ ────────────────────────>│              │
│         │                           │                          │              │
│         │                           │            TASK_PROGRESS │              │
│         │                           │ <────────────────────────│              │
│          │            (forward)         │                            │        │
│          │ <────────────────────────────│                            │        │
│          │                              │                            │        │
│          │                              │              TASK_COMPLETE │        │
│          │                              │ <──────────────────────────│        │
│          │            (forward)         │                            │        │
│          │ <────────────────────────────│                            │        │
│                                                                               │
└───────────────────────────────────────────────────────────────────────────────┘
```

WebSocket Events (UI Updates)

```
┌───────────────────────────────────────────────────────────────────────────────┐
│                          WEBSOCKET EVENTS                                     │
├───────────────────────────────────────────────────────────────────────────────┤
│                                                                               │
│  Events pushed to Admin Portal in real-time:                                  │
│                                                                               │
│  ┌────────────────────────┬──────────────────────────────────────────────┐    │
│  │  Event Type            │  Payload                                     │    │
│  ├────────────────────────┼──────────────────────────────────────────────┤    │
│  │  project:created       │  { projectId, name, clientName }             │    │
│  │  project:status        │  { projectId, status, progress }             │    │
│  │  agent:spawned         │  { projectId, agentId, agentType }           │    │
│  │  agent:working         │  { agentId, taskId, description }            │    │
│  │  agent:completed       │  { agentId, taskId, result }                 │    │
│  │  agent:blocked         │  { agentId, reason, needsAdmin }             │    │
│  │  demo:ready            │  { projectId, demoNumber, previewUrl }       │    │
│  │  demo:approved         │  { projectId, demoNumber, decision }         │    │
│  │  budget:warning        │  { projectId, spent, limit, percent }        │    │
│  │  budget:exceeded       │  { projectId, spent, limit }                 │    │
│  │  notification          │  { level, title, message, actions }          │    │
│  └────────────────────────┴──────────────────────────────────────────────┘    │
│                                                                               │
└───────────────────────────────────────────────────────────────────────────────┘
```

---

8. Security Model

Security Layers

```
┌───────────────────────────────────────────────────────────────────────────────┐
│                           SECURITY ARCHITECTURE                               │
├───────────────────────────────────────────────────────────────────────────────┤
│                                                                               │
│  LAYER 1: AUTHENTICATION                                                      │
│  ┌───────────────────────────────────────────────────────────────────────┐    │
│  │  • Admin authentication (session-based)                               │    │
│  │  • API key authentication (for external integrations)                 │    │
│  │  • No client direct access (admin controls everything)                │    │
│  └───────────────────────────────────────────────────────────────────────┘    │
│                                                                            │
│  LAYER 2: PROJECT ISOLATION                                                │
│  ┌──────────────────────────────────────────────────────────────────────┐  │
│  │  • Each project runs in isolated Docker container                    │  │
│  │  • Separate file system per project                                  │  │
│  │  • Network isolation (no cross-project communication)                │  │
│  │  • Separate database schemas per project (optional)                  │  │
│  └──────────────────────────────────────────────────────────────────────┘  │
│                                                                            │
│  LAYER 3: AGENT SANDBOXING                                                 │
│  ┌──────────────────────────────────────────────────────────────────────┐  │
│  │  • Claude Code's built-in sandboxing                                 │  │
│  │  • Per-agent tool permissions (see matrix in Section 5)              │  │
│  │  • File system access restricted to project directory               │  │
│  │  • Network access limited by agent type                              │  │
│  └──────────────────────────────────────────────────────────────────────┘  │
│                                                                            │
│  LAYER 4: SECRETS MANAGEMENT                                               │
│  ┌──────────────────────────────────────────────────────────────────────┐  │
│  │  • Environment variables for API keys                                │  │
│  │  • .env files excluded from git                                      │  │
│  │  • Secrets never logged or exposed in UI                             │  │
│  │  • Per-project secrets isolation                                     │  │
│  └──────────────────────────────────────────────────────────────────────┘  │
│                                                                            │
│  LAYER 5: AUDIT & COMPLIANCE                                               │
│  ┌──────────────────────────────────────────────────────────────────────┐  │
│  │  • All agent actions logged to audit_logs table                      │  │
│  │  • Immutable audit trail                                             │  │
│  │  • Cost tracking per action                                          │  │
│  │  • Checkpoint history for recovery                                   │  │
│  └──────────────────────────────────────────────────────────────────────┘  │
│                                                                            │
└────────────────────────────────────────────────────────────────────────────┘
```

---

9. Scalability Considerations

Current Design (Single Instance)

```
┌────────────────────────────────────────────────────────────────────────────┐
│                    INITIAL DEPLOYMENT (Single Node)                        │
├────────────────────────────────────────────────────────────────────────────┤
│                                                                            │
│    ┌─────────────────────────────────────────────────────────────────┐     │
│    │                        SINGLE SERVER                            │     │
│    │                                                                 │     │
│    │   ┌─────────────┐  ┌─────────────┐  ┌─────────────┐             │     │
│    │   │   Next.js   │  │  PostgreSQL │  │    Redis    │             │     │
│    │   │    (Web)    │  │     (DB)    │  │   (Queue)   │             │     │
│    │   └─────────────┘  └─────────────┘  └─────────────┘             │     │
│    │                                                                 │     │
│   │   ┌──────────────────────────────────────────────────┐       │      │
│   │   │            Docker (Project Containers)            │       │      │
│   │   │  ┌──────────┐  ┌──────────┐  ┌──────────┐        │       │      │
│   │   │  │Project A │  │Project B │  │Project C │        │       │      │
│   │   │  └──────────┘  └──────────┘  └──────────┘        │       │      │
│   │   └──────────────────────────────────────────────────┘       │      │
│   │                                                               │      │
│   └──────────────────────────────────────────────────────────────┘      │
│                                                                          │
│   Capacity: ~5-10 concurrent projects                                    │
│   Agent limit: ~50 total concurrent agents                               │
│   Cost tracking prevents runaway                                         │
│                                                                          │
└─────────────────────────────────────────────────────────────────────────┘
```

Future Scaling (Not in v1.0)

The architecture is designed to scale horizontally when needed:

- PostgreSQL → Managed service (RDS, Supabase)
- Redis → Managed service (ElastiCache, Upstash)
- Projects → Kubernetes pods
- Load balancing → Multiple web instances

---

10. Integration Points

External Services

```
┌─────────────────────────────────────────────────────────────────────────┐
│                         INTEGRATION POINTS                               │
├─────────────────────────────────────────────────────────────────────────┤
│                                                                          │
│  REQUIRED INTEGRATIONS:                                                  │
│                                                                          │
│  ┌──────────────────────────────────────────────────────────────────┐   │
│  │  Anthropic API       │  Claude models for agent intelligence      │   │
│  │  GitHub              │  Code repository for projects               │   │
│  └──────────────────────────────────────────────────────────────────┘   │
│                                                                          │
│  OPTIONAL INTEGRATIONS (Future):                                         │
│                                                                          │
│  ┌──────────────────────────────────────────────────────────────────┐   │
│  │  Vercel              │  Deployment for demos and production       │   │
│  │  Stripe              │  Payment processing for client billing     │   │
│  │  Slack/Discord       │  Notification delivery                     │   │
│  │  Email (Resend)      │  Client communication                      │   │
│  │  S3/Cloudflare R2    │  File storage for assets                   │   │
│  └──────────────────────────────────────────────────────────────────┘   │
│                                                                          │
└─────────────────────────────────────────────────────────────────────────┘
```

Summary

This architecture provides:

- **Simplicity**: Chat-first interface hides all complexity
- **Autonomy**: Agents work while you sleep
- **Control**: Admin approval gates at key points
- **Isolation**: Each project fully sandboxed
- **Learning**: System improves over time
- **Cost Awareness**: Hard budget limits prevent surprises

The initial implementation focuses on a single-node deployment that can handle 5-10 concurrent projects. The architecture supports horizontal scaling when business demands it.

Document generated for pre-implementation review. Subject to refinement during development.