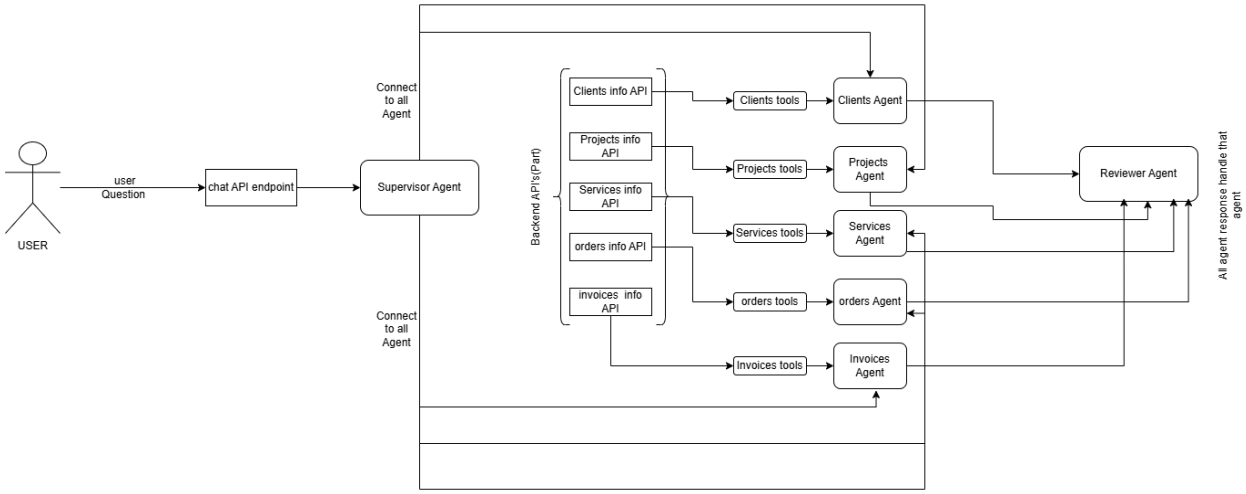
Hybrid AI Search / Natural Language Search Report

Multi-Agent Architecture Overview | Technical Documentation

1. Overview

Leafnet Hybrid AI Search is a conversational AI system that lets users ask natural-language questions about business data — clients, projects, services, orders, and invoices — and receive accurate, reviewed answers in real time. Instead of one large model trying to do everything, Leafnet uses a team of specialised AI agents, each an expert in one business domain, coordinated by a central Supervisor. This makes answers more accurate, easier to maintain, and simple to extend with new domains in the future.

2. Architecture Diagram



3. How a Question Flows Through the System

- **1. User asks —** The user sends a question to the Chat API endpoint (FastAPI).
- **2. Supervisor routes —** The Supervisor Agent analyses the question and decides which domain agent (or agents) should handle it.
- **3. Agent works —** The selected agent uses its MCP tools to call the relevant backend data API and fetch live data.
- **4. Reviewer validates —** The agent's draft answer goes to the Reviewer Agent, which checks correctness and quality.
- **5. Answer delivered —** The final, reviewed answer is returned to the user. The exchange is saved in Redis so follow-up questions keep full context.

4. The Agents

Agent	Responsibility
Supervisor Agent	Central coordinator. Receives every user question, understands the intent, and routes it to the correct domain agent(s).
Clients Agent	Answers questions about client records using the Clients Info API through its dedicated tool set.
Projects Agent	Handles project-related queries via the Projects Info API.
Services Agent	Handles service-related queries via the Services Info API.
Orders Agent	Handles order-related queries via the Orders Info API.
Invoices Agent	Handles invoice and billing queries via the Invoices Info API.
Reviewer Agent	Quality gate. Every agent response passes through the Reviewer, which validates and refines the answer before it is returned to the user.

5. Technology Stack

Technology	Role in Leafnet
Python + FastAPI	Exposes the Chat API endpoint that the client application calls. Handles requests asynchronously for fast, scalable responses.
LangChain + MCP Adapter	Framework used to build the Supervisor, domain agents, and Reviewer. The MCP (Model Context Protocol) adapter connects each agent to its tool server in a standard, plug-and-play way.
Backend Data APIs Laravel (PHP Framework)	REST APIs (Clients, Projects, Services, Orders, Invoices) provided by the backend team. Agents never touch the database directly — all data access goes through these APIs.
Redis Server	Stores per-user conversation memory, so Leafnet remembers the context of the ongoing chat and supports natural follow-up questions.

6. API Workflow – Laravel Integration

Each domain agent talks to the Laravel backend through its MCP tool server, which turns a tool call into a standard, authenticated REST API request. This keeps the AI layer and the PHP backend loosely coupled.

- **MCP Tool Server –** Acts as the bridge between an agent and Laravel, translating each tool call into a REST API request.
- **Authentication –** Requests are sent with a scoped API token (Laravel Sanctum), so each agent's access can be limited and revoked independently.
- **Request Handling –** Laravel routes the call through middleware, a Controller, and a Service layer before reading data via Eloquent.
- **Response –** Laravel returns a clean JSON payload (via an API Resource) that the agent uses to draft its answer.
- **Extensibility –** Adding a new domain only needs a new Laravel endpoint plus a matching MCP tool – no changes elsewhere in the pipeline.

7. Key Benefits

- **Accurate answers —** Each agent is a domain specialist working on live data from your existing APIs, with a Reviewer double-checking every response.
- **Conversational —** Redis-backed memory means users can ask natural follow-ups like "and what about his invoices?".
- **Easily extendable —** Adding a new domain (e.g. HR, Inventory) only requires a new tool server and agent — no changes to the rest of the system.
- **Secure by design —** Agents read data only through the approved backend APIs; nothing connects to the database directly.