# VC Copilot - Technical Overview

## 1. Project Overview

The goal is to build an AI-driven copilot for venture capital workflows.

The system connects to the tools that venture capitalists already use (e.g., AngelList, PitchBook, HubSpot, Google Workspace) and consolidates all the information into a single intelligent interface. It uses domain-specific reasoning to analyze and synthesize data across sources and will eventually be able to take automated actions through these integrations.

---

## 2. Product Description

### Core Functions

- **Data aggregation:** Pull and unify data from multiple platforms through APIs.
- **VC-specific reasoning:** Apply structured prompting and retrieval-augmented generation (RAG) to handle tasks such as sourcing, due diligence, portfolio analysis, and investment insights.
- **Unified interface:** Provide a single workspace where users can query, analyze, and visualize information across connected tools.

### Primary Goal

Create a centralized, AI-augmented environment that allows VCs to interact with all their operational data through one interface - reducing tool-switching and manual work.

---

## 3. Core Components

### 3.1 Integration Layer (APIs and Data Sync)

- Connect with third-party APIs (Google Workspace, Gmail, HubSpot, PitchBook, AngelList, etc.).
- Handle both data reads (fetching and aggregating) and, in later stages, limited data writes (e.g., creating files, updating records).

- Normalize incoming data into a consistent schema to allow for downstream reasoning and retrieval.

**Skills required:**

- REST API integration and OAuth2.0
- Data mapping and transformation
- Handling authentication tokens, rate limits, and webhooks

---

## 3.2 Reasoning Layer (AI + RAG Chatbot)

- The reasoning core uses **Retrieval-Augmented Generation (RAG)** to combine live or stored data with LLM reasoning.
- Incorporates a vector database to store embeddings of historical firm data, startup profiles, and past analysis.
- Uses structured prompting tailored to venture capital workflows (e.g., investment memos, market summaries, pipeline tracking).
- Produces context-aware responses and insights that reflect real VC reasoning patterns.

**Skills required:**

- LLM API usage (e.g., OpenAI, Anthropic, etc.)
- Prompt engineering and contextual memory design
- RAG pipeline development (indexing, embedding, and retrieval)
- Vector database setup (Pinecone, Weaviate, Supabase Vector, etc.)

---

## 3.3 Action Layer (Automation Engine) - *Later Stage Focus*

- Once the reasoning system is stable, introduce the ability to execute actions directly via connected APIs.
- Example actions include generating and saving an investment memo to Google Drive, updating CRM records, or drafting follow-up communications.
- Requires a secure execution framework with audit logging and user confirmations.

**Skills required:**

- Backend automation and workflow orchestration
- Secure API write operations and task queue management
- OAuth-based permission handling

*Note: Action execution is a secondary milestone and not the primary focus for the initial build.*

---

### 3.4 User Interface Layer

- Web-based interface where users interact with the copilot through natural language or structured commands.
- Chat interface paired with side panels for data visualization and source management.
- Emphasis on simplicity, fast responses, and clear feedback on data retrieval.

**Skills required:**

- Frontend development (React, Next.js, Tailwind CSS, shadcn/ui)
- OAuth integration and API authentication flows
- UX for chat-based and data-driven applications

---

## 4. System Architecture

**Frontend**

- Next.js / React for UI
- OAuth-based user authentication
- Chat interface connected to backend API endpoints

**Backend**

- Node.js or Python (FastAPI)
- REST endpoints for API integration and AI processing
- RAG pipeline for retrieval and contextual understanding

**Database**

- PostgreSQL or Supabase for structured data
- Pinecone, Weaviate, or Supabase Vector for embeddings

**Hosting**

- Vercel for frontend
- AWS or GCP for backend services
- Optional use of serverless functions for event-driven tasks

---

# 5. Development Phases

**Phase 1 - Core Infrastructure**

- Set up authentication and integration for 1–2 APIs (e.g., Google Workspace, HubSpot).
- Implement basic data ingestion and normalization.
- Deploy initial RAG chatbot that can query connected data sources and summarize insights.

**Phase 2 - Expanded Reasoning**

- Add more integrations (PitchBook, AngelList).
- Improve VC-specific prompt templates and retrieval logic.
- Implement history tracking, context caching, and data refresh cycles.

**Phase 3 - Action Execution (Later Stage)**

- Enable write operations (e.g., creating docs, updating CRM entries).
- Add workflow confirmation UI for safety and traceability.
- Introduce automation scheduling and multi-step processes.

**Phase 4 - UI/UX Polish and Reliability**

- Refine chat and dashboard design.
- Improve API stability and error handling.
- Optimize speed, context accuracy, and integration resilience.

---

# 6. Summary

The project focuses on building a retrieval-augmented, reasoning-based copilot tailored to the venture capital domain. The system aggregates data from multiple sources, provides domain-specific insights, and, at later stages, executes workflow automations.

The initial build centers on data connection, contextual reasoning, and a clean conversational interface. Automation and action execution come after the RAG-based foundation is stable and reliable.