AI-Product-CollabGen-Agent

Multi-AI Agent Collaboration System

Autonomous Cross-Functional AI Collaboration

Table of Contents

Project Overview

Purpose

Demonstrate autonomous AI agents simulating real-world cross-functional collaboration without human-in-the-loop

What It Does

- Generates product ideas, research insights, and marketing plans
- Works across multiple companies and domains (e.g., XR)
- Delivers structured Markdown reports automatically

Core Functionality

End-to-End Pipeline

- Dynamic inputs: company names, domain focus (e.g., XR)
- Sequential agent execution: Research → Product → Marketing
- Structured Markdown output stored in backend
- Downloadable and shareable reports

Key Features

Asynchronous

Non-blocking execution for faster processing and scalability

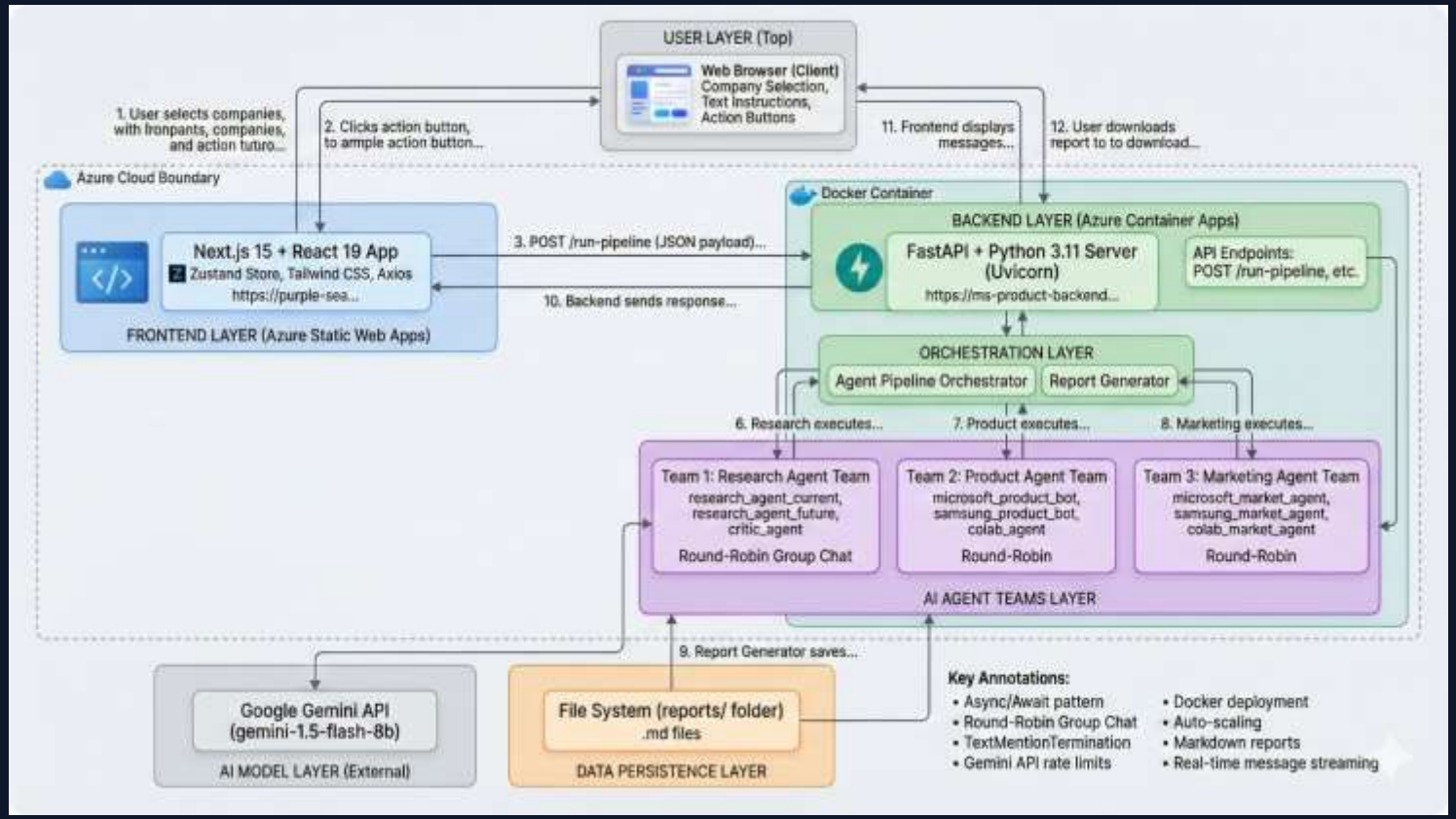
API-Driven

REST architecture for seamless integration

Cloud-Ready

Containerized for portability and CI/CD enabled

System Architecture



Complete Tech Stack

Backend :

Python 3.11
FastAPI (Uvicorn ASGI) •
Docker Microsoft Autogen •
Google Gemini 1.5 Flash
Pydantic • aiohttp •AsyncIO

Description:
Async, type-safe, AI-ready microservices architecture

Frontend :

Next.js 15 • React19
Tailwind CSS v4 • Zustand
Axios • Framer Motion
React Markdown •
Heroicons

Description:
Modern, performant, SSR-optimized responsive UI

Cloud & CI/CD:

Azure Container Apps
Azure Static Web Apps
Docker • GitHub Actions
OpenTelemetry

Description:
Scalable, observable, automated cloud deployment

Backend: Python & FastAPI

FastAPI

Modern Python web framework with automatic API docs (Swagger UI, ReDoc), type validation via Pydantic, and native async/await support

Uvicorn

ASGI server for production-grade performance, handling concurrent requests with minimal overhead

Python 3.x

Strong AI/ML ecosystem (NumPy, TensorFlow, Scikit-learn), rapid development cycle

Agent Framework: AutoGen

AutoGen v0.4 (January 2025)

- **autogen-core**: Agent lifecycle, state management, event-driven architecture
- **autogen-agentchat**: Multi-agent conversation patterns, structured messaging
- **autogen-ext[openai]**: OpenAI LLM integration with streaming support

Key Features

Asynchronous messaging, modular components, observability with OpenTelemetry

Agent Orchestration

Custom Agents

- Research Agent • Product Agent • Marketing Agent

Pipeline Logic

- Sequential execution using asyncio
- Output of one agent feeds the next
- Graceful cancellation support

Agent Responsibilities

Research Agent

Market research, technical analysis, competitive landscape

Product Agent

Product ideation, USP definition, feature planning

Marketing Agent

GTM strategy, regional insights, sales positioning

Agent	Role	Specific Tasks
Research Agent current	Current Business Analyst	Research present market operations of both companies, Gather current product portfolios & pricing, Analyze existing business relationships, Collect financial data & market share, Document current tech capabilities.
Research Agent Future	Future Technology Strategist	Explore future XR/VR roadmaps, Investigate R&D investments & patents, Analyze emerging technology trends, Identify potential collaboration opportunities, Assess future market positioning
Critic Agent	Data Validation Expert	Monitor discussion quality & completeness, Validate data accuracy with numbers, Ensure sufficient information gathered (min 3 data points), Approve when research is comprehensive, Trigger termination with "ENOUGH INFO"

Report Generation & APIs

Report Output

Structured Markdown with auto-formatted headers and bullet points, saved asynchronously

Core Endpoints

- /research-agent • /product-agent • /marketing-agent • /run-pipeline
- Utilities: List reports, download reports, delete reports

Backend Deployment

Infrastructure

- Dockerized backend service with multi-platform builds
- Azure Container Registry and Container Apps (auto-scaling)

Security

- API keys managed via environment variables

Frontend: Next.js 15 & React 19

Next.js 15

App Router, React Server Components, streaming & suspense, server actions, automatic static optimization

React 19

React Compiler, enhanced hooks, server component integration, improved form handling with use() hook

Styling & UX

Tailwind CSS v4, Framer Motion animations, Zustand state, Heroicons & Lucide icons

Frontend Architecture

App Router Structure

- Server Components for data fetching and optimization
- Client Components with Zustand for interactive state

Core Modules

- Company selector dropdown • Real-time chat interface with streaming
- Markdown report viewer with syntax highlighting

CI/CD Pipeline

GitHub Actions Workflow

- Triggered on push/PR to main branch
- Node.js setup with caching
- Build and static export
- Deploy to Azure Static Web Apps

Key Benefits

- Accelerates enterprise workflows through autonomous collaboration
- Demonstrates real-world AI agent orchestration patterns
- Modular and scalable design for future enhancements
- Cloud-native architecture enabling rapid deployment

Challenges

- API key dependency on external LLM providers
- LLM response variability and output consistency
- Experimental nature of Autogen agent libraries
- Context window limitations for large workflows

Use Cases

Enterprise

Collaboration tools for cross-functional teams

Research

Innovation and AI orchestration platforms

Future Enhancements

- Expand agent ecosystem: Sales, Finance, Legal agents
- Full Model Context Protocol (MCP) integration
- Advanced analytics dashboard and usage insights
- Role-based access control and multi-tenant support

Complete Tech Stack

Backend

Python 3.x • **FastAPI** (Uvicorn ASGI) • **AutoGen v0.4** • **Pydantic** • **aiohttp**

Async, type-safe, AI-ready

Frontend

Next.js 15 • **React 19** • **Tailwind CSS v4** • **Zustand** • **Framer Motion**

Modern, performant, SSR-optimized

Cloud & CI/CD

Azure Container Apps • **Docker** • **GitHub Actions** • **OpenTelemetry**

Scalable, observable, automated

FastAPI Deep Dive

Why FastAPI for This Project

- **Performance:** Benchmarks rival Node.js and Go, handles 1000s of concurrent requests
- **Auto Docs:** Swagger UI & ReDoc generated automatically from type hints
- **Type Safety:** Pydantic validation for request/response models
- **Async Native:** Built on Starlette + asyncio for true concurrent processing

AutoGen v0.4 Architecture

Agent Communication Model

- **Event-Driven:** Agents communicate via async message passing
- **Modular Design:** Pluggable components (custom agents, memory, LLMs)
- **Observability:** Built-in OpenTelemetry tracing for monitoring agent interactions
- **Scalability:** Supports long-running agents and proactive behaviors

Next.js 15 App Router Benefits

Modern Frontend Capabilities

- **Server Components:** Reduce JS bundle size, fetch data server-side securely
- **Streaming & Suspense:** Progressive UI rendering for faster perceived performance
- **Server Actions:** Form submissions and mutations without API layer overhead
- **Nested Layouts:** Share UI state across routes without full page reloads

Frontend-Backend Integration

Data Flow Architecture

- **Request:** Next.js client sends company + domain to FastAPI /run-pipeline
- **Processing:** AutoGen orchestrates Research → Product → Marketing agents concurrently
- **Streaming:** Agent outputs streamed in real-time to frontend via WebSocket/SSE
- **Storage:** Reports persisted to Azure Blob Storage, downloadable via signed URLs

Frontend-Backend Integration