TECHX

Asia's Largest

AI & Cloud Conference 2024

15 - 16, November 2024

Chennai Trade Center, Chennai

It’s me

Vikrant Guleria

Software Engineer III | Tune AI

About Me

A passionate technologist building innovative solutions with JavaScript & GenAI. I contribute to open-source, speak at conferences, and love sharing knowledge with the developer community. When I'm not coding, you'll find me exploring cutting-edge tech and bringing fresh ideas to life.

tune.

Live implementation of multi-agent system

tune.

Agenda

- Why Agents?
- What are AI Agents?
- How AI Agent works?
- Few Use case
- Diving into - Building a Zomato Reservation Agent with Multi-agent Architecture
  - High Level Architecture
  - Components to build agents
  - Agent implementation
- Live Demo
- Summary
- User Story

tune.


A.I. AGENTS
WE WERE WARNED
imgflip.com

What are AI Agents?

🤖 AI agents function as digital assistants that can be programmed to perform specific tasks autonomously

🎯 They can help automate repetitive processes, analyze data, and handle routine operations

💡 Create value by developing AI solutions that address real business needs and improve efficiency

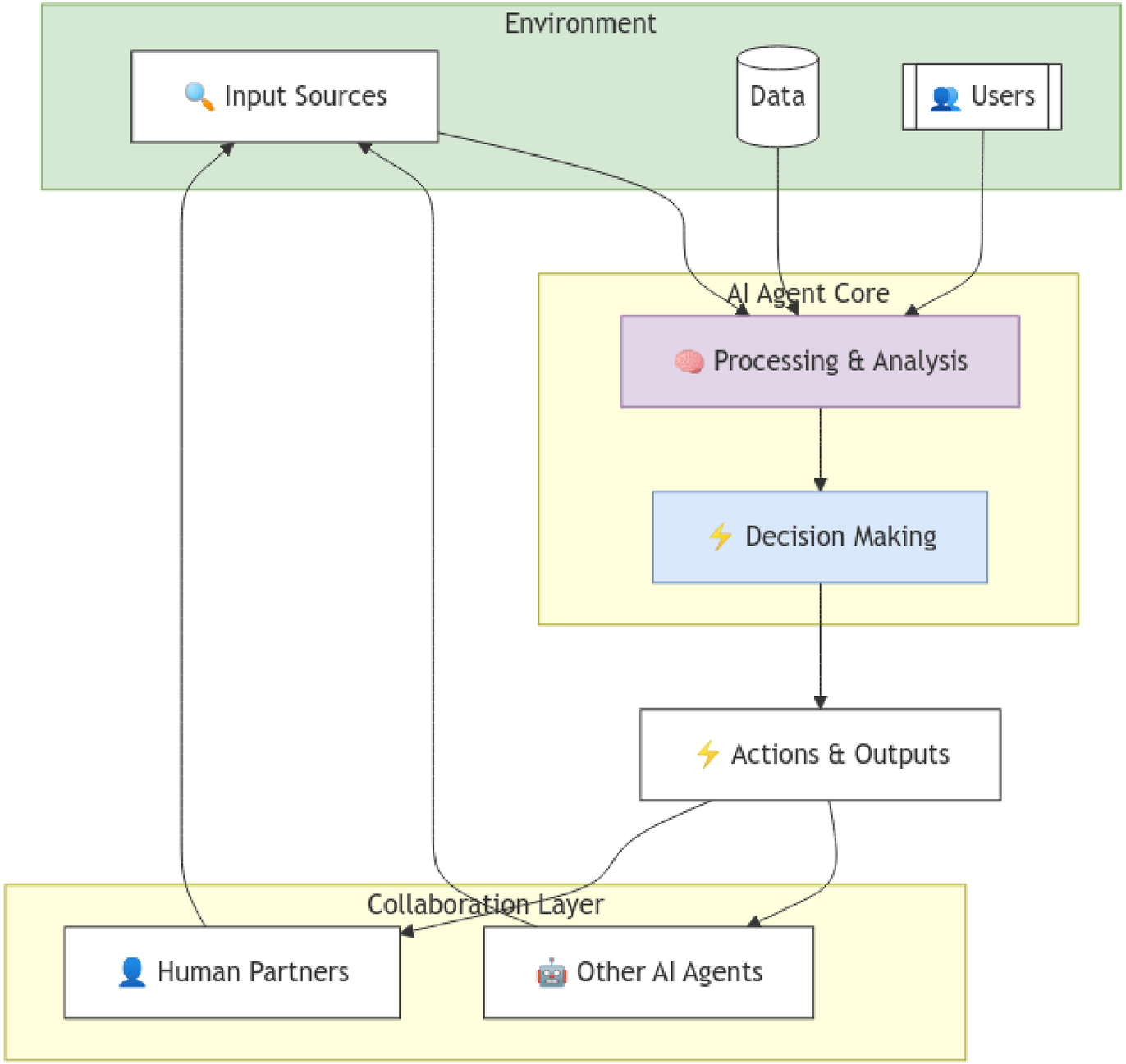
How AI Agents work

🔍 AI agents start by gathering and processing information from their environment through sensors, data inputs, or user interactions

🧠 They use sophisticated algorithms to analyze this information and understand patterns, just like a brain processing thoughts

⚡ Based on their analysis, agents make decisions and take actions autonomously, following their programmed objectives

🤝 They can collaborate with other AI agents and human users, sharing information and coordinating actions for better results


Environment
🔍 Input Sources
Data
👥 Users
AI Agent Core
🧠 Processing & Analysis
⚡ Decision Making
⚡ Actions & Outputs
Collaboration Layer
👤 Human Partners
🤖 Other AI Agents

Use cases

Use cases of LLM technology has gone from email writer to more intelligent agents that can operate autonomously.

Some of the cutting edge use cases are:

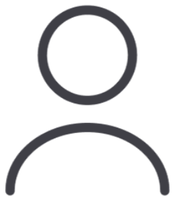
1. **Assistants to augment organisations as a 24x7 working employee**
    a. Sales Assistant
    b. Healthcare Assistant
    c. Research Assistant
    d. Software Engineer Assistant

2. **LLM technology to process data faster and cheaper**
    a. Processing PDFs to extract information
    b. Build knowledge graphs faster
    c. Reduce efforts in writing custom logic for complex documents types

Building a Zomato Reservation agent

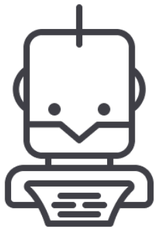
tune.

High Level Architecture

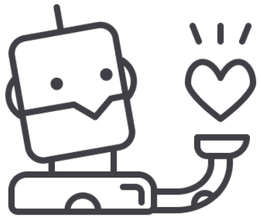


USER

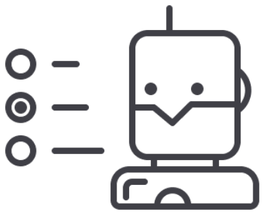
Manager Agent



Research Agent



Concierge Agent



Meeting Scheduler Agent

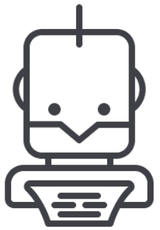
tune.

USER

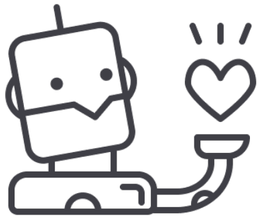
Hey can you conduct some research on TechXConf

Returns Response

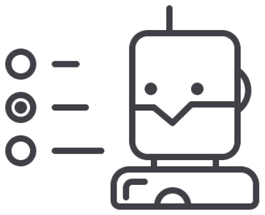
Manager Agent



Research Agent

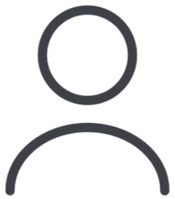


Concierge Agent



Meeting Scheduler Agent

tune.

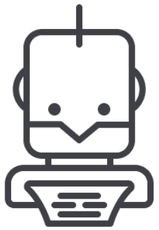


USER

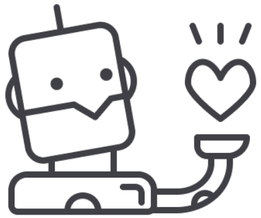
Can you book a meeting with vikrant at Madras Square for 8 PM

Returns Response

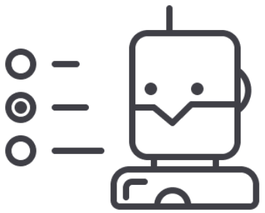
Manager Agent



Research Agent



Concierge Agent



Meeting Scheduler Agent

tune.

Components to build Zomato Reservation Agent

tune.

- **agents-js:** JavaScript package based on OpenAI swarm framework
- **Zomato API:** Get restaurants, slots and book table
- **Serper APIs:** Search web functionality
- **Tune/OpenAI API:** Access to AI models
- **Zapier API:** To Schedule meeting

Agent implementation

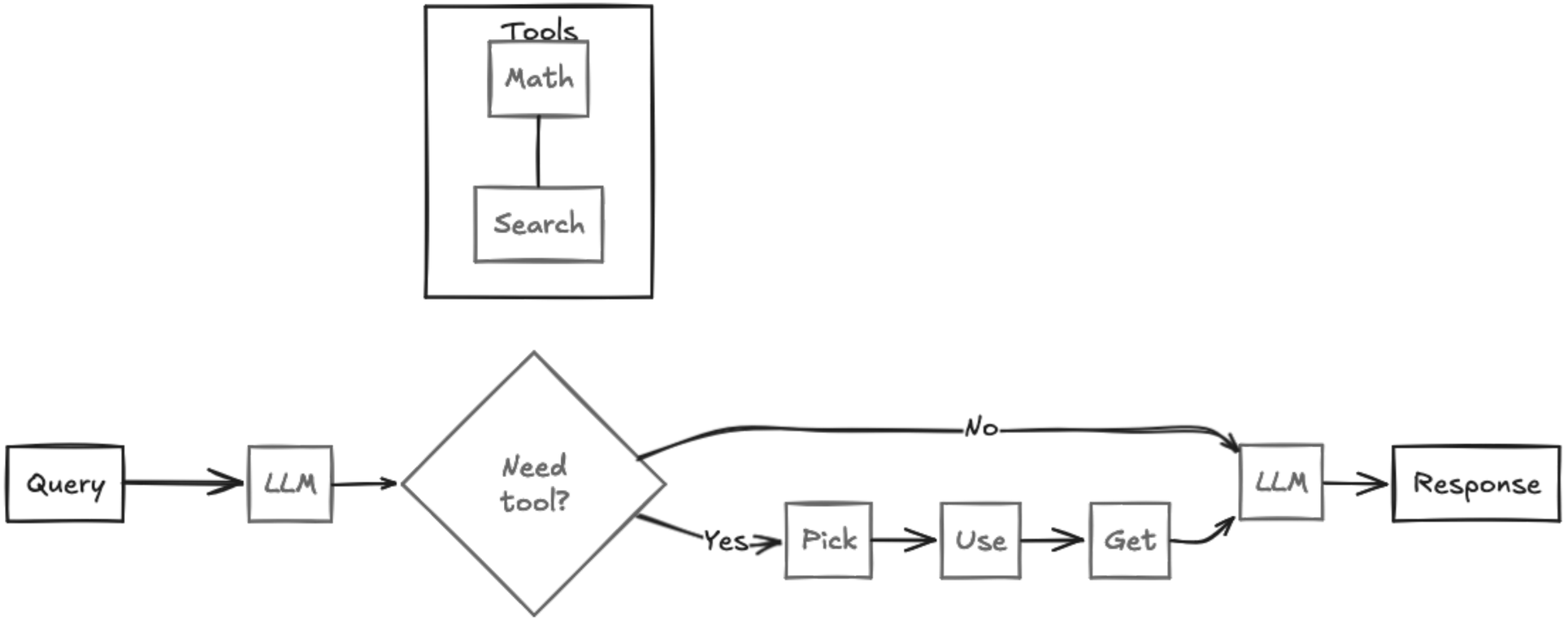
Create AI Agents

```
const agent = new Agent({
  name: "Calculator",
  instructions: "You can perform addition using the add function.",
  model: "gpt-4",
  functions: [add],
});
```

- **Name**: Identifies the agent.
- **Model**: Specifies the language model in use (e.g., "gpt-4”).
- **Instructions**: Provides the system prompt that defines the agent's behavior.
- **Functions**: Lists the callable functions enabling the agent to perform actions or access external features.

```
function add({ a, b }) {
  return new Result({ value: `${a + b}` });
}
```

Function Calling flow



Manager Agent

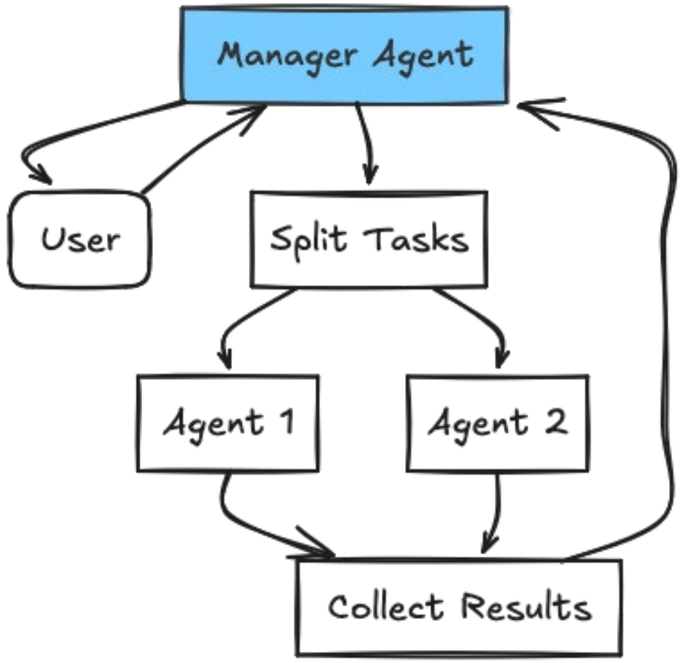
```
const managerAgent = new Agent({
  name: "Assistant",
  model: "gpt-4o-mini",
  instructions: (
    context
  ) => `You are a helpful personal assistant that coordinates with
specialized agents.

  Current context: ${JSON.stringify(context)}

  Key responsibilities:
  1. Coordinate multiple related tasks across different domains
  2. Maintain task context and state
  3. Ensure all related tasks are completed
  4. Handle task dependencies

  When handling multi-step tasks:
  1. Break down the task into subtasks
  2. Track completion of each subtask
  3. Pass relevant context between agents
  4. Confirm all subtasks are completed`,
  functions: [switchAgent],
  parallelToolCalls: false,
});
```

The Manager Agent acts as an orchestrator in a multi-agent system, efficiently coordinating complex tasks across specialized AI agents. Here are its key characteristics and responsibilities:

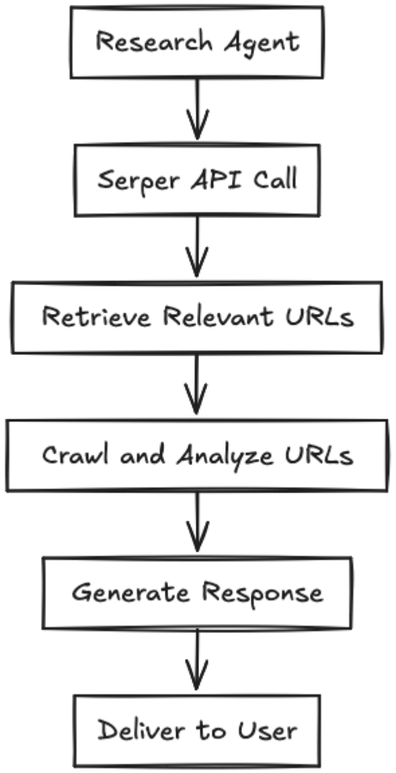


Research Agent

```
const researchAgent = new Agent({
  name: "Researcher",
  model: "gpt-4o-mini",
  instructions: `You are a research assistant specialized in gathering and
analyzing information.
  Help users with:
  - Conducting thorough research on specified topics
  - Analyzing and summarizing findings
  - Providing well-structured reports
  - Identifying credible sources and references

  Always verify the depth of research needed and ask clarifying questions to
narrow down the scope.
  Maintain academic rigor and cite sources when possible.
  Feel free to transfer to other agents for specialized tasks like scheduling
meetings based on research findings.`,
  functions: [conductResearch, switchAgent],
  parallelToolCalls: true,
})
```

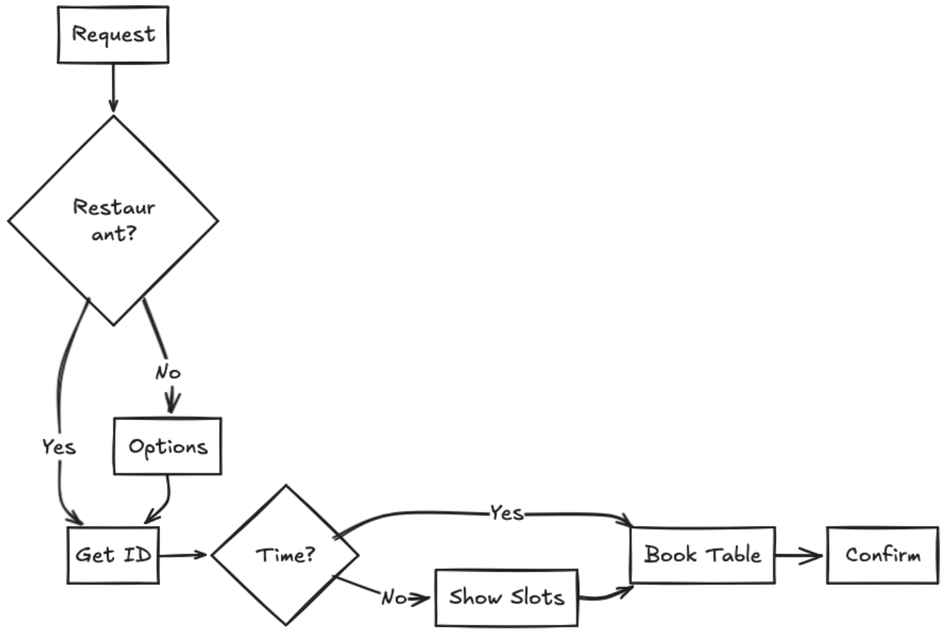
The Research Agent leverages the Serper API to perform targeted research on a given topic, crawling retrieved URLs to analyze content and generate comprehensive, accurate responses.



Concierge Agent

```
const conciergeAgent = new Agent({
  name: "Concierge",
  model: "gpt-4o-mini",
  instructions: `You are a helpful concierge assistant that helps users with
restaurant bookings and recommendations.
  Its important that when users ask about restaurants, help them find
suitable options and make reservations.
  If restaurant name is not given then show nearby restaurants, find
available time slots, and book tables.
  If slot id is not given then show available slots for the restaurant unless
time is specified, in which case book the table.
  Be proactive in asking for missing information.
  Feel free to transfer to other agents for specialized tasks.
  For getting restaurant id use getNearbyRestaurants function and for finding
slots use findSlots function.`,
  functions: [bookTable, getNearbyRestaurants, findSlots, switchAgent],
  parallelToolCalls: true,
})
```

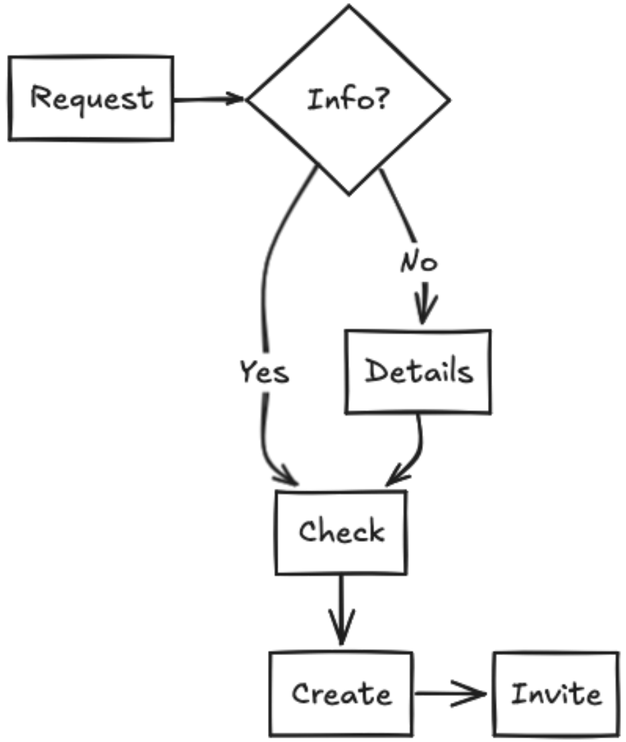
The Concierge Agent helps users find nearby restaurants, check table availability, and make reservations. It proactively fills in missing details and ensures a smooth booking experience using dedicated tools.



Meeting Scheduler Agent

```
const schedulerAgent = new Agent({
  name: "Scheduler",
  model: "gpt-4o-mini",
  instructions: `You are a scheduling assistant that helps users manage their
meetings and appointments.
  Help users schedule meetings, set reminders, and manage their calendar.
  '''Current Date and Time: ${new Date()} keep this in mind while scheduling
meetings.'''
  Always confirm date, time, and participants.
  Be proactive in suggesting suitable time slots and asking for missing
information.
  Feel free to transfer to other agents for specialized tasks like conducting
research to find suitable meeting times or booking a restaurant for a
meeting.`,
  functions: [scheduleMeeting, switchAgent],
  parallelToolCalls: true,
});
```

A Google Calendar-based scheduling agent that processes meeting requests by collecting necessary details (date, time, participants), checks availability and conflicts, creates calendar events, and sends email invites to participants.



LIVE DEMO

https://github.com/TuneHQ/multi-agent-demo

tune.

User Story

Tune AI has Saas and Enterprise users. I'll cover the Enterprise use cases from an F1 GP organiser from Middle East.

Every year they receive 1000+ RFPs for events to be organised from their clients. The team has to respond back with detailed 200+ pager proposals containing the precise details to be for the event. This proposal is then pitched back to the client by their sales team.

Our Assistant goes through 1000s of previous events and helps the sales teams build their proposals 10x faster, more accurate and creatively. It uses all the latest things like function calling, RAGs, etc. in a beautiful chat interface.

TECHX CONF

THANK YOU!

Vikrant Guleria

Software Engineer III | Tune AI

tune.