Crypto Market Intelligence Bot

1

Hi everyone, I'm Lucio, a Data Engineer originally from Italy, now living in Charlotte, NC.  
Today, I’d like to introduce a simple project I’ve been working on over the past week.  
I'll walk you through Agents, Workflows, and LLMs using a framework called Mastra AI.  
I’ll be presenting an automated workflow powered by Mastra that monitors cryptocurrency prices, analyzes trends, performs sentiment analysis to understand why a crypto is going up or down, and sends alerts to Telegram.

2

What is Mastra AI

Mastra AI is a powerful TypeScript framework designed for building intelligent automation workflows.  
It provides structured tools for orchestrating complex AI agent interactions, managing data flow, and easily integrating external APIs.  
The framework enables the creation of production-ready AI applications with built-in workflow management, agent coordination, and memory handling.

Here are the main components we’ll explore today. Some elements, like MCP, are not available in this demo.

The components include:

* **Agents** – AI-powered assistants with specific roles and capabilities.
* **Tools** – Modules for integrating external services and APIs.
* **Workflow** – Manages and coordinates multi-step processes.
* **Memory** – Stores context and conversation history

3

Let’s talk about the three key components that make AI automation really powerful: **Agents, Workflows, and LLM Integration.**

First, an **Agent** is basically your AI teammate. It’s designed to handle specific tasks or goals, using large language models to understand what’s going on, make decisions, and take action within a defined scope. You can think of it as a smart digital assistant that doesn’t just answer questions — it actually *does things*.

Then we have the **Workflow** — this is the backbone of automation. A workflow connects multiple steps together: API calls, agent actions, and data transformations. It’s what lets you move from a single task to an entire automated process, where everything happens in sequence or in parallel.

Finally, **LLM Integration** ties it all together. Large Language Models give agents their intelligence — they make sense of natural language, analyze messy or unstructured data, and generate clear, context-aware responses. That’s what allows an AI system to feel more “human” and adaptable.

4

This project is a **crypto intelligence bot** that brings together real-time market data, social media sentiment analysis, and automated notifications in a single, seamless workflow.  
It monitors cryptocurrency prices through the **CoinGecko API**, gathers relevant discussions from **Twitter**, analyzes community sentiment using **AI**, and sends clear, updates directly via **Telegram**.

Built with **TypeScript** and **Mastra AI**, the system is designed for reliability, featuring smart rate-limit handling and graceful fallback mechanisms to keep it running smoothly even under heavy load.

5

First, it **fetches cryptocurrency prices** from the **CoinGecko API** using REST endpoints.  
Next, it **collects recent social media posts** related to the same cryptocurrency.  
These posts are then passed to an **AI Agent**, which analyzes the tweets and **generates reasoning** behind the price movements.  
If the system hits a **rate limit**, this step is automatically skipped — I’ll explain more about that later.  
Finally, the **price data and sentiment analysis** are summarized, and a **notification message** is sent directly through **Telegram**.

6

Rate Limit

When the system reaches the **API rate limit**, we don’t want to waste tokens or make unnecessary calls that would only return a **429 error**.  
So, when this happens, the workflow automatically detects the 429 response and **skips the sentiment analysis step**, allowing the process to continue efficiently without interruption.

7

What Agents have we build: both use chat-gbt 4o

Crypto Price Agent : this is an assient for crypto currency market data

Analyze reason agent: this is a social sentiment analyst and market interprent

Together, these two agents work like a balanced team — one grounded in data, the other tuned into human emotion — giving us both the numbers *and* the narrative behind market movements.

8 9

Here’s a quick look at the technical side.  
The project is built in **TypeScript** using **Mastra AI** as the core framework.  
Each step in the workflow is defined with **Zod schemas**, which keeps the data validated and type-safe from start to finish.

Everything runs **asynchronously**, so API calls and AI responses happen efficiently without blocking the flow.  
We’ve also added smart **error handling**, so if we hit a rate limit, the system just skips that step instead of breaking.

The **Analyze-Reason Agent** even uses **streaming responses**, generating insights in real time, while sensitive credentials — like Telegram tokens — are securely managed through **environment variables**.

DEMO

Here we are for a live demo, so suppose we want to see the currencu for a Bitcoing. We type here ‘bitcoin’ select the currency in this case usd and run

We see here the workflow running, we can check the input and the output on each step

Here we go we receive our telegram message.

10

To conclude, this project shows how AI can turn data into real, actionable insight.  
With this workflow, **traders** get real-time market intelligence — not just prices, but the *mood* behind the market.  
**Investors** can automatically track their portfolios and receive meaningful updates without constantly checking charts.  
And **research teams** can analyze large volumes of sentiment data to understand market dynamics and refine investment strategies.

Beyond crypto, this demonstrates the power of **Mastra AI** — a framework that lets developers build complex, production-ready automation workflows that connect multiple data sources, integrate AI reasoning, and deliver real-time results — all with minimal code.