



# Talk to your Data Workshop

getAlignAI.com



# Agenda

## **Introduction & Setup**

Install Cursor, Python, and dependencies, then verify installations & environment setup

## **AI Chatbot Architecture**

Explain FAISS, SQL, and Graph-Based Search, and discuss why each component is needed and how it scales

## **Using Cursor**

Use Cursor to create and update the chatbot and create a webpage

## **Running and Testing the Chatbot**

Start the FastAPI server and send test queries using a web GUI or the webpage

## **Improvements, Debugging & Next Steps**

Troubleshoot common errors, discuss scalability, and explore enterprise deployment options



## Kristy Wedel

15+ years in analytics and visualization at marketing, finance, healthcare, retail & non-profit.

Organized internal analytics communities of collaboration & awareness in decentralized teams.



[linkedin.com/in/kristy-wedel/](https://linkedin.com/in/kristy-wedel/)





The content of this presentation is confidential. Do not share without explicit permission of AlignAI.



# AlignAI - AI Design Studio

Catalog and design business process, data sources, and decision points that can be augmented with AI

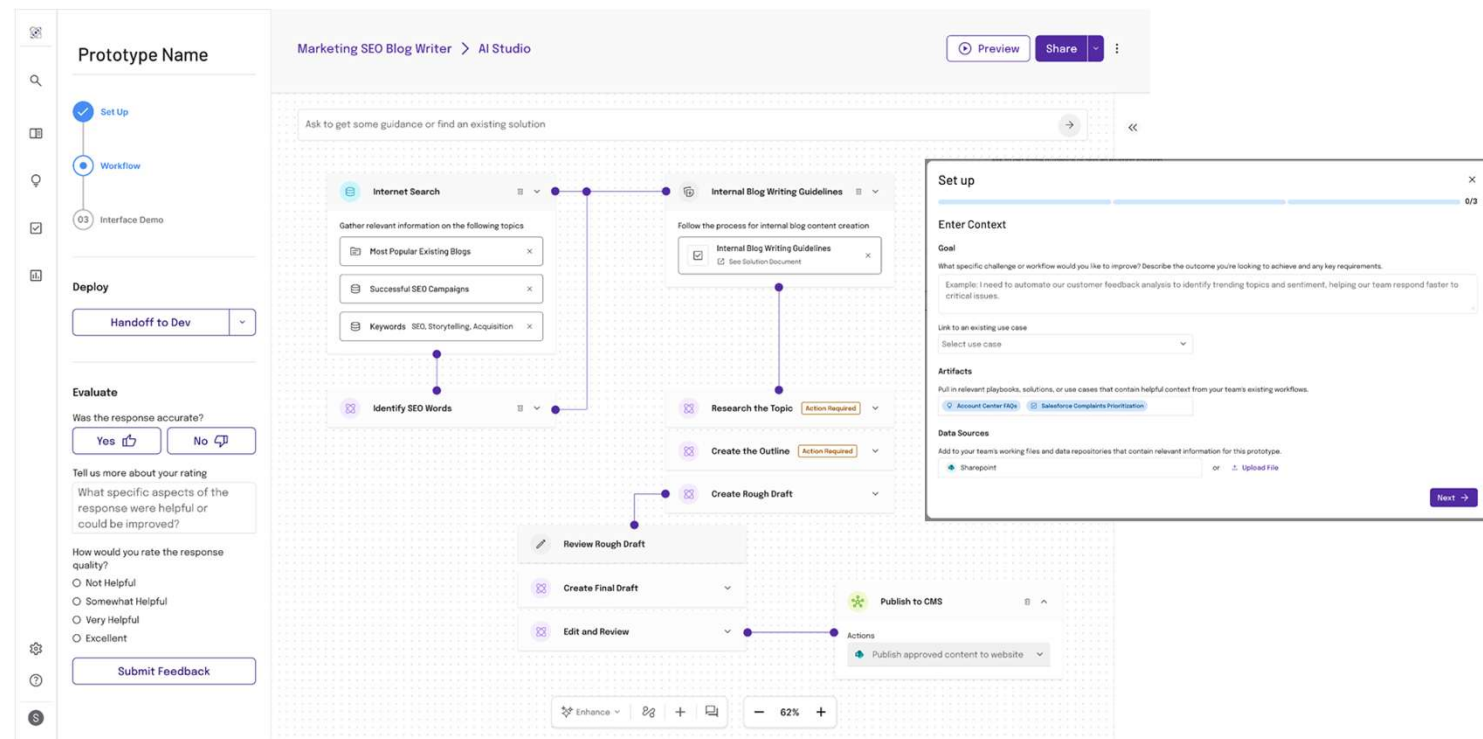
## Helps Feasibility

Speed up solution  
feasibility evaluation

### AI Design Studio

Creating & Managing  
Prototypes for AI Solutions

- Process Mapping
- Data Source Cataloging
- Artifact Referencing
- Response Evaluation
- Interface Testing
- Use Case Linking





# Installation

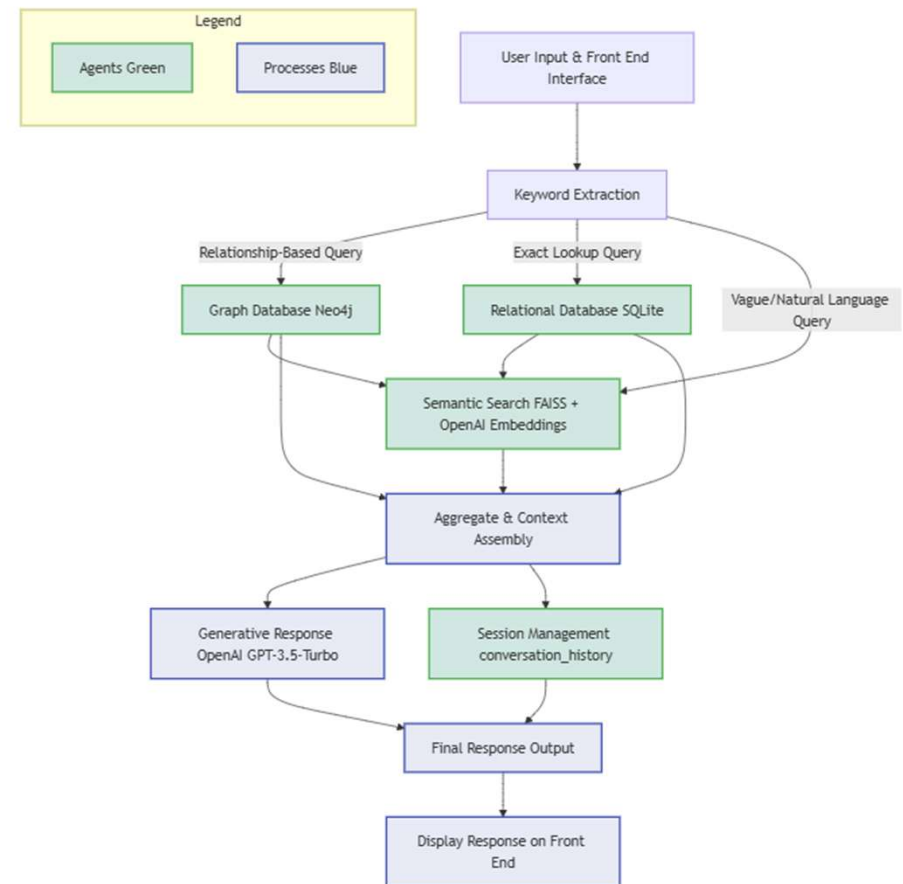


<https://bit.ly/4ibI6X7>

<https://github.com/kristywedel/DAMATalkWithYourData>

# What are we building

- **User Input:** Queries are submitted via the front-end interface.
- **Keyword Extraction:** The system identifies key terms to determine the query type.
- **Embedding Model:** Utilizes the embedding model to convert user queries and data into vector representations, enabling semantic similarity matching
- **Based on the query type, the system routes the query to one or more data sources:**
  - **Graph Database (Relationships in SQLite):** For relationship-based queries (e.g., "What ingredients are in CosmoLatte?").
  - **Relational Database (SQLite):** For exact data lookups (e.g., "What is the description of Lunar Macchiato?").
  - **Semantic Search (FAISS):** For vague or natural language queries (e.g., "What's your most energizing drink?").
- **Data Aggregation:** Results from all sources are combined.
- **Generative Response:** A free LLM generates a coherent response using aggregated data.



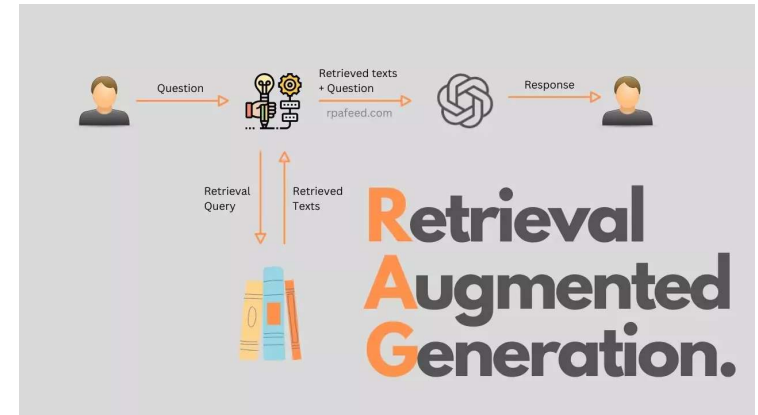
# Why is this Needed in Organizations

- Traditional chatbots rely on keyword-based search (limited results).
- Enterprises require context-aware, intelligent responses.
- AI-powered chatbots integrate structured and unstructured data for deeper insights.





# RAG Models



- **R**etrieval Component – Uses FAISS for similarity search and SQL/Graph-Based Search to retrieve relevant data efficiently.
- **A**ugmentation with Retrieved Data – Retrieves information from a database (or other source) before passing it to a language model for generating responses.
- **G**eneration Component – Integrated with an LLM (like GPT) for response generation after retrieval.

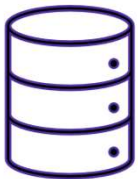


# How Chatbots Search

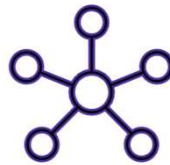
Feature	Traditional Apps	AI-Powered Chatbots
Data Retrieval	Exact keyword matching	Semantic similarity search
Context Awareness	Limited	High (via embeddings)
Scalability	Relational DBs (rigid schema)	Vector + Graph Search (scalable)
Knowledge Expansion	Pre-defined queries	Self-learning + evolving models

# The Three Pillars of AI Search

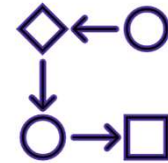
Structured Data (SQL /  
Relational DBs)



Semantic Search (Vector  
DB / FAISS)



Graph-Based Search  
(Relationships & Multi-Hop  
Queries)



# Workshop Data: BrewGalaxy Café

Beverage Details:	Menu and Specials:	General Inquiries:
“Tell me about CosmoLatte”	“What seasonal drinks are currently available at BrewGalaxy Café?”	“What makes BrewGalaxy Café unique?”

```
{
  "products": [
    { "name": "SolarFlare Espresso", "description": "A bold, triple-shot espresso blend with hints of caramelized citrus.", "category": "Beverage" },
    { "name": "CosmoLatte", "description": "A velvety latte infused with lavender and galaxy-inspired shimmer.", "category": "Beverage" },
    { "name": "Asteroid Cold Brew", "description": "A nitrogen-infused cold brew with a smooth, dark chocolate finish.", "category": "Beverage" },
    { "name": "Nebula Nitro", "description": "A nitro-charged coffee with a creamy texture, topped with stardust cinnamon.", "category": "Beverage" },
    { "name": "Lunar Macchiato", "description": "An ethereal macchiato layered with moon-spice syrup and a dash of starlight foam.", "category": "Beverage" },
    { "name": "Rocket Fuel Drip", "description": "An extra-caffeinated dark roast blend for an intergalactic energy boost.", "category": "Beverage" },
    { "name": "Milky Way Mocha", "description": "Rich dark chocolate with a hint of sea salt, blended with steamed oat milk.", "category": "Beverage" },
    { "name": "Zero-Gravity Matcha", "description": "A vibrant matcha latte topped with foamy clouds of vanilla cold foam.", "category": "Beverage" },
    { "name": "Interstellar Chai", "description": "Spiced chai tea latte infused with hints of black pepper and star anise.", "category": "Beverage" },
    { "name": "Galactic Refresher", "description": "A cold, sparkling hibiscus tea with floating boba pearls resembling tiny planets.", "category": "Beverage" },
    { "name": "Supernova Pumpkin Spice", "description": "A seasonal fall favorite with bold pumpkin spice flavors.", "category": "Seasonal Special" },
    { "name": "Comet Caramel Crunch", "description": "A winter special with caramel and crunchy toffee bits.", "category": "Seasonal Special" },
    { "name": "Meteor Mint Mocha", "description": "Holiday limited-edition mocha with cool mint and chocolate.", "category": "Seasonal Special" },
    { "name": "BrewGalaxy Mug", "description": "Glow-in-the-dark mug with constellations design.", "category": "Merchandise" },
    { "name": "Cosmo Cup", "description": "Reusable cup with NFC chip to track loyalty points.", "category": "Merchandise" },
    { "name": "Latte Art from Mars Workshop", "description": "Learn advanced latte art techniques with a cosmic twist.", "category": "Event" },
    { "name": "Caffeine & Constellations Night", "description": "Coffee tasting under the stars with expert astronomers.", "category": "Event" }
  ],
  "relationships": [
    { "entity1": "SolarFlare Espresso", "relationship": "similar_to", "entity2": "CosmoLatte" },
    { "entity1": "CosmoLatte", "relationship": "contains", "entity2": "Lavender" },
    { "entity1": "CosmoLatte", "relationship": "similar_to", "entity2": "Nebula Nitro" },
    { "entity1": "Nebula Nitro", "relationship": "similar_to", "entity2": "Milky Way Mocha" },
    { "entity1": "Asteroid Cold Brew", "relationship": "similar_to", "entity2": "Rocket Fuel Drip" },
    { "entity1": "Rocket Fuel Drip", "relationship": "stronger_than", "entity2": "Nebula Nitro" },
    { "entity1": "Lunar Macchiato", "relationship": "paired_with", "entity2": "Milky Way Mocha" },
    { "entity1": "Zero-Gravity Matcha", "relationship": "similar_to", "entity2": "Interstellar Chai" },
    { "entity1": "Supernova Pumpkin Spice", "relationship": "seasonal_with", "entity2": "Comet Caramel Crunch" },
    { "entity1": "Meteor Mint Mocha", "relationship": "seasonal_with", "entity2": "Comet Caramel Crunch" },
    { "entity1": "BrewGalaxy Mug", "relationship": "bundled_with", "entity2": "Cosmo Cup" },
    { "entity1": "Latte Art from Mars Workshop", "relationship": "event_for", "entity2": "CosmoLatte" },
    { "entity1": "Caffeine & Constellations Night", "relationship": "features", "entity2": "Galactic Refresher" }
  ]
}
```

# Structured Data - SQL for Knowledge Storage

## Why Use SQL Databases?

- Used for structured, transactional data.
- Guarantees consistency, reliability, and queryability.
- Example: Storing product descriptions, prices, inventory.

### Example Query in a Traditional App:

```
SELECT name, description FROM data_table WHERE category = 'Beverage';
```

#### Strengths:

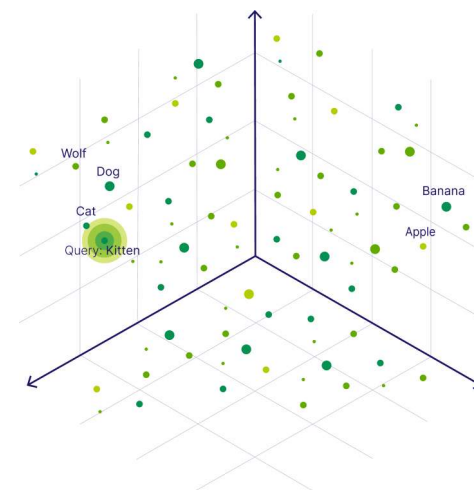
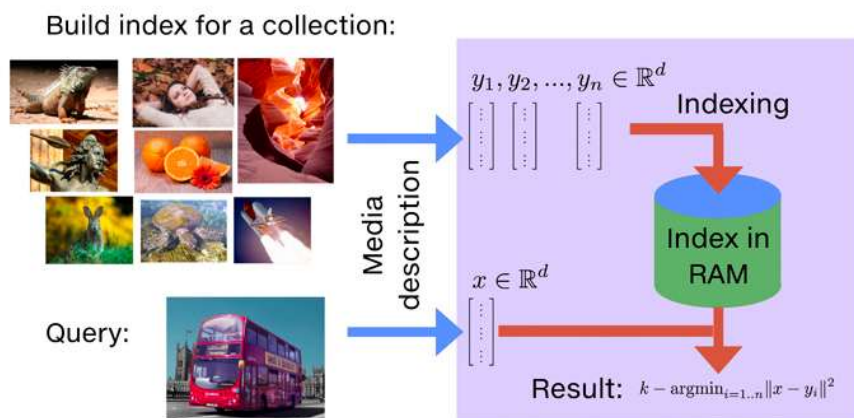
- ✓ Great for structured, tabular data.
- ✓ Ensures data integrity and accuracy.

✗ **Limitation:** Doesn't handle **fuzzy searches or meaning-based retrieval.**

# Semantic Search - FAISS and Vector Retrieval

**Facebook AI Similarity Search (Faiss)** is a library that allows us to quickly search for multimedia documents that are similar to each other

A **vector database** is a specialized database system designed for storing, indexing, and querying vector embeddings



# Computers need numbers to create relationships

Princess

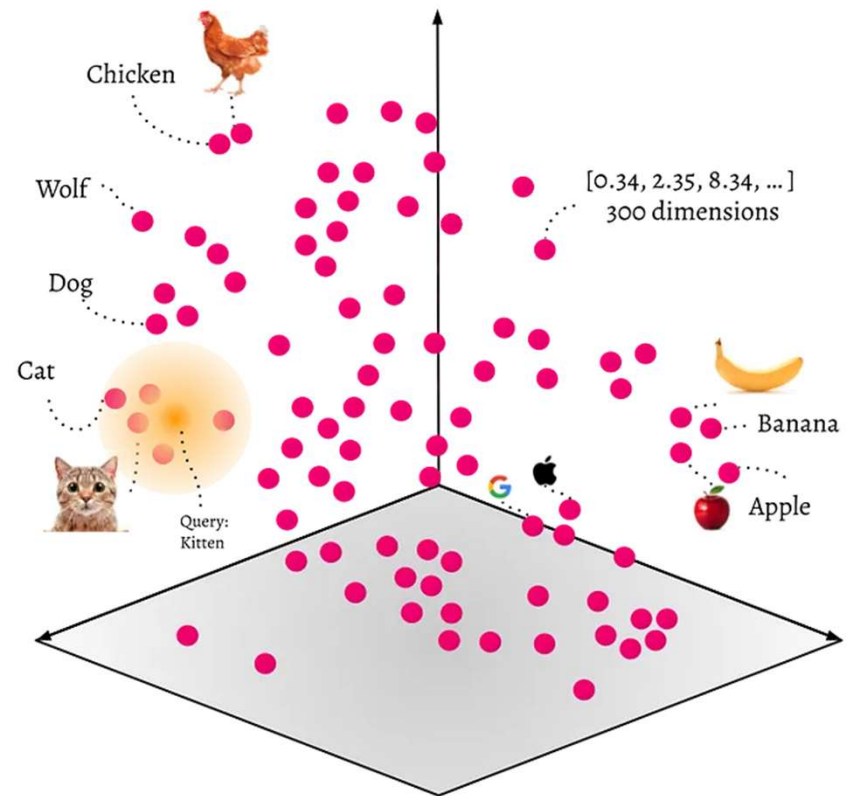


[0,0.2,1]

Queen



[0,0.2,2]



# Semantic Search - FAISS and Vector Retrieval

Factor	FAISS	Vector DB
Speed	✓ Faster for local search	◆ Optimized for distributed queries
Scalability	✗ Limited (RAM-based)	✓ Handles large-scale datasets
Persistence	✗ No built-in storage	✓ Data persistence & replication
Metadata Filtering	✗ Not supported	✓ Allows filtering with metadata
Cloud & Multi-user	✗ Local-only	✓ Supports cloud APIs, security, & multi-user access
Ease of Use	✓ Simple Python API	◆ Requires setup (managed or self-hosted)



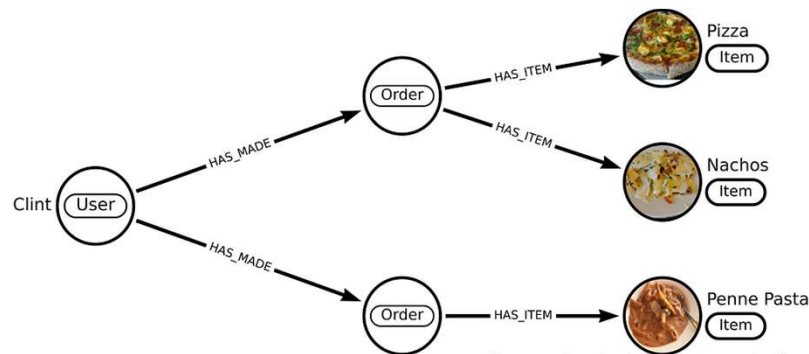
# Semantic Modeling and Search

## What is Semantic Modeling?

Semantic modeling defines relationships between data elements by embedding meaning and context into the data structure. It enables systems to understand, interpret, and reason about information in a way that mimics human cognition.

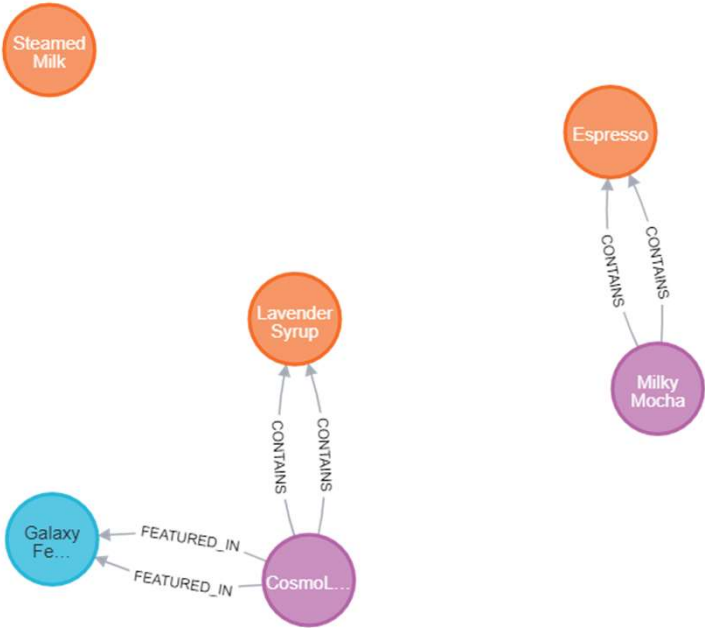
## What is Graph-Based Search?

Graph-based search is a technique that retrieves information based on the relationships between entities in a graph.



The content of this presentation is confidential. Do not share without explicit permission of AlignAI.

# Graph-Based Search - Finding Relationships in Data



entity1	relationship	entity2
CosmoLatte	contains	Lavender
SolarFlare Espresso	similar_to	CosmoLatte
Nebula Nitro	similar_to	Milky Way Mocha

# How These Components Scale in the Enterprise

## Scaling FAISS + Graph + SQL



### Component Breakdown & Scaling Strategy

- FAISS: Finds meaning-based matches
  - Scale using sharding across multiple servers
  - Scale with vector databases (e.g., Weaviate, Pinecone, Milvus)
- SQL: Stores structured business data
  - Scale with distributed databases (PostgreSQL, MySQL)
- Graph Queries: Maps relationships between data
  - Scale by moving to Neo4j if relationships grow complex

### Example Scaling Strategy for Large Enterprises

- Deploy FAISS on cloud servers (AWS, GCP, Azure)
- Store structured data in PostgreSQL/MySQL
- Use GraphQL or Neo4j for large relationship-based queries

# Workshop Data: BrewGalaxy Café

## Relationship-Based Queries

“What ingredients  
are in CosmoLatte?”

## Exact Lookup Queries (SQLite):

“What is the description of  
Lunar Macchiato?”

## Vague/Natural Language Queries (FAISS):

“What’s your most energizing  
coffee option?”



## All 3 Combined

“Tell me about CosmoLatte.”

# Privacy and Compliance – PII Detection

✗ Problem: Enterprises handle sensitive data (customer names, emails, phone numbers).

✓ Solution: Presidio for automated PII detection & anonymization.

🔍 Example:

- Input: “Hi, my name is John, and my email is john@example.com.”
- Output: “Hi, my name is [PERSON], and my email is [EMAIL\_ADDRESS].”



NEWS | PRIVACY

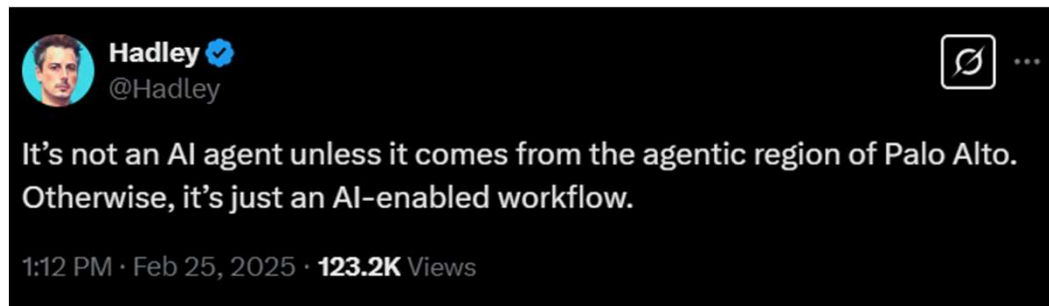
## City of Columbus breach affects around half a million citizens

Posted: November 4, 2024 by Pieter Arntz

# Improvements, Debugging and Next Steps

How could we make this solution better?

- Make it agentic
- Prompt Engineering
- Improve the UI (maybe with [Gen UI](#)?)
- Other ideas???





# Thank You!