AGC
THE CONSTRUCTION ASSOCIATION

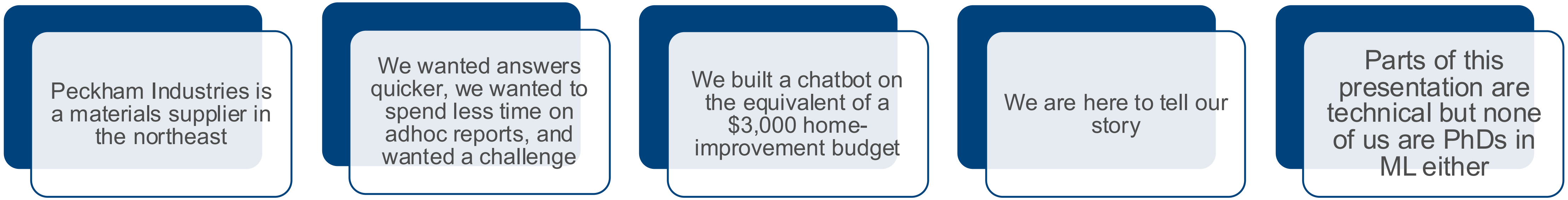
Chat With Your Data

Dan Decker, Peckham Industries

Russell Bradshaw, Peckham Industries

2025 AGC Technology Conference

Introduction



Peckham Industries is a materials supplier in the northeast

We wanted answers quicker, we wanted to spend less time on adhoc reports, and wanted a challenge

We built a chatbot on the equivalent of a $3,000 home-improvement budget

We are here to tell our story

Parts of this presentation are technical but none of us are PhDs in ML either

Quick Definitions

01

RAG: Retrieval Augmented Generation, using a vector store

02

RLHF: Reinforcement Learning Human Feedback (Thumbs Up or Thumbs Down)

03

Vector Store Database: Stores text embeddings, uses cosine similarity usually to identify text matches

Ex. Chroma DB

04

NLP: Natural Language Processing

Ex. Text to SQL

05

Fine-Tuning: In Large Language Models, it is a way to refine the way you want a model to respond to certain inputs

06

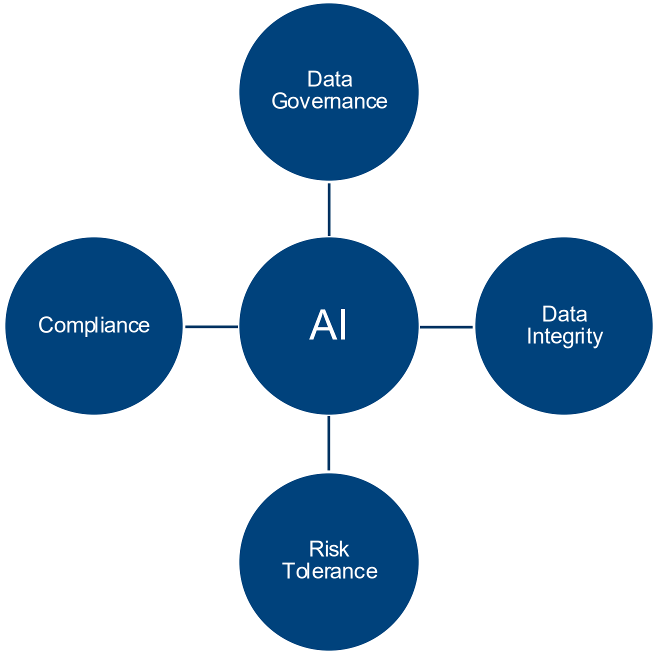
Tokens: The way a generative model measures inputs / outputs, usually a token is a word

Ex. Suggested word completion in a text

Key Considerations

GenAI is part of a bigger picture

- It is important to note that this presentation focuses on our technical journey, but we cannot neglect the big picture
- Data Integrity and Data Governance are key parts of the data and AI journey
- Risk tolerance and legal compliance are big factors when considering data you want exposed and how comfortable you are with it, and we are still sorting that out, too



The Start

It was a brute-force, part-time effort to figure any of it out, and we were in the dark

We deployed an instance of Azure OpenAI, and like any other novice, went to fine-tune before blowing close to $1000 in a day

We realized quickly fine-tuning would not be the answer, nor would most of the major services like Azure Cognitive search etc.

We did find that tokens were cheap, insanely cheap, which helped us find our way

The Breakthrough

We had several false starts and tries, but one of the first major breakthroughs came with NLP to SQL execution

By providing the database schema and some context, Azure OpenAI was able to intelligently write an SQL query, execute it, and return the results

We found an opensource project called Chainlit that helped with the UI, Authentication, and data persistence. Using Chainlit, most of these things became configurations rather than custom development

How it Works - Architecture

The bot is 100% Python, Chainlit serves a static React JS frontend for the user side

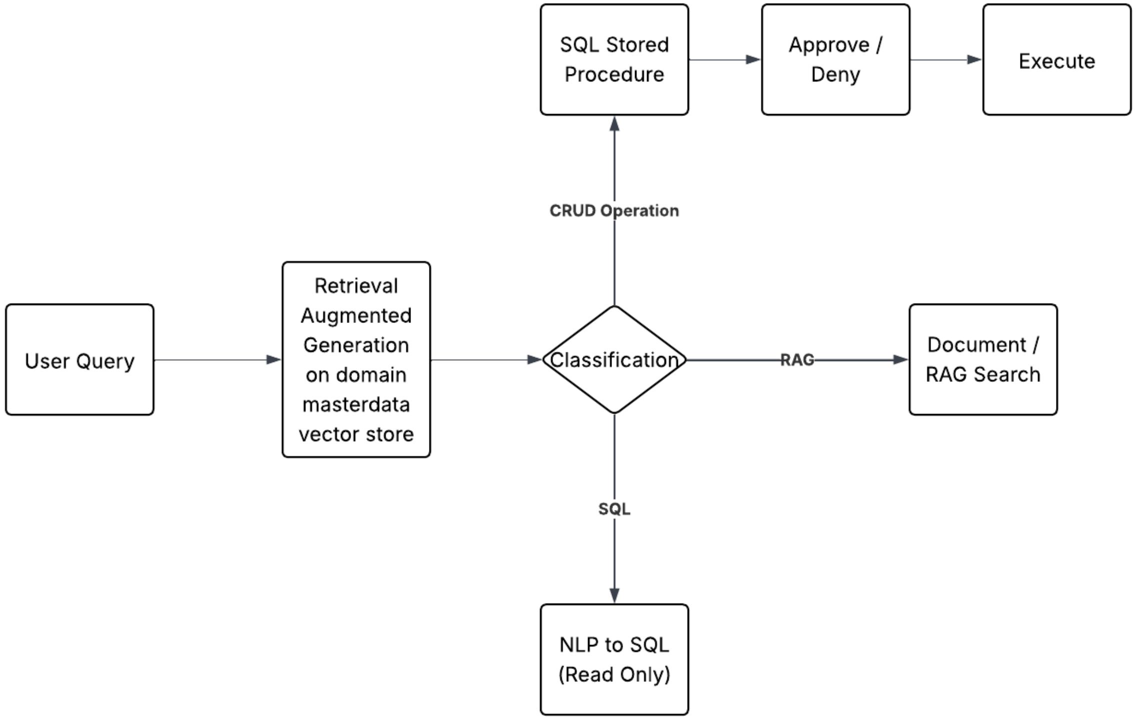
We use Azure OpenAI gpt-4o for the chat completion, which is just the 4o model for ChatGPT.

We use langgraph for orchestration, which allowed a modular approach to new data sources, meaning we do not update code to add data

The entire bot is built using context injection, meaning no fine tuning is ever used, keeping costs very low; we de-prioritize prior questions but include them in context for awareness of conversation thread

Every question is classified using Azure OpenAI + a RAG pipeline that looks at our metadata for customers, vendors etc. to inject additional context awareness

High-Level Flow



- Question comes in
- We send question to our vector store database (Chroma) to grab any domain meta data
- We add that to the prompt and then classify it
- Once classified, we either dynamically build and execute a SQL query, execute a stored procedure, or use RAG to grab document data

Cost

Beyond the labor / R & D of our people in-house, we have incurred just over $1,000 this year and will spend about $3,000 total for the entire year

$19.70 in tokens with Azure OpenAI YTD

$840 annualized spend on a vector store database

$2,760 Annualized on our App Service in Azure (Serves up full application)

Quick Demo

https://chat.peckham.com

Where We Go Next

Data Governance

Data Integrity

Optimization of performance and potentially RLHF

Questions?