AI-Powered Customer Support: for FAANG+ Companies

Instructor: Dr. Ashok Jallepalli

Introduction – Ashok Jallepalli

FACEBOOK Meta

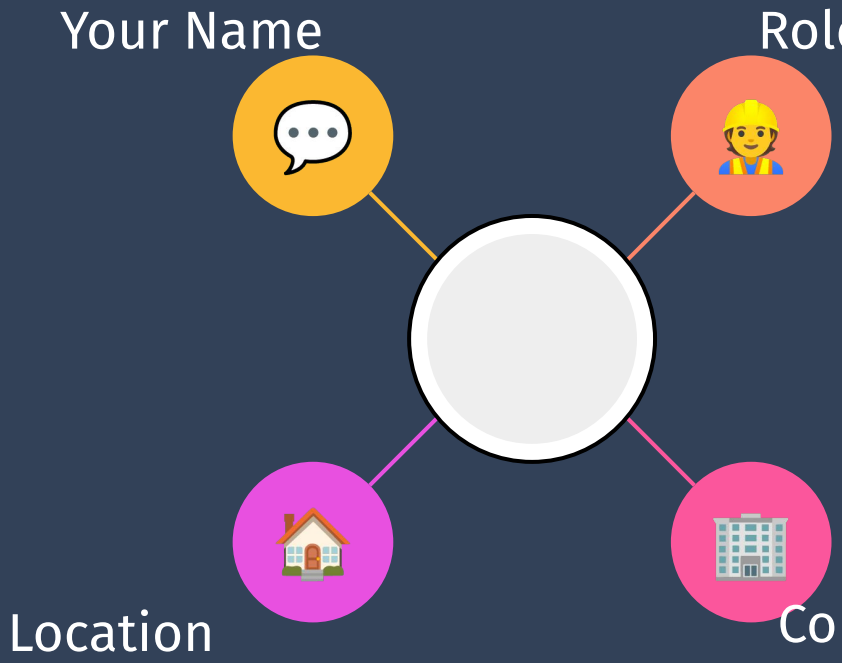
Senior Machine Learning Engineer, Master Control
focus on -> Efficient *LLMs*

formerly @
- Tebra (Engineering Manager)
- Meta/Facebook (Research Data Scientist))
- SCI at University of Utah (Researcher)
- Microsoft (Program Manager)

SCI

Education
Phd in Applied Numerical Mathematics.

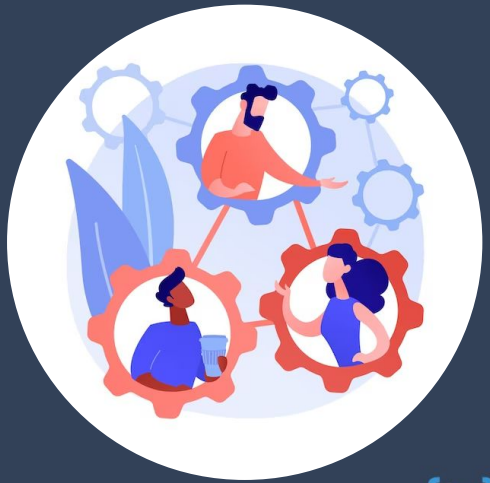
Welcome to today's class, before we begin, pop into the chat:



Your favourite super power

Optimize Your Experience

✅ Interact with your instructors via live class.
✅ Don’t be shy to speak up and get clarifications (Live Class..Duh!)



Today's Agenda

Introduction to the Problem Statement

How Generative AI Can Provide Solutions

Live Demo: Building a Chat Application Using LLM

High-Level Design of the Chat Application

Enhancing Capabilities with Retrieval-Augmented Generation (RAG)

Roles and Responsibilities Overview

Summary and Conclusion

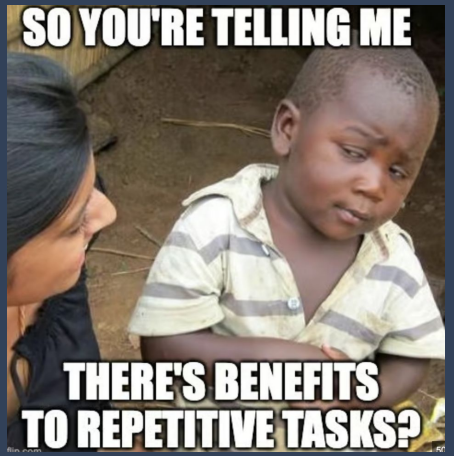
Problem Statement: Amazon Customer Support Overload

Real-World Challenge:

- Companies like Amazon receives thousands of support queries daily through multiple channels (email, chat, Slack, helpdesk).
- Many queries are repetitive, including:
  - Password resets
  - Troubleshooting common issues
  - Product feature explanations
- Current Issues:
  - Delayed responses due to manual handling.
  - Reduced customer satisfaction.
  - Increased operational costs.

Current Query Handling Process

- Information Sources:
  - Confluence: Comprehensive internal documentation. (Help docs 50 pages)
  - Slack: Active internal discussions and historical queries.
- Challenges:
  - Many repetitive questions are answered manually.
  - Support teams often need to sift through Slack for previous answers.
  - This leads to time inefficiencies and bottlenecks.



Coworker: This is the first time I've seen you smile at work, what's up?
Me: I'm about to quit

AI-Powered Solution

- Proposed Solution: AI chatbot to automate query handling.
- Core Features:
  - Leverage Gen AI (e.g., ChatGPT) to process and respond to customer queries.
  - Automate responses by referencing internal documentation and Slack conversations.
  - Intelligent routing for unresolved issues to human agents.
- Metrics
  - Direct metrics
    - Survey
    - like unlike button
  - In-direct metrics
    - How many resolved with in-person support interaction

AI Chatbot Capabilities

How It Works:

- Phase 1: Accesses all internal documentation to become a subject expert.
- Phase 2: Analyzes historical Slack conversations for context and previous solutions.
- LLM: Generates responses based on documentation and Slack info, supplemented with Retrieval-Augmented Generation (RAG) for more accurate answers.
- {In this presentation we shall see Phase 1 in detail.}

LLM and Google Colab demo

Demo-Chatbot:
https://colab.research.google.com/drive/1VKsKSkgaFeG3c0Io04JntT8rDGTK-3Gi?usp=sharing
Demo-chatbot2:
https://colab.research.google.com/drive/1P9VJBPMMUNdzkj85NQWj-Y2TR-N9IyEU?usp=sharing

High-Level Architecture

System Architecture:

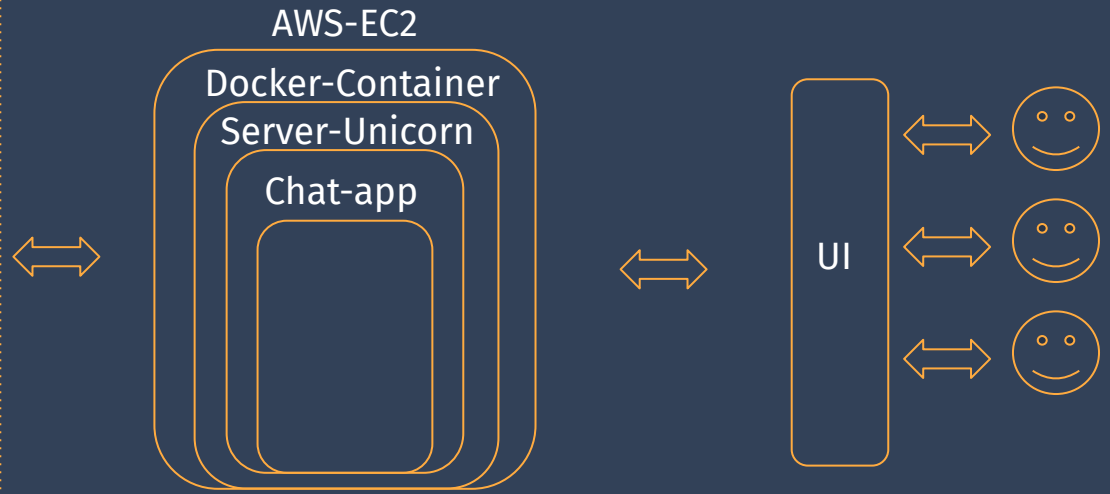
- Model: Hosted on EC2 or SageMaker.
- Microservice: Middle layer that handles chatbot queries, integrates with Slack, and routes responses.
- UI: Customer-facing chatbot interface for seamless interactions.

Full System Architecture

Open API
Groq
(Pay per token)

Deploy model in SageMaker and use

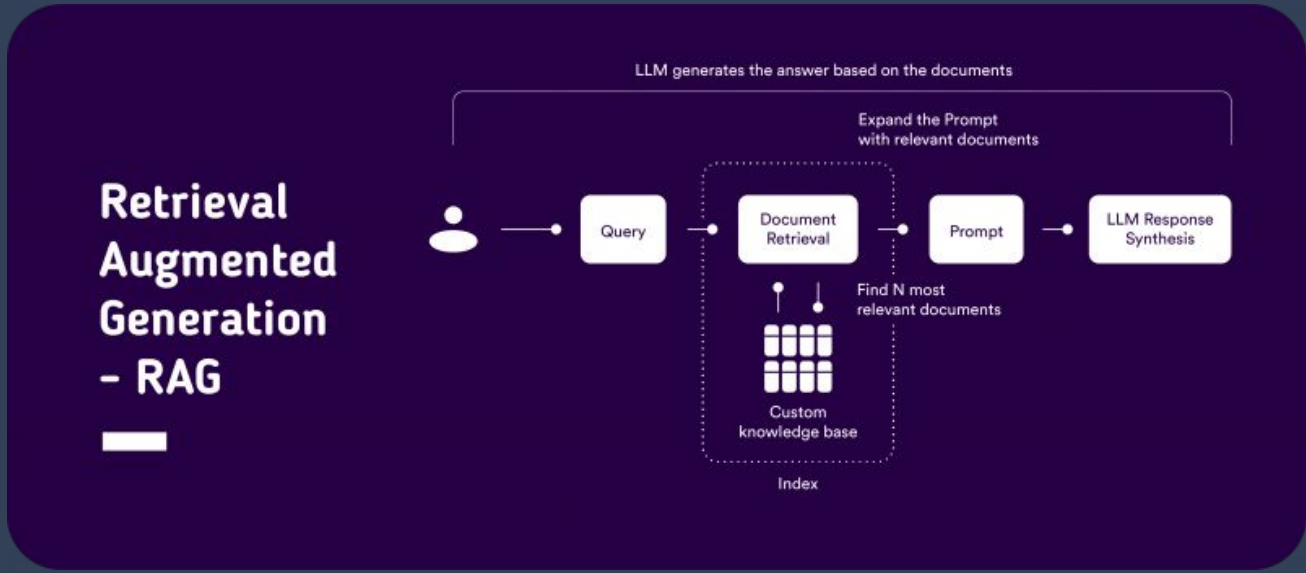
Deploy private model in EC2 and create end-point



RAG

Goal: Get relevant information

Idea: Vector DB, Keyword tags

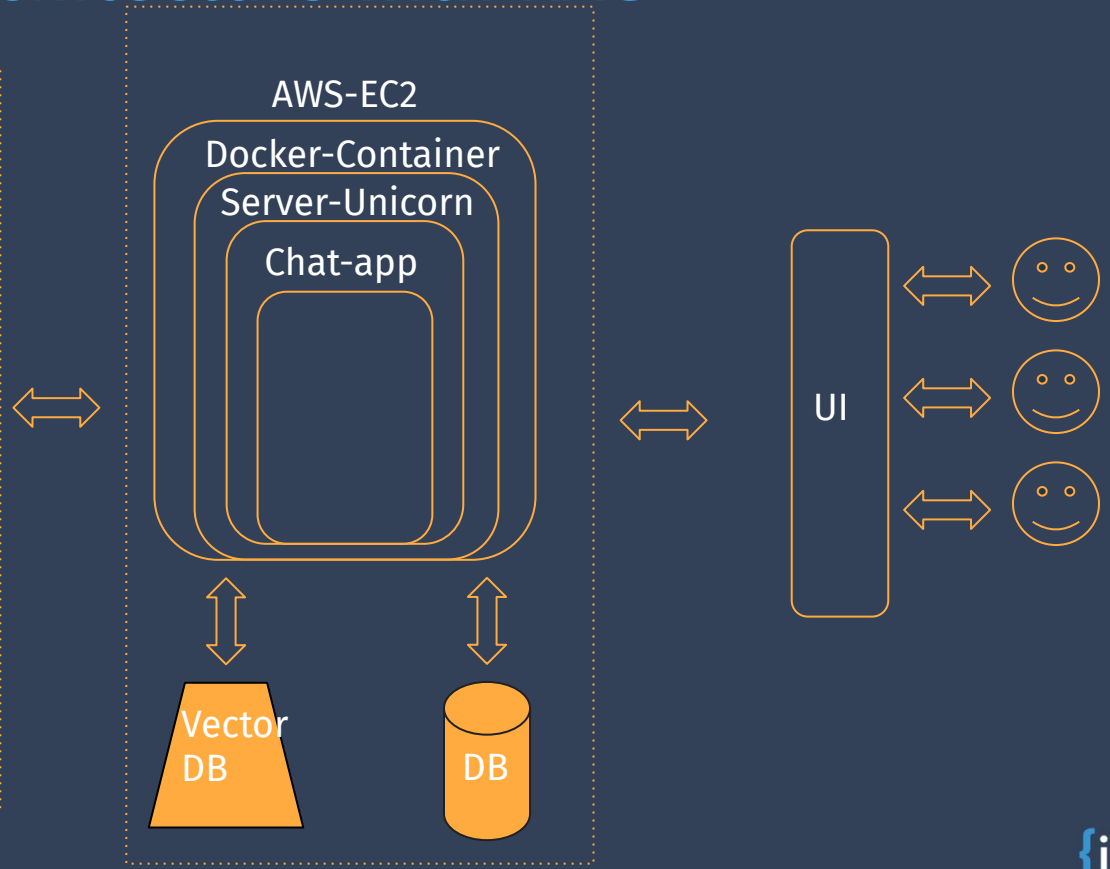


Full System Architecture with RAG

Open API
Groq
(Pay per token)

Deploy model in SageMaker and use

Deploy private model in EC2 and create end-point



Expected Outcomes

Impact on Operations:

- 80% of repetitive customer queries handled by the AI chatbot.
- Only 20% redirected to human agents for further assistance.
- Quicker response times and improved customer satisfaction.
- Reduced operational costs by scaling customer support without increasing staff.

Role wise Responsibilities

- **Product Manager**: Defines the product vision, features, and roadmap, ensuring the chatbot aligns with business goals and user needs.
- **TPM (Technical Program Manager)**: Coordinates cross-functional teams, tracks progress, and ensures timely delivery of technical components.
- **Engineering Manager**: Oversees the engineering team, ensuring technical execution, code quality, and alignment with product goals.
- **ML Engineer**: Develops and optimizes machine learning models to enhance chatbot performance and natural language understanding.
- **ML Ops**: Manages the deployment, monitoring, and lifecycle of ML models in production, ensuring stability and scalability.
- **LLM Ops**: Focuses on fine-tuning, monitoring, and maintaining large language models specifically, ensuring their efficiency and effectiveness.
- **DevOps**: Automates and streamlines infrastructure deployment, monitoring, and CI/CD pipelines for the chatbot system.
- **Software Engineers**: Build and integrate the chatbot's backend, APIs, and front-end components, ensuring seamless interaction with users.
- **UX Designer**: Designs the chatbot's conversational flow and user experience, focusing on accessibility, usability, and engagement

Feedback and Considerations

Key Considerations:

- Retrieval-Augmented Generation (RAG) for improved response accuracy.
- Build vs Buy: Should we build a custom solution or leverage existing tools?
- Metrics and Load: Estimate query load, model performance, and response accuracy.

Summary

01 AI-powered chatbots can automate up to 80% of customer queries in large SaaS companies.

02 Reductions in response times and operational costs.

03 Opportunity to further enhance support systems with AI integrations.