Harnessing the Power of Generative AI for Intelligent Applications

-Shivali Naik
Data Engineering | Cloud | AI

GenAI Chatbots: Unlocking Unstructured Data

-Shivali Naik
Data Engineering | Cloud | AI

Strategy GenAI Chatbots: A Journey from Prototype to Launch!

-Shivali Naik
Data Engineering | Cloud | AI

Agenda

- Artificial Intelligence Introduction
- AI Revolution & Unstructured Data
- Tech Stack for AI Chatbots
- Future of AI

Artificial Intelligence

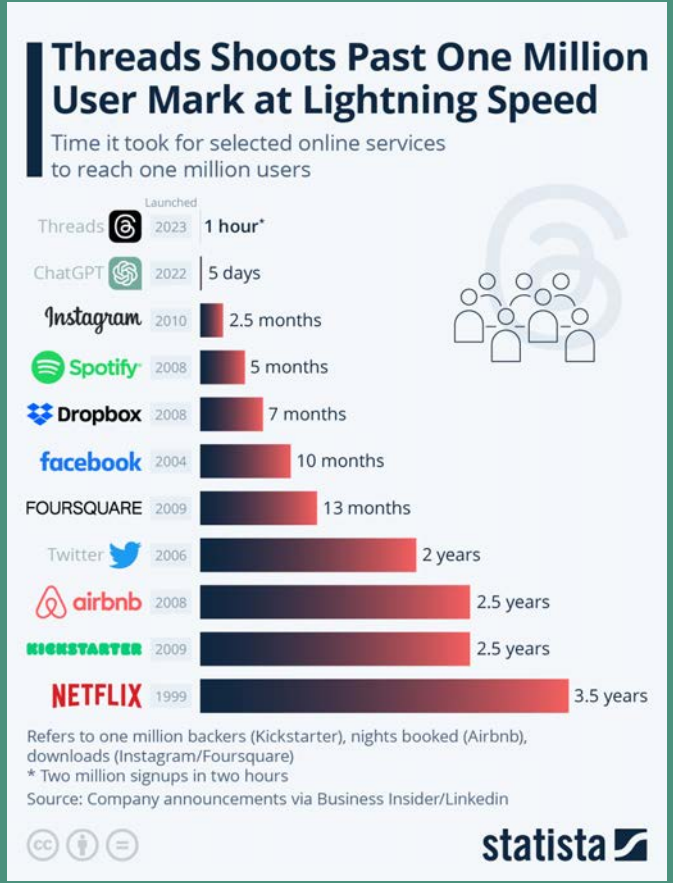
Artificial intelligence (AI), the ability of a machine to perform tasks commonly associated with intelligent beings

Artificial Intelligence Isn’t New

And then came ChatGPT...

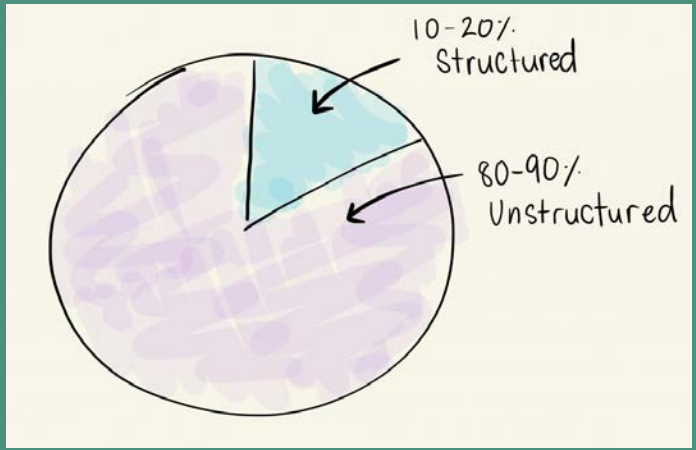
Adoption Chat GPT

OpenAI registered this breakthrough only five days after launching its chatbot. At the outset of 2023, the record ChatGPT set for fastest-growing base was unbeatable.



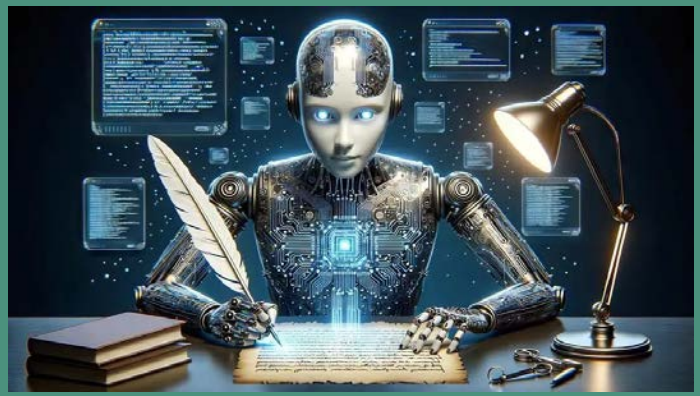
AI Revolution & Unstructured Data

- 90% of the world's data is unstructured (text, images, audio, video)
- **Challenge**: Traditional databases struggle to process it
- **Solution**: Generative AI extracts meaning & creates value



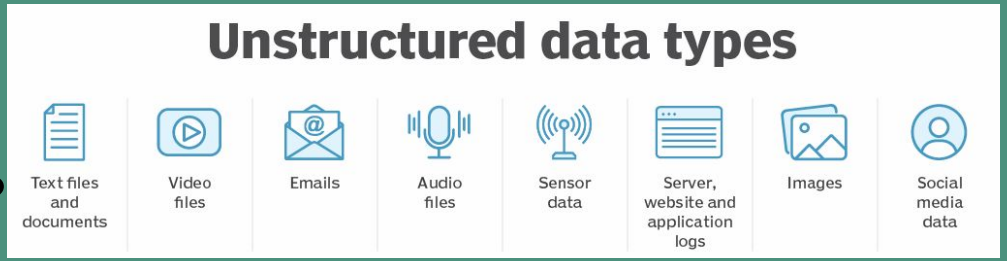
What is Generative AI?

- AI models that generate text, images, or insights
- Examples: GPT-4, Llama, Claude
- Why it’s powerful? Human-like responses, summarization, and analysis



Understanding Unstructured Data

- Types: Emails, PDFs, chat logs, Clinical notes, EHR records, pathology reports
- Why is it hard to process? No predefined schema or structure



How GenAI Processes Unstructured Data

- **Key Capabilities:**
  - Natural Language Processing (NLP) for text understanding.
  - Summarization and document classification.
  - Conversational AI for chatbots.
  - RAG - Vector - Chucks
- **Impact:**
  - Reduces administrative burden.
  - Enhances data-driven decision-making.

The Role of GenAI in Chatbots

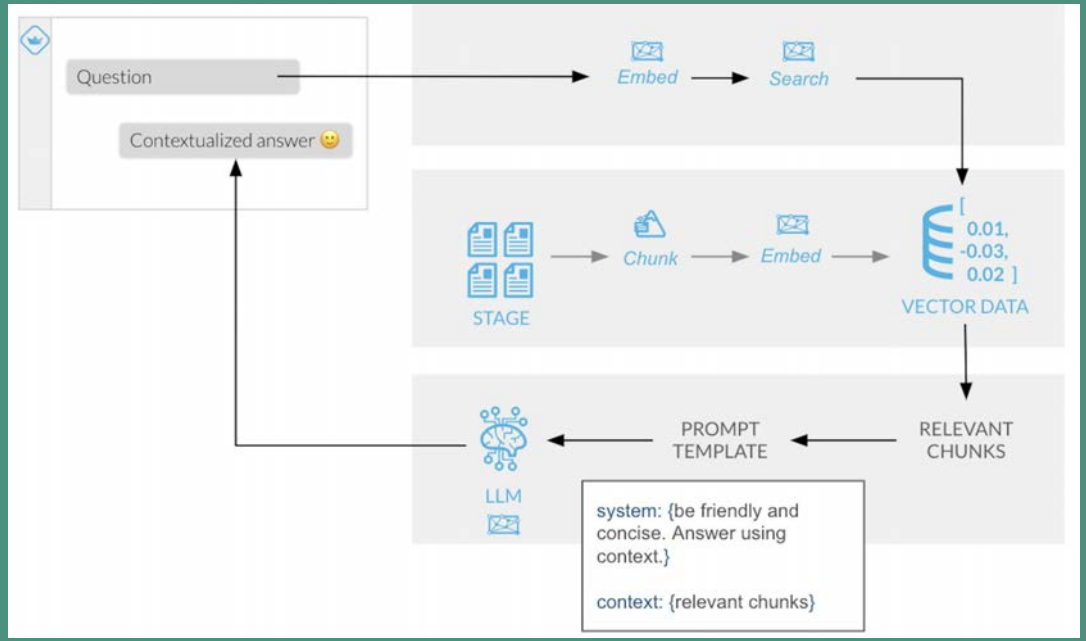
Traditional vs. AI-Powered Chatbots

Feature	Rule-Based Chatbots	AI-Powered Chatbots
Responses	Predefined	Context-aware, adaptive
Learning	No learning	Self-improving via data
Natural Language	Limited	Highly conversational

The Tech Stack for AI Chatbots

🛠️ **Key components**:

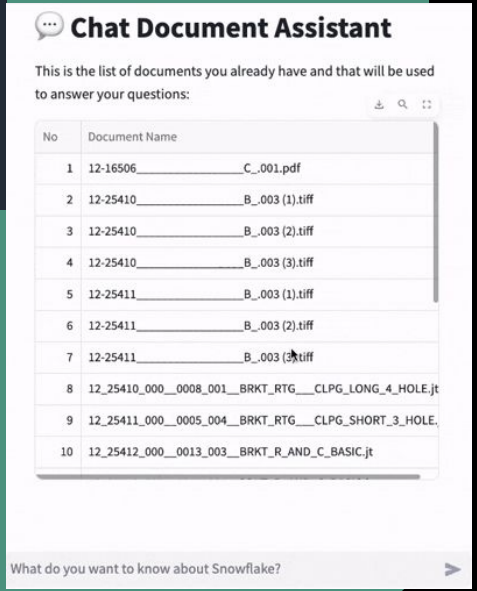
- **Large Language Models (LLMs)**: GPT, BERT, Claude
- **Embedding Databases**: Pinecone, FAISS, Weaviate
- **Vector Search for RAG**: Snowflake Cortex, OpenSearch
- **Frameworks**: LangChain, LlamaIndex, Streamlit



Script and Chatbot

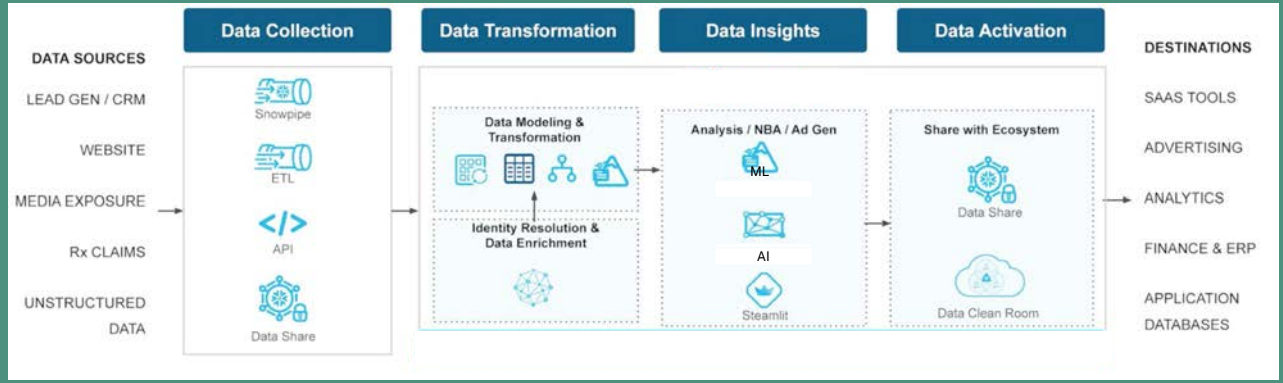
- def get_similar_chunks (question):Given a question, this function is going to calculate its embeddings and look for the most similar chunks within the table. Will return those chunks of text with the highest similarity.
- def get_chat_history():This function is going to return the previous conversation in the chat up to a limit defined by the global variable slide_window.
- def summarize_question_with_history():Takes the chat_history that we have got previously and the new question being asked. Remember that we are using the information from the PDF documents to answer questions. Therefore, we need to identify using vectors and cosine_similarity what chunk of text will be relevant to answer the question.
- def complete():Function where we have created a specific prompt to provide a chat history and the new question in order to get the query that will be used to find the right context.
- def create_prompt (myquestion):Builds a detailed prompt for the Snowflake Cortex LLM, incorporating chat history and document context to generate more accurate, context-aware responses and call complete() function to get the answer that is also printed.

```
import streamlit as st # Import python packages
from snowflake.snowpark.context import get_active_session
session = get_active_session() # Get the current credentials

import pandas as pd

pd.set_option("max_colwidth",None)

### Default Values
#model_name = 'mistral-7b' #Default but we allow user to select one
num_chunks = 3 # Num-chunks provided as context. Play with this to check how it affects
slide_window = 7 # how many last conversations to remember. This is the slide window.
#debug = 1 #Set this to 1 if you want to see what is the text created as summary and ser
#use_chat_history = 0 #Use the chat history by default

### Functions

def main():

    st.title(f":speech_balloon: Chat Document Assistant with Snowflake Cortex")
    st.write("This is the list of documents you already have and that will be used to an
    docs_available = session.sql("ls @docs").collect()
    list_docs = []
    for doc in docs_available:
        list_docs.append(doc["name"])
    st.dataframe(list_docs)

    config_options()
    init_messages()

    # Display chat messages from history on app rerun
    for message in st.session_state.messages:
        with st.chat_message(message["role"]):
            st.markdown(message["content"])

    # Accept user input
    if question := st.chat_input("What do you want to know about your products?"):
        # Add user message to chat history
        st.session_state.messages.append({"role": "user", "content": question})
        # Display user message in chat message container
        with st.chat_message("user"):
            st.markdown(question)
        # Display assistant response in chat message container
        with st.chat_message("assistant"):
            message_placeholder = st.empty()

            question = question.replace("'","")

            with st.spinner(f"{st.session_state.model_name} thinking..."):
                response = complete(question)
                res_text = response[0].RESPONSE

                res_text = res_text.replace("'", "")
                message_placeholder.markdown(res_text)
```



Building a Chatbot with GenAI – Step-by-Step

1. **Data Ingestion** (Collect emails, PDFs, transcripts)
2. **Data Preprocessing** (OCR, embeddings, cleaning)
3. **Model Selection** (LLM + RAG for knowledge retrieval)
4. **Fine-Tuning & Testing** (Improve accuracy, reduce hallucination)
5. **Deployment & Monitoring** (APIs, observability tools)



Case Studies & Industry Applications

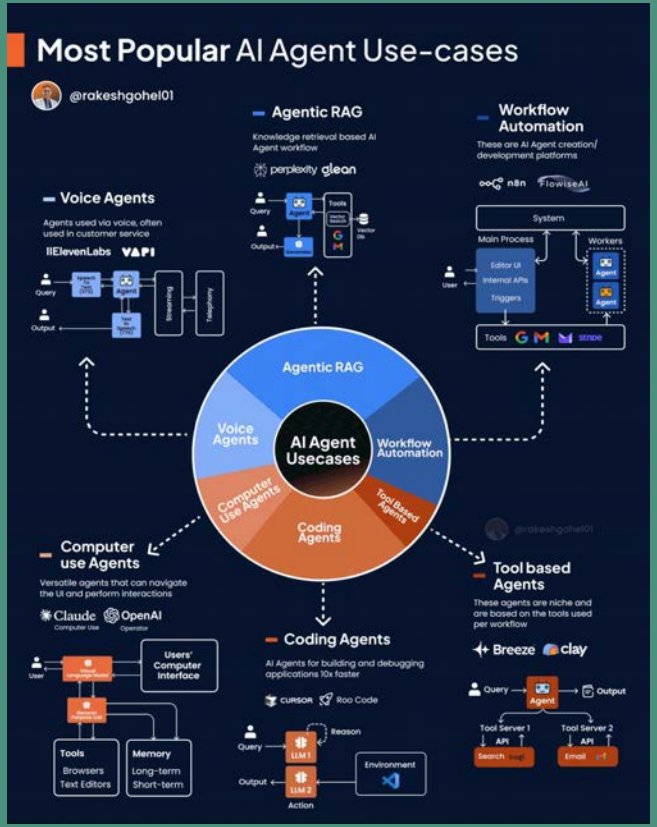
Use Cases:

✅ **Customer Support** (Chatbots for FAQs & troubleshooting)

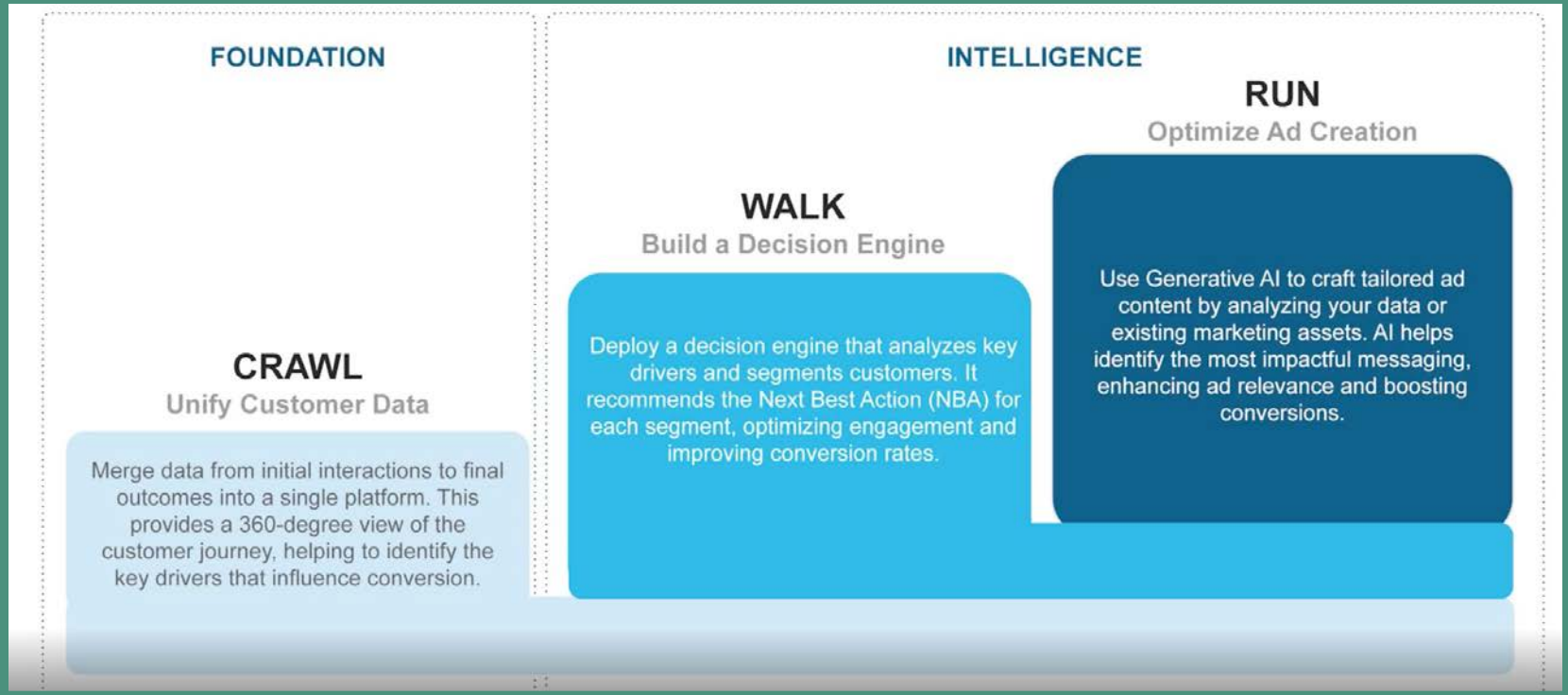
✅ **Healthcare** (Patient Q&A, medical research summarization, Predictions, Marketing)

✅ **Legal & Compliance** (Contract analysis, policy bots)

✅ **Finance** (Automated investment insights, fraud detection)



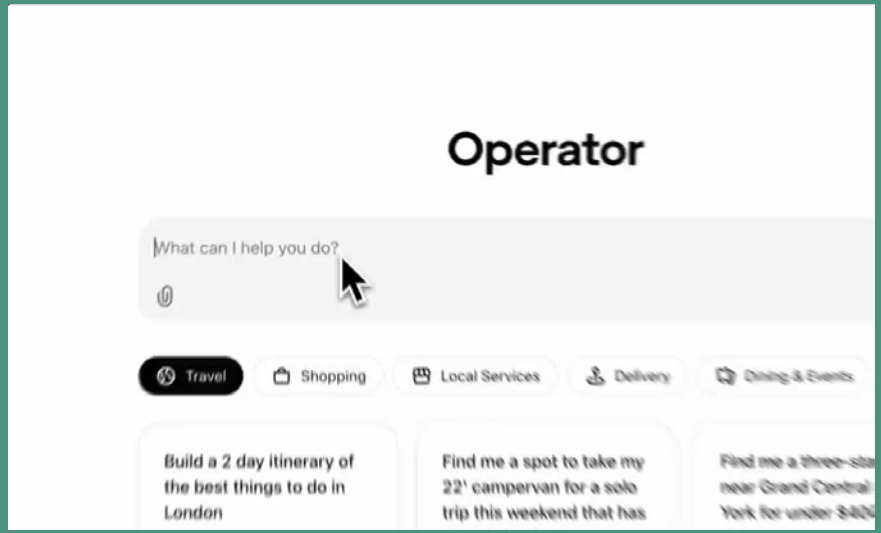
Next Steps to Adopt Gen AI



The Future of AI Chatbots

What's next?

- **Multimodal Chatbots** (Voice + Text + Vision)
- **Autonomous AI Agents** (Self-learning & decision-making)
- **AI-Powered Workflows** (Replacing manual processes)
- **Regulatory Compliance** (AI governance & safety measures)



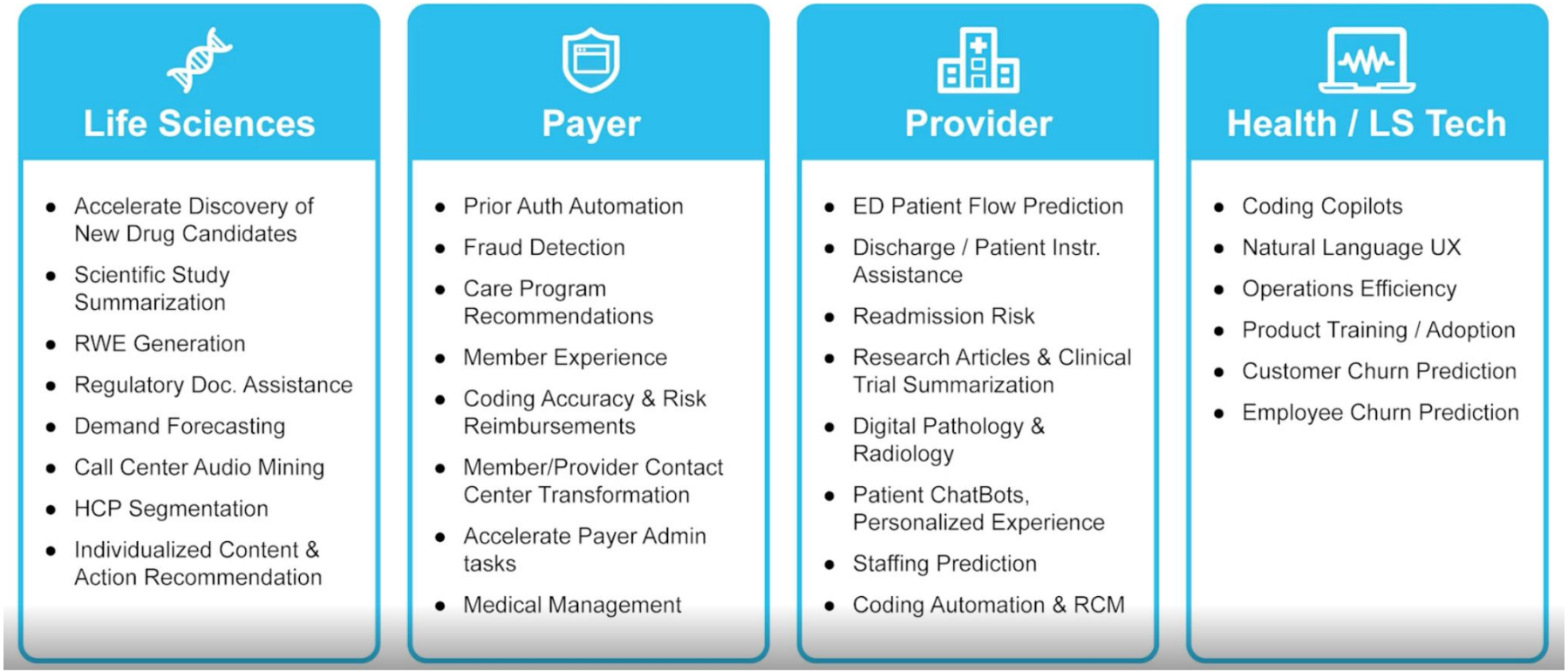
What will the future be?

Wall-E, humans had nothing to do and grew weaker over time

Discovery

Thankyou

Business Transformation: AI Use Cases



Demo