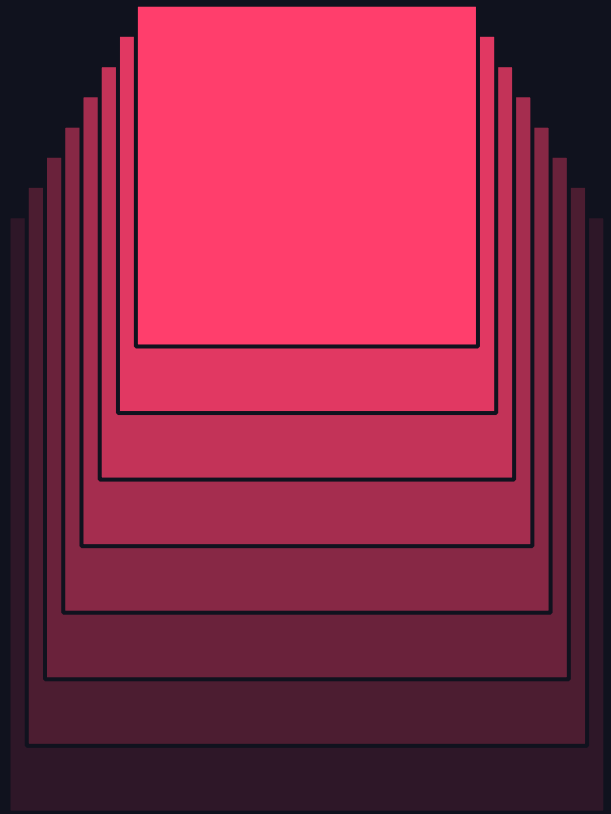
DATA+AI SUMMIT
BY databricks

RAPID LLM PROTOTYPING

W/ OPENAI, DATABRICKS, AND STREAMLIT



Alexandra Diem
13th June 2024

©2024 Databricks Inc. — All rights reserved

Classified: General Business

HI, I'M ALEX

- PhD Applied Math in Medicine
- 4 years in academia
- 2 years in consulting, both software engineering and data science
- Head of Cloud Analytics & MLOps at Gjensidige

- 36 yo born & raised in Germany, lived in Australia, UK, South Africa, USA, but like Norway best
- 2 cats
- Spend most of my spare time on a bike and on skis

Classified: General Business

©2024 Databricks Inc. — All rights reserved

GJENSIDIGE IS A LEADING GENERAL INSURER IN THE NORDIC MARKETS

Leading position

- Strong brand built over 200 years
- #1 in Norway (26% market share)
- 2 million customers
- Very high loyalty

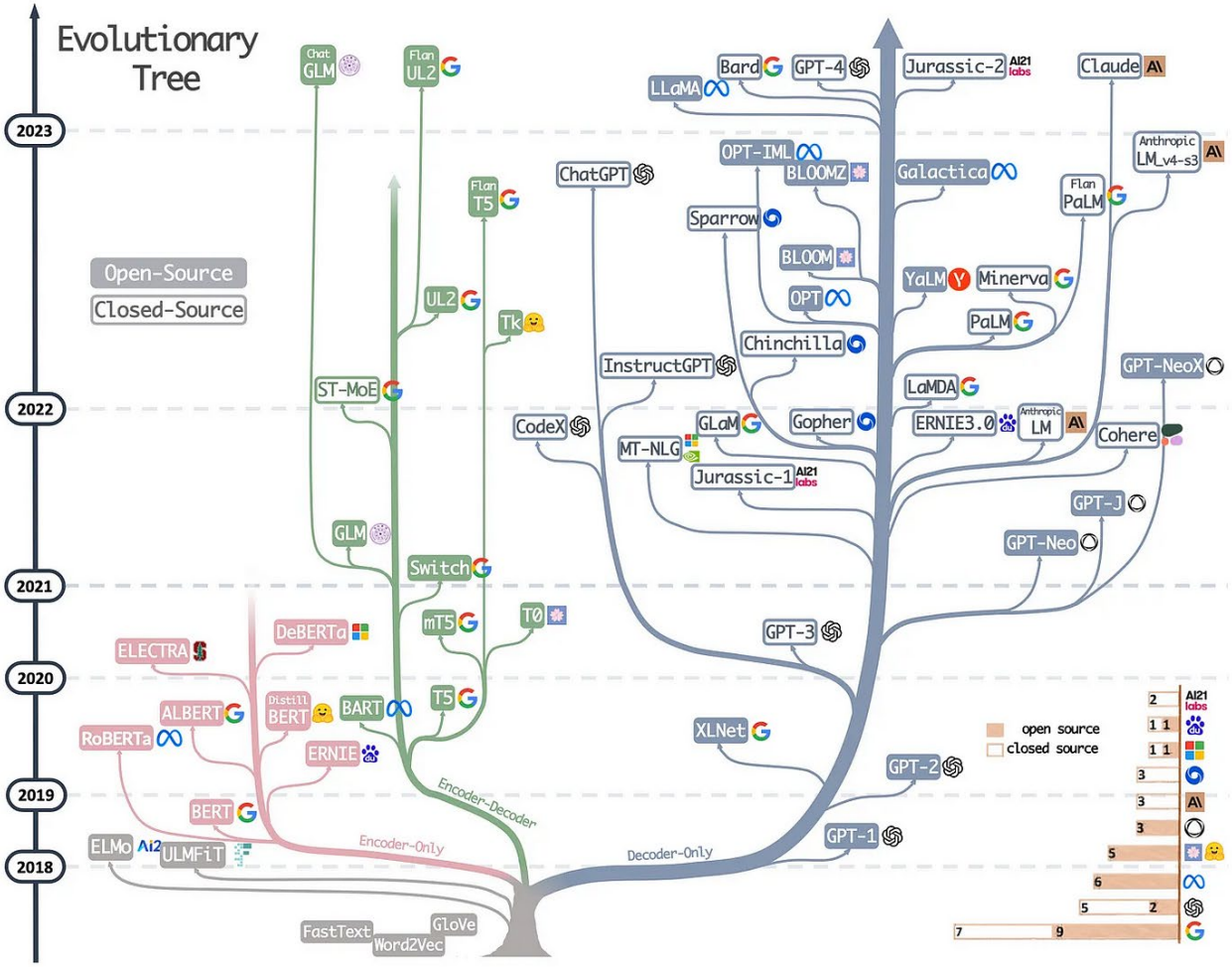
Efficient operation

- Superior customer experience
- Profitability before growth
- Analytical approach from A to Z
- Target cost ratio 13%

Classified: General Business
©2024 Databricks Inc. — All rights reserved

RAPID PROTOTYPING USING THE LEAN STARTUP METHOD

©2024 Databricks Inc. — All rights reserved



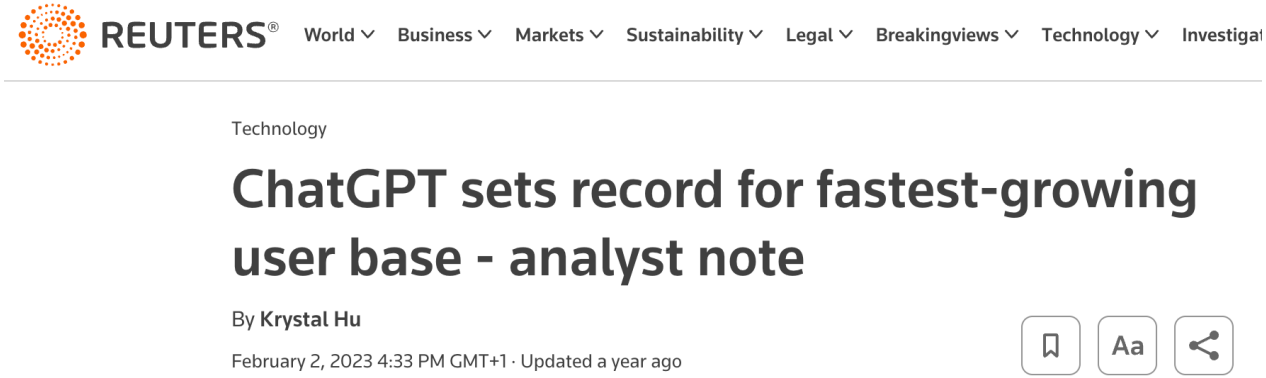
Large Language models have been with us for quite some time

Classified: General Business
©2024 Databricks Inc. — All rights reserved

NOVEMBER 2022

Classified: General Business
©2024 Databricks Inc. — All rights reserved


ChatGPT
NOVEMBER 2022



REUTERS® World Business Markets Sustainability Legal Breakingviews Technology Investigat

Technology

ChatGPT sets record for fastest-growing user base - analyst note

By **Krystal Hu**

February 2, 2023 4:33 PM GMT+1 · Updated a year ago

Aa

Feb 1 (Reuters) - ChatGPT, the popular chatbot from OpenAI, is estimated to have reached 100 million monthly active users in January, just two months after launch, making it the fastest-growing consumer application in history, according to a UBS study on Wednesday.

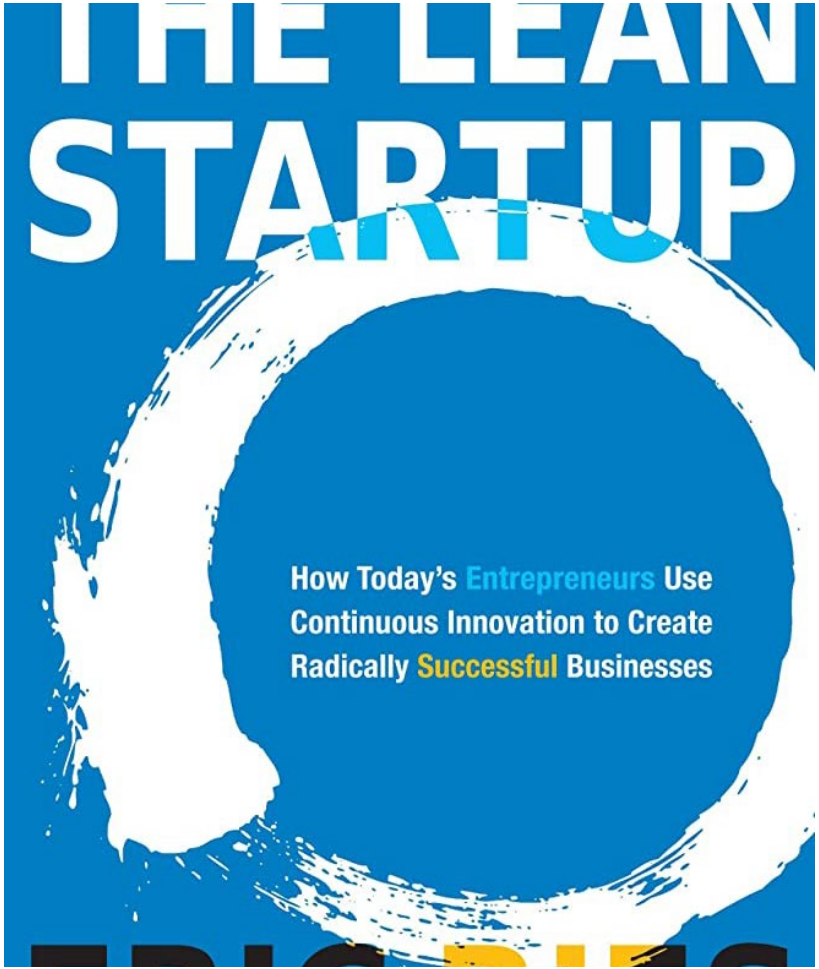
The report, citing data from analytics firm Similarweb, said an average of about 13 million unique visitors had used ChatGPT per day in January, more than double the levels of December.

Classified: General Business

©2024 Databricks Inc. — All rights reserved

IF YOU WANT PEOPLE TO WANT AI, IT HAS TO SOLVE REAL PROBLEMS. FAST.

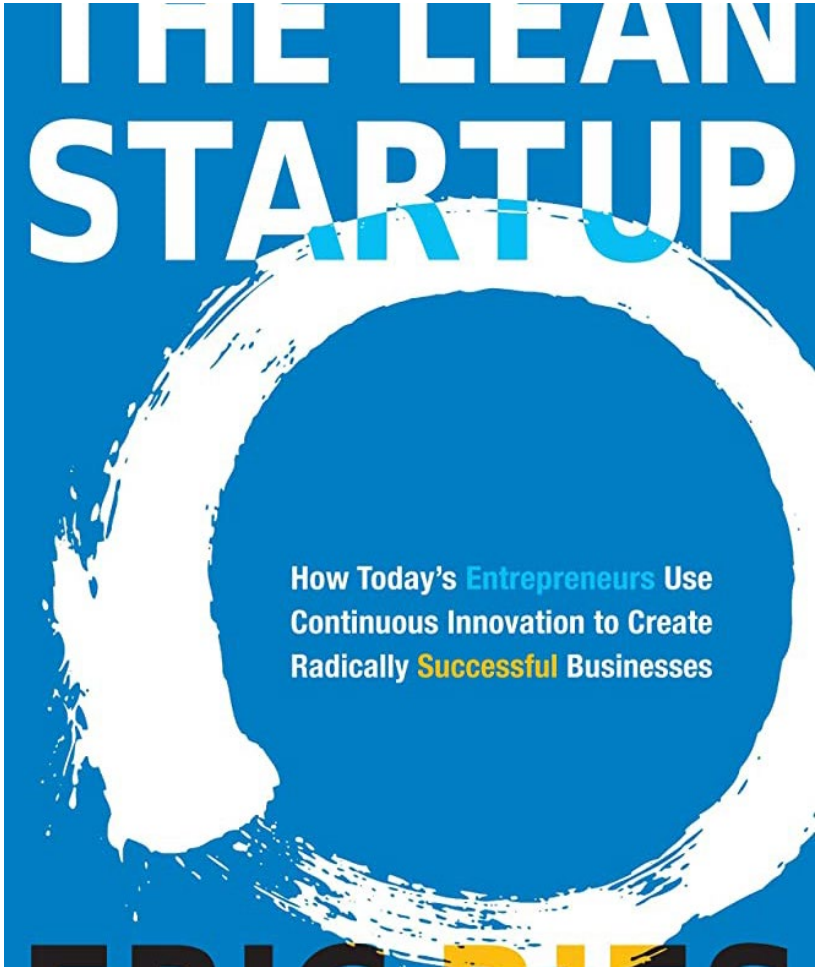
©2024 Databricks Inc. — All rights reserved



«A startup is a human institution designed to create a new product or service under conditions of extreme uncertainty.»

Note: A startup may very well be a team or product in a large, established organisation!

Classified: General Business
©2024 Databricks Inc. — All rights reserved



90% OF STARTUPS FAIL. WHY?

«A startup is a human institution designed to create a new product or service under conditions of extreme uncertainty.»

«Extreme uncertainty» means that the startup cannot know what its product or customers should be. Classical business analysis creates a false sense of certainty

©2024 Databricks Inc. — All rights reserved

Classified: General Business

THE LEAN STARTUP METHOD

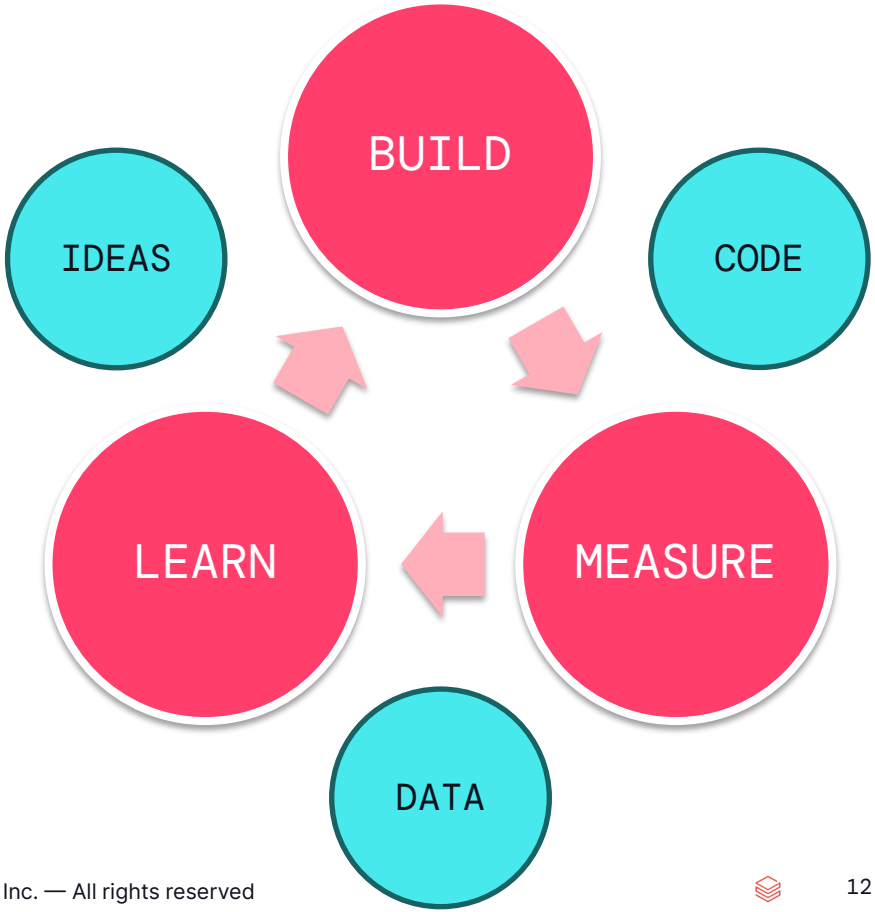
Build. Measure. Learn.

Ideas. Develop a falsifiable hypothesis that results in *validated learning*. Define the *Minimum Viable Product* (MVP) that will test your hypothesis.

Code. Implement and deploy the simplest possible realisation of your MVP.

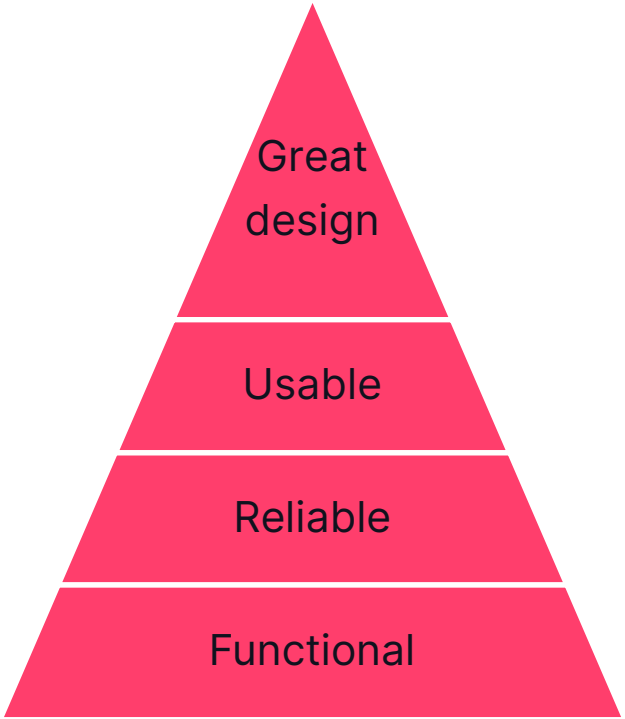
Data. Collect user feedback asap. Aim for recruiting *early adopters*.

Minimise time through the loop!



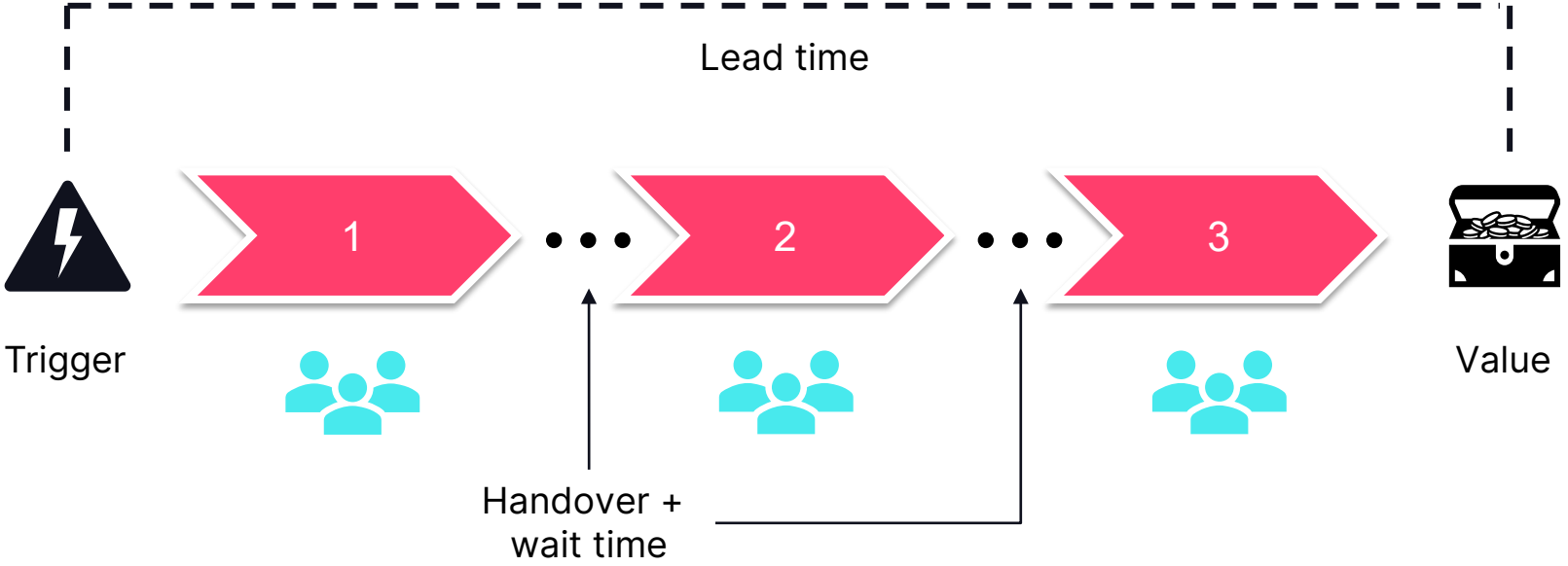
Classified: General Business
©2024 Databricks Inc. — All rights reserved

CLASSIC PRODUCT DEVELOPMENT



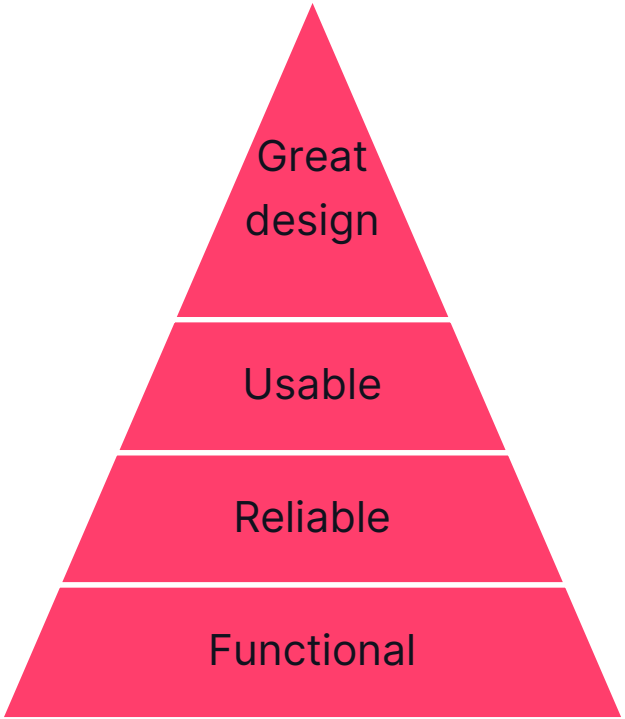
Classified: General Business
©2024 Databricks Inc. — All rights reserved

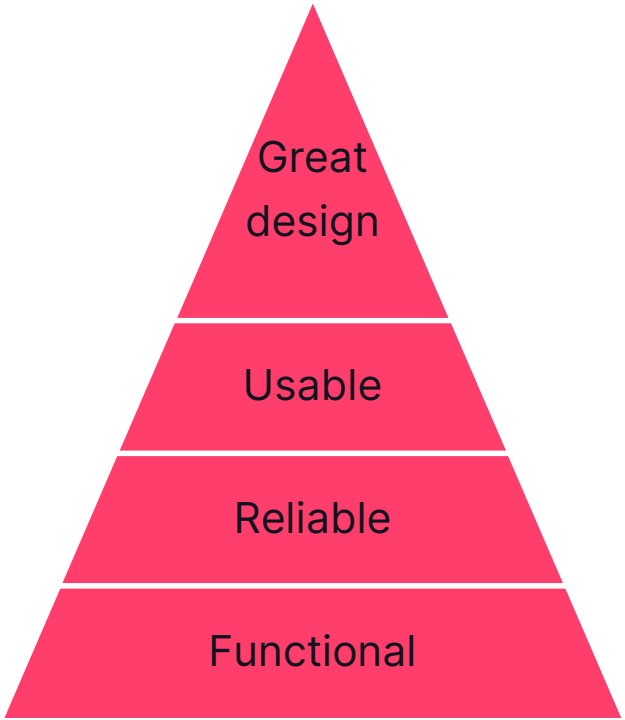
VALUE STREAM ANALYSIS: WHAT DOES IT TAKE?



Classified: General Business
©2024 Databricks Inc. — All rights reserved

MVP DEVELOPMENT



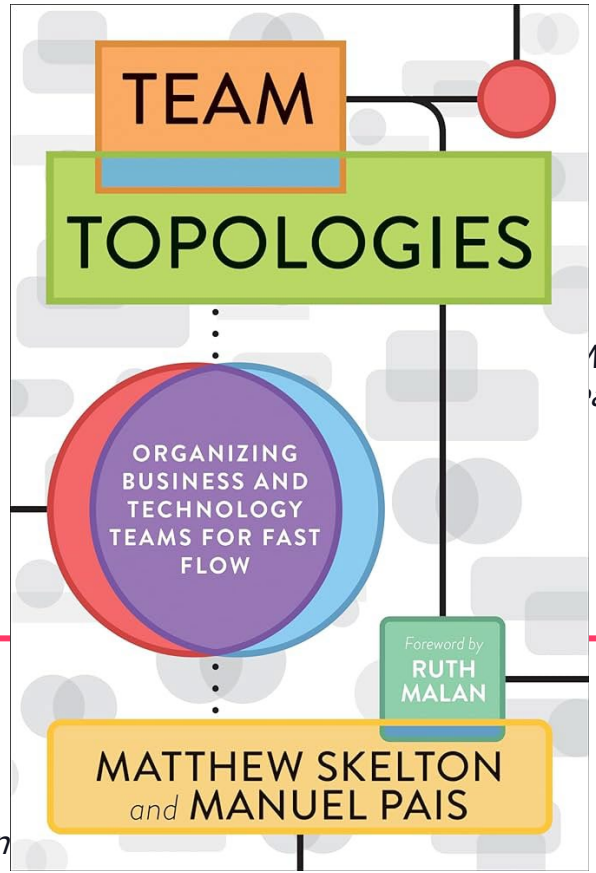


Classified: General Business
©2024 Databricks Inc. — All rights reserved

MVP ENABLEMENT

Identify

Identify use cases within the organisation and prioritise following set criteria



Pull out

We work to make ourselves redundant and ave maintenance and further development to the analyst teams

Stan

Classified: General Business
©2024 Databricks Inc. — All rights reserved

STREAMLIT AS A FRONTEND FOR LAKEHOUSE DATA

©2024 Databricks Inc. — All rights reserved

WHY STREAMLIT?

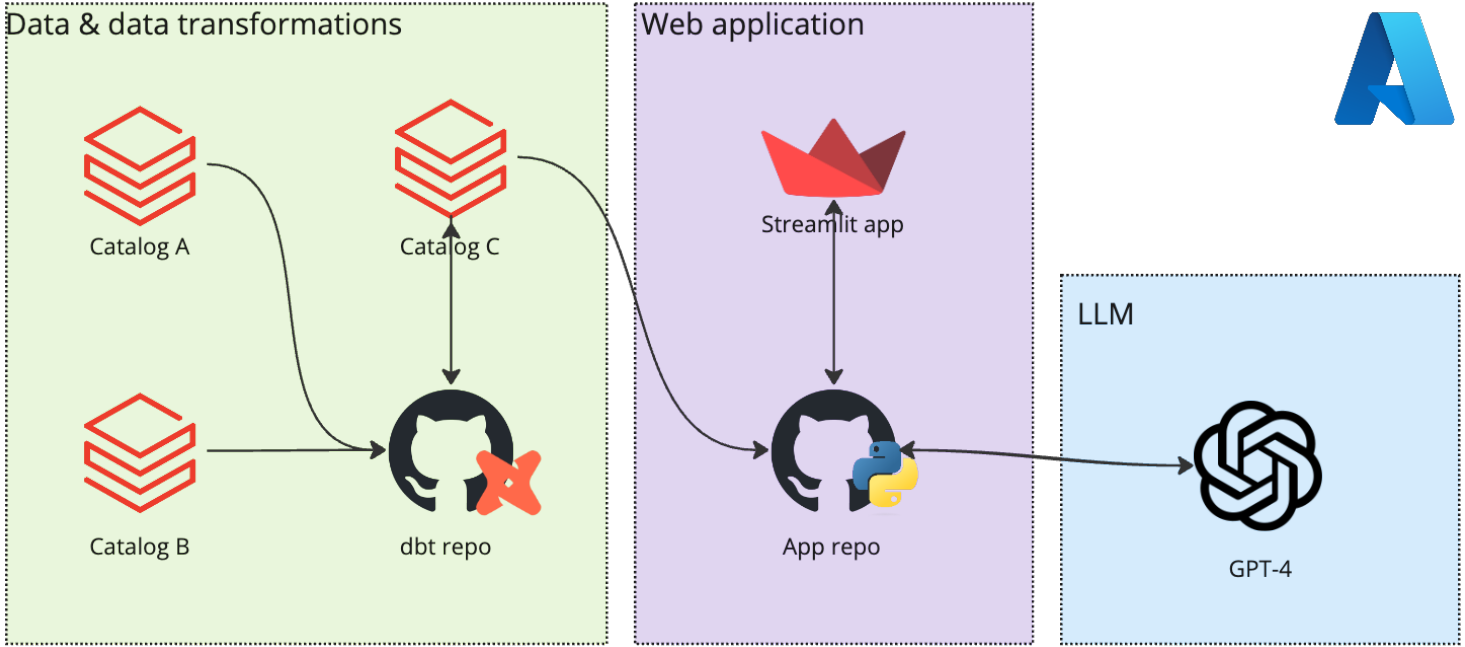
- **Written in Python:** All our data scientists work in Python on a daily basis, making it easy to learn
- **Ease of use:** Streamlit has a simple API with pre-built interactive data components, such that developers can focus on the data
- **Limited customisation:** Keeps focus on data-driven app development
- **Integration with Databricks:** Databricks SQL Connector for Python makes it easy to integrate data into Streamlit

```
def main():
    st.set_page_config(page_title="Eglev")
    sb = st.sidebar
    with sb:
        st.image(GJF_LOGO)

        st.image(EGLEV_AVATAR)

        st.markdown(introduction)

        authorized = auth_component()

        with st.expander("My scope"):
            st.markdown(scope)

        with st.expander("My limitations"):
            st.markdown(limitations)

        st.markdown(
            f"""
            <style>
                [data-testid="stSidebarNav"] + div {{
                    position:relative;
                    bottom: 0;
                    height:50%;
                    background-image: GJF_LOGO;
                    background-size: 85% auto;
                    background-repeat: no-repeat;
                    background-position-x: center;
                    background-position-y: bottom;
                }}
            </style>
            """,
```


Classified: General Business

©2024 Databricks Inc. — All rights reserved

ARCHITECTURE



Classified: General Business
©2024 Databricks Inc. — All rights reserved

COMPONENTS OF A STREAMLIT APP

app.py

PYTHON

```python
def main():
    st.set_page_config(page_title="Eglev")

    # Define layout of app components
    sb = st.sidebar
    with sb:
        st.image(GJF_LOGO)
        st.image(EGLEV_AVATAR)
        st.markdown(INTRODUCTION)

        # Check user authentication & authorisation
        authorized = auth_component()

        with st.expander("Scope"):
            st.markdown(SCOPE)
```

PYTHON

```python
from msal_streamlit_authentication import msal_authentication

# Check user authentication & authorisation
def auth_component():
    token = msal_authentication(...)

    if not token:
        st.write("Please log in to interact with the Eglev Chatbot")
        return False

    authorized = authorize(token)

    if token and not authorized:
        st.write("Please request access to the Eglev Chatbot")

    return authorized
```


Classified: General Business

©2024 Databricks Inc. — All rights reserved

COMPONENTS OF A STREAMLIT APP

app.py

PYTHON

```python
def main():
    ...
    if authorized:
        chat_component()


def chat_component():
    ...
    if question := st.chat_input("Type in your question..."):
        st.session_state.messages.append(...)

    with st.chat_message("assistant", avatar=EGLEV_AVATAR):
        with st.spinner(text="Working on it..."):
            # Orchestrate communication between Databricks and OpenAI
            response = answer_the_question(question)
            ...
```

PYTHON

```python
# Orchestrate communication between Databricks and OpenAI
def answer_the_question():
    response = OpenAIClient().completion(
        system_message=system_message_final_answer.format(question, query),
        message="What is the answer to the question?",
    )
    return response.choices[-1].message.content
```


Classified: General Business

©2024 Databricks Inc. — All rights reserved

COMPONENTS OF A STREAMLIT APP

databricks_client.py

PYTHON

```
class DatabricksSQLClient:
    connection: Connection
    max_retry_attempts = 3

    def __init__(self, hostname, http_path, access_token):
        ...
        self.connect()

    def connect():
        self.connection = sql.connect(hostname, ...)

    def retry(self):
        ...
        return True


    ...
```

PYTHON

```
def execute_query():
    try:
        with self.connection.cursor() as cursor:
            cursor.execute(query)
            self.reset_retry()
            if cursor.description:
                return cursor.fetchall()
            else:
                return None
    except Exception as e:
        logger.error("...")
        if self.retry():
            time.sleep(1)
            self.connect()
            self.execute_query()
```


Classified: General Business

©2024 Databricks Inc. — All rights reserved

COMPONENTS OF A STREAMLIT APP

openai_client.py

PYTHON

```python
from openai import AzureOpenAI

class OpenAIClient:

    def _connect(self, hostname, token):
        if not (hostname and token):
            raise EnvironmentError(...)
        self.client = AzureOpenAI(
            api_key=token,
            api_version=settings.openai_api_version,
            azure_endpoint=settings.openai_host,)
```

PYTHON

```python
def completion(self, system_message, message, examples, model):
    return self.client.chat.completions.create(
            model=model,
            messages=messages,
            temperature=0,
            max_tokens=800,
            frequency_penalty=0.16,
            presence_penalty=0.17,
            top_p=0.95,
            stop=None,
        )
```


Classified: General Business

©2024 Databricks Inc. — All rights reserved

DEVELOPING OUR AI ANALYST

©2024 Databricks Inc. — All rights reserved

INSURANCE IS A FUNDAMENTALLY DATA-DRIVEN BUSINESS

Which marketing strategy should we use?

Will the customer on the phone be profitable?

Our client just bought a house. Which advice should we provide?

Can we process this claim automatically?

Create & price insurance

Buy insurance

Have insurance

Use insurance

How should we price this product or service?

Should we contact this customer for cross-selling?

How much cash do we need for future claims?

Is this a legitimate claim?

Classified: General Business

©2024 Databricks Inc. — All rights reserved


Unstructured information (speech, text, images etc.)
Transaction data
External data sources
User data
Sensor data (IoT)
core
Unstructured information (speech, text, images etc.)
Transaction data
External data sources
User data
Sensor data (IoT)
core

Our analysts receive at least one request per week from the business related to data extraction.

~10,000
data extraction requests per year

~2.5
interactions per request after initial contact for clarifications

~5,000
total **work hours** for data extraction per year

VALUE STREAM ANALYSIS: WHAT DOES IT TAKE?

Business

Analyst / Data scientist

Business

1

2

3

Business needs to make a decision

Send message request for data

Runs SQL based on request

Use the data

Value

Classified: General Business

©2024 Databricks Inc. — All rights reserved

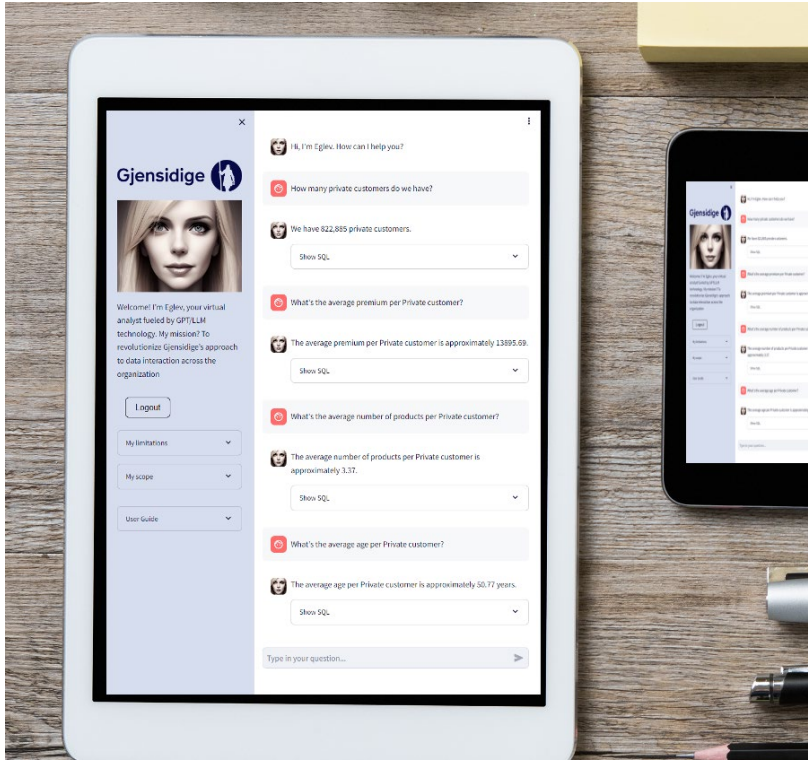
MEET EGLEV – OUR AI ANALYST

©2024 Da

80% of the requests can be answered with just two star schemas!

MVP SCOPE FOR EGLEV

Solving real problems fast.



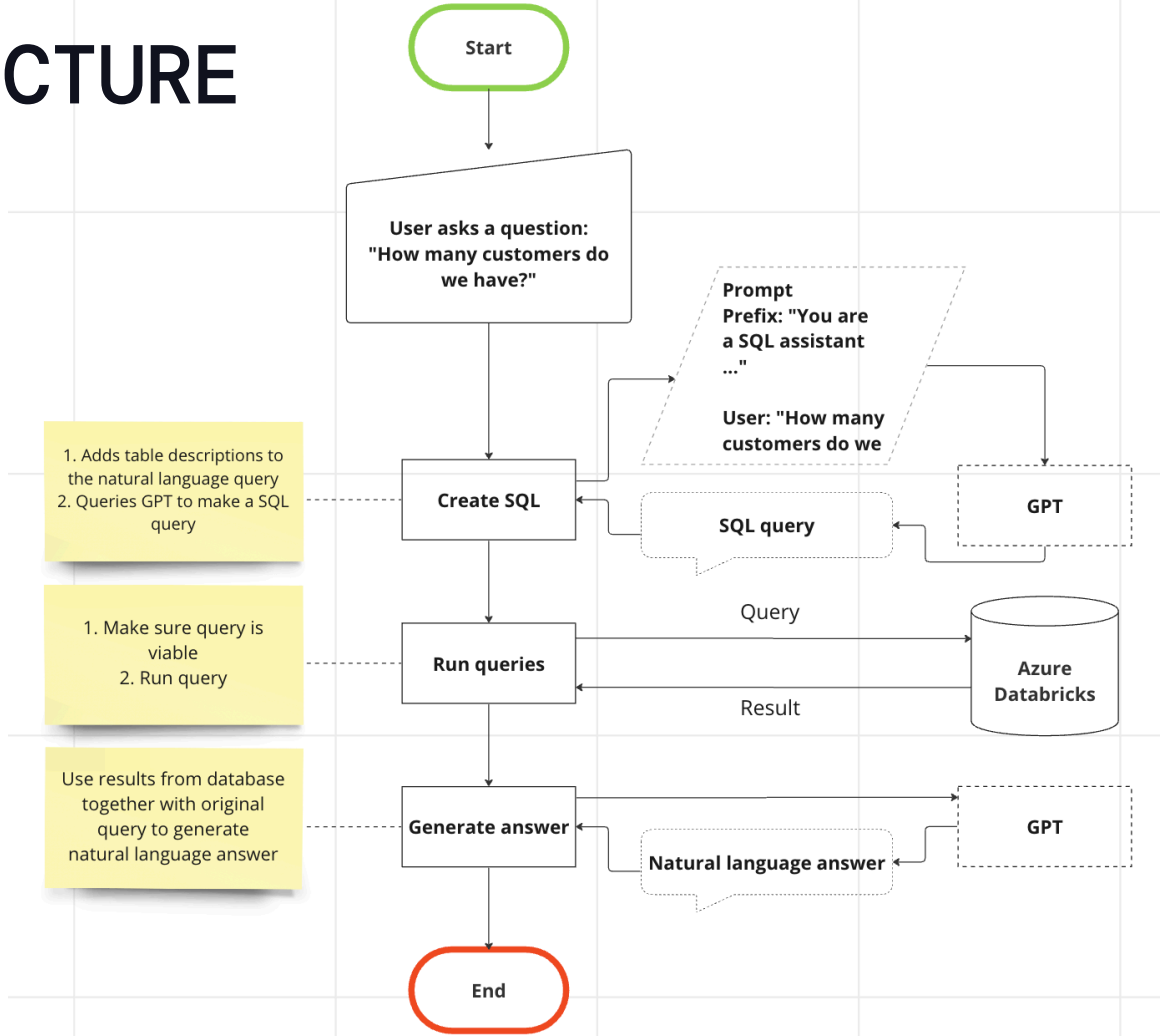
Objective: Demonstrate that LLMs can write correct SQL against our lakehouse data

Scope:

- Limit access to 2 star schemas
- Keep the user interface simple
- Recruit first adopter type test users
- Widen data access gradually and add functionality according to user feedback

Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE



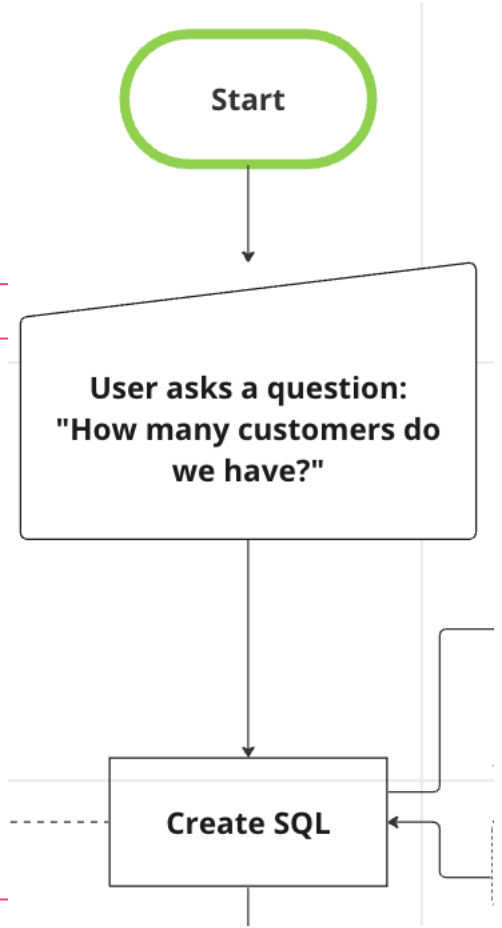
Classified: General Business

ARCHITECTURE

app.py

PYTHON

```
st.session_state.messages.append({"role": "user", "content": question})
st.session_state.generated_sql_dict.append(dummy_sql)
with st.chat_message("user"):
    st.markdown(question)

with st.chat_message("assistant", avatar=EGLEV_AVATAR):
    with st.spinner(text="Working on it..."):
        full_response, generated_sql_dict = answer_the_question(question)
        st.session_state.generated_sql_dict.append(
                generated_sql_dict
        )
        st.session_state.messages.append(
                {"role": "ai", "content": full_response}
        )
```



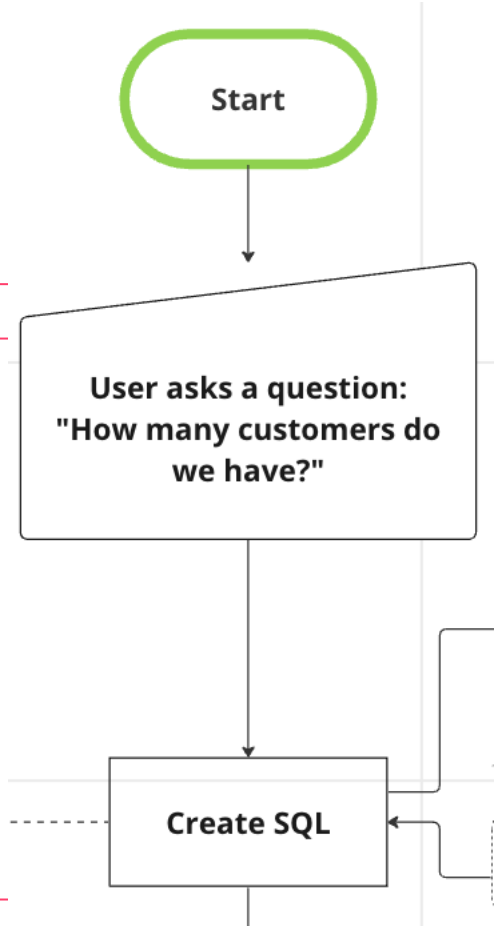
Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE

answer_the_question.py

PYTHON

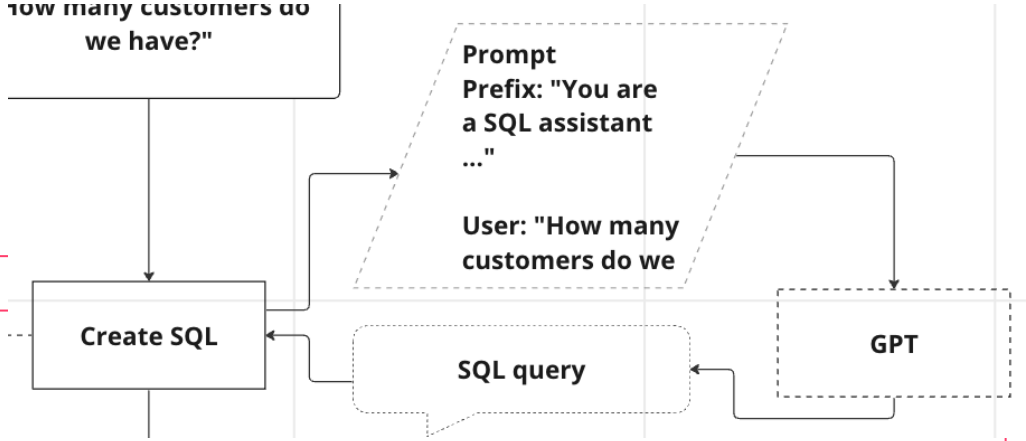
```
def answer_the_question(question):
    # STEP 1
    # This step takes the question from the user and creates a sql to get
    # the answer

    generated_sql = generate_sql(question)
    generated_sql.update({"question": question, "id": log_id})

    if "sql" not in generated_sql:
        generated_sql["sql"] = "N/A"

    if generated_sql["sql"] == "N/A":
        DatabricksSQLClient().store_log_message(generated_sql)
        return generated_sql["explanation"], generated_sql
```



Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE

generate_sql.py



PYTHON

```
def generate_sql(question, model:
    system_message = SystemMessage(question)

    response = OpenAIClient().completion(
            system_message=system_message.system_message, message=question,
            examples=system_message.examples, model=model)
    json_response = response.choices[-1].message.content

    try:
        response = json.loads(json_response)
        return response
    except json.JSONDecodeError:
        return {"explanation": json_response, "sql": "N/A"}
```

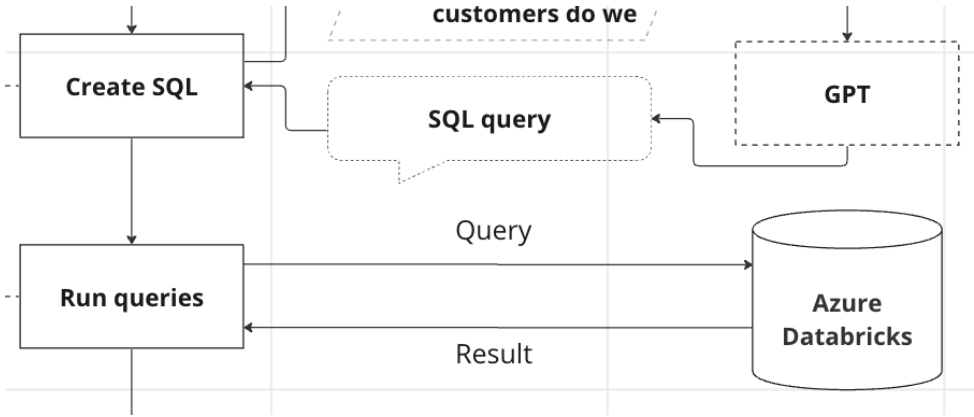

Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE

answer_the_question.py

PYTHON

```
def answer_the_question(question):
    # STEP 2
    # This step runs the sql-code from step 1 to
    # get numerical answer

    query_results =
DatabricksSQLClient().execute_query(generated_sql
["sql"])

    if query_results:
        query_results = limit_rows(query_results)
```



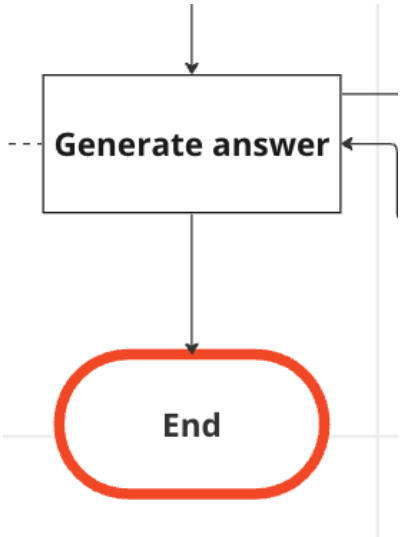
Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE

answer_the_question.py

PYTHON

```
def answer_the_question(question):
    # STEP 3
    # This step takes the sql-results along with the
    # question and generates a reply back to the user
    answer = generate_answer(question, query_results)

    generated_sql.update({"answer": answer})
    DatabricksSQLClient().store_log_message(generated_sql)

    return answer, generated_sql
```

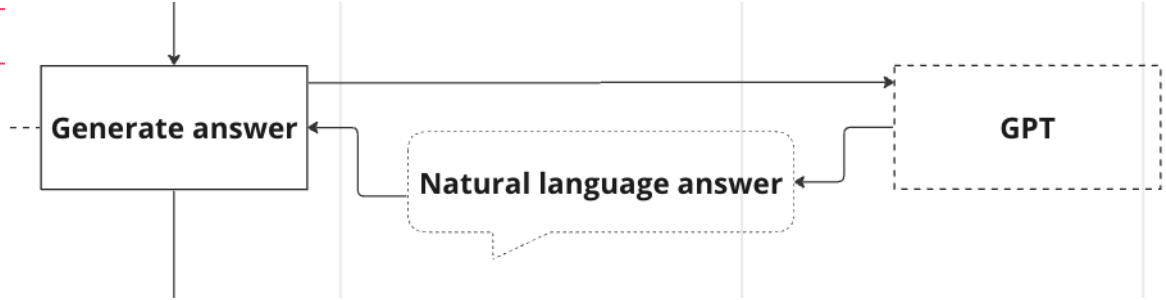


Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE

generate_answer.py

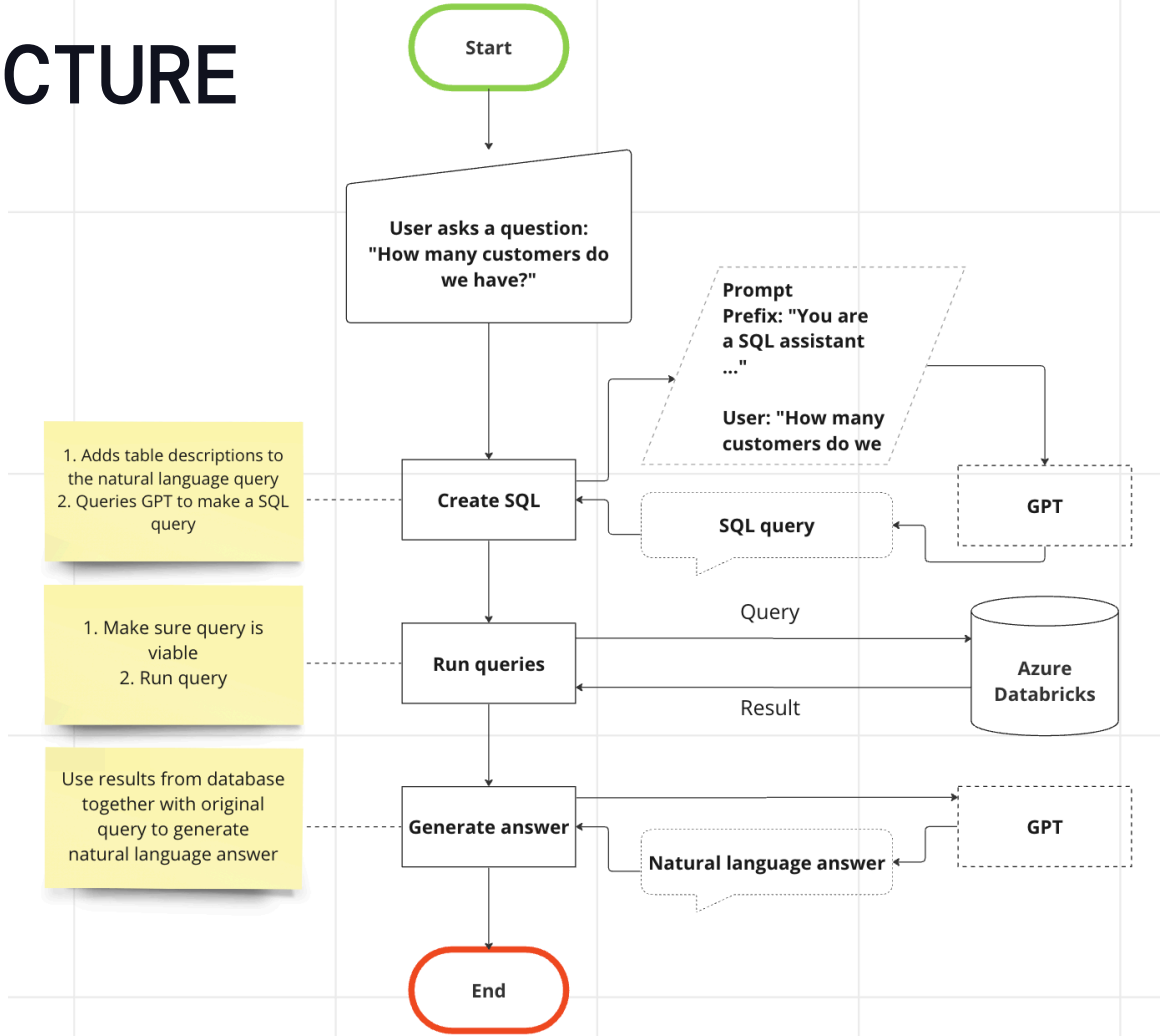
PYTHON




```
def generate_answer(question, query):
    response = OpenAIClient().completion(
        system_message=system_message_final_answer.format(
                question, query
        ),
        message="What is the answer to the question?",
    )
    return response.choices[-1].message.content
```


Classified: General Business
©2024 Databricks Inc. — All rights reserved

ARCHITECTURE



Classified: General Business

Gjensidige

Welcome! I'm Eglev, Gjensidiges virtual analyst fueled by GPT/LLM technology.

Logout

My scope

My limitations

User Guide

Hi, I'm Eglev. How can I help you?

Type in your question...