

# ServiceSAGE

## A GenAI-based RCA chat assistant



**Bhakti Kalghatgi**  
General Manager  
Tata Motors Ltd.



**Shubham Gupta**  
Senior Engineer  
Tata Motors Ltd.



**Koustubh Shirke**  
Sr. Application Engineer  
MathWorks

### OUR CULTURE CREDO

#### AT TATA MOTORS

We are connecting aspirations by being bold in thought and action, owning every opportunity and challenge, Solving together as one team and engaging all our stakeholders with empathy.

We are **MORE WHEN ONE!**

**BE BOLD**  
Taking calculated **risk** is key to making progress. We act with confidence and **agility** to accomplish our goals

**OWN IT**  
Feeling and acting **empowered** is critical to drive results. We have an **Owner's Mind-set** and each of us takes full responsibility for the outcomes

**SOLVE TOGETHER**  
Leveraging our collective genius while holding each other **accountable** helps us deliver the best. We **collaborate** proactively and transparently to achieve innovative solutions

**BE EMPATHETIC**  
**Embracing diversity** makes us stronger for differences are opportunities to learn. We work with **passion to delight customers** and deliver greater success to our stakeholders

# Agenda

01

**Automobile Diagnostics and Service Challenges**

02

**Generative AI**

03

**Retrieval Augmented Generation**

04

**Chatbot Workflow**

05

**Demo**

06

**ServiceSage Key features**



# Interactive RAG based Chatbot

The screenshot shows the ServiceSage App interface. At the top, there's a title bar with 'ServiceSage App' and window controls. Below that, a navigation bar has 'App' and 'Configuration' tabs. The main header displays 'ServiceSage' in a large blue font. On the left, a sidebar contains a 'Start a new conversation' button and a list of search suggestions: 'What are the trouble shooting', 'What is ISO code P0481-00?', 'What are the Preliminary Check', and 'What does P0122-14 tell?'. The main chat area shows a query: '\*\*Query: What are the trouble shooting steps for ISO code P0481-00 ?'. Below the query is the response: '^ ^ Response: 1. \*\*Initial Inspection and Preparation:\*\* - Turn the Ignition On, press the brake pedal for 3 or 4 times, then check the DTCs again. The DTC for P0481-00 should not reappear if the issue is resolved. 2. \*\*Key Evaluation:\*\* - If both keys fail to start the vehicle, replace the key set and perform key learning again with the new keys. 3. \*\*Wiring Harness Check:\*\* - Perform a visual inspection of the wiring harness connections as per the schematic. Rectify any loose or damaged connections. 4. \*\*Alternator Functionality Test:\*\* - Verify that the alternator is functional by checking the battery voltage when the engine is running, and ensuring the charging system operates correctly. 5. \*\*Igning System Check:\*\* - Switch off the IGN, wait for the ECU to reset, and then perform the vehicle start again in the next IGN ON cycle. 6. \*\*Fault Analysis:\*\* - This fault occurs when the manual switch is active while the Tap Up and Tap Down switches are also activated simultaneously. Rectify the wiring for this condition as per the schematic.' Below the response, a text box shows 'Vehicle Selected:Nano'. At the bottom of the chat area, there are 'Like' and 'Dislike' buttons. At the very bottom of the app, there's a text input field with the placeholder 'Enter your query and press Enter' and a send button. A footer at the bottom of the app window reads 'This is intellectual property of TATA Motors Ltd.'

# Generative AI

## What is

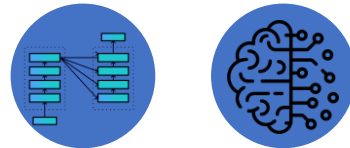


Advance AI that generates new, contextually relevant content such as Text, Images, or Designs based on learned patterns from large datasets

### In Traditional way which,

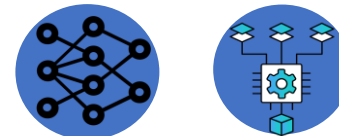
- Generates responses based on Predefined rules or Statistical models
- Requires significant retraining to adapt new information

## Build on



Built on Transformers and Large Language Models (LLMs), which analyze patterns and context in data to generate relevant outputs.

## How it works



Generative AI models use neural networks to identify patterns in existing data to generate new content

## Capabilities



**Text Generation**



**Image and Video Generation**



**Voice and Audio Generation**



**Automate manual & repetitive tasks**



**Predictive Analytics**

Hence, GenAI can multi-task and perform out-of-the-box tasks with minimal training required, foundation models can be adapted for targeted use cases with very little provided data.



# Generative AI in Automotive Industry



Chatbot



In - car Virtual Assistant



Synthetic Image / Video Data  
Generation - Autonomous Vehicles



Design & Prototyping



Simulation & Virtual Testing

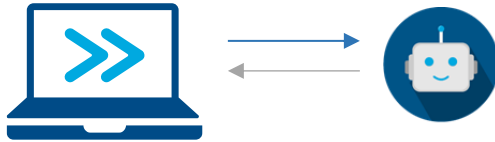


Adaptive Production Line

# Introducing Retrieval Augmented Generation (RAG)

01

## Prompt Engineering



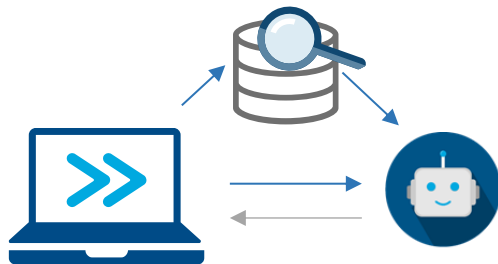
Input to guide LLMs in generating desired outputs

### Challenges & Limitations

- Hallucination
- Context and Relevance
- Data Privacy

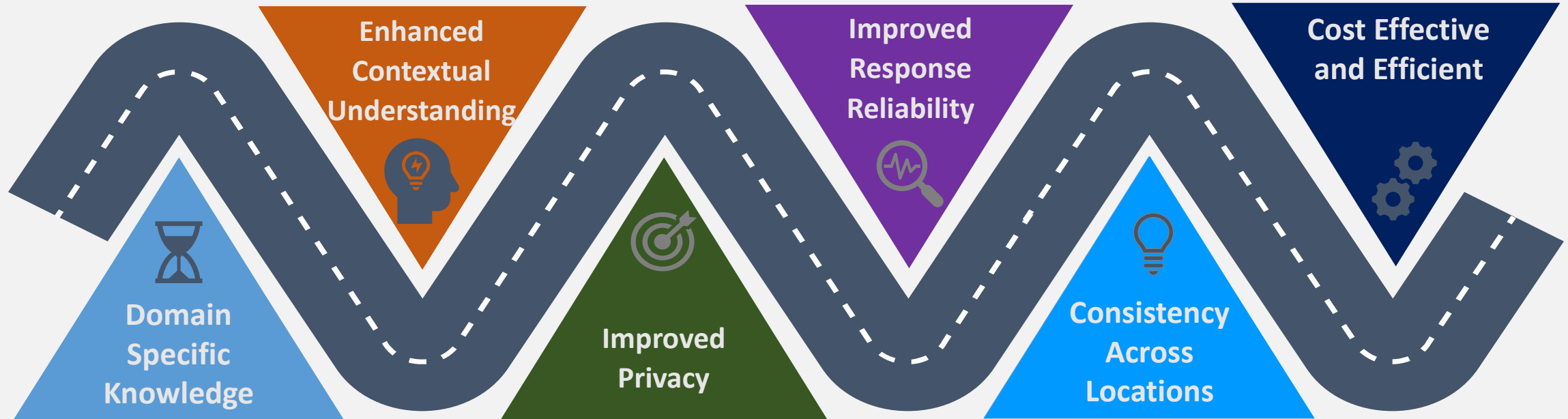
02

## Retrieval Augmented Generation



Input to guide LLMs with additional information from an external knowledge source to generate relevant outputs

# Significance of RAG

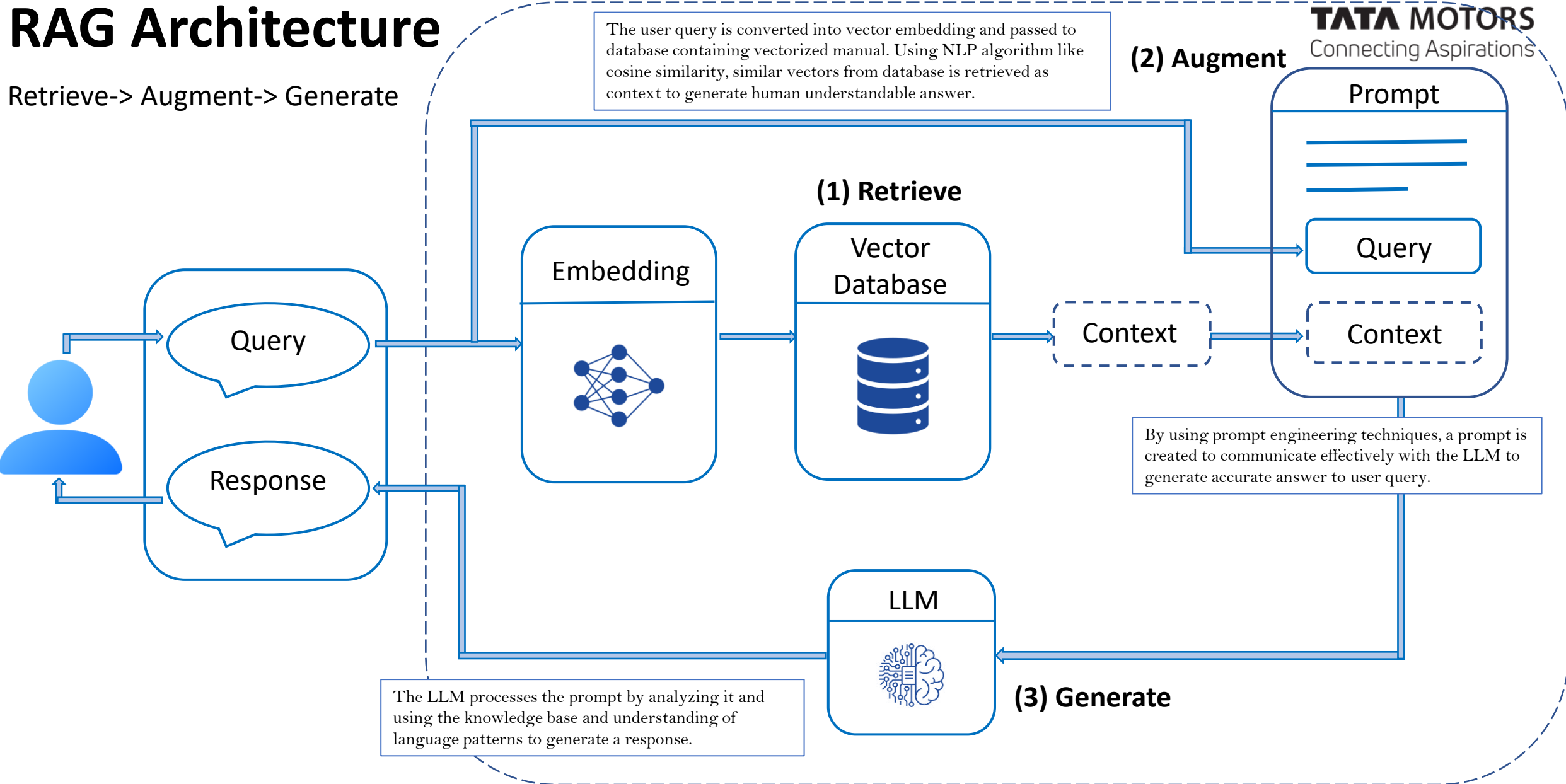


RAG enables the development of a domain-specific RCA assistant that's fast, reliable, and specialized for the unique demands of vehicle diagnostics.



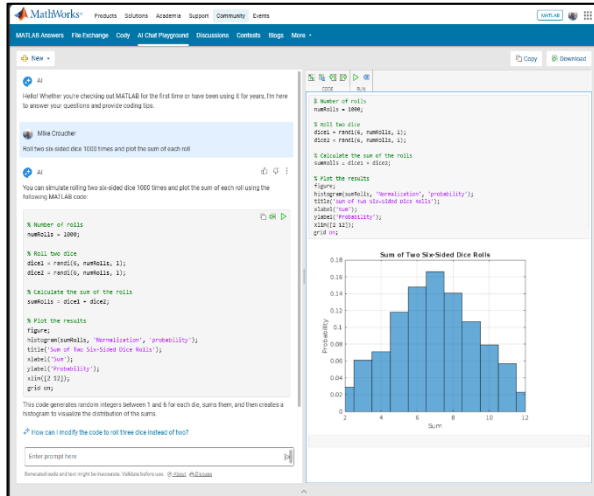
# RAG Architecture

Retrieve-> Augment-> Generate

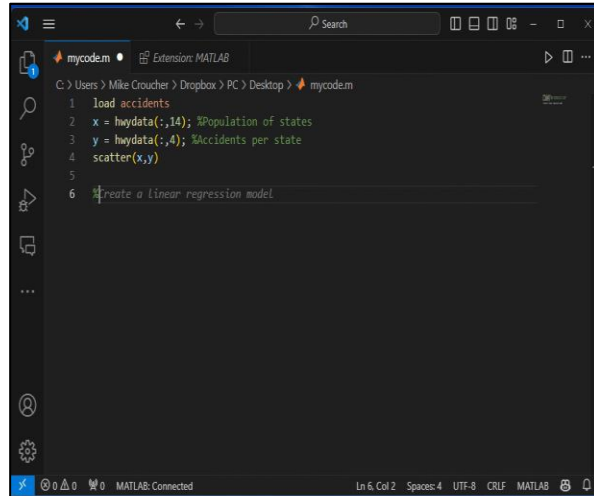


RAG Increases the model capability, By Retrieve the external up to date information, Augmenting the prompt that was given, Generating a response using contexts & information.

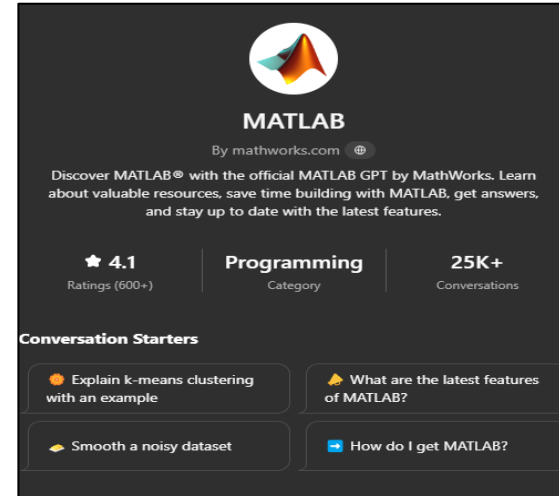
# 4 ways of using MATLAB with LLMs



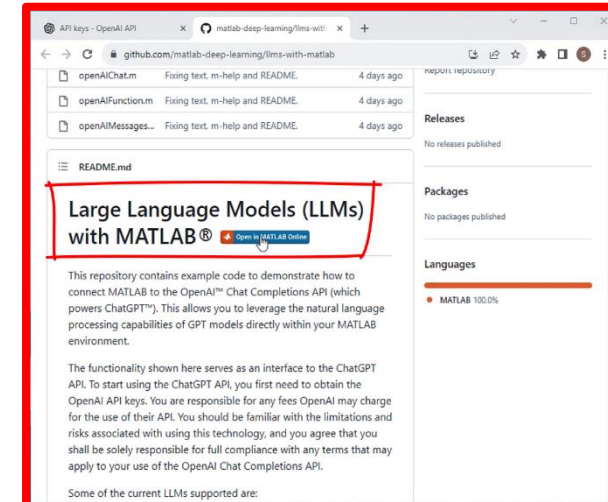
MATLAB AI Chat Playground



GitHub Co-pilot with the MATLAB extension for Visual Studio Code

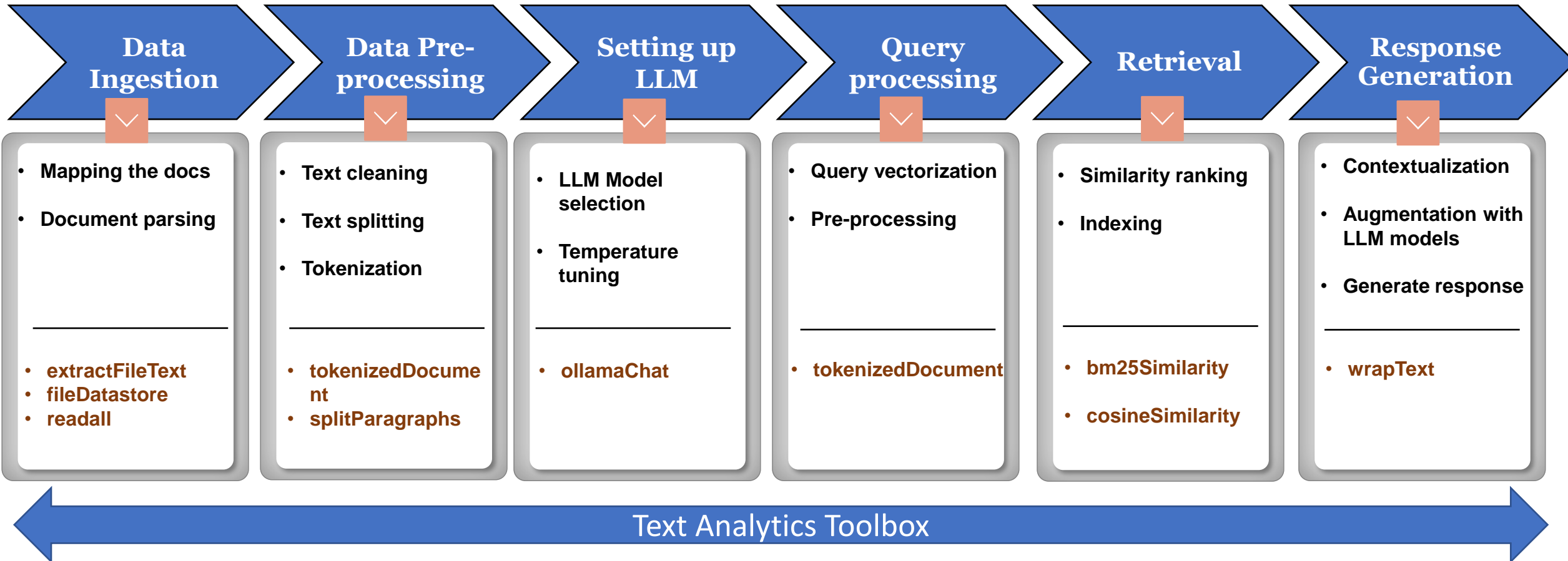


Custom MATLAB GPT in the OpenAI GPT Store

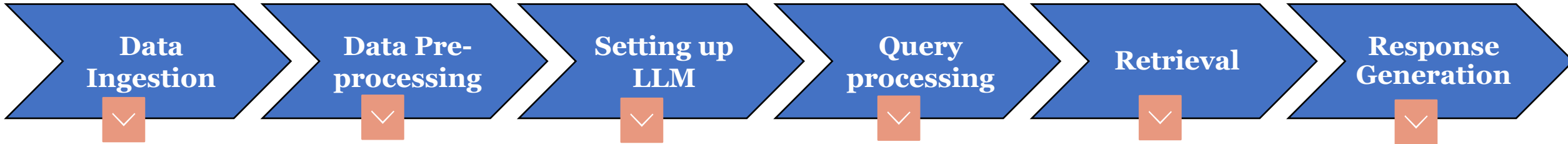


Large Language Models (LLMs) with MATLAB

# MATLAB based RAG Workflow



# MATLAB based RAG Workflow



No Code

**Preprocess Text Data** ▶  Autorun ? :

preprocessedText = Preprocessed text in tbl using lemmatization and word removal

▼ Select data  
Data:

▶ Clean up HTML

▶ Tokenize

▶ Add token details

▼ Change and remove words

Word normalization:  Case normalization:

Minimum word length:   Maximum word length:

Remove stop words  Erase punctuation

Replace words  
  +

Remove words  
 +

Remove empty documents  Ignore case

▼ Display results  
 Show tokenized text  Show token details  Show word cloud

▶ Show code

Low Code

```
function allDocs = preprocessDocuments(str)
    tokenized = tokenizedDocument(join(str,[newline newline]));
    allDocs = splitParagraphs(tokenized);
end

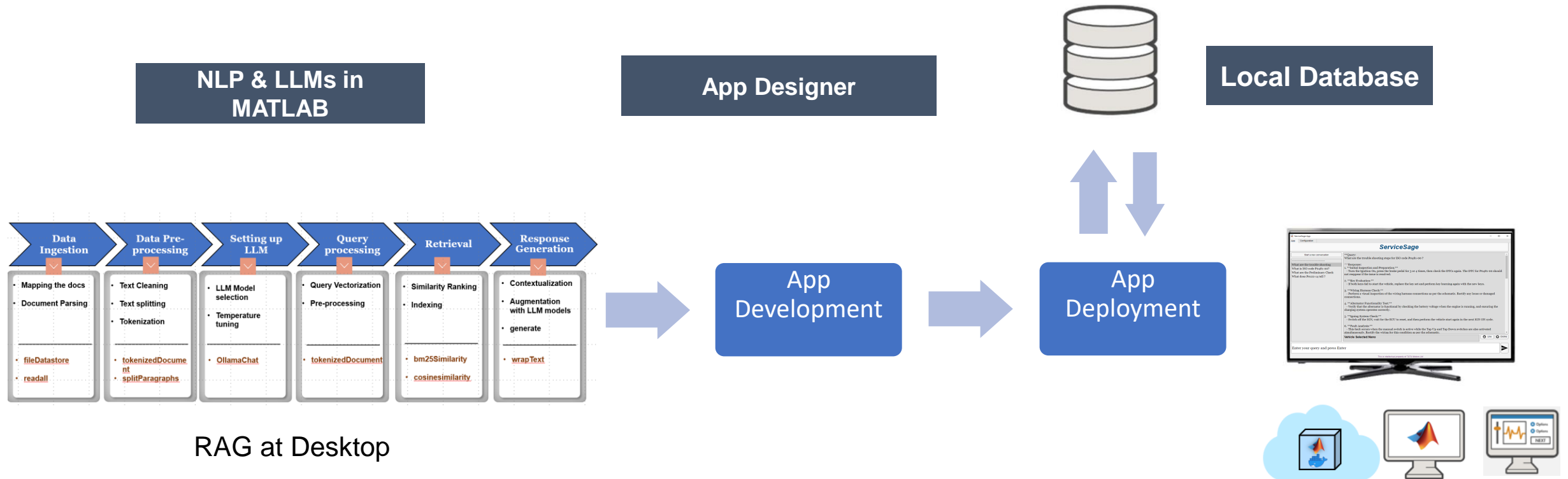
function wrappedText = wrapText(text)
    s = textwrap(text,80);
    wrappedText = string(join(s,newline));
end
```

```
chat = ollamaChat("llama3.1", ...
    "You are a helpful assistant. You will get a " + ...
    "context for each question, but only use the information " + ...
    "in the context if that makes sense to answer the question." + ...
    "Let's think step-by-step, explaining how you reached the answer.", ...
    TimeOut=600, Temperature=0.8)'
```

```
bm25Similarity
Document similarities with BM25 algorithm
Since R2020a

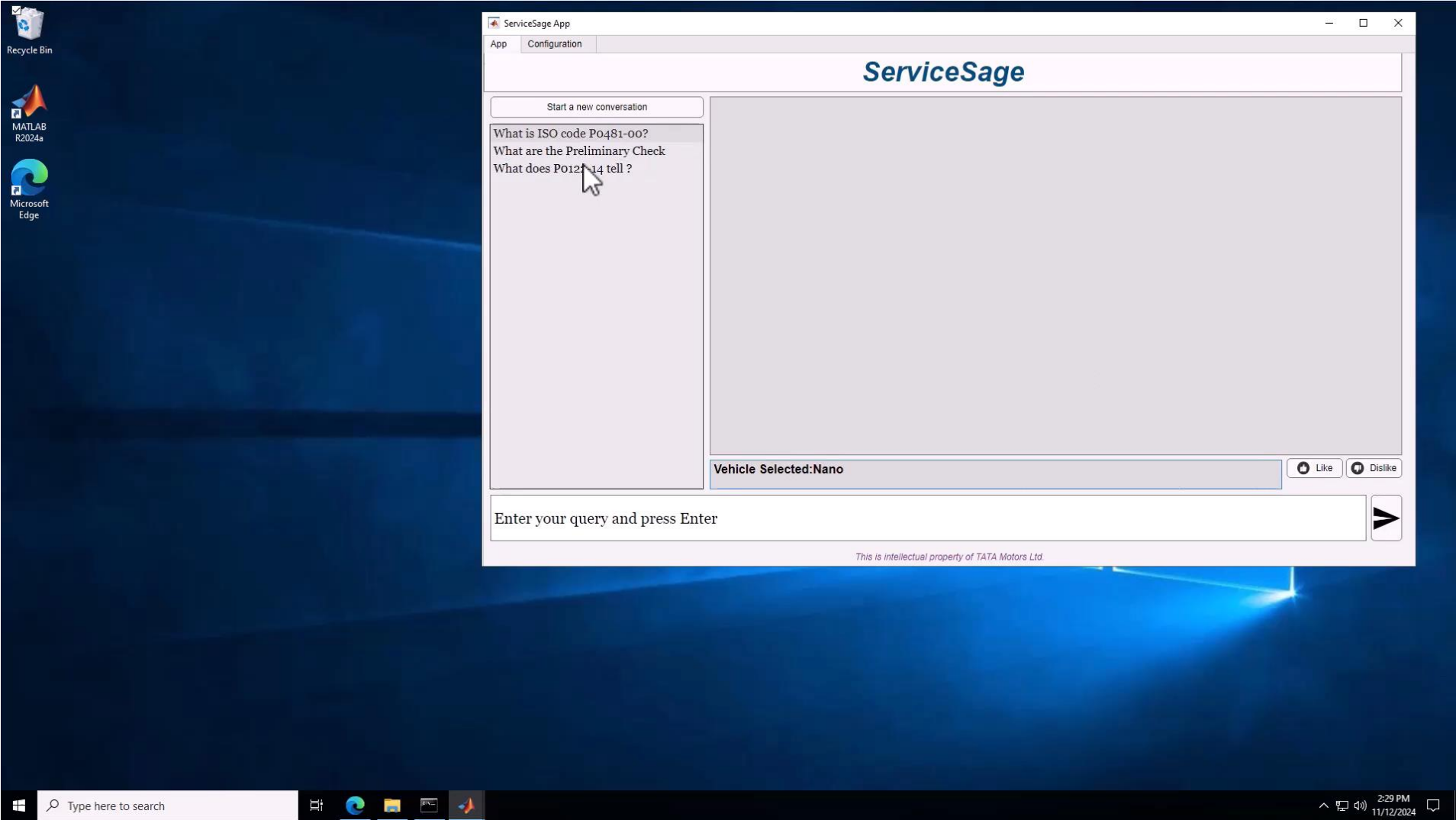
Syntax
similarities = bm25Similarity(documents)
similarities = bm25Similarity(documents,queries)
```

# MATLAB based RAG Workflow





# Interactive RAG based Chatbot



# Chatbot Highlights

The screenshot shows the ServiceSage App interface. At the top, there are tabs for 'App' and 'Configuration'. The main header displays 'ServiceSage'. On the left, there is a sidebar with a 'Start a new conversation' button and a list of historical chat topics: 'What are the trouble shooting', 'What is ISO code P0481-00?', 'What are the Preliminary Check', and 'What does P0122-14 tell?'. The main chat area shows a query: 'What are the trouble shooting steps for ISO code P0481-00?' and a detailed response with six numbered steps: 1. Initial Inspection and Preparation, 2. Key Evaluation, 3. Wiring Harness Check, 4. Alternator Functionality Test, 5. Ignition System Check, and 6. Fault Analysis. Below the response, it says 'Vehicle Selected: Nano'. At the bottom, there is a text input field with the placeholder 'Enter your query and press Enter' and a 'Like' button, a 'Dislike' button, and a send button. A footer note reads 'This is intellectual property of TATA Motors Ltd.'

To start new chat →

Historical chats {

Selected Vehicle →

Querying →

Queries and response display {

Interactive Feedback {

Choosing the vehicle variant

Limiting no of lines for chat history display

Choosing string length of chat history

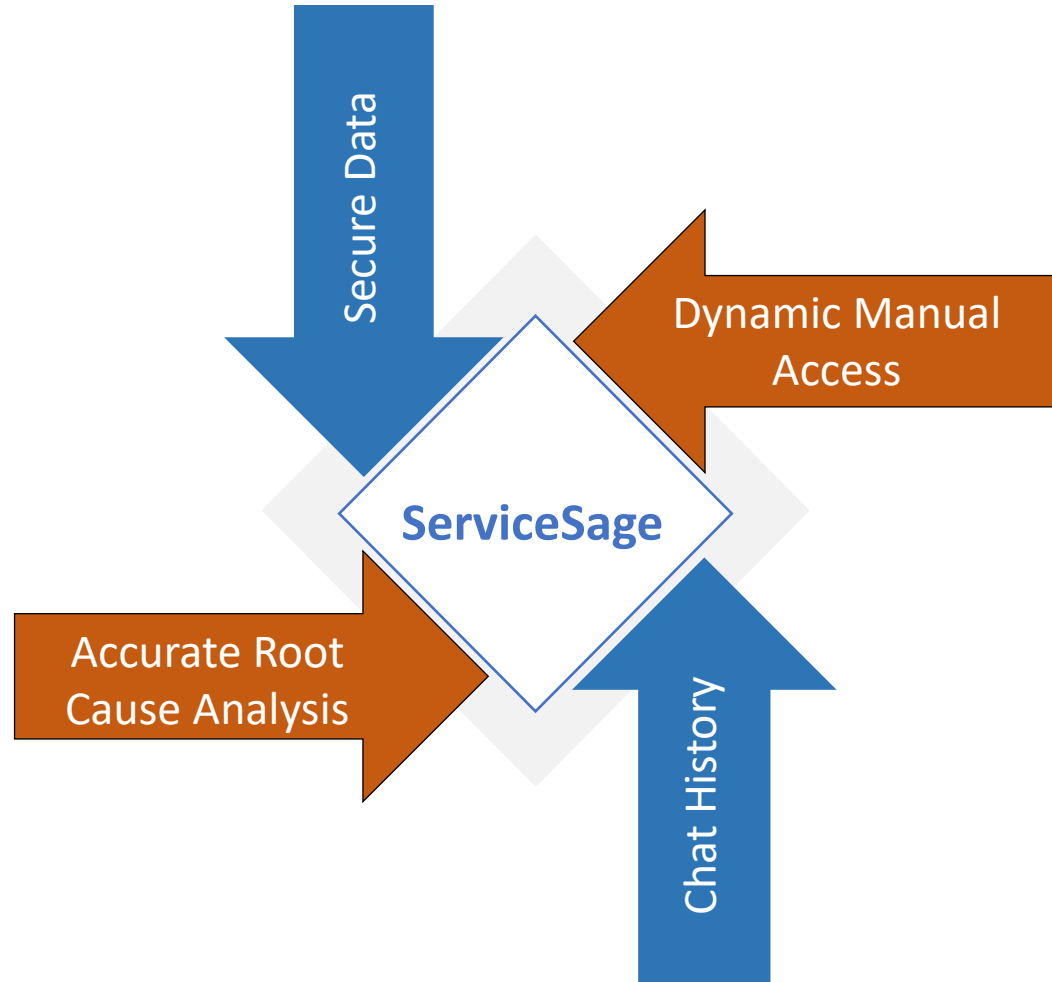
Set up maximum no. of queries accepted in chat window

The screenshot shows the 'ServiceSage App' configuration window with the following settings:

Configuration Item	Value
Select Vehicle	Nano
Number of Lines of Chat History	10
String Length of Chat history	30
Max Number Of Prompt Accepted	1000

Additional UI elements include an 'Open Service Manual' button next to the vehicle selection dropdown and a 'Close All text File' button at the bottom right.

# ServiceSage Key Features



Data Access stay Locally & within Organization

Easily select Vehicle Variant and Switch between different Manuals Instantly

Retains Context across Multiple Interactions

Provide Precise Data driven Diagnostic Insights

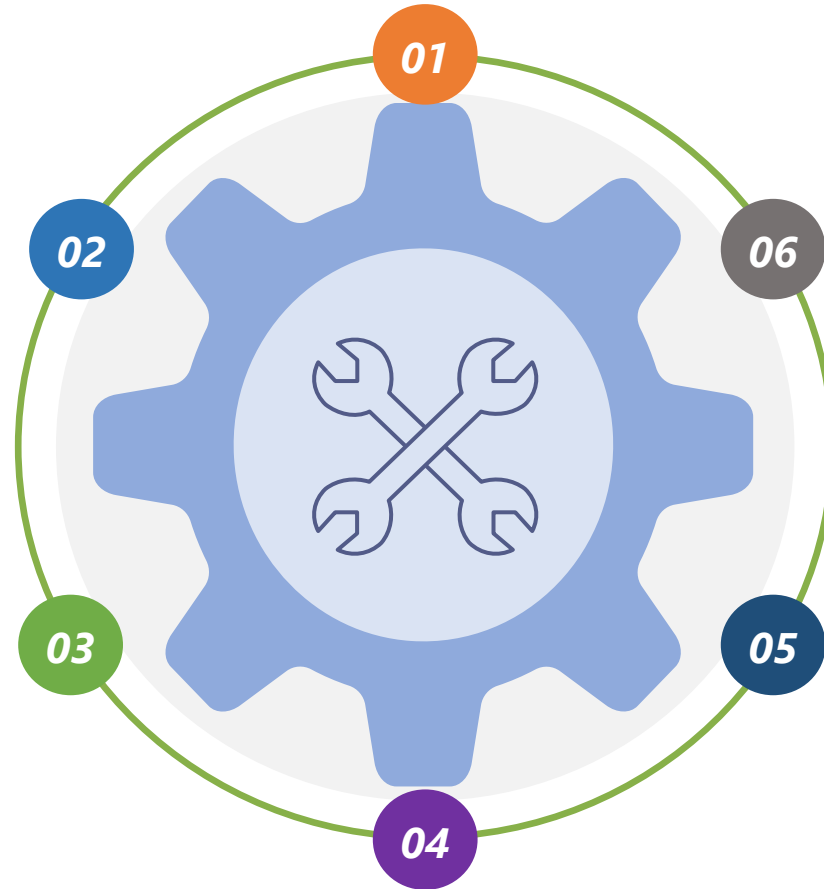
Fault Specific Troubleshooting Recommendations

# Potential Impact

**1** Enhance Customer Satisfaction

**2** Improve Service Reliability

**3** Reduce Turn over Time



**6** Improve Technician's Competency

**5** Improve Technician Productivity

**4** Improve Service Operations & Validation





# Thank you

Open to Query...