Context intro:

Context helps LLM with meaning of the text. Provides background information in a conversation that assists in understanding.

Guides the LLM in the “meaning” of the text

LLM APIs are stateless, they do not maintain a history or store information of prior requests. They can only process one request at a time.

The amount of text you send to LLM is limited by the size of the model’s context window. Context window sizes vary depending upon the model.

Two types of context:

Prompt context and Chatbot context

Prompt context: Additional text in your prompt that guides LLM to give you a useful completion

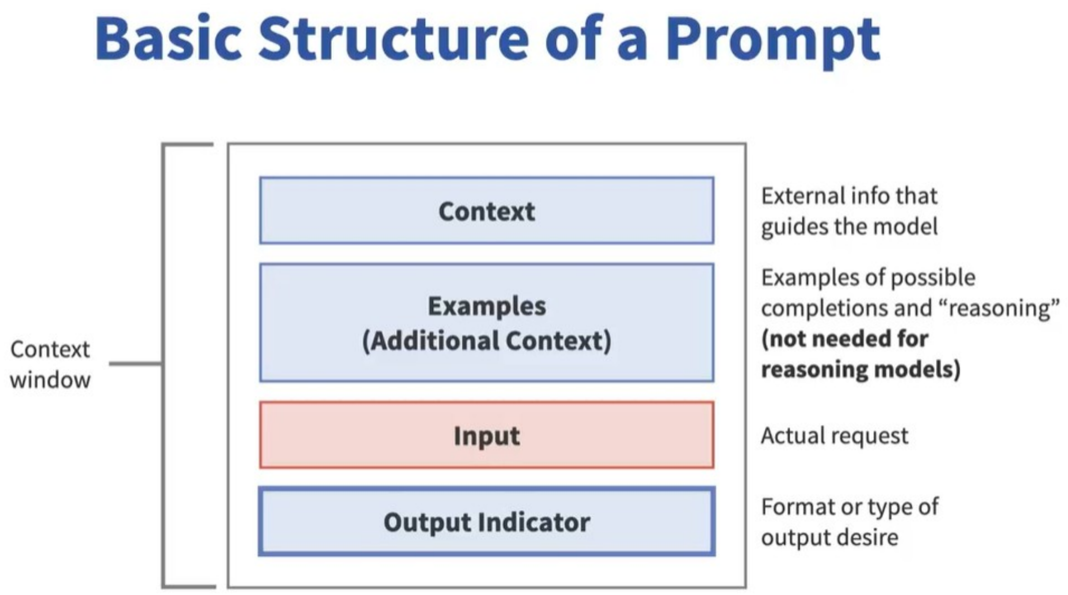
Chatbot context: Additional text in your ongoing conversation with a chatbot.

At the API level, they are both considered the same as prompt context.

Chatbots need to maintain the context of an active conversation. The more relevant the context, the better the response.

Basic structure of a prompt:

We have the core prompt surrounded by context. It could be basic context like contents of a file or information from a webpage or even from an internal database. If you are using non-reasoning model, you could add patterns of examples, or Chain of Thought problem-solving. If you are using a reasonable model, you do not need to add any patterns of examples. You may also have an output indicator such as the language you want the output to be in or JSON, HTML format.

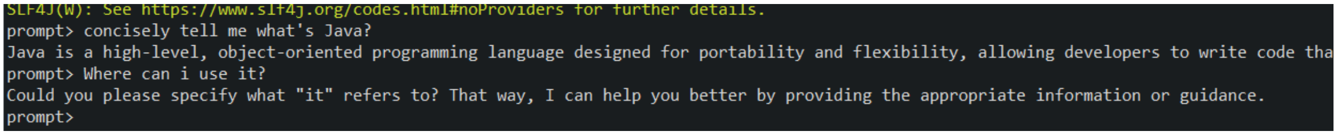

Basic Structure of a Prompt
Context window
Context
Examples (Additional Context)
Input
Output Indicator
External info that guides the model
Examples of possible completions and “reasoning” **(not needed for reasoning models)**
Actual request
Format or type of output desire

Prompts could be even entire collection of books. However, language models could forget information within a large context window. Sometimes, section order matters among LLMs.

LLM Rest calls are stateless. Each API invocation is totally independent. Prompt context is critical for generating useful responses as it guides language model’s output. Any text that exceeds the context window will be lost.

I ran Chat.Java without providing conversation history
prompt> concisely tell me what's Java?
Java is a high-level, object-oriented programming language designed for portability and flexibility, allowing developers to write code that can run on any device with a Java Virtual Machine (JVM). It is widely used for building web applications, Android apps, enterprise systems, and more, emphasizing features like robustness, security, and platform independence.
prompt> Where can i use it?
Could you please specify what "it" refers to? That way, I can help you better by providing the appropriate information or guidance.
prompt>

Model doesn’t know what “it” is.



```
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
prompt> concisely tell me what's Java?
Java is a high-level, object-oriented programming language designed for portability and flexibility, allowing developers to write code tha
prompt> Where can i use it?
Could you please specify what "it" refers to? That way, I can help you better by providing the appropriate information or guidance.
prompt>
```

Now I provide additional context then I ask the same question “Where can I use it?”

prompt> concisely tell me what is Java?
Java is a high-level, object-oriented programming language known for its platform independence, due to the Java Virtual Machine (JVM), which allows Java applications to run on any device with the JVM installed. It is widely used for building cross-platform applications, from web and mobile apps to enterprise-level systems, and is valued for its portability, scalability, and robust security features.
prompt> Where can i use it?
Could you please provide more context or specify what "it" refers to? I'd be happy to help with more information.
prompt> Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation). It is designed to be platform-independent, allowing developers to write code that can run on any device equipped with a compatible Java Virtual Machine (JVM). Known for its "write once, run anywhere" capability, Java is widely used for building web applications, mobile apps, and enterprise software.
Where can i use it?
Java is indeed a versatile and widely-used programming language, well-regarded for its platform independence and robustness. Its design philosophy, encapsulated in the "write once, run anywhere" principle, is achieved through the use of the Java Virtual Machine (JVM), which allows Java programs to run on any device or operating system that has a compatible JVM.

Java's object-oriented nature encourages modular programming and code reuse, making it a popular choice for complex software development projects. It is extensively used in various domains, such as:

1. **Web Applications:** Java is used in server-side applications due to its stability and scalability. Frameworks like Spring and JavaServer Faces (JSF) are popular in developing robust and secure web applications.

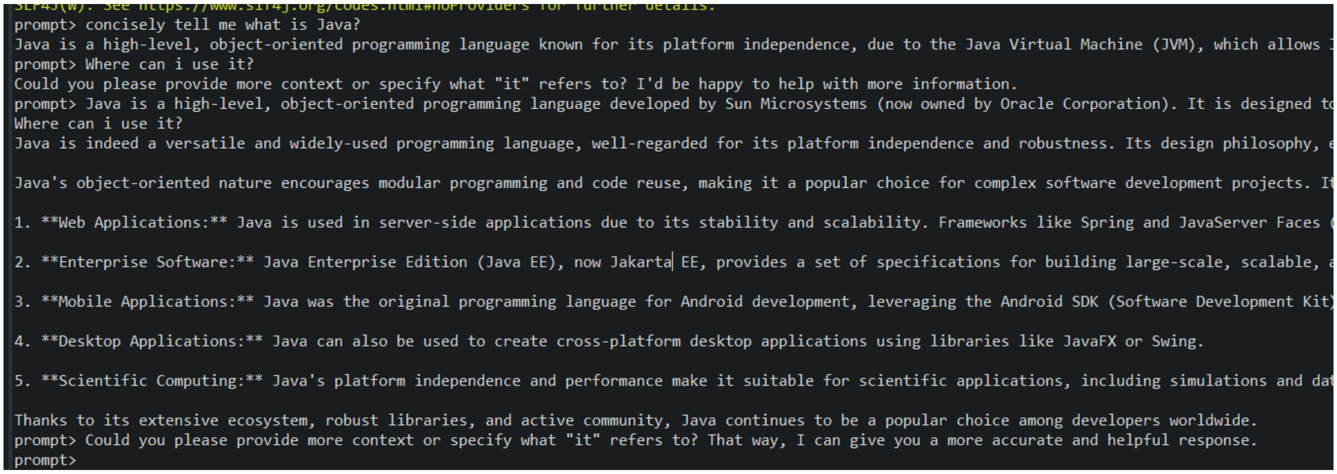
2. **Enterprise Software:** Java Enterprise Edition (Java EE), now Jakarta EE, provides a set of specifications for building large-scale, scalable, and secure enterprise applications. It includes APIs for database access, web services, and messaging.

3. **Mobile Applications:** Java was the original programming language for Android development, leveraging the Android SDK (Software Development Kit). Although Kotlin is now officially preferred, Java remains a significant part of Android app development.

4. **Desktop Applications:** Java can also be used to create cross-platform desktop applications using libraries like JavaFX or Swing.

5. **Scientific Computing:** Java's platform independence and performance make it suitable for scientific applications, including simulations and data processing.

Thanks to its extensive ecosystem, robust libraries, and active community, Java continues to be a popular choice among developers worldwide.
prompt> Could you please provide more context or specify what "it" refers to? That way, I can give you a more accurate and helpful response.
Prompt>



```
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
prompt> concisely tell me what is Java?
Java is a high-level, object-oriented programming language known for its platform independence, due to the Java Virtual Machine (JVM), which allows J
prompt> Where can i use it?
Could you please provide more context or specify what "it" refers to? I'd be happy to help with more information.
prompt> Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation). It is designed to
Where can i use it?
Java is indeed a versatile and widely-used programming language, well-regarded for its platform independence and robustness. Its design philosophy, e

Java's object-oriented nature encourages modular programming and code reuse, making it a popular choice for complex software development projects. It

1. **Web Applications:** Java is used in server-side applications due to its stability and scalability. Frameworks like Spring and JavaServer Faces (

2. **Enterprise Software:** Java Enterprise Edition (Java EE), now Jakarta EE, provides a set of specifications for building large-scale, scalable, a

3. **Mobile Applications:** Java was the original programming language for Android development, leveraging the Android SDK (Software Development Kit)

4. **Desktop Applications:** Java can also be used to create cross-platform desktop applications using libraries like JavaFX or Swing.

5. **Scientific Computing:** Java's platform independence and performance make it suitable for scientific applications, including simulations and dat

Thanks to its extensive ecosystem, robust libraries, and active community, Java continues to be a popular choice among developers worldwide.
prompt> Could you please provide more context or specify what "it" refers to? That way, I can give you a more accurate and helpful response.
prompt>
```

prompt> concisely tell me what is Java?
Java is a high-level, object-oriented programming language known for its platform independence, due to the Java Virtual Machine (JVM), which allows Java applications to run on any device with the JVM installed. It is widely used for building cross-platform applications, from web and mobile apps to enterprise-level systems, and is valued for its portability, scalability, and robust security features.
prompt> Where can i use it?
Could you please provide more context or specify what "it" refers to? I'd be happy to help with more information.
prompt> Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation). It is designed to be platform-independent, allowing developers to write code that can run on any device equipped with a compatible Java Virtual Machine (JVM). Known

for its "write once, run anywhere" capability, Java is widely used for building web applications, mobile apps, and enterprise software.

Where can i use it?

Java is indeed a versatile and widely-used programming language, well-regarded for its platform independence and robustness. Its design philosophy, encapsulated in the "write once, run anywhere" principle, is achieved through the use of the Java Virtual Machine (JVM), which allows Java programs to run on any device or operating system that has a compatible JVM.

Java's object-oriented nature encourages modular programming and code reuse, making it a popular choice for complex software development projects. It is extensively used in various domains, such as:

1. **Web Applications:** Java is used in server-side applications due to its stability and scalability. Frameworks like Spring and JavaServer Faces (JSF) are popular in developing robust and secure web applications.

2. **Enterprise Software:** Java Enterprise Edition (Java EE), now Jakarta EE, provides a set of specifications for building large-scale, scalable, and secure enterprise applications. It includes APIs for database access, web services, and messaging.

3. **Mobile Applications:** Java was the original programming language for Android development, leveraging the Android SDK (Software Development Kit). Although Kotlin is now officially preferred, Java remains a significant part of Android app development.

4. **Desktop Applications:** Java can also be used to create cross-platform desktop applications using libraries like JavaFX or Swing.

5. **Scientific Computing:** Java's platform independence and performance make it suitable for scientific applications, including simulations and data processing.

Thanks to its extensive ecosystem, robust libraries, and active community, Java continues to be a popular choice among developers worldwide.

prompt> Could you please provide more context or specify what "it" refers to? That way, I can give you a more accurate and helpful response.

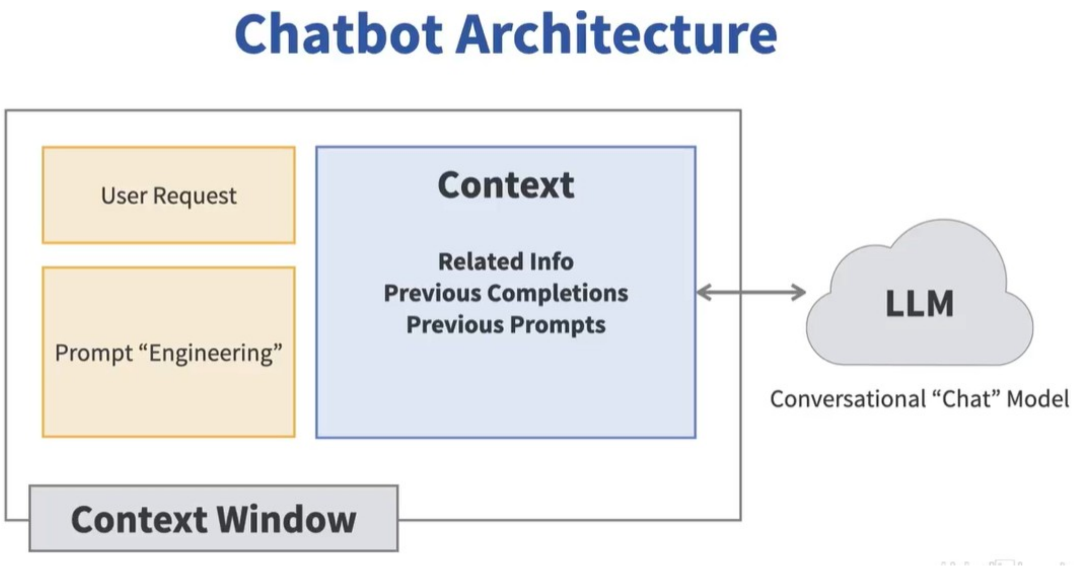
prompt> why is it popular?

Could you specify what "it" refers to? If you are referring to a specific topic, trend, product, or phenomenon, please provide more details so I can give you a more accurate answer.

prompt>

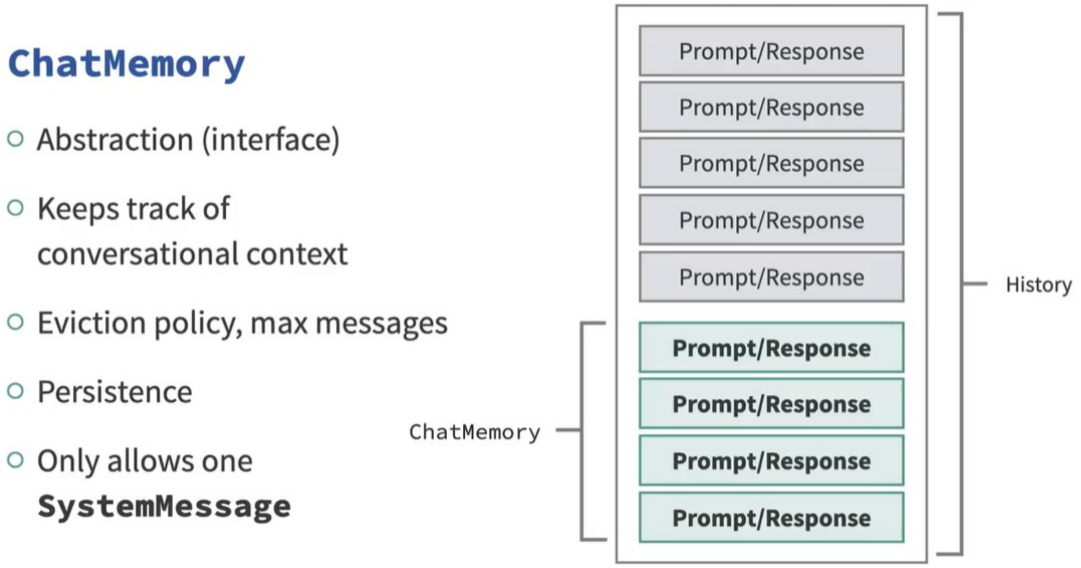
Again it doesn’t understand what “it” is

Chatbot Architecture with LangChain4J



To maintain conversational context, you could manually save ChatMessage and ChatResponse objects (text) to resend to the LLM.
The other option is, LangChain4j offers an abstraction called **ChatMemory**

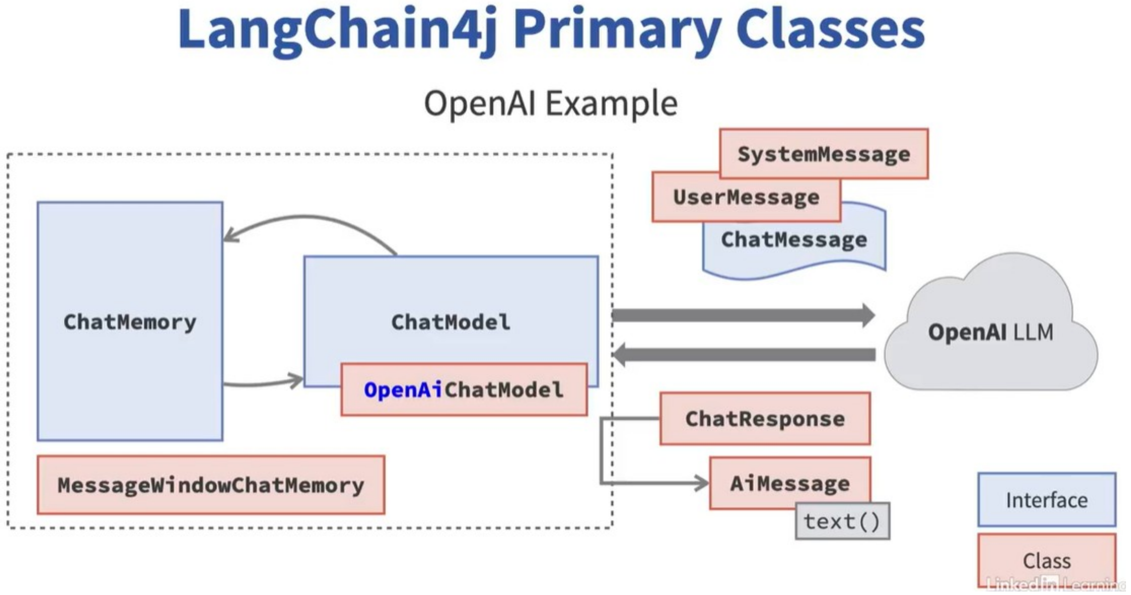
ChatMemory is a very helpful Java interface with some useful implementations for developers. You could specify how many messages you want to store in a ChatMemory. ChatMemory has an eviction policy, oldest messages are automatically removed from the ChatMemory instance that you create. One of the nice features of ChatMemory is, it allows its contents to be persistent. By making ChatMemory persistent, you can make an application that resumes a conversation with a user. One limitation with ChatMemory is that it allows only one SystemMessage. LangChain4j’s ChatMemory abstraction only allows one. If you add multiple SystemMessages to ChatMemory, only the last one is retained.



ChatMemory

- ○ Abstraction (interface)
- ○ Keeps track of conversational context
- ○ Eviction policy, max messages
- ○ Persistence
- ○ Only allows one **SystemMessage**

There is something called as “MessageWindowChatMemory”, which is an implementation of ChatMemory interface. We add the ChatMessageResponse to ChatMemory, along with user’s original user messages and system message. Next time, when we talk to LLM, we need to extract contents of the ChatMemory object and we send that along with a new prompt. Note size of the prompt is increasing each time, which further increases the computation costs and latency. It is your responsibility to make sure, total prompt length does not exceed the size of the context window of the language model you are using.



LangChain4j Primary Classes

OpenAI Example

<u>Output of ChatWithContext:</u>

it is not talking about Java, it is talking about some generic information

prompt> Can I use it for Generative AI?
Absolutely, you can certainly use Artificial Intelligence (AI) and Machine Learning (ML) for Generative AI. Generative AI is a subset of artificial intelligence that allows computers to creatively produce output on their own. This could take the form of writing text, music, drawing pictures, or even creating complex data structures.

The application of Machine Learning in Generative AI is usually seen through Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). These Machine Learning models can understand and generate new content with certain learned characteristics.

GANs, for example, consist of two parts: a generator that produces data and a discriminator that attempts to differentiate between real and generated data. Over time, the generator becomes proficient at producing data that is almost indistinguishable from the real data.

VAEs on the other hand, are probabilistic models that can capture the essence of the data distribution and generate new data that follows the same distribution.

So, to answer your question, yes. AI and Machine Learning indeed can and are extensively used for Generative AI. Whether you can do it depends on your familiarity and understanding of these technologies. If you need help learning these or implementing them, feel free to ask.
prompt>

I simply said “them” instead of “Java records” in the third prompt and it knows from the context and gives correct response

prompt> Can I use it for Generative AI?
Absolutely, you can use Artificial Intelligence (AI) and Machine Learning (ML) techniques for Generative AI.

Generative AI is a subset of AI that includes technologies capable of creating something new. It's a form of machine learning where AI models are trained to generate new data that are similar to a provided dataset.

For instance, chatbots and automatic text generators use generative AI to produce sentences. It also plays a significant role in creating images, improving photo resolution, creating human-like voices, etc.

To use Generative AI, one typically involves technologies such as Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) which are advanced ML algorithms that provide the ability for an AI to generate data. These methods can learn the essential characteristics of different types of data like images, music, voice, and even text, and then generate new, synthetic data that can pass for original data.

Of course, using these sophisticated technologies effectively will require a deep understanding of machine learning principles and techniques. If you're interested, I would be more than happy to guide your learning and help you leverage Generative AI in your work or projects.
prompt> What are Java records?
Java Record is a new feature introduced in Java 14 as a preview feature and finalized in Java 16. Records in Java provide a way to model plain data aggregates with less ceremony.

A Java Record is a special type of class that can hold pure data. It's similar to a normal class, but with some key differences:
1. A record is an immutable class that automatically implements several useful interfaces, including equals(), hashCode(), and toString().
2. A Record's state is defined in the record header, and this state is both final and private.
3. A record class will automatically create getter methods (but not setter, because it's immutable) for each of the variables defined in the record header, following the JavaBean convention.
4. Record classes implicitly declare a canonical constructor (one that takes all the fields), and it's not necessary to declare this manually.

Here is an example of a record:

```java
public record Employee(String firstName, String lastName, int employeeId) {
  // All other code, like methods or other logic, can go here
}
```

In this example, `Employee` is a record having a `firstName`, a `lastName`, and an `employeeId` as its state. And we get equals(), hashCode(), toString(), and corresponding getters automatically from the JVM. However, keep in mind that records are intended to be simple data carriers and may not be the best choice if you need more complex class behaviors.
prompt> Can I use them for GenAI applications?
Yes, Java Records can be used in Generative AI (GenAI) applications, provided you are using Java for your AI work.

Here's how: GenAI often involves processing large amounts of data. This data might be used for training models, generating new data, or other tasks. While the heavy lifting is typically done by machine learning libraries that may be written in languages best suited to numerical computing (like Python), there's often a significant amount of "glue code" and data handling code needed to support these tasks.

Java Records could be quite useful for handling this data. Because they are a way of creating simple, immutable data carrier classes, they could be used to model the data that you're processing. For example, if you're working with a dataset of images where each image has a label and a matrix of pixel data, you might create a Record to neatly encapsulate these two data items.

However, keep in mind that the core modeling, learning, and generation tasks in a GenAI workload are typically performed using specialized libraries (like TensorFlow, PyTorch, etc.) that have extensive support for the kind of high-performance numerical computing that these tasks require. Java is not typically the first choice of programming language for this part of a GenAI application, though it can serve an important role in the broader application architecture, for tasks like data preprocessing, system integration, etc.
prompt>

Conversation context across API invocations is now maintained. Chat context grows along with amount of data and with large number of users, amount of data could be substantial that means data costs could go up.

Some of that context data may not be very relevant though
We may need additional text in the form of other documents like video/audio transactions, or web for an improved response