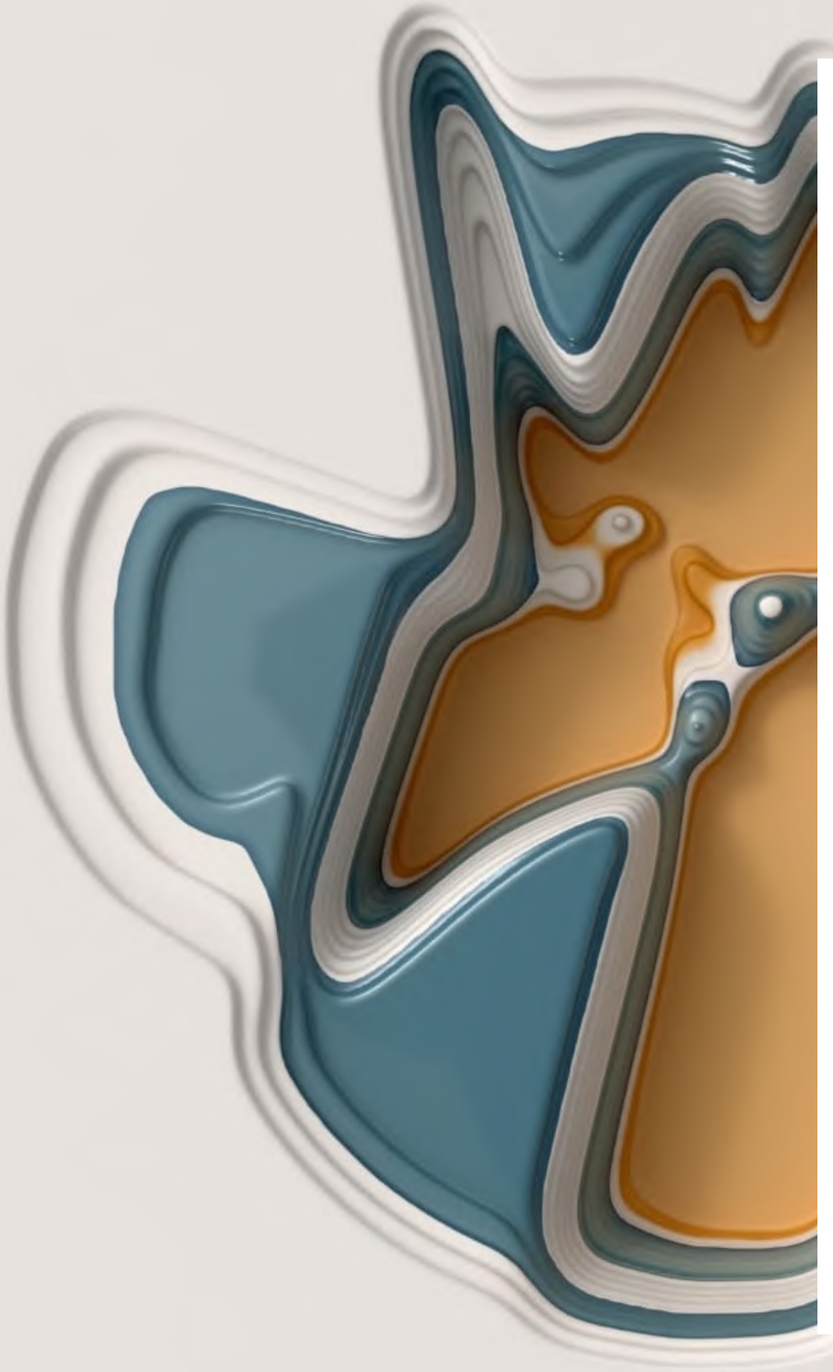


TECHNOLOGY DAY / 13.11.2023

Enhancing LLMs to build fact-based Chatbots for your Domain

INNOQ

MARCO STEINKE
CONSULTANT

2022

ChatGPT Is a Tipping Point for AI

by Ethan Mollick

December 14, 2022

The New Chatbots Could Change the World. Can You Trust Them?

Siri, Google Search, online marketing and your child's homework will never be the same. Then there's the misinformation problem.

ARTIFICIAL INTELLIGENCE >

ChatGPT achieves in six months what Facebook needed a decade to do: The meteoric rise of the AI chatbot

Here is a look at how the bot has grown compared to other pioneering platforms such as YouTube, Spotify and Instagram

ARTIFICIAL INTELLIGENCE

ChatGPT is going to change education, not destroy it

The narrative around cheating students doesn't tell the whole story. Meet the teachers who think generative AI could actually make learning better.

By Will Douglas Heaven

April 6, 2023

OpenAI's ChatGPT chatbot blocked in Italy over privacy concerns

By **Euronews** with **AFP**

Published on 31/03/2023 - 13:07 • Updated 13:46

ChatGPT answers more than half of software engineering questions incorrectly

You may want to stick to Stack Overflow for your software engineering assistance.

Written by **Sabrina Ortiz,** Editor

Aug. 9, 2023 at 4:23 p.m. PT

ChatGPT has mastered the confidence trick, and that's a terrible look for AI

It's very good, and that's very bad

> When there is great power, there is a great responsibility

probably Winston Churchill

Large Language Model

Large Language Model

Large Language Model

- 175 billion parameters (GPT3)
- Training: Estimated a hypothetical cost of around $4.6 million US dollars
- 355 years to train GPT-3 on a single GPU in 2020

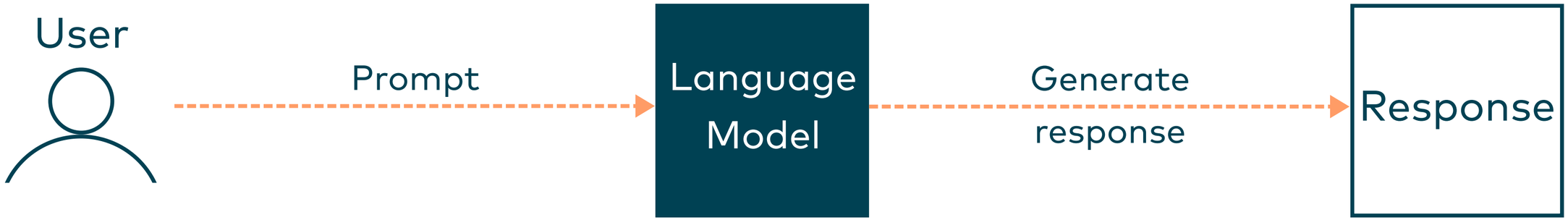
Large Language Model

Large Language Model

- general-purpose language understanding and generation

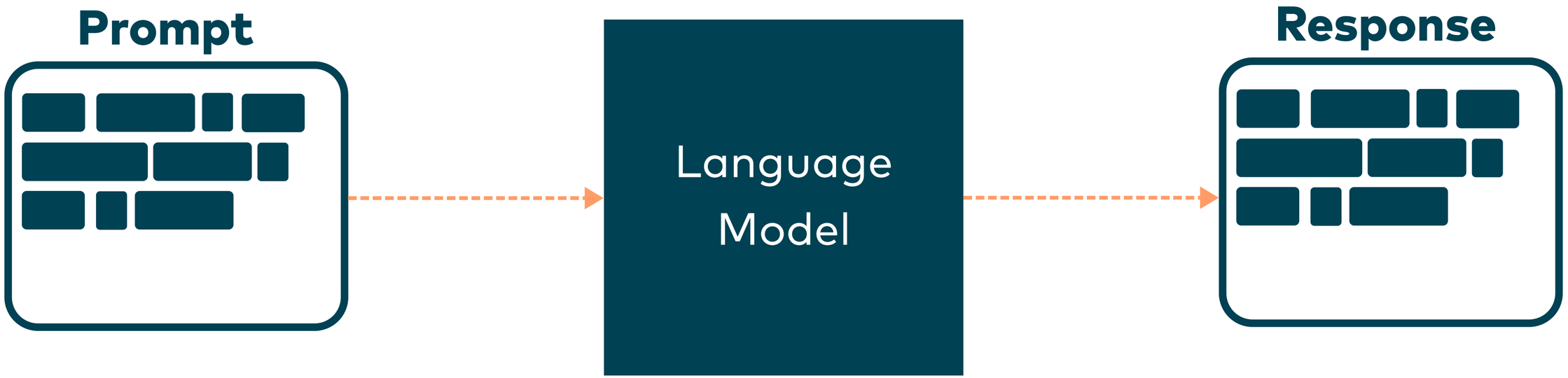
Large Language Model

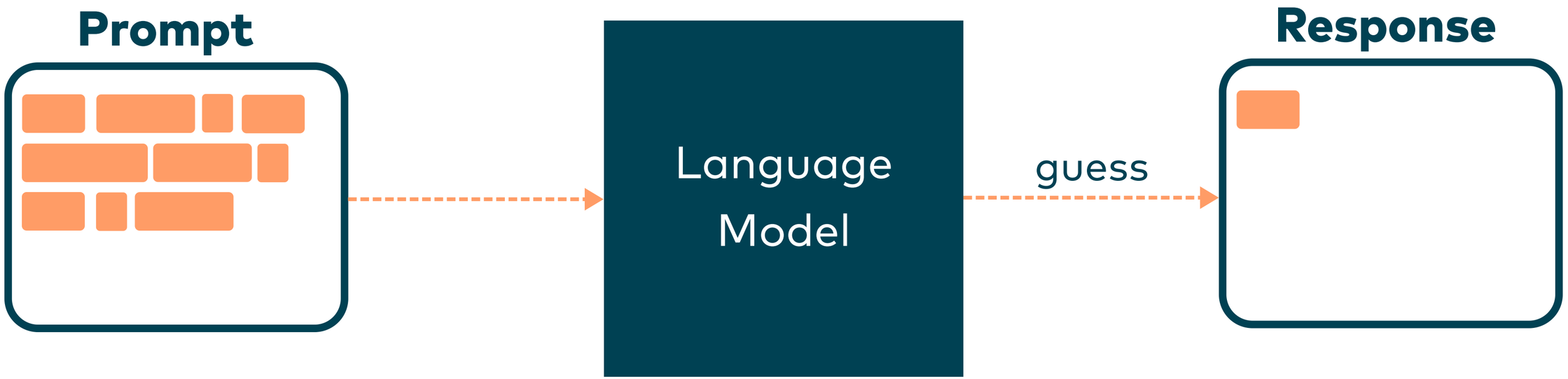
- general-purpose language understanding and generation

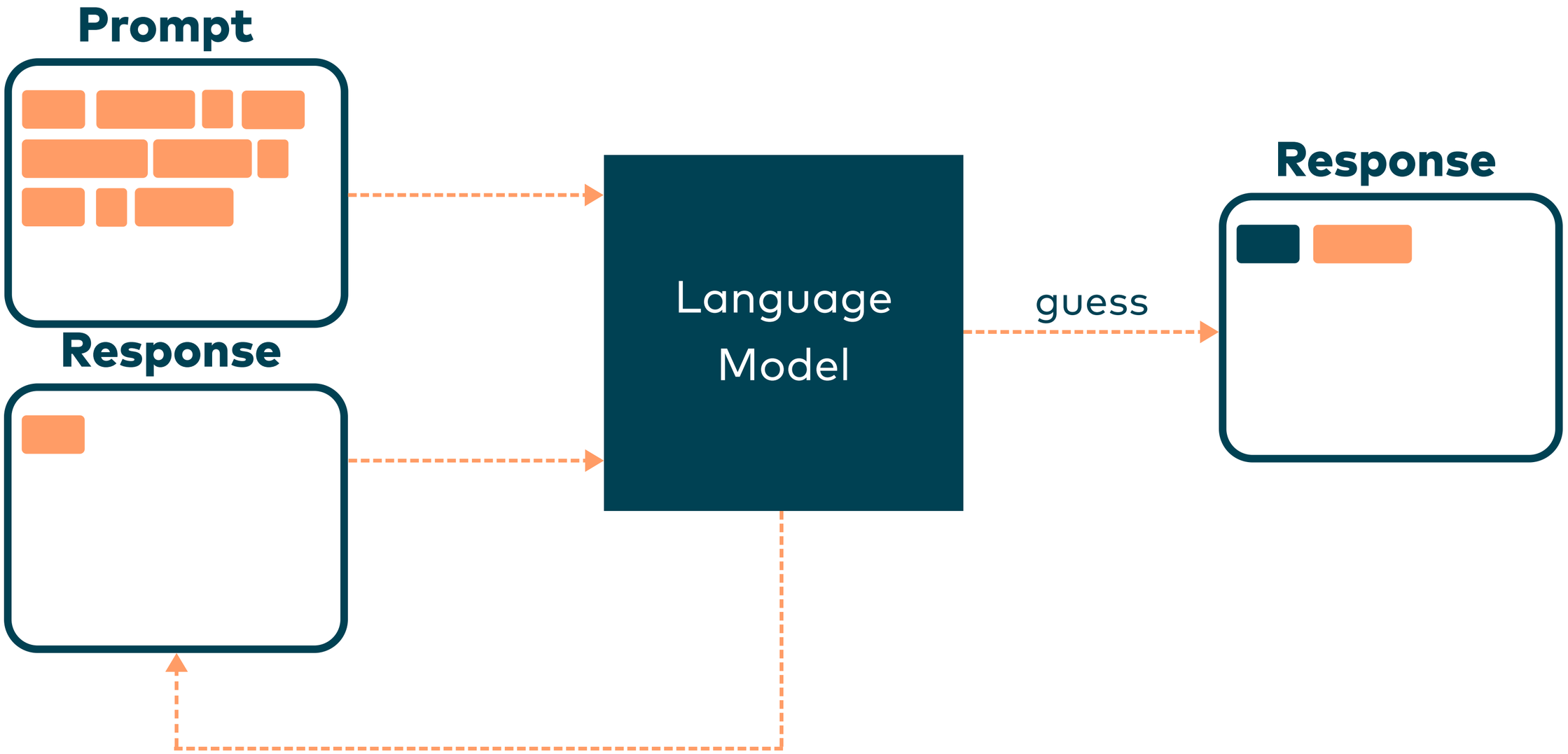


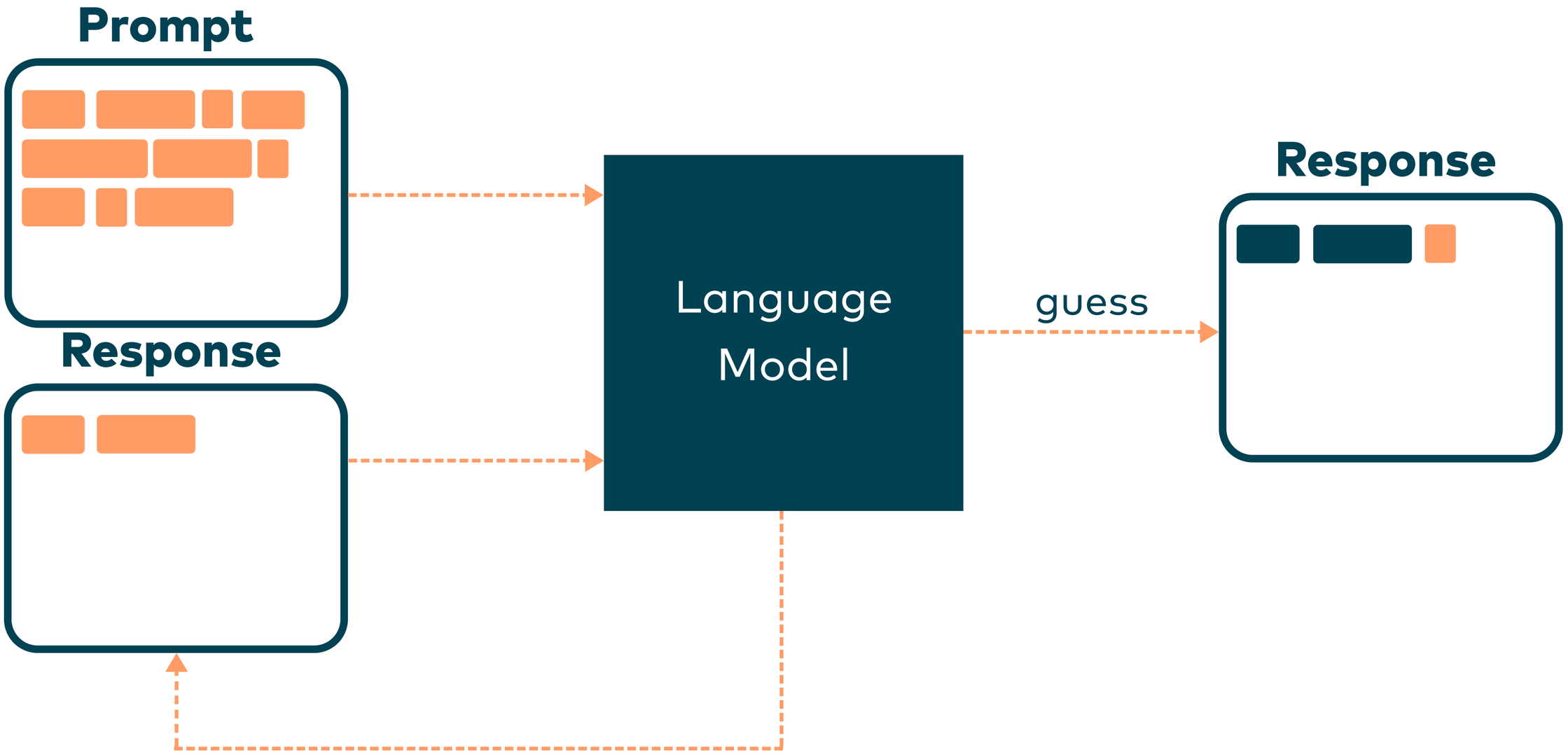
Large Language Model

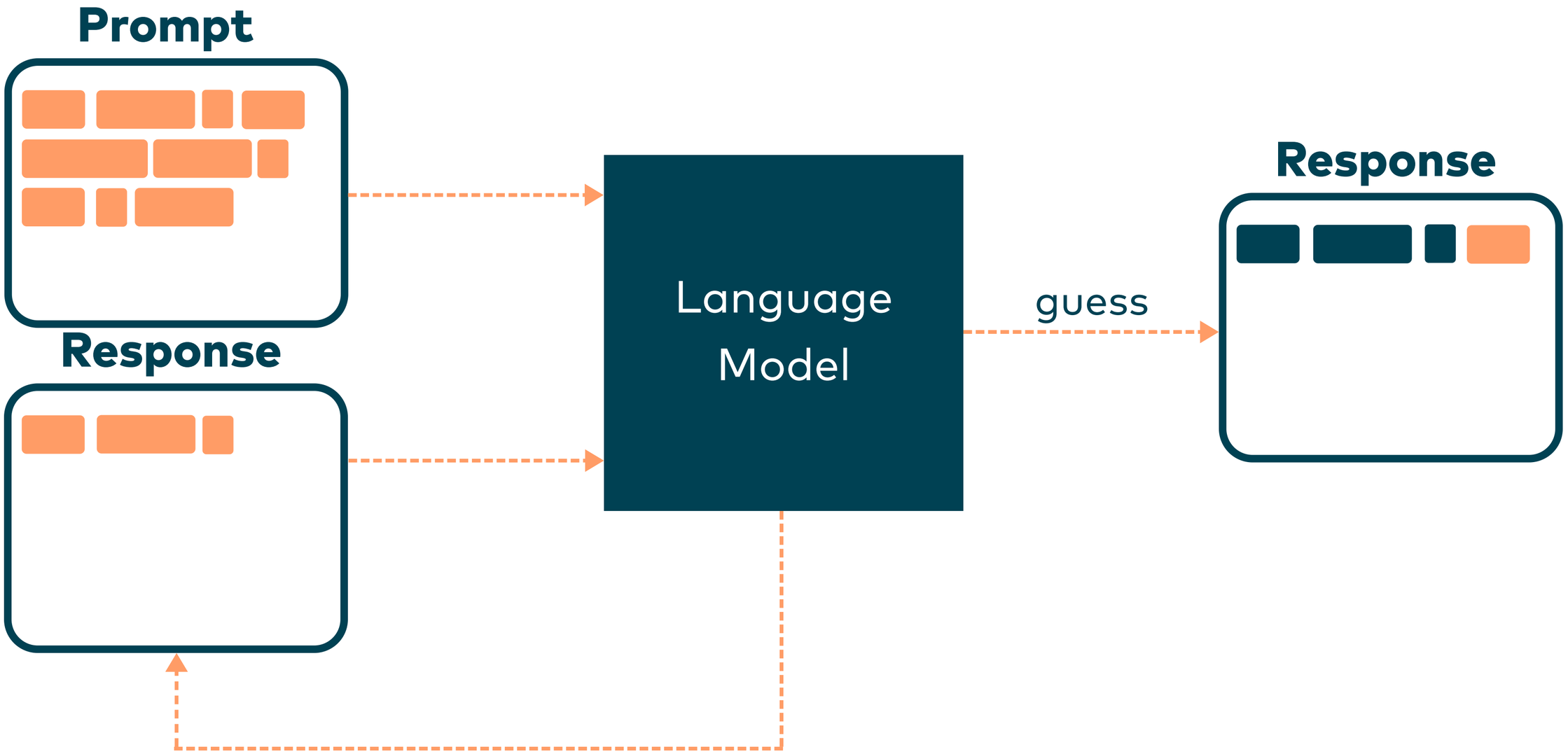
- general-purpose language understanding and generation



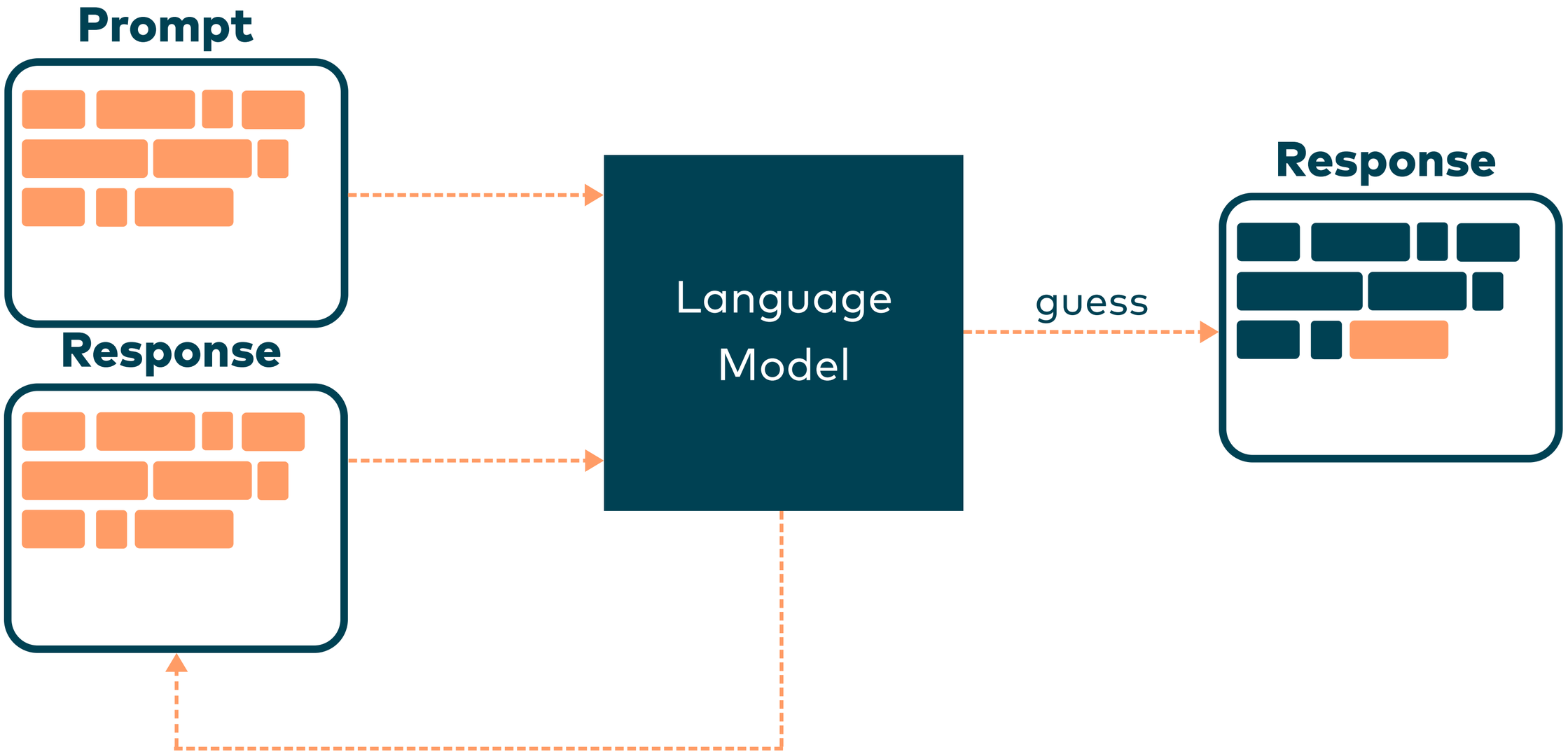

Prompt
Language Model
guess
Response


Prompt
Response
Language Model
guess
Response


Prompt
Response
Language Model
guess
Response


Prompt
Response
Language Model
guess
Response

A few moments later...


Prompt
Response
Language Model
guess
Response


Prompt
Language Model
Response

General-purpose language understanding

General-purpose language understanding

- Does a LLM really understand?

General-purpose language understanding

- Does a LLM really understand?
- LLMs generate answers by guessing it **word by word**

General-purpose language understanding

- Does a LLM really understand?
- LLMs generate answers by guessing it **word by word**
- Answers contain well-written sentences

General-purpose language understanding

- Does a LLM really understand?
- LLMs generate answers by guessing it **word by word**
- Answers contain well-written sentences
- But they were constructed from single words, which were chosen by finding the statistically most-fitting word

General-purpose language understanding

- Does a LLM really understand?
- LLMs generate answers by guessing it **word by word**
- Answers contain well-written sentences
- But they were constructed from single words, which were chosen by finding the statistically most-fitting word
- The LLM does not understand, why one word follows another

It's even more complicated

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🤚

Sequences of characters commonly found next to each other may be grouped together: 1234567890

https://platform.openai.com/tokenizer

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🤚

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Tokens 57

Characters 252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: ������

Sequences of characters commonly found next to each other may be grouped together: 1234567890

https://platform.openai.com/tokenizer

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🤚

Sequences of characters commonly found next to each other may be grouped together: 1234567890

```
[8607, 4339, 2472, 311, 832, 4037, 11, 719, 1063, 1541, 956, 25, 3687,
23936, 382, 35020, 5885, 1093, 100166, 1253, 387, 6859, 1139, 1690,
11460, 8649, 279, 16940, 5943, 25, 11410, 97, 248, 9468, 237, 122, 271,
1542, 45045, 315, 5885, 17037, 1766, 1828, 311, 1855, 1023, 1253, 387,
41141, 3871, 25, 220, 4513, 10961, 16474, 15]
```

https://platform.openai.com/tokenizer

General-purpose language understanding

- LLMs do not generate words, but tokens
- 1 token ~ 4 characters
  ~ 3/4 of a word
- The LLM does not even guess a whole word
- Again less understanding, of why a certain sentence was generated

Context window

- LLMs have a limited context window size

Context window

- LLMs have a limited context window size
- The context is all of the given input

Context window

- LLMs have a limited context window size
- The context is all of the given input
- e.g. GPT3.5-turbo, LLama2 have a maximum context size of 4096 tokens

Context window

- LLMs have a limited context window size
- The context is all of the given input
- e.g. GPT3.5-turbo, LLama2 have a maximum context size of 4096 tokens
- It is called "context window" because it will only use the last 4096 tokens of the given input
- Overhead is cut at the beginning

> Large Language Models. They are everywhere. They get some things amazingly right and other things very interestingly wrong.

Marina Danilevsky

Senior Research Scientist, IBM Research

Powerful as LLMs are, they suffer from several drawbacks

1. "Frozen in time" and lack up-to-date information
2. Lack of domain-specific knowledge
3. "Black boxes"- behavior
4. Costly to produce
5. "Hallucination"

Powerful as LLMs are, they suffer from several drawbacks

1. "Frozen in time" and lack up-to-date information ✔
2. Lack of domain-specific knowledge ✔
3. "Black boxes"- behavior ✔
4. Costly to produce ✔
5. "Hallucination"

OP

How many moons does Jupiter have

Jupiter, the largest planet in our solar system, has at least **79 known moons**. These moons vary considerably in size and properties. The four largest moons of Jupiter, known as the Galilean moons, are Io, Europa, Ganymede and Callisto. These moons were discovered by Galileo Galilei in 1610 and are still some of the best-studied moons in the solar system.

```
M.P.E.C. 2023-D46                             Issued 2023 February 22, 18:35 UT

     The Minor Planet Electronic Circulars contain information on unusual
         minor planets, routine data on comets and natural satellites,
         and occasional editorial announcements.  They are published
   on behalf of Division F of the International Astronomical Union by the
          Minor Planet Center, Smithsonian Astrophysical Observatory,
                          Cambridge, MA 02138, U.S.A.

             Prepared using the Tamkin Foundation Computer Network

                              MPC@CFA.HARVARD.EDU
            URL https://www.minorplanetcenter.net/    ISSN 1523-6714

                                  S/2022 J 3
                          Satellite of Jupiter


Observations:
    SK22J030*.C2022 08 30.17940 00 21 36.321+00 39 50.27         24.0 RVED046807
    SK22J030 .C2022 08 30.18956 00 21 36.057+00 39 48.83         24.1 RVED046807
    SK22J030 .C2022 09 02.18027 00 20 23.936+00 34 02.59         23.9 RVED046807
    SK22J030 .C2022 09 02.19185 00 20 23.668+00 34 01.57         23.9 RVED046807
    SK22J030  C2022 09 18.11539 00 13 08.195-00 01 14.52         24.2 rVED046304
    SK22J030  C2022 09 18.12112 00 13 08.070-00 01 15.15         24.1 rVED046304
    SK22J030  C2022 10 15.14645 00 00 34.254-01 01 33.66         24.2 rVED046304
    SK22J030  C2022 10 15.15702 00 00 34.015-01 01 34.86         24.0 rVED046304
    SK22J030  C2022 10 16.11616 00 00 11.265-01 03 20.03         24.0 rVED046304
    SK22J030  C2022 10 16.12677 00 00 11.015-01 03 21.27         24.0 rVED046304
    SK22J030  C2022 11 17.04785 23 53 09.406-01 30 27.43         24.2 rVED046304
    SK22J030  C2022 11 18.03710 23 53 08.488-01 30 10.22         24.1 rVED046304

Observer details:
304 Las Campanas Observatory.  Observer S. S. Sheppard.  6.5-m Magellan-Baade
    telescope + CCD.
807 Cerro Tololo Observatory, La Serena.  Observer S. S. Sheppard.  4.0-m CTIO
    reflector + CCD.

Orbital elements:
    S/2022 J 3
Epoch 2023 Feb. 25.0 TT = JDT 2460000.5                 Alexandersen
M  77.88890              (2000.0)            P               Q
n   0.58269615     Peri.   45.24901     -0.24065155     -0.81255434
a   0.1397905      Node   294.05598     -0.97016626     +0.21793735
e   0.2721788      Incl.  144.45215     +0.02939829     +0.54060962
P   1.69           H   17.4            P/d    617.82
From 12 observations 2022 Aug. 30-2022 Nov. 18, mean residual 0".17

Residuals in seconds of arc
220830 807  0.2+  0.0      220918 304  0.4-  0.0      221016 304  0.2-  0.0
220830 807  0.1-  0.3-     220918 304  0.3+  0.2+     221016 304  0.1-  0.1-
220902 807  0.3-  0.1-     221015 304  0.0   0.0      221117 304  0.0   0.0
220902 807  0.1+  0.2+     221015 304  0.3+  0.0      221118 304  0.0   0.0


M. P. C. Staff              (C) Copyright 2023 MPC          M.P.E.C. 2023-D46
```

Jupiter has **95** moons

International Astronomical Union
Minor Planet Centre (MPC)
22 February 2023

Hallucination

- In more complex technical questions, sometimes LLMs not only can't give a good answer but may also come up with a convincing-sounding but ultimately wrong response.

Hallucination

- In more complex technical questions, sometimes LLMs not only can't give a good answer but may also come up with a convincing-sounding but ultimately wrong response.
- We can not trace which parameters / data caused the LLM to give the wrong answer

Hallucination

- In more complex technical questions, sometimes LLMs not only can't give a good answer but may also come up with a convincing-sounding but ultimately wrong response.
- We can not trace which parameters / data caused the LLM to give the wrong answer

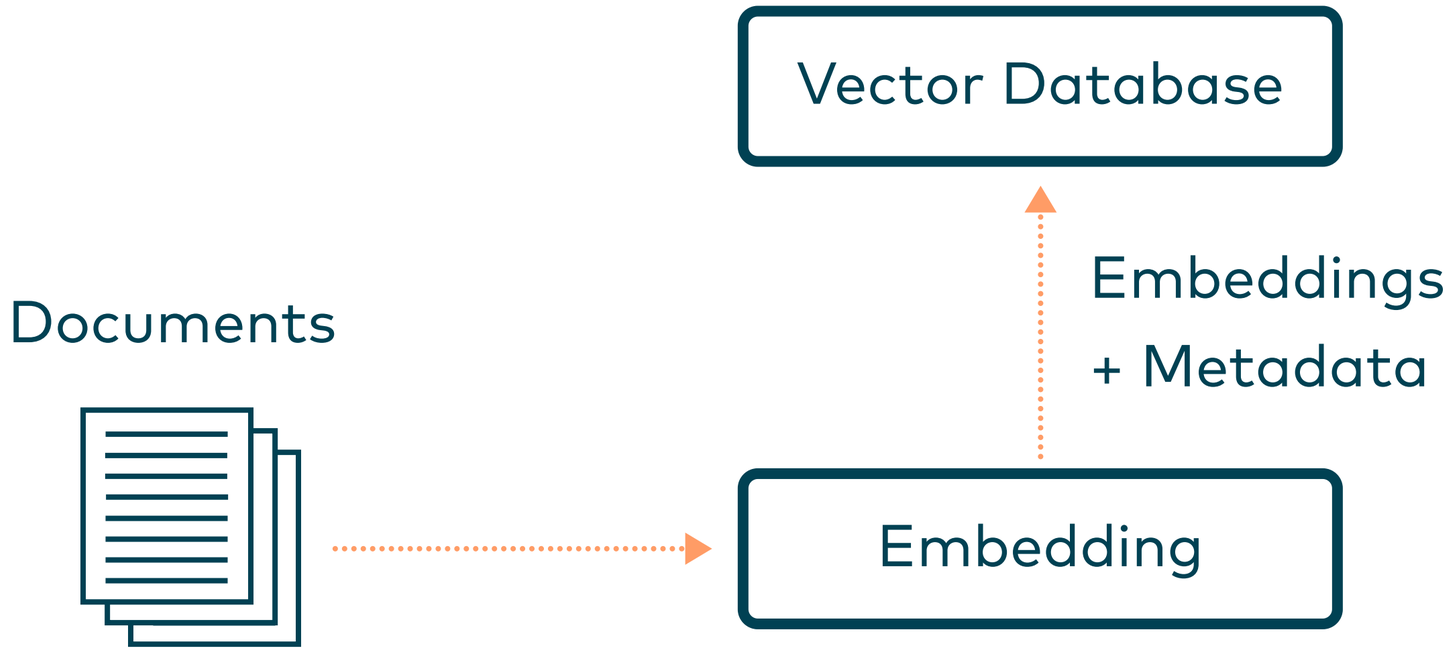
→ Provide your own data with sources to the LLM

RAG

Retrieval augmented generation

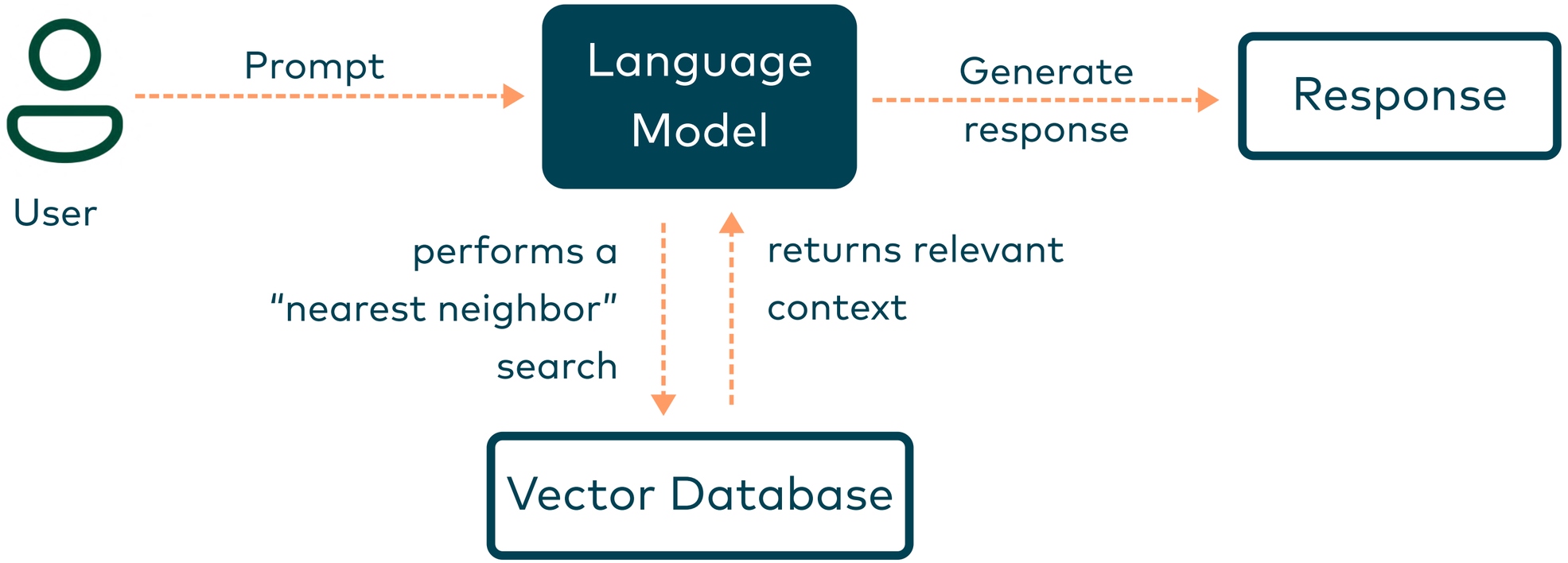
Retrieval Augmented Generation

- Retrieval Augmented Generation (RAG) is a technique that involves fetching up-to-date or context-specific data from an external database and making it available to a Large Language Model during the generation process.



Retrieval Augmented Generation

- By storing proprietary business data or information about the world, you can have your application retrieve this data when generating a response. This helps reduce the likelihood of generating inaccurate or unreliable information.



Chunking

Chunking

- Load complete dataset from a datasource:
  - website
  - database
  - documents

Chunking

- Load complete dataset from a datasource:
  - website
  - database
  - documents
- Divide data into chunks with a chosen maximum length

Chunking

- Load complete dataset from a datasource:
  - website
  - database
  - documents
- Divide data into chunks with a chosen maximum length

Chunking

- Load complete dataset from a datasource:
  - website
  - database
  - documents
- Divide data into chunks with a chosen maximum length
- For each chunk always remember the source (filepath, URL, paragraph and line)

MICHAEL VITZ

Dieser Artikel ist auch auf Deutsch verfügbar

The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the world. This also means that the features and changes from JDK 18, JDK 19 and JDK 20 will now be increasingly incorporated into our applications.

But wait a minute, why is there another LTS release after just two years? Wasn't the plan every three years? Yes, that was the plan until Oracle proposed along with the release of JDK 17 to adopt a two year cadence. Since all other relevant developers have agreed to follow this proposal, we now have a new release with at least five years of support after just two years, even though there will be yet another new version in two years time in the form of JDK 25.

It is, in fact, possible to update to the newest version of the JDK every six months, but it often makes sense to move from LTS to LTS release in order to somewhat slow the frequency of new features and enjoy greater stability. For precisely this reason, the present article offers an overview of the new features added since JDK 17 to show why it is worth upgrading to the new LTS release. This is not strictly necessary, however, since support for JDK 17 will continue for several more years.

Before we get started, it is worth noting that alongside final, and therefore stable, features, we now also have incubator (JEP11) and preview features (JEP12). Both are less stable and could change significantly in their final version. In the case of incubator features, there is even the risk that they may be removed before ever making it to a final version. Plus, to ensure that we do not unintentionally make use of unstable preview features, we must activate these both during compilation and at runtime by additionally including `--enable-preview`.

https://www.innoq.com/en/articles/2023/10/java-21/

MICHAEL VITZ

Dieser Artikel ist auch auf Deutsch verfügbar

The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the world. This also means that the features and changes from JDK 18, JDK 19 and JDK 20 will now be increasingly incorporated into our applications.

But wait a minute, why is there another LTS release after just two years? Wasn't the plan every three years? Yes, that was the plan until Oracle proposed along with the release of JDK 17 to adopt a two year cadence. Since all other relevant developers have agreed to follow this proposal, we now have a new release with at least five years of support after just two years, even though there will be yet another new version in two years time in the form of JDK 25.

It is, in fact, possible to update to the newest version of the JDK every six months, but it often makes sense to move from LTS to LTS release in order to somewhat slow the frequency of new features and enjoy greater stability. For precisely this reason, the present article offers an overview of the new features added since JDK 17 to show why it is worth upgrading to the new LTS release. This is not strictly necessary, however, since support for JDK 17 will continue for several more years.

Before we get started, it is worth noting that alongside final, and therefore stable, features, we now also have incubator (JEP11) and preview features (JEP12). Both are less stable and could change significantly in their final version. In the case of incubator features, there is even the risk that they may be removed before ever making it to a final version. Plus, to ensure that we do not unintentionally make use of unstable preview features, we must activate these both during compilation and at runtime by additionally including `--enable-preview`.

https://www.innoq.com/en/articles/2023/10/java-21/

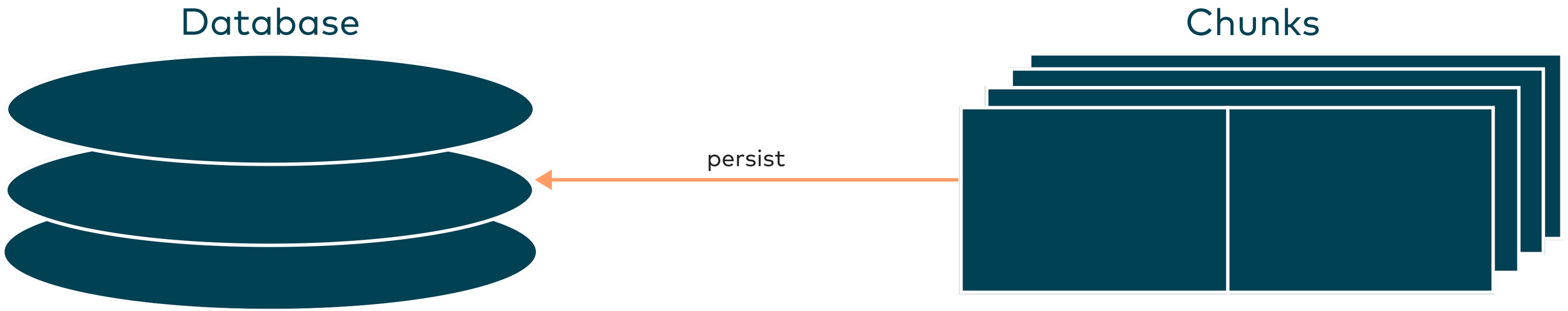

Chunk
text
source

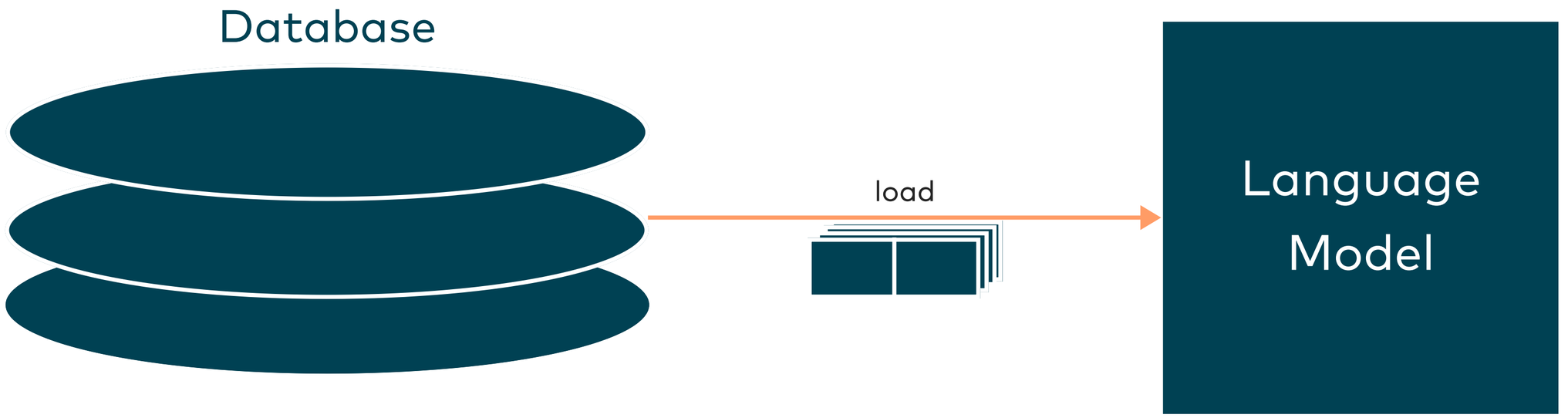
Chunk

The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the...	innoq.com/en/articles/2023/10/java-21/

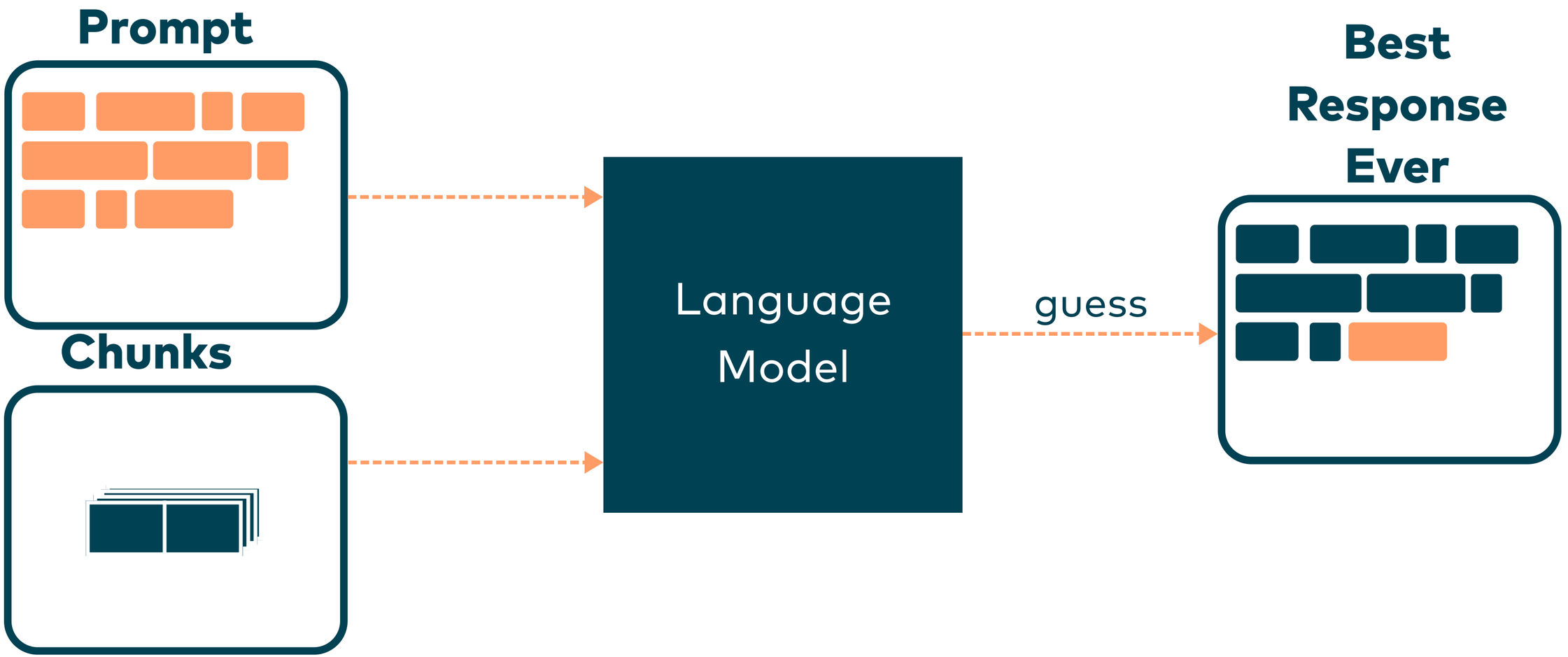
Chunks

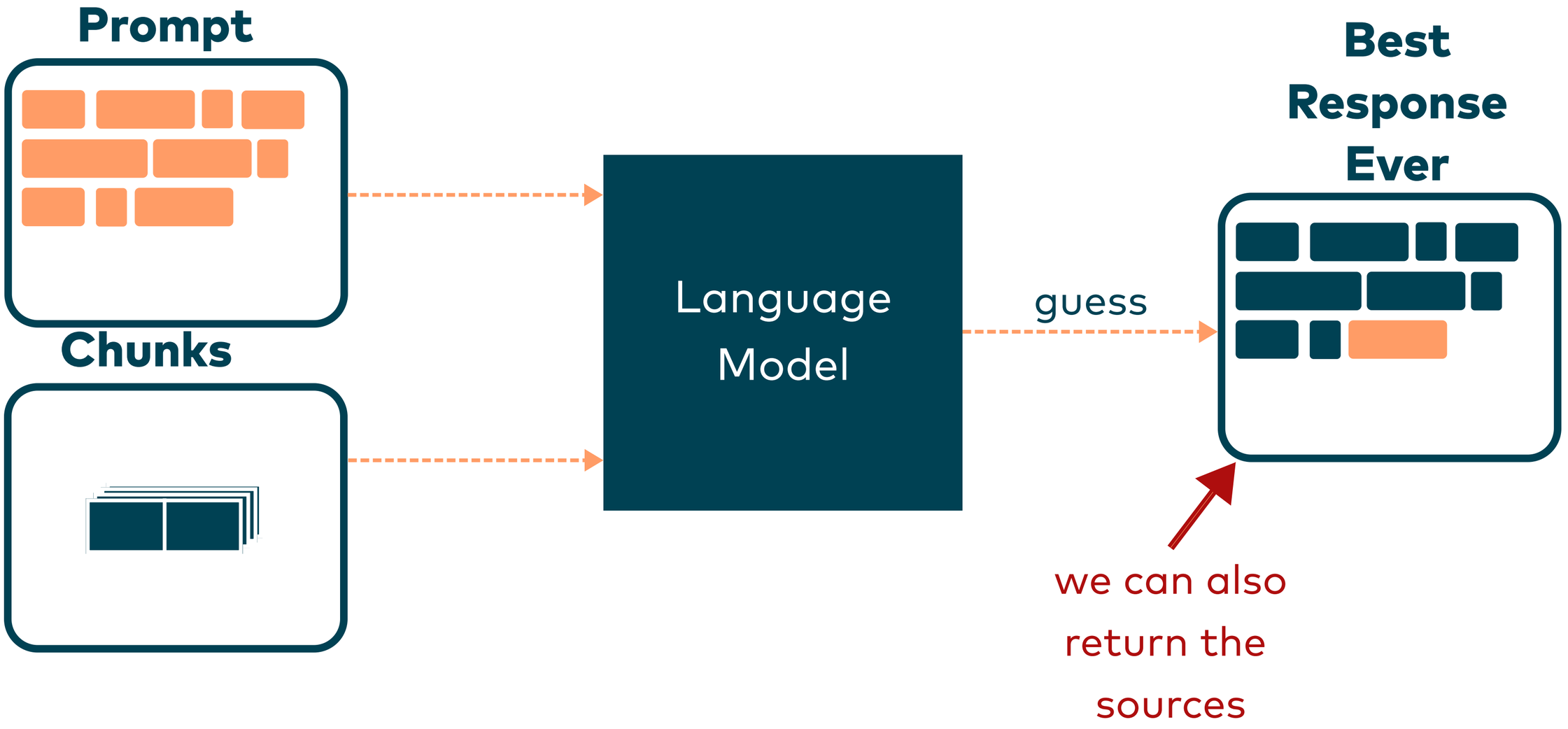
How to provide the chunks to our LLM?

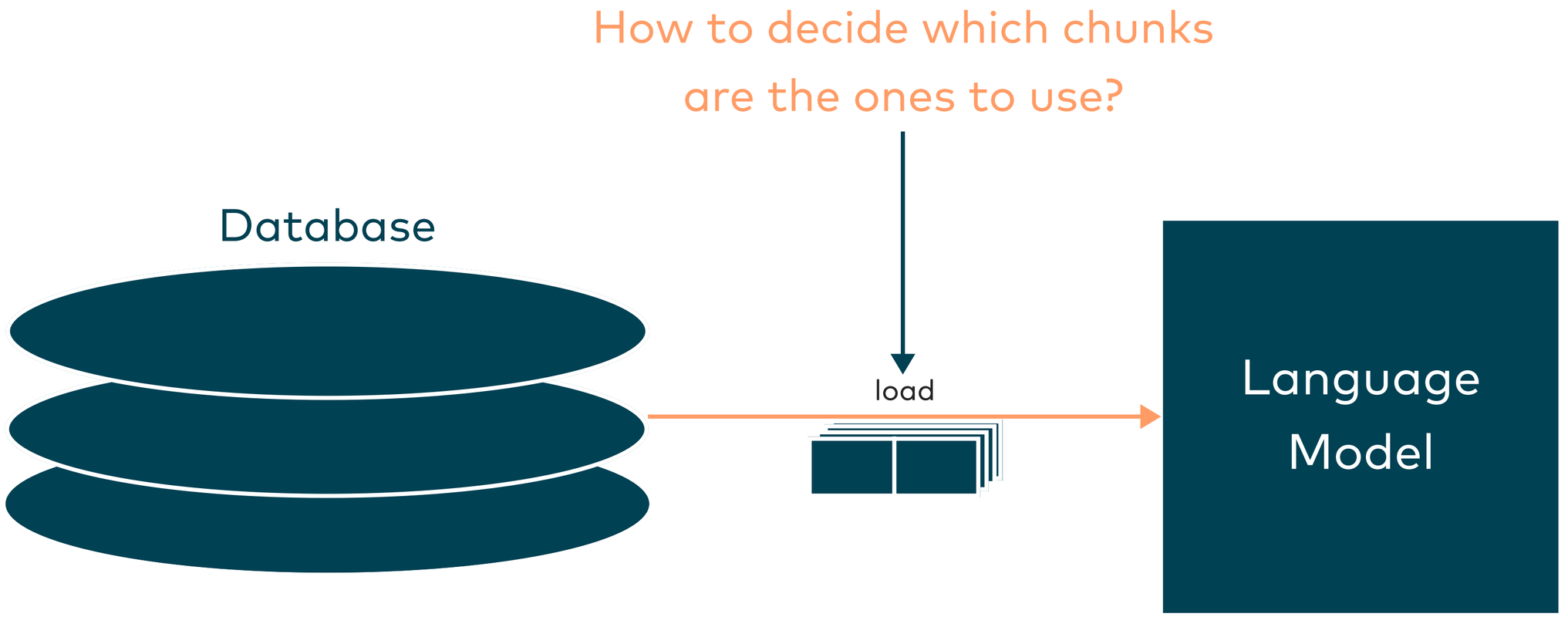

Database
persist
Chunks


Database
load
Language
Model

A few moments later...


Prompt
Chunks
Language Model
guess
Best Response Ever


Prompt
Chunks
Language Model
guess
Best Response Ever
we can also return the sources

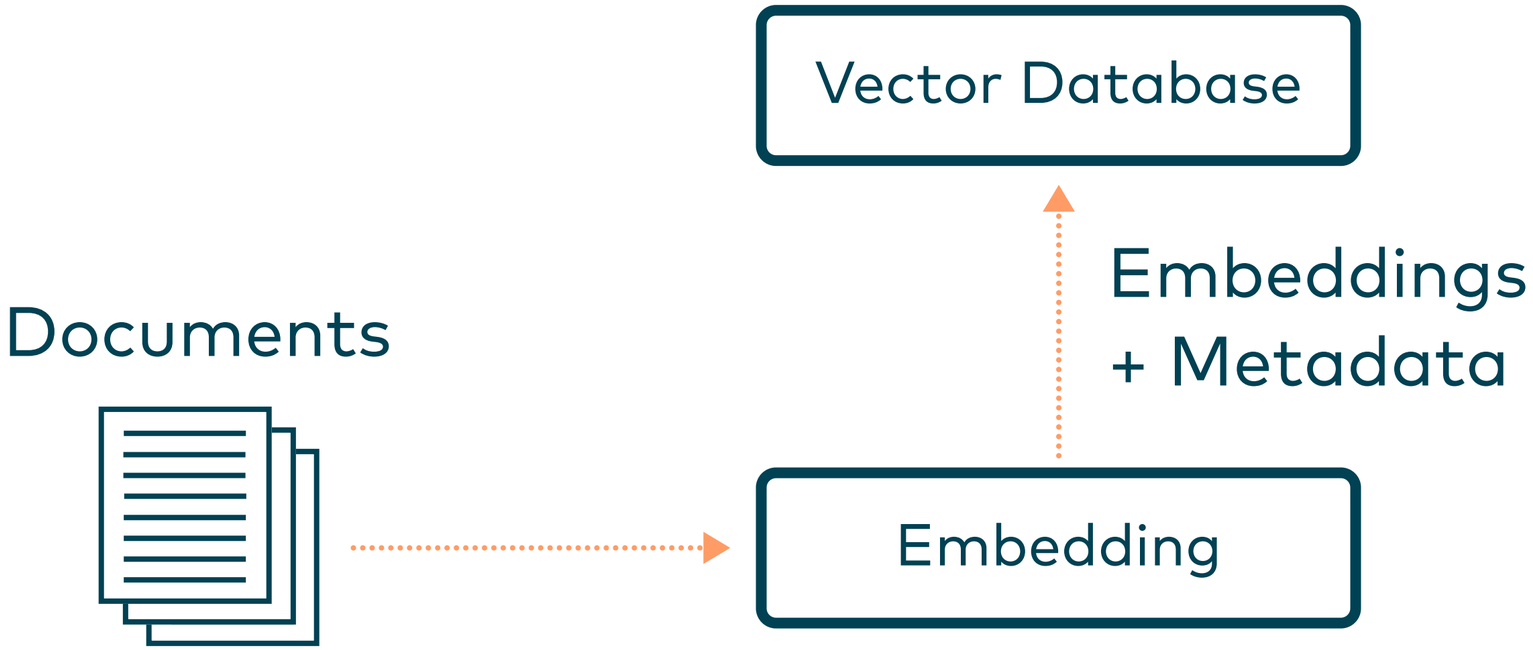

How to decide which chunks
are the ones to use?
Database
load
Language
Model

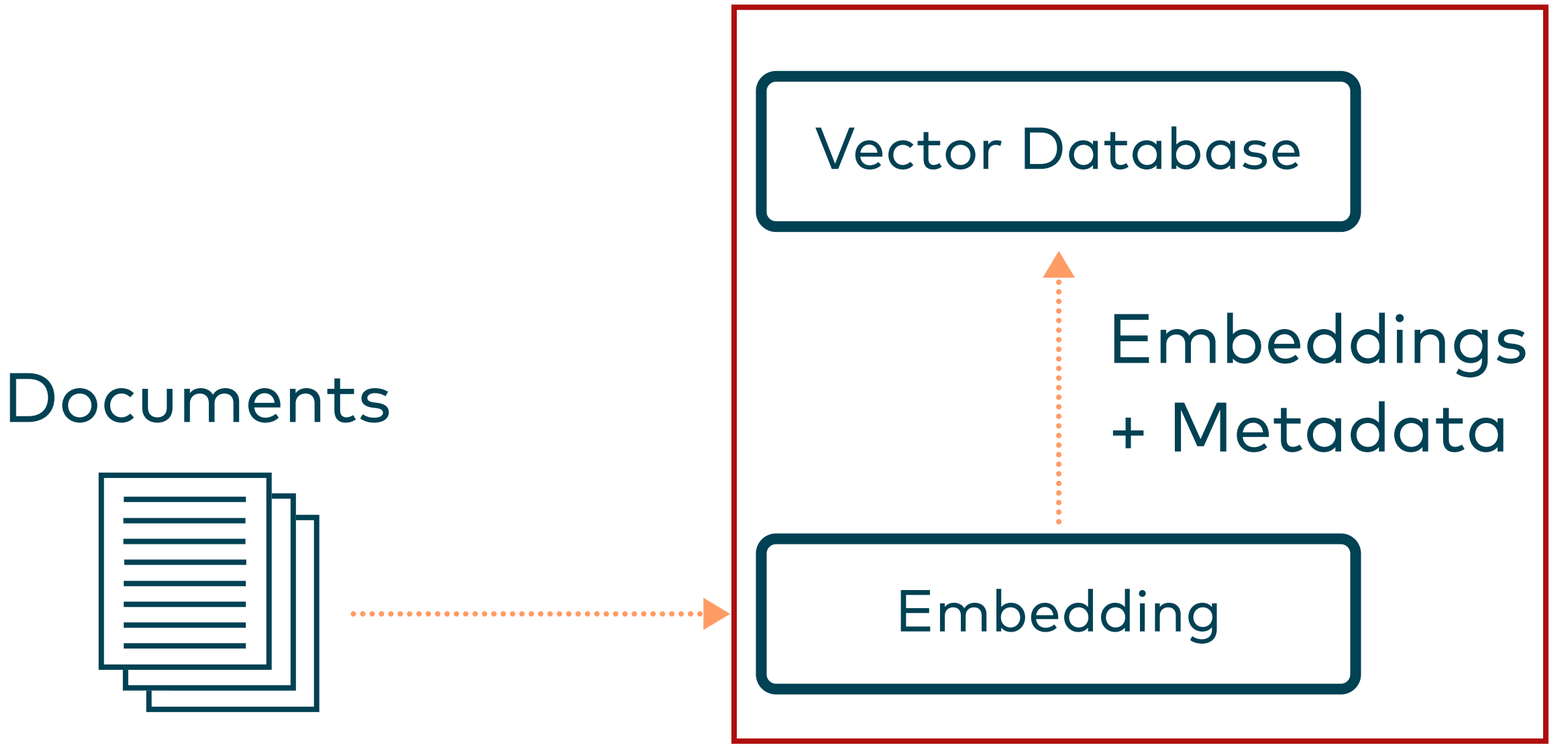
Retrieval Augmented Generation

Embedding

~~Embedding~~

Numerical representation of context


Vector Database
Embeddings
+ Metadata
Documents
Embedding


Vector Database
Embeddings
+ Metadata
Documents
Embedding

Vector representation

- Includes a representation of the given context
- It is not required to understand the elements of the vector representation
- Used to guess a single token
- And to make chunks comparable

MICHAEL VITZ

Dieser Artikel ist auch auf Deutsch verfügbar

The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the world. This also means that the features and changes from JDK 18, JDK 19 and JDK 20 will now be increasingly incorporated into our applications.

→ [0.1823313375201503, 0.16225175989, 0.8865212883473177, ...]

But wait a minute, why is there another LTS release after just two years? Wasn't the plan every three years? Yes, that was the plan until Oracle proposed along with the release of JDK 17 to adopt a two year cadence. Since all other relevant developers have agreed to follow this proposal, we now have a new release with at least five years of support after just two years, even though there will be yet another new version in two years time in the form of JDK 25.

→ [0.83278322, 0.923893278, 0.127387283, ...]

It is, in fact, possible to update to the newest version of the JDK every six months, but it often makes sense to move from LTS to LTS release in order to somewhat slow the frequency of new features and enjoy greater stability. For precisely this reason, the present article offers an overview of the new features added since JDK 17 to show why it is worth upgrading to the new LTS release. This is not strictly necessary, however, since support for JDK 17 will continue for several more years.

→ [0.6748728378, 0.4728378283, 0.3941673, ...]

Before we get started, it is worth noting that alongside final, and therefore stable, features, we now also have incubator (JEP11) and preview features (JEP12). Both are less stable and could change significantly in their final version. In the case of incubator features, there is even the risk that they may be removed before ever making it to a final version. Plus, to ensure that we do not unintentionally make use of unstable preview features, we must activate these both during compilation and at runtime by additionally including `--enable-preview`.

→ [0.192301503, 0.4578989023, 0.328903302, ...]

https://www.innoq.com/en/articles/2023/10/java-21/

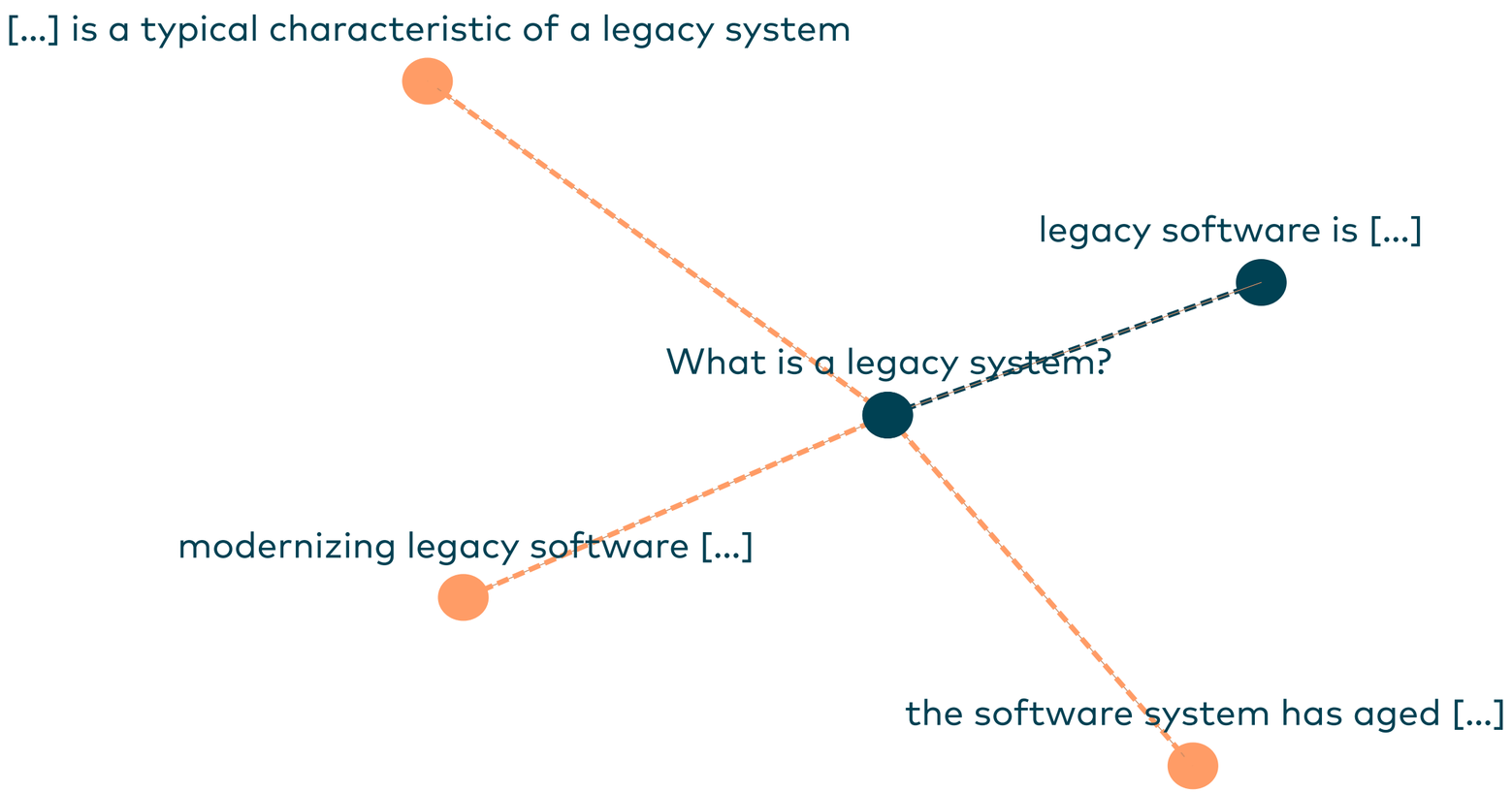
[...] is a typical characteristic of a legacy system

legacy software is [...]

What is a legacy system?

modernizing legacy software [...]

the software system has aged [...]


[...] is a typical characteristic of a legacy system
legacy software is [...]
What is a legacy system?
modernizing legacy software [...]
the software system has aged [...]

Indexing

Vector Database

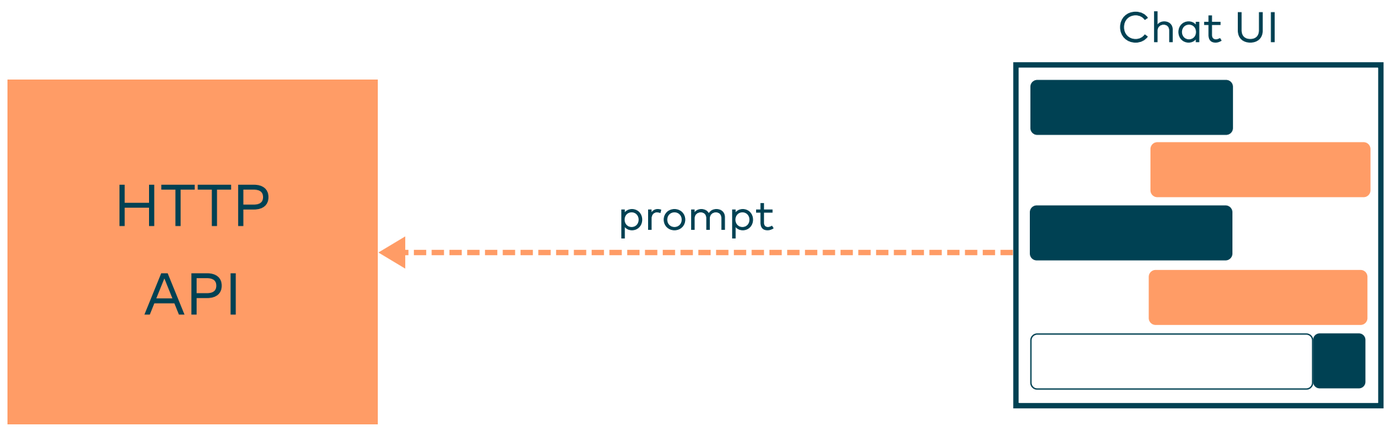
- Store the vector representations
- Optimized index for vector distance
- Query for a vector
- The database will return the N **approximate nearest neighbours**

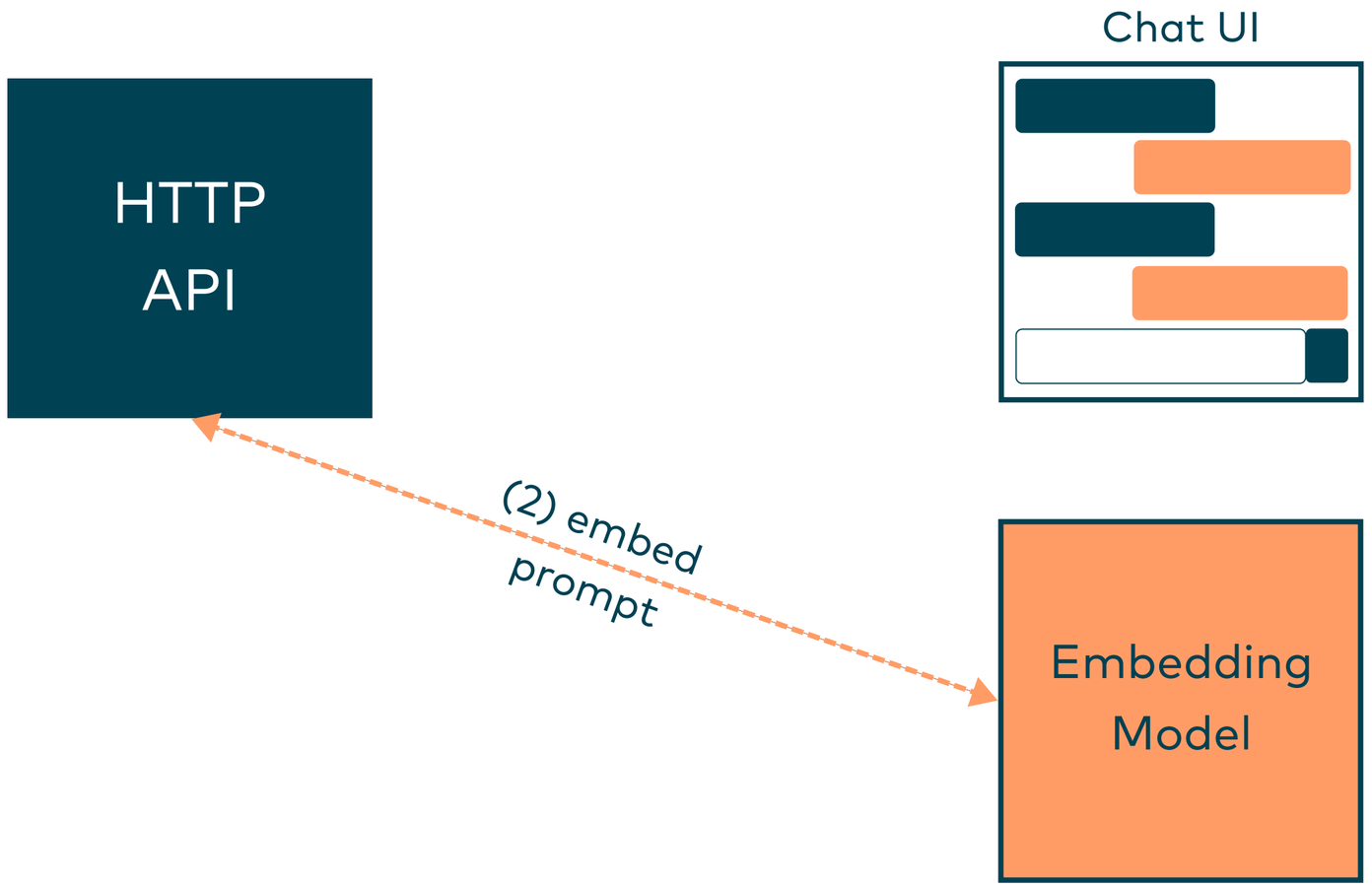
vector representation	chunk
[0.1823313375201503, 0.16225175989, 0.8865212883473177, ...]	The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the world. This also means that the features and changes from JDK 18, JDK 19 and JDK 20 will now be increasingly incorporated into our applications.
[0.83278322, 0.923893278, 0.127387283, ...]	But wait a minute, why is there another LTS release after just two years? Wasn't the plan every three years? Yes, that was the plan until Oracle proposed along with the release of JDK 17 to adopt a two year cadence.
...	...

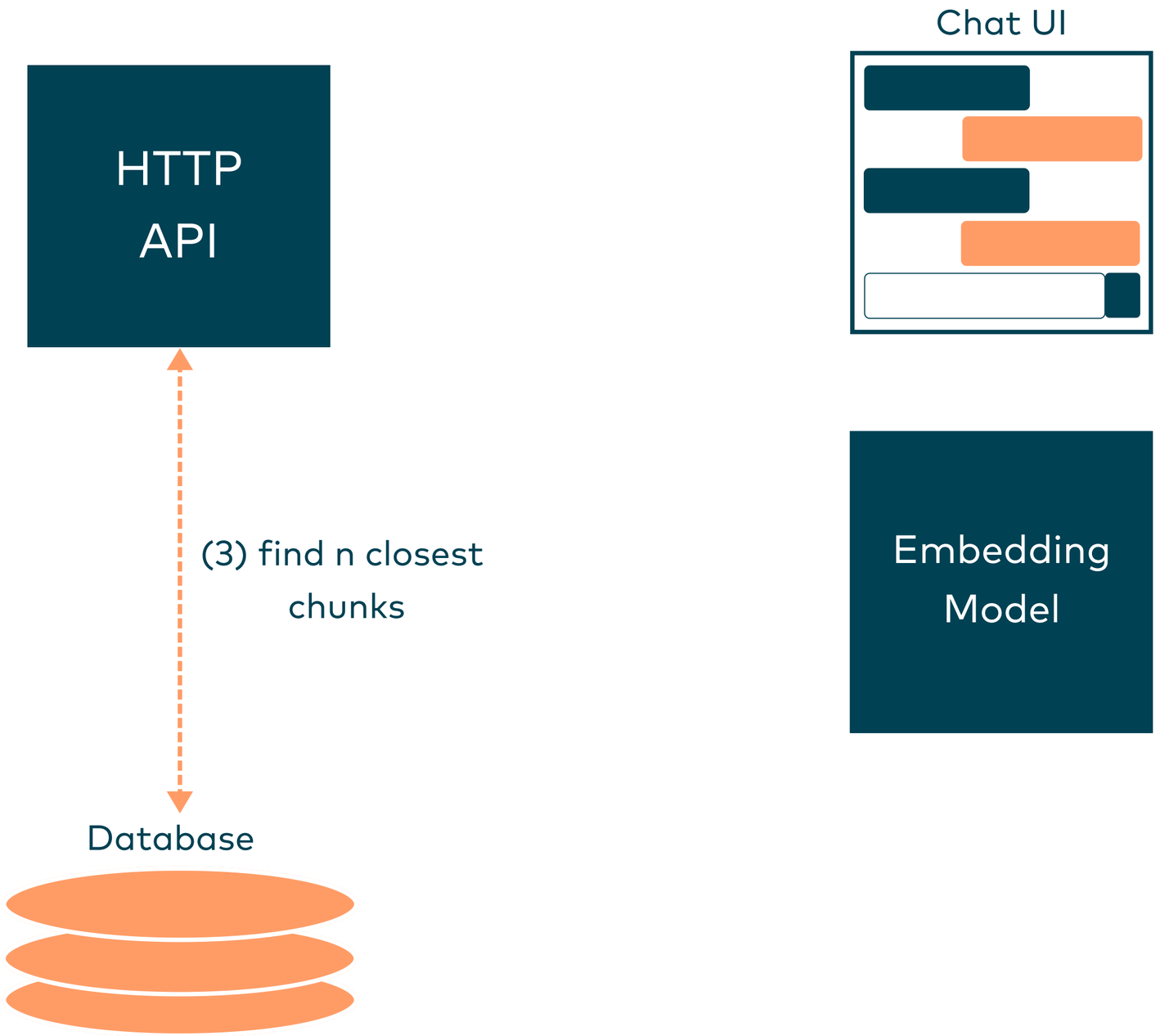
Index

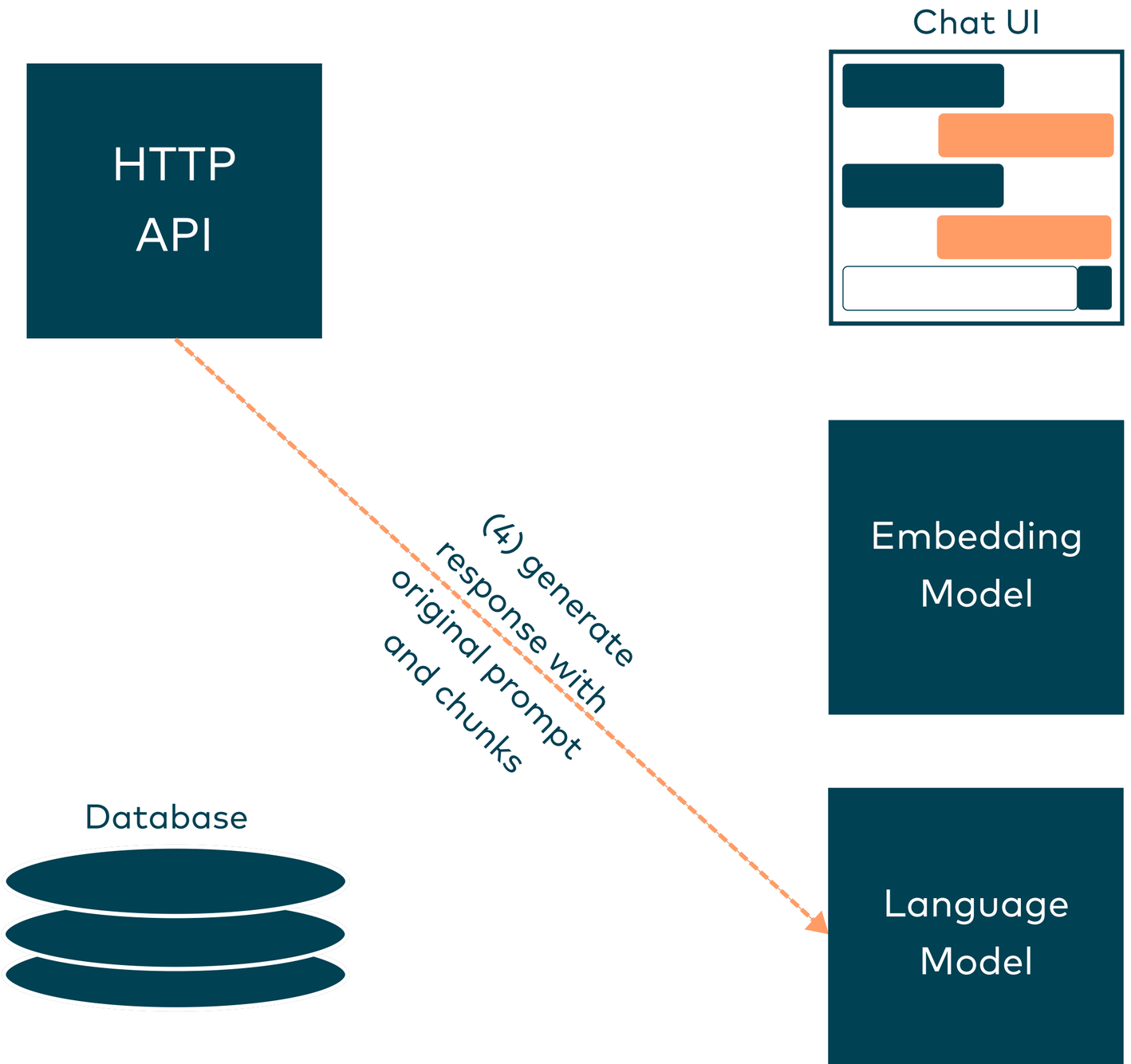
vector representation	chunk
[0.1823313375201503, 0.16225175989, 0.8865212883473177, ...]	The time has finally come, on September 19th, JDK 21, the newest long-term support (LTS) release after JDK 17, has come forth into the light of the world. This also means that the features and changes from JDK 18, JDK 19 and JDK 20 will now be increasingly incorporated into our applications.
[0.83278322, 0.923893278, 0.127387283, ...]	But wait a minute, why is there another LTS release after just two years? Wasn't the plan every three years? Yes, that was the plan until Oracle proposed along with the release of JDK 17 to adopt a two year cadence.
...	...

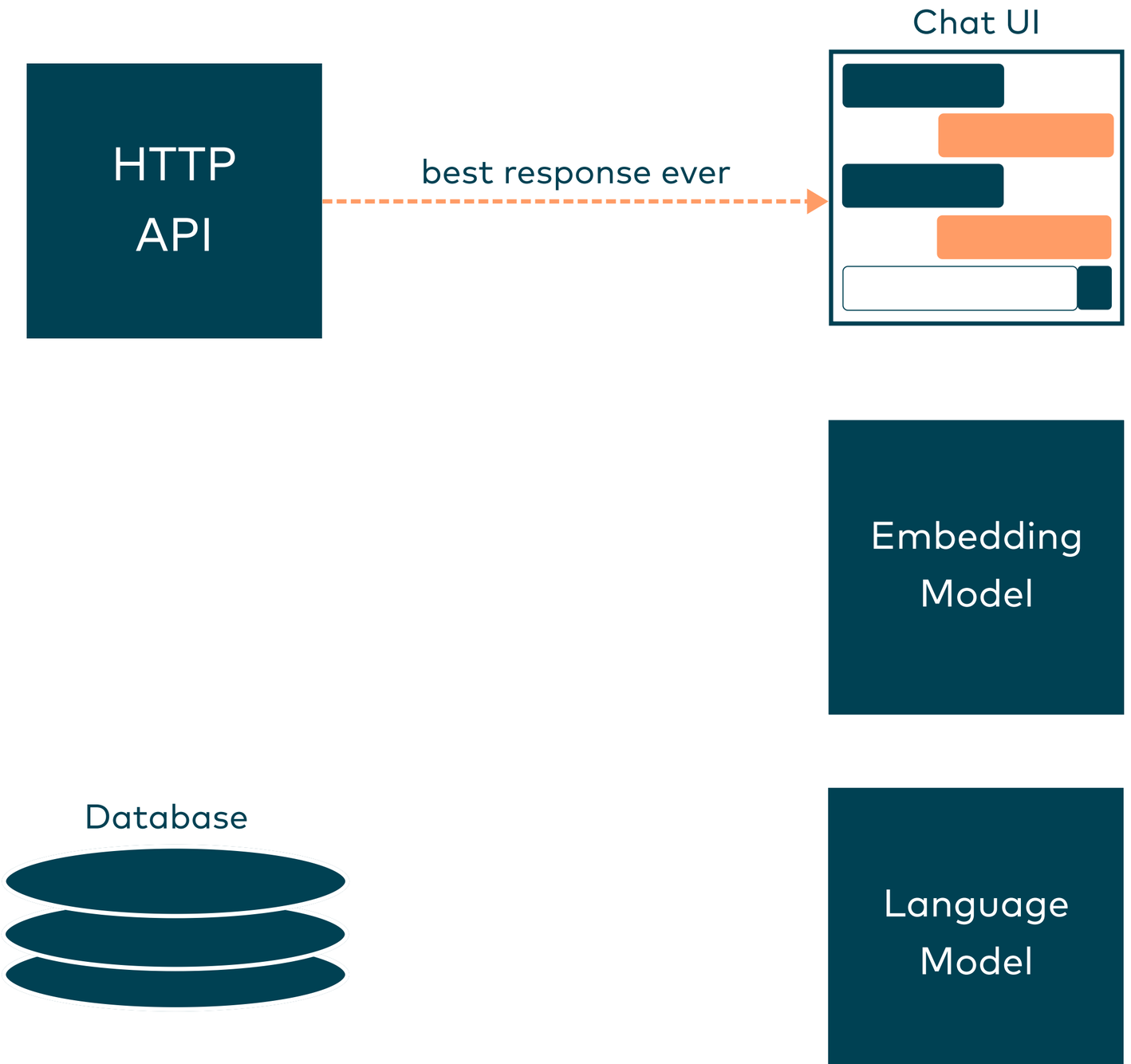
Bringing all parts together

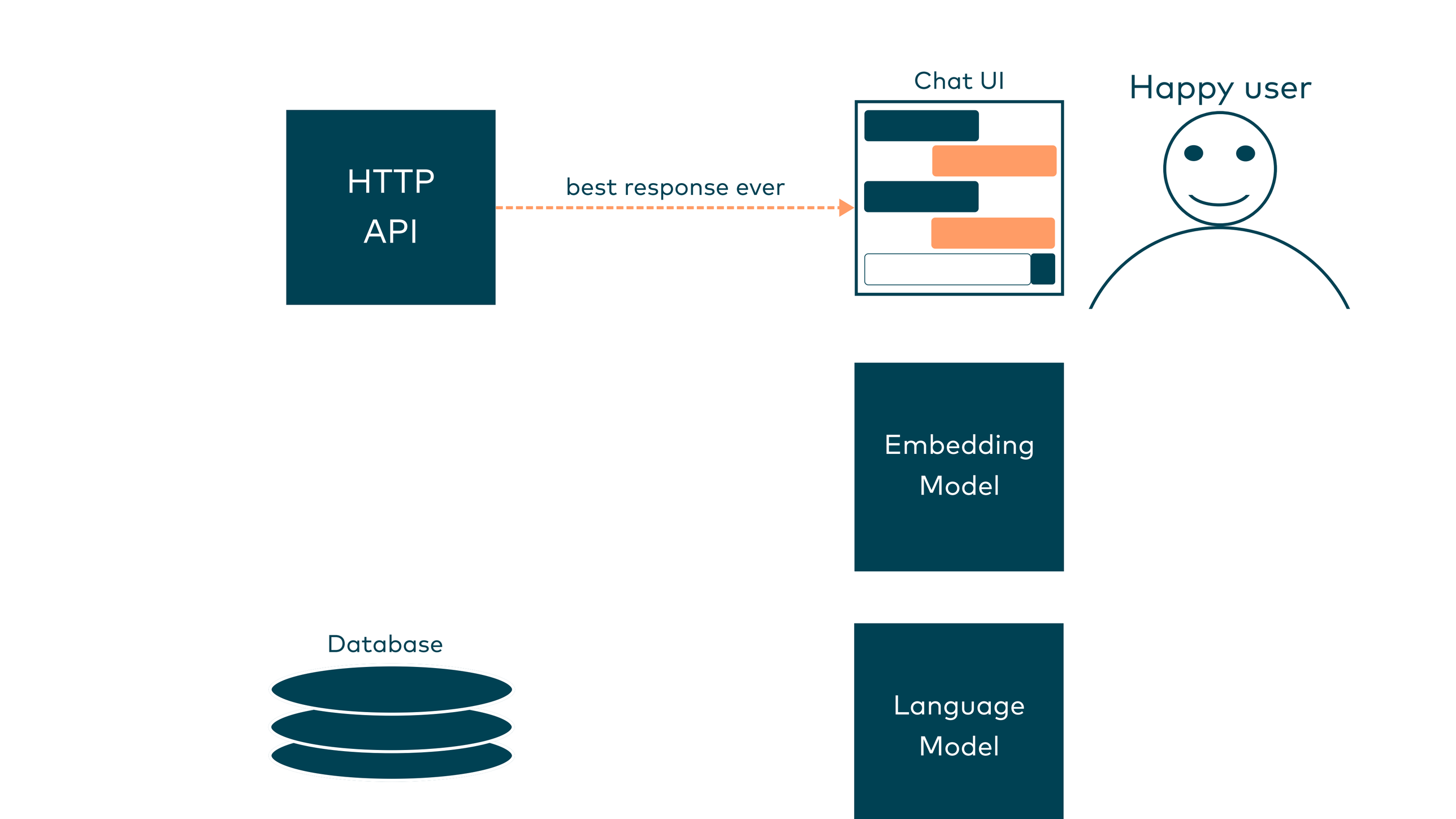

Chat UI
HTTP
API
prompt

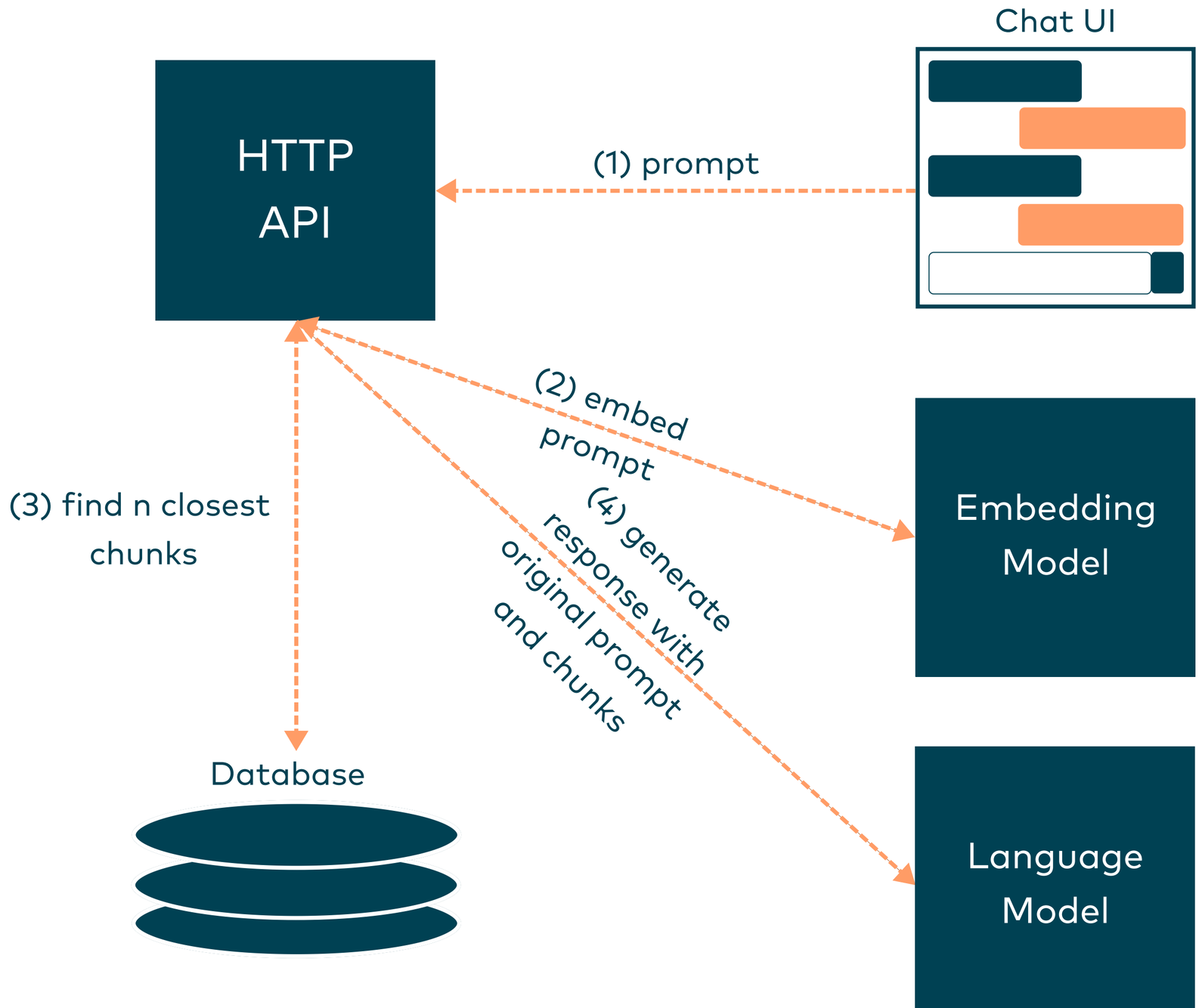

Chat UI
HTTP
API
(2) embed
prompt
Embedding
Model

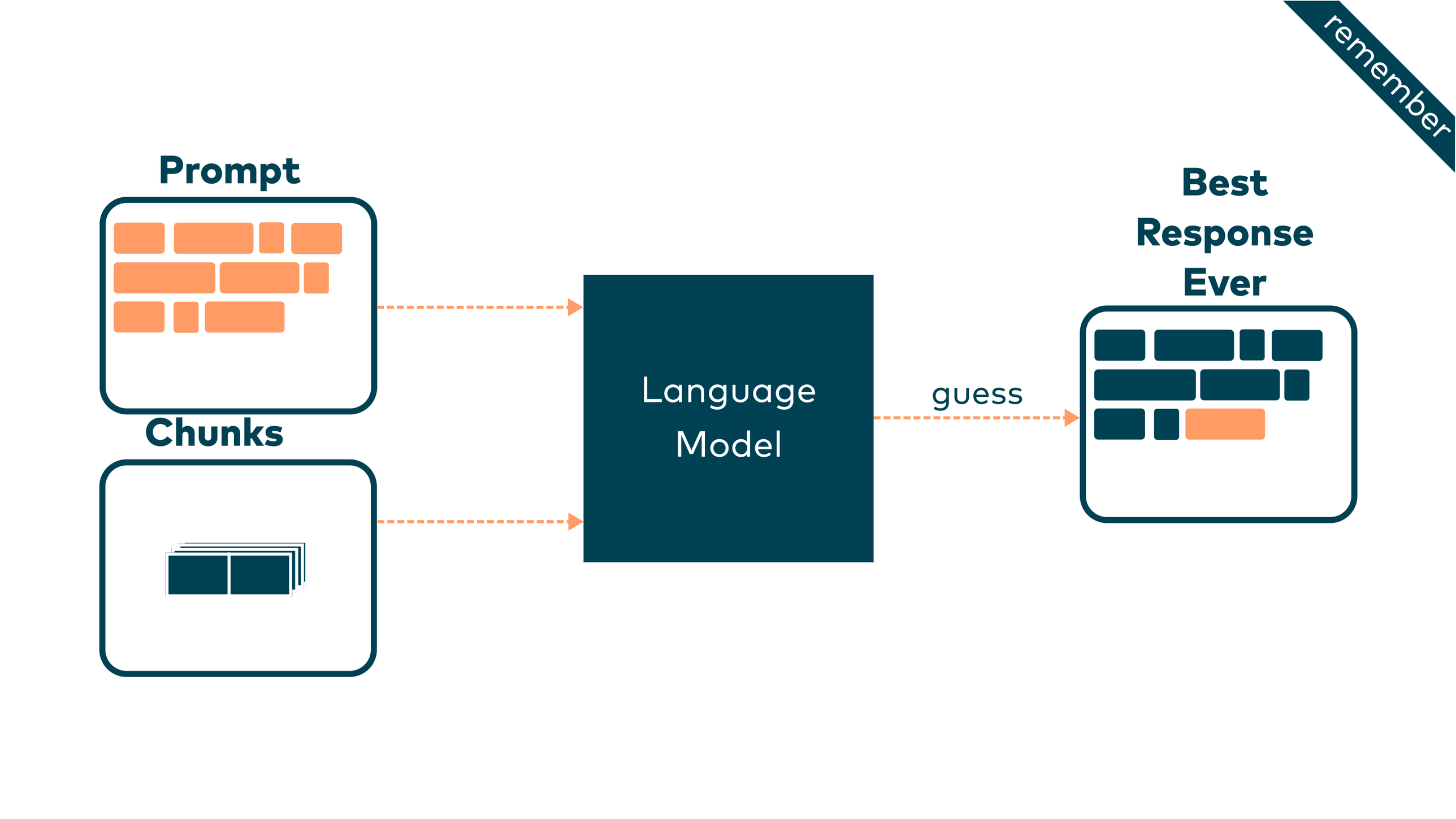

Chat UI
HTTP
API
(3) find n closest
chunks
Embedding
Model
Database


Chat UI
HTTP
API
(4) generate
response with
original prompt
and chunks
Embedding
Model
Database
Language
Model

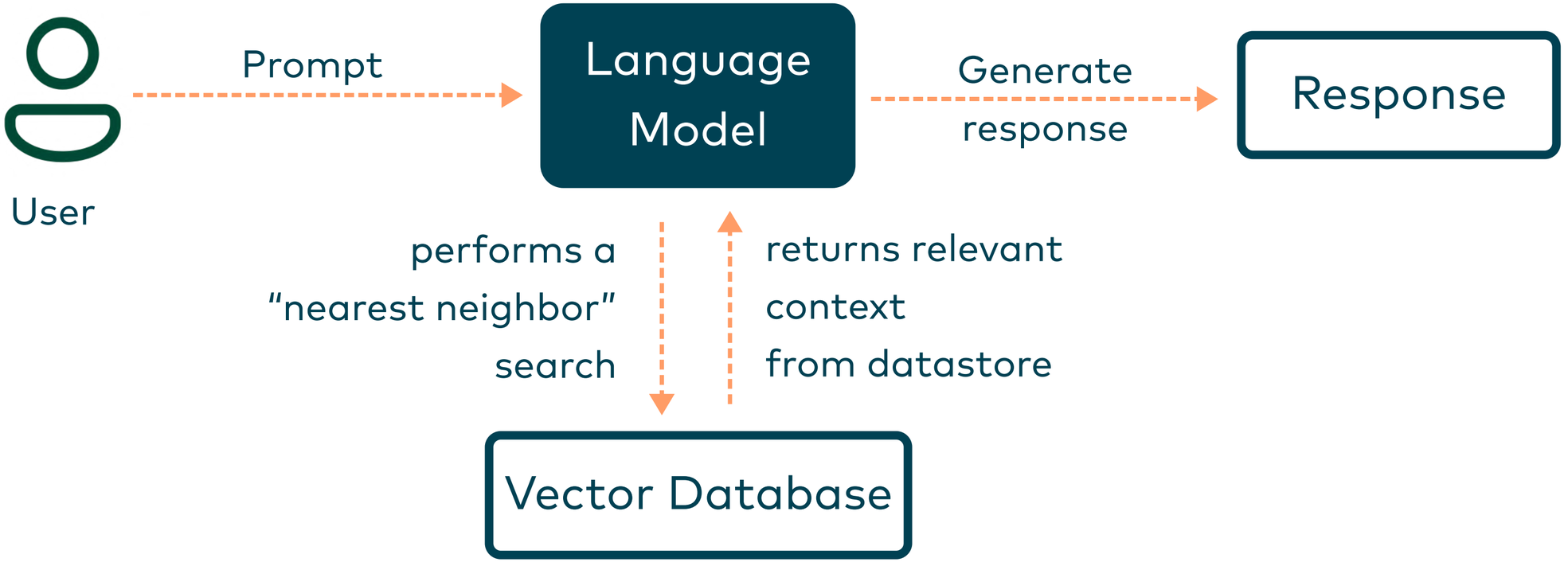

Chat UI
HTTP
API
best response ever
Embedding
Model
Database
Language
Model


HTTP
API
best response ever
Chat UI
Happy user
Embedding
Model
Database
Language
Model


Chat UI
HTTP API
(1) prompt
(2) embed prompt
(3) find n closest chunks
(4) generate response with original prompt and chunks
Embedding Model
Database
Language Model


remember
Prompt
Chunks
Language Model
guess
Best Response Ever

remember


User
Prompt
Language Model
Generate response
Response
performs a "nearest neighbor" search
returns relevant context from datastore
Vector Database

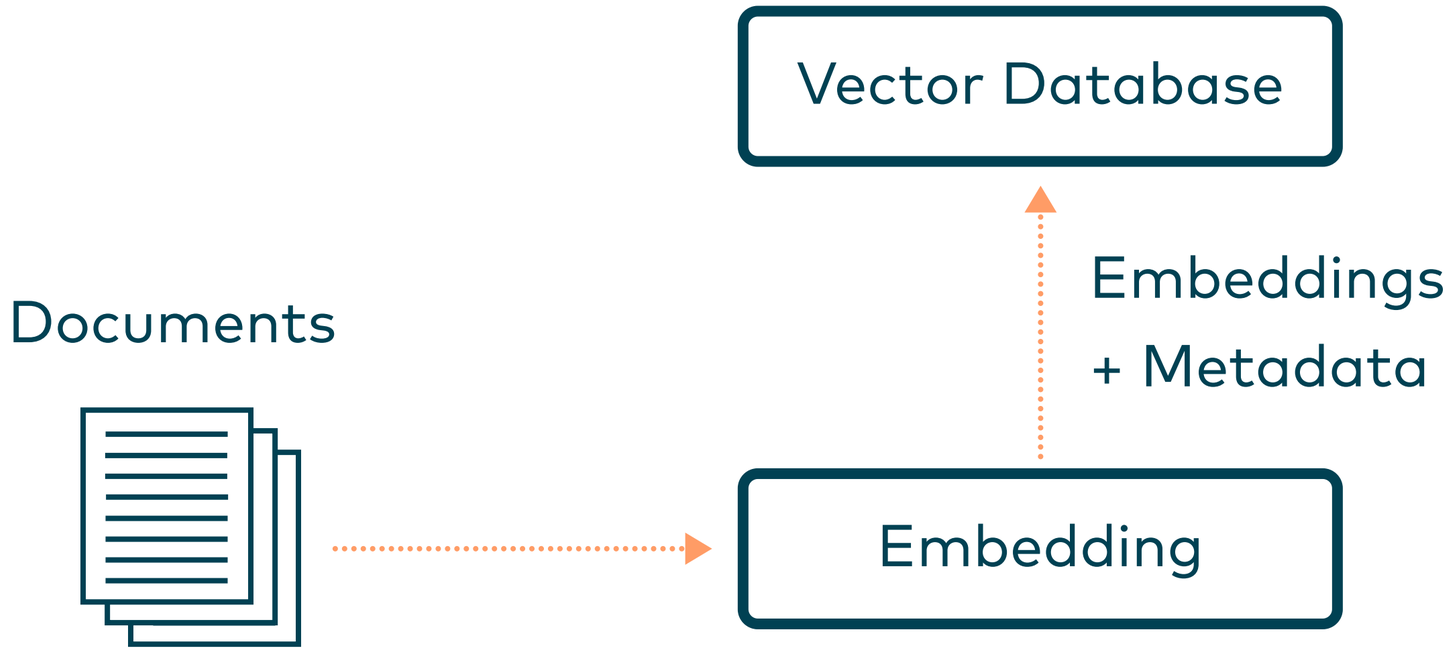
Results

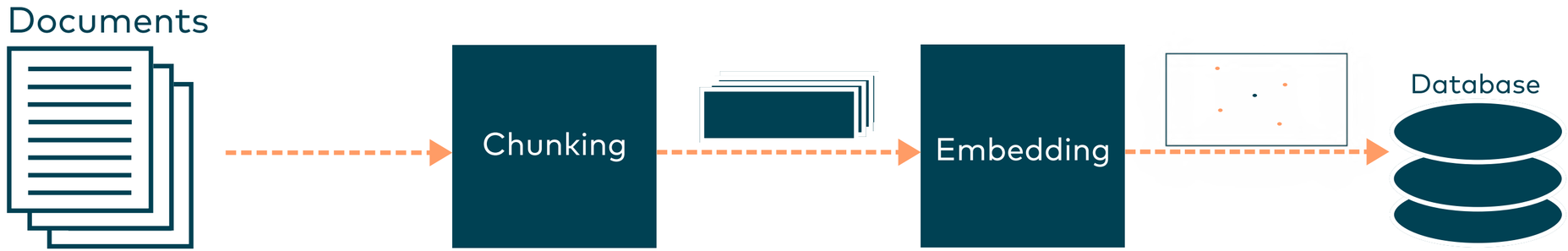
- Implementing RAG can result in a significant improvement in the performance and accuracy of your AI application.
- By basing an LLM on a set of external, verifiable facts, the model has fewer opportunities to incorporate information into its parameters. This reduces the likelihood of an LLM revealing sensitive data.

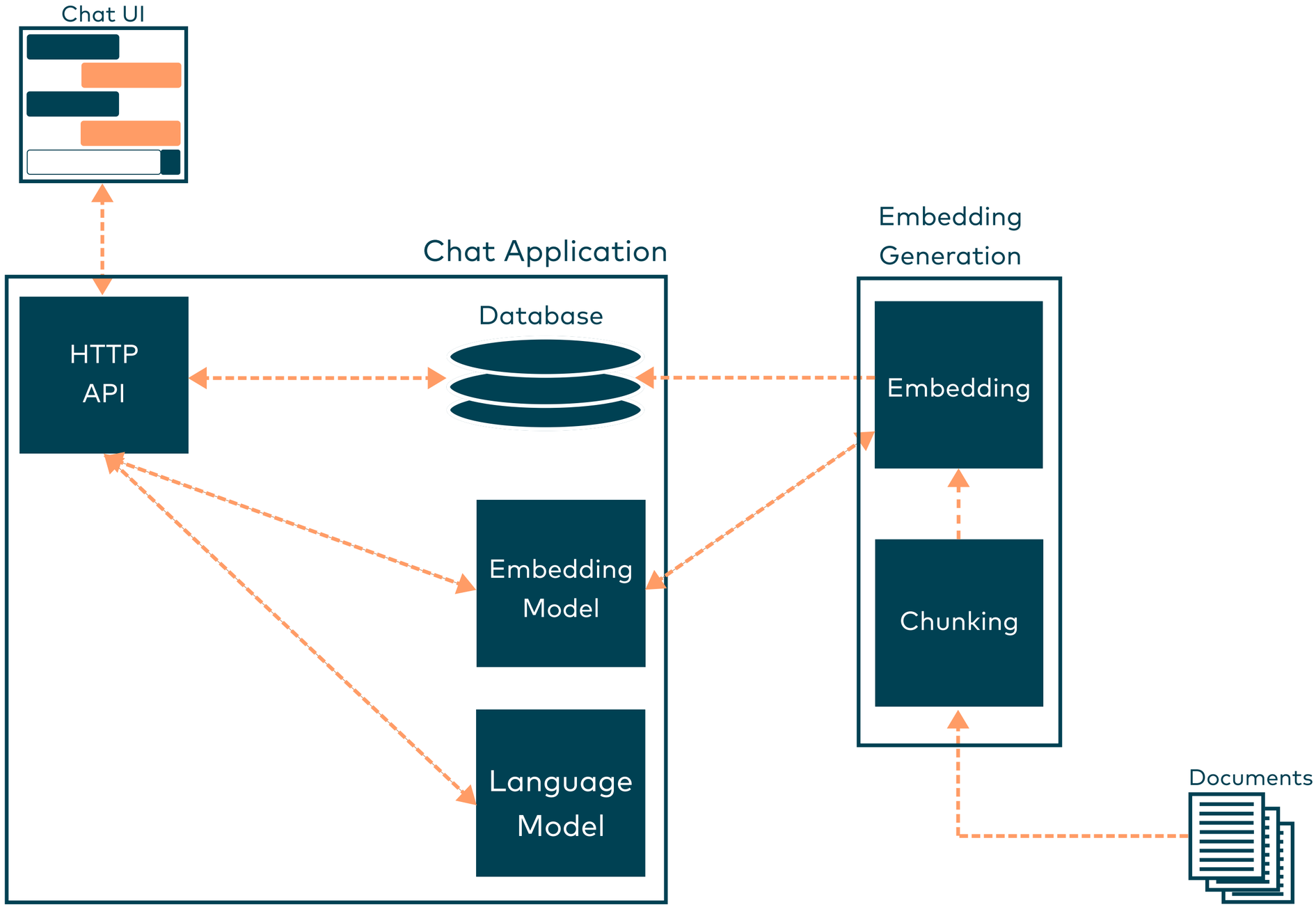
Additional benefits of RAG

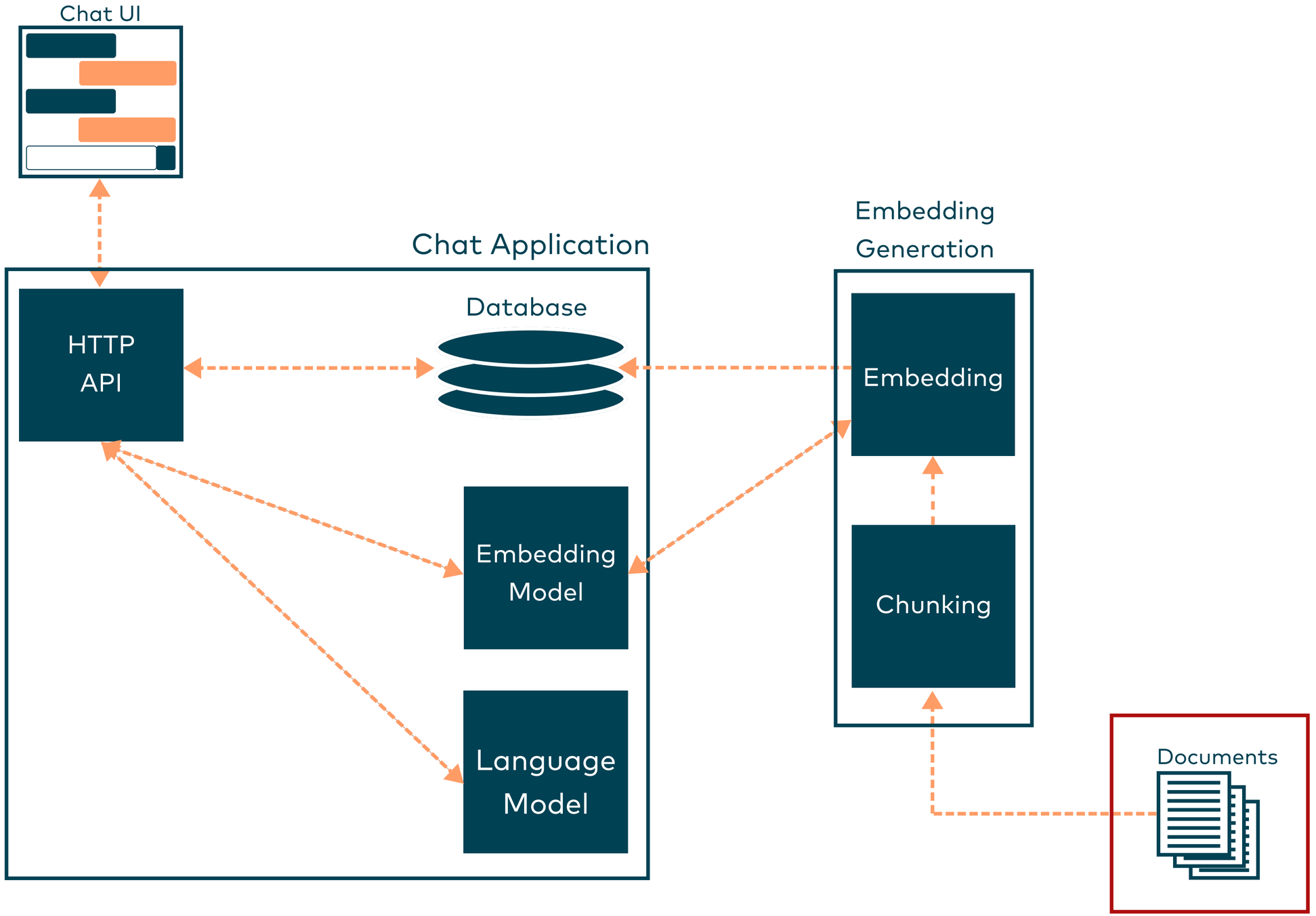
- The LLM only uses information from your domain
- For a given question, one can test if the application chooses the expected chunks from the domain data
- For expected chunks, one can test if the generated response is using the chunks
- Combine both to test if a given response is answering the question by using domain data
- This can be used for model testing and model evaluation

We still need to understand how to deal with this part




Documents
Chunking
Embedding
Database


Chat UI
Chat Application
Embedding
Generation
Database
HTTP
API
Embedding
Embedding
Model
Chunking
Language
Model
Documents


Chat UI
Chat Application
HTTP API
Database
Embedding Model
Language Model
Embedding Generation
Embedding
Chunking
Documents

Documents may be large and complex

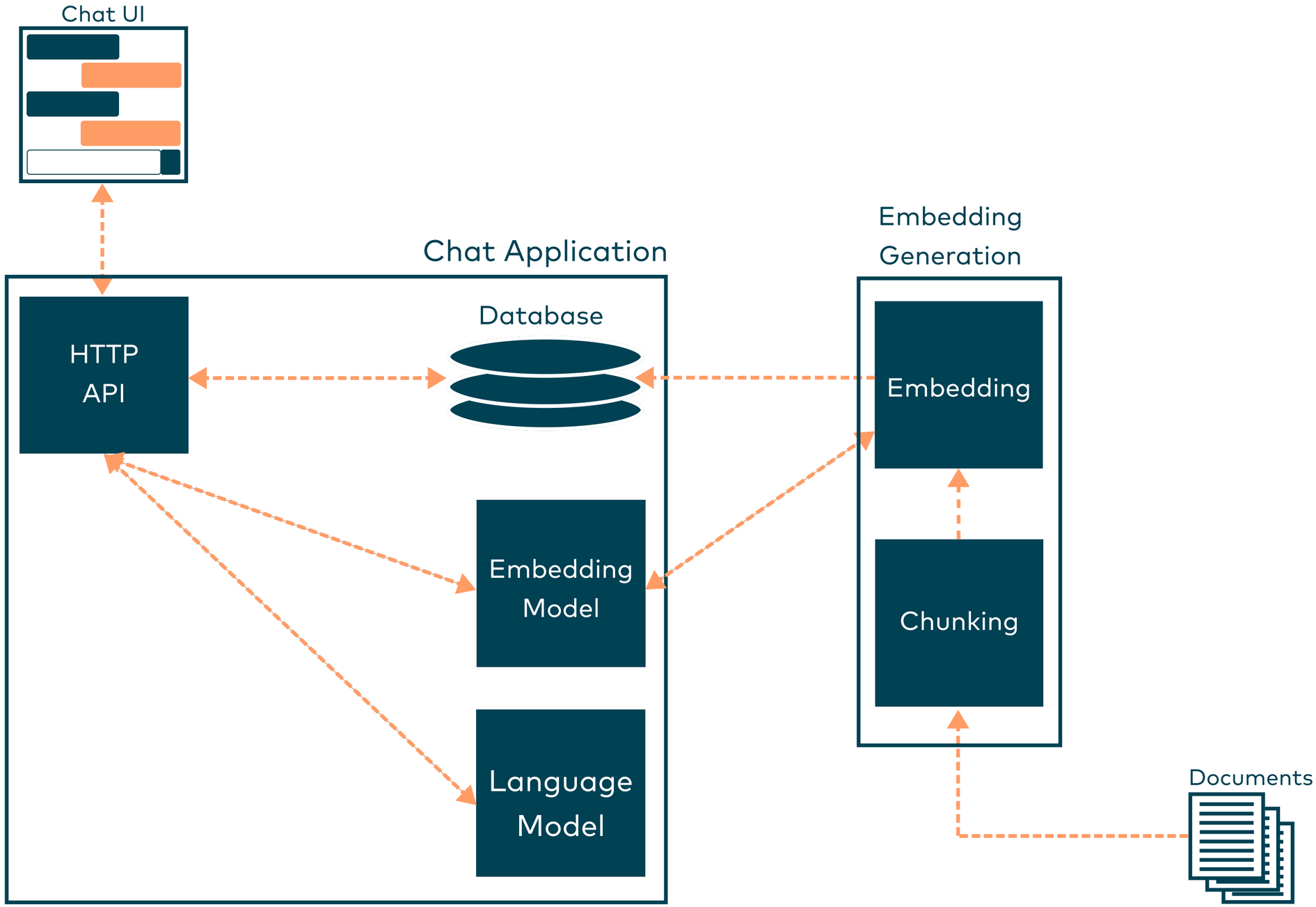
Example

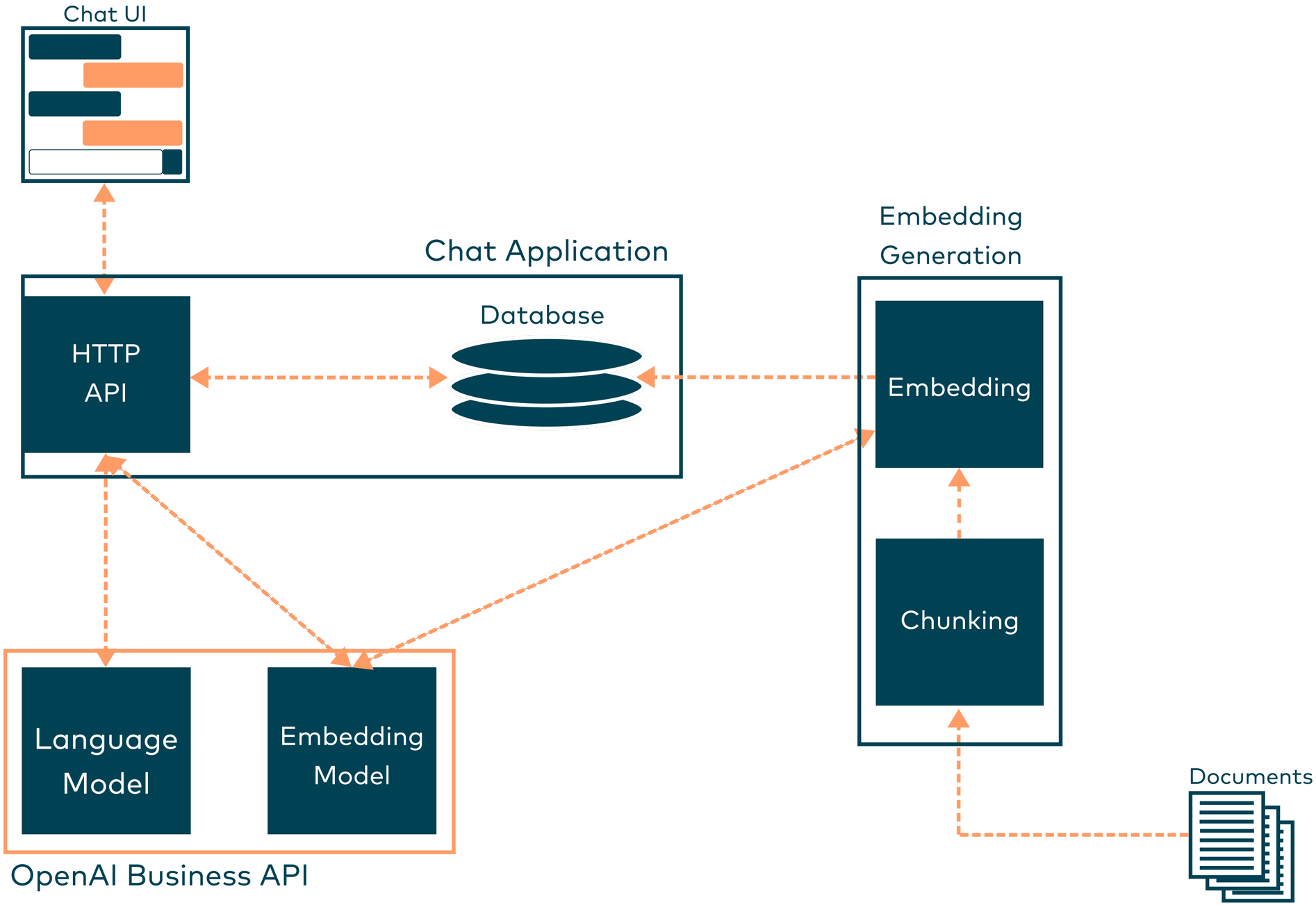
- innoq.com website
  - ~ 3500 pages (information, articles, transcripts)
  - 12000 chunks
  - 2 hours of embedding time

Example

- innoq.com website
  - ~ 3500 pages (information, articles, transcripts)
  - 12000 chunks
  - 2 hours of embedding time

This was not large or complex data


Chat UI
Chat Application
Embedding
Generation
Database
HTTP
API
Embedding
Embedding
Model
Chunking
Language
Model
Documents


Chat UI
Chat Application
Database
HTTP
API
Embedding
Generation
Embedding
Chunking
Language
Model
Embedding
Model
OpenAI Business API
Documents

Local Embedding Model

- alpaca-native-7B
- Based on Metas Llama2 model

Local Embedding Model

- alpaca-native-7B
- Based on Metas Llama2 model

→ 20 hours of embedding time

Local Generative Model

- Llama2-7B-chat, Llama2-13B-chat
- NVIDIA A2000 (12Gb GPU memory)

→ 45 second inference

Local Generative Model

- Llama2-7B-chat, Llama2-13B-chat ← small models
- NVIDIA A2000 (12Gb GPU memory)

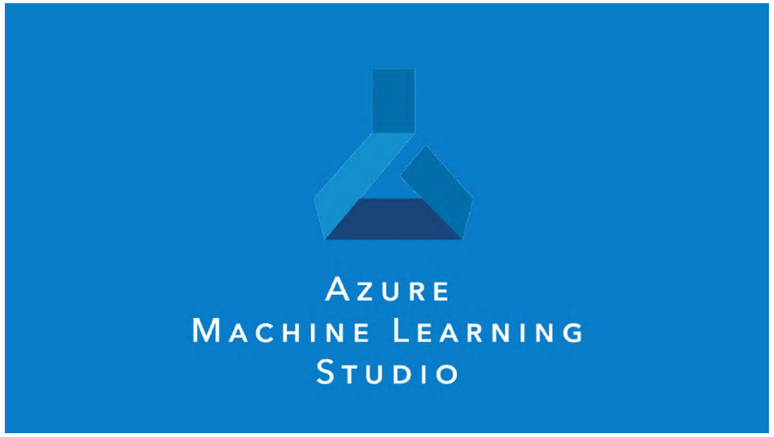
→ 45 second inference

→ after optimization 30 second inference ← inacceptable to be used in production

Self-hosted LLMs

(in the cloud)

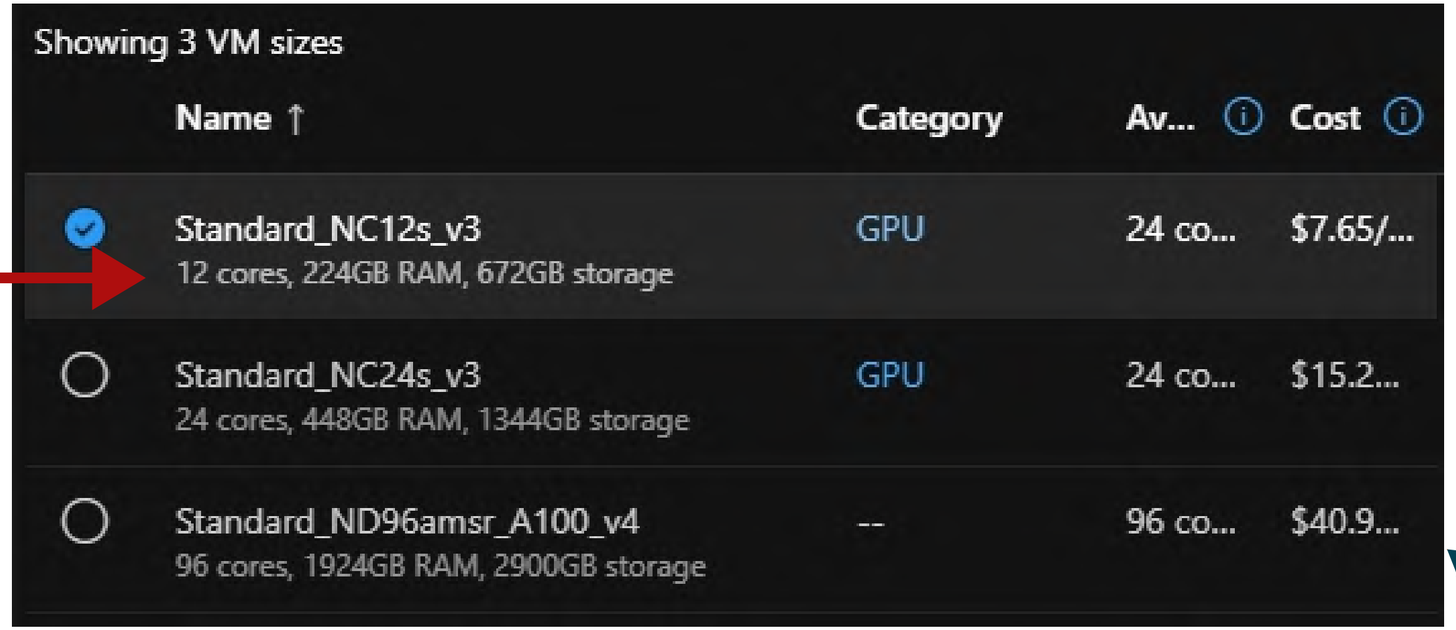
Amazon SageMaker



Vertex AI

LLM Cloud Services

- Deploy your own LLM
- Example from Azure Machine Learning Studio for **llama2-7b-chat**



requirements

$/hour

most performant VM

LLM Cloud Services

- $7.65/hr = $5691.60/month
- Cost for the smallest Llama2 with the lowest performing VM on Azure
- ~ 3s inference

LLM Cloud Services

- $7.65/hr = $5691.60/month
- Cost for the smallest Llama2 with the lowest performing VM on Azure
- ~ 3s inference

- On AWS Sagemaker:
  - ~ $36/day = $1080/month
  - ~ 10s inference

Instance-Größe	vCPUs	Instance-Arbeitsspeicher (GiB)	GPU-Modell	GPUs	Speicher insgesamt (GB)	Speicher pro GPU (GB)	Netzwerkbandbreite (Gbit/s)	EBS-Bandbreite (GBit/s)	Instance-Speicher (GB)
ml.g5n.xlarge	4	16	NVIDIA A10G	1	24	24	Bis zu 10	Bis zu 3,5	1 x 250
ml.g5.2xlarge	8	32	NVIDIA A10G	1	24	24	Bis zu 10	Bis zu 3,5	1 x 450

LLM Cloud Services

- Huge differences in pricing
- Examples only for the small models

LLM Cloud Services

- Huge differences in pricing
- Examples only for the small models
- There are also 13B and 70B versions of llama2
  - llama2-13b-chat: $15/hr = $11160/month
  - llama2-70b-chat: $40.9/hr = $30429.6/month

Azure pricings

llama2-13b-chat

Showing 2 VM sizes

	Name ↑	Category	Av...	Cost
☑	Standard_NC24s_v3 24 cores, 448GB RAM, 1344GB storage	GPU	24 co...	$15.2...
☐	Standard_ND96amsr_A100_v4 96 cores, 1924GB RAM, 2900GB storage	--	96 co...	$40.9...

llama2-70b-chat

Showing 1 VM sizes

	Name ↑	Category	Av...	Cost
☑	Standard_ND96amsr_A100_v4 96 cores, 1924GB RAM, 2900GB storage	--	96 co...	$40.9...

LLM model sizes

- Bigger LLMs provide higher accuracy
- And thus less hallucination
- As a trade-off, they are slower than smaller models

RAG

TO THE RESCUE

Reduce cost with RAG

- Using RAG the answer is only generated from given chunks

Reduce cost with RAG

- Using RAG the answer is only generated from given chunks
- This removes the hallucination from LLMs

Reduce cost with RAG

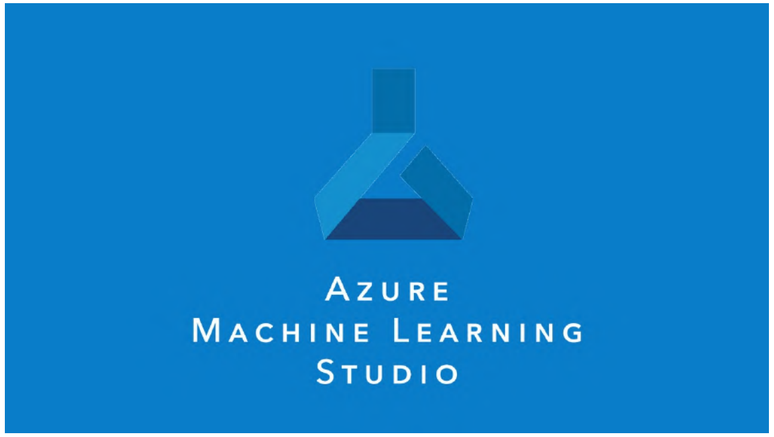
- Using RAG the answer is only generated from given chunks
- This removes the hallucination from LLMs
- Smaller models can be used without hallucination
- This also accelerates the inference

Reduce cost with RAG

- Llama2-7b-chat and Llama2-13b-chat are suitable for basic use-cases

Compliance

Amazon SageMaker



Vertex AI

Azure

- Azure Machine Learning Studio:
  - Model Catalog is still a "Preview", thus it can not guarantee that it follows the Azure DPA (last check October 25th 2023)

- Azure OpenAI:
  - Run your own ChatGPT deployment
  - Data is not shared with any service of OpenAI
  - Also still a preview feature, but may be the most promising model once it is fully established

https://docs.aws.amazon.com/sagemaker/latest/dg/data-protection.html

Amazon Web Services

- AWS Sagemaker:
  - AWS seems to not use prompts received from users

- AWS Bedrock:

> Amazon Bedrock doesn't use your prompts and continuations to train any AWS models or distribute them to third parties. Your training data isn't used to train the base Amazon Titan models or distributed to third parties. Other

https://docs.aws.amazon.com/bedrock/latest/userguide/data-protection.html

Google Cloud Platform

- Google Cloud Platform:

AI/ML Privacy Commitments for Google Cloud

Google Cloud customers benefit from:

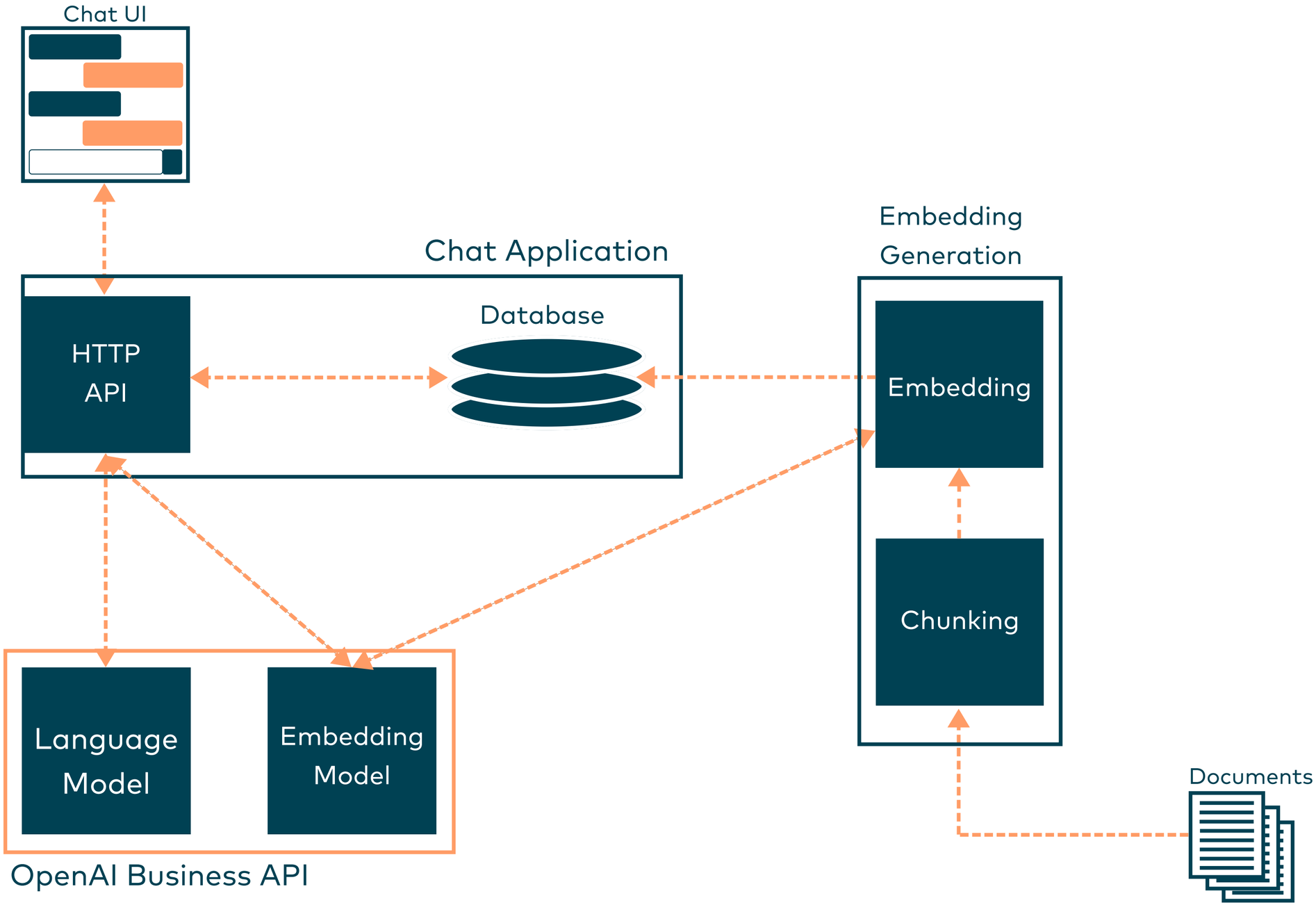
- **Your data is your data.** The data or content generated by a Generative AI Service prompted by Customer Data ("Generated Output") is considered Customer Data[1], that Google only process according to customer's instructions[2].

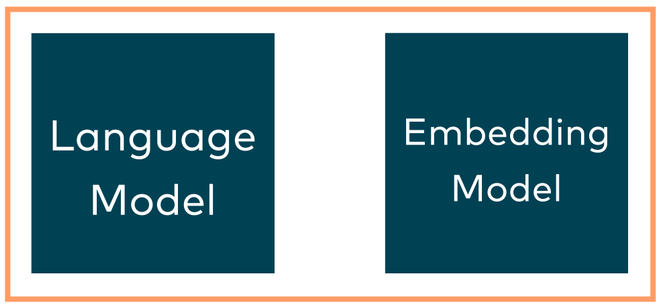
- Google Vertex AI:

You own and control your data and your data stays within your organization. Whether it is in our Vertex AI Platform or Generative AI App Builder (Gen App Builder), we recognize that customers want their data to be private and not be shared with the broader Google or Large Language Model training corpus. Customers maintain control over where their data is stored and how or if it is used, letting them safely pursue data-rich use cases while complying with various regulations. Google does not store, read, or use customer data outside your cloud tenant.

https://services.google.com/fh/files/misc/genai_privacy_google_cloud_202308.pdf

Always check the data protection for each new service


Chat UI
Chat Application
Database
HTTP
API
Embedding
Generation
Embedding
Chunking
Language
Model
Embedding
Model
OpenAI Business API
Documents


Language Model
Embedding Model
OpenAI Business API

OpenAI Business API

OpenAI Business API

- Access the OpenAI API as a business
- Protects your confidential data
- Data is not used by OpenAI

OpenAI Business API

- Access the OpenAI API as a business
- Protects your confidential data
- Data is not used by OpenAI
- Token-based pricing model

OpenAI Business API

- Access the OpenAI API as a business
- Protects your confidential data
- Data is not used by OpenAI
- Token-based pricing model

Token-based pricing

- Number of tokens is based on:
  - length of the prompt
  - length of the answer

Token-based pricing

- Number of tokens is based on:
  - length of the prompt
  - length of the answer

- Example:
  - prompt 200 tokens

    1. embedding the query (prompt)

Token-based pricing

- Number of tokens is based on:
  - length of the prompt
  - length of the answer

- Example:
  - prompt 200 tokens

    1. embedding the query (prompt)
    2. Retrieving chunks, n=5 nearest neighbours, chunk_length=300

Token-based pricing

- Number of tokens is based on:
  - length of the prompt
  - length of the answer

- Example:
  - prompt 200 tokens

    1. embedding the query (prompt)
    2. Retrieving chunks, n=5 nearest neighbours, chunk_length=300
    3. Send prompt (200 tokens) and chunks (1500 tokens) to OpenAI API

Token-based pricing

- Number of tokens is based on:
  - length of the prompt
  - length of the answer

- Example:
  - prompt 200 tokens

    1. embedding the query (prompt)
    2. Retrieving chunks, n=5 nearest neighbours, chunk_length=300
    3. Send prompt (200 tokens) and chunks (1500 tokens) to OpenAI API
    4. Generate answer (e.g. max_answer_length=200)

Token-based pricing

- In this example 2100 tokens will be processed
- 200 embedding (ada v2)
- 1700 input
- 200 output

→ $0.00212 per chat request → $1 ~ 500 chat requests

Model	Input	Output
gpt-4	$0.03 / 1K tokens	$0.06 / 1K tokens
gpt-4-32k	$0.06 / 1K tokens	$0.12 / 1K tokens
gpt-3.5-turbo-1106	$0.0010 / 1K tokens	$0.0020 / 1K tokens
gpt-3.5-turbo-instruct	$0.0015 / 1K tokens	$0.0020 / 1K tokens
ada v2	$0.0001 / 1K tokens	

Token-based pricing

- In this example 2100 tokens will be processed
- 200 embedding (ada v2)
- 1700 input
- 200 output

Update from OpenAI DevDay (November 7th)

GPT4-turbo

cheaper than GPT4

gpt-4-1106-preview	$0.01 / 1K tokens	$0.03 / 1K tokens
gpt-4-1106-vision-preview	$0.01 / 1K tokens	$0.03 / 1K tokens

Model	Input	Output
gpt-4	$0.03 / 1K tokens	$0.06 / 1K tokens
gpt-4-32k	$0.06 / 1K tokens	$0.12 / 1K tokens
gpt-3.5-turbo-1106	$0.0010 / 1K tokens	$0.0020 / 1K tokens
gpt-3.5-turbo-instruct	$0.0015 / 1K tokens	$0.0020 / 1K tokens
ada v2	$0.0001 / 1K tokens	

Measure, measure, measure!

chunk size

number of nearest neighbours (chunks)

maximum prompt length (tokens)

maximum answer length (tokens)

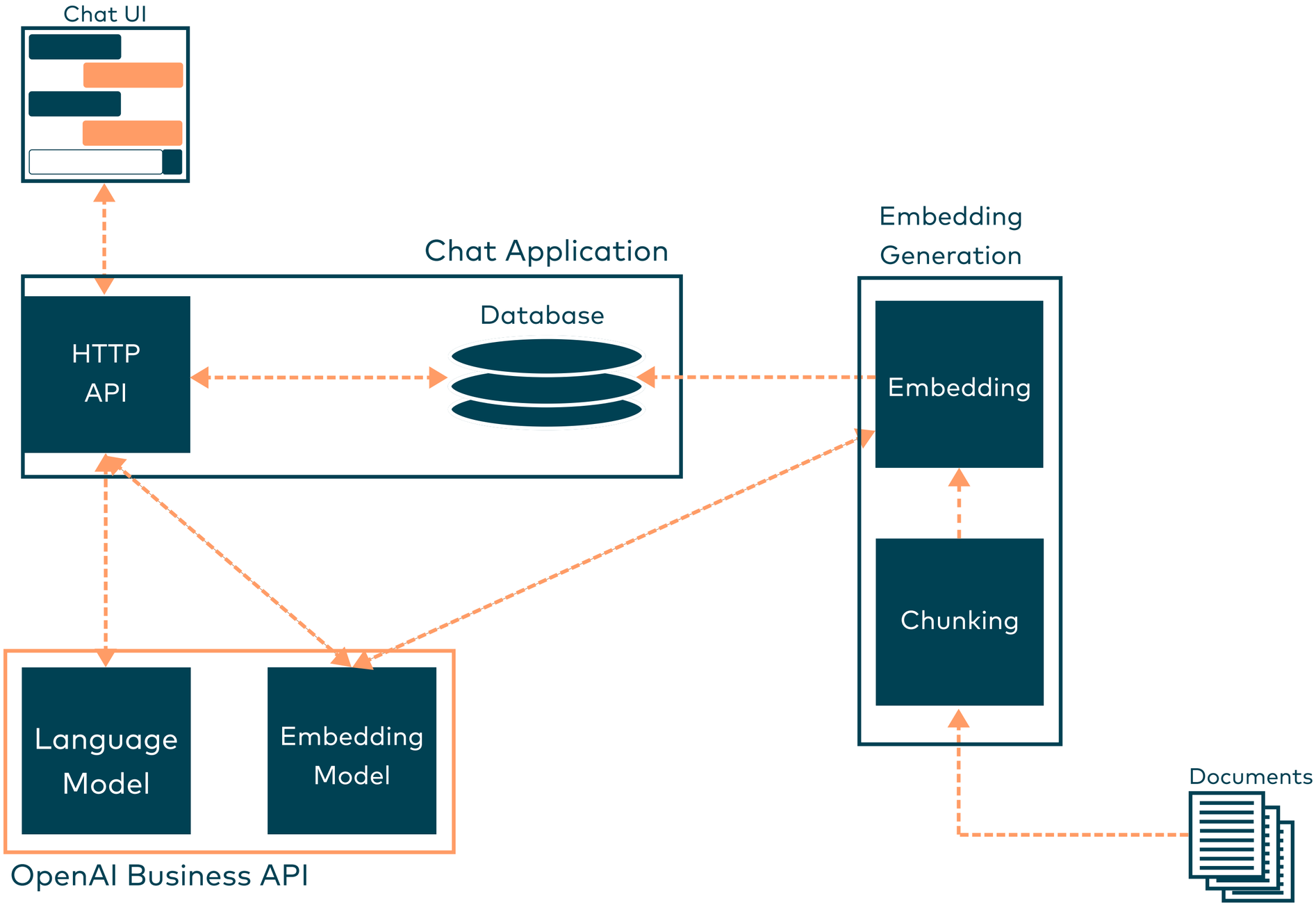
generative model (GPT3.5-turbo, GPT4)

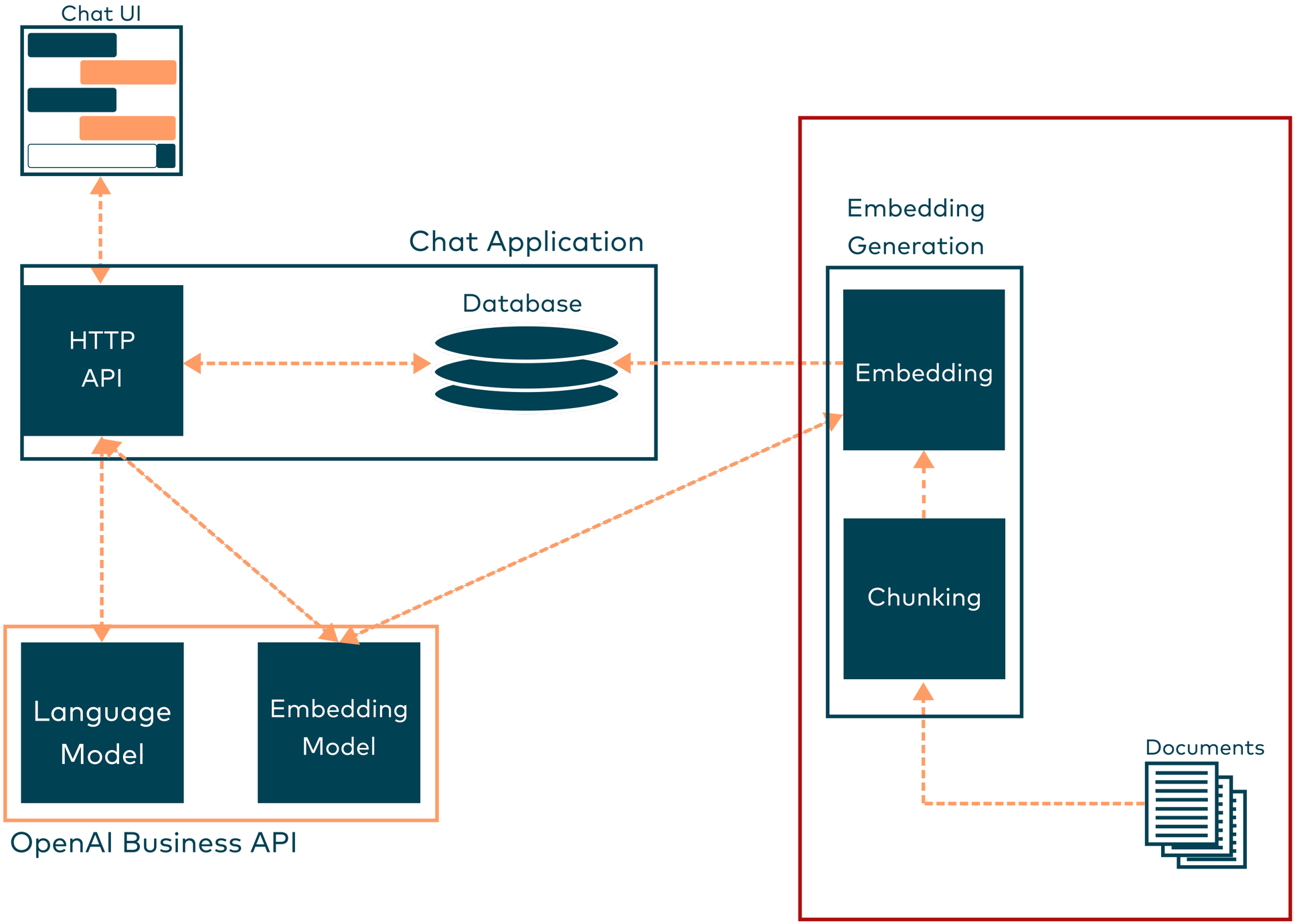
frequency of user requests

number of users

Set a cost limit

increase on demand


Chat UI
Chat Application
Database
HTTP
API
Embedding
Generation
Embedding
Chunking
Language
Model
Embedding
Model
OpenAI Business API
Documents


Chat UI
Chat Application
Database
HTTP
API
Embedding
Generation
Embedding
Chunking
Documents
Language
Model
Embedding
Model
OpenAI Business API

That can be a lot of tokens

Cost of Embedding Generation

- Evaluate the cost of your embedding pipeline
- Plan how frequently new chunks will be created
- Estimate the runtime of the pipeline

Our pipeline ran for 2 hours

with only a few chunks

Distributed Computation

- Using cloud resources you can distribute the pipeline

Distributed Computation

- Using cloud resources you can distribute the pipeline
- For example **Ray.io**

Distributed Computation

- Scale your pipeline with increasing amount of documents

Distributed Computation

- Scale your pipeline with increasing amount of documents
- Scale your chatbot with increasing amount of users

Distributed Computation

- Scale your pipeline with increasing amount of documents
- Scale your chatbot with increasing amount of users

Distributed Computation

- Scale your pipeline with increasing amount of documents
- Scale your chatbot with increasing amount of users
- Run the pipeline and chatbot in **your own data center**

Distributed Computation

- Scale your pipeline with increasing amount of documents
- Scale your chatbot with increasing amount of users
- Run the pipeline and chatbot in **your own data center**

Only works if you already have a data center

Have fun building a chatbot for your domain

Have fun building a chatbot for your domain

or contact us with your use-case

Thank you! Questions?

INNOQ
www.innoq.com

Marco Steinke
marco.steinke@innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 333660

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Wendenstr. 130
20537 Hamburg

Königstorgraben 11
90402 Nürnberg