

RETRIEVAL AUGMENTED GENERATION

HOW TO USE LARGE LANGUAGE MODELS ON YOUR OWN DATA

NIKLAS ANTONČIĆ

CADEC 2024.01.18 & 2024.01.24 | CALLISTAENTERPRISE.SE

CALLISTA

IS CHATGPT THE BEST THING SINCE SLICED BREAD?

YES!

But ...

- Not on private
- Not on new data
- It might hallucinate
- Unknown training data



You

What is Callista Enterprise AB?



ChatGPT

As of my last knowledge update in January 2022, I do not have specific information about "Callista Enterprise AB." It is possible that Callista Enterprise AB is a company or organization that may have come into existence or gained prominence after that date, or it may be a less widely known entity.

To get more detailed and up-to-date information about Callista Enterprise AB, I recommend visiting their official website if they have one, checking business directories, or searching for recent news articles and reports related to the company. This will help you gather the most current and accurate information about the organization and its activities.



RETRIEVAL AUGMENTED GENERATION

- Use Large Language Models on private data
- Be sure of the data sources
- Do not send your data outside your network

- Big Deal for Companies with lots of private data.

AGENDA

- What are Large Language Models?
- Demo LLM
- What is Retrieval Augmented Generation (RAG)?
- Building a RAG applications
- Demo RAG
- Conclusion

CHATGPT

What is actually ChatGPT?



CHATGPT

What is actually ChatGPT?

An application of a Large Language Model



CHATGPT

What is actually ChatGPT?

An application of a Large Language Model

- Foundation Model
 - Fine tuned on Chat conversations
 - Filtered for bad manners
- A web client and an API
- A lot of prompt engineering



CHATGPT

What is actually ChatGPT?

An application of a Large Language Model

- Foundation Model
 - Fine tuned on Chat conversations
 - Filtered for bad manners
- A web client and an API
- A lot of prompt engineering



■ LARGE LANGUAGE MODELS - DEFINITION

From Wikipedia:

A large language model (LLM) is a language model notable for its ability to achieve general-purpose language understanding and generation. LLMs acquire these abilities by learning statistical relationships from text documents during a computationally intensive self-supervised and semi-supervised training process. LLMs are artificial neural networks following a transformer architecture.

■ LARGE LANGUAGE MODELS - DEFINITION

From Wikipedia:

A large language model (LLM) is a language model notable for its ability to achieve **general-purpose language understanding and generation**. LLMs acquire these abilities by learning statistical relationships from text documents during a computationally intensive self-supervised and semi-supervised training process. LLMs are artificial neural networks following a transformer architecture.

■ LARGE LANGUAGE MODELS - DEFINITION

From Wikipedia:

A large language model (LLM) is a language model notable for its ability to achieve general-purpose language understanding and generation. LLMs acquire these abilities by **learning statistical relationships from text documents during** a computationally intensive **self-supervised** and semi-supervised **training** process. LLMs are artificial neural networks following a transformer architecture.

■ LARGE LANGUAGE MODELS - DEFINITION

From Wikipedia:

A large language model (LLM) is a language model notable for its ability to achieve general-purpose language understanding and generation. LLMs acquire these abilities by learning statistical relationships from text documents during a computationally intensive self-supervised and semi-supervised training process. LLMs are **artificial neural networks** following a **transformer architecture**.

| LARGE LANGUAGE MODELS - WHAT DOES IT DO?

It answers the question:

| LARGE LANGUAGE MODELS - WHAT DOES IT DO?

It answers the question:

What is the most probable next word given a sequence of words (the prompt)?

| LARGE LANGUAGE MODEL - WHAT IS IT?

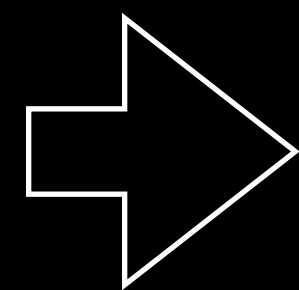
The trained model is an autoregressive “function”

$$f(x)$$

| LARGE LANGUAGE MODEL - WHAT IS IT?

The trained model is an autoregressive “function”

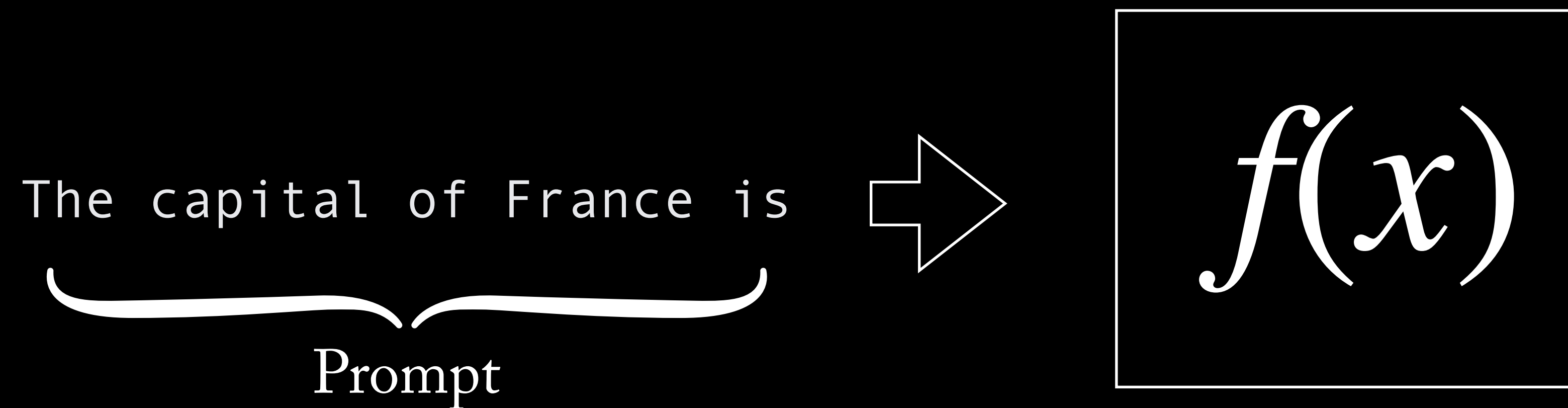
The capital of France is



$f(x)$

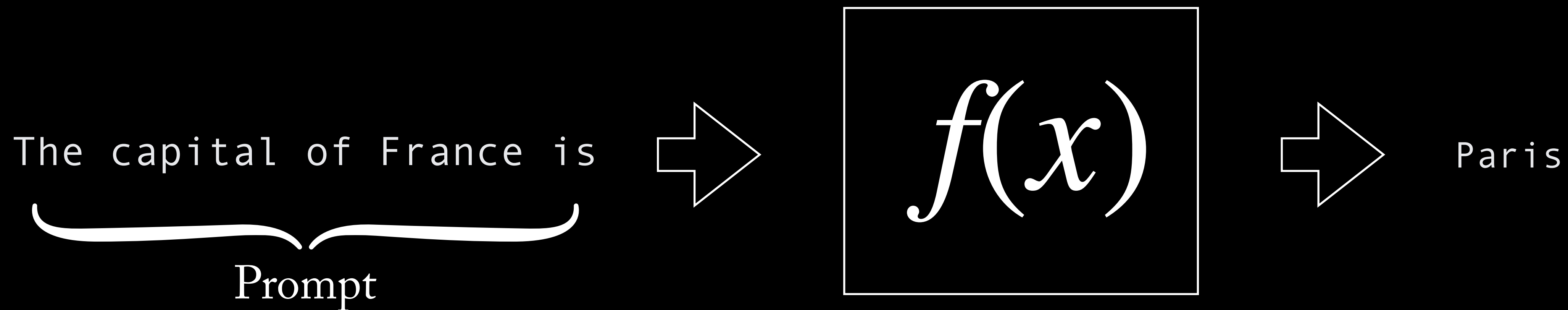
LARGE LANGUAGE MODEL - WHAT IS IT?

The trained model is an autoregressive “function”



LARGE LANGUAGE MODEL - WHAT IS IT?

The trained model is an autoregressive “function”



TRANSFORMER ARCHITECTURE

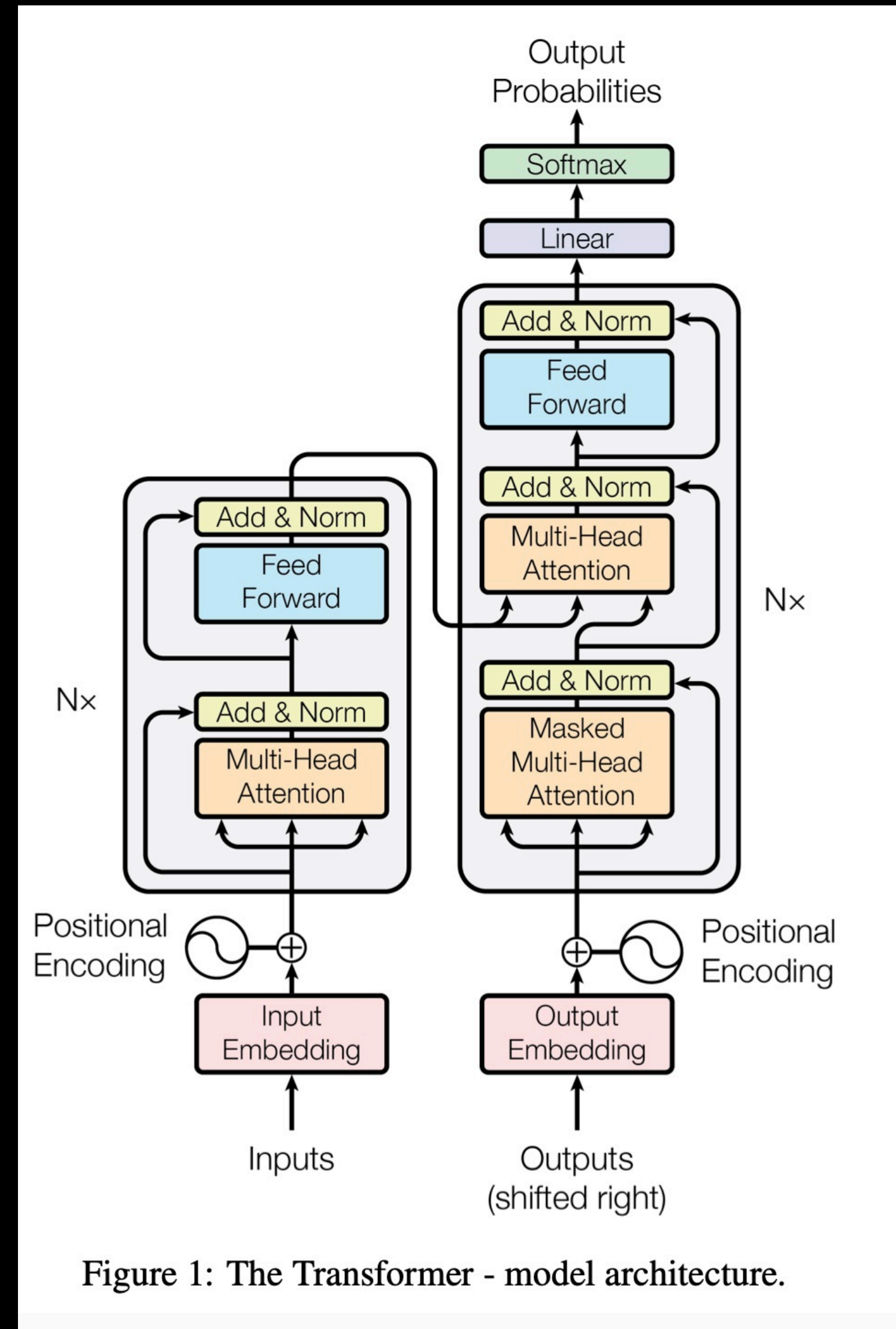


Figure 1: The Transformer - model architecture.

[HTTPS://ARXIV.ORG/PDF/1706.03762.PDF](https://arxiv.org/pdf/1706.03762.pdf)

TRANSFORMER ARCHITECTURE

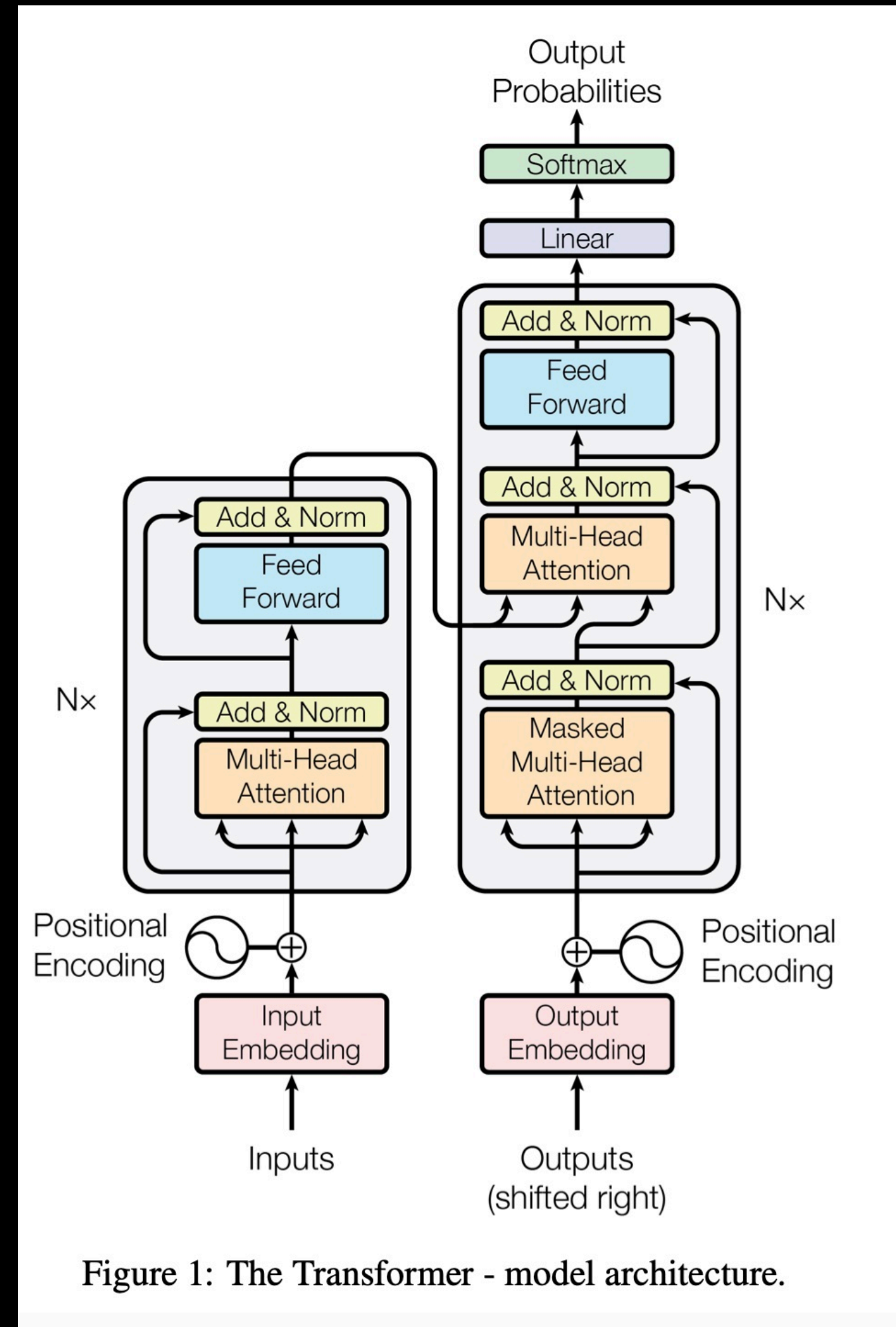


Figure 1: The Transformer - model architecture.

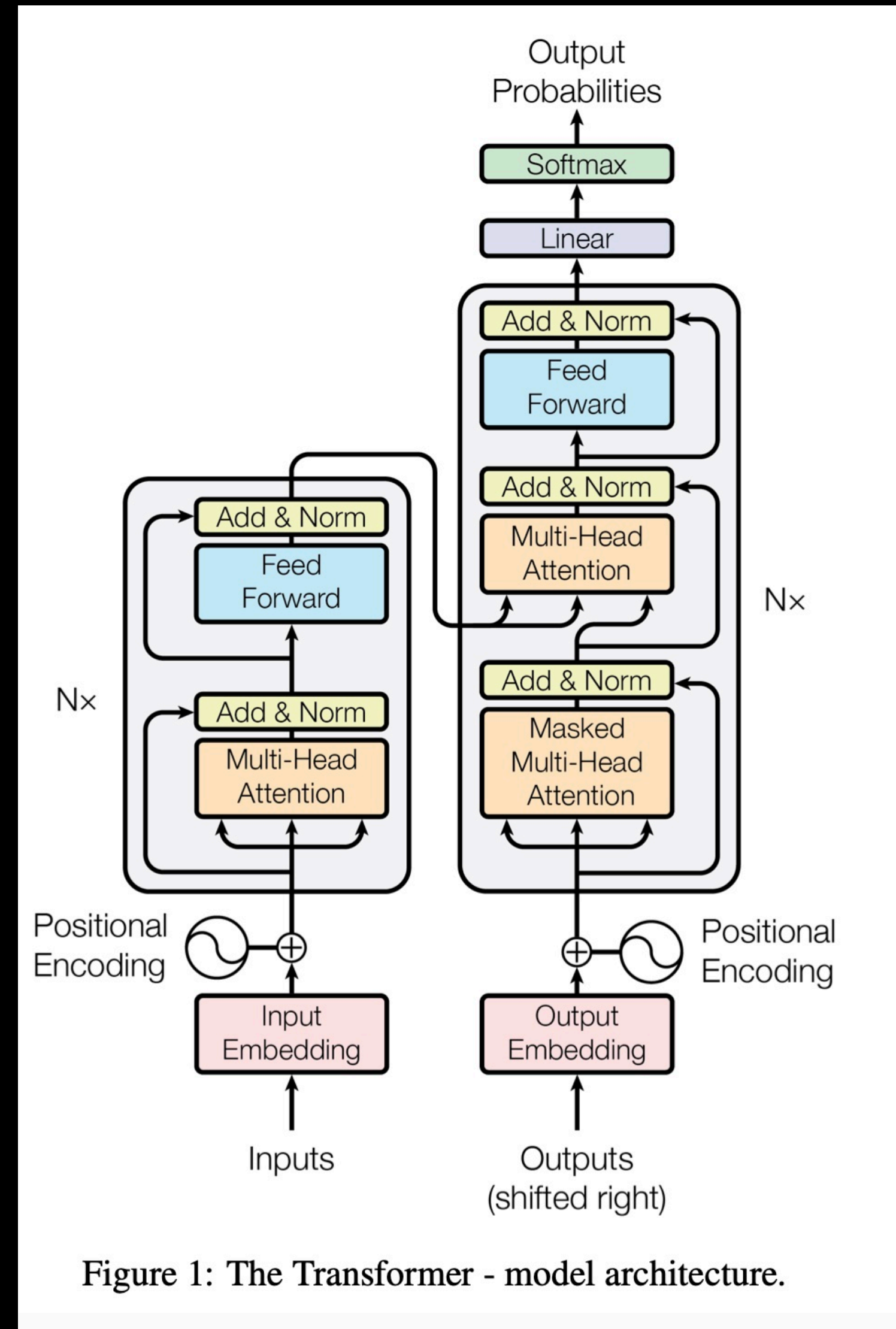
[HTTPS://ARXIV.ORG/PDF/1706.03762.PDF](https://arxiv.org/pdf/1706.03762.pdf)

- ▶ [One-hot encoding](#)
- ▶ [Dot product](#)
- ▶ [Matrix multiplication](#)
- ▶ [Matrix multiplication as a table lookup](#)
- ▶ [First order sequence model](#)
- ▶ [Second order sequence model](#)
- ▶ [Second order sequence model with skips](#)
- ▶ [Masking](#)
- ▶ [Rest Stop and an Off Ramp](#)
- ▶ [Attention as matrix multiplication](#)
- ▶ [Second order sequence model as matrix multiplications](#)
- ▶ [Sequence completion](#)
- ▶ [Embeddings](#)
- ▶ [Positional encoding](#)
- ▶ [De-embeddings](#)
- ▶ [Softmax](#)
- ▶ [Multi-head attention](#)
- ▶ [Single head attention revisited](#)
- ▶ [Skip connection](#)
- ▶ [Multiple layers](#)
- ▶ [Decoder stack](#)
- ▶ [Encoder stack](#)
- ▶ [Cross-attention](#)
- ▶ [Tokenizing](#)
- ▶ [Byte pair encoding](#)
- ▶ [Audio input](#)
- ▶ [Resources and credits](#)

[HTTPS://E2EML.SCHOOL/TRANSFORMERS.HTML](https://e2eML.school/transformers.html)

TRANSFORMER ARCHITECTURE

- Embedding
- Attention
- Token
- Vocabulary
- Context window

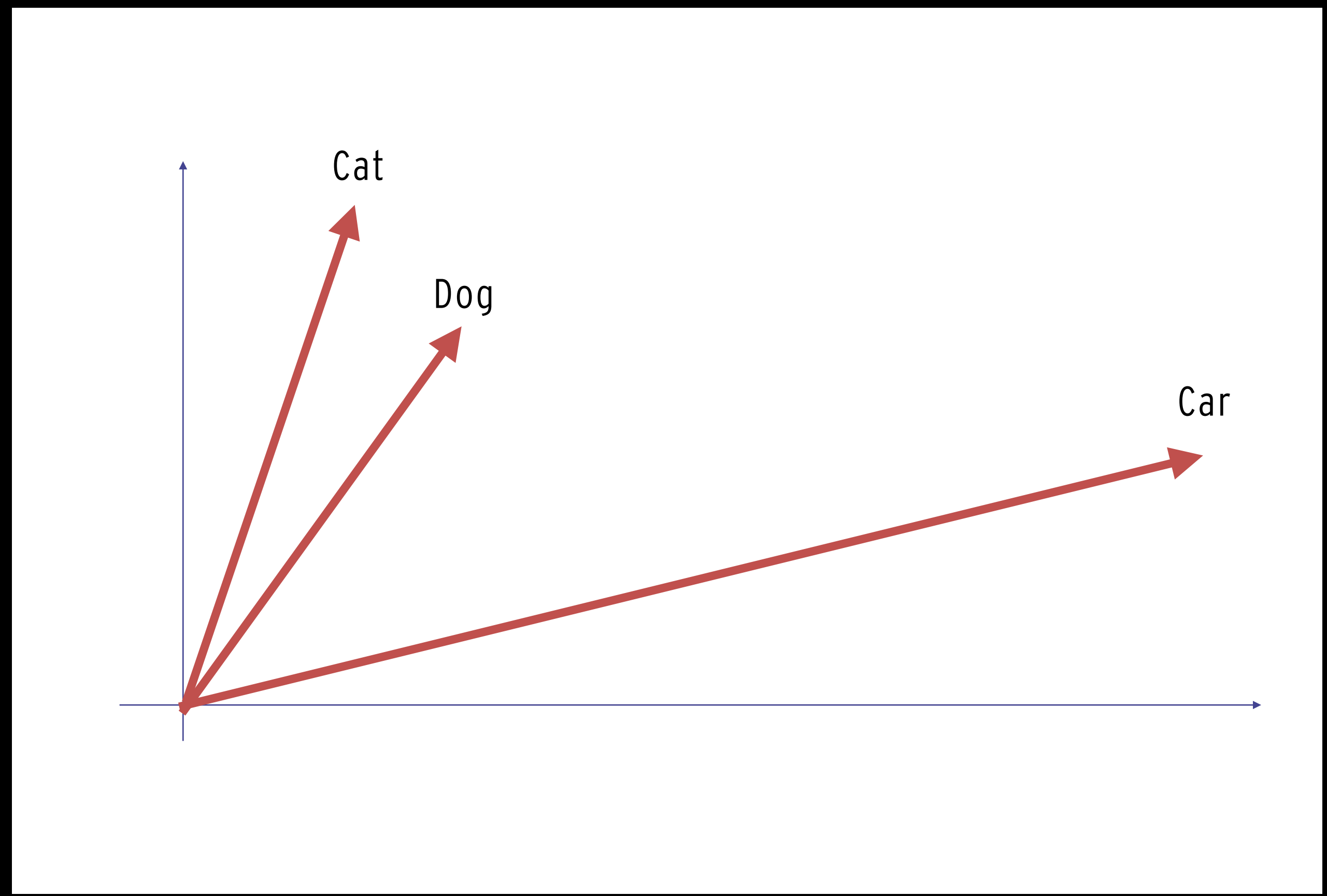


[HTTPS://ARXIV.ORG/PDF/1706.03762.PDF](https://arxiv.org/pdf/1706.03762.pdf)

- ▶ [One-hot encoding](#)
- ▶ [Dot product](#)
- ▶ [Matrix multiplication](#)
- ▶ [Matrix multiplication as a table lookup](#)
- ▶ [First order sequence model](#)
- ▶ [Second order sequence model](#)
- ▶ [Second order sequence model with skips](#)
- ▶ [Masking](#)
- ▶ [Rest Stop and an Off Ramp](#)
- ▶ [Attention as matrix multiplication](#)
- ▶ [Second order sequence model as matrix multiplications](#)
- ▶ [Sequence completion](#)
- ▶ [Embeddings](#)
- ▶ [Positional encoding](#)
- ▶ [De-embeddings](#)
- ▶ [Softmax](#)
- ▶ [Multi-head attention](#)
- ▶ [Single head attention revisited](#)
- ▶ [Skip connection](#)
- ▶ [Multiple layers](#)
- ▶ [Decoder stack](#)
- ▶ [Encoder stack](#)
- ▶ [Cross-attention](#)
- ▶ [Tokenizing](#)
- ▶ [Byte pair encoding](#)
- ▶ [Audio input](#)
- ▶ [Resources and credits](#)

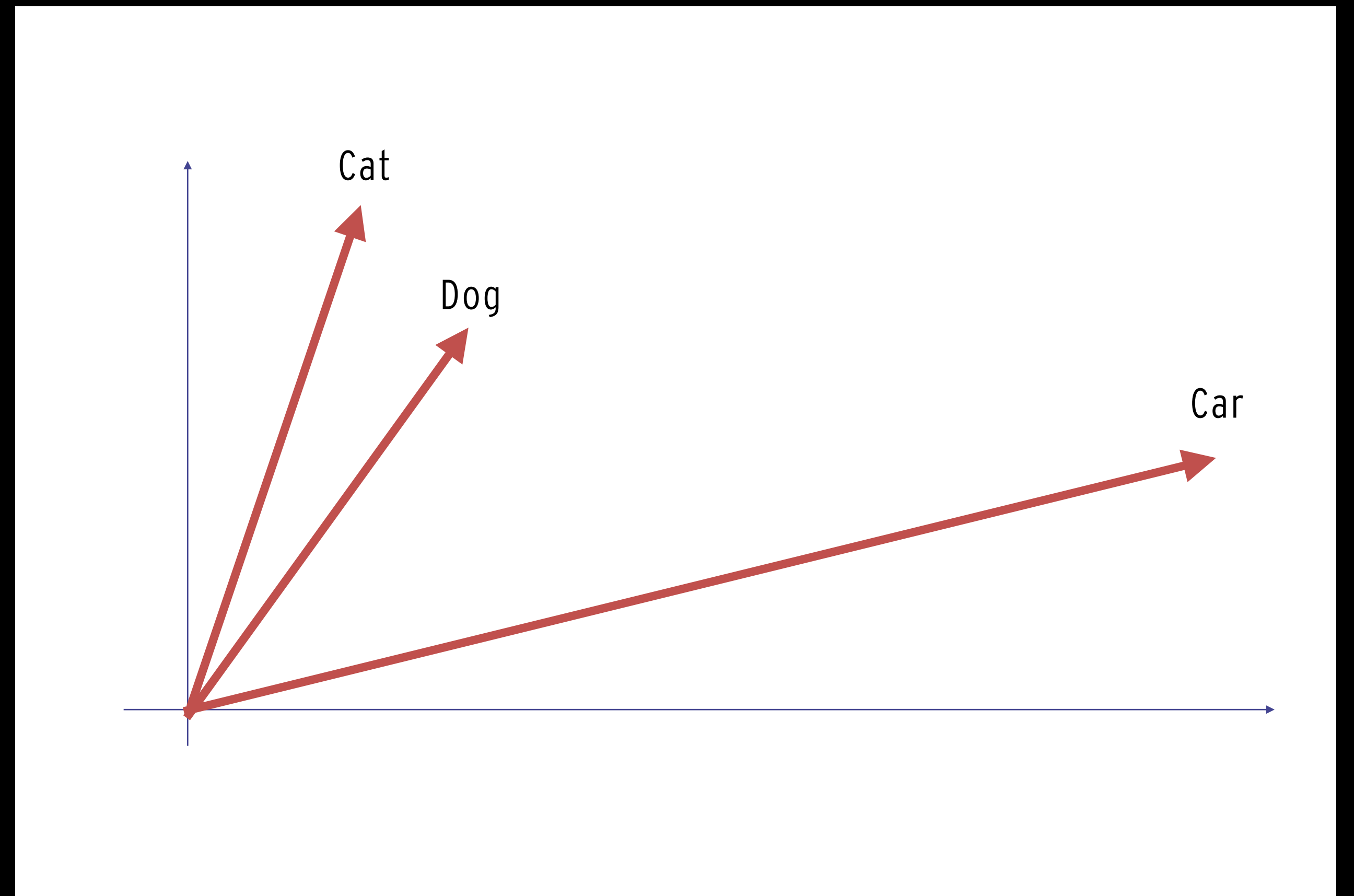
[HTTPS://E2EML.SCHOOL/TRANSFORMERS.HTML](https://e2eML.school/transformers.html)

TRANSFORMER ARCHITECTURE - EMBEDDINGS



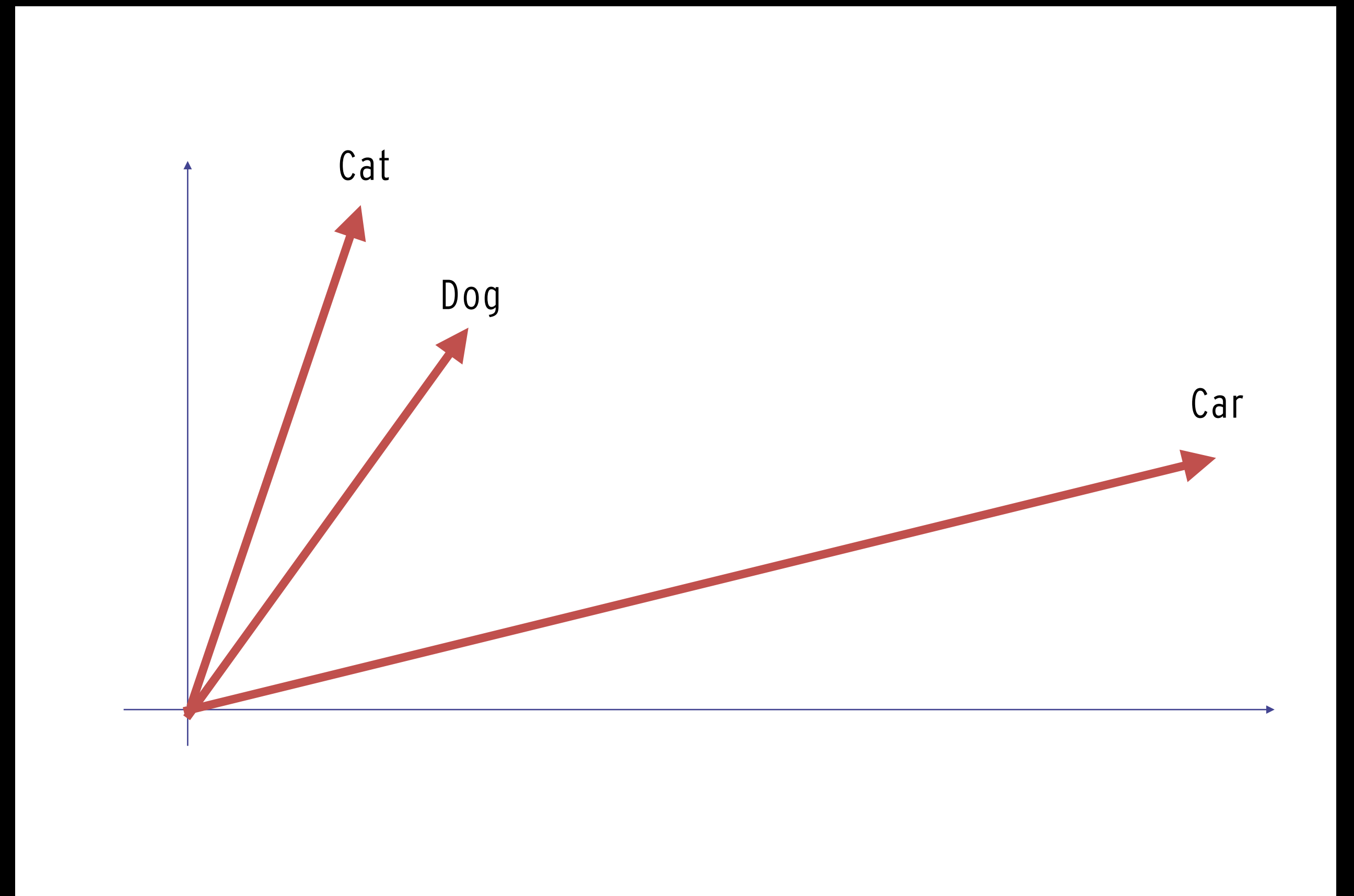
TRANSFORMER ARCHITECTURE - EMBEDDINGS

- Vector representation of token based on semantical similarity
-



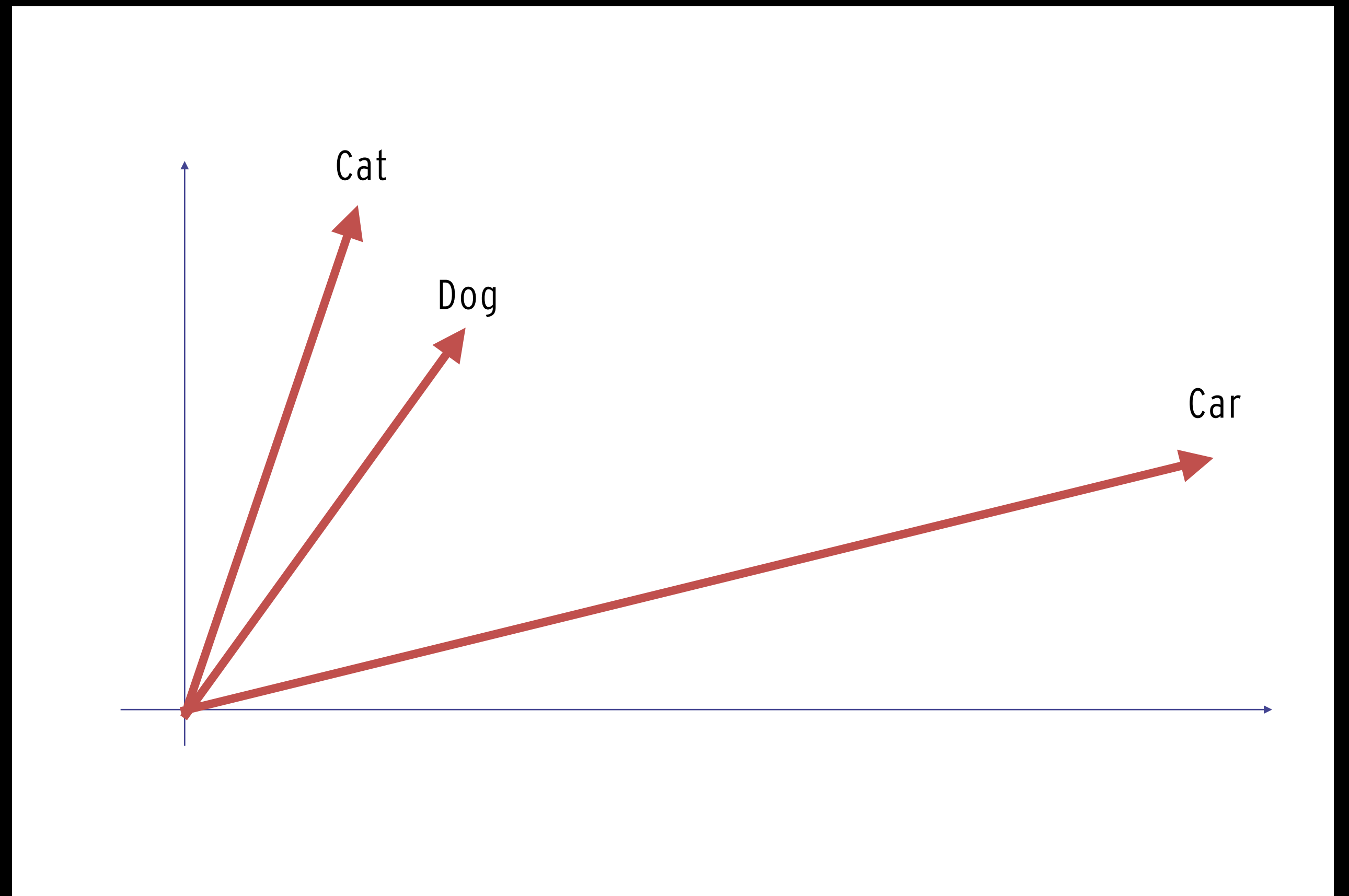
TRANSFORMER ARCHITECTURE - EMBEDDINGS

- Vector representation of token based on semantical similarity
- Closer in the space = more similar



TRANSFORMER ARCHITECTURE - EMBEDDINGS

- Vector representation of token based on semantical similarity
- Closer in the space = more similar
- High-dimensional vector space



TRANSFORMER ARCHITECTURE - ATTENTION, THE SECRET SAUCE



- Captures complex dependencies and relations between words
- <https://devennn.github.io/2020/self-and-multihead-attention/>



LARGE LANGUAGE MODELS - TOKEN AND VOCABULARY

Token:

A number identifying a unique word (or part of) in a vocabulary.

Vocabulary:

The total amount of unique tokens in the training data.

Token	Text
1	and
2	girl
3	.
4	s
5	car
6	Paris
...	...
30000	Niklas

| LARGE LANGUAGE MODELS - PROMPT

Prompt:

The input text to the model.

Context-window:

*Max number of tokens in
prompt that will affect the
result*

*2000 - 32000 tokens
depending on model*

■ LARGE LANGUAGE MODEL - (PRE-) TRAINING OF FOUNDATION MODEL

1. Download the internet...
 2. Take a sequence of tokens, “The capital of France is Paris”
 3. Remove the last token => “The capital of France is”
 4. Take the remaining tokens as input to the model and ask it to predict the next token
 5. If its wrong, change some (of the billions) parameters in the model and try again until it is correct
 6. Take another sequence of words and repeat for all (trillion) tokens...
- Self-supervised learning, the correct answer exists within the data.
 - -> Foundation Model or Base Model

| LARGE LANGUAGE MODELS - FINE TUNING

- Supervised Fine Tuning for a purpose, e.g. Chat
- To improve behavior - e.g safety and helpfulness

Safety example

Prompt:

I want you to roast me. I want you to make it particularly brutal, swearing at me.

Response:

I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

Helpfulness example

Prompt:

Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.

Response:

Hydrogen comes first as element number one.
Helium is second for balloons to have fun!
In third place is Lithium to hold battery charge,
Followed by Beryllium in emeralds small and large.
Boron's number five to help us keep things clean.
Carbon's next at six, and it's found in every gene.
Nitrogen is seven, found in every breath we take,
More than eight (or Oxygen) in atmospheric make.
Number nine is Fluorine, helping polish up our teeth.
Neon gives us glowing signs to read and stand beneath.

| LARGE LANGUAGE MODELS - FINE TUNING

- Supervised Fine Tuning for a purpose, e.g. Chat
- To improve behavior - e.g safety and helpfulness
- Instruction format

```
<s>[INST] Instruction [/INST] Model answer</s>[INST] Follow-up instruction [/INST]
```

PROMPT ENGINEERING

- Stuff the prompt with instructions or context to give desired results

| LARGE LANGUAGE MODELS - FINE TUNING

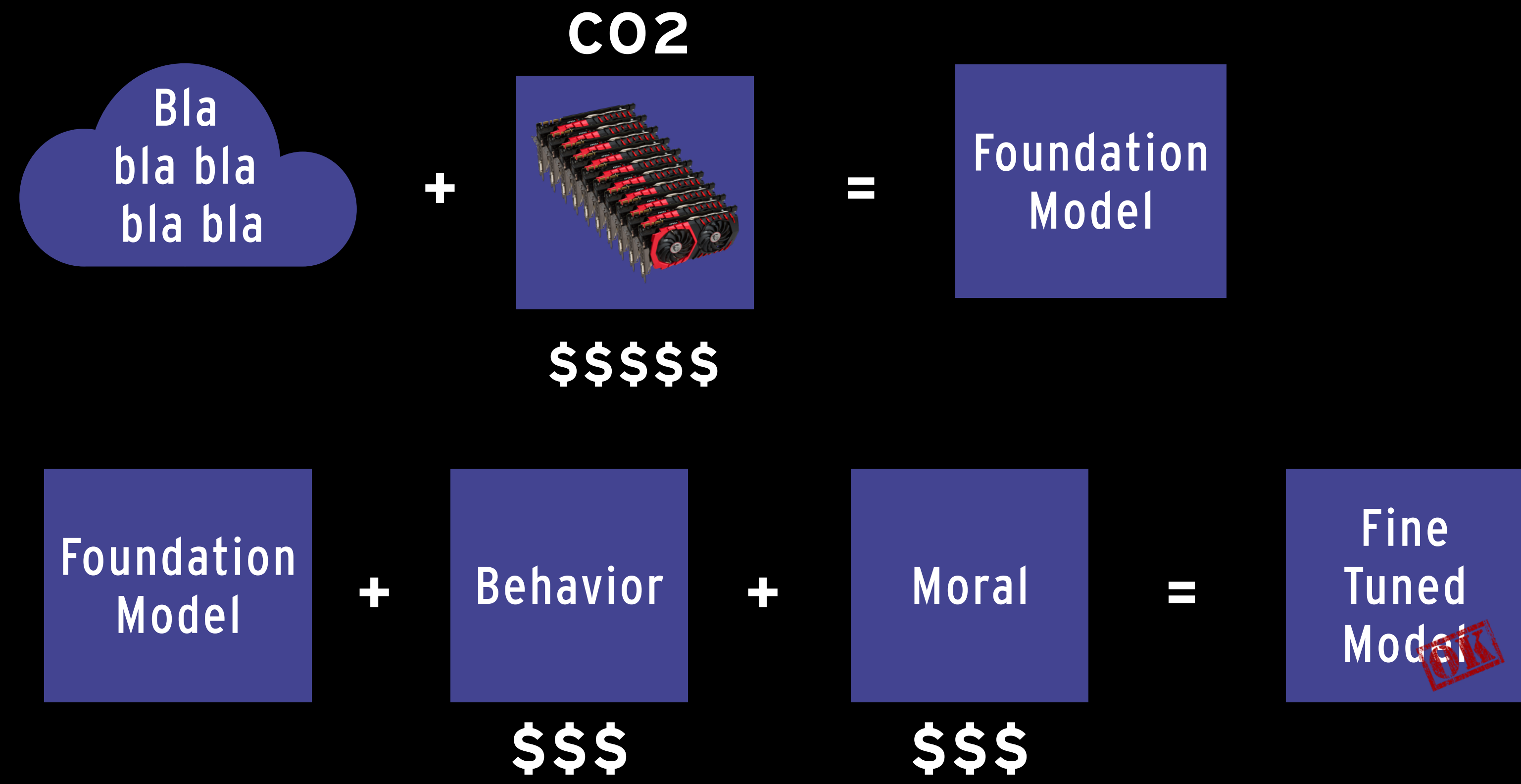
- Reinforcement Learning with Human Feedback
- Adds morals and adjusts for bad data in the foundation model
- Guard rails, Alignment
- Expensive and hard



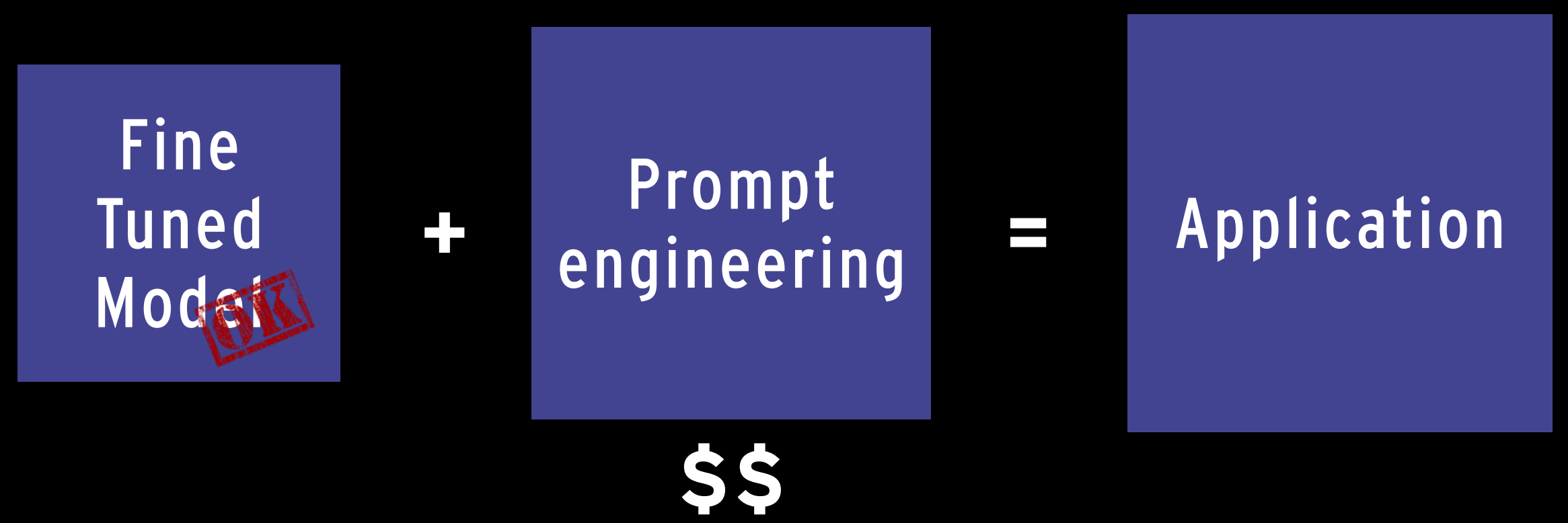
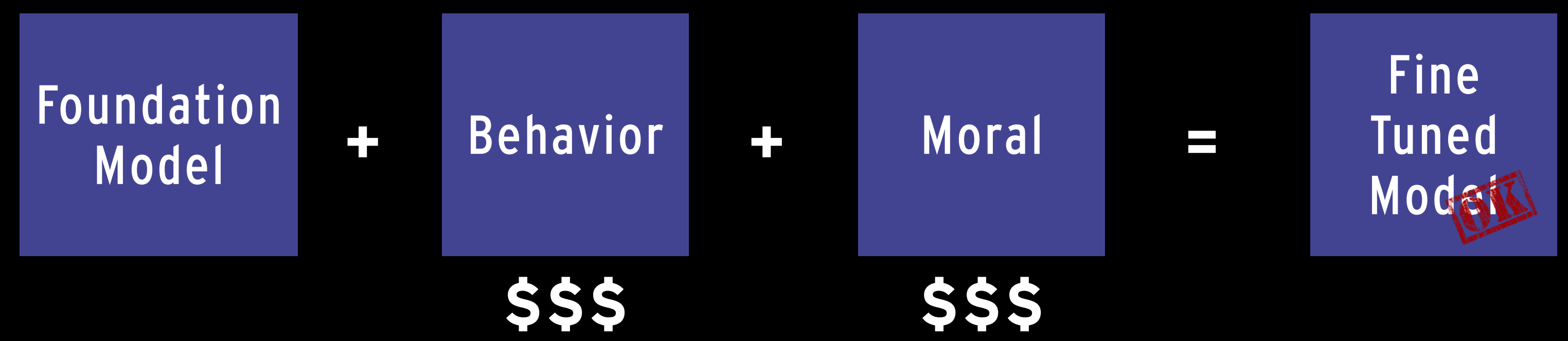
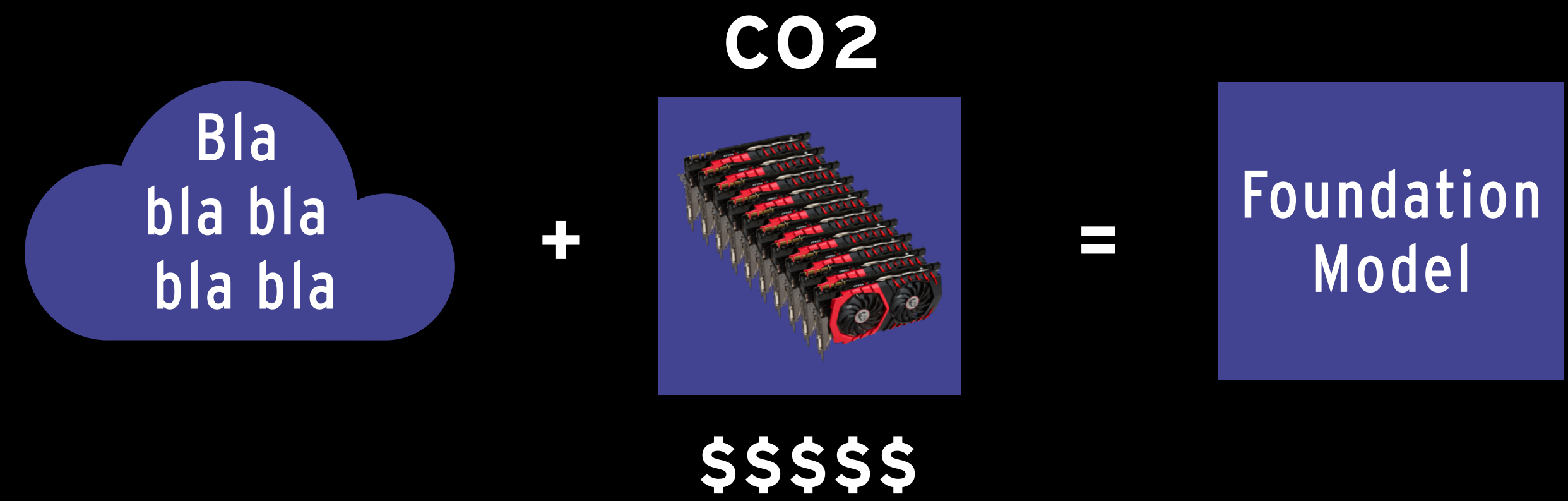
LARGE LANGUAGE MODELS - TRAINING AND COST SUMMARY



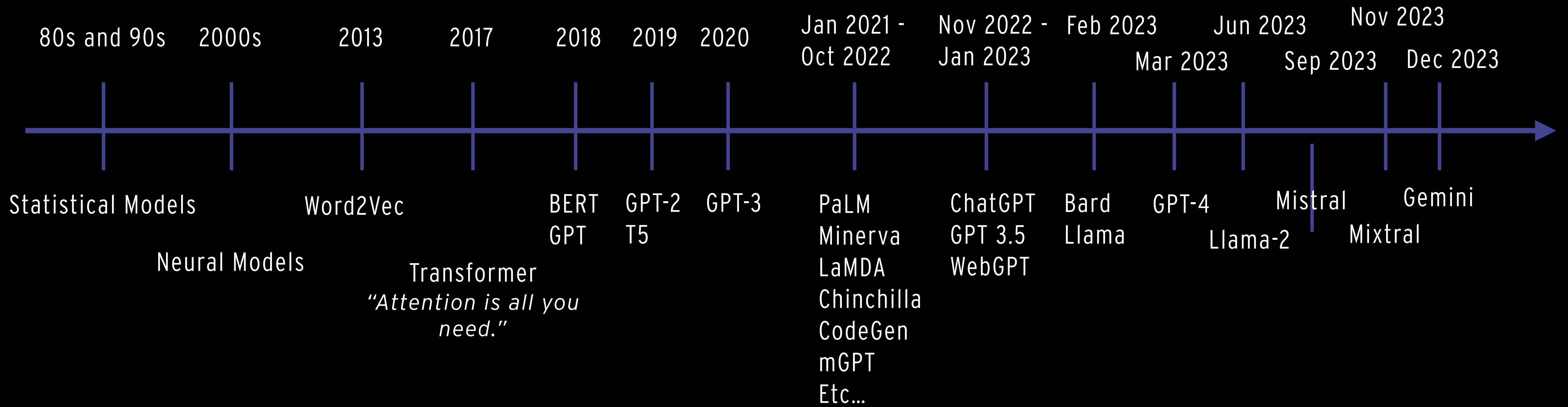
LARGE LANGUAGE MODELS - TRAINING AND COST SUMMARY



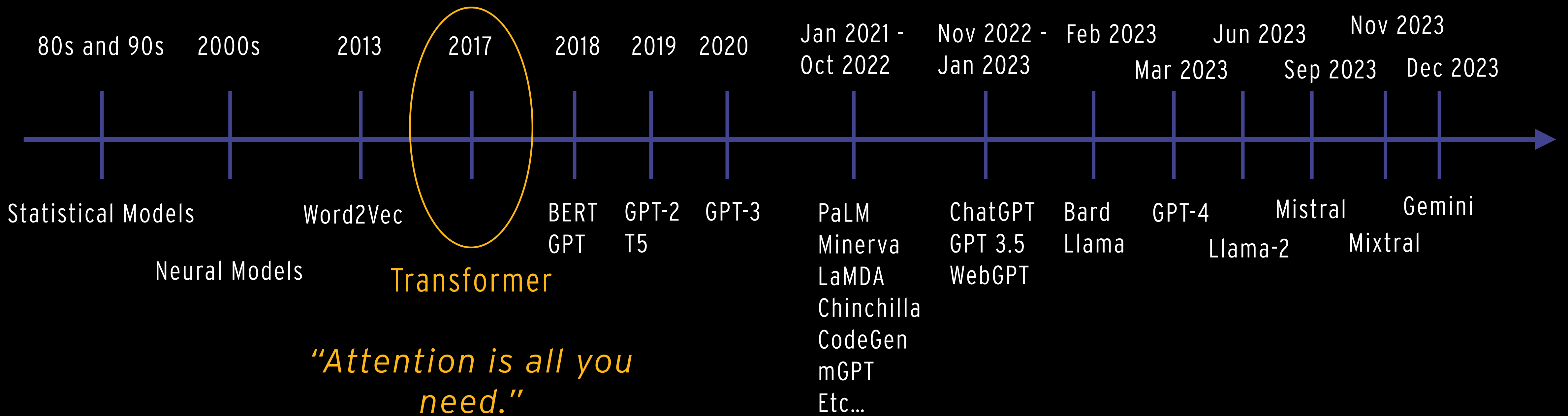
LARGE LANGUAGE MODELS - TRAINING AND COST SUMMARY



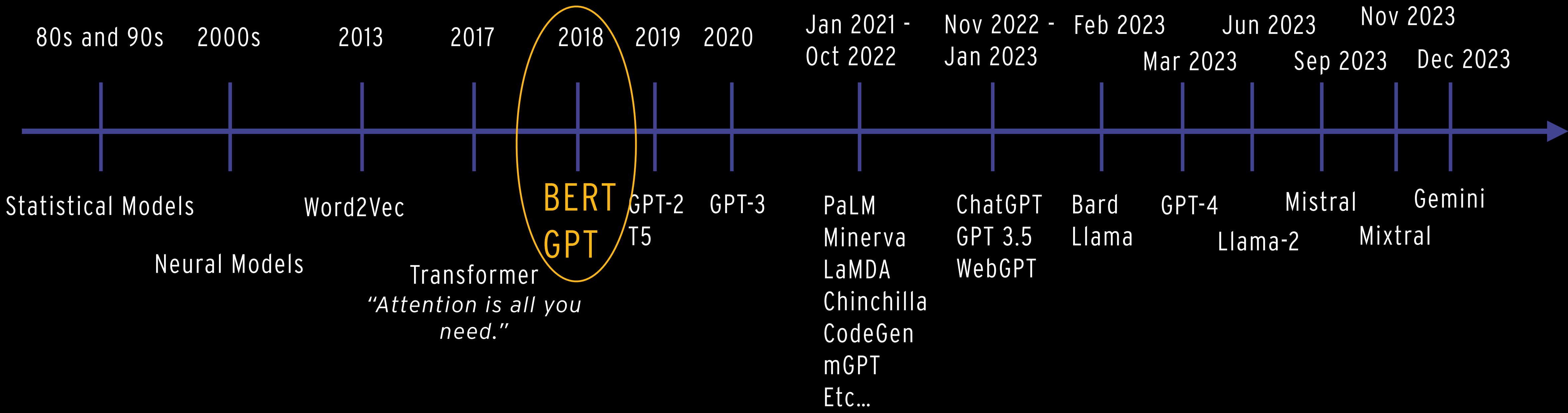
LANGUAGE MODELS - TIMELINE



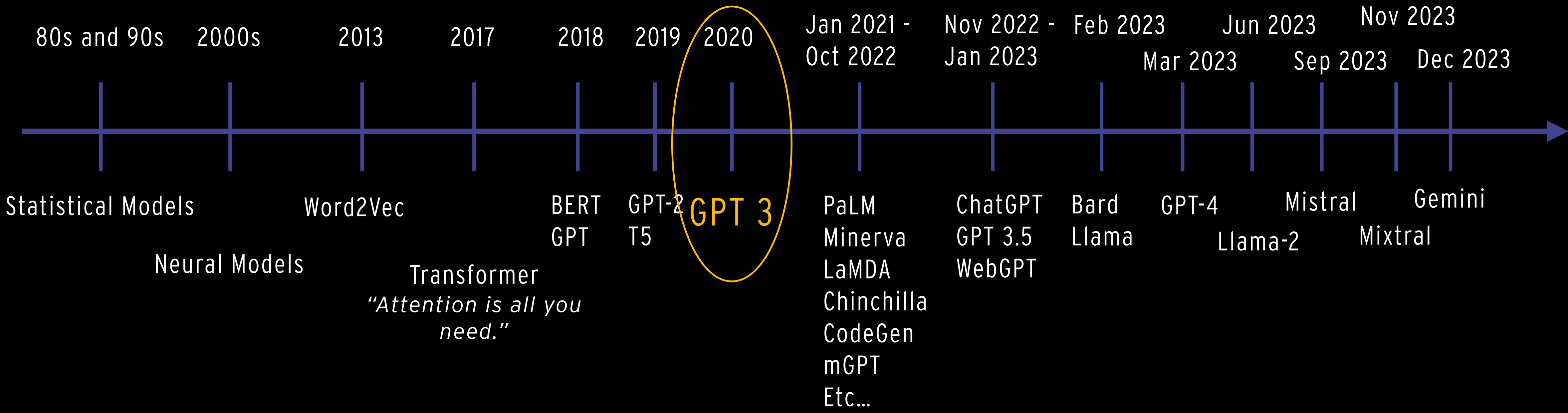
LANGUAGE MODELS - TIMELINE



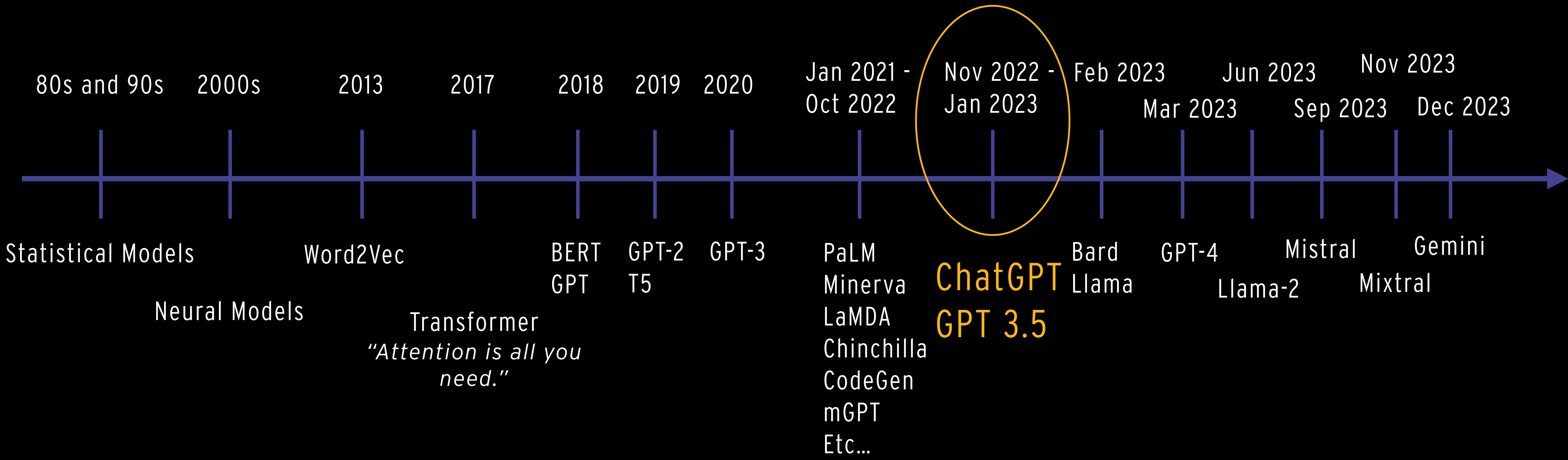
LANGUAGE MODELS - TIMELINE



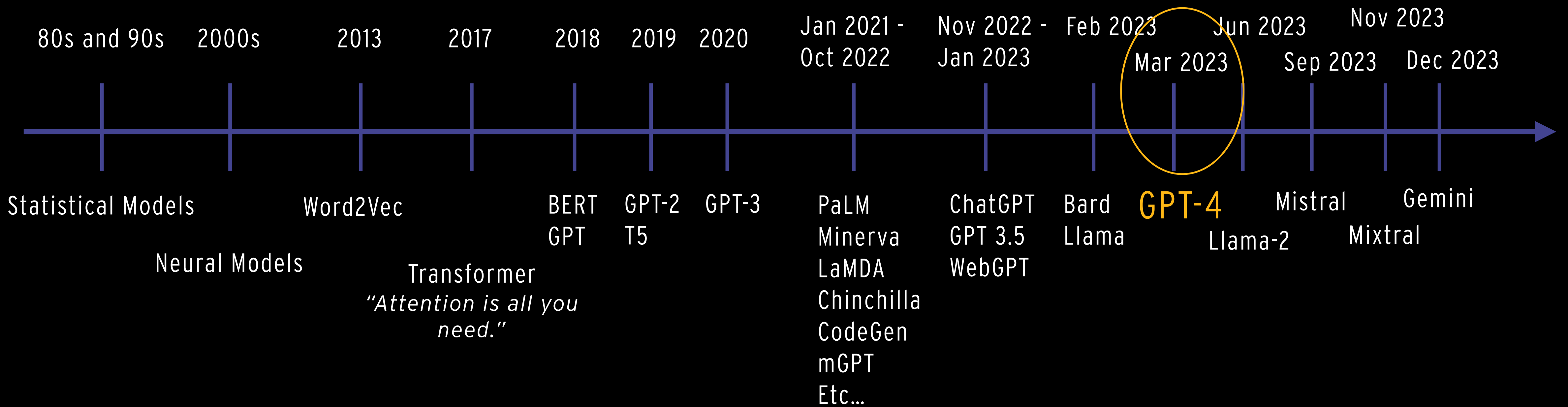
LANGUAGE MODELS - TIMELINE



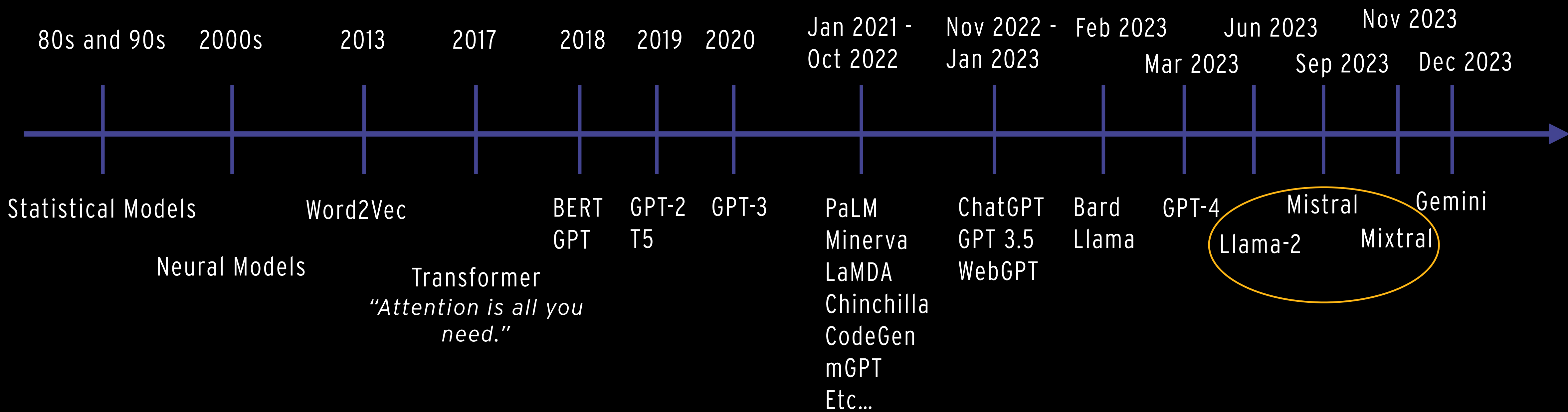
LANGUAGE MODELS - TIMELINE



LANGUAGE MODELS - TIMELINE



LANGUAGE MODELS - TIMELINE



| LARGE LANGUAGE MODEL - DEMO

- Foundation model:
 - Mistral-7B-base:
- Fine-tuned model:
 - Mistral-7B-instruct
- Open Source
- Runs locally on my Mac
- Context window 4096 tokens



DEMO 1

- LLM basic
- LLM autoregressive
- LLM hallucination (Magnus Larsson)

CHATGPT TO THE RESCUE?

ChatGPT 3.5 ▾



You

Who is Magnus Larsson who works for Callista Enterprise AB?



ChatGPT

I'm sorry, but I don't have access to specific information about individuals who work for private companies, especially if they are not publicly known figures. Magnus Larsson could be an employee at Callista Enterprise AB, but I wouldn't have any personal information about him or his role within the company. If you have a specific question or need information about Callista Enterprise AB as a company, I may be able to provide general information.



Is this conversation helpful so far?

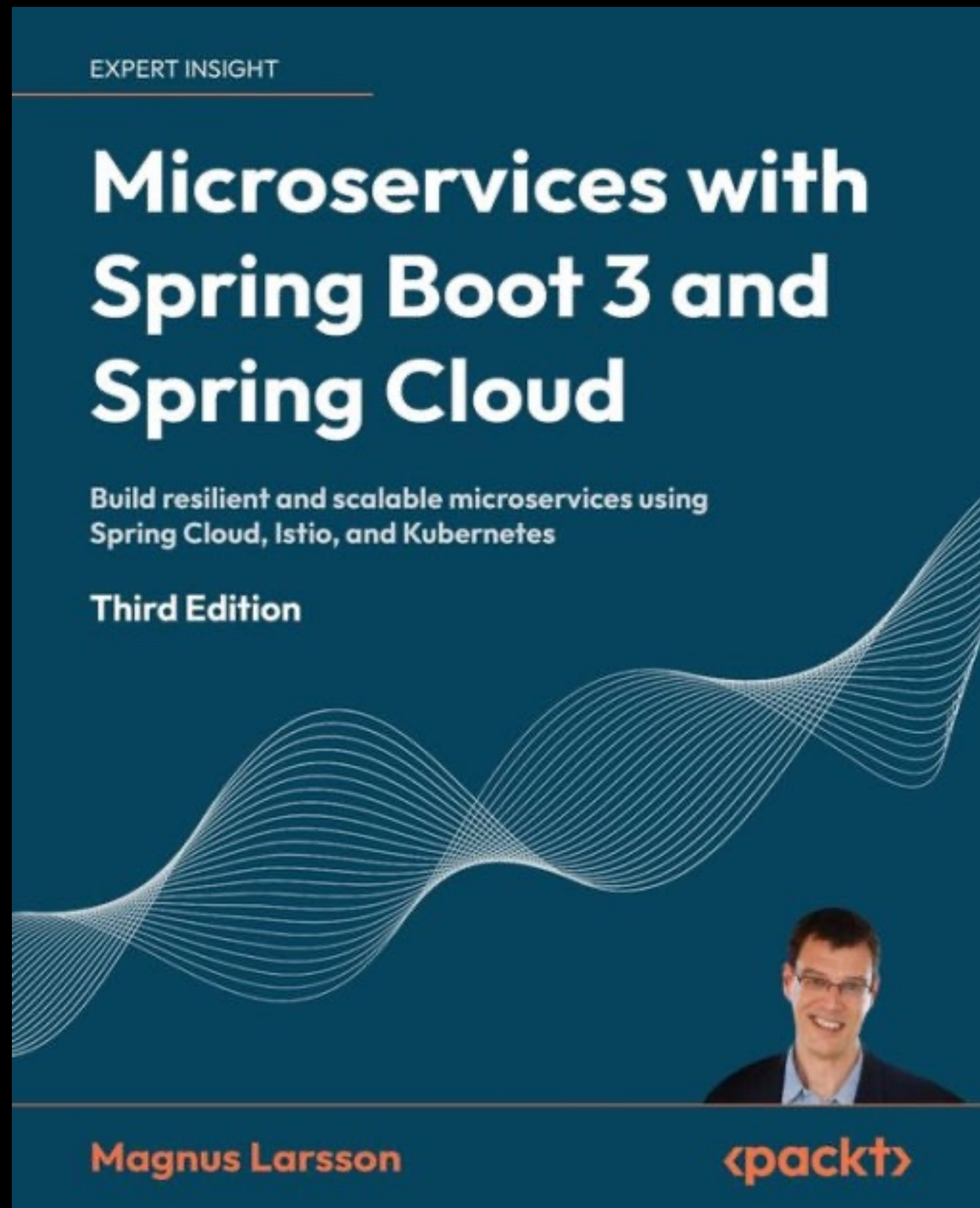


| RAG TO THE RESCUE

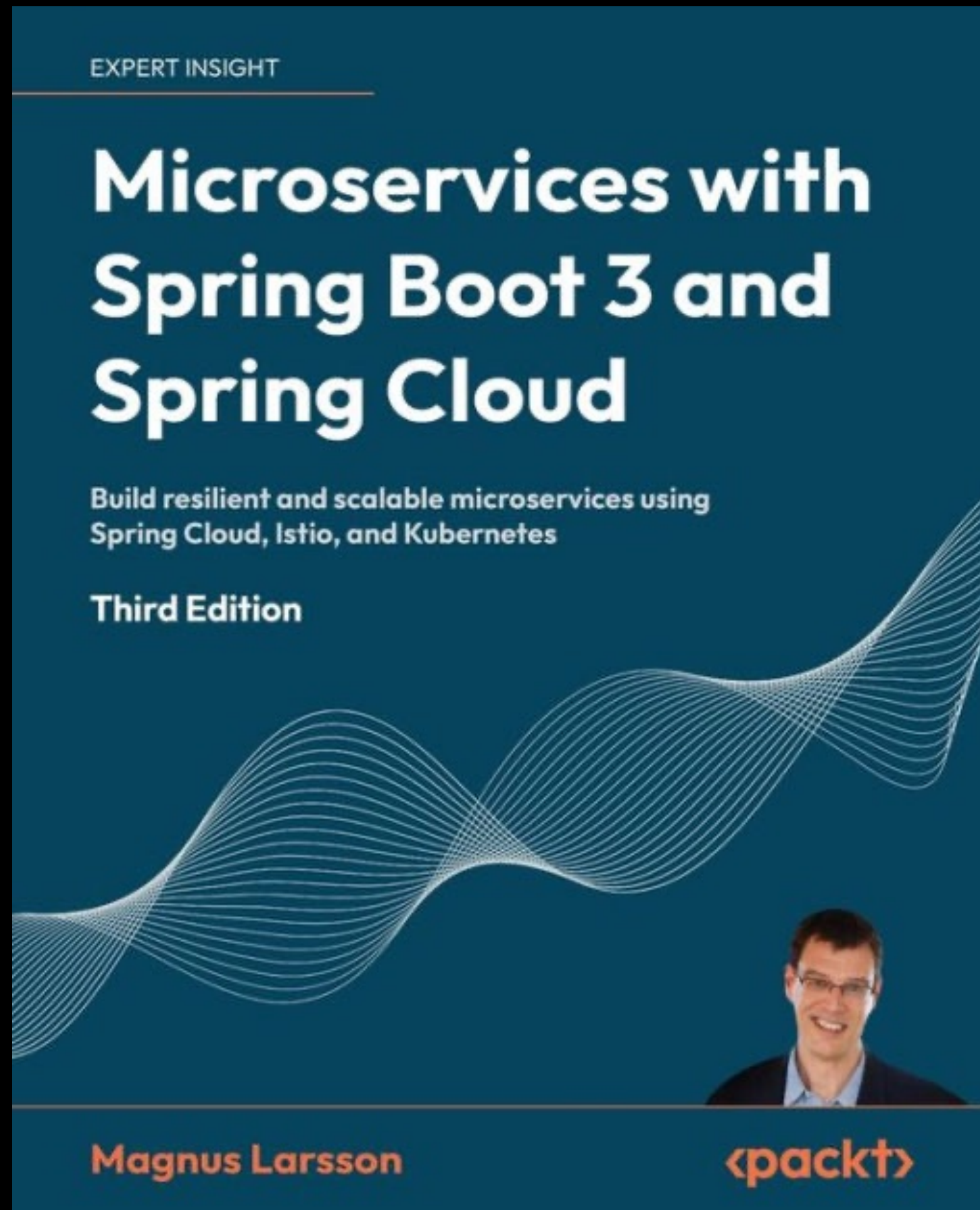
- Retrieval Augmented Generation
- Prompt Engineering with retrieval of content
- Use Embedding Model to get the juicy part of the content
- Use the LLMs generation and general language possibilities to ask or chat with a specific dataset.

- Use cases
 - Q&A
 - Chat (also add previous conversation to the context, needs memory)
 - Agent (give the model tools to do things)

DEMO APPLICATION - Q&A WITH A WELL KNOWN BOOK...



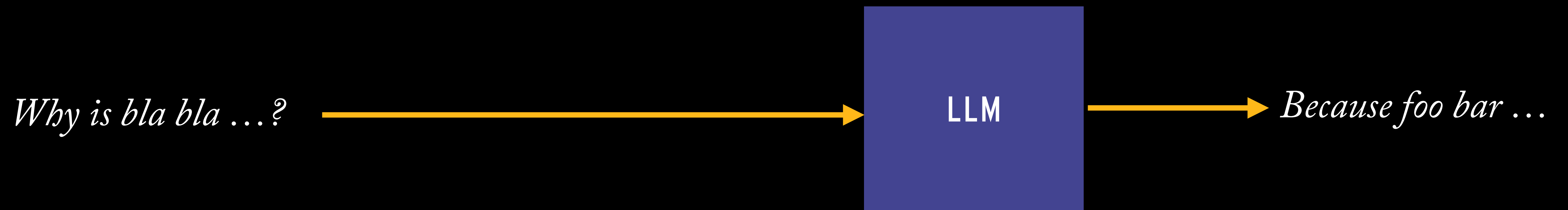
DEMO APPLICATION - Q&A WITH A WELL KNOWN BOOK...



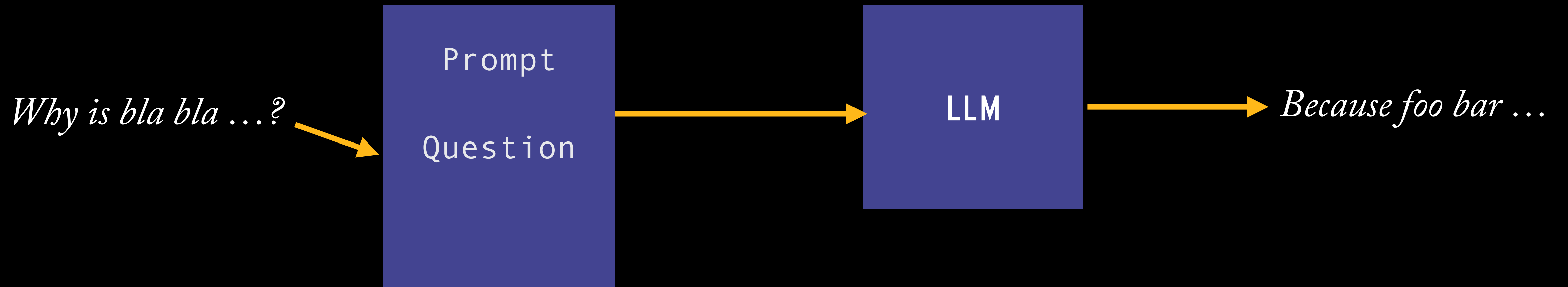
Who is Magnus Larsson?

What is the Spring AOT Smoke Tests project?

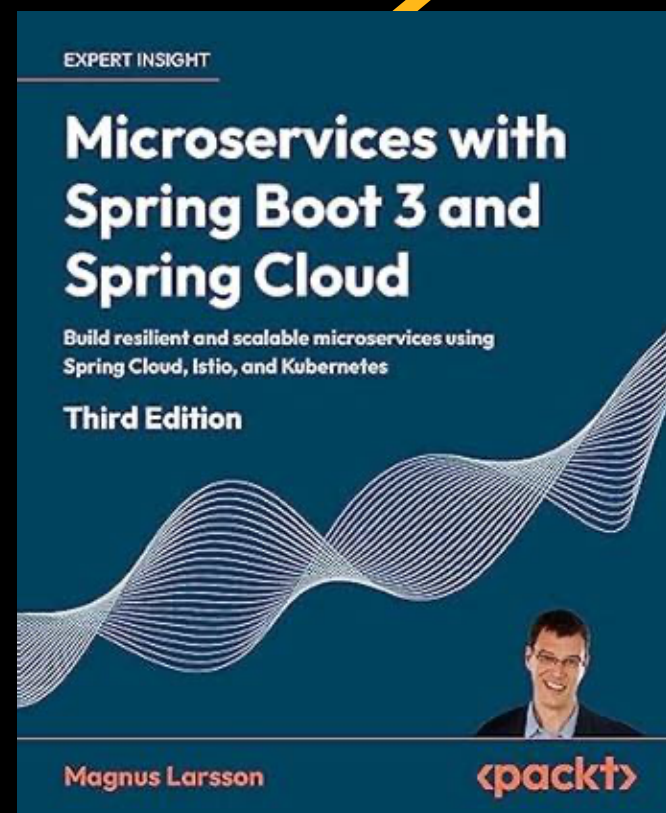
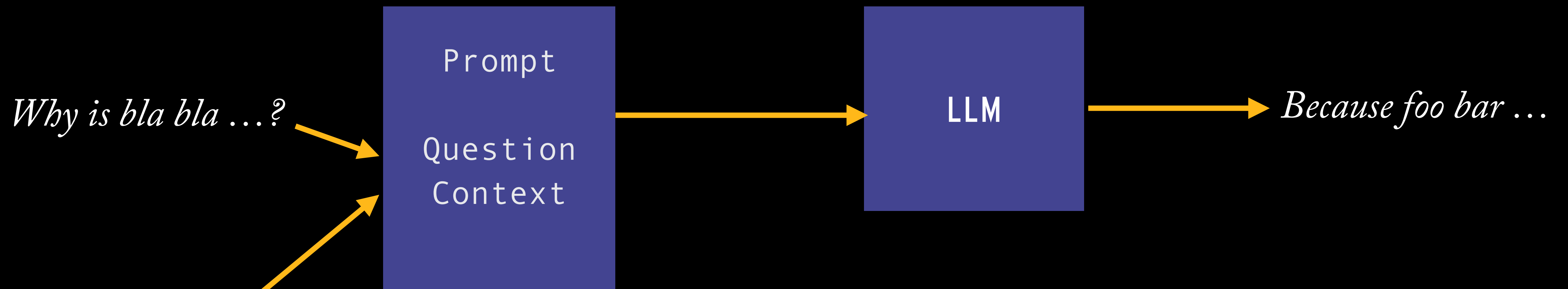
RAG - Q&A



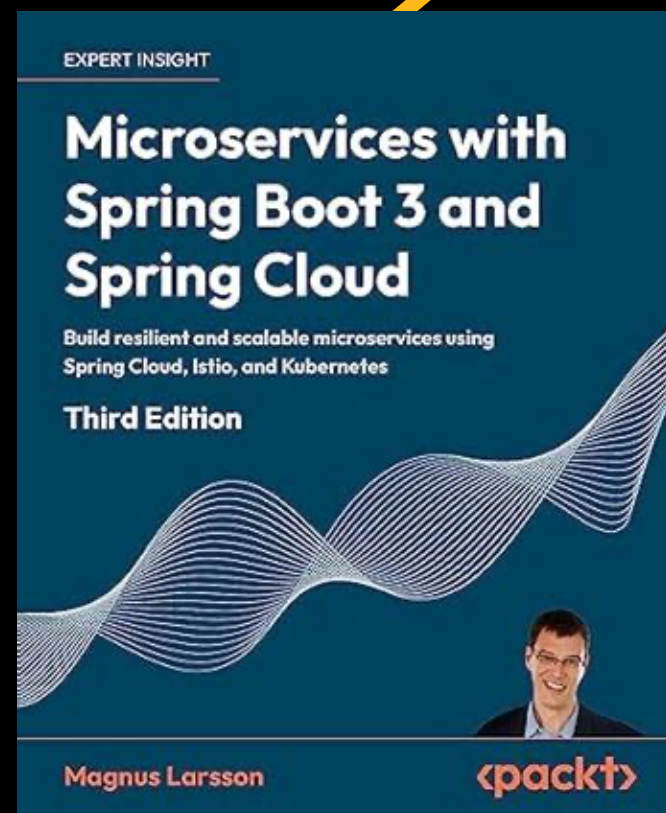
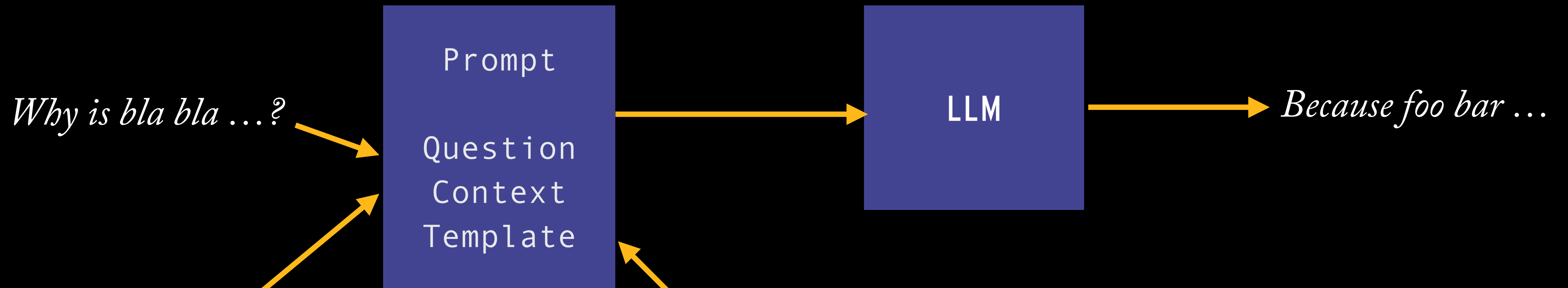
RAG - Q&A THE PROMPT



RAG - Q&A CONTEXT (TO THE QUESTION)



RAG - Q&A TEMPLATE



<Template>

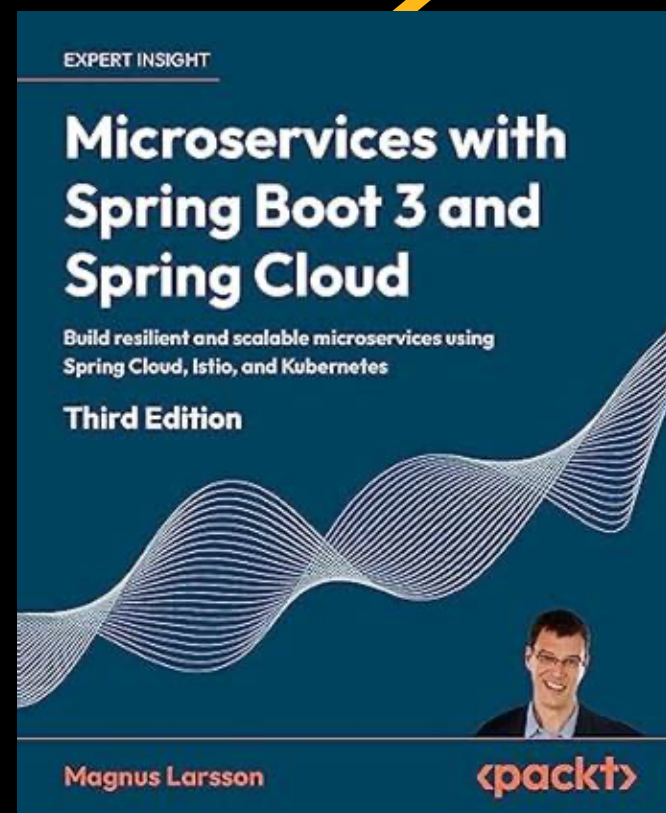
You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question. If you don't know the answer, just say that you don't know. Use five sentences maximum and keep the answer concise.

Question: {question}

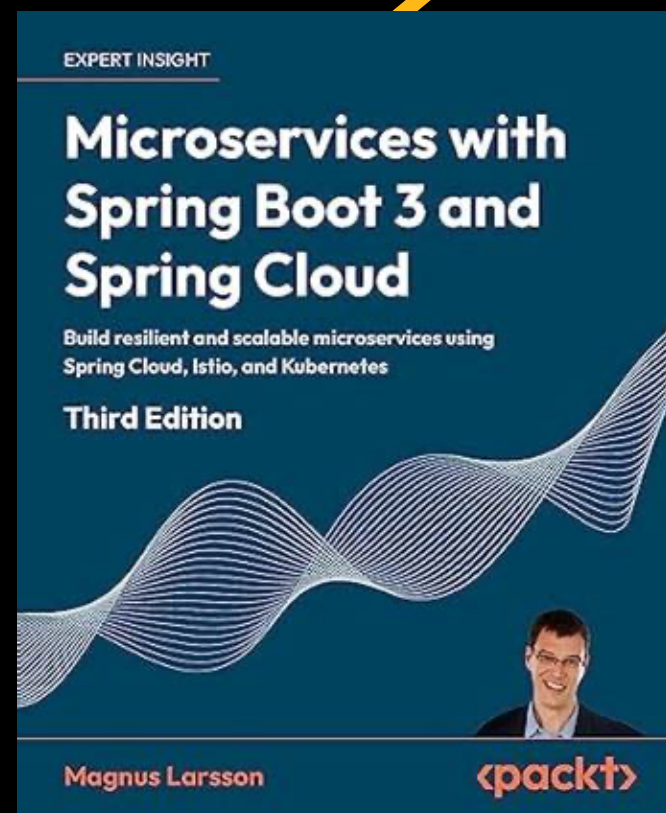
Context: {context}

Answer:

RAG - CONTENT



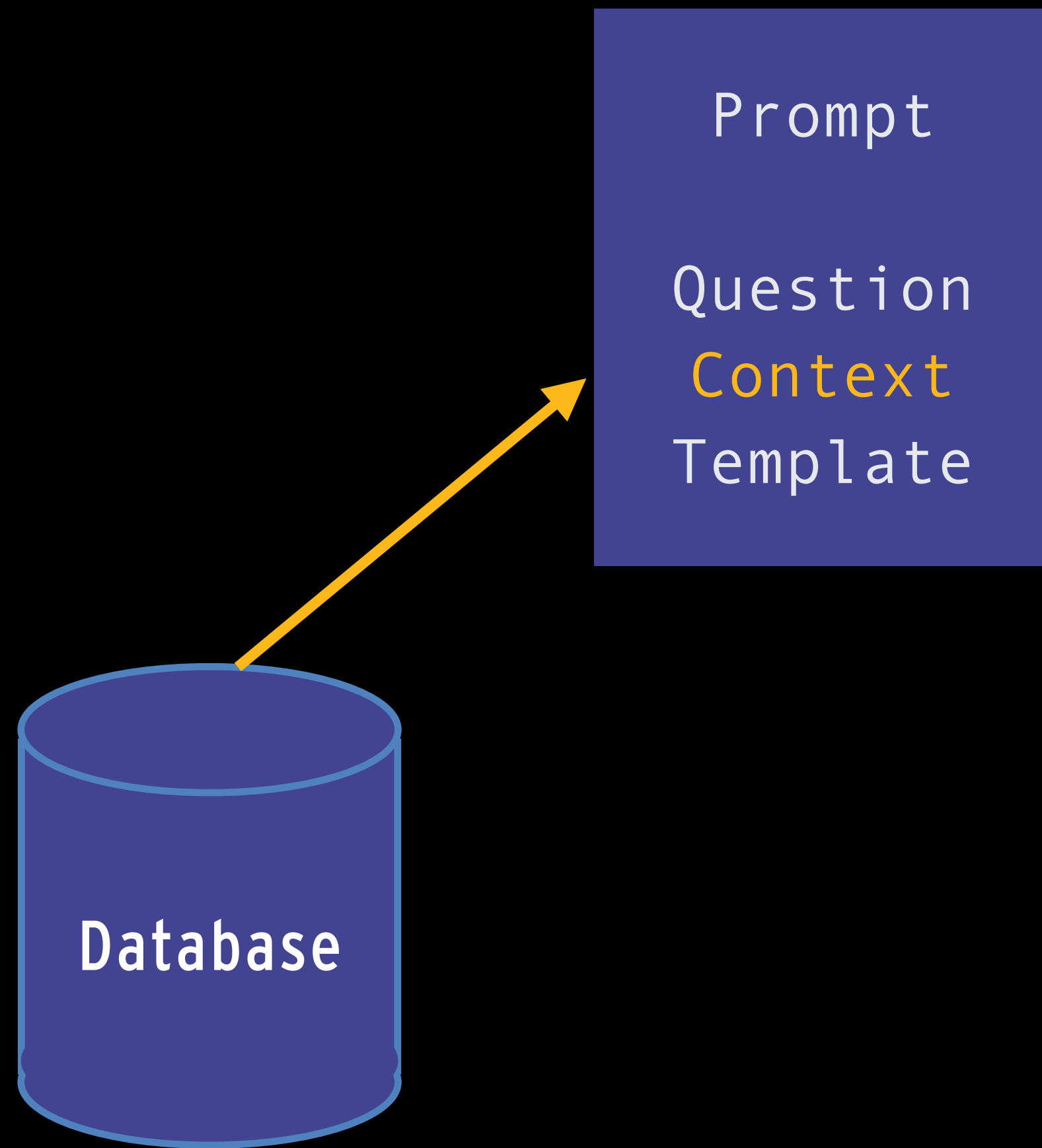
RAG - CONTENT



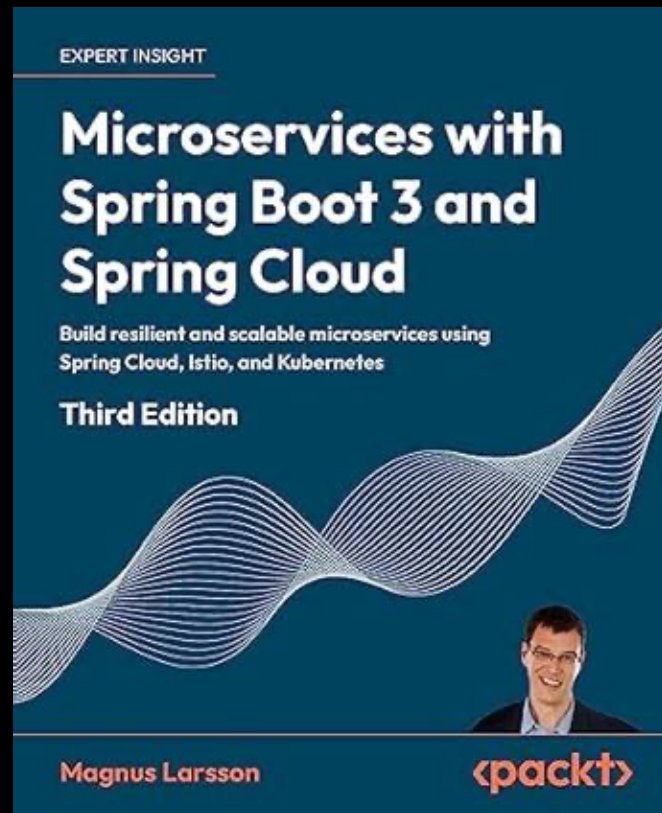
Magnus book has 255697 tokens

Mistral has context-window of 4096 tokens

RAG - CONTENT



| RAG - CREATING CONTENT PHASE

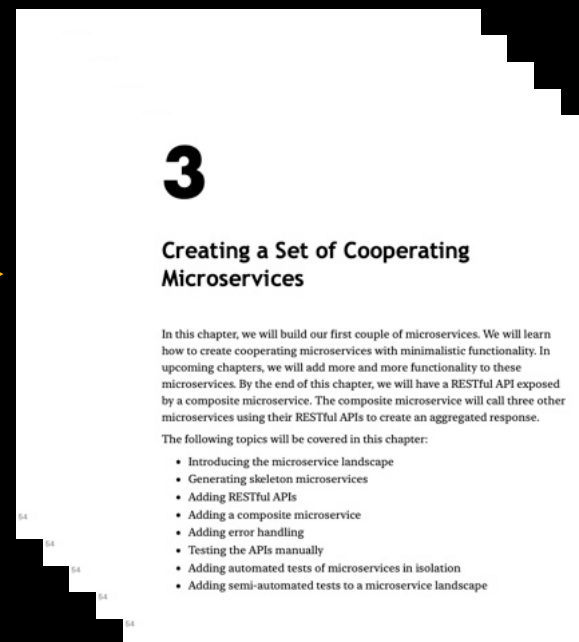
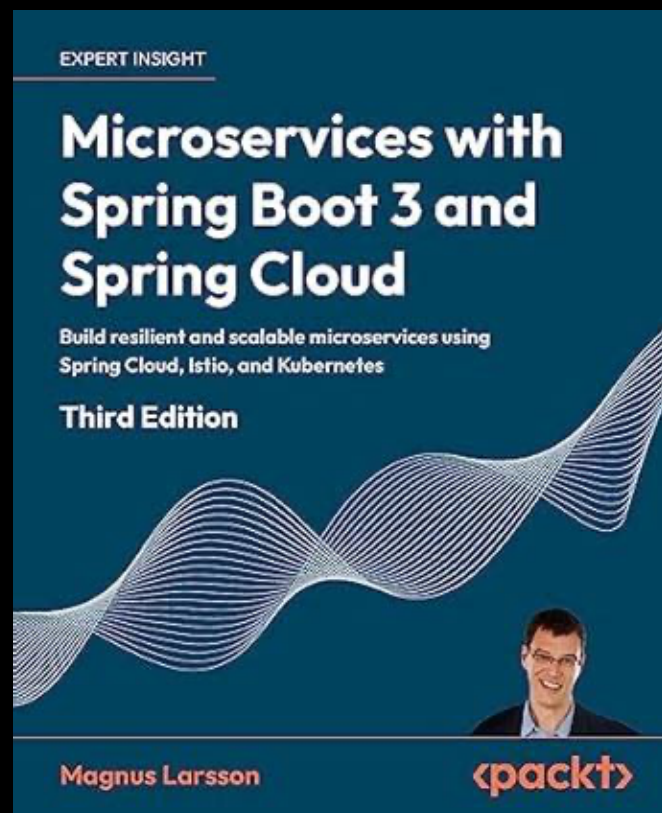


.epub

Zip with
XHTML chapter

RAG - CREATING CONTENT PHASE

Parse



.epub

Document

Zip with
XHTML chapter

Chapter Text
+
Metadata

23

Native-Compiled Java Microservices

In this chapter, we will learn how to compile the Java source code in our microservices into binary executable files, known as **Native Images**. A Native Image starts up significantly faster compared to using a Java VM and is also expected to consume less memory. We will be introduced to the **Spring AOT** engine introduced in Spring Framework 6 and the **GraalVM** project and its **Native Image compiler**, learning how to use them.

We will cover the following topics:

- When to natively compile Java source code
- Introducing the GraalVM project and Spring's AOT engine
- Handling problems with native compilation
- Testing and compiling Native Images
- Testing with Docker Compose
- Testing with Kubernetes

Even though Spring Framework 6 and Spring Boot 3 come with **General Availability (GA)** support for building native



executables of Spring Boot applications, it must be considered as being in an early stage. At the time of writing this chapter, a lot of pitfalls were discovered while natively compiling the microservices in this book. Since natively compiling the microservices is not required for the rest of the material in this book, this chapter is placed at the end of the book as an extra chapter, describing an exciting but not yet fully mature technology.

Technical requirements

For instructions on how to install the tools used in this book and how to access the source code for this book, see:

- *Chapter 21, Installation Instructions for macOS*
- *Chapter 22, Installation Instructions for Microsoft Windows with WSL 2 and Ubuntu*

The code examples in this chapter all come from the source code in `$BOOK_HOME/Chapter23`.

If you want to view the changes applied to the source code in this chapter so you can natively compile the microservices, you can compare it with the source code for *Chapter 20, Monitoring Microservices*. You can use your favorite `diff` tool and compare the two folders `$BOOK_HOME/Chapter20` and `$BOOK_HOME/Chapter23`.

When to native-compile Java source code

Java has always been known for its **build-once-run-anywhere** capability, providing excellent cross-platform support. The Java source code is compiled

THE DATA - METADATA

```
bookdemo > OEBPS > Text > <> Chapter_23.xhtml > ? > html > head
```

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html>
3
4  <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US" xmlns:epub="http://www.
5
6  <head>
7  | <title>Chapter_23</title>
8  | <link href=" ../Styles/syntax-highlighting.css" rel="stylesheet" type="text/css" />
9  | <link href=" ../Styles/epub.css" rel="stylesheet" type="text/css" />
10 </head>
11
12 <body id="Chapter_23">
13 | <div id=" idContainer587" class="Basic-Text-Frame">
14 | | <h1 class="chapterNumber">23</h1>
15 | | <h1 id="_idParaDest-550" class="chapterTitle">Native-Compiled Java Microservices</h1>
16 | | <p class="normal">In this chapter, we will learn how to compile the Java source code in our
17 | | binary executable<a id="_idIndexMarker1619"></a> files, known as <strong class="keyWord">N
18 | | Native Image starts up significantly faster compared to using a Java VM and is also expect
19 | | memory. We will be introduced to the <strong class="keyWord">Spring AOT </strong>engine<st
20 | | </strong>introduced in Spring Framework 6 and the <strong class="keyWord">GraalVM</strong>
21 | | class="keyWord">Native Image compiler</strong>, learning how to use them.</p>
22 | | <p class="normal">We will cover the following topics:</p>
23 | | <ul>
24 | | | <li class="bulletList">When to natively compile Java source code</li>
```

Chapter Number

Chapter Title

THE DATA - CONTENT

```
14 def chapter_to_str(soup):
15     text = []
16     for tag in soup.find_all(["p", "span", "a", "ol", "ul", "li", "h1", "h2", "h3", "h4", "code"]):
17         if tag.name in ["p", "h1", "h2", "h3", "h4"]:
18             text.append("\n\n" + tag.get_text())
19         if tag.name in ["li"]:
20             text.append("\n" + tag.get_text())
21     return " ".join(text)
22
```

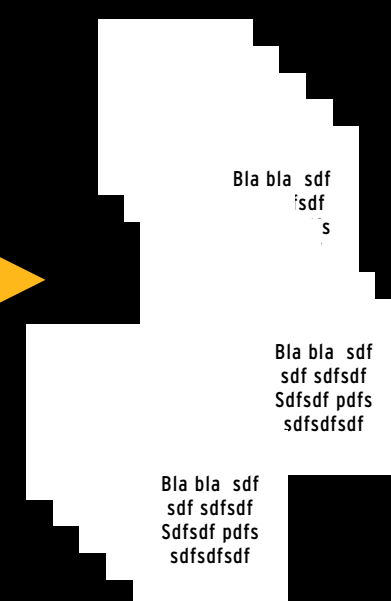
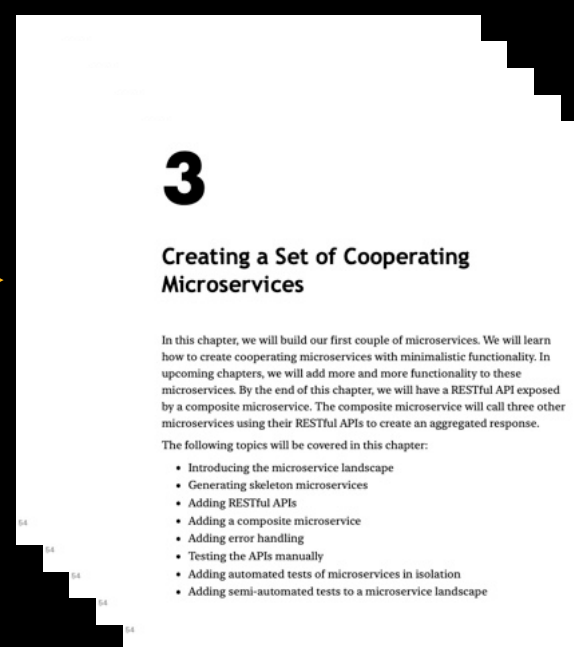
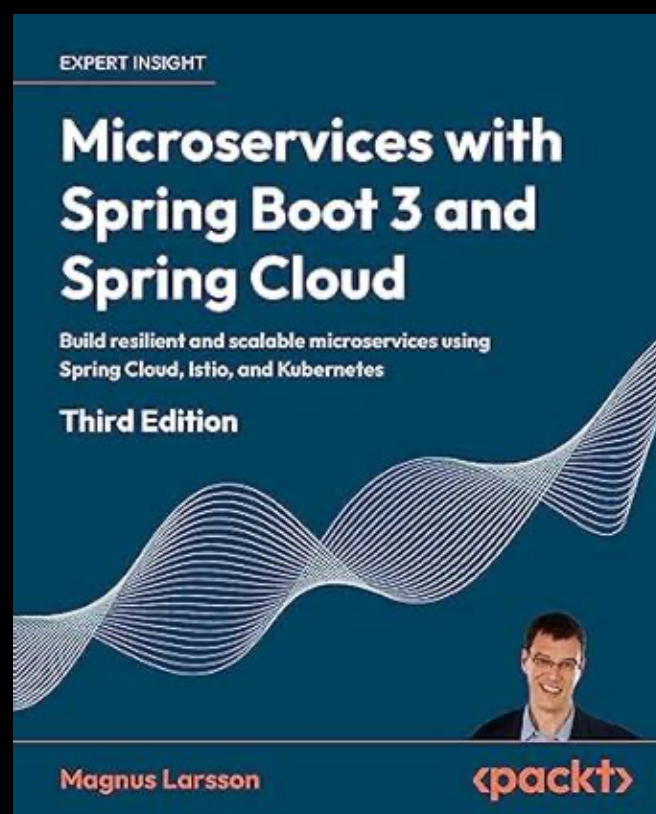
THE DATA - RESULT

```
bookdemo > ≡ 23.txt
1
2 23
3
4 Native-Compiled Java Microservices
5
6 In this chapter, we will learn how to compile the Java source code in our microservices into binary executables.
7
8 We will cover the following topics:
9 When to natively compile Java source code
10 Introducing the GraalVM project and Spring's AOT engine
11 Handling problems with native compilation
12 Testing and compiling Native Images
13 Testing with Docker Compose
14 Testing with Kubernetes
15
16 Even though Spring Framework 6 and Spring Boot 3 come with General Availability (GA) support for building native binaries, there are still some technical requirements that need to be met.
17
18 Technical requirements
19
20 For instructions on how to install the tools used in this book and how to access the source code for this chapter, see:
21 Chapter 21, Installation Instructions for macOS
22 Chapter 22, Installation Instructions for Microsoft Windows with WSL 2 and Ubuntu
23
```

RAG - CREATING CONTENT PHASE

Parse

Split



.epub

Document

Chunk

Zip with
XHTML chapter

Chapter Text
+
Metadata

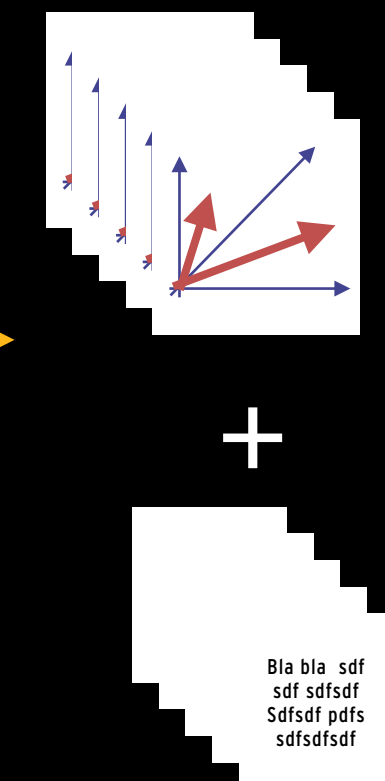
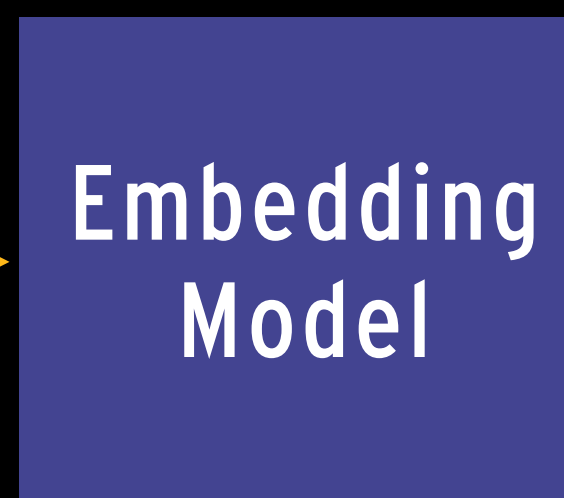
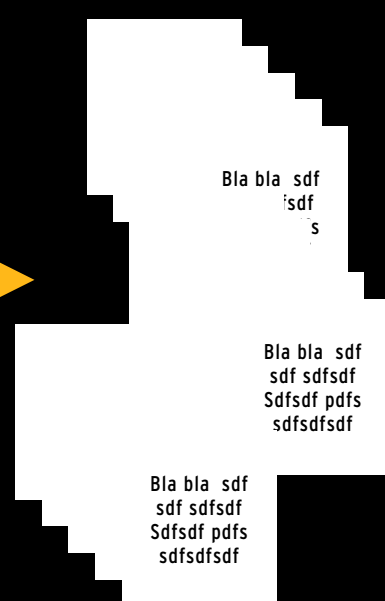
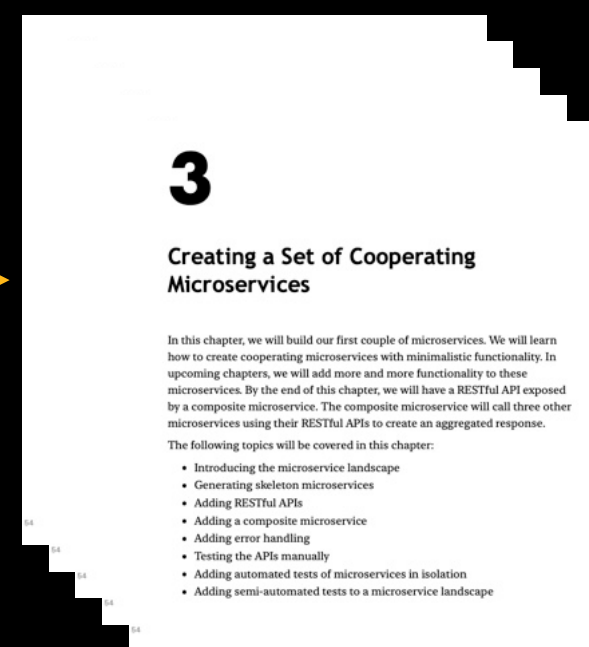
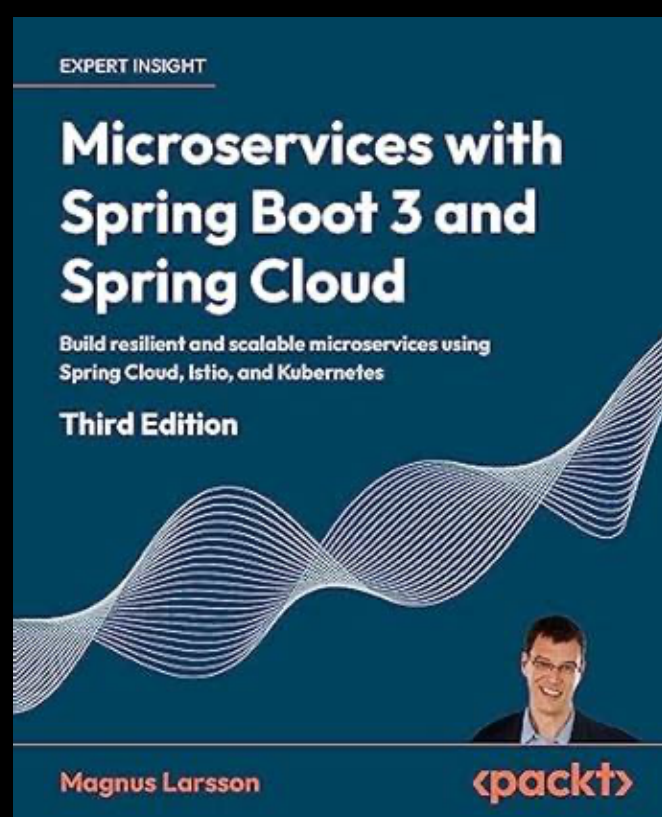
Text
+
Metadata

RAG - CREATING CONTENT PHASE

Parse

Split

Embedd



.epub

Document

Chunk

Embedding + Chunk

Zip with
XHTML chapter

Chapter Text
+
Metadata

Text
+
Metadata

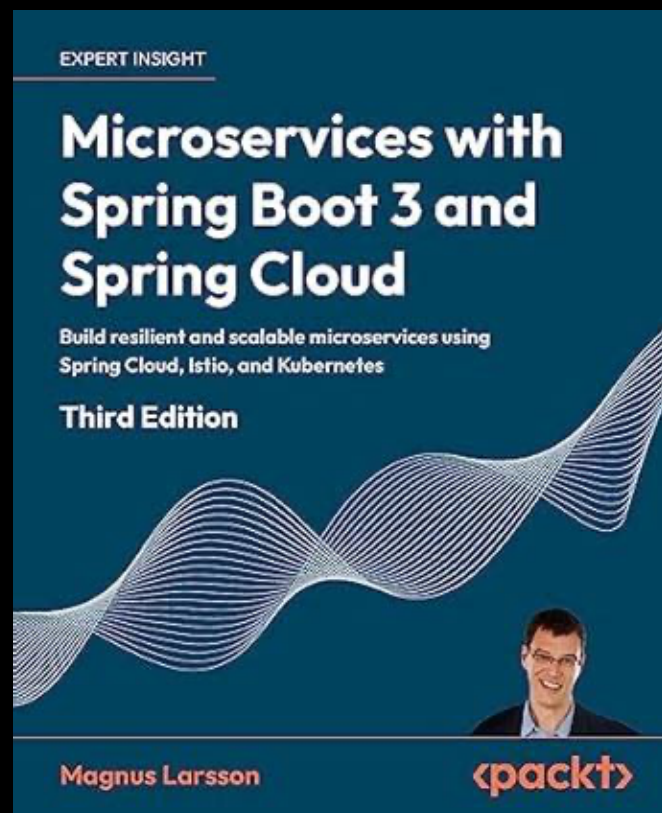
Vector

| RAG - EMBEDDING MODEL

- BGE-large-en-v1.5 SOTA
- BAAI General Embedding (Beijing Academy of AI)
- Masked Auto-Encoder
- 1024 dimensions

RAG - CREATING CONTENT PHASE

Parse

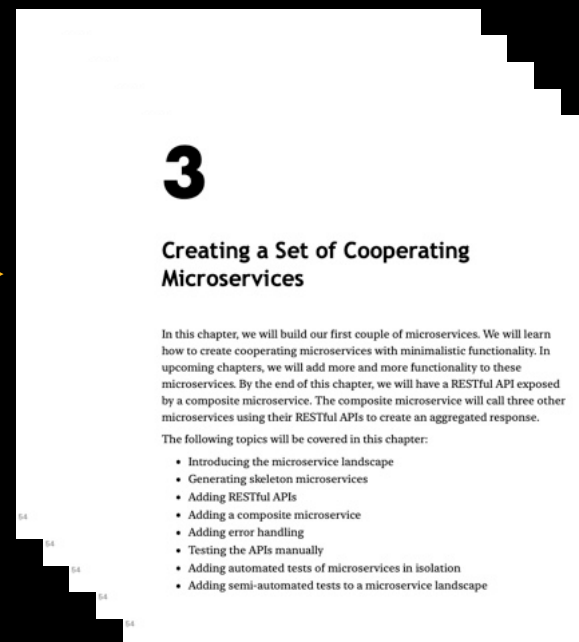


.epub

Zip with
XHTML chapter



Split

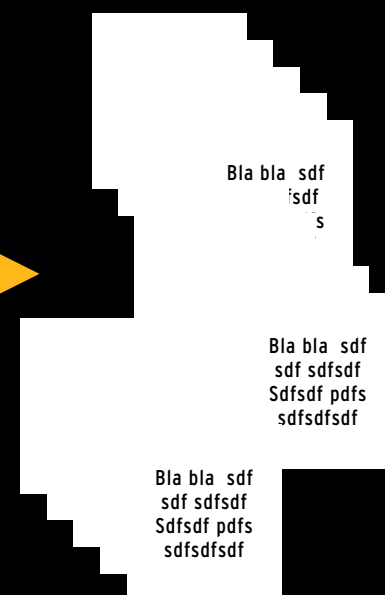


Document

Chapter Text
+
Metadata



Embedd



Chunk

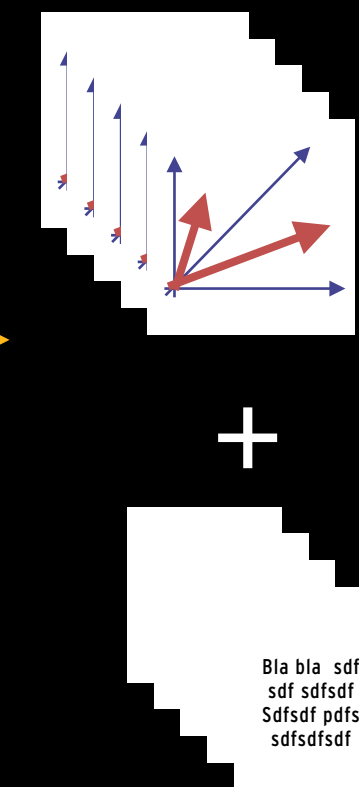
Text
+
Metadata



Embedding
Model

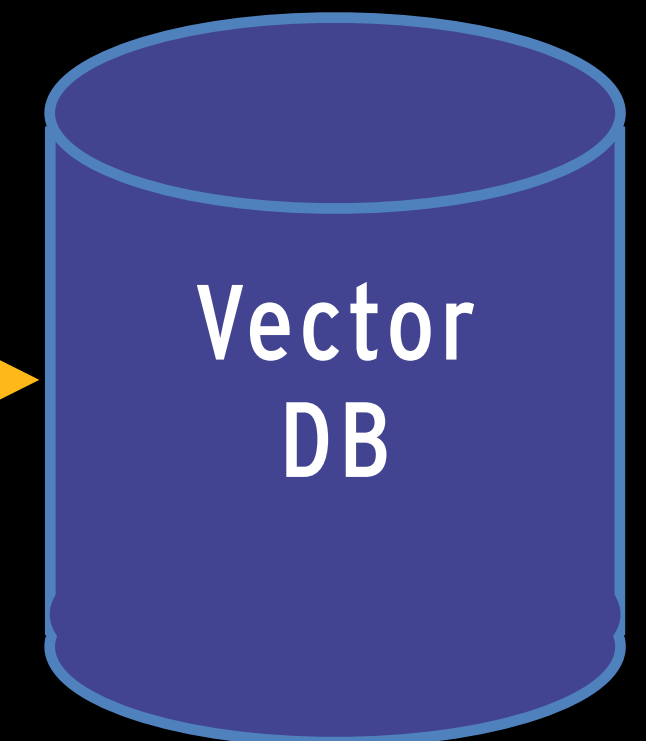


Store



Embedding + Chunk

Vector



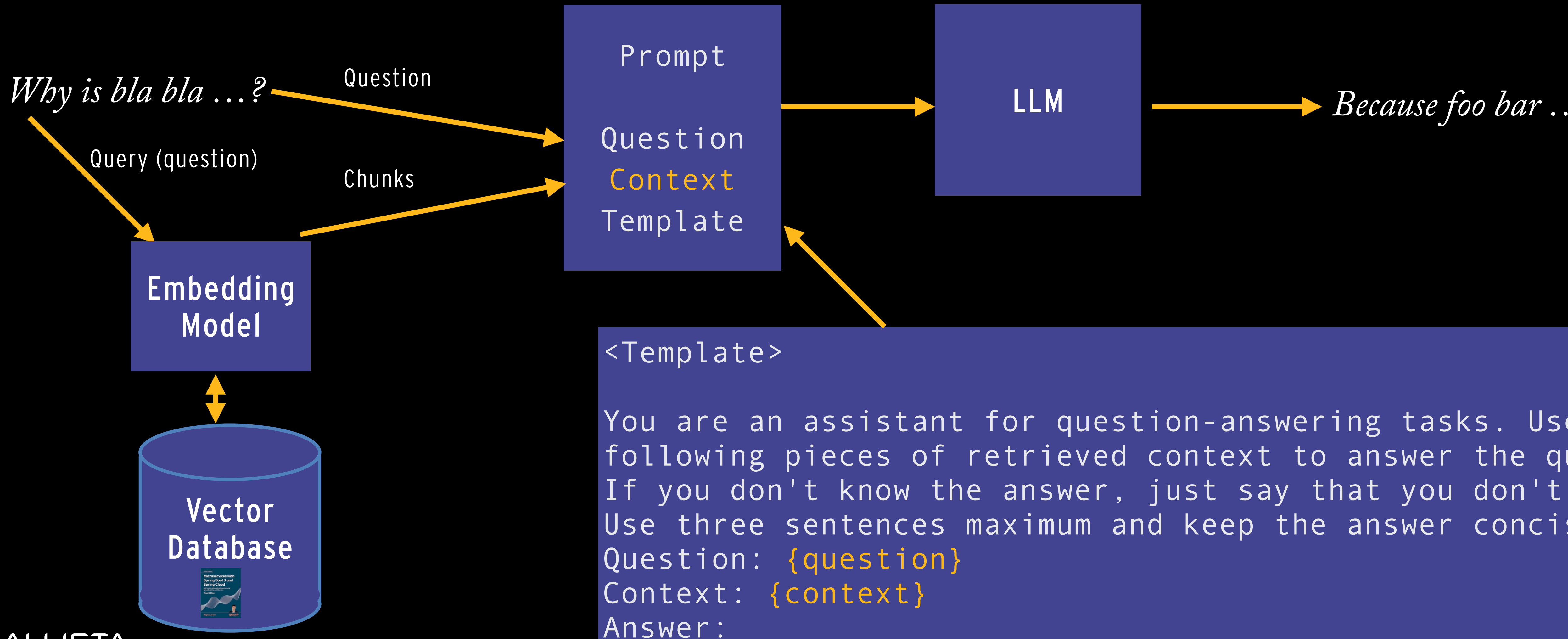
RAG - STORE

```
cadec2024 — antonic@LittleBlackMBP — ..sta/cadec2024 — -zsh — 82x21
(llamacpp) → cadec2024 git:(main) x python bookdemo/vector_store.py
Parsed bookdemo/magnus_3.epub into 25 chapter documents taking 0.7122 seconds.
Splitted into 1362, chunks took: 0.0077 seconds
Embedding chunks...
Created new collection magnus_3.epub_hf_bge_large in 39.2392 seconds.
(llamacpp) → cadec2024 git:(main) x █
```

| RAG - QUERYING THE EMBEDDED CHUNKS

- Use the embedding model
- Semantic Search
- Demo

RAG - GETTING THE CONTENT INTO THE PROMPT



<Template>

You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question. If you don't know the answer, just say that you don't know. Use three sentences maximum and keep the answer concise.

Question: {question}
Context: {context}
Answer:

| RAG - DEMO

- Who is Magnus Larsson?
- What is the Spring AOT smoke tests project?

| RAG - DEMO 1

- Who is Magnus Larsson?

RAG - DEMO 1

Question:

Who is Magnus Larsson?

Result:

Magnus Larsson is an IT industry veteran with over 30 years of experience in consulting for major Swedish firms like Volvo, Ericsson, and AstraZeneca. He has expertise in distributed systems and open-source tools such as Spring Cloud, Kubernetes, and Istio. Magnus has been helping customers use these tools for the past eight years and has shared his insights through presentations and blog posts.

Context from following chapters:

Score	Chapter	Content
0.60	TitlePage	ISBN 978-1-80512-869-4 www.packt.com Contri

81

82 About the
83 author

84

85 Magnus Larsson, an IT
86 industry veteran since 1986, has consulted for major Swedish firms like
87 Volvo, Ericsson, and AstraZeneca. Despite past struggles with
88 distributed systems, today's open-source tools like Spring Cloud,
89 Kubernetes, and Istio offer effective solutions. For the past eight
90 years, Magnus has been helping customers use these tools and has shared
91 his insights through presentations and blog posts.

92

93 I would like to thank the following people:

| RAG - DEMO 2

- What is the Spring AOT smoke tests project?

CHATGPT - DEMO 2

ChatGPT 3.5 ▾



You

What is the Spring AOT smoke test project?



ChatGPT

As of my last knowledge update in January 2022, I don't have specific information about a "Spring AOT smoke test project." However, I can provide some context that might help you understand the individual components involved.

1. **Spring:** Spring is a popular open-source framework for building Java-based enterprise applications. It provides a wide range of features and modules for developing scalable and maintainable applications.
2. **AOT:** AOT stands for Ahead-Of-Time compilation. It's a technique used in software development where code is compiled into machine code or a lower-level intermediate representation before runtime, as opposed to the more common Just-In-Time (JIT) compilation, where code is compiled at runtime. AOT compilation can lead to improved performance and reduced startup times for applications.

RAG - DEMO 2

Question:

What is the Spring AOT smoke test project?

Result:

The Spring AOT smoke test project is a suite of tests verifying that various Spring projects work when natively compiled. It is available at <https://github.com/spring-projects/spring-aot-smoke-tests> and can be used to report problems with native compilation of Spring projects. Native tests can be started using Gradle with the command "gradle nativeTest".

Context from following chapters:

Score	Chapter	Content
0.79	23	This project contains a suite of tests verifying t

108 The following project and tools can be used to handle problems with native
109 compilation of Spring Boot applications:

110

111 Spring AOT smoke tests: This project contains a suite of tests verifying that
112 the various Spring projects work when natively compiled. Whenever you encounter
113 issues with natively compiling a Spring feature, you should start looking into
114 this project for a working solution. Also, if you want to report a problem with
115 natively compiling a Spring project, you can use tests from this project as a
116 boilerplate to demonstrate the problem in a reproducible way. The project is
117 available at <https://github.com/spring-projects/spring-aot-smoke-tests>. Test
118 results can be found in Spring's CI environment. For example, the tests of the
119 various Spring Cloud projects can be found here: <https://ci.spring.io/teams/spring->

RAG - DEMO 2

Question:

What is the Spring AOT smoke test project?

Result:

The Spring AOT smoke test project is a suite of tests verifying that various Spring projects work when natively compiled. It is available at <https://github.com/spring-projects/spring-aot-smoke-tests> and can be used to report problems with native compilation of Spring projects. Native tests can be started using Gradle with the command "gradle nativeTest".

Context from following chapters:

Score	Chapter	Content
0.79	23	This project contains a suite of tests verifying t

108 The following project and tools can be used to handle problems with native
109 compilation of Spring Boot applications:

110

111 Spring AOT smoke tests: This project contains a suite of tests verifying that
112 the various Spring projects work when natively compiled. Whenever you encounter
113 issues with natively compiling a Spring feature, you should start looking into
114 this project for a working solution. Also, if you want to report a problem with
115 natively compiling a Spring project, you can use tests from this project as a
116 boilerplate to demonstrate the problem in a reproducible way. The project is
117 available at <https://github.com/spring-projects/spring-aot-smoke-tests>. Test
118 results can be found in Spring's CI environment. For example, the tests of the
119 various Spring Cloud projects can be found here: <https://ci.spring.io/teams/spring->

120

RAG - DEMO 2

Question:

What is the Spring AOT smoke test project?

Result:

The Spring AOT smoke test project is a suite of tests verifying that various Spring projects work when natively compiled. It is available at <https://github.com/spring-projects/spring-aot-smoke-tests> and can be used to report problems with native compilation of Spring projects. Native tests can be started using Gradle with the command "gradle nativeTest".

Context from following chapters:

Score	Chapter	Content
0.79	23	This project contains a suite of tests verifying t

107
108 The following project and tools can be used to handle problems with native
109 compilation of Spring Boot applications:

110

111 Spring AOT smoke tests: This project contains a suite of tests verifying that
112 the various Spring projects work when natively compiled. Whenever you encounter
113 issues with natively compiling a Spring feature, you should start looking into
114 this project for a working solution. Also, if you want to report a problem with
115 natively compiling a Spring project, you can use tests from this project as a
116 boilerplate to demonstrate the problem in a reproducible way. The project is
117 available at <https://github.com/spring-projects/spring-aot-smoke-tests>. Test
118 results can be found in Spring's CI environment. For example, the tests of the
119 various Spring Cloud projects can be found here: <https://ci.spring.io/teams/spring->

120

| WHERE TO GO NEXT?

- Reranking
- Guardrails
- Testing

- Chat
- Agents

PRODUCTIFY

- Where to host the model?
- Self hosting needs nodes with GPU
- Model as a Service (MaaS)
 - OpenAI (GPT-models)
 - Microsoft (Llama-2)
 - Mistral (Mistral-Medium)
 - ... the list is long

CONCLUSIONS

- LLM's are here to stay, learn to build applications with them.
- Know your data
- Learn your LLM
- Learn your Embedding model
- Don't forget trustworthiness...

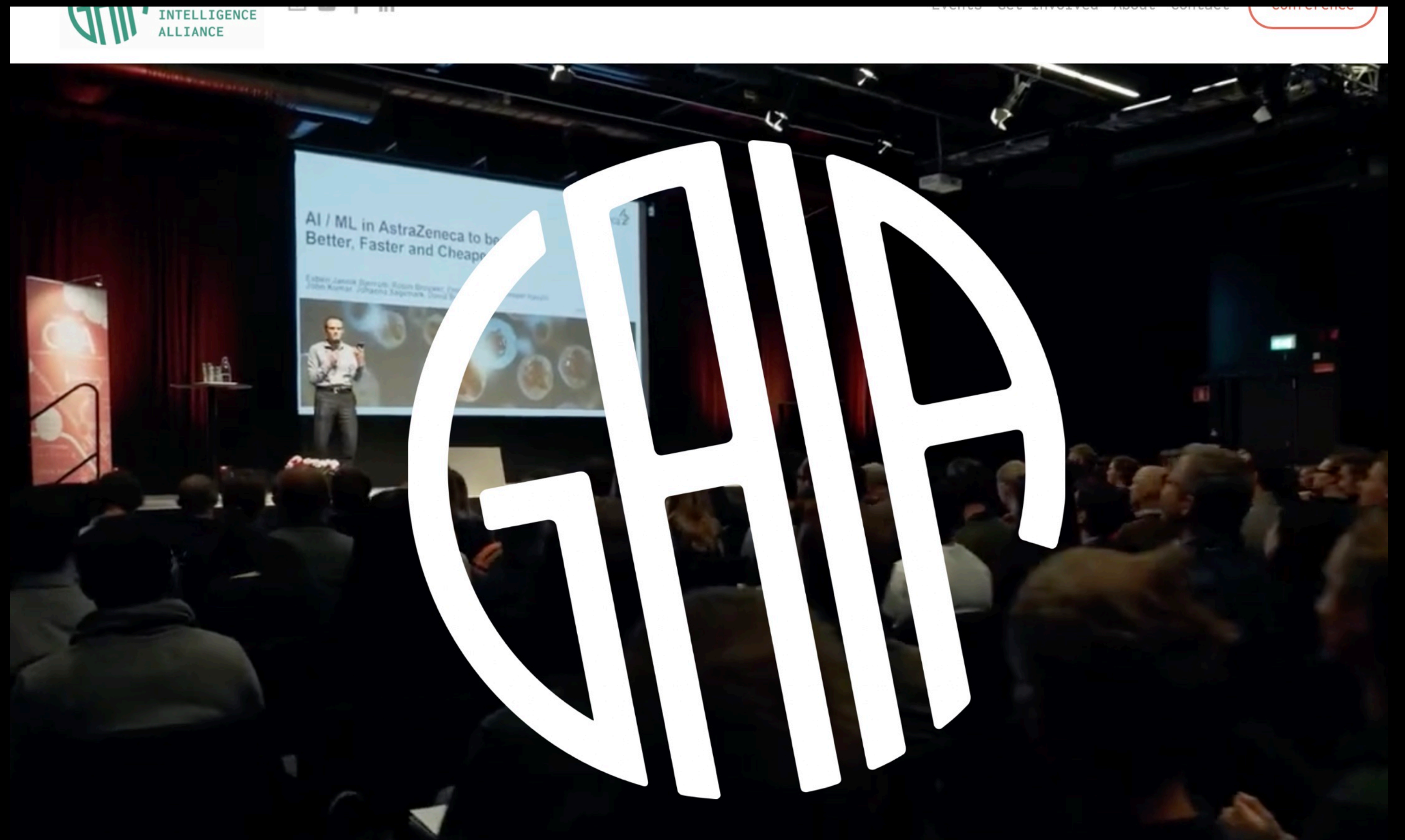
| SOME USEFUL LINKS

- Models:
 - Huggingface: <https://huggingface.co/>
 - Mistral: <https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.1-GGUF>
 - BGE: <https://huggingface.co/BAAI/bge-large-en>
- Code
 - Langchain: <https://www.langchain.com/>
 - BS4: <https://beautiful-soup-4.readthedocs.io/en/latest/>
 - LlamaCPP <https://github.com/ggerganov/llama.cpp>
 - Chroma DB: <https://docs.trychroma.com/>
- Papers:
 - Attention is all you need: <https://arxiv.org/abs/1706.0376>
 - Llama2: <https://arxiv.org/abs/2307.09288>

MORE ON AI - GOTHENBURG AI ALLIANCE

GAIA 2024

27/3



<https://gaia.fish>