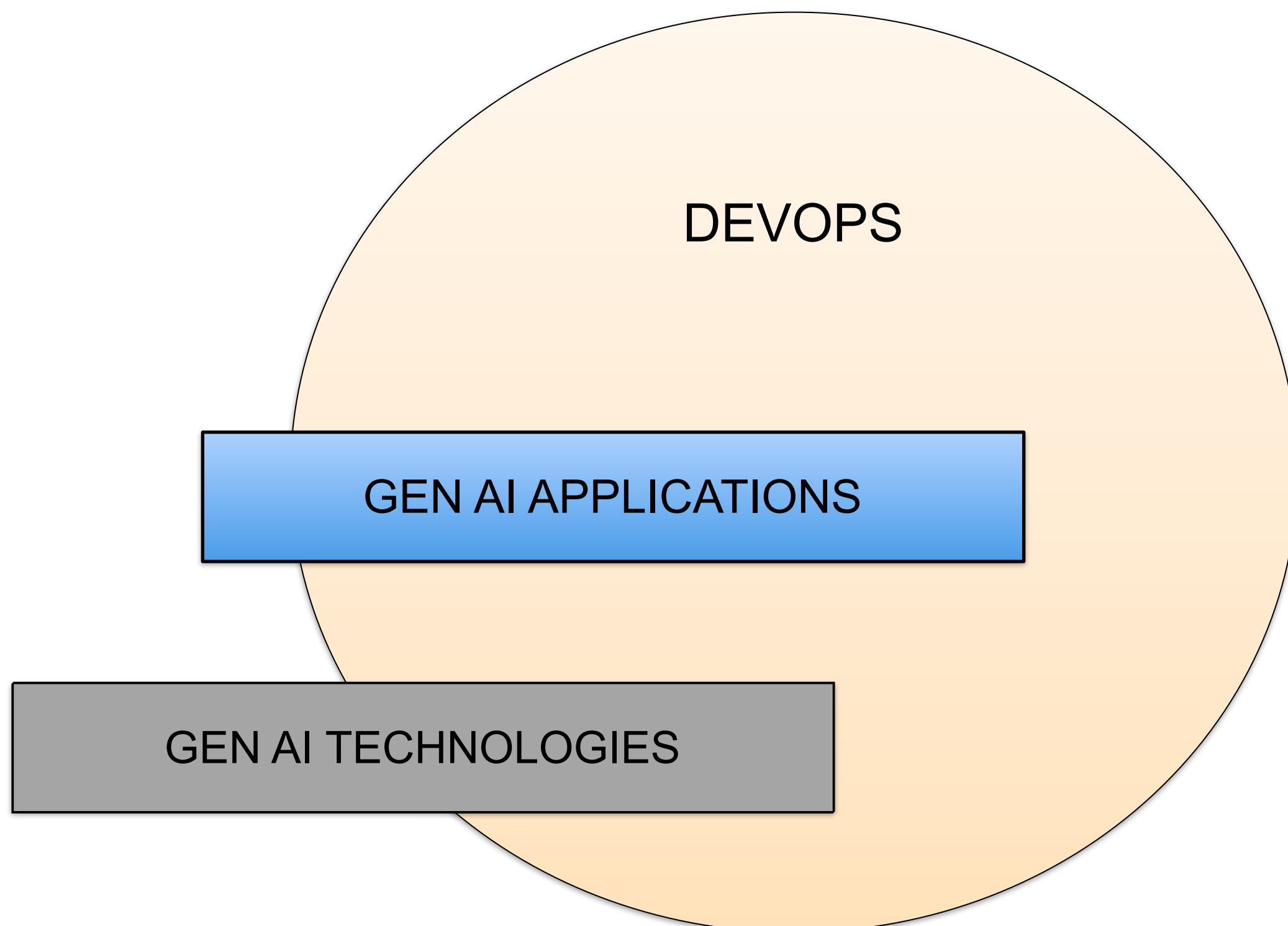




# GENERATIVE AI: APPLICATIONS IN THE SERVERLESS WORLD

DIANA TODEA - SITE RELIABILITY ENGINEER

SERVERLESS ARCHITECTURE & DEVOPS CON LONDON 2024



DEVOPS

GEN AI APPLICATIONS

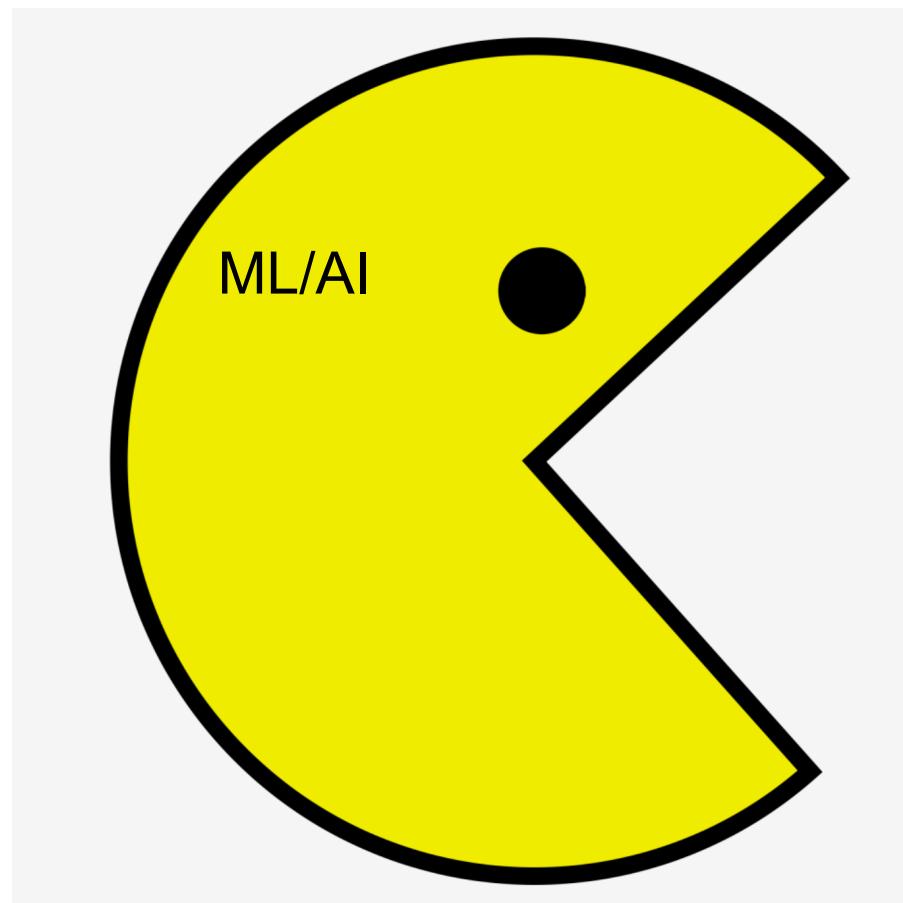
GEN AI TECHNOLOGIES

# TOP 5 TRENDS FOR SEARCH AND AI IN 2024

1. ANSWERS NOT LINKS
2. DATA SAFETY AND PRIVACY
3. PROVIDE PROPER CONTEXT FOR GENERATIVE AI
4. THE NEXT WAVE OF LARGE LANGUAGE MODELS
5. AI SKILLS AND EXPERTISE FOR DEVELOPERS

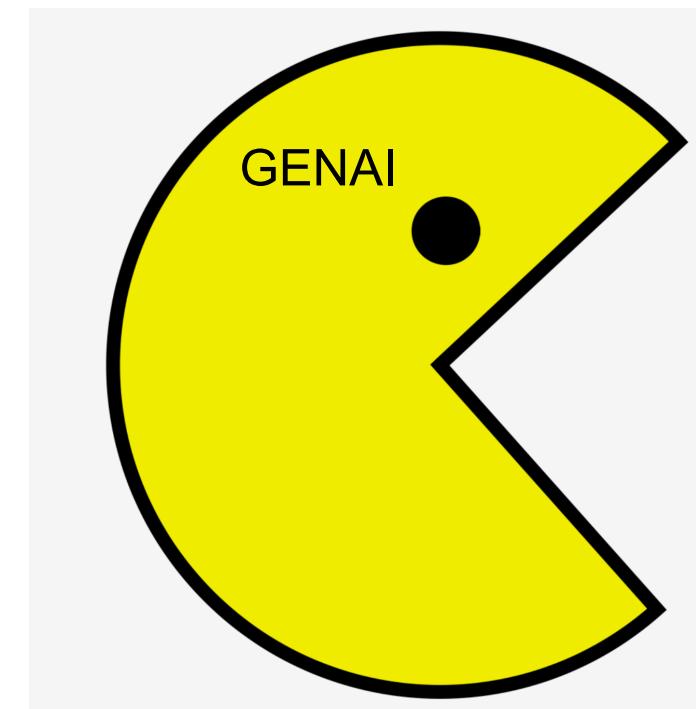
## **ALGORITHMS PROGRAMMED TO MAKE PREDICTIONS ON DATA**

IMAGE RECOGNITION, NLP,  
SPEECH RECOGNITION



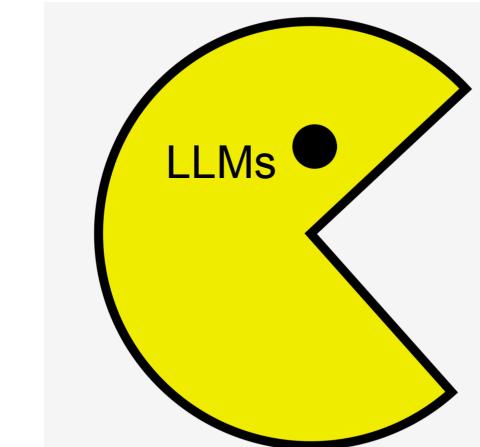
**AI ALGORITHMS DESIGNED  
TO CREATE HIGH QUALITY  
CONTENT, SPECIFICALLY: TEXT,  
IMAGE, AUDIO**

CHATBOTS, TEXT, IMAGE,  
MUSIC GENERATORS



**DEEP LEARNING ALGORITHMS  
THAT CAN GENERATE TEXT**

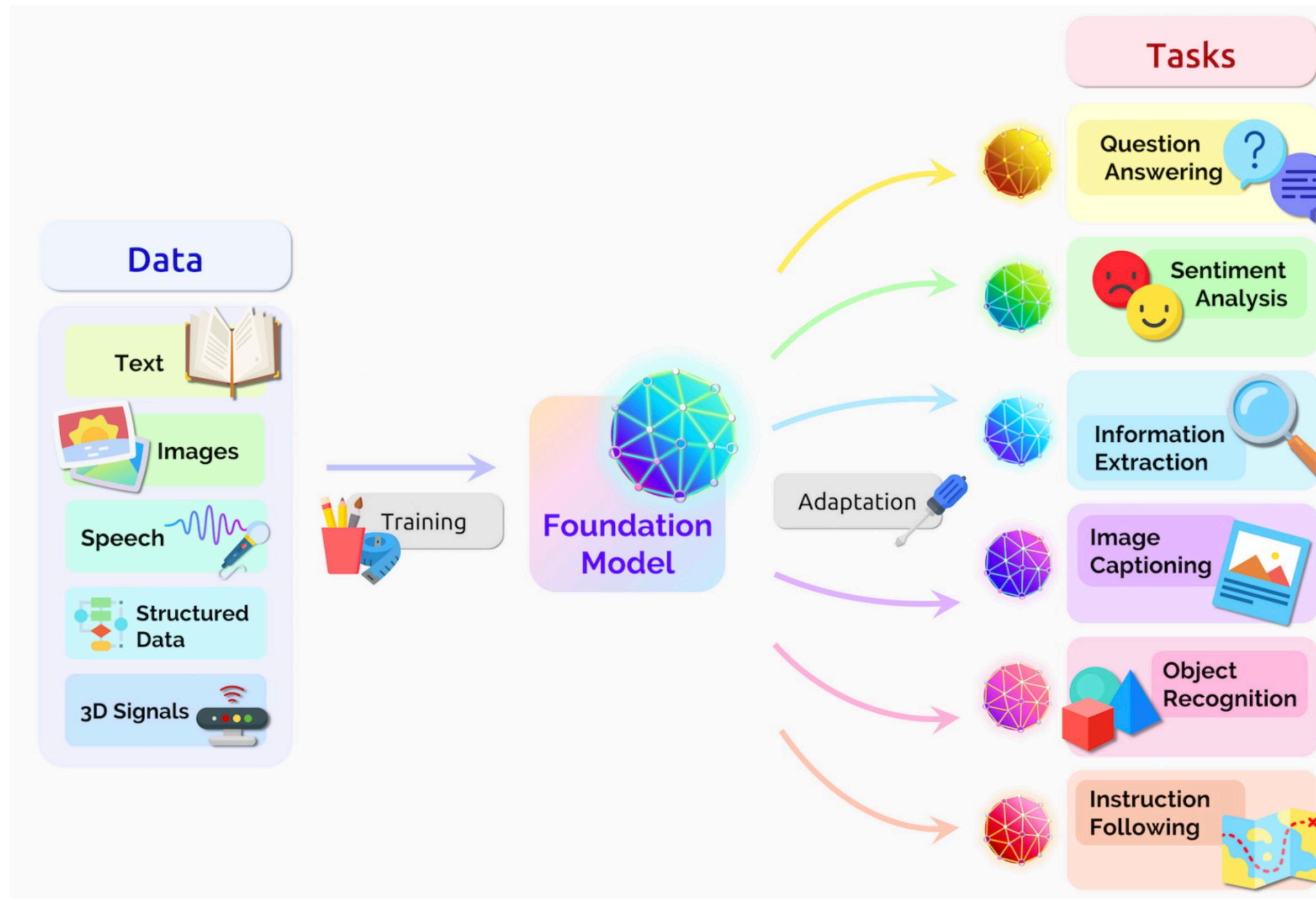
CHATBOTS, TEXT GENERATORS,  
TRANSLATION, WRITING, ANSWERING  
QUESTIONS



# TRANSFORMER DEFINITION

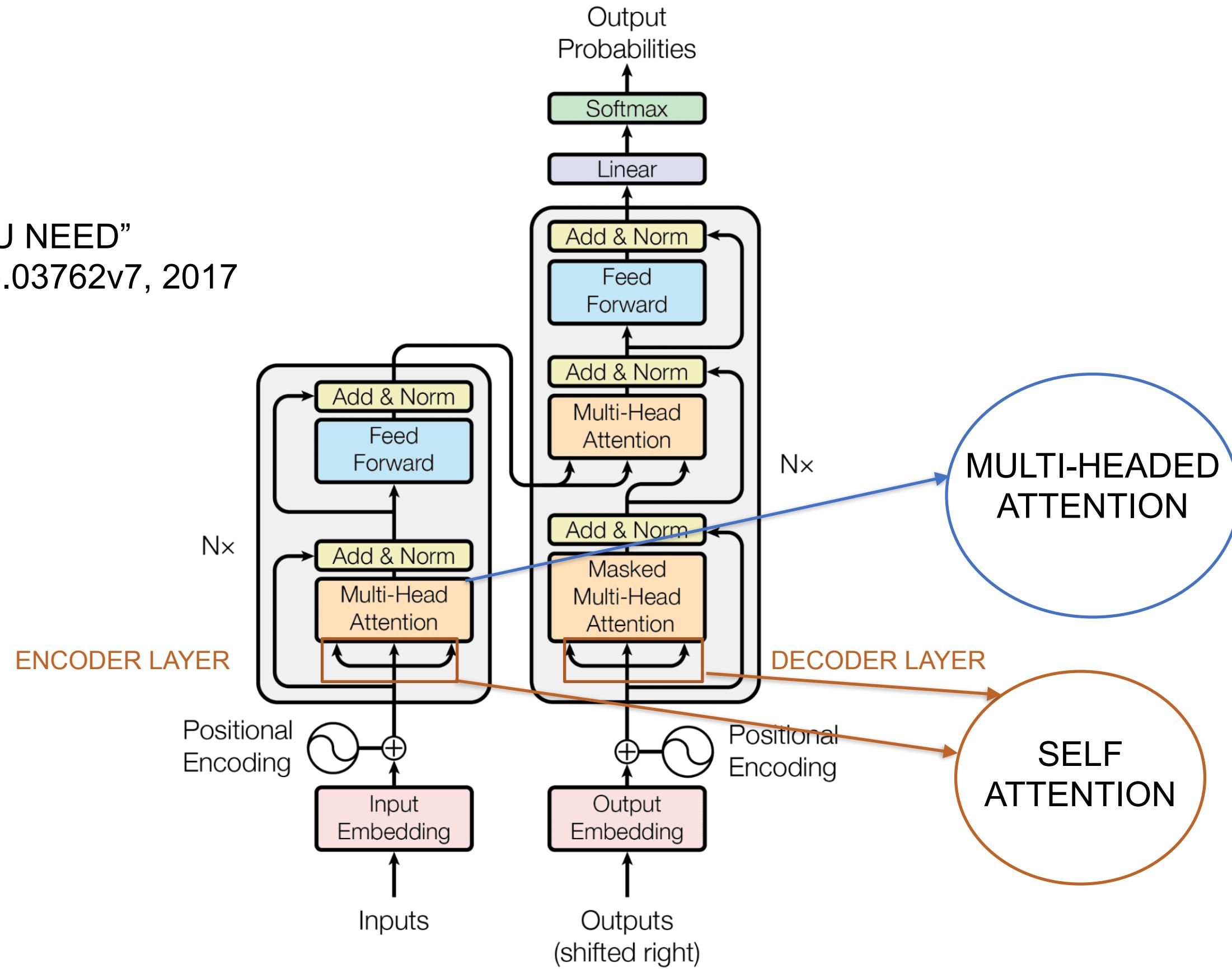
IN MACHINE LEARNING, A TRANSFORMER IS A NEURAL NETWORK THAT LEARNS CONTEXT AND MEANING BY TRACKING RELATIONSHIPS IN SEQUENTIAL DATA LIKE THE WORDS IN THE SENTENCE.

# WHAT CAN TRANSFORMERS DO?

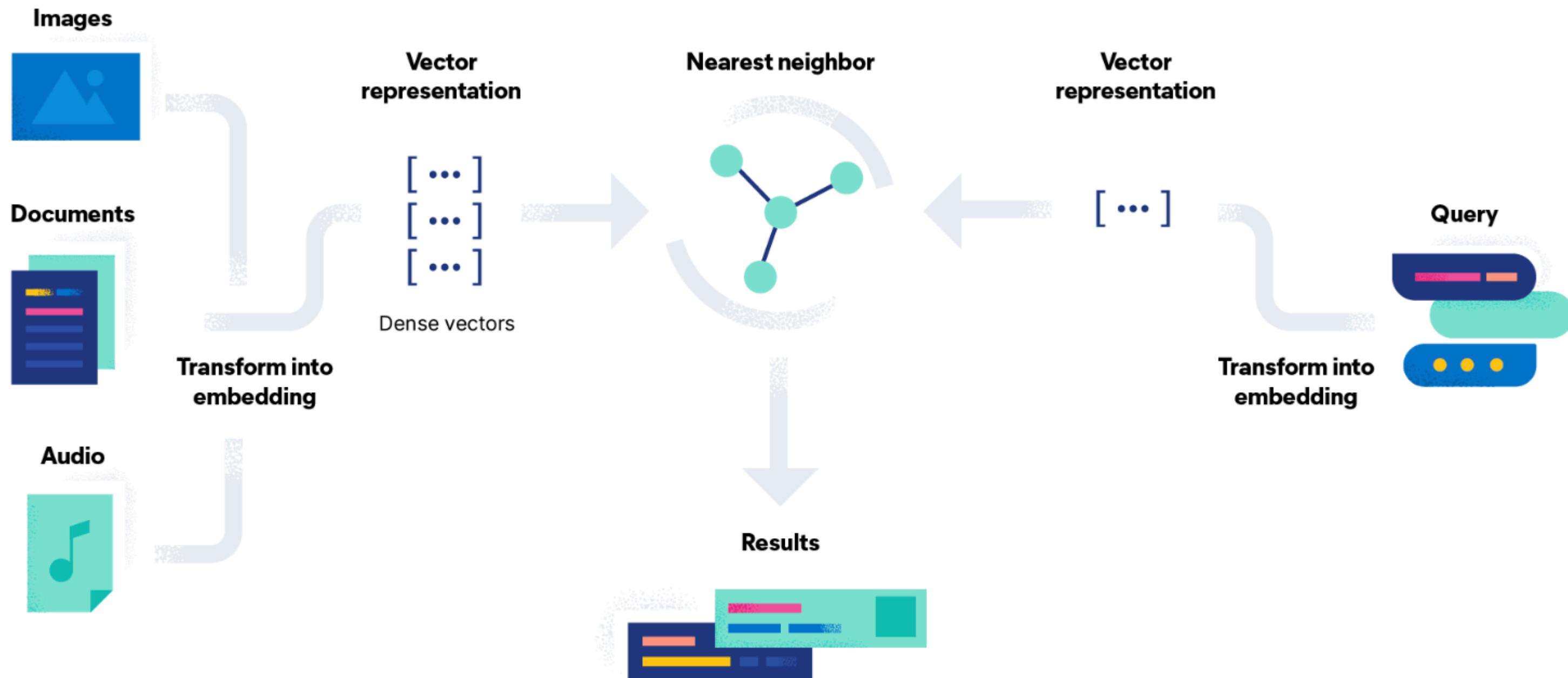


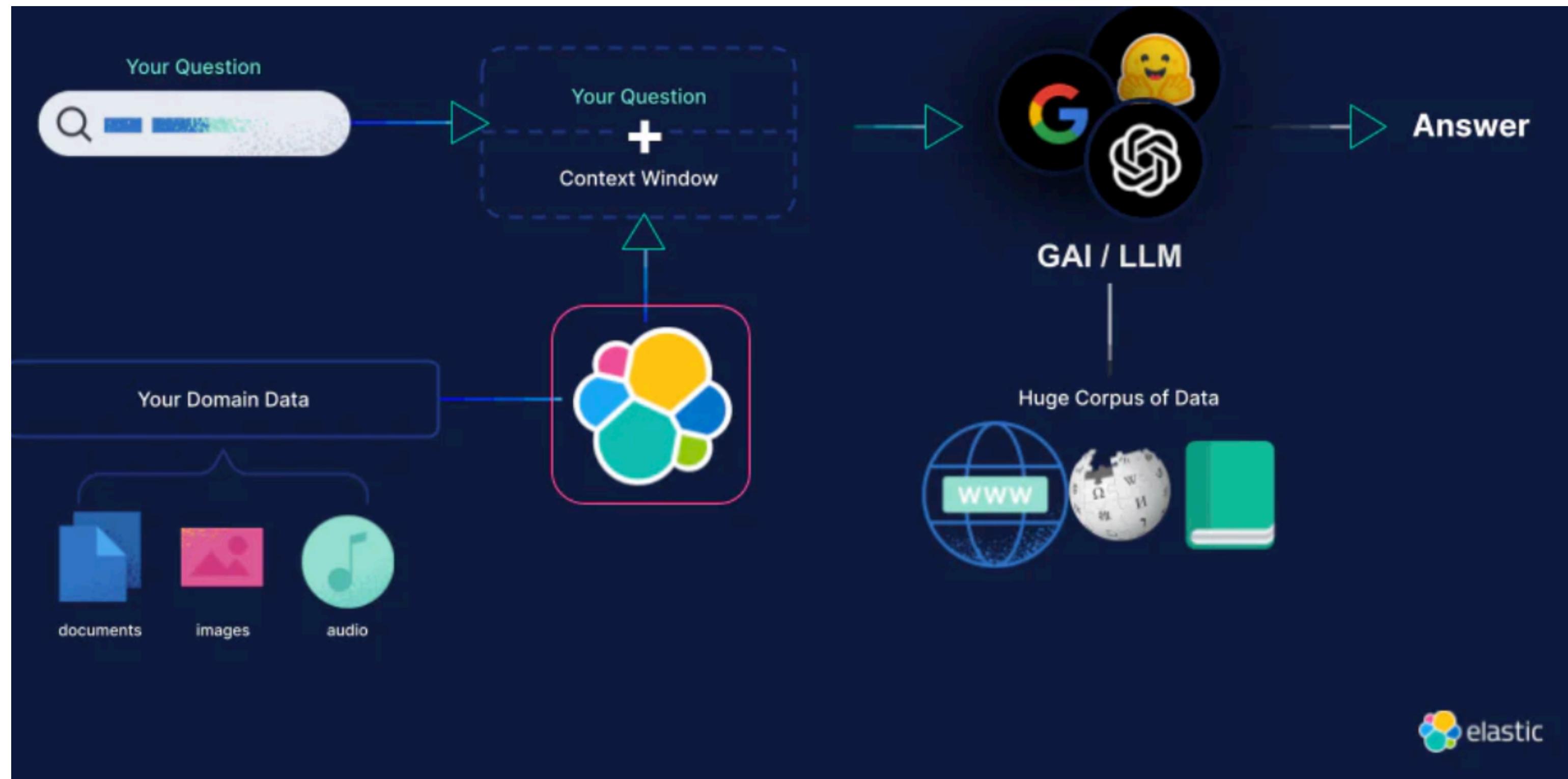
# TRANSFORMER ARCHITECTURE

“ATTENTION IS ALL YOU NEED”  
<https://arxiv.org/abs/1706.03762v7>, 2017

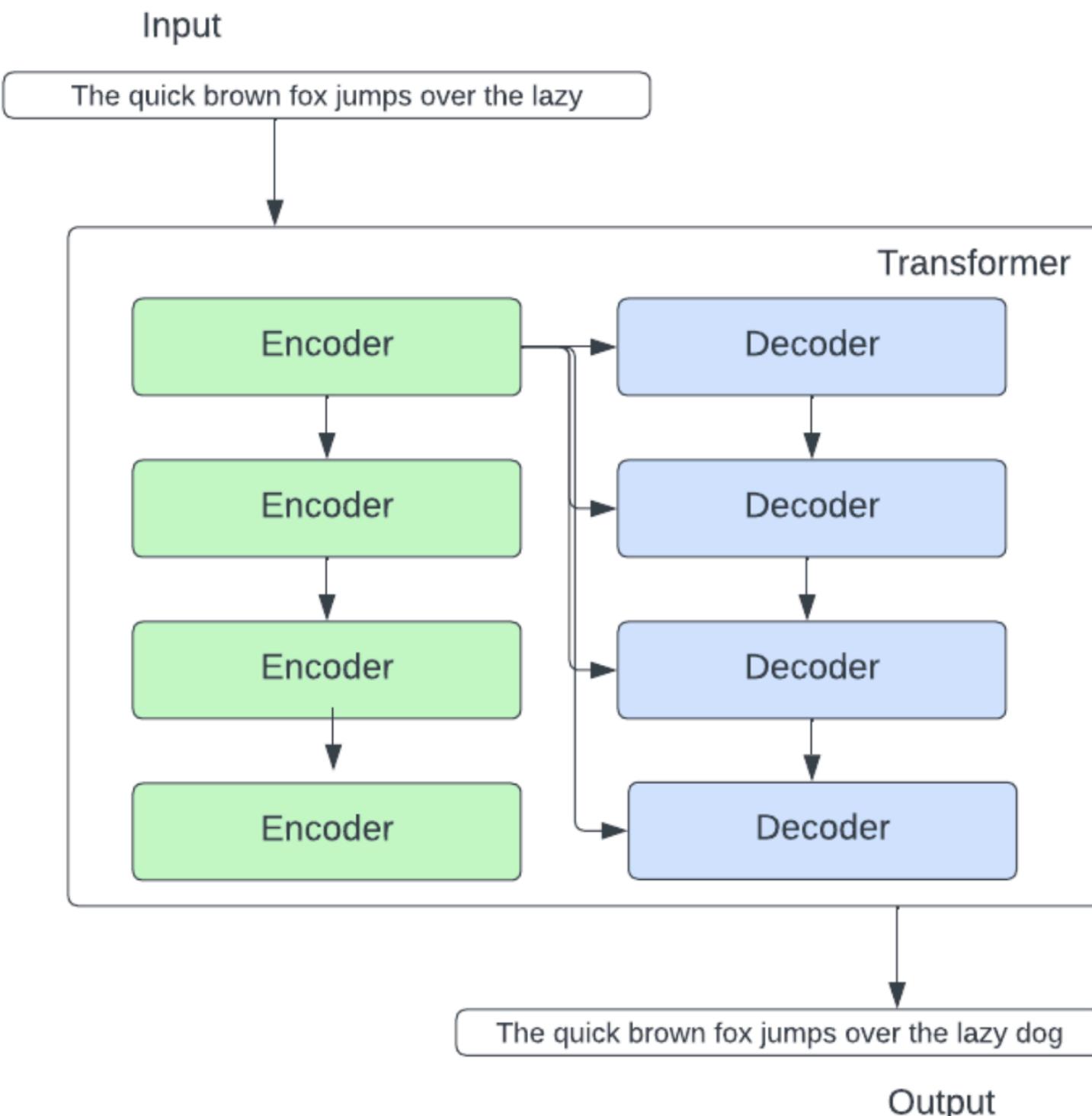


# GEN AI ARCHITECTURE



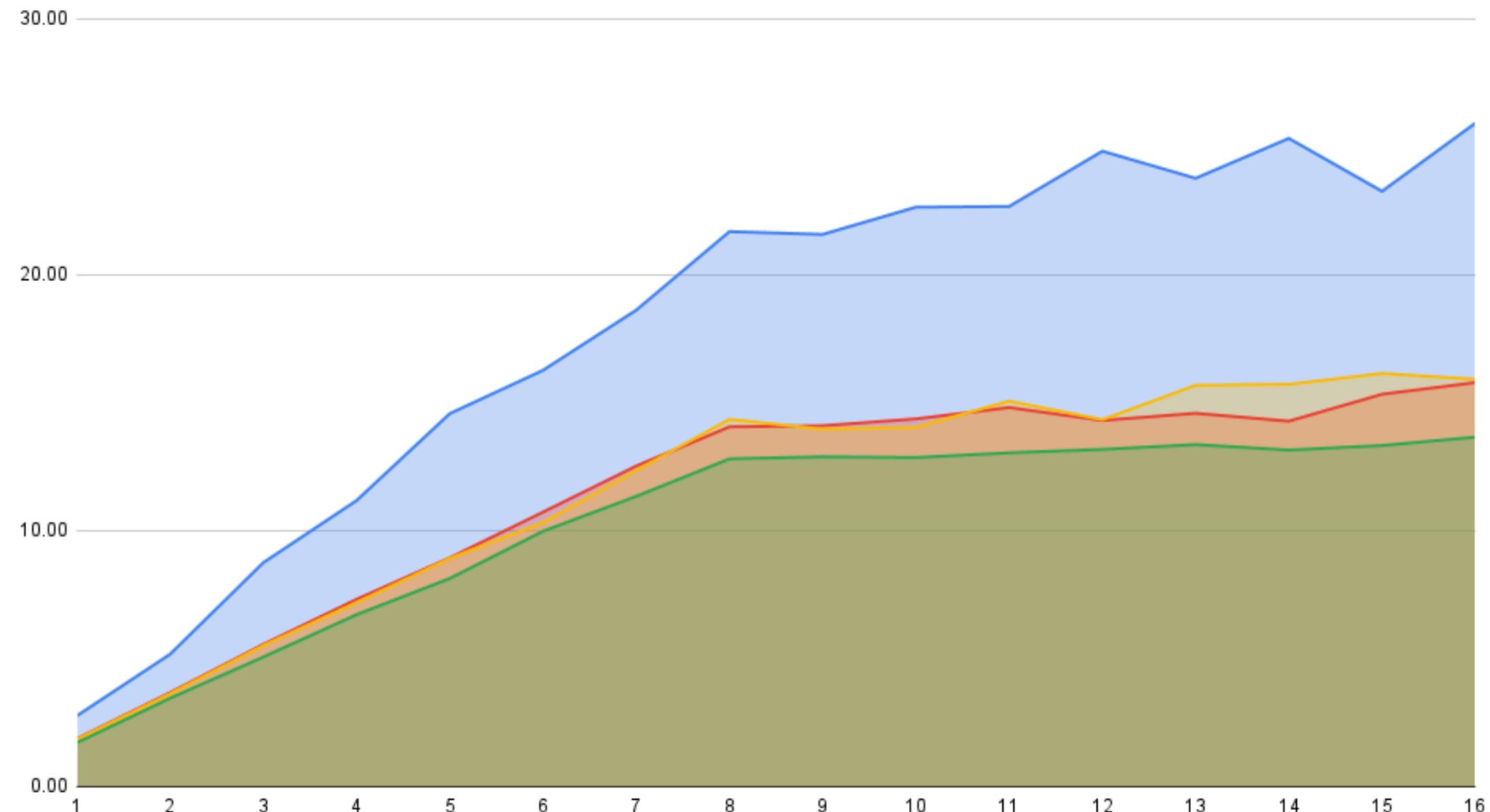


# RETRIEVAL AUGMENTED GENERATION (RAG)



Retrieval augmented generation (RAG) is a technique that supplements text generation with information from private or proprietary data sources.

- 1.RAG starts with an input query.
- 2.The retrieval model grabs the relevant information from databases or external sources.
- 3.The retrieved information is converted into vectors in a high-dimensional space, which are then stored into a vector database.
- 4.The retrieval model ranks the retrieved information based on its relevance to the input query. The documents with the highest scores get selected for further processing.



# RAG + ELSER

## Add a trained model

[Click to Download](#)    [Manual Download](#)

### ELSER (Elastic Learned Sparse EncodeR)

ELSER is Elastic's NLP model for English semantic search, utilizing sparse vectors. It prioritizes contextual meaning over literal term matching, optimized specifically for English documents on the Elastic platform.

[View documentation](#)

#### Choose a model



**Cross platform**  
.elser\_model\_2



**Intel and Linux optimized**  
.elser\_model\_2\_linux-x86\_64

Recommended



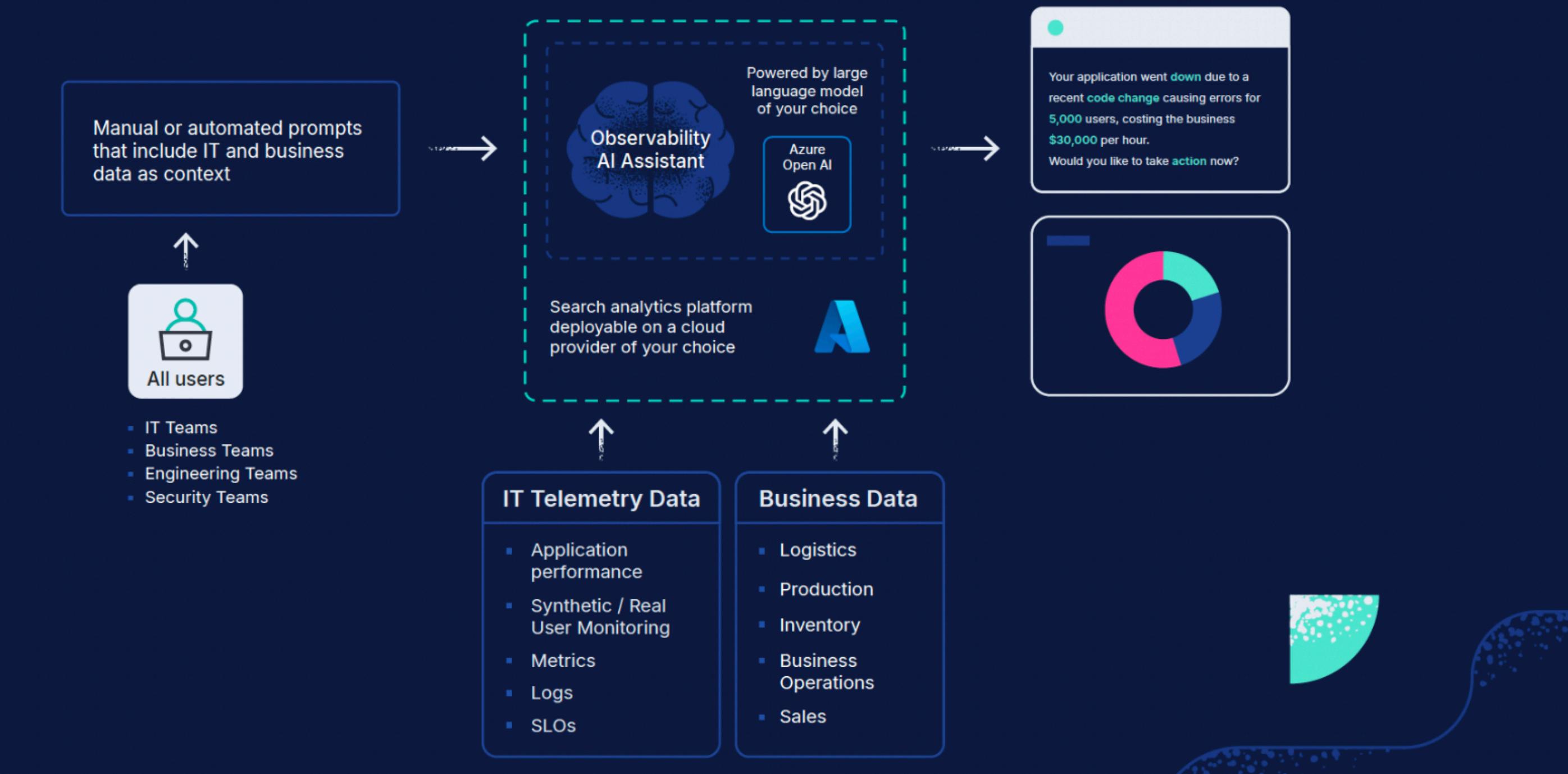
Sherlock

Research

Explain



# Blueprint for Business Observability



elastic

Find apps, content, and more.

Add integrations AI Assistant

## SLOs

Search your SLOs... Sort by slo status ▾

cartservice-otel APM availability cartservice-... 7 days Target 99.9% 99.519%

Checkout SLO APM availability checkoutSe... 7 days Target 95% 0%

cartservice-otel-latency APM latency cartservice-... 7 days Target 99% 99.498%

Rows per page: 25 < 1 >

Tell us what you think! Stop refreshing Create new SLO

Compact view

Observability Overview Alerts SLOs Cases

Logs Explorer BETA Stream Anomalies Categories

Infrastructure Inventory Metrics Explorer Hosts BETA

APM Services Traces Dependencies

Synthetics Monitors TLS Certificates

Uptime Uptime Monitors TLS Certificates

User Experience Dashboard

Universal Profiling

The screenshot shows the Elastic Observability SLOs dashboard. On the left, there's a sidebar with various monitoring categories like Logs, Infrastructure, APM, and User Experience. The main area displays three SLO cards. The first card, 'cartservice-otel', has a red background and shows a 99.519% success rate against a 99.9% target. It includes APM availability and cartservice-related metrics. The second card, 'Checkout SLO', also has a red background but shows 0% success against a 95% target, with APM availability and checkout-related metrics. The third card, 'cartservice-otel-latency', has a green background and shows a 99.498% success rate against a 99% target, with APM latency and cartservice-related metrics. Each card includes a 'Target' percentage, the current success percentage in large bold digits, and a status icon indicating performance relative to the target.



# OBSERVABILITY AI ASSISTANT

[Open conversation](#)

This feature is currently in **Tech Preview** and may contain issues. [Give feedback](#)

## Checking Active Alerts in Elastic Observability

You  
Do I have any active alerts in the last 20 minutes?

Elastic Assistant requested the function recall

You executed the function recall

Elastic Assistant requested the function alerts

You executed the function alerts

Elastic Assistant  
You have 5 active alerts in the last 20 minutes:

- Error count threshold | checkoutService:** Error count is 16 in the last 30 mins for service: checkoutService, env: prod. Alert when > 10.
- log.offset:** 6180 log entries in the last 15 mins. Alert when > 75.
- SLO latency checkout service burn rate rule:** MEDIUM: The burn rate for the past 24h is 3.71 and for the past 120m is 4.53. Alert when above 3 for both windows.
- latency threshold in frontend-rum service (page-load):** Avg. latency is 2.4 s in the last 5 mins for service: frontend-rum, env: prod, type: page-load. Alert when > 1.5 s.
- latency threshold in frontend-rum service (user-interaction):** Avg. latency is 1.7 s in the last 5 mins for service: frontend-rum, env: prod, type: user-interaction. Alert when > 1.5 s.

[Call function](#)

Send a message to the Assistant

Alerts [Go to conversations](#)

This feature is currently in **Tech Preview** and may contain issues. [Give feedback](#)

## New conversation

Elastic Assistant  
You have 5 active alerts in the last 20 minutes:

- Error count threshold | checkoutService:** Error count is 15 in the last 30 mins for service: checkoutService, env: prod. Alert when > 10.
- log.offset:** 4794 log entries in the last 15 mins. Alert when > 75.
- SLO latency checkout service burn rate rule:** MEDIUM: The burn rate for the past 24h is 3.68 and for the past 120m is 4.23. Alert when above 3 for both windows.
- latency threshold in frontend-rum service (page-load):** Avg. latency is 2.4 s in the last 5 mins for service: frontend-rum, env: prod, type: page-load. Alert when > 1.5 s.
- latency threshold in frontend-rum service (user-interaction):** Avg. latency is 1.7 s in the last 5 mins for service: frontend-rum, env: prod, type: user-interaction. Alert when > 1.5 s.

Stop generating

[Call function](#)

Send a message to the Assistant



# Discussing ArgoCD Workqueue Depth



## Elastic Assistant



The alert '[ML] Stateful: FATAL Log Messages' indicates that there are fatal log messages in your Elasticsearch cluster. Here are some general troubleshooting steps you can take:

- 1. Check the Logs:** The first step in troubleshooting this issue would be to check the logs for any fatal errors. You can do this by accessing the Kibana Discover app and filtering for fatal log messages. The alert indicates that these messages are coming from the 'logging-:cluster-elasticsearch-' index.
- 2. Identify the Source:** Try to identify the source of these fatal messages. They could be coming from a specific service, node, or part of your application. This can help you narrow down where the issue might be.
- 3. Check for Recent Changes:** Have there been any recent changes to your Elasticsearch cluster or the application that could have caused these errors? This could include things like configuration changes, updates, or new deployments.
- 4. Investigate the Error Messages:** The error messages themselves can often provide clues as to what's going wrong. They might indicate a problem with a specific part of your application, a resource issue, a network problem, etc.
- 5. Monitor Resource Usage:** Check the resource usage of your Elasticsearch cluster. Fatal errors can sometimes be caused by resource issues, such as running out of memory or disk space.
- 6. Contact Support:** If you're unable to resolve the issue yourself, you may need to contact Elastic support for further assistance.

**Elasticsearch engine availability**

Violated serverless.project.id.slo: acf162abba1c41558746dbae00373691

Last updated on Jan 24, 2024 Created on Jan 3, 2024

**Overview** Alerts 0

Observed value 99.02% (objective is 99.95%) Indicator type Custom KQL

Description Test Tags test, serverless

Burn rate TECHNICAL PREVIEW Acceptable value 0.80%

Open conversation

### Analyzing Current State of Elasticsearch

You started a conversation

You please provide an analysis of the current state of Elasticsearch.

Show 4 events

Elastic Assistant

The current state of the Elasticsearch engine availability over the last hour is 92.59 and for the past 5 minutes is 92.8. This is below the acceptable value of 0.80.

Was this helpful? Yes No

### Elastic Assistant

Creating a p99 SLO (Service Level Objective) involves defining the "good" events that contribute towards meeting the SLO, and the total events that are relevant for the SLO.

Here's a general process on how to create a p99 SLO:

- Identify the service and the SLI (Service Level Indicator):** The service could be any part of your system that you want to measure the reliability of. The SLI is a specific metric that you want to measure. For a p99 SLO, this could be the latency of a service.
- Define the SLO:** The SLO is the target percentage of the time that you want your service to meet the SLI. For a p99 SLO, this would be 99%.
- Collect data:** You need to collect data on the SLI and the total events. This could involve instrumenting your service to collect this data, or using existing data if it's already available.
- Calculate the SLO:** The SLO is calculated as the percentage of good events out of the total events. For a p99 SLO, you would calculate the percentage of events where the latency is below a certain threshold.
- Monitor and alert on the SLO:** Once you've defined and calculated your SLO, you need to set up monitoring and alerting. This will let you know when you're at risk of not meeting your SLO, so you can take action to improve reliability.

Creating a Service Level Objective (SLO) for a serverless architecture involves similar steps as with traditional architectures, but with a focus on the specific characteristics of serverless. Here's an example of how you might create an SLO for a serverless function:

- Identify the service and the SLI (Service Level Indicator):** In a serverless architecture, the service could be a specific serverless function. The SLI could be the latency of the function, the error rate, or another metric that is important for your application.
- Define the SLO:** The SLO is the target percentage of the time that you want your service to meet the SLI. For example, you might set an SLO that your serverless function should have a latency of less than 200ms 99% of the time.
- Collect data:** Serverless platforms often provide monitoring and logging capabilities that you can use to collect data on your SLIs. For example, AWS Lambda provides metrics on function invocations, errors, and duration.
- Calculate the SLO:** The SLO is calculated as the percentage of good events out of the total events. For the example SLO above, you would calculate the percentage of function invocations where the latency is less than 200ms.
- Monitor and alert on the SLO:** Once you've defined and calculated your SLO, you need to set up monitoring and alerting. This will let you know when you're at risk of not meeting your SLO, so you can take action to improve reliability.

Remember, the goal of an SLO is to help you understand the reliability of your service and make informed decisions about where to invest in improving reliability. The exact SLIs and SLOs you choose will depend on what is most important for your application and users.

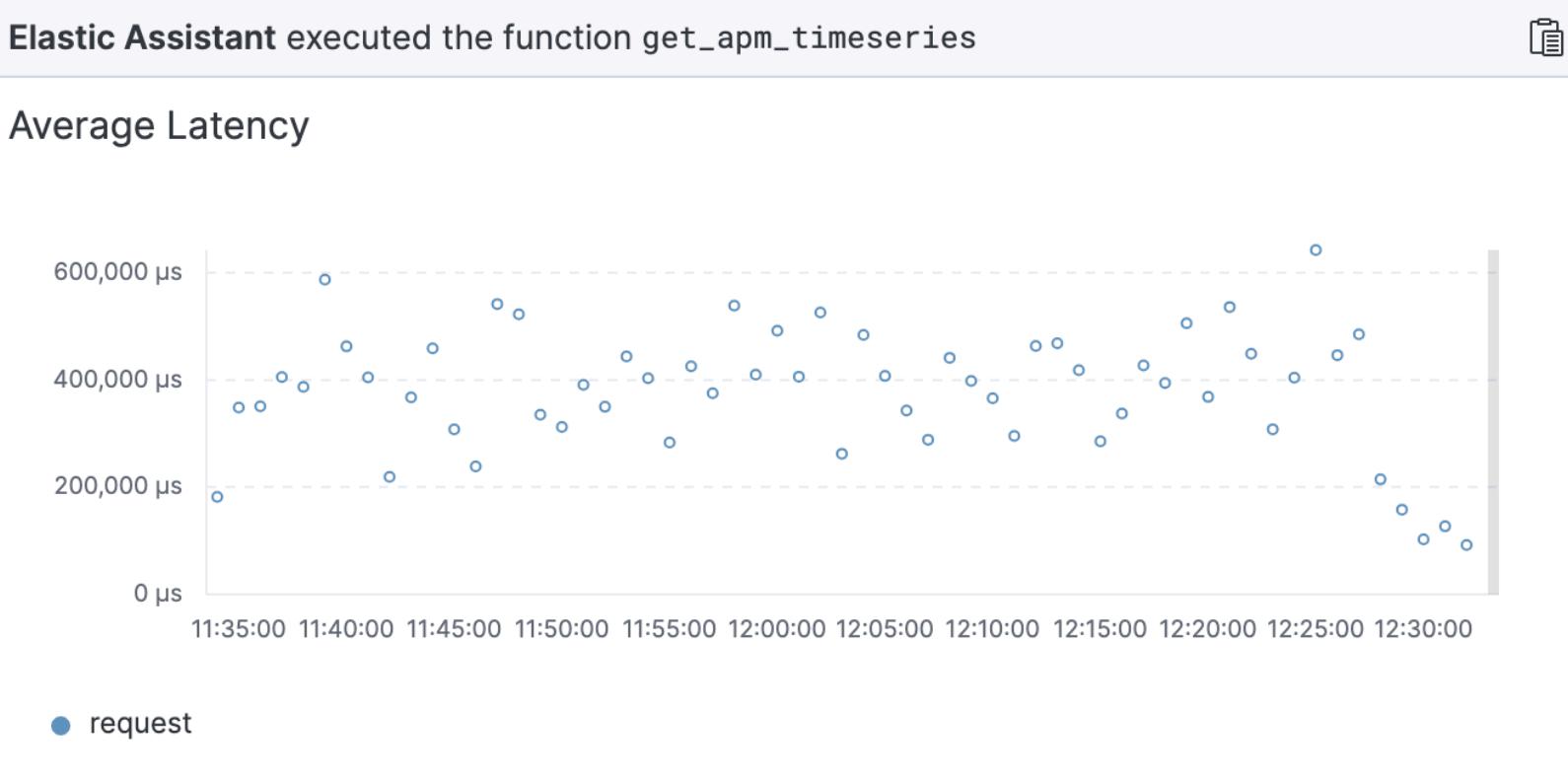
# serverless-proxy

Overview Transactions Dependencies Errors Metrics

request Search transactions (E.g. transaction.c)

Comparison Day before

Latency Metric Average



Elastic Assistant

The average latency for the serverless-proxy application over the last hour is 380 milliseconds. There have been no significant changes in latency during this period, indicating that the application's performance has been stable.

Open conversation

## Serverless-Proxy Application Performance Inquiry



### Elastic Assistant

Your serverless-proxy application is currently running in an environment that is not defined. It is written in Go and is running on 6 instances. There are no anomalies detected for this service at the moment.

However, there are several alerts associated with this service. Here are some of the most recent ones:

- Alert started at 2023-09-25T08:16:42.976Z
- Alert started at 2023-09-17T05:21:15.919Z
- Alert started at 2023-09-12T08:22:16.561Z
- Alert started at 2023-09-12T08:07:57.029Z
- Alert started at 2023-09-12T07:53:42.800Z

Please note that these alerts might indicate potential issues with the service. You might want to investigate these alerts further to understand their cause and impact.

There are no deployments associated with this service at the moment.

Observability

# Monitors

Overview Management

Search by name, URL, host, tag, project or location

Up Down Disabled

Current status: 1845 Up, 10 Down, 0 Disabled

Last 6 hours: 10 Errors

Showing 1,855 Monitors

APM Agent Versions: North America - US East Duration 32 ms

APM Agent Versions: Europe - United Kingdom

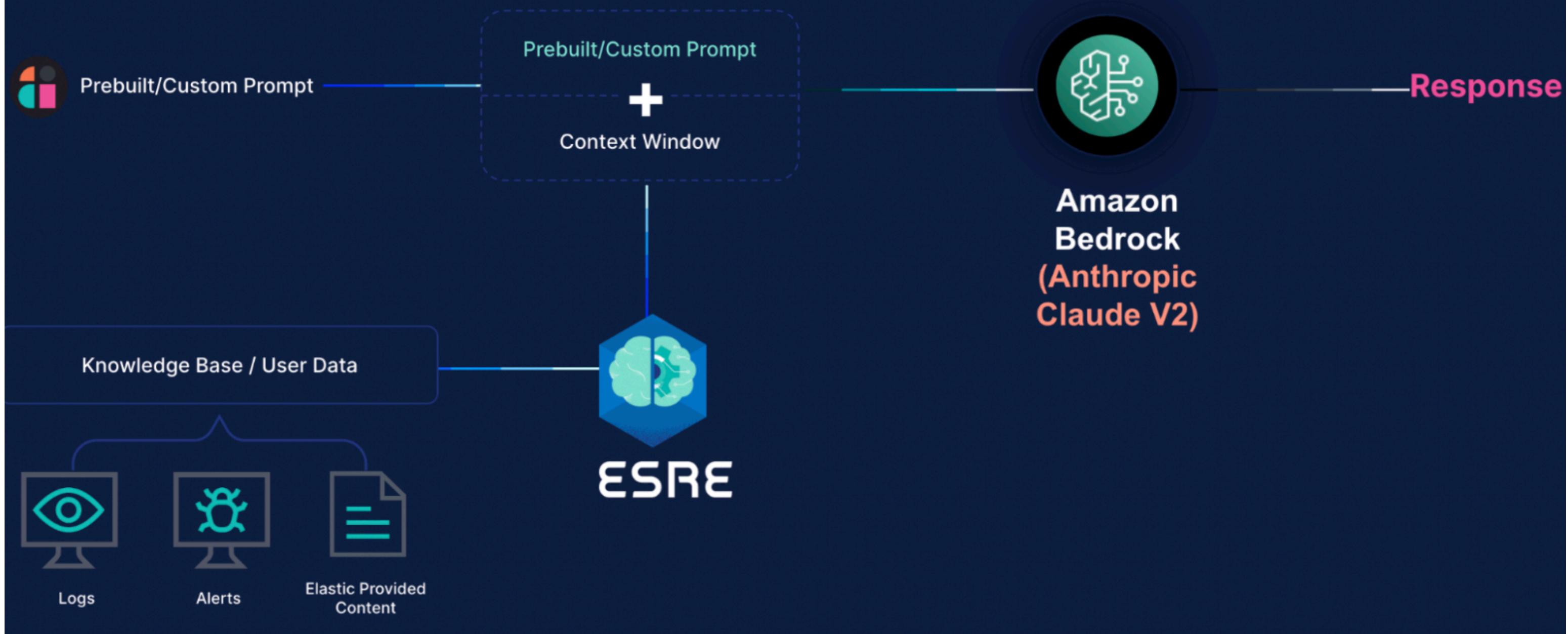
[Open conversation](#)

## Exploring Errors in Current Active Monitors

- Could not connect to 'metadata.google.internal:' with error: Get "<http://metadata.google.internal/computeMetadata/v1/?recursive=true&alt=text>": Could not connect to '169.254.169.254:80' with error: dial tcp 169.254.169.254:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
6. Monitor "Test Metadata Access" from Europe - United Kingdom is down. Error message: Could not connect to 'metadata.google.internal:' with error: Get "<http://metadata.google.internal/computeMetadata/v1/?recursive=true&alt=text>": Could not connect to '169.254.169.254:80' with error: dial tcp 169.254.169.254:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
7. Monitor "Test Metadata Access" from Europe - Germany is down. Error message: Could not connect to 'metadata.google.internal:' with error: Get "<http://metadata.google.internal/computeMetadata/v1/?recursive=true&alt=text>": Could not connect to '169.254.169.254:80' with error: dial tcp 169.254.169.254:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
8. Monitor "Test Metadata Access" from South America - Brazil is down. Error message: Could not connect to '169.254.169.254:80' with error: dial tcp 169.254.169.254:80: i/o timeout
9. Monitor "Test Metadata Access" from North America - US East is down. Error message: Could not connect to 'metadata.google.internal:' with error: Get "<http://metadata.google.internal/computeMetadata/v1/?recursive=true&alt=text>": Could not connect to '169.254.169.254:80' with error: dial tcp 169.254.169.254

Please note that all these errors are due to connection issues with 'metadata.google.internal'.

# Elastic AI Assistant - Behind the Scenes



<https://github.com/elasticsearch-labs>

 **elasticsearch-labs** Public

Watch 190 Fork 93 Star 340

main Branches Tags Go to file Add file Code

miguelgrinberg Upgrade to notebook test matrix to include 8.13 and 8.14... ab9a235 · 5 days ago 389 Commits

.github/workflows Upgrade to notebook test matrix to include 8.13 and 8.14 ... 5 days ago

bin Image similarity (#206) 2 weeks ago

datasets add a book dataset with chunked synopsis 5 months ago

example-apps Updated langchain-core dependency to resolve conflict. (...) 3 weeks ago

notebooks Update esql-getting-started.ipynb (#215) 5 days ago

supporting-blog-content Adding blog post notebook (#195) 3 weeks ago

.gitignore simplify DLS role descriptor (#180) 3 months ago

.pre-commit-config.yaml Add black formatter as pre-commit step (#189) 2 months ago

CONTRIBUTING.md Add black formatter as pre-commit step (#189) 2 months ago

LICENSE Initial commit 10 months ago

Makefile Add black formatter as pre-commit step (#189) 2 months ago

NOTICE Use standalone nbtest (#171) 3 months ago

README.md Update README.md last week

requirements-dev.txt Add a secrets scanner to CI and pre-commit (#168) 3 months ago

README Apache-2.0 license Security

## Elasticsearch Examples & Apps

Visit [Search Labs](#) for the latest articles and tutorials on using Elasticsearch for search and AI/ML-powered search experiences

About

Notebooks & Example Apps for Search & AI Applications with Elasticsearch

[www.elastic.co/search-labs](http://www.elastic.co/search-labs)

python search elasticsearch ai  
vector applications openai elastic  
chatlog chatgpt langchain  
openai-chatgpt langchain-python  
genai genaistack vectordatabase

Readme Apache-2.0 license Security policy

Activity Custom properties

340 stars 190 watching 93 forks

Report repository

Contributors 34

+ 20 contributors

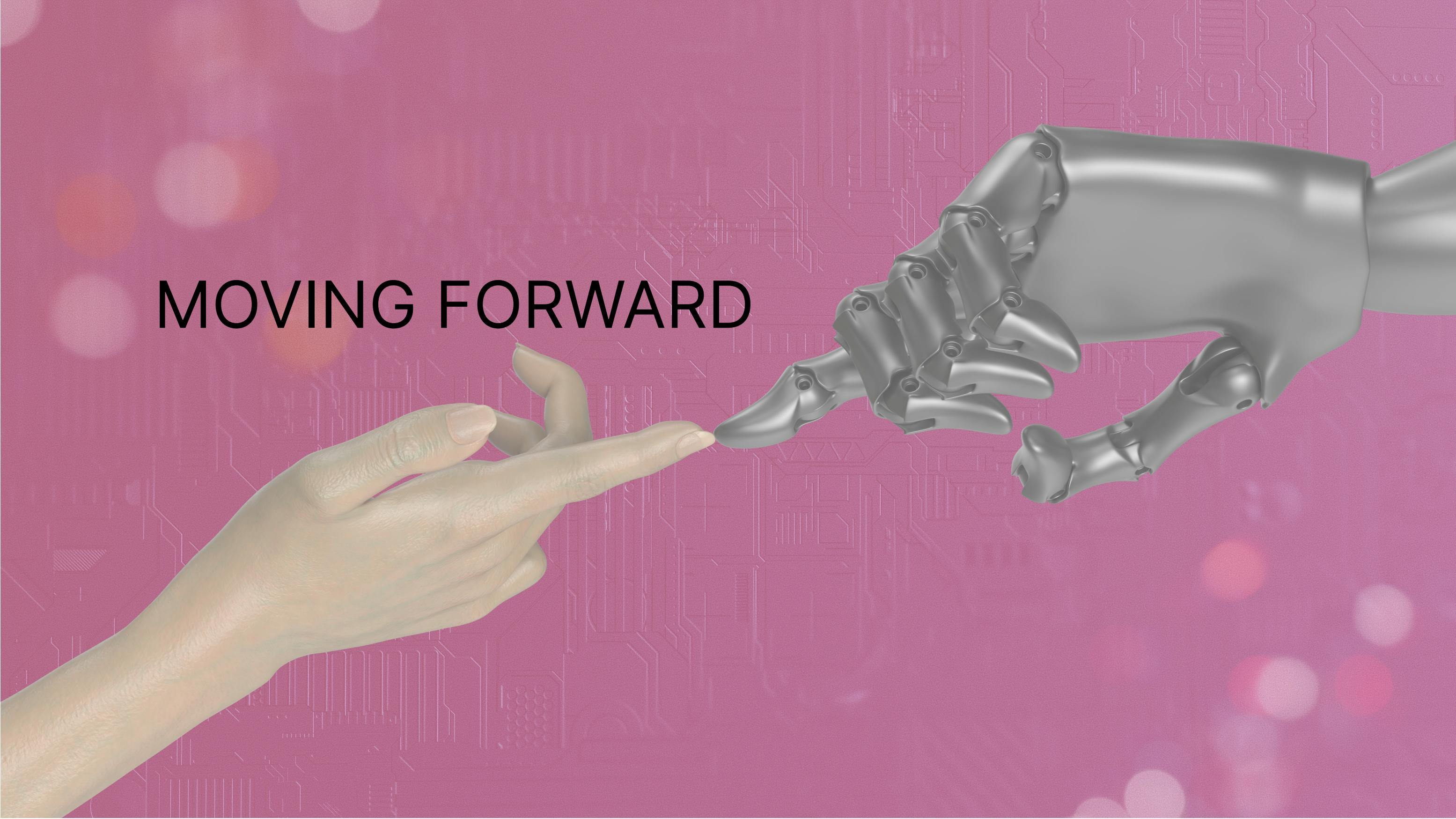
Languages

- Jupyter Notebook 91.7%
- Python 3.6% TypeScript 2.2%
- JavaScript 1.3% HTML 0.7%
- CSS 0.2% Other 0.3%



# AI ASSISTANTS ON SERVERLESS

1. FUNCTION AS A SERVICE (FaaS)
2. SCALABILITY
3. COST-EFFECTIVENESS
4. INTEGRATION WITH OTHER SERVICES
5. EVENT-DRIVEN ARCHITECTURE



# MOVING FORWARD

Photo by [Igor Omilaev](#) on [Unsplash](#)



Where do you expect that the MOST time and resources will be spent when building Generative AI use cases for your applications?

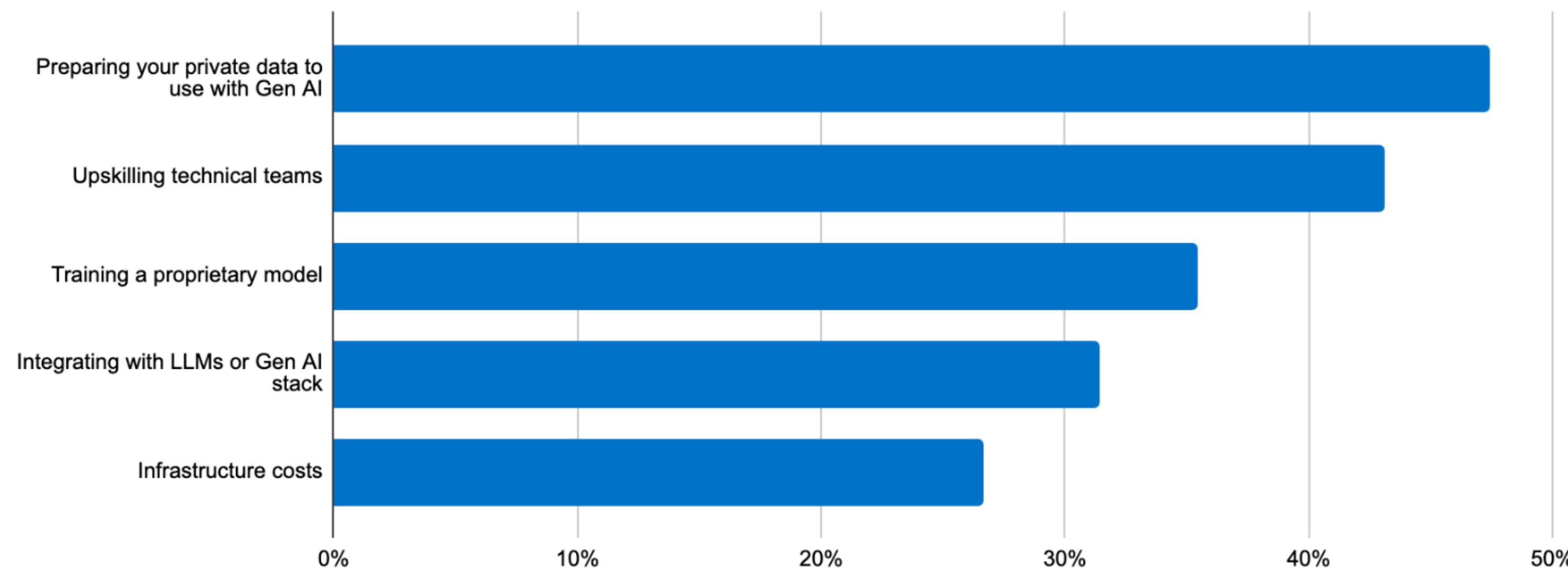


Chart: Where does your organization expect to spend the most time and resources when building generative AI use cases?

What are your organization's top considerations when selecting a vector search engine?

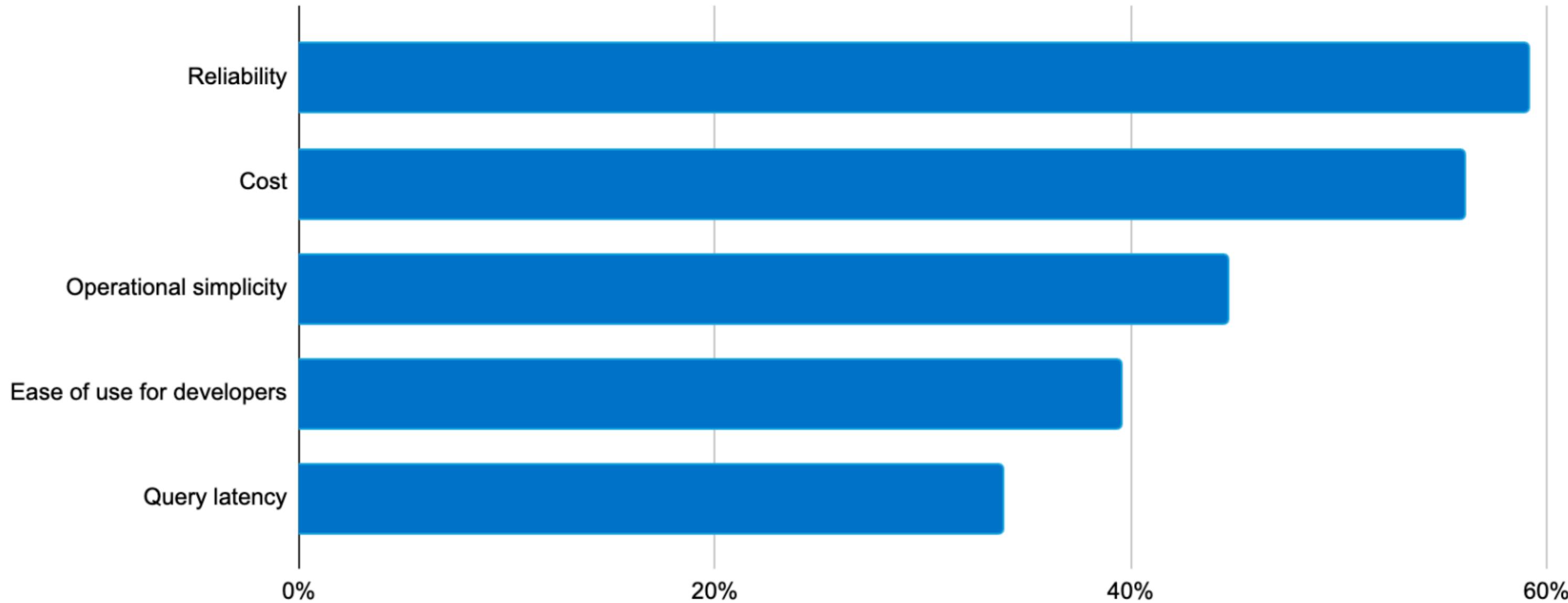


Chart: What are your organization's top considerations when selecting a vector search engine?

# RESOURCES

<https://www.elastic.co/what-is/generative-ai>

<https://www.elastic.co/search-labs/blog/generative-ai-transformers-explained>

<https://www.elastic.co/guide/en/observability/current/obs-ai-assistant.html>

<https://www.elastic.co/guide/en/observability/current/open-telemetry-direct.html>

<https://opentelemetry.io/blog/2023/ecs-otel-semconv-convergence/>

<https://www.elastic.co/blog/ecs-elastic-common-schema-otel-opentelemetry-faq>

<https://github.com/elastic/opentelemetry-demo>

<https://github.com/elastic/elasticsearch-labs>

<https://www.elastic.co/guide/en/machine-learning/current/ml-nlp-elser.html>

<https://www.elastic.co/guide/en/elasticsearch/reference/current/semantic-search-elser.html>

RESOURCES FOR DEVELOPERS | BY DEVELOPERS LIKE YOU!

# Elasticsearch Labs



BLOG / ML RESEARCH

## Evaluating RAG: A journey through metrics



In 2020, Meta published a paper titled "[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)". This paper introduced a method for expanding the knowledge of Language

 **elasticsearch-labs** Public

About  
Elasticsearch Guides, Notebooks & Example Apps for Search Applications  
[search-labs.elastic.co/search-labs](https://search-labs.elastic.co/search-labs)

python search elasticsearch ai  
vector applications openai elastic  
chatlog chatgpt langchain  
openai-chatgpt genai genaistack  
vectordatabase

Readme Apache-2.0 license Security policy Activity  
109 stars 178 watching 40 forks  
Report repository

Languages

Jupyter Notebook	93.7%
Python	2.9%
TypeScript	1.3%
Handlebars	0.1%
JavaScript	1.6%
CSS	0.2%
Other	0.2%



Generative AI  
ML Research  
Vector Search  
How-Tos  
Integrations  
Lucene

[elastic.co/search-labs](https://elastic.co/search-labs)

[github.com/elastic/elasticsearch-labs](https://github.com/elastic/elasticsearch-labs)



Star our repo!



Starred 276





# QUESTIONS?

YOUR FEEDBACK HERE!

# THANK YOU!

