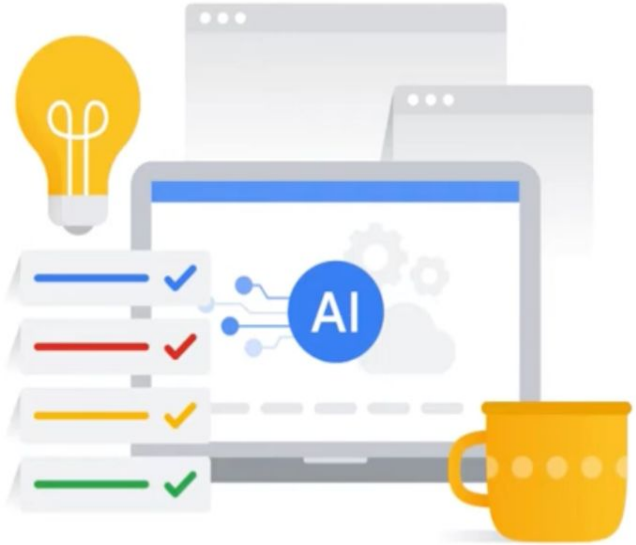
Generative AI

is transforming how we interact with technology.

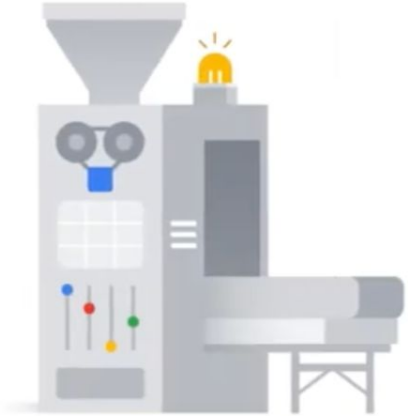


- Marketing manager
- Data scientist
- Application developer

A type of artificial intelligence that generates content for you.


Text
Code
Image
Speech
Video
3D

Prompt

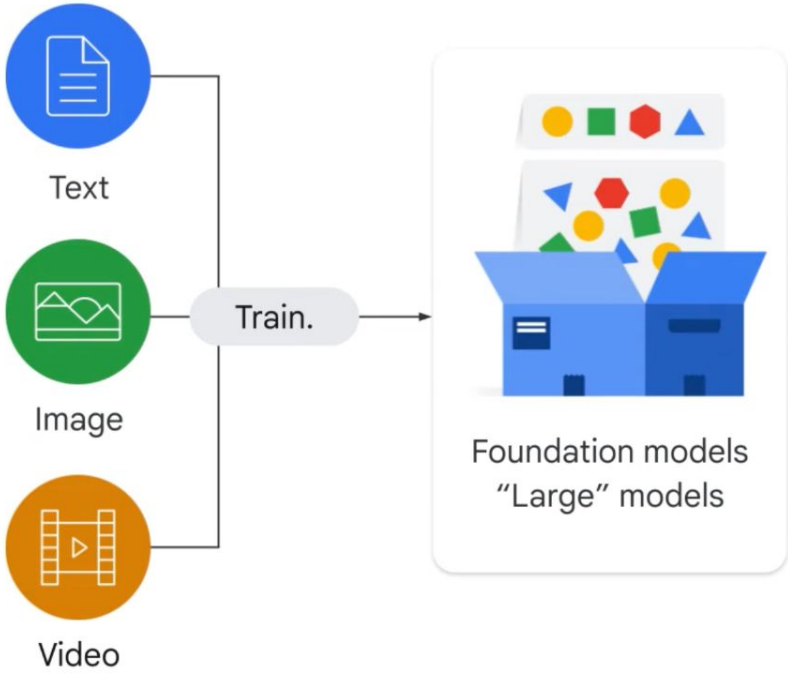


Generative AI

Tasks

- Marketing campaign
- Code generation
- Chatbot creation
- Information extraction
- Document summarization
- Virtual assistance

How does AI generate new content?



- Significant number of parameters
- Massive size of training data
- High requirements of computational power

This revolution started at Google and we continue to innovate

2017
Transformer

Transformer, invented by Google, starts the LLM revolution.

2018
BERT

BERT (Bidirectional Encoder Representations from Transformers), is Google's groundbreaking LLM.

2018
AlphaFold

AlphaFold predicts structures of all known proteins.

2019
T5

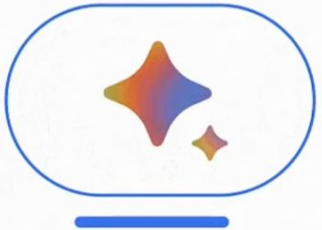
T5 (Text-to-Text Transfer Transformer) is based on Transformer and used to solve multiple NLP tasks.

2021
LaMDA

LaMDA (Language Model for Dialogue Applications) is trained to converse.

2022
PaLM

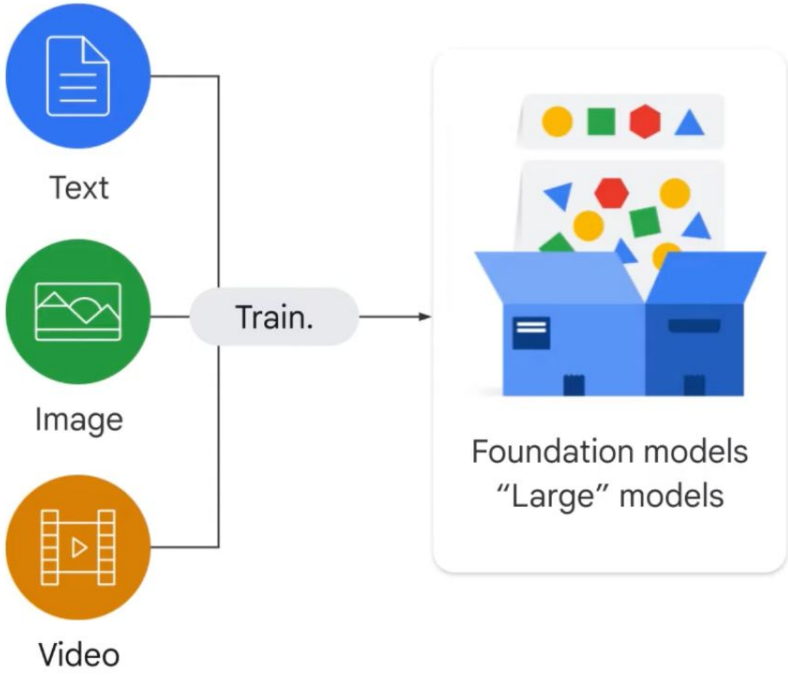
PaLM (Pathways Language Models) is designed for a wide range of general-purpose tasks.



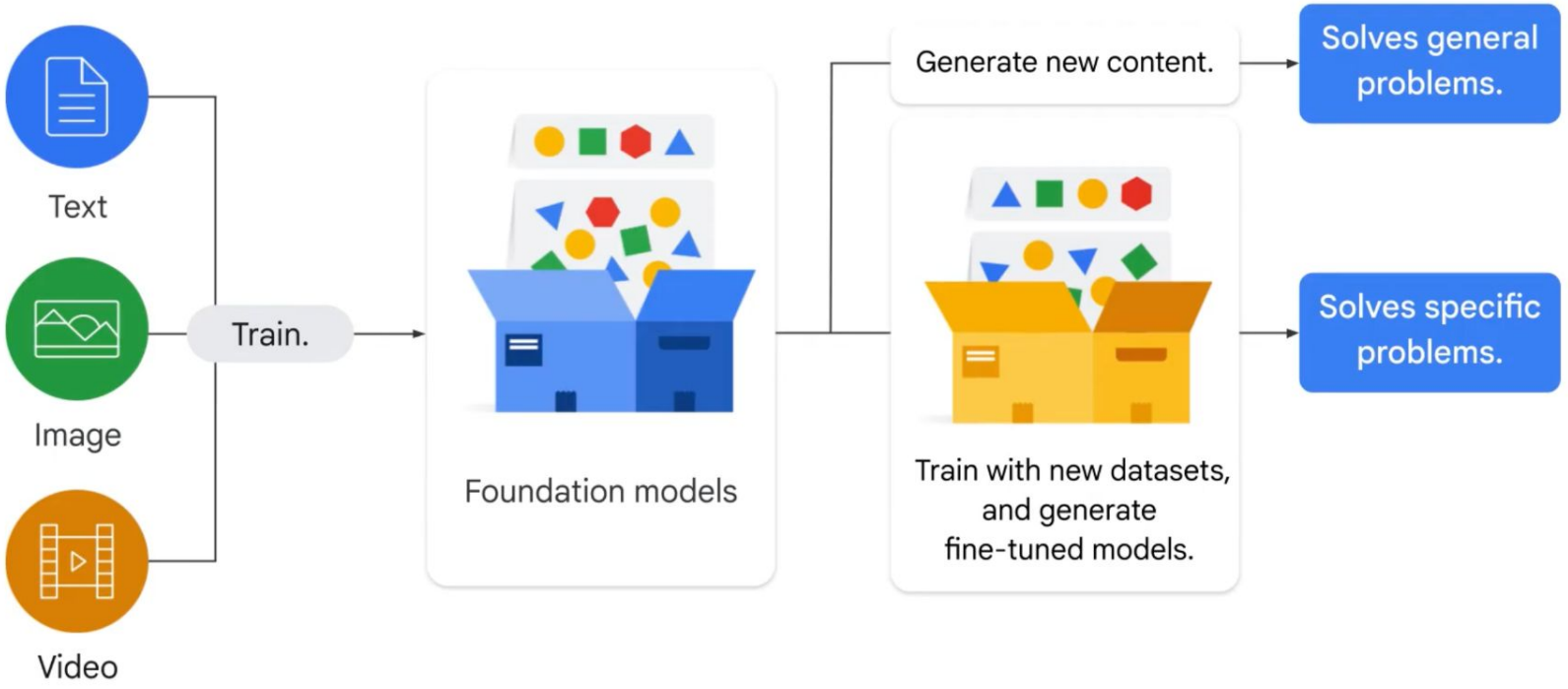
2023
Gemini

Gemini, Google's most recent foundation model, is capable of handling multimodal data.

Responsible AI at the foundation



Gemini for multimodal processing
Gemma for language generation
Codey for code generation
Imagen for image processing


Text
Image
Video
Train.
Foundation models
Generate new content.
Solves general problems.
Train with new datasets, and generate fine-tuned models.
Solves specific problems.

Pre-trained vs. fine-tuned

Training a dog

Sit

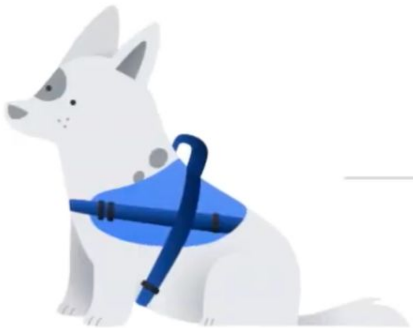
Sit

Come

Down

Stay

Training a dog



Special-service dog

Training a dog

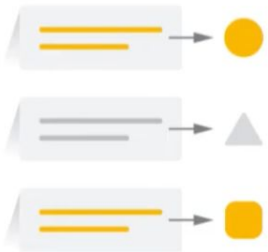
special trainings

Special-service dog

- police dog
- guide dog
- hunting dog

A similar idea applies to pre-trained versus fine-tuned models.

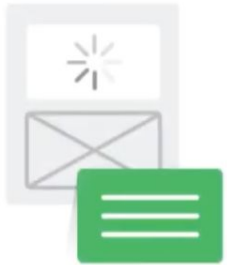
Large language models are trained to solve common language problems, such as:



Text classification

Question answering

Document summarization



Text generation

They can be tailored to solve specific problems in different fields, such as:

Retail

Finance

Entertainment

Trained with relatively small datasets from these fields.

Generative AI is driving new opportunities

Enhance productivity.

Saves operational costs.

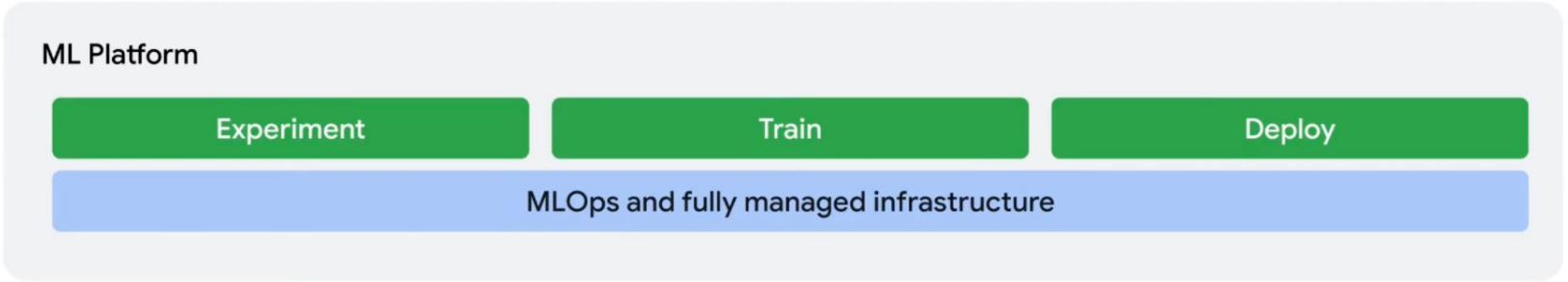
Creates new values.

Vertex AI

End-to-end ML development platform to build, deploy, and manage models

Predictive AI + Generative AI

Vertex AI


ML Platform
Experiment
Train
Deploy
MLOps and fully managed infrastructure

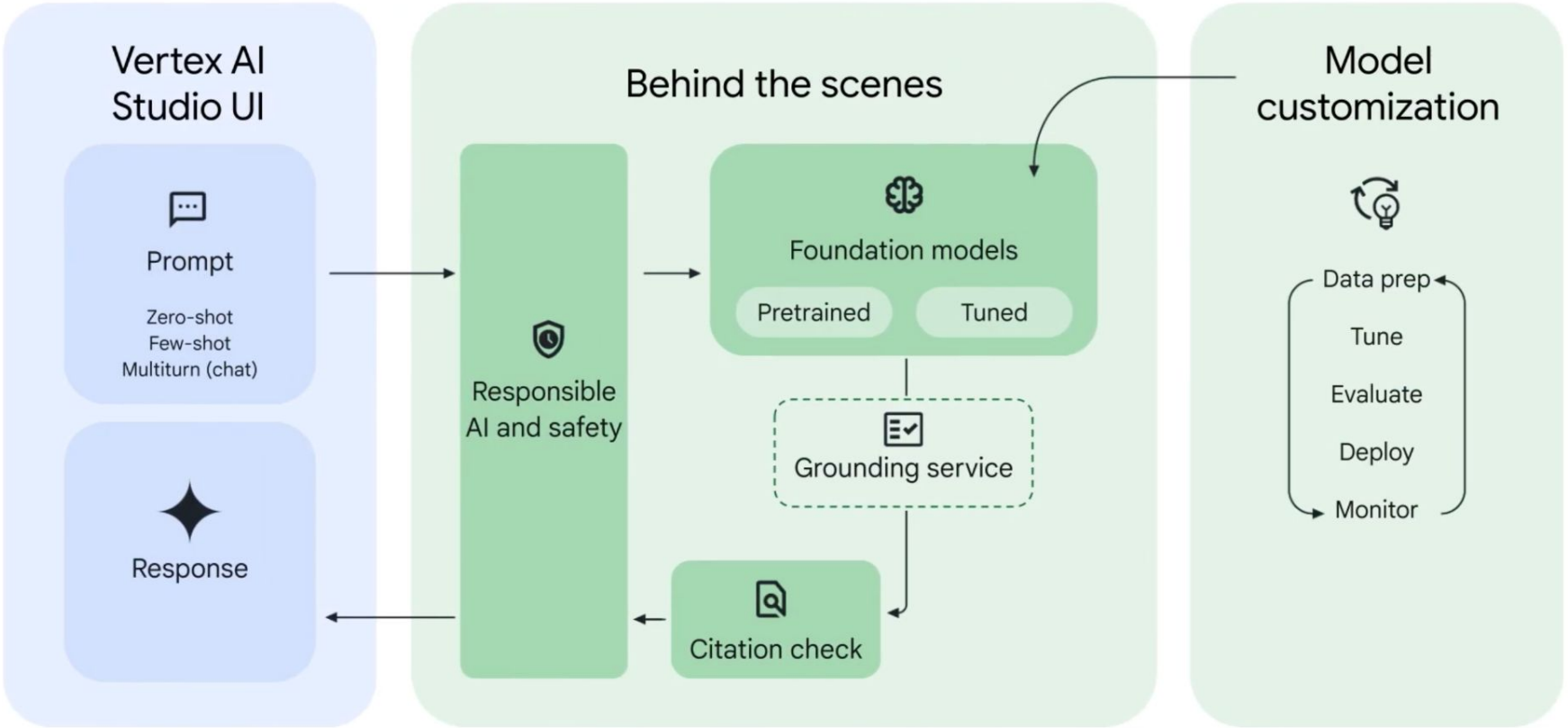
Vertex AI

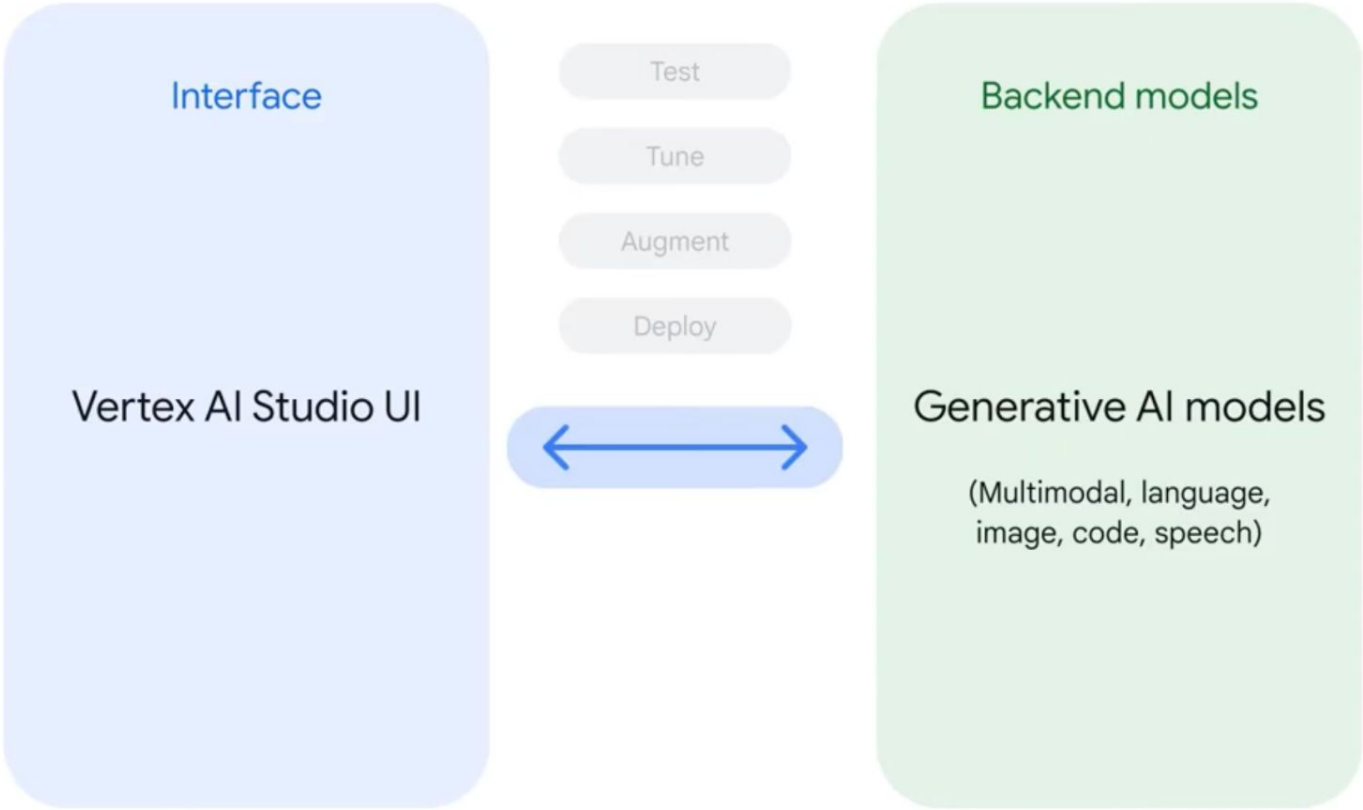
Generative AI

- Vertex AI Studio
  - Gemini multimodal
  - Prompt design
  - Model tuning
- Model Garden
  - Foundation models
  - Fine-tunable models
  - Task-specific solutions

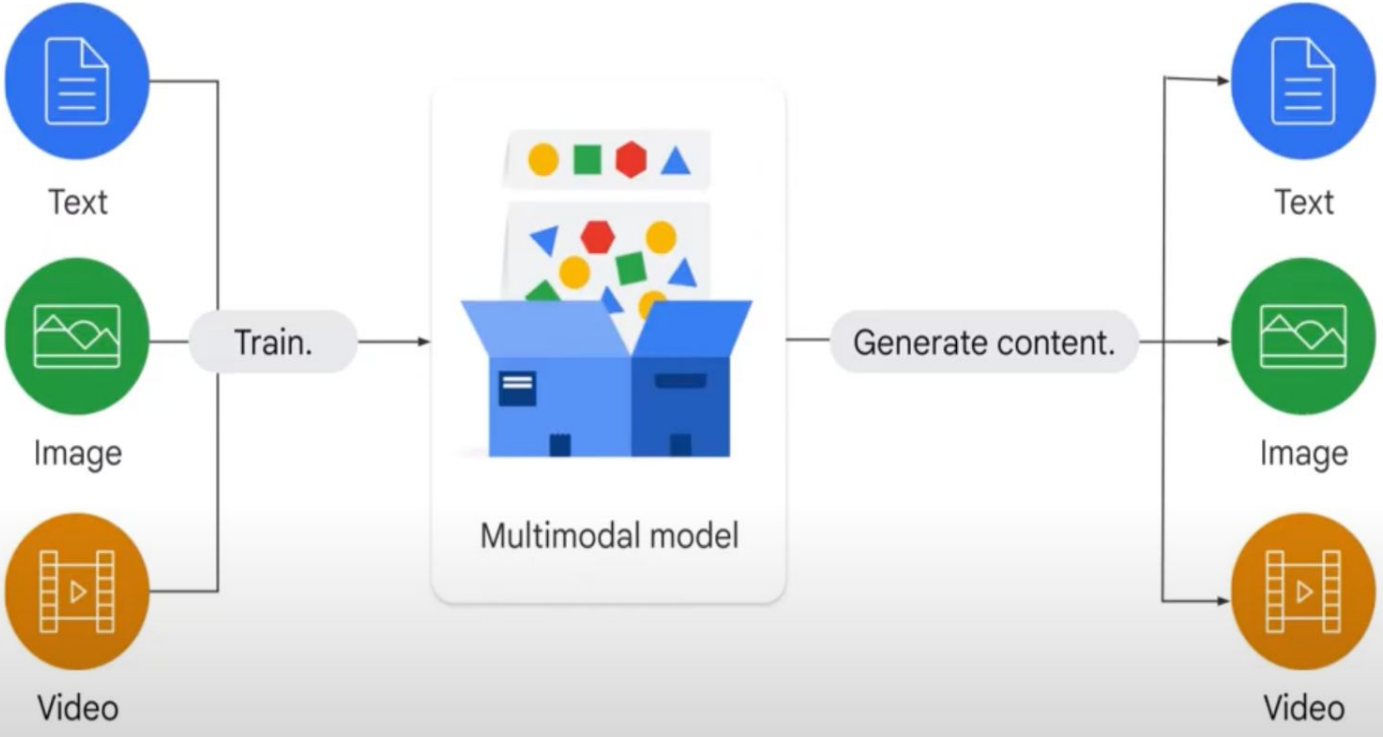
ML Platform

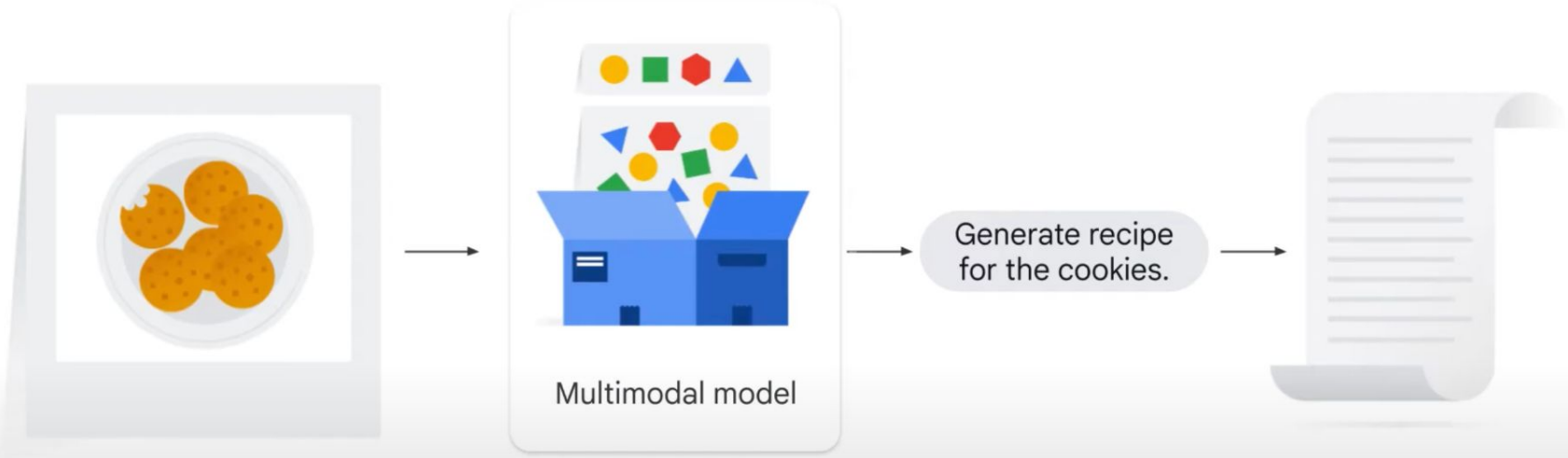
- Experiment
- Train
- Deploy
- MLOps and fully managed infrastructure


Vertex AI Studio UI
Prompt
Zero-shot
Few-shot
Multiturn (chat)
Response
Behind the scenes
Responsible AI and safety
Foundation models
Pretrained
Tuned
Grounding service
Citation check
Model customization
Data prep
Tune
Evaluate
Deploy
Monitor


Interface
Vertex AI Studio UI
Test
Tune
Augment
Deploy
Backend models
Generative AI models
(Multimodal, language, image, code, speech)

What is a multimodal model?


Text
Image
Video
Train.
Multimodal model
Generate content.
Text
Image
Video


Generate recipe for the cookies.
Multimodal model

Description and captioning

Identifying the objects and describing them.

Information extraction

Reading text and extracting information from images and videos.

Information analysis

Analyzing the images and videos based on prompts.

Information seeking

Answering questions or generating Q and A.

Content creation

Generating a story or advertisement.

Data conversion

Converting text responses to formats like JSON.

Can you think of a use case to apply Gemini multimodal?

How to interact with Gemini multimodal?

User interface (UI)

Using a UI with Google Cloud console (no-code).

SDKs

Using SDKs (for example, Java, Python, JavaScript) with a notebook.

APIs

Using APIs with command-line tools.

You start with a prompt.

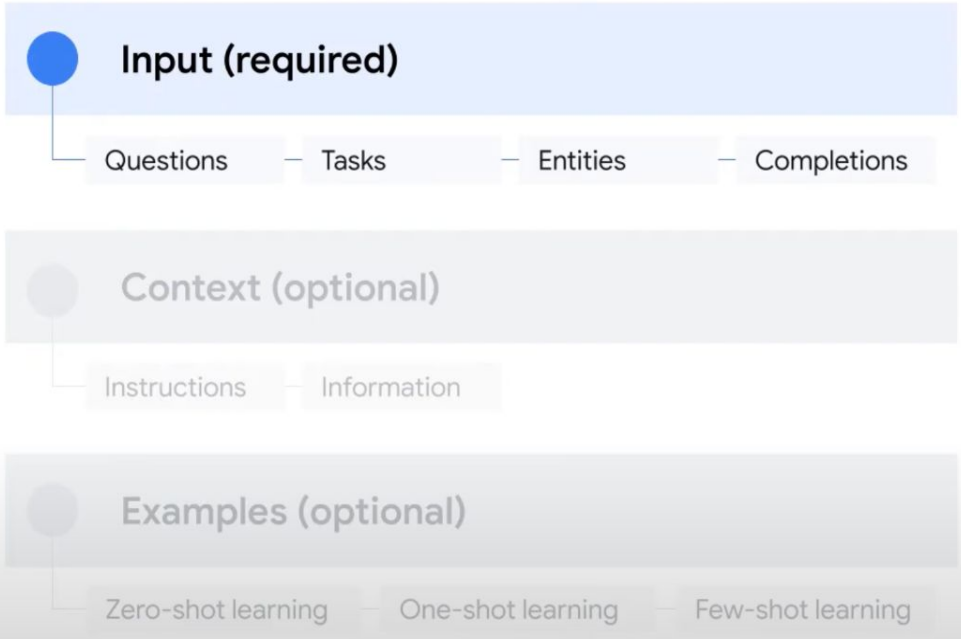
A prompt is a natural language request to a model to receive a response.

The **response** from the model.

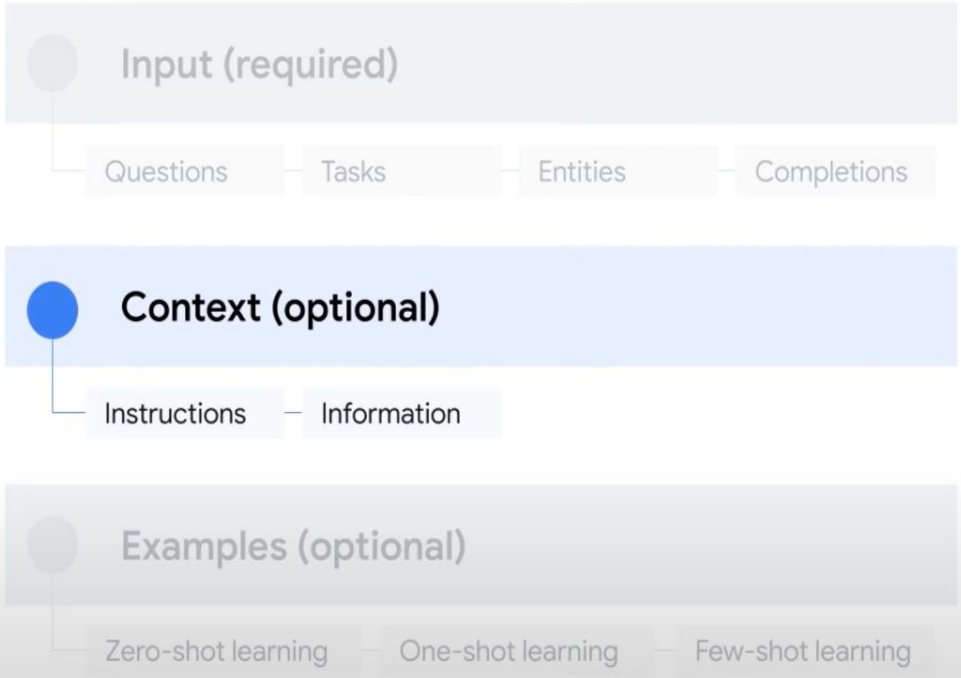
The answers you get depend on the questions you ask.

The **prompt** you designed.

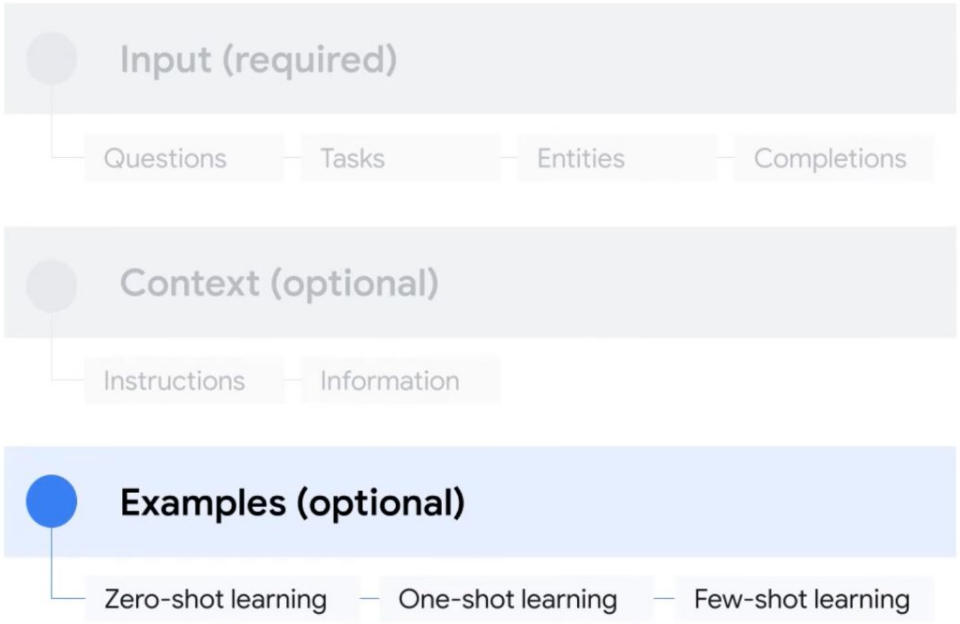
A prompt can include:



A prompt can include:



A prompt can include:



Prompt:

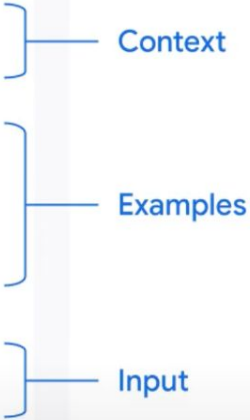
Context:
You are an IT help desk helping customers solve their inquiries.

Examples:
Could't log in: reset the password.
Lost internet connection: check the equipment like a modem and router.
Screen went black: restart the computer.

Input (test):
What should I do when my computer freezes?

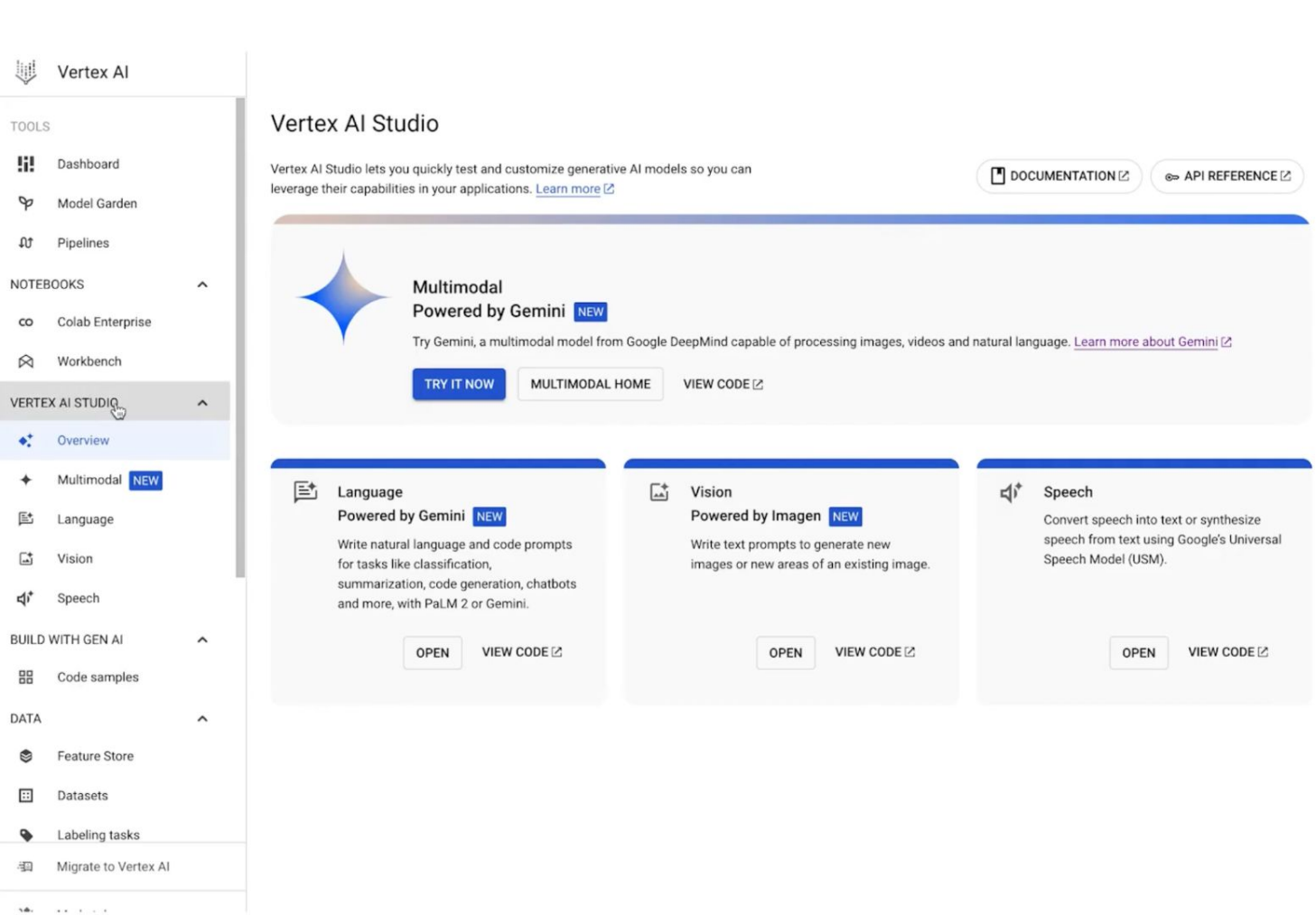
Response:

Please restart your computer.



Prompt design

The process of designing the input text to get the desired response back from the model.


Vertex AI
TOOLS
Dashboard
Model Garden
Pipelines
NOTEBOOKS
Colab Enterprise
Workbench
VERTEX AI STUDIO
Overview
Multimodal NEW
Language
Vision
Speech
BUILD WITH GEN AI
Code samples
DATA
Feature Store
Datasets
Labeling tasks
Migrate to Vertex AI
Vertex AI Studio
Vertex AI Studio lets you quickly test and customize generative AI models so you can leverage their capabilities in your applications. Learn more
DOCUMENTATION
API REFERENCE
Multimodal
Powered by Gemini NEW
Try Gemini, a multimodal model from Google DeepMind capable of processing images, videos and natural language. Learn more about Gemini
TRY IT NOW
MULTIMODAL HOME
VIEW CODE
Language
Powered by Gemini NEW
Write natural language and code prompts for tasks like classification, summarization, code generation, chatbots and more, with PaLM 2 or Gemini.
OPEN
VIEW CODE
Vision
Powered by Imagen NEW
Write text prompts to generate new images or new areas of an existing image.
OPEN
VIEW CODE
Speech
Convert speech into text or synthesize speech from text using Google's Universal Speech Model (USM).
OPEN
VIEW CODE

Zero-shot prompting

Providing a single command to the LLM without any example.

One-shot prompting

Providing a single example of the task to the LLM.

Few-shot prompting

Providing a few examples of the task to the LLM.

Best practices for prompt design

- Be concise.
- Be specific.
- Ask one task at a time.
- Include examples.

The garden was full of beautiful… → LLM → [flowers (0.55), trees (0.23), herbs (0.10), … , bugs (0.03)]

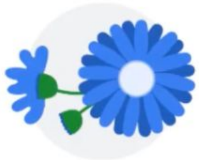
The garden was full of beautiful...

`[flowers (0.5), trees (0.23), herbs (0.05), …, bugs (0.03)]`

- Narrow the range to **high-possibility** words.
- Use it when you expect a more "typical" answer.

- Extend the range to **low-possibility** words.
- Use it when you want to generate more "creative" content.

The garden was full of beautiful...



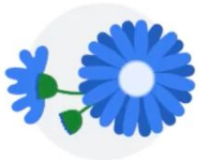

`[flowers (0.5), trees (0.23), herbs (0.05), …, bugs (0.03)]`

Top K

The model returns a random word from a set of top K possible words.

For example: K = 2 flower (0.5) and tree (0.23)

The garden was full of beautiful...




`[flowers (0.5), trees (0.23), herbs (0.05), …, bugs (0.03)]`

Top P

The model returns a random word from the smallest subset with the sum of the likelihoods that exceeds or equals to P.

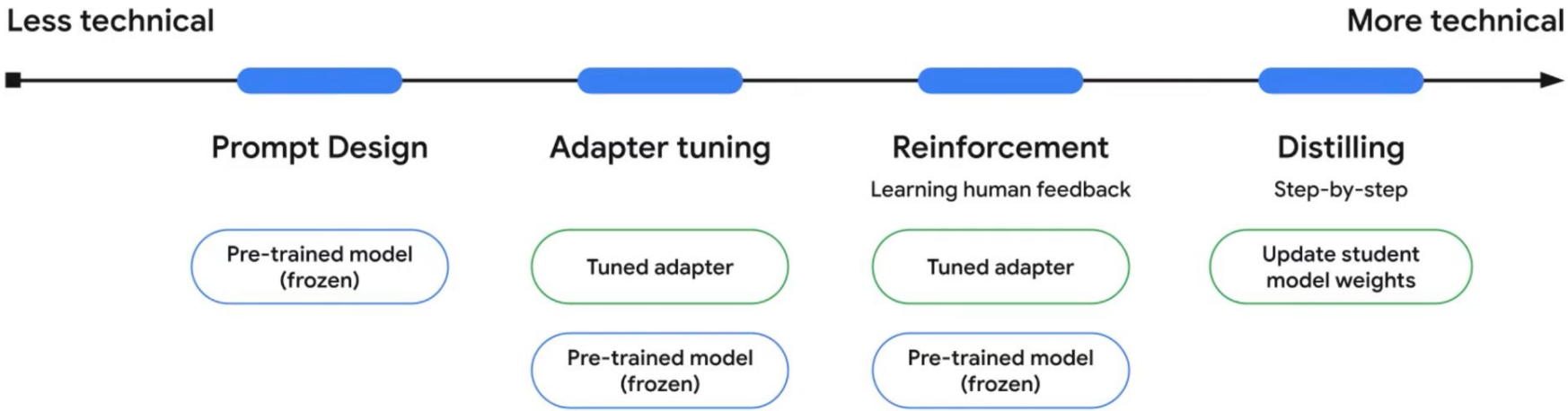
For example: P = 0.75 Flowers (0.5)+trees (0.23)+herbs (0.05)=0.78 (> 0.75)

Model parameters

- Model type
- Top K (number)
- Temperature
- Top P (probability)

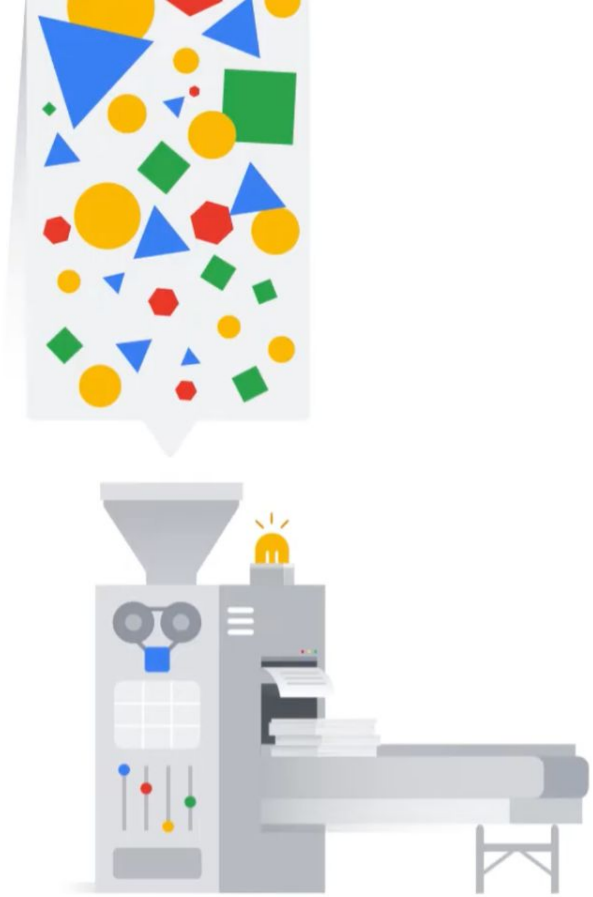
Are there ways to enhance the quality of responses beyond just prompt design?

How to customize and tune a gen AI model



Prompt design:

- Doesn’t change any parameters of the pre-trained model.
- Allows fast experimentation and fast customization.
- Doesn't require ML background nor code skills.

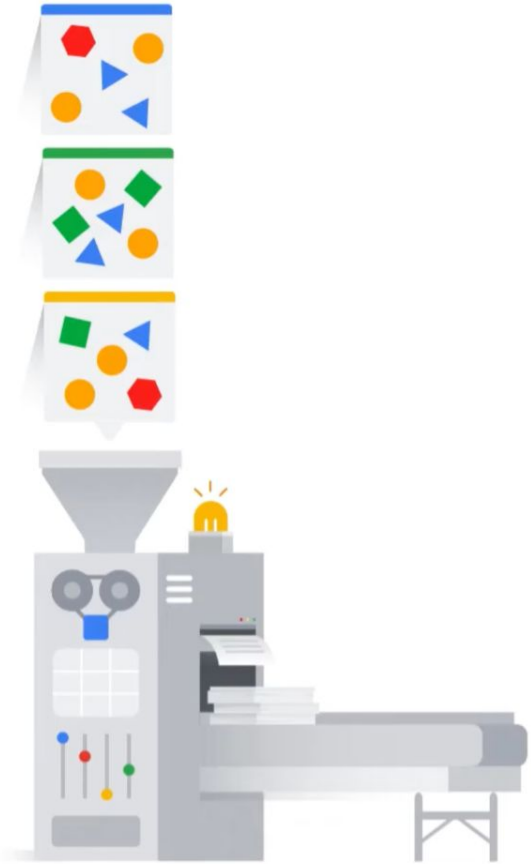


Fine tuning:

- High computation
- High cost
- Time consuming

Parameter-efficient tuning:

- Is a super exciting area of research.
- Aims to reduce the challenges of fine-tuning.
- Only trains a subset of parameters.



Foundation models

Gemini multimodal

Embeddings for text and multimodal

Gemini for text and chat

Imagen for image

Chirp for speech

Codey for code

Task-specific solutions

Entity analysis

Sentiment analysis

Syntax analysis

Content classification

Object detector

Text translation

Modalities	Language, vision, and speech
Tasks	Generation, classification, and detection
Features	Pipeline, notebook, and one-click deployment support

Neural Networks & Deep Learning – the foundation of generative models

Large Language Models (LLMs) – trained on massive text data to predict words

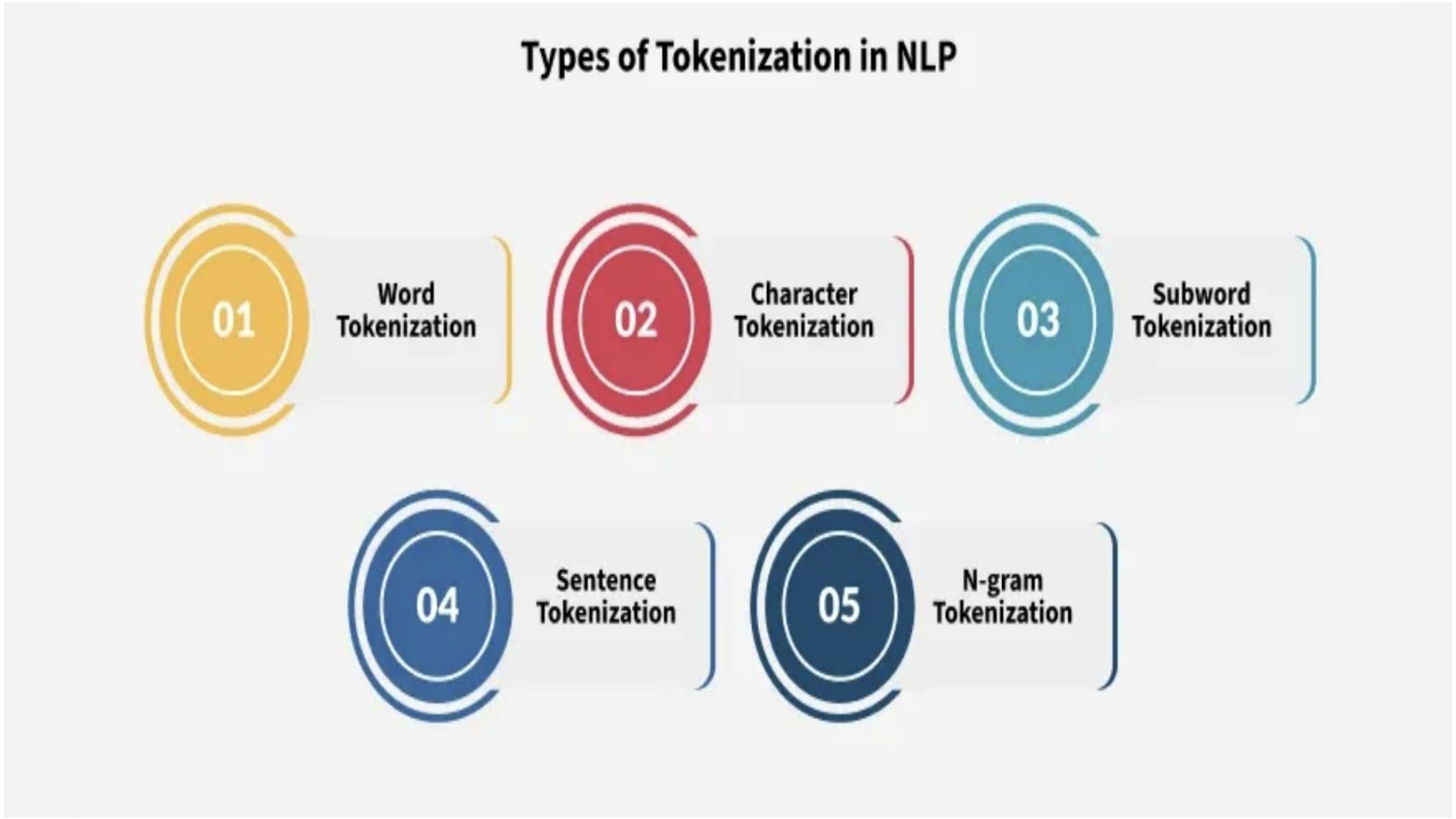
Tokens & Tokenization – how text is broken down for the model to understand

Training Process – how models learn from billions of examples

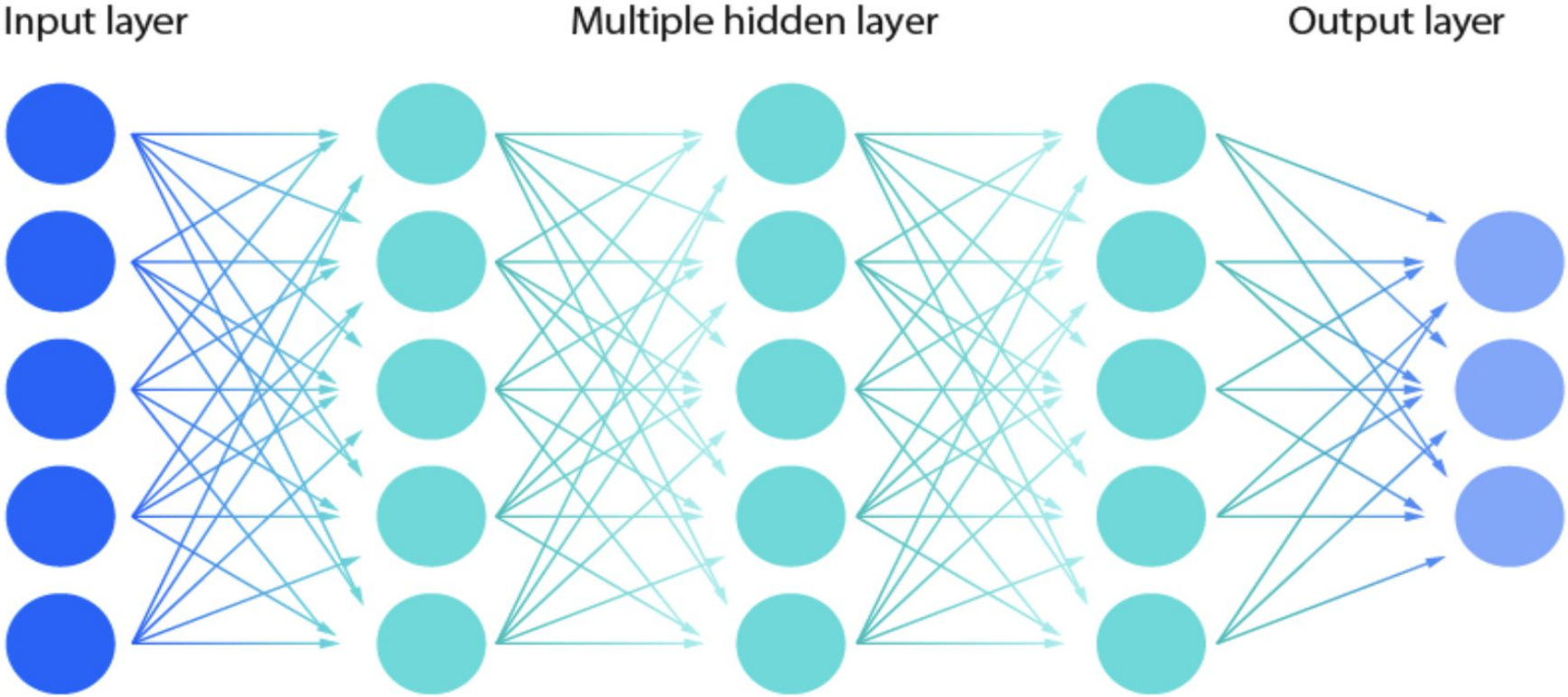
Inference – how outputs are generated from learned patterns

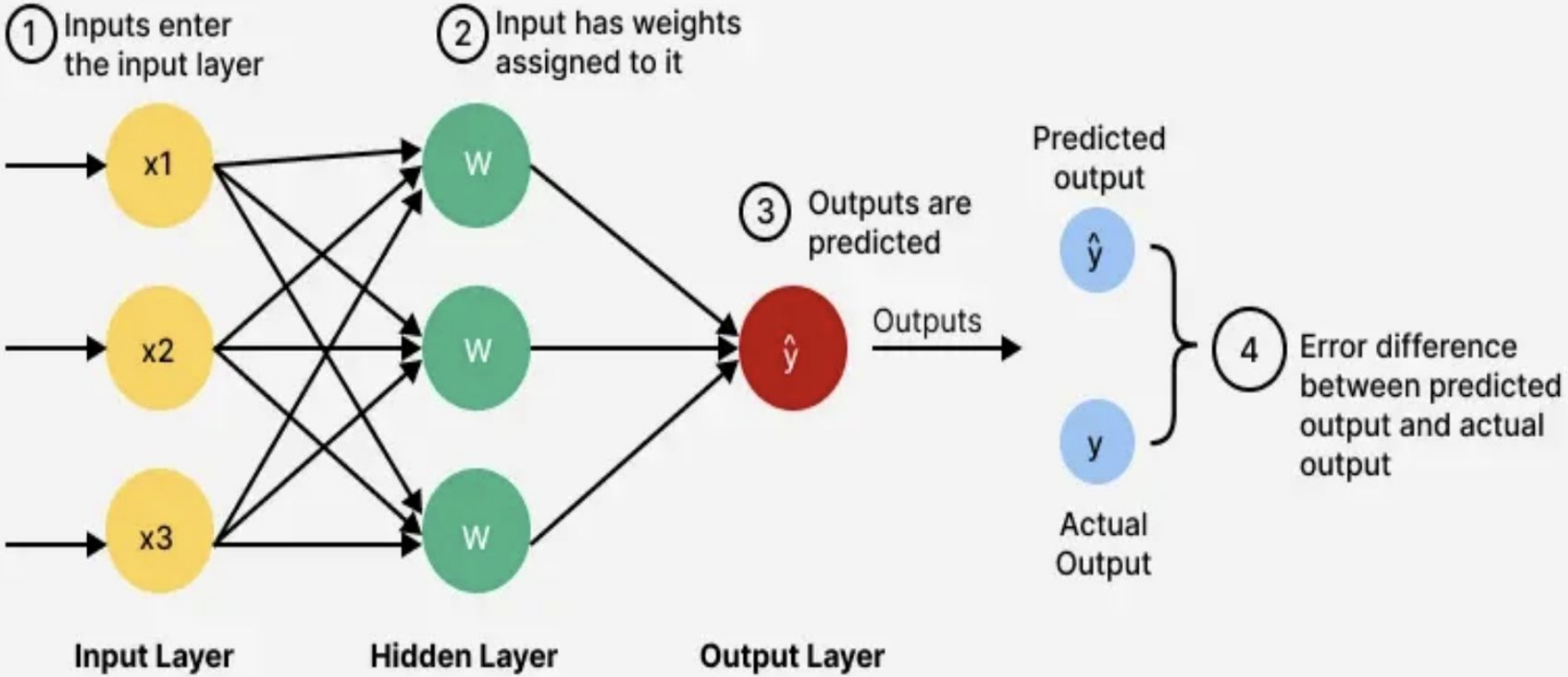
Generative Models Beyond Text – image, audio, video, and code generation

	AI training	AI fine-tuning	AI inference	AI serving
Objective	Build a new model from scratch.	Adapt a pre-trained model for a specific task.	Use a trained model to make predictions.	Deploy and manage the model to handle inference requests.
Process	Iteratively learns from a large dataset.	Refines an existing model with a smaller dataset.	A single, fast "forward pass" of new data.	Package the model and expose it as an API
Data	Large, historical, labeled datasets.	Smaller, task-specific datasets.	Live, real-world, unlabeled data.	N/A
Business focus	Model accuracy and capability.	Efficiency and customization.	Speed (latency), scale, and cost-efficiency.	Reliability, scalability, and manageability of the inference endpoint.

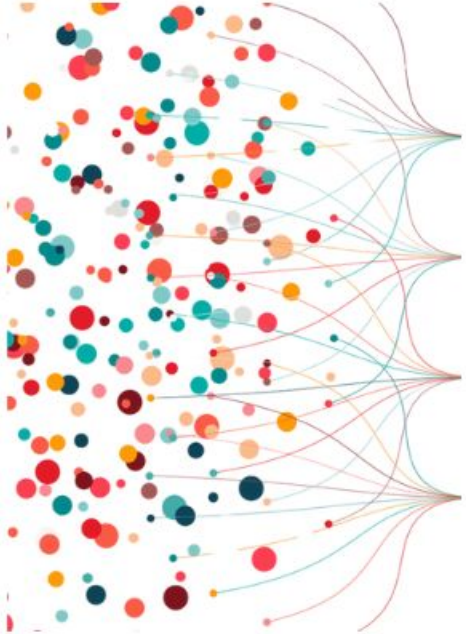

Types of Tokenization in NLP
01
Word Tokenization
02
Character Tokenization
03
Subword Tokenization
04
Sentence Tokenization
05
N-gram Tokenization

Deep neural network




1 Inputs enter the input layer
2 Input has weights assigned to it
x1
x2
x3
W
W
W
3 Outputs are predicted
ŷ
Outputs
Predicted output
ŷ
y
Actual Output
4 Error difference between predicted output and actual output
Input Layer
Hidden Layer
Output Layer

Data



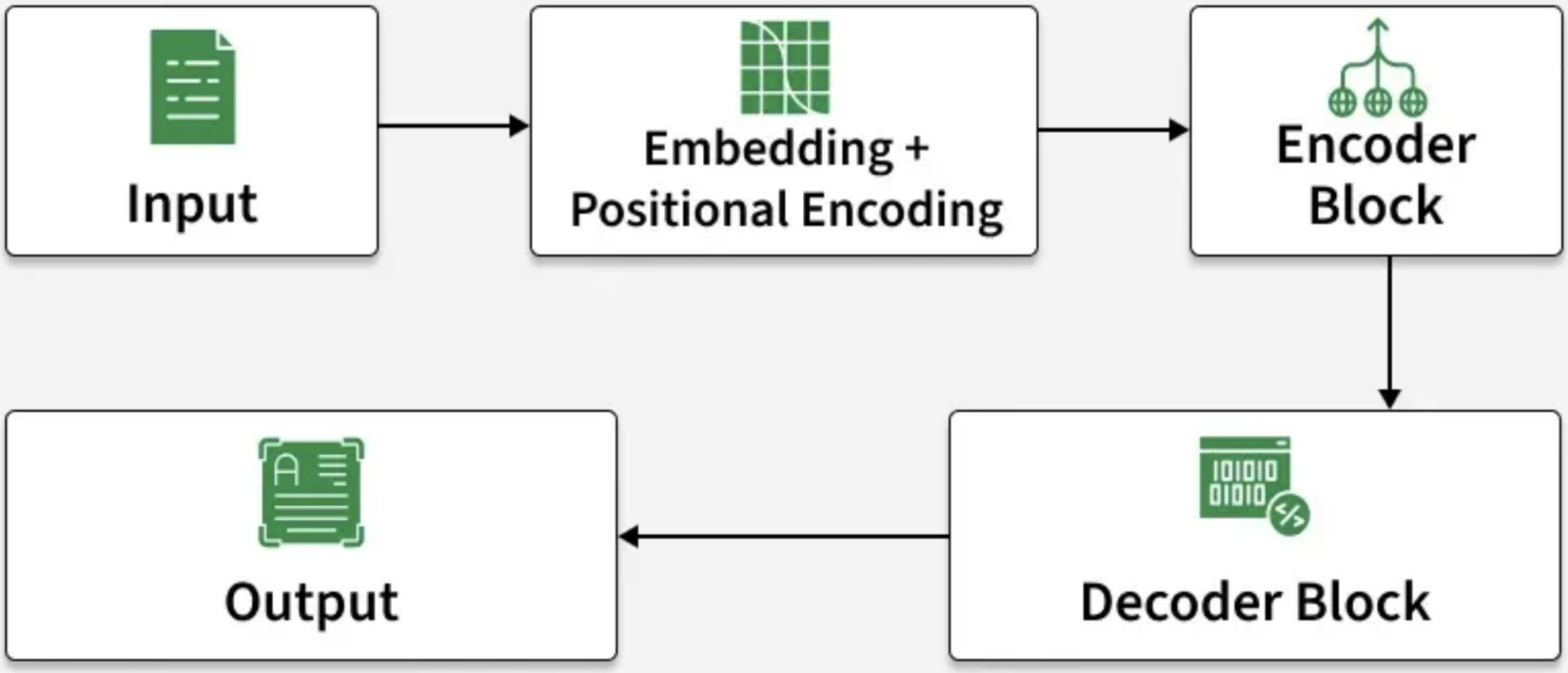
Vector Embeddings/
Vector Data

```
-0.32643065,
-0.12308089,
-0.2873811 ,
-0.99628943,
-0.2503798 ,
 0.24311952,
 0.5662387 ,
 0.17282294,
-0.1109335 ,
 0.15209009,
 0 47017908,
-0.19270805,
```

Vector Database

SingleStore

Transformers in LLMs



- **Input Embeddings** : Converting text into numerical vectors.
- **Positional Encoding** : Adding sequence/order information.
- **Self-Attention** : Understanding relationships between words in context.
- **Feed-Forward Layers** : Capturing complex patterns.
- **Decoding:** Generating responses step-by-step.
- **Multi-Head Attention** : Parallel reasoning over multiple relationships.

Showcase 2 Tools Live

Text generation tool

Image generation tool

Ethics, Risks, and Q&A

Bias, misinformation, deepfakes

Responsible use & regulations

https://github.com/Shashikumar-ezhilarasu/Gen-Ai