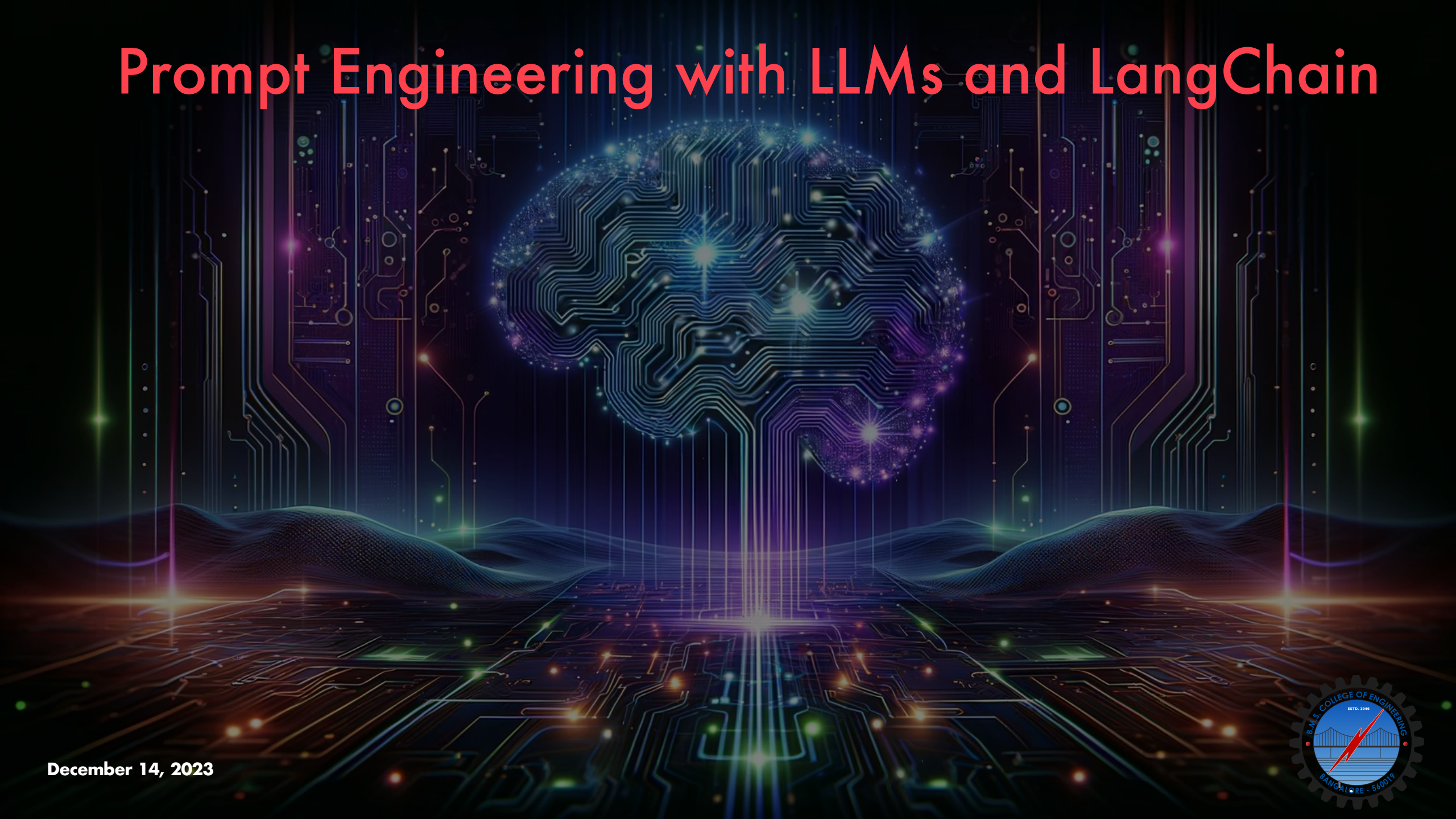
Prompt Engineering with LLMs and LangChain



December 14, 2023

About the Speaker

https://bit.ly/ravi-ranjan-03

GitHub https://bit.ly/raviranjan0309

Ravi Ranjan is working as Manager of Data Science at Publicis Sapient (India), specializing in creating scalable ML solutions. A certified Google Cloud Architect, he has extensive experience in designing and implementing AI and ML systems, including scalable recommendation platforms. In addition, he actively contributes to and is a member of Kubeflow, an ML platform by Google.

publicis
sapient

Today’s Session

1. What is Generative AI and its Evolution?
2. How LLM is driving the current market?
3. Understanding of prompt engineering
4. Basics of Langchain
5. Demo
6. QnA

Artificial Intelligence

Is the field of study

Machine Learning

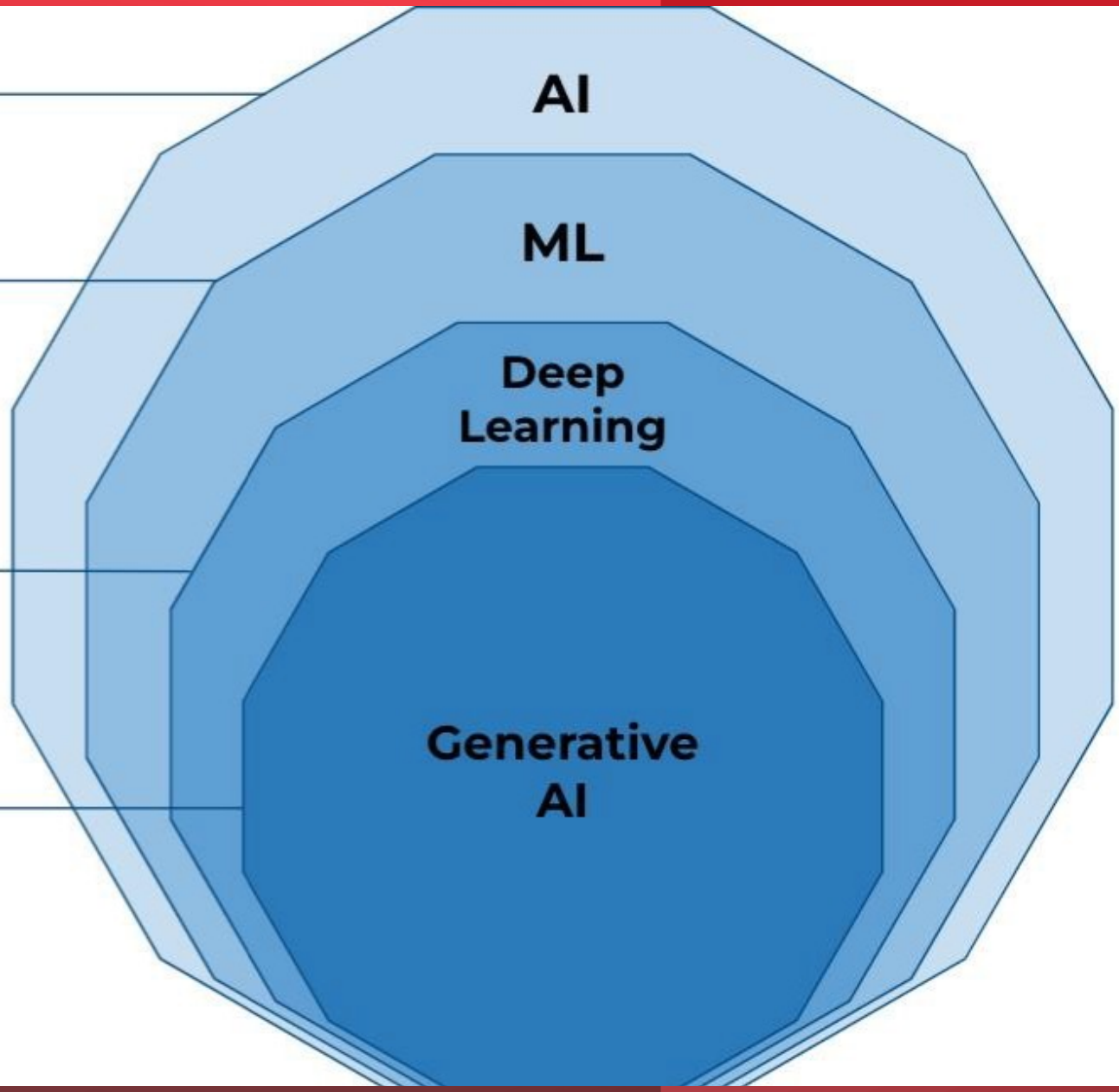
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

Deep Learning

Is a subset of Machine Learning methods, based on **Artificial Neural Networks.**
Examples: CNNs, RNNs

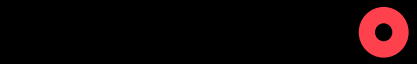
Generative AI

A type of ANNs that generate data that is similar to the data it was trained on.
Examples: GANs, LLMs

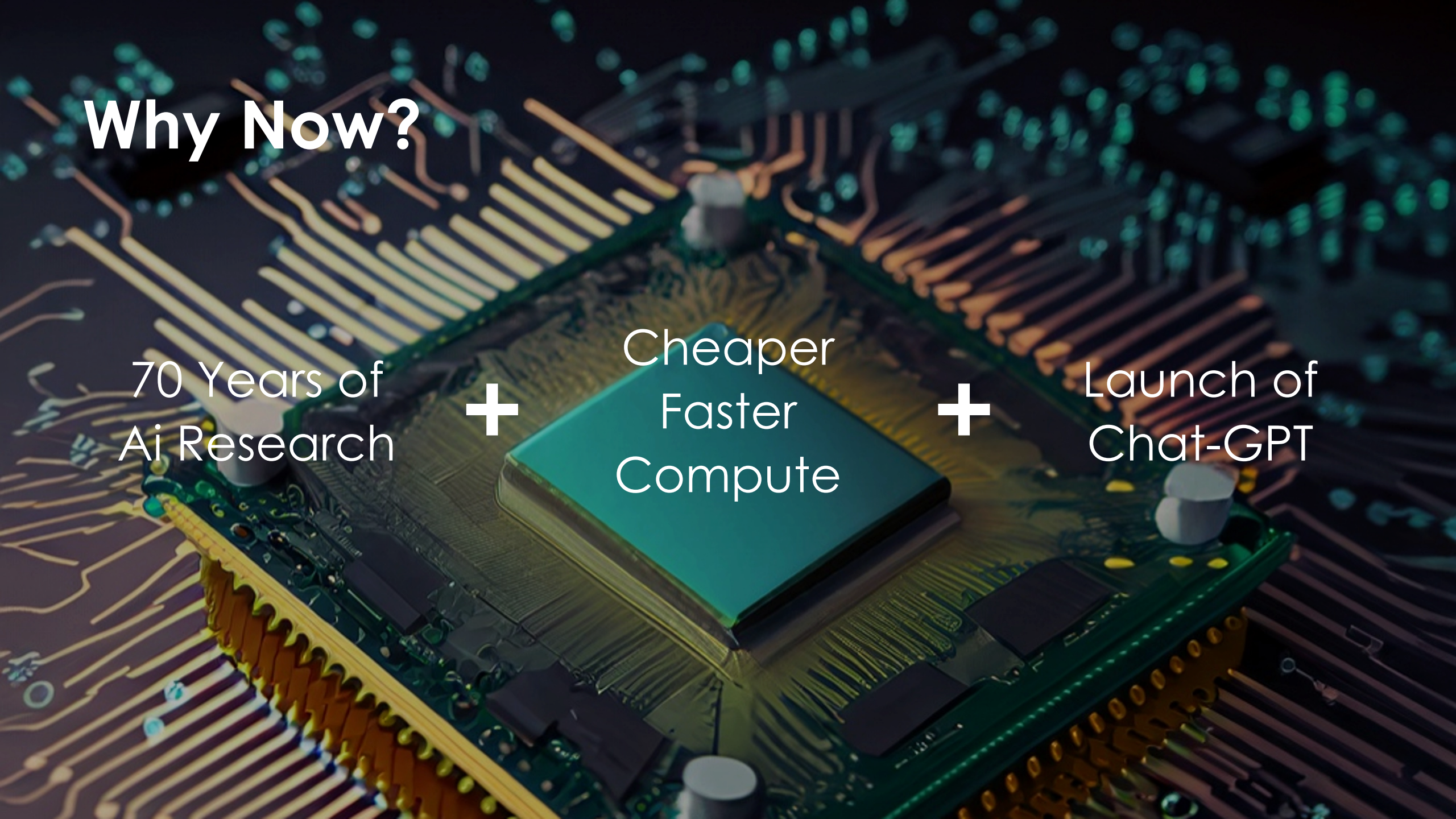



Generative AI Storm

30 NOVEMBER 2022
OPEN AI RELEASED CHATGPT



Why Now?

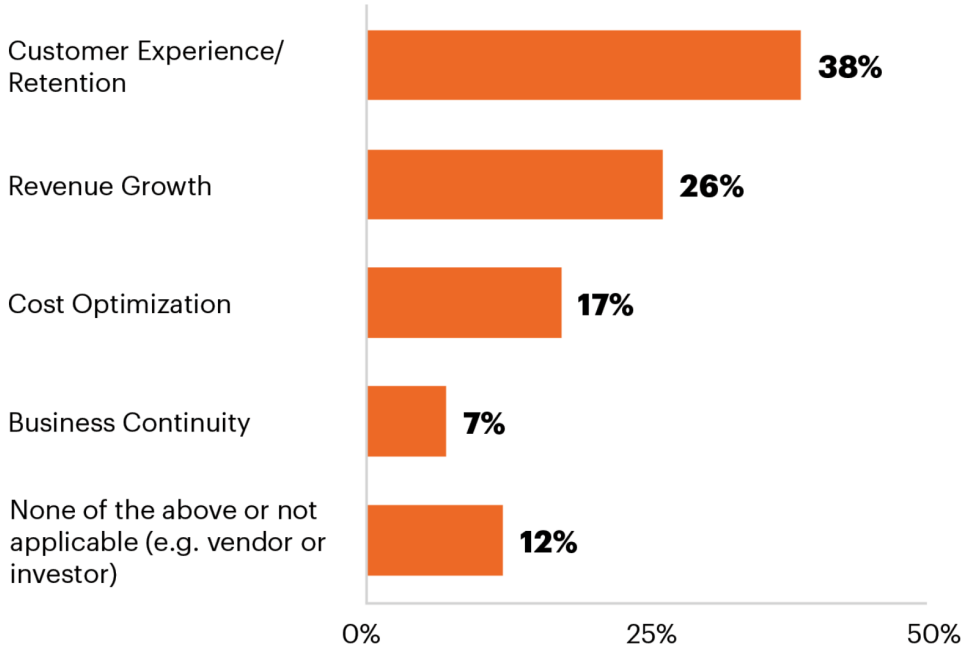

70 Years of
Ai Research
+
Cheaper
Faster
Compute
+
Launch of
Chat-GPT

The Journey to Generative AI

A Series of Increasingly Frequent Breakthroughs That Make Sense of Natural Language

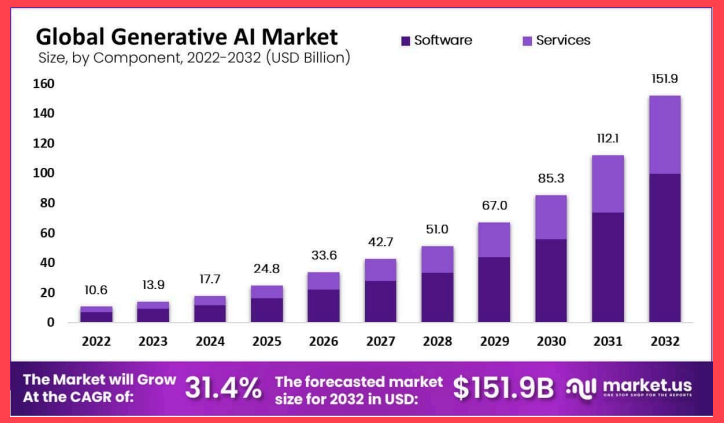
2010	2014	2017-2022	2022
Near-Perfect Translation of Natural Language	**Mastering the Meaning of Words**	**Large Language Foundation Models**	**Conversational Large Language Foundation Models**
Around 2010, AI researchers working on natural language translation discovered that models exposed to vast amounts of text produced much better results than models using top-down grammatical rules.	In 2014, language models began to make sense of the meaning of words in a natural language by analyzing the context in which the word appeared.	Advances made from 2017 to 2022 resulted in language models that can serve as a foundation for customization. Creating foundation models is cost-prohibitive, but once created, they can be customized using a small amount of additional data to achieve state-of-the-art performance on new tasks without significant investment.	2022 marked the arrival of ChatGPT, which gave users a simple way to access a large language foundational model.The brilliance of ChatGPT is not just in the incredibly advanced model at its core; equally, it is the ability to tap into this model by conversing with it in natural language. As AI researcher Andrej Karpathy quips, "Now the hottest programing language is English!"

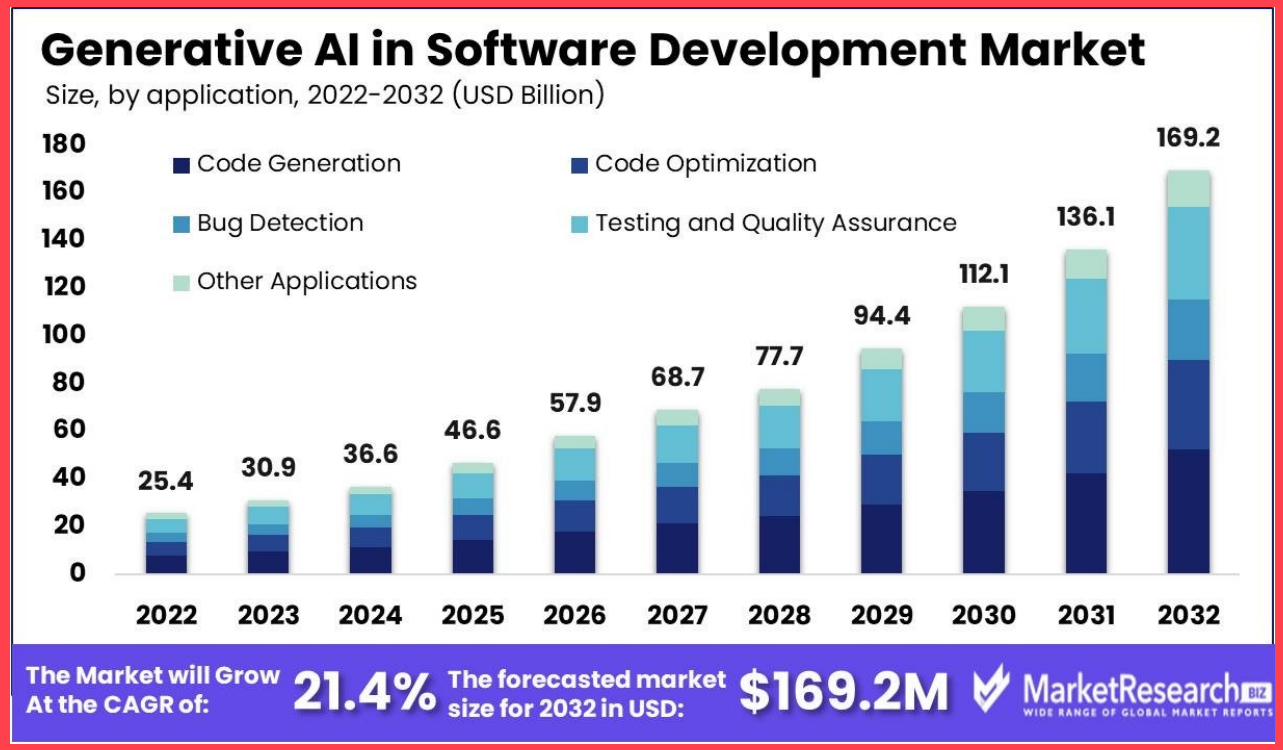
Primary Focus of Generative AI Initiatives



Gartner

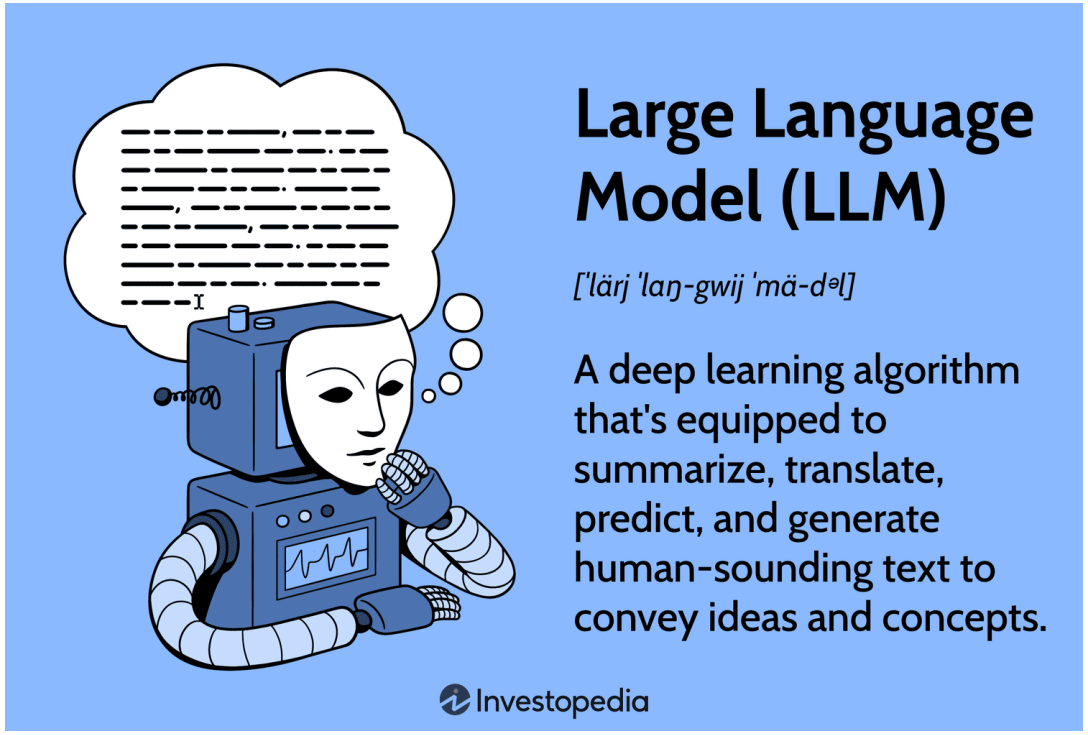
Gartner Says More Than 80% of Enterprises Will Have Used Generative AI APIs or Deployed Generative AI-Enabled Applications by 2026

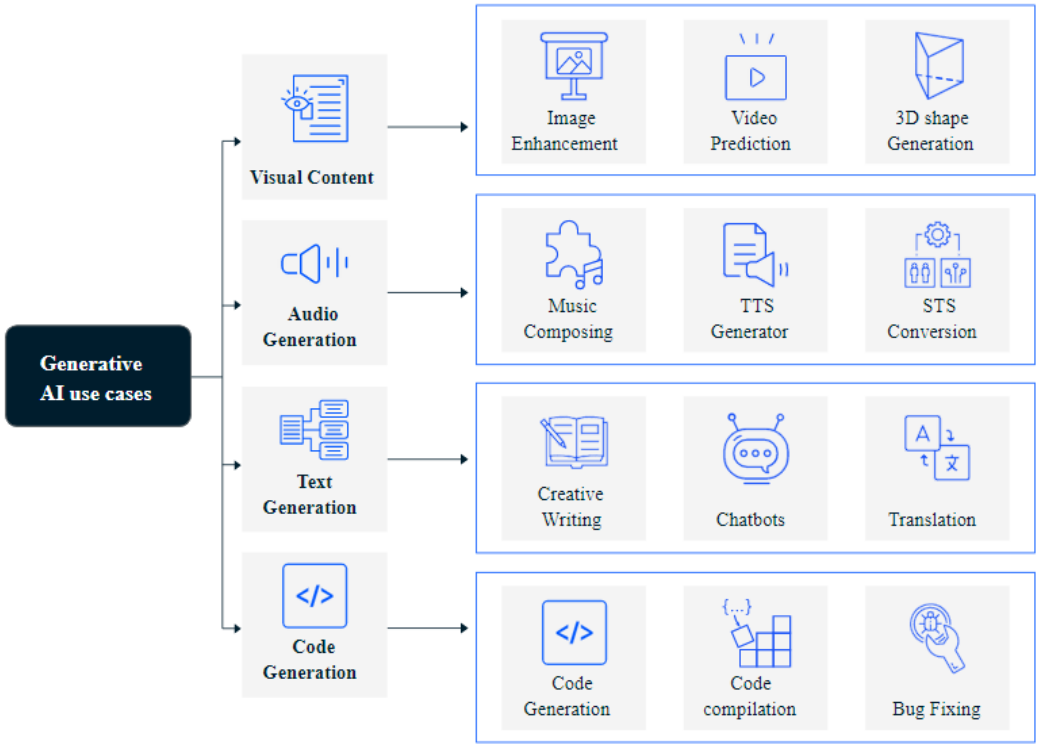


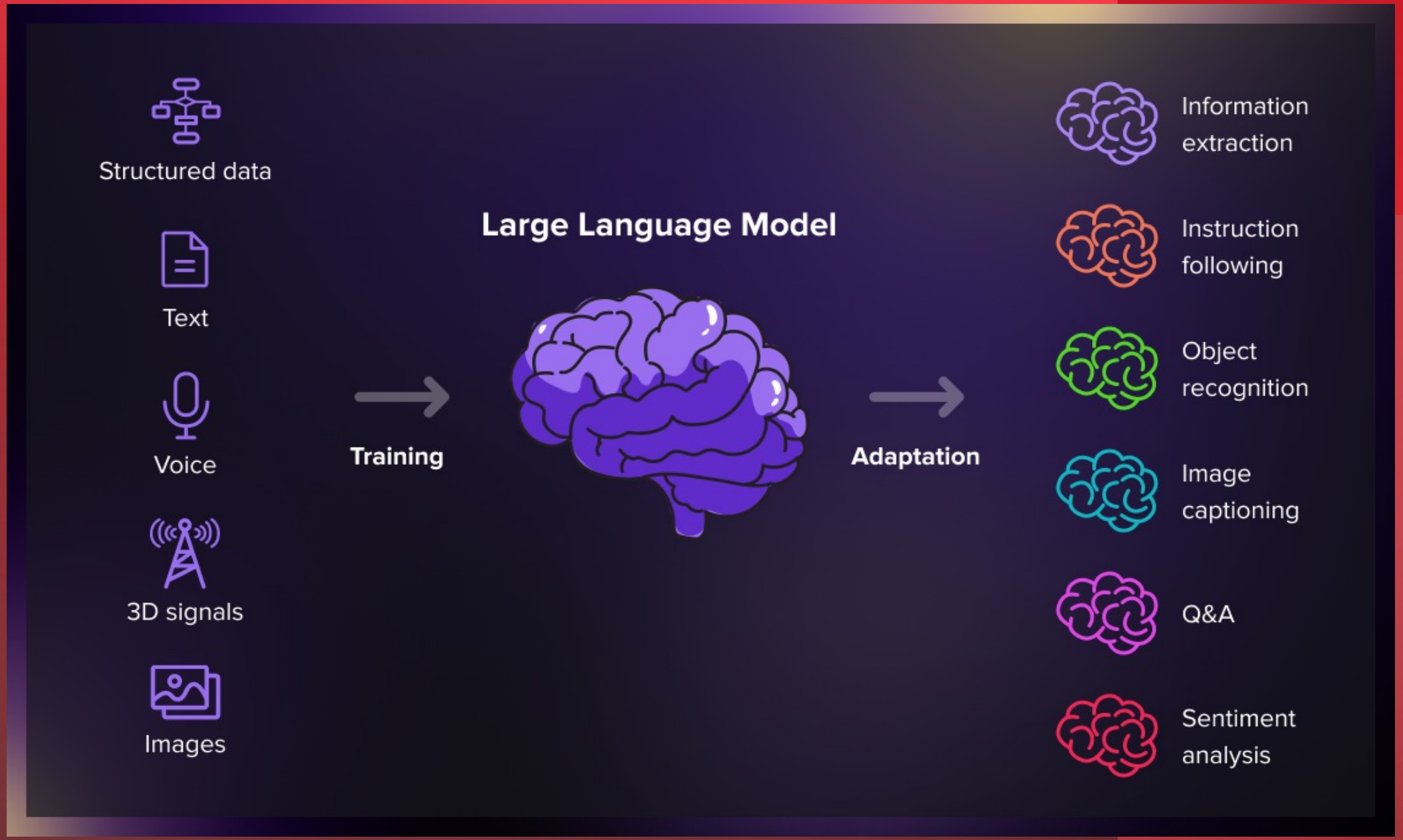


Generative AI = Massive Opportunity

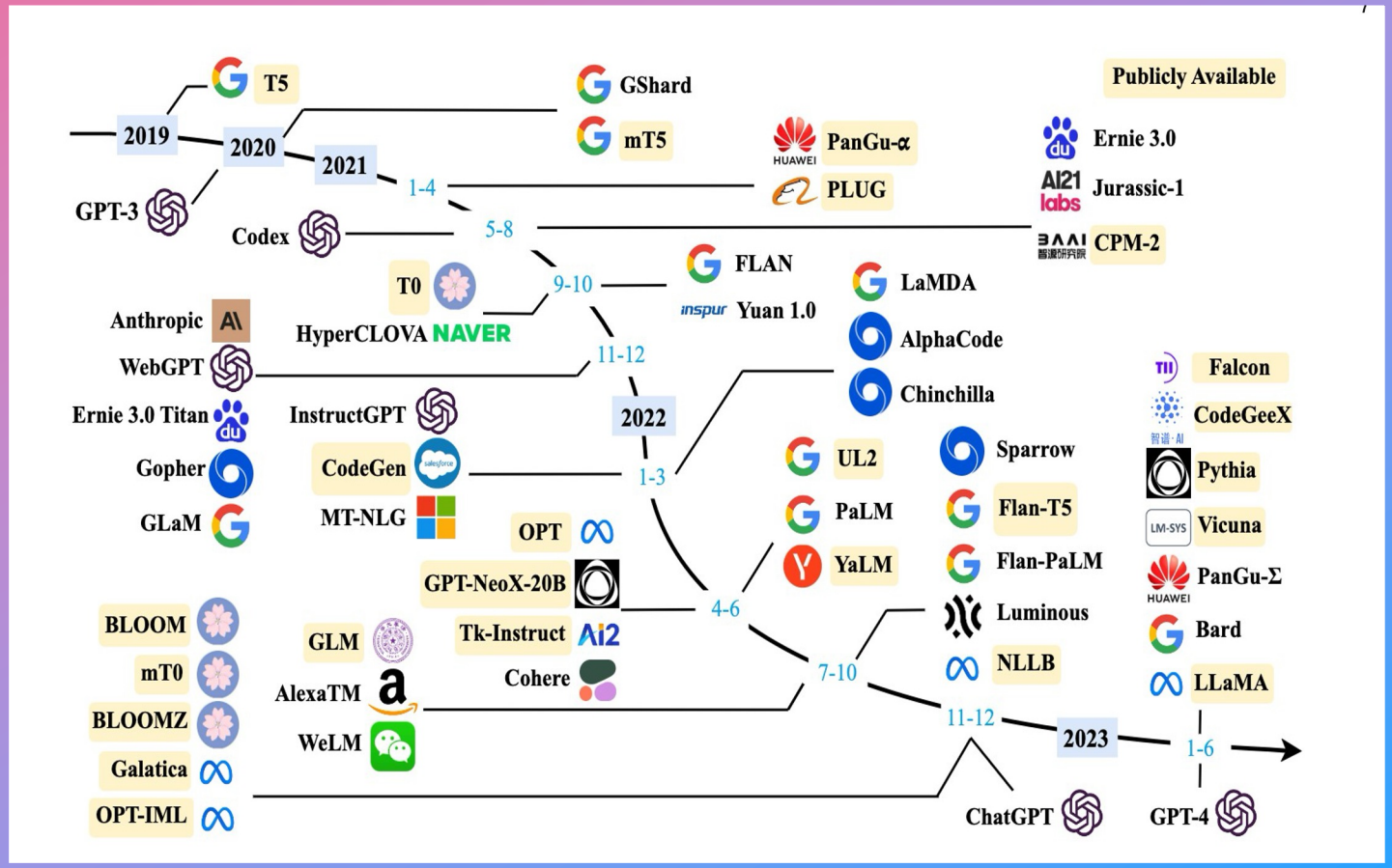
Generative AI Applications Use Cases






Structured data
Text
Voice
3D signals
Images
Large Language Model
Training
Adaptation
Information extraction
Instruction following
Object recognition
Image captioning
Q&A
Sentiment analysis

LLM world is expanding rapidly.



Source : https://arxiv.org/pdf/2303.18223.pdf

GPT?

Pretrained

The underlying model has already been trained on a vast amount of data speeding up time to value

Generative

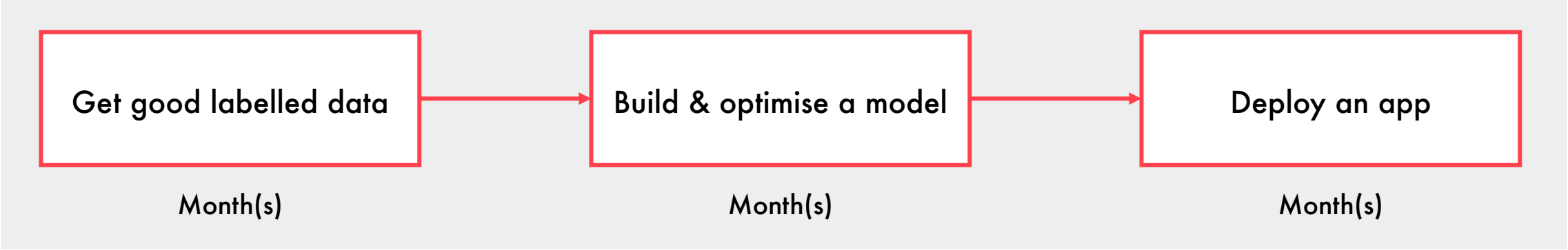
Can create content various forms of new content from language to video to audio

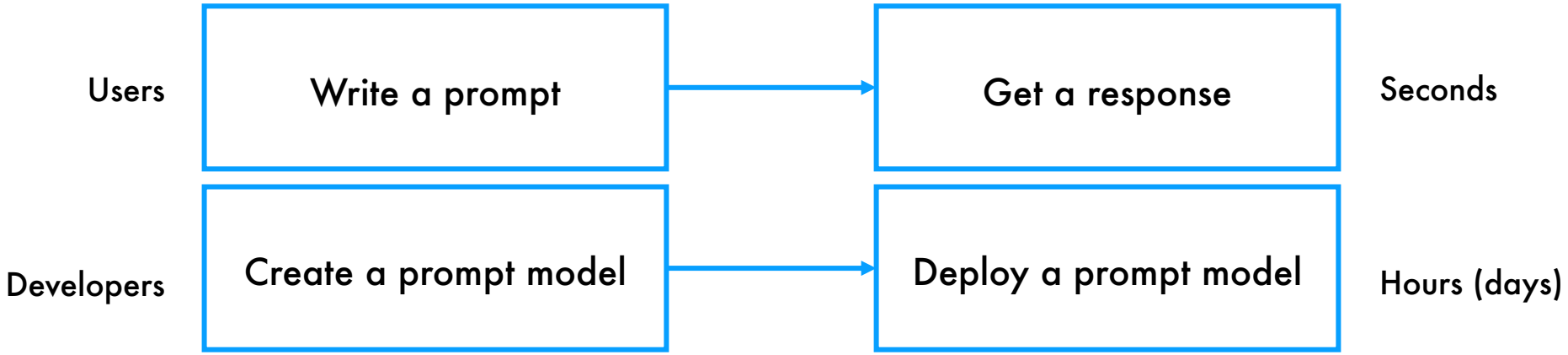
Transformer

The underlying neural network translates your request into numbers to calculate and then back to words probabilistically so you can understand

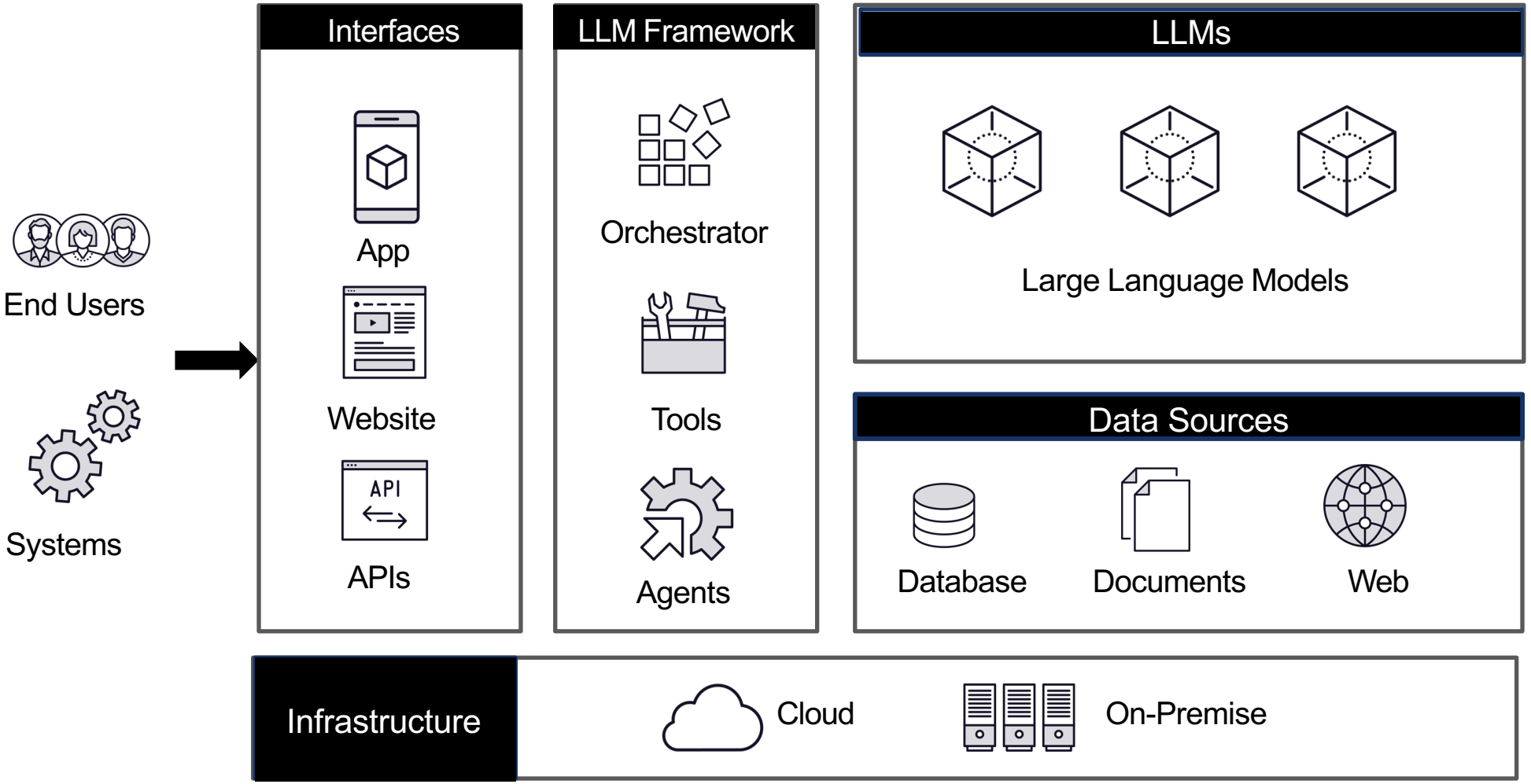
Generative AI is much faster to deploy than traditional AI

Traditional AI / ML / Data Science





Building LLM-Powered Applications



Standardising Tooling: LLM tools

End user access

Web Apps 

Developer access

Notebooks 

Playgrounds 

Model context

Pipelines 

Embeddings 

Vector DBs 

Plugins 

AI-OPS

Caching 

Logging 

Validation 

Source models & Cloud Infrastructure

Licensed APIs 

Open Source APIs 

Cloud Env 

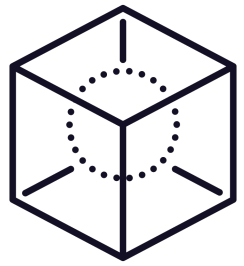
Packaged Cloud 

How can we implement LLM?

- Training foundational model from scratch
- Base commercial models as API
- Open source base model as weights
- Fine-tuning open source base models
- Aligning with human feedback

Base commercial models as API

Vertex.ai



OpenAI

Easy to start

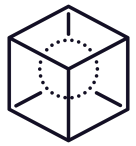
No Infra hassle

Cost could be low based on usage

Works for many case with available customization

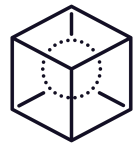
Case for Base commercial models API

Model Types



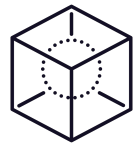
Foundational Model

Predicting Next Word



Chat Based Model

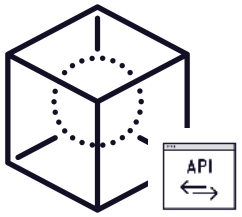
Engage in Conversations



Instruct Tuned Model

Act on instructions

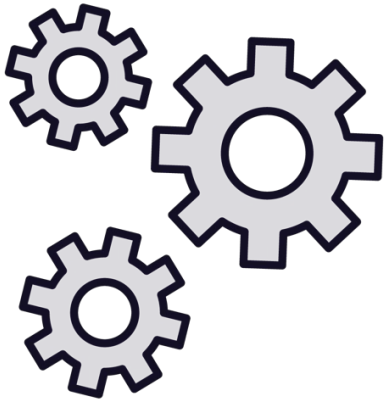
Base commercial models as API



Base Model API

Prompt Engineering

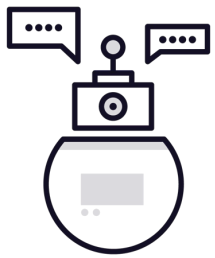
Evaluate



In-Context Learning: Zero-shot, One-Shot, Few-Shot

Inference Parameters: Tokens, Temperature, Sampling

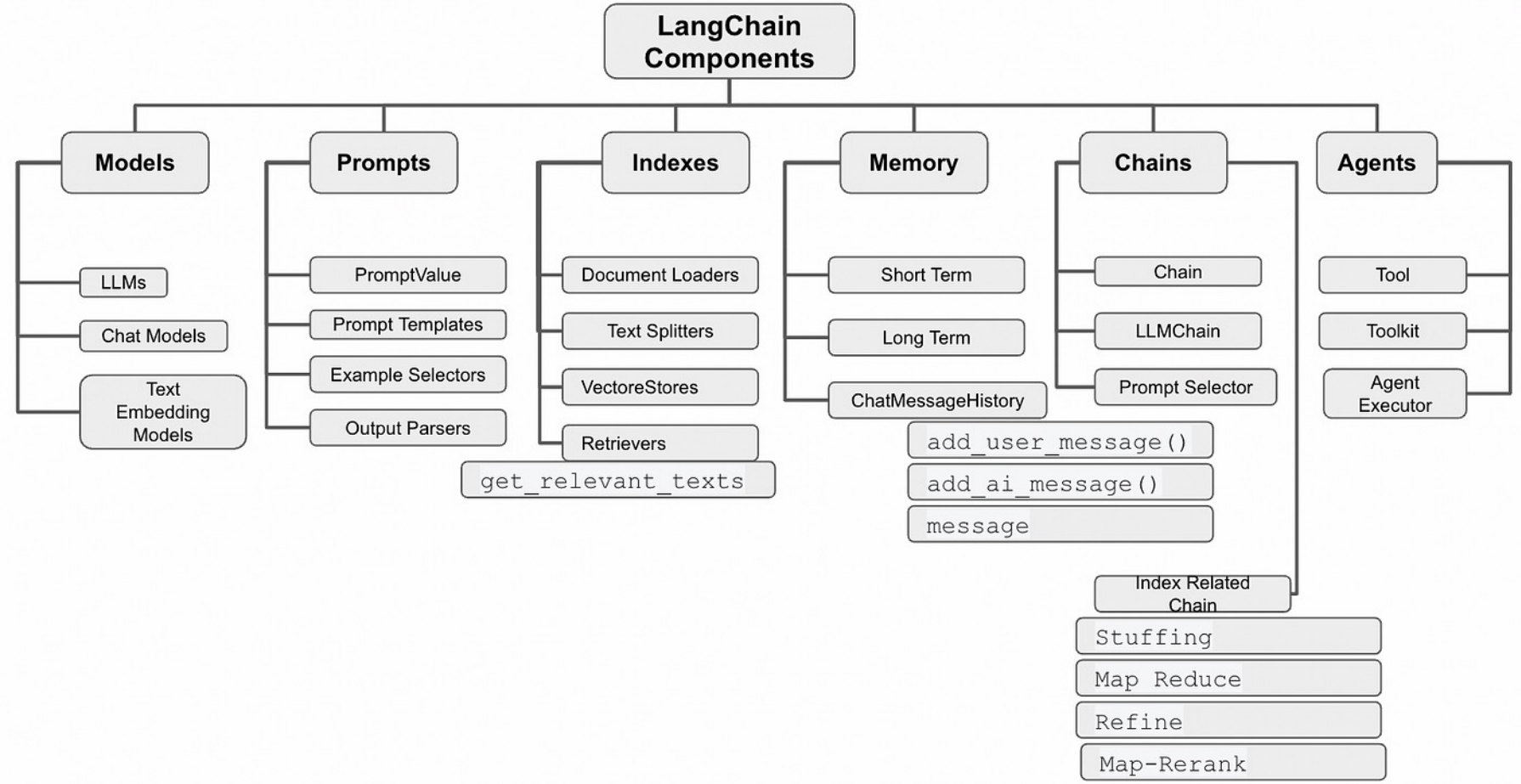
Chains, Tools and Agents : Configurations



App

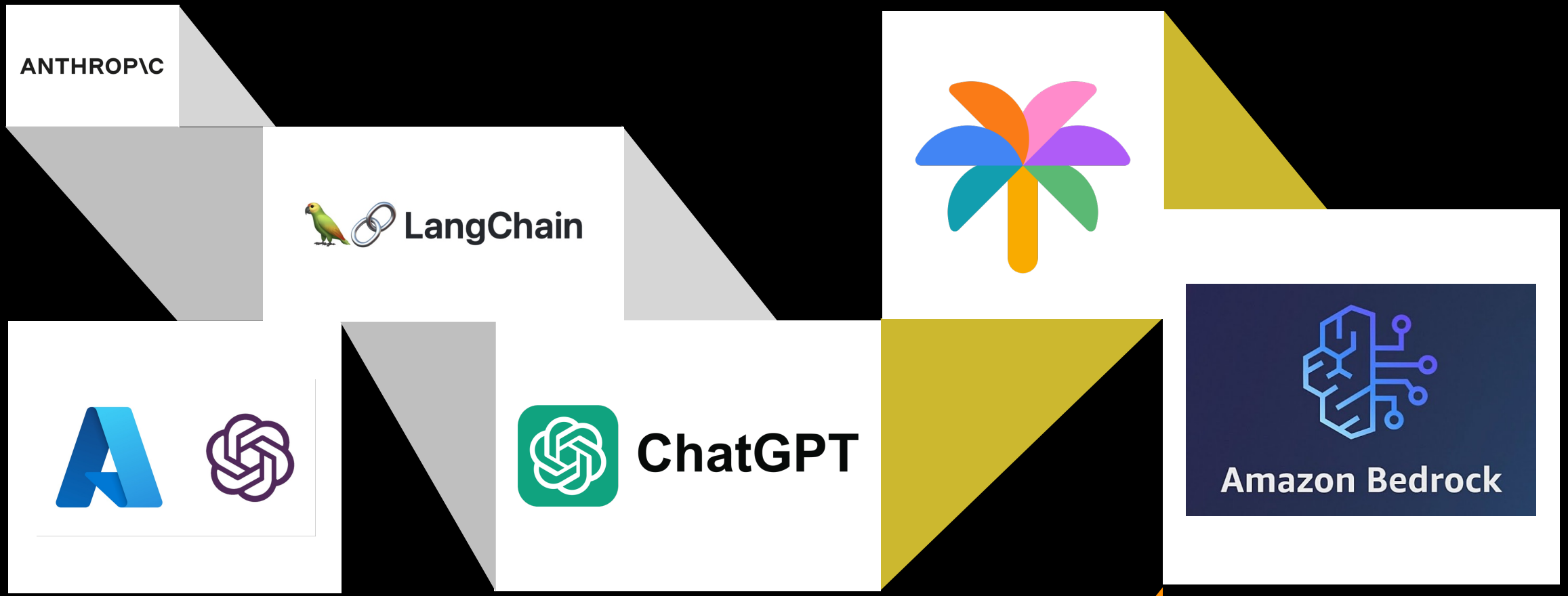
Prompt Management

Results evaluation
Tracing
Latency
Token Length Cost

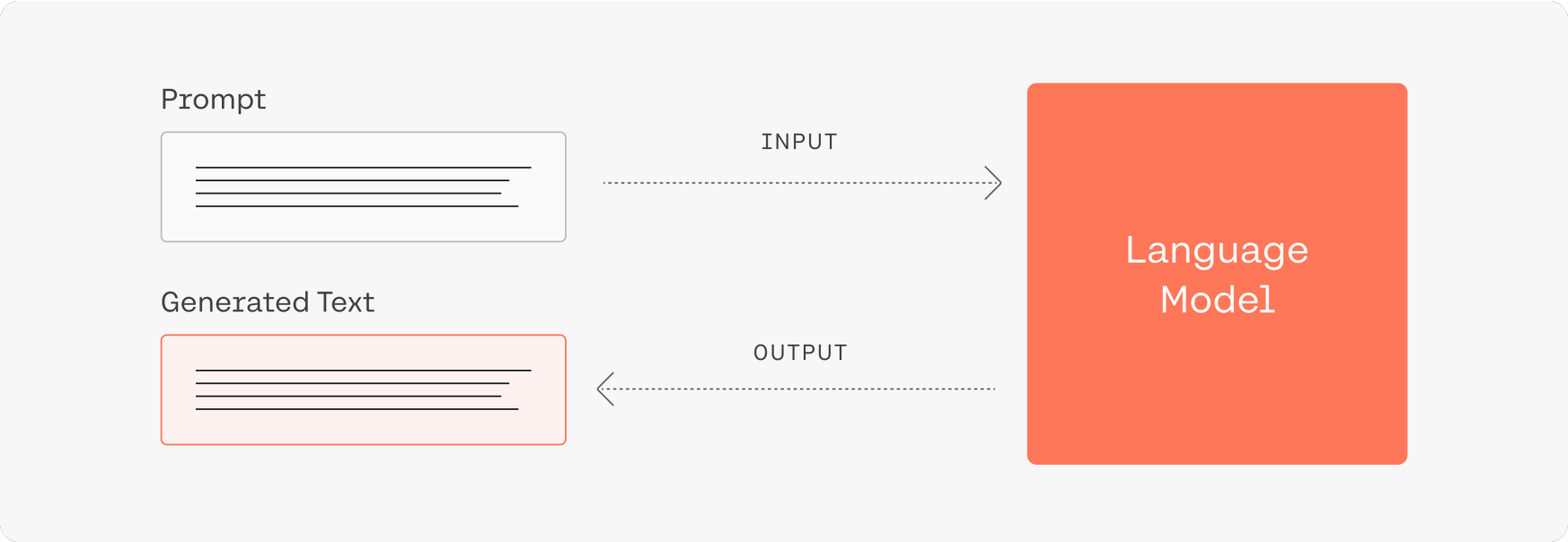


LangChain Components

Provides a sophisticated framework to interact with LLMs, external data sources, prompts, and User Interfaces



LangChain Models



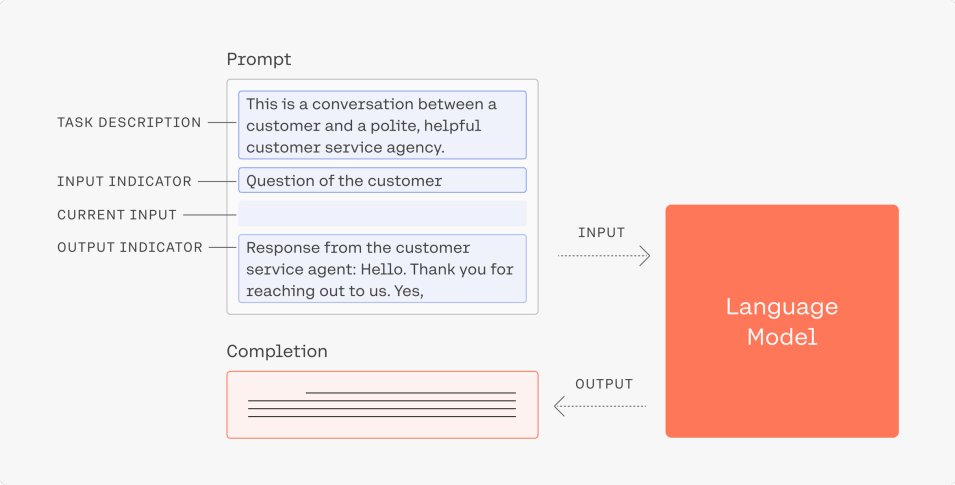
A prompt is a text input given to an LLM, which invokes a response from the model. The art and science of crafting this prompt is called prompt engineering.

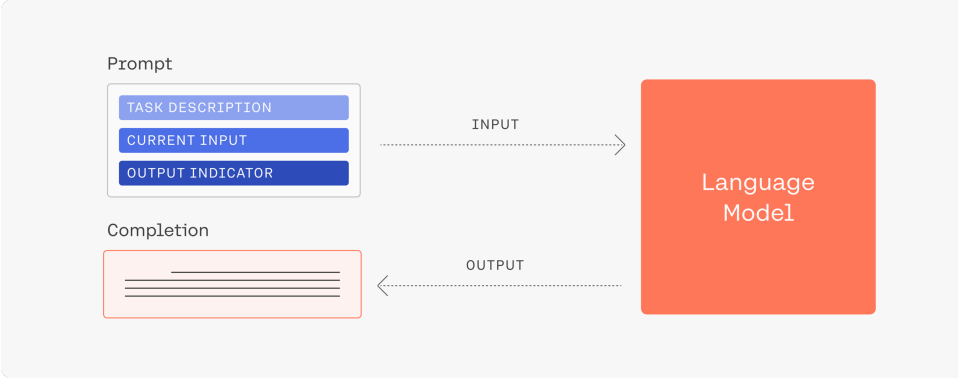
Usage: text summarization, question, and answer, code generation, information extraction, etc.

Data: Text, Code, Image

Prompt Engineering

Prompt Elements

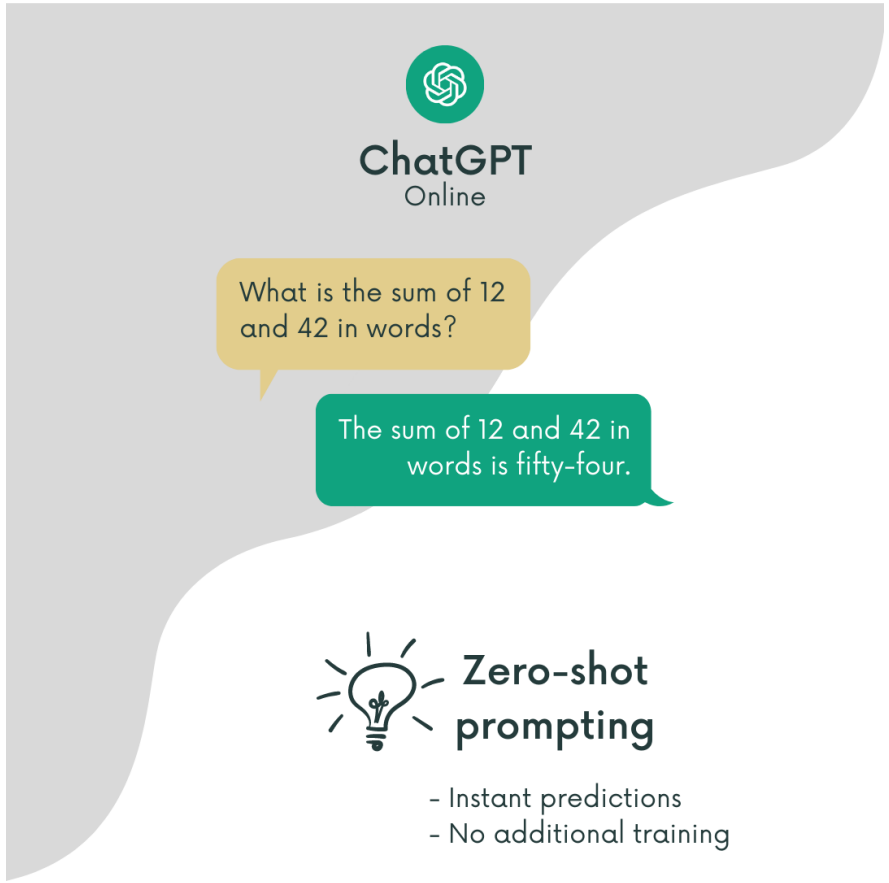




A prompt contains any of the following elements:

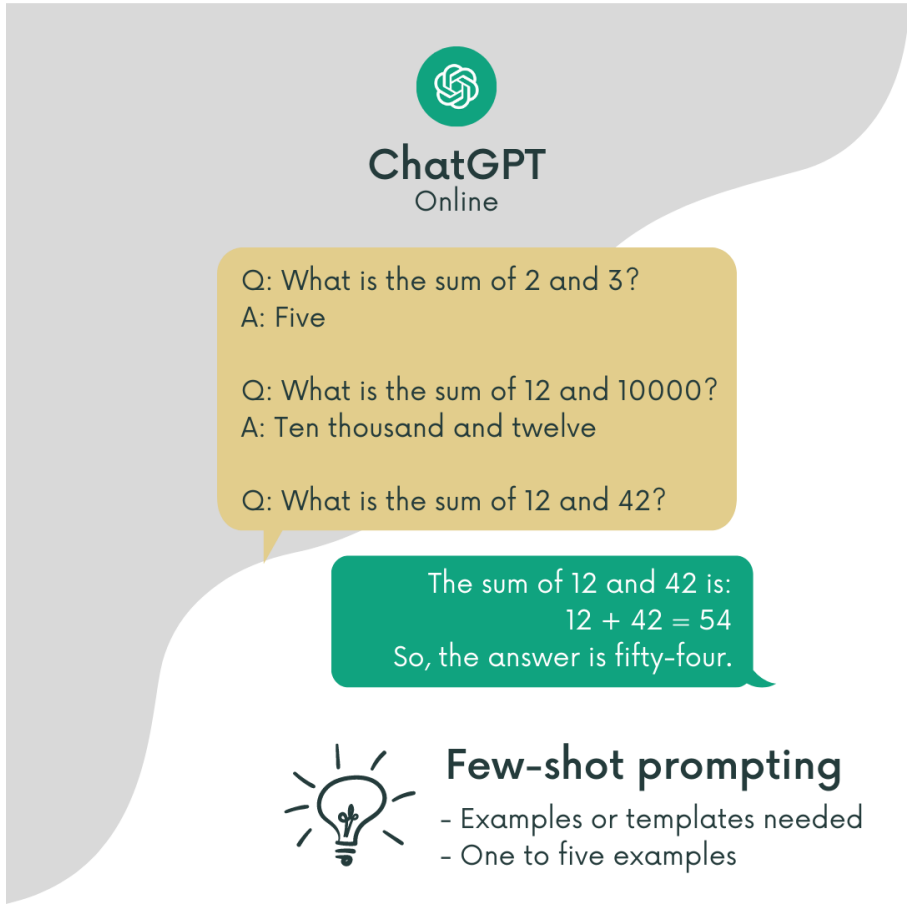
- **Instruction** - a specific task or instruction you want the model to perform
- **Context** - external information or additional context that can steer the model to better responses
- **Input Data** - the input or question that we are interested in finding a response for
- **Output Indicator** - the type or format of the output.

Prompt Techniques - Zero-Shot Prompting



Zero-shot provides a prompt that is not part of the training yet still performing as desired. In a nutshell, LLMs can generalize.

Prompt Techniques - Few-Shot Prompting

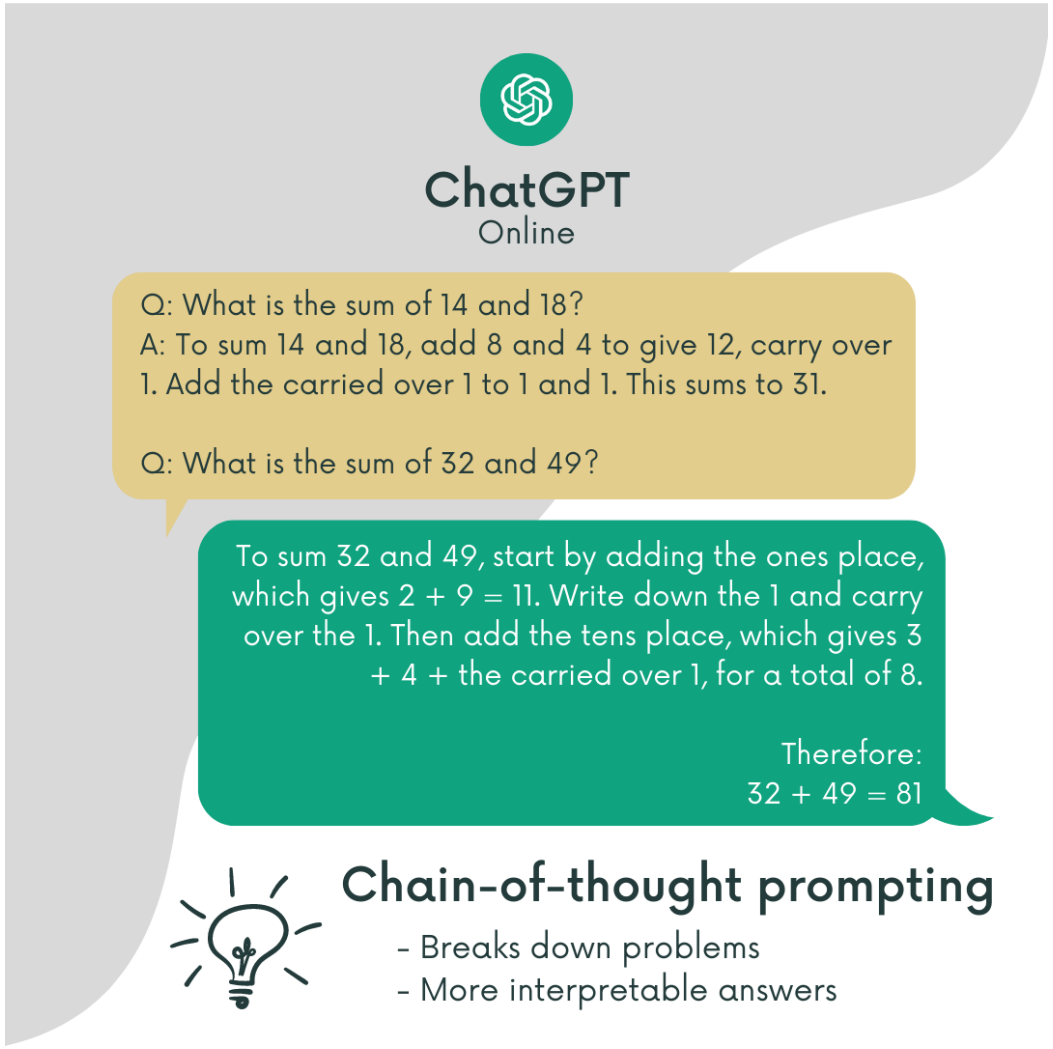


Few-shot prompting can be used as a technique to enable in-context learning where we provide demonstrations in the prompt to steer the model to better performance.

Differences in Prompting Apporach

ZERO-SHOT PROMPTING	BASIS OF DIFFERENCE	FEW-SHOT PROMPTING
Zero-shot prompting is a technique that allows a model to make predictions on unseen data without requiring additional training.	Definition	Few-shot prompting involves guiding the model on a small amount of task-specific data to fine-tune its performance.
No task-specific training data is required. The model relies on its pre-trained knowledge and reasoning abilities.	Training Approach	Requires a limited amount of task-specific training data, usually a few labeled examples.
High flexibility since it can handle a wide range of tasks without additional training.	Flexibility	Moderately flexible, as it requires some task-specific data but can still adapt to different tasks with limited examples.
Limited control over the output as the model relies on its pre-trained knowledge.	Control	More control and customization over the output as the model can be refined based on specific examples or data.
Fast response generation as the model uses its pre-trained knowledge to generate outputs.	Speed and Responsiveness	Slightly slower response generation compared to zero-shot prompting due to the fine-tuning process.
Faster training time as no model optimization is needed.	Training Time	Longer training time compared to zero-shot, but still quicker than full training from scratch.
When specific training data is unavailable or when rapid experimentation is required.	Applicability	When there is a need for task-specific customization or when the available training data is limited.

Prompt Techniques – Chain-of-Thought Prompting

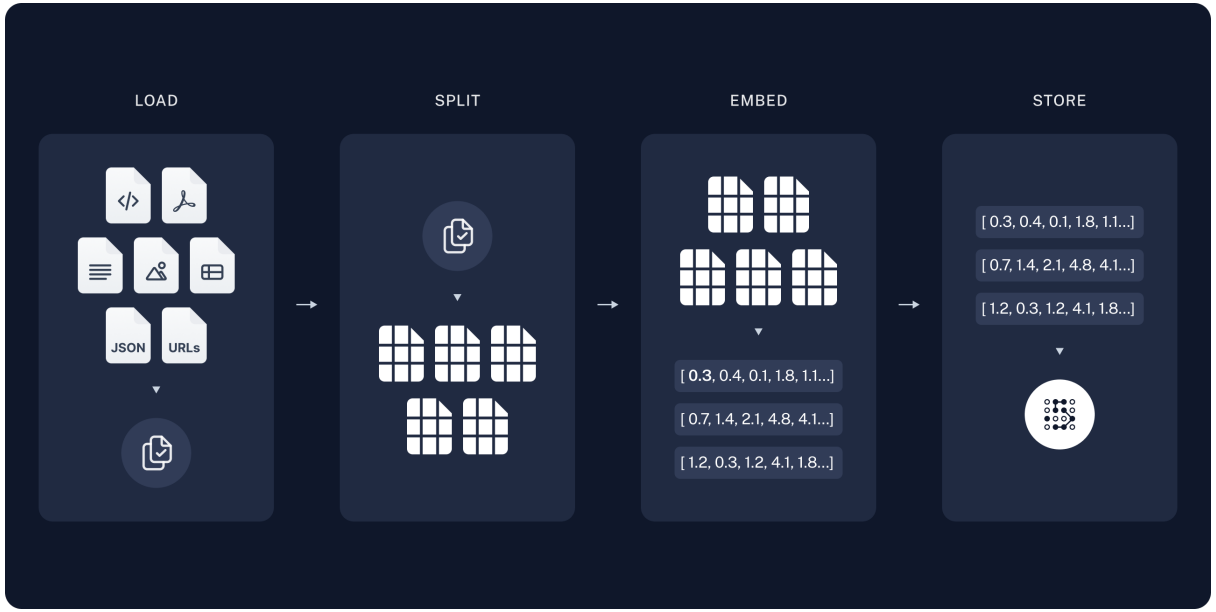


The "chain of thoughts" in prompts for large language models refers to a structured approach to breaking down a complex query into a series of simpler, logical steps. This method helps the LLM process and respond to multifaceted questions more effectively.

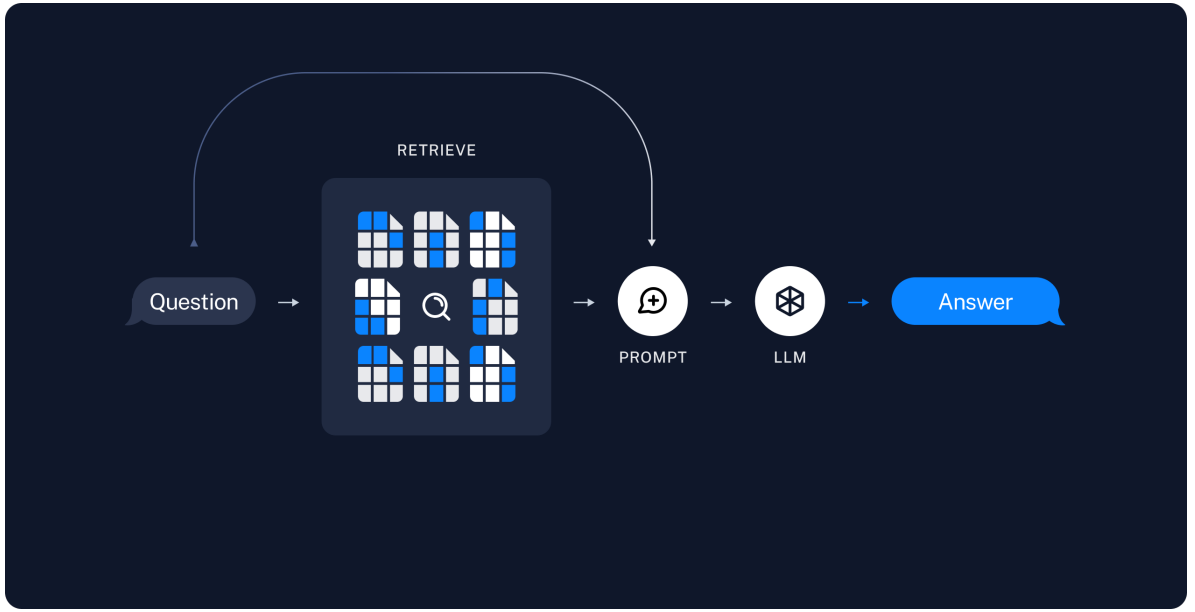
LangChain is a framework that allows you to build applications using LLM

Let's Code

Retrieval-Augmented Generation (RAG)



Indexing: a pipeline for ingesting data from a source and indexing it. *This usually happens offline*.



Retrieval and generation: the actual RAG chain, which takes the user query at run time and retrieves the relevant data from the index, then passes that to the model.

Options to customize LLM

Method	Definition	Primary use case	Data requirements	Advantages	Considerations
 **Prompt engineering**	Crafting specialized prompts to guide LLM behavior	Quick, on-the-fly model guidance	None	Fast, cost-effective, no training required	Less control than fine-tuning
 **Retrieval augmented generation (RAG)**	Combining an LLM with external knowledge retrieval	Dynamic data sets and external knowledge	External knowledge base or database (e.g., vector database)	Dynamically updated context, enhanced accuracy	Increases prompt length and inference computation
 **Fine-tuning**	Adapting a pre-trained LLM to specific data sets or domains	Domain or task specialization	Thousands of domain-specific or instruction examples	Granular control, high specialization	Requires labeled data, computational cost
 **Pre-training**	Training an LLM from scratch	Unique tasks or domain-specific corpora	Large data sets (billions to trillions of tokens)	Maximum control, tailored for specific needs	Extremely resource-intensive

Many of them can be used for commercial purpose ...

Model	License	Commercial use?	Pretraining length [tokens]	Leaderboard score
Falcon-7B	Apache 2.0	✅	1,500B	47.01
MPT-7B	Apache 2.0	✅	1,000B	48.7
Llama-7B	Llama license	❌	1,000B	49.71
Llama-2-7B	Llama 2 license	✅	2,000B	54.32
Llama-33B	Llama license	❌	1,500B	*
Llama-2-13B	Llama 2 license	✅	2,000B	58.67
mpt-30B	Apache 2.0	✅	1,000B	55.7
Falcon-40B	Apache 2.0	✅	1,000B	61.5
Llama-65B	Llama license	❌	1,500B	62.1
Llama-2-70B	Llama 2 license	✅	2,000B	*
Llama-2-70B-chat*	Llama 2 license	✅	2,000B	66.8

Future winners from Generative AI disruption will be those that:

Generative Content

Provide users with generative **AI tools** not more content

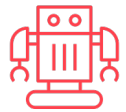
Generative Software

Accelerate development by freeing themselves from servicing technical debt of legacy systems

Generative Interfaces

Leverage their **proprietary data** as the source of competitive advantage

Generative Robots



Build **machine-to-machine** interfaces on the backbone of APIs

$$1.01^{365} = 37.8$$

THANK YOU

publicis
sapient

Feel free to connect over LinkedIn

Ravi Ranjan
@ravi-ranjan-03

Source Code and Presentation

SCAN ME

https://bit.ly/BMSCE-2023