LLMs in 2025 crash-course

Agent(MCP(tools(rag(llm)))))) ?????

Who am I?

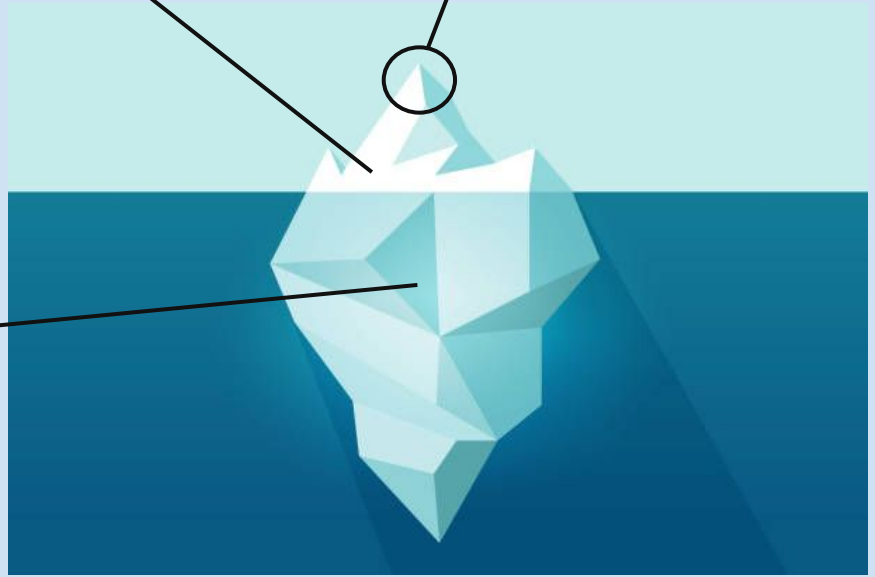
- Jiaqi Chen
- 24 yr
- BSc AI, University Utrecht
- Start-ups / Entrepreneurship
  - Own startup
  - incubator board year
  - DialogueTrainer Product Manager
- Bouldering, live music, photography

Preface

Techichal

Societal,
mental/cognitive,
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational

This presentation



"Crash course in LLMs"

Techichal

Societal,
Mental/cognitive,
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational

Costs

Costs

This presentation

Effects

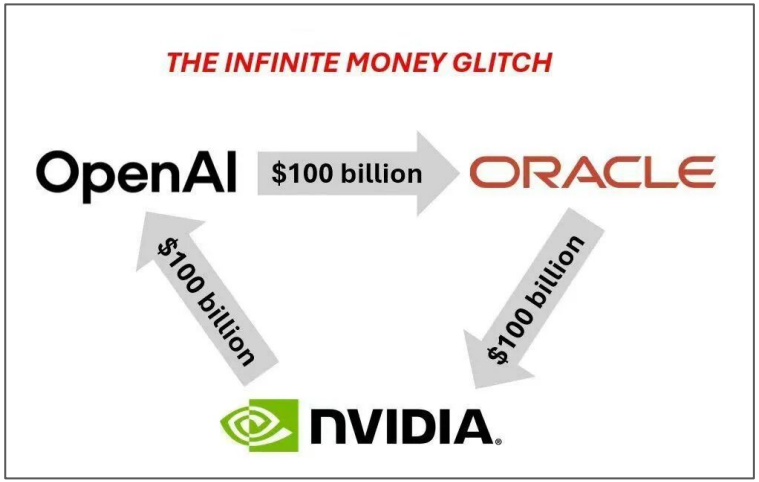
Effects

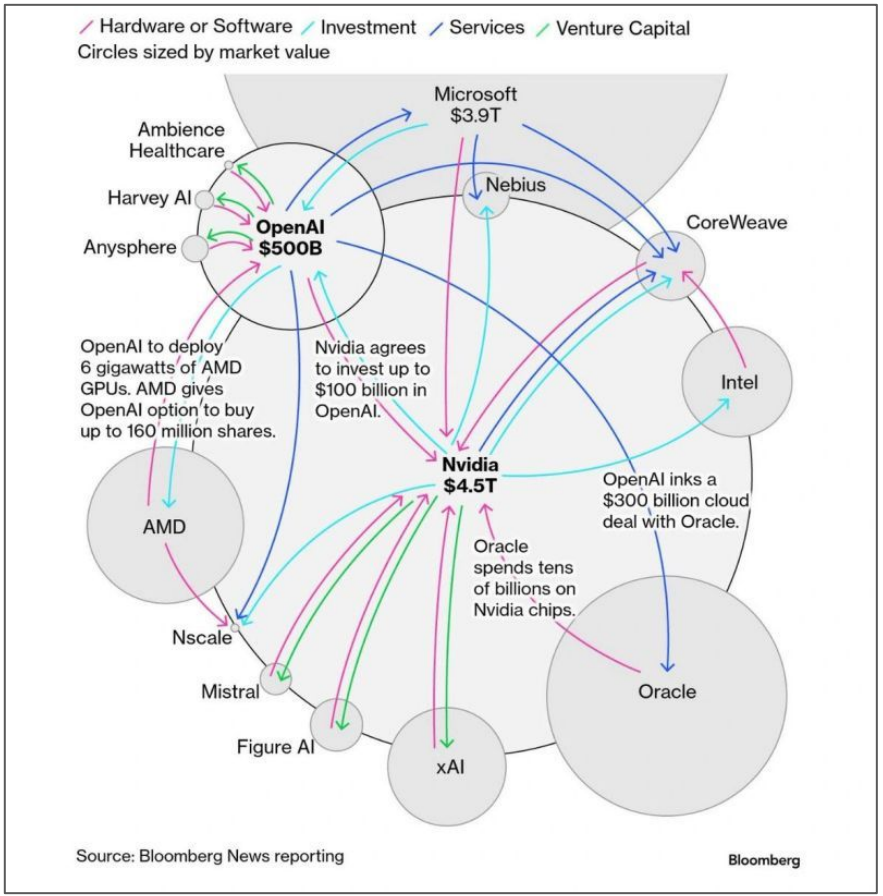
"Crash course in LLMs"


Techichal
Societal,
Mental / Cognitive
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational
This presentation
“Crash Course in LLMs”
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs
Costs Effects Costs Effects Costs

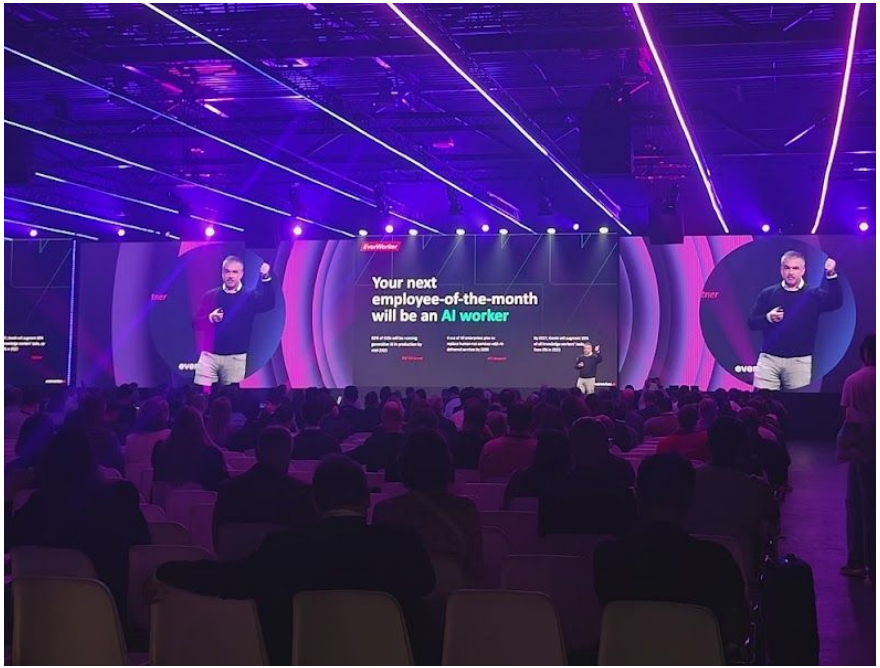
Financial ‘dimension’

The investment space of tech in 2025





Human centric AI



AI Overview

Tantalum is used primarily in electronics as a component for capacitors and resistors in devices like phones and computers. It is also used in high-temperature alloys for jet engines and nuclear reactors, and due to its corrosion resistance and biocompatibility, it is ideal for medical implants and surgical devices.

AI Overview

Cobalt is used primarily in lithium-ion batteries for electronics and electric vehicles, in high-strength, heat-resistant alloys for jet engines and turbines, and in pigments to create vibrant blue and green colors. It is also used in magnetic materials, as a catalyst in the petroleum industry, and the radioactive isotope cobalt-60 is used for cancer treatment and as a gamma-ray source.

Congo is one of the world's main sources of cobalt and other critical minerals that power batteries, devices, and AI hardware.

It should have been one of the most prosperous regions on earth, yet it has remained one of the hardest places to be born for over a century.

Resource extraction has repeatedly led to conflict, exploitation, and systemic instability. It is a tragic illustration of how technological progress, global markets, and human suffering can become entangled when cost is measured only in money or performance.

Arthur C. Clarke's Third Law:

> Any sufficiently advanced technology is indistinguishable from magic

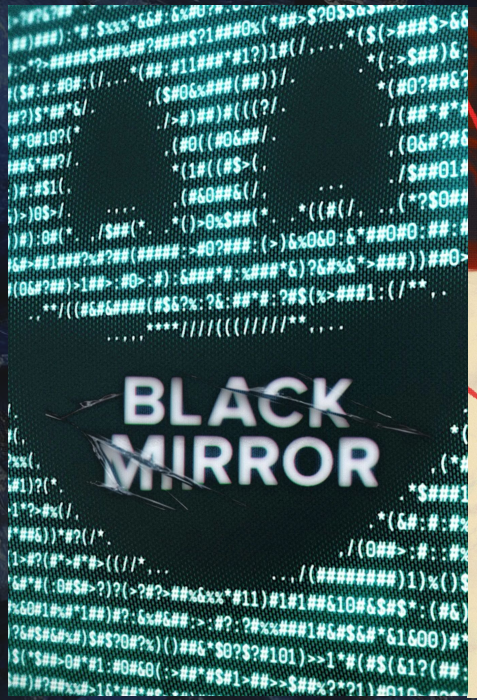
Techichal

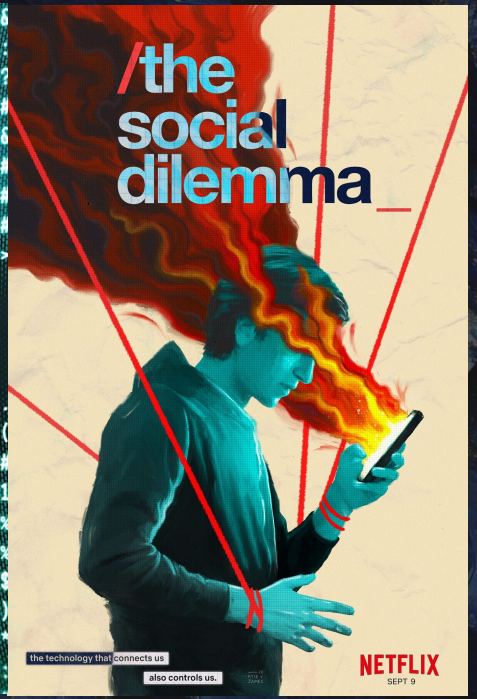
Societal,
mental/cognitive,
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational

Arthur C. Clarke's Third Law:

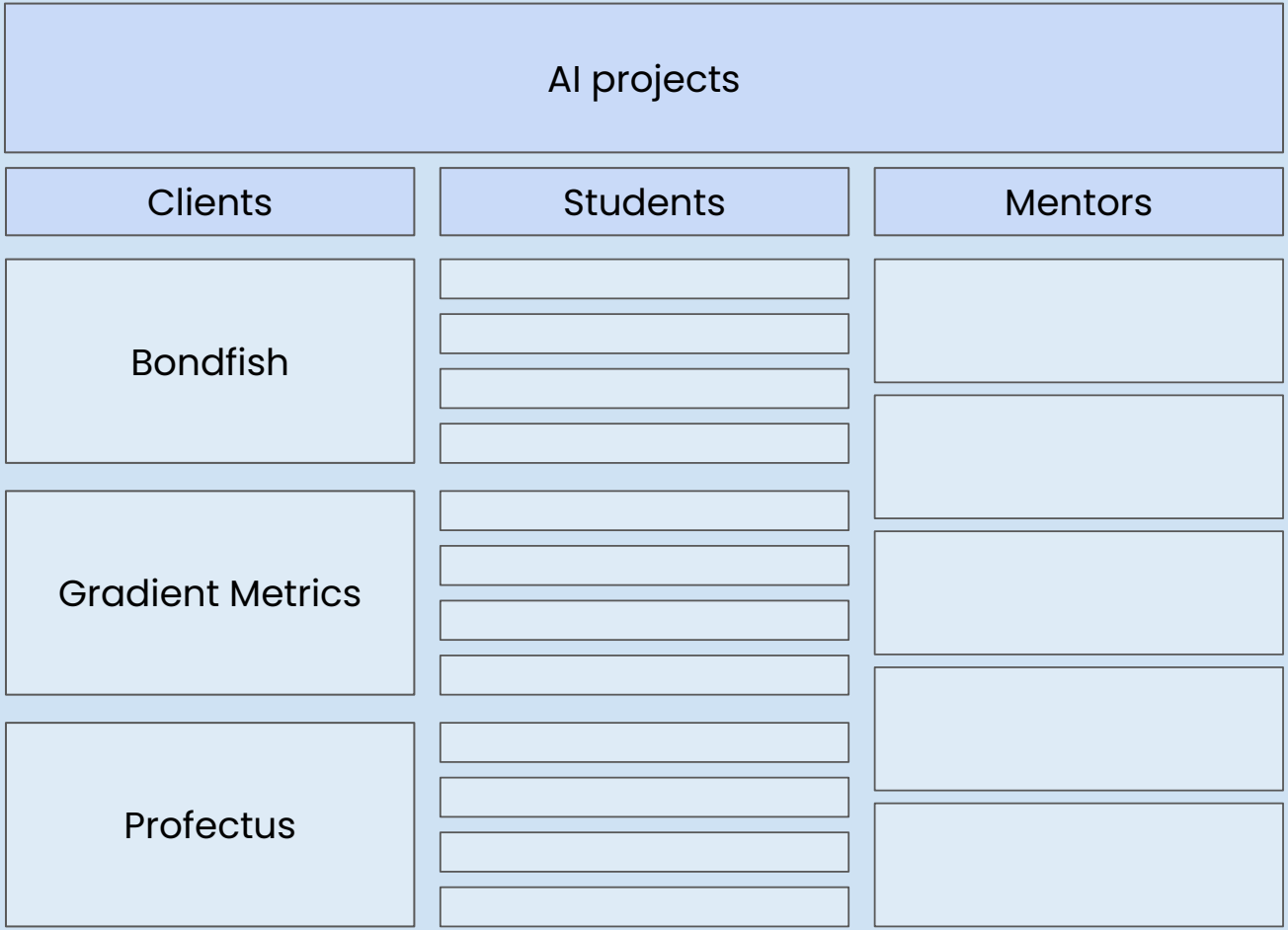
Any sufficiently advanced technology is indistinguishable from magic

Magic





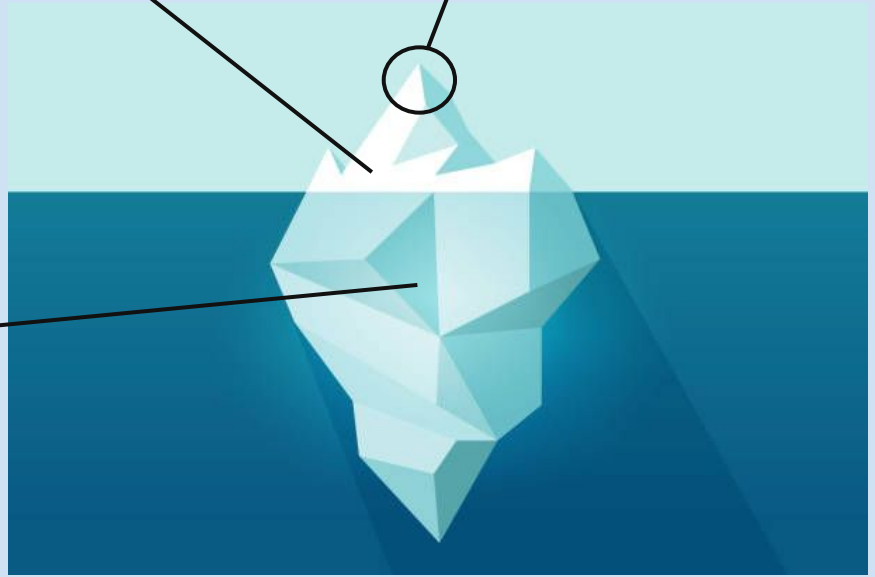
AI Projects


AI projects
Clients
Students
Mentors
Bondfish
Gradient Metrics
Profectus

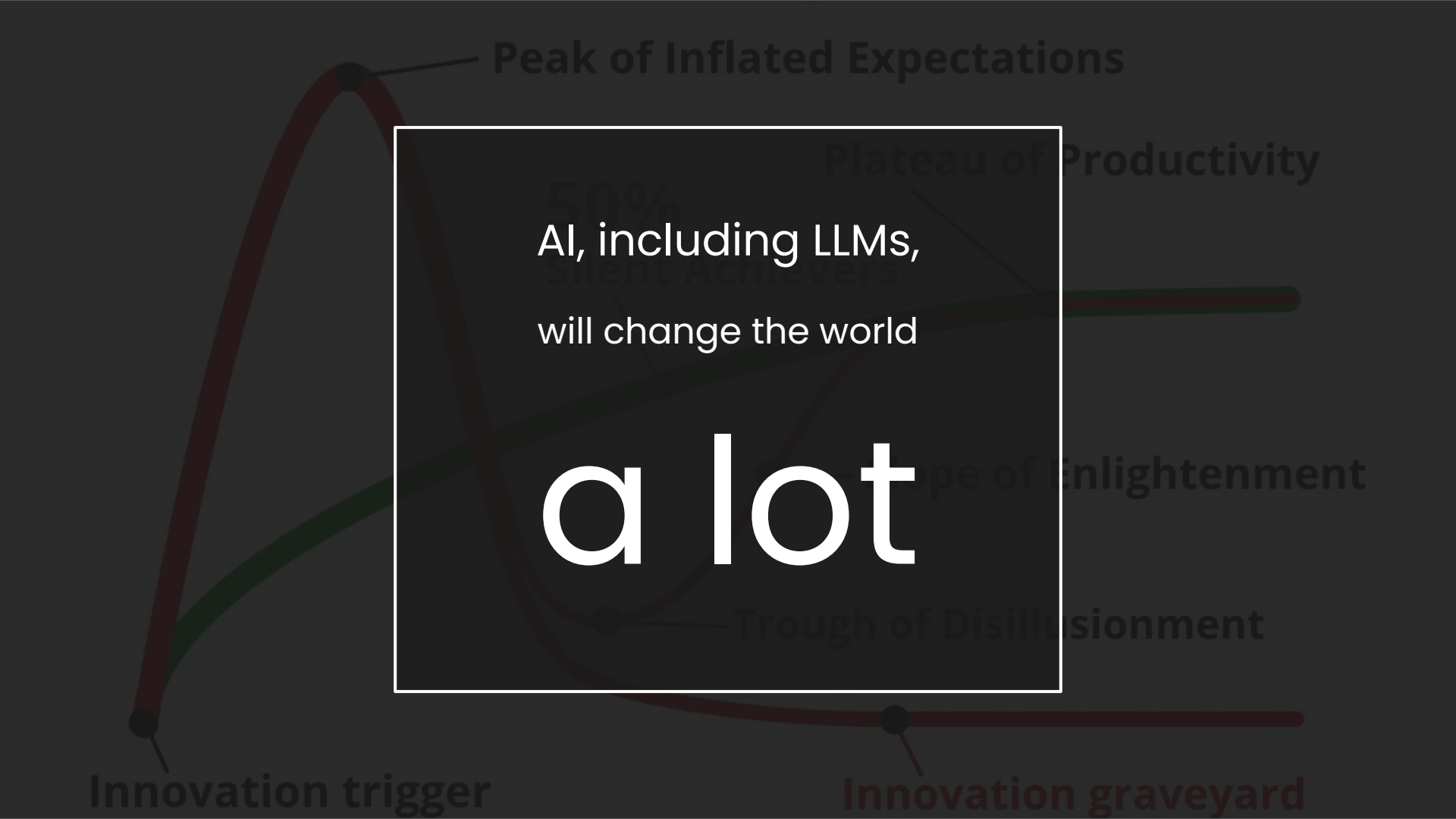
Techichal

Societal,
mental/cognitive,
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational

This presentation



"Crash course in LLMs"

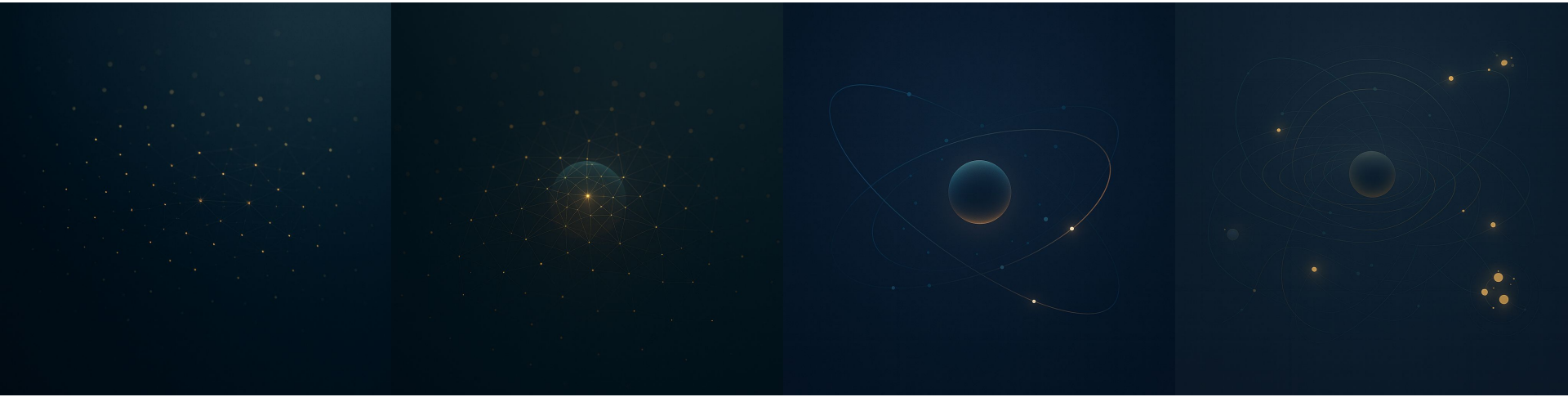


AI, including LLMs,

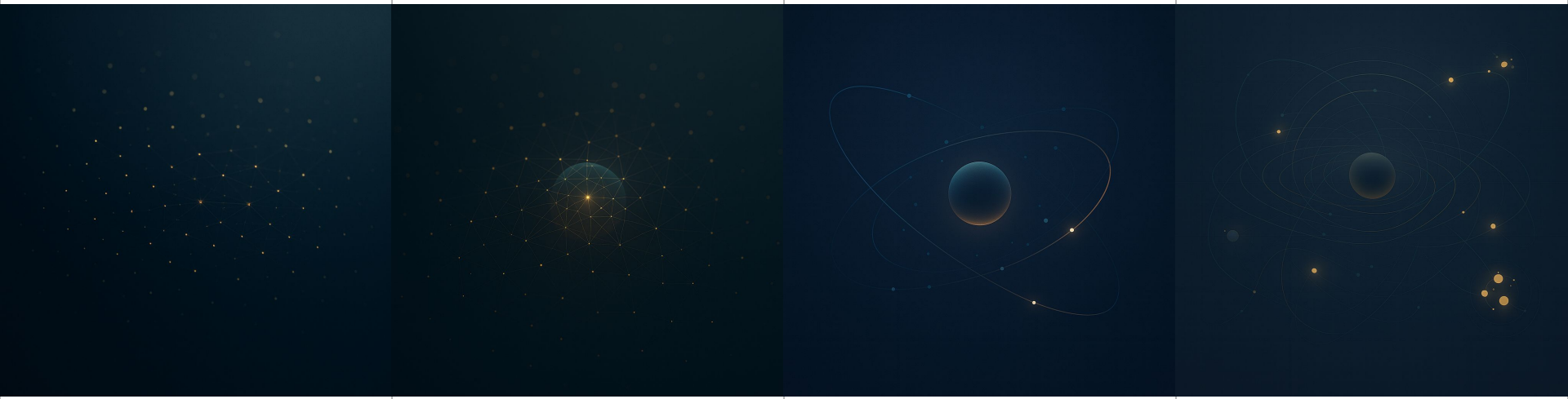
will change the world

a lot

Pre-LLM LLM RAG Agentic



Terms don’t strictly follow timeline



Symbolic AI

Attention mechanism

Transformer Architecture

System Prompting

Autonomous Workflow

Hybrid Retrieval

Vector Databases & Embeddings

Feature Engineering

Knowledge Distillatioin

LLM to SQL

Goal Decomposition

Multi-Agent systems

Semantic search

Word Embeddings

Tool Use
& Function Calling

Instruction-Tuning

Meta-Prompting

MoE

Tokenization

Chunking & Indexing

MCP

Context Window

Sequence Models (RNN/LSTM)

RLHF

Context Compression / Engineering

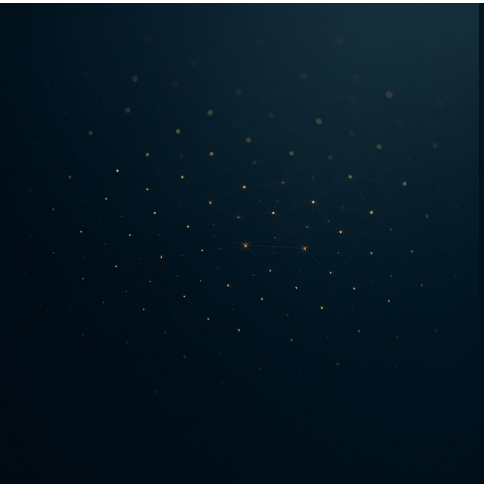
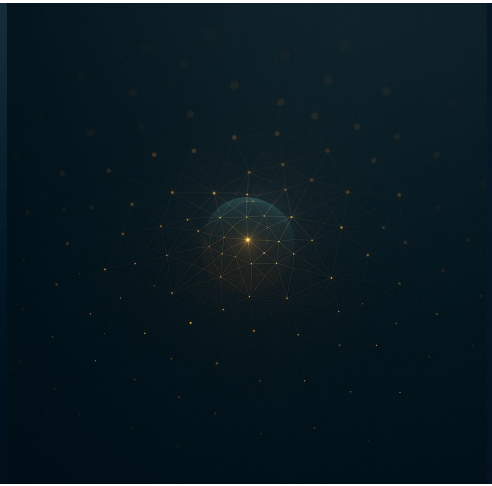
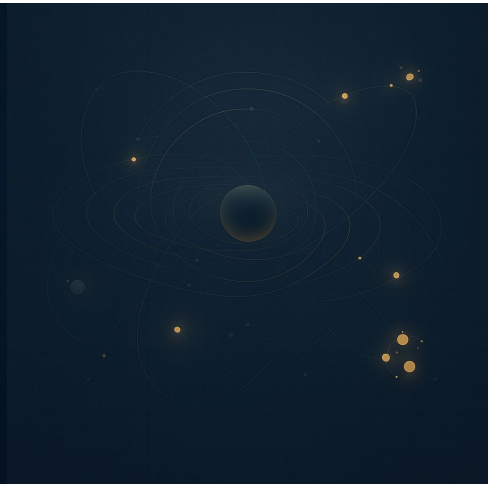
Prompt engineering

NLP (Natural Language Processing)

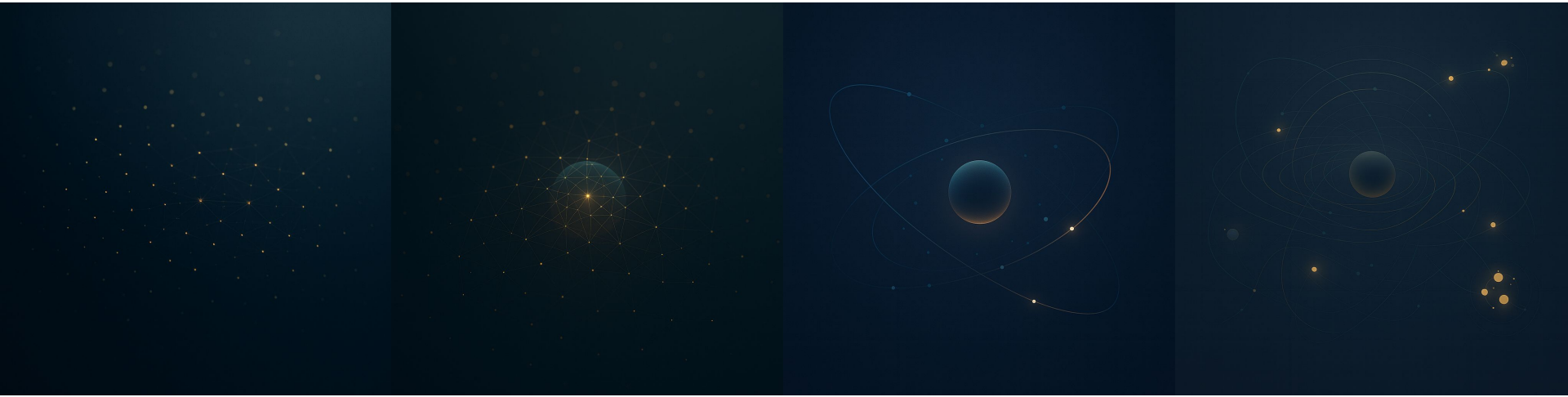
AGI

Human-centered AI

Human-Computer Interaction (HCI)

Pre-LLM	LLM	RAG	Agentic
			
Data pipelines, Sklearn pipelines Embeddings Tokenization	LLM/SLM LLM pipelines	Basic RAG (Hybrid) indexing	LLM interactions overview MCP

Pre-LLM | LLM | RAG | Agentic



Data-pipelines
Model-pipelines

Vectors &
embeddings

→

LLMs

(as a
data-pipeline)

→

RAG

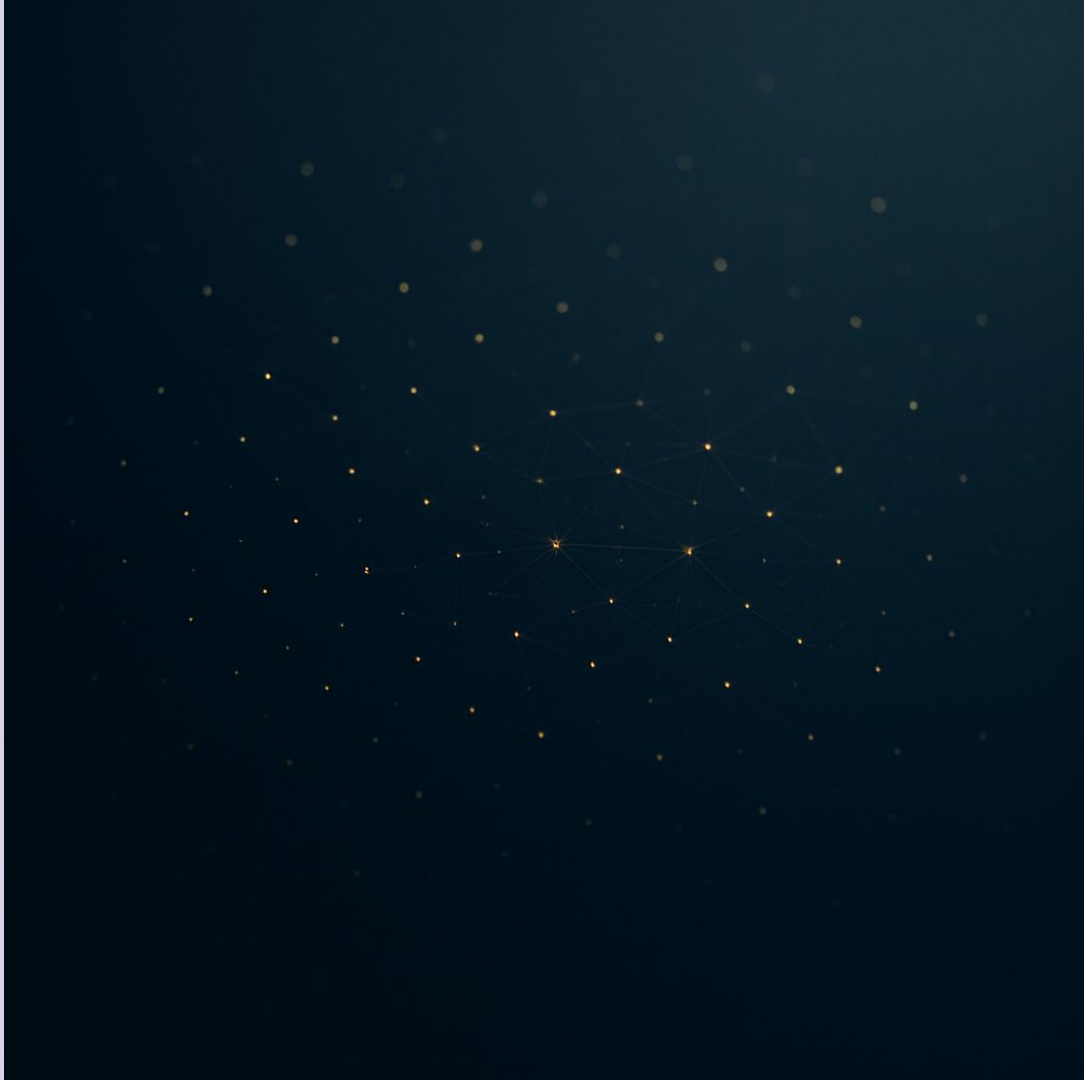
(as a
LLM-pipeline)

→

Agent

(as an LLM with
reasoning and
tools)

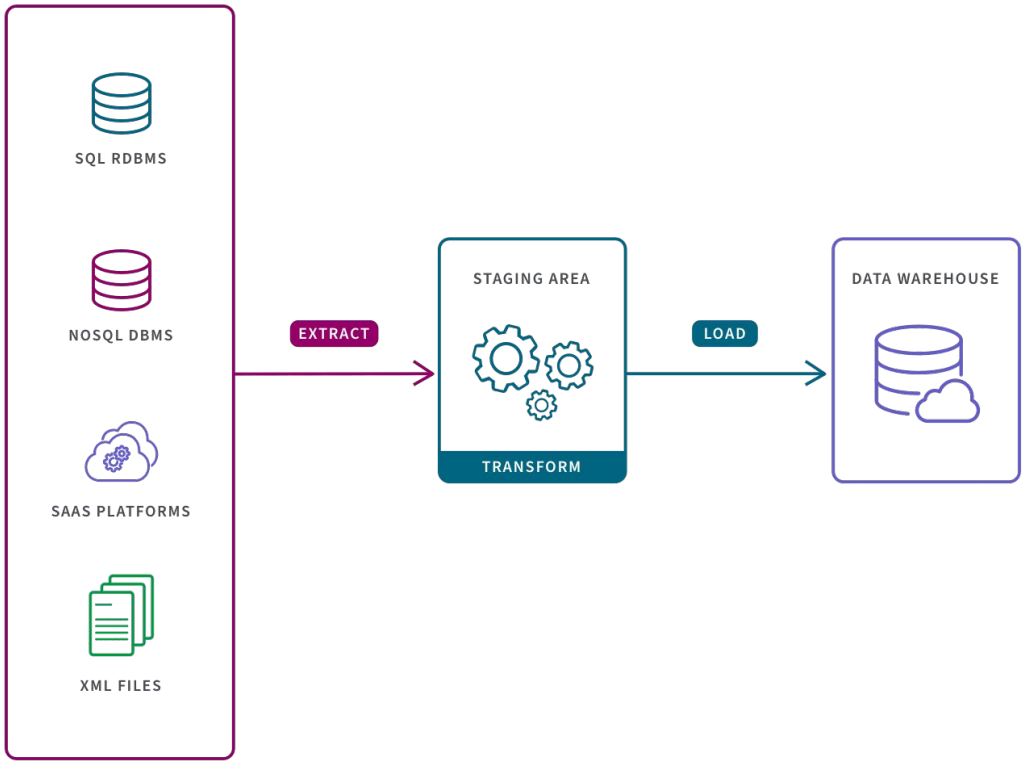
Pre LLM



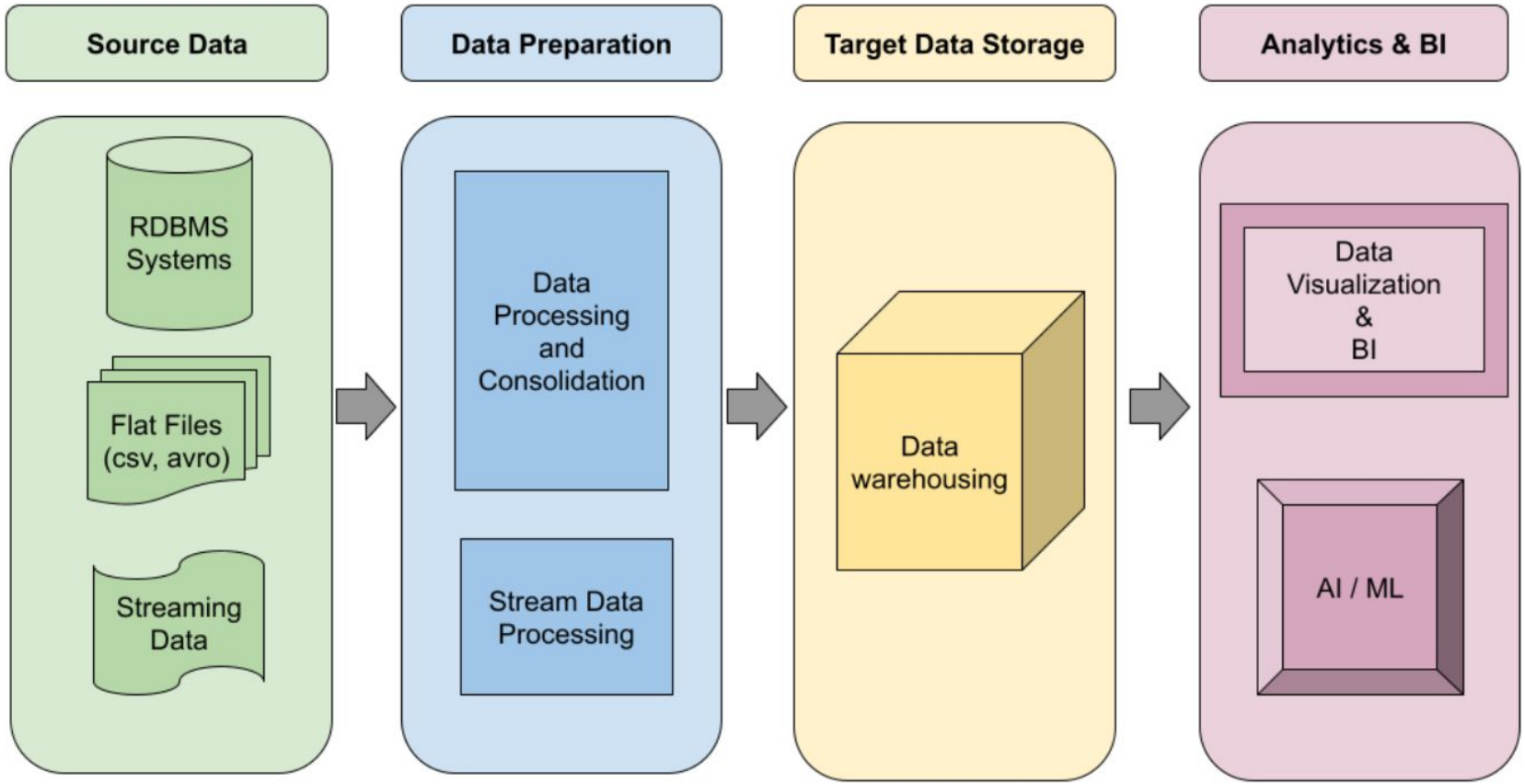
Data Pipelines

Level 1: input-output

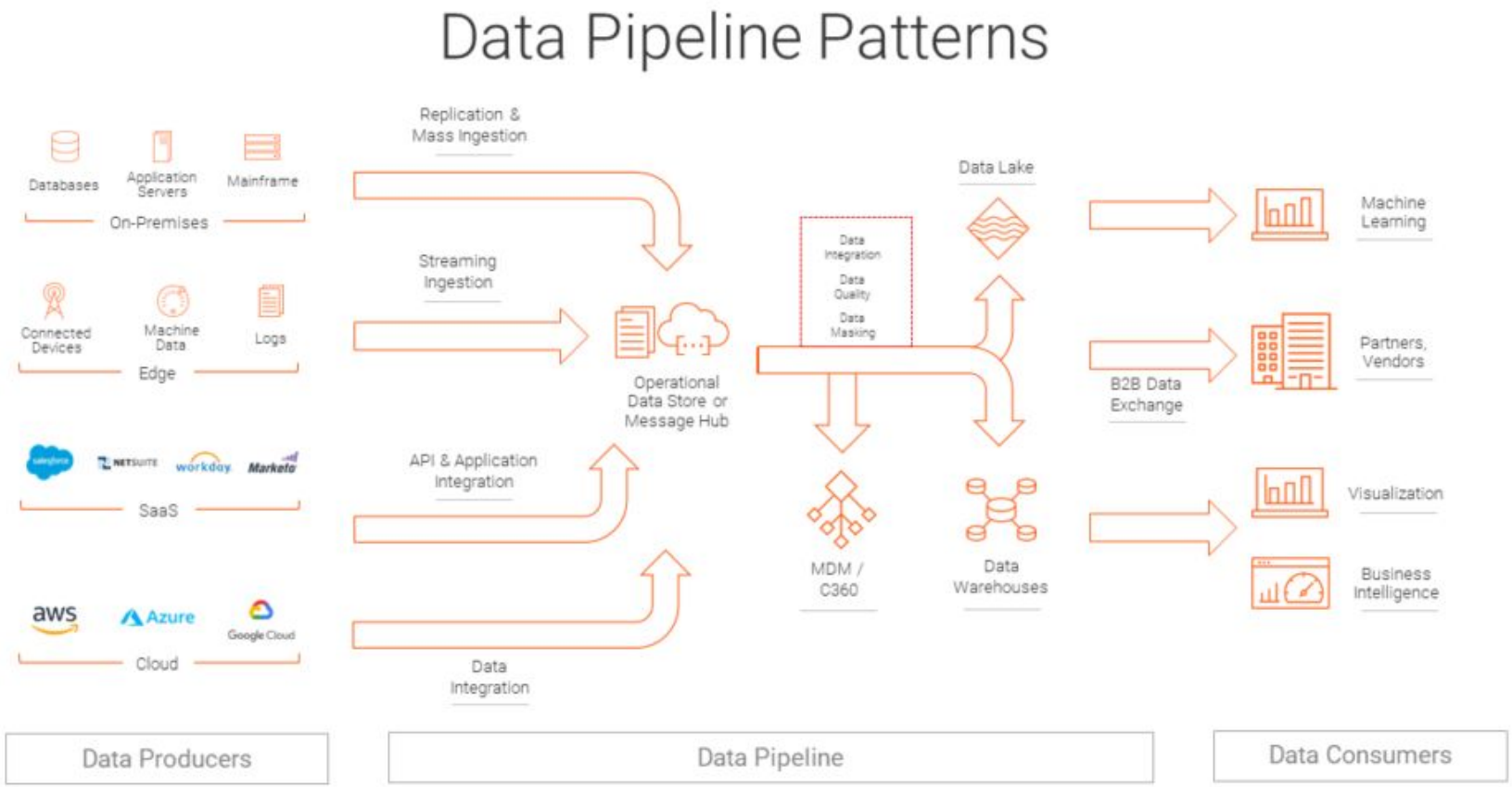
ETL (Extract, Transform, Load)



Level 2: ETL



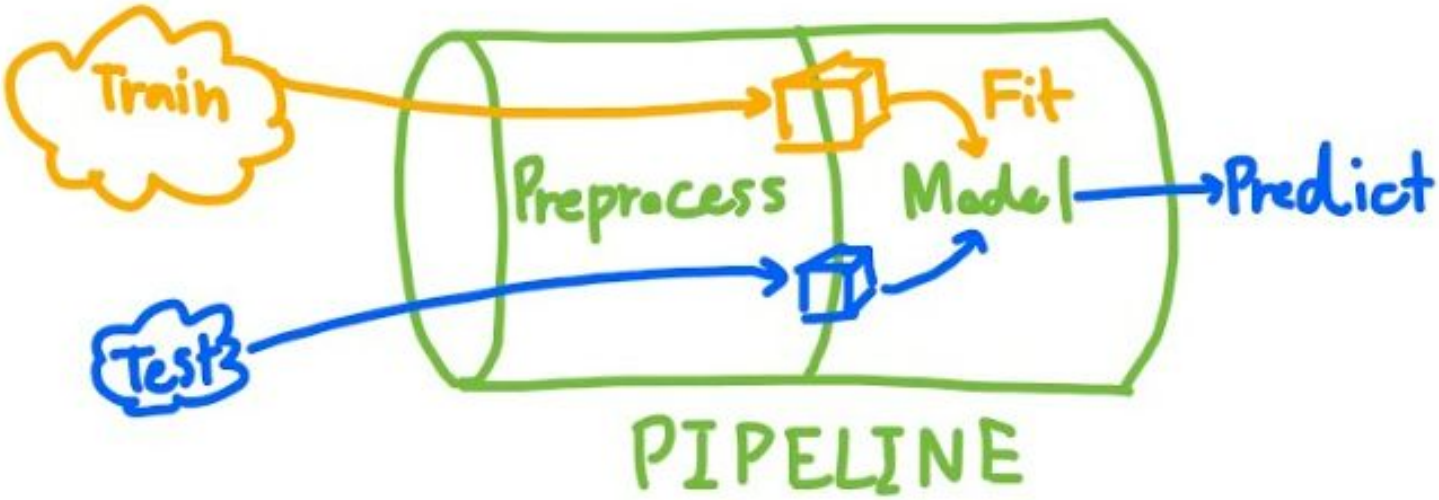
Level 3: high level end to end



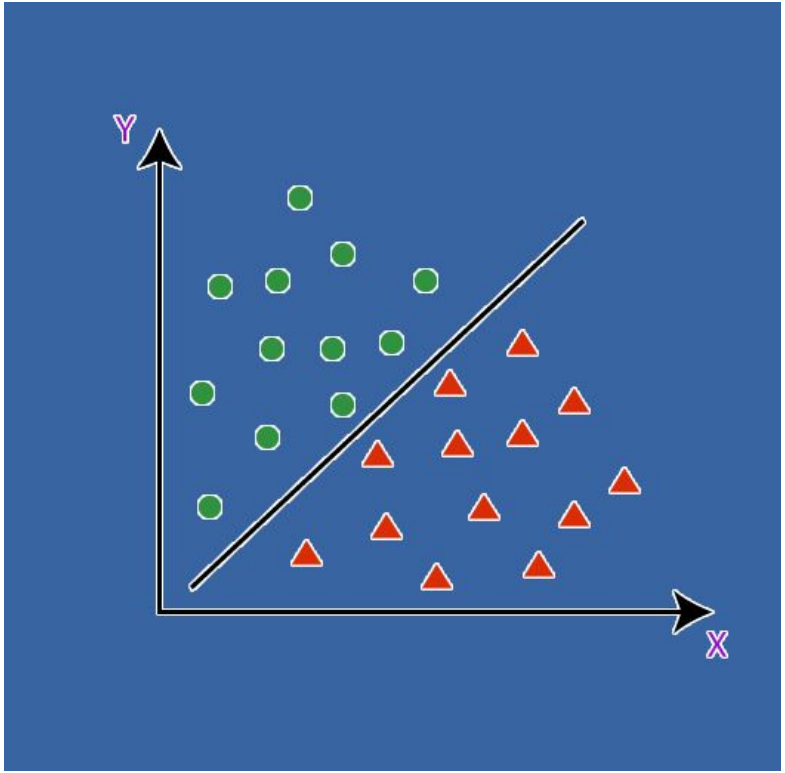
Level 4: integrations and more choices

Pre-llm models

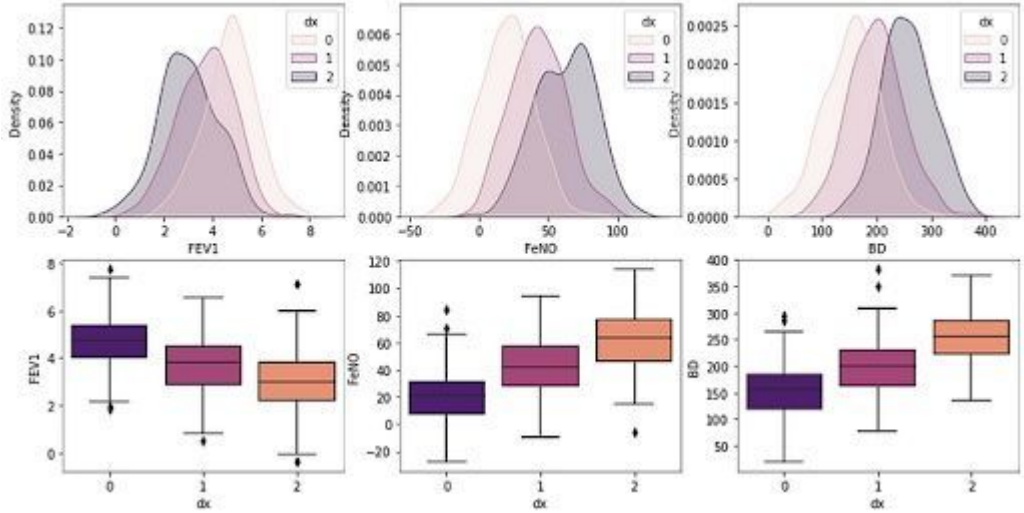
(AI pre-covid)


Train
Fit
Preprocess
Model
Predict
Test
PIPELINE

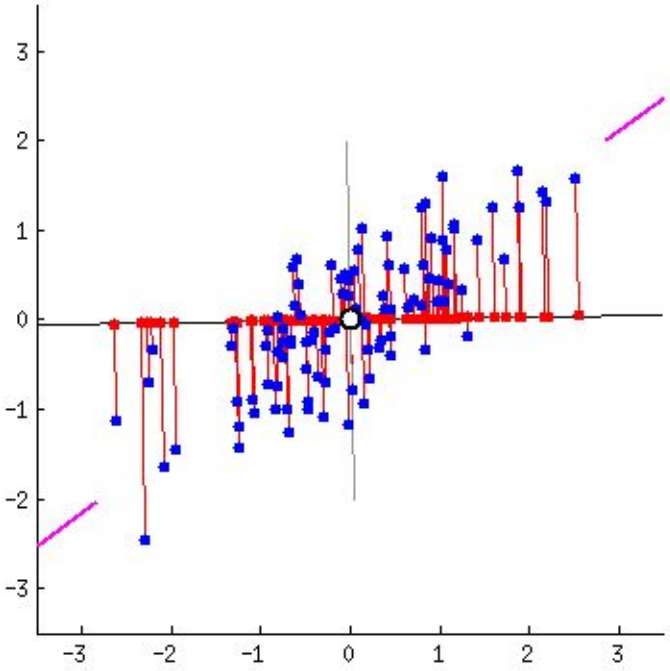
SVM



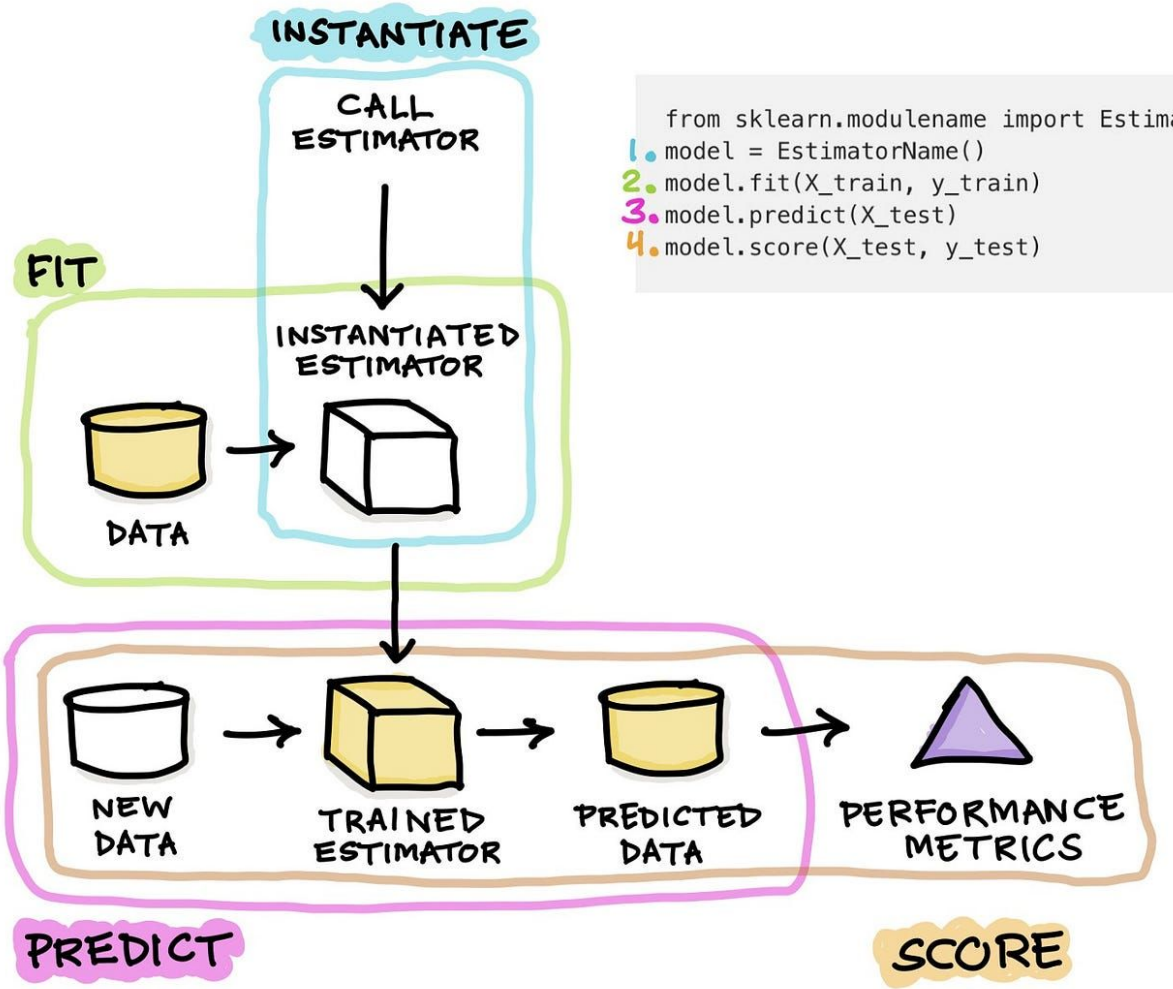
Bayes



PCA

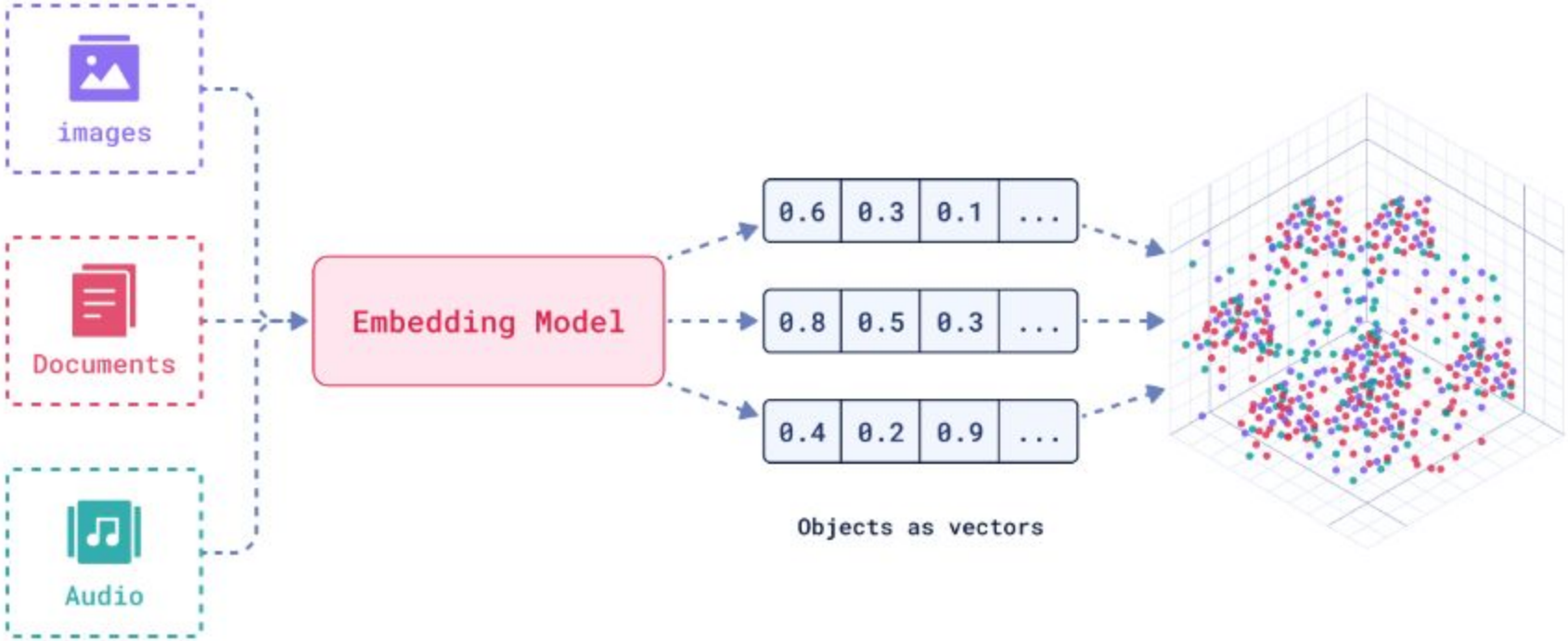


Sk-Learn

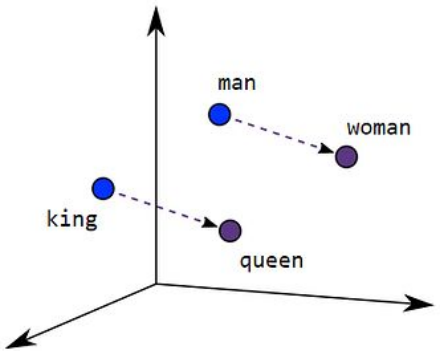



```
from sklearn.modulename import EstimatorName
model = EstimatorName()
model.fit(X_train, y_train)
model.predict(X_test)
model.score(X_test, y_test)
```

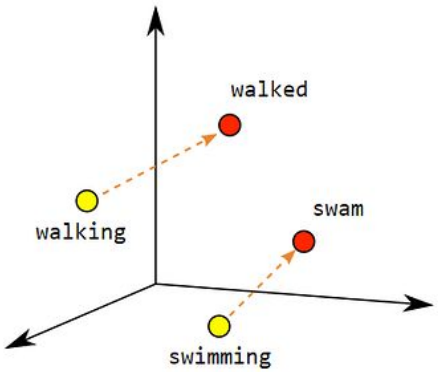
Embeddings


images
Documents
Audio
Embedding Model
0.6 0.3 0.1 ...
0.8 0.5 0.3 ...
0.4 0.2 0.9 ...
Objects as vectors

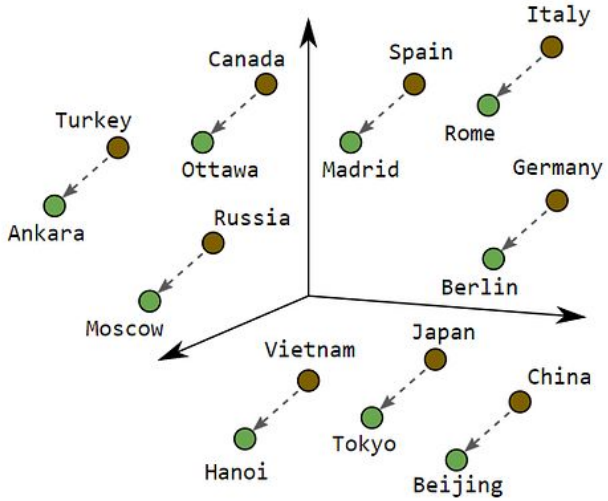
Embeddings can capture relations between the vectors (in a latent space)



Male-Female

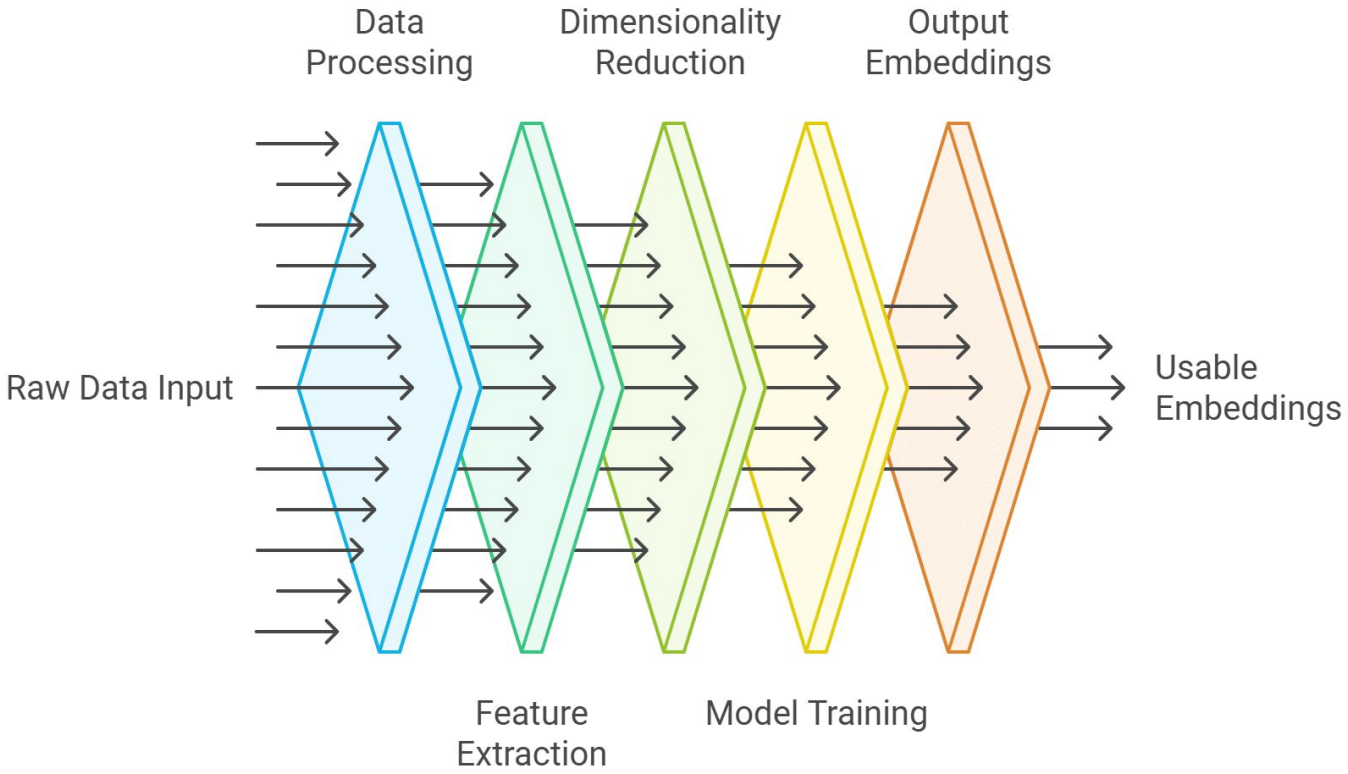


Verb Tense



Country-Capital

Embedding Model Process



Tokenization

Tiktokenizer

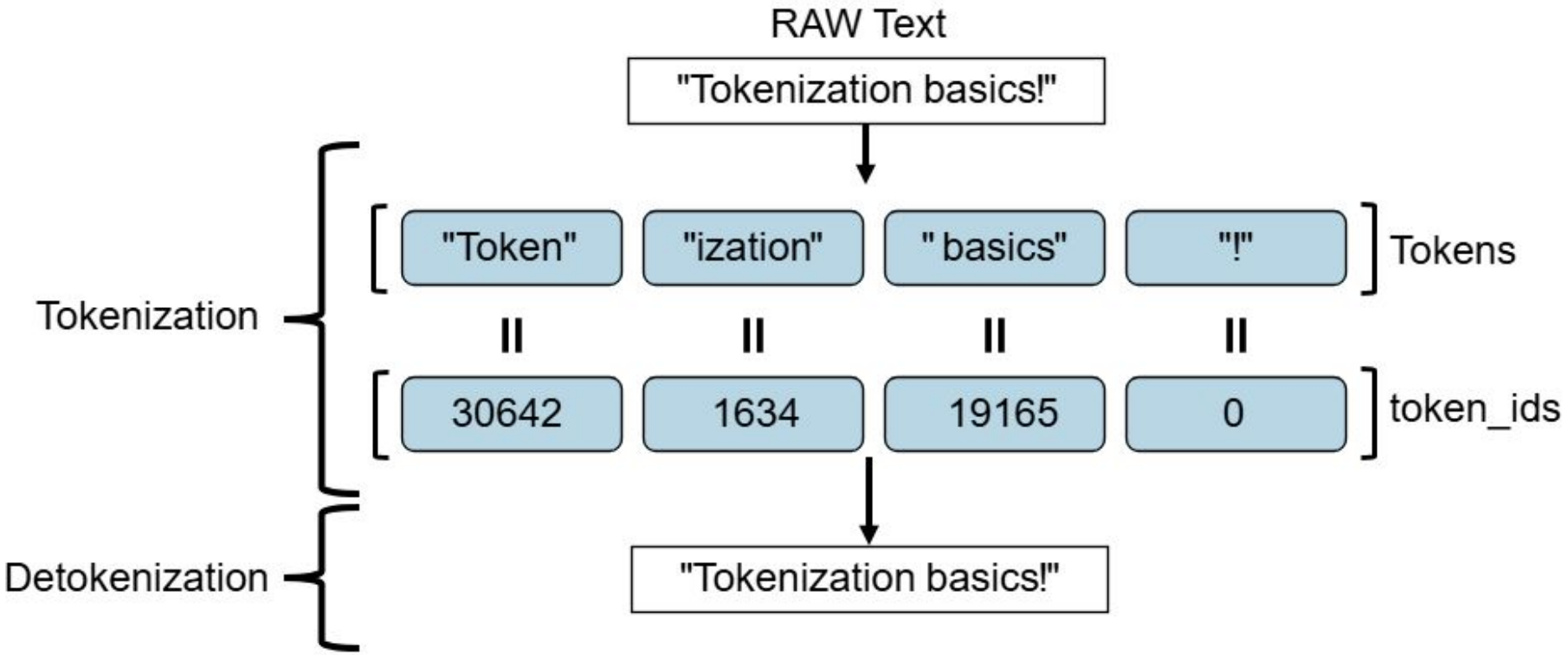
cl100k_base

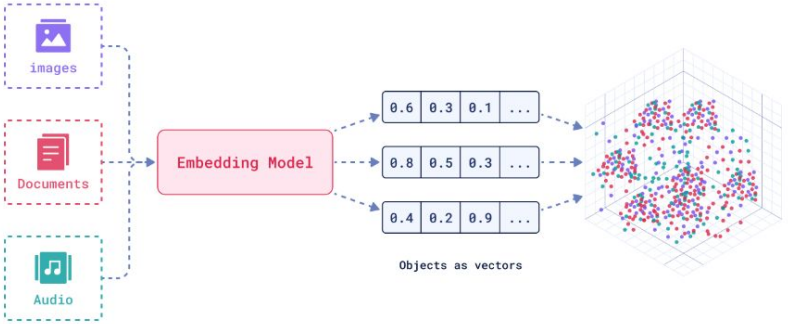
```
def fizz():
    for i in range(1, 101):
        if i % 5 == 0 and i % 3 == 0:
            print("fizzbuzz")
        elif i % 5 == 0:
            print("buzz")
        elif i % 3 == 0:
            print("fizz")
        else:
            print(i)
```

Token count

77

```
def·fizz():\n
····for·i·in·range(1,·101):\n
········if·i·%·5·==·0·and·i·%·3·==·0:\n
············print("fizzbuzz")\n
········elif·i·%·5·==·0:\n
············print("buzz")\n
········elif·i·%·3·==·0:\n
············print("fizz")\n
········else:\n
············print(i)
```


RAW Text
"Tokenization basics!"
Tokenization
"Token"
"ization"
" basics"
"!"
Tokens
30642
1634
19165
0
token_ids
Detokenization
"Tokenization basics!"



+

Tiktokenizer

cl100k_base

```
def fizz():
    for i in range(1, 101):
        if i % 5 == 0 and i % 3 == 0:
            print("fizzbuzz")
        elif i % 5 == 0:
            print("buzz")
        elif i % 3 == 0:
            print("fizz")
        else:
            print(i)
```

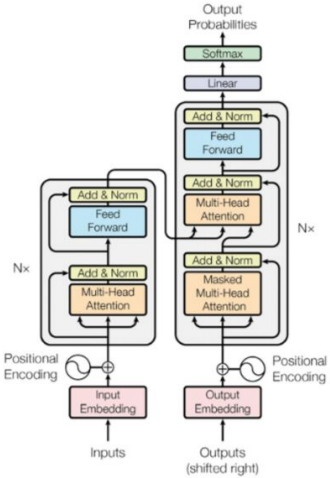
Token count
77

```
def·fizz():\n
····for·i·in·range(1,·101):\n
········if·i·%·5·==·0·and·i·%·3·==·0:\n
············print("fizzbuzz")\n
········elif·i·%·5·==·0:\n
············print("buzz")\n
········elif·i·%·3·==·0:\n
············print("fizz")\n
········else:\n
············print(i)
```

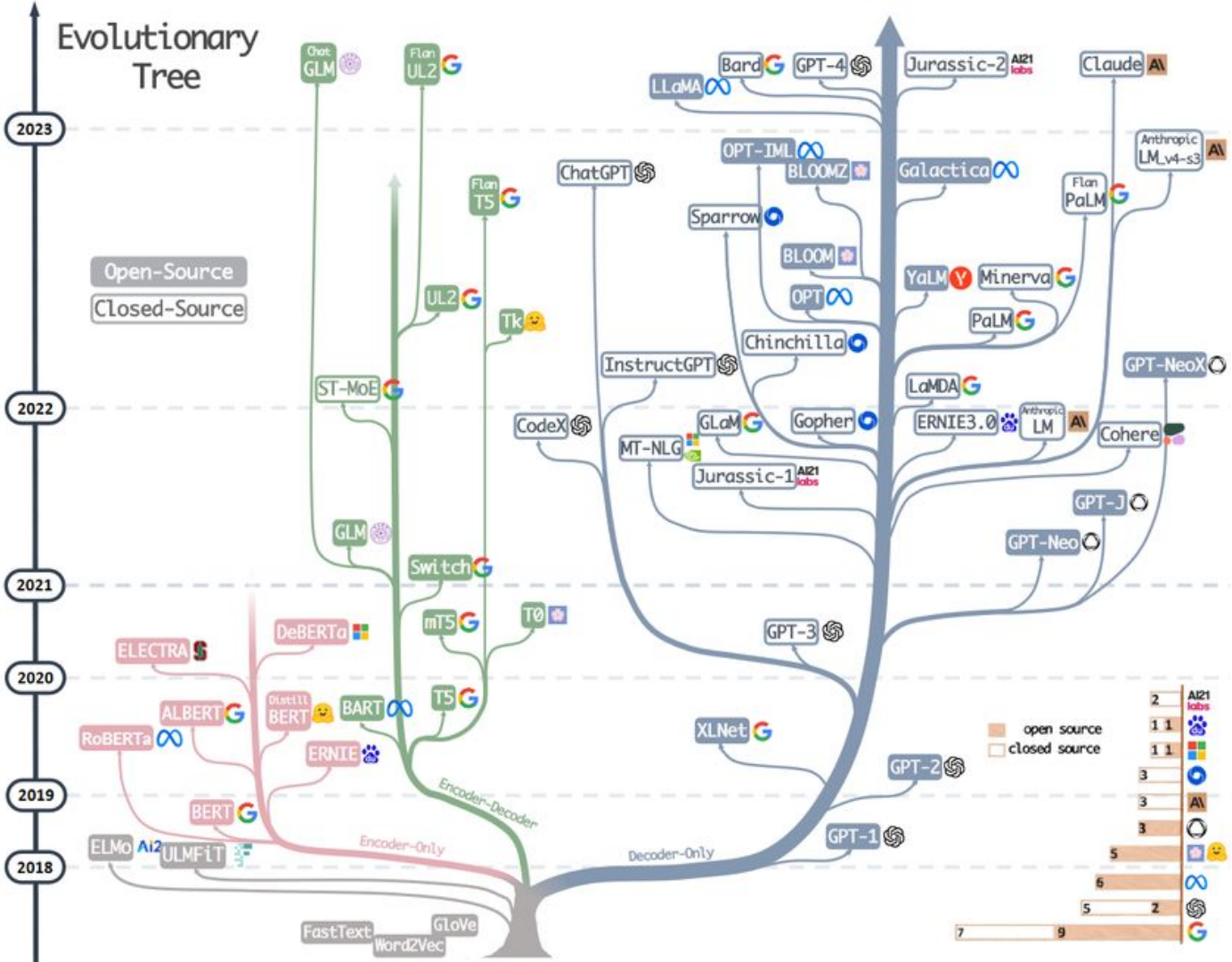
+

Transformer

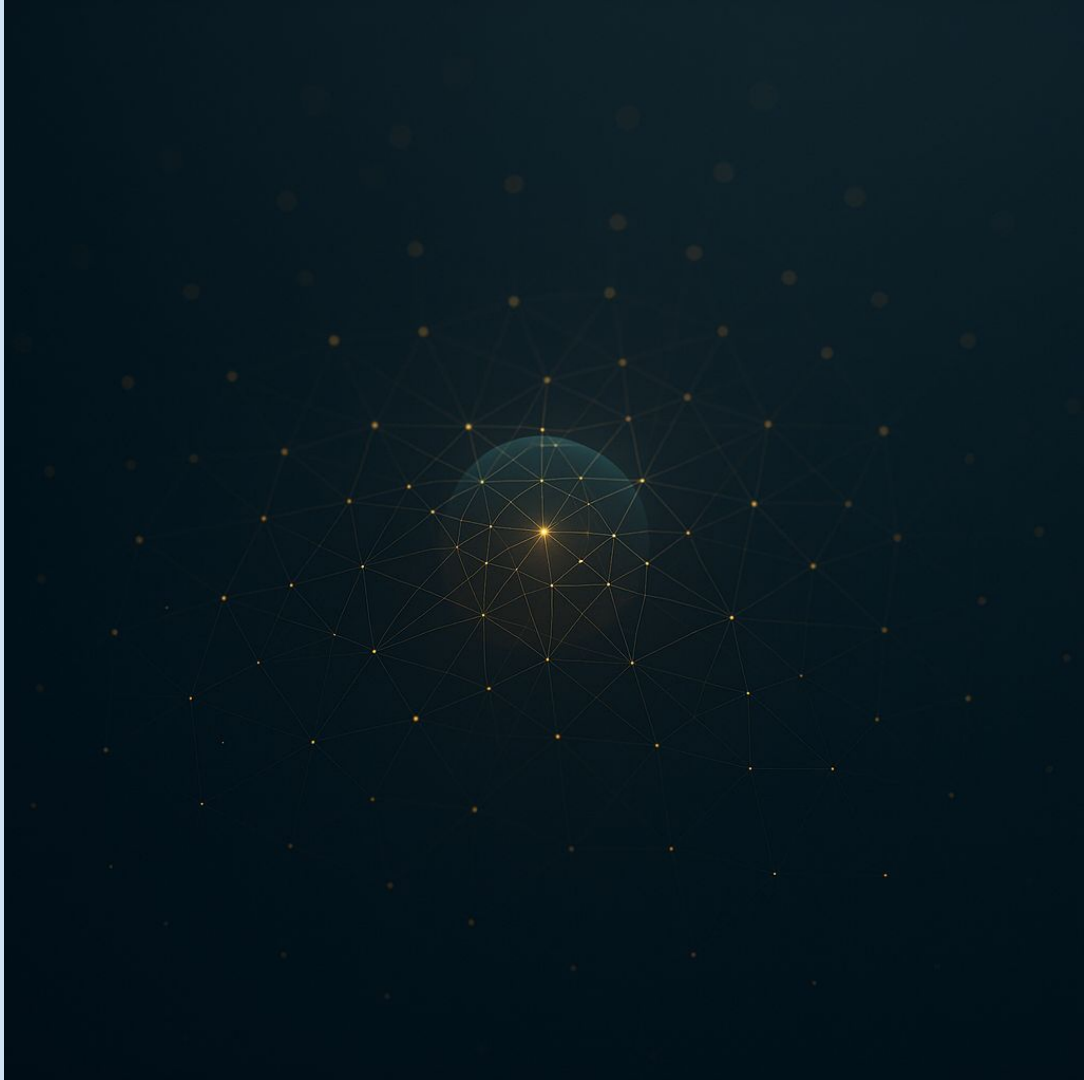
Attention Is All You Need

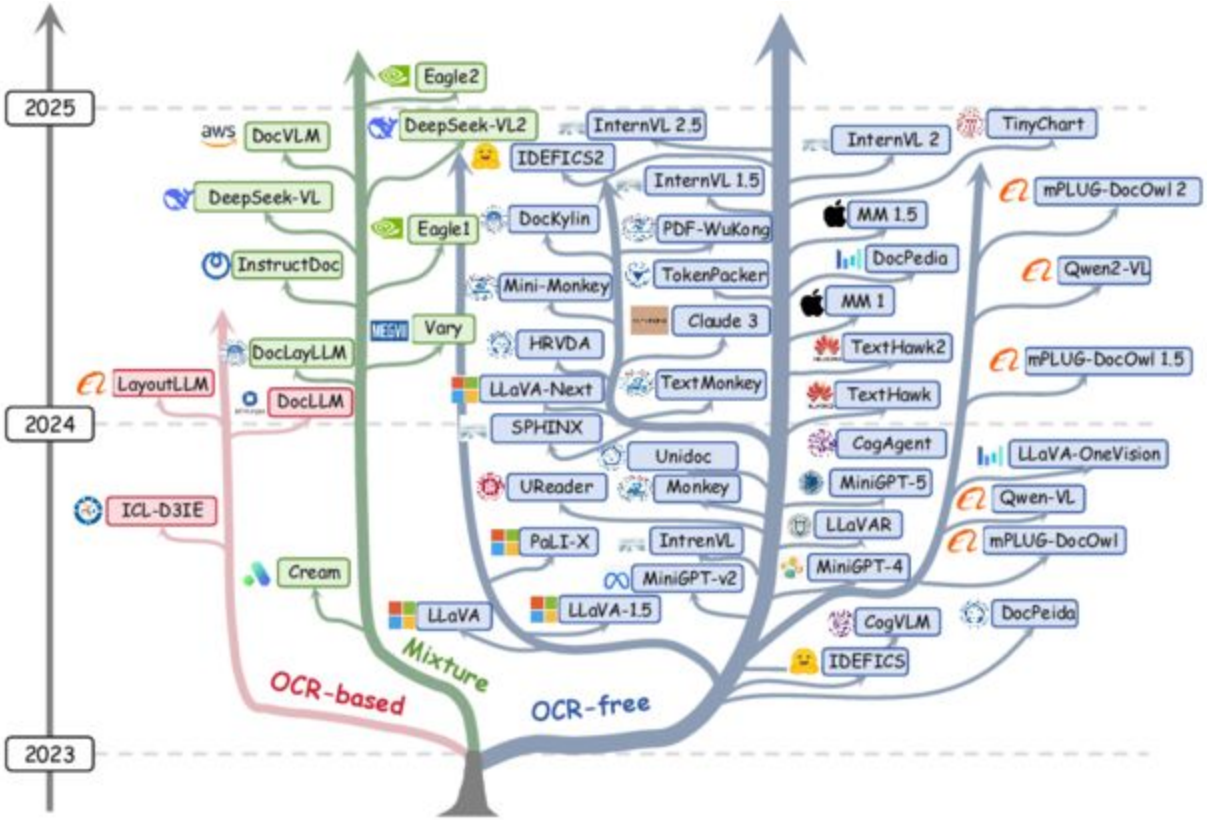


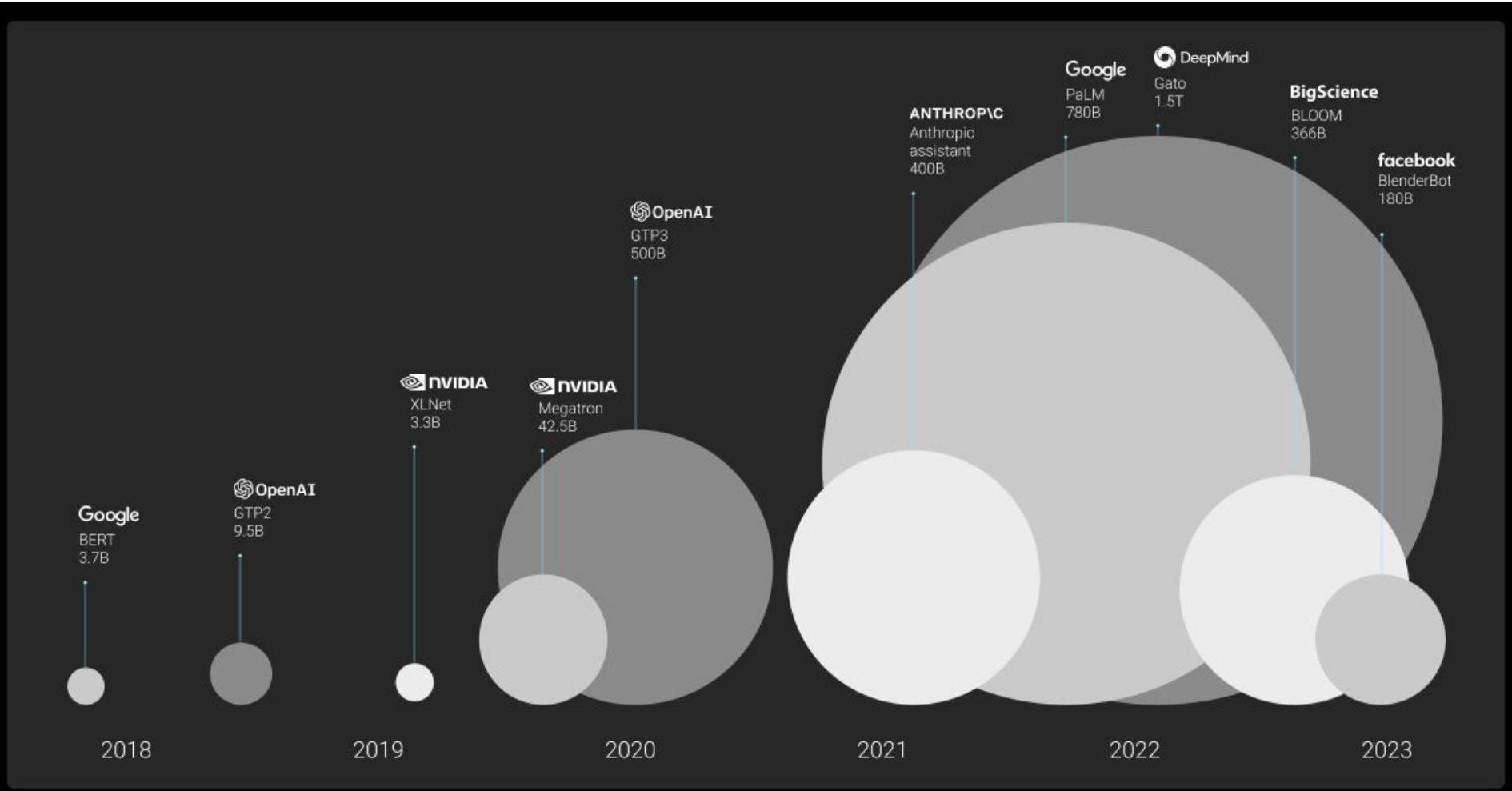
=


Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Chat GLM
Flan UL2
Bard
GPT-4
Jurassic-2
Claude
LLaMA
OPT-IML
BLOOMZ
ChatGPT
Galactica
Anthropic LM_v4-s3
Flan T5
Flan PaLM
Sparrow
BLOOM
YaLM
Minerva
UL2
OPT
Tk
PaLM
Chinchilla
InstructGPT
GPT-NeoX
ST-MoE
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GLM
GPT-Neo
Switch
mT5
T0
GPT-3
DeBERTa
ELECTRA
Distill BERT
BART
T5
ALBERT
RoBERTa
ERNIE
XLNet
GPT-2
BERT
Encoder-Decoder
ELMo
ULMFiT
Encoder-Only
Decoder-Only
GPT-1
FastText
GloVe
Word2Vec
open source
closed source

LLM




2025
2024
2023
Eagle2
DocVLM
DeepSeek-VL2
InternVL 2.5
InternVL 2
TinyChart
IDEFICS2
InternVL 1.5
mPLUG-DocOwl 2
DeepSeek-VL
Eagle1
DocKylin
PDF-WuKong
MM 1.5
InstructDoc
DocPedia
Qwen2-VL
Mini-Monkey
TokenPacker
MM 1
Vary
Claude 3
DocLayLLM
HRVDA
TextHawk2
mPLUG-DocOwl 1.5
LayoutLLM
LLaVA-Next
TextMonkey
TextHawk
DocLLM
SPHINX
CogAgent
LLaVA-OneVision
Unidoc
ICL-D3IE
UReader
Monkey
MiniGPT-5
Qwen-VL
LLaVAR
PoLI-X
IntrenVL
mPLUG-DocOwl
Cream
MiniGPT-v2
MiniGPT-4
LLaVA
LLaVA-1.5
CogVLM
DocPeida
IDEFICS
Mixture
OCR-based
OCR-free


Google
BERT
3.7B
OpenAI
GTP2
9.5B
NVIDIA
XLNet
3.3B
NVIDIA
Megatron
42.5B
OpenAI
GTP3
500B
ANTHROP\C
Anthropic
assistant
400B
Google
PaLM
780B
DeepMind
Gato
1.5T
BigScience
BLOOM
366B
facebook
BlenderBot
180B
2018
2019
2020
2021
2022
2023

SLM: Small (or smart) Language model

r/OpenAI • 2 mo. ago
bgboy089

GPT-5 is actually a much smaller model

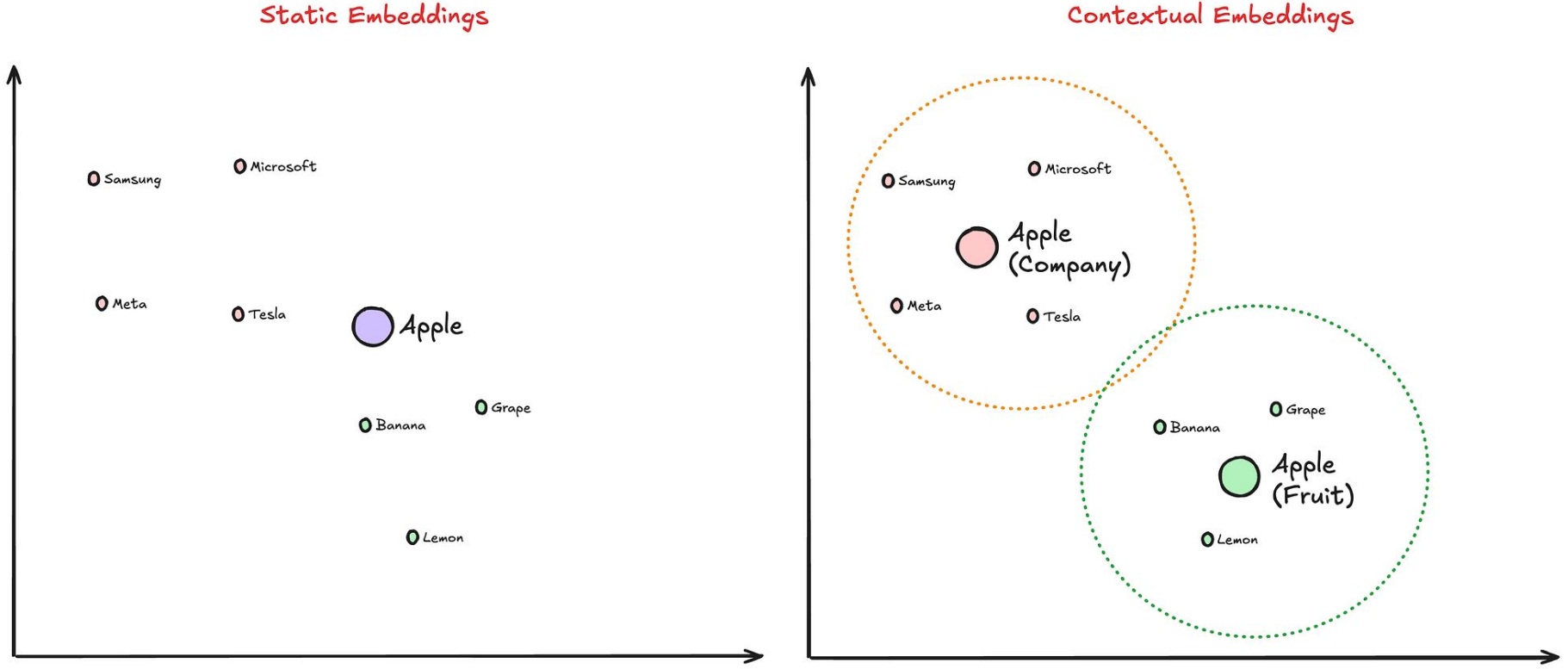
Discussion

Another sign that GPT-5 is actually a much smaller model: just days ago, OpenAI's O3 model, arguably the best model ever released, was limited to 100 messages per week because they couldn't afford to support higher usage. That's with users paying $20 a month. Now, after backlash, they've suddenly increased GPT-5's cap from 200 to 3,000 messages per week, something we've only seen with lightweight models like O4 mini.

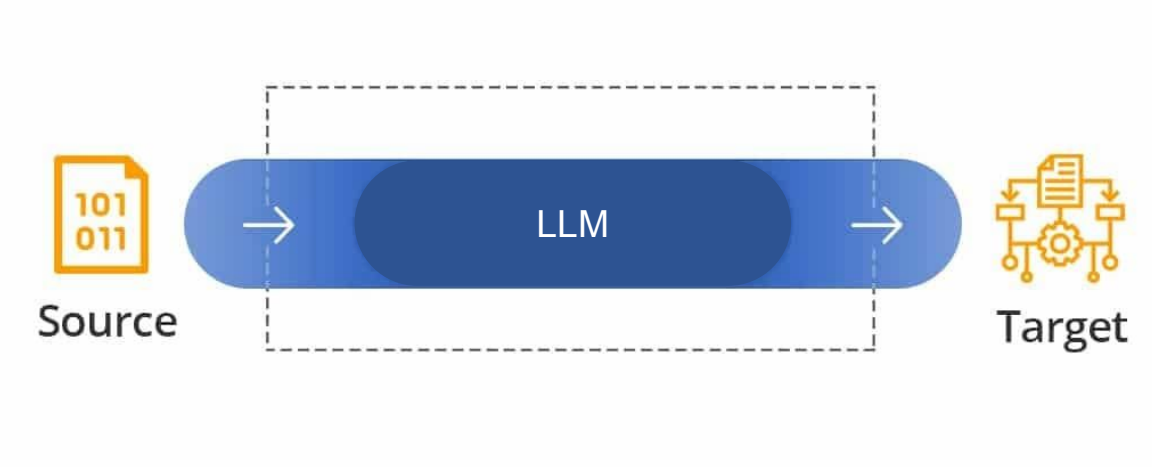
If GPT-5 were truly the massive model they've been trying to present it as, there's no way OpenAI could afford to give users 3,000 messages when they were struggling to handle just 100 on O3. The economics don't add up. Combined with GPT-5's noticeably faster token output speed, this all strongly suggests GPT-5 is a smaller, likely distilled model, possibly trained on the thinking patterns of O3 or O4, and the knowledge base of 4.5.

MoE: Mixture of Experts

Contextual tokens


Static Embeddings
Samsung
Microsoft
Meta
Tesla
Apple
Grape
Banana
Lemon
Contextual Embeddings
Samsung
Microsoft
Apple
(Company)
Meta
Tesla
Grape
Banana
Apple
(Fruit)
Lemon

Level 1: input-output



LLM pipelines

RAG

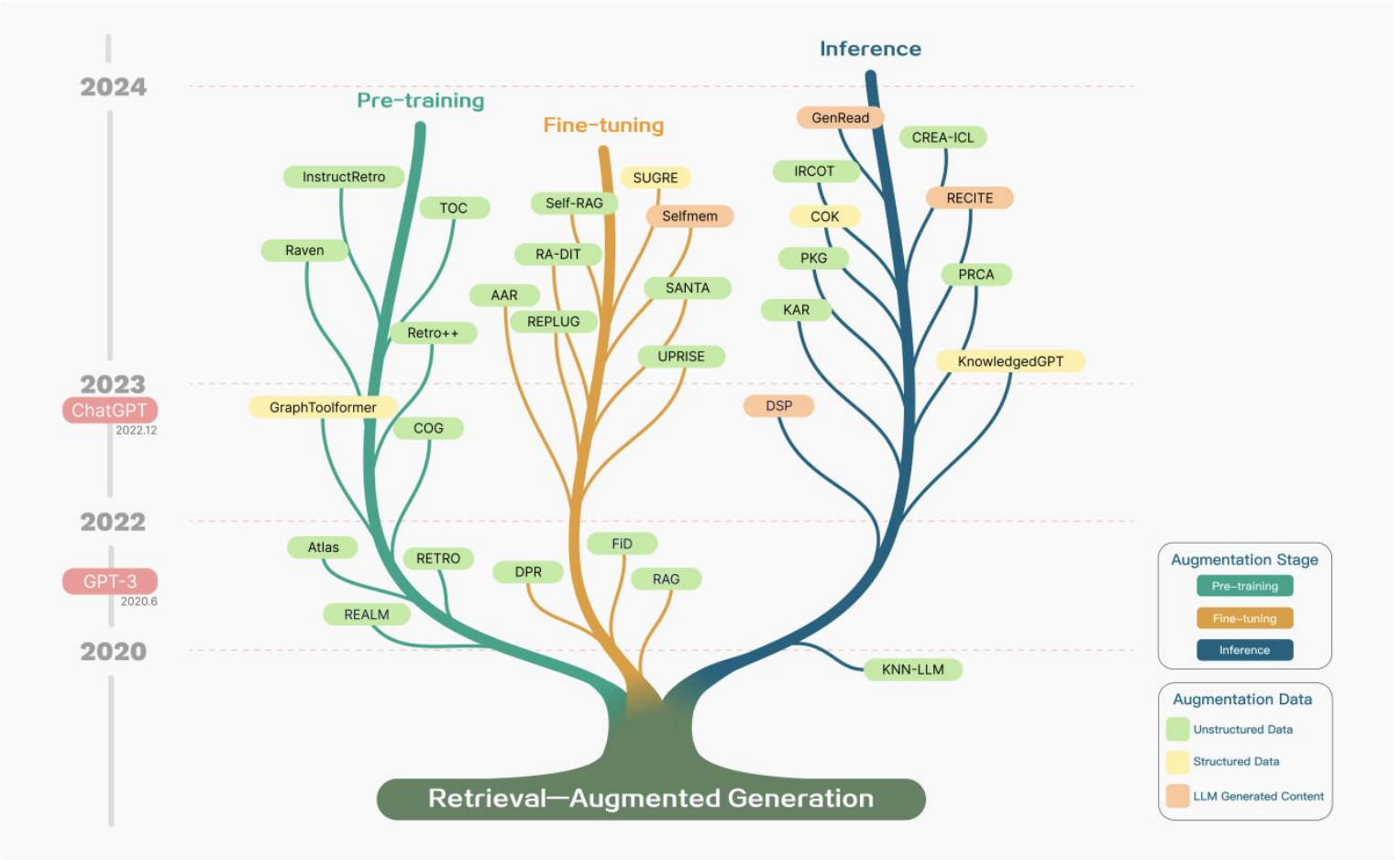
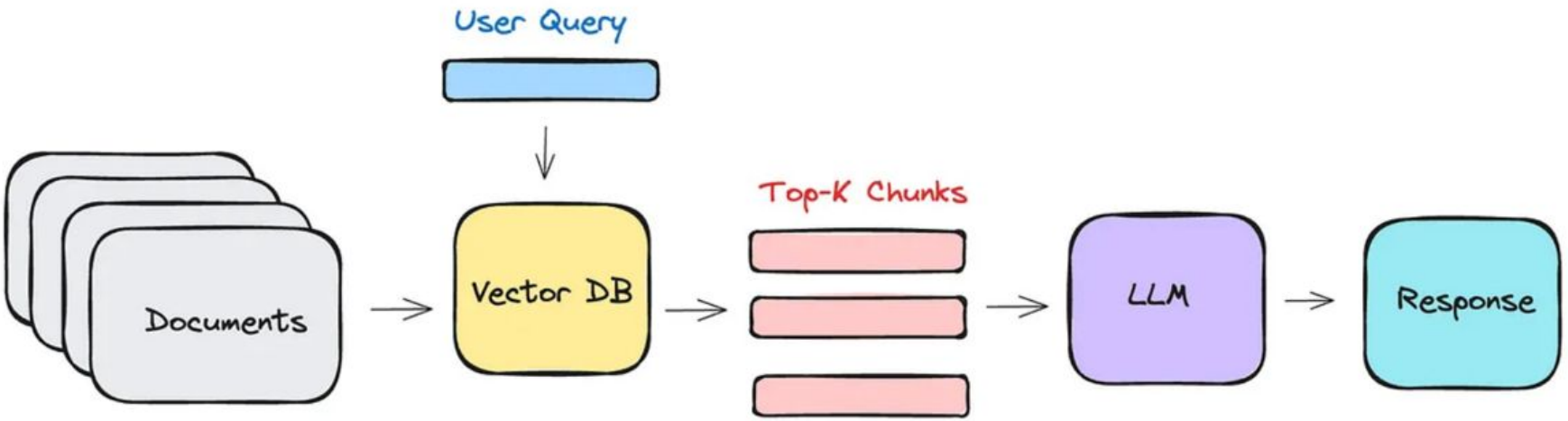


Figure 1: A timeline of existing RAG research. The timeline was established mainly according to the release date.

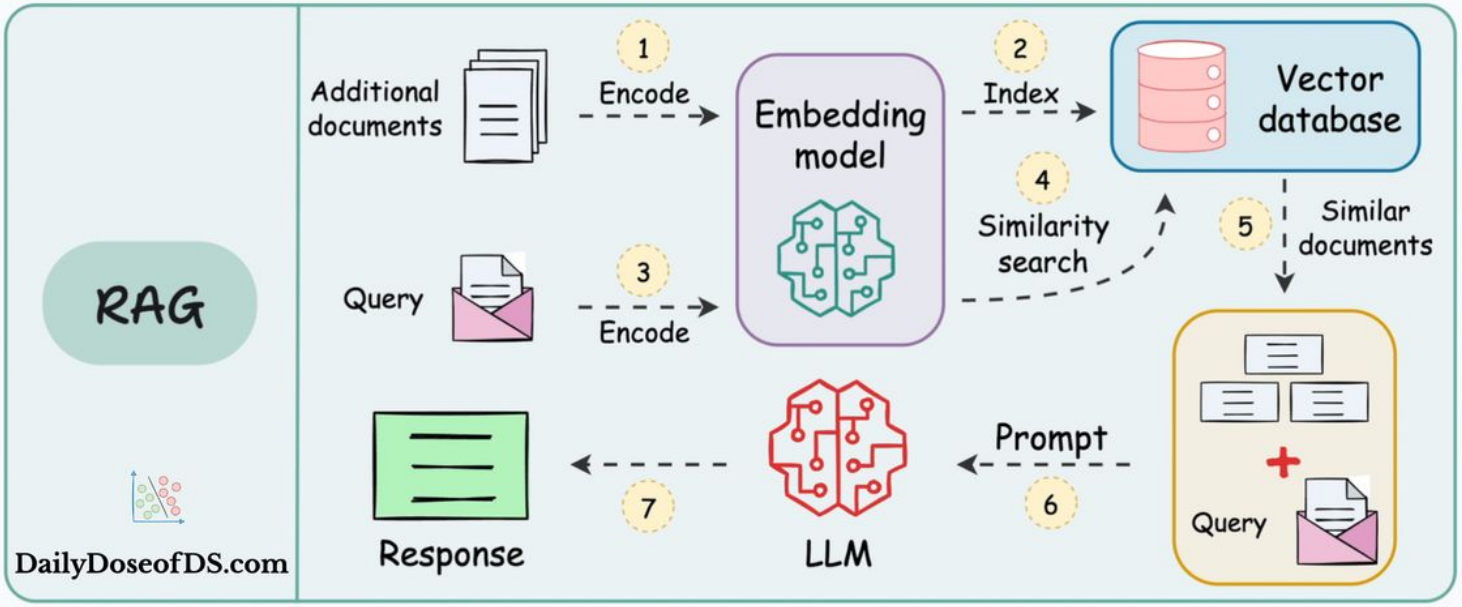
Basic RAG Pipeline



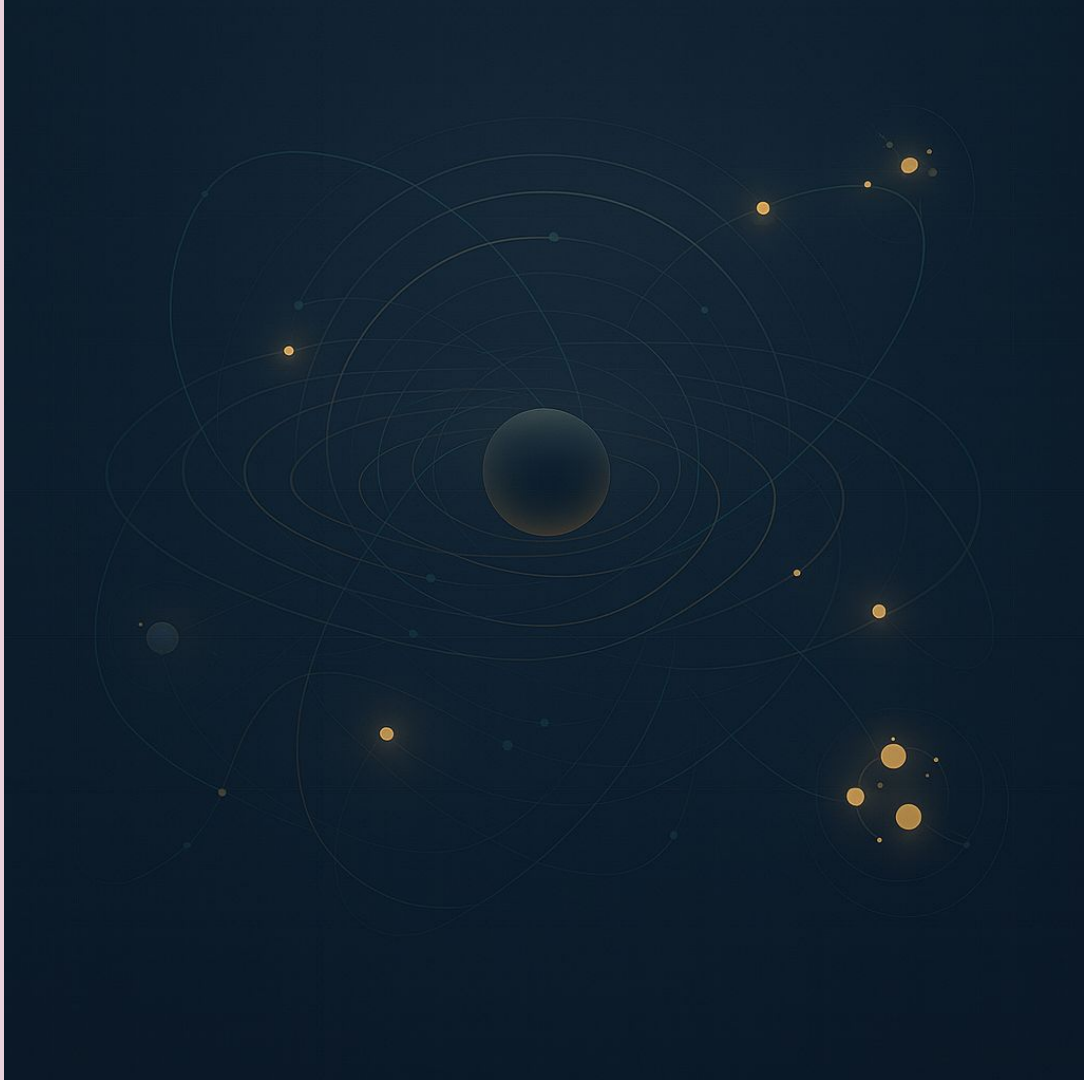
Step 1: Data Indexing

Step 2: Data Retrieval & Generation

16 techniques to supercharge RAG systems



Agentic



LLM (raw)

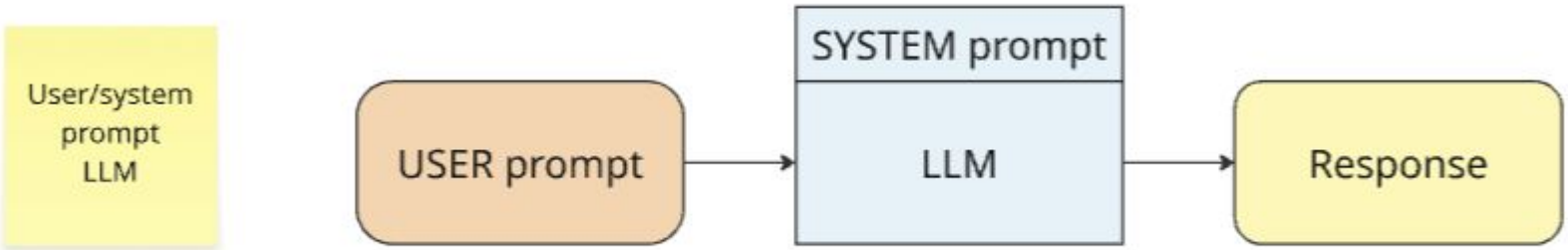
→ get an answer based on the trained model

Raw LLM

LLM (system prompt)

→ get an answer based on the trained model

→ Separates request (USER prompt) and model behaviour (POST-training SYSTEM prompt)



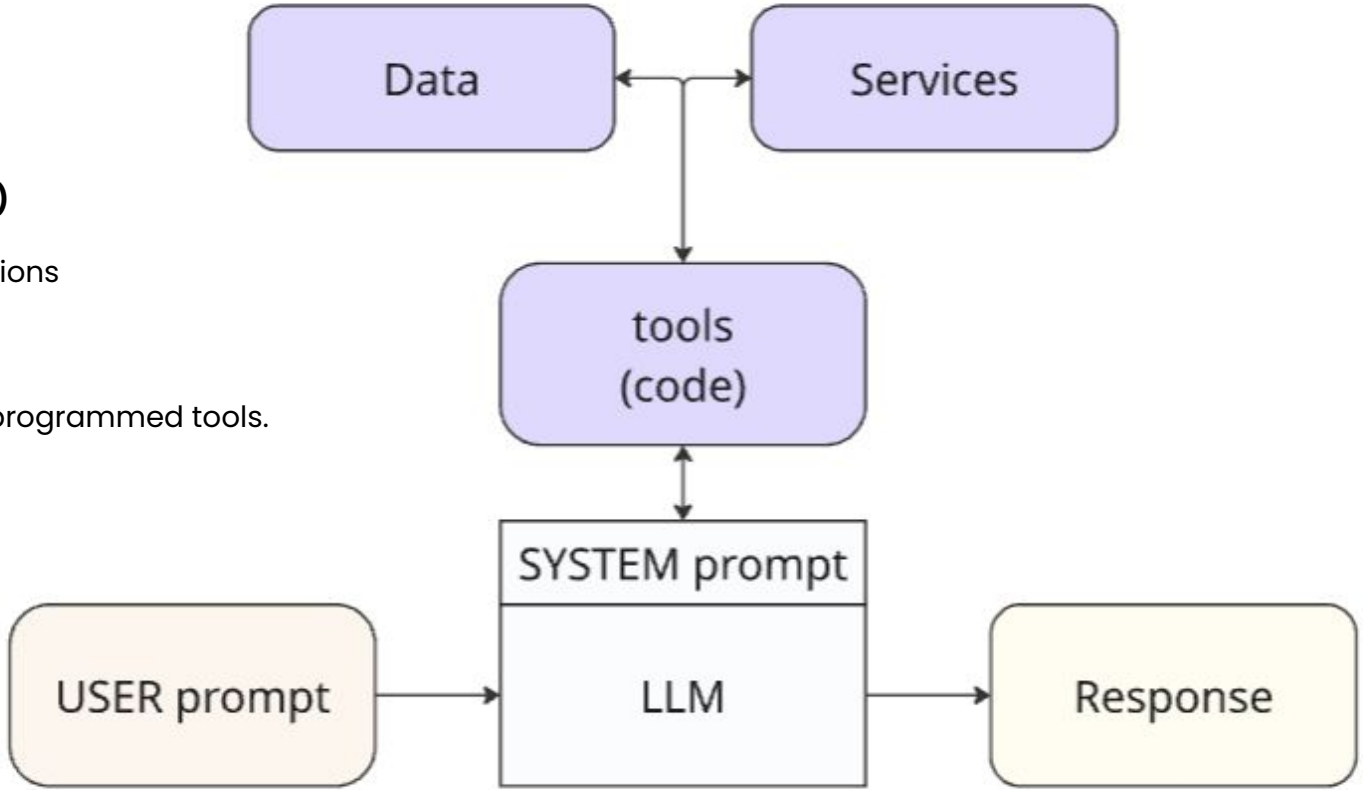
Tools (functions for LLMS)

→ Let LLMs run code functions

→ tool-augmented LLM:

Gives LLM access to pre-programmed tools.

Tools



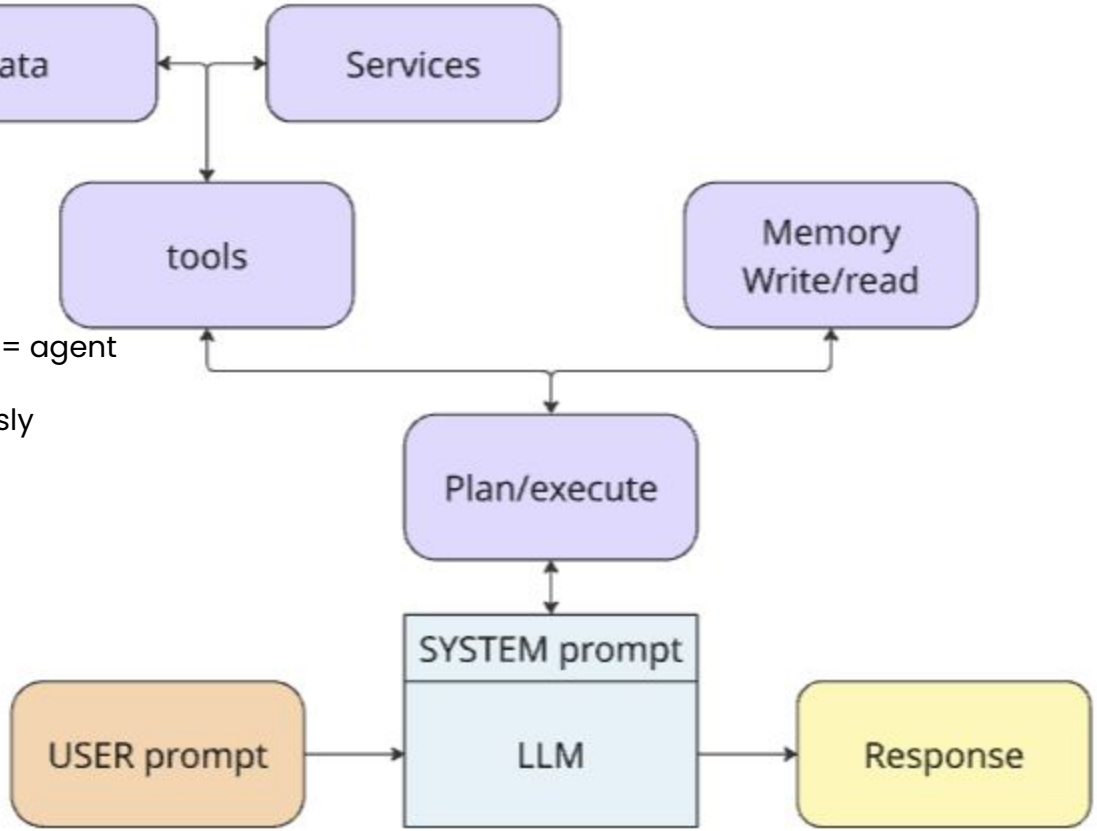

Final Response
User Question
LLM
PARSER
TOOL
OBSERVATION
Output
Loop until final response

Agents (LLMs with tools & planning)

→ Can break down big tasks into smaller ones

→ LLM + tools + control loop/memory/planning = agent

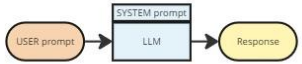
→ Able to complete requests more autonomously



planner & executor pattern

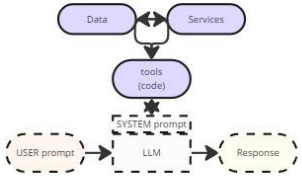
Agent

Raw LLM



User/system prompt LLM

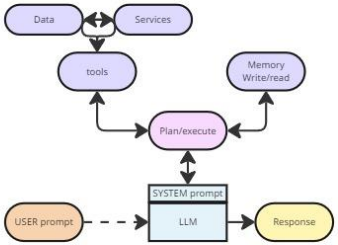
prompt engineering



LLM + Tools

prompt engineering

Tools



Agent

Context engineering

Tools Subagents

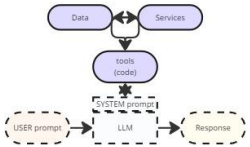
reasoning

MCP

Raw LLM

User/system prompt LLM

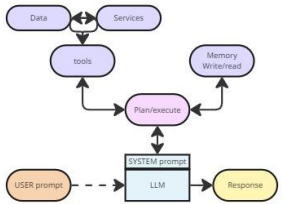
prompt engineering



LLM + Tools

prompt engineering

Tools



Agent

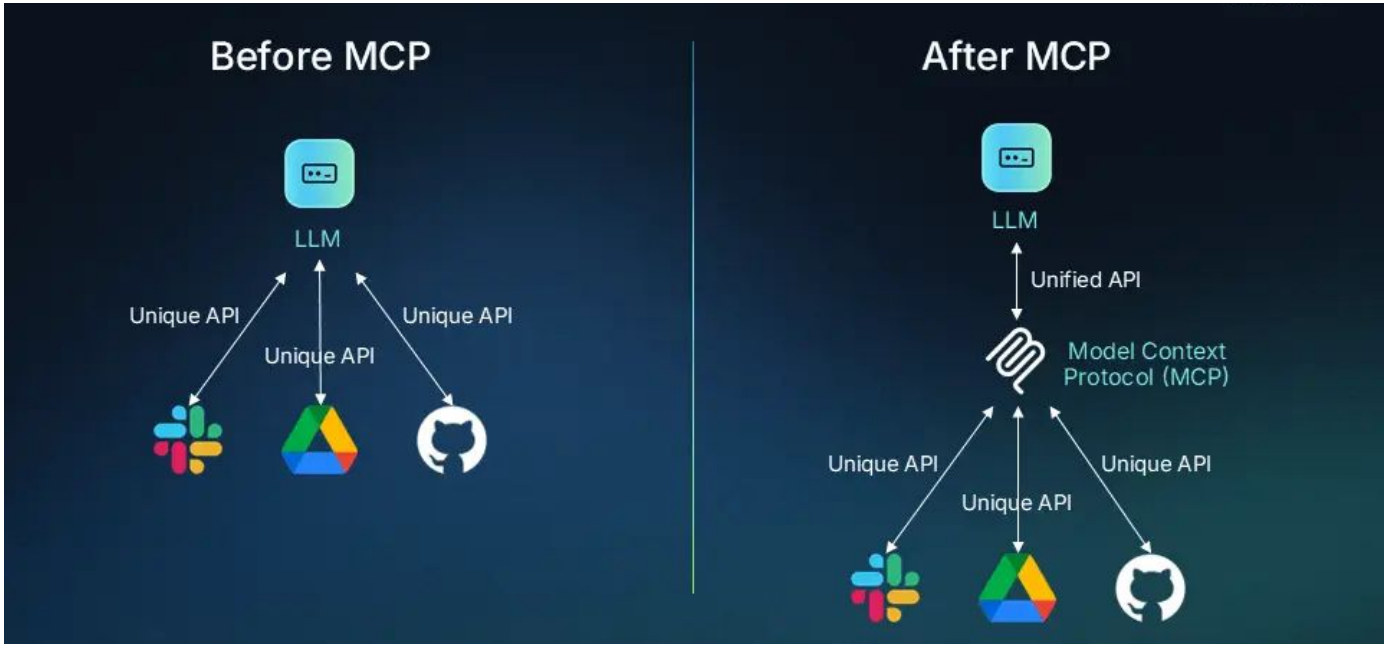
Context engineering

Tools Subagents

reasoning

MCP

MCP (Model Context Protocol)
→ standardized interface for how models access tools, prompts, and resources.
→ equip local LLM agent with discoverable external tools.



Server Key Components

MODEL CONTROLLED

Tools

Functions invoked by the model

- Retrieve / search
- Send a message
- Update DB records

APPLICATION CONTROLLED

Resources

Data exposed to the application

- Files
- Database Records
- API Responses

USER CONTROLLED

Prompts

Pre-defined templates for AI interactions

- Document Q&A
- Transcript Summary
- Workflow Automation

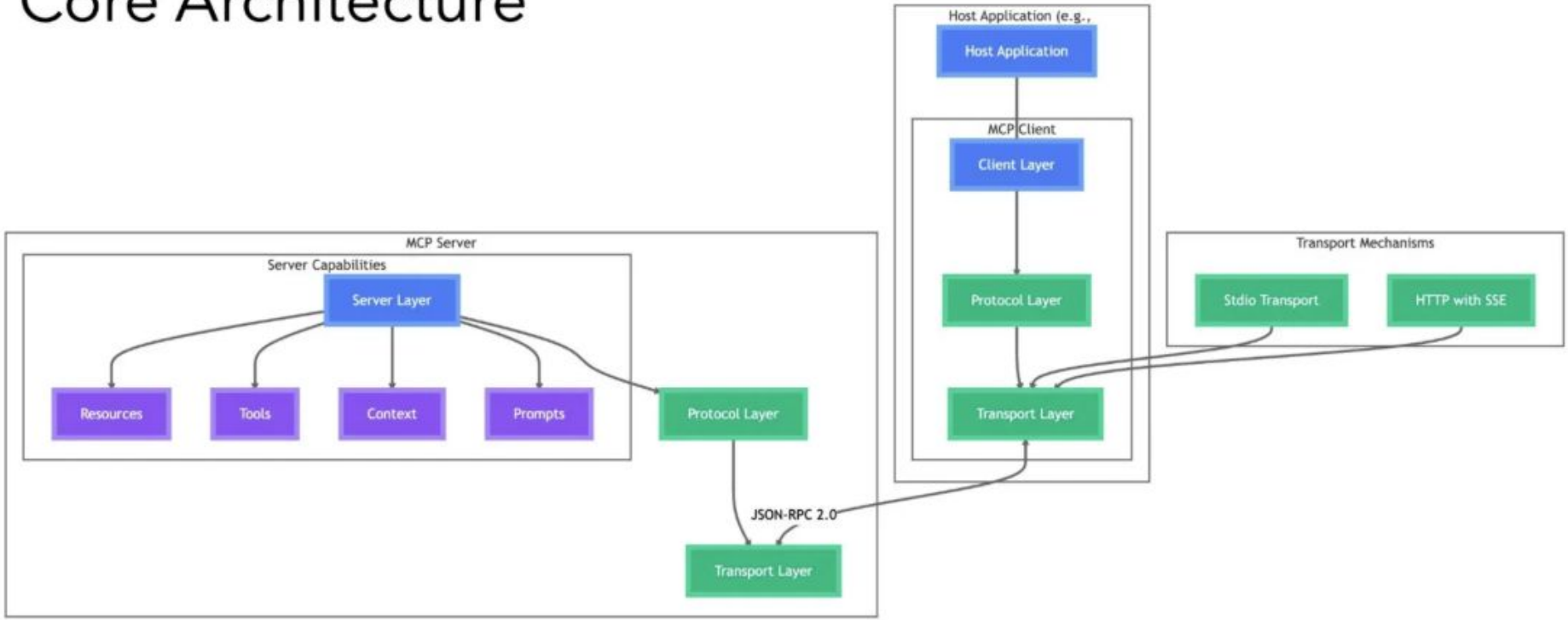
Protocol Layer

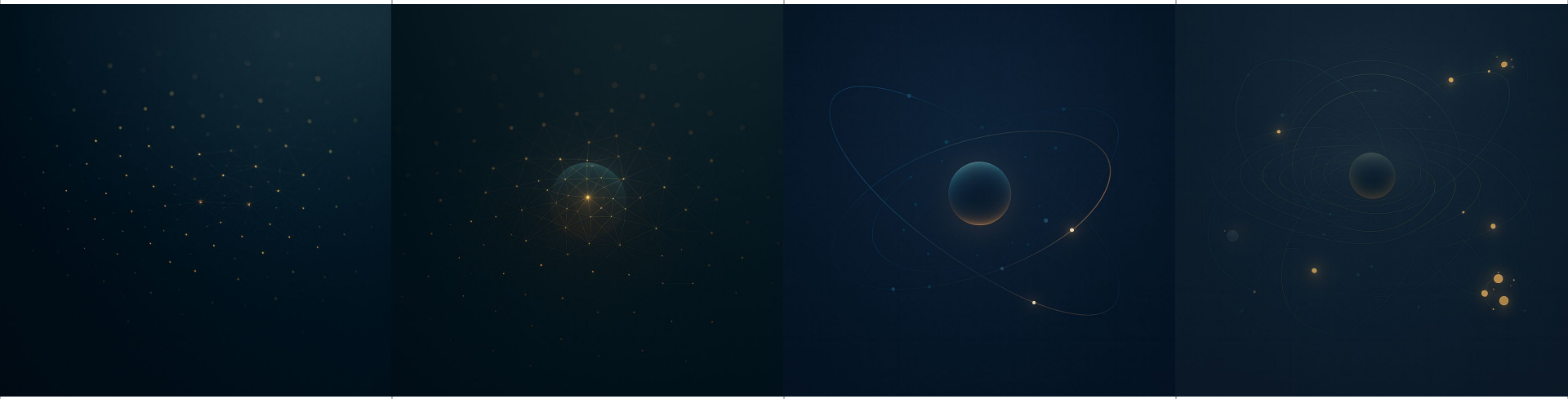
- Message Framing & Routing
- Request/Response Management
- Communication Patterns
- Versioned Protocol Schemas

Transport Layer

- JSON-RPC 2.0 Wire Format
- Protocol Message Transmission
- Multiple Transport Methods
- Extensible Channel System

Core Architecture

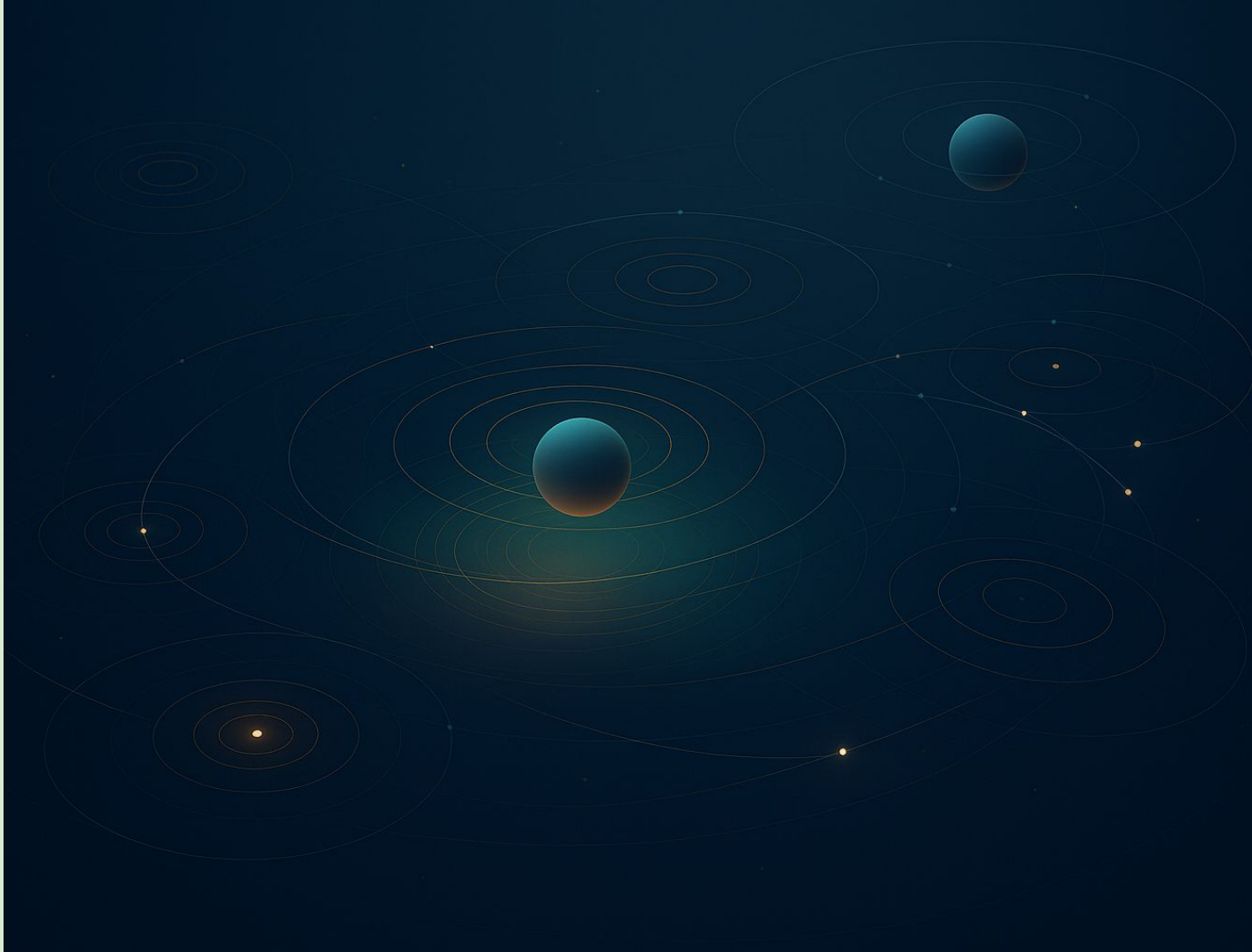




Symbolic AI
Attention mechanism
Transformer Architecture
System Prompting
Autonomous Workflow
Hybrid Retrieval
Vector Databases & Embeddings
Goal Decomposition
Feature Engineering
Knowledge Distillation
LLM to SQL
Multi-Agent systems
Semantic search
Word Embeddings
Tool Use
& Function Calling
Instruction-Tuning
Meta-Prompting
MoE
Tokenization
Chunking & Indexing
MCP
Context Window
Sequence Models (RNN/LSTM)
RLHF
Context Compression / Engineering
Prompt engineering
NLP (Natural Language Processing)
AGI
Human-centered AI
Human-Computer Interaction (HCI)

Whats next?

Agentic economy



Pre-LLM | LLM | RAG | Agentic | Agentic economy

Data-pipelines
Model-pipelines

Vectors & embeddings

→ LLMs

(as a data-pipeline)

→ RAG

(as a LLM-pipeline)

→ Agent

(as an LLM with reasoning and tools)

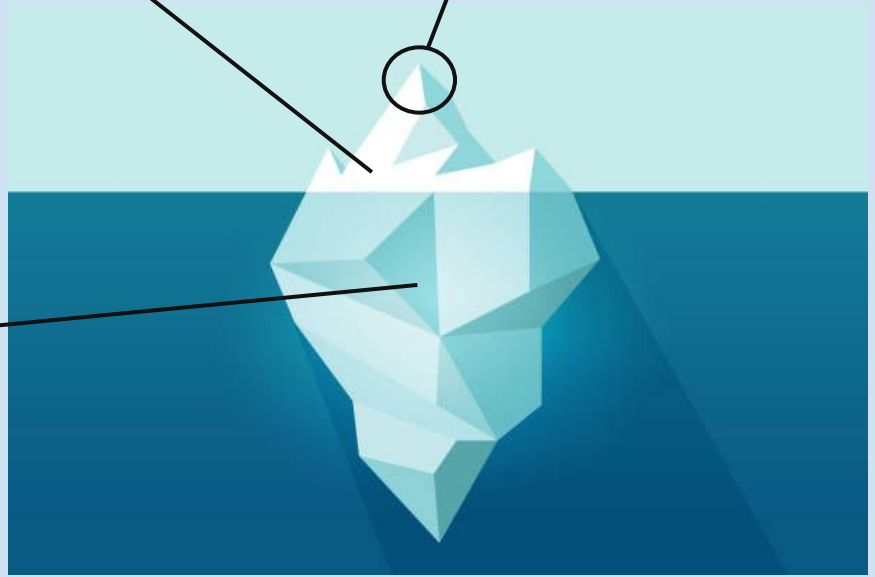
→ Agent to Agent

An economy of agents, exchanging information

Techichal

Societal,
mental/cognitive,
political,
Economic,
Cultural,
Academic,
Environmental,
Ethical,
Educational

This presentation



"Crash course in LLMs"

End / Q&a

Prompt Engineering Guide – https://www.promptingguide.ai/

Daily Dose of Data Science – https://www.dailydoseofds.com/

Model Context Protocol – https://modelcontextprotocol.io/

Costs Effects

Costs Effects

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs

Costs Effects Costs Effects Costs