



AI Pair Programmer

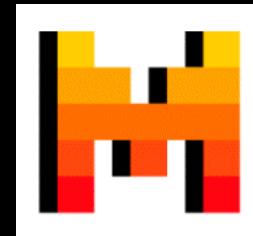
By GitHub, Mistral & RaKeTe-Technology

October, 8th, 2024

Ralph.Kemperdick@RaKeTeTechnology.com

Cloud Solution Architect, Advanced Analytics AI + Data & Analytics

LinkedIn www.linkedin.com/in/ralphke

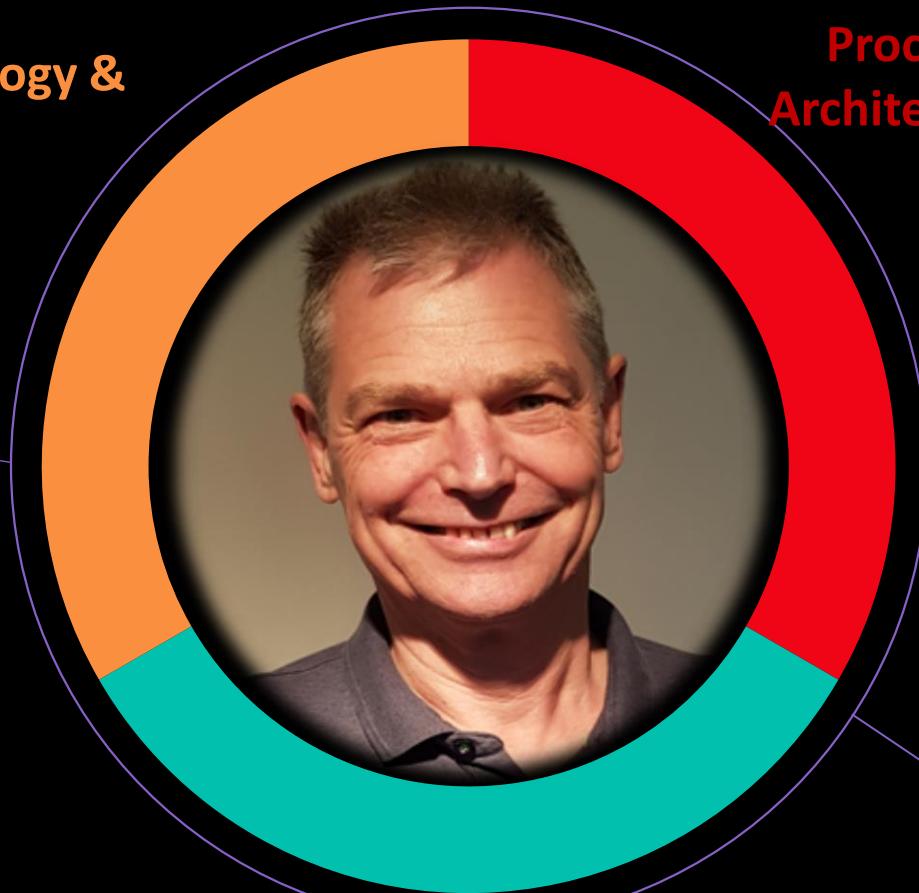


RaKeTe-Technology Offerings



**Technology &
Tools**

Coach on
Generative AI
Big Data
Advanced
Analytics



**Process &
Architecture**

Positioning,
architecture,
evaluations,
Cloud Cost
optimization,
Benchmarking

**People &
Transformation**

Teaming,
networking,
communication
focus on basics
and advanced
use cases

Pair Programming

A comparison of AI programming assistants, exploring the tools for coding with GitHub and Mistral, while also touching upon context and benchmarks.

```
mirror_mod = modifier_obj
# Set mirror object to mirror
mirror_mod.mirror_object = modifier_obj
if operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
elif operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
modifier_ob.select= 1
target_ob.select=1
context.scene.objects.active = modifier
modifier.select = 0
bpy.context.selected_objects.append(modifier)
data.objects[one.name].select = 1
int("please select exactly one object")
- OPERATOR CLASSES ----
types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"
    context):
        context.active_object is not None
        if context.active_object is not None:
```

Agenda

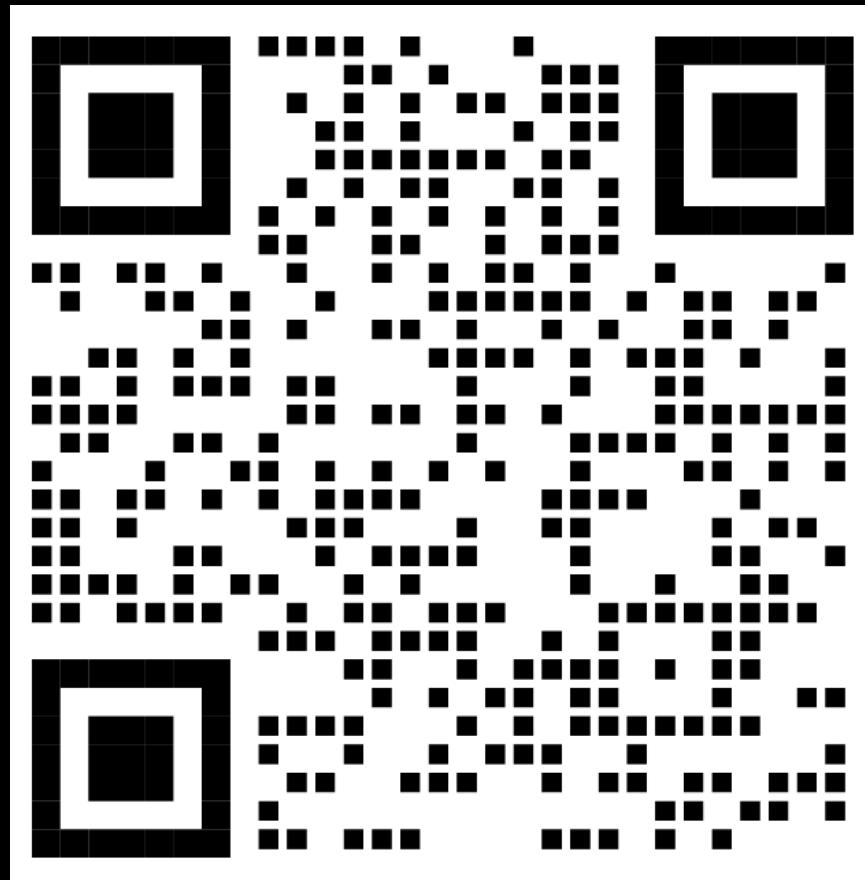
- Get Ready to participate (5)
- AI-Pair Programmer, Quick recap (10)
- Components and Tools
 - GitHub Enterprise (30)
 - Mistral Codestral (10)
- Considerations & Limitations (5)
- Choices & Benchmarking (10)
- Pricing (5)
- Prompt Engineering Exercise (30)
- Next Steps (5)
- Q&A (10)



Get ready to participate in the workshop

Scan this Code or enter this URL into your Browser of choice

<https://github.com/ralphke/AI-Pair-Programmer>



Requirements

1. Have a Laptop to work with and sufficient battery life >2h
2. Have your development machine ready to code
3. You need to have a
 1. GitHub account, Personal or Organization or
 2. Microsoft account [*.outlook.com, ...] and
 3. Azure Subscription with permission to deploy any Azure Service
4. You need a Development Environment (IDE):
 1. [GitHub Codespaces](#) (VS Code in the browser) or
 2. [Visual Studio Code - Code Editing. Redefined](#) or
 3. <https://visualstudio.microsoft.com/>
5. Microsoft Copilot can be used for some cases
 1. [Copilot \(microsoft.com\)](#)
6. You need to enable Copilot in your IDE

Visual Studio Code Plugins



- Github Copilot
- GitHub Copilot Chat
- GitHub Codespaces



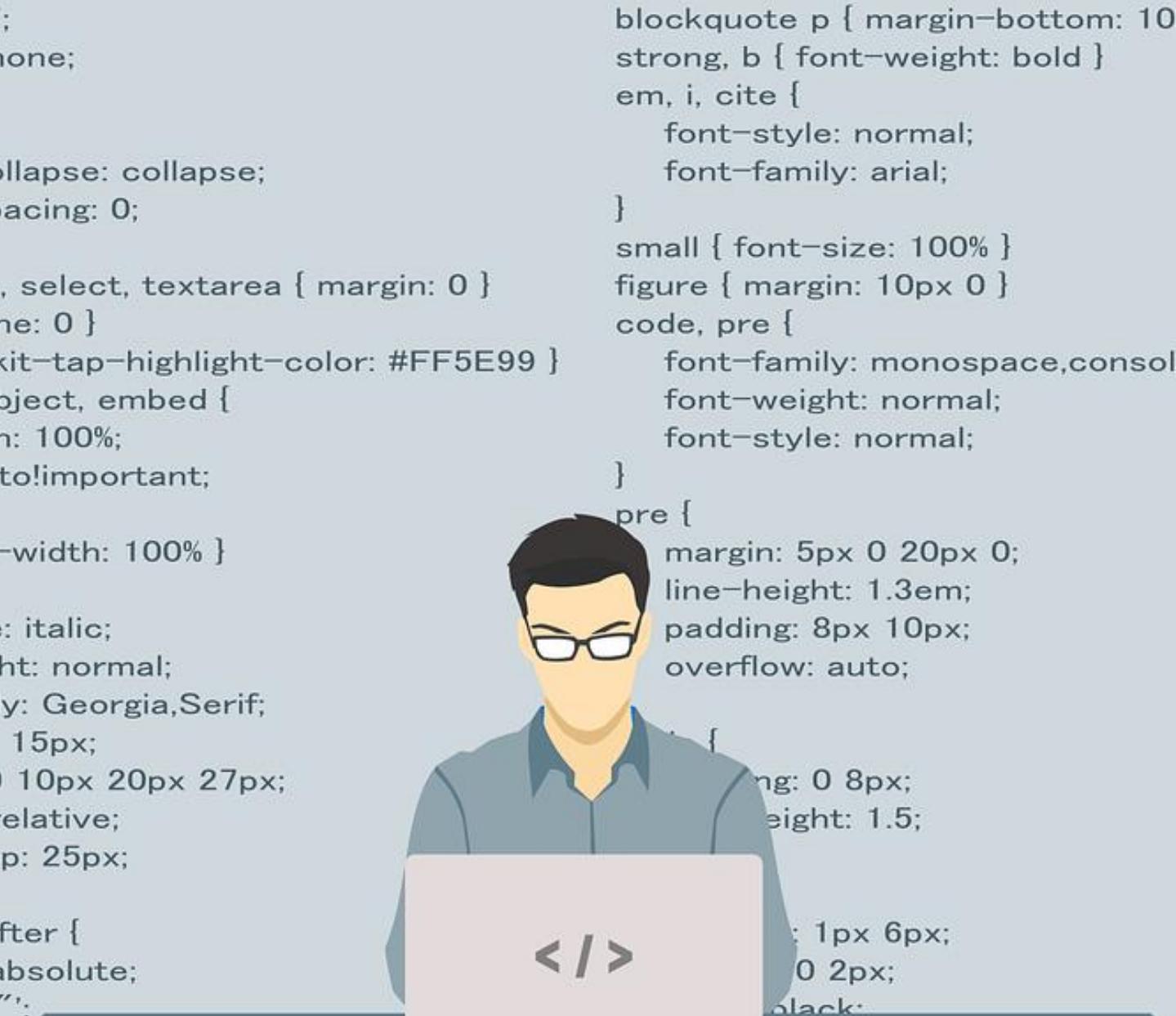
- VS Code Speech



- Continue, Codestral, Claude and more

With all the advances in AI, will we still need programmers in the future?

```
blockquote p { margin-bottom: 10px; }
strong, b { font-weight: bold; }
em, i, cite {
  font-style: normal;
  font-family: arial;
}
small { font-size: 100%; }
figure { margin: 10px 0; }
code, pre {
  font-family: monospace, consolas;
  font-weight: normal;
  font-style: normal;
}
pre {
  margin: 5px 0 20px 0;
  line-height: 1.3em;
  padding: 8px 10px;
  overflow: auto;
}
</>
```



How to think about AI? (Mindset)

No, AI can't do it

Positive attitude

Panic mode



I don't have time to look into AI! 😰

I will be insanely productive! 😎

AI will steal my job! 😥

Generative AI - a quick recap

Artificial Intelligence

Machine Learning

Deep Learning

Generative AI



Artificial Intelligence

the field of computer science that seeks to create intelligent machines that can replicate or exceed human intelligence



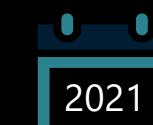
Machine Learning

subset of AI that enables machines to learn from existing data and improve upon that data to make decisions or predictions



Deep Learning

a machine learning technique in which layers of neural networks are used to process data and make decisions



Generative AI

Create new written, visual, and auditory content given prompts or existing data.

175B

GPT-3 | 2020

17B

Turing-NLG | 2020

340M

BERT-Large | 2018

OpenAI



Microsoft



530B

Megatron-Turing
NLG | 2021



175B

GPT-3 | 2020

17B

Turing-NLG | 2020

340M

BERT-Large | 2018

Consider Power Efficiency of LLMs

Don't use LLMs for trivial tasks. Millions of kWh will be wasted!



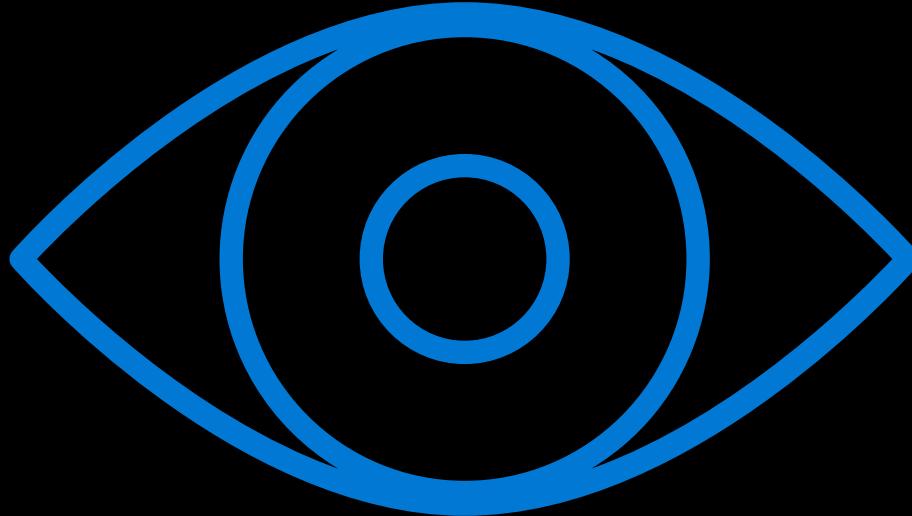
But Power Efficiency is getting better every day

Small Models (SLMs) and Agents can reduce the training and inferencing kWh!





OpenAI GTP-X

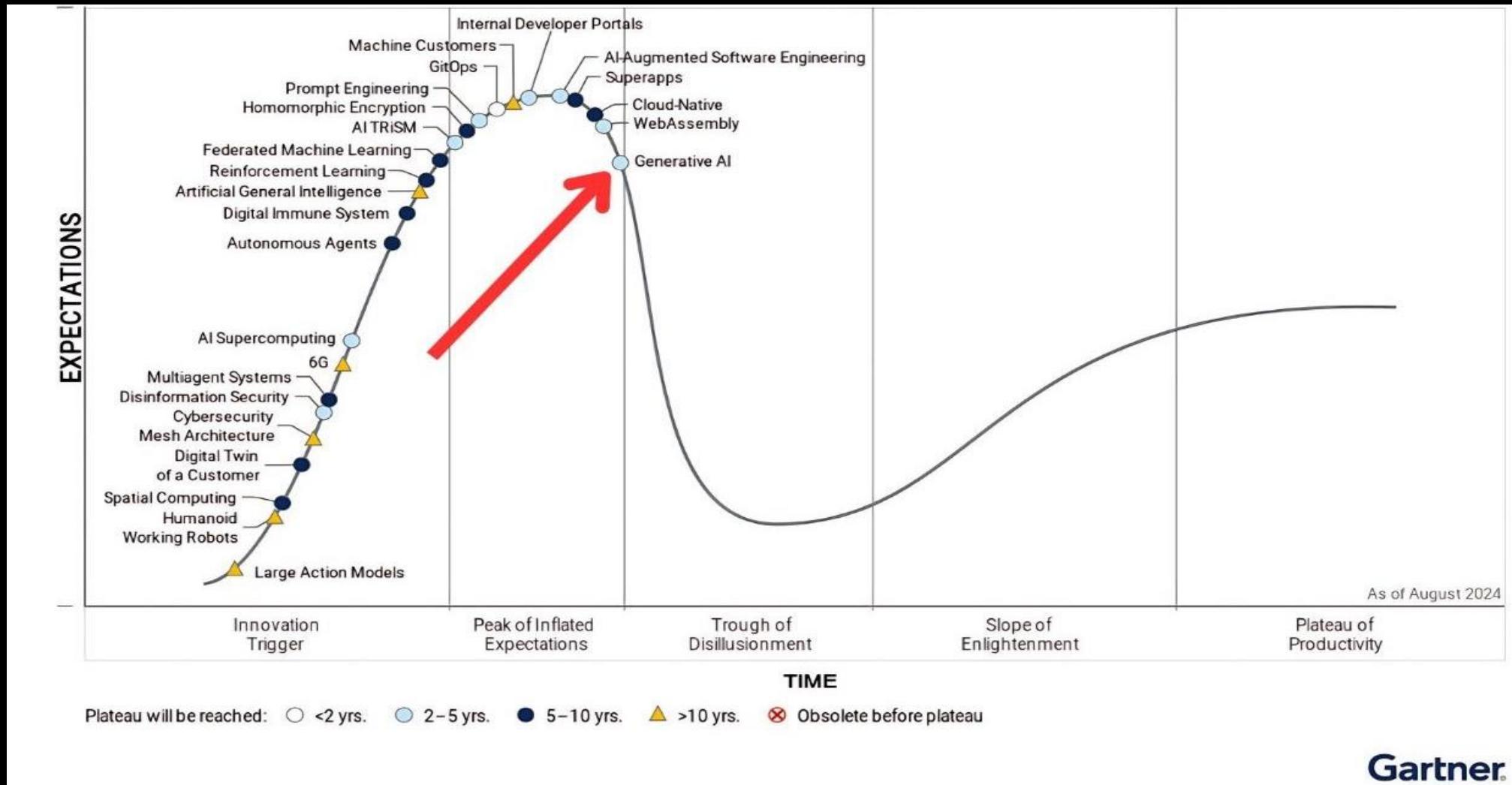


Why is the sky blue?

OpenAI GPT-X answer:

The blue color of the sky is because of a phenomenon called Rayleigh scattering. Sunlight, which is composed of different colors of light, gets scattered by molecules and particles in the Earth's atmosphere, with shorter blue and violet wavelengths being scattered more than longer wavelengths, making the sky appear blue during the day.

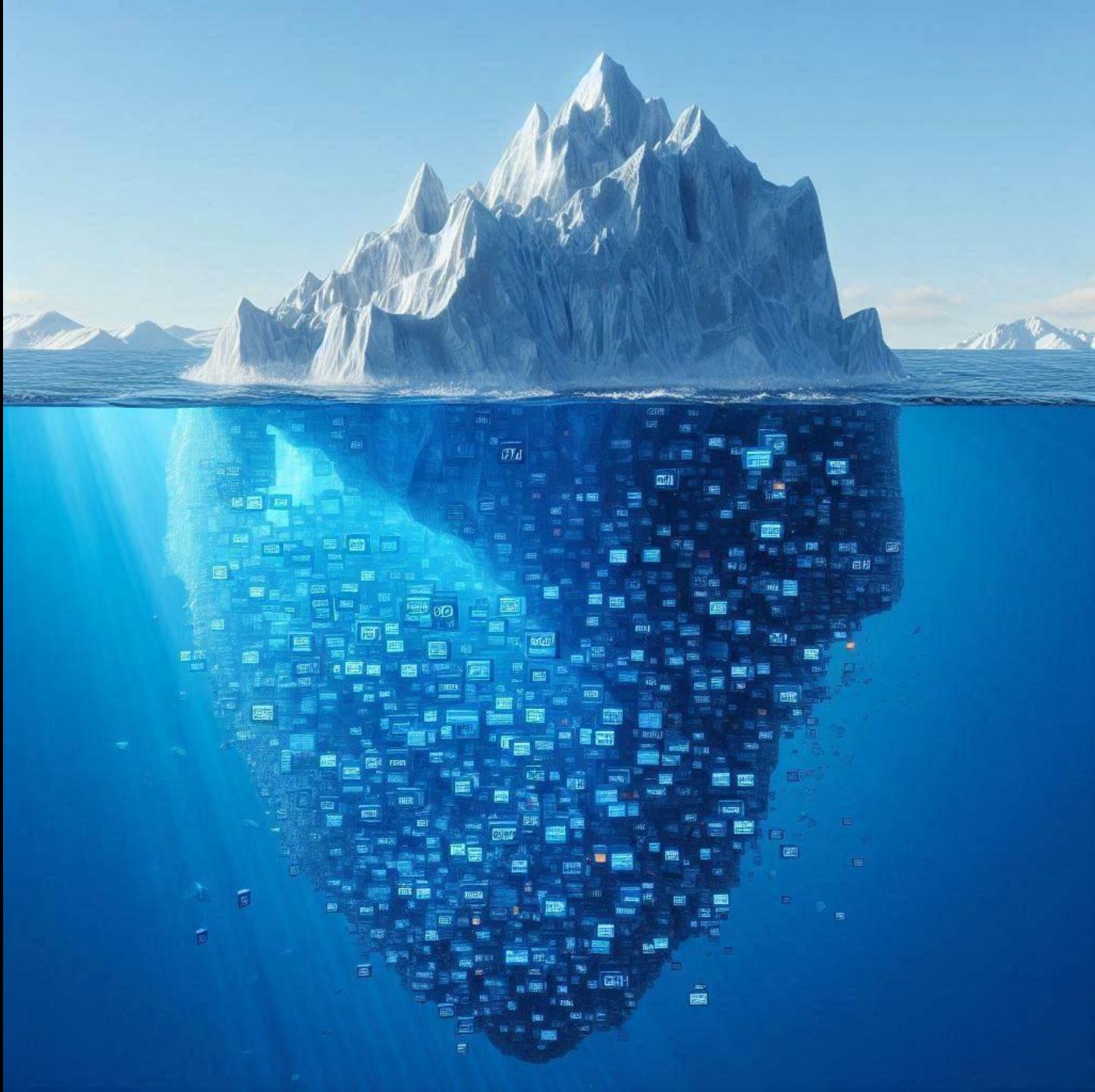
Gartner: Hyper Cycle of Emerging Technologies, 2024



Source: Gartner (August 2024)

Gartner

The amount of code
in closed source
repositories
outshines OSS by an
order of magnitude



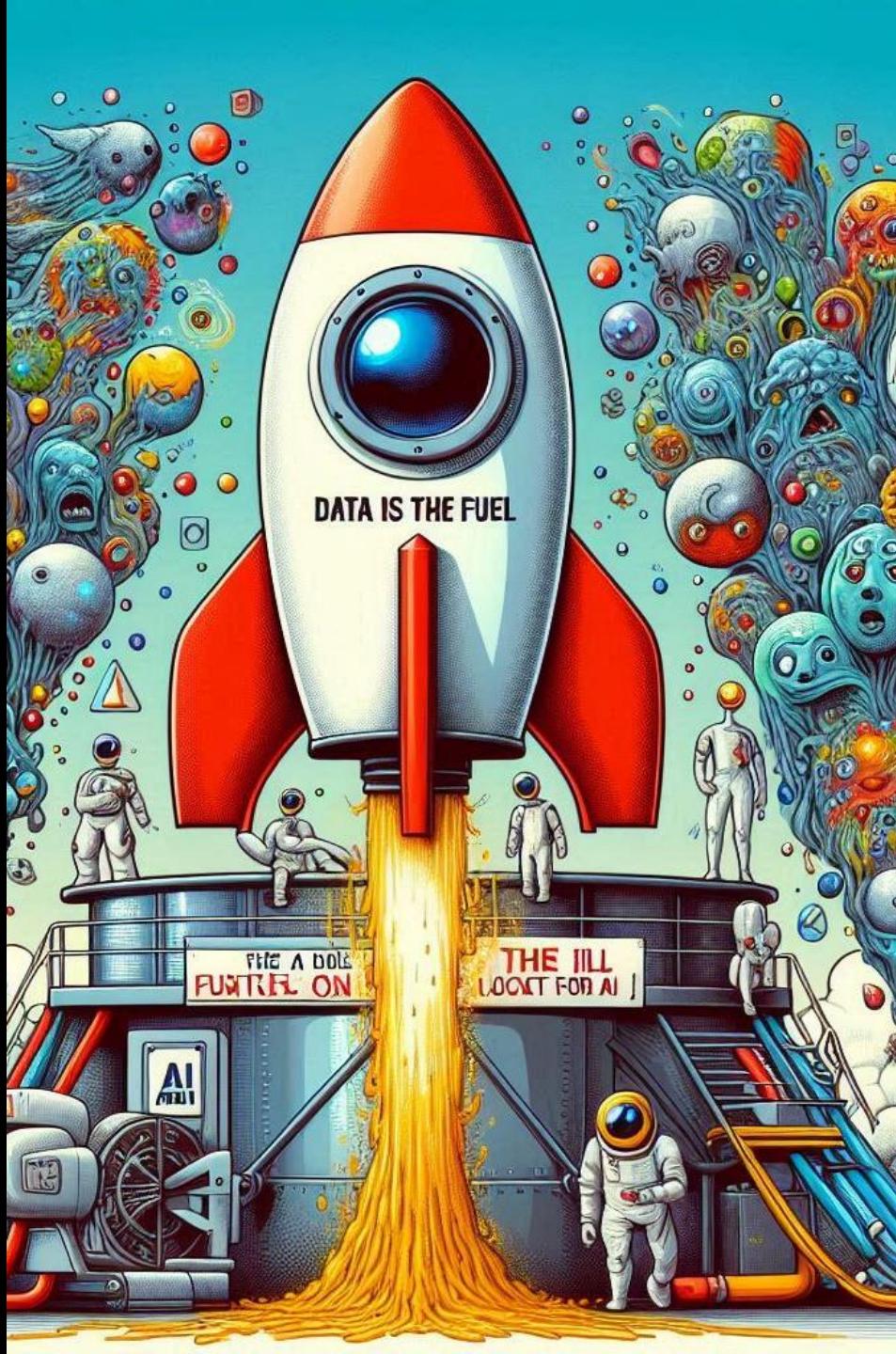
Data is the Fuel for AI

But...

if the fuel is contaminated, then the rocket will not take off... ;)

Look out for:

1. Quality and measure it over time
2. Availability and size
3. Protection
4. Ethics



GitHub Platform



A single integrated enterprise-ready platform

Industry-best collaborative tools for developers

Security at every step of the workflow

Powered by AI



Placeholder GitHub Copilot [Enterprise]

@Malte Lantin, GitHub



GitHub Platform

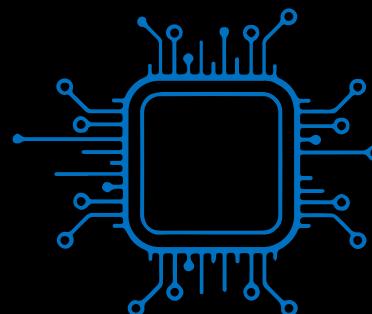


- A single integrated enterprise-ready platform
- Industry-best collaborative tools for developers
- Security at every step of the workflow
- Powered by AI





OpenAI Codex Model



Public code and text
on the internet

GitHub



GitHub
Copilot Service



Don't fly solo.

Provide editor context

Provide suggestions

Improve suggestions

Private code

JS fetch_pic.js

```
1 const fetchNASAPictureOfTheDay = () => {
2   return fetch('https://api.nasa.gov/mars/rovers/curiosity/images/latest?sol=1000')
3     .then(response => response.json())
4     .then(json => {
5       return json;
6     });
7 }
```

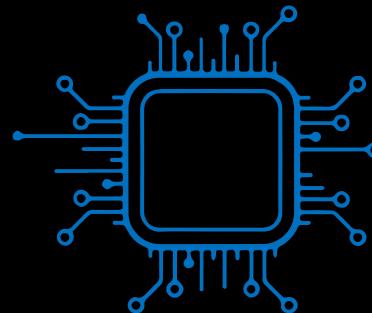
push_to

Copilot

A screenshot of a code editor window showing a JavaScript file named "fetch_pic.js". The code uses the Fetch API to retrieve a NASA Mars rover image. A "Copilot" button is visible at the bottom right of the editor.

OpenAI + GitHub + [Enterprise]

OpenAI Codex Model



Private code and Context
from GitHub Enterprise
Repos

GitHub



GitHub
Copilot Service

Don't fly solo.



Provide editor context

Provide suggestions

Improve suggestions

Private code

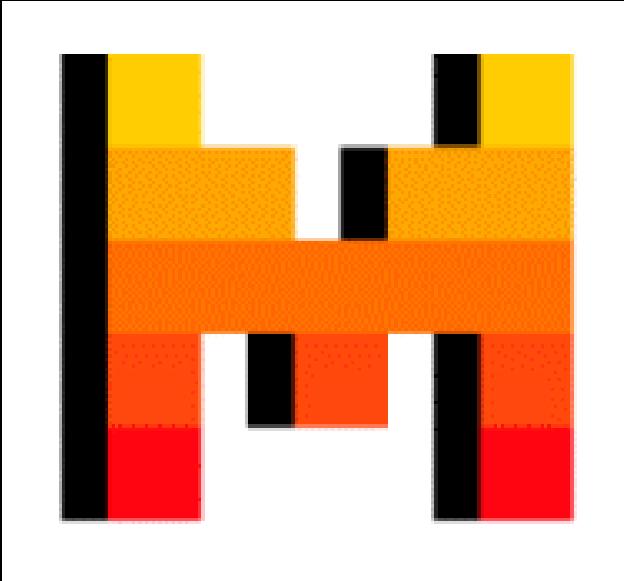


```
JS fetch_pic.js push_to

1 const fetchNASAPictureOfTheDay = () => {
2   return fetch('https://api.nasa.gov/mars/rovers/curiosity/images/latest')
3     .then(response => response.json())
4     .then(json => {
5       return json;
6     });
7 }
```



Placeholder Mistral & Codestral

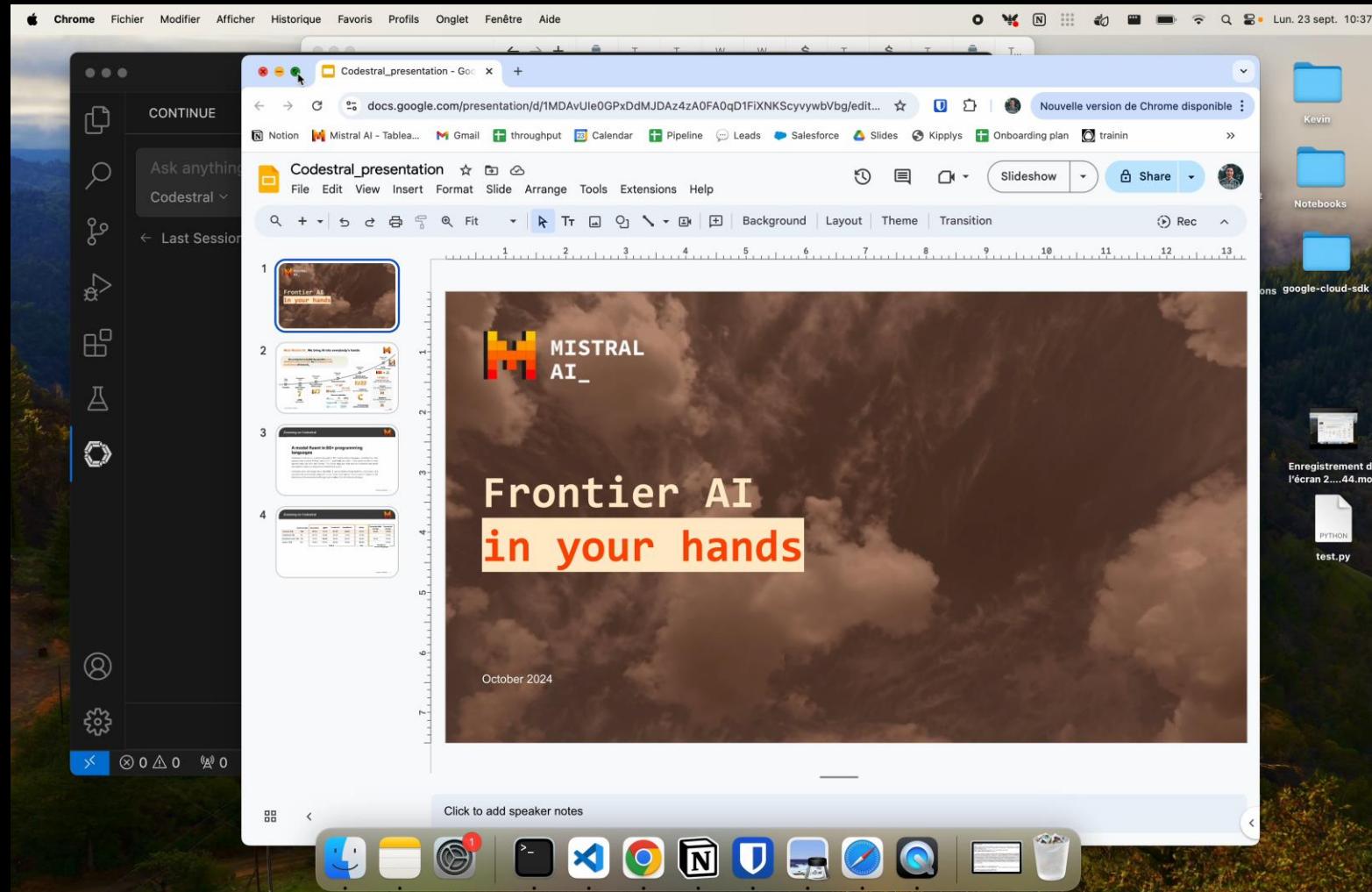


https://docs.mistral.ai/capabilities/code_generation/#integration-with-continuedev
https://docs.mistral.ai/capabilities/code_generation/
<https://mistral.ai/news/codestral/>
<http://chat.mistral.ai>

[Pricing](#) | Generative AI on Vertex AI | Google Cloud



Mistral Fallback Video



Limitations



Quality of Generated Code

While Copilot is impressive, it doesn't always produce code that adheres to best practices. [For instance, in JavaScript, it might generate `var` and `==` instead of `const` and `==`, which can lead to subtle bugs and shadowing¹.](#)



Biased Language and Libraries

Copilot tends to favor certain libraries or frameworks. [For example, in Python, it heavily leans toward using SQLAlchemy, which may not align with your preferred stack².](#)



Single Context

You can only provide one piece of context when asking Copilot a question. [This limitation can impact its ability to generate accurate code when dealing with complex scenarios².](#)

Best practices

Single, Specific, Short

- Single responsibility
- Specific prompt
- Short response

Patterns

- Regex, CRON, PowerShell

Trust but verify

Context is important

Iterate, iterate, iterate

Best Practices

1. Contextual Comments

2. Brain Dump Approach

3. Handle Connections Gracefully (Python + SQL Server)

GitHub Copilot is a powerful tool, but it's **essential** to **review** and refine the generated code to ensure it aligns with your specific requirements and best practices

Excuse: Tokenization

Transform all text into Vectors

- A very long list of integers
- Differs by algorithm

Typical numbers

- ~10-100k possible tokens
- 1 token ~ = 0.75 words
- Or 100 tokens ~ = 75 words

Typical algorithm

- Byte Pair Encoding

<https://platform.openai.com/tokenizer>

Raw Text

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens and excel at producing the next token in a sequence of tokens.

Tokens

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens and excel at producing the next token in a sequence of tokens.

Integers

```
[11505, 20185, 338, 1588, 3303, 4981, 357, 29810, 6412, 284, 355, 402, 11571, 338, 8, 1429, 2420, 1262, 16326, 11, 543, 389, 2219, 16311, 286, 3435, 1043, 287, 257, 900, 286, 2420, 13, 383, 4981, 2193, 284, 1833, 262, 13905, 6958, 1022, 777, 16326, 290, 27336, 379, 9194, 262, 1306, 11241, 287, 257, 8379, 286, 16326, 13]
```

<https://github.com/openai/tiktoken>

Limitations: Context Window

LLMs have different token limits!

- GPT-3 <= 4096 tokens for input & output combined
- LLaMA <= 4096 tokens for input & output combined
- GPT-4 <= 8192 tokens for input & output combined
- makes reasoning errors
- biases in its outputs
- lacks knowledge of events that have occurred recently
of its data cuts off (Model was trained some time
earlier)
- Does not learn from its experience

Recent LLMs significantly reduces hallucinations relative
to previous models.

GPT-4o scores 45% higher than the latest GPT-3.5 in
OpenAI's evaluations

See: [Benchmarks](#)

Choices

Amazon Anthropic [Anthropic \ Home](#)

- 100k Tokens
- Lower costs
- AWS only, US only
- Constitutional AI: Uses AI to make AI a safer place. Still jailbreaks are possible
- Base Model is Claude-2 API <https://claude.ai>

Google Gemini [Gemini – chatten und inspirieren lassen \(google.com\)](#)

- FREE!, US only
- Google only, now (23.09.23 introduces Rewards based optimizations
[Gemini-Apps: Release-Updates und Verbesserungen \(google.com\)](#)
Only add confidential data to Google Gemini that you are comfortable with Google having access to!
- Base Model is Palm-2
- Codey for code generation, <https://aistudio.google.com/>, Gemini Flash, Veo for text to Video

Poolside.AI [poolside – Building the next generation AI for software engineering](#)

- Interesting new spinoff from GitHub
- Currently AWS only service

Mistral Codestral

- European Start-Up hosted by all major cloud vendors, OSS Models
- Works very well with European language prompts

OpenAI GPT-4o

- US only with Azure worldwide availability



Code Benchmarks

Open Source LLMs are continuedly monitored by Huggingface

Azure AI Studio: Benchmarks for proprietary models

Common Benchmarks Data Sets for Code Generation are [HumanEval](#), [MBPP*](#), [CodeXGlue](#)

Many Benchmarks focus on common English language, Python, Java, JavaScript, C#, ..
Comprehensive list of Benchmark results: <https://paperswithcode.com/sota>

Common challenges are the **reproducibility** of results

*Mostly basic Python Problems

Source: [Big Code Models Leaderboard - a Hugging Face Space by bigcode](#)

LLM Benchmarks: [leobeeson/llm_benchmarks: A collection of benchmarks and datasets for evaluating LLM. \(github.com\)](https://github.com/leobeeson/llm_benchmarks)

★ Big Code Models Leaderboard

Inspired from the [Open LLM Leaderboard](#) and [Open LLM-Perf Leaderboard](#), we compare performance of base multilingual code generation models on [HumanEval](#) benchmark and [MultiPL-E](#). We also measure throughput and provide information about the models. We only compare open pre-trained multilingual code models, that people can start from as base models for their trainings.

Evaluation table Performance Plot About Submit results

See All Columns

Search for your model and press ENTER...

Filter model types: all, base, instruction-tuned, EXT external-evaluation

T	Model	Win Rate	humaneval-python	java	javascript	cpp
EXT	OpenCodeInterpreter-DS-33B	55.83	75.23	54.8	69.06	64.47
EXT	Nxcode-C0-7B-orpo	55.42	87.23	60.91	71.69	68.04
	CodeOwen1.5-7B-Chat					
EXT	CodeFuse-DeepSeek-33b					
EXT	DeepSeek-Coder-33b-instruct					
EXT	Artigenz-Coder-DS-6.7B					
EXT	DeepSeek-Coderx-7b-instruct					
EXT	OpenCodeInterpreter-DS-6.7B					
	Phind-CodeLlama-34B-v2					
	Phind-CodeLlama-34B-v1					
	Phind-CodeLlama-34B-Python-v1					
	CodeOwen1.5-7B					

Assess model performance with evaluated metrics

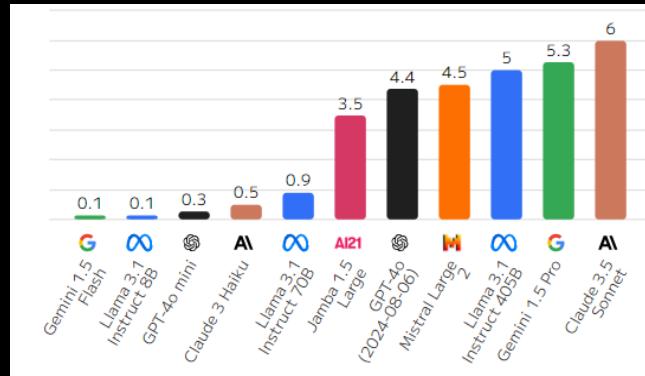
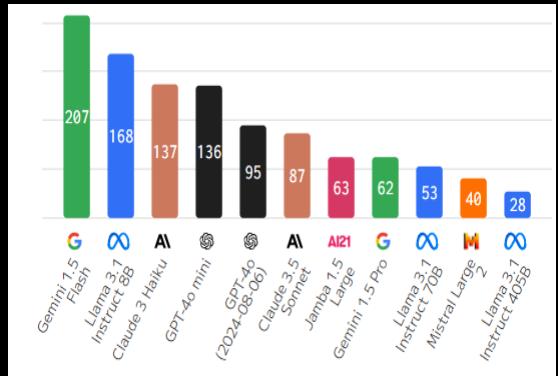
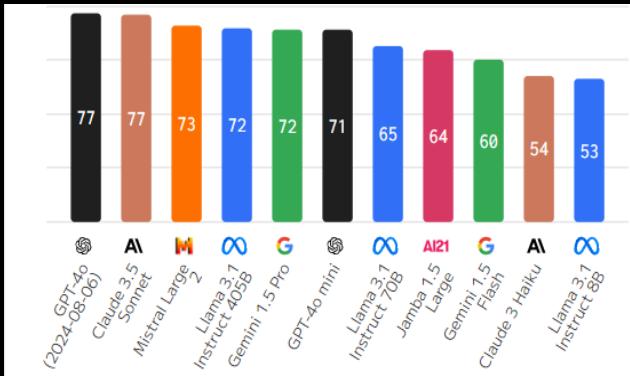
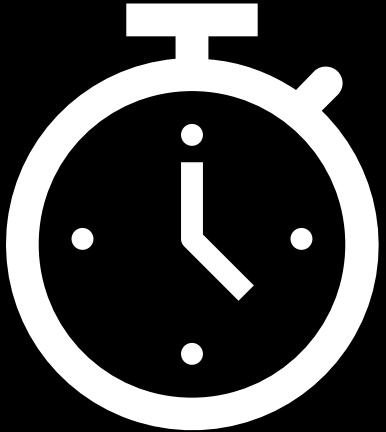
Quality benchmarks Embeddings benchmarks

Compare benchmarks across models and datasets available in the industry to assess which one meets your business scenario. Learn more about how model performance is scored

All filters Tasks: Text generation Collections Models Datasets: [human_eval](#) Metrics Clear all

Model	Model version	Dataset	Task	Accuracy	Groundedness	Fluency	Relevance	GPTSimilarity	Coherence
gpt-4o	5/13/2024	Average	--	0.921	--	--	--	--	--
	5/13/2024	human_eval	Text generation	0.921	--	--	--	--	--
gpt-4-turbo-2024-04-09	turbo-2024-0...	Average	--	0.884	--	--	--	--	--
	turbo-2024-0...	human_eval	Text generation	0.884	--	--	--	--	--
gpt-4-0125-preview	4	Average	--	0.878	--	--	--	--	--
	4	human_eval	Text generation	0.878	--	--	--	--	--
meta-llama-3.1-405b-instruc	1	Average	--	0.854	--	--	--	--	--
	1	human_eval	Text generation	0.854	--	--	--	--	--
gpt-4-0314	0314	Average	--	0.805	--	--	--	--	--
	0314	human_eval	Text generation	0.805	--	--	--	--	--
meta-llama-3.7-70b-instruct	2	Average	--	0.805	--	--	--	--	--
	2	human_eval	Text generation	0.805	--	--	--	--	--
gpt-4-0613	0613	Average	--	0.793	--	--	--	--	--
	0613	human_eval	Text generation	0.793	--	--	--	--	--
gpt-4-32k-0314	0314	Average	--	0.793	--	--	--	--	--
	0314	human_eval	Text generation	0.793	--	--	--	--	--
gpt-4-32k-0613	0613	Average	--	0.787	--	--	--	--	--
	0613	human_eval	Text generation	0.787	--	--	--	--	--
meta-llama-3.1-70b-instruct	1	Average	--	0.787	--	--	--	--	--
	1	human_eval	Text generation	0.787	--	--	--	--	--
phi-3-medium-4k-instruct	6	Average	--	0.738	--	--	--	--	--
	6	human_eval	Text generation	0.738	--	--	--	--	--

LLM Benchmarks



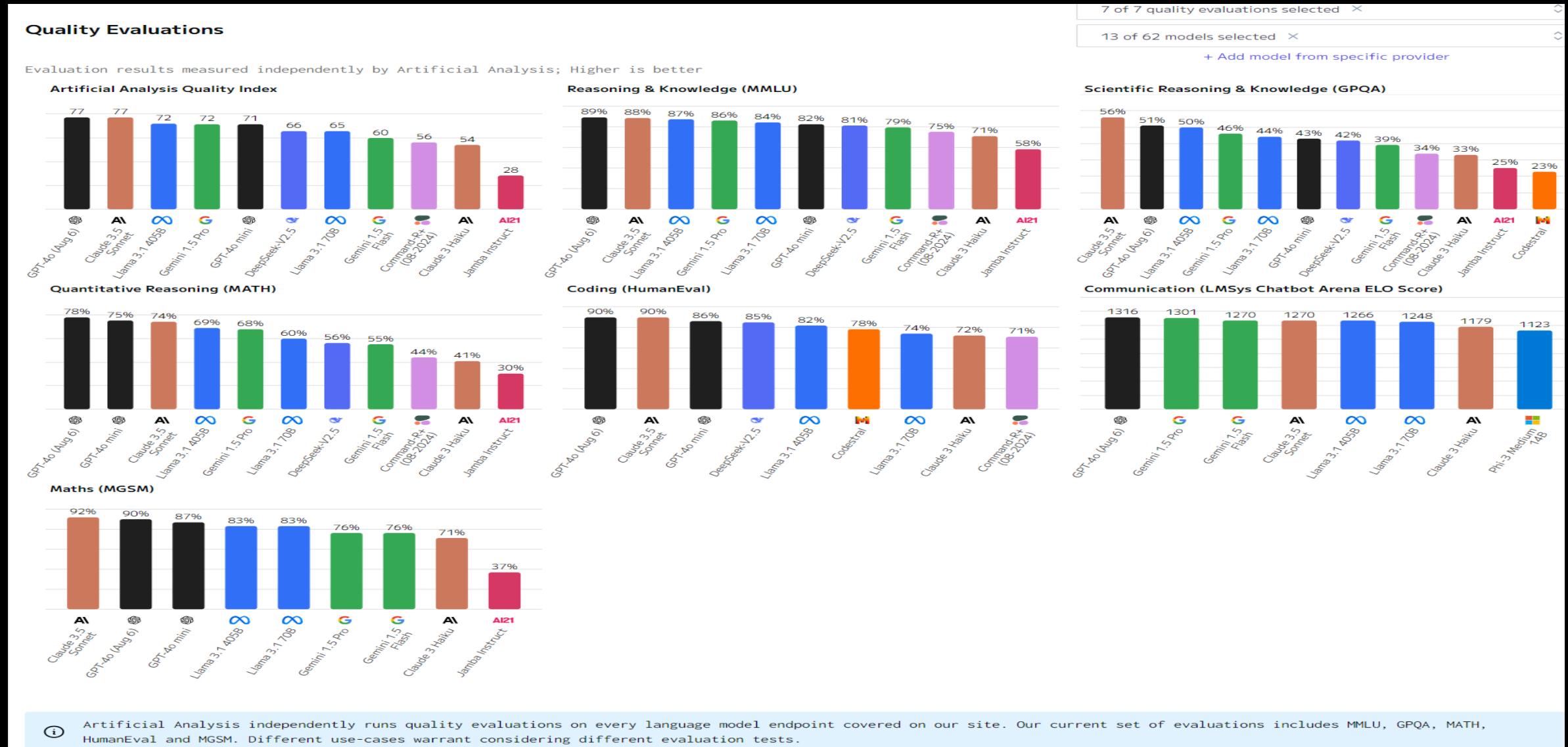
Source: [Comparison of AI Models across Quality, Performance, Price | Artificial Analysis](#)

Quality
AI Quality Index
Higher is better

Speed
Output Tokens per second
Higher is better

Price
USD per 1M token
Lower is better

Coding specialized LLM Benchmarks



APIs'

Azure AI Templates with Azure Developer CLI (2024)

For example:

- Creative Writing Assistant: Working with Agents using Prompty

Mega-Vendor Choice and AI Platform Offerings

Amazon

Bedrock

Google

Vertex-AI
Studio

Google AI-
Studio

Huggingface

Huggingface
Enterprise

Microsoft

Azure AI-Studio

Similarities & Performance

All platforms leverage their respective cloud infrastructures providing:

- Fully managed service designed for developers to streamline the development of generative AI applications.
- High-performing foundation models (FMs) from leading AI companies
- Providing reliable Performance
- Can be accessed from a single API

Identification and Authorization

- Entra-ID or Okta based Identity services
- Role-based access control (RBAC) for managing permissions

Thread Detection and Mitigation

- built-in threat detection mechanisms to identify and mitigate risks

Requirements?

Mega-Vendor AI Platform differences

Amazon

Bedrock

Partners with AI21 Labs, Hugging Face, Anthropic, and Stability AI Foundation Models like Titan and Falcon AI. Offers a wider range of models through these partnerships.

Seamlessly integrates with other AWS services, streamlining development.

Google

Vertex AI Studio

Google LLM models (PaliGemma, Gemini, Imagen2, Chirp, PaLM 2, **Codey**, Claude 3) Allows customization and fine-tuning of models Industry specific model provided Partners with Anthropic, Meta,

Offers an extensive API ecosystem, allowing integration with Azure services and external applications. Monitoring and logging via BigQuery Data Labeling services (humans in the loop) Prompt In-/Output filtering

Hugging Face

Hugging Face Enterprise

Provides a rich set of open source and community build LLMs. Provide largest set of models in the industry Rich set of data sets for training exercise Wide community support Has weak security concept for LLMs Cheap due to missing enterprise capabilities

Microsoft

Azure AI-Studio

Rich set of LLMs ~1700. Focuses on OpenAI's GPT series (GPT-4o, GPT-4-Turbo, and Embeddings). Allows customization and very efficient fine-tuning of models Partners with Meta, Mistral, Huggingface, OpenAI

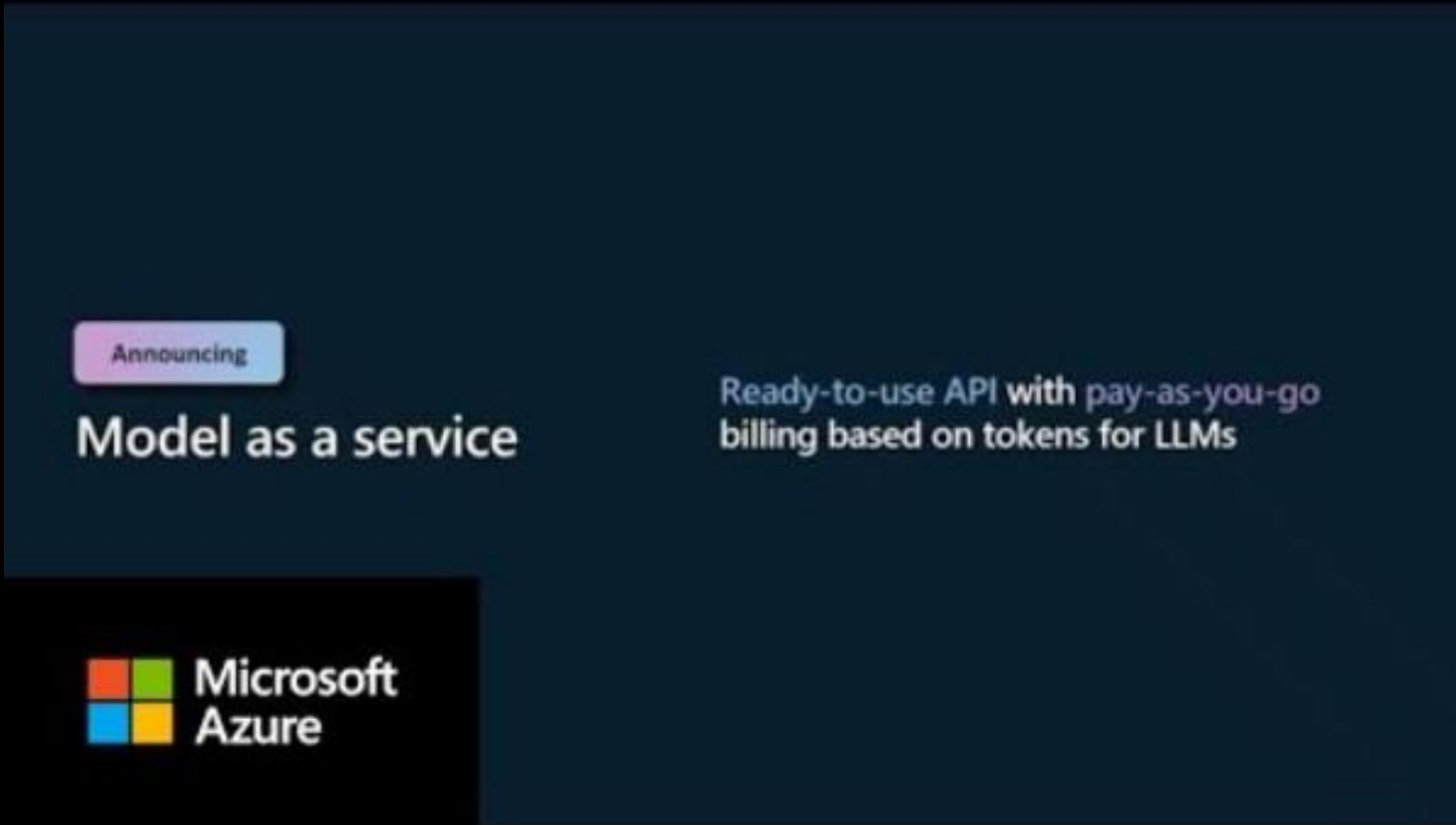
Offers an extensive API ecosystem, allowing integration with Azure services and external applications. Supports Model security (fingerprinting), debugging, benchmarking and on-demand inferencing

Functionality by Service

	Azure AI-Studio	Amazon Bedrock	Google Vertex-AI	Hugging Face
Pricing	Pay-as-you-Go Provisioned Throughput Units (PTU)	Pay-as-you-Go Amazon-Bedrock-Pricing-AWS	Pay-as-you-Go Pricing Vertex AI Google Cloud	\$20 USD per user per month Inference starts at \$0.032/h Hugging Face – Pricing
# Regions supported	US, EU, Asia (60+ Locations),	US, EU, Asia	US, EU, Asia, SA, MEA	US, EU partly
# of Models (Multi-modal)	147 prop/ 1520 OSS	6 pro / no OSS	40 prop / 110 OSS	688.461 OSS
Playground	✓	✓	✓	✓
Monitoring	✓	✓	✓	✓
Benchmarking	✓	🚫	🚫	✓
On-demand Training/inferencing	✓	🚫	🚫	✓
Retrieval-Augmented Generation (RAG)	✓	✓	✓	✓
Policy management	✓	🚫	✓	🚫
Audit logs	✓	✓	✓	✓
Debugging	✓	🚫	🚫	🚫
IP-Protection / CMK / XPIA	✓	🚫	🚫	🚫
Assistant API's	✓	🚫	✓	🚫

Model-as-a-Service

via Azure AI-Studio



An image showing a dark blue announcement card. In the top left corner, there is a light blue button-like shape containing the word "Announcing". Below this, the text "Model as a service" is displayed in white. To the right of the text, there is a description: "Ready-to-use API with pay-as-you-go billing based on tokens for LLMs". At the bottom left, the Microsoft Azure logo is visible, consisting of four colored squares (red, green, blue, yellow) followed by the text "Microsoft Azure".

Announcing

Model as a service

Ready-to-use API with pay-as-you-go
billing based on tokens for LLMs

 Microsoft
Azure

Pricing

Comparison of price per token across common LLM-Models

Tokens cost money and their efficiency varies widely across models

OpenAI is leading on English and Python, but Gemini in non-English content

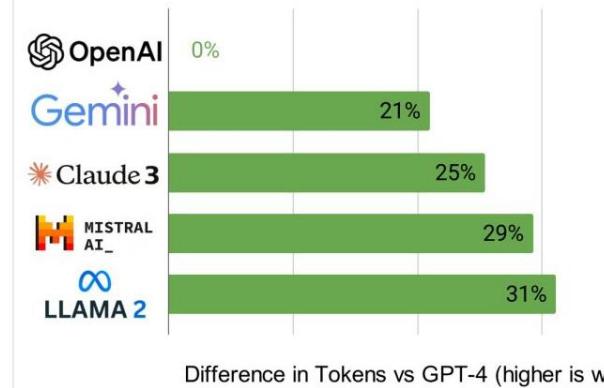
OpenAI: [Pricing | OpenAI](#)

Source: <https://www.linkedin.com/in/peter-gostev/>

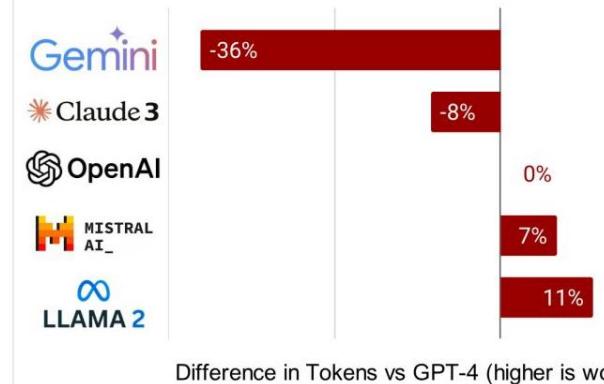
English Text



Python Code



Non-English Text (French, Chinese, Hebrew)



- GPT-4 & 3.5 is the most efficient for English text

- Claude 3 users will pay c.10% more

- GPT tokenizer has been specifically optimised for Python

- Saving 20-30% of tokens vs other models

- Google Gemini models are far more efficient at other languages

- French: 11% better
- Chinese: 43% better
- Hebrew: 54% better than GPT

General GitHub Pricing

- **GitHub Offerings at Microsoft Product Terms and GitHub's plans - GitHub Docs**
- **For GitHub Copilot for Business an Agreement must be in place:**
 - **Enterprise Agreement** [Enterprise Agreement | Microsoft Volume Licensing](#)
 - **Microsoft Customer Agreement** [Licensing Documents \(microsoft.com\)](#)
- GitHub Copilot Business, \$19 USD per user per month.
- GitHub Copilot Enterprise, \$39 USD per user per month.
- A connected Azure subscription by your organization or enterprise account will enable metered billing via Azure
- [About billing for GitHub Copilot - GitHub Docs](#)
- Upgrading from Business to [Enterprise](#) will upgrade all seats!
- GitHub Advanced Security licenses [About billing for GitHub Advanced Security - GitHub Docs](#)



GitHub Enterprise Licensing

Different Models per seat or metered apply

Unit Licensing applies to

- **GitHub Enterprise** (per seat)
- **GitHub Advanced Security**
 - (per active committer last 90day)
 - Subscription discount with annual billing
 - Substantial discount with central GitHub Enterprise environment (only)
- **Billing: Seat based** licensing, quarterly billed

Metered Licensing applies to

- Applied to GitHub **Copilot***
- **Assigned/Unassigned** by GitHub Org Owner ("in-app")
- **Billing:** Azure Subscription based, monthly billing on **Enterprise Account level****

Metered Consumption applied to

- **GitHub Actions**
- Packages
- CodeSpaces
- Storage for Actions and Packages
- Large File Storage (Git LFS)
- **Billing:** Azure Subscription based, monthly billing on **Enterprise Account level****

GitHub Enterprise Pricing for Bertelsmann

BCP GitHub Enterprise Billing Model benefits from large user base, Azure consumption discount associated with central Microsoft tenant/enrollment

Unit Licensing

- **Microsoft Discount Level D**
- **Pricing 2023-2026 (by SKU)**
- Copilot Business (**PEY-00002**) €14,96 /user/month 1)
- Actions (**QEJ-00005**) ~ €0,12 /user/month 2)
- CodeSpaces (**QEK-00004**) ~ €0,12 /user/month 3)
- Copilot Enterprise (TWT-00002) €34,86 /user/month 4)
- Licenses cannot be reserved, need to be purchased via reseller by mailto:SAM@bertelsmann.com

Metered Licensing

- 21% discount for these services (BGROUP Azure consumption discount) 5)
- **In case of GitHub Copilot for Business:**
- ~ **14,29 /user/month**, excluding internal service fees

Metered Consumption

- **21% discount** for these services (**BGROUP** Azure consumption discount) 5)

Prompt Engineering

In-context Learning: Integrate examples into prompts, enable LLMs to handle new tasks without fine-tuning

Zero-shot - Predicting with no sample provided



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1	Translate English to French:	<----- task description
2	Cheese =>	<----- prompt

One-shot - Predicting with one sample provided



One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

1	Translate English to French:	<----- task description
2	Sea otter => loutre de mer	<----- example
3	Cheese =>	<----- prompt

Few-shot – Predicting with a few samples provided



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed

1	Translate English to French:	<----- task description
2	Sea otter => loutre de mer	<----- examples
3	Peppermint => menthe poivre	<-----
4	Plush giraffe => girafe peluche	<-----
5	Cheese =>	<----- prompt

Source: [Prompt engineering - Wikipedia](#)

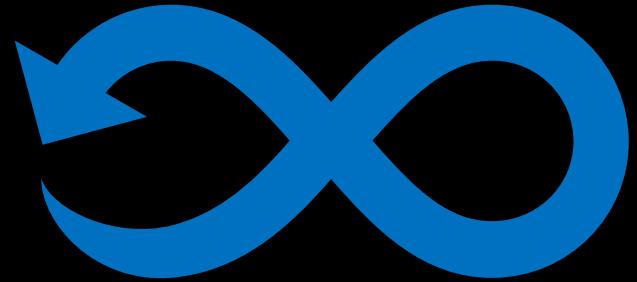
Effective Prompts for optimal Results

Clarity and Precision	Provide Context	Open Questions	Structured Requests	Provide Examples	Define the Goal	Seek Feedback	Experiment with Variations
<ul style="list-style-type: none">Formulate your questions or instructions clearly and precisely. Avoid ambiguous terms.	<ul style="list-style-type: none">Give enough background information so that the context is understood. This helps in obtaining more relevant answers.	<ul style="list-style-type: none">Ask open-ended questions that allow for detailed answers, rather than yes/no questions.	<ul style="list-style-type: none">Use lists or numbered points to clearly outline your requirements. For example: "Name three advantages of..."	<ul style="list-style-type: none">If you are looking for specific information, provide examples to clarify what you mean.	<ul style="list-style-type: none">Explain what goal you are pursuing with your request. This helps to tailor the answer accordingly.	<ul style="list-style-type: none">If you do not receive the desired results, reflect on your prompt and adjust it.	<ul style="list-style-type: none">Try different formulations and approaches to find out which works best.

Improve LLM response quality principals

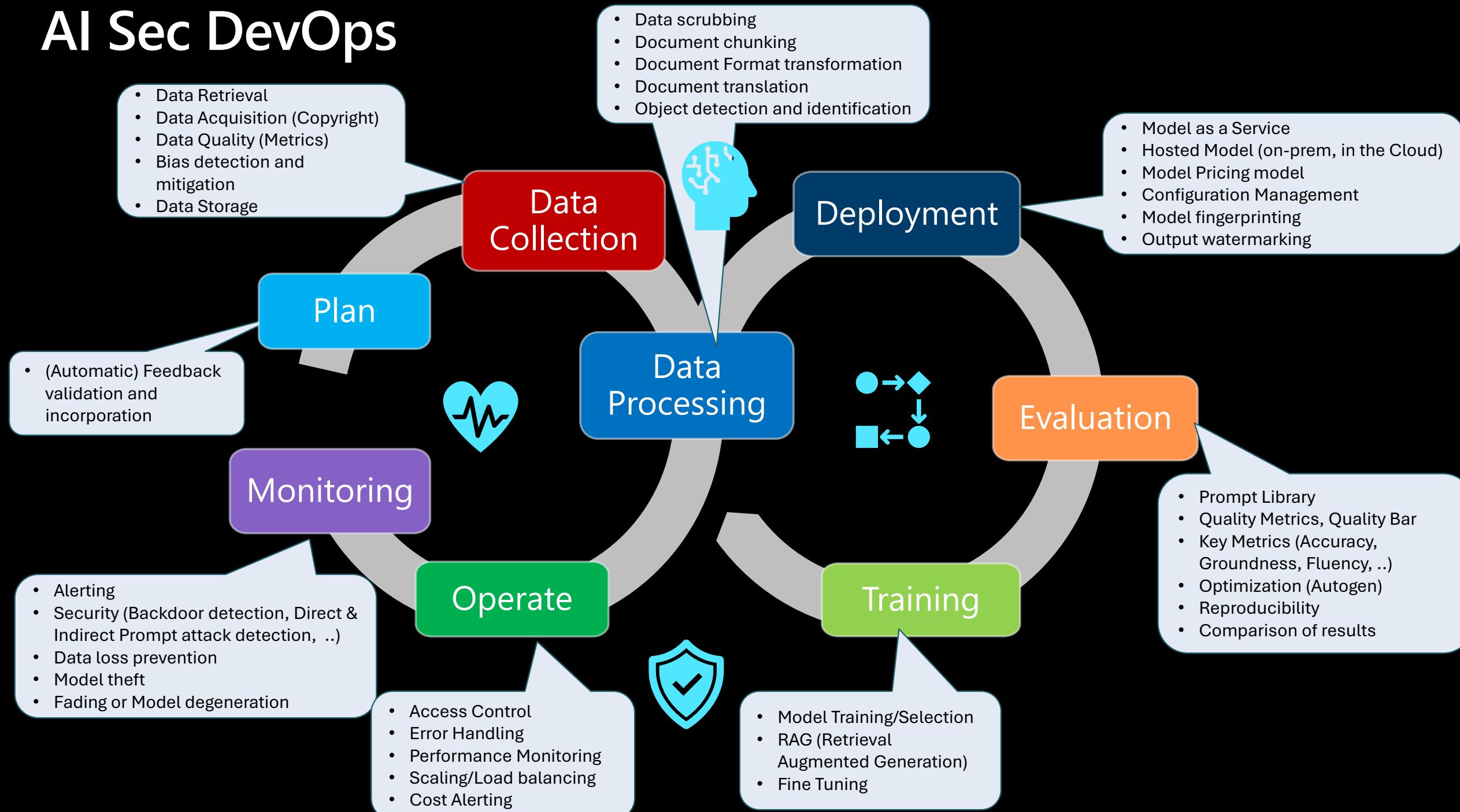
No.	Prompt Principle for Instructions
1	No need to be polite with LLM so there is no need for phrases like "please," "if you don't mind," "thank you," "I would like to" etc., and get straight to the point.
2	Integrate the intended audience in the prompt e.g., students in an expert level conversation.
3	Break down complex tasks into a sequence of simpler steps or pieces of information in the field. (Chain of thought)
4	When prompts: include clear instructions of where the audience is at in narrative context. When you need clarity or a deeper understanding of a topic, idea, or any piece of information, utilize the following format: <ul style="list-style-type: none">- Explain [insert specific topic] in simple terms.- Explain to me like I'm 11 years old.- Explain to me like I'm 5 years old. (in field).
6	Write [the essay/text/paragraph] using simple English like you're explaining something to a 5-year-old.
7	Add "I'm going to fix Sxxx for a better solution."
8	When formatting your prompt, start with [#Instruction#], followed by either [#Example#] or [#Example+explanation-driven prompting (use show-me prompt)#].
9	Integrate example-driven prompting with [#Instruction#], followed by either [#Example#] or [#Question+#+]. If relevant: Subsequently use context cues and more.
10	Incorporate the following phrases: "Your task is", "You MUST".

Prompt Engineering & DevOps



- Prompt Library ([The Big Prompt Library](#))
- [Prompty](#)
- [Promptflow](#)
- [Langchain](#)
- [Semantic-Kernel](#)
- [What's New in AutoGen? | AutoGen \(microsoft.github.io\)](#)

AI Sec DevOps



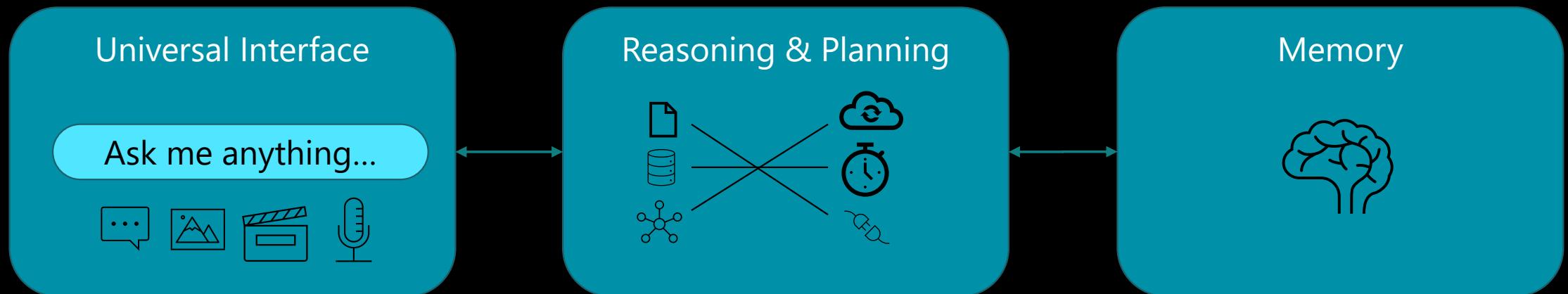
Azure AI Studio

Bring generative AI applications into production

With open AIOps in Azure AI Studio

**GenAIOps with
GitHub & Azure AI
Studio**

The next step - Copilots + Agents



Next – Small Models (SLMs)

Phi-3 – Family of „small“ language models

Small language model, Best in class performance for size, with frictionless availability

Φ

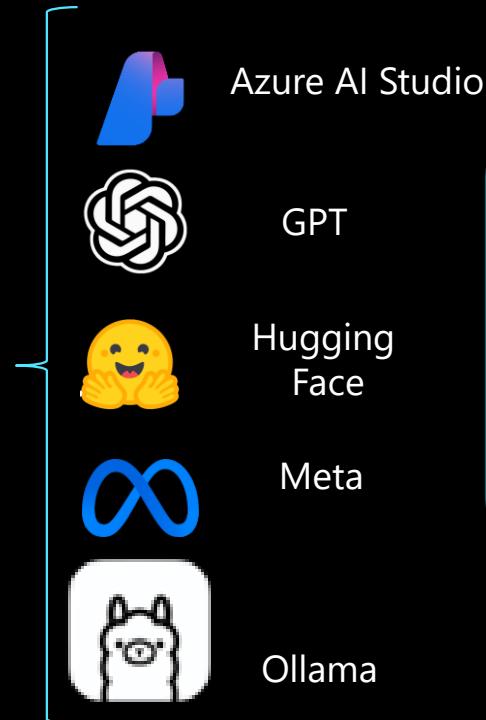
Phi-3-mini-V
(3.8B + 0.3B)

Φ

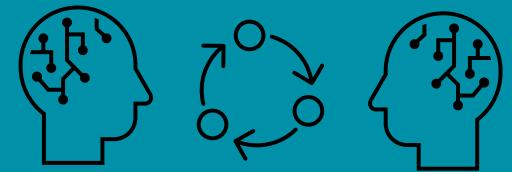
Phi-3- small
(7B)

Φ

Phi-3-mini
(3.8B)



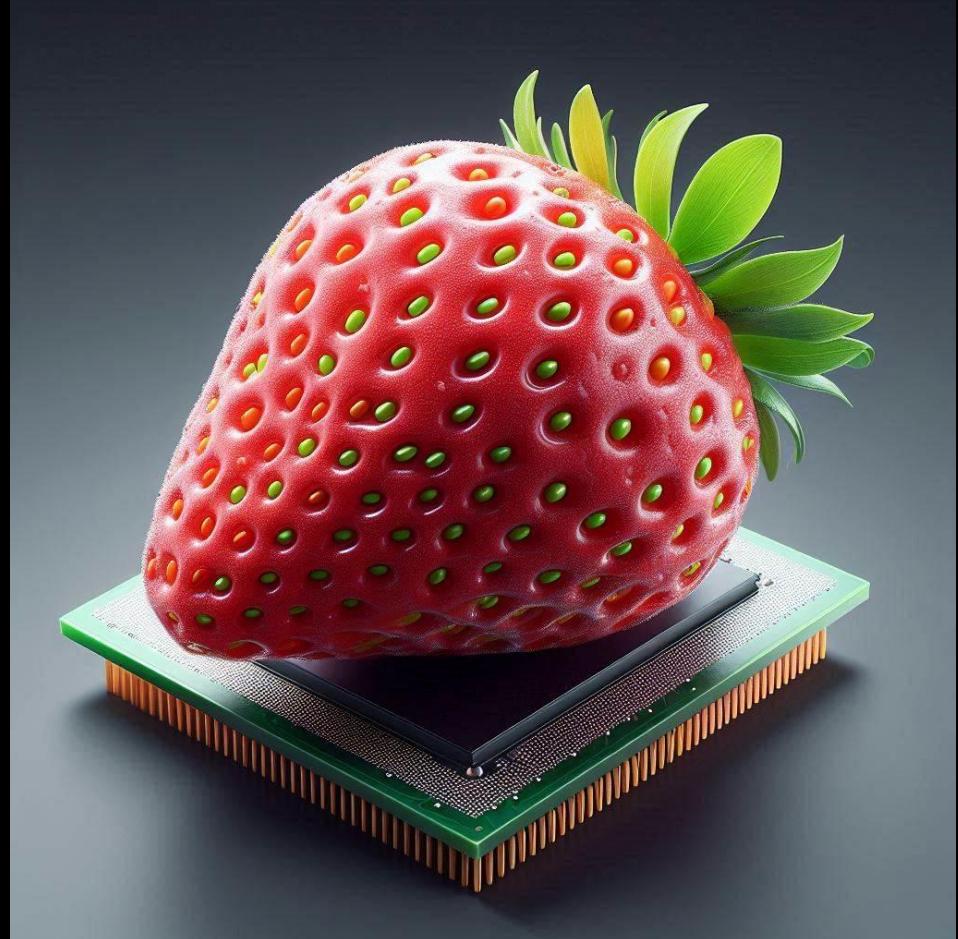
Multi-Agent AI



Project Strawberry aka OpenAI-o1

Project „Orion“ soon called GTP-5 comming this year

- OpenAI announced PhD-Level intelligence
- Reasoning on complex topics is a game changer
- OpenAI o1—a new series of AI models designed to autonomously leverage Chain of Thought (CoT) prompts, before they respond.
- This models can reason through complex tasks and solve harder problems than previous models in science, coding, and math.



GTP-4o Tools: Memory & Security

The screenshot shows the Chat playground interface with the following details:

- Setup:** Deployment selected is "gpt-4o (version:2024-08-06)".
- System message:** Add your data Parameters
- Give the model instructions and context:** You are an AI assistant that helps people find information.
Tools
bio
The 'bio' tool allows you to persist information across conversations. Address your message 'to=bio' and write whatever information you want to remember. The information will appear in the model set context below in future conversations.
- Add section:** + Add section
- Chat History:**
 - remember: I love fast bikes
 - to=bio User loves fast bikes.
 - to=bio User loves cats.
 - Here's what I have so far:
 - User loves fast bikes.
 - User loves cats.
 - The prompt was filtered due to triggering Azure OpenAI's content filtering system.
Reason: This prompt contains content flagged as **Jailbreak**
- Type user query here:** Type user query here. (Shift + Enter for new line)
- Tokens:** 171/128000 tokens to be sent

Source: TheBigPromptLibrary/Articles/chatgpt-bio-tool-and-memory/chatgpt-bio-and-memory.md at main

Summary of predictions

There will be a change in the way Developers and DevOps will work

Productivity will increase by an order of magnitude

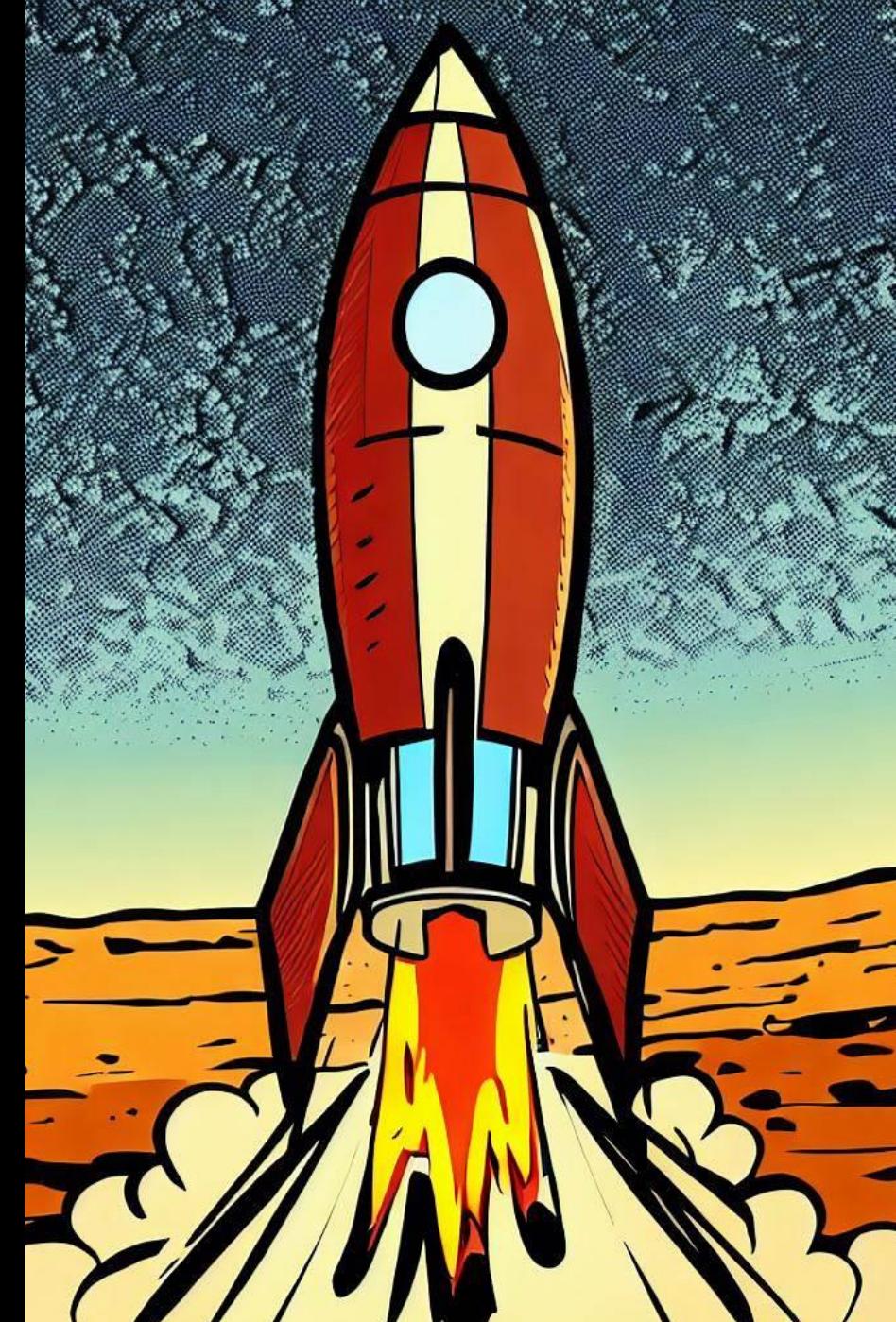
Learning new stuff will become much easier

Tackling complex challenges will get easier but at the edge still need expertise

Attacks will come in new shapes and unexpected forms

Creativity will boost

Learning the basics will be relevant for a long time to come



Next Steps

Learning

- Emphasizing continuous learning and upskilling in a fast-changing environment is key to able to push boundaries

Discovering

- Encourage experimenting to uncover what really works and strive for meaningful impact

Deploying

- Applying models to real-life and create value.
Monitor and finetune, learn and repeat the process.

Ressources

Todays Presentation

<https://github.com/ralphke/AI-Pair-Programmer>

Prompt Engineering

[ChatGPT Prompt Engineering for Developers - DeepLearning.AI](#)

[prompt engineering: Onlinekurse, Trainings und Tutorials auf LinkedIn Learning](#)

[Mega-Prompts: Turning Expertise into Code | by Chia Jeng Yang | Messy Problems | Medium](#)

[Prompt Engineer werden – so geht's \(computerwoche.de\)](#)

[WikipediA Prompt engineering](#)

DevOps Copilot

[Azure-Samples/azure-devops-copilot-extension: An Azure DevOps Copilot Extension Sample \(github.com\)](#)

DevOps Thread Matrix

[DevOps threat matrix | Microsoft Security Blog](#)

Learning Resources

- Azure OpenAI Documentation: [Azure OpenAI - Documentation, quickstarts, API reference - Azure Cognitive Services | Microsoft Learn](#)
- Business Process Automation: [Azure/business-process-automation: Business process automation solution accelerator using Azure services \(github.com\)](#)
- Azure OpenAI model family: <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/models>
- OpenAI cookbook: <https://crfm.stanford.edu/workshop.html>
- OpenAI Examples: [Examples - OpenAI API](#)
- GPT3: <https://arxiv.org/pdf/2005.14165.pdf>
- GPT3.5: <https://techcrunch.com/2022/12/01/while-anticipation-builds-for-gpt-4-openai-quietly-releases-gpt-3-5/>
- Transformers and self-attention: https://www.youtube.com/watch?v=5vcj8kSwBCY&ab_channel=StanfordOnline
- Foundation models: <https://arxiv.org/pdf/2108.07258.pdf>



Thank you!

