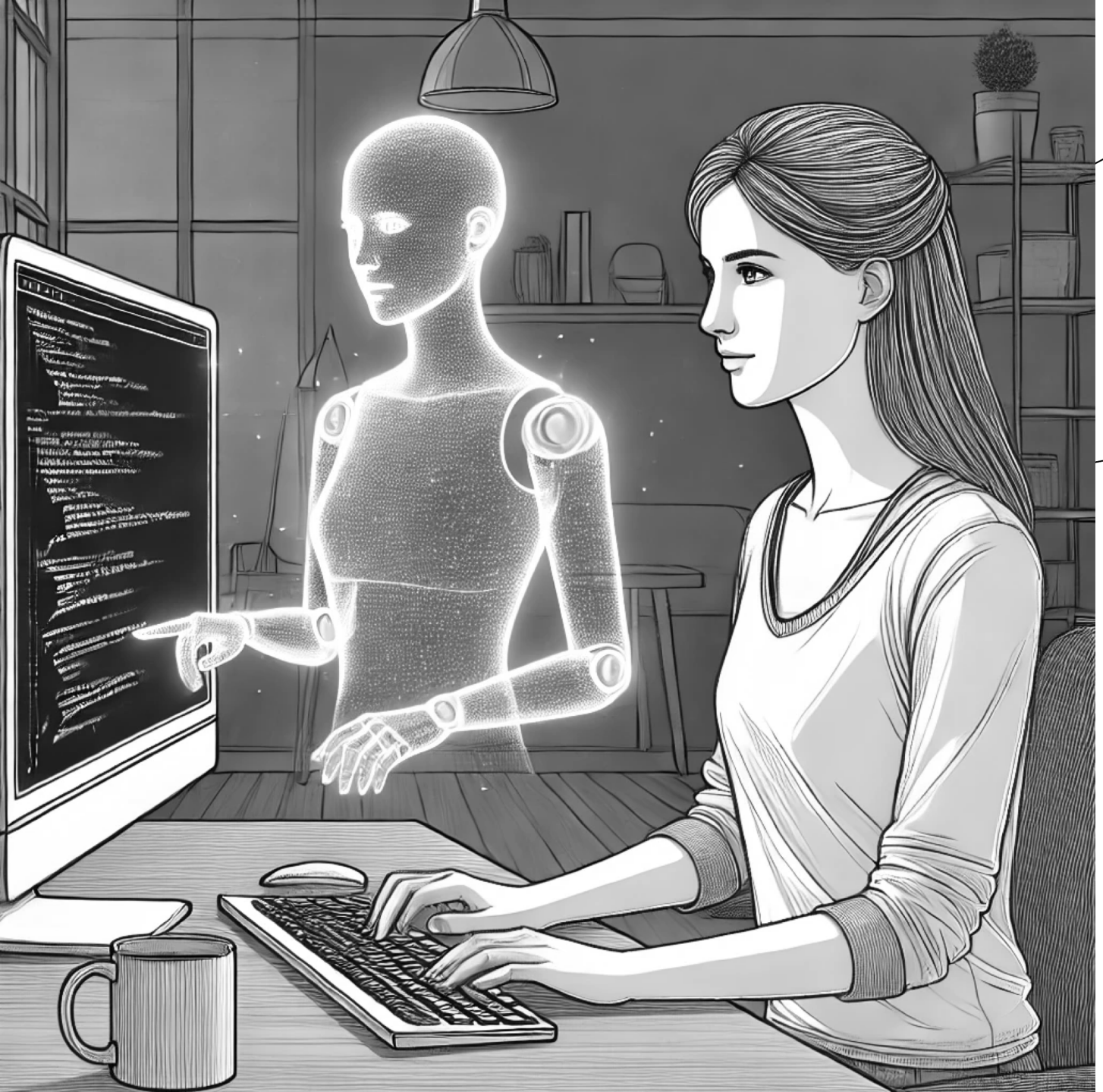


Join at slido.com
#3123011

CODING WITH AI

Thomas Vuillaume, 06/12/2024

Join at slido.com
#3123011

Est-ce que vous programmez ? / Do you code ?

ⓘ Start presenting to display the poll results on this slide.

slido

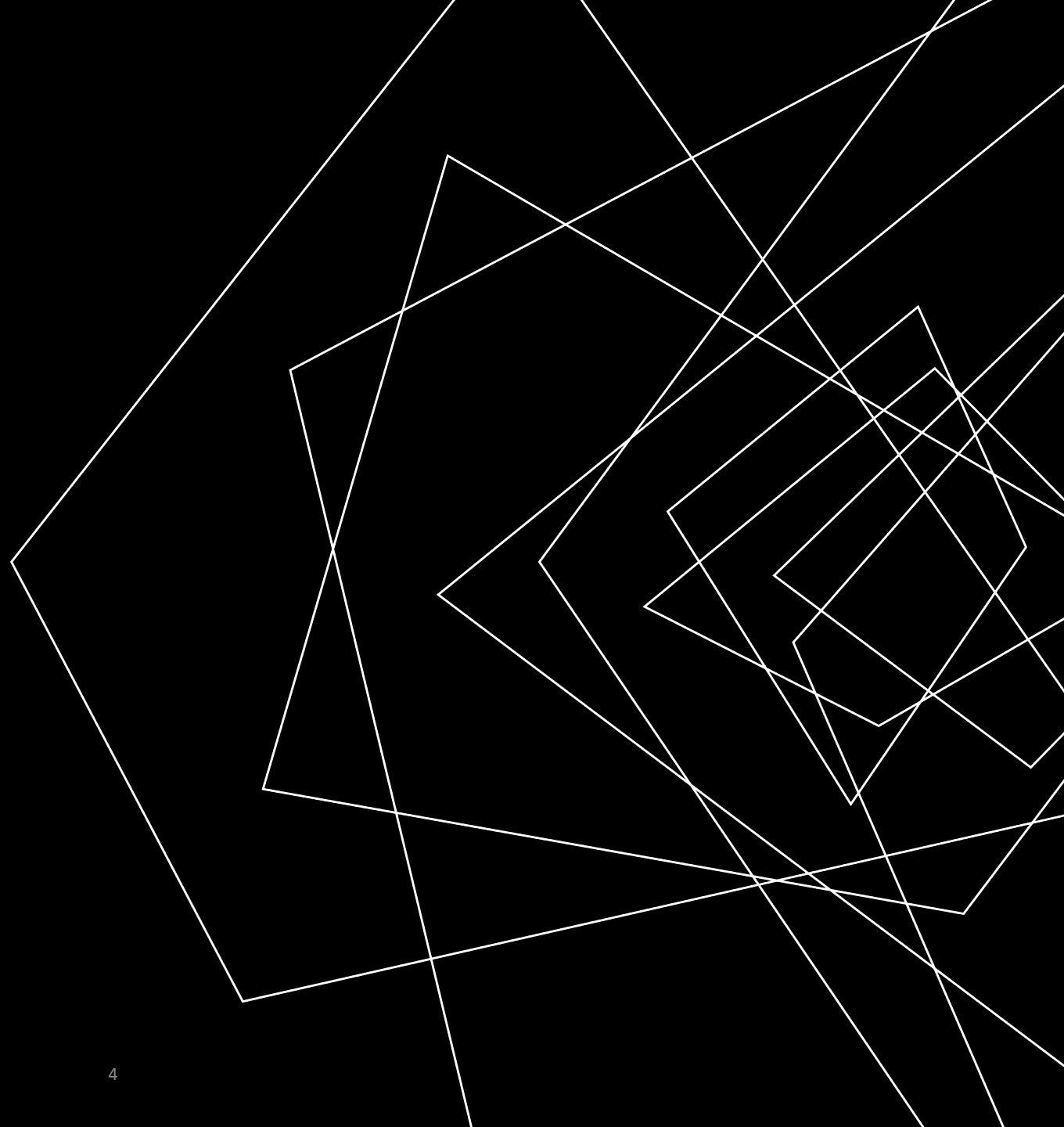
Please download and install the Slido app on all computers you use

Est-ce que vous avez déjà utilisé un assistant de code (si vous avez répondu oui à la question précédente) ? / Have you ever used a coding assistant ?

ⓘ Start presenting to display the poll results on this slide.

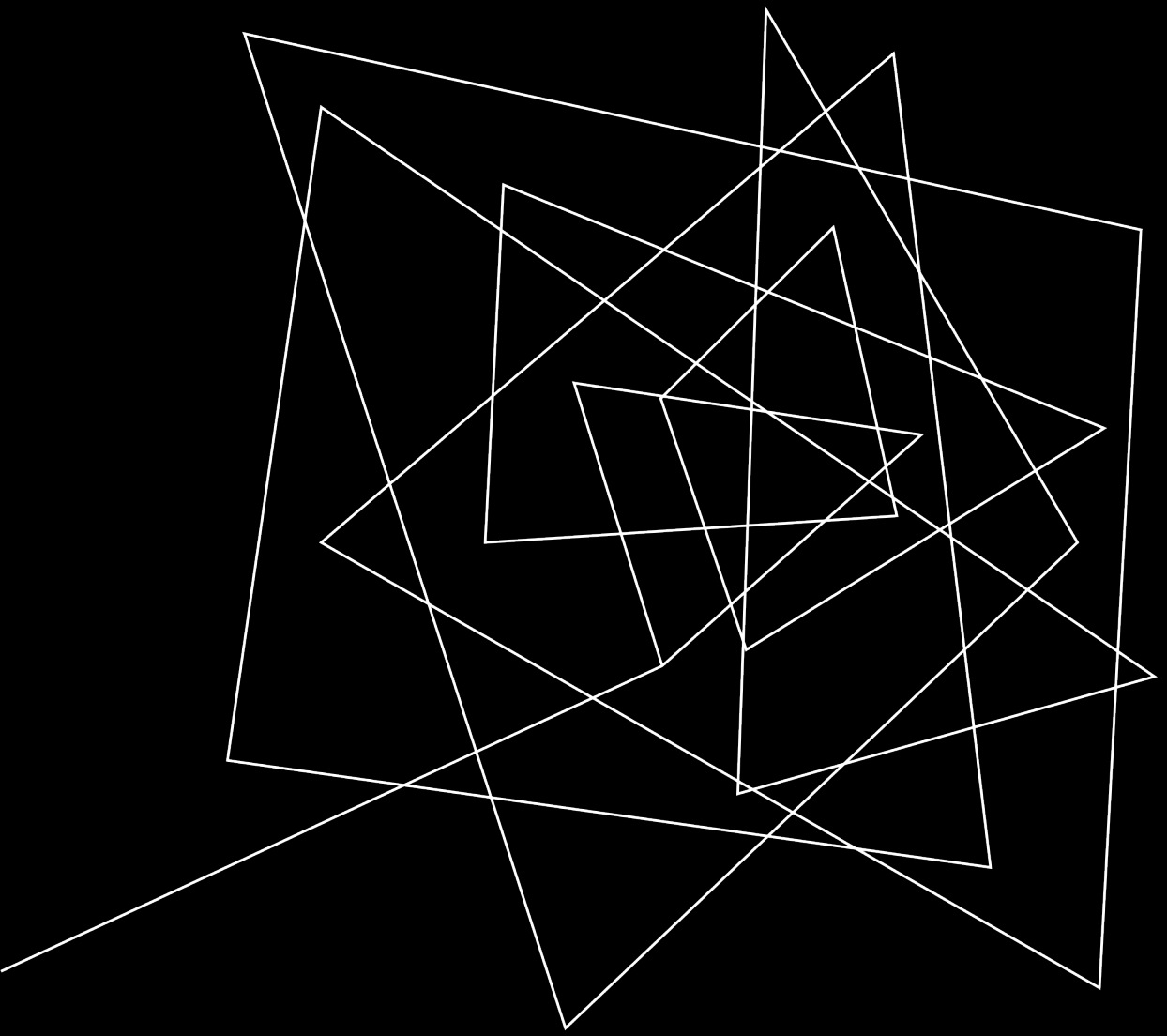
DISCLAIMERS

- I don’t consider myself an expert
- I will not go into the details of how models are trained
- I will focus on AI for coding tasks
- I don’t intend to be exhaustive
- I’d like to show you some tools and open a discussion



PLAN

1. Introduction to LLMs and their application as coding assistants
2. Overview of some AI coding assistants
3. Demo
4. New and future models - what to expect
5. Some considerations
6. Discussion



LARGE LANGUAGE MODELS

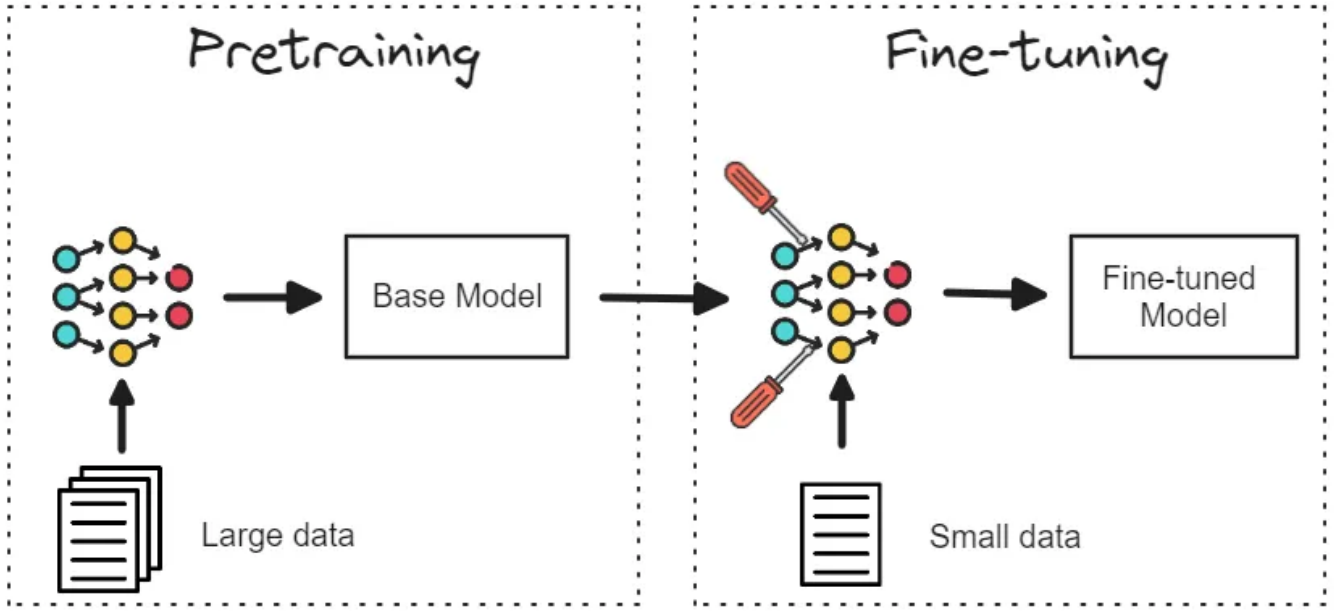
LARGE LANGUAGE MODELS

LLMs are advanced artificial intelligence (AI) models trained on massive amounts of text data to understand, generate, and predict human-like text.

. . .

*Some examples of companies or services you may know using LLMs

LARGE LANGUAGE MODELS



They are first trained to fill out hidden parts in texts

➔ foundation models

➔ very generic, can manipulate language

Then they are fine-tuned and aligned to specific tasks

➔ chatbots

➔ translators

➔ **coding assistants**

Fig: Prasad Mahamulkar, Fine-Tune Large Language Model in a Colab Notebook

LLM$_S$ AND CODE

- Code generation → *« I would like a function to … »*
- Debugging assistance → *« I have this error with this function: … »*
- Code explanation → *« What does this function do? … »*
- Generate dummy / lookup data → *« Generate dummy data to test this function »* or *« Generate a data sample corresponding to the following schema »*
- Use your own language → *« Explique cette fonction en français »*
- Tests generation → *« Generate unit tests for this function »*
- RegEX generation → *« What is the regex to find files of the form ‘{date_isoformat}_{number between 0 and 10}_epic_project_{version}.txt’ »*
- Translate code to other languages → *« Rewrite this function in JAVA »*
- … ?

OVERVIEW OF LLM FOR CODING TASKS (NOT AN EXHAUSTIVE LIST)

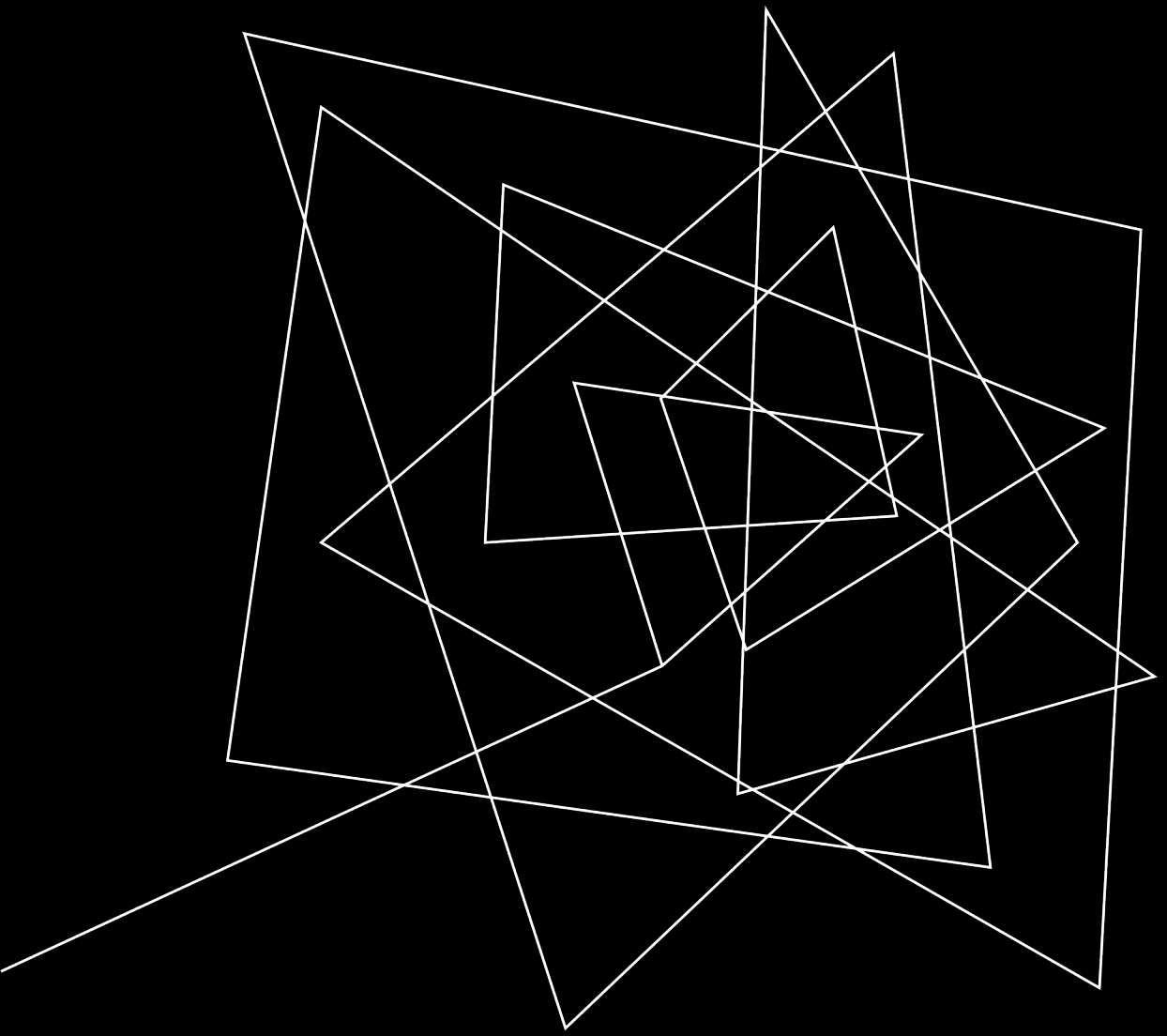
	General chatbots	Coding assistants[1]
Closed	• GPT-4 / O1 • Claude 3.5 Haiku/Sonnet • Gemini Flash/Advanced	• GitHub Copilot • Gemini code Assist • Tabnine models
Open weights	• Mistral large 🇫🇷 • llama models	• Codestral 🇫🇷 • CodeLlama • DeepSeek Coder • Qwen2.5-coder • codegemma
Open-source[2]	• BLOOM 🇫🇷 • MAP-NEO • OLMo	• StarCoder

1: fine-tuned for coding tasks

2: code, training data, training recipes, checkpoints

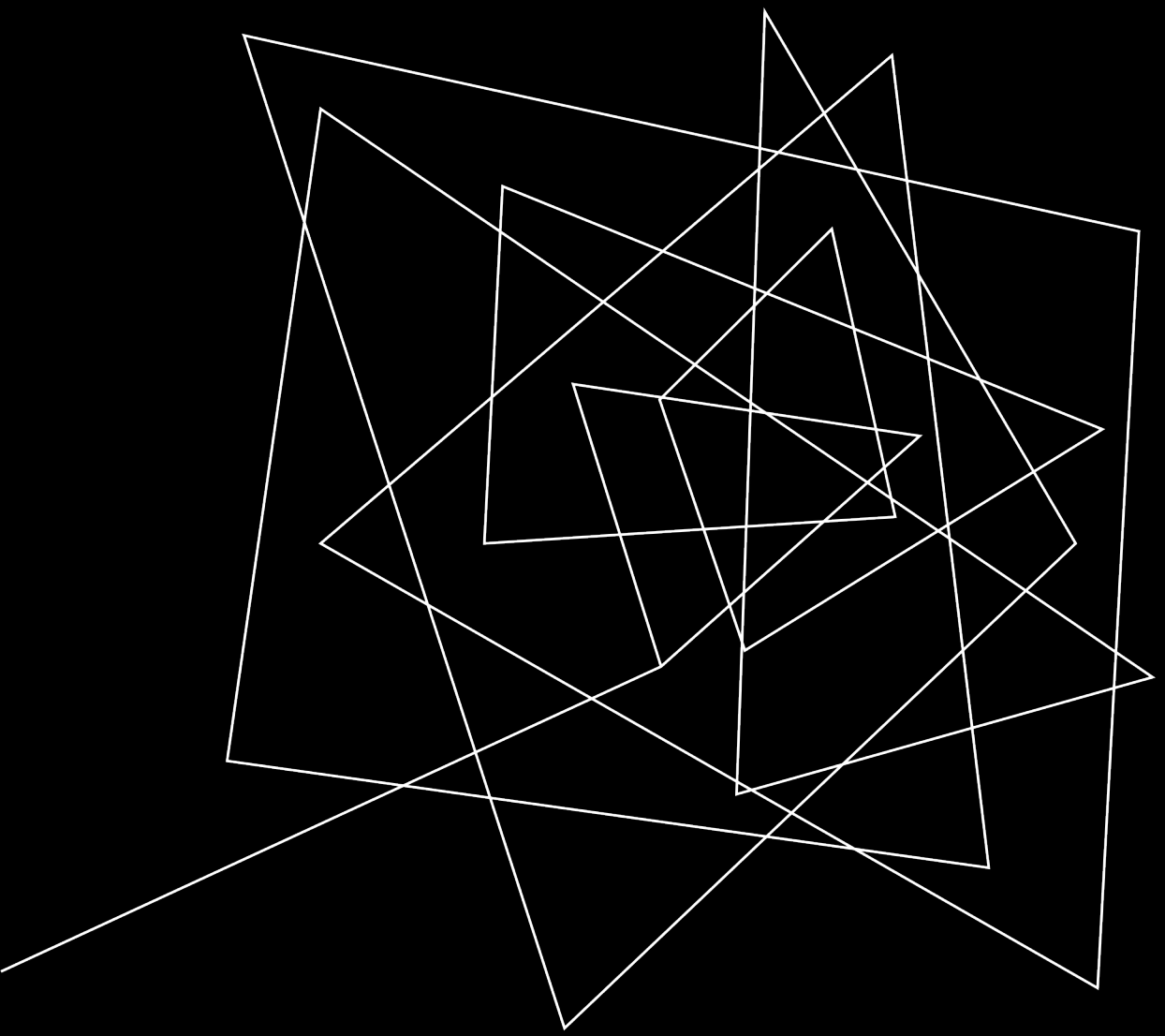
SERVICES FOR CODING YOU CAN USE *FOR FREE, AS OF 06/12/2024 (BEYOND THE ONLINE CHATBOTS YOU KNOW…)*

	provider	LLM	model	comment
GitHub copilot CLI	GitHub	GPT-4	closed, paid for, but free for students and teachers	command line help in terminal
GitHub copilot extension				IDE extension
GitHub code review				code review, integrated to GitHub
continue	continue.dev	Any	open-source	IDE extension
Google IDX	Google	Gemini	closed	Online IDE with AI assistant
tabnine	Tabnine	Tabnine et al	Closed, freeium	IDE extension
Cursor	Cursor	GPT-4, Claude 3.5, cursor-small	closed	VSCode fork with AI assistant
Windsurf	Codeium	llama 3.1 70B (for free users)	Closed, freemium, no data sharing	VSCode fork with AI assistant
codeium chat				per library chat
codeium forge				code review
ISDM chat	**ISDM mesocentre**	**mixtral 8x7b, Codestral 22B**	**free through edugain, no data sharing**	**online chatbot**
ISDM API				**to use with continue**



DEMO

- GitHub copilot command line
- GitHub copilot in VSCode
- WindSurf



SOME REFLEXIONS

THESE WERE (VERY) SIMPLE EXAMPLES

These demos can be very impressive.

Let’s not forget that they are **simple demos**, on **simple examples** (there are thousands of replicates of the weather app on the internet to copy from).

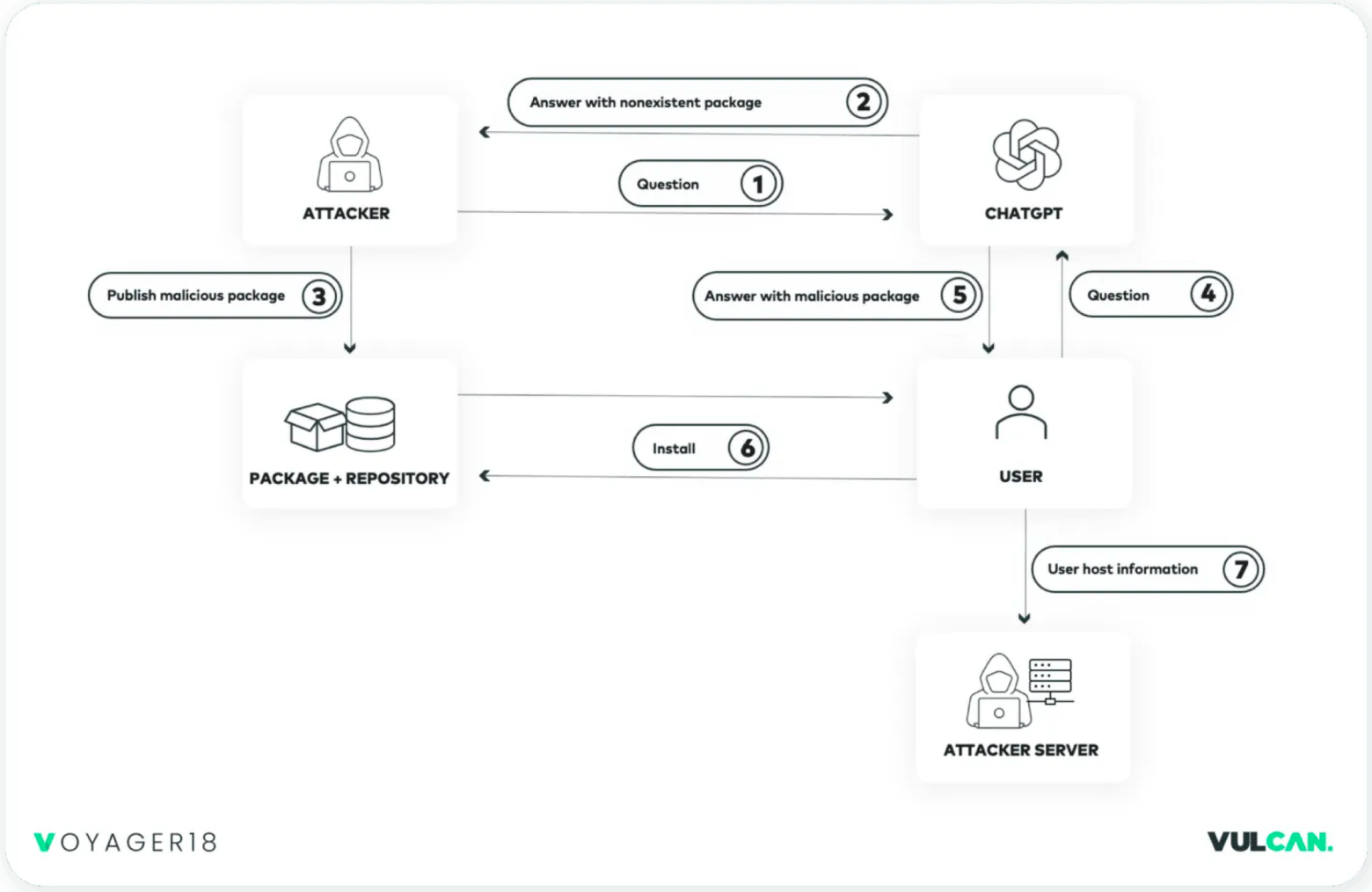
AI will also occasionnaly generate terrible & buggy code.

IN MY DEMO, WHAT DID I DO HORRIBLY WRONG?

- Auto-accepted all the code without even reading it
- Installed third-parties libraries blindly
- Copied my API key locally, which is now part of the AI context
  ➔ potential leak of private / sensitive info

POTENTIAL SECURITY ISSUES

Hallucination attack: LLMs (package) hallucination can be used by hackers to make you install malicious code !



https://vulcan.io/blog/ai-hallucinations-package-risk

CODE LICENSING

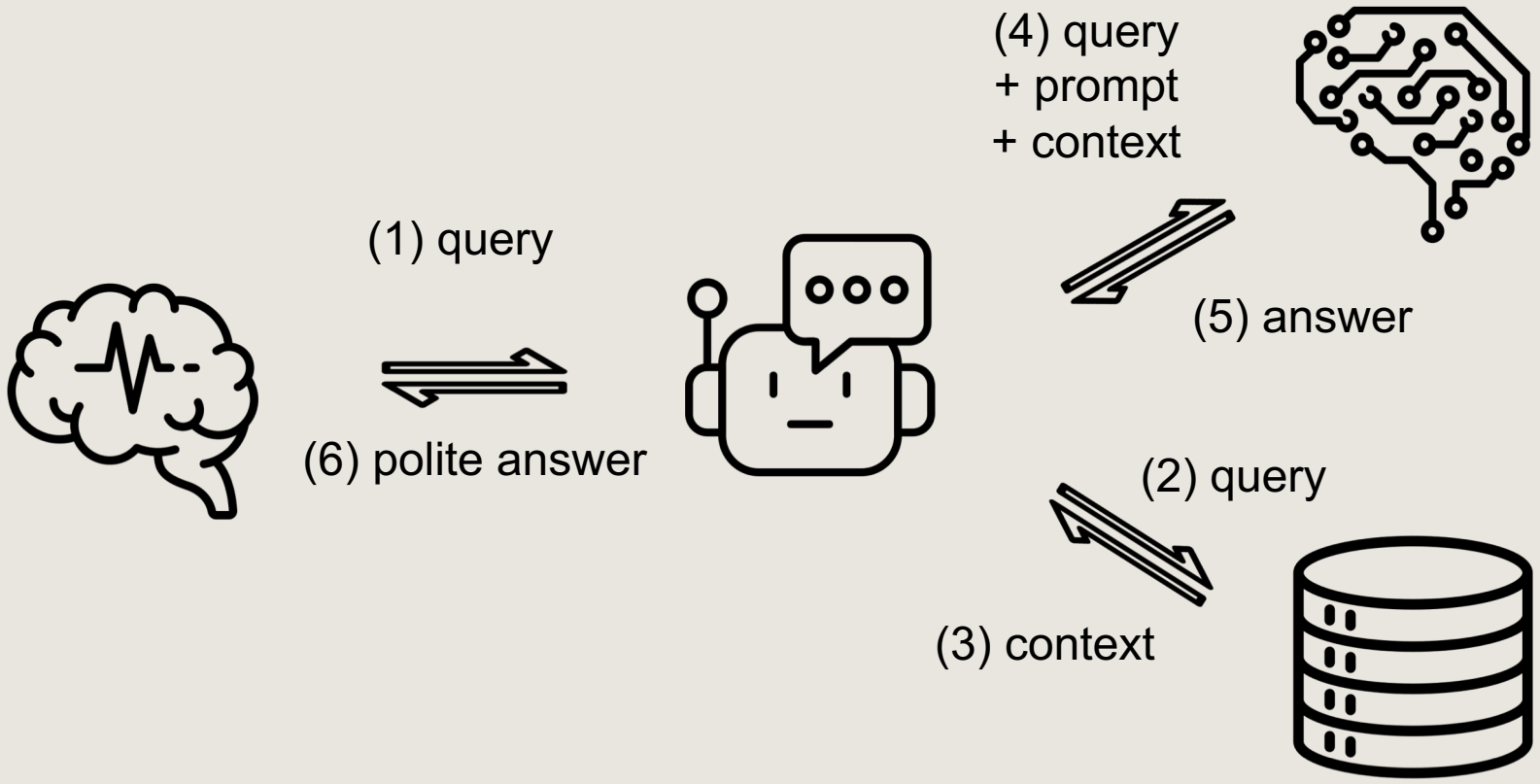
1. Online private services will use your data
   ➔ don't send code under closed licenses, don't share private data or novel ideas

2. Generated codes can be under specific licenses license or terms of use of the service used
   - chatGPT (openAI), Gemini (Google DeepMind) & Codestral (Mistral AI) give all rights to user
   - Claude is more restrictive (Anthropic retains ownership and intellectual property rights, no commercial use)

3. Some generated codes can actually be copies of a specific, licensed codes
   - be mindful when working on specific codes/cases
   - use Software Composition Analysis (SCA) tools to indentify these code snippets
   - use tools train on open-source code under permissive licenses: tabnine, codeium

RAG-LAB WORKING GROUP RI3 (IN2P3/IRFU)

- Working group to deploy LLMs in labs and fed them with internal documentation



- Easy specialization
- More up-to-date data
- Less hallucinations

RAG-LAB WORKING GROUP RI3 (IN2P3/IRFU)

- Led by Imed Magroune (CEA).
  - Electronic mailing list: https://listserv.in2p3.fr/cgi-bin/wa?A0=RI3-RAG-LABS-L
  - Site: https://gitlab.in2p3.fr/ri3/groupes-travail/gt-llm/raglabs
  - Kickoff meetings: June 14, September 10, October 29.
- Use cases:
  - User support, enhanced with documentation, intranet, tickets.
  - Log mining, laboratory notebooks, scientific publications databases.
  - Code review and generation for specific products.
  - Educational assistants, fed by courses.
- Actions?
  - Sharing experience
  - Software stack development
  - Sharing of hardware resources
  - Production of tutorials, trainings, schools ➔ aiming for an ANF next year

SOME FOOD FOR THOUGHTS – OR FOR DISCUSSION

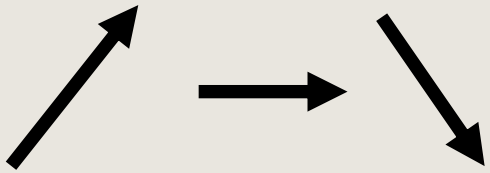
ECOLOGICAL COST

- About 10 times a google search

DILUTION OF RESPONSABILITIES

- Getting in « auto » mode and blindly trusting the AI is easy
- At the end, you are still the one in command and the one responsible
  - Use it responsibly
  - Check the code produced

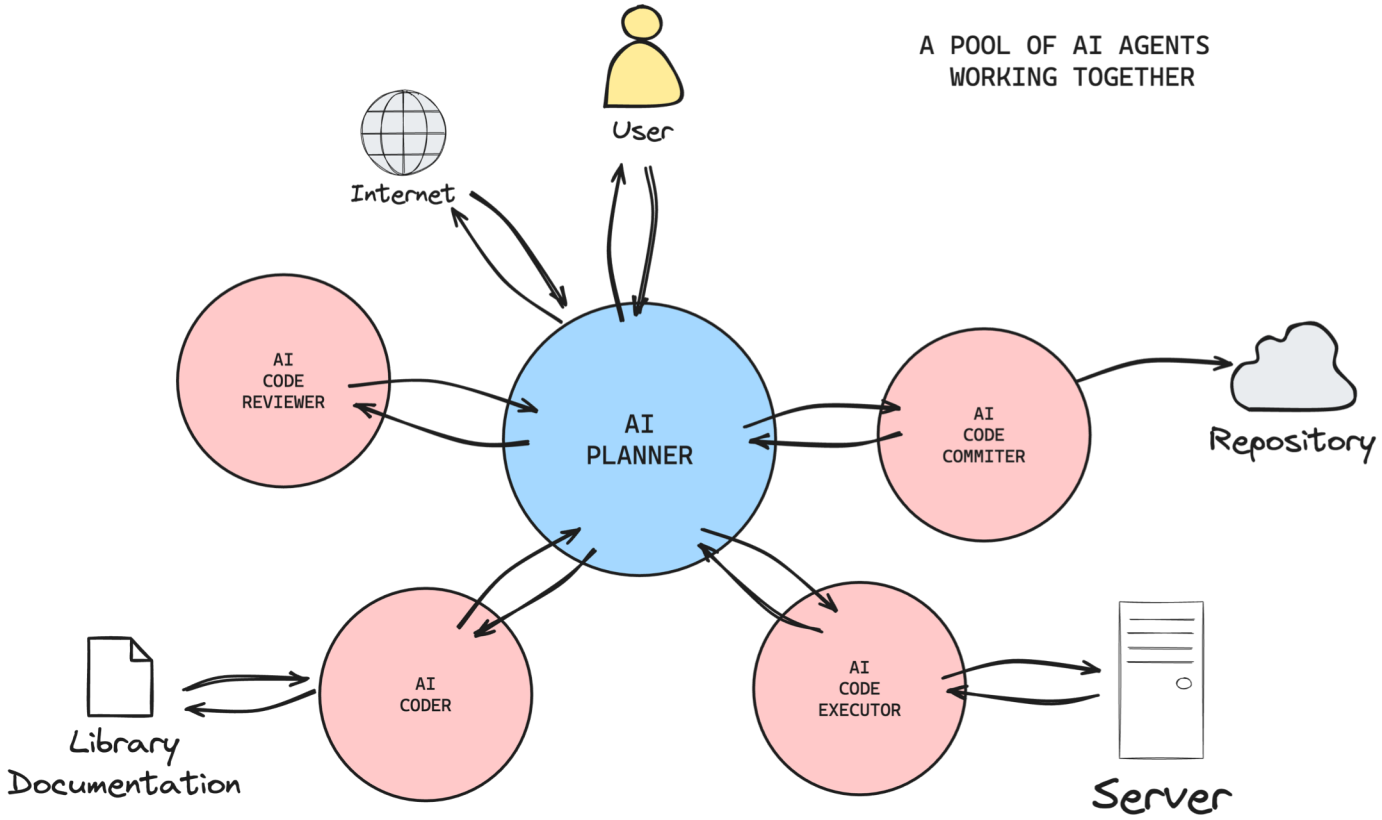
CODE QUALITY ?



SCIENTIFIC REPRODUCIBILITY

- If the code becomes so easy to generate, will people continue to archive it?

AI AGENTS: INCOMING (?) USAGE OF LLMS



- Next level automation
- Agents can autonomously perform tasks and make decisions
- They can interact with other systems
- Multiple, specialized, agents can work together to achieve a larger task

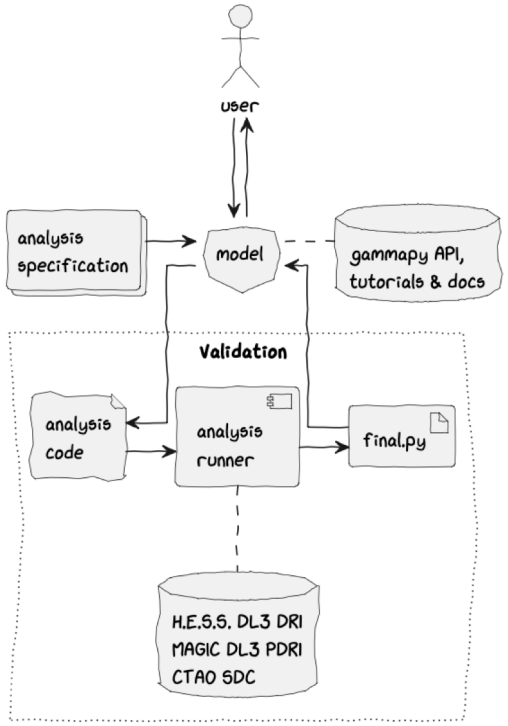
Examples:

- https://microsoft.github.io/autogen/docs/Getting-Started/
- https://www.cognition.ai/blog/introducing-devin
- https://www.anthropic.com/news/3-5-models-and-computer-use

AI AGENTS EXAMPLES IN ASTRO

First steps towards gammapy agent

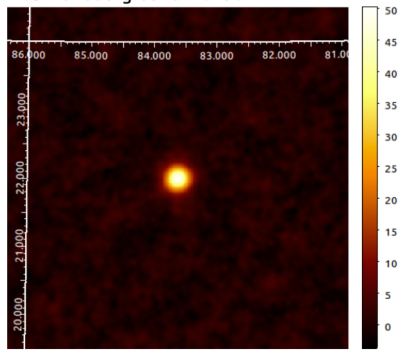
- Generic AI solutions (e.g., ChatGPT) feature shallow knowledge and outdated datasets
- **Gammapy is challenging for LLMs**
  - Different (incompatible) versions in training dataset
  - Namings, data structures, and logic are *unnatural* for machine
- **AstroAgent is addressing the challenge**
  - Field-specific prompts and Retrieval-Augmented Generation (RAG)
  - Iterative function calling and validation against data



D. Kostunin et al. AI Agents for Ground-Based Gamma Astronomy 11

First light (Crab Nebula detection from H.E.S.S. DL3 DR1)

The data for gammapy analysis are stored in `$PHOTON_STORAGE`. Generate a code which selects available observations of Crab Nebula. Using these observations, save a plot with a significance map (`sqrt_ts`) based on the `RingBackground` method. Use an exclusion mask for background maker.



```
# ... imports and comments are removed, some parts are shortened
crab_position = SkyCoord(ra=83.63308, dec=22.01450, unit="deg", frame="icrs")
selection = data_store.obs_table.select_sky_circle(
    center=crab_position, radius=2.5 * u.deg
)
observation_ids = selection["OBS_ID"].data
observations = data_store.get_observations(observation_ids)
energy_axis = MapAxis.from_energy_bounds(0.1, 10.0, nbin=10, unit="TeV", name="er
geom = WcsGeom.create(
    skydir=crab_position, axes=[energy_axis]
    width=(5, 5), binsz=0.02, frame="icrs",
)
exclusion_mask = WcsNDMap.from_geom(geom)
crab_exclusion_region = CircleSkyRegion(center=crab_position, radius=0.3 * u.deg)
exclusion_mask.data += geom.region_mask(regions=[crab_exclusion_region], inside=F
dataset_empty = MapDataset.create(geom=geom)
dataset_maker = MapDatasetMaker()
bkg_maker = RingBackgroundMaker(r_in="0.35 deg", width="0.3 deg", exclusion_mask=
safe_mask_maker = SafeMaskMaker(methods=["offset-max"], offset_max=4.0 * u.deg)
for obs in observations:
    dataset = dataset_maker.run(dataset_empty, obs)
    dataset = bkg_maker.run(dataset)
    dataset = safe_mask_maker.run(dataset, obs)
    dataset_empty.stack(dataset)
estimator = TSMapEstimator()
ts_map = estimator.run(dataset_empty)
significance_map = ts_map["sqrt_ts"]
print(f"Max significance value: {np.max(significance_map.data):.2f}")
# prints: Max significance value: 50.32
significance_map.write("significance_map.fits", overwrite=True)
```

D. Kostunin et al. AI Agents for Ground-Based Gamma Astronomy 12

1. CTAO: User (astronomer) requests in plain language (english) what data they want to analyse and how
2. https://astromlab.org/ develops LLMs specialized in astro to automatized research in the field

Presented at ADASS 2024: https://pretalx.com/adass2024/talk/TRFZKU/
Demo: https://majestix-vm8.zeuthen.desy.de/

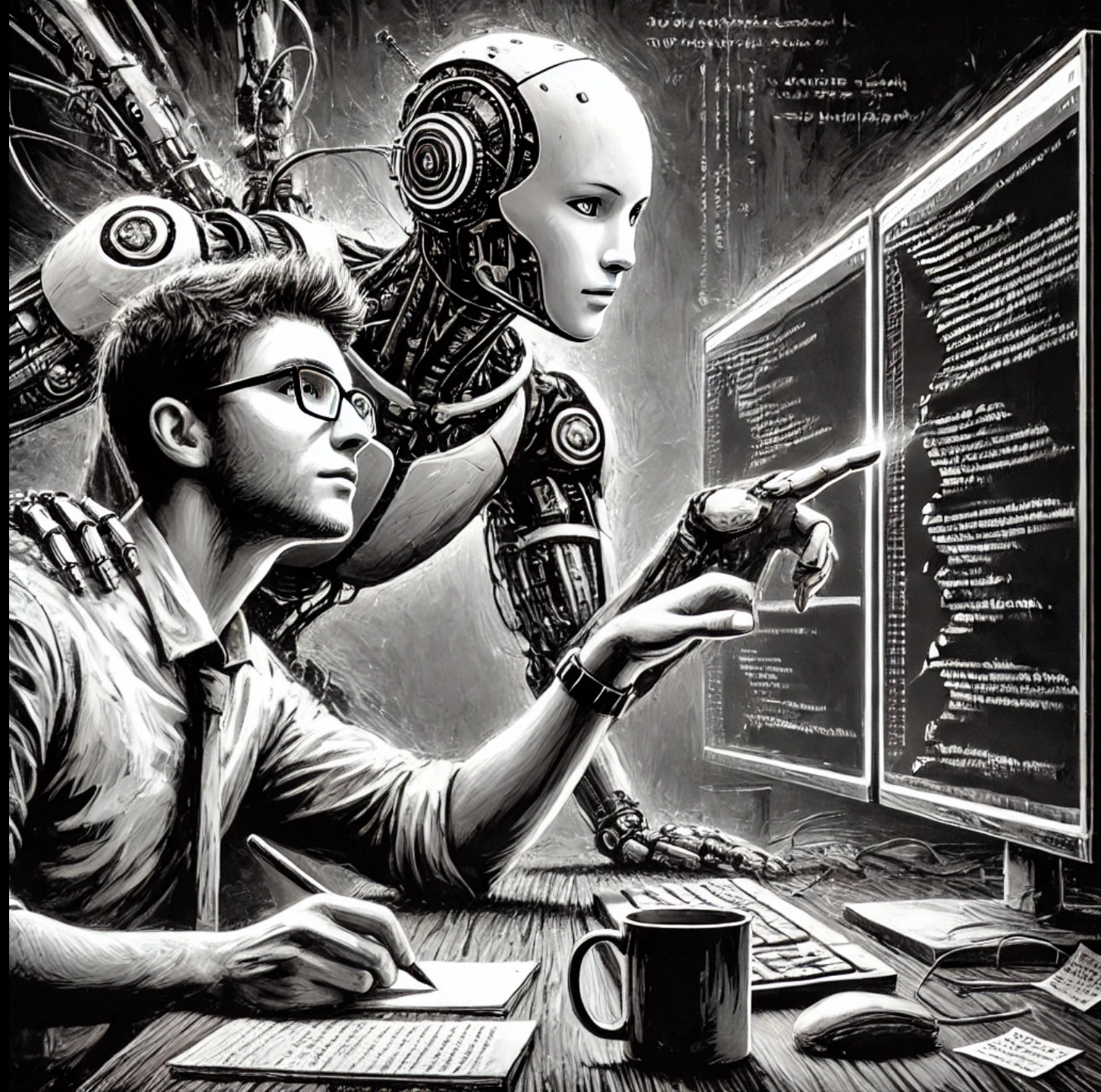
slido

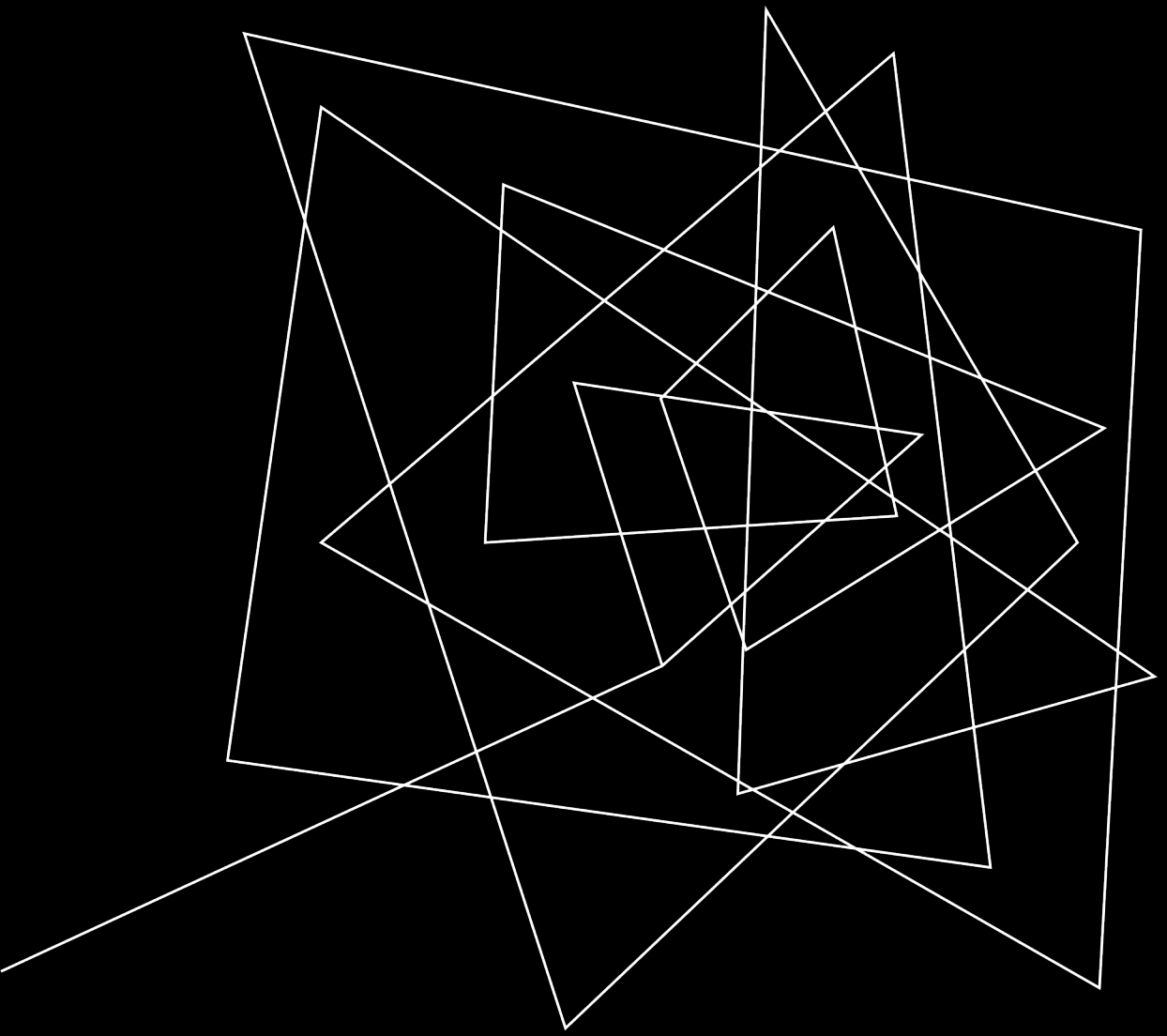
Please download and install the Slido app on all computers you use

Après cette présentation, est-ce que vous allez utiliser un assistant IA pour le code ? / So, now, are you going to use any of these AI powered tools?

ⓘ Start presenting to display the poll results on this slide.

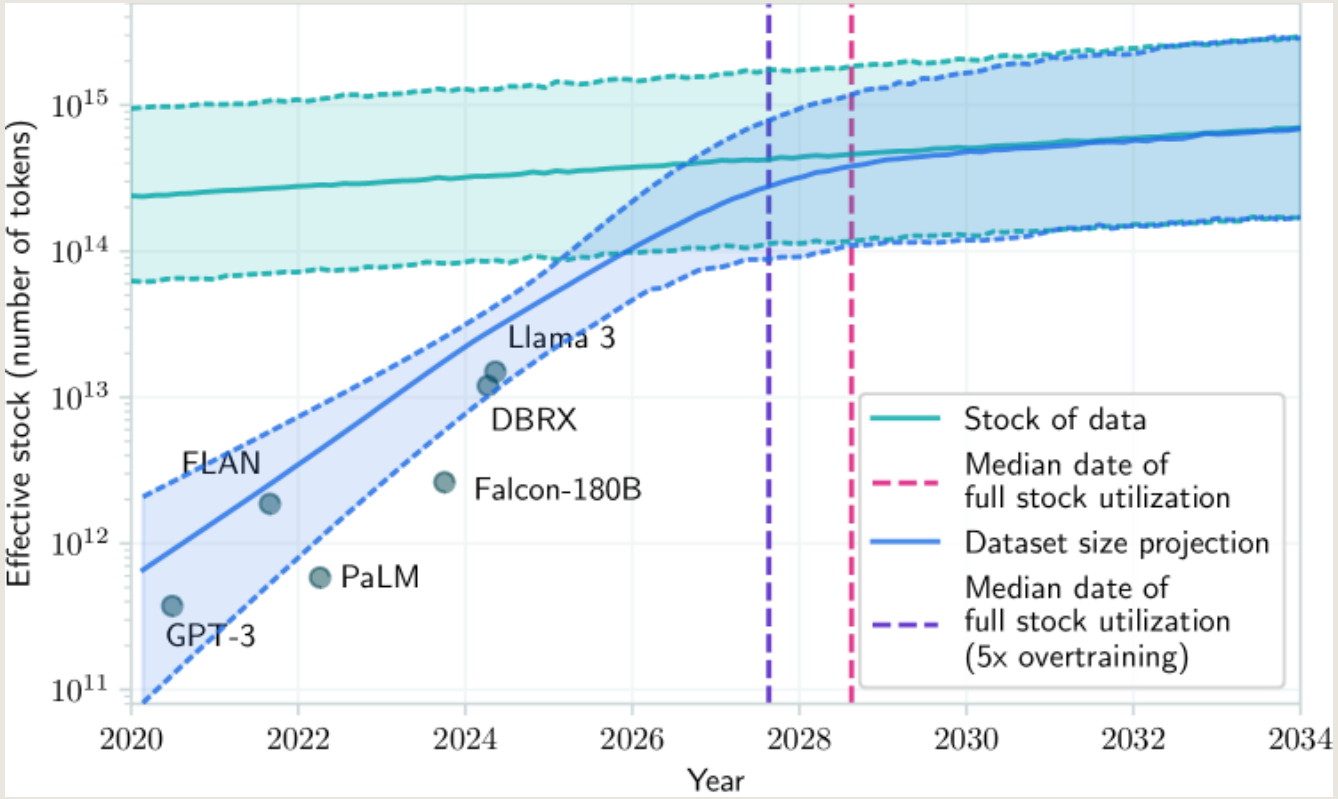
THANK YOU…





BACK-UP

THE LACK OF TRAINING DATA



LLM performances scale with:

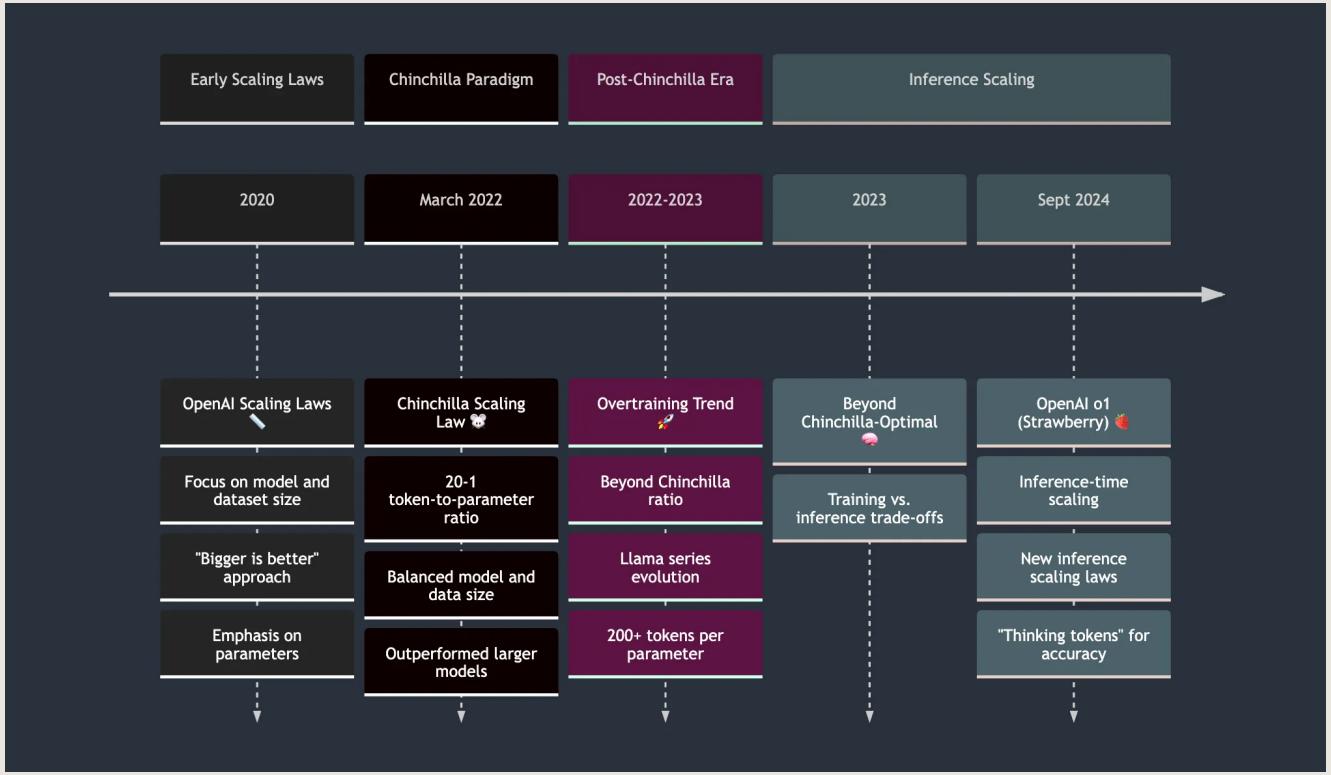
- Training data quantity
- Training data quality
- Model size
- Compute

3 ways to circumvent:

- Non-public data
- Use synthetic data (works well in domains where output can be checked, e.g. code, maths, games… can be extended?)
- Multi-modality (use videos)

https://arxiv.org/html/2211.04325v2

SCALING LAWS & THE NEW PARADIGM OF INFERENCE-TIME SCALING



- Early scaling laws (Kaplan et al., 2020) established power-law relationships between model size, data, and performance.
- The Chinchilla paradigm shift (2022) introduced the 20:1 token-to-parameter ratio for optimal training.
- Post-Chinchilla developments saw “overtraining” beyond the 20:1 ratio, yielding performance gains.
- Recent models like Llama-3 pushed token-to-parameter ratios to 200:1, challenging previous assumptions.
- Inference scaling (OpenAI’s o1 model, 2024) emerged as a new direction, focusing on optimising inference-time compute for improved reasoning.

Beyond Bigger Models: The Evolution of Language Model Scaling Laws

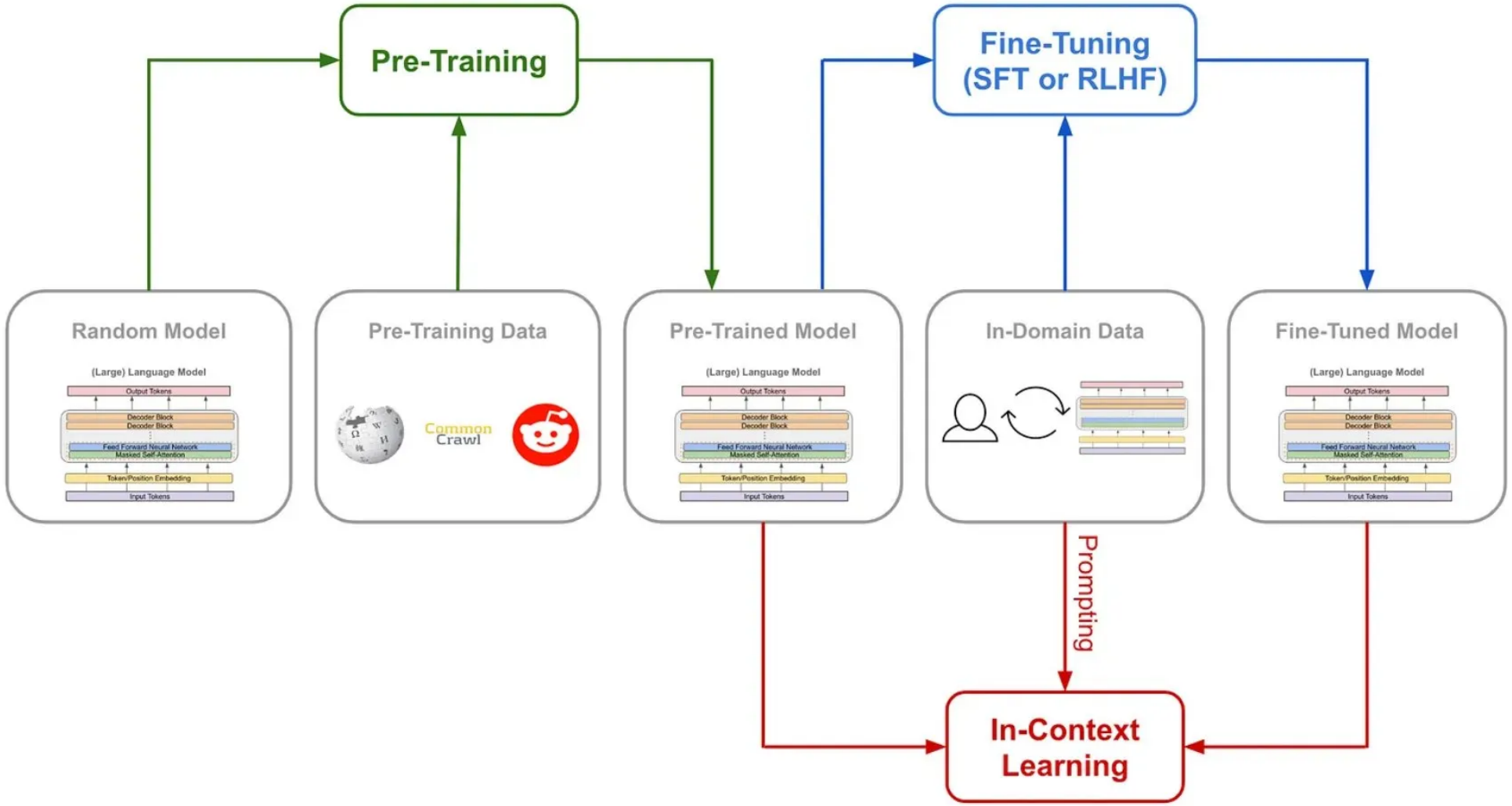
MODEL COLLAPSE OR « THE CURSE OF THE PHOTOCOPIER »

Internet → AI models → internet → AI models → internet …

➔ Decrease of performances
➔ Increase of biaises

No clear solution at the moment but potiential mitigation are:

- Maintain high-quality human-generated data
- Filter data
- Mitigate during training



[Empowering Language Models: Pre-training]