CIKLUM

SPEAKER'S CORNER

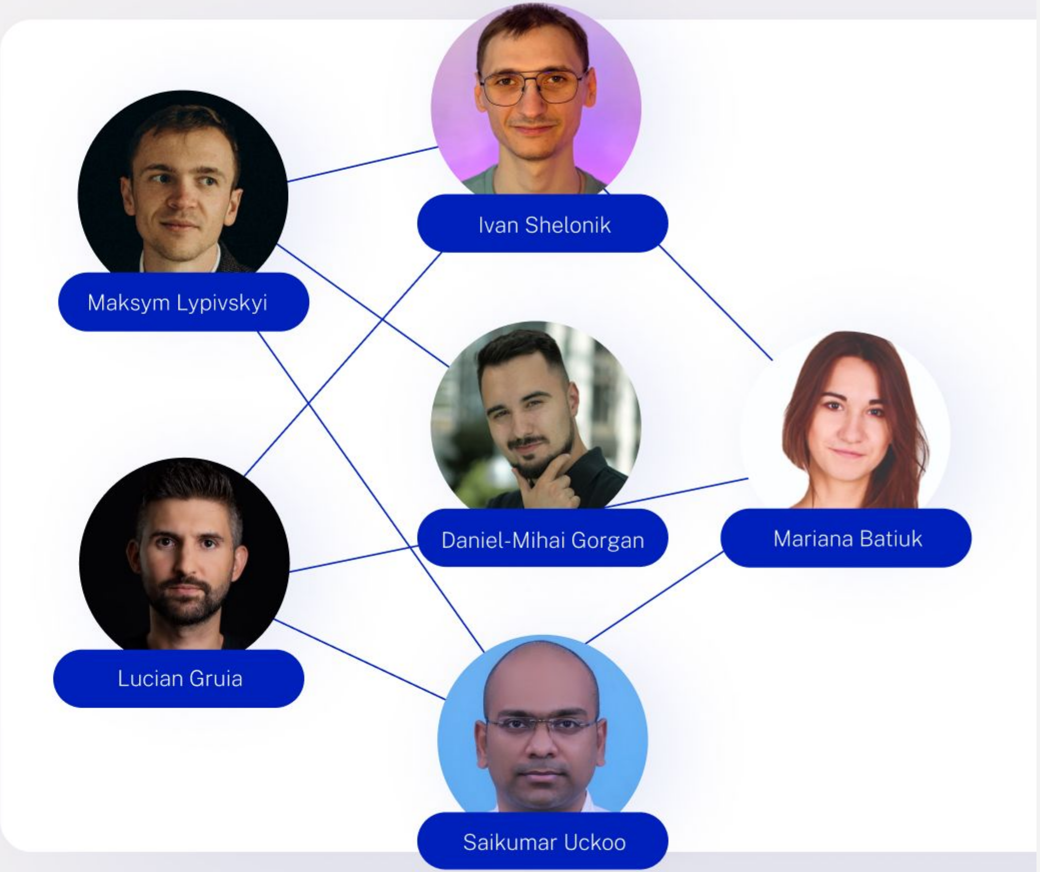
April, 16

16:30 (CET)

English

Architecting Scalable AI RAG Systems:

From Startup to Enterprise. A Live Coding Session



Experiences of tomorrow. Engineered together.

CIKLUM

We transform how people experience the business. All through next generation technology.

2002
founded

4000+
professionals

20+
offices

300+
clients

What we do:

- Product Engineering
- Intelligent Automation
- Data & Analytics

Leading companies choose us:

JUST EAT Takeaway.com | METRO MARKETS | eToro | ZURICH | pay | Greens | KANTAR RETAIL | dacadoo

Our Global Delivery Centres

Global Reach, Local Insight - Ciklum bridges the best in tech from the three key IT regions

Central & Eastern Europe

- Bulgaria
- Czech Republic
- Poland
- Romania
- Slovakia
- Spain
- Ukraine
- United Kingdom

Asia

- India
- Pakistan

LATAM

- Argentina
- Uruguay

Our speakers

CIKLUM

Lucian Gruia
Principal AI Technology Lead

- AI Tech Lead with over 11 years of hands-on experience in Telecom, Fintech, and Aerospace. He specializes in AI, data integrity, fraud detection, system performance, architecting frameworks and solutions for real-time systems.
- Develops an AI upskill program for 300 engineers at Ciklum.

Ivan Shelonik
Expert Data Scientist

- Certified Professional Machine Learning Expert with 7 years of commercial experience in developing Machine Learning projects from the ground to delivery into the Cloud (AWS 5+YoE).
- Has worked and delivered primarily for customers from S&P 500

Daniel-Mihai Gorgan
Senior JS Developer

- Tech enthusiast specializing in Node.js, SQL/NoSQL and Cloud technologies with 5+ years of experience
- Hands-on experience in projects across outsourcing and product companies, contributing to the development of in-house products, smart chatbots, and voicebots by leveraging different AI technologies

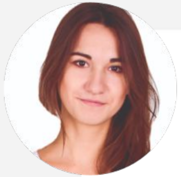
Our speakers

CIKLUM

Saikumar Uckoo

Conversational AI expert

- Cloud Architect specialized in building, deploying, and maintaining AI solutions on Microsoft cloud platforms. Leads deliveries on platforms like Microsoft PVA, KoreAI, and custom GenAI solutions built on open-source tech.



Mariana Batiuk

Principal TCoE Lead

- Mariana leads the technical council on the QA maturity assessment, test strategy, pre-sales, new services development, initiatives, and quality engineering activities.
- Has proficient experience in QA Management, Agile Methodologies, Testing, Team Management, and Coaching.

Maksym Lypivskyi

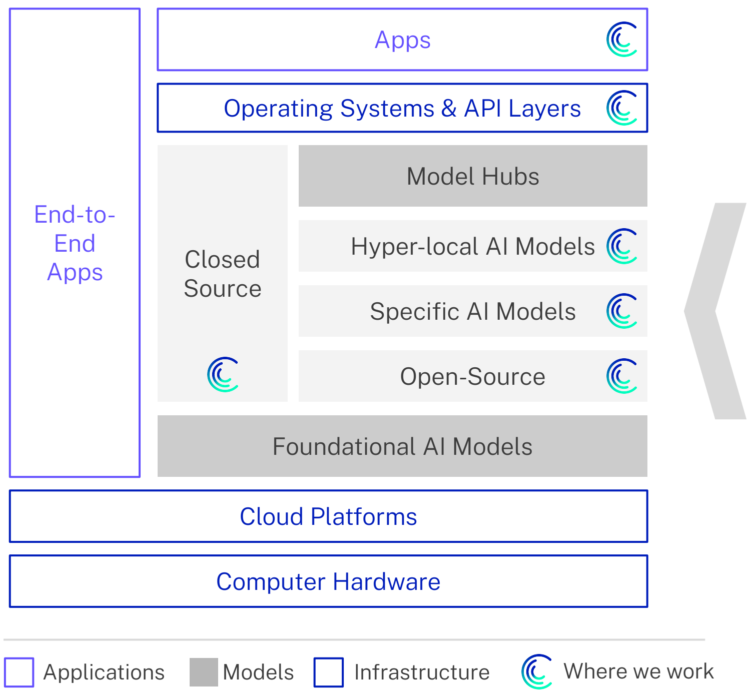
Global Head of Cloud Platforms

- Specializing in cloud computing architectures and generative AI applications, he focuses on creating, deploying, and optimizing cutting-edge solutions across global platforms.
- A mentor and community builder, he actively shares his insights on generative AI, cloud technologies, and leadership.

Playing in all parts of the AI stack

CIKLUM

User Experiences & Engagement



	Emerging Stack Trends	Partners
Applications	The rise of cloud-based generative AI and LLMs, accessible via **APIs** and **embedded** in other applications, will allow companies to use them as-is or **customize** with their data	OpenAI, Weights & Biases, Humanloop
Fine-Tuning	The need for model fine-tuning will drive demand for a **diverse skill set,** such as software engineering, psychology, linguistics, etc.	Hugging Face, Pinecone
Foundation Models	The market will evolve and diversify with the emergence of **more pre-trained models,** offering **options** for size, transparency, versatility & performance balance	OpenAI
Data	Mastery of **new and diverse data types** and volumes will be crucial for success, with **GenAI features in modern data platforms** facilitating **adoption at scale**	appen, redis, snowflake, databricks
Infrastructure	Essential for GenAI deployment, **cloud infrastructure** will help manage costs and **carbon emissions**, necessitating data center retrofitting and advancements in chipset architectures, **hardware** & algorithms	Azure, aws, Google Cloud, NVIDIA

Agenda

CIKLUM

01 What is RAG

02 LLM Wrappers and Docker

03 Build with Java

04 Build with Python

05 Build with Javascript

06 Deploy RAG app in AWS

07 Deploy RAG app in Azure

08 Challenges in QA and more

Session's Tech map

CIKLUM

Programming languages

Java

JS

Databases

FAISS

qdrant

Infrastructure

mac OS

docker

aws

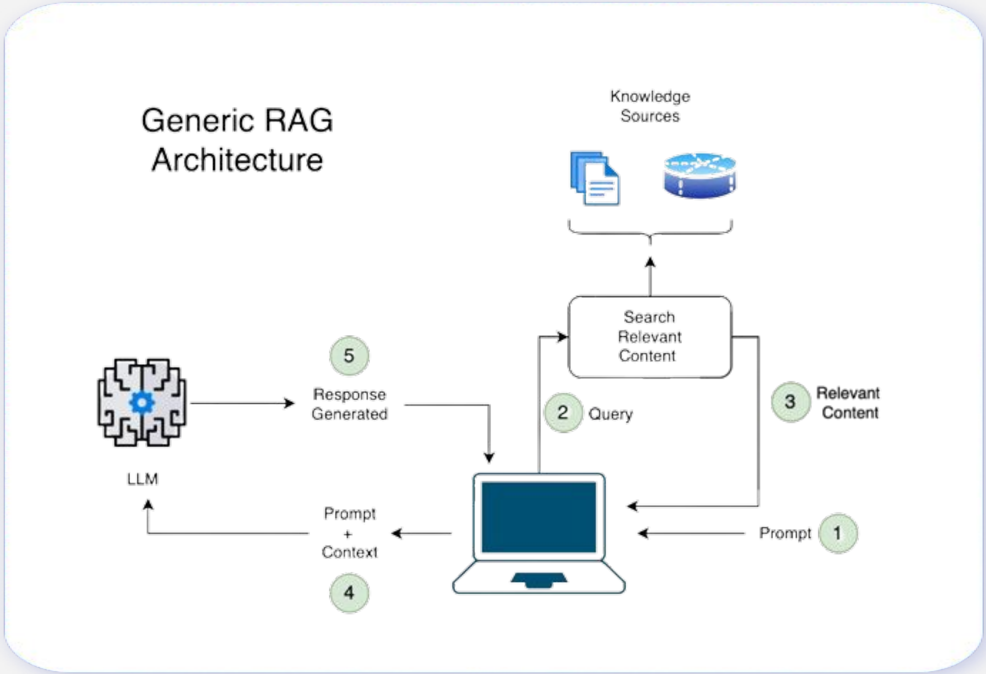
Azure

What is RAG

A RAG system essentially correlates a **user's prompt** with a relevant **data chunk.** It does this by identifying **the most semantically similar** chunk from the database.

This chunk then becomes **the context** for the prompt.

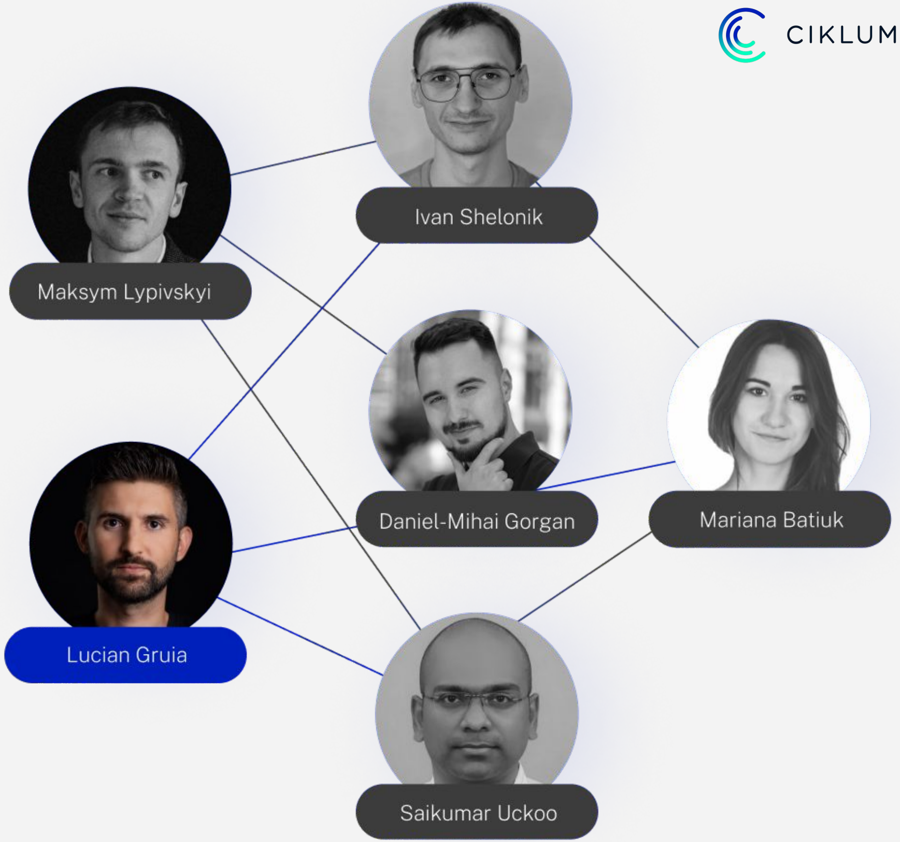
When **passed to the Large Language Model (LLM),** it enables the system to provide a relevant answer within the given context.



CIKLUM

LLM Wrapper

- Build with **Java**
- Deploy locally
- Integrate a 3rd party client



Why do we need RAG?

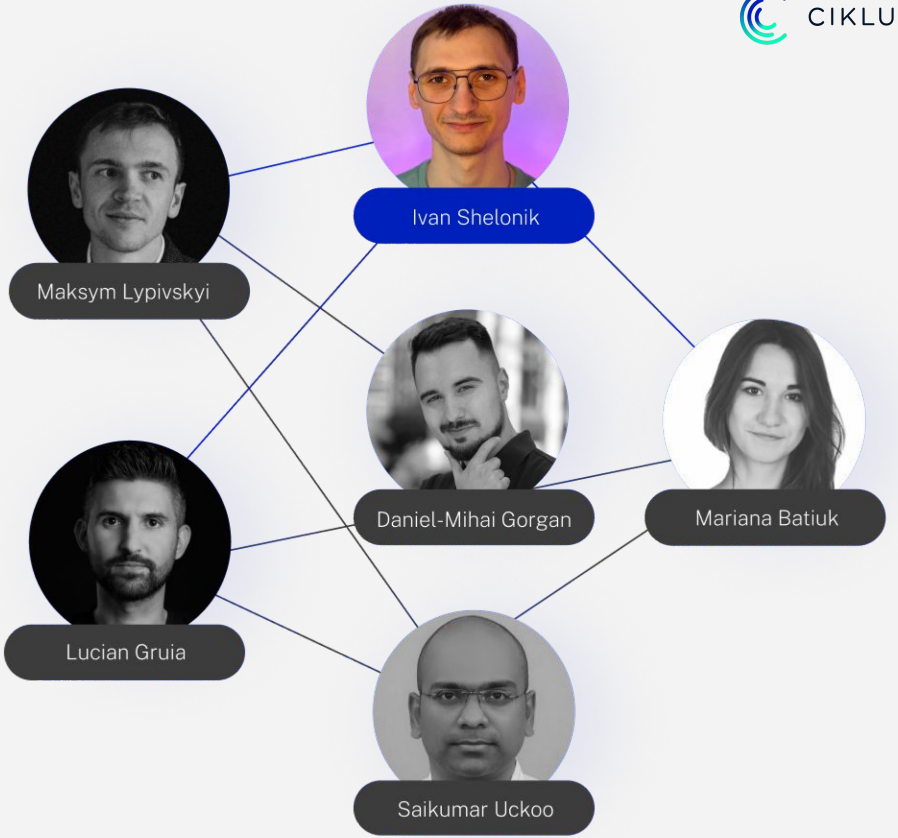
CIKLUM

- Expands Knowledge Base
  RAG accesses a vast external database, enriching its knowledge beyond initial training data
- Improves Accuracy
  Enhances response precision by integrating relevant, real-time information
- Adaptable
  Effectively handles novel and niche queries
- Increases Efficiency
  Streamlines information retrieval and generation process
- Versatile Applications
  Useful across various fields, from customer support to research

Source: What Is Retrieval-Augmented Generation, aka RAG?

AWS

- Build with **Python**
- Build Docker images
- Semantic search with FAISS
- Deploy on AWS



CIKLUM

Data Chunking and LLMs

CIKLUM

LLMs also have a limited capacity for context.

Just as humans **cannot digest unlimited context**, these models have a specific size limit for the content they can process.

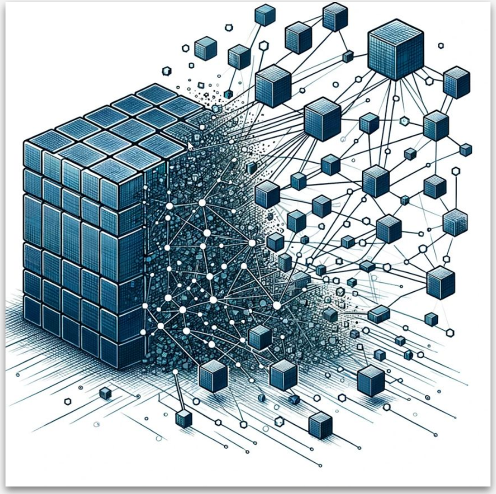
So, what about situations involving very large amounts of data?

Consider a specific use case, such as a book. It's too large to pass the entire book as **the context** for the current prompt, so it **needs to be divided** before being stored in the database.

This process is known as **data chunking**.

Types of Data chunking (by size):

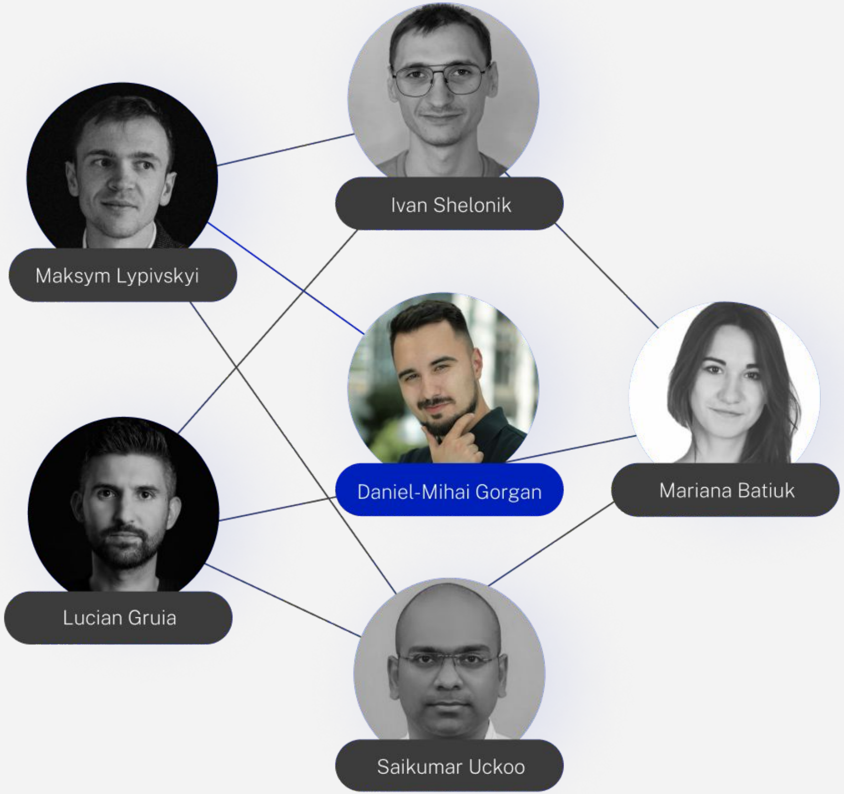
- Fixed-size
- Variable Chunking
- Semantic Chunking



Generated with DALL·E 3

JavaScript

- Build with **TypeScript**
- Semantic Search with Pg vector

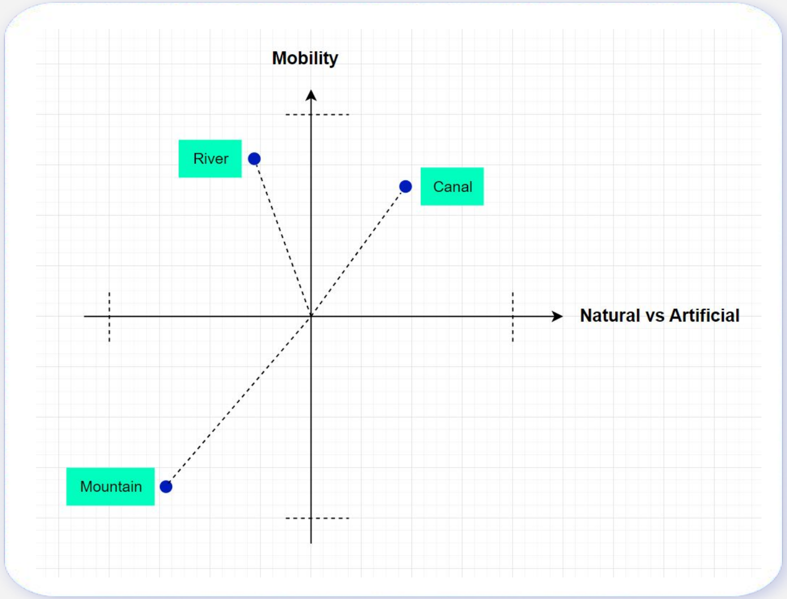


Embeddings. Similarity

- **Embeddings**
  Numerical representations of concepts, in a high-dimensional space, capturing semantic meaning.
- **Similarity:**
  - Lexical: entities are alike in appearance
  - Semantic: entities are alike in meaning
- **In RAG we represent entities by describing them.**
  This is a form of knowledge representation.

Example: Mountain, River, Canal

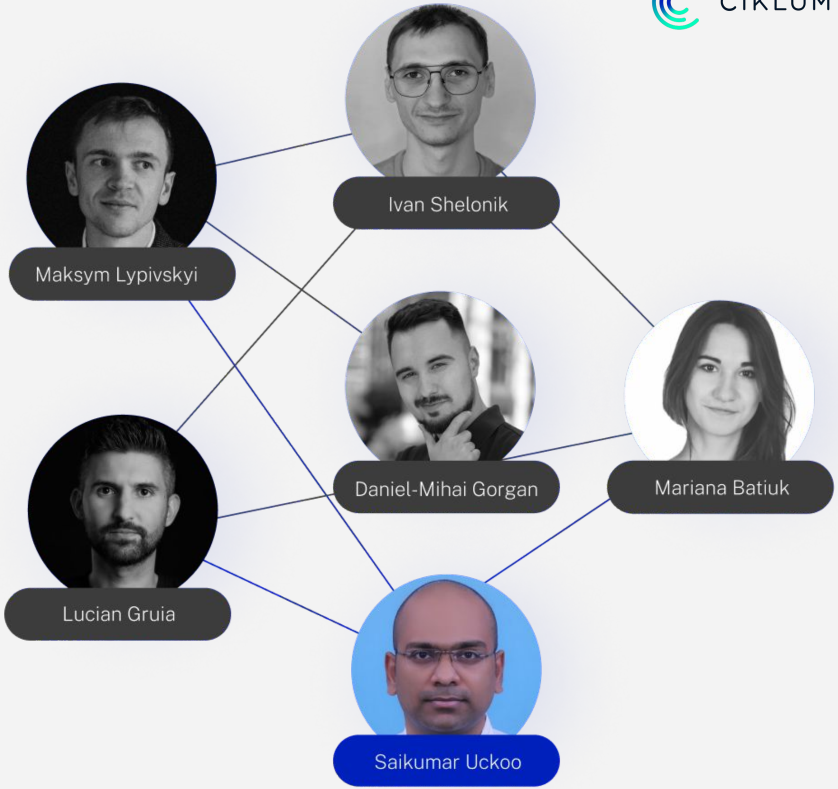
One hot encoding		2-Dimensional Space	[Natural vs Artificial, Mobility]
Mountain:	1	Mountain:	[-0.7, -0.8]
River:	2	River:	[-0.3, 0.7]
Canal:	3	Canal:	[0.4, 0.5]



Read more: Wikipedia - Cosine Similarity

Azure

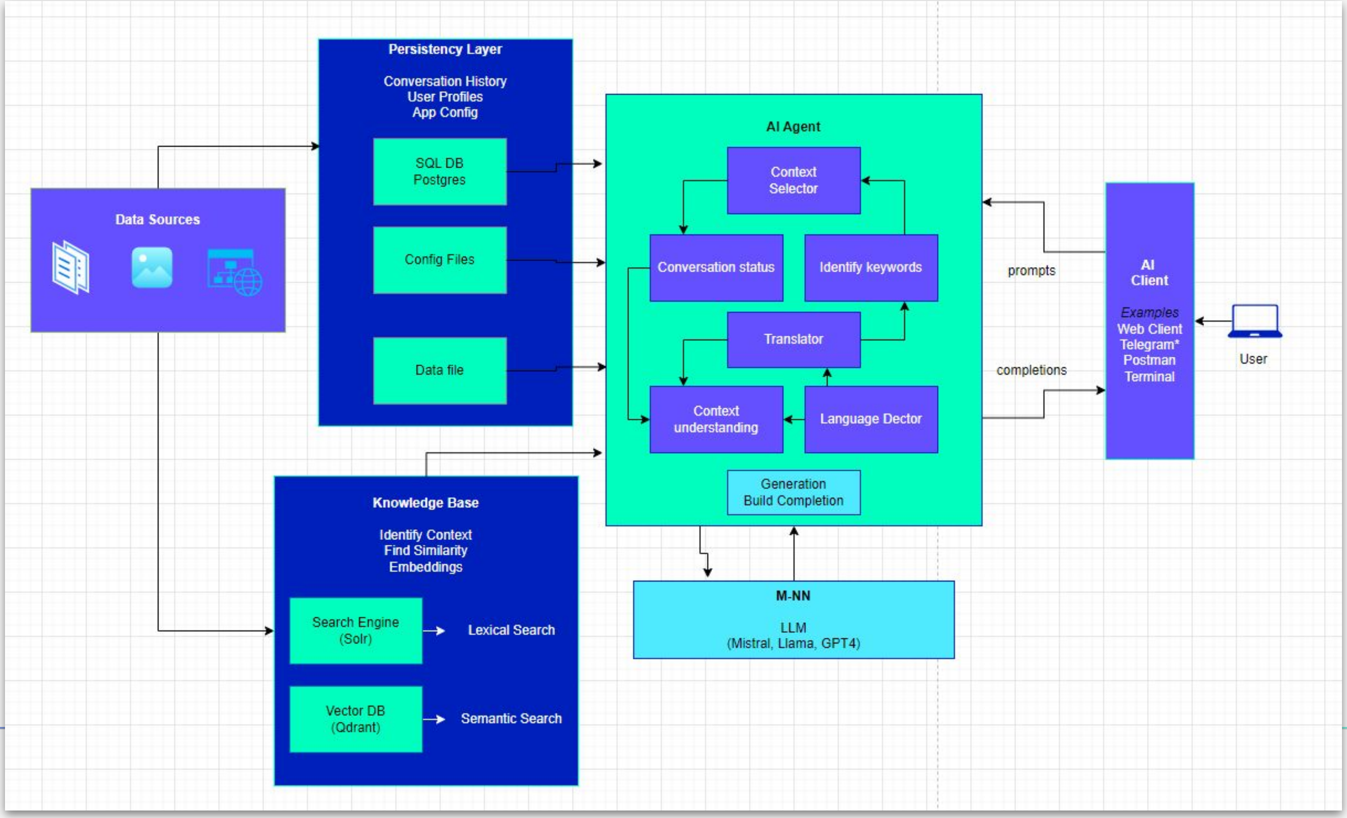
- Deploy on **Azure**
- Semantic Search with Qdrant
- Conversation history



CIKLUM

RAG Architecture

CIKLUM



Benefits of RAG

CIKLUM

1. Providing up-to-date and accurate responses
   RAG ensures that the response of an LLM is not based solely on static, stale training data. Rather, the model uses up-to-date external data sources to provide responses.

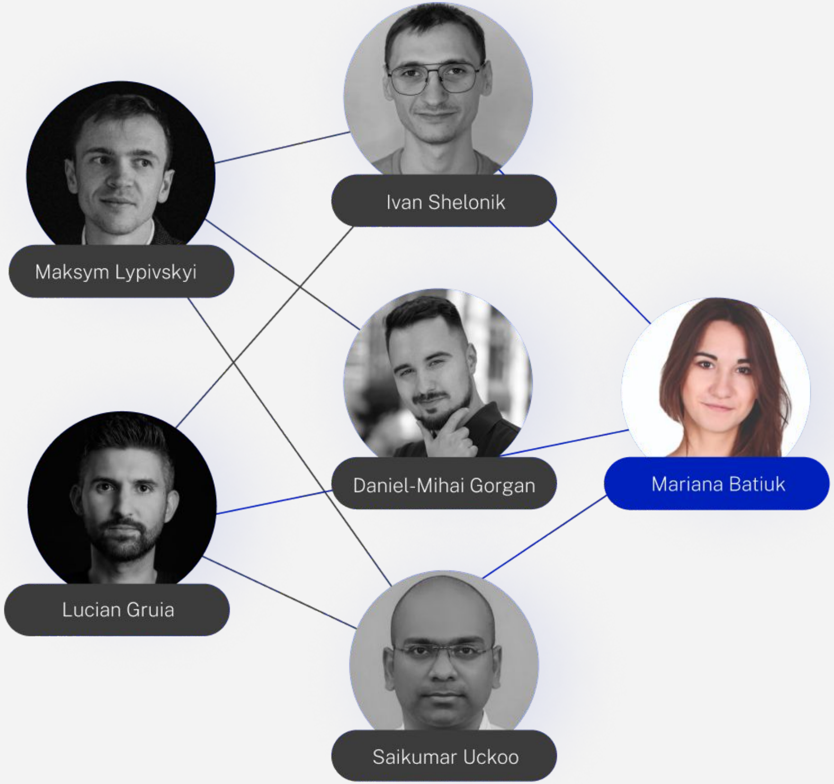
2. Reducing inaccurate responses, or hallucinations
   By grounding the LLM model's output on relevant, external knowledge, RAG attempts to mitigate the risk of responding with incorrect or fabricated information (also known as hallucinations). Outputs can include citations of original sources, allowing human verification.

3. Providing domain-specific, relevant responses
   Using RAG, the LLM will be able to provide contextually relevant responses tailored to an organization's proprietary or domain-specific data.

4. Being efficient and cost-effective
   Compared to other approaches to customizing LLMs with domain-specific data, RAG is simple and cost-effective. Organizations can deploy RAG without needing to customize the model. This is especially beneficial when models need to be updated frequently with new data.

QA & Testing

- SW characteristics
- Top 5 **risks**
- Methods and tools
- Balanced **success** factors



Software Characteristics

CIKLUM

ISO 25010 Product Quality Model							
Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability

Functional Testing → ***What*** *does the system do?*

Non-Functional Testing → ***How*** *do the system do this?*

AI-specific Characteristics							
Flexibility & Adaptability	Autonomy	Evolution	Bias	Side-effects & Reward Hacking	Ethics	Transparency, Interpretability & Explainability	Safety

ISTQB®
Certified Tester
AI Testing

Top 5 current shortcomings and risks

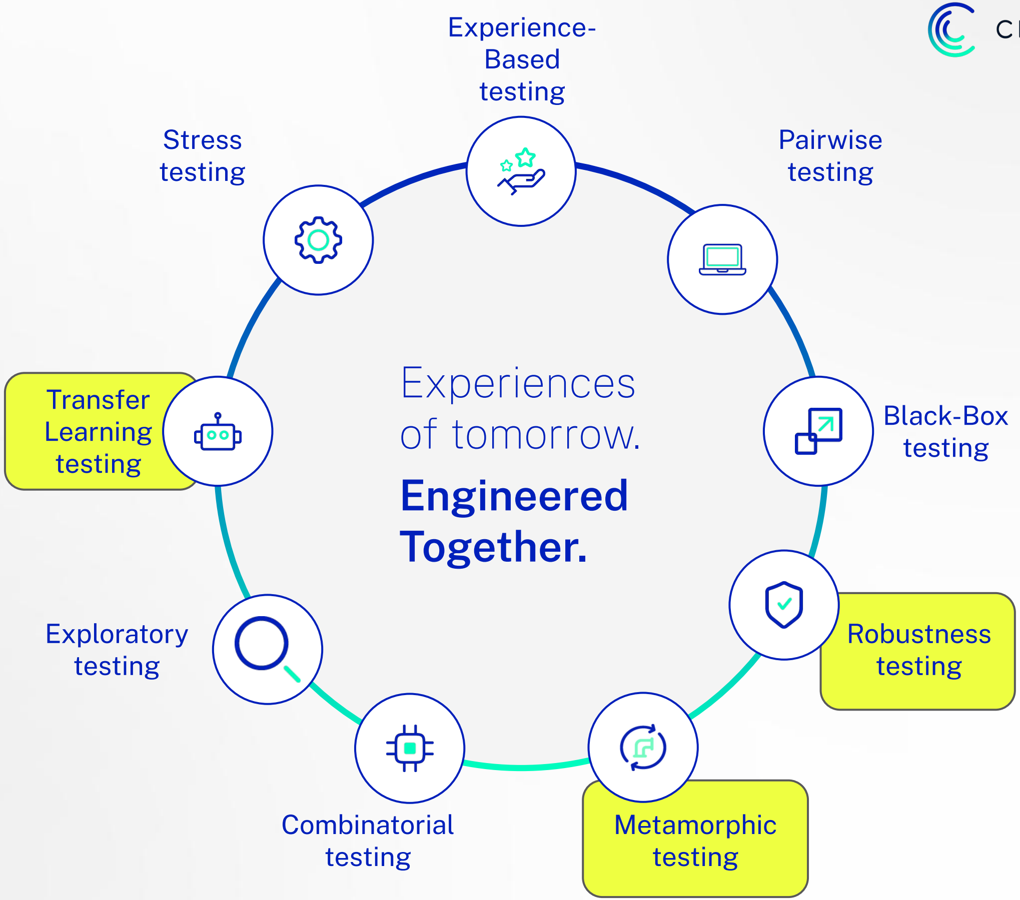
	Lack of Transparency	Ethical & Bias Concerns	Dynamic Learning	Loss of Control	Hallucinations
Impact on Testing / QA	Difficult to design test cases	Unfair outputs	Unpredictable system behavior	Impaired predictability	Misleading or inaccurate outputs
	Low interpretability	Challenging bias testing	Complex testing updates & maintenance	Difficulties in oversight	Unreliable testing & results

CIKLUM

Methods & tools

that help us mitigate risks and ensure proper testing of AI

gretel ragas Giskard

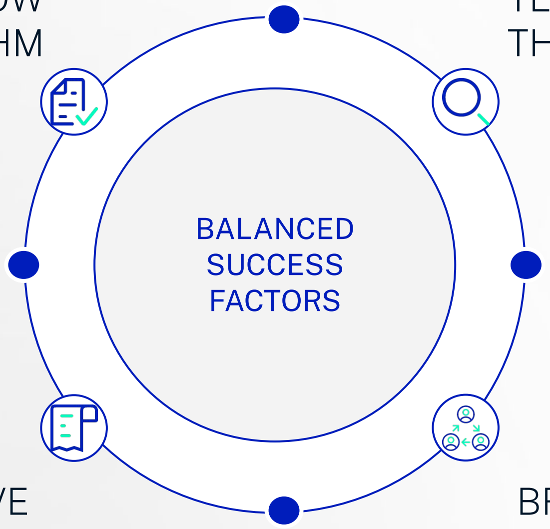


Some essential elements

that should be considered when verifying AI systems

KNOW
THE ALGORITHM

TEST
THE ALGORISM

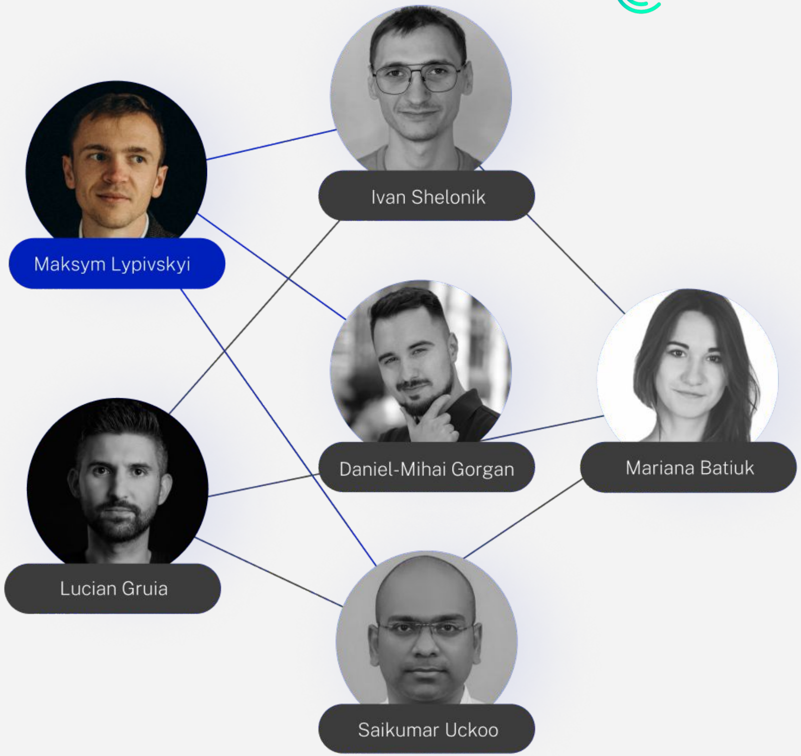


MAKE SURE TO HAVE
ENOUGH DATA

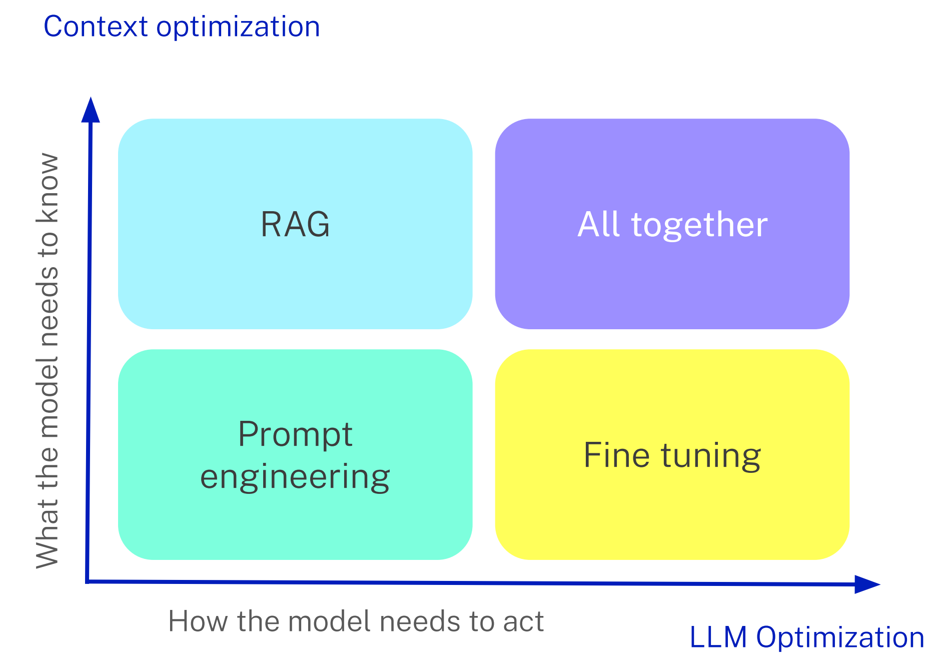
BRING THE
RIGHT PEOPLE

Interact

- **Prompt** Engineering
- Fine-tuning



The optimization flow



What is a good prompt

Act as an experienced Learning specialist. I need to improve my upselling skills. Prepare an educational program for me to improve that skills. Program should be for 2 month with 4 hours effort per week.

Please provide answer with the next output:

Topic: Name

- blocks
- ...

Books:

Example:

Topic: Negotiation basics

- Win-win strategy
- Active listening strategy

Books: "Getting to Yes" by Roger Fisher and William Ury

Instruction
Context
Role
Formatting
Tone
Examples

Prompt tactics

CIKLUM

* Shot Prompting

Zero
Add 2+2:

One
Add 3+3: 6
Add 2+2:

Few
Add 3+3: 6
Add 5+5: 10
Add 2+2:

Model-guided prompting

Before answering, I want you to first ask for any extra information that helps you produce a better answer.

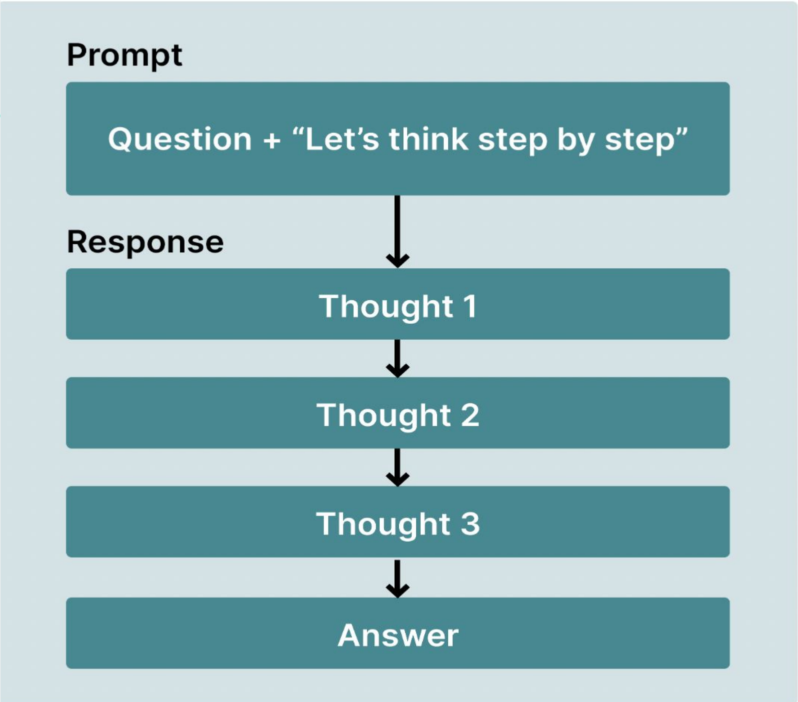
If you got no questions, please provide an answer instead.

Self-evaluating prompting

Can this program be improved?

Chain of thoughts

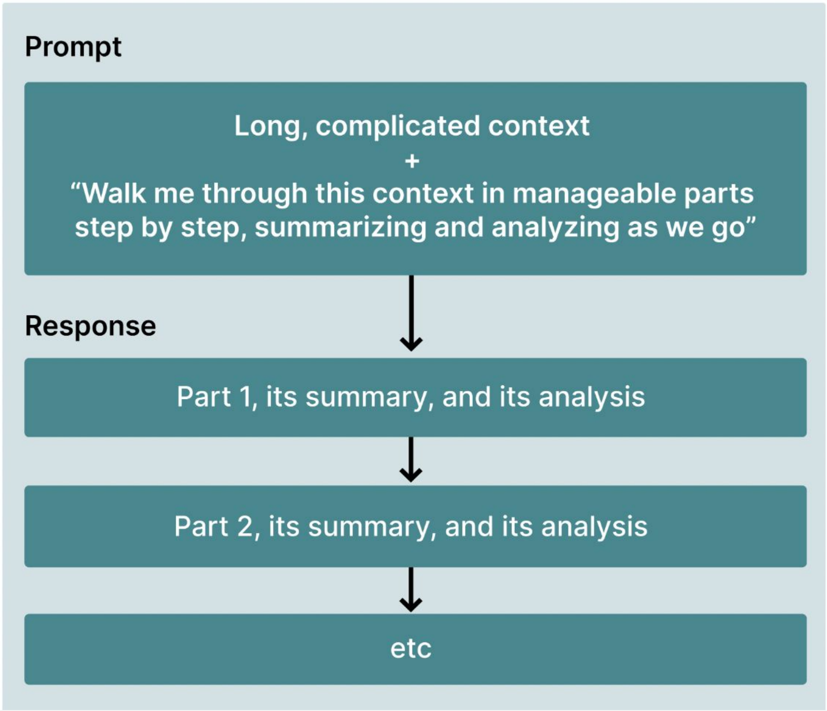
Virma has three bags, each of which fits five shirts. How many shirts can Virma fit in her bags?
Let's think step-by-step.



Thread-of-Thought

Virma has three bags, each of which fits five shirts. How many shirts can Virma fit in her bags?

Walk me through this context in manageable parts step by step, summarizing and analyzing as we go.



CIKLUM

Thank you!

CIKLUM

Share your feedback!

Join our team