

# Agenda

- AI Today: The Generative Landscape
- Serving a LLM On-Prem: Open WebUI
- *Coffee break*
- **RAG – Knowledge Bases and Linking a SQL Database**
- How Agents Talk to Tools: Towards MCP
- Closing Remarks

# My First LLM App, Intro to RAG

- Dr. Elena Nazarenko



**HSLU** Hochschule  
Luzern

# About me

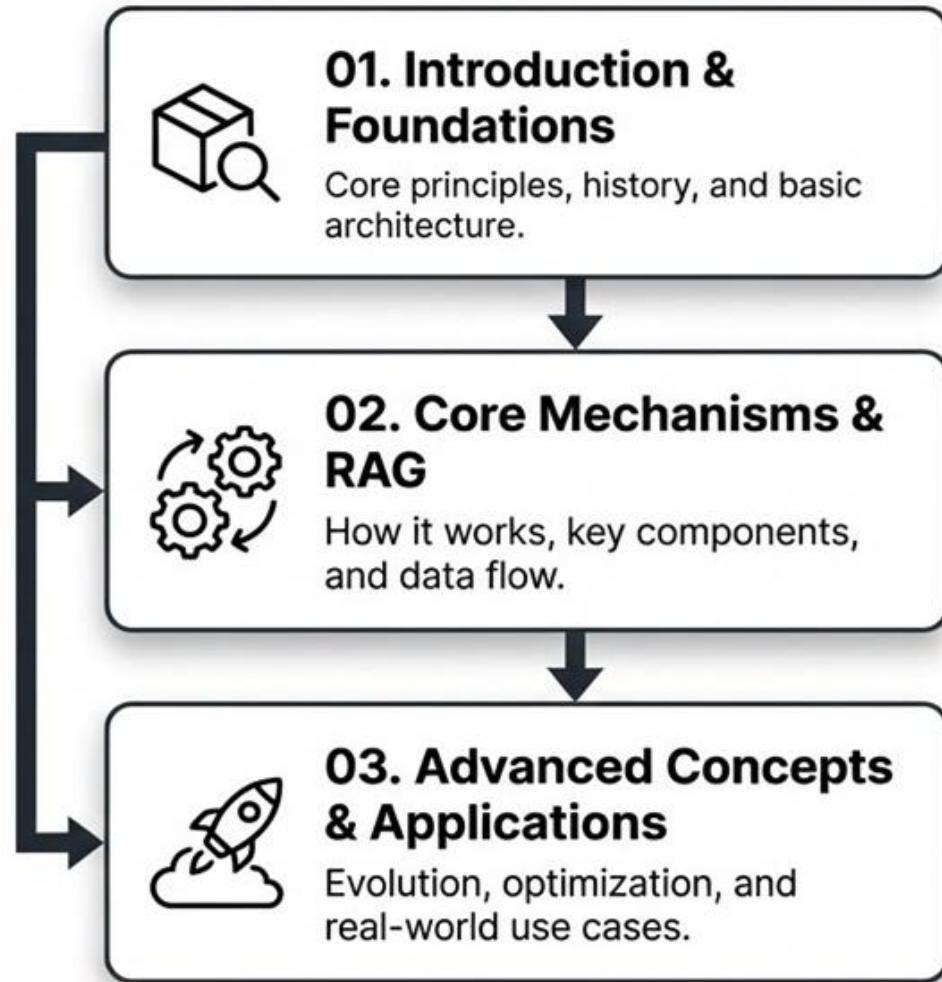


## Elena Nazarenko, PhD

- Senior Lecturer & Researcher in ML, AI, and LLMs @HSLU  
(Best Paper@Swiss Data Science Conference & Best Poster@SwissText Conference)
- Co-director LLMs& AI Agents Bootcamp @HSLU
- Organised workshops and conference tracks @AMLD & @SDS
- Former Head of Data & AI @Witty Works  
(part of the Hugging Face startup accelerator; finalist Microsoft Entrepreneurship for Positive Impact Cup)

### Background:

PhD University Grenoble Alpes, Research institutes in France, Sweden, Switzerland (Paul Scherrer Institute - ETH Domain)



## Section 1: Building the Base

- > Terminology & Concepts
- > Evolution of Retrieval
- > Fundamental Models
- > The "Why" of RAG

## Section 2: The Mechanics

- > Indexing & Storing
- > Query Processing
- > Generation Phase
- > Integration Strategies
- > Evaluating Performance

## Section 3: Pushing Boundaries

- > From Basic to Advanced RAG
- > Addressing Limitations
- > Hybrid Approaches
- > Future Trends & Research

# RAG – an Introduction

Understanding Retrieval-Augmented Generation Fundamentals and Core Concepts

BY **HARRY BOOTH**

SEPTEMBER 5, 2024 7:10 AM EDT

AI chatbots sound eerily convincing—until you ask them about a topic you know a lot about.

[Time](#)

# THE WALL STREET JOURNAL.

[English Edition ▾](#) | [Print Edition](#) | [Video](#) | [Audio](#) | [Latest Headlines](#) | [More ▾](#)

[Latest](#) [World](#) [Business](#) [U.S.](#) [Politics](#) [Economy](#) [Tech](#) [Markets & Finance](#) [Opinion](#) [Arts](#) [Lifestyle](#) [Real Estate](#) [Personal Finance](#) [Health](#)

CIO JOURNAL

## From RAGs to Vectors: How Businesses Are Customizing AI Models

A guide to popular tools and techniques businesses rely on to take generative AI to the next level

# RAG Fun Facts

## 1. RAG was pioneered by Meta AI in 2020

Researchers introduced RAG as a way to combine retrieval and generation, enhancing factual accuracy.

<https://arxiv.org/abs/2005.11401>

## 2. RAG reduces hallucinations—but doesn't eliminate them!

Even with external retrieval, AI can still make things up if the right data isn't available.

## 3. Early search engines pioneered retrieval techniques!

Before RAG, search engines like AltaVista (1995) and Google (1998) developed sophisticated information retrieval methods, including TF-IDF and PageRank, laying the groundwork for modern neural retrieval systems.

## 4. "Chunking" can make or break a RAG system.

If documents are split incorrectly, the model may retrieve unrelated or misleading information. Proper chunking is key!

## 5. RAG mimics human cognition!

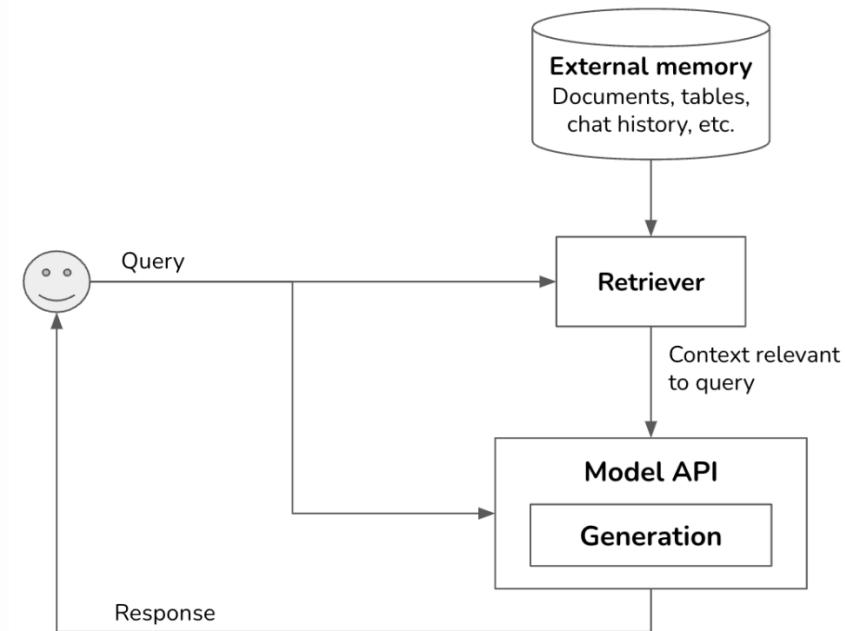
Cognitive scientists have found that humans don't just memorize facts—we retrieve memories and generate new ideas by combining past knowledge creatively. RAG does the same!

# RAG - Foundation Models and Architecture

Building Blocks of RAG, Architecture and Integration, Implementation Steps

# Retrieval-Augmented Generation

RAG consists of two components: a generator (e.g. a language model) and a retriever, which retrieves relevant information from external sources.



**Note:** Retrieval isn't unique to RAGs. It's the backbone of search engines, recommender systems, log analytics, etc. Many retrieval algorithms developed for traditional retrieval systems can be used for RAGs.

# RAG Architecture



## **Input Request**

The process begins with a user request or input query. This query can be a question, a statement, or a task that requires domain-specific knowledge.

## **Retrieval**

The RAG system utilizes an information retrieval component to fetch relevant documents or data points from the designated knowledge base.

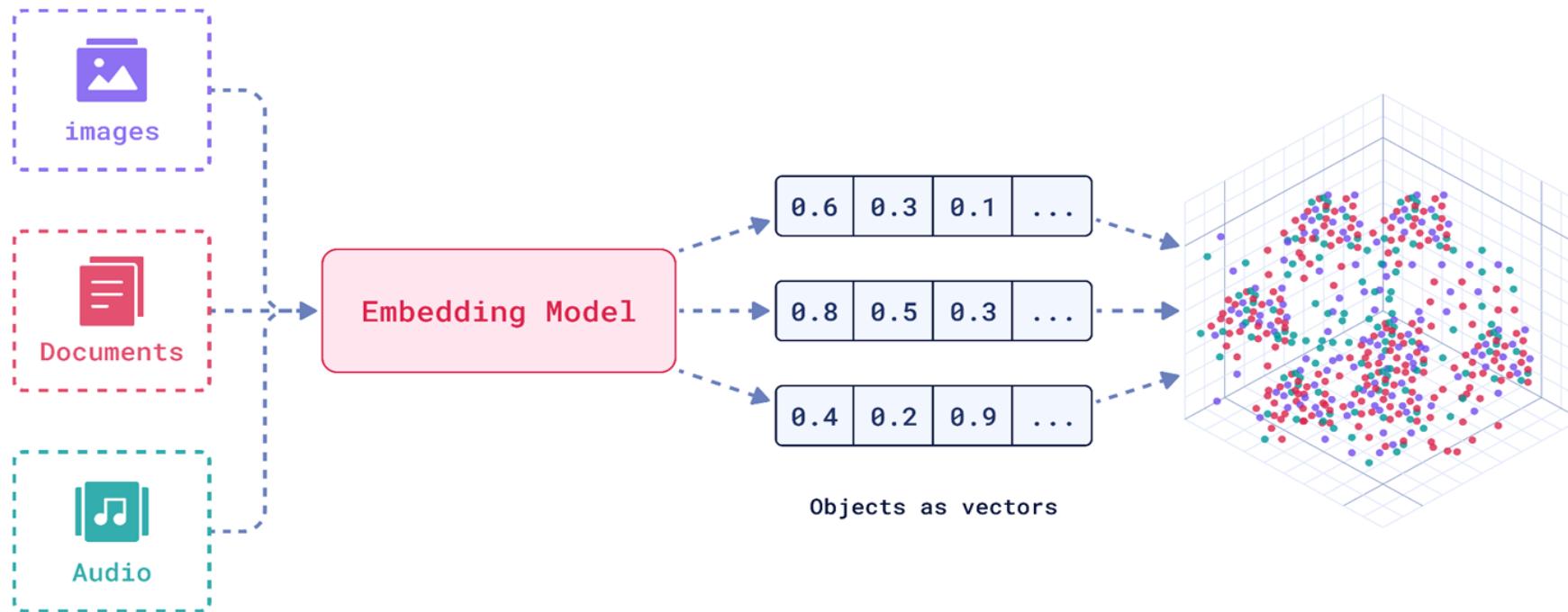
## **Context Enrichment**

The retrieved information is then used to enrich the context of the input request. This context augmentation helps the foundation model understand the nuances of the domain and generate more accurate responses.

## **Generation**

Finally, the foundation model uses the enriched context to generate a response to the input request. This response leverages both the general knowledge of the foundation model and the specific information retrieved from the knowledge base.

# How does it work? You need an embedding model



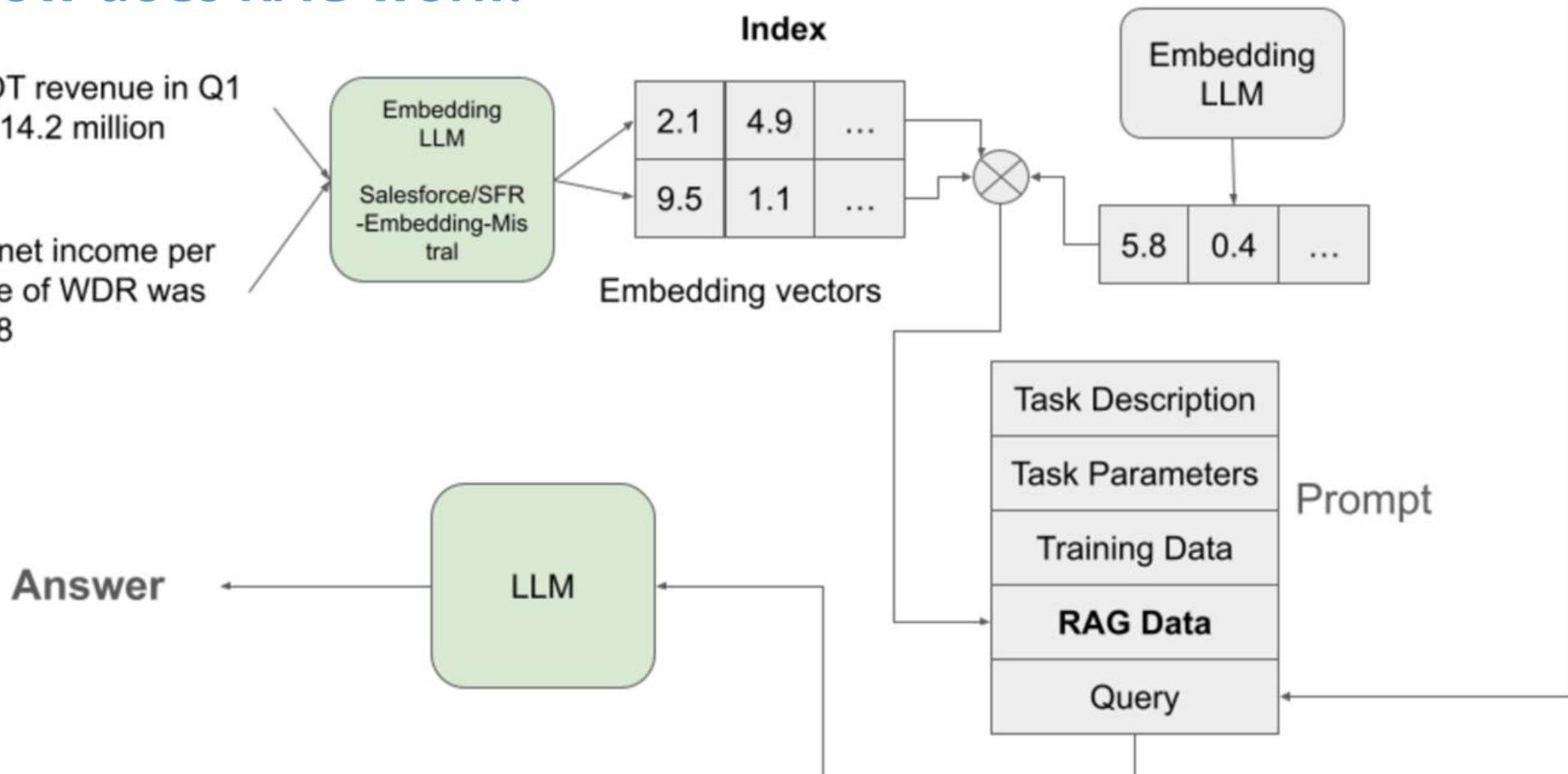
**Vector embeddings convert text into numbers that capture meaning. Similar texts get similar numbers, making it possible to find related information quickly.**

## How does RAG work?

GDOT revenue in Q1 was 14.2 million

The net income per share of WDR was \$0.38

Query: Tell me about GDOT revenue

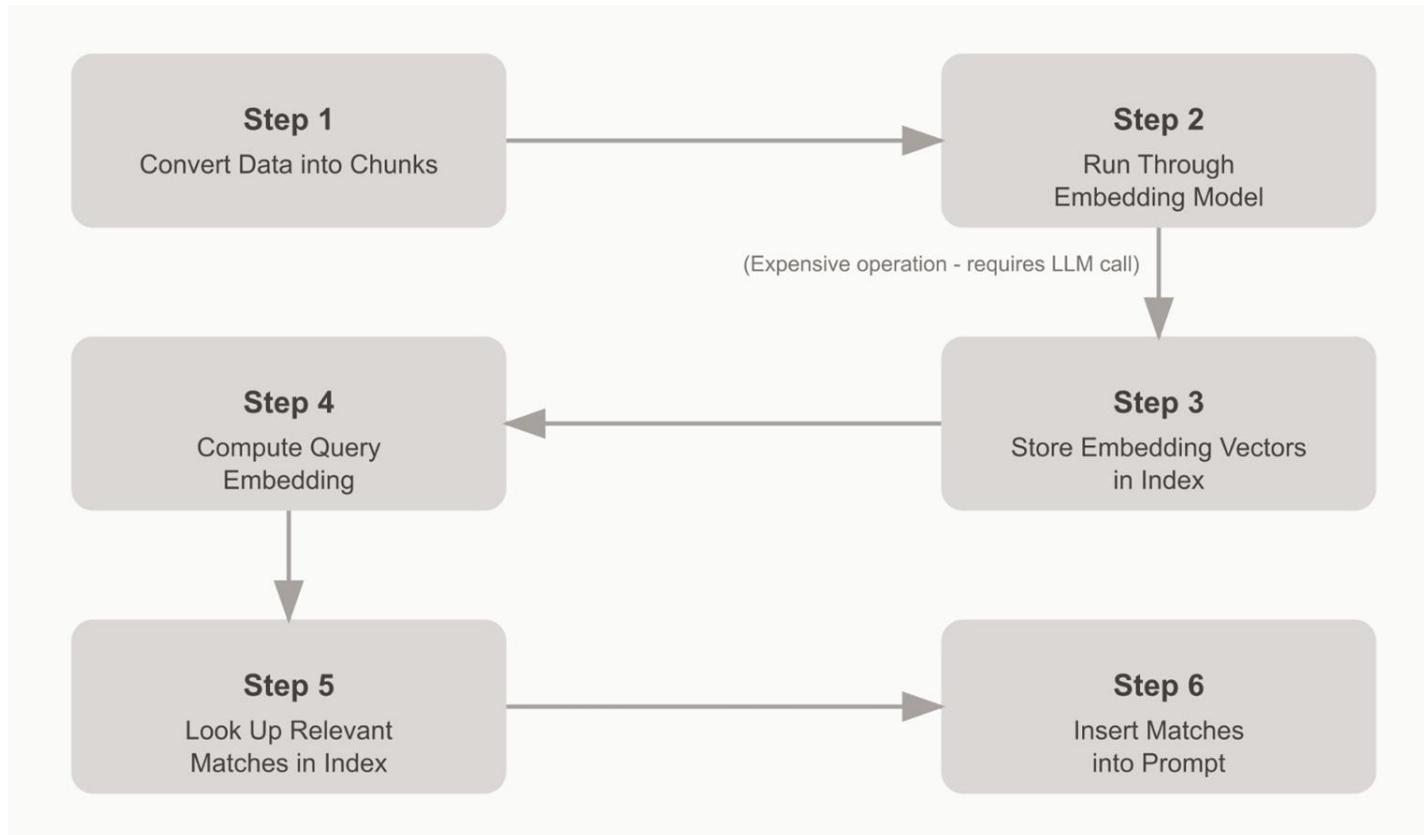


# Stages within RAG



- **Loading:**
  - Retrieve data from text files, PDFs, databases, APIs, or websites.
- **Indexing:**
  - Create a searchable data structure for querying.
  - Generate vector embeddings (numerical representations of meaning).
  - Use metadata strategies to improve relevance.
- **Storing:**
  - Save indexed data and metadata to avoid re-processing.
- **Querying:**
  - Use sub-queries, multi-step queries, or hybrid search methods.
  - Leverage LLMs + LlamaIndex for better responses.
- **Evaluation:**
  - Assess accuracy, faithfulness, and speed of responses.
  - Compare different strategies for optimization.

# RAG Implementation Steps



## Important Considerations

- Chunking affects context relevance and retrieval
- Vector quality depends on embedding model selection
- Retrieval accuracy impacts response quality
- Overall performance depends on the efficiency of both retrieval and LLM processing

# RAG Evolution: From Basic to Advanced

## Basic RAG

- Basic fixed-size chunking (e.g., 300–1000 tokens)
- Dense embeddings with static retrievers
- Simple similarity search (cosine distance)
- Fixed context window (typically 2–4K tokens)
- Primarily used for fact-based QA and internal search

# RAG Evolution: From Basic to Advanced

## Advanced RAG

- *Chunking Strategies:*
  - Semantic chunking: splits based on meaning, not length
  - Hierarchical chunking: combines document, section, paragraph levels
  - Sliding window / sentence overlap: preserves inter-sentence context
- *Advanced Retrieval:*
  - Hybrid retrieval: dense + sparse
  - Reranking: cross-encoders like ColBERT, Cohere Rerank
  - Query rewriting: Query decomposition for multi-hop reasoning

# Why RAG Fails - Understanding the Limitations

- **Multi-Step Reasoning (32%)**: Shows why RAG struggles when information must be found sequentially
- **General Queries (18%)**: Explains retrieval challenges with broad, open-ended questions
- **Complex Queries (15%)**: Addresses difficulties with long, multi-condition questions
- **Implicit Questions (27%)**: Highlights the need for holistic document understanding

[Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach](#)

# Building your own RAG system

- We will now implement RAG in our Open WebUI instance

Follow the instructions on the page “***Powering LLMs with RAG***  
-> ***Creating a Knowledge Base*** to launch your own MCP server

- **Password for PostgreSQL Db:** *amlid2026llmonprem*
- If you run into problems, please approach one of us!
- **Done?** Continue with the Section **Linking a SQL Database**

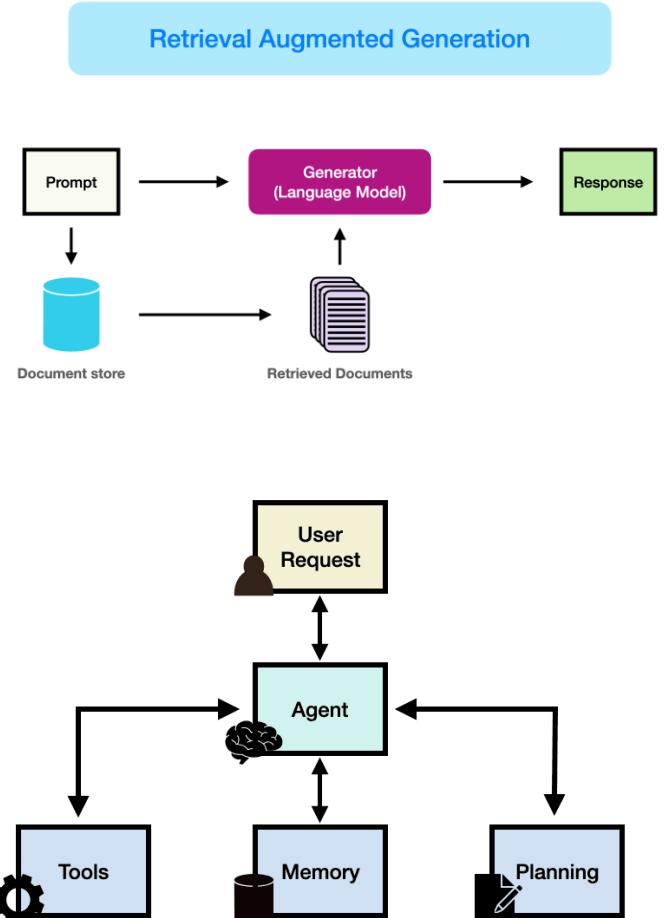
# Agenda

- AI Today: The Generative Landscape
- Serving a LLM On-Prem: Open WebUI
- *Coffee break*
- RAG – Knowledge Bases and Linking a SQL Database
- **How Agents Talk to Tools: Towards MCP**
- Closing Remarks

# Augmenting LLM capabilities

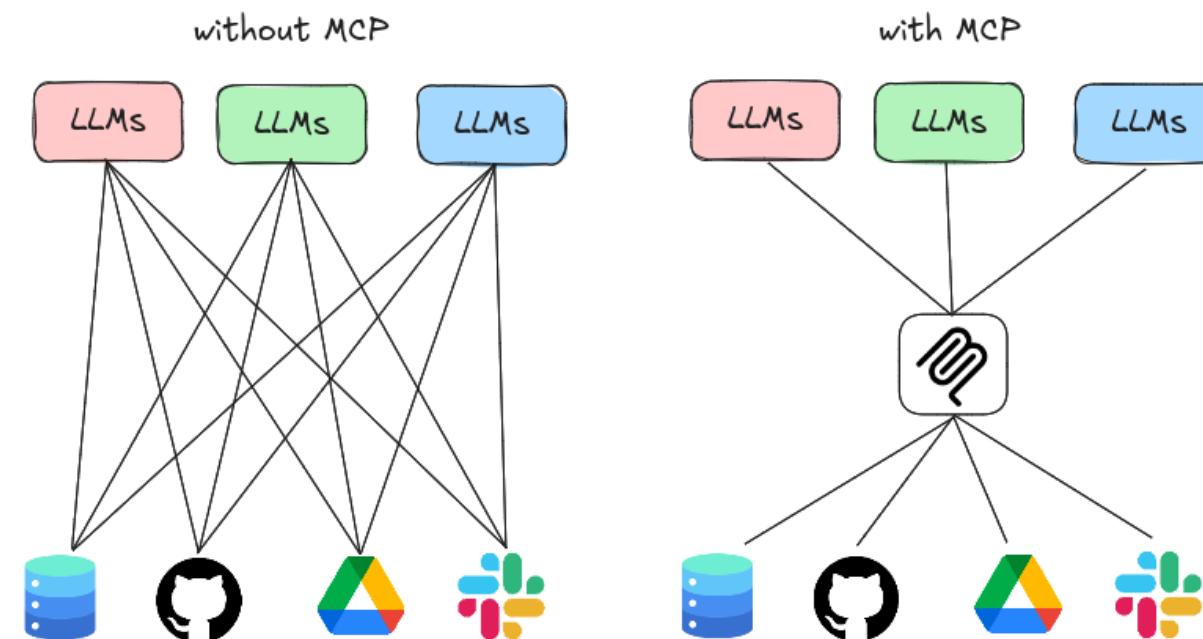
## Various options:

- Prompting strategies
- Retrieval-Augmented Generation (RAG)
  - Using a local knowledge base or websearch
- Agents with tools
  - Code execution, parsing of files, sending emails, ...



# Model Communication Protocol (MCP)

- **Advantage:** LLMs can interact with tools in a structured 2-way connection → no more “glue code” for every connection between a LLM and a tool



# MCP is becoming an industry-standard

- The biggest players are adopting MCP into their products
- Also many community-built MCP servers



# Launching your own MCP server

- We will now set up our own MCP server

Follow the instructions on the page ***Tools and MCP -> Model Context Protocol*** to launch your own MCP server

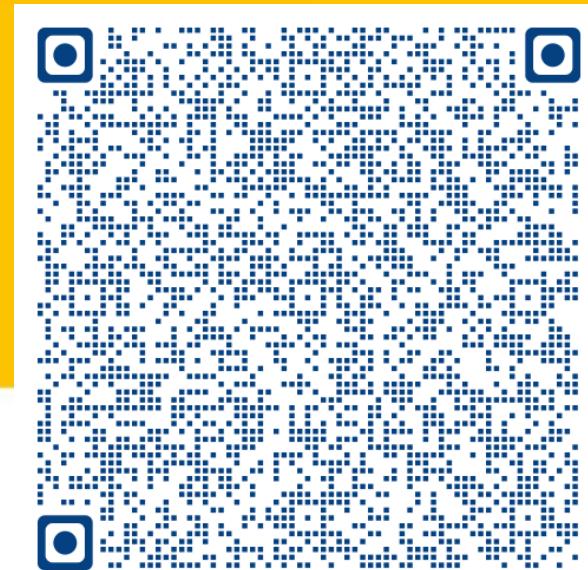
- If you run into problems, please approach one of us!

# More advanced MCP servers

- Have a look at the exercises in **Tools and MCP -> Different Tools**
  - Building MCP servers for different datasets
  - Building MCP servers with LLM- or API-powered tools
- What kind of tools for LLMs would come in handy in your daily workflow?

# Large Language Models and AI Agents

**From foundations to LLM and beyond**  
A bootcamp for compact knowledge



**Program Directors:** Dr. Aygul Zagidullina and Dr. Elena Nazarenko



**Applied Sciences and Arts**  
Continuing Education

# LLMs & AI Agents

Starts February 27

## Module 1 — Foundations of NLP & Transformers

Text, tokens, embeddings, attention, and how Transformer models understand language.

## Module 2 — Large Language Models & Prompt Engineering

How LLMs generate text, how prompts steer probabilities, and how to design reliable instructions.

## Module 3 — Retrieval-Augmented Generation (RAG)

Embeddings, vector search, chunking, ranking, and how to connect LLMs to real data.

## Module 4 — Fine-Tuning & Small Language Models (SLMs)

When and how to adapt models using parameter-efficient fine-tuning and domain-specific models.

## Module 5 — MLOps for LLMs in Production

Deployment pipelines, monitoring, security, cost control, and operating LLMs at scale.

## Module 6 — Evaluation, Product Design & Model Context

How to measure quality, build LLM products, optimize cost vs performance, and manage context.

## Module 7 — Agents & Multi-Agent Systems

Tool-using LLMs, planning, memory, and how to build agent-based workflows.

## Module 8 — Red Teaming, Guardrails & Safety

How to test, break, secure, and control LLM systems in the real world.

# Thank you!

**HSLU** Hochschule  
Luzern

**Hes-SO** VALAIS  
WALLIS  


**ARTEFACT**

**EPFL**  
**AMLD**

**EXOSCALE**

Please take 5 minutes  
to provide us with  
feedback:



**EXOSCALE**

Swiss Cloud - European Trust

Master your stack: a sovereign cloud, built for devs by devs

Code, deploy, scale – with full sovereignty

## Re-run the workshop at home

Nothing complicated: follow these 5 steps.

### Scan the QR code

Open the dedicated coupon page.

### Create your organization

Free Exoscale account. Use an email you check.

### Request A30 GPU access

Go to Instances, click Add, select region CH-GVA-2, then choose instance type GPUA30. Click Enable to submit your access request.

### 100 CHF coupon already applied

Your account is credited with 100 CHF prepaid, usable on GPU.

### Run your demos

If you want to go further and plan the next 90 days, contact us.

GPU access: for governance reasons, activation is not instant. Submit your request now. We can help at any time. Contact support@exoscale.com.



50 CHF coupon

Prefer a link? [Open the coupon page.](#)

Data residency EU/CH • Encryption by default • transparent egress

Managed Kubernetes and databases

Geneva • 2025