From ChatGPT to Agentic AI
The Common Thread of AI Development

INNOVATE. INTEGRATE. EXCEED.

Ivan Herreros Alonso
CAIML Meetup - TH Köln, 8.9.2026

inovex

November 2022 - Today

- >3.5 Years ChatGPT – from Chat Interfaces to autonomous Agents
- A rapid development – but not by chance
- How did we get here?
  → Science + Engineering
- Where are we headed next?
  → What are the relevant questions?

The Science

How can we build an agent based on a next-token prediction machine?

inovex

1. From Monologue ...

... to Dialogue

inovex

“Conversation” as Document Completion

- LLMs have always been document completion machines
  → Input: beginning of a document. Output: most likely continuation.

- The trick: structure the document like a script
  → Character 1 (human) asks, character 2 (AI) responds

The Script

```
User: Hello, how are you?
Assistant: I’m doing well, thank you! How can I help you
today?
User: What is the capital of France?
Assistant:
```

inovex

The Script

```
User: Hello, how are you?
Assistant: I'm doing well, thank you! How can I help you
today?
User: What is the capital of France?
Assistant: The capital of France is Paris.
```

- The LLM writes the next lines of the script
- With a stop-generation rule and prompt templating, building a PoC chat application was straightforward.

The Script

```
User: Hello, how are you?
Assistant: I’m doing well, thank you! How can I help you
today?
User: What is the capital of France?
Assistant: The capital of France is Paris.
User: Tell me more about it.
Assistant:
```

The Script

```
User: Hello, how are you?
Assistant: I'm doing well, thank you! How can I help you
today?
User: What is the capital of France?
Assistant: The capital of France is Paris.
User: Tell me more about it.
Assistant: Paris is the largest city in France...
```

- The LLM doesn't "talk" – it completes a document that *looks like* a conversation
- The user experiences it as talking to an AI.
- ChatGPT refined this idea through reinforcement learning.

inovex

2. From the Document into the World

From the Document into the World

- The LLM still *lives* inside the document
- But the LLM and the backend *agreed* on a “secret code”
  → The LLM writes: <function>get_weather("Barcelona")</function>
  → The backend recognizes the pattern and executes the function
- For the first time, LLMs could interact with the world *beyond the document*

inovex

Tool calling from the LLM's perspective

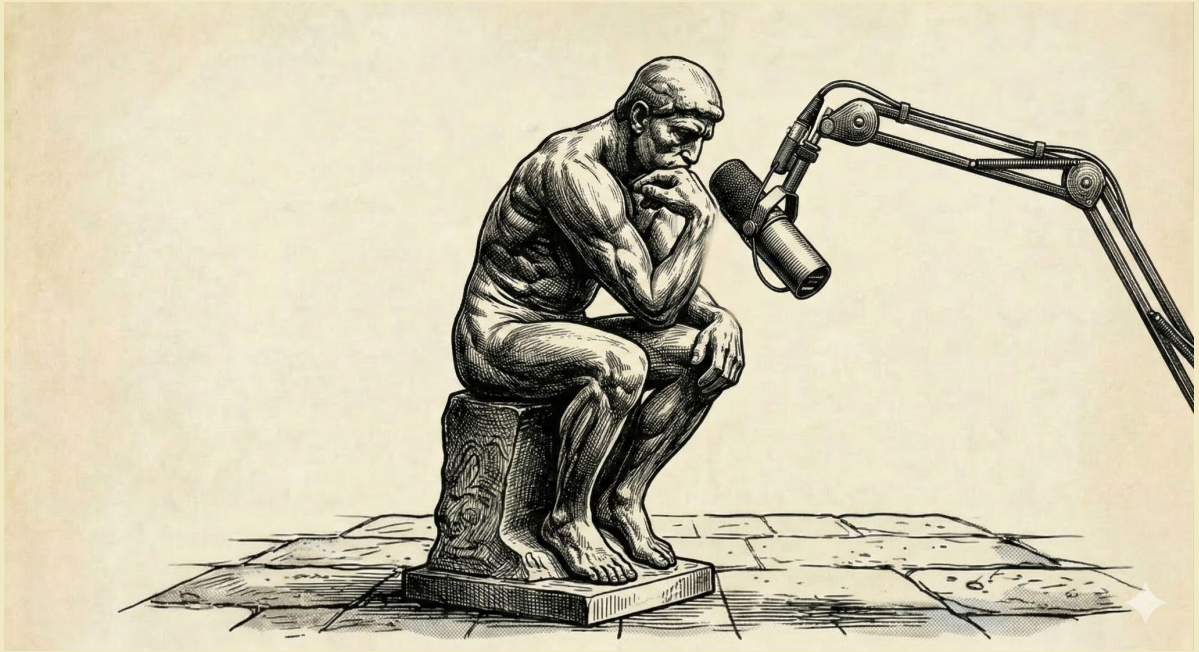
```
User: "What's the weather in Barcelona?"

LLM generates:
{
  "function": "get_weather",
  "arguments": {
    "city": "Barcelona", "country": "ES"
  }
}
System executes -> returns: {"temp": 24, "condition": "sunny"}

LLM generates:
"The weather in Barcelona is currently 24°C and sunny."
```

(A typical example, but also a *poor* example of tool calling)

Function calling made the entire digital world potentially accessible to LLMs

inovex

3. Thinking aloud



inovex

Reasoning Aloud: A Buffer for Thinking

- Two questions
  1. 2 + 2 = 4? Answer yes/no
  2. <20-page legal document>. Is this employment contract fair, and should I sign it? Answer yes or no
- Early LLMs had the same “thinking budget” for every question

Reasoning Aloud: A Buffer for Thinking

- The trick: don’t change the model – give it room to think
  - First came chain-of-thought prompting: “Solve it step-by-step”
  - Later came “reasoning” models (Sept. 2024)

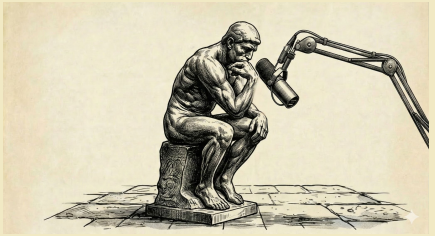
“Transformers –i.e., LLMs– need tokens to think” (Karpathy, 2023)

inovex

Building Blocks of Agentic AI

Speaking

Acting



Thinking

With these capabilities in place, it was no longer a question of **if**, but **when**

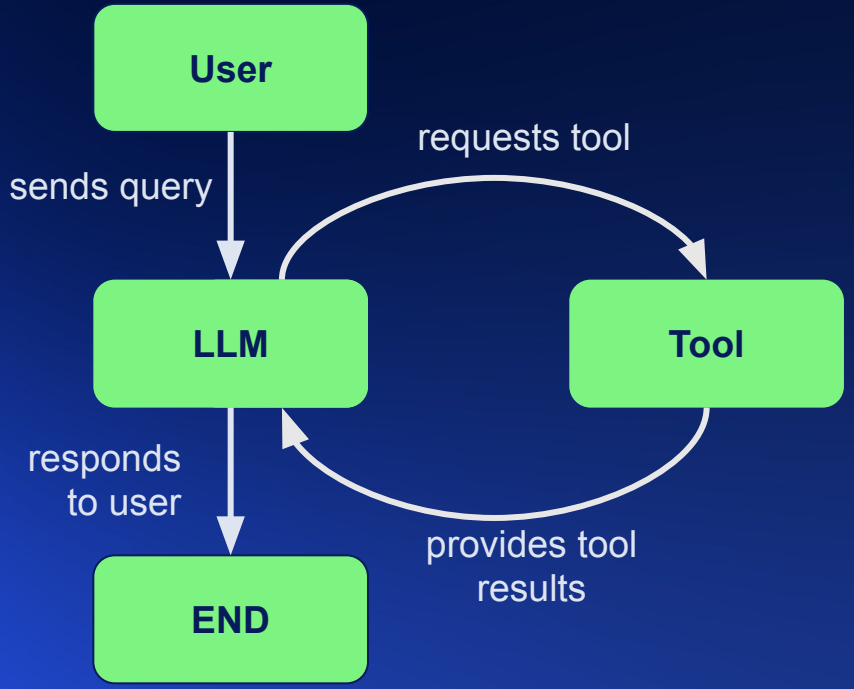
Since when have we known that all the building blocks for agentic AI were in place?

"*ReAct: Synergizing Reasoning and Acting in Language Models*" (Yao et al., 2022)

- Academic paper introducing the idea of implementing “agentic loops” to improve task performance
- Had a profound impact on the LLM community

The ReAct Agent

- ReAct introduces the concept of the “agentic loop”
- To answer a query, the LLM can iterate and call as many tools as necessary



The ReAct paper *predates* the release of ChatGPT

If everything was already known by 2022, why are we only talking about agentic AI in 2026?

- **LLM side**: Models simply weren’t capable enough yet
- **Backend side**: The necessary tools, architectural patterns, and integration standards were still missing

The Engineering

How do we build backends that enable useful (and safe) agents?

inovex

Optimizations

Foundation Tools
Which tools does almost every agent need (and work particularly well with)

Integrations with external systems
How does an agent reach the systems we actually run?

Harnesses
How do we orchestrate the whole thing *and keep it safe*?

Foundation Tools

Their shape is dictated by basic properties of how LLMs work and are trained

Web search → overcomes LLMs one structural deficit: they cannot form new memories

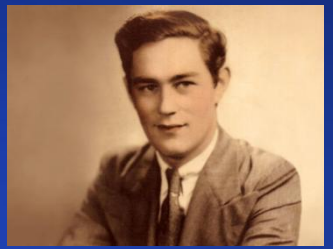
Code execution → exploits their one disproportionate strength: they were trained on code

The "Anterograde Amnesia" Problem

- LLMs are implemented as neural networks
  - “**Training**”: Knowledge is encoded in the model’s weights
  - “**Inference**”: The weights generate outputs but are not modified

- **LLMs have a *Cutoff Date***

- They work much like patient H.M.



Web-Search Tool: Partial Solution to the Cutoff Problem

- Allows the LLM to retrieve current information
- This allows it to answer questions such as
  “*Who won the 2026 World Cup?*”
- But
  → *LLMs* still *do not learn from our interactions*; they can only “take notes”
  → The anterograde amnesia problem remains

Code-Execution (or Bash) Tools

- The LLM can write code (or cli instructions) and the backend executes it in a sandbox (or in your computer)

- What this enables:
  → Code Interpreter: data analysis, transformation and visualization
  → Bash tool: ⚠️⚠️⚠️ anything a computer can do in text mode ⚠️⚠️⚠️

- Both are **open-ended tools**: you provide the tool once and it becomes better as models become better
  → counter-example: get_weather tool

Integrations with external systems

- **2023 - 2024: How can I use an LLM with my own data?**

  → Retrieval-Augmented Generation (RAG)
  → Built from scratch for every system, every project

  The question was too narrow

- **2025 - now: How can an AI Agent act on my digital ecosystem?**

  → Read, write, edit (and build it for all systems at once)
  → Model Context Protocol (MCP)

Model Context Protocol (MCP)

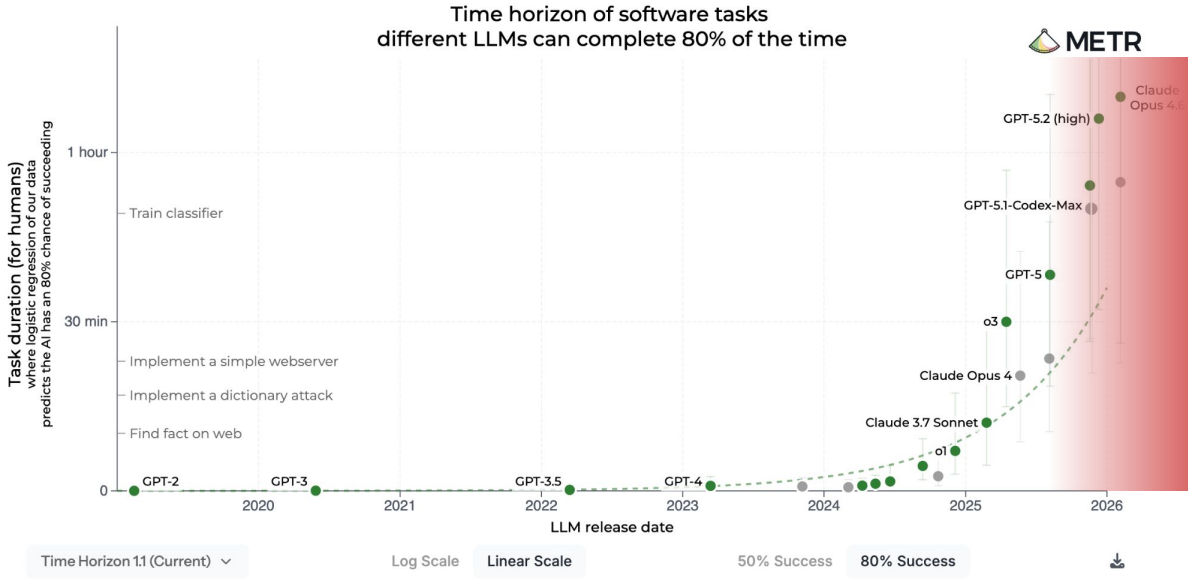
- A standard interface, proposed by Anthropic and adopted across the industry (e.g. Atlassian, GitLab, Slack, Google Drive, AWS, ...)
- As a result, far fewer ad-hoc implementations: instead, implement one standard client and plug into any system.

Harnesses

- Problem: left on its own, an LLM fills up its context window and drifts
- A harness is a backend that enables advanced, secure LLM-powered agents
- The agent harness “orchestrates” the LLM:
  → External Memory (File System) → Sub-agents → Planning …
- You are probably already using one (Claude Code, Manus, Hermes, …) or building your own (with DeepAgents by LangChain, …)
- First (partial) step beyond the “conversation-centred” ReAct agent blueprint

inovex

Agentic AI



The exponential improvement in LLM capabilities is why we are talking about agentic AI today (and not years ago)

TL;DR

How the relevant questions have changed in 4 years

2022: "How can LLMs use tools? How can they solve problems step-by-step?"

2026: "How do we architect systems in which LLMs can perform hours-long autonomous tasks without causing a major security incident?"

***2027**: "How do agents work for us when we're not in a conversation at all?"*

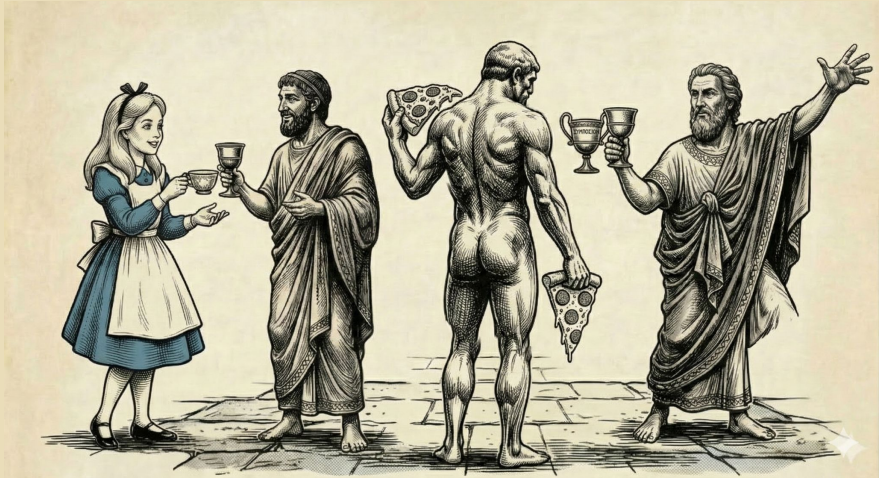
The Common Thread

Why Has Agentic AI Been Inevitable Since 2022?

- **“NLP is AI-complete”:** Whoever masters language can, in principle, master intelligence
  (Montalvo, 1987; cf. Collins, 2003, MIT)
- The moment LLMs mastered language, the rest was only a matter of time
- **Speaking + Acting + Thinking → Agents**: Not a coincidence: a causal chain

inovex

Thank you!



inovex

Ivan Herreros Alonso
Senior AI Consultant | ML Engineer
inovex GmbH

www.linkedin.com/in/ivan-herreros-b64a204