CIKLUM | SPEAKER'S CORNER

October, 02 | 17:00 (CET) | English

How to build multi-turn agents

Max Pavlov

AI Engineering Director at Ciklum

Experiences of tomorrow. Engineered together.

CIKLUM

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

What we do:

- Product Engineering
- Intelligent Automation
- Data & Analytics

2002 founded

4000+ professionals

20+ offices

300+ clients

Leading companies choose us:

JUST EAT Takeaway.com · FLIXBUS · METRO MARKETS · eToro · ZURICH · pay · Greensill · KANTAR RETAIL · dacadoo

Speaker

CIKLUM

Max Pavlov

AI Engineering Director

- In Software Engineering since 2003
- Former Head of .NET in Ciklum
- Lead 120+ FTE teams
- Former Engineering Director (NewSignature) and Solution Architect (Novax, Deloitte)
- Starting 2021 worked as Global Solutioning Director
- Currently part of AI R&D Developing next generation of Ciklum offerings and product in AI

Agenda

CIKLUM

01 AI Agents - definition

02 ReAct paradigm

03 Build from scratch vs off-the-shelf alternatives

04 Environments: ollama vs llama_cpp

05 Coding iterative agent

06 Advancing agent: styling

07 Advancing agent:
context counting and compacting

AI Agents-definition

CIKLUM

What is an AI Agent?

Autonomous system that perceives, reasons, acts, and learns to achieve goals.

Key characteristics

- Access to tools (e.g., terminal commands).
- Iterative problem-solving (loop until resolved).
- Powered by LLMs for decision-making.

LangGraph, for instance, define **tool calling**, **memory** and **planning** as key characteristics of flexible agents

Why Iterative?

Real-world tasks often need multiple steps; agents "think" and "act" in loops.

The ReAct Framework

CIKLUM

What is Re-Act?

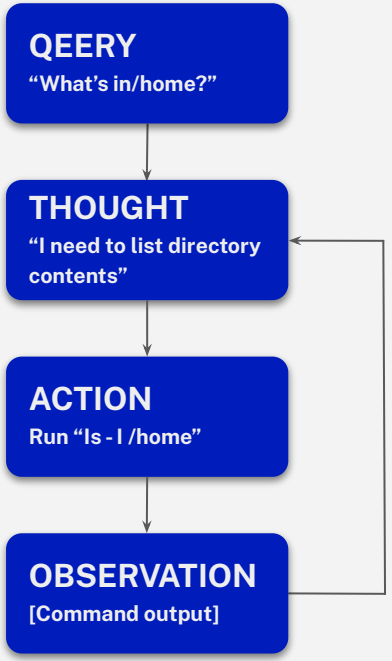
Reasoning + Acting — a simple pattern for agents to alternate between thinking and tool use.

How it Works:

- Reason: LLM thinks step-by-step about the query.
- Act: Decide on an action (e.g., tool call like "ls -l").
- Observe: Execute action, feed result back to LLM.
- Repeat until final answer.

Benefits:

Transparent (due to agent's thought process), flexible, powerful..



Build from scratch vs off-the-shelf alternatives

CIKLUM

Why build from scratch?

Better for learning. No core implementation hidden behind library facade.

Alternatives

- CrewAI agents
- LangGraph agents

What would I use in production?

Deeper analysis needed. For value focused implementations LangGraph offers good out-of-the-box value.

If agent is the core product-probably build from scratch to maintain full control

Environment setup

CIKLUM

Where to get LLM for thinking?

Core thinking abilities of agents come from an underlying model. LLMs or SLMs are at core of any agent, although logical orchestration is also important

When choosing a model

- Focus on models trained for tool calling
- Models have their own tool calling notation but even with custom prompt-these are better at identifying the need for tool calling
- Models with 8B+ params show incredible tool calling abilities

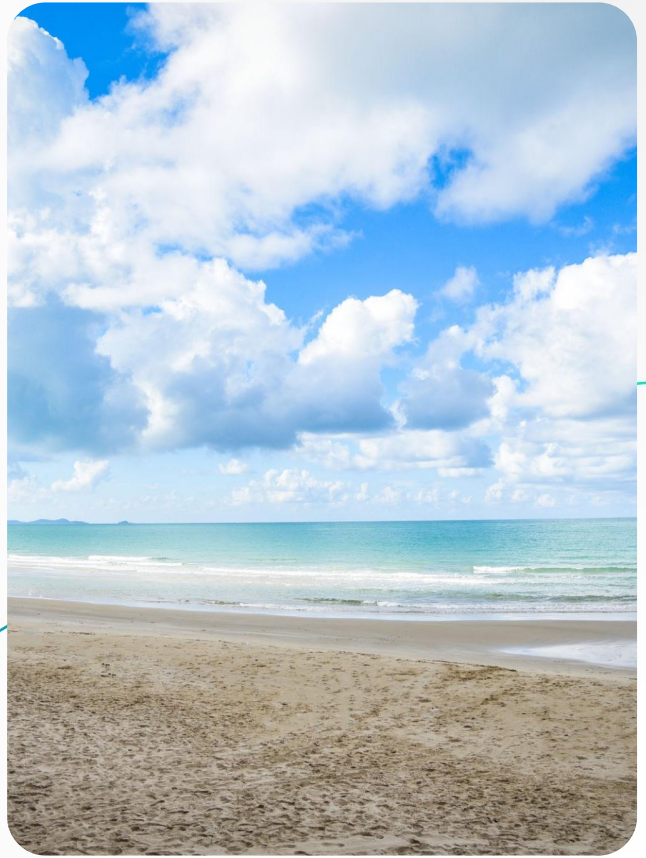
Application stack?

One needs to maintain the balance between control, performance and ease of use of an LM inference. You don’t want to bring in own MLOps pipeline, neither you want to only get one LLM API. Ollama is good for demo purposes. LLama.cpp is performant and format-agnostic enough to be the local inference go-to engine even in production

CIKLUM

DEMO 01

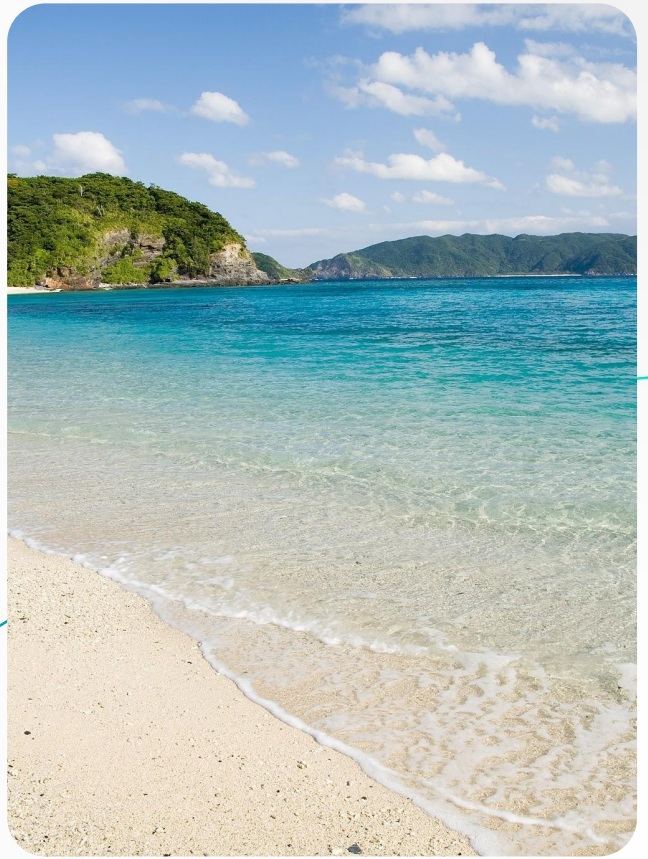
Coding a simple agent



CIKLUM

DEMO 02

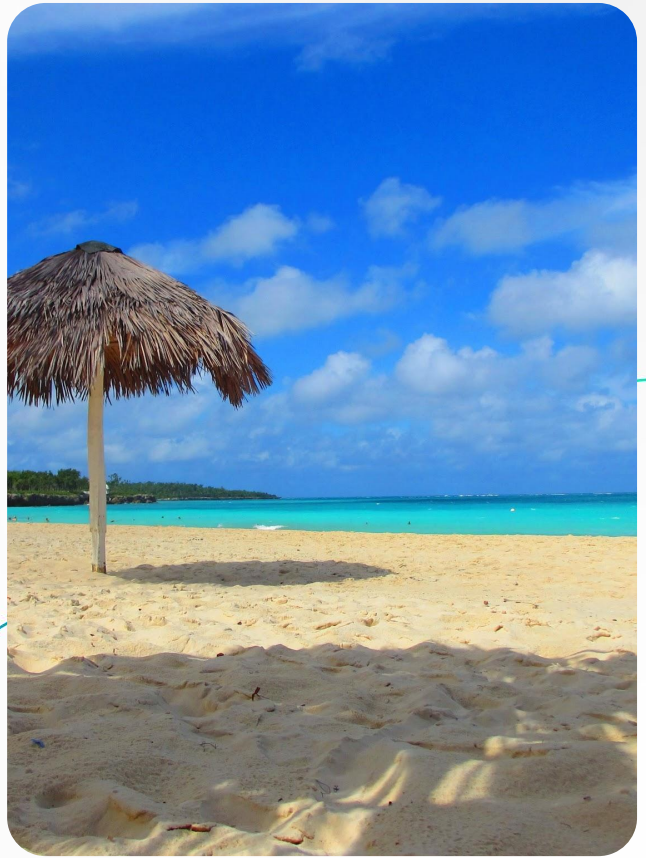
Styling agent output



CIKLUM

DEMO 03

Thinking about context utilization and compacting



CIKLUM

Q&A

CIKLUM

Share your **feedback!**

CIKLUM

Join our team

Thank you!