

LangChain v1 Explained: Agents, Tools, and Beyond

A large, dark blue hexagonal badge with rounded corners is centered on a background of overlapping purple and blue curved shapes. The badge contains the text "OSS AI" on top and "Summit" below it, all in white.

OSS AI
Summit



Reactor

Who are we?



Marlene Mhangami
Senior Developer Advocate
@ Python Team



Yohan Lasorsa
Principal Developer Advocate
@ JavaScript Team

LangChain? 🤔

- Open-source framework for building LLM-powered apps
- Provides high level abstractions (models, agents, memory...)
- Bridges data + reasoning
- Active community and momentum

Core overview

Component	Usage
Models	LLM and embeddings abstraction
Messages	Input/output content abstraction
Tools	Functions that can be called by agents
Agents	Loop models and tools towards a goal
Middleware	Hook into agents execution steps
Memory	Store interactions and checkpoints
Streaming	Surface real-time updates

Core overview

Component	Usage
Models	LLM and embeddings abstraction
Messages	Input/output content abstraction
Tools	Functions that can be called by agents
Agents	Loop models and tools towards a goal
Middleware	Hook into agents execution steps
Memory	Store interactions and checkpoints
Streaming	Surface real-time updates

Core overview

Component	Usage
Models	LLM and embeddings abstraction
Messages	Input/output content abstraction
Tools	Functions that can be called by agents
Agents	Loop models and tools towards a goal
Middleware	Hook into agents execution steps
Memory	Store interactions and checkpoints
Streaming	Surface real-time updates

Agents are the focus

LangChain

- Standard agent loop with `create_agent`
- Common conversational agents
- Simple and high-level

LangGraph

- Graph-based execution for fine-grained orchestration
- Custom workflow and automation
- Full control, more low-level

Middlewares

Customizable hooks in the agent loop for dynamic control

Built-in Middlewares

- **Human-in-the-Loop:** User approval before sensitive actions
- **Summarization:** Condense chat history to fit token limits
- **PII Redaction:** Detect and redact sensitive data

Deep Agents

Agents with extended **autonomy** and **persistence**.

Extra Capabilities

- **Persistent Memory:** Stores knowledge across sessions
- **Filesystem Access:** Read/write/edit files locally
- **Planning:** Break tasks into steps, built-in todos
- **Subagents:** Delegate to specialized subagents
- **Long-Running Workflows:** Resume after interruptions

Free course

LangChain for Beginners



Thank you!



Any question? Join us on
Microsoft Foundry Discord

aka.ms/oss-ai-summit

OSS AI
Summit