HITACHI
Inspire the Next

Agentic System

Developers:

Anh Quoc Nguyen

Anh Thi Ngoc Huynh

Hitachi Digital Services

Feb 2025

Agenda

1 Agents and Frameworks

2 From Single-Agent to Multi-Agent

3 Challenges & Future Work

4 Q&A

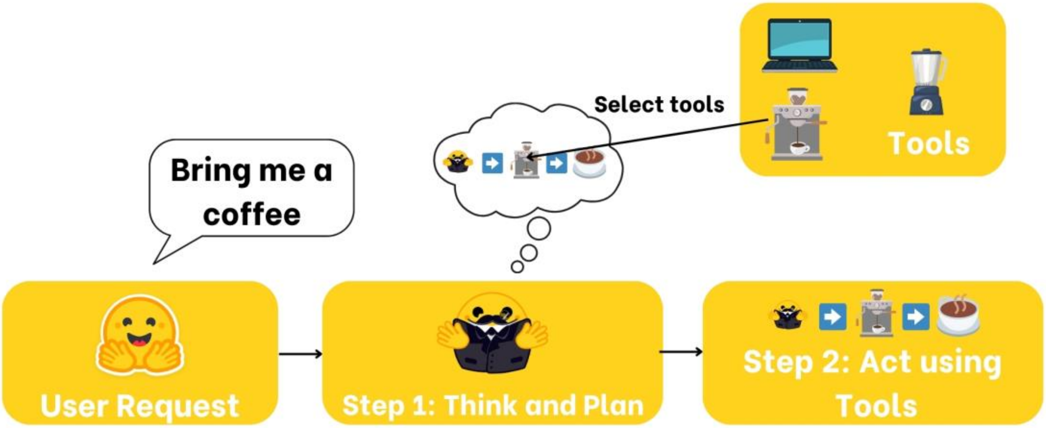
Agents and Frameworks

What are they?

Presenter:
Anh Thi Ngoc Huynh

What are AI agents?

- An Agent is a system that leverages an AI model to interact with its environment in order to achieve a user-defined objective. It combines reasoning, planning, and the execution of actions (often via external tools) to fulfill tasks.
- Agents work in a continuous cycle of: thinking (**Thought**) → acting (**Act**) and observing (**Observe**).



6

LLMs in AI Agents

- LLMs are a key component of AI Agents, **providing the foundation for understanding and generating human language.**

Claude 3 by Anthropic:
A model offering contextual understanding and multi-lingual proficiency.

GPT-4o by OpenAI:
A popular model known for its versatility and wide range of applications.

Llama 3.1 by Meta:
A resource-light, customizable model used for customer service and content creation.

Gemini 1.5 Pro by Google:
A multimodal model that handles text, images, and other data types.

PaLM 2 by Google:
A powerful model with extensive capabilities in natural language processing.

Grok-2 by xAI (Elon Musk):
A model adept at natural language processing, ML, and image generation.

Mistral 7B by Mistral AI:
An open-source model known for its high performance and innovative architectures.

Falcon 180B by Technology Innovation Institute:
An open-source model with a large parameter count.

Inflection-2.5 by Inflection AI:
A resource-light model known for coding and math, with integrated search capabilities.

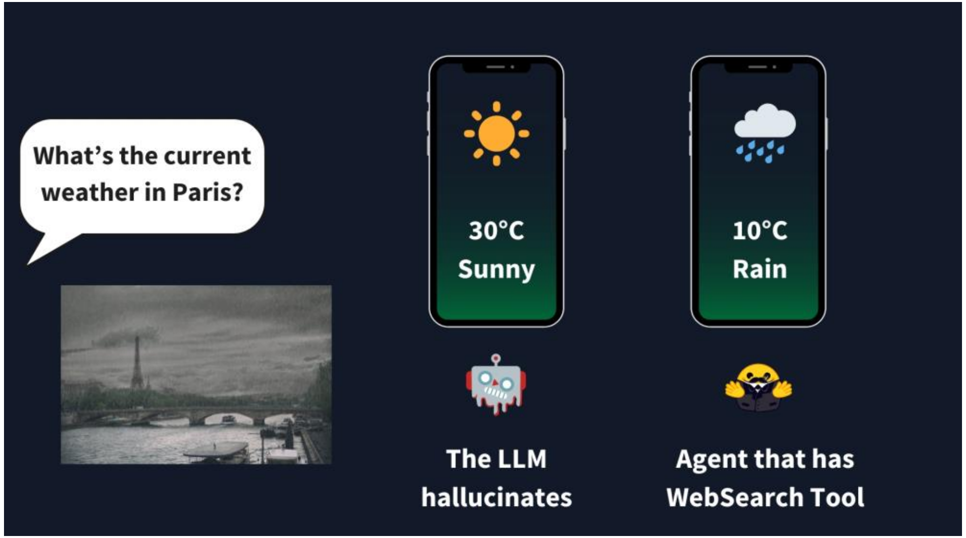
Command R by Cohere:
An open-source or proprietary model, known for its versatility.

Stable LM 2 by Stability AI:
A model known for its stability and efficiency in multilingual text processing and more.

Phi-3 by Microsoft:
Small language models known for high performance and cost-effectiveness.

Tools in AI Agents

- LLM agent tools can be intrinsic, embedded in your LLM, external, called upon when needed, or hybrid, a combination of the two.
  - **Intrinsic tools are built-in to your LLM:** Text processing, Natural Language Understanding (NLU), Natural Language Generation (NLG).
  - **External tools interact with other systems:** Database queries, Database queries, Custom logic


What's the current weather in Paris?
30°C
Sunny
10°C
Rain
The LLM hallucinates
Agent that has WebSearch Tool

Why Use Agents?

Feature	Agentic AI	Generative AI	Traditional AI
Primary Function	Goal-oriented action & decision-making	Content generation (text, code, images, etc.)	Focused on automating repetitive tasks
Autonomy	High – Operates with minimal human oversight	Variable – May require user prompts or guidance	Low – Relies on specific algorithms and set rules
Learning	Reinforced Learning – Improves through experience	Data-driven learning – Learns from existing data	Relies on predefined rules and human intervention

LangChain Framework

❖ Introduction

LangChain: A framework for developing applications powered by large language models (LLMs). LangChain simplifies every stage of the LLM application lifecycle: Development, Productionization, Deployment.

🦜🔗 LangChain

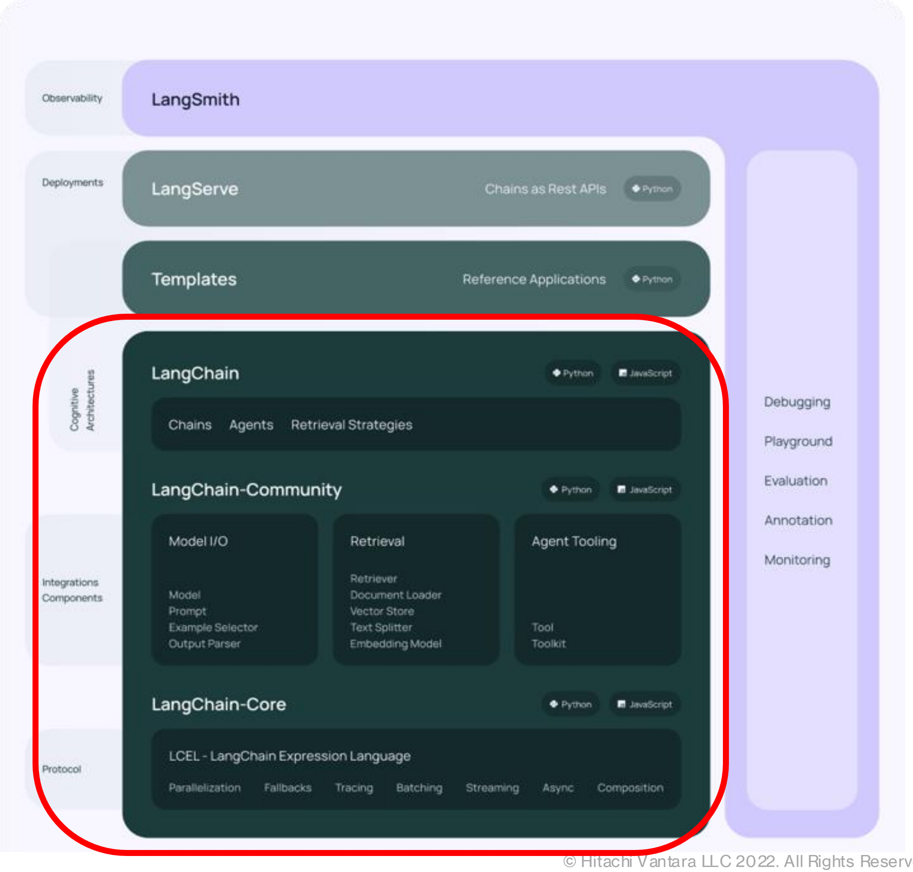
LangChain Framework

HITACHI
Inspire the Next

❖ Introduction

Development: Build your applications using LangChain's open-source **building blocks** and **components**. Hit the ground running using **third-party integrations** and **Templates**.

LangChain



Observability
LangSmith

Deployments
LangServe
Chains as Rest APIs
Python

Templates
Reference Applications
Python

Cognitive Architectures
LangChain
Python
JavaScript
Chains
Agents
Retrieval Strategies

Integrations Components
LangChain-Community
Python
JavaScript

Model I/O
Model
Prompt
Example Selector
Output Parser

Retrieval
Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model

Agent Tooling
Tool
Toolkit

Protocol
LangChain-Core
Python
JavaScript
LCEL - LangChain Expression Language
Parallelization
Fallbacks
Tracing
Batching
Streaming
Async
Composition

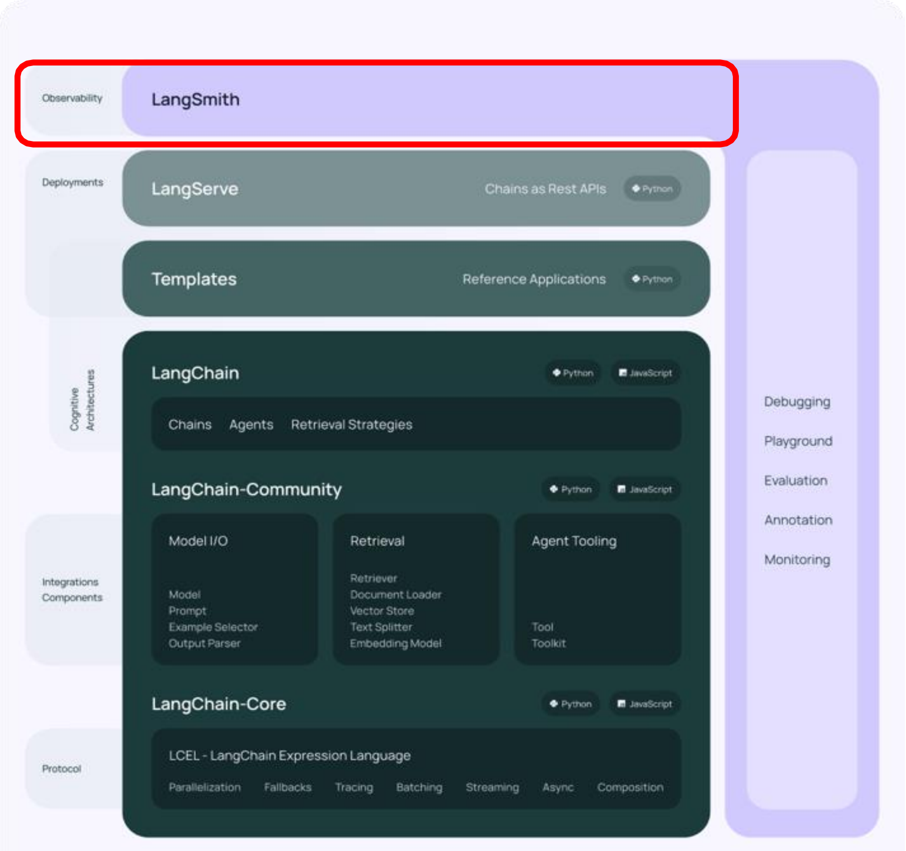
Debugging
Playground
Evaluation
Annotation
Monitoring

LangChain Framework

HITACHI
Inspire the Next

❖ Introduction

LangSmith: used to inspect, monitor and evaluate your chains, so that you can continuously optimize and deploy with confidence (productionization).



Observability — LangSmith

Deployments — LangServe — Chains as Rest APIs — Python

Templates — Reference Applications — Python

Cognitive Architectures — LangChain — Python — JavaScript

Chains Agents Retrieval Strategies

LangChain-Community — Python — JavaScript

Integrations Components

Model I/O: Model, Prompt, Example Selector, Output Parser

Retrieval: Retriever, Document Loader, Vector Store, Text Splitter, Embedding Model

Agent Tooling: Tool, Toolkit

Protocol — LangChain-Core — Python — JavaScript

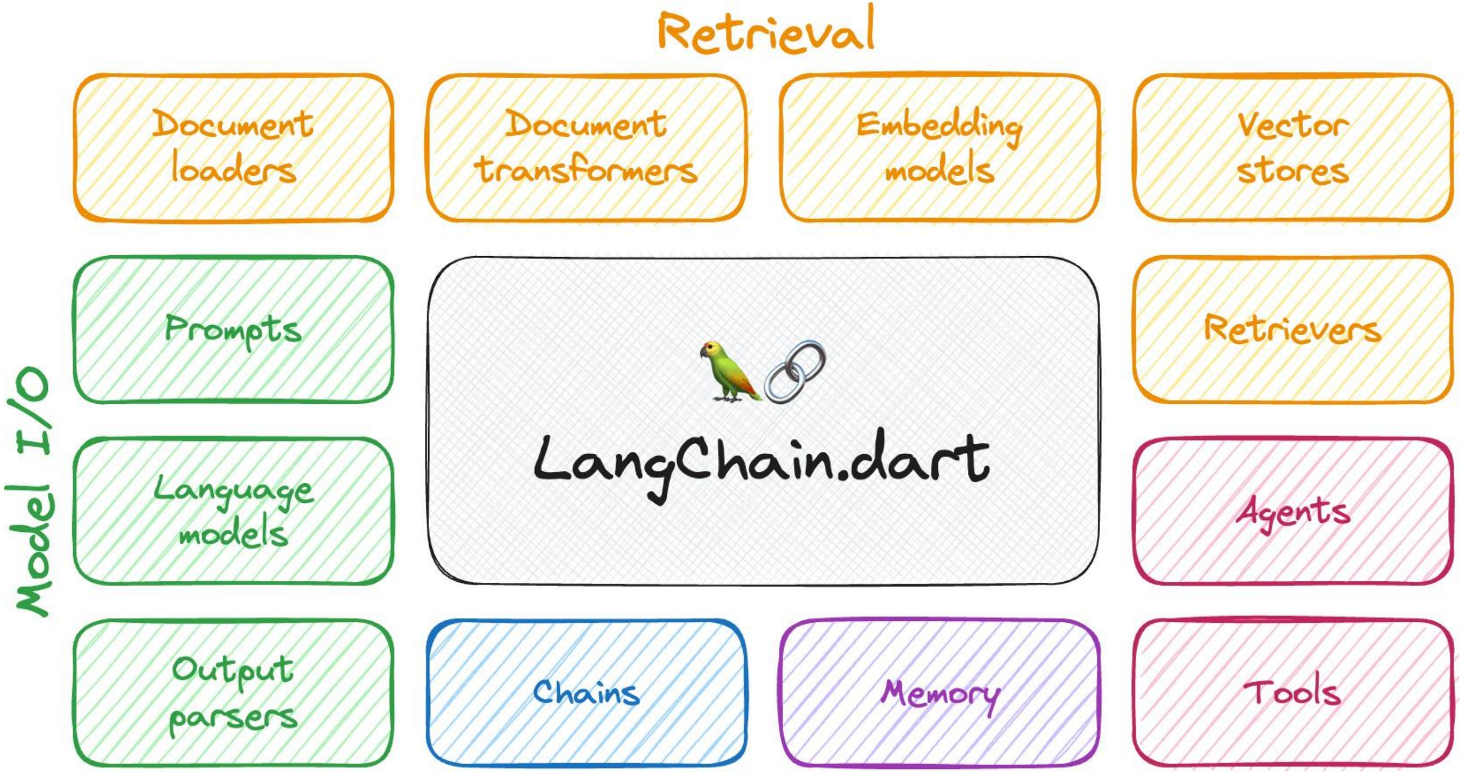
LCEL - LangChain Expression Language

Parallelization Fallbacks Tracing Batching Streaming Async Composition

Debugging
Playground
Evaluation
Annotation
Monitoring

LangChain Framework

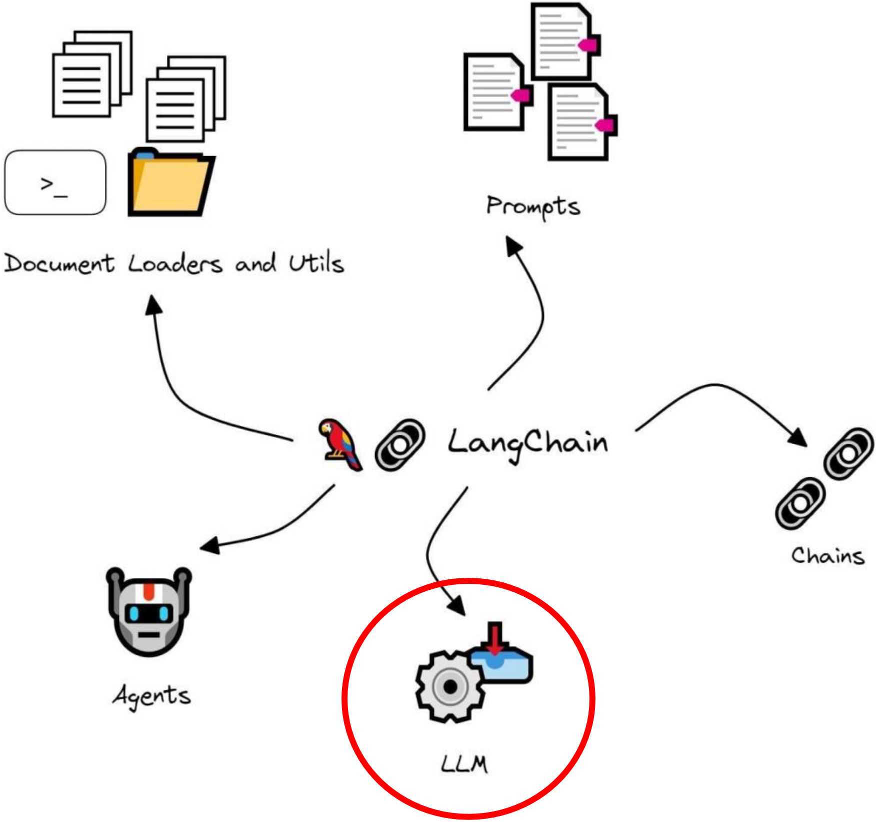
❖ LangChain Components



LangChain Framework

❖ LangChain Components

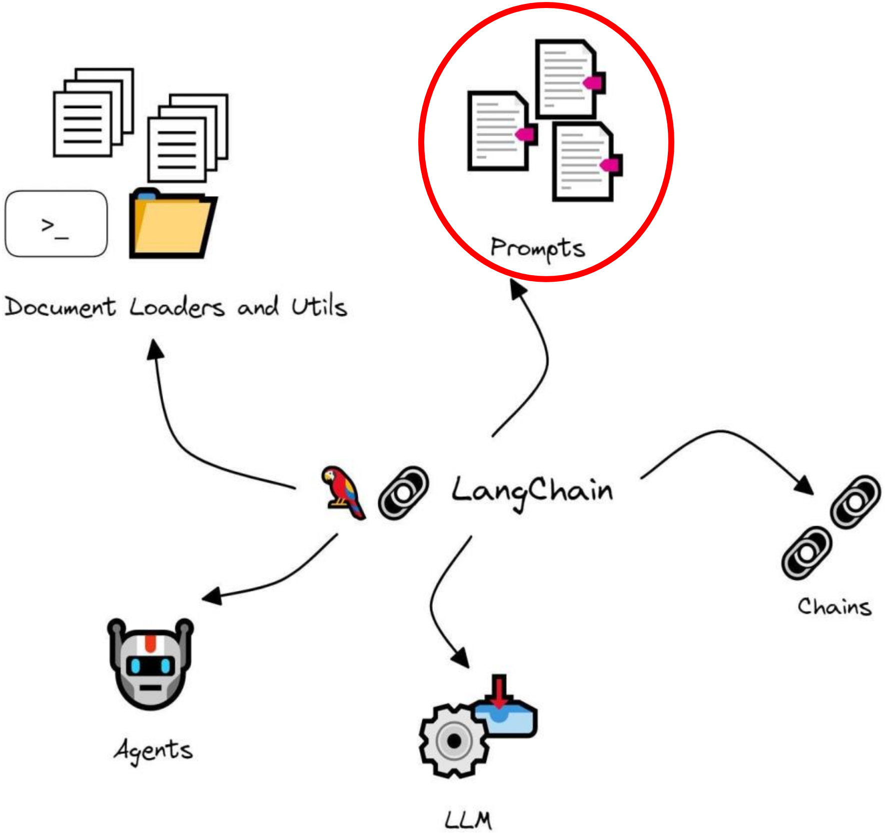
LLM (Large Language Model): The core AI model responsible for processing natural language and generating outputs.



LangChain Framework

❖ LangChain Components

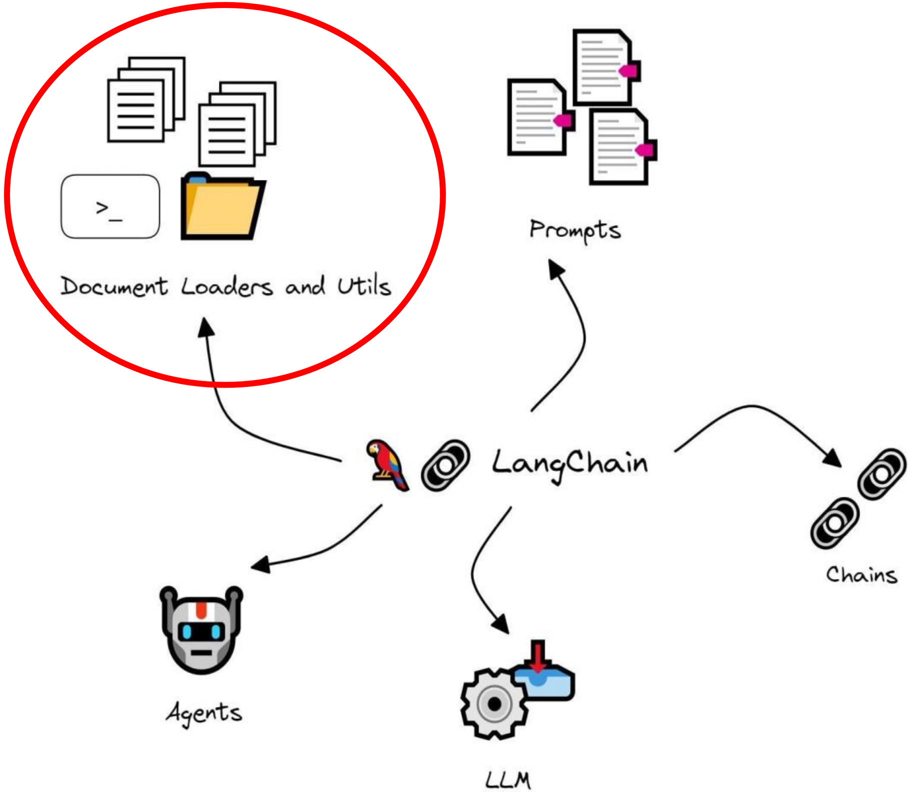
Prompts: Template or structures for crafting natural language prompts that can be fed into the LLM.



LangChain Framework

❖ LangChain Components

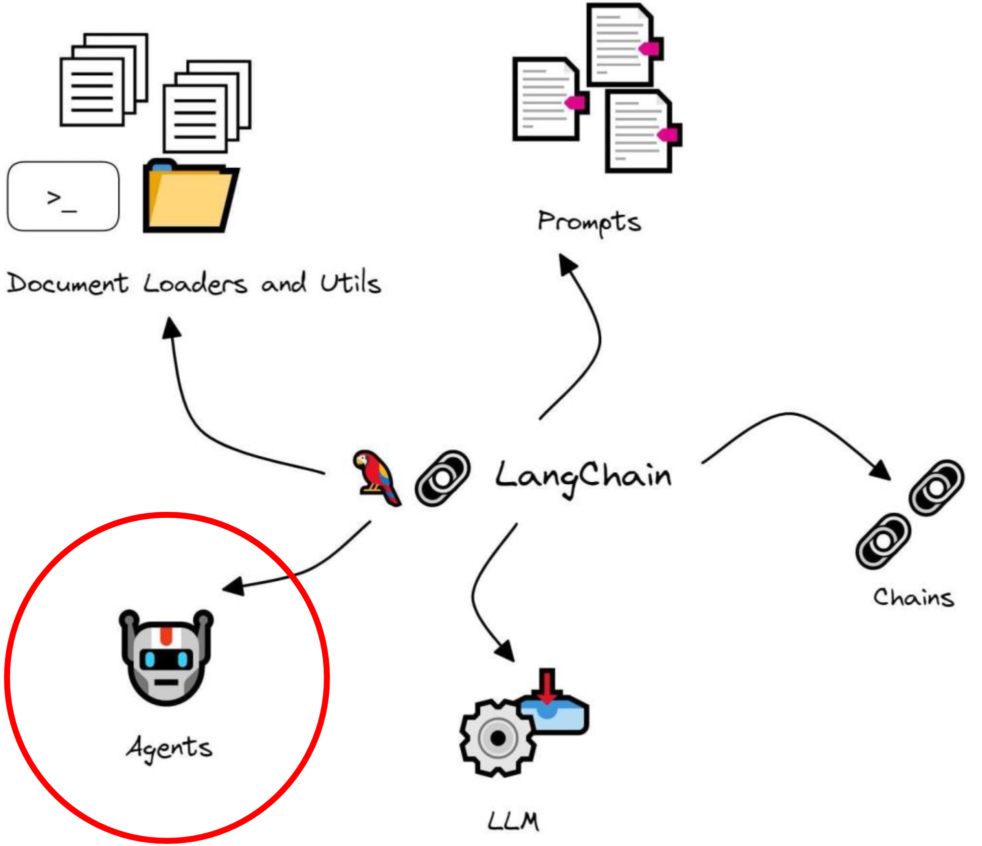
Document Loaders and Utils: Tools and utilities for loading and managing documents or data sources that can be used as inputs or references for the LLM.



LangChain Framework

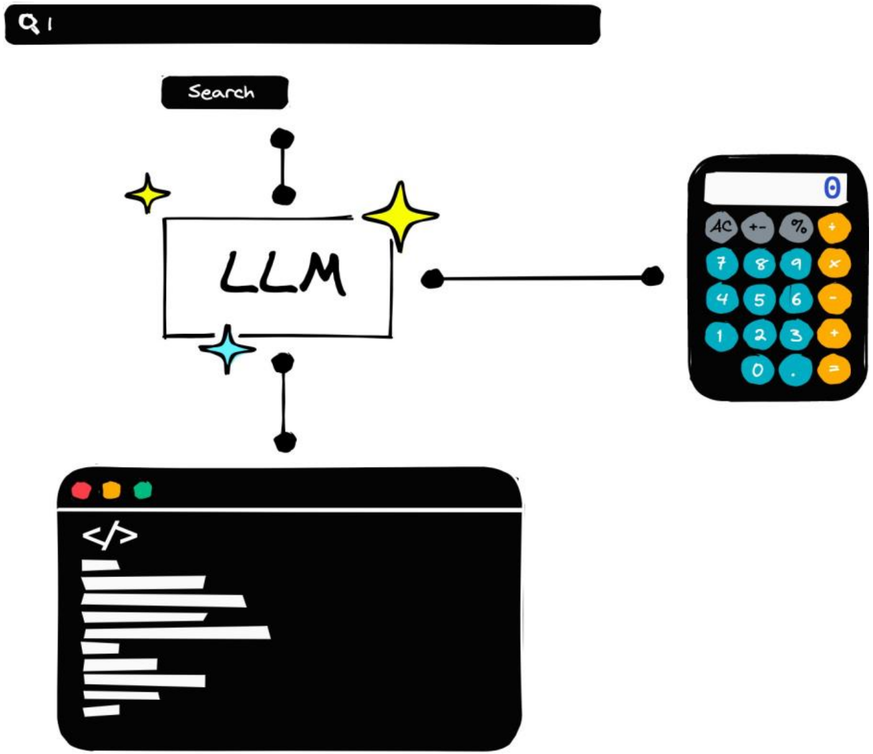
❖ LangChain Components

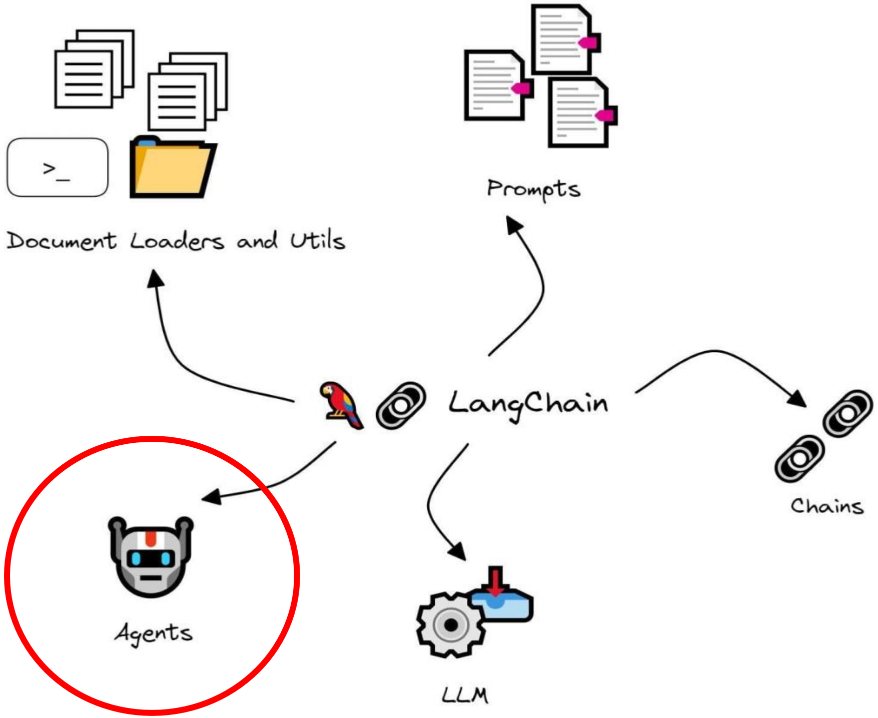
Agents: Software agents or programs that can interact with the LLM and other tools or services to perform specific tasks or actions.



LangChain Framework

❖ LangChain Components



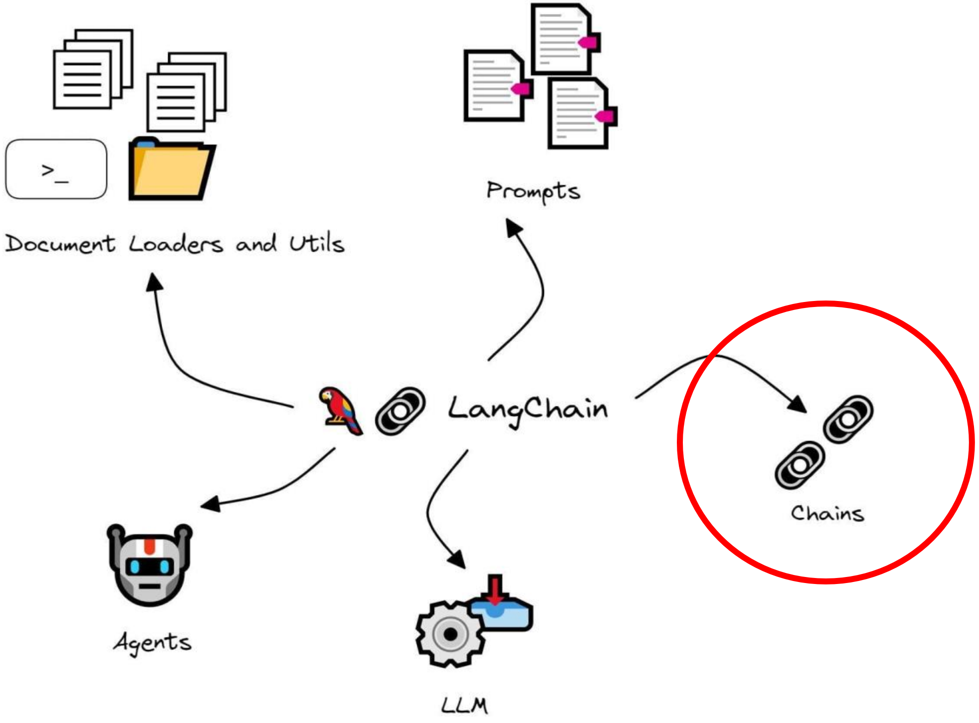


LangChain Framework

HITACHI
Inspire the Next

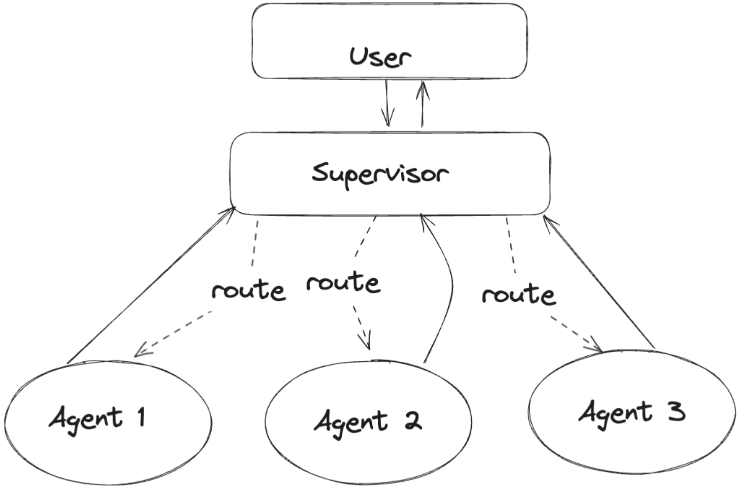
❖ LangChain Components

Chains: These are sequences or combinations of calls to the LLM and other components, allowing for the creation of more complex applications or workflows.



LangGraph

- **LangGraph** is an AI agent framework built on ***LangChain*** that allows developers to create more sophisticated and flexible agent workflows.
- **LangGraph** takes it a step further by offering a graph-based approach to orchestrate complex conversational flows and data pipelines. This makes LangGraph particularly suitable for projects that require managing multiple agents, conditional logic, and stateful interactions



From Single-Agent To Multi-Agent

Demo

Presenter
Anh Quoc Nguyen

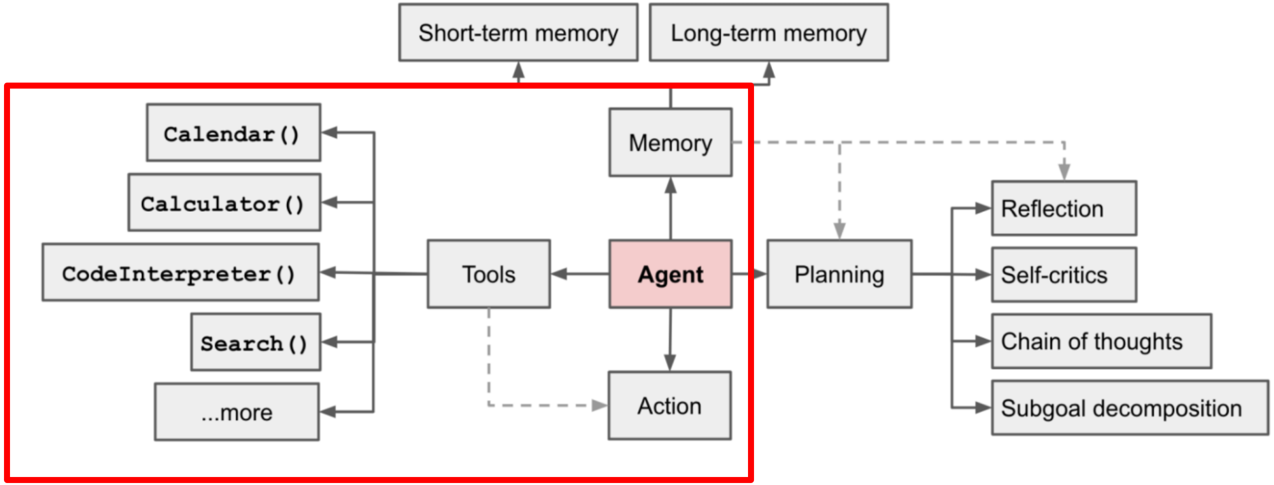
HITACHI
Inspire the Next

General Configuration

Common components for LLM-configuration are:

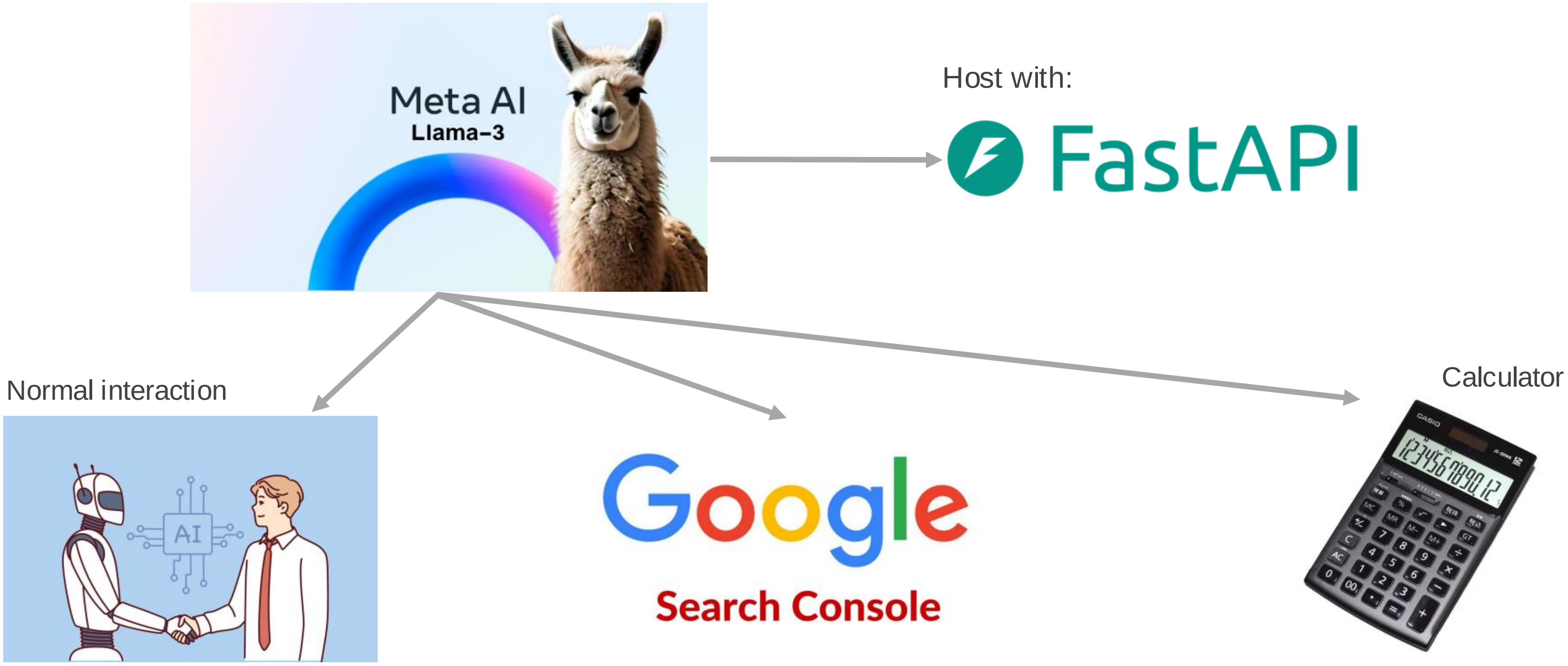
- **Tokens:** The basic units of text (words, subwords, or characters) that a language model processes to understand and generate language.
- **Prompts:** The input text or instructions provided to guide the model's responses or tasks.
- **Temperature:** A setting that controls the randomness of the model's output—higher values increase creativity, lower values make it more focused.

Single LLM-based Agent



- Be integrated with a pre-trained LLM (Llama3.3, DeepSeek, Gwen2.5, etc.)
- Equip pre-defined open-source tools (googlesearch, duckduckgo, etc.)

Demo: Models and Tools



Host with:

Normal interaction

Calculator

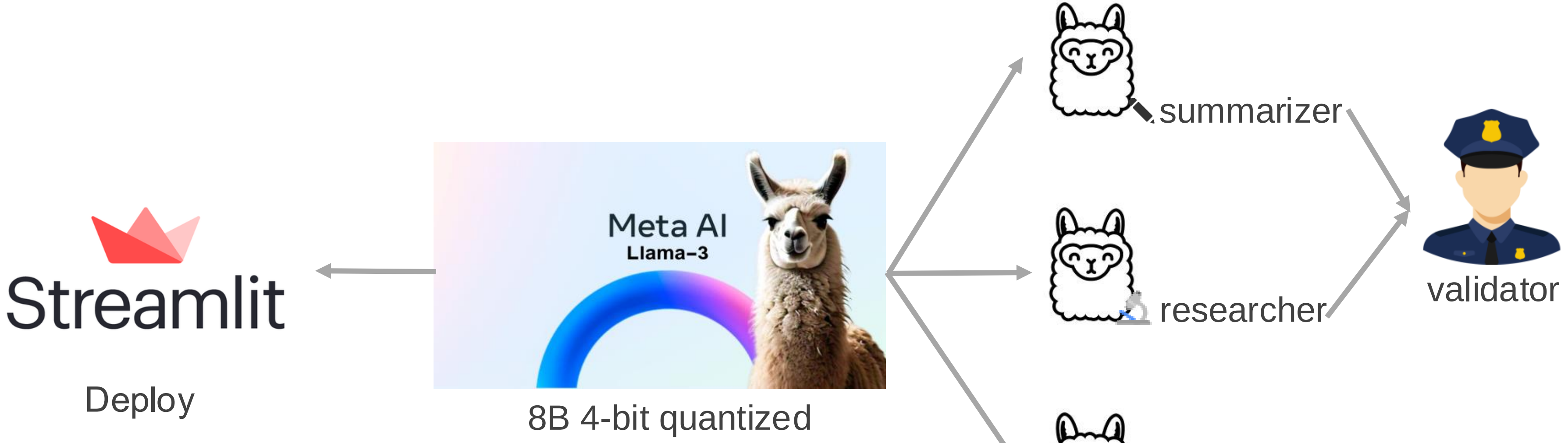
Llama-3: 8B parameter model with 4-bit quantization (and basic tools)

Limitations

HITACHI
Inspire the Next

- Lack the autonomy
- Being generalized across different fields
- Limited specialized capability
- Single point of failure
- No in-depth adaptability

Demo: LLM-based "Many" Agents



Deploy

8B 4-bit quantized

summarizer

researcher

lover

validator

Agents are now:

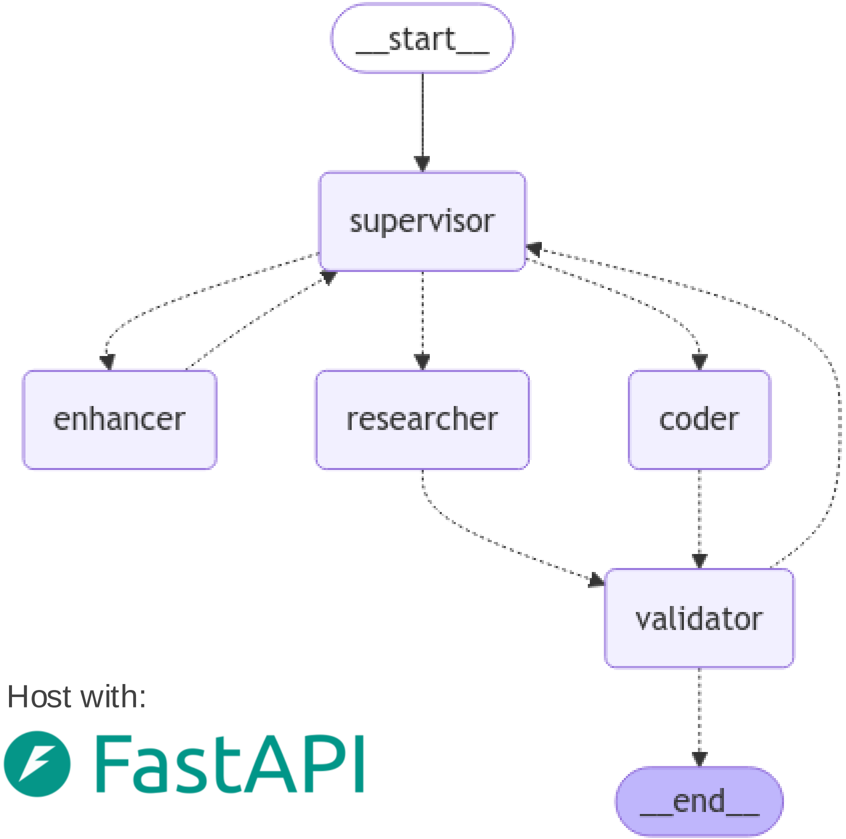
- Tailored with specialized instruction prompts, serving distinctive tasks
- Route to validator for quality/relevance check after each generation

Limitations

- Lack the autonomy
- Single point of failure
- Minor automation process

A LangGraph-based Multi-agent system

HITACHI
Inspire the Next



Host with:

FastAPI

Our architecture now offers:

- Seamless end-to-end operation and automation
- Greater controlling center with an active supervisor
- Adaptive route between agents in handling inputs
- Maintaining individual specialization
- Communicative entities

Challenges and The Future

As data grows

Presenter

Anh Quoc Nguyen

Key Takeaways

HITACHI
Inspire the Next

As clients' demand and agentic systems grow in parralel:

- Prompting could be lengthy, potentially surpassing the context-window of the system
- Hallucination remains an inherent problem as data increases in size
- Slow response time alongside stacking memory threaten developers

Nevertheless, there exists several potential:

- Text-to-SQL agent, Auto-Code agent, Co-operating debugger, etc.

Thank You

Hitachi Digital Services