

Harnessing the Agentic Force

How Open-Source AI Agents Power Chicory's Mission

Saurabh Sarkar

 Chicory AI

Welcome!

How open-source agent frameworks became core to Chicory's mission.



We stopped asking
AI for answers
and started giving
it jobs.

What is Agentic AI?

Not just bots – collaborators that reason, plan, and act...



reason



plan



act



Agentic Frameworks: The Squad

Frameworks for Dream Team

Framework	Primary Purpose	Strengths	Ideal Use Case	Current Use @ Chicory
LangChain	Prompt chaining and prototyping	Quick to start, lots of integrations	Prototyping LLM workflows	Used in early prototypes
LangGraph	Stateful agent orchestration	Fine-grained control, deterministic design	Complex decision-making pipelines	Primary orchestration layer
AutoGen	Multi-agent collaboration via chat	Supports agent-to-agent dialog easily	Agents verifying or discussing results	Explored during experimentation
crewAI	Role-based multi-agent workflow	Teams of agents with defined responsibilities	Task delegation in agent teams	Explored during experimentation
MetaGPT	Structured agent roles like startup teams	Encodes real-world org structure into agents	Building dev workflows with agent teams	Studied for inspiration
Temporal	Resilient async workflow orchestration	Reliable, production-grade task handling	Persistent or long-running workflows	Yet to be adopted

The Sidekicks: Observability Tools

Tool	Primary Purpose	Strengths	Limitations	Current Use @ Chicory
LangSmith	Debugging and tracing LangChain agent workflows	Detailed trace data; integrated with LangChain; easy prompt/tool inspection	No self-hosting; tied to LangChain ecosystem	Used for early experiments; good for replay analysis
Phoenix (Arize)	Visual monitoring and evaluation of LLM outputs	Open source; strong UI for agent trace visualization; metrics and drift tracking	No session stitching until run finishes; limited live debugging support	Primary tool in use; paired with LangGraph for observability
Weights & Biases (W&B)	Experiment tracking and logging for LLM workflows	Robust logging and tracking; versioning support; great for team experiments	More generic to ML; less focused on prompt or LLM-specific flows	Used for experiment logging and structured development cycles

Pick Your Jedi – LLMs We Use

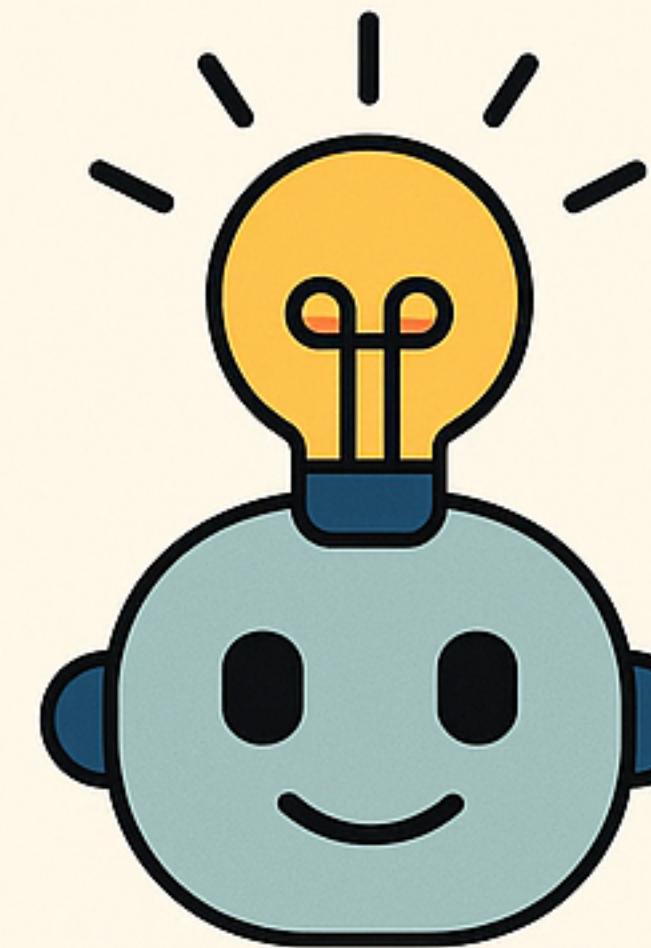
Model	Strengths	Ideal Use Case	Chicory Usage
OpenAI GPT-4.1	Deep reasoning, high accuracy, consistent output	Mission-critical logic, financial/health data workflows	General-purpose agents and pipeline validators
OpenAI GPT-4o	Fast responses, multi-modal input/output, low latency	Fast multi-modal agent interactions, UX tasks	Default model for fast iterative tasks
Claude 3.7	Massive context window (200K+), great at long-form analysis	Reading/understanding long documents or transcripts	Specialist model for documentation + policy-heavy tasks
Mistral/Mixtral	Custom fine-tuning, low cost, open deployment	Internal use where cost and control are priorities	Deployed with fine-tuning for internal agents
Command R+	Optimized for RAG, strong on retrieval-based tasks	Data catalog Q&A, lineage, vector retrieval	Primary RAG agent for internal knowledge graph
Gemini 1.5 Pro	Large context window, strong reasoning, multi-modal, API accessible	Document summarization, analytical tasks, RAG pipelines	Evaluating for multi-hop lineage queries and data doc parsing
Gemini 1.5 Flash	Optimized for speed and low latency, good for chat/UI integration	Conversational apps, fast agent loops, chat UI tasks	High-frequency agent interactions, live dashboards
OpenAI GPT-4.1-mini	Low-cost and lightweight GPT-4-level reasoning, fast response, good memory usage	Batch inference, microservice-based agents, rapid experimentation	Affordable default for dev/test and small-scale agent deployment

But Can It Be Trusted?

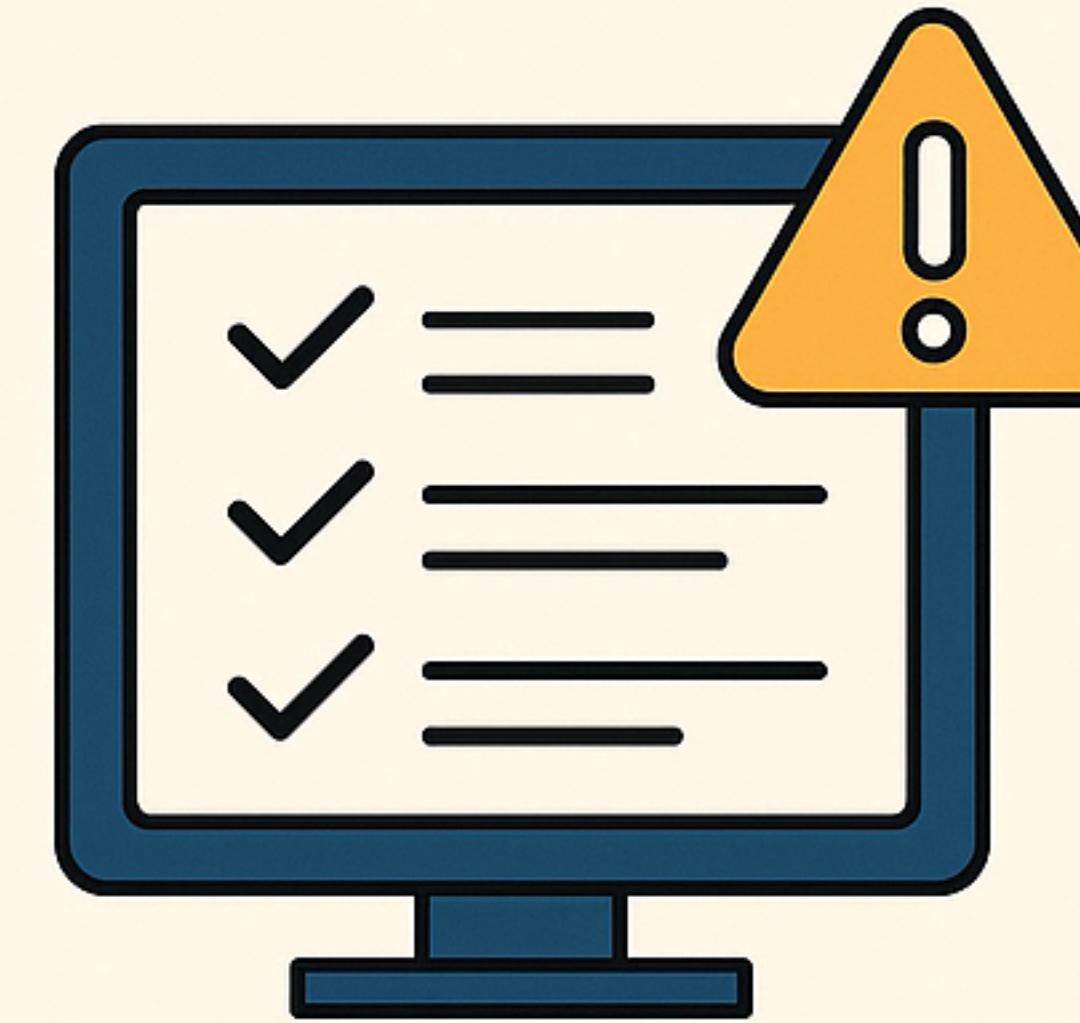
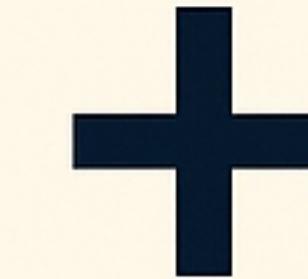
We made determinism a design principle

- Fixed seeds + top-1 decoding remove randomness.
- Scoped memory ensures focused, relevant decisions.
- Validation checkpoints mean nothing moves forward unless it's verified.

Neuro-Symbolic Agents



LLM



**Rule-based
validation**

LLM + Rule-based validation
Creativity with safeguards

Protocols for Agent Harmony

Protocol	Origin	Purpose	Strengths	Adoption at Chicory	Interoperability Focus
MCP (Model Context Protocol)	Anthropic	Structured tool invocation by agents using JSON-based format	Modular, clean tool use; follows JSON RPC pattern; integrates easily with tools	Adopted JSON-style invocation pattern in internal tooling	Tool-oriented, not cross-platform
A2A (Agent-to-Agent)	Google	Agent-to-agent collaboration through structured task-oriented messages	Designed for cooperative multi-agent workflows with clear status updates	Conceptually explored for coordinating multiple agents	Intra-system agent coordination
ACP (Agent Communication Protocol)	Linux Foundation / BeeAI	Cross-platform agent interoperability; agent discovery, messaging, and role negotiation	Aims to be a universal standard like HTTP for agent communication; enables plug-and-play agents	Not yet adopted; actively monitoring for future integration	Cross-platform agent networks (long-term goal)

Use Cases Solved at Chicory

- Warehouse Lineage Mapping
- Pipeline Management - Building & Debugging
- Auto-Docs
- Feature Engineering

It's Not All Serious...

- LangChain is like LEGO
- AutoGen: AI interns arguing in public
- Temporal: The operations wizard

Take Aways

"trying everything" to "scaling responsibly"

- Start with LangChain, grow into LangGraph
- Use observability tools to stay sane
- Chicory's secret: determinism + agentic design + humor
- Open source made it all possible

Quote from the Frontier

“AI is not going to replace people, but people who use AI will replace people who don’t.”

– Satya Nadella

Q&A