# Open Agent Specification (Agent Spec) — Overview

A Unified Declarative Standard for AI Agents

- From fragmented agent frameworks to interoperable agentic systems

- Source: **arXiv 2510.04173** (October 2025)

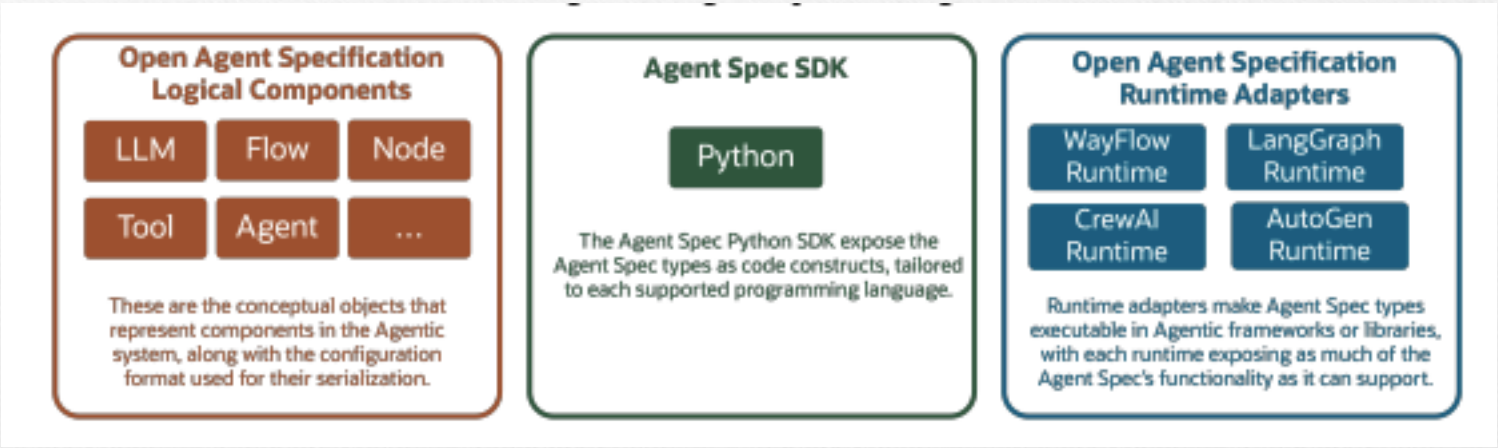
# Design Objectives

1. Portability and interoperability across frameworks
2. Declarative, modular, composable definition of agents
3. Explicit control and data flow
4. Validation and conformance testing
5. Support for multi-agent composition

# Core Concepts/Components

- **Agent** – conversational or reasoning entity
- **Flow** – structured workflow (nodes, branching, loops)
- **Tool** – API, function, or external service connector
- **Memory / Prompt Templates** – contextual state management
- **Edges** – define control and data flow relationships

# Components


Open Agent Specification
Logical Components
LLM
Flow
Node
Tool
Agent
...
These are the conceptual objects that represent components in the Agentic system, along with the configuration format used for their serialization.
Agent Spec SDK
Python
The Agent Spec Python SDK expose the Agent Spec types as code constructs, tailored to each supported programming language.
Open Agent Specification
Runtime Adapters
WayFlow Runtime
LangGraph Runtime
CrewAI Runtime
AutoGen Runtime
Runtime adapters make Agent Spec types executable in Agentic frameworks or libraries, with each runtime exposing as much of the Agent Spec's functionality as it can support.

# Serialization, SDKs & Runtime Adapters

- YAML/JSON schema for agent definitions
- Python SDK (PyAgentSpec) for building and validating Agents' specs
- Runtime adapters map spec to specific runtimes (OCI, LangGraph, AutoGen)
- Import/export between frameworks (Agent ↔ LangGraph ↔ AutoGen)

# Control Flow & Data Flow Semantics

- Directed edges define execution order, branching, loops
- Inputs/outputs mapped explicitly across nodes
- Nested and reusable flows or sub-agents
- Modular referencing for reuse and composability

# Benefits & Value Proposition

- **Developers**: portability, validation, reuse
- **Frameworks**: standardized interchange format
- **Researchers**: reproducibility and comparability
- **Enterprises**: governance, modularity, and no/reduced lock-in

# Limitations & Challenges

- Early-stage adoption and maturity
  - Proposed: October 2025
- Runtime mismatch across frameworks
- Performance overhead due to abstraction layer
- Safety and observability delegated to runtimes

# Roadmap & Future Directions

- Add Memory, Planning, Datastores, A2A protocols
- SDKs in more languages
- Expand adapter ecosystem
- Conformance tests and Visual Editors
- Community growth and registry of agents

# Critique & Strategic Considerations

- **Strengths**: framework-agnostic, modular, reusable
- **Risks**: adoption barriers, runtime complexity
- **Suggestions**: start modular, contribute adapters, focus on observability

# Summary & Key References

- Open Agent Specification is a declarative language for defining interoperable AI agents.

- Key links:
  - Paper: https://arxiv.org/abs/2510.04173
  - GitHub: https://github.com/oracle/agent-spec
  - Docs: https://oracle.github.io/agent-spec/index.html
  - Blog: https://blogs.oracle.com/ai-and-datascience/post/introducing-open-agent-specification