

THE MOSAICO CHRONICLES



Antonio Garcia-Dominguez 

Senior Lecturer in the Department of
Computer Science at the University of York



Dimitris Kolovos 

Professor of Software Engineering
at University of York

Collaboration between diverse agents with the MOSAICO AISP profile for A2A

It is now recognised that a mix of specialised agents can perform much better than a single all-purpose agent for software engineering tasks. This idea is now becoming more and more embedded in agentic coding tools and IDEs, such as specialised sub-agents in Claude Code and Google Antigravity. These subagents are often based on a high-level description and a selection of tools that they can use, and they generally live in the end-user's machine.

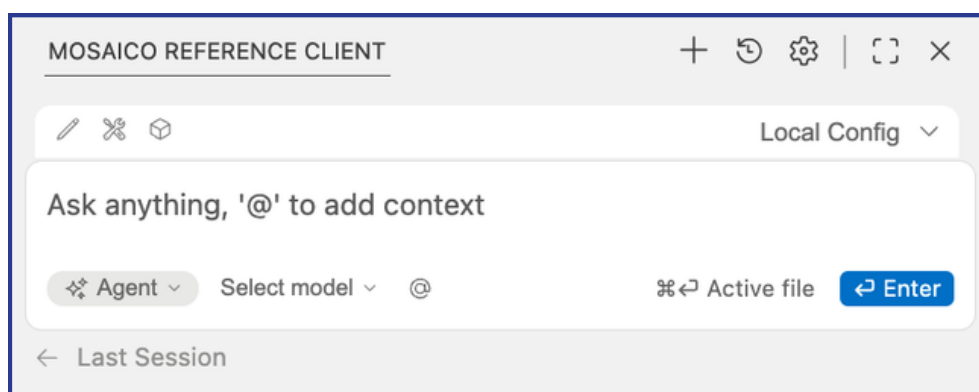
In MOSAICO, we have a broader concept for what an agent can be. We want to enable both local and remote agents, and we want the agent developer to have free choice on the programming language, libraries, internal structure, and LLM that the agent we will use (if any: we want to facilitate having deterministic agents in the mix, which don't use an LLM at all). This means, rather than mandating agents to all follow the same structure, what we want to standardise is how they talk to each other and to the user: in MOSAICO, we call this the AI Agent Server Protocol (the name is inspired by specifications such as the Language Server Protocol, which aimed to standardise language integration in IDEs).

When we started MOSAICO in January 2025, we commenced work on AISP by developing our own agent intercommunication protocol from scratch, using HTTP, JSON, and Server-Sent Events to allow for streaming updates. However, the release of Anthropic's Model Context Protocol (MCP) and Google's Agent-to-Agent Protocol (A2A, now hosted by the Linux Foundation) called for a re-evaluation of the approach. After comparing our new protocol, MCP, and A2A, we found that A2A already met many (although not all) of our requirements, had already significant buy-in from the community, and had built-in mechanisms for extension. We decided to pivot AISP into a profile of A2A, which would evolve together with A2A and with the needs of our industrial partners in MOSAICO.

AISP brings to the table unified definitions of what the various kinds of agents in MOSAICO expect to receive and send, and extensions to cover the MOSAICO requirements that A2A is not covering yet. In terms of kinds of agents, we have:

- The reference agent, which receives prompts from the client (e.g. a chat UI), finds the relevant agent among those registered in the organisation's repository (using a combination of vector search and an LLM), and delegates the task on that agent. It is also responsible for reporting back to the client any requests for information from the agent.
- Solution agents, which propose a solution given a prompt (which may include feedback and outputs from previous interactions).
- Supervision agents, which evaluate a solution according to a particular dimension. These agents may use an LLM, or may not (e.g. we may have an agent performing static analysis on code).
- Collaboration agents, which orchestrate other agents into a workflow. The workflow may be static (written in code), or may be generated on the fly (e.g. via an LLM). Collaboration agents typically look from the outside as regular solution or supervision agents, depending on their goal.
- Consensus agents, which take in multiple solutions and evaluations, and choose which solution to pick.

We already have initial versions of the reference agent, and a mix of solution, supervision, and collaboration agents in our [open-source demonstrator](#). Unlike the homogeneous agents that are typically seen in Claude or Antigravity, these agents are written with a mix of languages (Python, JavaScript, and Java) and libraries (LangGraph, Smolagents, or Spring AI). The agents don't need to run locally, and could be hosted anywhere. We also support multiple clients, such as our current [reference client](#) (a VS Code extension), or a native [Theia AI IDE client](#).



So far in MOSAICO, we have identified three things that A2A doesn't directly tackle that will require extensions, and we are working on those extensions:

- We want to have observability traces across an entire collaboration between agents, rather than having individual LLM call traces.
- We have implemented an extension that allows for organising A2A tasks into a hierarchy, which can then be used to produce OpenTelemetry traces (e.g. for Langfuse).
- Rather than treating agents strictly as black boxes, we want agents to be aware of what the IDE can do for them, and let them ask for information if they need it (agent-directed information retrieval).
- We have developed a first version of an extension where a client can report their tools to an agent, and agents can send input-required A2A task status updates requesting such tool calls. This can then interoperate with existing UIs for requesting permission to perform the tool call. We also allow for agents to ignore such tool definitions (e.g. for security reasons or for simpler agents that are meant to behave more deterministically) or to select a subset of such tools. We plan to further refine this extension to cover refusals from the user, and experiment with exposing not just the predefined IDE tools but also tools brought from MCP servers.
- We plan to flesh out how a user or an agent can provide feedback on a given output from an agent, in a more fine-grained manner than just a prompt (e.g. by pointing to specific subsets of the output).

We are in the process of extracting our AISP implementation across clients and agents into a streamlined specification of the technical requirements that MOSAICO agents must follow. We have recently updated AISP and the demonstrator to A2A 1.0 (the latest release of the spec), and we will be updating the technical requirements document soon to A2A 1.0 as well. We will also continue defining AISP-specific extensions for A2A as MOSAICO develops, including the support for fine-grained feedback on specific parts of the results from the agents.

Interested in future updates? Don't forget to **follow us!**



FOLLOW US!