Open Source Community Day

23 - 24 September

ECLIPSE FOUNDATION

CEI-Sphere part of EUCloudEdgeIoT.eu

Co-located with AIOTI Days 2025

MOSAICO: AI-agent communities for reliable AI in software engineering

Antonio Garcia-Dominguez

IMT Atlantique, LIST, U. of York, U. of L'Aquila

Collins Aerospace, Immersion, NBG, Unparallel

Netcompany, Qodo, Eclipse Foundation, F6S

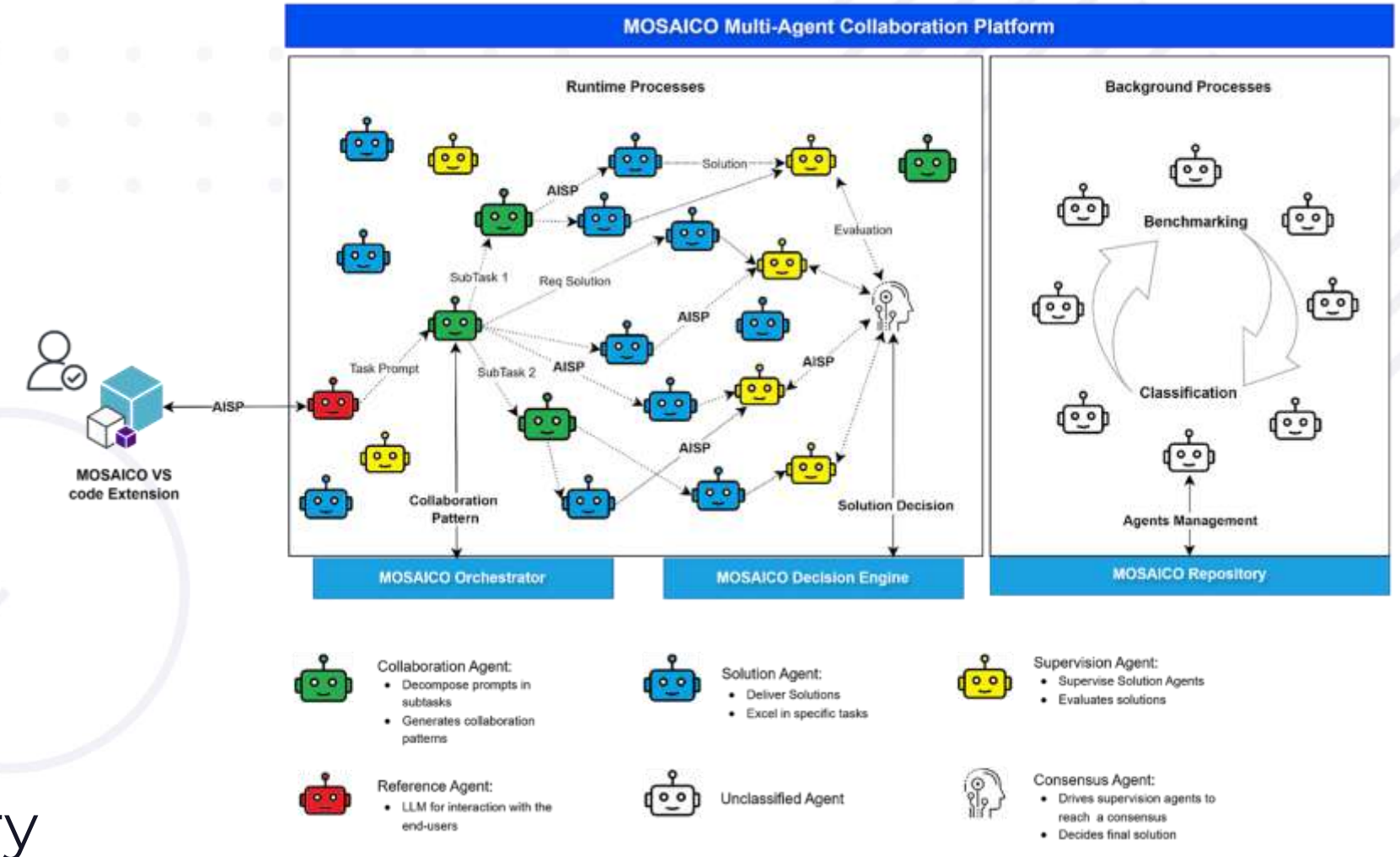
September 24th 2025

Why MOSAICO?

- AI has the potential to change how we develop software, at every stage of the lifecycle
- However, we need the appropriate infrastructure to select the most appropriate agents and use them well
- Research has shown that:
  - LLM-based agents perform best when they can observe the environment and plan their next action (e.g., ReAct)
  - Multiple specialized agents can solve complex tasks better than a single general-purpose agent
  - Current LLMs can struggle with task switches
- MOSAICO aims to develop this infrastructure

What is the MOSAICO vision?

- MOSAICO: EU research project to deliver modular platform for AI-powered SE
- Four agent types:
  - Collaboration
  - Solution
  - Supervision
  - Consensus
- Indexed in the MOSAICO Repository



Objectives of the MOSAICO project

- Design and implement an **AI-agent server protocol** to communicate agents (York)
- Deliver a **repository of AI agents** dedicated to various SE tasks which is aware of benchmarks and live performance (L'Aquila)
- Define a **framework for coordinating collaboration** in AI agent communities (IMT)
- Define a **governance language and decision engine** to enact governance policies (LIST)
- Integrate MOSAICO in 4 different **industrial use cases** (Collins Aerospace, Immersion, NBG, Unparallel)
- **Long-term adoption with OSS and community building**

WP1: AI-agent server protocol (York)

- We need agents to communicate with each other and with the developer, in a way that enables collaboration
- This implies a number of requirements, such as:
  - **Discovering** what SE tasks can be done by an agent
  - Starting and continuing **conversations**, where messages can be structured and contain a mix of content types
  - Being able to **observe** the steps in a collaborative SE task
  - Supporting **heterogeneous agents** made in different technologies, with different levels of intelligence
- All this while keeping access control and bandwidth use in mind: we will need reference client + server, too

WP2: Repository of AI agents for SE (L'Aquila)

- The community is more than a list of agents: we need to know which agents are good at a given task
- This requires:
  - Having a **taxonomy** to classify AI agents for SE
  - Defining the **KPIs** for agent performance and efficiency
  - Creating a **benchmarking framework** for AI agents
  - Creating the **repository** itself (as a network service)
- The repository will be used by the WP1 reference agent to find the agent that best solves the task
  - Often, this will be a collaboration agent, which may involve all four types of MOSAICO agents (including other collab. agents)

WP3: Coordination and collaboration of communities of AI agents (IMT)

- Belief-Desire-Intention is a well-known software model for programming agents (pre-LLM era)
  - Belief: what the agent believes about the world
  - Desire: what the agent wants to achieve
  - Intention: what the agent has chosen to do
- WP3 involves:
  - Adapting the **BDI framework** to LLM-based agents
  - Defining a **language** to describe collaboration patterns
  - Using the patterns to **produce collaboration agents**
  - Allowing the patterns to be **dynamic** (change over time)

WP4: AI community governance and supervision (LIST)

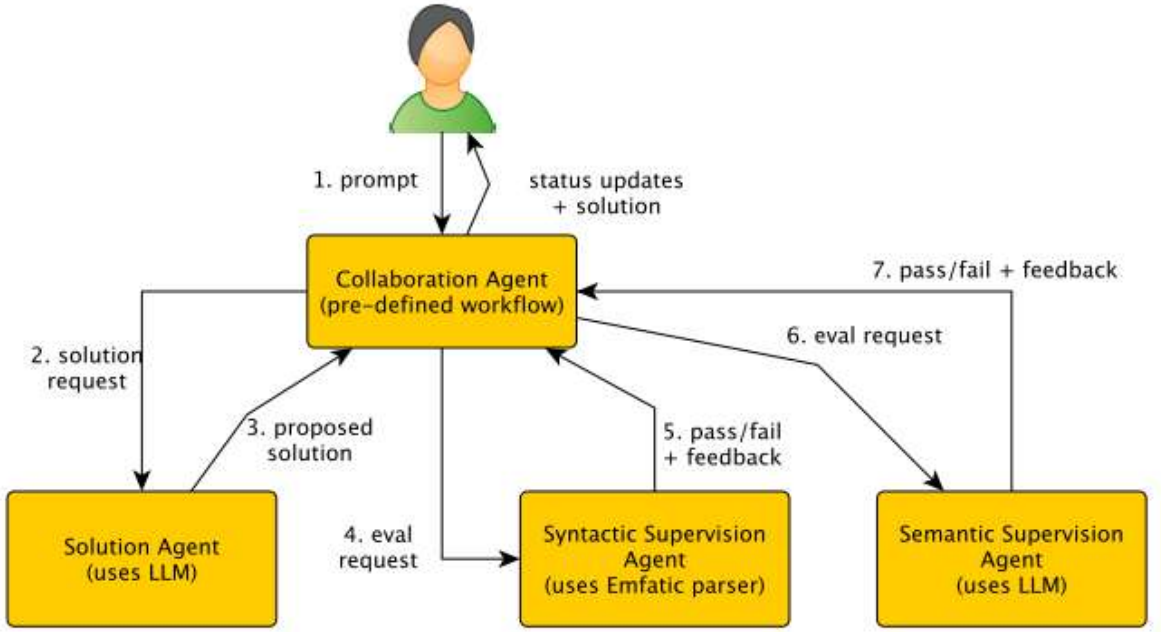
- Sometimes we will have more than one way to solve a problem: we need a governance mechanism to decide which way to go (which may also involve human input)
- This package involves:
  - Creating a **policy language** to define governance rules
  - Using the language to **model common** governance **scenarios**
  - Creating a rule-based decision engine for the language
  - Adding **argumentation** to the engine

Returning to WP1: OSS areas of interest

- As WP1 lead, I will talk more on our experience so far
- Agentic AI is a hot topic currently, with significant effort being put across academia and industry
- To stay up-to-date, we aim to collaborate on related projects, rather than reinventing the wheel:
- We examined OSS options in these areas:
  - Communication protocols and SDKs
  - Agentic frameworks
  - Local LLMs
  - Observability platforms
  - Conversational UIs for IDEs

OSS protocols and SDKs

- **Nov 2024**: Anthropic publishes their Model Context Protocol open standard
- **Feb 2025**: we start our own in-house agent protocol (OpenAPI-based, using SSE)
- **Apr 2025**: Google announces their Agent-to-Agent protocol (now under the Linux Foundation)
- Tried all three on example: A2A ticks many boxes but not all
  - Plan to combine A2A (with our OSS extensions) + MCP



Agentic frameworks

- Our WP1 example experimented with two frameworks:
  - LangGraph: open core, but proprietary server
  - Smolagents: Apache 2.0-licensed, no server
  - We see potential for mixing and matching frameworks (e.g., ReAct vs fixed flow, JSON vs Pythonic tool calling)
  - Protocols allow agents to communicate regardless of the underlying framework
- Within WP2, L'Aquila has reviewed a broader range:
  - AutoGPT, AutoGen, Dify, Flowise, Haystack, LlamaIndex…
  - Results to be published as a MOSAICO deliverable soon (D2.1)

Local LLMs

- Our industrial partners have different sensitivities on sending data to remote LLMs
- Sometimes, local LLMs must be used due to security, intellectual property, privacy, or latency concerns
- Within WP1, we have experimented so far with:
  - Proprietary LLMs (e.g., OpenAI gpt-4o-mini)
  - Open-weights LLMs (e.g., Qwen2.5-coder or OpenAI gpt-oss)
  - Open-source LLMs (e.g., Qwen3)
- We expect WP2 to provide benchmarking tools to better judge the various tradeoffs in LLM choice

Observability platforms

- Agent collaborations and LLM uses need to be tracked for transparency and accountability, and to help agent developers debug their software
- As a form of distributed software, we want distributed tracing: luckily there is OpenTelemetry for that
- Several types of AI-aware OT-based solutions exist:
  - SaaS-only, like Logfire (OSS SDK, closed backend)
  - Source-available, like Phoenix (Elastic License 2.0)
  - Open-core, like LangFuse (BSD/MIT-style core, closed extras)
- Any 100% OSS solution out there? 😊

Conversational UIs for IDEs

- WP1 will need to produce a reference client:
  - Conversational UI that talks to a reference agent
  - Reference agent asks repository for relevant agent for a task (which may run locally or remotely), and communicates with it
- So far, we have found two possible starting points:
  - Copilot Chat: MIT license (does not include Copilot code suggestions, and would need to release outside Marketplace)
  - Continue: Apache 2.0 license
- We are also considering a new project, depending on the results of our initial search (e.g., support for A2A)

Conclusion

- MOSAICO is an EU project to deliver a platform where AI agents collaborate on software engineering tasks
- MOSAICO involves 12 partners across 9 countries, covering protocols, repositories, collaboration, and consensus approaches
- We have gained some experience with OSS offerings across the platform: opportunities for reuse and contribution, but we see a broad mix of licenses
- We will open-source results as we go along: check our website and LinkedIn on the next slide!