ENI ISG PoC #24 Report

1 General

The following normative disclaimer shall be included on the front page of a PoC report:

> *Submission of this ENI ISG PoC Report as a contribution to the ENI ISG does not imply any endorsement by the ENI ISG of the contents of this report, or of any aspect of the PoC activity to which it refers.*

2 ENI ISG PoC Report

2.1 PoC Project Completion Status

Indicate the PoC Project Status. Can the PoC be considered completed? If this is a multi-stage PoC project, indicate the Reported Stage status and plans for future Project Stages/Milestones:

- Overall PoC Project Completion Status: Closed
- PoC Stage Completion Status (Optional - for Multi Stage projects only): Not applicable

2.2 ENI PoC Project Participants

Specify PoC Team; indicate any changes from the ENI ISG PoC Proposal:

- PoC Project Name: AI Core-Next Generation Network Architecture
- Network Operator/Service Provider:
  - Deutsche Telekom AG, Contact: Stephan Blicker (stephan.blicker@telekom.de)
  - China Telecommunications, Contact: (zengyu@chinatelecom.cn)
- Manufacturer:
  - Huawei Tech. GmbH, Contact: Onur Ayan (onur.ayan@huawei.com)
- Additional Members:
  - Futurewei, Contact: John Strassner (john.sc.strassner@futurewei.com)
  - Universitat Politècnica de València, Contact: David Lopez (d.lopez@iteam.upv.es)

2.3 Confirmation of PoC Event Occurrence

Figure 2-1: Demonstration scene for PoC #24

PoC #24 has been presented during the ETSI ENI Plenary #37 in Munich, Germany. Participants of the ETSI ENI plenary were on-site. It demonstrates AI-Core, a next-generation core network architecture that supports multi-agent collaboration and provides customized generative services to users. This PoC implements and presents a complete AI-Core framework comprising multiple LLM-empowered network AI agents, and a graphical user interface (GUI) for real-time human interaction. It uses two use cases, namely *network-assisted robot* and *AI-Phone*, to showcase the end-to-end processing pipeline and the characteristics of typical scenarios in AI-Core.

Figure 2-2 shows a screenshot of the graphical user interface used in PoC#24.

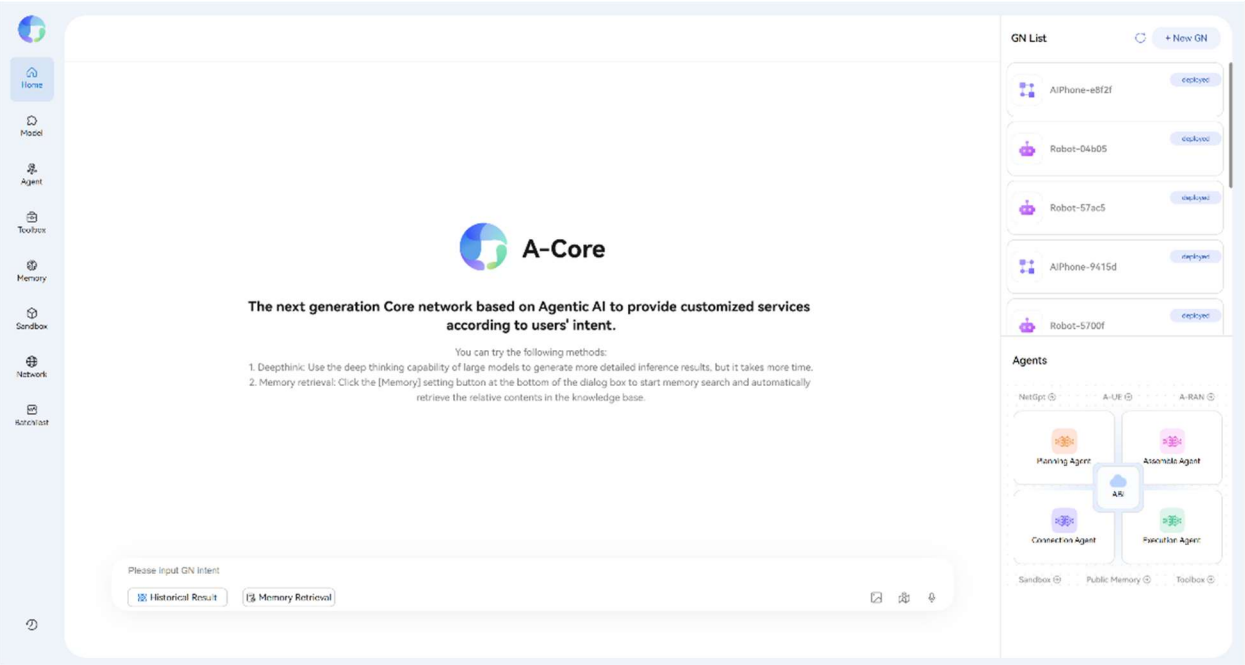


Figure 2-2: GUI of PoC #24

PoC#24 has also been presented at the ETSI AI and Data Conference, which was held in February 2026 at ETSI in Sophia-Antipolis, France.

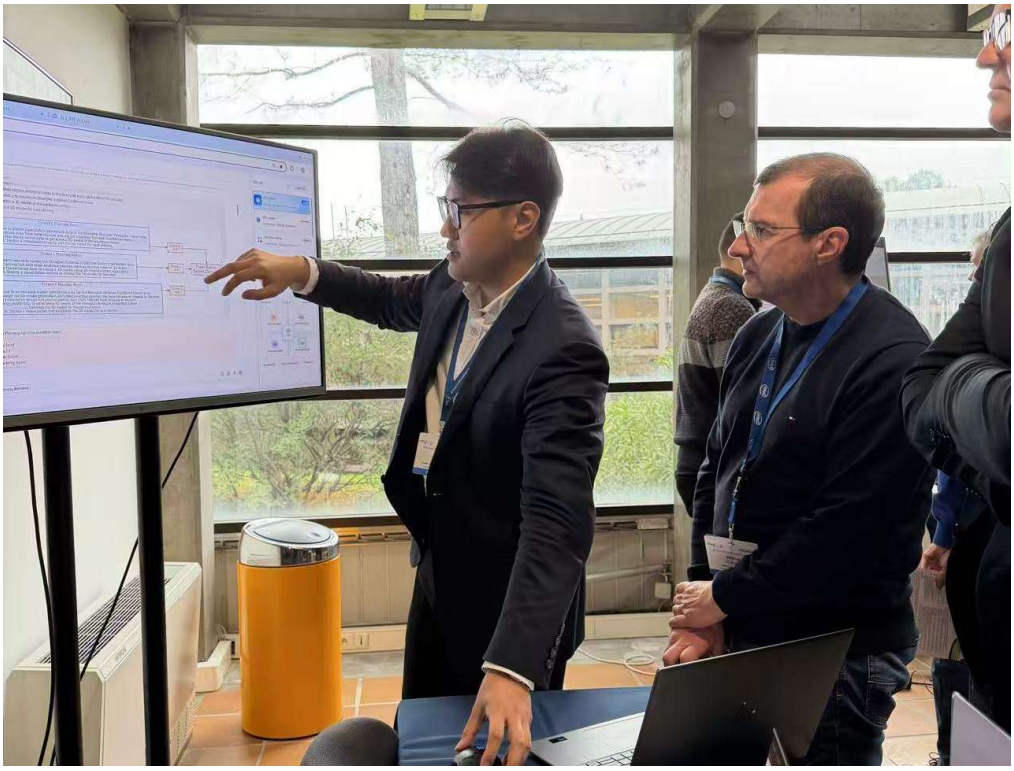


Figure 2-3: Live demonstration at the ETSI AI and Data Conference, Sophia Antipolis, February 2026

2.4 PoC Goals Status Report

PoC Project Goal #1: Feasibility verification of the AI-Core architecture. This verification ensures that the AI-Core system can provide users with an intent-based service entry point, enabling a series of tasks through collaboration among network agents, including understanding user intent, task decomposition, tool matching and assembly, workflow generation, connection configuration, sandbox verification, and tool deployment and execution. Additionally, it verifies that agents can interact through A2A/MCP-like interface and can also access to databases/toolkits. Furthermore, this PoC also verifies the agents' capabilities in tool invocation and memory retrieval.

PoC Project Goal #2: Verification of use cases. The PoC verifies that the AI-Core system can provide a unified intent entry for different types of users and generate minute-level customized services on demand using different guidelines, thereby meeting the differentiated requirements of individual consumers and robot terminals.

PoC Project Goal #3: New Business Opportunities for Operator. Demonstrate the new business opportunities brought by AI-Core to operator. AI-Core allows operators to become an innovative service platform and provide interfaces for third parties to access, so that third-party APIs and skills can be integrated into the unified platform, thereby providing flexible services for user and creating new revenue streams through generative services.

2.5 PoC Feedback Received from Third Parties (Optional)

This AI-Core PoC has been demonstrated to multiple operator customers with positive feedback. The most noteworthy comments are summarized as follows.

1. The AI-Core-based application service innovation platform presents an excellent opportunity to deeply integrate the sensing, connectivity, and computing capabilities of operator networks with third-party application operations. This integration provides a unified intent entry for end-users (C-end) and business users (B-end), especially when combined with the integrated sensing and communication (ISAC) capabilities on the RAN side.

2. For end-users (C-end), the customized capabilities of the network can also be leveraged to meet their strong communication-related demands, such as high-definition video calls and high-quality video conferences with customized in-network processing and storage features.

3. In the core network, ensuring the reliability of agents is a critical concern. It is essential to provide stability guarantees for multi-agent collaboration, even when these are provided by different vendors. Moreover, it is important to ensure the reliability of the core network control plane, and ensure quick recovery from potential issues affecting the generated services.

3 ENI PoC Technical Report (Optional)

PoC #24 demonstrates the working process of AI-Core system prototype and uses two typical use cases to illustrate how to customize a generative service based on user intent, including collaboration among multiple agents and tool invoking. The figure below shows a typical AI-Core architecture, which includes multiple network agents, toolbox, sandbox and other components. The toolchain generated based on the intent will be deployed in the generative network to provide services for users.

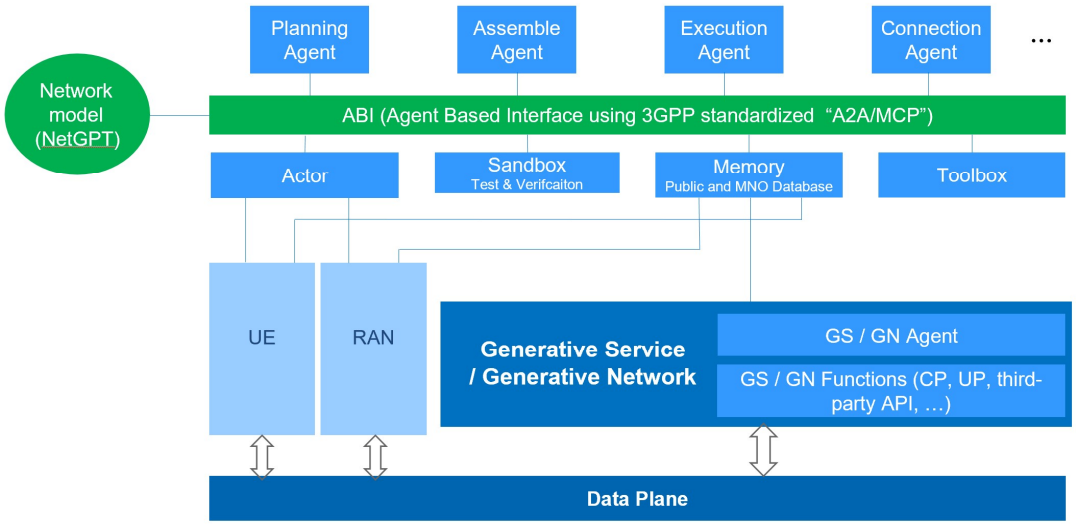


Figure 3-1: PoC architecture of AI-Core

PoC #24 implements two use cases (UCs): AI-Phone and network-assisted robot.

1. Use Case 1: AI-Phone

The motivation of this UC is to transform future mobile phone service from APP-driven service to model-driven service.

The procedure is that AI-Core creates a customized "generative network" to call the *personal_assistant_agent* to fulfill the user intent, including these steps:

1. The AI-Phone establishes network connectivity and the client sends an intent to Connection Agent.
2. The AI-Core agents (planning, assemble, connection, execution agent) coordinate to prepare GN components (Personal_Assistant_Agent) and send the authorized API scope to Personal Assistant Agent.
3. The GN processes the user intent through code generation and verification by LLM.
4. The Personal Assistant Agent invokes external APIs to accomplish tasks based on the generated code.
5. GN sends the task results to user client.

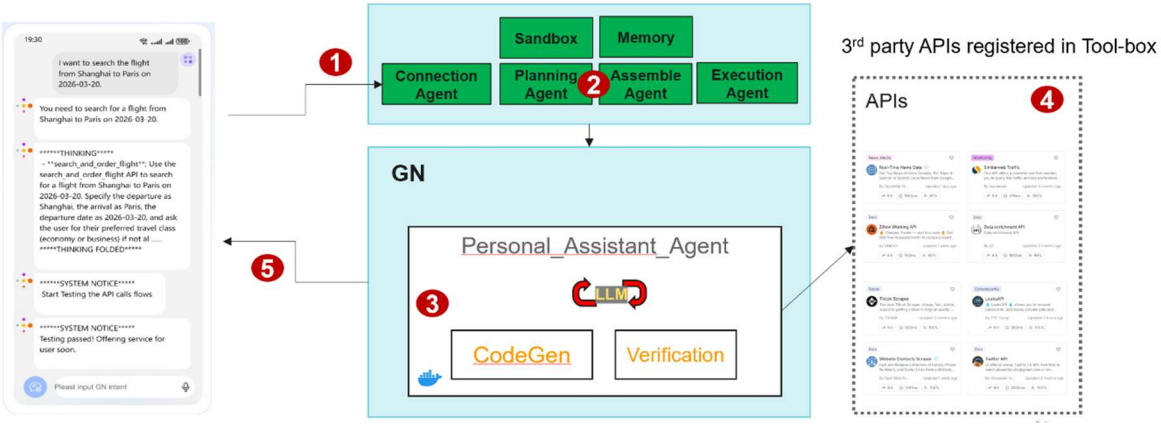


Figure 3-2: AI Phone Procedure

Here, we use the intent "*I want to search the flight from Paris to Shanghai on 2026-03-20*." to demonstrate the whole procedure.

After receiving the user's intent, the planning agent generates a GN ID to identify this task. It starts by analysing the category to which the user's intent belongs. Then, it launches multiple threads to process in parallel, analysing the user's requirements, examining the corresponding guidelines and breaking down the request into a series of subtasks.

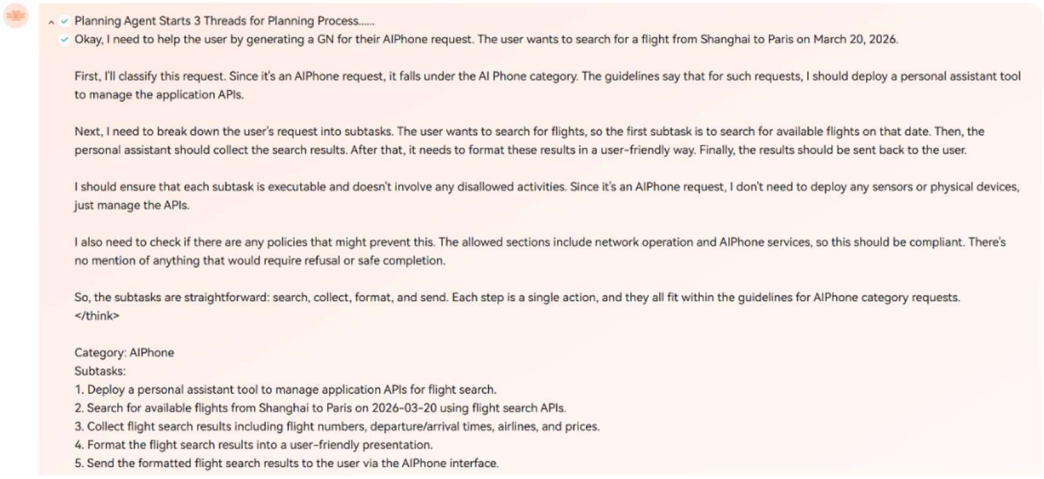


Figure 3-3: Subtask planning process based on user intent

Afterwards, the planning agent scores the results using the large language model (LLM) based on evaluation criteria, selecting the thread with the highest score as the final result.

The assessment criteria include: (1) Accuracy; (2) Completeness; (3) Logicality; (4) Efficiency; (5) Compliance

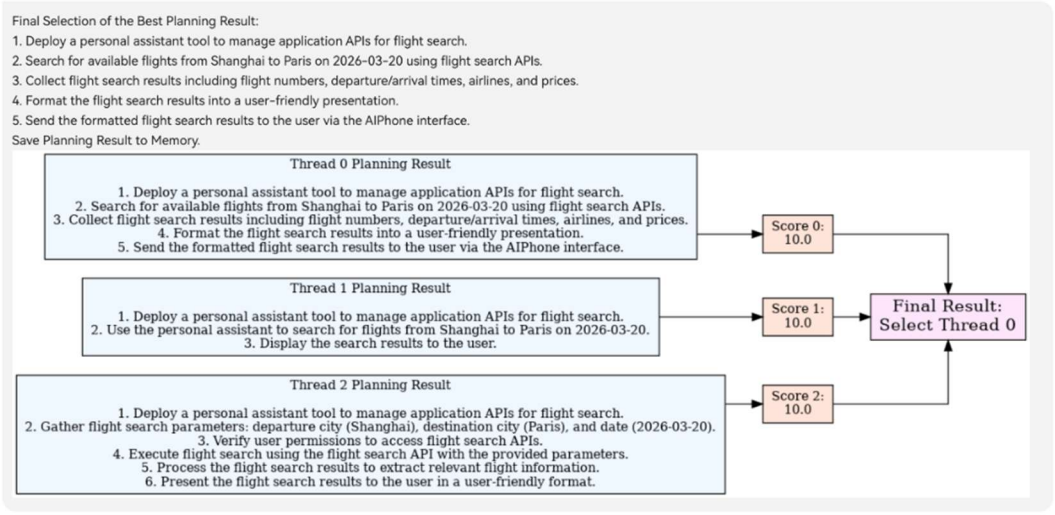


Figure 3-4: Comparison of different planning results

After the planning agent identifies the optimal planning result, it stores the result in memory for future use via the Memory Retrieval operation. The planning agent then sends this result to the assemble agent through the Agent-Based Interface (ABI), along with the specified intent.

Figure 3-5: ABI message from planning agent to assemble agent

Upon receiving the message, the assemble agent initiates a request to the Toolbox to retrieve available tools. To determine the best match for each subtask, a prompt has been implemented that guides the assemble agent to analyse the semantic similarity between the subtasks and the tool descriptions in the toolbox.

The assemble agent then evaluates all selected tools based on evaluation criteria, ultimately selecting the tools with the highest score as the final choice for task execution.

The Assemble Agent selects the most appropriate tool for each task as follows:

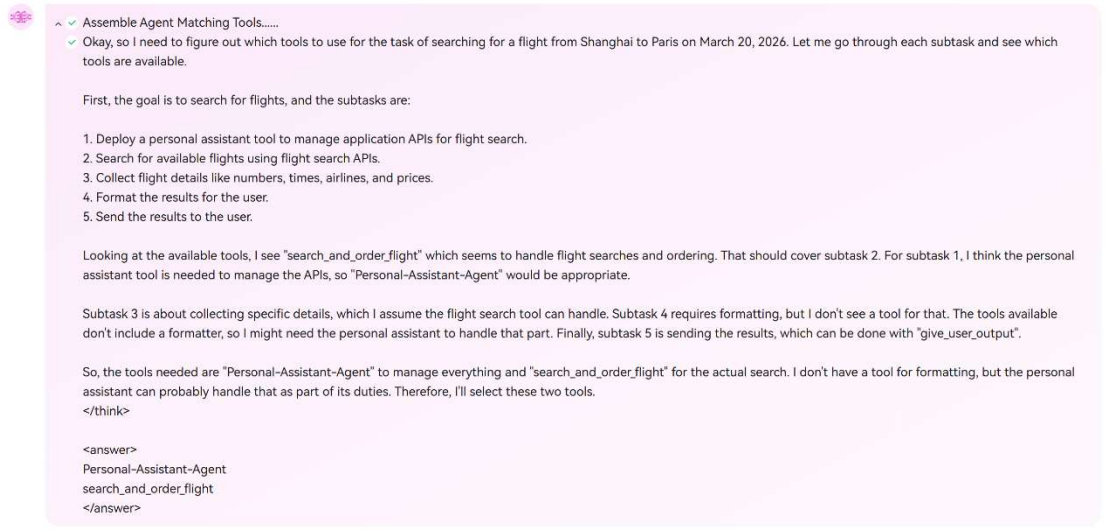


Figure 3-6: The process of selecting tools by the assemble agent

Additionally, the assemble agent designs a workflow for these tools. Firstly, it identifies the dependencies between their input and output parameters. Secondly, it connects the selected tools in a logical and executable order.

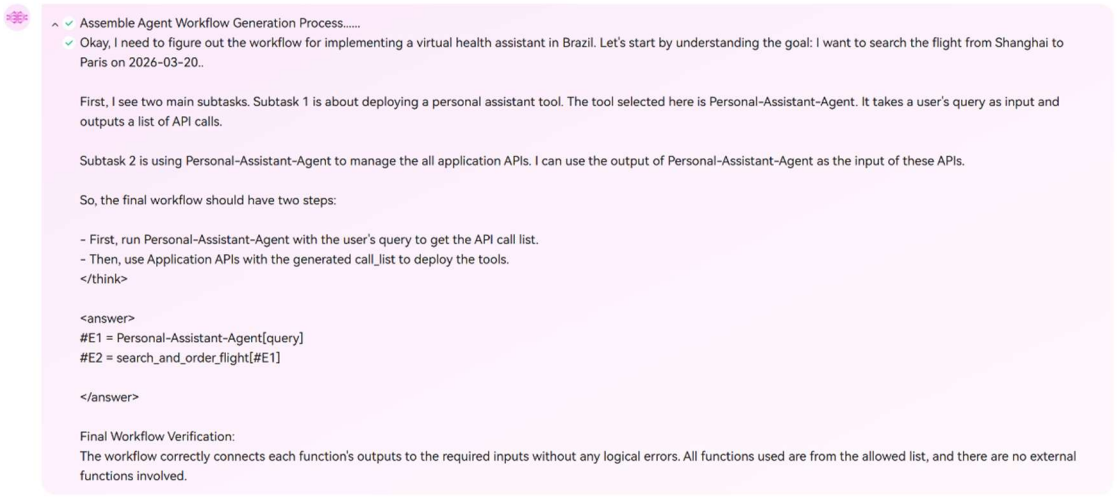


Figure 3-7: The workflow generation process of assemble agent

The workflow dependency chain is defined as follows:

#E1 = Personal-Assistant-Agent[query]

#E2 = search_and_order_flight[#E1]

This means that #E2 depends on #E1, as #E2's input parameter includes #E1. In the workflow diagram, #E2 is visually connected to #E1 to represent this dependency.

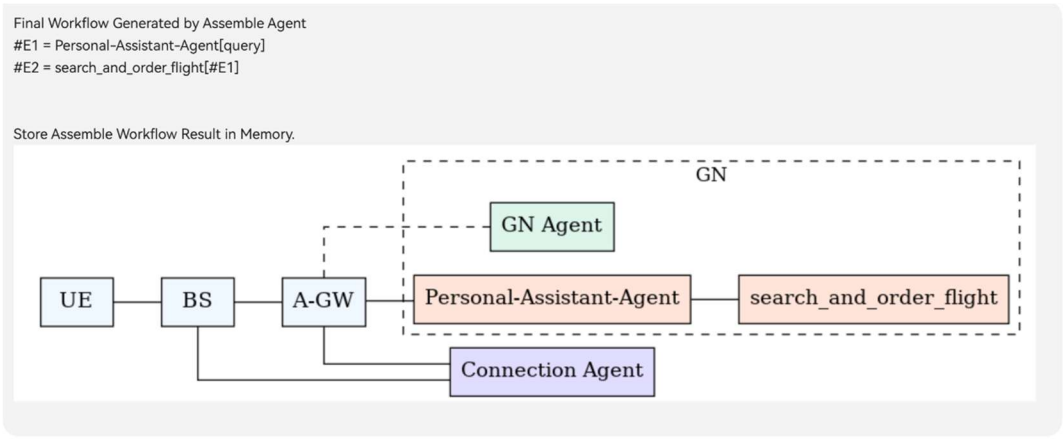


Figure 3-8: The generative network topology diagram

Finally, the assemble agent sends an ABI message to the execution agent, delivering the generated workflow.

In the deployment state, the execution agent deploys the GN workflow on the network. It begins by invoking the Sandbox to validate the workflow according to predefined rules, request sources, and runtime types.

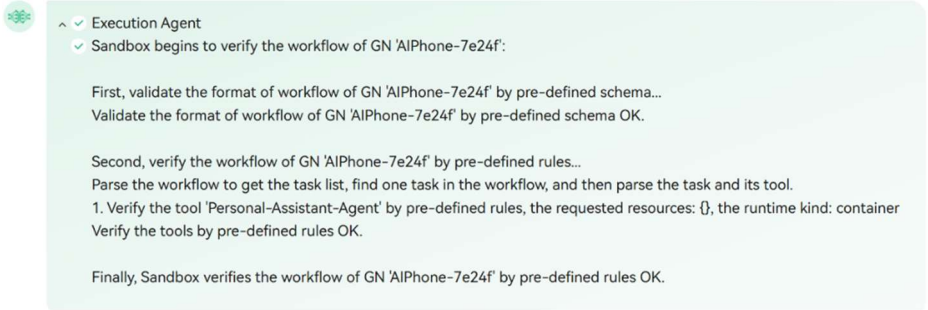


Figure 3-9: The verification process of Sandbox

If the validation result is successful, the execution agent employs its internal tools, such as GN-Controller, Scheduler, and Adaptor, to deploy tool instances for each task.

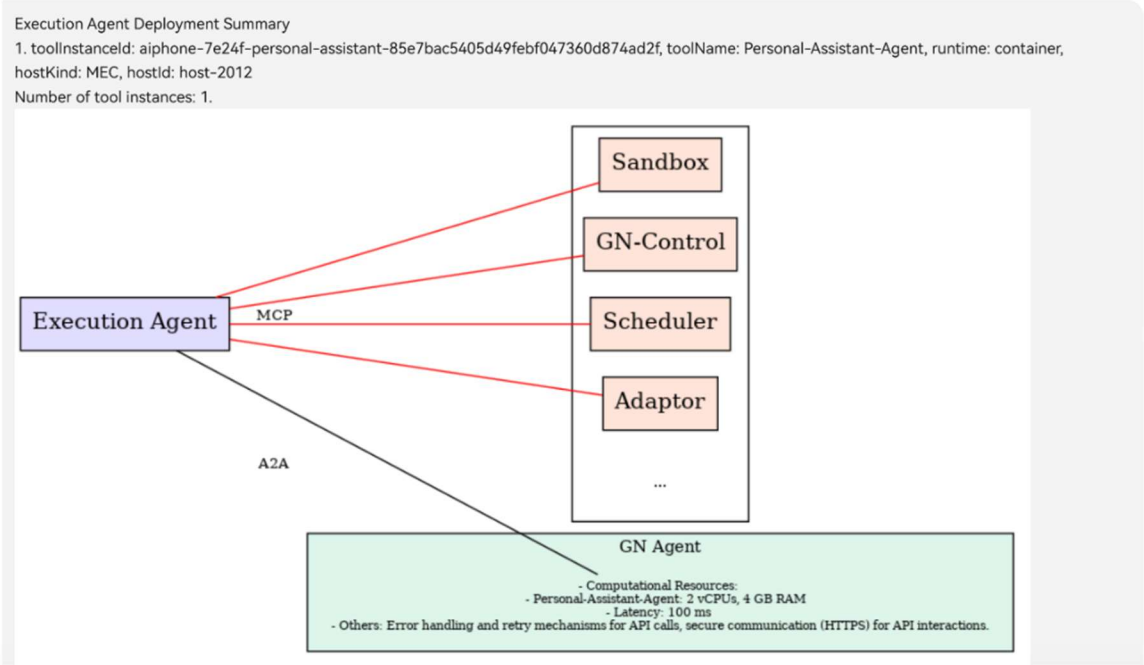


Figure 3-10: The deployment process of tool instances

The Connection Agent receives an ABI message from the Assemble Agent and begins configuring the GN entry. The configuration process involves the following steps:

(1) **Subscription Data Retrieval:** The Connection Agent retrieves subscription data from the Unified Data Management (UDM) via the MCP interface.

(2) **Task Priority and QoS Determination:** The Connection Agent interacts with the Policy Control Function (PCF) to determine the task priority, QoS parameter set, and routing information.

(3) **Session Establishment:** The Connection Agent utilizes the Session Management Function (SMF) to establish a session.

(4) **QoS Parameter Transfer and GN Monitoring:** The Connection Agent transfers the QoS parameter set to the GN Agent via the A2A interface. This enables the GN Agent to monitor the running status of the GN and ensure that the session adheres to the defined QoS constraints.

(5) **IP Address and Port Configuration for RAN:** The Connection Agent communicates the IP address and port number to the Radio Access Network (RAN) over the N2+ interface.

(6) **IP Address and Port Configuration for A-Gateway:** The Connection Agent sends the IP address and port number to the A-Gateway via the N4+ interface. This configuration specifies that the sensing data parameters from the base station are transmitted to the GN through the A-Gateway, ensuring available data streams.

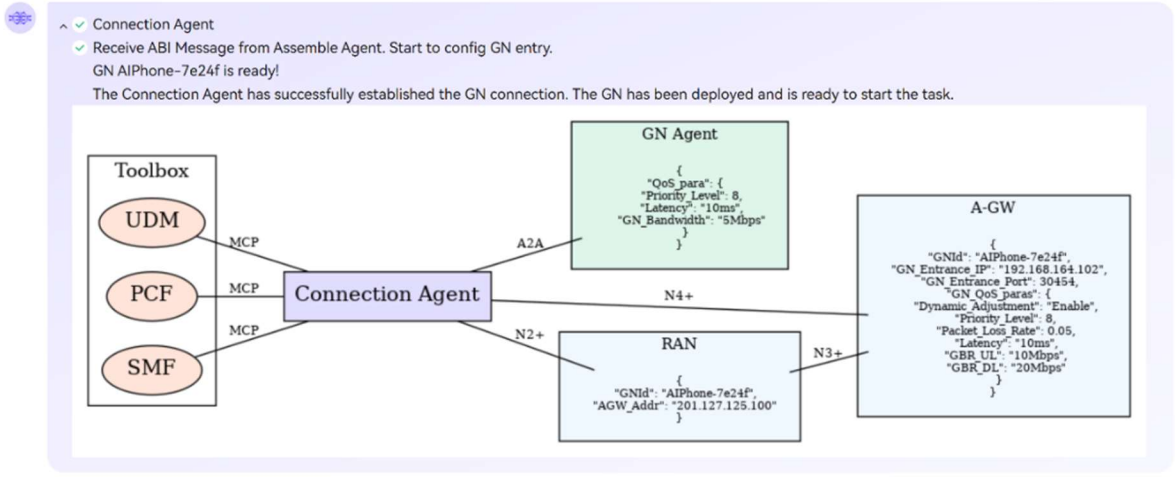


Figure 3-11: The deployment process of Connection Agent

Finally, the deployed tools start executing and generate the GN running log and the final result.

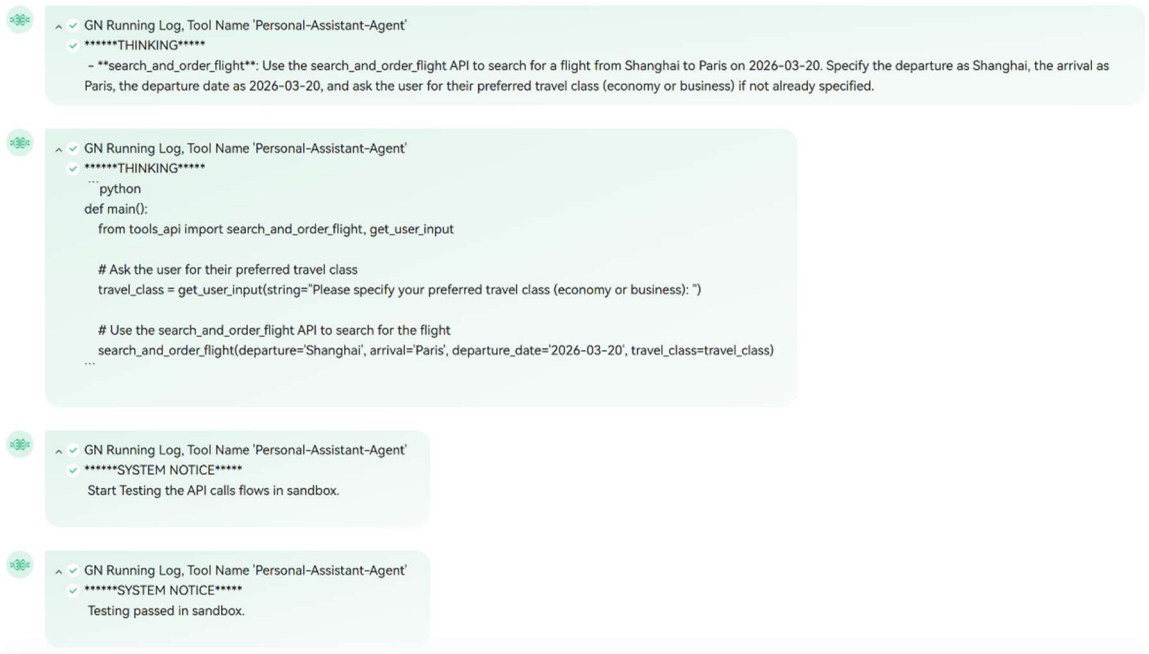
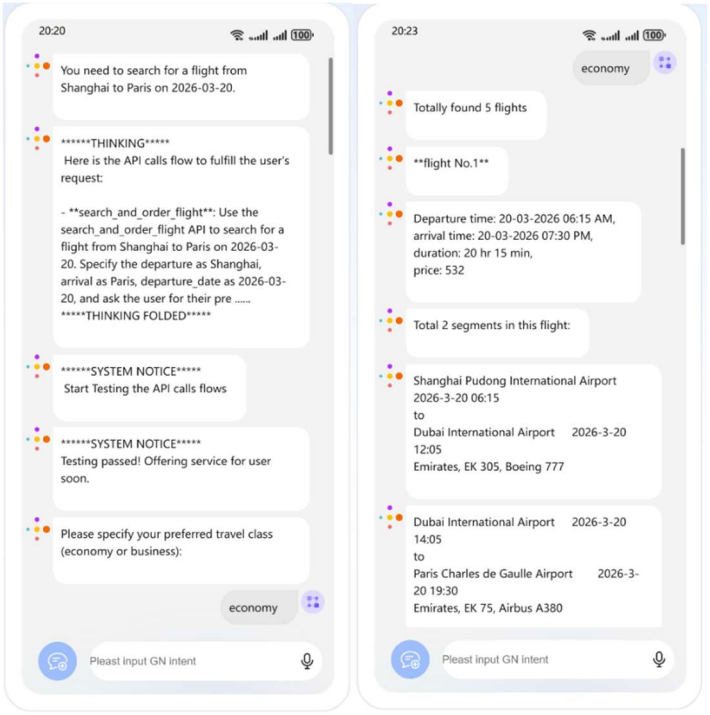


Figure 3-12: The running result of tool instances

Then, the execution result will be shown on AI-Phone GUI.

After the user selects the travel class, e.g., economy class, AI-Core will directly recommend suitable flights to the user. Then, once the user confirms the selection, AI-Core will send the booking link to the user, where the flight has already been pre-selected, including the user's specified date, departure, and arrival locations. The user can then purchase the ticket with just one click.



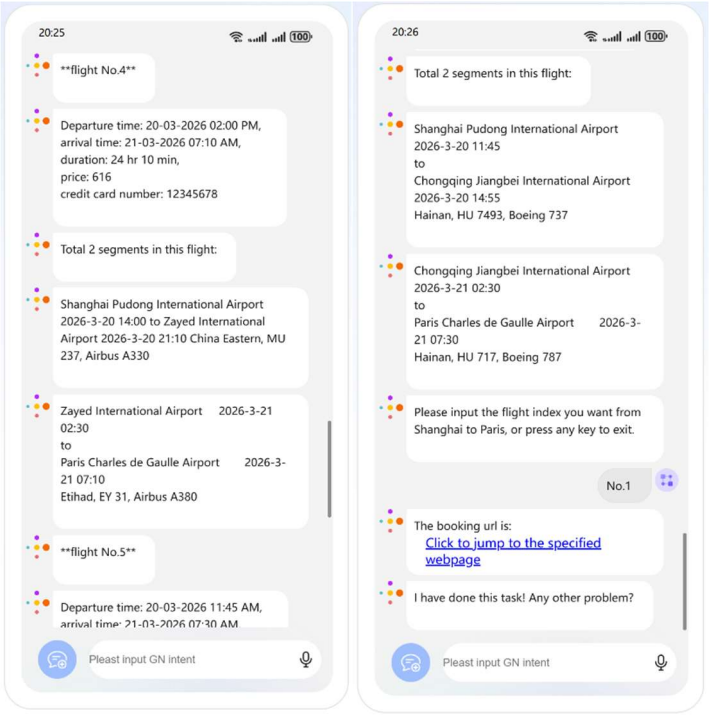


Figure 3-13: The result return to user

After execution, this GN will be decommissioned to release resources. At this point, the entire GN generation and execution process is complete.

2. Case 2: Network-assisted robot

Mobile robots are expected to become fully autonomous entities capable of handling complex tasks. AI-Core can empower robots through high-level planning, cross-agent task orchestration, world models, and guidance, enabling them to better complete tasks.

In this demo, the following characters will be displayed:

1. Co-creation of robotic service and tailored network services based on user intent to assist the robots in completing various tasks.
2. Generative network-assisted robotic tools, e.g., navigation, motion control, etc.
3. Sandbox verification – digital twin and GN real-time update.

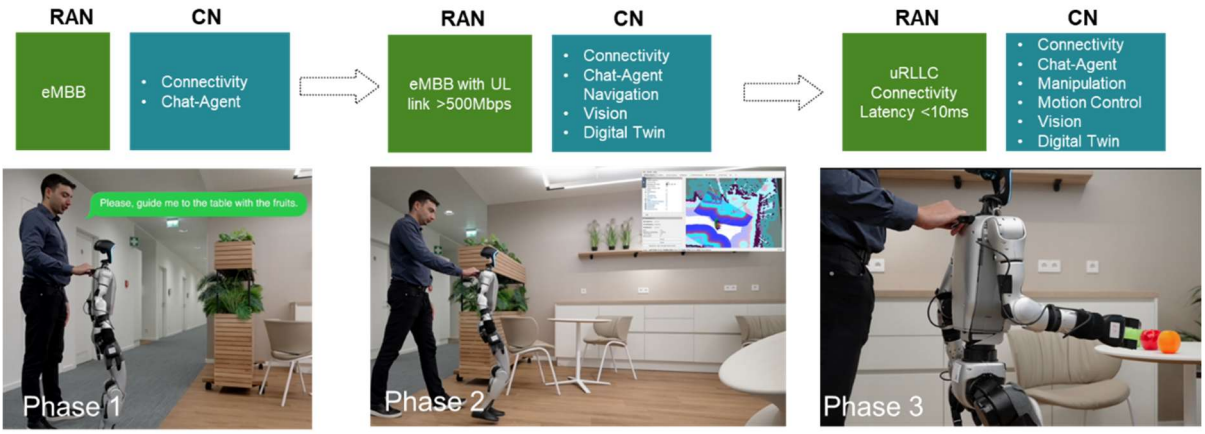


Figure 3-14: Three phases of network-assisted robot scenario

This case has three phases. In phase 1, the robot is in "ON" state and the basic GN is instantiated. Based on the basic GN service, the human could start talking with the robot.

In phase 2, GN is updated with navigation capabilities. Some other tools, such as robot-navigation-tool and robot-vision-tool, are added to the workflow.

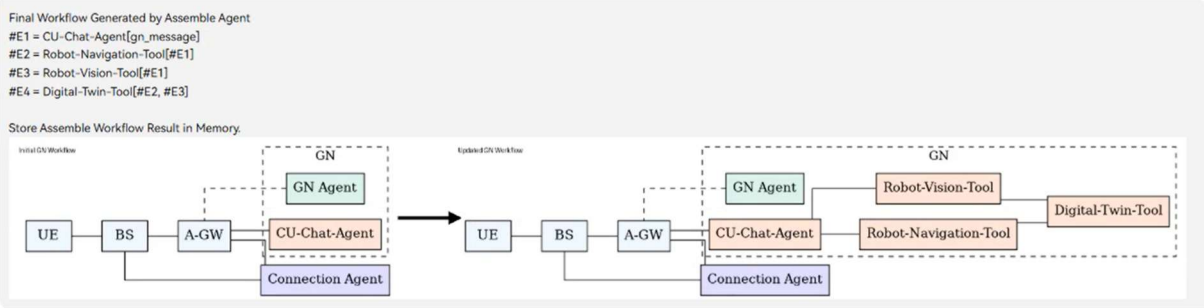


Figure 3-15: GN update of robot

Before executing, a customized agent referred as *cu_chat_agent* in figure 3-15, triggers the Sandbox (Digital-Twin-Tool) component for verification. If the verification process is successful, the response of the sandbox indicates the success to the assemble agent. If negative, a replanning is initiated.

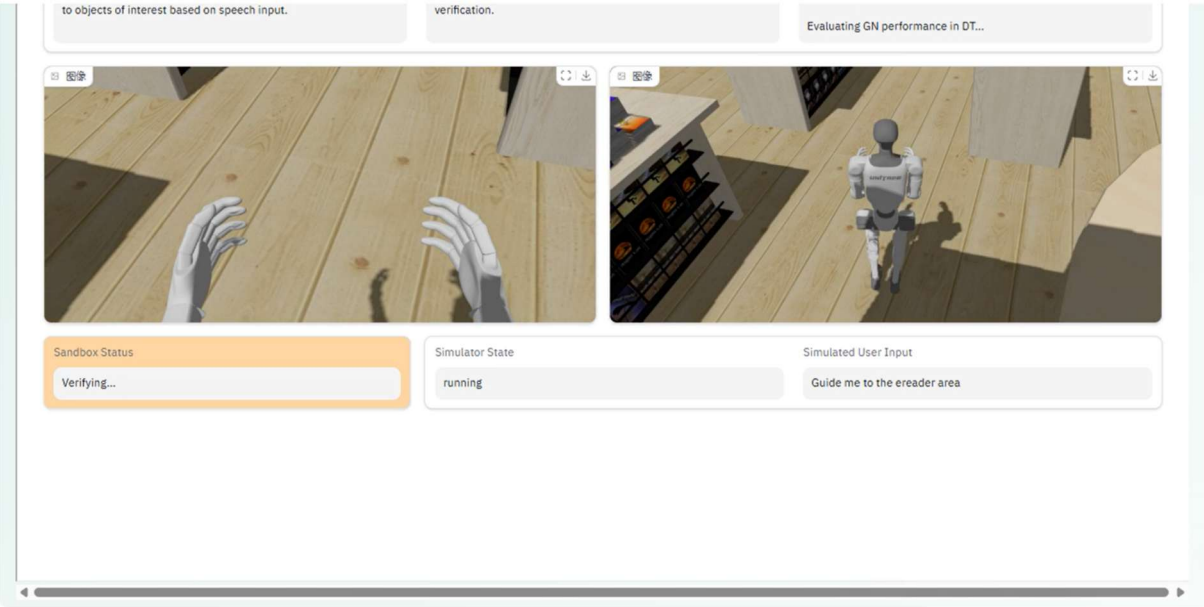


Figure 3-17: Process of Sandbox verification

Then, the GN is updated with new tools.

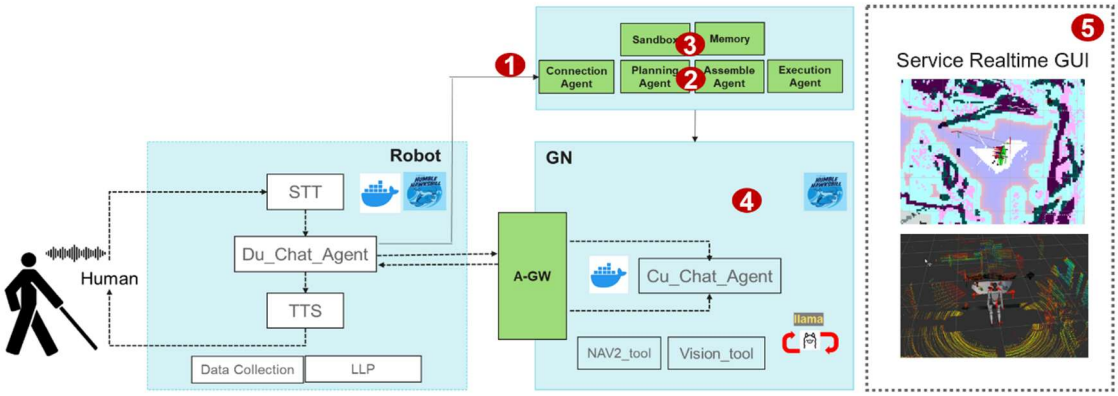


Figure 3-18: Process of Phase 2

In phase 3, AI-Core controls the robot to pick the items, e.g., fruits, and bring to the user to complete the task.

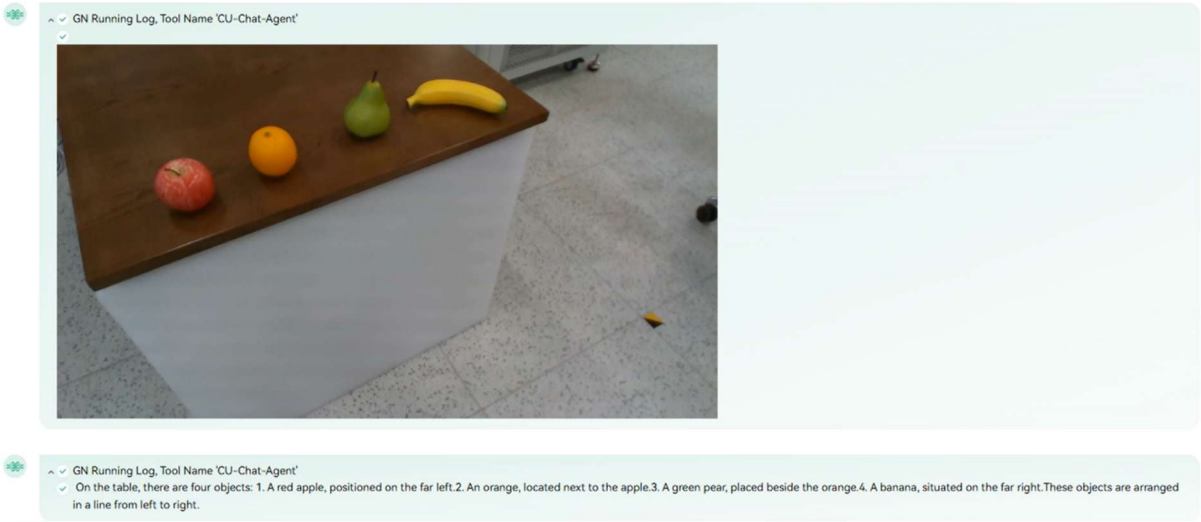


Figure 3-19: Process of Phase 3

The other intermediate steps are basically the same as those in AI-Phone use cases.

3. Conclusion

PoC #24 aims to demonstrate the collaboration between multiple AI agents via two use cases, where the user provides an intent in natural language to the mobile network. It shows the generation and execution of a workflow upon receiving the intent and provision of novel services to the user beyond connectivity. The intent processing capability calls for native integration of LLM-based AI agents into the mobile network architecture equipped and supported with further tools and verification capabilities to improve successful task completion rate. The PoC #24 demonstrates a good example of how new business opportunities and products for telecommunications sector can be created by leveraging generative AI.