Deployment of Genuine Multi-Agent Pipelines for Near-Real-Time Control of 6G Network Services

P. González[1], M. Angoustures[2], A. Zahir[3], S. Barzegar[1], M. Ruiz[1], M. Groshev[3], V. Lefebvre[2], and L. Velasco[1*]

[1] Universitat Politècnica de Catalunya, Spain; [2] sarl TAGES SOLIDSHIELD, France;
[3] Universidad Carlos III de Madrid, Spain;
**e-mail: luis.velasco@upc.edu*

ABSTRACT
The near real-time control of 6G network services requires decision-making to be placed as close to network devices as possible. A possible solution is to deploy a distributed system with intelligent agents that relieves the classical centralized control plane from such near-real-time control loops. However, distributing decision-making entails new vulnerabilities that attackers can exploit. This demonstration will showcase the solution devised by the DESIRE6G project to secure such distributed intelligence, which includes secure communications, as well as remote attestation to verify agents' integrity, both supported by immutable transactions based on a blockchain system.

Keywords: Distributed intelligence, Multi-agent systems; Mutual attestation, Blockchain

1. OVERVIEW

Autonomous network operation in near-real-time is essential to efficiently manage the expected high traffic variability and meet the stringent performance requirements of beyond 5G and 6G Network Services (NS). While centralized solutions can potentially decrease operational costs, they might lead to inefficient resource utilization due to slow response times. To minimize these delays, control algorithms, or (intelligent, e.g., based on Machine Learning (ML)) *multi-agents*, can be executed close to the data plane devices. Implementing such distributed multi-agent systems (MAS) entails agents interacting with other peers, sharing data and knowledge for end-to-end service control [1]. However, they also present several security concerns, as they are more vulnerable to some attacks than centralized approaches. Specifically, attacks can be crafted against such distributed systems that target: (*i*) the integrity of the agents, and (*ii*) the inter-agent communication. For the first type of attack, trusting agents is crucial and can be elaborated on first place through authentication, which assures the place of origin and integrity of the deployed agents. We have designed an ad-hoc remote attestation scheme adapted to the distributed nature of the MAS, featuring cumulative security above cumulative intelligence. For the second type of attack, we rely on encryption, where keys are distributed to the agents only after they successfully pass the attestation procedure. For both, attestation and key distribution, Distributed Ledger Technologies (DLT), such as *blockchain*, offer a promising solution by providing a secure, shared, immutable, and decentralized ledger of transactions. Starting from our previous work in [2],[3], in this demonstration, we present a set of strategies designed to improve the security of MAS, focusing on their integrity and internal communications.

2. INNOVATION

The demonstration will show a complete solution integrating several components, developed as part of the DESIRE6G (D6G) project under Horizon Europe [4]. The main innovations of this demonstration are: (*i*) a Machine Learning Function Orchestrator (MLFO) that coordinates the deployment of agents and their configuration, with the help of a Service Management and Orchestration System (SMO) and an Infrastructure Management Layer (IML); (*ii*) Security-as-a-Service (SECaaS) automatic executable rewriting tool, that creates hardened, authenticable and self-monitored agents; and (*iii*) the DESIRE6G DLT, combining blockchain and smart contracts. A key aspect is the showcase of agents being dynamically associated to the DLT to exchange keys and mutual attestation records through specifically designed smart contracts deployed for the NS.

Near-real-time autonomous network operation, e.g., based on ML algorithms, is limited by the centralized nature of software-defined networking (SDN) and therefore, new approaches need to be considered to control highly dynamic NSs [5]. One of these approaches consists in delegating the near-real-time decision-making to agents deployed close to the data plane to minimize response times, while providing the required overall supervision of the process. However, security issues related to such distributed approach need to be solved. This demonstration will answer these and other related questions, and, in consequence, it will attract the community's attention and be of interest to a broad ICTON audience. In particular, this demonstration is specifically designed for those operators and vendors interested in network automation and secure solutions.

This research was partially funded by the Smart Networks and Services Joint Undertaking under the European Commission Horizon Europe DESIRE6G project (G.A. 101096466), by the AEI through the IBON project (PID2020-114135RB-I00), and by the ICREA institution.

979-8-3503-7732-3/24/$31.00 ©2024 IEEE

3. DEMO CONTENT & IMPLEMENTATION

A. Architecture overview

Figure 1 shows an overview of the essential DESIRE6G components required for executing the secure distributed intelligence. The architecture comprises multiple D6G sites in different geographical locations, each equipped with local networking, computing, and storage resources. Central to this architecture is the D6G SMO layer, which is in charge of the orchestration and lifecycle management aspects of the NS, providing guidelines for the agents and enabling them to operate autonomously. Within the SMO, an MLFO determines the deployment locations and connections of agents. A SECaaS tool is used to improve the security of the agents by adding distributed attestation sub-routines for measurement and verification to their software. The Service Orchestrator (SO) is the responsible component in the SMO to coordinate the deployment, where the IML is in charge of the interactions with the local virtual infrastructure.

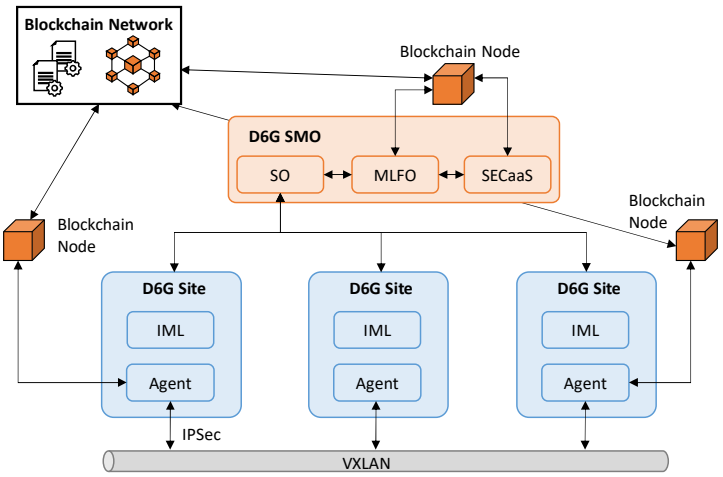

Figure 1: DESIRE6G architecture for secure distributed intelligence

Finally, a blockchain infrastructure is also part of the D6G solution that sets up two distinct smart contracts. The first contract dynamically manages the measurement and verification functions of agents, enabling mutual remote attestation mechanisms between agents. This allows each agent to check the integrity of another agent in the system, avoiding a centralized, Deny-of-Service-vulnerable verifier. The second contract ensures a secure exchange of keys/secrets for inter-agent communications. It is worth mentioning that the DESIRE6G architecture offers a multi-use case blockchain framework, which, besides securing the MAS, also supports a federation of services [6].

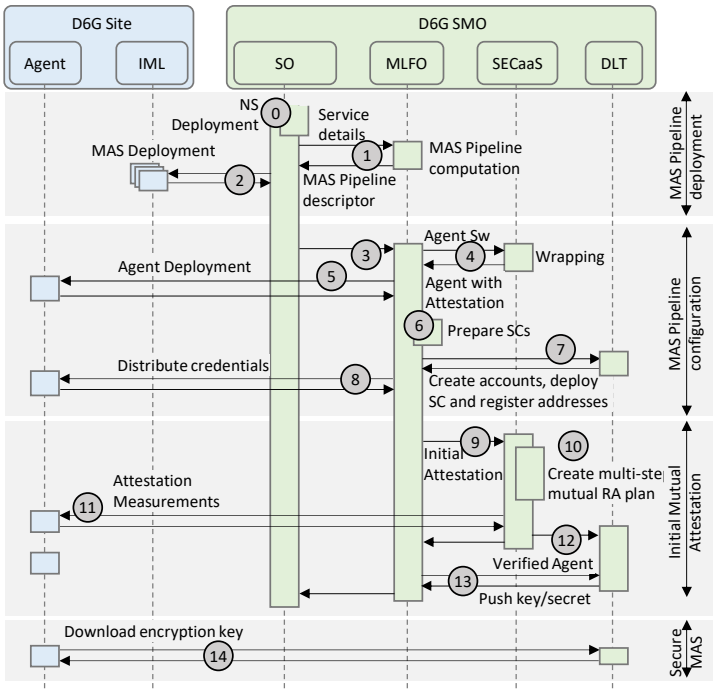

Figure 2: MAS pipeline deployment with initial mutual attestation

B. Overview of MAS deployment

Figure 2 outlines the proposed workflow for the MAS Pipeline deployment, which includes the MAS agents, their initial mutual attestation process, and their connectivity. This workflow consists of four phases: (*i*) deployment of the MAS pipeline; (*ii*) configuration of protected agents through a SECaaS binary hardening tool; (*iii*) mutual attestation; and (*iv*) secret/key exchange. The workflow is initiated by the SO during the deployment of the NS (step 0 in Figure 2).

The SO starts with deploying the MAS pipeline and requests the computation of the topology of a MAS pipeline given the relevant details of the NS, i.e., the location of the virtual network functions (VNF), the performance required for that service, etc. (1). The MLFO computes the optimal MAS pipeline and returns a descriptor that specifies the agents': deployment location, booting image, and connectivity. The SO then interacts with the IML component at different D6G sites to deploy each agent (2). Inter-agent connectivity based on a VXLAN is also deployed in this step (not shown in Figure 2).

Once the agents are running and the connectivity is established, the configuration of the agents starts (3). In this phase, the SECaaS system processes the native agent software (e.g., x86) and improves its security properties by including measurement, verification, and communication subroutines. Through this *wrapping* technique, SECaaS generates a reference measurement of the agent and stores it in a secure repository, ensuring its integrity for future verifications (4). The MLFO then deploys the wrapped images of the agents in the sites (5). Next, it creates smart contracts that will be used to record the state of agents during the remote attestation workflow and store the secrets/keys for the MAS pipeline (6). It also creates blockchain accounts for the agents, including a public address and private key, deploys smart contracts to the blockchain network, and registers the agents' addresses (7). This allows the MLFO to control access to the smart contracts and add or revoke permissions in case of MAS pipeline reconfiguration or detection of infected agents. Once the addresses are registered, the MLFO distributes the blockchain credentials to the agents involved in the MAS pipeline for interaction with the smart contracts (8).

The agent mutual attestation method elaborates a distributed security model where each agent is able to verify the genuineness of another agent it is interplaying. For that, each agent is able to measure itself and verify a measurement delivered by another agent, using the SECaaS stored reference measurement and leading to a remote attestation. A software *root of trust* (RoT) is created from one remote attestation to another, using the last verified agent for the current remote attestation verification task. An ad hoc attestation smart contract elects the verifier agent, the measured agent, and orchestrates the verification by transferring the reference measurement to the verifier; finally, it creates a blockchain block with the remote attestation result including the identification of the agent, the result of the verification, and the time tag. This distributed agent genuineness verification preludes to potential DoS attacks persistent to a central verifier. To face group attacks, where a corrupted agent verifies another corrupted agent, an initial RoT is established for the first agent verification. To that end, the first attestation is triggered by the SECaaS when the MLFO confirms that the MAS configuration has finished (9). For the initial attestation, SECaaS prepares a remote attestation plan and asks the agents to run their attestation sub-routines for measurement and verification (10). The resulting hash value from the measurements (11) is compared to the one stored in the SECaaS repository and if it matches the agent is verified. A transaction is created in the blockchain for each remote attestation, including the test result, the identifiers of the involved agents (i.e., measured and verifier ID), a timestamp, and the updated RoT (i.e., last verified agent) in the first smart contract (12). Once the NS operation phase starts, the smart contract automatically manages the remote attestation process through an internal function that receives a group attestation scheduling plan (the sequence of agents for verification) and notifies the RoT with the reference measurement of the agent, without compromising its confidentiality to other potentially infected agents. Finally, the MLFO pushes to the second smart contract the key/secret that the agents need to use to encrypt the inter-agent communication through the VXLAN (13-14). See the details in [2].

The algorithms and interfaces in the agents and the MLFO have been implemented in Python 3.10.4 and run inside Docker containers. The attestation tool has been implemented in Python 3.10.4 and makes use of three main libraries: *pyelftools* for parsing ELF files; *cryptography* to ensure the creation of and verification of signatures; and *web3.py* to interact with the DLT. The Web-UI displaying the attestation process has been developed using the React framework and the D3.js library for data visualizations. The smart contracts have been written in the Solidity programming language. Opensource MANO (OSM) is the selected orchestration system in charge of the deployment of NSs. OpenDaylight (ODL) SDN controller is on top of the packet network and used to create the connectivity for the MAS pipeline. A scenario with three D6G locations will be showcased, where (OpenStack) is the selected IML in charge of automating the deployment of VNFs. The setup will run in several VMs deployed at the UPC premises in Barcelona Spain, with Ubuntu Server 22.04 LTS as operating system. OSM v.14 and ODL release 16.0 Sulfur will be deployed in a single VM, while three instances of OpenStack release 2023.1 Antelope manage IT resources in the locations. Finally, four container-based DLT nodes, based on the Geth implementation of the Ethereum blockchain, will also be setup.

The workflow of the demonstration will be as follows: *i*) a NS with three VNFs is deployed, which triggers

the deployment of a secure MAS pipeline for its near-real-time control; *ii*) the MLFO configures the DLT for mutual attestation and key exchange and the agents download the key and join the VXLAN; *iv*) telemetry data are collected by the agents. A Web interface will facilitate iteration of the attendees with the system, so they can modify the configuration of the different components of the architecture. Specifically, the deployment of the agents can be interactively verified and controlled through the OSM Web-UI. Figure 3 shows an example of a NS and its dedicated MAS deployed and configured by the MLFO. Several actions can be performed through the Web-UI, including the reconfiguration of a NS, the configuration of an SDN controller or the instantiation of a new VNF descriptor to be used by a network service. In addition, each step of the attestation plan can be visualized through the Web-UI shown in Figure 4.

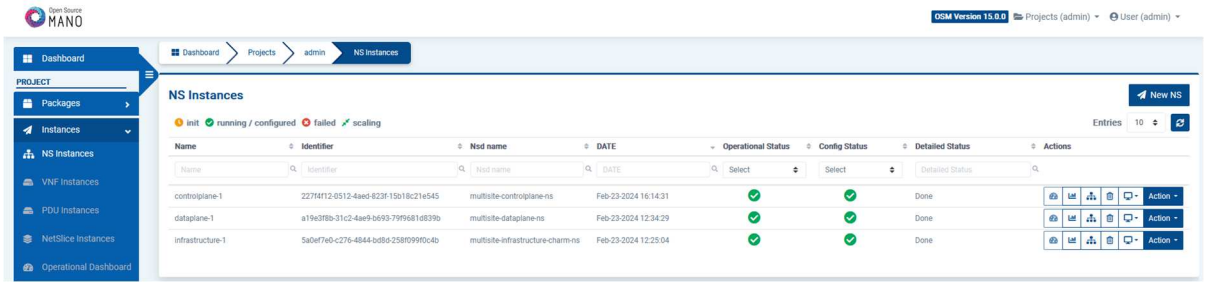

Figure 3 Open-Source MANO Web-UI

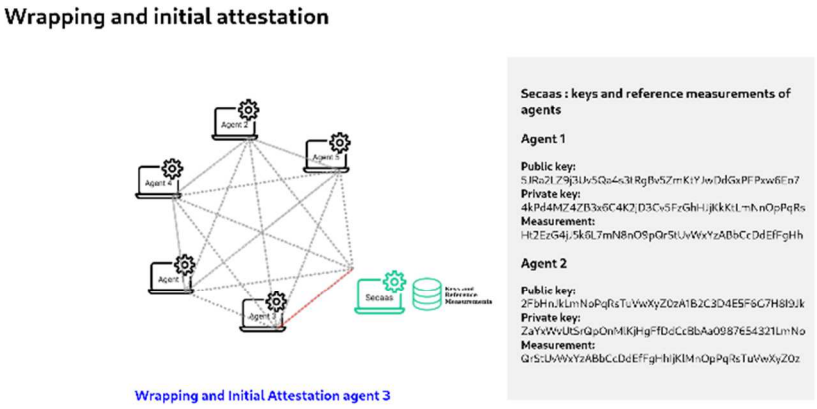

Figure 4 Attestation Tool Web-UI

REFERENCES

[1] S. Barzegar *et al*., "Autonomous Flow Routing for Near Real-Time Quality of Service Assurance," IEEE Transactions on Network and Service Management (TNSM), 2024.

[2] L. Velasco *et al*., "Securing multi-agent systems for near real-time control of 6G services," European Conference on Networks and Communications (EuCNC), 2023.

[3] P. Gonzalez *et al*., "Deployment of Secure Machine Learning Pipelines for Near-Real-Time Control of 6G Network Services," in Proc. Optical Fiber Communication Conference (OFC), 2024.

[4] C. Papagianni, "DESIRE6G: Deep Programmability & Secure Distributed Intelligence for Real-Time E2E 6G Networks," 2023. [Online] https://doi.org/10.5281/zenodo.10351667.

[5] H. Shakespear-Miles *et al*., "Centralized and Distributed Approaches to Control Optical Point-to-Multipoint Systems Near-Real-Time," IEEE/OPTICA Journal of Optical Communications and Networking (JOCN), 2024.

[6] A. Zahir *et al*., "Performance evaluation of private and public blockchains for multi-cloud service federation," in Proc. International Conference on Distributed Computing and Networking (ICDCN), 2024.