A demonstration of a network isolation solution for the computing continuum

A demonstration of a network isolation solution for the computing continuum

Francesco Pizzato, Daniele Bringhenti, Luca Cetino, Riccardo Sisto, Fulvio Valenza
Dip. Automatica e Informatica, Politecnico di Torino, Torino, Italy, Emails: {first.last}@polito.it

***Abstract*—The computing continuum integrates heterogeneous layers into a unified resource pool, introducing challenges in network-level workload isolation. This demonstration presents an intent-based solution to automate and enforce network isolation across such multi-user and non-uniform environments.**

***Index Terms*—computing continuum, network isolation**

I. Overview

The computing continuum is an innovative paradigm integrating cloud, edge, and IoT resources into a seamless pool of available assets, allowing for the transparent consumption of network, computation, and storage resources [1]. Initiatives like [2] are pushing for an ambitious vision of the continuum, where users are not just consumers of the shared resources, but also providers offering their own resources to other users.

While powerful, the computing continuum introduces significant security challenges. A highly relevant one is enforcing network isolation for workloads deployed within the continuum. In this context, the classical idea of static physical boundaries separating tenants' resources is not valid anymore. Instead, the dynamism of the continuum transformed them into ephemeral borders, which span across multiple heterogeneous machines and evolve dynamically following user-defined policies or runtime necessities. Within such a dynamic and multi-tenant environment, it is necessary to manage network communications between workloads, safeguarding them from possible interference from the host or other co-located applications.

To address this problem, we developed a novel solution as part of the FLUIDOS [3] project and presented it in [4]. This solution consists in a security orchestrator that automates network isolation in a multi-tenant computing continuum environment by refining high-level user intents into the configuration of low-level primitives, i.e., Kubernetes Network Policies. The designed workflow consists of three main steps: verification, harmonization, and optimization. Their combination allows the intelligent resolution of possible conflicts between users of the continuum and the decoupling of the security definition from its implementation. This demonstration showcases a working prototype [5] of the solution throughout its three main phases.

II. Innovation

The main innovations and challenges related to the way our solution aims to address this problem are the following:

High-level security intents. Due to the large scale of the continuum and its heterogeneous composition, the traditional manual approaches based on static configuration are nearly impossible to adopt. Our solution uses a high-level intent language designed to abstract this complexity away, as tenants express their security needs with a user-friendly but powerful syntax and our solution automatically handles the configuration of network security primitives and their enforcement.

Support for simultaneous and multiple goals. Multiple network isolation goals are simultaneously present within the continuum. The consumers want to be safeguarded from potential interference caused by the host or co-located tenants. Instead, the providers want to be protected from unintentional data exfiltration from hosted applications. Thanks to the designed intent language and multiple types of intents, our solution allows the expression of all these security isolation objectives.

Decoupling from specific implementation. The continuum is inherently heterogeneous, so different components support different network security primitives. The designed approach solves this challenge by decoupling the definition of network isolation intents from the specific implementation, which is carried out within a dedicated phase, i.e., translation, that can be specialized to different backend technologies.

Automated conflict resolutions. The continuum presents multiple and simultaneous network isolation goals. Therefore, conflicts arise between them, e.g., a communication requested by a consumer is not allowed by a provider. The presented solution analysed the set of conflicts that may arise in such environments and defined a resolution strategy for all of them. This is mainly carried out in the harmonization phase.

III. Demo

The demo proposal follows the architecture and workflow presented in Fig. 1. The setup includes multiple Kubernetes clusters: one consumer cluster runs an e-commerce application, and three providers offer some services and resources, but have different wills in terms of authorized communications for the hosted resources. The clusters are virtualized on a single laptop using KinD [6] and are configured with Calico CNI [7]. Every cluster contains a Secure Border Controller (SCB), i.e., a Kubernetes controller implementing our solution. In the demo setup, each cluster also features external components, which are not part of our proposed solution but necessary for its execution, such as the FLUIDOS Node, developed by the FLUIDOS project for resource advertising and acquisition, and Liqo [8], responsible for the interconnecting fabric.

Intent specification phase. Initially, the consumer specifies a request for resources and the associated network isolation

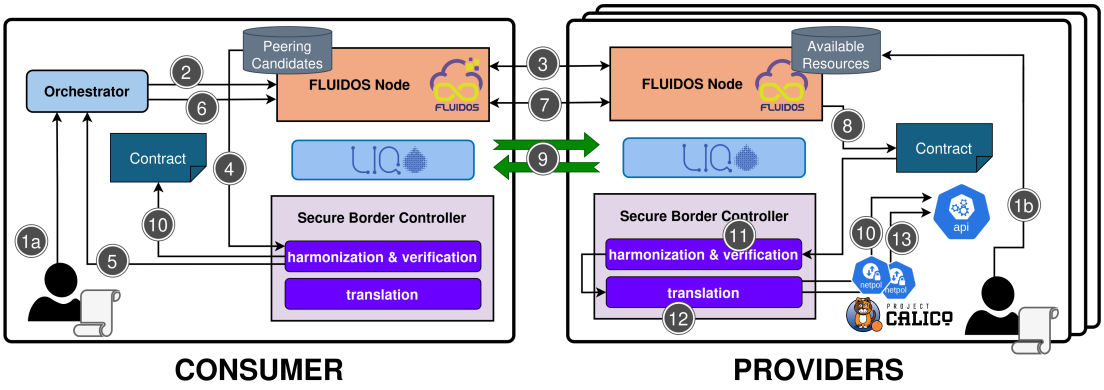

Fig. 1: Schema of the intent-based network isolation workflow.

intents. These requested intents, as well as the providers' authorization intents, are defined with XML files following the designed intent format. On the consumer's side, the intents are passed to an orchestrator (1a), whereas on each provider, these are injected into the specifications of available resources (1b).

Resource acquisition phase. Once it receives the consumer's request, the orchestrator interacts with the local FLUIDOS Node (2), which contacts all remote counterparts on providers' clusters to retrieve the specification of their available resources (3). Each specification describes the resource types and the associated authorizations as defined by the provider's intents, e.g., some blocked IP ranges or mandatory probes needed for monitoring or compliance reasons. Since these authorizations may be incompatible with the consumer's isolation requests, they are processed by the verification module within the SBC (4). Its goal is to aid the consumer in selecting the most compatible resource offer, among the three, with her network isolation needs. The result is communicated to the orchestrator (5) and acquired through the interaction between local and remote FLUIDOS Nodes (6-7). Once completed, a Contract is created (8). This element is a Kubernetes object containing information about the acquired resource, consumer, and provider, and is used by Liqo to create the fabric interconnecting the clusters (9). It also specifies the associated network isolation intents and will be the source for this information in the next phases.

Resource usage phase. Once the connection is established, the orchestrator starts to deploy resources on the rented share of the provider's resources. The SBC reacts to this event and pushes new configuration rules on the provider's cluster to isolate deployed applications from any other resource on the same cluster (10). Then, the harmonization module of the SBC processes the consumer's network isolation intents and the provider's network authorization intents (11). The goal is to solve all conflicts (e.g., a communication requested by a consumer toward an external endpoint is not allowed by the provider) and to produce a harmonized set of intents. Then, this is passed to the translation module (12), which refines the intents into YAML manifests for the Kubernetes Network Policy necessary to satisfy the network isolation objectives of both consumer and provider (13).

Validation phase. Finally, the demonstration involves several attempts at network connections among different pods. The scope is to visually confirm that the allowed communication succeeds, while all unauthorized traffic is effectively blocked, confirming the system's correctness.

IV. Relevance

This demonstration is highly relevant to the IEEE SDN/NFV audience and communities for several reasons. First, it introduces an innovative approach to manage computing continuum environments through high-level intents, aligned with the principles of Intent-Based Networking. As the complexity and heterogeneity of such environments make manual security configuration impractical, the adoption of intent-based automated solutions represents a compelling and effective alternative. Second, as modern infrastructures evolve beyond traditional boundaries, ensuring reliable and correct network isolation becomes essential, because it is complex to handle boundaries now spanning across cloud, edge, and IoT layers. Finally, the demonstration tackles challenges related to multi-tenancy and conflicting security objectives, issues that are becoming increasingly relevant as the cloud shifts toward interconnected ecosystems, spanning across multiple layers.

Acknowledgment

This work was partly supported by EU Horizon project FLUIDOS, under grant agreement 101070473.

References

[1] M. Tortonesi, "The compute continuum: Trends and challenges," *Computer*, vol. 58, no. 3, pp. 105–108, 2025. [Online]. Available: https://doi.org/10.1109/MC.2024.3520255
[2] EUCEI, Available: https://eucloudedgeiot.eu/, Visited: 2025-06-12.
[3] FLUIDOS, Available: https://www.fluidos.eu/, Visited: 2025-06-12.
[4] F. Pizzato, D. Bringhenti, R. Sisto, and F. Valenza, "An intent-based solution for network isolation in kubernetes," in *10th IEEE International Conference on Network Softwarization*, 2024, pp. 381–386. [Online]. Available: https://doi.org/10.1109/NetSoft60951.2024.10588939
[5] Secure Border Controller, Available: https://github.com/netgroup-polito/secure-border-controller , Visited: 2025-06-12.
[6] KinD, Available: https://kind.sigs.k8s.io, Visited: 2025-06-12.
[7] Calico, Available: https://docs.tigera.io, Visited: 2025-06-12.
[8] Liqo, Available: https://liqo.io, Visited: 2025-06-12.