An Intent-Based Management System for In-Network Computing Functions in Intelligent Moving Objects

Yoseop (Joseph) Ahn*, Mose Gu*, Jaehoon (Paul) Jeong*, and Yiwen Shen†
* Department of Computer Science & Engineering,
Sungkyunkwan University, Suwon, Republic of Korea
† Department of Software & Computer Engineering, Ajou University, Suwon, Republic of Korea
Email:{ahnjs124, rna0415, pauljeong}@skku.edu, chrisshen@ajou.ac.kr

***Abstract*—As real-time security and adaptive network control become critical in intelligent moving environments, centralized security frameworks like Interface to Network Security Functions (I2NSF) face latency challenges. To address this, we propose the Interface to In-Network Computing Functions (I2ICF) framework, which enables intent-based policy management by relocating security functions and translation closer to the data path. I2ICF performs policy translation, enforcement, and monitoring at the edge or within mobile entities, reducing communication delays. We implement I2ICF on a Kubernetes-based testbed and demonstrate its feasibility in orchestrating service functions for intelligent moving objects (IMOs). These results suggest that I2ICF can serve as a foundation for low-latency, intent-driven management in dynamic environments.**

***Index Terms*—Intent, I2ICF, Edge System, Kubernetes, Moving Object**

I. Introduction

Recent advances in artificial intelligence (AI) and digital infrastructure are accelerating the adoption of autonomous, scalable, and programmable systems, particularly in cloud and edge environments. Technologies such as Infrastructure-as-Code (IaC) and Kubernetes are enabling automation, dynamic configuration, and intent-based policy enforcement. Within this context, the Intent-Based System (IBS) paradigm has emerged, allowing users to define high-level goals while the system manages the corresponding execution autonomously to achieve the goals. However, traditional cloud-based security frameworks like I2NSF suffer from latency due to their centralized architecture, making them less suitable for real-time applications involving intelligent moving objects (IMOs). To address this limitation, we introduce the Interface to In-Network Computing Functions (I2ICF) framework, which brings policy interpretation, enforcement, and monitoring closer to the data plane for the data packets of the IMOs.

II. Intent-Based Management Architecture for Intelligent moving objects

Interface to Network Security Functions (I2NSF) [1] is a network security framework proposed by IETF, designed to centrally control and manage various network security functions (NSFs) based on policies. However, when implementing I2NSF on cloud infrastructure, latency becomes a major technical limitation. In the I2NSF architecture, I2NSF user, Security Controller, and NSFs are typically deployed across separate cloud nodes, communicating via application-layer protocols like REST API, RESTCONF, and NETCONF. Since data traffic is often redirected to cloud-based NSFs for security processing (e.g., decryption and inspection), this can introduce several milliseconds of delay, potentially degrading performance in real-time applications. To solve this latency problem, this study proposes an I2ICF framework, which enables policy processing and security functions to be performed on the local network or around the terminal as edge computing.

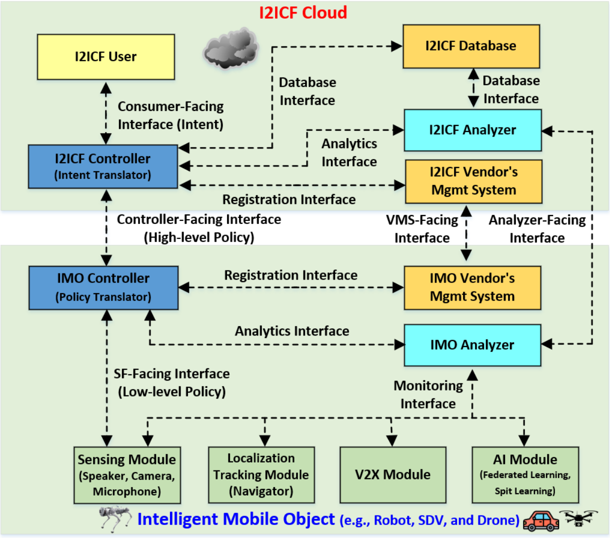

I2ICF Cloud
I2ICF User
I2ICF Database
Database Interface
Database Interface
Consumer-Facing Interface (Intent)
I2ICF Analyzer
Analytics Interface
I2ICF Controller (Intent Translator)
I2ICF Vendor's Mgmt System
Registration Interface
Controller-Facing Interface (High-level Policy)
VMS-Facing Interface
Analyzer-Facing Interface
IMO Controller (Policy Translator)
Registration Interface
IMO Vendor's Mgmt System
Analytics Interface
IMO Analyzer
SF-Facing Interface (Low-level Policy)
Monitoring Interface
Sensing Module (Speaker, Camera, Microphone)
Localization Tracking Module (Navigator)
V2X Module
AI Module (Federated Learning, Spit Learning)
Intelligent Mobile Object (e.g., Robot, SDV, and Drone)

Fig. 1. Interfaces and Components of the I2ICF Framework

An intent-based management is essential between an Edge System and Intelligent Moving Objects for the autonomous network configuration. Fig. 1 shows the interfaces and components for I2ICF Framework. The framework consists of an Edge System and intelligent moving objects. The Edge System consists of I2ICF User, I2ICF Controller, I2ICF Database, I2ICF Analyzer, and I2ICF Vendor's Management System. And the intelligent moving object consists of IMO Controller, IMO Vendor's Management System, IMO Analyzer, and Ser-

978-4-907626-55-6/25/$31.00 ©2025 IPSJ

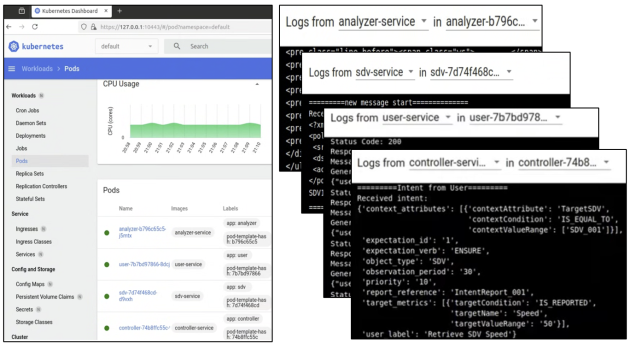
vice Functions (SFs).

- **I2ICF User:** It refers to the software (e.g., a web-based user interface) that allows administrators to deliver network intents into the framework.
- **I2ICF Database:** It is a database for managing data, including network and security configurations, location data, etc.
- **I2ICF Controller (Intent Translator):** It translates a user intent into network or application policies and manages system components.
- **I2ICF Vendor's Management System:** It registers service modules and resources with the I2ICF Controller via the Registration Interface.
- **I2ICF Analyzer:** It collects monitoring data from Intelligent Moving Objects (IMOs), analyzes it, and sends feedback to the I2ICF Controller for reconfiguration to maintain SF performance.
- **IMO Controller (Policy Translator):** It manages IMO modules, translating a high-level policy into a low-level policy that can be executed by the IMO modules (i.e., Service Functions (SFs) of the Intelligent Moving Objects).
- **IMO Vendor's Management System:** It registers IMO modules with the IMO Controller via the Registration Interface.
- **IMO Analyzer:** It collects monitoring data from IMO modules, analyzes their performance, and ensures their functionality.

The I2ICF can effectively reduce delay problems by handling policy decisions, running security functions, and checking system status inside the network or at the edge. I2ICF brings the place where security rules are applied closer to the user or data path by using computing devices (such as edge servers or smartphones) located along the network. This reduces the time the system needs to talk to the central controller and allows data to be handled right away without going through faraway cloud systems. In 5G networks, delay can be reduced even more by running security features directly on an edge server within a User Plane Function (UPF), which is close to where user data flows [2]. The I2ICF system can be set up as either a central cloud or an edge cloud, depending on the network design and what the service needs.

III. Implementation of the I2ICF Framework

We implemented the I2ICF framework testbed for the management of network functions and applications on intelligent moving objects. Fig. 2 represents the entire Kubernetes system [3], where each Service corresponds to an individual component of the I2ICF Framework. These components are encapsulated within a pod, which is the smallest deployable unit in Kubernetes. A pod can contain one or more containers, each running a specific component of the I2ICF Framework. Each component is implemented as a service, providing an abstraction that allows communication between different system parts. The following describes the roles and workflows of each service component in the implemented I2ICF framework.



Pods

Name	Images	Labels
analyzer-b796c65c5-j5mtx	analyzer-service	app: analyzer pod-template-hash: b796c65c5
user-7b7bd97866-8dcj	user-service	app: user pod-template-hash: 7b7bd97866
sdv-7d74f468cd-d9vxh	sdv-service	app: sdv pod-template-hash: 7d74f468cd
controller-74b8ffc55c-	controller-service	app: controller pod-template-hash: 74b8ffc55c

```
=========new message start==============
```

```
Status Code: 200
```

```
=========Intent from User=========
Received intent:
{'context_attributes': [{'contextAttribute': 'TargetSDV',
                         'contextCondition': 'IS_EQUAL_TO',
                         'contextValueRange': ['SDV_001']}],
 'expectation_id': '1',
 'expectation_verb': 'ENSURE',
 'object_type': 'SDV',
 'observation_period': '30',
 'priority': '10',
 'report_reference': 'IntentReport_001',
 'target_metrics': [{'targetCondition': 'IS_REPORTED',
                     'targetName': 'Speed',
                     'targetValueRange': '50'}],
 'user_label': 'Retrieve SDV Speed'}
```

Fig. 2. Implementation for I2ICF Framework on Kubernetes

- **I2ICF User Service:** The I2ICF user sends an intent to the I2ICF Controller, which interprets it and generates high-level network and application policies.
- **I2ICF Controller Service:** The network policy is sent to the Policy Translator within the Edge Controller. The application policy is sent to the IMO Controller, which interprets it for object configuration.
- **IMO Controller Service:** The IMO Controller applies the policy to adjust the operational parameters. Moving objects monitor their status (e.g., speed, direction, and camera image) and send updates to the IMO Analyzer.
- **I2ICF Analyzer Service:** The IMO Controllers create monitoring reports and forward them to the I2ICF Analyzer. The I2ICF Analyzer evaluates the reports and stores the data for further analysis.

IV. Conclusion

In this paper, we introduced the I2ICF framework for network and application policy adjustment in intelligent moving objects (IMOs). The orchestration of virtualized network and application functions for the IMOs can be seamlessly achieved in a Kubernetes-based environment. As future work, we will manage policy enforcement and adjustment for IMO network and application functions over a 5G network (e.g., Free5GC) using Kubernetes. We will also measure network performance when Kubernetes is expanded to multiple nodes in edge computing environments in 5G networks.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (MSIT), South Korea (No. RS-2024-00398199 and RS-2022-II221199). Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] D. Lopez, E. Lopez, L. Dunbar, J. Strassner, and R. Kumar, "Framework for Interface to Network Security Functions," RFC 8329, Feb. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8329

[2] S. Islam, A. Zainab Abdulsalam, B. Anil Kumar, M. Kamrul Hasan, R. Kolandaisamy, and N. Safie, "Mobile networks toward 5g/6g: Network architecture, opportunities and challenges in smart city," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 3082–3093, 2025.

[3] G. Budigiri, C. Baumann, J. T. Mühlberg, E. Truyen, and W. Joosen, "Network policies in kubernetes: Performance evaluation and security analysis," 2021, pp. 407–412.