



# IETF-122 Hackathon



## Interface to In-Network Computing Functions (I2ICF) Project

- March 15-21, 2025, Bangkok

**Champion: Jaehoon (Paul) Jeong**

**Members: [Xudong Wang](#), Yoseop Ahn, Byoungman Robert An**

**Department of Computer Science and Engineering at SKKU**

**Korea Electronics Technology Institute**

**Email: {wangxudong28, pauljeong, ahnjs124}@skku.edu, bman@keti.re.kr**

# IETF-122 Interface to In-Network Computing Functions (I2ICF)

Champion: Jaehoon (Paul) Jeong (SKKU)



## IETF-122 Interface to In-Network Computing Functions (I2ICF) Hackathon

### Professors:

- Jaehoon (Paul) Jeong (SKKU)
- Yiwen (Chris) Shen (AJU)

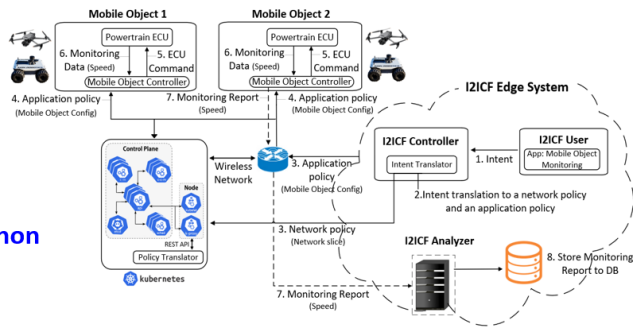
### Researchers:

- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)
- Byoungman Robert An (KETI)

### Students:

- Yoseop Ahn (SKKU)
- Xudong Wang (SKKU)
- Bien Aime Mugabarigira (SKKU)
- Mose Gu (SKKU)
- Juwon Hong (SKKU)
- Jiwon Suh (SKKU)

### Architecture



### What to pull down to set up an environment:

- OS: Ubuntu 18.04 & 20.04
- Kubernetes: Microk8s v1.32.2
- ROS version: Melodic
- GitHub Repository:  
<https://github.com/jaehoonpauljeong/I2ICF/tree/main/IE TF-122>

### Workflow of the I2ICF Testbed on Kubernetes

- I2ICF User sends an intent to the I2ICF Controller.
- The I2ICF Controller's Intent Translator converts the intent into Network and Application Policies.
- The translated Network Policy is forwarded to the wireless network components.
- The translated Application Policy for Mobile Object configuration is sent to each Mobile Object Controller.
- Each Mobile Object Controller enforces the received application policy on the Powertrain ECU, adjusting the operational parameters as needed.
- The Mobile Objects continuously monitor operational data (e.g., speed and direction) and transmit it to their Mobile Object Controllers.
- The Mobile Object Controllers set this data into Monitoring Report and forward them to the I2ICF Analyzer.
- The I2ICF Analyzer processes the monitoring report to assess the performance of the applied policies and stores the results in a database for further analysis.

### Objectives

- To demonstrate Interface to In-Network Computing Functions (I2ICF).
- To build a comprehensive framework for intent-based management of networks, security, and applications in Mobile Objects (e.g., SDV and Drone) within the Kubernetes environment.

### Future Work

- We plan to use In-Context Learning and Prompt Tuning to design and implement the intent translator.
- The development of data analytics models, including Machine Learning and Deep Learning, for real-time diagnosis of Service Functions (SFs) in Mobile Objects to enhance safety and security.

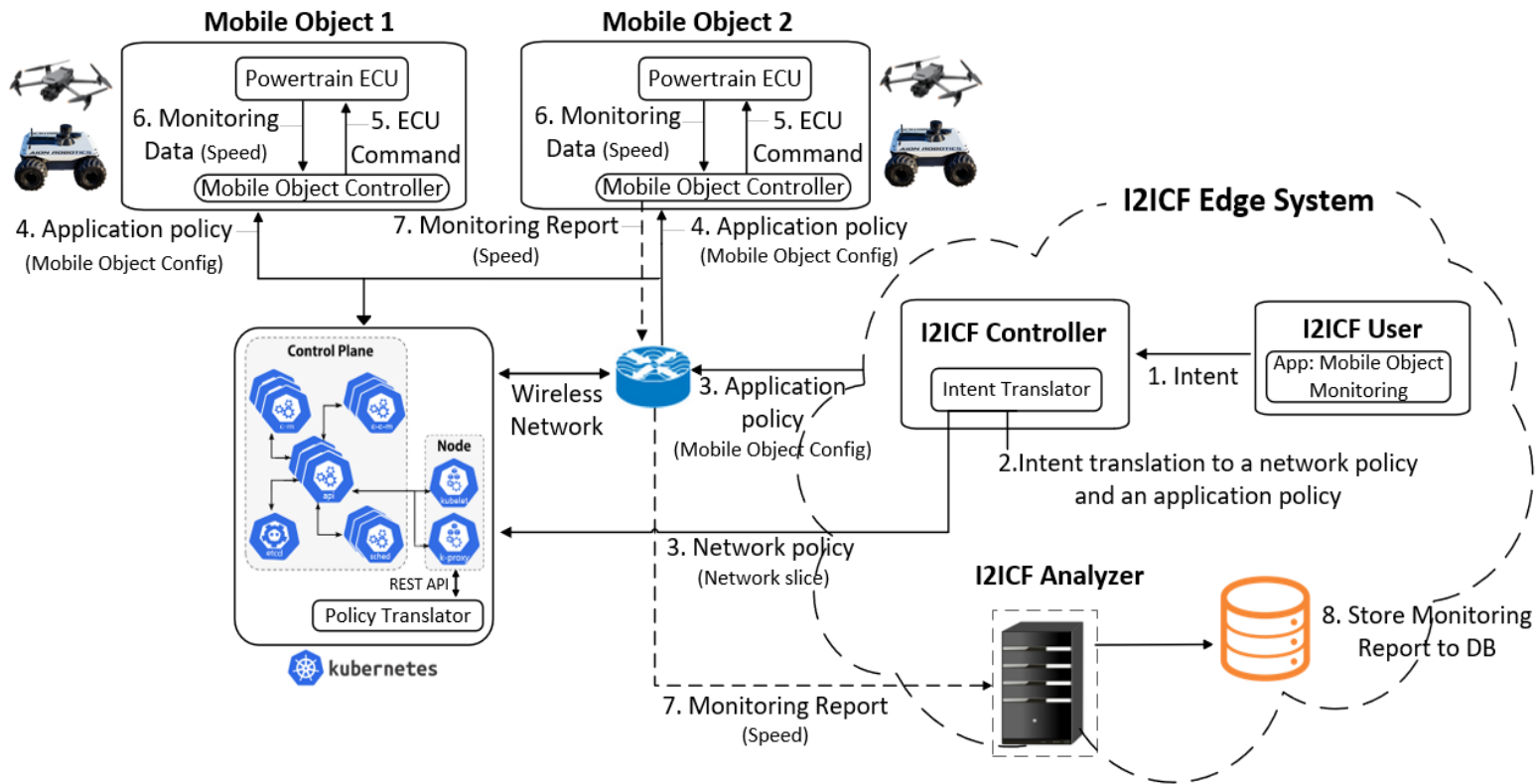
# Goal of Hackathon Project

- The goal is to showcase Feasibility of Intent Translation in Interfaces to In-Network Computing Functions (I2ICF) and its Framework.
  - **Intent Translation** and **Policy Provisioning**
    - Creation of a YAML Intent based on 3GPP 28.312 and its Deliverance to Mobile Objects.
- Internet Drafts for the I2ICF Project
  - <https://datatracker.ietf.org/doc/draft-ahn-opsawg-i2icf-cits/>
  - <https://datatracker.ietf.org/doc/draft-ywj-opsawg-i2icf-data-center-networking/>
  - <https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-framework/>
  - <https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-problem-statement/>

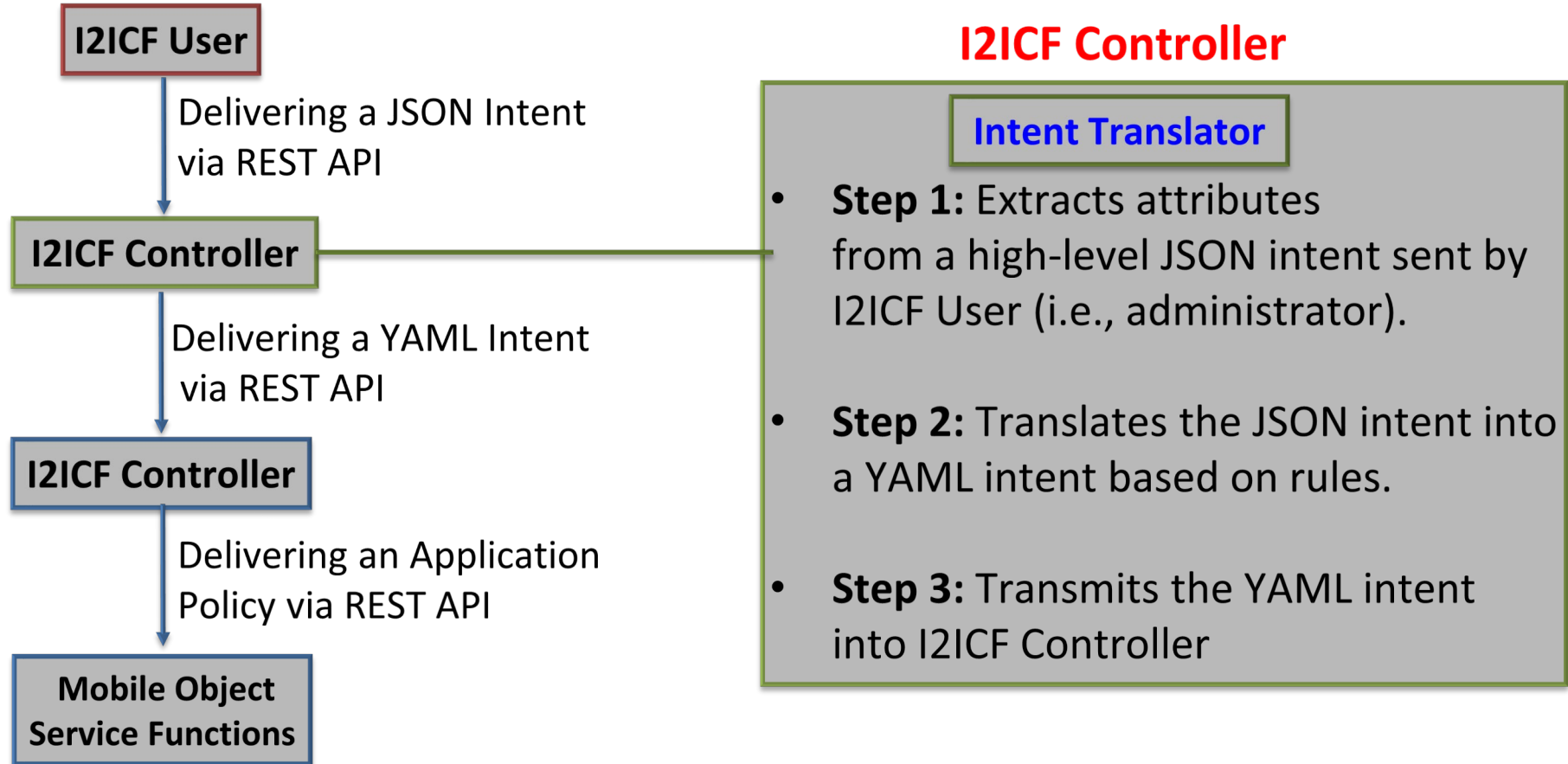
# I2ICF Technical Architecture

- Intent Translator processes high-level intents and converts them into YAML format.
- I2ICF Controller manages intent-based networking configurations.
- REST API facilitates communication between components.
- Kubernetes orchestrates I2ICF services for scalability and reliability.

# Interface to In-Network Computing Functions (I2ICF) for Mobile Objects



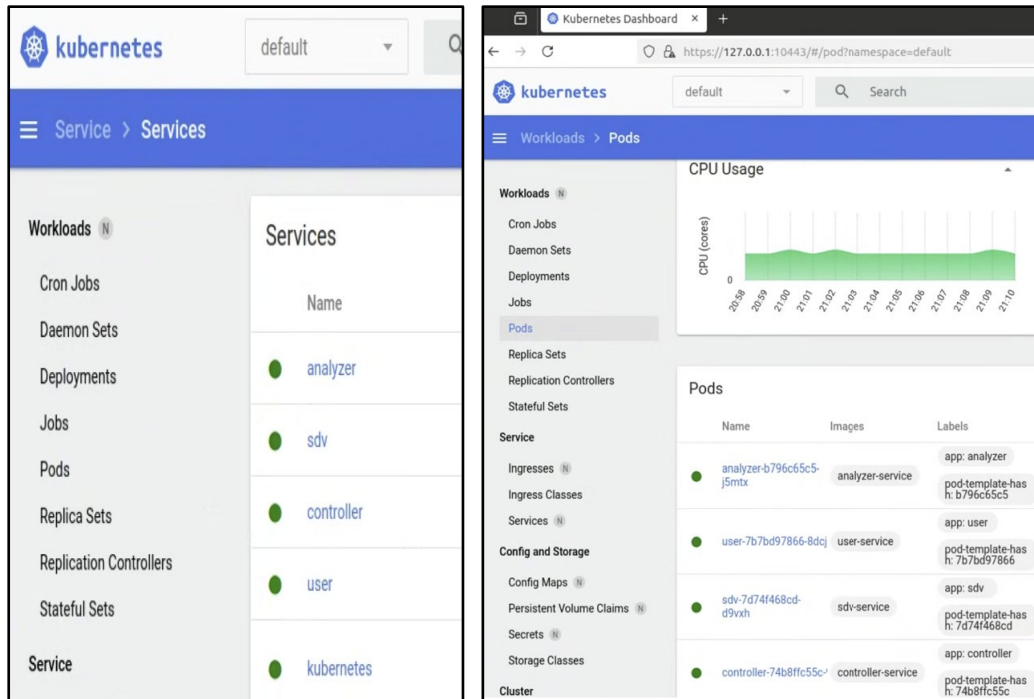
# Flow Diagram of Intent Translator



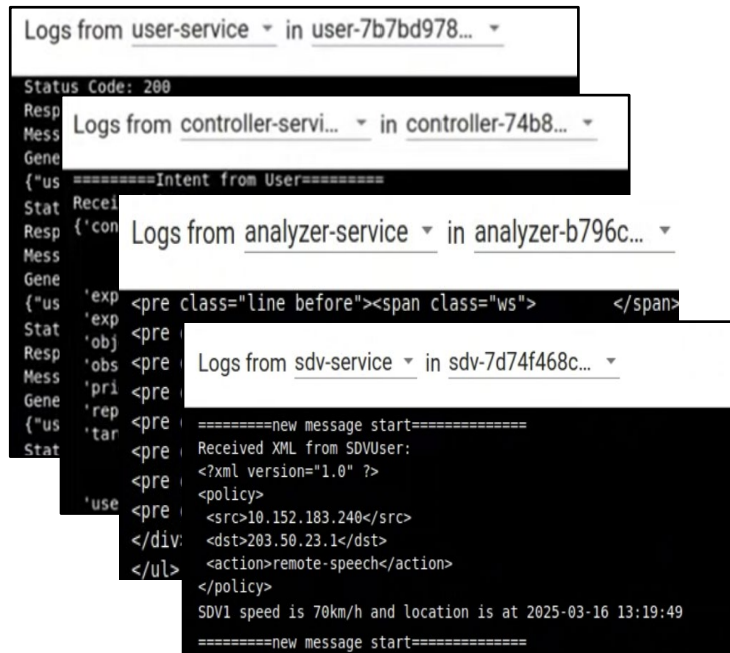
# What we learned

- We implemented **an Intent Translator for I2ICF Framework for Mobile Objects** in Wireless Networks.
- We demonstrated **Intent-Based Networking (IBN)** for the configuration and monitoring of **Mobile Objects** through the I2ICF Framework.

# Demonstration of an I2ICF Framework



I2ICF Services and Pods on Kubernetes



Logs of each I2ICF Service

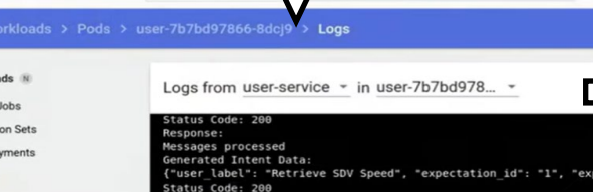


The screenshot displays the 'Intent-Translator' web application. The browser's address bar shows 'localhost:3000/registration'. The application's header includes the title 'Intent-Translator' and a 'Create-Intent' button. A modal window titled 'Creating Application Intent' is active, containing a form with the following fields and values:

- User Label (String, required): Retrieve SDV Speed
- Expectation ID (String, required): 1
- Expectation Verb (Enum: DELIVER, ENSURE, required): ENSURE
- Object Type (Enum, required): SDV
- Target SDV ID (String, required): SDV\_001
- Speed Value (String, optional): 50
- Priority (Integer, 1-99, required): 1
- Observation Period (Integer, in seconds, required): 60
- Report Reference (String, required): IntentReport\_002

A 'Submit' button is located at the bottom of the modal. The background of the entire image is a red and white checkered pattern.

## I2ICF User



The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, the name 'kubernetes', and search fields. The breadcrumb trail indicates the path: Workloads > Pods > user-7b7bd9786-8dcj9 > Logs. The left sidebar lists various Kubernetes resources, with 'Workloads' selected. The main content area displays the logs for the 'user-service' pod in the 'user-7b7bd9786-8dcj9' namespace. The logs show a successful GET request to the '/sdv-speed' endpoint, returning a 200 status and a response indicating the SDV speed is 100.

```

Status Code: 200
Response:
Messages processed
Generated Intent Data:
{"user_label": "Retrieve SDV Speed", "expectation_id": "1", "expectation..."
Status Code: 200
Response:
Messages processed
Generated Intent Data:
{"user_label": "Retrieve SDV Speed", "expectation_id": "1", "expectation..."
Status Code: 200
Response:
Messages processed
Generated Intent Data:
{"user_label": "Retrieve SDV Speed", "expectation_id": "1", "expectation..."
Status Code: 200
Response:
Messages processed
Generated Intent Data:
{"user_label": "Retrieve SDV Speed", "expectation_id": "1", "expectation..."
Status Code: 200
Response:
Messages processed

```

# I2ICF Controller

Workloads > Pods > controller-74b8ffc55c-989jn > Logs

Workloads N

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods**
- Replica Sets
- Replication Controllers
- Stateful Sets
- Service
- Ingresses N
- Ingress Classes

Logs from controller-servi... in controller-74b8...

```

=====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'}

===== SDV =====
Sent message to vehicle_1 at http://10.152.183.240:5000/receive_message
<?xml version="1.0" ?>
  
```

I2ICF Controller

## I2ICF Controller

## YAML Intent

# Open-Source Project for I2ICF

[URL] <https://github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-122>

The screenshot shows the GitHub web interface for the repository `jaehoonpauljeong / I2ICF`. The page is viewed on the `main` branch, specifically the `IETF-122` directory. The left sidebar displays the file tree, with `IETF-122` selected. The main content area shows the files and folders within `IETF-122`, including `Analyzer`, `Controller`, `SDV`, `User`, `readme.txt`, and `Side-Meeting`. The `readme.txt` file is selected, showing its content: "This is the IETF-122 Hackathon Project for IETF-122 Hackathon."

github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-122

Product Solutions Resources Open Source Enterprise Pricing

jaehoonpauljeong / I2ICF (Public)

Code Issues Pull requests Actions Projects Security Insights

Files

main

Go to file

- > BoF
- > IETF-120
- > IETF-121
- > IETF-122
  - > Analyzer
  - > Controller
  - > SDV
  - > User
  - readme.txt
- > Side-Meeting

README.md

Readme

I2ICF / IETF-122

ahnjs124 Add files via upload

Name	Last commit message
..	
Analyzer	Add files via upload
Controller	Add files via upload
SDV	Add files via upload
User	Add files via upload
readme.txt	Create readme.txt

readme.txt

This is the IETF-122 Hackathon Project for IETF-122 Hackathon.

[URL] <https://www.youtube.com/watch?v=hZgnWlQzXbE>

[URL] <https://www.youtube.com/watch?v=hZgnWlQzXbE>



# Next Steps

- We explored the implementation and design of **an Intent Translator** for the I2ICF Framework within **IBN-Based System** in wireless networks.
- In IETF 123, we will develop **an Intent Translator** that advances from a Rule-based scheme to an AI-based scheme (e.g., Large Language Model: LLM) on Kubernetes Container Orchestration System.
- Also, we will design **YANG Data Models for the Main I2ICF Interfaces**.
  - Refer to <https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-framework/>

# I2ICF Hackathon Team

- **Professors:**

- Jaehoon (Paul) Jeong (SKKU)
- Yiwen (Chris) Shen (AJU)

- **Researchers:**

- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)
- Byoungman Robert An (KETI)

- **Students:**

- Yoseop Ahn (SKKU), Xudong Wang (SKKU), Mugabarigira Bien Aime (SKKU), Mose Gu (SKKU), Jiwon Suh (SKKU), Juwon Hong (SKKU), Nobuo Aoki (SOKENDAI)

## Hackathon Team Photo

