IoT Drone GCS and Digital Twin as oneM2M Applications

Cheol-Min KIM (cmkim@keti.re.kr)
Korea Electronics Technology Institute
2023.12.08

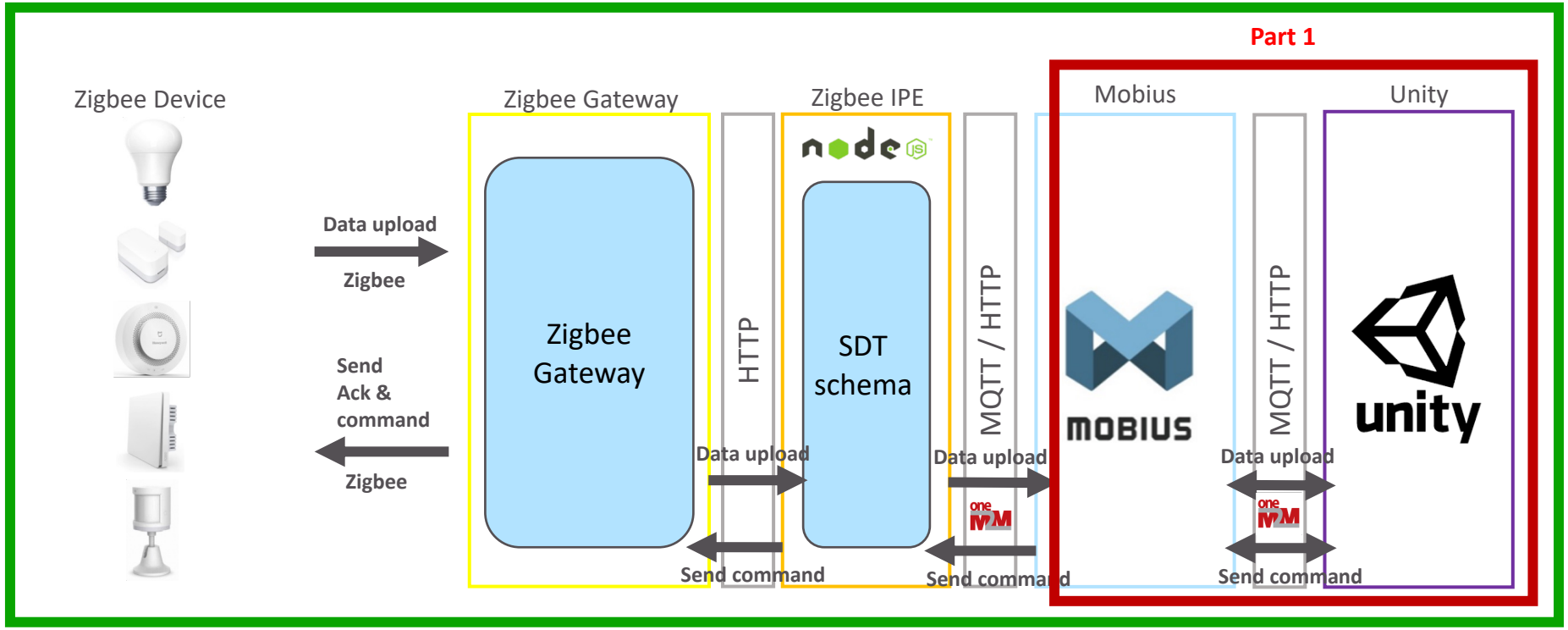
Outline

- oneM2M tutorial for digital twin
- oneM2M tutorial for IoT drone GCS
- R&D project on the GCS for unmanned vehicles

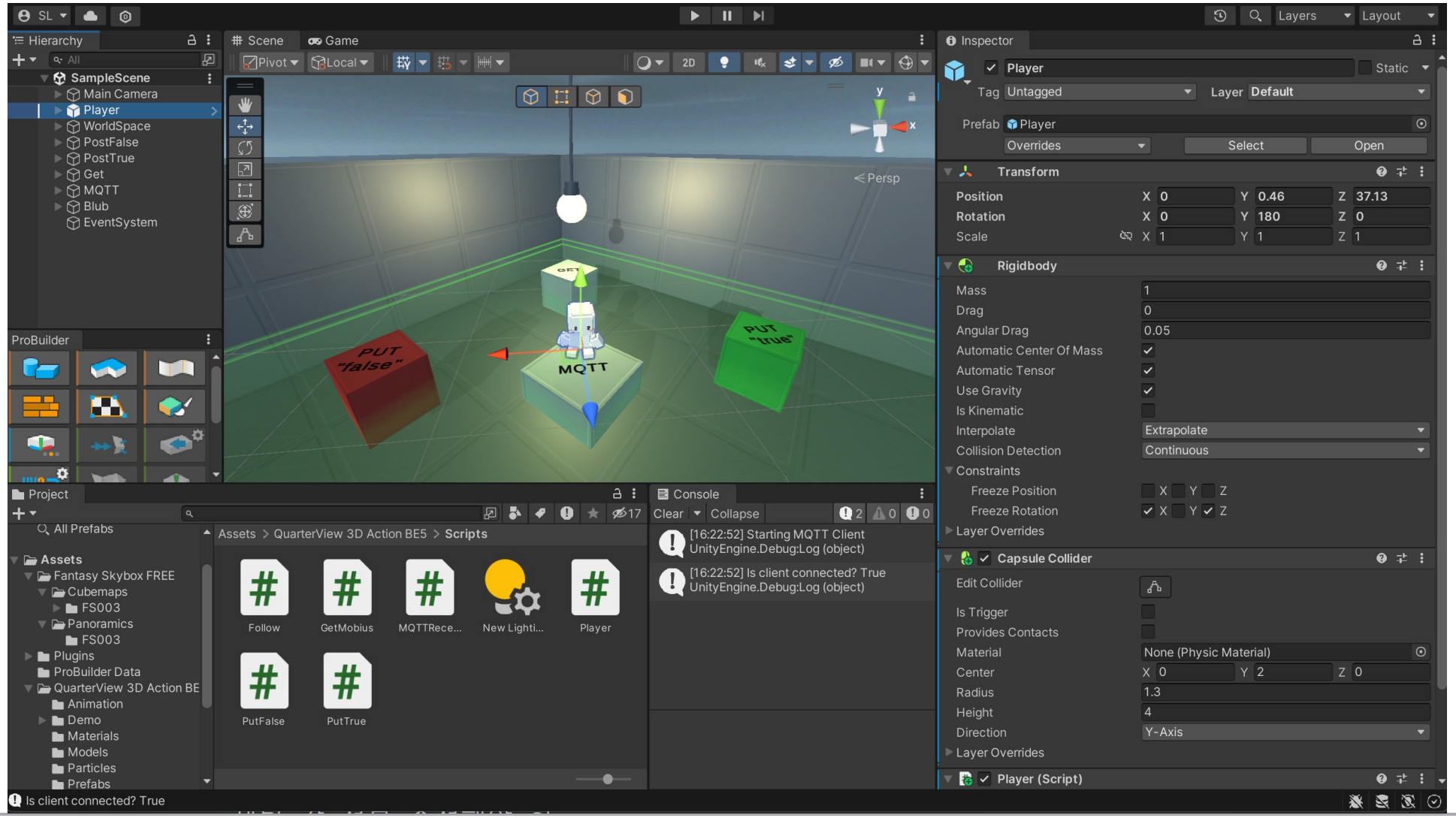
oneM2M tutorial for digital twin

Overview

- Part 1: Simple Unity 3D app (digital twin app) with virtual devices
- Part 2: Extend part 1 to interwork with physical device (Zigbee device to virtual device E2E interworking) via oneM2M platform

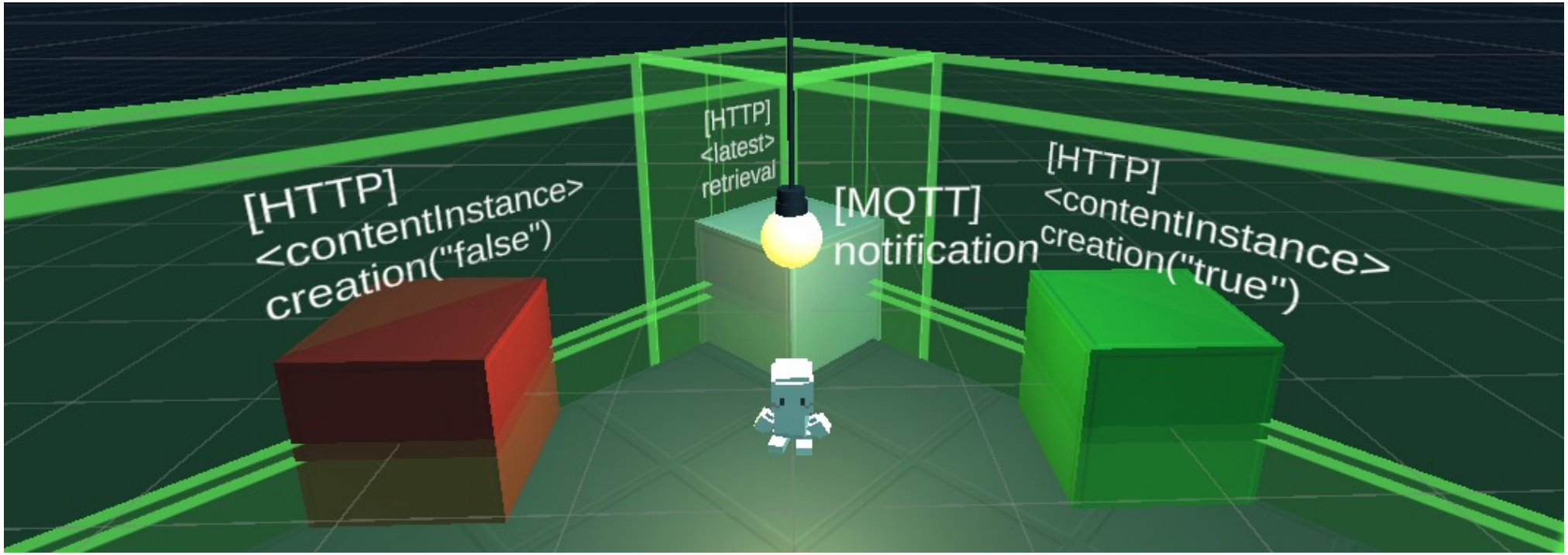


Glance of digital twin for oneM2M



Part1: Sample app for oneM2M

- oneM2M HTTP binding => <contentInstance> resource creation
- oneM2M MQTT binding => notification handling

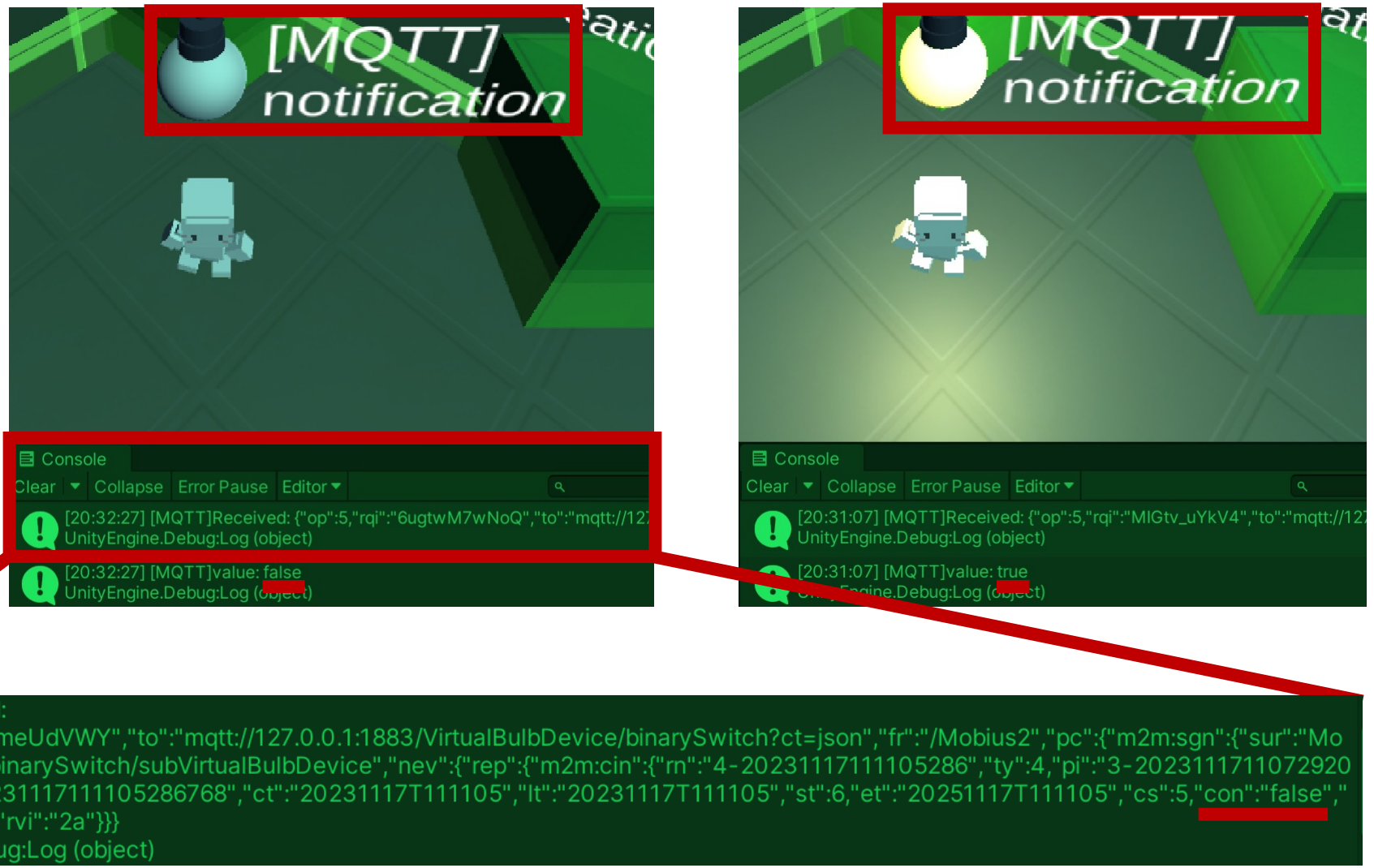


Part1: oneM2M HTTP binding

- When a character collides to the box, Unity app creates <contentInstance> ("**false**") into Mobius

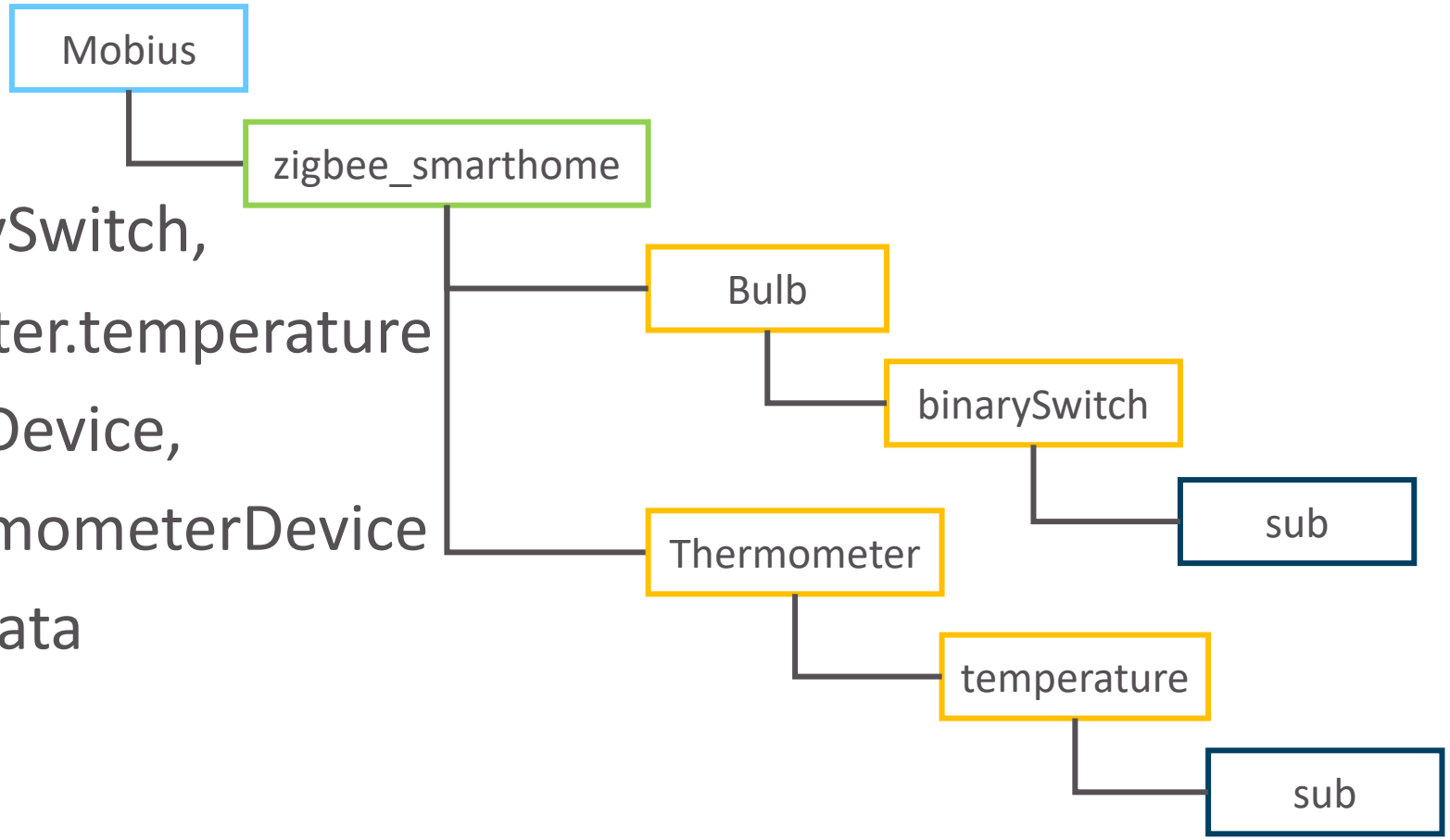
Part1: oneM2M MQTT binding

- Mobius publishes MQTT notification messages to Unity app with on/off status of the light bulb
- Unity app extracts the status from **"con"** field in the message



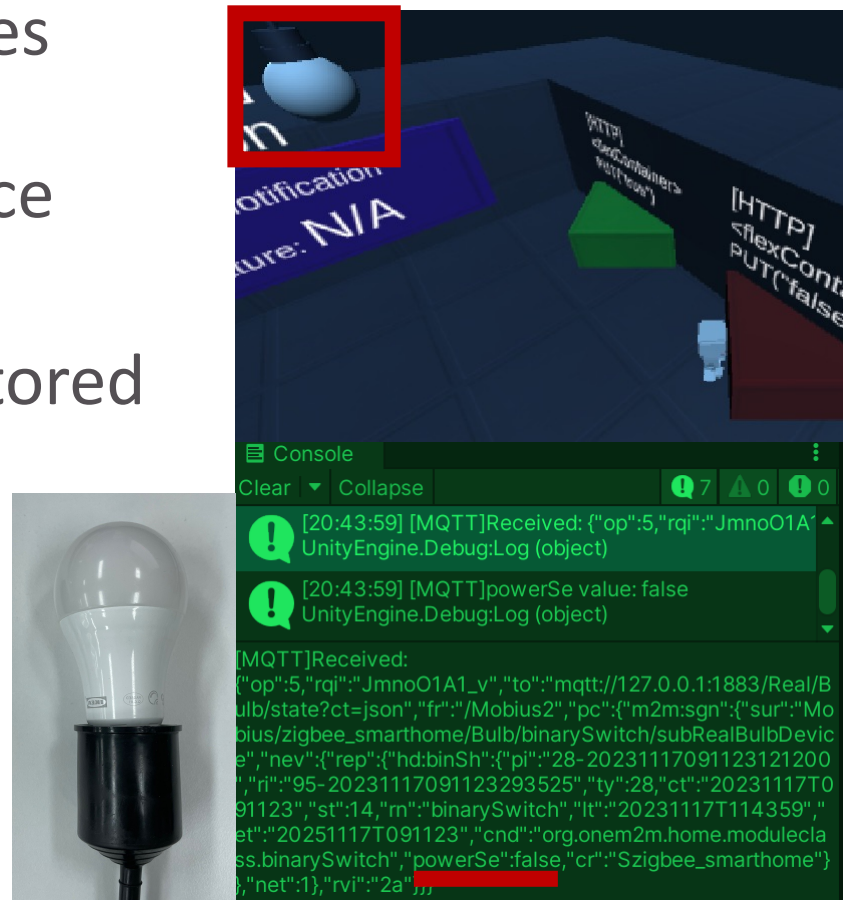
Resource tree design

- CSEBase => Mobius
- AE => zigbee_smarthome
- flexContainer => Bulb.binarySwitch,
  Thermometer.temperature
- Subcription => subRealBulbDevice,
  subRealThermometerDevice
- ContentInstentce => State Data



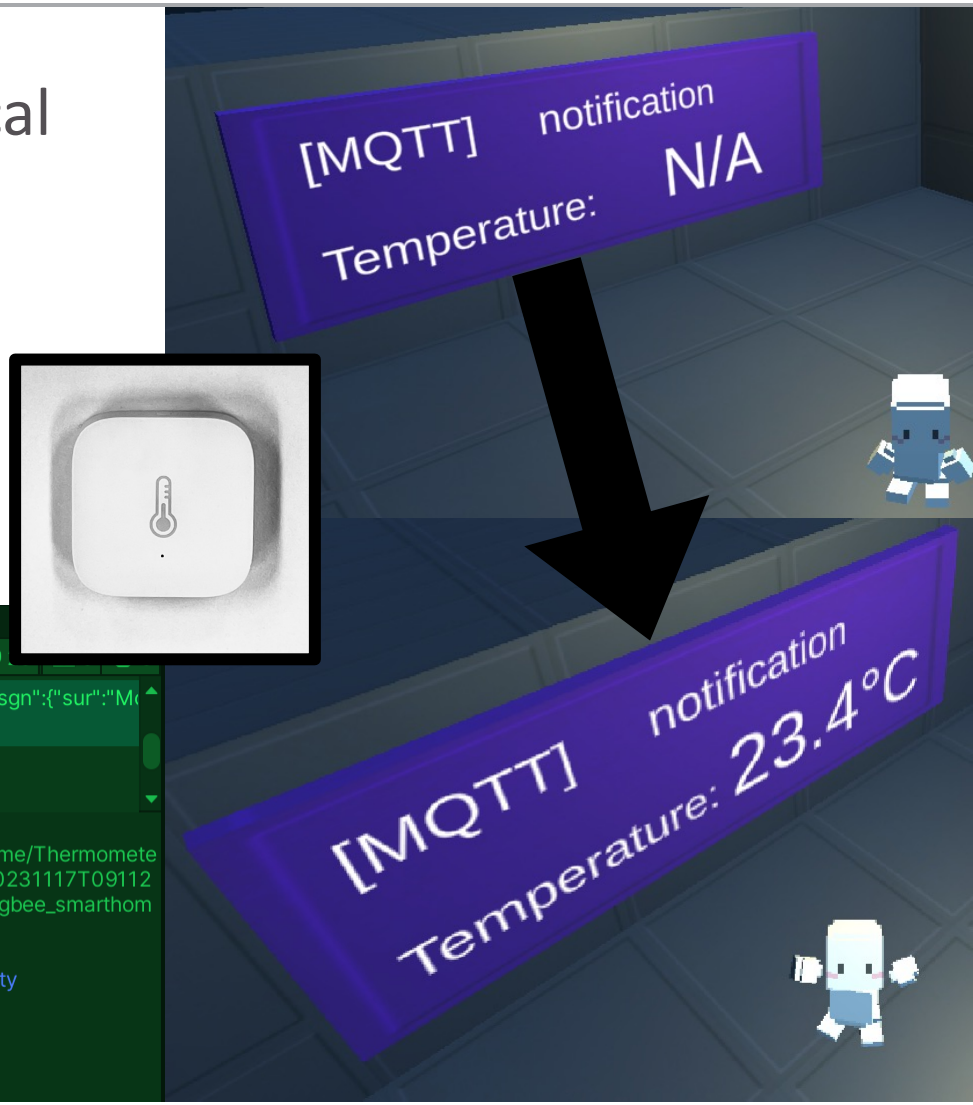
Part2: virtual to physical world

- When the character collides a box, Unity app changes attributes on the <flexContainer> resource for physical light bulb
- Interworking logs are stored at the logs panel, along with the game screen



Part2: physical to virtual world

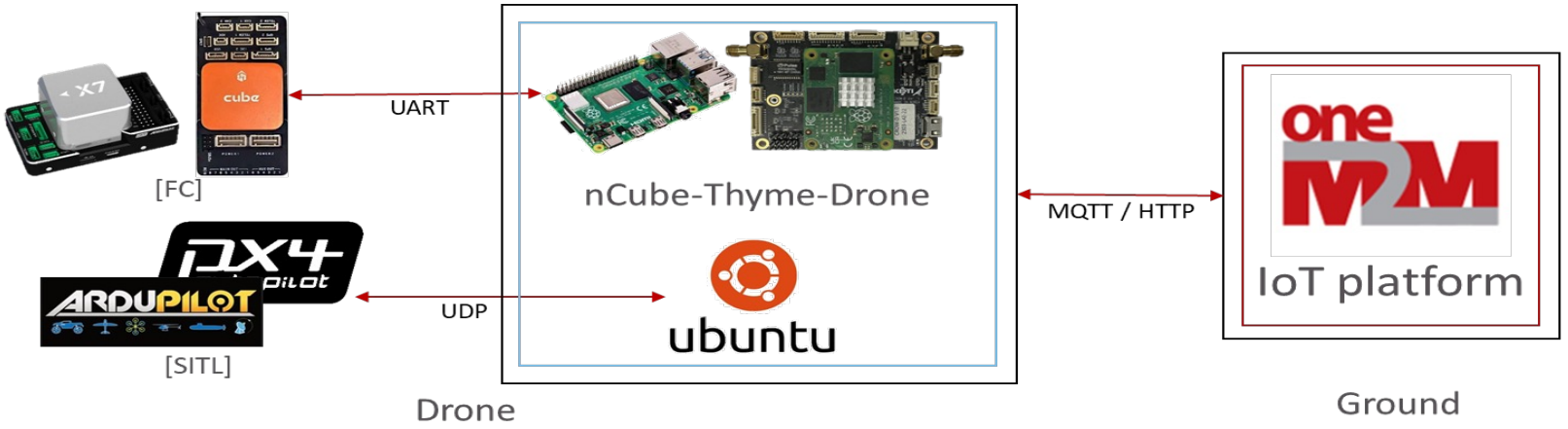
- When the temperature value is updates of a physical sensor, mobius publishes notification message via MQTT
- Unity app listens notification message via MQTT topic and updates the status of virtual world



oneM2M tutorial for IoT drone GCS

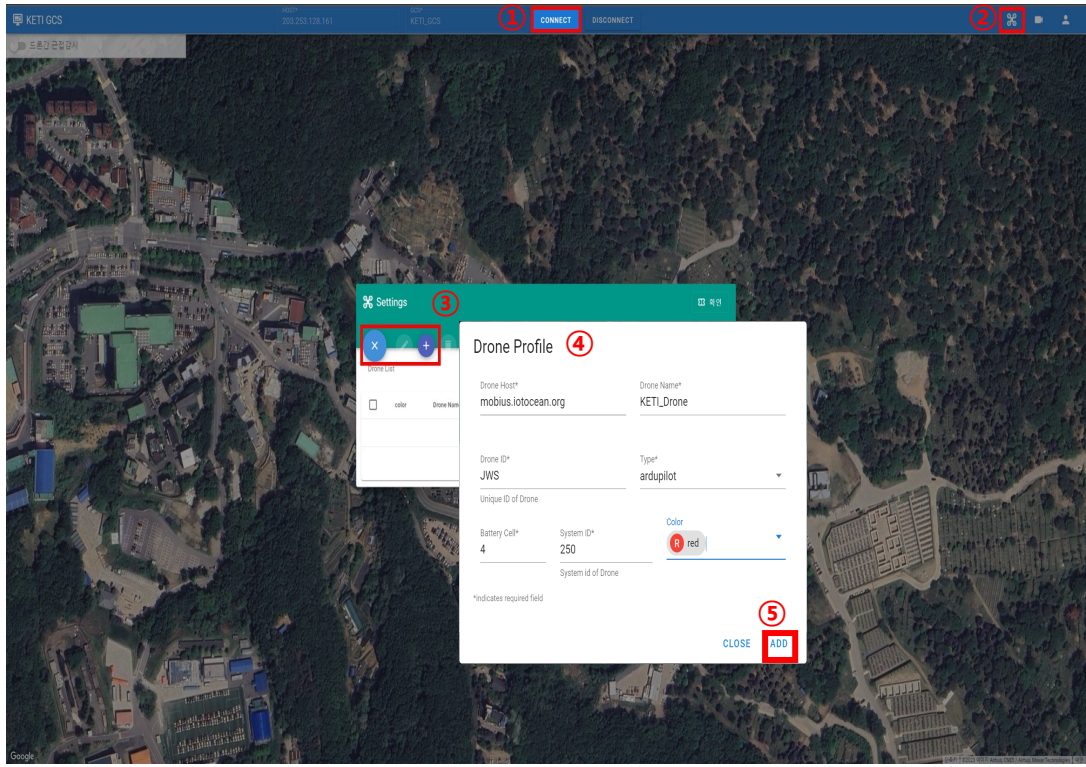
Overview of IoT drone

- Flight Computer (FC) and System In The Loop (SITL) that utilize the MAVLink protocol can be integrated as IoT devices in a oneM2M system
- nCube-Thyme-Drone (IoT Drone IPE) operates as an IoT device application on onboard computers or within an Ubuntu environment
- Due to the real-time nature of drones, MQTT communication is used for interfacing with the ground
- For the mission equipment of drone or backing up MAVLink data, HTTP protocol is used



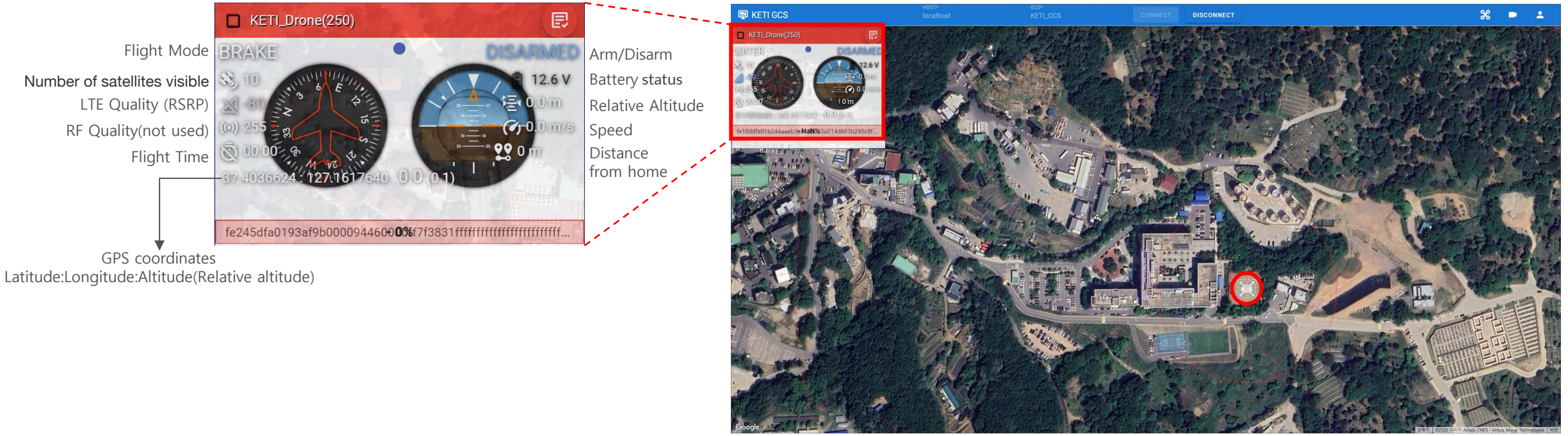
GCS web app as oneM2M AE

- GCS is a software program that monitors real-time flight data of unmanned aerial vehicles (UAVs) based on MAVLink protocol, and enables on-site or remote control
- That is a single program capable of real-time monitoring and control of multiple UAVs simultaneously
- Setups for the IoT drone GCS
  - 1. Connect to Server (Mobius)
  - 2. Manage Drone
  - 3. Register Drone
  - 4. Change Drone info
  - 5. Click ADD to complete the settings



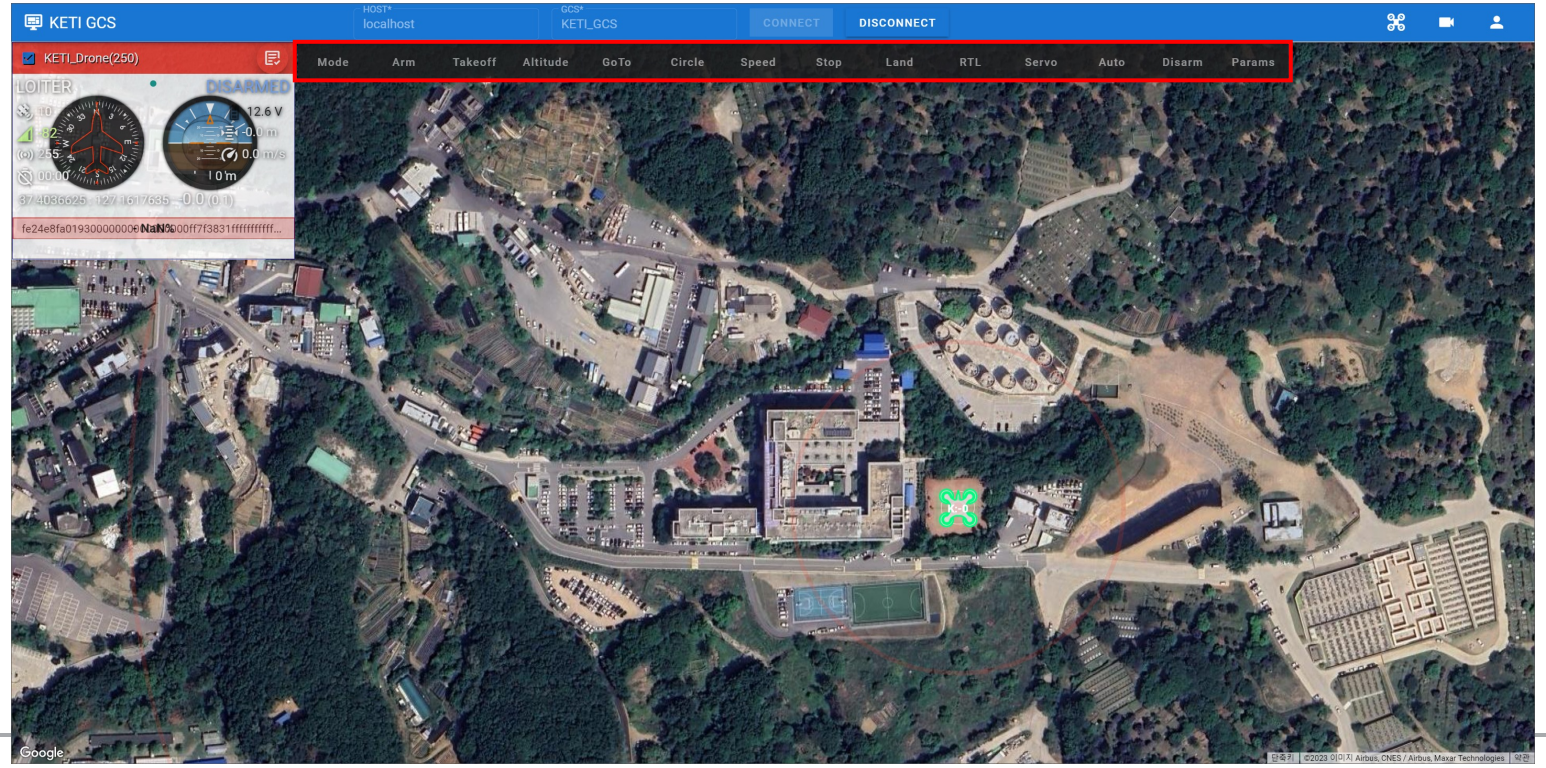
GCS web app

- Monitoring
  - The HUD window of GCS shows the status of the drone, including its flight mode, arm/disarm status, location, and distance from home



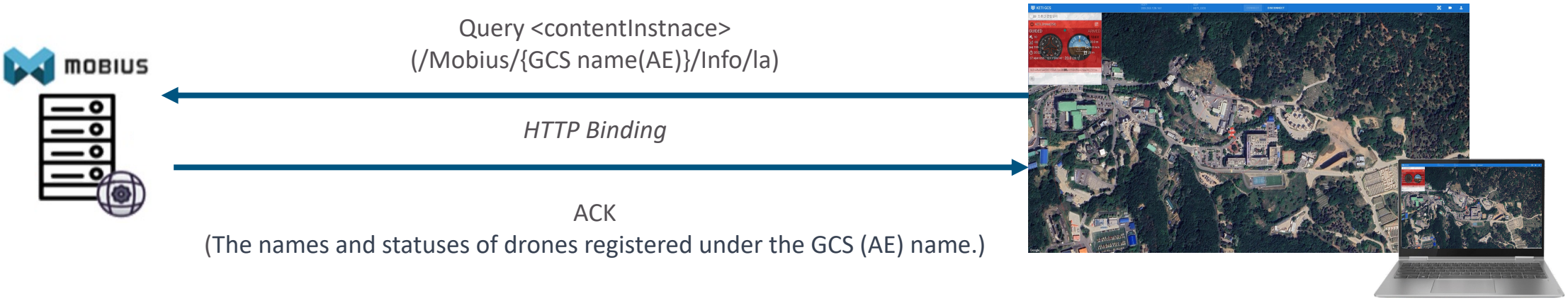
GCS web app

- Control
  - When you select a drone to control, you can control the drone through commands such as changing mode, arm, takeoff, go to, and circle in the command tab that appears



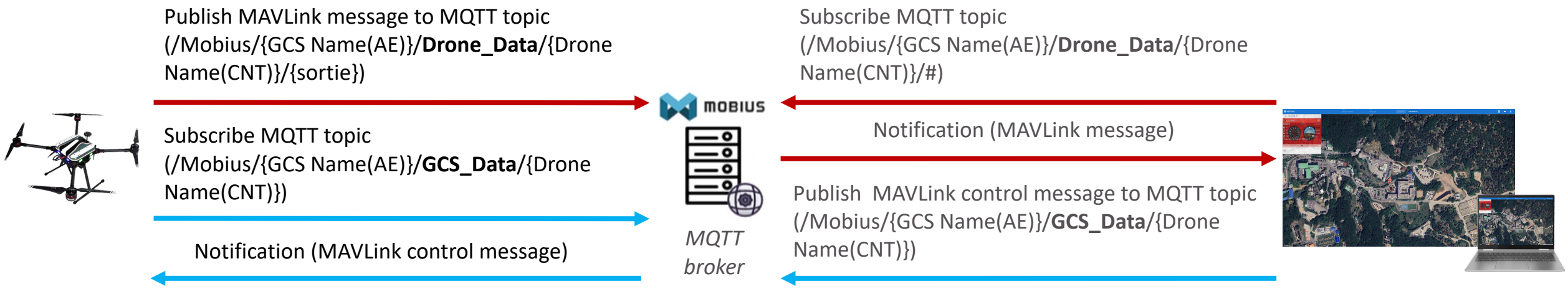
Call flows of IoT drone GCS

- Get drone list
  - GCS retrieves a list of drones and their status from mobius using the last <contentInstance> value within the 'Info' container through HTTP binding



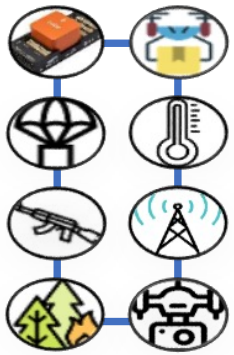
Call flows of IoT drone GCS

- Real-time drone monitoring and control with GCS
  - GCS subscribes to the 'Drone_Data' topic via MQTT for real-time drone monitoring
  - GCS publishes control messages generated within itself to the 'GCS_Data' topic for control purposes



Call flows of IoT Drone GCS

- Mission data integration
  - GCS creates a <subscription> resource to the mission data URL via HTTP binding
  - GCS receives notifications with the 'nu' attribute value within the subscription
    - whenever mission data is generated or not, this allows display the mission data in GCS

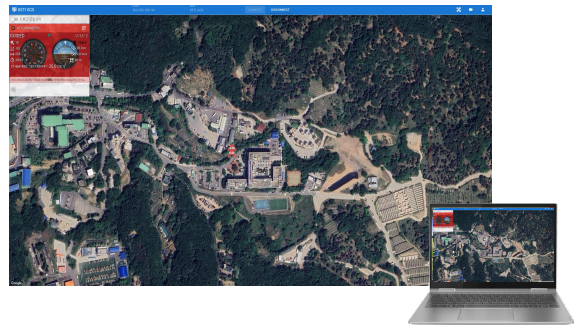


create <contentInstance> resource (/Mobius/{GCS Name(AE)}/**Mission_Data**/{Drone Name(CNT)}/{Mission Name(CNT)}/{Mission Data Name{CNT)})

HTTP Binding

create <subscription> resource (/Mobius/{GCS Name(AE)}/**Mission_Data**/{Drone Name(CNT)}/{Mission Name(CNT)}/{Mission Data Name{CNT)}/sub_{Drone_Name}_of_GCS)

Notification (mission data)



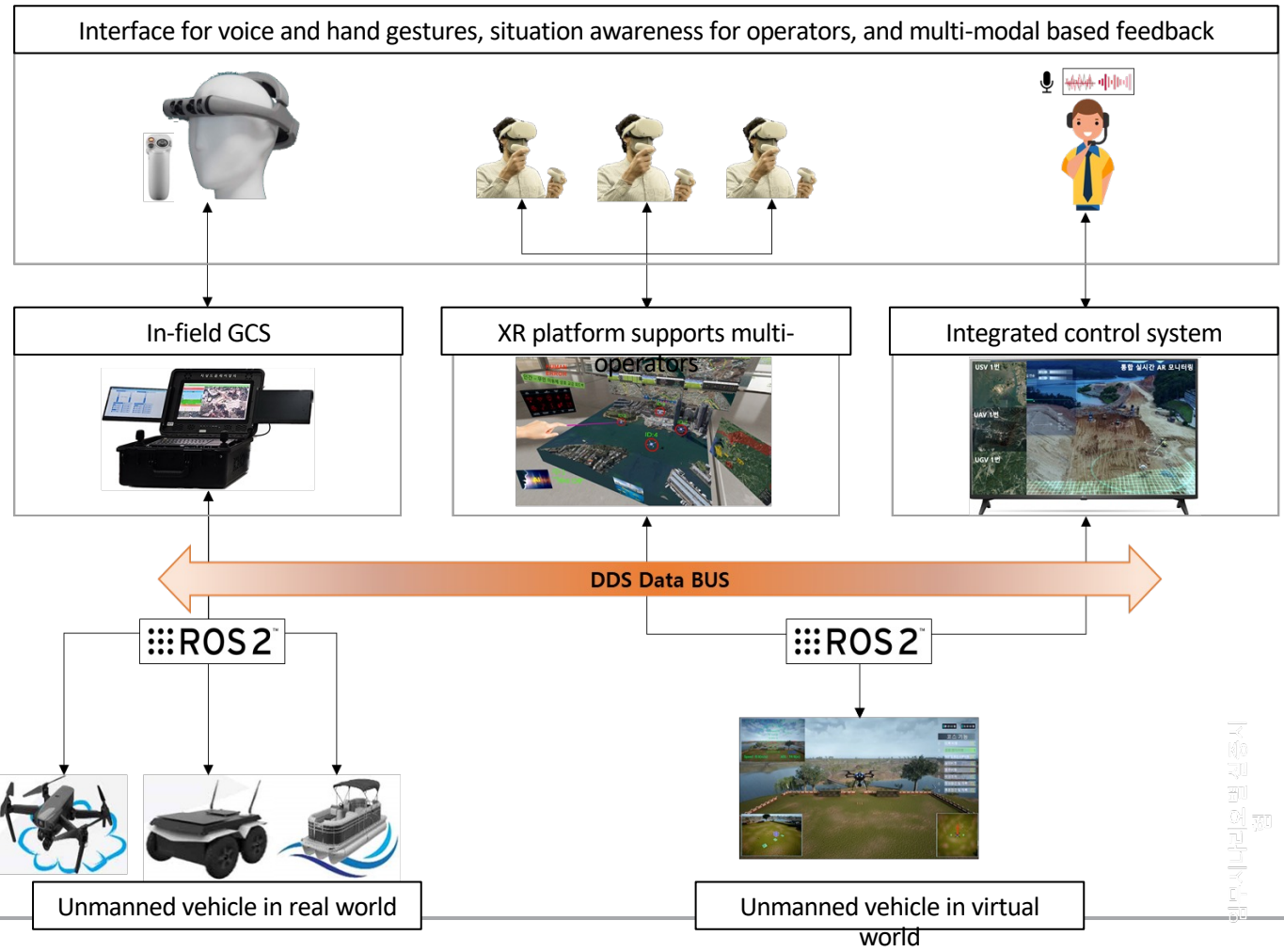
R&D project on the GCS for unmanned vehicles (UAV/UGV/USV)

High-level system architecture

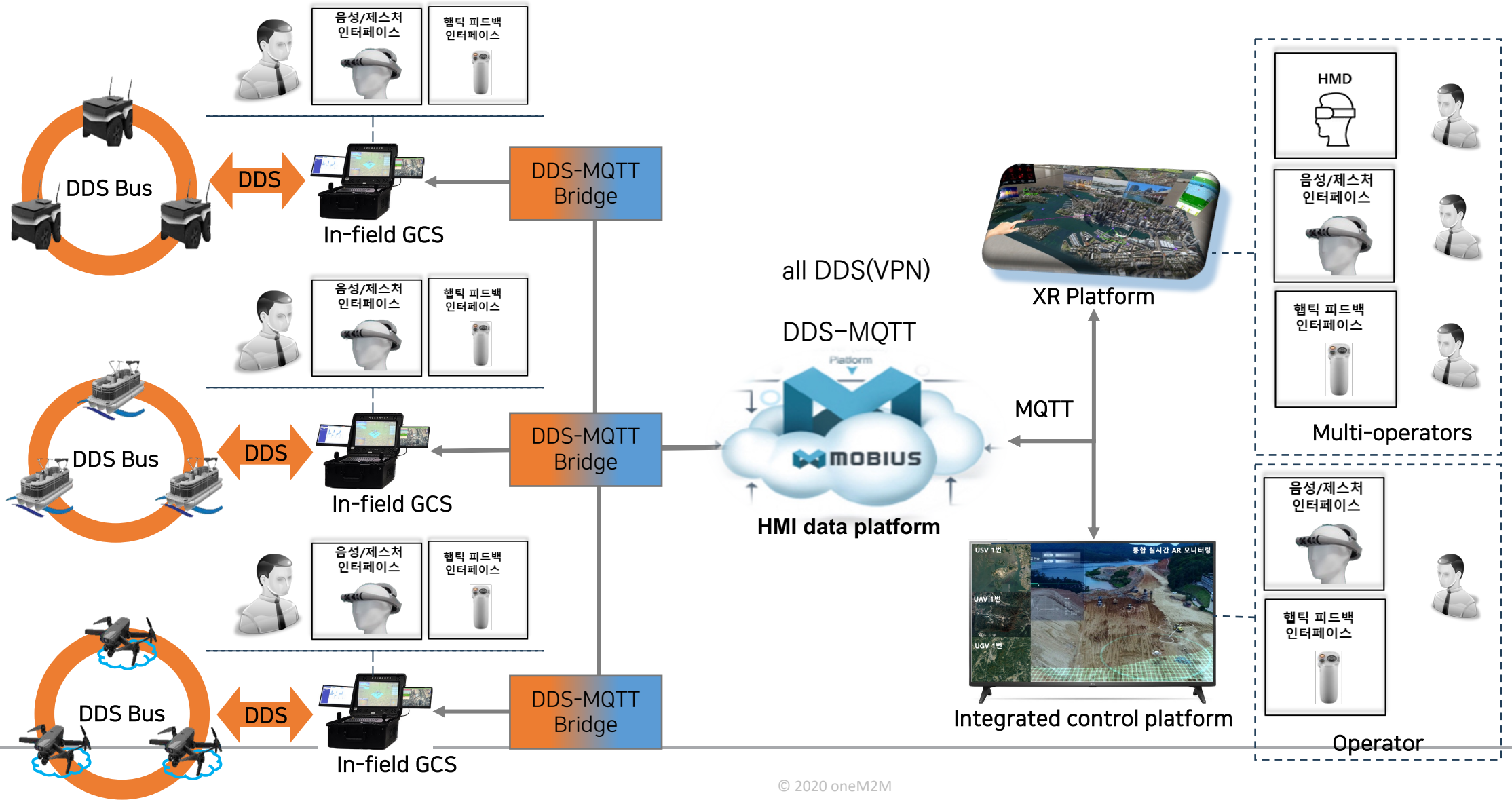
- oneM2M related topics
  - ROS interworking
  - XR (metaverse)
  - Simulator (digital twin)

Human-Mobility Interface

- ROS2-based SW framework
- Voice & hand gesture recognition (camera, IR, radar)
- Situation awareness of operator
- Feedback interface
- XR-based integrated control platform
- In-field GCS
- Real-world deployment testing



Use of oneM2M in the system



Thank you

Korea Electronics Technology Institute