GEdge(Griffin-Edge) Platform
- 초저지연 지능형 클라우드 엣지 SW 플랫폼 -

다중 클라우드 엣지간 데이터 고속처리를 위한 GS-MQ 기술

2021.12.09

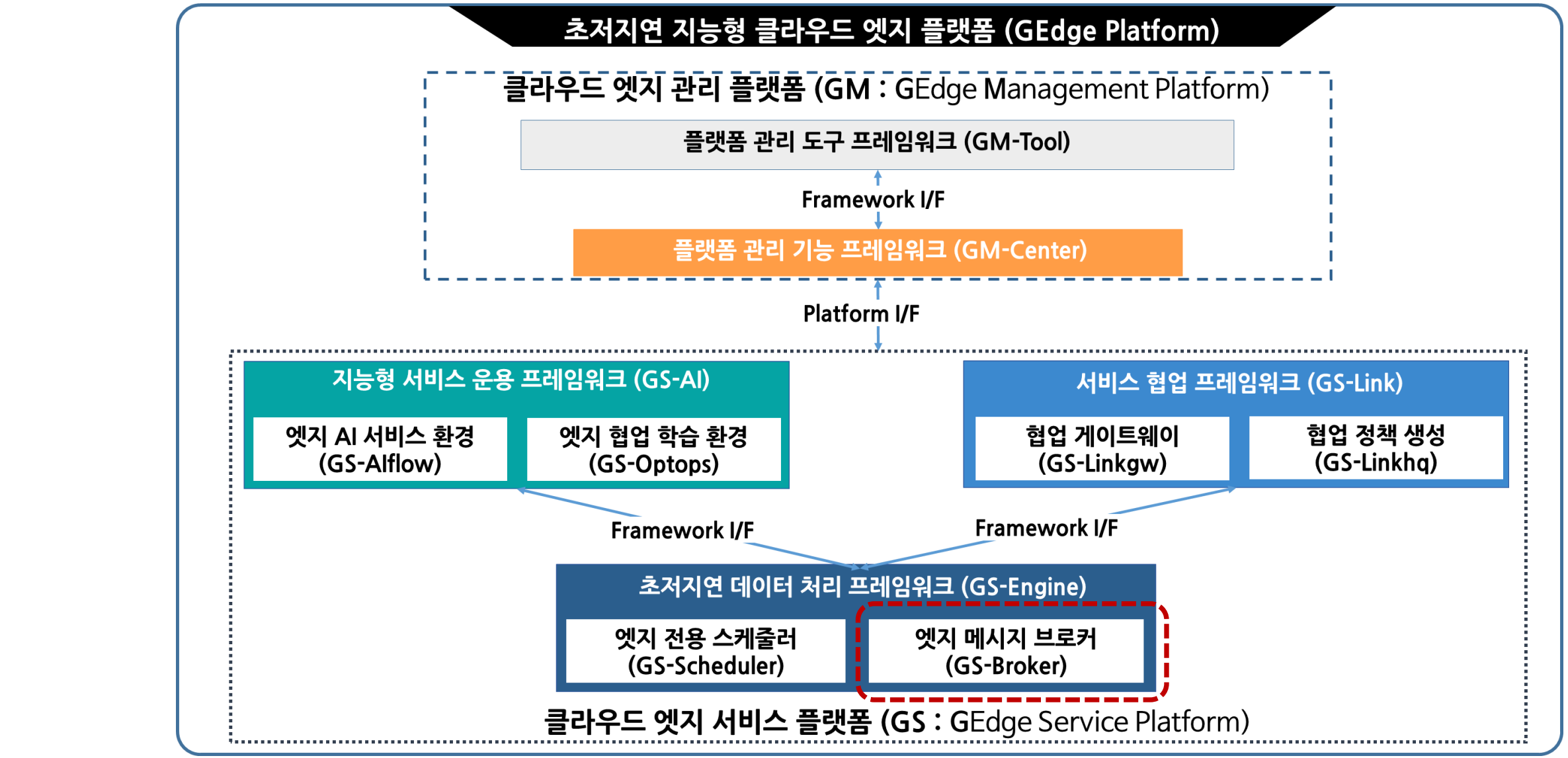
GS-Engine 프레임워크 코어 개발자 (GS-Broker)
김현우(hwkim@keti.re.kr)

"GEdge Platform" 은 클라우드 중심의 엣지 컴퓨팅 플랫폼을 제공하기 위한 핵심 SW 기술 개발 커뮤니티 및 개발 결과물의 코드명입니다.

- Developer-Friendly

GEdge Platform Community 3rd Conference (GEdge Platform v2.0 Release) -

이번 발표의 기술적 포지셔닝



Contents

Ⅰ GS-Broker 개요

Ⅱ GS-Broker의 필요성

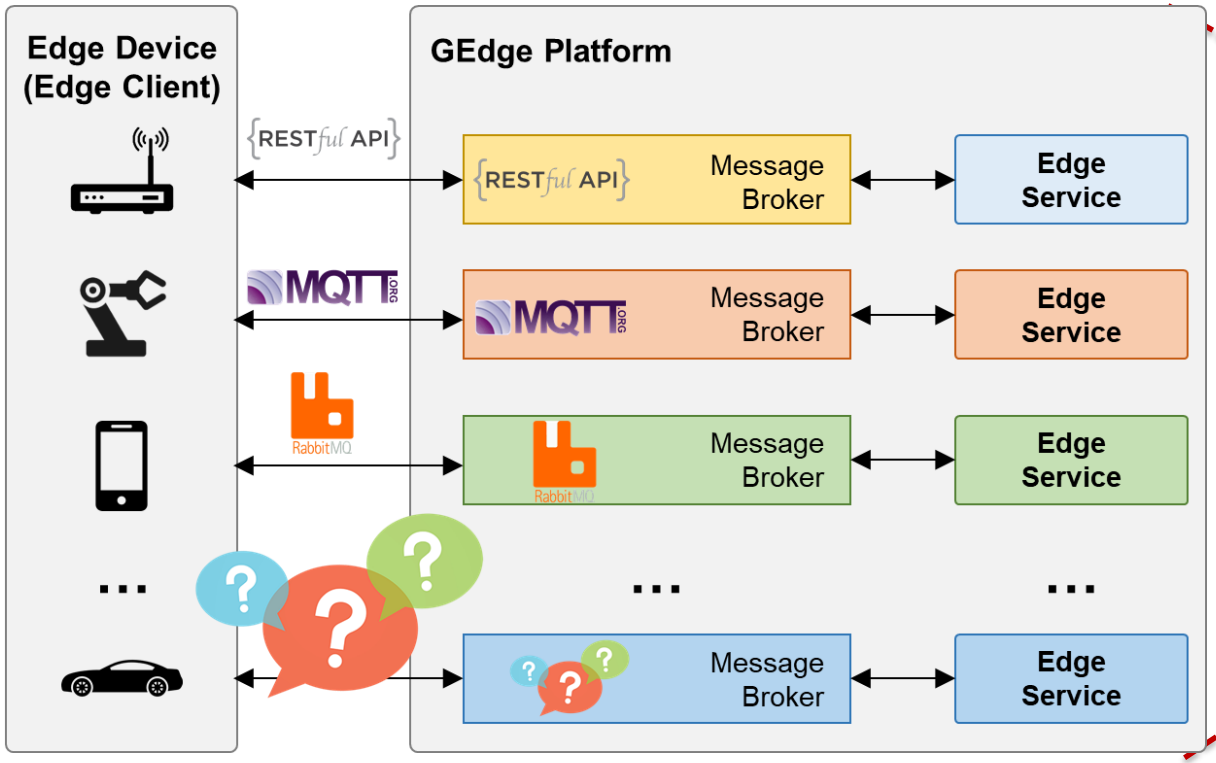
Ⅲ GS-Broker/GS-MQ 구조

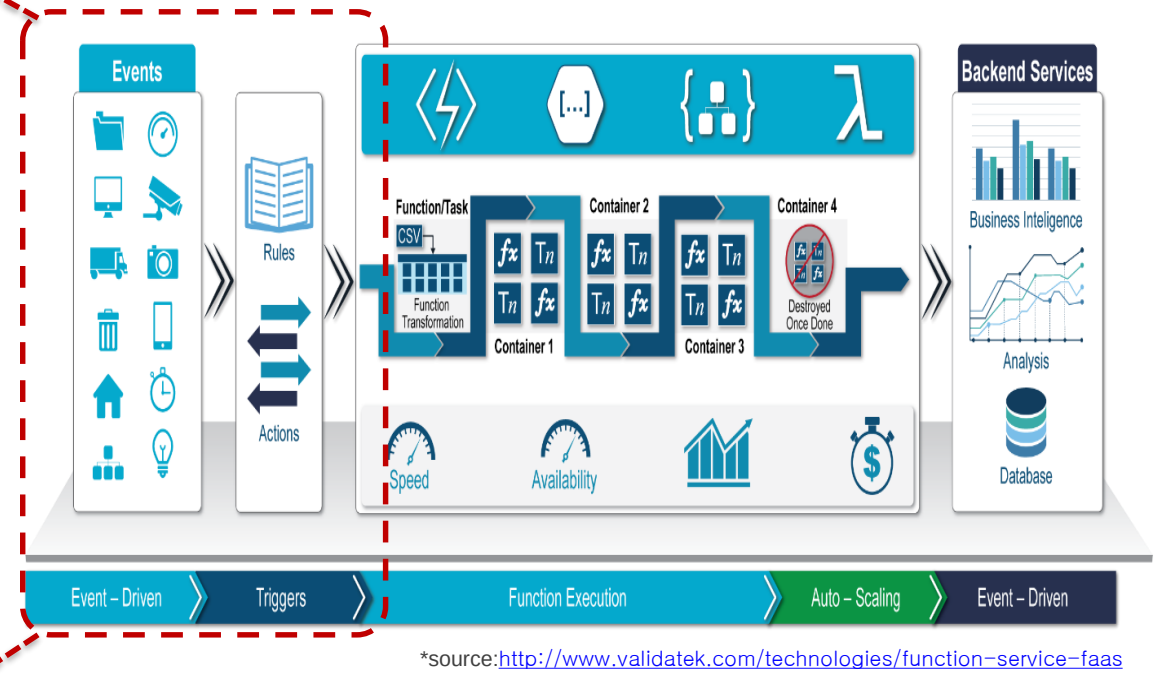
Ⅳ GS-Broker/GS-MQ 22년도 계획

GS-Broker 개요

1 GS-Broker 개요

- **GEdge Service Platform은 Edge디바이스로부터 수집되는 다양한 데이터를 지원하는 컴퓨팅 환경**
  - 엣지 디바이스별 상이한 메시지 프로코톨은 서비스 플랫폼의 복잡성을 증가 ➔ **(GS-Broker)**
  - 신규 엣지 디바이스 or 메시지 프로토콜 추가시 유연한 확장의 어려움(MS : 5종 지원, AWS : 4종 지원) ➔ **(GS-Broker : 8종 지원)**





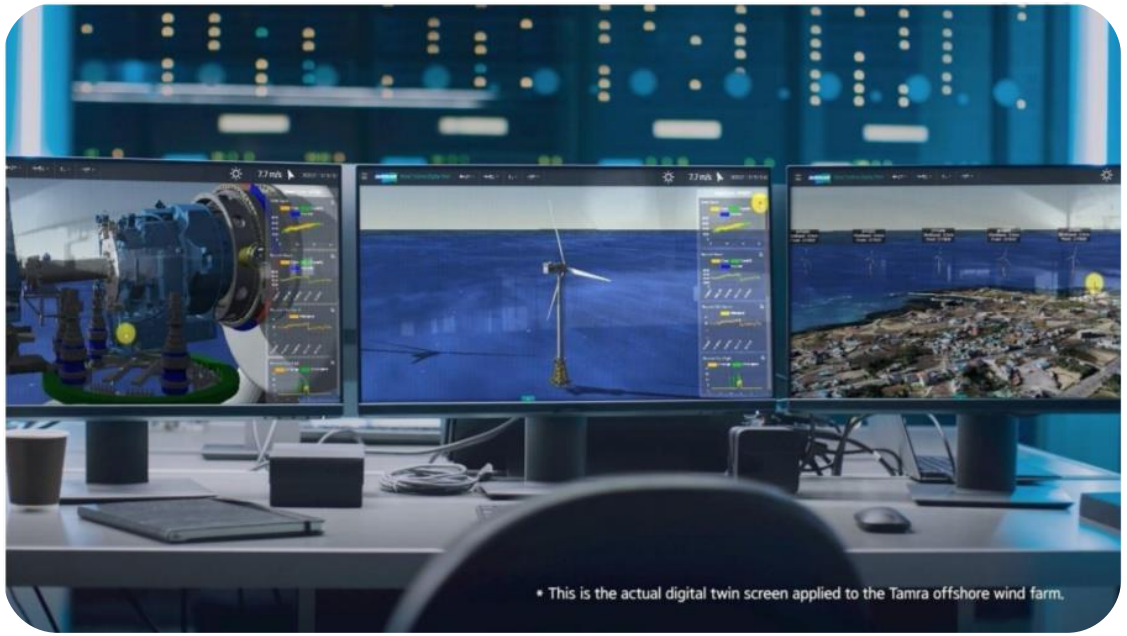
*source:http://www.validatek.com/technologies/function-service-faas

<플랫폼의 유연한 확장이 어려운 메시지 브로커 구조>

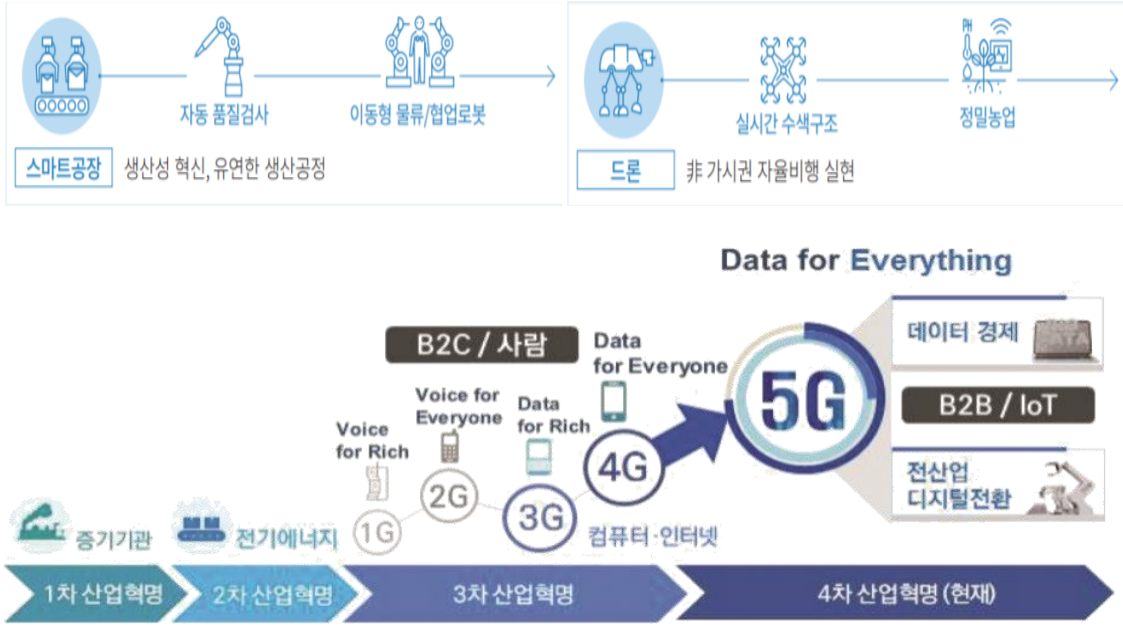
II GS-Broker의 필요성

1 GS-Broker 의 필요성

- **디지털 트윈 / 메타버스 등 데이터 중심 서비스에 초고속/저지연 대용량 데이터와 사물이 대량으로 연결**
  - Massive IoT 센서와 기기간 협업(효율극대화)과 같은 초연결 서비스가 증가
  - 기존 Event 발생 Regacy 플랫폼도 함께 활용가능한 아키텍처 필요성 증가



*source: KISA, Report 2021. Vol2.



*source: 관계부처합동, 혁신성장을 위한 5G+ 전략, 2019

1 GS-Broker 의 필요성

- **센서 디바이스의 소형화/고도화 및 5G Enabling Biz의 확산에 따른 산업현장 수요 폭증**
  - 5G 상용화를 위한 1차 표준화(Release15-`18.6) 완료 및 다양한 융합 서비스를 지원하는 2차 표준화(Release16) 진행(~`20.3)
  - 산업 현장의 기술적 특성에 따른 엣지 디바이스 추가가 요구됨.

5G N/W

eMBB[1)]

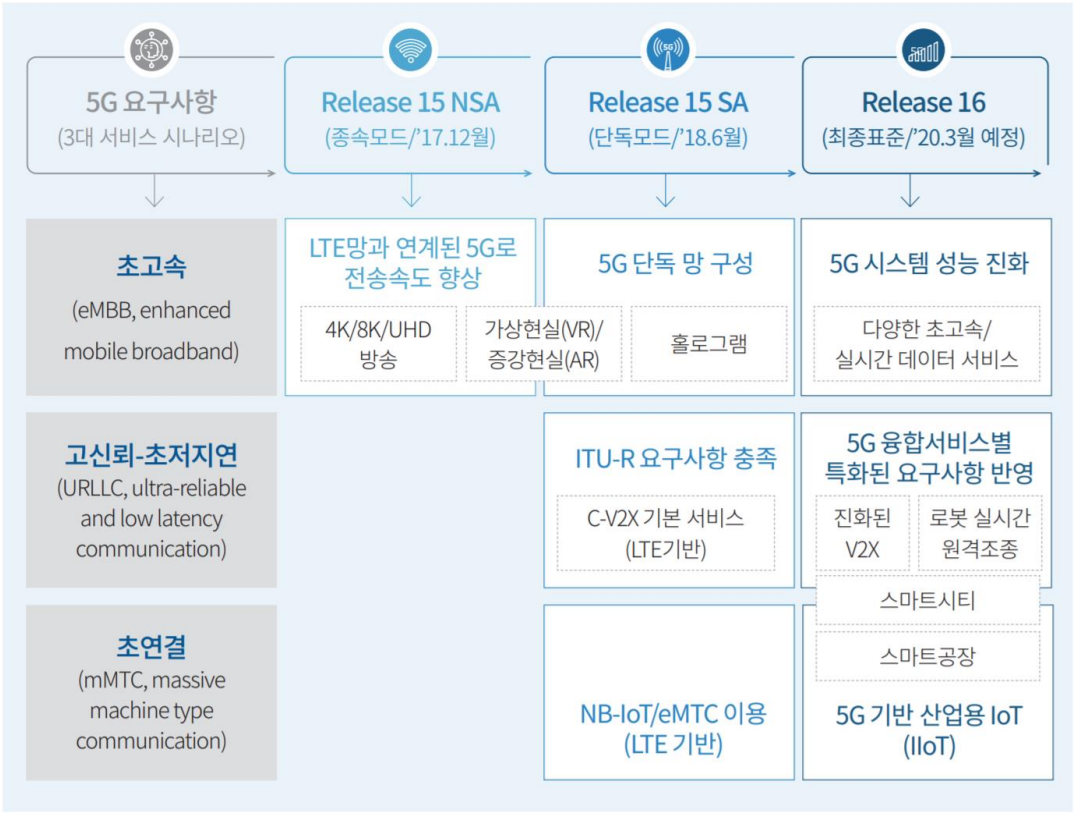
mMTC[2)]

uRLLC[3)]

5G Use-Case	5G 적용 가능 Biz.	Latency	Reliability	Coverage	Security
Time Critical process optimization	• Collaborative Robot • Wearables Adoption(3D AR[4)]) • 3D Scanning	Ultra-Low	Ultra-High	Indoor	Critical
Non Time Critical optimization	• Assets/Object 인식/Tracking • 대량의 Near Real-time Data 수집 • 생산 Simulation/Forecast	Less Critical	High	Indoor, On-site Outdoor	Critical
Remote maintenance and optimizing	• Remote Quality Inspection • Remote Diagnostics • Remote Virtual Back-Office	Less Critical	High	Wide Area	Critical
Seamless intra-/inter-enterprise communication	• Identification & Tracking of Goods • Reliable & Secure interconnection • Simulation & Design Data Exchange	Low	High	Wide Area, On-site Outdoor	Critical
Connected Goods	• Product Life-cycle Management • New Products & Service 기획 • Data Driven Computer-Aided Design	Less Critical	Low	Wide Area	Important

1) eMBB : Enhanced Mobile Broadband 2) mMTC : massive Machine Type Communications 3) uRLLC : Ultra-Reliable and Low-Latency Communications
4) AR : Augmented Reality

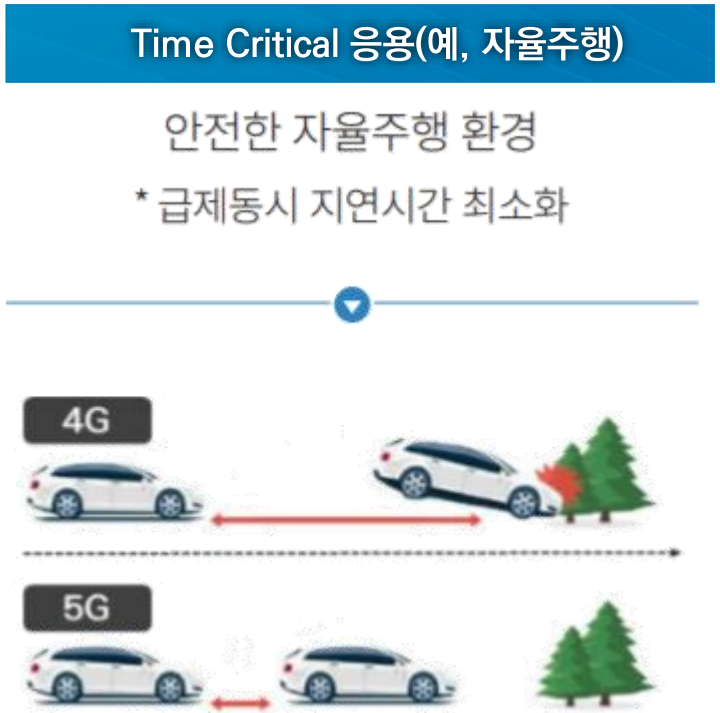
SK telecom

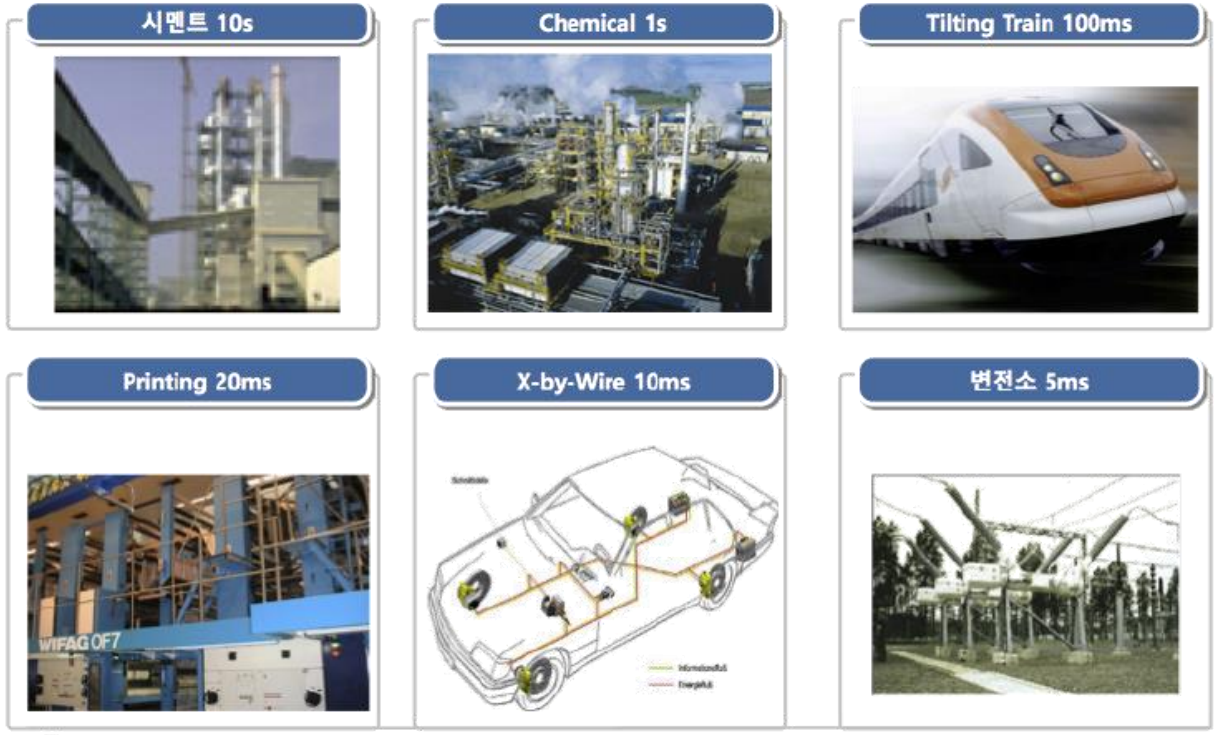


*source: 관계부처합동, 혁신성장을 위한 5G+ 전략, 2021, 3GPP

1 GS-Broker 의 필요성

- **(Critical Time) 응용별/산업군별로 요구되는 Time Limitation이 서로 다름**
  - 응용별/산업군별로 사용되거나 활용되는 엣지 디바이스가 다르고, 요구되는 Latency와 Service Critical Time이 상이함
  - Service Critical Point 조건에 부합하는 GS-Broker가 요구됨.



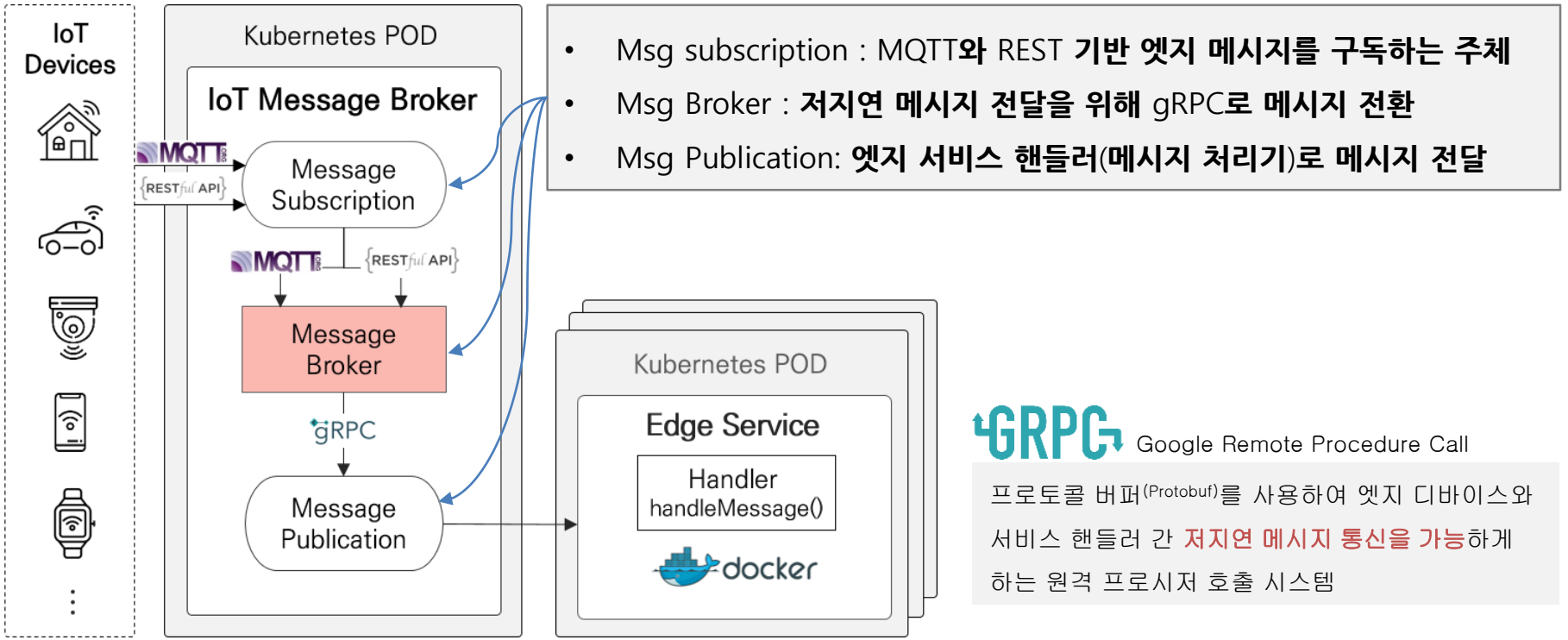


출처: LS산전 (2019.3, 5G-ACIA & 5G Forum Joint Workshop, 권대현)

GS-Broker/GS-MQ 구조

1 GS-Broker 구조

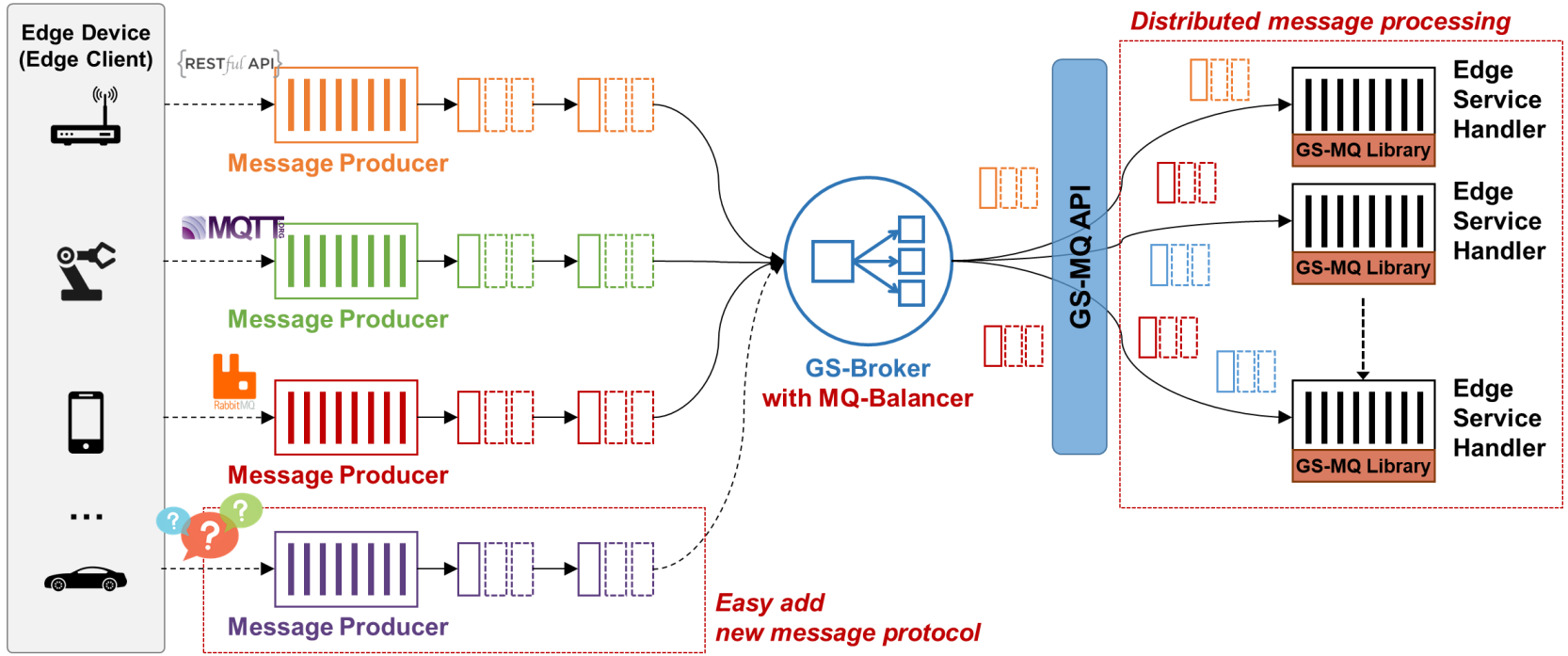
- **GS-MQ(Gedge Service Message Queue) 기반의 이종 엣지 디바이스 데이터 고속 처리 기술**
  - GEdge Platform 중심의 엣지 서비스와 IoT 디바이스 간 메시지 연계 프로토콜
  - gRPC를 활용한 저지연 메시지 전달 기술 적용



<GEdge Platform의 이종 엣지 메시지 브로커 구조>

1 GS-Broker 구조

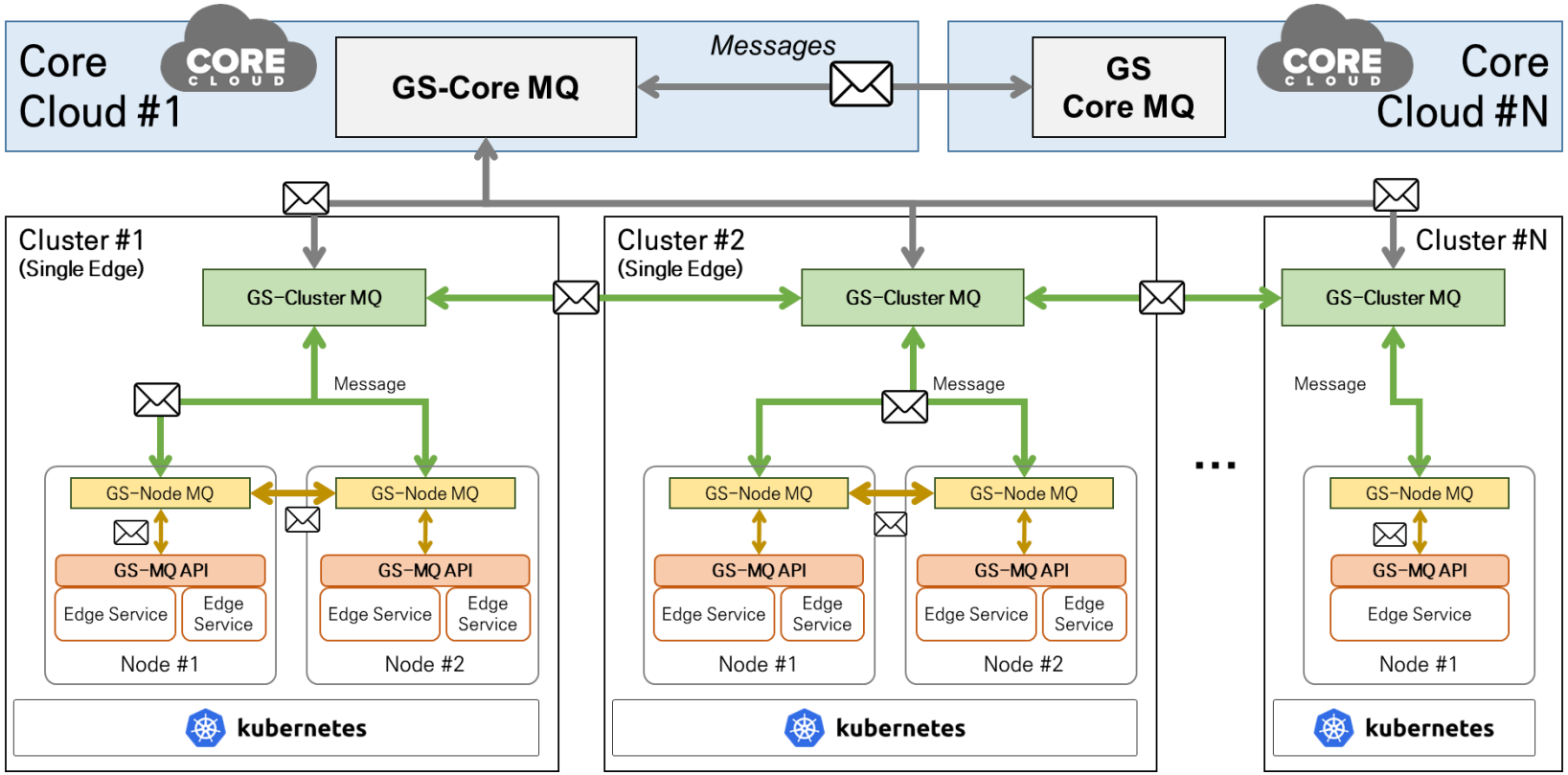
- **GS-MQ(Gedge Service Message Queue) 를 활용한 엣지 서비스 핸들러 간 메시지 공유**
  - 메시지 전송, 수신, 정책설정(데이터 분류/Priority선별 등) 을 통한 메시지 공유
  - GS-Broker를 활용한 분산 메시지 처리 환경 제공 및 유연한 확장이 가능한 메시지 브로커



<GEdge Platform의 이종 엣지 메시지 브로커 구조-MQ Balancer>

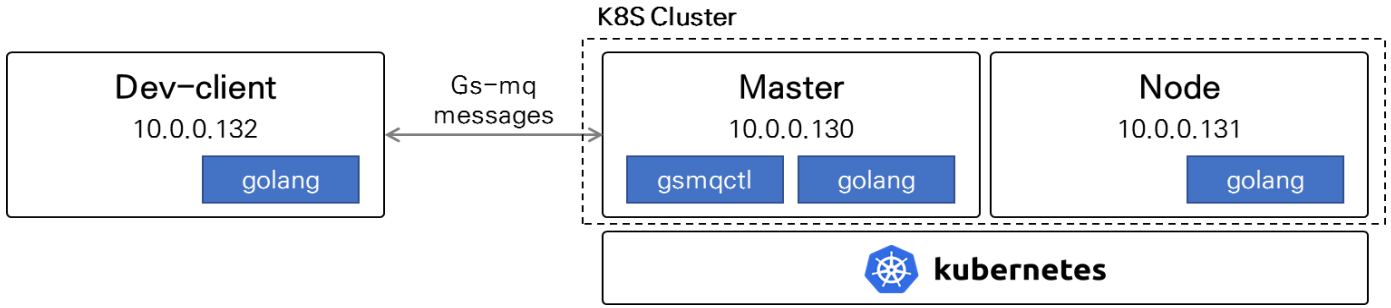
2 GS-MQ 구조

- **다중 클라우드 GS Engine 환경이 고려된 메시지 교환 도구 제공**
  - GS-MQ와 엣지 디바이스 브로커가 메시지를 교환하는 구조 설계
  - GS-CoreMQ, GS-ClusterMQ, GS-NodeMQ로 구성되어 수평적 또는 직 하위(Depth 1) MQ와 메시지 교환 가능



2 GS-MQ 구조

- **GS-MQ의 구성 및 메시지 공유 검증**
  - K8S 구성(1 Master & 1Node in k8s)-(1Dev-Client) : Kubernetes 1.22, Golang 1.16.6
  - GS-MQ Dev-Client 메시지 전송에 따른 메시지 교환 기능 검증.




```
func main() {
    (…)

    defer client.Close()
    channel := "dmkim_gsmq_channel"

    sendResult, err := client.NewQueueMessage().
        SetChannel(channel).
        SetBody([]byte("dmkim-kubemq-test-message_1")).
        Send(ctx)

    log.Printf("Send to Queue Result: MessageID:%s,
    Sent At: %s\n", sendResult.MessageID, time.Unix(0,
    sendResult.SentAt).String())
```

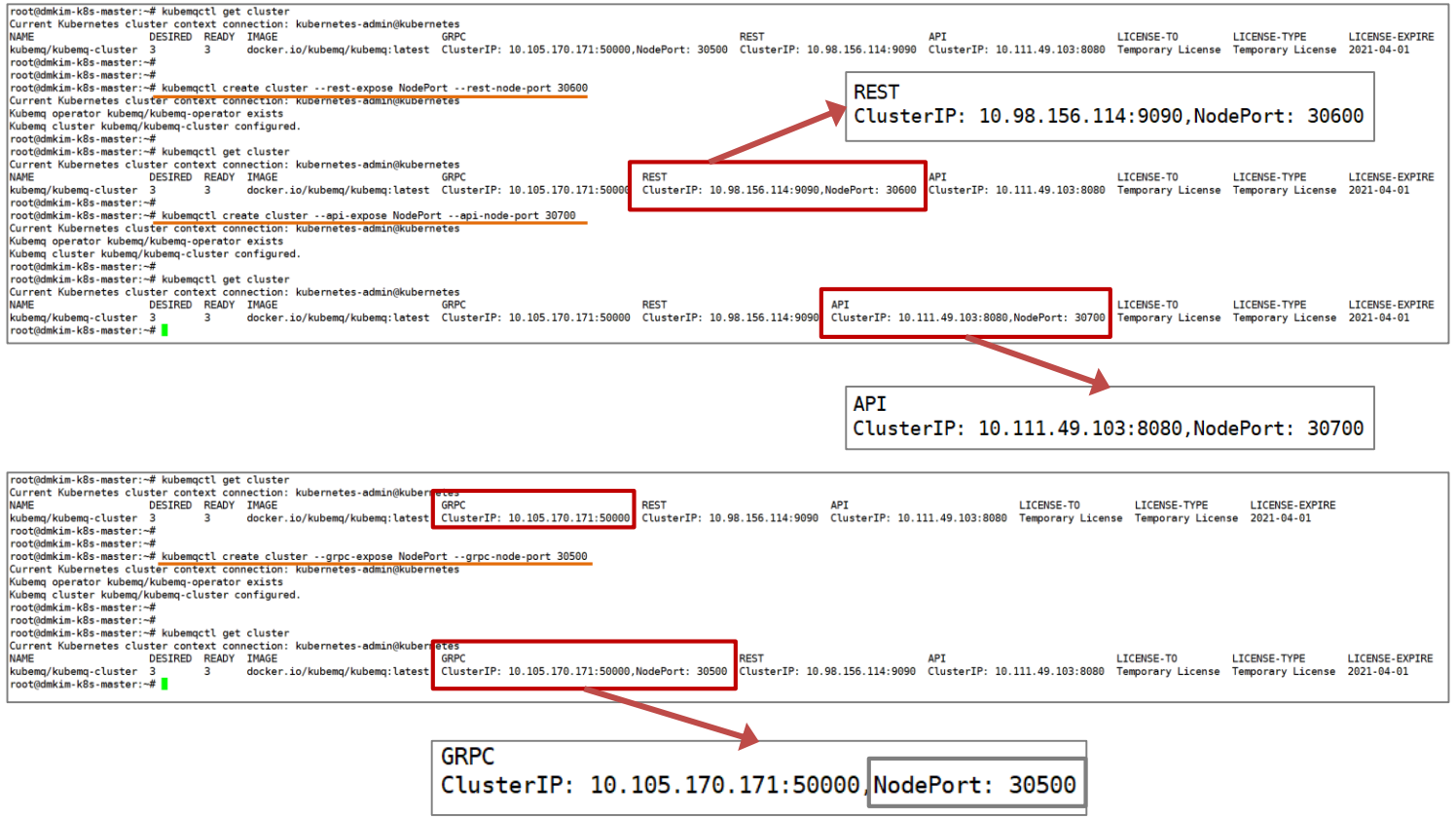
```
 receiveResult, err := client.NewReceiveQueueMessagesRequest().
        SetChannel(channel).
        SetMaxNumberOfMessages(1).
        SetWaitTimeSeconds(1).
        Send(ctx)

    log.Printf("Received %d Messages:\n",
    receiveResult.MessagesReceived)

    for _, msg := range receiveResult.Messages{
        log.Printf("MessageID: %s, Body: %s",
        msg.MessageID, string(msg.Body))
    }
}
```

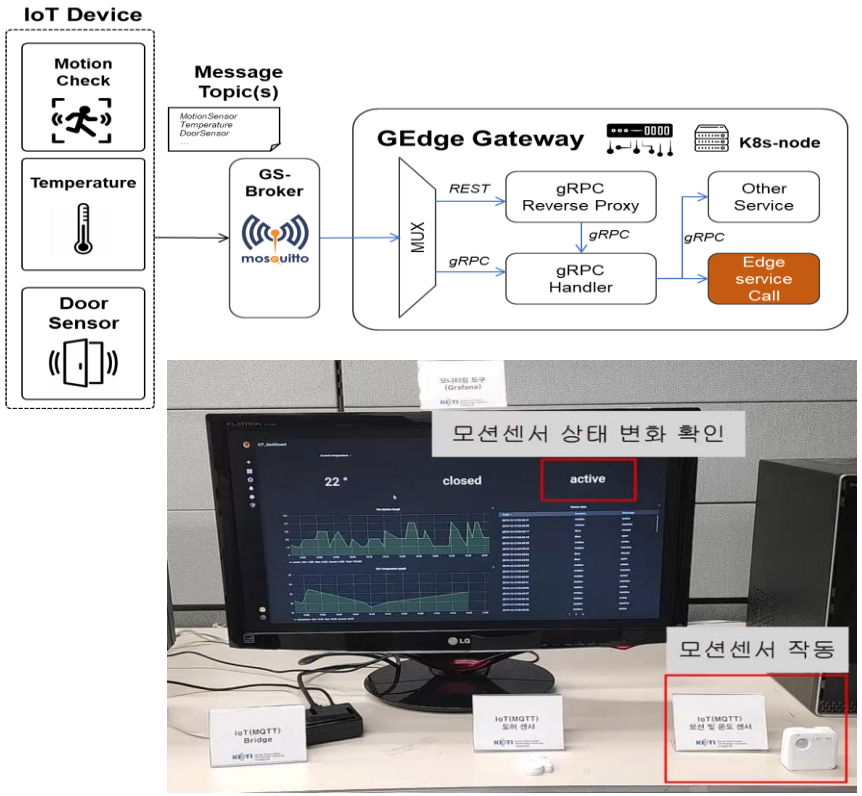
2 GS-MQ 구조

- **저지연 데이터 처리 및 클러스터 간 데이터 공유를 위한 GS-Broker&MQ 연계 구조**
  - GS-MQ와 GS-Broker간 연결을 위한 데이터 인터페이스 연결
  - GS-MQ와 gRPC 인터페이스 연결



3 GS-Broker/GS-MQ 구성

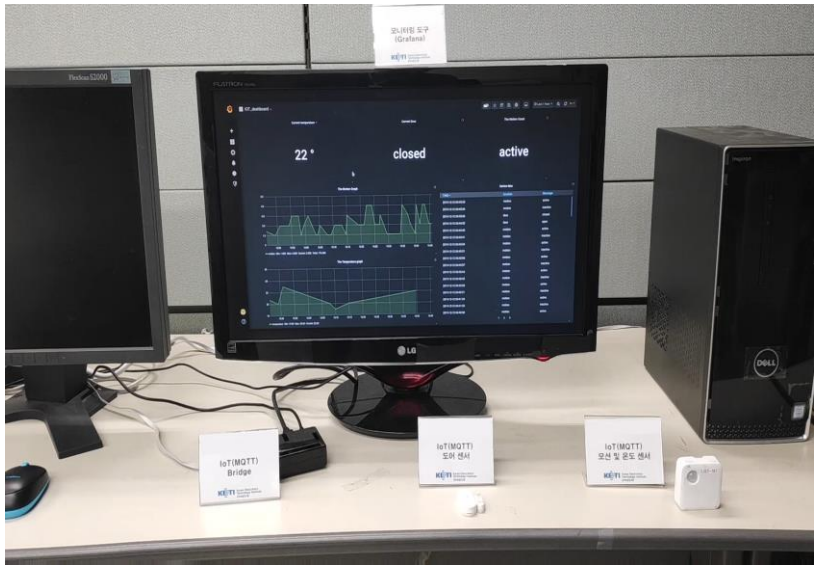
- **엣지 디바이스 구성을 통한 GS-Broker 기능 검증**
  - gRPC를 활용한 저지연 메시지 전송 기능 검증 : gRPC + K8S Pods
  - MQTT Broker와 센서(Motion/Door)를 연동한 GS-Broker 테스트 환경 구성 및 기능 검증



<GS-Broker 기초 기능 검증 구조 및 실험 환경

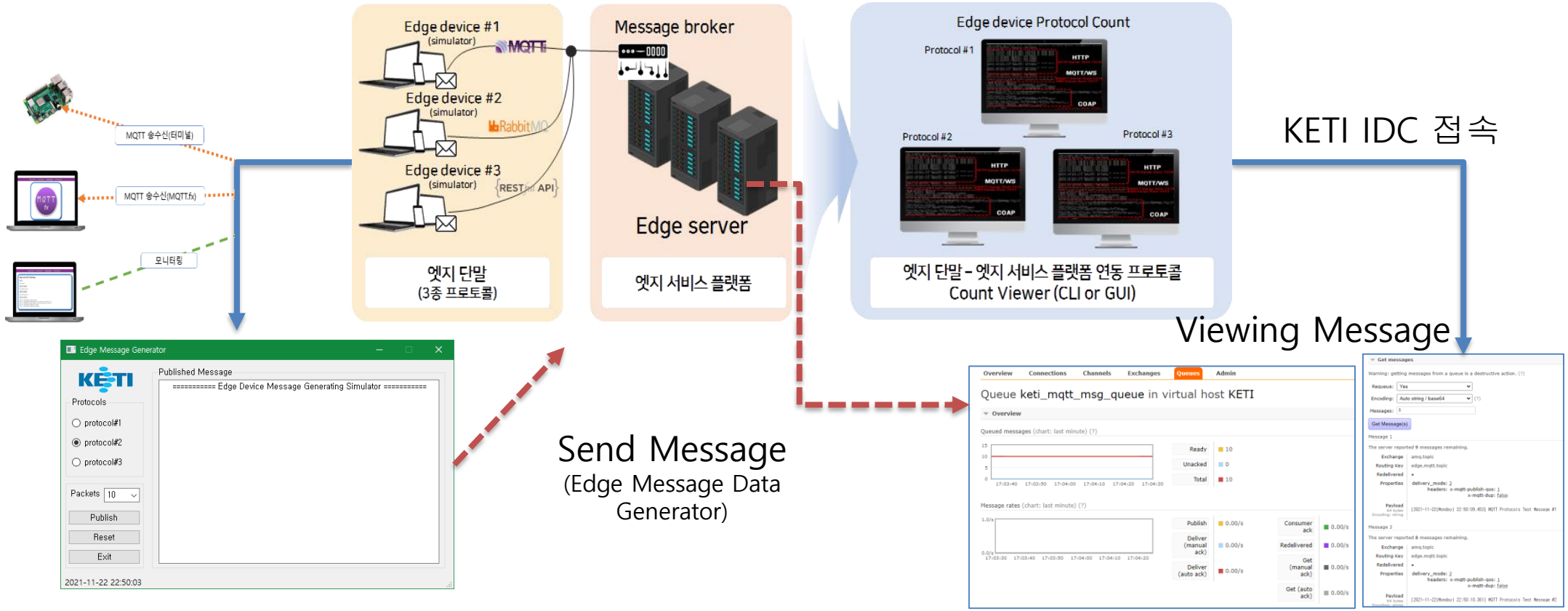
```
import datetime
import json
import os
import paho.mqtt.client as mqtt
import requests
topic_name = os.getenv("topic", "sensor-readings")
gateway_url = os.getenv("gateway_url", "http://127.0.0.1:8080")
print("Using gateway {} and topic {}".format(gateway_url,
topic_name))
def on_connect(client, userdata, flags, rc):
print("Connected with result code "+str(rc))
# Subscribing in on_connect() means that if we lose the
connection and
# reconnect then subscriptions will be renewed.
client.subscribe(topic_name)
# The callback for when a PUBLISH message is received from the
server.
def on_message(client, userdata, msg):
with open("./samples.txt", "a") as f:
r = json.loads(str(msg.payload))
r["created_at"] = str(datetime.datetime.now())
f.write(json.dumps(r) + "\n")
f.close()
print(msg.topic+" "+json.dumps(r))
res = requests.post(gateway_url + "/function/accept-sample",
json=r)
print("Log reading with function: ", res.status_code)
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("test.mosquitto.org", 1883, 60)
# Blocking call that processes network traffic, dispatches
callbacks and
# handles reconnecting.
# Other loop*() functions are available that give a threaded
interface and a
# manual interface.
client.loop_forever()
```

<Python 기반 GS-Broker 핵심 코어 개발>



3 GS-Broker/GS-MQ 구성

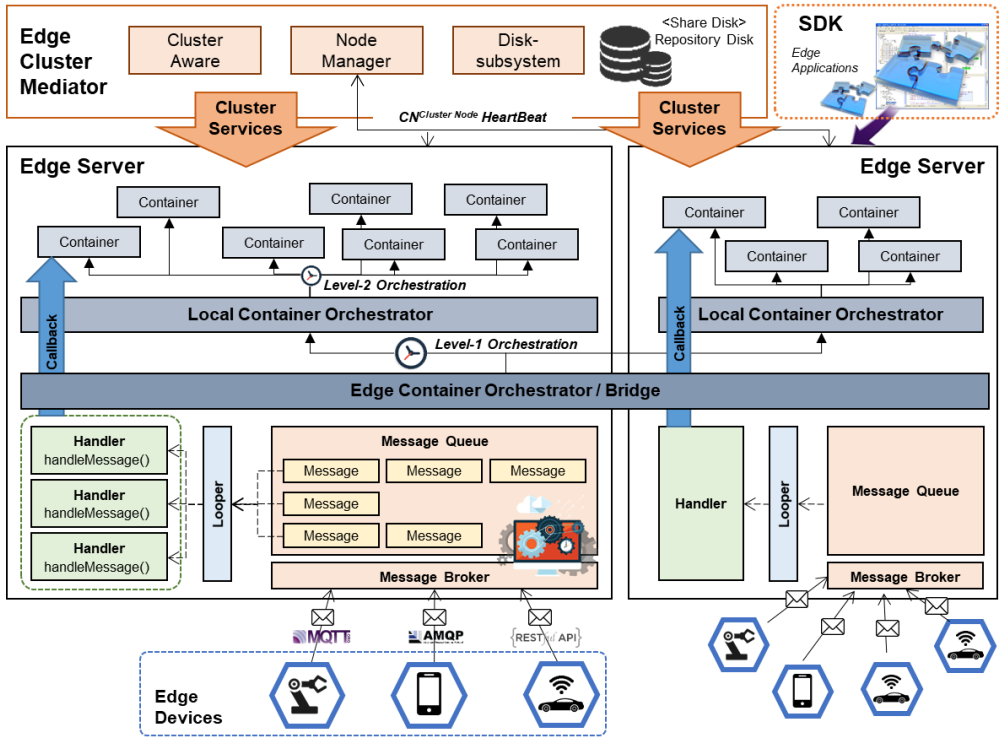
- **Edge Message Generator와 Edge Server 연결을 통한 대량 데이터 전송 테스트**
  - GS-Broker를 향해 메시지 전송 및 부하 테스트가 가능한 데이터 생성 도구 개발
  - GS-Broker에서 수집하고 Queue에 적재한 데이터를 확인할 수 있는 메시지 관리 대시보드 개발



IV GS-Broker/GS-MQ 22년도 계획

1 GS-Broker/GS-MQ 22년도 계획

- **GEdge Service Platform 통합 기술 개발**
  - 6종 이상의 이종 메시지 프로토콜 처리를 지원하는 GS-Broker 개발 및 GEdge Service Platform 통합
  - gRPC와 GS-MQ를 기반으로 저지연 데이터 처리 및 데이터 파이프라인 기술 개발



<GEdge Service Platform 통합 및 클러스터 구성>

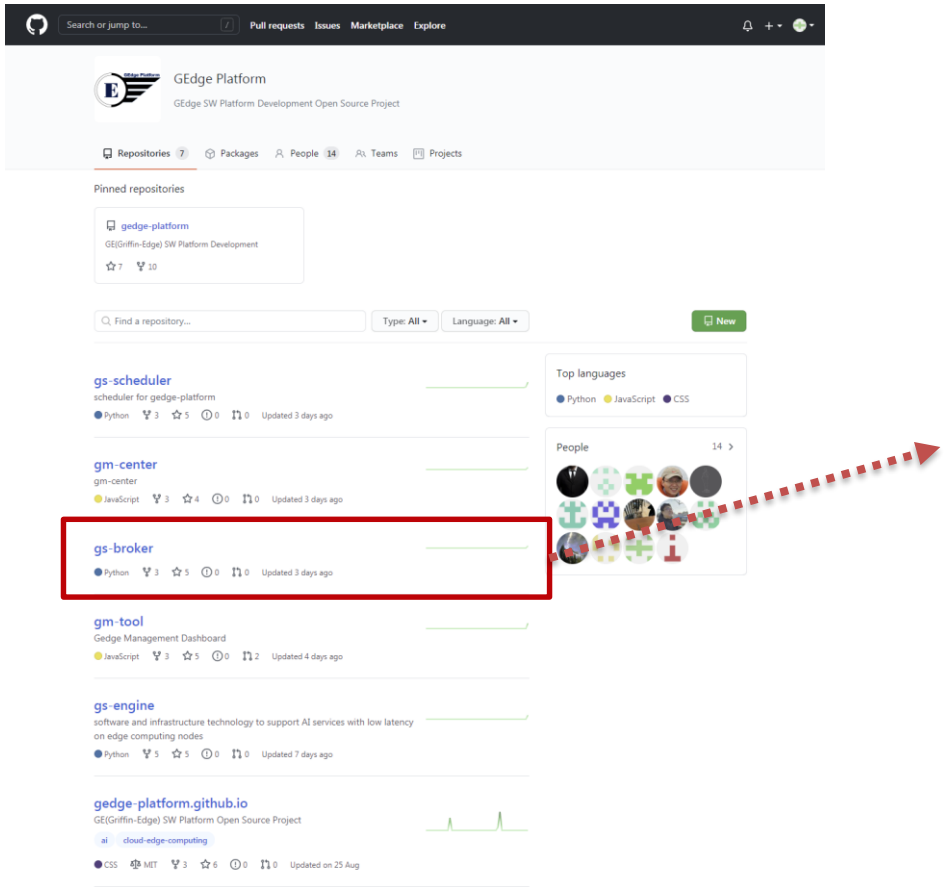
실증현장데이터수집/구축

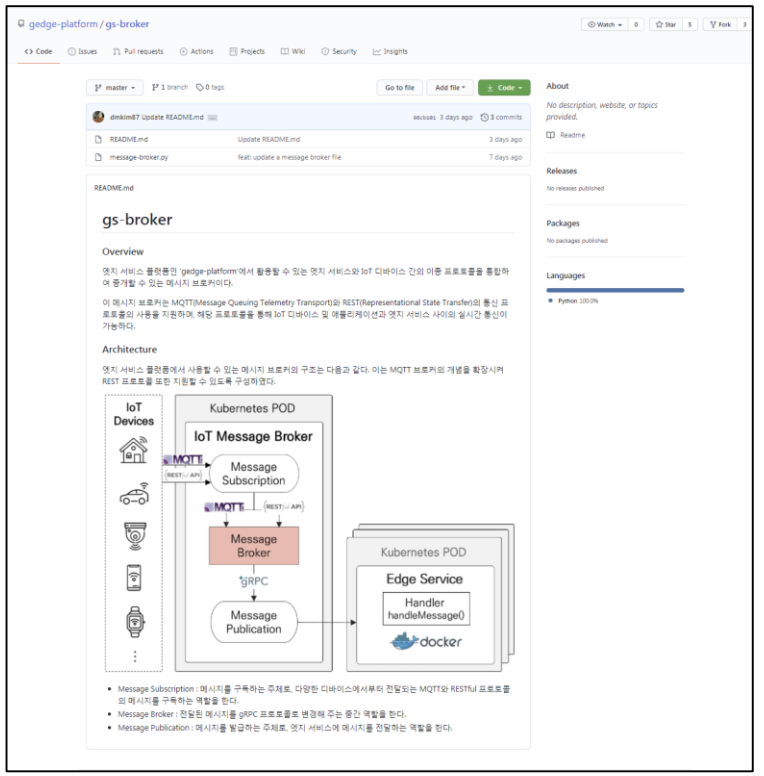
5G 데이터 전송 디바이스

스마트팜 엣지 데이터 수집(안)

1 GS-Broker/GS-MQ 22년도 계획

- **개발 결과물 공개를 위한 GitHub 활용 계획**
  - Gedge Platform 내 GS-Broker를 통하여 지속적인 결과물 공유
  - Message Broker의 다중 프로토콜 연동 구조 확보 및 gRPC 기능 검증





1 GS-Broker/GS-MQ 22년도 계획

- **공개 설명회 및 연구 결과물 시연 계획**
  - GEdge Platform 프로젝트 커뮤니티 활동 및 프로젝트 결과물 전시 및 시연을 통한 확산 도모
  - 공개설명회를 통한 GS-Broker 시연 및 플랫폼 연동 가이드 제공 예정

제 52회 한국전자전 2021 (Korea Electronics Show 2021, KES 2021)

행사 목적

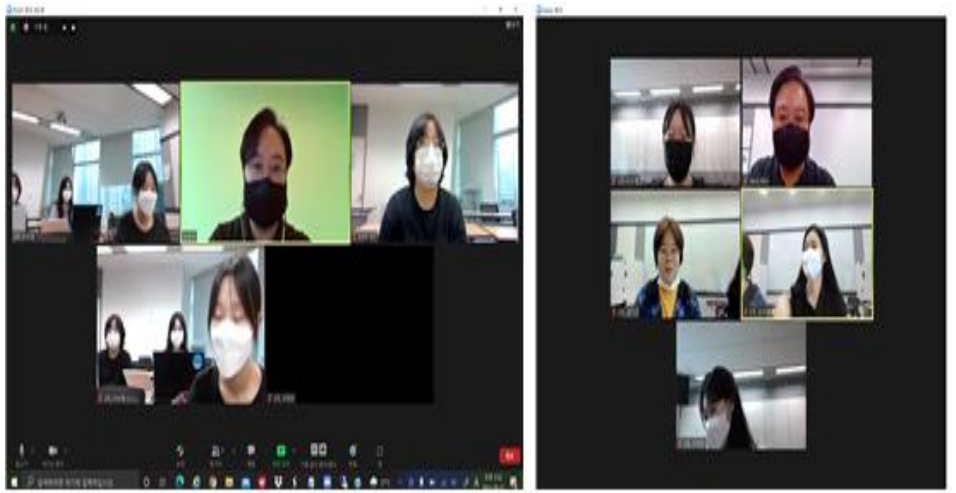
- 제 52회 한국전자전 KES 2021 참가를 통한 연구 결과물 대외 홍보 및 실증 시연 진행
  - → 산업통산자원부 주최, 제 52회 한국전자전 Korea Electronics Show 2021 (KES 2021)
  - → COEX 전시 홀, 2021. 10. 26 (화) ~ 2019. 10. 29(금) 진행

전시 내용

- 초저지연 지능형 클라우드 엣지 플랫폼을 위한 GEdge Platform/GS-Broker 소개 및 연구 결과물 실증 시연
  1. 다양한 엣지 디바이스를 지원하는 GEdge Platform/Broker
  2. GEdge Platform/Broker 기술의 필요성
  3. GS-Broker의 구조 및 핵심 기술
  4. GS-Broker 기능에 대한 실증 시연 데모 진행

전시 자료

결과물 설명회



감사합니다.

http://gedge-platform.github.io

GS-Engine Framework Core Developer (GS-Broker)
Hyun Woo Kim(hwkim@keti.re.kr)