

Microservice, DevOps & Multi-access Edge Computing

Mobile & Convergence Leading Company SPTEK

The contents of this material are confidential and proprietary to SPTEK Corporation and may not be reproduced, published, or disclosed to others without the prior written consent of SPTEK.

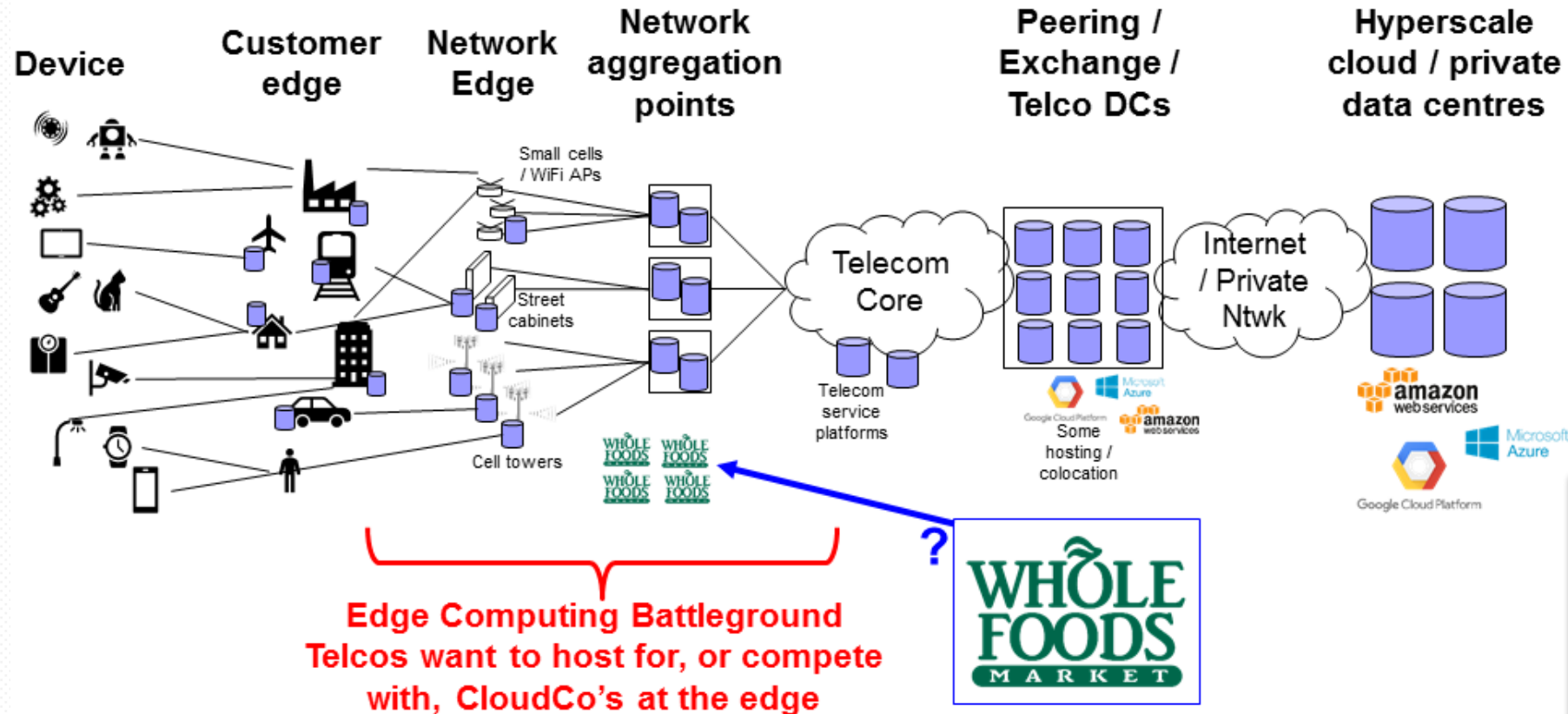


Contents

1. Microservice Architecture
2. DevOps
3. Multi-access Edge Computing

Amazon은 왜 Whole Foods를 인수하였는가?

Where **is** the edge, for edge computing?



Dean Bublely
@disruptivedean

If (Mobile) Edge Computing becomes a big deal for telcos, expect Amazon & co to cut deals with Tesco / Walmart etc for local mini data-ctrs

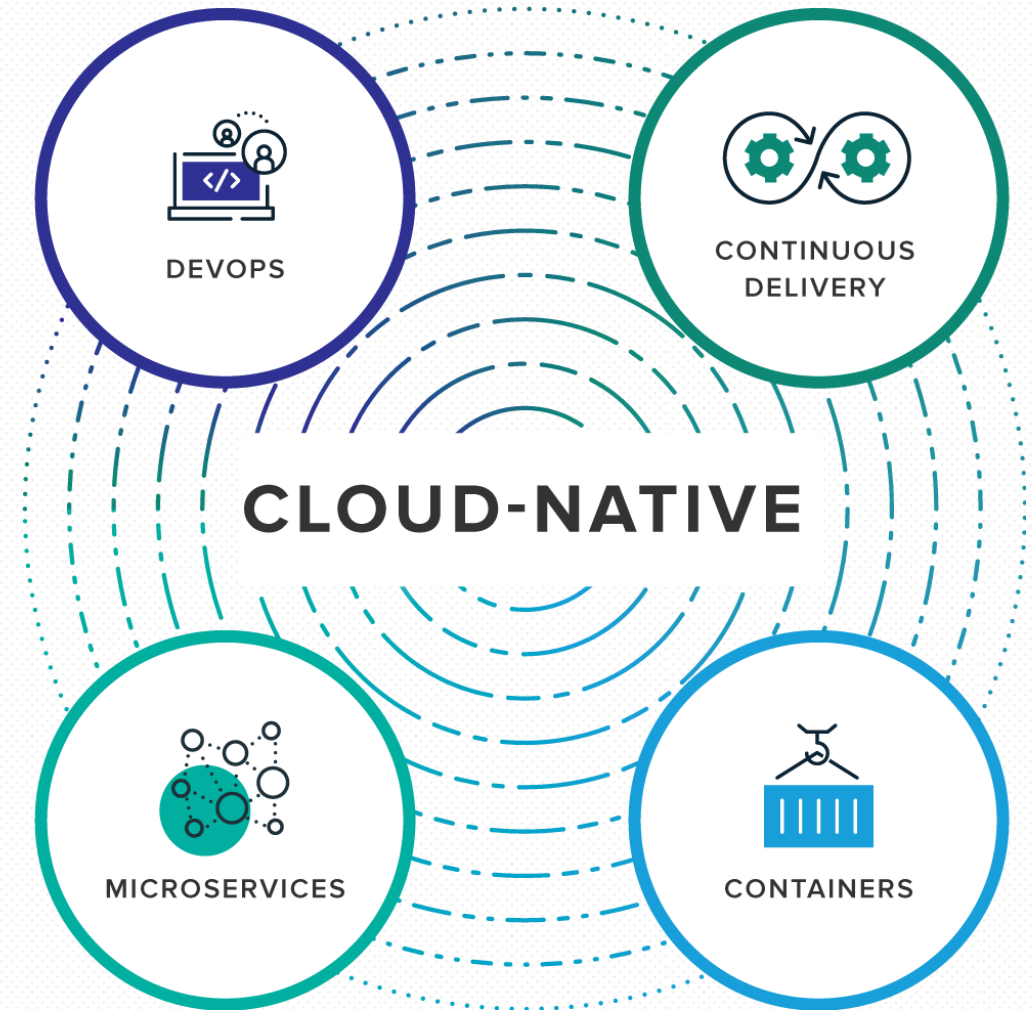
9:20 am · 16 Jun 2016

Source: <http://disruptivewireless.blogspot.com/2017/06/does-amazons-purchase-of-whole-foods.html>

Cloud Native Application

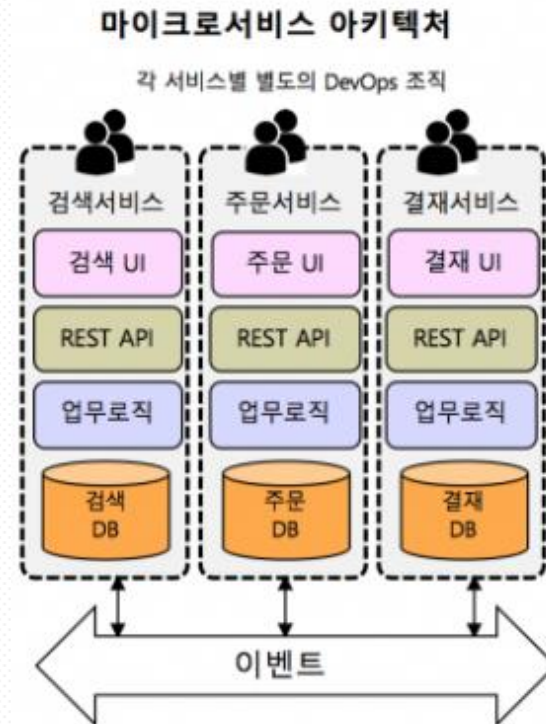
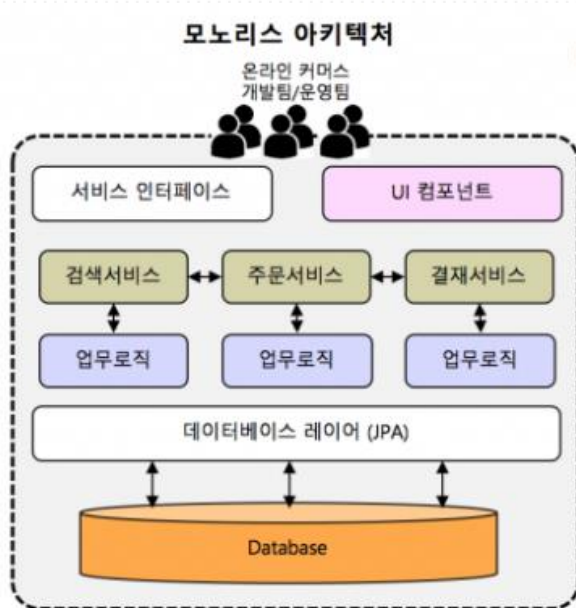
Cloud Native Application이란?

- 클라우드 컴퓨팅 모델의 장점을 모두 활용하는 애플리케이션을 개발하고 실행하기 위한 접근 방식
- Cloud Native Application을 개발/운영하기 위해 DevOps, CD, 마이크로서비스, 컨테이너 기술을 통합하고 자동화하는 플랫폼 필요



Microservices - What and Why?

- What is Microservices?
 - a software development technique that structures an application as a collection of loosely coupled services.



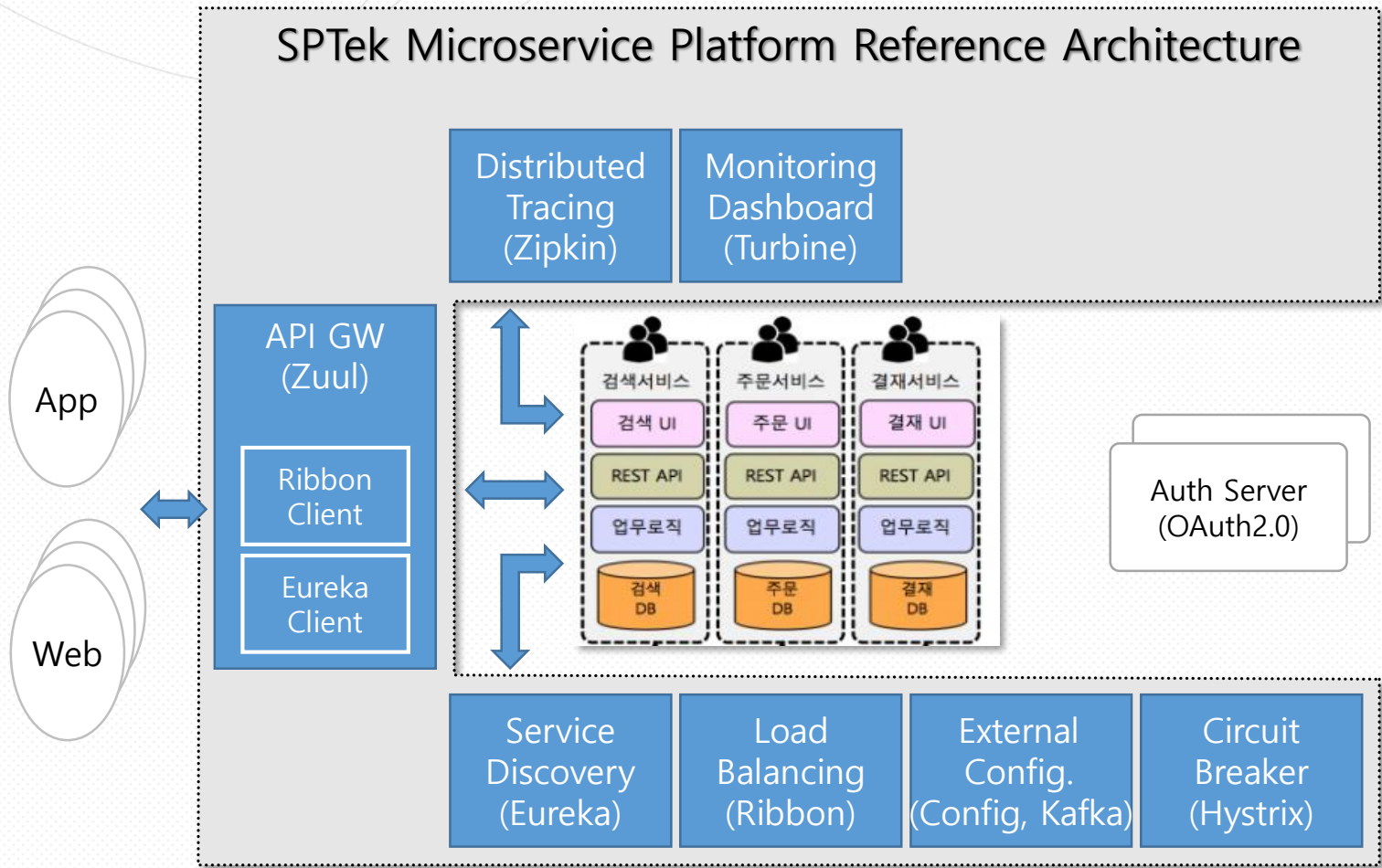
Why Microservices?

- **Independent Development** – All microservices can be easily developed based on their individual functionality
- **Independent Deployment** – Based on their services, they can be individually deployed in any application
- **Fault Isolation** – Even if one service of the application does not work, the system still continues to function
- **Mixed Technology Stack** – Different languages and technologies can be used to build different services of the same application
- **Granular Scaling** – Individual components can scale as per need, there is no need to scale all components together

Source: <https://developer.ibm.com/kr/developer-기술-포럼/2017/03/31/마이크로서비스를-위한-인프라스트럭처/>

SPTek Microservice Platform Reference Architecture

- Spring Cloud Netflix 기반 Microservice Platform Reference Architecture



Zuul

- 서비스 호출 관리 및 동적 라우팅
- 인증 및 보안 체크, 데이터 지표 수집

Eureka

- 동적 서비스 등록관리
- 미들티어 서버 load balancing, failover를 위한 서비스 배치

Ribbon

- Client side load balancing
- 라운드 로빈 등 부하 분산 방식 제공

Config

- 서비스와 설정 분리 관리
- 설정변경으로 인한 빌드/배포 필요 없음

Hystrix

- 원격 서비스 호출 구간 격리, 모니터링 라이브러리
- 모니터링 추가로 인한 부하 최소화

Zipkin

- 내부 마이크로 서비스 간 호출 경로, 시간 추적
- '샘플링'을 통한 '서비스의 외부 요청 처리 상태' 파악 목적

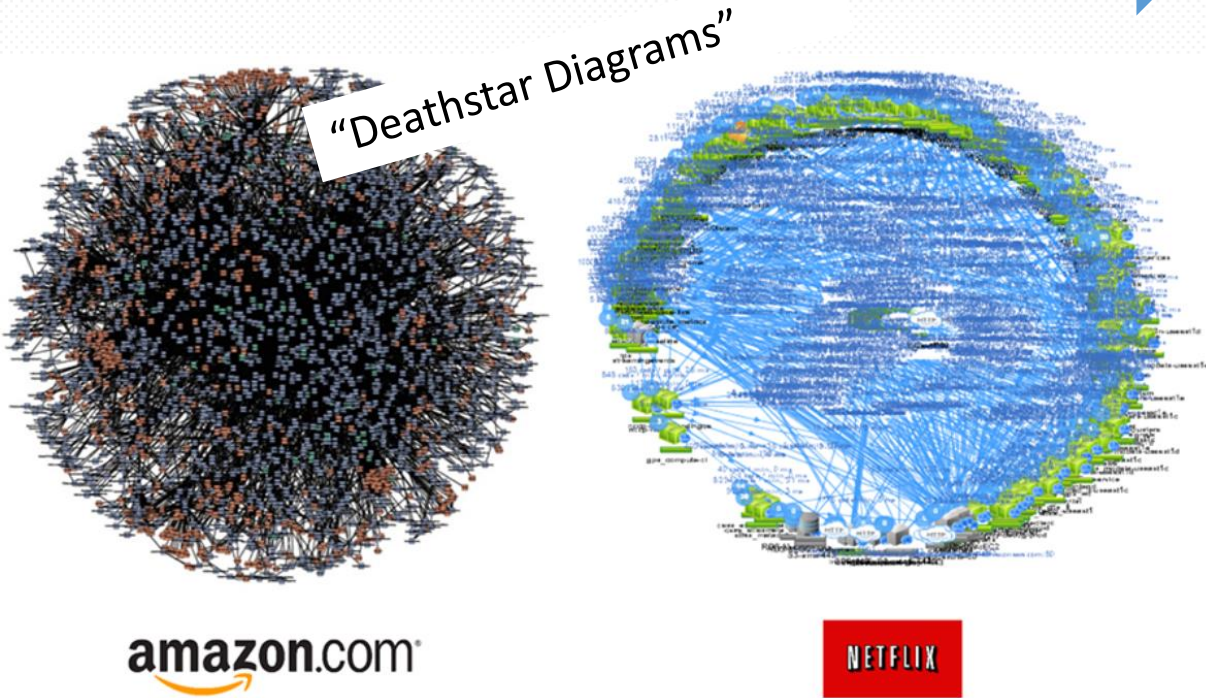
Turbine

- 각 WAS의 Command 실행 결과 수집, 통합 역할

Microservices & DevOps

Microservices Architecture는

- 필연적으로 SW복잡도를 증가시킨다.
- 빌드, 테스트, 배포, 유지관리를 어떻게?

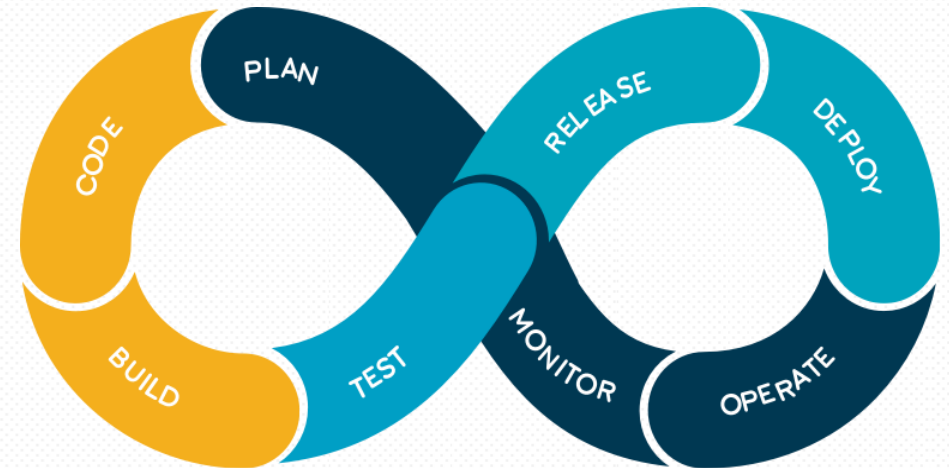


Source: <https://www.appcentrica.com/the-rise-of-microservices/>

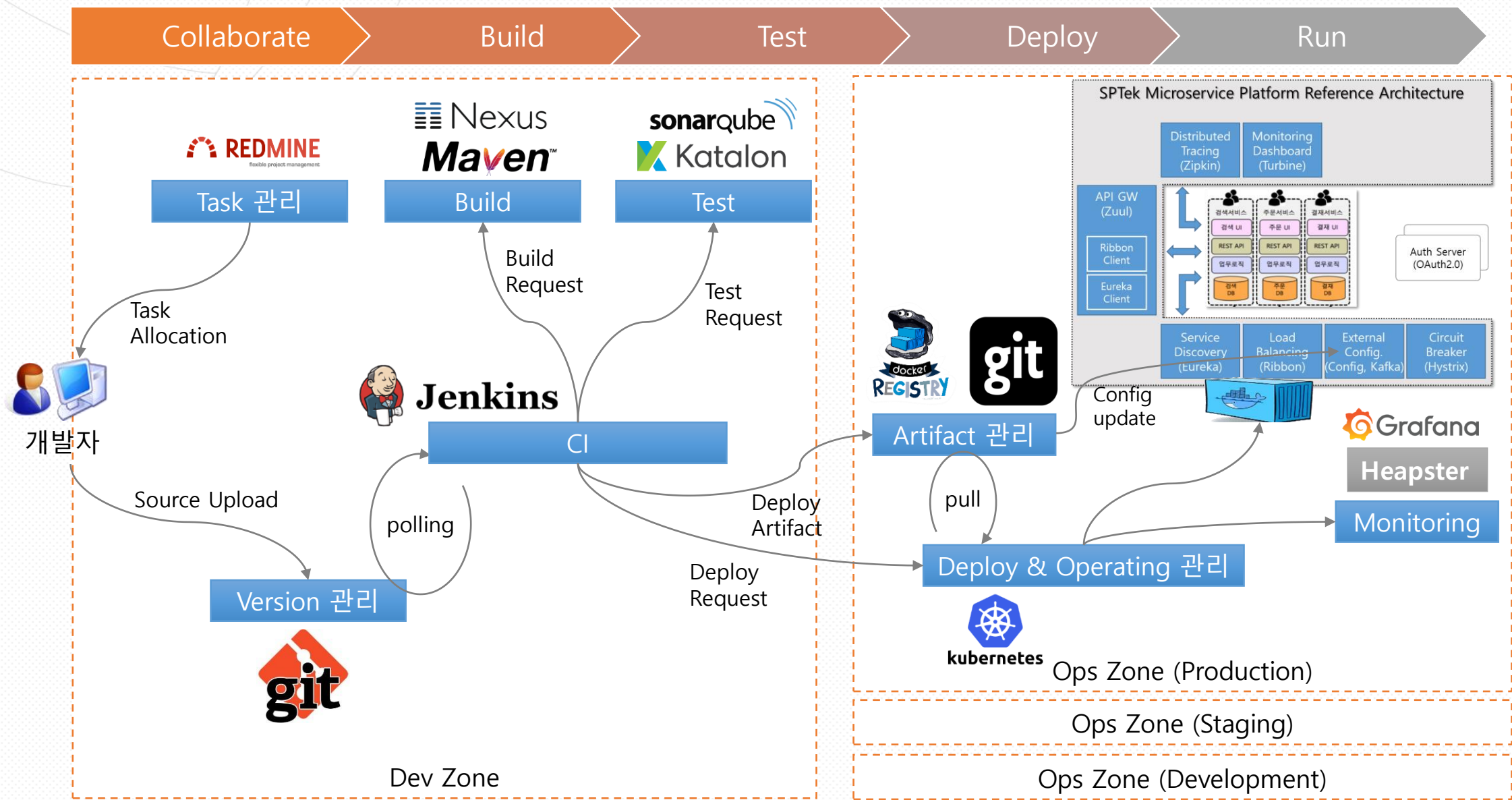
DevOps는

- Development와 Operation간의 통합을 강조하는 문화, 행동
- **"사업 목적 달성"**을 목표로
- 통합, 검증, 배포, Infra.관리 모든 단계에서 "자동화"와 "Monitoring"을 통해
- 개발 기간 단축, 배포 빈도 증가, 결과물 안정성 강화를 달성 하는 것

Source: Wikipedia

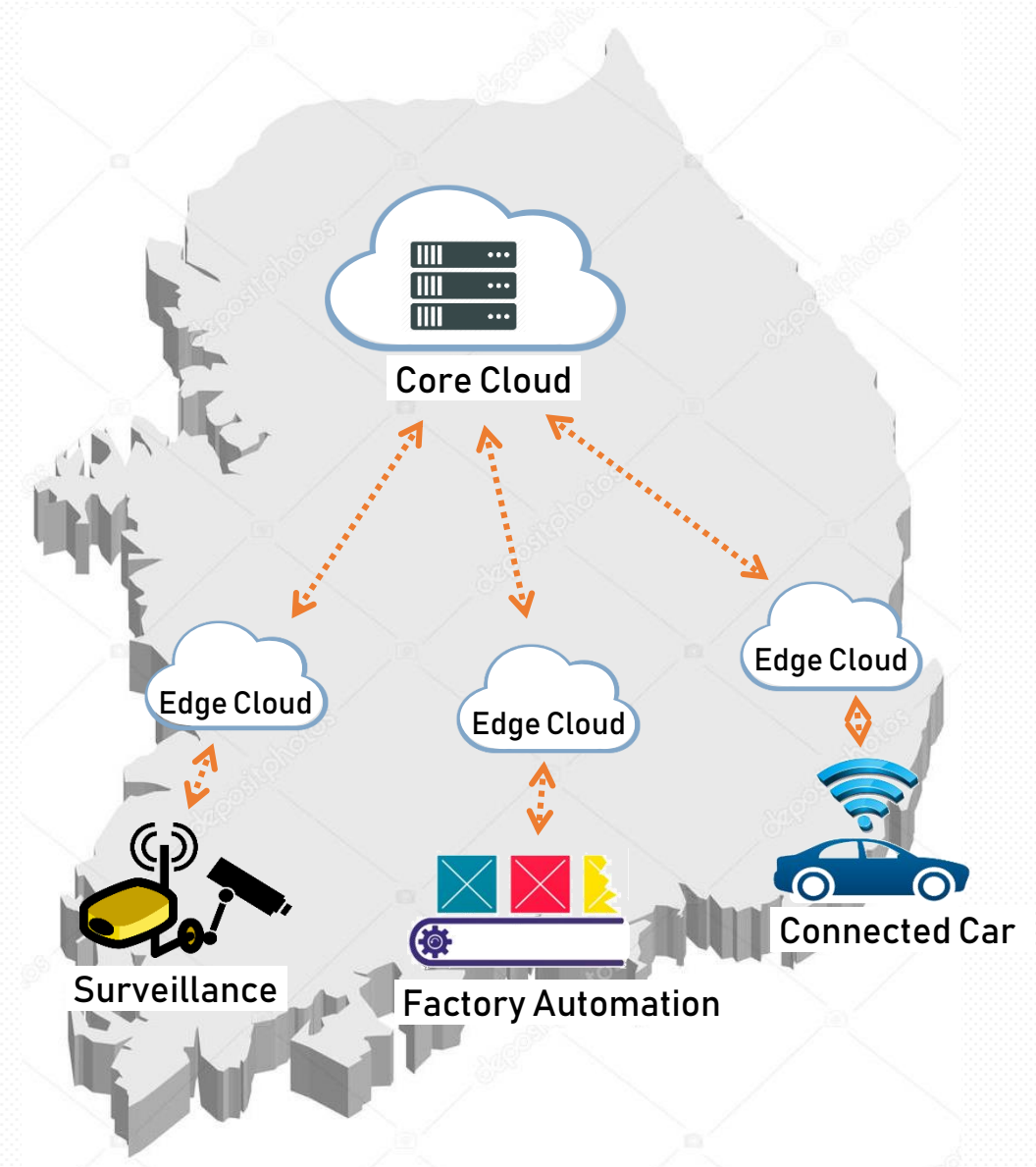


SPTek DevOps Stack & Flow for Microservice



Edge Computing이란?

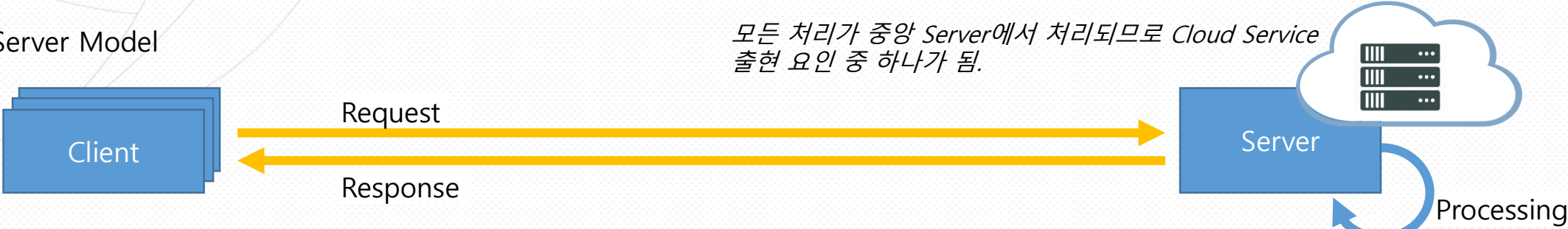
- Edge Computing?
 - Edge Computing refers to a broad set of techniques designed to move computing and storage out of the remote cloud (public or private) and closer to the source of data (ETSI)
 - Mesh network of micro data centers that process or store critical data locally and push all received data to a central data center or cloud storage repository, in a footprint of less than 100 square feet. (IDC)
- 주요 출현 동인
 - Roundtrip Latency
 - Connected Car와 같은 새로운 Application들은 매우 낮은 지연시간을 요구
 - 중앙 Cloud에 설치된 Server는 거리에 따른 네트워크 장비 처리지연 유발
 - Amount of Data
 - IoT device 활용 증가와 Vehicle to everything (V2X) 기술 등의 확산으로 네트워크에 전송되는 데이터의 양 폭증
 - Edge computing은 발생한 데이터를 중간 처리함으로 데이터 증가 완화



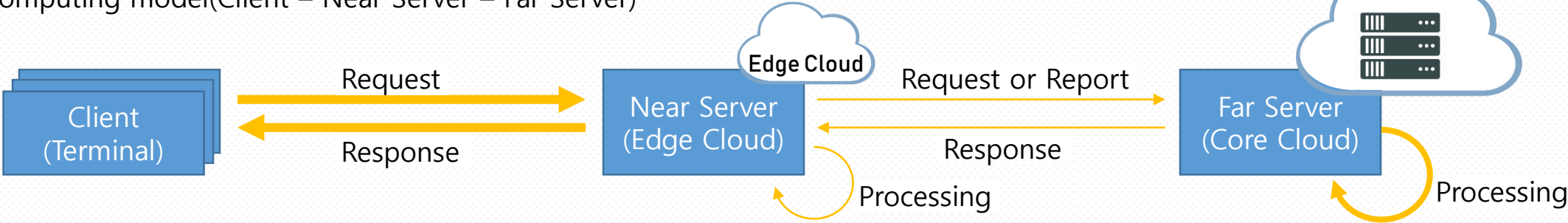
Edge Computing Application Design Model

- Application Design Model의 변화

- Client-Server Model

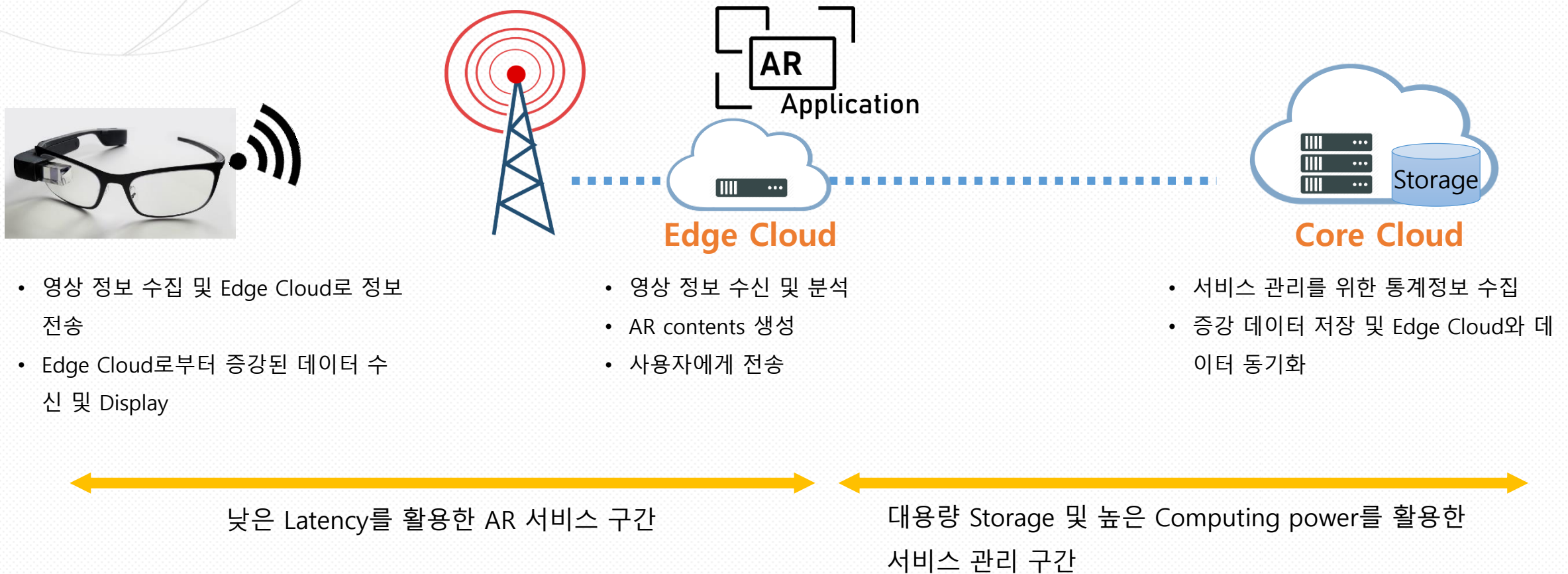


- Edge Computing model(Client – Near Server – Far Server)

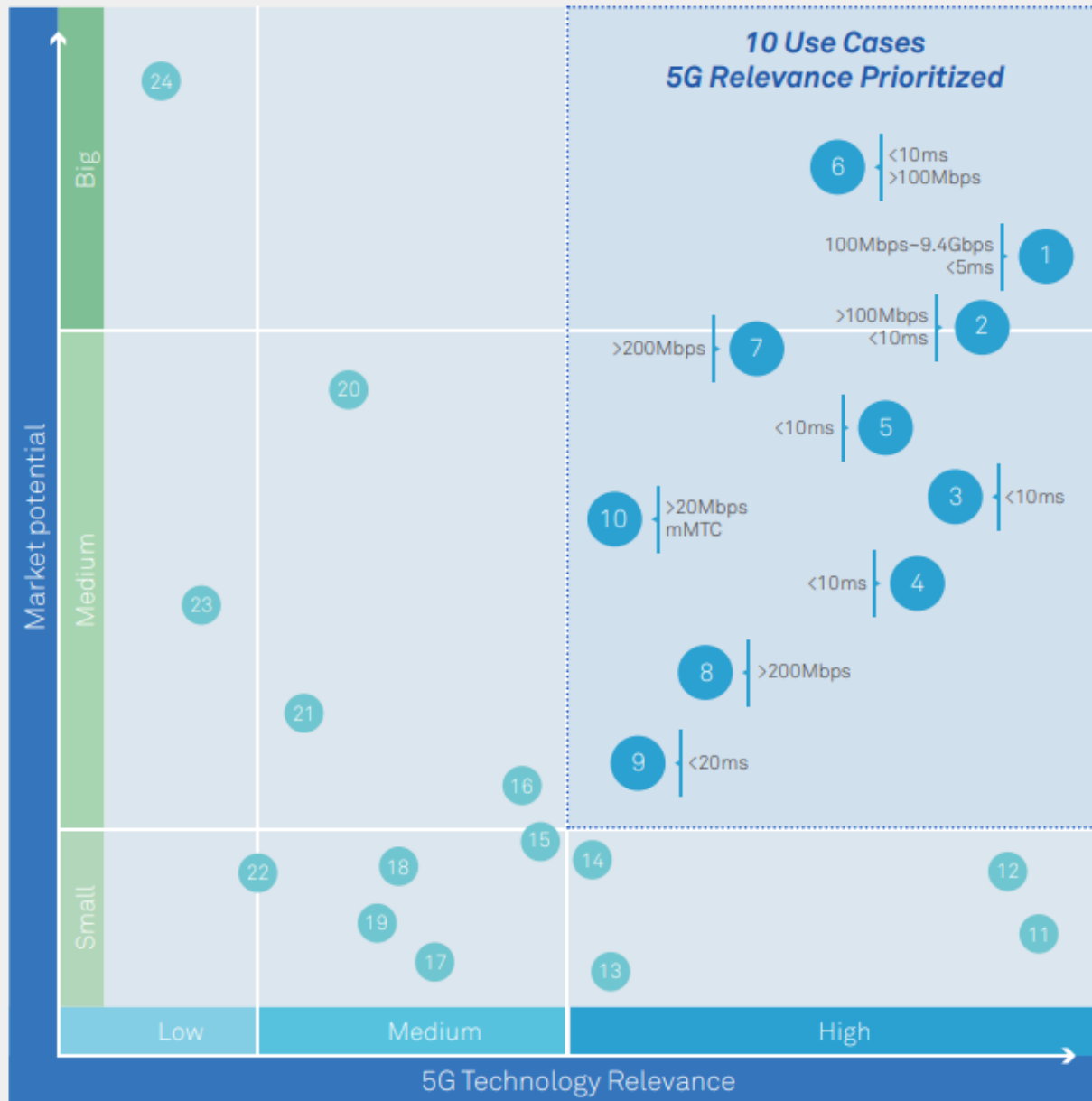


Terminal component	Event/Data에 대한 1차 처리 및 서버 추가 처리가 필요한지 결정 최소한의 computing power로 거의 지연없이 처리
Edge component	Terminal device에서 직접 처리한 것과 유사한 매우 낮은 처리 지연 상대적으로 높은 computing power를 바탕으로 Terminal device에서 처리되지 못한 Task 처리
Remote components	대용량 저장장치, Database, Computing Power를 이용한 Task 처리

Edge Computing Use case: Augmented reality application



TOP TEN 5G USE CASES by Huawei



1. Cloud Virtual & Augmented Reality – Real-time Computer Rendering Gaming/Modeling
2. Connected Automotive – ToD, Platooning, Autonomous Driving
3. Smart Manufacturing – Cloud Based Wireless Robot Control
4. Connected Energy – Feeder Automation
5. Wireless eHealth – Remote Diagnosis With Force-Feedback
6. Wireless Home Entertainment – UHD 8K Video & Cloud Gaming
7. Connected Drones – Professional Inspection & Security
8. Social Networks – UHD/Panoramic Live Broadcasting
9. Personal AI Assistant – AI Assisted Smart Helmet
10. Smart City – AI-enabled Video Surveillance

Multi-access Edge Computing

Multi-access Edge Computing (MEC)

- 이동통신시장을 위해 ETSI를 중심으로 Edge Computing을 규격화 한 것
- Multi-access란 LTE, 5G 이동통신network, WiFi network, 유선 network등 다양한 access network을 통해 접속할 수 있다는 의미

Multi-access Edge Computing의 본질

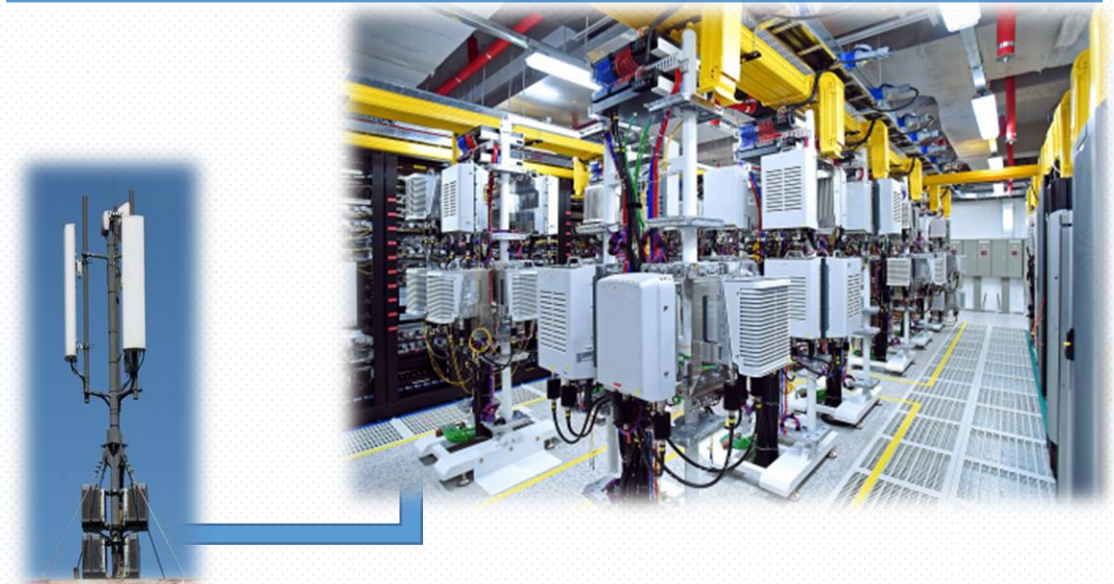
1. 통신사 및 IT기업에게 새로운 ecosystem과 value chain을 통한 Business 기회
 - 통신사는 Radio Access Network (RAN) edge를 IT기업에게 공개
 - IT기업은 이용자에게 근접한 Cloud Infra. 특성과 Radio Network 정보를 활용한 새로운 서비스 창출
 - 서비스 예. video analytics, location services, Internet-of-Things (IoT), augmented reality, optimized local content distribution and data caching

서비스

서비스

서비스

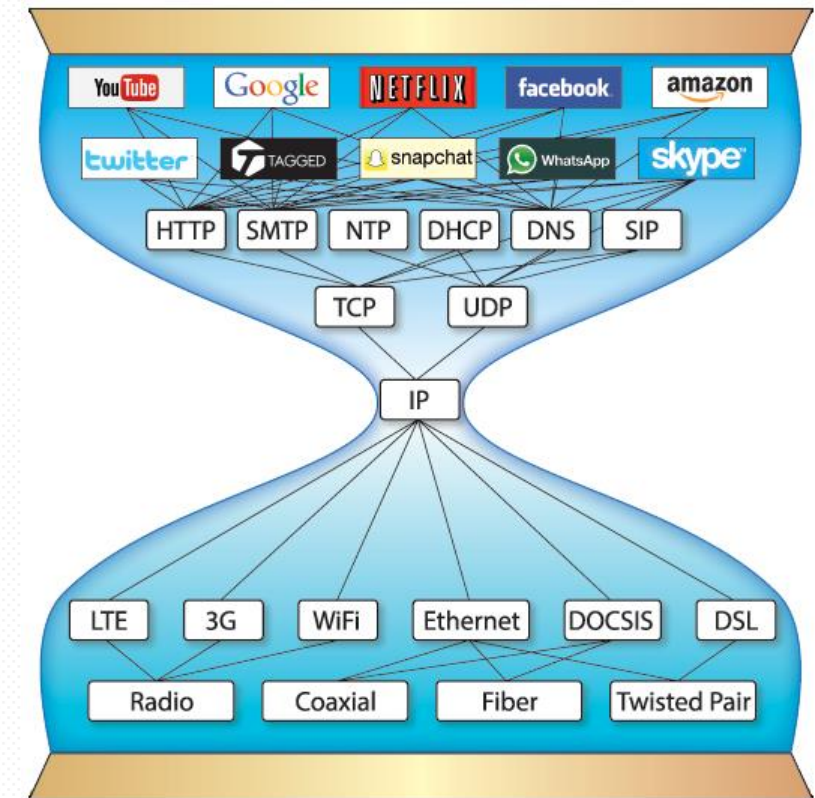
MEC Business Platform



Multi-access Edge Computing

2. IT와 telecommunications network간의 융합

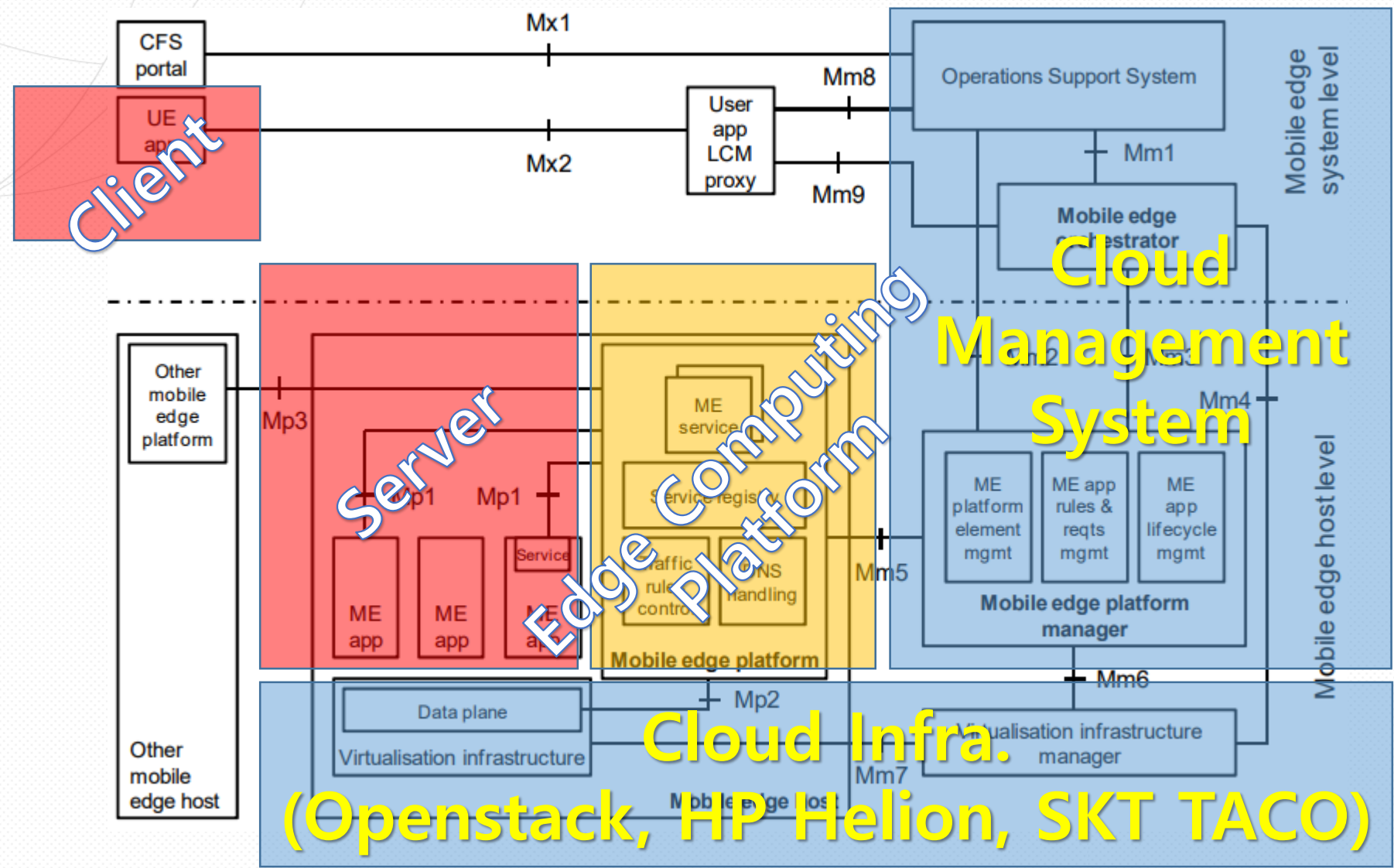
- MEC는 network(mobile base stations)의 Edge에서 Cloud-computing capabilities를 제공하는 것
 - 특징: ultra-low latency, high bandwidth, real-time access to radio network information
- MEC platform은 (Hourglass model 장점 + Application과 Network간 협업)
 - Hourglass model은 시스템 설계의 핵심 이론 – Application과 Network은 상호 불가지론적 관계
 - Application의 실행 환경 정보(latency, throughput, etc.) 수집, 활용 가능
 - Radio Network Information Service – Network 정보 모니터링, Bandwidth 할당 요청, 기타 Application 요구 정보 전달
 - Network이 Black-box에서 서비스를 위한 Resource 중의 하나
 - 반대로 Network은 Application 정보에 기반하여 Network 효율화



the hourglass model of the internet's architecture

Source: <http://if-we.clients.labzero.com/code/posts/what-title-ii-means-for-tcp/>

MEC – Reference Architecture



Source: ETSI GS MEC 003 V1.1.1 (2016-03)

Deployment of MEC in NFV Environment

ME App Orchestrator

- Mobile Edge Service Orchestration 기능
 - Triggering application instantiation, termination, relocation
- ME App.에 대한 On-boarding, ME host selection

ME platform manager

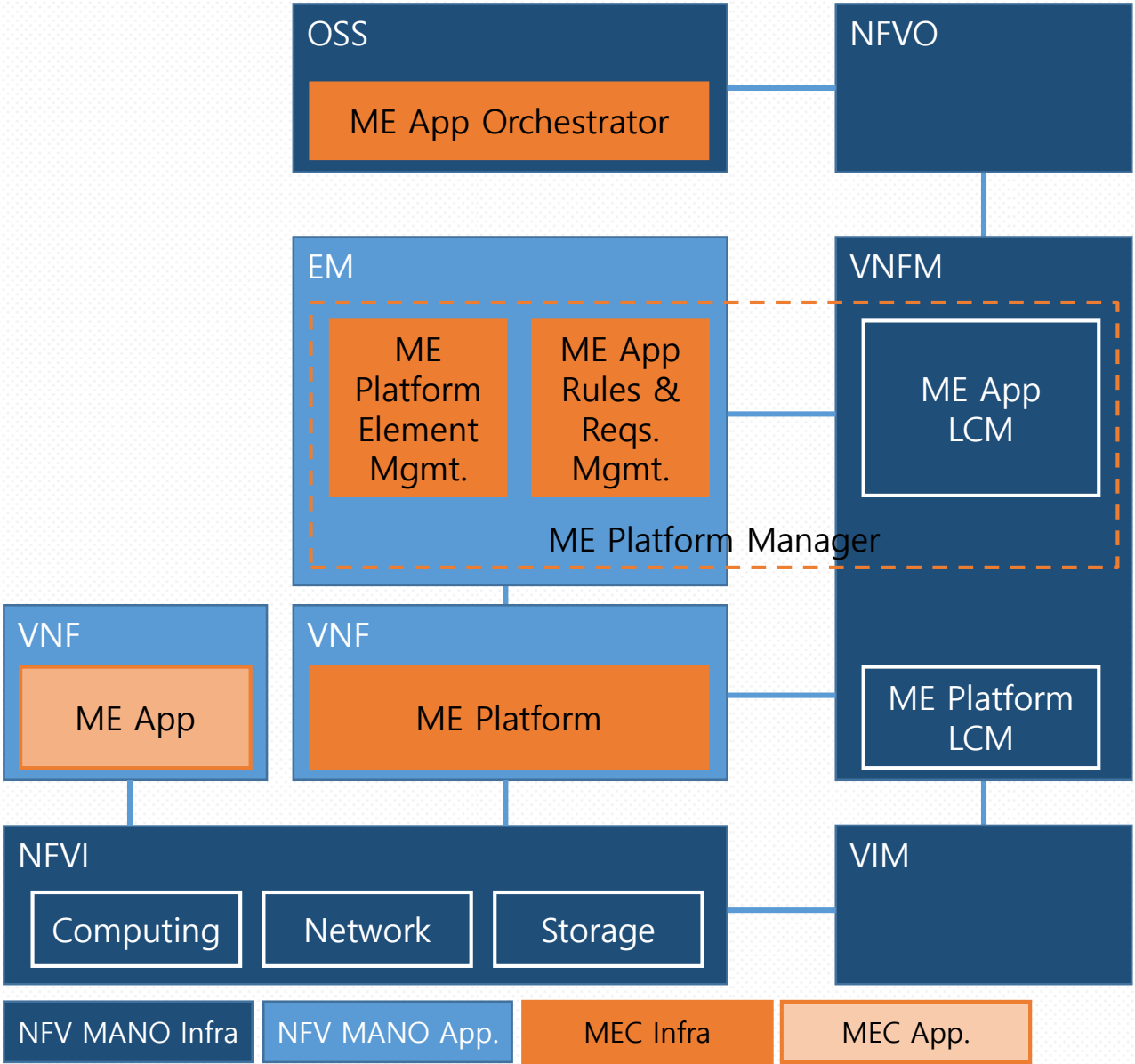
- Application life cycle management
- ME platform에 대한 element management 기능
- Application traffic rules/requirements 관리 기능

ME platform

- Service discover, advertise, consume, offer 기능
- 수신된 Application traffic rules에 따라 data plane 설정 기능
- DNS proxy/server, 영구 storage, time of day information 기능
- Radio Network Information, Location, Bandwidth Service 제공

ME application

- Microservice application
- maximum latency, required or useful services 등 요구사항 제시



Microservice/DevOps & Multi-access Edge Computing

MEC System 특징

- MEC platform은 많은 function들로 구성됨
 - DNS 관련 기능, Radio Network Information 기능, Location 정보기능, Bandwidth 관리 기능 등
- 이동통신Network 정보제공범위 확대 혹은 개선에 따라 신규 Service 추가, Upgrade
 - Monolithic 구조는 일부 기능 변경 시 전체 Application에 대한 re-writing and redeploying 필요
- 3 Location 구조(Terminal, Edge Cloud, Remote Cloud)를 기반으로 상황에 따른 자유로운 배포 필요
- ME App.은 환경에 따라 선택적으로 서비스 사용
- Application 개발사와 분리된 전문 서비스 공급자
- 국가 기간 망이므로 강력한 보안 구조 필요



Microservice Architecture

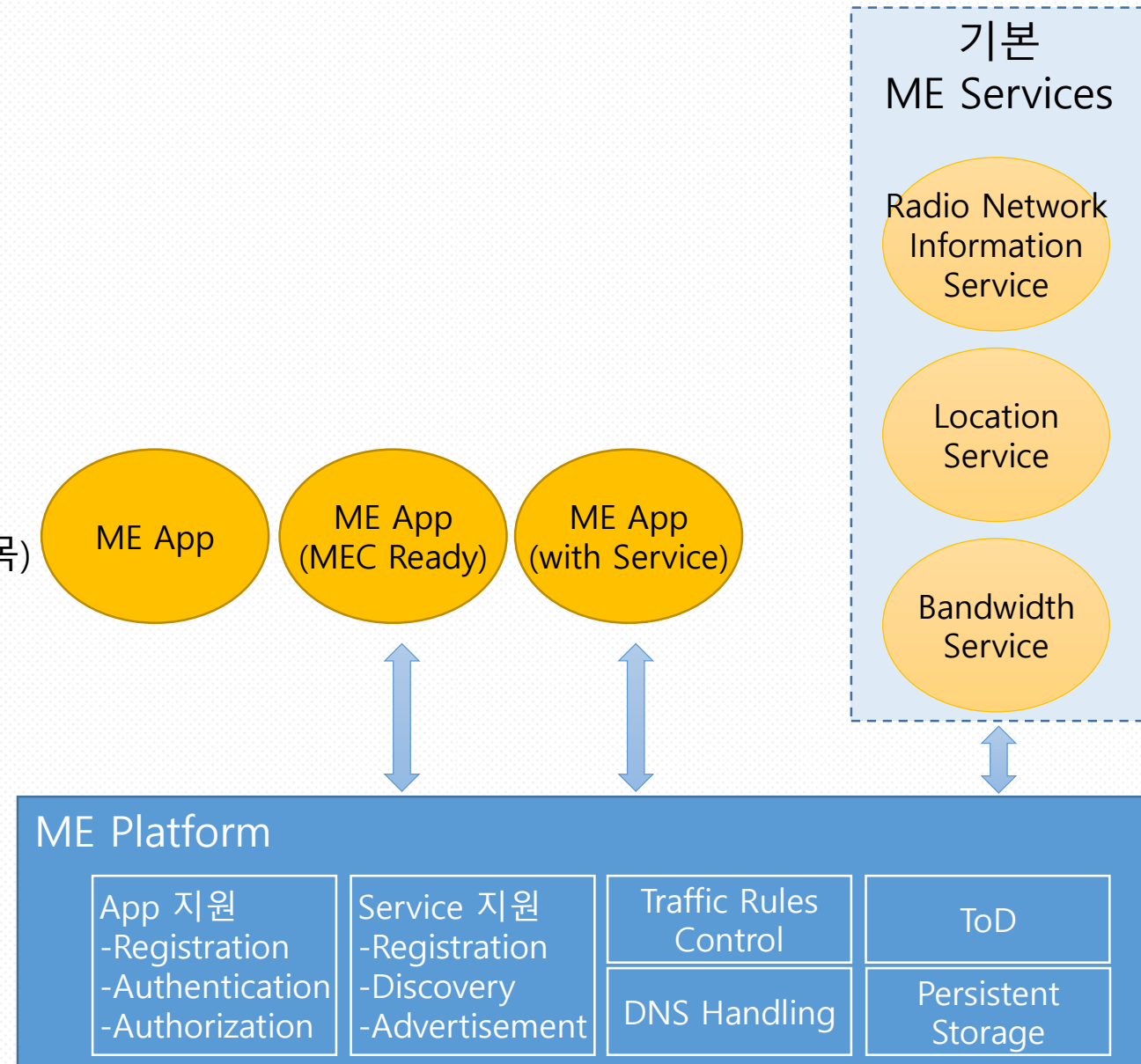
- **Independent Development**
- **Independent Deployment**
- **Fault Isolation**
- **Mixed Technology Stack**
- **Granular Scaling**

DevOps

- MEC는 새롭고 혁신적인 서비스를 위한 ecosystem으로 발전 중
- DevOps는 새로운 business value를 만들기 위한 좋은 방법론
- MEC ISG는 DevOps를 지원하기 위해 conformance and interoperability test를 위한 규격 정의 중

Functions of ME Platform

- ME Application 지원
 - ME App Registration/Authentication/Authorization/Lifecycle
 - DNS/Persistent storage/Time of Day Service 제공
- ME Service 지원
 - ME Service Registration/Discovery/Advertisement
 - 타 ME Platform Service 연동
- 기본 ME Services Hosting (MEC Technical Requirements 항목)
 - Radio Network Information/Location/ Bandwidth Service
- ME Application Relocation
 - Session state relocation support procedures
- Data Plane 관리 (optional)
 - Traffic Rule에 기반한 Data Plane 설정 & Routing



기본 Mobile Edge Services

Service	Description
Radio Network Information Service(RNIS)	Network의 무선 구간에 대한 정보 제공 <ul style="list-style-type: none">• 무선 구간 상태 정보 제공• User plane 품질 측정, 통계 정보 제공• 현 Edge Cloud에 속하는 무선 Network에 접속되어 있는 User Device 정보 및 정보의 변경사항
Location Service	위치 관련 정보 제공 <ul style="list-style-type: none">• 특정 (종류의) User Device or 전체 User Device의 위치 정보• 특정 영역에 위치한 User Device의 list• 특정 영역에 진입/진출 한 User Device 정보• 모든 radio nodes 자체의 위치 정보• User Device 위치 통계 정보
Bandwidth Management service (BWMS)	특정 ME app.으로(으로부터) 전송되는 Traffic에 대한 <ul style="list-style-type: none">• Bandwidth 설정 기능• Priority 할당 기능

MEC Key Concepts

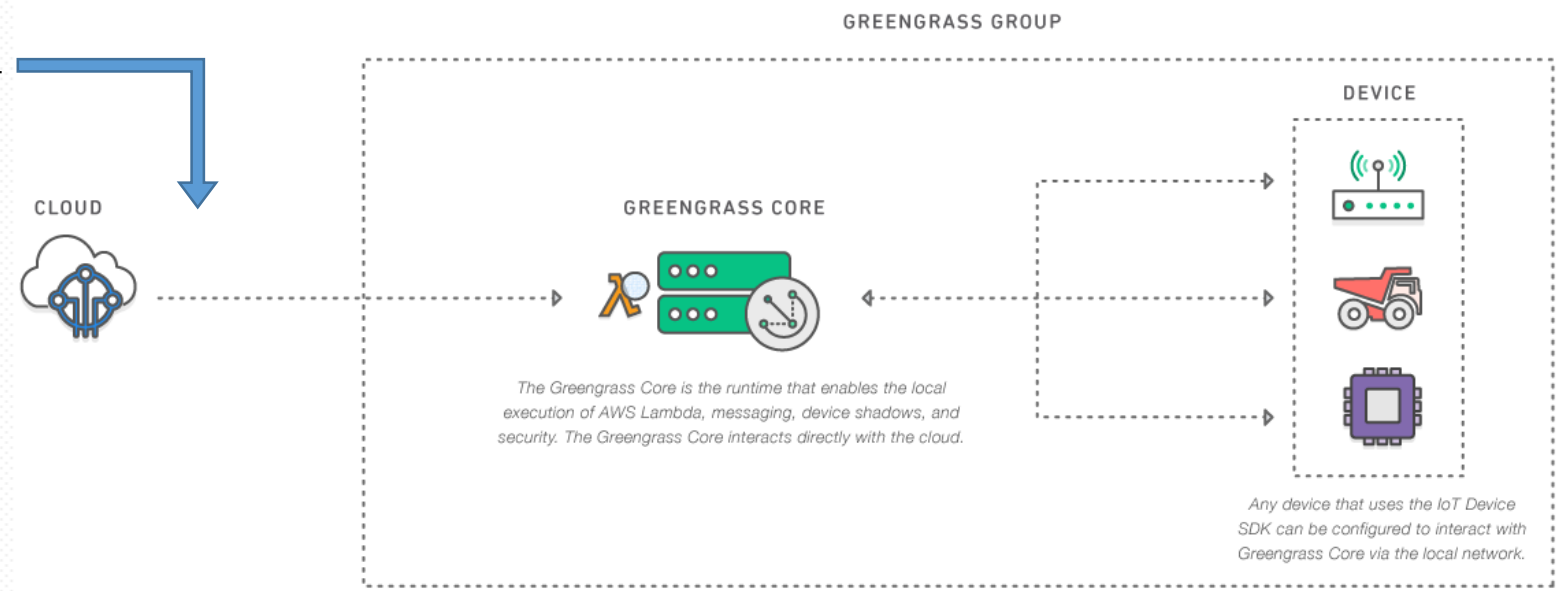
- Mobile Edge Host Selection
 - ME App.에 가장 적합한 Host를 선택하는 기능
 - ME orchestrator에서 OSS, a third-party, or a UE application의 요청을 받아 ME App.을 배포할 때 수행
 - ME Host Selection을 위해 사용되는 정보
 - 필요한 가상 자원 요구사항
 - Location 관련 요구사항 – 지리적인 배포 위치
 - Latency 관련 요구사항 – Latency 최대 한계
 - 필요한 ME Service List (필수 Service, 선택적인 Service)
 - Relocation 요구사항 (application state relocation, application instance relocation)
 - Network 연결 요구사항 (application 연결, local network 연결 or Internet 연결)
 - 필수 ME features List(VM relocation support or UE identity)
 - 영구 storage 이용에 관한 요구사항
 - Application 배포 형태 – 사용자(or Host) 당 하나의 Instance
 - 통신사의 ME System 배포 혹은 이동통신 Network 배포 현황 정보(topology, cost)
 - user traffic 접근방식에 대한 요구사항
- DNS support
 - ME Platform은 DNS 기능(a name server and a proxy/cache function) 제공
 - 사전 설정 or ME app.의 요청 시 제공된 DNS rules에 따라 IP/FQDN mapping table 구축

MEC Key Concepts

- Traffic Routing & Filtering
 - Traffic Rule에 따른 사용자 Traffic Routing(to ME applications/Enterprise Network/Internet) or Filtering 기능
 - Use case. 기업용 ME app.(ERP 등)은 Traffic rule을 활용하여 특정 Traffic을 Enterprise network으로 redirect
- UE mobility 지원
 - 단순 UE Traffic 모니터링/분석 App. 혹은 UE State를 신속히 재 생성 할 수 있는 App. 등은 UE mobility 불 필요
 - UE의 이동으로 서비스 거리가 증가하여 허용 Latency 한계를 넘거나 Network congestion 등이 발생한 경우 UE mobility 활용
 - 방안 1: 이동 지역의 ME Host에 설치된 ME App이 서비스 지원
 - 방안 2: Application state relocation
 - Source ME App과 Target ME App간 연동을 통해 UE State 정보 전송 방식 (ME app 기능 지원 필요)
 - 방안 3: Application instance relocation (VM handoff)
 - Source ME App instance 자체를 Target ME Host로 전송하는 방식

MEC와 타 Edge Computing Service

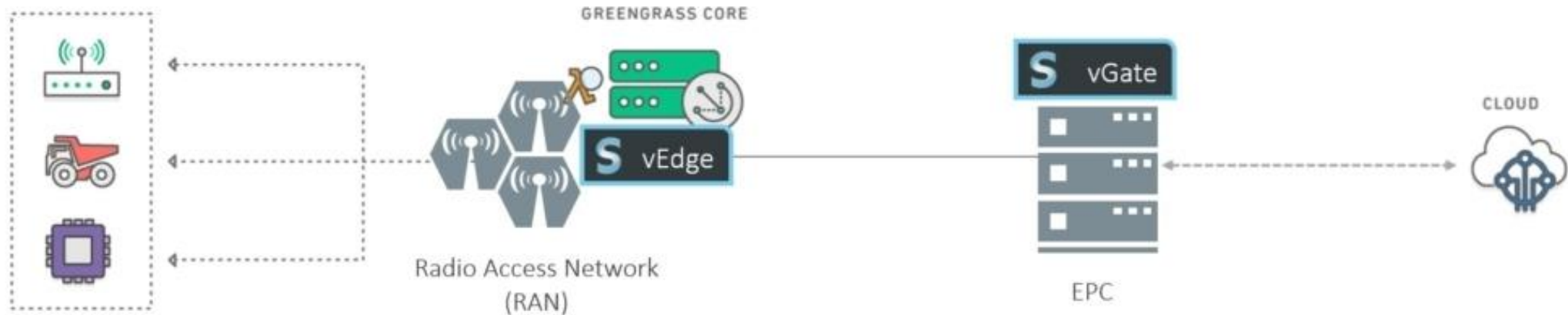
- Greengrass for Amazon's AWS Lambda
- Microsoft's Azure IoT stack
- GE's Predix



타 Edge Computing 대비 MEC 장점

- extreme user proximity, ultra low latency, high bandwidth, real time access to radio network/context information, location awareness

Greengrass on MEC 협력 Saguna Networks



Thank you |



Cloud Computing Expert Group
www.cloudtek.co.kr

정정문 (jmjung@sptek.co.kr)