

Spring 2023



소프트웨어 아키텍처 패턴: Edge-based

Seonah Lee

Gyeongsang National University

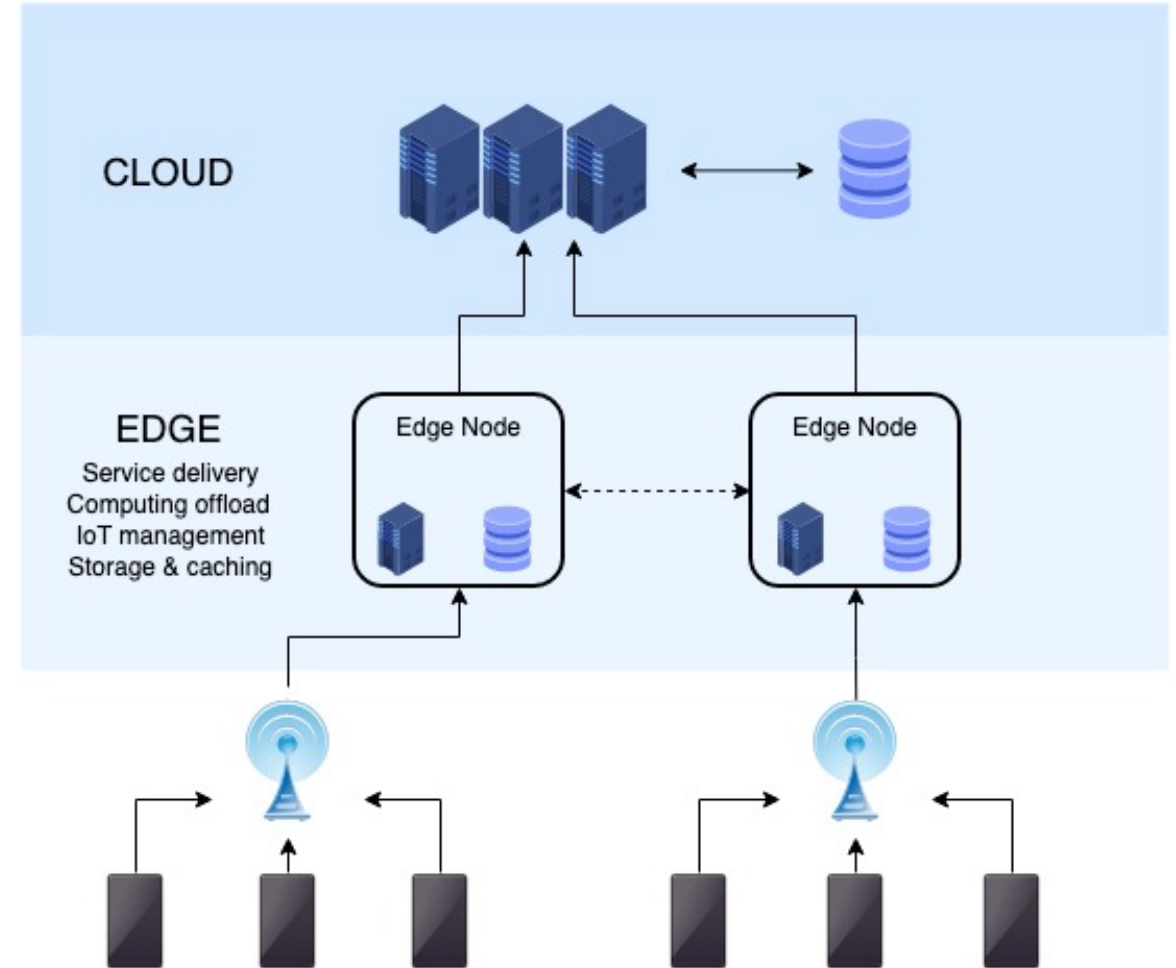
Edge-based 패턴

- ▶ 패턴 정의
- ▶ 패턴 예제
- ▶ 패턴 설명
- ▶ 패턴 컴포넌트, 구조 및 행위
- ▶ 패턴 구현
- ▶ 패턴 코드
- ▶ 패턴 장단점

Edge-based Pattern: Definition

▶ 정의

- ▶ 컴퓨팅 및 데이터 스토리지를 데이터 소스에 더 가깝게 가져오는 분산 컴퓨팅의 패러다임
- ▶ 응답 시간을 개선하고 대역폭을 절약하기 위해서 데이터 소스에 가까운 곳에 에지 컴퓨팅을 배치



Edge-based Pattern: Example

▶ 현대적인 제조 공장

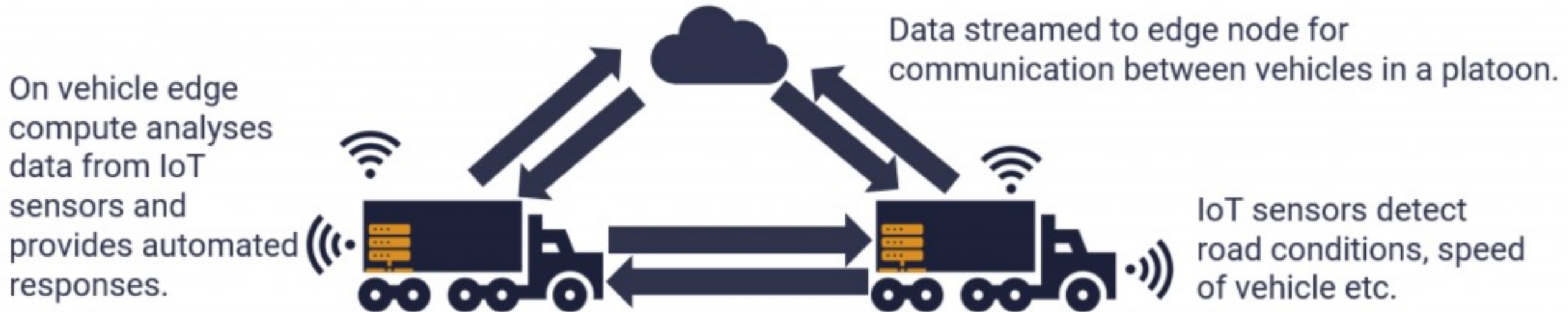
- ▶ 사물 인터넷(IoT) 센서가 가동 중단을 방지하고 운영을 개선하는데 사용할 수 있는 지속적인 데이터 스트림 생성
 - ▶ 2,000대의 장비 보유, 매월 2,200테라바이트 데이터 생성



Edge-based Pattern: Example

▶ Autonomous vehicles

- ▶ With edge computing, it will be possible to remove the need for drivers in all trucks except the front one
- ▶ The trucks will be able to communicate with each other with ultra-low latency

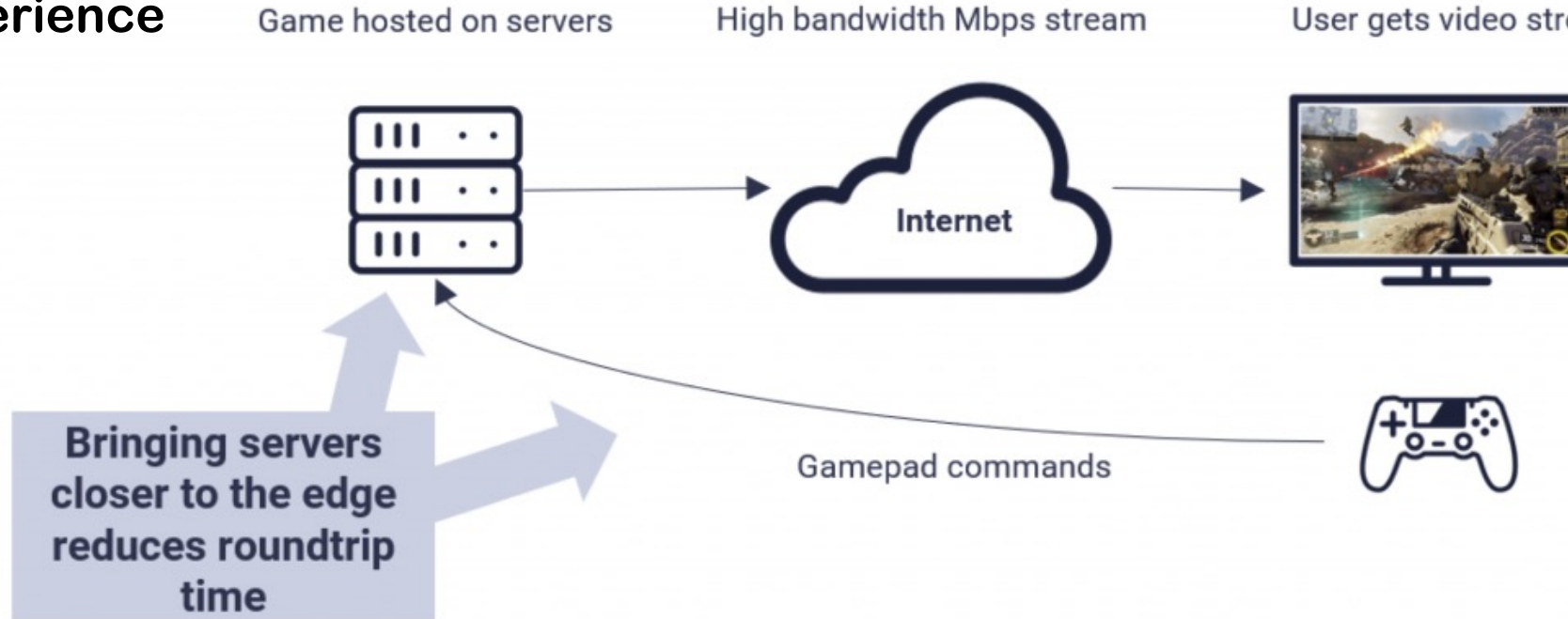


Source: STL Partners

Edge-based Pattern: Example

► Cloud gaming

- Cloud gaming companies are looking to build edge servers as close to gamers
- It is to reduce latency and provide a fully responsive and immersive gaming experience



Source: STL Partners

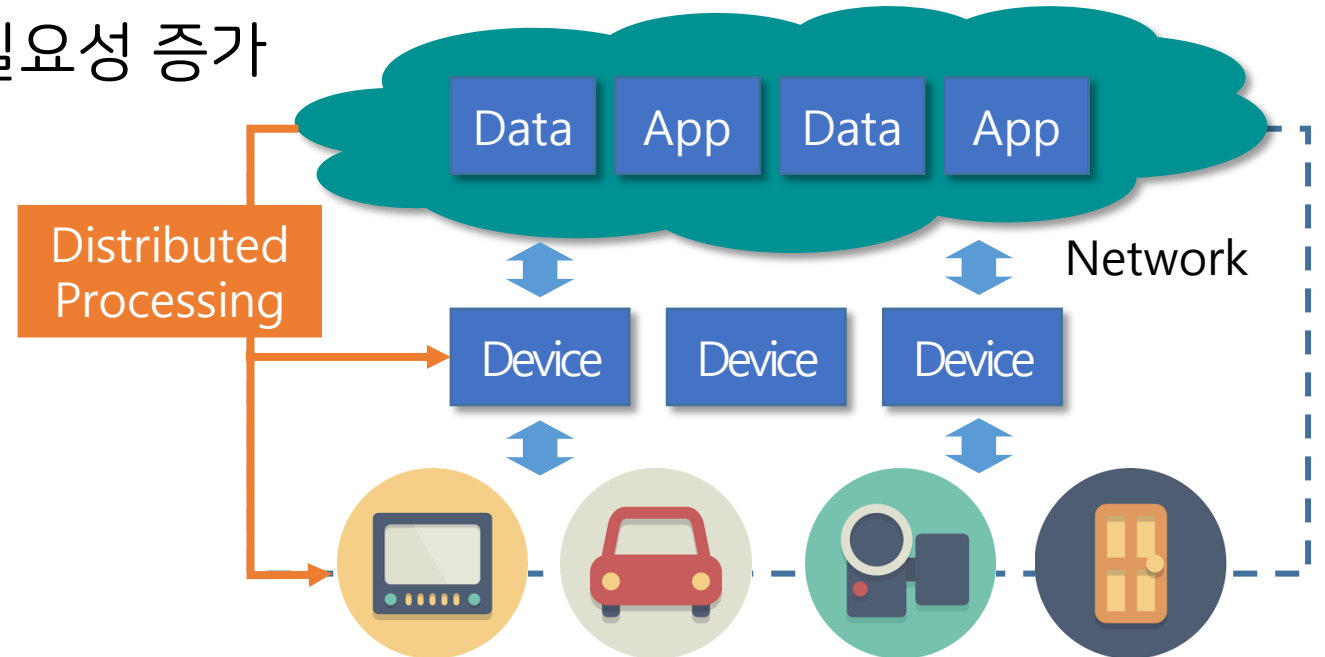
Edge-based Pattern: Example

- ▶ **Autonomous vehicles***
- ▶ Remote monitoring of assets in the oil and gas industry
- ▶ Smart grid
- ▶ Predictive maintenance
- ▶ In-hospital patient monitoring
- ▶ Virtualized radio networks and 5G (vRAN)
- ▶ **Cloud gaming***
- ▶ Content delivery
- ▶ Traffic management
- ▶ Smart homes

Edge-based Pattern: Description

▶ 정황 (Context)

- ▶ 종단 장치들의 증가로 인해 전송되는 데이터 양이 증가 (데이터 병목현상)
- ▶ 대용량의 컴퓨팅 리소스를 요구, 통신 지연에 따른 데이터 처리 지연
- ▶ 처리 응답의 지연, 실시간 대응 필요성 증가



Edge-based Pattern: Description

▶ 문제 (Problem)

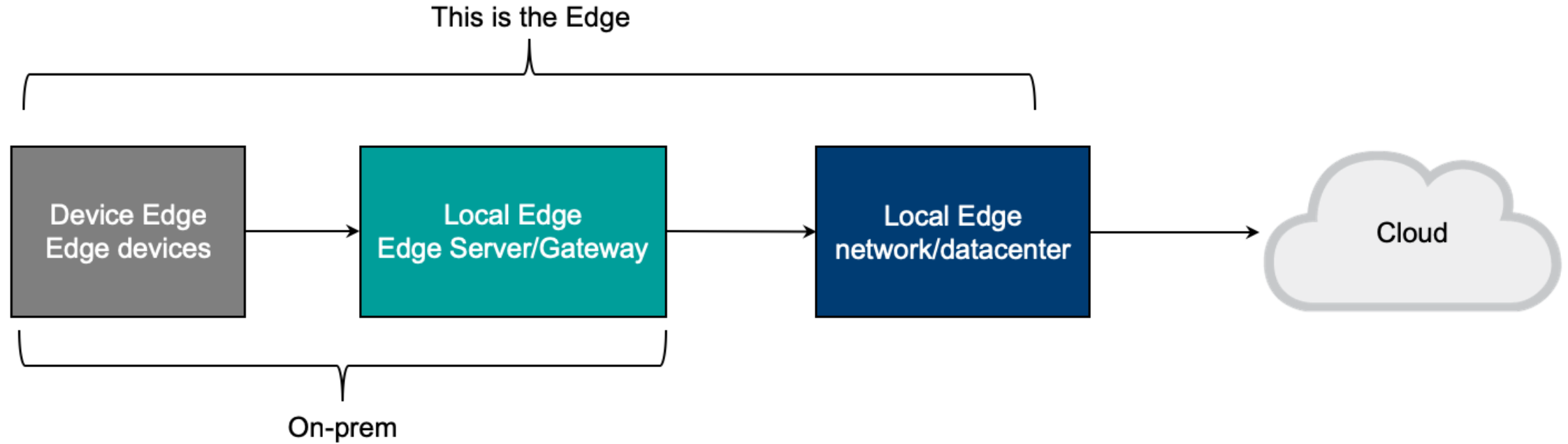
- ▶ 네트워크 **Edge**에서의 **IoT** 기기의 증가로 인해 데이터 센터에서 계산할 엄청난 양의 데이터가 생성되어 네트워크 대역폭 요구가 한계에 도달
- ▶ 또한 방대한 데이터의 처리를 클라우드 컴퓨팅 만으로 처리 불가
- ▶ 대역폭 및 스토리지 리소스의 절약 필요, 데이터 병목현상 감소 필요
- ▶ 실시간 대응 필요성, 보안성 필요

Edge-based Pattern: Description

▶ 해법 (Solution)

- ▶ 데이터의 일부가 네트워크 가장자리에서 처리되는 새로운 패러다임
- ▶ 최종 사용자와 클라우드 간의 중간 계층으로 운영되면서 응답 시간의 최소 지연으로 클라우드의 리소스와 서비스를 네트워크 가장자리로 가져오는 것
- ▶ 컴퓨팅을 데이터 센터에서 네트워크 **Edge**로 이동. 스마트 기기, 휴대폰 또는 네트워크 게이트웨이 등을 활용하여 클라우드를 대신하여 작업을 수행, 서비스 제공
 - ▶ 클라우드 컴퓨팅: 빅데이터를 처리함
 - ▶ 에지 컴퓨팅: 요청에 대해서 더 낮은 지연 시간으로 처리를 제공하는 모든 유형의 컴퓨터 프로그램을 의미함. 센서나 사용자가 생성하는 실시간 데이터인 “인스턴스 데이터”에 작동함

Edge-based Pattern: Description



Device edge	where the edge devices sit
Local edge	which includes both the infrastructure to support the application and also the network workloads
Cloud	or the nexus of your environment, where everything comes together that needs to come together

Edge-based Pattern: Realization

▶ Business solution layer

- ▶ 클라우드 관리, 인증, 분석

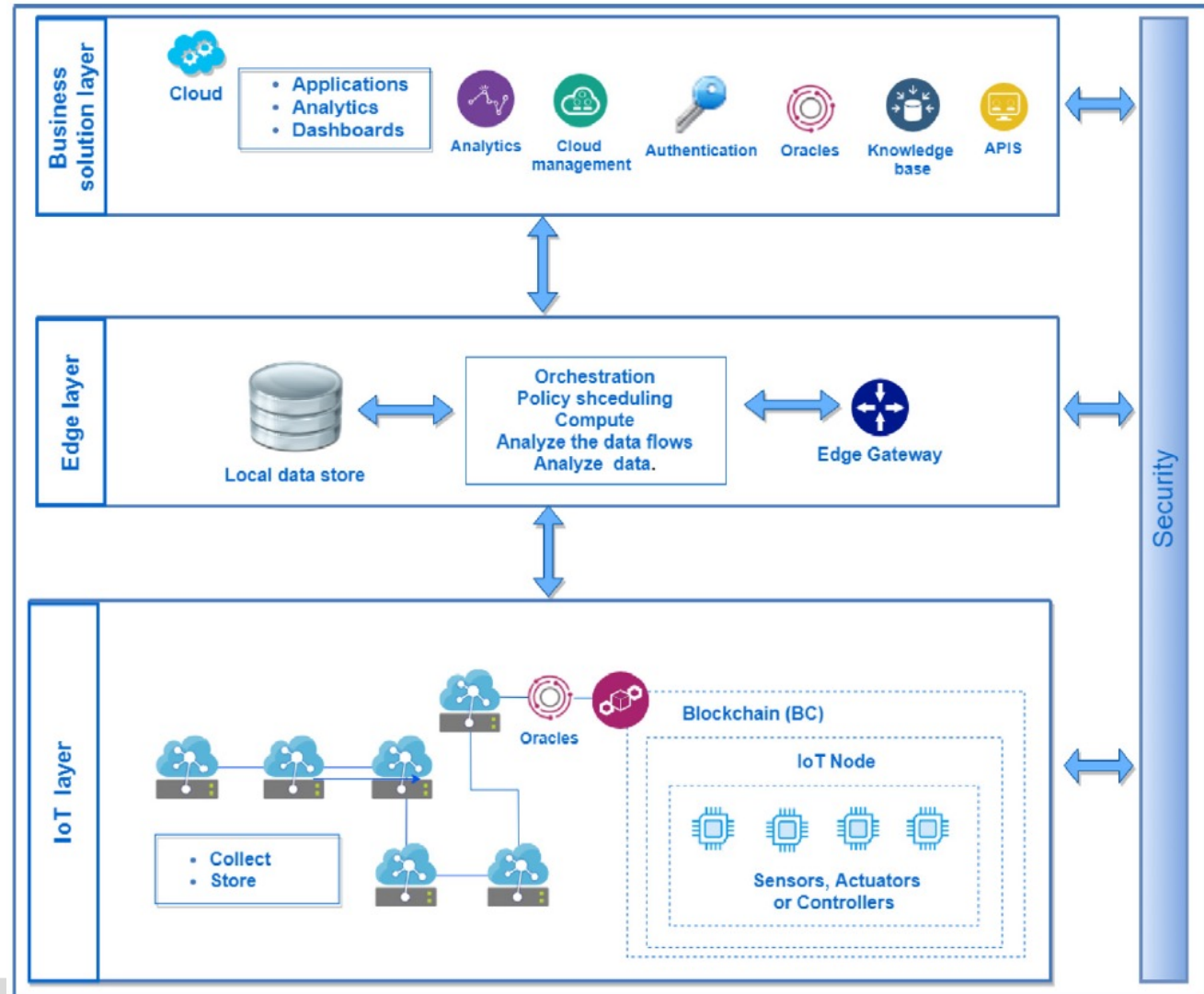
▶ Edge layer

- ▶ 데이터의 필터링 및 사전 처리를 담당
- ▶ **TensorFlowLite libraries**를 사용하여 머신러닝 기법을 적용 (**Data Analytics**)

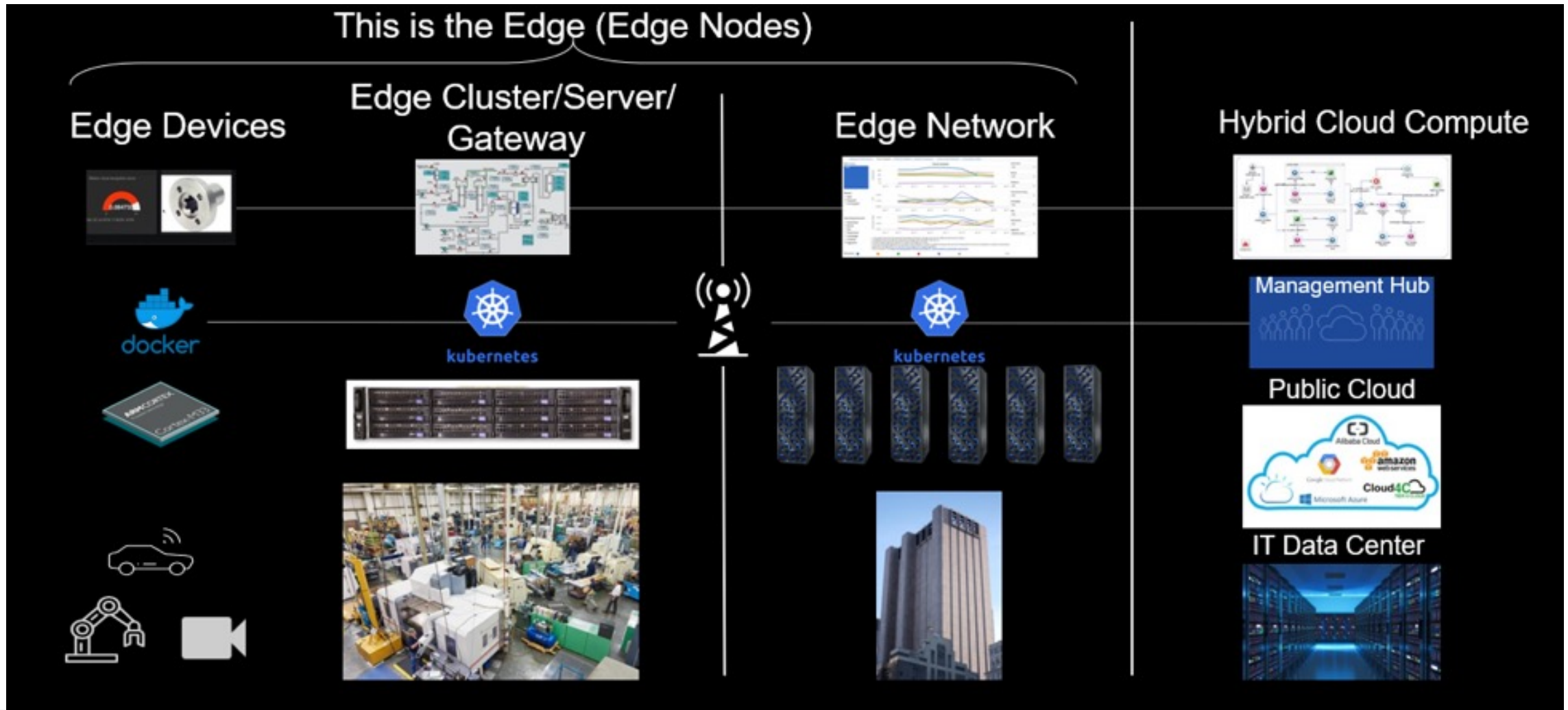
- ▶ **Raspberry Pi**에서 **Node.js** 서버 활용

▶ IoT layer

- ▶ IoT 장치의 계산 리소스를 관리하고 저장
- ▶ **SHA-256** 및 **RSA** 알고리즘과 함께 기본적인 블록체인을 사용



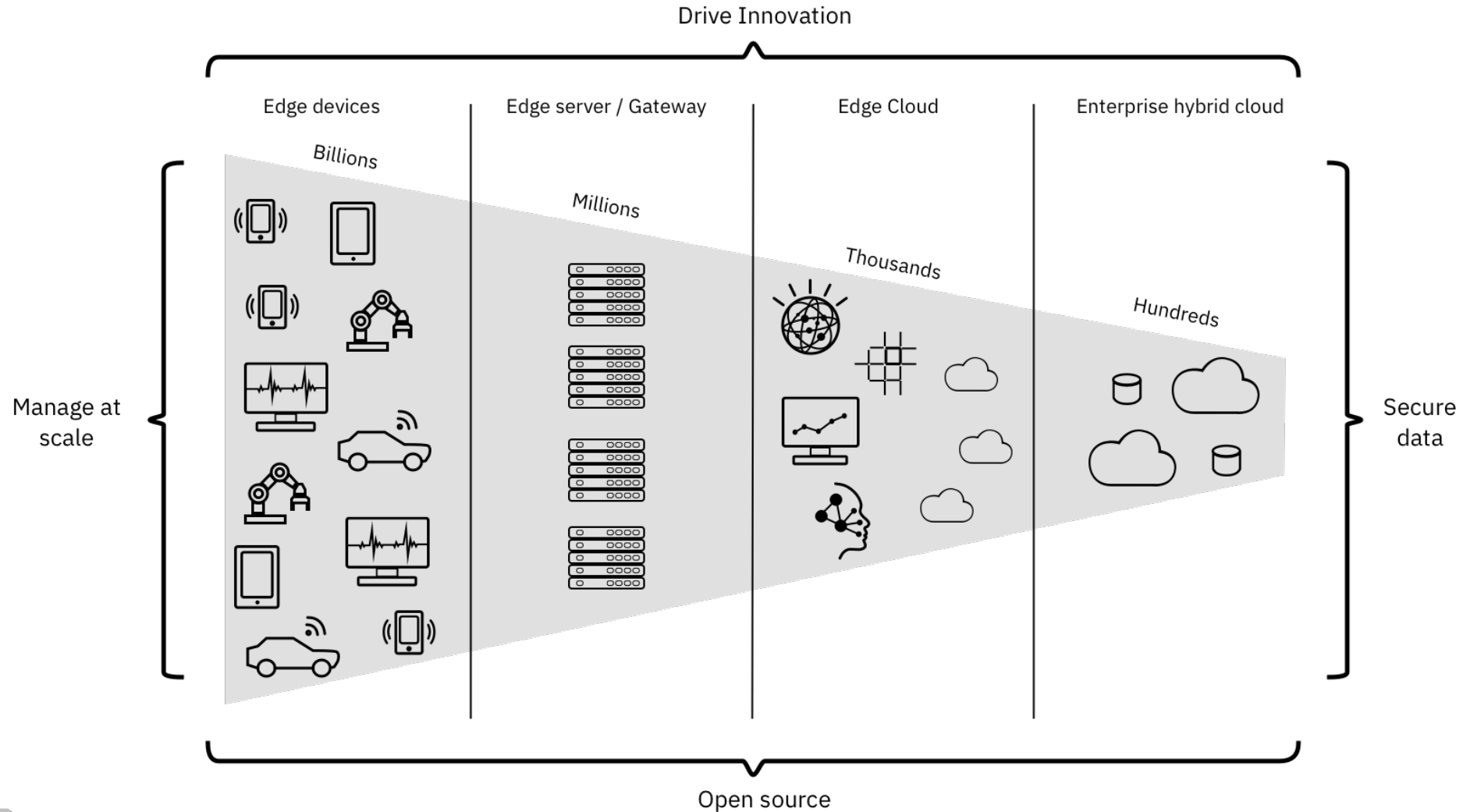
Edge-based Pattern: Realization



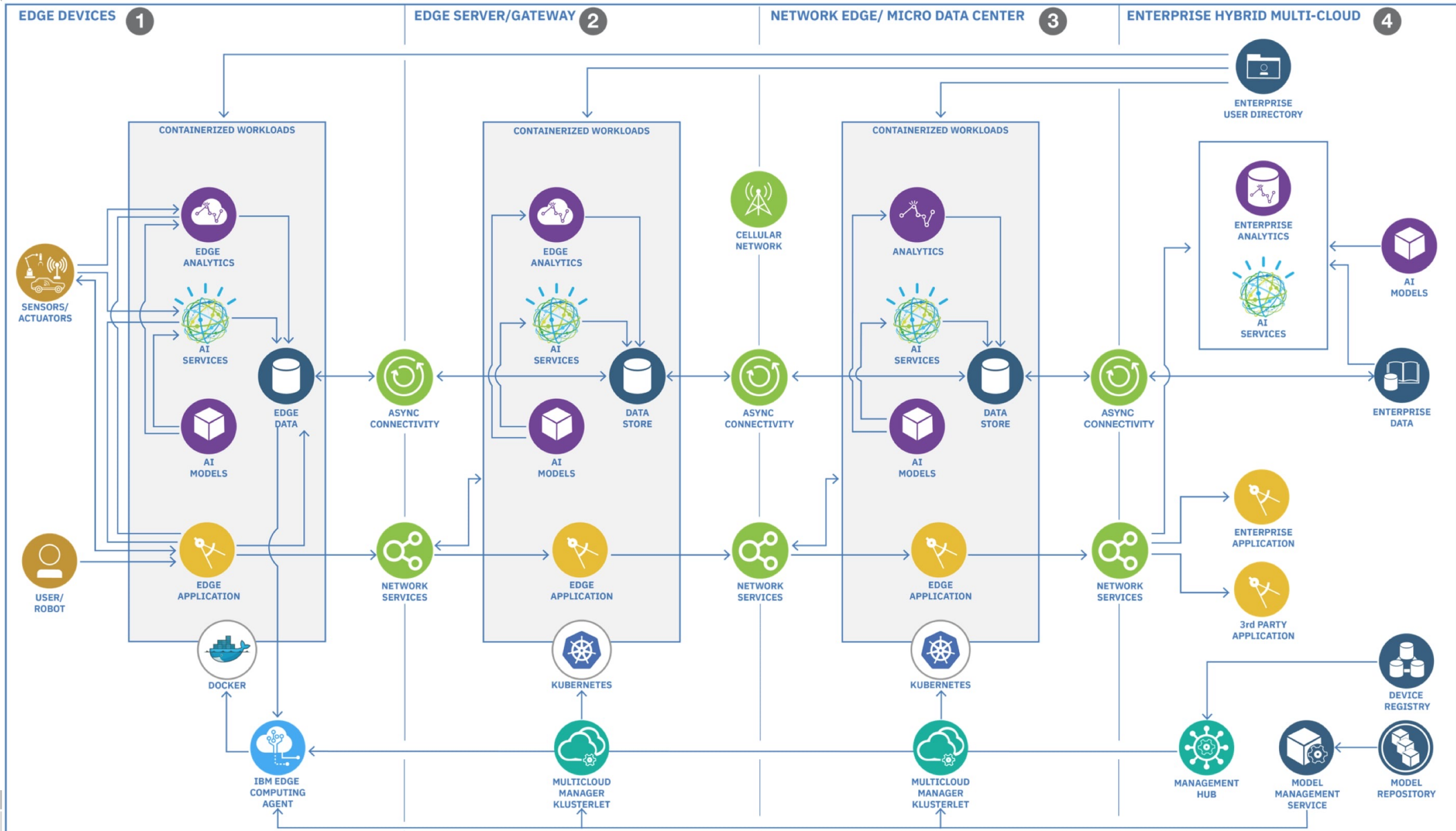
Edge-based Pattern: Realization

Components	Description
Cloud	These clouds also host and run the applications that are used to orchestrate and manage the different edge nodes.
Edge device	An edge device is a special-purpose piece of equipment that also has compute capacity that is integrated into that device.
Edge cluster/server	An edge cluster/server is a general-purpose IT computer that is located in a remote operations facility such as a factory, retail store, hotel, distribution center, or bank.
Edge gateway	An edge gateway is typically an edge cluster/server which, in addition to being able to host enterprise application workloads and shared services, also has services that perform network functions such as protocol translation, network termination, tunneling, firewall protection, or wireless connection
Edge node	An edge node is a generic way of referring to any edge device, edge server, or edge gateway on which edge computing can be performed.

Edge-based Pattern: Case Studies



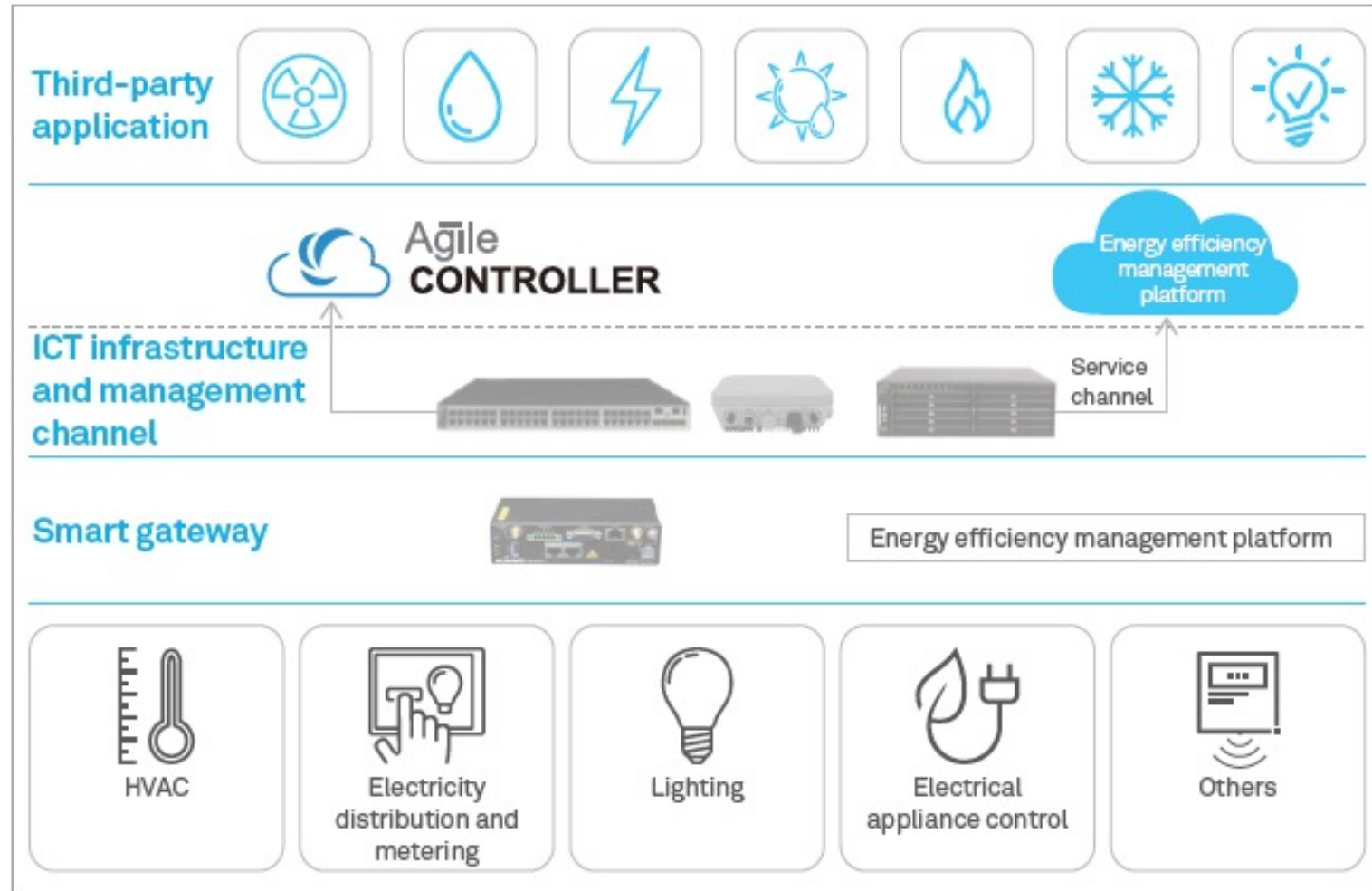
Edge-based Pattern: Case Studies



Edge-based Pattern: Case Studies

▶ Smart Building

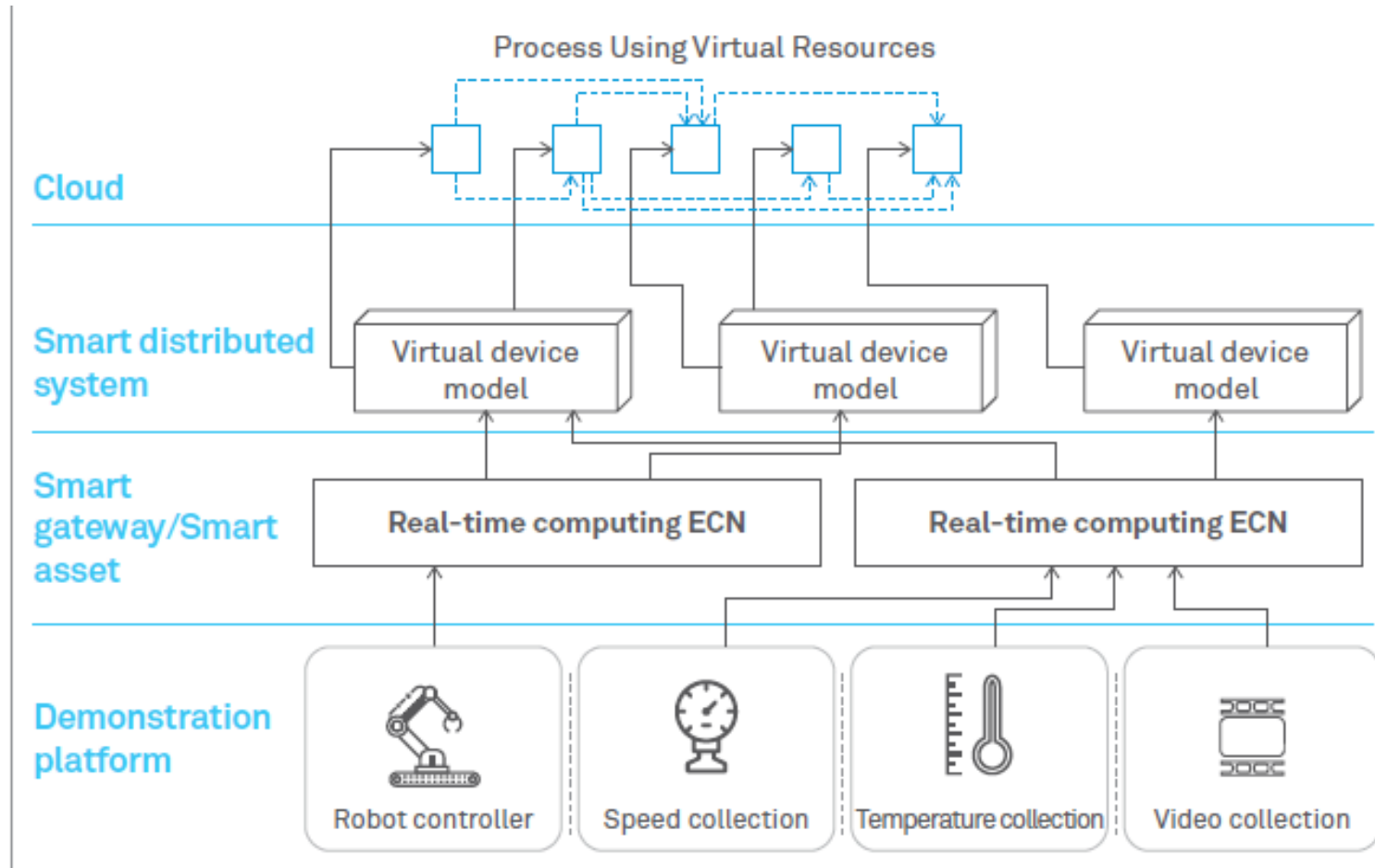
- ▶ 빌딩의 센서로부터 데이터를 모아 장비의 상태, 오류 발견, 예방적인 유지보수
- ▶ **Smart gateway:** 네트워크 연결 단순화를 위한 인터페이스 제공
- ▶ **Platform:** 실시간 대응 및 다양한 장치와의 지속 연결 제공
- ▶ **Agile Controller:** 애플리케이션 개발 및 효율 증가에 기여
- ▶ **Security:** 전반적인 보안, 데이터 전송, 시스템 보안이 다양한 수준으로 제공되어야 함



Edge-based Pattern: Case Studies

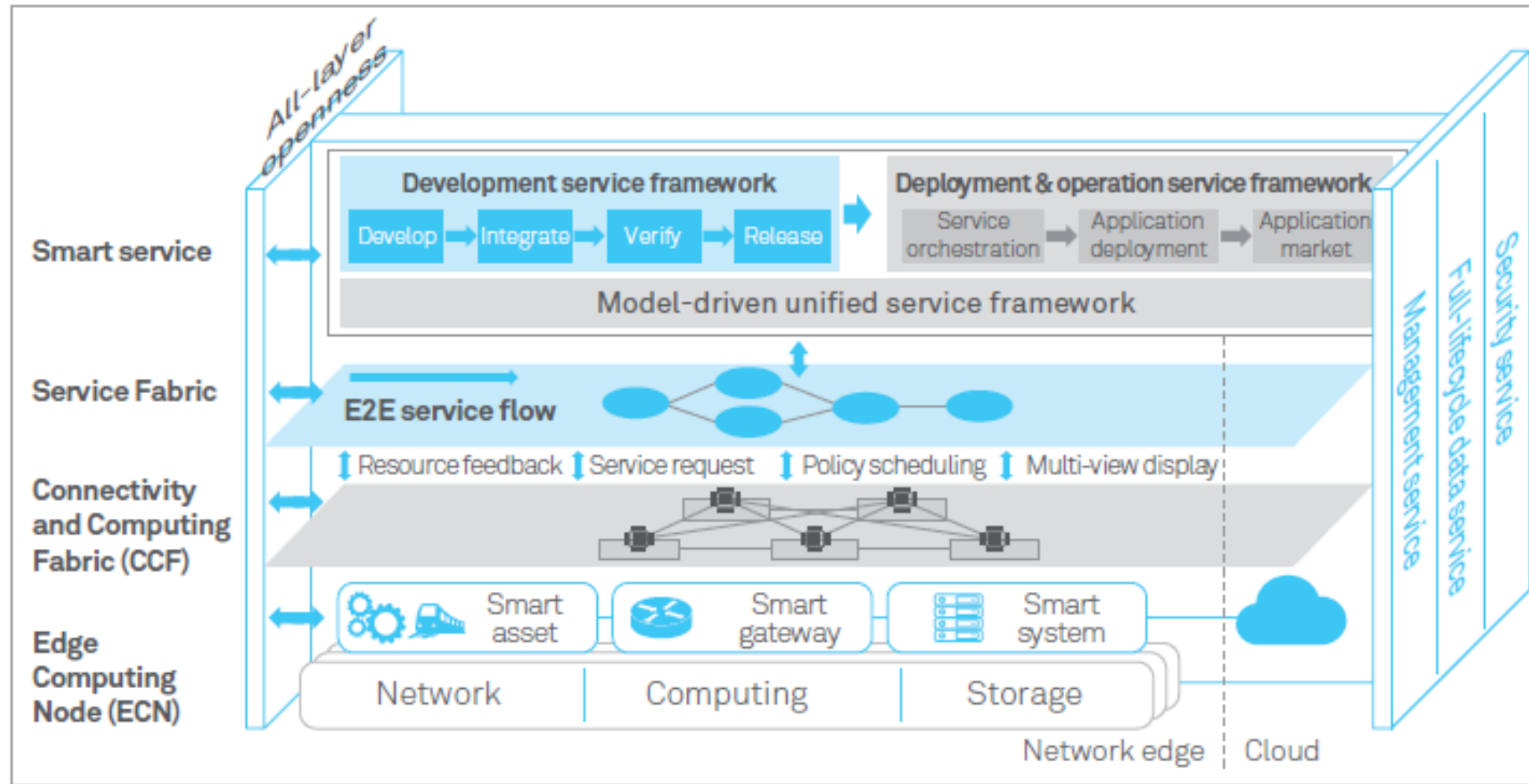
▶ Smart Manufacturing

- ▶ 서비스 조정 및 가시화를 위해 생산 장비를 스마트 분산 시스템에 연결하여 모니터링
- ▶ **Digital Twins**의 개념으로 디지털 장비 모델을 셋업. 디바이스에 연결하여 가상화, 모델링, 상호 관계 및 검색 수행
- ▶ **SDN** 기술을 활용하여 네트워크 자원을 상황에 맞게 적용. 기기를 위한 정보 전송에 대한 효과적인 수단 제공

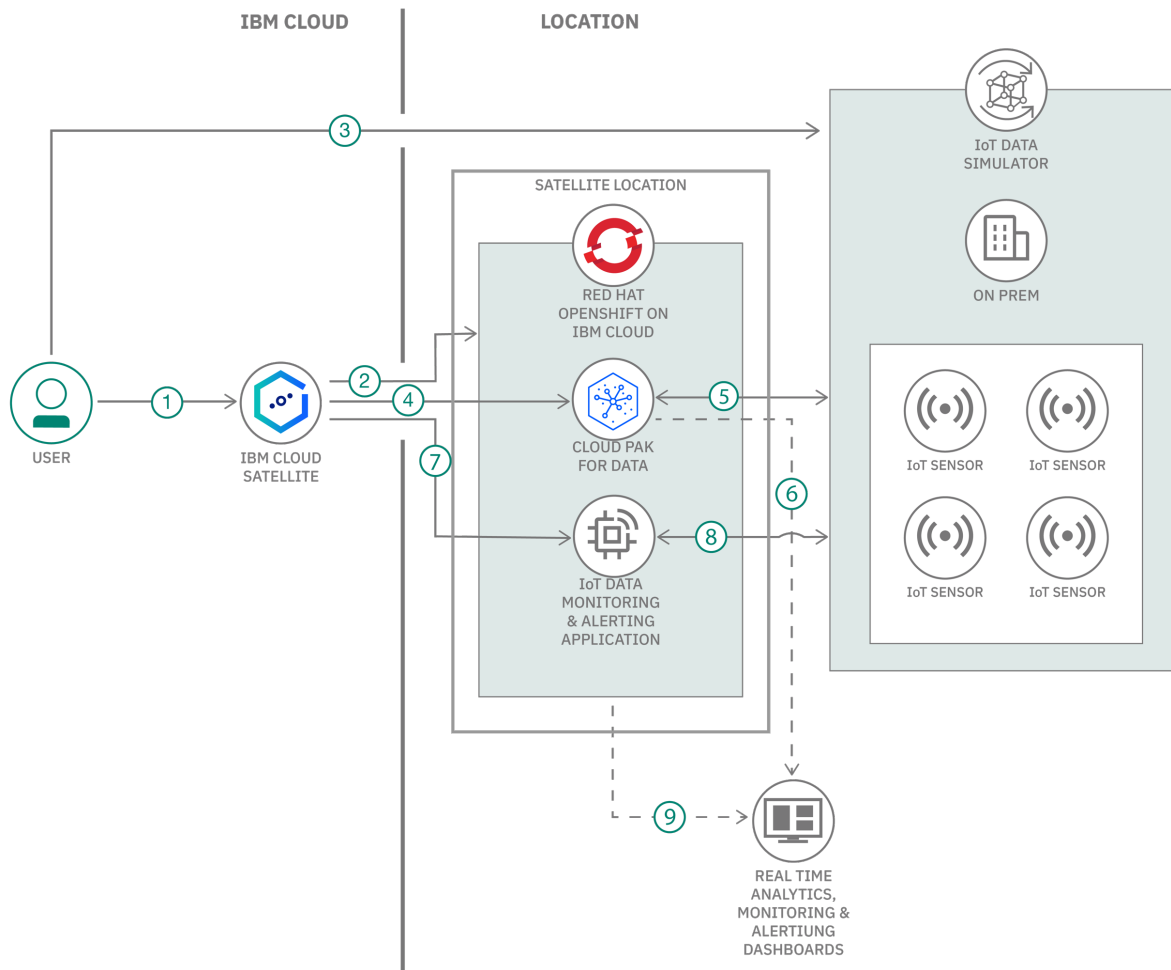


Edge-based Pattern: Case Studies

Figure 3-1 Edge computing reference architecture 2.0

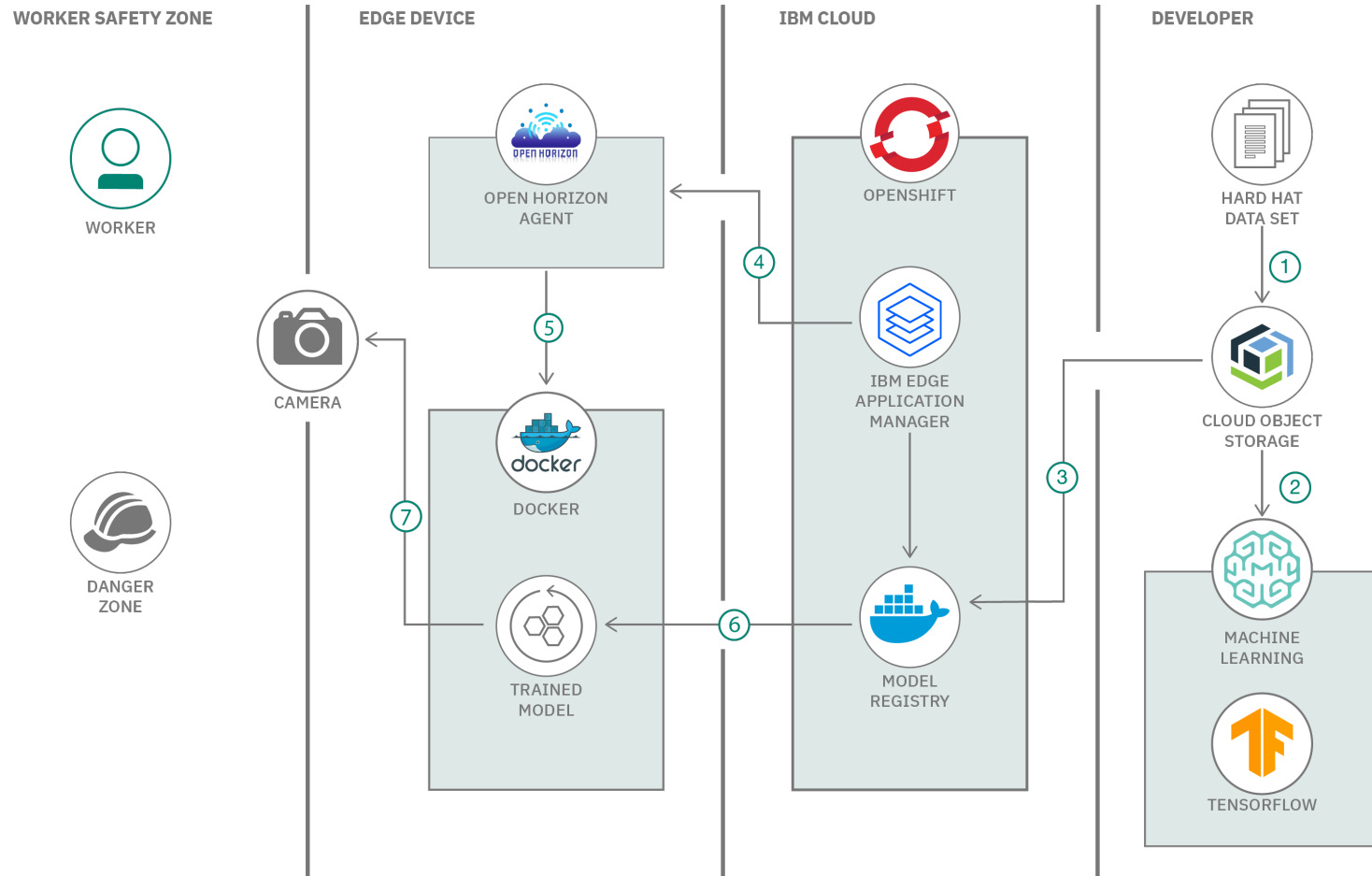


Edge-based Pattern: Case Studies



1. Create an IBM Cloud Satellite location and assign hosts.
2. Create the Red Hat OpenShift on IBM Cloud cluster that will reside in your Satellite location.
3. Deploy the IoT simulator application on premises.
4. Install IBM Cloud Pak for Data on the OpenShift cluster.
5. Build a streams flow on IBM Cloud Pak for Data for the incoming IoT data.
6. Visualize incoming IoT data by using the IBM Streams analytics service on IBM Cloud Pak for Data.
7. Deploy the IoT data monitoring and alerting application on the OpenShift cluster.
8. Consume the IoT data from the simulator in real time.
9. Monitor the data in real time by using the dashboard UI.

Edge-based Pattern: Case Studies



1. AI Model developer uploads the training data to IBM Cloud Object Storage.
2. Watson Machine Learning pulls the training data from IBM Cloud Object Storage and trains a model with TensorFlow. The trained model is saved back to IBM Cloud Object Storage.
3. The trained models are containerized using Open Horizon and stored in a model registry.
4. IBM Edge Application Manager orchestrates workloads to edge gateways or directly to edge devices (or both).
5. IBM Edge Application Manager (Open Horizon) Agent invokes and monitors containerized workload.
6. Models are downloaded from the model registry to the edge devices.
7. AI model runs on the edge device to perform object detection and edge prediction.

Edge-based Pattern: Benefits

- ▶ **응답 지연(Latency) 감소**
 - ▶ 최종 사용자에게 가까이에서 계산을 함으로써 실시간 응답을 돕고 지연 시간을 줄인다.
- ▶ **대역폭(Bandwidth) 요구 감소**
 - ▶ 엣지 컴퓨팅의 사용으로 클라이언트 기기 가까이에서 계산을 하는 것은 로컬 네트워크에서의 데이터 전송만 하면 되어 대역폭을 사용을 줄이고 효율성을 높인다.
- ▶ **신뢰성(Reliability) 증가**
 - ▶ 네트워크 병목 현상을 줄여서 승인할 수 없는 인터럽션을 발생할 확률 줄일 수 있다.
- ▶ **데이터 오너십 이동, 프라이버시 증대**
 - ▶ 데이터를 엣지에 두는 것은 데이터의 오너십을 서비스 제공자로부터 최종 사용자로 이동할 수 있도록 돕는다.

Edge-based Pattern: Liabilities

- ▶ 저장매체가 제한이 됨
- ▶ 전원의 소모가 큼



Question?



Seonah Lee
saleese@gmail.com