

Let's make Kubernetes Boring

Digital Infra CO

컨테이너솔루션개발팀장 안재석

Who are we?

2009: University of Texas at Austin, 소프트웨어엔지니어링 박사

2009~2014: KT 차세대클라우드플랫폼 개발팀장

2010: 한국 OpenStack 커뮤니티 설립 (초대 커뮤니티 리더)

2015: SKT 가상화 및 클라우드 기술 연구

2016: OpenStack on Kubernetes 개발 및 연구

2017-18: Airship Project (글로벌 오픈소스SW) 런칭 (w/ AT&T)

2018: LTE 통신망, 네트워크 가상화 환경 상용 적용

2019: Kubernetes (Apps & Services) 개발 및 연구

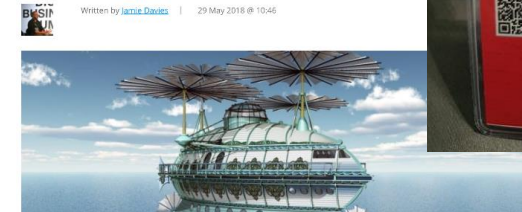
2020: Open Infrastructure Foundation Super User Awards

2020: SKB Btv 상용 적용 (현재, 3개 데이터센터, 200+노드, 2000+ 앱컨테이너)

2021~현재: Enterprise 기업과 Startups/SMB 기업을 위한 컨테이너 서비스 개발



Airship launched by AT&T and SK Telecom



SK Telecom 5GX Cloud Labs wins the 2020 Superuser Awards

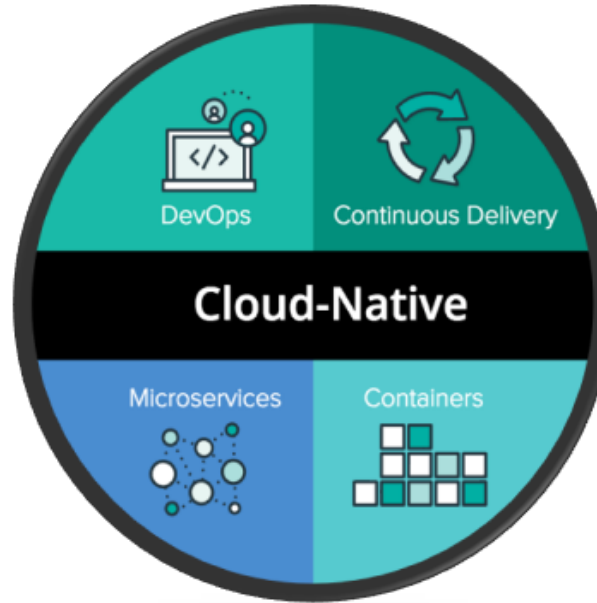
SK Telecom 5GX Labs wins the Superuser Awards for their Airship use case.

October 20, 2020 Ashlee Ferguson

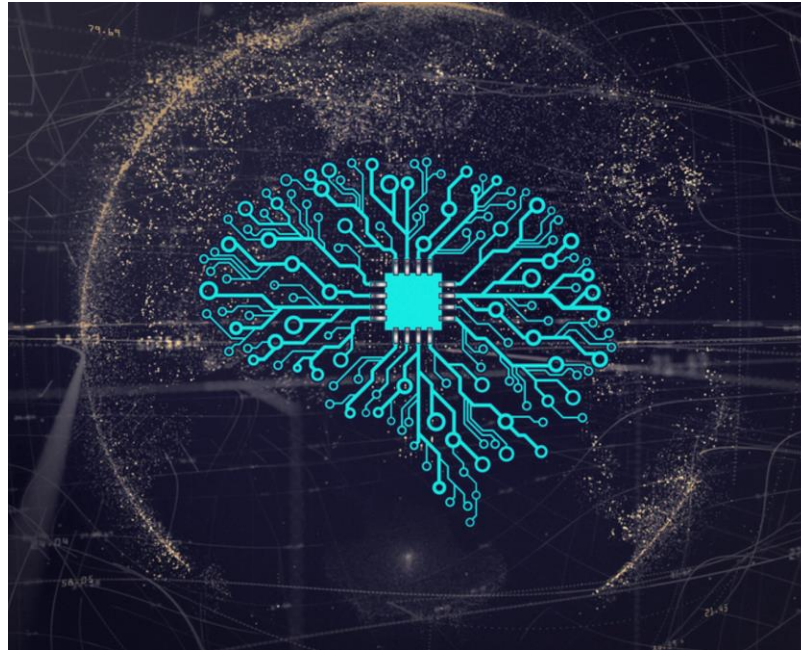


**Kubernetes is
fun
& interesting**

Cloud Native Apps



AI / Machine Learning



Gatner predicts that **cloud native platforms will serve as the **foundation** for more than **95% of new digital initiatives** by 2025. up from less than 40% in 2021**

2022 Kubecon & Cloud Native Con North America 키노트 에서 언급됨

**What you can do with
Kubernetes
is NOT
boring**

Kubernetes 에서 가장 중요한 건

Container Orchestration?

Kubernetes 에서 가장 중요한 건

~~Container Orchestration?~~

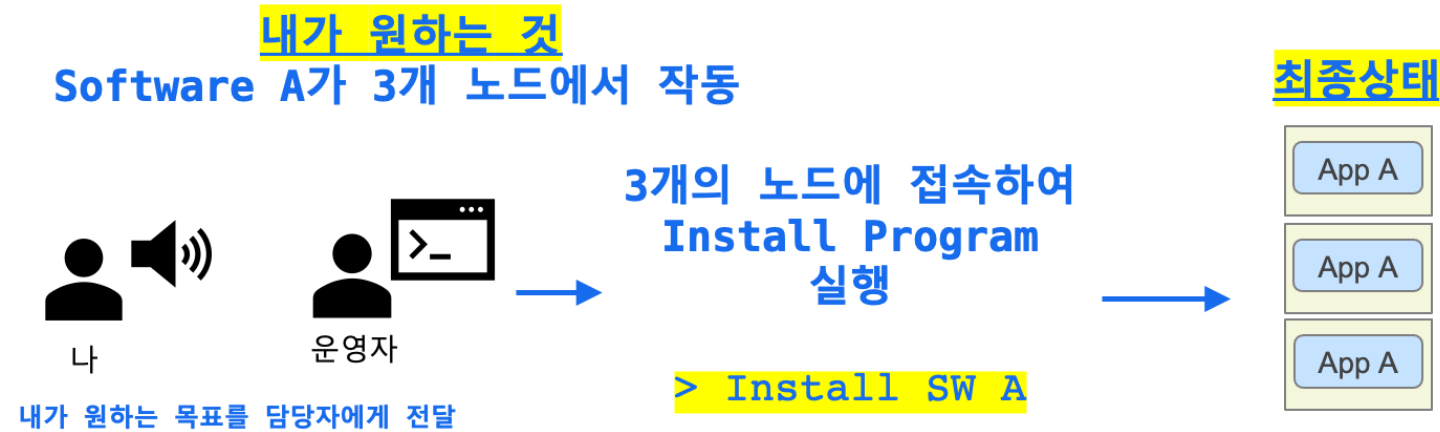
Declarative API



**declarative API들을
관리하는 플랫폼**

**K8s로 가능해진
API Machinery and
Patterns이 가장 오랫동안
살아남을 것이다**

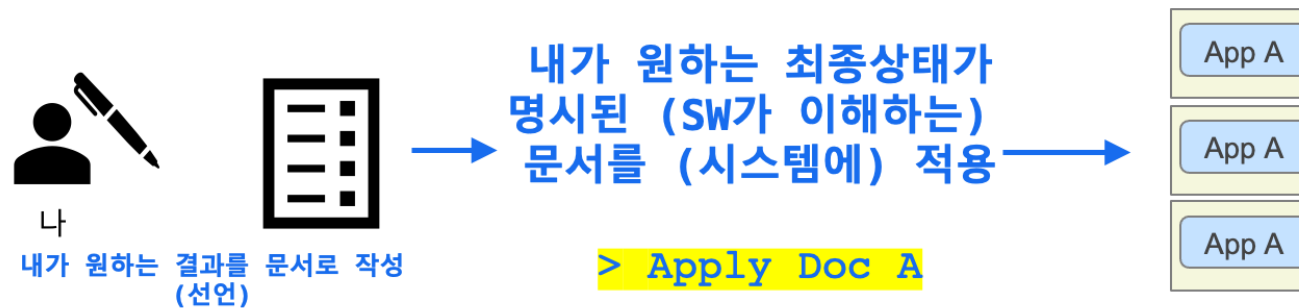
Declarative (선언적) 시스템



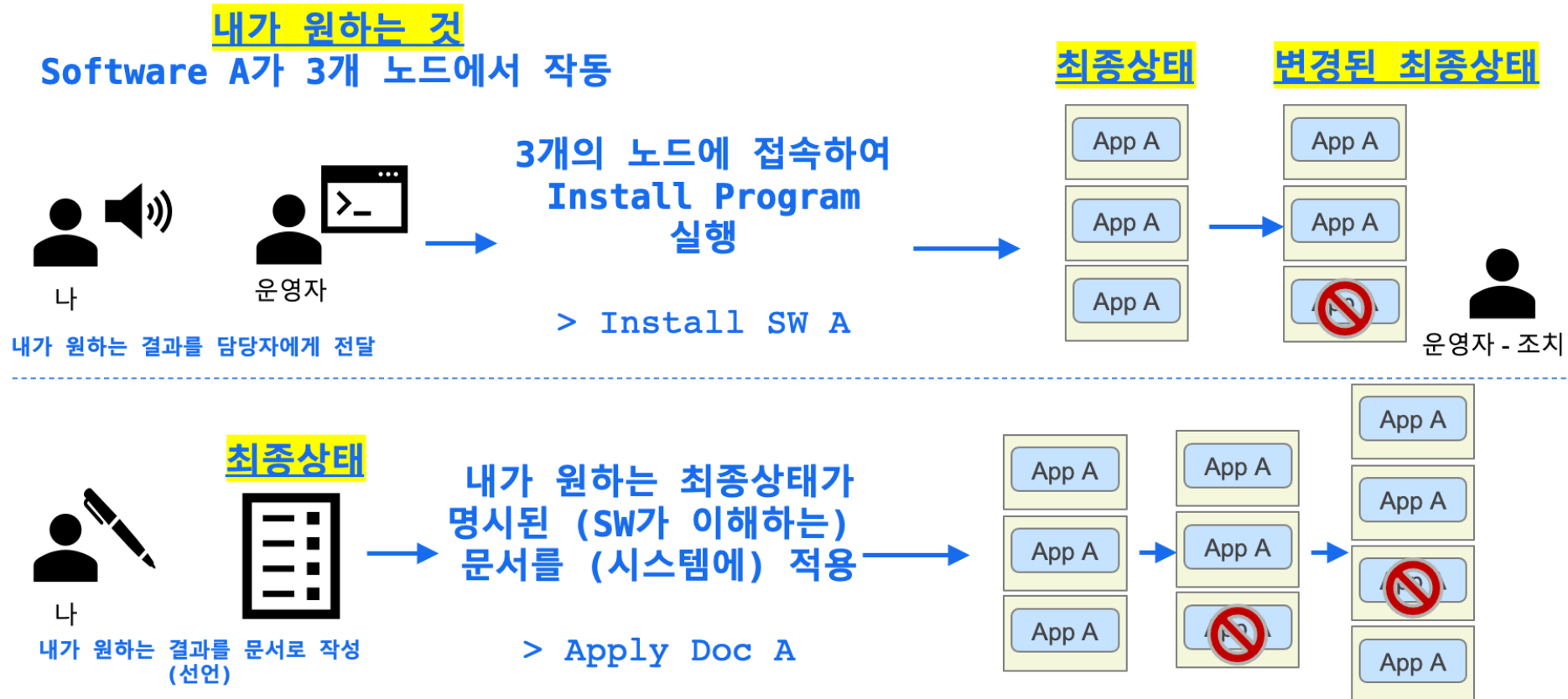
Declarative (선언적) 시스템



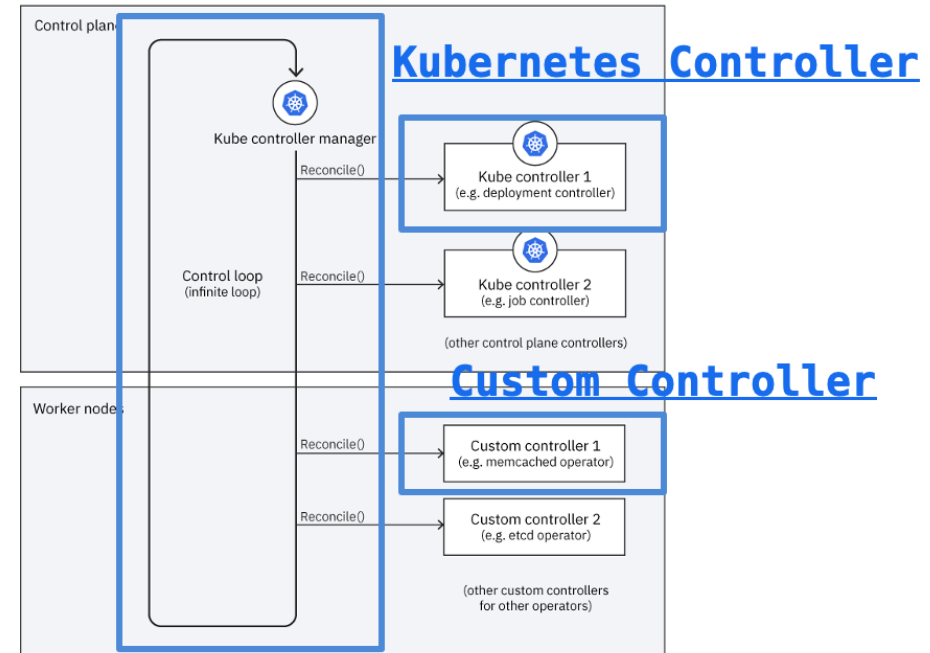
Declarative (선언적) 시스템



Declarative (선언적) 시스템



Everything as Declarative (선언적) Kubernetes API

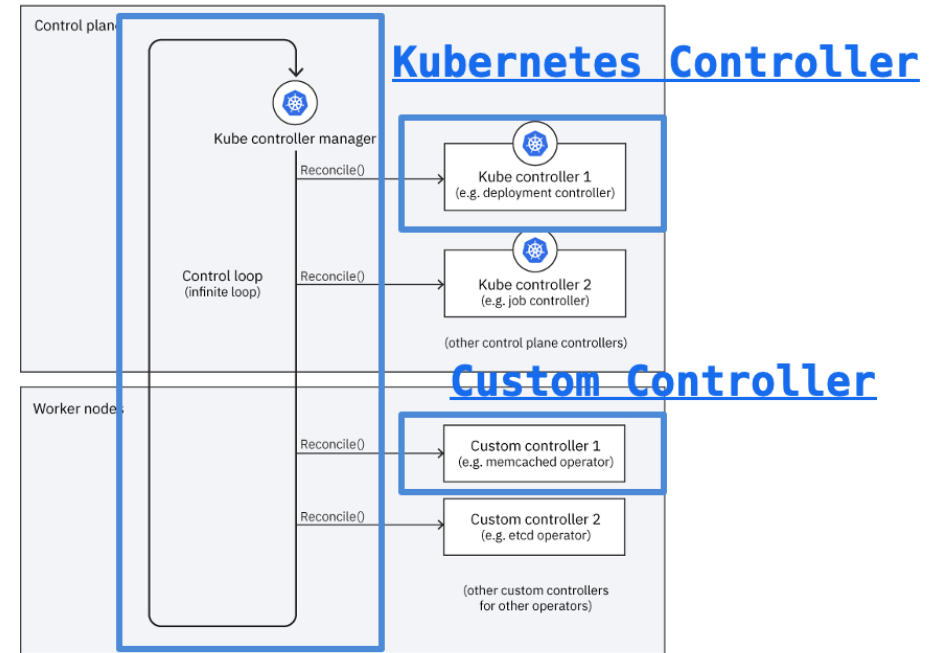


A **Kubernetes object is a "record of intent"**--once you create the object, the Kubernetes system will constantly work to ensure that object exists.

By creating an object, you're effectively telling the Kubernetes system what you want your cluster's workload to look like; this is your cluster's **desired state**.

The **Kubernetes declarative API** enforces a separation of responsibilities. You declare the **desired state** of your resource. The **Kubernetes controller keeps the current state of Kubernetes objects in sync with your declared desired state**.

Everything as Declarative (선언적) Kubernetes API



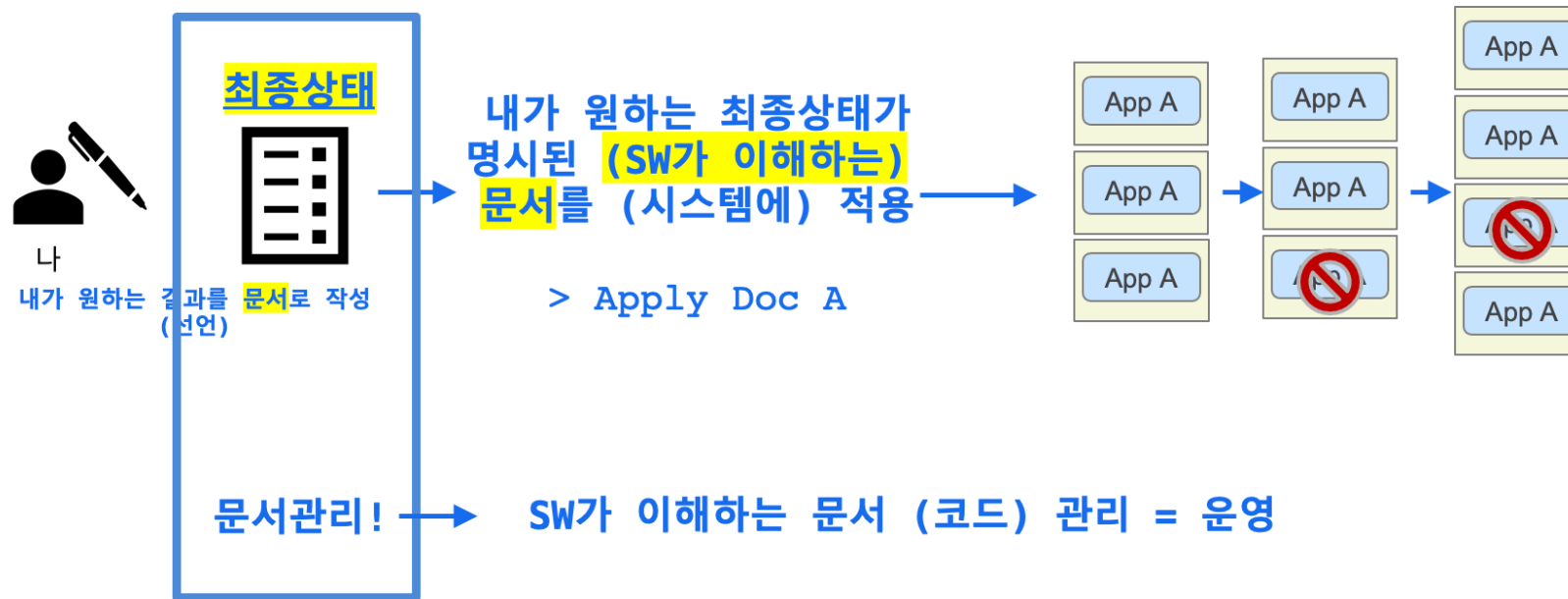
Kubernetes Controllers

- Control(reconciliation) loop for "desired state == current state"
- POD (Container) 등의 내장 자원을 관리 > Kubernetes에 내장된 기본 컨트롤러

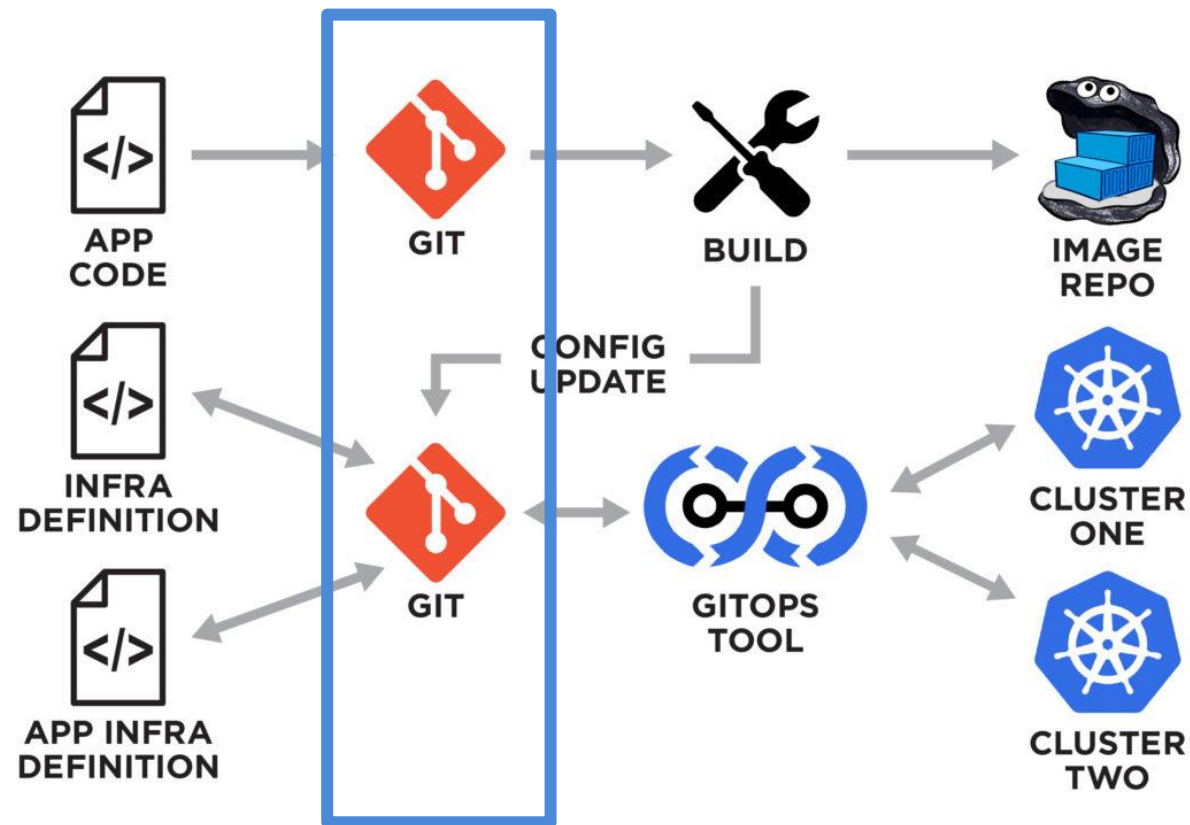
Operator 패턴

- Custom Resources + Custom Controllers
- POD 와 같이 k8s 내장 자원 외에 다른 "모든" 것들 도 K8S 자원으로 관리 가능
- 예) storage(rook), Infra (crossplane), db (vitensess), ACK (AWS controllers for Kubernetes)

Declarative <-> gitops

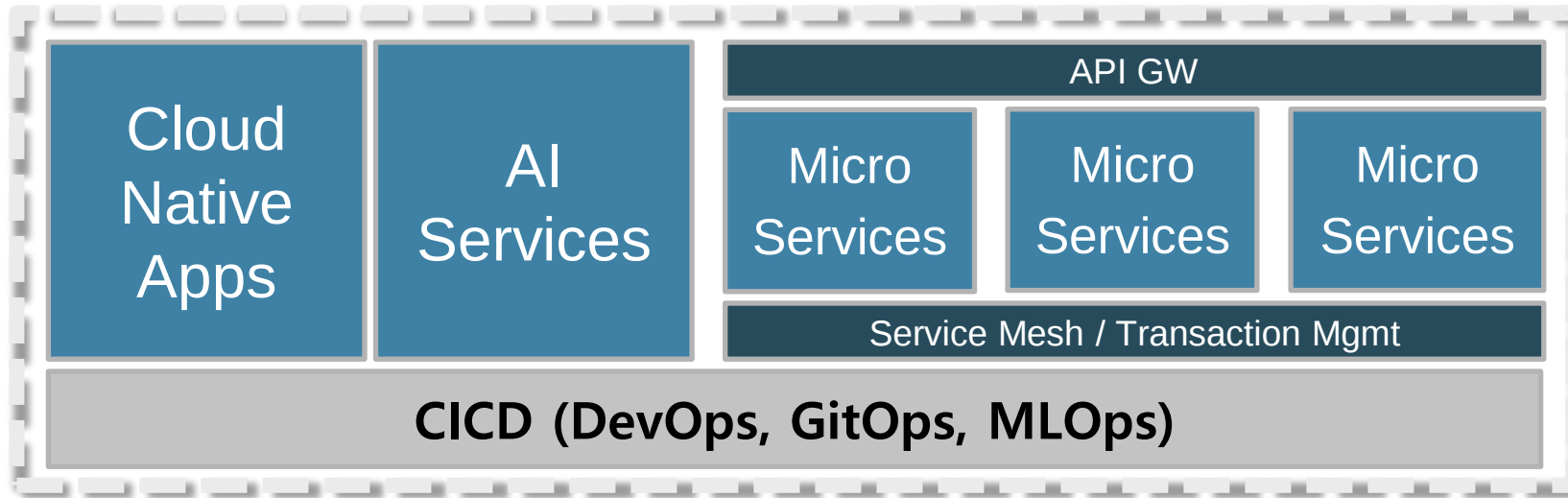


Declarative <-> gitops

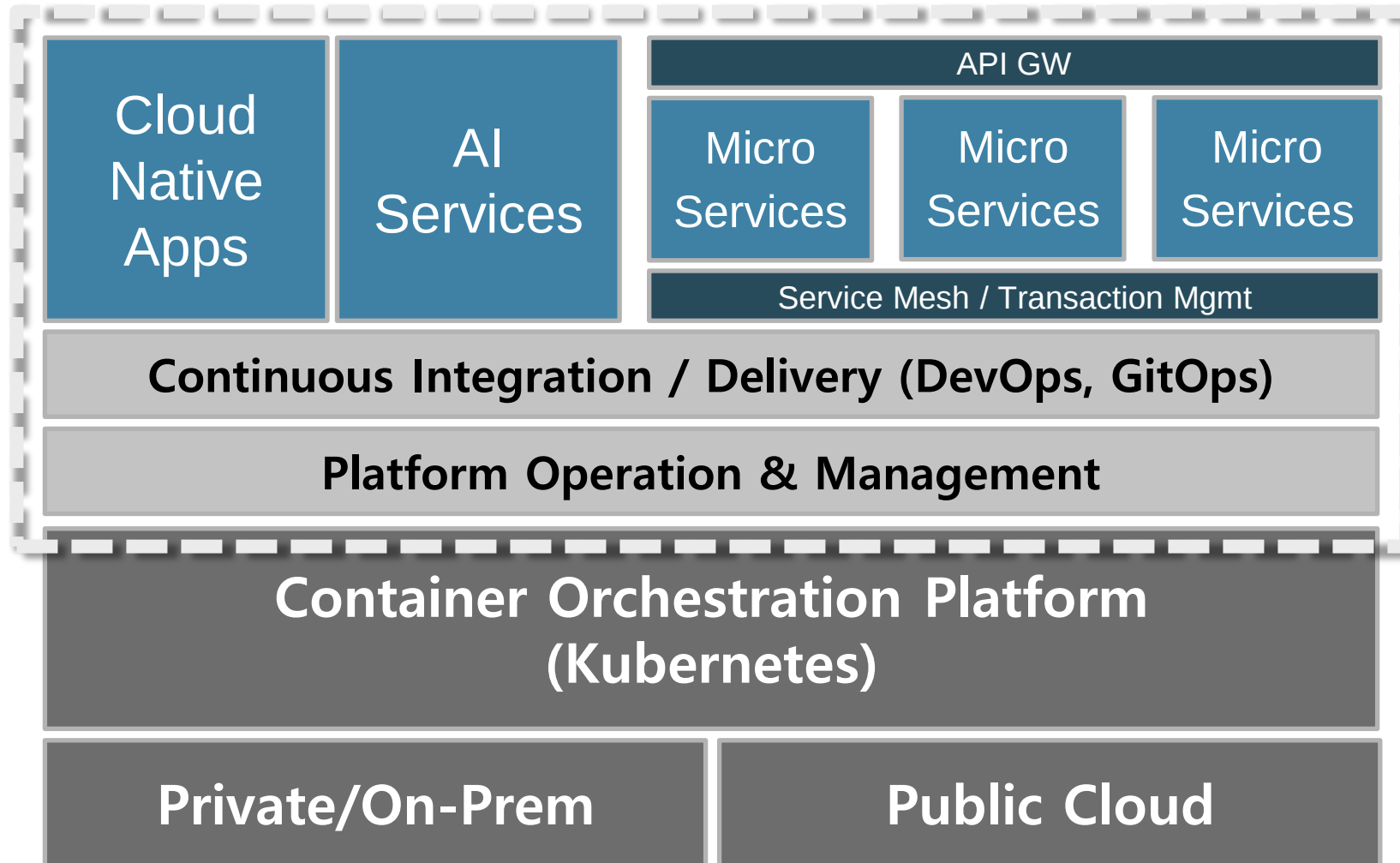


**What part of Kubernetes
Should be boring?**

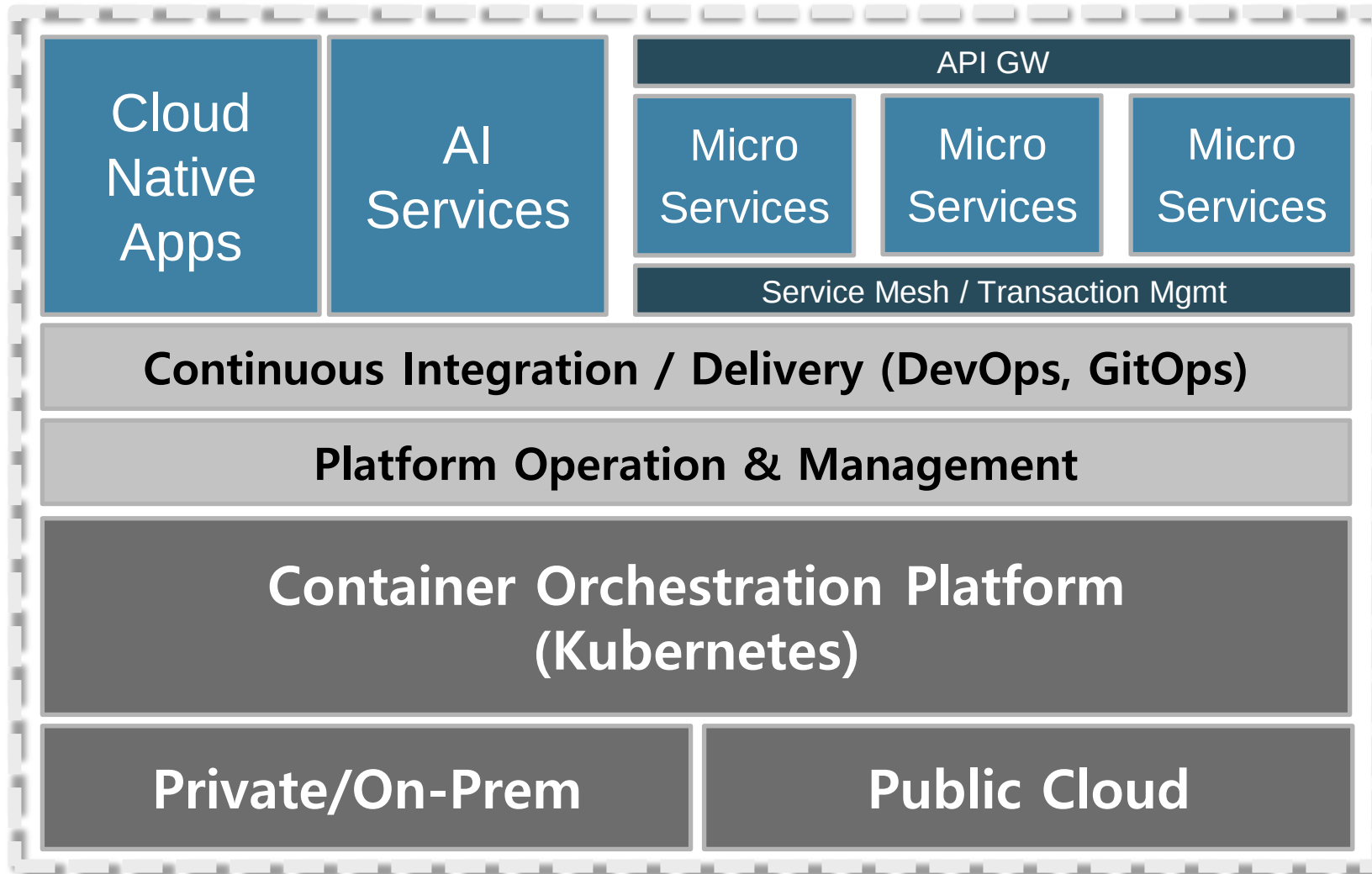
Cloud Native Infrastructure & Apps & Services



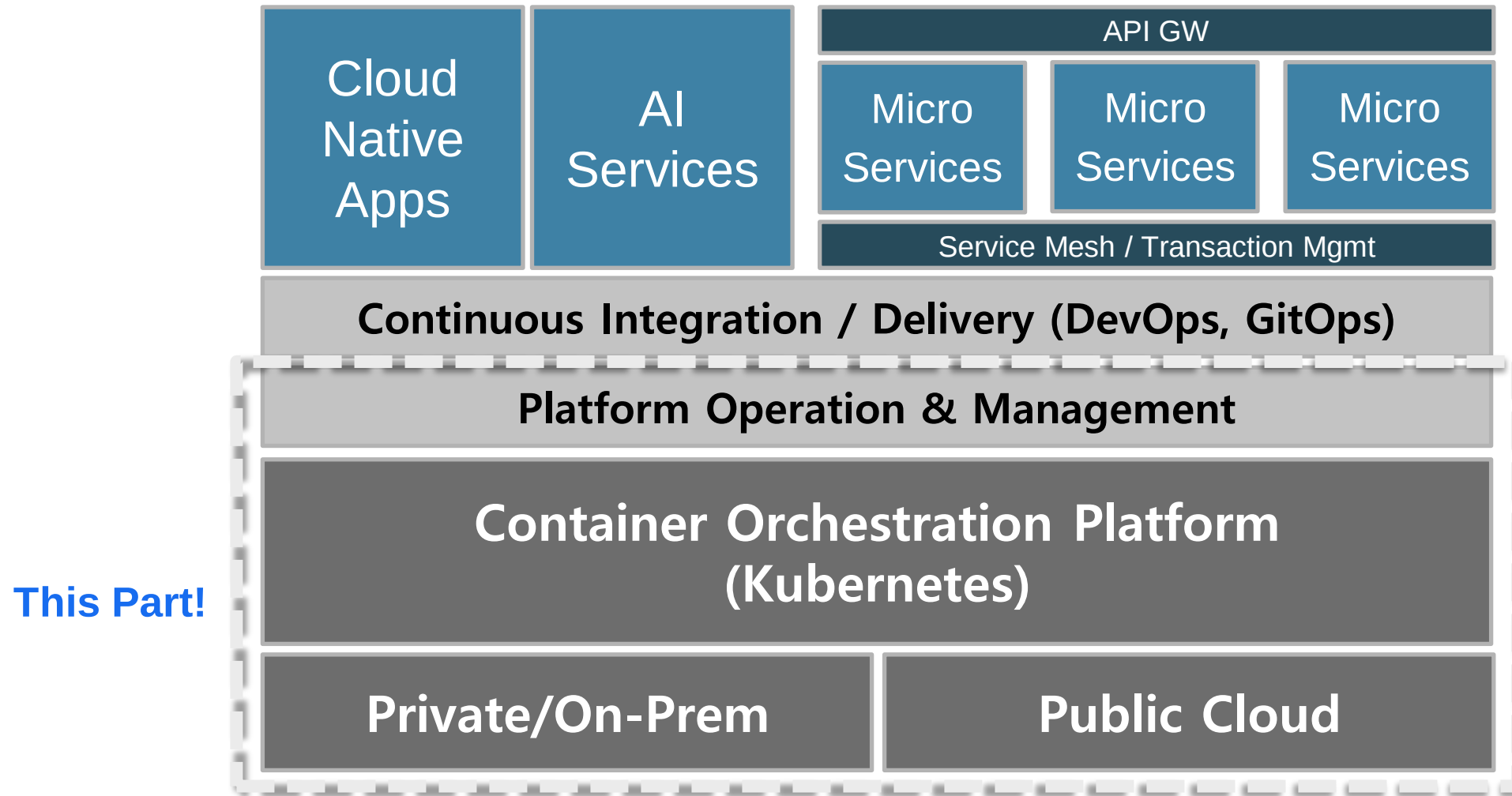
Cloud Native Infrastructure & Apps & Services



Cloud Native Infrastructure & Apps & Services



Cloud Native Infrastructure & Apps & Services

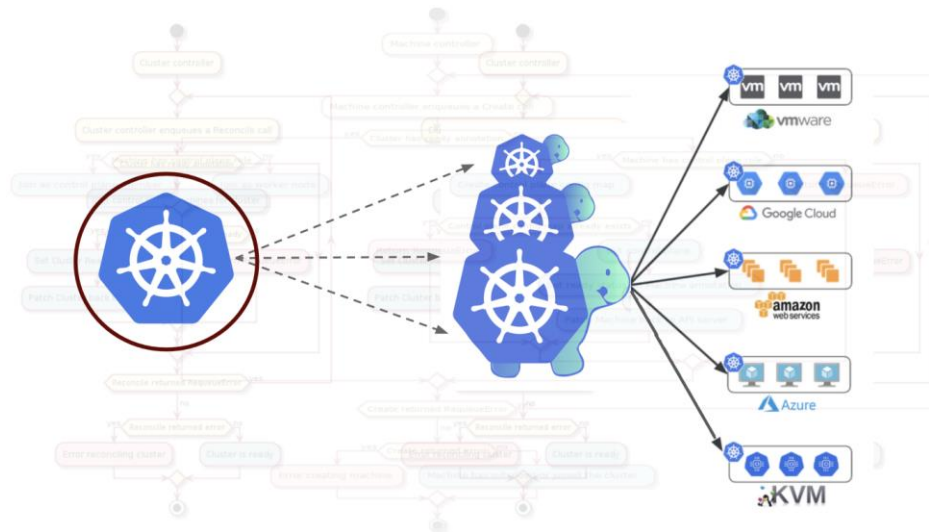


Kubernetes Cluster API

We make cluster lifecycle **boring**
(so reliable that no one pay anymore attention to it)

By Cluster Life Cycle SIG

SIG Cluster Lifecycle's objective is to simplify creation, configuration, upgrade, downgrade, and teardown of Kubernetes clusters and their components



2022 Kubecon & Cloud Native Con North America 에서 언급됨

SKT Container Service

Cluster API 기술을 활용하고

(서로 다른 인프라, Kubernetes를 표준화된 선언적 API로 관리)

GitOps 방식을 접목하여

(사용자 경험을 최대한 단순화 시키고, 일관성을 지키고)

Infra, K8S, Service들의 라이프사이클을 관리

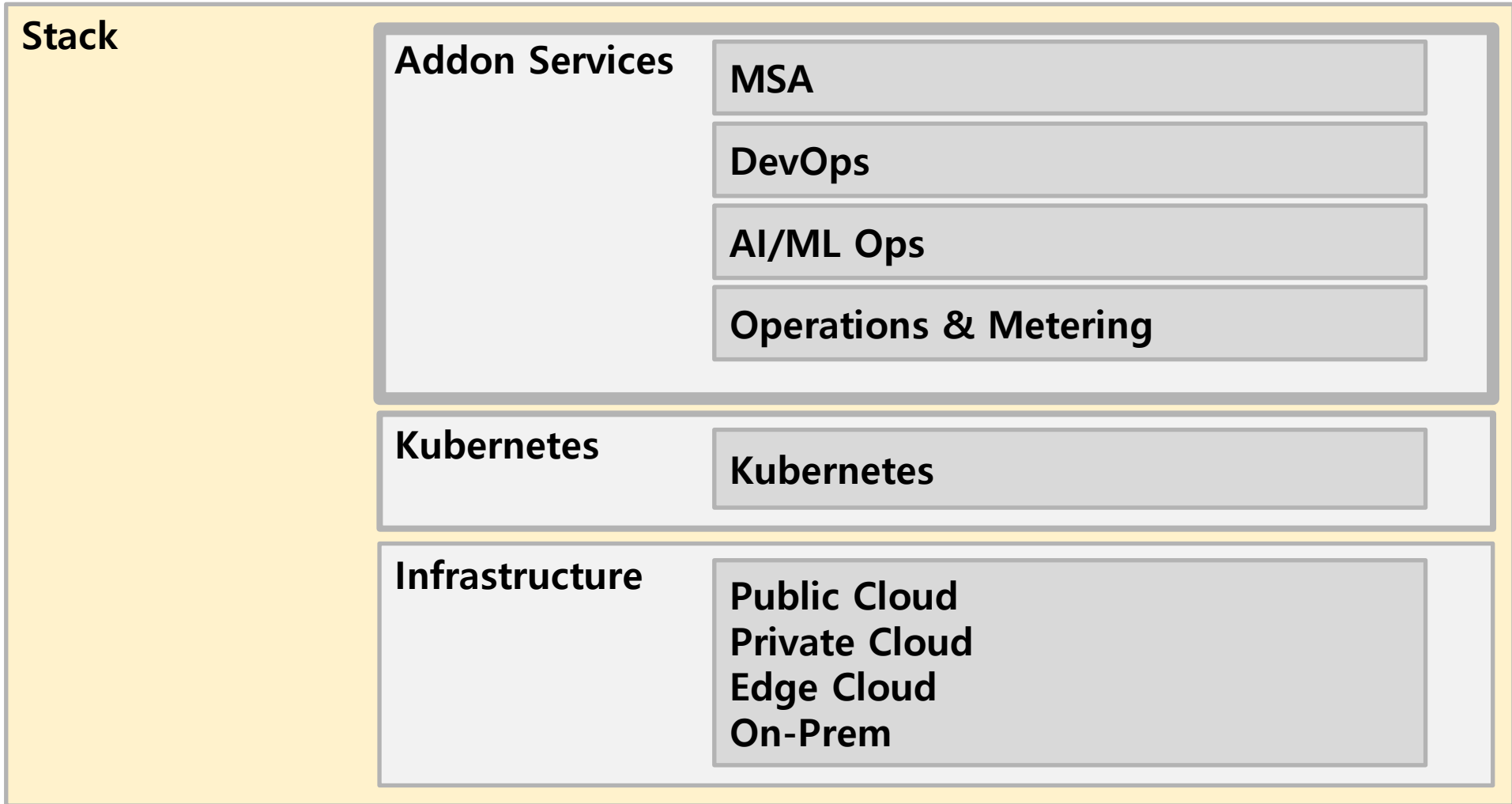
(부분이 아닌 필요한 모든 것을 관리)

**We make ~~cluster~~ “everything you need”
lifecycle boring
(so reliable that no one pay anymore attention to it)**

By SK Telecom

SK Telecom’s objective is to simplify creation, configuration, upgrade, downgrade,
and teardown of **Kubernetes clusters and their components**, and **necessary
services**

Stack (레고블럭처럼 선택 가능)



Stack (레고블럭처럼 선택 가능)

Stack

Addon Services

☒ MSA

DevOps

AI/ML Ops

☒ Operations & Metering

Kubernetes

☒ Kubernetes (SKT Kubernetes, EKS, AKS)

Infrastructure

☒ Public Cloud

Private Cloud

Edge Cloud

On-Prem

Stack (레고블럭처럼 선택 가능)

Stack

Addon Services

MSA

DevOps

☒ AI/ML Ops

☒ Operations & Metering

Kubernetes

☒ Kubernetes (SKT Kubernetes, EKS, AKS)

Infrastructure

☒ Public Cloud

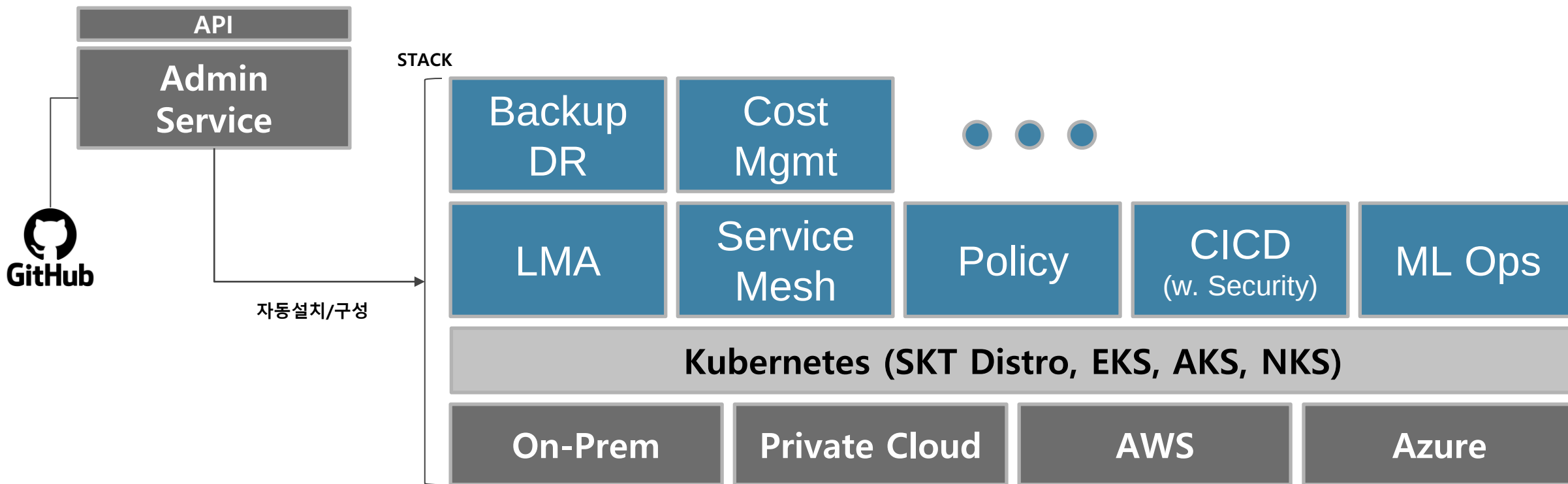
☒ Private Cloud

Edge Cloud

On-Prem

SKT Container Service Platform (개요)

Stack에 대한 라이프사이클 관리 API 제공 : GitOps 기반 Operation



SKT Container Service Platform (UI)

AWS Standard
AWS MSA

BYOH Standard
BYOH MSA

* BYOH
= Bring Your Own Host

TKS Admin Console

home > stacks > create

Create stack

제공되는 템플릿을 사용하여 클러스터와 함께 서비스를 provisioning 합니다.

Basic Properties

Name Info
Name of cluster
aws-reference-r9r336r1

Description
Description of cluster
aws-reference-r9r336r1

Service

☐ AWS Standard
Logging, Monitoring, Alerting 서비스가 포함된 클러스터를 생성합니다.

☒ AWS MSA
MSA 어플리케이션을 개발하는데 최적화된 클러스터를 생성합니다.

☐ BYOH Standard
Logging, Monitoring, Alerting 서비스가 포함된 클러스터를 생성합니다.

☐ BYOH MSA
MSA 어플리케이션을 개발하는데 최적화된 클러스터를 생성합니다.

AWS MSA (Micro Service Architecture)

AWS MSA는 Istio, Kiali, Elastic 등으로 구성된 Service Mesh 패키지와 로깅 & 모니터링 서비스를 제공합니다. 사용자의 앱을 MSA 구조로 배포/운영하는데 필요한 모든 구성요소가 포함되어 있습니다.

Install components

- TKS Kubernetes Cluster on AWS
- Monitoring Service (Thanos, Prometheus, Exporters, Dashboards)
- Logging Service (Grafana Loki, Fluent-bit, Dashboards)
- Alerting (Prometheus Alert Manager)
- Service Mesh (Istio, Envoy, Kiali, Dashboards)

License model SKT certified
SKT Trial Service License

Plan

Info

Startup Enterprise

Cancel + Create

Summary

Project
[p5d5raqno] test0920

Name
aws-reference-r9r336r1

Description
aws-reference-r9r336r1

Service template
aws-msa-reference

Plan
Startup
1 CPU, 4 RAM
3-Nodes HA supported set

Tags
no tags

Kubernetes (HA)

- CNI/CSI
- Keycloak

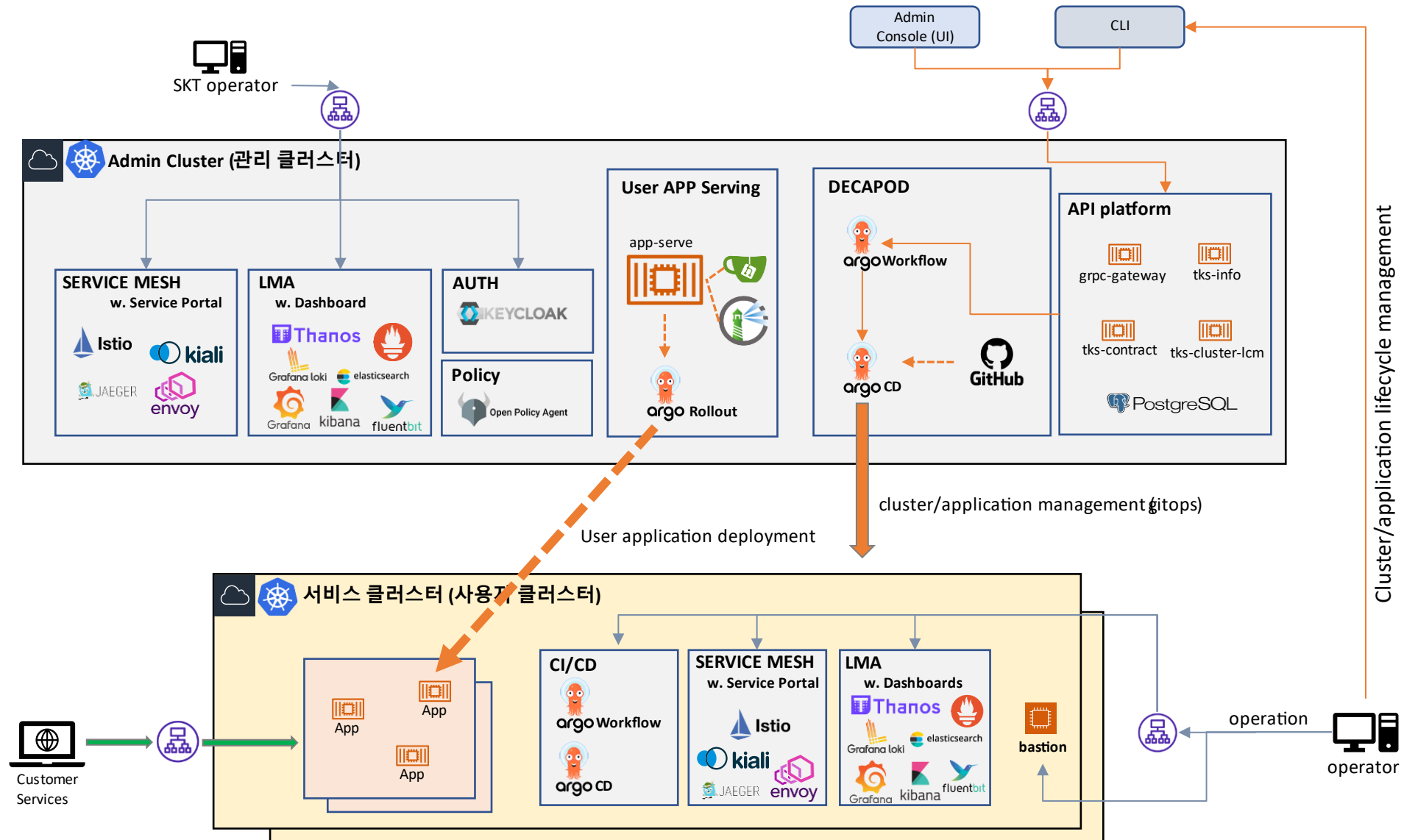
LMA

- Thanos System
- Prometheus System
- Exporters
- Dashboards
- Loki / Elastic Search
- Fluent-bit

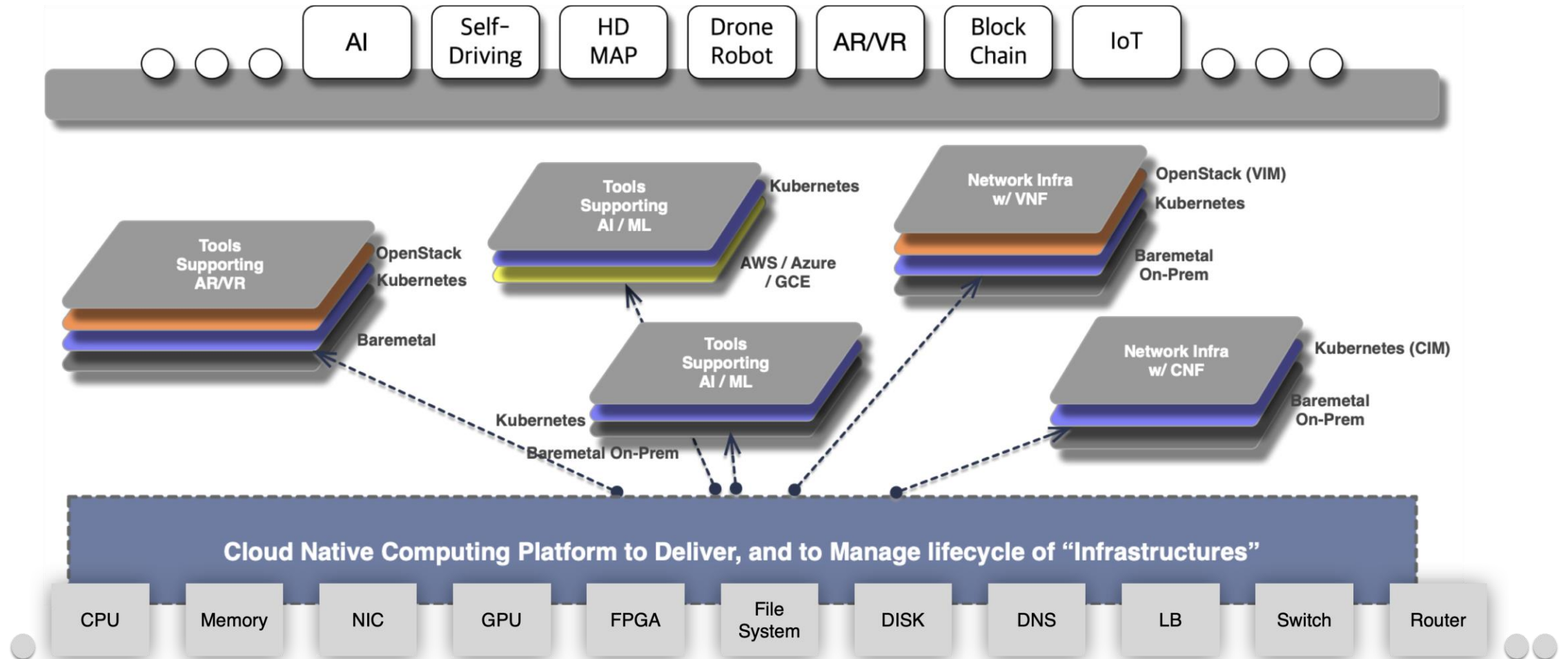
Service Mesh

- Istio
- Envoy
- Kiali
- Jaeger

SKT Container Service Platform (구성도)



Revisit Old Vision (2017년)



SKT Container Service 생태계내 누구든지

Cluster API 기술을 활용하고

(서로 다른 인프라, Kubernetes를 표준화된 선언적 API로 관리)

GitOps 방식을 접목하여

(사용자 경험을 최대한 단순화 시키고, 일관성을 지키고)

Infra, K8S, Service들의 라이프사이클을 관리

(부분이 아닌 필요한 모든 것을 관리)

SKT Container Service

생태계내 누구든지

<https://github.com/openinfradev>

: 프로젝트화는 못하지만 코드는 오픈하고 있습니다.

국내 CSP를 위한 Cluster API Provider (희망사항!)

- 커뮤니티 중심의 개발로도 가능할까? 라는 고민도.

기술발표

- 15:10 ~ 15:40 | **Track 105** – Application을 넘어 Infrastructure와 Kubernetes도 GitOps로 관리하기
- 16:00 ~ 16:30 | **Track 105** – Simple Java app Deployment Service for Kubernetes
- 16:40 ~ 17:10 | **Track 103** – DevRel 플랫폼 개발 여정 (Feat. AWS 마이그레이션)
- 17:10 ~ 17:40 | **Track 103** – MLOps: 쉽게 구현하는 ML Model Serving A to Z
- 온라인 발표 – 로그 수집, 코드로 관리하고 쉽게 적용하자

설문 & 경품 추첨 (Free Trial Service on AWS 신청 가능)

전시부스

- SKT Container Solution 소개
- SKT Cloud Business (MSP) 소개
- SKT 개발자 생태계 (Devocean) 소개 및 채용 정보





감사합니다