



일본에서도 VOD를 빠르게 업로드하고 재생할 순 없을까

NAVER

Emerging

TECHnology

NAVER ETECH.

포토 / 오디오 / 비디오의 <생산 - 클라우드 - 소비> 워크플로의 전구간 기술 연구와 개발을 담당합니다. 글로벌 환경에서 시간 / 공간 / 용량의 제약 사항을 극복하고 생생한 현장 느낌과 안정적인 지원을 위해 이머징 기술 연구와 개발을 통한 원격의 시대를 준비하고 있습니다.

생산

Audio / Video 편집 Engine

LIVE Streaming Engine

Visual Effect Engine

PRISM LIVE Studio App

미디어 Ingestion

SmartStudio

클라우드

포토 클라우드

AOD 클라우드

VOD 클라우드

LIVE 클라우드

소비

Mobile / Web / TV Player

VR / 360 Player (incl. HMD)

Immersive Playback

Media Casting

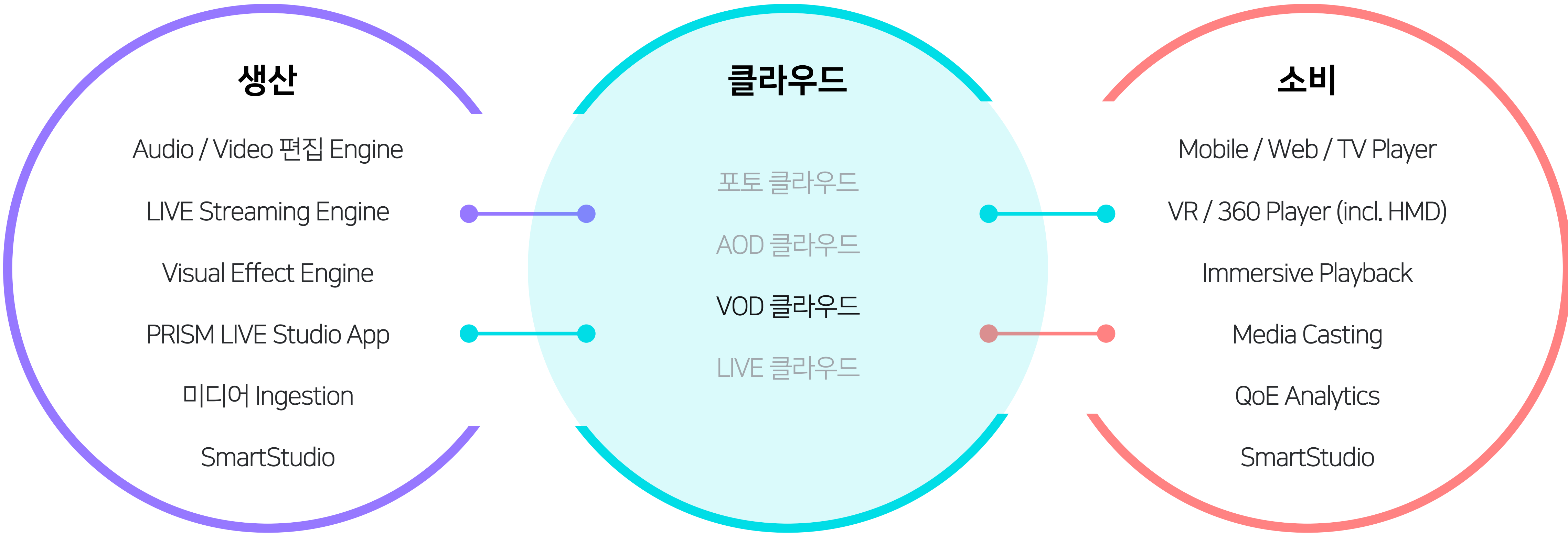
QoE Analytics

SmartStudio



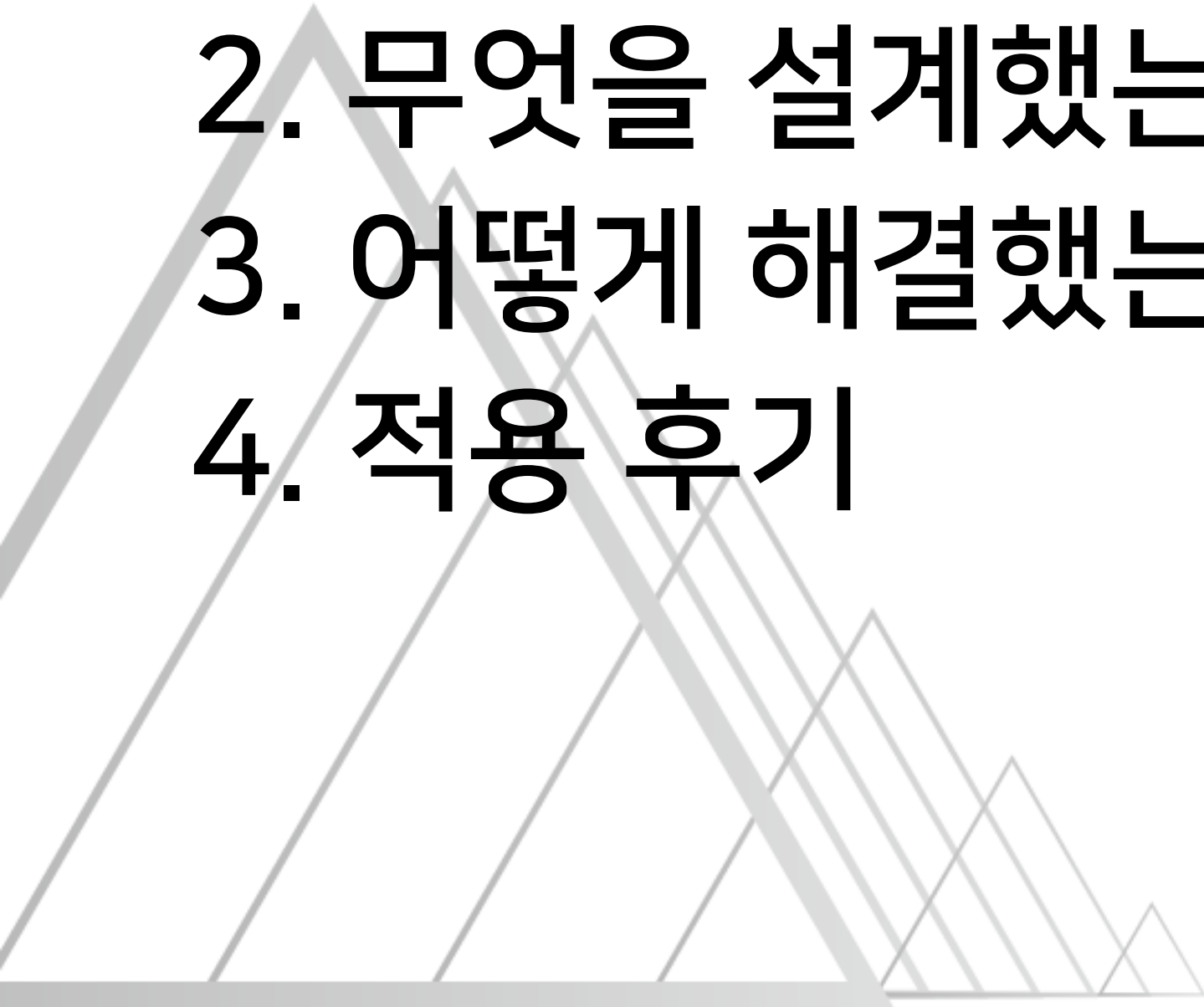
NAVER ETECH.

포토 / 오디오 / 비디오의 <생산 - 클라우드 - 소비> 워크플로의 전구간 기술 연구와 개발을 담당합니다. 글로벌 환경에서 시간 / 공간 / 용량의 제약 사항을 극복하고 생생한 현장 느낌과 안정적인 지원을 위해 이머징 기술 연구와 개발을 통한 원격의 시대를 준비하고 있습니다.



CONTENTS

1. 왜? 하게 되었는가?
2. 무엇을 설계했는가?
3. 어떻게 해결했는가?
4. 적용 후기

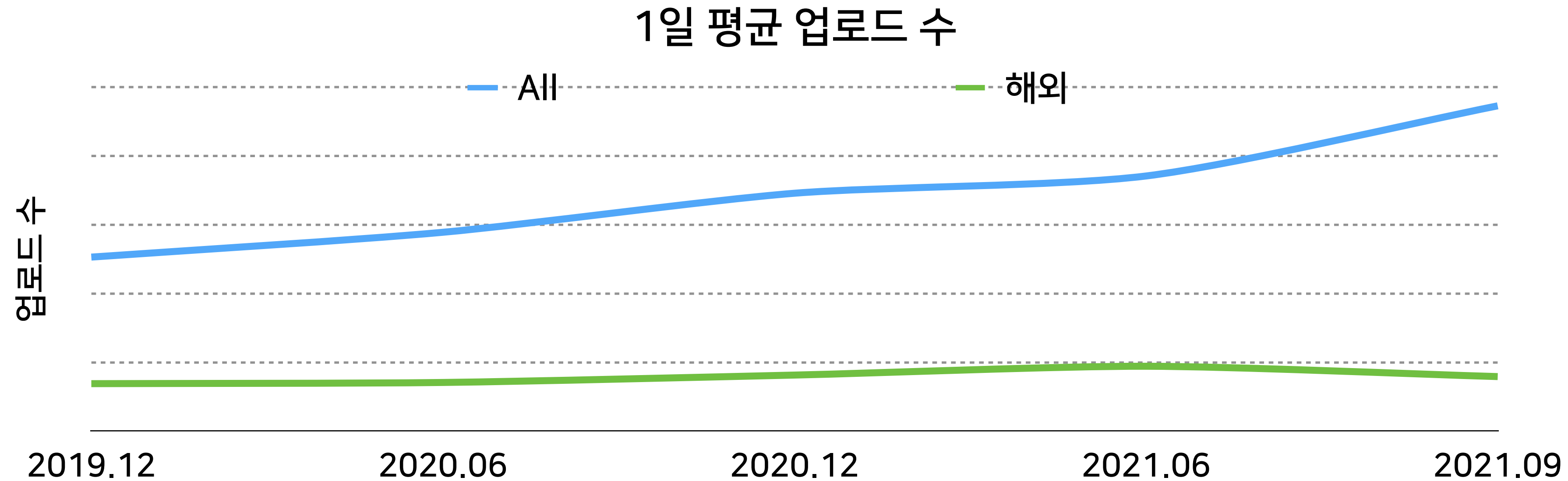


1. 왜? 하게 되었는가?

1.1 동영상이 대세

동영상 업로드는 해마다 33% 씩 증가

- VOD 사용량은 한국 뿐만 아니라 세계적으로 증가하고 있음
- 한국을 벗어난 곳에서도 사용자 경험이 좋을까?



1.2 공기 같은 존재 VOD

일본 사업 진출 요구사항 증가

- 네이버 일본 향 서비스가 출시할 때, VOD 클라우드도 구축되어 있어야 함

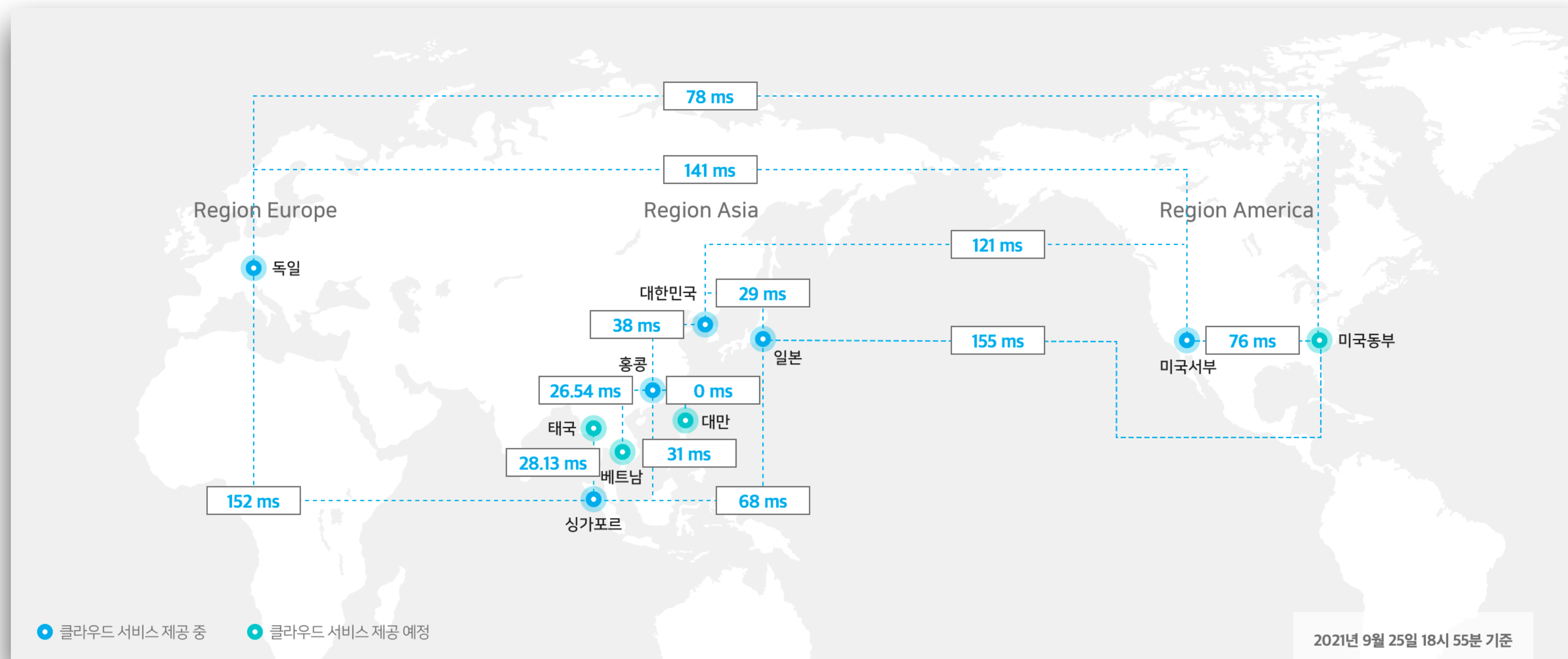


1.3 일본은 가까운데 한국 리전 사용하면 안 돼? N DEVIEW 2021

Global Latency Map

- Latency : 하나의 패킷이 두 지점 간 왕복에 소요되는 시간

Naver Cloud Platform 의 Global Latency Status

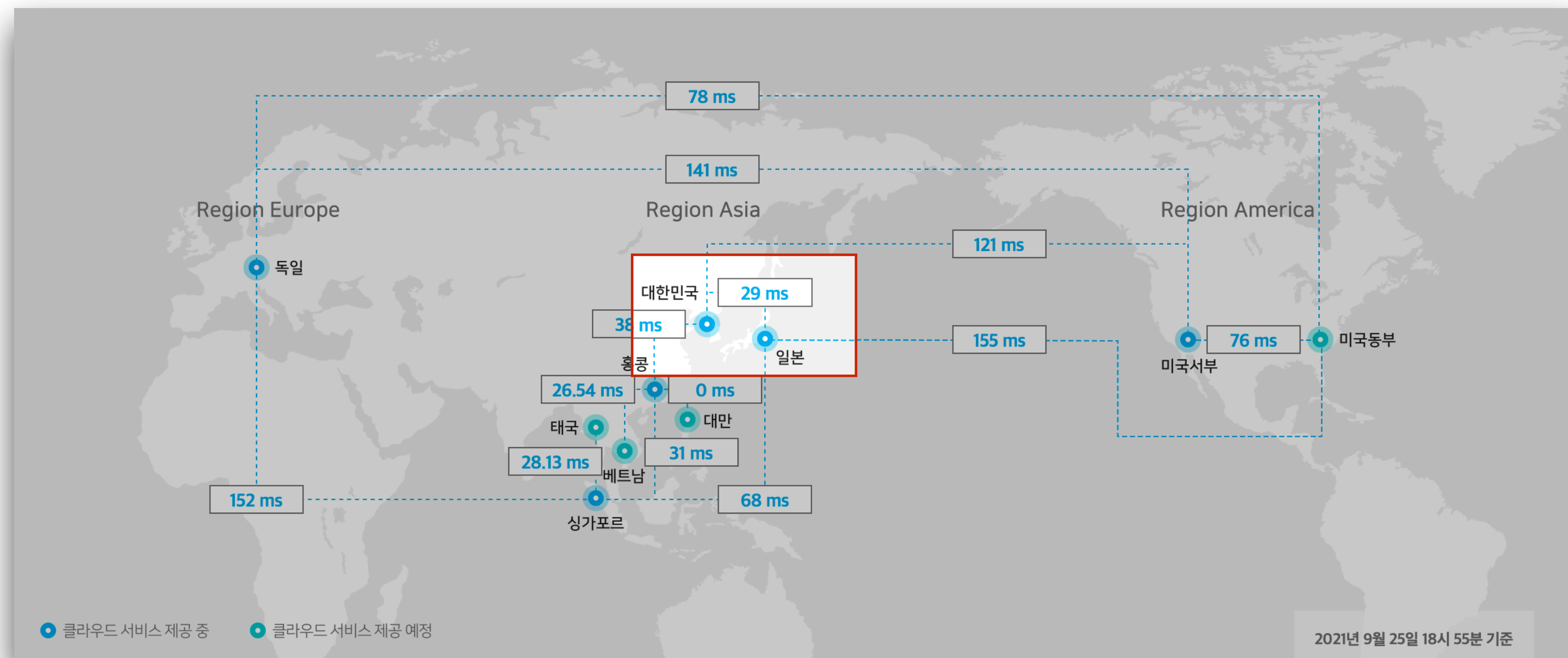


출처 : <https://www.ncloud.com/product/global/globalLatencyStatus>

1.3 일본은 가까운데 한국 리전 사용하면 안 돼? N DEVIEW 2021

한국과 일본의 Latency는?

- 한국과 일본간에는 29ms Latency 와 240 Gbps 대역폭이 있다
Naver Cloud Platform 의 Global Latency Status



출처 : <https://www.ncloud.com/product/global/globalLatencyStatus>

1.4 비즈니스 법적 이슈

로마에 가면 로마법을 따르라

- EU(유럽연합) 의 개인정보보호 법령인 GDPR
- 이슈 사전 예방 차원에서 로컬 리전에 서버 및 데이터를 관리

네이버 클라우드 플랫폼과 GDPR 준수

GDPR(General Data Protection Regulation)은 유럽연합 정보주체의 개인정보 보호를 위해 개정된 개인정보보호법으로, 근래 20년 내 알려진 프라이버시 관련 법률 중 규제의 범위와 개념이 가장 넓혀진 것으로 평가되고 있습니다.

EU 거주인의 개인정보를 처리하는 전 세계 모든 회사는 이 법을 준수해야 합니다.

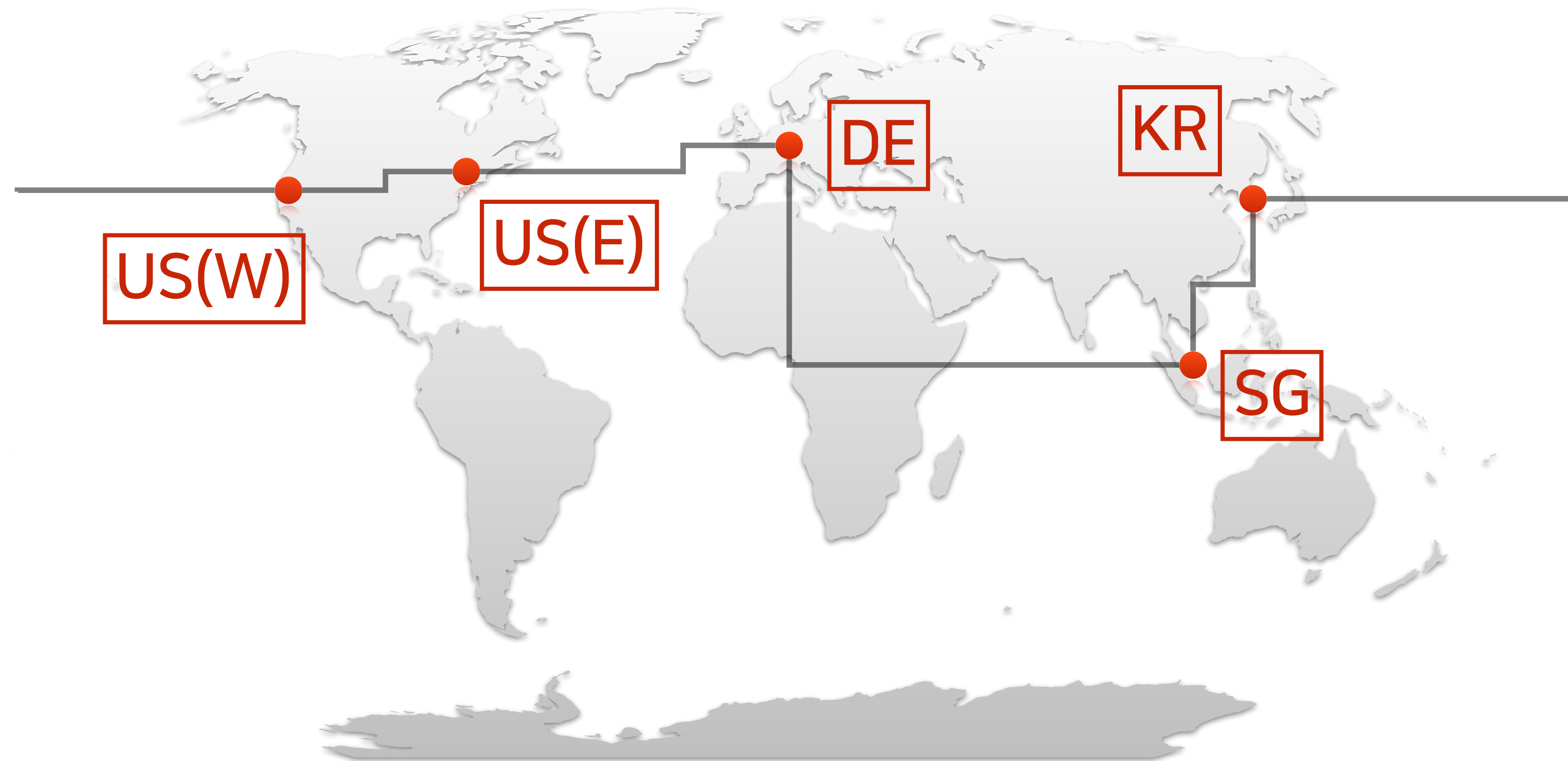
네이버 클라우드 플랫폼은 자체적인 GDPR 준수 뿐만 아니라, 고객이 스스로의 비즈니스 활동시, 적용되는 GDPR 준수에 도움이 되도록 최선을 다하고 있습니다.

출처 : <https://www.ncloud.com/intro/gdpr>

1.5 기존 VOD 클라우드의 문제

이미 글로벌 리전 VOD 클라우드가 있지만

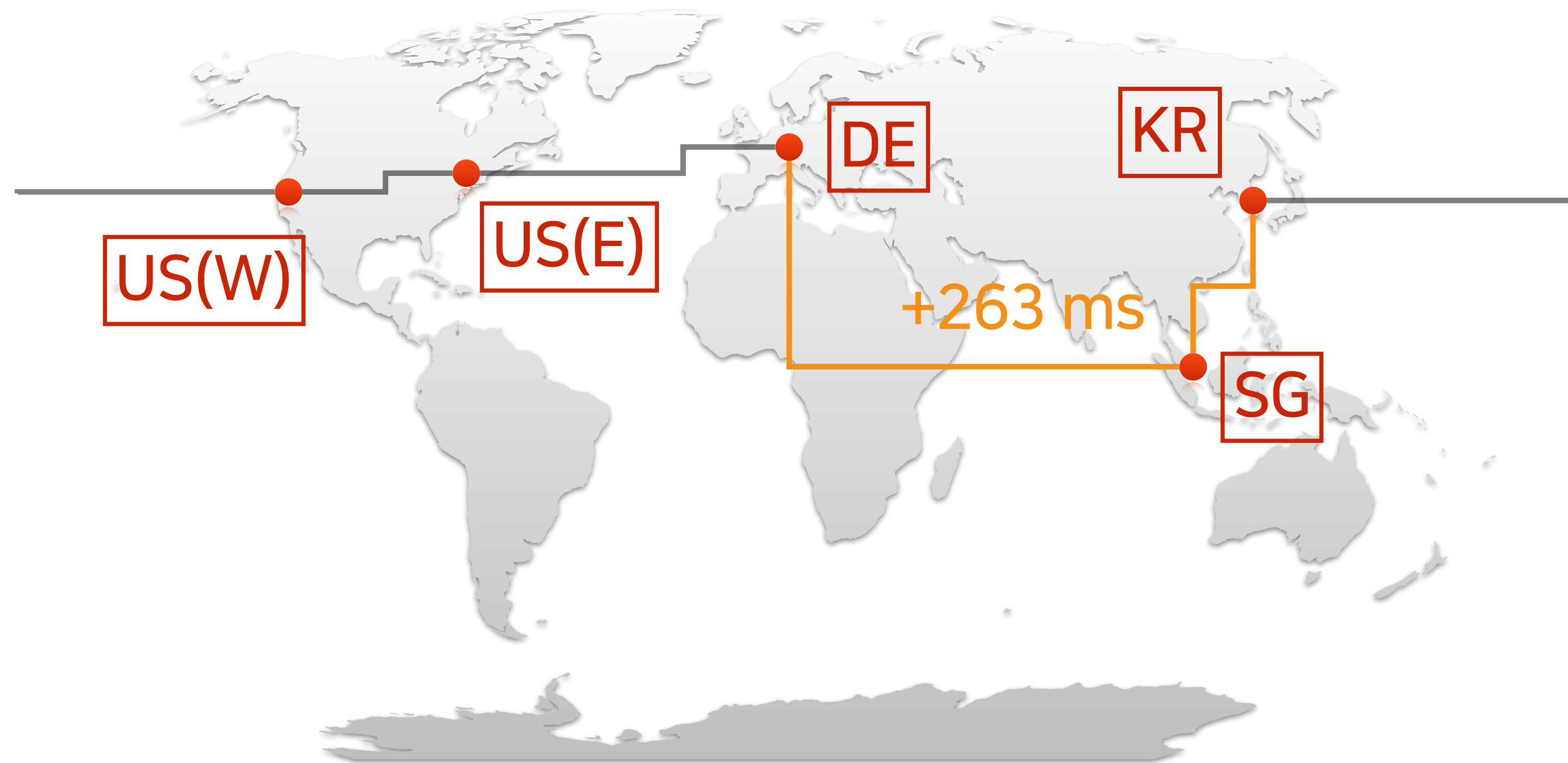
- 게이트웨이, 업로더는 세계 각지에 배치되어 있다



1.5 기존 VOD 클라우드의 문제

글로벌 엔드 유저는 한국과 멀수록 느려지게 되고

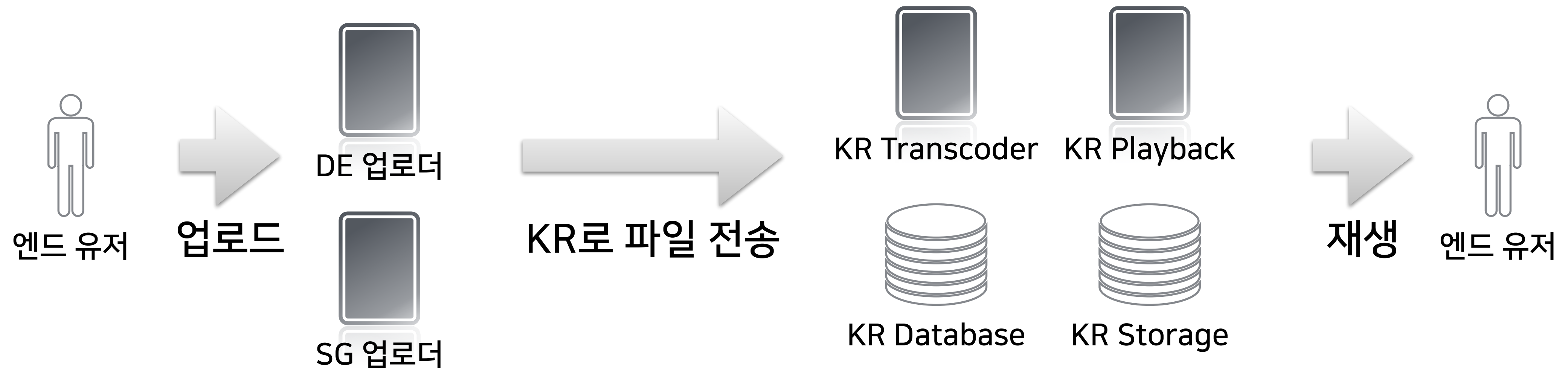
- 글로벌 엔드 유저는 한국 사용자에게 비해 떨어지는 성능을 경험



1.5 기존 VOD 클라우드의 문제

모든 길은 한국으로 통한다

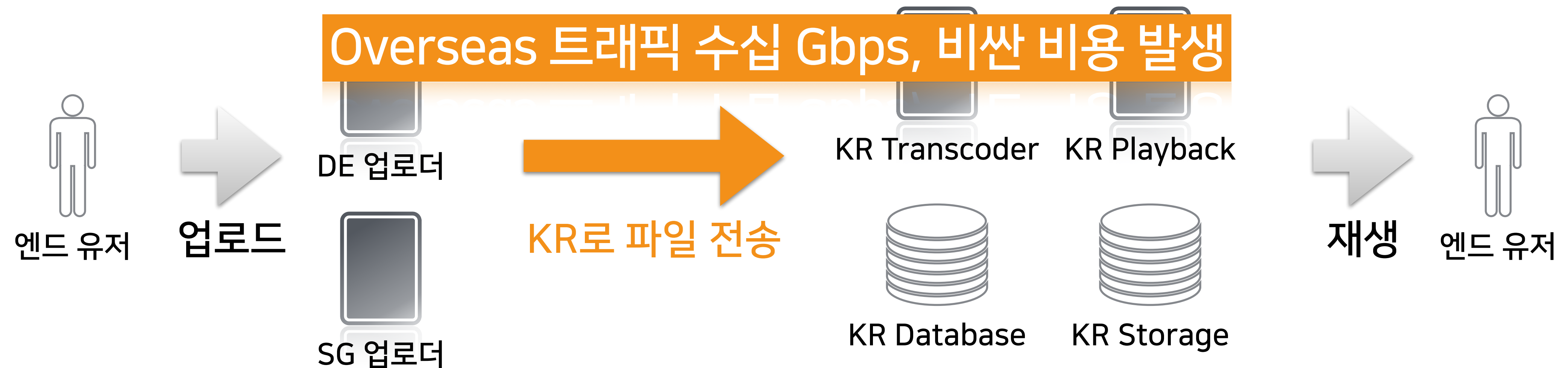
- 업로드 된 영상은 모두 한국으로 전송되어 트랜스코딩, 딜리버리 수행
- 글로벌 서비스의 사용자가 많아질수록 비용이 증가되는 구조



1.5 기존 VOD 클라우드의 문제

모든 길은 한국으로 통한다

- 업로드 된 영상은 모두 한국으로 전송되어 트랜스코딩, 딜리버리 수행
- 글로벌 서비스의 사용자가 많아질수록 비용이 증가되는 구조



1.6 기타 요구 사항

미국, 유럽에도 리전 만들어 주세요

- 일본 리전이 가장 급했으나 3rd 리전도 언젠가는 해야 한다

플랫폼 운영 비용 증가를 최소화 할 순 없을까?

- 리전이 늘수록 운영 비용 증가는 불 보듯 뻔하다. 어떻게 줄일 수 있을까?

네이버 서비스의 변동 사항은 적어야 하지 않을까?

- 빠른 서비스 출시를 위해서는 어느 리전을 연동하든 방식은 같아야 한다

2. 무엇을 설계했는가?

2.1 일본 리전 목표

목표

- 일본 엔드 유저도 업로드, 재생시 속도를 향상 시키고 최고의 성능을 경험하도록 하자
- 향후 다른 리전으로 확대를 빠르게 할 수 있는 VOD 클라우드 표준을 정립하자
- 네이버 서비스에게 같은 Entrypoint 를 제공하고 변경을 최소화하자

2.2 일본 리전 : the Full VOD CLOUD set

Region, Zone 개념

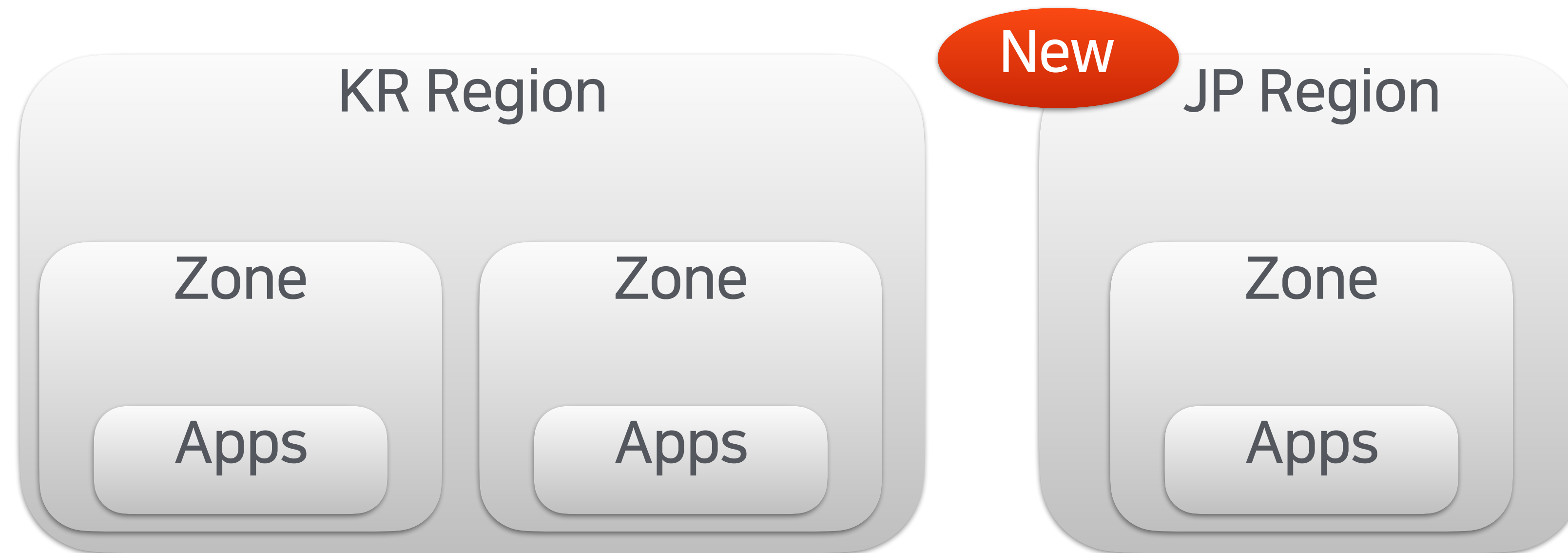
- Region : 데이터 센터 및 서비스의 클러스터
- Zone : 추상화 데이터 센터



2.2 일본 리전 : the Full VOD CLOUD set

한국 리전과 같은 구성의 일본 리전 VOD 클라우드 구축

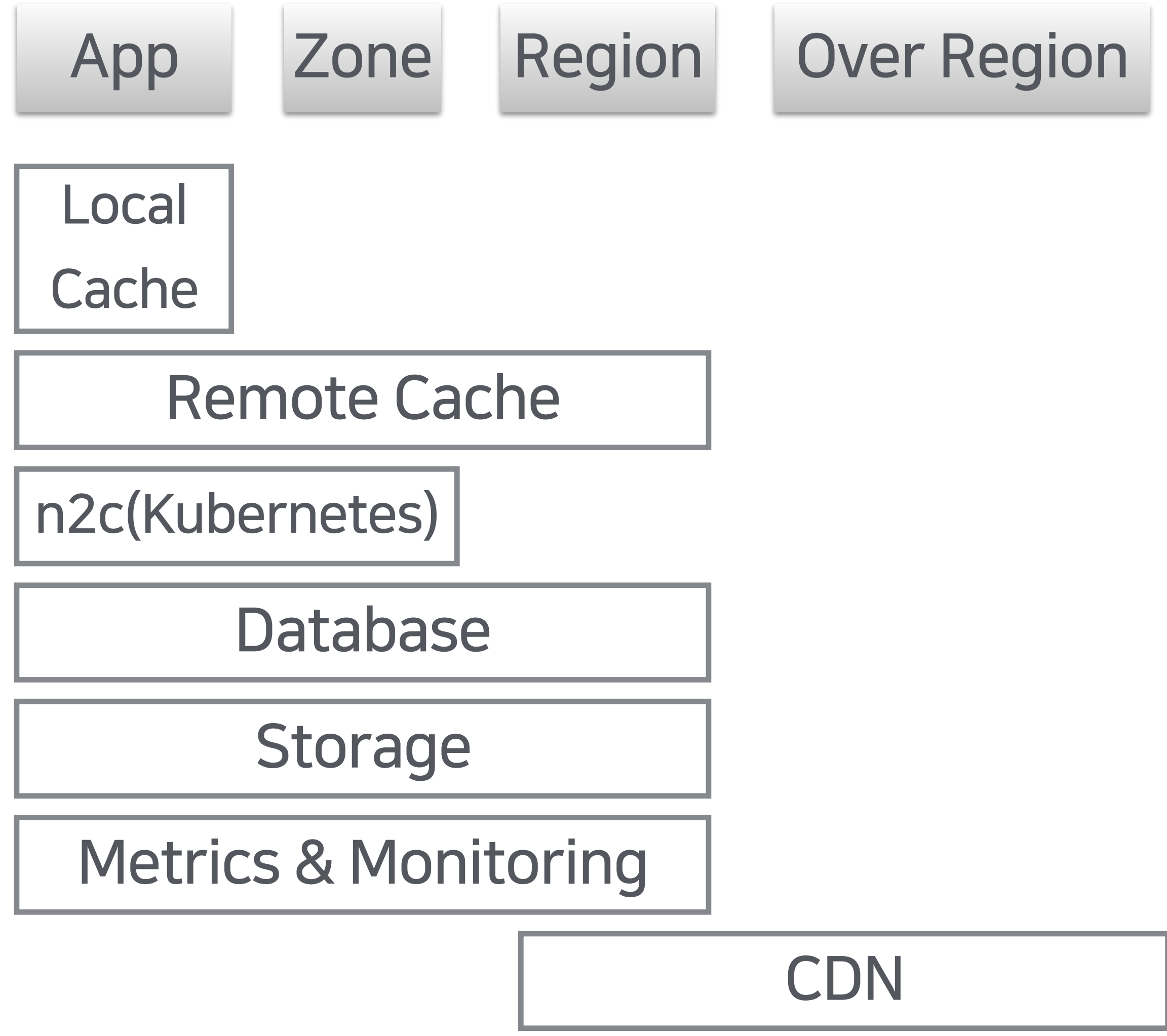
- 서버 수는 다를수 있어도 구성은 같다



2.2 일본 리전 : the Full VOD CLOUD set

Infrastructure 의 영향 범위

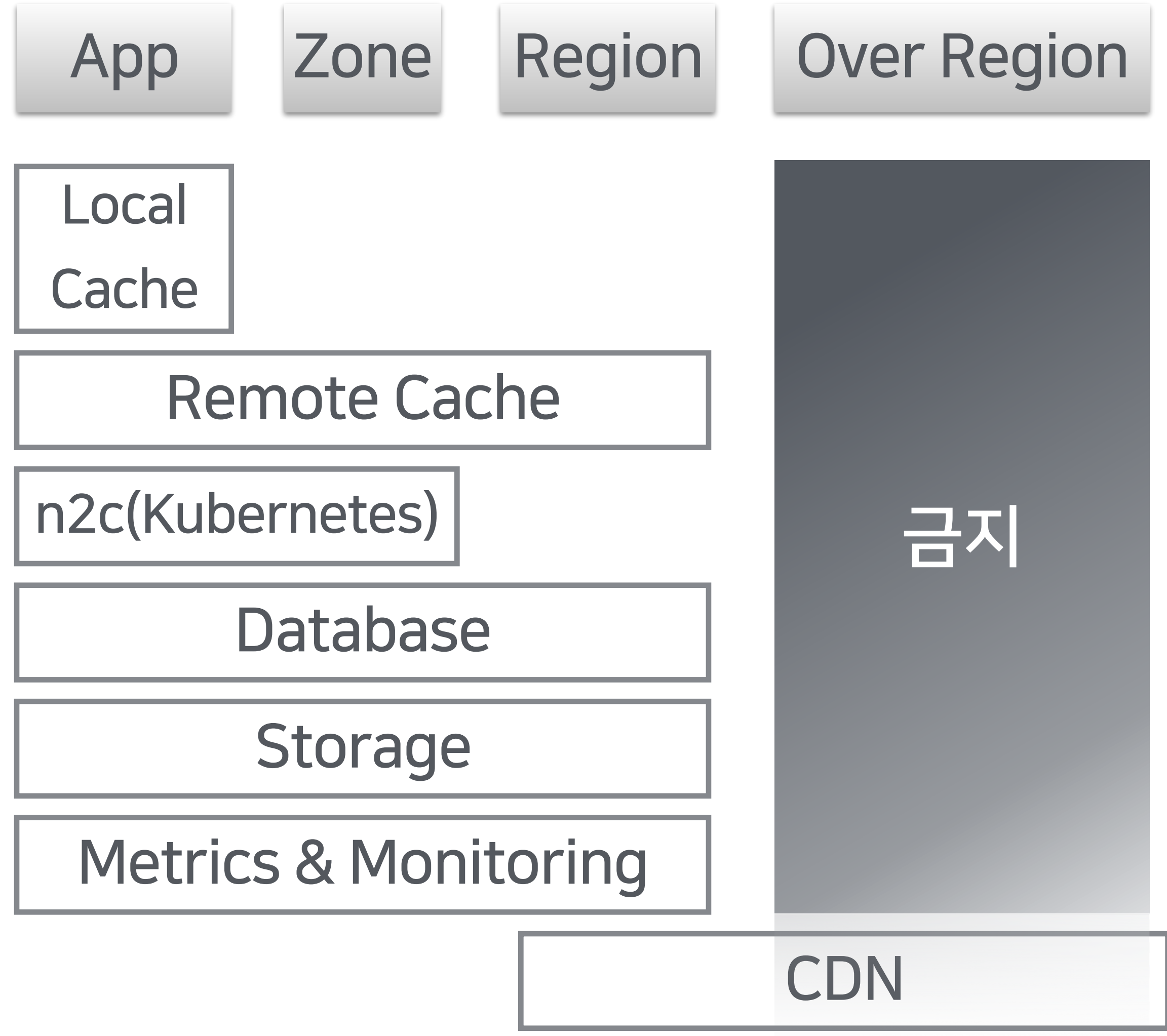
- Infrastructure 별로 어디까지 영향을 주는지 범위를 정의



2.2 일본 리전 : the Full VOD CLOUD set

Over Region 트래픽 최소화

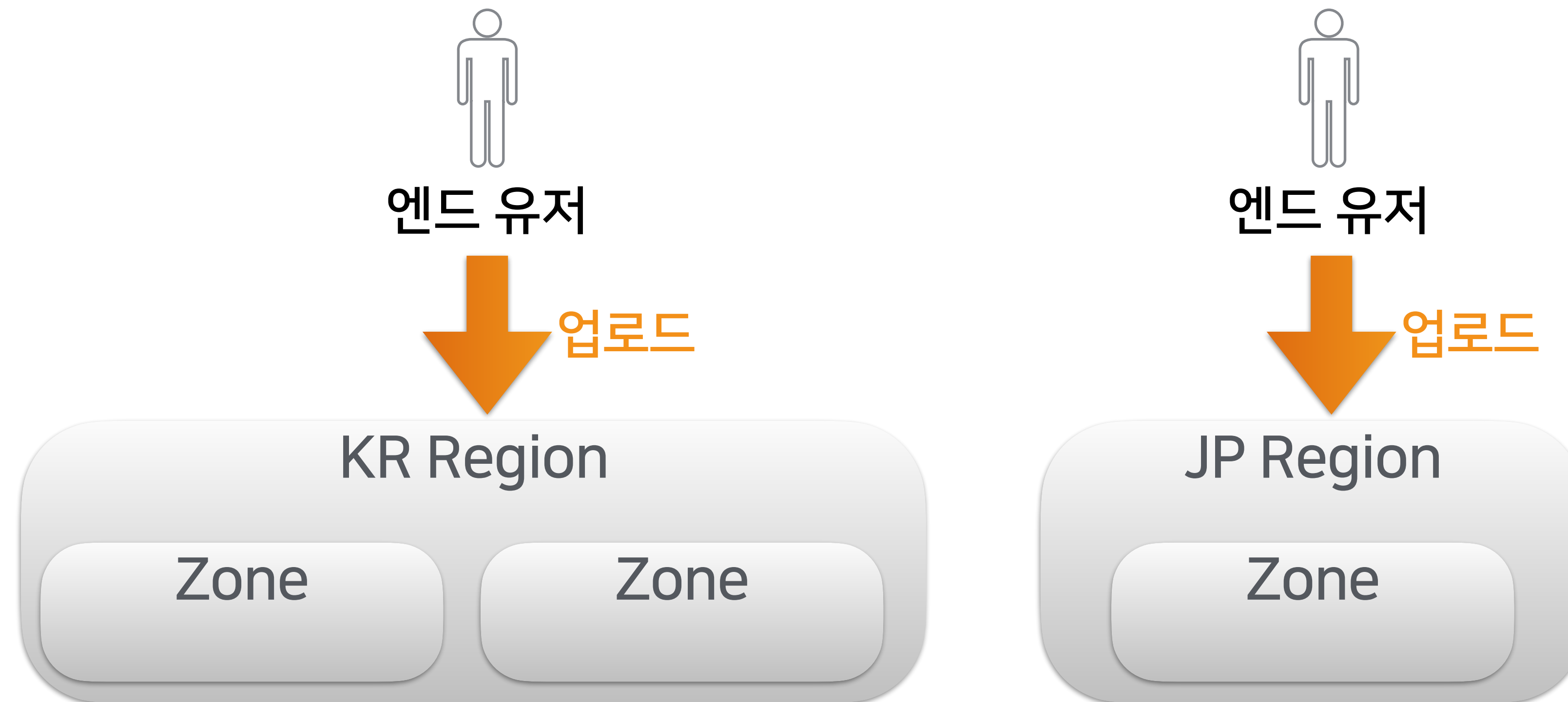
- Latency 증가 최소화
- 장애 발생시 Under Region 으로 격리



2.2 일본 리전 : 업로드 & 재생 플로우

업로드는 로컬 리전으로, 재생은 글로벌하게

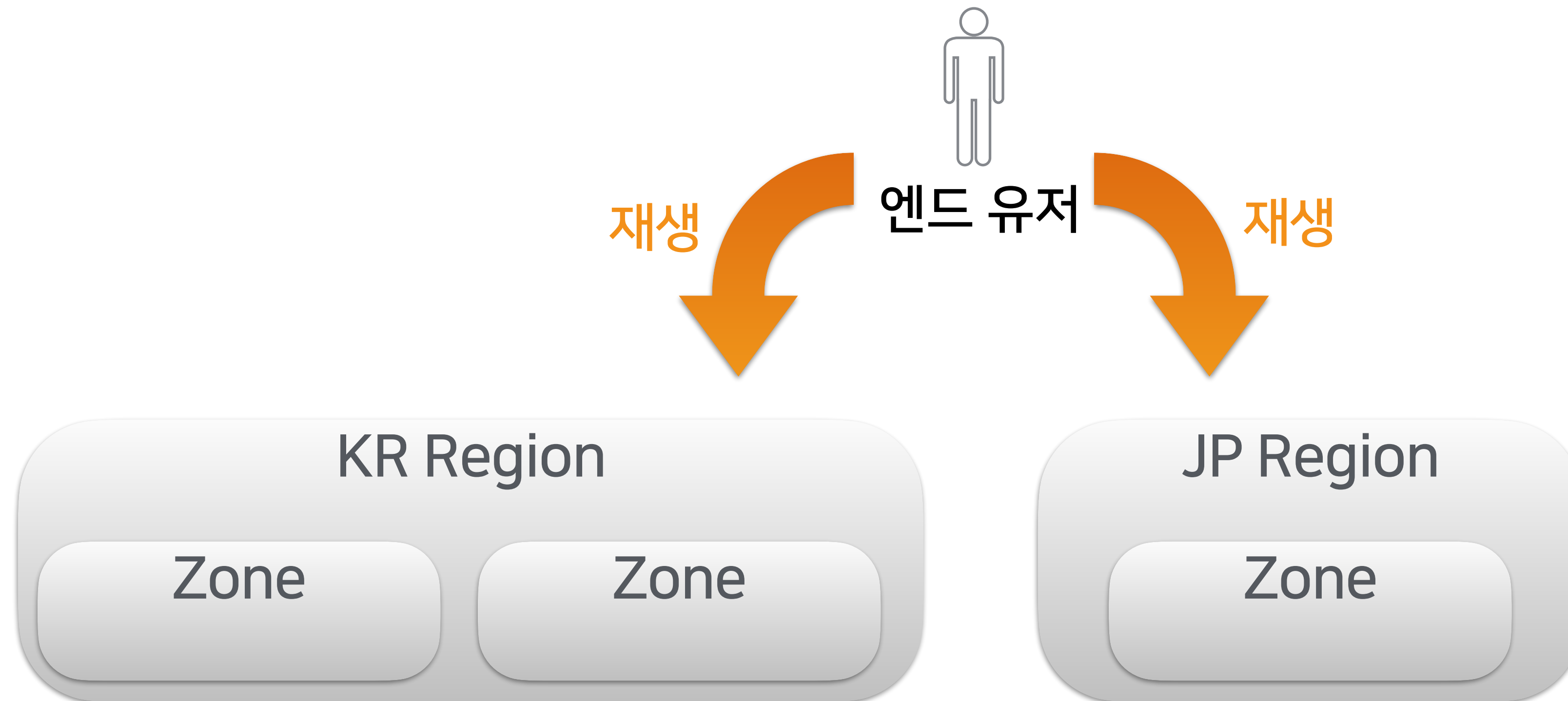
- 엔드 유저는 자신과 가장 가까운 리전으로 업로드



2.2 일본 리전 : 업로드 & 재생 플로우

업로드는 로컬 리전으로, 재생은 글로벌하게

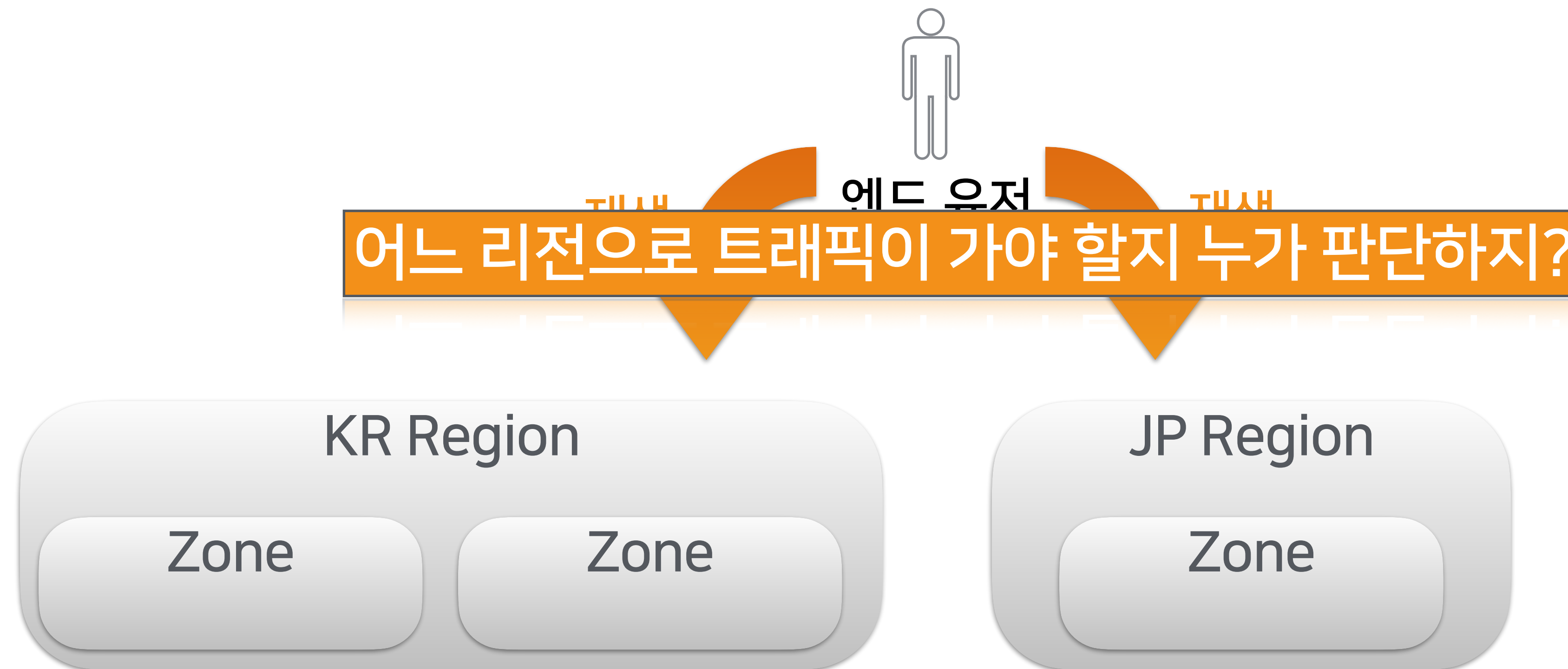
- 재생은 영상이 저장된 리전에서 데이터를 읽음 + GCDN(Geo Routing)



2.2 일본 리전 : 업로드 & 재생 플로우

리전간 트래픽 분리는 누가?

- 네이버 서비스의 변경 영향성을 최소화하고 빠르게 연동할 수 있도록

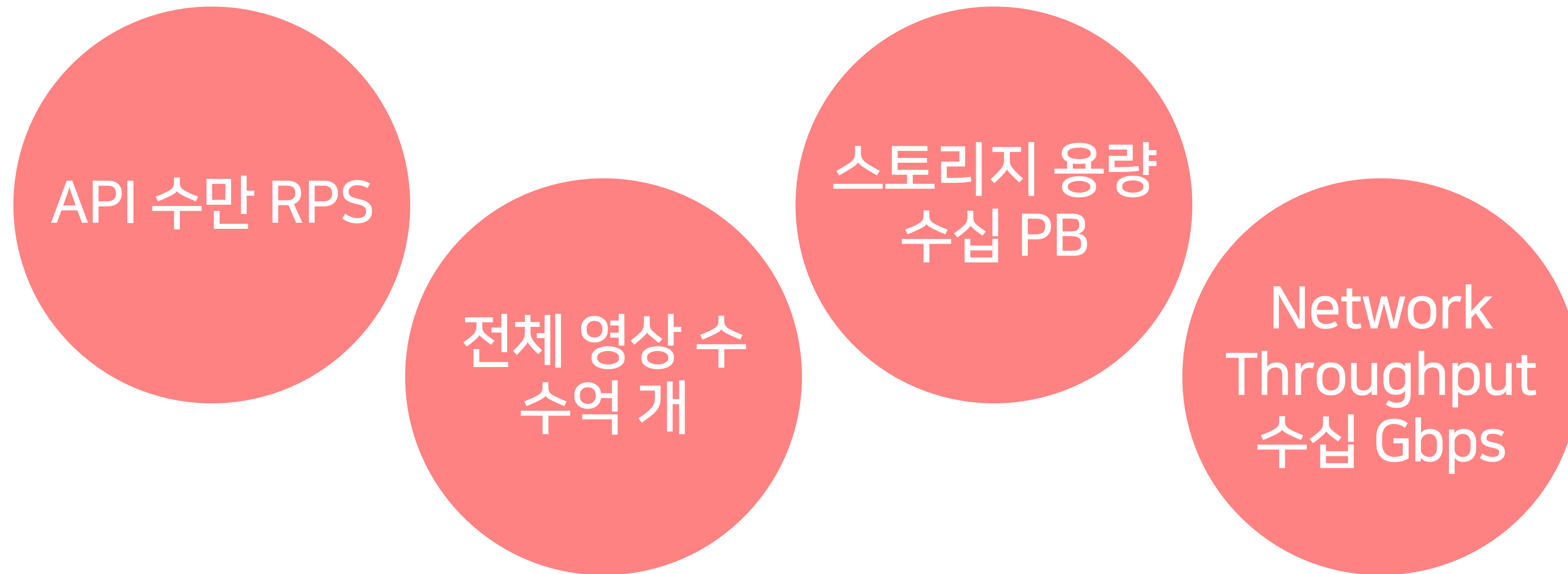


3. 어떻게 해결했는가?

3.1 대용량 트래픽을 어떻게 운영하는가?

VOD 클라우드의 볼륨

- API 요청수도 많지만, 업(Job)의 특성상 파일 볼륨이 크다



3.1 대용량 트래픽을 어떻게 운영하는가?

VOD 클라우드의 Lifecycle



더 자세히 알고 싶다면 'DEVIEW 2019 네이버 동영상 서비스를 지탱하는 VOD 플랫폼 개발기'를 참고하세요.

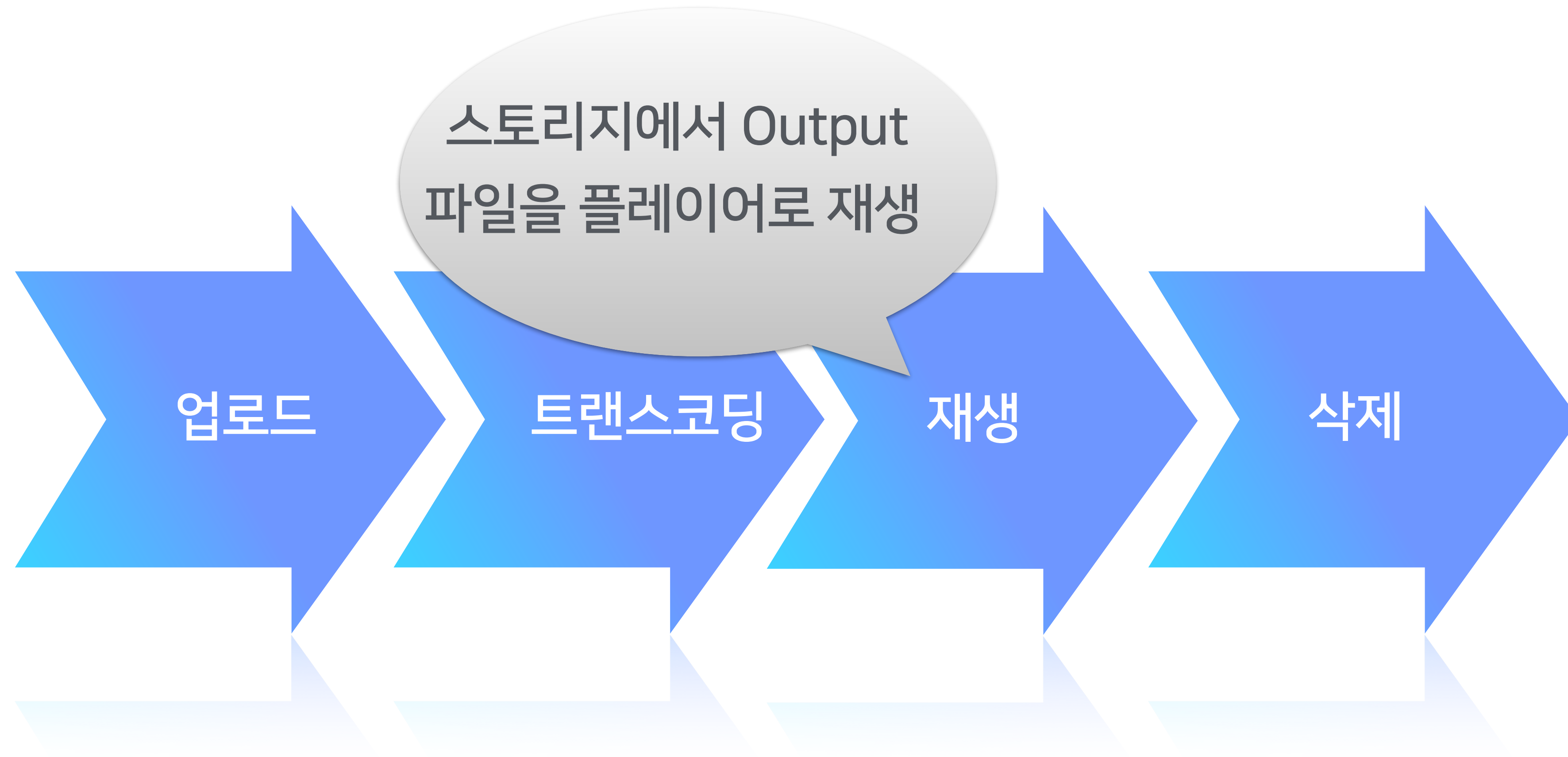
3.1 대용량 트래픽을 어떻게 운영하는가?

갑자기 대용량 트래픽이 들어오면 가장 약한 곳은?



3.1 대용량 트래픽을 어떻게 운영하는가?

갑자기 대용량 트래픽이 들어오면 가장 약한 곳은?



3.1 대용량 트래픽을 어떻게 운영하는가?

갑자기 대용량 트래픽이 들어오면 가장 약한 곳은?



3.1 대용량 트래픽을 어떻게 운영하는가?

갑자기 대용량 트래픽이 들어오면 가장 약한 곳은?

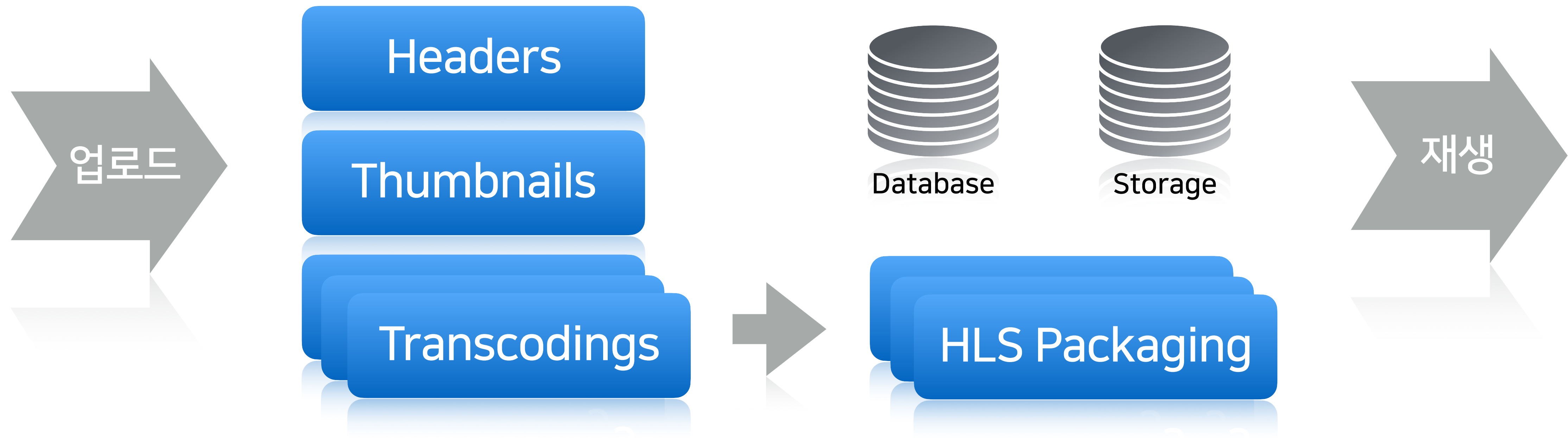
- 트랜스코딩 과정이 가장 오래, 리소스 많이, 복잡한 곳



3.1 대용량 트래픽을 어떻게 운영하는가?

가장 바쁘고 오래 걸리는 트랜스코딩

- 헤더, 섬네일, 트랜스코딩, Database / Storage 저장



3.1 대용량 트래픽을 어떻게 운영하는가?

가장 바쁘고 오래 걸리는 트랜스코딩

- 헤더, 섬네일, 트랜스코딩, Database / Storage 저장

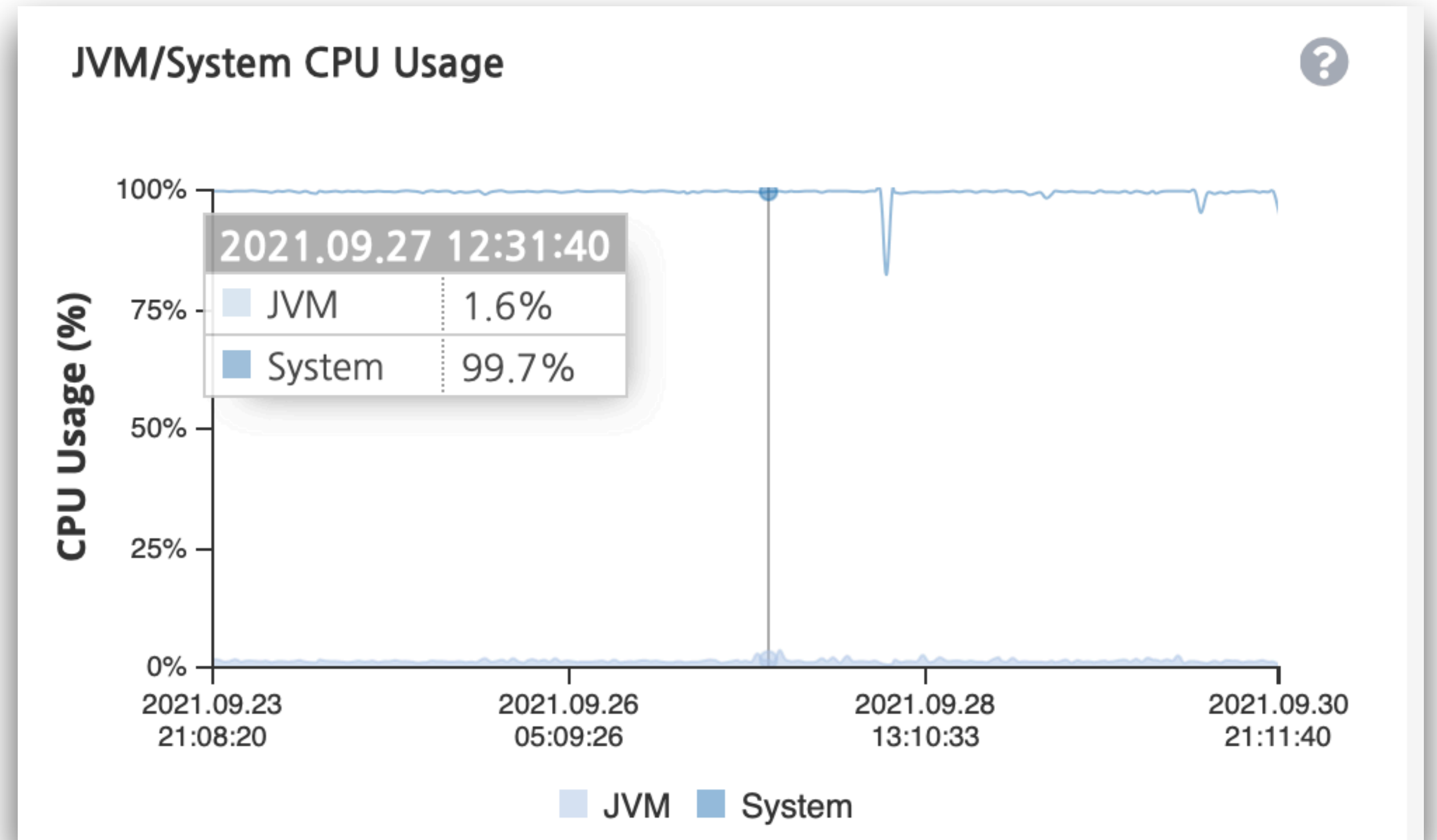


3.1 대용량 트래픽을 어떻게 운영하는가?

실 틈 없는 트랜스코딩 서버

- CPU 20 Core, 하이퍼스레딩
- 서버 CPU Usage 99.7%
- 전체 서버 수 약 수백 대

트랜스코딩 서버 1주일 CPU Usage



3.1 대용량 트래픽을 어떻게 운영하는가?

대용량 트래픽을 관리하는 다양한 방법들

가용량 및 Scaling

Scale-Up
Scale-Out
Autoscaling

접속 제어 및 부하 전파 차단

Circuit breaker
Load Shedding
Rate limiting

성능 및 캐시

WebFlux
JVM ZGC
Caching

아키텍처

Event Driven
MicroService

3.1 대용량 트래픽을 어떻게 운영하는가?

뿐만아닌 Event Driven Architecture 가 중요한 이유

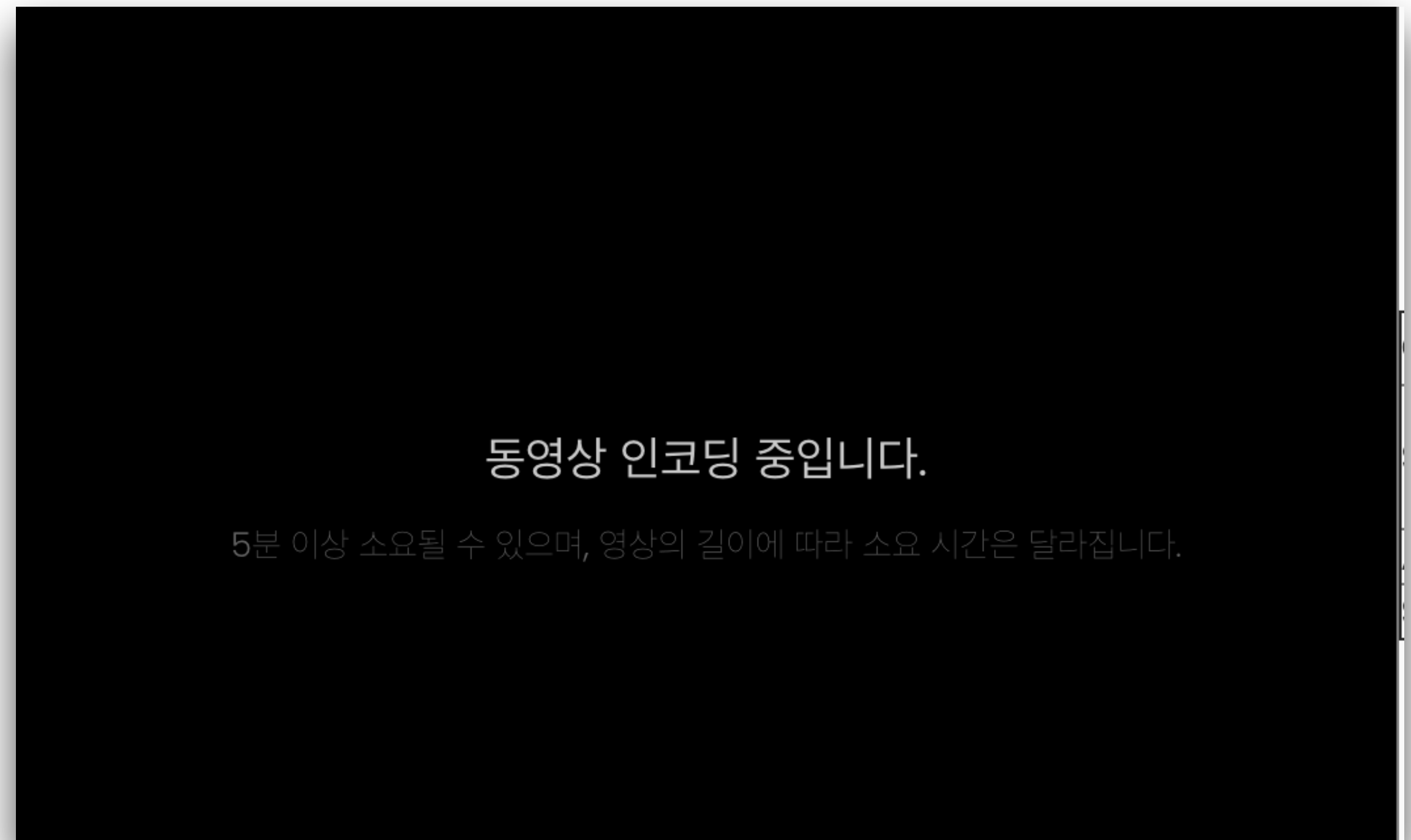
- VOD 클라우드의 Job 수행시간은 길다
- 서버 리소스를 많이 사용하며, Scale Out 한계가 있다
- 비동기로 Job 을 수행하며, 동시 처리수를 제한하는 수밖에 없다

3.1 대용량 트래픽을 어떻게 운영하는가?

모든 Job은 비동기로 처리

- VOD 만드는 것은 오래 걸리는 작업
- 기다릴 줄 아는 엔드 유저

플레이어에서 '인코딩 중입니다'라는 메시지 출력



3.1 대용량 트래픽을 어떻게 운영하는가?

Job에게도 우선순위가 있다

- RabbitMQ 의 priority 아규먼트 사용
- 같은 서비스에서 트래픽이 몰릴 경우 priority를 낮춤

{priority : 8} 입력

급한 Message

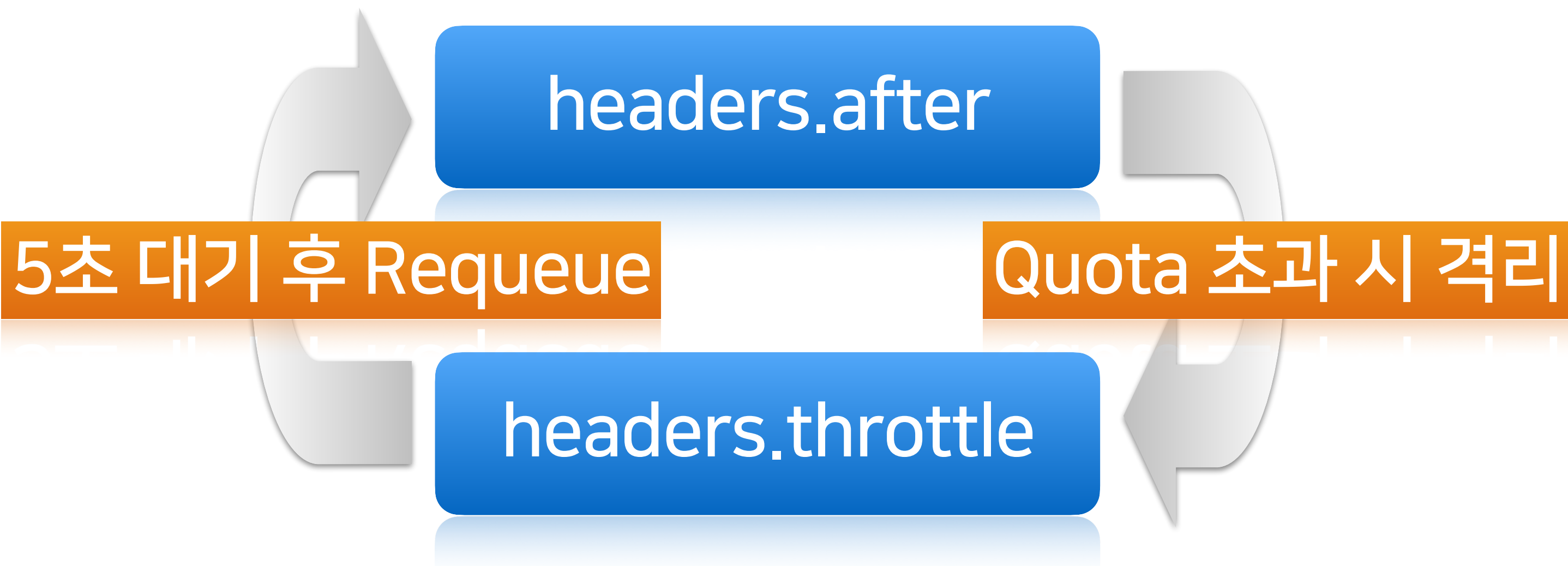
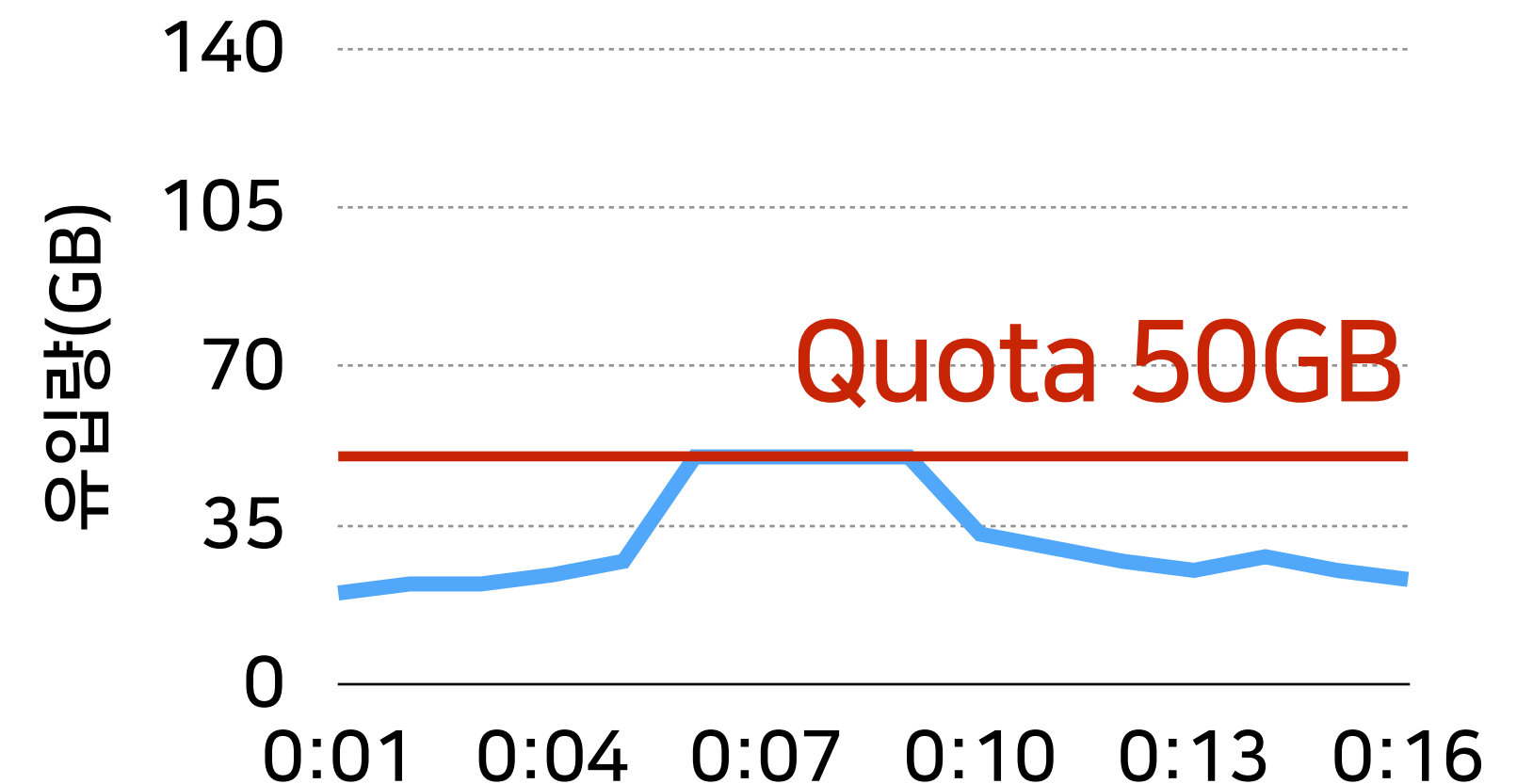
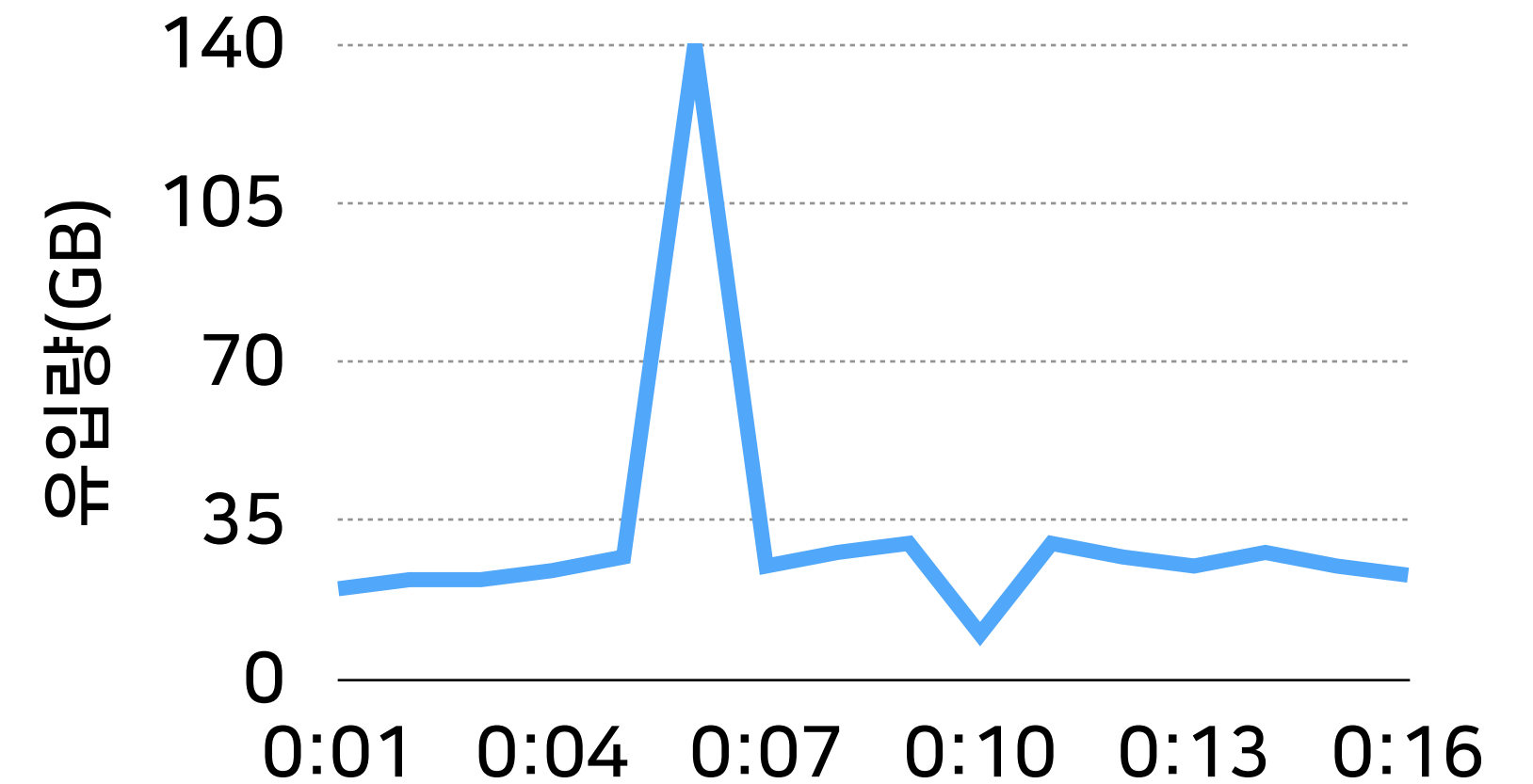
{priority : 3} 입력

덜 급한 Message

3.1 대용량 트래픽을 어떻게 운영하는가?

특정 서비스의 리소스 과다 점유 방지

- RabbitMQ 의 DeadLetter 활용
- 서비스 실시간 지표를 활용하여 Quota 적용



3.2 멀티 리전 트래픽 라우팅

분산된 스토리지의 등장으로 복잡해짐

AS-IS
모든 영상 정보를
한국 리전에만 저장

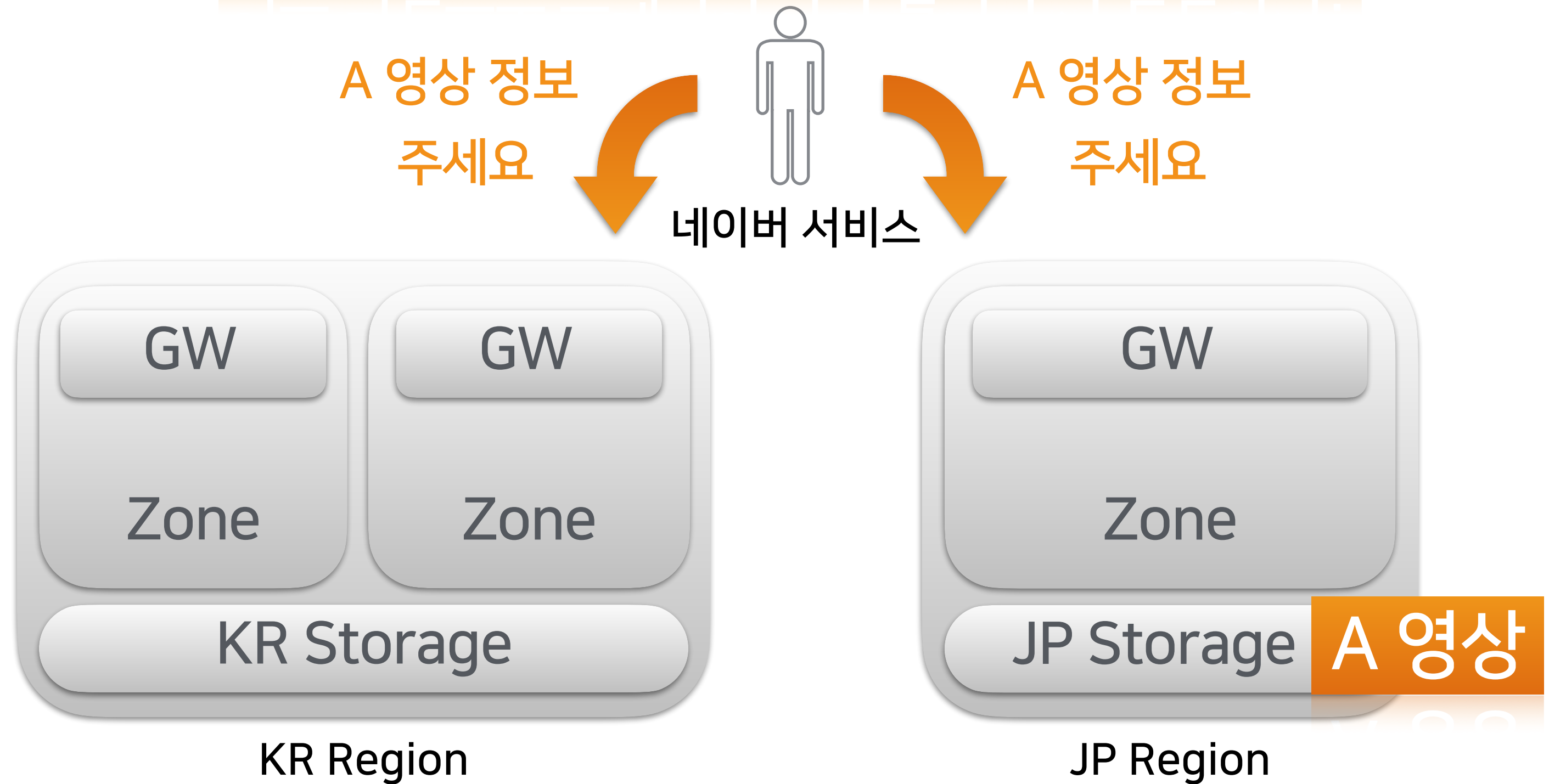


3.2 멀티 리전 트래픽 라우팅

분산된 스토리지의 등장으로 복잡해짐

어느 리전으로 트래픽이 가야 할지 누가 판단하지?

TO-BE
영상 정보를
리전별로 따로 저장



3.2 멀티 리전 트래픽 라우팅

트래픽을 누가 분배해야 할까?

네이버 서비스 제어
VS
VOD 클라우드 제어

3.2 멀티 리전 트래픽 라우팅

트래픽을 누가 분배해야 할까?

네이버 서비스 제어

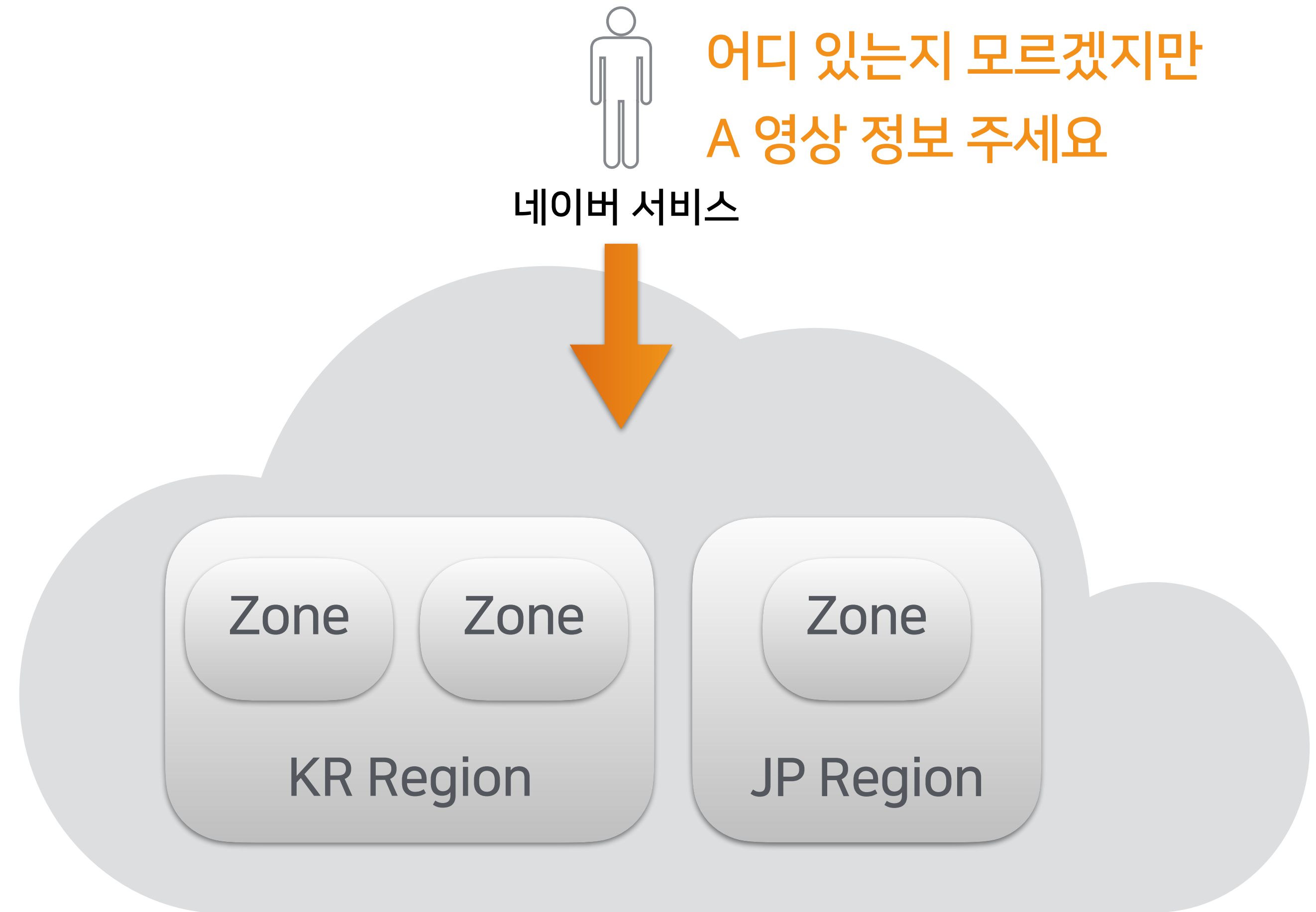
VS

VOD 클라우드 제어

3.2 멀티 리전 트래픽 라우팅

왜 VOD 클라우드가?

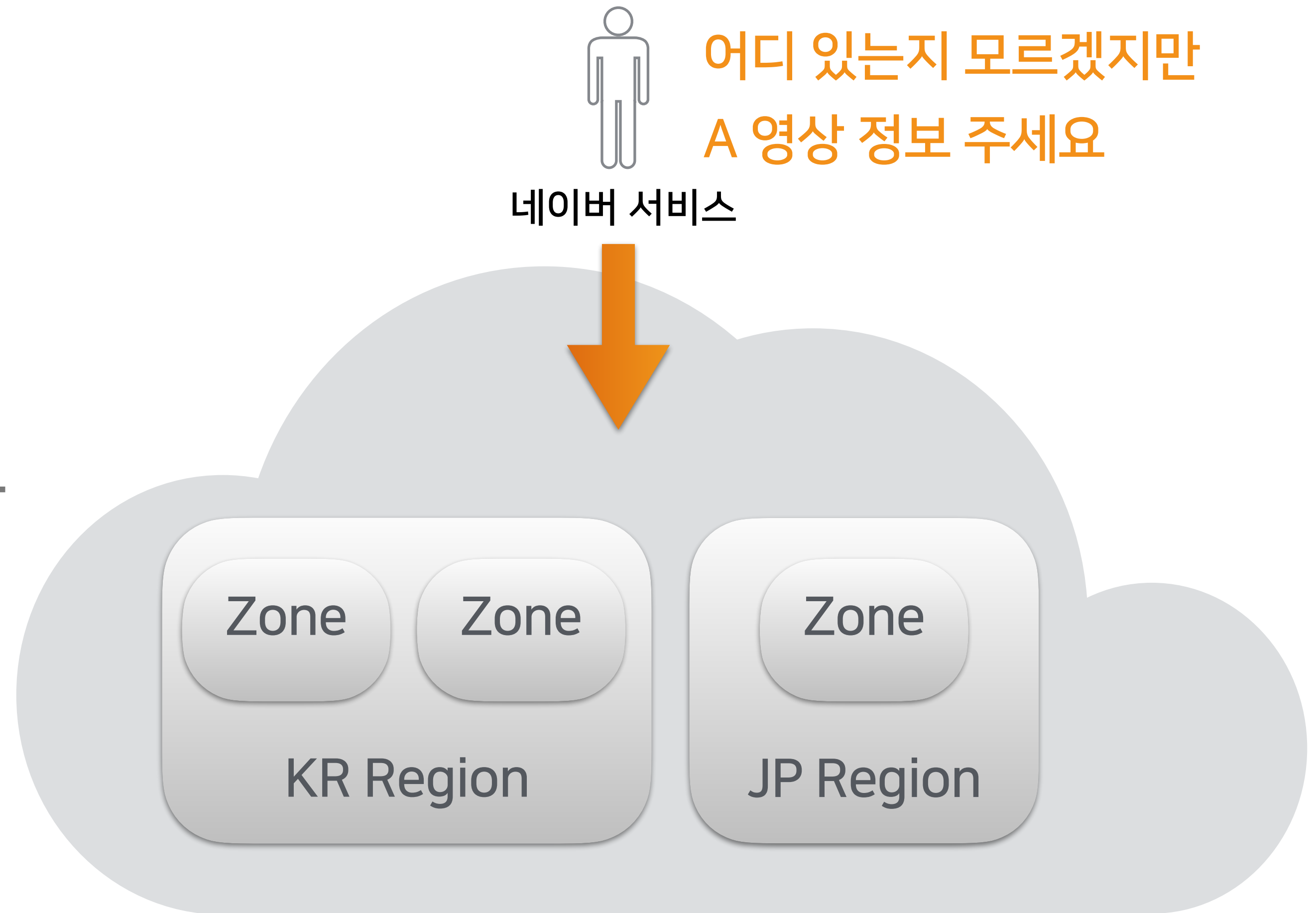
- 네이버 서비스 측면
- 더 빠른 VOD 클라우드 연동
- 더 빠른 서비스 릴리즈



3.2 멀티 리전 트래픽 라우팅

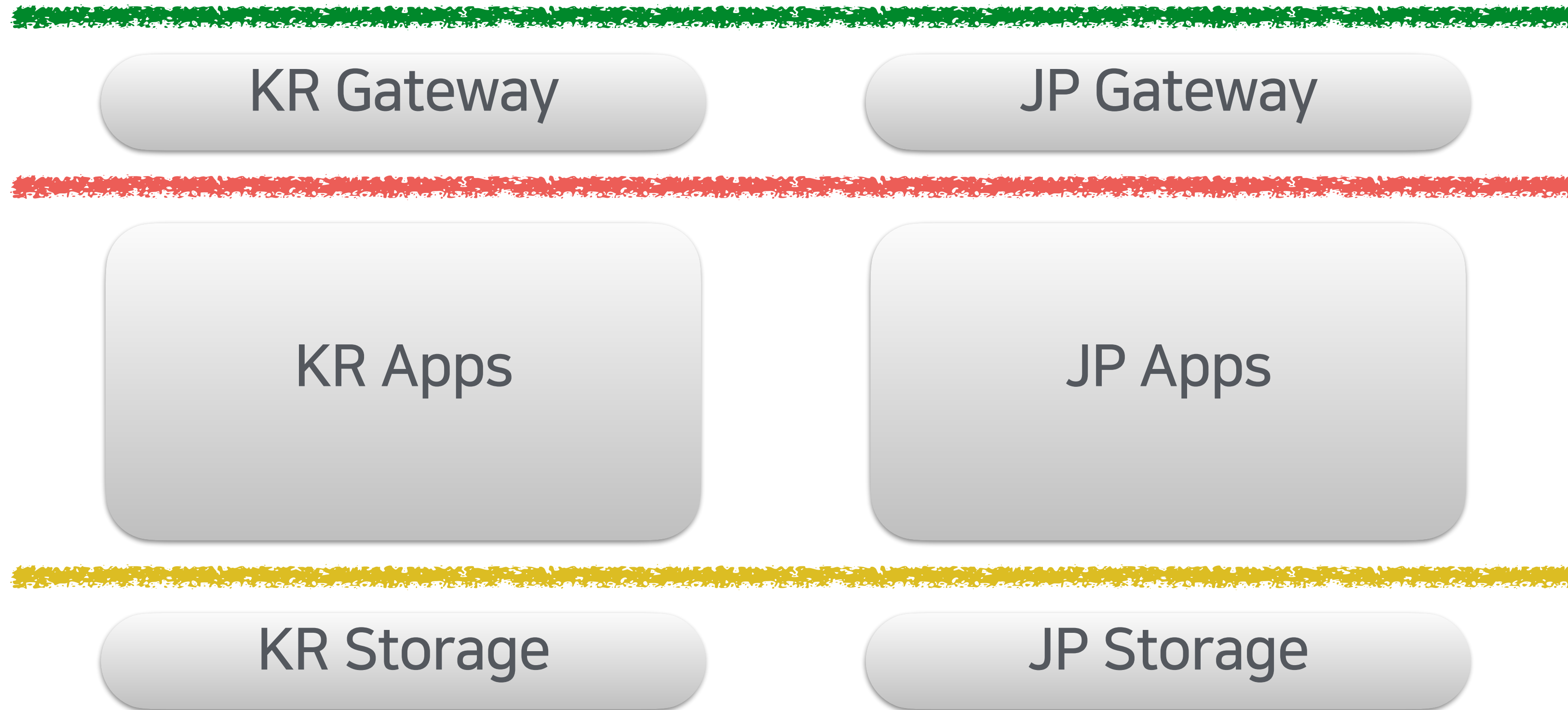
왜 VOD 클라우드가?

- 네이버 서비스 측면
- 더 빠른 VOD 클라우드 연동
- 더 빠른 서비스 릴리즈
- VOD 클라우드 측면
- 서비스 커뮤니케이션 비용 절감



3.2 멀티 리전 트래픽 라우팅

어디서 분배해야 할까?



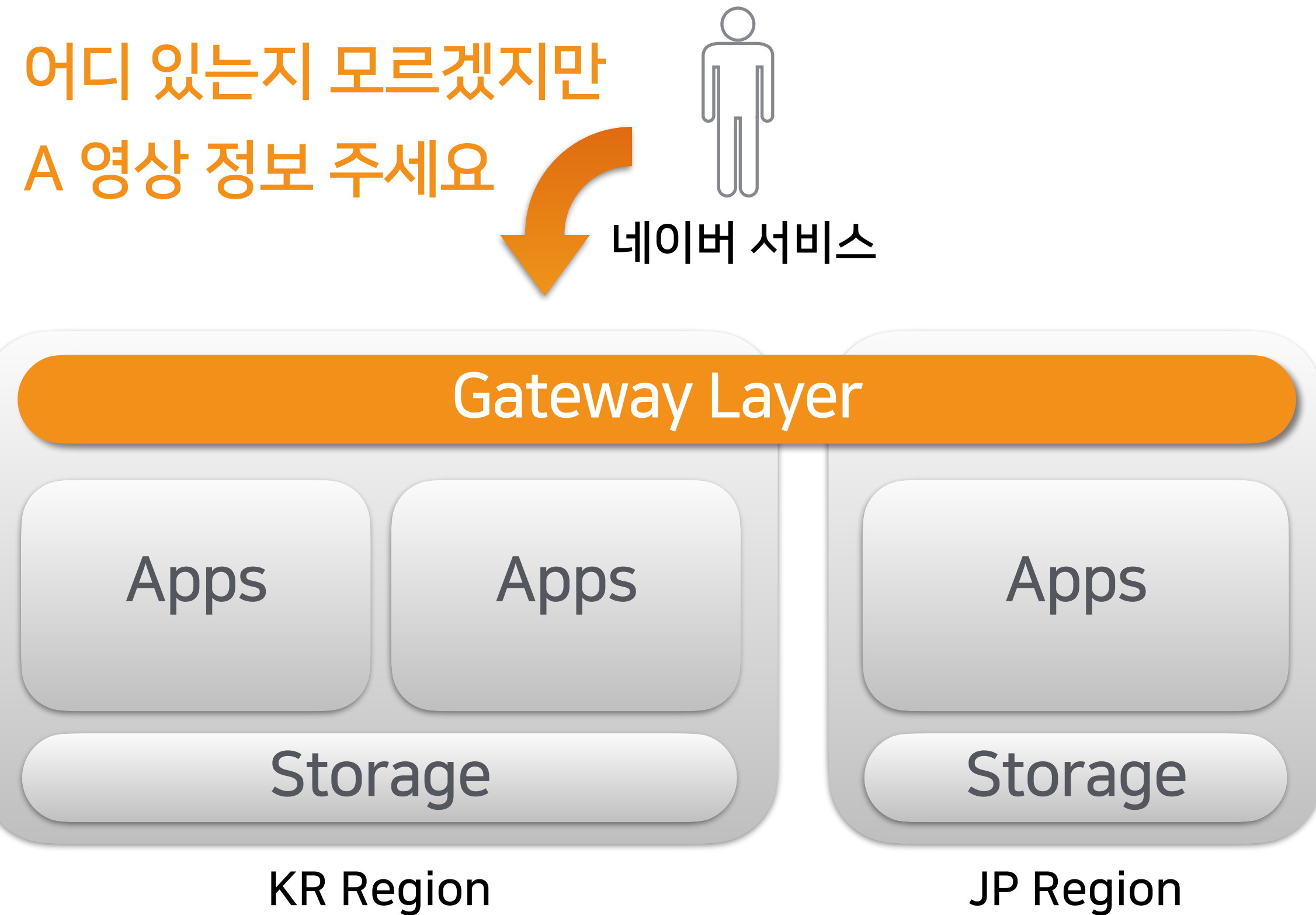
3.2 멀티 리전 트래픽 라우팅

어디서 분배해야 할까?

- A : 게이트웨이

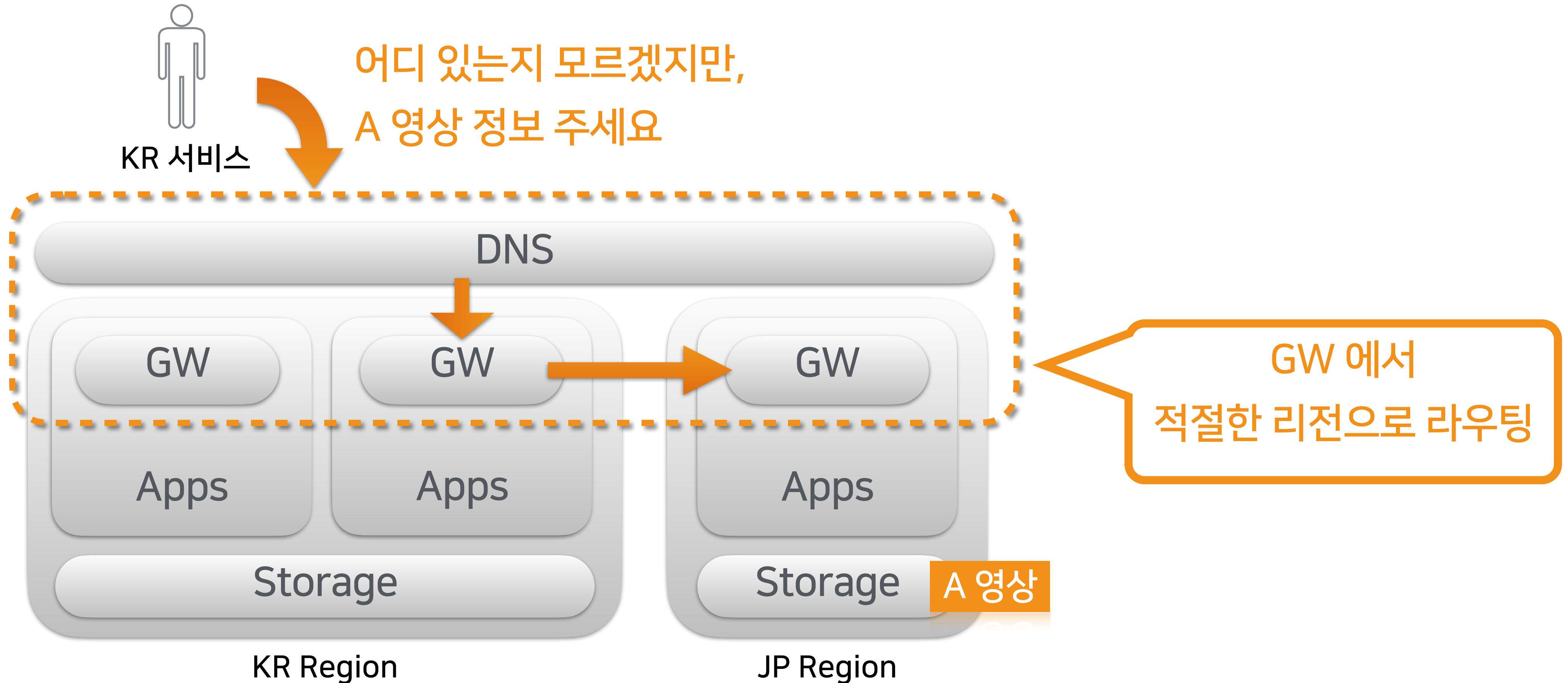
왜?

- 변경 최소화, 빠른 개발
- 리전 인입 트래픽 제어



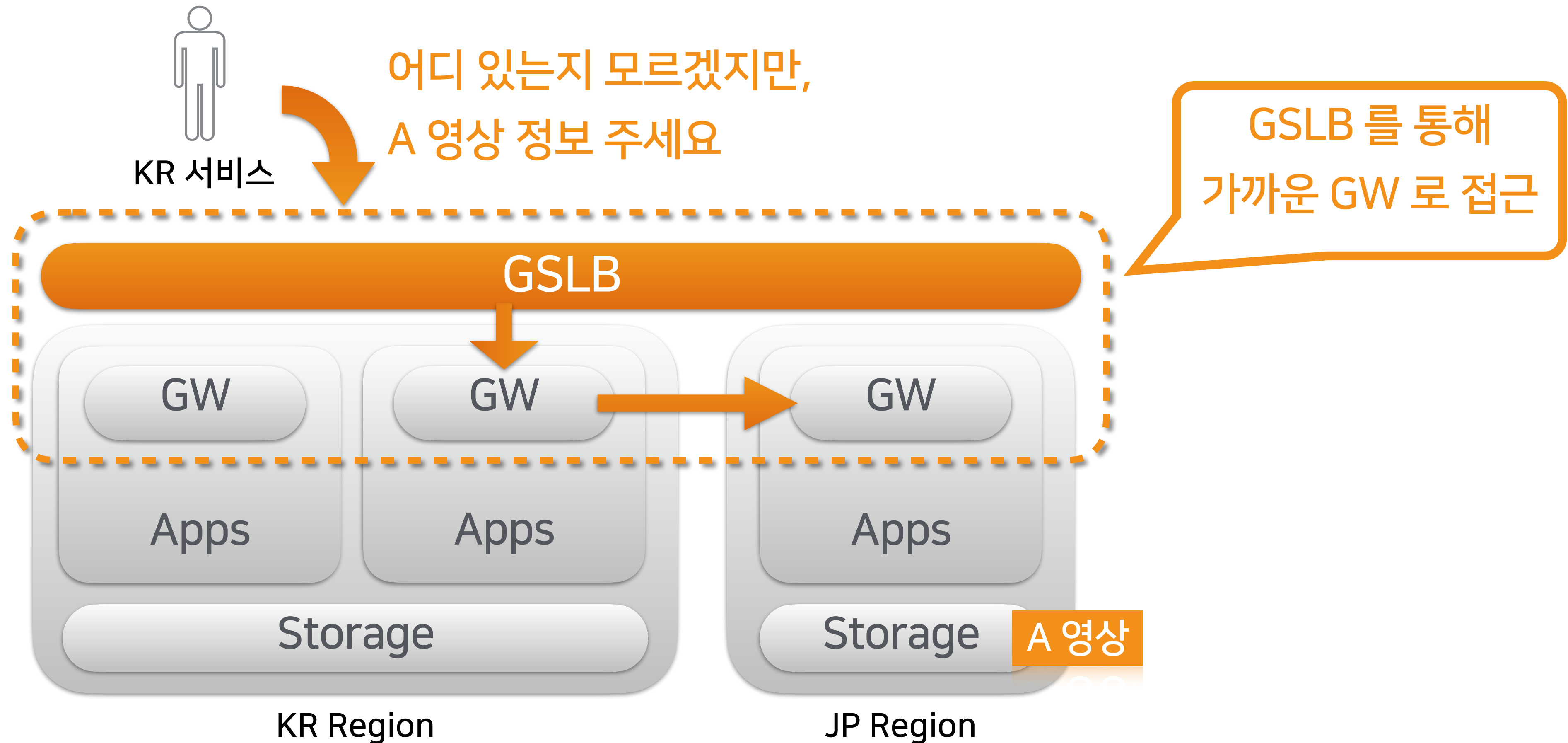
3.2 멀티 리전 트래픽 라우팅

한국에서 일본에 저장된 A 영상 찾기



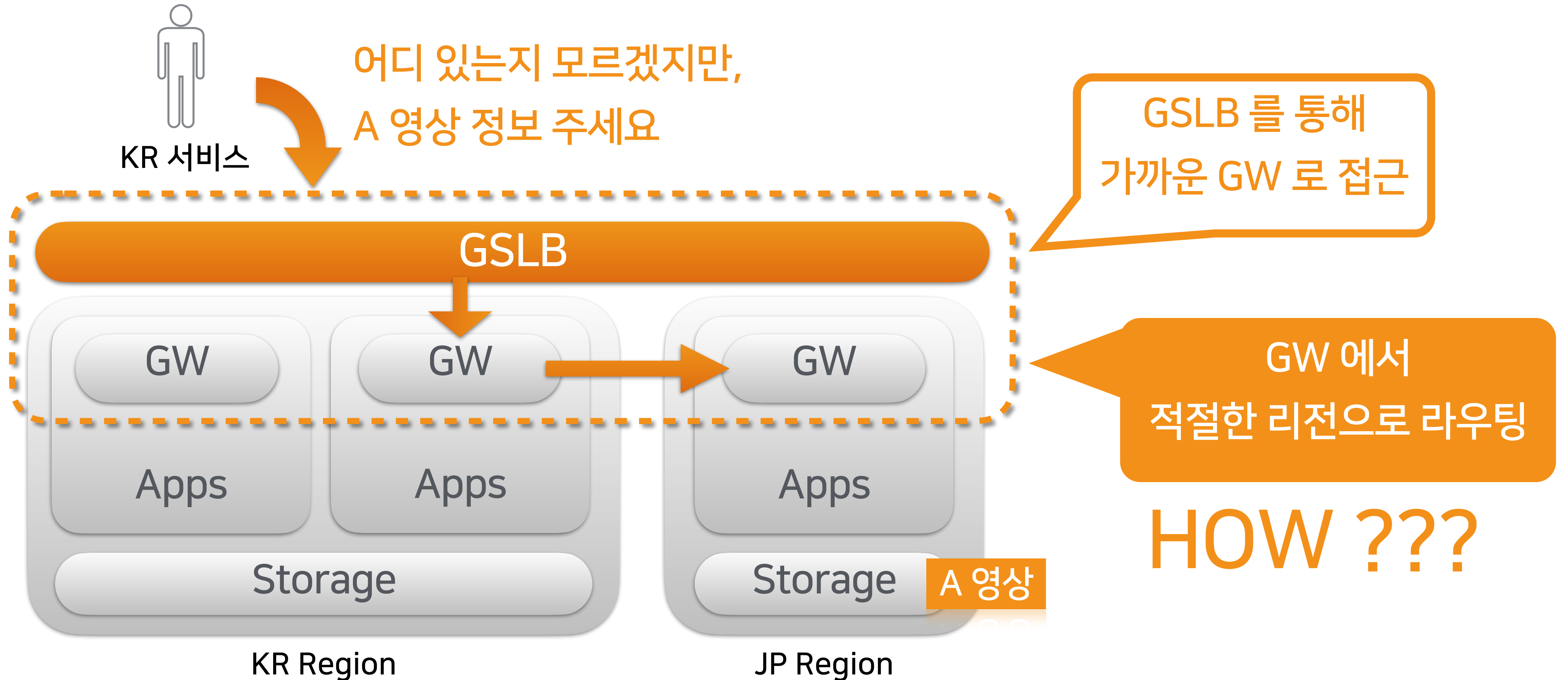
3.2 멀티 리전 트래픽 라우팅

DNS? 비효율적인 것 아니야?



3.2 멀티 리전 트래픽 라우팅

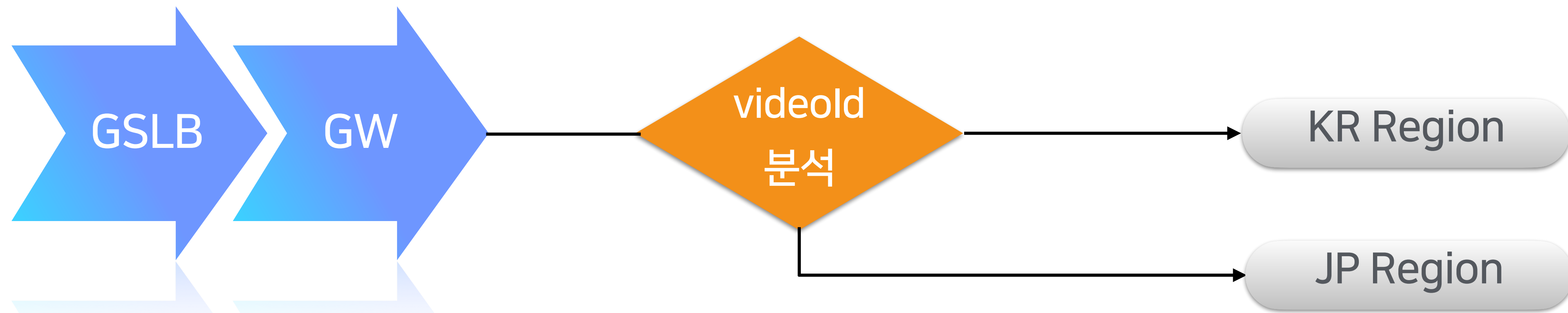
어떻게 라우팅하지?



3.2 멀티 리전 트래픽 라우팅

어떻게 라우팅하지?

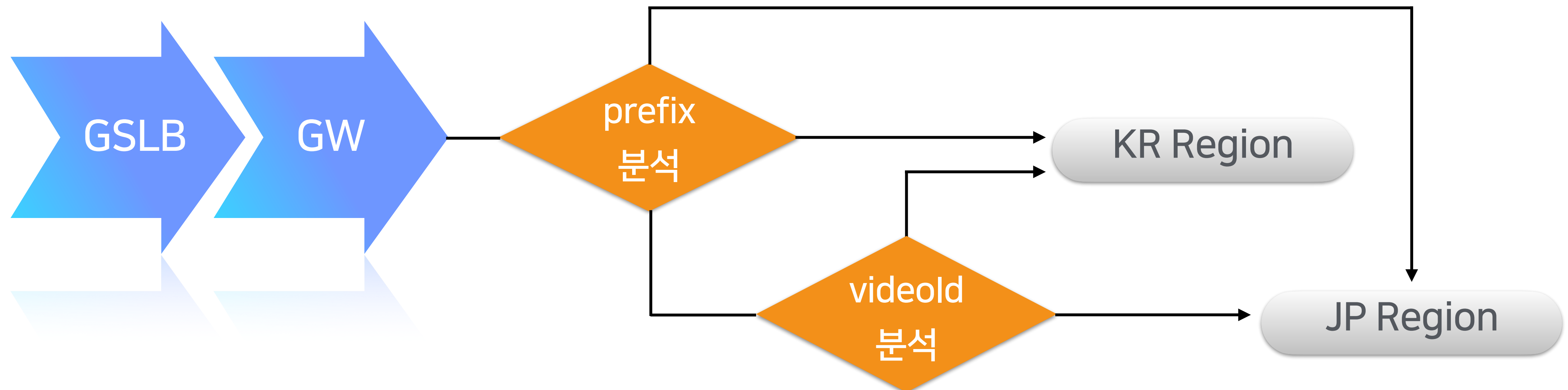
- 영상 고유키 기반 라우팅
- `/resource/{videoid}` 형식의 기존 패스 구성 활용
- * videoid : VOD 클라우드 영상의 고유키. 리전 정보를 추가로 심어 암호화



3.2 멀티 리전 트래픽 라우팅

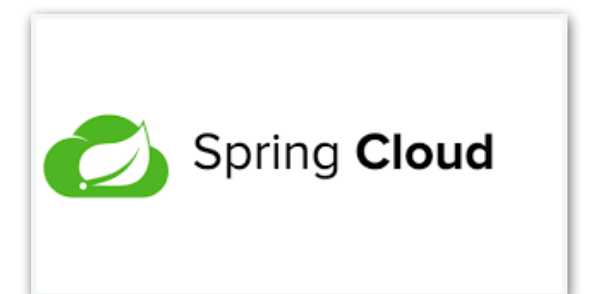
어떻게 라우팅하지?

- Prefix 기반 라우팅
- videold 가 패스에 없는 경우 적용
ex) /kr/{resource}/..., /jp/{resource}/...



3.2 멀티 리전 트래픽 라우팅

어떻게 구현했나?



3.2 멀티 리전 트래픽 라우팅

어떻게 구현했나?

Eureka :

Spring Cloud Netflix

MSA

Service Discovery

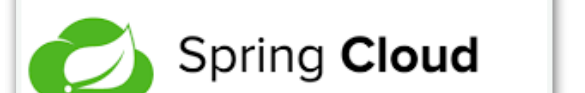
사용성

안정성

Region/Zone

Spring Cloud

Gateway :

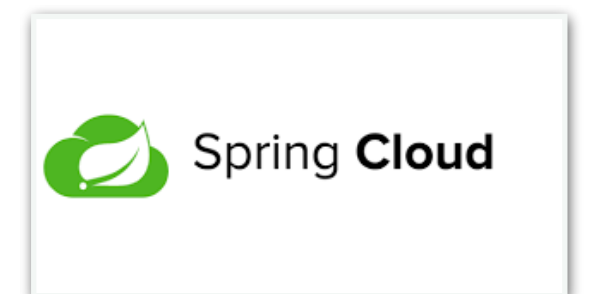


3.2 멀티 리전 트래픽 라우팅

어떻게 구현했나?

Eureka :
Spring Cloud Netflix
MSA
Service Discovery
사용성
안정성
Region/Zone

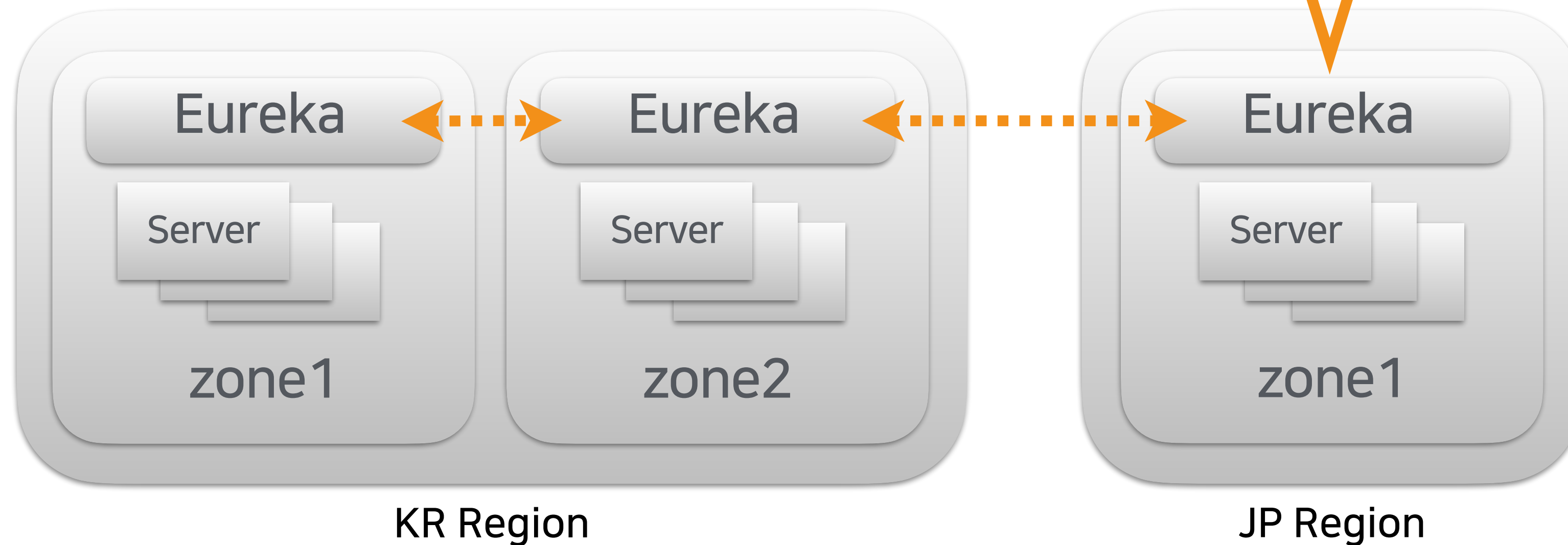
**Spring Cloud
Gateway :**
API Gateway
SCG **VS** Netflix Zuul
WebFlux
Netty
Region/Zone



3.2 멀티 리전 트래픽 라우팅

어떻게 구현했나?

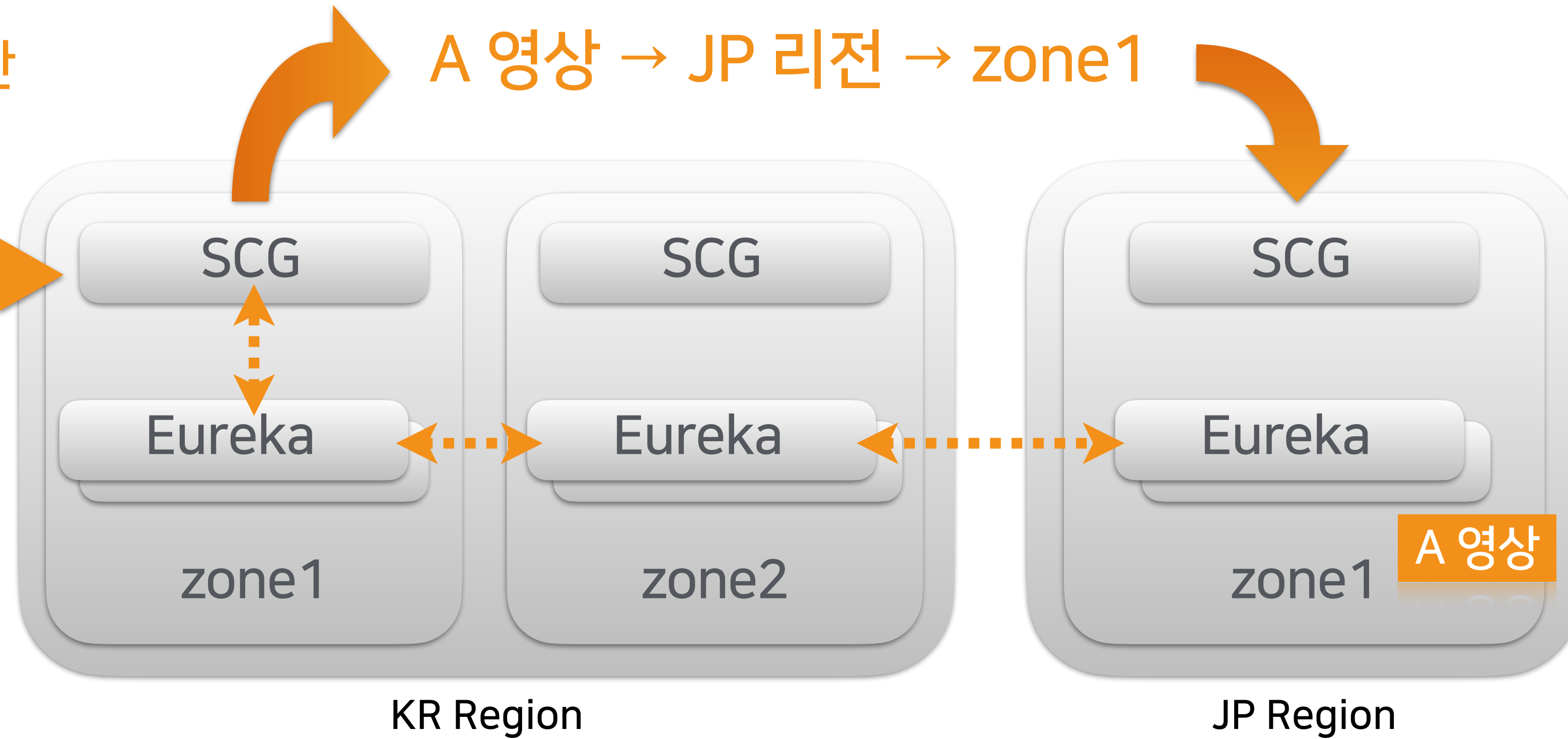
sever	zone	region
svr01	zone1	JP
svr02	zone1	JP
svr03	zone1	JP
...



3.2 멀티 리전 트래픽 라우팅

어떻게 구현했나?

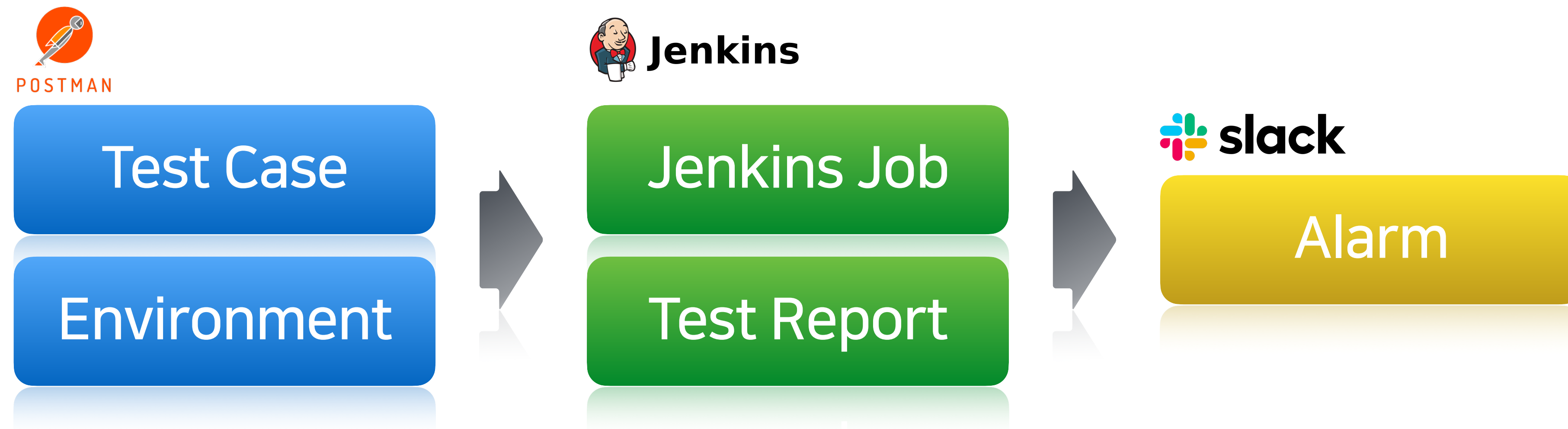
어디 있는지 모르겠지만
A 영상 정보 주세요



3.3 안정성 보장은 어떻게 하지?

통합 테스트 자동화로 안정성 강화

- 한국, 일본 리전에서 매일 평균 수백 건의 TC 수행
- 개발 속도와 코드 품질을 높임



3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

- Zone 장애나면 ?
- 리전 내 건강한 다른 Zone 으로 트래픽 분배

영향 범위	장애 대상	대응
App	Eureka 연동 서버 인스턴스	(자동) Service Discovery 를 통해 장애 서버 제외
Zone	Eureka 연동 앱	(자동) Service Discovery 를 통해 장애 앱 제외 다른 Zone 우회
	도메인 사용 앱	DNS 연동 설정 변경 다른 Zone 우회
	CDN Origin	CDN 연동 설정 변경 다른 Zone 우회
	스토리지(NAS)	서비스 설정 변경 다른 Zone 우회
Region		

더 자세히 알고 싶다면 'DEVIEW 2019 네이버 동영상 서비스를 지탱하는 VOD 플랫폼 개발기'를 참고하세요.

3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

- Zone 장애나면 ?
 - 리전 내 건강한 다른 Zone 으로 트래픽 분배
- Region 장애나면 ?
 - 리전 내 고립

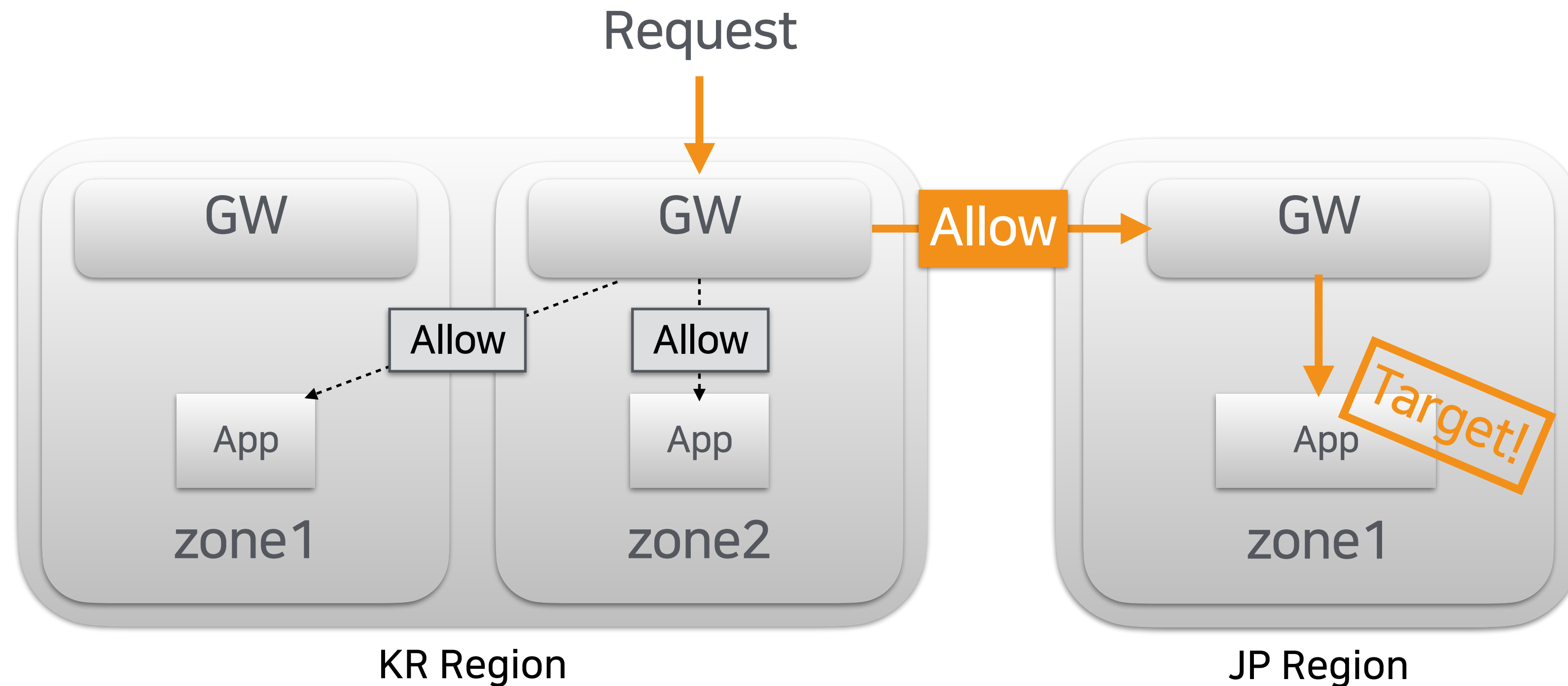
영향 범위	장애 대상	대응
App	Eureka 연동 서버 인스턴스	(자동) Service Discovery 를 통해 장애 서버 제외
Zone	Eureka 연동 앱	(자동) Service Discovery 를 통해 장애 앱 제외 다른 Zone 우회
	도메인 사용 앱	DNS 연동 설정 변경 다른 Zone 우회
	CDN Origin	CDN 연동 설정 변경 다른 Zone 우회
	스토리지(NAS)	서비스 설정 변경 다른 Zone 우회
Region	어떠한 장애든	Region 내에서 처리

더 자세히 알고 싶다면 'DEVIEW 2019 네이버 동영상 서비스를 지탱하는 VOD 플랫폼 개발기'를 참고하세요.

3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

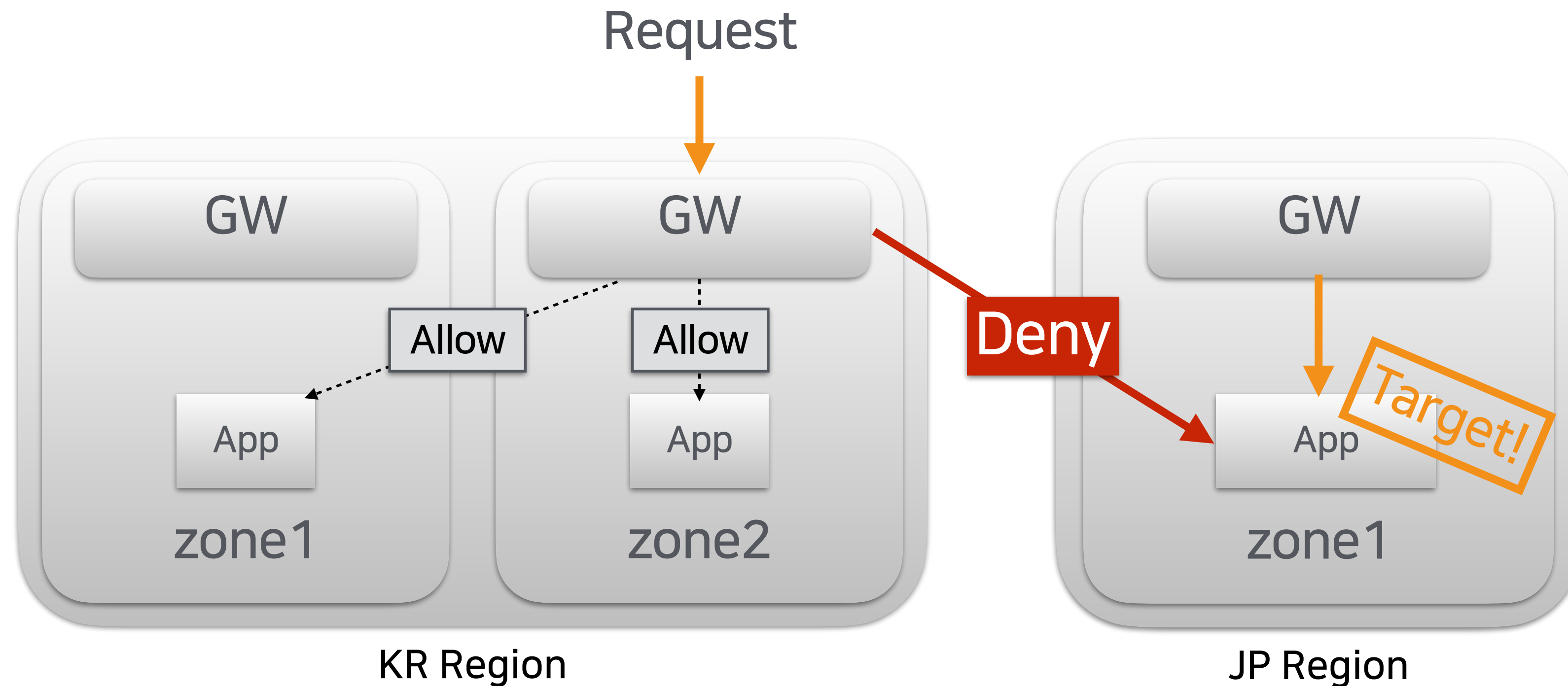
- 게이트웨이를 통한 리전 장애 고립시키기



3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

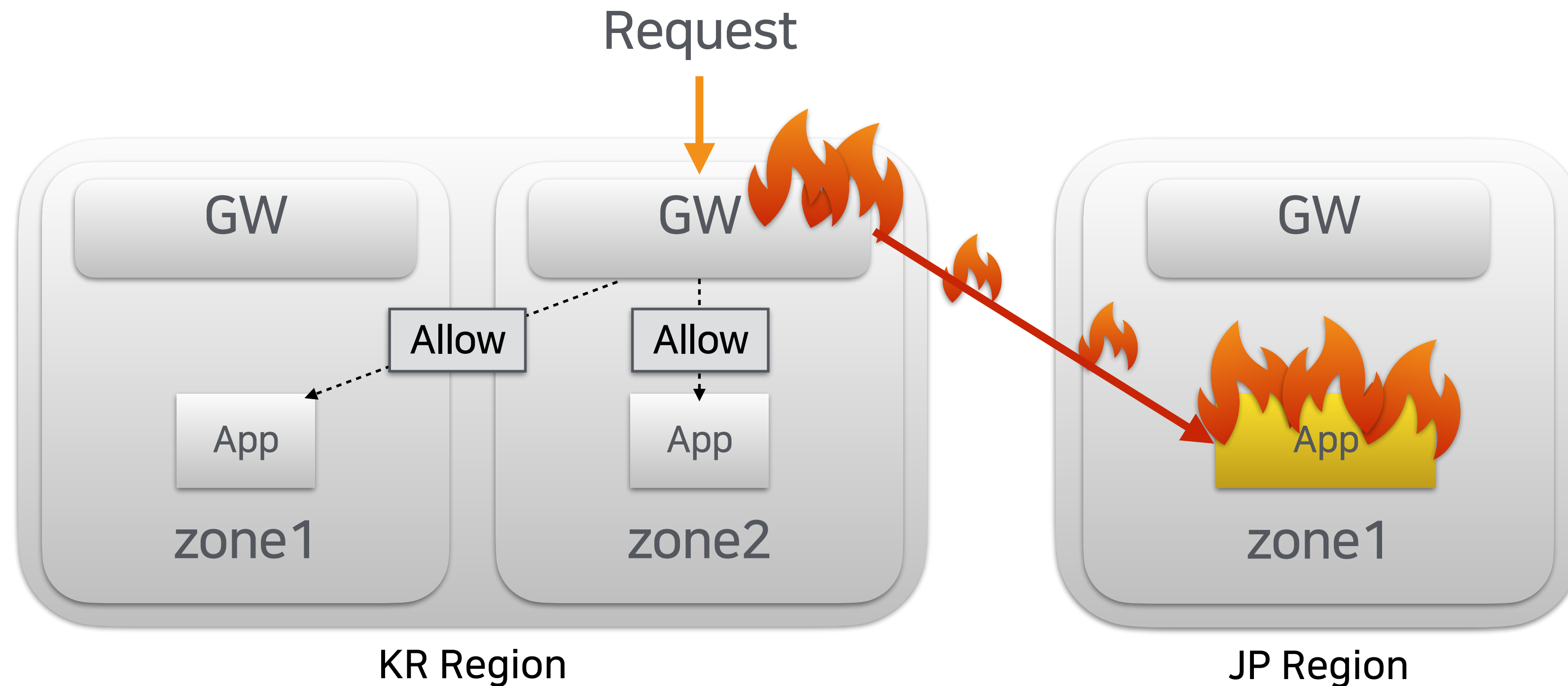
- 게이트웨이를 통한 리전 장애 고립시키기



3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

- 게이트웨이를 통한 리전 장애 고립시키기



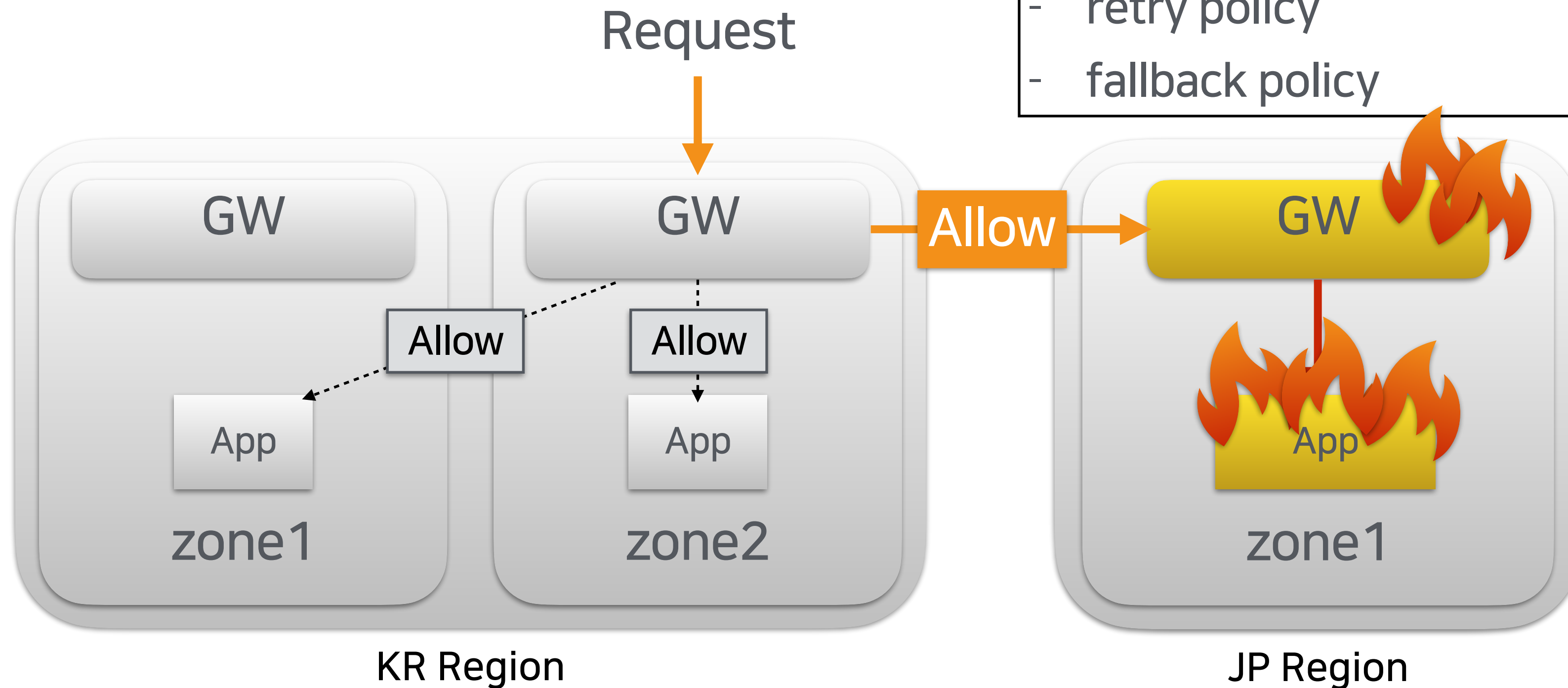
3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

- 게이트웨이를 통한 리전 장애 고립시키기

제어 포인트

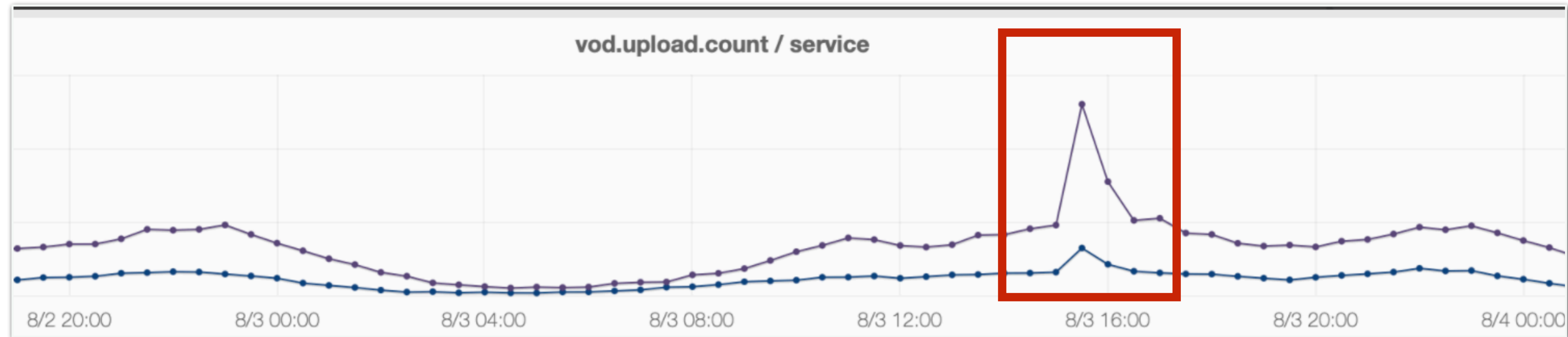
- time out
- circuit breaker pattern
- rate limiter pattern
- retry policy
- fallback policy



3.3 안정성 보장은 어떻게 하지?

멀티 리전 장애 대응 시나리오

- 게이트웨이를 통한 리전 장애 고립시키기
- 사례 : API 서버가 Load Shedding 이 필요한 상황, 리전 GW에서 트래픽 컨트롤



4. 적용 후기

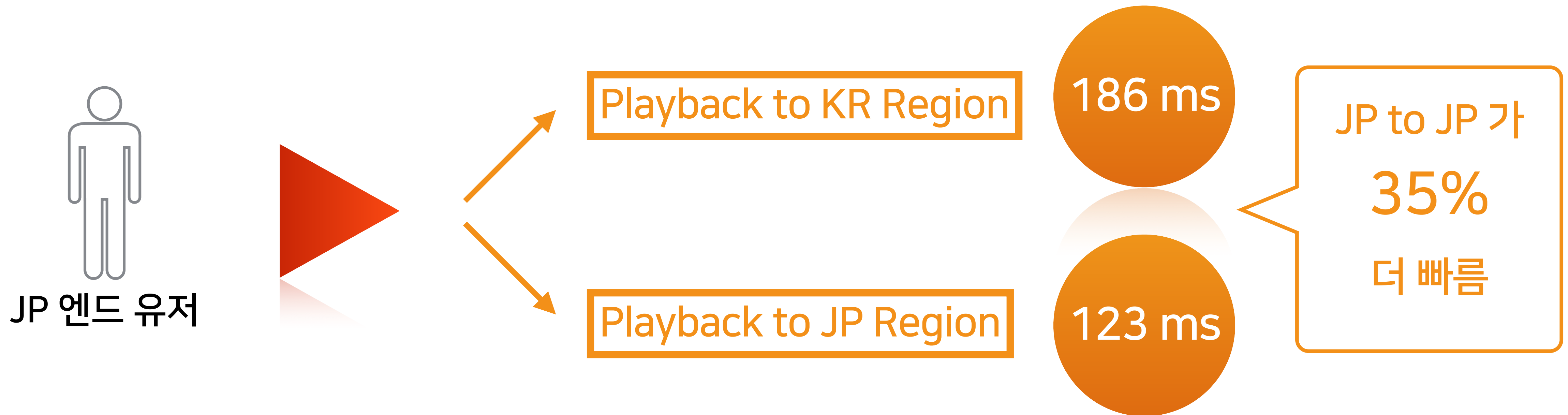
4.1 Latency 가 얼마나 좋아졌을까?

한국, 일본 업로드 수행 시간 비교



4.1 Latency 가 얼마나 좋아졌을까?

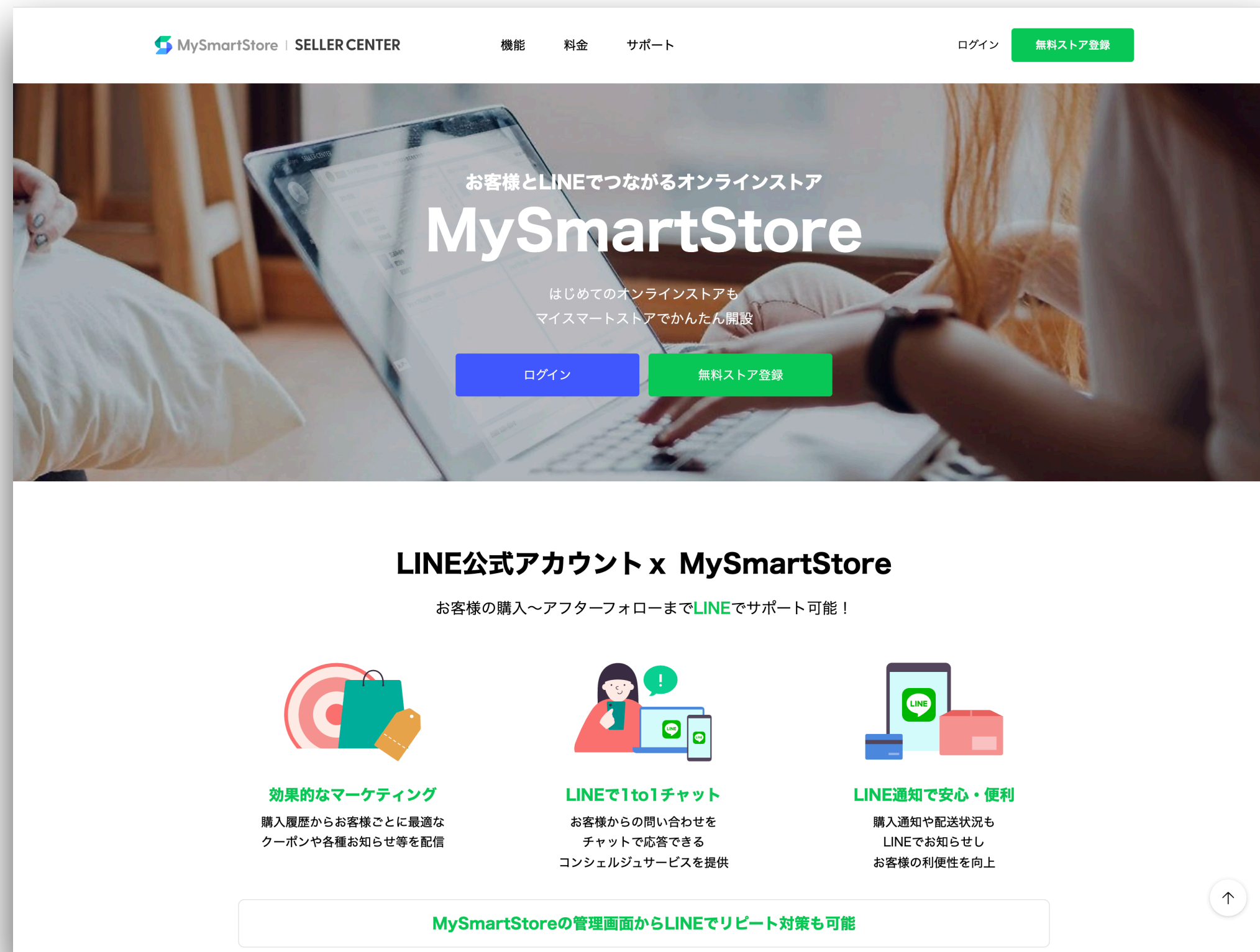
한국, 일본 재생 API Latency 비교



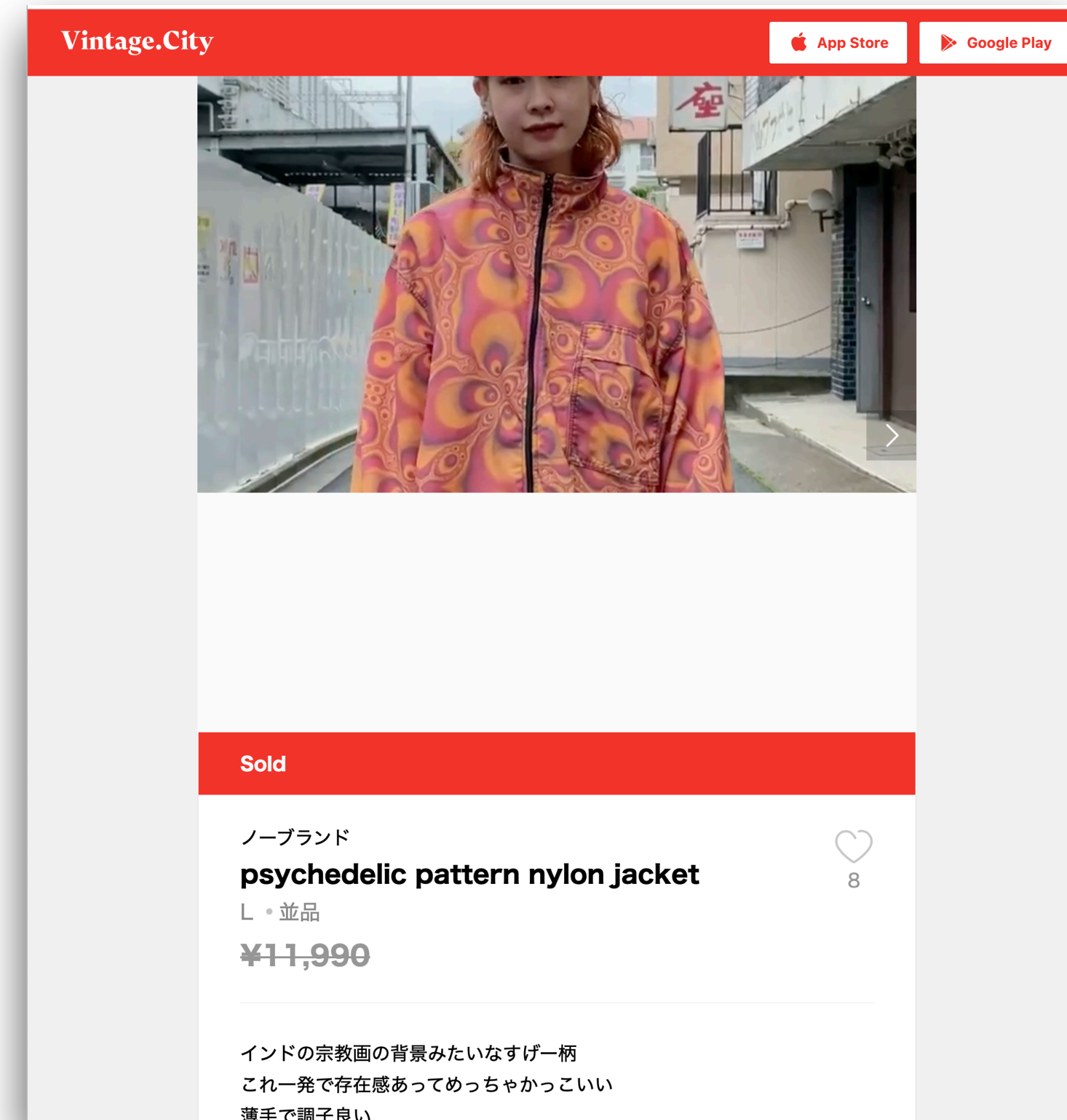
4.2 누가 쓰고 있을까?

일본 리전 VOD 클라우드 사용하는 네이버 서비스

MySmartStore



Vintage.City



4.3 아쉬웠던 점

멀티 리전 운영 비용 증가

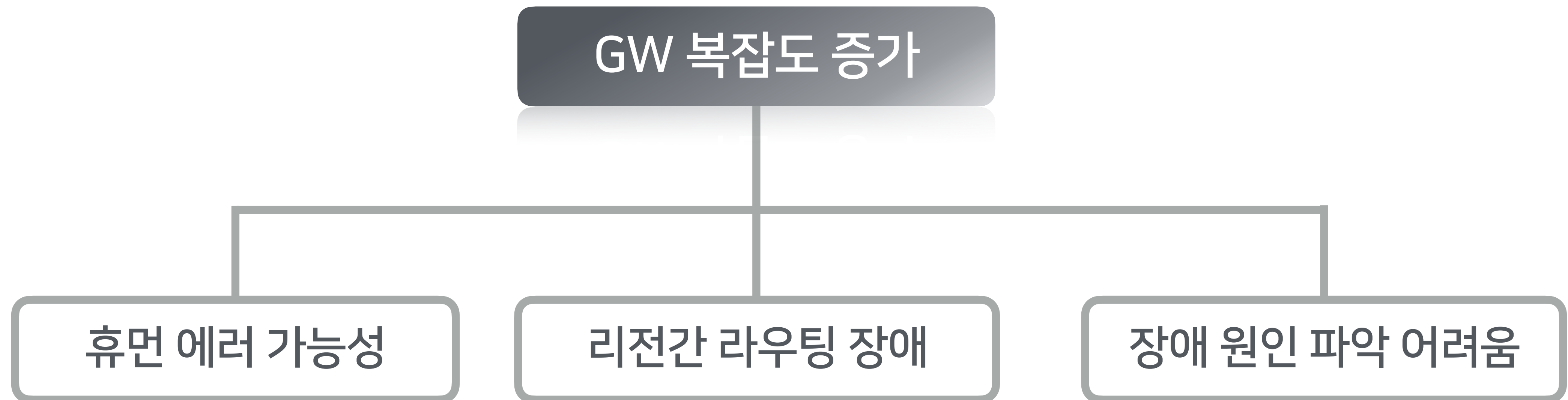
- 운영 비용을 줄이기위해 노력은 하였으나...
- 한국 대비 일본 지표 비교



4.3 아쉬웠던 점

구현 복잡도 증가

- 힘이 좀 많이 들어간 게이트웨이
- 사례 : GW 에서 서비스 요청을 전체 리전에 분산 라우팅



4.4 저희도 멀티 리전을 이렇게 해야 하나요?

진리의 “케바케”

요구사항, 플랫폼 구조, 비용 등을 종합적으로 고려하여 판단

NAVER

Emerging

TECHnology

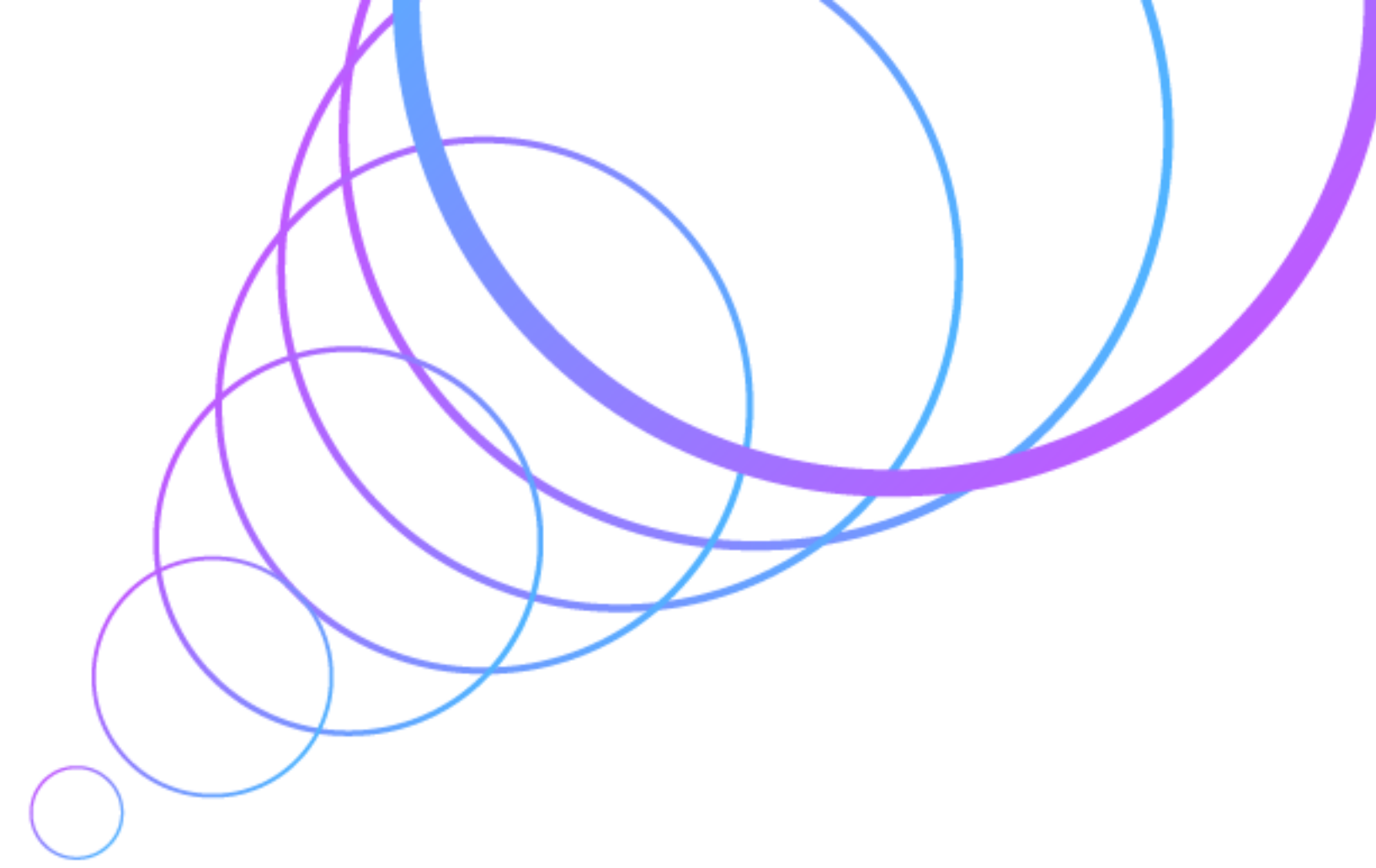
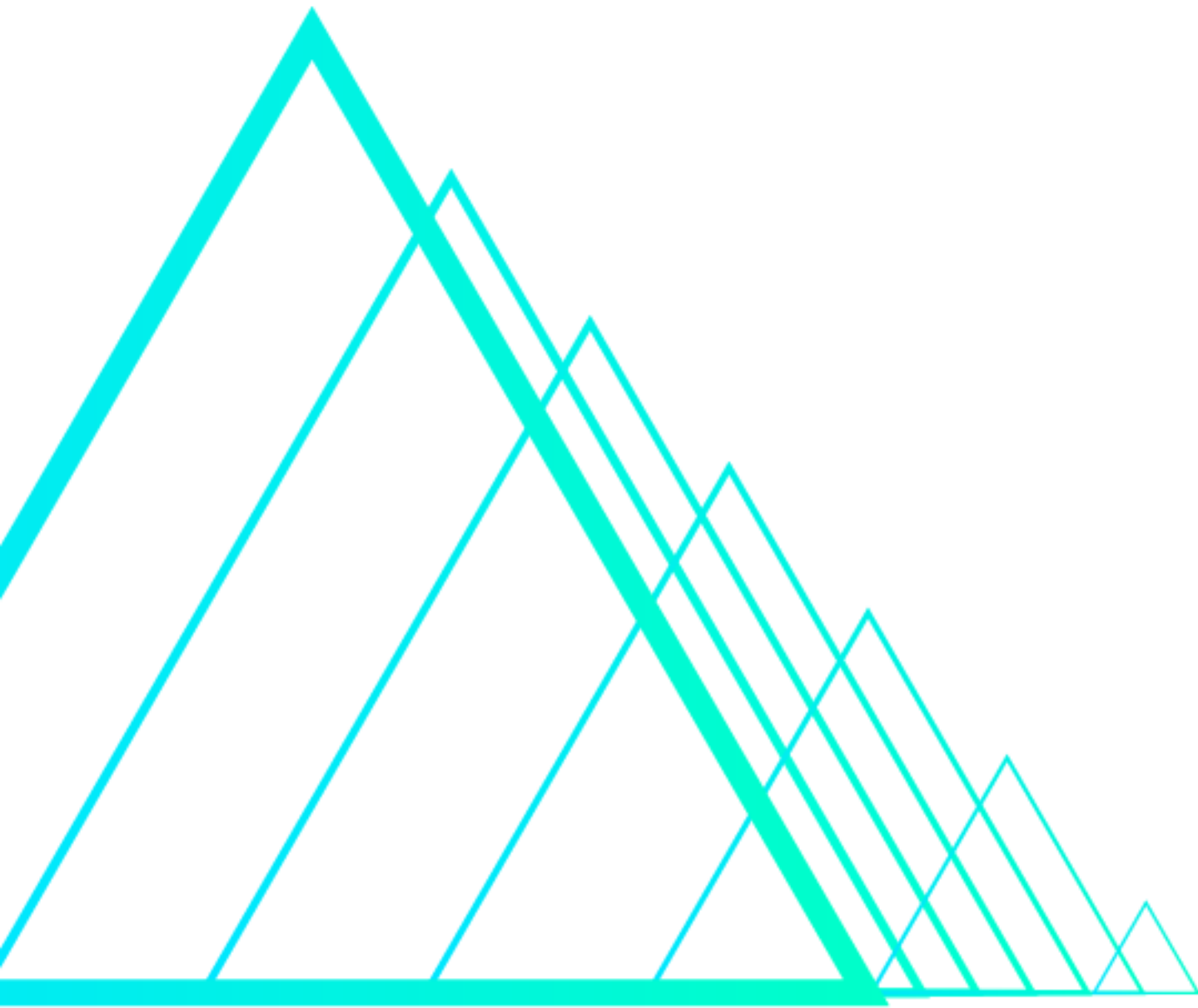
함께 성장하실 동료분을 모십니다

FE / BE / 앱 / SDK 개발 기술과 함께 멀티미디어의 핵심 기술을 배우고,
포토 / 오디오 / 비디오 / UGC 기술 도메인 전문가로 성장할 수 있도록 적극 지원하겠습니다.



ETECH 직무 소개

- ETECH 직무 소개 : <https://naver-career.gitbook.io/kr/service/etech>
- ETECH 기술 문의 : etech@navercorp.com
- ETECH 채용 문의 : etech-recruit@navercorp.com



Thank You

