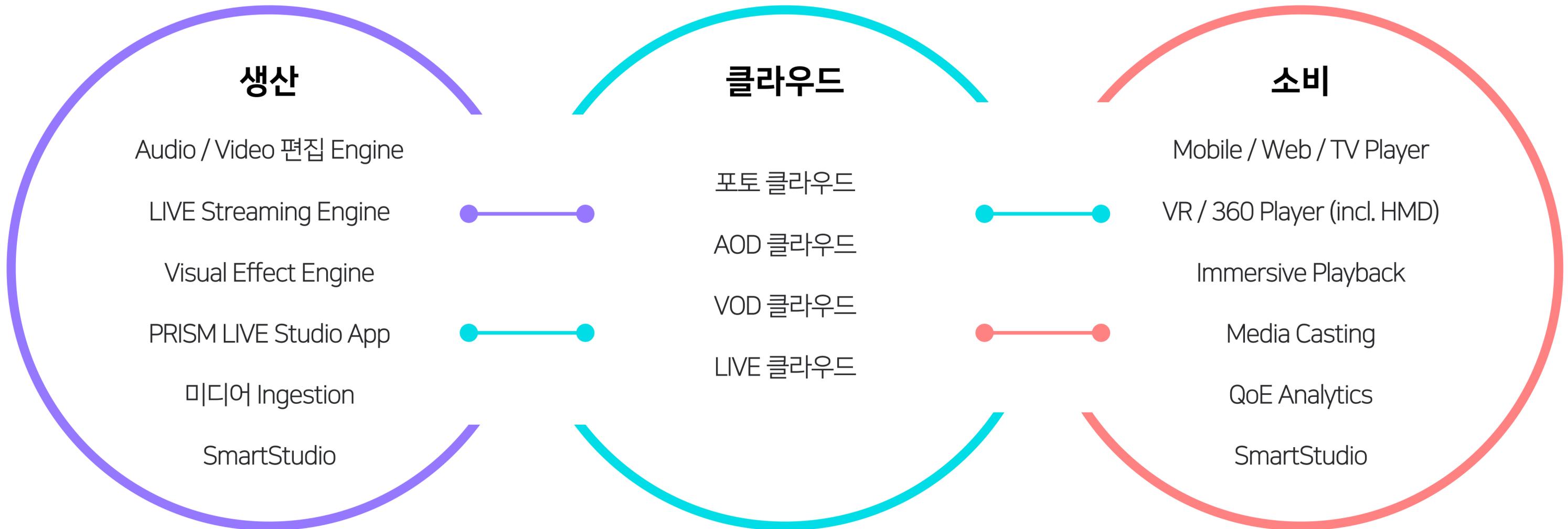


도쿄는 무관중, 네이버는 **무한관중**
라이브로 함께한 2020 올림픽

NAVER
Emerging
TECHnology

NAVER ETECH.

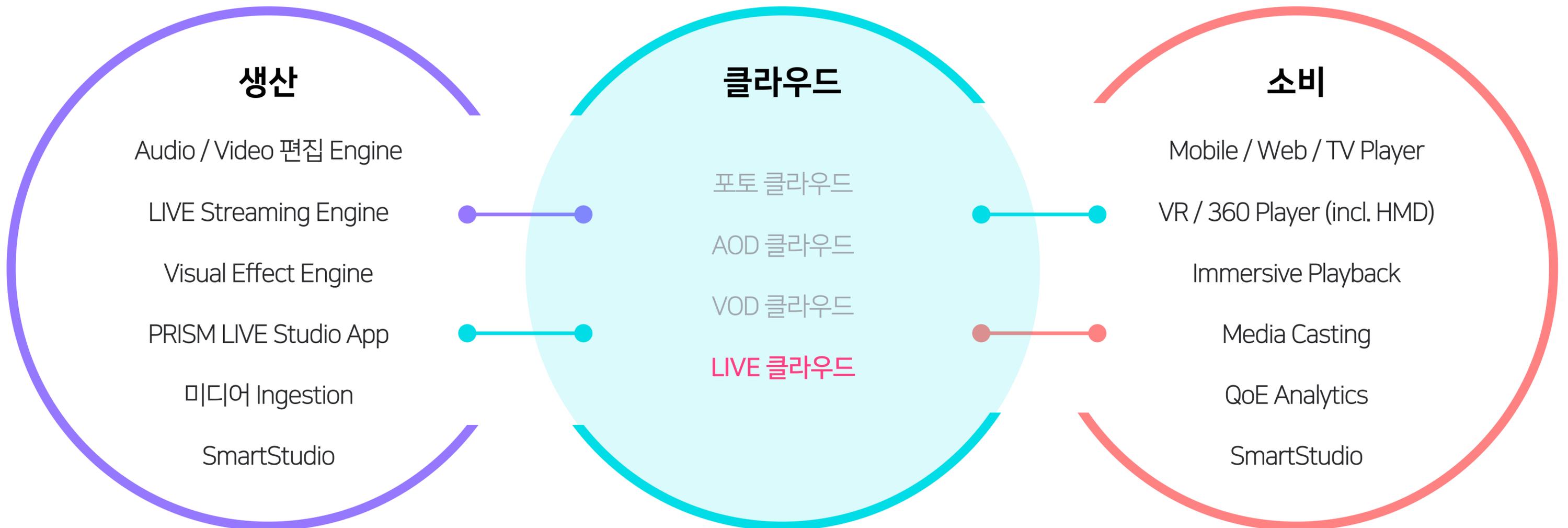
포토 / 오디오 / 비디오의 <생산 - 클라우드 - 소비> 워크플로의 전구간 기술 연구와 개발을 담당합니다. 글로벌 환경에서 시간 / 공간 / 용량의 제약 사항을 극복하고 생생한 현장 느낌과 안정적인 지원을 위해 이머징 기술 연구와 개발을 통한 원격의 시대를 준비하고 있습니다.



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.



CONTENTS

1. 2020 도쿄 올림픽 라이브 중계
2. NAVER LIVE CLOUD 구조
3. 무한관중을 위한 트래픽 제어 - 운영경험, 데이터, 튜닝, 성공적
4. 보다 나은 다음 올림픽을 위해

COVID-19

빅이벤트 (올림픽)

How many **lives**

How many **viewers**

1. 2020 도쿄 올림픽 라이브 중계

국가대표

금메달

승리 2020 도쿄 올림픽 배구

네이버 중계 한국야구

양궁

NAVER SPORTS

올림픽 중계

2020 올림픽 중계 뒤편 NAVER LIVE CLOUD 존재



NAVER SPORTS

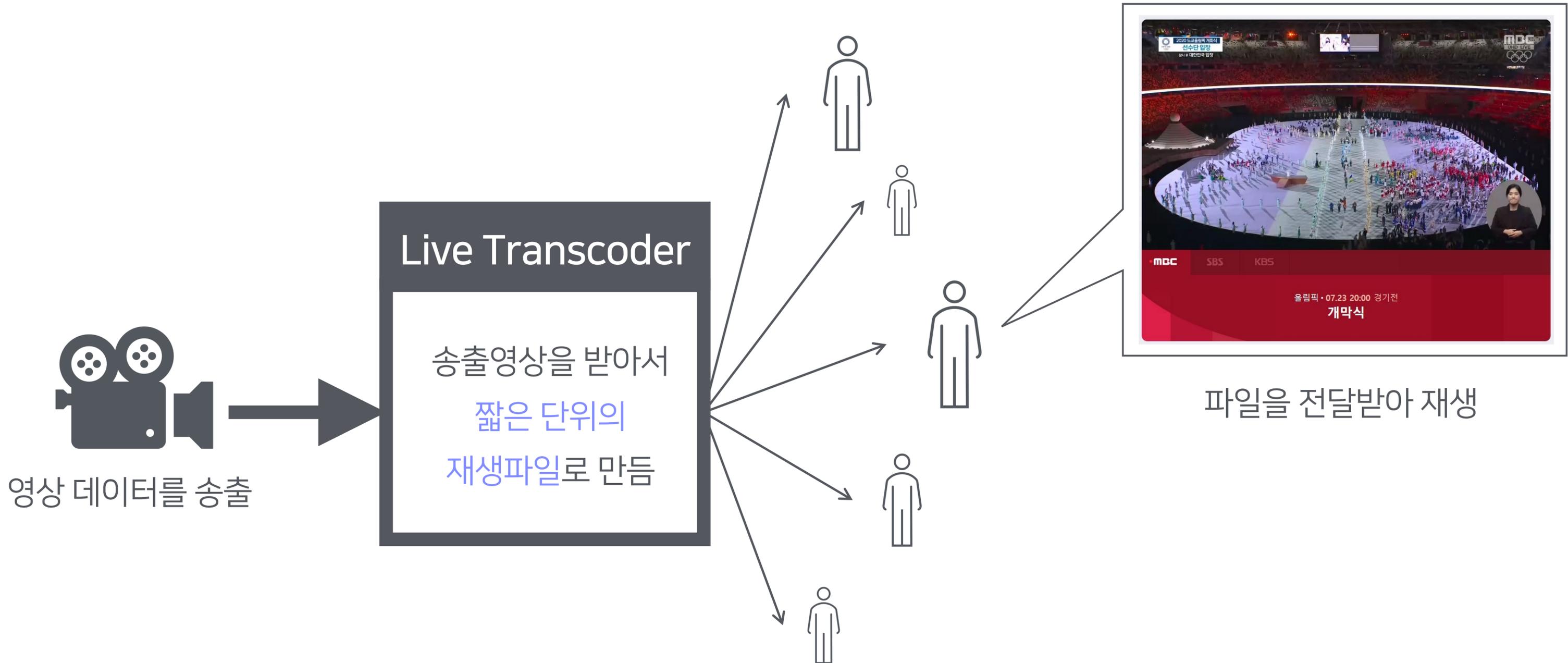
NAVER LIVE CLOUD

네이버 전사 서비스에 라이브 스트리밍 서비스를 제공



송출 >>> 중계 >>> 생산 >>> 전송 >>> 재생

라이브 스트리밍 서비스?



라이브 스트리밍 서비스의 특징

송출/생산 구간의 안정성



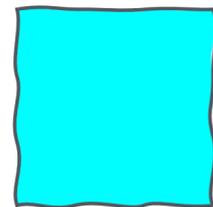
송출/생산 구간의 안정성이 무한대의 시청자의 품질을 보장할 수 있다.

라이브 스트리밍 서비스의 특징

라이브 스트리밍에 사용되는 파일

송출 영상을 잘게 나누어 영상 파일로 생성하고 전달한다.

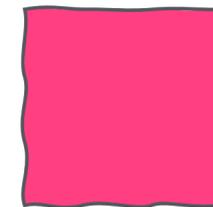
1초 단위로 나눈다고 가정한다면,



Manifest 파일 m3u8, mpd

1초마다 content가 변경된다.

update



재생파일 ts, m4v, m4a

1초마다 화질별로 각각 파일이 생성된다.

create, delete

라이브 스트리밍 서비스의 특징

라이브 스트리밍에 사용되는 파일

만약, 1시간 동안 4개의 화질로 HLS 스트리밍을 한다면?

$$60 \text{ (분)} * 60 \text{ (초)} = 3,600$$

$$3,600 * 4 \text{ (화질 개수)} = 14,400$$

 4개 파일 생성 후 각각 3600번의 수정

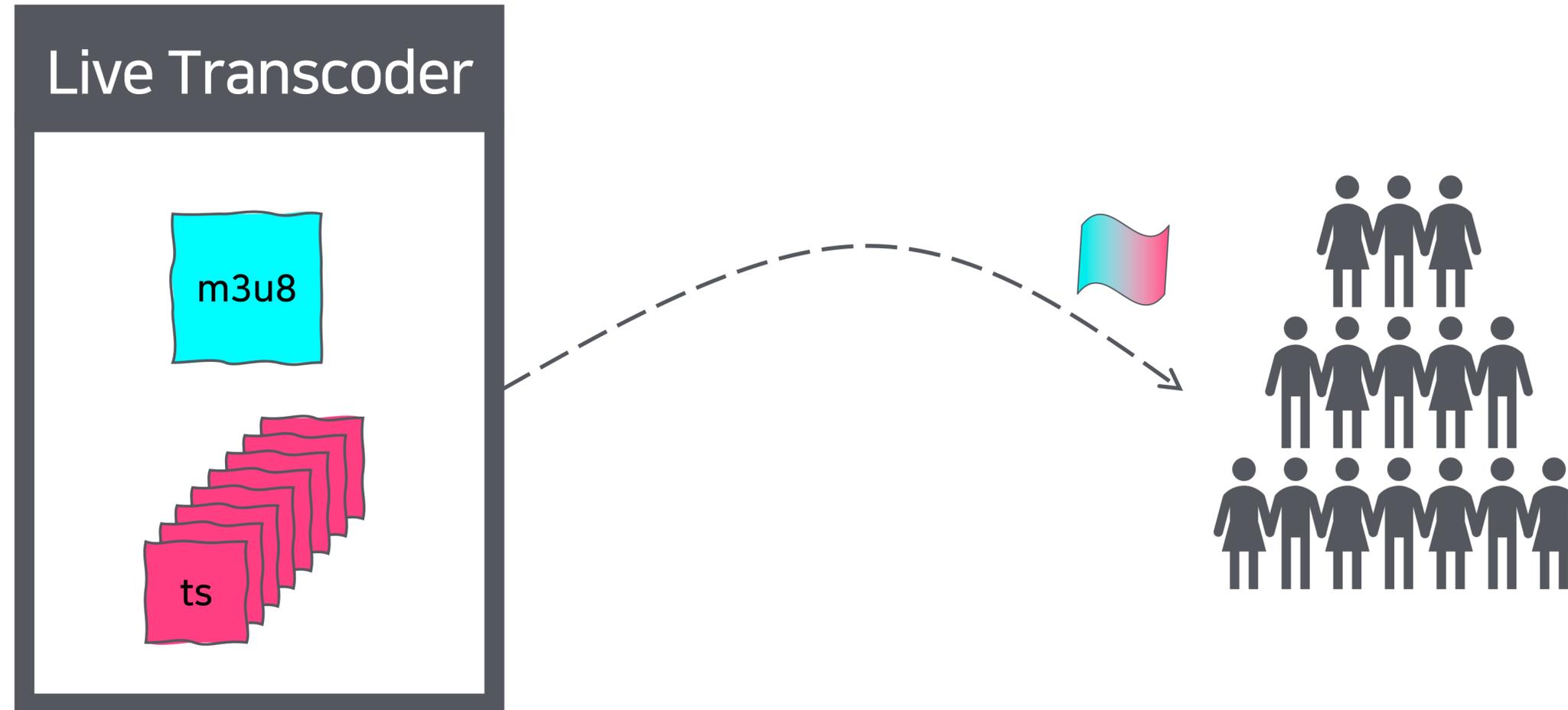
 14,400 개의 파일 생성 후 삭제

라이브 스트리밍에는 용량이 작고 개수가 많으며 휘발성이 강한 파일이 이용된다.

초 단위 파일이라 용량이 작다. 모이면 큰 용량

라이브 스트리밍 서비스의 특징

라이브 스트리밍 파일의 전송



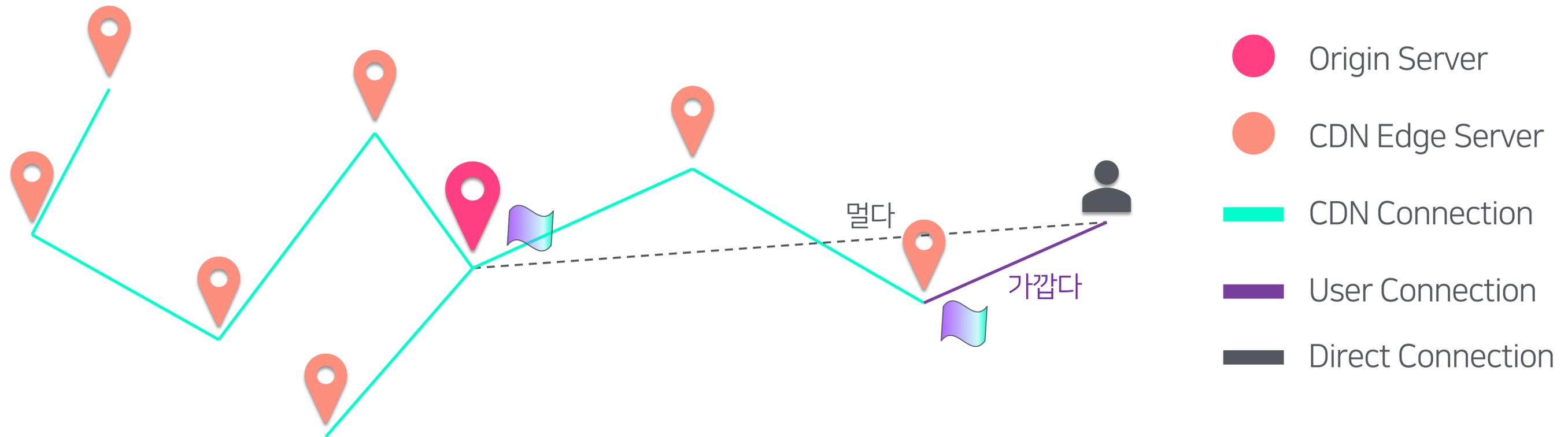
하나의 장비에서 수십만명의 사람들의 요청을 처리하긴 힘들다.
다루는 파일이 용량이 작고 갯수가 많고 휘발성이 높은 점도 힘든 요인

필수적인 CDN 사용

CDN

(Content delivery network) : 콘텐츠 전송 네트워크

전달되는 파일의 개수가 많고 라이브 특성상 빠르게 전달하기 위해 CDN의 사용이 필수적이다.



라이브 스트리밍 서비스의 특징

필수적인 CDN 사용

CDN

(Content delivery network) : 콘텐츠 전송 네트워크

전달되는 파일의 개수가 많고 라이브 특성상 빠르게 전달하기 위해 CDN의 사용이 필수적이다.

CDN의 특징

- CDN을 운영하는 회사마다 전송 품질과 비용이 제각각 차이가 있다.
- 전송할 수 있는 최대 가용량이 존재한다.
- 해당 가용량을 넘어선 트래픽이 발생할 경우 라이브 시청에 영향을 미침

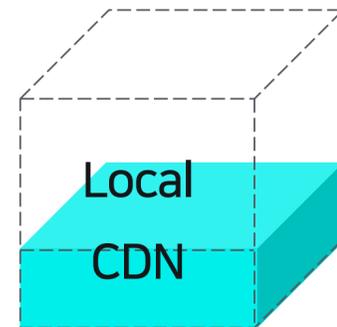
필수적인 CDN 사용

CDN

(Content delivery network) : 콘텐츠 전송 네트워크

전달되는 파일의 개수가 많고 라이브 특성상 빠르게 전달하기 위해 CDN의 사용이 필수적이다.

Local CDN



- **국내 전용** CDN
- 국내 전송 품질이 상대적으로 좋음 / 해외 전송 품질이 좋지 않음
- 가용할 수 있는 최대 트래픽의 양이 상대적으로 적음

Global CDN

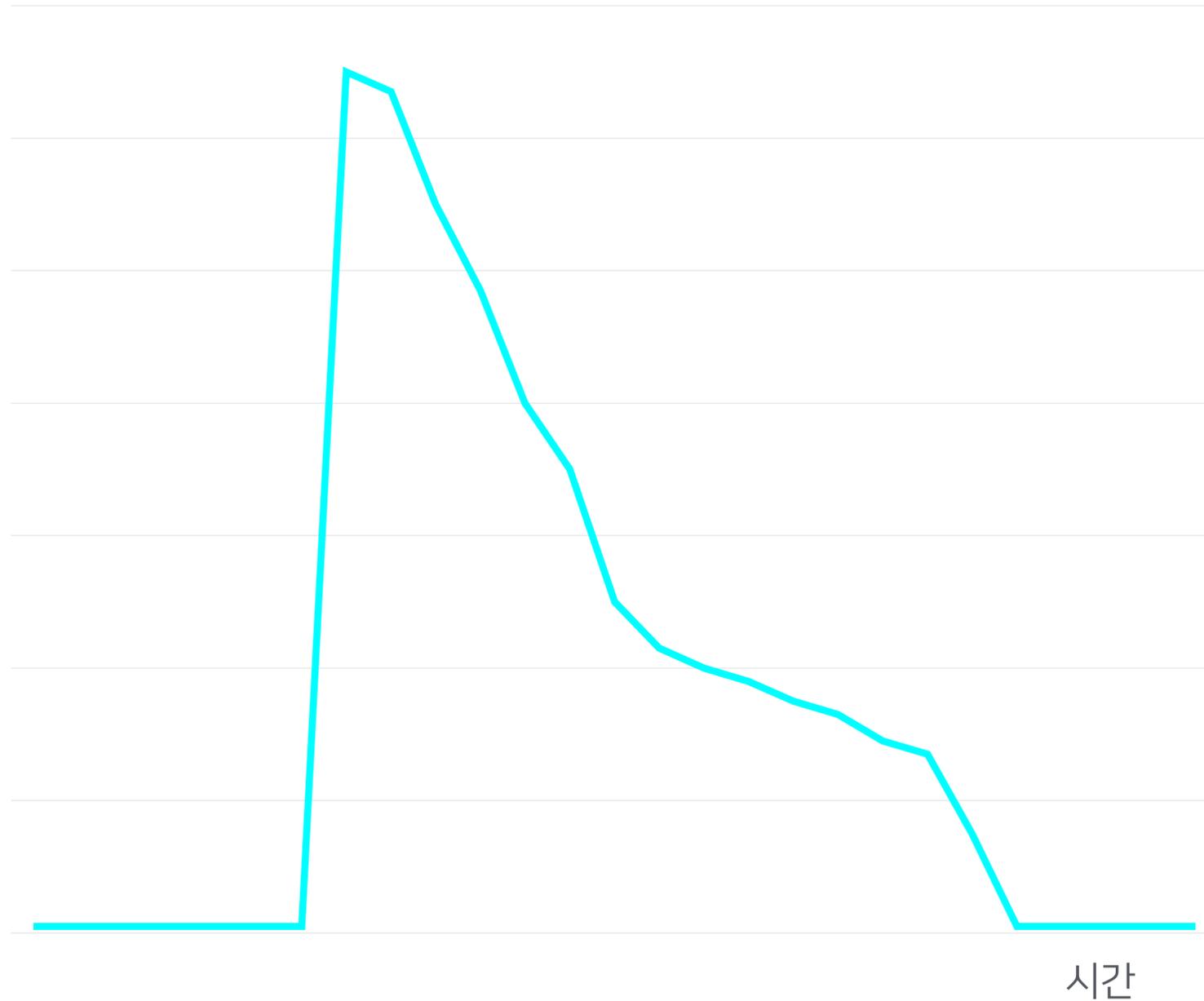


- **글로벌 전용** CDN
- 국내 전송 품질이 상대적으로 좋지 않음 / 해외 전송 품질이 좋음
- 가용할 수 있는 최대 트래픽의 양이 상대적으로 많음

라이브 스트리밍 서비스의 특징

예측할 수 없는 시청자 추이

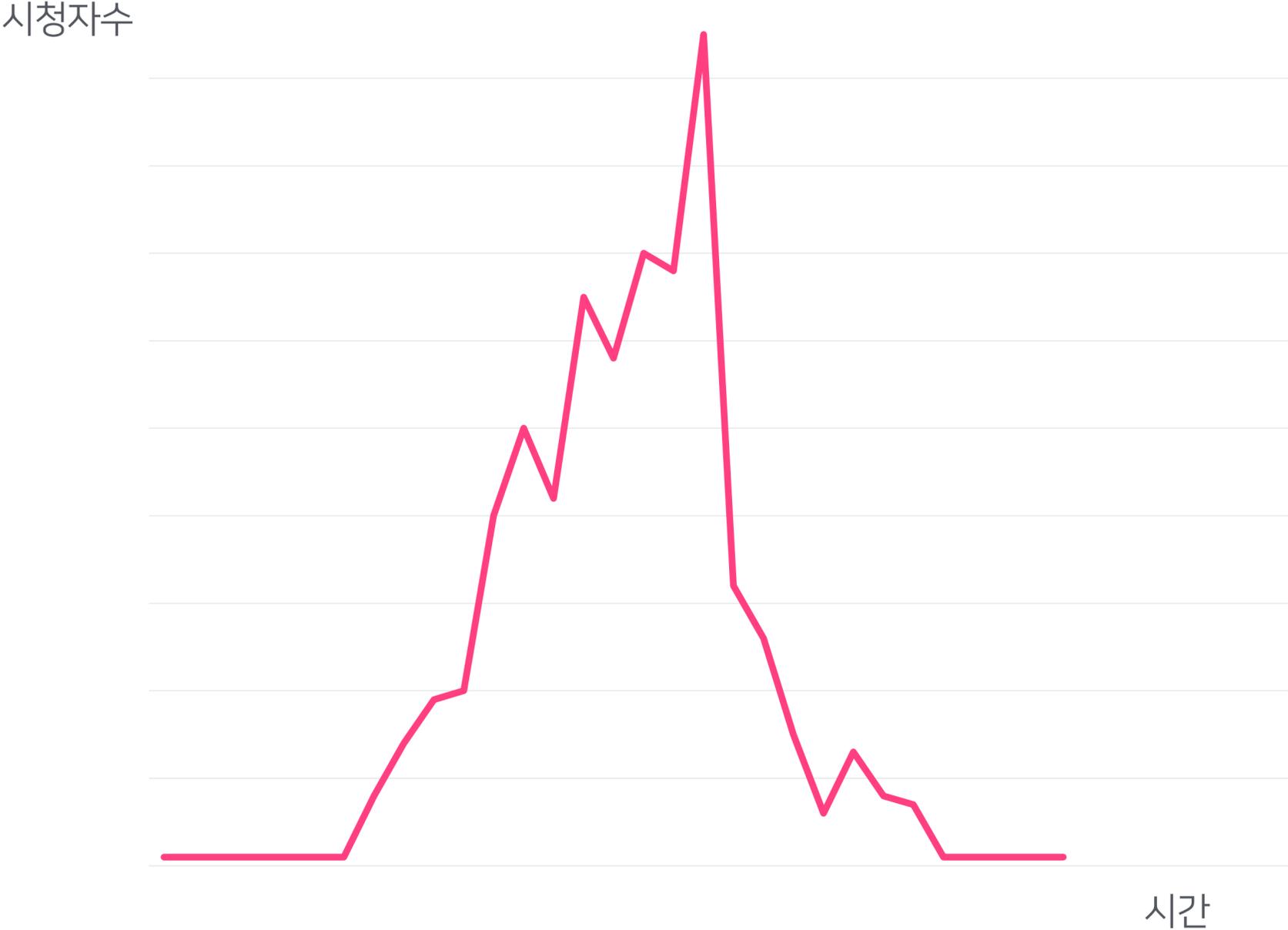
시청자수



유명 셀럽 라이브 / 쇼핑 라이브
시작하는 순간 시청자가 솟구치는 형태

라이브 스트리밍 서비스의 특징

예측할 수 없는 시청자 추이



스포츠 경기 라이브
경기 막바지로 갈수록 시청자가 쌓이는 형태
경기 세트 종료 때마다 출렁이는 특징이 있다.

라이브 스트리밍 서비스의 특징

예측할 수 없는 시청자 추이

시청자수



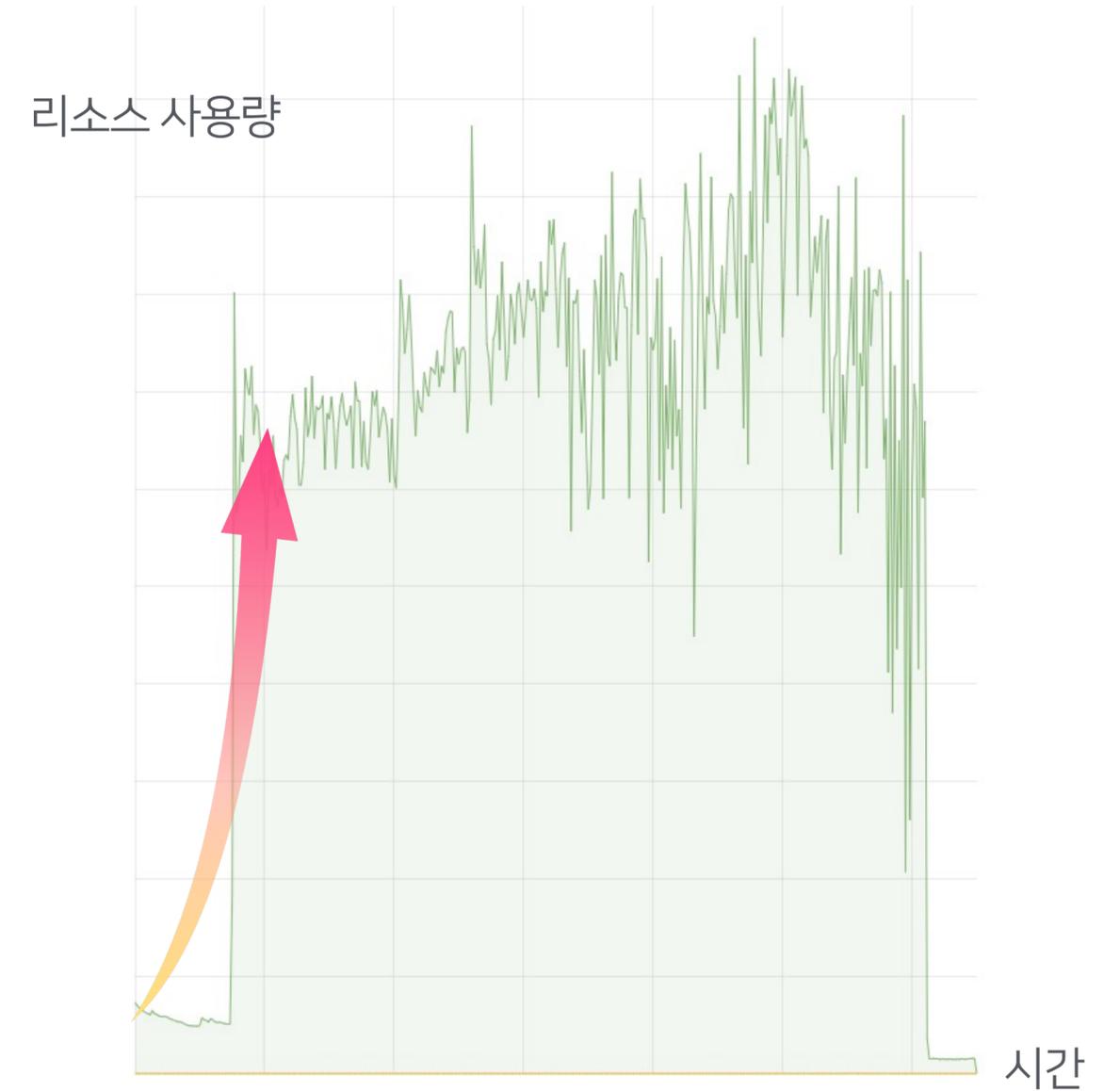
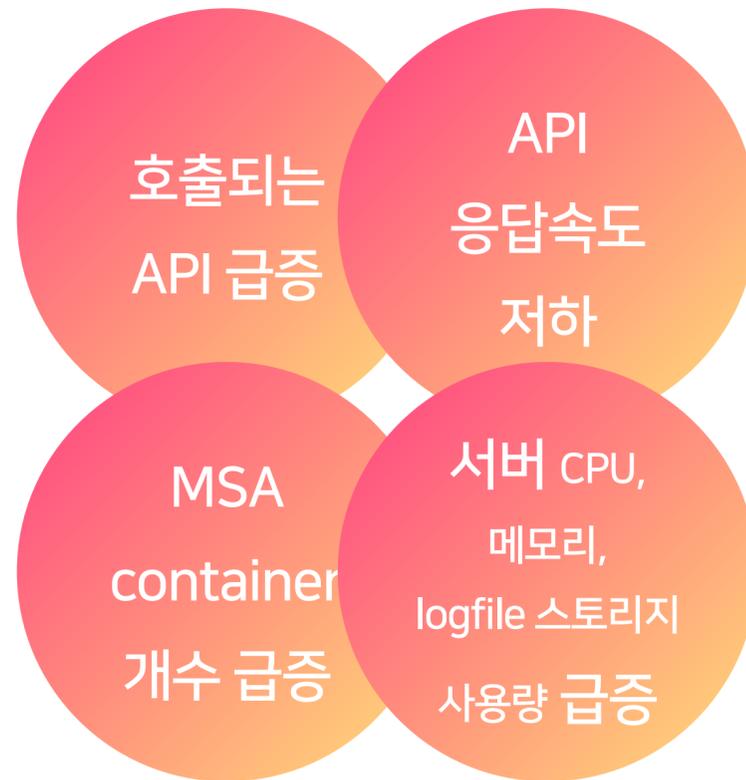
시간

뉴스 라이브

라이브 시작부터 종료까지 일정 수준의 시청자가 유지되는 형태
보통 매일 반복적으로 라이브를 하고 방송 시간이 긴 특징이 있다.

라이브 스트리밍 서비스의 특징

예측할 수 없는 시청자 추이



서버 리소스 사용량 Chart

잘 시청하고 있던 배구 경기중에 갑자기 ...



라이브에서 **장애**는 모든 시청자에게 전파

단 1초의 실수도 모두 전파

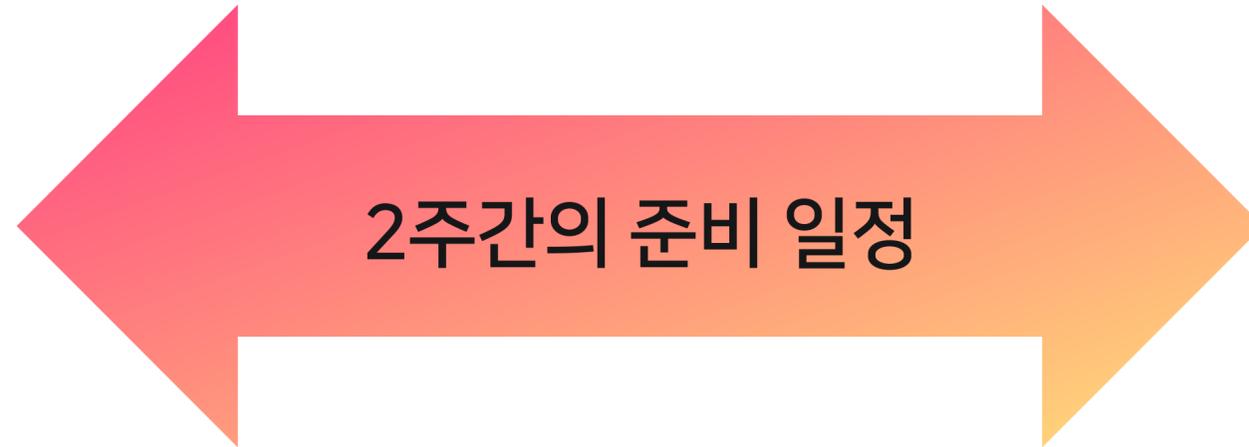
작은 실수라도 굉장히 잘 드러나는 서비스 😭

가정입니다. 이번 올림픽 배구
경기엔 일어나지 않았어요

2020 올림픽

올림픽 라이브 중계 확정

중계 확정
2021년 7월 8일



올림픽 일정
2021년 7월 23일 - 8월 8일

도쿄 올림픽 방송사 중계를 모두
온라인 라이브 중계 결정

2020 올림픽

여태 올림픽과 다르다! 준비 과정 중 고민거리

엄청난 수의 시청자 숫자가 예상

- └ 올림픽은 전세계인의 스포츠 이벤트
- └ 코로나로 인해 빅이벤트에 대한 관심도 증가
- └ 동시 시청자 수의 증가는 여러가지 고려사항이 많음

올림픽 경기 시간이 한국TIME 과 동일

- └ 워크타임에 진행되는 경기는 TV 중계가 아닌 라이브 스트리밍으로 시청
- └ 올림픽 외 평시 진행되는 다른 라이브의 시간과 겹침

국외 시청자가 거의 없고 대부분 국내 시청자일 것

- └ Local CDN의 가용량을 최대한 마련하지만 한계가 있음



성공적으로 잘 마무리한 2020 올림픽

N 도쿄 올림픽 네이버 여자배구 140만명

통합 VIEW 이미지 지식iN 인플루언서 동영상 쇼핑 뉴스 어학사전 지도 ...

● 관련도순 ● 최신순 ● 오래된순

PICK 언론사가 선정한 주요기사 혹은 심층기획 기사입니다.

동아일보 | A2면 1단 | 2021.08.05 | 네이버뉴스

"배구 안볼수가 없잖아" 140만명 동시접속 응원

4일 열린 2020 도쿄 올림픽 여자 배구 터키와의 8강전에 140만 명(네이버 집계)이 넘는 '방구석 관중'이 몰린 것으로 나타났다. 인기만큼은 '양궁 천재' 안산(20)도, 남자 수영 '샛별' 황선...



위키트리 | 2021.08.04.

"너무 떨린다..." 난리 난 여자 배구 경기, '네이버'에서 뜻밖의 일 벌어졌...

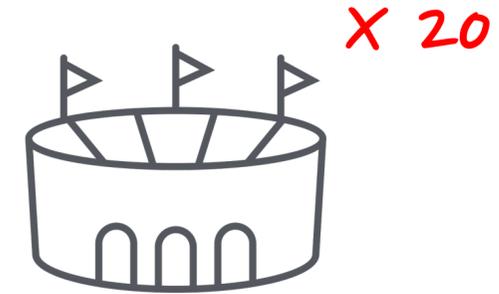
4일 한국은 일본 도쿄 아리아케 아레나에서 열린 2020 도쿄올림픽 배구 여자부 8강에서 터키를 3-2(17-25 25-17... 올림픽을 생중계하는 포털 사이트 '네이버'에는 최고 140만 명(누적...



140만명

성공적으로 잘 마무리한 2020 올림픽

동접자 140만명?!

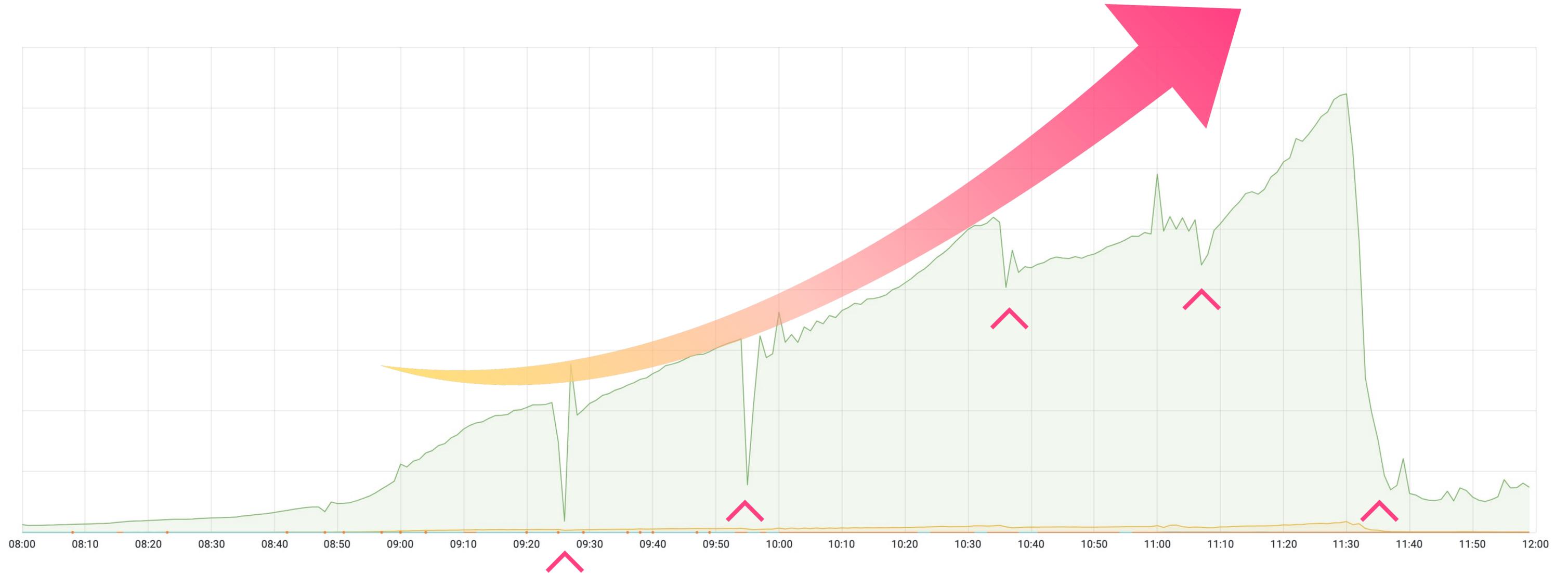


서울 월드컵 경기장 좌석수 66704석

수용인원의 20배

성공적으로 잘 마무리한 2020 올림픽

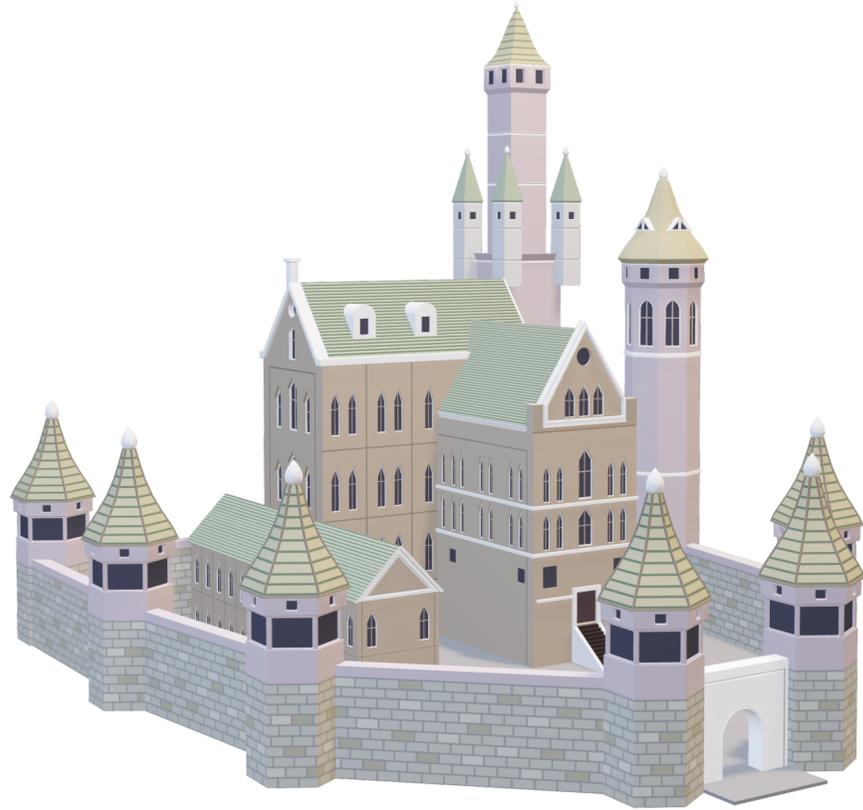
2021-08-04 올림픽 여자 배구 8강 (vs 터키) 트래픽 추이



^ : 세트 종료나 경기 종료시 시청자 급감

성공적으로 잘 마무리한 2020 올림픽

잘 마무리 할 수 있었던 Key point



튼튼하고
유연하고
결합도가 낮고
확장하기 쉬운

LIVE CLOUD 서버 구조



오랜기간 쌓인 운영 데이터와 노하우

올림픽에서 발생한 대용량 트래픽을 잘 컨트롤

2. NAVER LIVE CLOUD 구조

라이브 스트리밍 서비스가 뭐 어려운 건가?



외국 클라우드 상품 쓰면 되는거 아냐?

CDN만 잘 달면 되는 것 아닌가?

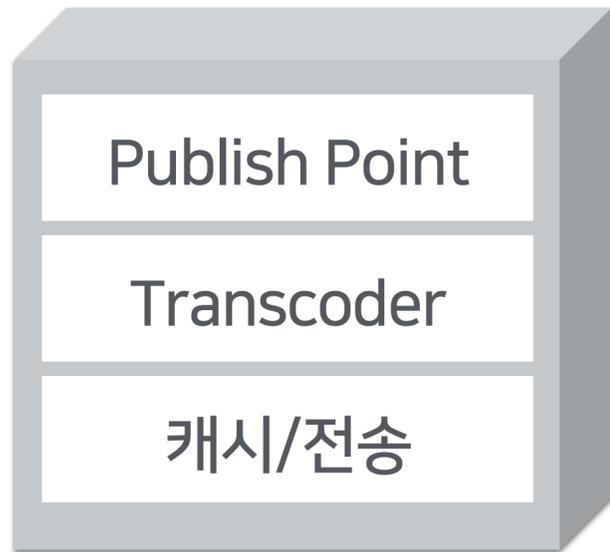
인코더랑 플레이어만 있으면 되는거 아니야?

하드웨어 성능만 좋으면 되는거 아니야?

이거 어려운 거야?

구버전 라이브 스트리밍 서비스의 구조

송출과 트랜스코딩이 고정적인 구조



Publish Point

송출 데이터를 받아주는 모듈

Transcoder

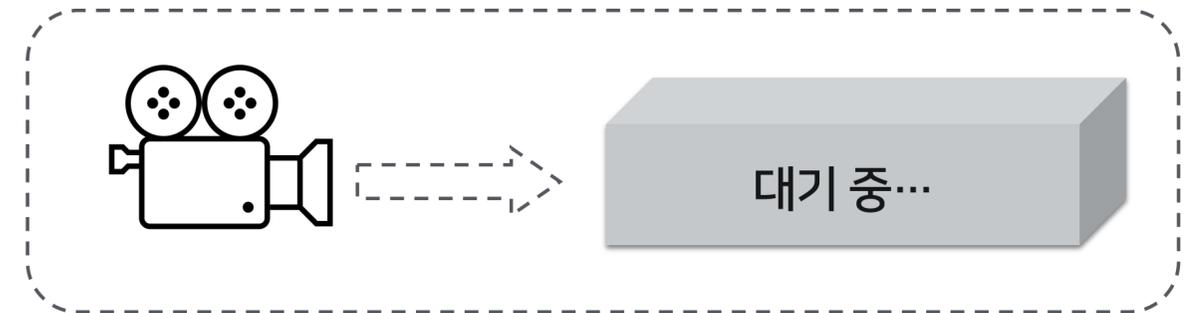
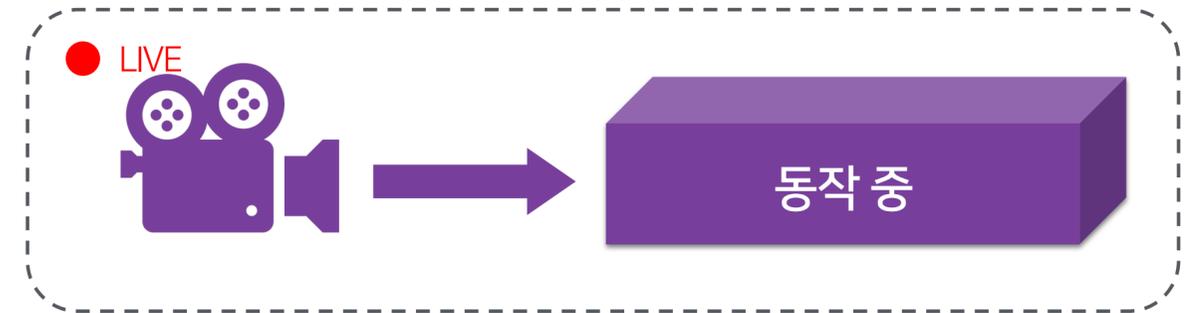
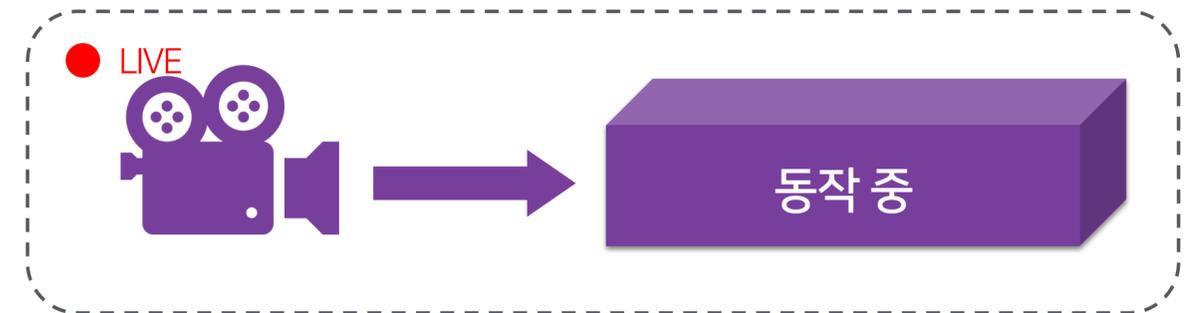
송출 영상 데이터를 각 화질별로 transcoding하는 모듈

캐시/전송

생산된 데이터의 http 전송을 담당하는 모듈

송출

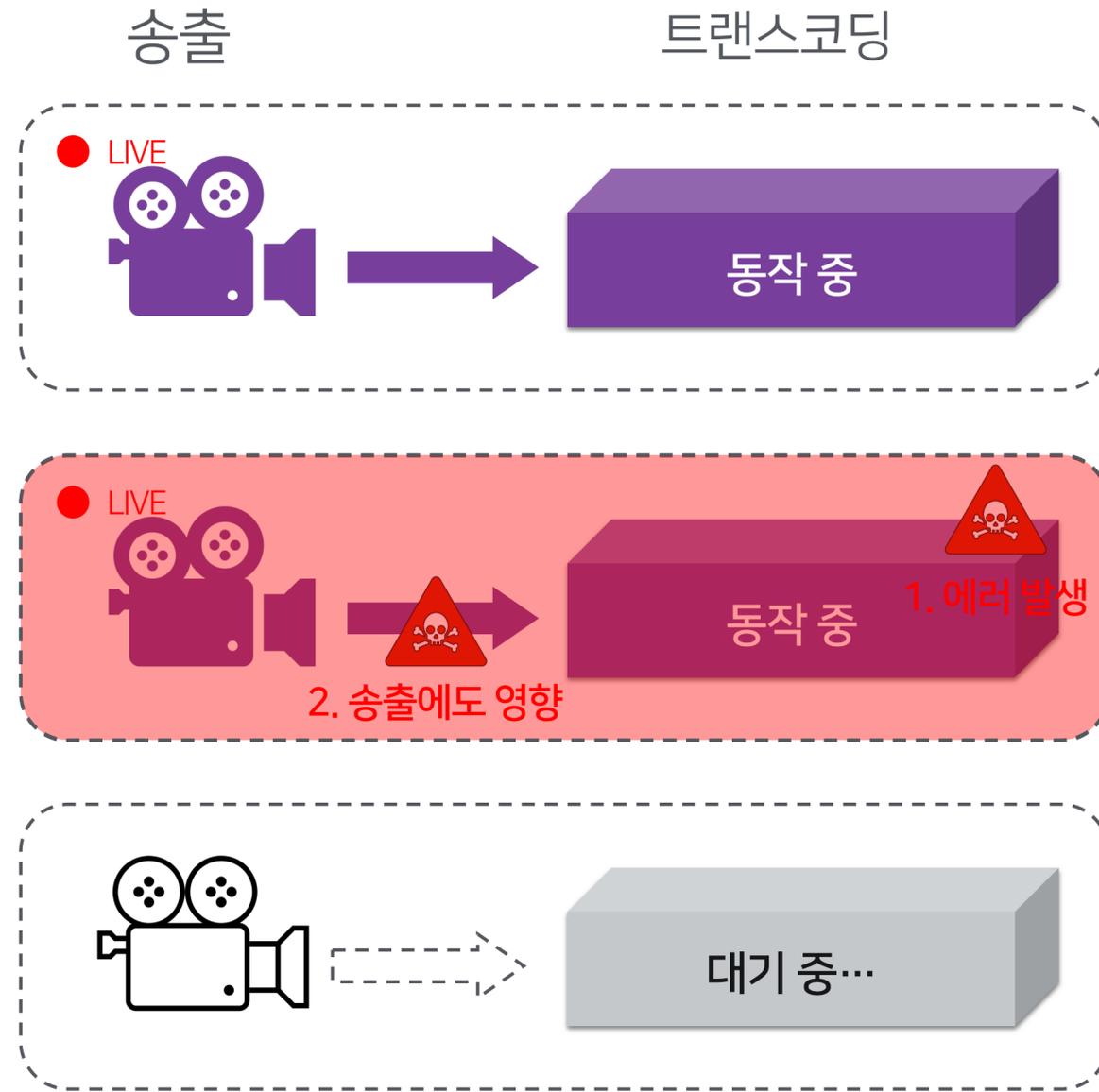
트랜스코딩



위 모듈들이 한 장비에 속해있는 구조

구버전 라이브 스트리밍 서비스의 구조

송출과 트랜스코딩이 고정적인 구조



장점

구조가 심플하여 초기 설계가 쉽다.

송출과 생산이 한 장비에서 이루어지므로 인프라 운영이 쉽다.

단점

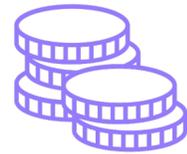
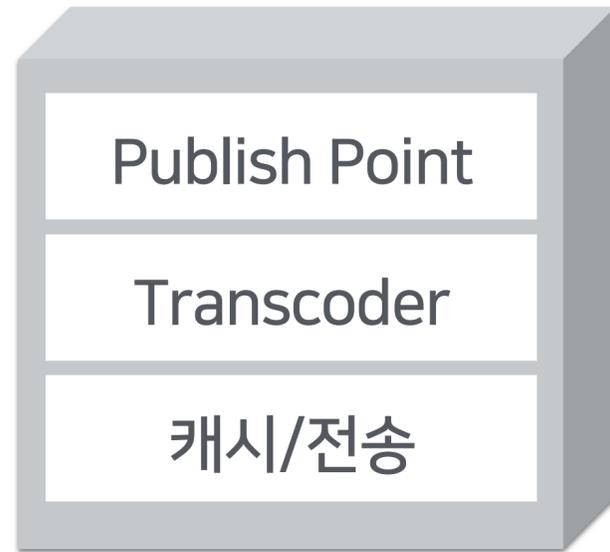
장비 관련 Single point error에 취약하다.

송출이 들어오지 않는 장비는 낭비가 된다.

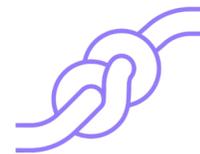
동시에 송출할 수 있는 방송이 고정적이다.

구버전 라이브 스트리밍 서비스의 구조

개선이 필요한 시점



송출이 없는 동안은 Transcoder 자원이 점거되어 낭비



추가되는 새로운 코덱과 프로토콜을 수용하기엔 불편한 구조



송출 주소가 고정되므로 Single point error 요소

LIVE CLOUD Transcoder module Farm 구성

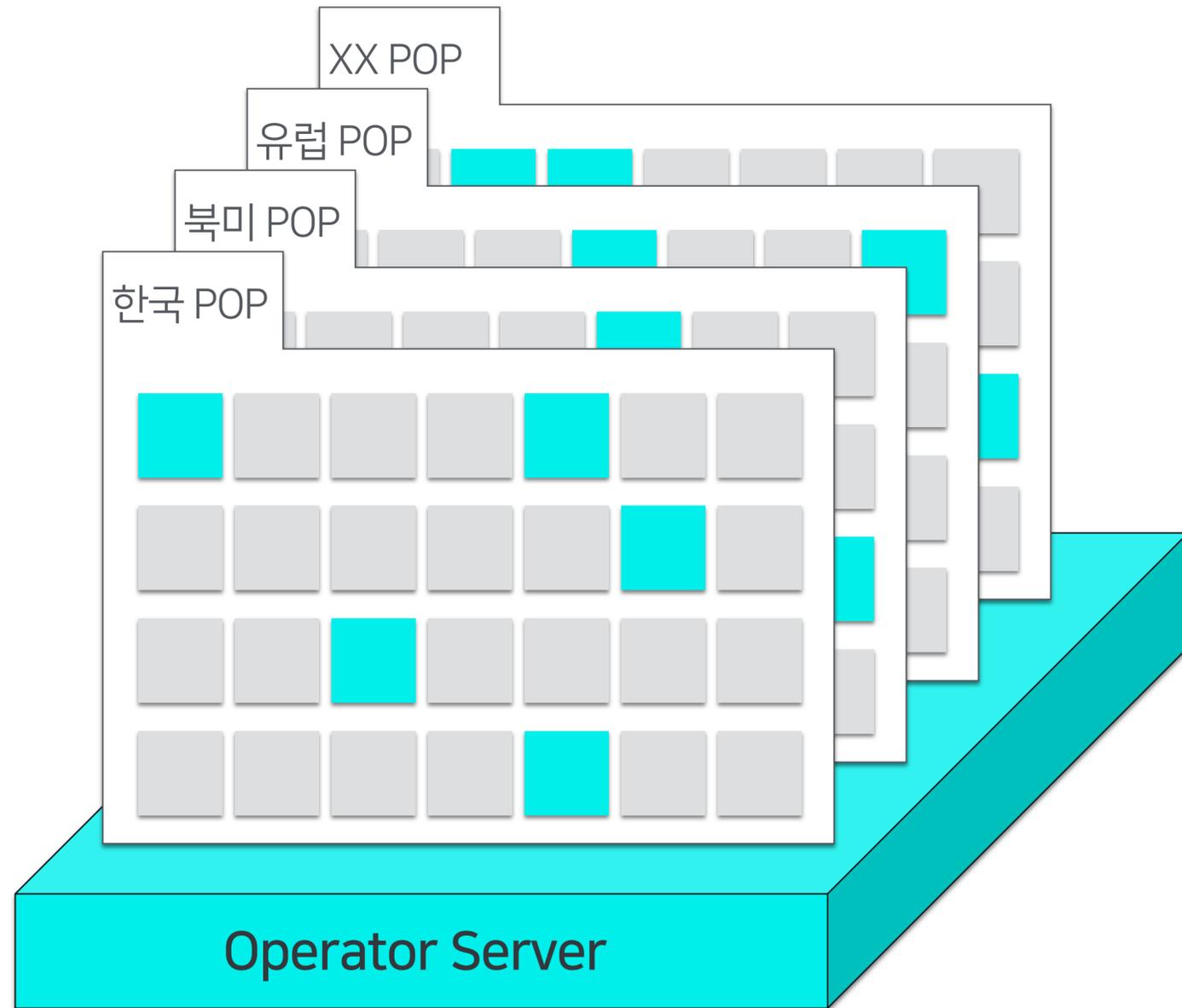
각 기능별 모듈 분리 후 각 모듈별 clustering



모듈별 자원을 별도로 관리 가능
Global POP 에 전진 배치에 용이
Single point error에 대응 가능

LIVE CLOUD Transcoder module Farm 구성

clustering farm 는 각각 region pop에 배치



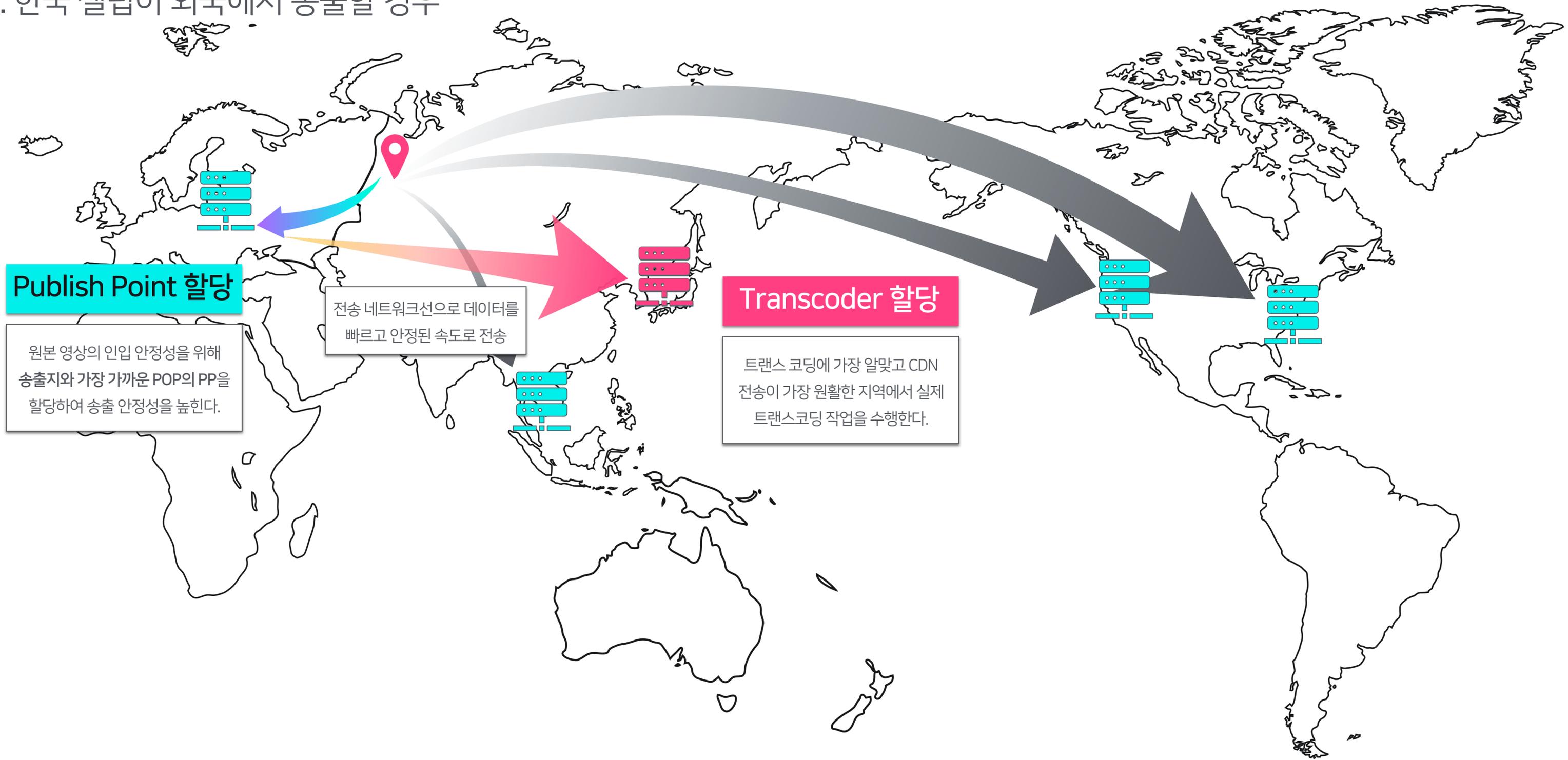
Operating Server

Global POP의 수 천대의 모듈을 효율적으로 관리

- 송출 인입지에서 네트워크 전송이 가장 용이한 global farm을 계산하여 Publish Point 할당
- 트랜스 코딩에 필요한 인코딩 자원을 미리 예측하여 가장 효율적인 Transcoder 할당

LIVE CLOUD Transcoder module Farm 구성

ex. 한국 셀럽이 외국에서 송출할 경우



LIVE CLOUD Operator server의 MSA 구조



docker
kubernetes



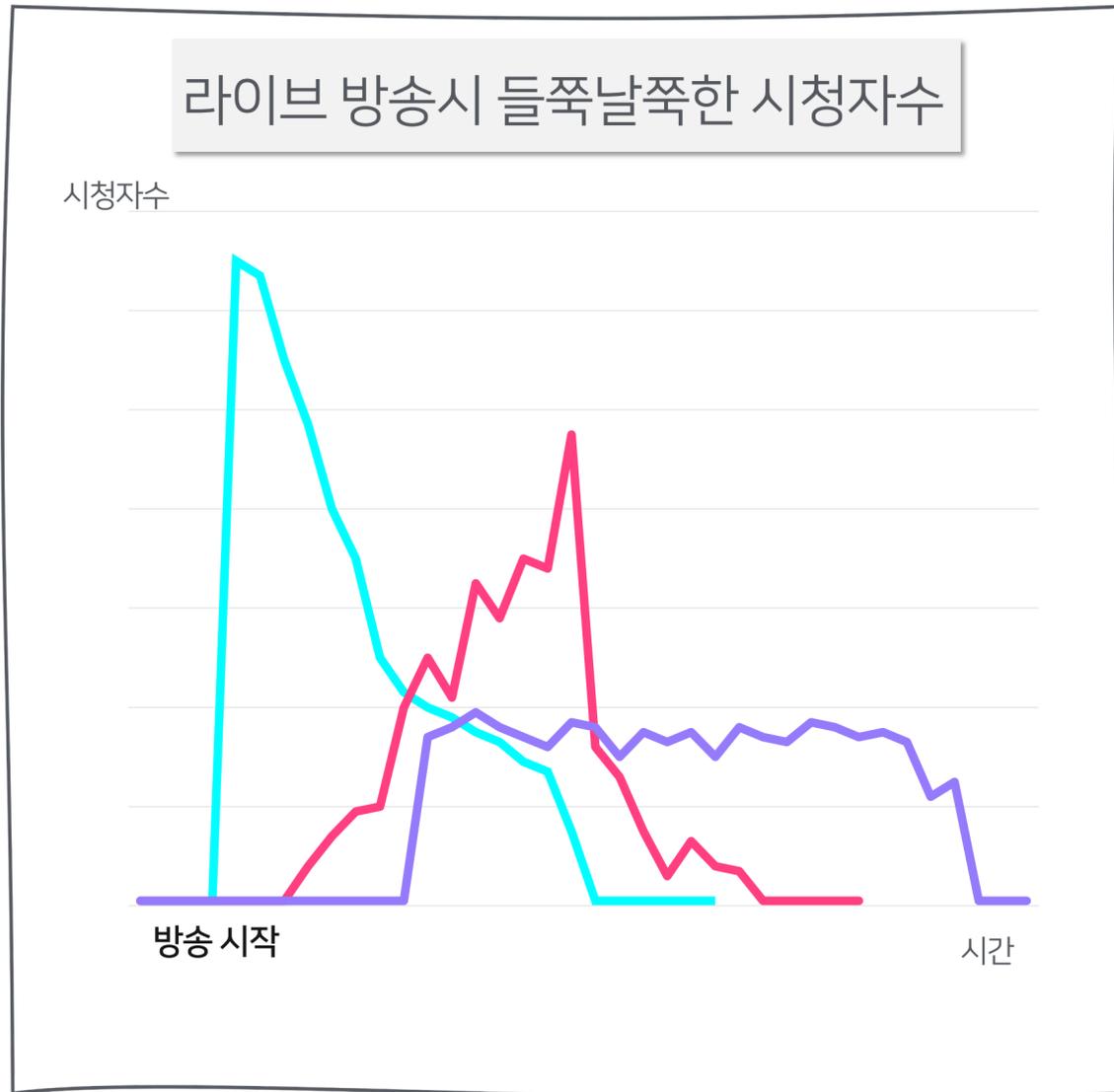
spring boot cloud



ELK stack, pinpoint, rabbitmq ...

LIVE CLOUD Operator server의 MSA 구조

docker & kubernetes



시청자의 급격한 증가



재생을 위한 API 호출 급증



서버 인프라 과부하 🔥



Kubernetes - Horizontal Pod Autoscaler
시청자수에 맞게 Server 숫자를 빠르게 증감

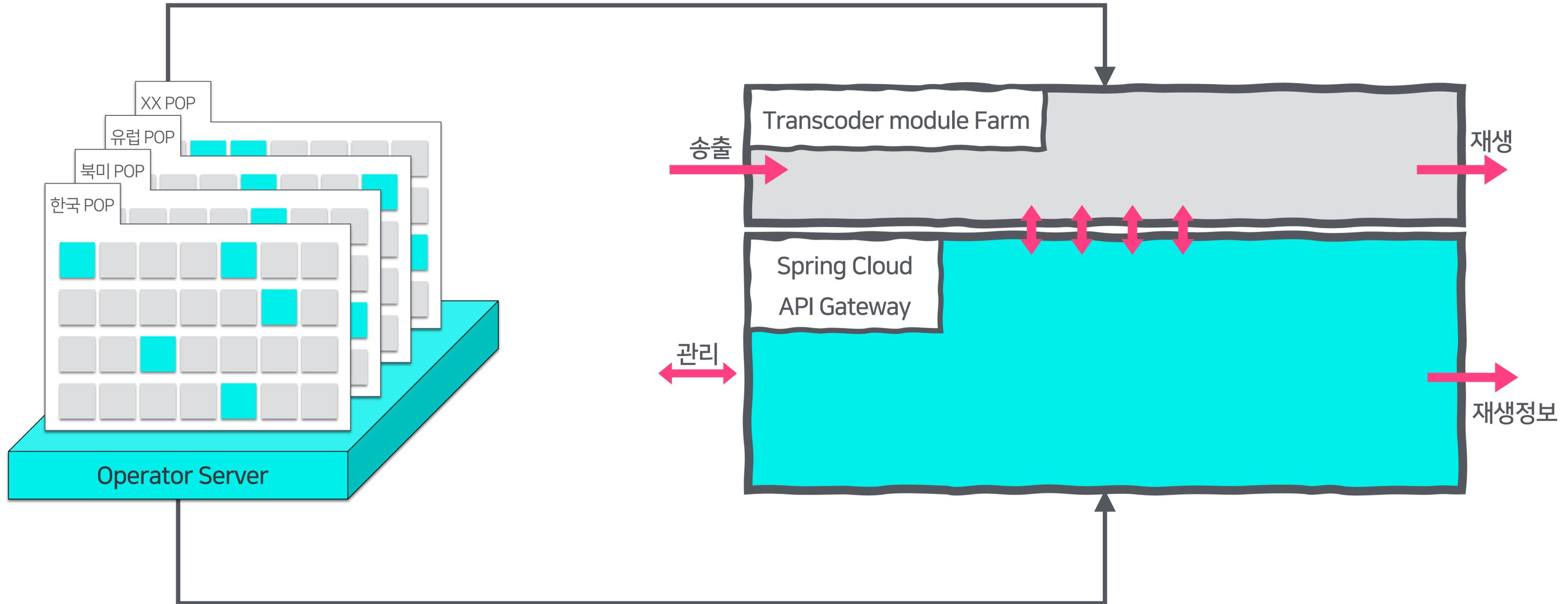
시청자의 급격한 감소



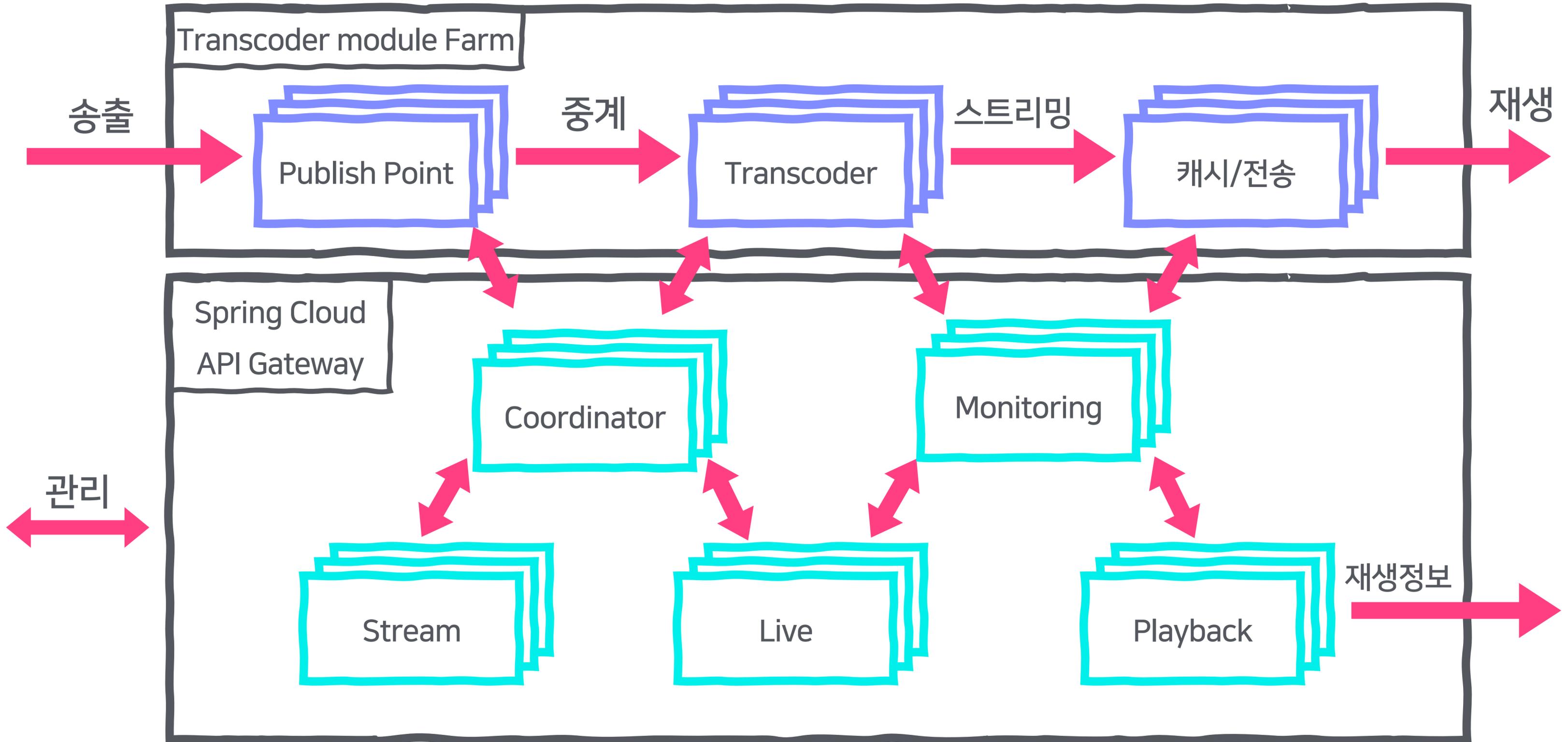
서버 인프라 낭비 😓



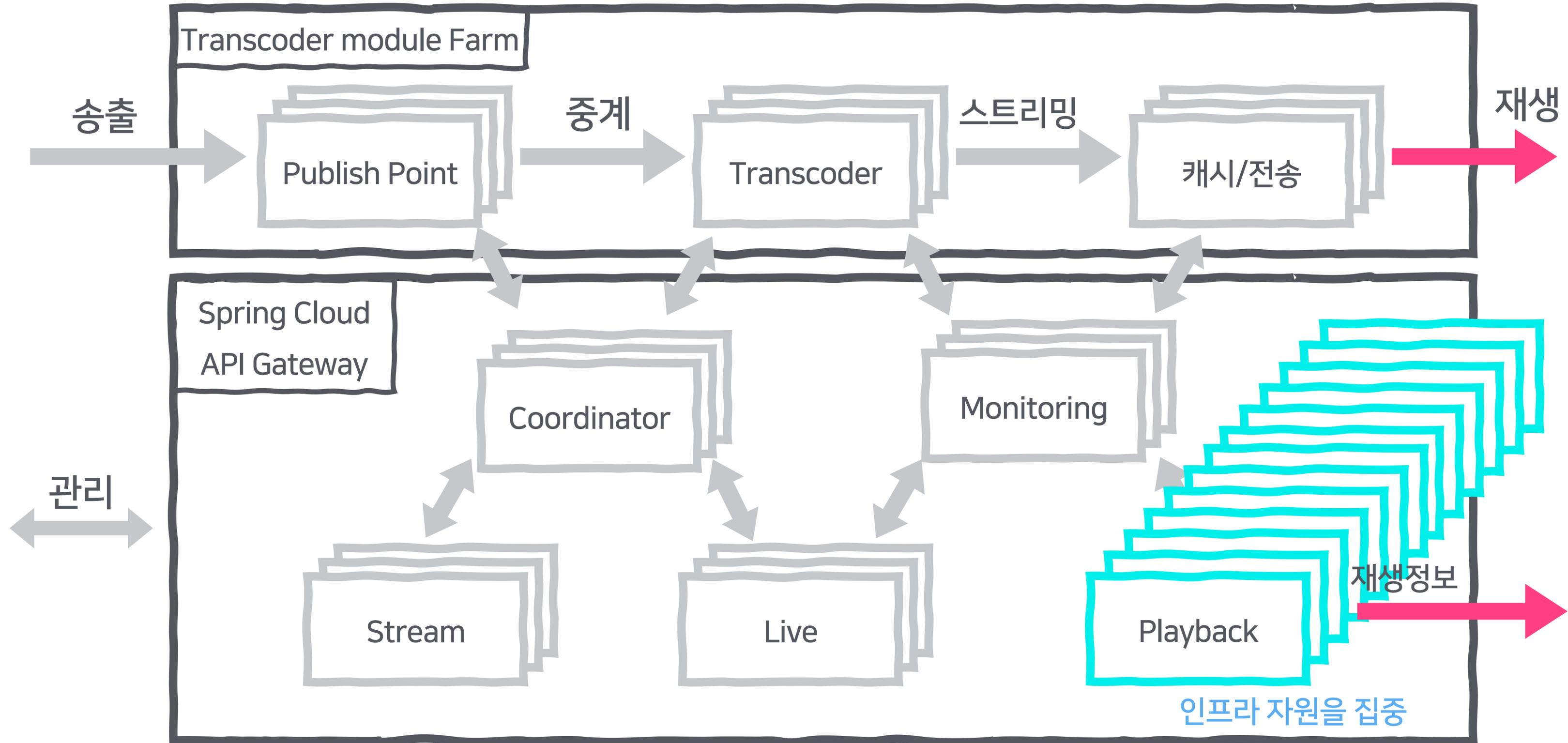
LIVE CLOUD Operator server의 MSA 구조



LIVE CLOUD Operator server의 MSA 구조



140만 동접시 API 호출량이 급증한 playback 서버



LIVE CLOUD Operator server의 MSA 구조

Spring cloud - eureka, zuul, hystrix, ribbon

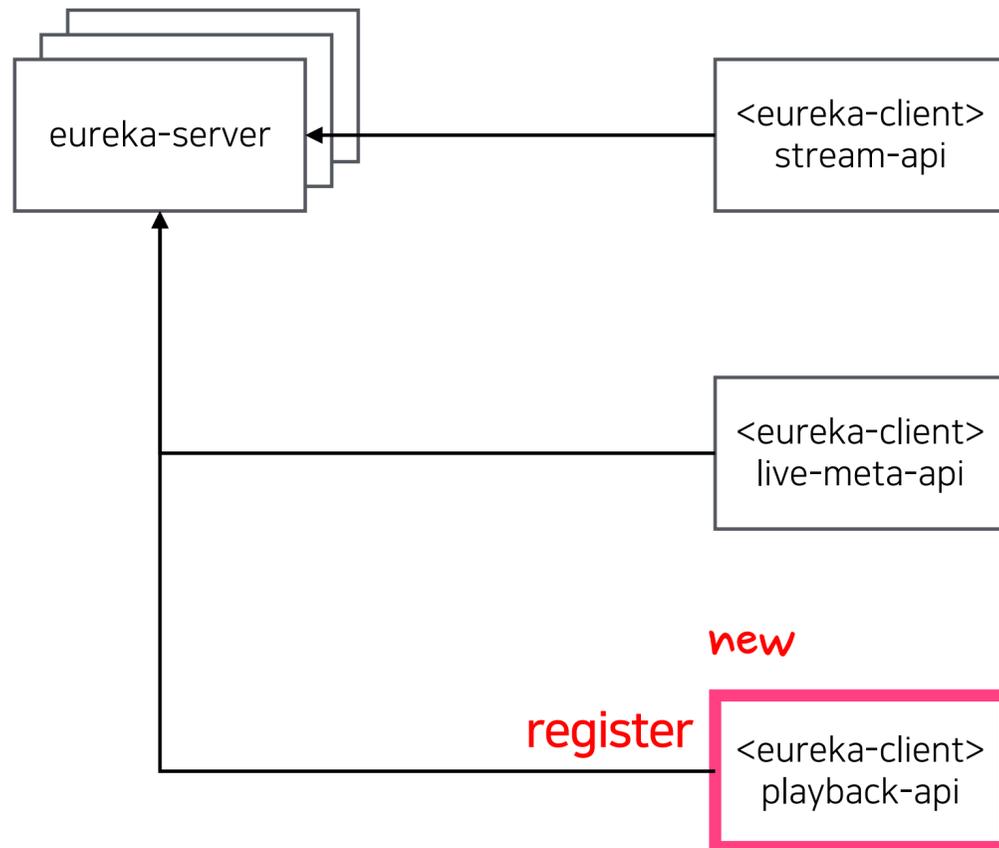
Netflix에서 공개한 Spring cloud 오픈소스

모듈	기능	사용 여부	중요도
eureka	Service Registry & Discovery	0	★★★★★★
zuul	Router & Load balancer	0	★★★
hystrix	Circuit breaker	0	★★★★★★★★★★
ribbon	Client-side load balancer	0	★★

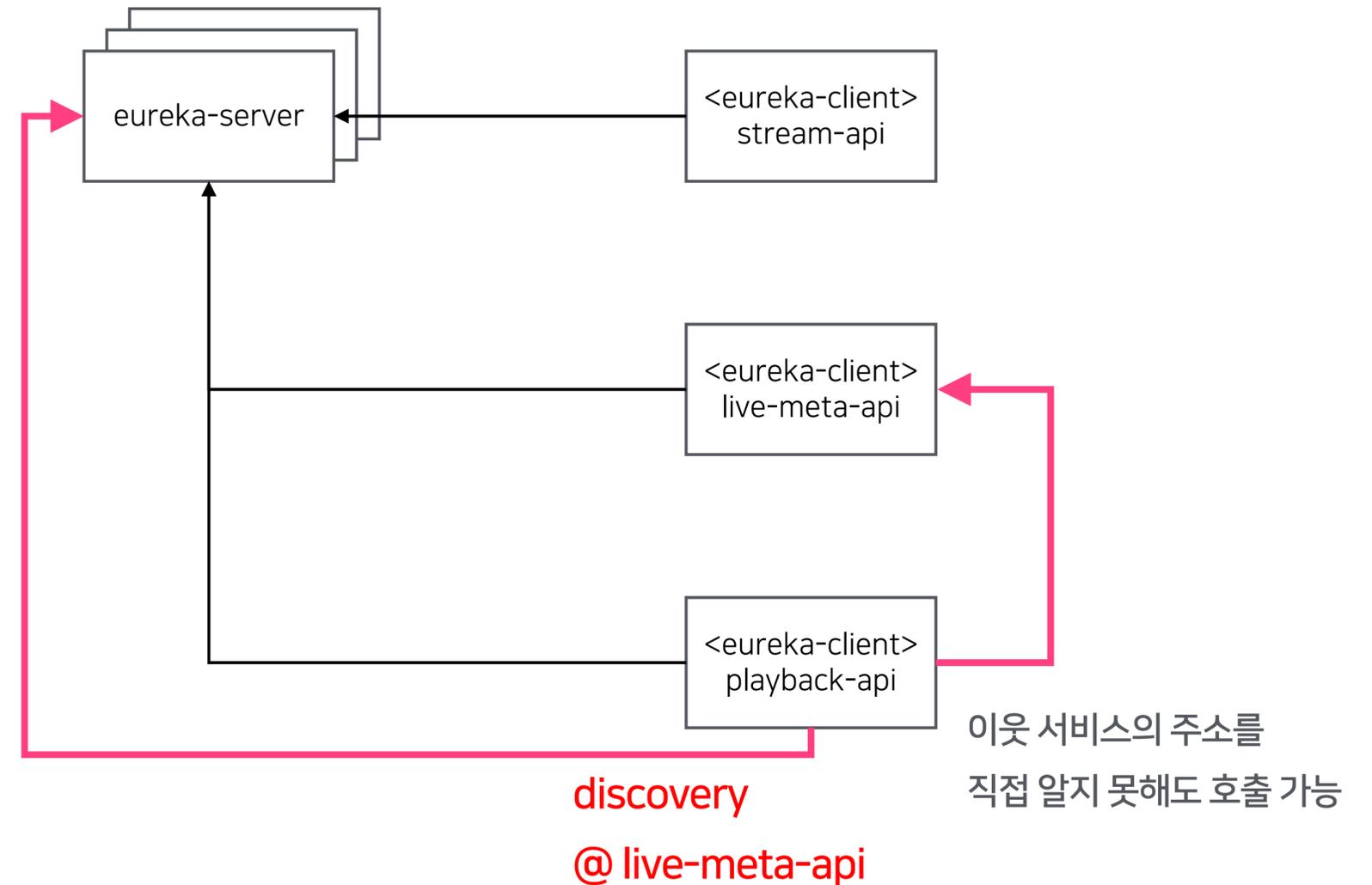
LIVE CLOUD Operator server의 MSA 구조

eureka - Service Registry & Discovery

Service registry



Service discovery



LIVE CLOUD Operator server의 MSA 구조

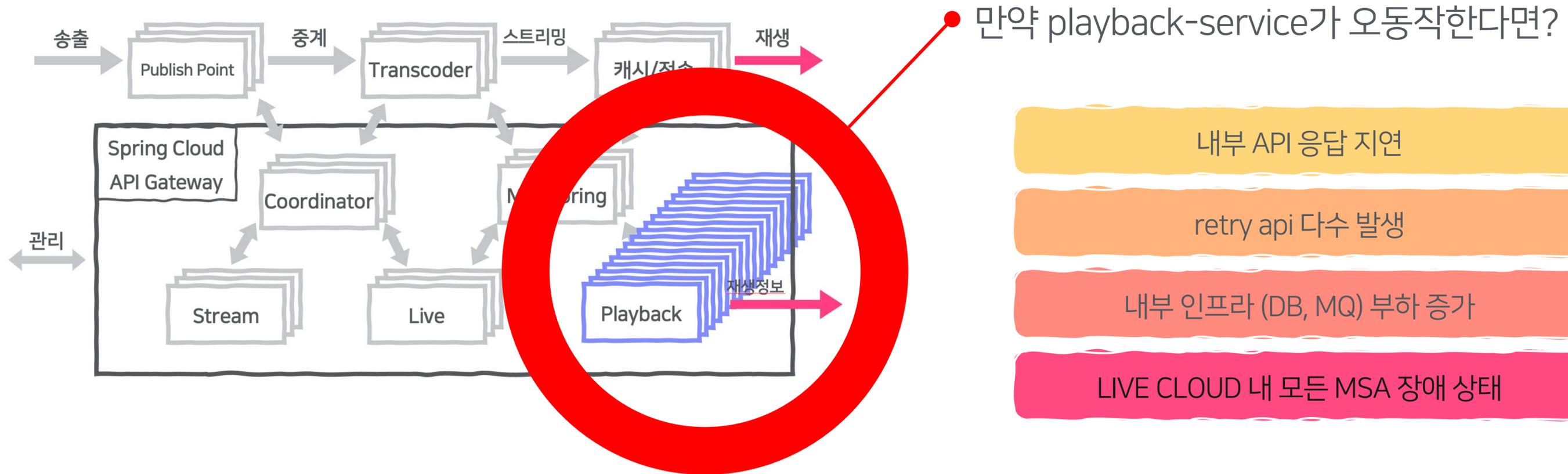
eureka - Service Registry & Discovery



eureka-client 설정만 하면 자동 register
eureka-client에 등록된 서버간 내부 API 호출이 매우 쉽게 가능

LIVE CLOUD Operator server의 MSA 구조

hystrix - Circuit breaker

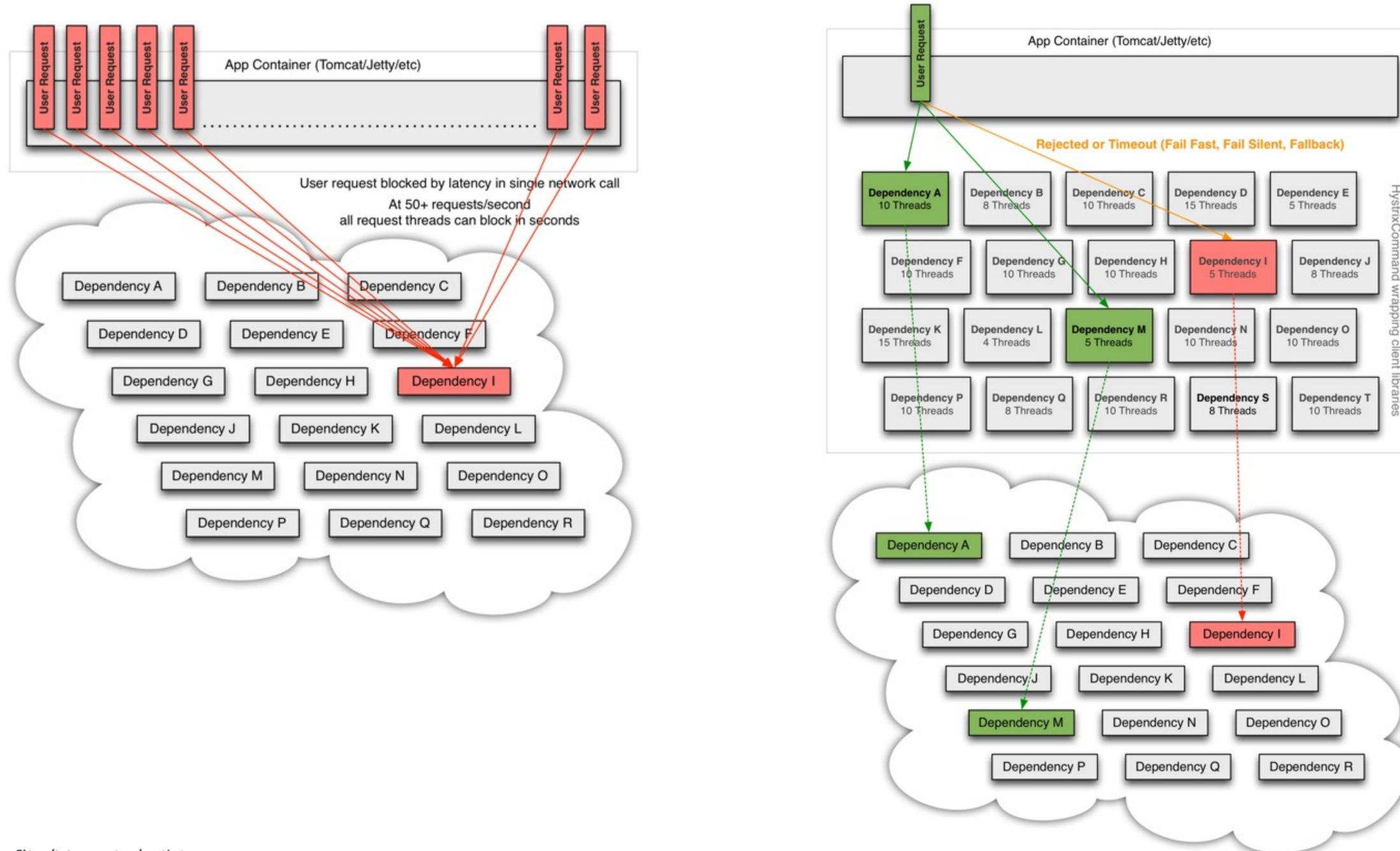


재생 정보를 전달하기 위해 client에 노출되어 있으며, 시청자수가 급등하면 qps가 급증하는 playback-service는...



LIVE CLOUD Operator server의 MSA 구조

hystrix - Circuit breaker

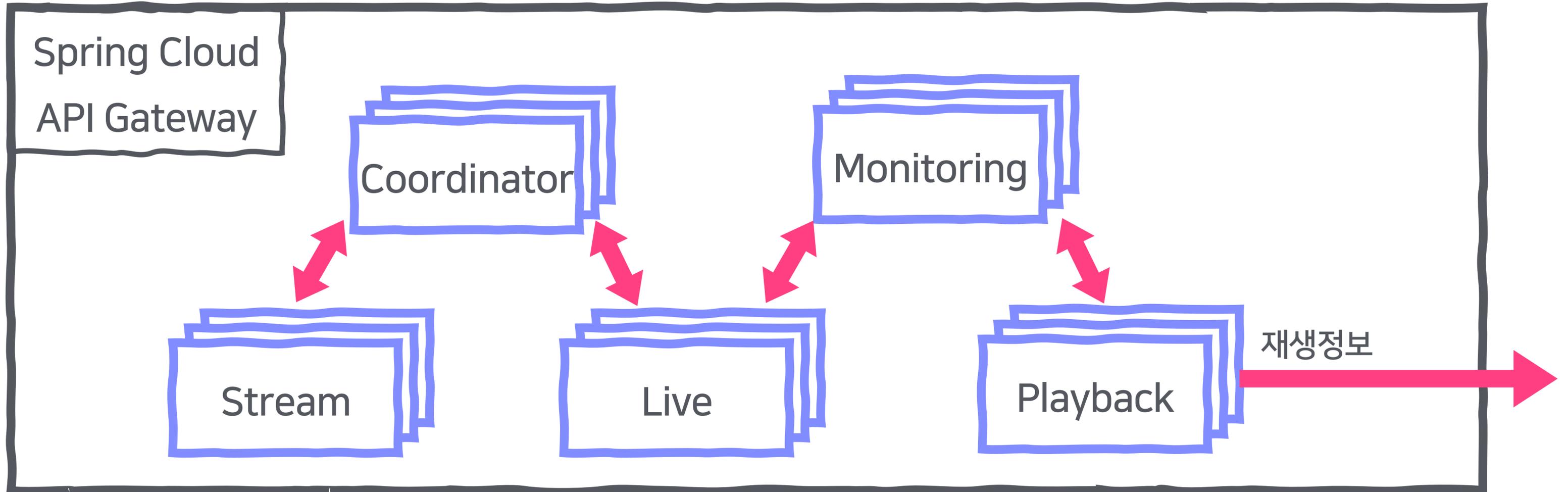


LIVE CLOUD Operator server의 MSA 구조

hystrix - Circuit breaker

@eureka

eureka-client에 속한 service로 API 호출 실패 시, service 단위로 격벽 처리하고 미리 정의해 둔 fallback 응답으로 리턴

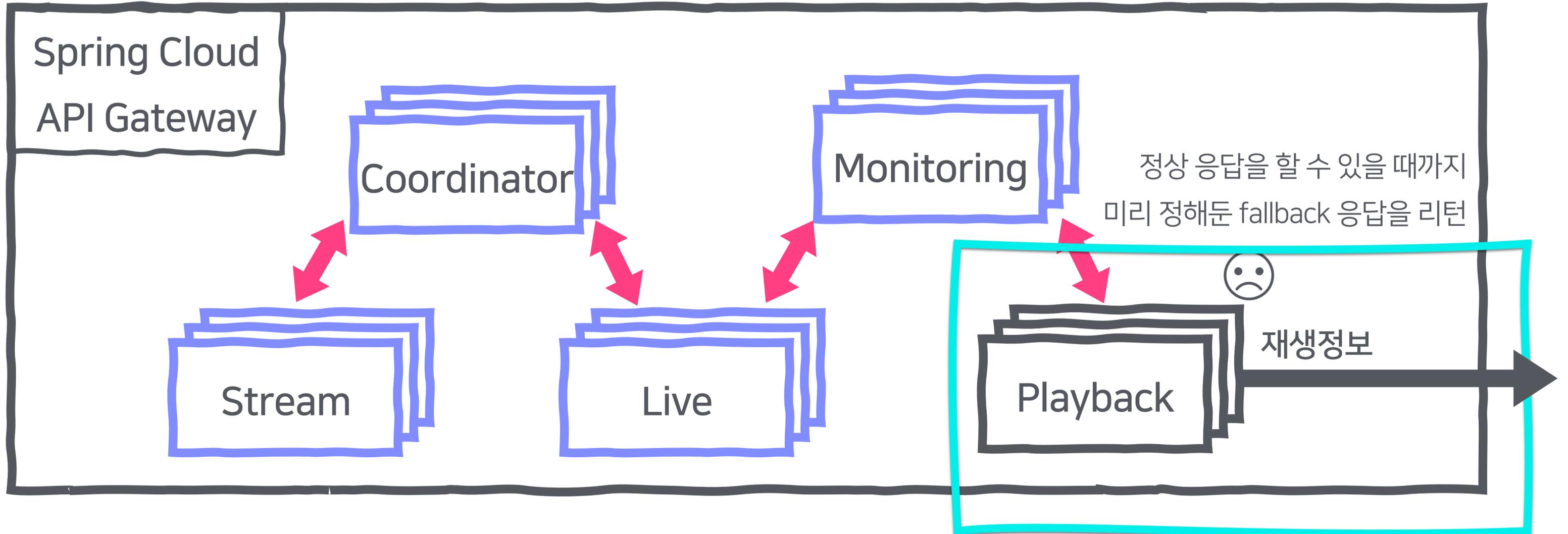


LIVE CLOUD Operator server 의 MSA 구조

hystrix - Circuit breaker

if 서비스 불능 판단

circuit breaker open으로 로직상 발생하는 내부 service api 호출을 차단하고 빠르게 fallback 응답 리턴



LIVE CLOUD Operator server 의 MSA 구조

hystrix - Circuit breaker

Service 불능 상태일시, 장애 상황을 격벽처리하여 장애 전파를 손쉽게 막는다.



튼튼하고 LIVE CLOUD 서버 구조

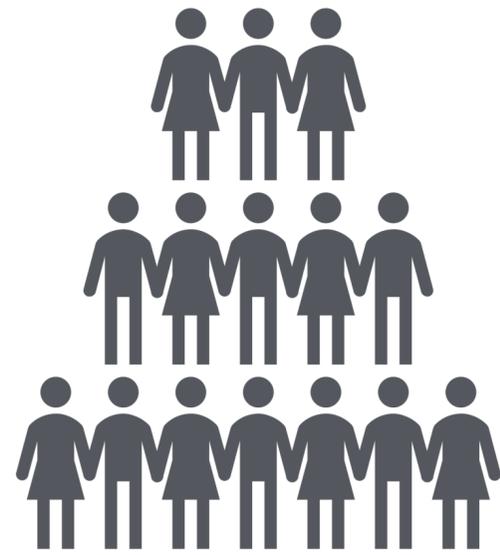
유연하고

결합도가 낮고

확장하기 쉬운



3. 무한관중을 위한 트래픽 제어 운영경험, 데이터, 튜닝, 성공적



시청자 → 트래픽

CDN
트래픽

영상 재생 파일의 다운로드에 사용되는 트래픽



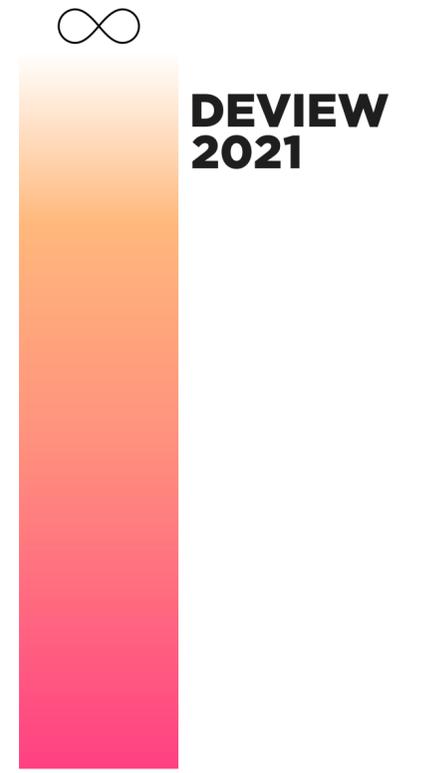
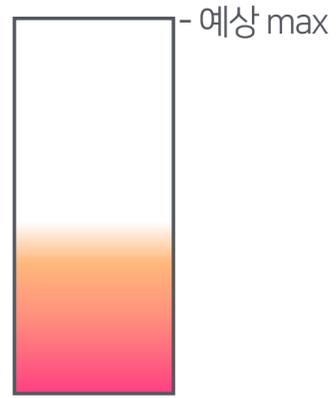
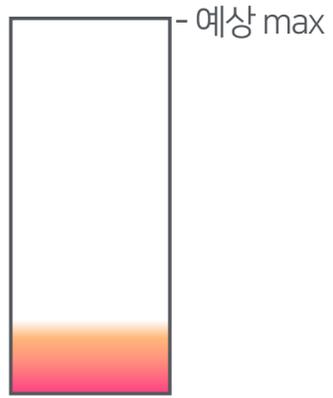
API
트래픽

Player에서 재생에 필요한 Data를 얻기 위해
Operating Server로 호출하는 API 트래픽

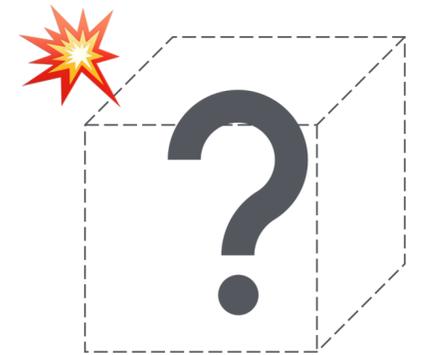
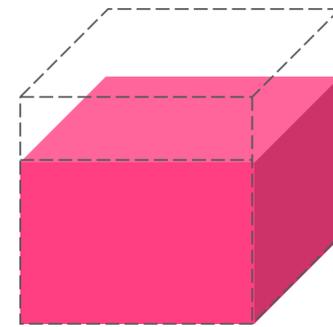
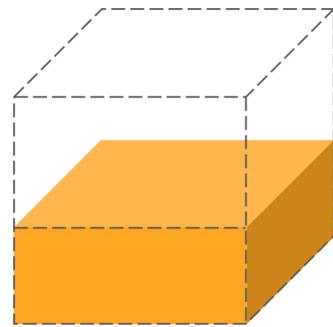
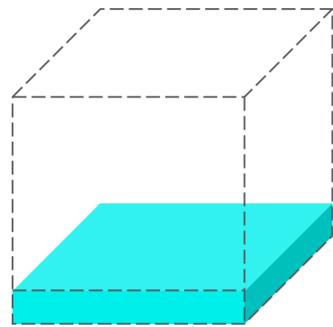
올림픽을 준비하면서 깊게 고민한 부분

트래픽

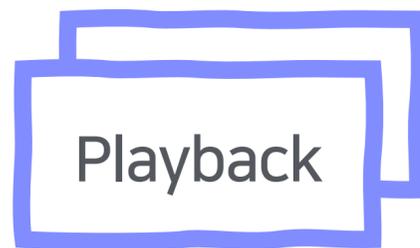
시청자수



CDN 트래픽

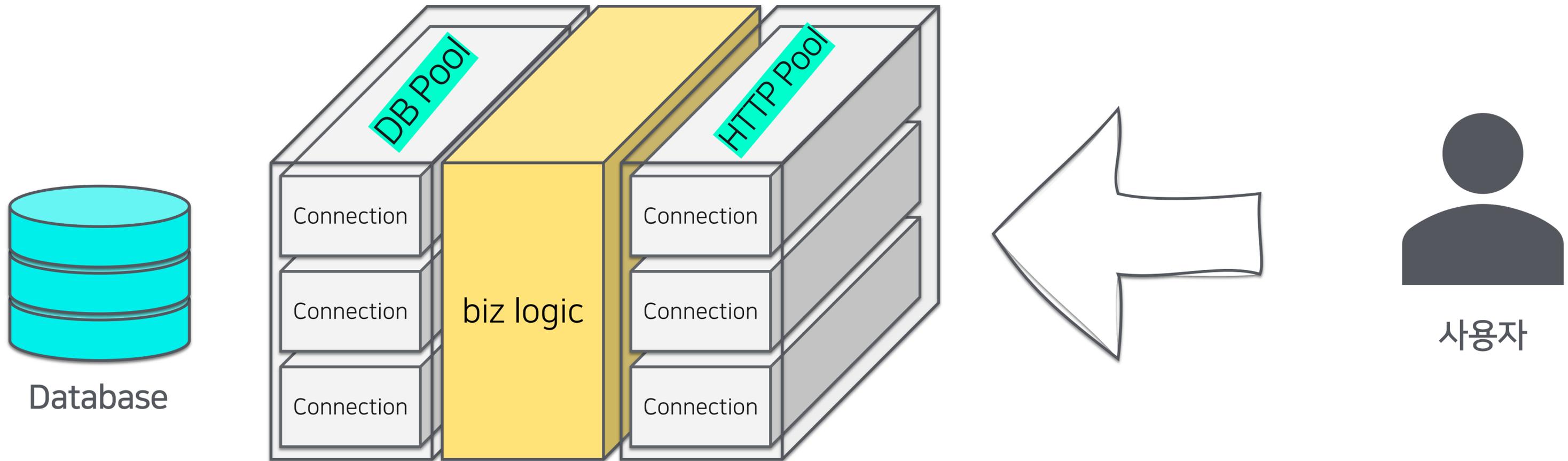


API 호출 트래픽



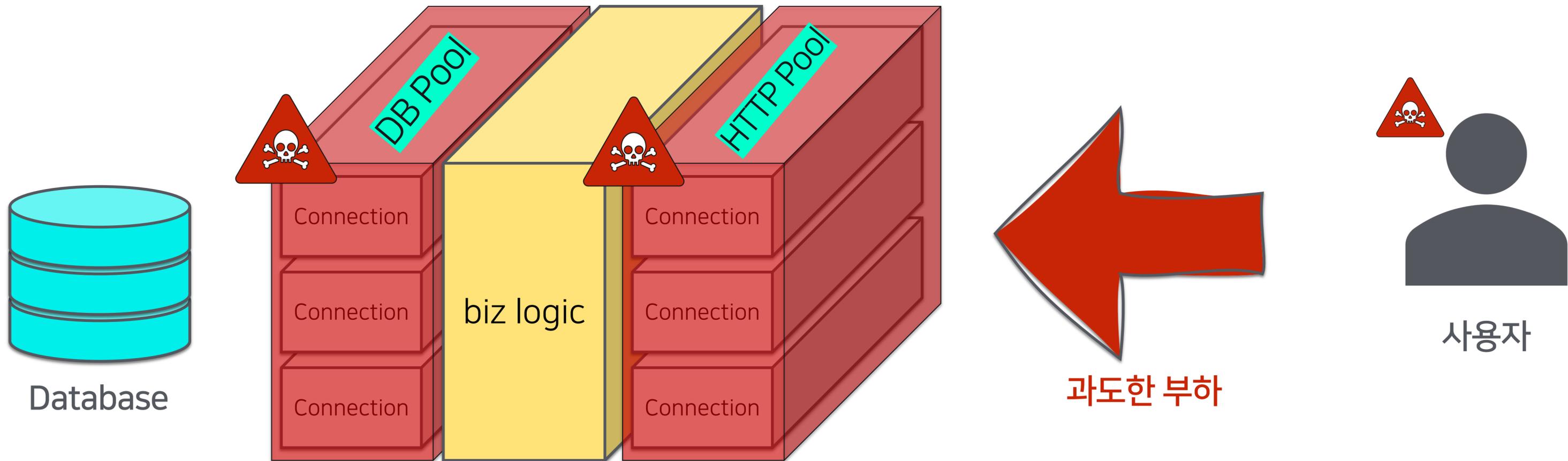
CDN 트래픽 처리에 대한 고민

일반적인 서비스에서 과부하 발생시



CDN 트래픽 처리에 대한 고민

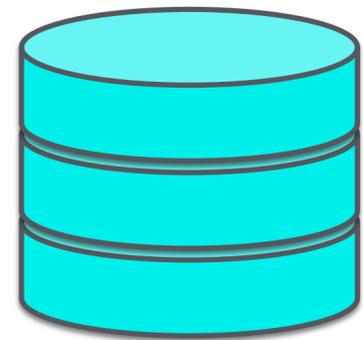
일반적인 서비스에서 과부하 발생시



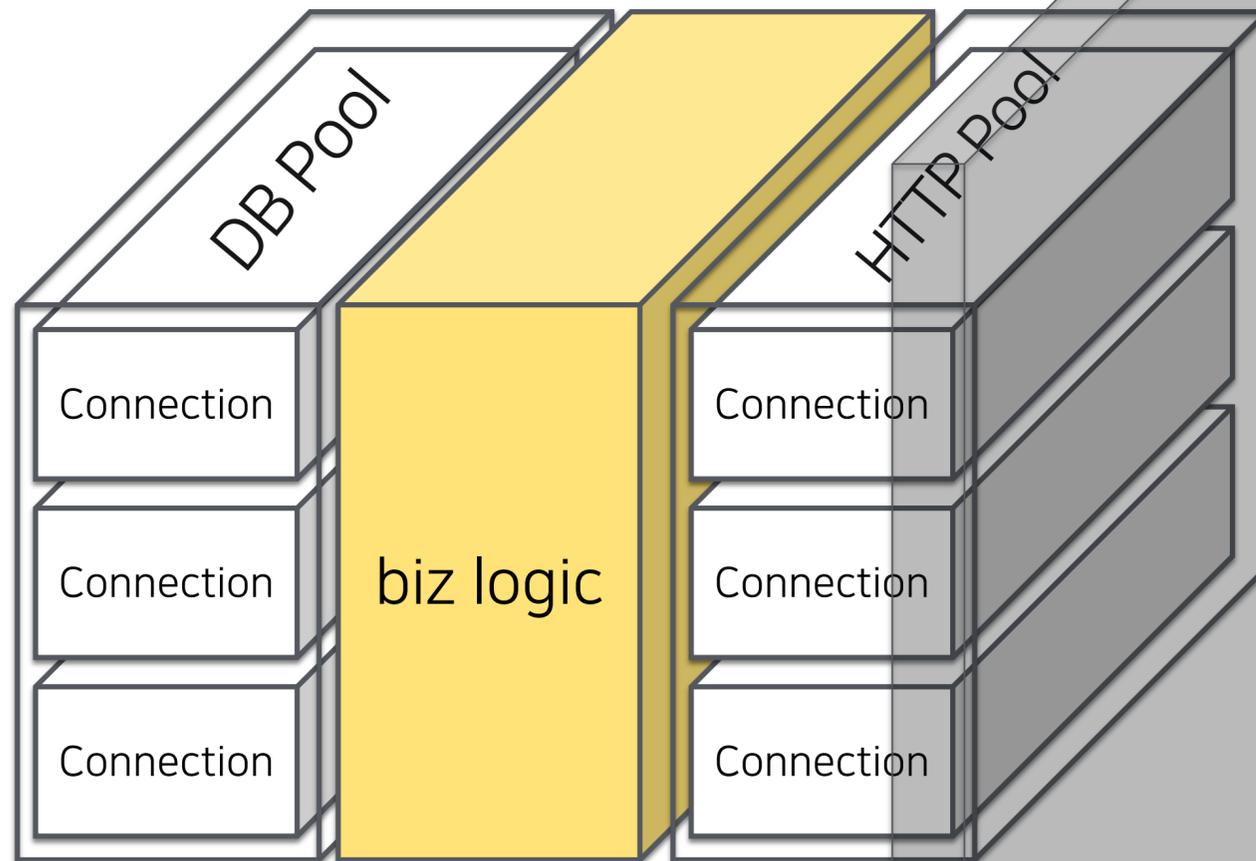
서버 내부 connection pool이나 혹은 다른 이슈 사항이 병목 포인트가 될 가능성이 높다.

CDN 트래픽 처리에 대한 고민

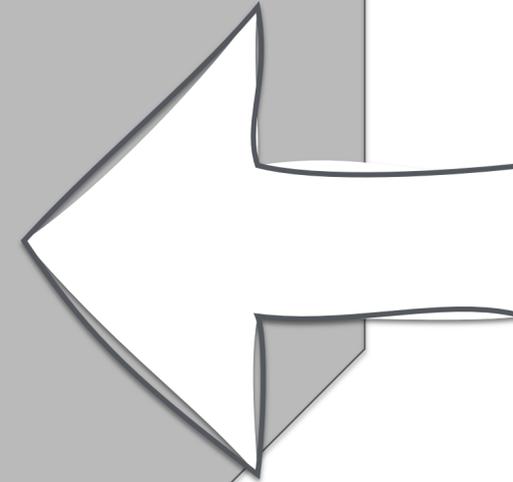
라이브 스트리밍 서비스에서 과부하 발생시



Database



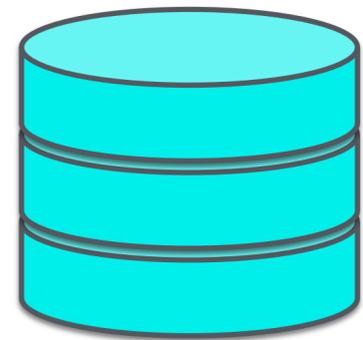
CDN 레이어



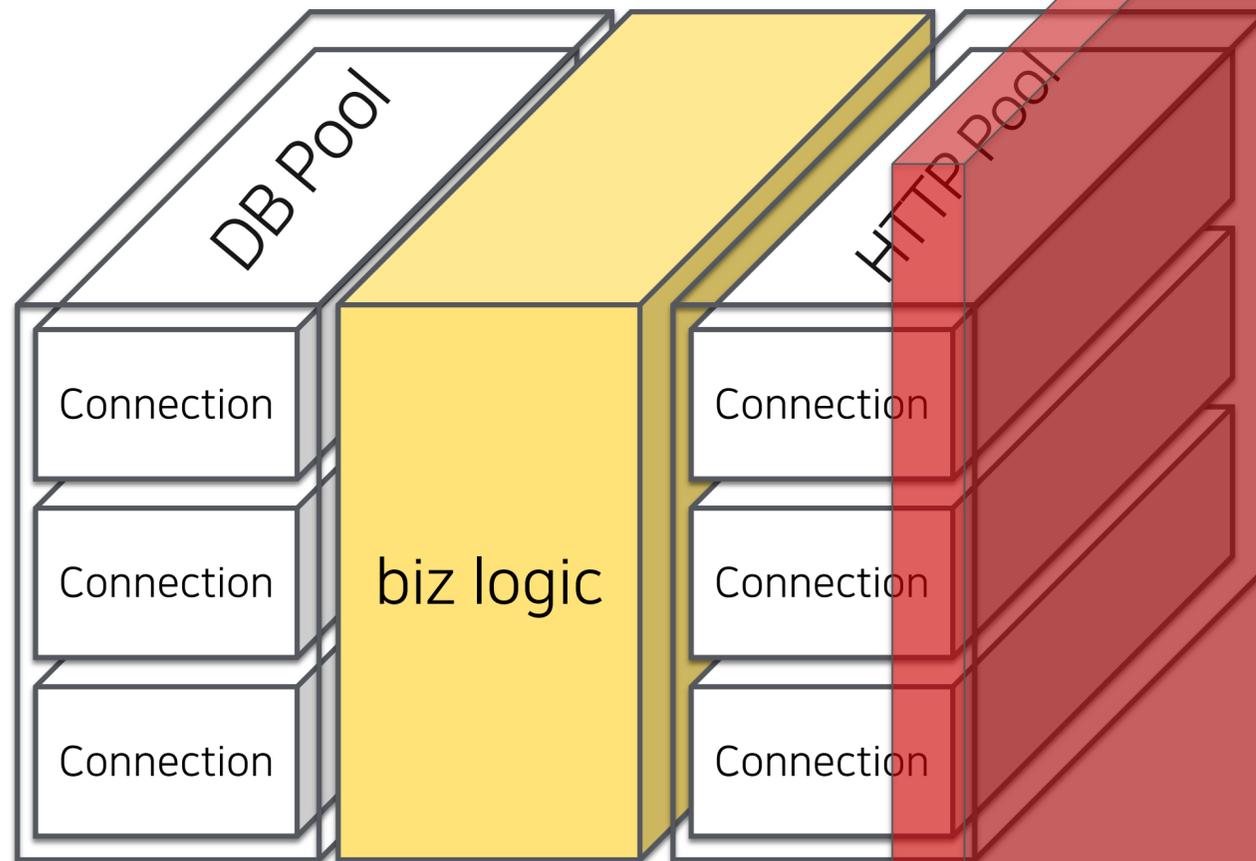
재생 플레이어

CDN 트래픽 처리에 대한 고민

라이브 스트리밍 서비스에서 과부하 발생시

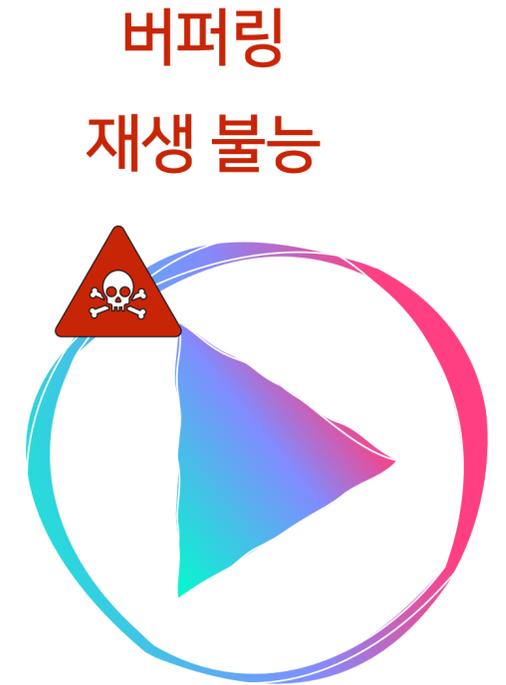


Database



CDN 레이어

과도한 부하



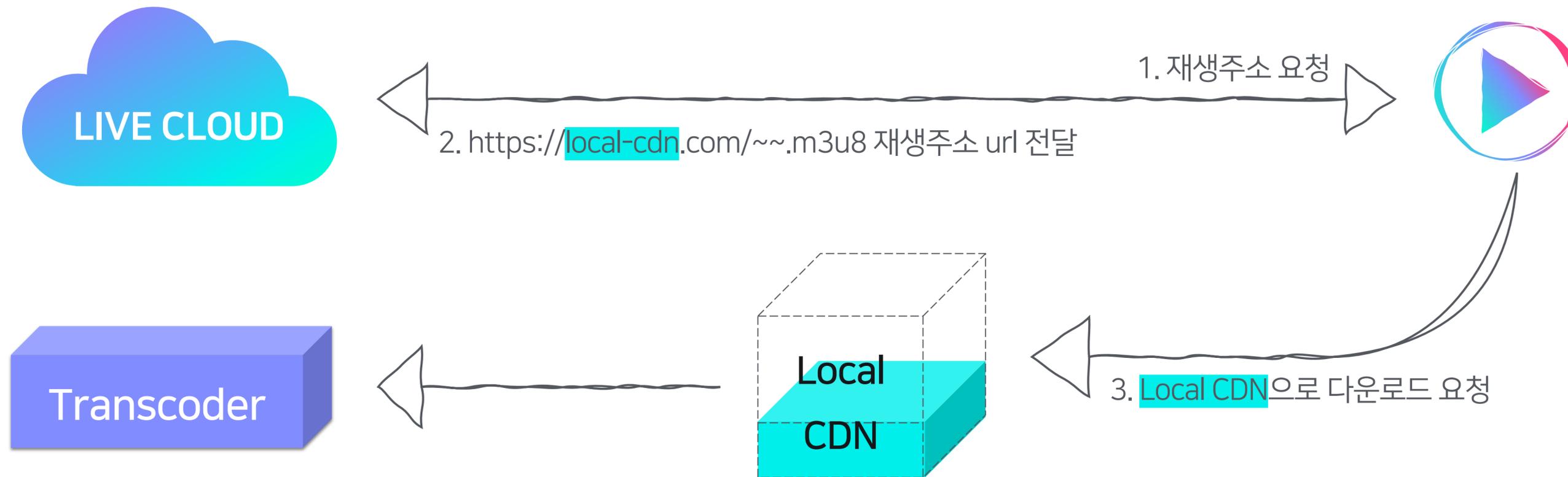
버퍼링
재생 불능

재생 플레이어

CDN의 가용트래픽을 넘어선 부하로 재생 파일 전달이 원활하지 않을 경우가 높다

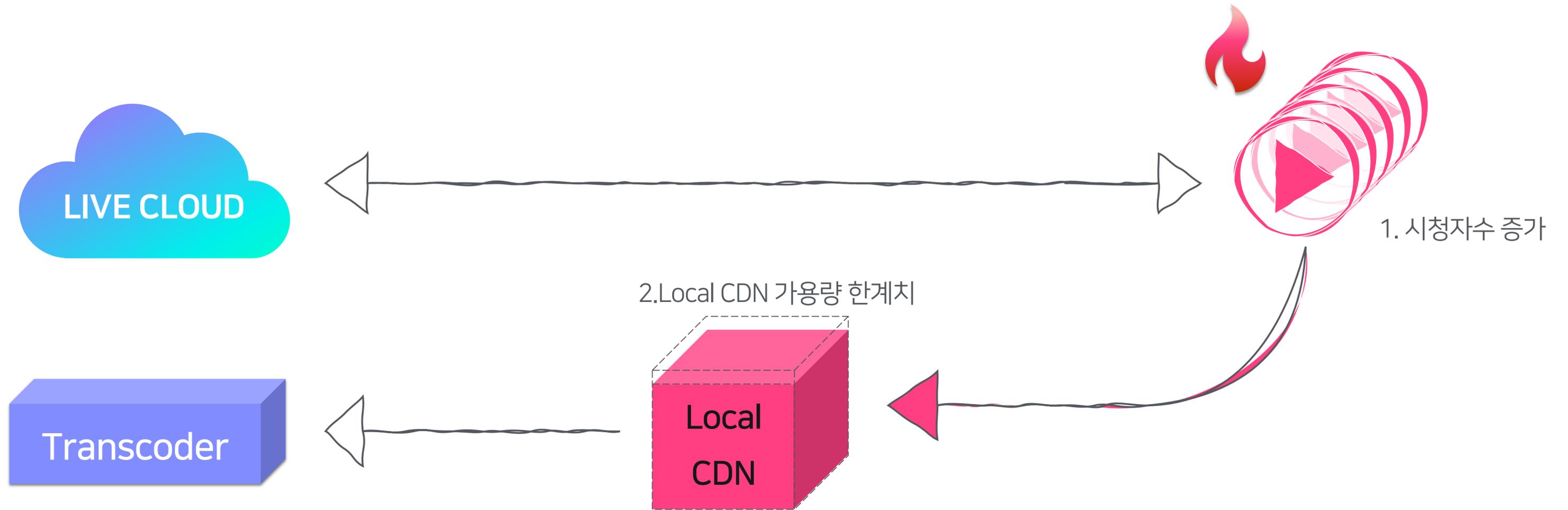
CDN 트래픽 처리에 대한 고민

대응방안 01. 멀티CDN 구성



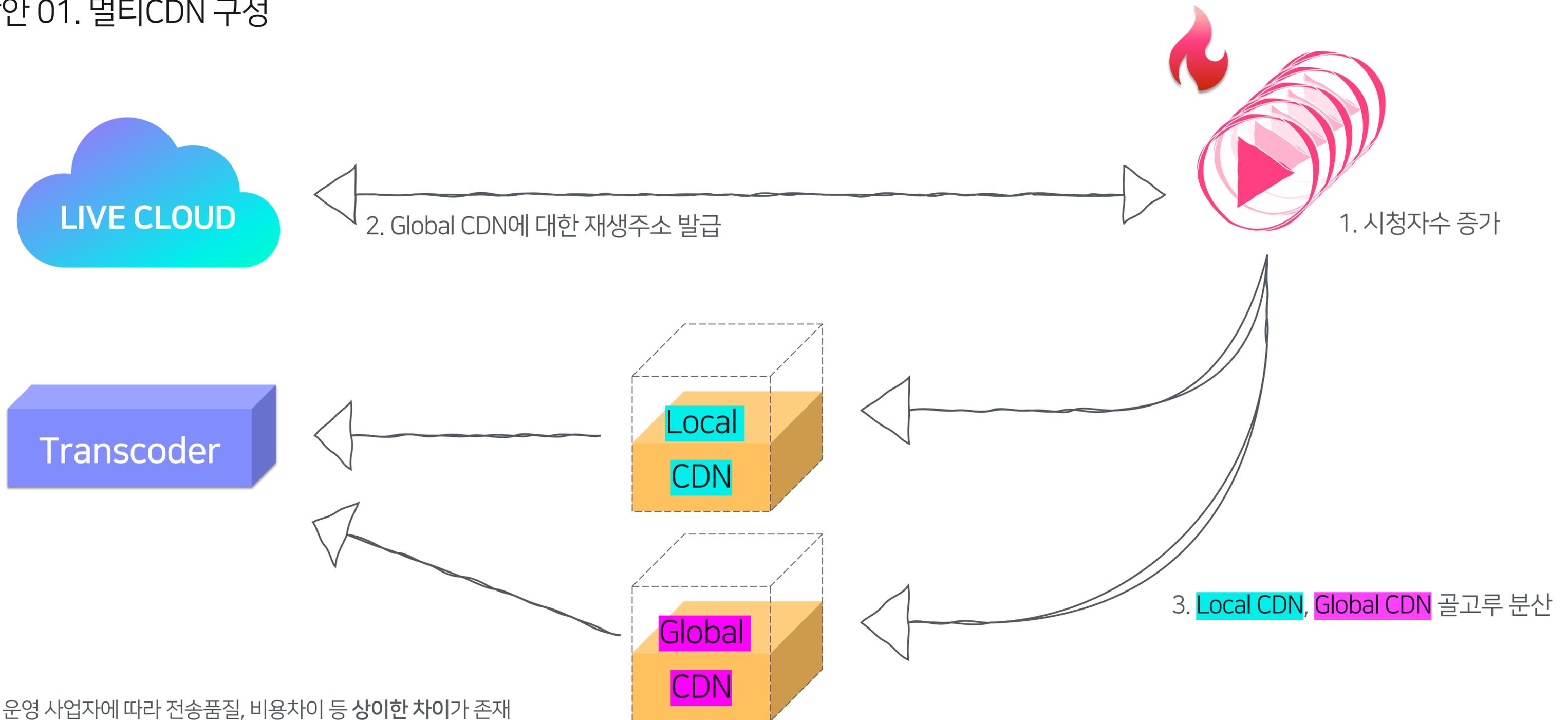
CDN 트래픽 처리에 대한 고민

대응방안 01. 멀티CDN 구성



CDN 트래픽 처리에 대한 고민

대응방안 01. 멀티CDN 구성



* CDN은 운영 사업자에 따라 전송품질, 비용차이 등 상이한 차이가 존재
평상시엔 국내 방송에는 효율성이 좋은 Local CDN만 운용하여 효율성을 가져감

CDN 트래픽 처리에 대한 고민

대응방안 01. 멀티CDN 구성

하지만,

Global CDN

- **글로벌 전용** CDN
- 국내 전송 품질이 상대적으로 좋지 않음 / 해외 전송 품질이 좋음
- 가용할 수 있는 최대 트래픽의 양이 상대적으로 많음

Global CDN

- **글로벌 전용** CDN
- 국내 전송 품질이 상대적으로 좋지 않음
- 가용할 수 있는 최대 트래픽 양 많지만 역시 한정적이다.

3. A CDN, B CDN 골고루 분산

멀티CDN이 좋은 대안이 될 순 있지만, 완벽한 해결책은 아니다.

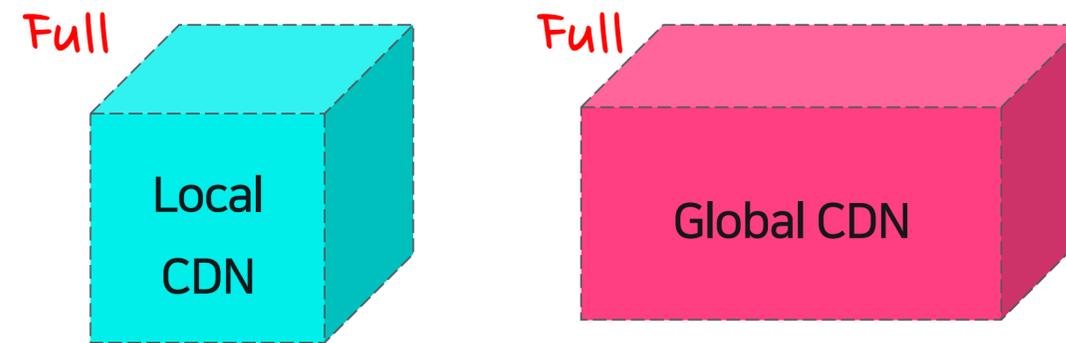
*

A, B CDN은 운영 사업자에 따라 전송품질이 다를 수 있으며, 평상시엔 국내 방송에는 효율성이 좋은 A CDN만 운용하여 효율성을 가시킵니다. 해외 방송에서는 B CDN을 운용

CDN 트래픽 처리에 대한 고민

대응방안 01. 멀티CDN 구성

멀티CDN을 사용할 수 있도록 준비했지만...

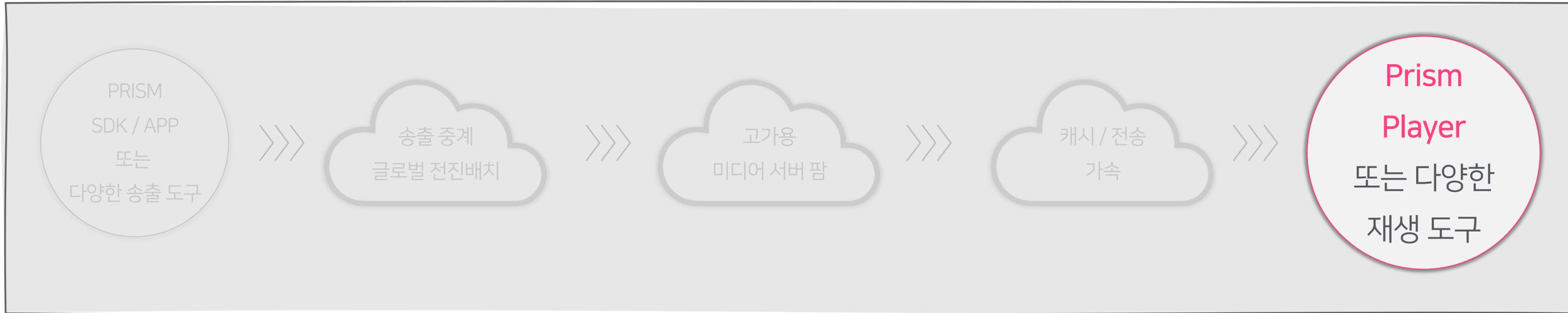


멀티CDN 가용량의 합을 모두 채우면?

시청자는 유지하면서 CDN 사용량을 줄여줄 방법이 필요하다. 🤔

CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan



Player가 Naver ETECH 영역에 포함되기 때문에 가능

LIVE CLOUD Server에서 **Player(client)의 동작 제어가 가능하다.**

서버에서 Player의 화질 변경도 되지 않을까?



시청자는 유지하면서 CDN 사용량을 줄여 줄 수 있다.

Contingency plan? : 트래픽 상황에 따라 전체 Player에서 동작을 순차적으로 제한하는 플랜

- 1단계** 국내 CDN에서 글로벌 CDN 추가 투입 (멀티CDN 적용) *CDN 가용량 확장*
- 2단계** Player에서 고화질 선택 옵션 제한 *CDN 사용량 제어*
- 3단계** 2단계 시행 전 이미 고화질로 재생중인 시청자들의 화질 하향 변경 *CDN 사용량 제어*
- 4단계** 기존 재생 유저의 재생을 보장하기 위해, 신규 재생 차단 *재생 환경 보장*

CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan

Contingency Plan을 이용하면 CDN 사용량을 줄일 수 있다.

하지만

🤔 고품질의 기준 ?

ABR, 1080, 720, 480, 360 ...

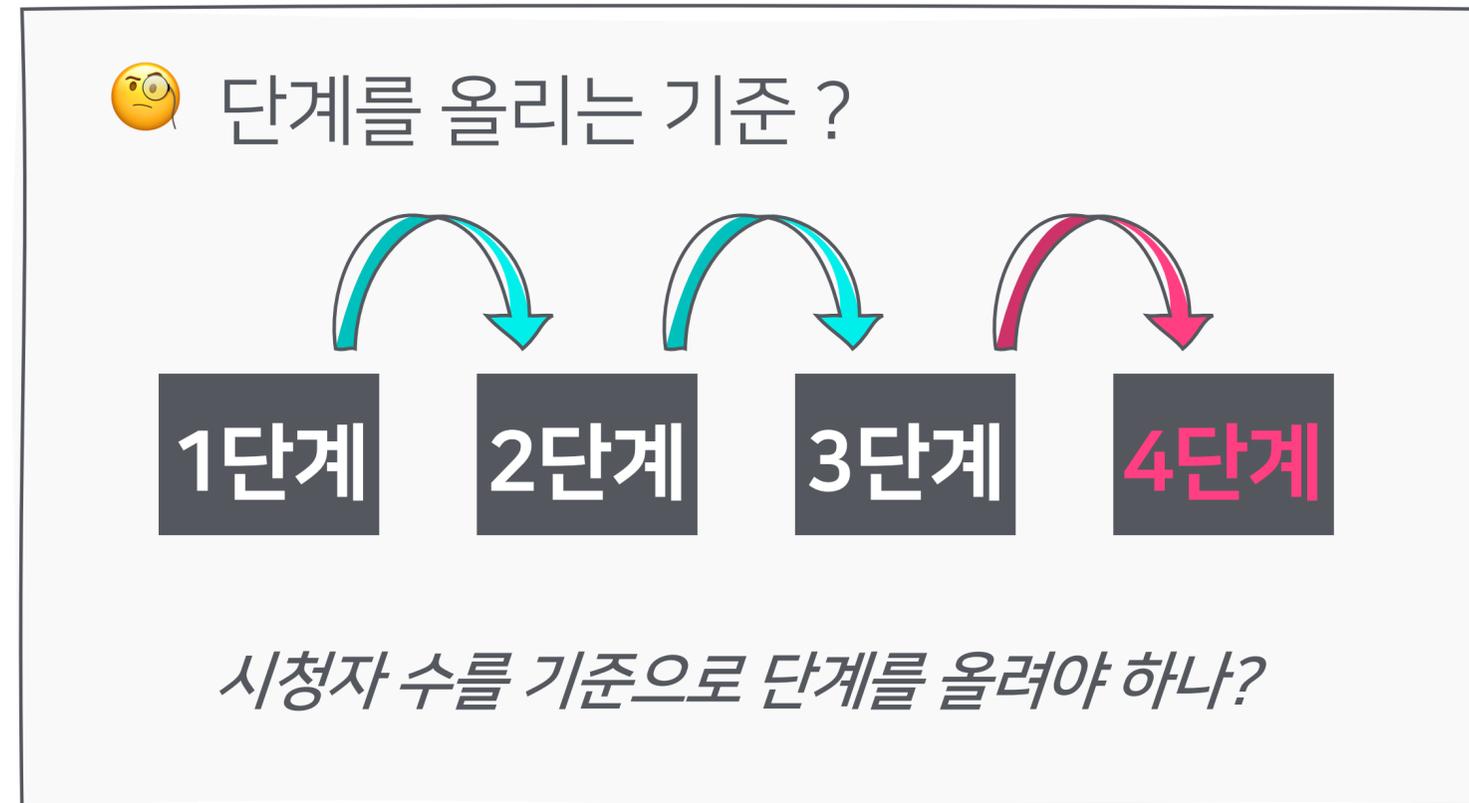
고품질의 기준은 1080일까? 720일까?

CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan

Contingency Plan을 이용하면 CDN 사용량을 줄일 수 있다.

하지만

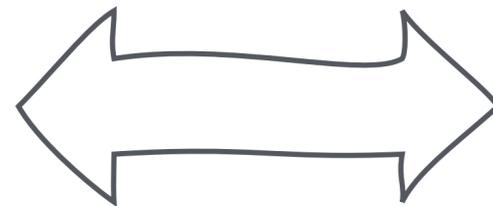


Contingency Plan을 사용할 근거(데이터)가 필요하다. 🤔

CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan

CDN 가용량은 한정적...



고품질 & 적은 사용자

저품질 & 많은 사용자

CDN 트래픽 처리에 대한 고민

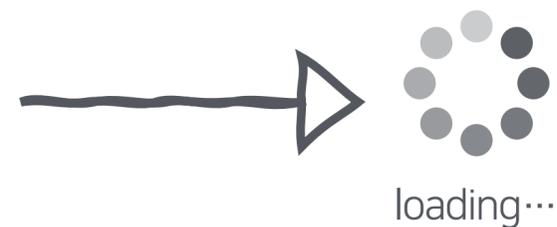
대응방안 02. Contingency Plan

QoE @player
(Quality of experience)

player에서 현재 재생중인 영상에 대한 품질 데이터를 실시간으로 전송
(단, 사용자 개인 정보는 전달하지 않습니다)

case 1.

720P → 1080P



case 2.



모바일 480p에서
계속 재생 중

QoE @player (Quality of experience)

player에서 현재 재생중인 영상에 대한 품질 데이터를 실시간으로 전송
(단, 사용자 개인 정보는 전달하지 않습니다)

시청환경

device (pc-web, mobile-app, mobile-web) & OS
시청 중인 화질
네트워크 (wifi, cellular)

재생환경

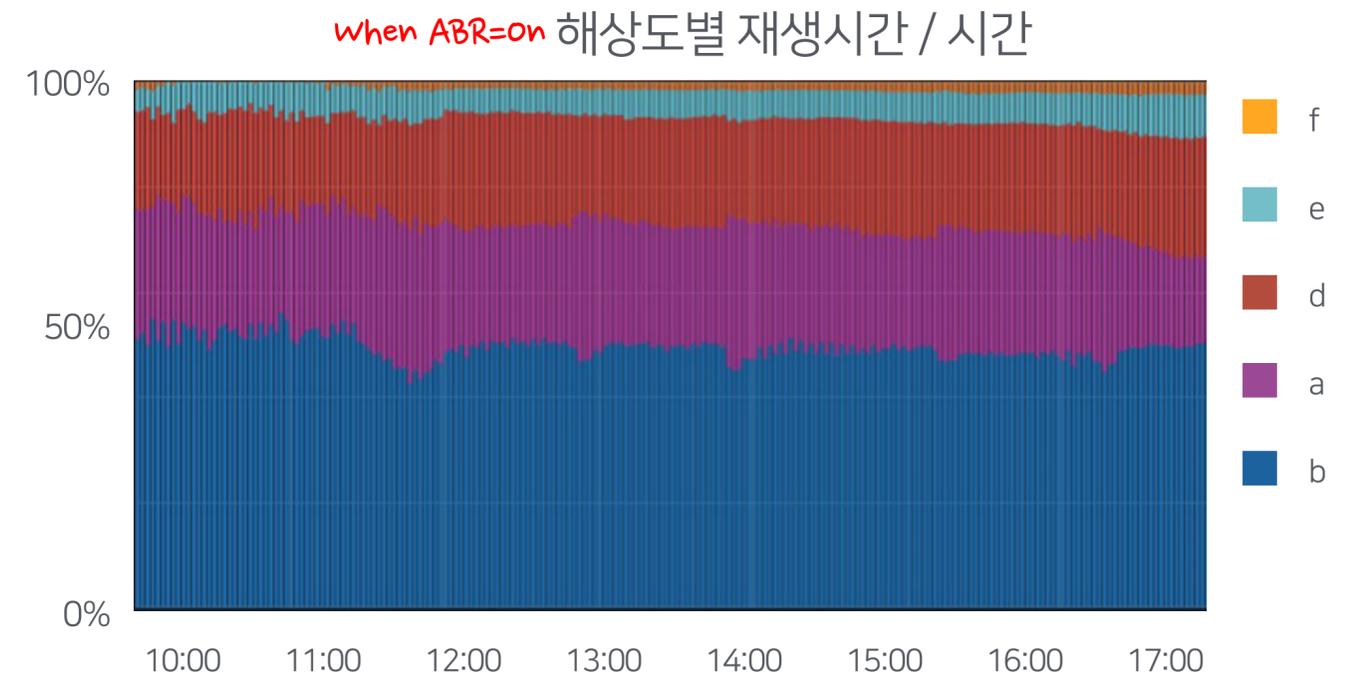
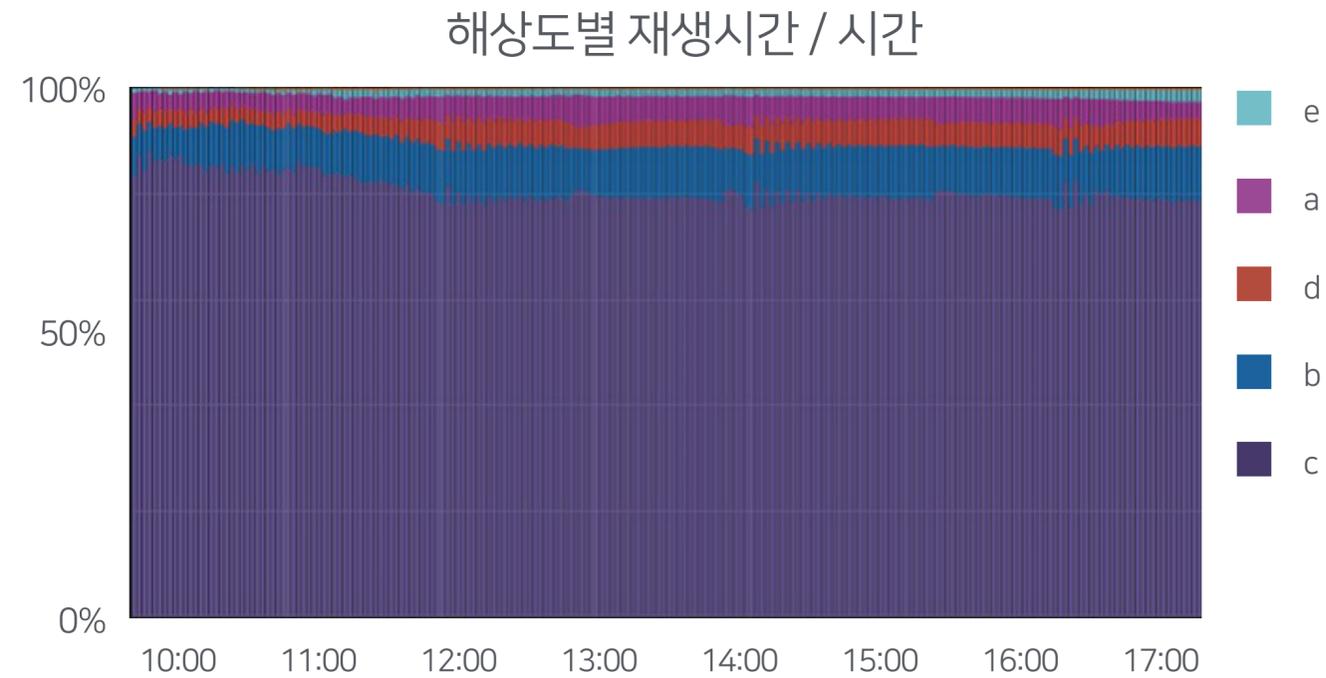
초기 재생에 걸린 시간
버퍼링 지연 시간

재생 동작성

타임머신 시청
화질 변경 시점

CDN 트래픽 처리에 대한 고민

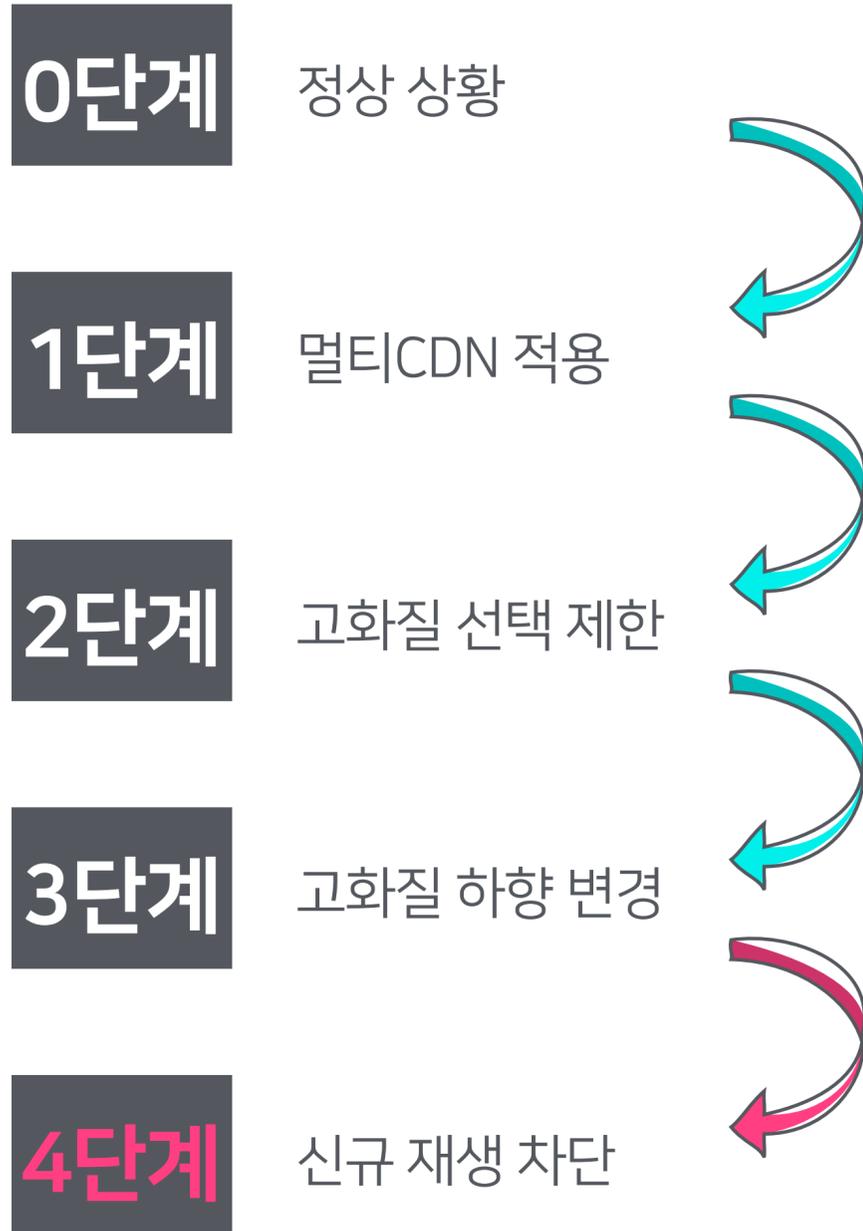
대응방안 02. Contingency Plan



QoE 수집 데이터 차트를 보고 현재 **우세한 화질, 환경을 실시간 체크** 가능

CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan

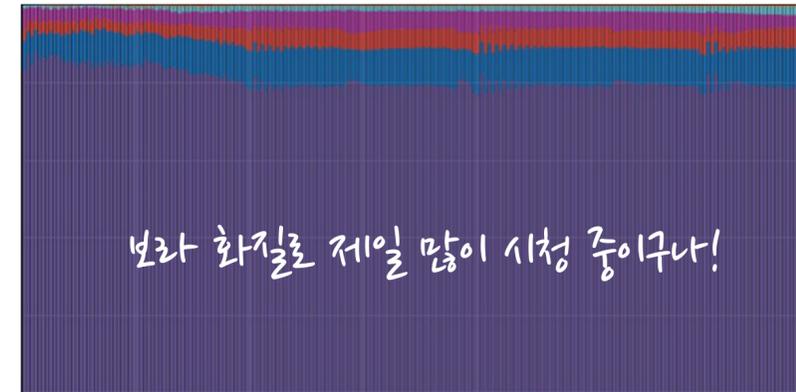


Local CDN 가용량 a% 이상

QoE 실시간 차트를 보고 제한할 고화질을 선택

QoE 차트상 제한한 화질의 사용률 여전히 높을 때

Local CDN, Global CDN 가용량 b% 이상



CDN 트래픽 처리에 대한 고민

대응방안 02. Contingency Plan



올림픽 대응 작업으로 잦은 변경과 배포



변경 사항에 무섭지 않도록

정답은 **테스트**

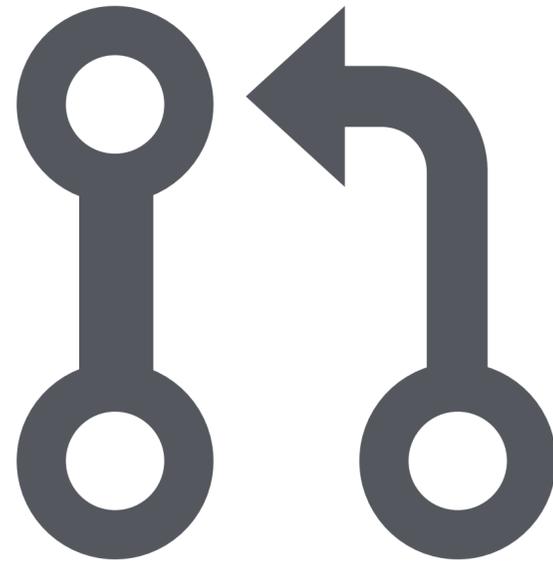
화이트 박스 테스트

블랙 박스 테스트

성능 부하 테스트

API 트래픽 처리에 대한 고민

화이트 박스 테스트 - 코드 pr시 Unit test 매번 실행



PULL REQUEST

Jenkins plugin : GitHub Pull Request Builder

pull request 생성을 감지하고 미리 지정한 Jenkins job 실행



Unit test result & coverage

코드 정적 분석

Jenkins job 성공해야만 merge 가능

API 트래픽 처리에 대한 고민

블랙 박스 테스트 - 배포 직후 시나리오 테스트 수행

이번에 수정한 부분은 테스트 완료.

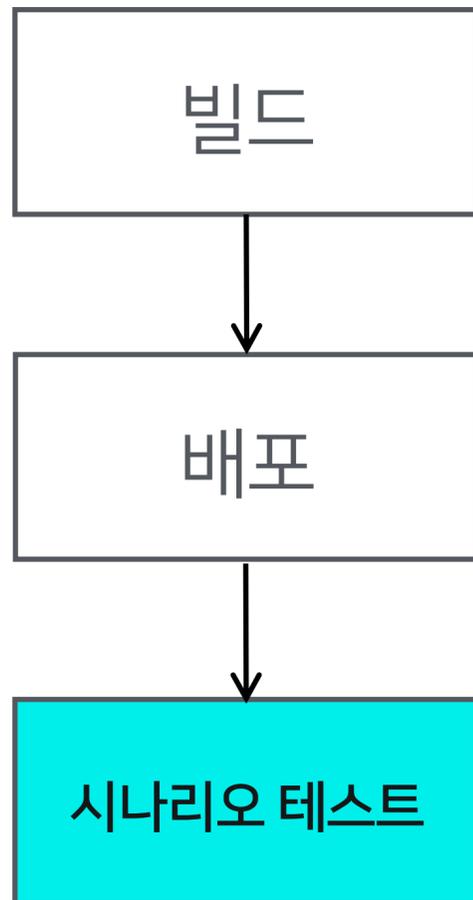
혹시... 이번 수정이 다른 로직에 영향을 주진 않을까?

모든 API와 로직을 직접 다 테스트 해봐야 하나?



API 트래픽 처리에 대한 고민

블랙 박스 테스트 - 배포 직후 시나리오 테스트 수행



시나리오 테스트

작성한 시나리오(Collections) trigger

Collections

01. A 라이브 정상 시나리오

POST /lives - 라이브 생성
GET /lives/{liveId} - 라이브 객체 생성 확인
POST /lives/{liveId}/start - 라이브 시작 요청
GET /lives/{liveId}/status - 라이브 시작 상태 확인
...

02. A 라이브 예외 케이스 시나리오 1

03. A 라이브 예외 케이스 시나리오 2

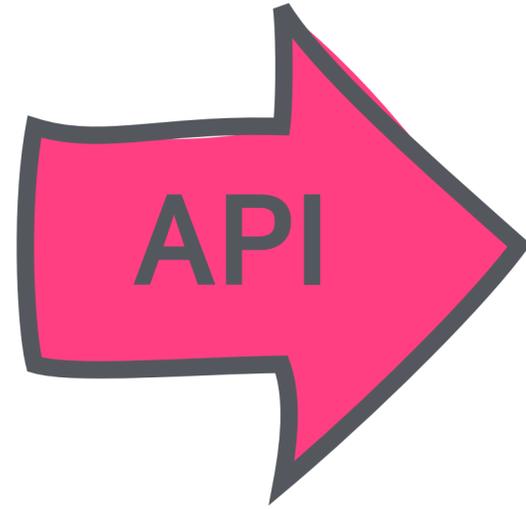
단편적인 API 호출이 아닌,
실제 사용하는 시나리오(API 묶음)을 실행

API 트래픽 처리에 대한 고민

성능 부하 테스트 - 만반의 준비를 위해 기계적인 테스트

기계적인 반복 테스트

API 부하 발생기

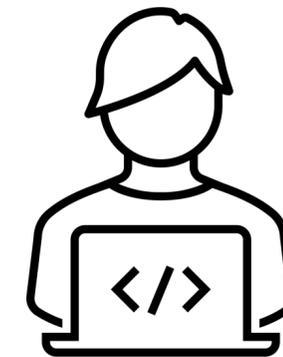


LIVE CLOUD
bulk test env for Olympic

API 요청량에 따른 container 증가
각 container 리소스(cpu, memory, log storage) 사용량 체크



사용량은 Grafana로 확인



반복 테스트로 문제점 발견!



Chat-bot

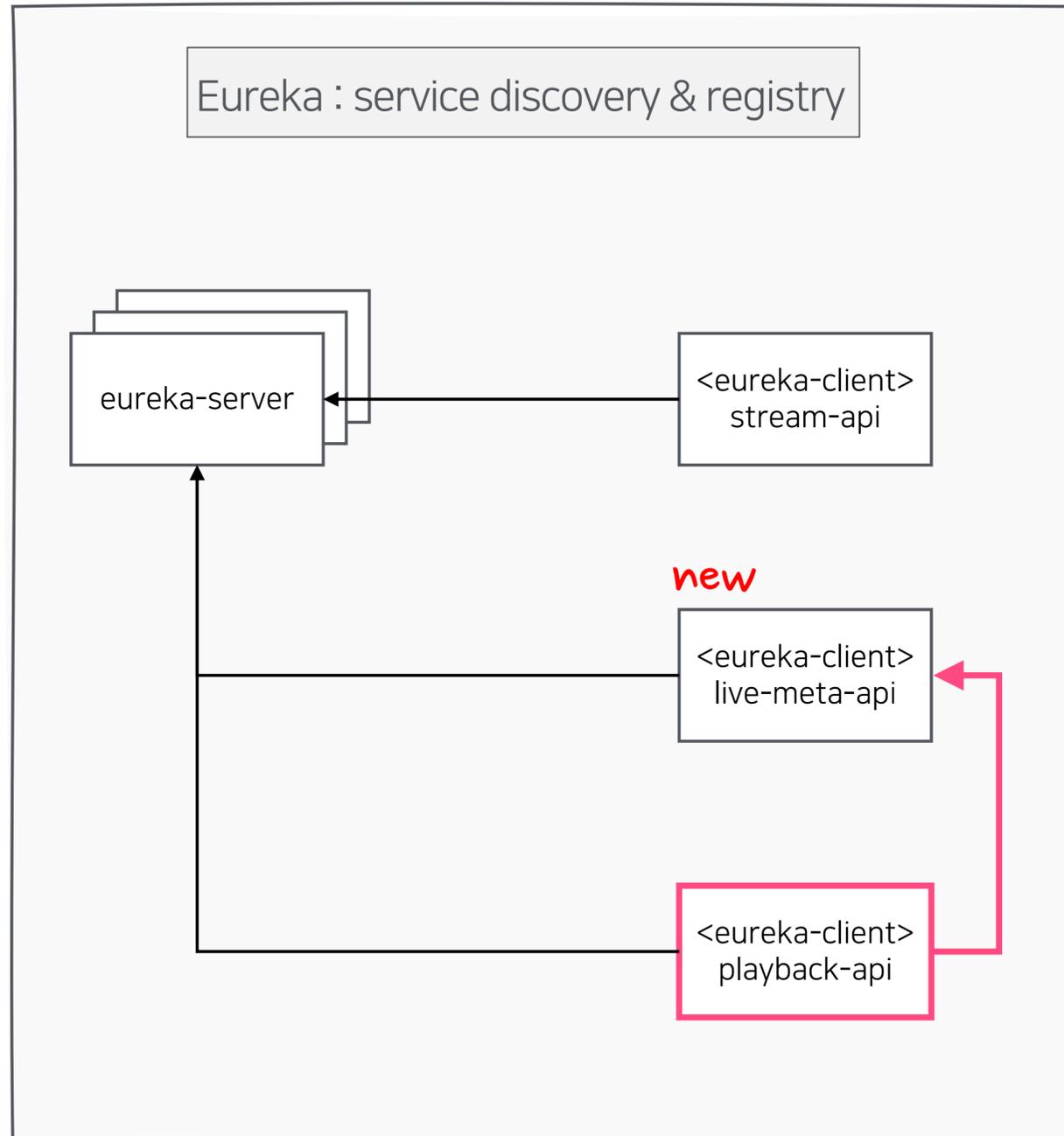
결과 데이터 분석

채팅으로 통보



API 트래픽 처리에 대한 고민

테스트에서 발견한 문제점



1. 과도한 API 부하를 주고 있는 상황

2. playback에서 eureka-server가 알려준 이웃한 Service로 API 호출

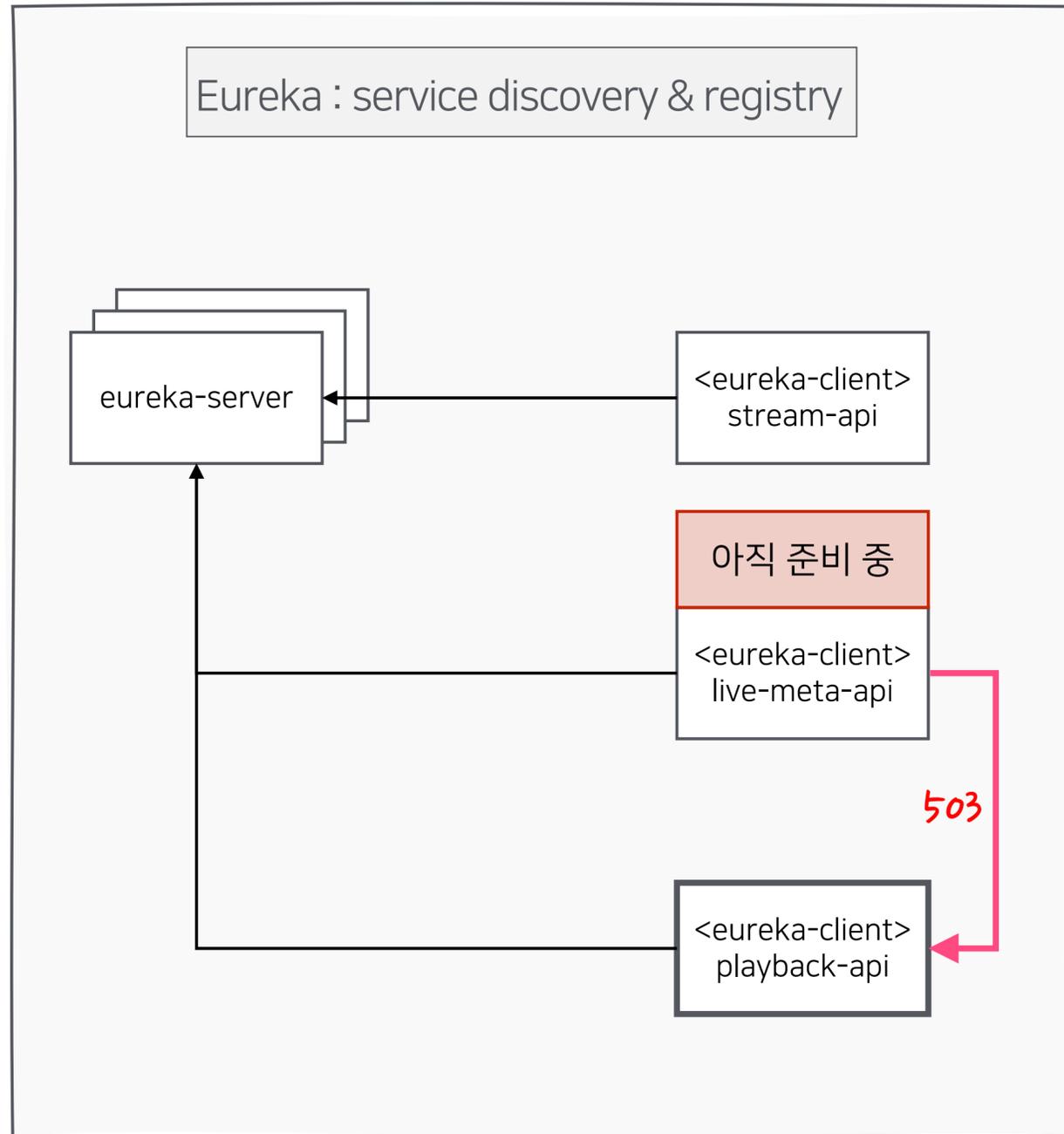
3. 여기서 전달받은 service가 트래픽이 몰린 상황에서 막

*새로 생성된 container*로 전달 받음

4. API 실패 응답 🤯

API 트래픽 처리에 대한 고민

테스트에서 발견한 문제점



원인은 호출을 받은 서버의 상태 healthiness O, liveness X

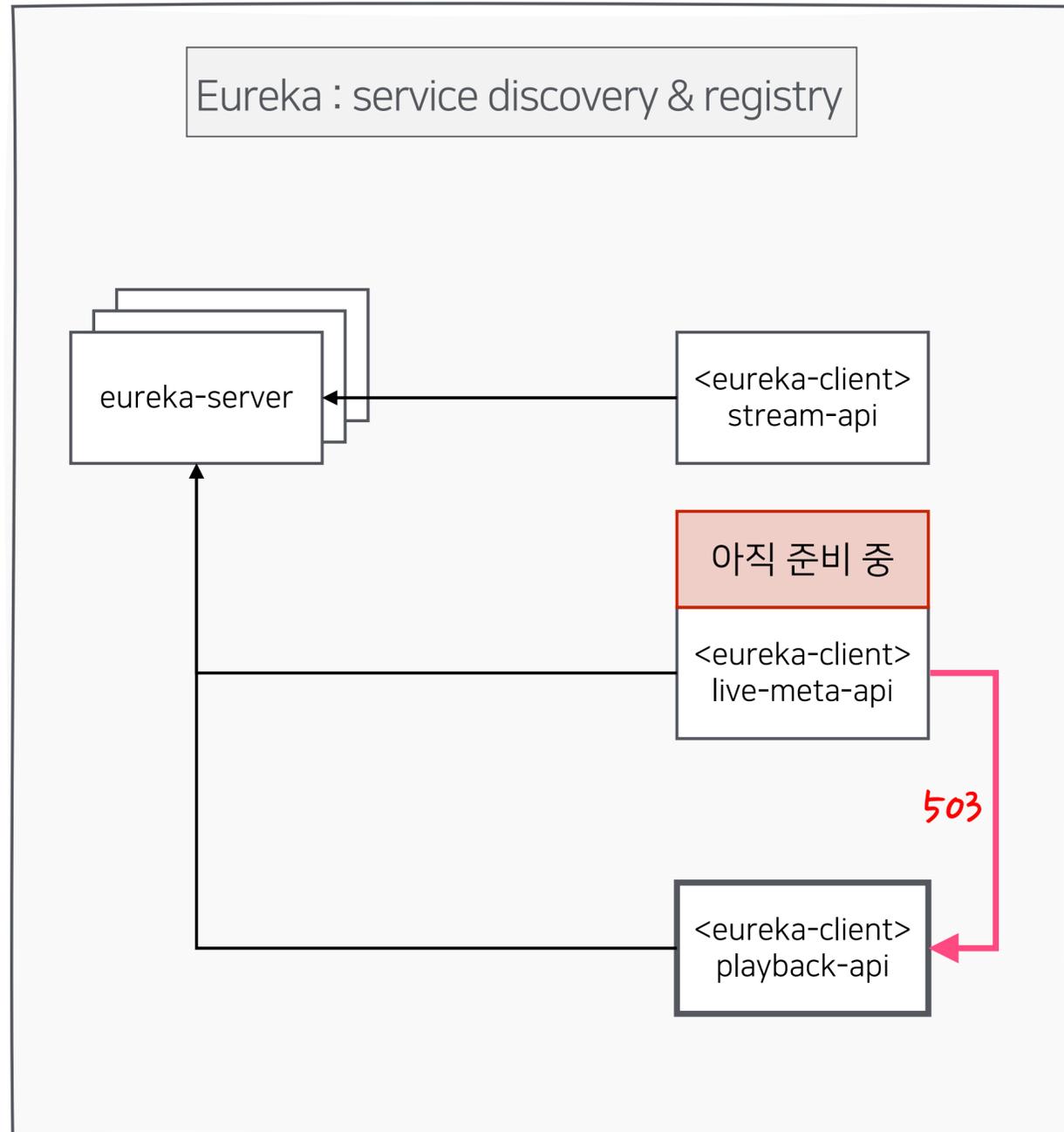
새로 생성된 *container*가 spring boot launch
→ eureka-client 등록하였지만

DB 접근을 위한 ACL 권한 획득에 지연 중인 상황

서버가 실행되어 API는 수신하였지만, 정상 서비스 준비는 아직 덜 된 상태

API 트래픽 처리에 대한 고민

테스트에서 발견한 문제점



게다가 DB ACL 뿐만 아니라 내부적으로
상태체크를 해야할 component 항목이 많음

위 component 체크는 매우 빠른 시간에 처리되긴 합니다.
하지만,

과도한 트래픽으로 container가 계속 생성 중인 상황에선
해당 문제의 발생 가능성이 높다.

API 트래픽 처리에 대한 고민

테스트에서 발견한 문제점

😊 개선 포인트

spring-boot 시작 시점과 eureka register 시점을 분리

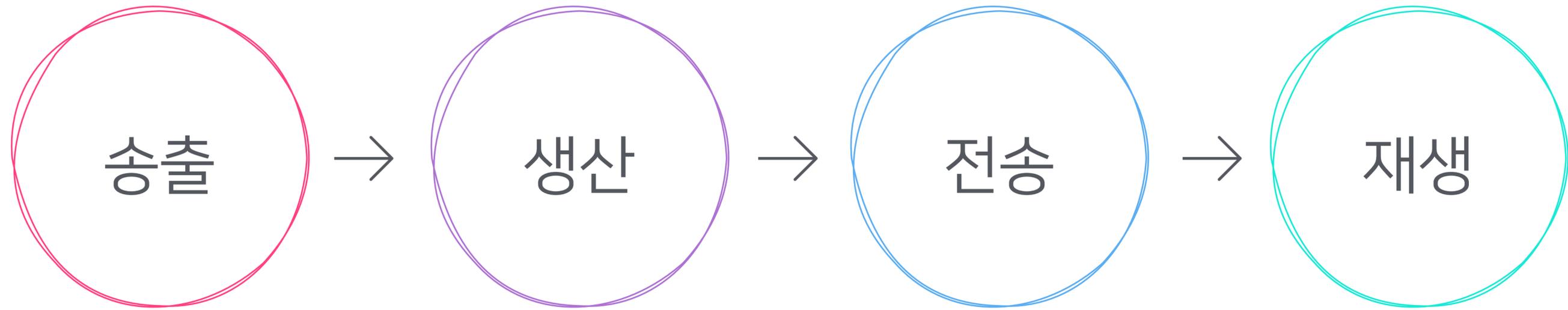
상태 체크가 완료되고 정상 서비스가 가능하다 판단 된 이후 eureka-client 등록

마음이 편한 배포
반복적인 성능 부하 테스트

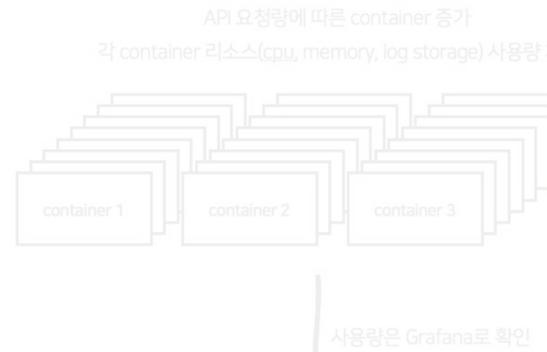


문제점을 조기에 발견, 해결!

한정된 자원으로 최대한의 시청자가 불편함 없이 재생 가능 하도록 최선의 노력

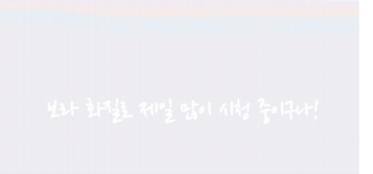


NAVER LIVE CLOUD



- 0단계 정상 상황
- 1단계 멀티CDN 적용
- 2단계 고화질 선택 차단
- 3단계 화질 강제 하향
- 4단계 CDN 재생

A CDN 가용량 a% 이상



QoE 실시간 차트를 보고 차단할 고화질을 선택 후 차단

QoE 차트상 차단한 화질의 사용률 여전히 높을 때

3단계를 하였지만 A CDN 가용량 a% 이상

최대 동접

올림픽 3사

무장애

QoE 140만

중계 70회

라이브 완료

시청환경	device (pc-web, mobile-app, mot) 시청 중인 화질 네트워크 (wifi, cellular)
재생환경	초기 재생에 걸린 시간 버퍼링 지연 시간
재생 동작성	타임머신 시청 화질 변경 시점

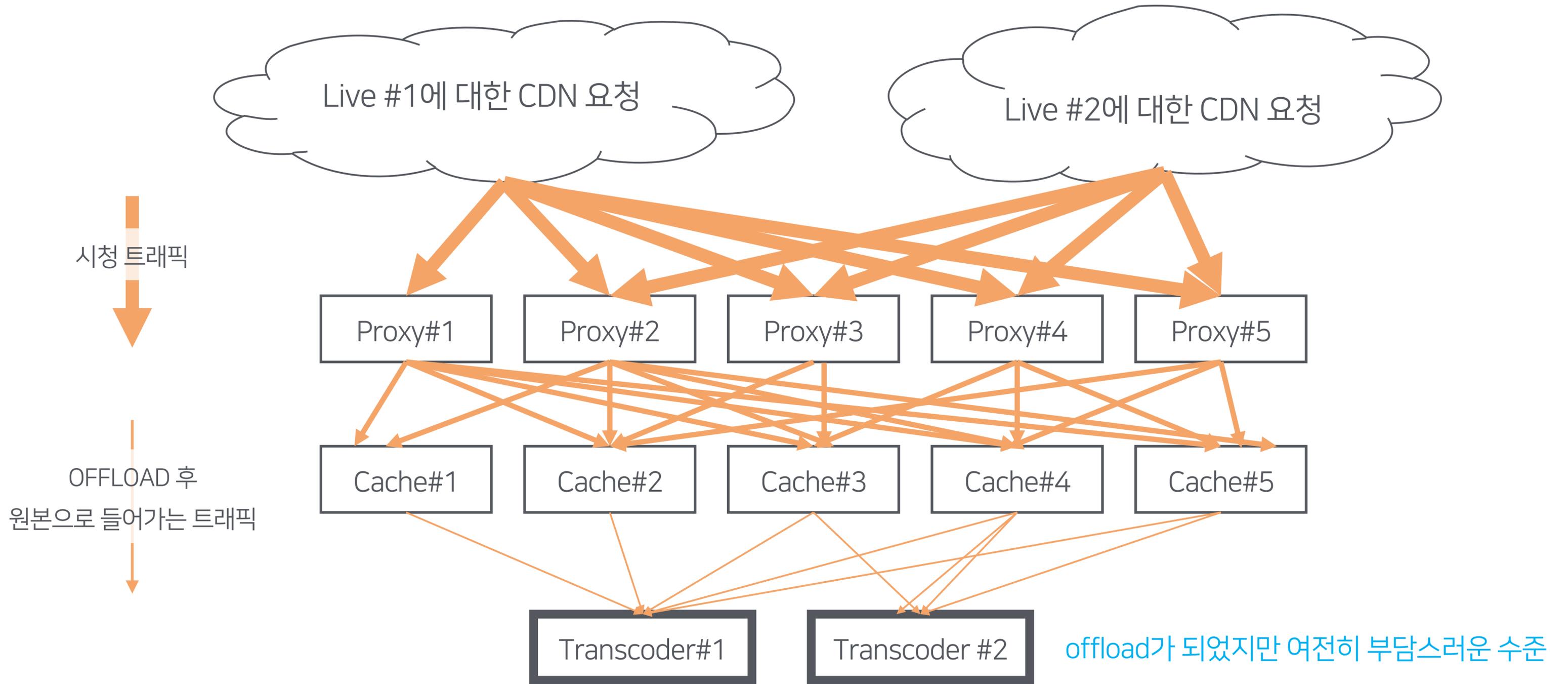


4. 보다 나은 다음 올림픽을 위해

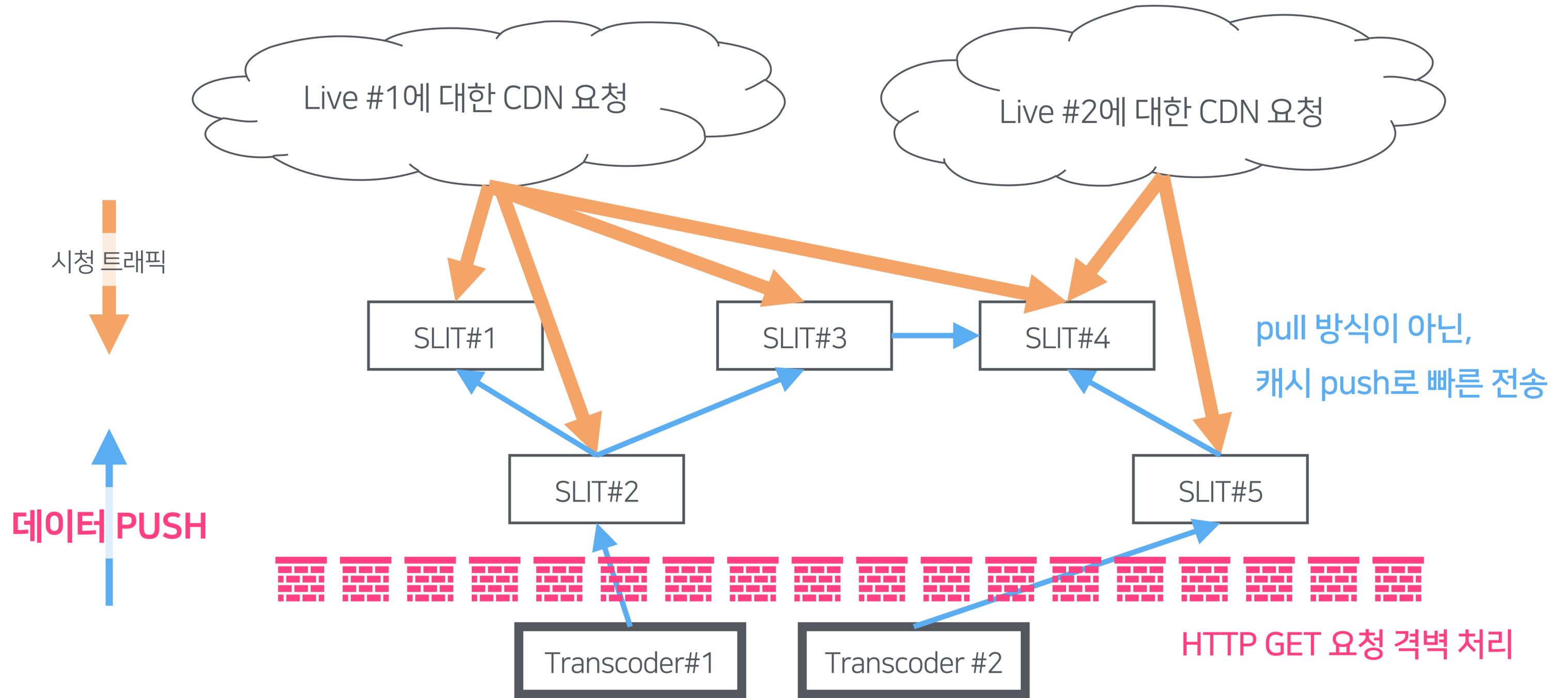
트래픽

새로운 라이브

트래픽 : 글로벌 콘텐츠 전송 플랫폼 SLIT



트래픽 : 글로벌 콘텐츠 전송 플랫폼 SLIT





Multiview Live

하나의 영상 파일로 여러 화면을 동시 시청

(위 사진은 설명을 위한 연출입니다)



Immersive Live

다중 음향 채널로 공감각을 극대화

NAVER

Emerging

TECHnology

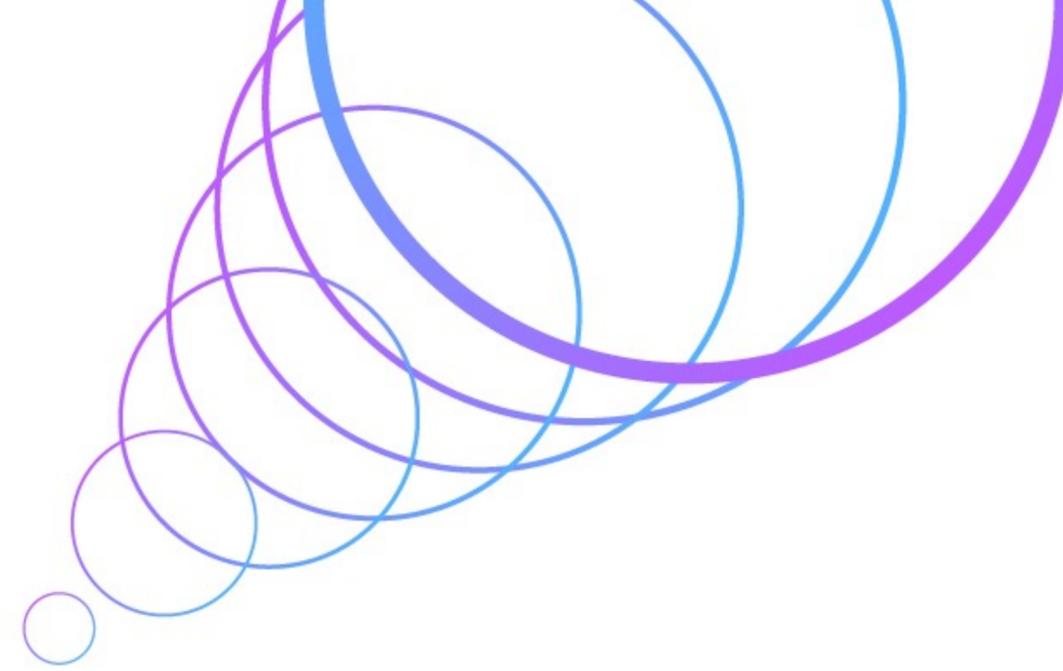
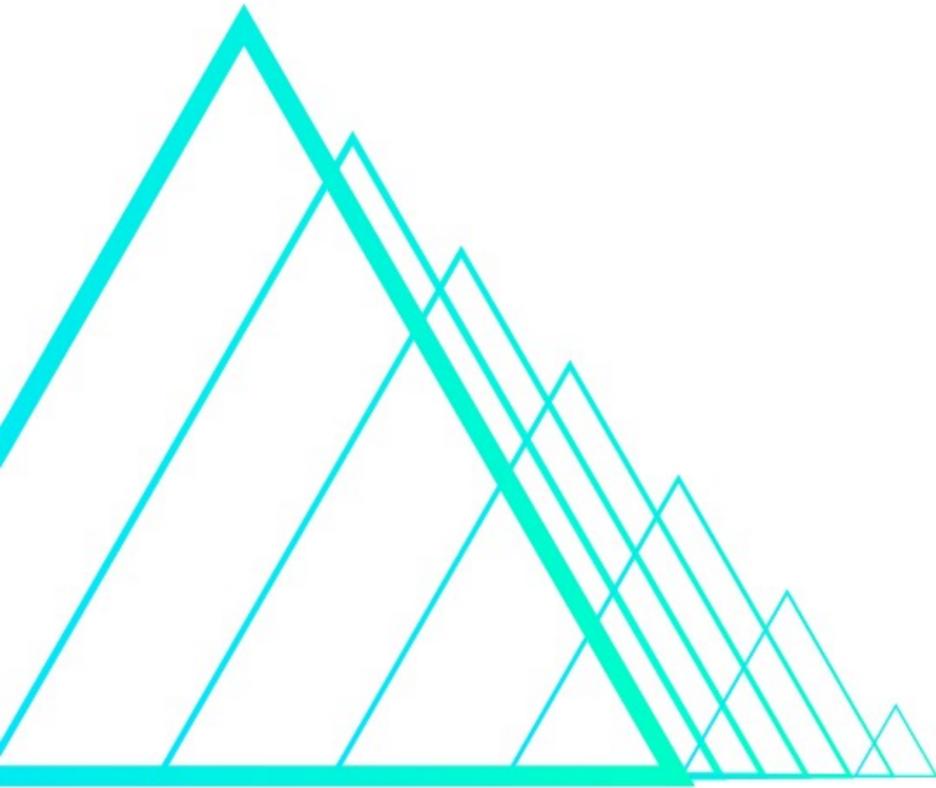
함께 성장하실 동료분을 모십니다

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

