

# CLOVA Vision :

NAVER AI, Cloud, IP Network Camera의 만남

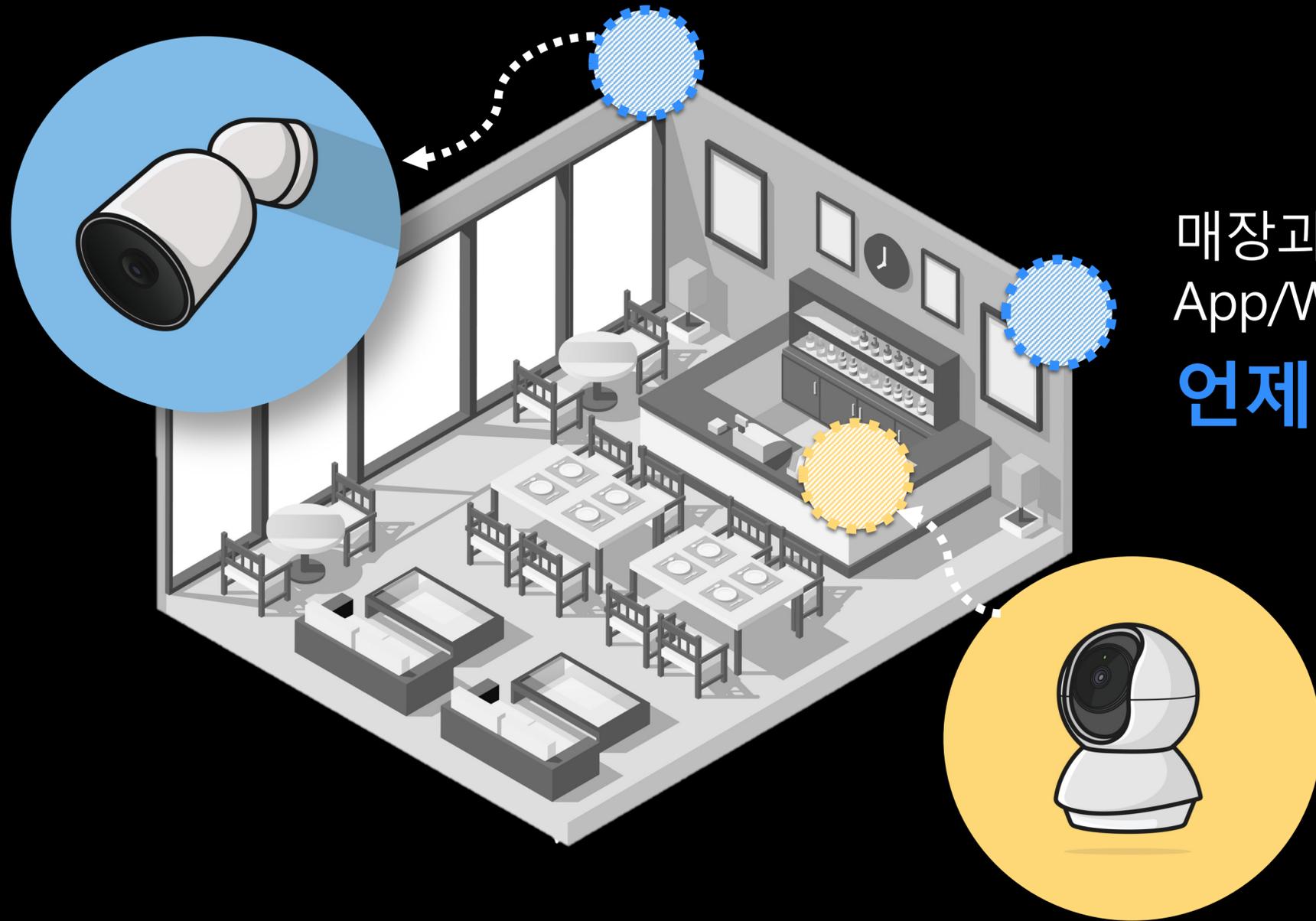
# 일본 AI Cloud Camera 개발 도전기

최순원/장준영/정재호

NAVER CLOVA

NAVER DEVIEW 2023

# Security Service 란?



매장과 같은 공간에 카메라를 설치하여  
App/Web을 통해 실시간/녹화영상 원격 확인 가능  
**언제 어디서나 안심할 수 있는 서비스**

# CONTENTS

1. CLOVA Vision 소개
2. CLOVA Vision 개발 과정 되돌아 보기
3. IP Network Camera 기반의 클라우드 카메라 디바이스 개발
4. Cloud Camera 연결 관리 및 제어
5. CLOVA Vision 영상 데이터 처리
6. CLOVA Vision과 AI의 만남

# 1. CLOVA Vision 소개

# 1.1 CLOVA Vision

LINE

NAVER CLOVA

NAVER Cloud

IDIS®

SoftBank

'22.09 Released

Security Service

To do

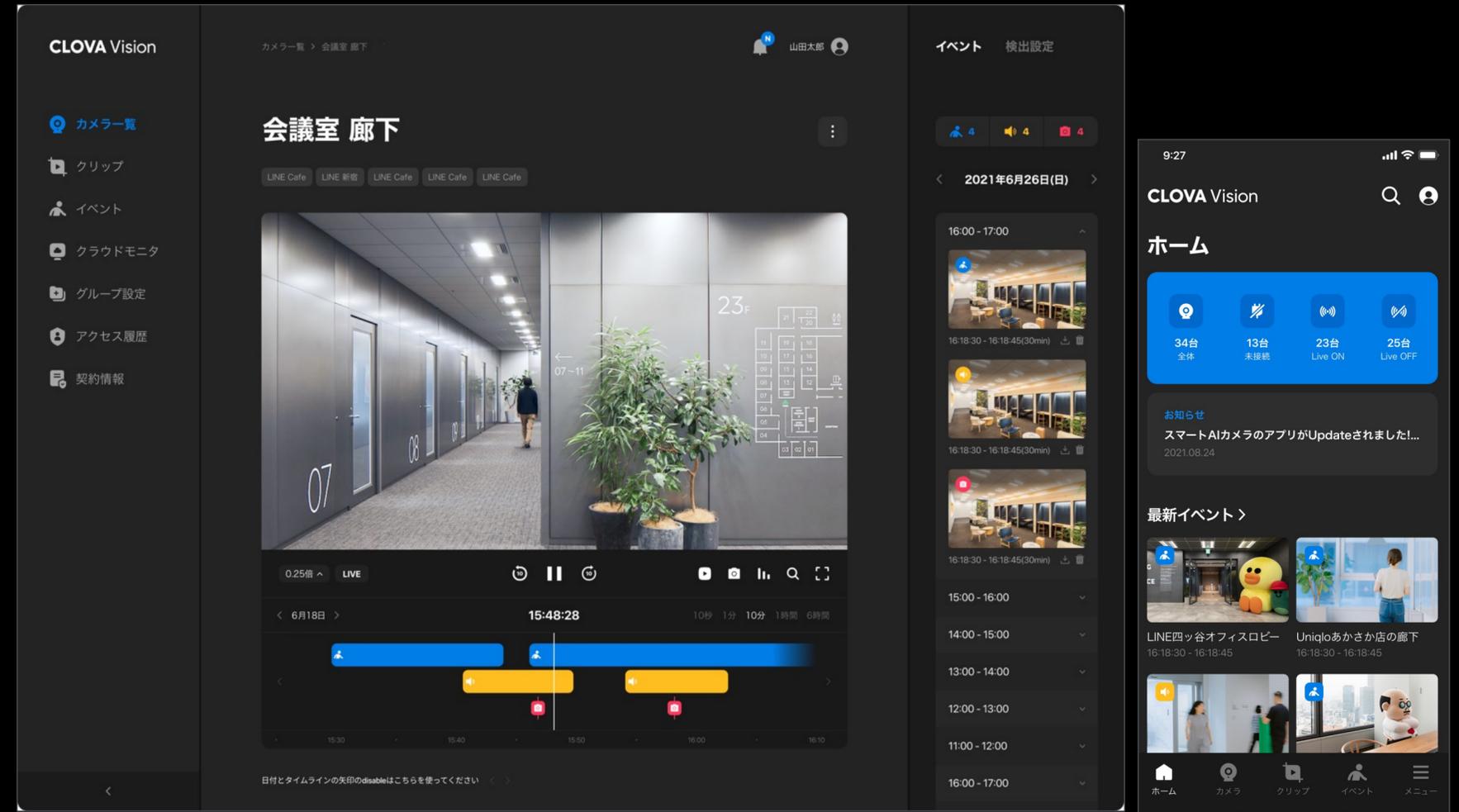
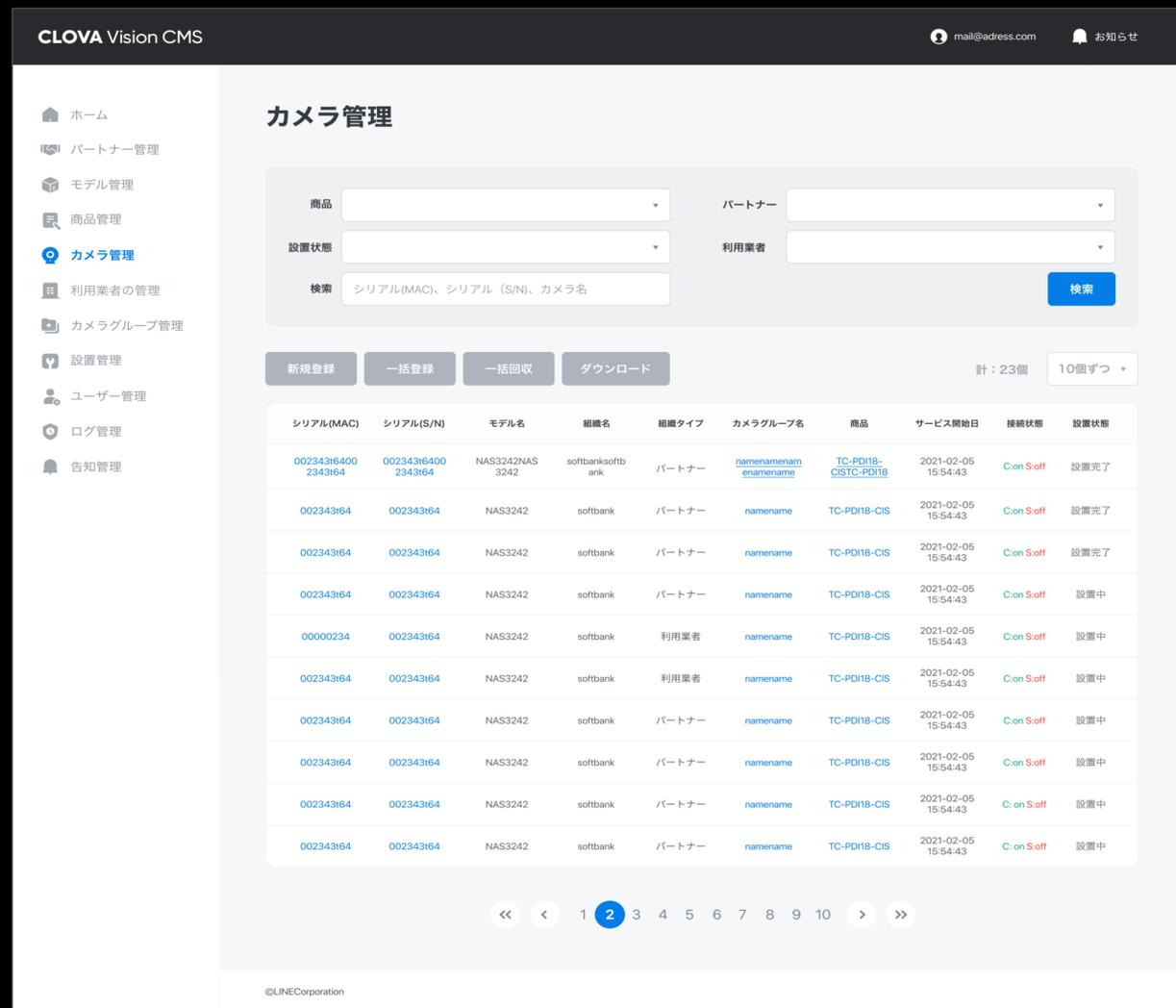
Vision AI



# 1.2 CMS & VMS

운영자의 효율적인 운영/관리를 돕는  
Camera Management System **CMS**

고객이 쉽고 빠르게 영상을 확인할 수 있는  
Video Management System **VMS**



- 카메라 관리/고객 어카운트 관리/설치 관리 등

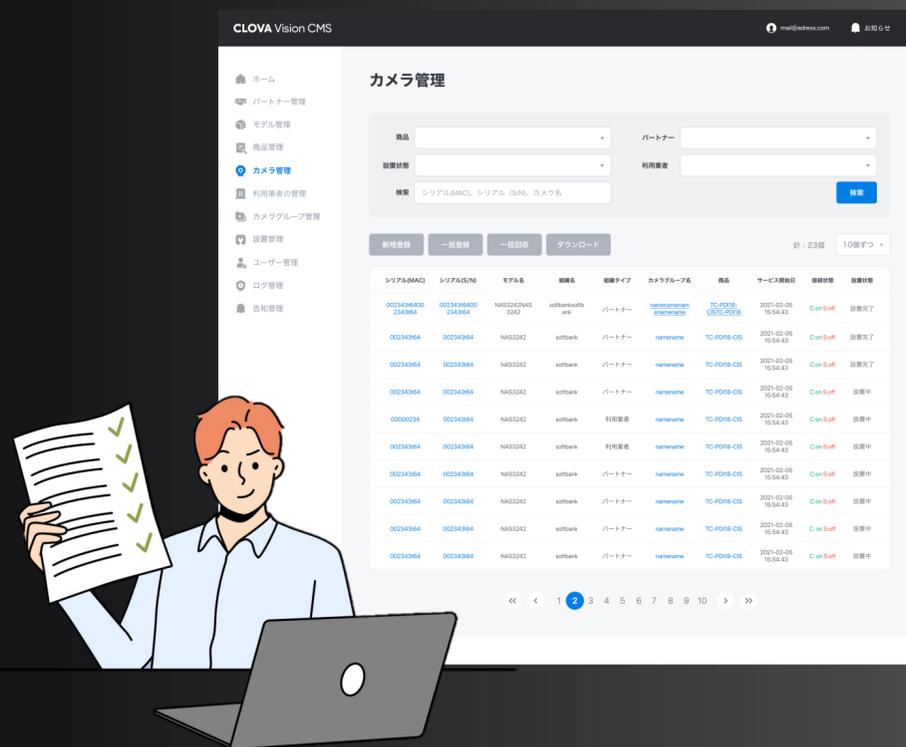
- 싱글뷰/멀티뷰 영상 모니터링, 영상 클립 생성, 권한설정 등

# 1.3 CLOVA Vision 설치 과정

Step1. 사전 준비(계약 후)

Step2. 설치

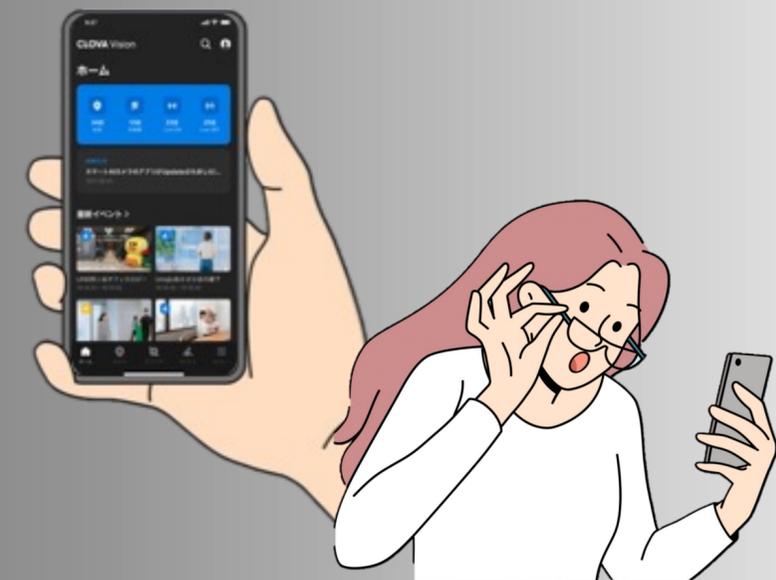
Step3. 서비스 이용



CMS를 통한 고객 정보 생성  
카메라 설치를 위한 사전 준비



고객 설치 희망일에 설치 공사 진행



서비스 개시일부터 VMS Web/App 이용

## 2. CLOVA Vision 개발 과정 되돌아보기

## 2.1 프로젝트 목표



## 2.2 개발 주요 대상 및 세부 과제

### [ 클라우드 카메라 ]

- H/W Vendor 선택 및 협업
- 빠른 개발 일정과 안정성
- 클라우드 연동 및 OTA 기능 개발
- 보안 강화 및 오류 복구

Device

#### 3. IP카메라 기반의 클라우드 카메라 개발

- 카메라 연결 및 재생 프로토콜
- 확장에 열려 있는 아키텍처
- 성능 효율화/극대화
- 서버 가용성 및 오류 복구

영상처리  
(저장 및 스트리밍)

#### 5. Cloud Camera 영상 데이터 처리하기

관리  
(연결 및 제어)

- 연결 및 제어 프로토콜 및 스펙
- 적은 비용 카메라 관리 성능
- 서버 확장성 및 가용성
- 무중단 운영

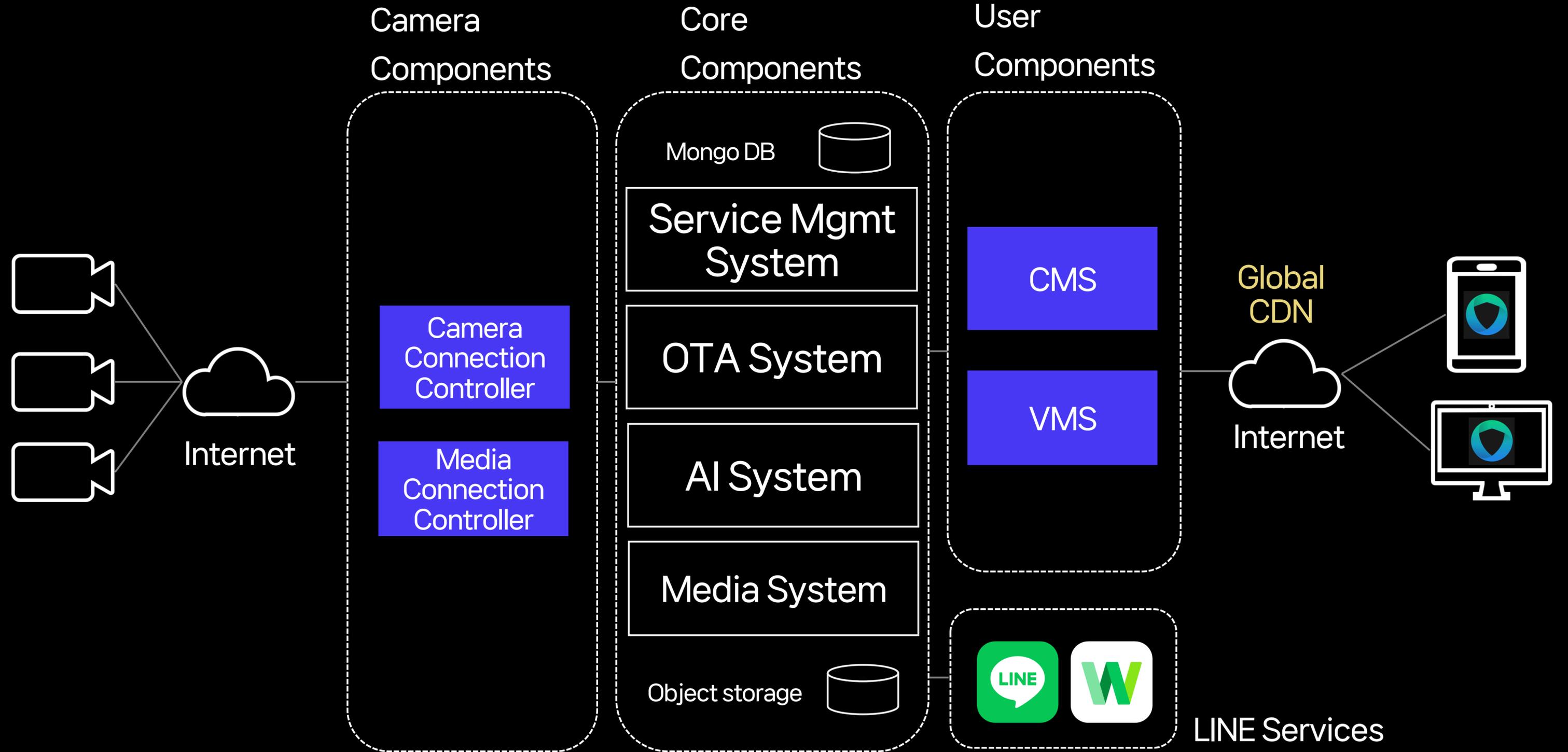
#### 4. 클라우드 카메라 연결 관리 및 제어

Vision AI  
서비스

- CLOVA Vision 미디어를 활용한 AI 추론
- 저비용/고부가 AI서비스 실현
- 확장 가능한 시스템 아키텍처
- 서버 가용성 및 오류 복구

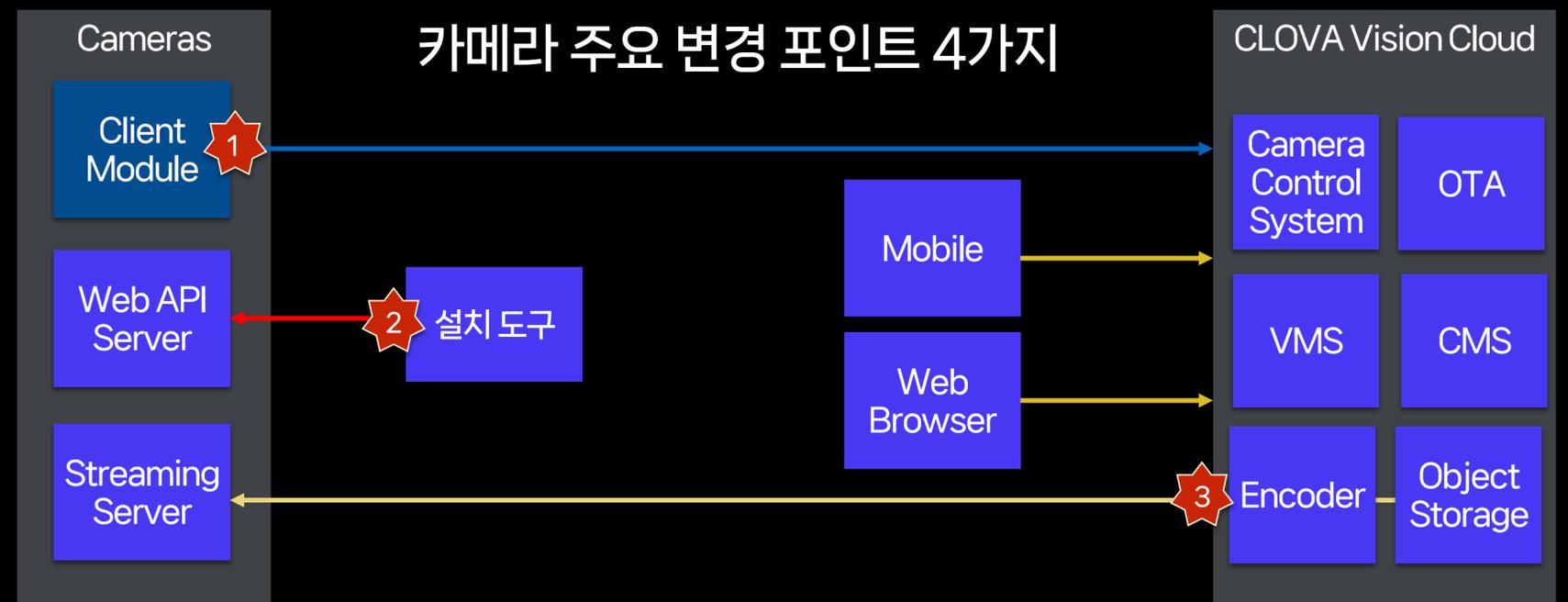
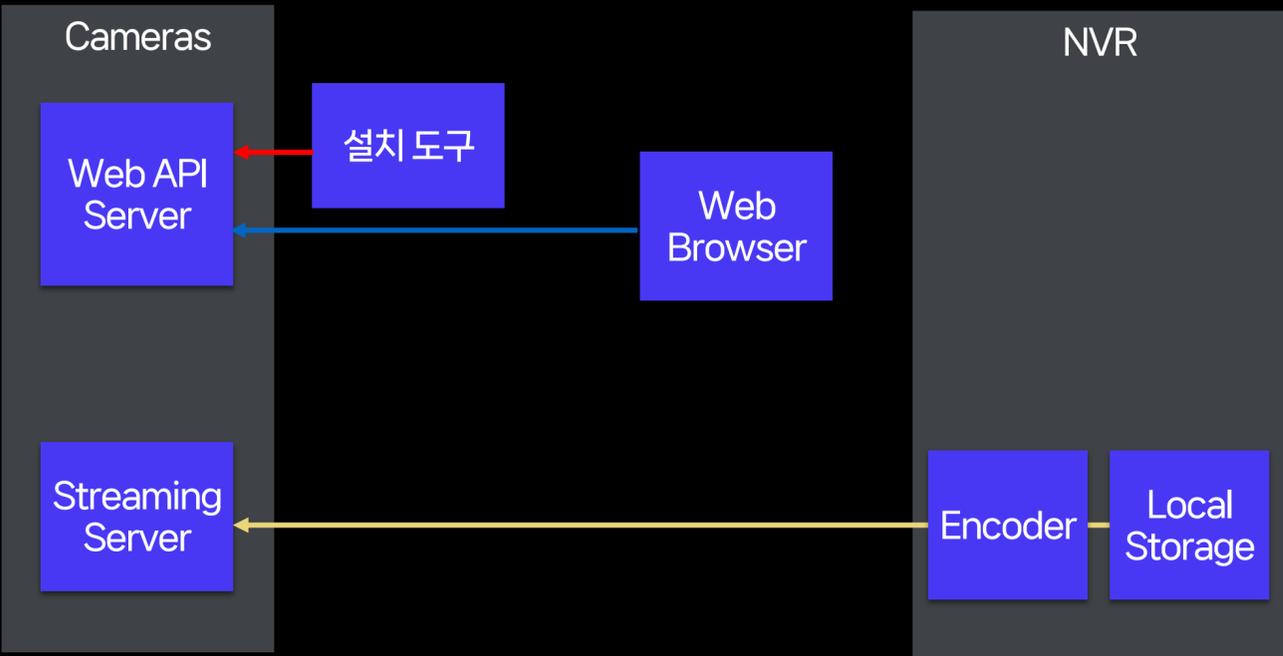
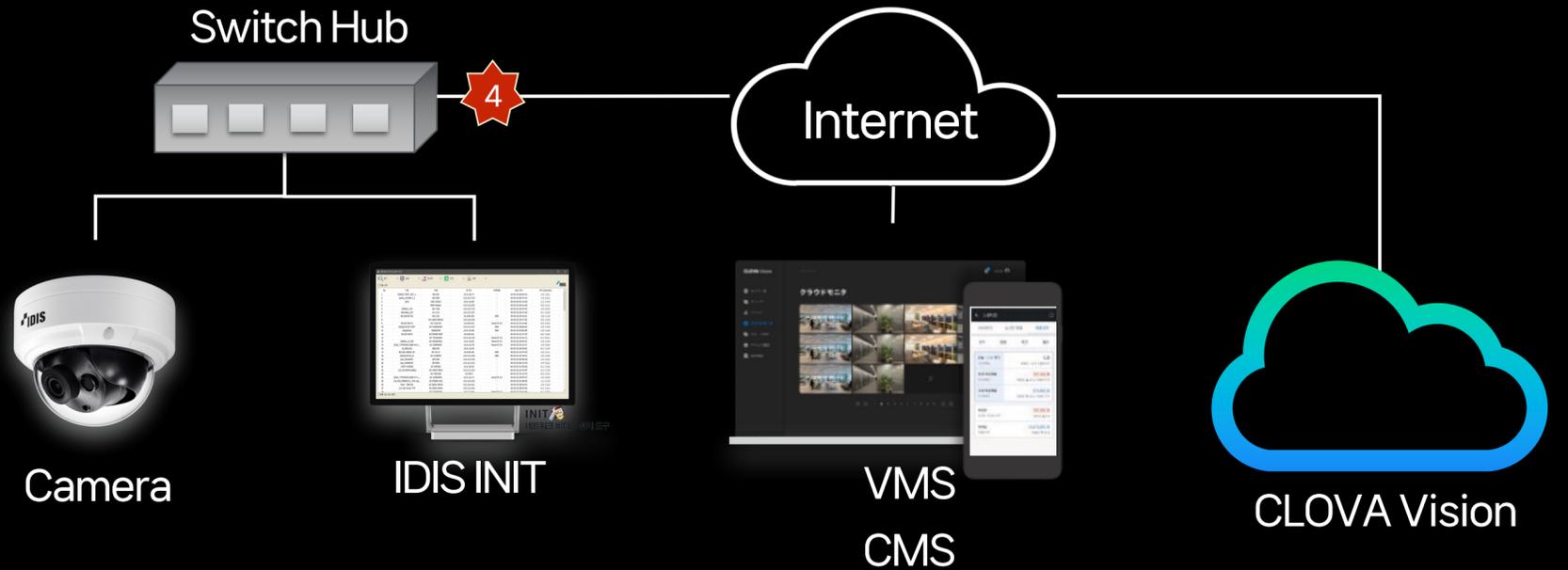
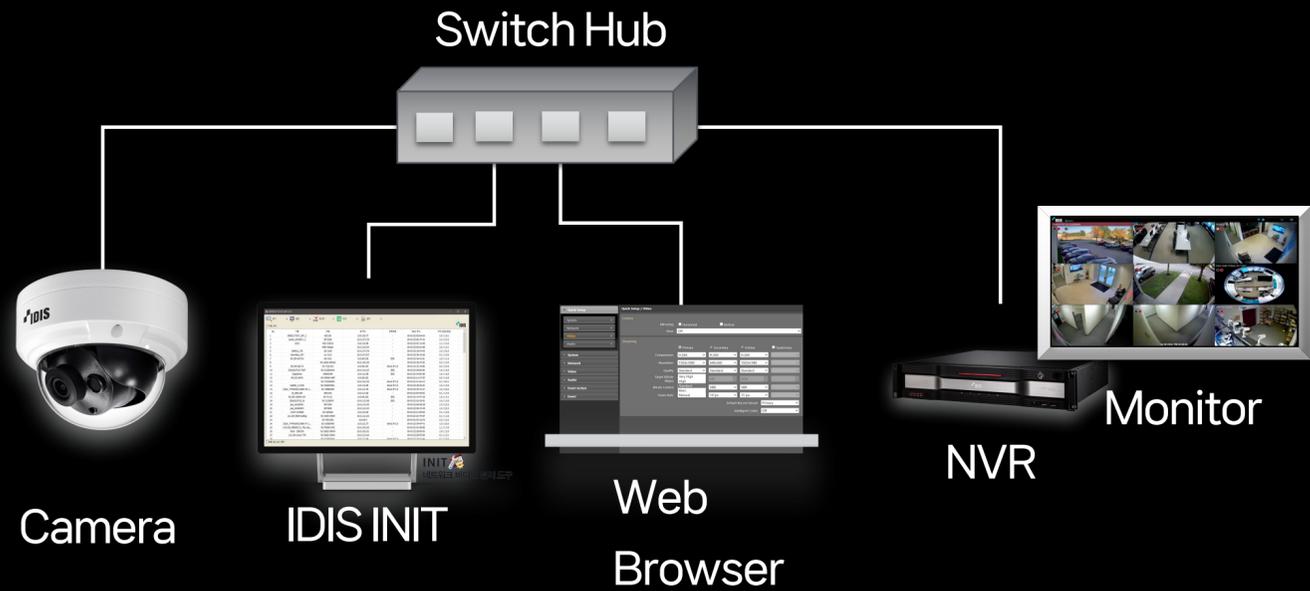
#### 6. CLOVA Vision과 AI의 만남

# 2.3 Overall Architecture



# 3. IP 카메라 기반의 클라우드 카메라 디바이스 개발

# 3.1 IP 네트워크 카메라 vs. 클라우드 시스템 구성



# 3.2 CLOVA Vision Client Module

## 인증

- CLOVA Access Token 획득 및 갱신 기능

## 카메라 설치

- CLOVA Vision Cloud 시스템에 카메라 설치 기능

## 카메라 컨트롤 프로토콜 및 규격

- VMS/CMS를 통해 카메라를 컨트롤 하고, 카메라의 상태 정보를 CLOVA Vision Cloud로 리포트 하는 기능

## 미디어 프로토콜 및 규격

- IP 카메라의 기본 기능인 RTSP/RTP Video Stream 서버에 접속할 수 있는 확장규격

## 에러 핸들링

- HTTP Ping-pong을 활용한 Health check 기능

## OTA

- CLOVA AI Speaker에서 사용하는 OTA 시스템 사용
- 자동 업데이트 기능 삭제 및 강제 업데이트 추가

### 용어 설명

RTSP (Real Time Streaming Protocol) : 실시간 스트리밍 프로토콜

RTP (Real-time Transport Protocol) : 실시간 전송프로토콜

OTA (Over-The-Air) 시스템 : SW 자동 업데이트 시스템

# 3.3 Command와 Context

## Command

- 카메라에 전달하는 명령어
- 구분을 위해 Command ID 정의
- 타입은 Set (mode 0) 와 Get (mode 1)
- 확정성을 위해 Key/Value 형태의 JSON Format
- 여러 Command를 한번에 전달하기 위해 Array를 사용해서 유연성 제공
- Action으로 명령어 구분
- HTTP2 ServerPush 통해 전달

## Context

- 카메라의 상태 정보
- 카메라의 상태 정보가 변경 될 때마다 CLOVA Vision에게 Report 한다

```
{
  "actions" : [
    {
      "parameters": [
        {
          "key": "action",
          "value": "eventMotion"
        },
        {
          "key": "mode",
          "value": "1"
        }
      ]
    },
    {
      "parameters": [
        {
          "key": "action",
          "value": "videoPrivacy"
        },
        {
          "key": "mode",
          "value": "1"
        }
      ]
    }
  ]
}
```

Request Get

```
{
  "id": "ef9727f1-d77d-4c8a-bf02-69bd1e07c1c7",
  "ruid": "I01S0f4de5b0139b4bc2a38b1a24755ba4a35FA242",
  "results": [
    {
      "parameters": [
        {
          "key": "returnCode",
          "value": "0"
        },
        {
          "key": "useMotion",
          "value": "on"
        },
        {
          "key": "daySensitivity",
          "value": "3"
        },
        {
          "key": "nightSensitivity",
          "value": "3"
        },
        {
          "key": "motionZone",
          "value": "0022_0015_000000_000000_000000"
        },
        {
          "key": "ignoreInterval",
          "value": "2"
        },
        {
          "key": "daytimeStart",
          "value": "09:00"
        },
        {
          "key": "daytimeEnd",
          "value": "18:00"
        },
        {
          "key": "numBlockWidth",
          "value": "22"
        },
        {
          "key": "numBlockHeight",
          "value": "15"
        }
      ]
    },
    {
      "parameters": [
        {
          "key": "returnCode",
          "value": "0"
        },
        {
          "key": "usePrivacy",
          "value": "on"
        }
      ]
    }
  ]
}
```

Response

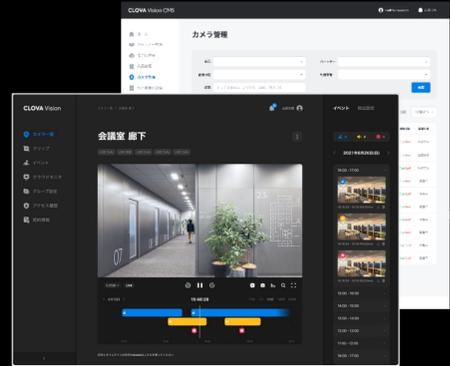
```
{
  "mediaStreaming": {
    "videos": [
      {
        "streamId": 1,
        "enable": true,
        "type": "recording",
        "codec": "h264",
        "resolution": {
          "horizontal": 1280,
          "vertical": 720
        },
        "quality": "standard",
        "qualityStream": 512,
        "bitrateControl": "vbr",
        "frameRate": 15,
        "activity": "started",
        "connection": {
          "status": "connected",
          "code": 1,
          "reason": "ok",
          "dateTime": "2023-02-05T22:26:34.357+09:00",
          "channelType": "prod"
        }
      }
    ],
    "audio": {
      "codec": "g711_ulaw",
      "input": {
        "enable": true,
        "volume": 8
      },
      "output": {
        "enable": false,
        "volume": 8
      }
    }
  },
  "device": {
    "system": {
      "language": "jp",
      "model": "LDC-02",
      "note": "",
      "hardwareVersion": "2.0",
      "softwareVersion": "3.7.0",
      "name": "",
      "ruid": "ABCERRSSEADF50643529809dbf8#$ASD87",
      "mac": "00:03:22:66:F2:87",
      "date": "2023-02-08",
      "time": "15:29:41",
      "timeZone": "+09:00"
    },
    "network": {
      "type": "dhcp",
      "ip": "192.168.1.xxx",
      "gateway": "192.168.1.1",
      "subnetMask": "255.255.255.0",
      "dns": {

```

Context

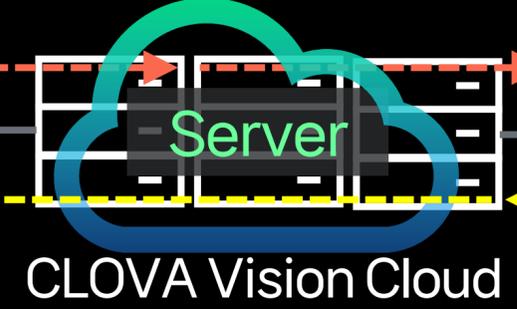
# 3.4 서버와의 통신

설정 변경,  
펌웨어 업데이트 등



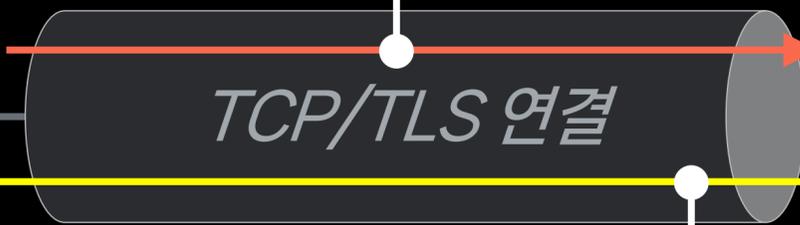
VMS / CMS

카메라에 요청할  
Command 구성



CLOVA Vision Cloud

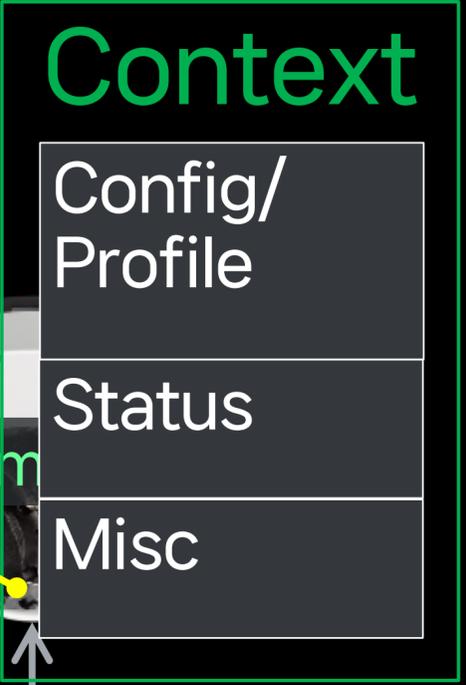
Server Push  
(PUSH\_PROMISE)



TCP/TLS 연결

Persistent Connection  
HTTP2

카메라 상태 정보



Context

Config/  
Profile

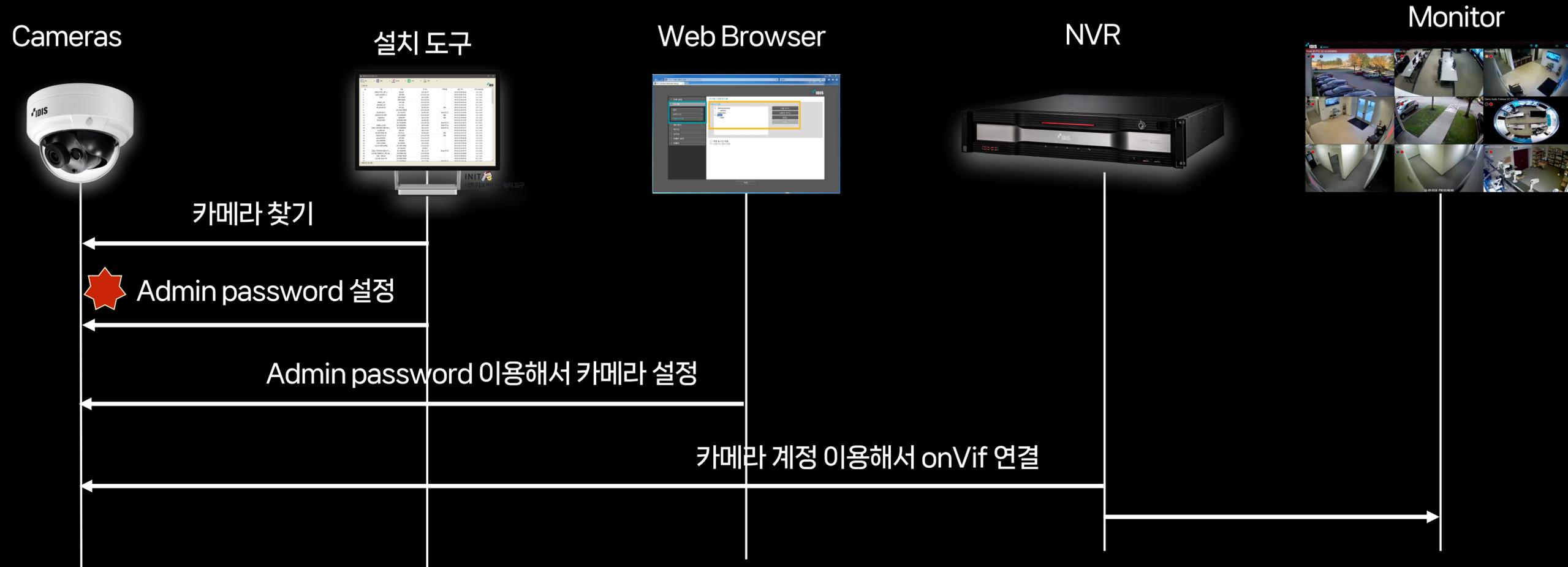
Status

Misc

응답 (요청, 조회 등),  
보고 (이벤트, 알람 등)

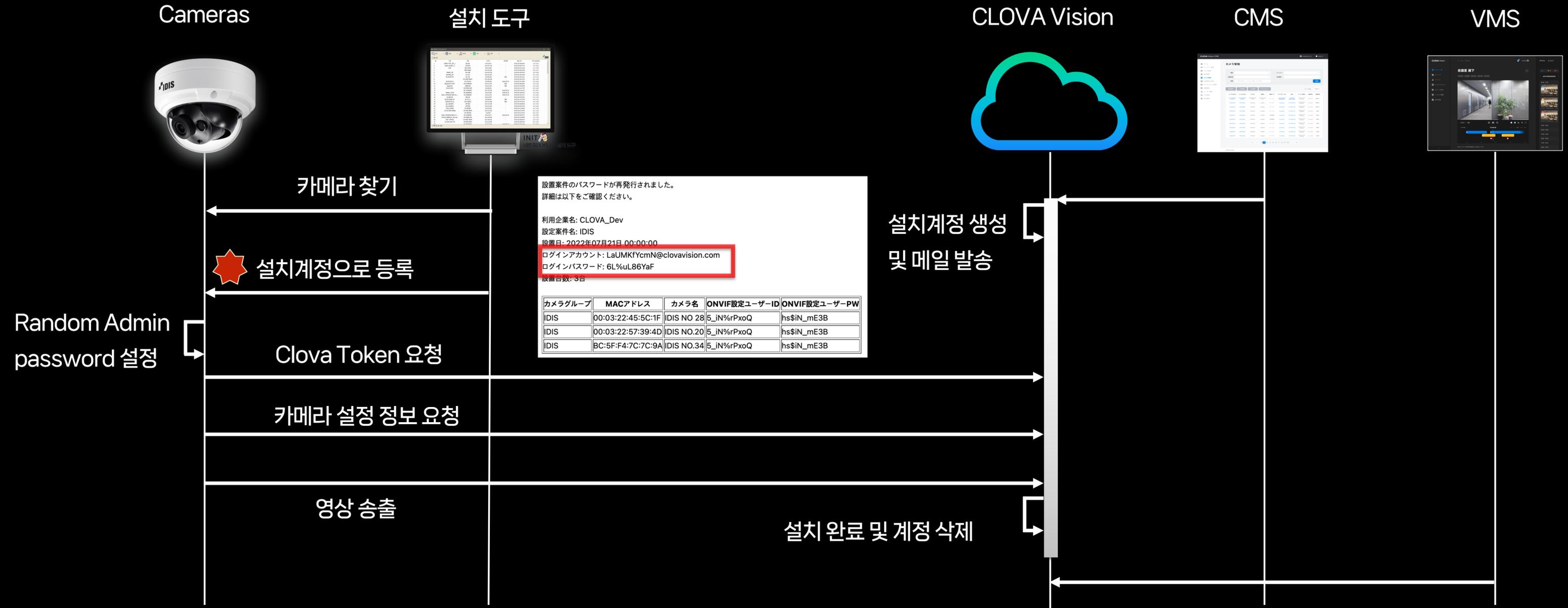
Command 와 Context 로 구성된 JSON 메시지 통신

# 3.5 IP Camera 설치 하기



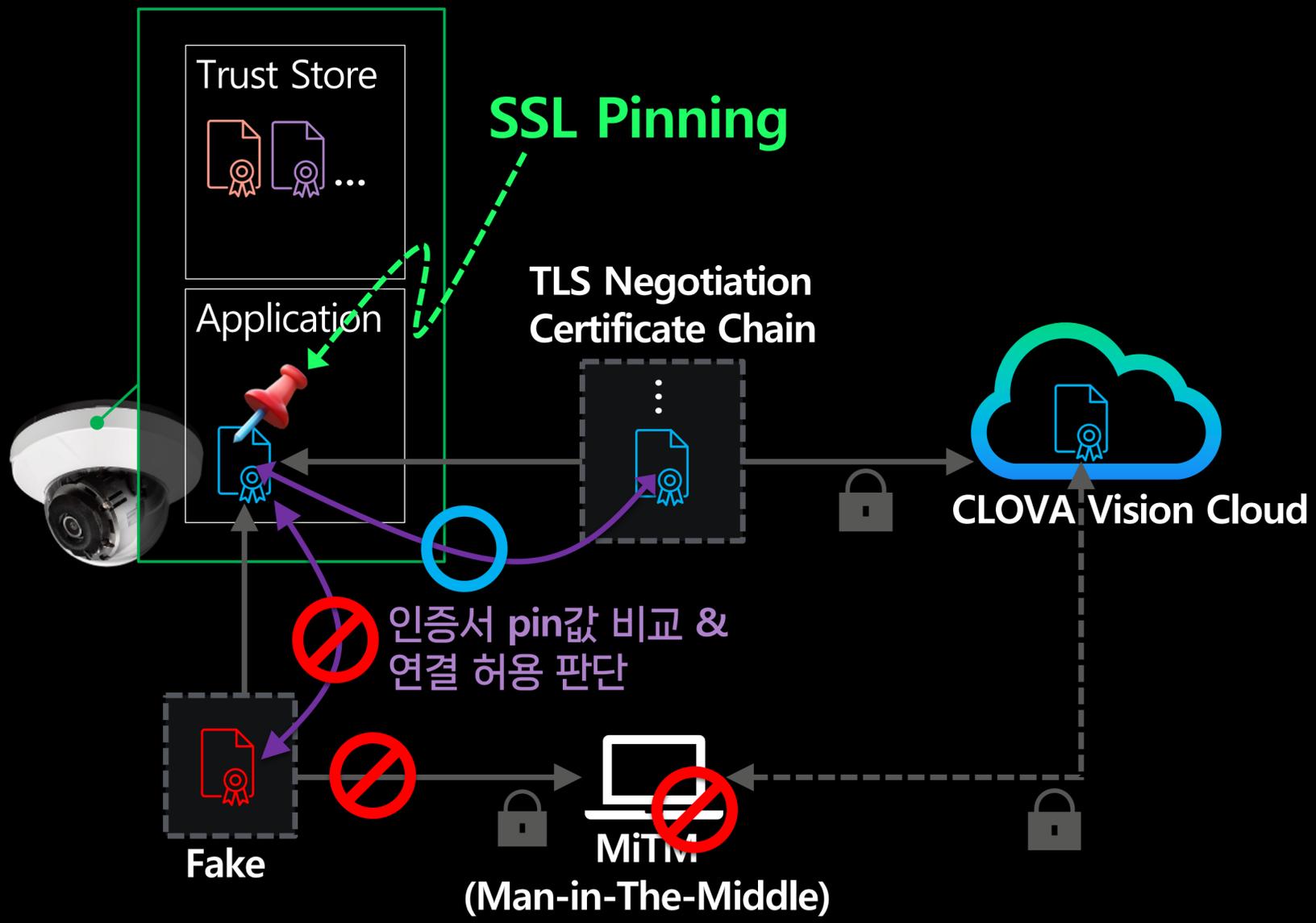
카메라 연동/설정 시 Admin password 가 필요함  
유출될 경우? 유실될 경우?

# 3.6 CLOVA Vision 설치 하기

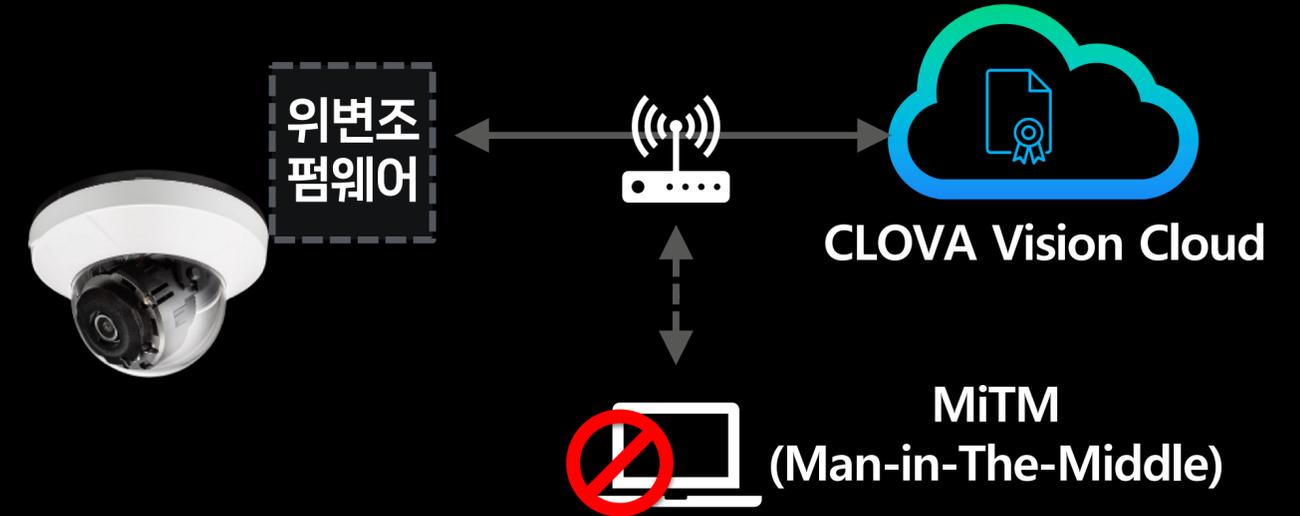


# 3.7 중간자공격(MITM) 대응

## MiTM 공격/취약점 대응



## 펌웨어 위변조 방어



## Brute force 공격 대응



## 3.8 정리

- 기존 IP카메라의 기능을 최대한 재사용해서 안정성을 확보해야 한다.
- 카메라에서 가장 중요한 기능은 *Cloud 연동하기*와 *OTA*이다.
- 고객의 다양한 네트워크 환경을 고려해서 보안기능을 강화해야 한다. 이때 클라이언트의 중간자 공격 대응은 필수이다.
- 확장성을 위한, 3<sup>rd</sup> party Vendor용 Camera SDK 준비중.

# 4. Cloud Camera 연결 관리 및 제어

# 4.1 카메라 컨트롤 시스템 개요

## 카메라에 커맨드를 전송

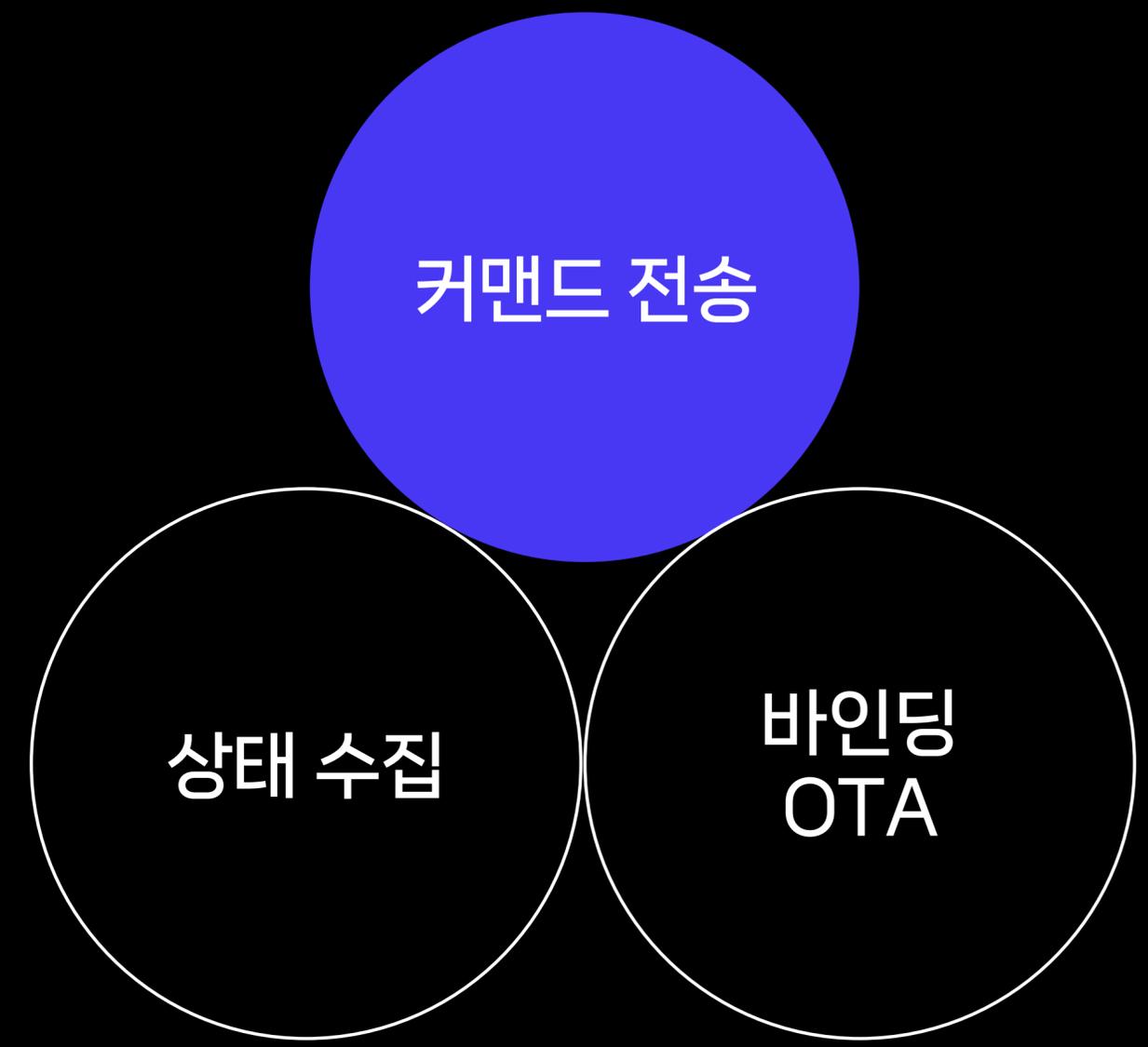
- GET
- SET

## 카메라의 상태 정보를 수집

- 컨텍스트 업데이트

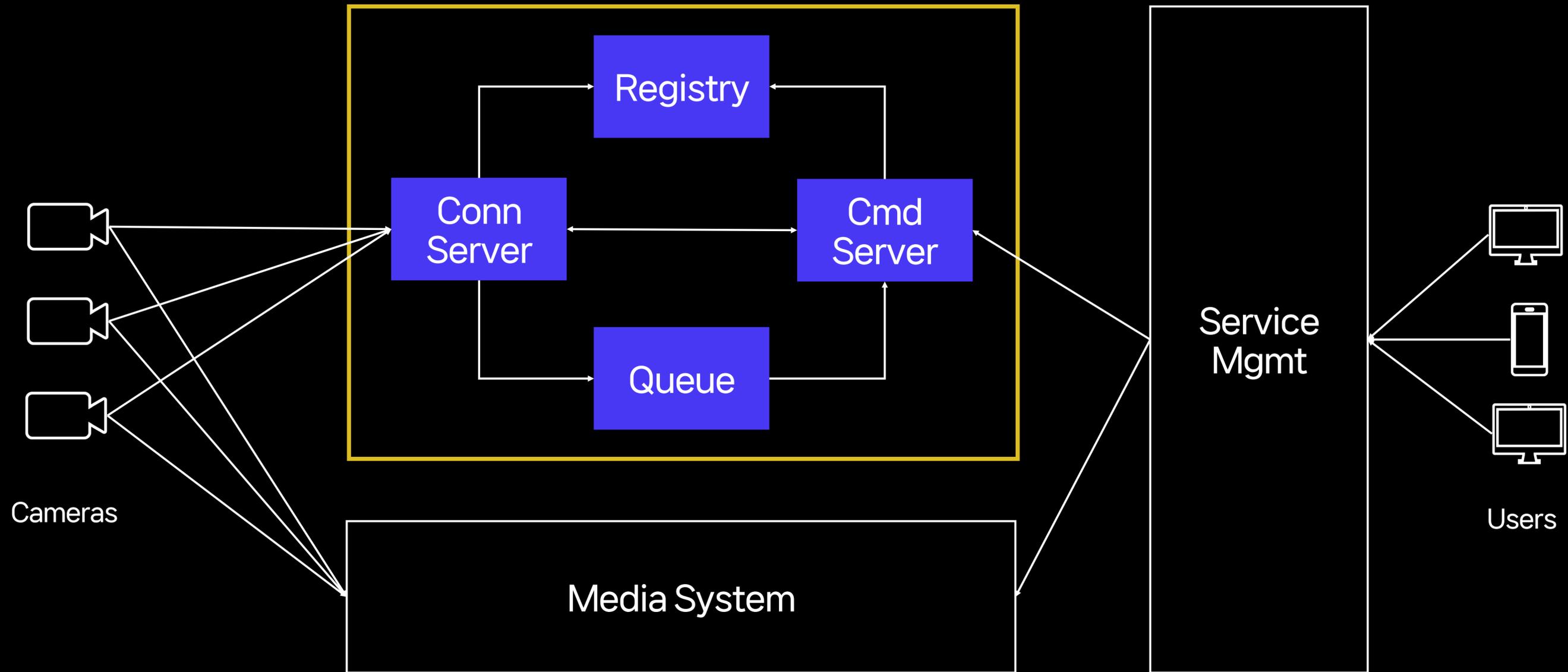
## 기타

- 카메라 바인딩
- 카메라 펌웨어 OTA 업데이트



# 4.2 카메라 컨트롤 시스템 구조

## Camera Control System

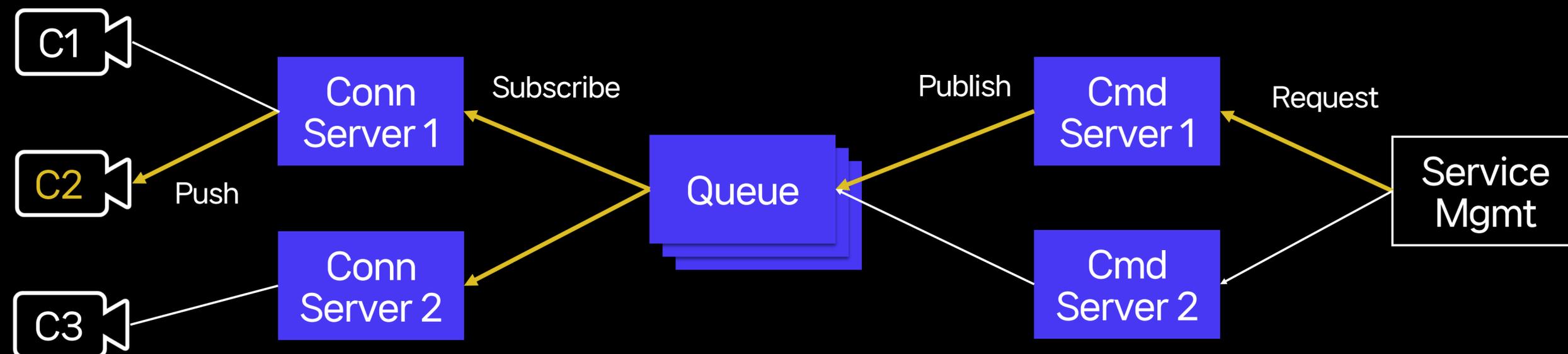


# 4.3 커맨드 전송 (1/2)

## 커맨드-커넥션 서버 간 커맨드 전달 방법

### 1. 메시지 큐 기반 커맨드 Pub/Sub

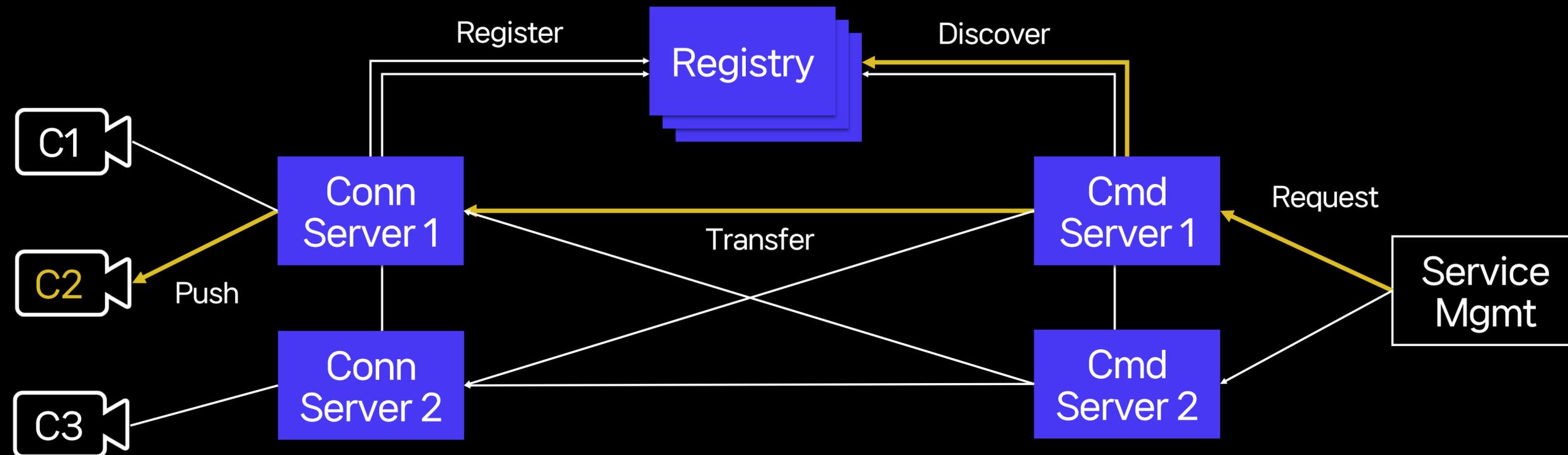
- 커맨드는 메시지 큐를 통해 모든 커넥션 서버에 전달
- 커넥션 서버는 모든 커맨드 메시지를 받지만, 자신에게 연결된 카메라에 대한 커맨드만 처리



# 4.3 커맨드 전송 (2/2)

## 2. 레지스트리 등록

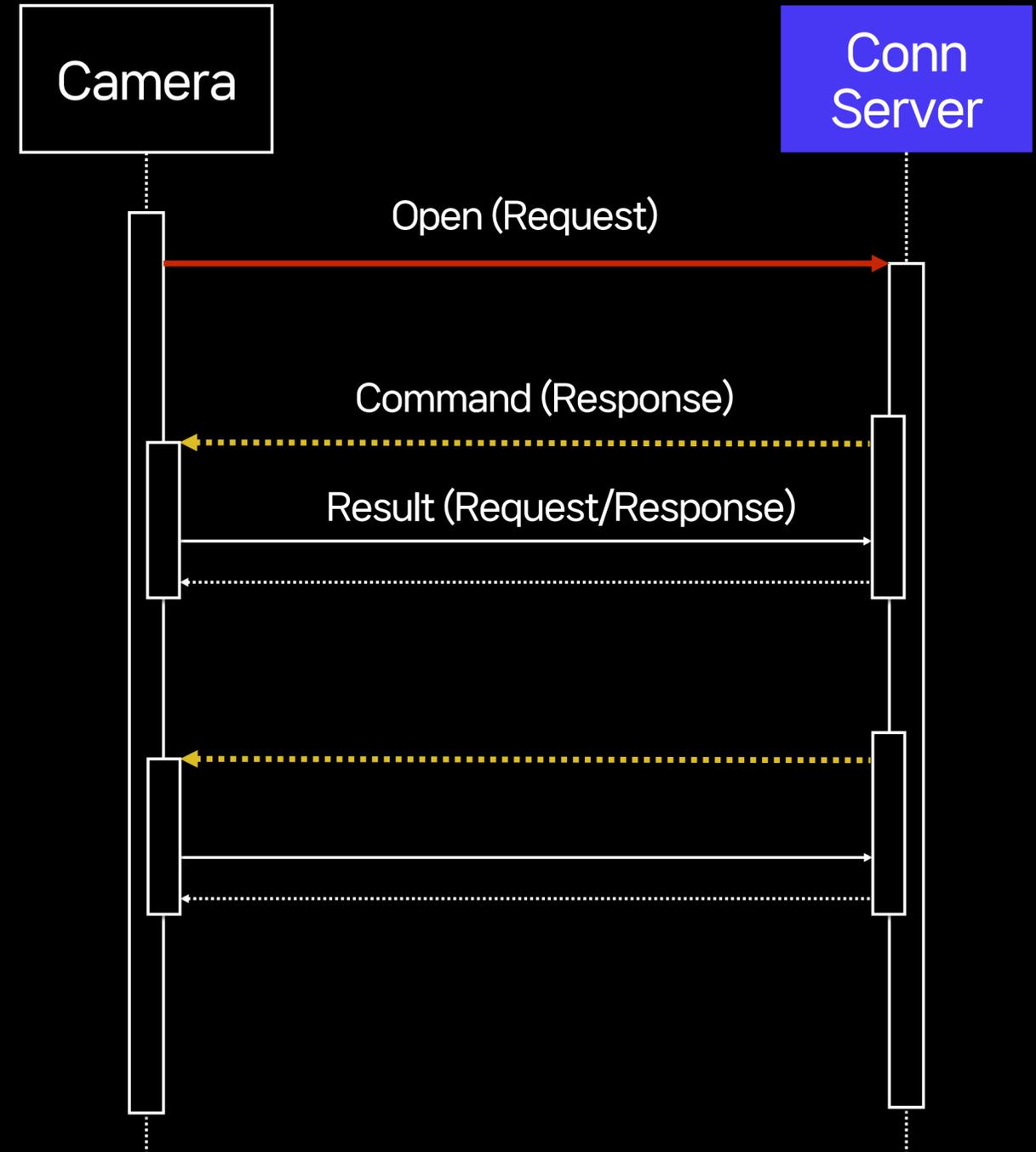
- 커넥션 서버는 자신에게 연결된 카메라를 레지스트리에 등록
- 커맨드 서버는 레지스트리에서 대상 카메라가 연결된 카메라 커넥션 서버를 찾아 커맨드를 전송



# 4.4 커맨드 푸시

## 커넥션 서버에서 카메라로 커맨드 푸시

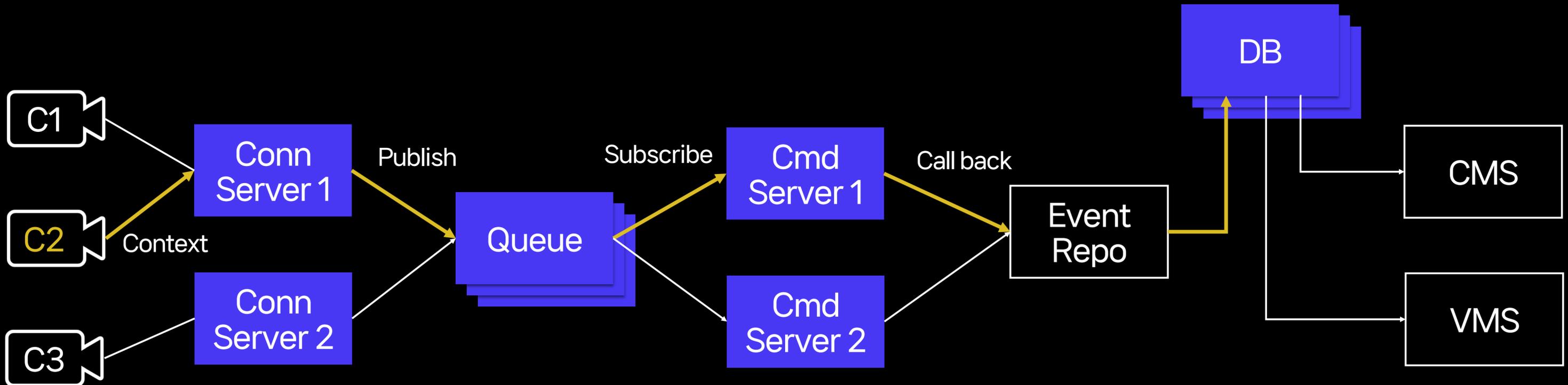
- 카메라는 커넥션 서버에 접속하여 푸시 스트림을 열어 둠 (카메라가 연결된 상태)
- 커넥션 서버는 열린 스트림을 통해 커맨드를 푸시 (PUSH\_PROMISE 프레임)
- 카메라는 받은 커맨드를 처리하고 그 결과를 서버에 요청으로 전달



# 4.5 컨텍스트 수집

## 카메라 컨텍스트는 메시지 큐를 통해 이벤트 리포지터리로 전송

- 이벤트 리포지터리는 컨텍스트의 변화를 감지하여 DB를 업데이트
- CMS/VMS 사용자는 카메라의 상태를 확인할 수 있음



## 4.6 서버 확장성

### 커넥션 서버 수평 확장성

- 커넥션 서버는 카메라 연결에 비례하여 수평 확장 필요
- 수평 확장에는 카메라 연결 분산(재연결)이 필요
- 카메라 연결 수는 사업적으로 예측이 가능 → 수동 스케일링 가능

### 커넥션 서버의 처리 용량 산정

- 가상의 카메라 클라이언트를 커넥션 서버에 연결하고 커맨드 전송 부하
- 가상의 카메라 클라이언트의 수를 증가시켜가면서 커넥션 서버 부하의 변화 확인

## 4.7 서버 가용성

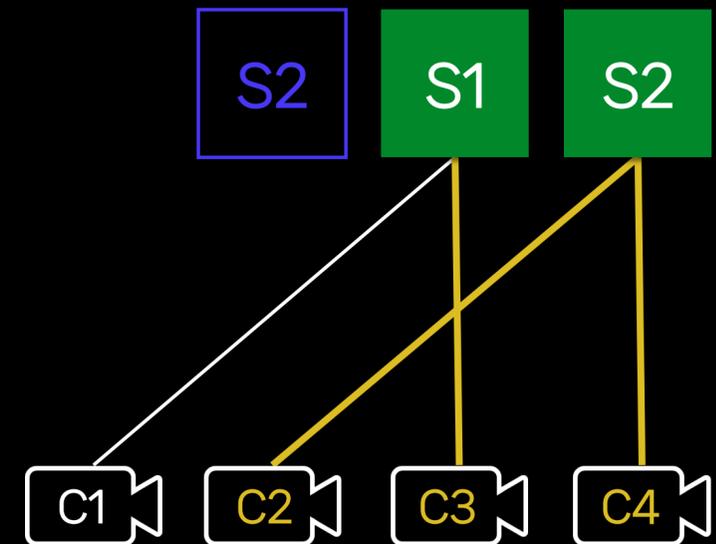
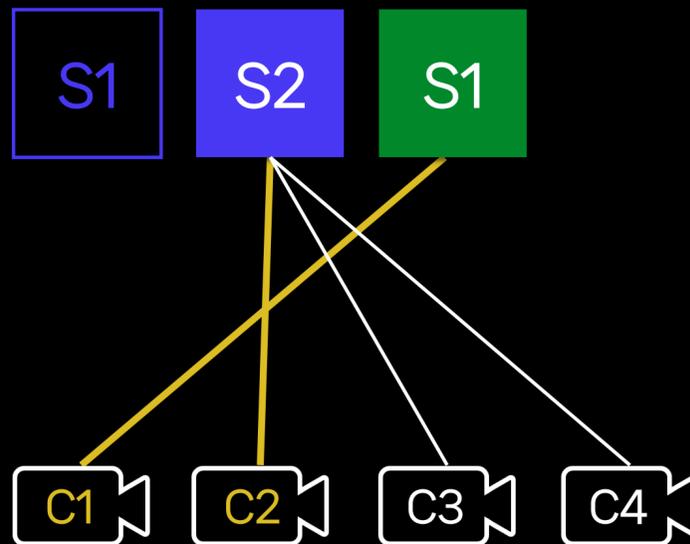
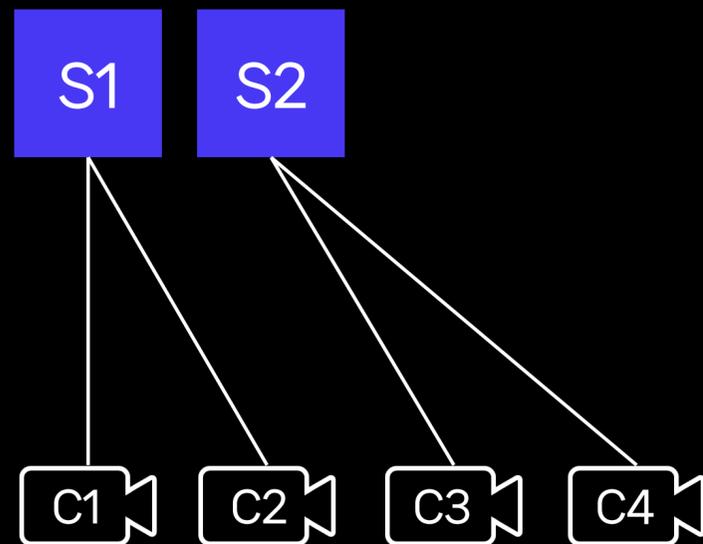
### 카메라 컨트롤 무중단 서비스

- 카메라는 주기적인 서버 헬스체크를 수행하며 서버 연결 단절 시 재연결을 시도
- 재연결을 피할 수 없는 상황
  - 노드의 물리적인 장애
  - 서버 업데이트 (배포)
- 재연결 동안 일시적인 단절이 발생하지만 치명적인 문제는 아님
- 영상은 미디어 시스템으로의 별도 연결로 전송 (버퍼링을 통한 영상 유실 최소화)

# 4.8 중단 최소화 배포 전략 (1/2)

## 롤링 업데이트

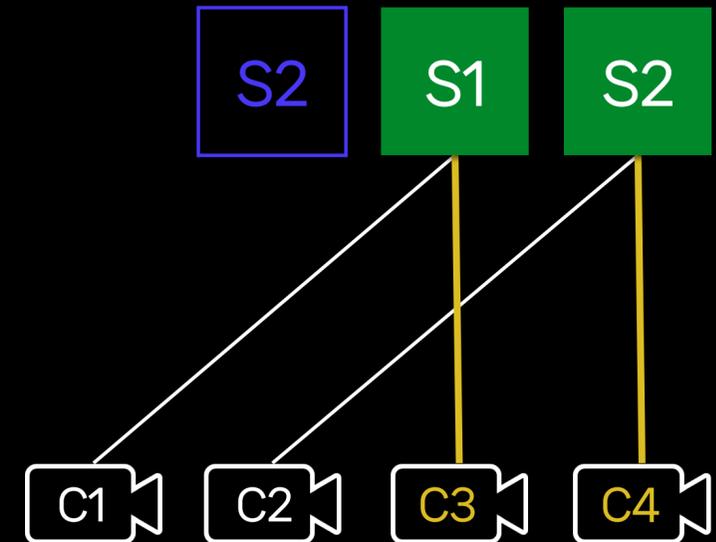
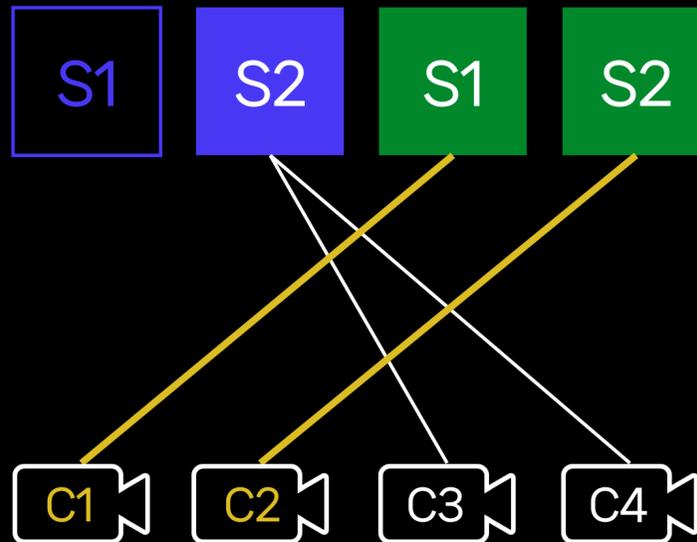
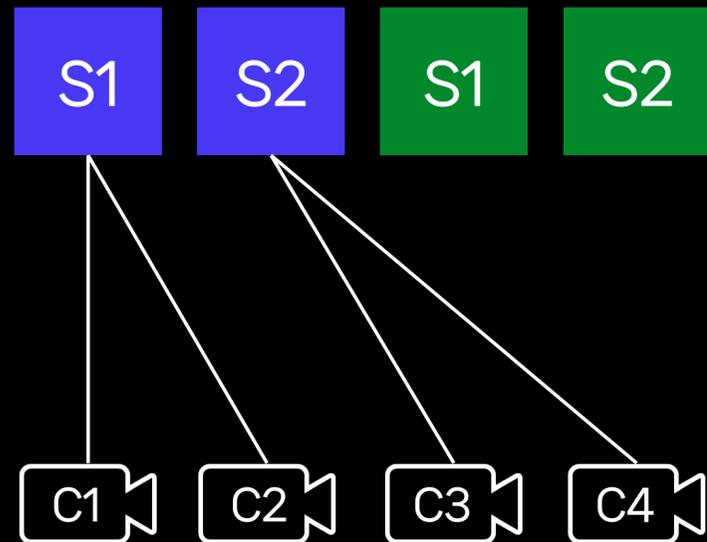
- 새 버전의 인스턴스를 점진적으로 늘려나감
- 불필요한 재연결 발생



# 4.8 중단 최소화 배포 전략 (2/2)

## 블루-그린 배포

- 새 버전의 인스턴스를 한꺼번에 생성해둠
- 단, 기존 서버는 시차를 두고 섀다운



## 4.9 정리

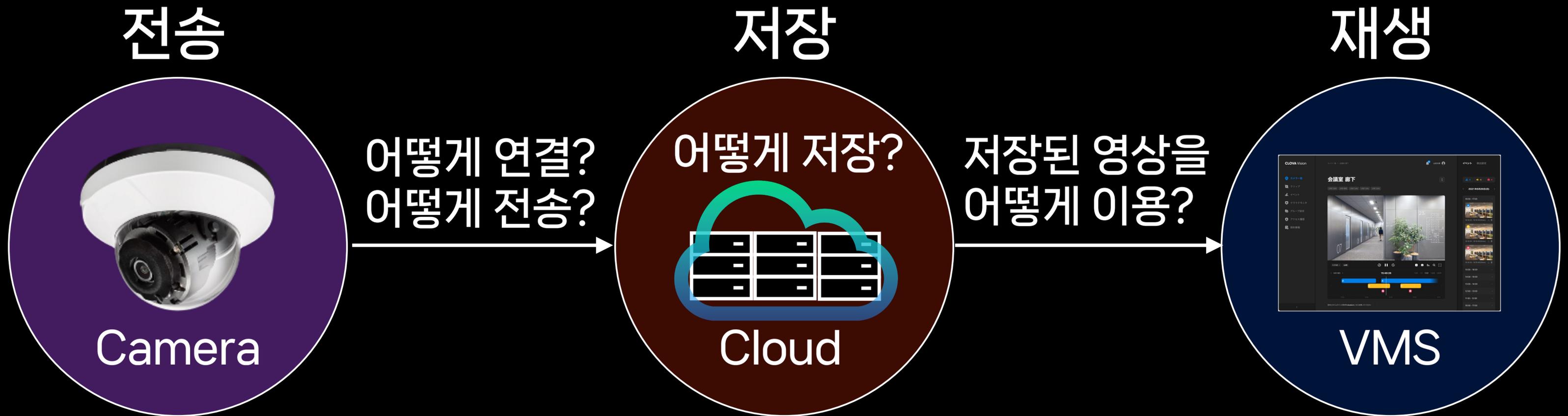
- 카메라 컨트롤 시스템은 커맨드와 컨텍스트를 효과적으로 전송하고 처리하는 역할
- 클라우드 서비스에서 서버의 확장성과고가용성은 비즈니스 니즈를 충족하도록 고려가 필요

# 5. Cloud Camera

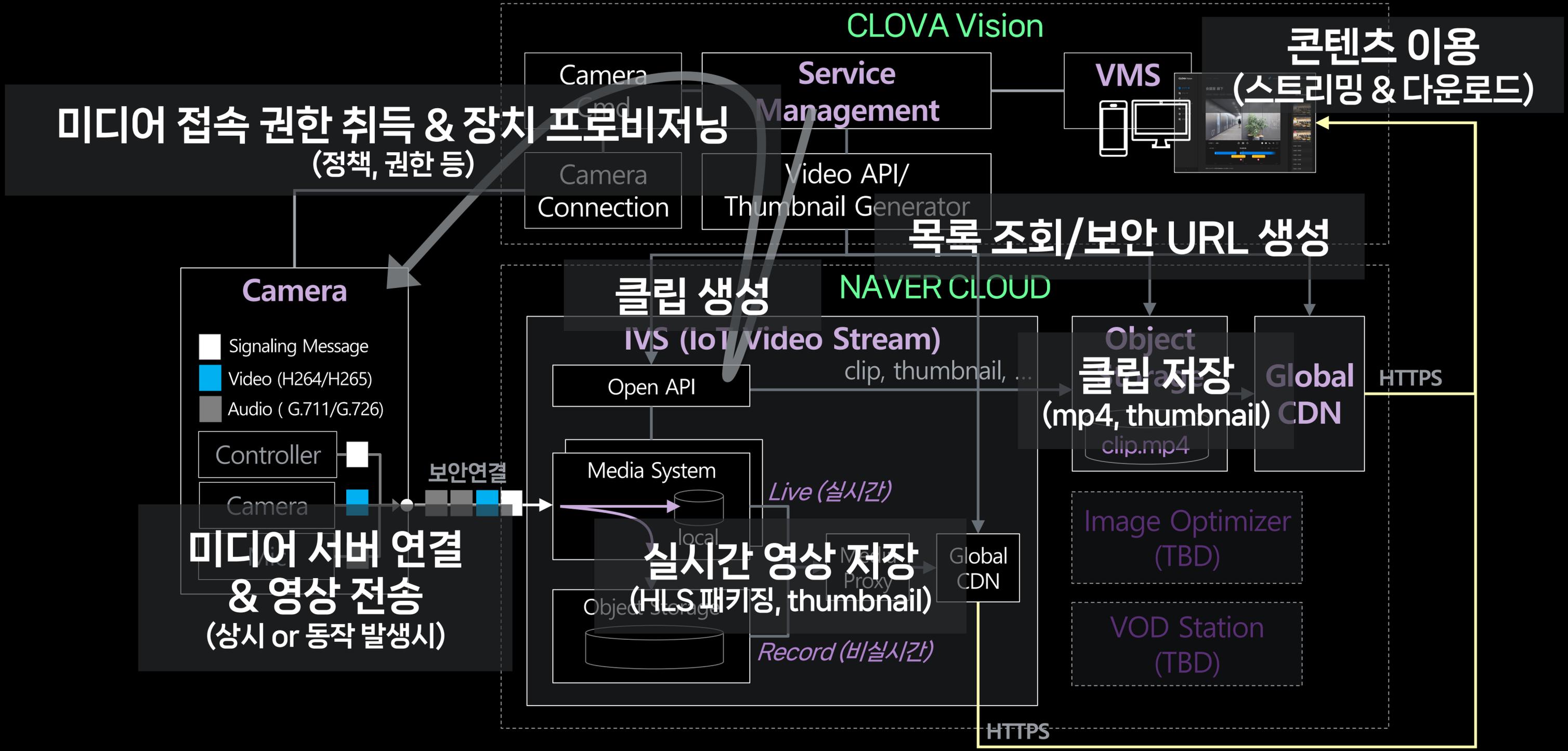
## 영상 데이터 처리

# 5.1 개요

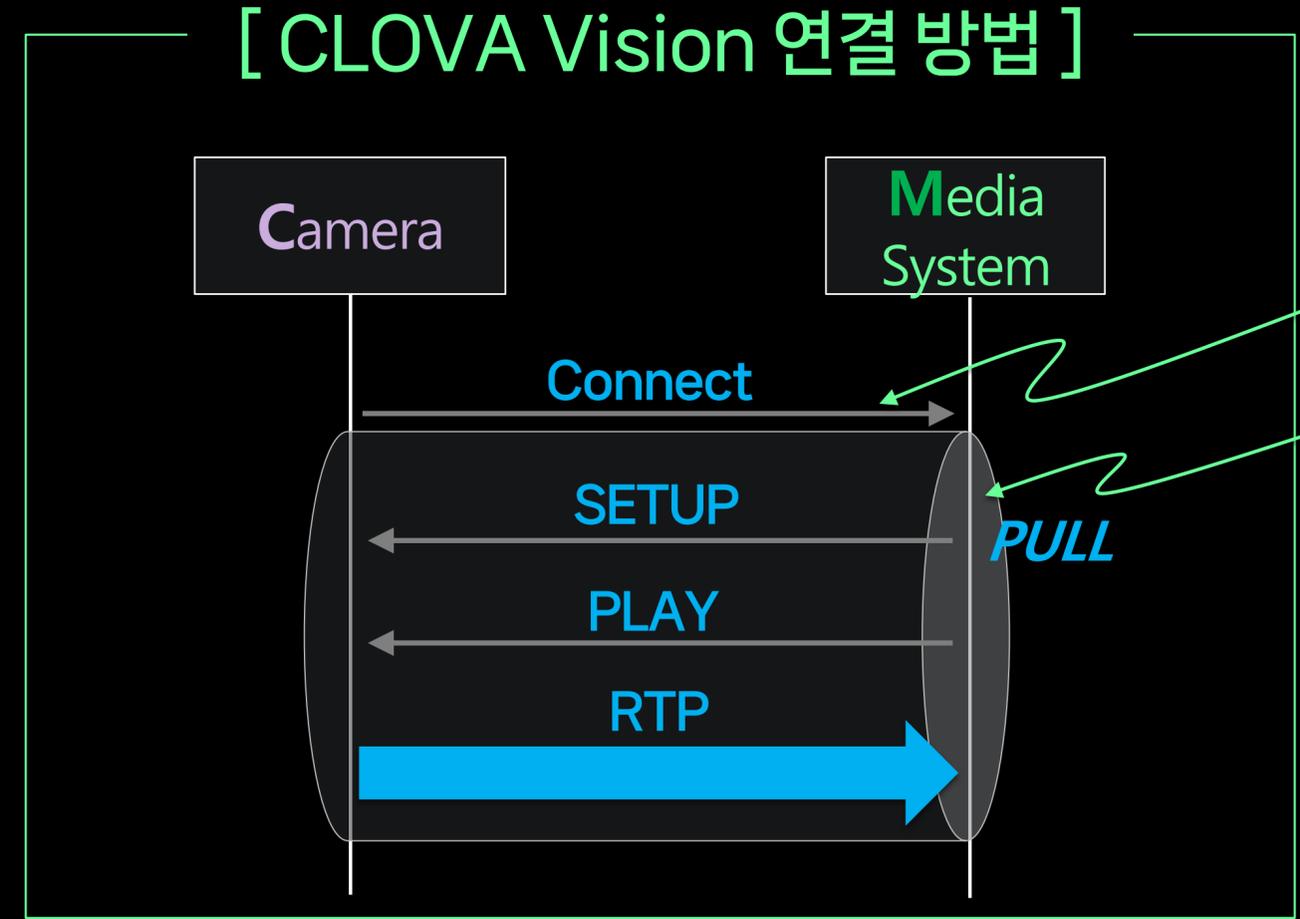
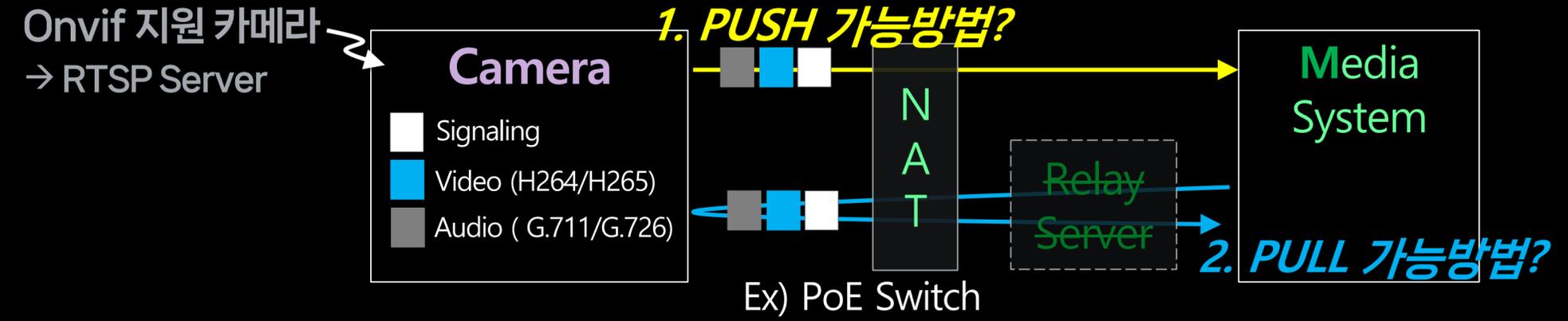
## 영상 데이터 처리 범위



# 5.2 시스템 아키텍처



# 5.3 Camera 영상을 Cloud에 연결하는 방법은?



C → M : TCP/TLS 연결

C ← M : PLAY 요청

참고

RTSP 1.0 : [RFC2326 \(April 1998\)](#)  
 Appendix A: RTSP Protocol State Machines  
 A.1,A.2 Client/Server State Machine

[ONVIF™ Streaming Specification](#)  
 5.2 Media Control protocol  
 5.2.2 RTSP : [ Table 2: RTSP methods ]

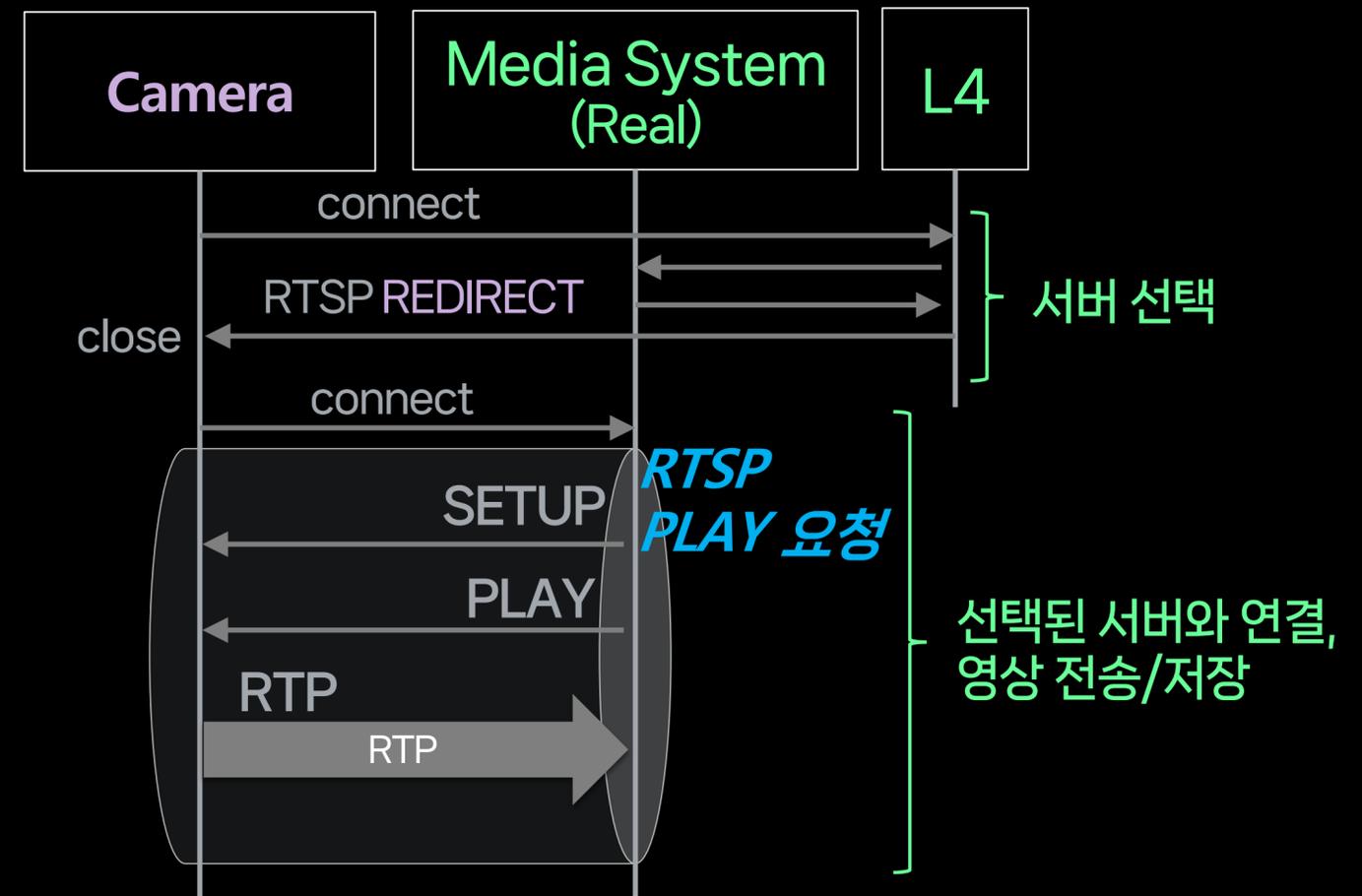
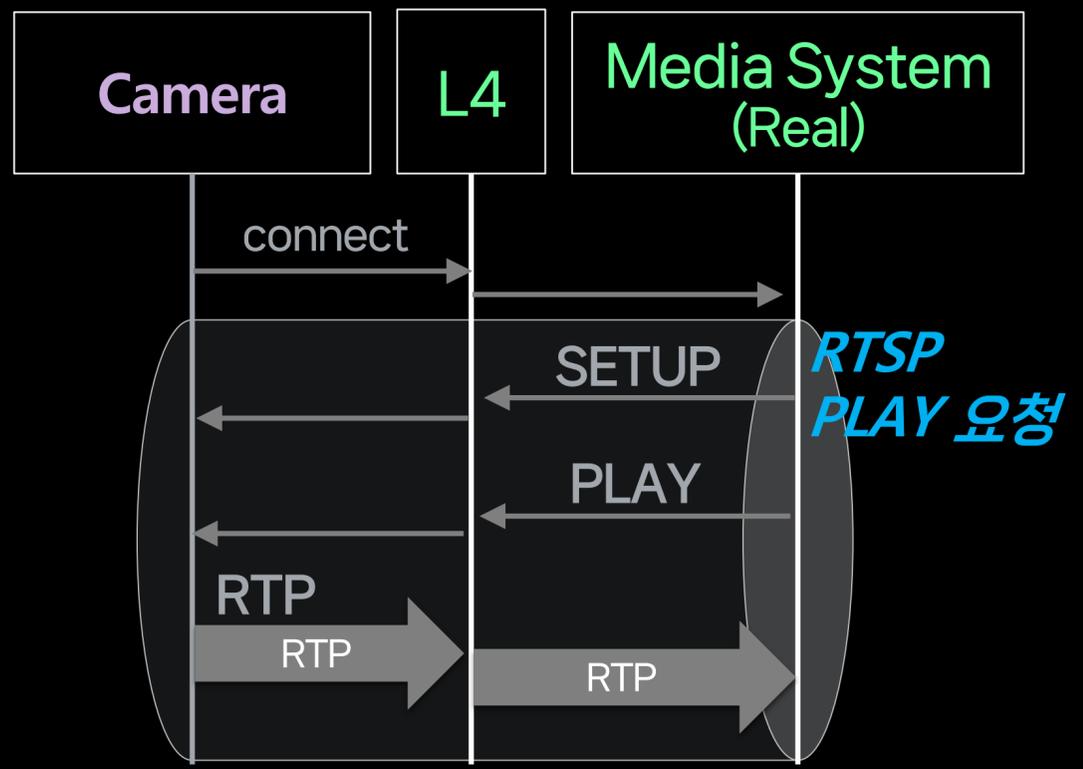
# 5.4 입력 트래픽의 효율적인 분산처리 전략은?

초기

현재

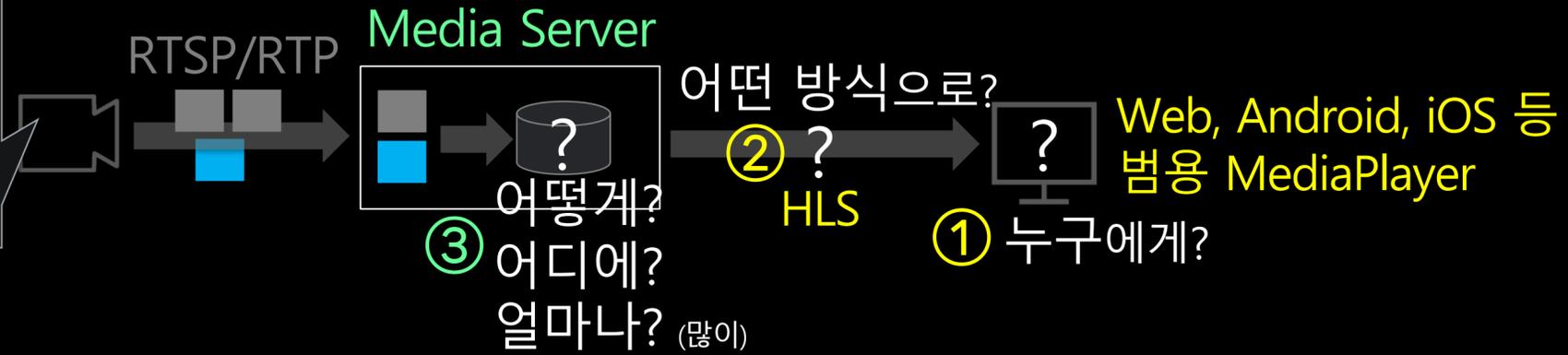
L4 VIP로 모든 패킷이 경유됨  
(영상 및 모든 트래픽이 L4를 경유함)

L4 VIP는 리얼 서버 연결 용도로만 사용  
(영상 트래픽(RTP)의 L4 네트워크 경로 제거)



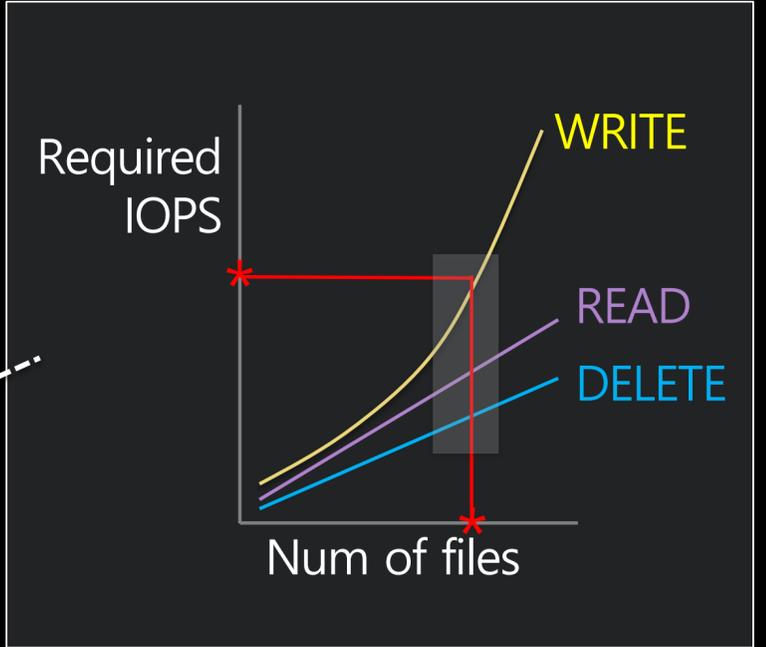
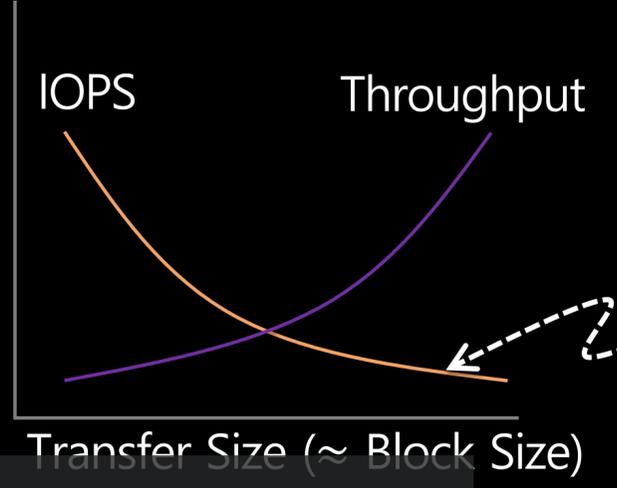
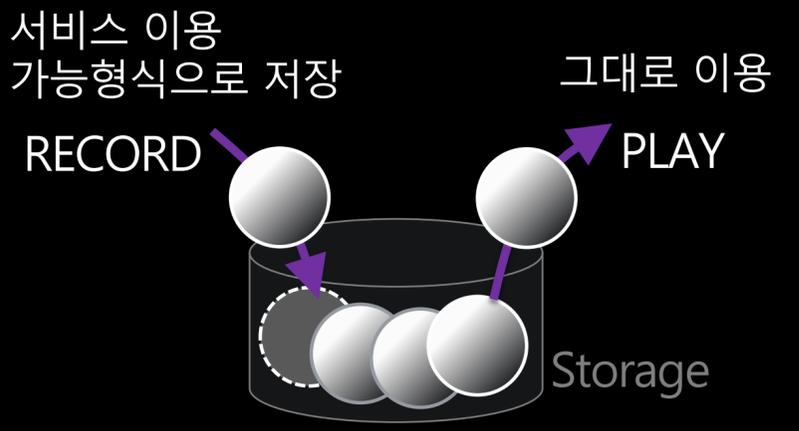
# 5.5 영상 저장 & 스트리밍 방식 (1/2)

Bitrate : 1.5Mbps  
GoP : 4sec  
Codec :  
- video : H.264/H.265  
- audio : G.711/G.726



어떻게? 어디에? ('성능'과 '비용'의 균형 지점) + 무중단 (fault-tolerant) 이중화 (redundant)

- [주요 고려사항]
- latency 절감위해 작은 파일 구성 → 높은 IOPS 필요
  - 낮은 IOPS사용을 고려한 큰 파일 저장 → re-packetizing 추가 비용 발생
  - ABR, DASH등을 고려한 여러 set저장 → 저장소 사용량 증가

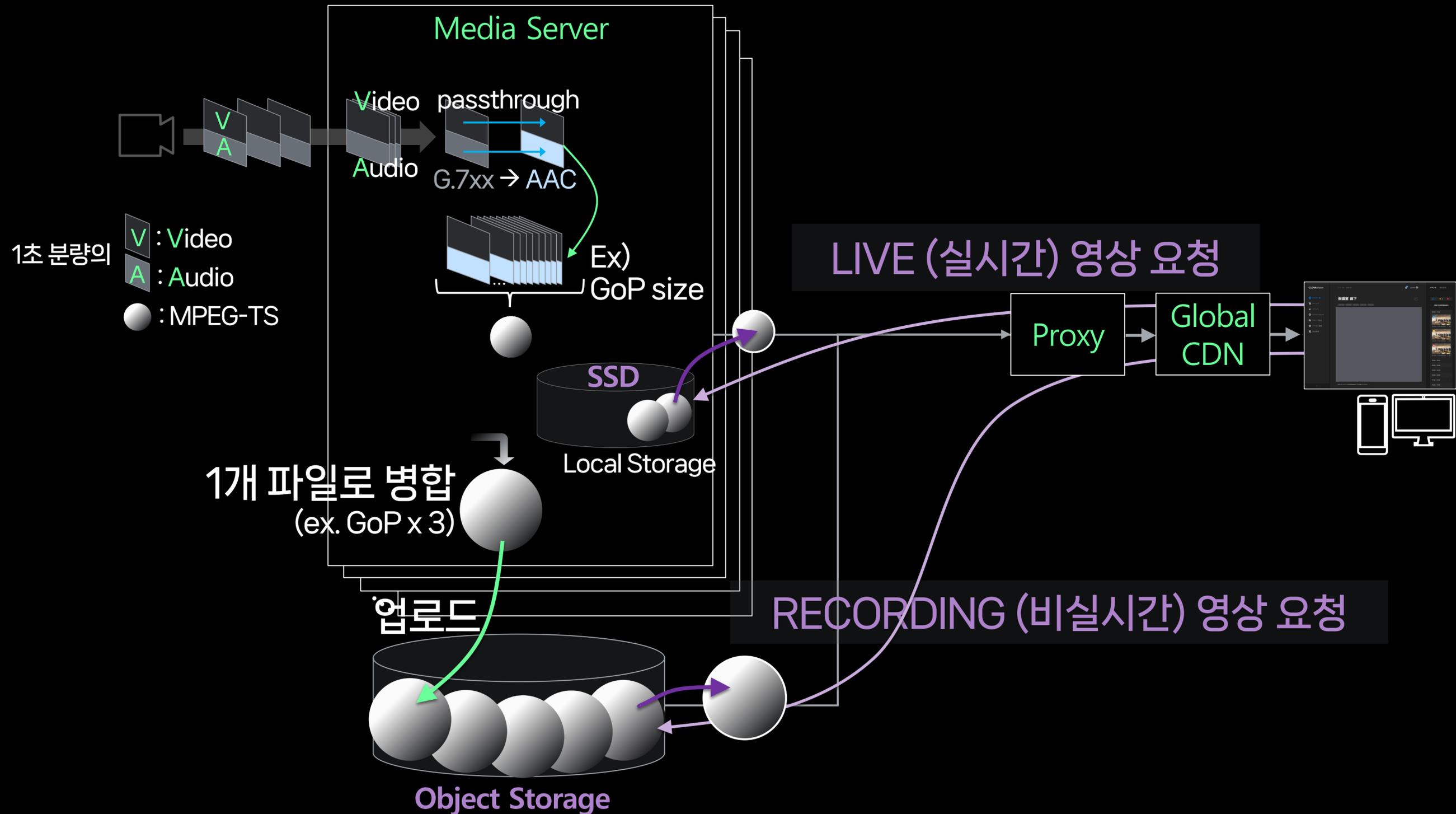


## 두 종류의 스토리지 활용 결정

- LIVE : SSD (Media Server개별)
- RECORD : Object Storage

얼마나? (특정용 용이) Object Storage?

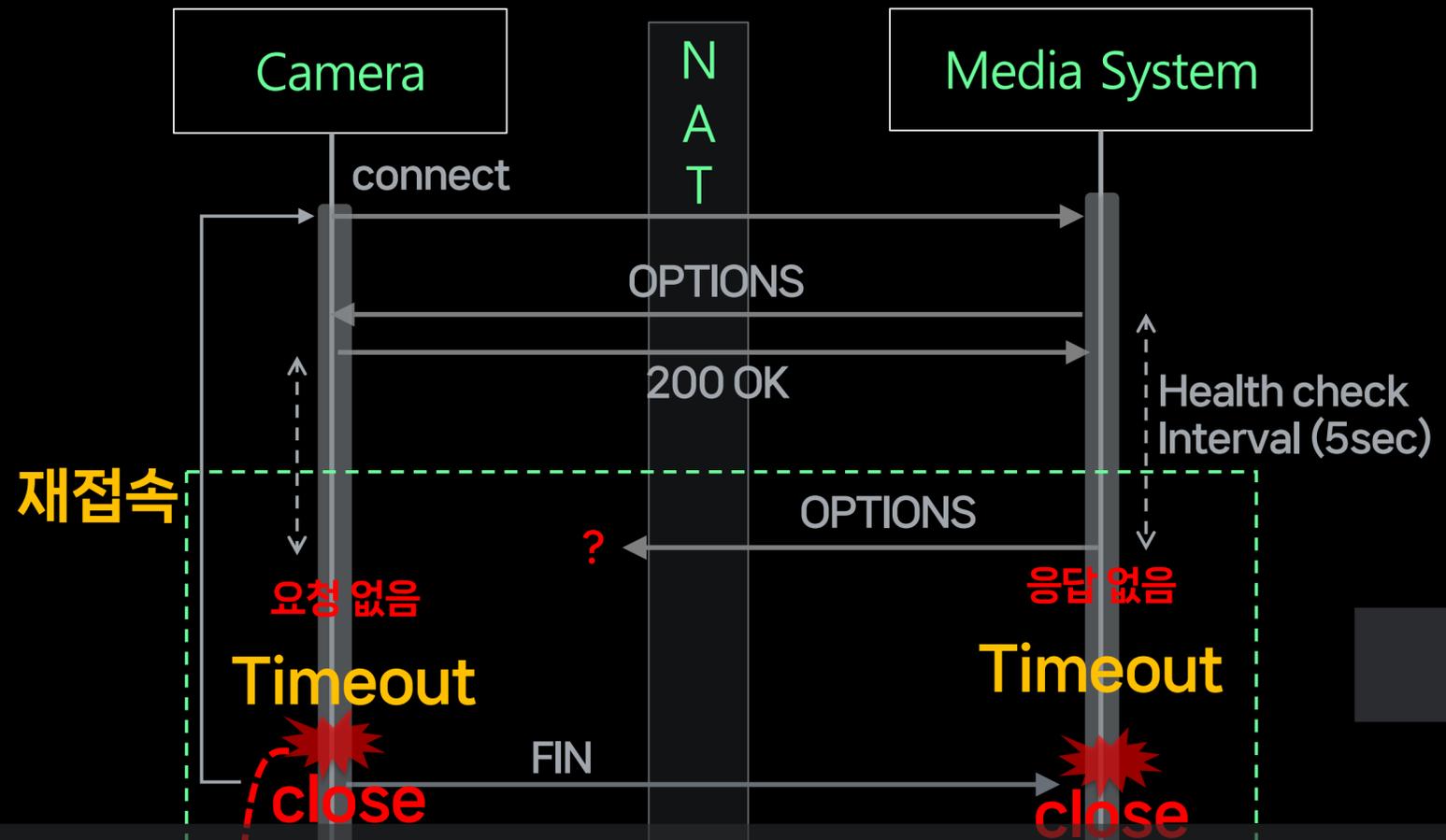
# 5.5 영상 저장 & 스트리밍 방식 (2/2)



# 5.6 실시간 스트리밍 지연시간 줄이기

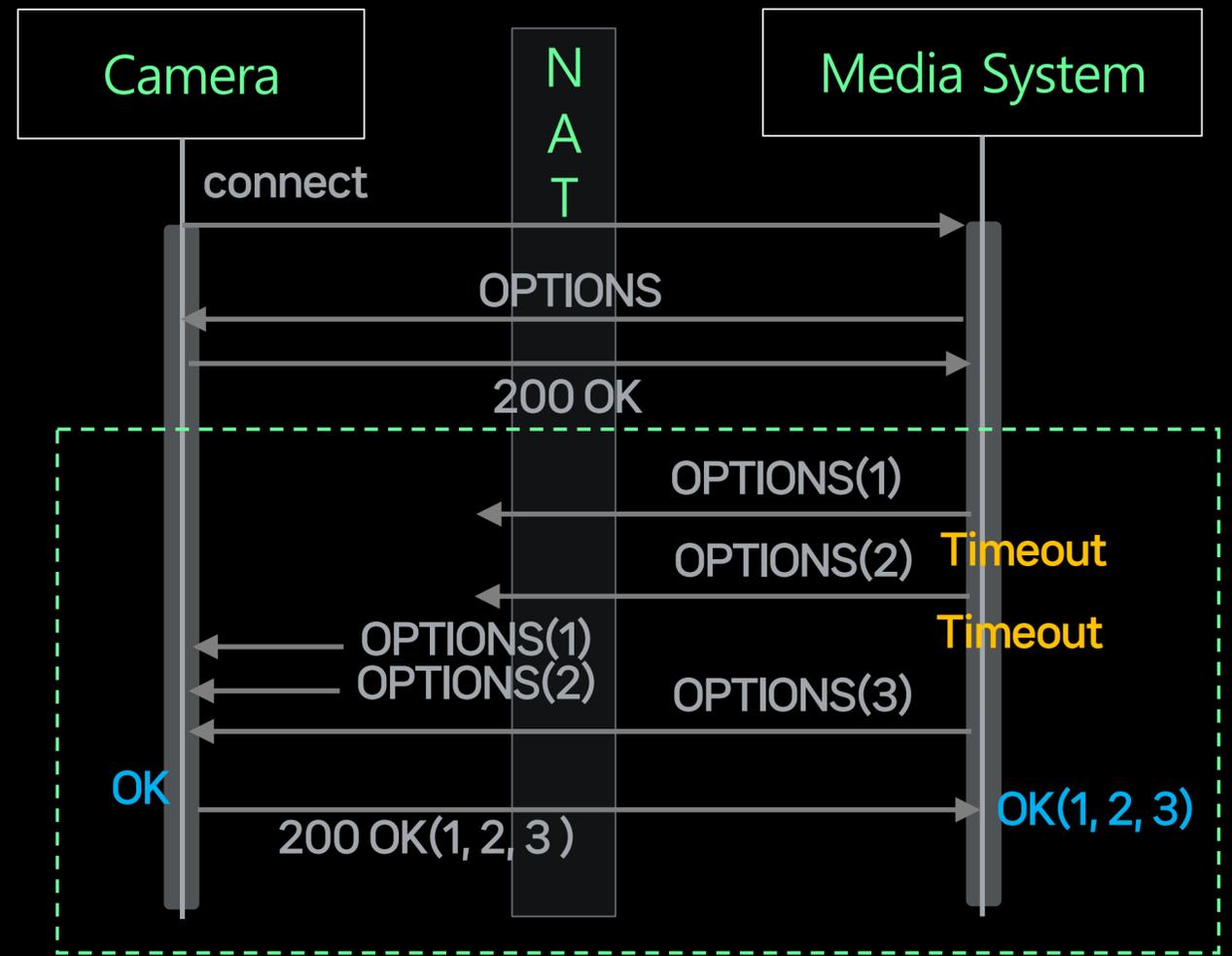


# 5.7 미디어 세션 유지 & 상태체크 방법



Timeout 처리 → 즉시 연결 해제 & 연결 재시도

패킷(상태 체크 메시지) 유실이 원인!



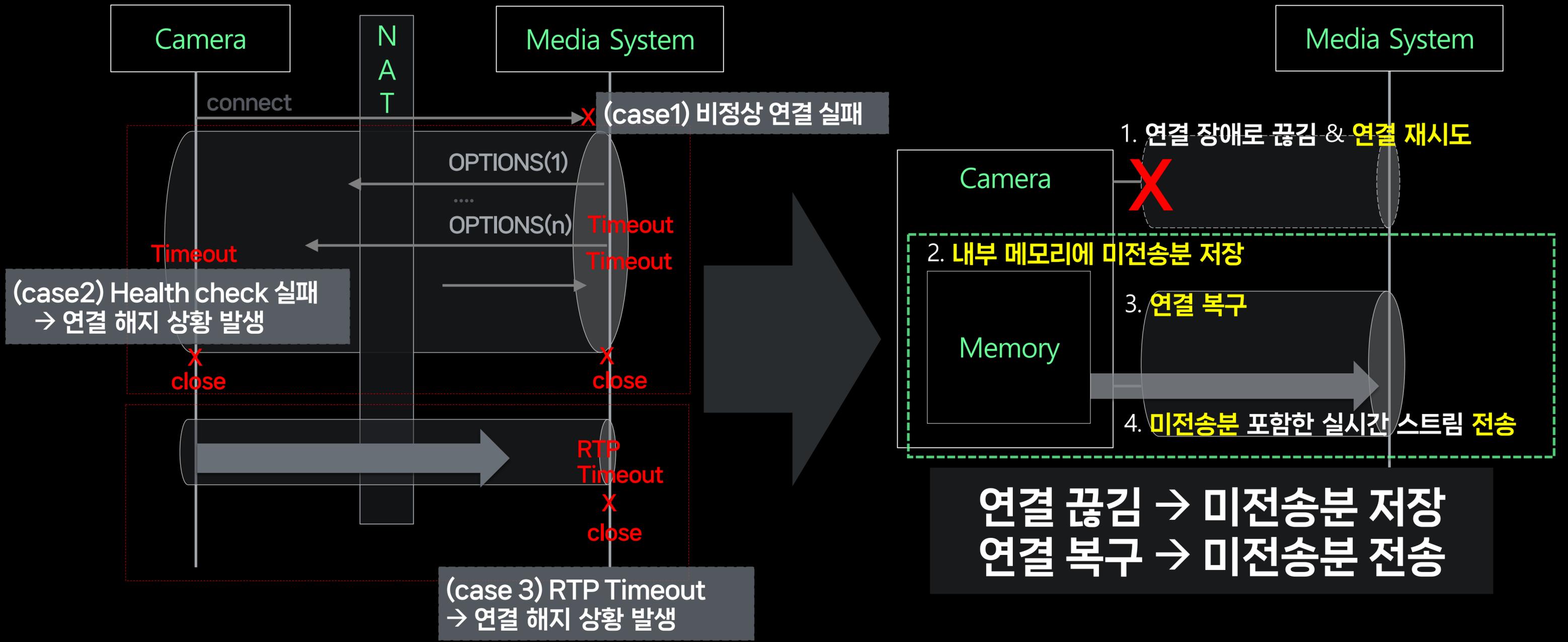
Timeout 처리 → 상태체크 n회 재시도  
(TCP retransmission에 의해 정상 처리)

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ All Addresses	6732				0.0418	100%	0.2300	19.965
192.168.1.184	6732							
192.168.1.1	1							
Media System 서버로부터 수신된 패킷	6731							
▼ All Addresses	24779				0.0422	100%	0.2600	69.637
10.213.88.28	24779				0.0422	100.00%	0.2600	69.637
10.213.88.28	24779				0.0422	100.00%	0.2600	69.637

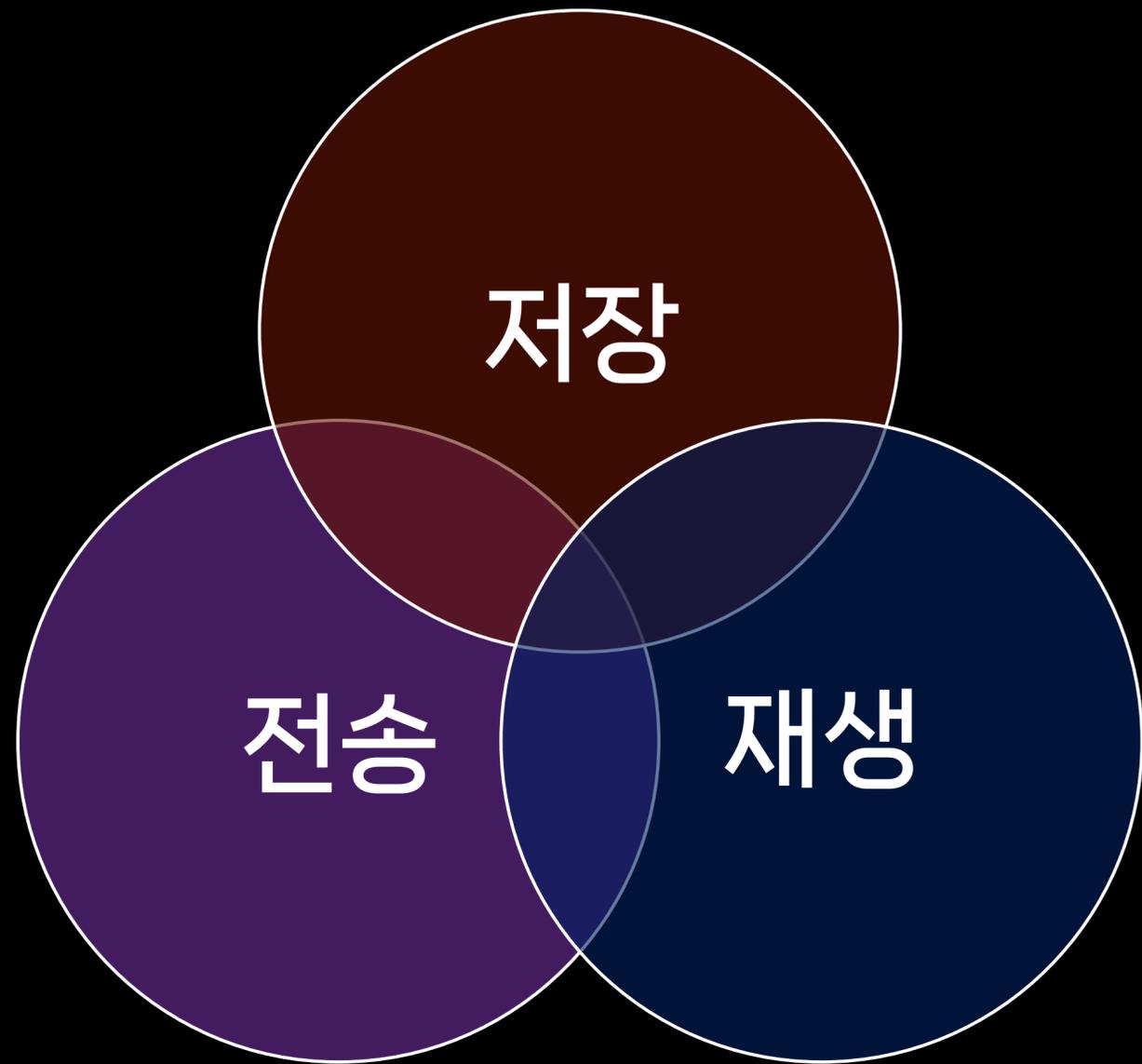
Camera 패킷

Media System 패킷

# 5.8 비정상 연결 해제 시 복구 과정



## 5.9 정리

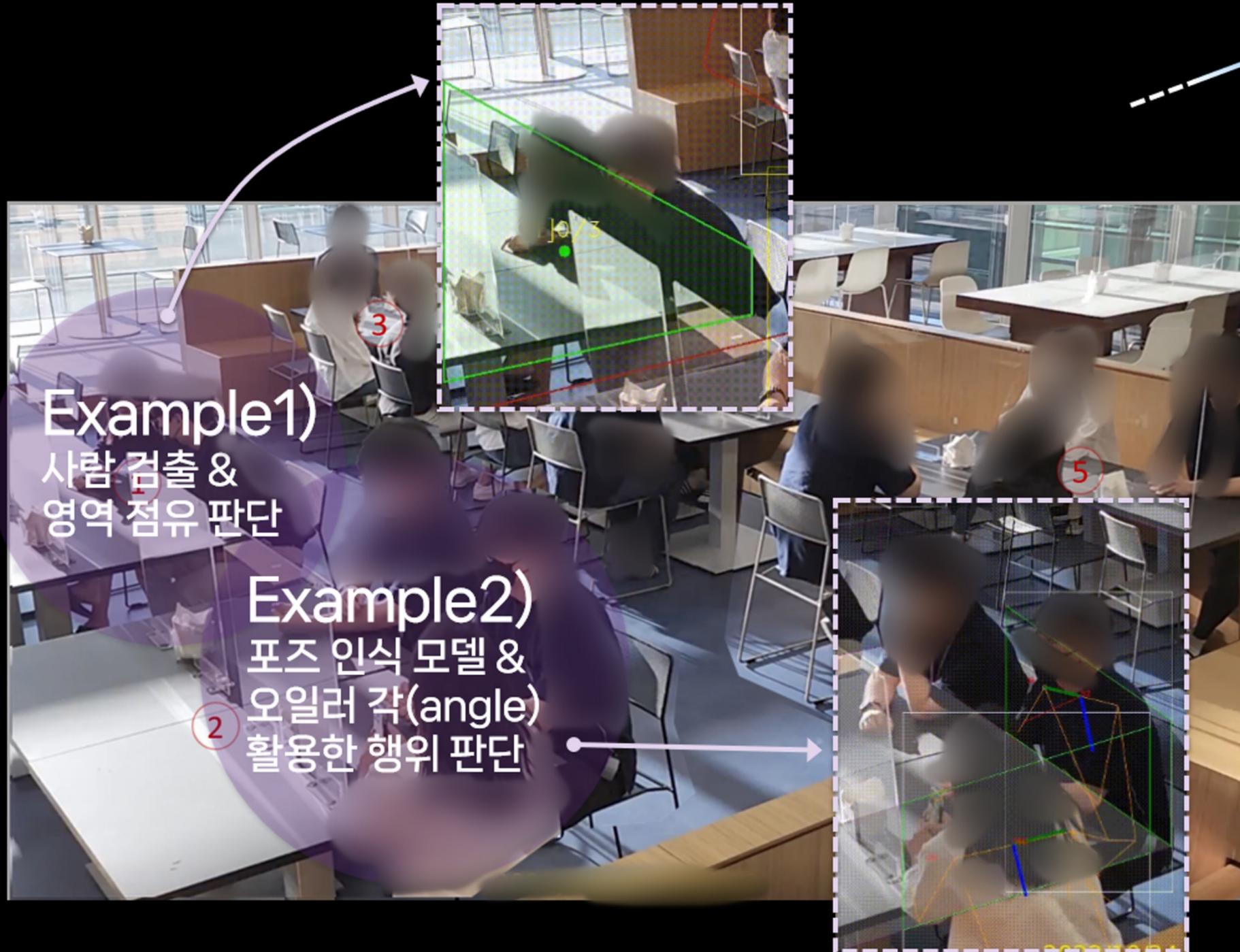


### 영상 데이터 처리 범위

1. 상호 밀접한 관계로 설계할수록, 같은 비용으로 & 더 나은 품질의 서비스
2. Cloud 보안 영상 서비스에 적합한 수준 선택 "최고" vs "적정"

# 6. CLOVA Vision과 AI의 만남

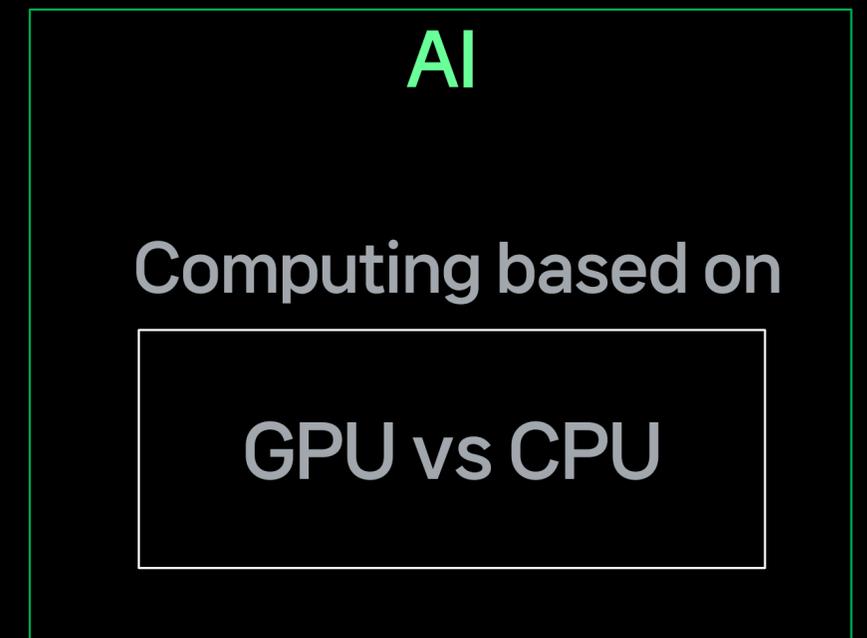
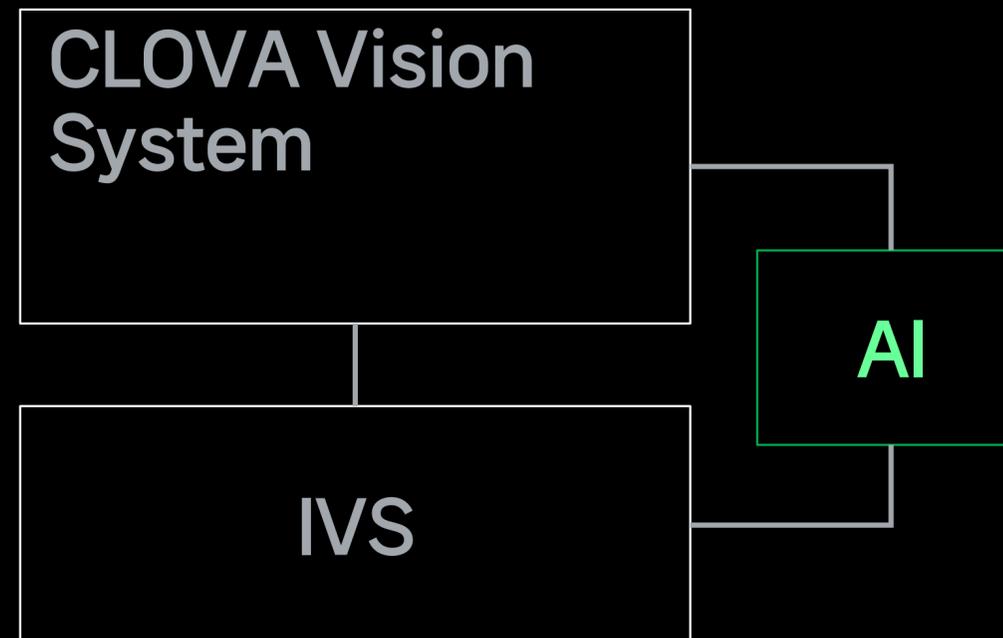
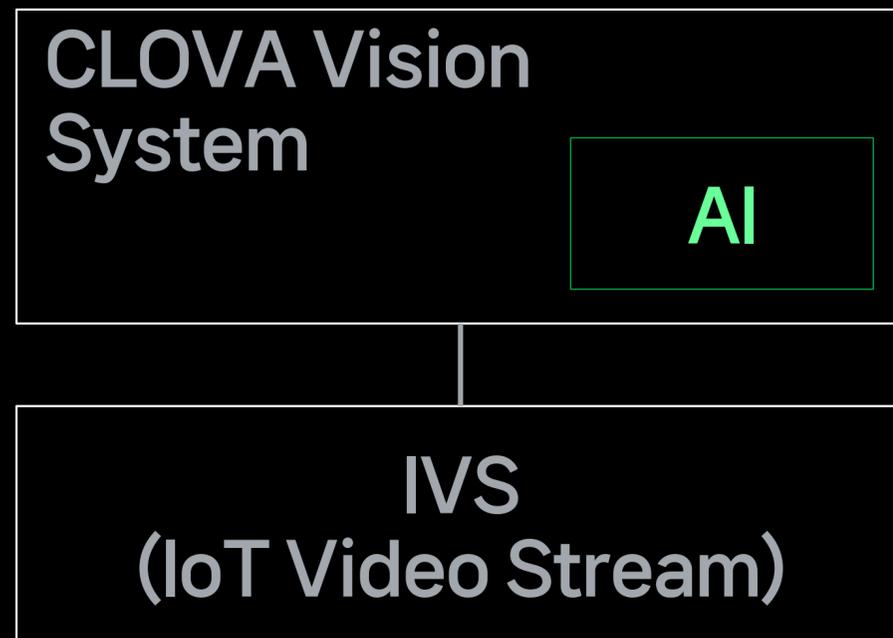
# 미리보기



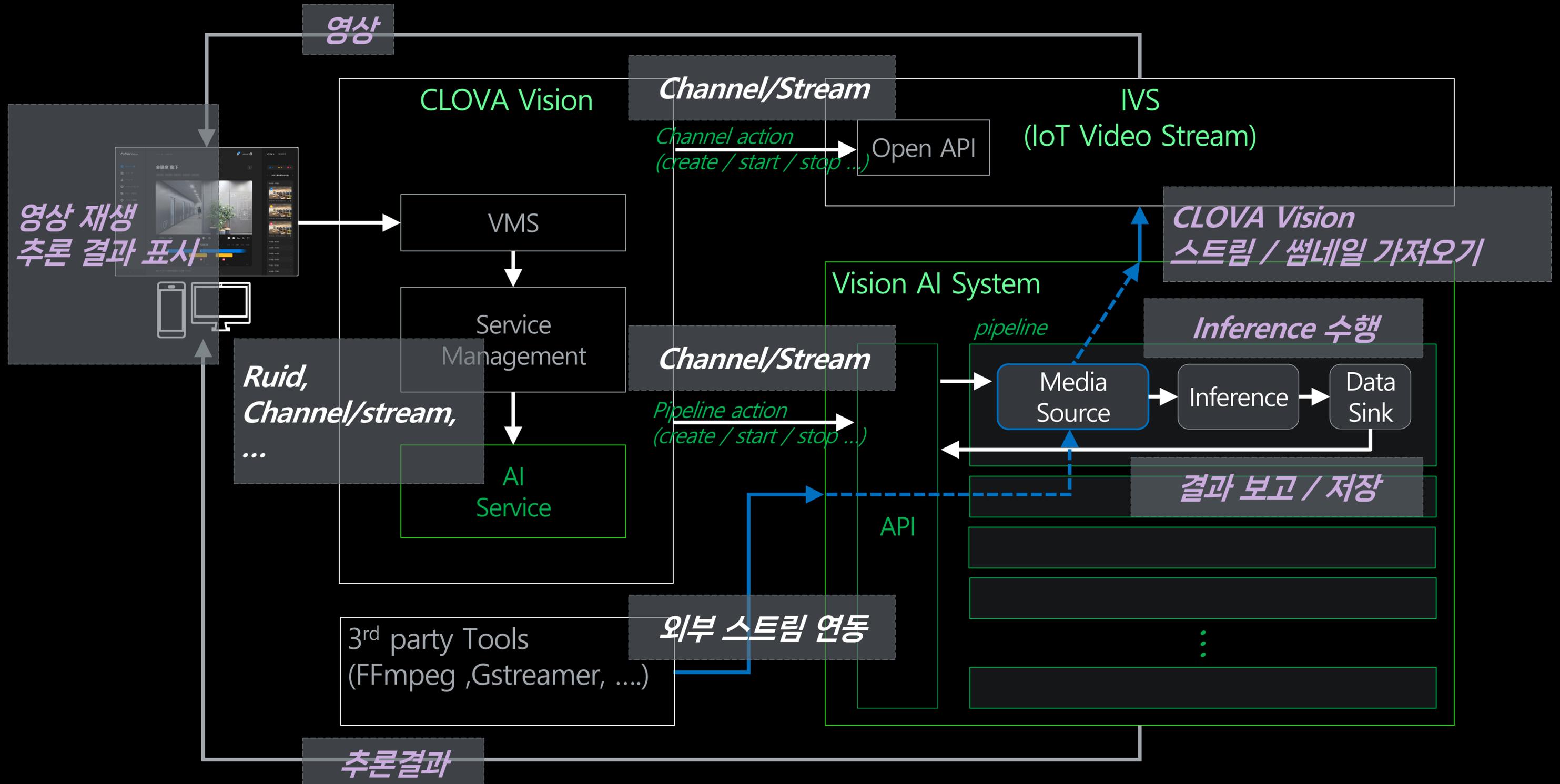
해당 이미지는 학습에 사용된 영상이 아닙니다.

# 6.1 개요

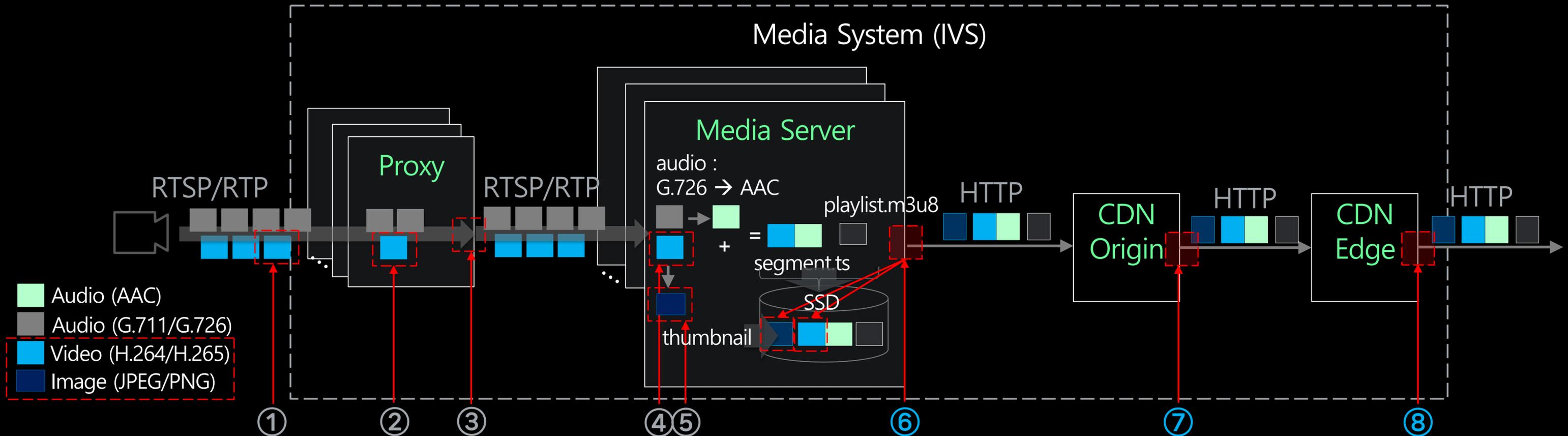
CLOVA Vision의 미디어를 어떻게 가져올 것인가?  
어떤 미디어 객체를 이용할 것인가?  
추론 결과는 어떻게 제공하고 이용하게 할 것인가?



# 6.2 Concepts



# 6.3 미디어 연결 지점 & 방법 & 유형



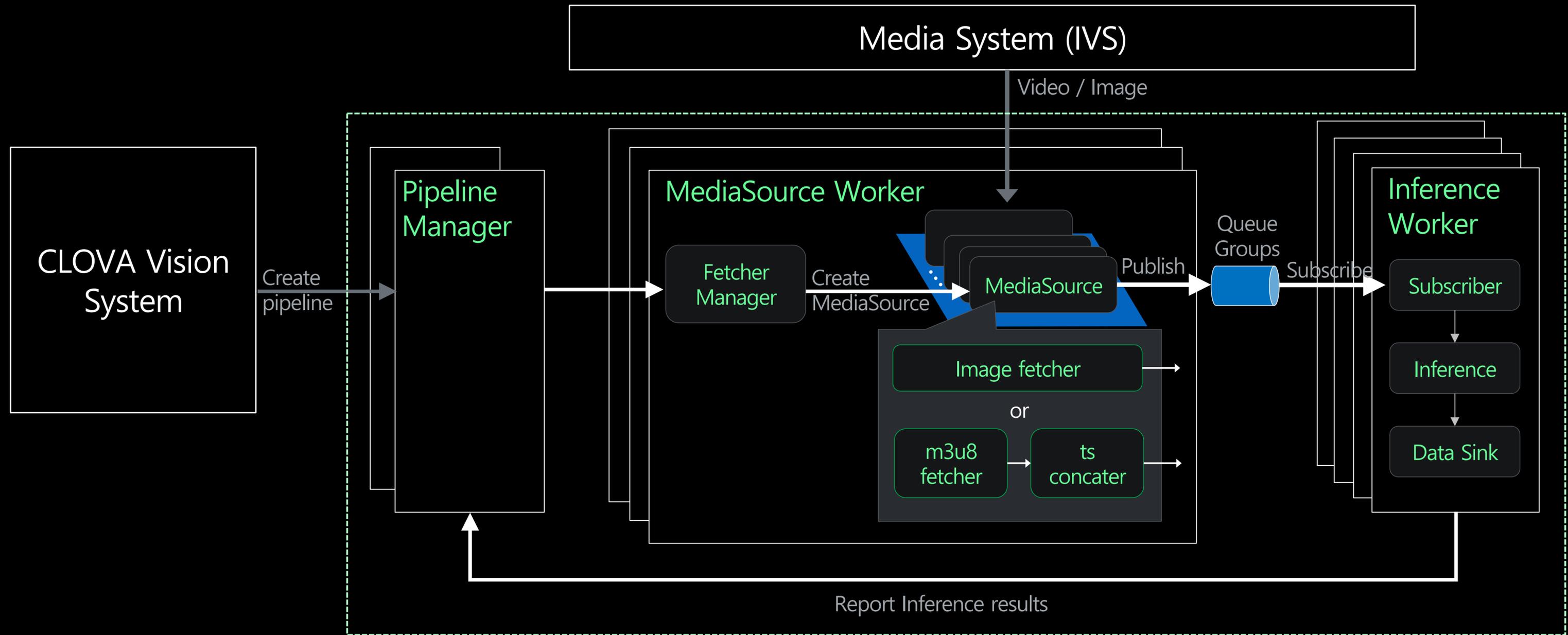
### 연결 가능 지점 & 방법

①② : push RTSP/RTP	④ : push video (H.264)	⑥⑦⑧ : pull (GET) HTTP
③ : pull RTSP/RTP	⑤ : push thumbnail (JPEG/PNG)	

### 이용 가능한 미디어 유형

HLS 방식의 mpeg-ts 파일 활용 (m3u8/ts)	→ Video : H.264 / H.265	} 2종류
자동 저장된 thumbnail image 활용	→ Image : JPEG / PNG	

# 6.4 아키텍처 구성요소



# 6.5 확장 가능한 애플리케이션 설계

Service Discovery : 감시 대상 key prefix

→ 서버 그룹별 host, sticky task, 등...



서버 그룹별 host 등록 / 조회 / 감시

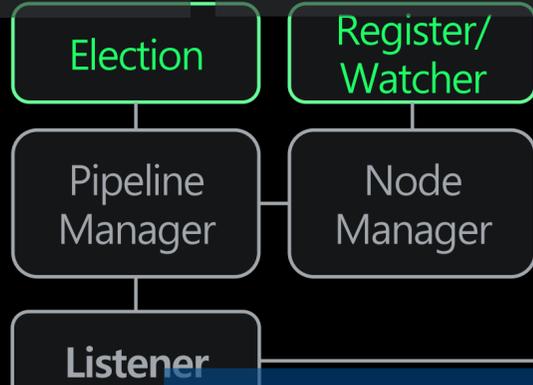
IVS

Leader  
선출

PM서버그룹  
등록/조회/감시

MS서버그룹  
등록/조회/감시

IF서버그룹  
등록/조회/감시

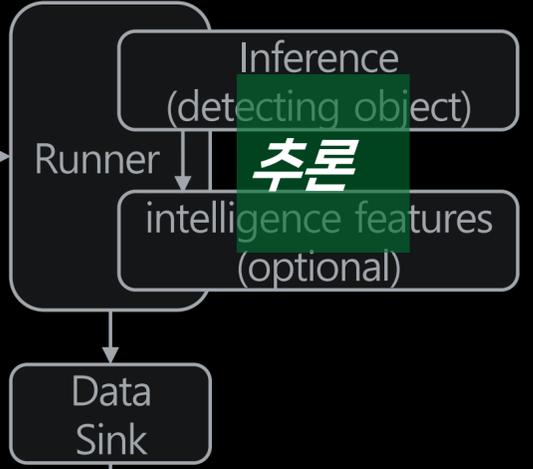
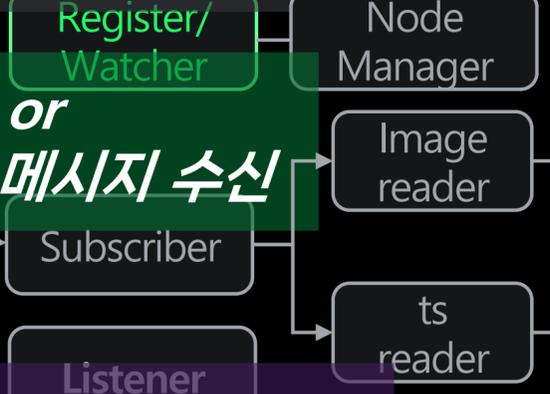


부하분산,  
서버선택 & pipeline제어  
(생성/조회/중지/삭제 등)



1. non-sticky → wildcard + QueueGroup
2. Sticky → 단일 subject

QueueGroup or  
Host subject 메시지 수신



결과 DB에 저장 or  
필요시 특정 MS에 응답

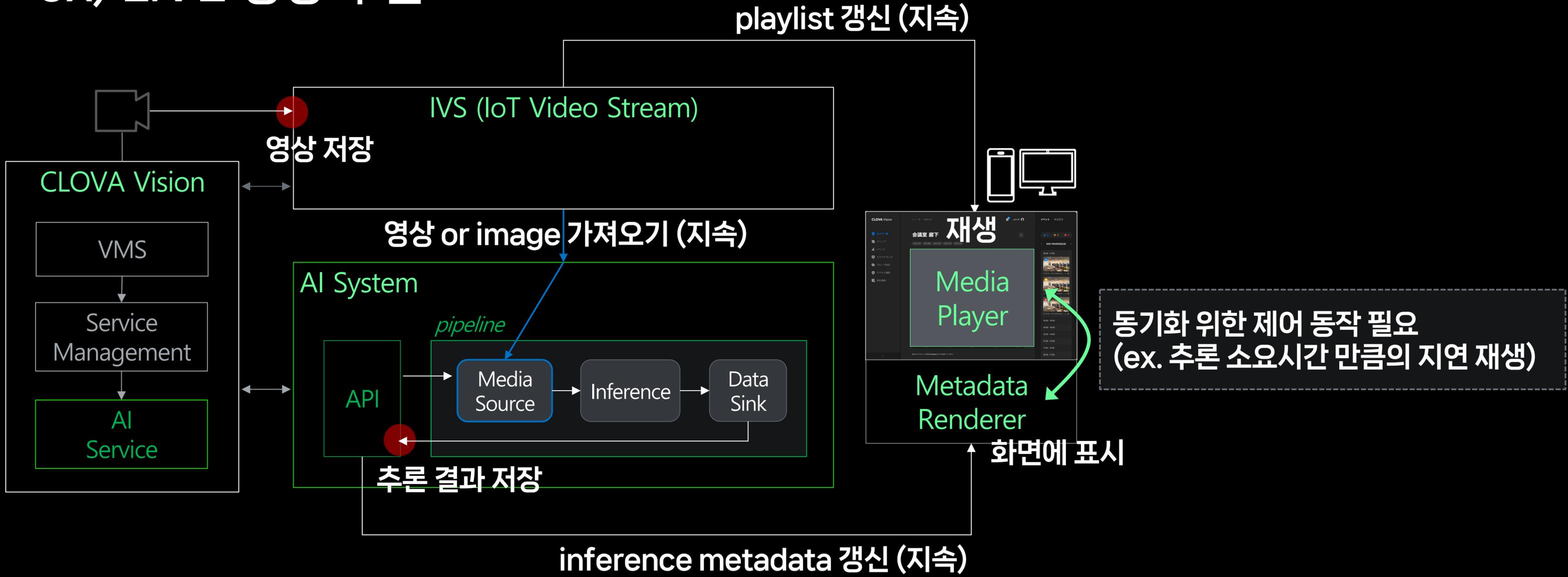
PM (Pipeline Manager)

MS (MediaSource)

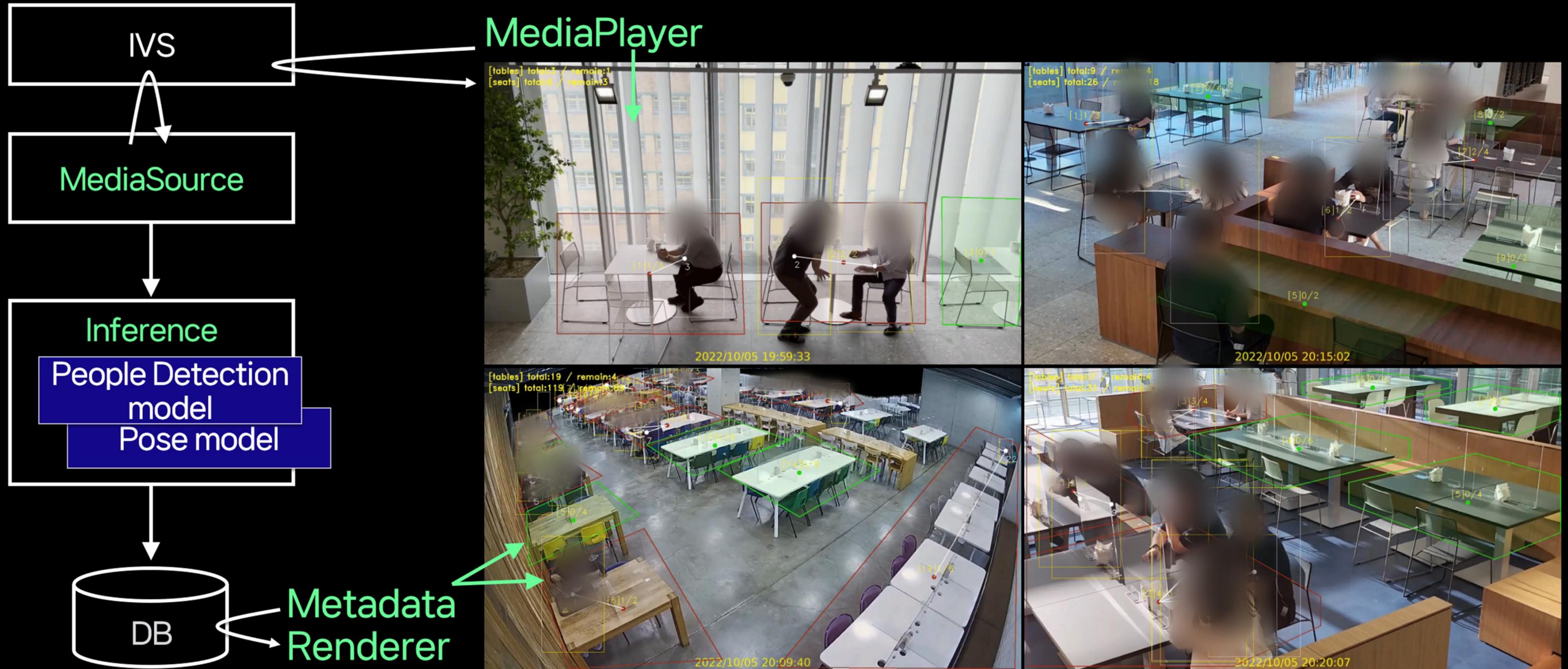
IF (Inference)

# 6.6 영상 & 추론 결과 스트리밍 방법은?

## ex) LIVE 영상 추론

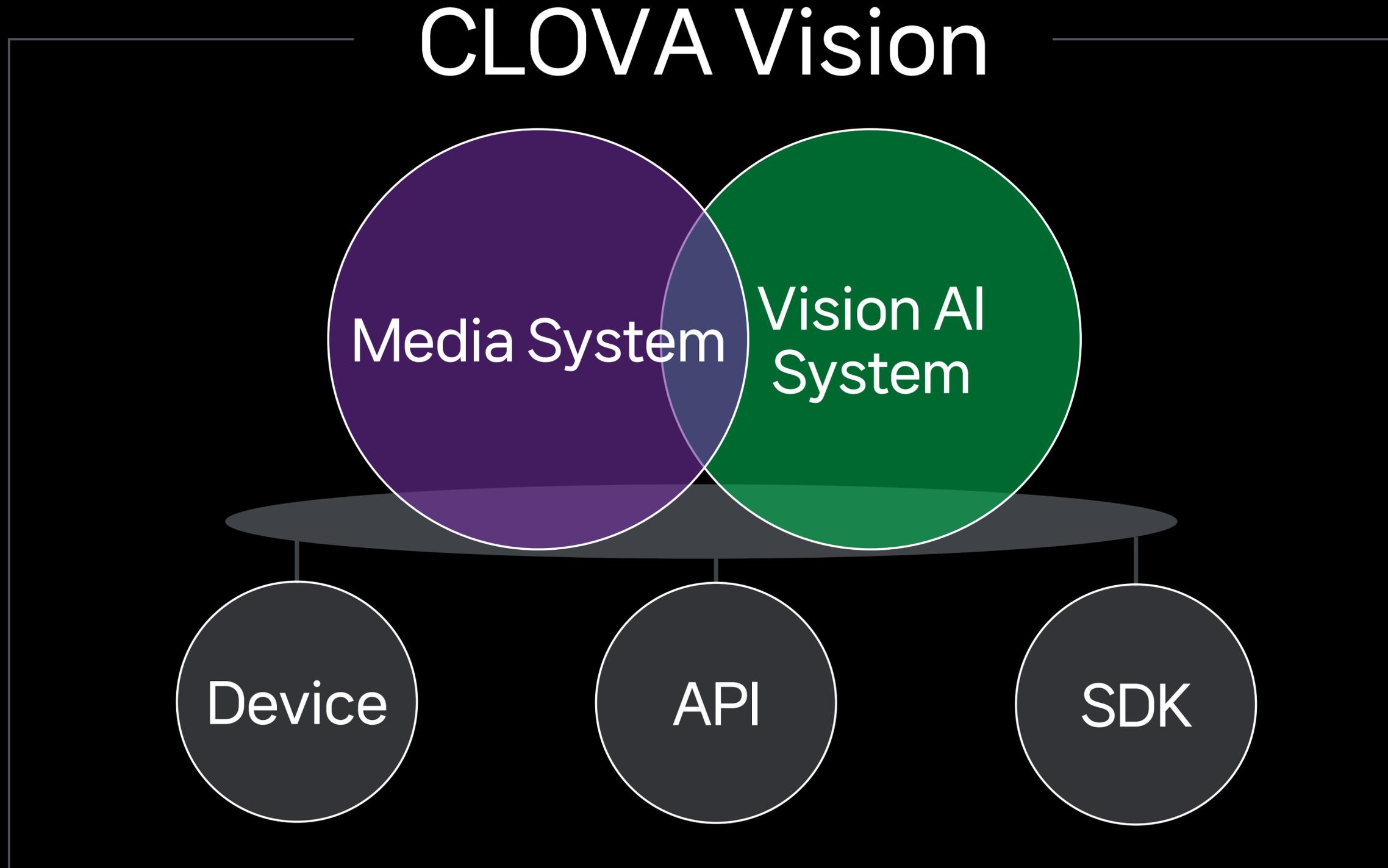


# 6.7 공식 감지 AI 서비스 : 데모



해당 이미지는 학습에 사용된 영상이 아닙니다.

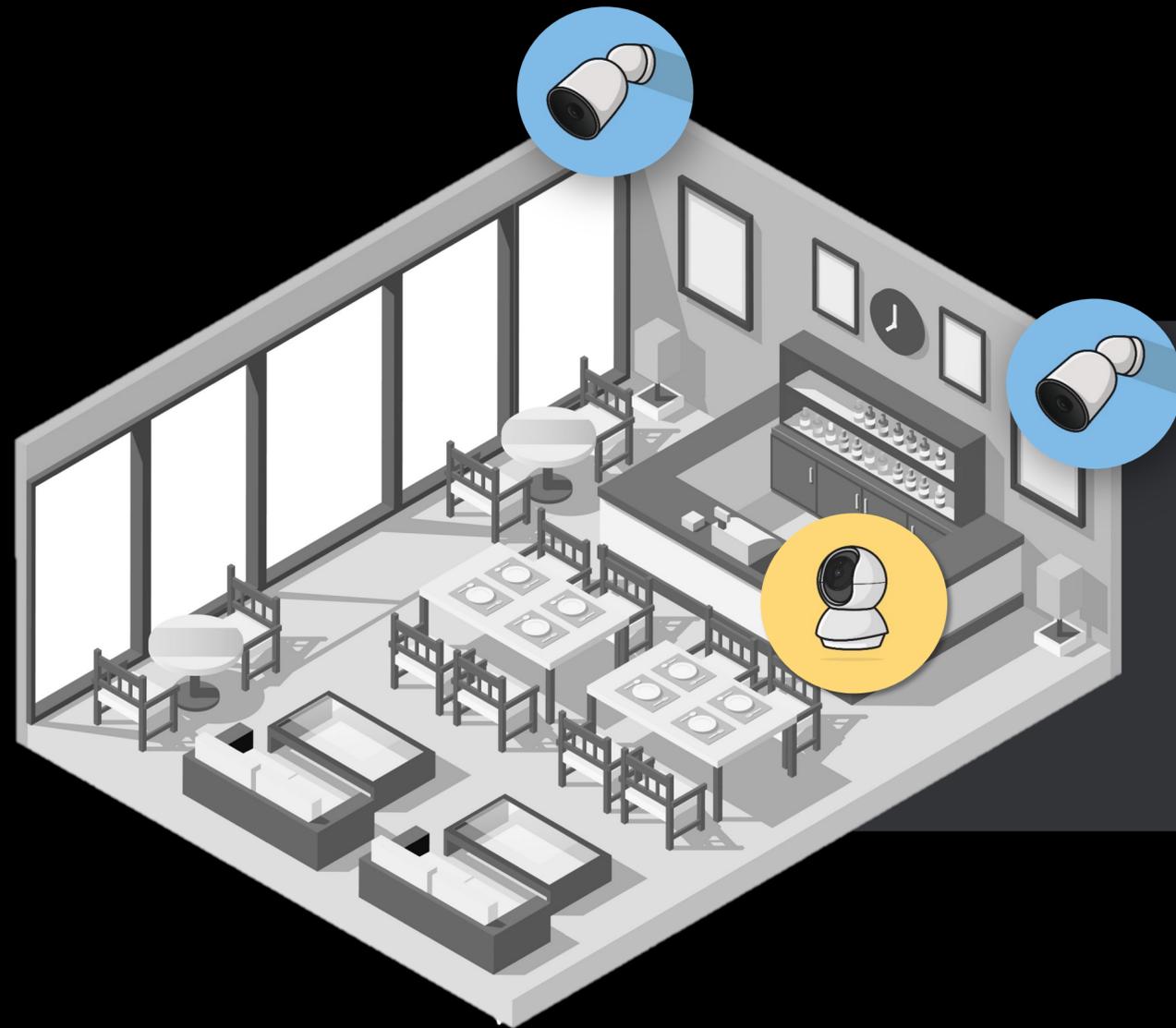
# 6.8 정리



Security Service (22.09)

Vision AI

→ 현재



CLOVA Vision

Camera Device

Camera Control System

Media System

Vision AI System

*in progress*



**Q & A**

**Thank You**