

2022

Capstone Project

최종 발표 - 11조

# CONTENTS. |

01

프로젝트 개요  
Metabusking

02

주요 기술 및 기능

03

기대 효과

04

Next Plan  
추후 계획

05

시연 영상

Chapter.1

# 프로젝트 개요

Metabusking

01

---

# MetaBusking

온라인 버스킹 및 음원 공유 플랫폼



Metaverse



Busking



MetaBusking

# 02

---

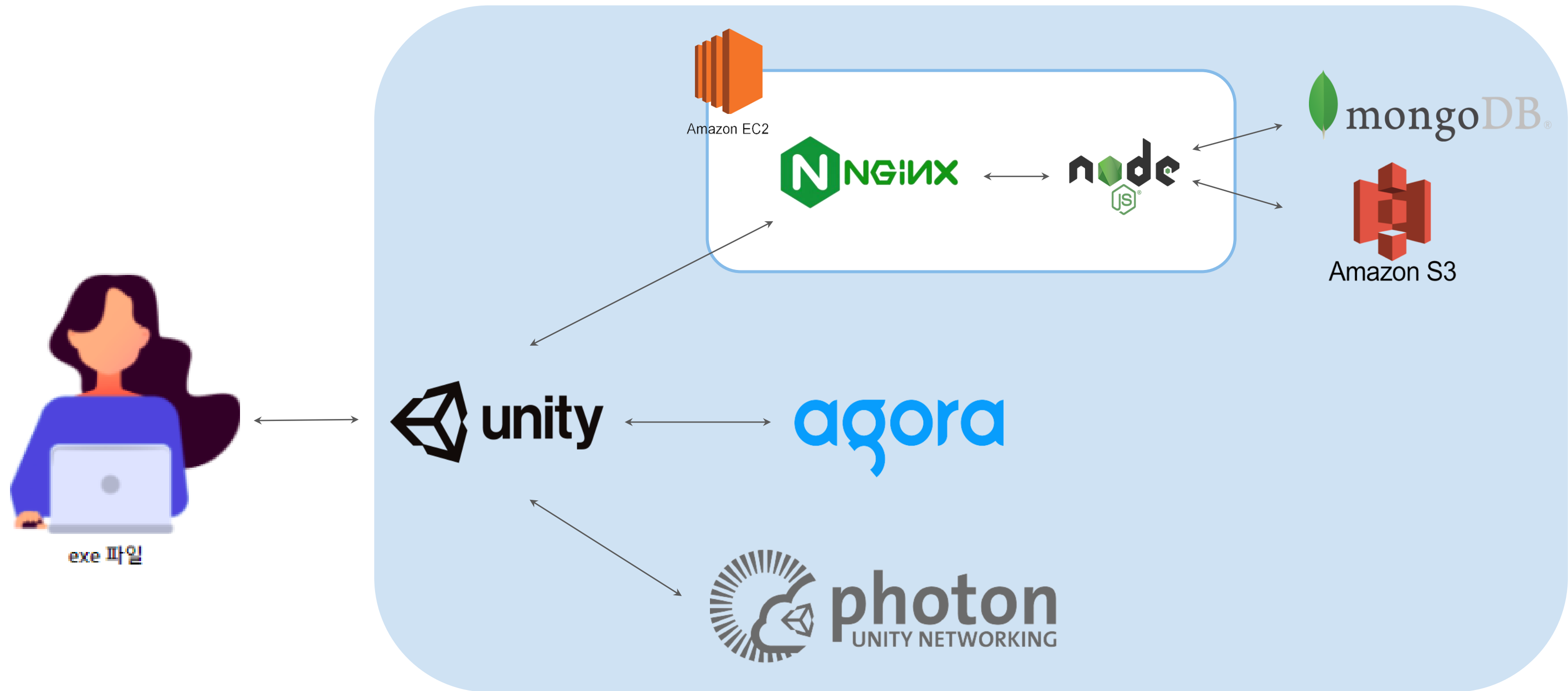
## 주요 기술 및 기능

# 주요 기술

---

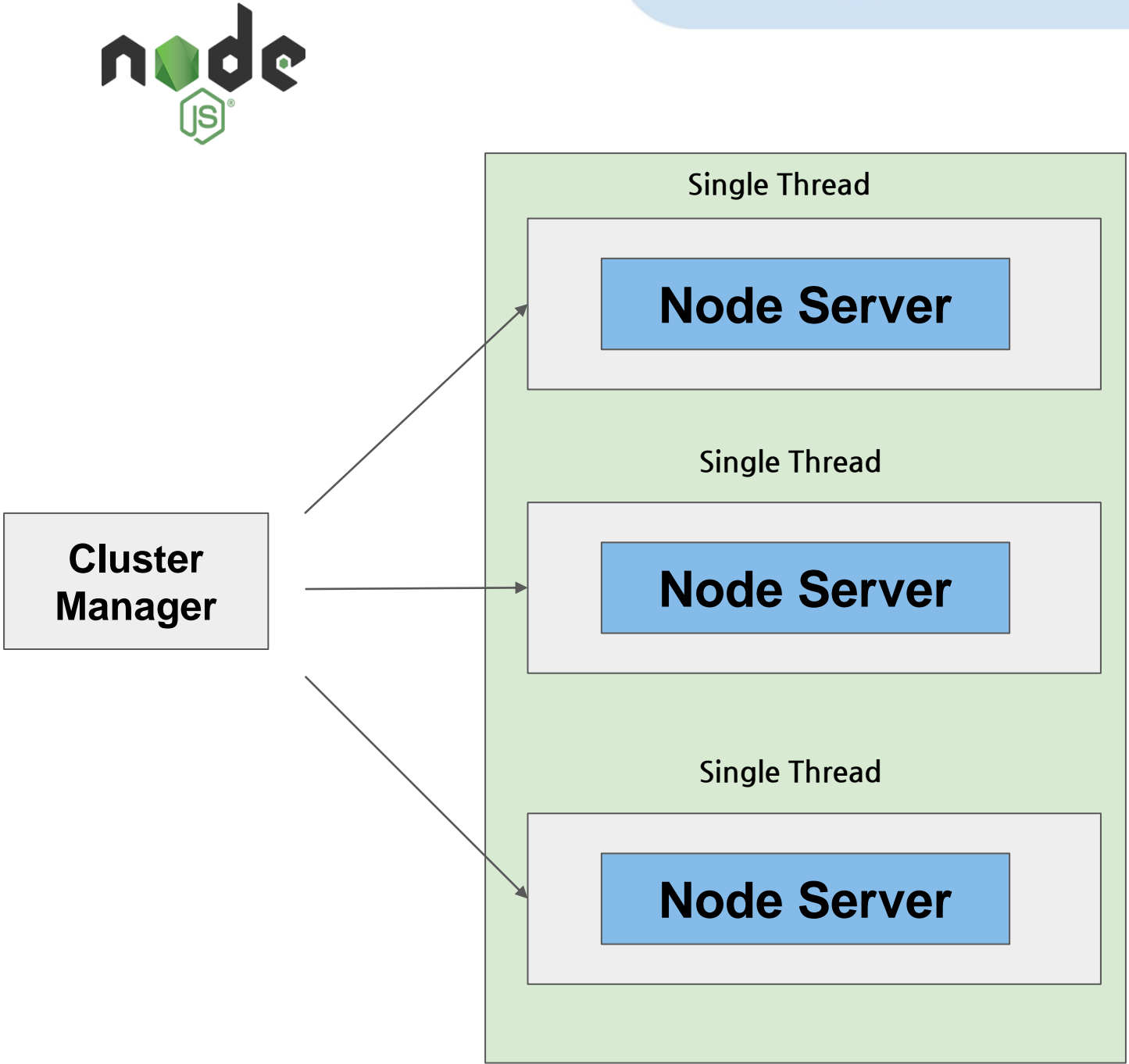
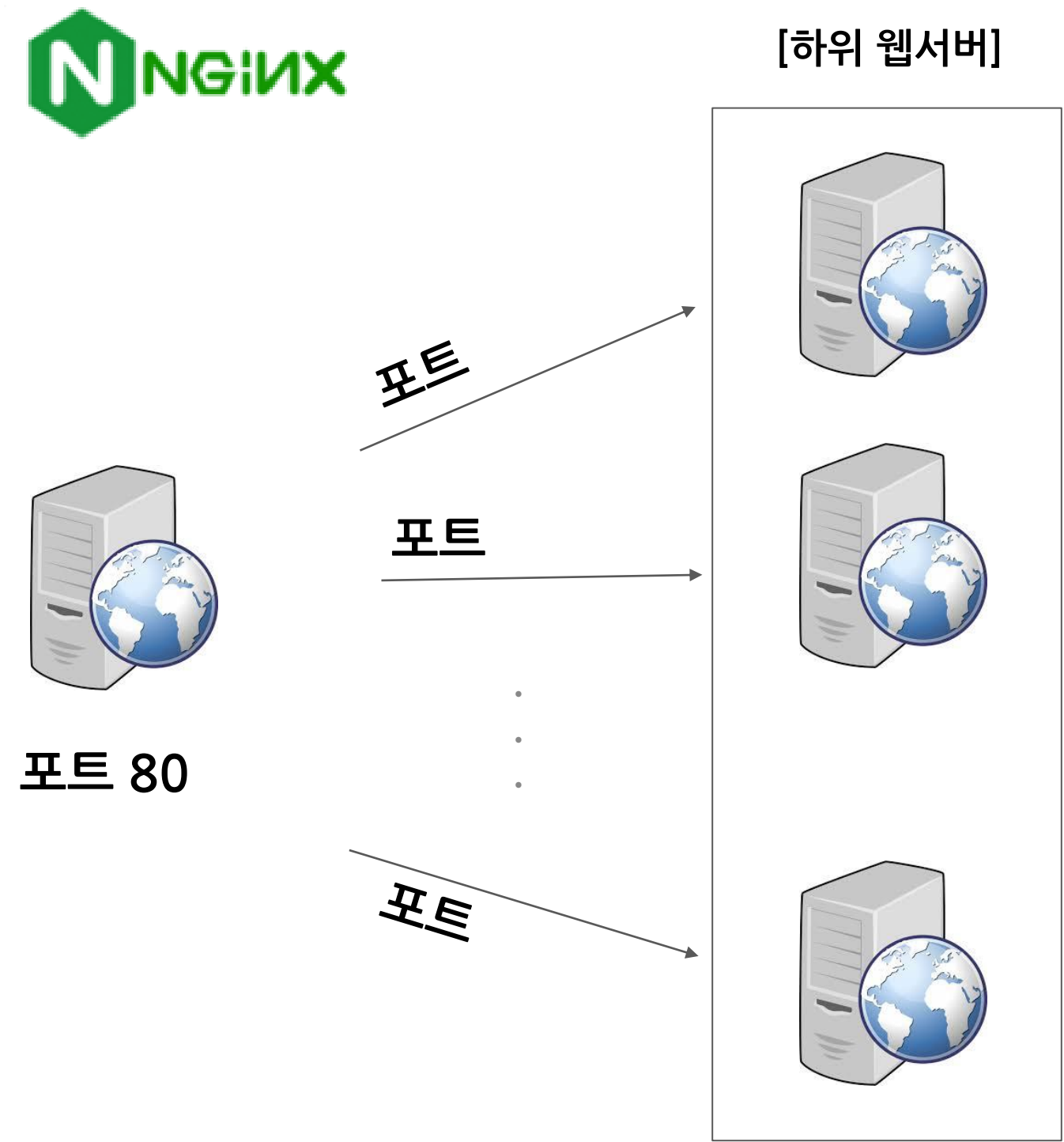
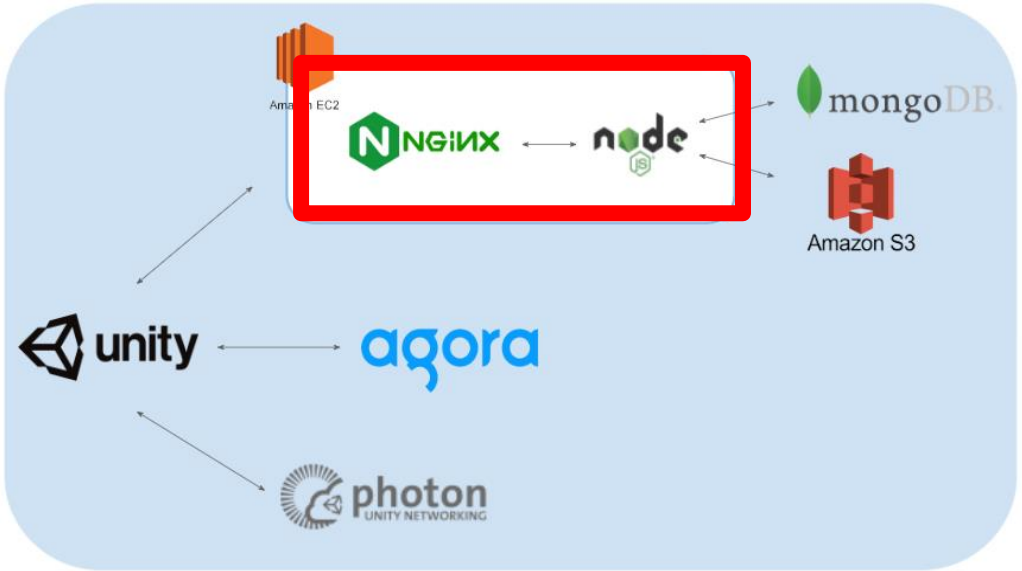
# 시스템 구성도

소프트웨어 아키텍처



# Nginx와 NodeJS

Metabusing 주요 기술





# 라이브 스트리밍

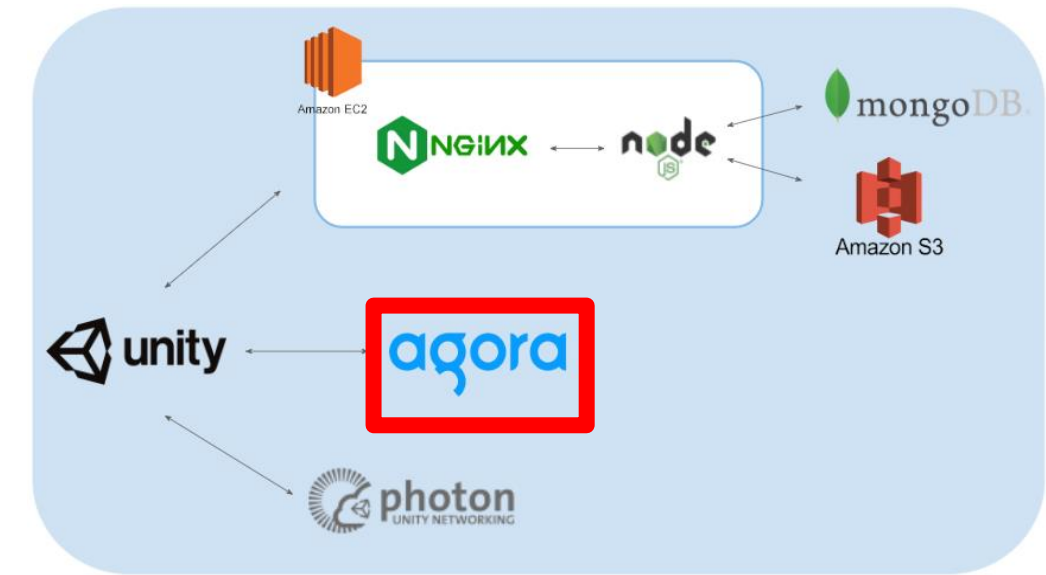
Metabusing 주요 기술

기존 기술(WebRTC) 변경 이유

- 유니티와 서버 사이의 통신 문제
- 유니티와 node.js 간의 API 구조 문제
- 유니티와 node.js 간의 미디어 타입 문제

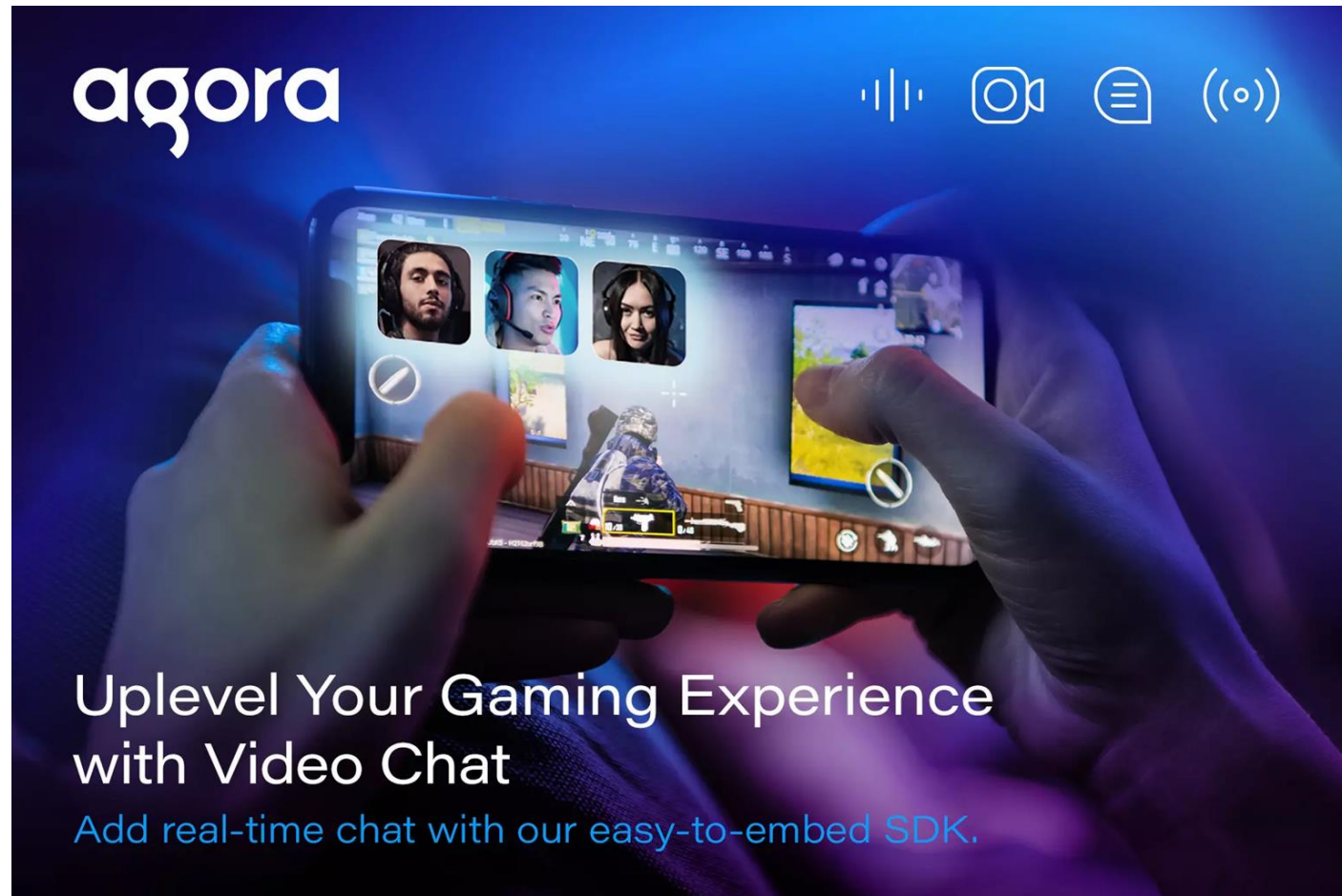


새로운 기술 (Agora)을 적용하기로 결정

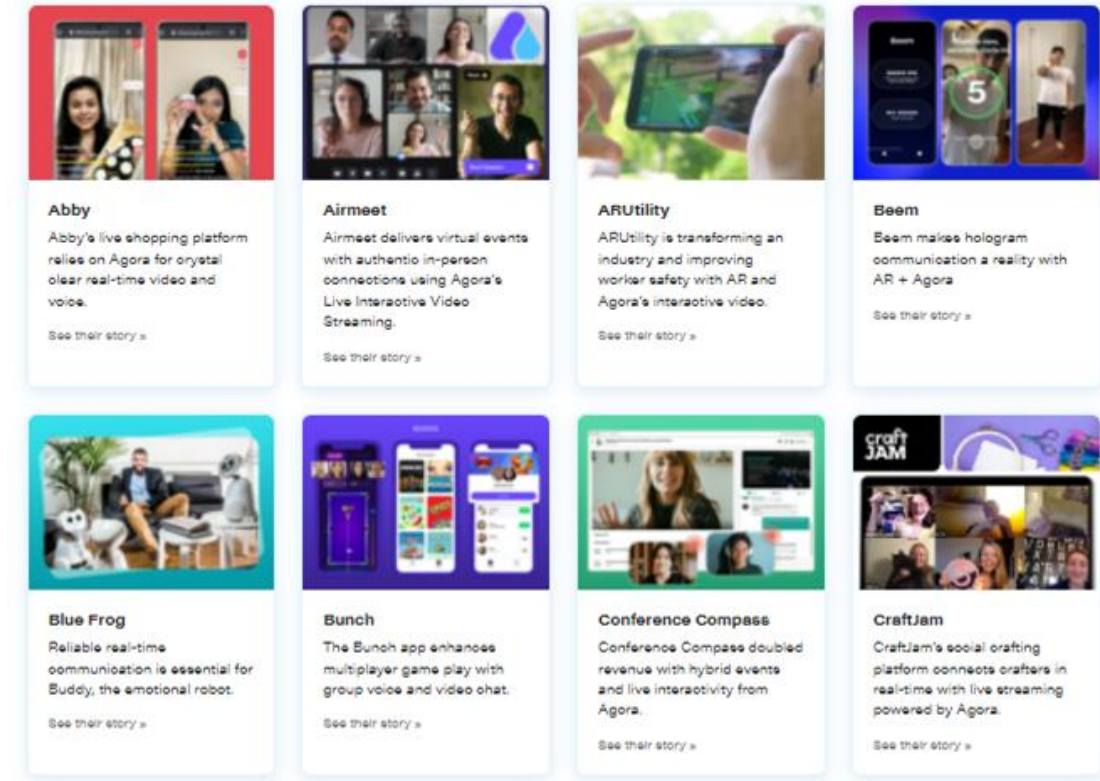


# Agora

Metabusking 주요 기술



Agora Unity Asset 발견



1. 시중에서 많이 사용



iOS

android



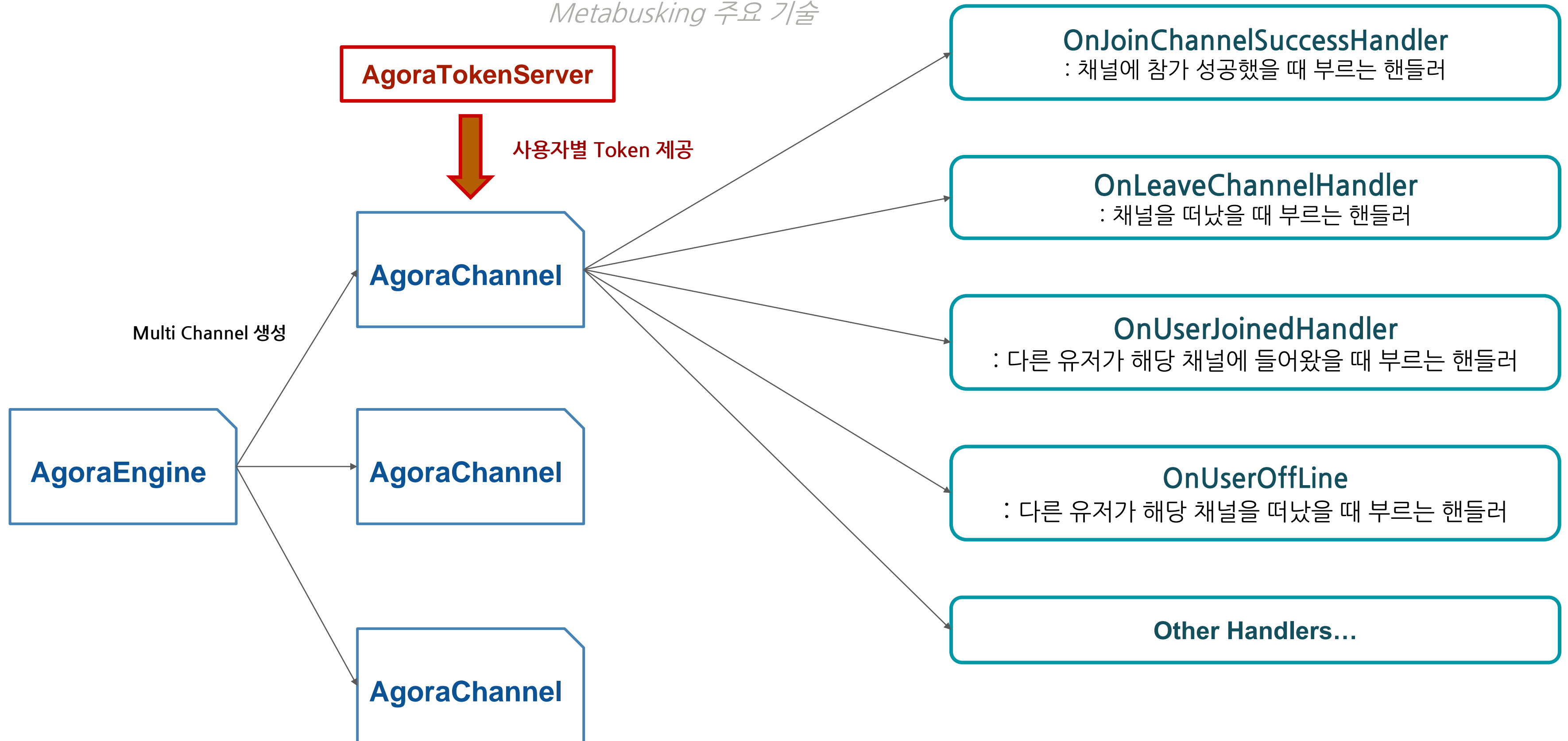
WEB

2. 높은 확장성과 멀티 플랫폼 지원

3. SD-RTN™ 네트워크로 인한 좋은 통신 환경 제공

# Agora

Metabusing 주요 기술

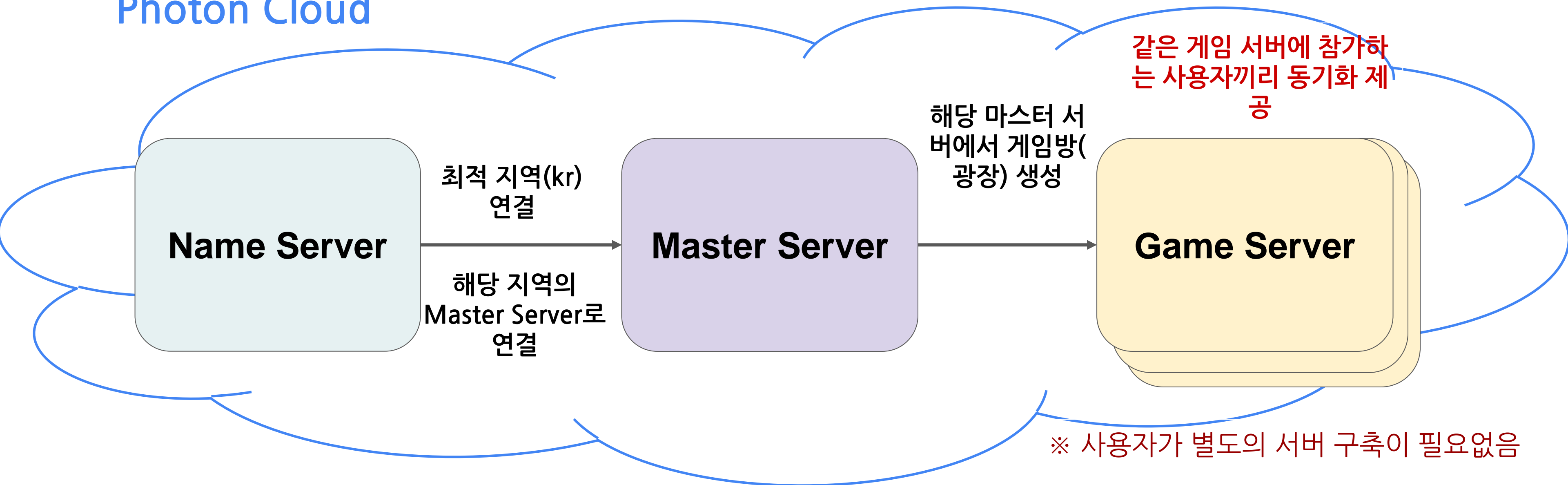


# Photon

Metabusing 주요 기술



## Photon Cloud



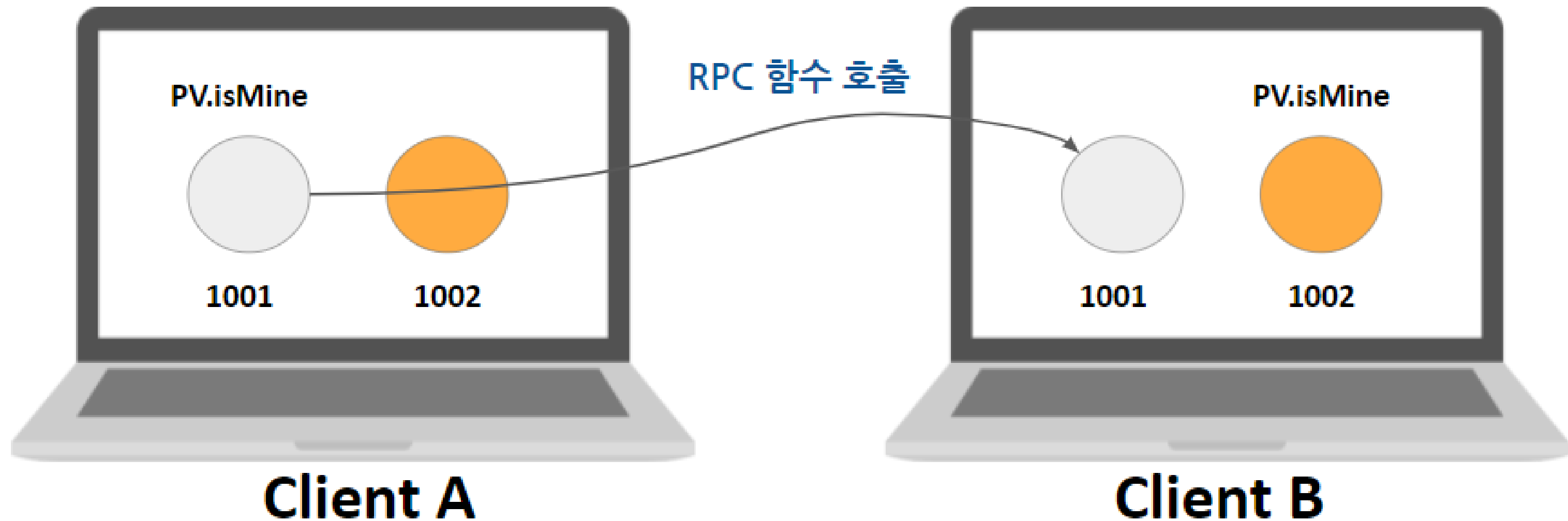
# Photon

*Metabusking* 주요 기술

PhotonView 컴포넌트를 단 객체는 동기화 함수 호출 가능

RPC(Remote Procedure Call) 함수 호출을 통한 리모트 유저들에게 로컬 유저를 동기화

동기화



# Router 종류

Metabusing 주요 기술





# MongoDB CRD - user

Metabusing 주요 기술

필드	Description	필수 사항
id	유저를 구분하는 고유 이름	O
email	유저의 이메일	O
password	로그인 시 사용하는 문자열	O
nickname	유저의 게임 상의 이름	O
character	유저의 외형을 나타내는 값	O
listName	유저의 재생목록의 이름을 가진 리스트	O
uploadList	유저가 업로드 한 리스트	O
preferredGenres	유저의 선호장르	O
totalNum	유저가 올린 음원의 개수	O
created	올린 음원의 id를 만들 때 사용	O
follow	유저가 팔로우 한 유저의 id 리스트	O
follower	유저를 팔로워 한 유저의 id 리스트	O
[musicList]	유저가 만든 음원 리스트	X

# MongoDB CRD - music

Metabusking 주요 기술

필드	Description	필수 사항
locate	음악이 저장된 위치	0
imageLocate	음악의 이미지가 저장된 위치	0
title	음원 제목	0
id	음원을 나타내는 고유 값	0
userID	음원을 올린 유저의 id	0
userNickname	음원을 올린 유저의 닉네임	0
category	음원의 카테고리	0
lyrics	음원의 가사	0
info	음원의 정보	0
time	음원의 재생시간	0
created	음원의 업로드 날짜	0
playedNum	음원의 재생 횟수	0



# MongoDB CRD - zone

Metabusking 주요 기술

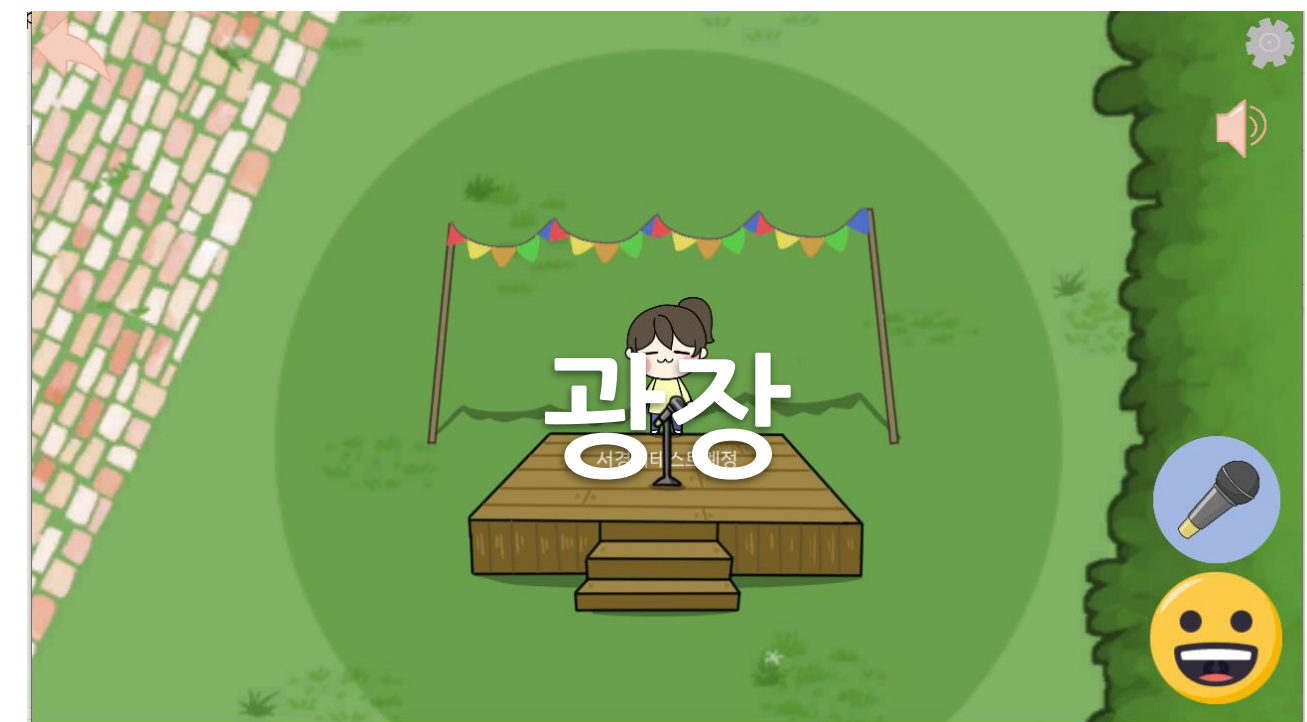
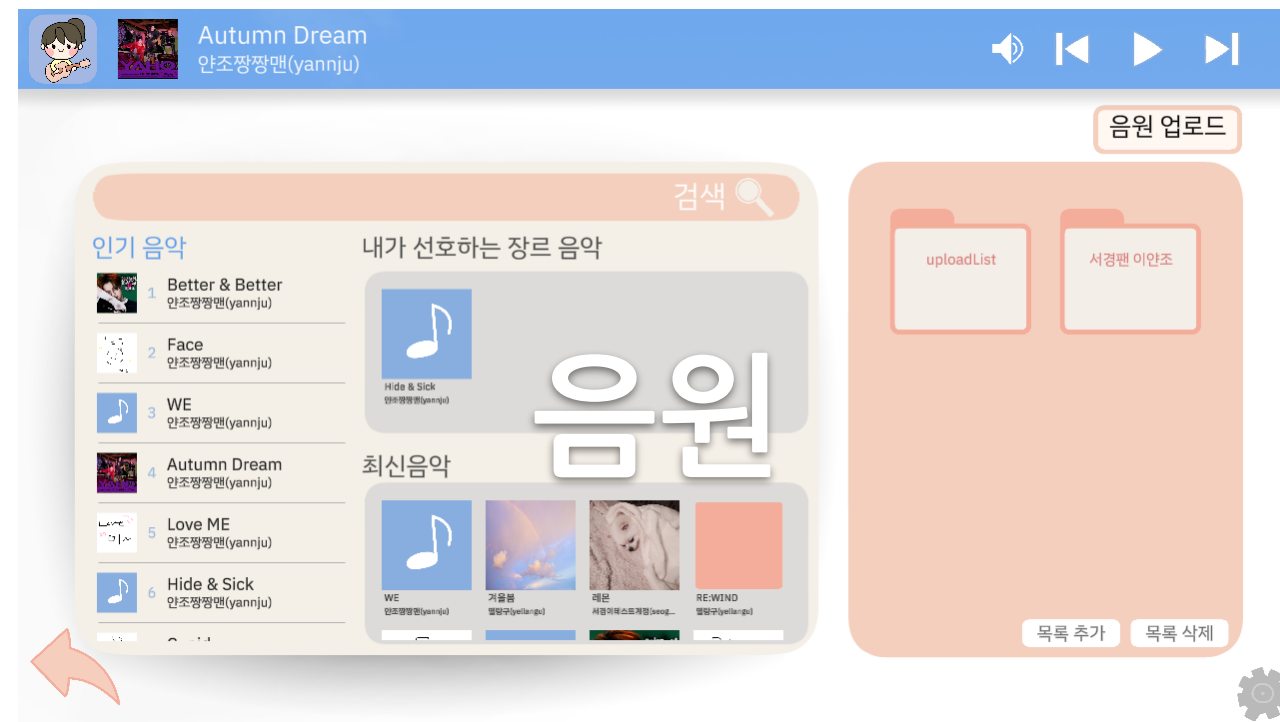
필드	Description	필수 사항
musicTitle	음원의 제목이 적힌 리스트	0
musicList	음원의 위치가 적힌 리스트	0
musicTime	음원의 재생시간이 적힌 리스트	0
totalTime	음원의 재생시간 합	0
id	음원 존을 구분하는 고유 값	0

# 주요 기능

---

# 화면 구성

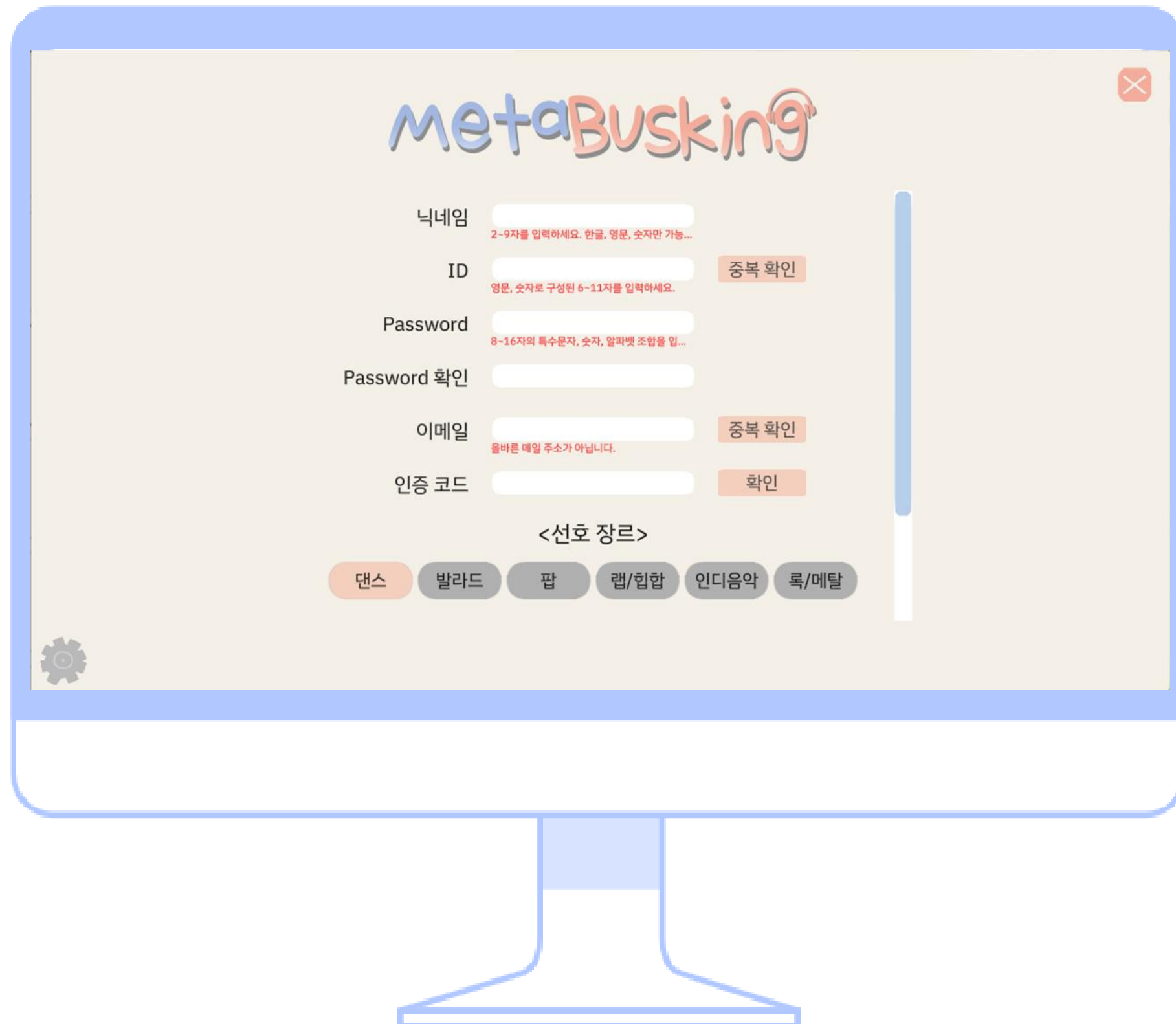
Metabusing



# 회원 가입

Metabusing 주요 기능

node JS + express



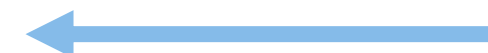
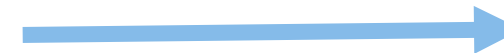
The image shows a registration form for 'Metabusing' on a computer monitor. The form includes fields for Nickname, ID, Password, Password confirmation, Email, and a verification code. It also features a genre selection section with buttons for Dance, Ballad, Pop, Rap/Hip-hop, Indie music, and Rock/Metal. The Metabusing logo is at the top, and a gear icon is in the bottom left corner.

id, email 전송

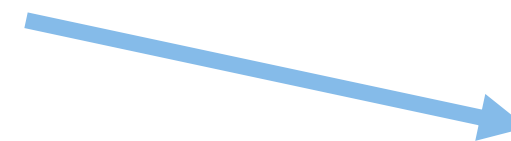


존재 여부 전송

메일 전송 요청



OK



데이터 존재 확인



Nodemailer

메일 전송

클라이언트에서 입력한 값 확인 후 인증 완료

# 회원 가입



The image shows a registration form for 'MetaBusking' displayed on a computer monitor. The form includes the following fields and elements:

- MetaBusking** logo at the top.
- 닉네임** (Nickname): Input field with a red error message: "2~9자를 입력하세요. 한글, 영문, 숫자만 가능..."
- ID**: Input field with a red error message: "영문, 숫자로 구성된 6~11자를 입력하세요." and a **중복 확인** (Check duplicate) button.
- Password**: Input field with a red error message: "8~16자의 특수문자, 숫자, 알파벳 조합을 입..."
- Password 확인**: Input field for password confirmation.
- 이메일** (Email): Input field with a red error message: "올바른 메일 주소가 아닙니다." and a **중복 확인** (Check duplicate) button.
- 인증 코드** (Verification code): Input field and a **확인** (Check) button.
- <선호 장르>** (Preferred genre): A section with buttons for **댄스** (Dance), **발라드** (Ballad), **팝** (Pop), **랩/힙합** (Rap/Hip-hop), **인디음악** (Indie music), and **록/메탈** (Rock/Metal).
- A gear icon for settings is located in the bottom left corner.

nickname, id,  
password, email,  
선호 장르 전송



User data 전송

입력 받은 데이터



데이터 저장  
(password는 해싱)

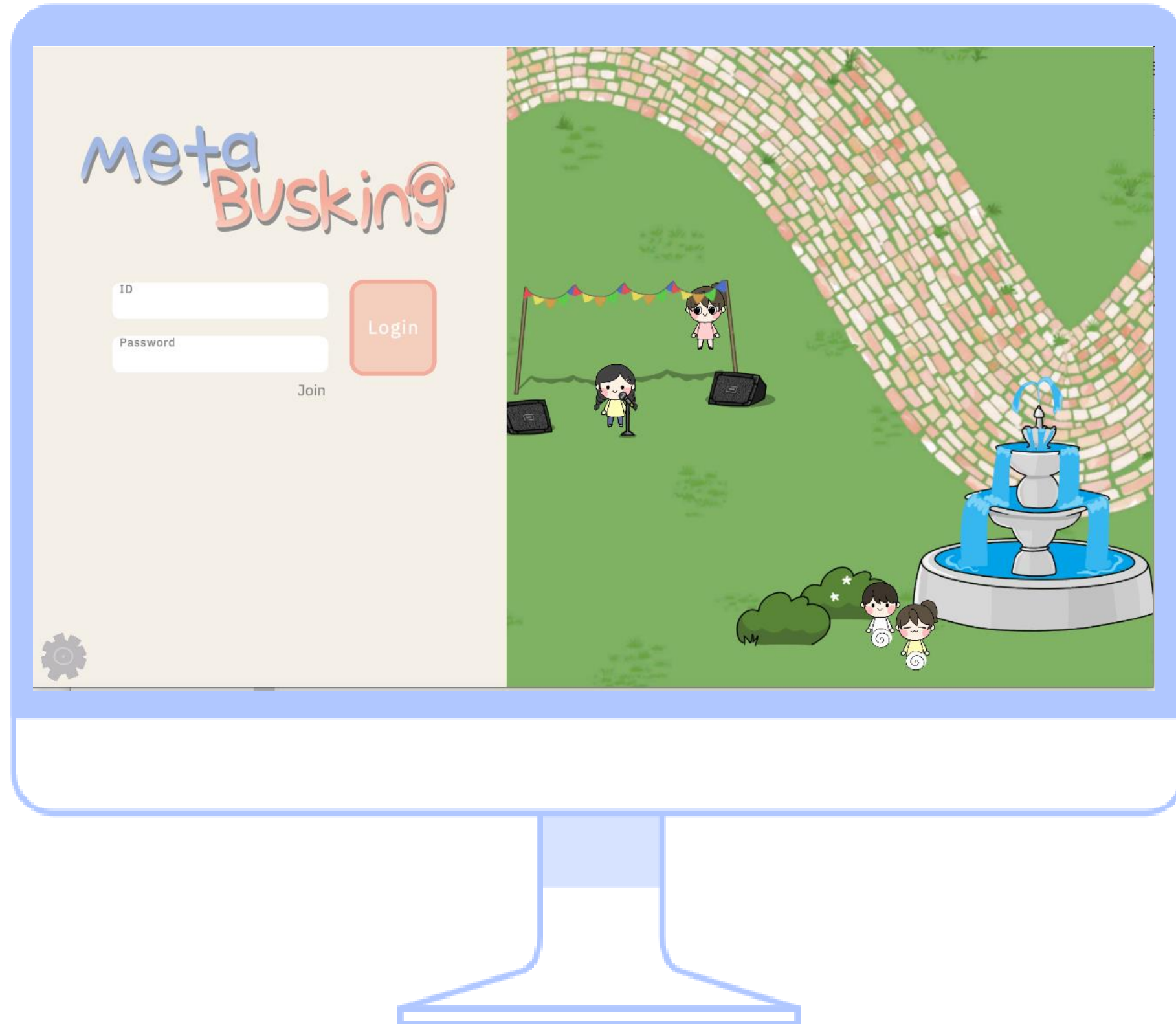


mongoDB®



# 로그인

Metabusking 주요 기능



node.js + express

id password 전송



token 전송

mongodb®



User의 데이터 값  
id, password

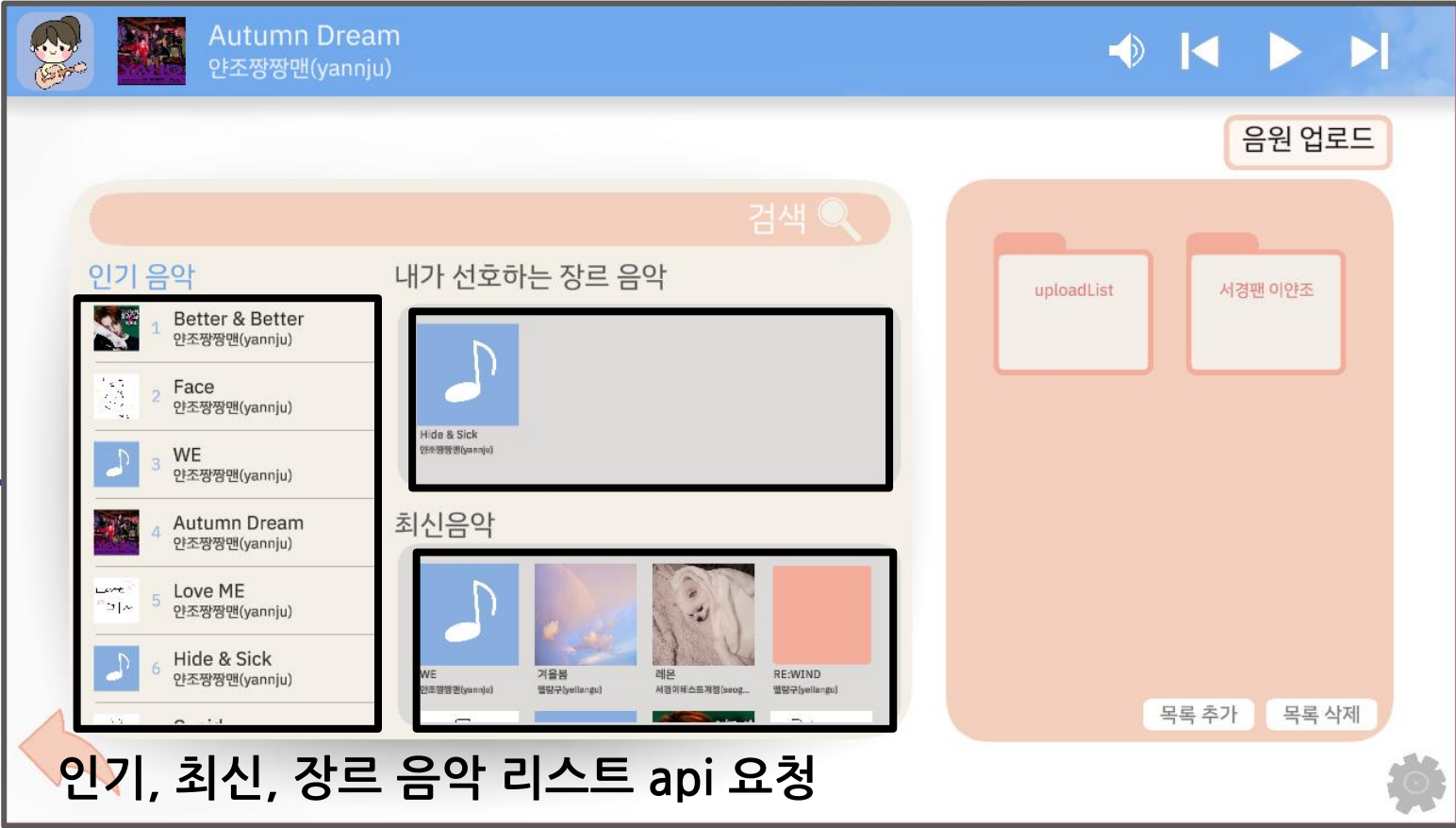


password 비교

토큰 생성

# 캐릭터 설정, 음원 추천

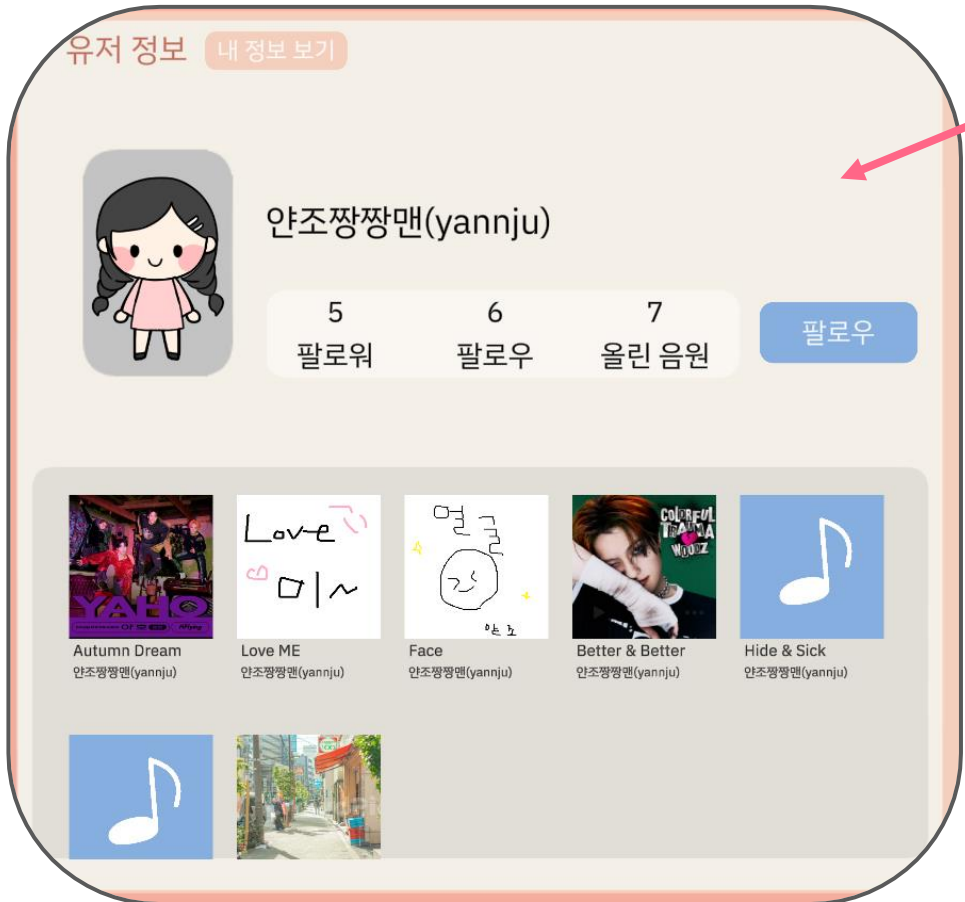
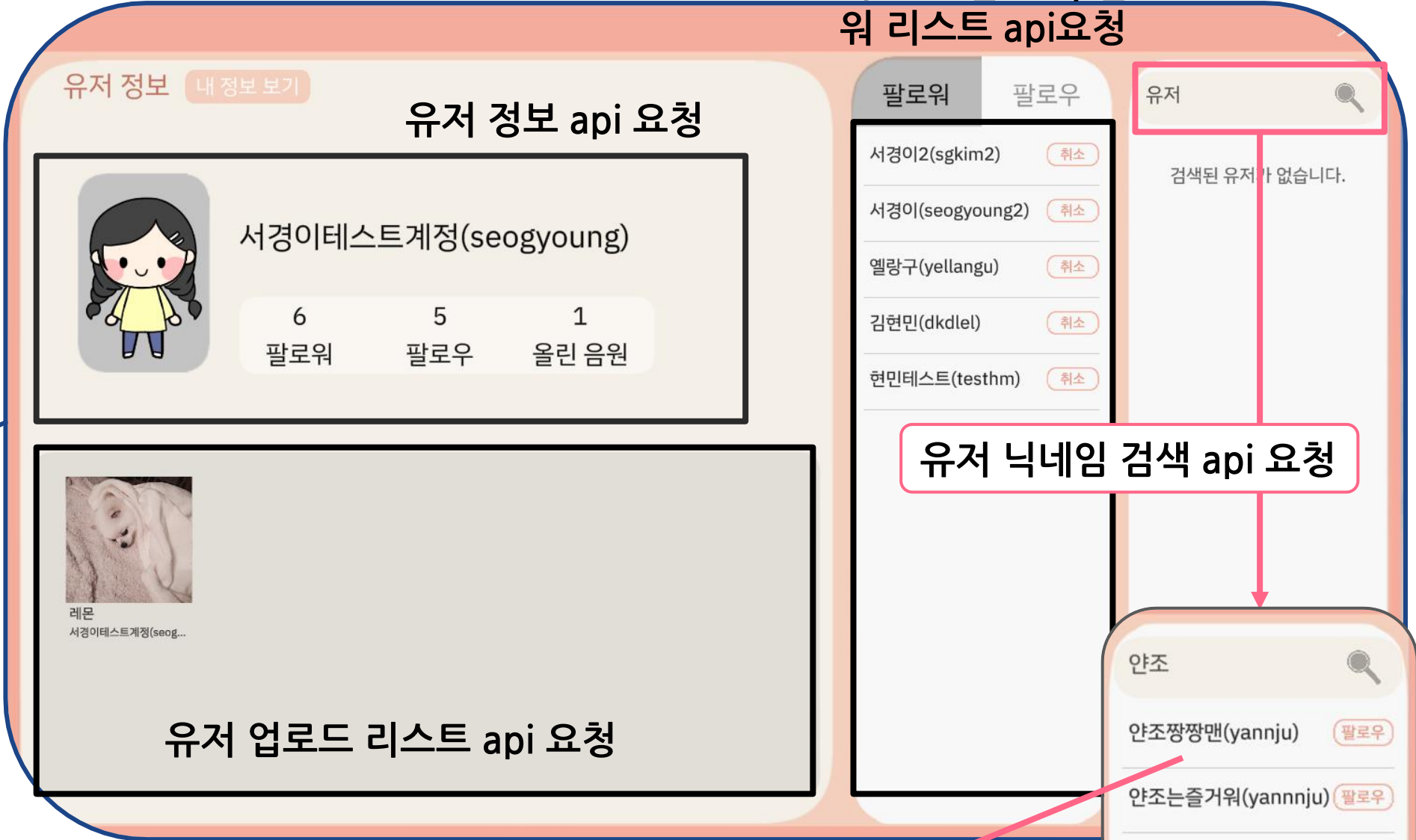
Metabusking 주요 기능



인기, 최신, 장르 음악 리스트 api 요청

# 팔로우 시스템

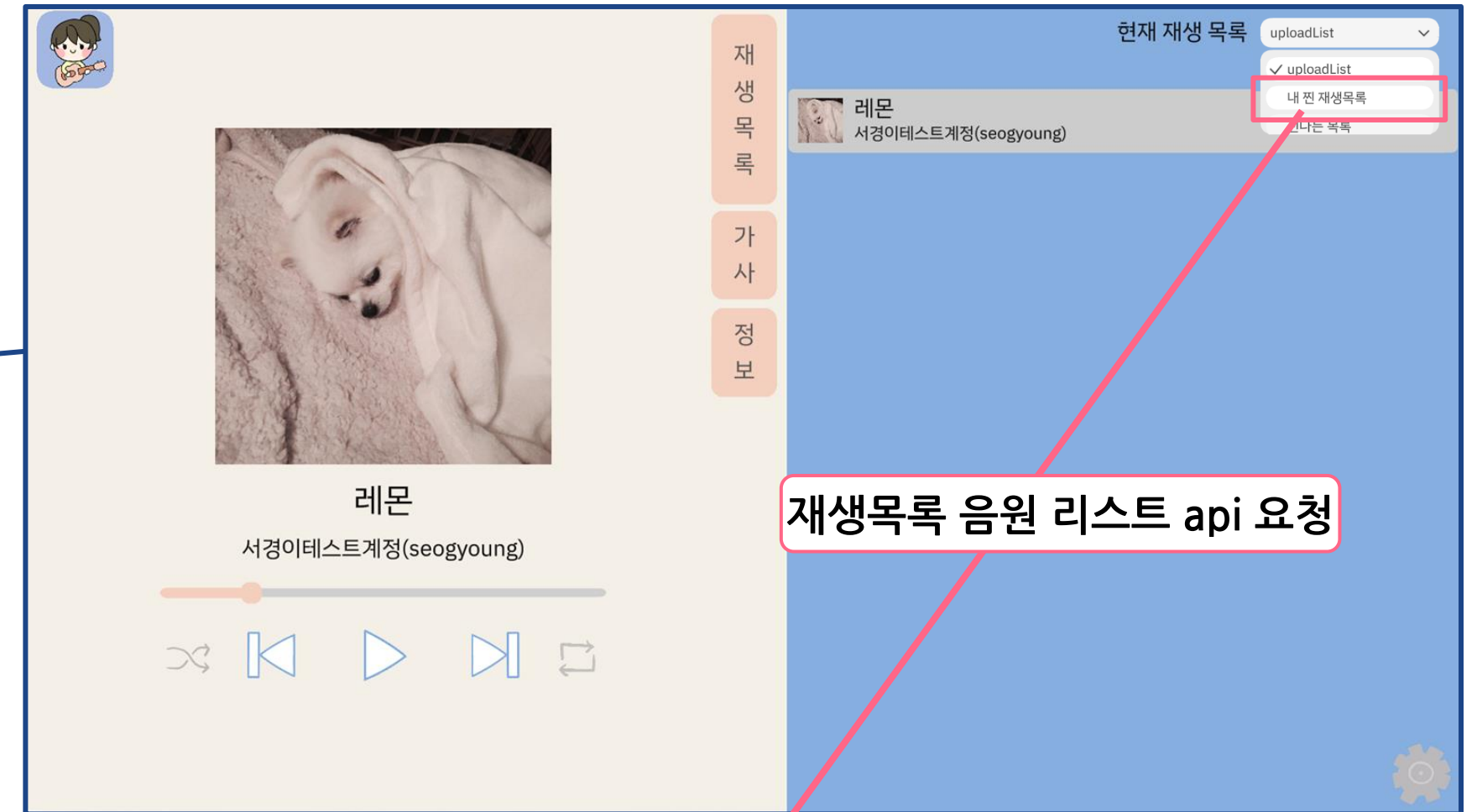
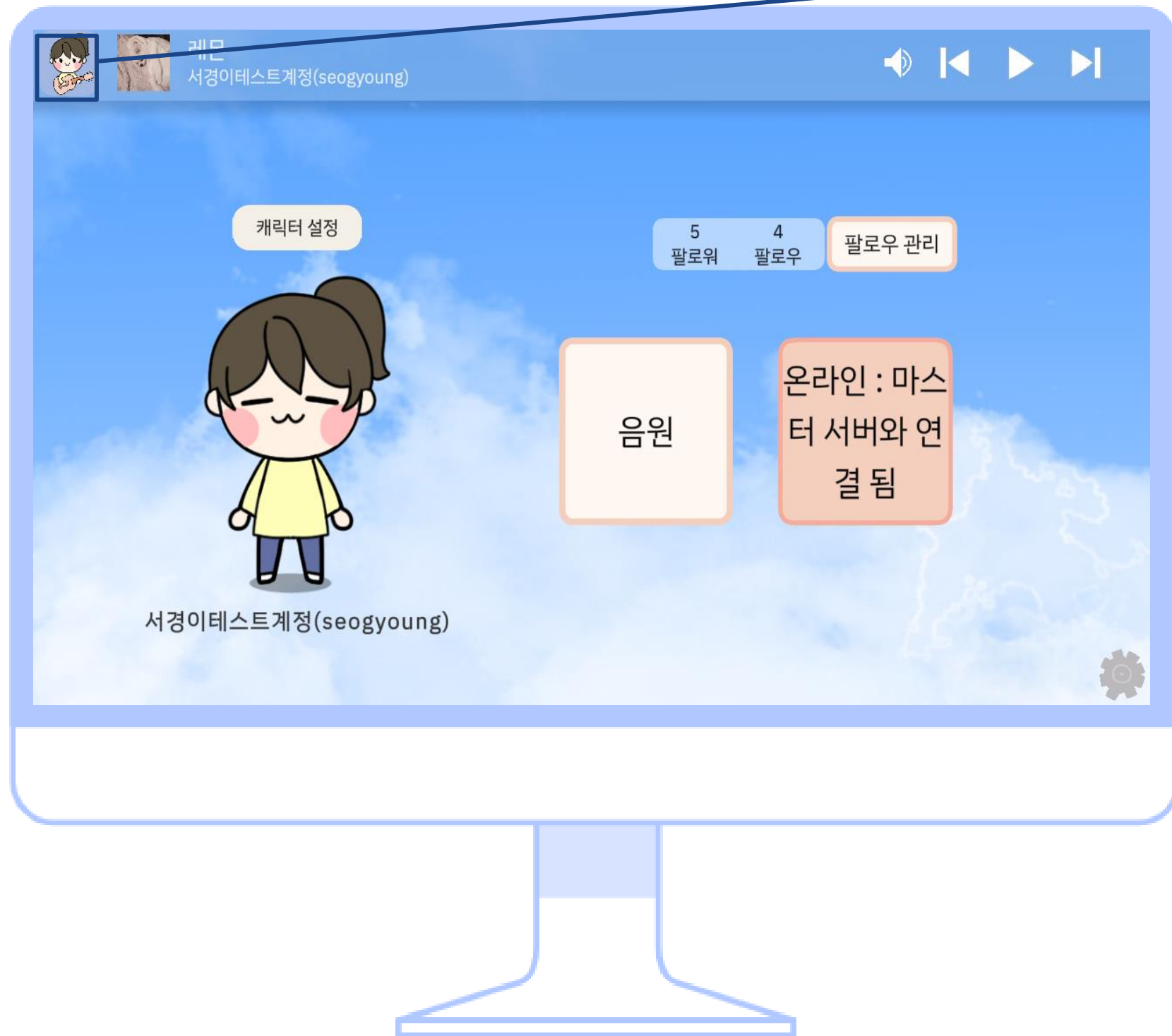
Metabusking 주요 기능





# 음원 재생

Metabusking 주요 기능



재생목록 음원 리스트 api 요청



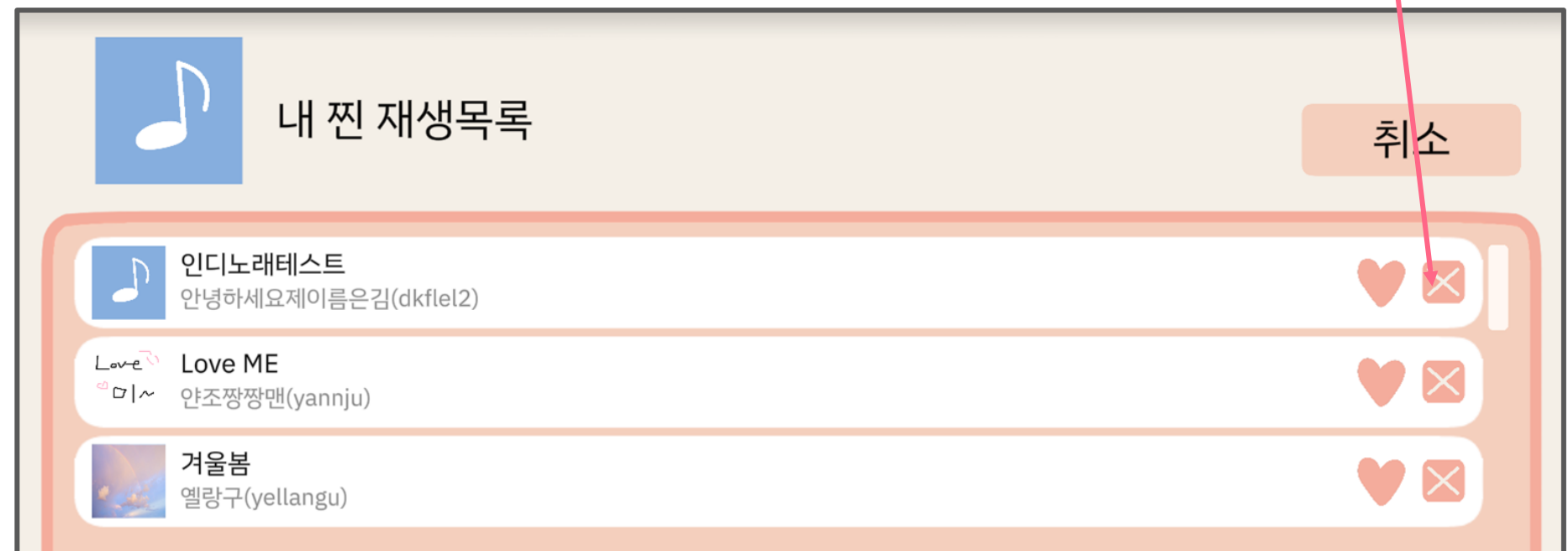
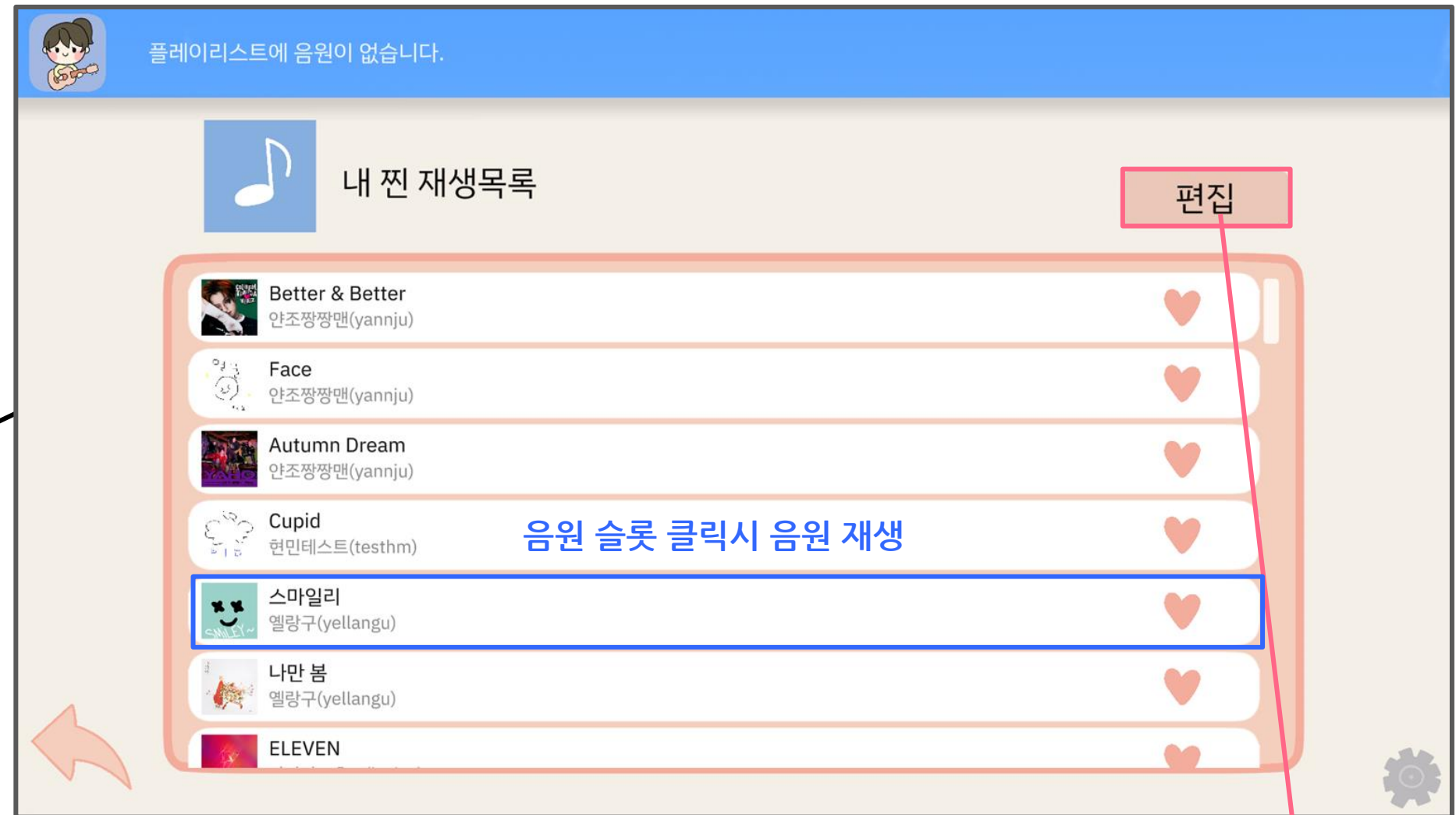
음원 재생 상태 컨트롤 오브젝트

# 음원 관리

Metabusking 주요 기능

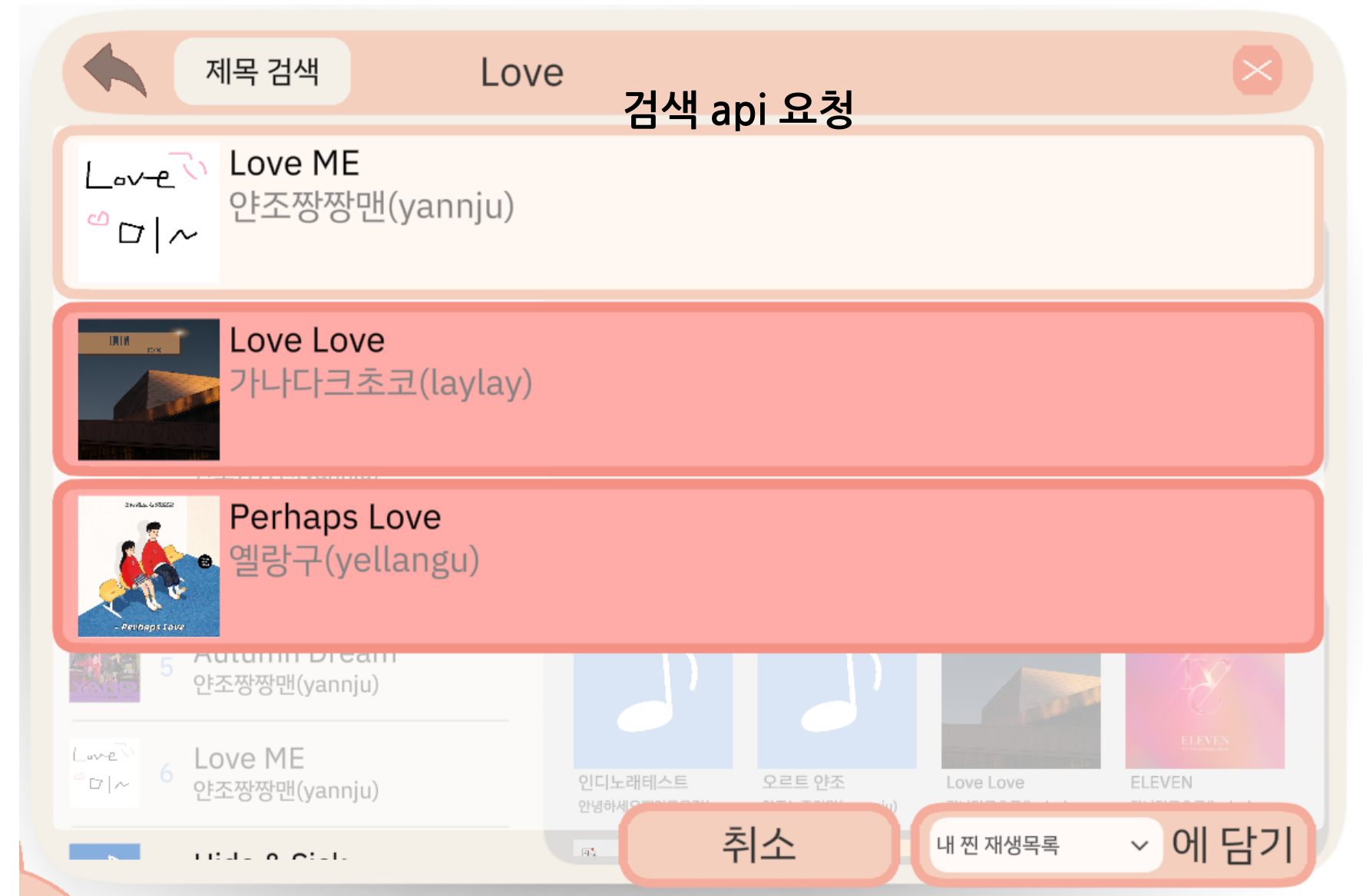
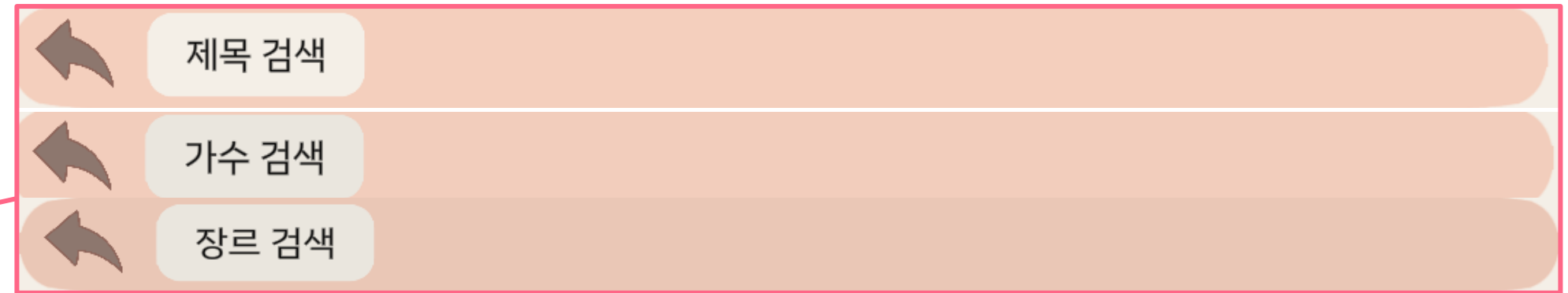


재생목록 음  
원 리스트  
api 요청



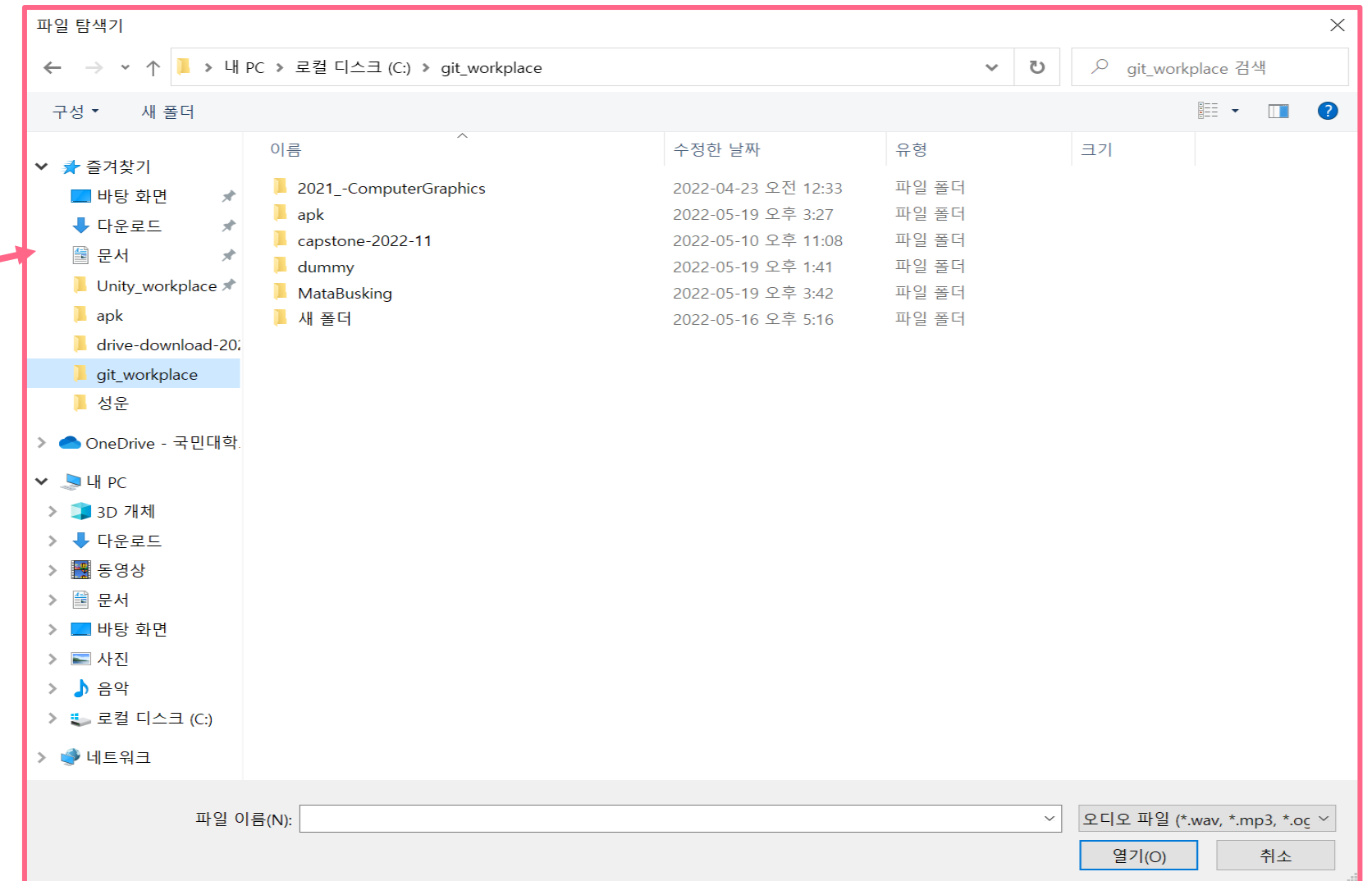
# 음원 검색

Metabusking 주요 기능



# 음원 업로드

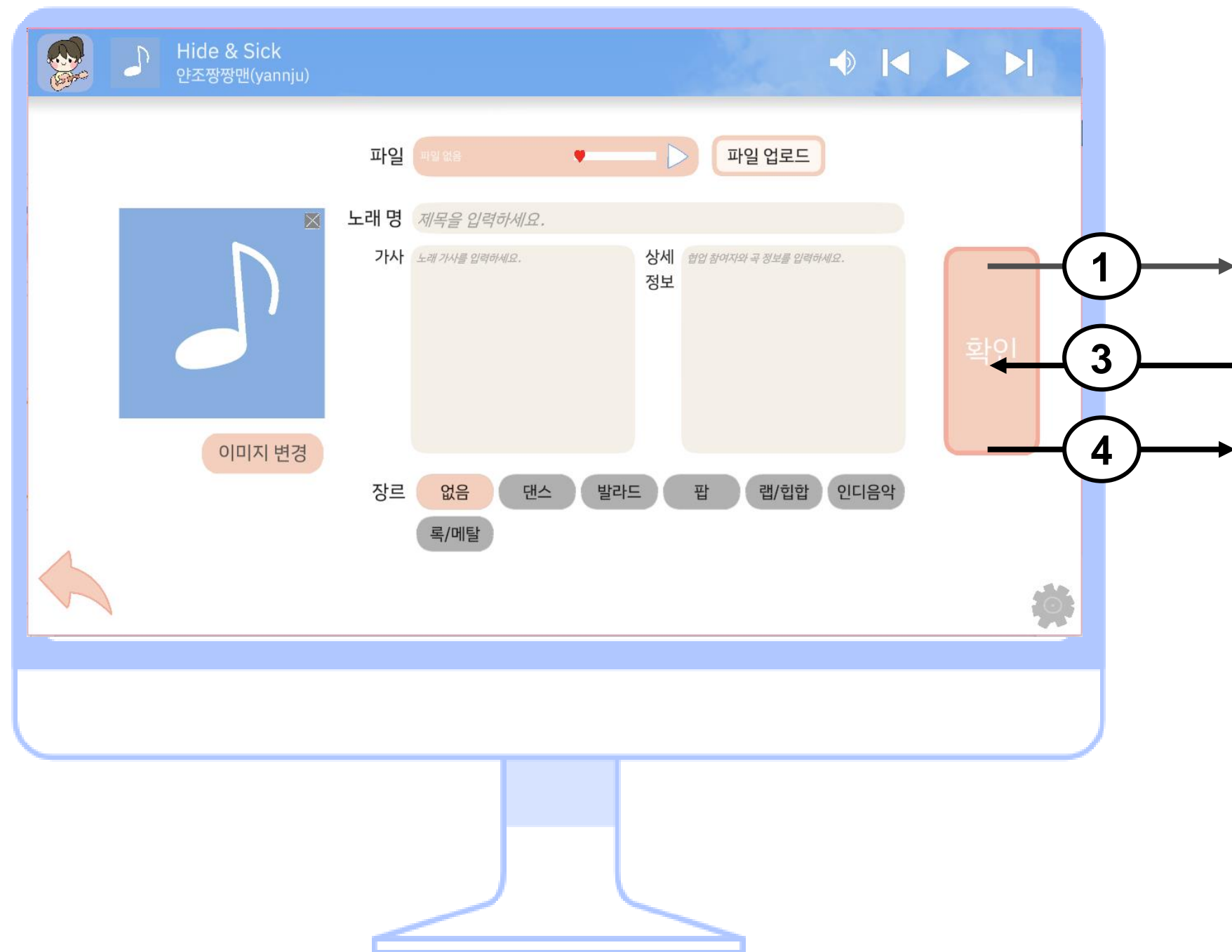
Metabusing 주요 기능



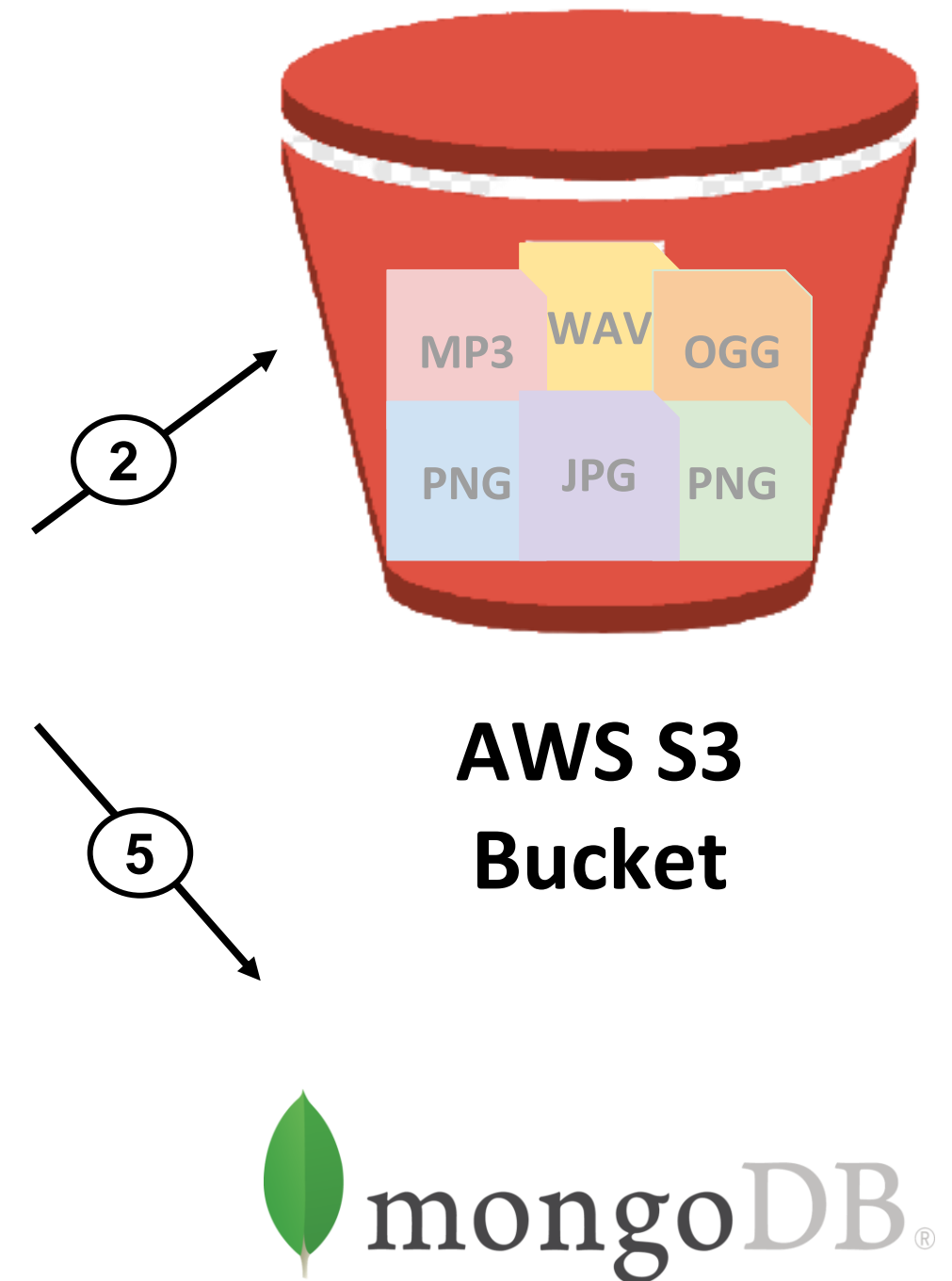


# 음원 업로드

Metabusing 주요 기능



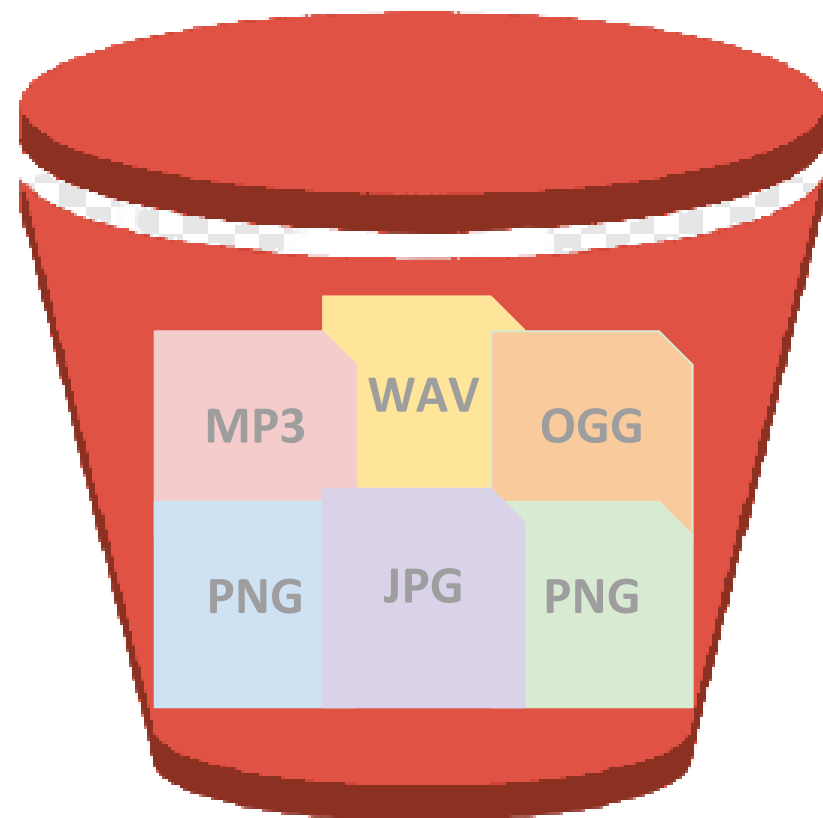
node  
JS  
express



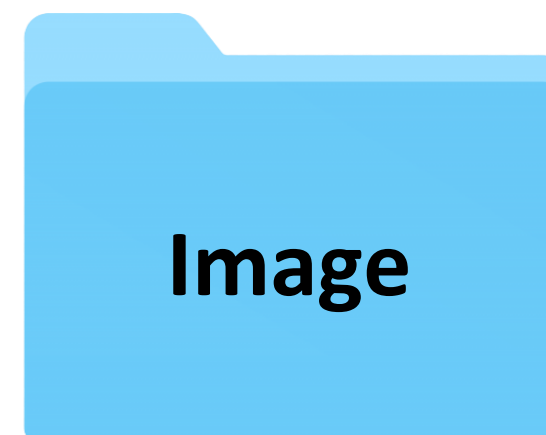
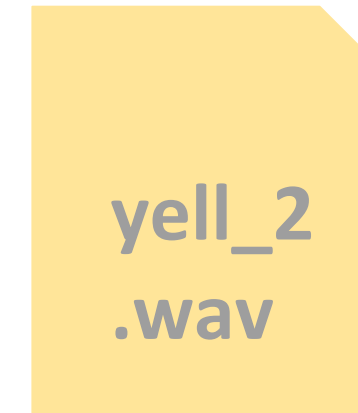
# 음원 업로드

- 파일명은 사용자의 id와 사용자가 올린 음원/이미지 파일의 총 개수로 저장

## 저장 방식



**AWS S3  
Bucket**



# 음원 삭제

- 음원 이미지가 없을 경우(음원 파일 삭제)

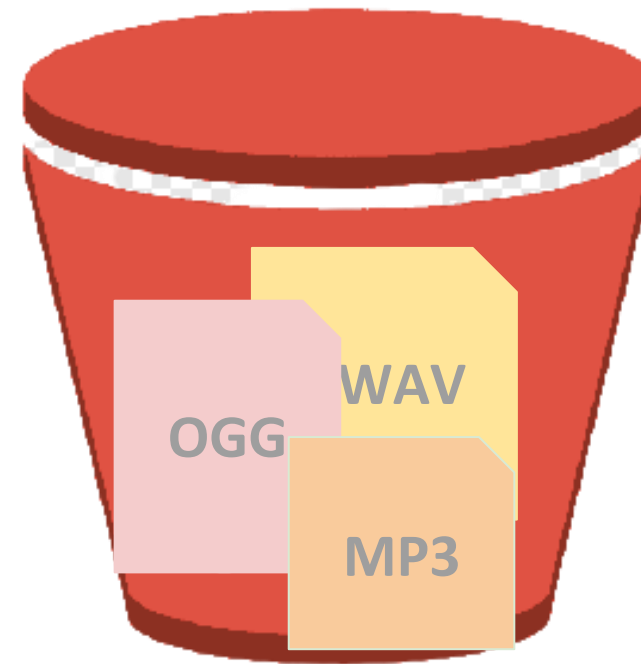
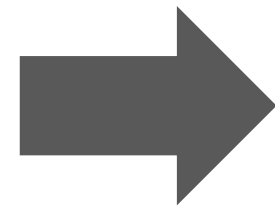


User: yell  
음원: yell\_3.mp3

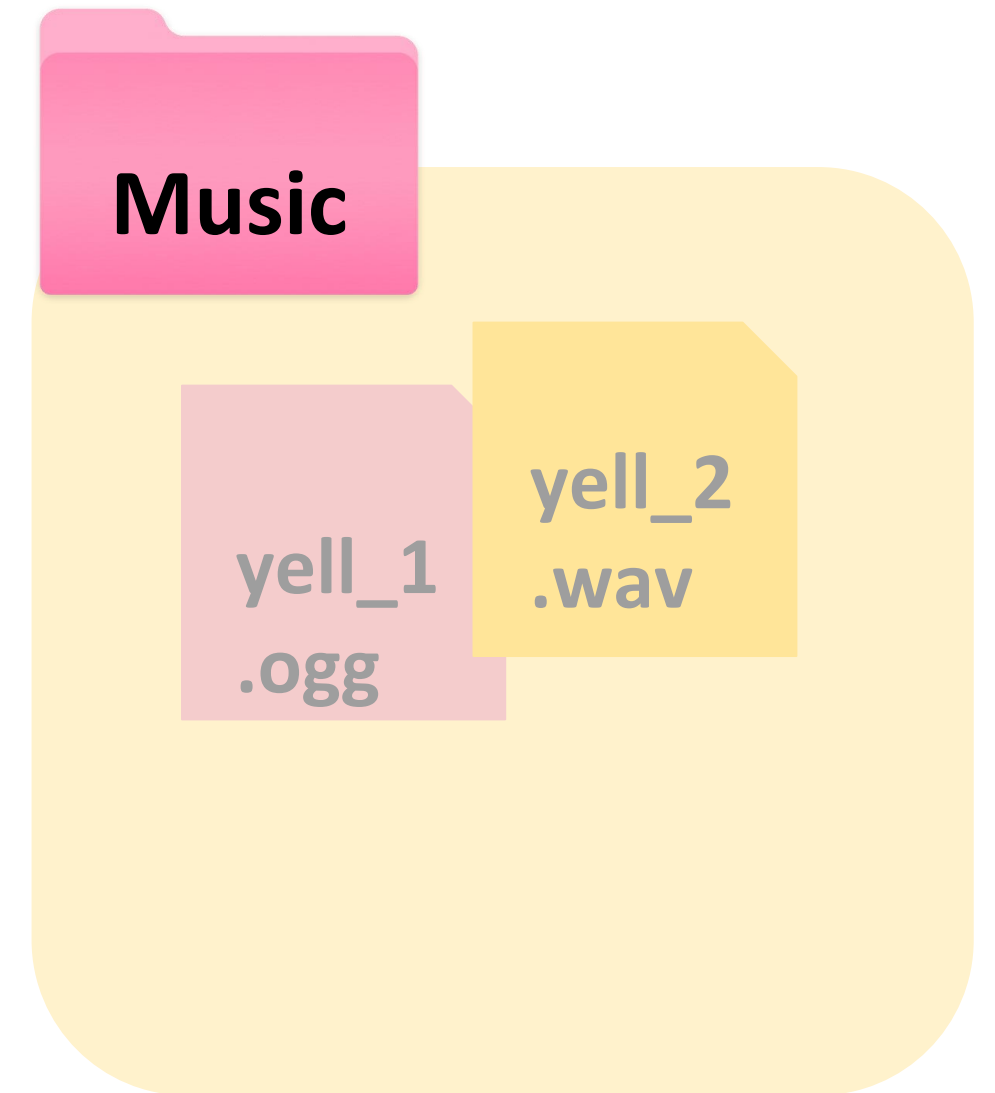
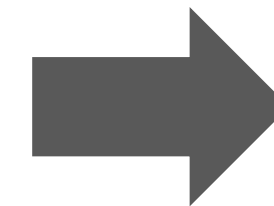
→ 삭제



삭제 전



AWS S3  
Bucket



삭제 후

# 음원 삭제

- 음원 이미지가 있는 경우 (음원, 이미지 파일 삭제)



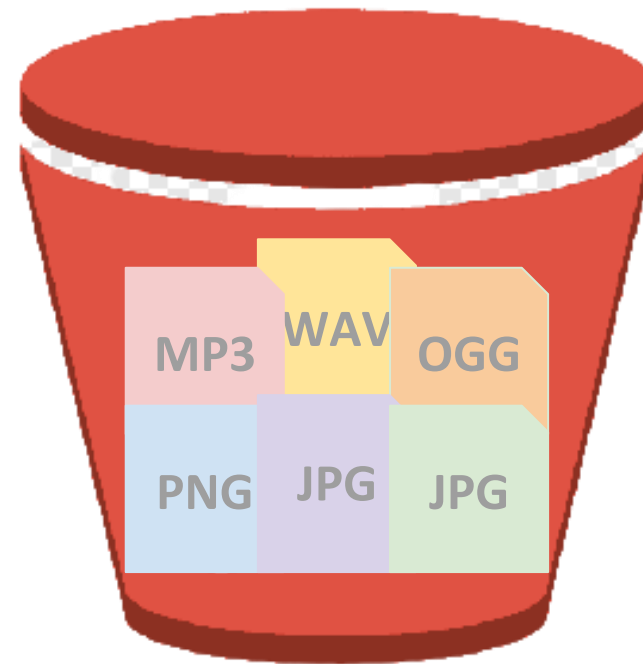
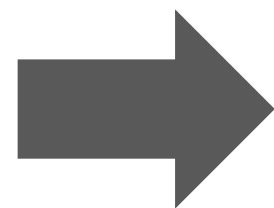
User : yell

음원 : yell\_2.mp3

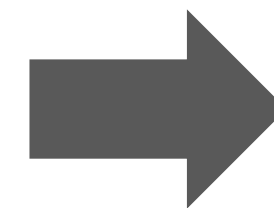
→ 삭제



삭제 전



AWS S3  
Bucket



삭제 후



# 음원 재생



음악 파일 경로 요청



음악 파일 경로 전달



AWS S3 Bucket에 업로드된 파일 읽기  
byte 배열 -> AudioClip 변환



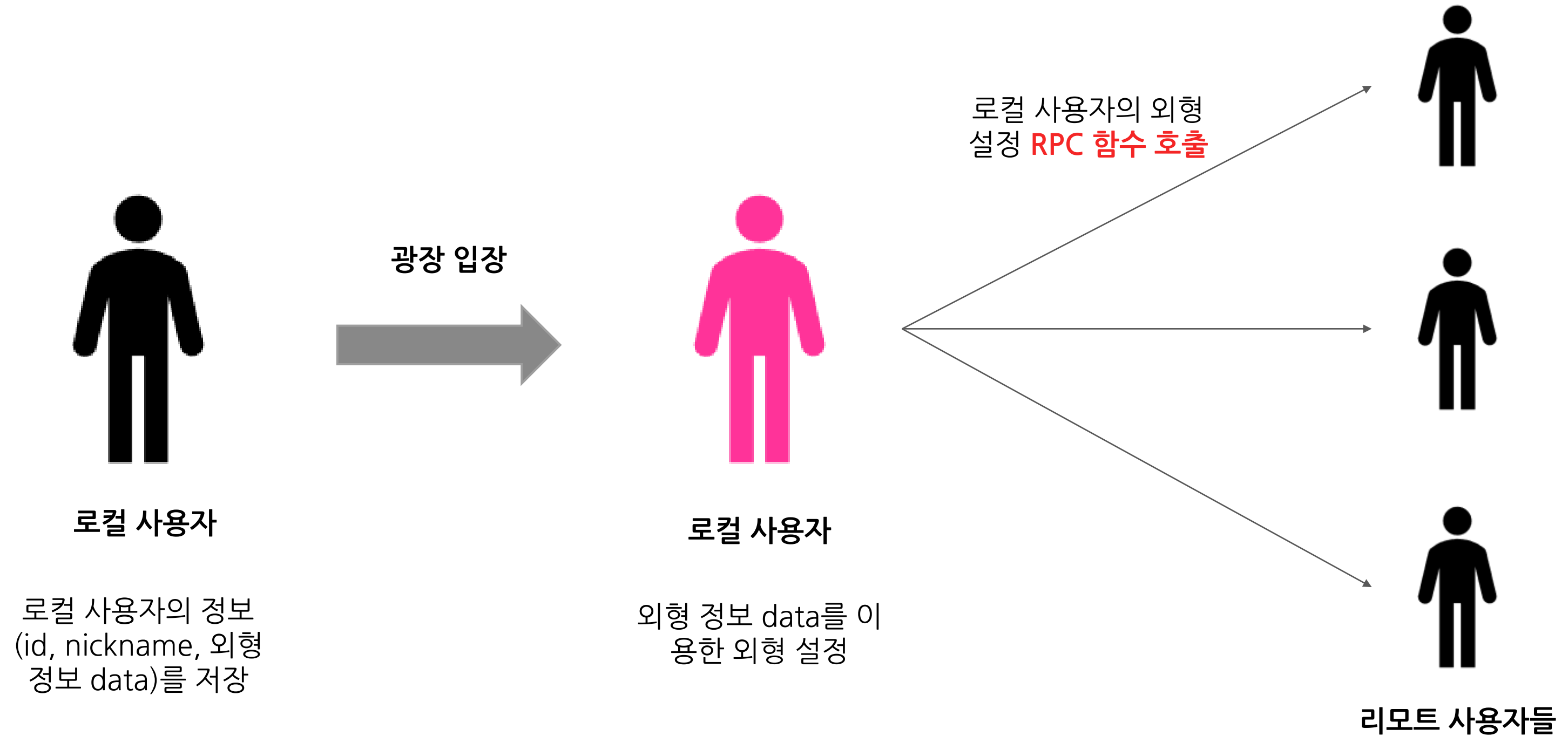
AWS S3 Bucket에 업로드된  
파일 주소 접근



**AWS S3  
Bucket**

# 멀티플레이

Metabusing 주요 기능



# 멀티플레이

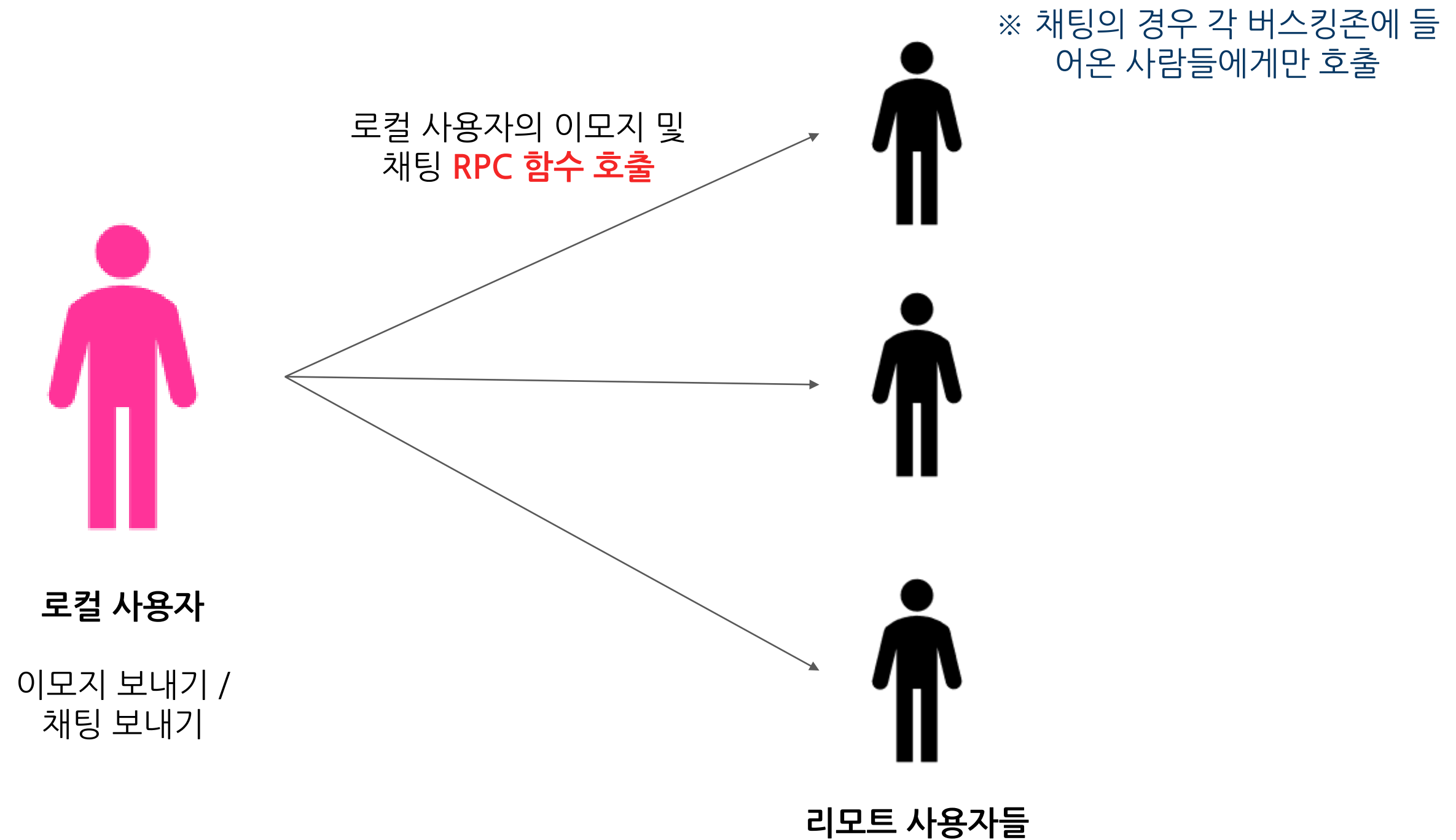
*Metabusking* 주요 기능



사용자 위치 동기화 + 사용자 애니메이션 동기화 + 사용자 외형 동기화

# 멀티플레이

Metabusking 주요 기능



# 멀티플레이

Metabusking 주요 기능



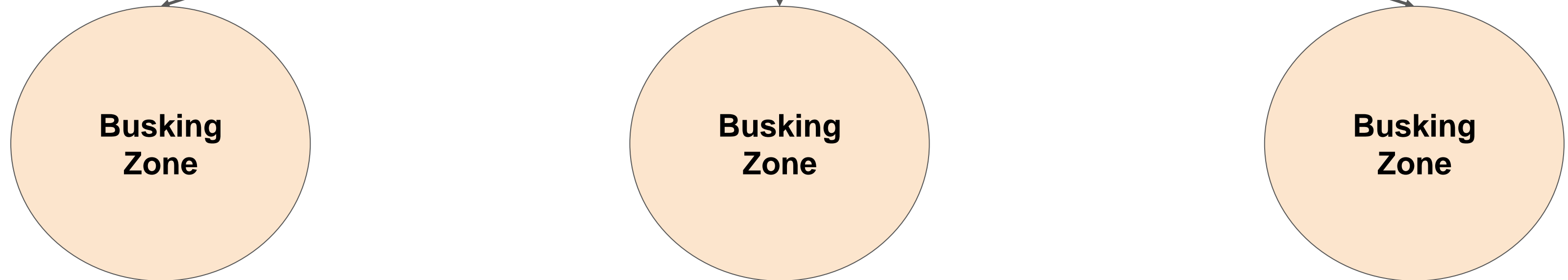
RPC 함수 호출을 통한 이모지 동기화 및 버스킹존마다 채팅 동기화

# 버스킹

*Metabusking* 주요 기능

## agora

multi Channel 방식

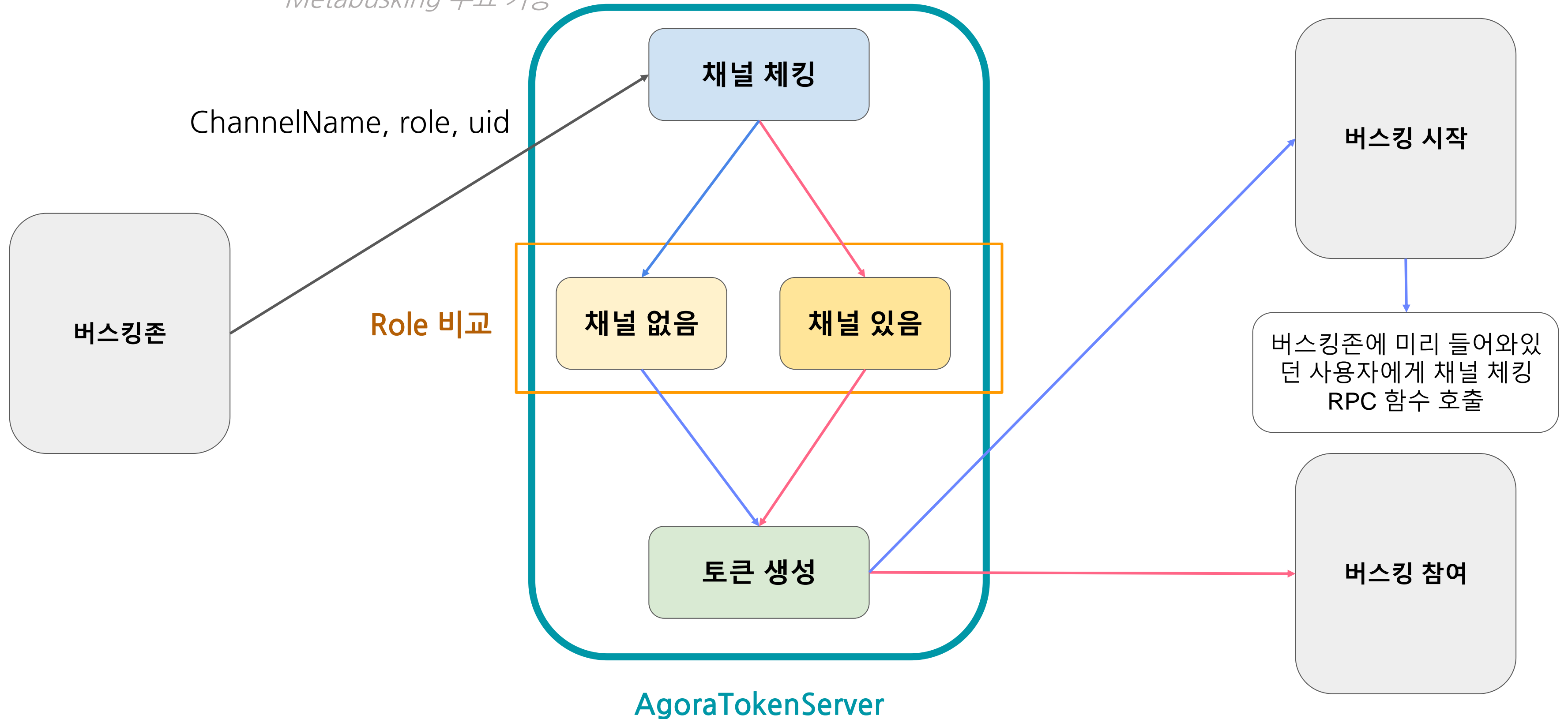


버스킹 존마다 AgoraChannel 생성

# 버스킹

청중  
버스커

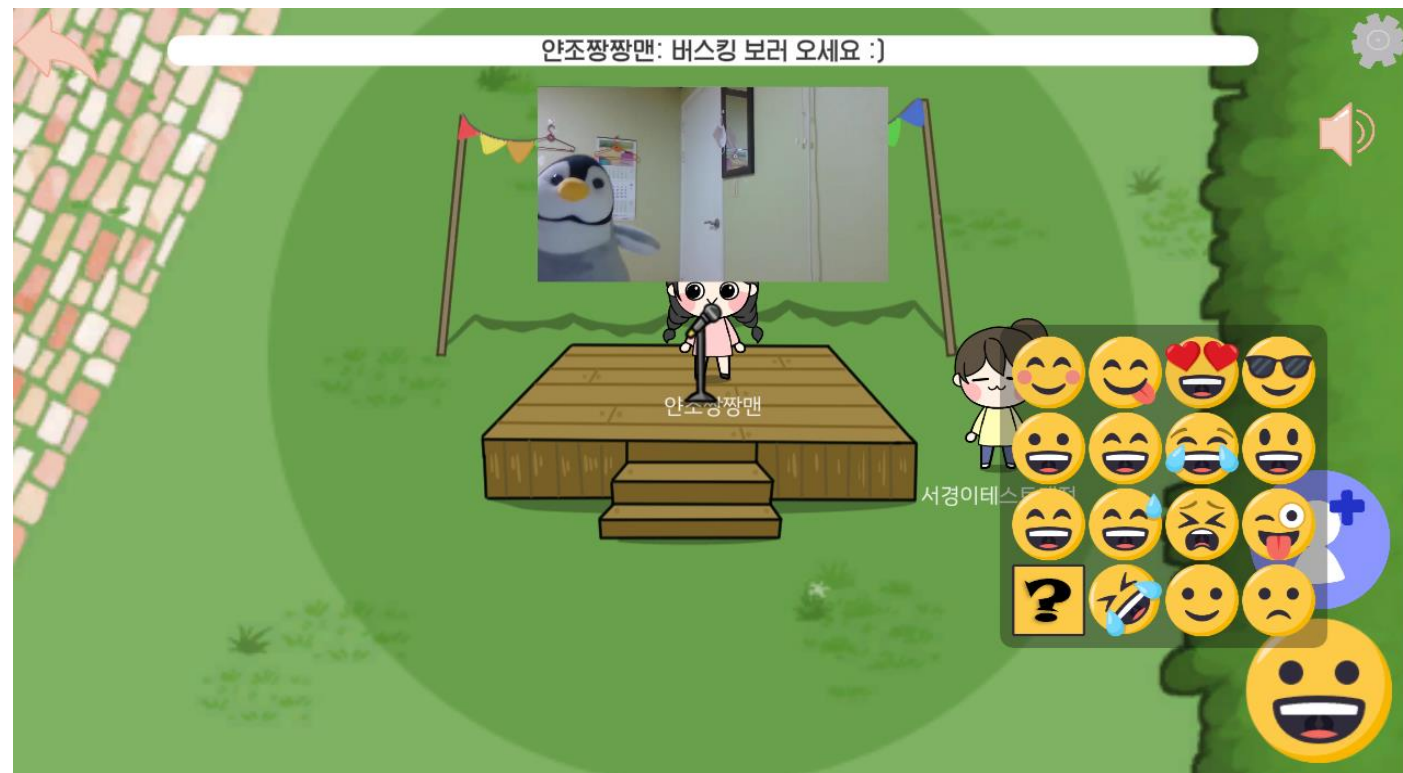
Metabusking 주요 기능





# 버스킹

Metabusking 주요 기능



작은 화면

화면 클릭



큰 화면



# 음원존

Metabusking 주요 기능

음원존 생성



Zone 번호, 음악들의 제목, 위치, 시간 리스트

성공: 200 OK  
실패: 410 이미 존재

node.js + express + mongoose  
elegant mongodb object modeling for node.js

new Zone 생성

```
{  
  제목리스트,  
  위치리스트,  
  시간리스트,  
  시작 시간,  
  총시간,  
  zone 번호  
}
```

저장

mongoDB®

```
{  
  musicTitle,  
  musicList,  
  musicTime,  
  totalTime,  
  startTime,  
  id : Zone번호  
}
```

# 음원존

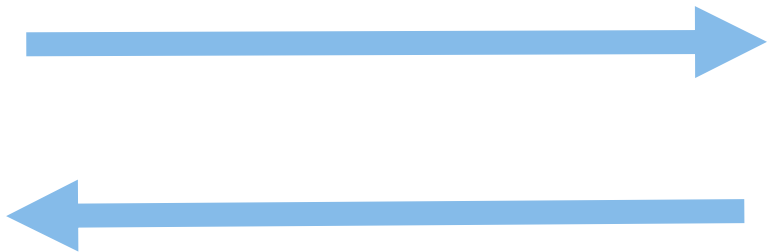
Metabusking 주요 기능



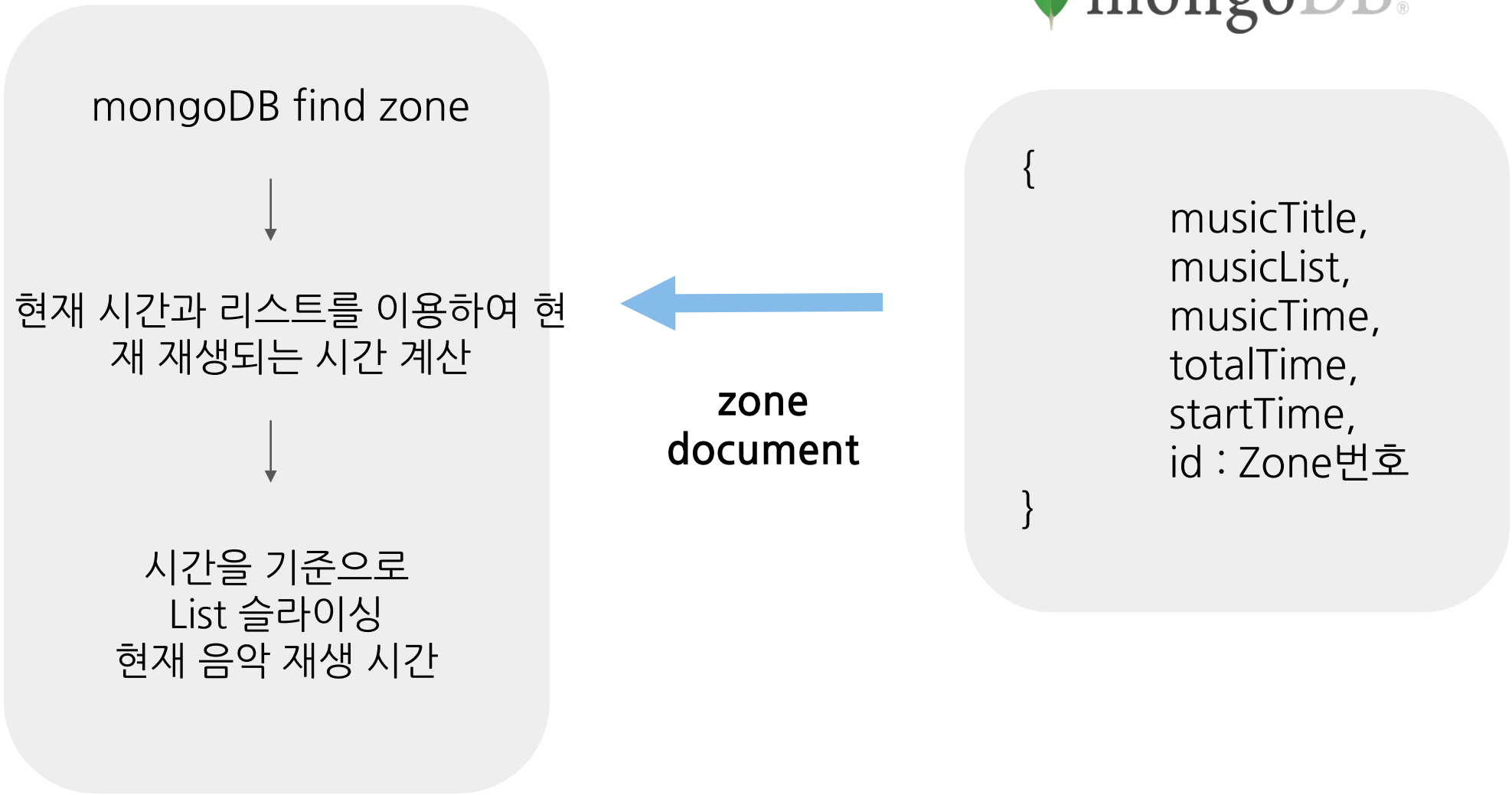
음원존 참가(영역 내에 입장)



GET 요청, zone 번호



성공 시: 남은 음악들, 현재 재생음악 시간  
실패 시: 510에러, 재생되는 음악 없음



Chapter.3

# 기대 효과

03

---

# 기대효과

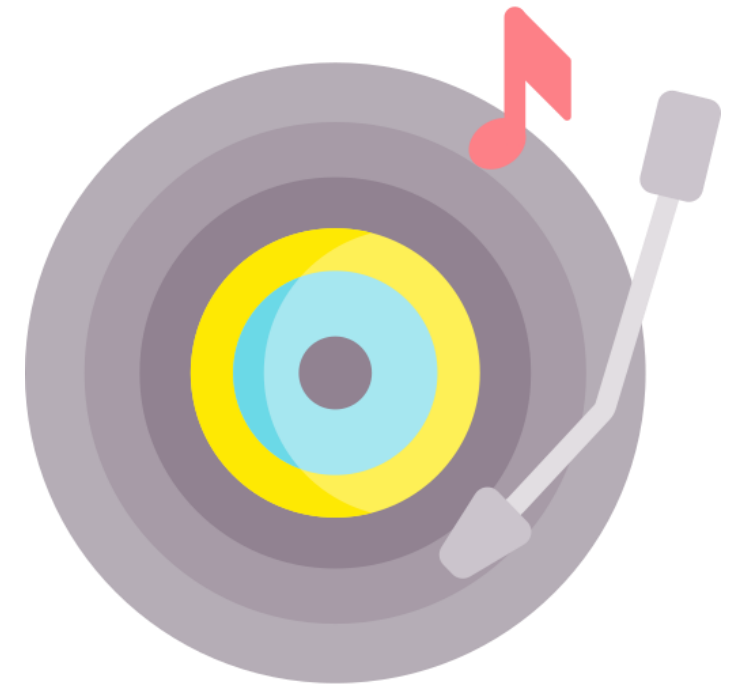
---



온라인 공연 확대



아티스트 홍보



버스킹 접근성 증가

Chapter.4

# Next Plan

추후 계획

04

---

# Next Plan

안드로이드 배포 확장 및 고음질 지원

## Android App

- 버스킹 기능 제외 모든 기능 적용완료
- 추후 버스킹 기능도 최적화할 예정

## Music Streaming

- 현재 스트리밍 방식은 음원 용량의 크기에 따라 속도 차이가 많이나서 음원의 용량을 제한하다보니 고음질의 파일을 듣기 힘들다.  
=> 분할 처리를 이용한 Stream Data 전송을 통한 속도 개선

Chapter.5

# 시연 영상

05

---

**감사합니다**