

2022

Capstone Project

최종 발표 - 11조

CONTENTS. |

01

시연 영상

02

프로젝트 개요

Metabusking

03

주요 기술 및 기능

04

기대 효과

05

Next Plan

추후 계획

Chapter.1

시연 영상

01

시연 영상



Chapter.2

프로젝트 개요

Metabusking

02

MetaBusking

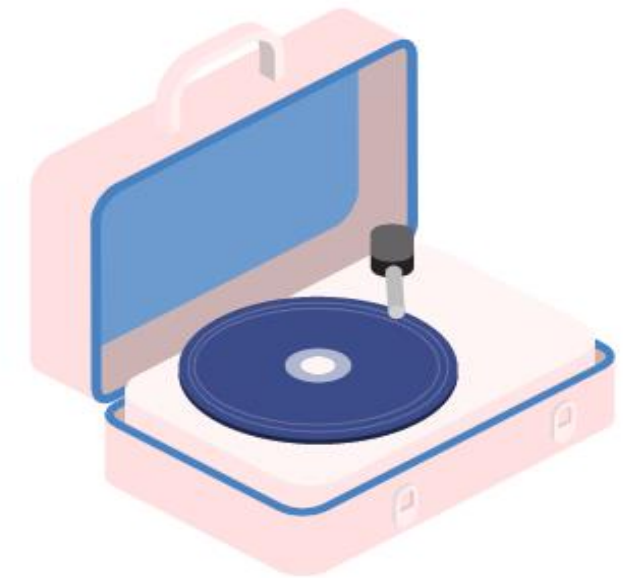
온라인 버스킹 및 음원 공유 플랫폼



메타버스 가상 공간에서



실제 버스킹처럼
다양한 사람들과 함께하는 버스킹

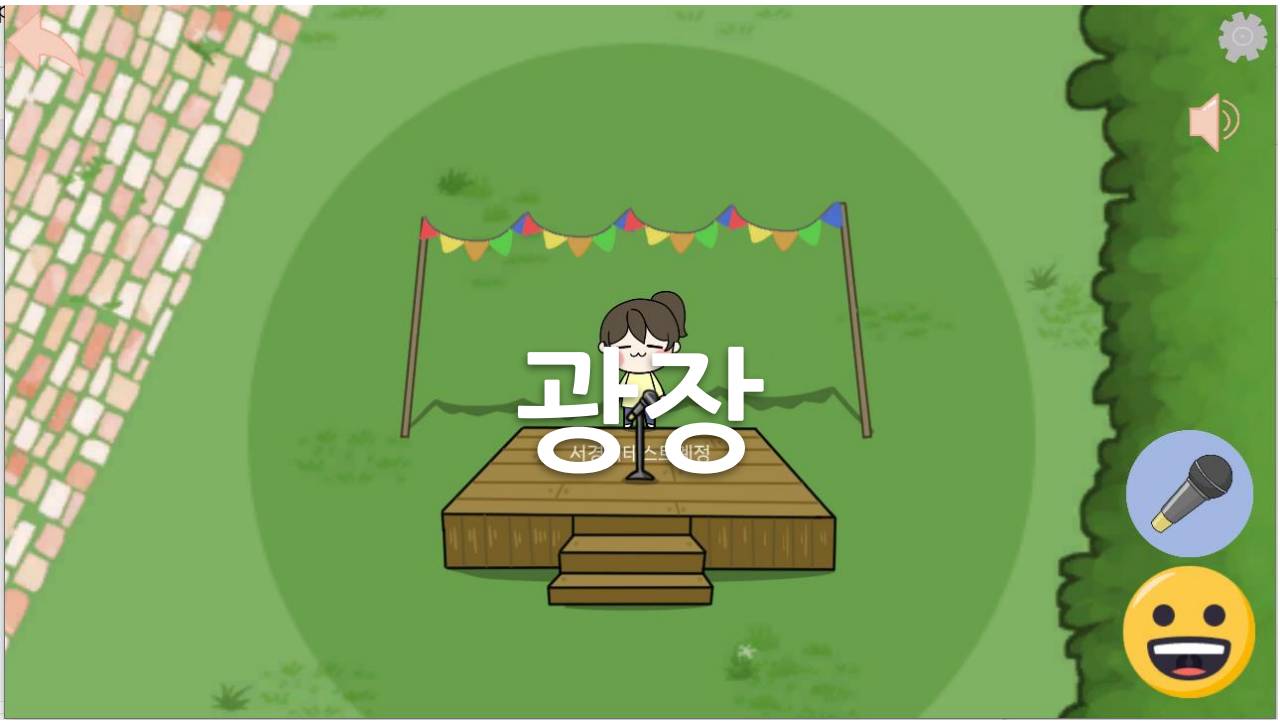
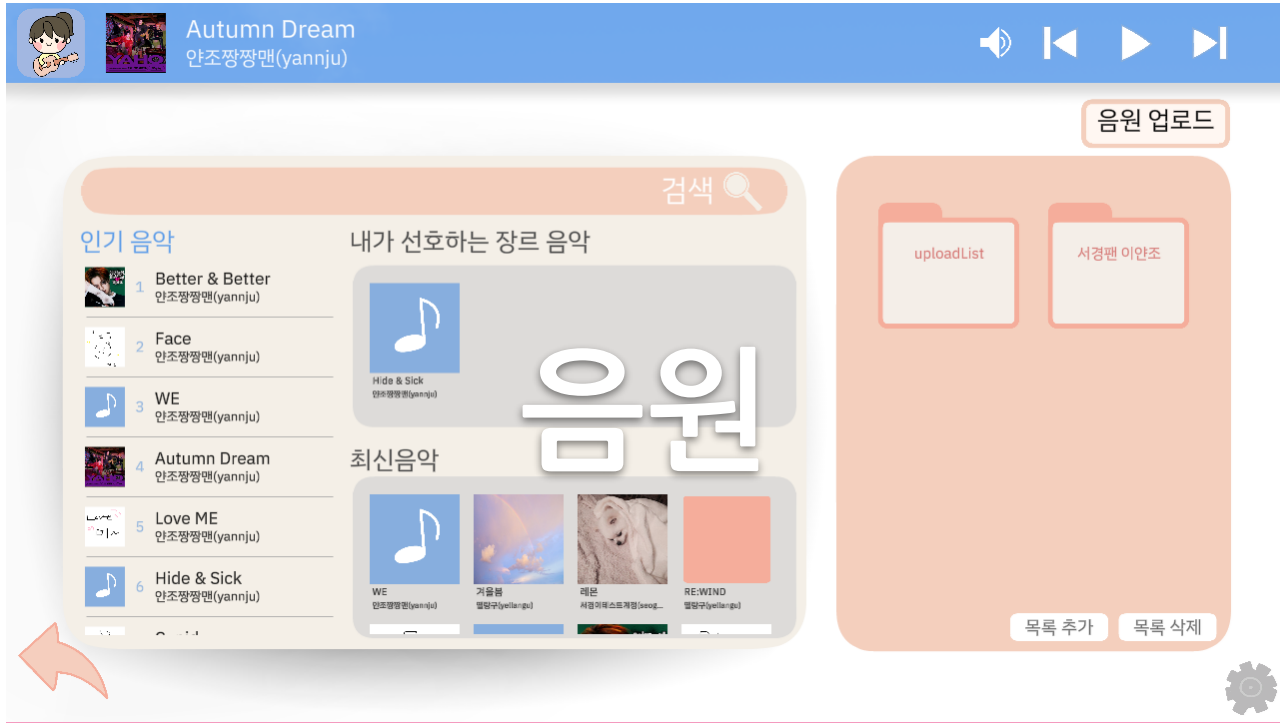


관심있는 음악을 바로 감상할 수 있는
스트리밍 서비스



프로젝트 소개

Metabusking



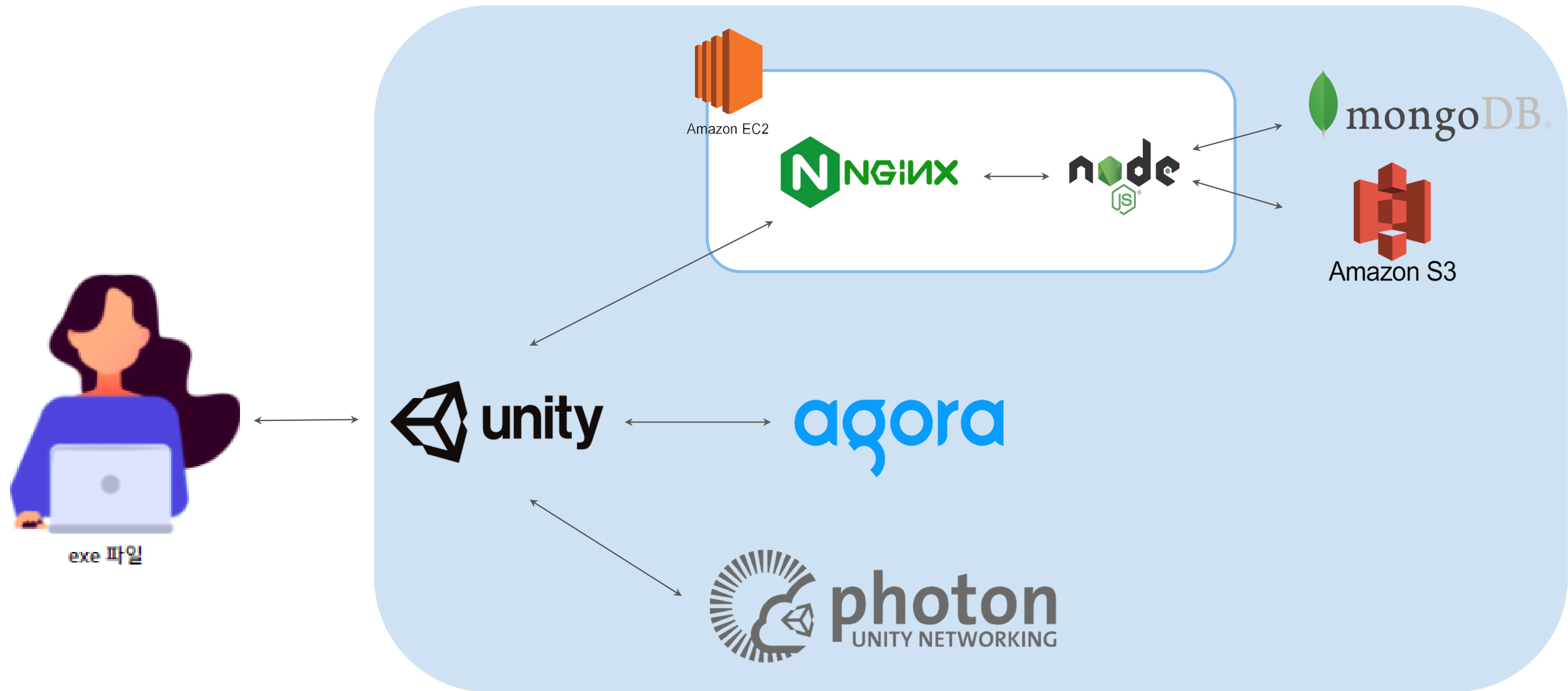
03

주요 기술 및 기능

주요 기술

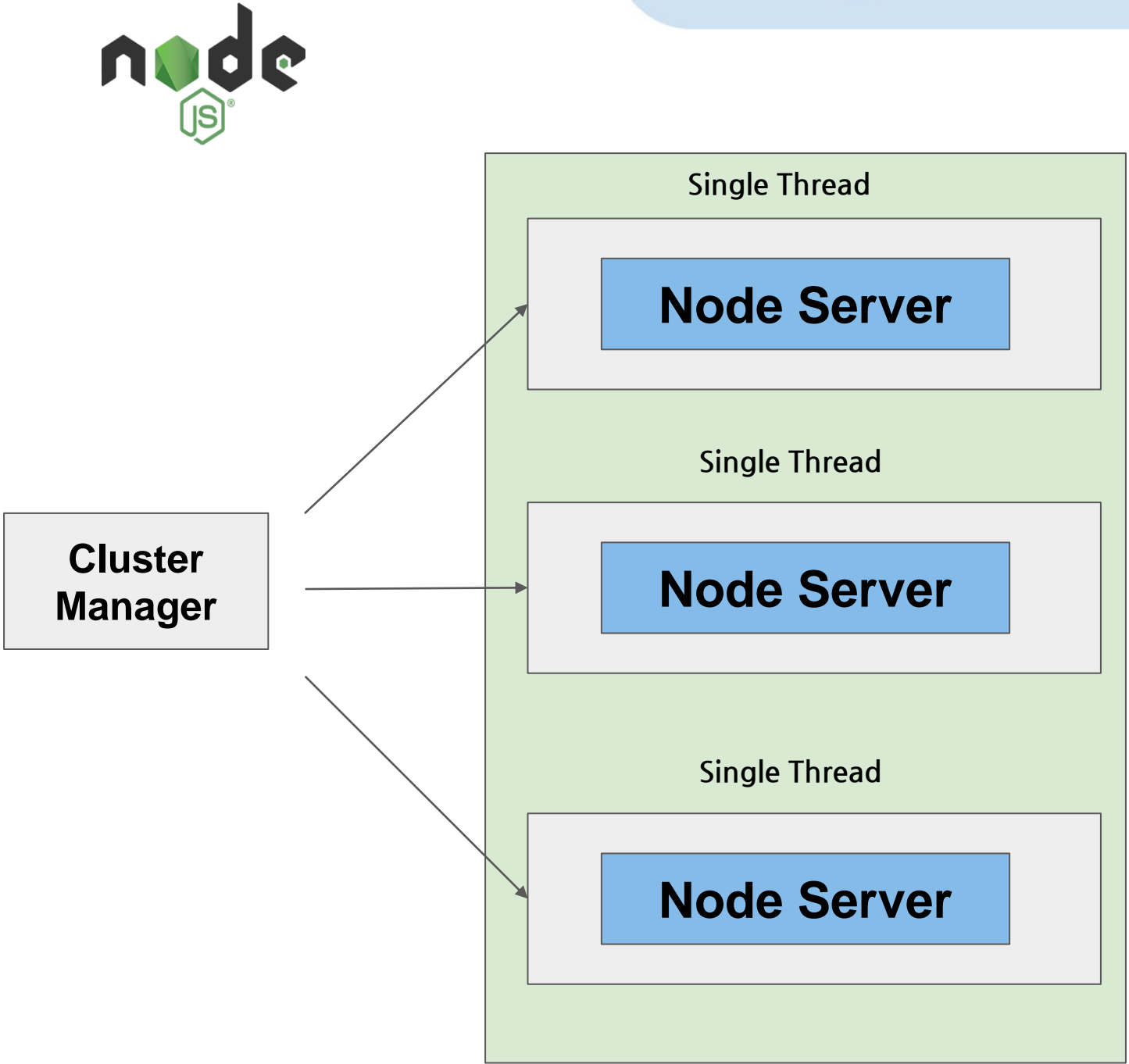
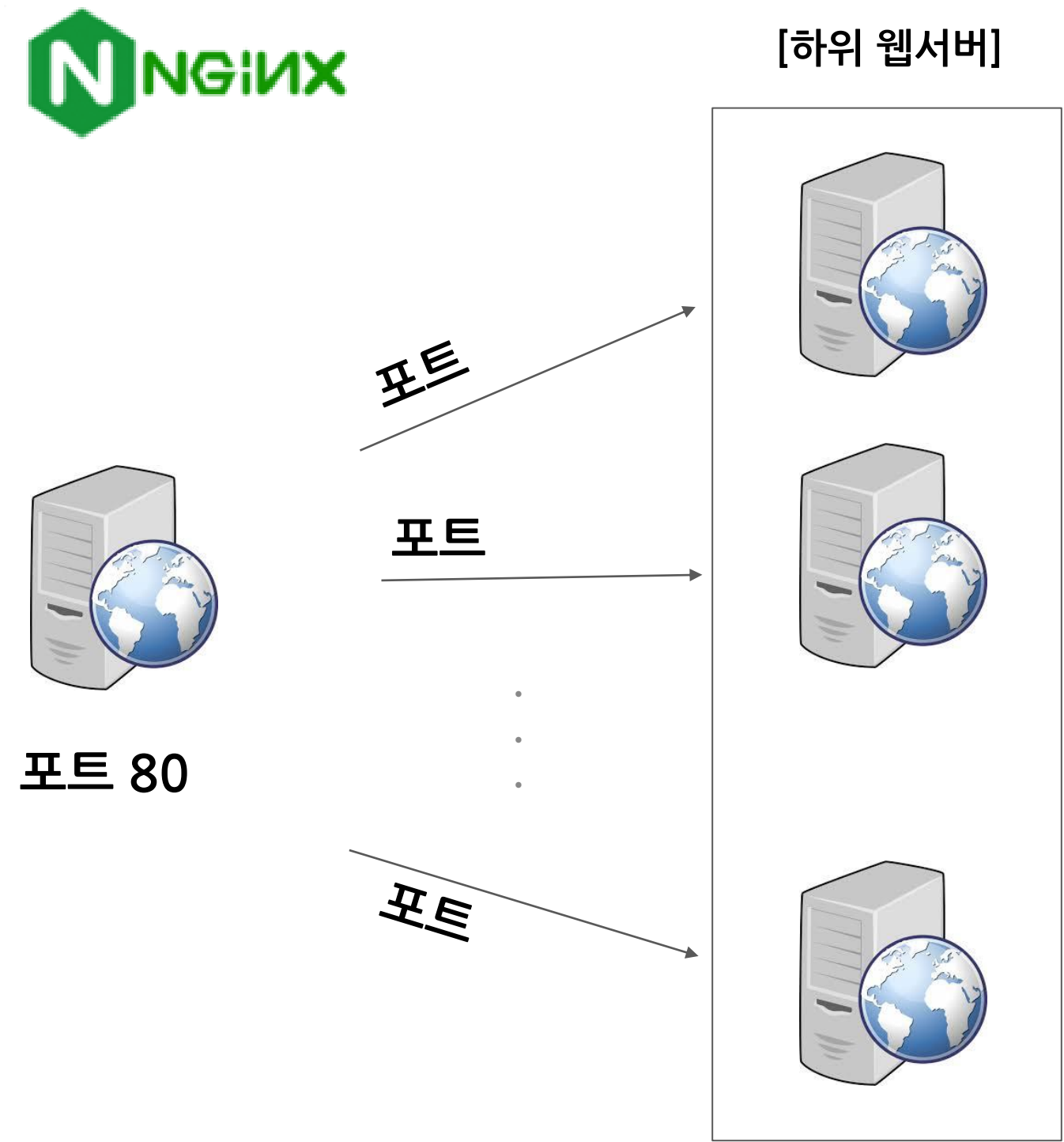
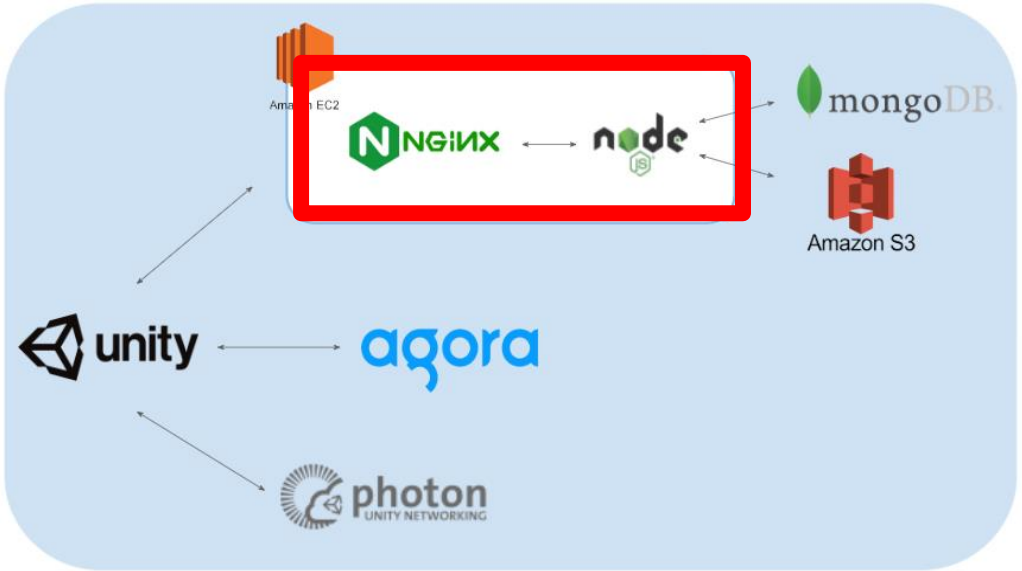
시스템 구성도

소프트웨어 아키텍처



Nginx와 NodeJS

Metabusing 주요 기술



라이브 스트리밍

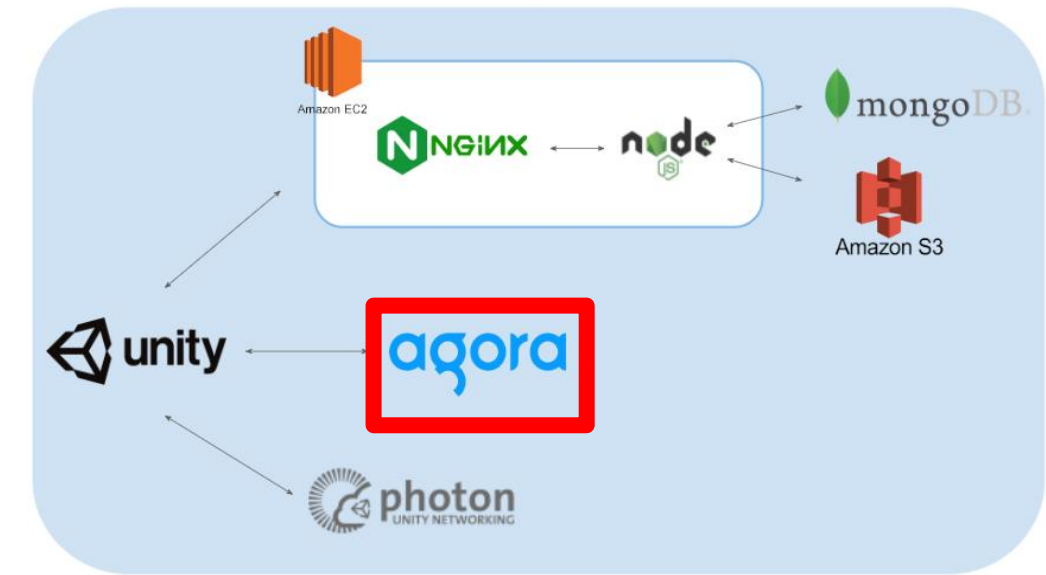
Metabusing 주요 기술

기존 기술(WebRTC) 변경 이유

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

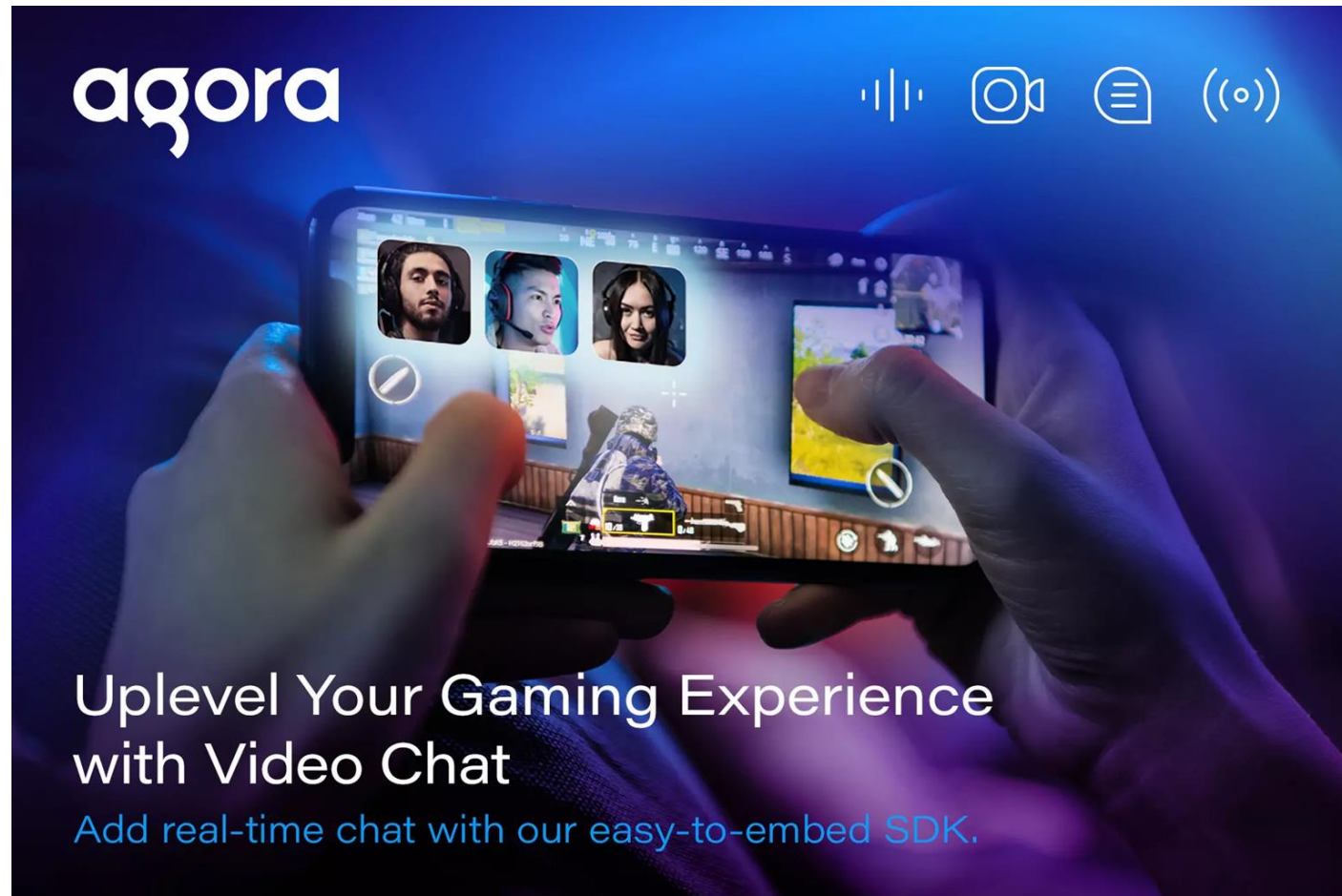


새로운 기술 (Agora)을 적용하기로 결정

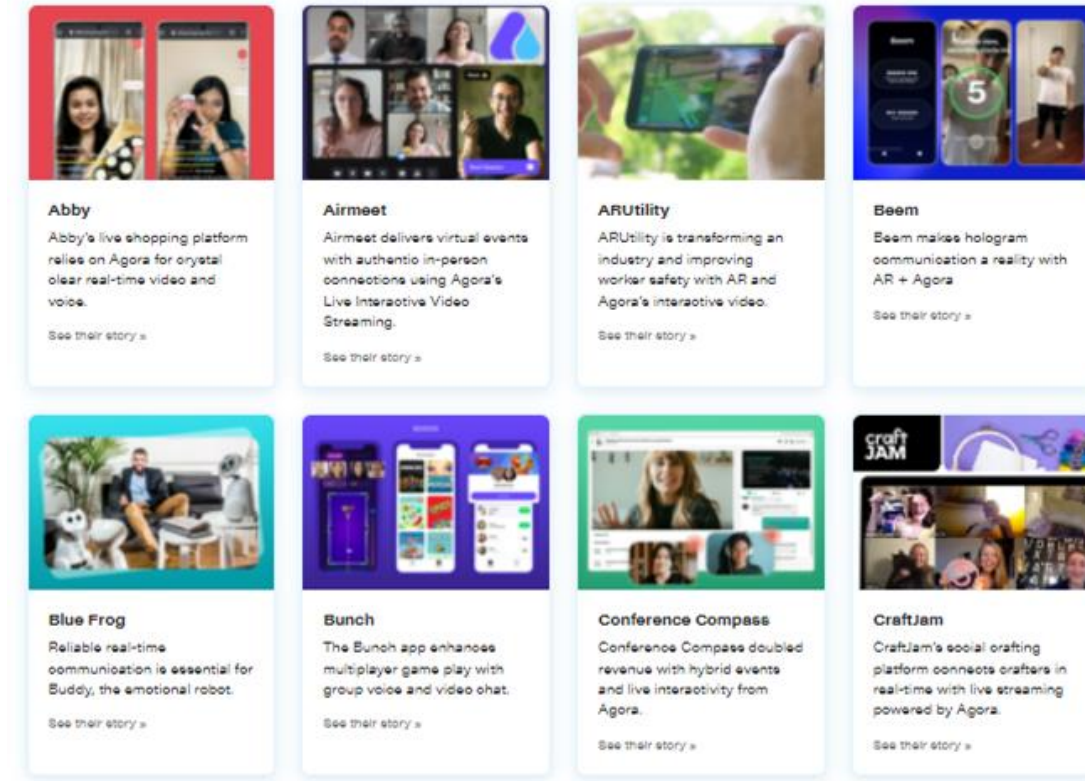


Agora

Metabusing 주요 기술



Agora Unity Asset 발견



1. 시중에서 많이 사용



iOS

android



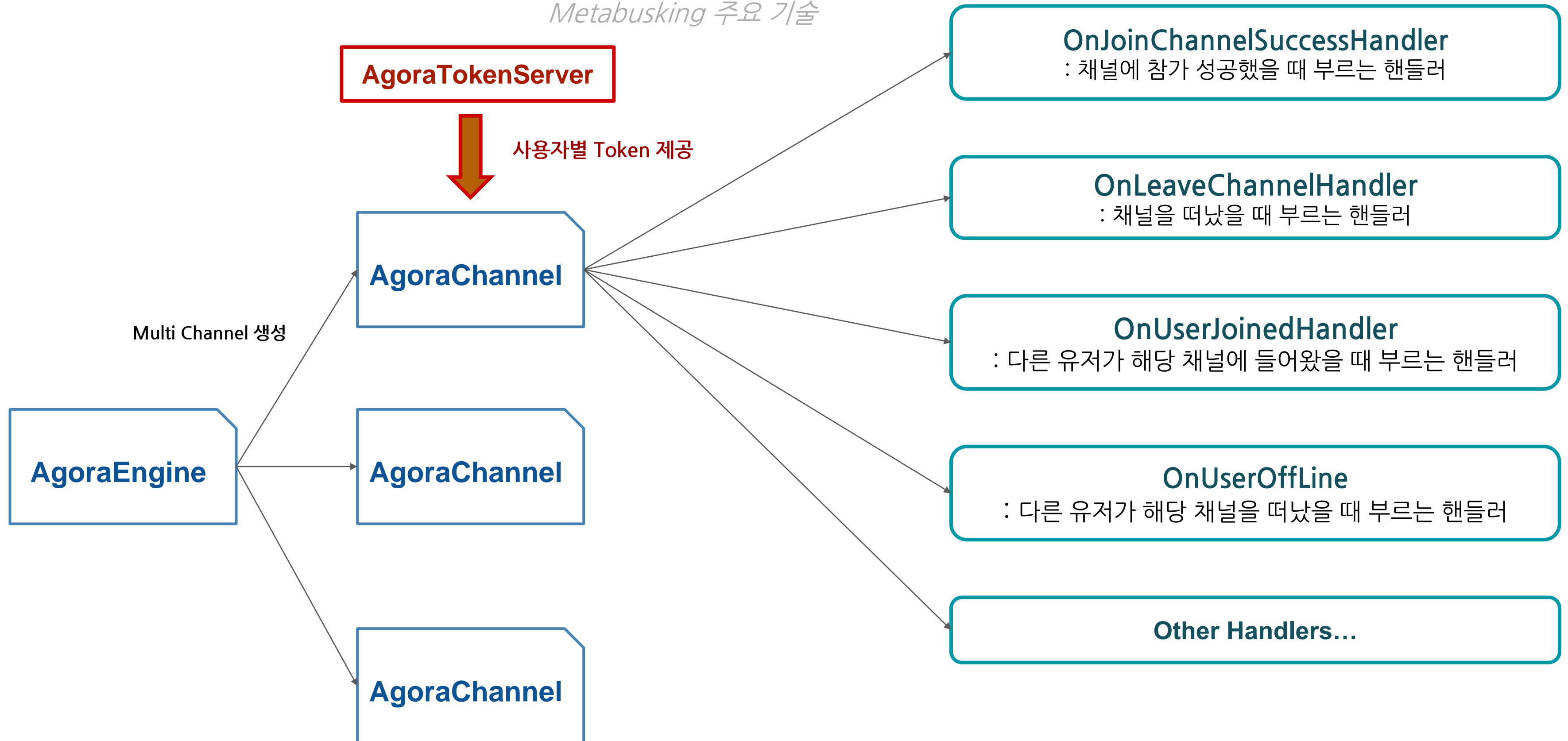
WEB

2. 높은 확장성과 멀티 플랫폼 지원

3. SD-RTN™ 네트워크로 인한 좋은 통신 환경 제공

Agora

Metabusing 주요 기술

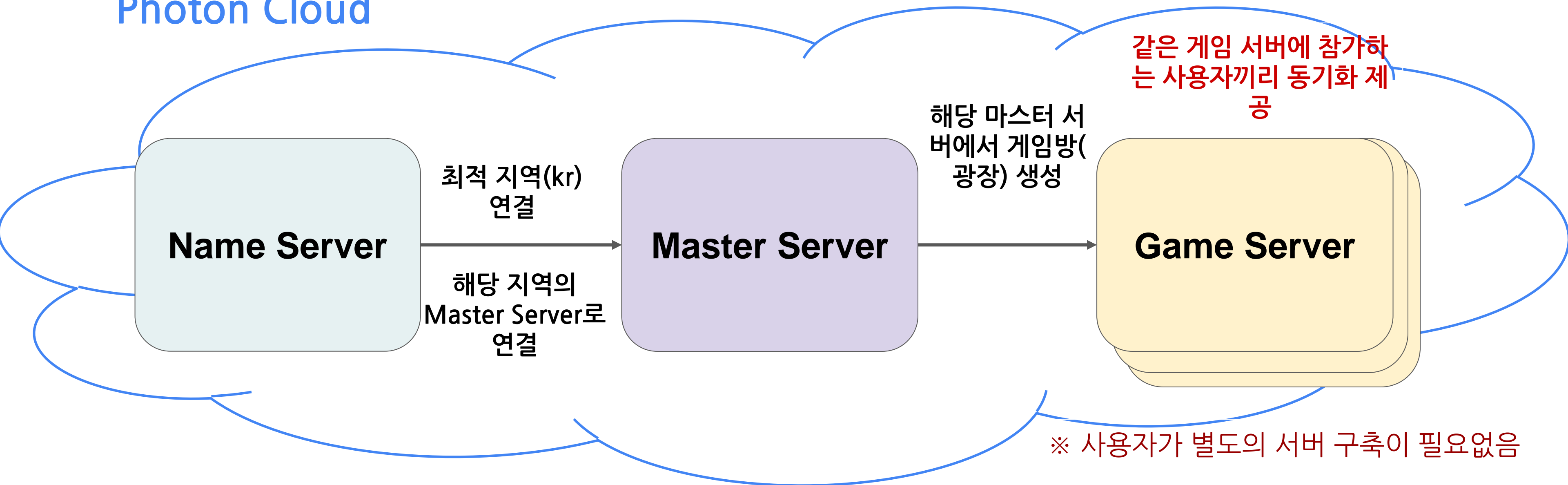


Photon

Metabusing 주요 기술



Photon Cloud



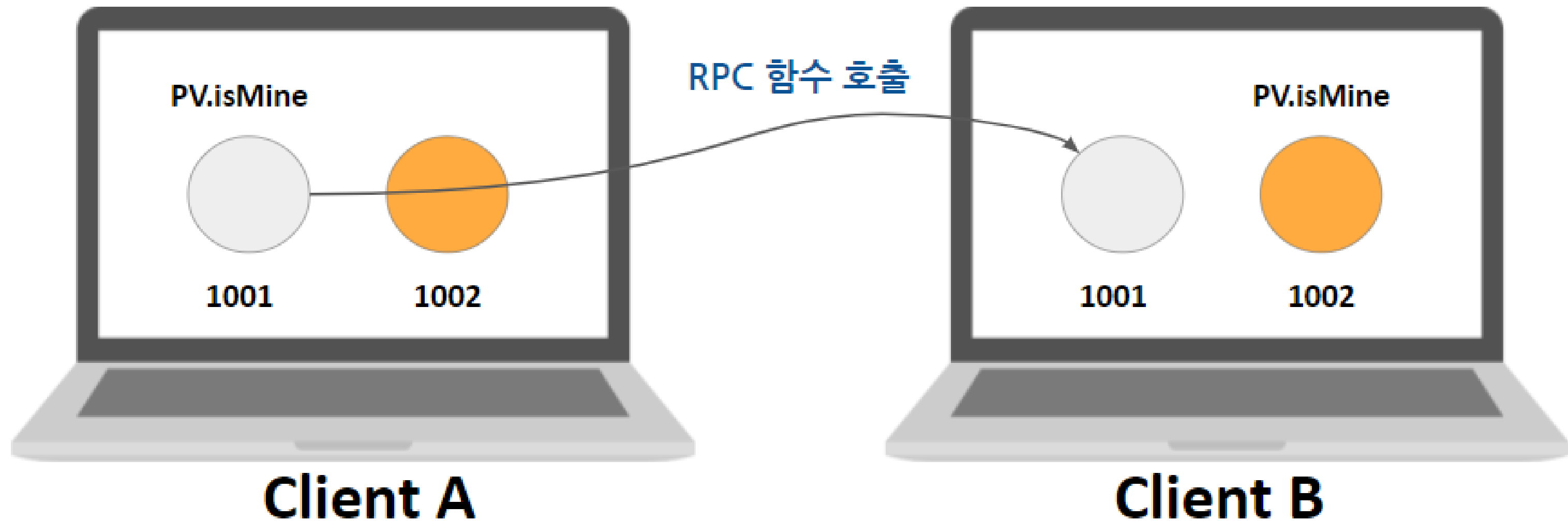
Photon

Metabusking 주요 기술

PhotonView 컴포넌트를 단 객체는 동기화 함수 호출 가능

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

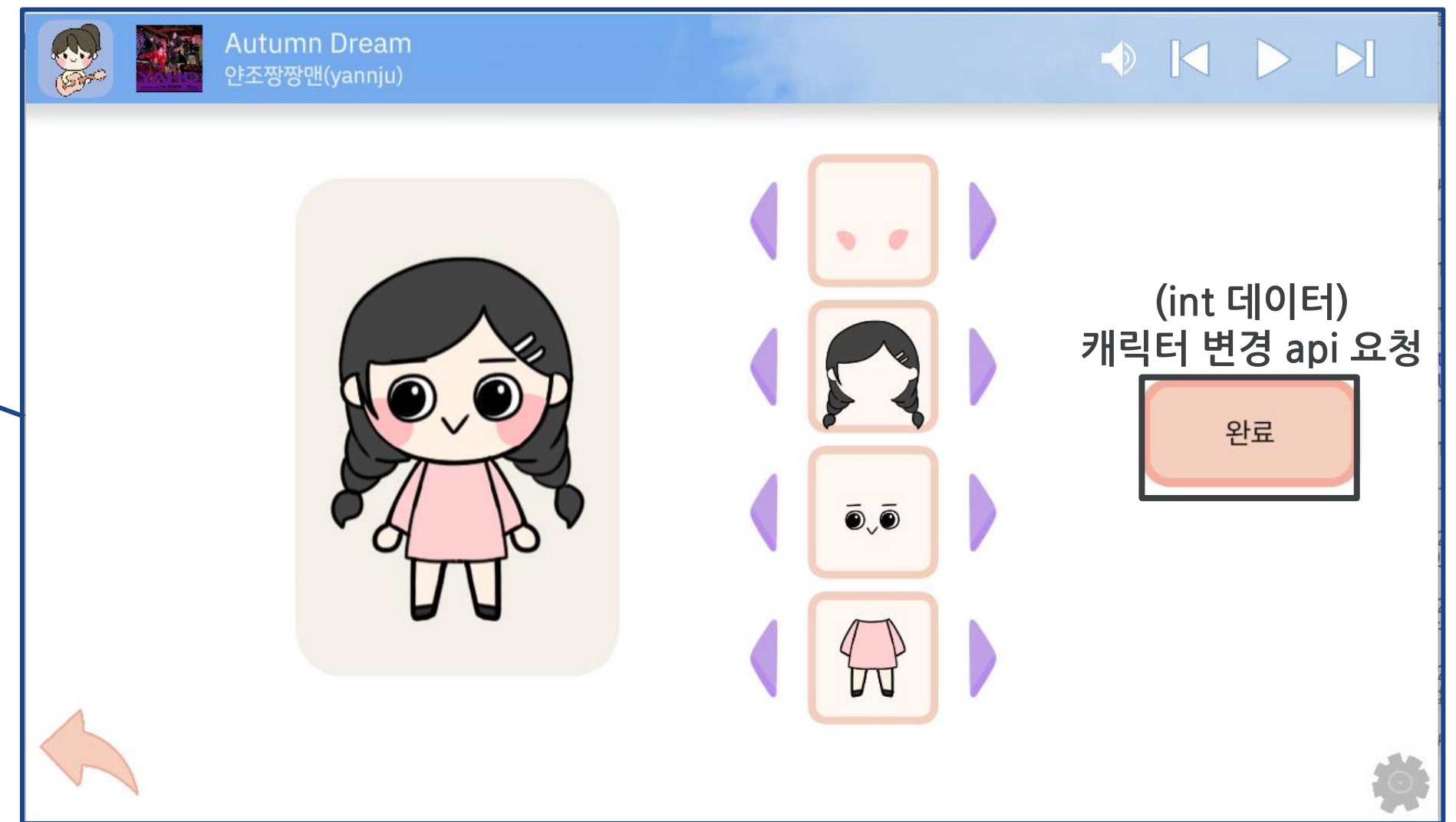
동기화



주요 기능

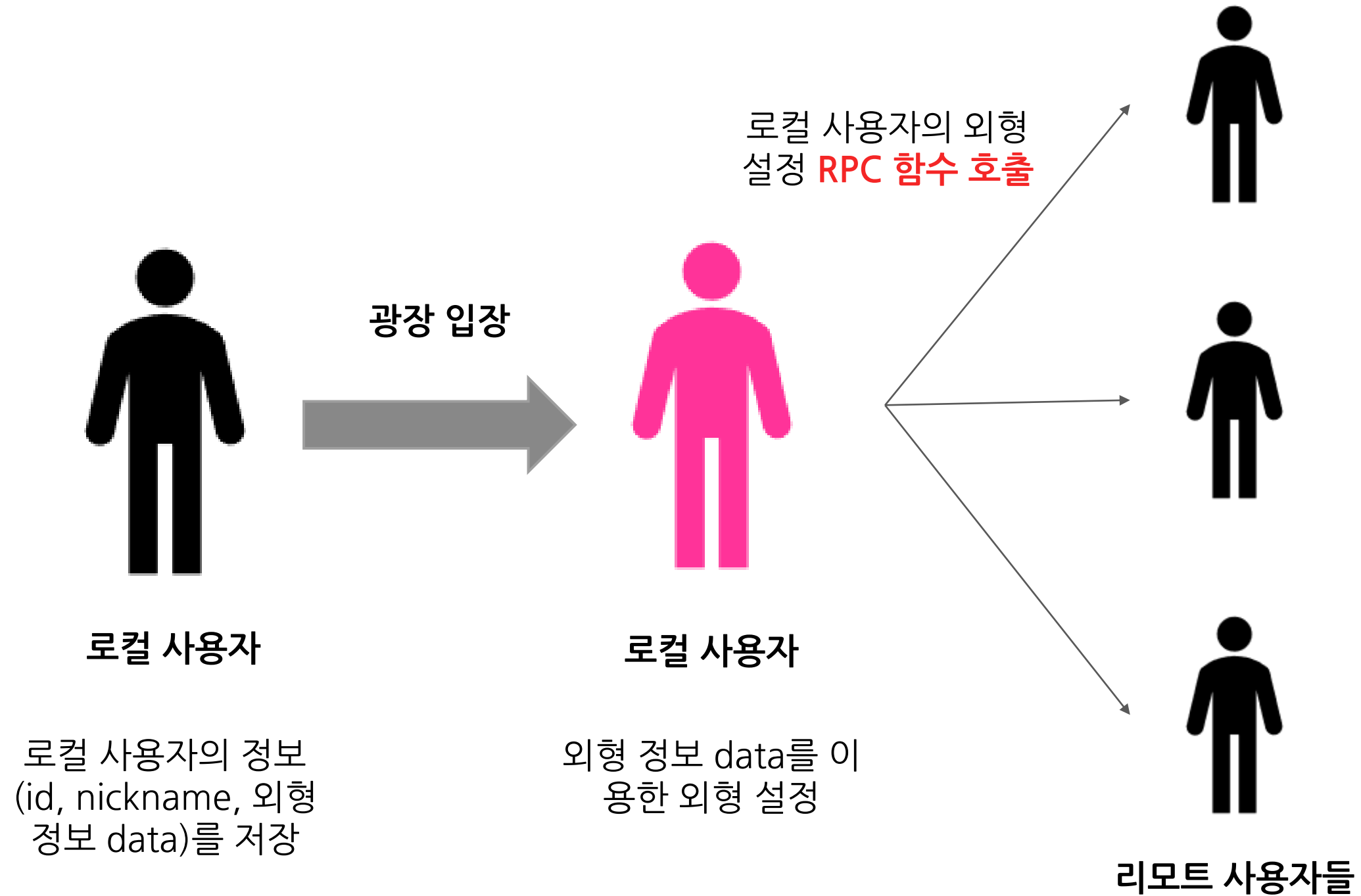
가상 캐릭터 설정

Metabusking 주요 기능



멀티플레이

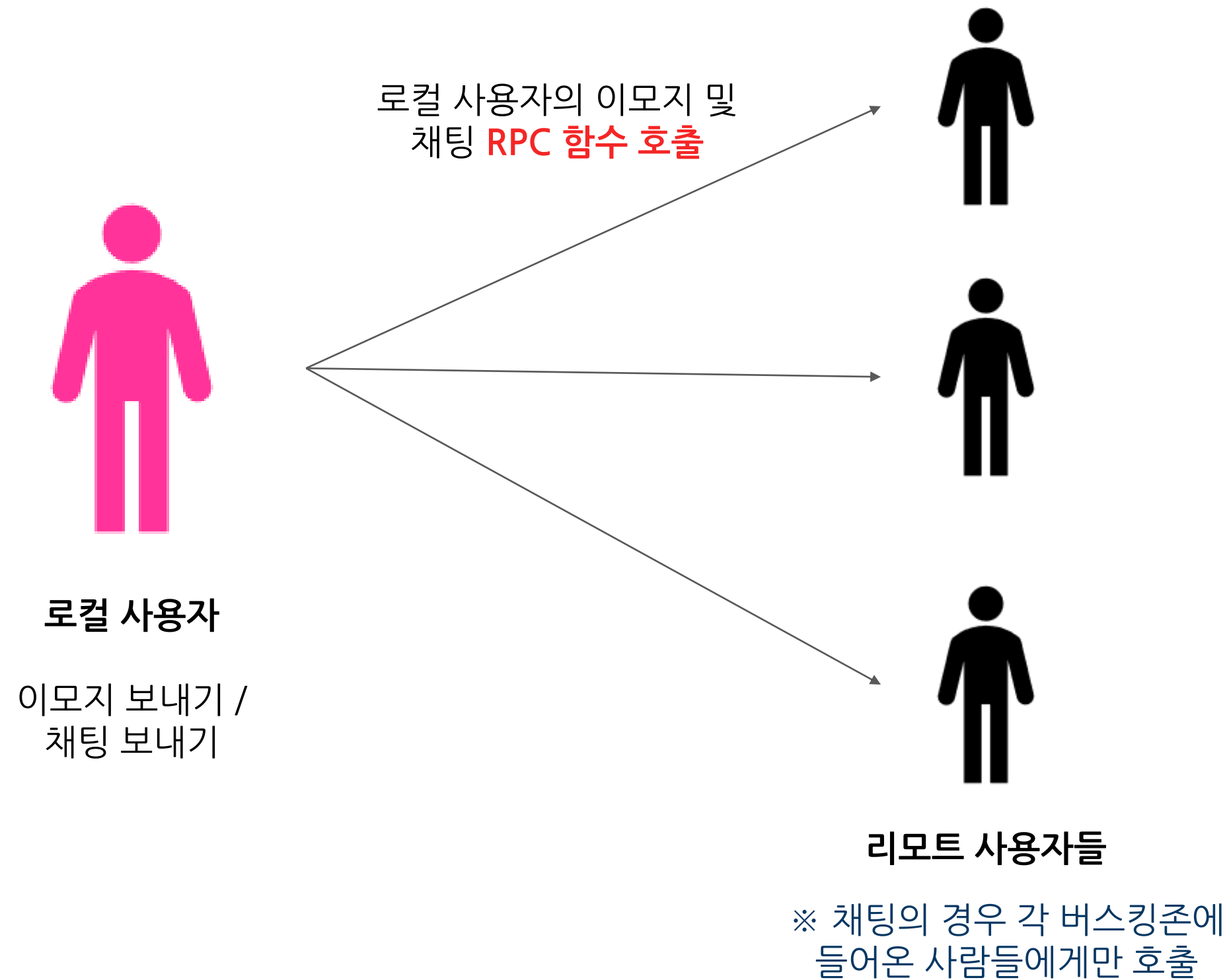
Metabusing 주요 기능



사용자 위치 동기화 + 사용자 애니메이션 동기화
+ 사용자 외형 동기화

멀티플레이

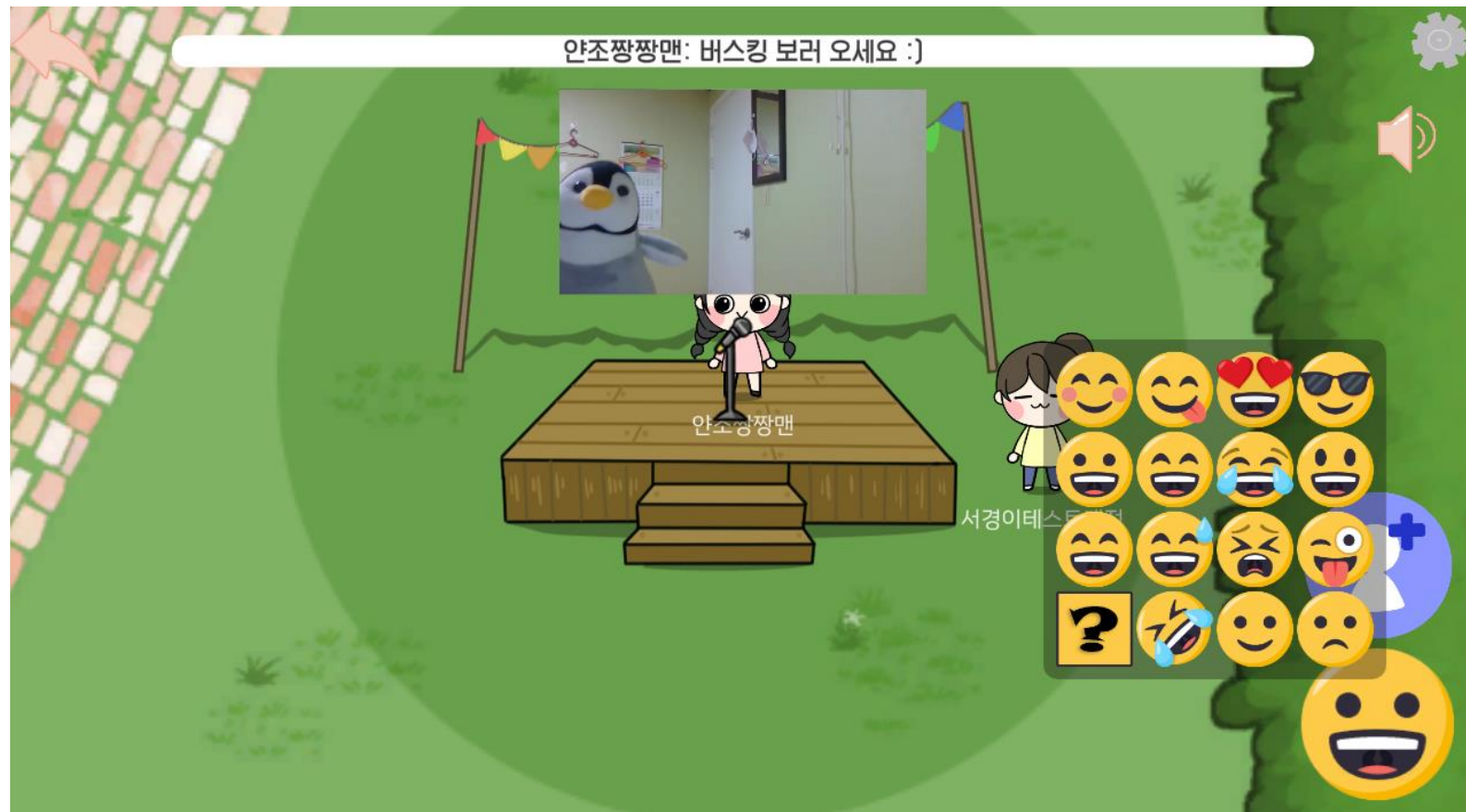
Metabusking 주요 기능



RPC 함수 호출을 통한 이모지 동기화
및 버스킹존마다 채팅 동기화

버스킹

Metabusking 주요 기능



작은 화면



큰 화면

버스킹존 입장

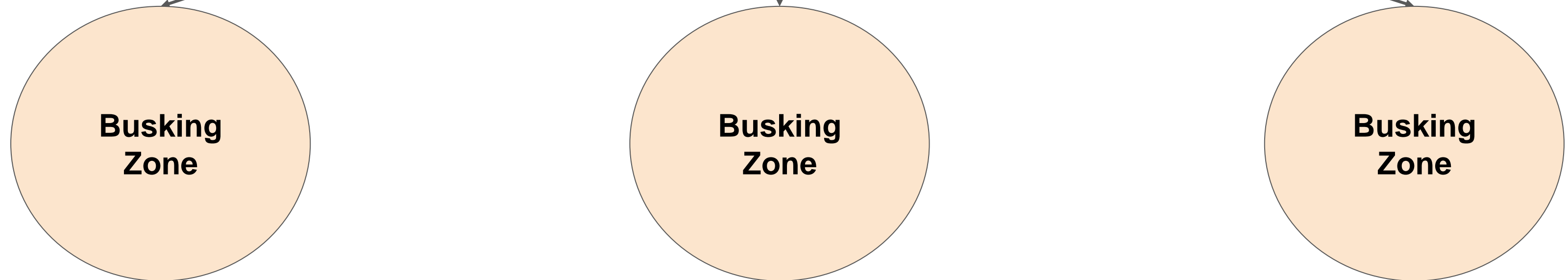
-> 버스킹이 진행되고 있으면 참여, 진행되지 않으면 버스킹 시작 가능

버스킹

Metabusking 주요 기능

agora

multi Channel 방식

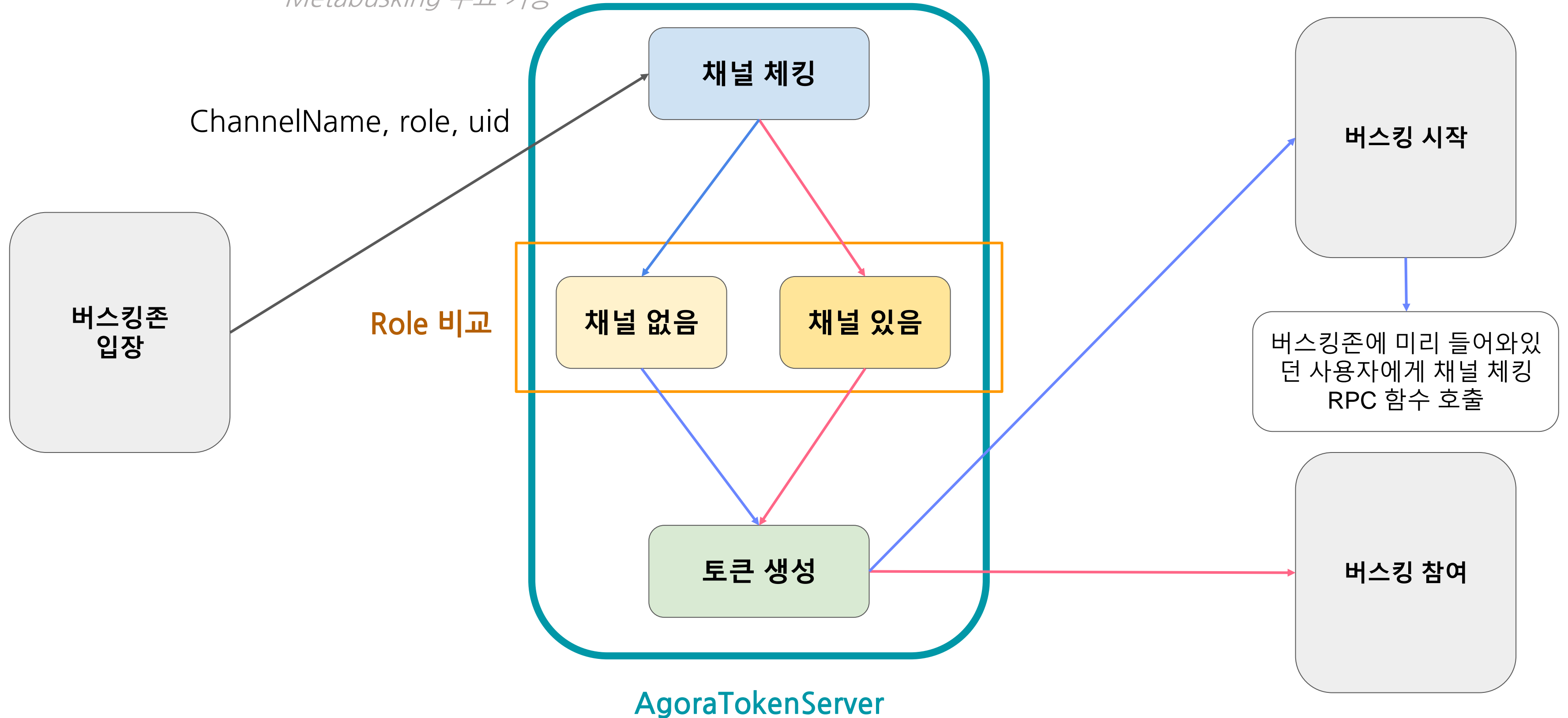


버스킹 존마다 AgoraChannel 생성

버스킹

청중
버스커

Metabusking 주요 기능



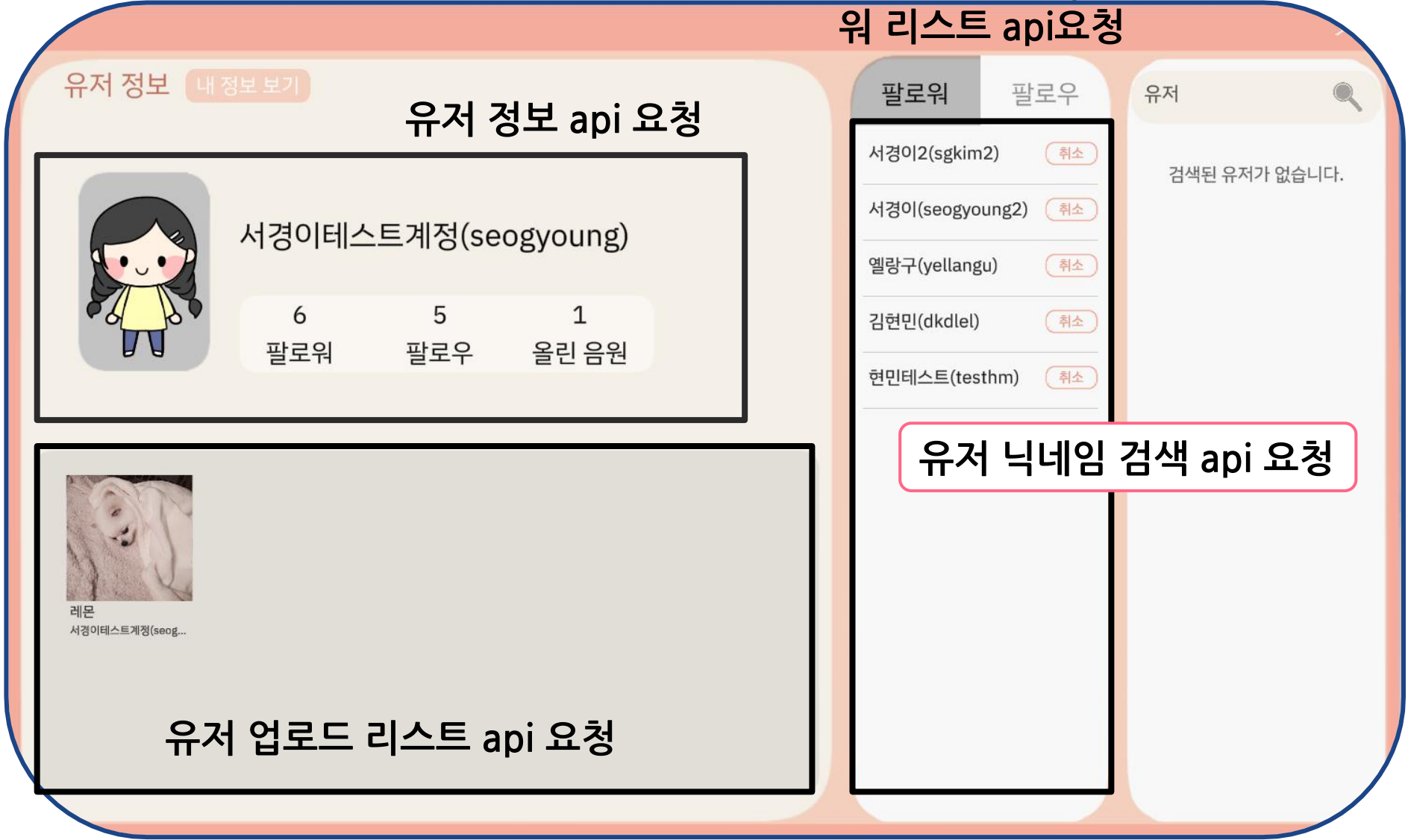
팔로우 시스템

Metabusking 주요 기능



로비에서 유저들이 올린 플레이리스트를 확인

유저 본인 팔로우, 팔로워 리스트 api요청



음원존

Metabusking 주요 기능



주크박스



유저의 **플레이리스트 등록**

=> 영역 안의 **모든 사람들에게** 같은 음원이 재생 된다.

음원존

Metabusking 주요 기능

음원존 생성



Zone 번호, 음악들의 제목, 위치, 시간 리스트



성공: 200 OK

실패: 410 이미 존재



저장



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

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

mongoDB find zone

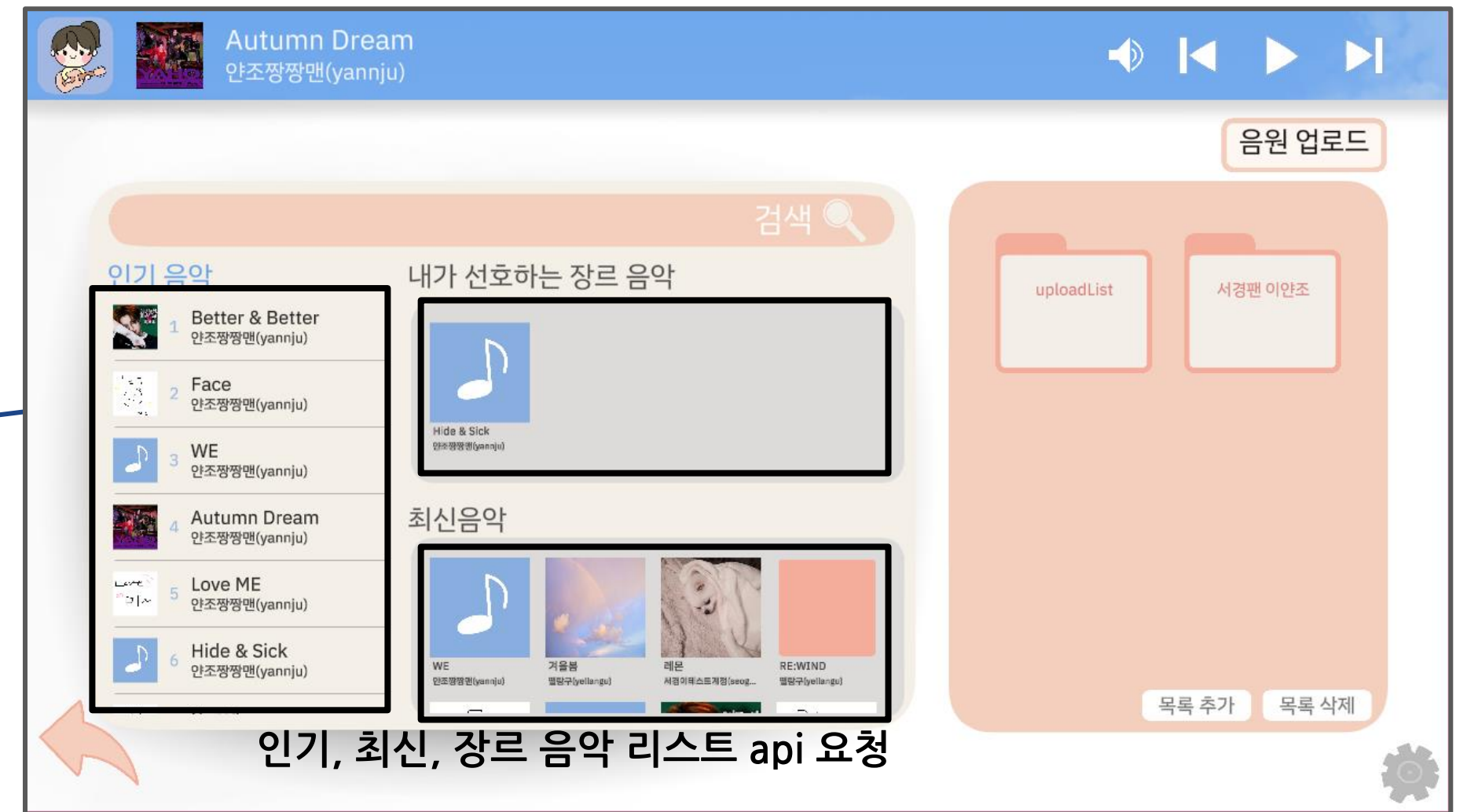
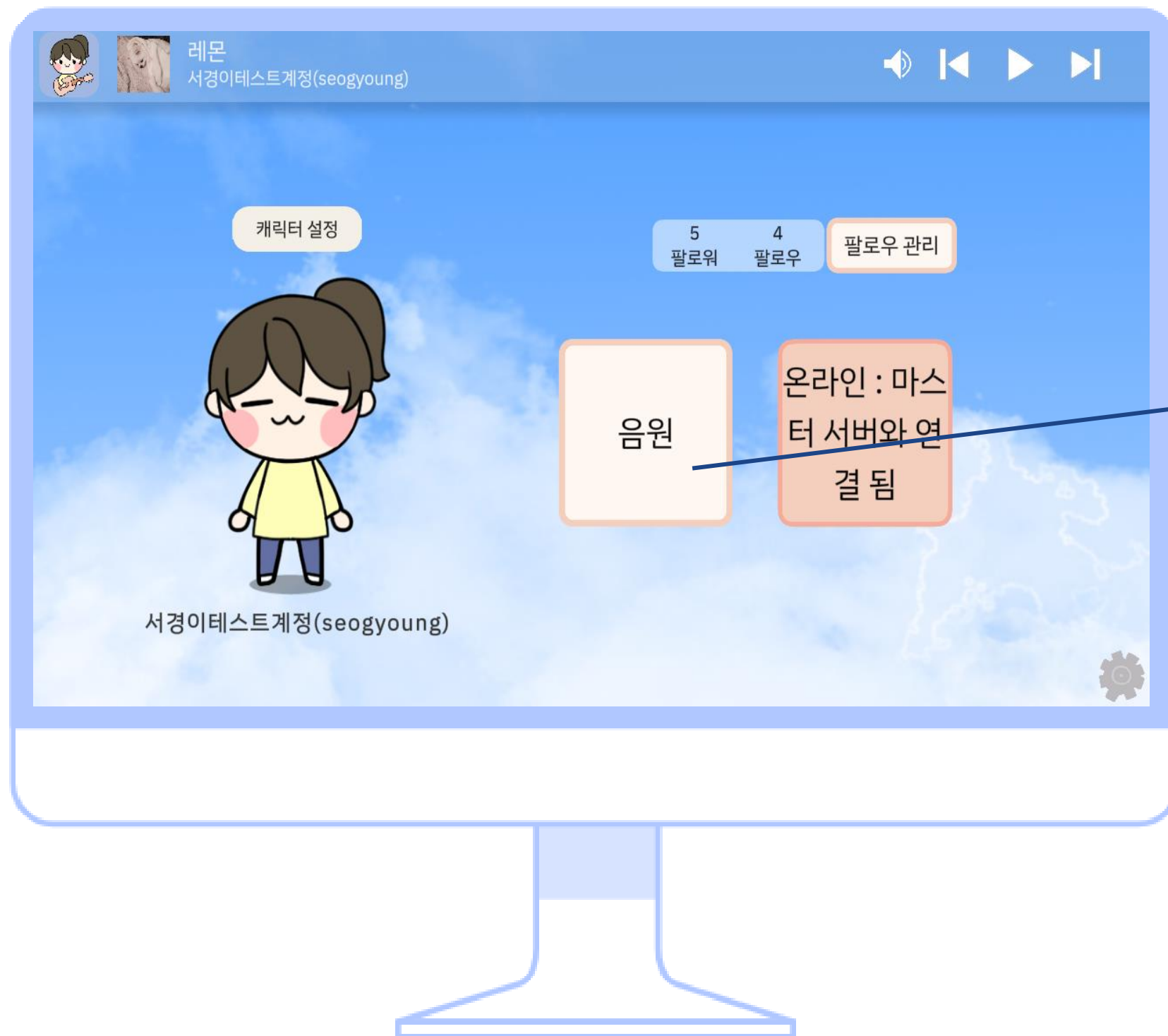
현재 시간과 리스트를 이용하여 현재 재생되는 시간 계산

시간을 기준으로
List 슬라이싱
현재 음악 재생 시간

← zone document
mongoDB®

음원 페이지

Metabusking 주요 기능



음원 업로드

Metabusking 주요 기능



나만의 음악을 만들거나,
버스킹한 노래를 녹음하고

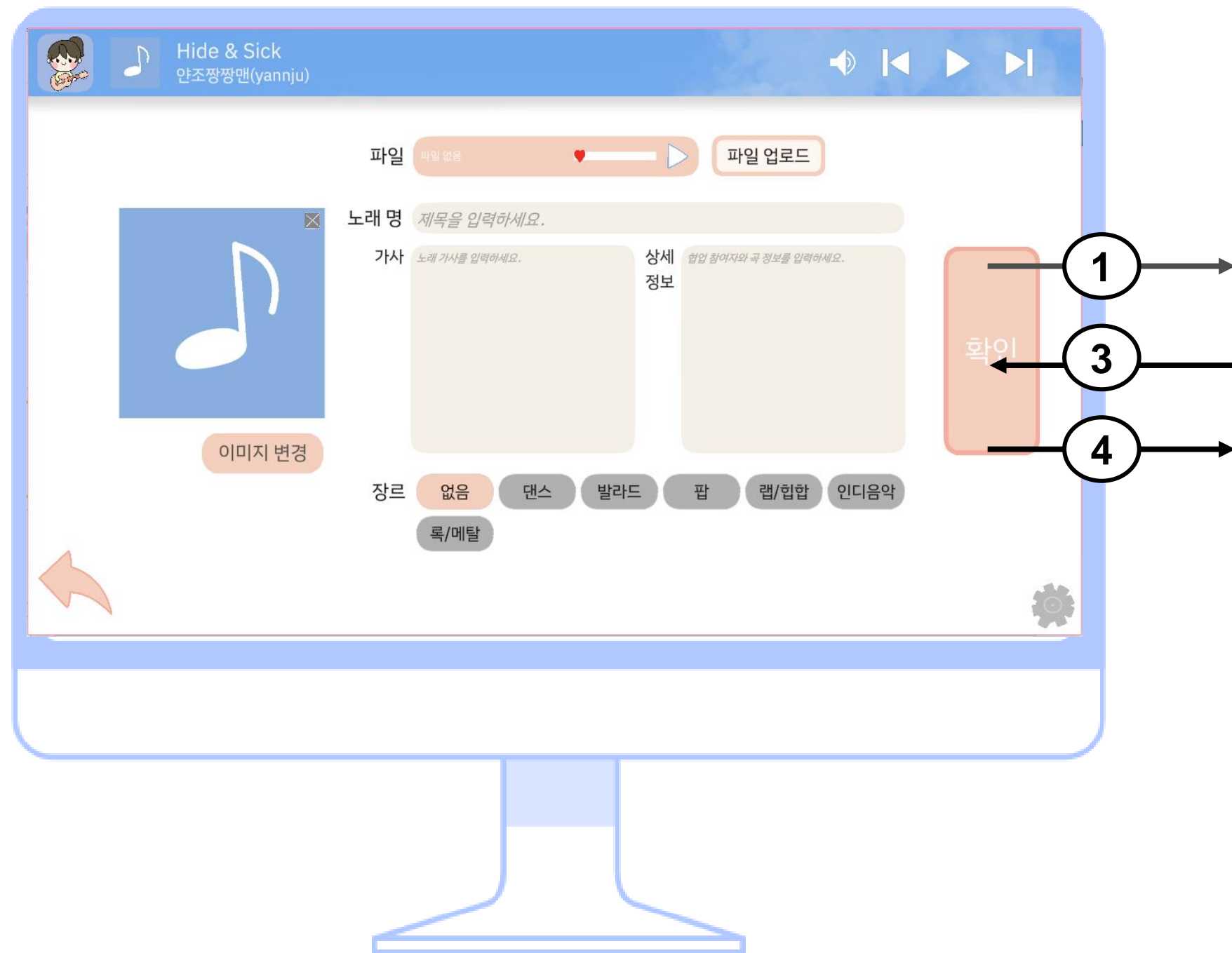
MetaBUSking



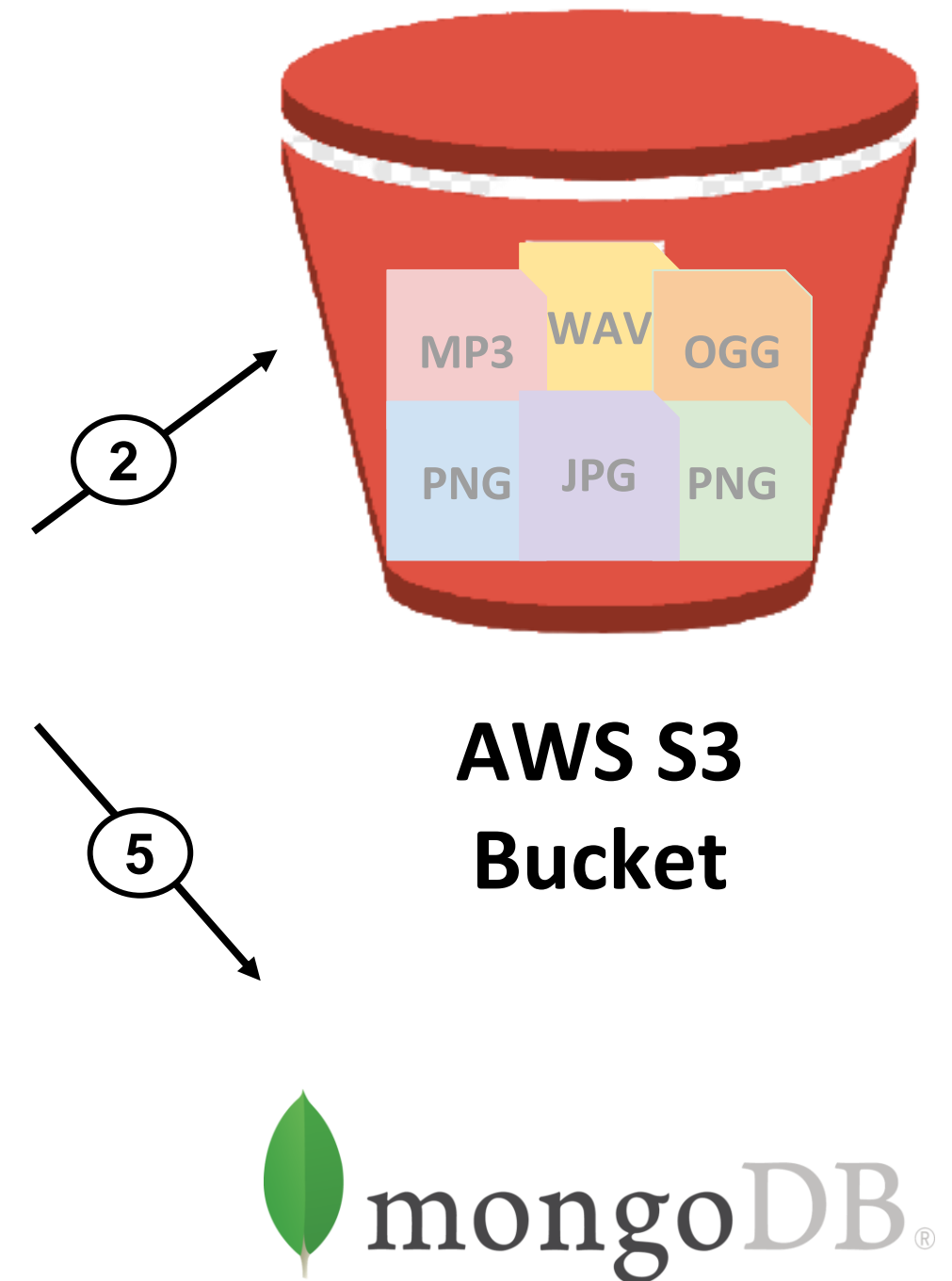
메타버스킹에 업로드!

음원 업로드

Metabusing 주요 기능

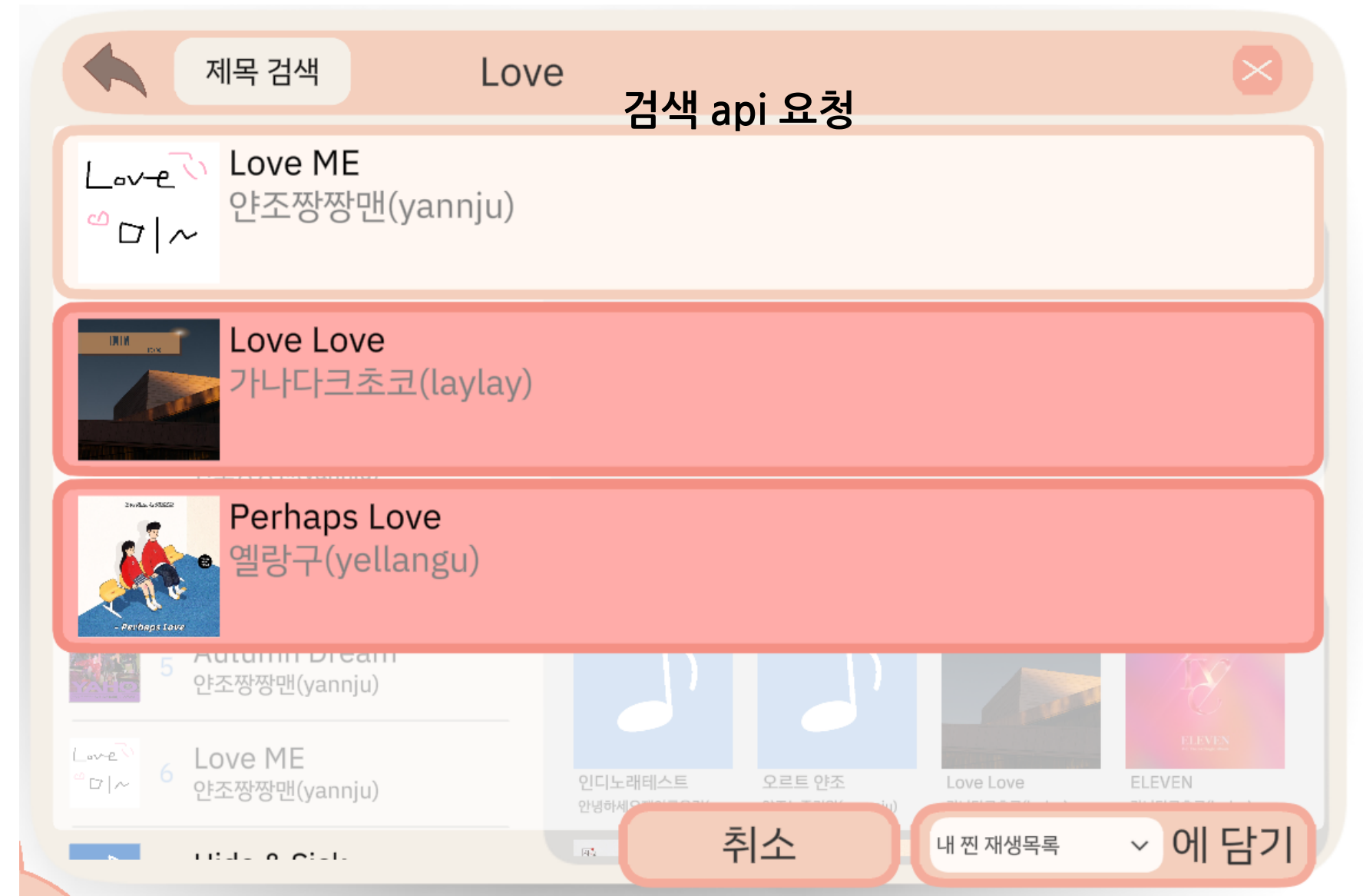
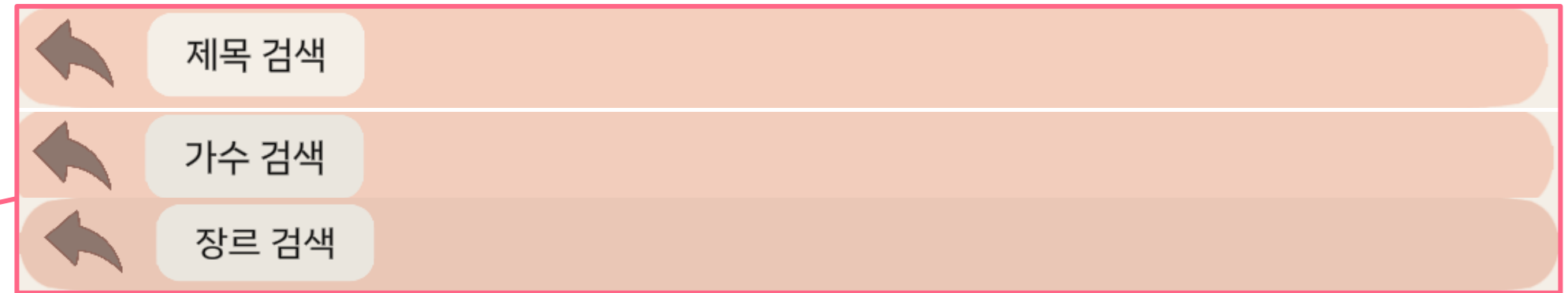


node
JS
express



음원 검색

Metabusking 주요 기능

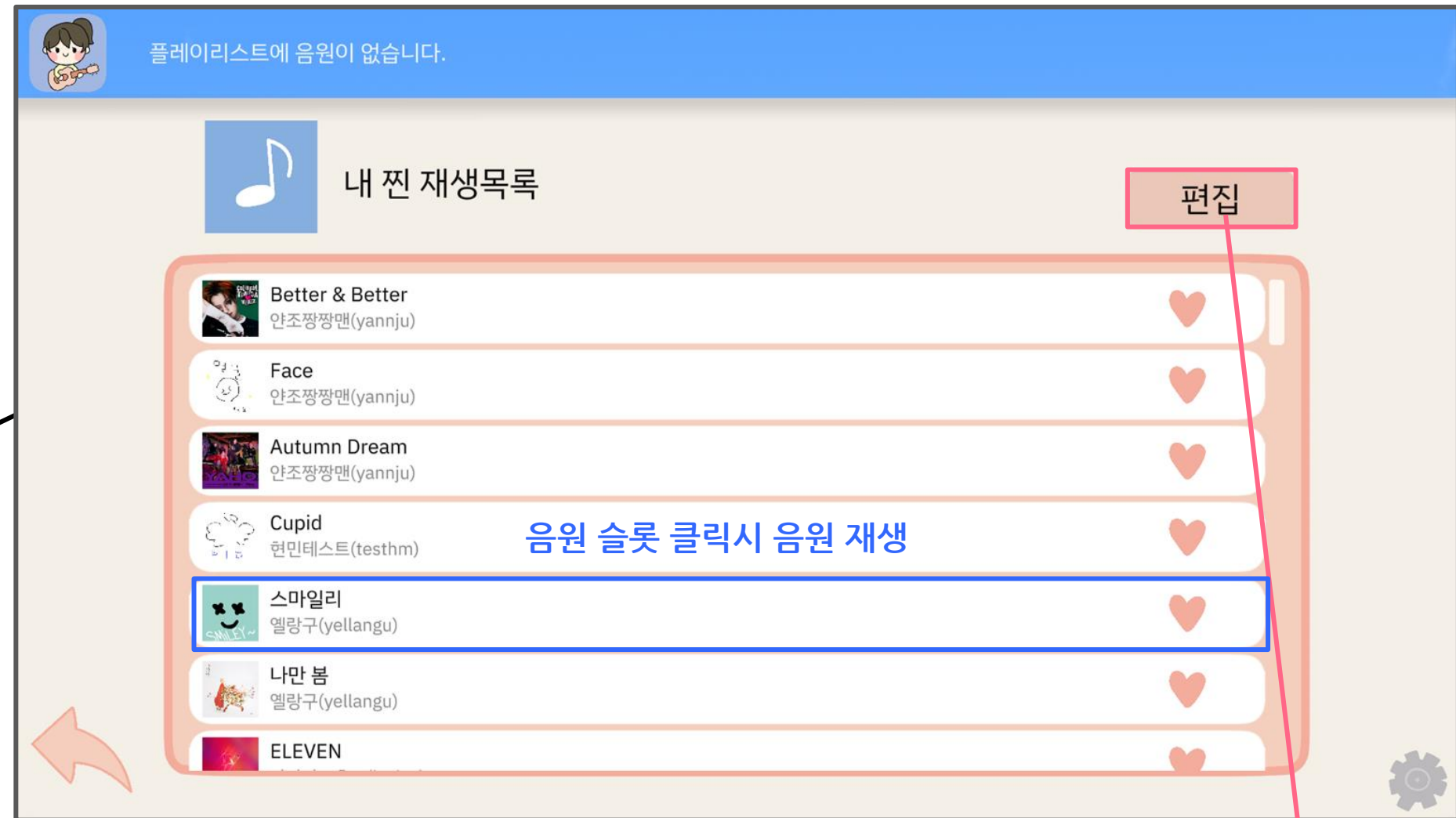


재생목록 편집

Metabusking 주요 기능

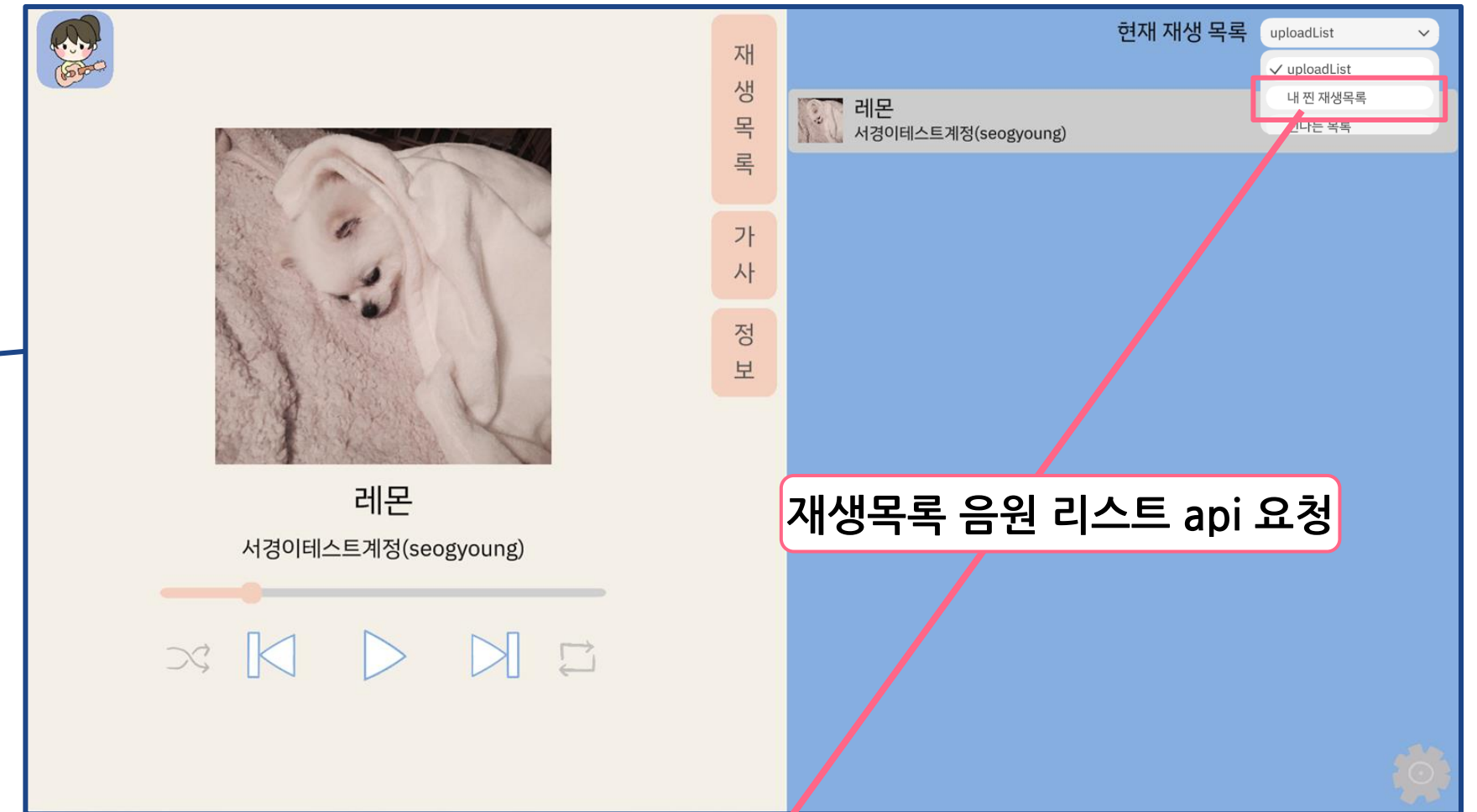
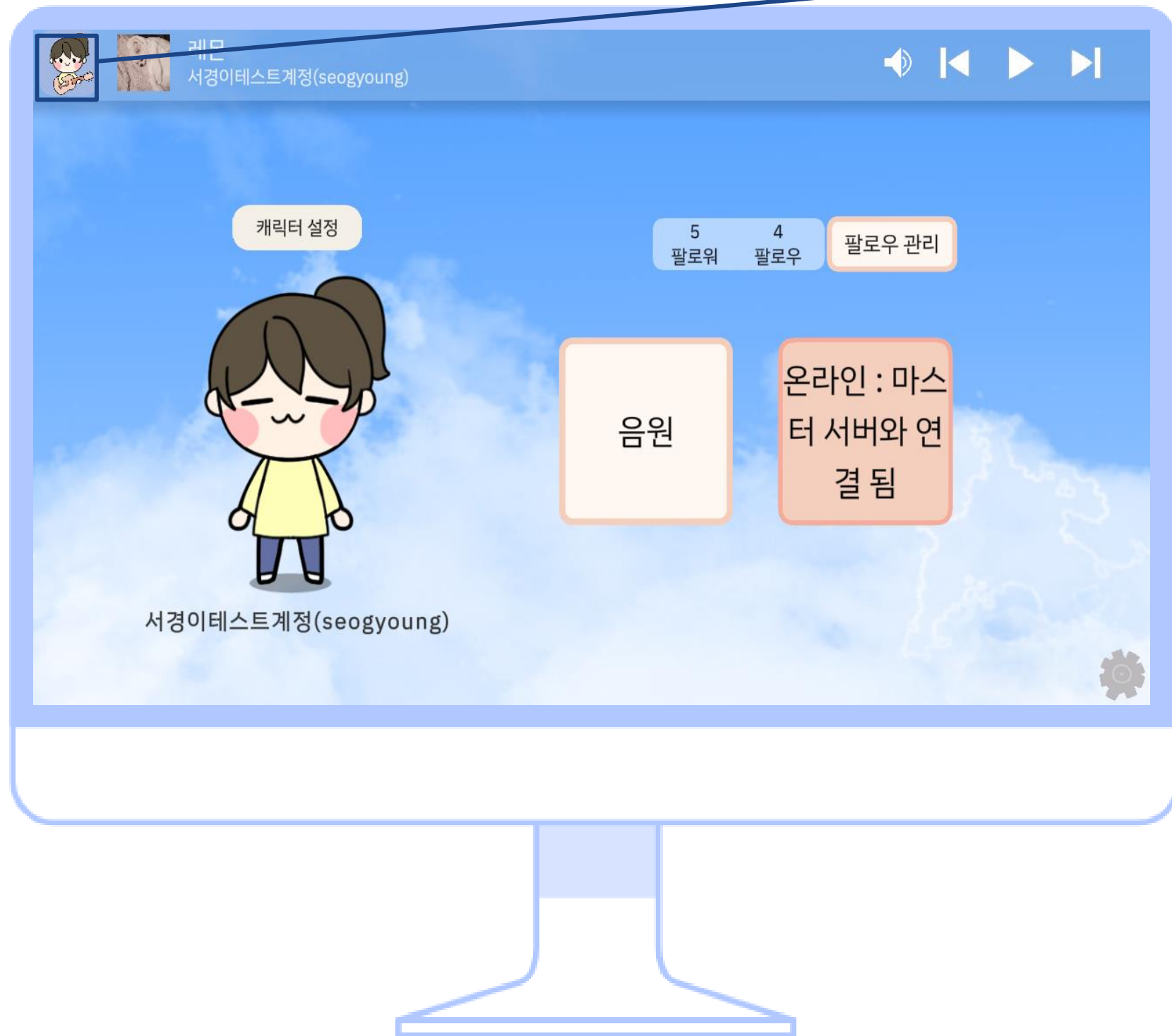


재생목록 음
원 리스트
api 요청



음원 재생

Metabusking 주요 기능



재생목록 음원 리스트 api 요청



음원 재생 상태 컨트롤 오브젝트

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**

Chapter.4

기대 효과

04

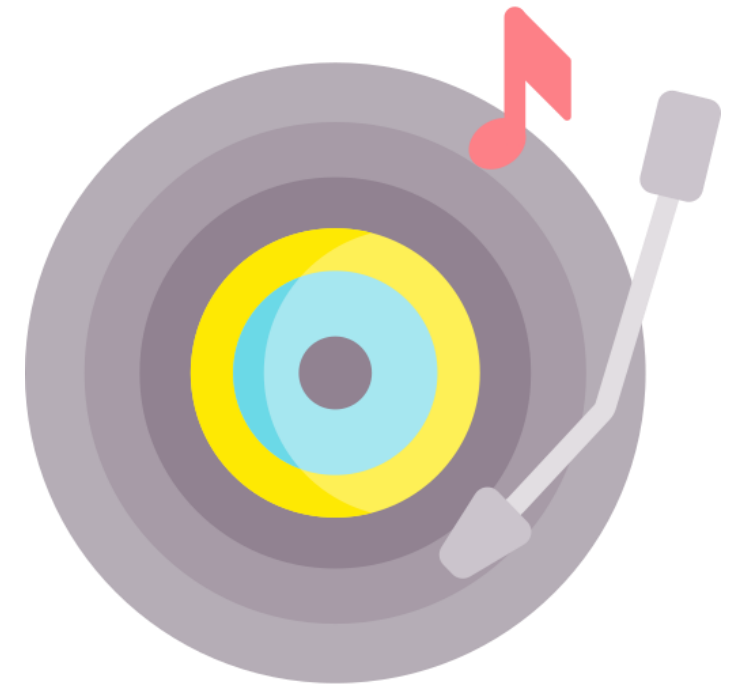
기대효과



온라인 공연 확대



아티스트 홍보



버스킹 접근성 증가

Chapter.5

Next Plan

추후 계획

05

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

감사합니다