

2022

Capstone Project

중간 발표 - 11조

CONTENTS. |

01

프로젝트 개요

Metabusking

02

다른 서비스와의
차별성

03

시스템 설명

04

Next Plan

추후 계획

Chapter.1

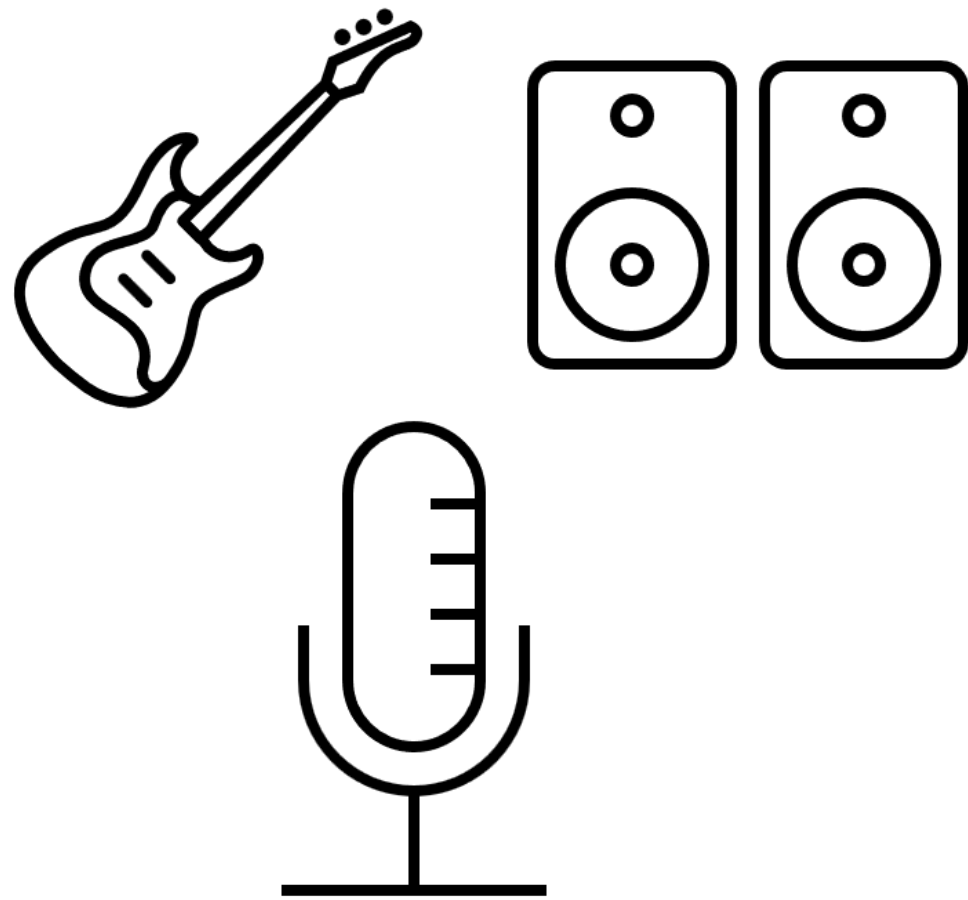
프로젝트 개요

Metabusking

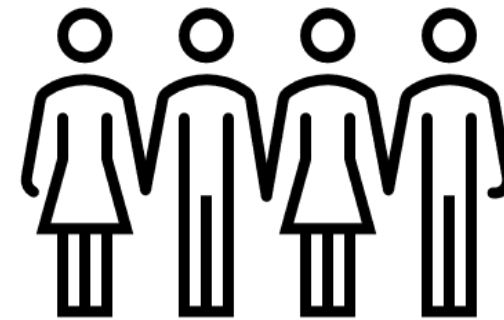
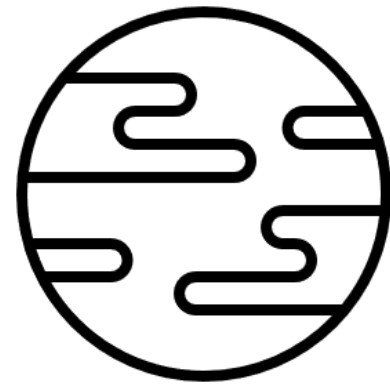
01

MetaBusking

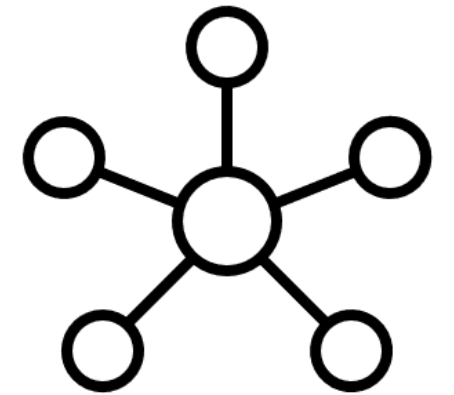
온라인 버스킹 및 음원 유통 플랫폼



버스킹



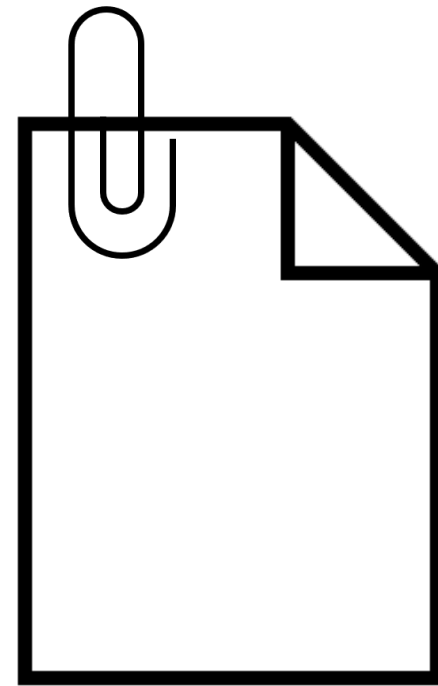
메타버스



음원 유통

MetaBusking

온라인 버스킹 및 음원 유통 플랫폼



실행 파일(.exe)

제공 플랫폼

기획 배경

코로나 팬데믹 시대

- 코로나 팬데믹으로 인한 모든 실생활의 온라인화
- 길거리 버스킹 문화도 주춤하기 시작함

오프라인 버스킹의 한계

- 좋아하는 뮤지션의 버스킹을 지역적/시간적 한계로 참여하지 못하는 사람들이 있음
 - 오프라인 버스킹은 날씨 등 외부적인 환경에 영향을 많이 받음

기획 배경

아티스트들의 홍보 시장 부족

- 자신을 알리고 싶은 아티스트들에겐 공연의 기회가 부족함
- 누구나 참여할 수 있으며 자신의 실력을 보여줄 수 있는 장소가 마련된다면, 아티스트들이 본인을 홍보할 수 있는 기회가 됨

프로젝트 목표 및 기획의도

1. 메타버스라는 가상세계 시스템을 도입해 오프라인 버스킹의 공간적/시간적 한계를 벗어나 온라인으로 버스킹 문화를 확장하고자 한다.
2. 나의 음악을 공유하고, 공연할 수 있으며 다양한 아티스트의 음악을 감상하는 음원 플레이 리스트를 제공해서 온라인 음원 유통 플랫폼의 역할을 겸한다.
3. 새로운 버스킹 플랫폼은 자신을 알리고 싶은 아티스트들의 새로운 홍보 시장이 되고자 한다.

예상 타겟층

아티스트

- 본인의 음악 홍보
- 공연 장소 마련

음악을 좋아하는 사람들

- 음원 플레이 리스트 사용
- 버스킹 관람 및 소통

02

다른 서비스와의 차별점

유사한 서비스

- 현재 유명한 메타버스 서비스는 ifland, 제페토, 게더타운, ZEP가 존재

ifland



ifland

- 동영상과 파일을 공유할 수 있음.
- 하지만 ifland는 캠 화면을 사용할 수 없기 때문에 버스킹은 불가능하다.



제페토

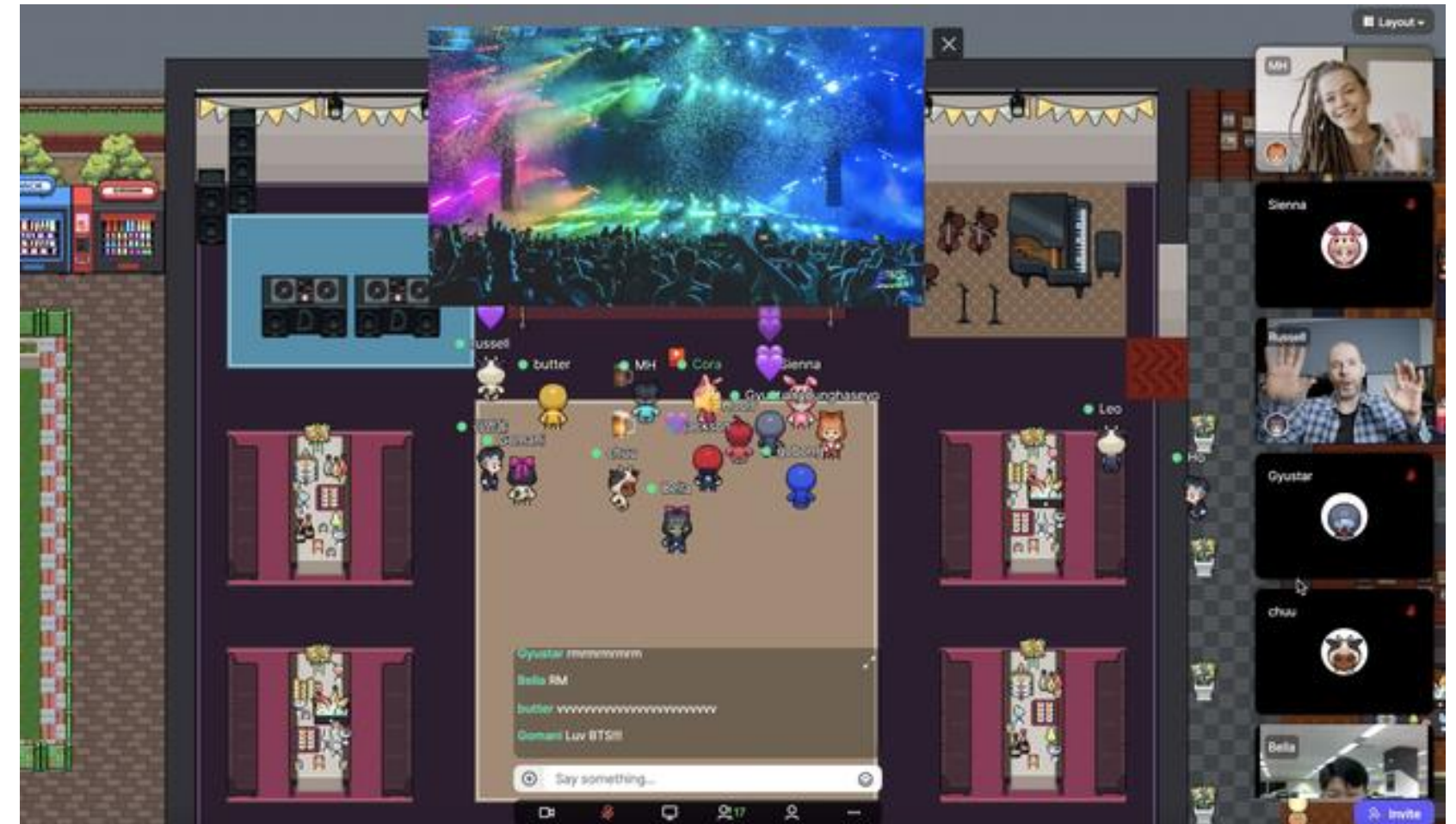


- 메타버스 공간속에서 다양한 유저들과 소통하고 다양한 공간을 이동할 수 있음.
- 제페토는 SNS 기능에 집중되어 있음.
- 제페토는 현재 라이브 스트리밍(라이브 방송) 기능이 일부 사용자에게만 제공되고 있음
- 모바일로 최적화가 되어 있어 PC접근을 위해서는 녹스, LD 등 애플레이어를 등을 설치하여야 함





- 유저들끼리 화상채팅이나 메신저 채팅을 통해 소통할 수 있음.
- 파일 공유는 불가능 하다.
- 스포트라이트 영역을 통해 화면/화상 공유를 할 수 있지만 맵 전체에 공유되기 때문에 일정 지역 안의 사람들끼리 볼 수 없어서 한명만 방송을 할 수 있음.



게더타운



- 유저가 공유한 링크를 통해 한 공간에서 화상채팅이나 메신저 채팅을 통해 여러 유저가 소통할 수 있음.
- 25명 이상부터는 금액을 내야 함.
- 아이디 없이 사용이 가능하여 편리하지만 내 정보를 저장할 수 없음



다른 서비스와의 차별점

MetaBusking의 특징

- 지나가면서 방송하는 일정 영역 안으로 들어갈 때 노래가 들리므로 실제 버스킹의 느낌을 받을 수 있다.
- 아티스트는 자신의 음원을 업로드 하여 다른 사람들이 노래를 들을 수 있다.
- 다른 사람들이 작곡한 노래들을 내 리스트에 추가하여 들을 수 있다.

다른 서비스

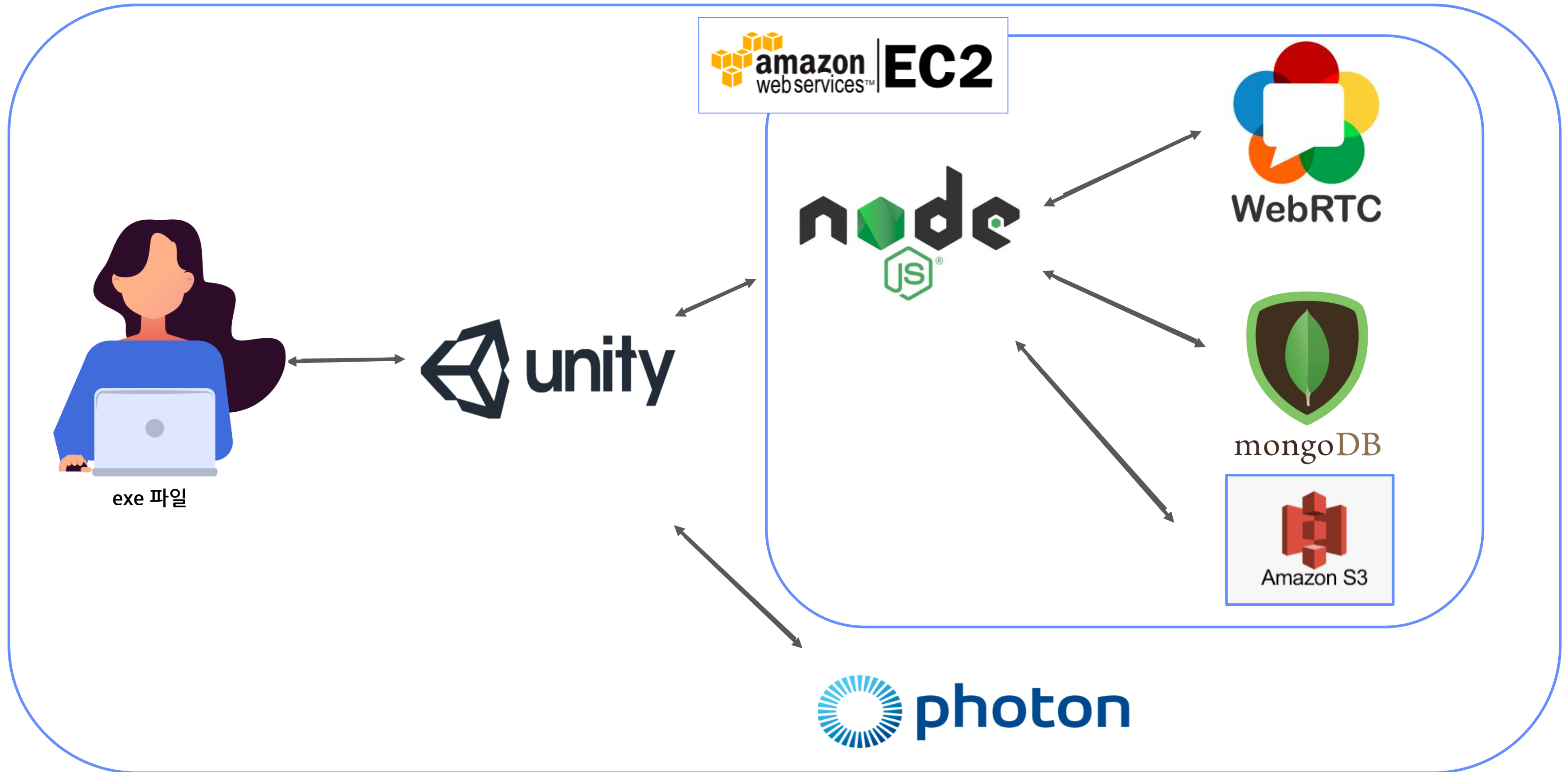
- 화상 공유를 하지 못하는 서비스들은 버스킹을 하기 힘들다. (Ifland, 제페토)
- 버스킹을 하는 공간을 만들어도 여러 방송을 나눠서 하기 힘들다. (ZEP)
- 일정 인원 이상을 수용하게 되면 비용 지출이 생긴다. (게더타운)
- 자신의 음원들을 업로드 할 수 없어 남들에게 공유하는 공간이 없다. (Ifland, 제페토, ZEP, 게더타운)

Chapter.3

시스템 설명

03

소프트웨어 아키텍처



선택 이유



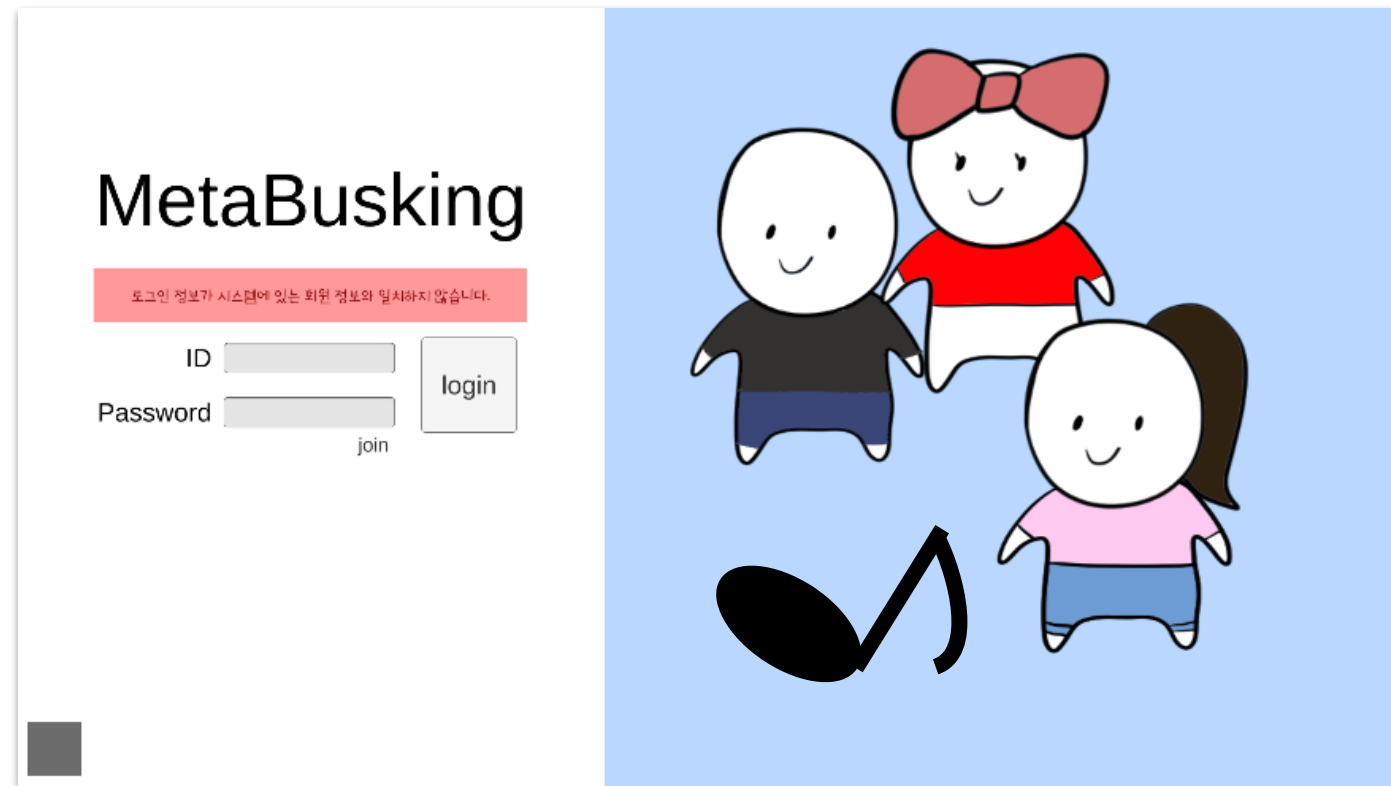
mongo DB 구조

User	
id	String
email	String
password	String
nickname	String
character	Number
totalNum	Number
musicList	Array
myList	Array
Created	Date

Music	
id	String
title	String
locate	String
userId	String
category	String
imagelocate	String
created	Date

- Music은 User의 id를 가지며 Reference 저장 방식으로 관계를 가지고 있음.
- User의 Array에는 music Id를 Reference 저장 방식을 사용하고 있음
- User는 musicList와 myList에 Music의 id를 가지고 있고, Music은 userID로 식별자를 가지고 있음.

1.로그인/회원가입 기능



※ 임시 이미지입니다.

MetaBusking

닉네임 중복 확인
영문, 숫자, 한글로 구성된 2~10자를 입...

ID 중복 확인
영문, 숫자로 구성된 6~11자를 입력하세요.

Password
8~16자의 특수문자, 숫자, 알파벳 조합...

Password 확인

이메일 보내기
올바른 메일 주소가 아닙니다.

인증 코드 확인

<선호 장르>

☒ 팝 ☐ 락 ☒ 힙합 ☐ R&B ☒ 트랩 ☐ 인디 ☒ 기타

회원가입과 로그인 기능

- 닉네임, ID, Password, email을 입력 받아 아이디와 이메일이 중복되지 않으면 디비에 추가
- token을 이용하여 인증하는 방식.

2. 음원 관리, 유저 관리 기능 로비 화면



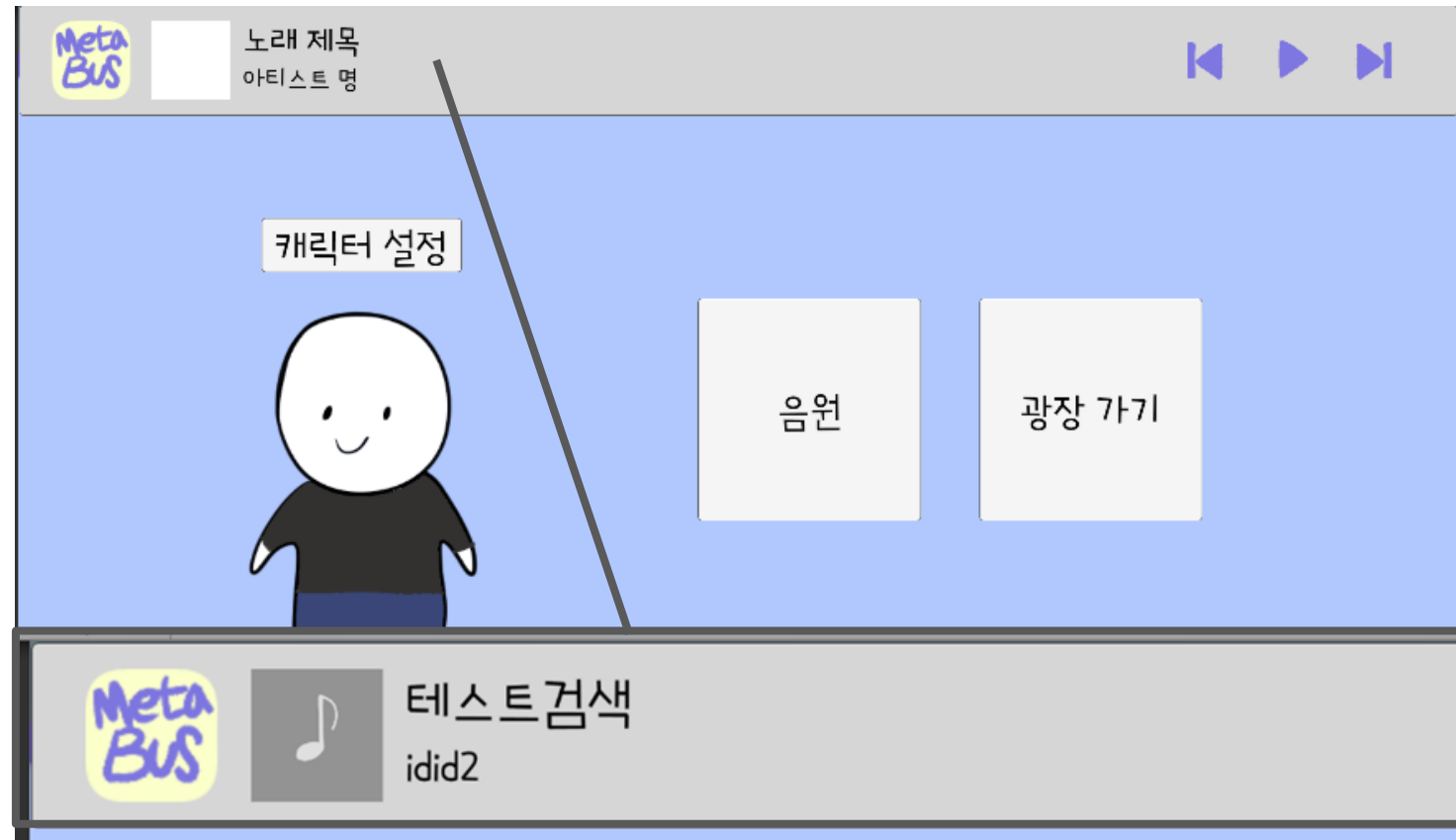
메타버스 로비

- 로그인할 때 서버에서 유저 정보를 받아오기 때문에, 로그인한 유저의 캐릭터를 확인할 수 있음.
- NodeJS 서버와 통신하여 캐릭터를 변경함. 변경된 값은 DB에 수정됨

※ 임시 이미지입니다.



2. 음원 관리, 유저 관리 기능 음원 재생

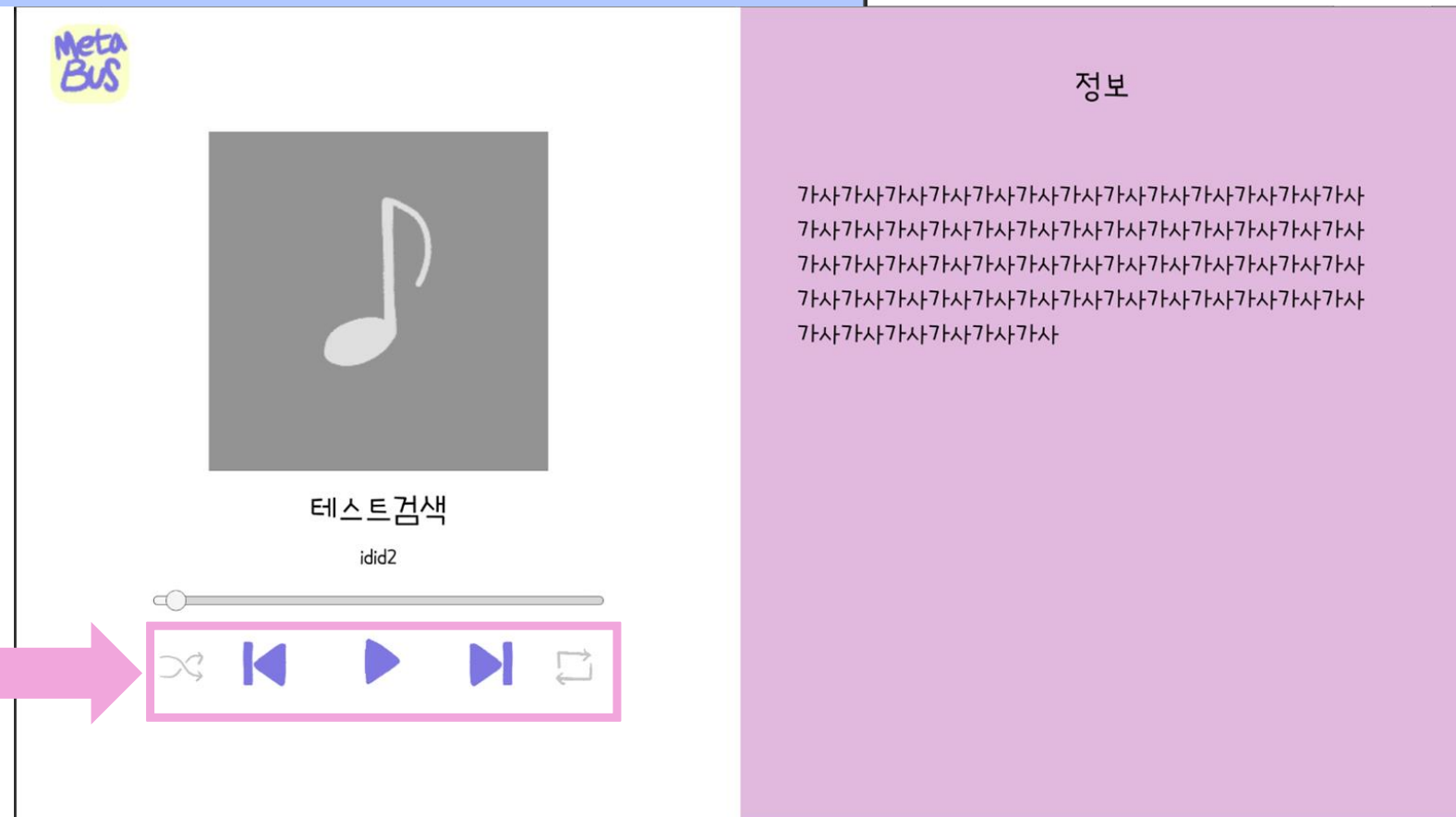


메타버스 로비

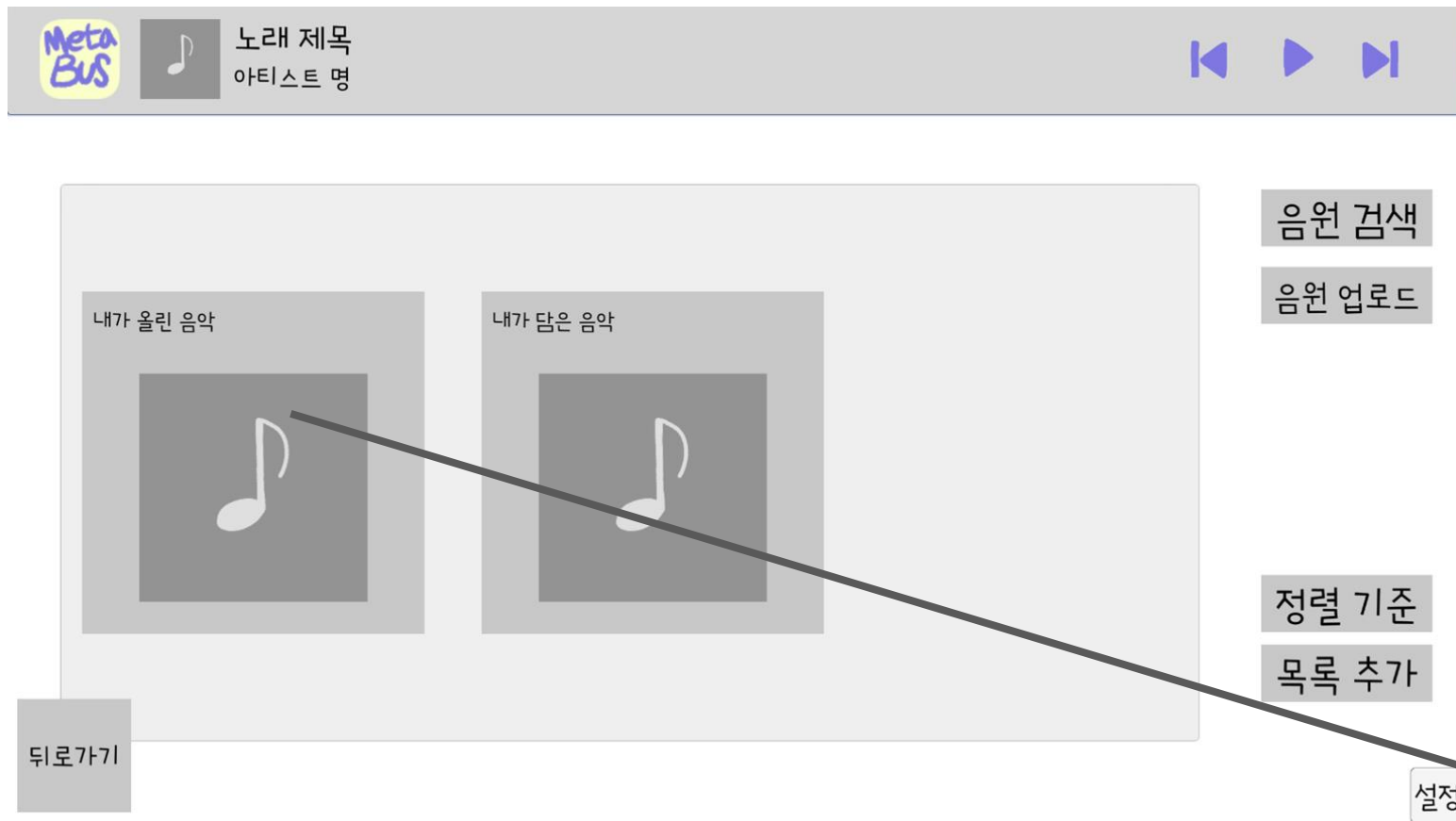
- 상단 음악 재생바를 통해 내 재생목록에 있는 음원을 재생함
- 상세버튼을 클릭하면, 음원 정보 페이지가 나타남.

※ 임시 이미지입니다.

랜덤 재생, 한 곡 반복, 전체 반복 기능



2. 음원 관리, 유저 관리 기능 음원 관리



※ 임시 이미지입니다.

음원 관리 페이지

- 재생 목록과 내가 올린 음원들을 관리하는 페이지

재생목록

- 해당 재생목록에 담겨져있는 음원들을 확인하고 편집
- NodeJS 서버를 통해 음원 리스트를 불러올 수 있음.

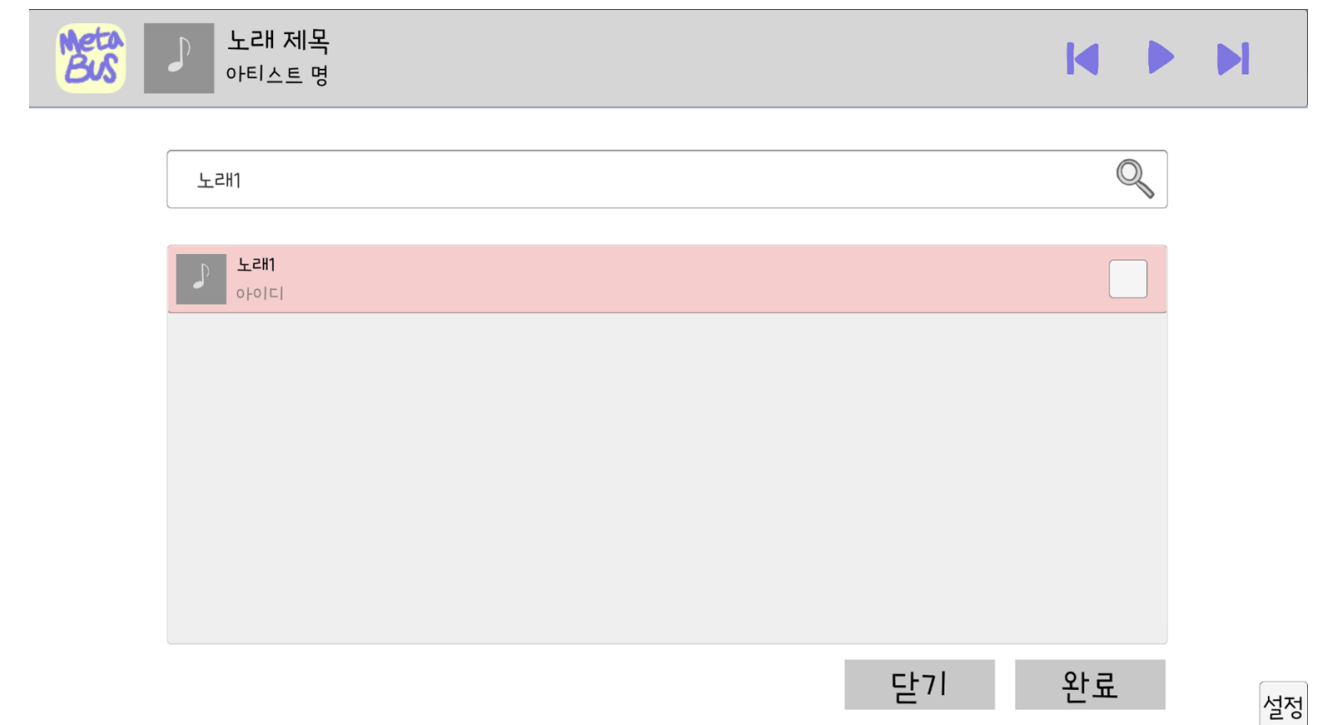
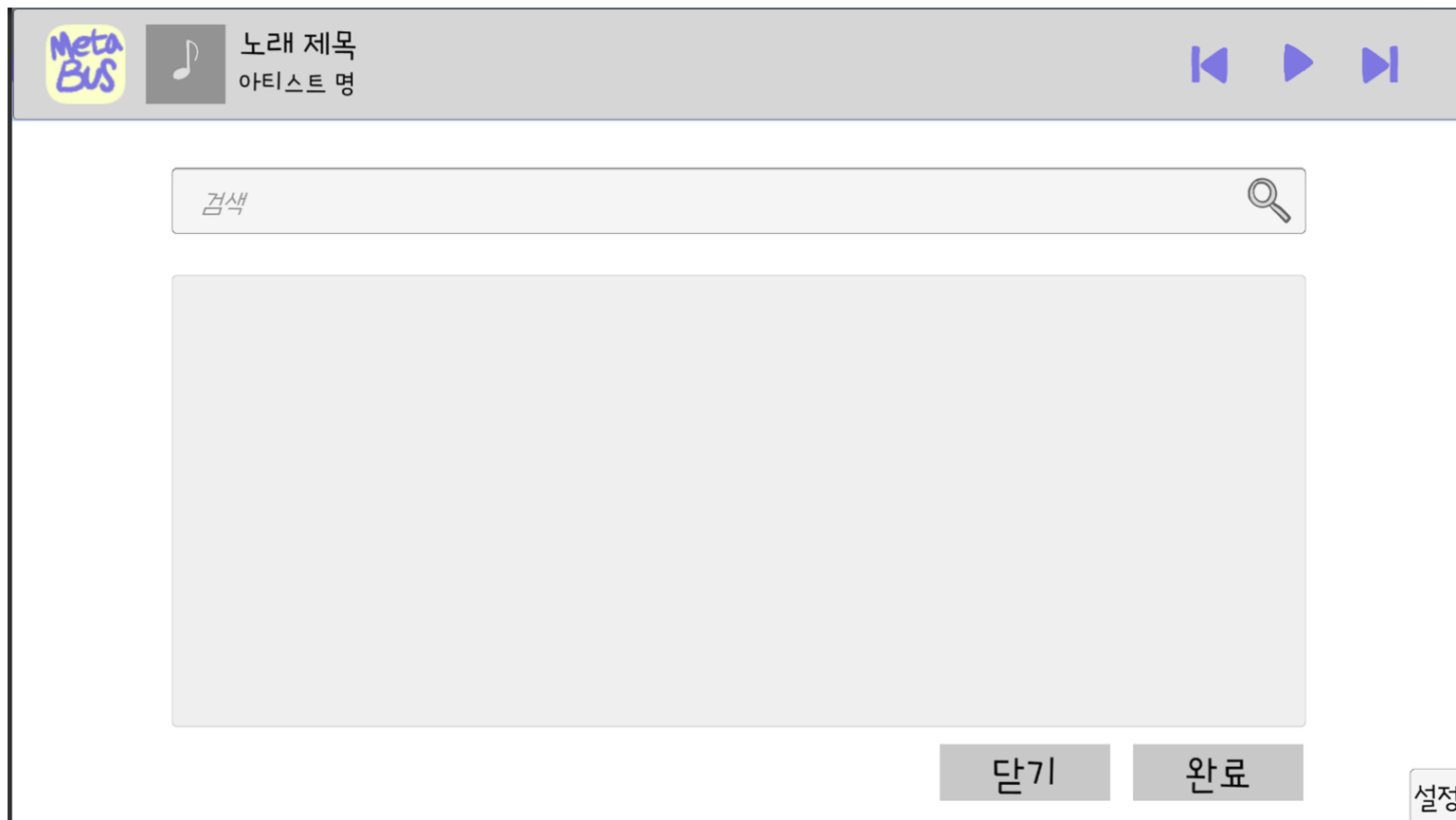


2. 음원 관리, 유저 관리 기능 음원 관리

음원 검색 페이지

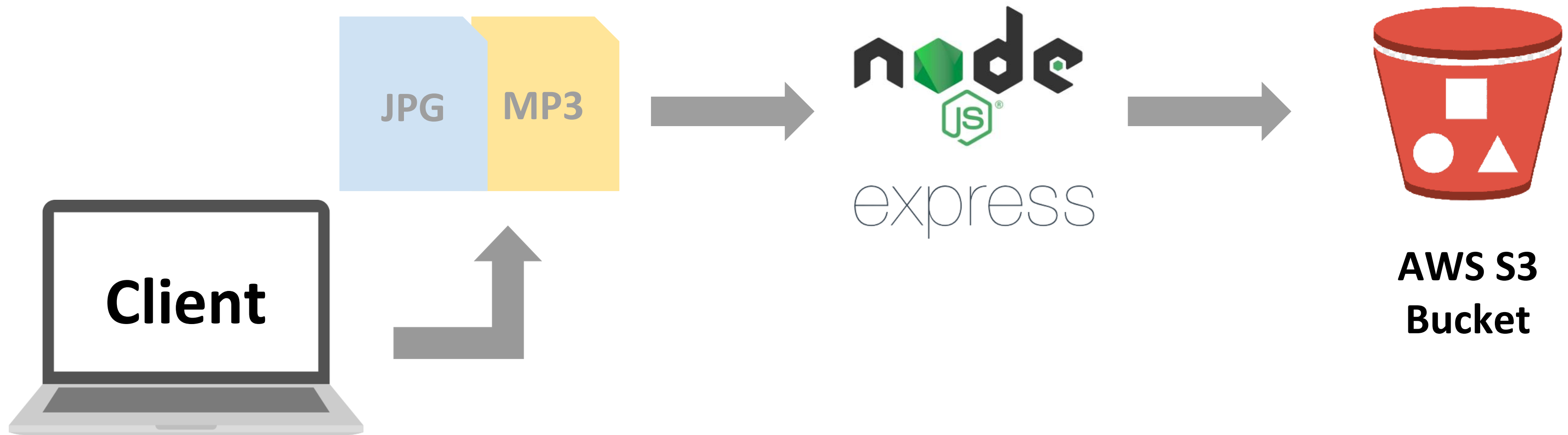
- DB에 있는 모든 음원들의 타이틀 기준 검색
- 재생 목록에 담을 수 있음.
- NodeJS 서버를 통해 음원 리스트를 받아옴.

※ 임시 이미지입니다.



3. 음원 업로드

- 음원 업로드 시 NodeJS 로 만든 서버로 파일이 전달되고 이를 AWS S3 버킷에 저장해줌.



3. 음원 업로드



이미지 변경

파일 :

노래 제목 :

상세 설명 :

○ □ ○ □ ○ □ ○ □

□ ○ ○ □ ○ □ ○ □ ○ □

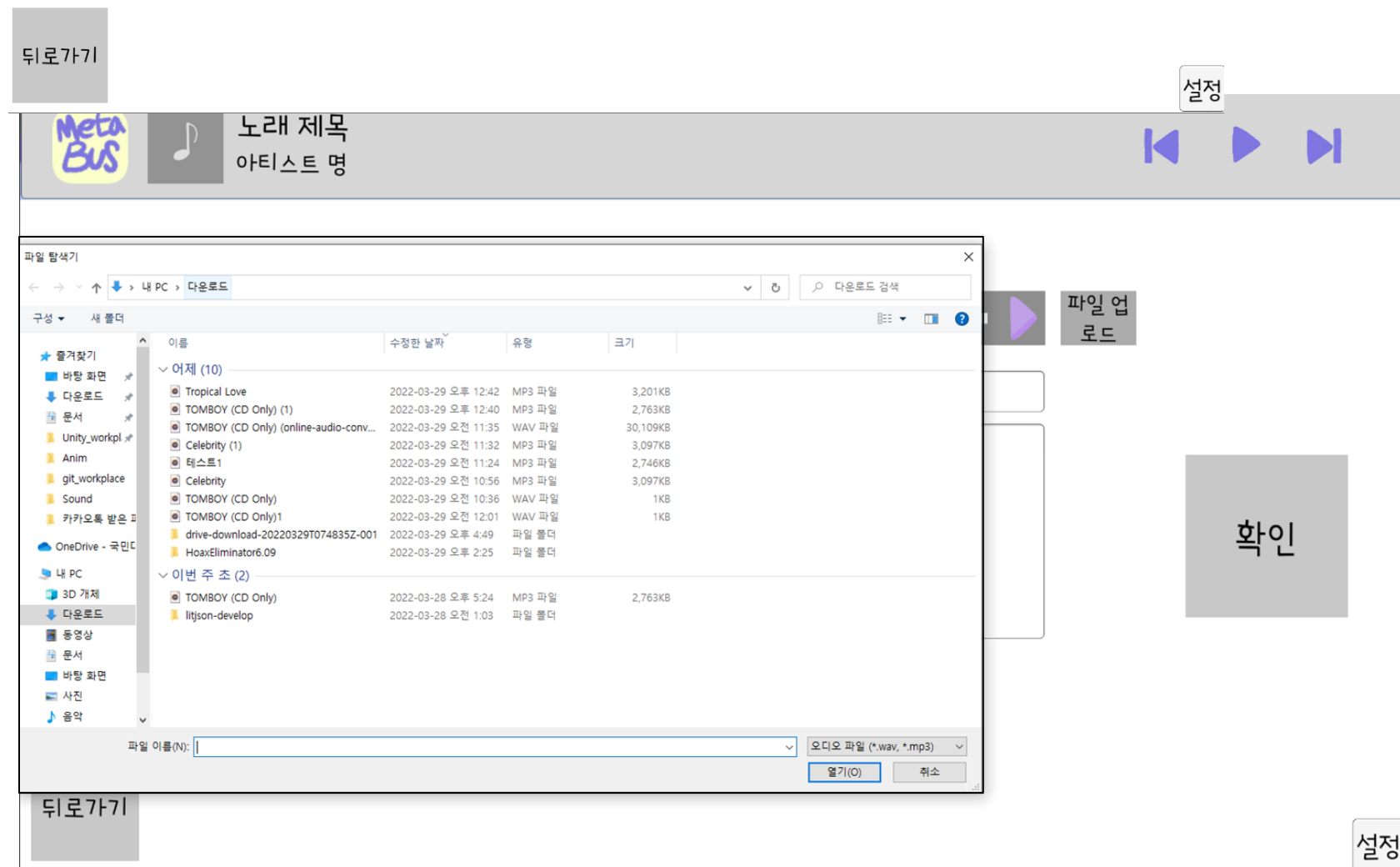
□ ○ ○

확인

음원 업로드 페이지

- 파일 탐색기를 열고 이미지, 음원 파일을 업로드할 수 있음.
- 파일 경로에 해당하는 파일을 byte배열로 변환하여 서버에 업로드

※ 임시 이미지입니다.



이미지 변경

파일 :

노래 제목 :

상세 설명 :

작곡가 : 유저1

작사가 : 유저2

가사 : 이의있음

확인

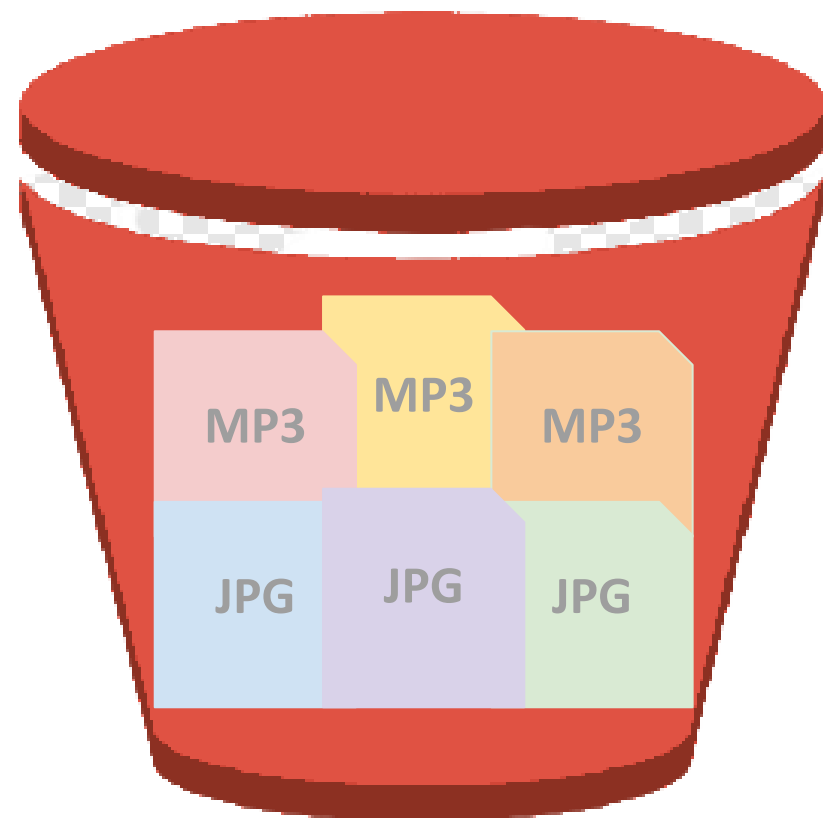
뒤로가기

설정

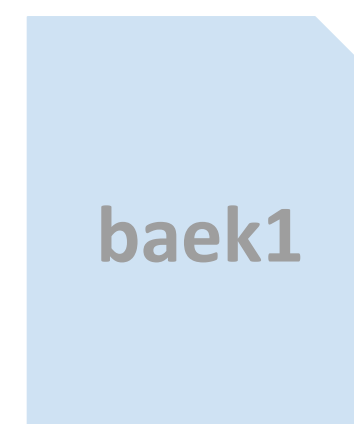
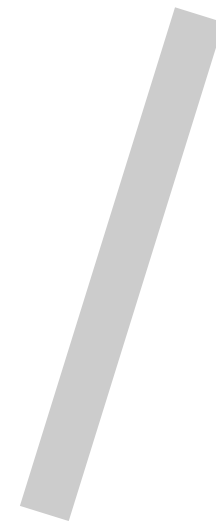
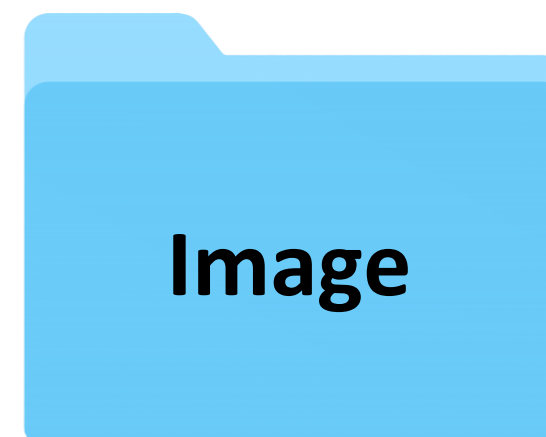
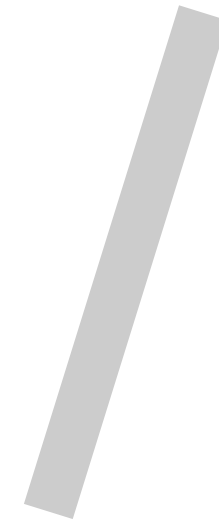
3. 음원 업로드

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

저장 방식



**AWS S3
Bucket**



4. 음원 재생 음원 스트리밍 과정



음악 파일 주소 요청



음악 파일 주소 전달



AWS S3 Bucket에 업로드된
파일 불러오기



AWS S3 Bucket에 업로드된
파일 주소 접근



**AWS S3
Bucket**

5. 멀티 플레이 기능



멀티 플레이 서버

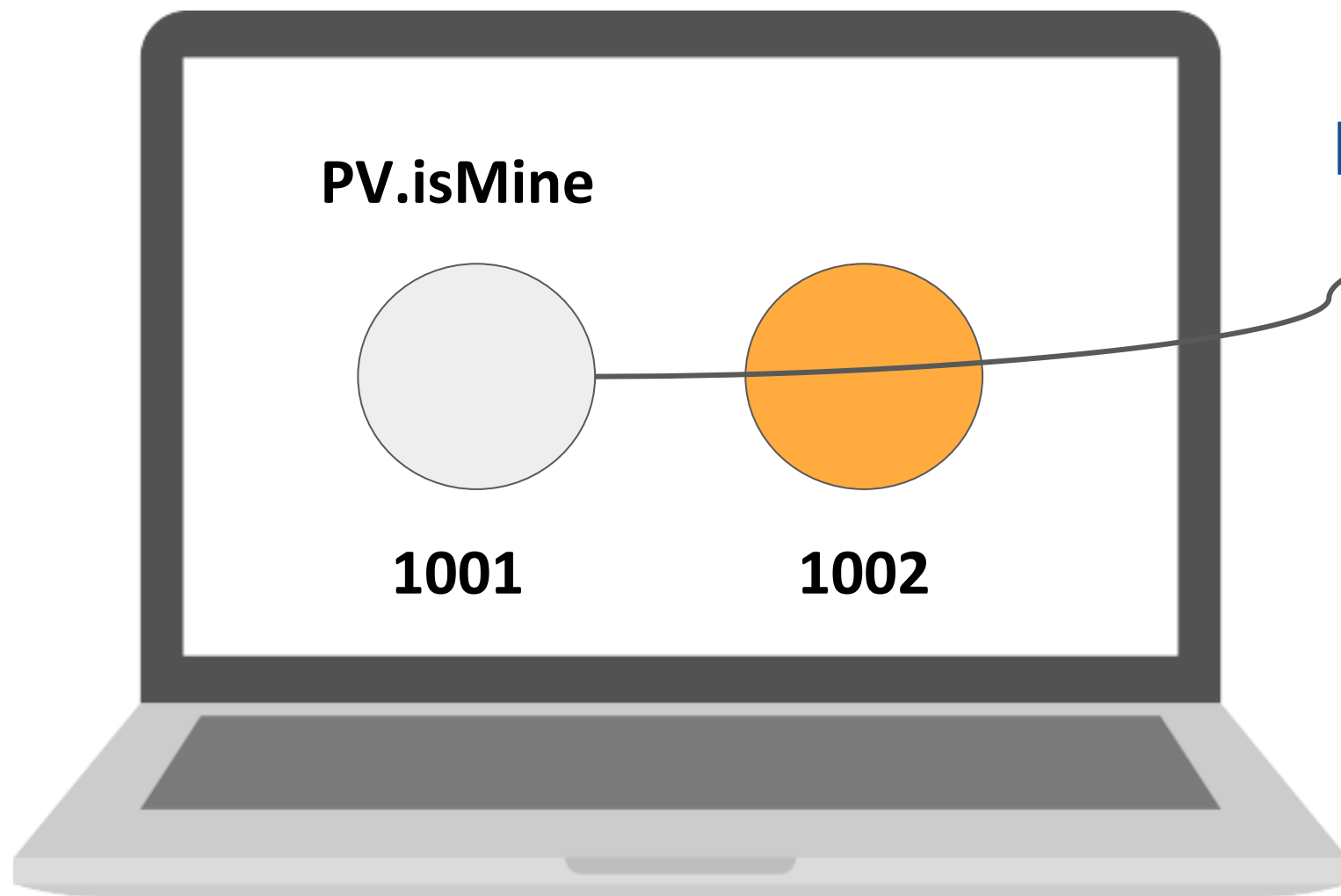
- Photon Cloud 사용
- Unity와 연동 가능
- RPC(Remote Procedure Call) 구조를 통한 클라이언트 간의 동기화 가능

5. 멀티 플레이 기능

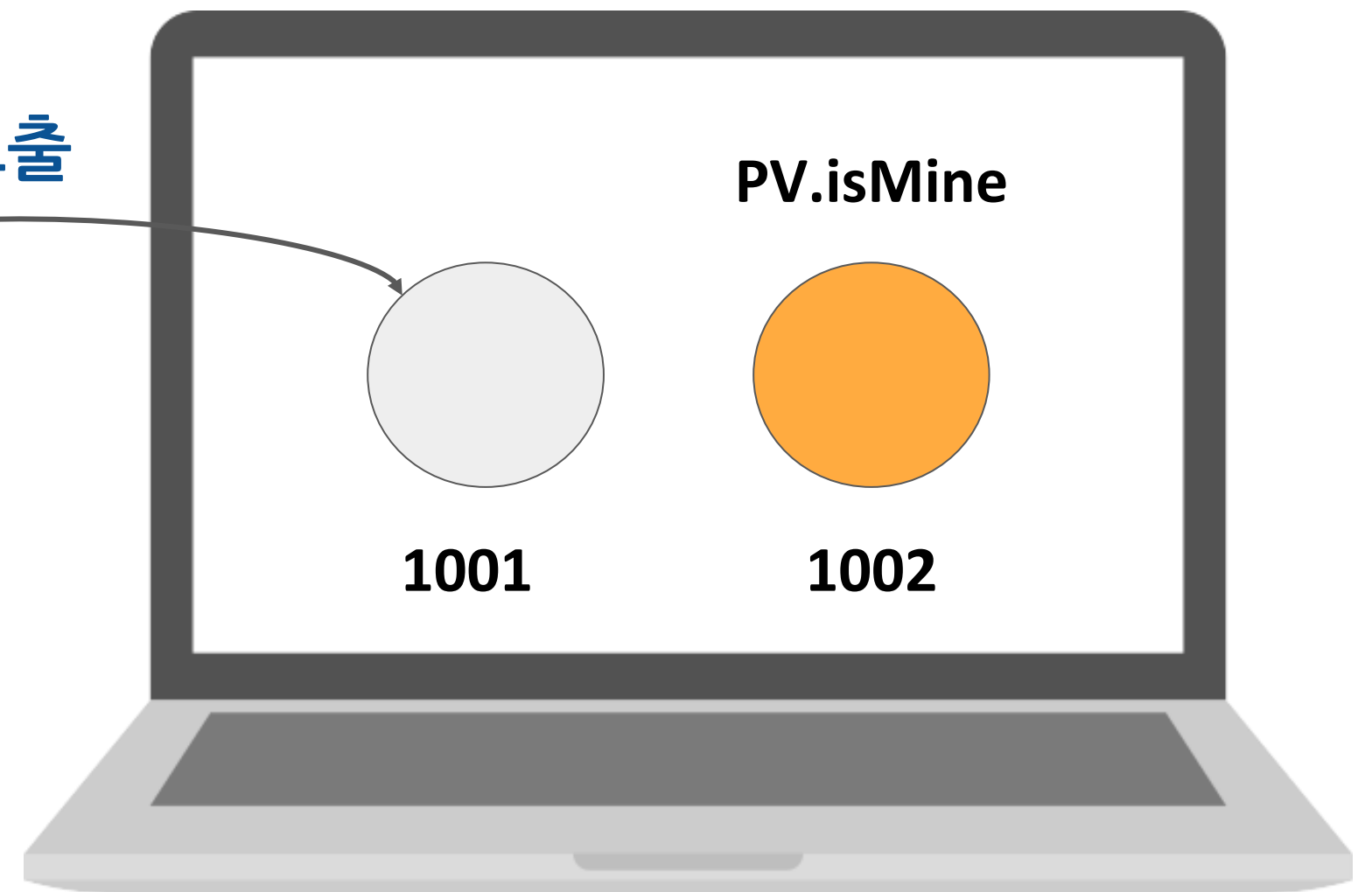


동기화

RPC 함수 호출

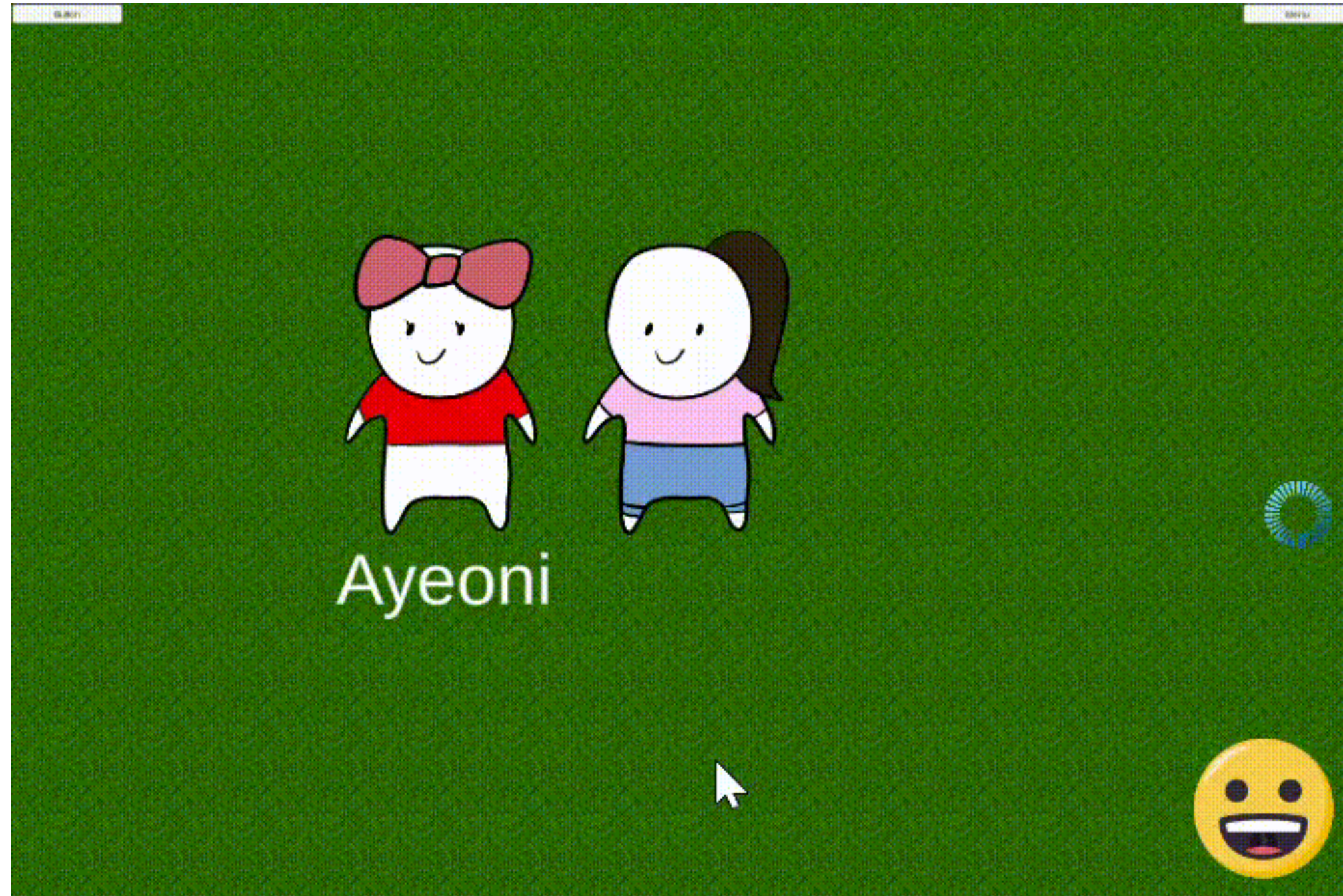


Client A



Client B

5. 멀티 플레이 기능



※ 임시 이미지입니다.

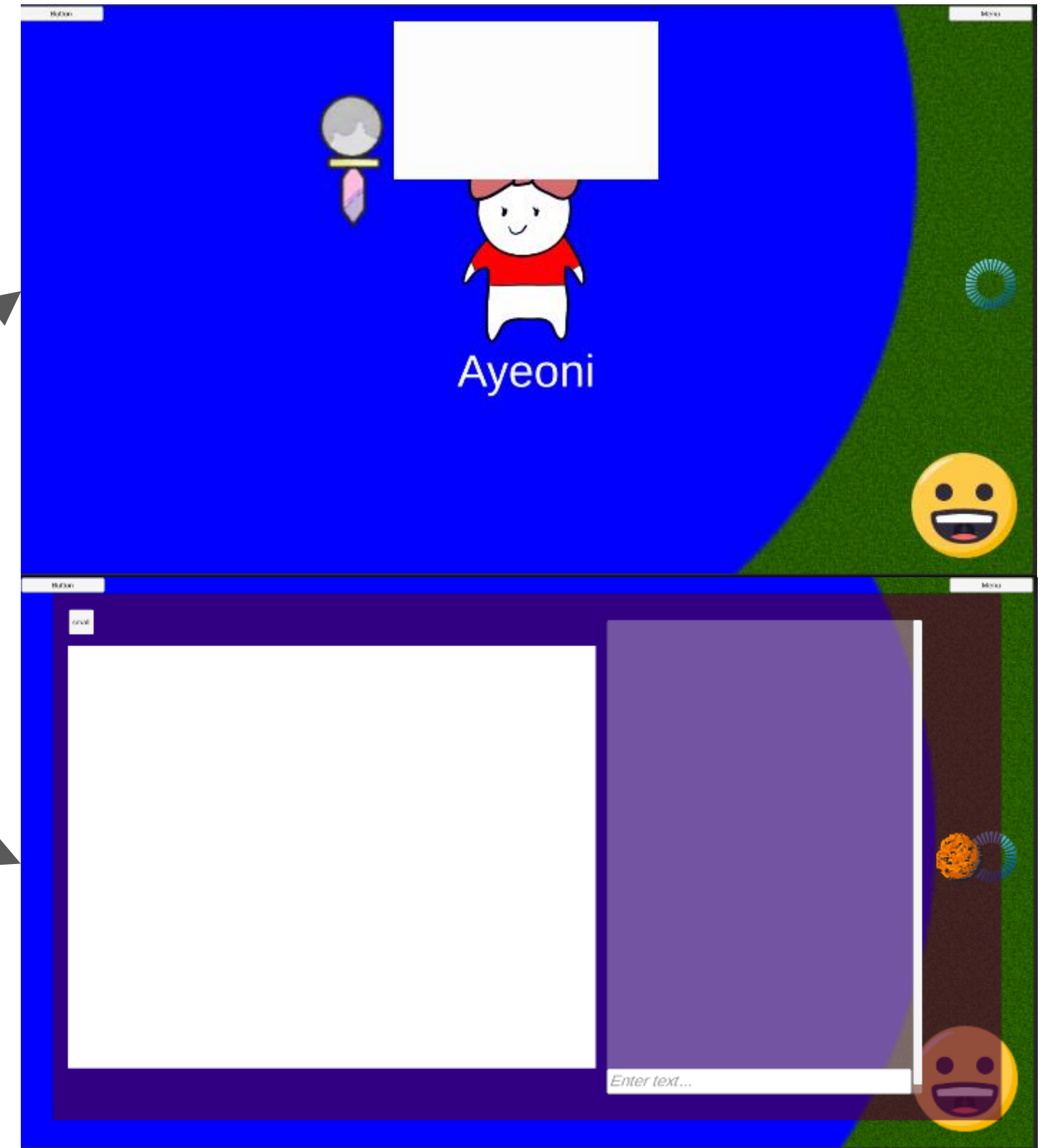
6. 버스킹 기능

※ 임시 이미지입니다.



버스커의 송출

- 본인의 화면과 소리를 공유함



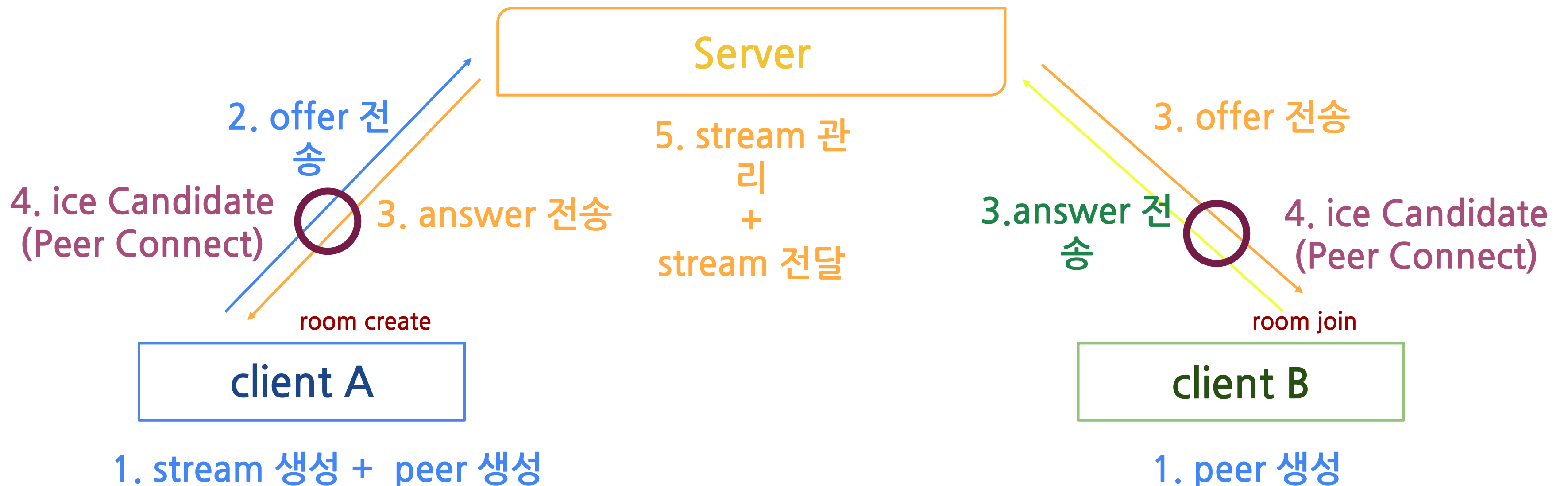
관객

- 버스킹을 진행중인 영역에 들어오면 화면이 수신됨
- 창 축소/확대 가능

6. 버스킹 기능

- WebRTC : 웹 브라우저 환경 및 Android, IOS 애플리케이션에서도 사용 가능한 비디오, 음성 및 일반 데이터가 피어간에 실시간으로 전송되도록 지원하는 오픈 소스

- WebRTC 서버(SFU)의 작동 원리 :



7. 현재까지 구현 완료한 기능

	구현 상태	상세 설명
회원 가입	O	- 유저 로그인 (완료) - 유저 회원 가입 (완료)
음원, 유저 관리 기능	△	- 캐릭터 설정 (완료) - 음원 담기, 재생목록 편집 (진행중)
음원, 이미지 업로드	△	- 파일 업로드 (완료) - 업로드 시 DB에 작성하는 작업 (진행중)
음원 재생	O	- 음원 스트리밍 (완료)
멀티 플레이	△	- 캐릭터 이동 멀티 (완료) - 캐릭터 및 오브젝트 동기화 (진행중) - 이모티콘 동기화 및 버스킹 화면 채팅 기능 (진행중) - 캐릭터 외형 동기화 (미완료)
버스킹	△	- WebRTC 서버 구축 (완료) - 서버와 클라이언트 연동 (진행중)
배포	X	- 현재 AWS 계정을 받지 못하여 배포는 시도하지 못함 (미완료)

Chapter.4

Next Plan

추후 계획

04

Next Plan

메인 메뉴(로그인)

- 회원가입
 - 이메일 인증

공유 공간(광장)

- 자판기
 - 최신 음원
- 순간이동기
 - 버스킹존에서 청중이 가장 많은 구역 표시
 - 해당 버스킹으로 이동하기

로비

- 내 재생목록
 - 사용자 지정 재생 목록
- 음원 관리
 - 음원 좋아요 관리
 - 재생목록 순서 관리
 - 재생목록 편집(추가 삭제)
- 유저 검색
- 팔로잉/팔로워 기능
 - 다른 유저 팔로우 가능
 - 다른 유저의 정보 확인
 - 해당 유저의 플레이 리스트 확인
- 음원 추천 목록

감사합니다