

---

# Mr. Pres

## 기본 설계서

---

한성대학교 컴퓨터공학과 설계 프로젝트

---

FrontFloor

엄두성 / 김강성 / 한종빈 / 조이슬

---

## 목차

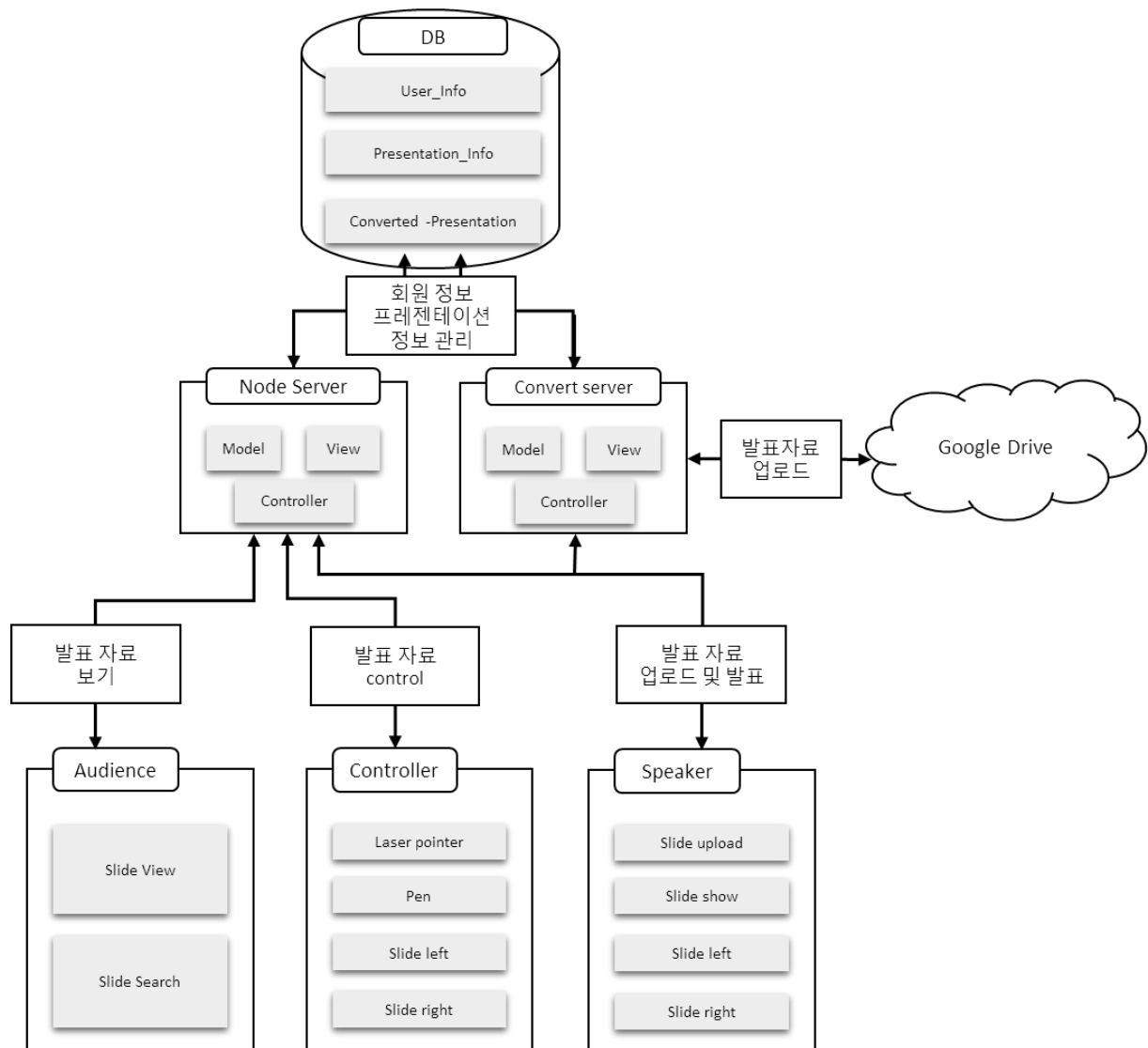
1	개요 .....	3
1.1	전체 시스템 구조 .....	3
1.2	Mr. Pres 시스템의 특징 .....	4
2	구조 .....	5
2.1	용어 정리 .....	5
2.2	기능 요구사항 .....	6
2.3	전체 구조 .....	6
2.3.1	시나리오 .....	6
2.3.2	클라이언트 구조 .....	7
2.3.3	서버 구조 .....	11
3	기능 .....	13

# 1 개요

Mr. Pres 는발표 자료를 웹에 업로드하여, 인터넷이 가능한 어느 곳에서도 프레젠테이션을 할 수 있는 온라인 프레젠테이션 서비스이다.

## 1.1 전체 시스템 구조

위의 설계 목표를 만족시키기 위하여 Mr. Pres 의 시스템 구조를 <그림 1-1>과 같이 제안한다.



<그림 1-1> 전체 시스템 구조도

## 1.2 Mr. Pres 시스템의 특징

### A. 경제성

상용 리모콘의 경우 따로 구매 비용이 드는데 본 프로젝트의 경우 기존 사용자의 Mobile 을 사용함으로써 따로 리모콘의 구매가 필요 없어 비용을 절감하였다.

### B. 윤리성

어플리케이션의 저작권 문제만 없으면 다른 비윤리적인 문제는 없고, 사용자는 개인의 웹에서 제공되는 애플리케이션을 사용하면 되므로 문제점은 없다.

### C. 안전성

본 프로젝트 결과물은 웹 서비스이므로 특별한 위험성은 없다.

### D. 신뢰성

본 프로젝트의 경우 PC, 모바일 등 모든 디바이스에서 정상적으로 실행이 가능하다. 하지만 통신의 문제로 원격 부분에 있어 접속에 대한 오류가 생길 수 있다. 응용 어플리케이션에 대한 신뢰성 문제는 존재하지 않는다.

### E. 환경에 미치는 영향

본 프로젝트의 결과물은 무형의 소프트웨어이므로 환경적으로 영향을 끼치지 않는다.

## 2 구조

Mr. Pres 는 프레젠테이션을 위한 온라인 웹 서비스이다. 이 장에서는 앞 장에서 설명한 설계 목표를 달성하기 위한 기능 요구 사항을 간략히 기술하고 그 기능을 구현하기 위한 개념적 구조를 기술한다.

### 2.1 용어 정리

용어	정의
사용자	회원과 비회원 모두를 이르는 말 회원 또는 비회원만 필요한 특정 경우는 따로 지칭하여 쓴다
회원	회원 가입 후 서비스를 이용하는 사람
비회원	방문자와 동일 회원 가입을 하지 않고 서비스를 이용하는 사람
프레젠테이션	PPT 혹은 PDF, ODF 등의 형태로 제작되는 발표 자료
슬라이드	프레젠테이션의 각 페이지
슬라이드 쇼	전체 화면으로 프레젠테이션을 보는 것
발표	프레젠테이션의 싱크를 공유하거나 리모트 컨트롤러를 연결하는 것
발표자	특정 프레젠테이션을 발표할 수 있는 사람
싱크 공유	발표자 중 현재 프레젠테이션의 상태(보고 있는 슬라이드)를 다른 사용자에게 공유 싱크 공유자가 아닌 사용자는 싱크 공유된 프레젠테이션의 상태를 보거나 자신이 보고 싶은 페이지를 볼 수 있음
뷰어	사용자의 프레젠테이션을 볼 수 있는 기능을 지칭

## 2.2 기능 요구사항

웹 브라우저 기반 소프트웨어로서 Mr. Pres 는 다음 표와 같은 총괄적인 기능들을 요구하고 있다. 이 총괄 기능 정의는 다음 절의 Mr. Pres 의 개념적 총괄 구조를 설명하기 위한 것이며 각 총괄 기능에 대한 세부적인 기능은 제 3 장 Mr. pres 의 기능에서 제시된다. 제 2 장의 나머지 부분에서는 각 기능을 구현하기 위한 개념적 세부 구조를 기술한다

기능	기능의 간략한 설명
SNS 로그인	구글 API 를 통한 로그인 기능
발표 자료 업로드	PPT, PDF, ODP 형식의 발표자료를 업로드 할 수 있는 기능.
발표 자료 저장	업로드 된 자료를 사용자의 구글 드라이브에 저장해주는 기능
발표 자료 실행	업로드 된 자료를 사용자의 구글 드라이브에 저장해주는 기능
리모트 컨트롤러	Mobile Device 를 통하여 PC 의 발표자료를 컨트롤 하는 기능
화면 공유	발표자의 현재 발표 슬라이드를 자신의 PC 에서 실시간으로 볼 수 있는 기능

<표 2-1> Mr. Pres 의 기능

## 2.3 전체 구조

### 2.3.1 시나리오

#### I. 사용자:

Mr. Pres 서비스를 통해 회원 가입을 할 수 있고, SNS 계정을 통한 회원 가입 또한 할 수 있다.

#### II. 회원:

##### i. 발표 자료 업로드, 리모트 컨트롤러와 화면 공유 사용 발표

A 씨는 어느 컨퍼런스에서 발표를 할 예정이다.

발표를 위해 Mr. Pres 서비스에 미리 발표 자료를 업로드해 두었다.

발표 당일, A 씨는 Mr. Pres 서비스에 접속하여 올려 놓은 발표 자료를 확인하였다.

발표 시간, 발표 장소는 매우 넓어 발표를 시작할 때 발표자 옵션에서 화면 공유 기능과 리모트 컨트롤러 기능을 활성화 하고, 발표를 시작하였다.

발표를 시작하면서 스마트폰을 꺼내어 리모트 컨트롤러를 실행하여 발표자료와 연결하고, 발표를 듣는 청중들에게 Mr. Pres 에 접속하여 슬라이드를 볼 수 있는 방법을 안내하였다.

ii. 기존 업로드 발표 자료 사용 발표

B 씨는 어느 소모임에서 팀원 A 씨가 발표한 프레젠테이션을 발표하게 되었다.

B 씨는 Mr. Pres 서비스에 접속하여 A 씨 이름으로 발표 자료를 검색하였다.

검색 된 자료 결과 중 A 씨의 해당 자료를 선택 후 프레젠테이션 시작 버튼을 눌러 발표를 시작 하였다.

III. 비회원(회원도 가능):

i. 발표자와 화면 공유가 되는 발표 듣기

C 씨는 어느 컨퍼런스에 A 씨의 발표를 들으러 갔다. 서둘러 갔지만 예상보다 늦게 도착한 C 씨는 상당히 뒤에 앉는 바람에 A 씨의 발표 자료가 잘 보이지 않아 발표를 듣지 않을까 했지만, A 씨가 Mr. Pres 서비스에 접속하여 발표 자료를 검색하여 접속하면 현재 발표자가 발표중인 화면을 실시간으로 볼 수 있다고 안내해주어 발표에 집중할 수 있었다.

ii. 발표 자료 보기

지난주에 발표한 자료가 다시 보고 싶어진 C 씨는 Mr. Pres 서비스에 접속하여 발표 자료를 검색하여 해당 자료를 다시 볼 수 있었다.

## 2.3.2 클라이언트 구조

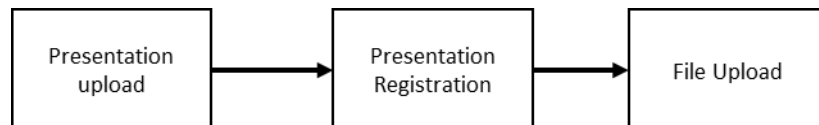
### I. 발표자

i. 발표 자료 업로드

① 설계 목표

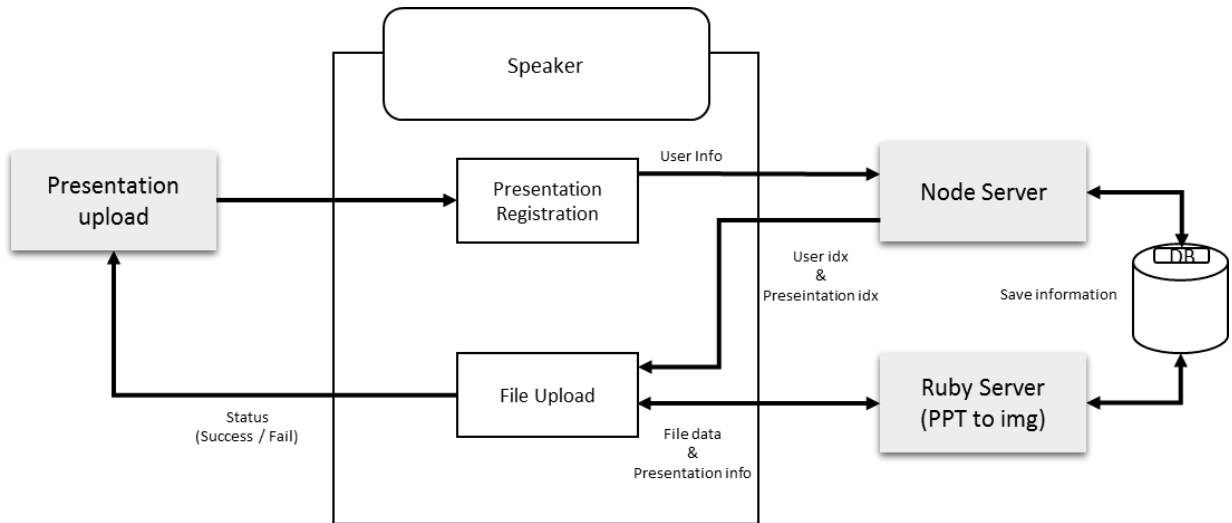
발표자료는 프레젠테이션 발표를 하기 위한 것이다. 발표자가 올린 파일과 발표자의 정보를 가지고 파일을 업로드하도록 구현한다.

② 구조



발표자료 업로드에서 모든 과정은 프레젠테이션 등록(Node server) 후 해당파일을 업로드하는 방식으로 진행된다.

아래 그림에서 보면 Node 서버의 결과(U-idx, P-idx)를 가지고 Ruby 서버에 프레젠테이션 파일을 Mr.에 맞는 양식으로 ppt 를 변환하는 요청을 처리한다. 처리결과는 파일이 정상적으로 업로드가 되었는지 실패하였는지 확인하는 status 를 돌려주게 된다.



## ii. 발표 자료 저장

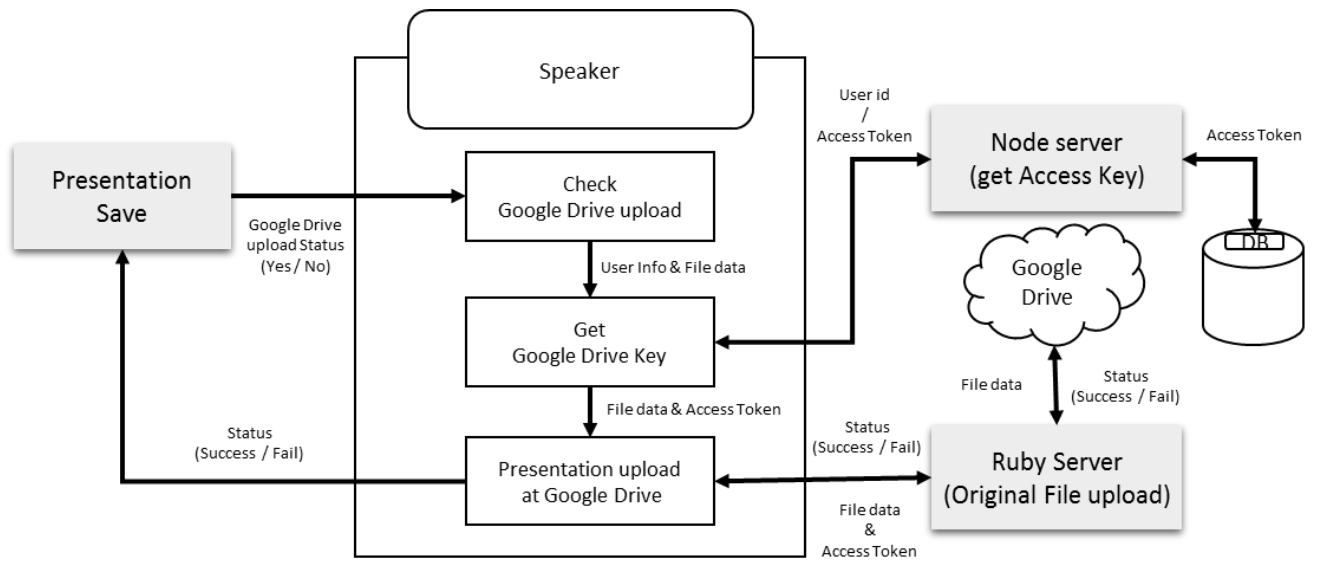
### ① 설계 목표

발표자료를 발표자의 Google Drive 에 upload 하는 process 는 사용자가 발표자료를 저장을 원하는 경우 파일저장을 제공하기 위한 것이다.

### ② 구조

아래 그림에서 보면 우선 google drive 에 프레젠테이션 파일을 업로드 할지를 확인 후 Google Key(Access Token)를 Node server 에서 받아온다. 받아온 Access Token 을 가지고 Ruby 서버에서 Google Drive 에 파일을 업로드를 요청 한다. Access Token 값이 없거나 Drive 에 업로드를 실패 시 status 는 fail 을 return 되고 성공 시 Success 를 돌려주게 된다.





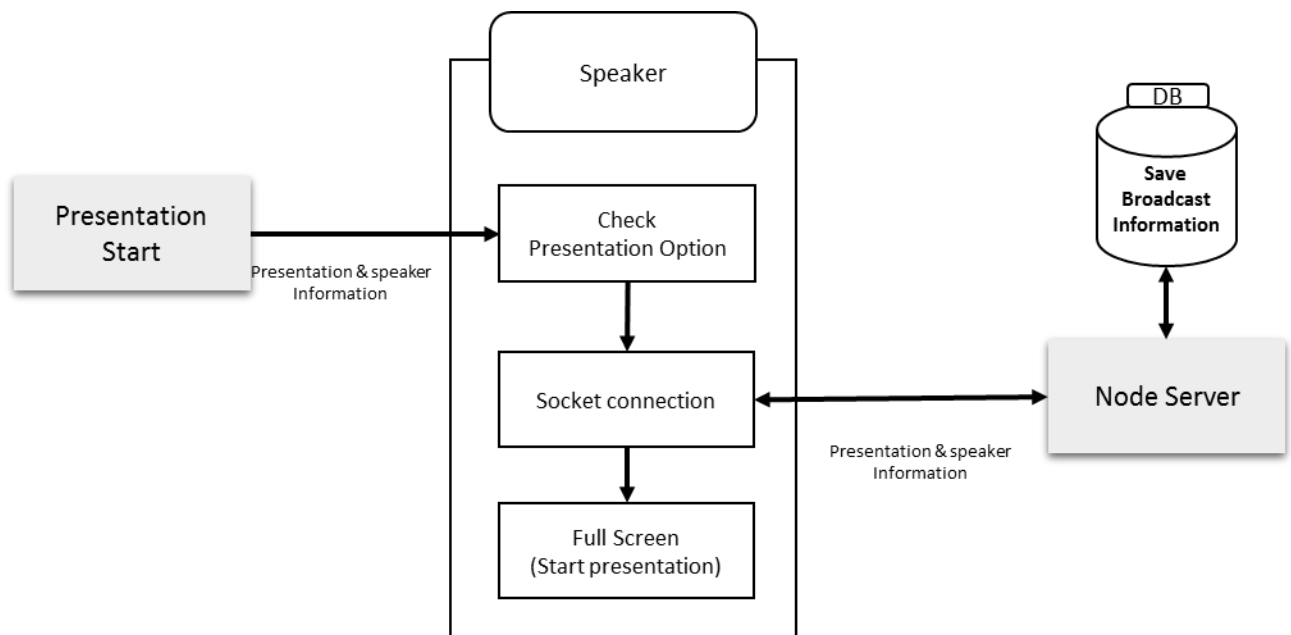
### iii. 발표 자료 실행

#### ① 설계 목표

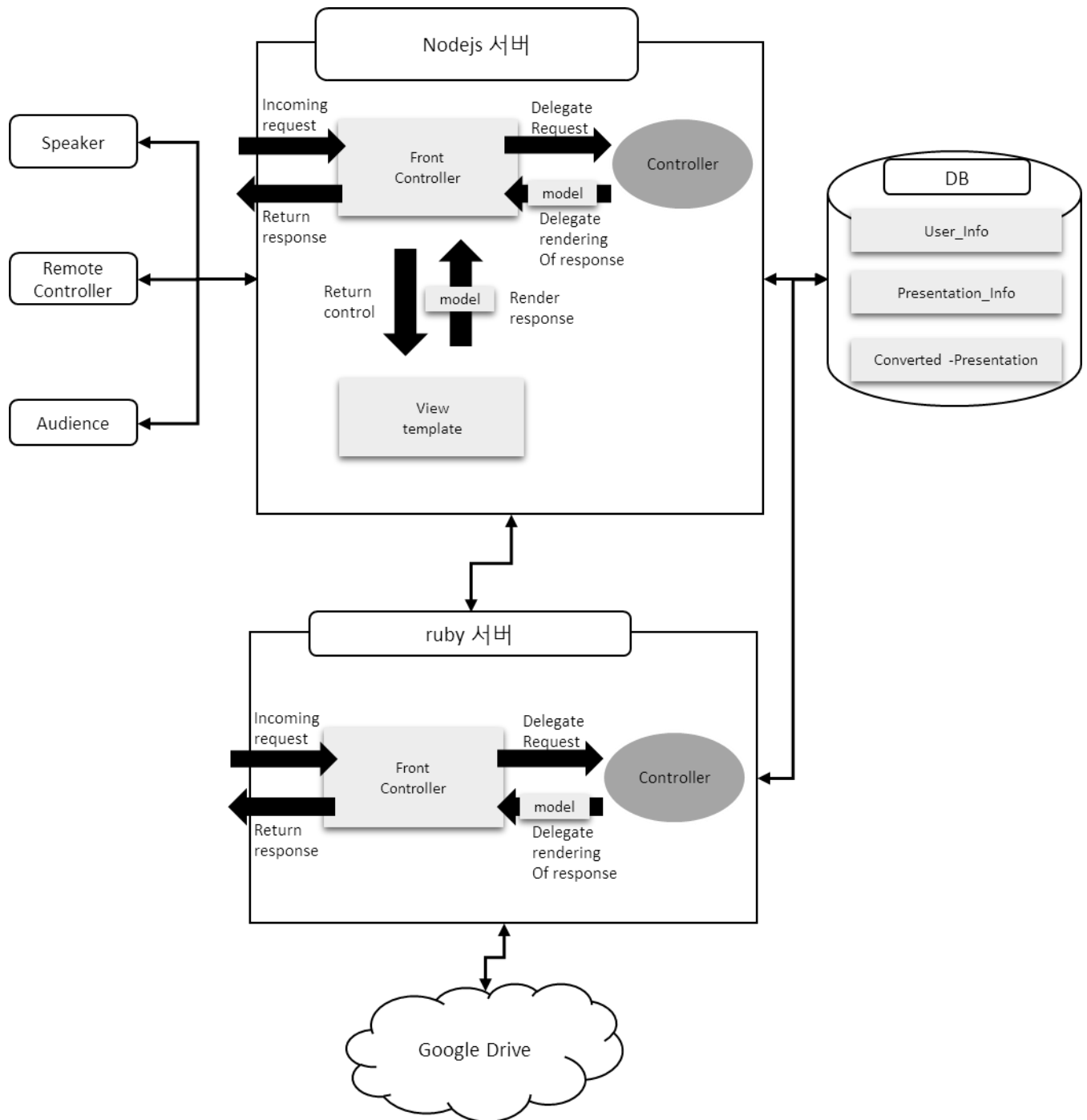
발표자의 Presentation 을 제공하기 위함이다.

#### ② 구조

발표 시작을 하면 우선 발표 옵션을 확인한다. Option 은 Sync 공유와 Remote Controller 사용 두 가지가 있다. 해당 옵션과 발표자료 정보를 node 서버에 보내 socket 을 연결을 한다. 하지만 option 이 모두 꺼져있는 경우 socket 연결하지 않는다. 그 후 발표를 할 수 있게 화면은 full screen 으로 변경된다.



### 2.3.3 서버 구조



## I. Node 서버

- i. Speaker 가 화면공유 기능 사용 요청 시, 해당 기능 사용을 승인한다.  
웹소켓을 사용하여 다른 사용자가 화면을 공유 받을 수 있도록 한다.
- ii. Speaker 가 Remote Controller 사용 요청 시, 해당 기능 사용을 승인한다.  
Speaker 가 모바일 기기를 Remote Controller 로 사용할 수 있도록 특정 페이지 주소를 리턴 해준다.  
웹소켓을 사용하여 Speaker 의 발표자료와 Remote Controller 를 연결 시켜준다.
- iii. Remote Controller 에서 슬라이드 이동 이벤트가 발생하면 Speaker 의 발표자료에 해당 이벤트를 전달해준다.
- iv. Audience 가 Speaker 의 발표를 듣기 위해 발표자료에 연결을 요청하면 승인 한다.
- v. DB 서버와 통신한다.
- vi. 사용자가 발표자료를 업로드하면 Ruby 서버로 해당 자료를 전달한다.

## II. Ruby 서버

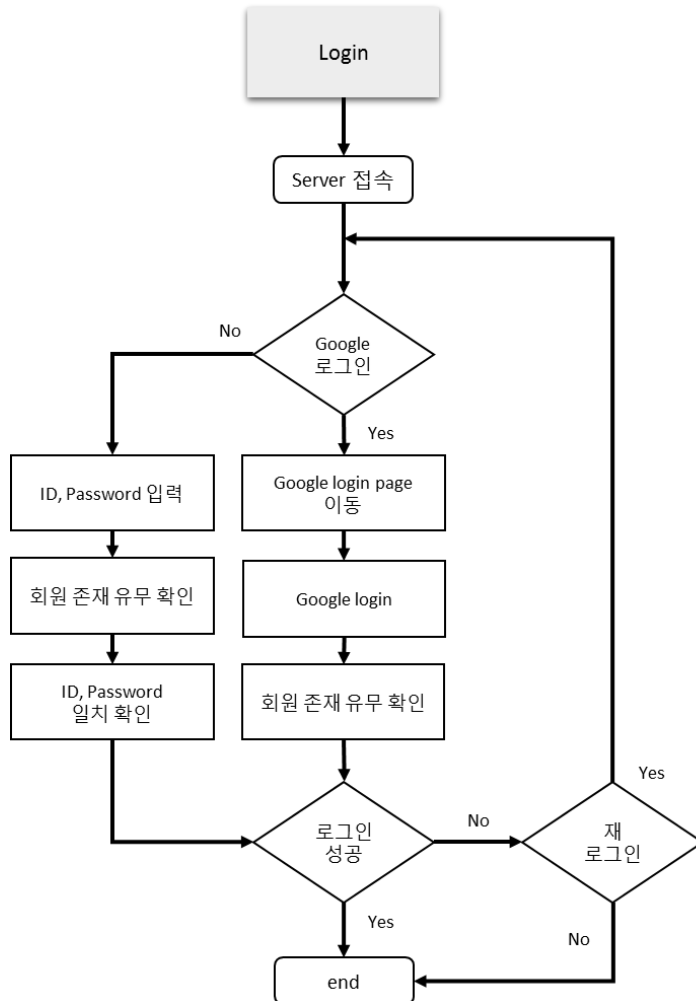
- i. node 서버로부터 업로드 할 발표자료를 전달 받는다.  
발표자료 업로드를 요청한 사용자의 구글 드라이브에 업로드한다.
- ii. 발표자료의 용량을 줄이기 위하여 발표자료를 이미지로 변환한다.
- iii. 변환 된 이미지를 서버에 저장한다.

## III. DB 서버

- i. 회원가입 시 사용자의 정보를 저장한다.
- ii. 사용자가 업로드 한 발표자료 정보를 저장한다.
- iii. 변환 된 발표자료 이미지파일 정보를 저장한다.

### 3 기능

#### 3.1.1 로그인



login\_1: 로그인을 google 로그인할지 local 로그인을 할지 선택한다

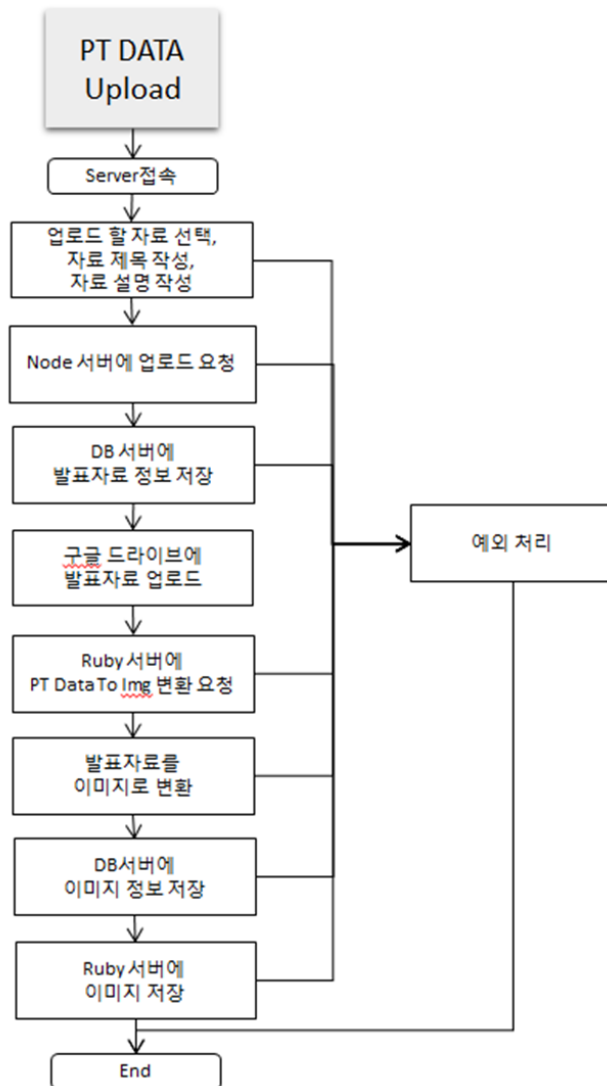
login\_2-1: google 로그인시 google페이지에 이동하여 google 로그인을 한다

login\_2-2: local 로그인시 자신의 ID Password를 입력한다.

login\_3-1: google 로그인에 해당 id가 Mr. Press의 사용자인지 확인한다. 존재 하지 않을 시 재 로그인 요청후 처리한다.

login\_3-2: local 로그인에 입력된 ID존재유무, Password 일치 여부를 확인한다. 존재하지 않는 id이거나 password가 틀릴 시 재 로그인 요청 후 처리한다.

### 3.1.2 발표자료 업로드



PTData\_Upload\_1: 업로드 할 파일을 선택한다. 업로드 할 자료의 제목과 설명을 작성 한다.  
업로드 하는 파일의 파일형식이 지정된 형식과 다를 경우 에러 처리한다.  
자료의 제목이 작성되지 않은 경우 에러 처리한다.

PTData\_Upload\_2: 선택된 파일과 제목, 설명을 Node 서버에 업로드 요청한다.

PTData\_Upload\_3: DB서버에 발표 자료 정보를 저장한다.

PTData\_Upload\_4: 업로드를 요청한 사용자의 구글 드라이브에 발표자료를 저장한다.

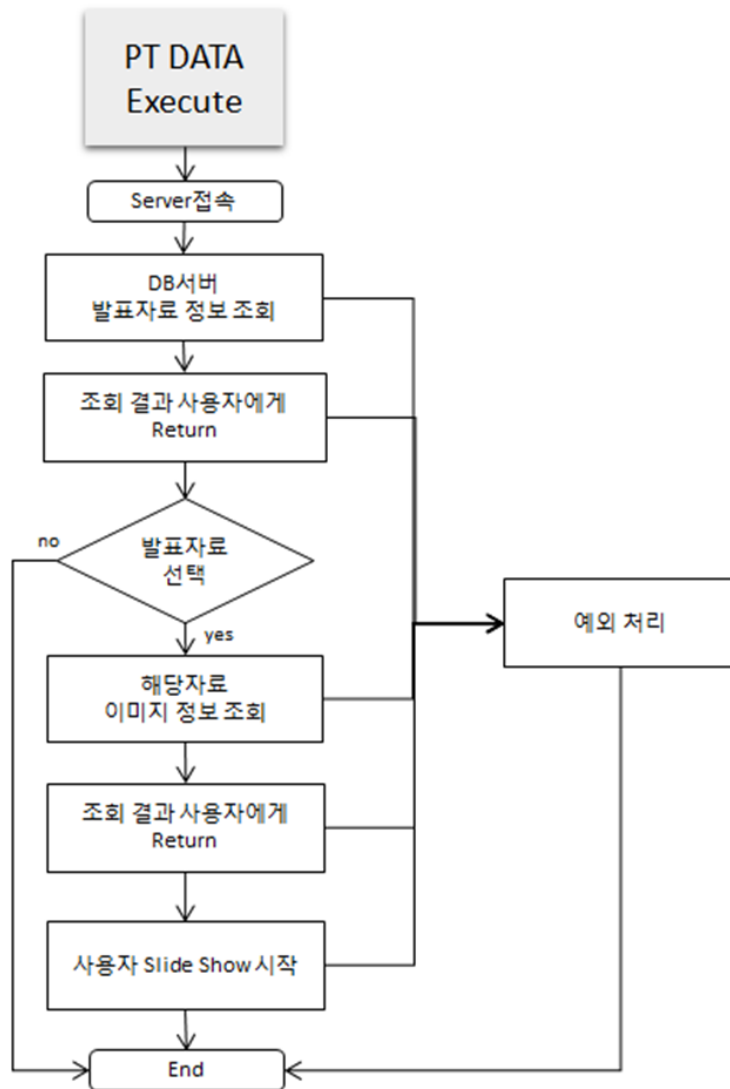
PTData\_Upload\_5: Ruby 서버에 발표자료를 이미지 파일로 변환 요청을 한다.

PTData\_Upload\_6: 발표자료를 이미지로 변환한다.

PTData\_Upload\_7: DB서버에 이미지 정보 저장

PTData\_Upload\_8: Ruby 서버에 이미지파일 저장.

### 3.1.3 발표자료 실행



PTData\_Execute\_1: DB서버에서 발표자료 정보를 조회한다.

PTData\_Execute\_2: 조회결과를 사용자에게 Return 해준다.

PTData\_Execute\_3: 사용자는 조회 결과 중 프레젠테이션에 실행하고자 하는 자료를 선택한다.

PTData\_Execute\_4: 사용자의 자료 선택 유무를 확인 한다.

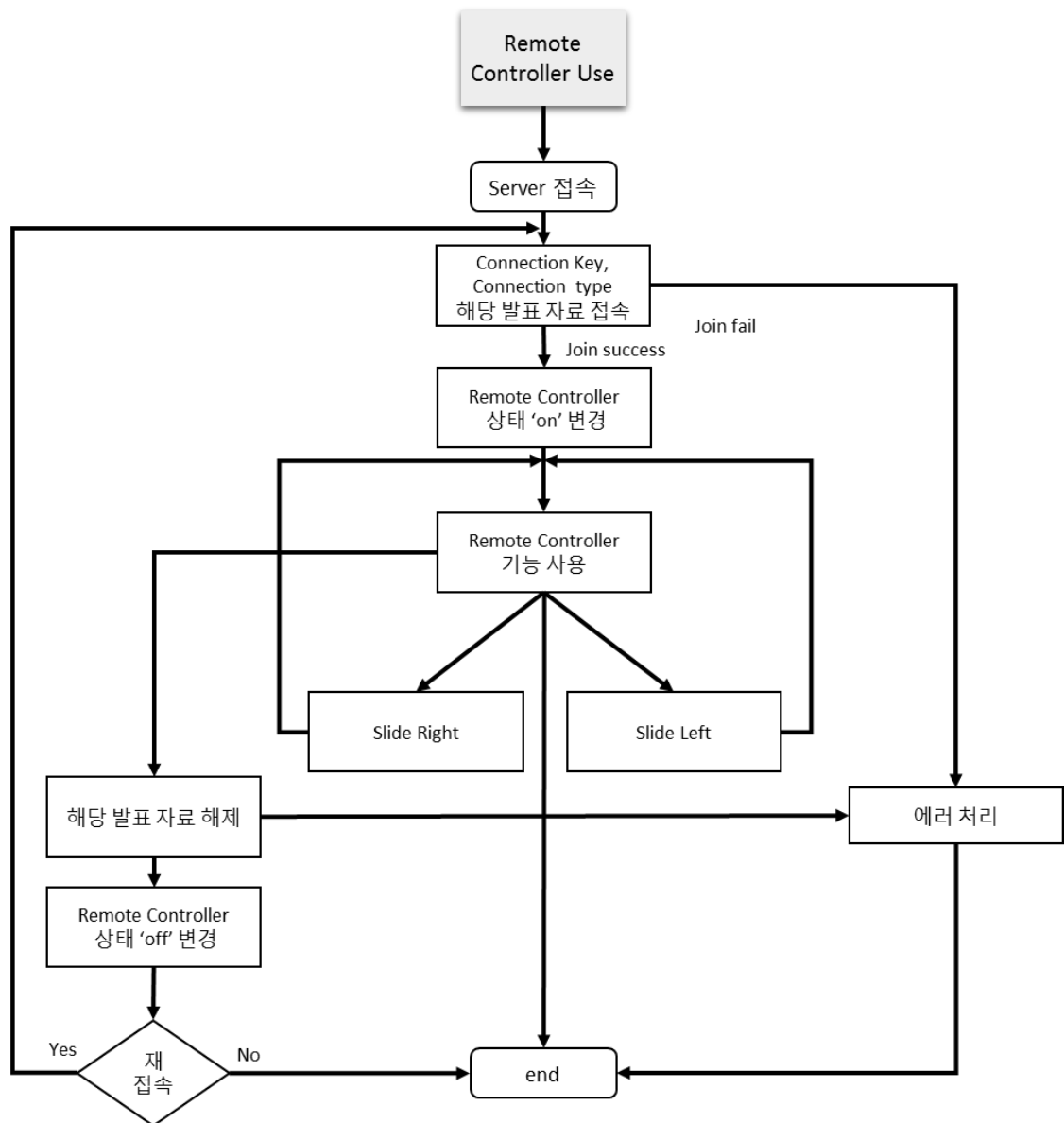
자료를 선택하지 않았다면 해당 기능은 종료 한다.

PTData\_Execute\_5: 사용자가 선택한 자료를 보여주기 위하여 DB서버에서 해당 자료의 이미지 정보를 조회 한다.

PTData\_Execute\_6: 이미지 파일을 Ruby 서버에서 사용자에게 Return 해준다.

PTData\_Execute\_7: 사용자는 Slide Show 누름으로써 해당 자료로 발표를 시작한다.

### 3.1.4 리모트 컨트롤러



REMOTECONTROLLER\_USE\_1: Connection key 값과 Connection Type을 가지고 해당 발표 자료에 접속한다. 접속 실패 시 에러를 처리한다.

REMOTECONTROLLER\_USE\_2: Remote Controller의 상태를 'on'으로 변경한다.

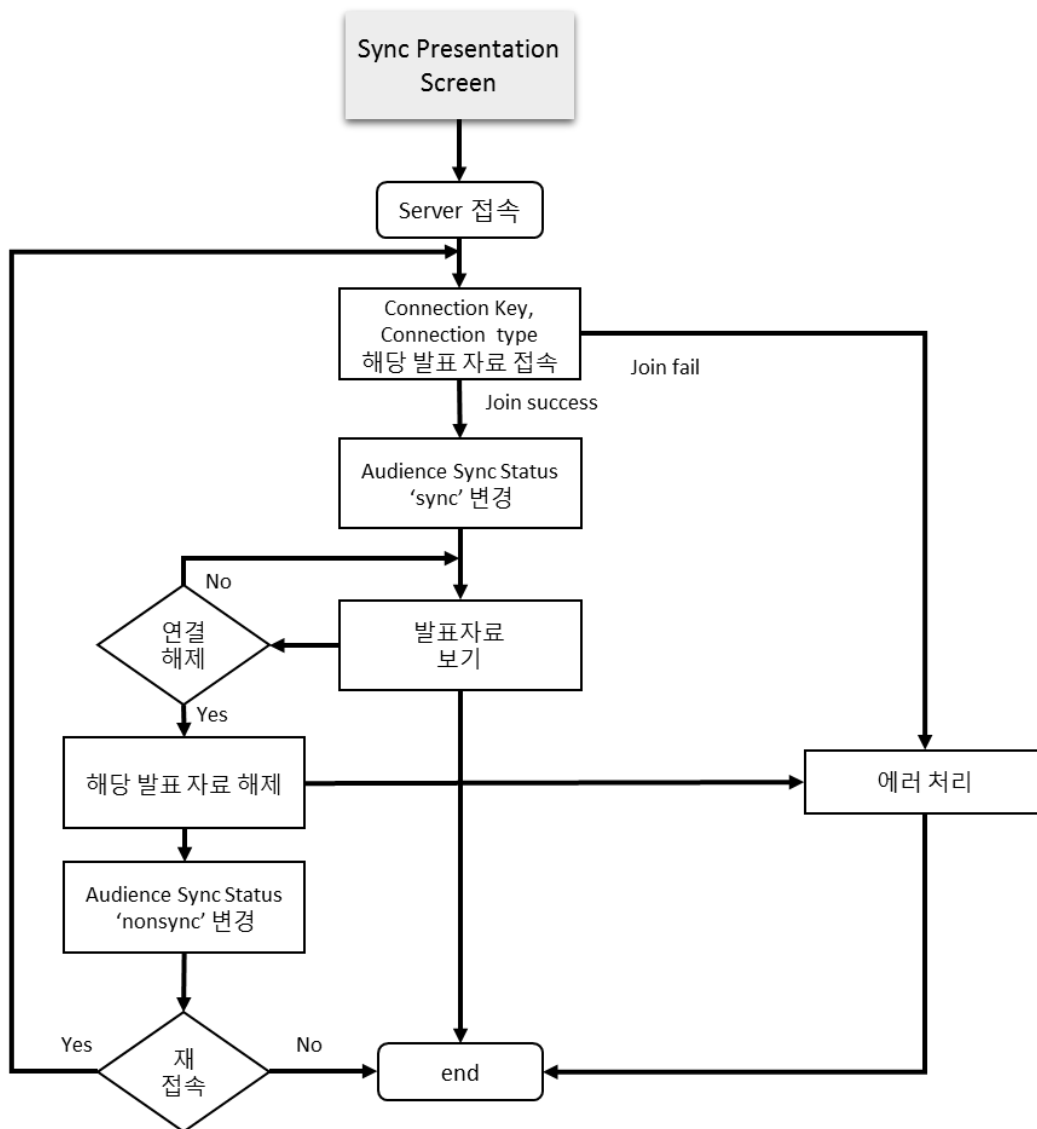
REMOTECONTROLLER\_USE\_3-1: Remote Controller의 기능(slide left, right)을 반복하여 사용한다.

REMOTECONTROLLER\_USE\_3-2: Remote Controller의 연결을 해제 한다

REMOTECONTROLLER\_USE\_4: Remote Controller연결을 원할 시 처음(접속)부터 시작한다.



### 3.1.5 화면 공유



Sync\_Screen\_1: Connection key값과 Connection Type을 가지고 해당 발표자료에 접속한다.  
접속 실패 시 에러를 처리한다.

Sync\_Screen \_2: Audience의 상태를 'sync'으로 변경한다

Sync\_Screen \_3-1: 발표자의 화면을 공유 받는다.

Sync\_Screen \_3-2: 발표자료의 연결 해제를 원할 시 발표자료의 접속을 해제한다.

Sync\_Screen \_4: 발표자료의 화면공유를 재 연결을 원할 시 처음(접속)부터 시작한다.