

프라이버시 보호 실시간 지원 서비스

시퀀스 다이어그램

12조

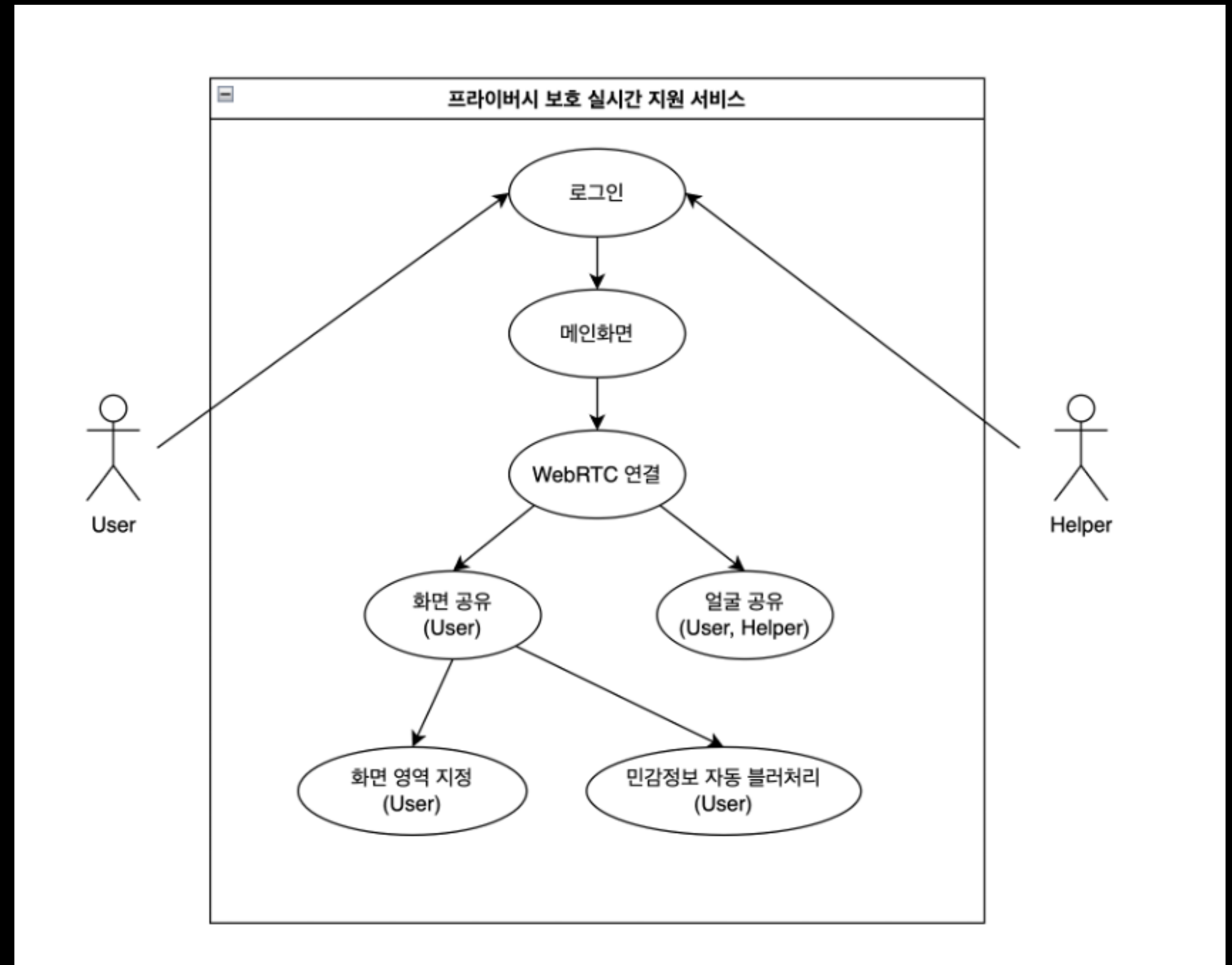
202202624 이예인

202002569 최동현

유스케이스 다이어그램

다이어그램 순서

1. 로그인
2. 메인 화면 접속
3. 방 입장 및 WebRTC 연결
4. 화면 영역 지정 기능
5. 민감 정보 자동 블러처리 기능

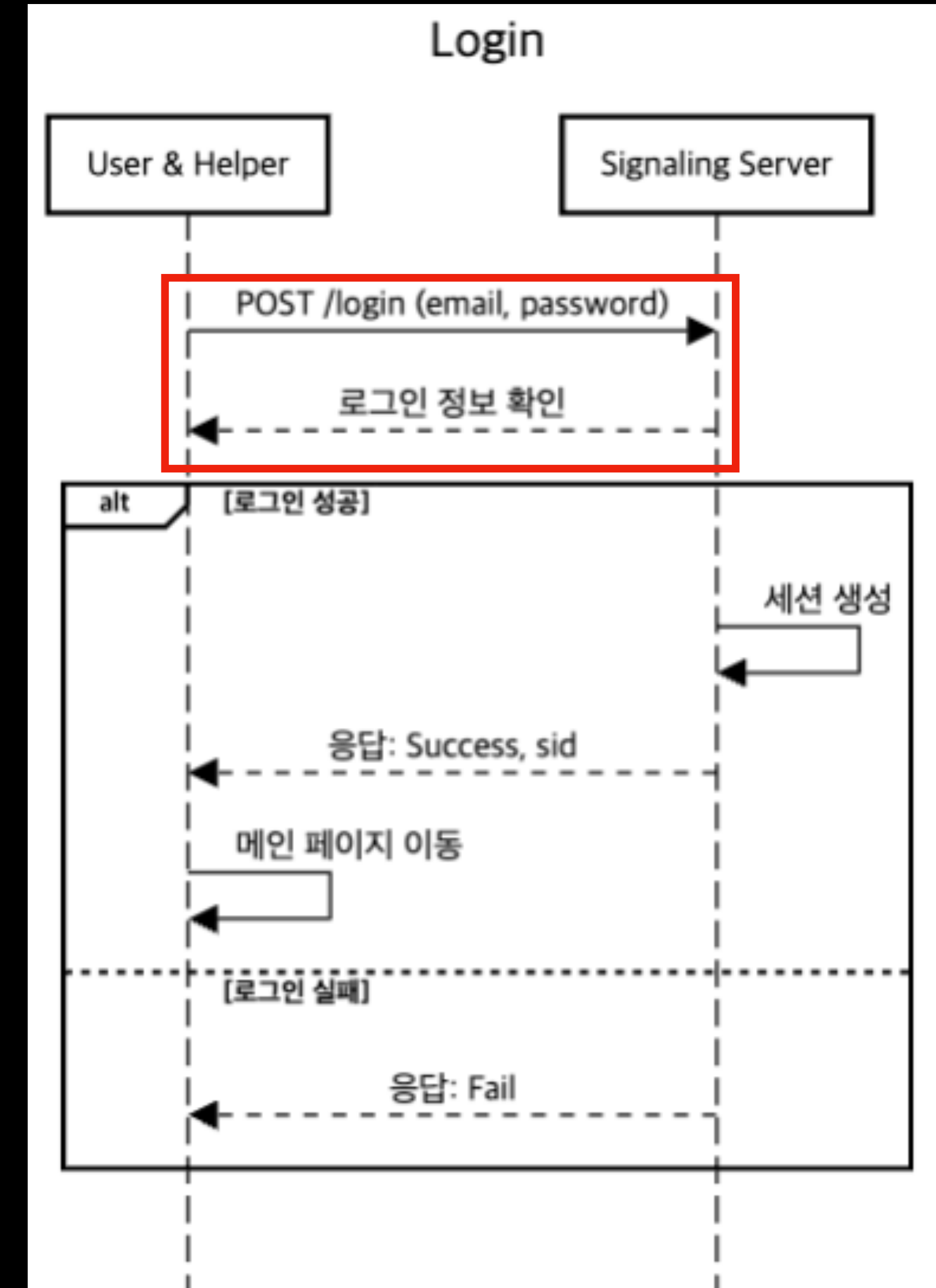


1. 로그인

로그인 정보 송수신

1. 유저가 시그널링 서버에 로그인 정보 (email, password)를 보냄.

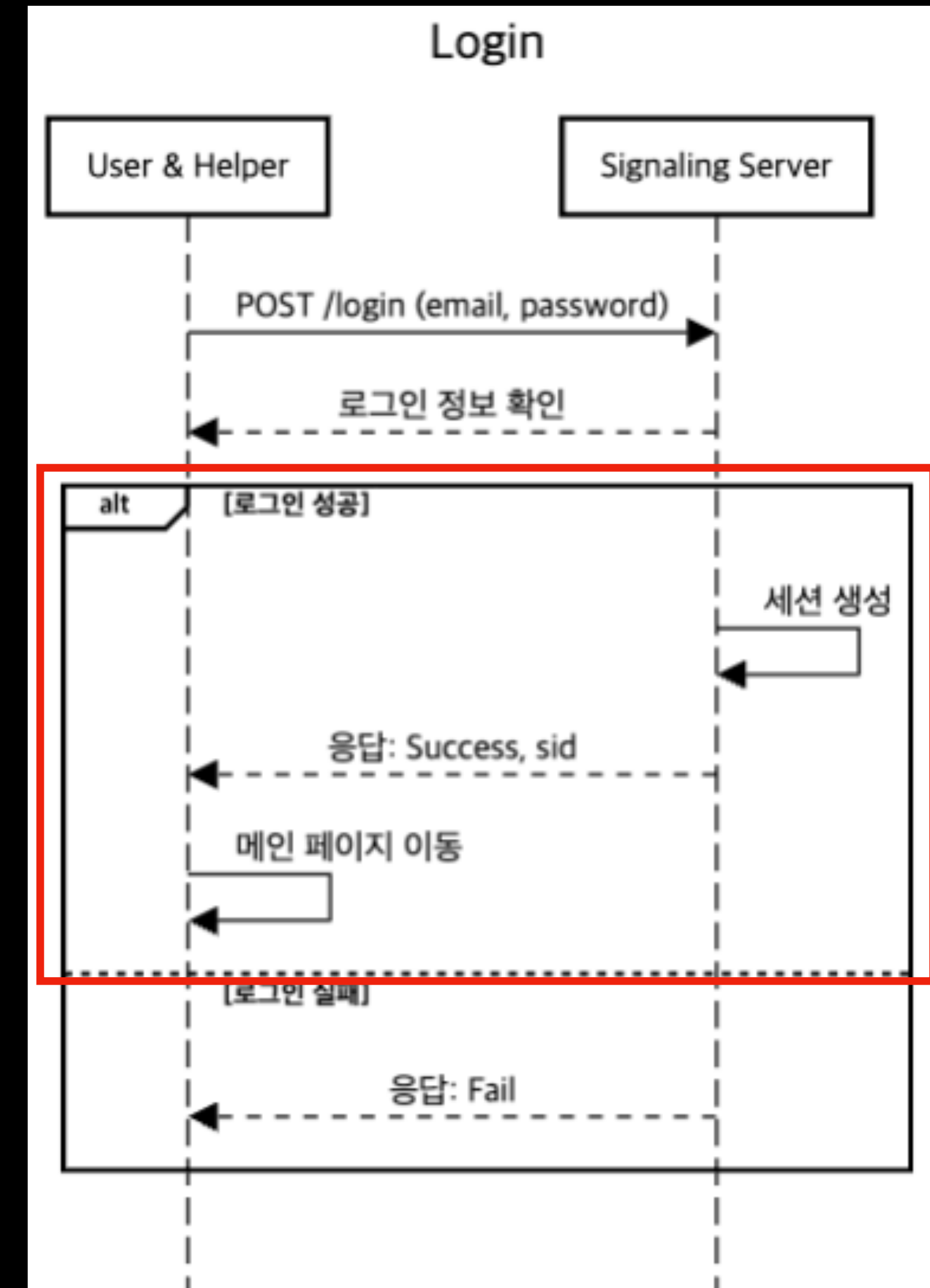
2. 시그널링 서버가 로그인 정보를 확인 하고 결과를 유저에게 전송



1. 로그인

로그인 성공

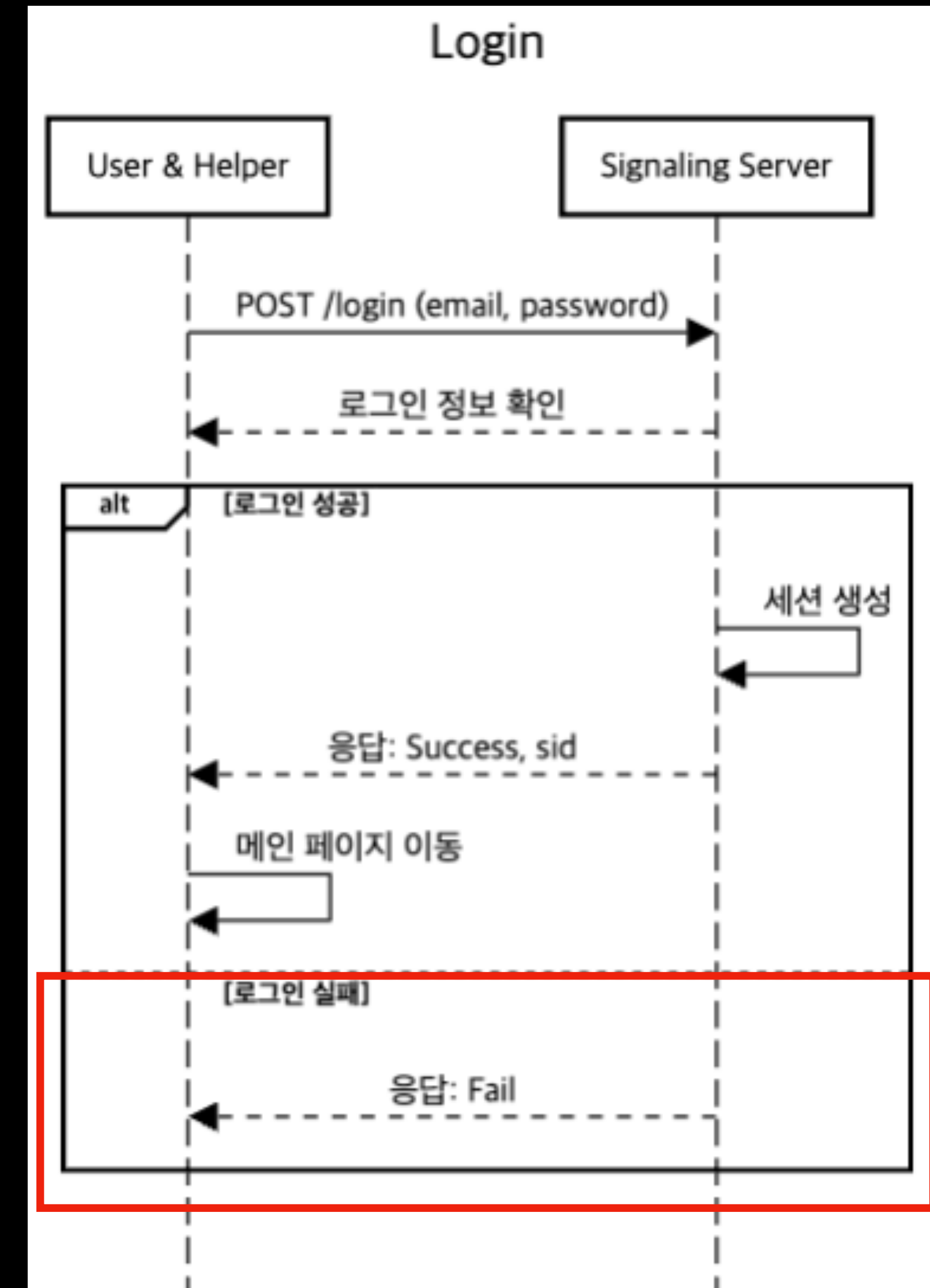
1. 시그널링 서버는 세션을 생성
2. 응답(Success, sid)을 유저에게 보냄
3. 사용자는 메인페이지로 이동됨



1. 로그인

로그인 실패

1. 로그인에 실패하면 Fail 응답을 보냄.

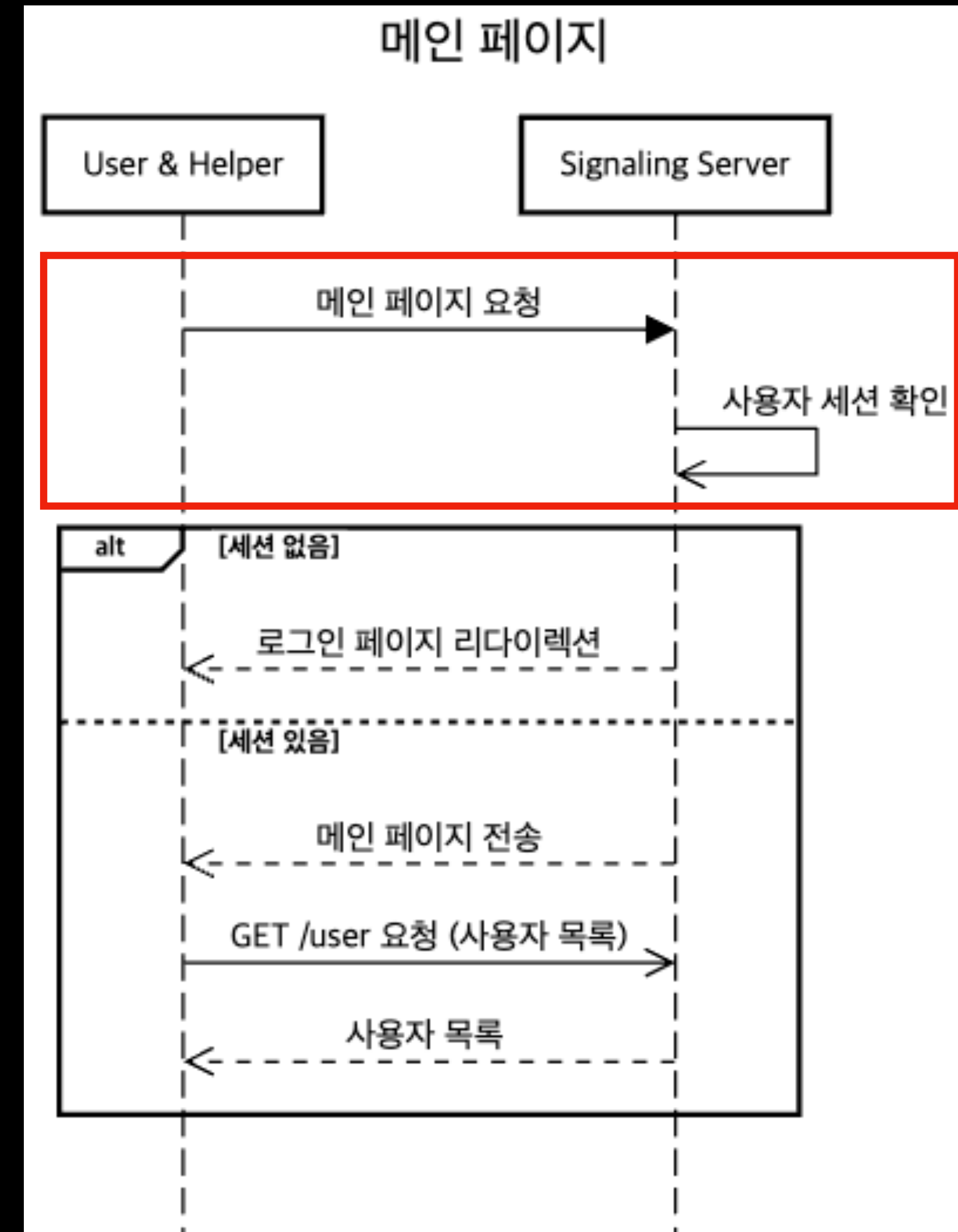


2. 메인 페이지

메인 페이지 요청

1. 유저는 메인 페이지를 시그널링 서버에 요청

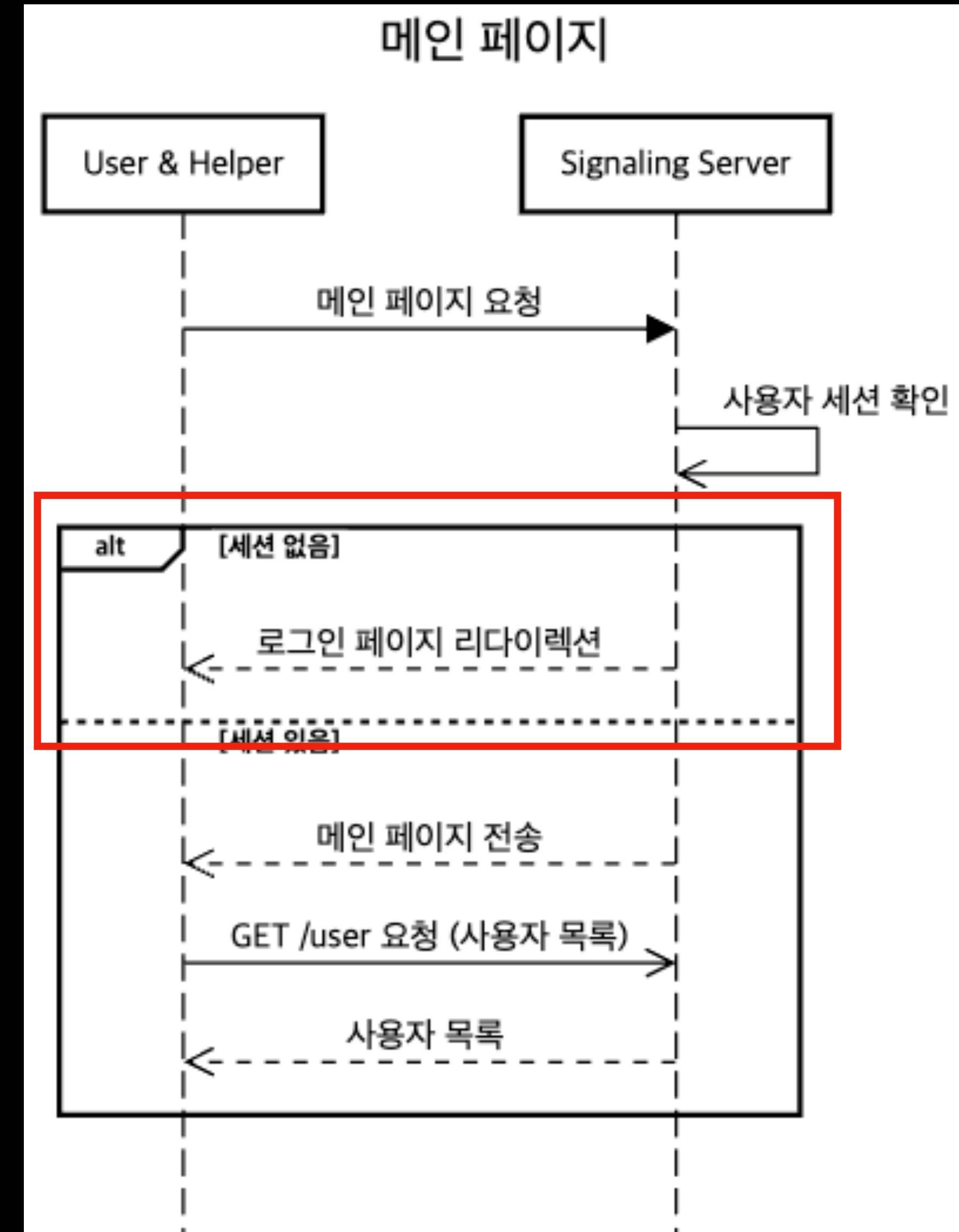
2. 시그널링 서버는 사용자 세션을 확인



2. 메인 페이지

세션 없음

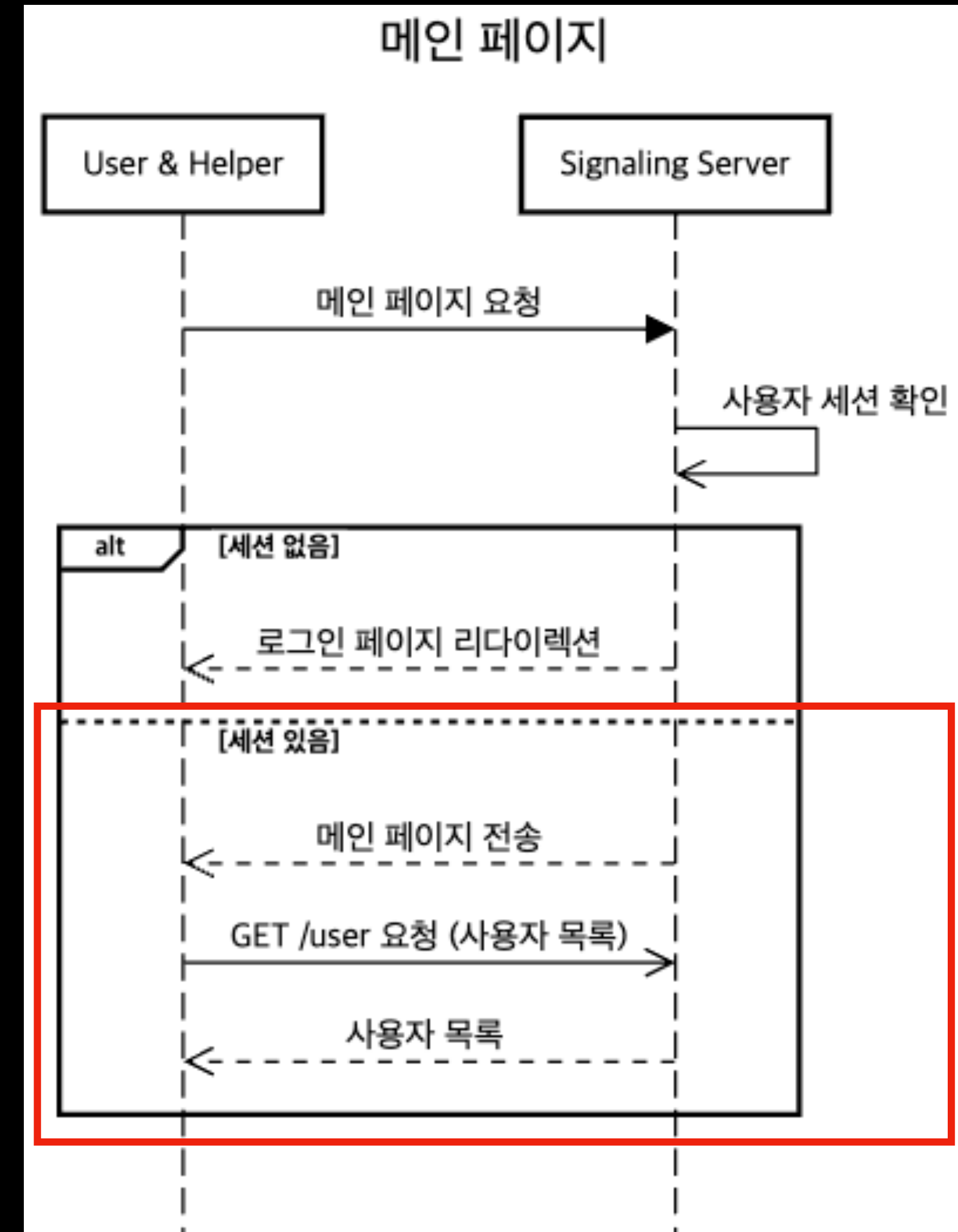
1. 유저는 로그인 페이지로 이동.



2. 메인 페이지

세션 있음.

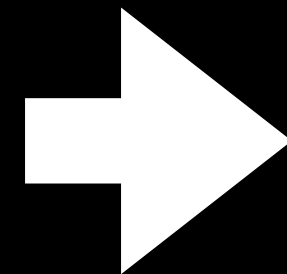
1. 시그널링 서버는 유저에게 메인 페이지를 전송
2. 유저는 시그널링 서버에 사용자 목록을 요청
3. 시그널링 서버는 세션 기반 사용자 목록을 조회하고 유저에게 전송



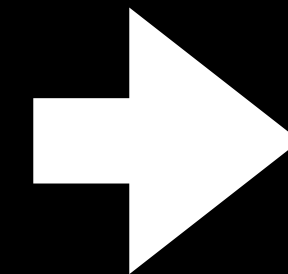
3. WebRTC 연결

과정 소개

Helper 연결 준비



User 연결 준비



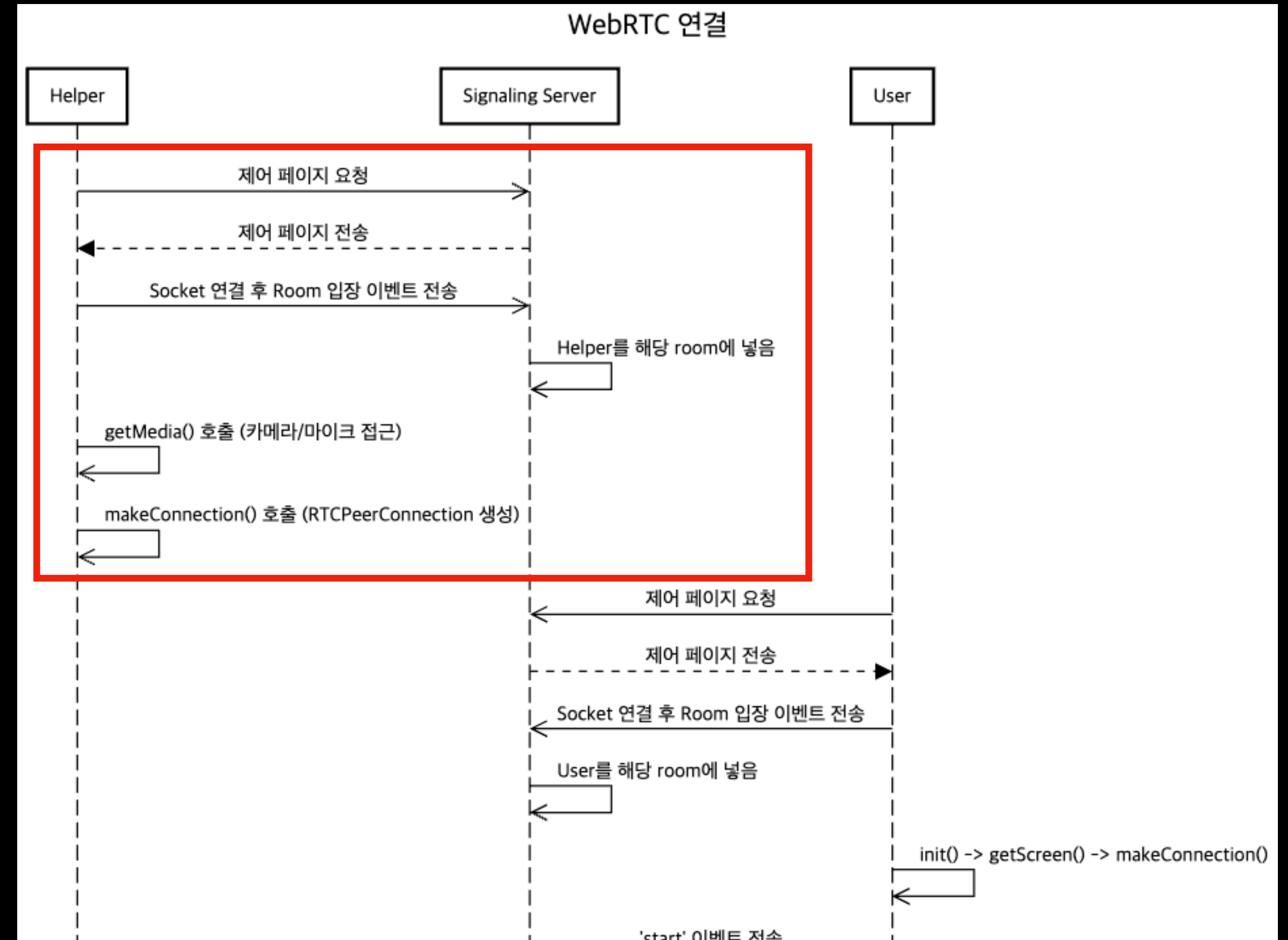
1. 연결 정보 교환
2. 전송 시작

WebRTC 연결 시작

3. WebRTC 연결

Helper의 연결 준비

1. Helper는 "도움주기" 버튼 클릭 → 제어 페이지로 이동
2. Helper는 Room 입장 이벤트를 Signaling Server에 보냄
3. Signaling Server는 Helper를 해당 room에 넣음
4. Helper는 `getMedia()` 호출하여 카메라/마이크 접근 권한 요청
5. Helper는 `makeConnection()` 호출하고 → `RTCPeerConnection` 생성



3. WebRTC 연결

User의 연결 준비

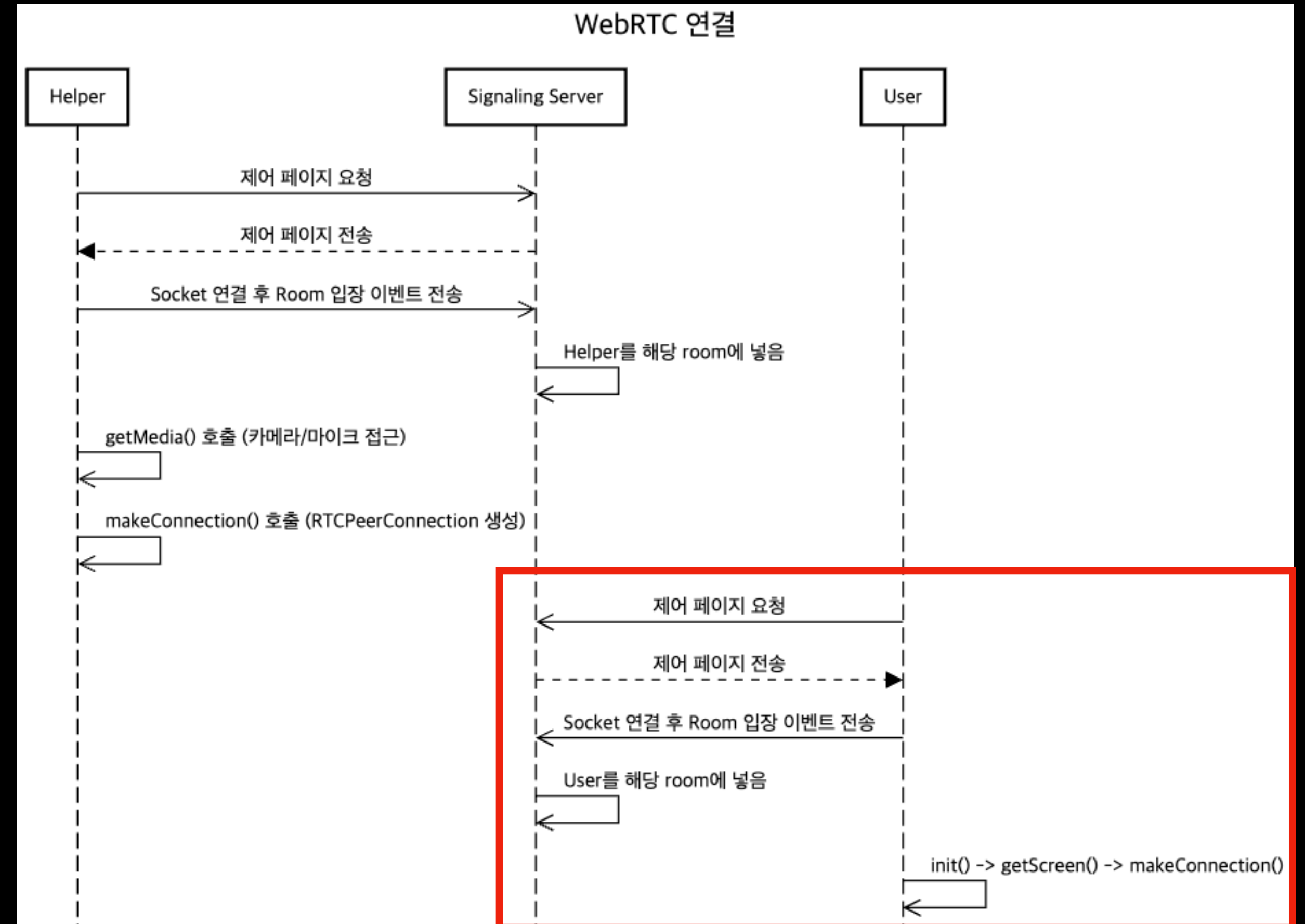
7. User는 "도움받기" 버튼 클릭 → 제어 페이지로 이동

8. User는 Socket 서버에 연결함

9. User는 Room 입장 이벤트를 Signaling Server에 보냄

10. Signaling Server는 User를 해당 room에 넣음

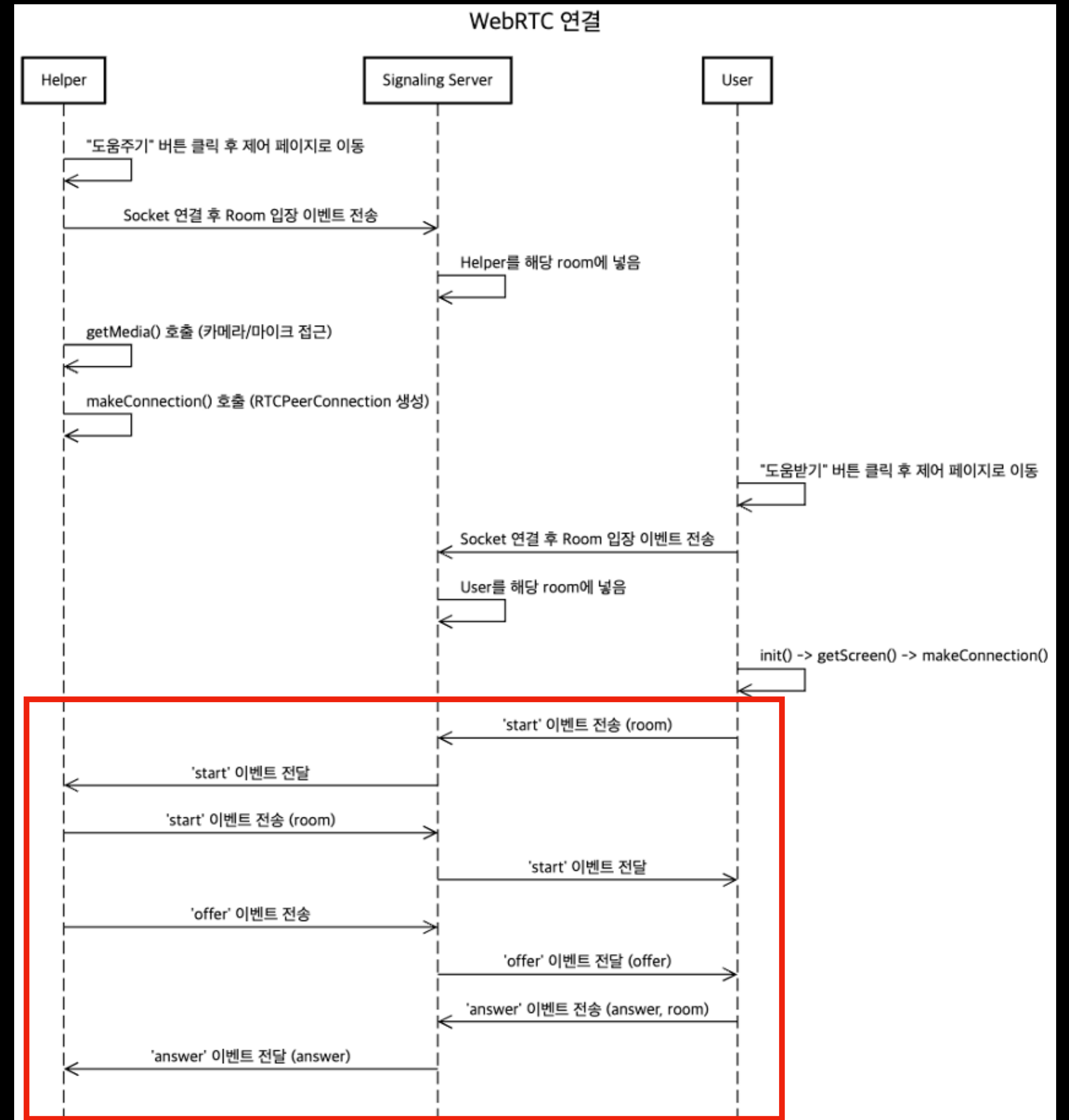
11. User는 `init()` → `getScreen()` → `makeConnection()` 호출로 화면 공유 준비 및 PeerConnection 생성



3. WebRTC 연결

연결 시작 및 시그널링 교환

1. User는 'start' 이벤트를 Signaling Server에 전송 (room 포함)
2. Signaling Server는 'start' 이벤트를 Helper에게 전달
3. Helper는 'offer' 생성 후 Signaling Server에 전송
4. Signaling Server는 'offer'를 User에게 전달함
5. User는 'answer' 생성 후 Signaling Server에 전송
6. Signaling Server는 'answer'를 Helper에게 전달함

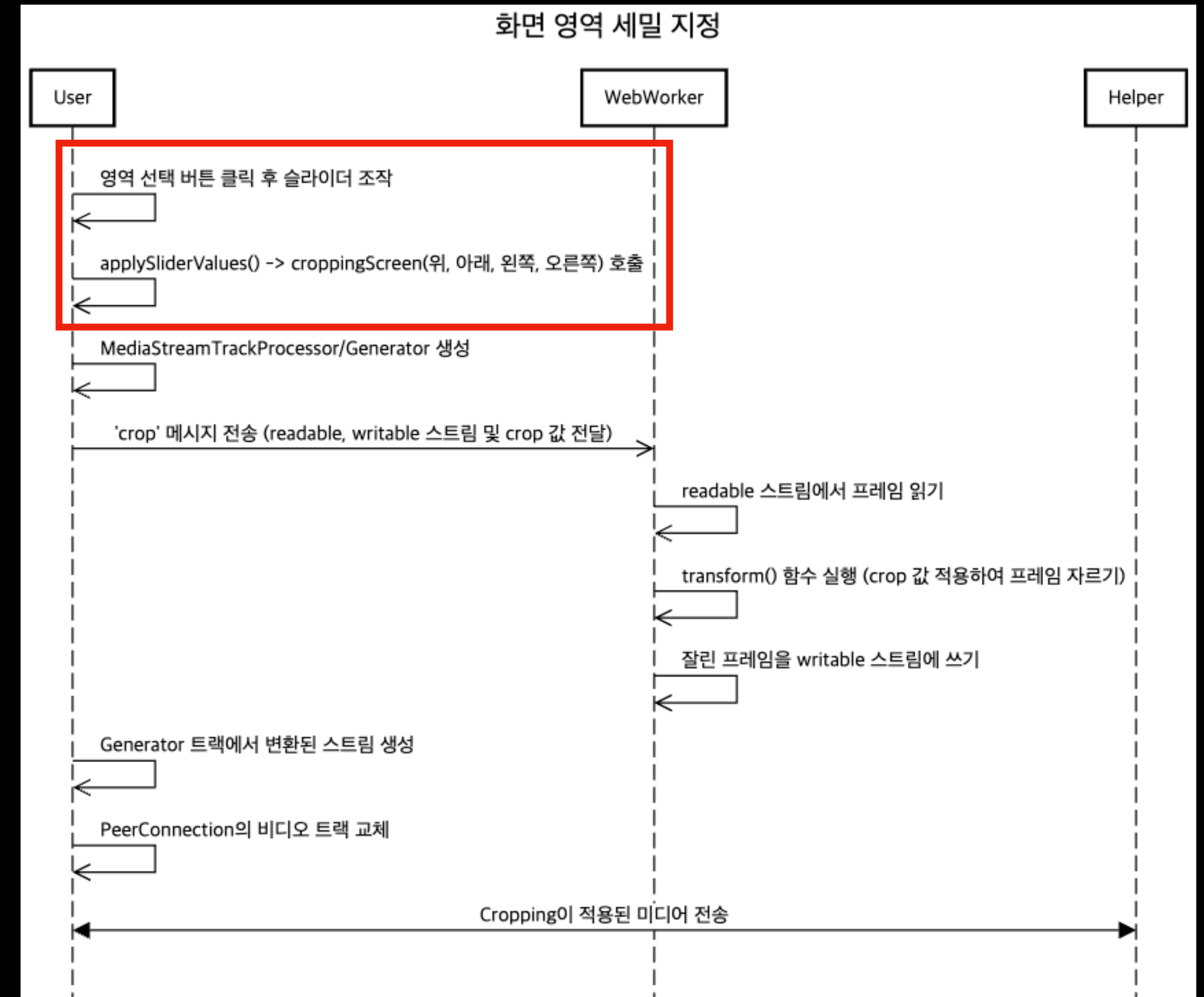


4. 화면 영역 정밀 지정

사용자 측 조작

1. User가 "영역 선택" 버튼 클릭 후 슬라이더로 상하좌우 조절

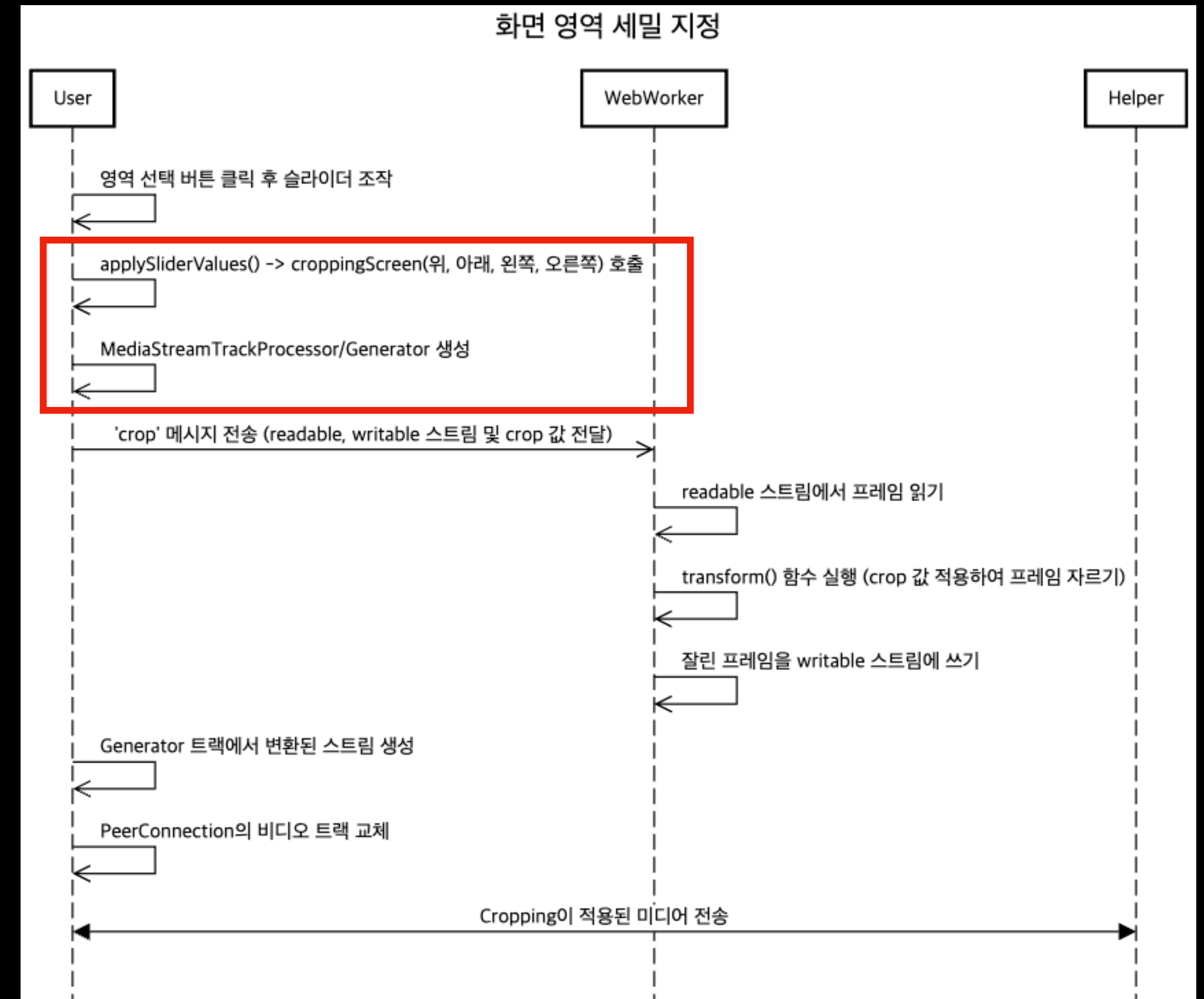
2. applySliderValues() → croppingScreen() 호출 (위, 아래, 왼쪽, 오른쪽 crop 값 전달)



4. 화면 영역 정밀 지정 스트림 처리 초기화

1. MediaStreamTrackProcessor 및
MediaStreamTrackGenerator 생성됨

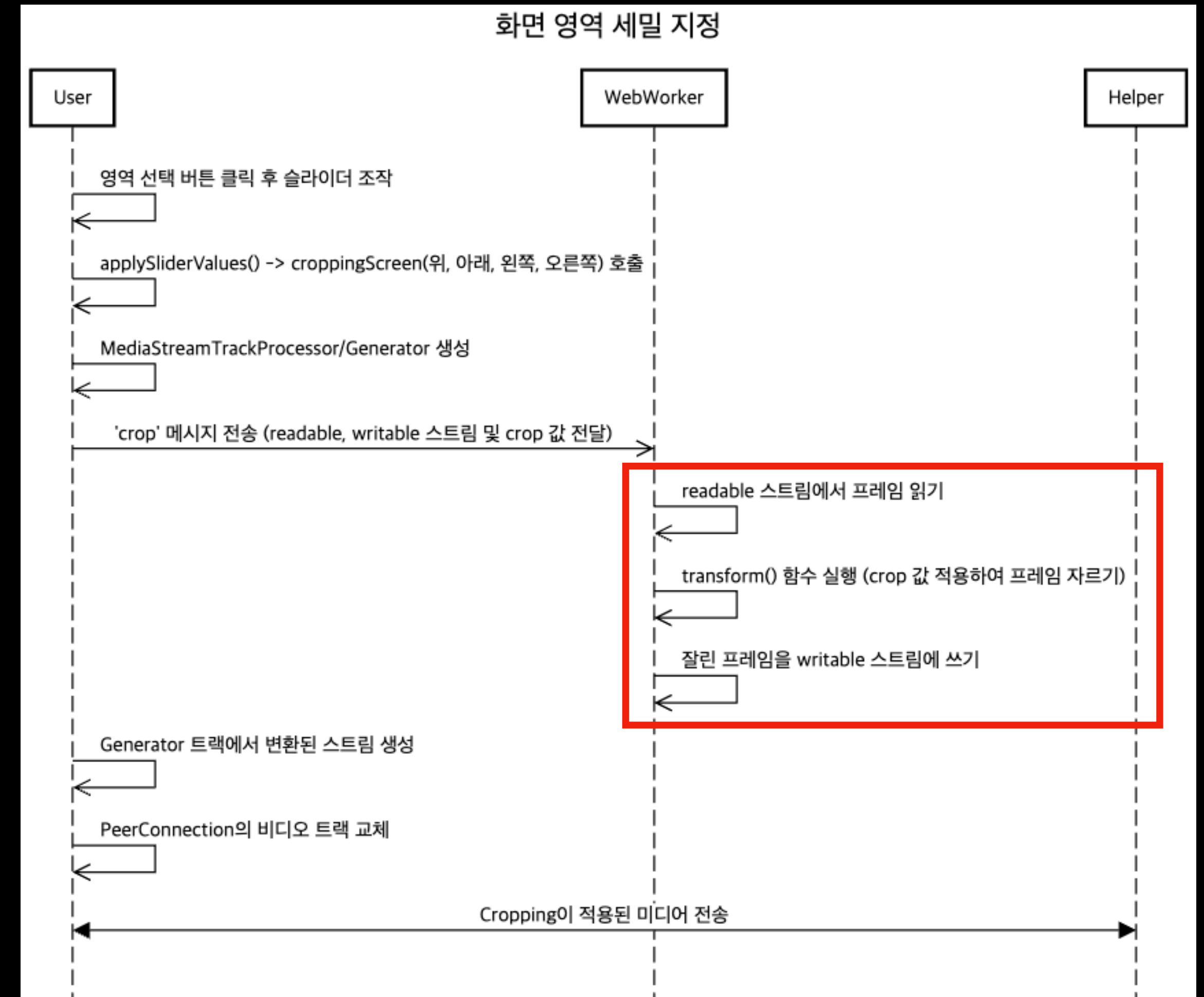
2. crop 메시지를 WebWorker로 전송
(readable, writable stream 객체 및
crop 값 전달)



4. 화면 영역 정밀 지정

WebWorker에서 영상 프레임 처리

1. WebWorker는 readable 스트림에서 프레임을 읽음
2. transform() 함수 실행 → crop 값 기준으로 프레임 자름
3. 자른 프레임을 writable 스트림에 씀

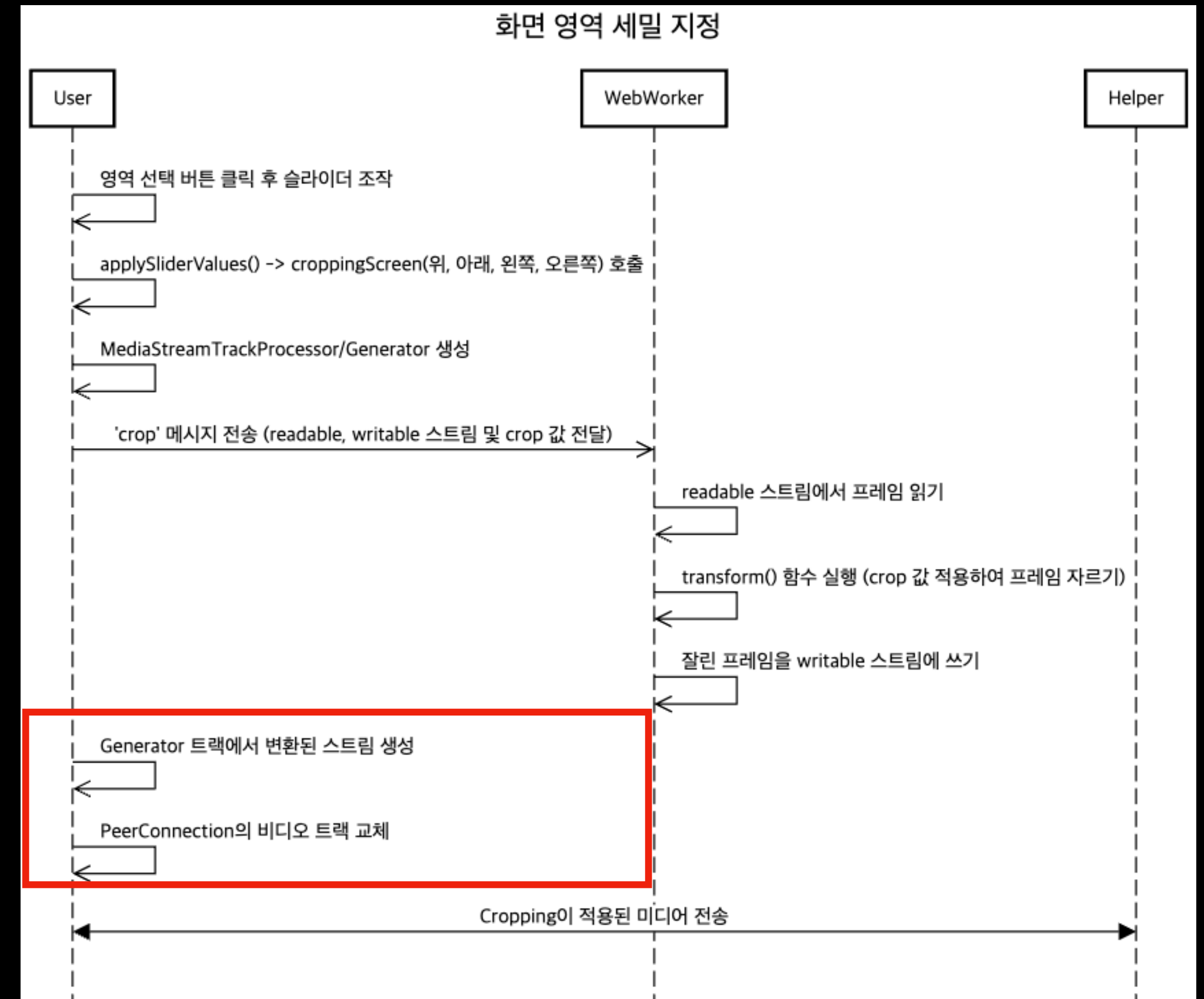


4. 화면 영역 정밀 지정

User 측 전송 준비

1. Generator에서 변환된 스트림 생성

2. 기존 PeerConnection의 비디오 트랙을 새로 변환된 트랙으로 교체

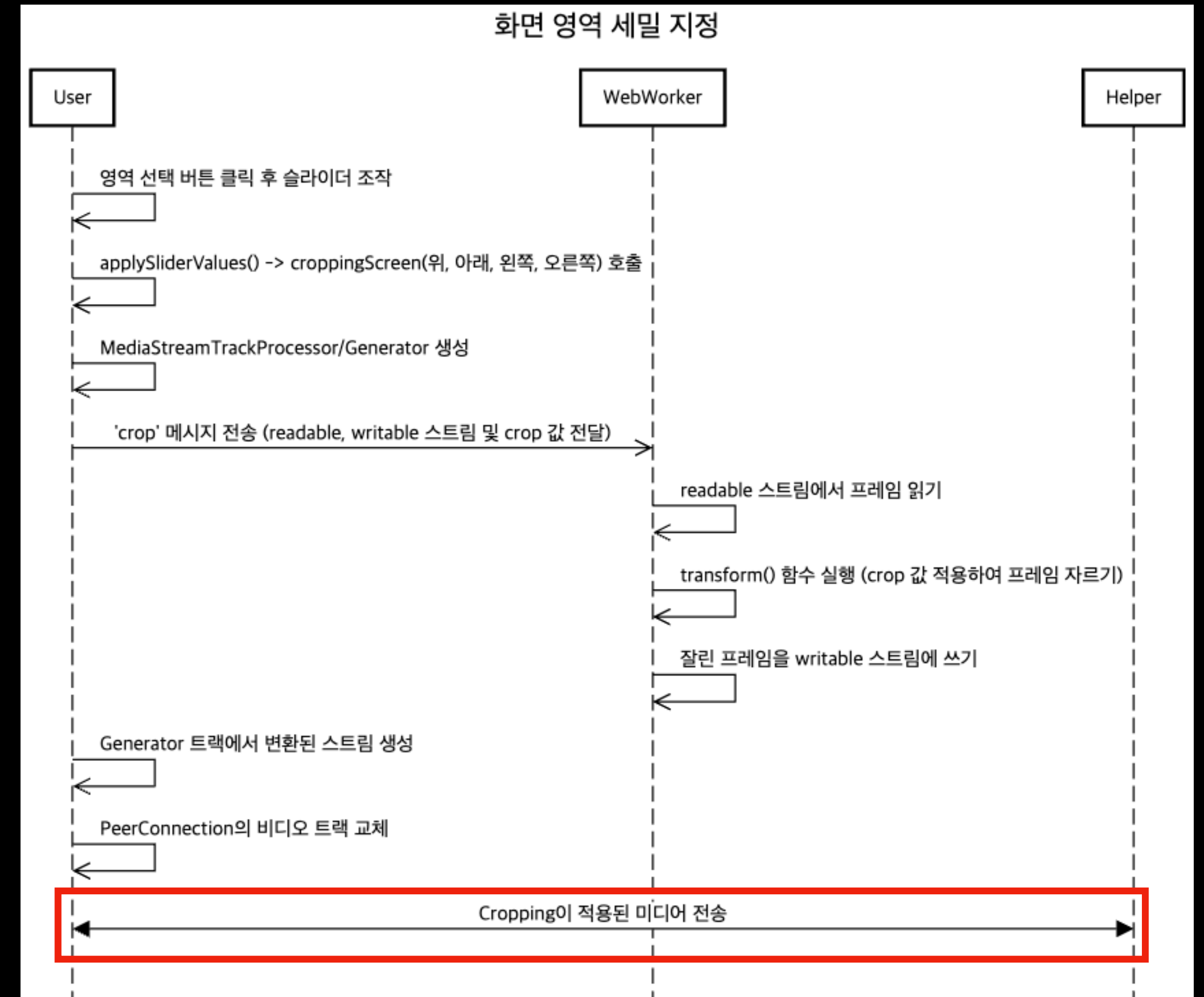


4. 화면 영역 정밀 지정

Helper 화면에 적용

1. PeerConnection의 스트림이 업데이트됨

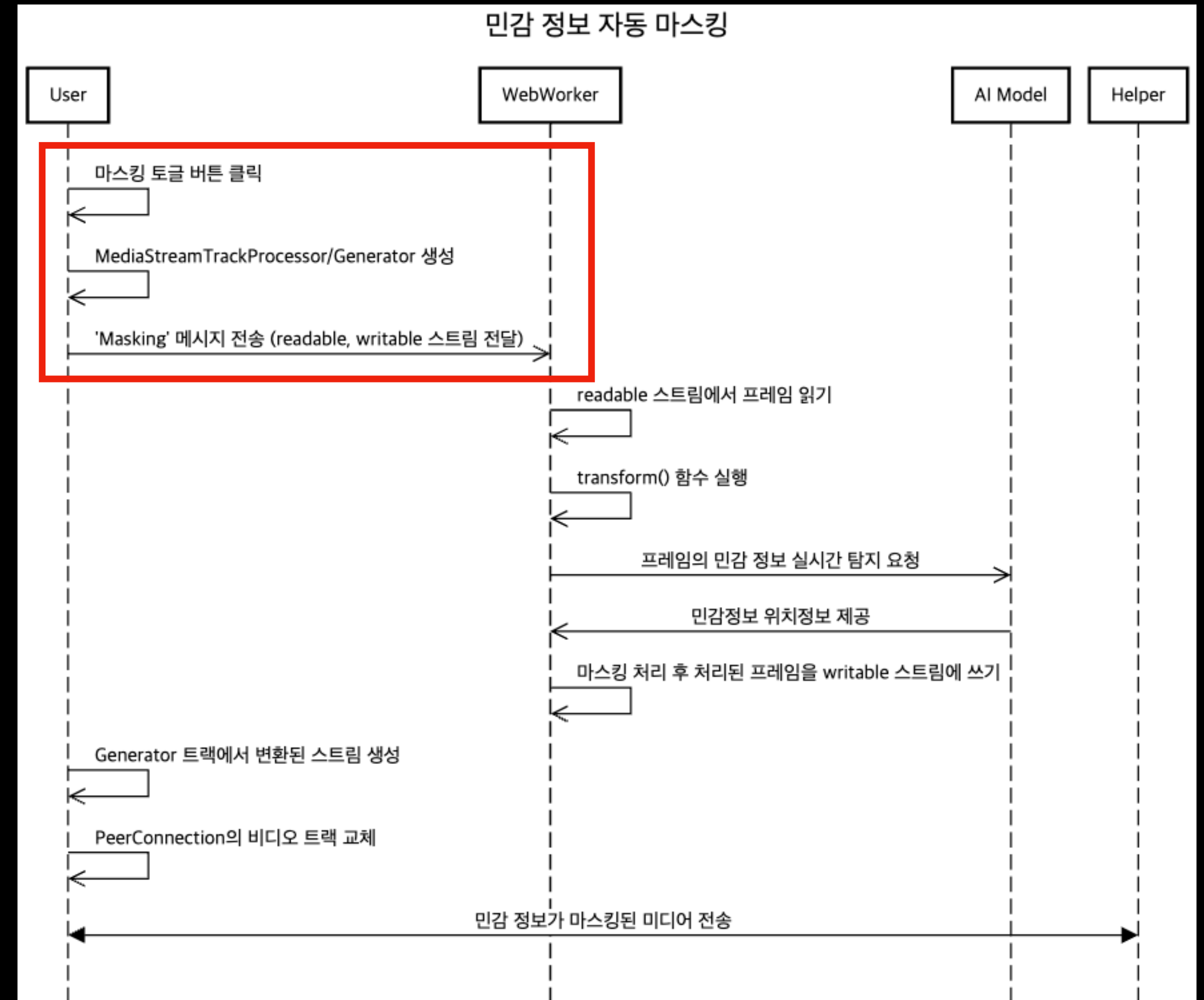
2. Helper는 User의 잘린 화면을 수신하고 표시



5. 민감 정보 자동 마스킹

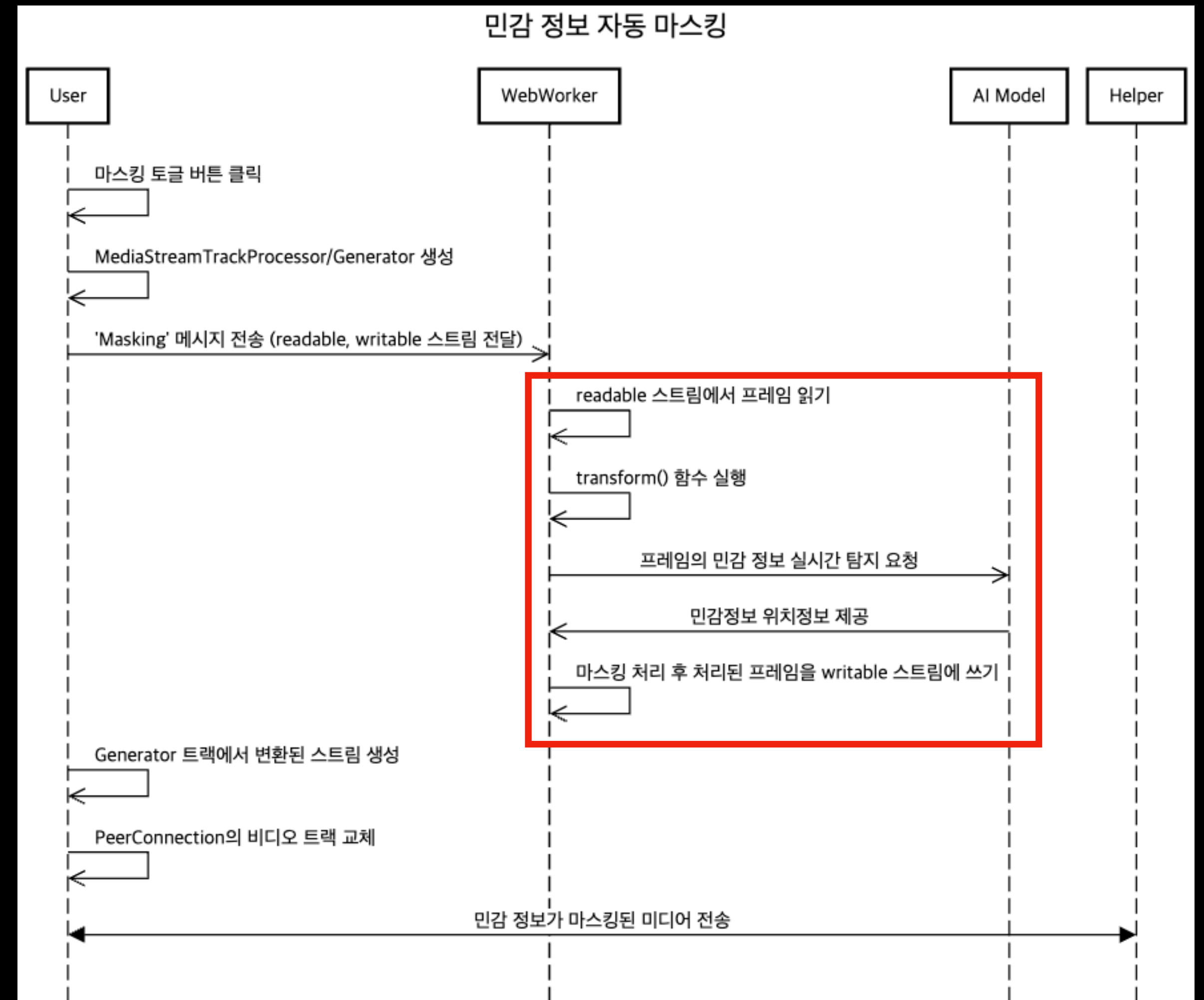
사용자 측 처리 시작

1. User가 “마스킹 토글” 버튼 클릭
2. MediaStreamTrackProcessor / Generator 생성
3. WebWorker에 'Masking' 메시지를 보냄 (readable, writable stream 객체 포함)



5. 민감 정보 자동 마스킹 WebWorker에서 프레임 처리

1. WebWorker는 readable 스트림에서 프레임 읽기
2. transform() 함수 실행 시작
3. 현재 프레임을 AI Model에 보내 민감 정보 탐지 요청
4. AI Model이 민감 정보 위치 좌표를 반환함
5. WebWorker는 해당 위치에 마스킹 처리 후, 처리된 프레임을 writable 스트림에 씀



5. 민감 정보 자동 마스킹 스트림 교체 및 표시

1. Generator에서 마스킹 처리된 스트림이 생성됨

2. 기존 PeerConnection의 비디오 트랙을 새로운 스트림으로 교체

3. Helper 화면에 마스킹된 User 화면이 실시간 표시됨

