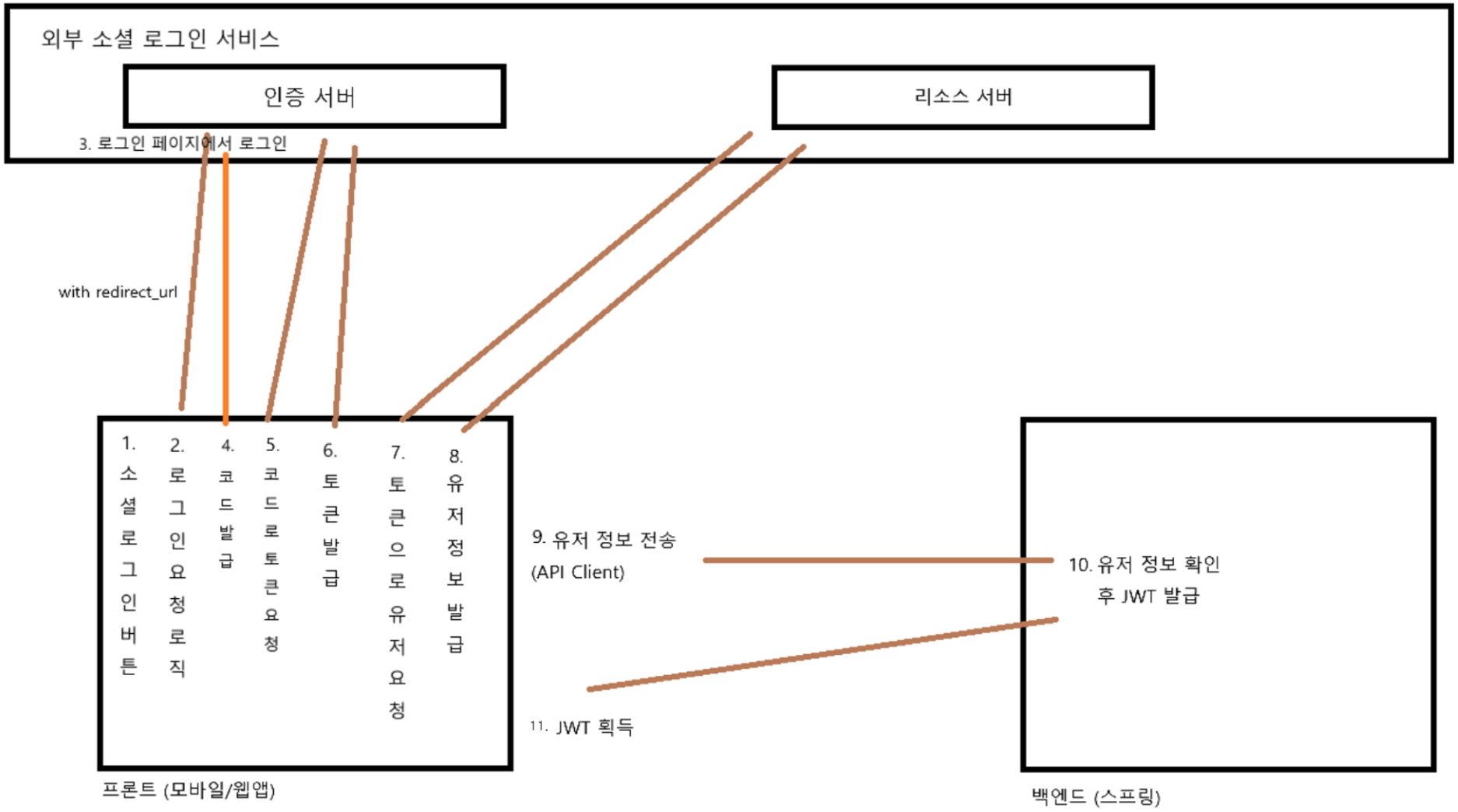
08

OAuth2 JWT

01. 기본 동작

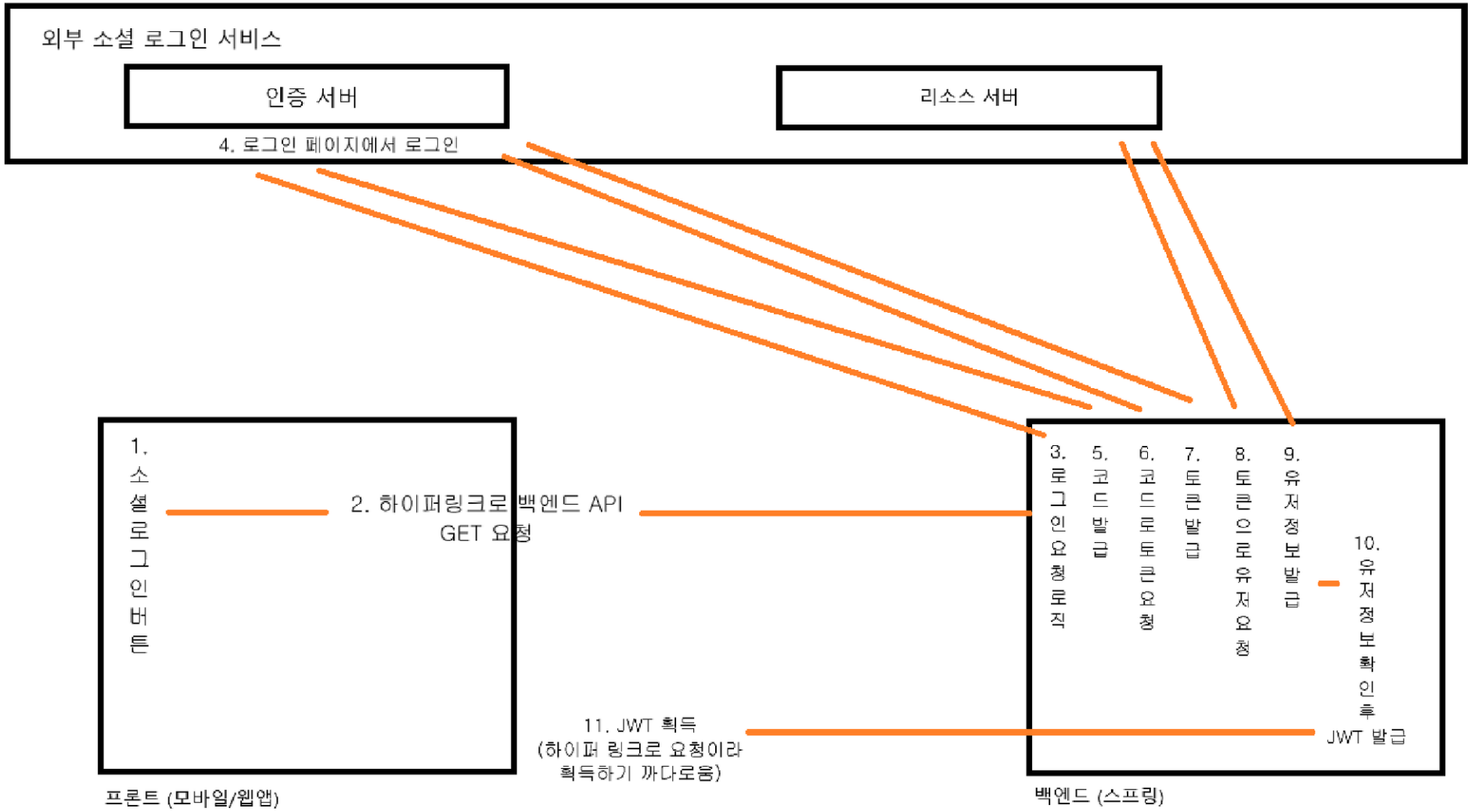
- 클라이언트 로그인 요청
- 소셜 인증 서버에 로그인 요청
- 소셜 인증 서버의 로그인 인증 완료 후 인가 코드 발급
- 인가 코드를 통해 다시 소셜 인증 서버에게 Access토큰 발급 요청
- 발급 받은 Access토큰을 통해 소셜 리소스 서버에 유저 정보 획득
- 획득한 유저정보로 JWT 클라이언트에게 발급

02. Front/Back 책임분배 (Frontend 중심방식)



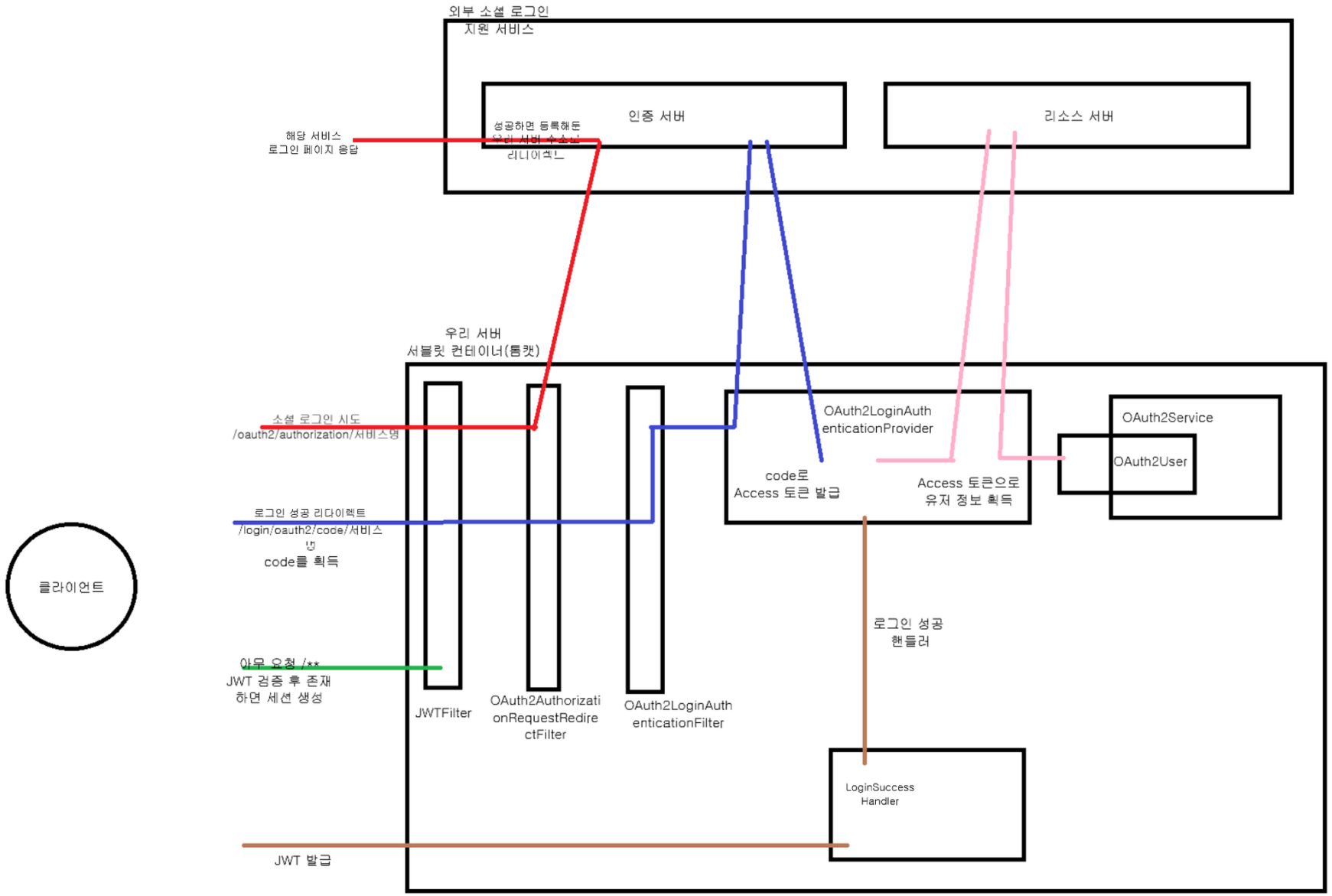
주로 네이티브 앱에서 사용하는 방식

02. Front/Back 책임분배 (BackEnd 중심방식)



웹앱/모바일앱 통합 환경 서버에서 사용하는 방식

03. 동작원리(브라우저 리다이렉트 방식)



04. 인증 흐름

1. 클라이언트 인증 요청(/oauth2/authorization/naver)
2. 애플리케이션 서버는 클라이언트를 소셜 인증 서버 로그인페이지로 리다이렉트
3. 소셜 인증 서버의 로그인 인증이 끝나면 클라이언트를 애플리케이션 서버가 미리 등록해둔 콜백 주소로 인가 코드와 함께 다시 리다이렉트
   (https://yourapp.com/callback?code=abcd1234&state=xyz)
4. 애플리케이션 서버는 인가 코드로 소셜 인증 서버에게 ACCESS 토큰을 발급받음
5. ACCESS 토큰으로 소셜 리소스 서버에게 유저정보를 획득
6. 획득한 유저정보를 이용하여 JWT 토큰을 클라이언트에게 발급

05. 추가 의존성

```
implementation 'io.jsonwebtoken:jjwt-api:0.12.3'
implementation 'io.jsonwebtoken:jjwt-impl:0.12.3'
implementation 'io.jsonwebtoken:jjwt-jackson:0.12.3'
implementation 'org.springframework.boot:spring-boot-
starter-oauth2-client'
```

06. 전역 변수 설정

```
#registration
spring.security.oauth2.client.registration.naver.client-name=naver
spring.security.oauth2.client.registration.naver.client-id=클라이언트 ID
spring.security.oauth2.client.registration.naver.client-secret=시크릿 키
spring.security.oauth2.client.registration.naver.redirect-uri=http://localhost:8080/login/oauth2/code/naver
spring.security.oauth2.client.registration.naver.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.naver.scope=name,email

#provider
spring.security.oauth2.client.provider.naver.authorization-uri=https://nid.naver.com/oauth2.0/authorize
spring.security.oauth2.client.provider.naver.token-uri=https://nid.naver.com/oauth2.0/token
spring.security.oauth2.client.provider.naver.user-info-uri=https://openapi.naver.com/v1/nid/me
spring.security.oauth2.client.provider.naver.user-name-attribute=response


#registration
spring.security.oauth2.client.registration.google.client-name=google
spring.security.oauth2.client.registration.google.client-id= 클라이언트 ID
spring.security.oauth2.client.registration.google.client-secret= 시크릿 키
spring.security.oauth2.client.registration.google.redirect-uri=http://localhost:8080/login/oauth2/code/google
spring.security.oauth2.client.registration.google.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.google.scope=profile,email
```