인증방식

7주차 장정훈

API Key

- 서비스들이 거대해짐에 따라 기능들을 분리하기 시작
- 이를 위해 Module이나 Application들 간의 공유와 독립성을 보장하기 위한 기능들이 등장
- 그 중 제일 먼저 등장하고 가장 널리 보편적으로 사용되는 기술이 API Key

API Key – 동작 방식

1. 사용자는 API Key를 발급받는다.
   - 발급 받는 과정은 서비스마다 다르다.
   - 공공기관 API같은 경우에는 신청에 필요한 양식을 제출
   - 관리자가 확인 후 Key를 발급
2. 해당 API를 사용하기 위해 Key와 함께 요청을 보낸다.
3. Application은 요청이 오면 Key를 통해 User정보를 확인하여 누구의 Key인지 권한이 무엇인지를 확인.
4. 해당 Key의 인증과 인가에 따라 데이터를 사용자에게 반환.

API Key – 문제점

- IPA Key를 사용자에게 직접 발급하고 해당 Key를 통해 통신을 하기 때문에
- 통신 구간이 암호화가 잘 되어 있더라도 Key가 유출된 경우에 대비하기 힘들다.
- 주기적인 Key 업데이트를 해야하기 때문에 번거로워짐
- 예기치 못한 상황(한쪽만 업데이트가 되어 매치가 안 될 때 등)이 발생할 수 있다.
- Key 하나로 정보를 제어하기 때문에 보안문제가 발생하기 쉬운 편이다.

OAuth2

- API Key의 단점을 메꾸기 위해 등장한 방식
- Third-Party 프로그램에게 리소스 소유자를 대신하여 리소스 서버에서 제공하는 자원에 대한 접근 권한을 위임하는 방식
- 대표적으로 페이스북, 트위터 등 SNS 로그인 기능에서 쉽게 볼 수 있다.
- 요청하고 요청 받는 단순한 방식이 아니라 인증하는 부분이 추가되어 독립적으로 세분화가 이루어졌다.

OAuth2 – 권한 부여 방식

1. Authorization Code Grant : 권한 부여 승인 코드 방식
2. Implicit Grant : 암묵적 승인 방식
3. Resource Owner Password Credentials Grant : 자원 소유자 가격 증명 승인 방식
4. Client Credentials Grant

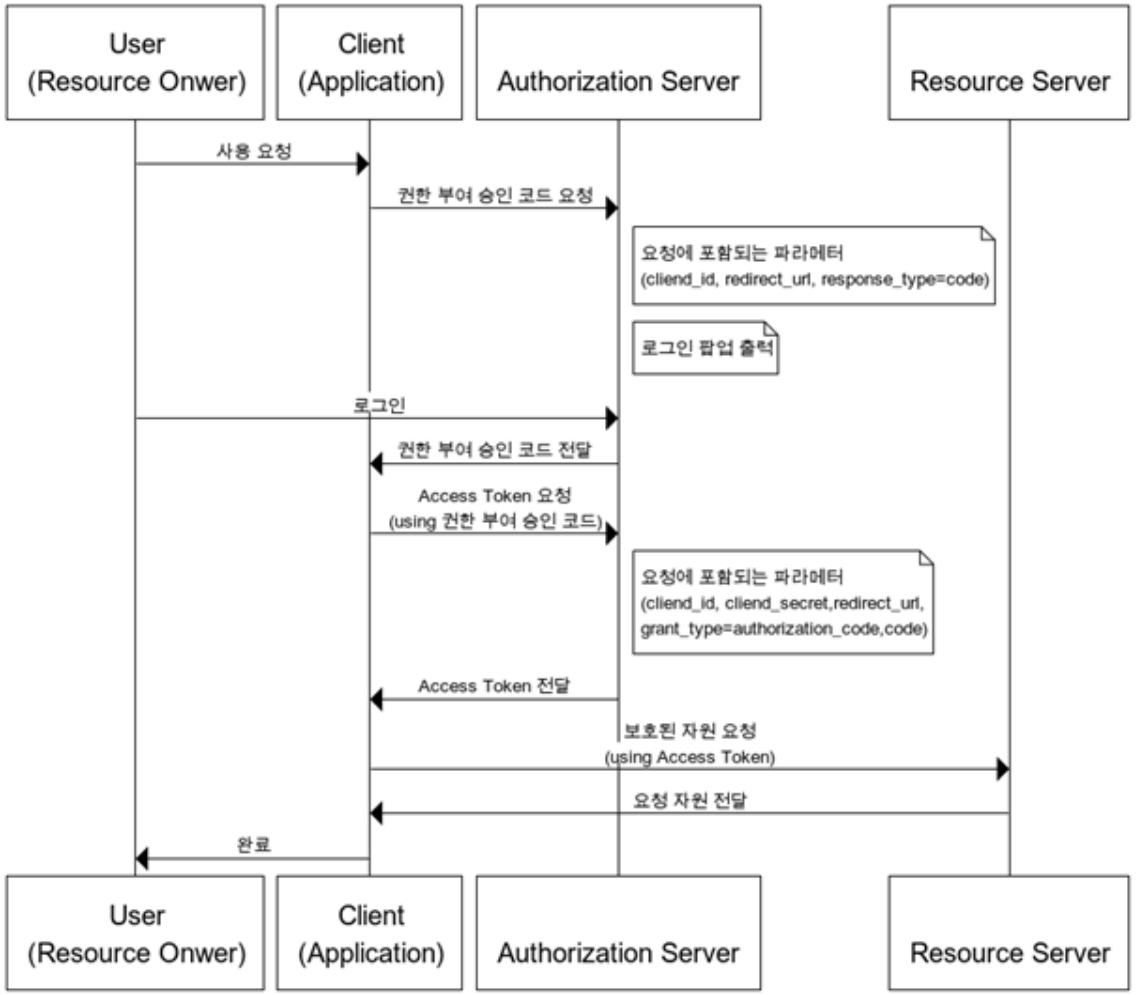
OAuth2 – Authorization Code Grant

- 권한 부여 승인을 위해 자체 생성한 Authorization Code를 전달하는 방식으로 많이 쓰이고 기본이 되는 방식

- 간편 로그인 기능에서 사용되는 방식으로 클라이언트가 사용자를 대신하여 특정 자원에 접근을 요청할 때 사용되는 방식

- 보통 타사의 클라이언트에게 보호된 자원을 제공하기 위한 인증에 사용.
- Refresh Token의 사용이 가능한 방식.

OAuth2 – Authorization Code Grant

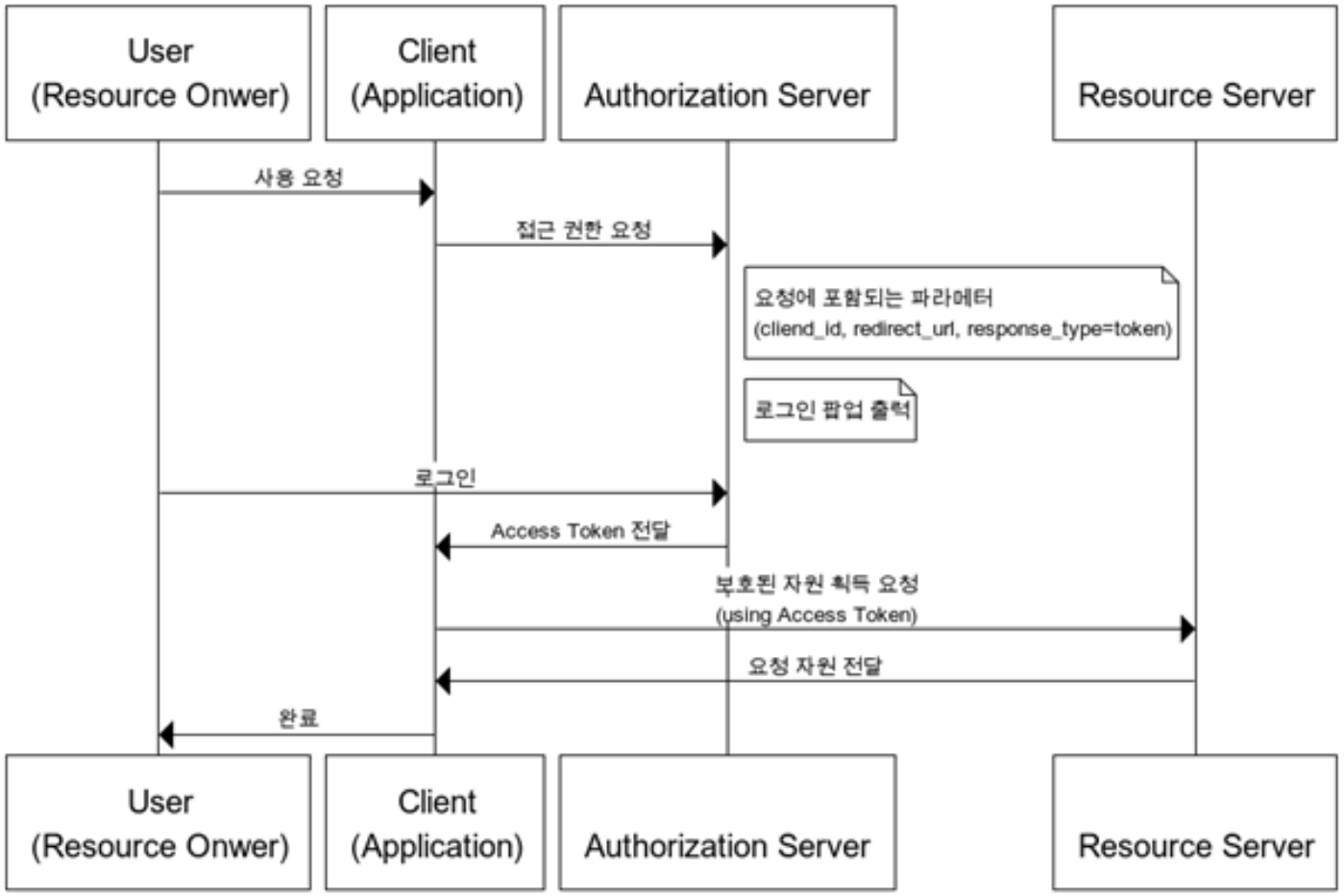


[그림 1] Authorization Grant: Authorization Code

OAuth2 – Implicit Grant

- 자격증명을 안전하게 저장하기 힘든 클라이언트에게 최적화된 방식.
- 암시적 승인 방식에서는 권한 부여 승인 코드 없이 바로 Access Token이 발급.
  - Access Token이 바로 전달되므로 만료 기간을 짧게 설정하여 누출의 위험을 줄일 필요가 있습니다.
- 응답성과 효율성은 높아지지만 Access Token이 URL로 전달된다는 단점.
- Refresh Token 사용이 불가능한 방식

OAuth2 – Implicit Grant

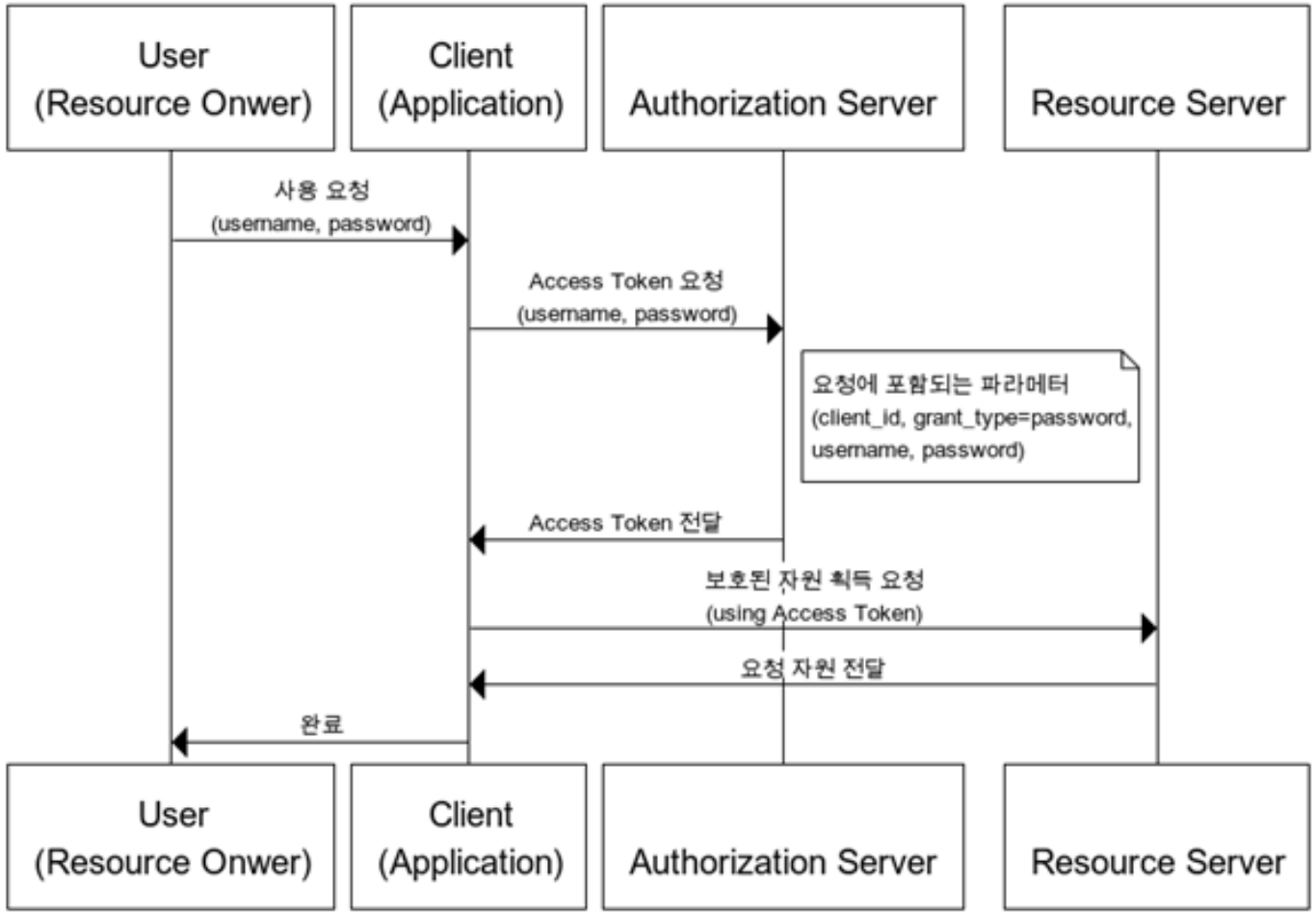


[그림2] Authorization Grant: Implicit

OAuth2 – Resource Owner Password Credentials Grant

- 간단하게 username, password로 Access Token을 받는 방식
- 클라이언트가 타사의 외부 프로그램일 경우에는 이 방식을 적용하면 안 됨.
- 자신의 서비스에서 제공하는 어플리케이션일 경우에만 사용되는 인증 방식.
- Refresh Token의 사용도 가능

OAuth2 – Resource Owner Password Credentials Grant

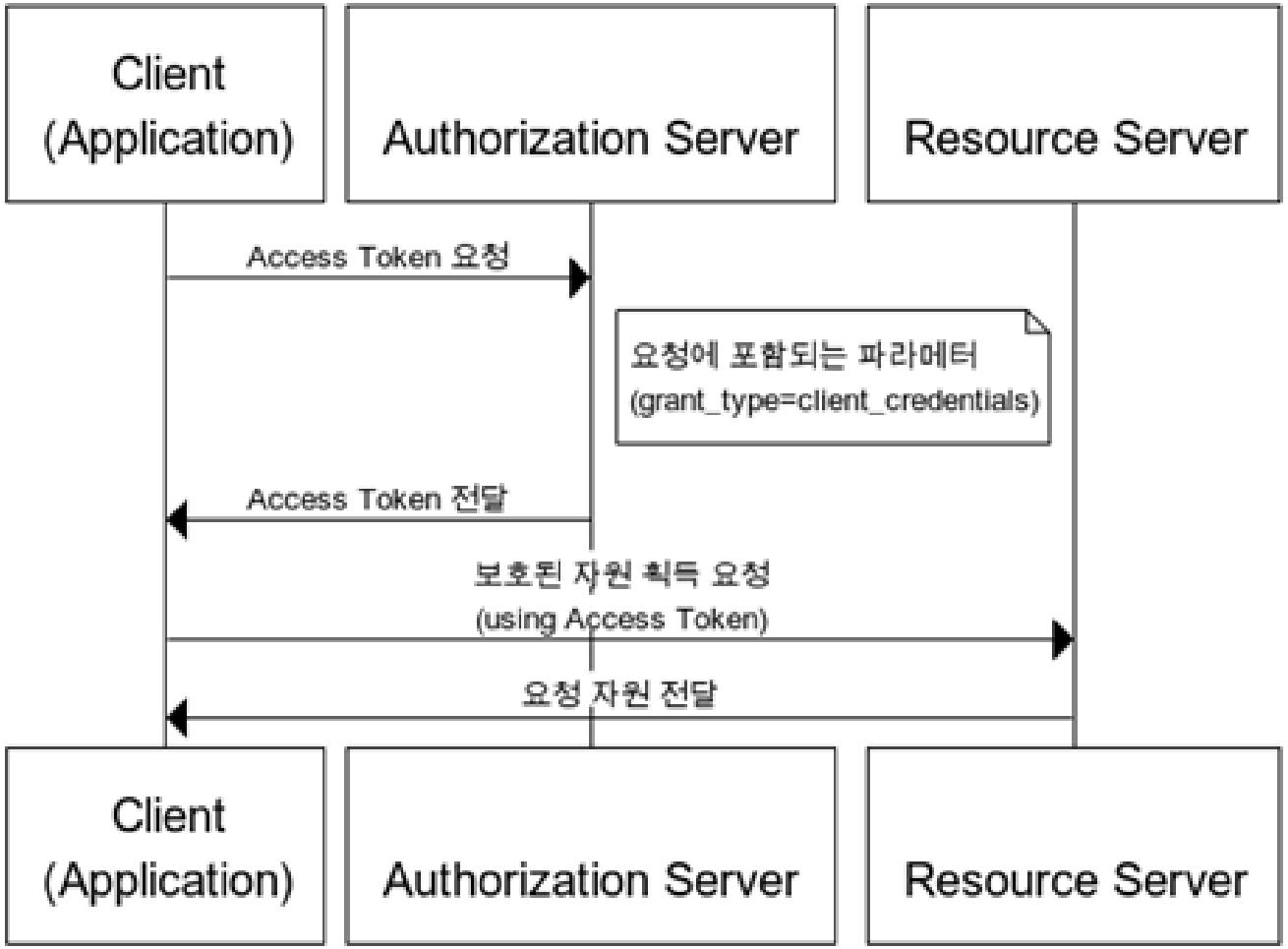


[그림 3] Authorization Grant: Password

OAuth2 – Client Credentials Grant

- 클라이언트의 자격증명만으로 Access Token을 획득하는 방식.
- OAuth2의 권한 부여 방식 중 가장 간단한 방식
- 클라이언트는 자신이 관리하는 리소스 혹은 권한 서버에 해당 클라이언트를 위한 제한된 리소스 접근 권한이 설정되어 있는 경우 사용.
- 이 방식은 자격증명을 안전하게 보관할 수 있는 클라이언트에서만 사용되어야 하고 Refresh Token은 사용할 수 없다.

OAuth2 – Client Credentials Grant



[그림 4] Authorization Grant: Client Credentials

OAuth2 – 문제점

- 단점은 기존 API Key방식에 비해 좀 더 복잡한 구조를 가짐.
- 통신에 사용하는 Token은 무의미한 문자열을 가지고 기본적으로 정해진 규칙없이 발행되기 때문에 증명 확인이 필요.
- 인증 서버에 어떤 식이든 DBMS 접근이든 다른 API를 활용하여 접근하는 등의 유효성 확인 작업이 필요하다는 공증 여부 문제.
- 유효기간 문제.

JWT

- JWT는 JSON Web Token의 줄임말로 인증 흐름의 규약이 아니라 Token 작성에 대한 규약

- 기본적인 Access Token은 의미가 없는 문자열로 이루어져 있어 Token의 진위나 유효성을 매번 확인해야 함.

- JWT는 인증 여부 확인을 위한 값, 유효성 검증을 위한 값 그리고 인증 정보 자체를 담고 있기 때문에 인증 서버에 묻지 않고도 사용할 수 있다.

JWT - 문제점

- 토큰 자체가 인증 정보를 가지고 있기 때문에 민감한 정보는 인증 서버에 다시 접속하는 과정이 필요하다.