Các công nghệ mới trong phát triển phần mềm

Bảo mật RESTful web service

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Phân biệt

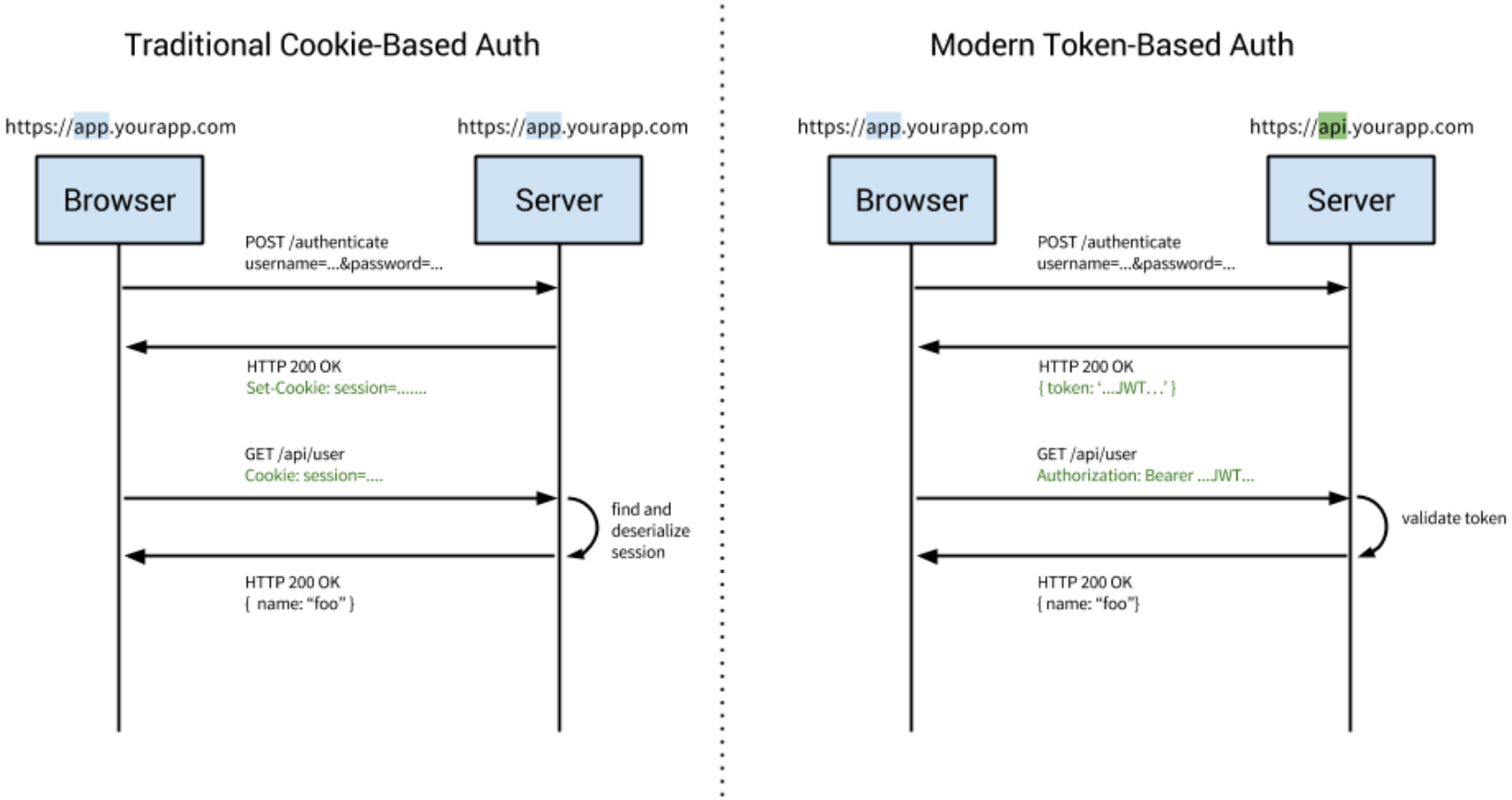
- Authentication
  - Bạn là ai?
- Authorization
  - Bạn được phép làm gì?

Basic Authentication

- `Authorization: Basic YWRtaW46MTIzNDU2`

- Cách dễ nhất để xác thực.
- Username và Password chỉ được *encode* bằng Base64 nên có thể dễ dàng đọc được.
- Password được gửi đi trong tất cả các request.
- Phải được bảo vệ trên TLS.

Cookie vs Token Authentication



https://auth0.com/blog/cookies-vs-tokens-definitive-guide/

Token based Authentication

- Token are stateless
- Token có thể được tạo từ nhiều server riêng biệt.
- Dễ phân quyền truy cập chi tiết

https://auth0.com/learn/token-based-authentication-made-easy/

HMAC

- HMAC: hash based message authentication

```
digest = base64encode(hmac("sha256", "secret", "GET+/users/johndoe/profile"));
```

HTTP Header:
GET /users/johndoe/profile HTTP/1.1
Host: example.org
Authentication: hmac johndoe:[digest]

- Server và client cần chia sẻ secret (không phải password) do server cần tính lại digest để so sánh

HMAC

- Cần bổ sung thêm timestamp và nonce.

```
digest = base64encode(hmac("sha256", "secret",
"GET+/users/johndoe/profile+1418746072+awWdY4"));
```

Định nghĩa JWT

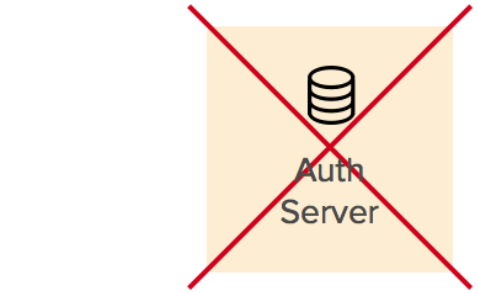
- JSON Web Token là chuẩn mở (RFC 7519) nhằm bảo vệ thông tin giữa các bên sử dụng đối tượng JSON để lưu thông tin mã hóa.

Cấu tạo

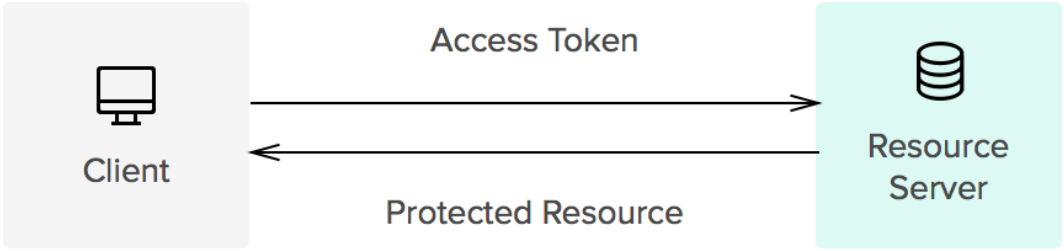
- Bao gồm 3 phần:
  - Header
  - Payload
  - Signature

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlIjoiSldU
IFJ1bGVzISIsImlhdCI6MTQ1OTQ0ODExOSwiZXhwIjoxNDU5NDU0NTE5f
Q.-yIVBD5b73C75osbmwwshQNRC7frWUYrqaTjTpza2y4
```

Access Token

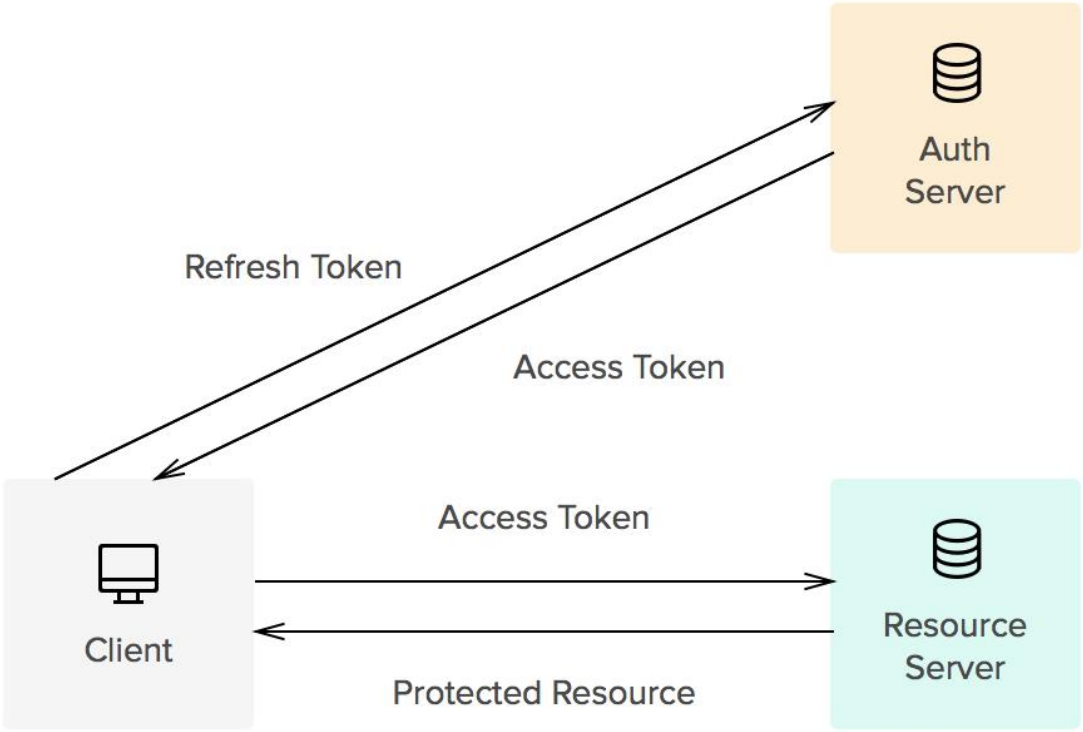


Access to the Auth Server is not necessary
to validate an access token



https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/

Refresh Token



https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/

Access Token vs Refresh Token

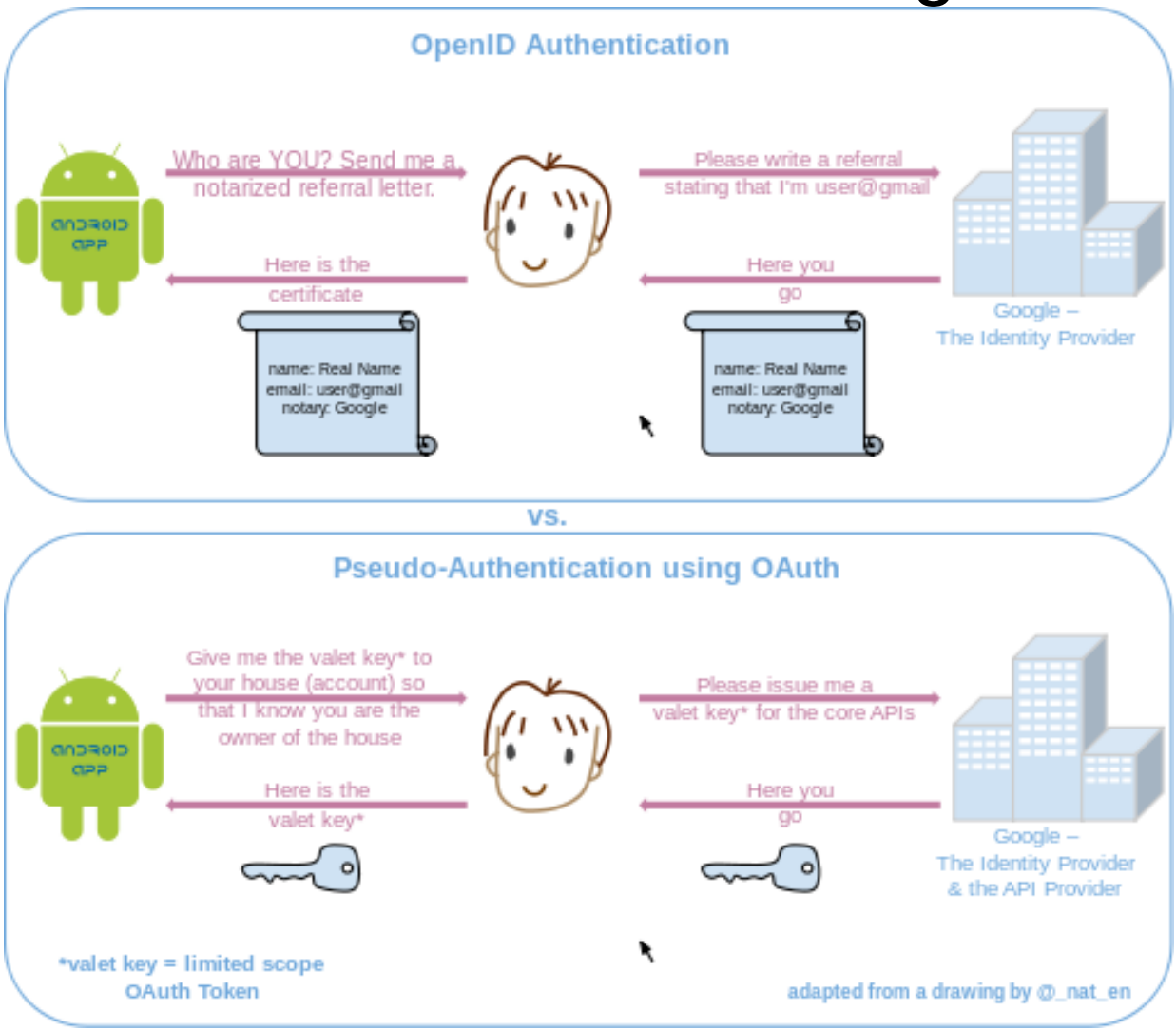
- Access Token
  - Giữ thông tin để truy cập tài nguyên
  - Thường có thời gian hết hạn ngắn
  - Không cần lưu trữ trên server
  - Không thể thu hồi lại
- Refresh Token
  - Giữ thông tin để tạo access token mới khi:
    - Cần truy xuất tài nguyên
    - Access token cũ hết hạn
  - Cũng có thể hết hạn nhưng thời gian hết hạn dài
  - Được lưu trữ trong cơ sở dữ liệu bảo mật nghiêm ngặt
  - Có thể thu hồi lại

OAuth 1.0a

- Là chuẩn mở để Authorization. http://tools.ietf.org/html/rfc5849
- Cho phép phát sinh các token tạm thời.
- Khó để xây dựng và chủ yếu dựa trên các thư viện có sẵn http://oauth.net/code/

IETF: Internet Engineering Task Force
RFC: Request for Comments

pseudo-authentication using OAuth

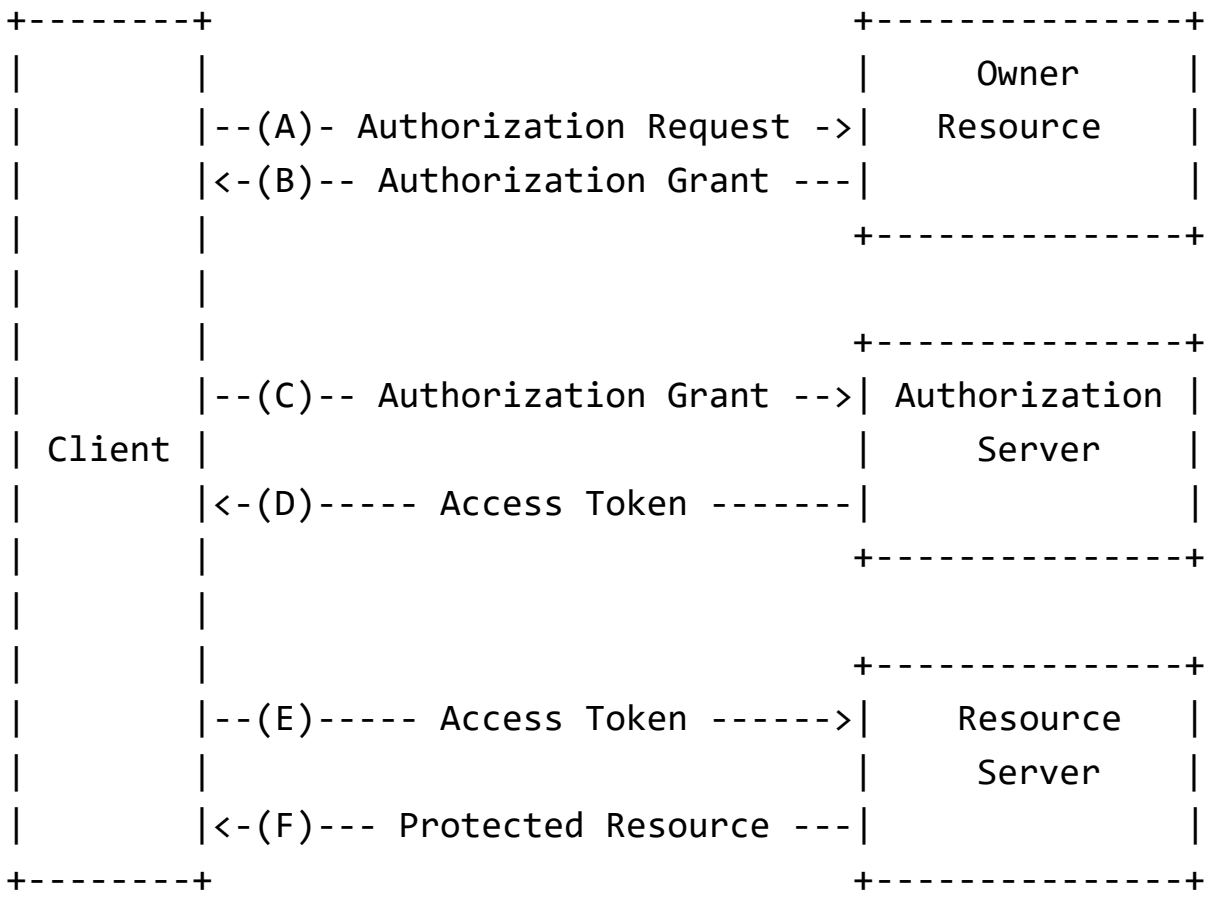


http://en.wikipedia.org/wiki/OAuth

OAuth 2.0

- Hoàn toàn không tương thích với OAuth 1.0
- Đơn giản hơn rất nhiều so với OAuth 1.0
- Thay thế chữ ký (HMAC, RSA, …) bằng TLS
- Mô tả rõ luồng cho:
  - Ứng dụng web
  - Ứng dụng trên desktop
  - Thiết bị di động
  - Thiết bị trong nhà

OAuth 2.0


Client
Owner
Resource
Authorization
Server
Resource
Server
(A)- Authorization Request ->
<-(B)-- Authorization Grant ---
--(C)-- Authorization Grant -->
<-(D)----- Access Token -------
--(E)----- Access Token ------>
<-(F)--- Protected Resource ---

Tham khảo thêm

- http://restcookbook.com/Basics/loggingin/
- http://oauthlib.readthedocs.org/en/latest/oauth2/oauth2.html
- http://www.asp.net/web-api/overview/security/individual-accounts-in-web-api
- https://auth0.com/learn/token-based-authentication-made-easy/
- https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/
- https://auth0.com/blog/cookies-vs-tokens-definitive-guide/
- https://auth0.com/blog/angularjs-authentication-with-cookies-vs-token/