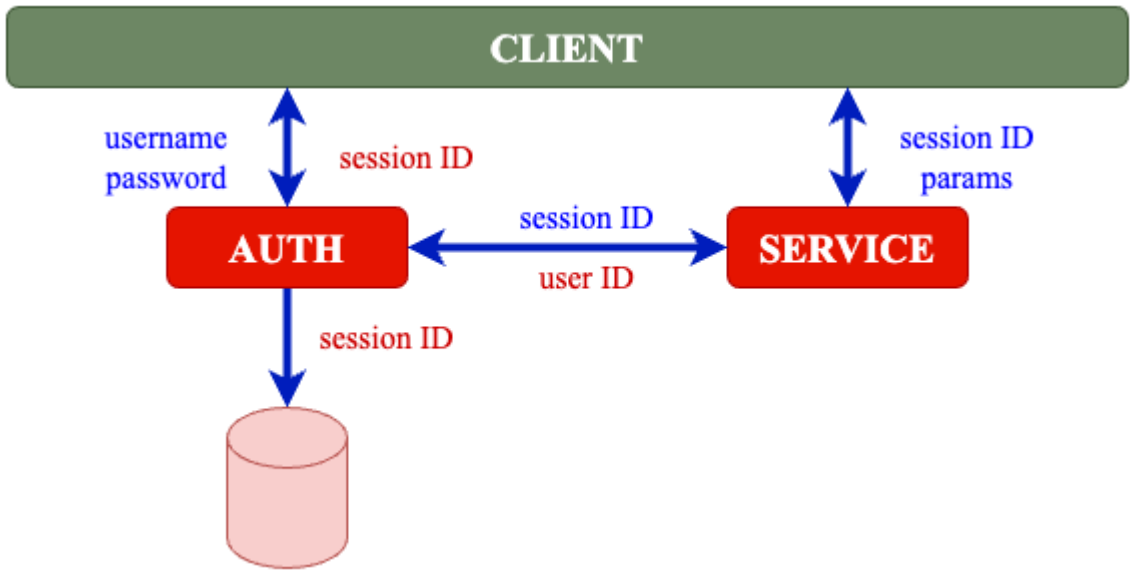
Session vs Token Based Authentication

Mục đích tài liệu

- Mọi request đều cần xác thực mới được xử lý. Theo đó, ngoài thông tin logic của request, cần có thêm thông tin giúp xác thực request là đáng tin cậy.
- Có hai phương pháp phổ biến: session và token.
- Ngoài ra, có thể dùng secret-key nhưng nội dung bài viết sẽ không đề cập đến.

Session based

Mỗi khi client đăng nhập/xác thực, server sẽ tạo ra một session ID và lưu trữ.
Session ID được trả về cho client để sử dụng trong các request sau.
Khi nhận request, server tìm kiếm session ID có tồn tại trong nơi lưu trữ.
Nếu có sẽ trả về các thông tin cần thiết: user ID, role ID, …



Ưu điểm:
Do session ID được lưu ở server, nên server toàn quyền quyết định hiệu lực của session. Trường hợp cần loại bỏ session, có thể thực thiện dễ dàng.
Nhược điểm:
Do tập trung lưu trữ session ID tại một nơi nên đây có khả năng thành thắt cổ chai của hệ thống.
Session ID được đặt trong Header (Auth Bearer) hoặc Cookie. Giá trị cookie có thể được truyền cross-site. Nên nếu đặt trong cookie sẽ phát sinh nguy cơ Session ID bị đánh cắp. Để phòng tránh, tốt nhất ta nên đặt trong header.

Token based

Sau khi xác thực, client sẽ được cấp một token. Token này chứa đầy đủ thông tin của user, được mã hoá bằng khoá bí mật quy định chung cho cả hệ thống.
Client tự lưu trữ token này. Mỗi khi gửi request, client đính kèm token này.
Sau khi nhận được token, server tự giải mã token để có được thông tin và xử lý tiếp.

Ưu điểm:
Mỗi service tự giải mã được token, giảm phụ thuộc vào một service session. Đồng thời, giảm số latency gọi session service (vừa không mất thời gian gọi, cũng không mất thời gian tìm kiếm ở lưu trữ), tăng performance hệ thống.
Nhược điểm:
Do token được lưu trữ ở client, giữa các service cũng không có liên hệ mà tự giải mã token nên phát sinh hai vấn đề. Thứ nhất, server không thể ngay lập tức vô hiệu hóa token, nếu attacker có được token sẽ truy cập không giới hạn vào hệ thống đến khi token bị hết hạn. Thứ hai, phải có chiến lược quản lý khóa bí mật thật tốt (vì lúc này, mọi public service đều có khoá để giải mã).

Tổng kết

	Session	**Token**
Nơi lưu trữ	Server	Client
Phương thức gửi	Header, cookie	Header
Phương thức để server xác thực	Tìm kiếm trong storage có tồn tại	Tự giải mã token, kiểm tra expire time của token có hết hạn.
Server ngay lập tức vô hiệu hoá	Có thể, vì lưu trữ ở server	Không thể, vì lưu trữ ở client
Phương thức tấn công	Main-in-the-middle, cross-site	Main-in-the-middle, lấy token, lấy secrete-key
Trường hợp sử dụng	user-to-server	server-to-server