FLEX - Authentication and Authorization

Simon Härer, Michael Sperber

Created: 2024-06-10 Mon 07:48

Authentication and Authorization

No microservice should be responsible for authentication on their own:

- User credentials should not reside on each service's data store
- Implementing secure authentication and authorization is hard, faulty implementation can lead to catastrophic results.
- Thus, there should be one well-tested auth method
- However, each service must be able to decide which roles to handle, and which to block

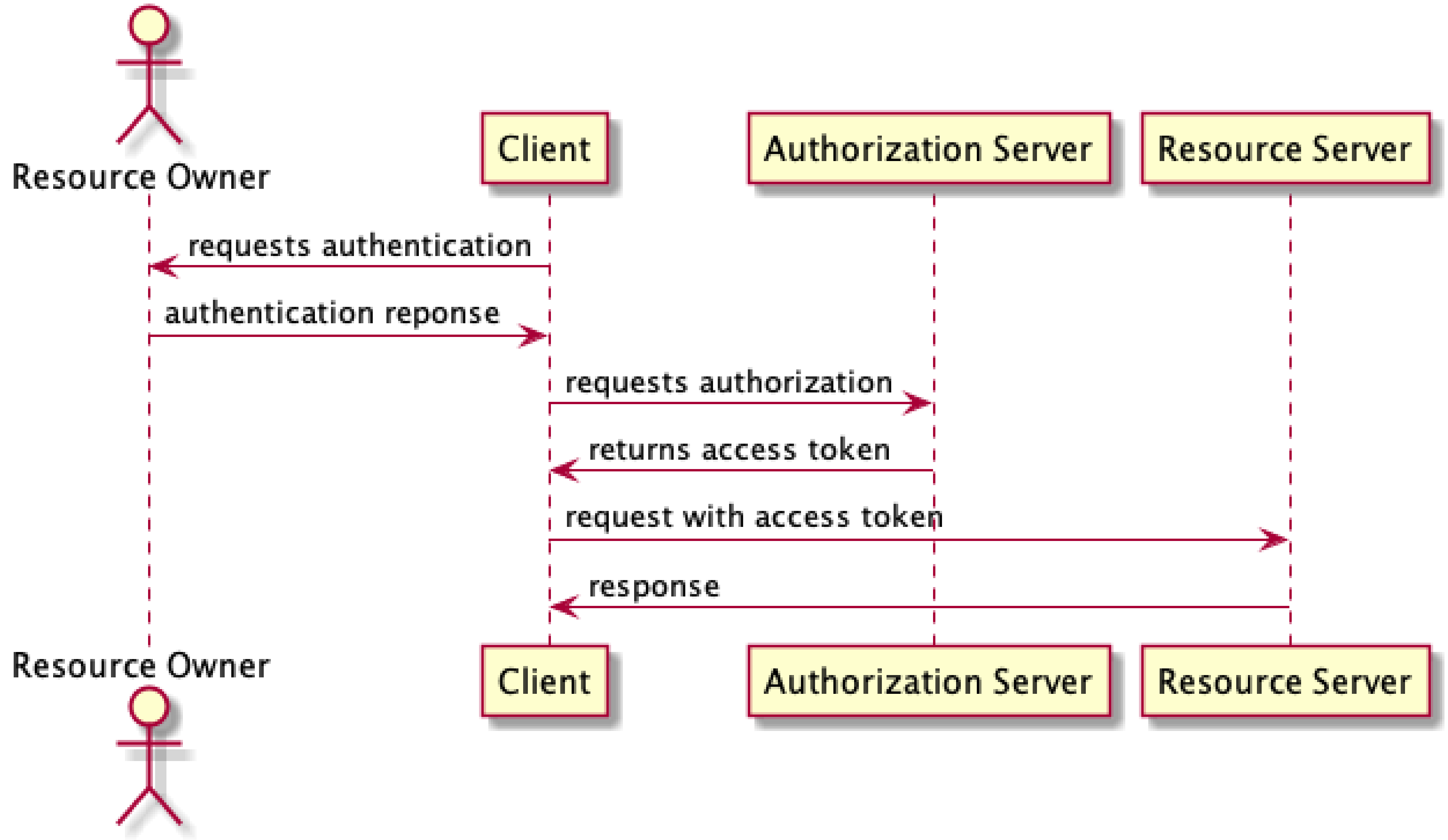
Authentication vs Authorization

- **Authentification** ensure that someone is who they say they are
- **Authorization** determine what an authenticated person may do, and allow only that

OAuth2

OAuth is an open standard for access delegation. It is widely used on the internet and supported by Microsoft, Google, Twitter, and XING. OAuth2 is a good match for a microservice architecture.

OAuth2 Procedure



OAuth2 Procedure

A typical OAuth2 flow looks as follows:

1. The client asks the resource owner, e.g. the user, to provide credentials for authentication. A very common way is to provide a login form
2. The client uses these credentials to request authorization at an authorization server. If granted, the authorization server returns an access token
3. Using this access token, the client can now request resources from a resource server (microservice), if the access token is accepted as valid

The access token mechanism allows us to shift authentication to a single service and authorize the client on multiple other servers.

SAML

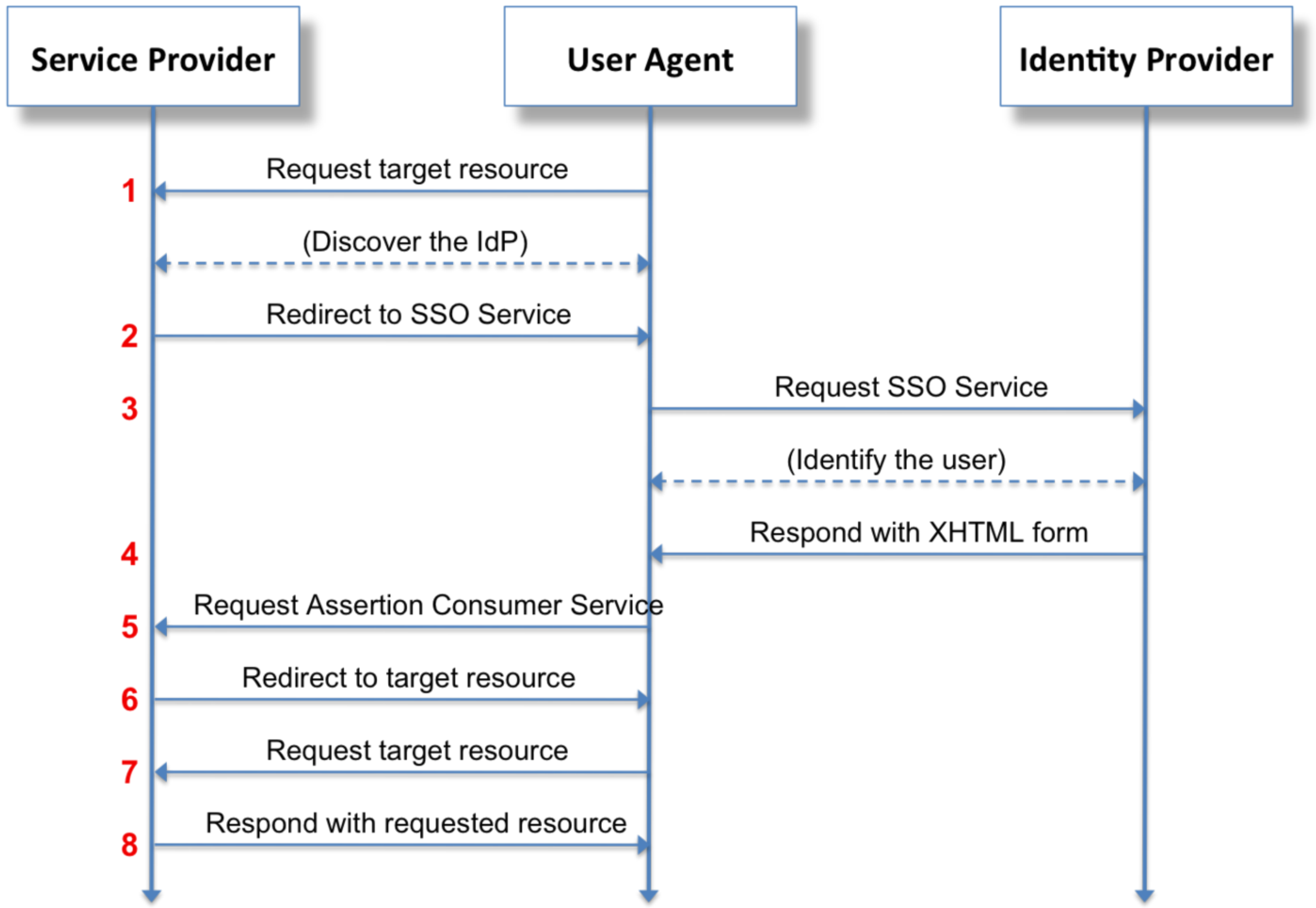
SAML is an open standard for exchanging authentication and authorization information. It includes:

- An XML-based Markup language
- Set of XML-based protocol messages
- Set of protocol message bindings
- Set of profiles including all the above

One common use case is single sign-on in web-browser based applications.

Security Assertion Markup Language, Wikipedia

SAML



Security Assertion Markup Language, Wikipedia

@ctive group

SAML

- Principal (human user) requests a service provider for a specific service
- The service provider asks the identify service for a authentication assertion
- Based on that, the service provider does the services or denies it
- The service provider and identify service are usually different entities

OAuth vs. SAML

OAuth	SAML
authorization	authentication + authorization
resources	applications
web apps	enterprise SSO

Kerberos

Kerberos is a system for distributed *authentication*.

- Three participants: Kerberos server, client, and a server the client wants to use
- The client and the server authenticate against the Kerberos server, as well as the kerberos server authenticates itself against the client and server

Kerberos

- The authentication flow is based on so called *Ticket Granting Ticket*, that the client requests from the Kerberos server
- *TGTs* can be used to request further tickets, that the client can then use to get authenticated on the server. Therefore, the server checks the validity of the ticket using the Kerberos server

Self-Made Cookies

Another way is to implement authentication based on cookies.

- A security service issues cookies that contain a cryptographic identity (using public key cryptography)
- The microservices can verify the validity of the cookie by verifying the identity of the issuer
- The cookie can contain custom information, e.g. needed for authorization

SSL/TLS

- can be used for encryption
- ... but also authentication

Exercise: Authentication/Authorization Design

Choose a method (or two) for authentication and authorization for the banking application. Design a macro architecture that specifies - using the concepts from your methods - how authentication and authorization - specifically the management of different roles - work. Specific questions:

- How and where are secrets managed?
- Evaluate the security implications of your approach
- Evaluate the quality implications of your approach