dzone.com

OAuth2 Tips: Token Validation

Nelia Loginova

5-7 minutos

Bearer Token Types

There are two types of OAuth2 bearer tokens:

- General Token that represents a string that has no meaning for the client (e.g., *2YotnFZFEjr1zCsicMWpAA*). That type of bearer token cannot be validated by the Resource Server without direct communication with an Authorization Server.
- JWT Token represents the JSON object with statements (claims) about the user and token. The JWT token contains three separate parts: header, payload, and signature — each of them are base64-encoded for transferring. JWT tokens are not a part of core

OAuth2 specification but mandatory for use with OpenID Connect. JWT token is the most popular way to exchange information about current authentication between microservices. More details can be found here.

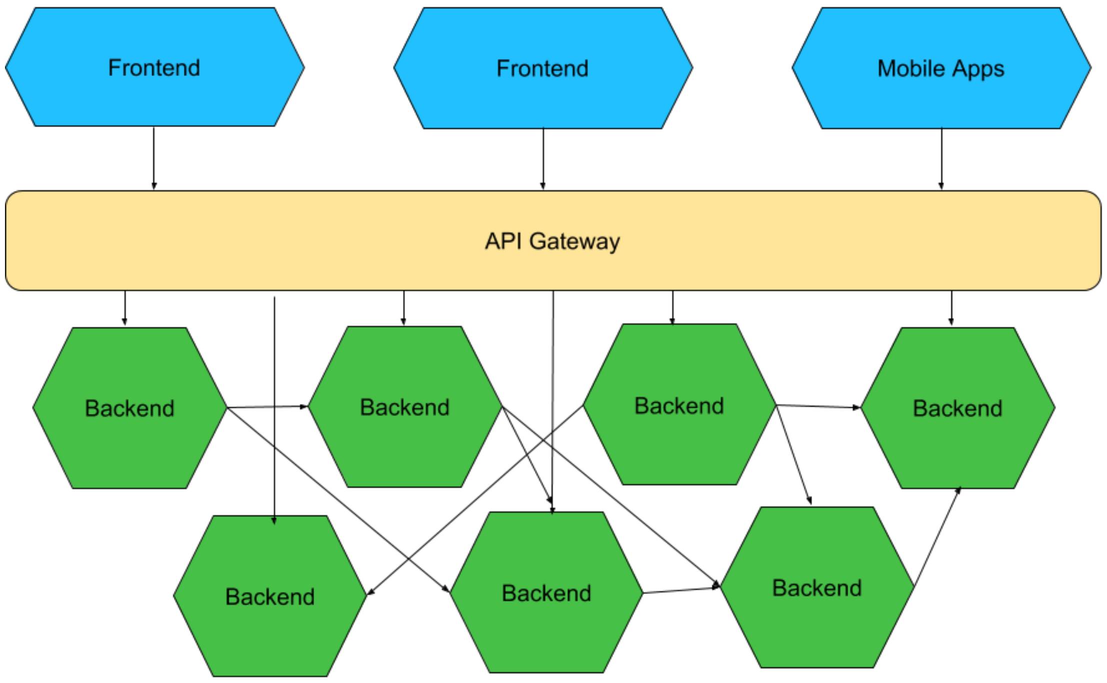
Token Validation Methods

OAuth2 tokens can be validated using the following methods:

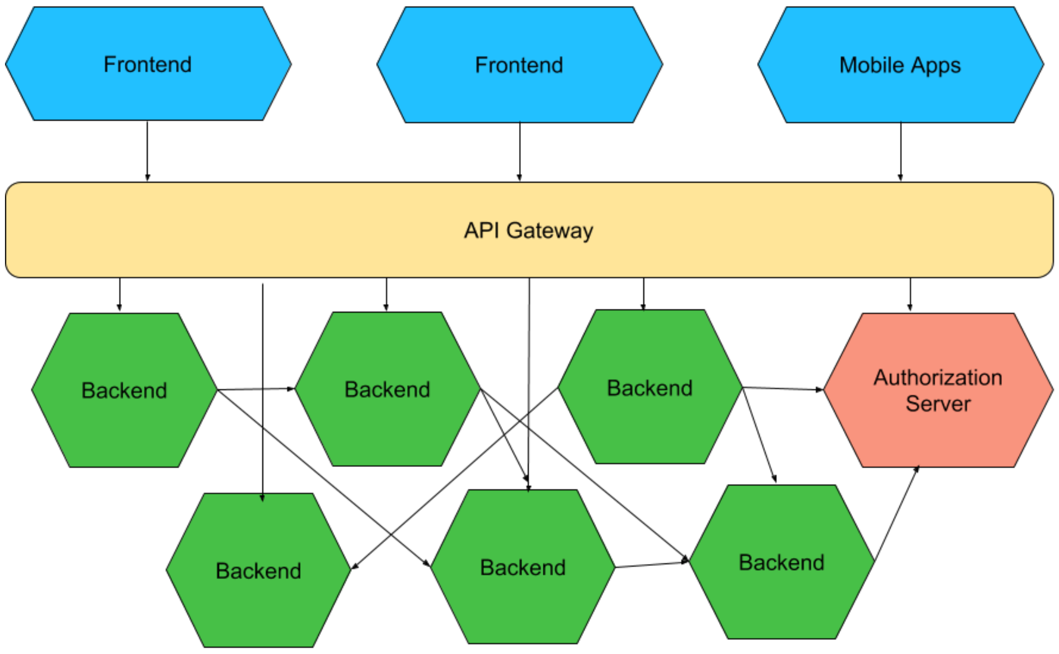
- Introspection. This is a method to get actual token information via special endpoint directly from the Authorization Server. Token information usually includes token type, status (active or not), user, client identifier, available OAuth2 scopes, and expiration time. A detailed description can be found in the specification https://tools.ietf.org/html/rfc7662 .The method requires direct interaction with Authorization Server for every token validation. It has high safety but low performance.
- Token validation by signature (JWT tokens only). This is a method when the token is validated according to its cryptographic signature and all required token information is received from token itself. It means that token validity is verified without interaction with an Authorization server, and if the token was revoked before its expiration, we'll never know about it. So, this method is fast but less secure than introspection.

Put Them Into Practice

Let's imagine that we have an application that has microservice architecture. It has several frontends, mobile applications, API Gateway, and a lot of different useful backends.



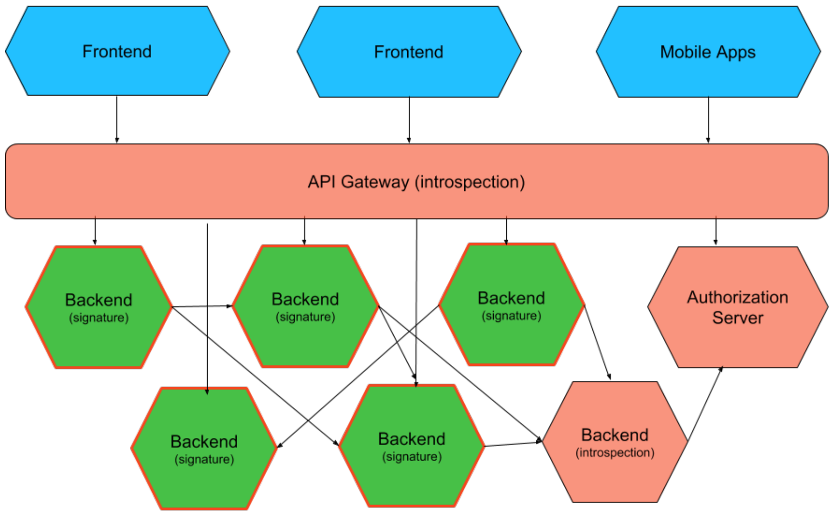
As we follow microservice architecture principles, we move authentication functions to a separate service. It is called Authorization Server in OAuth2 terminology.



Frontends and mobile applications successfully delegate authentication to the Authorization Server. As a result of an authentication process, they get an access token that can be used to call our backend services. So what's next?

Option 1. Secure API Gateway

API Gateway is usually a central entry point to the system, so at that point, we can verify all incoming requests and decide whether they go further or not. The main idea of that approach is to apply strong validation on API Gateway and light validation on backends.



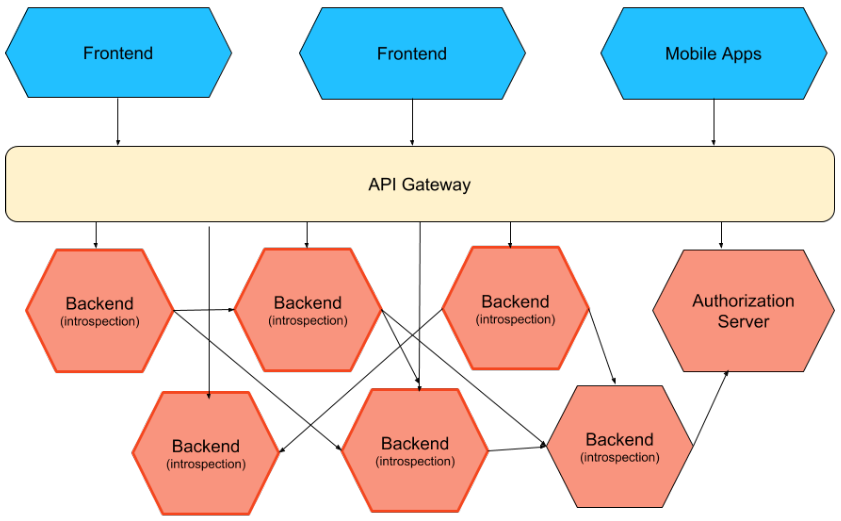
Implementation details:

- API Gateway verifies access token for all incoming requests via introspection. This approach guarantees that the token is valid, not expired or revoked.
- Every backend service validates access token only by signature.
- Some services that have strong security requirements (e.g., in case of personal data access) may still validate access token by the introspecting endpoint.

That option allows getting a good balance between security requirements and system performance.

Option 2. Introspection Only

In some architectures, the API Gateway doesn't have additional functions or there is no API Gateway at all. So, the only possible way is to verify access token on the backends.



Implementation details:

- Every backend service should validate the access token via introspection. Only introspection guarantees that the token is valid, not expired or revoked.

This approach adds one additional call for each service invocation. It can cause performance problems depending on real system architecture.

Summary

JWT token is the standard way to pass authentication between microservices. The token can be verified via introspect endpoint or by signature. The most common way to build built-in token verification into the system is to introspect the token on the API Gateway and verify the signature on other services. If we cannot add additional security functions on API Gateway, the token should be introspected on every service. Of course, other token validation combinations are also possible under specific conditions; a particular choice between the methods should be done according to system architecture and security requirements.

Topics:

oauth tokens, oauth2, authenciation, architechture, jwt, security