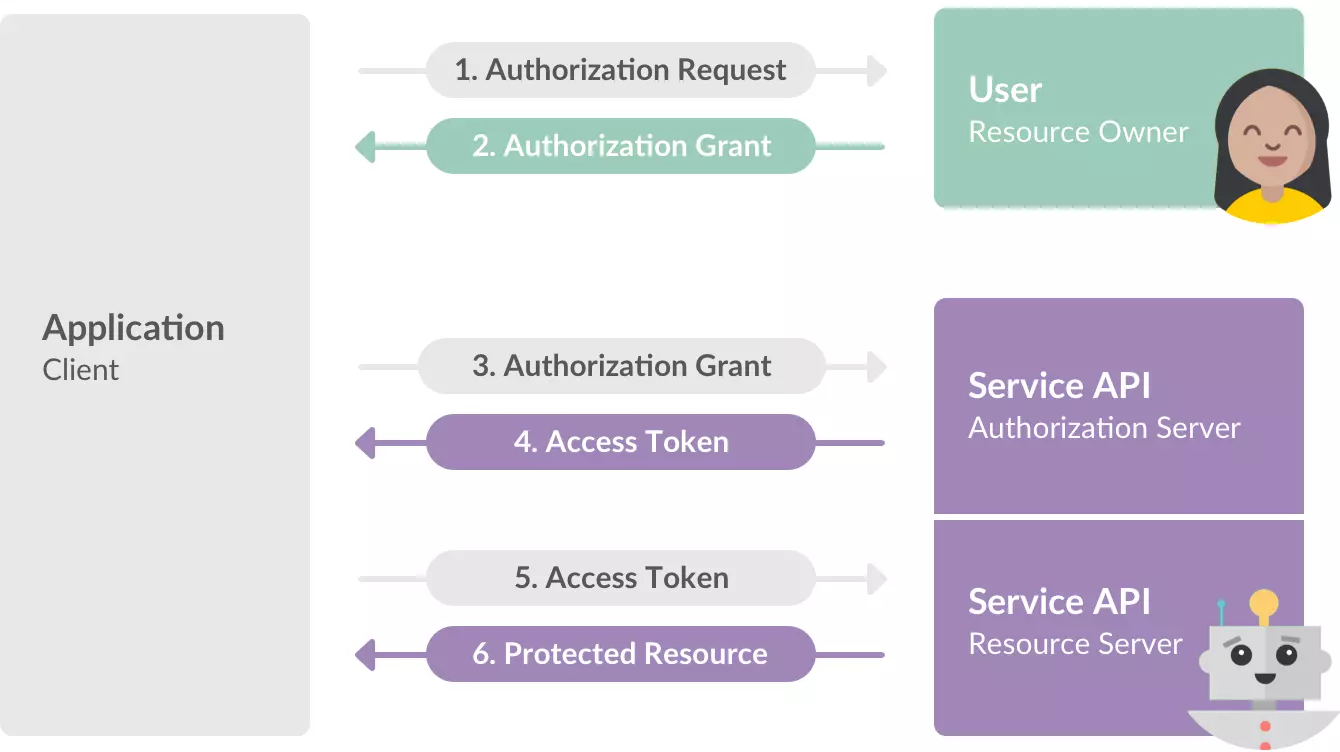
**OAUTH**

OAuth comes from the root of the development of Open ID for Twitter, when its developer, Blaine Cook, received a special request to see if Open ID could authorize dashboard widgets to access their services. After Blaine Cook, Chris Messina and Larry Halff met, they realized that there was no free software standard to be able to delegate access to the different APIs.

In 2007, the discussion group was formed on how what we know today as OAuth should be. Its function at that time was none other than to write a preliminary project with all the ideas and proposals to create a free and open protocol, that is, free software so that any company and user could incorporate it into their websites or online services. Later, DeWitt Clinton, a Google worker, found out about the creation of said group and decided to support said project. The draft was finally finished in July 2007 after only four months and in October 2007 the Oauth Core 1.0 draft came out definitively creating a big change in access to services.

OAuth 2.0 is really an authorization framework, which allows applications to gain limited access to user accounts for some services like Facebook, Google, Twitter and GitHub. Its operation basically consists of delegating the user's authentication permission to the service that manages these accounts, so that it is the service itself that grants access to third-party applications.

**Client**

It would be the application that wants to access the user account of a certain service, such as Facebook, Twitter, Google, etc. For example, if we install an application on the mobile and it asks us for permission to see our data on any of these social networks or platforms. In this way we will avoid having to fill in all the content again and it will take it automatically.

**User**

The user is the one who authorizes the application to access his account, through a pop-up window that asks for authorization, and information about the data that is going to be shared with the new service is usually included. When we try to link any program with Facebook or Twitter, for example, we will have to authorize the application that can transfer that information.

**Server**

The authorization server receives access requests from applications that want to use the login of some of the services such as Facebook, Twitter or Google, for example, to log in to a web page, game, etc. This server is responsible for verifying the identity of the user and the service requesting access, allowing or denying access.

**Bibliography**

Espinosa, O. (23 of July 2021). *Redes Zone*. Obtained from https://www.redeszone.net/tutoriales/seguridad/que-es-oauth/

230 -300 resumen

No cuenta capturas ni bibliografía

¿Para qué sirve?

una plataforma que permite usuarios para ejecutar y alojar sus aplicaciones web en la infraestructura de Google.

Entorno

El entorno de Google App Engine contiene algunas características como las siguientes:

* Servicio web dinámico, con soporte completo para tecnologías web comunes.
* Almacenamiento persistente con consultas, clasificación y transacciones.
* Escalado automático y equilibrio de carga.
* API para autenticar usuarios y enviar correo electrónico usando Cuentas de Google.
* Un entorno de desarrollo local con todas las funciones que simula Google App Engine en la computadora del usuario.

What is the functionality of google app engine?

A platform that enables users to run and host their web applications on Google's infrastructure.

The Google App Engine environment contains the following features

Dynamic web service, with full support for common web technologies.

Persistent storage with queries, classification and transactions

Automatic scaling and load balancing

API to authenticate users and send email using Google Accounts

A full-featured local development environment that simulates Google App Engine on the user's

computer

**What is it for?**

A platform that enables users to run and host their web applications on Google's infrastructure.

Google App Engine is a service that allows you to host web applications within Google's infrastructure, which allows you to maintain and scale more easily. It allows the user the free use of the domain appspot.com, in turn has the option to use your own domain. It also provides the option to limit access to the website to members of the organization or share it to the global network.

**Development environment**

The Google App Engine environment contains the following features:

* Dynamic web service, with full support for common web technologies.
* Persistent storage with queries, classification and transactions
* Automatic scaling and load balancing
* API to authenticate users and send email using Google Accounts
* A full-featured local development environment that simulates Google App Engine on the user's computer.
* Google App Engine es **del tipo Plataforma como Servicio o Platform as a Service (PaaS)**, nos permite publicar aplicaciones web en línea sin necesidad de preocuparnos por la parte de la infraestructura y con un enfoque 100% en la construcción de nuestra aplicación.

Google App Engine is of the type Platform as a Service or Platform as a Service (PaaS), it allows us to publish web applications online without having to worry about the infrastructure part and with a 100% focus on the construction of our application.

Guilherme Magalhaes, Eduardo Roloff. (August of 2012). *ResearchGate*. Obtained of https://www.researchgate.net/publication/296707221\_Developing\_on\_Google\_App\_Engine

Zahariev, A. (27 of April of 2009). *Helsinki University of Technology*. Obtained of http://www.cse.hut.fi/en/publications/B/5/papers/1Zahariev\_final.pdf