Posibilidades para implementar la seguridad en un API Rest

Implementa la seguridad de tu API Rest con Spring Boot

Mecanismos de autenticación en una web con UI

- La mayoría de las webs que utilizamos suelen proveer un mecanismo para la autorización a través de un formulario de login.
- Normalmente se les proporcionan dos datos: usuario y contraseña.
- El servidor suele ser el encargado de almacenar la sesión (usuario activo en la aplicación).
- ¿Qué hacer si nuestra aplicación no tiene UI? Como por ejemplo, con un API REST.

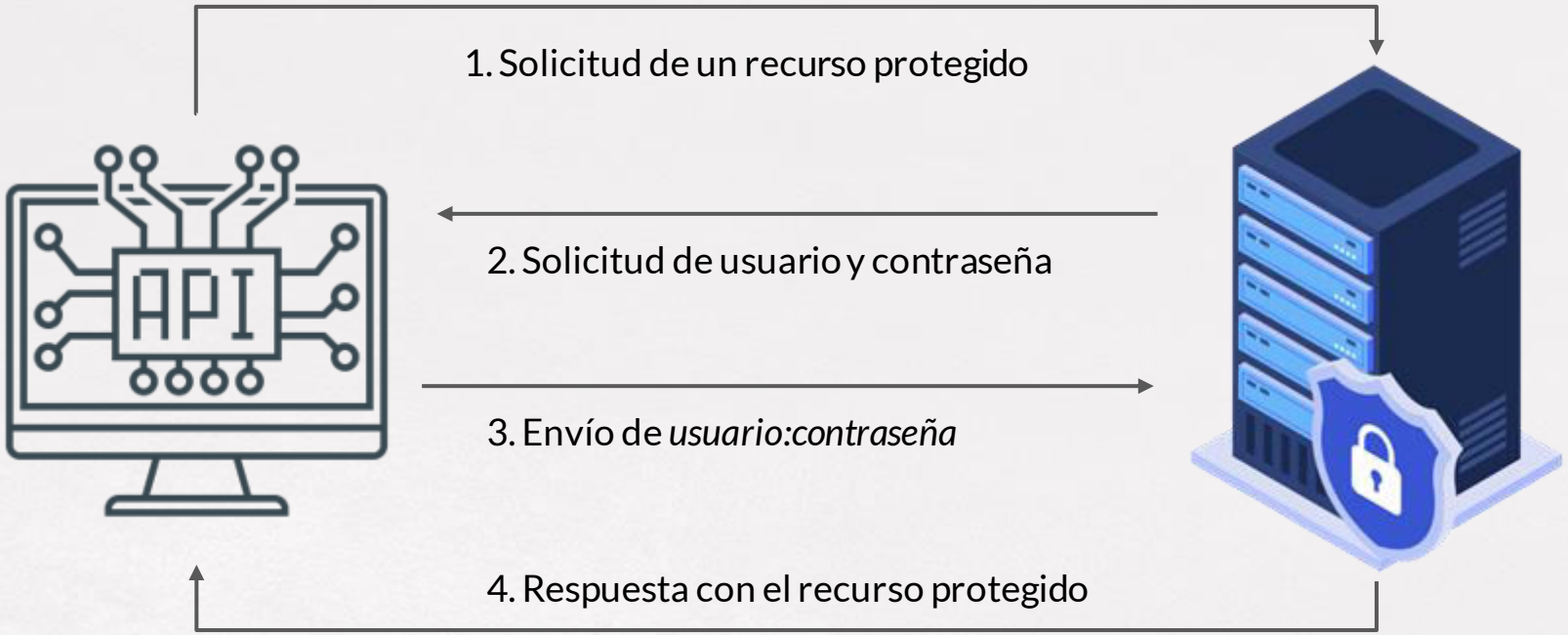
Diferentes mecanismos de autenticación en un API REST

- Básica (*basic*)
- JWT (Json Web Tokens)
- OAuth 2.0

Autenticación básica

- Mecanismo más elemental de autenticación a través de HTTP.
- Definido en el RFC 1945 y RFC 2617
- *No es elegante, pero cumple su función.*
- Es simple, pero poco fiable.
- No obliga al uso de cookies ni de formularios de acceso.

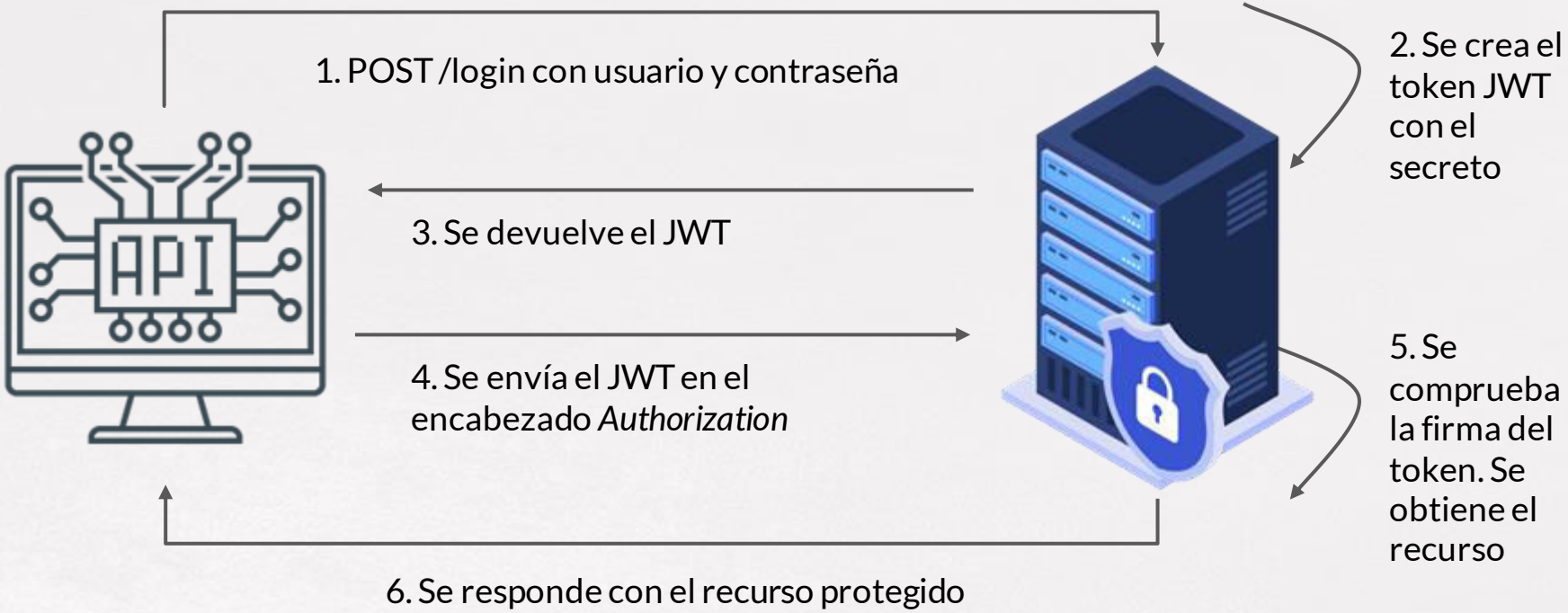
Autenticación básica



Autenticación utilizando JWT

- Json Web Token.
- Realmente no es un estándar de autenticación.
- Se trata de un estándar para la creación de tokens de acceso que permiten propagar la identidad y privilegios.
- La información puede ser verificada y confiable, porque está firmada digitalmente.
- No obliga a que el servidor maneje la sesión.

Autenticación utilizando JWT



Autenticación OAuth 2.0

- Estándar abierto para la autorización de APIs.
- Permite compartir información entre sitios sin compartir de la identidad.
- Mecanismo utilizado por grandes compañías, como Google, Facebook, Microsoft, Twitter y Github.
- Implementa diferentes flujos de autenticación: *authorization code flow*, *resource owner password credential flow*, *implicit flow*, ...

Autenticación OAuth 2.0

- Se definen varios roles
  - Dueño del recurso
  - Cliente
  - Servidor de recursos protegidos
  - Servidor de autorización.

Autenticación OAuth 2.0