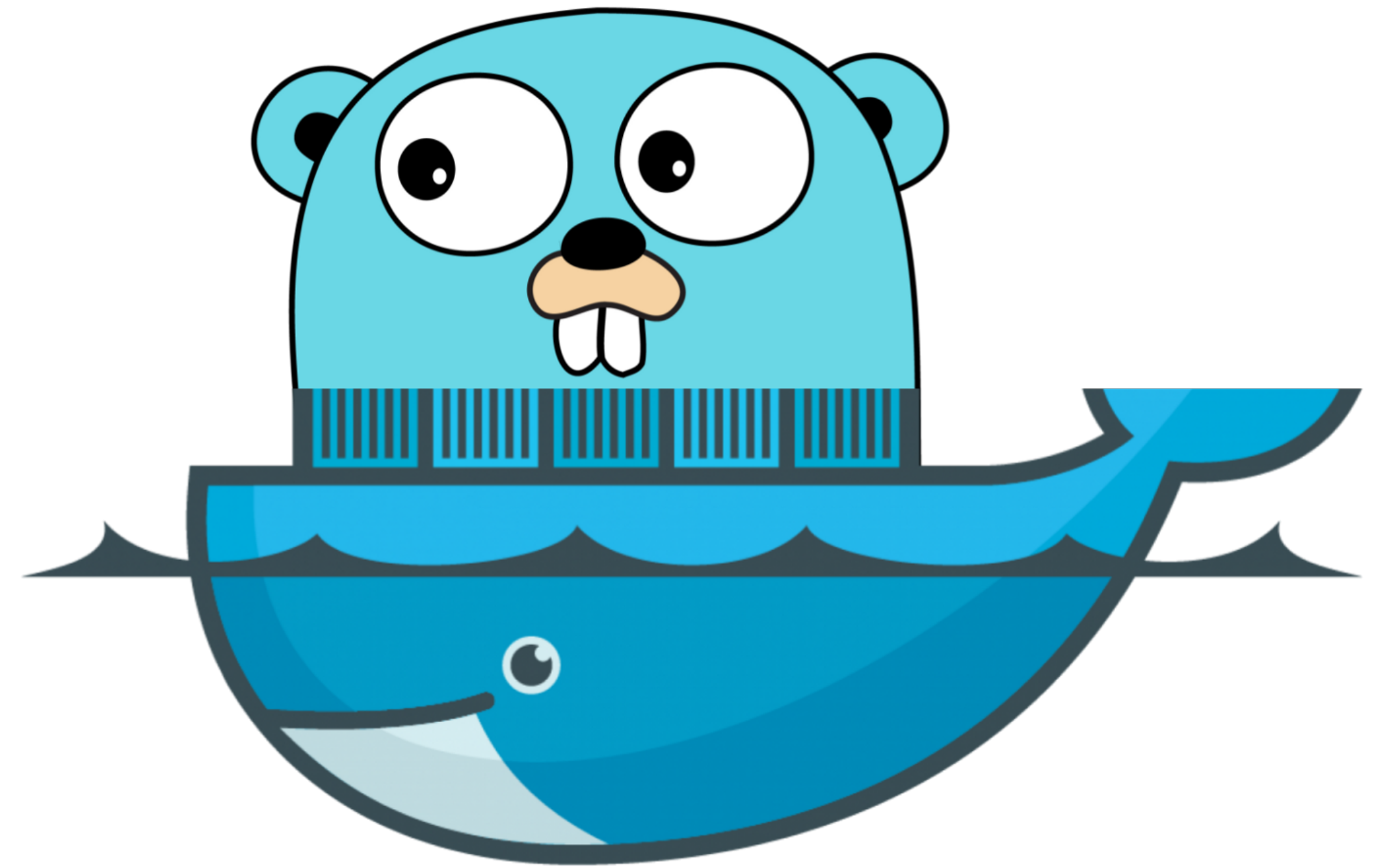


# Golang Engineer Training

Authentication with JWT



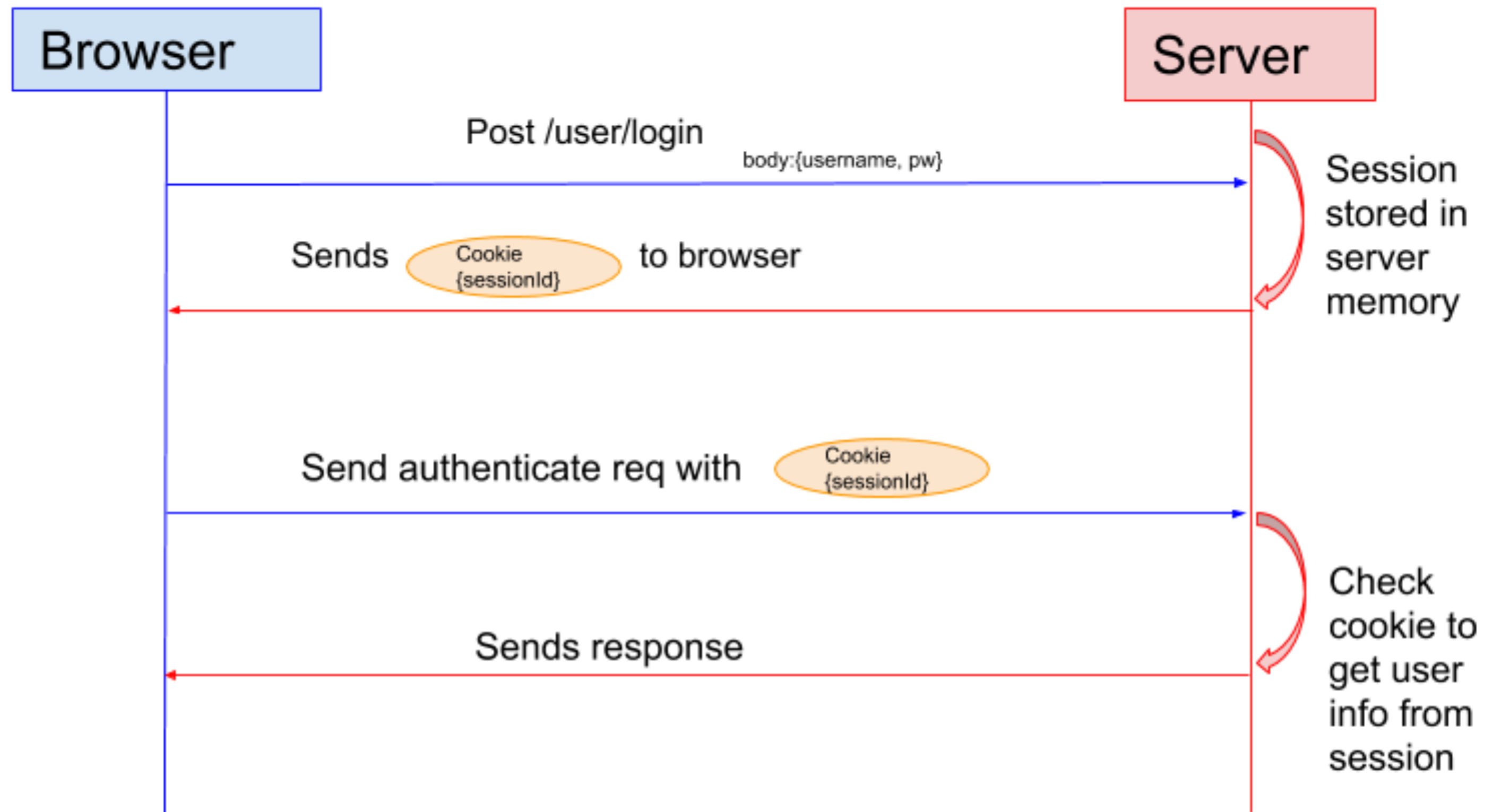
# Agenda

- Authentication - What & Why
- Authentication Flow
- JWT & Practice in Golang
- Linking user data - In practice (bonus)

# Authentication - What & Why

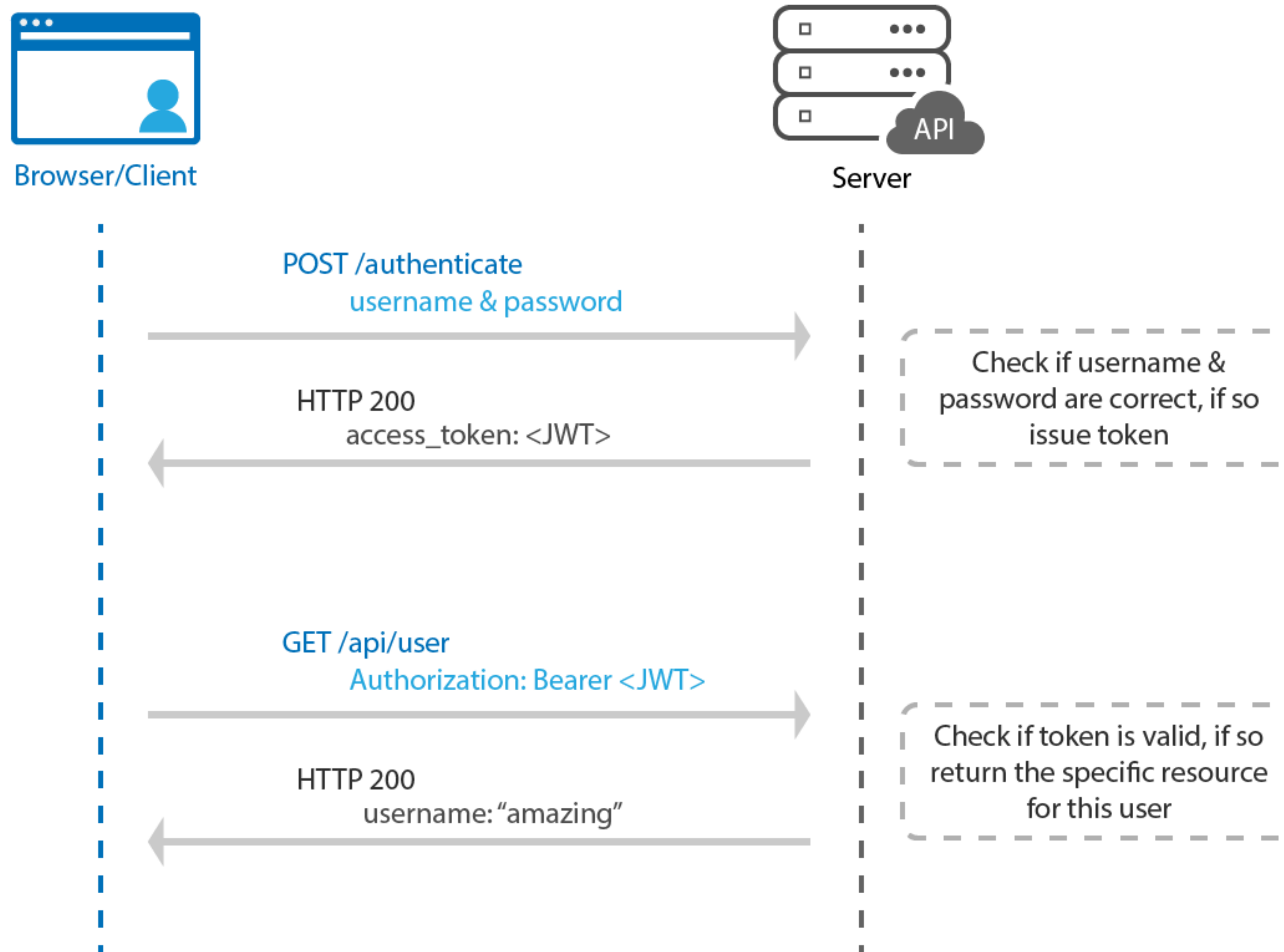
- The process or action of proving or showing something to be true, genuine, or valid (Google Translate)
- Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials. The credentials provided are compared to those on a file in a database of the authorized user's information on a local operating system or within an authentication server (Wikipedia)

# Authentication Flow - Old



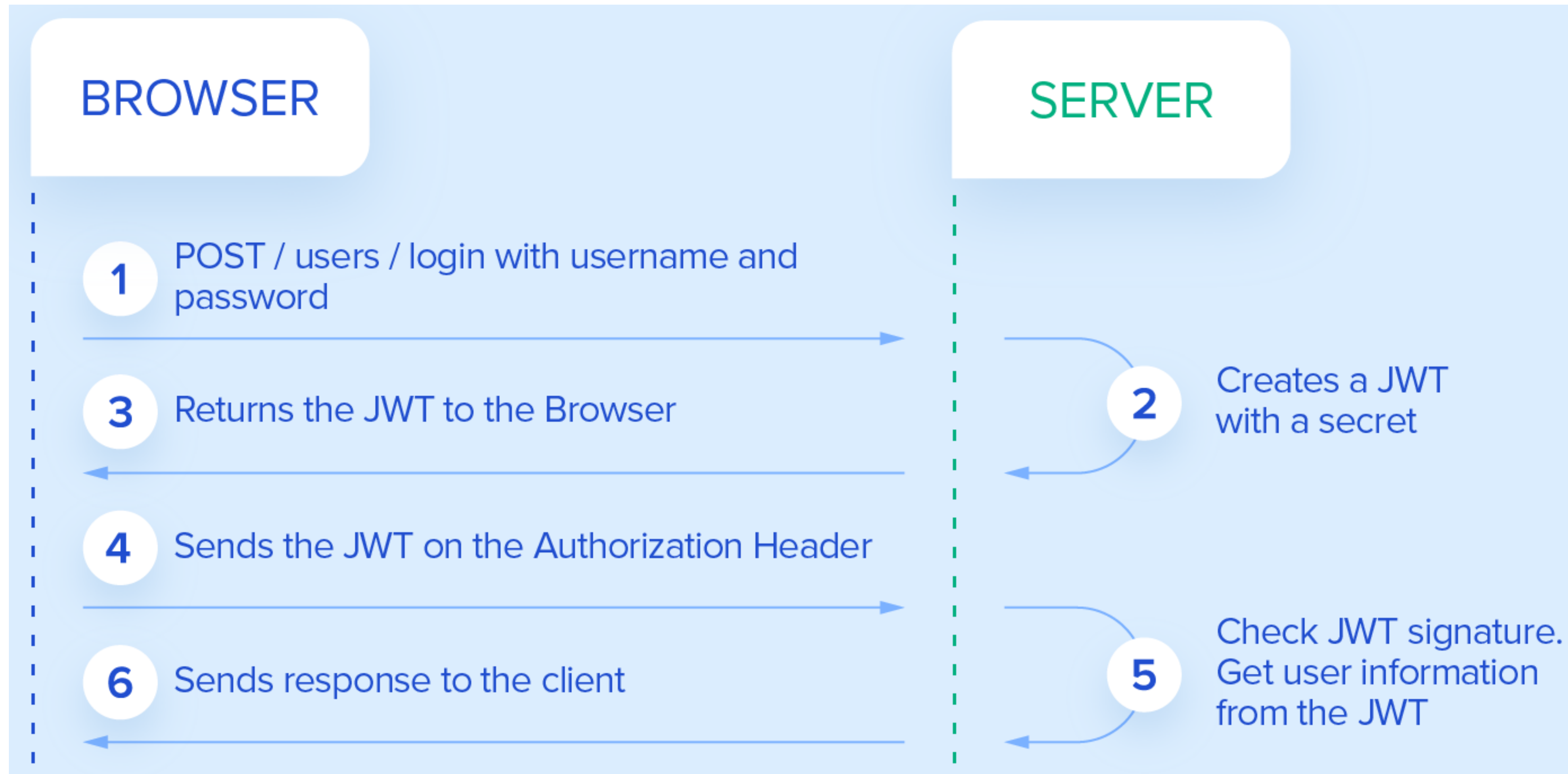
- Easy to use
- Server use session for storing user info.
- Good for web client but very bad for mobile apps.
- Server is stateful (storing user info).

# Authentication Flow - New



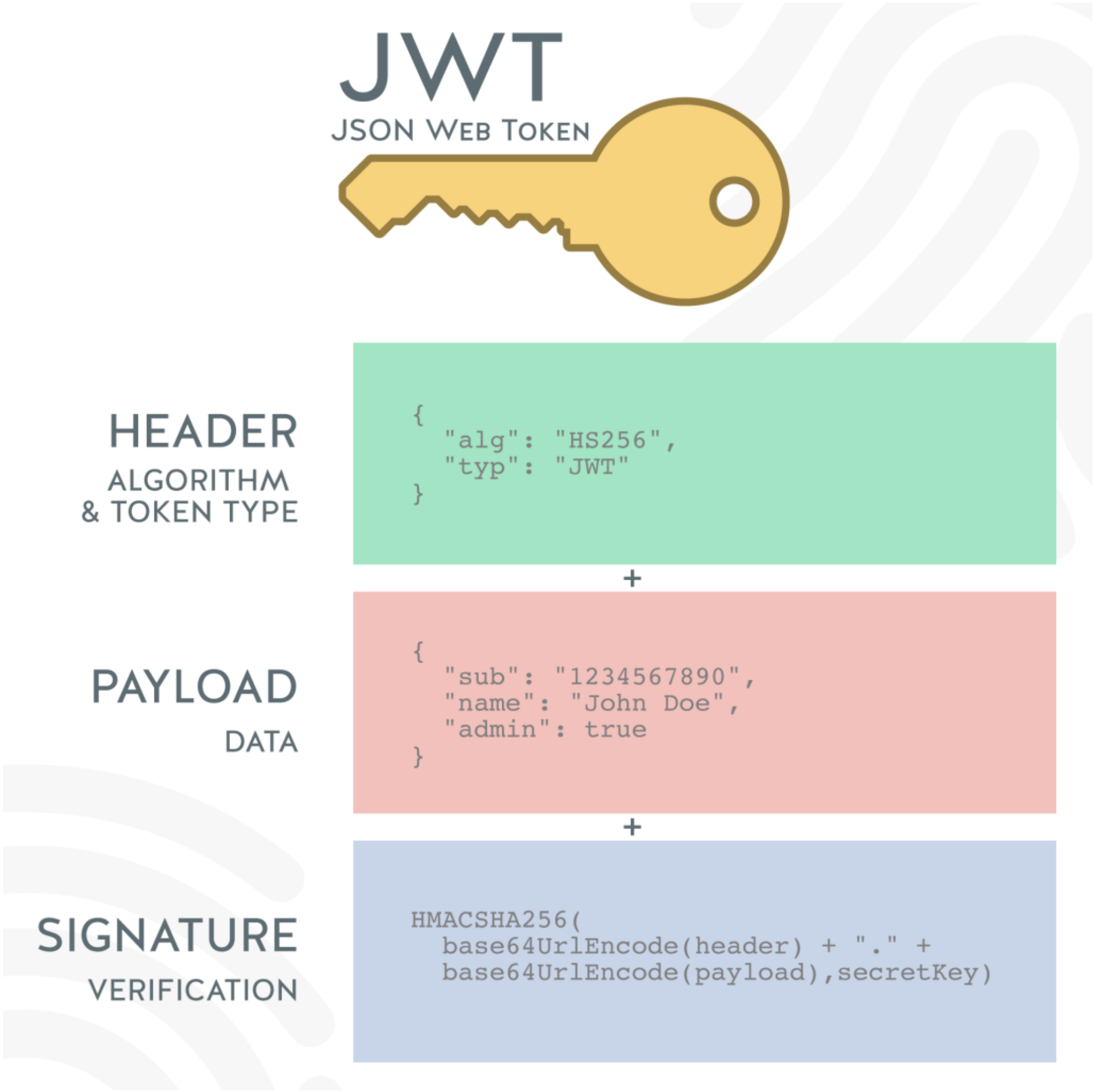
- Use token to identify.
- Good for both web and mobile apps, even for backend to backend.
- Need a mechanism for creating token
- Server is stateless (not storing user token).

# Authentication Flow - JWT





# JWT - Introduction



# JWT - Introduction (.cont)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzkwMjQ.DIyfQ.XbPfbIHMI6arZ3Y922BhjWgQzWXcXNrZ0ogtVhfEd2o

1

Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

2

Payload

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

3

Signature

```
HMACSHA256 (
  BASE64URL(header)
  .
  BASE64URL(payload) ,
  secret)
```



# JWT (.cont)

## JWT in practice

<https://github.com/dgrijalva/jwt-go>

Thank you.