Without JWT

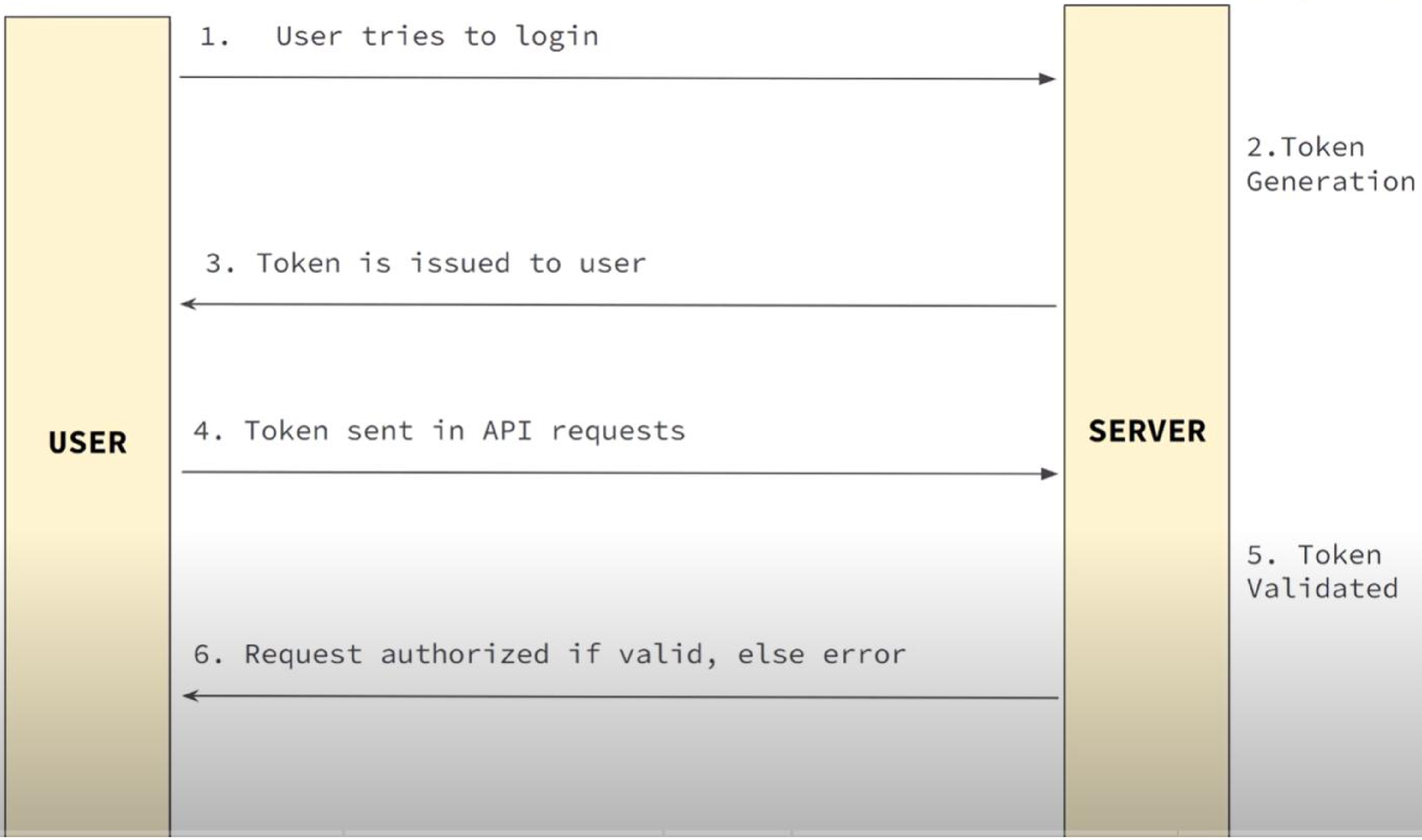
→ *No advanced features like expiration time*

→ *Can be decoded easily*

→ *Should we go for "Custom token system"*

JWT = **JSON Web Token**

JSON Web Tokens are an **open, industry standard**

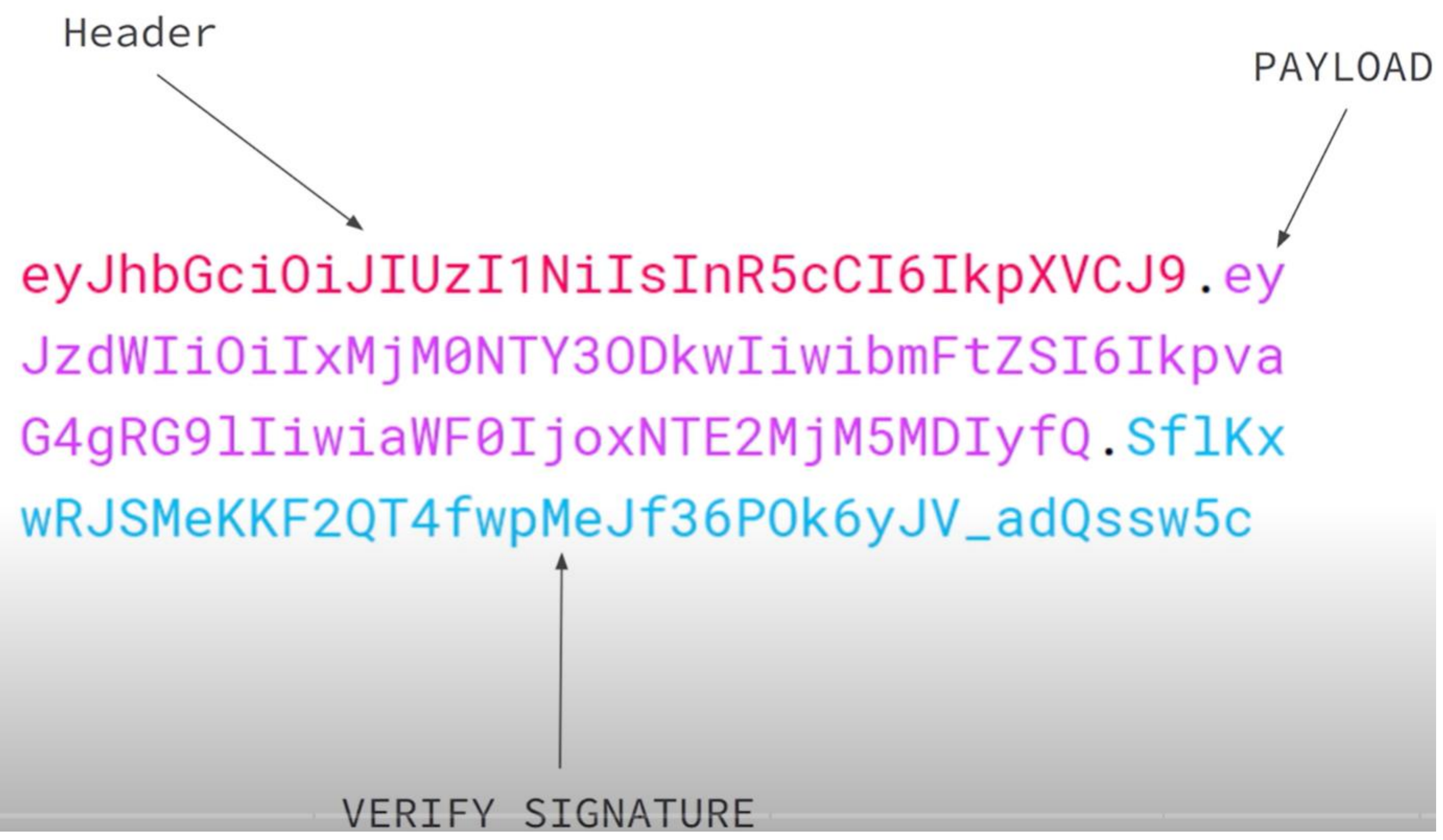

1. User tries to login
2.Token Generation
3. Token is issued to user
USER
4. Token sent in API requests
SERVER
5. Token Validated
6. Request authorized if valid, else error

How is Token sent

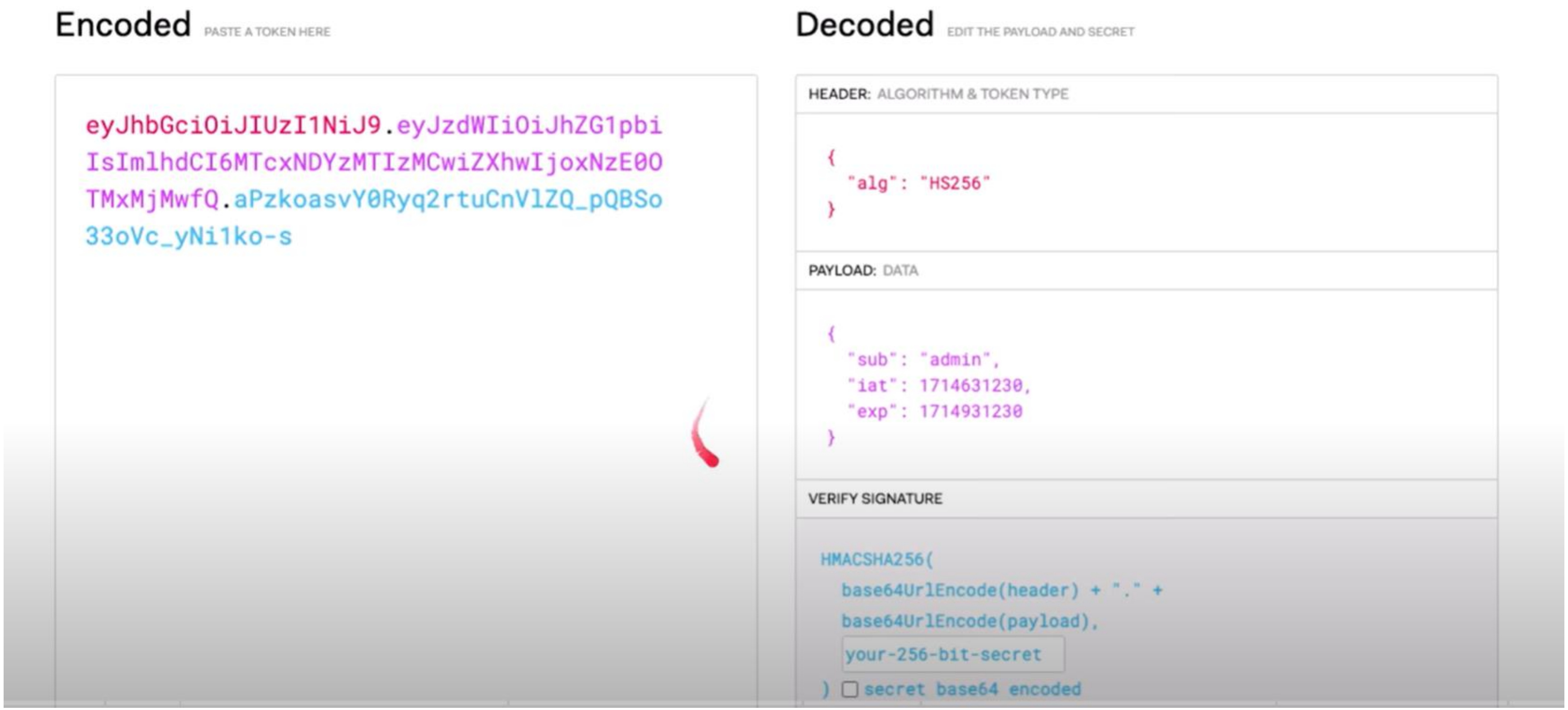
Tokens are sent using HTTP Authorization header

Format

Authorization: Bearer <token>



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbi
IsImlhdCI6MTcxNDYzMTIzMCwiZXhwIjoxNzE0O
TMxMjMwfQ.aPzkoasvY0Ryq2rtuCnVlZQ_pQBSo
33oVc_yNi1ko-s
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256"
}
```

PAYLOAD: DATA

```
{
  "sub": "admin",
  "iat": 1714631230,
  "exp": 1714931230
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

JwtUtils

AuthTokenFilter

AuthEntryPointJwt

SecurityConfig

JwtUtils

→ Contains utility methods for generating, parsing, and validating JWTs.

→Include generating a token from a username, validating a JWT, and extracting the username from a token.

JwtUtils

AuthTokenFilter

AuthEntryPointJwt

SecurityConfig

AuthTokenFilter

→ Filters incoming requests to check for a valid JWT in the header, setting the authentication context if the token is valid.

→Extracts JWT from request header, validates it, and configures the Spring Security context with user details if the token is valid.

JwtUtils

AuthTokenFilter

AuthEntryPointJwt

SecurityConfig

AuthEntryPointJwt

→ Provides custom handling for unauthorized requests, typically when authentication is required but not supplied or valid.

→When an unauthorized request is detected, it logs the error and returns a JSON response with an error message, status code, and the path attempted.

JwtUtils

AuthTokenFilter

AuthEntryPointJwt

SecurityConfig

SecurityConfig

→ Configures Spring Security filters and rules for the application

→Sets up the security filter chain, permitting or denying access based on paths and roles. It also configures session management to stateless, which is crucial for JWT usage.