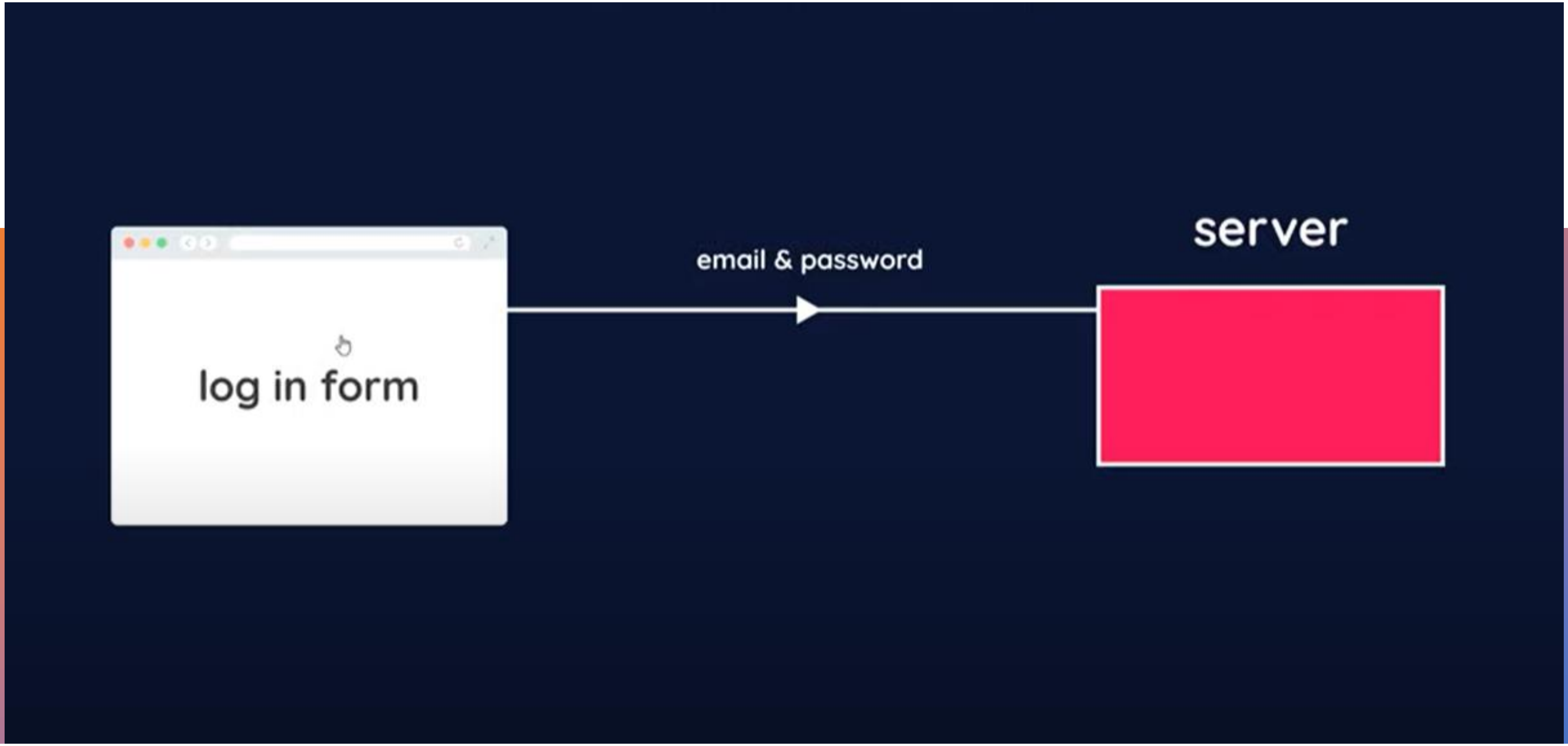
Node Auth Tutorial (JWT)

#10 - JSON Web Tokens

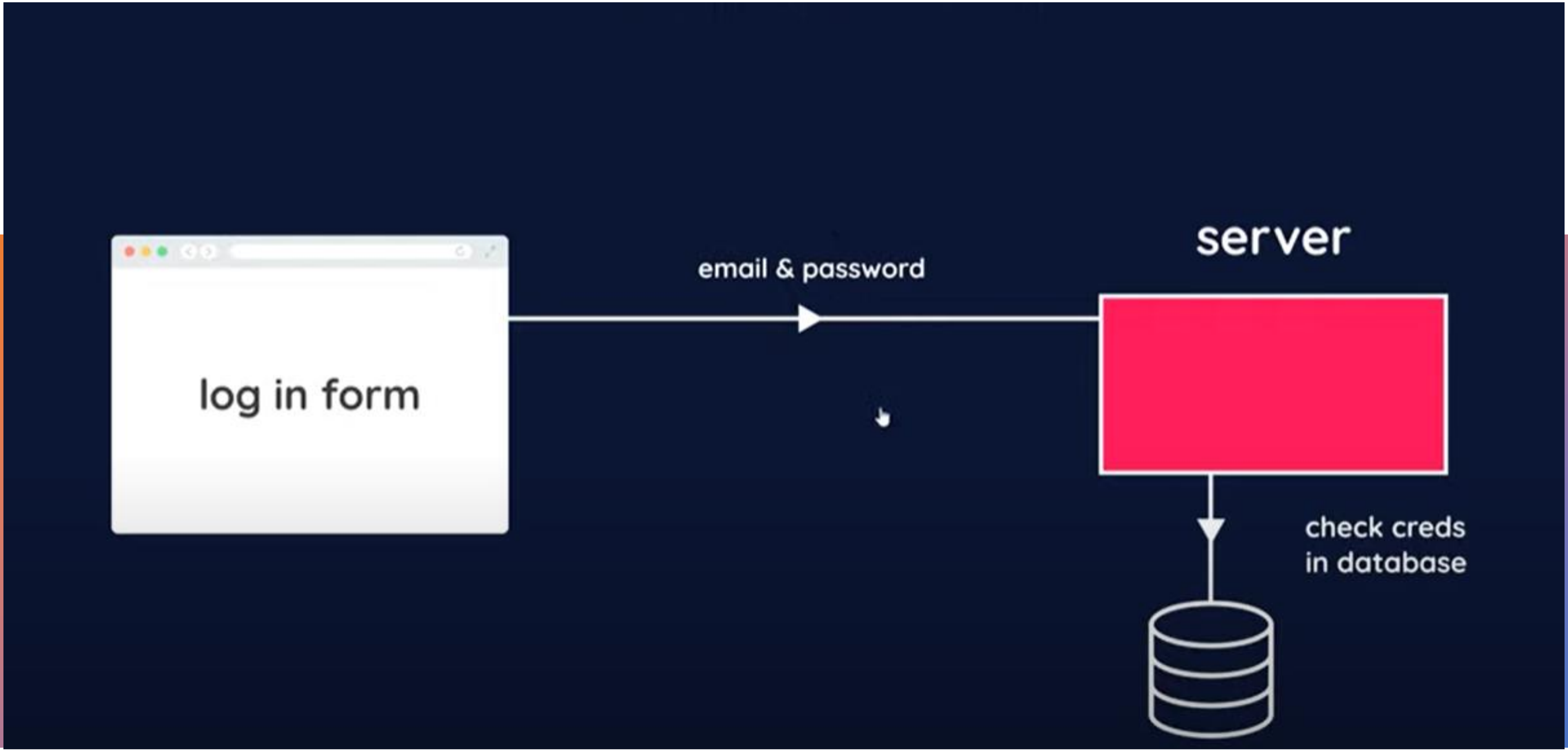
(theory)

In this node authentication tutorial we'll talk a little bit more about JSON web tokens (JWTs) and how we can use them to authenticate users.

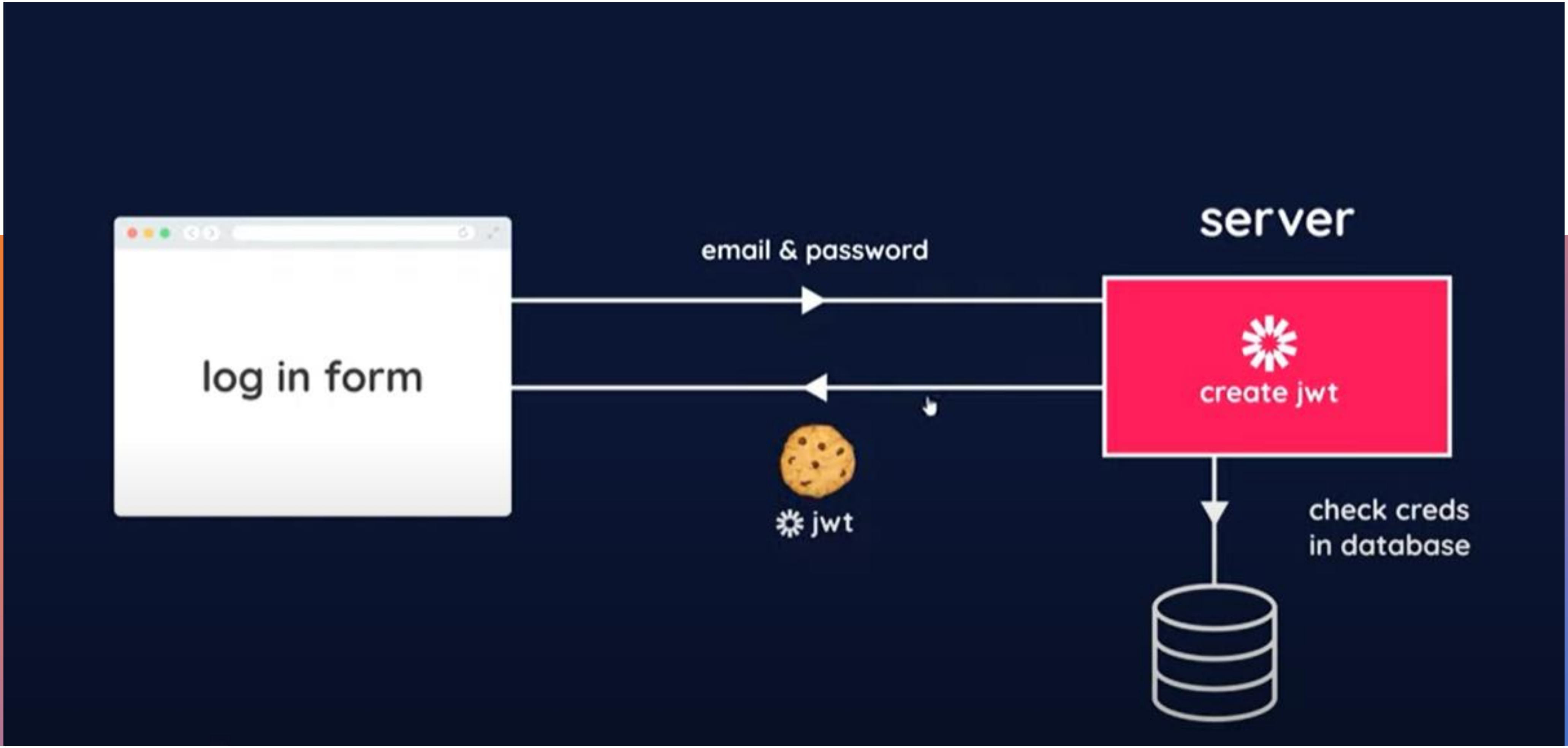
JSON Web Tocken



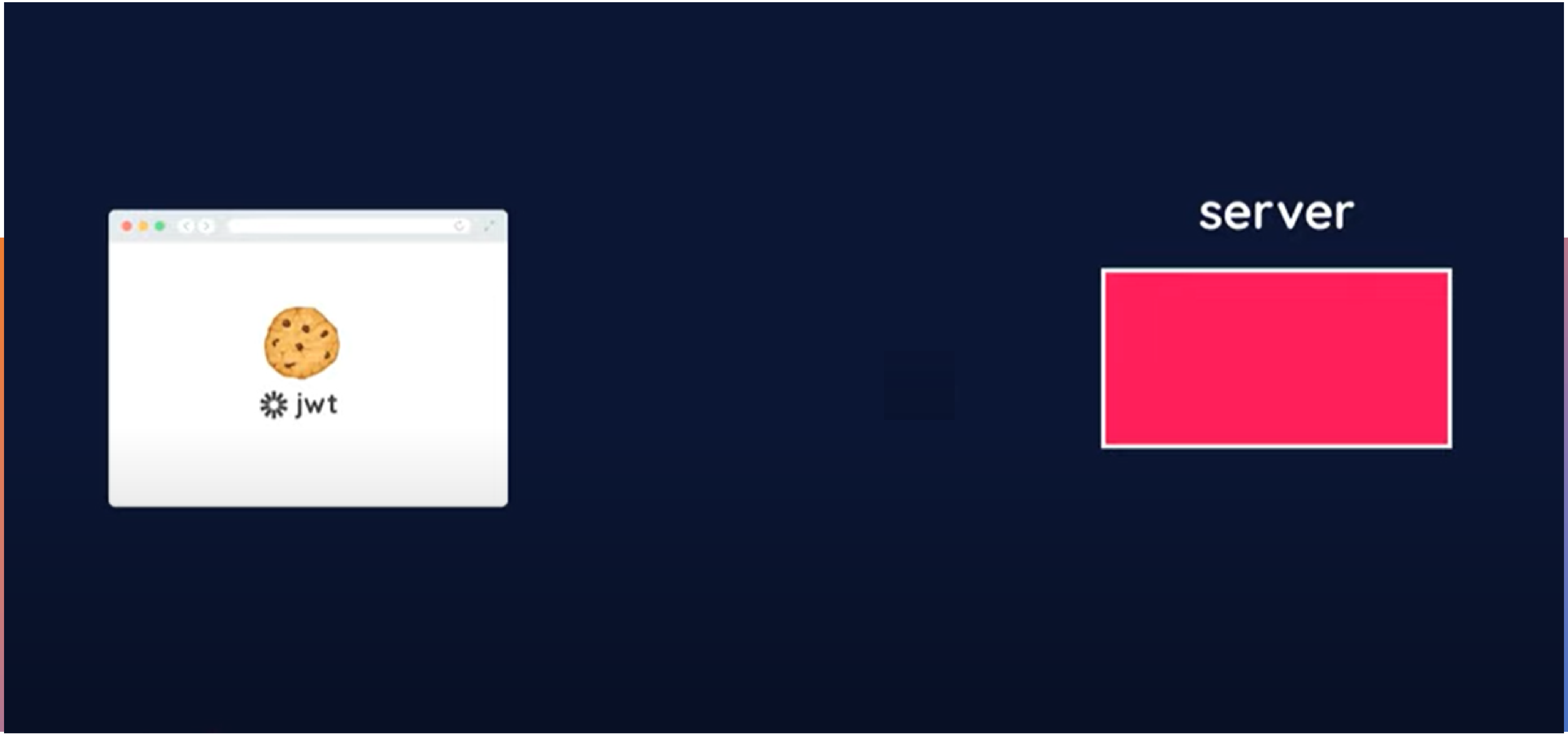
JSON Web Tocken



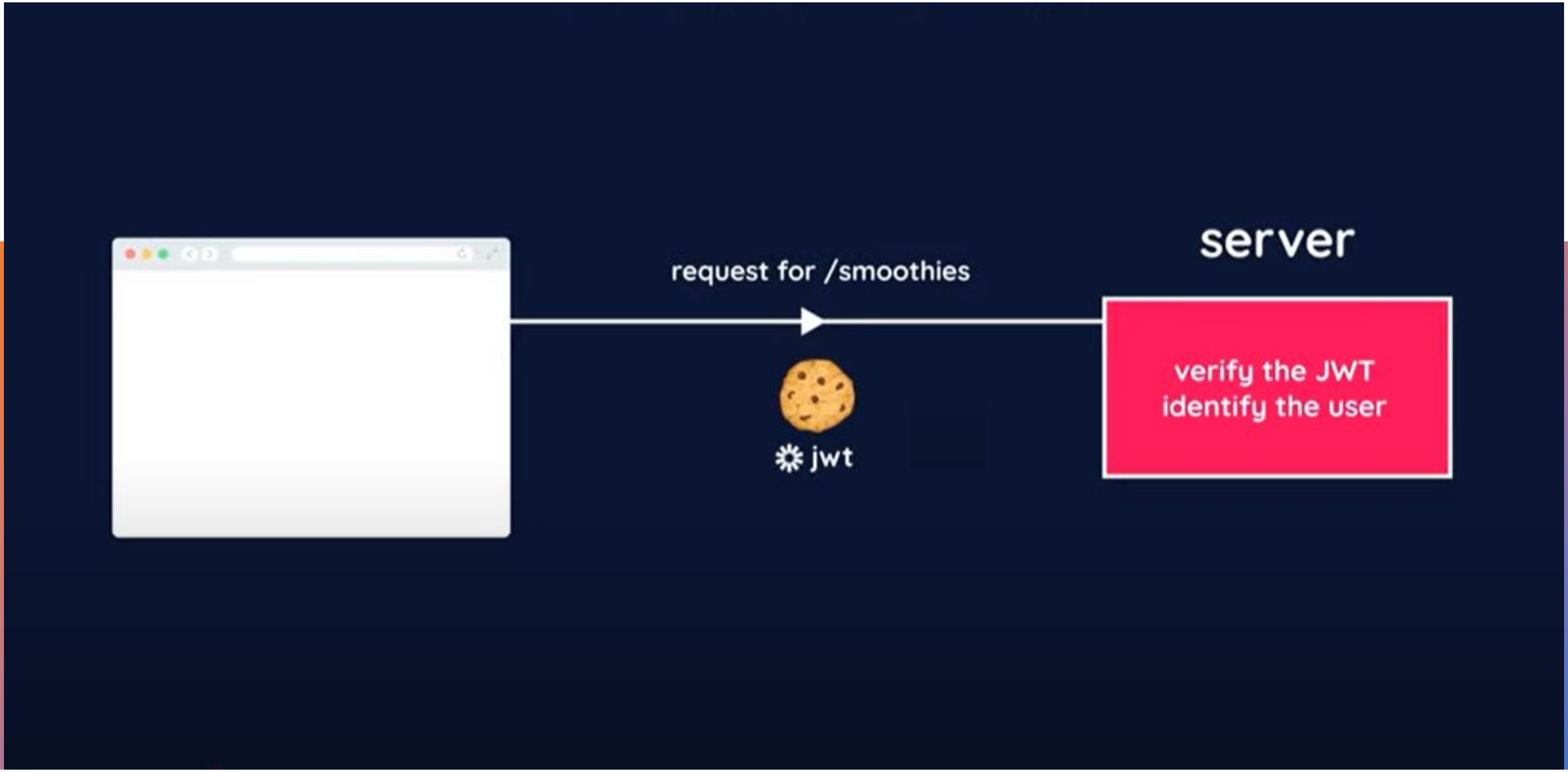
JSON Web Tocken

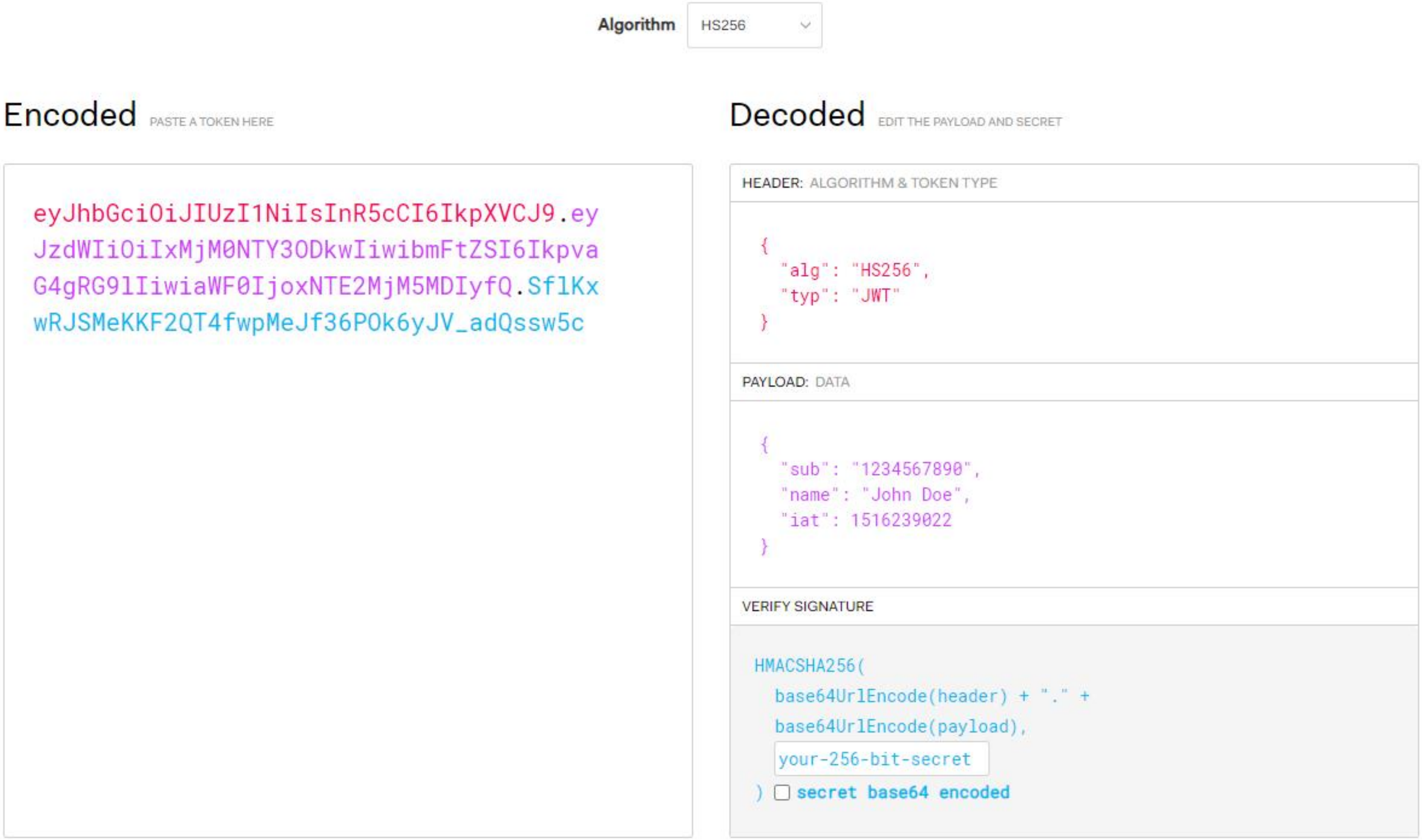


JSON Web Tocken



JSON Web Tocken





Encoded

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

Decoded

HEADER: ALGORITHM & TOKEN TYPE

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

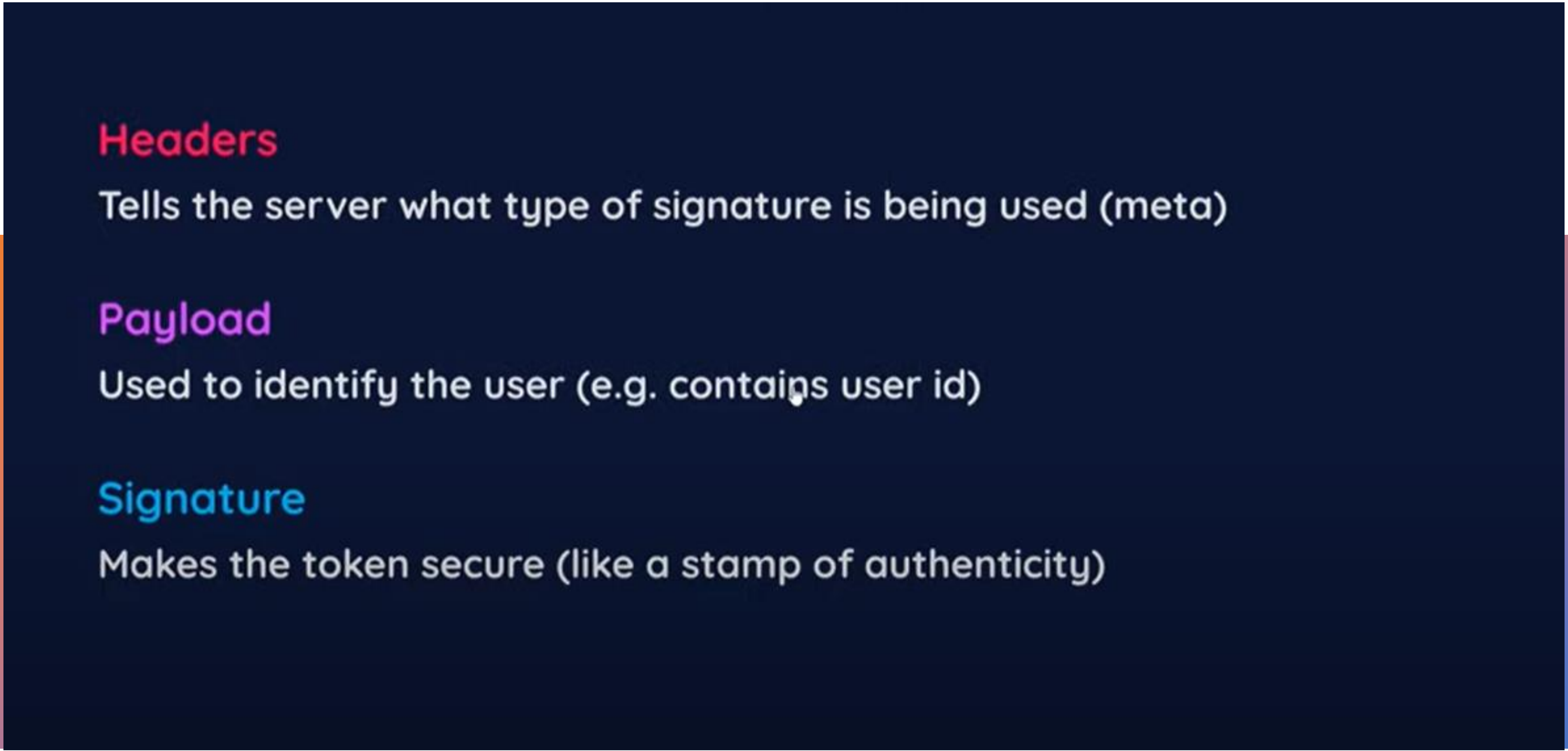
PAYLOAD: DATA

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

VERIFY SIGNATURE

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

JWT Signing



Headers

Tells the server what type of signature is being used (meta)

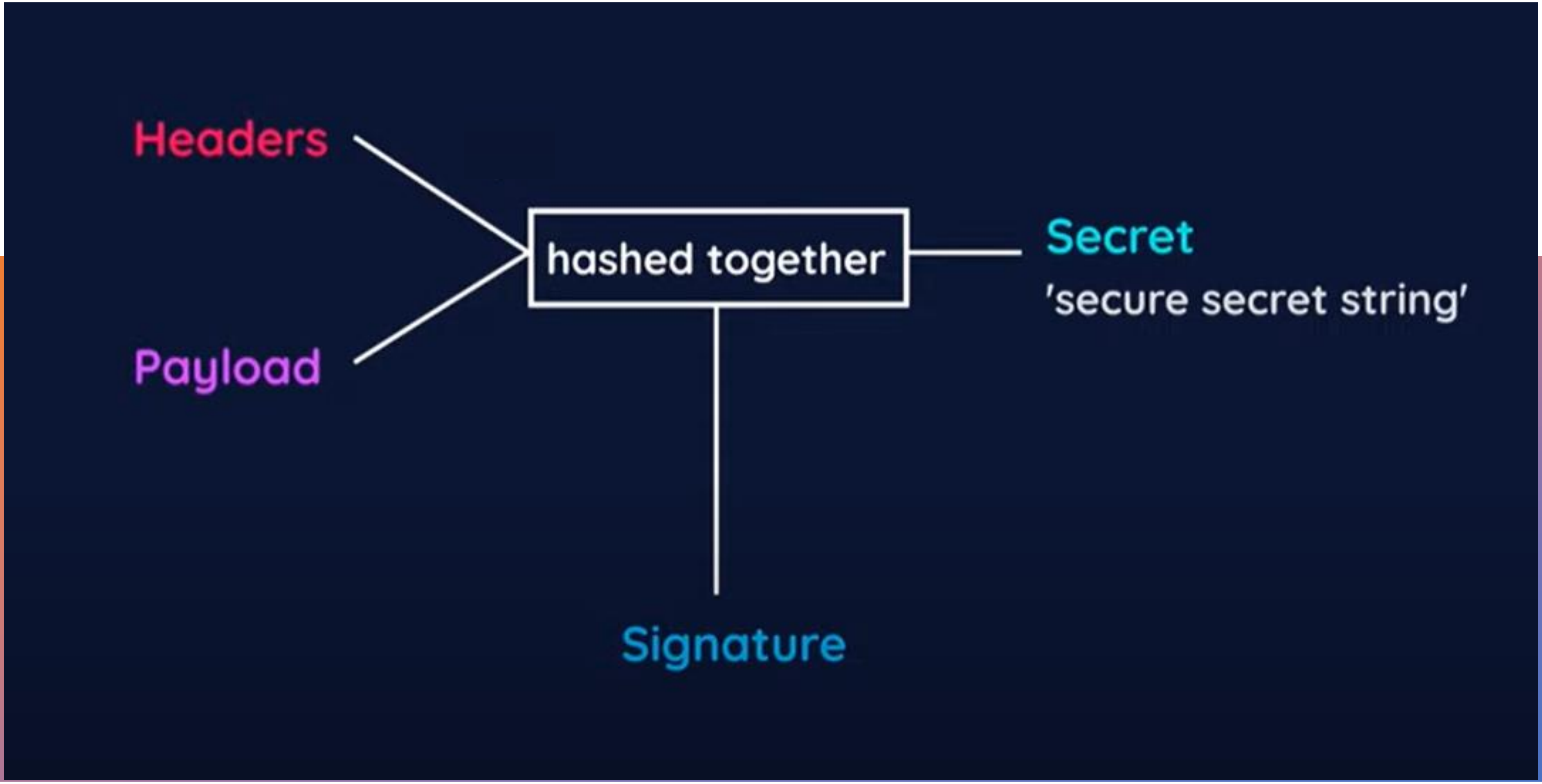
Payload

Used to identify the user (e.g. contains user id)

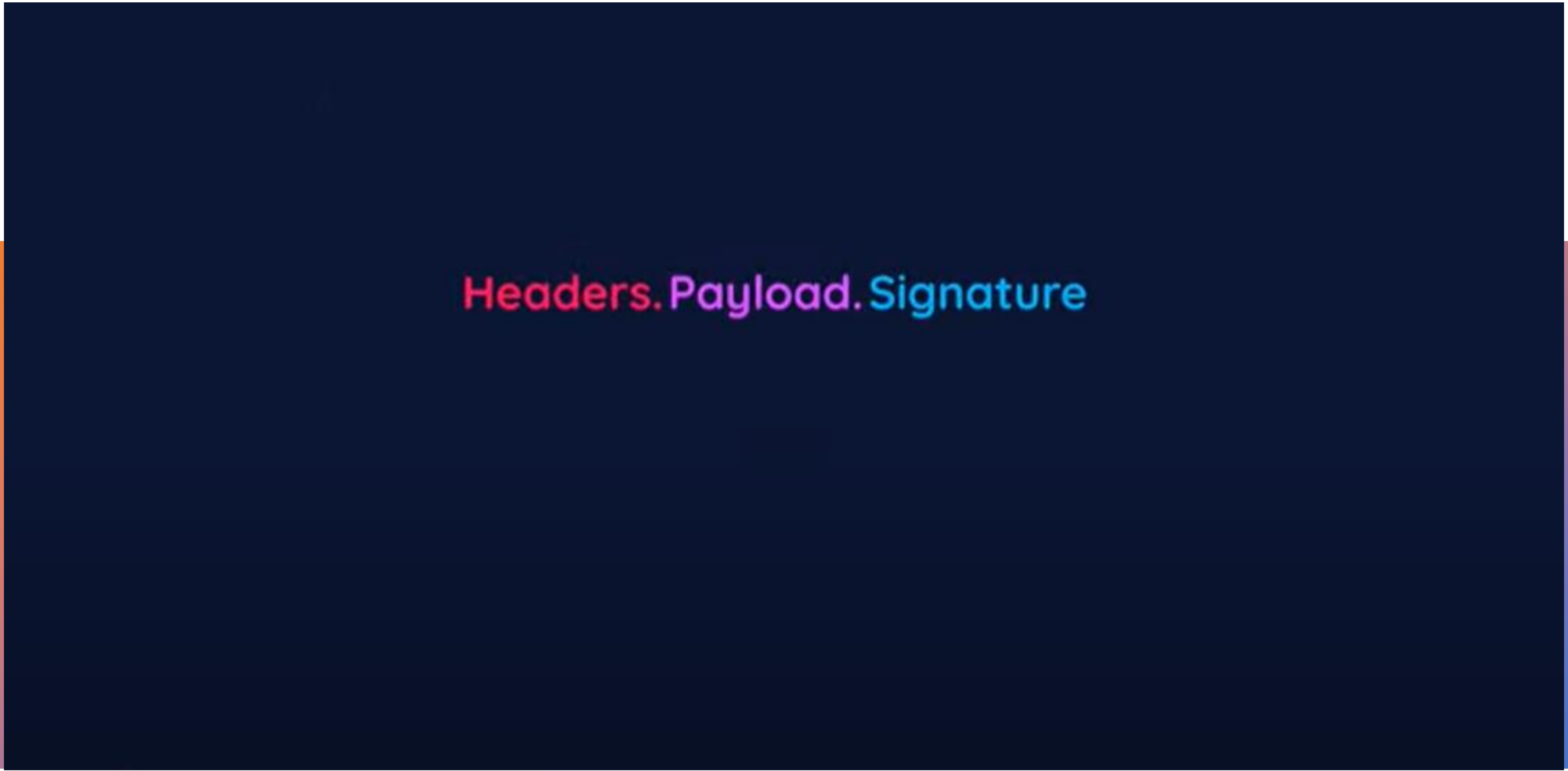
Signature

Makes the token secure (like a stamp of authenticity)

JWT Signing

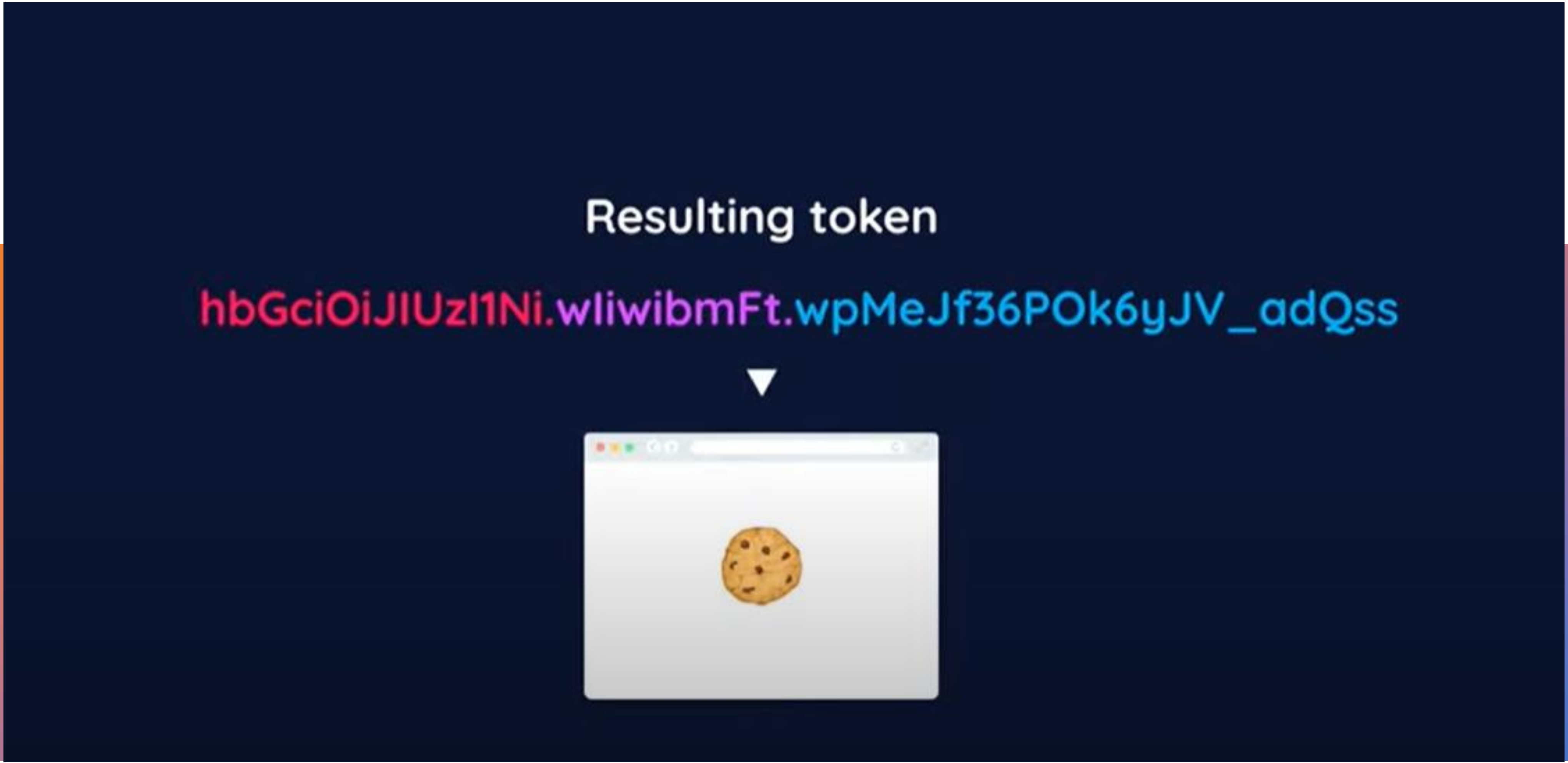

Headers
Payload
hashed together
Secret
'secure secret string'
Signature

JWT Signing



Headers.Payload.Signature

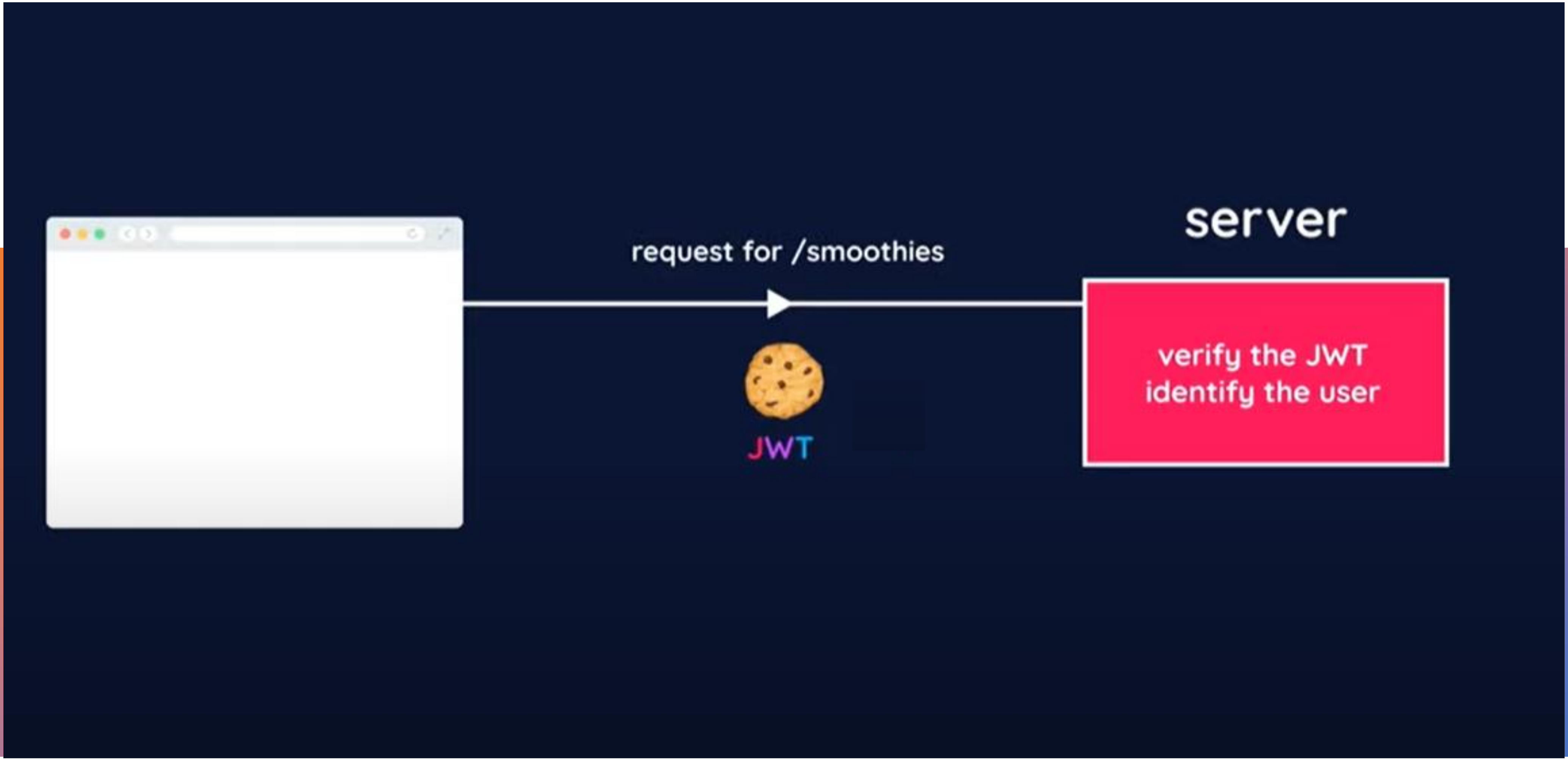
JWT Signing



Resulting token

hbGciOiJIUzI1Ni.wIiwibmFt.wpMeJf36POk6yJV_adQss

JSON Web Tocken



server
request for /smoothies
JWT
verify the JWT
identify the user