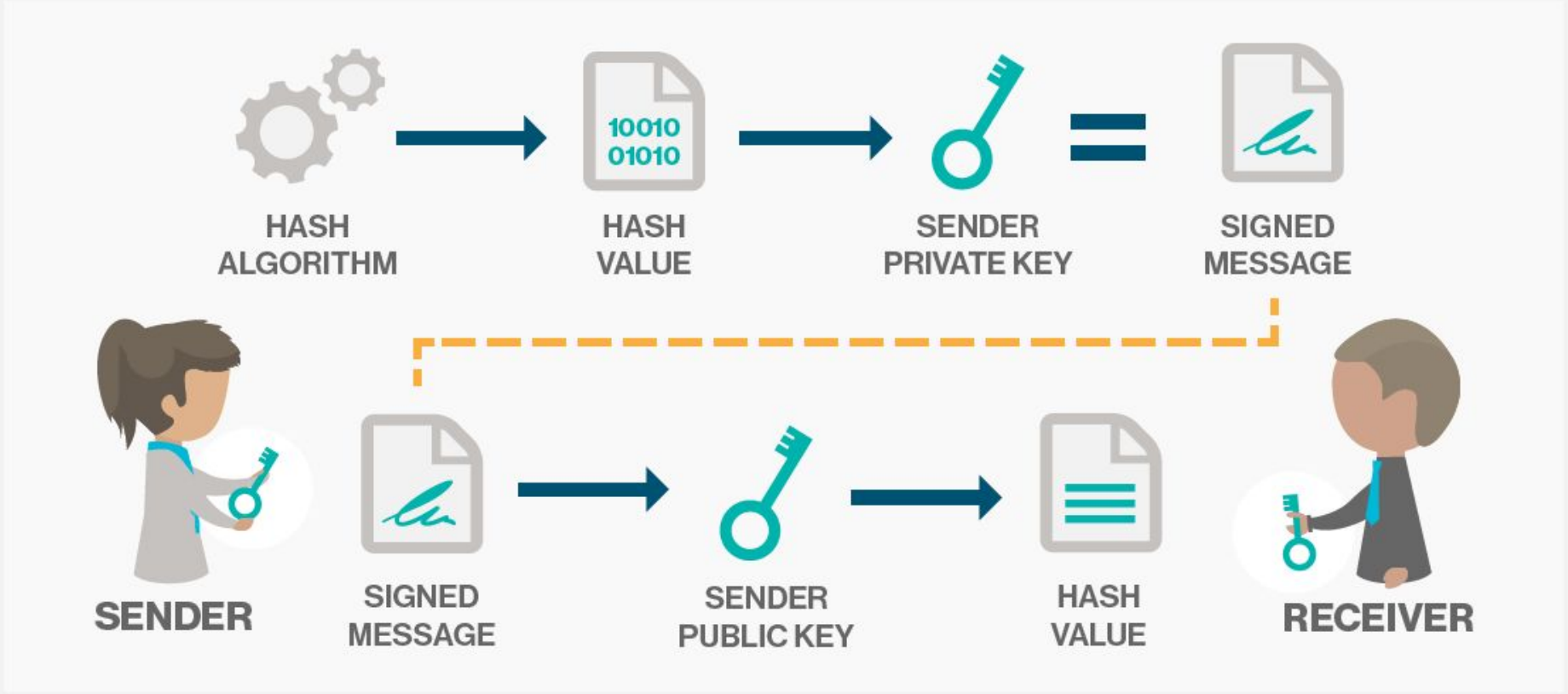
JSON WEB TOKEN

JWT ?

It ships information
that can be verified
and trusted
with a digital signature

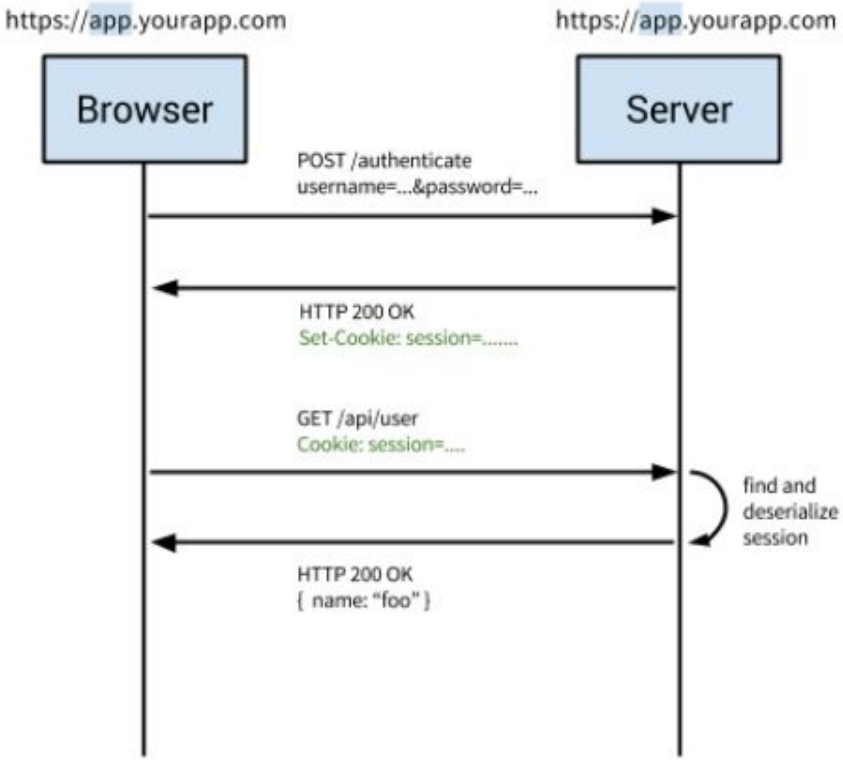
DEFINITION
DIGITAL SIGNATURE



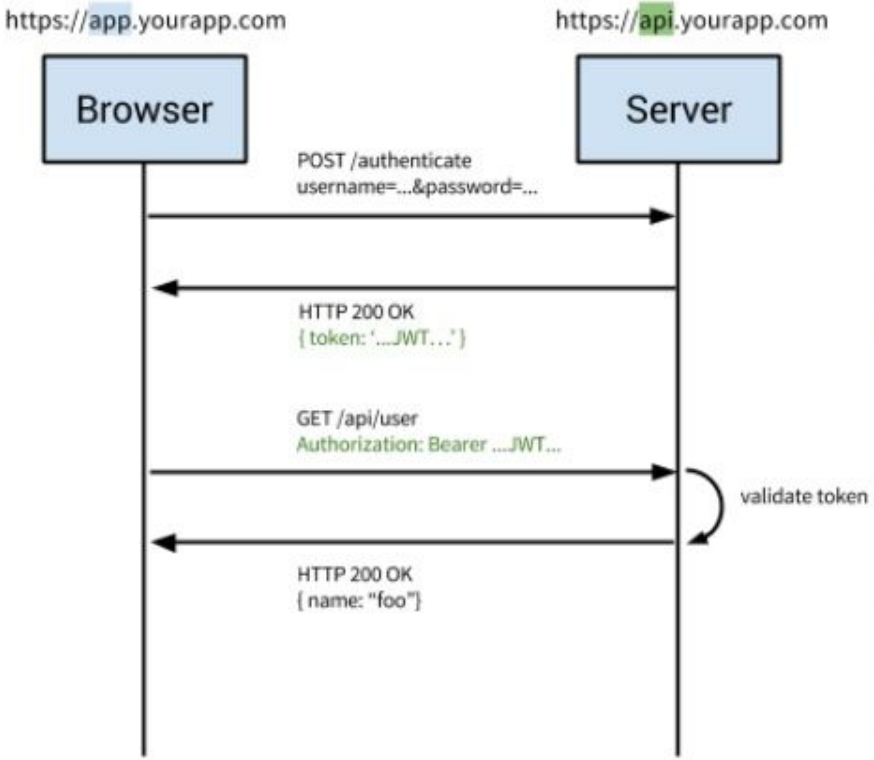
Stateless

JWT allow the server to verify the information contained in the JWT without necessarily storing state on the server.

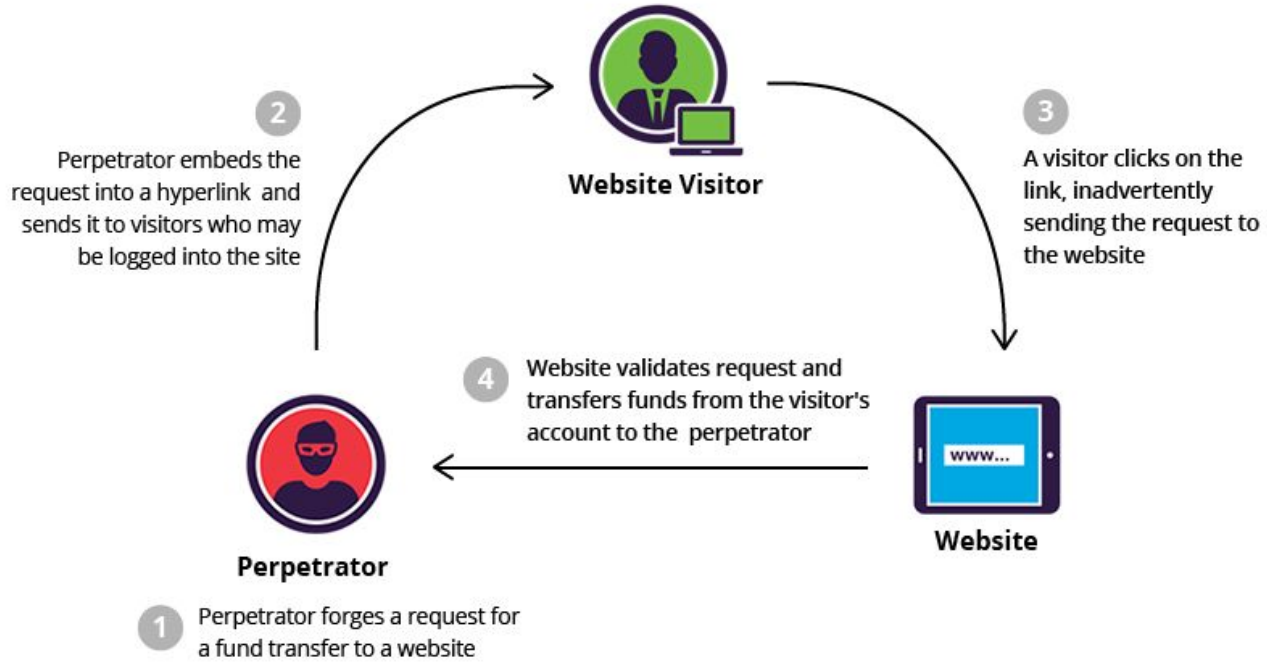
Traditional Cookie-Based Auth



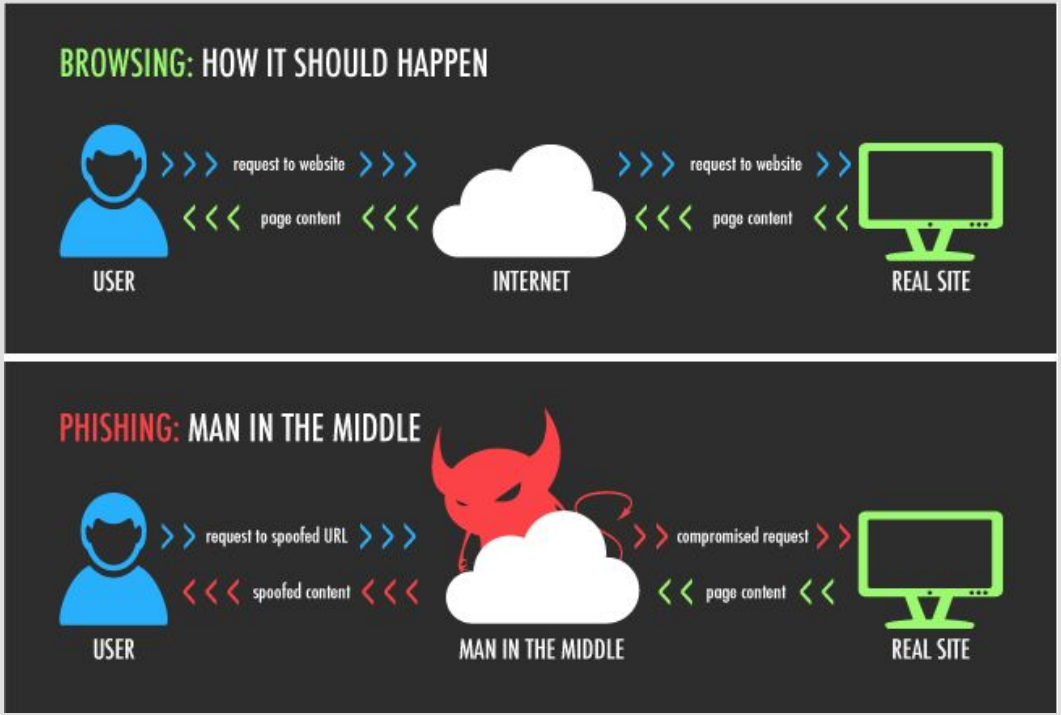
Modern Token-Based Auth



NO NEED PROTECT AGAINST CSRF



AVOID MAN-IN-THE-MIDDLE



JSON WEB TOKEN

Header

Parts of The header :

- declaring the type, which is JWT
- the hashing algorithm to use

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

Common JWT Signing Algorithms

HS256	HMAC using SHA-256
RS256	RSASSA-PKCS1-v1_5 using SHA-256
ES256	ECDSA using P-256 and SHA-256

Payload

Carry the information that we want to transmit, also called the JWT Claims.


```
{
  "iss": "scotch.io",
  "exp": 1300819380,
  "name": "Chris Sevilleja",
  "admin": true
}
```

Reserved Claims

- **iss**
  The issuer of the token.
- **sub**
  The subject of the token.
- **aud**
  The audience of the token.
- **exp**
  This will define the expiration.
- **nbf**
  the time before which the JWT MUST NOT be accepted.
- **iat**
  The time the JWT was issued.
- **jti**
  Unique identifier for the JWT.

https://tools.ietf.org/html/rfc7519#section-3

Signature

Made up of a hash of the following components:

- the header
- the payload
- secret

```
var encodedString =
base64UrlEncode(header) + "." +
base64UrlEncode(payload);
```

```
HMACSHA256(encodedString,'secret');
```

Full JSON of JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. HEADER

eyJpc3MiOiJzY290Y2guaW8iLCJleHAiOjEz
MDA4MTkzODAsIm5hbWUiOiJDaHJpcyB CLAIMS
TZXZpbGxlamEiLCJhZG1pbiI6dHJ1ZX0.

03f329983b86f7d9a9f5fef85305880101d5e302 SIGNATURE
afafa20154d094b229f757