Token vs. Cookies

JWT – the silver bullet for authentication

in modern application stacks?

Markus Schlichting

About

Markus Schlichting

Senior Software Engineer

Basel, Switzerland

Hackergarten Basel

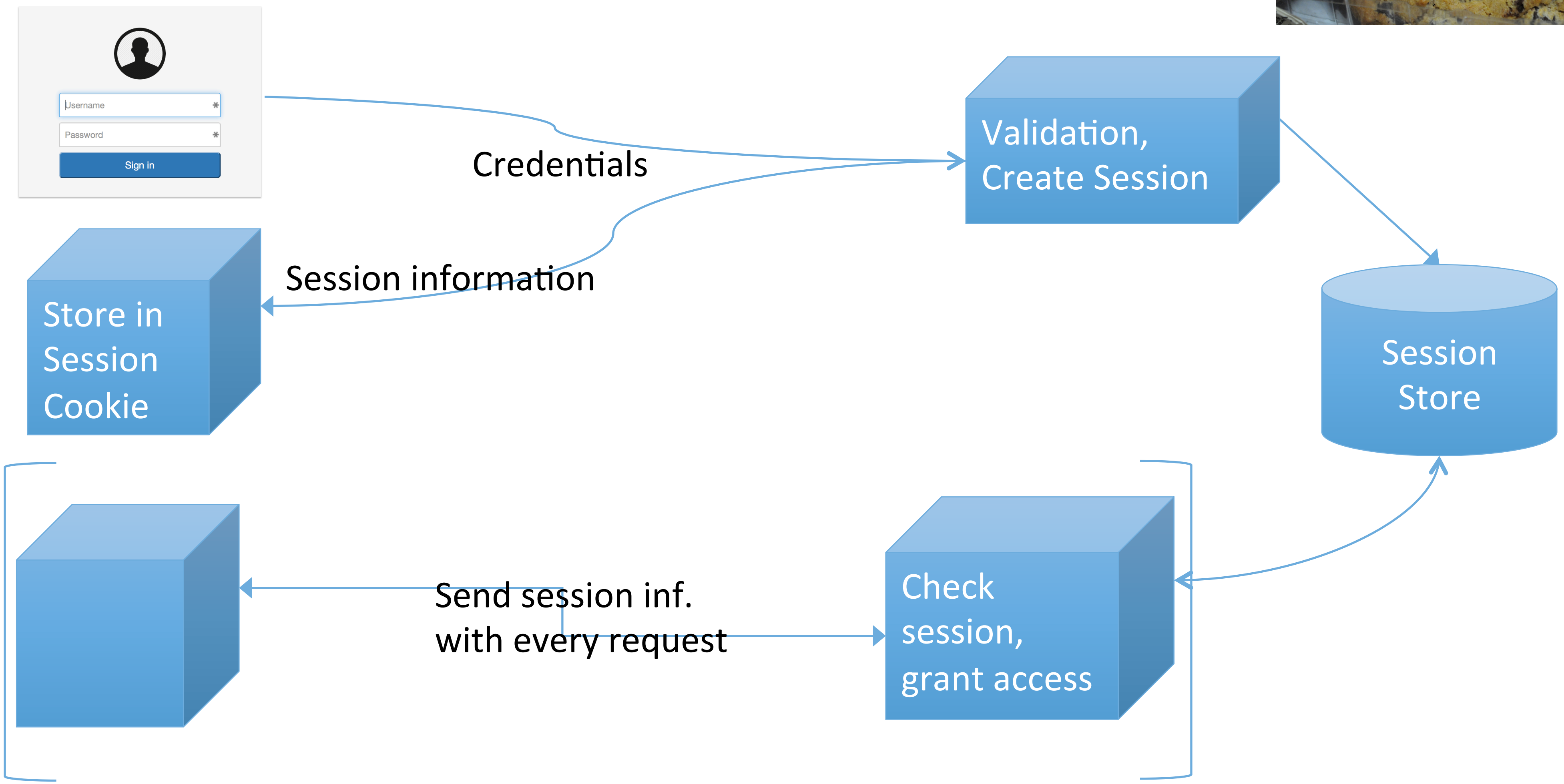
markus.schlichting@canoo.com

@madmas

Cookies & Sessions

https://app.yoursite.ma

https://app.yoursite.ma



Credentials

Validation, Create Session

Session information

Store in Session Cookie

Session Store

Send session inf. with every request

Check session, grant access

Cookies & Sessions

- **load balancing** requires **shared session pool**
- **separate services** need to **sync** via session pool
- cross origin resource sharing (CORS)
- CSRF vulnerabilities
- other clients than browsers?

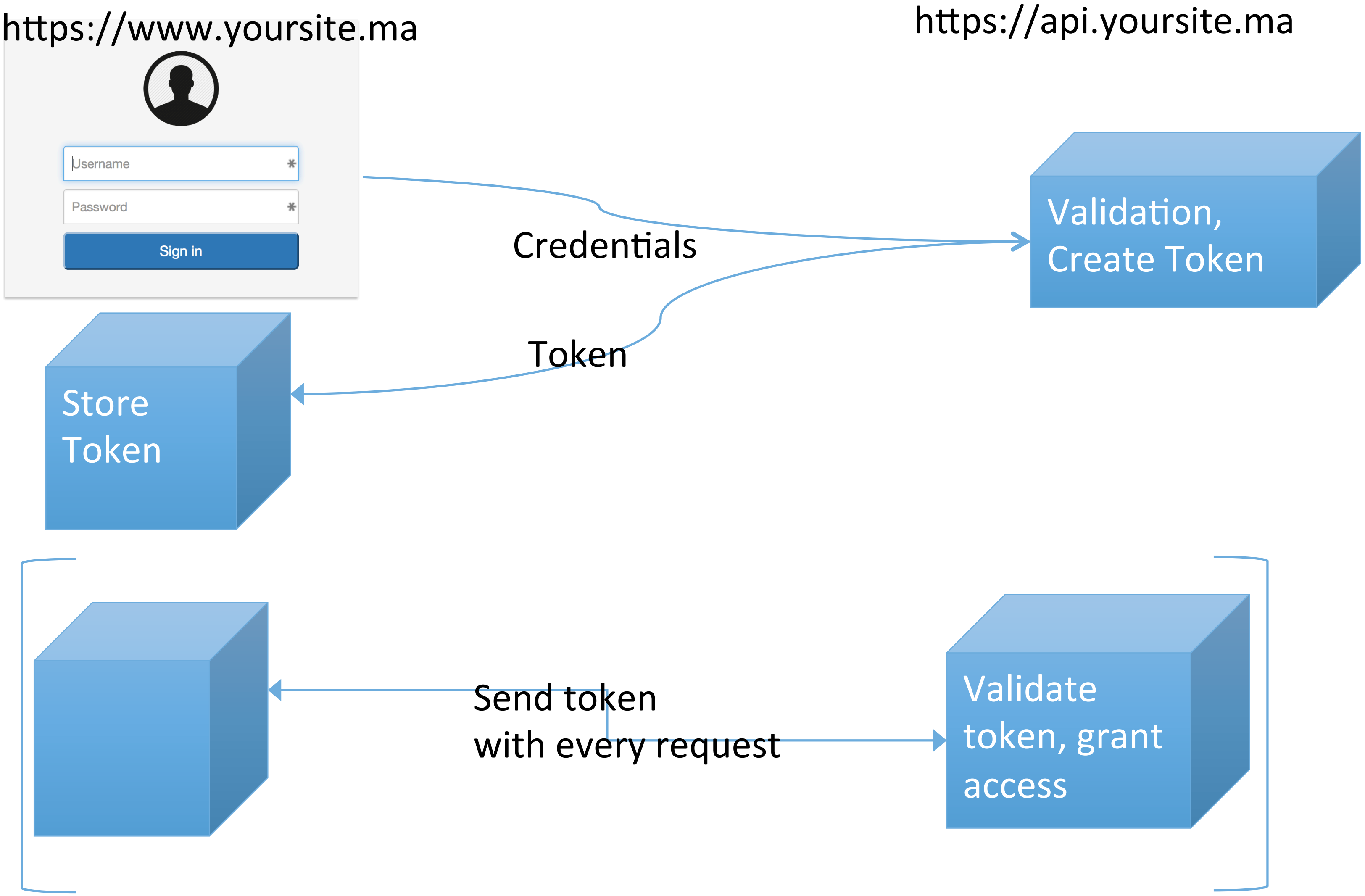
JSON Web Token

JSON Web Tokens are an open, industry standard (RFC 7519) method for representing claims securely between two parties.

- relies on other JSON-based standards:
  - JWS (JSON Web Signature)
  - JWE (JSON Web Encryption)
- Libraries widely available

…

JWT – How?



https://www.yoursite.ma

https://api.yoursite.ma

Credentials

Validation,
Create Token

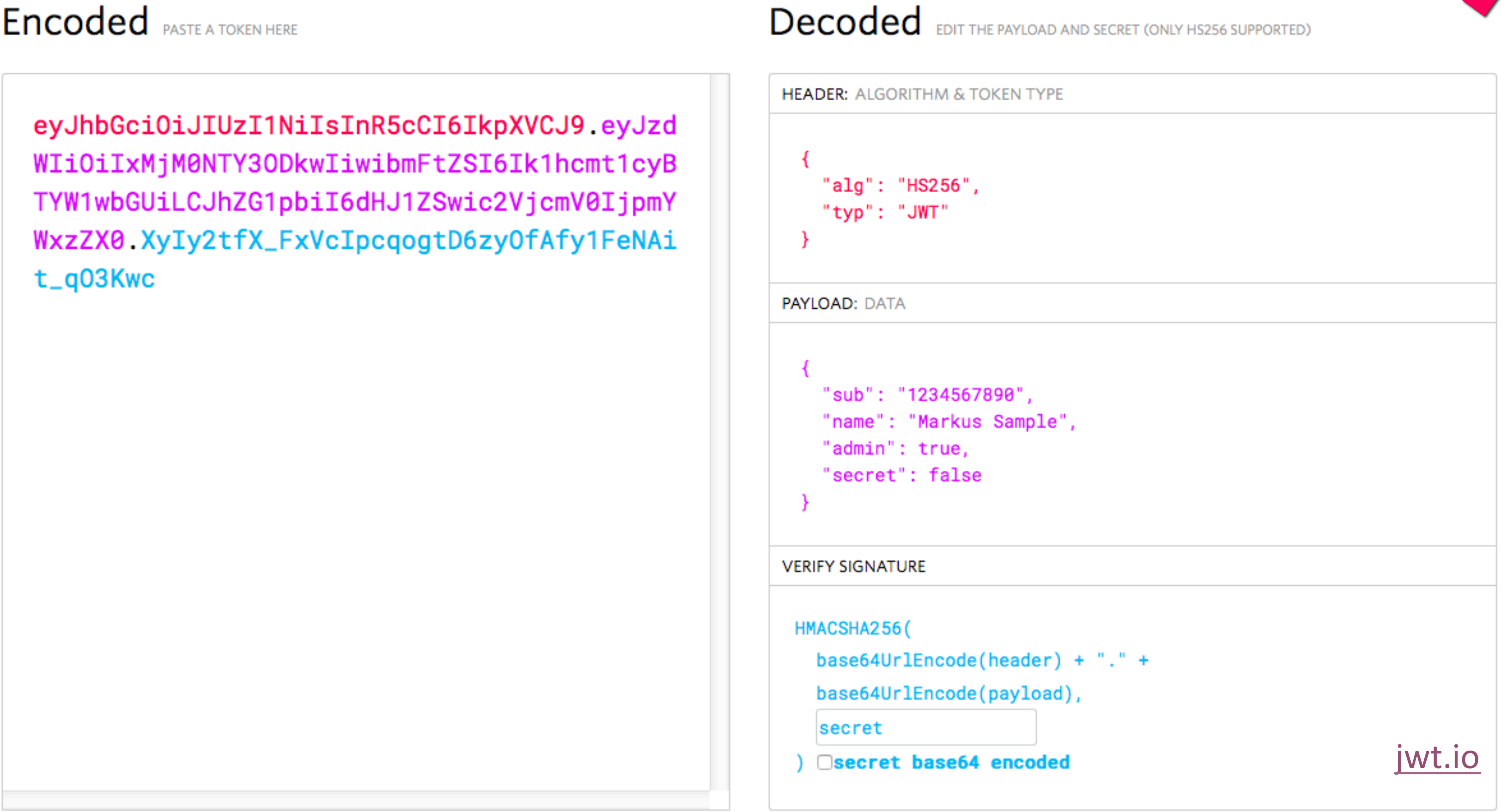
Token

Store
Token

Send token
with every request

Validate
token, grant
access

JWT – What’s inside?



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzd
WIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ik1hcmt1cyB
TYW1wbGUiLCJhZG1pbiI6dHJ1ZSwic2VjcmV0IjpmY
WxzZX0.XyIy2tfX_FxVcIpcqogtD6zyOfAfy1FeNAi
t_qO3Kwc
```

Decoded EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "Markus Sample",
  "admin": true,
  "secret": false
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) □secret base64 encoded
```

jwt.io

JWT in action

Demo time!

JWT security aspects

- use on encrypted connection only (HTTP**S**!)

- avoid URL tokens

  https://yoursite.ma/service/action?token=jwt.goes.here

- in securing Session Cookies a lot of effort has been made
  - HttpOnly, etc
  - be aware of the implications coming with tokens

JWT summary

- embraces **JSON**, heavily adopted across many stacks
- **simple** to use, **simple** to implement
  - more libs, fewer interoperability issues
- supports both **symmetric** and **asymmetric crypto**
  - majority of use cases solved
- **reduce** the **dependency** between services to a minimum
  - shared secret, public/private keys
- help to achieve *one basic principle* in REST based architecture: **State** *transfer*

Conclusion

- Cookies are not completely overdue, but JWT provide a **lot of benefits**!
- JWT for **scalability** and **flexibility**
- Very useful to provide a **cross plattform** API
- **ServiceWorkers** to ease up handling within the browser

Thank you!

Markus Schlichting

Senior Software Engineer

Basel, Switzerland

Hackergarten Basel

markus.schlichting@canoo.com

@madmas

canoo

[delivering end-user happiness]

[HACKERGARTEN

A COMPUTER PROGRAMMING CONTRIBUTOR GROUP]

Resources

- RFC 7519 - JSON Web Token (JWT)
- Dwyl/learn-json-web-tokens
- Auth0: 10 Things You Should Know about Tokens
- Does JWT put you webapp at risk?
- Make your REST services attack proof – Alex Soto Bueno