



Markus Schlichting

Senior Software Engineer

canoo

[delivering end-user happiness]

Basel, Schweiz



Hackergarten Basel


[HACKERGARTEN]
A COMPUTER PROGRAMMING CONTRIBUTOR GROUP

markus.schlichting@canoo.com



@madmas

Sessions & Cookies



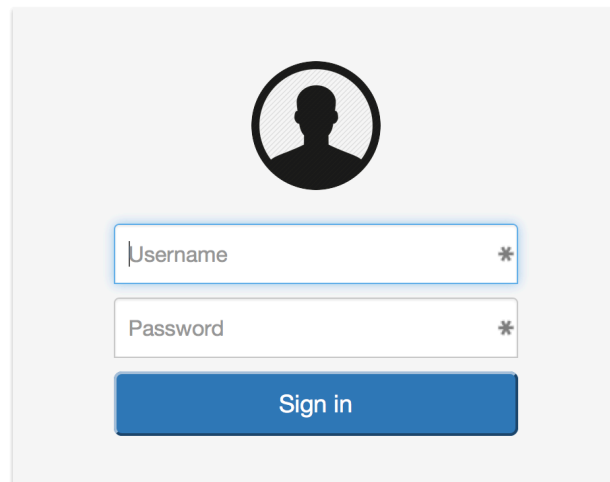
A circular icon containing a black silhouette of a person's head and shoulders, representing a user profile.

Username *

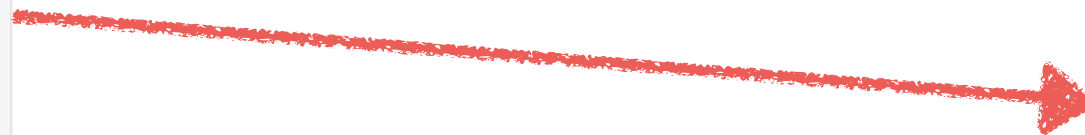
Password *

Sign in

Sessions & Cookies

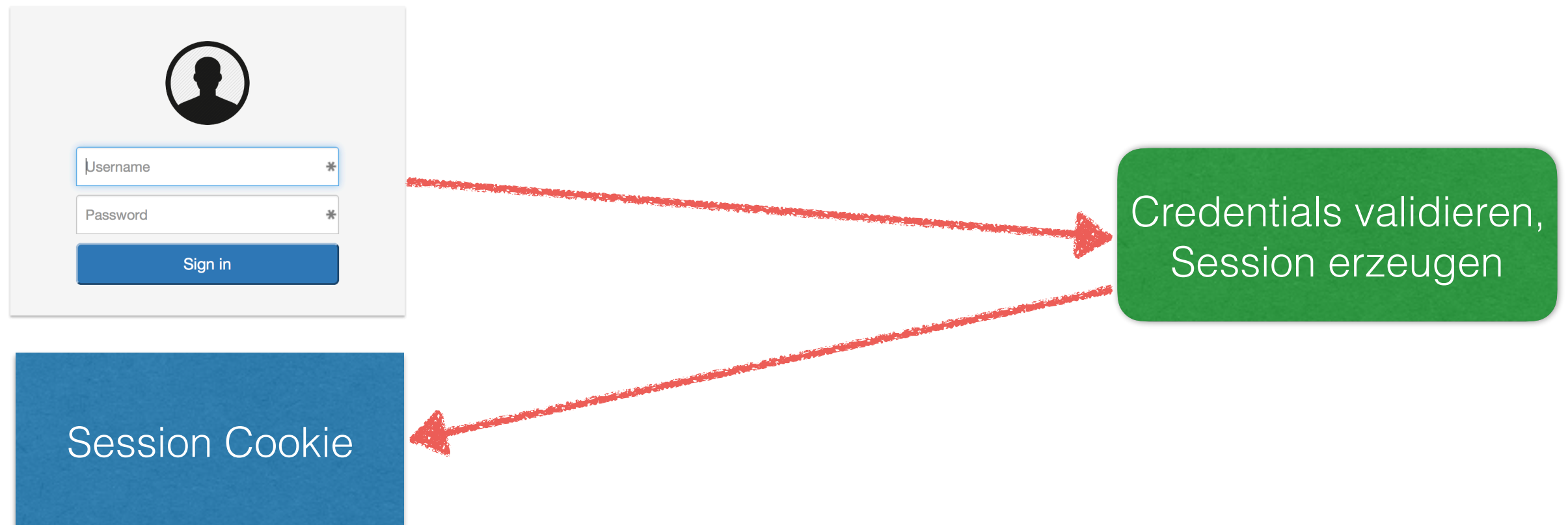


A user login form with a light gray background. At the top is a circular icon containing a black silhouette of a person. Below the icon are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both fields have a small asterisk icon on the right side. Below the password field is a blue button with the text 'Sign in' in white.

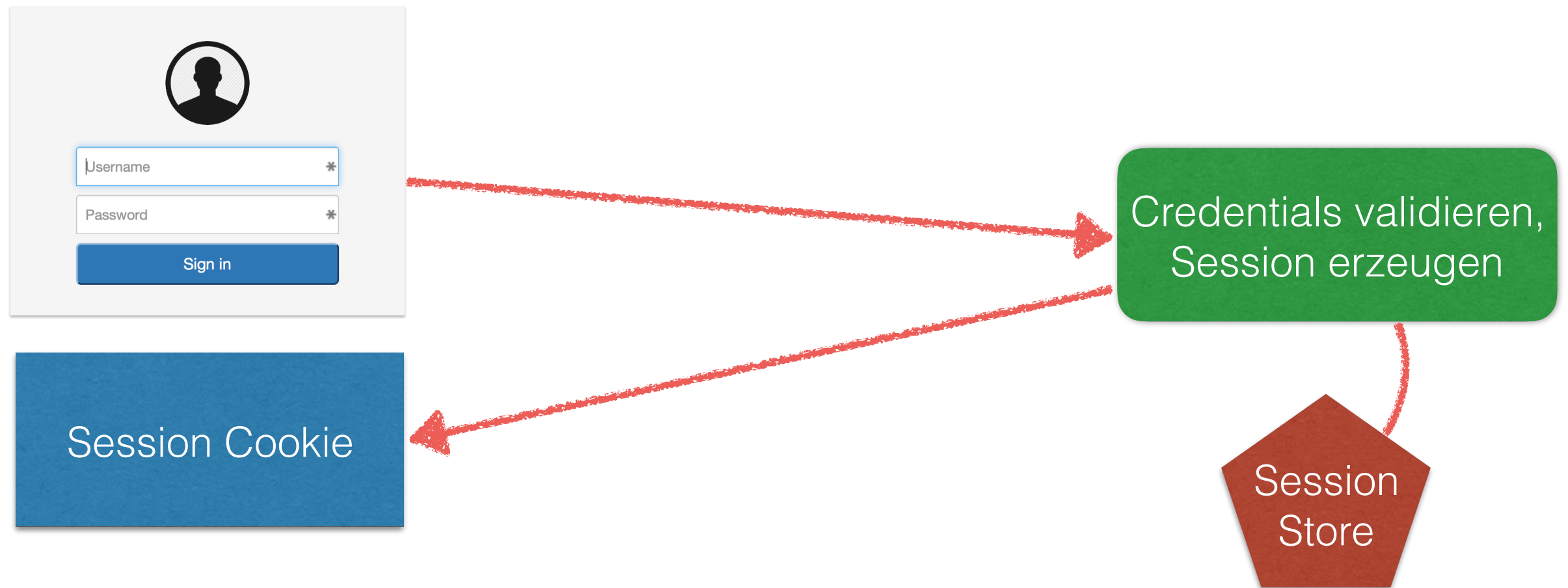


Credentials validieren,
Session erzeugen

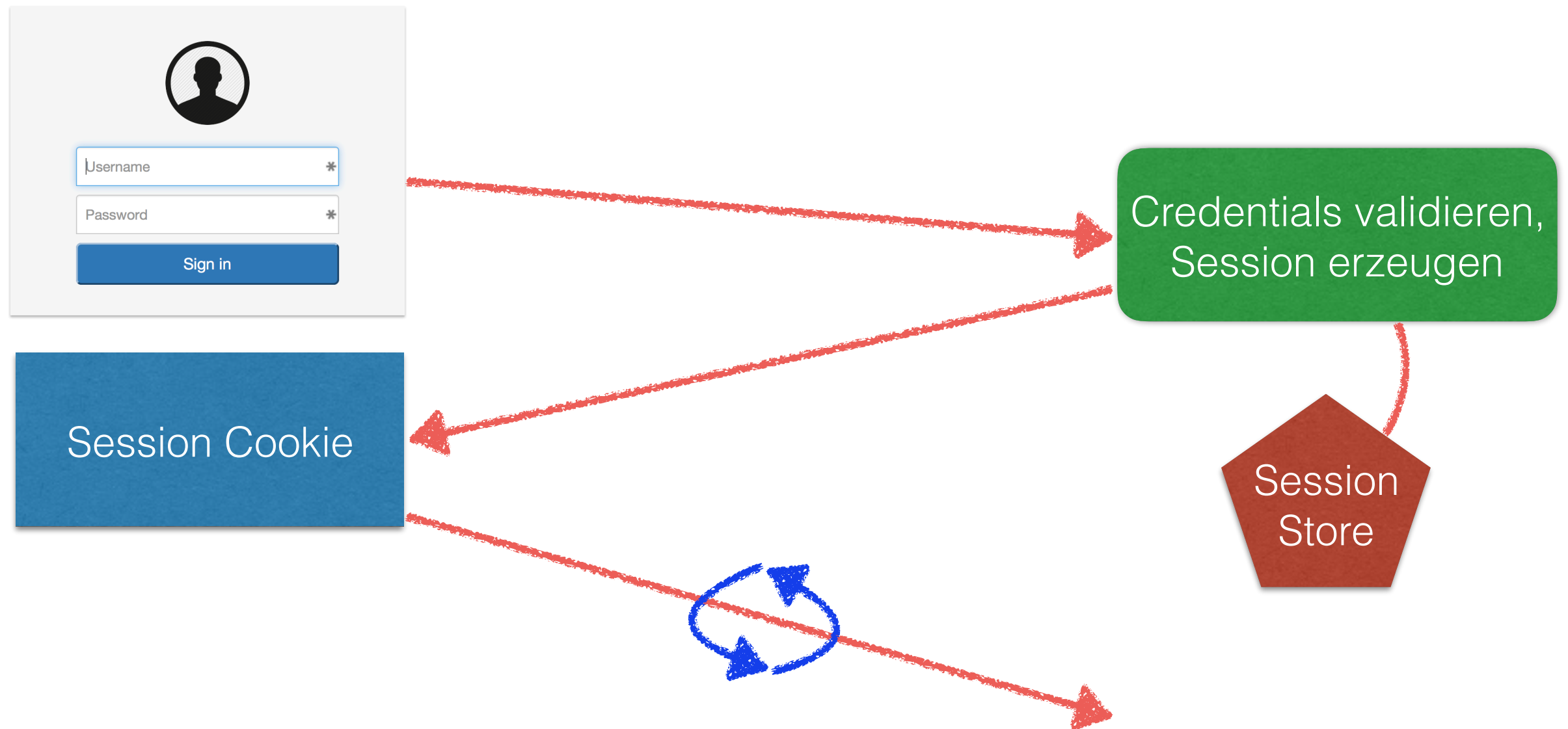
Sessions & Cookies



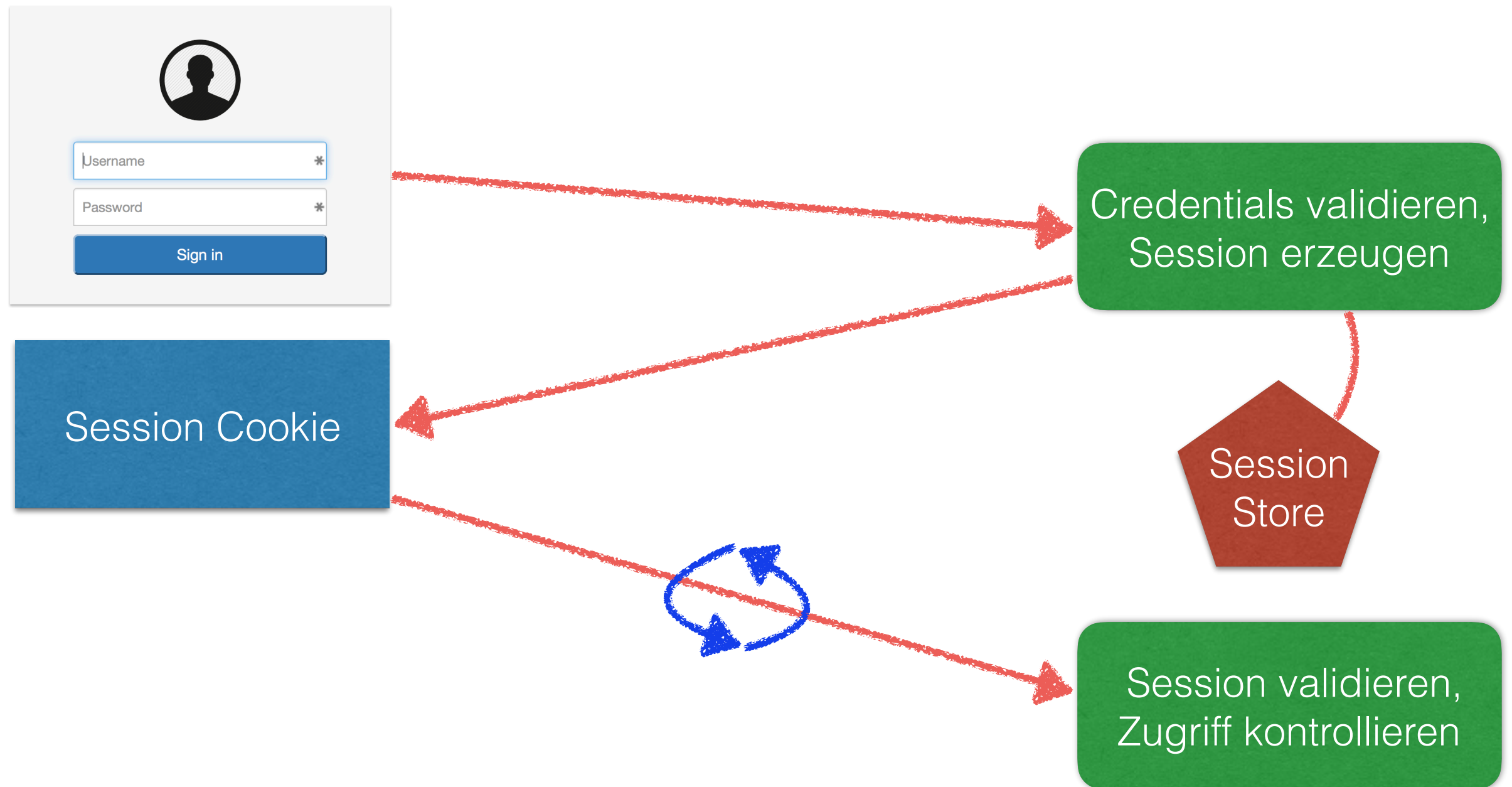
Sessions & Cookies



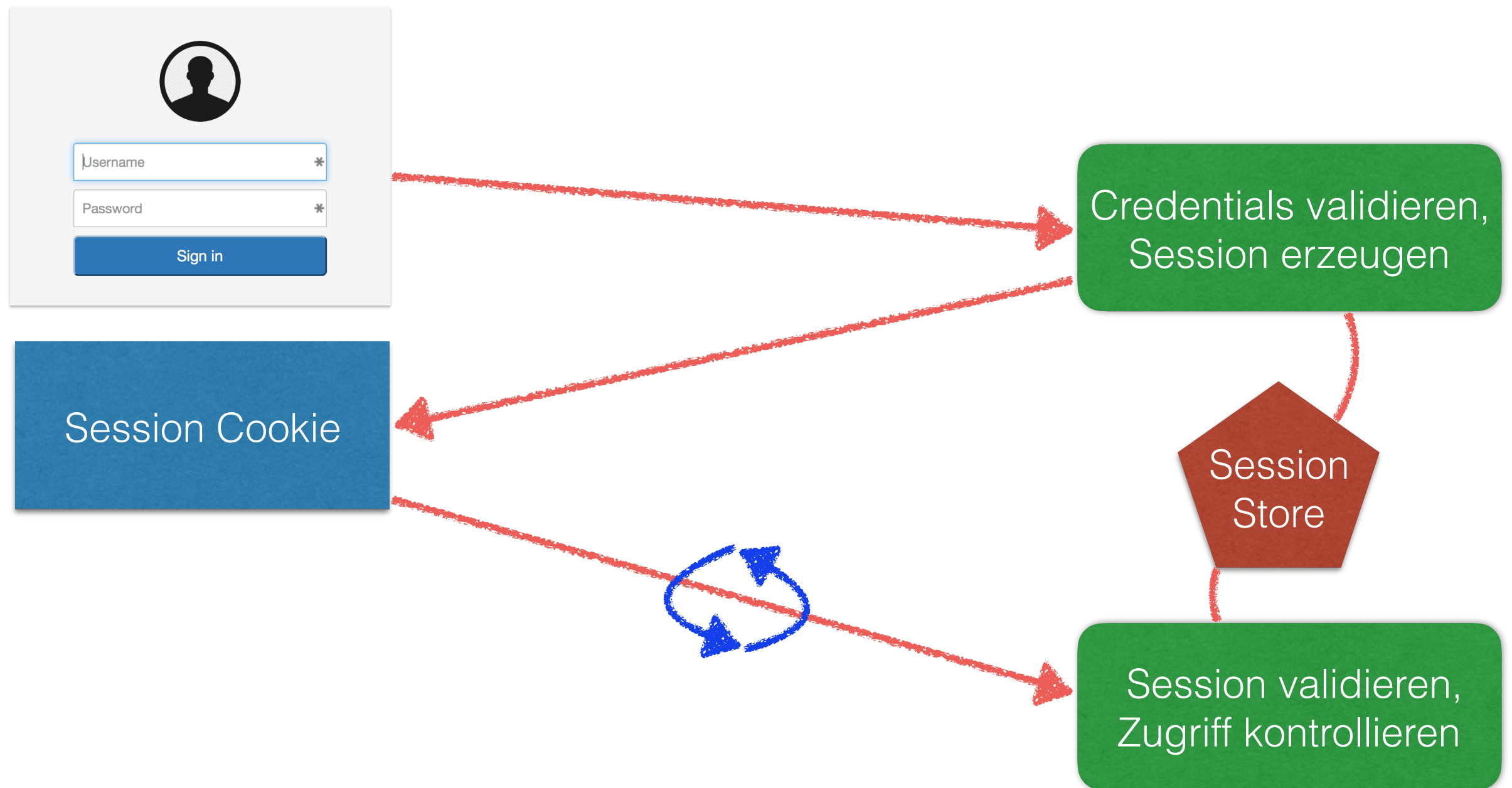
Sessions & Cookies



Sessions & Cookies

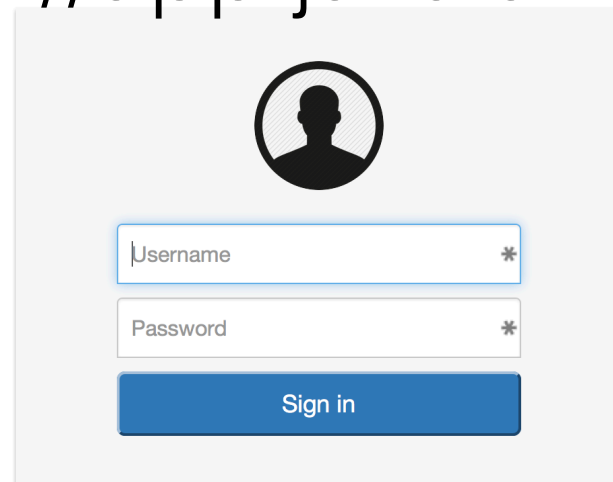


Sessions & Cookies



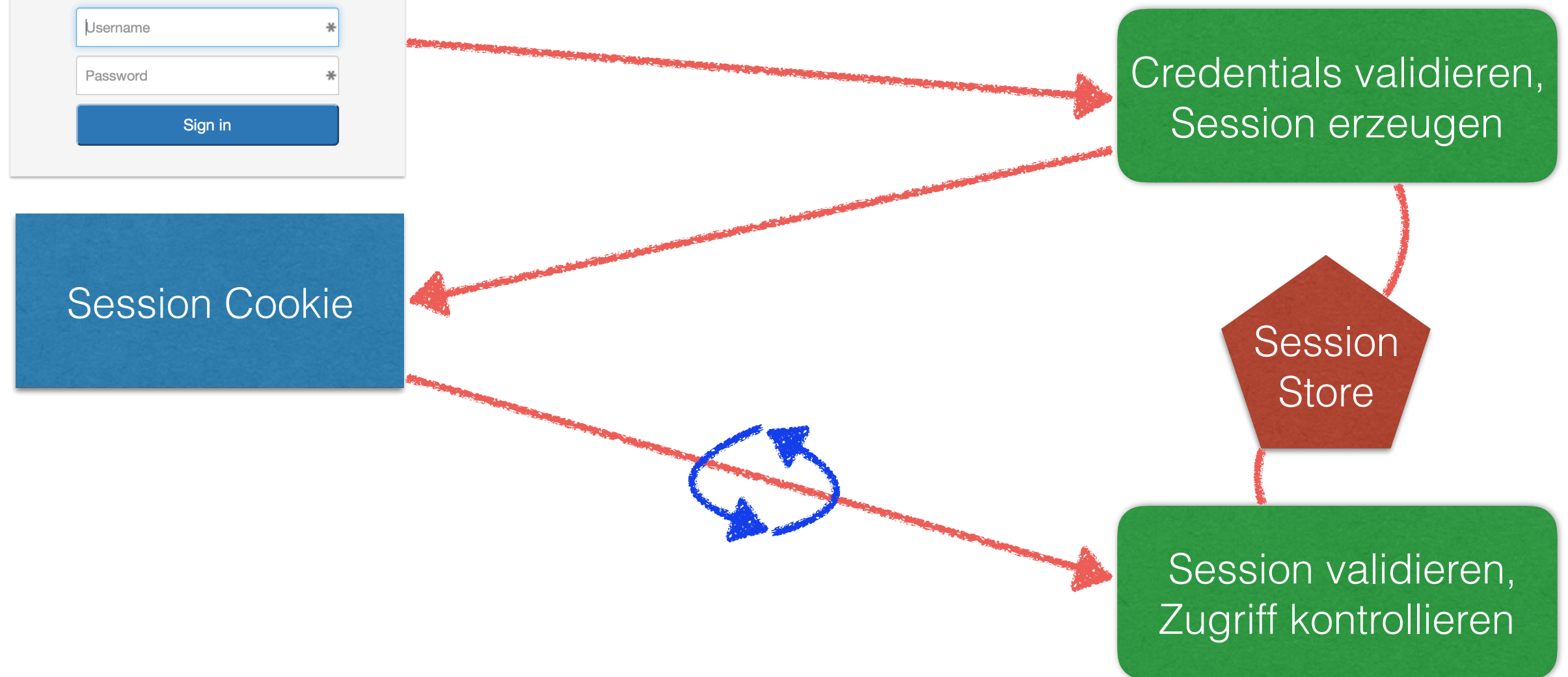
Sessions & Cookies

http://app.javaland.de



A login form with a user icon, a 'Username' input field with an asterisk, a 'Password' input field with an asterisk, and a 'Sign in' button.

http://app.javaland.de



Sessions & Cookies

Sessions & Cookies

Loadbalancing benötigt **geteilten Sessionpool**

Sessions & Cookies

Loadbalancing benötigt **geteilten Sessionpool**

Services werden **gekoppelt**

Sessions & Cookies

Loadbalancing benötigt **geteilten Sessionpool**

Services werden **gekoppelt**

Cross Domain Authentifizierung: **CORS**

Sessions & Cookies

Loadbalancing benötigt **geteilten Sessionpool**

Services werden **gekoppelt**

Cross Domain Authentifizierung: **CORS**

CSRF Verwundbarkeit

Sessions & Cookies

Loadbalancing benötigt **geteilten Sessionpool**

Services werden **gekoppelt**

Cross Domain Authentifizierung: **CORS**

CSRF Verwundbarkeit

Andere Clients ausser Browser?

JSON Web Token



JSON Web Token

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

(RFC 7519)



JSON Web Token

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

(RFC 7519)

The suggested pronunciation of JWT is the same as the English word "jot".



JSON Web Token

basierend auf anderen JSON-Standards:

JWS (JSON Web **Signature**)

JWE (JSON Web **Encryption**)



JSON Web Token

basierend auf anderen JSON-Standards:

JWS (JSON Web **Signature**)

JWE (JSON Web **Encryption**)


Bibliotheken für..



...



JWT in Action




Username *

Password *

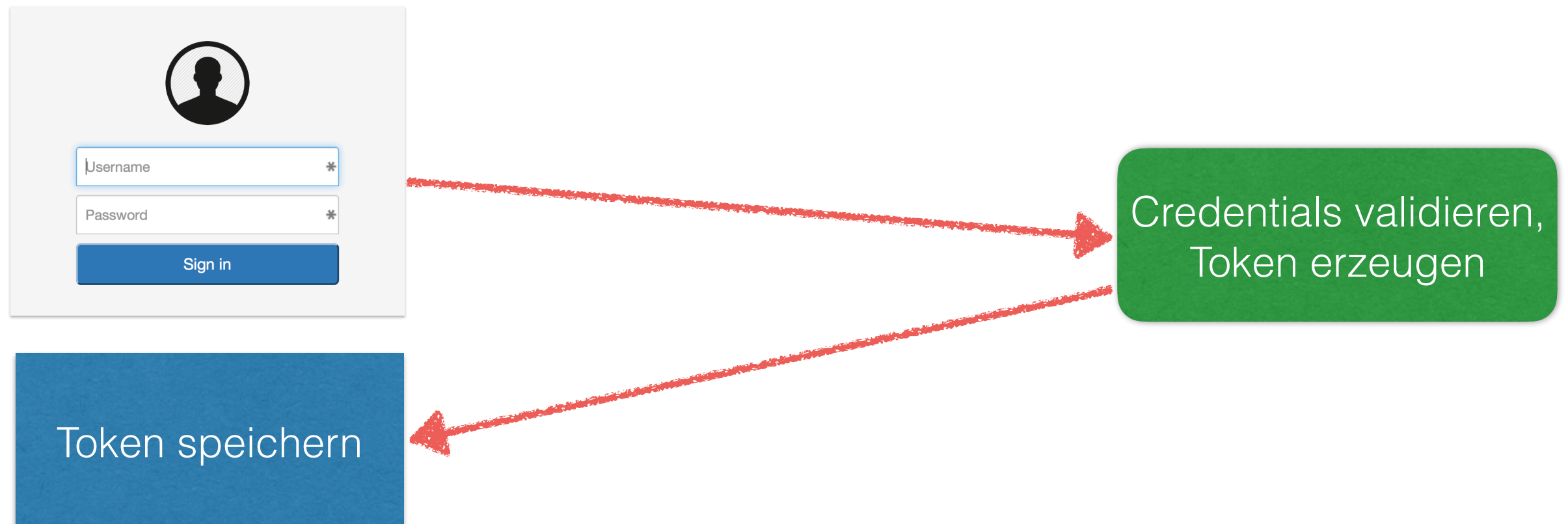
Sign in

JWT in Action

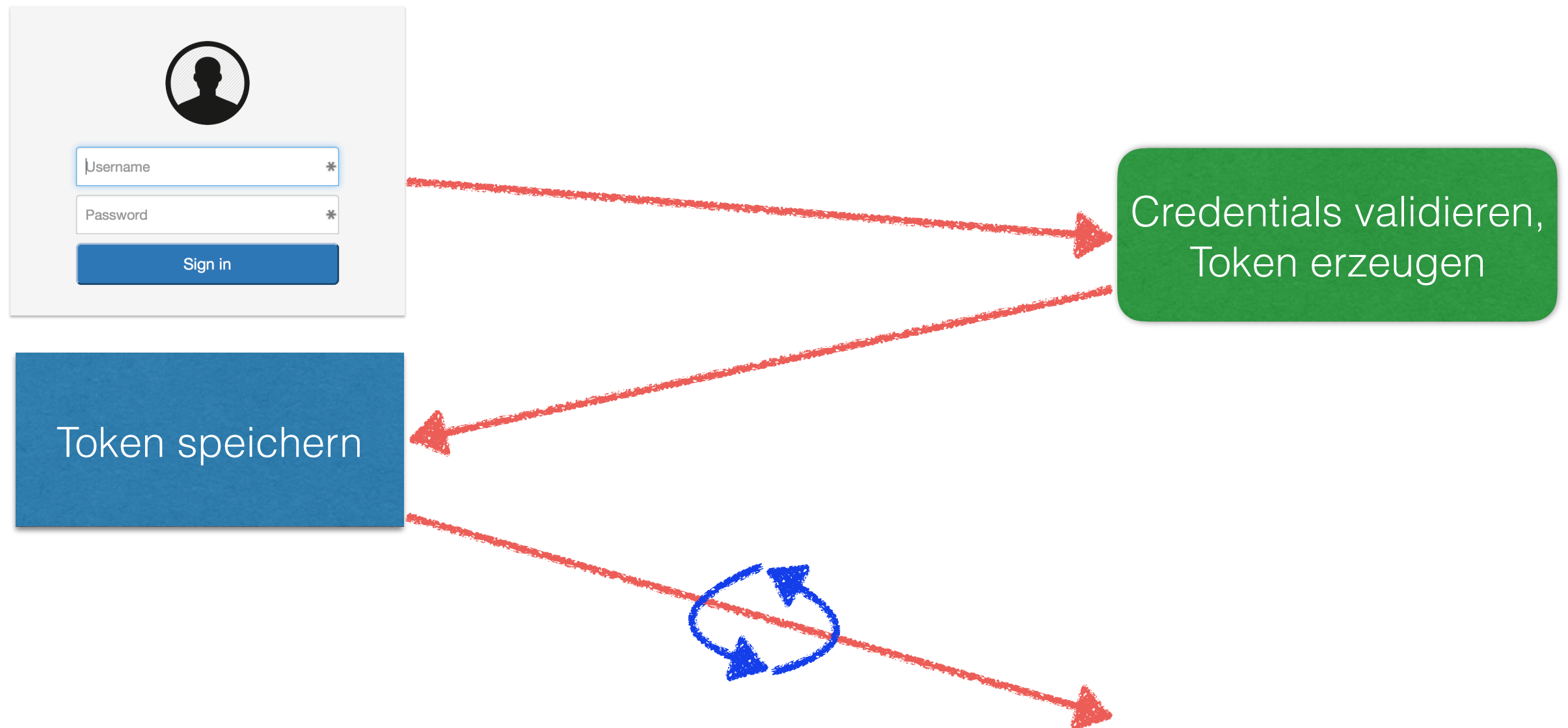

 *
 *


Credentials validieren,
Token erzeugen

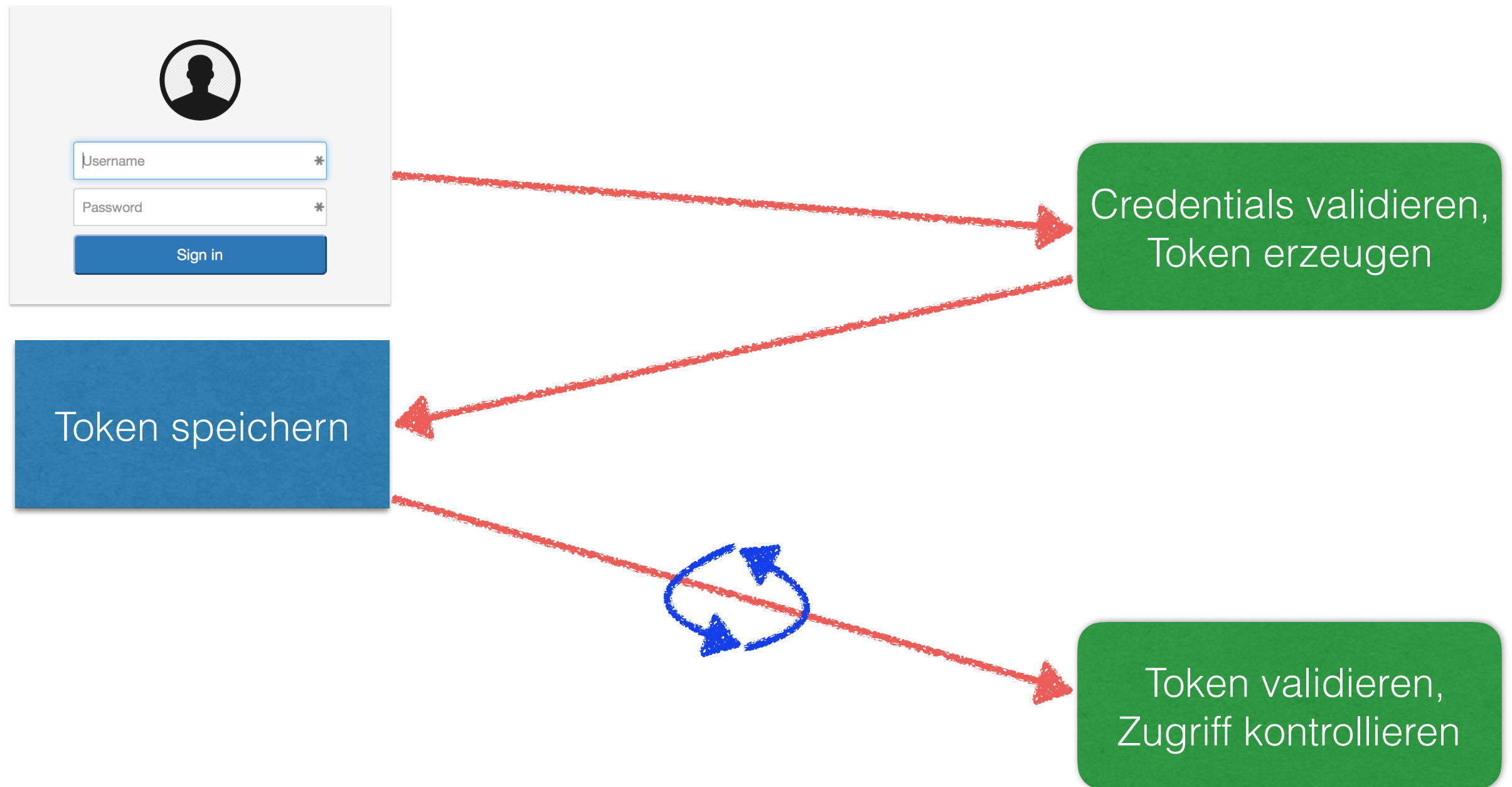
JWT in Action



JWT in Action



JWT in Action



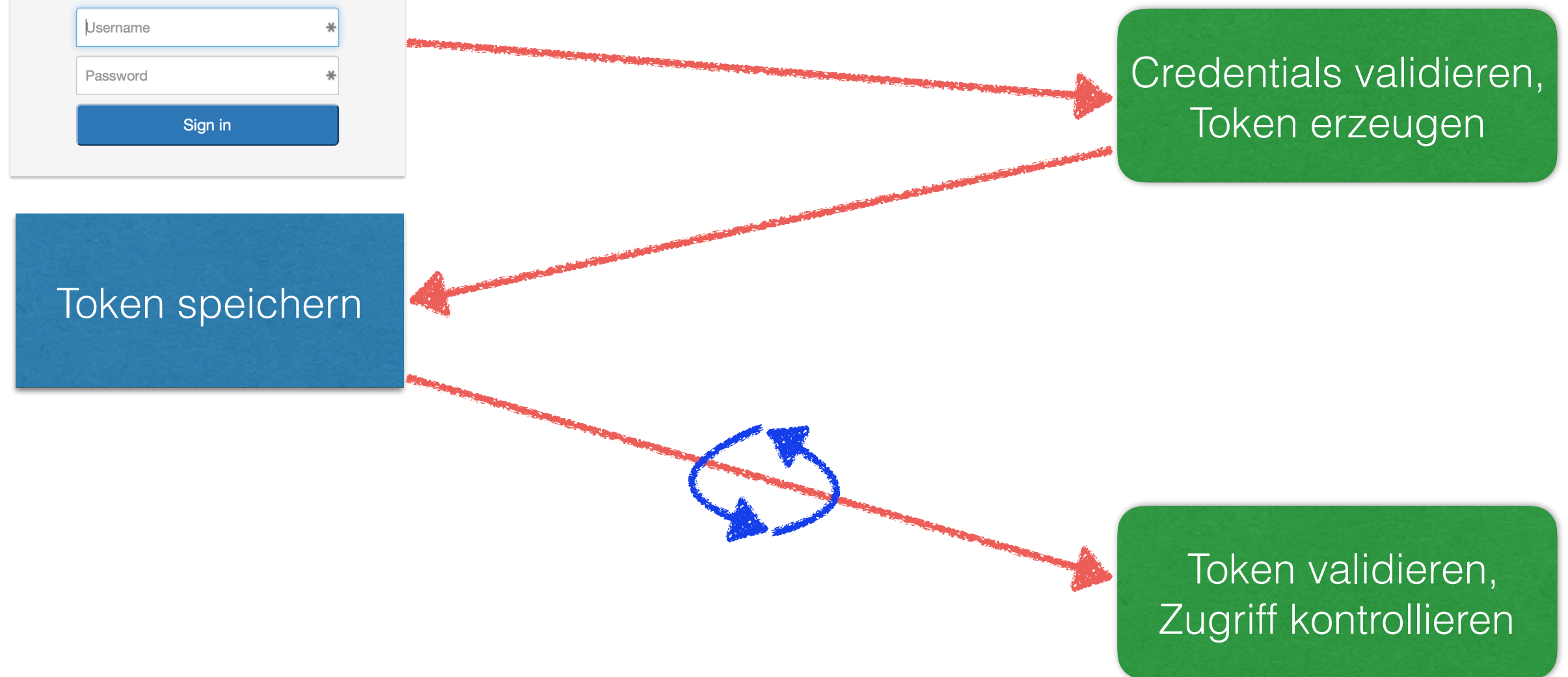
JWT in Action



<http://app.javaland.de>

A user login form with a grey background. At the top is a circular icon of a person's silhouette. Below it are two input fields: 'Username' and 'Password', both with asterisks indicating required fields. At the bottom is a blue button labeled 'Sign in'.

<http://api.javaland.de>



JWT - inside

Encoded

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikhcmt1cyBTYW1wbGUuLCJhZG1pbSI6dHJ1ZSwic2VjcmV0IjpmYXxzZX0.XyIy2tfX_FxVcIpcqogtD6zy0fAfy1FeNAit_q03Kwc

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 practice

Demo!

JWT Security Aspekte

JWT Security Aspekte

immer verschlüsselt Kommunizieren (HTTPS!)

JWT Security Aspekte

immer verschlüsselt Kommunizieren (HTTPS!)

URL Token vermeiden

<https://yourpage.de/service/action?token=jwt.goes.here>

JWT Security Aspekte

immer verschlüsselt Kommunizieren (HTTPS!)

URL Token vermeiden

<https://yourpage.de/service/action?token=jwt.goes.here>

Token Handling: Invalidation, Reset

JWT Überblick



JWT Überblick



basiert auf **JSON**

JWT Überblick



basiert auf **JSON**

einfach zu nutzen, **einfach** zu implementieren

JWT Überblick



basiert auf **JSON**

einfach zu nutzen, **einfach** zu implementieren

symmetrische und **asymmetrische** Crypto

JWT Überblick



basiert auf **JSON**

einfach zu nutzen, **einfach** zu implementieren

symmetrische und **asymmetrische** Crypto

reduziert Abhängigkeiten

JWT Überblick



basiert auf **JSON**

einfach zu nutzen, **einfach** zu implementieren

symmetrische und **asymmetrische** Crypto

reduziert Abhängigkeiten

Basisprinzip von REST: **State transfer**

Zusammenfassung



Zusammenfassung



Cookies nicht obsolet, **Token** bieten jedoch **viele Vorteile**

Zusammenfassung



Cookies nicht obsolet, **Token** bieten jedoch **viele Vorteile**

JWT für **Skalierbarkeit** und **Flexibilität**

Zusammenfassung



Cookies nicht obsolet, **Token** bieten jedoch **viele Vorteile**

JWT für **Skalierbarkeit** und **Flexibilität**

Cross Plattform

Zusammenfassung



Cookies nicht obsolet, **Token** bieten jedoch **viele Vorteile**

JWT für **Skalierbarkeit** und **Flexibilität**

Cross Plattform

Cookies oder **Token?**

Anforderungen und Implikationen abwägen!

Demo Sources und Slides:

<https://github.com/madmas/TokenVsCookies>