

Loeng 11

Spring Security
JSON Web Token
OAuth2.0

Kordamine

- Kuidas toimub Spring Mvc raamistiku laadimine?

Raamistiku laadimine

WebApplicationInitializer // onStartUp()

```
org.springframework.web.context.AbstractContextLoaderInitializer  
    org.springframework.web.servlet.support.AbstractDispatcherServletInitializer  
        org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer
```

Raamistiku laadimine

```
public class MyApplicationInitializer extends
    AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected String[] getServletMappings() {
        return new String[] { "/api/*" };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { MvcConfig.class };
    }

}
```

Kordamine (Http protokoll)

Http protokoll on olektuta

GET /index.html HTTP/1.1

GET /products.html HTTP/1.1

Cookie

päring

```
POST /login HTTP/1.1  
Host: localhost  
...  
user=alice&password=s3cret
```

vastus

```
HTTP/1.1 200 OK  
Set-Cookie: JSESSIONID=fd0j2u7hlav4bsstjubihvlta0; path=/  
...
```

järgnevad päringud

```
GET /contacts HTTP/1.1  
Host:localhost  
Cookie:JSESSIONID=fd0j2u7hlav4bsstjubihvlta0
```

Sessioon

- Võti väärtus paarid
- Serveri poolt hallatav

JSESSIONID=h07by9xdaz2772y3zi2llijo

Session

```
@GetMapping("/count")
public String counter(HttpSession session) {

    Object count = session.getAttribute("count");

    if (count instanceof Integer) {
        count = (Integer) count + 1;
    } else {
        count = 0;
    }

    session.setAttribute("count", count);

    return String.valueOf(count);
}
```

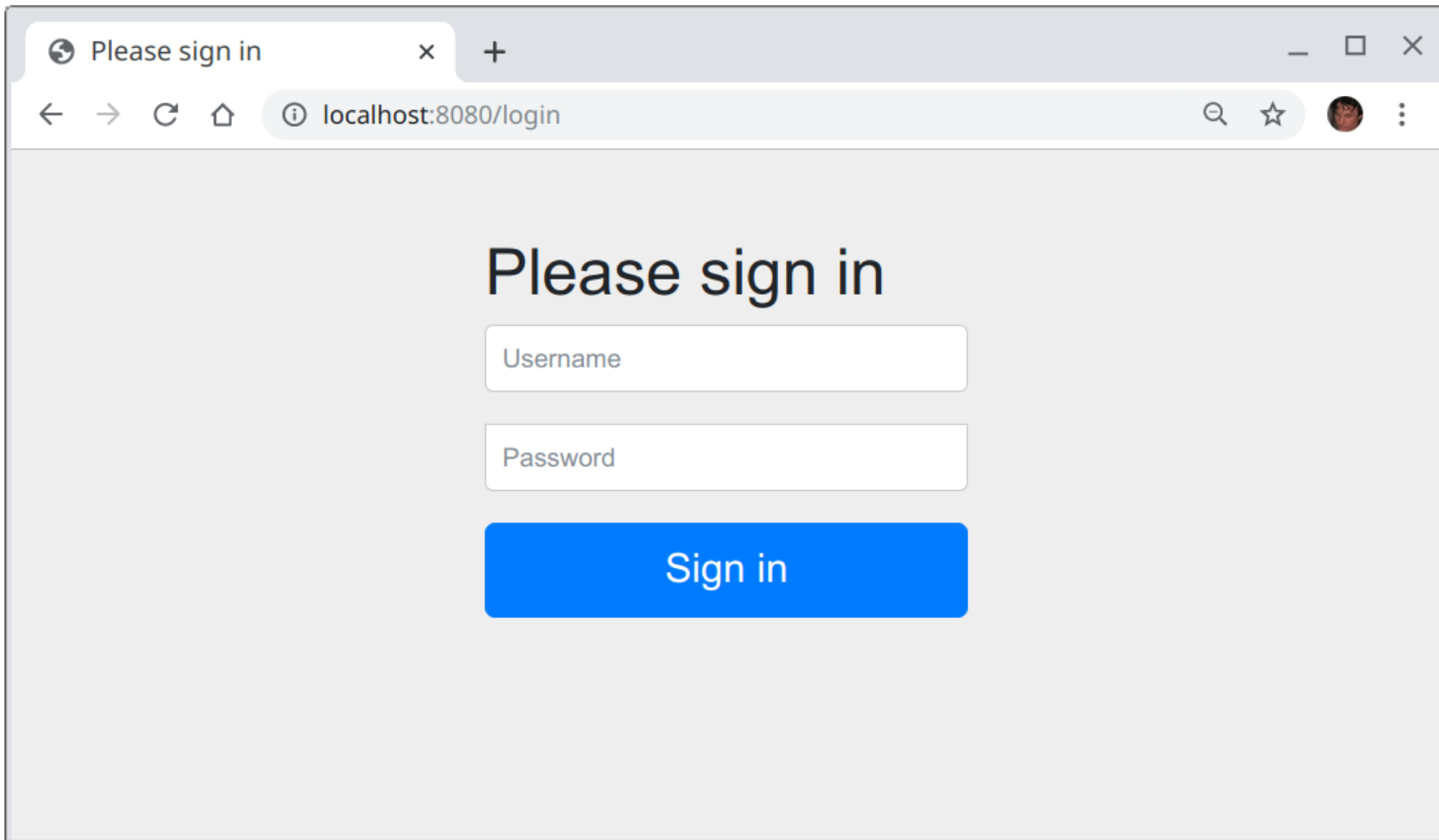
Autentimise protsess

1. Küsime mingit lehte (http päring)
2. Server näeb, et selle lehe näitamiseks on vaja autentida
3. Server teatab sellest (nt. suunamine sisselogimise lehele)
4. Kasutaja täidab ja postitab vormi
5. Server kontrollib saadetud info õigsust
6. Server märgib kasutaja autendituks
7. Server seab küpsise ja suunab punktis 1. küsitud lehele

Spring Security

Stsenarium

http://localhost:8080 -> http://localhost:8080/login -> http://localhost:8080



A screenshot of a web browser window. The browser has a single tab titled "Please sign in". The address bar shows the URL "localhost:8080/login". The page content is a simple sign-in form with a light gray background. At the top, the text "Please sign in" is displayed in a large, dark font. Below this, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both fields are white with a light gray border. Below the input fields is a blue button with the text "Sign in" in white. The browser's navigation bar includes back, forward, and refresh buttons, as well as a search icon, a star icon, and a profile picture icon.

Please sign in

Username

Password

Sign in

Teegid

```
implementation group: 'org.springframework.security',  
               name: 'spring-security-web',  
               version: '5.4.1.RELEASE'
```

```
implementation group: 'org.springframework.security',  
               name: 'spring-security-config',  
               version: '5.4.1.RELEASE'
```

Põhimõte



Mvc raamistiku laadimine

```
public class MyApplicationInitializer extends
    AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected String[] getServletMappings() {
        return new String[] { "/api/*" };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { MvcConfig.class };
    }

}
```

Spring Security laadimine

```
public class SecurityWebApplicationInitializer  
    extends AbstractSecurityWebApplicationInitializer {  
  
}
```



Spring Security konfiguratsiooni laadimine

```
public class MyApplicationInitializer extends
    AbstractAnnotationConfigDispatcherServletInitializer {

    ...

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { MvcConfig.class };
    }

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] { SecurityConfig.class };
    }

    ...
}
```

Seadistamine

@Configuration

@EnableWebSecurity

public class SecurityConfig **extends** WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) **throws** Exception {

 http.authorizeRequests()

 .antMatchers("/static/**").permitAll()

 .antMatchers("/admin/**").hasRole("ADMIN")

 .antMatchers("/**").authenticated();

 http.formLogin();

}

}

/* - /home

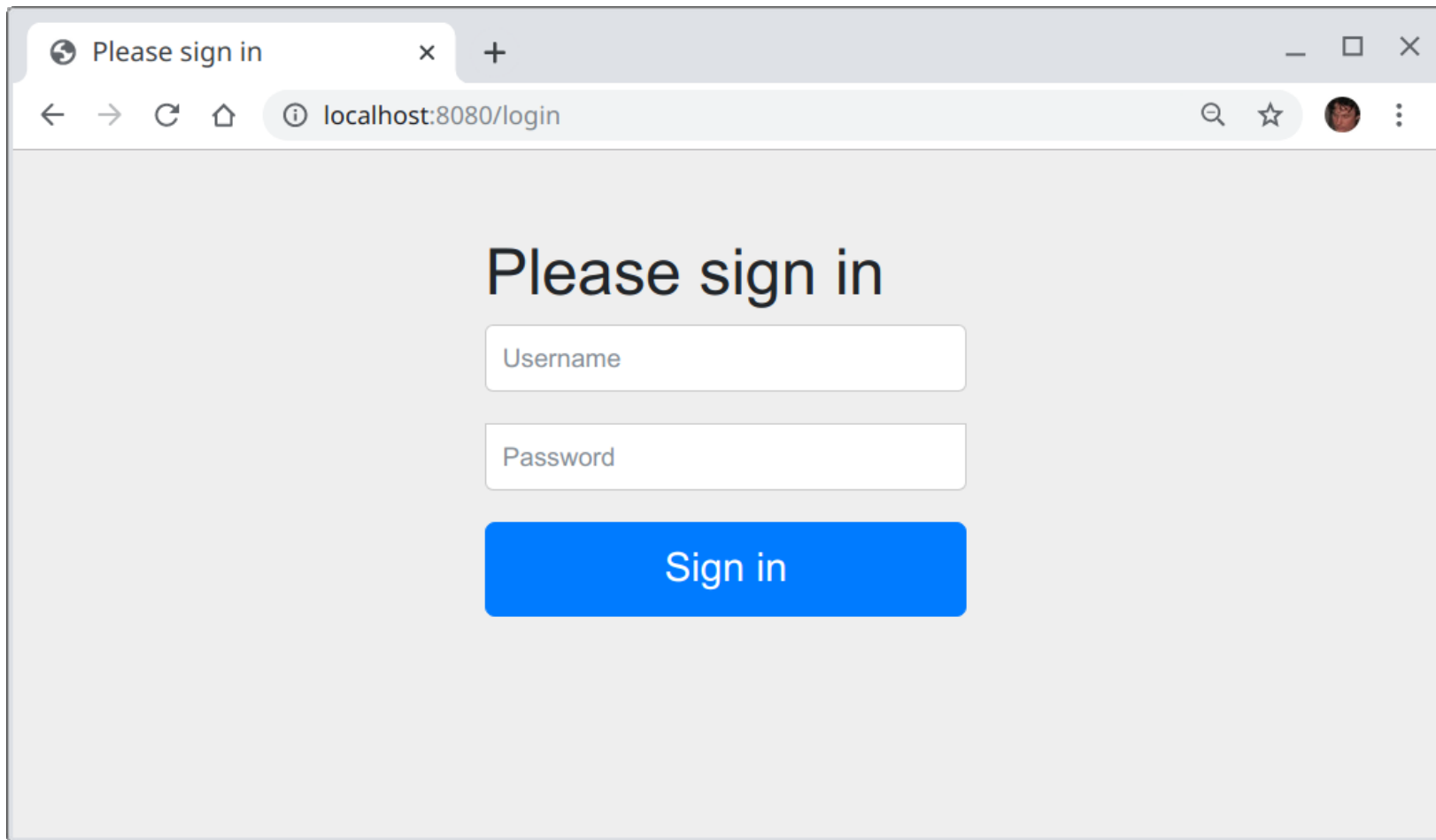
/* - /home?lang=en

/** - /home

/** - /products/88135/details

Standard login vorm

http://localhost:8080



The image shows a web browser window with a single tab titled "Please sign in". The address bar displays "localhost:8080/login". The page content features a heading "Please sign in" followed by two input fields: "Username" and "Password". Below these fields is a prominent blue button labeled "Sign in".

Please sign in

Username

Password

Sign in

Seadistamine (Spring conf)

@Override

```
protected void configure(AuthenticationManagerBuilder builder) {  
  
    builder.inMemoryAuthentication()  
        .passwordEncoder(new BCryptPasswordEncoder())  
        .withUser("user")  
        .password("$2a$10$3rN/1Dt4gMt4NacjYJn5LeVFXlB7a...")  
        .roles("USER")  
        .and()  
        .withUser("admin")  
        .password("$2a$10$WID7JrpmmWCGzQ0DSHjNo0p6/RC8t...")  
        .roles("ADMIN", "USER");  
}
```

Spring Security ja Web API

Web API autentimise protsess

1. Küsime mingit ressurssi (http päring).
 2. Server näeb, et selle näitamiseks on vaja autentida ja teatab sellest [http koodiga](#) (nt. 401 või 403).
-
1. Klientrakendus saadab vajaliku info (nt [JSON](#) kujul).
 2. Server kontrollib saadetud info õigsust.
 3. Server märgib kasutaja autendituks ja seab küpsise. [Edasi ei suuna.](#)

Vajalikud muudatused

- Vormi asemel saata http kood (nt 401)
- Logimise infot tuleks vastu võtta Json kujul
- Õnnestunud sisselogimise korral teatada ainult koodiga (200)
- Ebaõnnestunud sisselogimise korral teatada ainult koodiga (401)

Spring Security filter chain



ChannelProcessingFilter.class
ConcurrentSessionFilter.class
WebAsyncManagerIntegrationFilter.class
SecurityContextPersistenceFilter.class
HeaderWriterFilter.class
CorsFilter.class
CsrfFilter.class
LogoutFilter.class
X509AuthenticationFilter.class
...

Vajalikud muudatused

@Override

protected void configure(HttpSecurity http) **throws** Exception {

...

~~http.formLogin();~~

http.**exceptionHandling()**

 .authenticationEntryPoint(**new** ApiEntryPoint());

http.**exceptionHandling()**

 .accessDeniedHandler(**new** ApiAccessDeniedHandler());

http.logout()

 .logoutUrl("/**api/logout**")

 .logoutSuccessHandler(**new** ApiLogoutSuccessHandler());

Kasutaja info Json-ist

@Override

protected void configure(HttpSecurity http) ...

...

var loginFilter = **new** ApiAuthenticationFilter(
 authenticationManager(), **"/api/login"**);

http.addFilterAfter(loginFilter, LogoutFilter.**class**);

Dokumentatsioon

<https://docs.spring.io/spring-security/site/docs/current/reference/html5/>

Miks Spring Security?

- Isetehtud asjad sisaldavad enamasti turvaauke
- OpenID ja OAuth 2.0
- Meetodipõhine juurdepääsu kontroll
- Muud mugavused (nt. soolaga salasõna valideerimine)

Salasõnade hoidmine

1	Alice Smith	alice	h43#kP!o
2	Bob Jones	bob	s3cret
...

Krüptograafiline räsi

Sisend: secr3t

Räsi: 2bb80d537b1da3e38bd30361aa85568...

Salasõnade hoidmine

1	Alice Smith	alice	36AED8FB9E4CEA72FD5746...
2	Bob Jones	bob	5FECEB66FFC86F38D958A7...
...

Brute Force rünnak

- Suured tähed, väikesed tähed, numbrid, sümbolid teeb umbes 70 kombinatsiooni.
- 999999 - 10^6 kombinatsiooni
- g3#F@8 - 70^6 kombinatsiooni

Brute Force rünnak

- 70^6 kombinatsiooni (~100 miljardit) ~ 45 tundi (ühes lõimes)

BCrypt

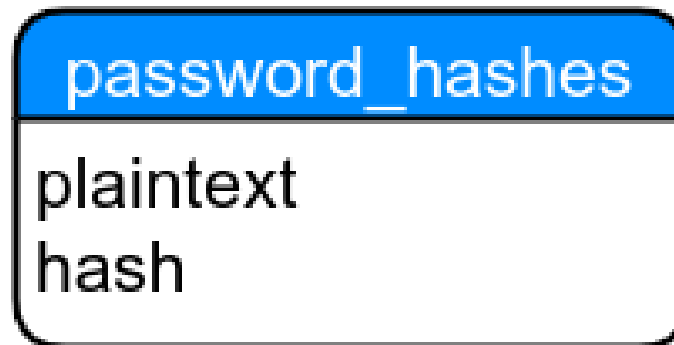
keerukus 10 tähendab $2^{10} = 1024$

```
BCryptPasswordEncoder encoder = new BCryptPasswordEncoder(10);  
encoder.encode("s3cret");
```

BCrypt

- 70^6 kombinatsiooni keerukusega 10 ~ 240 aastat (ühes lõimes)
- 70^6 kombinatsiooni keerukusega 11 ~ 480 aastat

Rainbow Tables



00ab	36AED8FB9E4CEA72FD5746FADCD9B9C5AA93B11B27FF8E5...
0	5FECEB66FFC86F38D952786C6D696C79C2DBC239DD4E91B...
1	6B86B273FF34FCE19D6B804EFF5A3F5747ADA4EAA22F1D4...
secr3t	959C9F50AEF1BC129A0E16564319A1B36515D57051307...
2	D4735E3A265E16EEE03F59718B9B5D03019C07D8B6C51F9...
	...

Rainbow Tables

- Tüüpilised salasõnad
- Varastatud salasõnad
- Modifikatsioonid tüüpilistest salasõnadest (s3cret, secret83, etc...)

Rainbow tables

- <http://md5decrypt.net/en/Sha256/>
- „24,896,331 cracked hashes since 11/2015”
- „There are 15,183,605,161 words in the database”

Sool (Salt)

Sisend: “s3cret”

Räsi: 1ec1c26b50d5d3c58d9583181af807665...

Sisend: “5Gs0P!#x%gAp” + “s3cret”

Räsi: 781397c8a4c2fcbff3c1401359866f10448...

BCrypt

```
BCryptPasswordEncoder encoder = new BCryptPasswordEncoder(10);  
  
System.out.println(encoder.encode("h43#kP"));  
System.out.println(encoder.encode("h43#kP"));  
System.out.println(encoder.encode("h43#kP"));
```

```
$2a$10$FKeriJfGsK.RTLiYyyi1YugQ.uf0./hYFUxmnuW3b5wZMT5vYZPBy  
$2a$10$KQOz.6sAJXQ5BEloSASSK.94V40fJ/g5Gk8GqHEG.22XdgZ33CnRa  
$2a$10$FUAaKwoA8QZYpmBz0aJoquTPISOUWE.KsKRxwgFPK3jvSHPiAhS0W
```


BCrypt

\$2a\$10\$N9qo8uLOickgx2ZMRZoMyeljZAgcfl7p92ldGxad68LJZdL17lhWy

Räsi kasutamine

@Override

```
protected void configure(AuthenticationManagerBuilder builder) {  
  
    builder.inMemoryAuthentication()  
        .passwordEncoder(new BCryptPasswordEncoder())  
        .withUser("user")  
        .password("$2a$10$3rN/1Dt4gMt4NacjYJn5LeVFXlB7a...")  
}
```

Eristamine meetodite või andmete tasemel

- nt. müügiinimene näeb ainult enda sisestatud klientide infot

Authentication, Principal

```
@RestController
public class UserController {

    @GetMapping("/users/{username}")
    public User getUserByName(@PathVariable String username,
                             Authentication auth,
                             Principal principal) {

        System.out.println("Principal: " + principal.getName());
        System.out.println("Authorities: " + auth.getAuthorities());
        System.out.println("auth.principal: " + auth.getPrincipal());

        if (...)

        ...
    }
}
```

@PreAuthorize

```
@RestController
public class UserController {

    ...

    @GetMapping("/users/{username}")
    @PreAuthorize("#username == authentication.name")
    public User getUserByName(@PathVariable String username) {
        return dao.findByUsername(username);
    }
}
```

Info hoidmisest sessioonis

- Iga uus kasutaja võtab mälu (probleem bot-idega)
- Probleem serverite klastritega

```
@GetMapping("/home")  
public String counter(HttpSession session) {  
    // loob sessiooni, kui seda veel pole
```

Lahendus

- Küpsises või header-is on kogu info (Kes? Mis õigustega? Kauda kehtib?).

JSESSIONID=n0ql62cg2clc8u1jr6hl4jvqavi0

vs.

TOKEN=eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ1c2VyliwiZXhwIjoxNTczNTUwOTQ5LCJyb2xlcyI6InVzZXIsIGFkbWluIn0.6zRxQA8MGWbLVwz0pWh_zHUBmhiyZG9phvs22bQ8PHI82mKwuOw5duQVI3kpKNZQjMvxhRTnzkDJ4bJaUZ9qPw

Token

päring

```
POST /login HTTP/1.1  
Host: localhost  
...  
{ "user" = "alice", "password" = "s3cret" }
```

vastus

```
HTTP/1.1 200 OK  
Authorization: eyJhbGciOiJIUzUxMiJ9...  
...
```

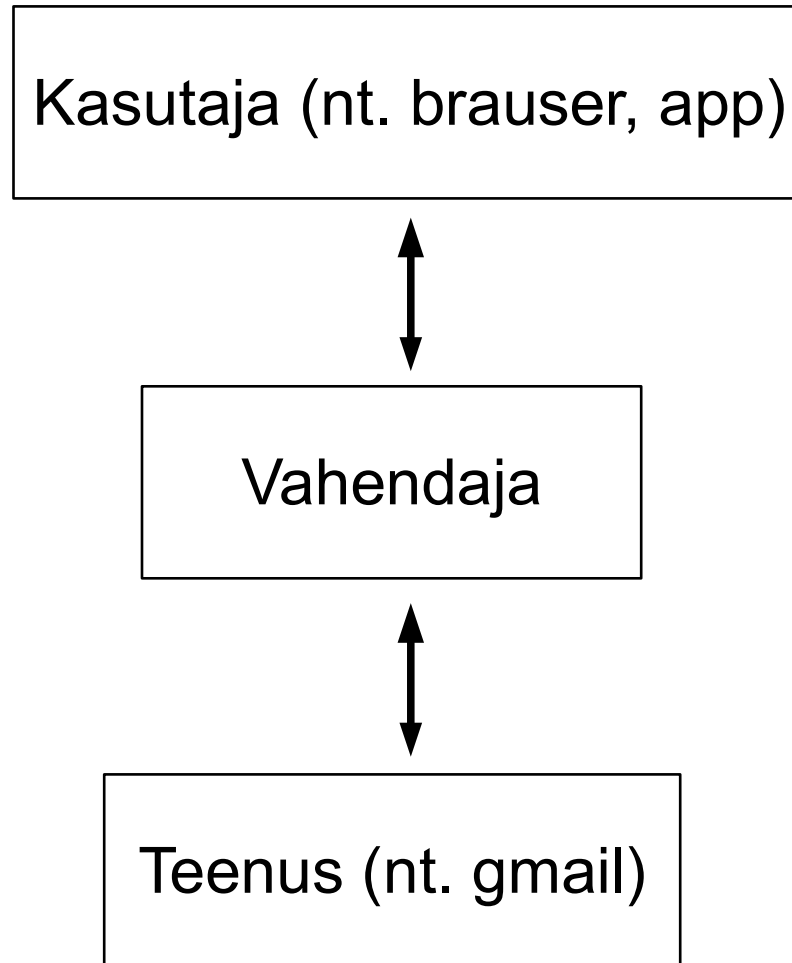
järgnevad päringud

```
GET /contacts HTTP/1.1  
Authorization: eyJhbGciOiJIUzUxMiJ9...  
Host:localhost
```

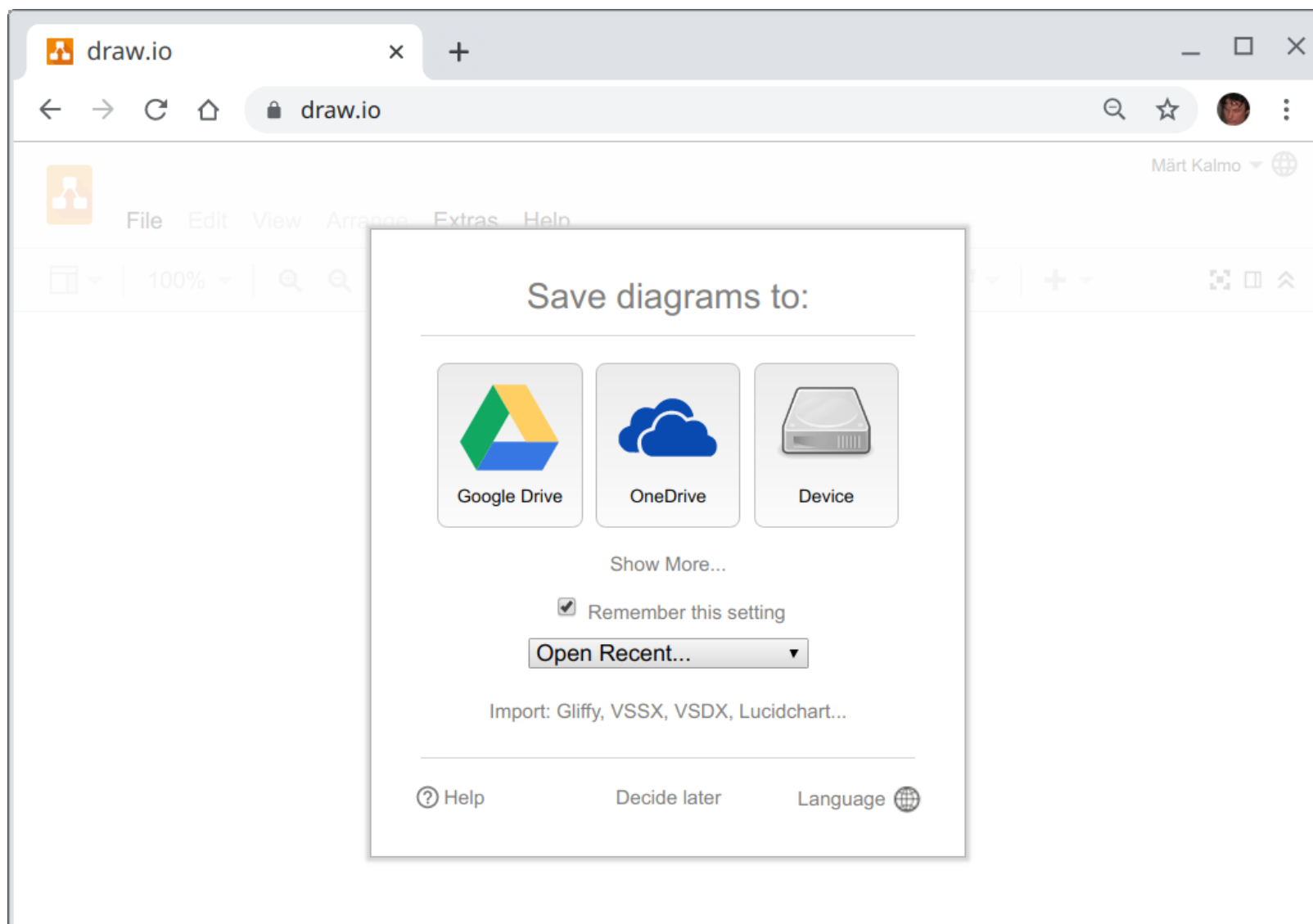

JSON Web Token

- https://en.wikipedia.org/wiki/JSON_Web_Token
- **Authorization: Bearer** eyJhbGciOiJIUzUxMiJ9...
- Info on loetav (base64 kodeering)
- Info pole muudetav (sisaldab info ja salajase võtme räsi).

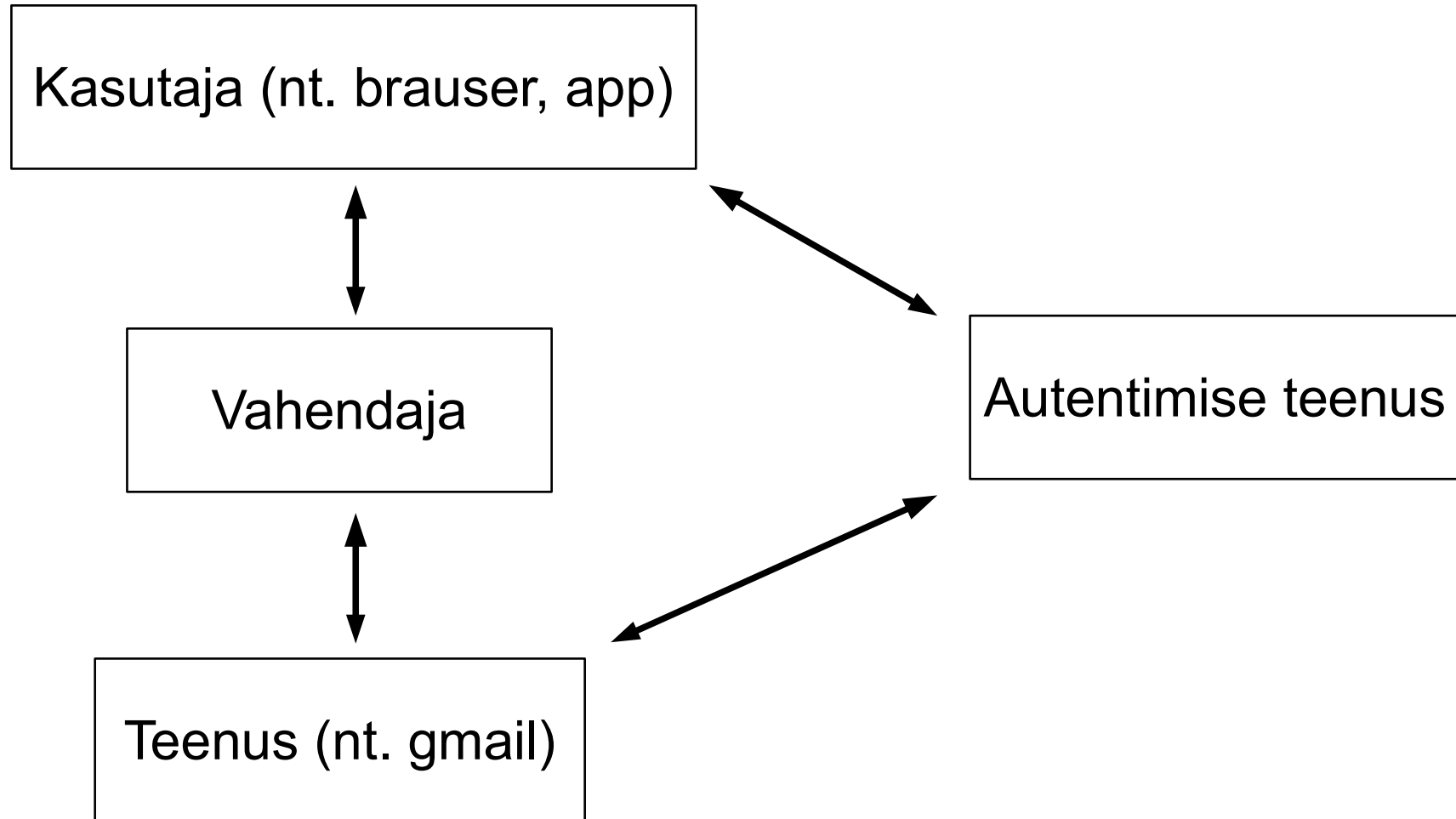
Teenuste vahendamine



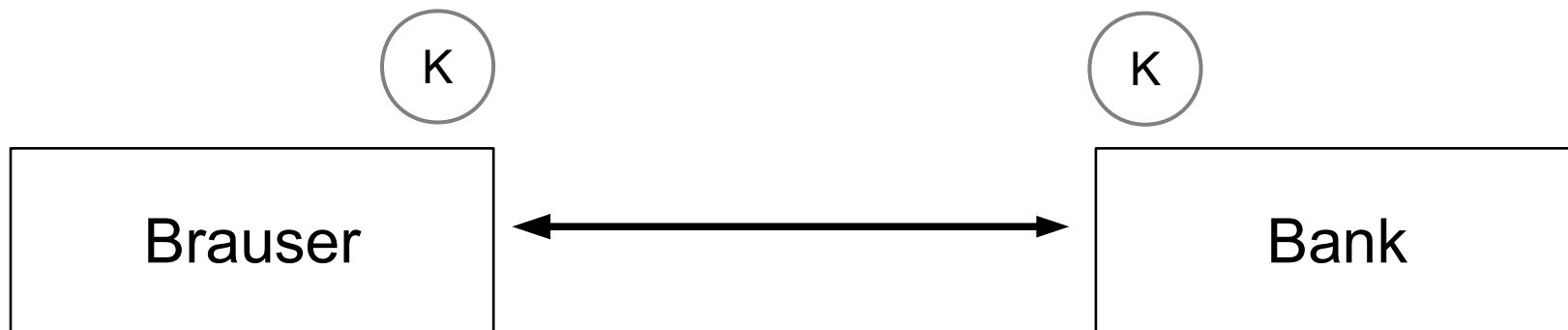
Teenuste vahendamine



OAuth2.0 (lihtsustatud)

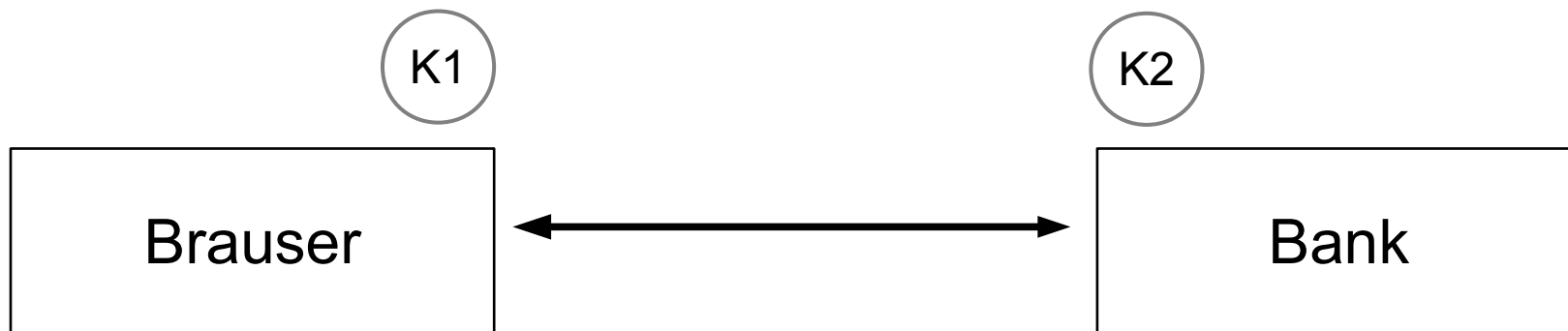


Sümmeetrilise võtme krüptograafia

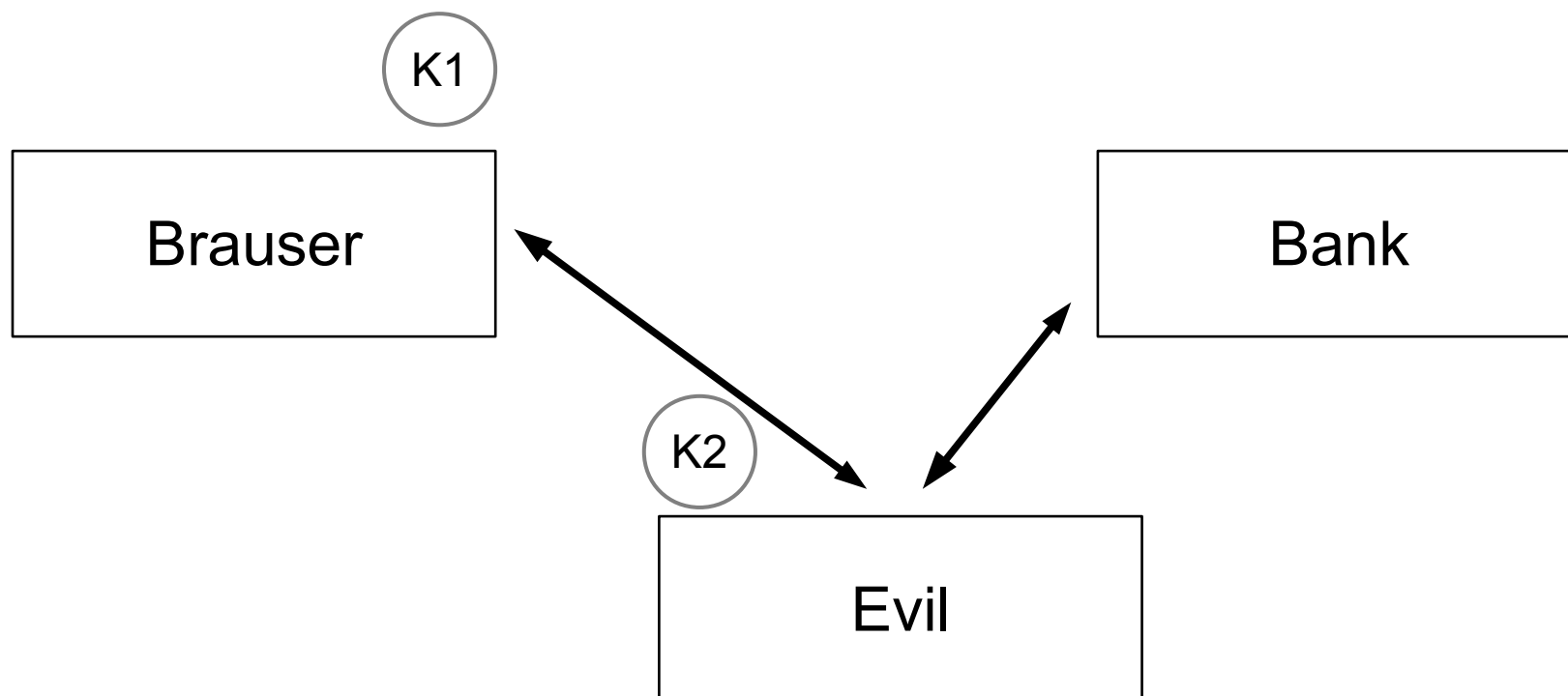


Avaliku võtme krüptograafia

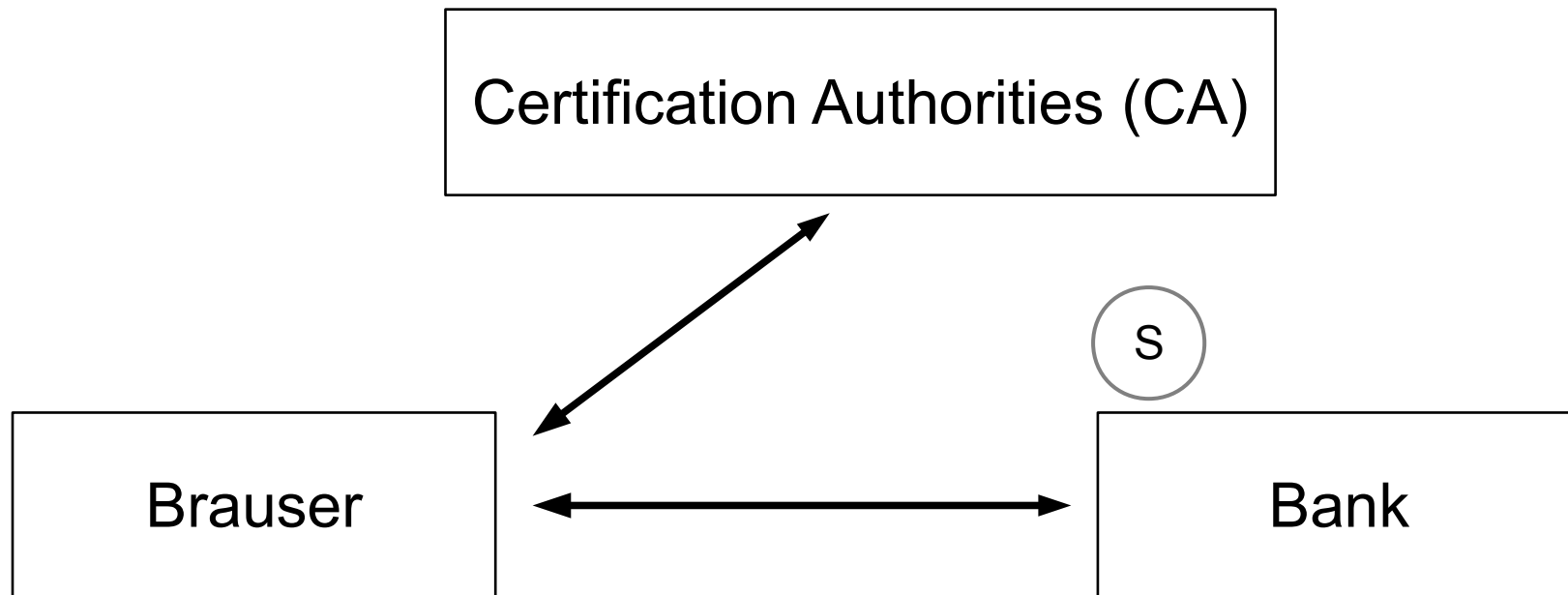
- Lahendab võtme vahetuse probleemi
- Üks võti krüpteerimiseks, teine dekrüpteerimiseks



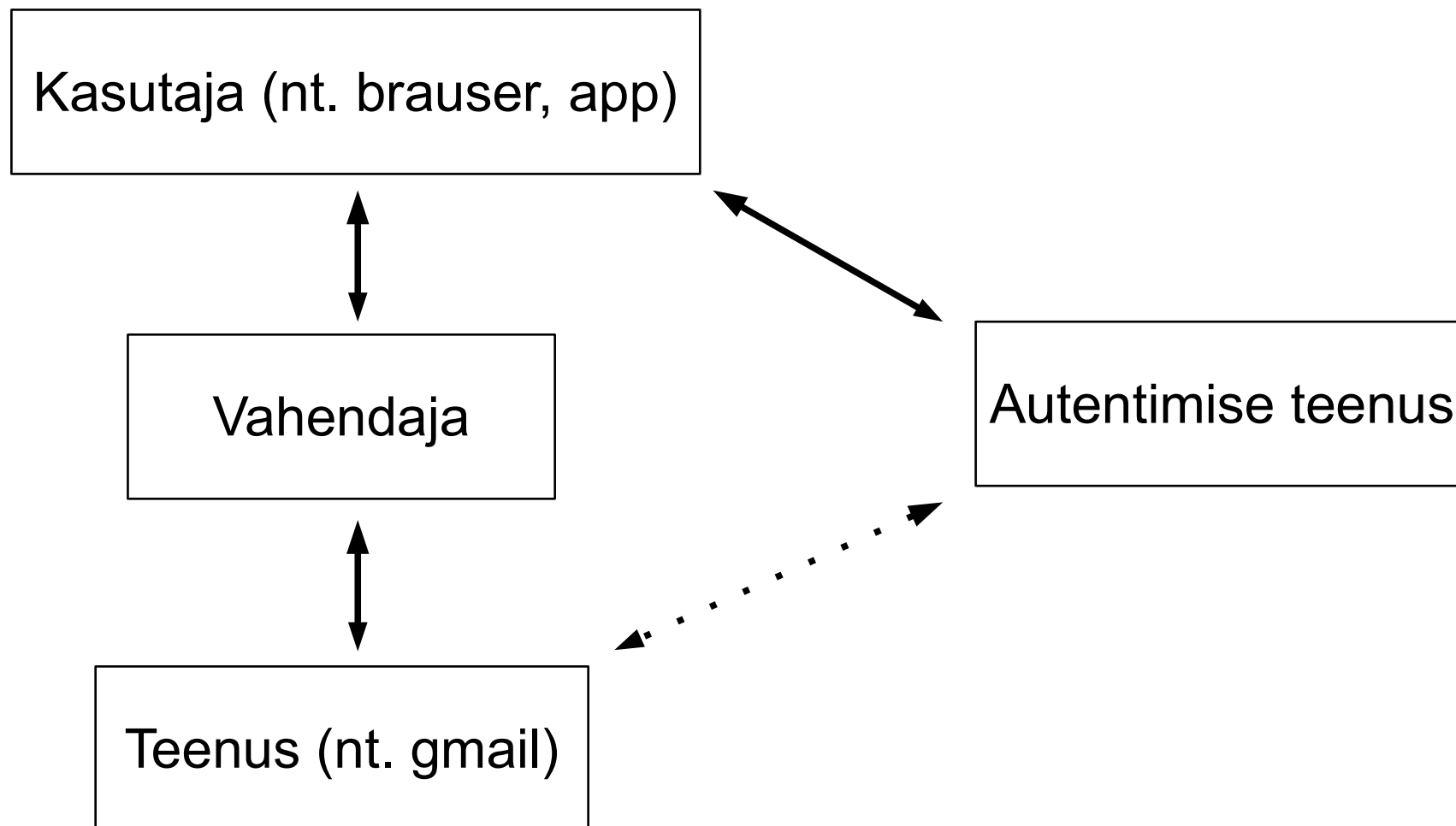
Avaliku võtme krüptograafia



Avaliku võtme krüptograafia



OAuth2.0 (lihtsustatud)



Probleemid

- Lahenduse keerukus
- Token-i uuendamine
- Väljalogimine

Millist lahendust kasutada?

- Kas kasutame klastrit?
- Kas väljalogimine on oluline?
- Kas sobib mingi hübriidlahendus (nt. sessioon ilma küpsiseta)?
- Kui keeruline on meie süsteem (nt. sso)?
- Milliste muude süsteemidega peab süsteem ühilduma?

Spring Security kasutajad andmebaasist

- Tabelid
- Andmed
- Seadistus

Tabelid

```
CREATE TABLE USERS (  
    username VARCHAR(255) NOT NULL PRIMARY KEY,  
    password VARCHAR(255) NOT NULL,  
    enabled BOOLEAN NOT NULL,  
    first_name VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE AUTHORITIES (  
    username VARCHAR(50) NOT NULL,  
    authority VARCHAR(50) NOT NULL,  
    FOREIGN KEY (username) REFERENCES USERS  
        ON DELETE CASCADE  
);
```

```
CREATE UNIQUE INDEX ix_auth_username  
    ON AUTHORITIES (username, authority61);
```

Andmed

```
INSERT INTO users (USERNAME, PASSWORD, ENABLED, FIRST_NAME)
VALUES ('user', '$2a$04$3rN/1Dt4gMt4...', true, 'Jack');
```

```
INSERT INTO users (USERNAME, PASSWORD, ENABLED, FIRST_NAME)
VALUES ('admin', '$2a$10$WID7JrpmmWC...', true, 'Jill');
```

```
INSERT INTO AUTHORITIES (USERNAME, AUTHORITY)
VALUES ('user', 'ROLE_USER');
```

```
INSERT INTO AUTHORITIES (USERNAME, AUTHORITY)
VALUES ('admin', 'ROLE_ADMIN');
```

Seadistus

@Override

```
protected void configure(AuthenticationManagerBuilder builder) {  
  
    builder.jdbcAuthentication()  
        .dataSource(dataSource)  
        .passwordEncoder(new BCryptPasswordEncoder());  
}
```

MockMvc (Spring Security)

```
implementation group: 'org.springframework.security',  
               name: 'spring-security-test',  
               version: '5.4.1.RELEASE'
```


Cross-Site Request Forgery

@Override

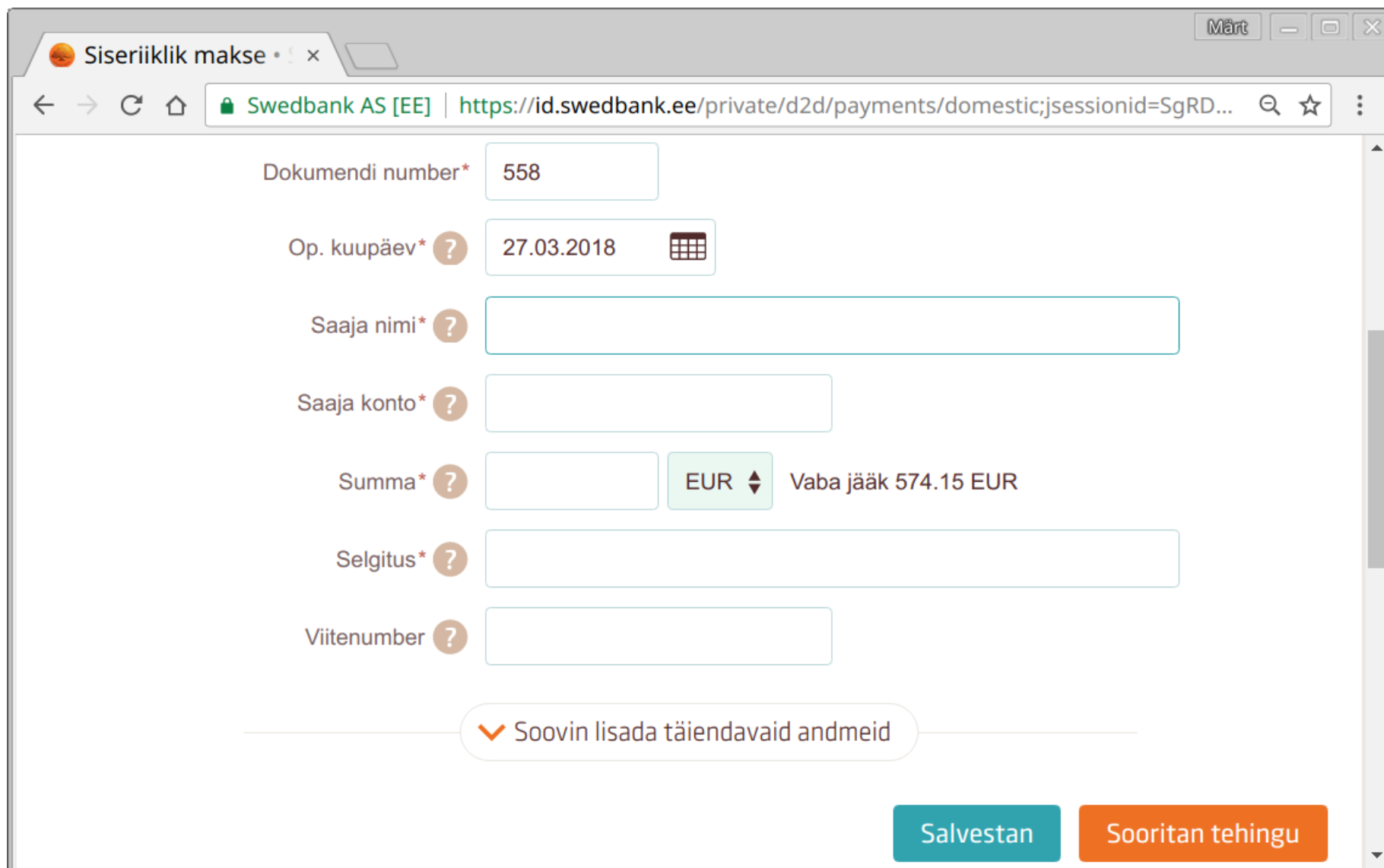
protected void configure(HttpSecurity http) {

http.csrf().disable();

...

}

CSRF rünnak



Siseriiklik makse

Swedbank AS [EE] | <https://id.swedbank.ee/private/d2d/payments/domestic;jsessionid=SgRD...>

Dokumendi number* 558

Op. kuupäev* ? 27.03.2018

Saaja nimi* ?

Saaja konto* ?

Summa* ? EUR ▼ Vaba jääk 574.15 EUR

Selgitus* ?

Viitenumber ?

✓ Soovin lisada täiendavaid andmeid

Salvestan Sooritan tehingu

CSRF rünnak

POST /transfer HTTP/1.1

Host: swedbank.ee

Cookie: JSESSIONID=randomid;

Content-Type: application/x-www-form-urlencoded

amount=100.00&toAccount=456&...

CSRF rünnak

```
<form action="https://swedbank.ee/transfer"
      method="post">

  <input type="hidden"
        name="amount"
        value="100.00"/>
  <input type="hidden"
        name="account"
        value="evilsAccountNumber"/>
  <input type="submit"
        value="Win Money!"/>

</form>
```

CSRF kaitse

POST /transfer HTTP/1.1

Host: bank.example.com

Cookie: JSESSIONID=randomid; Domain=bank.example.com;

Content-Type: application/x-www-form-urlencoded

amount=100.00&routingNumber=1234&account=9876&_csrf=...

CSRF ja Web API

@Override

protected void configure(HttpSecurity http) {

http.csrf().disable();

}