Single sign-on Looking Easy - SLEASY

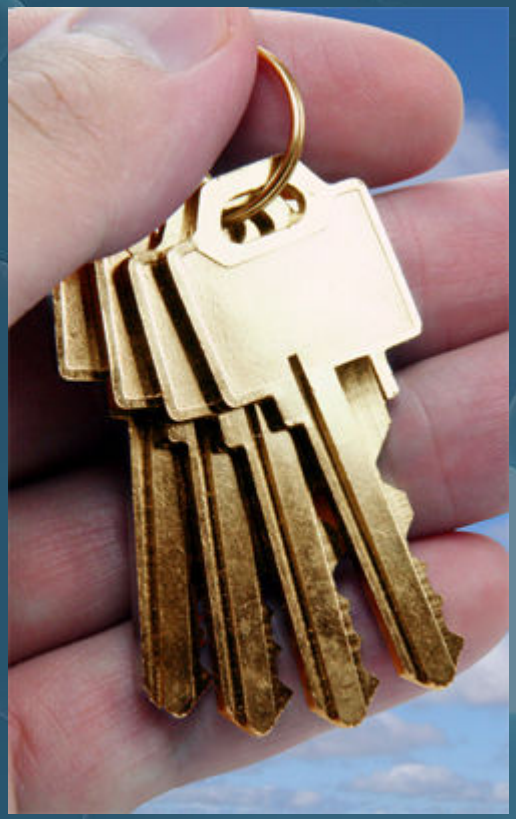
Darko Grabar
Mate Boban

studeni, 2007

Uvod

- Uvod u SSO
- Prednosti i nedostaci
- Postojeća rješenja
- Arhitektura sustava SLEASY
- Zaključak

SSO



SSO

- Open Group definira SSO kao mehanizam putem kojeg korisnik samo s jednom autentikacijskom i autorizacijskom procedurom može koristiti sva računala i sustave za koje ima pravo pristupa bez potrebe za ponovnim unosom lozinke
- Osnovni pojmovi
  - Autentikacija
    - Mehanizam za jednoznačnu identifikaciju korisnika
    - Daje odgovor na dva osnovna pitanja:
      - Tko je korisnik
      - Da li je korisnik zaista taj za kojeg se predstavlja
  - Autorizacija
    - Određuje koju razinu pristupa ima autenticirani korisnik (koja su njegova prava)

Prednosti

- Smanjenje operativnih troškova
- Skraćeno vrijeme pristupa podacima
- Poboljšan korisnički doživljaj
- Poboljšana sigurnost sustava
- Podrška za više autentikacijskih mehanizama
- Centralizirano upravljanje korisnicima

Nedostaci

- Centralno mjesto pada sustava (Single Point of Failure)
- Promjene u radu/trening
- Troškovi razvoja/implementacije

Tipovi SSO sustava?

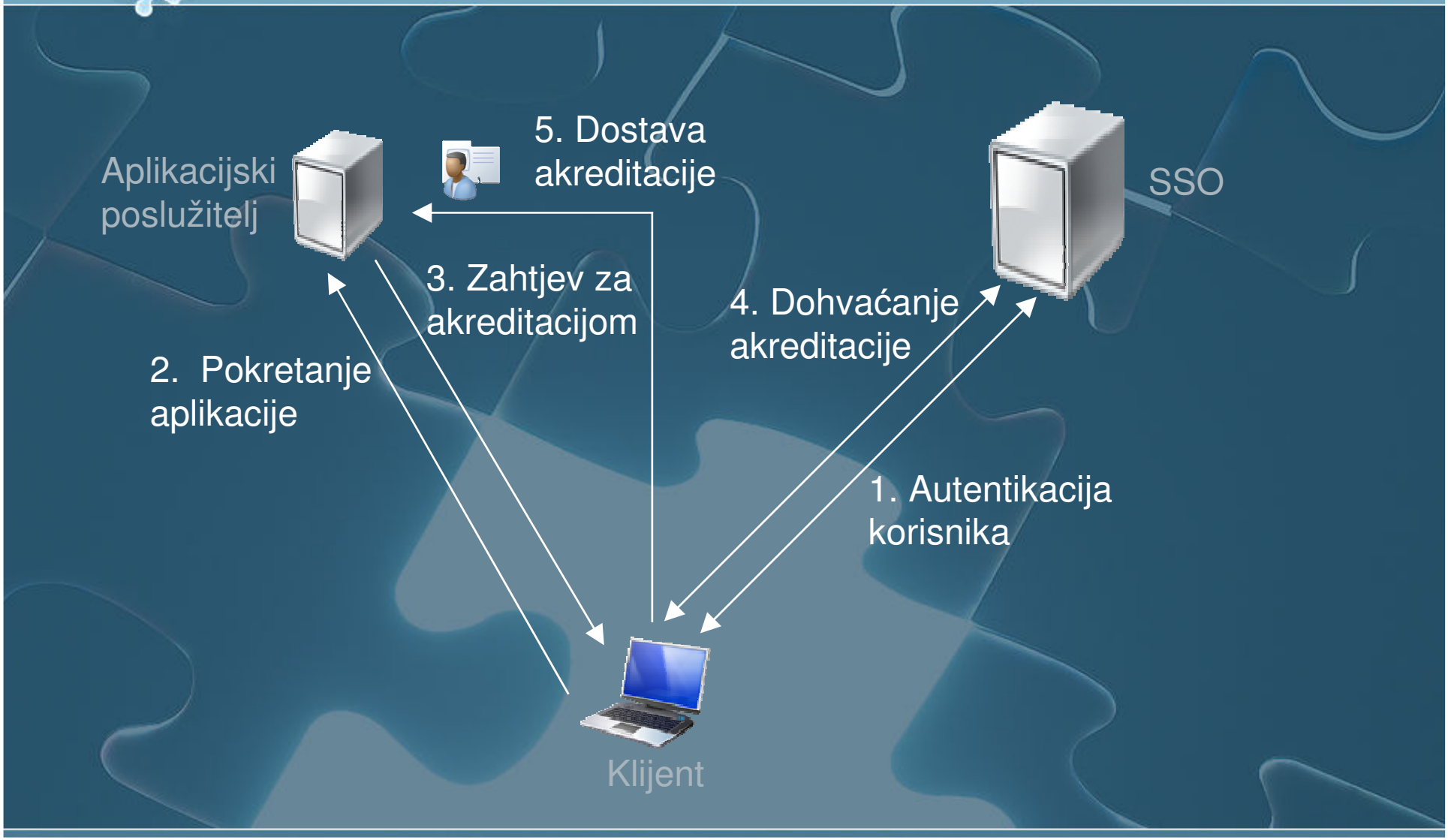
- Sinkronizacija lozinki
- SSO za naslijeđene sustave
- Pristup Web aplikacijama
- SSO preko višestrukih domena
- Federiran SSO (SAML, WS-Security)

Osnovni mehanizmi

- Sustavi temeljeni na kartama (tickets)
- Cookies
- Javni i privatni ključevi te digitalni certifikati

- Osnovni princip
  - Sustav kojem vjerujemo
    - Vrši autentikaciju i autorizaciju
    - Preusmjerava na neki vanjski resurs
  - Vanjski sustav
    - Zna da sam došao iz sustava kojem vjeruje
    - Može reći tko sam ovisno o informaciji dostavljenoj od strane sustava kojem se vjeruje

Arhitektura SSO



Postojeća rješenja

- The Liberty Alliance - SAML
- OASIS Web Services Security (WS-Security) – SAML
- XACML (+SAML)
- Internet2 Shibboleth project – SAML
- JOSSO - Java Open Single Sign-On

SAML

- Okvir baziran na XML-u
- Omogućava komunikaciju korisničke autenikacije, atributa te privilegija
- Komponente
  - Tvrdnja (Assertions): autentikacija, atributi, autorizacija
  - Protokoli: brojni request/response protokoli
  - Korice (Bindings): SOAP, HTTP redirect…
  - Profili: ograničenja i/ili ekstenzije za korištenje SAML-a s nekom aplikacijom

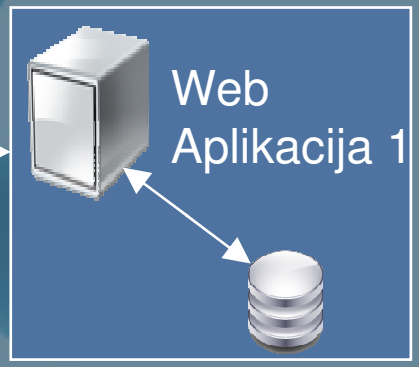
Single sign-on Looking Easy

- Tehnologije
  - Apache Web poslužitelj
  - PHP programski jezik (OpenSSL)
  - LDAP
  - PostgreSQL/MySQL baza podataka
- Karakteristike
  - Jednostavnost
  - Prilagođeno specifičnim potrebama
  - Minimalni zahtjevi na postojećim aplikacijama

Arhitektura [SLEASY]

(Prva prijava korisnika u bilo koju od aplikacija)

1. Korisnik zahtijeva pristup aplikaciji 1



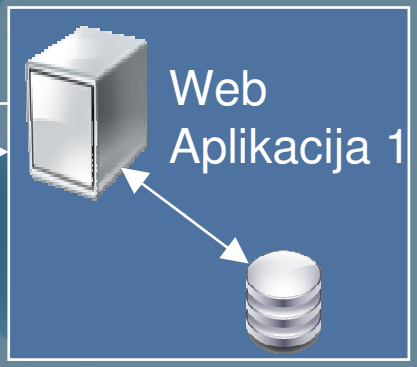
1

SSO – Single Sign-on
C – Klijent (Client)
A – Aplikacija

Arhitektura [SLEASY]

(Prva prijava korisnika u bilo koju od aplikacija)

1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**



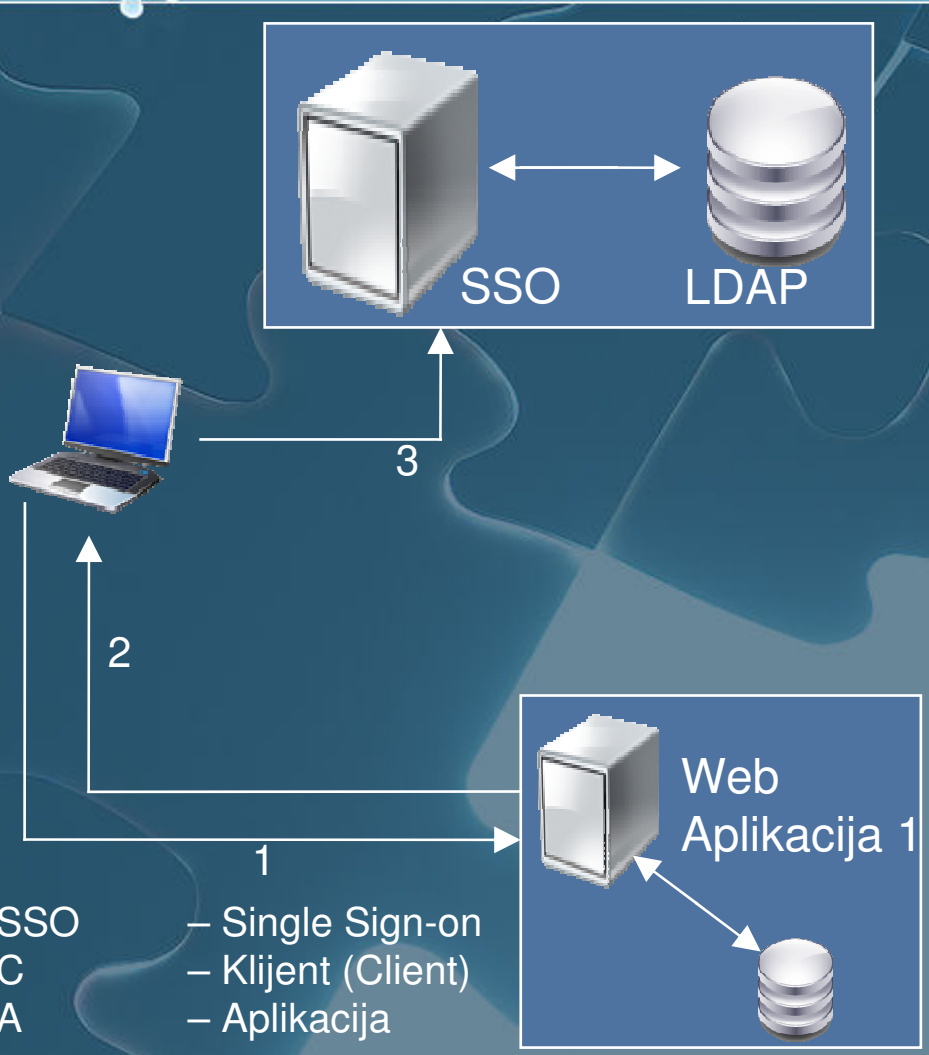
2

1

SSO – Single Sign-on
C – Klijent (Client)
A – Aplikacija

Arhitektura [SLEASY]

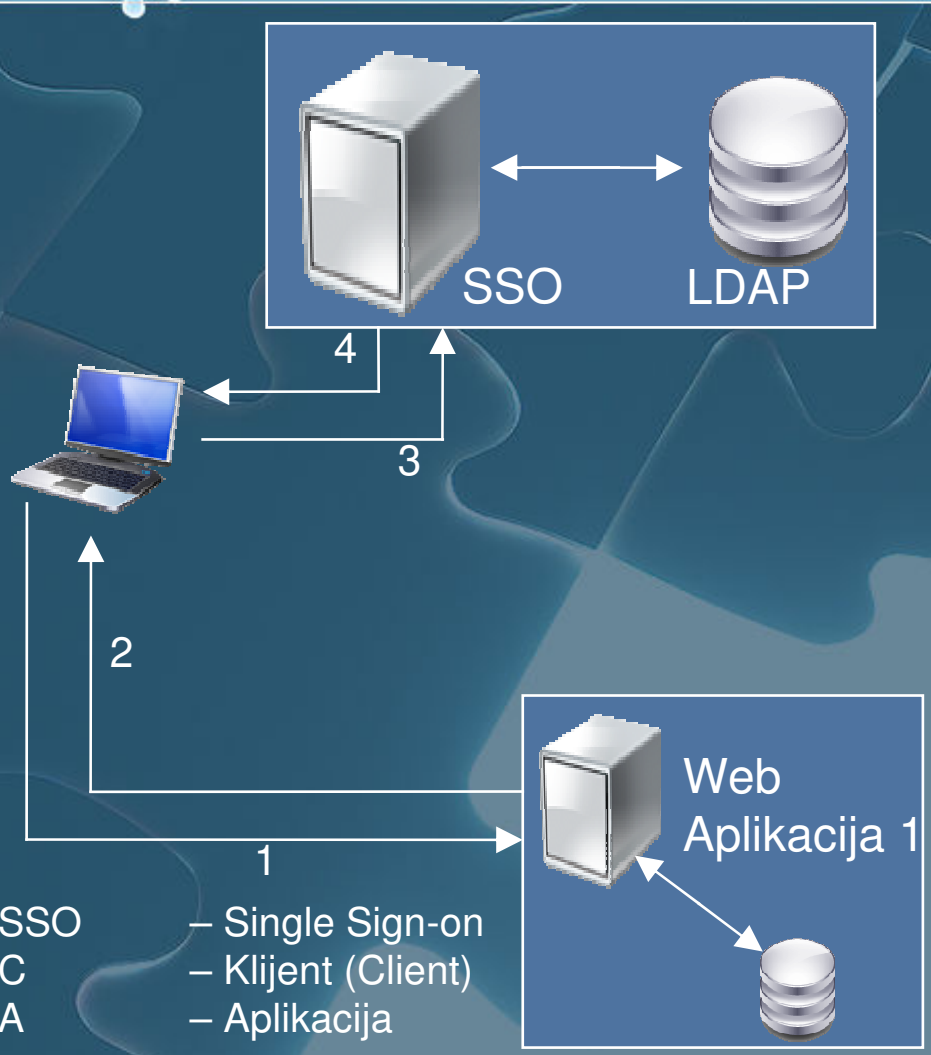
(Prva prijava korisnika u bilo koju od aplikacija)



1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju

Arhitektura [SLEASY]

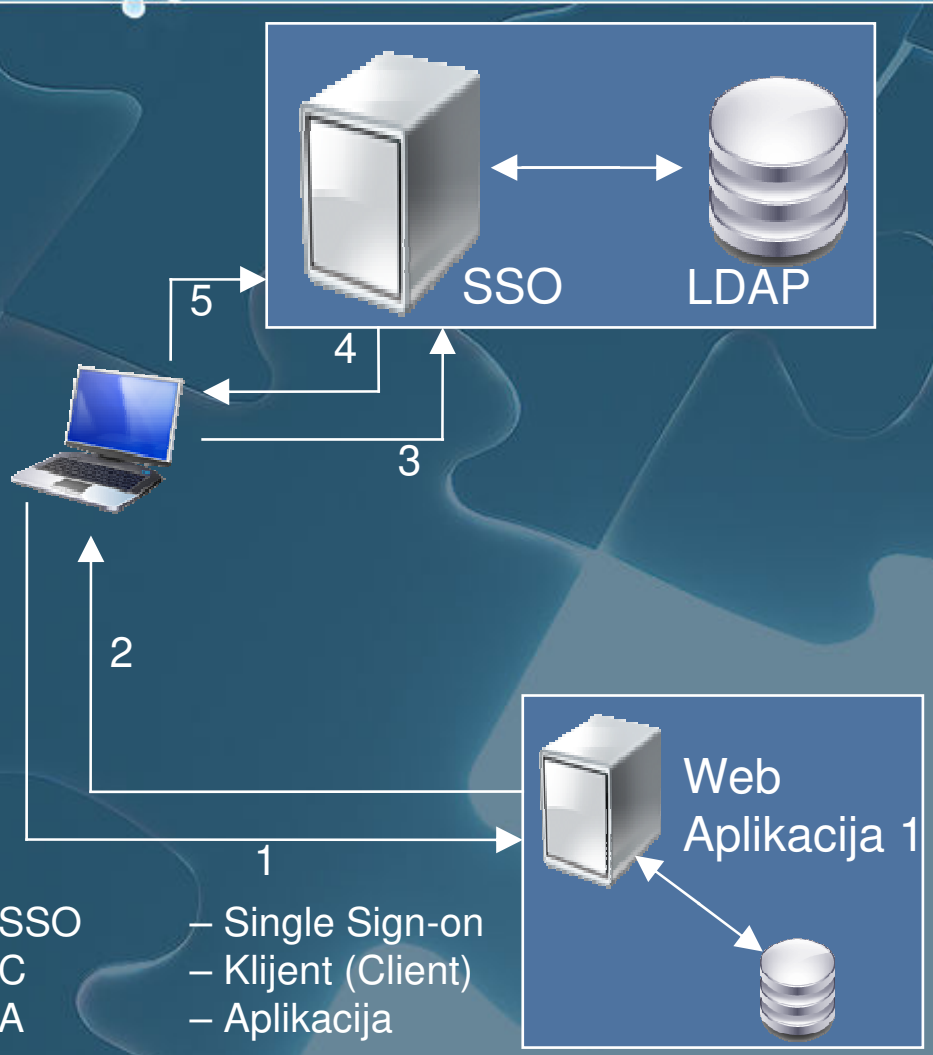
(Prva prijava korisnika u bilo koju od aplikacija)



1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. SSO dešifrira poruku te korisniku prikazuje formu za prijavu

Arhitektura [SLEASY]

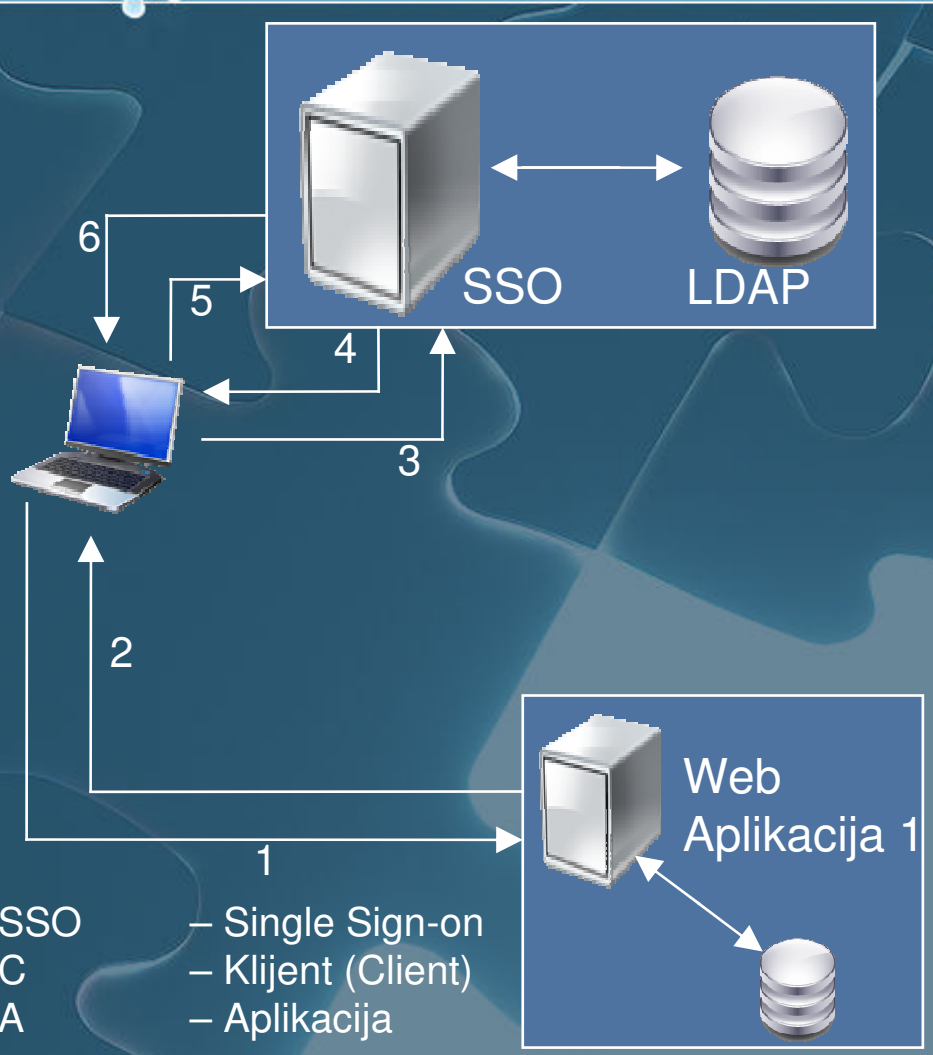
(Prva prijava korisnika u bilo koju od aplikacija)



1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. SSO dešifrira poruku te korisniku prikazuje formu za prijavu
5. Korisnik se autenticira

Arhitektura [SLEASY]

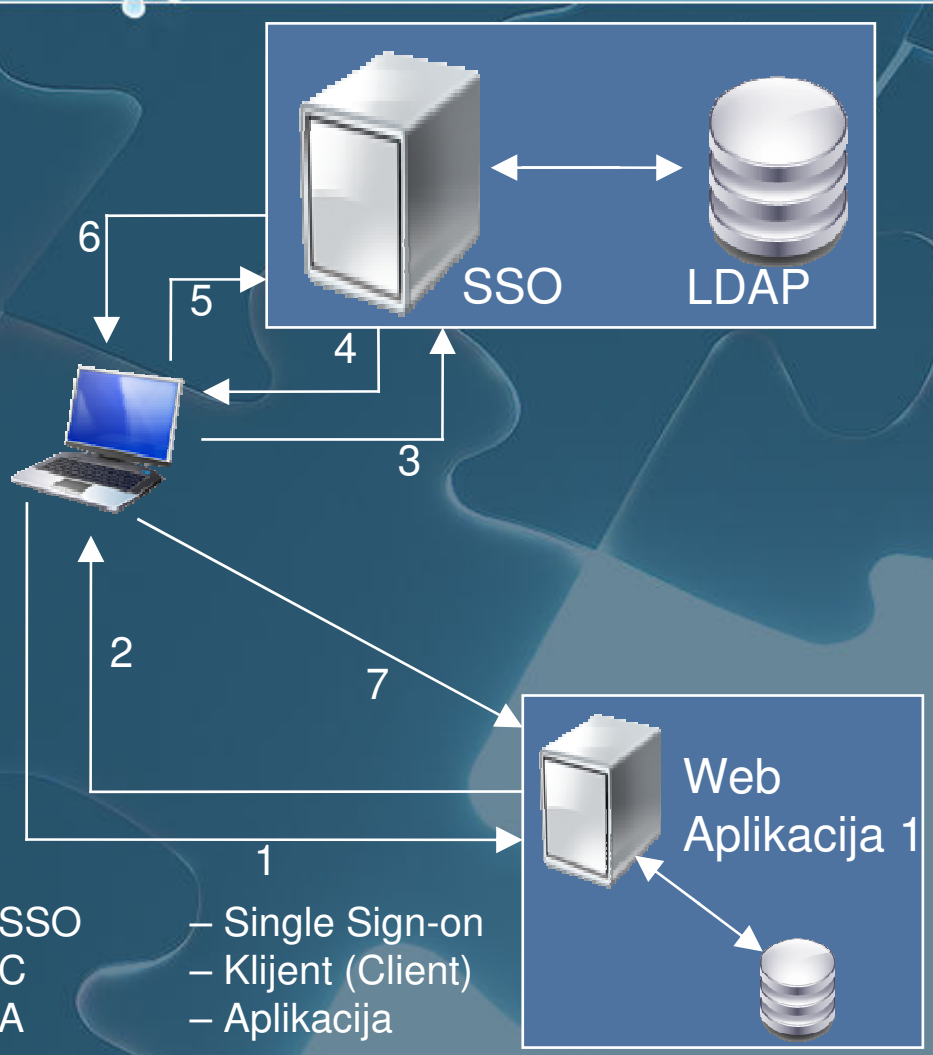
(Prva prijava korisnika u bilo koju od aplikacija)



1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^+ (Ka^- (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. SSO dešifrira poruku te korisniku prikazuje formu za prijavu
5. Korisnik se autenticira
6. SSO kreira odgovor te prosljeđuje poruku aplikaciji 1 : **(Ka^+ (Ks^- (poruka))**

Arhitektura [SLEASY]

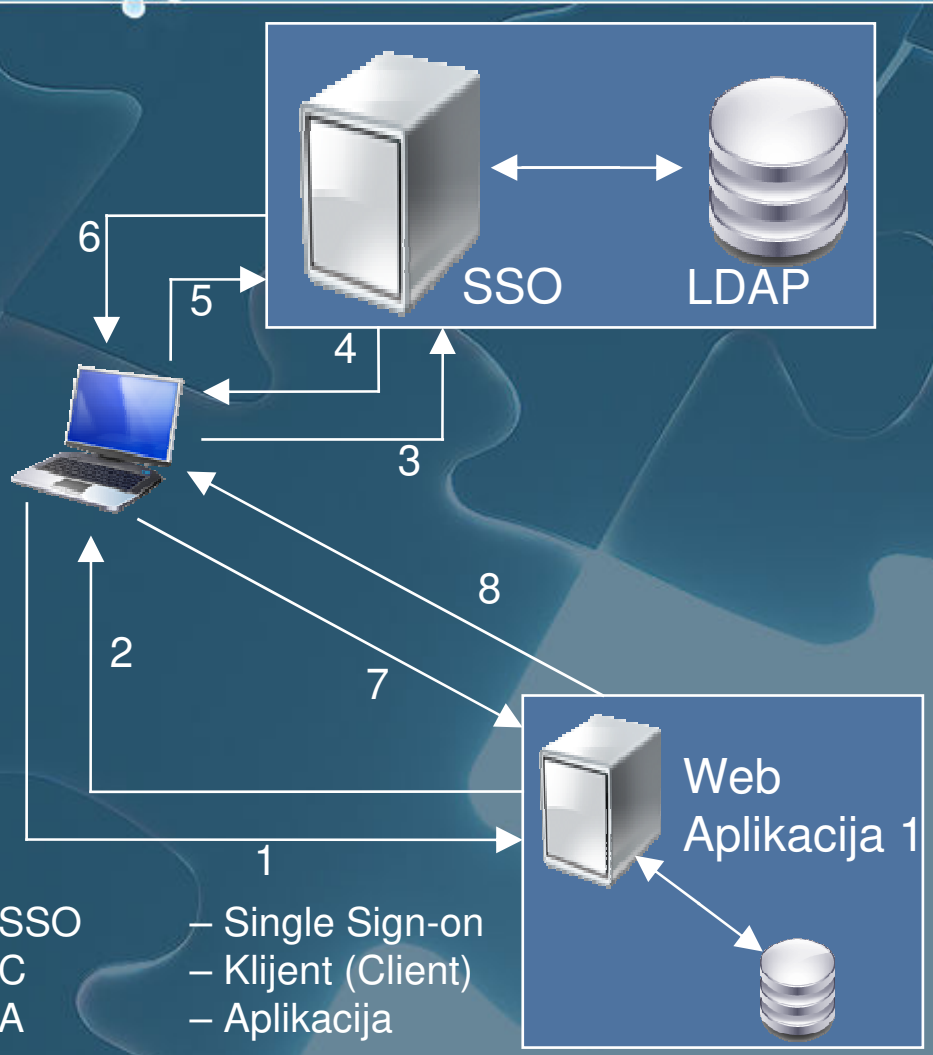
(Prva prijava korisnika u bilo koju od aplikacija)



1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. SSO dešifrira poruku te korisniku prikazuje formu za prijavu
5. Korisnik se autenticira
6. SSO kreira odgovor te prosljeđuje poruku aplikaciji 1 : **(Ka^{+} (Ks^{-} (poruka))**
7. Web preglednik prosljeđuje kriptiranu poruku aplikaciji 1

Arhitektura [SLEASY]

(Prva prijava korisnika u bilo koju od aplikacija)

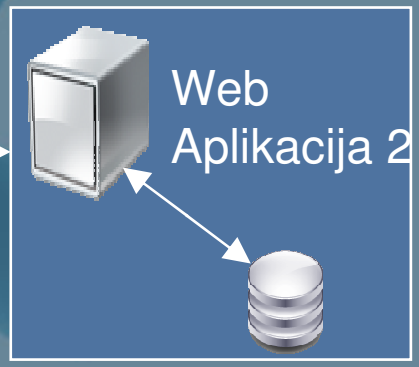


1. Korisnik zahtijeva pristup aplikaciji 1
2. Web preglednik je usmjeren na SSO poslužitelj - zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. SSO dešifrira poruku te korisniku prikazuje formu za prijavu
5. Korisnik se autenticira
6. SSO kreira odgovor te prosljeđuje poruku aplikaciji 1 : **(Ka^{+} (Ks^{-} (poruka))**
7. Web preglednik prosljeđuje kriptiranu poruku aplikaciji 1
8. Nakon što je dešifrirala poruku, aplikacija 1 potvrđuje da se korisnik uspješno prijavio (ili ne)

Arhitektura [SLEASY]

(Naknadne prijave u aplikacije)

1. Korisnik zahtijeva pristup aplikaciji 2



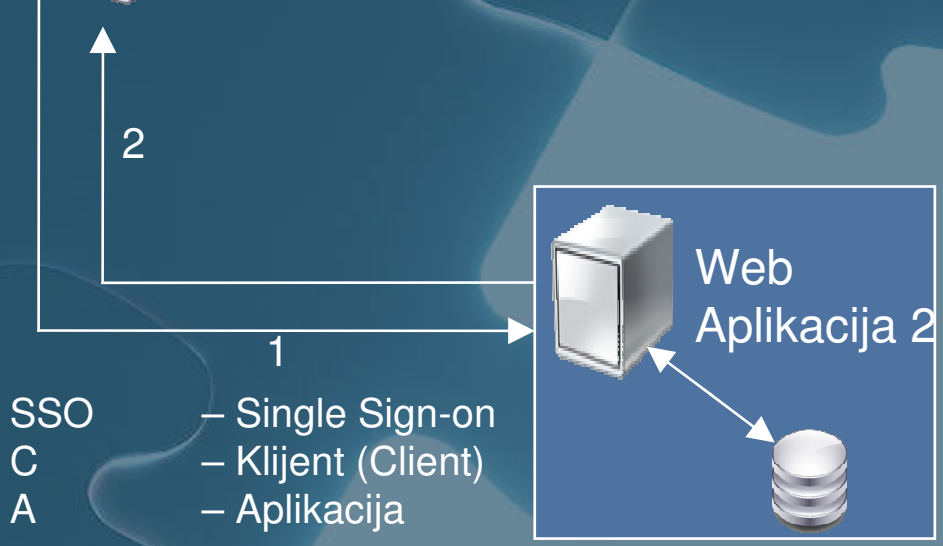
1

SSO – Single Sign-on
C – Klijent (Client)
A – Aplikacija

Arhitektura [SLEASY]

(Naknadne prijave u aplikacije)

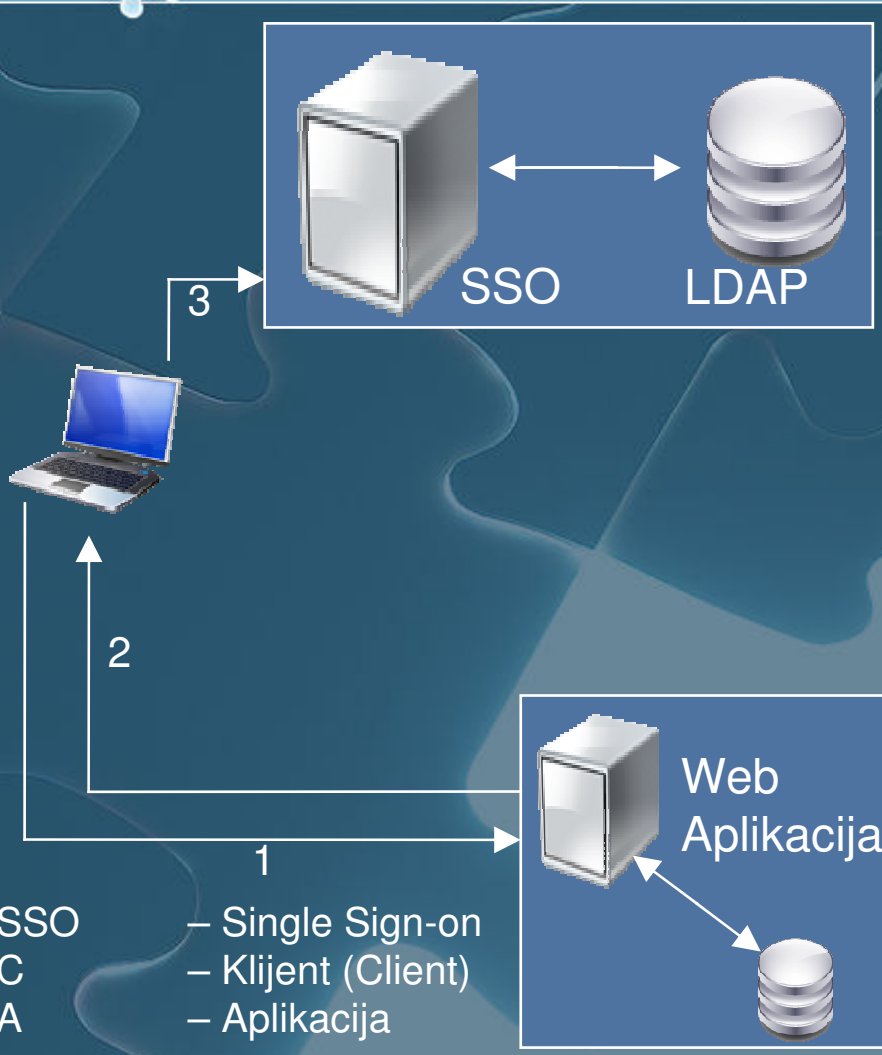
1. Korisnik zahtijeva pristup aplikaciji 2
2. Web preglednik je usmjeren na SSO poslužitelj (zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**



SSO – Single Sign-on
C – Klijent (Client)
A – Aplikacija

Arhitektura [SLEASY]

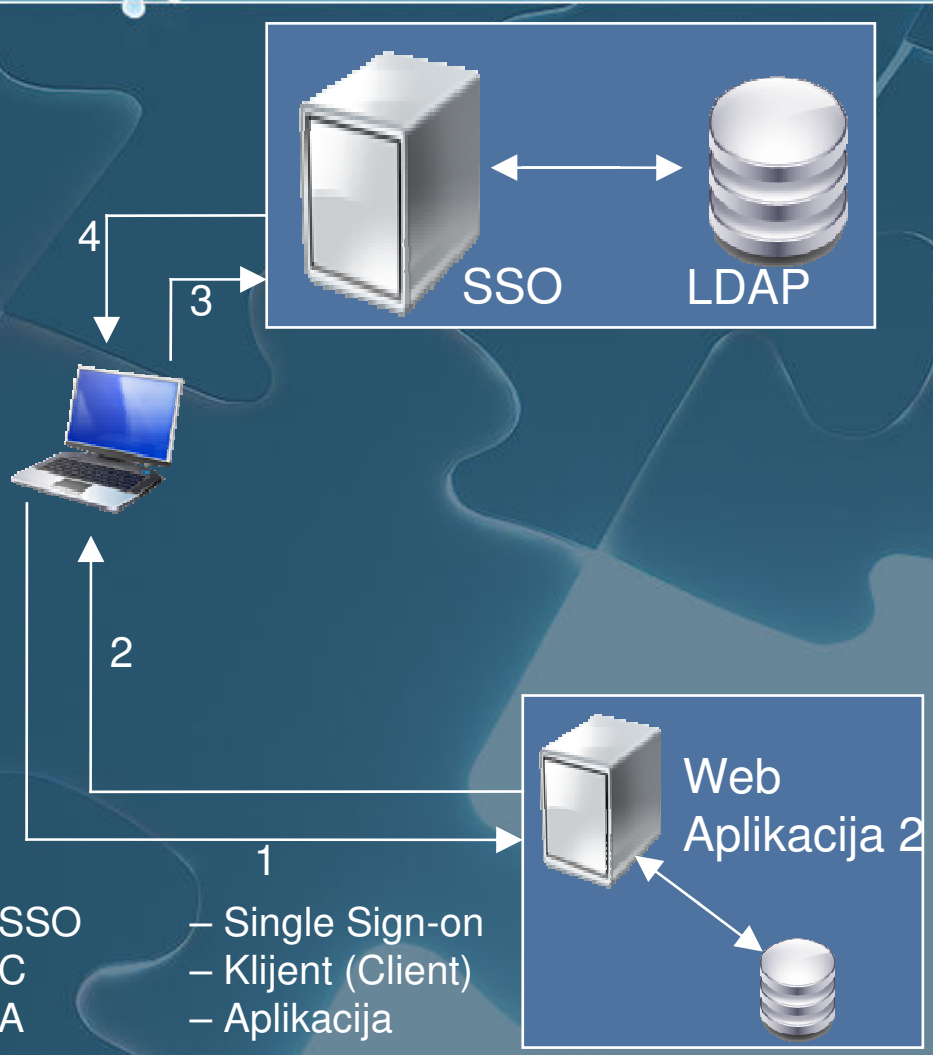
(Naknadne prijave u aplikacije)



1. Korisnik zahtijeva pristup aplikaciji 2
2. Web preglednik je usmjeren na SSO poslužitelj (zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju

Arhitektura [SLEASY]

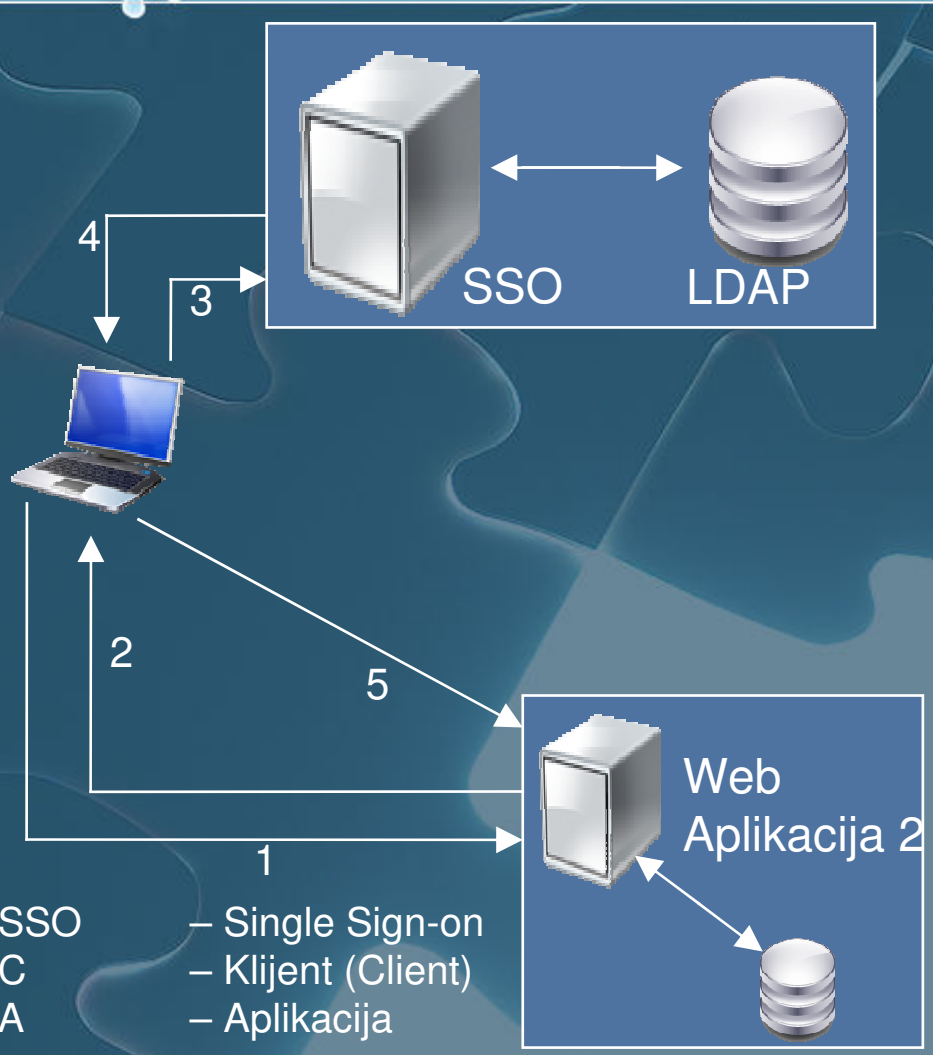
(Naknadne prijave u aplikacije)



1. Korisnik zahtijeva pristup aplikaciji 2
2. Web preglednik je usmjeren na SSO poslužitelj (zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. Budući da je korisnik već prijavljen, poruka se dešifrira te se kreira odgovor: **(Ka^{+} (Ks^{-} (poruka))**

Arhitektura [SLEASY]

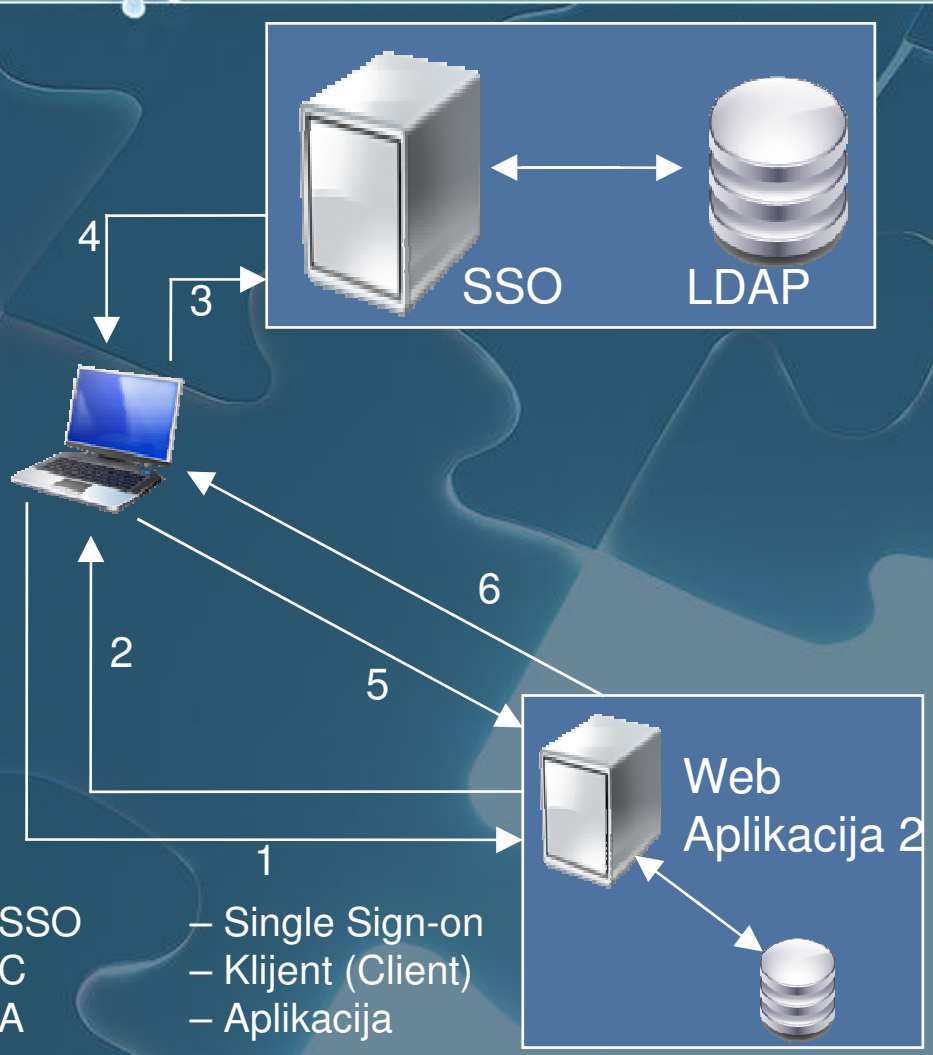
(Naknadne prijave u aplikacije)



1. Korisnik zahtijeva pristup aplikaciji 2
2. Web preglednik je usmjeren na SSO poslužitelj (zahtjev za autentikaciju: **Ks^{+} (Ka^{-} (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. Budući da je korisnik već prijavljen, poruka se dešifrira te se kreira odgovor: **(Ka^{+} (Ks^{-} (poruka))**
5. Web preglednik prosljeđuje kriptiranu poruku aplikaciji 2

Arhitektura [SLEASY]

(Naknadne prijave u aplikacije)



1. Korisnik zahtijeva pristup aplikaciji 2
2. Web preglednik je usmjeren na SSO poslužitelj (zahtjev za autentikaciju: **Ks^+ (Ka^- (poruka))**
3. Web preglednik pristupa SSO poslužitelju
4. Budući da je korisnik već prijavljen, poruka se dešifrira te se kreira odgovor: **(Ka^+ (Ks^- (poruka))**
5. Web preglednik prosljeđuje kriptiranu poruku aplikaciji 2
6. Nakon što je dešifrirala poruku, aplikacija 2 potvrđuje da se korisnik uspješno prijavio (ili ne)

Kako to stvarno radi

- Logika aplikacije

```
...
<head>
    <?php if (!prijavljen()){
      <script type="text/javascript" src="www.foi.hr/sso/validate.php" />
    <?php } ?>
</head>
...
```

JS validate.php

- Ako logiran na SSO
  - JS *validate.php* sadrži kriptiranu povratnu poruku
  - AJAX call: *login.php* (dekodiranje podataka i logiranje korisnika)
- Ako nije logiran SSO
  - JS *validate.php* sadrži naznaku da je potrebno izvršiti autentikaciju
  - JS redirekcija na *redirect.php*
    - kodira zahtjev za autentikaciju koji se proslijeđuje na SSO

Zaključak

- Identificirati potrebe
- Uzeti u obzir poslovna pravila i procese
- Integrirati sa svim direktorijima
- Podići cjelokupnu sigurnosnu politiku
- Utvrditi potrebu za revizijom (Audit)
- Uključiti korisnike u projekt
- Koristiti SSO kao polazište za buduće projekte vezane uz identitet i prava pristupa
- Koristite najjednostavnije rješenje (ono koje zadovoljava vaše potrebe)