Razvoj Web Aplikacija

Predavanje 6

JWT: JSON Web Token

- Što je JWT?
- Kako JWT radi?
- JWT i federacijski identitet
- Tokeni za osvježavanje
- Zaštićeni/nezaštićeni tokeni

JSON Web Token

Što je JWT?

JWT je kodiran podatak koji se prenosi između klijenta i poslužitelja.

Primjer:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpv
aG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

…što se dekodira u…

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Povijest

- JOSE (JSON Object Signing and Encryption) grupa je formirana 2011 godine
- Zadatak:
  - standardizirati mehanizam za zaštitu integriteta
  - koristiti standardnu enkripciju
  - koristiti standardni oblik
  - koristiti standardne algoritme
  - koristiti JSON
- Do 2013. godine objavili su nacrte za JWT, zajedno s JWS (potpis), JWE (enkripcija), JWK (ključ), JWA (algoritam)
- JWT ovisi o JWS i JWE
  - JWA je identifikator algoritma, pa JWT koristi i to
- Do 2016. godine razvoj je završio i JWT je postao standard

Za što se koristi JWT?

- Sustav prijave
- **Provjera autentičnosti** – provjera identiteta nekoga ili nečega, pazeći da je to onaj za kojeg se izdaje
- **Autorizacija** – ograničavanje pristupa određenim resursima
- **Federacijski identitet** – povezivanje identiteta kroz više sustava (provjera autentičnosti/autorizacije)
- **Klijentske sesije** – održavanje spremljenih podataka na više klijenata

Kako klijent i server koriste JWT?

- Klijent šalje korisničko ime i lozinku poslužitelju (login krajnja točka)
- Poslužitelj izrađuje JWT token s vlastitim potpisom i vraća ga pregledniku
- Prilikom izdavanja zahtjeva zaštićenom resursu, klijent poslužitelju šalje JWT token u zaglavlju autorizacije (engl. Authorization header)
- Poslužitelj provjerava JWT potpis
  - Ako je valjan, vraća podatke pregledniku
  - Ako nije valjan, vraća pogrešku pregledniku (npr. 401 Unauthorized)

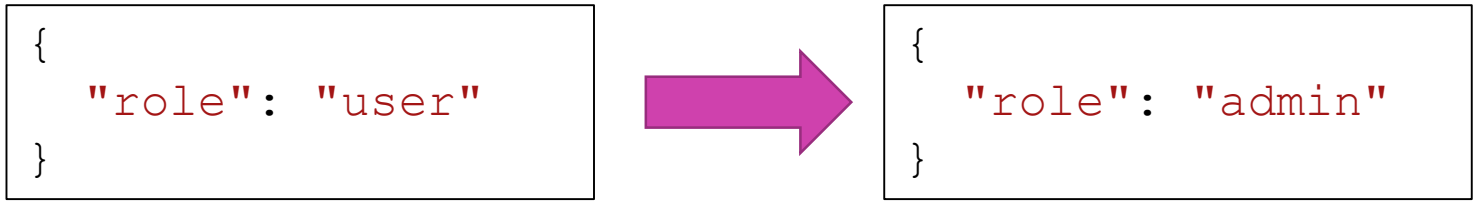
- Ovaj token se također naziva **JWT pristupni token** (engl. JWT Access Token)

Osobine JWT

- Koristi se u web tehnologijama
- Koristi standardni obrazac
- Kodiran base64 kodiranjem
- Sastoji se od tri dijela: **zaglavlja**, **tijela** i **potpisa**
- Dekodirani, zaglavlje i tvrdnje (claims) prikazuju **JSON** strukturu, dok je dekodirani potpis **binaran**
- Zaglavlje i tvrdnja (claim) ne koriste enkripciju, smatramo ih običnim tekstom, dok potpis koristi enkripciju
  - Zapravo se tvrdnje također *mogu* šifrirati

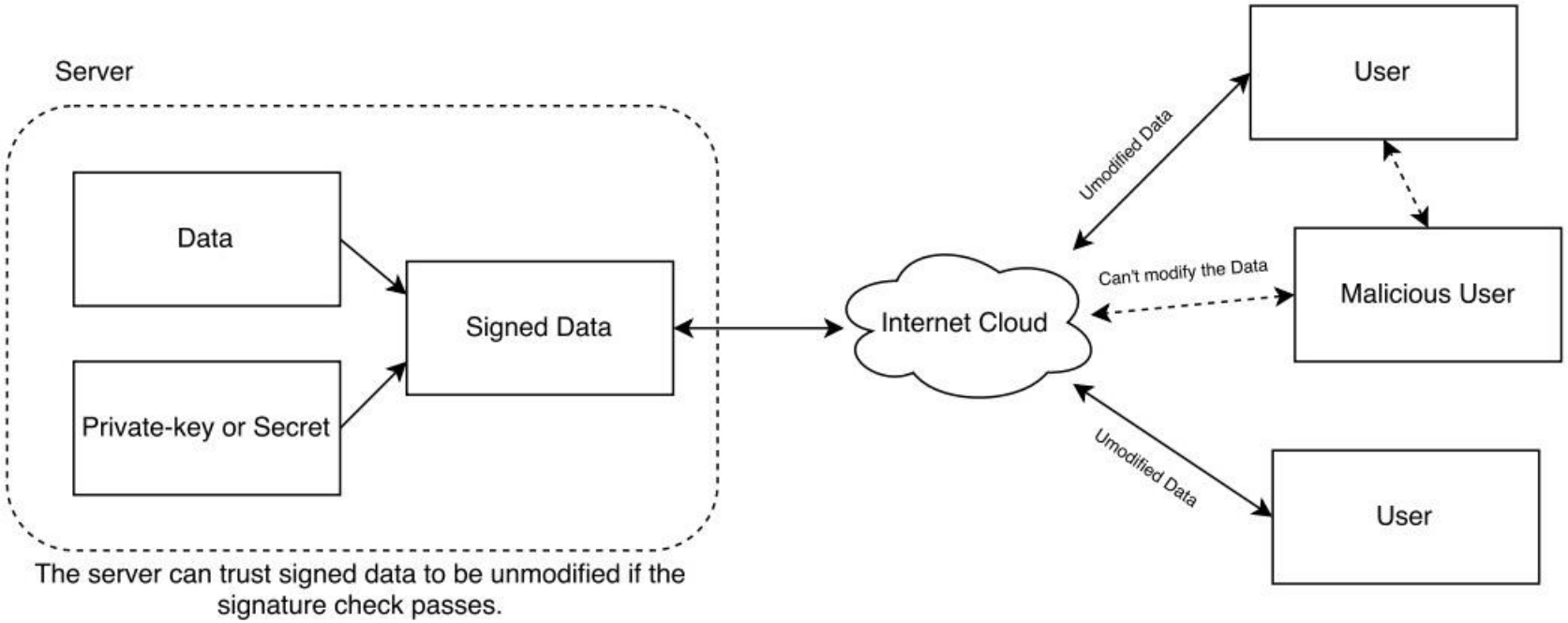
Primjer

Podaci s običnim tekstom poslani s poslužitelja da bi bili pohranjeni na klijentu mogu se lako lažirati



- Međutim, na temelju svog tajnog ključa i sadržaja poruke, poslužitelj može dodati kriptografsku vrijednost na sam sadržaj
- Ako to učinimo, sposobnost klijenta da lažira podatke odjednom se *drastično smanjuje*
- **Potpis** je binarni, a binarna vrijednost lako se kodira u base64:
  `04wRHoeP0SL7-IWcxX-KFt6fgXT8urkjy8vyEwB0Gbc`
- Taj base64 potpis se dodaje u payload i vraća natrag na poslužitelj kako bi ga isti mogao provjeriti
- Kažemo da su podaci zaštićeni od **neovlaštenog mijenjanja**
- Potpis **ne** štiti podatke od **čitanja treće strane**

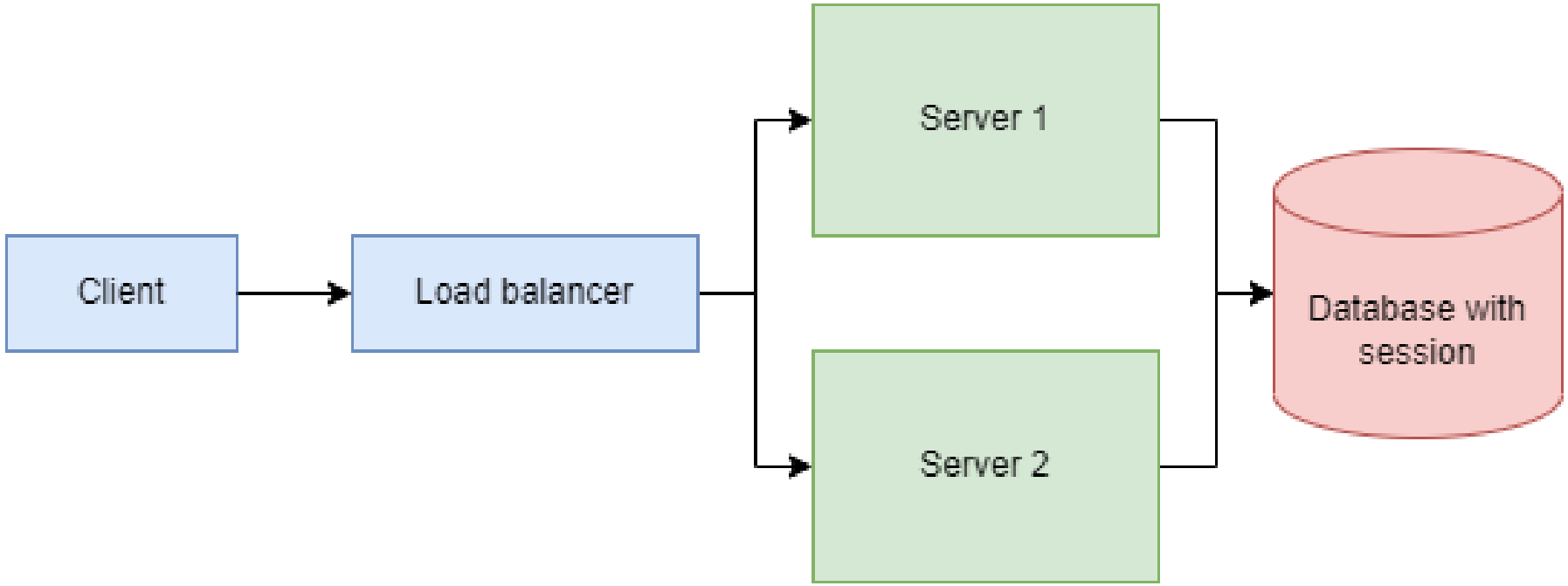
Kako je JWT osiguran potpisom?



JWT i sesije

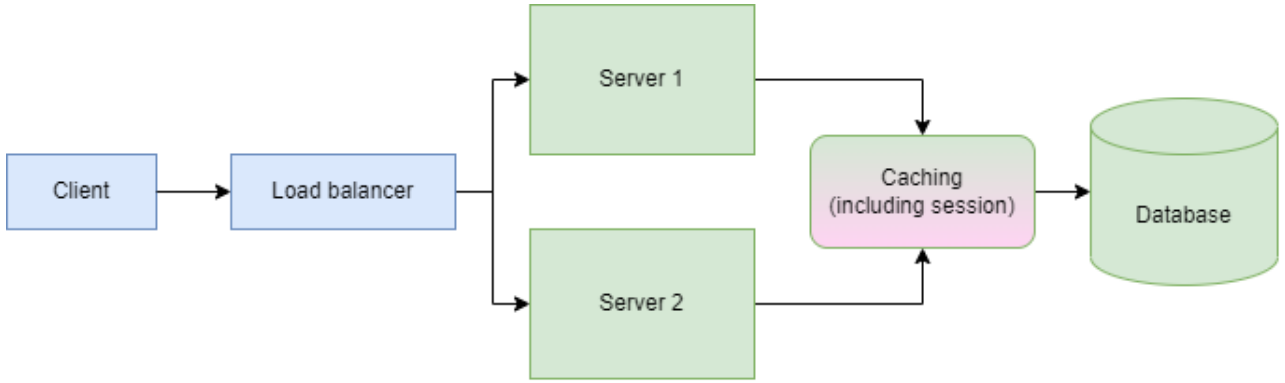
- Podaci za isti identitet se dijele između nekoliko klijenata
- Poznato kao klijentske sesije (nasuprot serverskim sesijama za koje može biti potrebna baza podataka)
- Problem: dijeljenje puno podataka uzrokuje "napuhavanje" komunikacije
- Rješenje: pronaći ravnotežu između onoga što je pohranjeno u klijentskoj sesiji (JWT) i onoga što je pohranjeno na poslužitelju (npr. baza podataka)

Serverska sesija – spremanje u bazu podataka

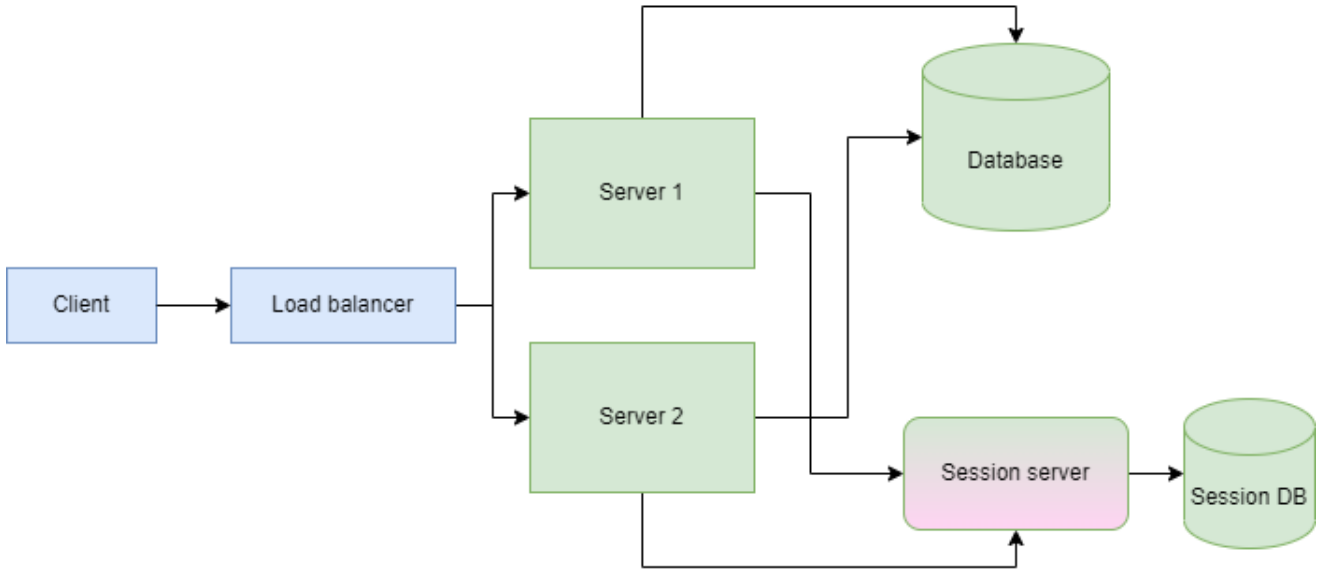


Serverska sesija – ublažavanje problema sa skalabilnošću

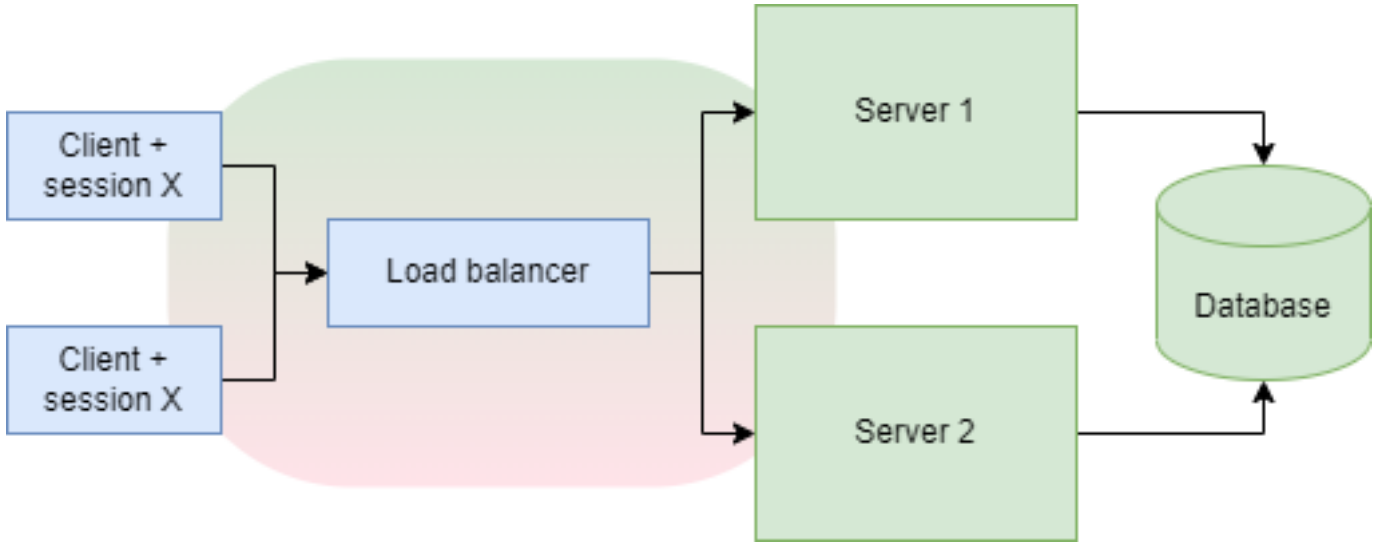
- **Rješenje 1** – uvesti predmemorirani (cached) sloj (npr. Redis)
- Predmemorirani sloj preuzima dodatno opterećenje, ali još uvijek postoje granice



- **Rješenje 2** – uvesti namjenski poslužitelj sesija koji brine o opterećenju
- Taj poslužitelj tada može postati usko grlo



Klijentska sesija i skaliranje

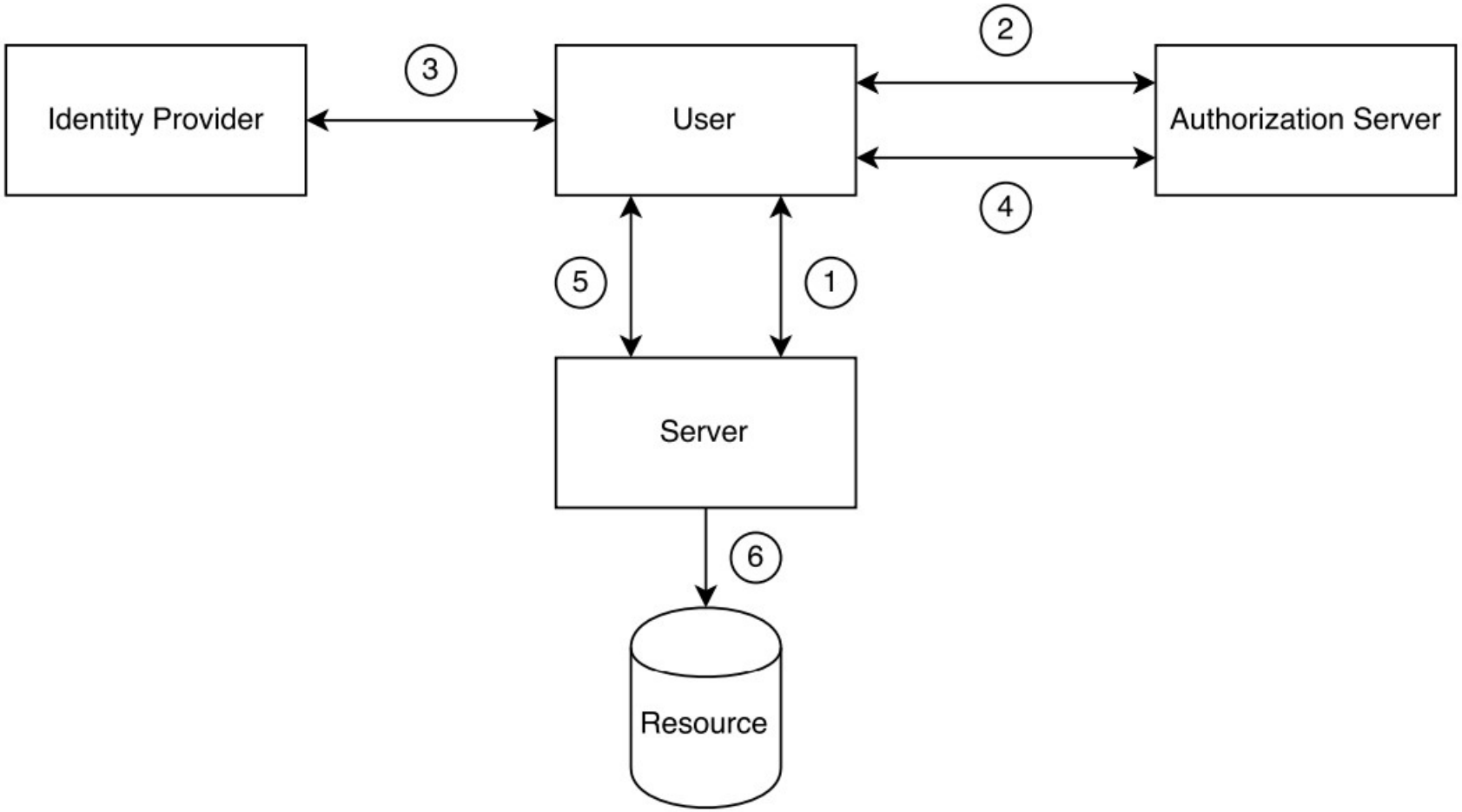


- **JWT rješenje** – na poslužitelju se ne održava sesija
- Više podataka šalje se natrag i s klijenta i poslužitelja radi održavanja sesije

Što je federacijski identitet?

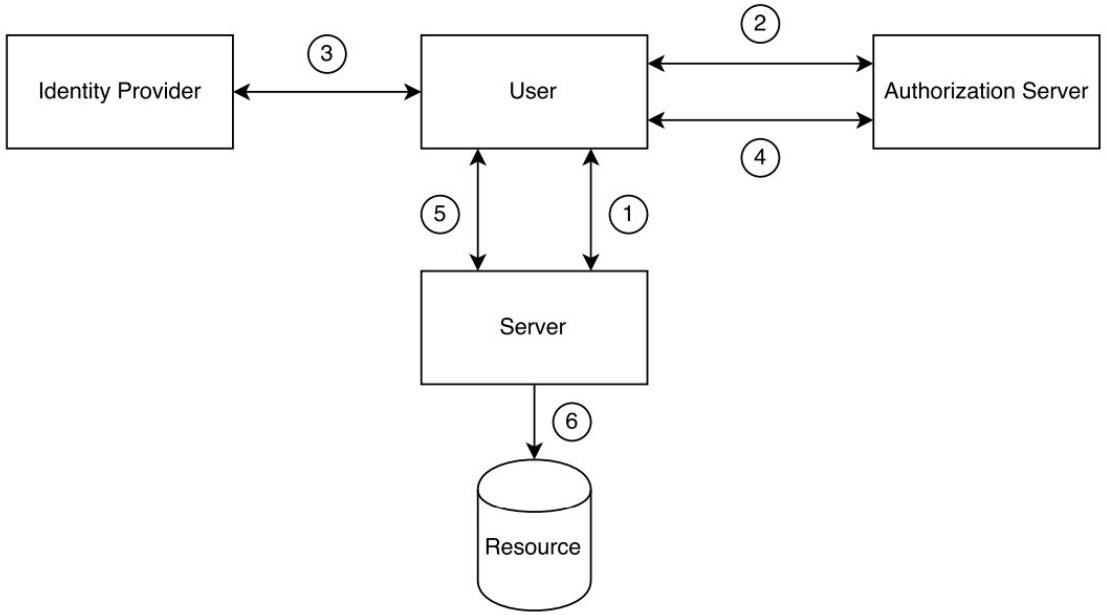
- Više sustava (aplikacija, entiteta) uključeno je u provjeru autentičnosti i/ili autorizaciju
- Omogućuje korisniku da se prijavi u sustav samo jednom, a zatim pristupi svim aplikacijama koje koriste uslugu jedinstvene prijave ("single sign-on")
  - SSO ➔ Ne moram ponovno unositi vjerodajnice (credentials) ☺
- Entiteti federacije:
  - Pružatelj identiteta/autorizacije – usluga koja omogućuje provjeru autentičnosti i autorizaciju
  - Pružatelj resursa – usluga koja pruža resurse
  - Krajnji korisnik ili klijent
- Usluga mora moći preusmjeriti korisnika na sljedeću u lancu

Federated identity



Federated identity flow

1. Korisnik pokušava pristupiti resursu kojeg kontrolira poslužitelj.
2. Korisnik nema odgovarajuće vjerodajnice za pristup resursu, pa poslužitelj preusmjerava korisnika na poslužitelj za autorizaciju. Poslužitelj za autorizaciju konfiguriran je tako da korisnicima omogućuje prijavu pomoću vjerodajnica kojima upravlja davatelj identiteta.
3. Poslužitelj za autorizaciju preusmjerava korisnika na stranicu za prijavu davatelja identiteta.



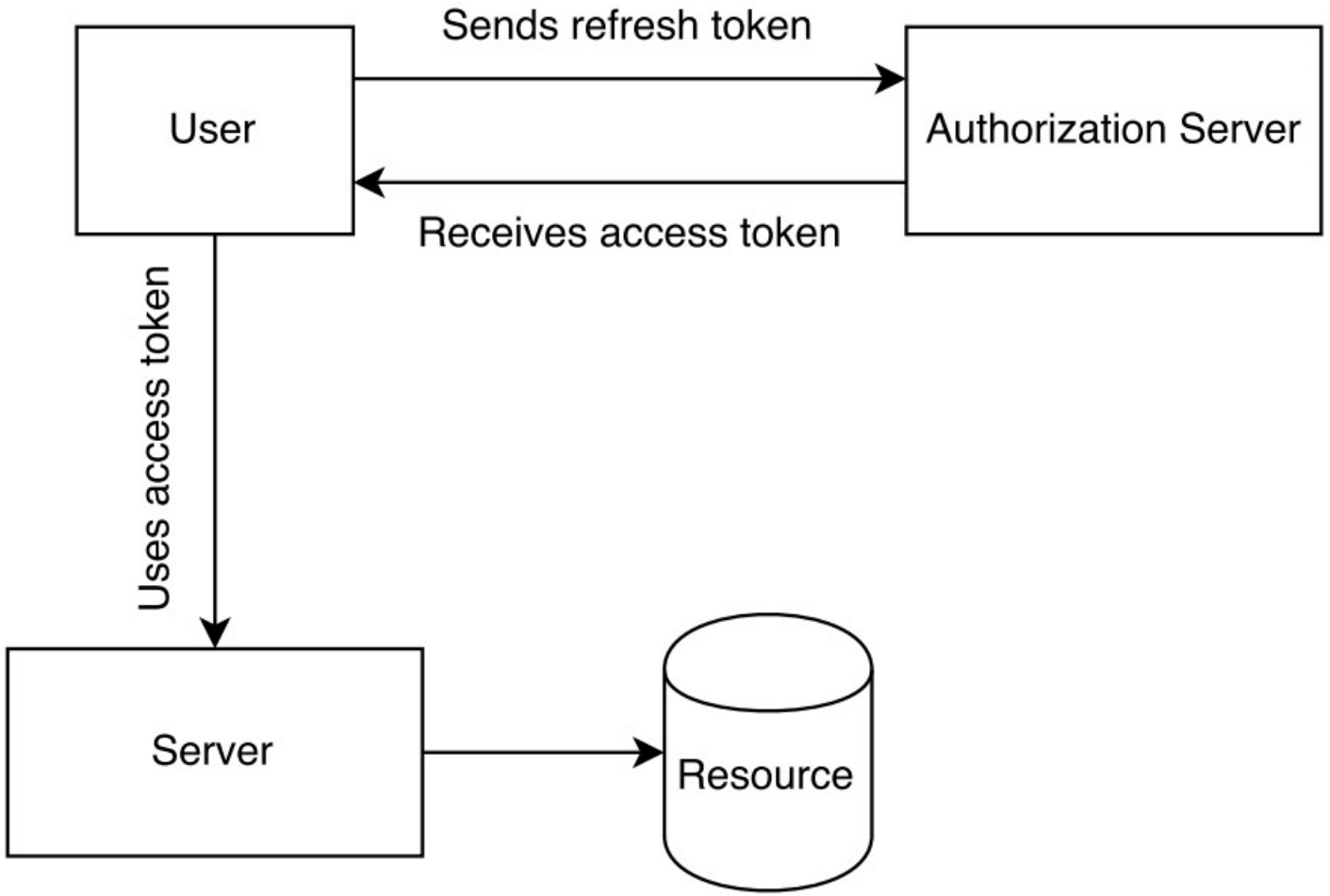
Federated identity flow

4. Korisnik se uspješno prijavljuje i preusmjerava ga se na poslužitelj autorizacije. Poslužitelj za autorizaciju koristi vjerodajnice davatelja identiteta za pristup vjerodajnicama koje zahtijeva poslužitelj resursa.
   ➔ JWT
5. Poslužitelj za autorizaciju preusmjerava korisnika na poslužitelj resursa. Zahtjev sada ima ispravne vjerodajnice potrebne za pristup resursu.
   ➔ JWT
6. Korisnik uspješno dobiva pristup resursu.

Tokeni za osvježavanje (refresh tokens)

- **Pristupni token (access token)** – oni tokeni o kojima smo do sada pričali
- Trebali bi imati **kratak život** – zašto?
  - Za razbijanje enkriptiranog tokena potrebno je dovoljno vremena – ako je život tokena kratak, napadač nema dovoljno vremena za razbijanje šifre
  - Da bismo prisilili klijenta na ažuriranje podataka – klijent može odlučiti koristiti token dok ne istekne i ne tražiti nove podatke s poslužitelja, a ako je vijek trajanja pristupnog tokena kratak, klijent uskoro mora zatražiti novi token i tako dobiti nove podatke
- **Tokeni za osježavanje (refresh tokens)** – klijentu omogućuju osvježavanje pristupnih tokena
  - Ako pristupni token istekne i nema tokena za osvježavanje, korisnik mora ponovno unijeti svoje korisničko ime i lozinku kako bi dobio novi token
  - Ako pristupni token istekne i tokena za osvježavanje je raspoloživ, klijent može dobiti novi pristupni token na temelju tokena za osvježavanje, bez ponovnog unosa korisničkog imena i lozinke
- Stoga ima smisla da je **život tokena za osvježavanje dug**

Tokeni za osvježavanje (refresh tokens)



JWT tokeni - zaglavlje

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.
TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

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

- Sadrži tvrdnje o sebi
- Tvrdnja **typ** – treba bi biti **JWT** ako se koristi šifriranje, a inače je ne treba biti (nezaštićeni token)
- Tvrdnja **alg** – ovisi o kriptografskom algoritmu koji se koristi za potpisivanje tokena
  - RS256 – RSA (RSASSA) sa SHA-256
  - ES256 – ECDSA sa SHA-256
  - HS256 – HMAC sa SHA-256
  - EdDSA
  - …

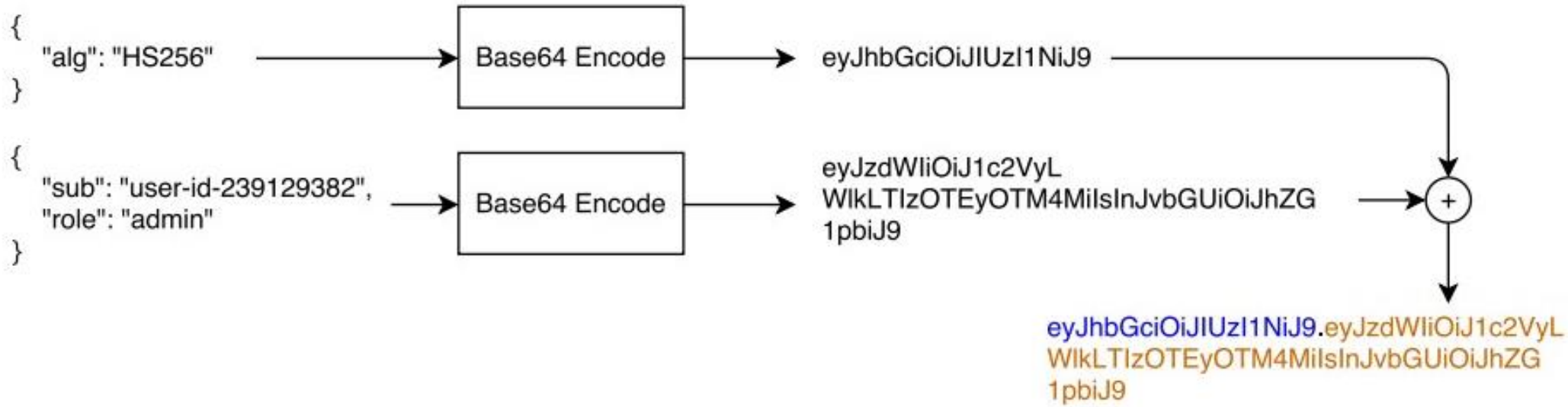
JWT tokeni - tijelo

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.
TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

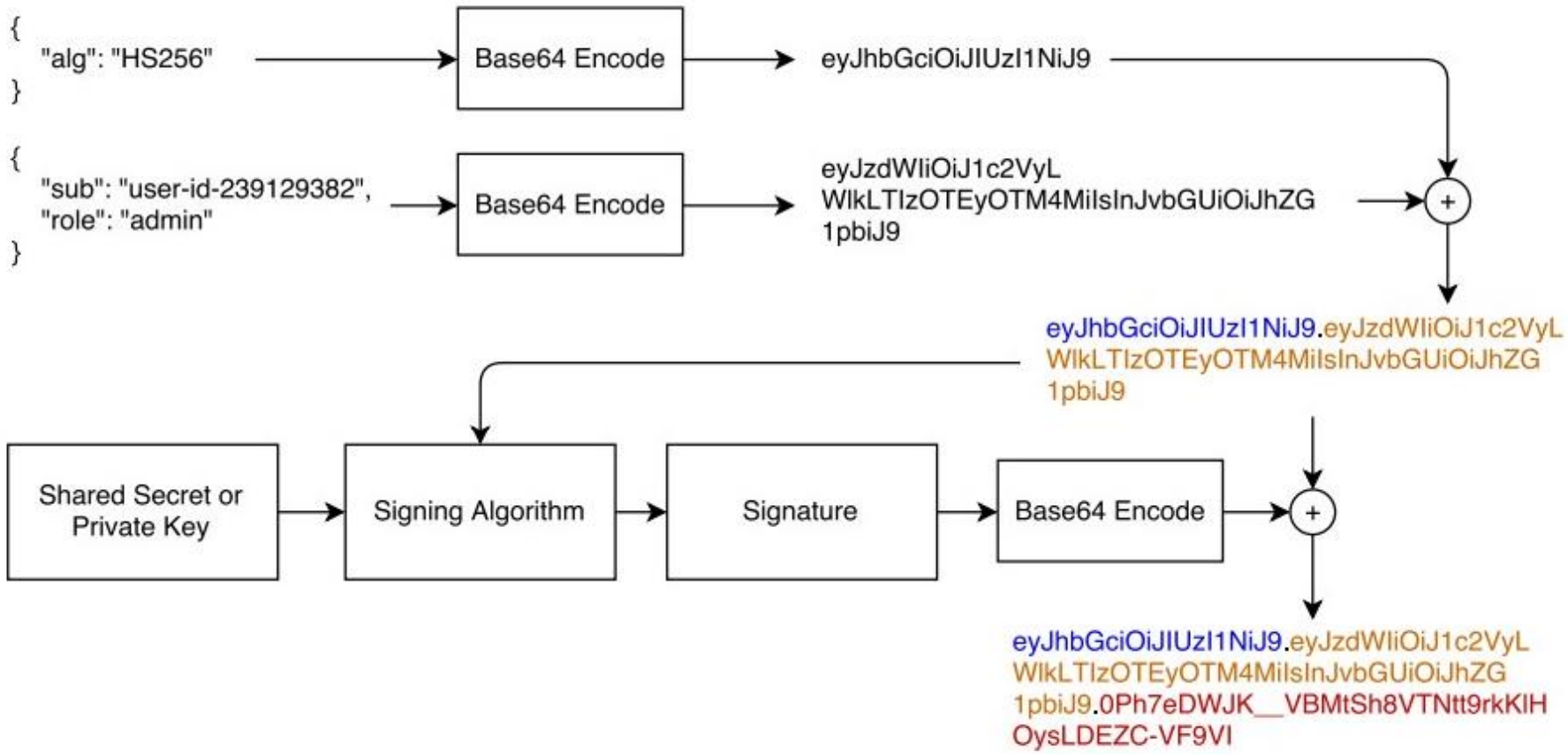
```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

- Sadrži **registrirane** tvrdnje
- **iss**: izdavatelj (issuer), case-sensitive tekst ili URI koji jedinstveno identificira stranu koja je izdala JWT
- **sub**: subjekt, case-sensitive tekst ili URI koja jedinstveno identificira stranku o kojoj taj JWT nosi informacije (identitet)
- **aud**: publika (audience), case-sensitive tekst ili URI koji jedinstveno identificira predviđene primatelje ovog JWT-a
- **exp**: istek (expiration), broj koji predstavlja određeni datum i vrijeme u "epoch" obliku (sekunde)
- **nbf**: ne prije (not before), broj koji predstavlja određeni datum i vrijeme u "epoch" obliku (sekunde) od kojeg nadalje se ovaj JWT smatra valjanim
- **iat**: vrijeme izdavanja (issued at), broj koji predstavlja određeni datum i vrijeme izdavanja ovog JWT-a
- **jti**: JWT ID, tekst koji predstavlja jedinstveni identifikator za ovaj JWT
- Može sadržavati i **neregistrirane** tvrdnje koje aplikacija razumije (npr. **name**)

JWT tokeni – nezaštićeni token



JWT tokeni – zaštićeni token



Hvala na pažnji!