SYNCLAB

Spring Security
Oauth 2

Cos'e oauth2

- Un protocollo per gestire le autorizzazioni attraverso un token. Non e un protocollo di authenticazione, e tipicamente si usa con OpenID Connect.

  Cos'e OpenID Connect?
  OpenID Connect 1.0 è un semplice livello di identità in cima al protocollo OAuth 2.0. Consente ai Clienti di verificare l'identità dell'Utente finale in base all'autenticazione eseguita da un Server di autorizzazione, nonché di ottenere informazioni di base sul profilo dell'Utente finale in modo interoperabile e simile a REST.
  OpenID Connect consente a client di tutti i tipi, inclusi client Web, mobili e JavaScript, di richiedere e ricevere informazioni sulle sessioni autenticate e sugli utenti finali. La suite di specifiche è estensibile, consentendo ai partecipanti di utilizzare funzionalità opzionali come la crittografia dei dati di identità, l'individuazione dei provider OpenID e la gestione delle sessioni, quando ha senso per loro.

  In che modo OpenID Connect è diverso da OpenID 2.0?
  OpenID Connect esegue molte delle stesse attività di OpenID 2.0, ma lo fa in un modo che sia API-friendly e utilizzabile da applicazioni native e mobili. OpenID Connect definisce meccanismi opzionali per una firma e una crittografia robuste. Mentre l'integrazione di OAuth 1.0 e OpenID 2.0 richiedeva un'estensione, in OpenID Connect le funzionalità di OAuth 2.0 sono integrate con il protocollo stesso.

Cos'e oauth2

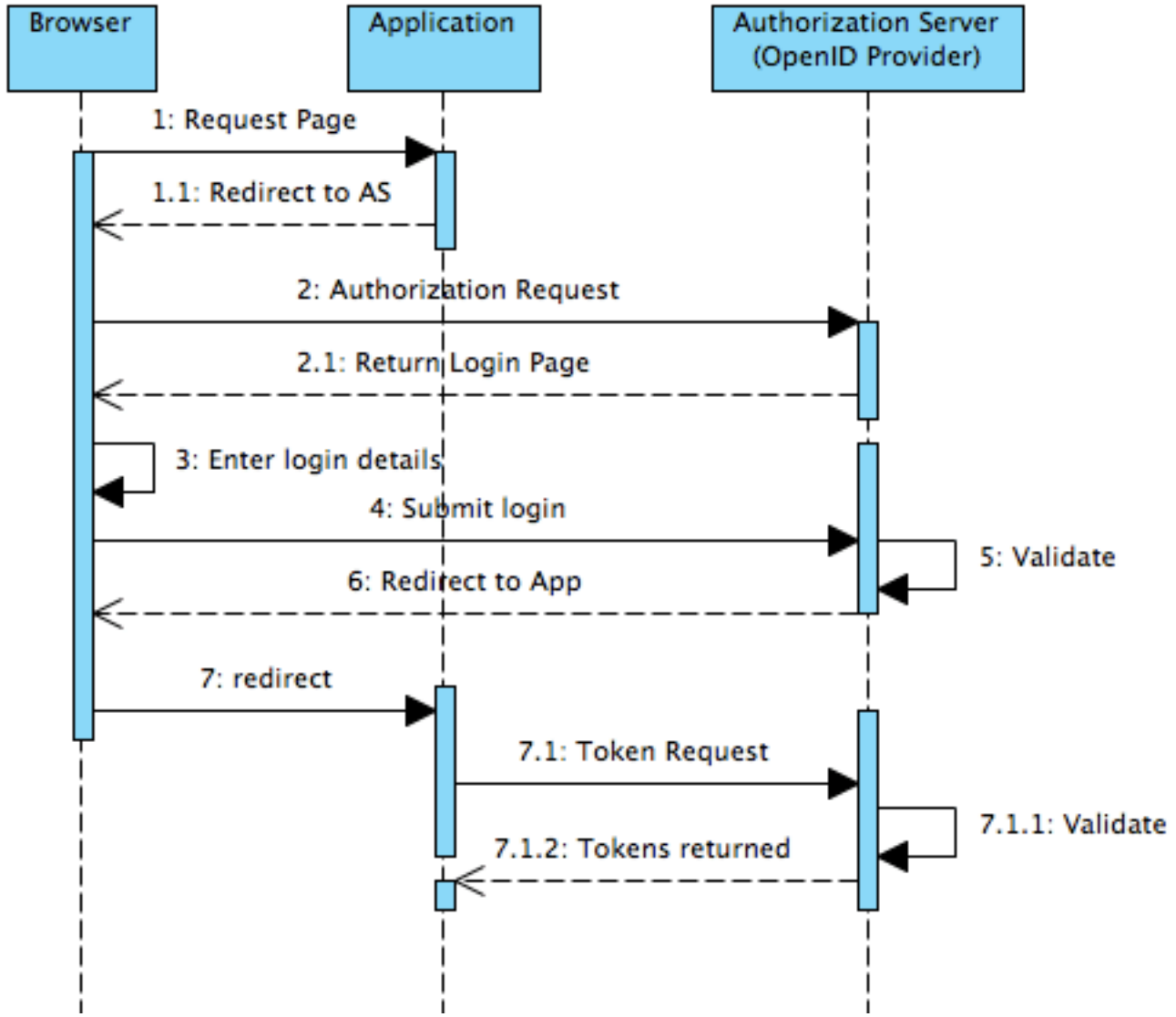
- Standardizzazione nel ottenere il token. Ci sono 4 tipi di concessione:
  - Authorization code
  - Password
  - Implicit
  - Client Credentials

- Separazione dell'applicazione client dal proprietario della risorsa
  - E il proprietario della risorsa a concedere i 'permessi' ad un client

Cosa NON e oauth2

- Non e un protocollo di autenticazione
  - L'utente deve essere autenticato per ottenere un token
  - Come l'utente si autentica e fuori dalle specifiche
  - Come il token viene validato e fuori dalle specifiche
  - Cosa il token continene e fuori dalle specifiche
- OpenID Connect gestisce la parte di autenticazione

OpenID Connect flow



Perche Oauth2

- devi solo autenticare la richiesta una volta per utente
- Se l'utente non ha una sessione
  - Controlle le credenziali dell'utente
  - Creo una sessione per l'utente
  - Fornisco un accesso basato sul ruolo
- Se l'utente ha gia una sessione
  - Verifico che la sessione non sia scaduta
- Le chiamate sono verificate

Oauth2: Pro e contro

- Pro
  - Riduzione dei servizi in caso di un attacco

- Contro
  - una volta fornite le autorizzazioni si possono usare per il resto dell'applicazione incluso l'accesso al database. Se venissero rubate l'intera applicazione potrebbe essere in pericolo

Oauth2: micro services

- Single sign on (SSO)
  - SSO su tutti i nostri servizi
- Stateless
  - I servizi di backend non devono salvare le credenziali di accesso
  - I servizi di backend non devono salvare le sessioni degli utenti
- L'accesso viene delegato
- Le credenziali vengono gestite solo dal server di autenticazione
- Gestione granulare delle authority dei vari servizi
- Compatibile con tutti i client
  - browser, mobile, services

Oauth2: Authorization code flow