OAuth 1.0a und OAuth 2

Am Beispiel von Atlassian-Anwendungen

Disclaimer: Dieser Talk behandelt das Thema nur oberflächlich.

Allgemein

- Standard zur Autorisierung von Zugriffen auf Ressourcen
- OAuth (seit 2006, veröffentlicht 2007)
  - RFC-5849
- OAuth 2 (RFC veröffentlicht 2012)
  - RFC-6749
  - vereinfacht den Zugriff
  - Signierung fällt weg, Verantwortlichkeit liegt bei TLS
  - Scopes
  - Refresh Tokens

Begriffe

OAuth 1.0a	OAuth 2 (Rollen)	Beispiel
User	Resource owner	Mein Jira-Account
Service Provider	Resource server	Jira
Consumer	Client	Meine App
---	Authorization Server	Jira, Crowd

Begriffe

OAuth 1.0a	OAuth 2	Kontext
Consumer Key und Secret	Client Id und Client Secret	Anwendung
Request Token und Secret	---	Benutzer
Access Token und Secret	Access Token	Session
---	Scope	Resource

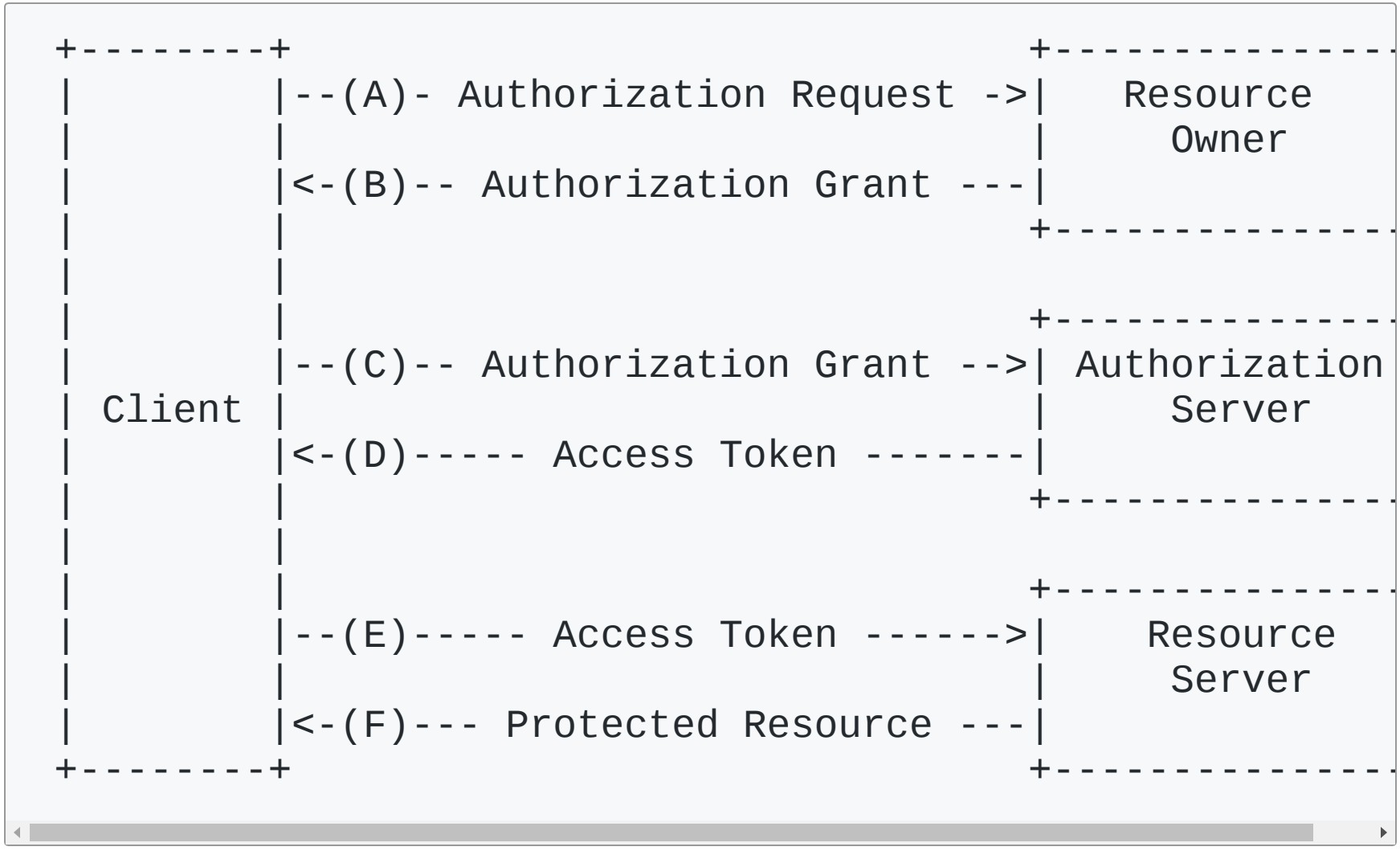
OAuth 1.0a - 2-legged vs. 3-legged

- 3-legged beinhaltet Access Token und Secret
  - wird z. B. bei Web-Apps eingesetzt. Consumer Key/Secret hinterlegt der Betreiber der Web-App, Access Token und Secret der Benutzer der Web-App.
- 2-legged hat **kein** Access Token und Secret
  - wird z. B. bei mobilen Twitter-Clients eingesetzt

OAuth 1.0a - Workflow

Siehe http://oauthbible.com/

OAuth2 - Workflow



siehe https://tools.ietf.org/html/rfc6749

Alles nicht so einfach

- OAuth 1.0a Workflow komplex
- Art und Weise der Datenübergabe abhängig vom Service Provider / Resource Server (Header, Query-Parameter)
- Workflow hängt vom Service Provider / Resource Server ab (Twitter)
- Spring: Konfiguration von OAuth 2 deutlich einfacher

Beispiel

- Zugriff auf Atlassian-Dienste via OAuth 1.0a
  - OAuth2 (mit JWT) nur für Connect-Anwendungen :-/
- Atlassian bietet Beispielquellcode an
- jira-rest-client unterstützt nur Basic Auth

getRequestToken

```
public TokenSecretVerifierHolder getRequestToken() {
        // provider enthält u.a. consumerKey
        // und consumerSecret und erzeugt den OAuthAccesor
        OAuthAccessor accessor = provider.getAccessor();
        OAuthClient oAuthClient = createOAuthClient();
        List<OAuth.Parameter> callBack;

        callBack = ImmutableList
                                .of(new OAuth.Parameter(OA

        OAuthMessage message = oAuthClient.getRequestToker

        TokenSecretVerifierHolder tokenSecretVerifier = ne
                                accessor.tokenSecret, mess

        return tokenSecretVerifier;
}
```

swapRequestToken

```
public String swapRequestTokenForAccessToken(String reques
        OAuthAccessor accessor = provider.getAccessor();
        OAuthClient oAuthClient = createOAuthClient();
        accessor.requestToken = requestToken;
        accessor.tokenSecret = tokenSecret;
        OAuthMessage message = oAuthClient.getAccessToken(
                                ImmutableList.of(new OAuth

        return message.getToken();
}
```

makeAuthenticatedRequset

```
public String makeAuthenticatedRequest(String url, String
        OAuthAccessor accessor = provider.getAccessor();
        OAuthClient oAuthClient = createOAuthClient();
        accessor.accessToken = accessToken;
        OAuthMessage response = oAuthClient.invoke(accesso

        return response.readBodyAsString();
}
```

Aufruf

```
AtlassianOAuthProvider atlassianOAuthProvider = new Atlass
        "http://jira.atlassian.net",
        "my-consumer-key",
        "my-applications-private-key",
        "http://localhost" /* or null */);

        AtlassianOAuthClient client = new AtlassianOAuthCl
        TokenSecretVerifierHolder tokenSecretVerifierHolde

        // present user with the redirect
        System.out.println(
                        "Go to " + atlassianOAuthProvider.
        // after user has accepted, continue

        String accessToken = client.swapRequestTokenForAcc
                        tokenSecretVerifierHolder.getSecre

        System.out.println("Storing your access token '" +
                        + "' in the database. Token is use
                        + atlassianOAuthProvider.getBaseUr
```

Fazit

- Wenn möglich, OAuth 2 einsetzen
- Workflow ausdrucken
- RFC lesen, Doku lesen
- Workflow ausdrucken