schülerVZ studiVZ meinVZ

OAuth

Ein offener Standard für die sichere Autentifizierung in APIs

Max Horváth, Andre Zayarni, Bastian Hofmann

schülerVZ studiVZ meinVZ

Vorstellung der Speaker

schülerVZ studiVZ meinVZ

Was ist OAuth?

Was ist OAuth?

- OAuth ermöglicht dem Endnutzer einer Webanwendung ...

Was ist OAuth?

- OAuth ermöglicht dem Endnutzer einer Webanwendung:
  - private Ressourcen mit anderen Webanwendungen auszutauschen

Was ist OAuth?

- OAuth ermöglicht dem Endnutzer einer Webanwendung
  - private Ressourcen mit anderen Webanwendungen auszutauschen
  - ohne dabei Nutzername und Passwort der Webanwendung der Konsumenten-Applikationen preiszugeben.

Was ist OAuth?

- OAuth unterstützt schwerpunktmäßig einen zentralisierten Autorisierungs- und Datenaustauschprozess.

Was ist OAuth?

- OAuth unterstützt schwerpunktmäßig einen zentralisierten Autorisierungs- und Datenaustauschprozess.

- Wie sich der Nutzer (etwa über einfaches Login mit Name und Passwort) autorisieren lässt, spielt eine untergeordnete Rolle.

Was ist OAuth?

- Es hat seinen Ursprung in bekannten Autorisierungsmechanismen bekannter APIs (beispielsweise Google, Yahoo und Amazon).

Was ist OAuth?

- OAuth muss keine Authentifizierung des Endnutzers direkt anbieten.

Was ist OAuth?

- OAuth muss keine Authentifizierung des Endnutzers direkt anbieten.

- OAuth stellt ausschließlich einen Autorisierungs- und Datenaustausch-Mechanismus zur Verfügung.

schülerVZ studiVZ meinVZ

Wie funktioniert OAuth?

schülerVZ studiVZ meinVZ

Wie funktioniert OAuth?

- OAuth zu verwenden ist mit der Bezahlung per EC-Karte zu vergleichen.

Wie funktioniert OAuth?

- Möchte man in einem Supermarkt mit der EC-Karte bezahlen.

Wie funktioniert OAuth?

- Möchte man in einem Supermarkt mit der EC-Karte bezahlen.

- Der PIN-Code wird dem Kellner nicht genannt.

schülerVZ studiVZ meinVZ

Wie funktioniert OAuth?

- Stattdessen tippt der Karteninhaber den PIN-Code direkt in das Eingabegerät des Kartenlesers ein.

schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

Rollen und Begriffe in OAuth

- Service-Provider

Rollen und Begriffe in OAuth

- Service-Provider
- Endnutzer

Rollen und Begriffe in OAuth

- Service-Provider
- Endnutzer
- Konsument

Rollen und Begriffe in OAuth

- Service-Provider
- Endnutzer
- Konsument
- Private Ressourcen

Rollen und Begriffe in OAuth

- Service-Provider
- Endnutzer
- Konsument
- Private Ressourcen
- Token

Rollen und Begriffe in OAuth

- **Service-Provider**
  - Stellt seine private Ressourcen zur Verfügung, auf die sich von anderen Anwendungen zugreifen lässt.

schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

- **Service-Provider**
  - Stellt seine private Ressourcen zur Verfügung, auf die sich von anderen Anwendungen zugreifen lässt.
  - Muss nicht gleichzeitig derjenige sein, der eine Authentifizierung des Endnutzers durchführt.

Rollen und Begriffe in OAuth

- **Endnutzer**
  - Besitzt private Ressourcen, die er, nachdem er zugestimmt hat, anderen Webanwendungen zur Verfügung stellen will.

Rollen und Begriffe in OAuth

- **Konsument**
  - Ist eine Applikation, die einen bestimmten Service oder private Endnutzer-Ressourcen eines Service-Providers in Anspruch nehmen möchte.

Rollen und Begriffe in OAuth

- **Konsument**
  - Ist eine Applikation, die einen bestimmten Service oder private Endnutzer-Ressourcen eines Service-Providers in Anspruch nehmen möchte.
  - Kann unterschiedliche Formen annehmen.

Rollen und Begriffe in OAuth

- **Private Ressourcen**
  - Private Ressourcen können in unterschiedlichen Formen auftreten.

Rollen und Begriffe in OAuth

- **Private Ressourcen**
  - Private Ressourcen können in unterschiedlichen Formen auftreten:
    - Daten

schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

- **Private Ressourcen**
  - Private Ressourcen können in unterschiedlichen Formen auftreten:
    - Daten
    - Aktivitäten

Rollen und Begriffe in OAuth

- **Private Ressourcen**
  - Private Ressourcen können in unterschiedlichen Formen auftreten:
    - Daten
    - Aktivitäten
    - Einfache URL.

Rollen und Begriffe in OAuth

- **Token**
  - Verwendet man statt des Nutzernamens und des Passworts, um Autorisierung auf privaten Ressourcen zu ermöglichen.

Rollen und Begriffe in OAuth

- **Token**
  - Verwendet man statt des Nutzernamens und des Passworts, um Autorisierung auf privaten Ressourcen zu ermöglichen.
  - Besteht meistens aus einem zufälligen String (Buchstaben und Zahlen).

schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

- **Token**
  - Ist eindeutig und schwer zu erraten. Wird zudem mit einem Konsumenten-Geheimnis kombiniert.

Rollen und Begriffe in OAuth

- **Token**
  - OAuth verwendet zwei unterschiedliche Token-Typen.

schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

- **Token**
  - OAuth verwendet zwei unterschiedliche Token-Typen:
    - Anfrage-Token

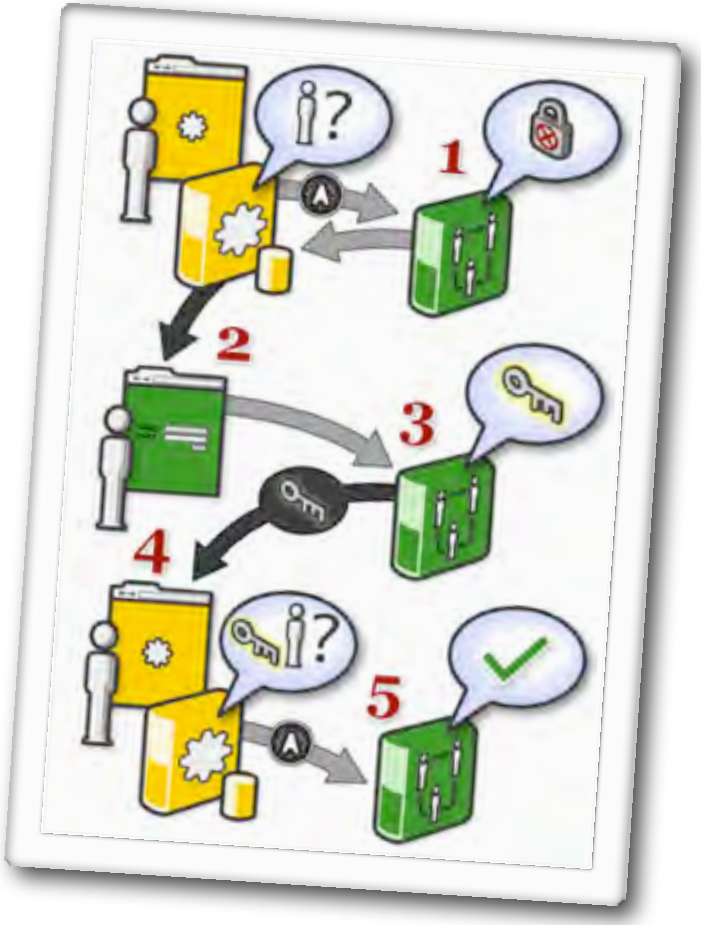
schülerVZ studiVZ meinVZ

Rollen und Begriffe in OAuth

- **Token**
  - OAuth verwendet zwei unterschiedliche Token-Typen:
    - Anfrage-Token
    - Zugriffs-Token.

schülerVZ studiVZ meinVZ

Anwendungsszenarien



Anwendungsszenarien

- Es gibt zwei typische Anwendungen für OAuth.

Anwendungsszenarien

- Es gibt zwei typische Anwendungen für OAuth:
  - Dreibeiniger OAuth-Typ

Anwendungsszenarien

- Es gibt zwei typische Anwendungen für OAuth:
  - Dreibeiniger OAuth-Typ
  - Zweibeiniger OAuth-Typ.

Dreibeiniger OAuth-Typ

- Ermöglicht eine Interaktion mit Endnutzern.

Dreibeiniger OAuth-Typ

- Ermöglicht eine Interaktion mit Endnutzern.
- Service-Provider, Konsument und Endnutzer stellen die drei Elemente dar, die diesen Typ ausmachen.

schülerVZ studiVZ meinVZ

Dreibeiniger OAuth-Typ

- Ermöglicht eine Interaktion mit Endnutzern.
- Service-Provider, Konsument und Endnutzer stellen die drei Elemente dar, die diesen Typ ausmachen.
- Er ist für Datenaustausch und Webservice-Aufrufe zwischen Webanwendungen mit der Zustimmung des Endnutzers gedacht.

schülerVZ studiVZ meinVZ

Zweibeiniger OAuth-Typ

- OAuth ohne Endnutzeraktivitäten.

Zweibeiniger OAuth-Typ

- OAuth ohne Endnutzeraktivitäten.

- Es geht ausschließlich um eine Maschine-zu-Maschine-Kommunikation, ohne Interaktion von Endnutzern.

Zweibeiniger OAuth-Typ

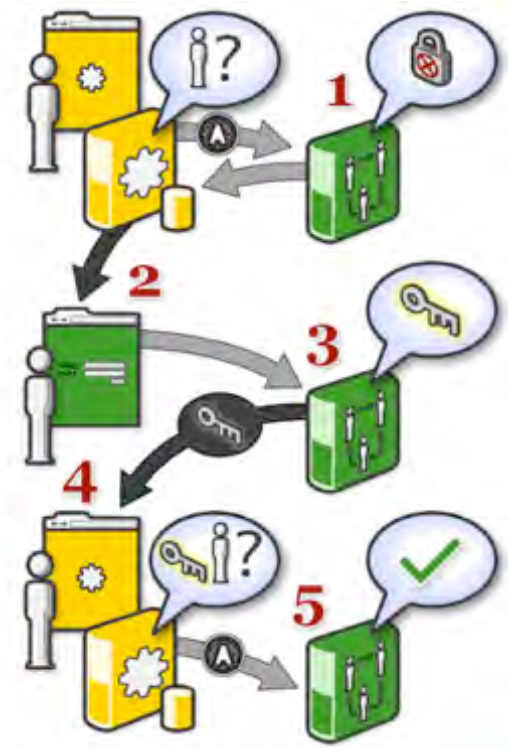
- Bei diesem Typ ist keine Authentifizierung des Endnutzers nötig.

Zweibeiniger OAuth-Typ

- Bei diesem Typ ist keine Authentifizierung des Endnutzers nötig.
- Der Typ ist für einen sicheren Zugriff über Webservices gedacht.

schülerVZ studiVZ meinVZ

Ablauf vom dreibeinigen OAuth



schülerVZ studiVZ meinVZ

Ablauf vom dreibeinigen OAuth

- Bevor eine Konsumenten-Anwendung auf einem Service-Provider zugreifen kann, ist sie zuvor beim ihm anzumelden.

schülerVZ studiVZ meinVZ

Ablauf vom dreibeinigen OAuth

- Bevor eine Konsumenten-Anwendung auf einem Service-Provider zugreifen kann, ist sie zuvor beim ihm anzumelden.

- Anschließend erhält die Konsumenten-Applikation Konsumentenschlüssel und -geheimnis.

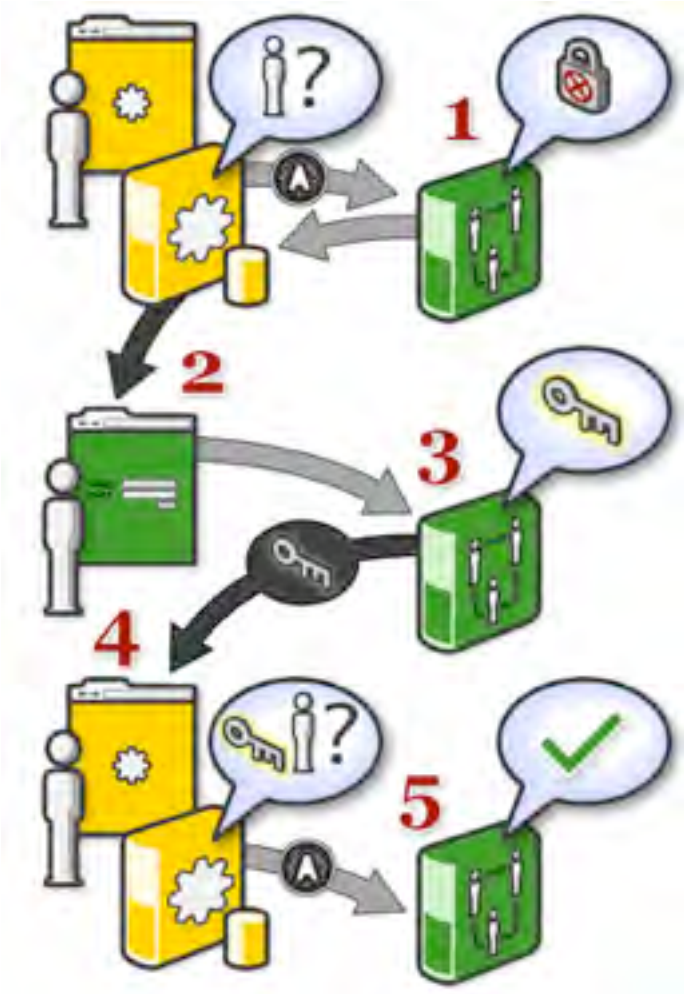
Ablauf vom dreibeinigen OAuth

- Der Schlüssel wird während der gesamten Kommunikation zwischen Konsumenten und Providern innerhalb einer Anfrage mit angegeben.

schülerVZ studiVZ meinVZ

Demo

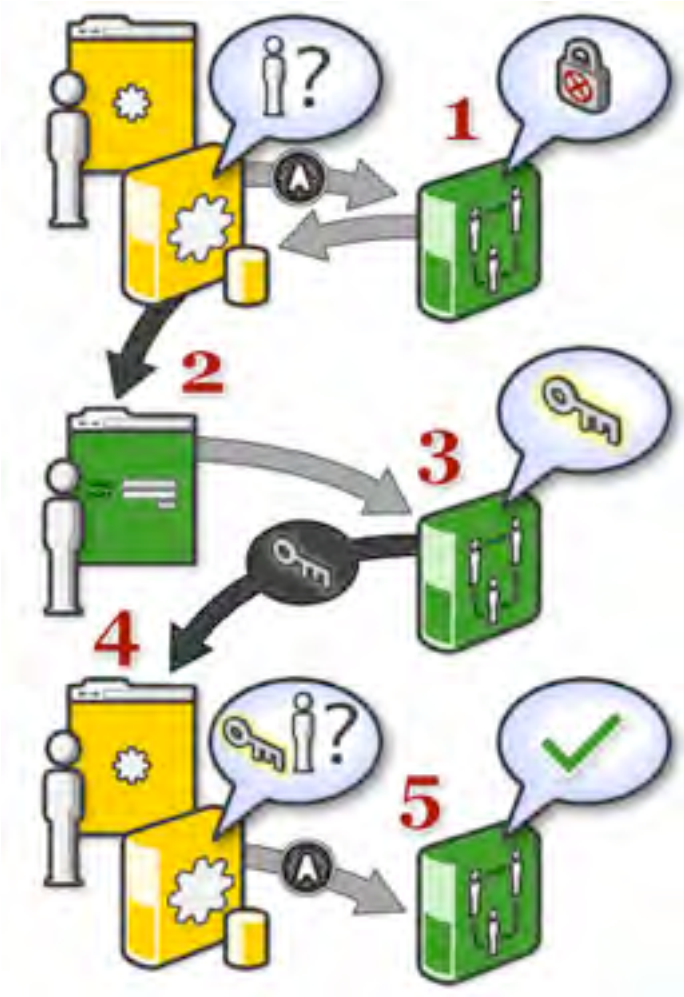
- Der Konsument fragt nach einem nicht bestätigtem Anfrage-Token des Service-Providers.



schülerVZ studiVZ meinVZ

Demo

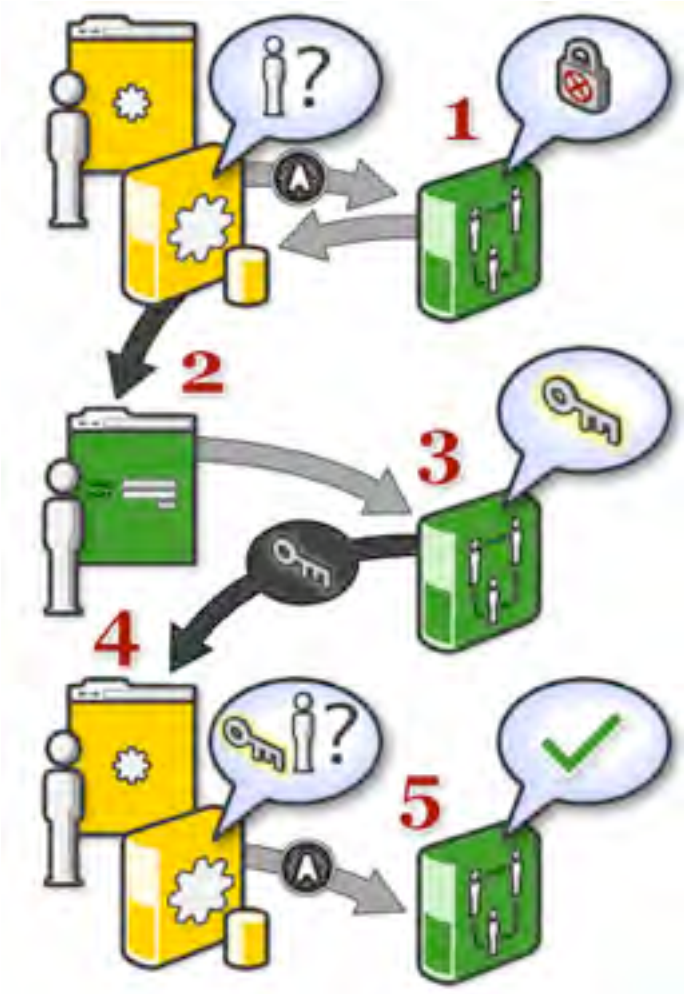
- Der Konsument fragt nach einem nicht bestätigtem Anfrage-Token des Service-Providers.
- Der Endnutzer bestätigt das Anfrage-Token mit einer Authentifizierung durch ein Login beim Service-Provider.



schülerVZ studiVZ meinVZ

Demo

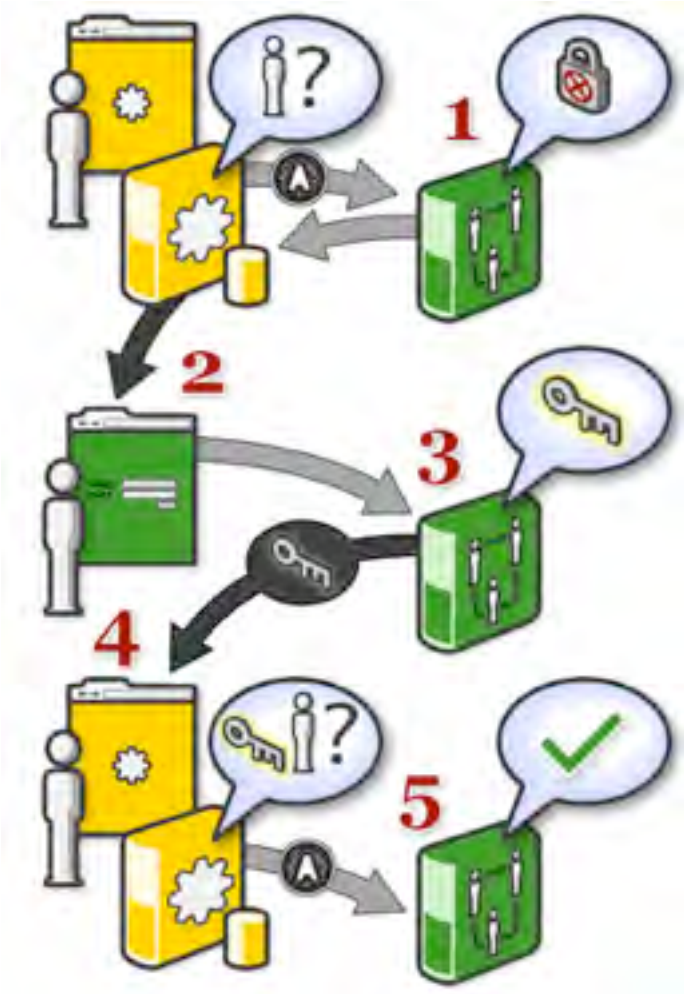
- Der Konsument tauscht das bestätigte Anfrage-Token mit einem Zugriffs-Token aus.



schülerVZ studiVZ meinVZ

Demo

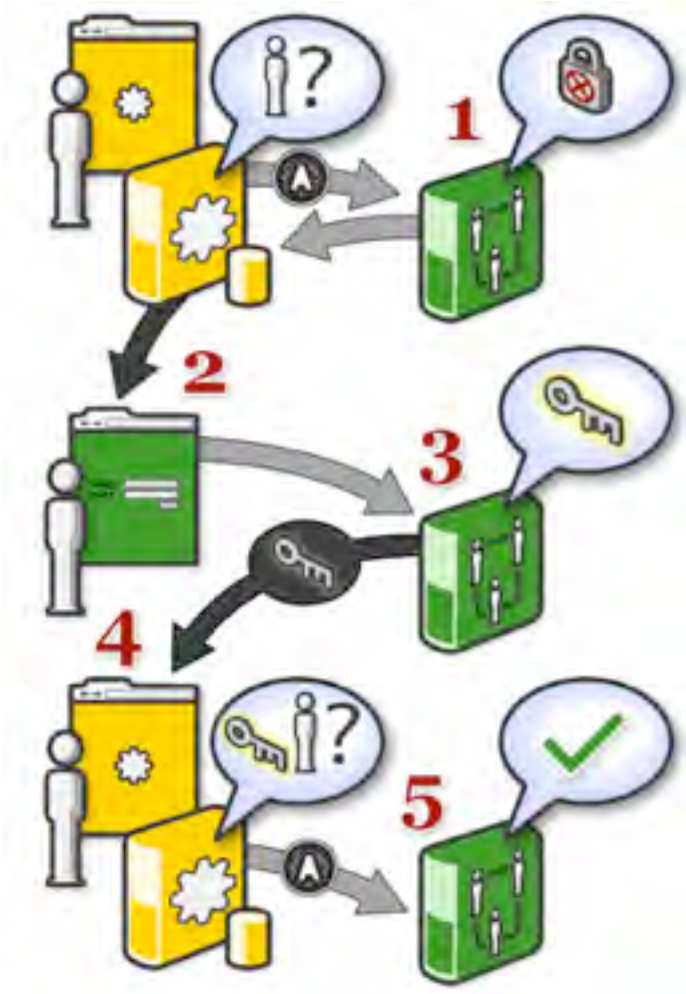
- Mit dem zuvor empfangenen Zugriffs-Token fragt der Konsument anschließend aus den privaten Ressourcen des Service-Providers heraus nach den Daten.



schülerVZ studiVZ meinVZ

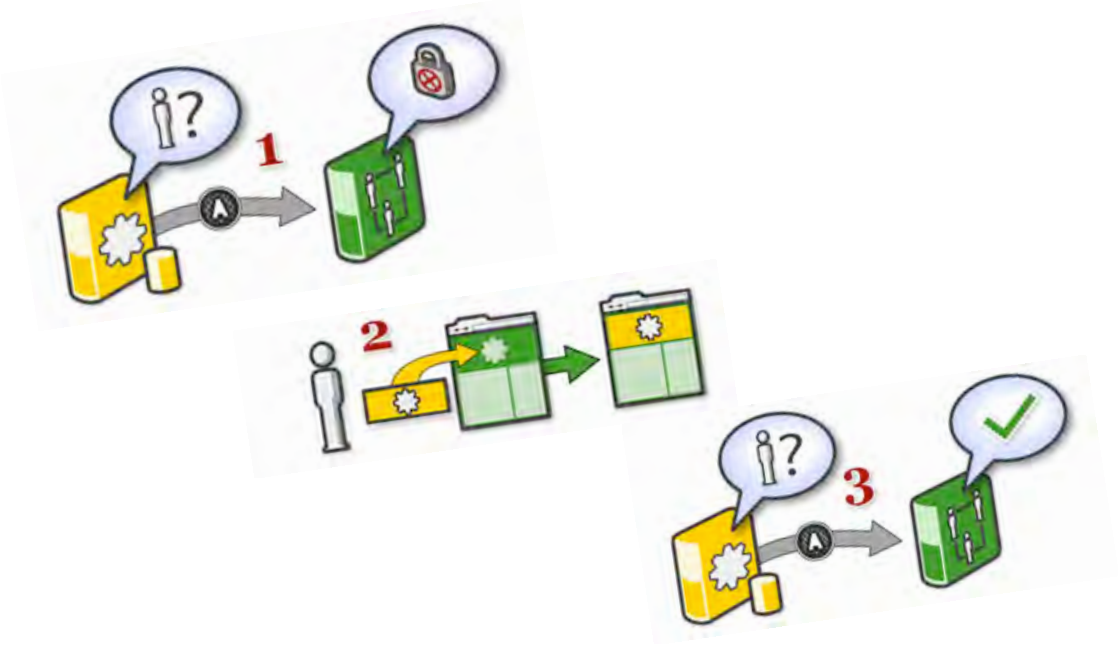
Demo

- Der Service-Provider antwortet mit den erfragten Daten.



schülerVZ studiVZ meinVZ

Ablauf vom zweibeinigen OAuth



schülerVZ studiVZ meinVZ

Ablauf vom zweibeinigen OAuth

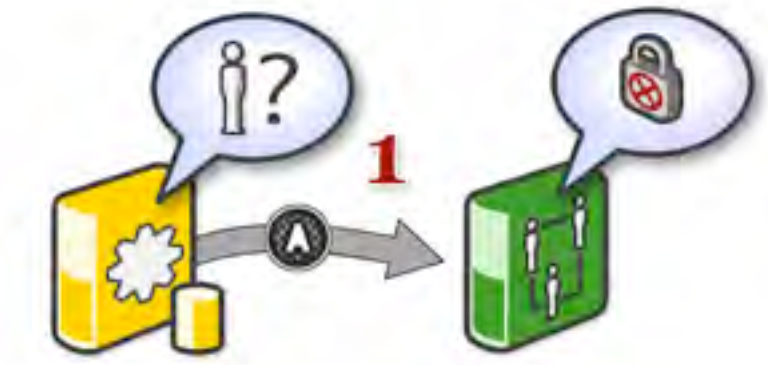
- Konsument benötigt ebenfalls ein Konsumentenschlüssel und -geheimnis

- Beispiel OpenSocial:
  - Anfragen der Konsumenten werden immer im Kontext einer vom Nutzer installierten Applikation gestellt

schülerVZ studiVZ meinVZ

Beispiel

- Server möchte mit seinem Schlüssel und Geheimnis Daten für einen Nutzer beim Provider anfragen.
  => Anfrage wird verweigert.



schülerVZ studiVZ meinVZ

Beispiel

- Der Nutzer installiert zu einem anderen Zeitpunkt die OpenSocial App

schülerVZ studiVZ meinVZ

Beispiel

- Dadurch autorisiert er den Konsumenten und der Server des Konsumenten kann Daten für den Nutzer abholen

Mehr dazu...

- Dienstag, 10:30: OpenSocial in der Praxis

schülerVZ studiVZ meinVZ

Sicherheitsprobleme in OAuth

schülerVZ studiVZ meinVZ

Sicherheitsprobleme in OAuth

Erweiterung des Protokolls

Sicherheitsprobleme in OAuth

- Die OAuth 1.0 Spezifikation nimmt nur `application/x-www-form-urlencoded` Daten aus dem Request-Body in die Signatur auf

- Dadurch sind Man-In-The-Middle Attacken möglich, in denen ein nicht enkodierter Request-Body ausgetauscht wird

Sicherheitsprobleme in OAuth

- Lösung: Zusätzlicher Parameter oauth_body_hash

Sicherheitsprobleme in OAuth

- Signieren:
  - Hash über Request Body erstellen
  - Base64 Enkodierung
  - oauth_body_hash Parameter hinzufügen
  - Request normal signieren

Sicherheitsprobleme in OAuth

- Validierung:
  - Request nach OAuth Standard validieren
  - Bestimme ob oauth_body_hash notwendig
    - Nicht notwendig, nicht vorhanden => OK
    - Nicht notwendig, vorhanden => nicht OK
    - Notwendig, nicht vorhanden => Verhalten Provider abhängig
    - Notwendig, vorhanden => Parameter validieren

schülerVZ studiVZ meinVZ

Sicherheitsprobleme in OAuth

Man-In-The-Middle-Attacke

schülerVZ studiVZ meinVZ

Sicherheitsprobleme in OAuth

- Es gab ein Sicherheitsproblem für das dreibeinige OAuth-Protokoll.

Sicherheitsprobleme in OAuth

- Es gab ein Sicherheitsproblem für das dreibeinige OAuth-Protokoll.

- Dabei ging es um das Ausnutzen des Anfrage-Tokens und der Session beim Autorisierungsprozess in OAuth.

Sicherheitsprobleme in OAuth

- Der Angreifer loggt sich in eine Konsumenten-Webanwendung ein und initiiert den Autorisierungsprozess zu einem Service-Provider.

Sicherheitsprobleme in OAuth

- Der Angreifer loggt sich in eine Konsumenten-Webanwendung ein und initiiert den Autorisierungsprozess zu einem Service-Provider.

- Statt jedoch den Prozess weiter durchzuspielen, speichert er den Anfrage-URI inklusive des Anfrage-Tokens.

Sicherheitsprobleme in OAuth

- Später versucht der Angreifer ein Opfer zu finden, das den Autorisierungsprozess mit dem gespeicherten Anfrage-URI aus dem gleichen Konsumenten weiterführt.

Sicherheitsprobleme in OAuth

- Später versucht der Angreifer ein Opfer zu finden, das den Autorisierungsprozess mit dem gespeicherten Anfrage-URI aus dem gleichen Konsumenten weiterführt.

- Damit ermöglicht das Opfer dem Service-Provider einen Zugriff auf seine privaten Ressourcen.

Sicherheitsprobleme in OAuth

- Mit dem Klick auf den Anfrage-URI des Angreifers führt das Opfer den Autorisierungsprozess des Angreifers mit dem vom Konsument ausgestellten Anfrage-Token weiter.

Sicherheitsprobleme in OAuth

- Nachdem das Opfer seine privaten Ressourcen beim Service-Provider freigegeben hat, kann der Angreifer den gespeicherten Anfrage-URI wieder ausführen.

- Er kann nun auf sämtliche vom Opfer freigegebenen Ressourcen zugreifen.

schülerVZ studiVZ meinVZ

Sicherheitsproblem gelöst

- Um den Man-In-The-Middle-Angriff zu unterbinden, wurde die Spezifikation um folgende Schritte erweitert:

Sicherheitsproblem gelöst

- Callback-Parameter werden beim Request-Token-Request angehängt
- Nach der Autorization wird an den Callback ein Verifier angehängt
- Diesen Verifier muss der Client zusätzlich zum Access-Token mitschicken

schülerVZ studiVZ meinVZ

Fragen?

schülerVZ studiVZ meinVZ

Danke fürs Zuhören!

developer.studivz.net

max@vz.net twitter.com/MaxHorvath

andre@vz.net twitter.com/andre_z

bhofmann@vz.net twitter.com/BastianHofmann