Applications of OAuth in OpenACS

Sebastian Scheder
Learn@WU Systemmanager
Overview

- Introduction to OAuth
  - Motivation
  - Architecture
- Implementation in OpenACS
- Application areas + Demos
About me

- **2011 – 2016**
  - Business School
  - A-Levels in Business Information Systems (“Wirtschaftsinformatik”)

- **2019 – 2022**
  - Co-Worker in some Ruby-on-Rails based projects (e.g., norasports.at)

- **October 2022**
  - Member of the LEARN-Team

- **July 2023**
  - BSc in Business Economics and Social Sciences / IS major at WU

- **October 2023**
  - Master in Software Engineering & Internet Computing at TU Wien
Introduction to OAuth

- OAuth enables a third-party application to gain limited access to an HTTP service:

  a) on behalf of the resource owner
  b) on behalf of the application (= client) itself

- Version 1.0 → RFC 5849
- Version 2.0 → RFC 6749

- Uses **tokens instead of resource owner’s credentials**
  - Standard: Bearer Token (RFC 6750)
Motivation – Business Perspective

- Usage of third-party applications becomes almost inevitable, leading to
  - System breaks between the UIs of multiple applications → workflow interruptions
  - Bad user experience
  - Security issues

- Objectives
  - Improve the integration of third-party clients
  - Contribute to security standards
  - Improve user experience
Main objective:
- overcome the drawbacks of traditional client-server authentication
- Third-party applications use/store credentials of the resource owner
- Permissions cannot be set on a granular level (e.g., resources, duration)
- Identical credentials are used for multiple clients → Security issue
OAuth 2.0 Protocol Flow

Client (= application)

(1) Authorization Request

(2) Authorization Grant
(i.e., client_secret, client_id)

(3) Authorization Grant

(4) Access Token

(5) Access Token

(6) Protected Resource

Resource Owner

Authorization Server

Resource Server
OAuth 2.0 – Implementation in OpenACS

- xooauth package
- Originally developed by Knowledge Markets (see https://km.at)

- Application areas
  1. **App Communication**
     exchange of information with external applications over REST interfaces (e.g., with MS Teams over the MS Graph API)
  2. **LTI Tools**
     communication with external applications in the context of LMS (e.g., BigBlueButton, Jupyter)
  3. **Authentication**
     user authentication with Single-Sign-On on multiple software systems (e.g., via Microsoft Azure AD, GitHub)
Communicating with REST APIs

/usr/local/ns/config-oacs-${oacs_version}.tcl

# e.g., for MS Graph

ns_section ns/server/${server}/acs/oauth/ms/graph {
  ns_param tenant ad0c8c4b-1fc3-4d69-8aea-7532e8b5310c
  ns_param client_id 5e5ad3aa-...
  ns_param client_secret qvj8Q-...
}

xooauth/tcl/ms-procs.tcl

::public method "team get" {
  team_id
  { -expand ""}
  { -select ""}
} {
  set r [:request :method GET :token [::token] \n    -url /teams/${team_id}[:params {expand
    select}]]
  return $r
}

@odata.context {https://graph.microsoft.com/v1.0/$metadata#teams/$entity} id
1335050c-0f9c-4db2-b0dc-e77072e123e createdDateTime 2023-07-20T13:17:51.533Z
displayName {3119 Betriebliche Informationssysteme 1} ...}
Demo: Creating an MS Team in DotLRN
Communication with LTI Services (1)

- Learning Tools Interoperability
  - Standard, allowing the integration of rich learning applications into LMS
  - Developed by the IMS Global Learning Consortium
    (see e.g., http://www.imsglobal.org/activity/learning-tools-interoperability)

- Versions:
  - LTI <= 1.1 (using OAuth 1.0)
  - LTI 1.3 (using OAuth 2.0)

- Components:
  - Tool Provider (e.g., server, running an external tool, such as Jupyter)
  - Tool Consumer ( = an LMS, e.g., LEARN)
Communication with LTI Services (2)

```
/usr/local/ns/config-oacs-${oacs_version}.tcl
# e.g., for Jupyter
ns_section ns/server/${server}/lti/jupyter {
  ns_param launch_url http://${some_url}/hub/lti/launch
  ns_param shared_secret ${some_shared_secret}
  ns_param oauth_consumer_key ${some_consumer_key}
}
* shared_secret and oauth_consumer_key are created on the server via
  $ openssl rand –hex 32
```

- `{launch-jupyter}` includelet in an xowiki page
  - Defined in xooauth/tcl/lti-includelet-procs.tcl

```xo::lti::Jupyter
::xo::lti::Jupyter instproc render {
  :get_parameters
  return [:render_form_button \n    -class ::xo::lti::Jupyter
```
User Authentication (1)

- Authorization Code Grant Flow (defined in RFC 6749)
  - Authorization Endpoint
  - Query parameters
    - **response_type** (e.g., code, code+id_token in Microsoft-specific hybrid flows)
    - **client_id**
    - **redirect_uri**
    - **scope** (e.g., open_id, offline_access, profile)
    - **state** (to encode information about several aspects, e.g., last visited page)
      - e.g.,
      - https://login.microsoftonline.com/common/oauth2/authorize?
        response_type=code+id_token
        &scope=openid+offline_access+profile
        &client_id=5e5ad3aa-e158-48d2-af2f-…
        &response_mode=form_post
User Authentication (2)

- Response = Claims $\rightarrow$ mapped to OpenACS-internal variables

```
  e.g., xooauth/tcl/ms-procs.tcl: get_user_data
  claims {
    ...
    family_name Scheder $\rightarrow$ last_name
    given_name Sebastian $\rightarrow$ first_names
    upn sscheder@wwzgl.onmicrosoft.com $\rightarrow$ email
    ...
  }
  perform_login {
    ...
  }
  {  
    set data [:get_user_data -token $token]
    set user_id [:lookup_user_id -email [dict get $data email]]
    ad_user_login -external_identity [self] $user_id
  }

  instance of ::xo::Authorize
```
Demo: “Login with Microsoft”
Summary and Future Work

- OAuth2
- 3-Layer authorization procedure
- “Never use resource owner credentials to authorize multiple clients”

Application areas implemented in xooauth
- Communication with REST APIs
- LTI Services
- Authentication

Future Work
- Restructure the xooauth package
- Update functionalities (e.g., oauth-server-procs.tcl)
- Add functionality (e.g., MS Graph: list deleted teams, restore deleted teams)
- Separate functionalities into dedicated packages
LEARN-Team
IT-Services
Welthandelsplatz 1, 1020 Vienna, Austria

Sebastian Scheder
T +43-1-313 36-5277
sebastian.scheder@wu.ac.at
www.wu.ac.at