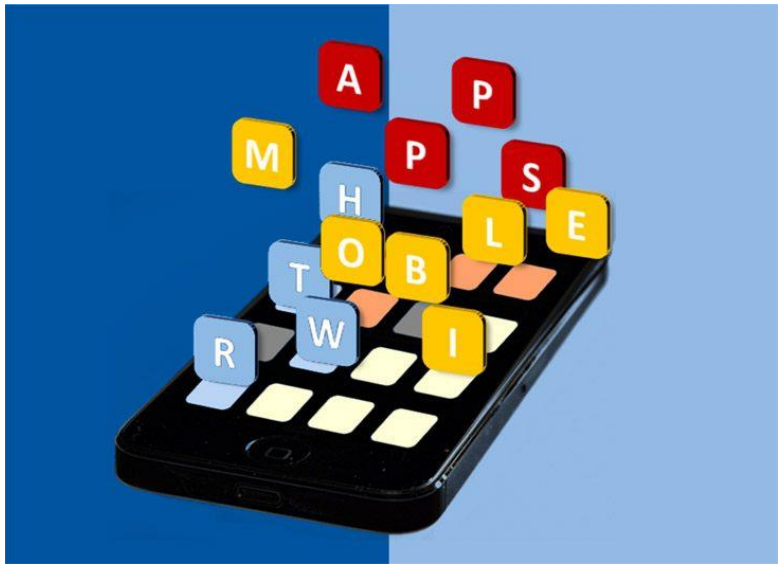
App Development APIs at RWTH Aachen

OAuth2 and L²P 2013 APIs for mobile Applications

Marius Politze
Ahmed Raihan
Dr. Mohamed Amine Chatti

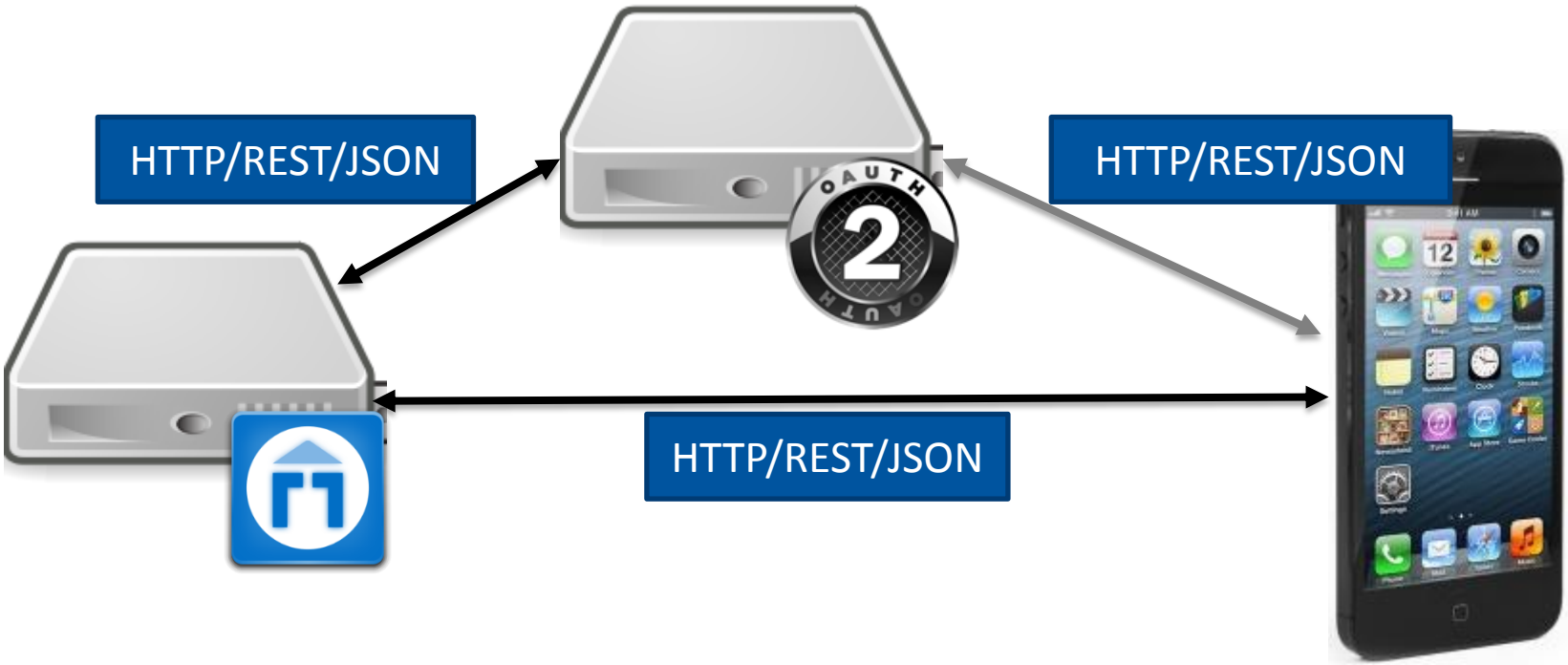
Overview

- **Getting started with RWTH APIs**
  - → OAuth2
  - → L²P 2013
- **Formal stuff**
  - → Register your app
  - → Publishing you app
- **Sample Code**
  - → OAuth2 Authorization
  - → Calling L²P 2013 API



Getting Started

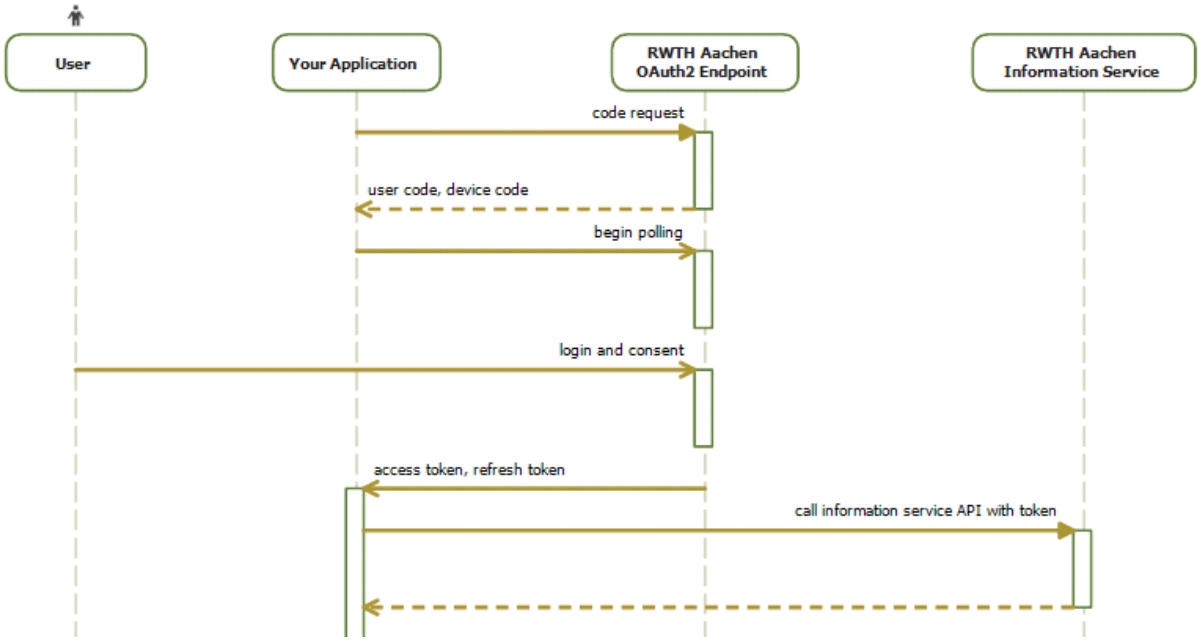
The idea (extremely basic)



OAuth2 Getting started

- **De-facto standard for authorization of apps: REST-API to obtain user consent to use RWTH Aachen APIs in the users name**

- **Documentation: https://oauth.campus.rwth-aachen.de/doc/**



OAuth2 Overview

1. **Obtain token from OAuth2 service by**
   a) Identifying your app
   b) Requesting the user to consent your request
   → The token is personalized and unique for the combination of user, device and application

2. **Use the access token to make API calls for 30 minutes**
   → After 30 minutes the access token will be automatically invalidated

2. **Use the refresh token to re-gain access**
   → A new access token will be issued and is agian valid for 30 minutes

L²P 2013 Web Services

- **Web service endpoints**

  https://www.elearning.rwth-aachen.de/_vti_bin/l2pservices/api.svc/v1/[methodName]

- **Structure equivalent to L²P 2013 curse rooms**

  → Divided into modules

  → Modules provide View, ViewAll, Add, Delete and Update actions

  → Usage limited to sample course rooms

- **Documentation**

  https://www.elearning.rwth-aachen.de/_vti_bin/l2pservices/api.svc/v1/documentation

  → Provides description of web service methods and sample calls

General things about L²P 2013

- **Quite new**
  - → released in April 2014
  - → currently in pilot phase
- **API has only few users**
  - → so far: only you!
  - → still some bugs
- **Feedback appreciated!**
  - → API and L²P are under active development
  - → We can only fix bugs that we know of
  - → Bugtracker: https://www.elearning.rwth-aachen.de/l2p/apibugtracker

Formal Stuff

You and your app

- **When providing an app …**
- **… you will become a service provider**
  - → Users will use your app and may require support
- **… you are handling sensitive, personal data**
  - → Apps using APIs need to be reviewed concerning privacy issues
- **To protect the users from malicious apps every app must be registered**

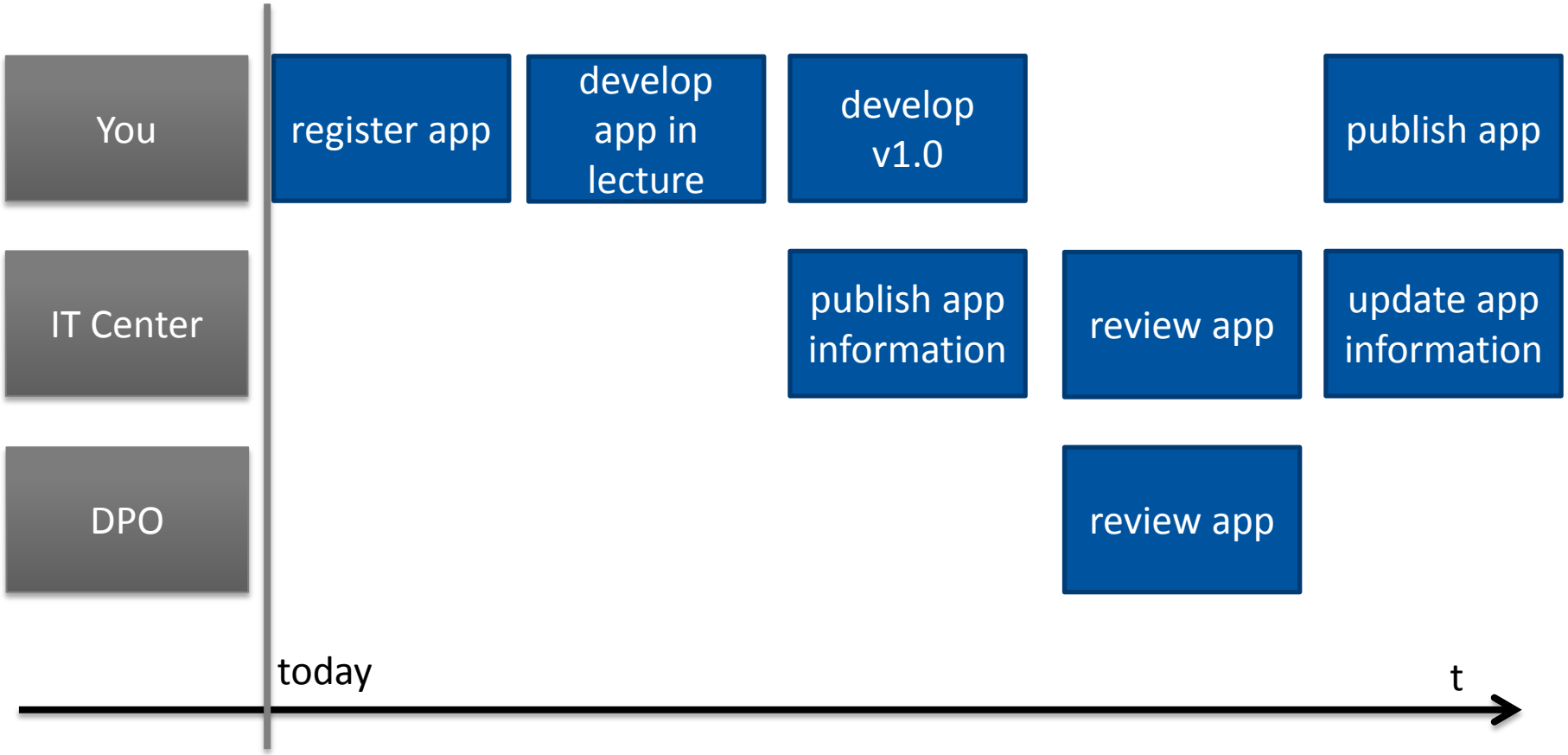
Register your App

- **To use OAuth2 and the L²P 2013 APIs you need to register your app**
  - → Apps will be tracked for reporting and security issues
- **You will obtain an app identifier that is only intended for your app**
  - → The identifier needs to be kept secret!
  - → Store them at a secure location and DO NOT upload them to public source code repositories (Github…)

OAuth2 Prerequisites

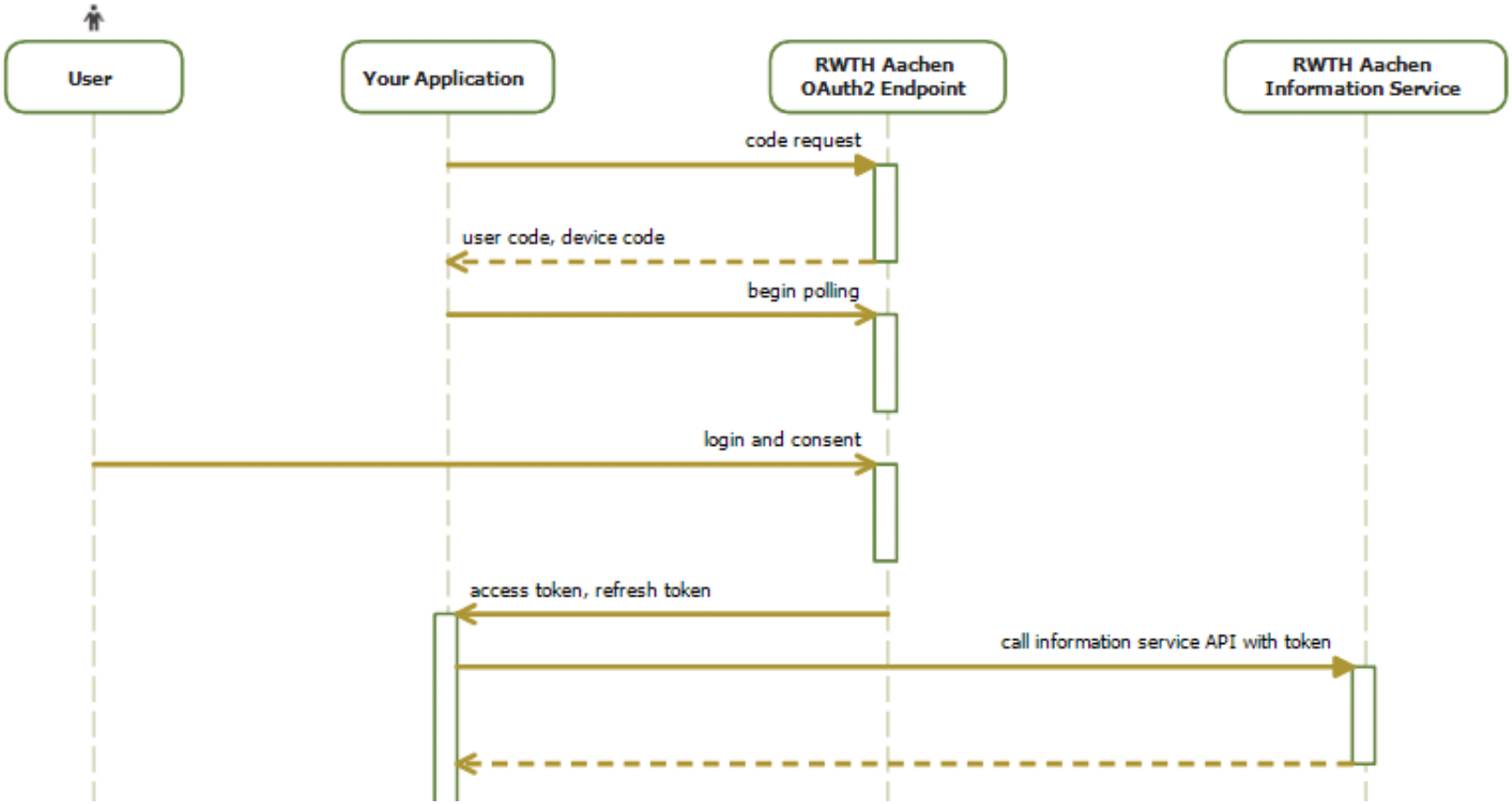
- **Application Identifier**
  - → You will get this after registering your app
- **API scope for L²P 2013 APIs:**
  - → l2p2013.rwth
- **Development title and description of your app**
- **Contact name(s) and email adress(es)**
- **Send your information by email**
  - → to politze@itc.rwth-aachen.de

The road to publish your app



Sample Code

OAuth Device Workflow



Sample Code Helper Methods I

```
private static WebResponse PostRequest(Uri u, string postData){
    byte[] byteArray = Encoding.UTF8.GetBytes(postData);
    WebRequest codeRequest = WebRequest.Create(u);
    codeRequest.Method = "POST";
    codeRequest.ContentType = "application/x-www-form-urlencoded";
    codeRequest.ContentLength = byteArray.Length;
    Stream dataStream = codeRequest.GetRequestStream();
    dataStream.Write(byteArray, 0, byteArray.Length);
    dataStream.Close();
    WebResponse response = codeRequest.GetResponse();
    return response;
}

private static WebResponse GetRequest(Uri u, string query)
{
    WebRequest codeRequest = WebRequest.Create(String.Format("{0}?{1}", u.OriginalString, query));
    codeRequest.Method = "GET";
    WebResponse response = codeRequest.GetResponse();
    return response;
}
```

Sample Code Helper Methods II

```
private static Dictionary<string, string> ReadResponse(WebResponse response){
    Stream responseStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(responseStream);
    string responseFromServer = reader.ReadToEnd();
    reader.Close();
    responseStream.Close();
    response.Close();
    var jss = new JavaScriptSerializer();
    return jss.Deserialize<Dictionary<string, string>>(responseFromServer);
}

private static Dictionary<string, object> ReadResponse2(WebResponse response){
    Stream responseStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(responseStream);
    string responseFromServer = reader.ReadToEnd();
    reader.Close();
    responseStream.Close();
    response.Close();
    var jss = new JavaScriptSerializer();
    return jss.Deserialize<Dictionary<string, object>>(responseFromServer);
}
```

Sample Code OAuth2 I

```
Uri oAuthUri = new Uri("https://oauth.campus.rwt-aachen.de/oauth2waitress/oauth2.svc/");
Uri codeEndpoint = new Uri(oAuthUri, "code");
Uri tokenEndpoint = new Uri(oAuthUri, "token");
Uri tokenInfoEndpoint = new Uri(oAuthUri, "tokeninfo");

string clientId = "5dXWaUetctIMN9qZ39cD3mQWCK.app.rwth-aachen.de";
string scopes = "l2p2013.rwth";

//---------------------------------------------------------------------
// Create Token Request
//---------------------------------------------------------------------
string postData = String.Format("client_id={0}&scope={1}", clientId, scopes);
WebResponse response = PostRequest(codeEndpoint, postData);
Dictionary<string, string> responseFromServer = ReadResponse(response);

//---------------------------------------------------------------------
// Let The User Verify The Token
//---------------------------------------------------------------------
string verifyUrl = String.Format("{0}?q=verify&d={1}", responseFromServer["verification_url"],
responseFromServer["user_code"]);
Process.Start(verifyUrl);
```

Sample Code OAuth2 II

```
//---------------------------------------------------------------------
// Get Access Token & Refresh Token
//---------------------------------------------------------------------
postData = String.Format("client_id={0}&code={1}&grant_type=device", clientId,
responseFromServer["device_code"]);
do
{
    response = PostRequest(tokenEndpoint, postData);
    responseFromServer = ReadResponse(response);
    Thread.Sleep(5100);
}
while (responseFromServer["status"].Contains("error: authorization pending"));
var refreshToken = responseFromServer["refresh_token"];
var accessToken = responseFromServer["access_token"];

//---------------------------------------------------------------------
// Verify Token Info
//---------------------------------------------------------------------
postData = String.Format("client_id={0}&access_token={1}", clientId, accessToken);
response = PostRequest(tokenInfoEndpoint, postData);
responseFromServer = ReadResponse(response);
```

Sample Code L²P 2013

```
//---------------------------------------------------------------------
// Call L2P 2013 API
//---------------------------------------------------------------------
string query = String.Format("accessToken={0}&cid={1}", acessToken, "14ss-33627");
response = PostRequest(tokenEndpoint, postData);
Dictionary<string, object> responseFromL2P = ReadResponse2(response);

foreach(object o in responseFromL2P)
{
    Dictionary<string, object> announcement = (Dictionary<string, object>)o;
    // Do whatever needed with the announcement...
}
```

Sample Code OAuth2 III

```
//-----------------------------------------------------------------
// After 30 minutes use Refresh Token
//-----------------------------------------------------------------
postData = String.Format("client_id={0}&refresh_token={1}&grant_type=refresh_token",
           clientId, refreshToken);
response = PostRequest(tokenEndpoint, postData);
responseFromServer = ReadResponse(response);

//-----------------------------------------------------------------
// Invalidate Refresh Token / Logout the user
//-----------------------------------------------------------------
postData = String.Format("client_id={0}&refresh_token={1}&grant_type=invalidate",
           clientId, refreshToken);
response = PostRequest(tokenEndpoint, postData);
responseFromServer = ReadResponse(response);
```