Token-Based Authentication Using JSON Web Token on SIKASIR RESTful Web Service

Muhamad Haekal
Faculty of Computer Science, Mercu Buana University
Jakarta, Indonesia
rekale1123@gmail.com

Eliyani
Faculty of Computer Science, Mercu Buana University
Jakarta, Indonesia
eliyani@mercubuana.ac.id

Abstract*— The role of SMEs in the economy of Indonesia is significant. However, the problems faced by the SMEs as less professional management often become an obstacle to the progress of the business. To solve these problems, we need an application that can help small business owners to help manage their business and can also be accessed by a variety of platforms. All these problems can be solved with the application SIKASIR. In order SIKASIR application can be accessed by multiple platforms, it takes a web service that resides on a server. Each application SIKASIR of various platforms to be connected to the service RESTful Web Service so interconnected and synchronized. However, there are some problems when making a multi-platform service, one of them is an authentication problem. In REST architecture, we need authentication methods that are stateless, one of which is a token-based authentication methods. On this occasion, this study will use JSON Web Token. JSON Web Token is the new industry standard based (RFC 7519). JSON Web Token is stateless, so it is suitable to be implemented on a RESTful Web Service application. In this research will be discussed how to implement authentication technique using token-based authentication method with a JSON Web Token, which will produce a secure authentication and make web service SIKASIR be multi-platform.

Keywords— Token-based authentication; JSON Web Token; REST; Web Service; Multi-Platform

I. Introduction

The role of SMEs in the economy of Indonesia is quite significant. SMEs in Indonesia has proved resilient to face the economic crisis [1] and even became one of the biggest contributions of Indonesian GDP in 2012 which share 57,2 % of GDP [2]. However, most of entrepreneurs still face some problems in developing their businesses. One of the problems is weak organizational management. Several strategic managements suggested by [3] to solve the problem which all strategies can be implemented efficiently by using information technology. But government data showing 70% of the 56.5mn Indonesian SMEs use no IT solutions [4]. Similarly, in China, the application of information technology in SMEs was still at the low level. Compare to US, nearly 70% of service behavior of SMEs was already on the internet [5].

Many obstacles in impelementing IT in SMEs. Some are low of IT adoption awareness, limited human resources, knowledge, IT infrastructure, and funding, undefined operational procedure and simple business processes. ERP implementation in SMEs require some adjustments to the needs of SMEs [2]. Regarding ERP adoption, [6] suggested that small and medium-sized enterprises should not be considered as one homogenous group.

In 2016, Agung Tri Laksono , Muhamad Haekal and Gagas Julio developed an application named SIKASIR as one IT solution to improve managerial performance of SMEs in Indonesia . The increasing popularity of SPA (Single Page Application) web based application and mobile encourages SIKASIR application to be accessed in multi-platform so the users are able to use SIKASIR application without certain gadget or platform.

To ensure the SIKASIR application can be integrated and is multi-platform, web services is created to link many platforms, such as desktop, mobile and web. Web Services can access to Services through the SOAP protocol as used by [7] in designing the digital campus but the SIKASIR used REST (Representational State Transfer). Business logic and presentation logic processes are detached. The back-end/server part is about business logic and data processing, while the presentation logic is exclusively transferred to front-end. Such changes have led to new ways of implementing authentication in modern applications.

REST is a web standard architecture that uses HTTP protocol for data communication. Such architecture was established based on the data source in which each component is a data source. The data source is accessed by the same interface using HTTP standard method [8].

In the REST architecture, the server which follows REST architecture provides access to the data source and clients who retrieve data. Each data source is identified using URI link. REST uses various formats for presenting data, such as text, JSON and XML. Following are the HTTP methods used in REST architecture:

- *GET* – providing access to read the data source
- *PUT* – updating available data
- *DELETE* – deleting data
- *POST* – creating new data

Various techniques to secure RESTful web services, which are developed by using ASP.NET WEB MVC and have been investigated by [9], use basic authentication, claim based authentication, and token based.

978-1-5090-1648-8/16/$31.00 ©2016 IEEE

Authentication is one of the most important parts in every application. For several decades, server based cookies and authentication is the easiest solution for authentication in web-based application. Other used behavior biometrics such as used by [10]. However, the authentication handling for multi-platform might be complicated if it uses server based authentication. XML was used by [7], SAML (Security Assertion Markup Language) was used by [11], but SIKASIR used JSON Web Token (JWT) as its format was shorter.

The general concept of token-based authentication is allowing users to enter their username and password to obtain a token that enables them to take a particular resource. Once the token is obtained, the user can send a token which offers access to certain resources in the certain time period to remote sites [12]. The token given is the one that has been created on the server with a particular algorithm. In such token, we can know who the owner is as well as the right of access that can be conducted by the client.

JSON Web Token (JWT) is a representation of the claim format which is intended for limited environment space such as HTTP Authorization headers and URI query parameters. JWTs encodes the claim to be sent as JSON objects that is used as payload structure from Signature JSON Web (JWS) or as a plaintext structure from JSON Web Encryption (JWE), enabling claims to be digitally signed/protected with Message Authentication Code (MAC) [13].

JWT consists of three structures separated by a full stop (.), namely:

- *Header*. There are two parts in the header, one of them is token type, namely JWT, and another one is hashing algorithm used, such as HMAC SHA256 or RSA.
- *Payload*. It is the second part of token which consists of claim. Claim is the statement about an entity (usually users) and supplementary metadata.
- *Signature*. To create a signature, JWT must follow the JWS specification. JWS is the content secured by digital sign or MACs which uses JSON as the basic data structure [13].

The purposes of this research are:

1. To implement token-based authentication using JWT on SIKASIR RESTful web service as the alternative technique to replace server-based authentication.
2. To ensure SIKASIR web service can be accessed by multi-platform.

II. Design and Implementation

SIKASIR web service application was developed by using Waterfall method, whereas the trial of this application was using Black Box Testing.

A. Token-Based Authentication

The architecture of SIKASIR web service in general can be seen on Fig. 1.

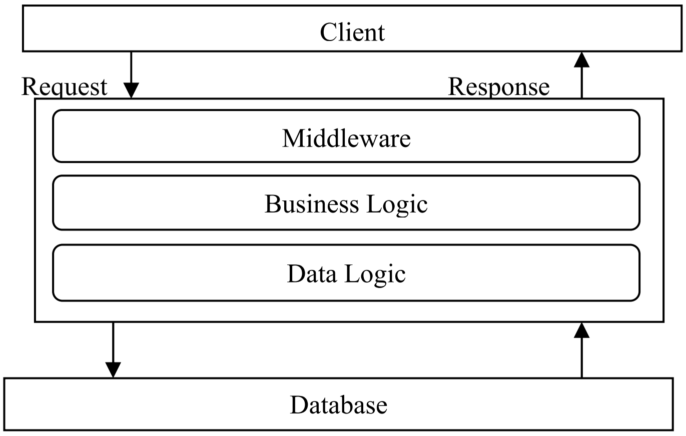


Fig. 1. The Architecture of SIKASIR Web Service

The architecture of SIKASIR consists of three main parts, namely:

1) *Middleware*: the function is to check token authentication, access right, as well as to check whether the clients are active or not.
2) *Business Logic*: the function is as the core functionality of the application, summarizing all business logics implemented by SIKASIR.
3) *Data Logic*: it summarizes algorithms to obtain, delete, update, and create data. It usually contains the query to communicate with database.

The first thing to do to use SIKASIR application is register. The owner of the company has to fill out the data form, such as email and password. Once the registration succedded, the owner of the company must make a payment to activate the account. The owner of the company and the employees will not be able to access their account if they have not completed the payment.

After the registration, the clients may have an access to any resources they owned. However, there is another step must be conducted by clients. They have to login to their account first. The response from the server after login is token which will be used as the key to access any resources in the server. The work flow of login is describe in Fig. 2.

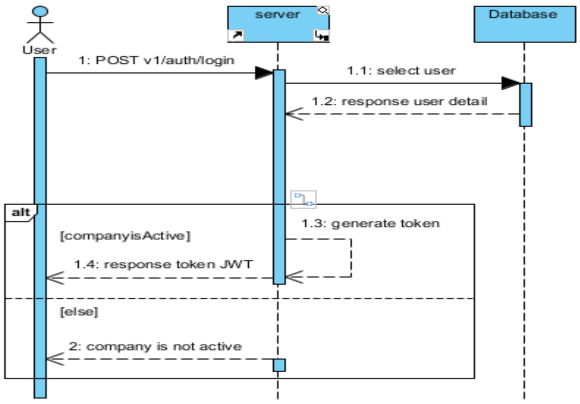


Fig. 2. The work flow of SIKASIR login

During the login process, server will check the email and the password sent by client. If it is found in the database, server will then check whether the client's company is active. If the company is active, server will use the related client's data to create JWT. The structures of JWT created by the server are as follow:

1) *Header :* consists of information about the various algorithms used to create signature by utilizing HMAC256 algorithm. The format of JSON header is as follow:
   ```
   {
      ”alg”:”HS256”,
      ”type”: ”jwt”
   }
   ```
   The JSON above will be encoded with base64 algorithm, for example the pseudocode to be encoded to base64 is: $header = base64_encode($jsonHeader)
2) *Payload :* comprises client ID, client company ID, who gives the token, the expiry date of the token, and access right to the token. Following is the JSON format for payload:
   ```
   {
    “iss”: “api.SIKASIR.com”,
    “exp”: 1426420800,
    “access”: [{
       ”read-order”
       ”read-outlet”
        ....
    }],
    “company_id”: 1,
    ”klien_id”: 1
   }
   ```
   For example: the pseudocode to be encoded to base64 is: $payload = base64_encode($jsonHeader)
3) *Signature* : it is created from header and payload which are joined and encoded using base64 algorithm, then is encrypted using HMAC-256 encryption algorithm. For example: the pseudocode to create signature is as follow:
   $signature= HMAC($header + $payload, $key);

The result of those three processes will be joind and will result JWT. The example of pseudocode to join header, payload and signature is as follow:

$token = $header + ’.’ + $payload + ’.’ + $signature;

With the token obtained from the server, the client can access resources of SIKASIR web service by inputing token in HTTP header in the authorization method. Following is the example of HTTP header:

```
GET /v1/employees HTTP/1.1
Host: api.sikasir.com
Content-Type: application/json
Authorization:bearer
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEsI
mlzcyI6Imh0dHA6XC9cL2xvY2FsaG9zdDo4MDAwXC92M
VwvYXV0aFwvcmVmcmVzaCIsImlhdCI6MTQ2MDQwMT
A1OCwiZXhwIjoxNDYxMjIxMTYyLCJuYmYiOjE0NjEwN
DgzNjIsImp0aSI6IjUzYjEwMzg4YjE5Yzc0YjAzYmZmOTh
mZGNjODYwYTYzIn0.64gy82KgvwhZ2OtN_5_-
BiUmYQWO2bhVKeqCsLdfw5g"
```

When the client sends request to the resources, SIKASIR web service will first verify the token. After being verified, the server will check the expiry date of the token. If it is expired, the server will provide an information to client that the token is no longer able to use. Meanwhile, if it is not expired, the last stage will be checking the access right owned by the toked in the payload. If it has the access right to the expected resources, SIKASIR web service will respond it with the resources needed by the client. The description of token verification algorithm when the client is about to access the employee data in api.sikasir.com/v1/employees is presented in the Fig. 3.

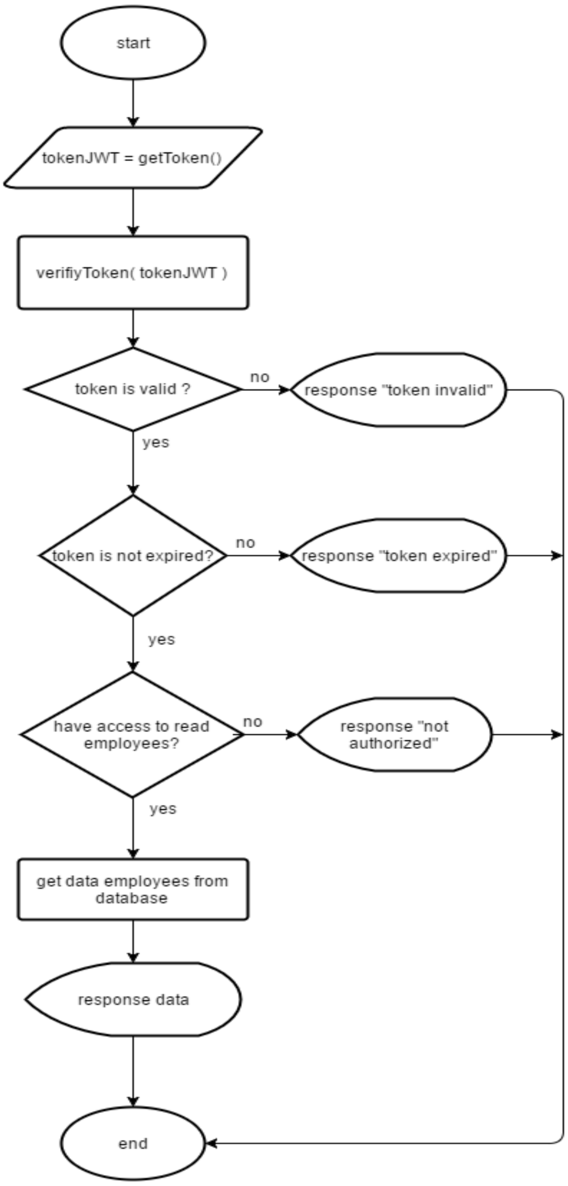


Fig. 3. The Flowchart of Token Verification

There are 3 different types of role in the access right on SIKASIR application, namely:

1. Cashier

The main duties of cashier are to run the selling and payment processes and to record all transactions. The access right owned by the cashier is presented on Fig. 4:

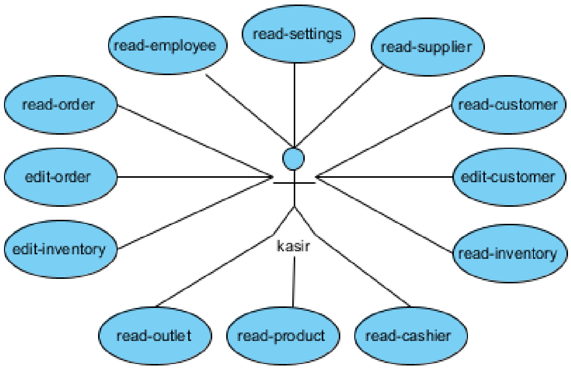


Fig. 4. The Access Right of Cashier

2. Staff

The main duties of staff are to manage the transaction report and the inventory. Staff owns the similar access right with the cashier with an additional access right explained in Fig. 5:

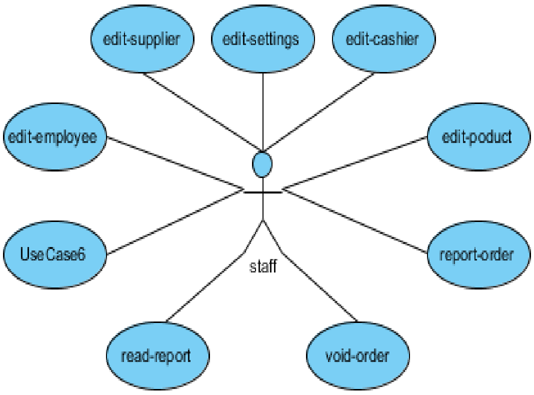


Fig. 5. The Access Right of Staff

3. Owner

Owner is the one who registers to the cashier application. Owner has all access right owned by cashier and staff as well as other access rights exmpained in Fig. 6:

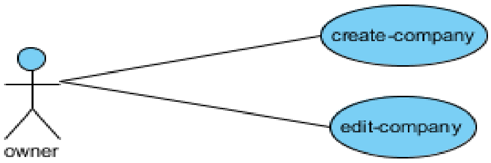
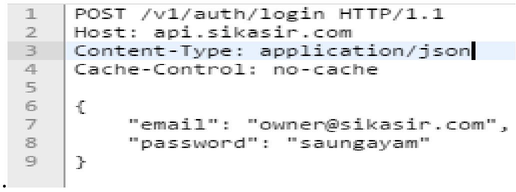


Fig. 6. The Access Right of Owner

B. Black Box Testing

Before the testing started, a dummy data, which consists of company data, outlet, employee, etc, is created. Next, login with several existing roles, such as owner, staff, and cashier. Then, each role tries to access some resources links with the token obtained during the login process through Postman application, to test the verification of client token and access right owned by the token. Fig. 7 is the content of HTTP Header request to test the login process with role as owner.

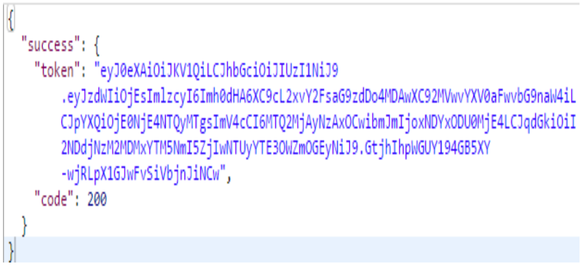


```
POST /v1/auth/login HTTP/1.1
Host: api.sikasir.com
Content-Type: application/json
Cache-Control: no-cache

{
    "email": "owner@sikasir.com",
    "password": "saungayam"
}
```

Fig. 7. HTTP Header login as owner

The testing of login is conducted on the following link: *api.sikasir.com/v1/auth/login*. The HTTP method used is POST. Host is the place where web service resouces belongs, namely on api.sikasir.com. The content will be sent to server is in the JSON format which is elaborated in the content-type. Last, the HTTP body which consists of email and password is in JSON format. After sending request to the web service with different roles, then the server will response as presented in Fig. 8.

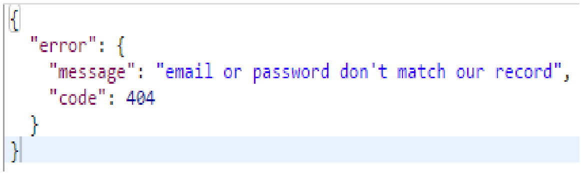


```
{
  "success": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
        .eyJzdWIiOjEsImlzcyI6Imh0dHA6XC9cL2xvY2FsaG9zdDo4MDAwXC92MVwvYXV0aFwvbG9naW4iL
        CJpYXQiOjE0NjE4NTQyMTgsImV4cCI6MTQ2MjAyNzAxOCwibmJmIjoxNDYxODU0MjE4LCJqdGkiOiI
        2NDdjNzM2MDMxYTM5NmI5ZjIwNTUyYTE3OWZmOGEyNiJ9.GtjhIhpWGUY194GB5XY
        -wjRLpX1GJwFvSiVbjnJiNCw",
    "code": 200
  }
}
```

Fig. 8. Login Response as Owner

The response received from the result of login testing is JSON which consists of token and HTTP code. Token obtained from each roll will be different because it has different user biodata and access rights. Meanwhile, HTTP code which is worth 200 means the request has been successfully executed [14].

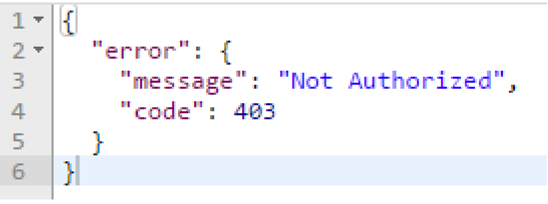
The next testing is to log in with the wrong password/email or the inexistant ones. The responses obtained is like in Fig. 9 which means the requested resource is not found [14].



```
{
  "error": {
    "message": "email or password don't match our record",
    "code": 404
  }
}
```

Fig. 9. Login Response Fails

The next experiment is to test access rights owned by token from each role. Each role will try to access the resource links that have restricted access rights by the web service. The result showed that the responses were the same as expected such as presented in Fig. 10.



```
{
  "error": {
    "message": "Not Authorized",
    "code": 403
  }
}
```

Fig. 10. Response Has No Access Rights

The last experiment is to try to change the token obtained. This experiment uses a user token that has a staff role. The token is as follow:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0IiwiaXNzIjoiaHR0cDpcL1wvYXBpLnNpa2FzaXIuY29tXC92MVwvYXV0aFwvbG9naW4iLCJpYXQiOjE0NjE5Njg2MDksImV4cCI6MTQ2MjE0MTQwOSwibmJmIjoxNDYxOTY4NjA5LCJqdGkiOiJiNGM1OTNjMDNjZGQ2NGI0MmRkNGI1MzlhNDg5ZjA1ZiJ9.K7tKHbPMV2TSUpAxi8FpZnwYRGG-dFexhmEbAx9KE7A

From the token above, a token is changed by inserting one letter to examine whether the user will still be able to access the web service if the token is changed. Following is the token that has been changed:

e**L**yJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0IiwiaXNzIjoiaHR0cDpcL1wvYXBpLnNpa2FzaXIuY29tXC92MVwvYXV0aFwvbG9naW4iLCJpYXQiOjE0NjE5Njg2MDksImV4cCI6MTQ2MjE0MTQwOSwibmJmIjoxNDYxOTY4NjA5LCJqdGkiOiJiNGM1OTNjMDNjZGQ2NGI0MmRkNGI1MzlhNDg5ZjA1ZiJ9.K7tKHbPMV2TSUpAxi8FpZnwYRGG-dFexhmEbAx9KE7A

The bold letter is the letter inserted to the initial token. The result obtained after accessing resources link with the changed token is the message of token invalid.

III. Conclusion and Suggestion

From this research, we can conclude that token-based authentication using JSON Web Token can be applied in SIKASIR web service. With the token, we can also know who access the application, what their role is, and any links that are allowed to access. Such token can also be used by a various platform, such as mobile, desktop, as well as web application. Thus, SIKASIR web service is multi-platform. If there is someone who attempts to change or edit the content of the token, the web service will send an error message telling that the token is invalid. Therefore, the authentication using JSON Web Token is considered secure.

The limitation of this research is there is no discussion about where the token obtained by users should be kept to prevent from being stolen by unexpected people. There should be more research to find out the safest place to keep the token, either on mobile, desktop or web application.

References

[1] A. Subardjo, M.I. Rahmawati, A. Sukmaaji. “Disocial media synchronization model: In association with developing Small Medium Enterprise’s sales information system”, Journal of Theoretical and Applied Information Technology Vol. 82 No. 3 pp. 341 – 346, 1st December 2015.

[2] P. W. Handayani, A.N. Hidayanto, I. Budi. “Business process requirements for Indonesian Small Medium Enterprises (SMEs) in implementing Enterprise Resource Planning (ERP)”, International Journal of Innovation, Management and Technology Vol. 4 No. 1 pp. 93-97, February 2013.

[3] A. A. Rumanti and K. J. Syauta. “Determining strategies based on strategic position analysis in Small and Medium Enterprises”, International Journal of Information and Education Technology Vol. 3 No. 4 pp. 442-447, August 2013.

[4] BMI Research. “Indonesia Technology Report Q3 2016”, April 2016.

[5] J. Xie and Q. Qin, “Research on the Small and Medium Sized Enterprise informatization’s model and strategy of development”, Applied Mechanics and Materials Vols. 63-64 pp. 309-312, 2011.

[6] S. Laukkanen, S. Sarpola, P. Hallikainen. “Enterprise size matters: objectives and constraints of ERP adoption”, Journal of Enterprise Information Management Vol. 20 No. 3 pp. 319-334, 2007.

[7] X. Li. “The design of digital campus unified identity authentication system based on web services”, Applied Mechanics and Materials Vols. 427-429 pp. 2301-2304, 2013.

[8] R. T. Fielding, “Representational State Transfer”. University of California, Irvine. 2000.

[9] M. I. Hussain and N. Dilber. “Restful web services security by using ASP.NET web API MVC based,” Journal of Independent Studies and Research – Computing Vol. 12 no. 1, pp 4-10, January 2014.

[10] H. Dozono, N. Yamasaki, M. Nakakuni. “A method for authentication using behavior biometrics on WEB”, Proceedings of the International Conference on Security and Management (SAM): 1-5, Athens: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014.

[11] T. Shiroma, T. Nagata, Y. Taniguchi, M. Nakamura, S. Tamaki, “Extension of openID connect for utilizing attributes”, INFORMATION Vol. 19 No. 2 pp. 705-715, 2016.

[12] Chris Schmidt. “Token Based Authentication - Impelementation Demonstration”. Workshop Friend of a Friend, Social Networking and the Semantic Web, 1st-2nd September 2004.

[13] M. Jones, J. Bradley, N. Sakimura, “JSON Web Token (JWT)”. IETF, May 2015.

[14] R. Fielding & J. Reschke, “Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content” IETF, June 2014.