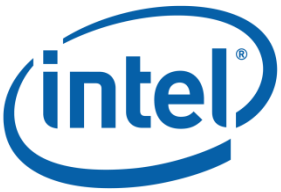


Intel Accelerated Storage Manager

Security REST API

INTEL CONFIDENTIAL

November 2018

Document Revision 2.0.1

Table of Contents

Revision History

Revision	Contributor	Description	Date
0.1	Bartosz Furtak	Initial version of document.	March 2015
0.1.1 - 0.1.9	Team	Peer editing	March-December 2015
0.2	Marek Bielańczuk	Merge and cleanup	December 2015
0.2.1	Piotr Wróblewski	Password credentials POST response. IP scopes. Style and language fixes. Optional redirect_uri access token request. Added pragma and cache control to response examples. Fixed bearer to Bearer in responses. Added expires_in seconds unit.	January 2016
0.2.2	Piotr Wróblewski	Code grant and access token examples for *redirect_uri.*	February 2016
0.2.3	Piotr Wróblewski	Added state parameter for authorization request in implicit grant and code grant.	March 2016
0.2.4	Piotr Wróblewski	Removed *charset=UTF-8* from responses and requests Refresh token changed to POST.	May 2016
0.2.5	Piotr Wróblewski	Style and language fixes. Removed optional for *refresh_token* authorization request for access token. Added *domain* and *username* for authorization request for access token.	May 2016
0.2.6	Piotr Wróblewski	Added *scope* parameter to *password credentials* and *authorization code* response. Updated product name.	June 2016
1.1.0	Marcin Dembiński	Added state field for code grant token request.	June 2017
1.1.1	Marcin Dembiński	Removed duplicated response from authorization_code request.	June 2017
1.1.2	Marcin Dembiński	Added scope in refresh_token response	November 2017
2.0	Marcin Dembiński	Global refactor of documentation	March 2018
2.0.1	Klaudia Jabłońska	Add code parameter for authorization request for access token in code grant.	November 2018

1. Introduction

This document is a reference for Security Provider API. Scope of this API covers available mechanisms of authorization in Intel ASM Service. All used mechanisms are described in RFC6749 and RFC6750.

2. URI structure

Access to resources is provided via URI paths. Client applications should send requests over the HTTP protocol using standard GET and POST methods.

Intel ASM Security Provider URI structure:

```
{protocol}://{host}[:{port}]/v1/oauth/{resource}[?{parameter}={value}&{parameter}={value}]
```

Where:

{protocol}	Communication protocol – **http** or **https**
{host}	Host machine IP address or domain name where IASM is installed
{port}	Port number on which IASM service is listening. Optional if port is default for its protocol (**80** for http, **443** for https)
v1	Stands for first version of API definition
oauth	Namespace for all resources related to commands sent to Security Provider
{resource}	Name of resource to be returned
{parameter}	Name of URI query parameter
{value}	Value of URI query parameter

3. Error response

If any request results with failure, appropriate message will be returned in JSON or HTTP format. Type of response depends on specific use case, for details, see responses for certain request.

3.1 JSON Error Response Format

Error response with JSON payload		
error	string	General message error. Possible values (from RFC): invalid_request, invalid_scope, invalid_token, invalid_grant, invalid_client, unauthorized_client, unsupported_response_type, unsupported_grant_type, access_denied, server_error.
errorDescription	string	Optional, detailed error message about the issue.

```
Status: 400 Bad Request
Content-Length: 73
Content-Type: application/json
{
    "error": "invalid_request",
    "errorDescription": "Parameter missing: scope"
}
```

3.2 HTTP Error Response Type

For specific requests, error response can be returned in HTTP format. In this case, error information is sent back as redirection in query or fragment of the URI to address from "redirect_uri" parameter of the request.

Important Note: Exceptions from redirect response in HTTP are two situations: invalid redirect URI or invalid client id (as described in OAuth specification). These two are always returned as JSON response without redirection.

Http error response format		
error	string	General message error. Possible values (from RFC): invalid_request, invalid_scope, invalid_token, invalid_grant, invalid_client, unauthorized_client, unsupported_response_type, unsupported_grant_type, access_denied, server_error.
errorDescription	string	Optional, detailed error message about the issue.

```
Status: 302 Found
Content-Length: 0
Location: https://example.com#error=access_denied&errorDescription=Invalid+username+or+password&state=773c1cf7d2
```

4. Authorization methods

Below is a list of all available authorization methods for Security Provider.

4.1 Implicit grant

Implicit grant is a token based authorization method dedicated for such clients like JavaScript applications (working in a user agent – typically browser) that supports redirection. It does not provide support for refresh token mechanism (described in OAuth specification).

4.1.1 Authorization request

[GET] /v1/oauth/authorize		
Obtains access token if authorized.		
Request parameters		
client_id	string	Identifier of a registered client.
response_type	string	Must be set to: **token**.
scope	string	List of scopes of the token separated by +. Each scope is in form [IP@]scope, where IP is an address of the resource machine (which is optional). Scopes are defined in plugins documentations.
redirect_uri	string	*[Optional parameter]* Uri to which a user will be redirected after completing the authorization process. If provided, responses are embedded in HTTP Response. If not, JSON object is returned.

state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Using this field is recommended.
Response parameters (after successful authentication)		
access_token	string	Token obtained from the authorization server to access resources.
domain	string	Domain provided during authorization.
expires_in	integer	Time in seconds before access token is flagged as expired. Configurable in the configuration file.
scope	string	List of granted scopes. Not all scopes can be provided to the user in the configuration file and that is why response list can differ from request list.
state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Returned if request had **state** field.
token_type	string	Type of a token returned within request to attach it with all other requests ("Authorization": "token_type access_token").
username	string	Username provided during authorization.

Available Errors		
invalid_request	400	Required field is missing, field type mismatch, not allowed field is present or field's value does not meet specific requirements (e.g. username is too long).
unsupported_response_type	400	response_type is not equal "token" or "implicit" grant is not available.
invalid_client	401	Client ID does not exist.
invalid_scope	400	Every provided scope is invalid.
access_denied	400	Username, password or domain is invalid or login attempts limit is reached.
unauthorized_client	400	Implicit grant is not available or none of the scopes are authorized for the client.
server_error	500	An unexpected error occurred during authentication process.

Request Example (no redirect):

```
GET https://example.com/v1/oauth/authorize?client_id=Example+Responsive+App&response_type=token&scope=info+vroc+ssd&state=x85ds91c1f
```

```
# After successful authorization
{
    "access_token": "3a32e7de582ed09f772e5ce505abd76dfae7c556072bdc7ae2dc883aecf259970f4a84
4a2f988d9e2dc77393617a87f54b0526d59bf0558ba8bf551537a67ddd",
    "domain": "corp.example.com",
    "expires_in": 3600,
    "scope": "info+vroc+ssd",
    "state": "x85ds91c1f",
    "token_type": "Bearer",
    "username": "root"
}
```

```
# Error response (e.g. invalid credentials)
Status Code: 400 Bad Request
{
    "error": "access_denied",
    "errorDescription": "Invalid username or password",
    "state": "x85ds91c1f"
}
```

Request Example (with redirect):

```
GET https://example.com/v1/oauth/authorize?client_id=Example+Responsive+App&response_type=t
oken&scope=info+vroc+ssd&state=x85ds91c1f&redirect_uri=https://example.com/responseApp
```

```
# After successful authorization
Status Code: 302 Found
Location: https://example.com/responseApp#access_token=6fda04dcd30362318d4a96eb21000581be3a
6ba54d07c3e2766c822eb000838a5050b3261101dabe5da2436d00220fb9eec106b7a48e91ee920261988d033bb
3&domain=&expires_in=3600&scope=info%2Bvroc%2Bssd&state=x85ds91c1f&token_type=Bearer&userna
me=root
```

```
# Error response (e.g. invalid credentials)
Status Code: 302 Found
Location: https://example.com/responseApp#error=access_denied&errorDescription=Invalid+user
name+or+password&state=x85ds91c1f
```

```
# Error response for invalid client id or redirect URI
Status Code: 401 Unauthorized
{
    "error": "invalid_client",
    "errorDescription": "Invalid client identificator",
    "state": "x85ds91c1f"
}
```

4.2 Code grant

Token based authorization method supporting refresh token mechanism. It is mostly designed for confidential applications with capability of redirection.

4.2.1 Authorization request for code grant

[GET] /v1/oauth/authorize		
Obtains access code if authorization succeed. Access code is not yet an access token, user has to send another request to obtain a fully valid access token.		
Request parameters		
client_id	string	Identifier of a registered client.
response_type	string	Must be set to: **code**.
scope	string	List of scopes of the token separated by +. Each scope is in form [IP@]scope, where IP is an address of the resource machine (which is optional). Scopes are defined in plugins documentations.
redirect_uri	string	*[Optional parameter]* Uri to which a user will be redirected after completing the authorization process. If provided, responses are embedded in HTTP Response. If not, JSON object is returned.
state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Using this field is recommended.
Response parameters (after successful authentication)		
code	string	Token obtained from the authorization server to access resources.
state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Returned if request had **state** field.

Available Errors		
invalid_request	400	Required field is missing, field type mismatch, not allowed field is present or field's value does not meet specific requirements (e.g. username is too long).
unsupported_response_type	400	response_type is not equal "code" or "code" grant is not available.
invalid_client	401	Client ID does not exist.
invalid_scope	400	Every provided scope is invalid.
access_denied	400	Username, password or domain is invalid or login attempts limit is reached.
unauthorized_client	400	Code grant is not available or none of the scopes are authorized for the client.
server_error	500	An unexpected error occurred during authentication process.

Request Example (no redirect):

```
GET https://example.com/v1/oauth/authorize?client_id=Example+Responsive+App&response_type=code&scope=info+vroc+ssd&state=x85ds91c1f
```

```
# After successful authorization
{
    "code": "61e88984474d3c25e3800b1bf1945a02ae717373575befca1e1762107b33b6f3ccf88a7fd17141
06511433a4bf989fdc11e417f02c8ac78a73f40f2466f481a7",
    "state": "x85ds91c1f"
}
```

```
# Error response (e.g. invalid credentials)
Status Code: 400 Bad Request
{
    "error": "access_denied",
    "errorDescription": "Invalid username or password",
    "state": "x85ds91c1f"
}
```

Request Example (with redirect):

```
GET https://example.com/v1/oauth/authorize?client_id=Example+Responsive+App&response_type=c
ode&scope=info+vroc+ssd&state=x85ds91c1f&redirect_uri=https://example.com/responseApp
```

```
# After successful authorization
Status Code: 302 Found
Location: https://example.com/responseApp#code=7ef0a9fa56e7255c282e21da7074c071994a02f7e9b5
7dfa68a6bdc44a754c36a6ba600abd9bf3a6cd1e30cc1f13adfcffea81b9280a50f4a53603826c06d8f3&state=
x85ds91c1f
```

```
# Error response (e.g. invalid credentials)
Status Code: 302 Found
Location: https://example.com/responseApp#error=access_denied&errorDescription=Invalid+user
name+or+password&state=x85ds91c1f
```

```
# Error response for invalid client id or redirect URI
Status Code: 401 Unauthorized
{
    "error": "invalid_client",
    "errorDescription": "Invalid client identificator",
    "state": "x85ds91c1f"
}
```

4.2.2 Authorization request for access token

[POST] /v1/oauth/authorization_code		
Gets access token based on authorization code.		
Request parameters		
client_id	string	Identifier of a registered client.
grant_type	string	Must be set to: **authorization_code.**
code	string	Token obtained from the authorization server to access resources. See response of authorization request for code grant.

redirect_uri	string	*[Optional parameter]* Uri to which a user will be redirected after completing the authorization process. If provided, responses are embedded in HTTP Response. If not, JSON object is returned.
state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Using this field is recommended.
Response parameters		
access_token	string	Token obtained from the authorization server to access resources.
domain	string	Domain provided during authorization.
expires_in	integer	Time in seconds before access token is flagged as expired. Configurable in the configuration file.
refresh_token	string	Token obtained from the authorization server to refresh expired access token.
scope	string	List of granted scopes. Not all scopes can be provided to the user in the configuration file and that is why response list can differ from request list.
state	string	*[Optional parameter]* An opaque value used by the client to maintain the state between the request and the callback to prevent cross-site request forgery. Returned if request had **state** field.
token_type	string	Type of a token returned within request to attach it with all other requests ("Authorization": "token_type access_token").
username	string	Username provided during authorization.
Available Errors		
invalid_request	400	Required field is missing, field type mismatch, not allowed field is present or field's value does not meet specific requirements (e.g. username is too long).
unsupported_grant_type	400	grant_type is not equal "authorization_code".
invalid_client	401	Client ID does not exist.
invalid_grant	400	Provided code is not found or code is expired or provided client ID or redirect URI mismatch with values from "authorize" request.
server_error	500	An unexpected error occurred during authentication process.

Request Example (no redirect):

```
Content-Type: application/x-www-form-urlencoded
POST https://example.com/v1/oauth/authorization_code?code=d414ca80a308756a4bb5da6fe594e6352
91396e3201eb6bcfbe86188dd138987703c401f5daca1969c2cb011182b3e82e97f39e17ca36179ec846d1fad64
badb&state=x85ds91c1f
```

```
# Response on success
{
    "access_token": "f9aced6bf9cc1890b70d92fa7dd92786f36dea6b543ded9670b149d3883e0fcf031b3e
41fb1bc31afa88fc861c6678d600ed69b037bd8e38b631ee8709d78964",
    "domain": "corp.example.com",
    "expires_in": 3600,
    "refresh_token": "974a1c4ee30bec33081db7ba7f0418b559061e68b6f90c7719a44469b283905a5adf5
b79ad04141fe8beefe6e7dbfc302a269ad3d4a6a1e3266195defad20d19",
    "scope": "info+vroc+ssd",
    "state": "x85ds91c1f",
    "token_type": "Bearer",
    "username": "root"
}
```

```
# Error response
Status Code: 400 Bad Request
{
    "error": "invalid_grant",
    "errorDescription": "Invalid authorization code",
    "state": "x85ds91c1f"
}
```

Request Example (with redirect):

```
Content-Type: application/x-www-form-urlencoded
POST https://example.com/v1/oauth/authorization_code?code=d414ca80a308756a4bb5da6fe594e6352
91396e3201eb6bcfbe86188dd138987703c401f5daca1969c2cb011182b3e82e97f39e17ca36179ec846d1fad64
badb&state=x85ds91c1f&redirect_uri=https://example.com/responseApp
```

```
# Response on success
Status Code: 302 Found
Location: https://example.com/responseApp#access_token=f9aced6bf9cc1890b70d92fa7dd92786f36d
ea6b543ded9670b149d3883e0fcf031b3e41fb1bc31afa88fc861c6678d600ed69b037bd8e38b631ee8709d7896
4&domain=&expires_in=3600&refresh_token=974a1c4ee30bec33081db7ba7f0418b559061e68b6f90c7719a
44469b283905a5adf5b79ad04141fe8beefe6e7dbfc302a269ad3d4a6a1e3266195defad20d19&scope=info%2B
vroc%2Bssd&state=x85ds91c1f&token_type=Bearer&username=root
```

```
# Error response
Status Code: 302 Found
Location: https://example.com/responseApp#error=invalid_grant&errorDescription=Invalid+auth
orization+code&state=x85ds91c1f
```

```
# Error response for invalid client id or redirect URI
Status Code: 401 Unauthorized
{
    "error": "invalid_client",
    "errorDescription": "Invalid client identificator",
    "state": "x85ds91c1f"
}
```

4.3 Refreshing access token

[POST] /v1/oauth/refresh_token		
If the authorization server generates refresh token, client application can use it in a refresh request to get a new access token without providing any credentials.		
Request parameters		
grant_type	string	Must be set to: **refresh_token**.
refresh_token	string	Token obtained from the authorization server to refresh expired access token.
scope	string	*[Optional parameter]* List of scopes of the token separated by +. Each scope is in form [IP@]scope, where IP is an address of the resource machine (which is optional). Scopes are defined in plugins documentations. Must not include any new scope not originally granted by the resource owner. This can be used to remove one of the scopes from the client in the session.
Response parameters		
access_token	string	Token obtained from the authorization server to access resources.
expires_in	integer	Time in seconds before access token is flagged as expired. Configurable in the configuration file.
refresh_token	string	Token obtained from the authorization server to refresh expired access token.
scope	string	List of granted scopes.
token_type	string	Type of a token returned within request to attach it with all other requests ("Authorization": "token_type access_token").

Available Errors		
invalid_request	400	Required field is missing, field type mismatch or not allowed field is present.
unsupported_grant_type	400	grant_type is not equal "refresh_token".
invalid_scope	400	The new scope include at least one not originally granted by the resource owner.
invalid_grant	400	Provided refresh token is not found or is expired.
server_error	500	An unexpected error occurred during authentication process.

Request Example:

```
Content-Type: application/x-www-form-urlencoded
POST https://example.com/v1/oauth/refresh_token?refresh_token=de4955a423a51b8e4ffc7ad2a2918ccafd1660a14c4560fbac25d8cd317dbd83292cfd7bdefd28fde226e1ce60a41085dd4d263044bd6091c3194bfa8743db67&grant_type=refresh_token&scope=info+ssd
```

```
# Response on success
{
    "access_token": "6501988d9b07f71fc6160ed2220bd8a0263195bb13e3a21bfc7265acd78fa7413bb921
bc185af00b615eff59177e261dee37daba5301597cd50af13058016dba",
    "expires_in": 3600,
    "refresh_token": "75750a294c721b94fa8f587e9324f445623e6a07093c650c9fb4f69c445a60ef4e6fa
d34d94d7120d52b1afd5fcdb46ce2ddaf3dd11eacb29b2554286956af5b",
    "scope": "info+ssd",
    "token_type": "Bearer"
}
```

```
# Error response
Status Code: 400 Bad Request
{
    "error": "invalid_scope",
    "errorDescription": "The requested scope exceeds the scope granted originally"
}
```

4.4 Resource owner password credentials grant

Grant type used when client application has special privilege or trust with authorization server. In this method, username and password are send directly to authorization server.

4.4.1 Authorization request

[POST] /v1/oauth/password_credentials		
Request access token from authorization server using password credentials method. Useful for scripts.		
Request parameters		
grant_type	string	Must be set to: **password**.
username	string	User login in the local system or in domain.
password	string	User password in the local system or in domain.
domain	string	*[Optional parameter]* Domain of the user account. When empty, client logs into local system account.
scope	string	List of scopes of the token separated by +. Each scope is in form [IP@]scope, where IP is an address of the resource machine (which is optional). Scopes are defined in plugins documentations.
encoded	flag	*[Optional parameter]* When present, password must be encoded in base64. Value is ignored.
Response parameters		
access_token	string	Token obtained from the authorization server to access resources.
domain	string	Domain of the authorized user account.
expires_in	integer	Time in seconds before access token is flagged as expired. Configurable in the configuration file.
refresh_token	string	Token obtained from the authorization server to refresh expired access token.
scope	string	List of granted scopes.

token_type	string	Type of a token returned within request to attach it with all other requests ("Authorization": "token_type access_token").
username	string	Authorized user login.
Available Errors		
invalid_request	400	Required field is missing, field type mismatch, not allowed field is present or field's value does not meet specific requirements (e.g. username is too long).
unsupported_grant_type	400	grant_type is not equal "password" or "password" grant is not available.
invalid_scope	400	Every provided scope is invalid.
invalid_grant	400	Login or password is not correct or reached failed login attempts or user does not belong to provided domain.
server_error	500	An unexpected error occurred during authentication process.

Request Example:

```
Content-Type: application/x-www-form-urlencoded
POST https://example.com/v1/oauth/password_credentials?grant_type=password&username=root&password=ZXhhbXBsZQ%3D%3D&scope=info+vroc+ssd&encoded=
```

```
# Response on success
{
    "access_token": "9cdf0b94f1bdfcba253e6907825644ea83a7e31c68fb817d6afabd4595c77977fd2379c01da564e5e4ad1288c664921f9d83396fd628695e490d766e56bd7313",
    "domain": "",
    "expires_in": 3600,
    "refresh_token": "0ee113704ba75a0c62027eb1c91575aa45bf9c25eb1296d50b6bb687b72fdf3d62e82e471cd24b022e37a8df505c9956b1419e0542e133cfba7ce39883f590c7",
    "scope": "info+vroc+ssd",
    "token_type": "Bearer",
    "username": "root"
}
```

```
# Error response
Status Code: 400 Bad Request
{
    "error": "invalid_grant",
    "errorDescription": "Invalid username or password"
}
```

5. Using access token

After obtaining a valid access token, user or application should provide an access token along with a request. An access token should be included in query parameter or in the Authorization header.

5.1 Example – Access token in query parameter

```
https://example.com/v1/info/version?access_token=39ecee8b79216d98259a46f0dff561acdfd17b8fcb45762dd1b4340c5bd145d6dcec4b6f24c1b85c2a4dd21661a9cfd3dad9929b2c5e592d27ab685bb379dd22
```

After this request, the user will obtain information about the version of the IASM, if the access token is valid.

5.2 Example – Access token in the Authorization header

```
GET https://example.com/v1/info/version
Authorization: Bearer 39ecee8b79216d98259a46f0dff561acdfd17b8fcb45762dd1b4340c5bd145d6dcec4b6f24c1b85c2a4dd21661a9cfd3dad9929b2c5e592d27ab685bb379dd22
```

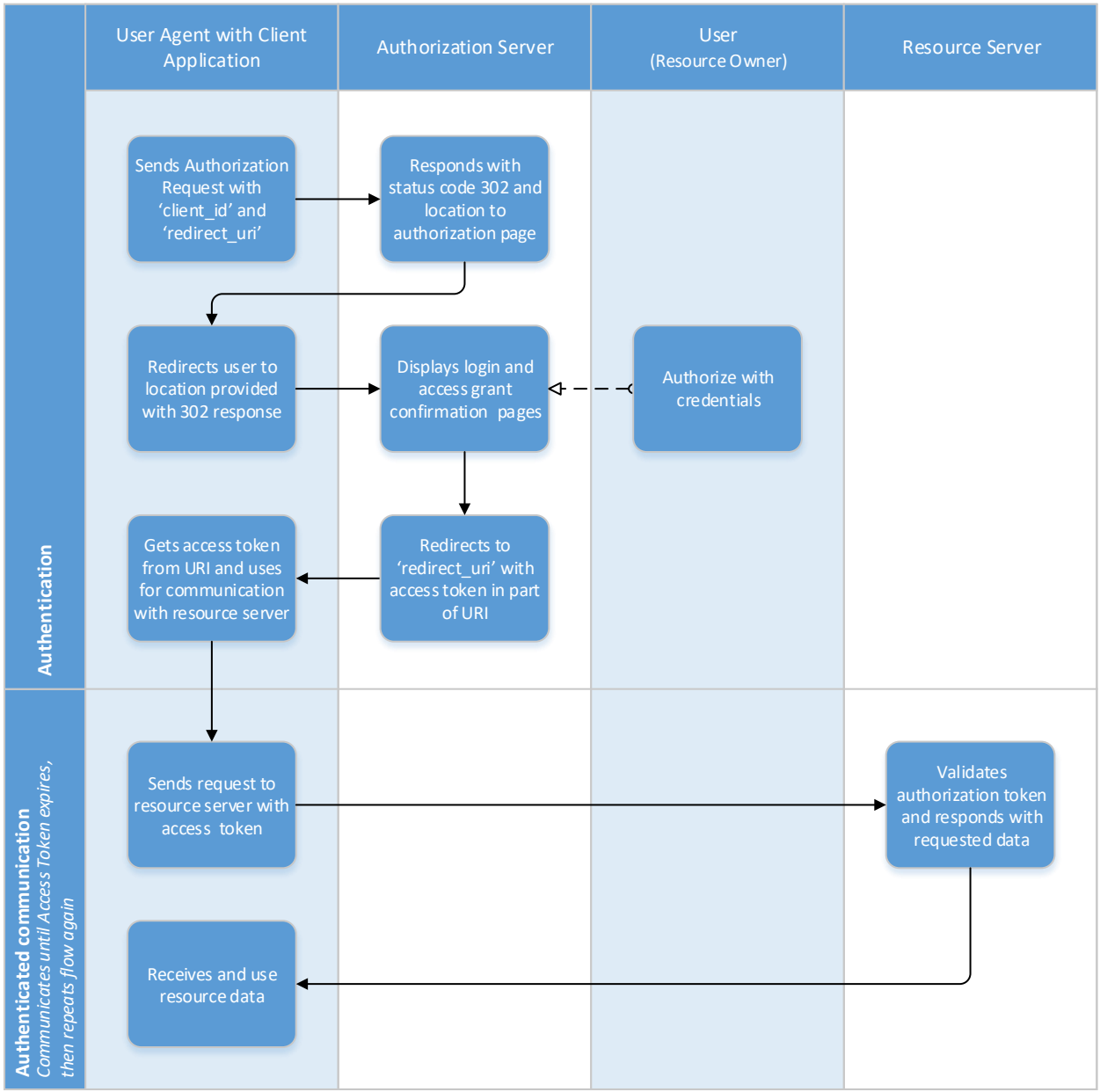
5.3 Error handling

If the user will not provide an access token or will provide not an existing access token or a token that has expired already, appropriate HTTP error with error code 401 will be returned. Information will be in WWW-Authenticate header accordingly to RFC.

6. Authorization Methods Flow

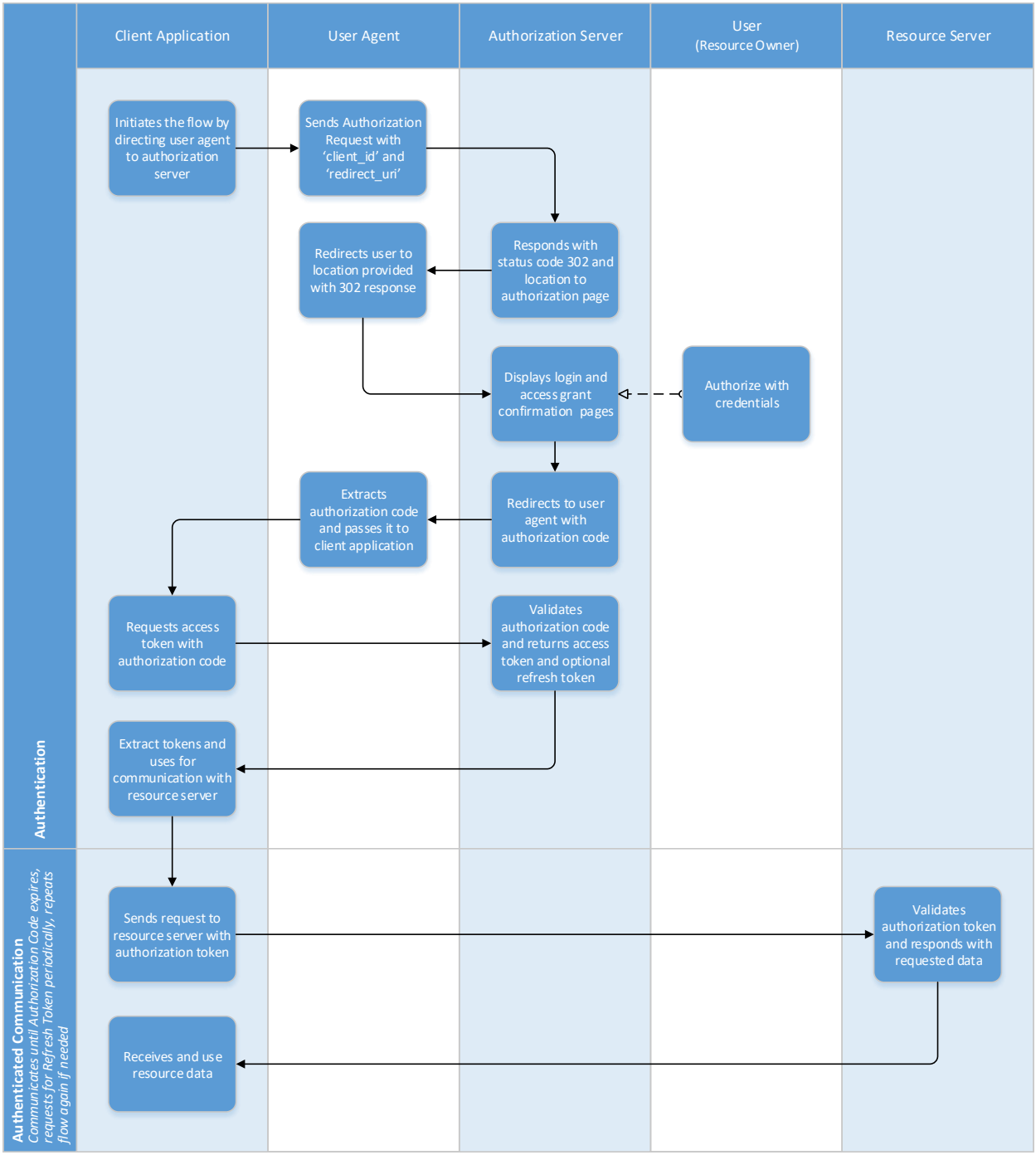
6.1 Implicit Grant Flow

Detailed information about this grant type can be found in RFC6749 in section 4.2.



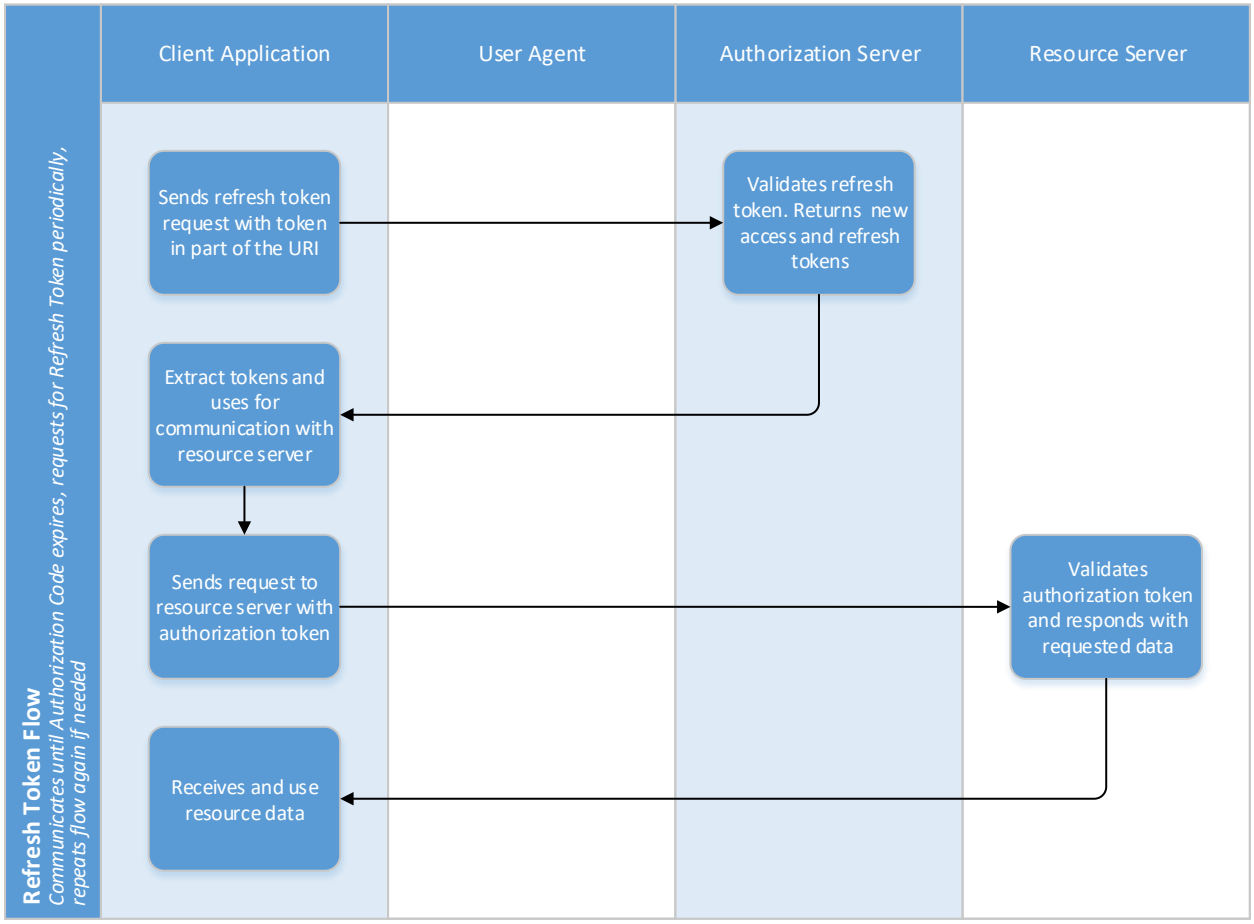
6.2 Code Grant Flow

Detailed information about this grant type can be found in RFC6749 in section 4.1.


Client Application
User Agent
Authorization Server
User (Resource Owner)
Resource Server
Initiates the flow by directing user agent to authorization server
Sends Authorization Request with 'client_id' and 'redirect_uri'
Redirects user to location provided with 302 response
Responds with status code 302 and location to authorization page
Displays login and access grant confirmation pages
Authorize with credentials
Extracts authorization code and passes it to client application
Redirects to user agent with authorization code
Requests access token with authorization code
Validates authorization code and returns access token and optional refresh token
Extract tokens and uses for communication with resource server
Authentication
Sends request to resource server with authorization token
Validates authorization token and responds with requested data
Receives and use resource data
Authenticated Communication
Communicates until Authorization Code expires, requests for Refresh Token periodically, repeats flow again if needed

6.3 Code Grant – Token Refreshing Flow

Detailed information about refreshing token can be found in RFC6749 in section 6.



6.4 Resource Owner Password Credentials Grant Flow

Detailed information about this grant type can be found in RFC6749 in section 4.3.