INTERNATIONAL STANDARD

ISO 20078-3

Second edition
2021-11

Road vehicles — Extended vehicle (ExVe) web services —

Part 3: Security

Véhicules routiers — Web services du véhicule étendu (ExVe) —

Partie 3: Sécurité

Reference number
ISO 20078-3:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles,* Subcommittee SC 31, *Data communication*.

This second edition cancels and replaces the first edition (ISO 20078-3:2019), which has been technically revised.

The main changes are as follows:

— defined authorization domains for the offering party and the accessing party;

— added new requirements and description related to push method to make the offering party authorized to push resources to the accessing party;

— added Annex B containing description of reference implementation for push.

A list of all parts in the ISO 20078 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Road vehicles — Extended vehicle (ExVe) web services —

Part 3: Security

1 Scope

This document defines how to authenticate users and accessing parties on a web-services interface. It also defines how a resource owner can delegate access to its resources to an accessing party. Within this context, this document also defines the necessary roles and required separation of duties between these in order to fulfil requirements stated on security, data privacy and data protection.

All conditions and dependencies of the roles are defined towards a reference implementation using OAuth 2.0[1] compatible framework and OpenID Connect 1.0[2] compatible framework.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 20078-1, *Road vehicles — Extended vehicle (ExVe) web services — Content and definitions*

3 Terms and definitions

For the purposes of this document, the convention, terms and definitions given in ISO 20078-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
identity token
ID token
digitally signed JWT and contains *claims* (3.3) about the authenticated resource owner

3.2
authorization code
intermediate result of a successful resource-owner authorization process and that is used by authorized clients to obtain access tokens and optionally refresh tokens

3.3
claim
asserted information about a certain entity

EXAMPLE ROID, resource owner's first name, last name, address, connected vehicle's capability and/or other attributes.

3.4
token issuer
entity that generates and provides *identity tokens* (3.1), access tokens, and refresh tokens

3.5
authorization domain
domain of activity where an entity controls the authorization

Note 1 to entry: The offering party controls the authorization in the offering-party authorization domain and the accessing party controls the authorization in the accessing-party authorization domain.

Note 2 to entry: Due to the description of push communication in 5.2 there exists two different authorization providers. One at the side of the accessing party and one at the side of the offering party.

4 General

4.1 Processes

The following processes are specific to each offering party. The definition of these processes is not part of this document but shall be in place in order to apply this specification.

REQ_04_01_01	The process to register a resource owner at the identity provider shall be the responsibility of the offering party.
REQ_04_01_02	The process to register an accessing party at the authorization provider shall be the responsibility of the offering party.
REQ_04_01_03	The process to confirm the technical eligibility of connected vehicles and provision of their associated ExVe resources shall be the responsibility of the offering party.
REQ_04_01_04	The process to verify a resource owner's current and valid ownership of the concerned resource shall be the responsibility of the offering party.
REQ_04_01_05	The process to register the offering party in the accessing party authorization domain shall be the responsibility of the accessing party. This is only needed if resources shall be pushed.

4.2 Conditions

REQ_04_02_05	The offering party shall be able to restrict or deny the accessing party and/or the resource owner access to the offering party's web services and portals.

NOTE 1 This can be done, for example, to fulfil security and legislation requirements.

REQ_04_02_06	If the offering party revokes a granted registration of an accessing party, the offering party may delete all containers created by the accessing party, if containers are used.

NOTE 2 Revocation of the registration can be due to access violation or other misuse of the web services.

REQ_04_02_07	The accessing party shall be able to restrict or deny the offering party's ability to push resources.

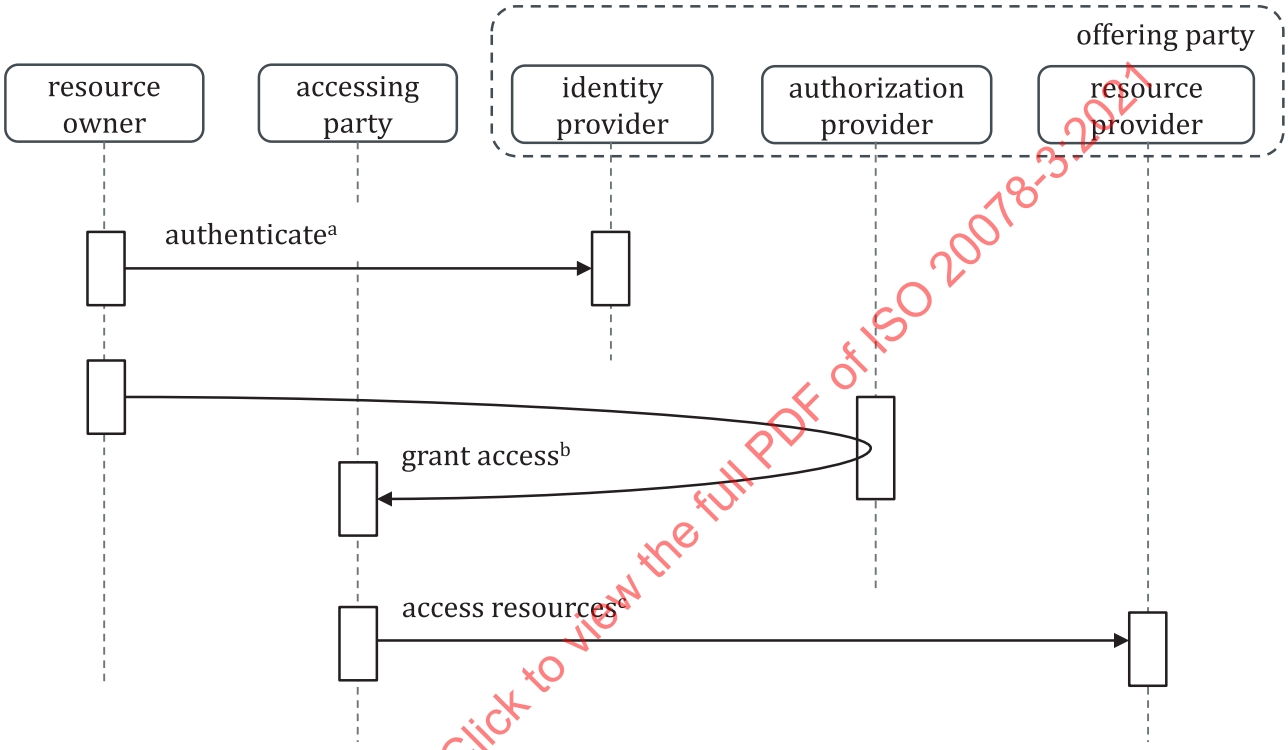
NOTE 3 This can be done, for example, to fulfil security and legislation requirements.

5 Basic communication flow

5.1 Offering party authorization domain

5.1.1 General

This document separates the activities necessary for authentication, authorization and resource access into three distinct communication flows with separate duties (see Figure 1).



[a] Step 1: the resource owner is authenticated by the identity provider.

[b] Step 2: the resource owner is granting access to the accessing party. The granting is handled by the authorization provider.

[c] Step 3: the accessing party is accessing resources from the resource provider.

Figure 1 — The roles and the three distinct communication flows

5.1.2 Authentication

The identity provider is responsible for authenticating the resource owner and managing the resource owner profile, based on the resource owner registration. The resource owner credentials are revealed only to the identity provider, and the identity provider confirms a successful authentication to concerned parties. If the resource owner has given consent, the accessing party will be authorized to access the resource owner's profile (Figure 2).

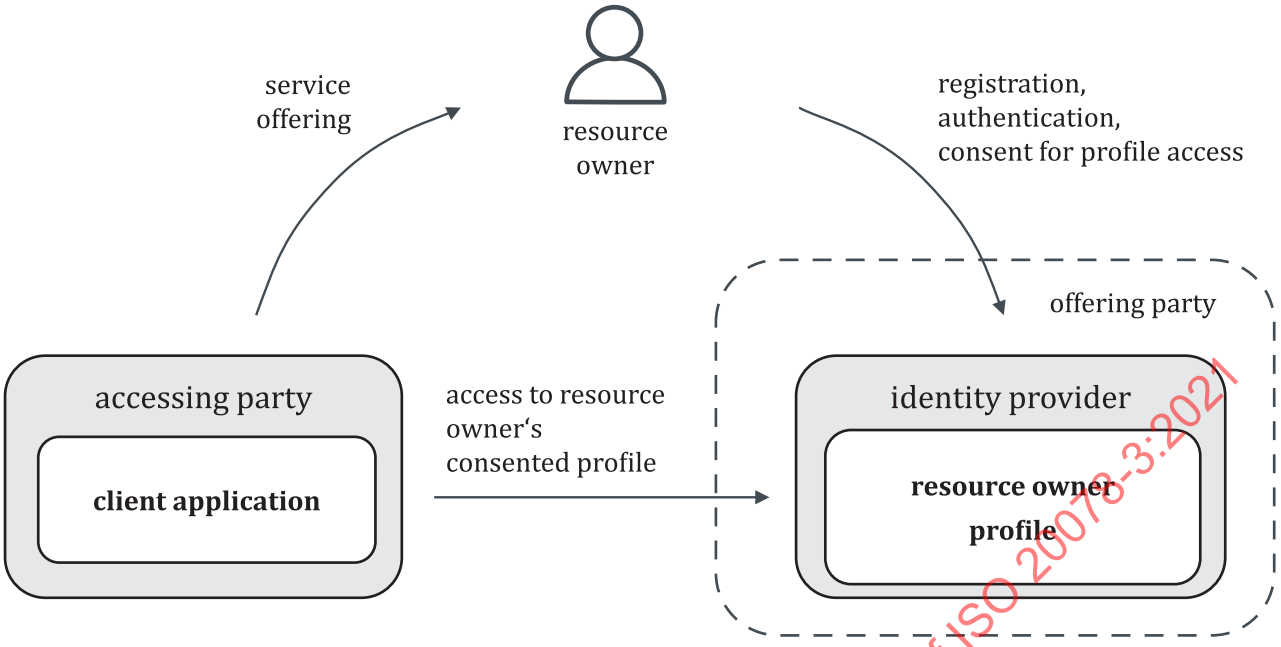


Figure 2 — Resource owner authentication and access to resource owner's profile

REQ_05_01_01	The identity provider shall offer a suitable authentication method and shall perform the authentication process. After a successful authentication, the identity provider shall confirm the identity of the authenticated resource owner.

REQ_05_01_02	The resource owner's credentials shall only need to be known by the resource owner and be possible to be verified by the identity provider.

REQ_05_01_03	The resource owner's registration and authentication (at the identity provider), shall be separated from the authorization process to grant access to resources (via the authorization provider).

REQ_05_01_04	If the identity provider is able to expose the resource owner's profile to the accessing party, it is only the resource owner that shall be able to grant or deny access.

5.1.3 Authorization

The client application as a component of the accessing party requires access to resources on behalf of the resource owner. At the authorization step, the accessing party requests authorization to access the resources provided by the resource provider (offering party). The required authorization is requested at the authorization provider, providing the intended scope. By the consent of the resource owner, the authorization provider returns a limited authorization to the client application of the accessing party. Using the obtained authorization, the client application can access resources. authorization to access resources is done in the same way regardless, if the resources are fetched by the accessing party using request/reply or pushed by the offering party (see Figure 3). See ISO 20078-2 for details regarding request/reply and push.

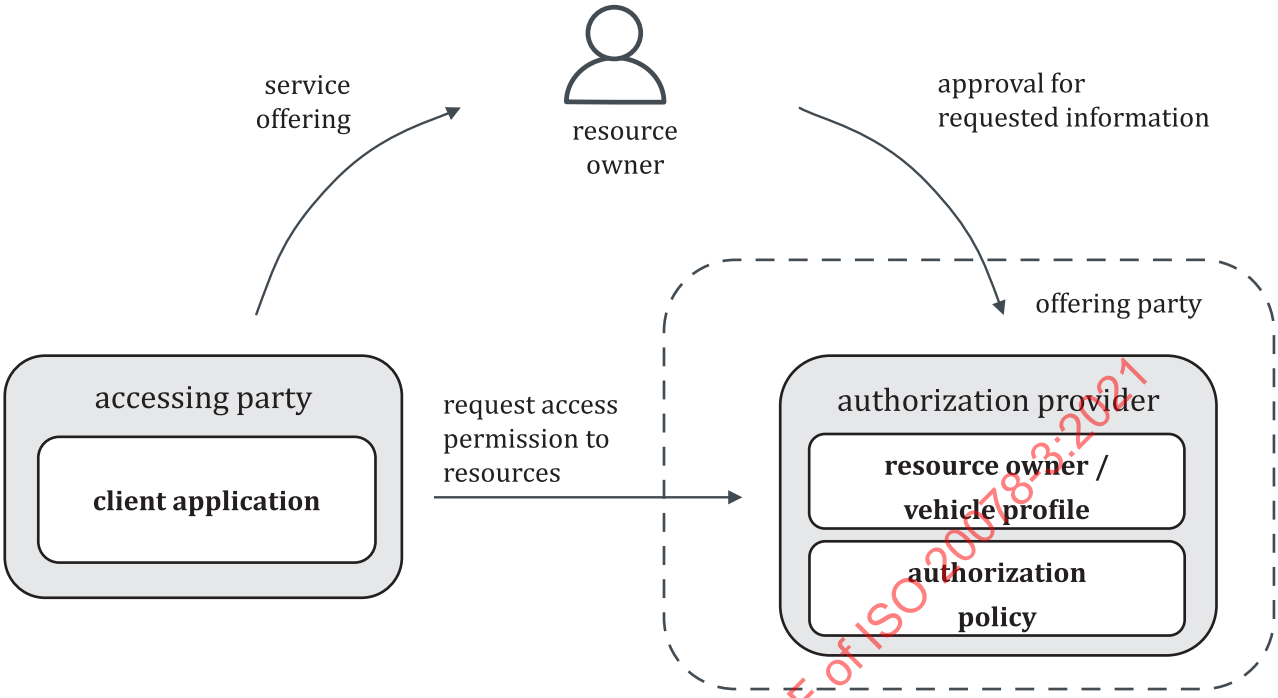


Figure 3 — Requesting access to resources

REQ_05_01_05	Before accessing the resource, the accessing party shall request access at the authorization provider providing the intended scope.

REQ_05_01_06	The authorization provider shall be responsible for the management of the authorization policy and shall manage all granted accesses.

NOTE 1 The authorization policy, for example, defines the permissions of the accessing party, primarily the conditions to be met for granted access to resources.

REQ_05_01_07	The authorization provider shall trust the confirmation of successful authentication as provided by the identity provider.

REQ_05_01_08	The authorization policy shall be defined by the offering party concerning the authorization process.

REQ_05_01_09	The authorization provider shall be able to verify the relationship between resource owners and their resources.

REQ_05_01_10	Only the resource owner shall be able to grant access to a resource.

NOTE 2 The access is granted to an accessing party at the offering party.

REQ_05_01_11	Granting access to resources shall be done either directly or via containers. The offering party decides if one or both of the granting methods shall be provided to the accessing parties.

REQ_05_01_12	The resource owner shall be able to revoke a granted access to a resource at any time.

REQ_05_01_13	If containers are used, the resource owner shall be able to revoke a granted access to a containers at any time.

REQ_05_01_14	The authorization provider shall ask the resource owner for the approval before providing the authorization to the accessing party resulting in a granted access.

REQ_05_01_15	Upon request the offering party shall present a resource owner's granted accesses to the resource owner.

REQ_05_01_16	The resource owner shall be able to deny an access request to a resource, or if containers are used, to a container at any time.

REQ_05_01_17	If the ownership of a resource or the relationship between the resource owner and the resource ends, access to the corresponding resources, and if containers are used, also to containers, shall be revoked.

REQ_05_01_18	If containers are used and if a container is deleted, all access granted to that container shall be revoked.

5.1.4 Resource access

Using the access, the accessing party can access the resources, hosted by the resource provider (see Figure 4).

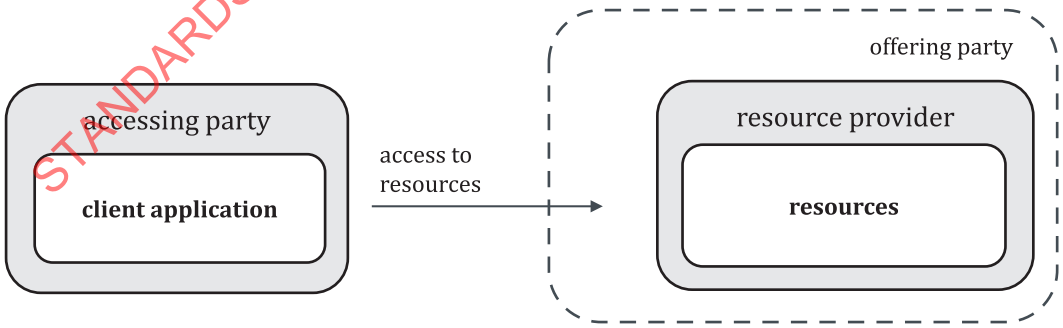


Figure 4 — Access to resources via the resource provider

REQ_05_01_19	The resource provider shall perform access control to the resources according to the authorization policy.

5.1.5 Separation of duties

Separation of duties concerns the separation of tasks and responsibilities between entities involved in the authentication, authorization and access to resources.

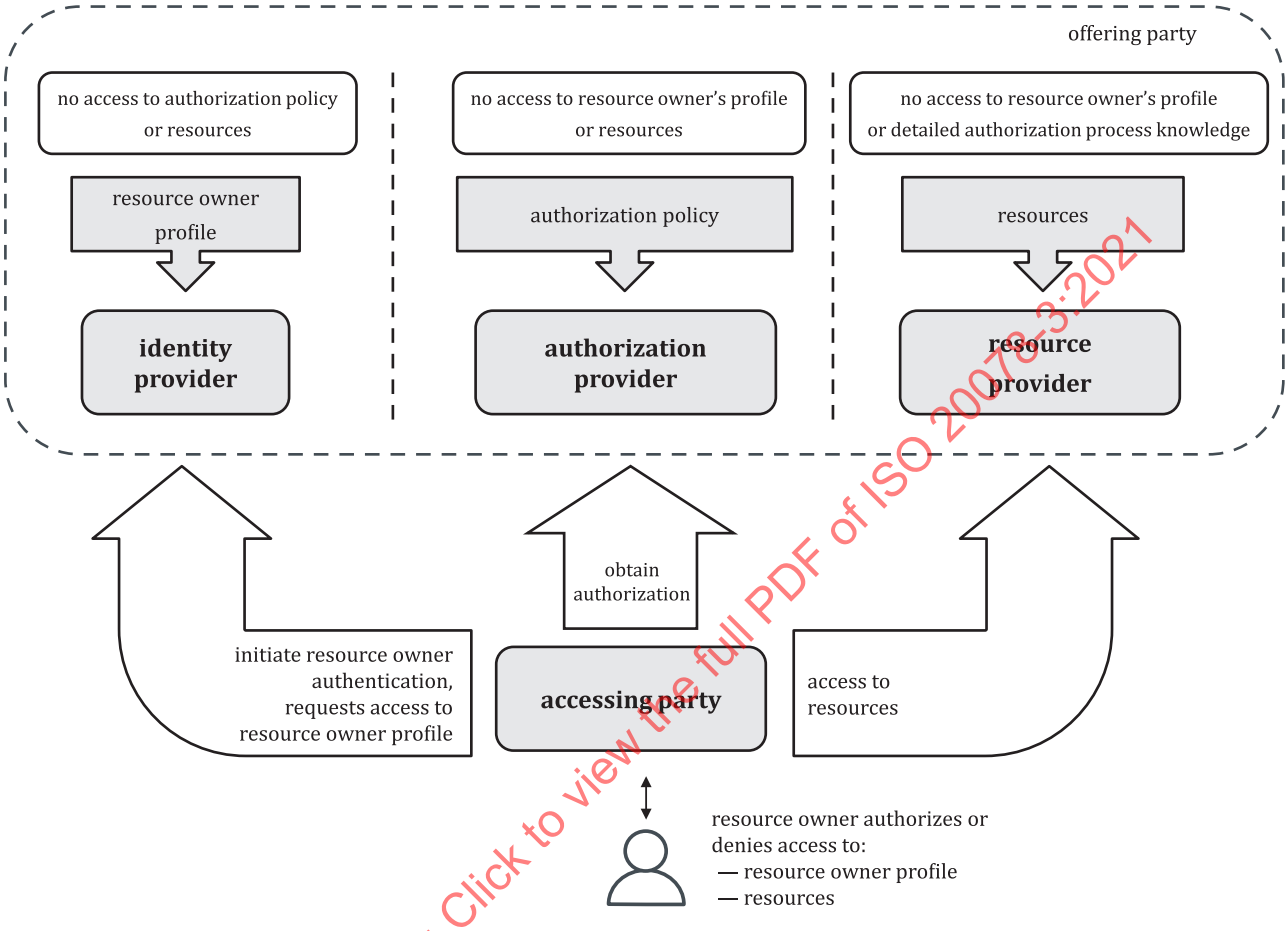


Figure 5 — Separation of duties between involved roles

Figure 5 describes the separation of duties between involved roles, where the offering party has the three roles: identity provider, authorization provider, and resource provider.

REQ_05_01_20	The identity provider shall not be dependent on the authorization policy.

REQ_05_01_21	The identity provider shall not influence the authorization policy.

REQ_05_01_22	The identity provider shall not access the resources.

REQ_05_01_23	The authorization provider shall not access the resource owner profile.

REQ_05_01_24	The authorization provider shall only use the unique ResourceOwnerID to identify the resource owner.

NOTE 1 The ResourceOwnerID is generated and communicated by the trusted identity provider.

REQ_05_01_25	The authorization provider shall not have access to resources provided by the resource provider.

REQ_05_01_26	The resource provider shall not access the resource owner profile.

REQ_05_01_27	The resource provider shall not know details about the authorization process.

REQ_05_01_28	The resource provider trusts the authorization provider and shall verify whether the provided authorization matches the access control rules defined for the requested resources.

REQ_05_01_29	The resource owner shall not need to share credentials with the accessing party to enable the accessing party to access the resources.

REQ_05_01_30	The accessing party shall only access the resources with the consent of the resource owner.

NOTE 2 The requirements stated above do not impose requirements on specific architecture, design or organizational structure.

Figure 6 shows the major logical components of the involved roles and the associated entities.

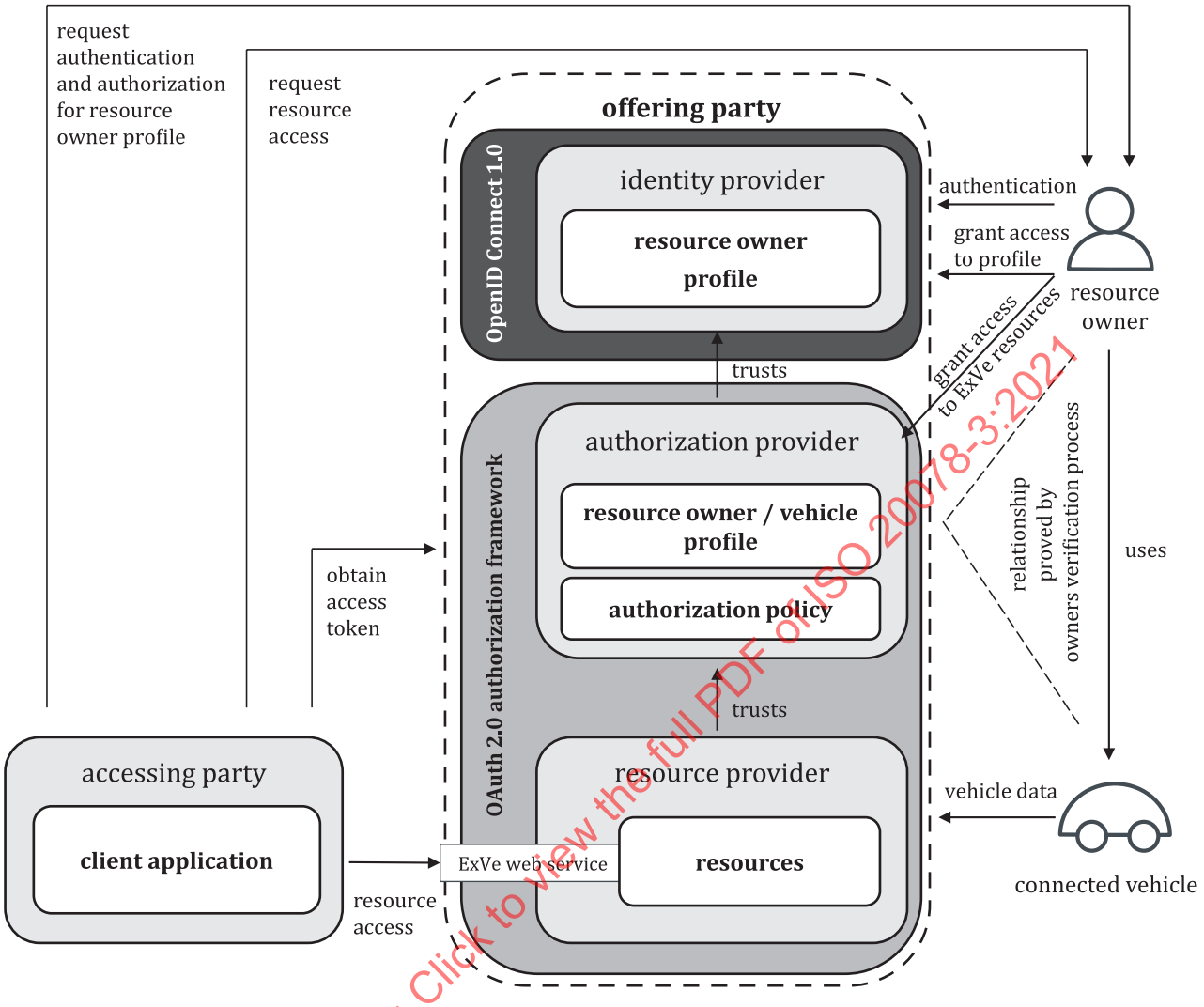


Figure 6 — Involved roles and associated entities

5.1.6 Implementation related considerations

The physical implementation and assignment of roles to real parties differs from the logical representation as shown in Figure 6, and follows the defined requirements as referenced by Figure 7.

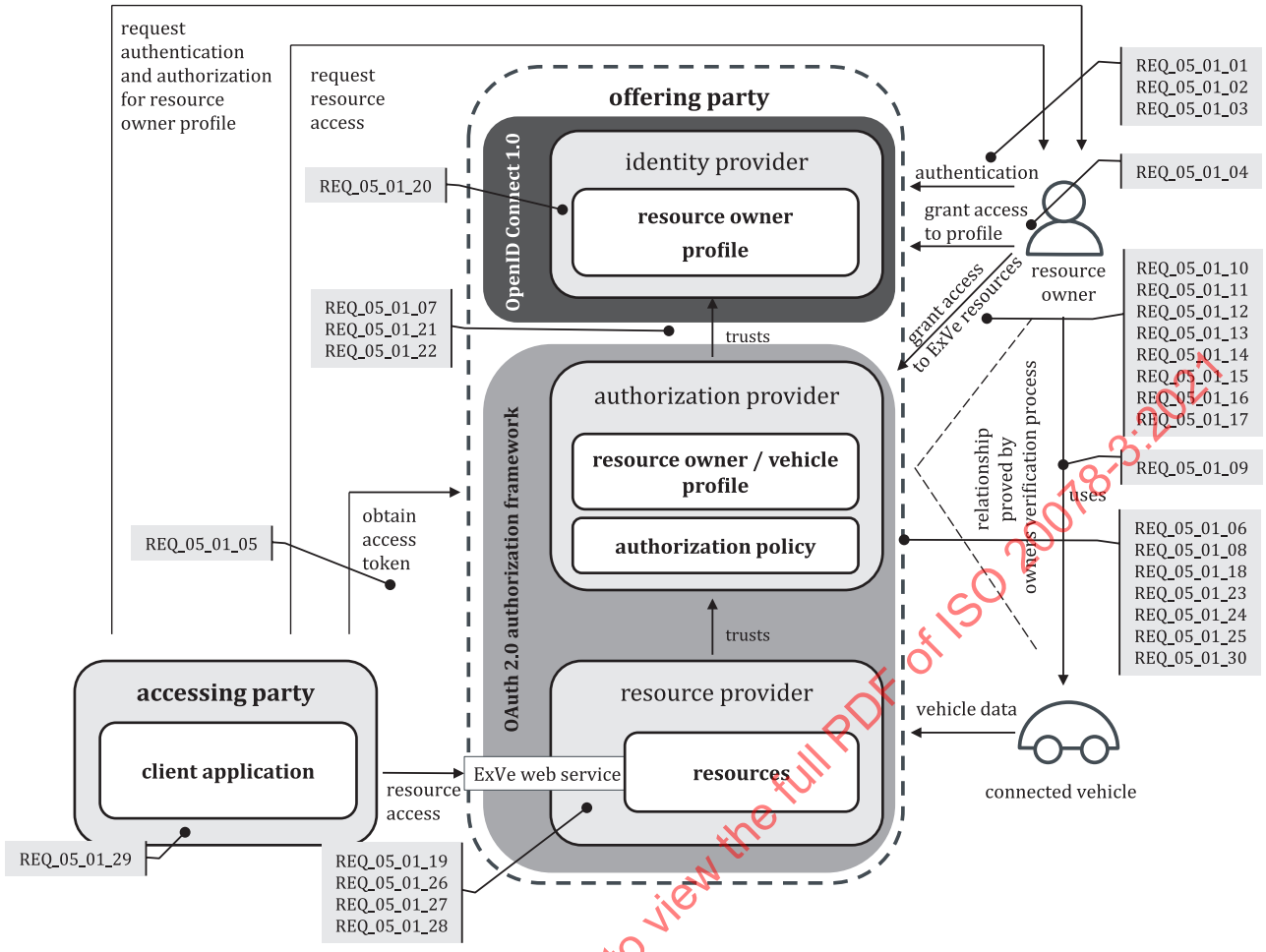


Figure 7 — Logical representation of involved roles and their entities in reference to the defined requirements

Additionally, actual implementation often depends on national legal requirements (e.g. handling of resource owner profile, implemented resource owner's verification process etc.) and the required trusted relationship between involved components especially identity provider, authorization provider, and resource provider.

REQ_05_01_31	All communication paths between involved entities shall use secured connections.

REQ_05_01_32	The identity provider, authorization provider, and resource provider are responsible for ensuring that only recent cipher suites are used.

NOTE 1 Changes in the interface are communicated to accessing parties within a reasonable notice period.

If the offering party encounters an unreliable accessing party, the offering party can temporarily or permanently revoke the accessing party's access. This is done in order to protect the resource owners. Examples of circumstances that could trigger this are: insecure smartphone applications, disabled host verification, data breach of database, forbidden caching or storage of resource data, usage of discouraged security algorithms.

REQ_05_01_33	It shall be possible to validate the authenticity and integrity of information provided by the identity provider, authorization provider and resource provider.

REQ_05_01_34	To ensure the interoperability between involved entities in different physical environments, an implementation shall follow a framework compatible with OAuth 2.0 and OpenID Connect 1.0.

Annex A provides one example of how to implement OAuth 2.0 and OpenID Connect 1.0.

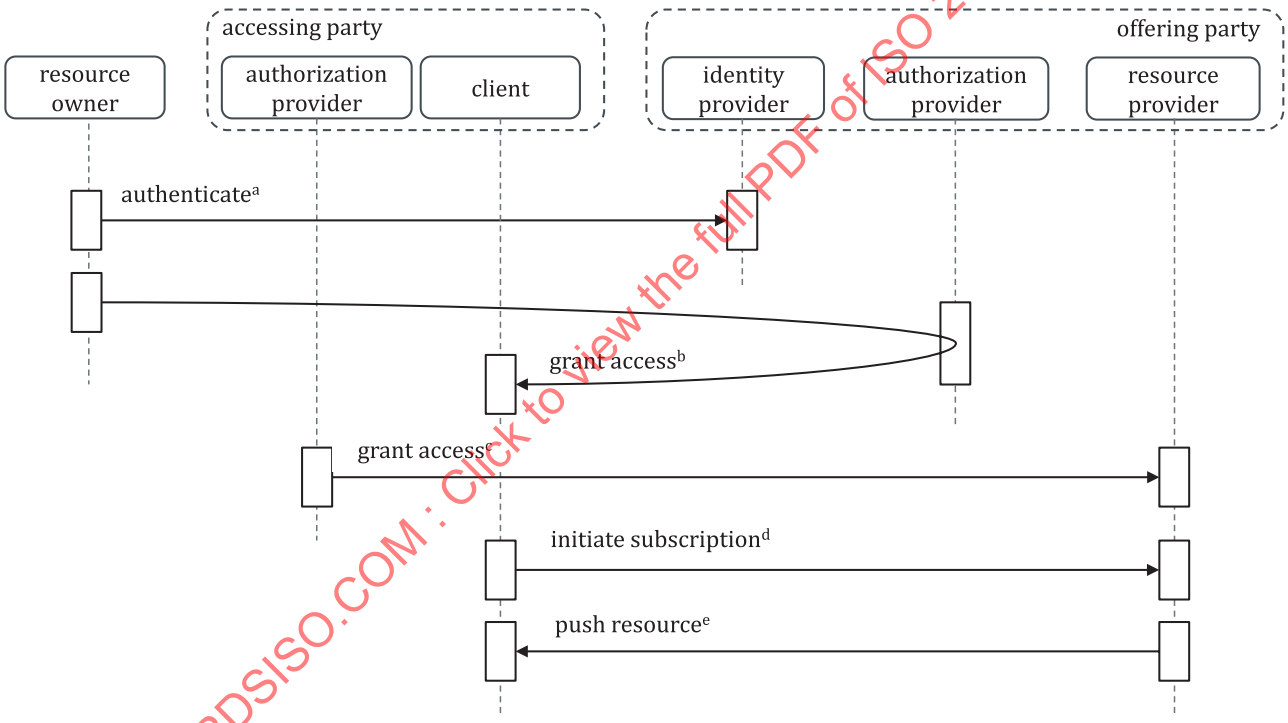
5.2 Accessing party authorization domain

5.2.1 General

Subclause 5.2 describes the steps needed to make the offering party authorized to push resources to the accessing party. Figure 8 shows an example of the order of these steps. Steps 1 and 2 are already covered in 5.1 and are only included in Figure 8 to show a complete flow.

NOTE 1 In step 5, resource provider is using authorization information provided in step 2.

NOTE 2 In case the offering party is also the resource owner, step 2 is implicit.



a Step 1: the resource owner is authenticated by the identity provider (see 5.1).

b Step 2: the resource owner is granting access to the accessing party. The granting is handled by the authorization provider of the offering party (see 5.1).

c Step 3: the accessing party grants access to the offering party to push resources.

d Step 4: the accessing party initiates the subscription, either explicitly or through other means.

e Step 5: the offering party pushes the resource to the accessing party.

Figure 8 — Overview of communication flows

5.2.2 Authorization

The resource provider as a component of the offering party requires access to push resources to the client application of the accessing party.

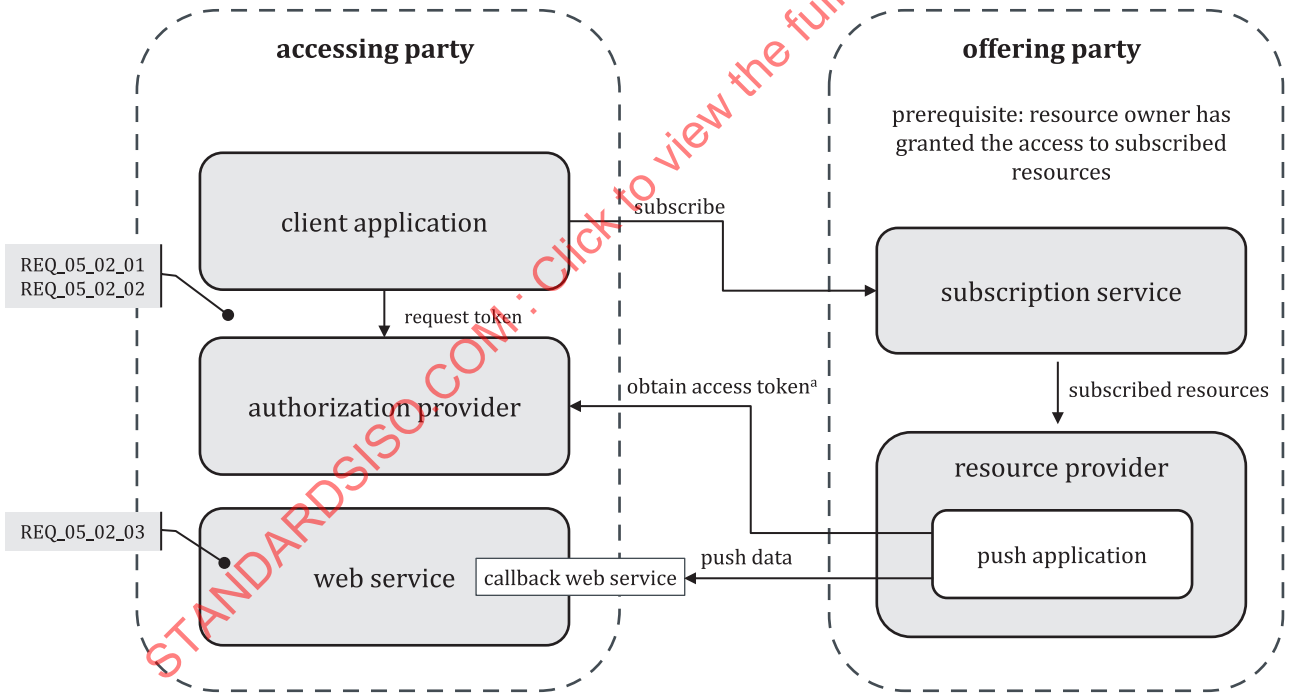
REQ_05_02_01	The accessing party shall authorize the offering party to push resources when there is an active subscription.

REQ_05_02_02	The accessing party shall provide a bearer token and expiry time to the offering party. Or The accessing party shall provide a refresh token and expiry time to the offering party. The refresh token can be used to renew the access token. Either one or both of these options shall be implemented by the offering party when pushing resources.

5.2.3 Pushing resources

The offering party uses the bearer token to be authorized at the accessing party when pushing resources (see Figure 9).

REQ_05_02_03	The accessing party shall verify the bearer token before accepting the pushed resource(s).



[a] Request of a new access token is required if refresh token grant type is used in the subscription.

Figure 9 — Logical representation of push method in reference to the defined requirements

Annex B provides reference implementation for push.

Annex A
(informative)

Reference implementation using OAuth 2.0 and OpenID Connect 1.0

A.1 Introduction

This reference implementation is designed in accordance with the general approach (see Clause 4) using OAuth 2.0 framework[1] and OpenID Connect 1.0[2] specifications. OAuth 2.0 is used to implement an authorization mechanism for requesting of authorization and accessing resources. OpenID Connect 1.0 is used as an authentication layer on top of the OAuth 2.0 framework for resource-owner related scenarios, where the proof of the resource-owner identity using appropriate authentication method through an identity provider is required (see Figure A.1).

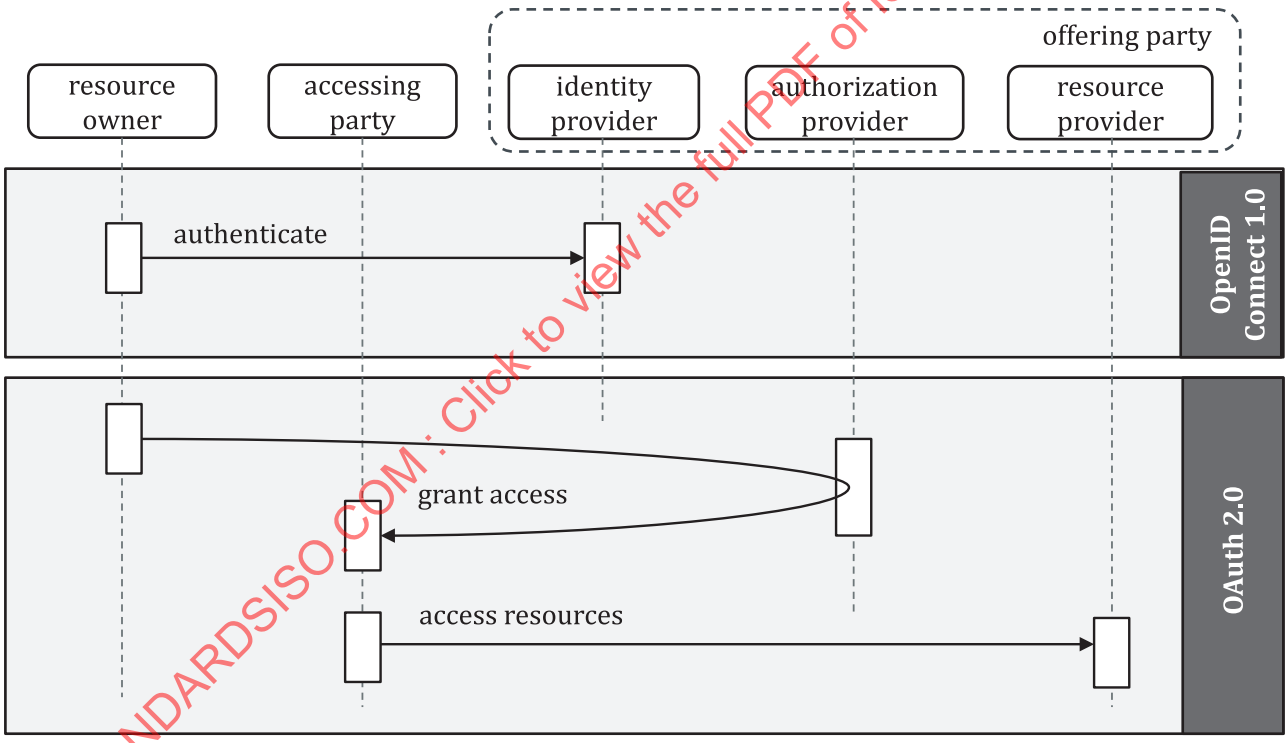


Figure A.1 — Extended vehicle split to the usage of OAuth 2.0 and OpenID Connect 1.0

The client application of the accessing party should support an implementation of the standard OAuth 2.0[1] for authorization requests and access to protected resources, and may support OpenID Connect 1.0[2] for resource owner authentication and access to the profile of the resource owner.

Both standards are using the term "authorization server". However, this document differentiates between **logical** components, the "identity server" maintained by the identity provider and the "authorization server" maintained by the authorization provider. In this reference implementation, the ExVe identity server refers to OpenID Connect 1.0 authorization server and the ExVe authorization server refers to OAuth 2.0 authorization server.

The reference implementation does not cover all of the technical details. The terms and definitions to facilitate the understanding of the referenced implementation are provided in Clause 3.

The implementation of the components should comply with the following guidelines.

— For resource owner authentication, OpenID Connect 1.0 Authorization Code Flow, OIDC Core[2] should be used by the accessing party.

— The identity provider should provide a “UserInfo” end point as defined in OpenID Connect 1.0[2] to make the resource owner profile available.

— OAuth 2.0 grant type "authorization code” is recommended when requesting authorization for protected resources owned by a resource owner, RFC 6749[1]. Offering party and accessing party can agree on other grant types.

— In the authorization code flow, the client application will first get an authorization code which then needs to be exchanged for the identity token (identity provider) or the access token (authorization provider).

— The identity provider and/or the authorization provider may request a registration of the client application before the client application can consume services provided by the identity server and/or the authorization server. With successful registration the client application will receive client credentials. The design of the client registration process, the credential type and the client authentication method are under the responsibility of the identity provider and the authorization provider.

— OAuth 2.0 grant type "client credentials" can be used for resources, where runtime interaction with the resource owner is not required, RFC 6749[1].

— The authorization server and the identity server should provide a service for revocation of granted permissions in accordance with RFC 7009[7].

— The issuer of tokens (identity server, authorization server) may expose OAuth 2.0 token introspection end points according to RFC 7662[4].

— All tokens (identity token, refresh token, access token) should be digitally signed using asymmetric keys as defined in RFC 7515[8]. Allowed algorithms are defined in RFC 7518[5].

— The token issuer should provide all valid public keys for signature validation as defined in RFC 7517[3].

— The access token type should be bearer as defined in RFC 6750[9].

— The access tokens may be self-contained or may reference the authorization information stored at the token issuer. Self-contained access tokens allow the resource server to perform an authorization decision without further interaction with the authorization server. To allow the reliable revocation of self-contained tokens the lifetime should be limited to maximum one hour.

— If issued, the client application should store refresh tokens in a long-term secure storage and continue to use them as long as they remain valid. Refresh tokens should be treated by the clients as a secret and need only be sent exclusively to the issuer of the refresh token.

— Implementers should pay attention to the section Security Considerations in RFC 6749[1], RFC 7517[3], RFC 7662[4], RFC 7518[5], RFC 6819[6], RFC 7009[7], RFC 7515[8], RFC 6750[9], RFC 7636[10].

A.2 Claims

A.2.1 General

For ExVe specific claims the prefix *exve.* can be used.

A.2.2 ID token claims

In addition to required claims defined in OpenID Connect 1.0[2], an ID token can contain the following custom claim:

exve.roid (Unique ResourceOwnerID)

A.2.3 Access token claims

In addition to required claims defined in RFC 7519[11], an access token may contain the following custom claims:

exve.roid (ResourceOwnerID)

exve.cid (ContainerID)

exve.rid (ResourceID)

One or more access IDs are linked to the ResourceOwnerID, ResourceIDs and/or ContainerIDs.

A.2.4 Refresh token claims

In addition to required claims defined in RFC 7519[11], a refresh token should at a minimum contain the following custom claim:

exve.roid (Unique ResourceOwnerID)

NOTE The refresh token is used with the scope to request a new access token, as the refresh token only contains the ResourceOwnerID.

A.3 Use cases

A.3.1 Access to protected resources with resource owner's approval at runtime

The accessing party wants to access resource owner-related resources and an authorization at runtime is required (approval).

This initial access needs four steps:

1) resource owner authentication (and optionally granting access to the resource owner's profile) by the identity provider;
2) obtain basic profile information about the resource owner from the identity provider;
3) requesting authorization for required resources at the authorization provider;
4) access to resources at the resource provider.

In the following example the accessing party has implemented an application running on a web server (client application) and interacting with the resource owner via a user agent (browser).

NOTE Different colours in Figure A.2 are used to facilitate understanding of the diagram, they do not have any technical meaning.

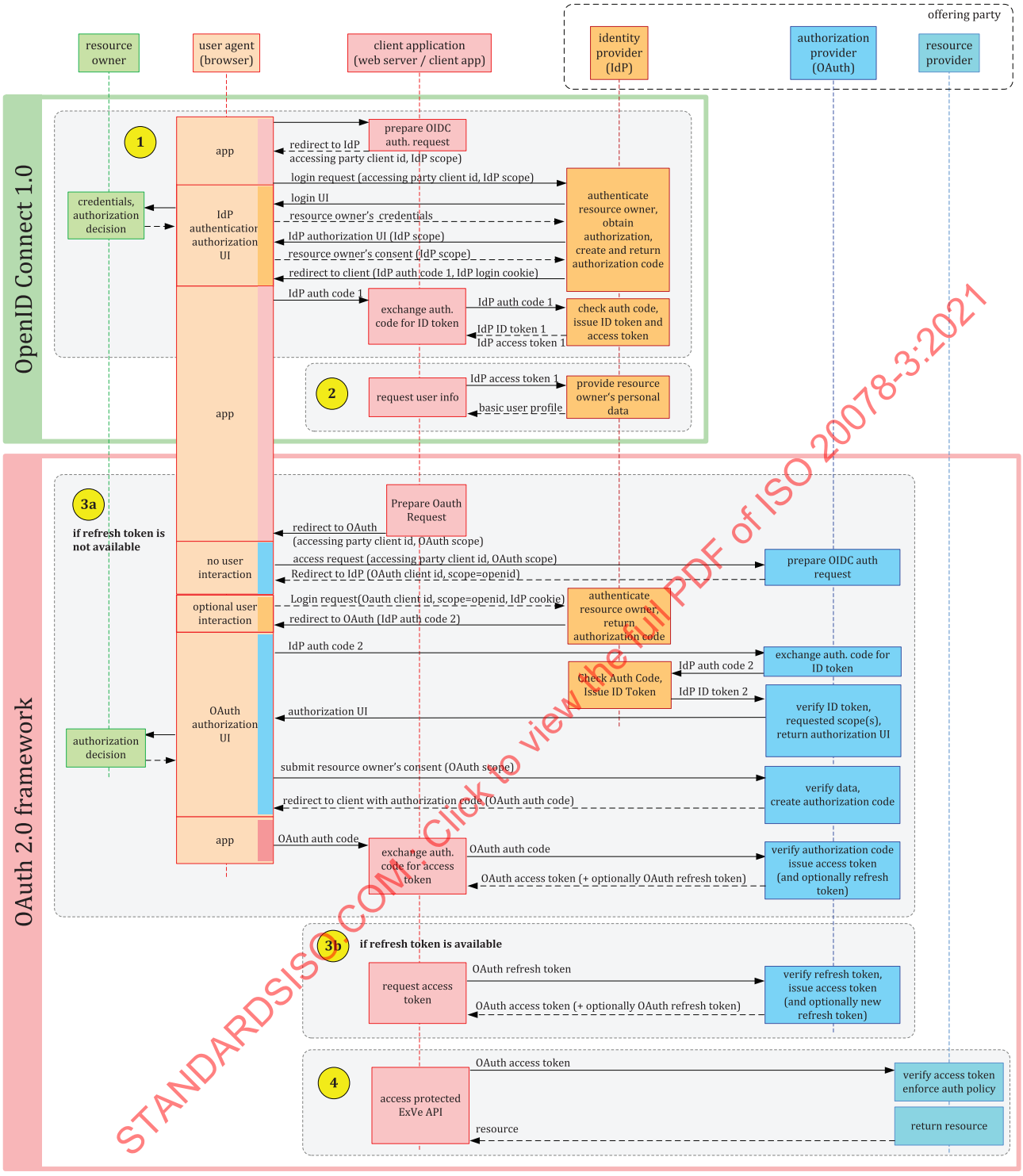


Figure A.2 — Obtaining authorization for protected resources owned by a resource owner

Detailed description of the diagram as shown in Figure A.2.

1) Resource owner identity should be verified by the associated identity provider as defined in OpenID Connect 1.0[2] section *Authentication using the authorization code flow*. For this, the accessing party's client application redirects the browser to the identity provider and initiates the authentication process where the resource owner authenticates directly with the identity provider. The identity provider checks the credentials and shows the resource owner the scope of personal data the client application server wants to access. Optionally, the client application can also request access to the

resource owner profile of the resource owner ("UserInfo" end point). As a result of the successful resource owner authentication (credentials correct and user grants permission for the requested identity scope), the identity provider returns an authorization code and redirects the browser back to the client application server. This quite complex process ensures that the resource owner does not have to provide their credentials to the client application. The resource owner can also check during the redirection to be connected to their well-known identity provider (HTTPS) and is in control of the provided personal data (scope).

The browser hands over the authorization code to the client application server. With the authorization code the client application server requests a digitally signed ID token from the identity provider. If optionally requested and granted, the client application can get additionally the access token for the "UserInfo" end point, issued by the identity server for the scope granted by the resource owner. The issued tokens are typically only valid for the requesting client, i.e. in this example the client application.

2) The client application can use the access token for the "UserInfo" end point to obtain (as an example) the basic profile of the resource owner. The basic profile contains, for example, a subset of the resource owner's profile data. This step is optional; the ID token itself can hold enough personal information for some use cases. The needed identity scope (subset) can vary depending on the use case and is granted by the resource owner (cf. step 1).

3) A request for authorization is based on the authorization code grant as defined in RFC 6749[1]. Depending on the availability and validity of the refresh token, the client application can request the authorization following either step 3 a) (first-time access) or step 3 b) (subsequent authorization requests if refresh token is available and valid).

   a) If the refresh token is not available on the client side, the resource owner approval is required. The client application requests authorization at the authorization provider, providing the intended authorization scope and the client application id. The authorization provider will redirect the browser to the identity provider to be able to authenticate the resource owner, in the same way as in step 1. As the resource owner has already been authenticated by the same identity provider in the previous step, the resource owner will in most cases just confirm its identity. The authorization provider validates the requested authorization scope, the resource owner ID, the resource owner's relationship with the connected vehicle and other subjects according to the defined authorization policy. If successfully validated the authorization provider requests resource owner's approval providing the authorization UI. This process uses the similar technical browser redirections as step 1. The step might look complicated, but enables that the resource owner can check to be connected to their well-known authorization provider (HTTPS) and is in control of the granted authorization scope. With resource owner's consent, the authorization provider issues the digitally signed access token for the requested resources and returns the access token to the client application. Optionally, the authorization provider may issue a refresh token, limited to the scope granted by the resource owner.

   b) For the subsequent access to resources, the client application should use the refresh token, issued by the authorization server to retrieve new access tokens, as long as the new authorization request is within the scope of the refresh token. Steps 1, 2, and 3 a) can be omitted.

4) The client application accesses resources by providing the access token. The resource server validates the access token claims and access token signature, checks whether the access token matches the defined access control rules and, if successful, processes the request.

A.3.2 Access to protected resources with resource owner's approval at runtime (simplified)

If the accessing party does not need to access basic profile information, the flow can be somewhat simplified and the order of the steps changes compared to A.3.1.

This initial access needs two steps.

1) Requesting authorization for required resources at the authorization provider including resource owner authentication via the identity provider.

2) Access to resources at the resource provider.

In the following example (Figure A.3) the accessing party has implemented an application running on a webserver (client application) and interacting with the resource owner via a user agent (browser).

NOTE Different colours in Figure A.3 are used to facilitate understanding of the diagram, they do not have any technical meaning.

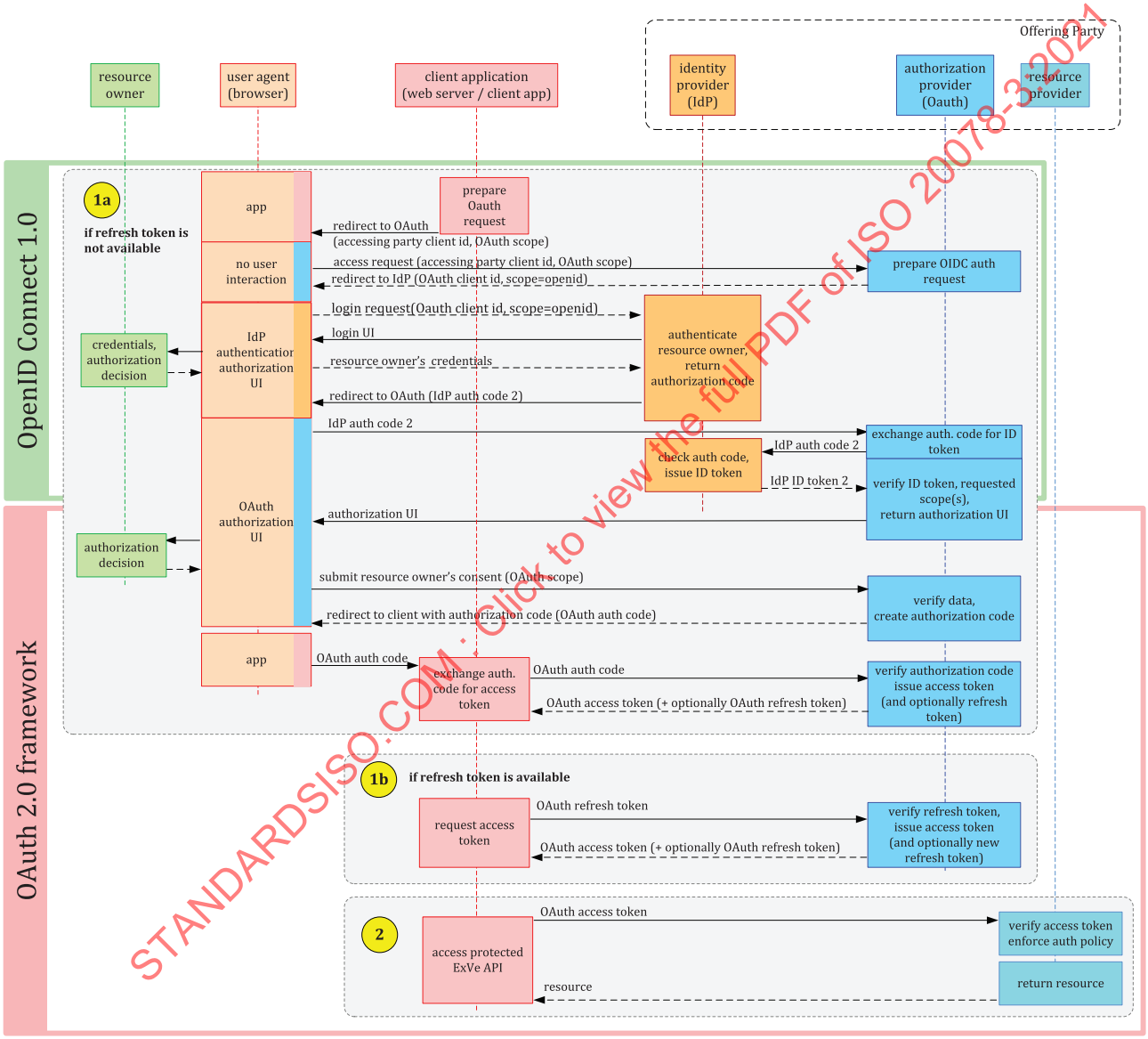


Figure A.3 — Simplified flow for obtaining authorization for protected resources owned by a resource owner

As this is just a variant of the flow described in A.2, it will not be described in the same level of detail.

1) Request authorization.

   a) If the client application lacks a refresh token, the client application redirects the browser to the authorization server. The authorization server redirects the browser to the identity provider. When the user is authenticated, the authorization server asks the user to grant

access. Is successful, the authorization server will issue refresh and access tokens to the client application.

b) If the client application has a refresh token, it uses this token to request an access token from the authorization provider.

2) The client application uses the access token to access resources.

A.3.3 Access to protected resources without resource owner runtime interaction

The client application requests authorization for access to resources and runtime interaction with the resource owner (e.g. customer) is not needed. The resources are either not personalized or the accessing party has received in advance the resource owner's approval, using other processes which are accepted by the resource owner and supported by the offering party. Definition of these approval processes is beyond the scope of this document and should be agreed between the accessing party and the offering party.

Three roles are involved in the communication flow:

— the accessing party using the client application;

— the authorization provider with the authorization server; and

— the resource provider with the resource server.

Obtaining authorization is based on the client credentials grant as specified in RFC 6749[1]. The mutual TLS authentication may be used for client authentication. Other suitable client authentication methods may be agreed between accessing party and offering party. Access to the basic profile of the resource owner is not covered by this use case.

NOTE 1 Different colours in Figure A.4 are used to facilitate understanding of the diagram, they do not have any technical meaning.

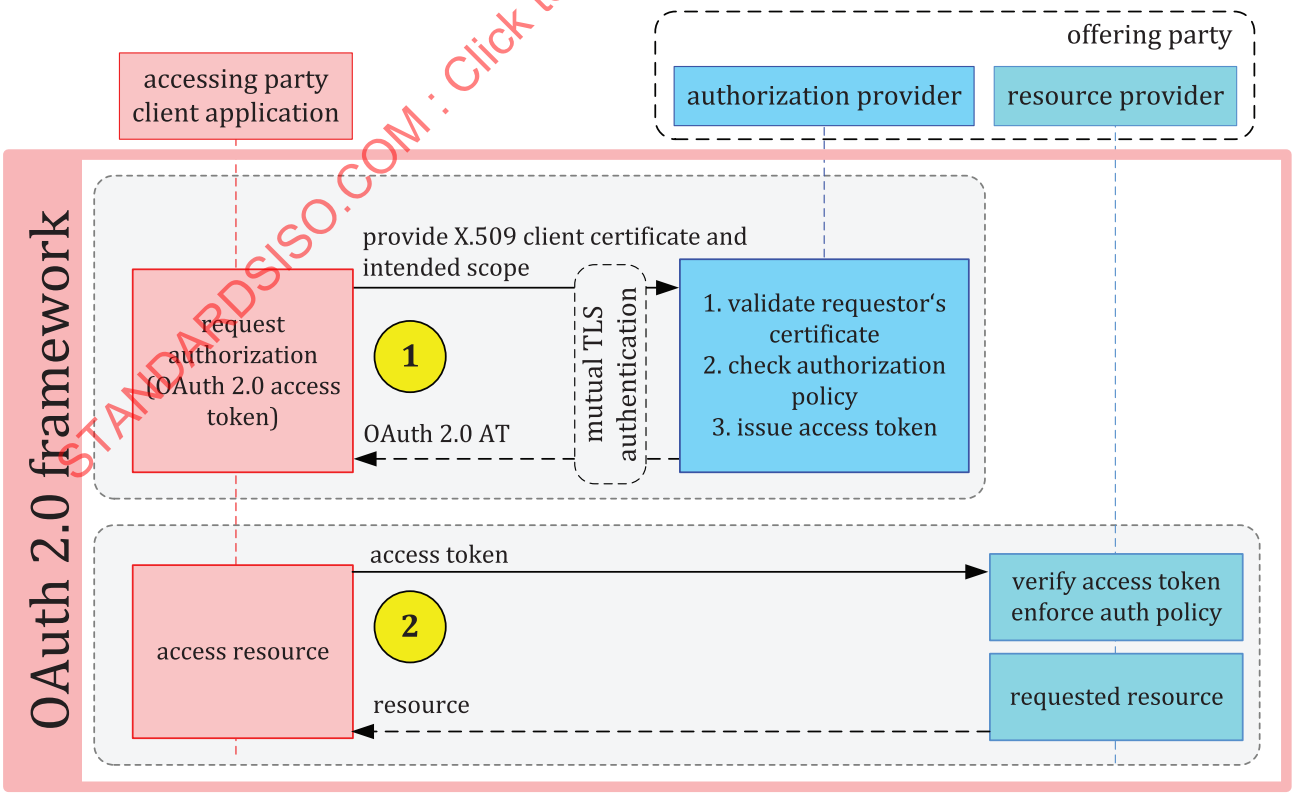


Figure A.4 — Requesting an authorization and access to resource based on client credentials grant

This flow (Figure A.4) consists of two steps.

1) The client application and the authorization server are establishing a secure HTTP connection (HTTPS) by using mutual TLS authentication. The client application requests an access token for the required resources, providing the intended scope. The authorization server validates the client identity, the requested scope and the availability of corresponding resource owner's consent. After a successful validation, the authorization server issues and returns the digitally signed access token to the client application.

2) The client application accesses the resources by providing the obtained access token. The resource server validates the access token claims and the access token signature, checks whether the access token matches to the defined access control rules and, if successful, processes the request.

NOTE 2 The second step — access to resources — is not different in comparison with the last step described in use case A.3.1. access to resources provided by the resource server works exactly the same way, independent of the authorization process.

Internet exposed web interfaces typically have a large attack surface. To mitigate potential cross site attacks, the accessing party should use a TLS certificate for the mutual authentication with ExVe interfaces which is not used for any other purpose.

NOTE 3 Figure A.4 shows the generic flow of Oauth 2.0 using a public key infrastructure.

A.3.4 Revocation of granted permission

The resource owner should be able to revoke authorization that was previously given to the accessing party. Alternatively, the client application of the accessing party may notify the authorization server, that the access to resources is no longer needed (e.g. due to logout process). Depending on the implemented authorization process, this implies.

— If issued, the refresh token should be revoked. The authorization server should provide the revocation end point in accordance with RFC 7009[7].

— If an immediate revocation (of access tokens) is required: the associated access tokens should be revoked. In case of self-contained access tokens, an additional backend interaction between the resource server and the authorization server is required every time the client application presents an access token. Otherwise, the client application could still use the revoked access token as long as the access token is not expired since the resource server will not be notified about the revocation status. To enable checking the revocation status of the access token, the authorization server can provide a token Introspection End point to the resource server as defined in RFC 7662[4].

— If client credentials grant was used as authorization flow, the authorization server should deny renewal of access tokens of the client application or implement other suitable methods.