FIDO2 authentication for eIDAS remote signing

White Paper

Copyright

Trademarks

Disclaimer

Contact Information

Yubico AB
Olof Palmes gata 11
SE-111 37 Stockholm
Sweden
yubi.co/contact

Document Release Date

Revision 1.1 - October 18, 2019.

Contents

Abstract **7**

Introduction **7**

eIDAS legislation 7

eIDAS QTSPs 7

eIDAS QTSP remote signing service 7

eIDAS assurance levels for electronic identification 8

Commission Implementing Decision (EU) 2016/650 9

CEN/TC 224 standards 10

CEN EN 419 241-1 (General System Security Requirements) 10

CEN EN 419 241-2 (Protection Profile for QSCD for Server Signing) 10

CEN EN 419 221-5 (Protection Profiles for TSP Cryptographic Modules) 10

FIDO2 standard 10

Terminology **11**

References 11

CEN EN 419 241-1 (General System Security Requirements) 11

CEN EN 419 241-2 (Protection Profile for QSCD for Server Signing) 11

CEN EN 419 221-5 (Protection Profiles for TSP Cryptographic Modules) 11

eIDAS Regulation EU 910/2014 11

eIDAS Implementing Regulation EU 2015/1502 on Assurance Levels 11

eIDAS Implementing Decision EU 2016/650 on QSCD 11

Fast Identity Online v2 (FIDO2) 12

FIDO Alliance CTAP2 12

W3C WebAuthn 12

Definitions and abbreviations 12

Assertion Signature (AsSig) 12

Attestation Object 12

Attestation Signature (AttSig) 13

Authentication Assertion 13

Authenticator 13

AuthenticatorAttestationResponse 13

Authenticator Data 13

AuthenticatorGetAssertion operation 14

AuthenticatorMakeCredential operation 14

Authenticator Response 14

Certification Authority (CA) 14

Client 14
Client Data 14
Credential Id (CrId) 15
Credential Key-pair (CrK) 15
Credential Private Key (CrPrK) 15
Credential Public Key (CrPuK) 15
Document To Be Signed (Doc-TBS) 15
Hardware Security Module (HSM) 15
PublicKeyCredentialCreationOptions.challenge (Ch_Reg) 16
PublicKeyCredentialRequestOptions.challenge (Ch_Auth) 16
Qualified Certificate (QC) 16
Qualified Certificate Key-pair (QcK) 16
Qualified Certificate Public Key (QcPuK) 17
Qualified Certificate Private Key (QcPrK) 17
Qualified Certificate Key Identifier (QcKId) 17
Qualified Certificate Credential Signature (QcCrSig) 17
Qualified Electronic Signature (QES) 17
Qualified Signature Creation Device (QSCD) 18
Qualified Trust Service Provider (QTSP) 18
Relying Party (RP) 18
Relying Party Identity (RP ID) 18
Server Signing Application (SSA) 19
Signature Activation Data (SAD) 19
Signature Activation Module (SAM) 19
Signature Activation Protocol (SAP) 19
Signed Document (SigDoc) 20
Signer 20
Signer's Interaction Component (SIC) 20
Signer Interface 20
Sole Control Assurance Level (SCAL2) 21
User 21
User Handle 21
UserId 21

Scope and design goals 21

System overview 22

System overview 22
Entities in the system overview 23

Registration and issuance process 24

Registration and issuance process overview 24
Registration and issuance process steps 24
Registration and issuance step 1 25
Registration and issuance step 2 25
Registration and issuance step 3 25
Registration and issuance step 4 25
Registration and issuance step 5 25
Registration and issuance step 6 26
Registration and issuance step 7 26
Registration and issuance step 8 26
Registration and issuance step 9 26
Registration and issuance step 10 26
Registration and issuance step 11 26
Registration and issuance step 12 26
Registration and issuance step 13 26
Registration and issuance step 14 27
Registration and issuance step 15 27
Registration and issuance step 16 27
Registration and issuance step 17 28

Authentication and signing process 28

Authentication and signing process overview 28
Authentication and signing process steps 29
Authentication and signing step 1 29
Authentication and signing step 2 29
Authentication and signing step 3 29
Authentication and signing step 4 30
Authentication and signing step 5 30
Authentication and signing step 6 30
Authentication and signing step 7 30
Authentication and signing step 8 30
Authentication and signing step 9 30
Authentication and signing step 10 30
Authentication and signing step 11 30
Authentication and signing step 12 30
Authentication and signing step 13 31
Authentication and signing step 14 31
Authentication and signing step 15 31
Authentication and signing step 16 32

Authentication and signing step 17 32

Abbreviations **32**

List of abbreviations 32

Mapping of CEN and FIDO2 terms 33

Abstract

This white paper describes an architecture for how to use FIDO2 as authentication protocol to attain high assurance level to an eIDAS compliant Qualified Trust Service Provider. More specifically, a user can use FIDO2 for strong end-to-end authentication to its Qualified Certificate's private key residing in a centralized Qualified Signature Creation Device, which is operated by a Qualified Trust Service Provider. When the end-user is authenticated to its remote private key, it can be used for creating remote Qualified Electronic Signatures. Based on this design, there is a cryptographic binding of the FIDO2 authentication process with the remote eIDAS Qualified Electronic Signature creation. The extensions to FIDO2 proposed in this white paper are backwards compatible with the FIDO2 standard. The design is compatible with the Signature Activation Protocol, Signature Activation Module and security requirements specified in the CEN EN 419 241 standard suite, in conjunction with the Qualified Signature Creation Device specified in CEN EN 419 221-5. Hence, the user gets sole control of the remote signature creation process.

Introduction

eIDAS legislation

eIDAS QTSPs

The eIDAS regulation (EU 910/2014) introduced the concept of Qualified Trust Service Providers (QTSPs), which are supervised service providers that are accredited to perform trusted services. The trust services are: certification authority for issuing signature/seal certificates, signature validation, time-stamping, registered delivery, preservation, and signature/seal creation.

The QTSPs that are in scope of this document are Certification Authority (CA) for issuing qualified electronic certificates and creation of qualified electronic signatures.

Hence, the QTSPs must be operated according to the accreditation made by the Conformity Assessment Body (according to ETSI TS 319 403) and the approval by the national supervisory body. Therefore, the QTSP CAs must adhere to the relevant policy, security and operational requirements in the ETSI standards EN 319 401, EN 319 411-1, EN 319 411-2, and TR 119 411-4. Similarly, the QTSPs for remote signing services must adhere to the relevant policy, security and operational requirements in the ETSI standards EN 319 401, TS 119 431-1, TS 119 431-2, and TS 119 432.

eIDAS QTSP remote signing service

When the QTSP creates Qualified Electronic Signatures centrally for remote end-users, the process is denoted as remote signing. In this setup, the QTSP operates a Qualified Signature Creation Device (QSCD), which is typically a Hardware Security Module, where the users' Qualified Certificates and associated private keys are securely stored and managed.

One important aspect is the concept of sole control, which is defined in recital 52 of the eIDAS regulation:

"In order to ensure that such electronic signatures receive the same legal recognition as electronic signatures created in an entirely user-managed environment, remote electronic signature service providers should apply specific management and administrative security procedures and use trustworthy systems and products, including secure electronic communication channels, in order to guarantee that the electronic signature creation environment is reliable and is used under the sole control of the signatory."

The user's authentication to the QTSP is therefore of major importance when creating a remote Qualified Electronic Signature.

eIDAS assurance levels for electronic identification

The eIDAS regulation also introduced three levels of assurance for electronic identification: low, substantial and high; those assurance levels are defined in article 8 in the eIDAS regulation.

- The low assurance level requires the electronic identification scheme to use at least one authentication factor, for example username and password.

- The substantial assurance level requires the electronic identification scheme to use at least two authentication factors from different categories (possession, knowledge or inherent). The user should be in control of or in possession of the authentication factors and the authentication process shall include dynamic authentication. An example of substantial assurance level is one-time passwords that are distributed by SMS to mobile phones.

- The high assurance level this requires the substantial level and means to protect the electronic identification scheme against duplication and tampering. An example of substantial assurance level is a PKI based authentication scheme with a hardware authentication token. FIDO2 with a hardware authenticator falls in the category of a high assurance electronic identification scheme.

The assurance levels substantial and high are required for authentication to a remote signing QTSP according to article 24(1) in the eIDAS regulation:

"1. When issuing a qualified certificate for a trust service, a qualified trust service provider shall verify, by appropriate means and in accordance with national law, the identity and, if applicable, any specific attributes of the natural or legal person to whom the qualified certificate is issued.

The information referred to in the first subparagraph shall be verified by the qualified trust service provider either directly or by relying on a third party in accordance with national law:

(a) by the physical presence of the natural person or of an authorised representative of the legal person; or

(b) remotely, using electronic identification means, for which prior to the issuance of the qualified certificate, a physical presence of the natural person or of an authorised representative of the legal person was ensured and which meets the requirements set out in Article 8 with regard to the assurance levels 'substantial' or 'high'; or [...]"

In order to implement a QTSP remote signing service that meets the sole control requirement set out in recital 52, however, an electronic identification scheme with assurance level high is needed.

Commission Implementing Decision (EU) 2016/650

The Commission Implementing Decision (EU) 2016/650 lays down standards for the security assessment of qualified signature and seal creation devices (QSCDs) pursuant to Articles 30(3) and 39(2) of the eIDAS regulation. The Implementing Decision (EU) 2016/65 is however primarily focused on the security requirements and assessment of a QSCD that is managed locally by an individual.

When the Commission Implementing Decision (EU) 2016/650 was written, however, there were no available standards for signing devices operated by a trust service provider in a secure environment that could meet the requirements in Regulation (EU) 910/2014 Annex II for qualified signature or seal creation devices. Currently, the assurance level for authentication to a QTSP with a QSCD for remote signing is set to substantial or high, according to article 24(1) in the eIDAS regulation. Therefore, there exists a gap on how to achieve sole control of the remote qualified signing process.

In order to close this gap, the CEN/TC 224 technical committee has published three CEN standards that address how QTSPs should operate QSCDs and manage signature creation data on behalf of a remote user to create qualified electronic signatures or seals:

- CEN EN 419 241-1 (Trustworthy Systems Supporting Server Signing Part 1: General Security Requirements)
- CEN EN 419 241-2 (Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing)
- CEN EN 419 221-5 (Protection Profiles for TSP Cryptographic Modules - Part 5 - Cryptographic Module for Trust Services)

If the CEN standards are implemented in conjunction with each other, the user will get sole control of the remote creation of qualified electronic signatures.

Furthermore, ENISA has published a report that describes how the EU Commission Implementing Decision 2016/650 may be updated to reference the CEN standards above to regulate the security and operations of a QSCD at a QTSP with the purpose of creating remote qualified electronic signatures.

CEN/TC 224 standards

CEN EN 419 241-1 (General System Security Requirements)

The CEN 419 241-1 standard specifies security requirements and recommendations for Trustworthy Systems Supporting Server Signing that generate digital signatures. The Trustworthy Systems Supporting Server Signing is composed of a Server Signing Application (SSA) and a remote Signature Creation Device. A remote Signature Creation Device is extended with remote control provided by a Signature Activation Module (SAM) executed in a tamper protected environment. This SAM module uses the Signature Activation Data (SAD), collected through a Signature Activation Protocol (SAP), in order to guarantee that the signing keys are used under sole control of the signer (user). The SSA uses a remote Signature Creation Device in order to generate, maintain and use the signing keys under the sole control of their authorized signer. The standard is protocol agnostic, so the Signature Activation Protocol (SAP) is not specified in this standard, although FIDO2 is explicitly listed as a viable alternative.

CEN EN 419 241-2 (Protection Profile for QSCD for Server Signing)

The CEN 419 241-2 standard specifies a Common Criteria protection profile for a Signature Activation Module (SAM), which is aimed to meet the requirements of a QSCD being operated remotely. To ensure the signer (user) has sole control of his signing keys, the signature operation needs to be authorized. This is carried out by a Signature Activation Module (SAM), which can handle one endpoint of SAP, verify SAD and activate the signing key within a QSCD Cryptographic Module. Both the QSCD Cryptographic Module and the SAM are to be located within a tamper protected environment such as an HSM. SAD verification means that the SAM checks the binding between the three SAD elements as well as checking that the signer is authenticated. The assurance requirement of the CEN EN 419 241-2 SAM Protection Profile is Common Criteria EAL4 augmented.

CEN EN 419 221-5 (Protection Profiles for TSP Cryptographic Modules)

The CEN EN 419 221-5 standard specifies a Common Criteria Protection Profile for cryptographic modules, which is intended to be suitable for use by trust service providers supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time-stamp operations, and authentication services, as identified by the eIDAS regulation. The protection profile also describes how the cryptographic module can be deployed as a remote QSCD at a QTSP.

FIDO2 standard

The FIDO2 standard has been developed by the World Wide Web Consortium (W3C) together with the FIDO Alliance in order to create a strong authentication solution for the web. The FIDO2 specifications are comprised of the W3C Web Authentication (WebAuthn) specification and FIDO Alliance's corresponding Client-to-Authenticator Protocol v2 (CTAP2). The WebAuthn specification describes the registration and authentication ceremonies between the client and the Relying Party, while the CTAP2 protocol specifies the communication between the client and the FIDO2 authenticator.

The FIDO2 standard is a dynamic authentication protocol with support for a tamper-proof hardware authenticator device. Furthermore, FIDO2 is phishing proof, due to the design with origin bound keys. Hence, FIDO2 meets the eIDAS requirements on authentication to a QTSP remote signing service.

Terminology

References

The reference documents that are relevant for this white paper are described in the subsections below.

CEN EN 419 241-1 (General System Security Requirements)

CEN/TC 224 standard EN 419 241-1 on Trustworthy Systems Supporting Server Signing – Part 1: General System Security Requirements[1].

CEN EN 419 241-2 (Protection Profile for QSCD for Server Signing)

CEN/TC 224 standard EN 419 241-2 on Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing.

CEN EN 419 221-5 (Protection Profiles for TSP Cryptographic Modules)

CEN/TC 224 standard EN 419 221-5 on Protection profiles for TSP Cryptographic modules - Part 5: Cryptographic Module for Trust Services.

eIDAS Regulation EU 910/2014

Regulation EU 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS Regulation) adopted on 23 July 2014 provides a regulatory environment to enable secure and seamless electronic interactions between businesses, citizens and public authorities.

eIDAS Implementing Regulation EU 2015/1502 on Assurance Levels

Commission Implementing Regulation EU 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014.

eIDAS Implementing Decision EU 2016/650 on QSCD

Commission Implementing Decision EU 2016/650 of 25 April 2016 laying down standards for the security assessment of qualified signature and seal creation devices pursuant to Articles 30(3) and 39(2) of Regulation (EU) No 910/2014.

[1] The CEN/TC 224 standard EN 419 241-1 is not publicly available, and must be purchased.

Fast Identity Online v2 (FIDO2)

Fast Identity Online v2 (FIDO2) is the overarching term for FIDO Alliance's set of specifications. FIDO2 enables users to leverage common devices to easily authenticate to online services in both mobile and desktop environments. The FIDO2 specifications are comprised of the W3C Web Authentication (WebAuthn) specification and FIDO Alliance's corresponding Client-to-Authenticator Protocol v2 (CTAP2).

FIDO Alliance CTAP2

The Client To Authenticator Protocol v2 (CTAP2) is a FIDO2 specification created by the FIDO Alliance. The CTAP2 specification describes an application layer protocol for communication between a roaming authenticator and another client/platform, as well as bindings of this application protocol to a variety of transport protocols using different physical media.

W3C WebAuthn

The W3C WebAuthn standard defines an API enabling the creation and use of strong, attested, scoped, public key-based credentials by web applications, for the purpose of strongly authenticating users.

Definitions and abbreviations

All definitions and abbreviations that are relevant for this white paper are described in the subsections below. In several cases the terms are defined in the W3C WebAuthn or eIDAS specifications; when this is the case there is a short description and a reference pointing to the specification.

Assertion Signature (AsSig)

Assertion Signature is a FIDO2 term that is defined in the WebAuthn specification.

An Assertion Signature is produced when the authenticatorGetAssertion method is invoked. It represents an assertion by the Authenticator that the user has consented to a specific transaction, such as logging in, or completing a purchase. Thus, an Assertion Signature asserts that the user requesting the transaction is in possession of the authenticator controlling a particular credential private key. An Assertion Signature also indicates whether the authenticator has authenticated the user via a second authentication factor, such as a PIN or biometric.

Attestation Object

Attestation Object is a FIDO2 term that is defined in the WebAuthn specification.

An Attestation Object is a data structure containing a newly created credential public key, an Attestation Signature by an attestation private key over that credential public key and a challenge, as well as a certificate or similar data providing provenance information for the attestation public key.

Attestation Signature (AttSig)

Attestation Signature is a FIDO2 term that is defined in the WebAuthn specification.

An Attestation Signature is produced by an Authenticator when a new Credential Key-pair is created via an AuthenticatorMakeCredential operation

Authentication Assertion

Authentication Assertion is a FIDO2 term that is defined in the CTAP2 specification.

The cryptographically signed Authentication Assertion object is returned by an Authenticator as the result of an AuthenticatorGetAssertion operation. The Authentication Assertion object contains the Credential ID of the Credential Key-pair used, the Assertion Signature and the Authenticator Data signed over, and the User Handle of the Credential Key-pair if available, as defined in the CTAP2 specification.

Authenticator

Authenticator is a FIDO2 term that is defined in the WebAuthn specification.

An Authenticator is a cryptographic entity used by a WebAuthn Client to (i) generate a public key credential and register it with a Relying Party, and (ii) authenticate by potentially verifying the user, and then cryptographically signing and returning, in the form of an Authentication Assertion, a challenge and other data presented by a WebAuthn Relying Party.

AuthenticatorAttestationResponse

AuthenticatorAttestationResponse is a FIDO2 term that is defined in the WebAuthn specification.

The AuthenticatorAttestationResponse contains the Client Data and the Attestation Object.

Authenticator Data

Authenticator Data is a FIDO2 term that is defined in the WebAuthn specification.

The Authenticator Data structure is signed over by Attestation Signatures and Assertion Signatures, and encodes contextual bindings made by the Authenticator. It includes a hash of the RP ID, and whether the authenticator authenticated the user prior to generating the signature. For Attestation Signatures, the Authenticator Data also contains the newly created Credential Public Key and its Credential ID.

These bindings are controlled by the authenticator itself, and derive their trust from the WebAuthn Relying Party's assessment of the security properties of the Authenticator via the Authenticator's Attestation Signature.

AuthenticatorGetAssertion operation

AuthenticatorGetAssertion operation is a FIDO2 term that is defined in the WebAuthn specification.

The AuthenticatorGetAssertion operation is carried out during the FIDO2 authentication ceremony, and returns an Authentication Assertion object.

AuthenticatorMakeCredential operation

AuthenticatorMakeCredential operation is a FIDO2 term that is defined in the WebAuthn specification.

The AuthenticatorMakeCredential operation is carried out during the FIDO2 registration ceremony, and returns an Attestation Object.

Authenticator Response

Authenticator Response is a FIDO2 term that is defined in the WebAuthn specification.

The Authenticator Response contains the Client Data and the Authentication Assertion.

Certification Authority (CA)

The Certification Authority is operated by an eIDAS accredited Qualified Trust Service Provider (QTSP), and is used for issuing Qualified Certificates to the users. In this context, the users' Qualified Certificates are stored centrally at the QTSP and are thus available to the QTSP's Remote Signing Service.

Client

WebAuthn Client is a FIDO2 term that is defined in the WebAuthn specification.

A WebAuthn Client is an intermediary entity typically implemented in the user agent. Conceptually, it underlies the WebAuthn API and embodies the implementation of the internal WebAuthn methods. It is responsible for validating the RP ID of the Relying Party, marshalling the inputs for the underlying Authenticator operations, through the CTAP2 protocol, and for returning the results of the latter operations to the WebAuthn API's callers.

Client Data

Client Data is a FIDO2 term that is defined in the WebAuthn specification.

The Client Data is signed over by Attestation Signatures and Assertion Signatures, and represents the contextual bindings of both the WebAuthn Relying Party and the Client. It includes the challenge given by the Relying Party, and the origin of the script that initiated the ceremony.

Credential Id (CrId)

Credential Id is a FIDO2 term that is defined in the WebAuthn specification.

Credential Id is a probabilistically-unique byte sequence identifying a Credential Key-pair and its Authentication Assertions.

Credential Key-pair (CrK)

The Credential Key-pair is comprised of the Credential Public Key (CrPuK) and the Credential Private Key (CrPrK).

Credential Private Key (CrPrK)

Credential Private Key is a FIDO2 term that is defined in the WebAuthn specification.

The Credential Private Key is the private key portion of the credential key pair, and is exclusively controlled by the Authenticator that created it.

Credential Public Key (CrPuK)

Credential Public Key is a FIDO2 term that is defined in the WebAuthn specification.

Credential Public Key is the public key portion of a Relying Party-specific credential key pair, generated by an Authenticator and returned to a Relying Party in a registration ceremony.

Document To Be Signed (Doc-TBS)

The Document To Be Signed (Doc-TBS) is the document that the user's Client submits to the QTSP Remote Signing Service. When the QTSP has authenticated the user, the Doc-TBS will be signed with a Qualified Electronic Signature.

Hardware Security Module (HSM)

The Hardware Security Module (HSM) is deployed at the QTSP Remote Signing Service. The HSM generates, hosts, protects and operates the keypairs of the users' Qualified Certificates. The connection between the QTSP and the HSM must be authenticated and encrypted. In particular, the HSM invokes the user's Qualified Certificate Private Key to sign the Document To Be Signed.

In order to implement the end-to-end security protocol based on FIDO2, the HSM needs to support safe code to be executed within the HSM. Such safe code implementation will result in the Signature Activation Module (SAM) that will cater for the security critical operations of the FIDO2 registration (when the FIDO2 credentials are cryptographically associated with the user's Qualified Certificate Key-pair), and the FIDO2 authentication (when the FIDO2 credentials are used for authentication to the user's Qualified Certificate Private Key).

PublicKeyCredentialCreationOptions.challenge (Ch_Reg)

The PublicKeyCredentialCreationOptions.challenge (Ch_Reg) is a FIDO2 term that is defined in the WebAuthn specification.

This is a challenge that the Relying Party generates during the FIDO2 registration ceremony, and is signed by the Authenticator as part of the AuthenticatorMakeCredential operation.

Within the context of this specification, the Ch_Reg is computed as the hash over the concatenation of the hashed Qualified Certificate Public Key (QcPuK) and a nonce (i.e., a random number):

```
Ch_Reg = hash(hash(QcPuK) || nonce)
```

Hence, this registration challenge is compatible with the standardized FIDO2 registration ceremony, and it also creates a binding to the Qualified Certificate Key-pair in the HSM.

PublicKeyCredentialRequestOptions.challenge (Ch_Auth)

The PublicKeyCredentialRequestOptions.challenge (Ch_Auth) is a FIDO2 term that is defined in the WebAuthn specification.

This is a challenge that the Relying Party generates during the FIDO2 authentication ceremony, and is signed by the Authenticator as part of the AuthenticatorGetAssertion operation.

Within the context of this specification, the Ch_Auth is computed as the hash over the concatenation of the hash of the Document To Be Signed (Doc-TBS), the hash of the Qualified Certificate Public Key (QcPuK), and a nonce:

```
Ch_Auth = hash(hash(Doc-TBS) || hash(QcPuK) || nonce)
```

Hence, this authentication challenge is compatible with the standardized FIDO2 authentication ceremony, and it also creates a binding to the Document To Be Signed as well as the Qualified Certificate Key-pair in the HSM.

Qualified Certificate (QC)

Qualified Certificate (QC) is an eIDAS term that is defined in the eIDAS Regulation.

A Qualified Certificate is issued under a QC-policy to a user by an accredited QTSP Certification Authority. The Qualified Certificate can be hosted centrally at the QTSP thus allowing for remote signing operations, which is the scope of this specification.

Qualified Certificate Key-pair (QcK)

The Qualified Certificate Key-pair is comprised of the Qualified Certificate Public Key (QcPuK) and the Qualified Certificate Private Key (QcPrK).

Qualified Certificate Public Key (QcPuK)

The Qualified Certificate Public Key (QcPuK) is the public part of the Qualified Certificate Key-pair used when issuing the Qualified Certificate. The QcPuK is wrapped in the Qualified Certificate that is issued by the QTSP Certification Authority.

Qualified Certificate Private Key (QcPrK)

The Qualified Certificate Private Key (QcPrK) is the private part of the Qualified Certificate Key-pair used when issuing the Qualified Certificate. The QcPrK is protected in the QSCD, which is an HSM in this scenario, and can only be accessed when the user has been authenticated. In this specification, the authentication is based on FIDO2 with a safe code implementation in the HSM.

Qualified Certificate Key Identifier (QcKId)

The Qualified Certificate Key Identifier (QcKId) is a handle that is provided by the HSM. The QcKId uniquely identifies the Qualified Certificate Private Key in the HSM.

Qualified Certificate Credential Signature (QcCrSig)

The Qualified Certificate Credential Signature (QcCrSig) is a new definition for this specification.

The QcCrSig is defined as the concatenation of the hash over the Qualified Certificate Public Key (QcPuK) with the hash over FIDO2 Credential Public Key (CrPuK), which is signed by the Qualified Certificate Private Key (QcPrK):

```
QcCrSig  = sig(hash(QcPuK) || hash(CrPuK), QcPrK)
```

Hence, the QcCrSig creates a cryptographic binding of the Qualified Certificate Private Key (QcPrK) to the FIDO2 Credential Public Key (CrPuK).

The signature format of QcCrSig, the asymmetric encryption algorithm and the hash algorithm are out of scope of this document. For example, PKCS #1 (RFC 8017) with 2048-bit RSA keys and SHA-256 as hash function can be used.

Qualified Electronic Signature (QES)

Qualified Electronic Signature (QES) is an eIDAS term that is defined in the eIDAS Regulation in Article 3, §12:

"(12) 'qualified electronic signature' means an advanced electronic signature that is created by a qualified electronic signature creation device, and which is based on a qualified certificate for electronic signatures."

A QES can be created in different signature formats such as CAdES, XAdES and PAdES, which are specified by ETSI. The details for how to create those formats go beyond the scope of this document.

When the Qualified Certificate and the associated Qualified Certificate Private Key are hosted and operated centrally at a QTSP, the QES is created remotely for a user.

Qualified Signature Creation Device (QSCD)

Qualified Signature Creation Device (QSCD) is an eIDAS term that is defined in Annex II of the eIDAS Regulation.

The QSCD must be Common Criteria certified by an accredited certification body according to the requirements outlined in the eIDAS Implementing Decision (EU 2016/650) on QSCD. A list of approved QSCDs is available at this official EU website.

In the context of a QTSP operating Remote Signing Services, the QSCD is a centrally operated HSM that generates, hosts, protects and operates the keypairs of the users' Qualified Certificates.

Qualified Trust Service Provider (QTSP)

Qualified Trust Service Provider (QTSP) is an eIDAS term that is defined in the eIDAS Regulation in Article 3, §20:

"(20) 'qualified trust service provider' means a trust service provider who provides one or more qualified trust services and is granted the qualified status by the supervisory body;"

A QTSP is audited by a conformity assessment body according to ETSI guidelines, and is accredited by the supervisory body in the relevant EU member state.

A QTSP can be accredited for different types of services. The trust services that are in scope of this document are Certification Authority (for issuing of Qualified Certificates that are stored centrally at the CA) and Remote Signing Service.

Relying Party (RP)

The Relying Party is a FIDO2 term that is defined in the WebAuthn specification.

This is the entity whose web application utilizes the WebAuthn API to register and authenticate users.

Relying Party Identity (RP ID)

The Relying Party Identifier is a FIDO2 term that is defined in the WebAuthn specification.

The Relying Party Identifier is a valid domain string that identifies the WebAuthn Relying Party on whose behalf a given registration or authentication ceremony is being performed. A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with. WebAuthn Clients enforce that only web origins matching the RP ID are allowed to invoke the WebAuthn API with that RP ID.

Server Signing Application (SSA)

The Server Signing Application (SSA) is a CEN term as defined in CEN EN 419 241-1. The SSA is operated by the eIDAS compliant Qualified Trust Service Provider (QTSP). It has the capability of creating Qualified Electronic Signatures for remote users with the users' Qualified Certificates and Qualified Certificate Private Keys hosted in the Qualified Signature Creation Device (QSCD).

Signature Activation Data (SAD)

The Signature Activation Data (SAD) is a CEN term defined in CEN EN 419 241-1 as follows:

"To reach the SCAL2, the use of the SAD to ensure control over the signer's key SHALL be enforced by the SAM. Signature activation at SCAL2 requires fulfilment of several conditions as signer authentication and authenticity of signature operation request from the signer. Both properties MAY be given directly by the SAD. It is however for instance also possible to perform signer authentication prior to SAD generation, e.g. using delegation of authentication. SAD can be a set of data or be a result of cryptographic operations from which the same information can be derived. SAD contributes to authenticate directly or indirectly the signer. When the signer authentication takes place prior to collection of the SAD, the SAD SHALL contain items to identify the signer asserted by a known source. This assertion MAY come from either the SIC or from a trusted electronic identity provider. The source of the assertion SHALL be authenticated."

In the context of this document, the SAD is a combination of an Authentication Assertion, created by the user's Authenticator, and a QcCrSig, which authorizes that Authenticator to authorize signing with a Qualified Certificate Private Key.

Signature Activation Module (SAM)

The Signature Activation Module (SAM) is a CEN term defined in CEN EN 419 241-1 as follows:

"The SAM is a software that uses the SAD in order to guarantee with a high level of confidence that the signing keys are used under sole control of the signer for SCAL2. The SAM is required to be used in a tamper protected environment. If the SAM is not used in the same tamper protected environment as the SCDev, then a secure channel between both tamper protected environments is required."

Signature Activation Protocol (SAP)

The Signature Activation Protocol (SAP) is a CEN term defined in CEN EN 419 241-1 as follows:

"The SAP SHALL be designed to allow secure use of the signing key for the creation of a digital signature to be performed by the cryptographic module on behalf of a signer. The SAP is a protocol where the signer (via the SIC) and the TW4S communicate in order to generate the SAD. The design of the SAP SHALL include as a minimum the following verifications:

- *signer authentication when using the signing key,*
- *authenticity of the signature request with specific SAD,*
- *the selected signing key is valid and active,*
- *secure transfer of all the elements of the SAD.*

When the signing key is not used to sign a proof of possession in order to obtain a certificate, the SAP SHOULD include the following verification:

- *presence of valid certificate associated to signing key."*

The FIDO2 based architecture described in this document is a SAP.

Signed Document (SigDoc)

The Signed Document (SigDoc) is the Document To Be Signed as input data, then signed by a Qualified Electronic Signature.

Signer

The signer is defined in CEN EN 419 241-1 as the user who is signing a document using a remote signing service.

Signer's Interaction Component (SIC)

The Signer's Interaction Component (SIC) is a CEN term defined in CEN EN 419 241-1 as follows:

"The SIC is a piece of software and/or hardware, operated on the signer's environment under its sole control. Using this component is essential in the SAP process and for the creation of a digital signature by the Signature Creation Device. The SIC always participates in the SAP in order to authenticate the signer or to generate the SAD:

- *the SIC can directly generate the SAD, or*
- *the SIC can be used to authenticate the signer, and the assertion that identify the signer will be used in the SAD generation.*

This component can be for example (or a combination of):

- *an application executed by a browser (e.g. a form HTTP POST over TLS),*
- *an application executed by a mobile device (e.g. smartphone, tablet),*
- *a secure element of a cell-phone (e.g. sim-card that receives SMS),*
- *a cryptographic device owned by the signer (e.g. eID token, eToken, FIDO-Token),*
- *[...]*

The SIC enforces the link between the signer and the signature operation within the SAP."

In the context of this document, the SIC is a WebAuthn Authenticator.

Signer Interface

The Signer Interface is a CEN term defined in CEN EN 419 241-1, which represents the Signer's local computer or mobile device.

Sole Control Assurance Level (SCAL2)

Sole control assurance level (SCAL2) is defined in the CEN standard CEN EN 419 241-1 as follows:

"Sole control assurance level 2 (SCAL2):

- *The signing keys are used, with a high level of confidence, under the sole control of the signer.*
- *The authorized signer's use of its key for signing is enforced by the SAM by means of SAD provided by the signer using the SAP, in order to enable the use of the corresponding signing key."*

User

The User is a FIDO2 term representing the person who is using the Authenticator for FIDO2 authentication.

User Handle

The User Handle is a FIDO2 term that is defined in the WebAuthn specification.

The User Handle is specified by a Relying Party, and used to map one or more Credential Key-pairs to a specific user account with the Relying Party. Authenticators in turn map RP ID and User Handle pairs to Credential Private Keys.

UserId

The UserId in the context of this document means the account a user has pre-registered at the QTSP.

Scope and design goals

The scope of this white paper is to describe how FIDO2 can be used as a Signature Activation Protocol (SAP) to a Signature Activation Module (SAM), which resides in a tamper-proof HSM at a QTSP. In the tamper-proof HSM, the SAM interacts with a QSCD, which ultimately protects the user's Qualified Certificate key-pair.

In this setup the FIDO2 Authenticator credentials are bound to the end-user's Qualified Certificate Key-pair, which resides remotely in the QSCD at the QTSP. Hence, the user can use the FIDO2 Authenticator to authenticate end-to-end to the Qualified Certificate Private Key and unlock this for signing operations. Transposed into CEN EN 419 241-1 terminology, the FIDO2 Authenticator is equivalent to the Signer's Interaction Component (SIC), and the FIDO2 Assertion Response is one half of the Signature Activation Data (SAD); the QcCrSig makes up the other half of the SAD.

With such a setup, the user attains Sole Control Assurance Level 2 (SCAL2) of the remote Qualified Signature creation process.

The SAM in the context of this white paper is assumed to be implemented and executed as safe code within the HSM. In particular, the SAM implementation should allow for validating sensitive portions of the FIDO2 authentication process and verifying the binding of the FIDO2 credentials to the end-user's key-pair in the HSM, and thereby unlock the end-user's Qualified Certificate private key.

Hence, the design goals are the following:

- The standard FIDO2 protocol should be used in the communication over WebAuthn and CTAP2.
- The FIDO2 protocol registration parameters are extended with elements for registration to the Qualified Certificate key-pair in the QSCD residing at the QTSP.
- The FIDO2 protocol authentication parameters are extended to allow for a CEN EN 419 241-1 compliant Signature Activation Protocol (SAP).
- Authentication to the Qualified Certificate private key is based on Signature Activation Data (SAD), which in turn is derived from the FIDO2 Assertion Response originating from the FIDO2 Authenticator.
- The solution should minimize the amount of data that needs to be stored long-term inside the SAM and the QSCD.
- Only the SAM should have the power to create or edit a binding between a FIDO2 credential and a Qualified Certificate Private Key.
- It should be possible to replace FIDO2 credentials without replacing the Qualified Certificate or re-signing any documents, i.e., FIDO2 credentials should not make long-lived document signatures.

System overview

System overview

An overview of the system is illustrated in figure 1.

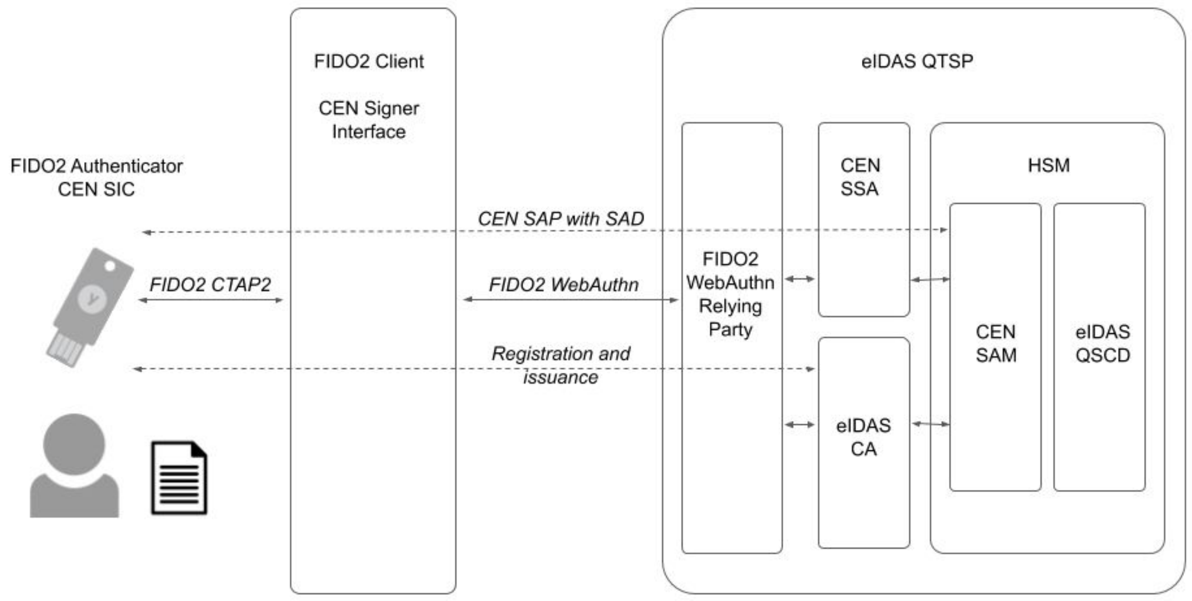


Figure 1 *- System overview*

Entities in the system overview

The entities in the system are the following:

- The Authenticator is the user's FIDO2 authenticator device; this term is equivalent to the CEN EN 419 241-1 compliant Signer's Interaction Component (SIC).

- The Client is the user's device, such as a laptop or mobile phone, with a web browser or app; this is equivalent to the CEN EN 419 241-1 term Signer Interface.

- The QTSP (Qualified Trust Service Provider) is comprised of the following components:

  - The QTSP operates a Certification Authority (CA) with the capabilities of issuing Qualified Certificates.

  - The QTSP is also operating a Server Signing Application (SSA) with the capabilities of creating remote Qualified Electronic Signatures for an authenticated user. The users' Qualified Certificates issued previously by the QTSP CA are available to the QTSP SSA.

  - The QTSP has also implemented a WebAuthn RP (Relying Party) that is used for FIDO2 registration and authentication.

  - The QSCD (Qualified Signature Creation Device) is operated by the QTSP. The QSCD is part of a tamper-proof HSM (Hardware Security Module) that hosts the users' Qualified Certificate key-pairs.

- The Signature Activation Module (SAM) is implemented as safe code in the HSM.
- Furthermore, the QTSP is assumed to operate a database, which can be used for storing the end-user data, Qualified Certificates and associated FIDO2 elements per user record.

Registration and issuance process

Registration and issuance process overview

The registration process for issuing a Qualified Certificate and creating FIDO2 credentials, and binding them together cryptographically, is illustrated in figure 2.

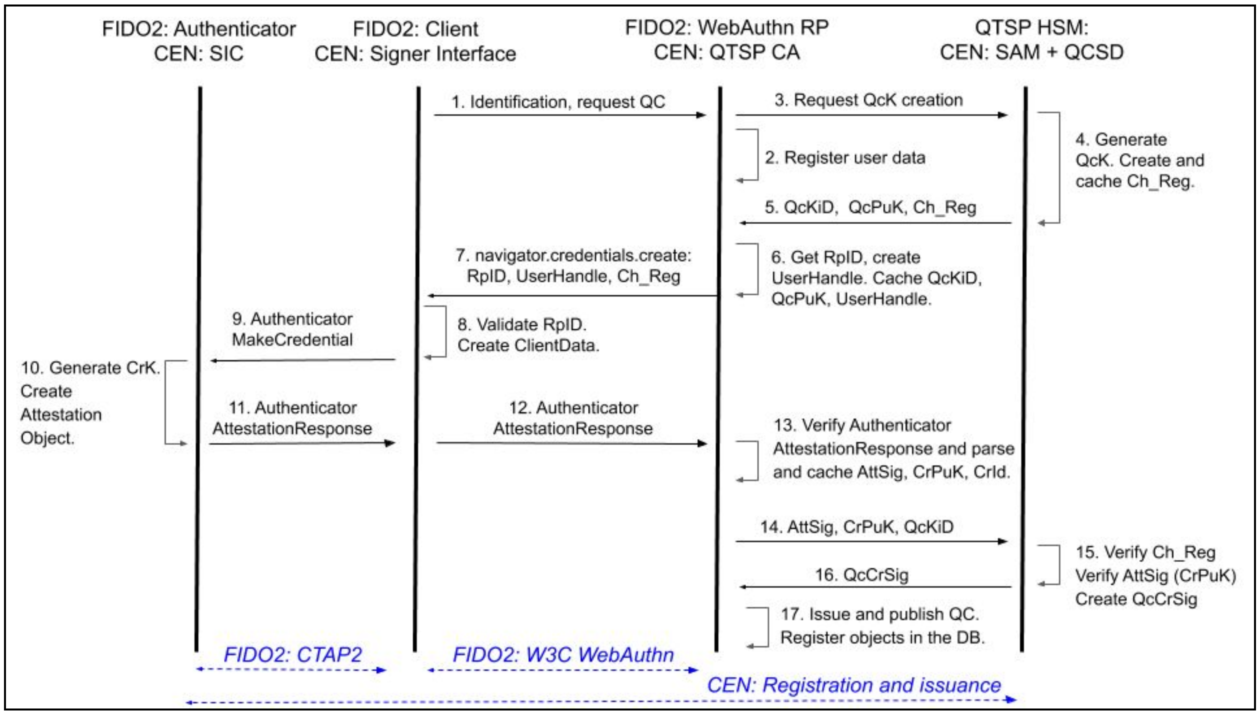

FIDO2: Authenticator
CEN: SIC
FIDO2: Client
CEN: Signer Interface
FIDO2: WebAuthn RP
CEN: QTSP CA
QTSP HSM:
CEN: SAM + QCSD
1. Identification, request QC
2. Register user data
3. Request QcK creation
4. Generate QcK. Create and cache Ch_Reg.
5. QcKiD, QcPuK, Ch_Reg
6. Get RpID, create UserHandle. Cache QcKiD, QcPuK, UserHandle.
7. navigator.credentials.create: RpID, UserHandle, Ch_Reg
8. Validate RpID. Create ClientData.
9. Authenticator MakeCredential
10. Generate CrK. Create Attestation Object.
11. Authenticator AttestationResponse
12. Authenticator AttestationResponse
13. Verify Authenticator AttestationResponse and parse and cache AttSig, CrPuK, CrId.
14. AttSig, CrPuK, QcKiD
15. Verify Ch_Reg Verify AttSig (CrPuK) Create QcCrSig
16. QcCrSig
17. Issue and publish QC. Register objects in the DB.
FIDO2: CTAP2
FIDO2: W3C WebAuthn
CEN: Registration and issuance

Figure 2 *- Registration process*

Both the FIDO2 terms and the equivalent CEN EN 419 241 terms are described in figure 2 to illustrate how they are related. In the process steps below, however, only the FIDO2 terms are used if there is an overlap between the FIDO2 and CEN terminology.

Registration and issuance process steps

The steps in the registration process are described in the subsections below. The registration process in this section describes the initial registration and association of FIDO2 credentials with a Qualified Certificate key-pair.

Registration and issuance step 1

The user gets identified by the QTSP's Registration Authority in order for the QTSP Certification Authority to issue a Qualified Certificate. The identification process goes beyond the scope of this document, but it must adhere to the requirements on a QTSP CA for issuing a Qualified Certificate.

The user should also have a FIDO2 Authenticator available throughout the registration process. The Authenticator may be the user's personal device or be handed out by the QTSP during or prior to the registration process.

Registration and issuance step 2

The QTSP registers the user's personal data, the certificate application forms, identity document copies, etc in the database. This is a sub-process for registering the certificate application documentation at the QTSP. The process for enrolling FIDO2 credentials and issuing Qualified Certificate continues at Registration and issuance step 3 and onwards.

Registration and issuance step 3

The QTSP CA invokes the SAM with a request to generate the Qualified Certificate Key-pair (QcK) in the QSCD.

Registration and issuance step 4

The SAM invokes the QSCD to generate an asymmetric key-pair for the end-user's Qualified Certificate. The output of this operation is the Public Key (QcPuK) and an identifier (QcKiD) to the key-pair in the QSCD.

Furthermore, the FIDO2 registration challenge Ch_Reg is calculated by the SAM. The definition of Ch_Reg is:

```
Ch_Reg = hash(hash(QcPuK) || nonce)
```

Since the Ch_Reg contains the hash value of the QcPuK, this creates an assertion with the Qualified Certificate key-pair that was generated in the QSCD.

Furthermore, the SAM caches the Ch_Reg and its nonce in secure short-term storage.

Note: Until the registration process reaches Registration step 15, the SAM has access to sign with the Private Key (QcPrK) in the QSCD. At Registration step 15 the SAM will create the signed object QcCrSig to bind the user's FIDO2 credentials to the user's Qualified Certificate key-pair in the QSCD; at this step the user's Private Key will be protected with the authentication algorithm in Authentication and signing step 15.

Registration and issuance step 5

The QSCD returns the QcPuK, QcKiD and Ch_Reg to the QTSP CA.

Registration and issuance step 6

The WebAuthn Relying Party (RP) at the QTSP generates the UserHandle and gets the Relying Party Identifier (RP ID). The QcKiD, QcPuK, UserHandle and Ch_Reg are stored temporarily at the Relying Party during the registration session.

Registration and issuance step 7

The Relying Party invokes the WebAuthn navigator.credentials.create method at the Client's browser or app. As part of this call, the input parameters RpID, UserHandle and Ch_Reg are passed on to the Client.

The Relying Party sets the parameter publicKey.attestation: "direct" in order to enforce that the Authenticator used is permitted by the Relying Party's security policy.

Registration and issuance step 8

The Client validates the RpId and creates Client Data.

Registration and issuance step 9

The Client invokes the Authenticator by calling the AuthenticatorMakeCredential method over the CTAP2 protocol.

Registration and issuance step 10

The Authenticator generates the Credential Key-pair (CrK) and creates the Attestation Object. The Attestation Object contains the Credential Public Key (CrPuK) and implicitly the Ch_Reg, since its hash value is part of the Client Data.

Registration and issuance step 11

The Authenticator returns the AuthenticatorAttestationResponse over the CTAP2 protocol to the Client.

Registration and issuance step 12

The Client passes the AuthenticatorAttestationResponse over to the Relying Party over the WebAuthn protocol.

Registration and issuance step 13

The Relying Party validates the AuthenticatorAttestationResponse as part of the standardized WebAuthn validation operation. As part of this validation operation, the Ch_Reg is validated by the Relying Party according to the WebAuthn standard. Furthermore, the Relying Party parses the Attestation Signature (AttSig), Credential Public Key (CrPuK), Client Data, and Credential ID (CrId) from the AuthenticatorAttestationResponse and caches the CrPuK and CrId.

Registration and issuance step 14

The QTSP invokes the SAM to calculate QcCrSig; the input parameters for this operation are AttSig, CrPuK, Client Data, and CrKiD (that were extracted in Registration and issuance step 13).

Registration and issuance step 15

The SAM performs the following operations:

1. The SAM validates the challenge Ch_Reg, which is embedded in the Client Data, as follows:
    a. Parse the challenge Ch_Reg from Client Data.
    b. Use Ch_Reg to retrieve the nonce cached by the SAM in Registration and issuance step 4.
    c. Reconstruct Ch_Reg' from QcPuK and the nonce according to section Ch_Reg.
    d. Verify that the challenge Ch_Reg in the Client Data is exactly equal to the reconstructed Ch_Reg'.

2. The SAM verifies the Attestation Signature (AttSig) according to section Defined Attestation Formats in the WebAuthn specification. Note that there are five different attestation formats, so the validation algorithm depends on the format used by the Authenticator. If the Attestation Signature is properly validated, that is cryptographic evidence that the user's Credential Public Key is valid and that the user gives consent for the QTSP CA to issue a Qualified Certificate for the user.

If the verification procedure is successful, the SAM creates the QcCrSig object by invoking the QSCD within the HSM. The definition of QcCrSig is:

```
QcCrSig  = sig(hash(QcPuK) || hash(CrPuK), QcPrK)
```

Hence, the QcCrSig object constitutes the cryptographic binding of the FIDO2 Credential Public Key (CrPuK) and the Qualified Certificate Public Key (QcPuK).

The SAM sets the authentication mechanism to protect the Qualified Certificate Private Key (QcPrK) in the QSCD. The QcPrK authentication mechanism is HSM implementation specific. The principle is that the QcPrK should be unlocked given that the authentication algorithm in Authentication and signing step 15 is executed successfully by the SAM.

Note: The WebAuthn registration validation operation usually involves verifying a chain of attestation certificates in order to link the Attestation Signature to a trusted attestation root CA certificate and thus verify the make and model of the Authenticator. Here the SAM verifies only the first signature in the chain - the Attestation Signature - and it is assumed that the Relying Party verifies the certificate chain in accordance with the Relying Party's attestation trust policy.

Registration and issuance step 16

The SAM returns the QcCrSig object to the QTSP.

Registration and issuance step 17

The QTSP CA issues the Qualified Certificate to the end-user and publishes the certificate at the QTSPs repository, which could be an LDAP directory or an HTTP server.

Once the Qualified Certificate has been issued, the QTSP registers the following objects in the QTSP database and associates them with the user's record:

- The QcKiD, QcPuK and UserHandle (that were cached in Registration and issuance step 6),
- the CrPuK and CrId (that were cached in Registration and issuance step 13),
- and the QcCrSig object (that was returned in Registration and issuance step 16).

Finally, the QTSP Registration Authority acknowledges to the end-user that the Qualified Certificate has been issued and is published for the user at the QTSP.

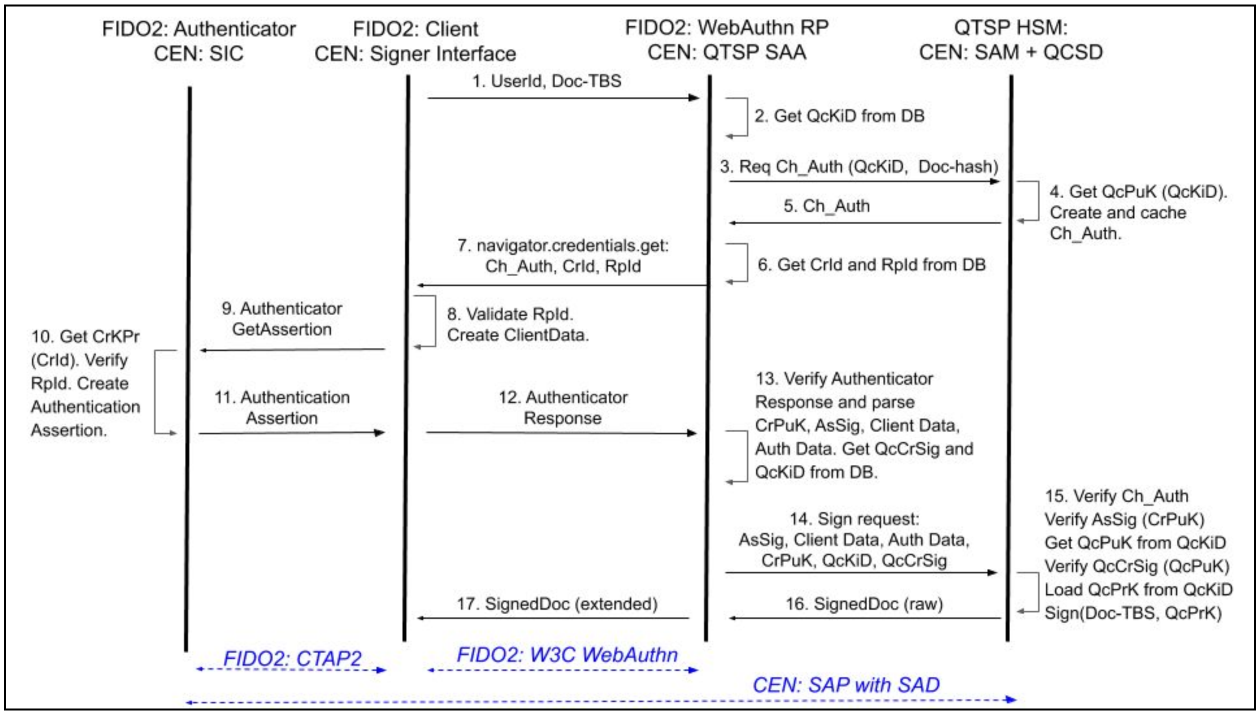
Note: In this scenario the issued Qualified Certificate is not returned to the user, but stored centrally at the QTSP CA.

Optionally, the registration process can be immediately verified with this additional step: The user may use the Authenticator to create a qualified signature according to the authentication and signature process to verify that this results in a qualified signature with the user's qualified certificate.

Authentication and signing process

Authentication and signing process overview

The process for FIDO2 authentication to a QTSP remote signing service and creating a Qualified Electronic Signature is illustrated in figure 3.



FIDO2: Authenticator
CEN: SIC
FIDO2: Client
CEN: Signer Interface
FIDO2: WebAuthn RP
CEN: QTSP SAA
QTSP HSM:
CEN: SAM + QCSD
1. UserId, Doc-TBS
2. Get QcKiD from DB
3. Req Ch_Auth (QcKiD, Doc-hash)
4. Get QcPuK (QcKiD). Create and cache Ch_Auth.
5. Ch_Auth
6. Get CrId and RpId from DB
7. navigator.credentials.get: Ch_Auth, CrId, RpId
8. Validate RpId. Create ClientData.
9. Authenticator GetAssertion
10. Get CrKPr (CrId). Verify RpId. Create Authentication Assertion.
11. Authentication Assertion
12. Authenticator Response
13. Verify Authenticator Response and parse CrPuK, AsSig, Client Data, Auth Data. Get QcCrSig and QcKiD from DB.
14. Sign request: AsSig, Client Data, Auth Data, CrPuK, QcKiD, QcCrSig
15. Verify Ch_Auth Verify AsSig (CrPuK) Get QcPuK from QcKiD Verify QcCrSig (QcPuK) Load QcPrK from QcKiD Sign(Doc-TBS, QcPrK)
16. SignedDoc (raw)
17. SignedDoc (extended)
FIDO2: CTAP2
FIDO2: W3C WebAuthn
CEN: SAP with SAD

***Figure 3** - Authentication and signing process*

Both the FIDO2 terms and the equivalent CEN EN 419 241 terms are described in figure 3 to illustrate how they are related. In the process steps below, however, only the FIDO2 terms are used if there is an overlap between the FIDO2 and CEN terminology.

Authentication and signing process steps

The steps in the authentication and signing process are described in the subsections below.

Authentication and signing step 1

The user submits its UserId and the Document To Be Signed (Doc-TBS) to the QTSP Server Signing Application (SSA).

Authentication and signing step 2

The SSA uses the UserId as key to find and retrieve the Qualified Certificate Key Identifier (QcKid) from the QTSP database.

Authentication and signing step 3

The SSA makes a call to the Signature Activation Module (SAM) with a request for a FIDO2 authentication challenge (Ch_Auth). The input parameters for this operation are the QcKid and the hash of the Doc-TBS.

Authentication and signing step 4

The SAM retrieves the QcPuK for the QcKid from the QSCD. Next, the SAM creates the Ch_Auth according to the process in section Ch_Auth. The definition of Ch_Auth is:

```
Ch_Auth = hash(hash(Doc-TBS) || hash(QcPuK) || nonce)
```

Since Ch_Auth is computed as the hash value over a nonce, the Doc-TBS hash and the QcPuK hash, there is a cryptographic binding in the FIDO2 challenge to the Doc-TBS and QcPuK.

Furthermore, the SAM caches the Ch_Auth and its nonce in secure short-term storage.

Authentication and signing step 5

The SAM returns the Ch_Auth to the SSA.

Authentication and signing step 6

The QTSP Relying Party uses the UserId as key to find and retrieve the FIDO2 credential ID (CrId) and Relying Party Identifier (RpId) from the QTSP database.

Authentication and signing step 7

The Relying Party invokes the WebAuthn navigator.credentials.get method at the Client's browser or app. As part of this call, the input parameters RpId, CrId and Ch_Auth are passed on to the Client.

Authentication and signing step 8

The Client validates the RpID and creates Client Data.

Authentication and signing step 9

The Client invokes the Authenticator by calling the AuthenticatorGetAssertion method over the CTAP2 protocol.

Authentication and signing step 10

The Authenticator retrieves the Credential Private Key (CrPrK) based on the CrId, verifies that the CrPrK is bound to the given RpId, and creates the Authentication Assertion. The Authentication Assertion contains the Ch_Auth, since its hash value is part of the Client Data.

Authentication and signing step 11

The Authenticator returns the Authentication Assertion over the CTAP2 protocol to the Client.

Authentication and signing step 12

The Client creates the Authenticator Response and passes the Authenticator Response over to the Relying Party over the WebAuthn protocol.

Authentication and signing step 13

The Relying Party validates the Authenticator Response as part of the standardized WebAuthn validation operation. As part of this validation operation, the Ch_Auth is validated. Furthermore, the Relying Party parses the Client Data, Ch_Auth, Credential Public Key (CrPuK) and Assertion Signature from the Authenticator Response. Next, the Relying Party uses the UserId as key to find and retrieve the QcCrSig and QcKiD from the database.

Authentication and signing step 14

The QTSP submits a signing request to the SAM. The input parameters to this operation are Assertion Signature, Client Data, Credential Public Key (CrPuK), the hash of the Document To Be Signed (DocTBS-Hash), Qualified Certificate Key Identifier (QcKiD), and QcCrSig.

Authentication and signing step 15

The SAM performs the following operations:

1. The SAM validates the challenge Ch_Auth as follows:
    a. Parse the challenge Ch_Auth from Client Data.
    b. Use Ch_Auth to retrieve the nonce cached by the SAM in step 4.
    c. Get QcPuK from the QSCD by using the QcKiD.
    d. Reconstruct Ch_Auth' from QcPuK, DocTBS-Hash and the nonce according to section Ch_Auth.
    e. Verify that the challenge Ch_Auth in the Client Data is exactly equal to the reconstructed Ch_Auth'.

2. The SAM verifies the Assertion Signature according to section Verify an Authentication Assertion in the WebAuthn specification:
    a. Let ClientData-Hash be the result of computing a hash over the Client Data using SHA-256.
    b. Using the Credential Public Key, verify that Assertion Signature is a valid signature over the binary concatenation of Authenticator Data and ClientData-Hash.

    If the Assertion Signature is properly validated, that is cryptographic evidence that the end-user's Credential Public Key is valid and hence that the user consents to sign the DocTBS.

3. The SAM loads the Qualified Certificate Public Key (QcPuK) from the QSCD key-store with QcKiD as identifier.

4. The SAM verifies the QcCrSig object according to the following algorithm:
    a. Let QcPuK-Hash be the result of computing a hash over the Qualified Certificate Public Key (QcPuK) using SHA-256.
    b. Let CrPuK-Hash be the result of computing a hash over the Credential Public Key (CrPuK) using SHA-256.
    c. Using the Qualified Certificate Public Key (QcPuK), verify that QcCrSig signature is a valid signature over the binary concatenation of QcPuK-Hash and CrPuK-Hash.

If the QcCrSig signature is properly validated, that is cryptographic evidence that the Qualified Certificate Key-pair is associated with the end-user's Credential Public Key and hence that the user is authorized to sign with this Qualified Certificate.

5. Given that the authentication process in steps 1-4 above is successful, the SAM loads and unlocks the Qualified Certificate Private Key (QcPrK) from the QSCD key-store with QcKiD as identifier.

6. The SAM calls the QSCD to sign the Document To Be Signed (Doc-TBS) using the Qualified Certificate Private Key (QcPrK). The output should be the raw signature format. For example, PKCS #1 (RFC 8017) with 2048-bit RSA keys and SHA-256 as hash function can be used, but other elliptic curve signature algorithms may be used as well.

Authentication and signing step 16

The raw signature is returned from the SAM to the QTSP SSA.

Authentication and signing step 17

The QTSP SSA may extend the raw signature into an ETSI standardized CAdES, XAdES or PAdES advanced signature format.

Abbreviations

List of abbreviations

API	Application Programming Interface
AsSig	Assertion Signature
AttSig	Attestation Signature
CA	Certification Authority
CAdES	CMS Advanced Electronic Signature
CC	Common Criteria
CEN	Committee European Normalization
Ch_Auth	Challenge in FIDO2 authentication ceremony
Ch_Reg	Challenge in FIDO2 registration ceremony
CMS	Cryptographic Message Syntax
CrId	Credential Identifier (for FIDO2 Authenticator)
CrK	Credential Key-pair (for FIDO2 Authenticator)
CrPrK	Credential Private Key (for FIDO2 Authenticator)
CrPuK	Credential Public Key (for FIDO2 Authenticator)
CTAP2	Client To Authenticator Protocol v2
DB	Database
Doc-TBS	Document To Be Signed
EAL	Evaluation Assurance Level
eIDAS	electronic IDentification, Authentication and trust Services
ENISA	European Union Agency for Cybersecurity

ETSI	European Telecommunications Standards Institute
FIDO2	Fast Identity Online v2
HSM	Hardware Security Module
HTTP	HyperText Transfer Protocol
PAdES	PDF Advanced Electronic Signatures
PDF	Portable Document Format
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standards
QC	Qualified Certificate
QcCrSig	Qualified Certificate Credential Signature (binding)
QcK	Qualified Certificate Key-pair
QcKId	Qualified Certificate Key Identifier
QcPrK	Qualified Certificate Private Key
QcPuK	Qualified Certificate Public Key
QES	Qualified Electronic Signature
QSCD	Qualified Signature Creation Device
QTSP	Qualified Trust Service Provider
RFC	Request For Comments
RP	Relying Party
RP ID	Relying Party Identifier
RSA	Rivest Shamir Adleman
SAD	Signature Activation Data
SAM	Signature Activation Module
SAP	Signature Activation Protocol
SCAL2	Sole Control Assurance Level
SHA	Secure Hash Algorithm
SIC	Signer's Interaction Component
SigDoc	Signed Document
SMS	Short Messaging Service
SSA	Server Signing Application
TC	Technical Committee
TLS	Transport Layer Security
TSP	Trusted Service Provider
W3C	World Wide Web Consortium
WebAuthn	Web Authentication
XAdES	XML based Advanced Electronic Signatures
XML	eXtended Markup Language

Mapping of CEN and FIDO2 terms

In the table below the CEN EN 419 241 terms are mapped to then FIDO2 terms (when applicable) and briefly explained.

CEN term	FIDO2 term	Description
Signature Activation Data (SAD)	Authenticator Response	The CEN Signature Activation Data is equivalent to the Authenticator Response in the WebAuthn

		protocol. This is the activation data to the Signature Activation Module for gaining remote access to the user's private key in the QSCD.
Signature Activation Protocol (SAP)	CTAP2 and WebAuthn	The CEN Signature Activation Protocol is equivalent to the combination of CTAP2 and WebAuthn protocols specified in the FIDO2 standard. This is the authentication protocol used to provide Signature Activation Data for remote access to the Signature Activation Module for gaining remote access to the user's private key in the QSCD.
Signer	User	The CEN Signer is equivalent to the FIDO2 User. This is the person who is operating the Client and authenticates with the Authenticator in order to sign a document remotely.
Signer's Interaction Component (SIC)	Authenticator	The CEN Signer's Interaction Component is equivalent to the FIDO2 Authenticator. This is the device with credentials used for creating the Signature Activation Data.
Signer Interface	Client	The CEN Signer Interface is equivalent to the FIDO2 Client. This is essentially the user's laptop or mobile device with a user agent (web-browser).

Table 1 *- Mapping of CEN and FIDO2 terms*

- End of the document -