

Credential Exchange Format

Proposed Standard, March 09, 2026



This version:

<https://fidoalliance.org/specs/cx/cxf-v1.0-ps-errata-20260309.html>

Issue Tracking:

[Github](#)

[GitHub](#)

Editors:

[René Léveillé](#) (1Password)

[Rew Islam](#) (Dashlane)

[Oscar Hinton](#) (Bitwarden)

[Max Crone](#) (1Password)

Contributors:

[Nick Steele](#) (1Password)

[Anders Åberg](#) (Bitwarden)

[Jonathan Salamon](#) (Dashlane)

[Ayman Bedair](#) (NordPass)

[Lee Campbell](#) (Google)

[Reema Bajwa](#) (Google)

[Hans Reichenbach](#) (Okta)

[Stephan Drab](#) (Google)

[Luc Fauvel](#) (Devolutions)

[Priya Tirounarayanane](#) (Dashlane)

Copyright © 2026 [FIDO Alliance](#). All Rights Reserved.

Abstract

This document defines the data structures and format of credentials being passed or referenced between two applications during credential exchange.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current FIDO Alliance publications and the latest revision of this technical report can be found in the [FIDO Alliance specifications index](https://www.fidoalliance.org/specifications/) at <https://www.fidoalliance.org/specifications/>.

This document was published by the [FIDO Alliance](#) as a Proposed Standard Specification. If you wish to make comments regarding this document, please [Contact Us](#). All comments are welcome.

Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The FIDO Alliance, Inc. and its Members and any other contributors to the Specification are not, and shall not be held, responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS FIDO ALLIANCE SPECIFICATION IS PROVIDED "AS IS" AND WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1	Introduction
1.1	Motivation
1.2	Scope
1.3	Terminology
2	Format Overview
2.1	Encoding Considerations
2.1.1	Enumerations as recommended
2.1.2	Array encoding
3	Data Structure Specification
3.1	Header
3.1.1	Version
3.2	Entity Data Types
3.2.1	Account Entity
3.2.2	Collection Entity
3.2.2.1	LinkedItem Dictionary
3.2.3	Item Entity
3.2.4	CredentialScope
3.2.4.1	AndroidAppId
3.2.4.1.1	AndroidAppCertificateFingerprint
3.3	Credential Data Types
3.3.1	Address
3.3.2	APIKey
3.3.3	BasicAuth
3.3.4	CreditCard
3.3.5	CustomFields Dictionary
3.3.6	DriversLicense
3.3.7	File
3.3.8	GeneratedPassword Dictionary
3.3.9	IdentityDocument
3.3.10	ItemReference Dictionary
3.3.11	Note
3.3.12	Passkey Dictionary
3.3.12.1	Editability of passkey fields
3.3.12.2	Fido2Extensions dictionary
3.3.12.3	Fido2HmacCredentials
3.3.12.4	Fido2HmacCredentialAlgorithm
3.3.12.5	Fido2LargeBlob
3.3.13	Passport
3.3.14	PersonName
3.3.15	SSHKey
3.3.16	TOTP
3.3.16.1	OTPHashAlgorithm Enumeration
3.3.17	Wi-Fi Passphrase
3.3.17.1	WIFINetworkSecurityType
3.4	Supporting Data Structures
3.4.1	CredentialType Enumeration
3.4.2	EditableField Dictionary
3.4.3	FieldType Enumeration
3.5	Defined Extension
3.5.1	Sharing an Entity (Shared)

- 3.5.1.1 SharingAccessor
- 3.5.1.2 SharingAccessorType Enumeration
- 3.5.1.3 SharingAccessorPermission Enumeration

4 IANA Considerations

- 4.1 Credential Types Registry
- 4.2 Extension Registry

5 Security Considerations

- 5.1 Orchestrating Party Requirements
 - 5.1.1 Documentation Requirements
- 5.2 Validation of Files

Appendix A: Example Payload

Conformance

Index

- Terms defined by this specification
- Terms defined by reference

References

- Normative References
- Informative References

CDDL Index

1. Introduction§

Credential migration has traditionally been an infrequent occurrence, when a user is attempting to migrate credentials from one credential provider to a new one, such as moving to a new password manager or mobile device. This has historically been a very manual process for credential providers, as there exists no normative structure to the credentials being exported by a credential provider. The goal of CXF is to define those normative data structures to allow for interoperability and control by resource owners over credentials that need to be migrated or referenced by one or more providers.

1.1. Motivation§

Historically, there is no normative structure for passing credentials between credential providers, leading to a lack of interoperability and in some cases, the loss of credentials during transfer. While the Credential Exchange Protocol aims to define the standard protocol for the import and export of credentials, there additionally needs to be a standard format for the credential data being exchanged. The Credential Exchange Format aims to solve non-normative credential transfer for this protocol and other forms of credential exchange between providers to help make the process easier for users and organizations to securely handle exchange events.

1.2. Scope§

This document outlines the data structures and format needed to exchange credentials and does not make any assumptions about the protocol used for the transfer, such as the protocol outlined by [CXP](#).

1.3. Terminology§

Credential Owner

An entity that is able to access and authenticate with the credentials stored within [a credential provider](#). The credential owner is in charge of authorizing or delegating authorization of the migration between the [\[EXP\]](#) and the [\[IMP\]](#). In the case of a credential owner being an individual or is authorized by an organization to manage these credentials, they can be referred to as the end-user.

Credential Provider

Hardware or software capable of storing and managing credentials on behalf of [a credential owner](#). While there can be assumptions in this document that the [credential providers](#) are distinct participants, a single provider MAY operate as both the [\[IMP\]](#) and [\[EXP\]](#) in an exchange.

Entity

A data structure that contains information or credentials that belong to the [credential owner](#). This is any data structure that can be contained in the [Account](#) dictionary, including the [Account](#) dictionary itself.

Exporting Provider

[\[EXP\]](#)

The exporting provider, or exporter, transfers the credential data to the [importing provider](#). The exporting provider MUST ensure the data is transferred securely by either encrypting it themselves or by relying on an orchestrator's security guarantees, see more information in section [§ 5.1 Orchestrating Party Requirements](#).

Unique Identifier

Identifier

A probabilistically-unique string of bytes identifying an [entity](#). Identifiers MUST be unique for a given exchanged [Account](#) and have a maximum of 64 bytes in length. Identifiers SHOULD NOT have any personally identifying information contained as they will be shared in clear text during any given [\[CXP\]](#) exchange sessions. An identifier for a given [entity](#) SHOULD be the same across different creations of a CXF document.

Importing Provider

[\[IMP\]](#)

The importing provider, or importer, is the final storage destination for the exported credentials.

Orchestrating Party

An orchestrating party coordinates the credential exchange between [exporting provider](#) and the [importing provider](#). This may be the operating system or other higher-level software that hosts the [importing provider](#) and the [exporting provider](#).

Transfer Protocol

The secure transfer protocol which transmits the CXF JSON document and subsequent [Credentials](#) between the [importing provider](#) and [exporting provider](#). This transfer protocol MUST ensure confidentiality between the providers following the [§ 5.1 Orchestrating Party Requirements](#). The preferred transfer protocol SHOULD be [\[CXP\]](#).

2. Format Overview§

CXF defines a schema around an account owner and all of its associated secrets. These secrets are defined in a way where the most common attributes have dedicated fields, all the while allowing extra fields to be added as extensions.

2.1. Encoding Considerations§

This document uses the Concise Data Definition Language (CDDL) as defined in [\[RFC8610\]](#) to detail the format types in the code fragments. Conforming participants MUST support encoding the types defined in this format to [\[JSON\]](#). Therefore code fragments in this document will use the JSON compatible subset of the CDDL prelude as defined in [\[RFC8610\]](#) § Appendix E with the addition of `b64url`.

The `b64url` CDDL type is a [\[JSON\]](#) string which contains a [\[RFC4648\]](#) compliant URL-safe Base 64 encoded bytes. The `b64url` tag is therefore omitted when encoded. When a field in this document specifies a length in bytes where the type is `b64url`, the length applies to the number of bytes before Base 64 encoding them.

```
b64url = tstr
```

2.1.1. Enumerations as recommended§

Unless otherwise specified, the use of enumeration types in this document will be given the choice to be represented as text strings (`tstr`). This allows for other values to be used that are not specified by the version of the current document. It's important for backwards compatibility that [importing providers](#) handle unknown values gracefully. Should the [importing provider](#) encounter an unknown enumeration value, the [importing provider](#) SHOULD follow these RECOMMENDATIONS:

- If the field member holding the unknown enumeration is OPTIONAL, the field member SHOULD be ignored as though the field was not provided at all.
- If the field member holding the unknown enumeration is REQUIRED, the entire structure holding the unknown value SHOULD be ignored as though the structure was not provided at all. This effect SHOULD cascade up to a parent structure that holds this or these child structure(s) optionally. If the exchange can no longer be continued due to this operation, the [importing provider](#) should abort the exchange.

2.1.2. Array encoding§

The format described in this document has required and optional arrays denoted in the following way in CDDL:

- Required array: `example: [* Example]`,
- Optional array: `? example: [* Example] .default []`

The required array MUST be present in the encoded payload even if the array is empty, while the optional array MUST NOT be present when the array is empty.

3. Data Structure Specification§

3.1. Header§

```
Header = {  
  version: Version,  
  exporterRpId: tstr,  
  exporterDisplayName: tstr,  
  timestamp: uint .size 8,  
  accounts: [ * Account ],  
}
```

version

The version of the format definition contained within this exchange payload. The version MUST correspond to a published level of the CXF standard.

exporterRpId

The name of the exporting app as a [relying party identifier](#).

exporterDisplayName

The display name of the exporting app to be presented to the user.

timestamp

The UNIX timestamp in seconds during at which the export document was completed.

accounts

The list of [Account](#) being exported.

3.1.1. Version§

This is a versioned format where the current document details version 1.0. Where this document uses MAJOR.MINOR format for readability, the parts are separated in the [version](#) dictionary for ease of machine comprehension.

Any participant using this format MUST ignore unknown fields or enumeration values. This is important since any of the following additions to the protocol are NOT considered breaking changes.

- New OPTIONAL fields on dictionaries
- New enumeration values, see [§2.1.1 Enumerations as recommended](#) for more details.

```
Version = {  
  major: uint .size 1,  
  minor: uint .size 1,  
}
```

major

The major version of the payload's format. Changes to this version indicates an incompatible breaking change with previous versions. The current document's major format is 1.

minor

The minor version of the payload's format. Changes to this version indicates new functionality which is purely additive and that is compatible with previous versions under the same [major](#). The current document's minor version is 0.

3.2. Entity Data Types§

3.2.1. Account Entity§

The [Account entity](#) representing a [credential owner's](#) account in the [exporting provider](#).

```
Account = {  
  id: b64url,  
  username: tstr,  
  email: tstr,  
  ? fullName: tstr,  
  collections: [ * Collection ],  
  items: [ * Item ],  
  ? extensions: [ * Extension ] .default [],  
}
```

id

A [unique identifier](#) for the [Account](#) which is machine-generated and an opaque byte sequence with a maximum size of 64 bytes. It SHOULD NOT to be displayed to the user.

username

A pseudonym defined by the user to name their account. If none is set, this should be an empty string.

email

The email used to register the account in the previous provider.

fullName

This OPTIONAL field holds the user's full name.

collections

All the collections this account owns. If the user has collections that were shared with them by another account, it MUST NOT be present in this list.

items

All items that this account owns. If the user has access to items that were shared with them by another account, it MUST NOT be present in this list.

extensions

This OPTIONAL field contains all the extensions to the [Account](#)'s attributes.

3.2.2. Collection [Entity](#)

The [Collection entity](#) allows a [credential owner](#) to organize their [Items](#) together into groups.

```
Collection = {
  id: b64url,
  ? creationAt: uint .size 8,
  ? modifiedAt: uint .size 8,
  title: tstr,
  ? subtitle: tstr,
  items: [ * LinkedItem ],
  ? subCollections: [ * Collection ] .default [],
  ? extensions: [ * Extension ] .default [],
}
```

id

A [unique identifier](#) for the [Collection](#) which is machine-generated and an opaque byte sequence with a maximum size of 64 bytes. It SHOULD NOT be displayed to the user.

creationAt

This OPTIONAL member contains the UNIX timestamp in seconds at which this [Collection](#) was originally created. If this member is not set, but the importing provider requires this member in their proprietary data model, the importer SHOULD use the current timestamp at the time the provider encounters this [Collection](#).

modifiedAt

This OPTIONAL member contains the UNIX timestamp in seconds of the last modification brought to this [Collection](#). If this member is not set, but the importing provider requires this member in their proprietary data model, the importer SHOULD use the current timestamp at the time the provider encounters this [Collection](#).

NOTE: Changes to [Items](#) in this [Collection](#) SHOULD not affect this timestamp.

title

The display name of the [Collection](#).

subtitle

This OPTIONAL field is a subtitle or a description of the [Collection](#).

items

Enumerates all the [LinkedItem](#) in this [Collection](#). A [LinkedItem](#) contains the necessary data to indicate which [Items](#) are part of this [Collection](#).

subCollections

Enumerates any sub-collections if the provider supports recursive organization.

extensions

This enumeration contains all the extensions to the [Collection](#)'s attributes.

3.2.2.1. [LinkedItem](#) Dictionary

```

LinkedItem = {
  item: b64url,
  ? account: b64url,
}

```

item

The [Item's id](#) that this [LinkedItem](#) refers to. Note that this [Item](#) might not be sent as part of the current exchange.

account

This OPTIONAL member indicates the [Account's id](#) the referenced [Item](#) belongs to. If not present, the [Item](#) belongs to the current [Account](#) being exchanged.

3.2.3. Item Entity

The [Item entity](#) contains metadata for the enclosed [Credentials](#).

```

Item = {
  id: b64url,
  ? createdAt: uint .size 8,
  ? modifiedAt: uint .size 8,
  title: tstr,
  ? subtitle: tstr,
  ? favorite: bool .default false,
  ? scope: CredentialScope,
  credentials: [ * Credential ],
  ? tags: [ * tstr ] .default [],
  ? extensions: [ * Extension ] .default [],
}

```

id

A [unique identifier](#) for the [Item](#) which is machine-generated and an opaque byte sequence with a maximum size of 64 bytes. It SHOULD NOT be displayed to the user.

createdAt

This OPTIONAL member contains the UNIX timestamp in seconds at which this item was originally created. If this member is not set, but the importing provider requires this member in their proprietary data model, the importer SHOULD use the current timestamp at the time the provider encounters this [Item](#).

modifiedAt

This OPTIONAL member contains the UNIX timestamp in seconds of the last modification brought to this [Item](#). If this member is not set, but the importing provider requires this member in their proprietary data model, the importer SHOULD use the current timestamp at the time the provider encounters this [Item](#).

title

This member's value is the user-defined name or title of the item.

subtitle

This OPTIONAL member is a subtitle or description for the [Item](#).

favorite

This OPTIONAL member denotes whether the user has marked the [Item](#) as a favorite to easily present in the UI.

scope

This OPTIONAL member defines the scope where the [credentials](#) SHOULD be presented. The credentials SHOULD only be presented within this scope unless otherwise specified by a specific [Credential](#) type.

credentials

This member contains a set of [Credentials](#) that are considered related.

tags

This OPTIONAL member contains user-defined tags that they may use to organize the item.

extensions

This member contains all the extensions the exporter MAY have to define the [Item](#) type that is being exported to be as complete of an export as possible.

3.2.4. CredentialScope

This is an object that describes an appropriate context in which the [item](#)'s [credentials](#) can to be used.

```
CredentialScope = {  
  urls: [ * uri ],  
  androidApps: [ * AndroidAppId ],  
}
```

urls

This member holds strings which SHOULD follow the Uniform Resource Identifier (URI) syntax as defined in [\[RFC3986\]](#).

androidApps

This member defines the android apps that have been validated to be appropriate for the credentials to be used.

3.2.4.1. AndroidAppId

An [AndroidAppId](#) credential contains the information required to verify and identify an [Android](#) application for automatically filling other [Credentials](#) associated to the same [item](#) as this one.

```
AndroidAppId = {  
  bundleId: tstr,  
  ? certificate: AndroidAppCertificateFingerprint,  
  ? name: tstr  
}
```

bundleId

The application identifier. A non-normative example of an application identifier is "com.example.myapplication".

certificate

The fingerprint of the public certificate used to sign the android application. This member is OPTIONAL but is highly recommended to be stored for validation during an autofill operation.

name

The [human-palatable](#) name for the application, this can be fetched from the android system when associating the app to an item. It is highly recommended for providers to store this name.

3.2.4.1.1. AndroidAppCertificateFingerprint

```
AndroidAppCertificateFingerprint = {  
  fingerprint: b64url,  
  hashAlg: "sha256" / "sha512" / tstr,  
}
```

fingerprint

This is the hash of the application's public certificate using the hashing algorithm defined in [hashAlg](#). The

bytes of the hash are then encoded into base64url directly.

hashAlg

The algorithm used to hash the [fingerprint](#). This SHOULD be of value "sha256" or "sha512".

3.3. Credential Data Types

The following **Credential** base credential defines the basic required fields a data type must have in order to be considered a credential.

[Credentials](#) contain the [credential owner's](#) secrets and are designed to be composable within the [Item entity](#). While some [exporting providers](#) MAY allow certain credential combinations within an [Item](#), [importing providers](#) MAY not support the same combinations. In such cases, the [Credentials](#) MAY be split into separate [Items](#) when processing the CXF document.

```
Credential = $Credential .within {  
  type: CredentialType / tstr  
}
```

type

This member contains a **string representation of the credential type**. The value SHOULD be a member of [CredentialType](#) but importers MAY attempt to store unknown item types in their own way as a best effort.

NOTE: The [type](#) value will be the same for all items implementing a particular credential which means that developers can rely on `obj.type` returning a string that unambiguously represents the specific kind of [Credential](#) they are dealing with.

\$Credential

The [\\$Credential socket](#) is the extension point for the [Credential](#) grouping. This socket is used in all [Credential](#) types below to expand the options that may represent a credential. This socket can also be used by other standards that may register their types in the IANA registry. See [#sctn-iana-considerations] for more information.

3.3.1. Address

An [Address](#) credential provides information for autofilling address forms.

NOTE: Because address formats vary widely around the world, the group entries are kept to a minimum with few restrictions on their types.

```
$Credential /= Address  
Address = {  
  type: "address",  
  ? streetAddress: EditableField<"string">,  
  ? postalCode: EditableField<"string">,  
  ? city: EditableField<"string">,  
  ? territory: EditableField<"subdivision-code">,  
  ? country: EditableField<"country-code">,  
  ? tel: EditableField<"string">,  
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [address](#).

streetAddress

The address line for the address. This is intentionally flexible to accommodate different address formats.

Implementers MUST support multi-line addresses for this field, where each line is separated by a `\n` line feed. This field is OPTIONAL.

postalCode

The ZIP or postal code for the address. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

city

The city for the address. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

territory

The province, state, or territory for the address. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `subdivision-code`.

country

The country for the address. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `country-code`.

tel

The phone number associated with the address. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

3.3.2. APIKey

A `APIKey` credential contains information to interact with an Application's Programming Interface (API).

```
$Credential /= APIKey
APIKey = {
  type: "api-key",
  ? key: EditableField<"concealed-string">,
  ? username: EditableField<"string">,
  ? keyType: EditableField<"string">,
  ? url: EditableField<"string">,
  ? validFrom: EditableField<"date">,
  ? expiryDate: EditableField<"date">,
}
```

type

This overridden member from `Credential` MUST be present and MUST have a value of `api-key`.

key

The key to communicate with the API. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `concealed-string`.

username

The username associated with the key. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

keyType

The type of the API key, such as bearer token or JSON Web Token. This field is flexible to allow any type and not restrict it to a set list of types. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

url

The url the API key is used with. This SHOULD conform to the `URL Standard`. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `string`.

validFrom

The date the API key is valid from. This field is OPTIONAL and its internal `fieldType` SHOULD be of type `date`.

expiryDate

The date on which the API key expires. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

3.3.3. BasicAuth

A [BasicAuth](#) credential contains a username/password login credential. Can either represent a [Basic access authentication](#) or a form on a web page.

A [BasicAuth](#) SHOULD have an accompanying [CredentialScope](#) in the [credentials](#) array. This indicates in which websites or applications these fields SHOULD be presented.

```
$Credential /= BasicAuth
BasicAuth = {
  type: "basic-auth",
  ? username: EditableField<"string">,
  ? password: EditableField<"concealed-string">
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [basic-auth](#).

username

The username associated with the credential. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

password

The password associated with the credential. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [concealed-string](#).

3.3.4. CreditCard

A [CreditCard](#) credential contains information about a credit or debit card.

NOTE: The Payment Card Industry Data Security Standard prohibits storing payment card values for `pin` and `verificationNumber` as outlined in Section 3.3.1 of [\[PCI-DSS-4.0.1\]](#).

```
$Credential /= CreditCard
CreditCard = {
  type: "credit-card",
  ? number: EditableField<"concealed-string">,
  ? fullName: EditableField<"string">,
  ? cardType: EditableField<"string">,
  ? verificationNumber: EditableField<"concealed-string">,
  ? pin: EditableField<"concealed-string">,
  ? expiryDate: EditableField<"year-month">,
  ? validFrom: EditableField<"year-month">,
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [credit-card](#).

number

The credit card number. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [concealed-string](#).

fullName

The full name printed on the card. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

cardType

The vendor of the card. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

verificationNumber

The verification number/value/code (CVV, CVC). This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [concealed-string](#).

pin

The personal identification number (PIN). This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [concealed-string](#).

expiryDate

The expiry date of the card. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [year-month](#).

validFrom

The date from which the card is valid. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [year-month](#).

3.3.5. CustomFields Dictionary

This credential allows the user to add additional information to the [item](#) organized as a grouping that MAY have a label. If the [exporting provider](#) allows custom fields to be added to items but does not have a grouping concept, it SHOULD use this object without setting the label or id fields.

```
$Credential /= CustomFields
CustomFields = {
  type: "custom-fields",
  ? id: b64url,
  ? label: tstr,
  fields: [ * EditableField<FieldType> ],
  ? extensions: [ + Extension ] .default [],
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [custom-fields](#).

id

An OPTIONAL unique identifier for the [CustomFields](#). It MUST be a machine-generated opaque byte sequence with a maximum size of 64 bytes. It SHOULD NOT be displayed to the user.

label

This OPTIONAL member is a [human-palatable](#) title to describe the section. This value MAY be set by the [credential owner](#).

fields

The collection of miscellaneous fields under this section. The internal [fieldType](#) SHOULD be a member of [FieldType](#).

extensions

This OPTIONAL member permits the [exporting provider](#) to add additional information associated to this [CustomFields](#). This MAY be used to provide an exchange where a minimal amount of information is lost.

3.3.6. DriversLicense

A [DriversLicense](#) credential contains information about a person's driver's license. The fields reflect the relevant set of mandatory data fields defined by [ISO 18013-1](#).

```

$Credential /= DriversLicense
DriversLicense = {
  type: "drivers-license",
  ? fullName: EditableField<"string">,
  ? birthDate: EditableField<"date">,
  ? issueDate: EditableField<"date">,
  ? expiryDate: EditableField<"date">,
  ? issuingAuthority: EditableField<"string">,
  ? territory: EditableField<"subdivision-code">,
  ? country: EditableField<"country-code">,
  ? licenseNumber: EditableField<"string">,
  ? licenseClass: EditableField<"string">,
}

```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [drivers-license](#).

fullName

The full name of the license holder. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

birthDate

Day, month, and year on which the license holder was born. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

issueDate

The date on which the license was issued. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

expiryDate

The date on which the license expires. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

issuingAuthority

The official body or government agency responsible for issuing the license. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

territory

The principal administrative subdivision of the license's country of origin. Examples of administrative subdivisions are states or provinces. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [subdivision-code](#).

country

The license's country of origin. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [country-code](#).

licenseNumber

The number assigned by the issuing authority. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

licenseClass

The vehicle types the license holder is authorized to operate. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

3.3.7. File

A [File](#) credential acts as a placeholder to an arbitrary binary file holding its associated metadata. The [transfer protocol](#) SHOULD provide a method to transfer securely and confidentially transfer the binary file after the [importing provider](#) has consumed this credential data.

```
$Credential /= File
File = {
  type: "file",
  id: b64url,
  name: tstr,
  decryptedSize: uint,
  integrityHash: b64url
}
```

id

The file's identifier, used to identify the package when it is provided by the [transfer protocol](#).

name

The file name with the file extension if applicable.

decryptedSize

The file's decrypted size in bytes.

integrityHash

The SHA256 hash of the decrypted file. This hash MUST be used by the [importing provider](#) when the file is decrypted to ensure that it has not been corrupted.

3.3.8. GeneratedPassword Dictionary

A [GeneratedPassword](#) credential type represents a credential consisting of a machine-generated password.

NOTE: A GeneratedPassword credential is used when a password is generated independently of creating a new BasicAuth credential. Some providers may offer a dedicated password generator feature. In such cases, the provider may create GeneratedPassword instances as deemed appropriate for the use of this feature.

```
$Credential /= GeneratedPassword
GeneratedPassword = {
  type: "generated-password",
  password: tstr,
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [generated-password](#).

password

The machine-generated password. The UI of this field SHOULD follow the requirements of [concealed-string](#).

3.3.9. IdentityDocument

An [IdentityDocument](#) credential is for any document, card, or number identifying a person or entity. Examples include national ID cards, Social Security Numbers (SSN), Tax Identification Numbers (TIN), health insurance cards, or Value-Added Tax (VAT) numbers.

Credentials like the SSN can still be encoded as an [IdentityDocument](#) by only providing the [identificationNumber](#) field, since the others are generally considered to be undefined in its case.

NOTE: Driver's licenses and passports may be accepted as identity verification in some countries, but they are specified separately in the [DriversLicense](#) and [Passport](#) credential types, respectively.

```

$Credential /= IdentityDocument
IdentityDocument = {
  type: "identity-document",
  ? issuingCountry: EditableField<"country-code">,
  ? documentNumber: EditableField<"string">,
  ? identificationNumber: EditableField<"string">,
  ? nationality: EditableField<"string">,
  ? fullName: EditableField<"string">,
  ? birthDate: EditableField<"date">,
  ? birthPlace: EditableField<"string">,
  ? sex: EditableField<"string">,
  ? issueDate: EditableField<"date">,
  ? expiryDate: EditableField<"date">,
  ? issuingAuthority: EditableField<"string">,
}

```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [identity-document](#).

issuingCountry

The document's issuing country. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [country-code](#).

documentNumber

The document's identifying number. This identifying number is tied to the issuance of the document and is expected to change upon its reissuance, even when the person's information might remain the same. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

identificationNumber

The person's or other entity's identification number. This identifying number is generally expected to remain stable across reissuances of the identity document itself. For identification numbers that are not an identity document (e.g., SSN, TIN, or VAT), this field is generally the only one that's expected to be present in the credential. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

nationality

The person's nationality. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

fullName

The person's full name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

birthDate

The person's date of birth. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

birthPlace

The person's place of birth. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

sex

The person's sex or gender. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

issueDate

The date on which the document was issued. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

expiryDate

The date on which the document expires. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

issuingAuthority

The official body or government agency responsible for issuing the document. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

3.3.10. ItemReference Dictionary

An [ItemReference](#) credential is a pointer to another [Item](#), denoting that the two items MAY be logically linked together.

```
$Credential /= ItemReference
ItemReference = {
  type: "item-reference",
  reference: LinkedItem,
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [item-reference](#).

reference

A [LinkedItem](#) which references another [Item](#).

NOTE: The other [item](#) SHOULD be in the exchange if it is owned by the same [Account](#). However, the other item MAY NOT be in the exchange if it is owned by a different account and shared with the currently exchanged account.

3.3.11. Note

```
$Credential /= Note
Note = {
  type: "note",
  content: EditableField<"string">,
};
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [note](#).

content

This member is a user-defined value encoded as a UTF-8 string. This field's internal [fieldType](#) SHOULD be of type [string](#).

3.3.12. Passkey Dictionary

The private credential which is used for a [\[webauthn-3\]](#) ceremony. This credential MUST NOT be used outside of the scope defined by the enclosed [rpId](#). The associated [CredentialScope](#) MUST NOT apply to credentials of this type.

```
$Credential /= Passkey
Passkey = {
  type: "passkey",
  credentialId: b64url,
  rpId: tstr,
  username: tstr,
  userDisplayName: tstr,
  userHandle: b64url,
  key: b64url,
  ? fido2Extensions: Fido2Extensions,
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [passkey](#).

credentialId

This member contains a [WebAuthn Credential ID](#) which uniquely identifies the passkey instance. The decoded raw value MUST be equal to the value given in [PublicKeyCredential](#)'s [rawId](#) field during [registration](#).

rpId

This member specifies the [WebAuthn Relying Party Identifier](#) to which the passkey instance is tied to. The value MUST be equal to the [RP ID](#) that was defined by the authenticator during [credential registration](#).

username

This member contains a [human-palatable](#) identifier for the [user account](#) to which the passkey instance is tied to. The value SHOULD be equal to the value in [PublicKeyCredentialUserEntity](#)'s [name](#) member given to the authenticator during [registration](#).

The only case where the value MAY not be the one set during [registration](#) is if the [exporting provider](#) allows the user to edit their username. In such a case, the value of this field MUST be the user edited value. See [§ 3.3.12.1 Editability of passkey fields](#) for more details.

userDisplayName

This member represents a [human-palatable](#) name for the [user account](#), intended only for display. The value SHOULD be equal to the value in [PublicKeyCredentialUserEntity](#)'s [displayName](#) member given to the authenticator during [registration](#).

The only case where the value MAY not be the one set during [registration](#) is if the [exporting provider](#) allows the user to edit their user display name. In such a case, the value of this field MUST be the user edited value. See [§ 3.3.12.1 Editability of passkey fields](#) for more details.

userHandle

This member contains the [user handle](#) which is the value used to identify the [user account](#) associated to this passkey instance. The value MUST be equal to the value in [PublicKeyCredentialUserEntity](#)'s [id](#) member given to the authenticator during [registration](#).

key

The [private key](#) associated to this passkey instance. The value MUST be [PKCS#8 ASN.1 DER](#) formatted byte string which is then [Base64url](#) encoded. The value MUST give the same [public key](#) value that was provided by the original authenticator during [registration](#).

fido2Extensions

This OPTIONAL member denotes the [WebAuthn](#) or [CTAP2](#) extensions that are associated to this passkey instance.

NOTE: Passkeys using a non-zero signature counter MUST be excluded from the export and the exporter SHOULD inform the user that such passkeys are excluded from the export. Importers MUST set a zero value for the imported passkey signature counters and MUST NOT increment them after the fact.

3.3.12.1. Editability of passkey fields

Note that there are certain members of the [Passkey](#) dictionary that are marked as being editable by the user. Only [human-palatable](#) values MAY be edited by the user since these are not REQUIRED for [WebAuthn](#) ceremonies. These member also represent values that MAY be changed by the user on the [relying party](#). [Exporting providers](#) MAY let users to edit these members to mirror the changes on the [relying party](#). In such cases the value at the time of exchange MUST be the user edited value. The only accepted user editable [Passkey](#) fields are:

- [username](#)
- [userDisplayName](#)

All other members of the [Passkey](#) dictionary MUST NOT be user editable as they are required for the [WebAuthn](#) ceremonies to be successful.

3.3.12.2. *Fido2Extensions* dictionary

The [Fido2Extensions](#) dictionary holds extension data that is directly associated to a [Passkey](#) credential. The extensions supported are defined in either [\[webauthn-3\]](#) or [\[FIDO-V2.1\]](#).

```
Fido2Extensions = {  
  ? hmacCredentials: Fido2HmacCredentials,  
  ? credBlob: b64url,  
  ? largeBlob: Fido2LargeBlob,  
  ? payments: bool,  
}
```

hmacCredentials

This OPTIONAL member holds the information necessary for either the [\[webauthn-3\] prf extension](#) or the [\[FIDO-V2.1\] hmac-secret extension](#).

credBlob

This OPTIONAL member holds the information necessary for the [\[FIDO-V2.1\] credential blob extension](#). The value is a base64url-encoded byte string of the stored binary blob.

largeBlob

This OPTIONAL member holds the information necessary for the [\[webauthn-3\] large blob storage extension](#).

payments

This OPTIONAL member denotes whether this credential is used for [\[secure-payment-confirmation\]](#).

3.3.12.3. *Fido2HmacCredentials*

This extension is used to export a credential that supports the [\[webauthn-3\] Pseudo Random Function \(PRF\) extension](#) or the [\[FIDO-V2.1\] hmac-secret extension](#). Since the primary use of these extensions is encryption, it is of utmost importance that the operation generating the symmetric key MUST provide the same output regardless of the provider generating the secret.

It is the [exporting provider's](#) responsibility to provide the final byte sequence used as the HMAC seed. Regardless of how the credentials enclosed are created, the [importing provider](#) MUST store and use these credential as-is during the HMAC operation. There MUST NOT be any additional derivation or domain separators.

Should the original [exporting provider](#) only support one of the two credentials, that [exporting provider](#) MUST generate the missing credential and associate it to the existing credential. This newly generated credential MUST be identical across all future exchange sessions. If these credentials are exchanged to multiple [importing providers](#) who support both values, they MUST all have the same value.

```
Fido2HmacCredentials = {  
  algorithm: Fido2HmacCredentialAlgorithm / tstr,  
  credWithUV: b64url,  
  credWithoutUV: b64url,  
}
```

algorithm

The HMAC algorithm used to generate the shared secret from the enclosed credentials and a given salt. The value SHOULD be a member of [Fido2HmacCredentialAlgorithm](#). [Importing providers](#) that encounter an unknown algorithm SHOULD ignore this entry.

credWithUV

The credential associated to be used when user verification was performed during the authentication ceremony. This value SHOULD be 32 bytes in length which is then base64url-encoded.

credWithoutUV

The credential associated to be used when user verification was not performed during the authentication ceremony. This value SHOULD be 32 bytes in length which is then base64url-encoded.

3.3.12.4. *Fido2HmacCredentialAlgorithm*

[Fido2HmacCredentialAlgorithm](#) lists the HMAC algorithms currently recognized by the specification to be used for shared secret generation for the [Fido2HmacCredentials](#) extension.

```
Fido2HmacCredentialAlgorithm =  
  "hmac-sha256"
```

hmac-sha256

The official algorithm for [\[FIDO-V2.1\] hmac-secret extension](#). This SHOULD be the default algorithm for providers providing support for [\[webauthn-3\] prf extension](#).

3.3.12.5. *Fido2LargeBlob*

This dictionary holds the associated data for the [\[webauthn-3\] large blob extension](#).

```
Fido2LargeBlob = {  
  uncompressedSize: uint,  
  data: b64url,  
}
```

uncompressedSize

The claimed uncompressed size of the DEFLATE compressed data in [data](#). This corresponds to the [origSize](#) field from the specification's section [6.10.4 Reading per-credential large-blob data](#)

data

The contents of the large blob value which has been DEFLATE compressed.

3.3.13. *Passport*

A [Passport](#) credential contains the details of a person's passport. The fields reflect the relevant set of data elements defined by [ICAO Doc 9303 Part 4](#).

```
\$Credential /= Passport  
Passport = {  
  type: "passport",  
  ? issuingCountry: EditableField<"country-code">,  
  ? passportType: EditableField<"string">,  
  ? passportNumber: EditableField<"string">,  
  ? nationalIdentificationNumber: EditableField<"string">,  
  ? nationality: EditableField<"string">,  
  ? fullName: EditableField<"string">,  
  ? birthDate: EditableField<"date">,  
  ? birthPlace: EditableField<"string">,  
  ? sex: EditableField<"string">,  
  ? issueDate: EditableField<"date">,  
  ? expiryDate: EditableField<"date">,  
  ? issuingAuthority: EditableField<"string">,  
}
```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [passport](#).

issuingCountry

The passport's issuing country. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [country-code](#).

passportType

The passport's document type. This MUST be a valid document code as defined in [ICAO Doc 9303 Part 4](#). This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

passportNumber

The passport's identifying number. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

nationalIdentificationNumber

The person's national identification number. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

nationality

The person's nationality. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

fullName

The person's full name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

birthDate

The person's date of birth. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

birthPlace

The person's place of birth. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

sex

The person's sex or gender. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

issueDate

The date on which the passport was issued. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

expiryDate

The date on which the passport expires. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [date](#).

issuingAuthority

The official body or government agency responsible for issuing the passport. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

3.3.14. PersonName

A [PersonName](#) credential represents a person's name as fields derived from [Unicode Locale Data Markup Language Part 8: Person Names](#).

All fields are marked as optional because an exporting provider SHOULD refrain from making decisions about splitting up a name into any parts that were not explicitly provided as such, since that often introduces errors.

```

$Credential /= PersonName
PersonName = {
  type: "person-name",
  ? title: EditableField<"string">,
  ? given: EditableField<"string">,
  ? givenInformal: EditableField<"string">,
  ? given2: EditableField<"string">,
  ? surnamePrefix: EditableField<"string">,
  ? surname: EditableField<"string">,
  ? surname2: EditableField<"string">,
  ? credentials: EditableField<"string">,
  ? generation: EditableField<"string">,
}

```

title

The person's title or honorific qualifier. For example, "Ms.", "Mr.", or "Dr". This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

given

The person's given name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

givenInformal

The person's nickname or preferred name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

given2

The person's additional names or middle names. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

surnamePrefix

The prefix of the person's surname. For example, "van der" in "van der Poel" or "bint" in "bint Fadi". This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

surname

The person's family name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

surname2

The person's secondary surname, which is used in some cultures. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

credentials

The person's credential or accreditation qualifier. For example, "PhD" or "MBA". This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

generation

The person's generation qualifier. For example, "Jr." or "III". This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

3.3.15. SSHKey

An [SSHKey](#) credential represents an SSH (Secure Shell) key pair.

```

$Credential /= SSHKey
SSHKey = {
  type: "ssh-key",
  keyType: tstr,
  privateKey: b64url,
  ? keyComment: tstr,
  ? creationDate: EditableField<"date">,
  ? expiryDate: EditableField<"date">,
  ? keyGenerationSource: EditableField<"string">,
}

```

keyType

The type of SSH key algorithm used. Common values include "ssh-rsa", "ssh-ed25519", or "ecdsa-sha2-nistp256". This MUST be a string value representing a valid SSH public key algorithm as defined in [IANA SSH Protocol Parameters](#).

privateKey

The private part of the SSH key pair. This MUST be a [PKCS#8 ASN.1 DER](#) formatted byte string which is then [Base64url](#) encoded.

keyComment

This OPTIONAL member contains a user-defined string to identify or describe the key.

creationDate

This OPTIONAL member indicates when the key was created. When present, its internal [fieldType](#) SHOULD be of type [date](#).

expiryDate

This OPTIONAL member indicates when the key will expire, if applicable. When present, its internal [fieldType](#) SHOULD be of type [date](#).

keyGenerationSource

This OPTIONAL member indicates where the key was originally generated. E.g., <https://github.com/settings/ssh/new> for GitHub. When present, its internal [fieldType](#) SHOULD be of type [string](#).

3.3.16. TOTP

NOTE: Enrollment in TOTP credentials historically has been quite non-standardized but typically authenticator and RP implementations have more or less aligned with the early Google Authenticator implementation spelled out at <https://github.com/google/google-authenticator/wiki/Key-Uri-Format>. This specification was designed with that in mind.

```

$Credential /= TOTP
TOTP = {
  type: "totp",
  secret: tstr,
  period: uint .size 2,
  digits: uint .size 2,
  ? username: tstr,
  algorithm: OTPHashAlgorithm / tstr,
  ? issuer: tstr,
}

```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [totp](#).

secret

The [shared secret](#) used to generate the OTPs. This MUST be a [Base32](#) string.

period

The time step used to refresh the OTP in seconds. The default SHOULD be 30 seconds, although the [relying party](#) MAY customize this to a different value.

digits

The number of digits to generate and display to the user each period. The default SHOULD be 6, although the [relying party](#) MAY customize this to a different value.

username

This OPTIONAL member contains the username of the account this [TOTP](#) credential is used for. This is sometimes referred to as the account name.

NOTE: While this member is optional, it is strongly recommended to be included if available.

algorithm

The algorithm used to generate the OTP hashes. This value SHOULD be a member of [OTPHashAlgorithm](#) but importers MUST ignore [TOTP](#) entries with unknown algorithm values. The default SHOULD be [sha1](#), although the relying party MAY customize this to a different value.

issuer

This OPTIONAL member contains the relying party that issued the credential and should be user consumable.

NOTE: While this member is optional, it is strongly recommended to be included if available.

3.3.16.1. OTPHashAlgorithm Enumeration

```
OTPHashAlgorithm =
  "sha1" /
  "sha256" /
  "sha512"
```

sha1

This algorithm denotes that [SHA1](#) MUST be used to generate the OTP hash.

sha256

This algorithm denotes that [SHA256](#) MUST be used to generate the OTP hash.

sha512

This algorithm denotes that [SHA512](#) MUST be used to generate the OTP hash.

3.3.17. Wi-Fi Passphrase

A [WIFI](#) credential provides the necessary information to connect to a Wi-Fi network. It can be used to connect to networks secured with a passphrase or to unsecured networks that do not require authentication.

NOTE: The fields of the [wifi](#) type can be utilized to construct a URI as specified in Section 7 of [\[WPA3\]](#). This URI can then be encoded into a QR code, facilitating the provisioning of credentials to a device in a user-friendly manner.

```

$Credential /= WIFI
WIFI = {
  type: "wifi",
  ? ssid: EditableField<"string">,
  ? networkSecurityType: EditableField<"wifi-network-security-type"> /
    EditableField<"string">,
  ? passphrase: EditableField<"concealed-string">,
  ? hidden: EditableField<"boolean">
}

```

type

This overridden member from [Credential](#) MUST be present and MUST have a value of [wifi](#).

ssid

This field represents the Wi-Fi network name. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [string](#).

networkSecurityType

This field represents the Wi-Fi network security type. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [wifi-network-security-type](#) or [string](#).

passphrase

This field represents the Wi-Fi network passphrase. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [concealed-string](#).

hidden

This field represents whether the Wi-Fi network operates in hidden mode. This field is OPTIONAL and its internal [fieldType](#) SHOULD be of type [boolean](#).

3.3.17.1. WIFINetworkSecurityType§

```

WIFINetworkSecurityType = {
  "unsecured" /
  "wpa-personal" /
  "wpa2-personal" /
  "wpa3-personal" /
  "wep"
}

```

unsecured

This member specifies that the Wi-Fi network security is unsecured.

wpa-personal

This member specifies that the Wi-Fi network security is based on WPA-Personal.

wpa2-personal

This member specifies that the Wi-Fi network security is based on WPA2-Personal.

wpa3-personal

This member specifies that the Wi-Fi network security is based on WPA3-Personal.

wep

This member specifies that the Wi-Fi network security is based on WEP.

3.4. Supporting Data Structures§

3.4.1. CredentialType Enumeration§

```
CredentialType =  
  "address" /  
  "api-key" /  
  "basic-auth" /  
  "credit-card" /  
  "custom-fields" /  
  "drivers-license" /  
  "file" /  
  "generated-password" /  
  "identity-document" /  
  "item-reference" /  
  "note" /  
  "passkey" /  
  "passport" /  
  "person-name" /  
  "ssh-key" /  
  "totp" /  
  "wifi"
```

address

This type denotes that the remaining members of the containing dictionary MUST be an [Address](#).

api-key

This type denotes that the remaining members of the containing dictionary MUST be an [APIKey](#).

basic-auth

This type denotes that the remaining members of the containing dictionary MUST be a [BasicAuth](#).

credit-card

This type denotes that the remaining members of the containing dictionary MUST be a [CreditCard](#).

custom-fields

This type denotes that the remaining members of the containing dictionary MUST be a [CustomFields](#).

drivers-license

This type denotes that the remaining members of the containing dictionary MUST be a [DriversLicense](#).

file

This type denotes that the remaining members of the containing dictionary MUST be a [File](#).

generated-password

This type denotes that the remaining members of the containing dictionary MUST be a [GeneratedPassword](#).

identity-document

This type denotes that the remaining members of the containing dictionary MUST be an [IdentityDocument](#).

item-reference

This type denotes that the remaining members of the containing dictionary MUST be an [ItemReference](#).

note

This type denotes that the remaining members of the containing dictionary MUST be a [Note](#).

passkey

This type denotes that the remaining members of the containing dictionary MUST be a [Passkey](#).

passport

This type denotes that the remaining members of the containing dictionary MUST be a [Passport](#).

person-name

This type denotes that the remaining members of the containing dictionary MUST be a [PersonName](#).

ssh-key

This type denotes that the remaining members of the containing dictionary MUST be an [SSHKey](#).

totp

This type denotes that the remaining members of the containing dictionary MUST be a [TOTP](#).

wifi

This type denotes that the remaining members of the containing dictionary MUST be a [WIFI](#).

3.4.2. EditableField Dictionary

This structure defines a field which is editable by the [credential owner](#). Any structure using this as a field MUST define the expected [fieldType](#). However, in accordance with [§ 2.1.1 Enumerations as recommended](#), an [importing provider](#) that encounters an unknown [FieldType](#) SHOULD ignore this structure entirely and act as though the member holding this structure was not provided. If the member holding this structure is REQUIRED, the [importing provider](#) SHOULD ignore the [Credential](#) where that member is defined.

```
EditableField<F> = {  
  ? id: b64url,  
  fieldType: F / tstr,  
  value: tstr,  
  ? label: tstr,  
  ? extensions: [ + Extension ] .default [],  
}
```

id

An OPTIONAL unique identifier for the [EditableField](#). This MUST be a machine-generated opaque byte sequence with a maximum size of 64 bytes. It SHOULD NOT be displayed to the user.

NOTE: The ID can be used to identify an [EditableField](#) used in multiple [Credentials](#) on the same [Item](#). It can also be used to reference to this specific field within a [Credential](#) from another [Credential](#) or from an [Extension](#).

F

The F generic slot is used to indicate the expected field type in a credential. The generic value MUST be a member of [FieldType](#) OR [FieldType](#) itself if the field can contain any type.

fieldType

This member defines the meaning of the [value](#) member and its type. This meaning is two-fold:

1. The string representation of the value if its native type is not a string.
2. The UI representation used to display the value.

The value SHOULD belong to the [F](#) generic slot and the [importing provider](#) SHOULD ignore any unknown values and default to [string](#).

value

This member contains the [fieldType](#) defined by the user.

label

This member contains a user facing value describing the value stored. This value MAY be user defined.

extensions

This OPTIONAL member permits the [exporting provider](#) to add additional information associated to this [EditableField](#). This MAY be used to provide an exchange where a minimal amount of information is lost.

3.4.3. FieldType Enumeration

```
FieldType =
  "string" /
  "concealed-string" /
  "email" /
  "number" /
  "boolean" /
  "date" /
  "year-month" /
  "wifi-network-security-type" /
  "country-code" /
  "subdivision-code"
```

string

A UTF-8 encoded string value which is unconcealed and does not have a specified format.

concealed-string

A UTF-8 encoded string value which should be considered secret and not displayed unless the user explicitly requests it.

email

A UTF-8 encoded string value which follows the format specified in [\[RFC5322\]](#) Section 3.4. This field SHOULD be unconcealed.

number

A stringified numeric value which is unconcealed.

boolean

A boolean value which is unconcealed. It MUST be of the values "true" or "false".

date

A string value representing a calendar date which follows the full-date format specified in [\[RFC3339\]](#). This is equivalent to the YYYY-MM-DD format specified in [\[ISO-8601\]](#).

year-month

A string value representing a calendar date which follows the date-fullyear "-" date-month pattern as established in [\[RFC3339\]](#) Appendix A. This is equivalent to the YYYY-MM format specified in [\[ISO-8601\]](#).

wifi-network-security-type

A string value representing a value that SHOULD be a member of [WIFINetworkSecurityType](#).

country-code

A string value which MUST follow the [\[ISO3166-1\]](#) alpha-2 format.

subdivision-code

A string which MUST follow the [\[ISO3166-2\]](#) format.

3.5. Defined Extensions

```
Extension = $Extension .within {
  name: tstr
  ; Should there be an included schema? or use a URI to define the schema?
}
```

name

The name of the extension which will define the contents associated. If the extension is defined in this document then the value will directly use that name. If this is a custom extension defined by the exporter, then the value MUST take the following format: EXPORTER_RP_ID/EXTENSION_NAME. As an example provider.example/VaultType.

\$Extension

The [\\$Extension socket](#) is the extension point for the [Extension](#) grouping. This socket is used in all defined [Extensions](#) below to expand the options that may represent an [Entity](#). This socket can also be used by

other standards that may register their types in the IANA registry. See [#sctn-iana-considerations] for more information.

3.5.1. Sharing an [Entity](#) (Shared)[§]

[Entities](#) are shared by applying the [Shared](#) extension to them. This extensions MUST only be applied to [Collections](#) and [Items](#).

[Entities](#) that are shared MUST only be included in the exports for accounts that are [credential owners](#) or admins of the [entity](#).

```
$Extension /= Shared
Shared = {
  name: "shared",
  accessors: [ * SharingAccessor ],
}
```

name

This overridden member from [Extension](#) MUST be present and MUST have a value of "shared".

accessors

A list of [SharingAccessor](#) objects that represents users or groups and their permissions with respect to access on the [entity](#) to which the [Shared](#) extension is applied.

3.5.1.1. [SharingAccessor](#)[§]

A [SharingAccessor](#) represents a user or group and their access permissions with respect to an [entity](#).

```
SharingAccessor = {
  type: SharingAccessorType / tstr,
  accountId: b64url,
  name: tstr,
  permissions: [ * SharingAccessorPermission / tstr ],
}
```

type

This member indicates the type of accessor for which permissions are defined. The value SHOULD be a member of [SharingAccessorType](#). Importers MUST ignore any [SharingAccessor](#) entries with an unknown [type](#).

accountId

This member specifies the account, identified by its [Account id](#), that has been given access to the shared [entity](#) by the current [Account](#).

name

This member contains the accessor's account name. If [type](#) has the value [user](#) this SHOULD be set to the [username](#). If [type](#) has the value [group](#) this SHOULD be set to the group's name.

permissions

This member lists the permissions that this [accountId](#) has with respect to access on the shared [entity](#). The values SHOULD be members of [SharingAccessorPermission](#). Importers MUST ignore unknown entries. Importers MUST ignore any [SharingAccessors](#) that have an empty [permissions](#) list, whether it's been exported as empty or when it's empty as a result of ignoring all unknown entries.

3.5.1.2. [SharingAccessorType](#) Enumeration[§]

A [SharingAccessorType](#) indicates the type of accessor for which a [SharingAccessor](#) defines access permissions to the respective [entity](#). This MUST be either "user" or "group".

```
SharingAccessorType =  
    "user" /  
    "group"
```

user

Indicates the respective [SharingAccessor](#) is describing a user's permissions on the shared [entity](#).

group

Indicates the respective [SharingAccessor](#) is describing a group of users' permissions on the shared entity.

3.5.1.3. *SharingAccessorPermission Enumeration*

The [SharingAccessorPermission](#) enumeration encodes the level of access the accessor is given to the respective [entity](#).

```
SharingAccessorPermission =  
    "read" /  
    "readSecret" /  
    "update" /  
    "create" /  
    "delete" /  
    "share" /  
    "manage"
```

read

Indicates that the respective [SharingAccessor](#) has read permissions on the associated [entity](#), excluding its secrets. This generally means that the client prevents the user from revealing the secret (e.g., a password) in its interface. However, the user is often still allowed to use the secrets in an autofill context.

NOTE: Users can technically extract the secrets of the shared entities if they are allowed to use them in autofill contexts. This aspect of the permission therefore acts more as a client-side restriction instead.

readSecret

Indicates that the respective [SharingAccessor](#) has read permissions on the associated [entity](#), including its secrets.

update

Indicates that the respective [SharingAccessor](#) has update permissions on the associated [entity](#).

create

Indicates that the respective [SharingAccessor](#) has the permission to create sub-entities for the associated [entity](#), if applicable.

delete

Indicates that the respective [SharingAccessor](#) has the permission to delete any of the associated [entity's](#) sub-entities, if applicable.

share

Indicates that the respective [SharingAccessor](#) can share any of the associated [entity's](#) sub-entities with users or groups, if applicable.

manage

Indicates that the respective [SharingAccessor](#) can manage the associated [entity](#), meaning they can edit the [entity's](#) attributes, share it with others, etc.

4. IANA Considerations§

A request to IANA has been put forth in an upcoming RFC to create a new registry for Credential Exchange. This request follows the rules in [\[RFC8126\]](#), and requests that two registries be created under the same umbrella registry.

4.1. Credential Types Registry§

The request includes a registry for [Credential](#) entries. A "Credential Exchange Format Credential Types" registry using the following template for new registration requests:

- Credential type identifier (JSON compatible string)
- Description
- Requires an additional payload (Y/N)
- Specification Document (URL)

The initial values for this table are the credentials defined in the document and are as follows:

Address

- Credential type identifier: address
- Description: A credential representing a street address.
- Additional payload: N
- Specification Document: Section [§ 3.3.1 Address](#) of this document.

APIKey

- Credential type identifier: api-key
- Description: A credential representing an authentication key to access an application's programmable interface.
- Additional payload: N
- Specification Document: Section [§ 3.3.2 APIKey](#) of this document.

BasicAuth

- Credential type identifier: basic-auth
- Description: A credential containing a username password for either [basic access authentication](#) or a form on a web page.
- Additional payload: N
- Specification Document: Section [§ 3.3.3 BasicAuth](#) of this document.

CreditCard

- Credential type identifier: credit-card
- Description: A credential representing credit card information for form filling.
- Additional payload: N
- Specification Document: Section [§ 3.3.4 CreditCard](#) of this document.

CustomFields

- Credential type identifier: custom-fields
- Description: A credential representing additional unstructured fields.
- Additional payload: N
- Specification Document: Section [§ 3.3.5 CustomFields Dictionary](#) of this document.

DriversLicense

- Credential type identifier: drivers-license

- Description: A textual representation of a physical driver's license.
- Additional payload: N
- Specification Document: Section [§ 3.3.6 DriversLicense](#) of this document.

File

- Credential type identifier: file
- Description: A credential representing a binary file by providing its metadata. The actual binary file is transmitted in an additional payload managed by the protocol.
- Additional payload: Y
- Specification Document: Section [§ 3.3.7 File](#) of this document.

GeneratedPassword

- Credential type identifier: generated-password
- Description: A credential representing a machine generated password
- Additional payload: N
- Specification Document: Section [§ 3.3.8 GeneratedPassword Dictionary](#) of this document.

IdentityDocument

- Credential type identifier: identity-document
- Description: A textual representation of a physical identity document. An example being the United States' social security number.
- Additional payload: N
- Specification Document: Section [§ 3.3.9 IdentityDocument](#) of this document.

ItemReference

- Credential type identifier: item-reference
- Description: A pointer from one [Item](#) to another.
- Additional payload: N
- Specification Document: Section [§ 3.3.10 ItemReference Dictionary](#) of this document.

Note

- Credential type identifier: note
- Description: A credential representing a multi-line note.
- Additional payload: N
- Specification Document: Section [§ 3.3.11 Note](#) of this document.

Passkey

- Credential type identifier: passkey
- Description: A credential representing a [\[webauthn-3\] private key](#).
- Additional payload: N
- Specification Document: Section [§ 3.3.12 Passkey Dictionary](#) of this document.

Passport

- Credential type identifier: passport
- Description: A textual representation of a physical passport.
- Additional payload: N
- Specification Document: Section [§ 3.3.13 Passport](#) of this document.

PersonName

- Credential type identifier: person-name
- Description: A decomposed textual representation of a person's name.
- Additional payload: N

- Specification Document: Section [§ 3.3.14 PersonName](#) of this document.

SSHKey

- Credential type identifier: ssh-key
- Description: A credential representing a secure shell key pair.
- Additional payload: N
- Specification Document: Section [§ 3.3.15 SSHKey](#) of this document.

TOTP

- Credential type identifier: totp
- Description: A credential representing a time based one time password seed.
- Additional payload: N
- Specification Document: Section [§ 3.3.16 TOTP](#) of this document.

WIFI

- Credential type identifier: wifi
- Description: A credential providing the necessary information to connect to a wireless network.
- Additional payload: N
- Specification Document: Section [§ 3.3.17 Wi-Fi Passphrase](#) of this document.

4.2. Extension Registry§

The request includes a registry for [Extension](#) entries. A "Credential Exchange Format Extensions" registry using the following template for new registration requests:

- Extension name identifier (JSON compatible string)
- Description
- Requires an additional payload (Y/N)
- Specification Document (URL)

The initial values for this table are the credentials defined in the document and are as follows:

Shared

- Extension name identifier: shared
- Description: An extension defining how an [entity](#) is shared across user [accounts](#).
- Additional payload: N
- Specification Document: Section [§ 3.5.1 Sharing an Entity \(Shared\)](#) of this document.

5. Security Considerations§

5.1. Orchestrating Party Requirements§

NOTE: [\[CXP\]](#) ensures that the credential exchange remains confidential between the [exporting provider](#) and [importing provider](#).

Implementations of credential exchange MAY involve an [orchestrating party](#). In such cases:

1. The [orchestrating party](#) MAY implement CXF without implementing [\[CXP\]](#).
2. If [\[CXP\]](#) is not implemented, the [orchestrating party](#) developer MUST:

- a. Document the security mechanisms of the credential exchange.
- b. Publish this documentation on a publicly accessible website.

5.1.1. Documentation Requirements§

The published documentation MUST:

1. Explicitly state that the [orchestrating party](#) only transmits the credential exchange data between the [exporting provider](#) and [importing provider](#).
2. Confirm that credential exchange data is not persisted or used for any purpose other than transmission.
3. Confirm that the credential exchange data received from the [exporting provider](#) is unaltered before being provided to the [importing provider](#).

NOTE: These requirements ensure transparency and maintain trust in the credential exchange process while accommodating a variety of implementation scenarios.

5.2. Validation of Files§

When importing the binary files associated with the [File](#) credential, the [importing provider](#) MUST validate that the provided [integrityHash](#) matches a hash of the received file. If the hashes do not match, the [importing provider](#) MUST reject the binary file exchange. At such a point, the [transfer protocol](#) SHOULD provide a method for the [importing provider](#) to request the file again

Appendix A: Example Payload§

This appendix provides an example of a credential exchange export in JSON which includes each credential type defined within this specification.

```
{
  "version": {
    "major": 1,
    "minor": 0
  },
  "exporterRpId": "exporter.example.com",
  "exporterDisplayName": "Exporter app",
  "timestamp": 1705228800,
  "accounts": [
    {
      "id": "DZSXp7iBQY-Fg-0ofakQtQ",
      "username": "jane_smith",
      "email": "jane.smith@example.com",
      "fullName": "Jane Smith",
      "items": [
        {
          "id": "90F-QjVDQo2Wp2xWPw6ZhA",
          "creationAt": 1705142400,
          "modifiedAt": 1705228800,
          "title": "GitHub Login",
          "subtitle": "Work GitHub account",
          "scope": {
            "urls": ["https://github.com"],
            "androidApps": []
          },
          "credentials": [
```

```

    {
      "type": "basic-auth",
      "username": {
        "id": "-eZX0Gw-Tz0sBFwt67N7ZA",
        "fieldType": "string",
        "value": "johndoe",
        "label": "Username field"
      },
      "password": {
        "id": "wgu3wTcXSYawrGMMtaAng",
        "fieldType": "concealed-string",
        "value": "securepassword123",
        "label": "Password field"
      }
    },
    {
      "type": "totp",
      "secret": "JBSWY3DPEHPK3PXP",
      "period": 30,
      "digits": 6,
      "issuer": "GitHub",
      "algorithm": "sha256",
      "username": "jane.smith@example.com"
    }
  ],
  "tags": ["development", "git", "work"]
},
{
  "id": "akKA3Y0jQRuK7sKplB0Y9w",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "WebAuthn.io",
  "subtitle": "johndoe",
  "credentials": [
    {
      "type": "passkey",
      "credentialId": "Y3JlZGVudGlhbElkRXhhbXBsZQ",
      "rpId": "webauthn.io",
      "username": "johndoe",
      "userDisplayName": "John Doe",
      "userHandle": "cnEzaNHwCYK3coWZjvoaV1Hj9gnI12mKe2dL2HZVFLY",
      "key": "MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgARu_0sCt2
0EpgVxb4Puq3Ga5VVLpuTY75ngvZlyq3X6hRANCAASmdk1xLsK0o0lhxIPp0d1ZuS0sT9nf6BztSelhqvLBW0f0L33l_b)
sr_STUHjCLn8l6gcRJwe70QvbQubZ1dY",
      "fido2Extensions": {
        "hmacCredentials": {
          "algorithm": "hmac-sha256",
          "credWithUV": "j3N5T9qLpWz2rYf4vS6lDn1KpQx8E0fRc2a7Bm5nUs
w",
          "credWithoutUV": "y2R8tL3eWf5qBz0sK4hHn9rVgX7pD1cQm6uTj2
aP8Fs"
        },
        "credBlob": "eyJ1c2VyTmFtZSI6ICJkb2huIERvZSI6ICJ1c2VySWQiOiAi
amRvZS0wMDEiLCAiZWlhaWwiOiAiamRvZUBleGFtcGxlLmNvbSJ9",
        "largeBlob": {
          "uncompressedSize": 129,
          "data": "HYxBCoUwDESvMgcQT-Hqr71ATIMtLkQkky-3twrDLIb3Zq8t
MEMKufZ7pBR08lNwdDtAEcaN3vXfsuJnVbGZrNhF82PYrL5ma1JTXCg0QkkLaAxETnmB0b53051GbYwQdsLYHw"
        }
      },
      "payments": true
    }
  ]
}

```

```

},
{
  "id": "iz0Q6JWoQ_CbDRboCPJ1Tg",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "Visa Credit Card",
  "subtitle": "Personal Visa card",
  "credentials": [
    {
      "type": "credit-card",
      "number": {
        "id": "MTIz",
        "fieldType": "concealed-string",
        "value": "4111111111111111",
        "label": "Card Number"
      },
      "fullName": {
        "fieldType": "string",
        "value": "John Doe",
        "label": "Cardholder Name"
      },
      "cardType": {
        "fieldType": "string",
        "value": "Visa",
        "label": "Card Type"
      },
      "verificationNumber": {
        "fieldType": "concealed-string",
        "value": "123",
        "label": "CVV"
      },
      "pin": {
        "fieldType": "concealed-string",
        "value": "0000",
        "label": "PIN"
      },
      "expiryDate": {
        "fieldType": "year-month",
        "value": "2027-08",
        "label": "Expiry Date"
      },
      "validFrom": {
        "fieldType": "year-month",
        "value": "2024-02",
        "label": "Valid From"
      }
    }
  ],
  "tags": ["finance", "credit card", "personal"]
},
{
  "id": "2cGy6PN0SQ2cW43NVxjGSg",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "Wifi",
  "subtitle": "Home Wifi",
  "credentials": [
    {
      "type": "wifi",
      "ssid": {
        "fieldType": "string",
        "value": "Home_Network",

```

```

        "label": "Wi-Fi SSID"
    },
    "networkSecurityType": {
        "fieldType": "wifi-network-security-type",
        "value": "wpa2-personal",
        "label": "Security Type"
    },
    "passphrase": {
        "fieldType": "concealed-string",
        "value": "mypassword123",
        "label": "Wi-Fi Password"
    },
    "hidden": {
        "fieldType": "boolean",
        "value": "false",
        "label": "Hidden Network"
    }
}
]
},
{
    "id": "s4TK1UNTRhG4j1DQawUz8g",
    "createdAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "Home alarm",
    "subtitle": "instructions",
    "credentials": [
        {
            "type": "note",
            "content": {
                "fieldType": "string",
                "value": "some instructions to enable/disable the alarm",
                "label": "alarm"
            }
        }
    ]
},
{
    "id": "BQzS9Ws3Rn0abLzFuy0u7Q",
    "createdAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "Driver License",
    "subtitle": "US",
    "credentials": [
        {
            "type": "drivers-license",
            "fullName": {
                "fieldType": "string",
                "value": "John Doe",
                "label": "Full Name"
            },
            "birthDate": {
                "fieldType": "date",
                "value": "1990-05-15",
                "label": "Date of Birth"
            },
            "issueDate": {
                "fieldType": "date",
                "value": "2020-06-01",
                "label": "Issue Date"
            },
            "expiryDate": {

```

```

        "fieldType": "date",
        "value": "2030-06-01",
        "label": "Expiry Date"
    },
    "issuingAuthority": {
        "fieldType": "string",
        "value": "Department of Motor Vehicles",
        "label": "Issuing Authority"
    },
    "territory": {
        "fieldType": "subdivision-code",
        "value": "US-CA",
        "label": "Territory"
    },
    "country": {
        "fieldType": "country-code",
        "value": "US",
        "label": "Country"
    },
    "licenseNumber": {
        "fieldType": "string",
        "value": "D12345678",
        "label": "License Number"
    },
    "licenseClass": {
        "fieldType": "string",
        "value": "C",
        "label": "License Class"
    }
}
]
},
{
    "id": "HHL63ybfQG6GBRHlyrvKfg",
    "creationAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "House Address",
    "subtitle": "US",
    "credentials": [
        {
            "type": "address",
            "streetAddress": {
                "fieldType": "string",
                "value": "123 Main Street",
                "label": "Street Address"
            },
            "postalCode": {
                "fieldType": "string",
                "value": "12345",
                "label": "Postal Code"
            },
            "city": {
                "fieldType": "string",
                "value": "Springfield",
                "label": "City"
            },
            "territory": {
                "fieldType": "subdivision-code",
                "value": "US-CA",
                "label": "State"
            },
            "country": {
                "fieldType": "country-code"

```

```

        "fieldType": "Country-code",
        "value": "US",
        "label": "Country"
    },
    "tel": {
        "fieldType": "string",
        "value": "+1-555-123-4567",
        "label": "Telephone"
    }
}
],
},
{
    "id": "Z4cFmc21Q5-vCVwd1wJx1g",
    "creationAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "SSH Key",
    "subtitle": "GitHub",
    "credentials": [
        {
            "type": "ssh-key",
            "keyType": "ssh-rsa",
            "privateKey": "MIIG_QIBADANBqkqhkiG9w0BAQEFAASCbucwggbjAgEAAoIBgQ
Cn4-QiJojZ9mgc9KYJIvDWGaz4qFhf0CButg6L8zEoHKwuiN-mqcEciCC0a9BNIJmm8NTTehZvrrglGG59zIbqYtDAHjVr
vtb49xPzIv-M651Yqj08lIbR9tEIHKCq7aH8GLDm8NgG9EzJGjLL7okQym4TH1MHL-s4mUyr_qb2unLZBDixAQsphU8iCl
tukWCiKmqg4CSj1Gh3WbBlZ-EX5eW0EXuAw4XsSbBTWV9CHRowVIpYqPvEYSpHsoCjEcd988p19hpiGknA0J4z7JfUlnGy
_1chb8GCTDT-2DCBRAppS Ig6TOBVS-PR6emAQ3eZzUW0-3_oRM4ip0ujltQy8uU6gvYIAqx5wXGMTThVpZcUgahKiSsVo_
b84iMe4DG3W8jz4qi6yyNv0VedEzPUZ1LXd1GJFoy9uKNUStE-1ksicAcLuZN6LuNsPHcPxFcZ0cmoNnVXEKAXInt-ys_
CDVasroZSAHZnDjUD4oNsLI3VI0nGxgXrkwSH0CAwEAAQKCAyAA2SDMf70BHW10GM90Qa1ZS4u-ktfQHhn31-FxbrhWgp
Dt8gYABVf6Y4dKN6rMtn7D9gVSALZCAn3Hx8aWAvCXHaspxe9YXiZDTh-Kd8EIXxBQn-TiDA5LH0dryABqmMp20vYKtR7(
3LIIXfFBSrBmWdunKzLwmKwZLWq0SWf6vVbwpXrYR9CyByodF6DjmZK3QB2qQ3jqL1LHWXL0VnyArY7HLvUvflLK4vMPqr
SH-FdHvhcEhwqMLWT44g-fhqWtCJNnjDgLK3FPbI8Pz9TF8dWJvOmp5Q6iSBua1e9x2LizVuNSqiFc7ZTLeoG4ndj7T2Bf
B0E1rNUDEN1aBo-UZmHJK7Lrzfw_B-ssi2WwIpfXya1l06HFod5_YQiXV1GunnyH1chCsbv0FtXvAHAS04HTKLJNbWhRF1(
qnKpAaHDPCVuwP3eq6Yf0oLbXrL3KFZ3jwWiWbpQXRVvpqzaJwZn3CN1yQgYS9j17a9wrPky-BoJxxjZ_oImWLEcgcEA0(
LwiHvmTYFTCC7PN938Agk9_NQs5PQ18MRn90JmyfSpYqf_gNp-Md7xUgtF_MTif7uelp2J7DYf6fj9EYf9g4EuW-SQgFP
fiJn1-zGFeTQq1ISvwsA4E8ZSt-GIumjZTg6YiL1_A79u4wm24swt7iqnVVi0PtPGOM34S1tAamjZzq2eZdMAF6pAfm
dinCMR1E1kNJYbxeqLiQcXuwBBnH000Jofn3AkzvjRUBB9udvniqYxH3PQcxPxAoHBAMxT5KwBhZhnJedYN87KkcpL7x
kpU8b-aXeZoNyKCeoc-wgIQexnSwmFk4HPkCNxvCwLbk0T1MHRtAKFna0ww230b-Vi6A9n0rozo9vtoJig114GB0gUqEm
Lh01P5AE8yzogE-ILHyp0BqXt8vGIzfzPdnCkN-GKl8g00MPRr4NAcL0-Rshc5nLs7BGB4SEi126Y6mSfp85m0--1QhWMz
zqJEHCWKVcZydcDeONP9js04EUnk33KtLJIWzZTQKBWAt0pBpGwmZRp35Lpx2gBitZhcVxrg0NBna02fNyAGPvZD8SLQL
dAiov_a23Uc_PDbWLL5Pp9gwzj-s5glrssV0XdE8aUscr1b5rARdNnL1_Tos6u8ZUZ3sNqGaZx7a8U4gyYboexWyo9EC1(
AdkGBm7-AkM4euFwC9N6xsa_t5zKK5d676hc0m-8SxivYCBkgkrqlfeGuZCQxU-mVsC0it6U-va8ojUjLGkZ800uCWbF4
l3-acU7vx9o8_gQKBwB7BrhU6MWrsc-cr_1KQaXum9mNyckomi82RFYvb8Yrilcg38FBy9XqNRKeBa9MLw1HZYpHbzsXs
7u4eQMLoDTLVNUC5L6dKAIlowoyTa24uH90wWTg_a8mTZMe1jhgrew-AJq27NV6z4PswR9GenDmyshDDud7rBsflZCQR
UfwReLV7BHU6UPBsXn4ASF4xnRyM6WvcKy9coKZcUqqgm3fLM_90izCCMJgfXHBRE-x7nBqst746qlEedSRrQKBwQCVYw
CHNLzXl0_NMkdJ-hp7_InHF6mz_3V058iCb19TLDVUC2dGXPXNYwTT9PcLefwV5HNBHcAfTzG4bdpQyNiDyV914HL7DFI
duoPnwBYjeFre54v0YjjnskjJ07myircdbdX_i-7LMUw5aZZXCC8a5BD_rdv6IKJWJG5QBXbe5fVf1Xw0jBTzlhIPIqhf
fSu-mFikp5BRwHGBqsKMju6inYmW6YADeY_Sv0QjDEB37RqGZxqyIx8V2ZYwU",
            "keyComment": "Work SSH Key",
            "creationDate": {
                "fieldType": "date",
                "value": "2023-01-01",
                "label": "Creation Date"
            },
            "expiryDate": {
                "fieldType": "date",
                "value": "2025-01-01",
                "label": "Expiry Date"
            },
            "keyGenerationSource": {
                "fieldType": "string",
                "value": "Generated using OpenSSH",
                "label": "Key Generation Source"
            }
        }
    ]
}
]
}
}

```

```

    }
  }
]
},
{
  "id": "EWM-4m3pSEi0ZBQbFVB92g",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "ID card",
  "subtitle": "US",
  "credentials": [
    {
      "type": "file",
      "id": "VGVzdEZpbGVJRA",
      "name": "example-file.txt",
      "decryptedSize": 21,
      "integrityHash": "crWLSY08T2f3yGaHQlDrMVvSvuu1YdXZewkjps422lQ"
    }
  ]
},
{
  "id": "U9TPhd80SsWKKUtX3HxVsA",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "ID card",
  "subtitle": "US",
  "credentials": [
    {
      "type": "identity-document",
      "issuingCountry": {
        "fieldType": "country-code",
        "value": "US",
        "label": "Issuing Country"
      },
      "documentNumber": {
        "fieldType": "string",
        "value": "123456789",
        "label": "Document Number"
      },
      "identificationNumber": {
        "fieldType": "string",
        "value": "ID123456789",
        "label": "Identification Number"
      },
      "nationality": {
        "fieldType": "string",
        "value": "American",
        "label": "Nationality"
      },
      "fullName": {
        "fieldType": "string",
        "value": "Jane Doe",
        "label": "Full Name"
      },
      "birthDate": {
        "fieldType": "date",
        "value": "1990-04-15",
        "label": "Birth Date"
      },
      "birthPlace": {
        "fieldType": "string",
        "value": "New York, USA",

```

```

        "label": "Birth Place"
    },
    "sex": {
        "fieldType": "string",
        "value": "F",
        "label": "Sex"
    },
    "issueDate": {
        "fieldType": "date",
        "value": "2020-01-01",
        "label": "Issue Date"
    },
    "expiryDate": {
        "fieldType": "date",
        "value": "2030-01-01",
        "label": "Expiry Date"
    },
    "issuingAuthority": {
        "fieldType": "string",
        "value": "Department of State",
        "label": "Issuing Authority"
    }
}
]
},
{
    "id": "K4BBLNWTS21ZqzTUn0H6Q",
    "creationAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "Passport",
    "subtitle": "US",
    "credentials": [
        {
            "type": "passport",
            "issuingCountry": {
                "fieldType": "country-code",
                "value": "US",
                "label": "Issuing Country"
            },
            "passportType": {
                "fieldType": "string",
                "value": "PP",
                "label": "Passport Type"
            },
            "passportNumber": {
                "fieldType": "string",
                "value": "A12345678",
                "label": "Passport Number"
            },
            "nationalIdentificationNumber": {
                "fieldType": "string",
                "value": "ID123456789",
                "label": "National Identification Number"
            },
            "nationality": {
                "fieldType": "string",
                "value": "American",
                "label": "Nationality"
            },
            "fullName": {
                "fieldType": "string",
                "value": "John Doe",
                "label": "Full Name"
            }
        }
    ]
}
]
}
}

```

```

        "label": "Full Name"
    },
    "birthDate": {
        "fieldType": "date",
        "value": "1990-01-01",
        "label": "Birth Date"
    },
    "birthPlace": {
        "fieldType": "string",
        "value": "Los Angeles, USA",
        "label": "Birth Place"
    },
    "sex": {
        "fieldType": "string",
        "value": "M",
        "label": "Sex"
    },
    "issueDate": {
        "fieldType": "date",
        "value": "2015-06-15",
        "label": "Issue Date"
    },
    "expiryDate": {
        "fieldType": "date",
        "value": "2025-06-15",
        "label": "Expiry Date"
    },
    "issuingAuthority": {
        "fieldType": "string",
        "value": "U.S. Department of State",
        "label": "Issuing Authority"
    }
}
]
},
{
    "id": "LmInpZjdRwKIKZFdbBz19g",
    "creationAt": 1705142400,
    "modifiedAt": 1705228800,
    "title": "John Doe",
    "subtitle": "personal name",
    "credentials": [
        {
            "type": "person-name",
            "title": {
                "fieldType": "string",
                "value": "Dr.",
                "label": "Title"
            },
            "given": {
                "fieldType": "string",
                "value": "John",
                "label": "Given Name"
            },
            "givenInformal": {
                "fieldType": "string",
                "value": "Johnny",
                "label": "Informal Given Name"
            },
            "given2": {
                "fieldType": "string",
                "value": "Michael",
                "label": "Second Given Name"
            }
        }
    ]
}
]
}

```

```

    },
    "surnamePrefix": {
      "fieldType": "string",
      "value": "van",
      "label": "Surname Prefix"
    },
    "surname": {
      "fieldType": "string",
      "value": "Doe",
      "label": "Surname"
    },
    "surname2": {
      "fieldType": "string",
      "value": "Smith",
      "label": "Second Surname"
    },
    "credentials": {
      "fieldType": "string",
      "value": "PhD",
      "label": "Credentials"
    },
    "generation": {
      "fieldType": "string",
      "value": "III",
      "label": "Generation"
    }
  }
},
{
  "id": "TMrjj3uIRtitVmIpiwXmyg",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "API key",
  "subtitle": "john_doe",
  "credentials": [
    {
      "type": "api-key",
      "key": {
        "fieldType": "concealed-string",
        "value": "AIzaSyAyRofL-VJHZofHc-q0SkqV0dhvgQoJADk",
        "label": "API Key"
      }
    },
    "username": {
      "fieldType": "string",
      "value": "john_doe",
      "label": "Username"
    },
    "keyType": {
      "fieldType": "string",
      "value": "Bearer",
      "label": "Key Type"
    },
    "url": {
      "fieldType": "string",
      "value": "https://api.example.com",
      "label": "API URL"
    },
    "validFrom": {
      "fieldType": "date",
      "value": "2025-01-01",
      "label": "Valid From"
    }
  ]
}

```

```

    },
    "expiryDate": {
      "fieldType": "date",
      "value": "2026-01-01",
      "label": "Expiry Date"
    }
  }
],
{
  "id": "QtvgfXSgS806ukLNZZKmlw",
  "creationAt": 1705142400,
  "modifiedAt": 1705228800,
  "title": "Generated Password",
  "subtitle": "john_doe",
  "credentials": [
    {
      "type": "generated-password",
      "password": "KozyS!cf#Nc9C799"
    }
  ]
},
"collections": [
  {
    "id": "0dimBl7dRRyPLGKGxEEm5Q",
    "creationAt": 1705228800,
    "modifiedAt": 1705228800,
    "title": "Work Accounts",
    "subtitle": "A collection of pro accounts for various services",
    "items": [
      {
        "item": "TMrjj3uIRtitVmIpiwXmyg",
        "account": "DZSXp7iBQY-Fg-0ofakQtQ"
      },
      {
        "item": "Z4cFmc21Q5-vCVwd1wJx1g",
        "account": "DZSXp7iBQY-Fg-0ofakQtQ"
      },
      {
        "item": "90F-QjVDQo2Wp2xWPw6ZhA",
        "account": "DZSXp7iBQY-Fg-0ofakQtQ"
      }
    ]
  }
]
}
]
}

```

The content of example-file.txt is:

this is a sample file

Conformance§

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in the normative parts of this document are to be

interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [\[RFC2119\]](#)

Examples in this specification are introduced with the words “for example” or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 1

This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with `class="note"`, like this:

Note, this is an informative note.

Index

Terms defined by this specification

["accessors"](#)

[accessors](#)

["account"](#)

[Account](#)

[account](#)

["accountId"](#)

[accountId](#)

["accounts"](#)

[accounts](#)

["address"](#)

[cddl-value for Address/type](#)

[cddl-value for CredentialType](#)

[Address](#)

[address](#)

[cddl-value for Address/type](#)

[cddl-value for CredentialType](#)

["algorithm"](#)

[cddl-key for Fido2HmacCredentials](#)

[cddl-key for TOTP](#)

[algorithm](#)

[cddl-key for Fido2HmacCredentials](#)

[cddl-key for TOTP](#)

[AndroidAppCertificateFingerprint](#)

[AndroidAppId](#)

["androidApps"](#)

[androidApps](#)

["api-key"](#)

[cddl-value for APIKey/type](#)
[cddl-value for CredentialType](#)

api-key

[cddl-value for APIKey/type](#)
[cddl-value for CredentialType](#)

[APIKey](#)

[b64url](#)

"basic-auth"

[cddl-value for BasicAuth/type](#)
[cddl-value for CredentialType](#)

basic-auth

[cddl-value for BasicAuth/type](#)
[cddl-value for CredentialType](#)

[BasicAuth](#)

"birthDate"

[cddl-key for DriversLicense](#)
[cddl-key for IdentityDocument](#)
[cddl-key for Passport](#)

birthDate

[cddl-key for DriversLicense](#)
[cddl-key for IdentityDocument](#)
[cddl-key for Passport](#)

"birthPlace"

[cddl-key for IdentityDocument](#)
[cddl-key for Passport](#)

birthPlace

[cddl-key for IdentityDocument](#)
[cddl-key for Passport](#)

"boolean"

[cddl-value for FieldType](#)
[cddl-value for WIFI/hidden](#)

boolean

[cddl-value for FieldType](#)
[cddl-value for WIFI/hidden](#)

["bundleId"](#)

[bundleId](#)

["cardType"](#)

[cardType](#)

["certificate"](#)

[certificate](#)

["city"](#)

[city](#)

[Collection](#)

["collections"](#)

[collections](#)

"concealed-string"

[cddl-value for APIKey/key](#)

[cddl-value for BasicAuth/password](#)
[cddl-value for CreditCard/number](#)
[cddl-value for CreditCard/pin](#)
[cddl-value for CreditCard/verificationNumber](#)
[cddl-value for FieldType](#)
[cddl-value for WIFI/passphrase](#)

concealed-string

[cddl-value for APIKey/key](#)
[cddl-value for BasicAuth/password](#)
[cddl-value for CreditCard/number](#)
[cddl-value for CreditCard/pin](#)
[cddl-value for CreditCard/verificationNumber](#)
[cddl-value for FieldType](#)
[cddl-value for WIFI/passphrase](#)

"content"

[content](#)

"country"

[cddl-key for Address](#)
[cddl-key for DriversLicense](#)

country

[cddl-key for Address](#)
[cddl-key for DriversLicense](#)

"country-code"

[cddl-value for Address/country](#)
[cddl-value for DriversLicense/country](#)
[cddl-value for FieldType](#)
[cddl-value for IdentityDocument/issuingCountry](#)
[cddl-value for Passport/issuingCountry](#)

country-code

[cddl-value for Address/country](#)
[cddl-value for DriversLicense/country](#)
[cddl-value for FieldType](#)
[cddl-value for IdentityDocument/issuingCountry](#)
[cddl-value for Passport/issuingCountry](#)

"create"

[create](#)

"creationAt"

[cddl-key for Collection](#)
[cddl-key for Item](#)

creationAt

[cddl-key for Collection](#)
[cddl-key for Item](#)

"creationDate"

[creationDate](#)

"credBlob"

[credBlob](#)

[\\$Credential](#)

[Credential](#)

["credentialId"](#)

[credentialId](#)

[Credential Owner](#)

[Credential Provider](#)

"credentials"

[cddl-key for Item](#)

[cddl-key for PersonName](#)

credentials

[cddl-key for Item](#)

[cddl-key for PersonName](#)

[CredentialScope](#)

[CredentialType](#)

"credit-card"

[cddl-value for CredentialType](#)

[cddl-value for CreditCard/type](#)

credit-card

[cddl-value for CredentialType](#)

[cddl-value for CreditCard/type](#)

[CreditCard](#)

["credWithoutUV"](#)

[credWithoutUV](#)

["credWithUV"](#)

[credWithUV](#)

"custom-fields"

[cddl-value for CredentialType](#)

[cddl-value for CustomFields/type](#)

custom-fields

[cddl-value for CredentialType](#)

[cddl-value for CustomFields/type](#)

[CustomFields](#)

["data"](#)

[data](#)

"date"

[cddl-value for APIKey/expiryDate](#)

[cddl-value for APIKey/validFrom](#)

[cddl-value for DriversLicense/birthDate](#)

[cddl-value for DriversLicense/expiryDate](#)

[cddl-value for DriversLicense/issueDate](#)

[cddl-value for FieldType](#)

[cddl-value for IdentityDocument/birthDate](#)

[cddl-value for IdentityDocument/expiryDate](#)

[cddl-value for IdentityDocument/issueDate](#)

[cddl-value for Passport/birthDate](#)

[cddl-value for Passport/expiryDate](#)

[cddl-value for Passport/issueDate](#)

[cddl-value for SSHKey/creationDate](#)

[cddl-value for SSHKey/expiryDate](#)

date

[cddl-value for APIKey/expiryDate](#)

[cddl-value for APIKey/validFrom](#)

[cddl-value for DriversLicense/birthDate](#)

[cddl-value for DriversLicense/expiryDate](#)

[cddl-value for DriversLicense/issueDate](#)

[cddl-value for FieldType](#)

[cddl-value for IdentityDocument/birthDate](#)

[cddl-value for IdentityDocument/expiryDate](#)

[cddl-value for IdentityDocument/issueDate](#)

[cddl-value for Passport/birthDate](#)

[cddl-value for Passport/expiryDate](#)

[cddl-value for Passport/issueDate](#)

[cddl-value for SSHKey/creationDate](#)

[cddl-value for SSHKey/expiryDate](#)

"decryptedSize"

[decryptedSize](#)

"delete"

[delete](#)

"digits"

[digits](#)

"documentNumber"

[documentNumber](#)

"drivers-license"

[cddl-value for CredentialType](#)

[cddl-value for DriversLicense/type](#)

drivers-license

[cddl-value for CredentialType](#)

[cddl-value for DriversLicense/type](#)

[DriversLicense](#)

[EditableField](#)

"email"

[cddl-key for Account](#)

[cddl-value for FieldType](#)

email

[cddl-key for Account](#)

[cddl-value for FieldType](#)

[Entity](#)

[\[EXP\]](#)

"expiryDate"

[cddl-key for APIKey](#)

[cddl-key for CreditCard](#)

[cddl-key for DriversLicense](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

[cddl-key for SSHKey](#)

expiryDate

[cddl-key for APIKey](#)

[cddl-key for CreditCard](#)
[cddl-key for DriversLicense](#)
[cddl-key for IdentityDocument](#)
[cddl-key for Passport](#)
[cddl-key for SSHKey](#)

["exporterDisplayName"](#)

[exporterDisplayName](#)

["exporterRpld"](#)

[exporterRpld](#)

[Exporting Provider](#)

[\\$Extension](#)

[Extension](#)

["extensions"](#)

[cddl-key for Account](#)
[cddl-key for Collection](#)
[cddl-key for CustomFields](#)
[cddl-key for EditableField](#)
[cddl-key for Item](#)

[extensions](#)

[cddl-key for Account](#)
[cddl-key for Collection](#)
[cddl-key for CustomFields](#)
[cddl-key for EditableField](#)
[cddl-key for Item](#)

[E](#)

["favorite"](#)

[favorite](#)

["fido2Extensions"](#)

[Fido2Extensions](#)

[fido2Extensions](#)

[Fido2HmacCredentialAlgorithm](#)

[Fido2HmacCredentials](#)

[Fido2LargeBlob](#)

["fields"](#)

[fields](#)

["fieldType"](#)

[FieldType](#)

[fieldType](#)

["file"](#)

[cddl-value for CredentialType](#)
[cddl-value for File/type](#)

[File](#)

[file](#)

[cddl-value for CredentialType](#)
[cddl-value for File/type](#)

"fingerprint"

fingerprint

"fullName"

cddl-key for Account

cddl-key for CreditCard

cddl-key for DriversLicense

cddl-key for IdentityDocument

cddl-key for Passport

fullName

cddl-key for Account

cddl-key for CreditCard

cddl-key for DriversLicense

cddl-key for IdentityDocument

cddl-key for Passport

"generated-password"

cddl-value for CredentialType

cddl-value for GeneratedPassword/type

generated-password

cddl-value for CredentialType

cddl-value for GeneratedPassword/type

GeneratedPassword

"generation"

generation

"given"

given

"given2"

given2

"givenInformal"

givenInformal

"group"

group

"hashAlg"

hashAlg

Header

"hidden"

hidden

"hmacCredentials"

hmacCredentials

"hmac-sha256"

hmac-sha256

"id"

cddl-key for Account

cddl-key for Collection

cddl-key for CustomFields

cddl-key for EditableField

[cddl-key for File](#)

[cddl-key for Item](#)

id

[cddl-key for Account](#)

[cddl-key for Collection](#)

[cddl-key for CustomFields](#)

[cddl-key for EditableField](#)

[cddl-key for File](#)

[cddl-key for Item](#)

["identificationNumber"](#)

[identificationNumber](#)

[Identifier](#)

["identity-document"](#)

[cddl-value for CredentialType](#)

[cddl-value for IdentityDocument/type](#)

identity-document

[cddl-value for CredentialType](#)

[cddl-value for IdentityDocument/type](#)

[IdentityDocument](#)

[\[IMP\]](#)

[Importing Provider](#)

["integrityHash"](#)

[integrityHash](#)

["issueDate"](#)

[cddl-key for DriversLicense](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

issueDate

[cddl-key for DriversLicense](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

["issuer"](#)

[issuer](#)

["issuingAuthority"](#)

[cddl-key for DriversLicense](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

issuingAuthority

[cddl-key for DriversLicense](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

["issuingCountry"](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

issuingCountry

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

["item"](#)

[Item](#)

[item](#)

"item-reference"

[cddl-value for CredentialType](#)

[cddl-value for ItemReference/type](#)

item-reference

[cddl-value for CredentialType](#)

[cddl-value for ItemReference/type](#)

[ItemReference](#)

"items"

[cddl-key for Account](#)

[cddl-key for Collection](#)

items

[cddl-key for Account](#)

[cddl-key for Collection](#)

"key"

[cddl-key for APIKey](#)

[cddl-key for Passkey](#)

key

[cddl-key for APIKey](#)

[cddl-key for Passkey](#)

["keyComment"](#)

[keyComment](#)

["keyGenerationSource"](#)

[keyGenerationSource](#)

"keyType"

[cddl-key for APIKey](#)

[cddl-key for SSHKey](#)

keyType

[cddl-key for APIKey](#)

[cddl-key for SSHKey](#)

"label"

[cddl-key for CustomFields](#)

[cddl-key for EditableField](#)

label

[cddl-key for CustomFields](#)

[cddl-key for EditableField](#)

["largeBlob"](#)

[largeBlob](#)

["licenseClass"](#)

[licenseClass](#)

["licenseNumber"](#)

[licenseNumber](#)

[LinkedItem](#)

["major"](#)

[major](#)

["manage"](#)

[manage](#)

["minor"](#)

[minor](#)

"modifiedAt"

[cddl-key for Collection](#)

[cddl-key for Item](#)

modifiedAt

[cddl-key for Collection](#)

[cddl-key for Item](#)

"name"

[cddl-key for AndroidAppId](#)

[cddl-key for Extension](#)

[cddl-key for File](#)

[cddl-key for Shared](#)

[cddl-key for SharingAccessor](#)

name

[cddl-key for AndroidAppId](#)

[cddl-key for Extension](#)

[cddl-key for File](#)

[cddl-key for Shared](#)

[cddl-key for SharingAccessor](#)

["nationalIdentificationNumber"](#)

[nationalIdentificationNumber](#)

"nationality"

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

nationality

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

["networkSecurityType"](#)

[networkSecurityType](#)

"note"

[cddl-value for CredentialType](#)

[cddl-value for Note/type](#)

[Note](#)

note

[cddl-value for CredentialType](#)

[cddl-value for Note/type](#)

"number"

[cddl-key for CreditCard](#)

[cddl-value for FieldType](#)

number

[cddl-key for CreditCard](#)

[cddl-value for FieldType](#)

[Orchestrating Party](#)

[OTPHashAlgorithm](#)

"passkey"

[cddl-value for CredentialType](#)

[cddl-value for Passkey/type](#)

Passkey

passkey

[cddl-value for CredentialType](#)

[cddl-value for Passkey/type](#)

"passphrase"

passphrase

"passport"

[cddl-value for CredentialType](#)

[cddl-value for Passport/type](#)

Passport

passport

[cddl-value for CredentialType](#)

[cddl-value for Passport/type](#)

"passportNumber"

passportNumber

"passportType"

passportType

"password"

[cddl-key for BasicAuth](#)

[cddl-key for GeneratedPassword](#)

password

[cddl-key for BasicAuth](#)

[cddl-key for GeneratedPassword](#)

"payments"

payments

"period"

period

"permissions"

permissions

"person-name"

[cddl-value for CredentialType](#)

[cddl-value for PersonName/type](#)

person-name

[cddl-value for CredentialType](#)

[cddl-value for PersonName/type](#)

PersonName

"pin"

pin

"postalCode"

postalCode

"privateKey"

privateKey

["read"](#)

[read](#)

["readSecret"](#)

[readSecret](#)

["reference"](#)

[reference](#)

["rpld"](#)

[rpld](#)

["scope"](#)

[scope](#)

["secret"](#)

[secret](#)

["sex"](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

[sex](#)

[cddl-key for IdentityDocument](#)

[cddl-key for Passport](#)

["sha1"](#)

[sha1](#)

["sha256"](#)

[cddl-value for AndroidAppCertificateFingerprint/hashAlg](#)

[cddl-value for OTPHashAlgorithm](#)

[sha256](#)

[cddl-value for AndroidAppCertificateFingerprint/hashAlg](#)

[cddl-value for OTPHashAlgorithm](#)

["sha512"](#)

[cddl-value for AndroidAppCertificateFingerprint/hashAlg](#)

[cddl-value for OTPHashAlgorithm](#)

[sha512](#)

[cddl-value for AndroidAppCertificateFingerprint/hashAlg](#)

[cddl-value for OTPHashAlgorithm](#)

["share"](#)

[share](#)

["shared"](#)

[Shared](#)

[shared](#)

[SharingAccessor](#)

[SharingAccessorPermission](#)

[SharingAccessorType](#)

["ssh-key"](#)

[cddl-value for CredentialType](#)

[cddl-value for SSHKey/type](#)

[ssh-key](#)

[cddl-value for CredentialType](#)

[cddl-value for SSHKey/type](#)

SSHKey

["ssid"](#)

[ssid](#)

["streetAddress"](#)

[streetAddress](#)

["string"](#)

[cddl-value for APIKey/keyType](#)

[cddl-value for APIKey/url](#)

[cddl-value for APIKey/username](#)

[cddl-value for Address/city](#)

[cddl-value for Address/postalCode](#)

[cddl-value for Address/streetAddress](#)

[cddl-value for Address/tel](#)

[cddl-value for BasicAuth/username](#)

[cddl-value for CreditCard/cardType](#)

[cddl-value for CreditCard/fullName](#)

[cddl-value for DriversLicense/fullName](#)

[cddl-value for DriversLicense/issuingAuthority](#)

[cddl-value for DriversLicense/licenseClass](#)

[cddl-value for DriversLicense/licenseNumber](#)

[cddl-value for FieldType](#)

[cddl-value for IdentityDocument/birthPlace](#)

[cddl-value for IdentityDocument/documentNumber](#)

[cddl-value for IdentityDocument/fullName](#)

[cddl-value for IdentityDocument/identificationNumber](#)

[cddl-value for IdentityDocument/issuingAuthority](#)

[cddl-value for IdentityDocument/nationality](#)

[cddl-value for IdentityDocument/sex](#)

[cddl-value for Note/content](#)

[cddl-value for Passport/birthPlace](#)

[cddl-value for Passport/fullName](#)

[cddl-value for Passport/issuingAuthority](#)

[cddl-value for Passport/nationalIdentificationNumber](#)

[cddl-value for Passport/nationality](#)

[cddl-value for Passport/passportNumber](#)

[cddl-value for Passport/passportType](#)

[cddl-value for Passport/sex](#)

[cddl-value for PersonName/credentials](#)

[cddl-value for PersonName/generation](#)

[cddl-value for PersonName/given](#)

[cddl-value for PersonName/given2](#)

[cddl-value for PersonName/givenInformal](#)

[cddl-value for PersonName/surname](#)

[cddl-value for PersonName/surname2](#)

[cddl-value for PersonName/surnamePrefix](#)

[cddl-value for PersonName/title](#)

[cddl-value for SSHKey/keyGenerationSource](#)

[cddl-value for WIFI/networkSecurityType](#)

[cddl-value for WIFI/ssid](#)

string

[cddl-value for APIKey/keyType](#)
[cddl-value for APIKey/url](#)
[cddl-value for APIKey/username](#)
[cddl-value for Address/city](#)
[cddl-value for Address/postalCode](#)
[cddl-value for Address/streetAddress](#)
[cddl-value for Address/tel](#)
[cddl-value for BasicAuth/username](#)
[cddl-value for CreditCard/cardType](#)
[cddl-value for CreditCard/fullName](#)
[cddl-value for DriversLicense/fullName](#)
[cddl-value for DriversLicense/issuingAuthority](#)
[cddl-value for DriversLicense/licenseClass](#)
[cddl-value for DriversLicense/licenseNumber](#)
[cddl-value for FieldType](#)
[cddl-value for IdentityDocument/birthPlace](#)
[cddl-value for IdentityDocument/documentNumber](#)
[cddl-value for IdentityDocument/fullName](#)
[cddl-value for IdentityDocument/identificationNumber](#)
[cddl-value for IdentityDocument/issuingAuthority](#)
[cddl-value for IdentityDocument/nationality](#)
[cddl-value for IdentityDocument/sex](#)
[cddl-value for Note/content](#)
[cddl-value for Passport/birthPlace](#)
[cddl-value for Passport/fullName](#)
[cddl-value for Passport/issuingAuthority](#)
[cddl-value for Passport/nationalIdentificationNumber](#)
[cddl-value for Passport/nationality](#)
[cddl-value for Passport/passportNumber](#)
[cddl-value for Passport/passportType](#)
[cddl-value for Passport/sex](#)
[cddl-value for PersonName/credentials](#)
[cddl-value for PersonName/generation](#)
[cddl-value for PersonName/given](#)
[cddl-value for PersonName/given2](#)
[cddl-value for PersonName/givenInformal](#)
[cddl-value for PersonName/surname](#)
[cddl-value for PersonName/surname2](#)
[cddl-value for PersonName/surnamePrefix](#)
[cddl-value for PersonName/title](#)
[cddl-value for SSHKey/keyGenerationSource](#)
[cddl-value for WIFI/networkSecurityType](#)
[cddl-value for WIFI/ssid](#)

"subCollections"

subCollections

"subdivision-code"

[cddl-value for Address/territory](#)
[cddl-value for DriversLicense/territory](#)
[cddl-value for FieldType](#)

subdivision-code

[cddl-value for Address/territory](#)

[cddl-value for DriversLicense/territory](#)

[cddl-value for FieldType](#)

"subtitle"

[cddl-key for Collection](#)

[cddl-key for Item](#)

subtitle

[cddl-key for Collection](#)

[cddl-key for Item](#)

"surname"

[surname](#)

"surname2"

[surname2](#)

"surnamePrefix"

[surnamePrefix](#)

"tags"

[tags](#)

"tel"

[tel](#)

"territory"

[cddl-key for Address](#)

[cddl-key for DriversLicense](#)

territory

[cddl-key for Address](#)

[cddl-key for DriversLicense](#)

"timestamp"

[timestamp](#)

"title"

[cddl-key for Collection](#)

[cddl-key for Item](#)

[cddl-key for PersonName](#)

title

[cddl-key for Collection](#)

[cddl-key for Item](#)

[cddl-key for PersonName](#)

"totp"

[cddl-value for CredentialType](#)

[cddl-value for TOTP/type](#)

[TOTP](#)

totp

[cddl-value for CredentialType](#)

[cddl-value for TOTP/type](#)

[Transfer Protocol](#)

"type"

[cddl-key for APIKey](#)

[cddl-key for Address](#)

[cddl-key for BasicAuth](#)

[cddl-key for Credential](#)
[cddl-key for CreditCard](#)
[cddl-key for CustomFields](#)
[cddl-key for DriversLicense](#)
[cddl-key for File](#)
[cddl-key for GeneratedPassword](#)
[cddl-key for IdentityDocument](#)
[cddl-key for ItemReference](#)
[cddl-key for Note](#)
[cddl-key for Passkey](#)
[cddl-key for Passport](#)
[cddl-key for PersonName](#)
[cddl-key for SSHKey](#)
[cddl-key for SharingAccessor](#)
[cddl-key for TOTP](#)
[cddl-key for WIFI](#)

type

[cddl-key for APIKey](#)
[cddl-key for Address](#)
[cddl-key for BasicAuth](#)
[cddl-key for Credential](#)
[cddl-key for CreditCard](#)
[cddl-key for CustomFields](#)
[cddl-key for DriversLicense](#)
[cddl-key for File](#)
[cddl-key for GeneratedPassword](#)
[cddl-key for IdentityDocument](#)
[cddl-key for ItemReference](#)
[cddl-key for Note](#)
[cddl-key for Passkey](#)
[cddl-key for Passport](#)
[cddl-key for PersonName](#)
[cddl-key for SSHKey](#)
[cddl-key for SharingAccessor](#)
[cddl-key for TOTP](#)
[cddl-key for WIFI](#)

["uncompressedSize"](#)

[uncompressedSize](#)

[Unique Identifier](#)

["unsecured"](#)

[unsecured](#)

["update"](#)

[update](#)

["url"](#)

[url](#)

["urls"](#)

[urls](#)

["user"](#)

[user](#)

["userDisplayName"](#)

[userDisplayName](#)

["userHandle"](#)

[userHandle](#)

"username"

[cddl-key for APIKey](#)

[cddl-key for Account](#)

[cddl-key for BasicAuth](#)

[cddl-key for Passkey](#)

[cddl-key for TOTP](#)

username

[cddl-key for APIKey](#)

[cddl-key for Account](#)

[cddl-key for BasicAuth](#)

[cddl-key for Passkey](#)

[cddl-key for TOTP](#)

"validFrom"

[cddl-key for APIKey](#)

[cddl-key for CreditCard](#)

validFrom

[cddl-key for APIKey](#)

[cddl-key for CreditCard](#)

["value"](#)

[value](#)

["verificationNumber"](#)

[verificationNumber](#)

["version"](#)

[Version](#)

[version](#)

["wep"](#)

[wep](#)

"wifi"

[cddl-value for CredentialType](#)

[cddl-value for WIFI/type](#)

[WIFI](#)

wifi

[cddl-value for CredentialType](#)

[cddl-value for WIFI/type](#)

"wifi-network-security-type"

[cddl-value for FieldType](#)

[cddl-value for WIFI/networkSecurityType](#)

wifi-network-security-type

[cddl-value for FieldType](#)

[cddl-value for WIFI/networkSecurityType](#)

[WIFINetworkSecurityType](#)

["wpa2-personal"](#)

[wpa2-personal](#)

["wpa3-personal"](#)

[wpa3-personal](#)

["wpa-personal"](#)

[wpa-personal](#)

"year-month"

[cddl-value for CreditCard/expiryDate](#)

[cddl-value for CreditCard/validFrom](#)

[cddl-value for FieldType](#)

year-month

[cddl-value for CreditCard/expiryDate](#)

[cddl-value for CreditCard/validFrom](#)

[cddl-value for FieldType](#)

Terms defined by reference

[CTAP2] defines the following terms:

credential blob extension

CTAP2

hmac-secret extension

reading large blobs

[CXP] defines the following terms:

CXP

[RFC4226] defines the following terms:

shared secret

[RFC4648] defines the following terms:

Base32

Base64url

[RFC8610] defines the following terms:

b64url

bool

false

socket

tstr

uint

uri

[WebAuthn] defines the following terms:

PublicKeyCredential

PublicKeyCredentialUserEntity

Credential ID

displayName

human-palatable

id

large blob extension

name

prf extension

private key

public key

rawld
registration
Relying Party
Relying Party Identifier
RP ID
User Account
user handle
WebAuthn

References§

Normative References§

[CXP]

N. Steele. *Credential Exchange Protocol*. Oct 10, 2024. FIDO Alliance Working Draft. URL: <https://fidoalliance.org/specs/cx/cxp-v1.0-wd-20241003.html>

[FIDO-V2.1]

Client to Authenticator Protocol (CTAP). Editor's Draft. URL: <https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-errata-20220621.html>

[IANA-SSHPP-PK-ALG-NAMES]

Secure Shell (SSH) Protocol Parameters Public Key Algorithm Names. URL: <https://www.iana.org/assignments/ssh-parameters/ssh-parameters.xhtml#ssh-parameters-19>

[ICAO9303-4]

Doc 9303 - Part 4- Machine Readable Travel Documents 2021. URL: https://www.icao.int/publications/Documents/9303_p4_cons_en.pdf

[ISO3166-1]

Codes for the representation of names of countries and their subdivisions — Part 1: Country code August 2020. Published. URL: <https://www.iso.org/standard/72482.html>

[ISO3166-2]

ISO 3166: Codes for the representation of names of countries and their subdivisions – Part 2: Country subdivision code. August 2020. Published. URL: <https://www.iso.org/standard/72483.html>

[ITU-X690-2008]

X.690: Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). (T-REC-X.690-200811). November 2008. URL: <https://www.itu.int/rec/T-REC-X.690-200811-S>

[JSON]

ECMA-404 The JSON Data Interchange Format. October 2013. URL: <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels* March 1997. Best Current Practice. URL: <https://tools.ietf.org/html/rfc2119>

[RFC3174]

D. Eastlake 3rd; P. Jones. *US Secure Hash Algorithm 1 (SHA1)*. September 2001. Informational. URL: <https://www.rfc-editor.org/rfc/rfc3174>

[RFC3339]

G. Klyne; C. Newman. *Date and Time on the Internet: Timestamps*. July 2002. Proposed Standard. URL: <https://www.rfc-editor.org/rfc/rfc3339>

[RFC3986]

T. Berners-Lee; R. Fielding; L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax* January 2005. Internet Standard. URL: <https://www.rfc-editor.org/rfc/rfc3986>

[RFC4226]

D. M'Raihi; et al. *HOTP: An HMAC-Based One-Time Password Algorithm*. December 2005. Informational. URL: <https://www.rfc-editor.org/rfc/rfc4226>

[RFC4648]

S. Josefsson. *The Base16, Base32, and Base64 Data Encodings (RFC 4648)*. October 2006. URL: <http://www.ietf.org/rfc/rfc4648.txt>

[RFC5322]

P. Resnick, Ed.. *Internet Message Format*. October 2008. Draft Standard. URL: <https://www.rfc-editor.org/rfc/rfc5322>

[RFC5958]

S. Turner. *Asymmetric Key Packages*. August 2010. Proposed Standard. URL: <https://www.rfc-editor.org/rfc/rfc5958>

[RFC6234]

D. Eastlake 3rd; T. Hansen. *US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF) (RFC 6234)*. May 2011. URL: <http://www.ietf.org/rfc/rfc6234.txt>

[RFC7617]

J. Reschke. *The 'Basic' HTTP Authentication Scheme*. September 2015. Proposed Standard. URL: <https://httpwg.org/specs/rfc7617.html>

[RFC8610]

H. Birkholz; C. Viganò; C. Bormann. *Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures*. June 2019. Proposed Standard. URL: <https://tools.ietf.org/html/rfc8610>

[SECURE-PAYMENT-CONFIRMATION]

Rouslan Solomakhin (Google); Stephen McGruer (Google). *Secure Payment Confirmation*. 31 August 2021. TR. URL: <https://www.w3.org/TR/secure-payment-confirmation/>

[URL]

Anne van Kesteren. *URL Standard*. Living Standard. URL: <https://url.spec.whatwg.org/>

[WebAuthn]

Dirk Balfanz (Google); et al. *Web Authentication: An API for accessing Public Key Credentials Level 2* 8 April 2021. TR. URL: <https://www.w3.org/TR/webauthn-2/>

[WEBAUTHN-3]

Michael Jones; et al. *Web Authentication: An API for accessing Public Key Credentials - Level 3* URL: <https://w3c.github.io/webauthn/>

Informative References§

[ISO-18013]

Information technology — Personal identification — ISO-compliant driving licence — Part 1: Physical characteristics and basic data set. 2018-08. International Standard confirmed. URL: <https://www.iso.org/standard/63798.html>

[ISO-8601]

Date and time — Representations for information interchange 2019-02. International Standard to be revised. URL: <https://www.iso.org/standard/70907.html>

[PCI-DSS-4.0.1]

Payment Card Industry Data Security Standard, Requirements and Testing Procedures June 2024. URL: https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4_0_1.pdf

[RFC8126]

M. Cotton; B. Leiba; T. Narten. *Guidelines for Writing an IANA Considerations Section in RFCs* June 2017. Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc8126>

[UTS-35-PART-8]

Mark Davis; et al. *Unicode Locale Data Markup Language (LDML) Part 8: Person Names* October 21, 2024. URL: <https://unicode.org/reports/tr35/tr35-personNames.html>

[WPA3]

CDDL Index§

```
b64url = tstr

Header = {
    version: Version,
    exporterRpId: tstr,
    exporterDisplayName: tstr,
    timestamp: uint .size 8,
    accounts: [ * Account ],
}

Version = {
    major: uint .size 1,
    minor: uint .size 1,
}

Account = {
    id: b64url,
    username: tstr,
    email: tstr,
    ? fullName: tstr,
    collections: [ * Collection ],
    items: [ * Item ],
    ? extensions: [ * Extension ] .default [],
}

Collection = {
    id: b64url,
    ? createdAt: uint .size 8,
    ? modifiedAt: uint .size 8,
    title: tstr,
    ? subtitle: tstr,
    items: [ * LinkedItem ],
    ? subCollections: [ * Collection ] .default [],
    ? extensions: [ * Extension ] .default [],
}

LinkedItem = {
    item: b64url,
    ? account: b64url,
}

Item = {
    id: b64url,
    ? createdAt: uint .size 8,
    ? modifiedAt: uint .size 8,
    title: tstr,
    ? subtitle: tstr,
    ? favorite: bool .default false,
    ? scope: CredentialScope,
    credentials: [ * Credential ],
    ? tags: [ * tstr ] .default [],
    ? extensions: [ * Extension ] .default [],
}

CredentialScope = {
```

```

    urls: [ * uri ],
    androidApps: [ * AndroidAppId ],
}

AndroidAppId = {
    bundleId: tstr,
    ? certificate: AndroidAppCertificateFingerprint,
    ? name: tstr
}

AndroidAppCertificateFingerprint = {
    fingerprint: b64url,
    hashAlg: "sha256" / "sha512" / tstr,
}

Credential = $Credential .within {
    type: CredentialType / tstr
}

$Credential /= Address
Address = {
    type: "address",
    ? streetAddress: EditableField<"string">,
    ? postalCode: EditableField<"string">,
    ? city: EditableField<"string">,
    ? territory: EditableField<"subdivision-code">,
    ? country: EditableField<"country-code">,
    ? tel: EditableField<"string">,
}

$Credential /= APIKey
APIKey = {
    type: "api-key",
    ? key: EditableField<"concealed-string">,
    ? username: EditableField<"string">,
    ? keyType: EditableField<"string">,
    ? url: EditableField<"string">,
    ? validFrom: EditableField<"date">,
    ? expiryDate: EditableField<"date">,
}

$Credential /= BasicAuth
BasicAuth = {
    type: "basic-auth",
    ? username: EditableField<"string">,
    ? password: EditableField<"concealed-string">
}

$Credential /= CreditCard
CreditCard = {
    type: "credit-card",
    ? number: EditableField<"concealed-string">,
    ? fullName: EditableField<"string">,
    ? cardType: EditableField<"string">,
    ? verificationNumber: EditableField<"concealed-string">,
    ? pin: EditableField<"concealed-string">,
    ? expiryDate: EditableField<"year-month">,
    ? validFrom: EditableField<"year-month">,
}

$Credential /= CustomFields
CustomFields = {
    type: "custom-fields"
}

```

```

    type: Custom-Fields,
    ? id: b64url,
    ? label: tstr,
    fields: [ * EditableField<FieldType> ],
    ? extensions: [ + Extension ] .default [],
}

$Credential /= DriversLicense
DriversLicense = {
    type: "drivers-license",
    ? fullName: EditableField<"string">,
    ? birthDate: EditableField<"date">,
    ? issueDate: EditableField<"date">,
    ? expiryDate: EditableField<"date">,
    ? issuingAuthority: EditableField<"string">,
    ? territory: EditableField<"subdivision-code">,
    ? country: EditableField<"country-code">,
    ? licenseNumber: EditableField<"string">,
    ? licenseClass: EditableField<"string">,
}

$Credential /= File
File = {
    type: "file",
    id: b64url,
    name: tstr,
    decryptedSize: uint,
    integrityHash: b64url
}

$Credential /= GeneratedPassword
GeneratedPassword = {
    type: "generated-password",
    password: tstr,
}

$Credential /= IdentityDocument
IdentityDocument = {
    type: "identity-document",
    ? issuingCountry: EditableField<"country-code">,
    ? documentNumber: EditableField<"string">,
    ? identificationNumber: EditableField<"string">,
    ? nationality: EditableField<"string">,
    ? fullName: EditableField<"string">,
    ? birthDate: EditableField<"date">,
    ? birthPlace: EditableField<"string">,
    ? sex: EditableField<"string">,
    ? issueDate: EditableField<"date">,
    ? expiryDate: EditableField<"date">,
    ? issuingAuthority: EditableField<"string">,
}

$Credential /= ItemReference
ItemReference = {
    type: "item-reference",
    reference: LinkedItem,
}

$Credential /= Note
Note = {
    type: "note",
    content: EditableField<"string">,
}
}

```

```

$Credential /= Passkey
Passkey = {
    type: "passkey",
    credentialId: b64url,
    rpId: tstr,
    username: tstr,
    userDisplayName: tstr,
    userHandle: b64url,
    key: b64url,
    ? fido2Extensions: Fido2Extensions,
}

Fido2Extensions = {
    ? hmacCredentials: Fido2HmacCredentials,
    ? credBlob: b64url,
    ? largeBlob: Fido2LargeBlob,
    ? payments: bool,
}

Fido2HmacCredentials = {
    algorithm: Fido2HmacCredentialAlgorithm / tstr,
    credWithUV: b64url,
    credWithoutUV: b64url,
}

Fido2HmacCredentialAlgorithm =
    "hmac-sha256"

Fido2LargeBlob = {
    uncompressedSize: uint,
    data: b64url,
}

$Credential /= Passport
Passport = {
    type: "passport",
    ? issuingCountry: EditableField<"country-code">,
    ? passportType: EditableField<"string">,
    ? passportNumber: EditableField<"string">,
    ? nationalIdentificationNumber: EditableField<"string">,
    ? nationality: EditableField<"string">,
    ? fullName: EditableField<"string">,
    ? birthDate: EditableField<"date">,
    ? birthPlace: EditableField<"string">,
    ? sex: EditableField<"string">,
    ? issueDate: EditableField<"date">,
    ? expiryDate: EditableField<"date">,
    ? issuingAuthority: EditableField<"string">,
}

$Credential /= PersonName
PersonName = {
    type: "person-name",
    ? title: EditableField<"string">,
    ? given: EditableField<"string">,
    ? givenInformal: EditableField<"string">,
    ? given2: EditableField<"string">,
    ? surnamePrefix: EditableField<"string">,
    ? surname: EditableField<"string">,
    ? surname2: EditableField<"string">,
    ? credentials: EditableField<"string">,
}

```

```

    ? generation: EditableField<"string">,
}

$Credential /= SSHKey
SSHKey = {
    type: "ssh-key",
    keyType: tstr,
    privateKey: b64url,
    ? keyComment: tstr,
    ? creationDate: EditableField<"date">,
    ? expiryDate: EditableField<"date">,
    ? keyGenerationSource: EditableField<"string">,
}

$Credential /= TOTP
TOTP = {
    type: "totp",
    secret: tstr,
    period: uint .size 2,
    digits: uint .size 2,
    ? username: tstr,
    algorithm: OTPHashAlgorithm / tstr,
    ? issuer: tstr,
}

OTPHashAlgorithm =
    "sha1" /
    "sha256" /
    "sha512"

$Credential /= WIFI
WIFI = {
    type: "wifi",
    ? ssid: EditableField<"string">,
    ? networkSecurityType: EditableField<"wifi-network-security-type"> /
        EditableField<"string">,
    ? passphrase: EditableField<"concealed-string">,
    ? hidden: EditableField<"boolean">
}

WIFINetworkSecurityType = {
    "unsecured" /
    "wpa-personal" /
    "wpa2-personal" /
    "wpa3-personal" /
    "wep"
}

CredentialType =
    "address" /
    "api-key" /
    "basic-auth" /
    "credit-card" /
    "custom-fields" /
    "drivers-license" /
    "file" /
    "generated-password" /
    "identity-document" /
    "item-reference" /
    "note" /
    "passkey" /
    "passport" /

```

```

    "person-name" /
    "ssh-key" /
    "totp" /
    "wifi"

EditableField<F> = {
  ? id: b64url,
  fieldType: F / tstr,
  value: tstr,
  ? label: tstr,
  ? extensions: [ + Extension ] .default [],
}

FieldType =
  "string" /
  "concealed-string" /
  "email" /
  "number" /
  "boolean" /
  "date" /
  "year-month" /
  "wifi-network-security-type" /
  "country-code" /
  "subdivision-code"

Extension = $Extension .within {
  name: tstr
  ; Should there be an included schema? or use a URI to define the schema?
}

$Extension /= Shared
Shared = {
  name: "shared",
  accessors: [ * SharingAccessor ],
}

SharingAccessor = {
  type: SharingAccessorType / tstr,
  accountId: b64url,
  name: tstr,
  permissions: [ * SharingAccessorPermission / tstr ],
}

SharingAccessorType =
  "user" /
  "group"

SharingAccessorPermission =
  "read" /
  "readSecret" /
  "update" /
  "create" /
  "delete" /
  "share" /
  "manage"

```

