|  |
| --- |
| 戦略的イノベーション創造プログラム（ＳＩＰ）第２期／ビッグデータ・ＡＩを活用したサイバー空間基盤技術／分野間データ連携基盤技術／分野・組織を超えたデータ活用とサービス提供を実現する基盤の研究 |
| トラスト基盤連携 |
| トラスト共通API（Ver2.0）技術仕様書 |

セコムトラストシステムズ株式会社

2023年3月



変更履歴

| # | 年月日 | 変更者 | 変更内容 |
| --- | --- | --- | --- |
| 1 | 2023年3月17日 |  | ・初版作成 |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

目次

[1. トラスト共通API仕様（要求仕様）策定 5](#_Toc130881206)

[1.1 全体スキーム 5](#_Toc130881207)

[1.2 Cloud Signature Consortium Standardのスコープ 5](#_Toc130881208)

[1.3 ETSI ESIにおける概念 8](#_Toc130881209)

[2. トラスト共通API技術仕様 10](#_Toc130881210)

[2.1複数認証局との相互運用モデル 10](#_Toc130881211)

[2.1.1 リモート署名における相互運用モデル 10](#_Toc130881212)

[2.1.2 CA initiative model 12](#_Toc130881213)

[2.1.3 Signer initiative model 12](#_Toc130881214)

[2.1.4 Signer and CA initiative model (Hybrid) 13](#_Toc130881215)

[2.2 複数認証局との相互運用モデル 14](#_Toc130881216)

[2.2.1　管理API（manages/generateKey） 14](#_Toc130881217)

[2.2.2　管理API（manages/linkEid） 14](#_Toc130881218)

[2.2.3　管理API（manages/linkCert） 14](#_Toc130881219)

[2.2.4　管理API（manages/deleteKey） 15](#_Toc130881220)

[2.3 複数タイムスタンプとの相互運用 15](#_Toc130881221)

[2.4 アカウント定義 16](#_Toc130881222)

[2.5 署名検証/証明書検証 17](#_Toc130881223)

[2.5.1 検証API の仕様策定方針 17](#_Toc130881224)

[2.5.2 検証API概要 18](#_Toc130881225)

[2.5.2.1 /validates/validSign 19](#_Toc130881226)

[2.5.2.2 /validates/validCert 20](#_Toc130881227)

[2.5.2.3 /info拡張 22](#_Toc130881228)

[2.6 eSeal利用における処理仕様 23](#_Toc130881229)

[2.6.1 複数署名（シール）：一括（1回のAPIで複数署名計算） 24](#_Toc130881230)

[2.6.2 複数署名（シール）：連続（複数の署名APIの連続呼び出し） 25](#_Toc130881231)

[3. トラスト共通API定義 26](#_Toc130881232)

[3.1 共通API　入出力パラメータ 26](#_Toc130881233)

[3.1.1 info 28](#_Toc130881234)

[3.1.2 manages/generatekey 30](#_Toc130881235)

[3.1.3 manages/linkEid 31](#_Toc130881236)

[3.1.4 manages/linkCert 32](#_Toc130881237)

[3.1.5 manages/deleteKey 33](#_Toc130881238)

[3.1.6 credentials/list 34](#_Toc130881239)

[3.1.7 credentials/info 35](#_Toc130881240)

[3.1.8 credentials/authorize 37](#_Toc130881241)

[3.1.9 credentials/extendTransaction 38](#_Toc130881242)

[3.1.10 credentials/sendOTP 39](#_Toc130881243)

[3.1.11 timestamps/list 40](#_Toc130881244)

[3.1.12 timestamps/info 41](#_Toc130881245)

[3.1.13 signatures/signHash 42](#_Toc130881246)

[3.1.14 signatures/timestamp 43](#_Toc130881247)

[3.1.15 validates/validSign 44](#_Toc130881248)

[3.1.16 validates/validCert 46](#_Toc130881249)

# トラスト共通API仕様（要求仕様）策定

1. ここでは、利用者認証、リモート署名、ｅシール、タイムスタンプの各トラストサービスについて必要とされる共通API要求仕様を策定する。

## 1.1 全体スキーム

共通APIは、RSSPを利用する外部サービスとのインターフェースとし、署名鍵を預かる環境設定および電子証明書発行管理のための「鍵管理共通API」と、実際にユーザ（利用者）が遠隔で署名をするための「RSサービス共通API」を用意した。

本仕様では、ユーザの利便性を確保すべく、複数のCAおよび複数のTSAを利用できることを想定した。

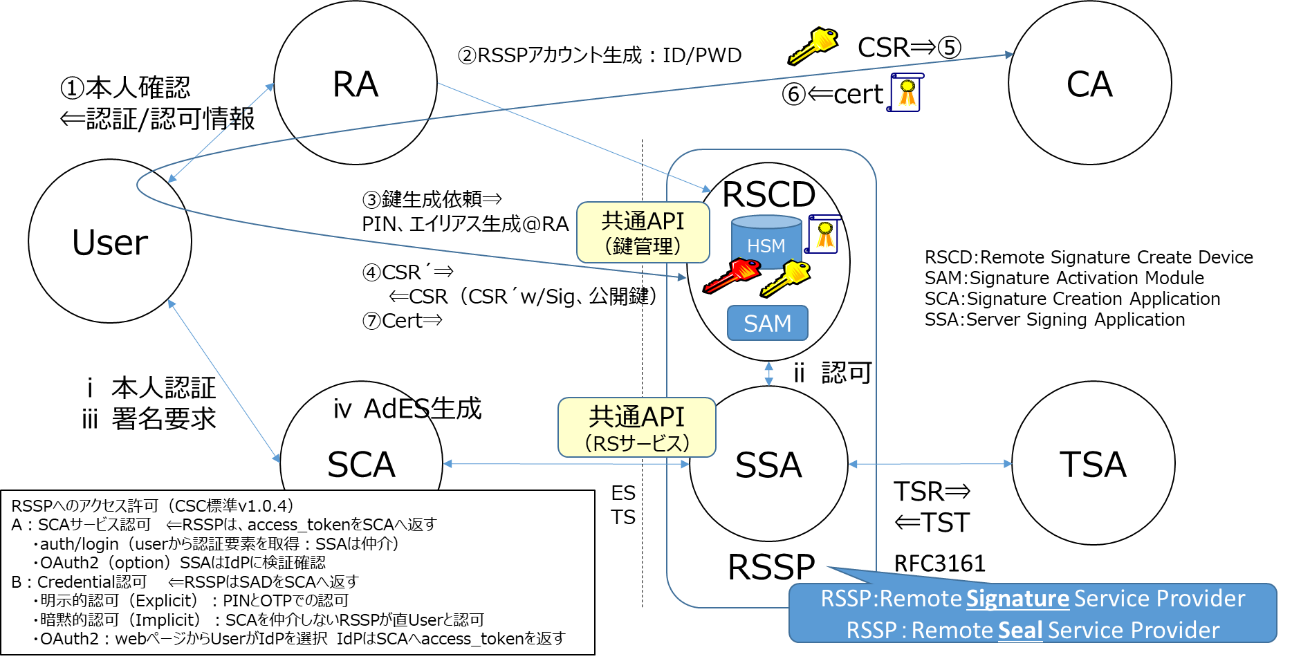


図 1.1　全体スキーム

## 1.2 Cloud Signature Consortium Standardのスコープ

　リモート署名サービスを構築・実装するうえで、現状最もスタンダードと考えるものとしてCloud Signature Consortium Stanadard（以降CSCと呼ぶ）がある。バージョン1.0.4.0 (2019-06)が最新バージョンである。(2021年3月現在)

図1.2はCSC（v1.0.4.0）がサポートするリモート署名のアーキテクチャを表しており、

大きく以下の4つの面とSigner Interfaceで表現されている。

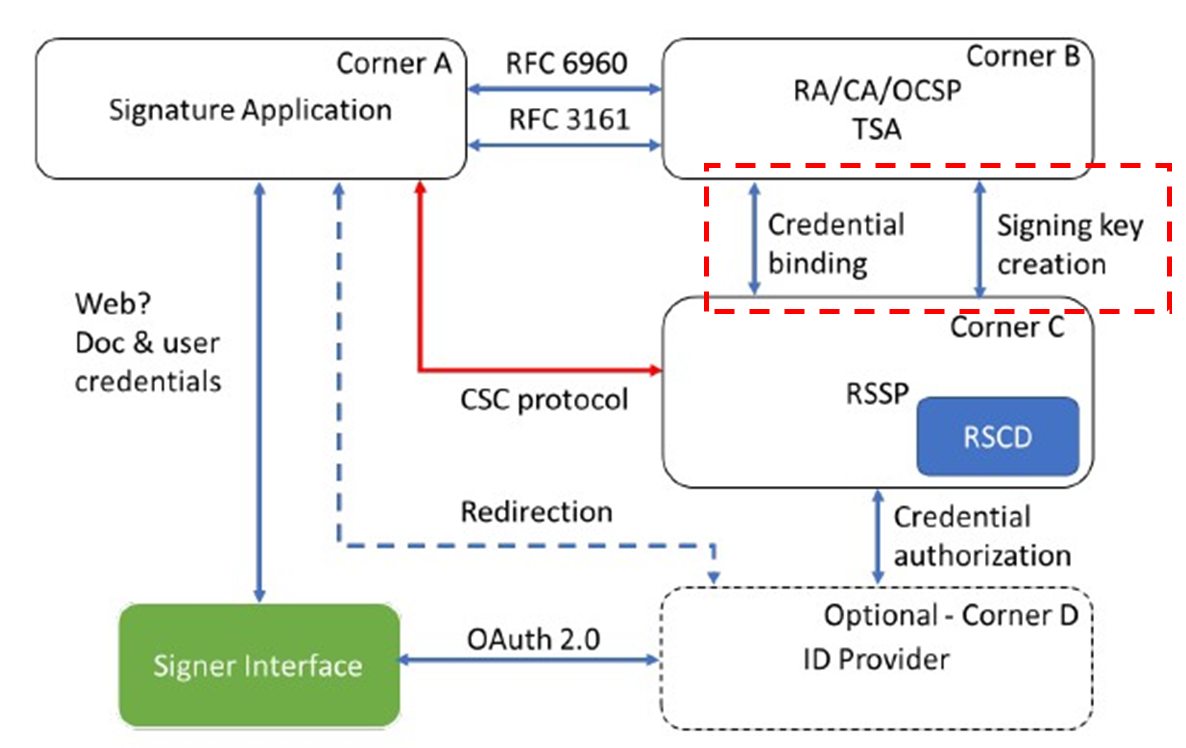


図 1.2　Remote signing corners

(出典：CSC Standard Architectures and protocols for remote signature applications　Published version 1.0.4.0 (2019-06)　P.12)

(A) Signature Application：

署名アプリケーションは、署名するドキュメントをユーザから取得し、必要に応じて、対応する信頼サービスプロバイダーから証明書、失効情報、およびタイムスタンプを取得します。リモート署名サービスプロバイダーに、ハッシュ値の署名を作成するように要求します。

(B)RA/CA/OCSP/TSA：

　RA/CA/OCSPはいずれも認証局サービスの一部でありそれぞれ以下の役割を担っている。

　　・RA（Registration Authority）：登録局を意味し、利用者からの証明書発行申請を受け付けて本人確認審査を実施し、CAへ証明書発行の指示を行う役割である。

・CA（Certification Authority）：認証局を意味しており、広義ではRAの役割も含んでいると考えられ

るが、ここではRAからの指示を受けて証明書発行を行う役割である。

　　・OCSP（Online Certificate Status Protocol）：一般的には認証局が提供する機能であり、証明書の有効性（発行元、失効、有効期限）の確認を行うインターネットサービスである。そのプロトコルはRFC6960として標準化されている。

(C) RSSP（Remote Signature Service Provider）：

リモート署名サービスプロバイダーを表しており署名者の鍵をリモート署名サービスで生成し安全に保管し署名者からの署名依頼に応じてリモート署名サービス内の鍵を使用して署名を行う役割である。

(D) ID Provider (Optional)：

署名者の認証を行う役割である。RSSPが本役割を担うのが一般的である。

(E) Signer Interface：

署名者がSignature ApplicationやID Providerとのやり取りをするためのインターフェースであり、実際にはPCやスマートフォン内のWebブラウザやアプリケーションがこの役割を担う。

現状CSCでは、Signature ApplicationとRSSPの間（赤線部分）のみがCSC protocolとしてAPI定義され、Signer InterfaceとID Providerとのやり取りはOAuth2プロトコルを参照している。図1.2の赤点線で囲まれたRA/CAとRSSPの間で行われる以下２つの連携については現状スコープ外の扱いとなっている。

・Credential binding：署名者と署名鍵の紐づけ

・Signing key creation：署名鍵の生成

## 1.3 ETSI ESIにおける概念

欧州ではリモート署名サービスに関する技術要件やポリシーに関してまとめられたものとして以下のETSI TS 119 431-1がある。

* ETSI TS 119 431-1 V1.1.1 (2018-12)　Electronic Signatures and Infrastructures (ESI);Policy and security requirements for trust service providers;　Part 1: TSP service components operating a remote QSCD /SCDev

このドキュメントの中では、署名者の要求に従ってサーバー側で署名を行うサービスServer Signing Application Service Component(SSASC)と呼ばれ、さらにSSASCは以下6つのサブコンポーネントが定義されている。

表 1.3 SSASCサブコンポーネント

|  |  |  |  |
| --- | --- | --- | --- |
| No | 名前 | 原文説明 | 和訳説明 |
| 1 | OVR: | General requirement (requirement applicable to more than 1 service component) | 全コンポーネント共通の一般的な技術要件 |
| 2 | GEN: | Signing Key Generation Service | 署名鍵生成を行うコンポーネント |
| 3 | LNK | LNK: Certificate/eID means Linking Service | 署名鍵と証明書の紐づけや  署名鍵と署名者との紐づけを行うコンポーネント |
| 4 | SIG | Signature Activation Service | 署名生成時の署名鍵の活性化および署名生成を行うコンポーネント |
| 5 | DEL | Signing Key Deletion Service | 署名鍵の削除を行う |
| 6 | EID | eID Means Provision (optional) | 署名者の認証情報の提供を行うコンポーネント（オプショナル） |

これらのサブコンポーネントと証明書を発行するトラストサービスプロバイダーである認証局との関係を表したものが図1.3である。この関係図はリモート署名システムを署名者が利用開始時に鍵を生成してから、利用終了時に鍵を削除するまでの鍵のライフサイクルが記載されており、まさにリモート署名システム全体を表していると考えることができる。また、図1.2で示したCSCの概念図と似ており、CSCではサブコンポーネントSIG（図1.3 緑部分）と署名アプリの部分がAPI化されていると考えることができる。また、CSCではスコープ外となっている認証局とSSASC（CSCではRSSPと呼ばれる）の関連をここではサブコンポーネントごとに分かりやすく細分化して整理されている。

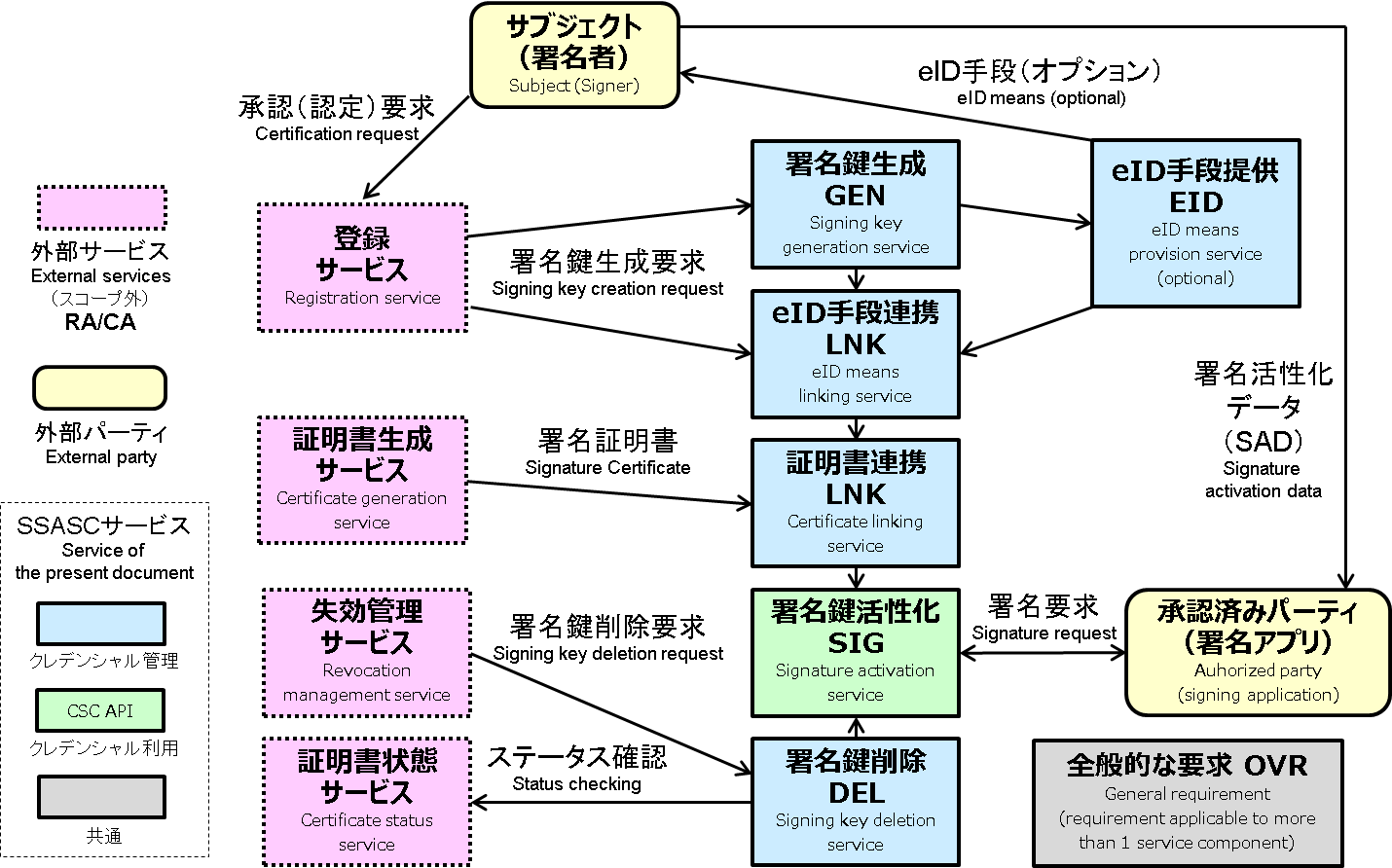


図 1.3 SSASCサブコンポーネント　(出典：ETSI TS 119 431-1 V1.1.1 (2018-12) )

# トラスト共通API技術仕様

本章では、1章で示した欧州の標準仕様やCSC Standardを考慮し、国内および国際間での相互運用が可能なトラストサービスを構築するうえで必要なAPIを定義する。

## 2.1複数認証局との相互運用モデル

　1章で示した通り、CSCではリモート署名システムと認証局の間の手続きについてはスコープ外となっており現状グローバルスタンダードと呼べるAPIは存在していない。日本国内を見ると認証局とリモート署名サービス事業者は複数存在しており、必ずしも同一事業者であるわけではない。また、認証局が多数のリモート署名サービス事業者および多数の署名アプリケーション事業者が相互運用を行うことができるプラットフォームの実現は、電子署名基盤の普及促進につながると考えられる。

そこで本研究では、認証局とリモート署名システムとの間での手続きについてAPI化をすることは相互運用を実現したプロットフォームを構築するうえで有用であると考える。

本研究では、SSASCサブコンポーネントと認証局との関係とそれぞれのコンポーネントの技術要件を考慮し以下のようなAPIを定義した。

### リモート署名における相互運用モデル

実際のビジネスでは、リモート署名サービスの具体的なユースケースが少なくとも3つ存在する可能性がある。 いずれの場合も、SCA、CA、RSSP の間に契約に基づく信頼関係が必要である。

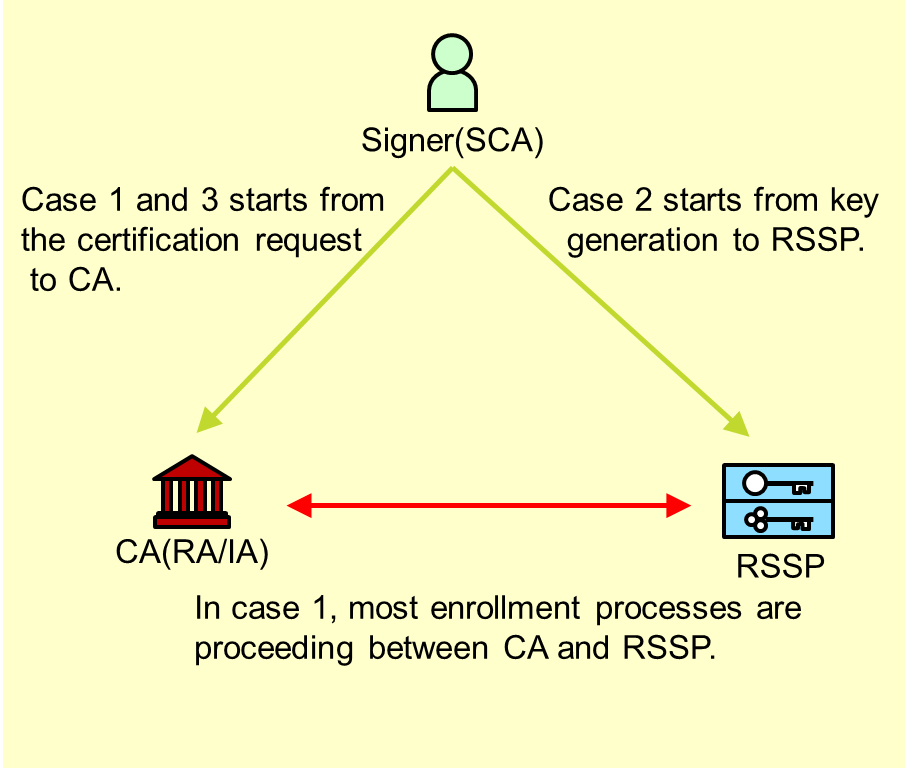


図 2.2 トラストサービスの利用モデル

#### Case1: CA initiative model

署名者は CA に証明書要求を送信する。ほとんどの登録処理は、CAとRSSPの間で進められる。

#### Case2: Signer initiative model

まず、署名者はSCAを経由してRSSPに利用申請する。次に、RSSPは（Ｑ）ＳＣＤ内でキーペアを生成する。署名者は（Ｑ）ＳＣＤ内の署名鍵でCSRに署名し、SCA経由でCAに送信する。

#### Case3: Signer and CA initiative model (Hybrid)

署名者は CA に証明書要求を送信し、SCA を介して自ら CSR に署名する。

### CA initiative model

本モデルはCAが署名者からの証明書発行申請を受けて、署名者の代わりに責任を持ってRSSPへ鍵生成から証明書登録までを行うモデルである。

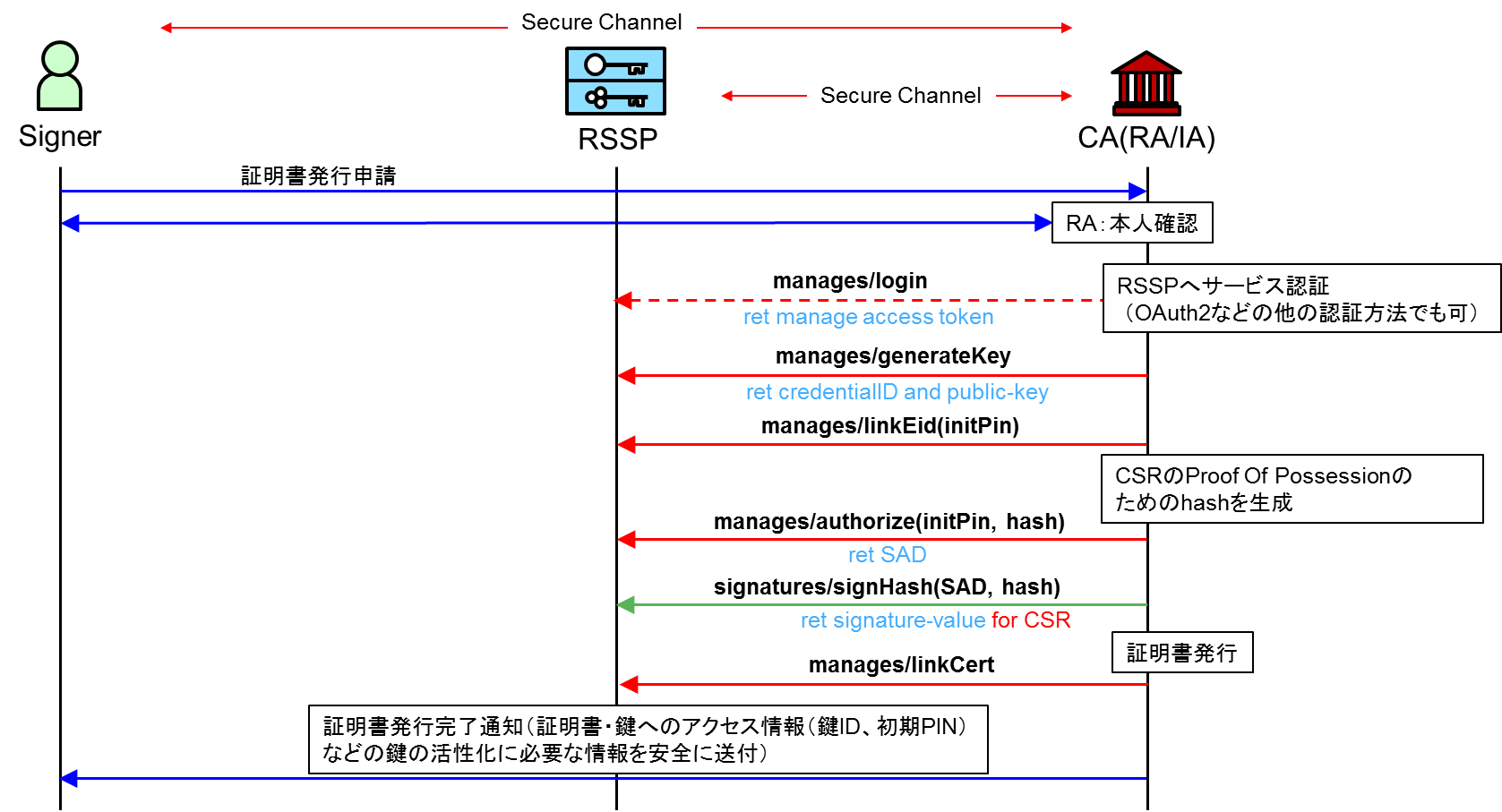


図 2.1.1 Case１：CAがサービス利用のフロントになるモデル(CA initiative model)

### Signer initiative model

署名者自身が鍵生成から証明書の登録まで処理を行うモデルである。

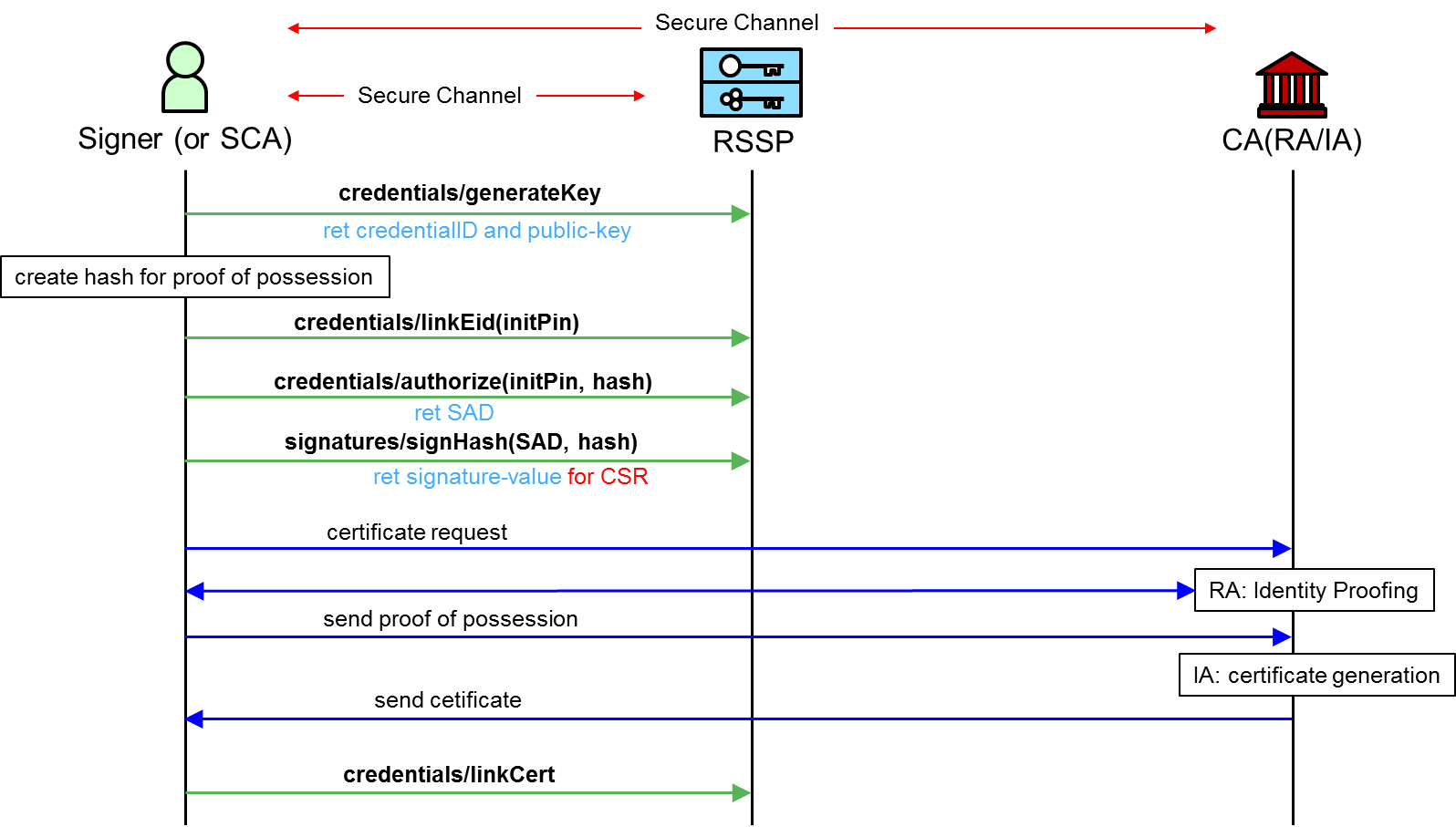


図 2.1.2 Case2：CAがサービス利用のフロントになるモデル(CA initiative model)

### Signer and CA initiative model (Hybrid)

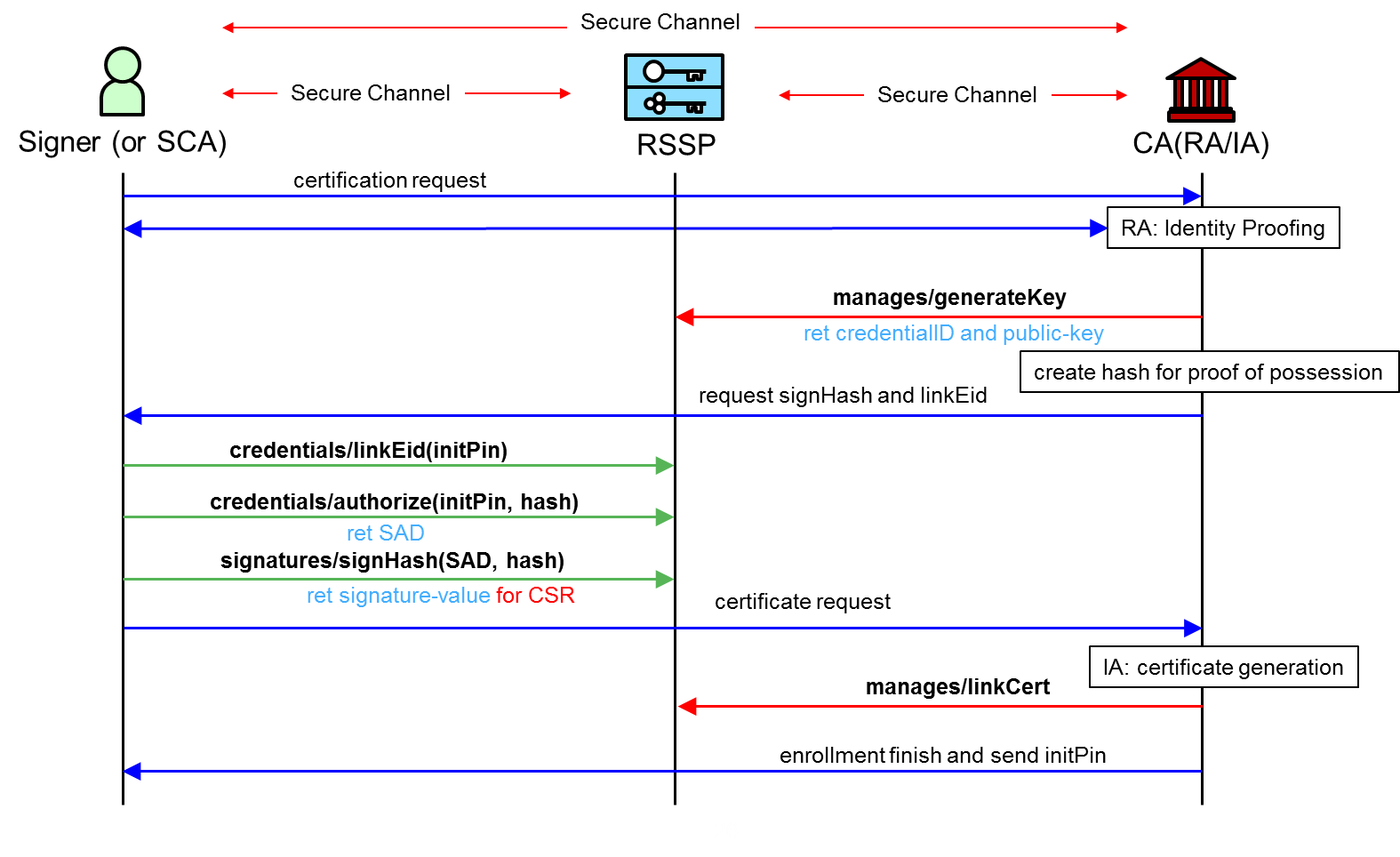


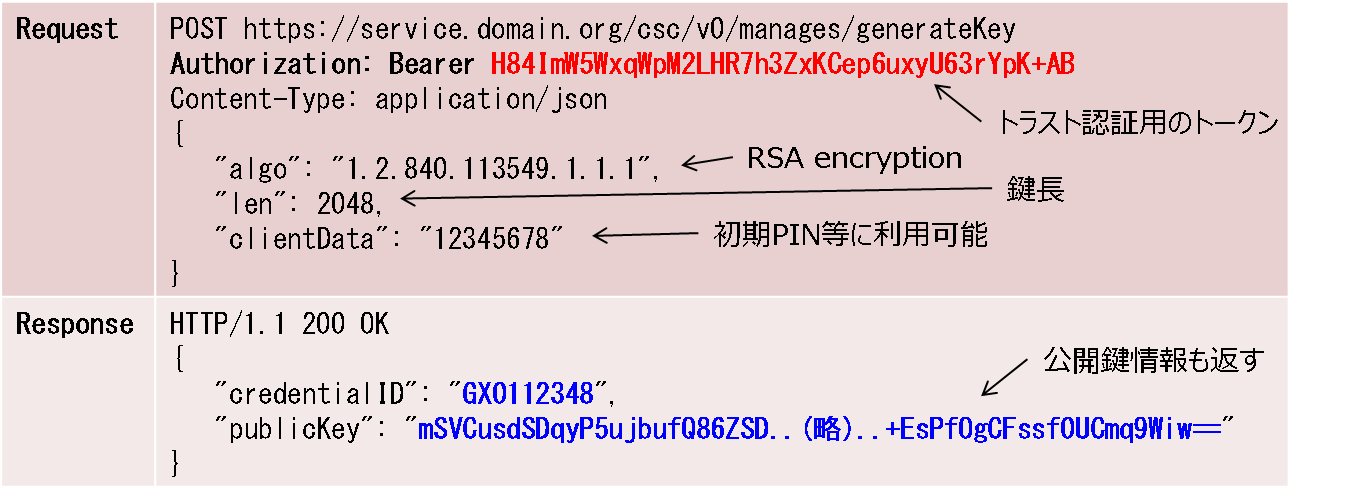
図 2.1.3 Case3：CAがサービス利用のフロントになるモデル(CA initiative model)

## 2.2 複数認証局との相互運用モデル

### 2.2.1　管理API（manages/generateKey）

鍵ペアを生成する為のTSP（認証局）向けAPIである。初期PINとしてclientDataを使う等の応用に関してはベンダー依存となる。

表 2.2.1　/manages/generateKey サンプル

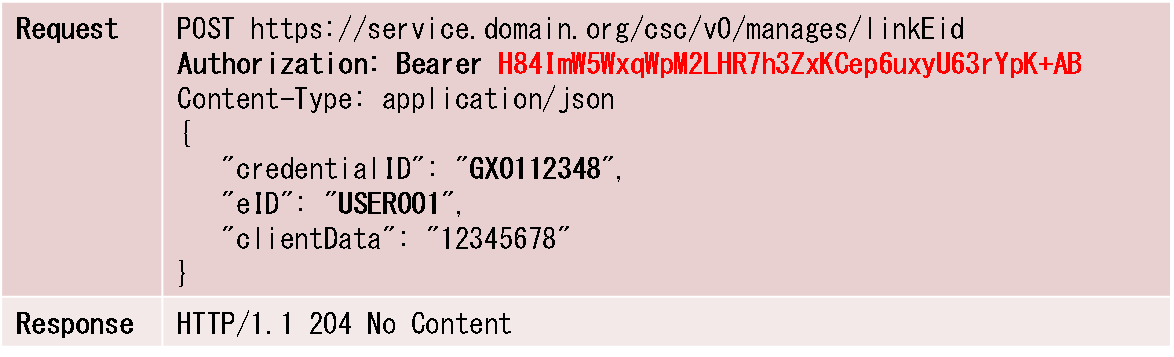


### 2.2.2　管理API（manages/linkEid）

鍵のユニークな識別子であるクレデンシャルIDと署名者ID（eID）を関連付けする為のTSP（認証局）向けAPIである。

eID では無く serviceID やクレデンシャル認可の情報もここで紐付け出来てもよい。

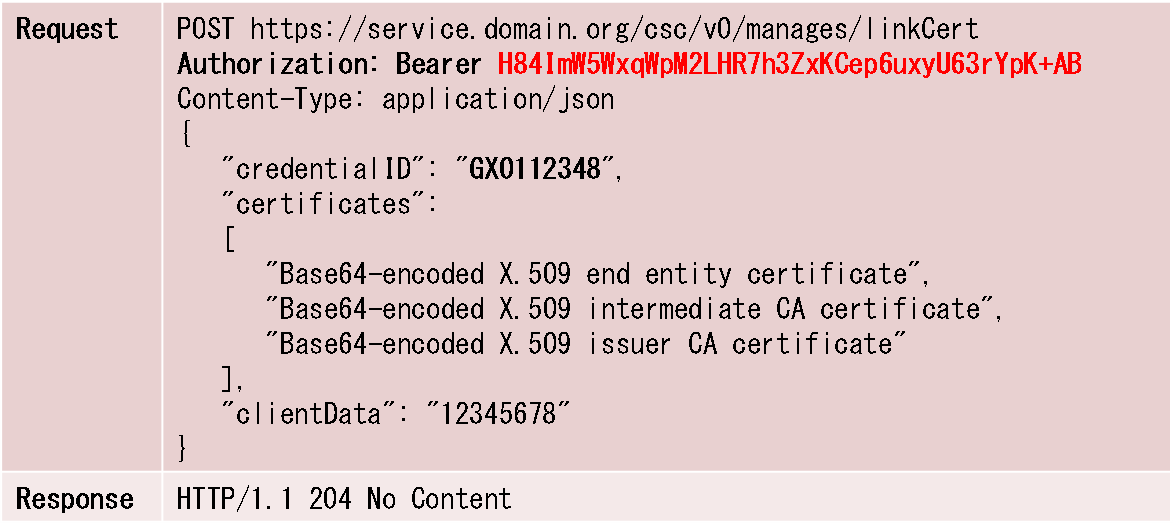
表 2.2.2　/manages/linkEid サンプル



### 2.2.3　管理API（manages/linkCert）

クレデンシャルIDと証明書（認証パス）を関連付けする為のTSP（認証局）向けAPIである。中間CA証明書とルート証明書の扱いに関しては、ここでは対象外としたがさらなる議論が必要かもしれない。

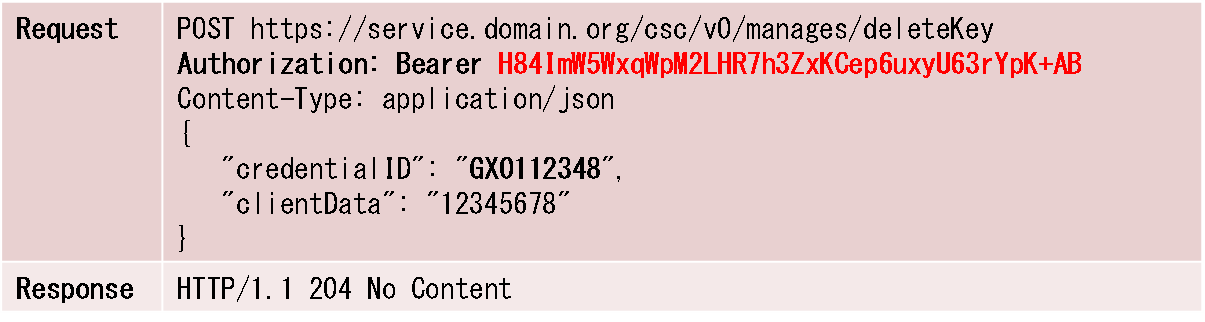
表 2.2.3 /manages/linkCert サンプル



### 2.2.4　管理API（manages/deleteKey）

クレデンシャルIDで指定された鍵を削除する為のTSP（認証局）向けAPIである。

表 2.2.4 /manages/deleteKey サンプル



## 2.3 複数タイムスタンプとの相互運用

ユーザが利用できるタイムスタンプ局のリストを取得するためのAPIおよびタイムスタンプ局の詳細な情報を紹介するためのAPIを新規定義した。

タイムスタンプ

timestamps/list 利用できるタイムスタンプ局のリストを返却する。

timestamps/info 指定のタイムスタンプ局の情報を返却する。

signature/timestamp タイムスタンプ局を指定してタイムスタンプ署名を得る。

## 2.4 アカウント定義

ビジネスで利用されることを前提に、運用の異なる自然人の電子署名と法人・組織の電子署名（ｅシール）で、アカウントを定義した。

電子署名の場合は、特定の自然人であり、ｅシールの場合は、管理者は組織が認めた特定者であり、署名を実施するのは、利用を許可された複数の組織内の担当者となることを考慮する。

表 2.3 アカウント定義

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 定義項目 | | 管理者アカウント | ユーザカウント |
| 1 | 業務上の役割 | | RSサービスを利用するユーザの登録や削除などの管理を担うアカウント | RSサービスを用いて電子署名（=否認防止）やｅシールの生成を行う利用者  ※ユーザカウントは電子署名やｅシールの生成に使用する署名鍵と紐づけられる。 |
| 2 | 対象者 | RSSP  （Signature）  署名サービス | ・ユーザから委託された事業者  ・鍵管理（△） | ・自然人  ・鍵管理 |
| 3 | RSSP  （Seal）  シールサービス | ・組織のｅシール管理者  ・組織から委託された事業者  ・鍵管理 | ・組織内担当者  その鍵で発行できる人  後日トレースできる必要がある。 |

## 2.5 署名検証/証明書検証

### 2.5.1 検証API の仕様策定方針

(1)検証対象データ

* 署名値の検証は数学的な処理なのでRSSPで行う必要性が無いので検討しない。
* 証明書の検証はトラストリストの関連もあり意味があるのでスコープ内とする。
* 署名ドキュメントの検証は連携する署名検証サービスに依存する部分もあるが、RSSPのAPIとして提供する前提でスコープ内とする。

(2)証明書検証API

誰が何のために利用するAPIか想定する必要がある。トラストリストを使った証明書検証を可能にすると言う意味ではGPKIのCVS（証明書検証サーバー）的な使い方と、CVSでも可能だが各種情報の取得を想定。

* 検証者が証明書の有効性を確認する為にレポートを取得する。
* SCAが長期署名を行う為の認証パスと有効性情報を取得する。

(3)署名文書検証API

基本的には署名検証サービスと連携してレポートを返す。レポート形式を指定可能とするが、現在想定されるレポート形式は以下となる。

* ETSI Validation Report（ETSI TS 119 102-2）

　　https://www.etsi.org/standards-search#search=TS119102-2

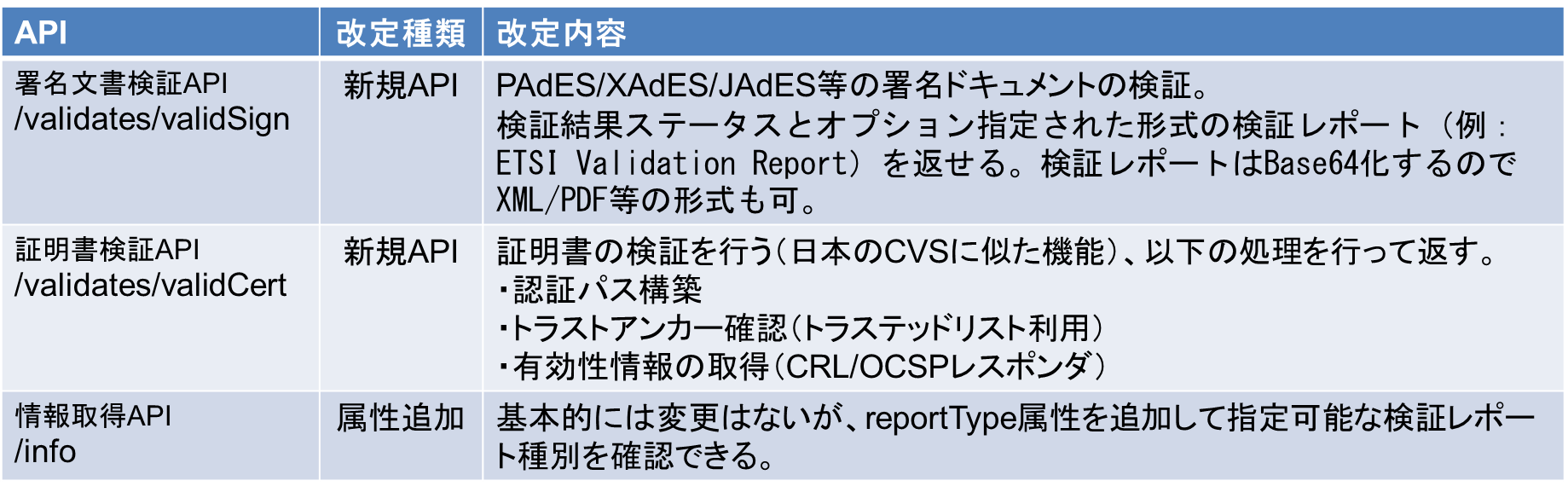
* DSS Diagnostic Data（DSSの独自形式でデモサイトでも取得可能）

　　https://ec.europa.eu/cefdigital/DSS/webapp-demo/doc/dss-documentation.pdf

### 2.5.2 検証API概要

以下に検証APIの概要を示す。「新規API」とはトラスト共通API（V1.0）に対して新たに追加したAPIであり、「属性追加」はトラスト共通API（V1.0）の既存APIに対して属性を追加したことを表している。

表 2.4.2　検証API概要

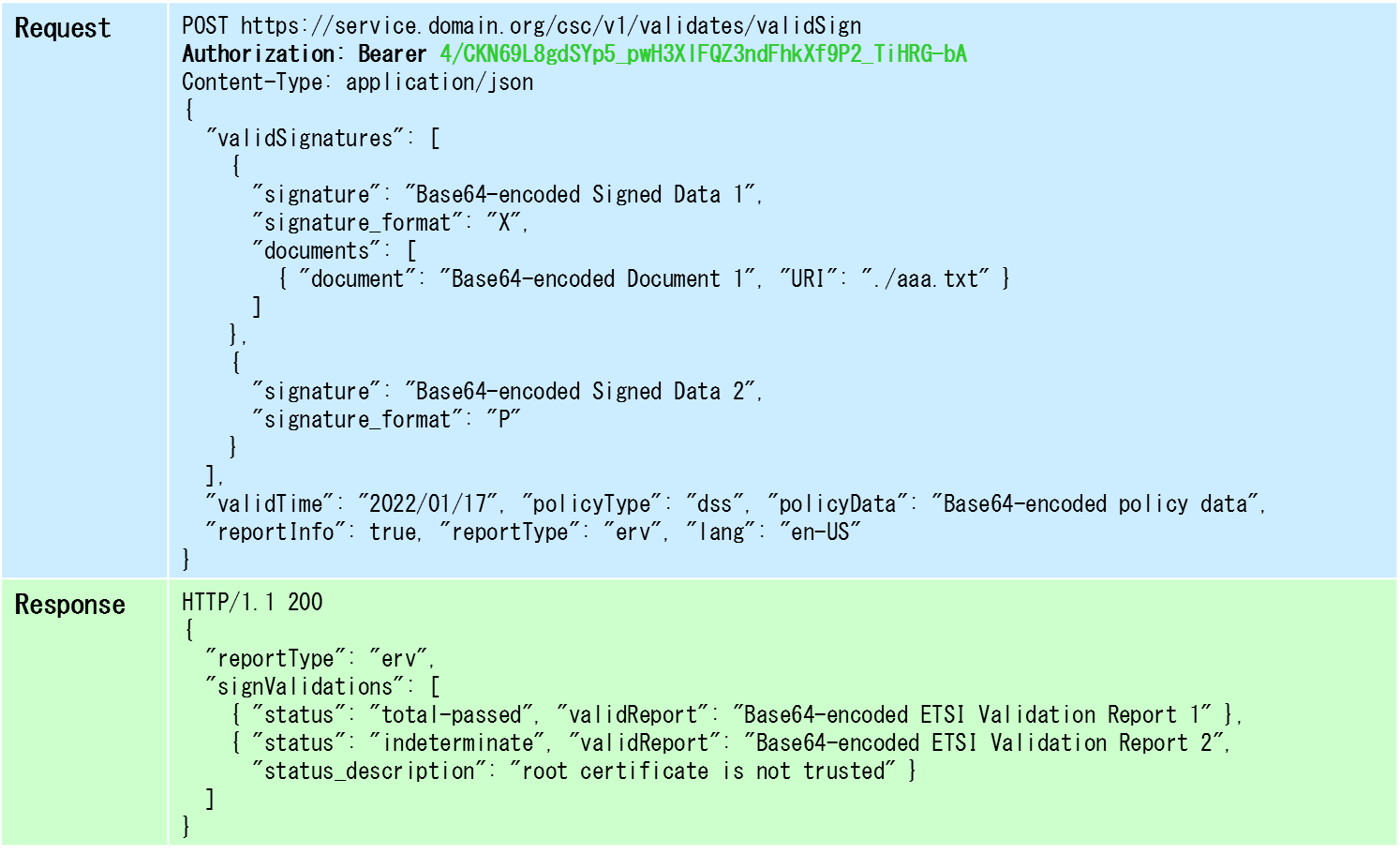


* 検証API用のパス /validates の下のAPI利用にはサービス認証が必要とする。
* validSign の検証結果はステータス以外は指定されたレポート形式をBase64化して返すこととする。検証結果の内容の仕様はスコープ外とする。
* validCert は長期署名作成に必要な情報を返すことがメイン機能となるが validSign と同じく標準化された検証レポートも返すことができる。
* 検証結果に関係無く検証が行えた場合はHTTPステータス200で返し、エラーはHTTPステータス400で返す。

### 2.5.2.1 /validates/validSign

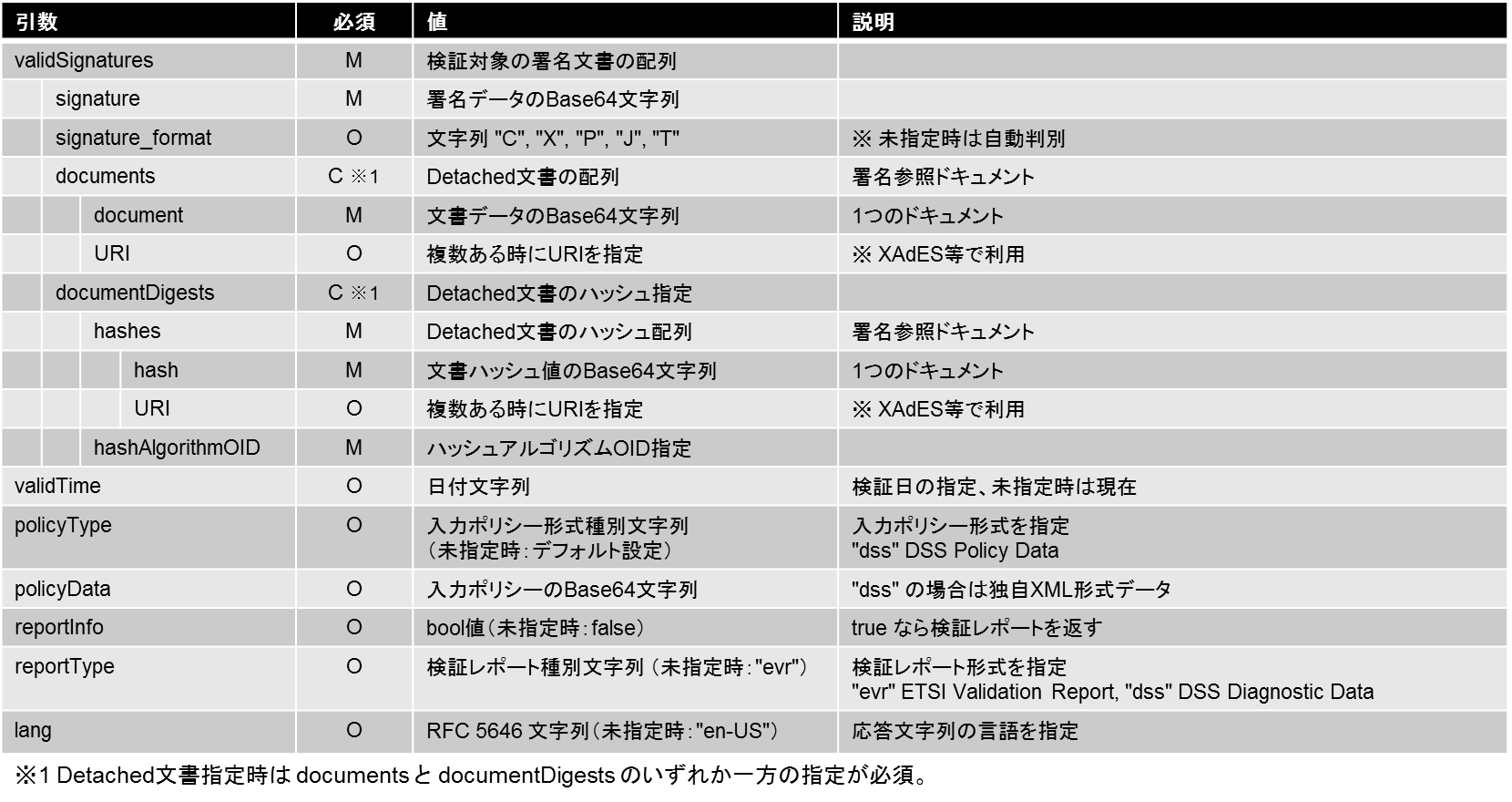
以下に署名データの検証を行うAPIとしてvalidSignの仕様を示す。

表 2.5.2.1-1　/validates/validSign サンプル



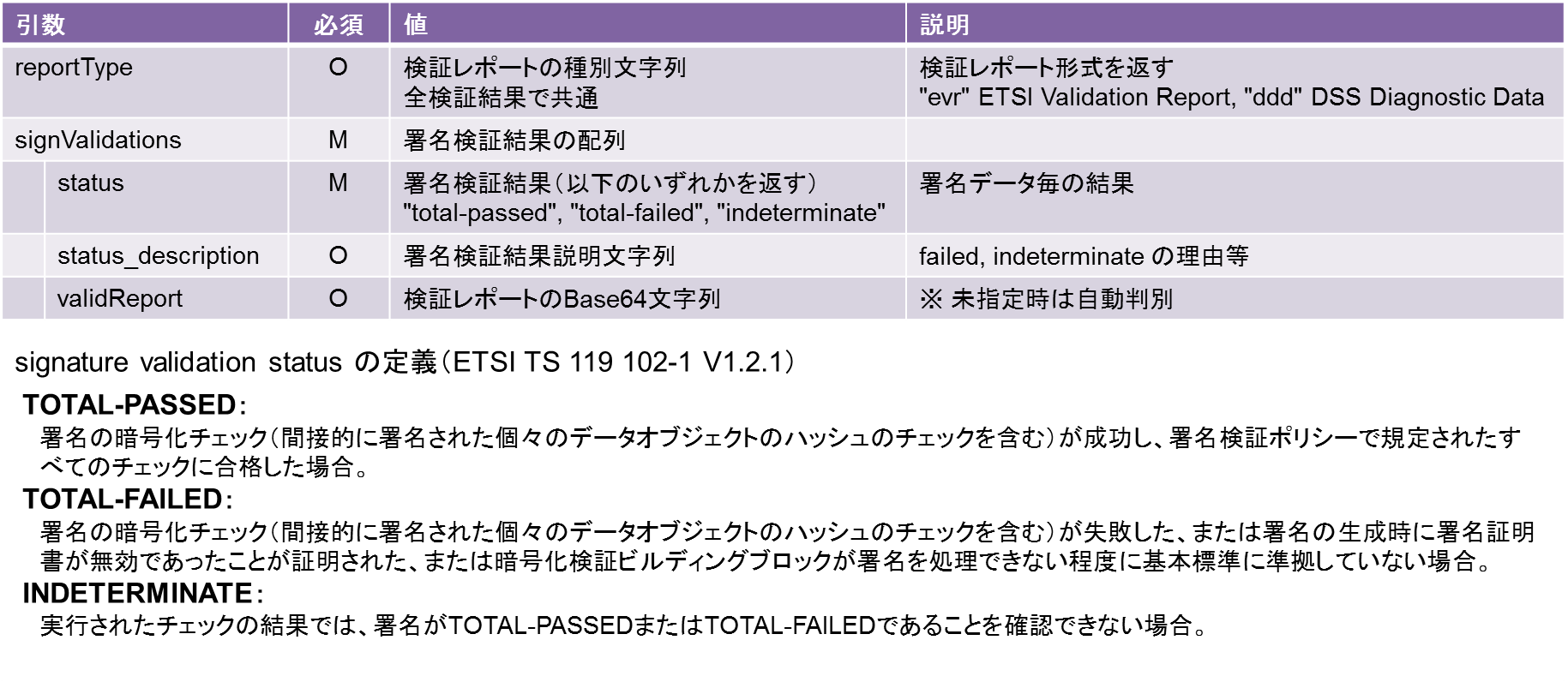
・Request仕様

表 2.5.2.1-2　/validates/validSign Request仕様



・Response仕様

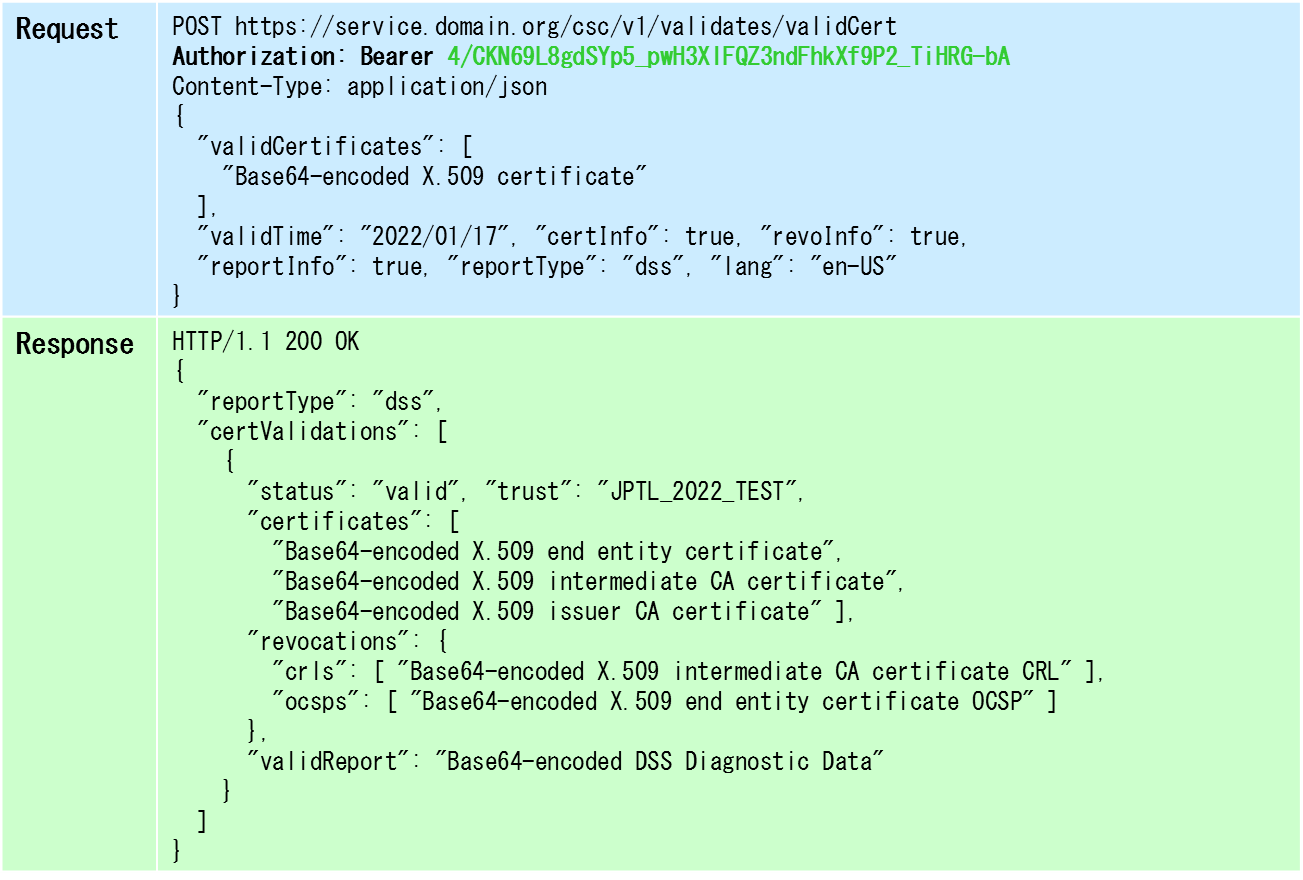
表 2.5.2.1-3　/validates/validSign Response仕様



### 2.5.2.2 /validates/validCert

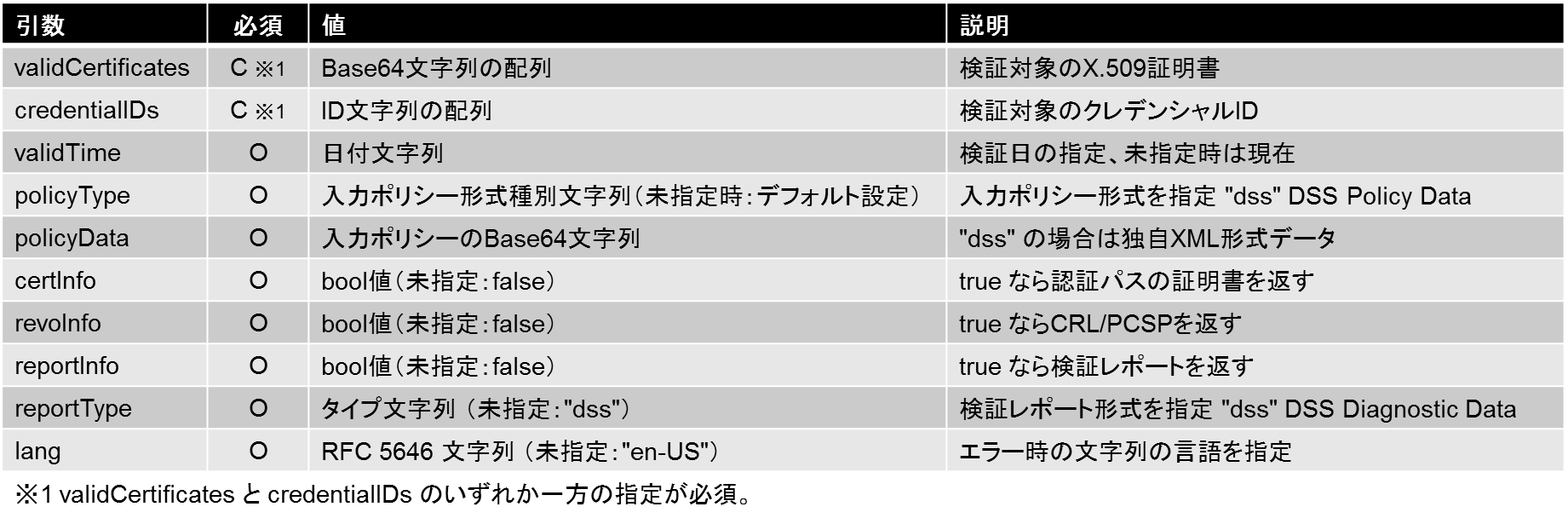
以下に証明書の検証を行うAPIとしてvalidCertの仕様を示す。

表 2.5.2.2-1　/validates/validCerts サンプル



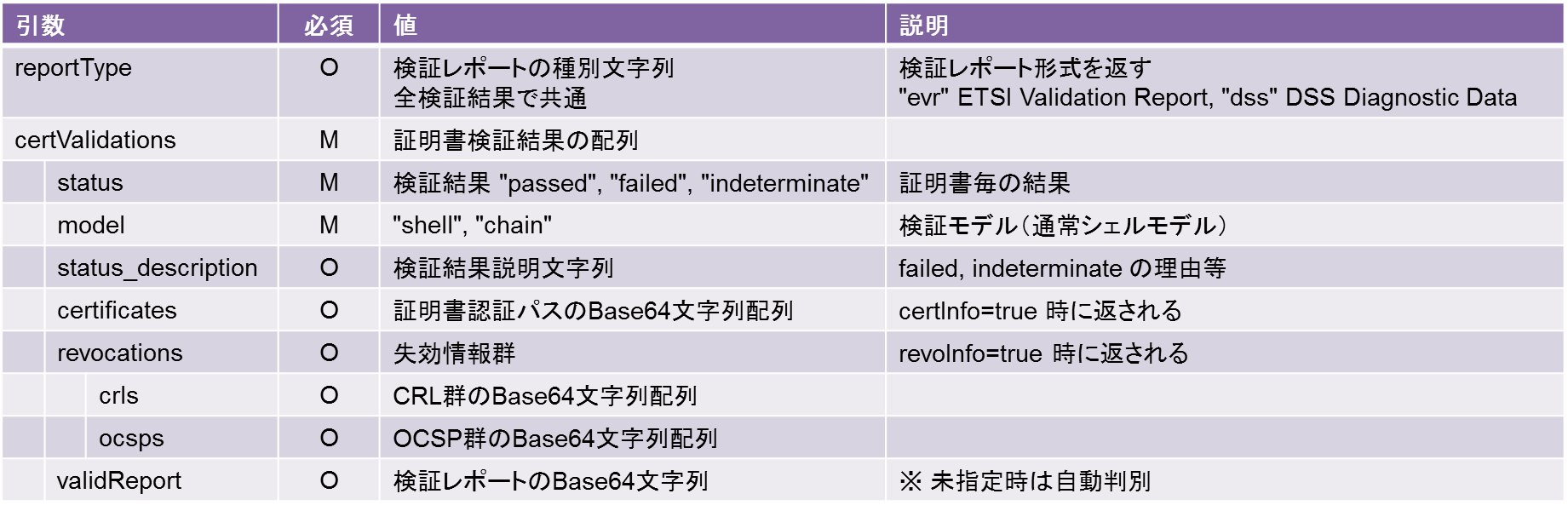
・Request仕様

表 2.5.2.2-2　/validates/validCert Request仕様



・Response仕様

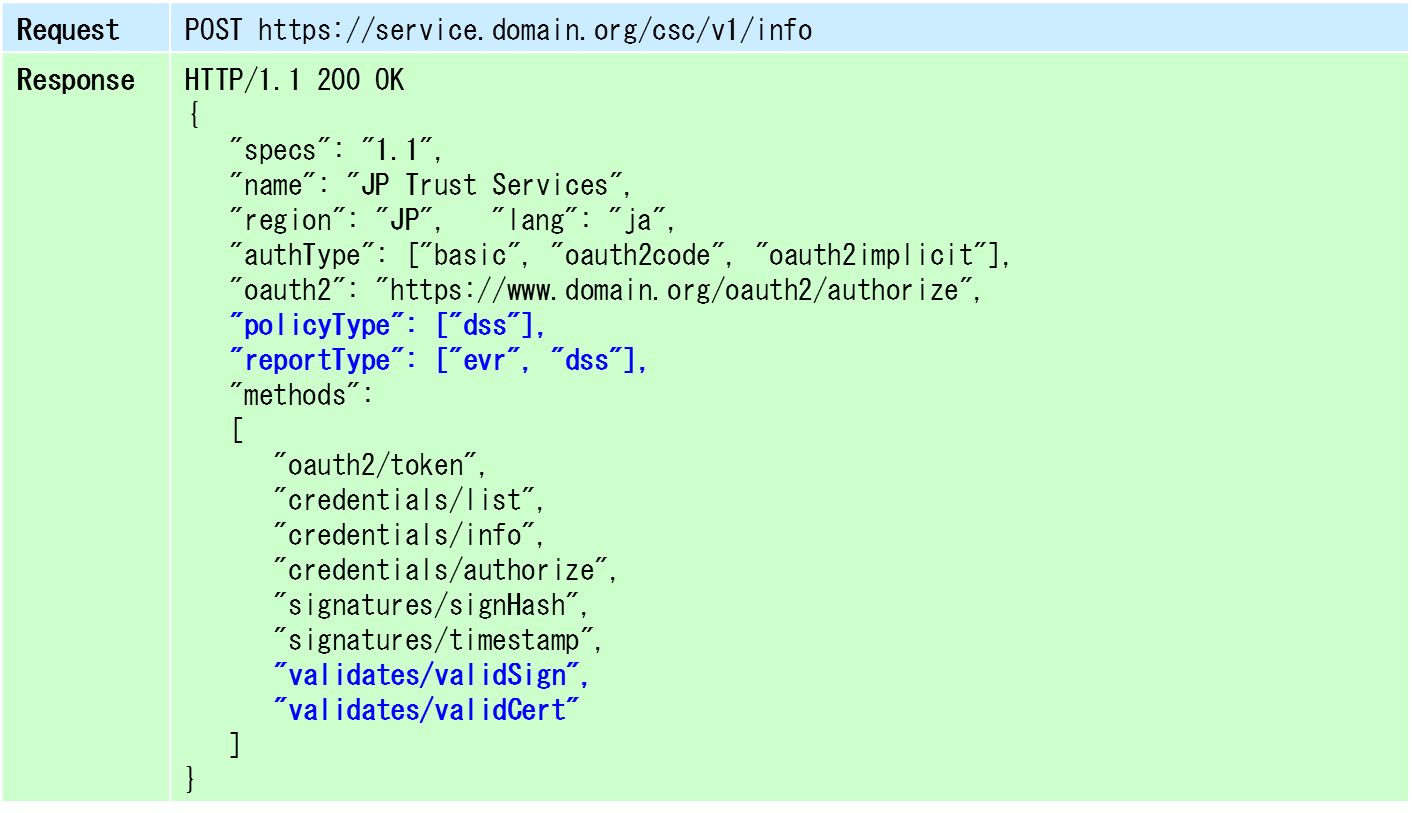
表 2.5.2.2-3　/validates/validCert Response仕様



### 2.5.2.3 /info拡張

上述した検証APIの追加に伴い、/infoの拡張を行う必要がある。以下にその仕様例を示す。

表 2.5.2.3-1　/info サンプル



* 利用可能な policyType と reportType が配列で返される。
* mothods で /validates/ 以下の validSign と validCert の指定が可能になる。

## 2.6 eSeal利用における処理仕様

リモート署名システムにおけるeSealとeSignature(電子署名)の利用における処理仕様の差異について述べる。

1. 署名鍵とIDが 1:1 だけでは無く 1:n になる可能性

・署名者毎にIDを発行して複数IDを署名鍵と紐付けると 1:n が実現可能。シール認可の認証要素を複数の署名者で共有可能性があるが望ましくない。

1. 大量文書への一括シール付与（既存 CSC API で実現可能）

・１回の signatures/signHash で複数のデジタル署名（シール付与）が可能。

・更に signatures/extendTransaction でリフレッシュ複数回シール付与が可能。

1. シール認可の考え方（サーバー自動シール付与）

・SCAとRSSPが一体化する必要があるので CSC API を使わないケースとなる。

・署名認可は設計時に許諾された条件を満たした場合に行われる。

（例：自サーバー発行のPDFファイル（領収書等）のみ認可する。）

・記録（シール付与のログ）の保存は必須。

### 2.6.1 複数署名（シール）：一括（1回のAPIで複数署名計算）

credentials/signHash は numSignatures の数だけ　1回の呼び出しで複数署名値の計算が可能となっている。

・処理シーケンス

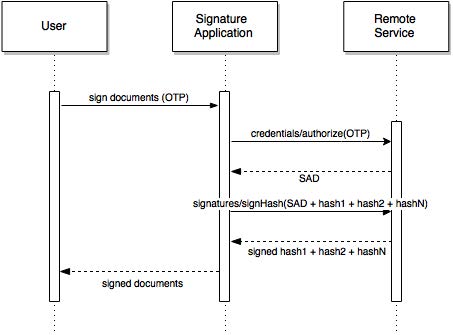
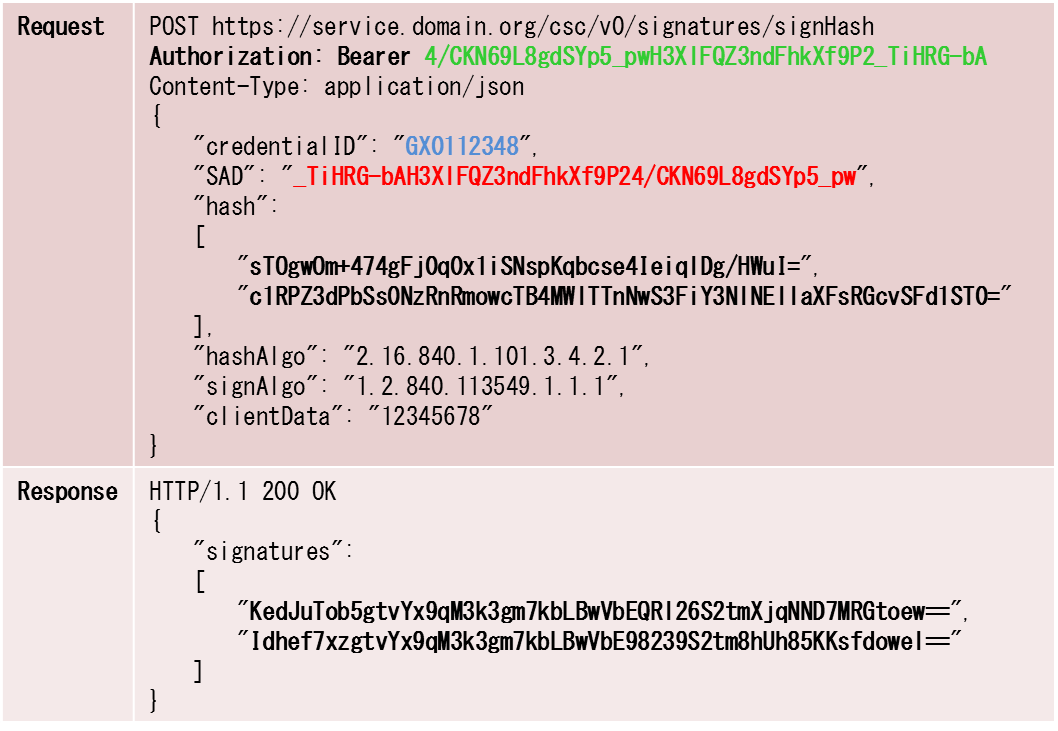


図 2.6.1　/signatures/signHash（eシール利用時）処理シーケンス

・/signatures/signHashのeSeal利用時の使用例

表 2.6.1　/signatures/signHash（eシール利用時）サンプル



### 2.6.2 複数署名（シール）：連続（複数の署名APIの連続呼び出し）

credentials/extendTransaction は使い終わったSADから新しいSADを取得するAPIであり、繰り返すことで何度も連続して署名値計算の credentials/signHash を呼び出すことが可能となっている。

・処理シーケンス

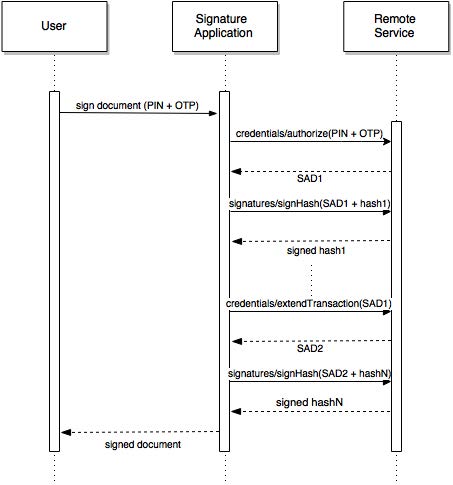
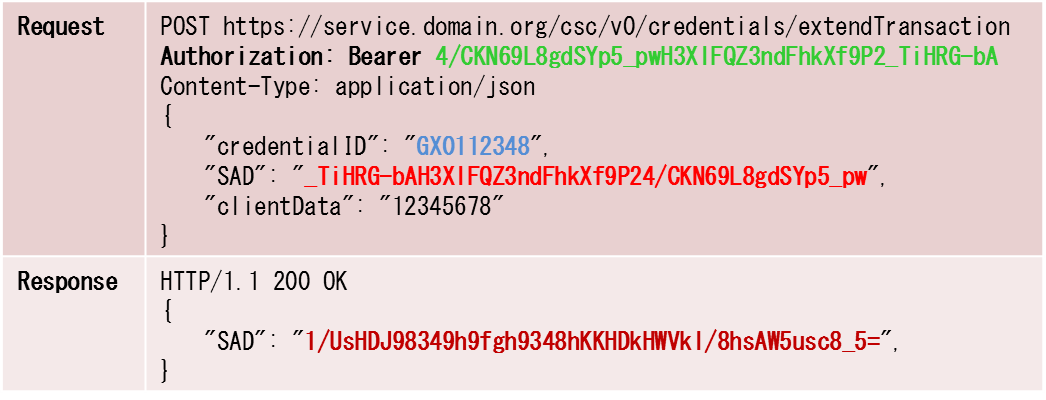


図2.6.2　/ credentials/extendTransaction（eシール利用時）処理シーケンス

・/credentials/extendTransactionのeSeal利用時の使用例

表 2.6.2　/ credentials/extendTransaction（eシール利用時）サンプル



# トラスト共通API定義

## 3.1 共通API　入出力パラメータ

本節では、共通APIとして定義したAPIの入出力パラメータ仕様について設計を行う。表3.1は共通APIとして定義したAPI一覧である。区分が「CSCv1」となっているものはCSC Standard V1.0として既に定義されているものを参照している。「新規」は、共通APIとして新しく定義したものである。「拡張」となっているものは、CSC Standard V1.0の定義を拡張したものを表している。なお、項番17～項番19の3つのAPIについては、図1.2の中でSigner InterfaceとRSSP(またはID　Provider)の間で行われる認証・認可（サービス認証・鍵認可）を行う機能を提供するものであり、その仕様についてはOAuth2.0プロトコルを参照しているため、本ドキュメントでは対象外とする。

表 3.1 API一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 業務内容 | エンドポイント | 区分 | SSASC  サブコンポーネント |
| 1 | 情報取得 | info | CSCv1 | OVR |
| 2 | 管理 | manages/generateKey | 共通APIv1.0 | GEN |
| 3 | manages/linkEid | 共通APIv1.0 | LNK |
| 4 | manages/linkCert | 共通APIv1.0 | LNK |
| 5 | manages/deleteKey | 共通APIv1.0 | DEL |
| 6 | 署名鍵情報管理 | credentials/info | CSCv1 | SIG |
| 7 | credentials/list | CSCv1 | SIG |
| 8 | 署名鍵認証・認可 | credentials/authorize | CSCv1 | SIG |
| 9 | credentials/extendTransaction | CSCv1 | SIG |
| 10 | credentials/sendOTP | CSCv1 | SIG |
| 11 | タイムスタンプ情報照会 | timestamps/list | 共通APIv1.0 | SIG |
| 12 | timestamps/info | 共通APIv1.0 | SIG |
| 13 | 署名 | signatures/signHash | CSCv1 | SIG |
| 14 | タイムスタンプ | signatures/timestamp | 共通APIv1.0  （CSCv1拡張） | SIG |
| 15 | 検証 | validates/validSign | 共通APIv2.0 | - |
| 16 | validates/validCert | 共通APIv2.0 | - |
| 15 | 認証・認可（サービス認証・鍵認可） | oauth2/authorize\* | OAuth2.0 | SIG |
| 16 | oauth2/token\* | OAuth2.0 | SIG |
| 17 | oauth2/revoke\* | OAuth2.0 | SIG |

### 3.1.1 info

　　　[区分]　CSCv1に定義されているAPI

（a）概要

RSサービスに関する情報と、それが実装したAPIメソッドのリストを返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/info

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| lang | String | 任意 | RFC 3066に定義された言語コード（このAPIではja-JP固定） |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| specs | String | 必須 | プロバイダによって実装されたこの仕様のバージョン |
| name | String | 必須 | RSサービスの商品名 |
| logo | String | 必須 | オンラインで公開するRSサービスのロゴを含む画像ファイルのURI |
| region | String | 必須 | RSSPが設立されている国のISO 3166-1 Alpha-2コード |
| lang | String | 必須 | RFC 3066に定義された言語コード（このAPIではja-JP固定） |
| atuthType | Array of String | 必須 | RSサービスでサポートされる認証メカニズムに対応する1つ以上の値を指定します |
| oauth2 | String | 任意 | RSサービスによって提供されるOAuth 2.0サービス許可エンドポイントの完全なURIを指定します |
| policyType | Array of String | 任意 | RSサービスによって提供される検証APIがサポートしているポリシータイプ |
| reportType | Array of String | 任意 | RSサービスによって提供される検証APIがサポートしている検証レポートのフォーマットタイプ |
| methods | Array of String | 必須 | すべてのAPIメソッドの名前のリスト |

### 3.1.2 manages/generatekey

　　　[区分]　新規に定義したAPI

（a）概要

鍵ペアを生成する為のTSP（認証局）向けAPIである。初期PINとしてclientDataを使う等の応用に関してはベンダー依存となる。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/manages/generateKey

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| algo | String | 必須 | 鍵の公開鍵暗号アルゴリズム |
| len | String | 必須 | 鍵長 |
| subjectDn | String | 任意 | CSRに記載するsubjectDN情報。指定された場合はCSRを出力する。 |
| csrInfoTemplate | String | 任意 | CSR生成のためのテンプレートデータ。指定された場合はCSRを出力する。 |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| publicKey | String | 必須 | 生成された公開鍵データ（base64形式） |
| csr | String | 任意 | 生成した鍵から生成した証明書発行申請(CSR)データのBase64文字列。入力パラーメータのsubjectDnまたはcsrInfoTemplateが指定された場合のみ出力される。 |

### 3.1.3 manages/linkEid

　　　[区分]　新規に定義したAPI

（a）概要

鍵のユニークな識別子であるクレデンシャルIDと署名者ID（eID）を関連付けする為のTSP（認証局）向けAPIである。eID では無く serviceID やクレデンシャル認可の情報もここで紐 付 け出来てもよい。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/manages/linkEid

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| eID | String | 任意 | 署名者を識別するために利用するID |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

　なし。

### 3.1.4 manages/linkCert

　　　[区分]　新規に定義したAPI

（a）概要

クレデンシャルIDと証明書（認証パス）を関連付けする為のTSP（認証局）向けAPIである。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/manages/linkCert

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | credentialID | 必須 | クレデンシャル情報を識別するID |
| certificates | String | 必須 | クレデンシャル情報に紐づく公開鍵証明書データ（base64形式） |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

　なし。

### 3.1.5 manages/deleteKey

　　　[区分]　新規に定義したAPI

（a）概要

クレデンシャルIDで指定された鍵を削除する為のTSP（認証局）向けAPIである。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/manages/deleteKey

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

　なし。

### 3.1.6 credentials/list

　　　[区分]　CSCv1に定義されているAPI

（a）概要

ユーザに関連付けられた資格情報のリストを返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/credentials/list

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| maxResults | Number | 任意 | 返されるアイテムの最大数 |
| pageToken | String | 任意 | アイテムの新しいページのページトークン |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialIDs | Array of String | 必須 | 一つ以上のクレデンシャル情報を識別するIDの配列 |
| nextPageToken | String | 任意 | アイテムの次のページのページトークン |

### 3.1.7 credentials/info

　　　[区分]　CSCv1に定義されているAPI

（a）概要

署名資格情報、関連する証明書、およびサポートされている承認メカニズムの説明に関する情報を返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/credentials/info

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| certificates | String | 任意 | 取得する証明書のフィルター条件（none/single/chain） |
| certInfo | Boolean | 任意 | エンドエンティティ証明書の情報を印刷可能な文字列として返すかどうかの指定 |
| authInfo | Boolean | 任意 | この資格情報（PINおよびOTPグループ）でサポートされる許可メカニズムに関する情報を返すかどうかの指定 |
| lang | String | 任意 | RFC 3066に定義された言語コード（このAPIではja-JP固定） |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| key | JSON | 必須 | 署名キーに関する情報 |
| status | String | 必須 | 資格情報の署名キーの有効化のステータス（enabled/disabled） |
| algo | Array of String | 必須 | サポートされているキーアルゴリズムのOIDのリスト |
| len | Integer | 必須 | ビット単位の暗号化キーの長さ |
| curve | String | 任意 | ECDSA曲線のOID |
| cert | JSON | 必須 | 署名証明書に関する情報 |
| status | String | 任意 | ステータススキーマ（valid|expired/revoked/suspended） |
| certificates | Array of String | 任意 | 証明書チェーンからの1つ以上のBase64エンコードX.509v3証明書 |
| issuerDN | String | 任意 | 印刷可能な文字列形式のX.509v3エンドエンティティ証明書からの発行者サブジェクト識別名 |
| serialNumber | String | 任意 | 16進エンコード形式のX.509v3証明書のシリアル番号 |
| subjectDN | String | 任意 | 印刷可能な文字列形式のX.509v3証明書の識別名 |
| validFrom | String | 任意 | 印刷可能な文字列形式のX.509v3証明書からの有効開始日 |
| validTo | String | 任意 | 印刷可能な文字列形式のX.509v3証明書の有効期間終了日 |
| authMode | String | 必須 | 許可モードの1つを指定（implicit/explicit/oauth2code/oauth2token） |
| SCAL | String | 任意 | CEN EN 419 241-1で定義されている、資格情報に必要な唯一のコントロール保証レベルを指定（1/2） |
| multisign | Integer | 必須 | 作成できる署名の最大数 |
| lang | String | 必須 | RFC 3066に定義された言語コード（このAPIではja-JP固定） |

### 3.1.8 credentials/authorize

　　　[区分]　CSCv1に定義されているAPI

（a）概要

署名用の資格情報へのアクセスを許可する。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/credentials/authorize

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| numSignatures | Integer | 必須 | 一度に承認する署名の数 |
| hash | Array of String | 任意 | 署名対象の一つ以上のbase64エンコードハッシュ値 |
| PIN | String | 任意 | ユーザから収集されたPIN |
| OTP | String | 任意 | ユーザから収集されたOTP |
| description | String | 任意 | langで指定した言語での承認トランザクションの自由形式の説明が含まれています |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| SAD | String | 必須 | signatures/signHashメソッドへの入力として提供する署名アクティベーションデータ |
| expiresIn | Number | 任意 | サービスアクセストークンの有効期間（秒） |

### 3.1.9 credentials/extendTransaction

　　　[区分]　CSCv1に定義されているAPI

（a）概要

マルチ署名トランザクションの有効性を拡張する。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/credentials/extendTransaction

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| hash | Array of String | 任意 | 署名対象の一つ以上のbase64エンコードハッシュ値 |
| SAD | String | 必須 | signatures/signHashメソッドへの入力として提供する署名アクティベーションデータ |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| SAD | String | 必須 | signatures/signHashメソッドへの入力として提供する署名アクティベーションデータ |
| expiresIn | Number | 任意 | サービスアクセストークンの有効期間（秒） |

### 3.1.10 credentials/sendOTP

　　　[区分]　CSCv1に定義されているAPI

（a）概要

クレデンシャル情報に関連付けられ、リモートサービスによって管理されるオンラインのワンタイムパスワード（OTP）生成する。ワンタイムパスワードはリモートサービス（SMS、電子メール、アプリなど）によって管理される合意済みの通信チャネルを介してクレデンシャル情報を所有するユーザーに配信される。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/credentials/sendOTP

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

なし。

### 3.1.11 timestamps/list

　　　[区分]　 新規に定義したAPI

（a）概要

利用可能なタイムスタンプ局IDのリストを返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/timestamps/list

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| maxResults | Number | 任意 | 返されるアイテムの最大数 |
| pageToken | String | 任意 | アイテムの新しいページのページトークン |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| timestampIDs | Array of String | 必須 | 一つ以上のタイムスタンプ局IDの配列 |
| defTimestampID | String | 必須 | timestampIDsが複数利用可能な場合、デフォルトで利用されるタイムスタンプ局ID |
| nextPageToken | String | 任意 | アイテムの次のページのページトークン |

### 3.1.12 timestamps/info

　　　[区分]　新規に定義したAPI

（a）概要

タイムスタンプ局の情報を返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/timestamps/info

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| timestampID | String | 必須 | タイムスタンプ局ID |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| description | String | 任意 | langで指定した言語での承認トランザクションの自由形式の説明 |
| status | String | 任意 | 現在のサービスの利用可否。 |
| hashAlgos | String | 任意 | messegeInprintに指定可能なハッシュアルゴリズムの配列 |
| policies | String | 任意 | 指定可能なポリシーOIDの配列 |
| lang | String | 任意 | descriptionで使用される言語 |

### 3.1.13 signatures/signHash

　　　[区分] CSCv1に定義されているAPI

（a）概要

1つ以上のハッシュ値から生のデジタル署名を計算する。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/signatures/signHash

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| credentialID | String | 必須 | クレデンシャル情報を識別するID |
| SAD | String | 必須 | signatures/signHashメソッドへの入力として提供する署名アクティベーションデータ |
| hash | Array of String | 必須 | 署名対象の一つ以上のbase64エンコードハッシュ値 |
| hashAlgo | String | 任意 | ハッシュ値の計算に使用されるアルゴリズム |
| signAlog | String | 必須 | 署名に使用するアルゴリズムのOID |
| signAlgoParams | String | 任意 | Base64エンコードのDERエンコードASN.1署名パラメータ |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| signatures | Array of String | 必須 | Base64でエンコードされた署名付きハッシュの配列 |

### 3.1.14 signatures/timestamp

　　　[区分]　CSCv1の定義を拡張したAPI

（a）概要

入力ハッシュ値のタイムスタンプトークンを返す。

（b）エンドポイントURL

/jcsc/v1/csc-api/v1/signatures/timestamp

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| timestampID  【拡張したパラメータ】 | String | 必須 | タイムスタンプ局ID。 |
| hash | String | 必須 | タイムスタンプを付与する対象のbase64エンコードハッシュ値 |
| hashAlgo | String | 必須 | ハッシュ値の計算に使用されるアルゴリズム |
| nonce | String | 任意 | 呼び出し元アプリケーションによって一度だけ生成される可能性が高い大きな乱数 |
| clientData | String | 任意 | 呼び出し元アプリケーションからの任意のデータ |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| timestamp | String | 必須 | RFC 5816で更新されたRFC 3161で定義されているBase64エンコードのタイムスタンプトークン |

### 3.1.15 validates/validSign

　　　[区分]　トラスト共通API V2.0で新規に定義したAPI

（a）概要

PAdES/XAdES/JAdESなどの署名ドキュメントの有効性検証を行う。

（b）エンドポイントURL

/jcsc/v2/csc-api/v2/validates/validSign

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| validSignatures | JSON | 必須 | 検証対象の署名文書の配列 |
| trust | String | 任意 | 検証に使用するトラストアンカーを示す文字列。  サービス事業者ごとに定義。  （例）“JPTL" , "EULOTL" |
| trustCert | String | 任意 | 検証に使用するトラストアンカー証明書。Base64文字列の配列 |
| validTime | String | 任意 | 検証日を指定 |
| policyType | String | 任意 | 入力ポリシー形式を指定  （例）"dss" DSS Policy Data |
| policyData | String | 任意 | "dss" の場合は独自XML形式データ |
| reportInfo | String | 任意 | true なら検証レポートを返す |
| reportType | String | 任意 | 検証レポート形式を指定  （例）  "evr" ETSI Validation Report, "ddd" DSS Diagnostic Data |
| lang | String | 任意 | 出力文字列の言語を指定 |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| reportType | String | 任意 | 検証レポート形式を返す  (例)  "evr" ETSI Validation Report, "ddd" DSS Diagnostic Data |
| trust | String | 任意 | 検証に使用するトラストアンカーを示す文字列。  サービス事業者ごとに定義。  （例）“JPTL" , "EULOTL" |
| trustCert | String | 任意 | 検証に使用するトラストアンカー証明書。Base64文字列の配列 |
| signValidations | JSON | 必須 | 署名検証結果の配列 |

### 3.1.16 validates/validCert

　　　[区分]　トラスト共通API V2.0で新規に定義したAPI

（a）概要

証明書(X.509)の以下有効性検証を行う。

・証明書パスの構築

・トラストアンカー確認

・証明書の有効性（有効期限、失効確認）の確認を行う。

（b）エンドポイントURL

/jcsc/v2/csc-api/v2/validates/validCert

（c）入出力パラメータ

以下の入出力パラメータはJSON形式によって設定される。

・入力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| validCertificates | String | 条件付き必須 | 検証対象のX.509証明書  (Base64文字列の配列) |
| trust | String | 任意 | 検証に使用するトラストアンカーを示す文字列。  サービス事業者ごとに定義。  （例）“JPTL" , "EULOTL" |
| trustCert | String | 任意 | 検証に使用するトラストアンカー証明書。Base64文字列の配列 |
| validTime | String | 任意 | 検証日を指定 |
| policyType | String | 任意 | 入力ポリシー形式を指定  （例）"dss" DSS Policy Data |
| policyData | String | 任意 | "dss" の場合は独自XML形式データ |
| certInfo | String | 任意 | true なら認証パスの証明書を返す |
| revoInfo | String | 任意 | true ならCRL/PCSPを返す |
| reportInfo | String | 任意 | true なら検証レポートを返す |
| reportType | String | 任意 | 検証レポート形式を指定  （例）検証レポート形式を指定  "evr" ETSI Validation Report, "ddd" DSS Diagnostic Data |
| lang | String | 任意 | 出力文字列の言語を指定 |

・出力パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| 項目名 | 項目値 | 必須 | 説明 |
| reportType | String | 任意 | 検証レポート形式を返す  (例)  "evr" ETSI Validation Report, "ddd" DSS Diagnostic Data |
| trust | String | 任意 | 検証に使用するトラストアンカーを示す文字列。  サービス事業者ごとに定義。  （例）“JPTL" , "EULOTL" |
| trustCert | String | 任意 | 検証に使用するトラストアンカー証明書。Base64文字列の配列 |
| certValidations | JSON | 必須 | 証明書検証結果の配列 |