ArubaTrustedPlatform API - ver. 1.0.0

22/Sep/2025 15:36

- General informations
  - Glossary of terms
  - General Notes
  - List of supported web methods
  - Flow, Work Unit and Utilities logic
    - ExecuteFlow method
  - Developer API docs
  - SOAP interface
  - REST interface
- Types and Constants
  - Resources
  - Devices
  - SignatureConstraint
  - Enums
  - Encryption Enum
  - Other Types
  - Date time convention
- Work Units
  - Signature work units
    - Result
    - CAdES
      - Sample CAdES-T
    - PAdES
      - Sample PAdES with appearance
    - XAdES
      - Sample XAdES counter signature
    - JAdES
      - Sample JAdES counter signature
    - ASIC-S
      - Sample Asic-s signature
    - ASIC-E
      - Sample Asic-s signature
    - A special case: the external signature
      - Sample
  - Advanced Electronic Signature (FEA) work units
  - Validation work units
    - Attached validation
      - Sample
    - Detached validation
      - Sample
    - Certificate validation
      - Sample
    - Transient validation
      - Sample
    - Remote validation
  - Timestamp work units
    - Result
    - TSD attached timestamp
      - Sample
    - TSR detached timestamp
      - Sample
  - Encryption work units
    - Result
    - Encryption
      - Sample
      - Sample
    - Decryption
      - Sample
      - Sample
- Utility methods
  - Device utilities
    - Sign Hash
      - Sample
    - Open session
    - Close session
    - Send credentials
      - FeaSessionCredentialInfo
      - FeaTransactionCredentialInfo

        - RemoteCredentialInfo
    - Device info
  - I/O utilities
    - Upload file
      - Rest
      - SOAP
    - Download file by ID
      - Rest
      - SOAP
    - Delete file by ID
- Error codes
- OID

General informations

ArubaTrustedPlatform (**ATP**) is an integrated web service that exposes methods for

- digital signature
- signature validation
- timestamp
- Advanced Electronic Signature (FEA)

and several utility methods related to devices, FEA processes and data upload/download.

Moreover ATP will support legacy interfaces of some deprecated web services (ARSS, VOL, ROSS, ...) which singularly offered the same features.

ATP supports both SOAP and REST interfaces.

Glossary of terms

Term	Description
WSDL	Web Services Description Language is an XML-based interface description language that is used for describing the functionality offered by a web service, usually related to SOAP API
WADL	Web Application Description Language is a machine-readable XML description of HTTP-based web services, usually related to REST API
ROSS	Legacy FEA web service
VOL	Legacy validator web service APIs
ARSS	Legacy digital signature web service
SOAP	Simple Object Access Protocol is an XML based a message specification for exchanging information between systems and applications.
REST	Representational State Transfer. Alternative architectural API style for applications interoperability
PKCS11	PKCS #11 is a Public-Key Cryptography Standards that defines a programming interface to create and manipulate cryptographic tokens and secret cryptographic keys.
HSM	Hardware Security Module is a physical computing device that safeguards and manages secrets (e.g. digital keys), and performs encryption and decryption functions for digital signatures, strong authentication and other cryptographic functions.
PKCS7	PKCS #7 (Cryptographic Message Syntax, CMS) is a standard syntax for storing signed and/or encrypted data.

General Notes

- Please note that samples included in this document are not intended to a real execution and may contains incomplete or invalid data.
  The sample purpose is to explain the API request/response structure only.

List of supported web methods

name	URI	method	http status codes
executeFlow	*executeflow*	POST	200, 401, 403, 404, 500
signHash	*sign_hash*	POST	200, 401, 500
deviceInfo	*device_info*	POST	200, 401, 500
openSession	*open_session*	POST	200, 401, 500

closeSession	*close_session*	POST	200, 401, 500
sendCredential	*send_credential*	POST	200, 401, 500
changePassword	*change_password*	POST	200, 401, 500
uploadFile	*upload_file*	POST	200, 401, 403, 404, 500
downloadFile	*download_file*	POST	200, 401, 403, 404, 500
deleteFile	*delete_file*	POST	200, 401, 403, 404, 500
extractInternalDocument	*extract_internal_document*	POST	200, 401, 500

Flow, Work Unit and Utilities logic

ATP implements three kinds of web methods:

- **executeFlow**, is the method to process a flow composed by a set of one or more work units and zero or more enabledDevices
- **work units**, are submethods part of a flow that combines in chain inputs and outputs of every single unit. They can't be accessed directly from interface, but are executed through the executeFlow.
  The final result of the chained process is returned as response. If the flow contains a static (logical) error or gets a runtime error (see Error codes) a global failure is returned instead.
- **other utilities**, are auxiliary methods that can be called to prepare a work flow or to get status information, etc.

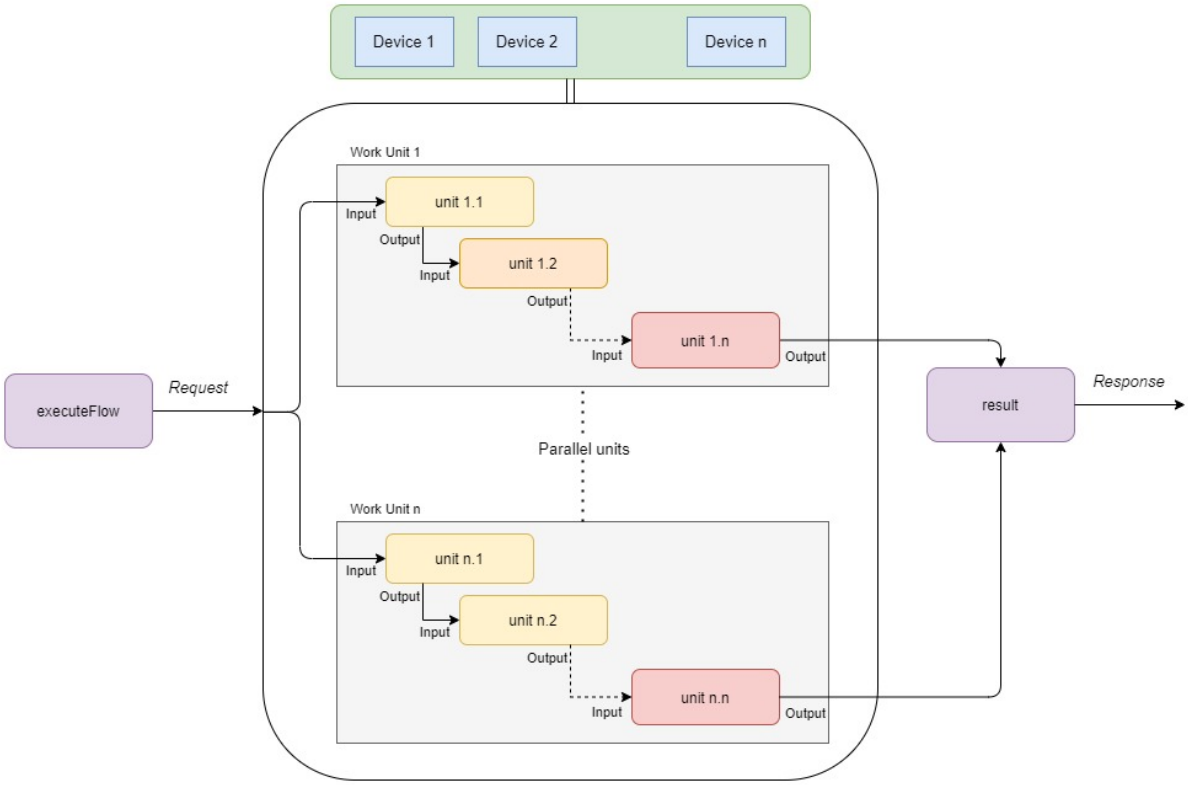
ExecuteFlow method

A flow is represented by *executeflow* method, that can include a single or multiple work units (that is work units in parallel).
The flow coherence and integrity is statically checked before the flow's units are executed.

Into a chained work unit flow the output resource of every "not final" unit become the input resource of the next one.
The ending unit can be final because

- is a "terminal" type, for instance validation units
- hasn't any next unit

and in this case the output resource (if present) is returned according with request indications (or using the same type of request's input resource if no indication is present).



Moreover *executeflow* admits some *enabledDevices*, that are devices globally available to all units during the whole process.
Note that not all work units requires a device (e.g. validation work units don't need).

A sample of REST *executeflow* could be the following, where two devices are available to a work unit composed by three subwork units in flow (cades signature cades signature (parallel) validation) and to a single pades signature work unit.

Flow Sample

```
{
        "enabledDevices": [
                {
            "type": "ATP",
            "deviceId": "device01",
            "identity": {
                "userid": "remote_signature_user@remote_signature_domain",
                "password" : "remote_signature_password"
            },
            "oneTimePassword" : "remote_signature_otp"
        },
                {
                        "type": "SOFTWARE",
                        "deviceId": "device02",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGSIb3DQEHAaCCDVsE..."
                        },
                        "pin": "device_pin"
                }
        ],
        "works": [
                {
                        "type": "CADES_SIGNATURE",
                        "deviceId": "device01",
                        "signatureLevel": "BES",
                        "input": {
                                "type": "BUFFERED",
                                "data": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw=="
                        },
                        "nextRequest": {
                                "type": "CADES_SIGNATURE",
                                "deviceId": "device02",
                                "signatureLevel": "BES",
                                "addSignaturePath": "P",
                                "nextRequest": {
                                        "type": "ATTACHED_VALIDATION",
                                        "validationProfile": "QUALIFIED_ELECTRONIC_SIGNATURE_PROFILE",
                                        "buildPDFReport": true,
                                        "nextRequest": null
                                }
                        }
                },
                {
                        "type": "PADES_SIGNATURE",
                        "deviceId": "device01",
                        "signatureLevel": "BES",
                        "input": {
                                "type": "BUFFERED",
                                "data": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw..."
                        }
                }
        ]
}
```

A executeFlow result is composed by a global result and an array of sub results, one for each work unit (for chained units only final result is reported).

Result


```
{
        "code": "0000",
        "message": "OK",
        "status": "OK",
        "results": [
                {
                        "type": "ATTACHED_VALIDATION",
                        "validatedDocument": {
                                ....
                        "signedFile": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw==",
                        "code": "0000",
                        "message": "OK",
                        "status": "OK",
                        "pdfReport": "JVBERi0xLjQKJfbk/N8KMSAwIG9iago8o=",
                        "htmlReport": null
                },
                {
                        "type": "CADES_SIGNATURE",
                        "code": "0000",
                        "message": "OK",
                        "status": "OK",
                        "output": {
                                "type": "BUFFERED",
                                "data": "MIAGCSqGSIb3DQEHAqCAMIACAQExDzA=="
                        },
                        "signingTime": null
                }
        ]
}
```


- the **code** is the result numerical code identifier (see the return error code)
- the **message** can be OK or can contain the error message
- the **status** is OK for success or KO for errors

Developer API docs

OpenApi documentation openapi.yaml

SOAP interface

The ATP SOAP WSDL is downloadable at

`<host>:<port>/ATPService?wsdl`

address and the xsd is available at

`<host>:<port>/ATPService?xsd=1`

The prefix url for any SOAP call is **https://<host>:<port>/ATPService**

REST interface

The ATP REST WADL is downloadable at

`<host>:<port>/api/v1/application.wadl`

address and the detailed version is available at

`<host>:<port>/api/v1/application.wadl?detail=true`

The prefix url for any REST call is **https://<host>:<port>/api/v1/atpservice**

Types and Constants

Resources

Any abstract data involved in a web method process (both in input and in output) is represented by a Resource:

BUFFERED, byte array resource

- "**data**": String<1>, Base64 encoded byte array

```
{
        "type": "BUFFERED",
        "data": "WW5WbVptVnlYMlJoZEdFPQ=="
}
```

BYID, file uploaded to ATP, identified by an ID

- "**resourceId**": String<1>, UUID of resource

```
{
        "type": "BYID",
        "resourceId": "237e9877-e79b-12d4-a765-321741963000"
}
```

CLOUD, resource hosted in a cloud space

Aruba

```
{
    "type": "CLOUD",
        "name": "",
        "bucket": "",
         "filePath": "",
         "endPoint": "",
         "secretKey": "",
          "userId": ""
}
```

Amazon

```
{
    "type": "CLOUD",
        "name": "",
        "bucket": "",
         "filePath": "",
         "secretKey": "",
          "userId": ""
}
```

URL, resource hosted in a HTTP web space

- "**url**": String<1>, the remote resource URL
- "**auth**": Authentication<0,1>, remote resource access data

```
{
        "type": "URL",
        "url": "https://endpoint1",
        "auth": {
                "username": "<username>",
                "password": "xxxxxx"
        }
}
```

FILE, file resource already present in the ATP file system, inside a specific shared folder which must be configured beforehand

- "**filePath**": String<1>, the relative path of the file inside the shared folder

```
{
        "type": "FILE",
        "filePath": "./file"
}
```

STREAMED, stream of data

```
{
        "type": "STREAMED",
        "data": "WW5WbVptVnlYMlJoZEdFPQ=="
}
```

Devices

The ATP interface provides some devices to describe a token.
The devices' common field is

"**deviceId**": String<1>, the device identifier, can be any string, used to select the device in the work

ATP, a remote signature token for HSM middleware

- "**identity**": RemoteHsmIdentity<1>, the remote signature account. "delegatedBy" is needed for automatic signature only
- "**url**": String<0,1>, the HSM middleware frontend URL, is optional (nullable)
- "**alternativeUrl**"; String<0,1>, is optional (nullable)
- "**oneTimePassword**": String<0,1>, the static or dynamic OTP to open signature session if needed
- "**sessionId**": String<0,1>, the signature session, if already opened
- "**applicationUser**": String<0,1>, signature Domain admin username
- "**applicationPassword**": String<0,1>, signature Domain admin password
- "**ignoreCache**": Boolean<0,1>, if true ignore user's cache

```
{
        "type": "ATP",
        "deviceId": "<deviceId>",
        "identity": {
                "userid": "user@domain",
                "password": "xxxxxx",
                "delegatedBy": {
                        "userid": "delegated@domain",
                        "password": "xxxxxx"
                }
        },
        "url": "https://endpoint1",
        "alternativeUrl": "https://endpoint2",
        "oneTimePassword": "xxxxxx"
}
```

SOFTWARE, a soft token keystore like p12 or pfx

- "**resource**": Resource<1>, the resource containing the keystore
- "**pin**": String<1>, the keystore password

```
{
        "type": "SOFTWARE",
        "deviceId": "<deviceId>",
        "resource": {
                "type": "BUFFERED",
                "data": "WW5WbVptVnlYMlJoZEdFPQ=="
        },
        "pin": "xxxxxx"
}
```

PKCS11, a generic pkcs11 token related to middleware

- "**slot_id**": Integer<1>, the numerical based zero index of the smartcard slot
- "**card_serial**": String<1>, serial number of the smartcard
- "**pin**": String<1>, the pin of the PKCS11 token

```
{
        "type": "PKCS11",
        "card_serial": "0004444778982",
        "slot_id": 0,
        "pin": "xxxxxx"
}
```

FEA, an Advanced Electronic Signature remote token

- "**auth**": Authentication<1>, FEA user credentials
- "**transactionId**": String<0,1>, identifier of the FEA signature's transaction
- "**sessionId**": String<0,1>, identifier of the FEA signature's session
- "**otp**": String<1>, FEA one time password

```
{
        "type": "FEA",
        "deviceId": "<deviceId>",
        "auth": {
                "username": "username",
                "password": "xxxxxx"
        },
        "transactionId": "237e9877-e79b-12d4-a765-321741963000",
        "otp": "xxxxxx",
}
```

```
{
        "type": "FEA",
        "deviceId": "<deviceId>",
        "auth": {
                "username": "username",
                "password": "xxxxxx"
        },
         "sessionId": "237e9877-e79b-12d4-a765-321741963000",
}
```

JWT, a device that accept a JWT instead of user and password authentication

- "jwt": String<1>, JWT for signature authentication, Base64 encoded
- "otp": String<0,1>, OTP for signature authorization
- "sessionId": String<0,1>, signature session's identifier

```
{
        "type": "JWT",
        "deviceId": "<deviceId>",
        "jwt": "WW5WbVptVnlY...",
        "otp": "xxxxxx",
}
```

ATP, JWT, PKCS11 and FEA are "sessioned" devices, that is they support a session that can be opened when the process starts and closed when it finishes.
The performances of any operation done when the session is open take advantage of status of the token, which is already initialized.

SignatureConstraint

A SignatureConstraint is a constraint that need to be validated before the application of the signature.

If a SignatureConstraint verification fail the WorkUnit return error **0043 - "Constraint type[value] validation failed"**, where type is the value of SignatureConstraint's type parameter and value a specific value from the constraint (es. oid for SubjectDnConstraint or IssuerDnConstraint)

SubjectDnConstraint, verify the presence of a specific value in subject DN (List of supported OID)

```
{
    "type": "SUBJECT_DN_CHECK",
    "oid": "2.5.4.5",
    "value": "TINIT-AAABBB80A11A111A"
}
```

IssuerDnConstraint, verify the presence of a specific value in issuer DN (List of supported OID)

```
{
    "type": "ISSUER_DN_CHECK",
    "oid": "2.5.4.3",
    "value": "Actalis EU Qualified Certificates CA G2"
}
```

Enums

Some string struct objects are used in requests/responses fields

- **SignatureLevel**
  - "BES", simple
  - "T", with timestamp included
  - "LT", with timestamp, certificate chain and certificate revocation responses included
  - "LTA", with timestamp, certificate chain, certificate revocation responses and seal timestamp included
- **SignatureMode**
  - "ENVELOPED", the signature is wrapped into the signed file
  - "ENVELOPING", the signature wraps the signed file
  - "DETACHED", the signature is detached from signed file (PKCS7)
- **PadesCertificationLevel**
  - "NOT_CERTIFIED", signature not certified, all is allowed to signed pdf
  - "CERTIFIED_SIGN_FORM_FILLING_AND_ANNOTATIONS", signature, form filling and annotations are allowed to signed pdf
  - "CERTIFIED_SIGN_FORM_FILLING", signature, form filling are allowed to signed pdf
  - "CERTIFIED_NO_CHANGES_ALLOWED", no change is allowed to signed pdf
- **PdfAppearanceResizeMode**
  - "RESIZE_MODE_ALL", resize both dimensions
  - "RESIZE_MODE_CENTER", resize best side and align to center
  - "RESIZE_MODE_R", resize best side and align to right
  - "RESIZE_MODE_L", resize best side and align to left
  - "RESIZE_MODE_NONE", no resize
- **SignatureAlgorithm**
  - "SHA256withRSAEncryption"
  - "SHA256withECDSA"
  - "SHA384withECDSA"
  - "SHA512withECDSA"
  - "SHA256withRSAANDMGF1"

    - "SHA384withRSAANDMGF1"
    - "SHA512withRSAANDMGF1"
- **ValidationProfile**
    - "QUALIFIED_ELECTRONIC_SIGNATURE_PROFILE", requires a validation at Qualified level
    - "CNS_ADVANCED_ELECTRONIC_SIGNATURE_PROFILE", requires a validation at CNS level
    - "ADVANCED_ELECTRONIC_SIGNATURE_PROFILE", requires a validation at FEA level
    - "GRAPHOMETRIC_SIGNATURE_PROFILE", requires a validation at graphometric signature level
- **SignedFileLevel**
    - "NONE", no signed file is returned
    - "FIRST", returns the first level signed file
    - "LAST", returns the most internal file
- **CredentialsType**
    - "SMS", auth via SMS
    - "ARUBACALL", auth via ARUBACALL
    - "CNS2"
- **AsicSignatureType**
    - "CADES"
    - "XADES"
    - "TIMESTAMP"
- **SignatureMethodType**
    - "NEW", apply new signature
    - "PARALLEL", add a parallel signature to an already signed file
    - "COUNTER", add a countersignature to another signature
    - "UPDATE", update signature level
- **JwsSignatureProfile**
    - "AGID_ID_AUTH_REST_02", profile specified in Linee Guida sull'interoperabilità tecnica | 5.4. [ID_AUTH_REST_02] Direct Trust con certificato X.509 su REST con unicità del token/messaggio (italia.it)
    - "AGID_INTEGRITY_REST_01", profile specified in Linee Guida sull'interoperabilità tecnica | 6.2. [INTEGRITY_REST_01] Integrità del payload messaggio REST (italia.it)
    - "ETSI_EN_119_182_1_v1_1_1", profile specified in TS 119 182-1 - V1.1.1 - Electronic Signatures and Infrastructures (ESI); JAdES digital signatures; Part 1: Building blocks and JAdES baseline signatures (etsi.org)
- **JwsSerializationType**
    - "COMPACT", Serialization defined in RFC 7515: JSON Web Signature (JWS) (rfc-editor.org)
    - "JSON_FLATTENED", Serialization defined in RFC 7515: JSON Web Signature (JWS) (rfc-editor.org)
    - "JSON", Serialization defined in RFC 7515: JSON Web Signature (JWS) (rfc-editor.org)
- **LanguagesCode**
    - "EN", English
    - "IT", Italian
- **MessageLevel**
    - "INFO"
    - "WARNING"
    - "ERROR"
    - "NONE"
- **SignatureFormat**
    - "CADES"
    - "PADES"
    - "XADES"
    - "JADES"
    - "TST"
- **ContainerFormat**
    - "SELF"
    - "ASIC"
- **PDFProfile**
    - "BASIC"
    - "PADESBES"
    - "PADESLTV"
- **XadesSignatureProfile**
    - "ETSI_EN_319_132_1_v1_1_1"
    - "ETSI_TS_103171_v2_1_1"

Encryption Enum

This string struct object is used in Encryption request

- **EncryptionAlgorithm**
    - *DES_EDE3_CBC*: "1.2.840.113549.3.7"
    - *RC2_CBC*: "1.2.840.113549.3.2"
    - *IDEA_CBC*: "1.3.6.1.4.1.188.7.1.1.2"
    - *CAST5_CBC*: "1.2.840.113533.7.66.10"
    - *AES128_CBC*: "2.16.840.1.101.3.4.1.2"
    - *AES192_CBC*: "2.16.840.1.101.3.4.1.22"
    - *AES256_CBC*: "2.16.840.1.101.3.4.1.42"
    - *CAMELLIA128_CBC*: "1.2.392.200011.61.1.1.1.2"

    - *CAMELLIA192_CBC*: "1.2.392.200011.61.1.1.1.3"
    - *CAMELLIA256_CBC*: "1.2.392.200011.61.1.1.1.4"

Other Types

There are some complex objects that identify a set of fields related to each other.

- **RemoteHsmIdentity**
    - "**userid**": String<1>, remote signature username
    - "**password**": String<1>, remote signature user password
    - "**delegatedBy**": RemoteHsmIdentity<0,1>, automatic signature delegated user
- **Authentication**
  Authentication data
    - **"username"**, String<1>
    - **"password"**, String<1>
- **PreparedSignatureData**
  support data for "Prepared signature" steps
    - "**signerCertificate**", byte[]<1>, the signer certificate encoded in Base64
    - "**hash**", byte[]<1>, hash to be signed encoded in Base64
    - "**signedHash**", byte[]<1>, the signed hash encoded in Base64
- **AtpTSAData**
  Credentials and data about TSA
    - "**url**", String<1>, TSA url address
    - "**username**", String<1>, TSA account username
    - "**password**", String<1>, TSA account password
    - "**policy**", String<0,1>, account policy
- **PdfSignatureAppearance**
  The appearance of a PAdES graphical signature
    - "**customImage**", Resource<0,1>, a custom image (JPG, PNG, GIF) for graphical signature
    - "**imageOnly**", Boolean<0,1>, if true signature will show only the image, default false
    - "**leftX**", Integer<0,1>, the bottom-left x pixel coordinate of signature's box (ignored if fieldName is used), default 0
    - "**leftY**", Integer<0,1>, the bottom-left y pixel coordinate of signature's box (ignored if fieldName is used), default 0
    - "**rightX**", Integer<0,1>, the top-right x pixel coordinate of signature's box (ignored if fieldName is used), default 0
    - "**rightY**", Integer<0,1>, the top-right y pixel coordinate of signature's box (ignored if fieldName is used), default 0
    - "**pageIndex**", Integer<0,1>, the page where signature is applied, starting from 1, default 1
    - "**customText**", String<0,1>, the custom signature label
    - "**scaleFont**", Boolean<0,1>, true if text must be scaled to fit the bounding box, default false
    - "**showDateTime**", Boolean<0,1>, true if the date-time must be inserted in the signature, default true
    - "**resizeMode**", PdfAppearanceResizeMode<0,1>, the mode to resize the image to fit the signature's bounding box, default RESIZE_MODE_CENTER
    - "**fontSize**", Integer<0,1>, the font size, default 8
    - "**pageToSign**", String<0,1>, the page where signature is applied, **FIRST** for page 1 or **LAST** for last page.
- **GraphometricInfo**
  The PAdES graphometric signature data
    - "**blob**", byte[]<1>
    - "**fingerPrint1**", byte[]<0,1>
    - "**fingerPrint2**", byte[]<0,1>
    - "**fingerPrintSigned**", byte[]<1>
    - "**userNotice**", String<0,1>
    - "**personalData**", String[]<0,1>
- **Info**
  Information for Fea signature
    - **"name"**: String<1>, name of the info entry
    - **"value"**: String<1>, value of the info entry
    - **"blobvalue"**: String<0,1>, binary data of the info entry
- **SignatureMethod**
  Signature's method to use
    - **"type"**, SignatureMethodType<1>, type of method to use, NEW for matrioska, PARALLEL for parallel signature, COUNTER for countersignature and UPDATE to update the level of a signature
    - **"path"**, String<0,1>, path of the signature to update or countersign, mandatory if "type" is COUNTER or UPDATE
- **InfoCertificate**
  Info about device's certificate
    - "**certificate**": byte[]<1>, certificate associated with the device, encoded Base64
    - "**alias**": String<1>, alias of the certificate
    - "**name**": String<1>, Certificate subject common name
    - "**chains**": byte[]<0,n>, certification chain of the certificate associated with the device, encoded Base64
    - "**notBefore**": String<1>, Certificate's validity start date dd/MM/yyyy HH:mm:ss zzz
    - "**notAfter**": String<1>, Certificate's validity end date dd/MM/yyyy HH:mm:ss zzz
- **JwsSignatureParameter**
  JAdES signature specific parameters
    - "**jwtId**": String<0,1>, custom id to put in signature's header
    - "**audience**": String<0,1>, info to put in signature's header
    - "**jwsSerializationType**": JwsSerializationType<0,1>, type of signature serialization, default JSON
    - "**signatureProfile**":JwsSignatureProfile<0,1>, specify the signature structure, default ETSI_EN_119_182_1_v1_1_1

- **FeaUserInfo**
  User information that will be stored in FEA certificate
  - "**name**": String<1>, name of the information
  - "**value**": String<0,1>, text value of the information, only one between value and blobvalue must be provided
  - "**blobvalue**": byte[]<0,1>, blob value of the information, only one between value and blobvalue must be provided
- **ValidatedDocument**
  Document validation result
  - "**overallOk**": Boolean<1>, true if this validation result does not contain error messages
  - "**status**": MessageLevel<1>, worst level found in messages
  - "**messages**": ValidationMessage<0,n>, list of validation messages
  - "**documentId**": String<1> validated document identifier
  - "**documentName**": String<1> name of the validated document
  - "**signed**": Boolean<1> true if the document contains at leas one signature
  - "**signers**": ValidatedSigner<0,n> list of validated signer
  - "**subResults**": ValidatedDocument<0,n> list of internal validated document
- **ValidatedCertificate**
  Certificate validation result
  - "**overallOk**": Boolean<1>,true if this validation result does not contain error messages
  - "**status**": MessageLevel<1>, worst level found in messages
  - "**messages**": ValidationMessage<0,n>, list of validation messages
- **ValidatedSigner**
  Signer validation result
  - "**overallOk**": Boolean<1>,true if this validation result does not contain error messages
  - "**status**": MessageLevel<1>, worst level found in messages
  - "**messages**": ValidationMessage<0,n>, list of validation messages
  - "**name**": String<1> name of the signer
  - "**certificate**": ValidatedCertificate<0,1> validation result of the certificate associated to this signer
  - "**signatureFormat**": SignatureFormat<1> format of the validated signature
  - "**containerFormat**": ContainerFormat<1> format of the validated signature container
  - "**signatureLevel**": SignatureLevel<1> Level of the validated signature
  - "**signatureDate**": String<0, 1> "dd/MM/yyyy hh:mm:ss UTC" Date of the signature
  - "**validationDate**": String<0, 1> "dd/MM/yyyy hh:mm:ss UTC" Date of the validation
  - "**docReferences**": DocumentReference<0,n> List of document referenced by this signature
  - "**counterSigners**": ValidatedSigner<0,n> List of validated counter signature of this signature
  - "**timeStamps**": ValidatedSigner<0,n> List of validated timestamps of this signature
- **ValidatorMessage**
  Validation message
  - "**level**": MessageLevel<1>, level of this message
  - "**code**": String<1>, identifier of this message
  - "**etsiCode**": String<0,1>ETSI identifier of the message, see ETSI EN 319 102-1
  - "**message**": String<1> human readable message
  - "**notes**": ValidationMessage<0,n>, list of validation sub messages
- **DocumentReference**
  Reference to a signed document
  - "**docId**": String<1>, identifier of the referenced document
  - "**messageDigestMatch**": Boolean<1> true if the digest of the referenced document match with the signature one
- **DictionarySignedAttributes**
  - "**T**": String<1> a Signature Dictionary's attribute

Date time convention

All date time in input and in output are strings formatted as "dd/MM/yyyy hh:mm:ss UTC", e.g. "22/10/2021 08:38:53 UTC"

Work Units

All following methods are part of a flow processed by executeFlow method.
Every work unit admits in input following fields

- "**nextRequest**": WorkUnit<0,1>, that can be null or another work unit in the flow.
- "**id**":<String><0,1>, an optional identifier to link every work unit to its result

Note: coherence checks of work unit connections are under service caller responsibility.

Note: every result includes an "extra" field, consisting in a Key-Value Map reserved for any future custom use

Signature work units

Signature work units can produce data signed in CAdES, PAdES, XAdES, JAdES format or signed containers (ASIC or ASICE).
Every signature unit has following common fields

- "**deviceId**": String<1>, the token device id in enabledDevices, mandatory

- "**output**": Resource<0, 1>, if null the output resource will be the same kind of input one (ignored for non final work unit)
- "**keyAlias**": String<0, 1>, if null is used the first alias in the token
- "**tsaData**": AtpTSAData<0, 1>, mandatory if signature level is not BES and TSA data are not defined into service configuration
- "**signatureLevel**": SignatureLevel<0,1>, BES default
- "**signatureMethod**": SignatureMethod<0, 1>, signature method to apply, NEW default
- "**signingTime**": String<0, 1> "dd/MM/yyyy hh:mm:ss UTC", if null the (server) system time will be used
- "**signatureAlgorithm**": SignatureAlgorithm<0, 1>, default SHA256withRSAEncryption
- "**generatePreparedSignature**": Boolean <0,1>, default false
- "**preparedSignatureData**": PreparedSignatureData<0,1>, mandatory if generatePreparedSignature is true
- "**constraints**": SignatureConstraint<0,n>, List of constraint to validate before the application of the signature

Each signature unit uses one device defined into the *enabledDevices* array, defined as follows

```
"enabledDevices": [
             {
            "type": "SOFTWARE",
            "deviceId": "device01",
            "resource": {
                "type": "BUFFERED",
                "data" : "MIINsQIBAzCCDWoGCSqGSIb3DQEHA..."
            },
            "pin": "xxxxxx"
        },
        {
            "type": "ATP",
            "deviceId": "device02",
            "identity": {
                "userid": "user@domain",
                "password" : "xxxxxx"
            },
            "url": "https://endpoint",
            "oneTimePassword" : "xxxxxx"
        },
        ...
    ]
```

At least one device must be present in the array.

Result

A typical signature result is

```
{
        "code": "0000",
        "message": "OK",
        "status": "OK",
        "results": [
                {
                        "type": "CADES_SIGNATURE",
                        "code": "0000",
                        "message": "OK",
                        "status": "OK",
                         "extra": null,
                        "id": null,
                        "output": {
                                "type": "BUFFERED",
                                "data": "MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABAA"
                        },
                        "signingTime": null
                }
        ]
}
```

CAdES

A CAdES signature has in addition following fields

- "**input**": Resource<1>, the input resource to be signed
- "**detached**": Boolean<0,1>, true if only pkcs7 is required, default false
- "**excludeSigningTime**": Boolean<0,1>, if true exclude signing time from signature
- "**returnDer**": Boolean<0,1>, if true return signed data in DER format
- "**originalDocumentData**": Resource<0,1> Original data to update detached signature

Sample CAdES-T

```
{
        "enabledDevices": [
                {
                        "type": "SOFTWARE",
                        "deviceId": "device01",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGS..."
                        },
                        "pin": "xxxxxx"
                }
        ],
        "works": [
                {
                        "type": "CADES_SIGNATURE",
                        "deviceId": "device01",
                        "tsaData": {
                                "url": "https://endpoint",
                                "username": "user",
                                "password": "password"
                        },
                        "signatureLevel": "T",
                        "input": {
                                "type": "BUFFERED",
                                "data": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw=="
                        }
                }
        ]
}
```

PAdES

A PAdES signature has in addition following fields

- "**input**": Resource<1>, the input resource to be signed
- "**ownerPassword**": String<0,1>, the PDF document password protection
- "**reason**": String<0,1>, the reason of the signature
- "**location**": String<0,1>, the phisical place where the signer is
- "**fieldName**": String<0,1> a preset field to be signed
- "**preservePdfA**": Boolean<0,1>, used together appearance, preserve or not the PDF/A specification features (avoid image transparency and include fonts), default false
- "**certificationLevel**": PdfCertificationLevel<0,1>, the certification level needed after signature, default NOT_CERTIFIED
- "**appearance**": PdfSignatureAppearance<0,1>, the graphical signature appearance
- "**graphometricInfo**": GraphometricInfo<0,1>, the graphometric data
- "**pdfProfile**": PDFProfile<0,1>, Pdf Signature Profile desired, default PADESBES
- "**dict_signed_attributes**": DictionarySignedAttributes<0,1>Possible signed attribute of Signature Dictionary, for Graphometric signature

Sample PAdES with appearance

```
{
    "enabledDevices": [{
            "type": "SOFTWARE",
            "deviceId": "device01",
            "resource": {
                "type": "BUFFERED",
                "data" : "MIINsQIBAzCCDWoGCSqG..."
            },
            "pin": "xxxxxx"
        }
    ],
    "works": [{
            "type": "PADES_SIGNATURE",
            "deviceId": "device01",
            "signatureLevel": "BES",
            "input": {
                "type": "BUFFERED",
                "data" : "JVBERi0xLjcNCiW1tbW1DQoxINCjk3MjUwDQolJUVPRg=="
            },
            "fieldName" : null,
            "preservePdfA": true,
            "appearance" : {
              "type" : "PADES_APPEARANCE",
                       "imageOnly" : false,
                       "leftX" : 10,
                       "leftY" : 15,
                       "rightX" : 100,
                       "rightY" : 120,
                       "pageIndex" : 2,
                       "scaleFont" : false,
                       "showDateTime" : true,
                       "resizeMode" : "RESIZE_MODE_CENTER",
                       "fontSize" : 8,
                       "customImage" : {
                         "type": "BUFFERED",
                         "data" : "iVBORw0KGgoAAAANSUhEUgAAADAAAAAwCAYAAABXAvmHAA5CYII="
                     }
                     }
        }
    ]
}
```

XAdES

A XAdES signature has in addition following fields

- "**input**": Resource<1>, the input resource to be signed
- "**mode**": SignatureMode<0,1>, which kind of signature should be applied, default ENVELOPING
- "**signatureProfile**": XadesSignatureProfile<0,1>, XAdES signature's profile to use, default ETSI_TS_103171_v2_1_1
- **"signerId"**: String<0,1>, signature id
- **"objectId"**: String<0,1>, id of the signed object
- "**originalDataInput**": Resource<0,1>, Original data to update detached signature

Sample XAdES counter signature

```
{
        "enabledDevices": [
                {
                        "type": "SOFTWARE",
                        "deviceId": "device01",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGSIb3DQEHAaCCDVsEgg1XMIINUMBhqA="
                        },
                        "pin": "xxxxx"
                }
        ],
        "works": [
                {
                        "type": "XADES_SIGNATURE",
                        "deviceId": "device01",
                        "signatureMethod": {
                                "type": "COUNTER",
                                "path": "0"
                        },
                        "input": {
                                "type": "BUFFERED",
                                "data": "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluzpTaWduYXR1cmU+"
                        }
                }
        ]
}
```

JAdES

A JAdES signature has in addition following fields

- "**input**": Resource<1>, the input resource to be signed
- "**signatureParameters**": JwsSignatureParameter<0,1> JAdES specific parameters
- "**detached**": Boolean<0,1>, if true the signature will not contain payload data
- "**originalDataInput**": Resource<0,1>, Original data to update detached signature

Sample JAdES counter signature

```
{
        "enabledDevices": [
                {
                        "type": "SOFTWARE",
                        "deviceId": "device01",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGSIb3DQEHAaCCDVsEgg1XMIINUMBhqA="
                        },
                        "pin": "xxxxx"
                }
        ],
        "works": [
                {
                        "type": "JADES_SIGNATURE",
                        "deviceId": "device01",
                        "signatureMethod": {
                                "type": "NEW"
                        },
                        "signatureParameters": {
                                "jwsSerializationType": "COMPACT"
                        }
                        "input": {
                                "type": "BUFFERED",
                                "data": "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluzpTaWduYXR1cmU+"
                        }
                }
        ]
}
```

ASIC-S

Asic-s signature has in addition following fields

- "**input**": Resource<1>, the input resource to be signed
- **"asicSignatureType"**: AsicSignatureType<0,1>, type of the internal signature

Sample Asic-s signature

```
{
        "enabledDevices": [
                {
                        "type": "SOFTWARE",
                        "deviceId": "device01",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGSIb3DQEHAaCCDVsEgg1XMIINUMBhqA="
                        },
                        "pin": "xxxxx"
                }
        ],
        "works": [
                {
                        "type": "ASICS_SIGNATURE",
                        "asicSignatureType": "CADES",
                        "deviceId": "device01",
                        "input": {
                                "type": "BUFFERED",
                                "data": "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluzpTaWduYXR1cmU+"
                        }
                }
        ]
}
```


ASIC-E

Asic-e signature has in addition following fields

- "**inputs**": Resource<1,n>, the input resources to be signed
- **"asicSignatureType"**: AsicSignatureType<0,1>, type of the internal signature

Sample Asic-s signature

```
{
        "enabledDevices": [
                {
                        "type": "SOFTWARE",
                        "deviceId": "device01",
                        "resource": {
                                "type": "BUFFERED",
                                "data": "MIINsQIBAzCCDWoGCSqGSIb3DQEHAaCCDVsEgg1XMIINUMBhqA="
                        },
                        "pin": "xxxxx"
                }
        ],
        "works": [
                {
                        "type": "ASICE_SIGNATURE",
                        "asicSignatureType": "CADES",
                        "deviceId": "device01",
                        "inputs": [
                                {
                                         "type": "BUFFERED",
                                        "data": "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluzpTaWduYXR1cmU+"
                                },
                                 {
                                         "type": "BUFFERED",
                                        "data": "ZGF0YSB0byBzaWdu"
                                }
                        ]
                }
        ]
}
```

A special case: the external signature

External signature is a special work unit to sign Resource with an external Device.
Is divided in two step, the first one prepare the hash to sign and the second one take signed hash and build the signed file.
Input Resource and work unit's type must be the same in both steps.

It work with CadesSignatureWorkunit, PadesSignatureWorkUnit and XadesSignatureWorkUnit by adding following fields:

- **generatePreparedSignature**: Boolean<1>, if true activate external signature
- **preparedSignatureData**: PreparedSignatureData<1>, contains signing certificate, hash and signed hash

Only for XadesSignatureWorkUnit :

- **"signerId"**: signature id
- **"objectId"**: id of the signed object

XAdES signature preparation returns signerId and objectId that are to be passed as input to the finalization request, otherwise the signature will fail.

enabledDevices field is ignored.

Sample

PrepareSignature

```
{
    "enabledDevices": [
    ],
    "works": [
        {
            "type": "CADES_SIGNATURE",
                        "signerID": "sID",
                         "objectID": "oID",
            "input": {
                "type": "BUFFERED",
                "data": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw=="
            },
            "generatePreparedSignature": true,
            "preparedSignatureData": {
                "signerCertificate": "YXJ1YmE..."
            }
        }
    ]
}
```

FinalizeSignature

```
{
    "enabledDevices": [
    ],
    "works": [
        {
            "type": "CADES_SIGNATURE",
                        "signerID": "sID",
                         "objectID": "oID",
                        "input": {
                "type": "BUFFERED",
                "data": "YXJ1YmEgcHJvYw0KYXJ1YmEgcHJvYw=="
            },
            "generatePreparedSignature": true,
            "preparedSignatureData": {
                "signerCertificate": "YXJ1YmE...",
                "hash": "YXJ1YmE...",
                "signedHash": "YXJ1YmE..."
            }
        }
    ]
}
```

Advanced Electronic Signature (FEA) work units

Fea signature work unit is a special type of Signature work unit, it contains all the fields from signature work unit and add some specific ones:

- **"numberOfSignatures"**: Long, number of max signature in this transaction
- **"extraInfo"**: Info<0,n>, List of Objects that contains the user information that will be store in the corresponding certificate
- **"input"**: Resource<1>, Pdf document to sign
- **"transactionInfo"**: Info<0,n>, List of Object that contains the transaction information
- **"appereance":** PdfSignatureAppearance<0,1>, Object that contains the parameters to append a visibile sign in the PDF
- **"reason"**: String<0,1>, the reason of the signature
- **"location"**: String<0,1>, the physical place where the signer is
- **"preservePdfA"**: Boolean<0,1>, used together with appearance, preserve or not the PDF/A specification features (avoid image transparency and include fonts), default false
- "**fieldName**": String<0,1> a preset field to be signed

To request a FEA signature, send credentials must be called before with one of **FeaSessionCredentialInfo** or **FeaTransactionCredentialInfo**.

```
{
        "enabledDevices": [
                {
                        "type": "FEA",
                        "deviceId": "device01",
                        "auth": {
                                "username": "username",
                                "password": "xxxxxx"
                        },
                        "transactionid": "xxxxxxxx",
                        "sessionId": "xxxxxxxx",
                        "otp": "xxxxxx",
                }
        ],
        "works": [
                {
                        "type": "FEA_SIGNATURE",
                        "deviceId": "device01",
                        "numberofsignatures": 1,
                        "extraInfo": [
                                {
                                        "name": "NAME",
                                         "value": "user"
                                },
                                {
                                        "name": "FISCAL_CODE",
                                        "value": "AAAAAA00A00A000A"
                                }
                        ]
                        "input": {
                                "type": "BUFFERED",
                                "data": "PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluzpTaWduYXR1cmU+"
                        }
                }
        ]
}
```

Validation work units

Validation work unit produces a validation result and optionally returns a PDF human readable report and the internal "original" file.

All validation units have following common fields:

- "**verifyAtDate**": String<0, 1> "dd/MM/yyyy hh:mm:ss UTC", the date-time when the validation must be applied. If null the system time will be used
- "**validationProfile**": ValidationProfile<0,1>, the level of validation, default "QUALIFIED_ELECTRONIC_SIGNATURE_PROFILE"
- "**buildPDFReport**": Boolean<0,1>, if the PDF report must be returned, default false
- **"returnCertificates"**: Boolean<0,1>, if the certificate in base64 report must be returned, default false
- "**language**": LanguagesCode<0, 1>, language to be used for response's verification messages, default EN (English)
- "**inputDataPassword**": String<0, 1>, optional password to open input PDF resource

all validation result share the following result parameter:

- "**pdfReport**": BufferedResource<0,1> validation pdf report Base64 encoded

and for all units the result is similar to following sample

Validation result Sample

```
{
        "code": "0000",
        "message": "OK",
        "status": "OK",
        "results": [
                {
```

```
                        "type": "ATTACHED_VALIDATION",
                        "validatedDocument": {
                                "documentId": "5041d7de6c2721e67bb142476f9d0483fa2d899a",
                                "signed": true,
                                "signers": [
                                        {
                                                "name": "MonacelliTestIncard 02 2016 MarcoTestIncard 02
2016",
                                                "signatureFormat": "CADES",
                                                "containerFormat": "SELF",
                                                "signatureLevel": "T",
                                                "signatureDate": "10/05/2016 10:57:37 UTC",
                                                "validationDate": {
                                                        "date": "2016-05-10T10:57:38.000+00:00",
                                                        "dateSource": "TIMESTAMP"
                                                },
                                                "certificate": {
                                                        "serial": "29d60ffcc8e43a85394f4fd90954948c",
                                                        "certificateName": "MonacelliTestIncard 02 2016
MarcoTestIncard 02  2016",
                                                        "subject": "DNQ=15374967,
SURNAME=MonacelliTestIncard 02 2016,GIVENNAME=\"MarcoTestIncard 02  2016\",SERIALNUMBER=IT:AAAAAA00A00A000A,
CN=\"MonacelliTestIncard 02 2016 MarcoTestIncard 02  2016\",O=non presente,C=IT",
                                                        "issuer": {
                                                                "serial":
"6cad805e30383cc586f31fab2f6e95f7",
                                                                "certificateName": "ArubaPEC S.p.A. NG CA
3",
                                                                "subject": "CN=ArubaPEC S.p.A. NG CA 3,
OU=Certification AuthorityC,O=ArubaPEC S.p.A.,C=IT",
                                                                "issuer": null,
                                                                "issuerName": "CN=ArubaPEC S.p.A. NG CA 3,
OU=Certification AuthorityC,O=ArubaPEC S.p.A.,C=IT",
                                                                "notBefore": "22/10/2010 00:00:00 UTC",
                                                                "notAfter": "22/10/2030 23:59:59 UTC",
                                                                "rsVerified": false,
                                                                "certStatus": {
                                                                        "valid": true,
                                                                        "corrupted": false,
                                                                        "timeValid": true,
                                                                        "trusted": true,
                                                                        "revoked": false,
                                                                        "invalidityReason": "",
                                                                        "revocationDate": null,
                                                                        "revocationReason": "",
                                                                        "revocationStatus": "NO_CHECK",
                                                                        "revocationVerificationType": ""
                                                                },
                                                                "certificate": "MIIHJDCCBQyg...=",
                                                                "overallOk": true,
                                                                "status": "INFO",
                                                                "messages": [
                                                                        {
                                                                                "level": "INFO",
                                                                                "code": "IC0002",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message": "The certificate
is reliable",
                                                                                "notes": [
                                                                                        {
                                                                                                "level":
"INFO",
                                                                                                "code":
"IC0011",
                                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                                "message":
"Verify at system date: 30/11/21, 16:15",
```

```
                                                                        "notes":
null
                                                                },
                                                                {
                                                                        "level":
"INFO",
                                                                        "code":
"IC0017",
                                                                        "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                        "message":
"Certificate validation done with OCSP",
                                                                        "notes":
null
                                                                }
                                                        ]
                                                }
                                        ],
                                        "extra": null
                                },
                                "issuerName": "CN=ArubaPEC S.p.A. NG CA 3,
OU=Certification AuthorityC,O=ArubaPEC S.p.A.,C=IT",
                                "notBefore": "22/03/2016 00:00:00 UTC",
                                "notAfter": "22/03/2019 23:59:59 UTC",
                                "rsVerified": false,
                                "certStatus": {
                                        "valid": true,
                                        "corrupted": false,
                                        "timeValid": true,
                                        "trusted": true,
                                        "revoked": false,
                                        "invalidityReason": "",
                                        "revocationDate": null,
                                        "revocationReason": "",
                                        "revocationStatus": "VALID",
                                        "revocationVerificationType": "OCSP"
                                },
                                "certificate": "MIIHJDCCBQyg...=",
                                "overallOk": true,
                                "status": "INFO",
                                "messages": [
                                        {
                                                "level": "INFO",
                                                "code": "IC0002",
                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                "message": "The certificate is
reliable",
                                                "notes": [
                                                        {
                                                                "level": "INFO",
                                                                "code": "IC0013",
                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                "message": "Date-
time of signature certified by timestamp: 10/05/16, 12:57",
                                                                "notes": null
                                                        },
                                                        {
                                                                "level": "INFO",
                                                                "code": "IC0017",
                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                "message":
"Certificate validation done with OCSP",
                                                                "notes": null
                                                        }
                                                ]
                                        },
```

```
                                                        {
                                                                "level": "INFO",
                                                                "code": "IC0001",
                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                "message": "The certificate has
legal validity",
                                                                "notes": [
                                                                        {
                                                                                "level": "INFO",
                                                                                "code": "IC0003",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message":
"Qualified Certificate compliant with Regulation (EU) No. 910/2014 - eIDAS",
                                                                                "notes": null
                                                                        },
                                                                        {
                                                                                "level": "INFO",
                                                                                "code": "IC0005",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message":
"Retention period of certification information: 20 years",
                                                                                "notes": null
                                                                        },
                                                                        {
                                                                                "level": "INFO",
                                                                                "code": "IC0006",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message": "The
private key associated with the certificate resides in a secure device compliant with the Regulation (EU)
No. 910/2014 (QSCD - Qualified Signature/Seal Creation Device)",
                                                                                "notes": null
                                                                        }
                                                                ]
                                                        }
                                                ],
                                                "extra": null
                                        },
                                        "docReferences": [
                                                {
                                                        "docId":
"3d919875d802904475723710d126d6a8bb1cecf8_sub_287602708023320964_",
                                                        "messageDigestMatch": true
                                                }
                                        ],
                                        "counterSigners": null,
                                        "timeStamps": [
                                                {
                                                        "name": "ArubaPEC Time Stamping Authority A
NG 028",
                                                        "signatureFormat": "TST",
                                                        "containerFormat": "SELF",
                                                        "signatureLevel": "BES",
                                                        "signatureDate": "10/05/2016 10:57:38 UTC",
                                                        "validationDate": {
                                                                "date": "2016-05-10T10:57:38.000+00:
00",
                                                                "dateSource": "TIMESTAMP"
                                                        },
                                                        "certificate": {
                                                                "serial":
"2c2aee6552369ccf44dac5b53cbfcbca",
                                                                "certificateName": "ArubaPEC Time
Stamping Authority A NG 028",
                                                                "subject": "CN=ArubaPEC Time
Stamping Authority A NG 028,O=ArubaPEC S.p.A.,C=IT",
```

```
                                                                    "issuer": {
                                                                            "serial":
"0613c15685d349ab2c60258bf4dfa162",
                                                                            "certificateName":
"ArubaPEC S.p.A. NG TSA 2",
                                                                            "subject": "CN=ArubaPEC S.p.
A. NG TSA 2,OU=Time Stamping AuthorityB,O=ArubaPEC S.p.A.,C=IT",
                                                                            "issuer": null,
                                                                            "issuerName": "CN=ArubaPEC
S.p.A. NG TSA 2,OU=Time Stamping AuthorityB,O=ArubaPEC S.p.A.,C=IT",
                                                                            "notBefore": "05/08/2008 00:
00:00 UTC",
                                                                            "notAfter": "04/08/2028 23:
59:59 UTC",
                                                                            "rsVerified": false,
                                                                            "certStatus": {
                                                                                    "valid": true,
                                                                                    "corrupted": false,
                                                                                    "timeValid": true,
                                                                                    "trusted": true,
                                                                                    "revoked": false,
                                                                                    "invalidityReason":
"",
                                                                                    "revocationDate":
null,
                                                                                    "revocationReason":
"",
                                                                                    "revocationStatus":
"NO_CHECK",
                                                                            "revocationVerificationType": ""
                                                                            },
                                                                            "certificate":
"MIIHJDCCBQyg...=",
                                                                            "overallOk": true,
                                                                            "status": "INFO",
                                                                            "messages": [
                                                                                    {
                                                                                            "level":
"INFO",
                                                                                            "code":
"IC0002",
                                                                                            "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                            "message":
"The certificate is reliable",
                                                                                            "notes": [
                                                                                                    {
"level": "INFO",
"code": "IC0011",
"etsiCode": "NONE_OR_OTHER_NOT_MAPPED",
"message": "Verify at system date: 30/11/21, 16:15",
"notes": null
                                                                                                    },
                                                                                                    {
"level": "INFO",
"code": "IC0017",
"etsiCode": "NONE_OR_OTHER_NOT_MAPPED",
"message": "Certificate validation done with OCSP",
```

```
"notes": null
                                                                                                                }
                                                                                                        ]
                                                                                                }
                                                                                        ],
                                                                                        "extra": null
                                                                                },
                                                                                "issuerName": "CN=ArubaPEC S.p.A.
NG TSA 2,OU=Time Stamping AuthorityB,O=ArubaPEC S.p.A.,C=IT",
                                                                                "notBefore": "06/05/2016 00:00:00
UTC",
                                                                                "notAfter": "04/05/2026 23:59:59
UTC",
                                                                                "rsVerified": false,
                                                                                "certStatus": {
                                                                                        "valid": false,
                                                                                        "corrupted": false,
                                                                                        "timeValid": true,
                                                                                        "trusted": true,
                                                                                        "revoked": false,
                                                                                        "invalidityReason": "",
                                                                                        "revocationDate": null,
                                                                                        "revocationReason": "",
                                                                                        "revocationStatus": "VALID",

"revocationVerificationType": "CRL"

},
                                        "certificate": "MIIHJDCCBQyg...=",
                                                                                "overallOk": true,
                                                                                "status": "INFO",
                                                                                "messages": [
                                                                                        {
                                                                                                "level": "INFO",
                                                                                                "code": "IC0002",
                                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                                "message": "The
certificate is reliable",
                                                                                                "notes": [
                                                                                                        {

"level": "INFO",

"code": "IC0013",

"etsiCode": "NONE_OR_OTHER_NOT_MAPPED",

"message": "Date-time of signature certified by timestamp: 10/05/16, 12:57",

"notes": null
                                                                                                        },
                                                                                                        {

"level": "INFO",

"code": "IC0018",

"etsiCode": "NONE_OR_OTHER_NOT_MAPPED",

"message": "Certificate validation done with CRL",

"notes": null
                                                                                                        }
                                                                                                ]
                                                                                        }
                                                                                ],
                                                                                "extra": null
                                                                        },
```

```
"docReferences": null,
"counterSigners": null,
"timeStamps": null,
"overallOk": true,
"status": "INFO",
"messages": [
        {
                "level": "INFO",
                "code": "IS0006",
                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                "message": "Timestamp is
valid",
                "notes": [
                        {
                                "level":
"INFO",
                                "code":
"IS0002",
                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                "message":
"Timestamp format TST",
                                "notes":
null
                        },
                        {
                                "level":
"INFO",
                                "code":
"IS0004",
                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                "message":
"The timestamp signature is intact",
                                "notes":
null
                        }
                ]
        },
        {
                "level": "NONE",
                "code": "IS0013",
                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                "message": "Timestamp
details",
                "notes": [
                        {
                                "level":
"INFO",
                                "code":
"IS0007",
                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                "message":
"Timestamp issued on 10/05/2016 10:57:38 UTC",
                                "notes":
null
                        },
                        {
                                "level":
"INFO",
                                "code":
"IS0008",
                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                "message":
"Policy Id: 1.3.6.1.4.1.29741.1.1.6",
```

```
                                                                                "notes":
null
                                                                        },
                                                                        {
                                                                                "level":
"INFO",
                                                                                "code":
"IS0009",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message":
"Serial number: 354241383637",
                                                                                "notes":
null
                                                                        },
                                                                        {
                                                                                "level":
"INFO",
                                                                                "code":
"IS0010",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message":
"Hash algorithm: SHA256",
                                                                                "notes":
null
                                                                        },
                                                                        {
                                                                                "level":
"WARNING",
                                                                                "code":
"WS0010",
                                                                                "etsiCode":
"SIG_CONSTRAINTS_FAILURE",
                                                                                "message":
"eIDAS compliance : Not qualified (under EU Regulation 910/2014 – eIDAS)",
                                                                                "notes":
null
                                                                        },
                                                                        {
                                                                                "level":
"WARNING",
                                                                                "code":
"WS0011",
                                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                                "message":
"Validity : National validity only",
                                                                                "notes":
null
                                                                        }
                                                                ]
                                                        }
                                                ],
                                                "extra": null
                                        }
                                ],
                                "overallOk": true,
                                "status": "INFO",
                                "messages": [
                                        {
                                                "level": "INFO",
                                                "code": "IS0005",
                                                "etsiCode": "NONE_OR_OTHER_NOT_MAPPED",
                                                "message": "Signature is intact",
                                                "notes": [
                                                        {
                                                                "level": "INFO",
                                                                "code": "IS0001",
```

```
                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                "message": "The signature
format is CADES-T",
                                                                "notes": null
                                                        },
                                                        {
                                                                "level": "INFO",
                                                                "code": "IS0003",
                                                                "etsiCode":
"NONE_OR_OTHER_NOT_MAPPED",
                                                                "message": "The signature
is intact",
                                                                "notes": null
                                                        }
                                                ]
                                        }
                                ],
                                "extra": null
                        }
                ],
                "subResults": [
                        {
                                "documentId":
"3d919875d802904475723710d126d6a8bb1cecf8_sub_287602708023320964_",
                                "signed": false,
                                "signers": null,
                                "subResults": null,
                                "overallOk": false,
                                "status": "INFO",
                                "messages": null,
                                "extra": null
                        }
                ],
                "overallOk": true,
                "status": "INFO",
                "messages": null,
                "extra": null
        },
        "signedFile": "PD9...9zZXJ2aWNlPg0KPC9kZWZpbml0aW9ucz4=",
        "code": "0000",
        "message": "OK",
        "status": "OK",
         "extra": null,
        "id": null,
        "pdfReport": null,
        "htmlReport": null
    }
  ]
}
```


Attached validation

An attached validation gets in input a signed file.

- "**input**": Resource<1>, the file to be validated
- "**signedFileLevel**": SignedFileLevel<0,1>, set the behaviour for unwrapping "signed" file to be returned with validation result, default "LAST"
- "**signedFileOutput**": Resource<0,1>, the output Resource where to save the original file, default is BufferedResource.
- "**failOnDetached**": Boolean<0,1>, if true validation is aborted immediately if the input document is detached

Sample

```
{
        "works": [
                {
                        "type": "ATTACHED_VALIDATION",
                        "validationDate": "22/10/2021 08:38:53 UTC",
                        "validationProfile": "QUALIFIED_ELECTRONIC_SIGNATURE_PROFILE",
                        "buildPDFReport": false,
                        "input": {
                                "type": "BUFFERED",
                                "data": "WW5WbVptVnlYMlJoZEdFPQ=="
                        },
            "signedFileOutput": {
                "type": "FILE",
                "filePath": "path/to/file.out"
            }
                }
        ]
}
```

Attached validation result has the following parameter:

- "**validatedDocument**": ValidatedDocument<1>
- "**signedFile**": Resource<0,1>, extracted origial file, if requested

Detached validation

A validation of a detached signature (PKCS7 and signed file)

- "**detachedEnvelope**": Resource<1>, the signature data
- "**envelopedData**": Resource<1>, the signed data

Sample

```
{
        "works": [
                {
                        "type": "DETACHED_VALIDATION",
                        "validationProfile": "ADVANCED_ELECTRONIC_SIGNATURE_PROFILE",
                        "buildPDFReport": true,
                        "detachedEnvelope": {
                                "type": "BUFFERED",
                                "data": "WW5WbVptVnlYMlJoZEdFPQ=="
                        },
                        "envelopedData": {
                                "type": "BUFFERED",
                                "data": "WW5WbVptVnlYMlJoZEdFPQ=="
                        }
                }
        ]
}
```

Detached validation result has the following parameter:

- "**validatedDocument**": ValidatedDocument<1>

Certificate validation

A signer certificate validation

- "**certificate**": byte[]<1>, the X509Certificate to be validated, encoded Base64

Sample

```
{
        "works": [
                {
                        "type": "CERTIFICATE_VALIDATION",
                        "validationDate": "22/10/2021 08:38:53 UTC",
                        "validationProfile": "CNS_ADVANCED_ELECTRONIC_SIGNATURE_PROFILE",
                        "buildPDFReport": false,
                        "certificate": "WW5WbVptVnlYMlJoZEdFPQ=="
                }
        ]
}
```


Certificate validation result has the following parameter:

- "**validatedCertificate**": ValidatedCertificate<1>

Transient validation

A transient document is a specific XML document that stores all details needed by validation process, but not the original content.
The features extraction from signed file to the transient document is a client side process.

Note: for further information about Transient document see specific documentation.

- "**transientDocument**": TransientDocument<1>, the transient document data

Sample

```
{
        "works": [
                {
                        "type": "TRANSIENT_VALIDATION",
                        "validationProfile": "QUALIFIED_ELECTRONIC_SIGNATURE_PROFILE",
                        "buildPDFReport": true,
                        "transientDocument": {
                                "id": "563ed6972cd181298b726f32f8809057c6fbb632_signed.txt.p7m",
                                "shortName": "signed.txt.p7m",
                                "signed": true,
                                "refsDocs": [
                                        {
                                                "id":
"1e4e888ac66f8dd41e00c5a7ac36a32a9950d271_sub_6192516265362621616_signed.txt",
                                                "shortName": "sub_6192516265362621616_signed.txt",
                                                "signed": false,
                                                "refsDocs": [],
                                                "certs": [],
                                                "signerInfo": [],
                                                "crls": [],
                                                "ocspresps": []
                                        }
                                ],
                                "certs": [
                                        {
                                                "id":
"6eb9d942c8776ad7c3fda8683de85a1c4e4b1ba245f7fb78f9927036304f41e2",
                                                "content": "MIIEoDCCA4igAwIBAgIQWe/1Sx/HWmqQS+by.....MHyA"
                                        }
                                ],
                                "signerInfo": [
                                        {
                                                "format": "CADES",
                                                "containerFormat": "SELF",
                                                "refsCertId":
"6eb9d942c8776ad7c3fda8683de85a1c4e4b1ba245f7fb78f9927036304f41e2",
                                                "refDocs": [
                                                        {
                                                                "id":
"1e4e888ac66f8dd41e00c5a7ac36a32a9950d271_sub_6192516265362621616_signed.txt",
                                                                "messageDigestMatch": true
                                                        }
                                                ],
                                                "counterSignaturesInformation": [],
                                                "timeStampTokens": [],
                                                "corrupted": false,
                                                "containsGraphometricInfo": false,
                                                "signatureInfo": "MIICWQIBATCBgDBsMQ......kY=",
                                                "padesSpecific": null,
                                                "signerName": "Risurname002 Rinname002",
                                                "signatureDate": "21/09/2016 11:06:44 UTC",
                                                "counterSignature": false,
                                                "tstSpecific": null
                                        }
                                ],
                                "crls": [],
                                "ocspresps": []
                        }
                }
        ]
}
```


Transient validation result has the following parameter:

- "**validatedDocument**": ValidatedDocument<1>

Remote validation

Remote validation is a special case of validation works, allows verification to be performed on a remote ATP, without sending the input data.

Input data parsing is done on local ATP, then signature's information are sent to a remote ATP for verification, this way local ATP can do everything without connecting to remote services such as CRL, OCSP, TSL and the like, the only external request is made to the remote ATP.

To enable this functionality, the ATP must be configured with the remote verification endpoint.

After enabling this mode, all verification requests will be performed remotely.

Timestamp work units

Timestamps work units have following common fields:

- "**tsaData**": AtpTSAData<0,1>, the TSA info. They are mandatory if TSA data are not defined into service configuration
- "**input**": Resource<1>, the resource to be timestamped
- "**output**": Resource<0,1>, the timestamp

Result

The time stamp process result is like following

```
{
    "code": "0000",
    "message": "OK",
    "status": "OK",
    "results": [{
            "type": "TSD_TIMESTAMP",
            "code": "0000",
            "message": "OK",
            "status": "OK",
            "output": {
                "type": "BUFFERED",
                "data": "MIILlAYLKoZIhvcNAQkQAR+........6pNPNvOOw=="
            }
        }
    ]
}
```

TSD attached timestamp

An "attached" timestamp (timestamp together the original file) is done as following

Sample

```
{
        "type": "TSD_TIMESTAMP",
        "tsaData": {
                "url": "https://endpoint1",
                "username": "<username>",
                "password": "xxxxxx",
                "policy": ""
        },
        "input": {
                "type": "FILE",
                "filePath": "./file"
        },
        "output": {
                "type": "FILE",
                "filePath": "./file"
        }
}
```

TSR detached timestamp

A detached timestamp (timestamp only) is done as following

Sample

```
{
        "type": "TSR_TIMESTAMP",
        "tsaData": {
                "url": "https://endpoint1",
                "username": "<username>",
                "password": "xxxxxx",
                "policy": ""
        },
        "input": {
                "type": "FILE",
                "filePath": "./file"
        },
        "output": {
                "type": "BUFFERED",
                "data": "WW5WbVptVnlYMlJoZEdFPQ=="
        }
}
```

Encryption work units

Encryption/Decryption work units have following common fields:

- "**type**": *ENCRYPTION* or *DECRYPTION* based on the operation to be performed
- "**input**": Resource<1>, the resource to be timestamped
- "**output**": Resource<0,1>, the timestamp

Result

Encryption/Decryption process result is like following (The wording changes depending on the operation used)

```
{
    "code": "0000",
    "message": "OK",
    "status": "OK",
    "results": [
        {
            "type": "DECRYPT_ENVELOPE",
            "code": "0000",
            "message": "OK",
            "status": "OK",
            "extra": null,
            "id": null,
            "output": {
                "type": "BUFFERED",
                "data": "MIILlAYLKoZIhvcNAQkQAR+........6pNPNvOOw=="
            }
        }
    ]
}
```

Encryption

Encryption has in addition following fields:

- "**recipients**": Byte[]<1,n> list of X509Certificates to use for encryption, Base64 encoded
- "**algorithm**": EncryptionAlgorithm<1>, algorithm to use for encryption

An Encryption *BUFFERED* is done as following.

Sample

```
{
    "works": [
        {
            "type": "ENCRYPTION",
                       "algorithm": "AES256_CBC",
            "recipients": [
                "MIIFg...Nu4iU=",
                "DVRAv...Vf6bU="
            ],
            "input": {
                "type": "BUFFERED",
                "data": "MIILlAYLKoZIhvcNAQkQAR+........6pNPNvOOw=="
            }
        }
    ]
}
```

An Encryption *FILE* is done as following

Sample

```
{
    "works": [
        {
            "type": "ENCRYPTION",
                       "algorithm": "AES256_CBC",
            "recipients": [
                "MIIFg...Nu4iU=",
                "DVRAv...Vf6bU="
            ],
            "input": {
                "type": "FILE",
                "filePath": "path/to/file"
            }
        }
    ]
}
```

Decryption

Encryption has in addition following fields:

- "**deviceId**": String<1>, id of the device to use for decryption

A Decryption *BUFFERED* is done as following

Sample

```
{
    "enabledDevices": [
        {
            "deviceId": "1",
            "type": "SOFTWARE",
            "resource": {
                "type": "BUFFERED",
                "data": "DVRAv...Vf6bU="
            },
            "pin": "password"
        }
    ],
    "works": [
        {
            "type": "DECRYPTION",
                   "algorithm": "AES256_CBC",
            "deviceId": "1",
            "input": {
                "type": "BUFFERED",
                "data": "MIILlAYLKoZIhvcNAQkQAR+........6pNPNvOOw=="
            }
        }
    ]
}
```

A Decryption *FILE* is done as following

Sample

```
{
    "enabledDevices": [
        {
            "deviceId": "1",
            "type": "SOFTWARE",
            "resource": {
                "type": "FILE",
                "filePath": "absolutePath\\file.p12"
            },
            "pin": "password"
        }
    ],
    "works": [
        {
            "type": "DECRYPTION",
            "deviceId": "1",
            "input": {
                "type": "FILE",
                "filePath": "./file"
            }
        }
    ]
}
```

Utility methods

Utilities are extra work flow methods (before or after but not inside). They can get information, change the status of objects, etc.

There are utilities dedicated to

- devices

- device creation
- resource I/O

Device utilities

Every devices utilities take in input a Device (this is a single device, not the enabledDevices array reference).

- "**device**": Device<1>, the device for which the method is called.

Sign Hash

This utility applies a "raw" signature to a byte array hash. This is commonly done between the "Prepare" and "Finalize" signature steps, to implement an external signature (using a remote custom token).

- "**inputs**": byte[]<1,n>, array of hashes to be signed, encoded Base64
- "**keyAlias**": String<0,1>, if null is used the first alias in the Device
- "**signatureAlgorithm**": SignatureAlgorithm<0,1>, algorithm used to sign the hash

Sample

```
{
        "type": "HASH_SIGNATURE",
        "device": {
                "type": "PKCS11",
                "deviceId": "<deviceId>",
                "card_serial": "<card serial>",
                "slot_id": 0,
                "pin": "xxxxxx"
        },
        "inputs": [
                "WW5WbVptVnlYMlJoZEdFPQ==",
        "WW5WbVptVnlYMlJoZEdFPQ=="
        ],
        "keyAlias": "<signer alias>"
}
```

the result contains the

- "**outputs**": byte[]<1,n>, array of signed hash resources, in the same order of inputs.

```
{
        "code": "0000",
        "message": "OK",
        "status": "OK",
    "extra": null,
        "outputs": [
                "RV2Vs3Ug0BCDBeav....IUIm8Zl8nYCnWupr7Nx2lOHMNcw==",
                "RV2Vs3Ug0BCDBeav....IUIm8Zl8nYCnWupr7Nx2lOHMNcw=="
        ]
}
```

Open session

The utility is used to open a signature session for a device. It takes no additional parameters:

```
{
    "type": "OPEN_SESSION",
    "device": {
        "type": "ATP",
        "deviceId": "<deviceId>",
        "identity": {
            "userid": "user@domain",
            "password": "xxxxxx",
            "delegatedBy": {
                "userid": "delegated@domain",
                "password": "xxxxxx"
            }
        },
        "url": "https://endpoint1",
        "alternativeUrl": "https://endpoint2",
        "oneTimePassword": "xxxxxx"
    }
}
```

Close session

The utility is used to close a signature session of a device. It takes no additional parameters:

```
{
    "type": "CLOSE_SESSION",
    "device": {
        "type": "ATP",
        "deviceId": "<deviceId>",
        "identity": {
            "userid": "user@domain",
            "password": "xxxxxx",
            "delegatedBy": {
                "userid": "delegated@domain",
                "password": "xxxxxx"
            }
        },
        "url": "https://endpoint1",
        "alternativeUrl": "https://endpoint2",
                "sessionId": "467a5b0f-7fb7-4bc6-9145-d7aa341f7c2f"
    }
}
```

Send credentials

The utility is used to send secondary credentials (otp) to the user for FEA and Remote Signature.

- "**credentialInfo**": CredentialInfo object implementation, one of FeaSessionCredentialInfo, FeaTransactionCredentialInfo or RemoteCredentialInfo
- "**authtype**": CredentialsType<1> type of the credential to send, SMS, ARUBACALL

FeaSessionCredentialInfo

- "**sessionId**": String<1>, user session id
- "**certInfo**": FeaUserInfo<3,n> list of user information, NAME, SURNAME and FISCAL_CODE are mandatory
- "**authid**": String<1>, Authetication identifier
- "**auth**": Authentication<1> user credential

FeaTransactionCredentialInfo

- "**numberofsignatures**": Integer<0, 1>, number of signature to authorize in this transaction
- "**transactionid**": String<0,1>, Transaction identifier
- "**certInfo**": FeaUserInfo<3,n> list of user information, NAME, SURNAME and FISCAL_CODE are mandatory
- "**authid**": String<1>, Authetication identifier

- "**auth**": Authentication<1> user credential

RemoteCredentialInfo

- "**device**": Device<1>, Device of type ATP or JWT

RemoteCredential

```
{
        "authtype": "SMS",
        "credentialInfo": {
        "device": {
            "identity": {
                            "userid": "username@domain",
                            "password": "xxxxxx"
                    }
        }
        }
}
```

FeaTransaction

```
{
        "authtype": "SMS",
        "credentialInfo": {
                "numberofsignatures": 1,
                "transactionid": "",
                "certInfo": [
                        {
                                "name": "",
                                "value": "",
                                "blobvalue": ""
                        }
                ],
                "authid": "",
                "auth": {
                        "username": "feaUser",
                        "password": "xxxxxx"
                }
        }
}
```

FeaSession

```
{
        "authtype": "SMS",
        "credentialInfo": {
                "sessionId": "",
                "certInfo": [
                        {
                                "name": "",
                                "value": "",
                                "blobvalue": ""
                        }
                ],
                "authid": "",
                "auth": {
                        "username": "feaUser",
                        "password": "xxxxxx"
                }
        }
}
```

Device info

The device info utility return some information about current device token

- list of key aliases
- list of certificates
- the device status
- list of supported digest algorithms
- list of supported external authentication methods
- kind of token (sessioned or not)

It takes no additional parameters.

```
{
    "type": "DEVICE_INFO",
    "device": {
        "type": "ATP",
        "deviceId": "<deviceId>",
        "identity": {
            "userid": "user@domain",
            "password": "xxxxxx",
            "delegatedBy": {
                "userid": "delegated@domain",
                "password": "xxxxxx"
            }
        },
        "url": "https://endpoint1",
        "alternativeUrl": "https://endpoint2",
        "oneTimePassword": "xxxxxx"
    }
}
```

result includes

- "**supportedExtAuth**": CredentialsType<0,n>, list of extended authentication methods
- "**supportedSignatureAlgorithm**": SignatureAlgorithm<1,n>, list of supported digest algorithms
- "**certificates**": InfoCertificate<1,n>, list of certificates info
- "**deviceStatus**": String<1>, status of device
- "**sessioned**":Boolean<1>, type of token, default false
- "**automaticTimeStamp**":Boolean<1>, if this device has automatic timestamp enabled

```
{
        "code": "0000",
        "message": "OK",
        "status": "OK",
         "extra": null,
        "supportedExtAuth": null,
        "supportedSignatureAlgorithm": [
                "SHA256withRSAEncryption"
        ],
        "certificates": [
                {
                        "certificate": "MIIHEDCCBPigAwIBAgIQfu...qO0wNME+m4v8wQsVTiUHZb",
                        "alias": "THALES_6279_1044_2D3:47642615b7040ead86b9694c103c419b95d42e01-AS0",
                        "chains": [
                                "MIIHEDCCBPigAwIBAgIQfu...qO0wNME+m4v8wQsVTiUHZb"
                        ]
                }

        ],
        "deviceStatus": "OK",
        "sessioned": true
}
```


I/O utilities

I/O utilities let you upload, download and delete files in ATP filesystem.

Upload file

Upload file request send a document to ATP's filesystem and return a unique identifier that can be passed as BYIDResource in other services.

This service is different between Rest and SOAP.

Rest

To upload a file using Rest service the request ContentType must be multipart/form-data and the file itself must have "file" as key.

curl

```
curl --location --request POST 'http://hostname/api/v1/atpservice/upload_file' \
--form 'file=@"//file/to/upload.path"'
```

JavaScript

```
var data = new FormData();
data.append("file", <file_binary_data>, "//file/to/upload.path");

var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});

xhr.open("POST", "http://hostname/api/v1/atpservice/upload_file");

xhr.send(data);
```

SOAP

Soap request contains the following field

- **input:** DataHandler<1>, binary data to upload

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.atp.
esecurity/">
    <soapenv:Header/>
    <soapenv:Body>
        <ser:uploadFile>
            <uploadFileRequest ser:type="UploadFileRequest">
                <input>MIAGCSqGS...==</input>
            </uploadFileRequest>
        </ser:uploadFile>
    </soapenv:Body>
</soapenv:Envelope>
```

Download file by ID

This utility let you download a file from ATP's filesystem.

Download by id request contains following field:

- **resourceId**: String<1>, id of the document to download

```
{
    "type": "DOWNLOAD_FILE_BYID_IO",
    "resourceId": "467a5b0f-7fb7-4bc6-9145-d7aa341f7c2f"
}
```

As for the upload this service behaviour change for Rest and Soap.

Rest

The file is downloaded as an attachment to the response

SOAP

The file is returned as binary in the field

- **output:** DataHandler<1>, file data

Delete file by ID

This utility let you delete a file from ATP's filesystem.

Delete by Id request contains following field:

- **resourceId**: String<1>, id of the document to download

```
{
    "type": "DELETE_FILE_IO",
    "resourceId": "467a5b0f-7fb7-4bc6-9145-d7aa341f7c2f"
}
```

Error codes

Code	Message	Http status	Description
0000	OK	200	
0001	Generic error	500	

0002	Invalid empty request	500	
0003	Module not enabled: <module_name>	500	A specific ATP module (signature, validation etc) is not enable in configuration
0004	Device is required	500	A Device is required in the service's request
0005	Unhandled work unit: <work_unit_name>	500	The WorkUnit type is not recognized
0006	Unhandled exception: <exception_info>	500	
0007	Not implemented yet	500	The required operation is not implemented yet
0008	Undefined result	500	
0009	Method error: <method_name>	500	
0010	Some works got an error	500	
0011	Duplicated device id found	400	
0012	Prepared signature is not compatible with containers	400	
0013	Prepared signature is not compatible with workflow	400	
0014	No device matches with requested deviceId	400	
0015	Error creating token: <error_info>	500	Generic error on sign token creation
0016	Inconsistent signature format	400	The requested signature options are not compatible with current format, like requiring a detached and parallel signature
0017	Error creating keystore	500	Error during sign keystore associated with sign token
0018	Error handling inputs	400	Error while reading input data from resource
0019	Error handling output	400	Error while writing data to output resource
0020	Unsupported signature type	400	
0021	Unsupported container type	400	
0022	Required parameter '<parameter_name>' is missing	400	
0023	Unable to handle output resource of a non terminal work unit	400	output parameter must not be set if the WorkUnit has a nextRequest
0024	Unsupported token type	400	The token associated with the current Device is not supported by ATP or the required operation is not supported by the token
0025	Document is not a pdf	400	
0026	Number of max signature for transaction has been reached	400	
0027	Invalid signature, please check otp or session input parameter	400	
0028	Unsupported credentialInfo type	400	
0029	FEA OSS user not configured	500	
0030	OSS error <error_info>	500	
0031	Unable to obtain transactionExtraInfo	500	
0032	Error sending credentials	500	Generic error sending credential
0033	File not found	404	
0034	Unable to delete file: <file_info>	500	
0035	Unable to access file	403	
0036	Invalid device: <device_info>	400	
0037	Invalid session id	400	
0038	Unable to get certificate	500	An error occurred while retrieving user certificate
0039	Invalid TSA credentials	500	
0040	Invalid Signature Algorithm	400	
0041	Invalid input hash size	400	
0042	Invalid Signature Profile	400	The selected profile is not valid for JAdES or XAdES signature

0043	Constraint <constraint_type> [<constraint_oid>] validation failed	400	The validation of the requested signature's constraint failed
0044	Constraint <constraint_type> not recognized	400	
0045	No sendable credential: <credential_type>	400	The user does not have a valid credential for the selected type
0046	Input signature type not recognized	400	The signature type of input data is not recognized
0047	Invalid credentials	400	
0048	Invalid OTP	400	
0049	User temporarily suspended	400	Remote signature user is temporarily suspended
0050	User certificate changed	400	Remote signature user's certificate is changed. Returned when the old certificate id is passed in the request
0051	User locked	400	Remote signature user is locked
0052	Invalid user status	400	
0053	Invalid certificate	400	
0054	Expired password	400	
0055	Password mismatch	400	
0056	Invalid password length	400	
0057	Network error	500	A network error occurred while contacting remote services
0058	Token not present for selected device	400	Error loading the sign token associated with the current device
0059	Input document is not signed	400	
0060	Invalid parameter '<parameter_name>' format	400	
0061	Internal document not found	400	
0062	TSA server network error	500	A network error occurred while contacting TSA services
0063	No more timestamp	400	
0064	TSA Error missing TSQ	400	
0065	TSA Error wrong TSQ	400	
0066	TSA Error wrong TSQ hash algorithm	400	
0067	TSA internal error	500	
0068	TSA Error too many request	400	
0069	TSA Error service unavailable	400	
0070	TSA account disabled	400	
0071	TSA account expired	400	
0072	TSA error resource not found	400	
0073	TSA Error bad request	400	
0074	Automatic timestamp error	500	
0075	Timestamp generic error	500	
0076	Remote validation service error: <error_details>	500	
0077	Invalid pdf page	400	
0078	Error signing pdf	400	
0079	Invalid pin	400	
0080	User operational status blocked	400	
0081	Invalid application credential	400	
0082	Signature timeout	500	
0083	Max signature request exceeded	400	
0084	Open session denied	400	
0085	Invalid signature parameters	400	

0086	Invalid user last update	400	
0087	Change password grace period expired	400	
0088	Invalid key format	400	
0089	An error occurred during signing: <signing_error>	500	
0090	Invalid signature	400	
0091	Invalid algorithm	400	
0092	Encryption error	400	
0093	Error reading file	400	
0094	Error writing file	400	
0095	Error decoding Base64	400	
0096	Date format error	400	
0097	Number format error	400	
0098	Timezone format error	400	
0099	Token not found	400	
0100	Token already initialized	400	
0101	Token not initialized	400	
0102	User not logged in	400	
0103	Alias not found	400	
0104	Wrong alias	400	
0105	No space on device	400	
0106	Invalid key length	400	
0107	Error generating key	400	
0108	Public key not corresponding	400	
0109	Generic token error	500	
0110	Pkcs11 functionality not supported	400	
0111	Pkcs11 slot invalid	400	
0112	Expired certificate	400	
0113	Invalid CRL	400	
0114	Invalid OCSP	400	
0115	Envelope data is missing or unreadable	400	
0116	Envelope is not signed	400	
0117	Envelope is already signed	400	
0118	Error reading pdf	400	
0119	Invalid pdf password	400	
0120	Invalid format: <format>	400	
0121	Operation not allowed	400	
0122	Pin blocked	400	
0123	Invalid puk	400	
0124	Expired pin	400	
0125	Expired puk	400	
0126	Puk blocked	400	
0127	Invalid delegated credentials	400	
0128	Pdf page out of bounds	400	

OID

oid	Label	Description
2.5.4.6	C	country code
2.5.4.10	O	organization
2.5.4.11	OU	organizational unit name
2.5.4.12	T	Title
2.5.4.3	CN	common name
2.5.4.9	STREET	street
2.5.4.5	SN, SERIALNUMBER	device serial number name
2.5.4.7	L	locality name
2.5.4.8	ST	state, or province name
2.5.4.4	SURNAME	
2.5.4.42	GIVENNAME	
2.5.4.43	INITIALS	
2.5.4.44	GENERATION	
2.5.4.45	UNIQUE_IDENTIFIER	
2.5.4.13	DESCRIPTION	
2.5.4.15	BUSINESS_CATEGORY	
2.5.4.17	POSTAL_CODE	
2.5.4.46	DN_QUALIFIER	
2.5.4.65	PSEUDONYM	RFC 3039 Pseudonym
2.5.4.72	ROLE	
1.3.6.1.5.5.7.9.1	DATE_OF_BIRTH	RFC 3039 DateOfBirth - YYYYMMDD000000Z
1.3.6.1.5.5.7.9.2	PLACE_OF_BIRTH	RFC 3039 PlaceOfBirth
1.3.6.1.5.5.7.9.3	GENDER	RFC 3039 Gender
1.3.6.1.5.5.7.9.4	COUNTRY_OF_CITIZENSHIP	RFC 3039 CountryOfCitizenship - ISO 3166
1.3.6.1.5.5.7.9.5	COUNTRY_OF_RESIDENCE	RFC 3039 CountryOfResidence - ISO 3166
1.3.36.8.3.14	NAME_AT_BIRTH	ISIS-MTT NameAtBirth
2.5.4.16	POSTAL_ADDRESS	RFC 3039 PostalAddress
2.5.4.54	DMD_NAME	RFC 2256 dmdName
2.5.4.20	TELEPHONE_NUMBER	
2.5.4.41	NAME	
2.5.4.97	ORGANIZATION_IDENTIFIER	
1.2.840.113549.1.9.1	EmailAddress	
1.2.840.113549.1.9.2	UnstructuredName	
1.2.840.113549.1.9.8	UnstructuredAddress	
1.2.840.113549.1.9.1	E	email address in Verisign certificates
0.9.2342.19200300.100.1.25	DC	domain components
0.9.2342.19200300.100.1.1	UID	LDAP User id