MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

REMOTE SIGNING

SERVICE PROVIDER

SOAP API Specification

Version 2.1.211017

Ho Chi Minh – 2021

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

History

Version	Date	Description	Author
1.0	20180824	Begin	VUDP
1.1	20180828	Added below 12 funtionalities - approveCertificateForSignCloud - getCertificateDetailForSignCloud - prepareRenewCertificateForSignCloud - prepareRevokeCertificateForSignCloud - uploadFileForSignCloud - downloadFileForSignCloud - changePasscodeForSignCloud - forgetPasscodeForSignCloud - authorizeSingletonSigningForSignCloud - prepareHashSigningForSignCloud - authorizeHashSigningForSignCloud - assignCertificateForSignCloud Added timeStamp on/off for counter signer Added postbackUrl for asynchronization API	KHANHPX
1.2	20180831	Added: notificationSubject as Email subject (authorization method is OTP Email) Added: ErrorCode 1009 – AGREEMENT EXISTED Added: messaging mode for authorization of signing process Added below 1 funtionality - getSignatureValueForSignCloud	VUDP
1.3	20180904	Added some appendices - CredentialData Details (Java and Python helper classes) - SignCloudMetadata Details - Test Vectors for integration Added XSL:template for prepareFileForSignCloud (transaction optimization) Changed Response Code from 1006 -> 1010 Move SignCloudMetadata from prepareFile to authorizeCounterSigning/authorizeSingletonSiging	VUDP
1.4	20180926	Added relyingPartyBillCode for RP for all functionalities Added billCode for Remote Signing for all functionalities Changed agreementUUID to agreementUUID	VUDP
1.5	20181001	Revised the function of prepareFileForSignCloud for optimize 2 cases - Accepted both of certificate request and file signing request at the same time - Get the signed file once request submission completely in case of using default passcode and preconfigured authorization in the file request	VUDP
1.6	20181010	Added 2 parameters dmsMetaData and p2pEnabled in 2 functionalities - getSignedFileForSignCloud - downloadFileForSignCloud Change 4th function to getSignedFileForSignCloud	VUDP

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

Version	Date	Description	Author
		Added xmlSignature into 3 functionalities - prepareFileForSignCloud - authorizeCounterSigningForSignCloud - authorizeSingletonSigningForSignCloud	
	20181114	Added Response Codes and Messages (1011 → 1019)	VUDP
	20181120	Edited Response Codes and Messages (1011 → 1019)	VUDP
1.9	20190802	Added Response Codes and Messages (1025) Edited Objects in PrepareHashSigningForSignCloud	VUDP
1.10	20190813	Update regenerateAuthorizationCodeForSignCloud	VUDP
	20190920	Added prepareChangeCertificateForSignCloud Added attributes: keepOldKeysEnabled; revokeOldCertificateEnabled	VUDP
	20191209	Add new 3 APIs - declineCertificateForSignCloud - disableCertificateForSignCloud - enableCertificateForSignCloud	VUDP
2.0	20200107	Document is re-formatted/changed logo	VUDP
	20200109	Class AgreementDetials, add new attribute "citizenID" to support "CCCD:" in certificate information Add RESPONSE_CODE: 1026;1027;1028;1029	VUDP
	20200131	Added authMode in SignCloudReq object **authorizeMethod** will be deprecated in the future (session 2.2.2; 2.15.2; 2.19.2)	VUDP
	20200203	Added authModeSupported in SignCloudResp object (session 2.6)	VUDP
	20200302	Added promotion attribute in SignCloudReq object (session 2.1.2) Added signingProfile attribute in SignCloudReq object (session 2.1.2) Added contractExpiration attribute in SignCloudResp (session 2.6) Added ownerUUID, ownerUsername,OwnerPassword in SignCloudReq object (session 2.1.1)	VUDP
2.1	20200526	Added 3 new APIs 2.24 activateTSEDeviceForSignCloud 2.25 authorizeTSEForSignCloud 2.26 signTSEForSignCloud	VUDP
	20200528	At item 2.25, authorizationRequest JSON add	VUDP
	20200704		

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

Content

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

1.INTRODUCTION

This document describes in details the API of Remote Signing service. In this version, API is now including 20 functions:

- prepareCertificateForSignCloud
- approveCertificateForSignCloud
- prepareFileForSignCloud
- prepareHashSigningForSignCloud
- authorizeCounterSigningForSignCloud
- authorizeSingletonSigningForSignCloud
- authorizeHashSigningForSignCloud
- getSignedFileForSignCloud
- getSignatureValueForSignCloud
- getCertificateDetailForSignCloud
- setCertificateDetailForSignCloud
- prepareRenewCertificateForSignCloud
- prepareRevokeCertificateForSignCloud
- prepareChangeCertificateForSignCloud
- uploadFileForSignCloud
- downloadFileForSignCloud
- changePasscodeForSignCloud
- forgetPasscodeForSignCloud
- assignCertificateForSignCloud
- regenerateAuthorizationCodeForSignCloud
- declineCertificateForSignCloud
- disableCertificateForSignCloud
- enableCertificateForSignCloud
- activateTSEDeviceForSignCloud
- authorizeTSEForSignCloud
- signTSEForSignCloud

1.1. Target

Banks, Finance/Insurance companies who wants to apply digital signature for loan approval.

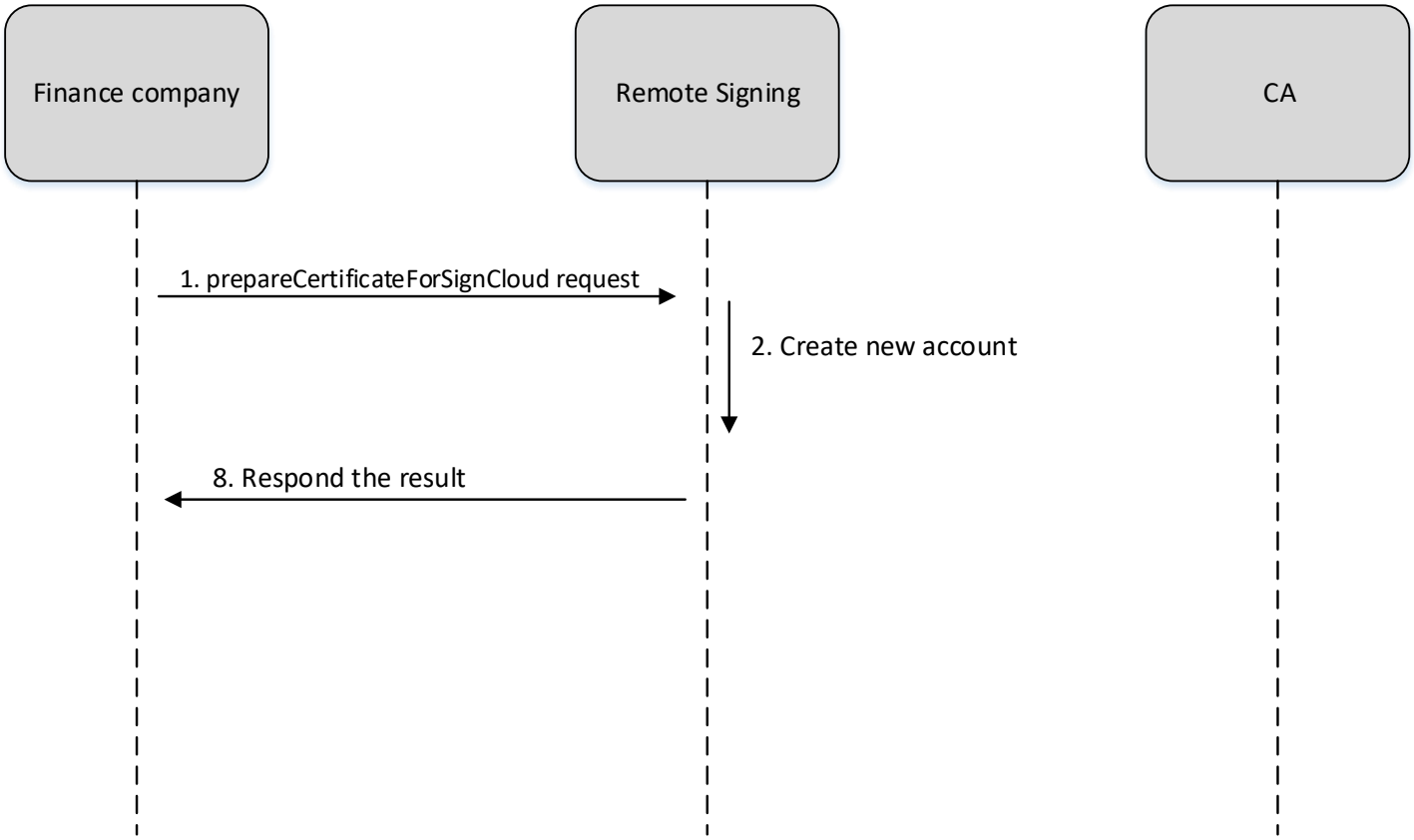
1.2. Abbreviation

Word	Description
CA	Certification Authority
PKI	Public Key Infrastructure
M	Mandatory
O	Optional
M/O	Mandatory/Optional depends on each specific case
AP	Application Provider
RP	Relying Party

2. API SPECIFICATION

2.1. prepareCertificateForSignCloud

This method will create a new account on Remote Signing service and request to CA for preparing digital certificate.



2.1.1. Prototype

```
SignCloudResp prepareCertificateForSignCloud(SignCloudReq signCloudReq);
```

2.1.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudReq Attributes				
No	Name	Type	Require	Description
4	ownerUUID	String	O	System will find OWNER_ID based on ownerUUID and the newly registered certificate will be belonged to this owner. If this value is NULL, new owner will be created.
5	ownerUsername	String	O	ownerUsername and ownerPassword are both required not NULL to take effect. Newly created owner will be assigned this username/password. Error will be returned if the ownerUsername existed.
6	ownerPassword	String	O	
7	mobileNo	String	O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication. In case of the Relying Party keep the customer information confidentially, it should be absence
8	email	String	O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence
9	certificateProfile	String	O	The profile of certificate will be used to enroll. Once this parameter is absence, the default certificate profile is used based on RelyingParty properties
10	promotion	int	O	The extra free day will be added for end-user certificate instead of only the day configured in certificate profile.
11	signingProfile	String	O	The profile decides the number of signing times the end-user can use. This value will be decreased once user performs signing.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				User cannot sign if the signing counter is 0. If signing counter is -1, user's siging times is unlimited.
12	agreementDetails	AgreementDetails	M	Customer information. This one will be used as certificate information.
13	sharedMode	int	O	Which sharing mode will be used in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE Default value is PRIVATE_MODE. It's meant once Relying Party don't use this parameter, the PRIVATE_MODE is used instead
14	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate enrollment process is completely finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties Default value is **FALSE**, Postback mechanism shouldn't be used
15	csrRequired	Boolean	O	If csrRequired set **TRUE**, Remote Signing service will generate keypair for customer and build the CSR for the response.
16	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudReq Attributes				
No	Name	Type	Require	Description
				could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

In case of AgreementDetails, (*) is mandatory and (/*) is at least ONE in mandatory

AgreementDetails Attributes				
No	Name	Type	Require	Description
1	**personName (*)**	String	M/O	Name of customer
2	organization	String	M/O	Company name
3	organizationUnit	String	M/O	Department nam
4	title	String	M/O	Title of employee
5	email	String	M/O	Personal email
6	telephoneNumber	String	M/O	Personal phone number
7	**location (*)**	String	M	
8	**stateOrProvince (*)**	String	M	City or Province
9	country	String	M	Country (Just two letter) E.g: VN
10	**personalID (/*)**	String	M/O	Personal ID
	passportID (/*)	String	M/O	Personal Passport ID
11	taxID (/*)	String	M/O	Tax ID
	budgetID (/*)	String	M/O	Budget ID
12	applicationForm	Byte[]	O	Remote Signing Service application form
13	requestForm	Byte[]	O	Remote Signing Service request form for recovery, revocation …
14	authorizeLetter	Byte[]	O	Authorize Letter by Government/Corporate
15	photoIDCard	Byte[]	O	Scanned photo of ID Card that applied for personal certificate (In case only 1 file available)
16	photoFrontSideIDCard	Byte[]	O	Scanned front side of photo of ID Card that applied for personal certificate
17	photoBackSideIDCard	Byte[]	O	Scanned back side side of photo of ID Card that applied for personal certificate

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

AgreementDetails Attributes				
No	Name	Type	Require	Description
18	photoActivityDeclaration	Byte[]	O	Scanned photo of 'Initial Activity/Alteration Declation' that applied for corporate certificate

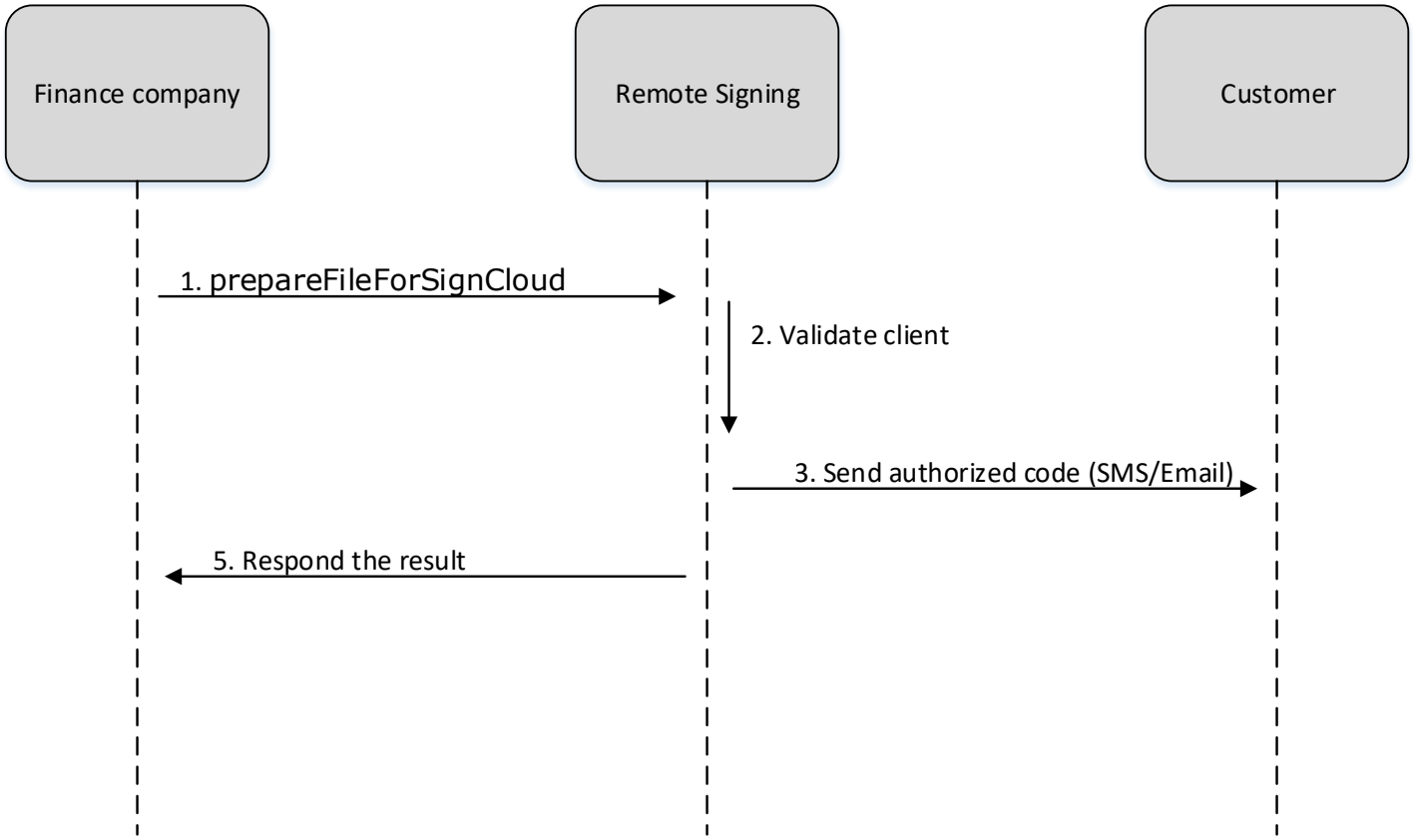
SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally. Remote Signing will response **SUCCESSFULLY**
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	certificateDN	String	M/O	Certificate Distinguished Name (DN) ***E.g: CN=Nguyen Van A, O=B Company,C=VN***
5	certificateSerialNumber	String	M/O	Certificate serial number E.g: 5405ABCDEF
6	certificateThumbprint	String	M/O	Certificate thumprint (certificate hash)
7	validFrom	Date	M/O	The date of certificate begin to be valid
8	validTo	Date	M/O	The date of certificate begin to be expired
9	issuerDN	String	M/O	Issuer Certifcate Distinguished Name (DN) ***E.g: CN=FPT Certification Authority, O=FPT-CA,C=VN***
10	certificate	String	M/O	The data of certificate in PEM format
11	certificateStateID	int	M/O	Certificate state ID • INITIALIZED = 2 • GENERATED = 3 • OPERATED = 4 • REVOKED = 5 • EXPIRED = 6 • DECLINED = 7 • RENEWED = 8 • RENEWED_EXPIRED = 9 • REVISED = 10 • RENEWED_KEEP_SN = 11

SignCloudResp Attributes				
No	Name	Type	Require	Description
				• BLOCKED = 12 REVISED_KEEP_SN = 13
12	signingCounter	int	M/O	How many times the certificate is used for signing. The default value is 1. For document signing this value should be greater than 1

CredentialData Attributes				
No	Name	Type	Require	Description
1	username	String	M	Username of relyingParty provided by Remote Signing
2	password	String	M	Password of relyingParty provided by Remote Signing
3	signature	String	M	Signature of relyingParty provided by Remote Signing
4	pkcs1Signature	String	M	Value is signed from client private key. Key is generated and provided by Remote Signing Value=username + password + signature + timestamp
5	timestamp	String	O	Current timestamp, format in yyyyddmmHHMMss or epoch time If timestamp is absence, the pkcs1Signature is fixed for all transactions

2.2. prepareFileForSignCloud

This method is known as "Sign Request", it means that client request to Remote Signing. Remote Signing will send back an Authorize Code via SMS or Email to authorize the client. This method also requires the binary of document which needs to be signed.



2.2.1. Prototype

```
SignCloudResp prepareFileForSignCloud(SignCloudReq signCloudReq);
```

2.2.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	mobileNo	String	M/O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				In case of the Relying Party keep the customer information confidentially, it should be absence
5	email	String	M/O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence
6	certificateProfile	String	M/O	The profile of certificate will be used to enroll. Once this parameter is absence, the default certificate profile is used based on RelyingParty properties
7	agreementDetails	AgreementDetails	M/O	Customer information. This one will be used as certificate information.
8	sharedMode	int	M/O	Which sharing mode will be used in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE Default value is PRIVATE_MODE. It's meant once Relying Party don't use this parameter, the PRIVATE_MODE is used instead
9	postbackEnabled	Boolean	M/O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate enrollment process is completely

SignCloudReq Attributes				
No	Name	Type	Require	Description
				finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties Default value is **FALSE**, Postback mechanism shouldn't be used
10	signingFileData	Byte[]	M/O	File binary that will be signed
11	mimeType	String	M/O	In case of PDF signer, mimeType is '**application/pdf**'
12	signingFileUUID	String	M/O	Prepared file UUID that identified by our Remote Signing service Once signingFileData is present, we will ignore the signingFileUUID (it means P2P method should be used instead)
13	xslTemplateUUID	String	M/O	In case of optimization, Relying Party just used the XSLT and transform an XML Document into PDF file for signing Relying Party will send the XSLT UUID that already registered in Remote Signing. This UUID need to be negotiated between both once running the Remote Signing service
14	xslTemplate	String	M/O	In case of optimization, Relying Party just used the XSLT and transform an XML Document into PDF file for signing
15	xmlDocument	String	M/O	XML Document is customer data that need to be signed
16	multipleSigningFileData	List<MultipleSigningFileData>	M/O	In case of multiple files need to be signed by one time

SignCloudReq Attributes				
No	Name	Type	Require	Description
				authentication. This object contains a list of document will be signed.
17	notificationTemplate	String	M	Message contains Authorize Code will be sent to customer's Phone/Email. Authorize Code is generated on Remote Signing and embedded into template. E.g: Your authorize code: **{AuthorizeCode}**. Authorize Code is valid within 5 minutes.
18	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization. If OTP Email is used, this parameter is mandatory It should be used in case of the Remote Signing will use owned SMTP to send OTP Email to customer.
19	timestampEnabled	Boolean	O	Timestamp on/off, if this value set **TRUE**, the signing process should be added the TSA configuration that already configured in RelyingParty properties Default value is **FALSE**
20	language	String	O	Possible value: VN, EN
21	authorizeMethod	int	RC	Which method will be used to authorize to client. 1: OTP SMS (default) 2: OTP Email 3: OTP Mobile 4: Passcode

SignCloudReq Attributes				
No	Name	Type	Require	Description
				5: UAF *This attribute will be deprecated in the future and replaced by authMode*
22	authMode	String	RC	Specifies one of the authorization modes. ▪ "EXPLICIT/PIN": the authorization process is managed by the signature application, authentication method is passcode. ▪ "EXPLICIT/OTP-SMS": the authorization process is managed by the signature application, authentication method is otp sms. ▪ "EXPLICIT/OTP-EMAIL": the authorization process is managed by the signature application, authentication method is otp email. ▪ "IMPLICIT/CYBER-ID": the authorization process is managed by the remote service autonomously. Authentication factors are managed by the RSSP by interacting directly with the user, and not by the signature application. ▪ "IMPLICIT/BIP-CATTP": the authorization process is managed by Cyber-ID – a mobile application – which

SignCloudReq Attributes				
No	Name	Type	Require	Description
				could interact to PKI USIM for signing. ▪ "EXPLICIT/OTP-MOBILE": the authorization process is managed by OTP Mobile Application. ▪ "OAUTH2": not implement yet.
23	signCloudMetaData	signCloudMetaData	O	SignCloudMetaData contains additional signer properties which are used for PDF signature display. - Coordinate - Background image - Signer location - Signer reason - Text color In case of XML sigining, it will indicate which type should be applied XMLDSig, XML-XaDES, signature zone Once this parameter is not used, Remote Signing should use the default value that already configured in the system In case of counter signing process, SignCloudMetaData included 2 MetaDatas for customer and Relying Party signer properties
24	authorizeCode	String	M/O	Authorize Code provided by customer In this case of passcode (4) is used, authorize code is passcode

SignCloudReq Attributes				
No	Name	Type	Require	Description
25	messagingMode	int	M/O	Which messaging mode will be used **1**: ASYNCHRONOUS_CLIENTSERVER (default) **2**: ASYNCHRONOUS_SERVERSERVER **3**: SYNCHRONOUS If default value 0 is used, the API downloadSignedFileForSignCloud will be used for getting the signed file
26	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	notificationMessage	String	M/O	In case of Authorize Code isn't sent by Remote Signing, notification message which has already included Authorize Code will be sent back to Finance/Insurance company to send to their end-user.
5	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization.

SignCloudResp Attributes				
No	Name	Type	Require	Description
				It should be used in case of RelyingParty used their owned SMTP server for notify customer
6	authorizeCredential	String	M/O	In case of Authorize Code isn't sent by Remote Signing, Email/Mobile number of client/customer also is responded back to Finance/Insurance company and they use that information to send Authorize Code to their end-user.
7	signedFileData	Byte[]	M/O	File binary that be signed
8	mimeType	String	M/O	In case of PDF file that be downloaded, mimeType is '**application/pdf**'
9	signedFileUUID	String	M/O	Signed file UUID that identified by our Remote Signing service
10	multipleSignedFileData	List<MultipleSignedFileData>	M/O	In case of signing multiple files, the multiple signed files will be responded in this object.
11	xmlSignature	String	M/O	XML Signature (XMLDSig or XML-XadDES)
12	certificateDN	String	M/O	Certificate Distinguished Name (DN) ***E.g: CN=Nguyen Van A, O=B Company,C=VN***
13	certificateSerialNumber	String	M/O	Certificate serial number E.g: 5405ABCDEF
14	certificateThumbprint	String	M/O	Certificate thumprint (certificate hash)
15	validFrom	Date	M/O	The date of certificate begin to be valid

SignCloudResp Attributes				
No	Name	Type	Require	Description
16	validTo	Date	M/O	The date of certificate begin to be expired
17	issuerDN	String	M/O	Issuer Certifcate Distinguished Name (DN) ***E.g: CN=Mobile-ID Trusted Network, O=MOBILE-ID,C=VN***
18	sharedMode	int	M/O	Which sharing mode applied to this agreement in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE

MultipleSigningFileData Attributes				
No	Name	Type	Require	Description
1	signingFileData	Byte[]	M/O	The binary of document
2	signingFileName	String	M/O	The document name
3	mimeType	String	M/O	The document mime type. The possible values: • application/pdf • application/xml • application/xhtml+xml • application/msword • application/ vnd.openxmlformats-officedocument.wordprocessingml.document • application/vnd.ms-powerpoint • application/vnd.openxmlformats-officedocument.presentationml.presentation • application/vnd.ms-excel • application/vnd.openxmlformats-officedocument.spreadsheetml.sheet • application/vnd.visio
4	xslTemplate	String	M/O	The xslt. It is used to combine with XML document and create the final pdf file.

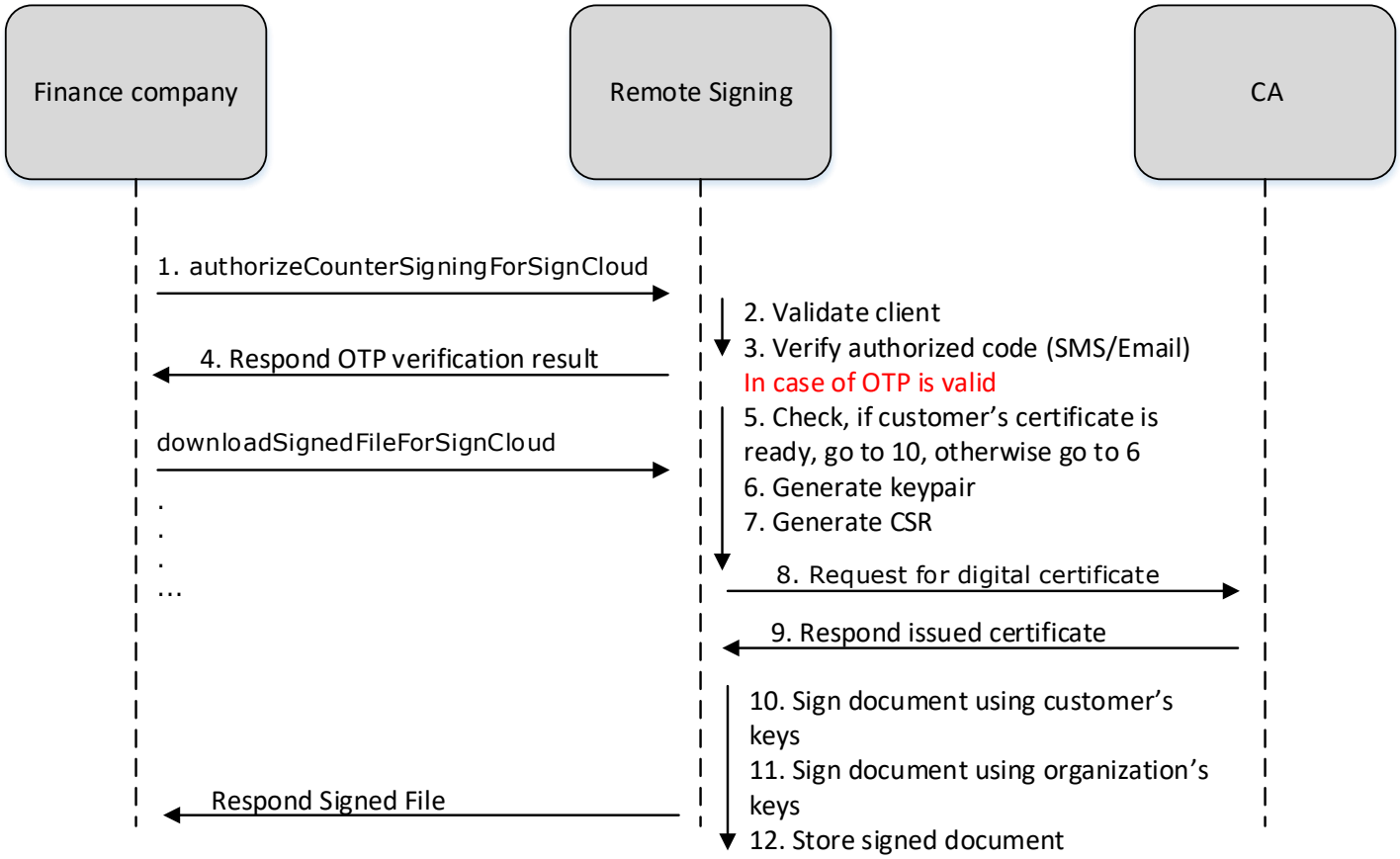
MultipleSigningFileData Attributes				
No	Name	Type	Require	Description
5	xmlDocument	String	M/O	The document in xml format.

MultipleSignedFileData Attributes				
No	Name	Type	Require	Description
1	signedFileData	Byte[]	M/O	The signed document binary
2	mimeType	String	M/O	The signed document mime type
3	signedFileName	String	M/O	The signed document name
4	signatureValue	String	M/O	The PKCS#1 Signature as Base64 String will be sent to RelyingParty
5	signedFileUUID	String	M/O	The signed document file UUID in DMS
6	dmsMetaData	String	M/O	The DMS information used for signed document download

2.3. authorizeCounterSigningForSignCloud

This method is known as "Sign Response", that means client responds the Authorize Code to Remote Signing. Remote Signing will verify the received Authorize Code and response signed document (contract) whether it's correct. In case of the agreement don't own the certificate, the Remote Signing will enroll the corresponding ceritificate for business optimization and later will sign the document

Once the customer's private key is used for document signing, the organization private key that corresponded to the RelyingParty owned will be used for document co-signing process



2.3.1. Prototype

```
SignCloudResp authorizeCounterSigningForSignCloud(SignCloudReq signCloudReq);
```

2.3.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	billCode obtained from response of **prepareFileForSignCloud**
5	authorizeCode	String	M	Authorize Code provided by customer
6	messagingMode	int	O	Which messaging mode will be used **1**: ASYNCHRONOUS_CLIENTSERVER (default) **2**: ASYNCHRONOUS_SERVERSERVER

SignCloudReq Attributes				
No	Name	Type	Require	Description
				3: SYNCHRONOUS If default value 0 is used, the API downloadSignedFileForSignCloud will be used for getting the signed file
7	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail. Normally, this responses **SUCCESSFULLY**
3	billCode	String	M	Receipt for each transaction
4	remainingCounter	int	M	Counter decreases once invalid Authorization Code provided. AgreementUUID will be blocked if counter goes to zero and it's automatically unblocked within 5 minutes. The default value for automatic unblock is 5 minutes, it could be reconfigured in the system
5	signedFileData	Byte[]	M/O	File binary that be signed
6	mimeType	String	M/O	In case of PDF file that be downloaded, mimeType is '**application/pdf**'
7	signedFileUUID	String	M/O	Signed file UUID that identified by our Remote Signing service
8	multipleSignedFileData	List<MultipleSignedFileData>	M/O	In case of signing multiple files, the multiple signed files will be responded in this object.

SignCloudResp Attributes				
No	Name	Type	Require	Description
9	xmlSignature	String	M/O	XML Signature (XMLDSig or XML-XadDES)

2.4. getSignedFileForSignCloud

This method will download signed file that already signed by Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.4.1. Prototype

```
SignCloudResp getSignedFileForSignCloud(SignCloudReq signCloudReq);
```

2.4.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	O	Receipt for each transaction. If omitted, get the latest signed files of the agreement ID. The number of file is mentioned by "filesLimit"
5	filesLimit	int	O	Number of file will be responded. The default is 5 and maxium file allowed to download is 10
6	p2pEnabled	boolean	O	Must provide TRUE Once this parameter is TRUE, RP should download directly from Remote Signing service. Vice versa, it should be downloaded by DMS
7	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.

SignCloudResp Attributes				
No	Name	Type	Require	Description
2	responseMessage	String	M	Message describes the result in detail. 2 cases we got - SIGNING PROCESSING - SIGNING FINALIZED
3	billCode	String	M	Receipt for each transaction
4	signedFileData	Byte[]	M/O	File binary that be signed (in case of only one signed file available)
5	mimeType	String	M/O	In case of PDF file that be downloaded, mimeType is **'application/pdf'**
6	signedFileUUID	String	M/O	Signed file UUID that identified by our Remote Signing service
7	multipleSignedFileData	List<MultipleSignedFileData>	M/O	In case of signing multiple files, the multiple signed files will be responded in this object.
8	dmsMetaData	String	M/O	DMS Metadata included DMS credential (URI, username, password)

2.5. approveCertificateForSignCloud

This method will approve the cerficate prepare for enroll (create/renew) a certificate on Remote Signing service. Normally RelyingParty will allow to approve this request once checked all the detail information that customer owned

[WORKFLOW WILL BE UPDATED SOON]

2.5.1. Prototype

```
SignCloudResp approveCertificateForSignCloud(SignCloudReq signCloudReq);
```

2.5.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudReq Attributes				
No	Name	Type	Require	Description
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate enrollment process is completely finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties Default value is FALSE
5	certificate	String	M	The data of certificate in PEM format
6	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	certificateDN	String	M/O	Certificate Distinguished Name (DN) ***E.g: CN=Nguyen Van A, O=B Company,C=VN***
5	certificateSerialNumber	String	M/O	Certificate serial number E.g: 5405ABCDEF
6	certificateThumbprint	String	M/O	Certificate thumprint (certificate hash)
7	validFrom	Date	M/O	The date of certificate begin to be valid
8	validTo	Date	M/O	The date of certificate begin to be expired
9	issuerDN	String	M/O	Issuer Certifcate Distinguished Name (DN) ***E.g: CN=Mobile-ID Trusted Network, O=MOBILE-ID,C=VN***

SignCloudResp Attributes				
No	Name	Type	Require	Description
10	sharedMode	int	M/O	Which sharing mode applied to this agreement in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE

2.6. getCertificateDetailForSignCloud

This method will get certificate detail for corresponding agreement in Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.6.1. Prototype

```
SignCloudResp getCertificateDetailForSignCloud(SignCloudReq signCloudReq);
```

2.6.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	credentialData	CredentialData	M	Credential information used for client-server handshake
5	agreementDetails	AgreementDetails	O	Customer information. It is required for signer double checking before the detail of certificate information responded. One of below information should be provided: + personalName + personalID or citizenID or passportID + organization + taxID or budgetID

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	certificateDN	String	M/O	Certificate Distinguished Name (DN) ***E.g: CN=Nguyen Van A, O=B Company,C=VN***
5	certificateSerialNumber	String	M/O	Certificate serial number E.g: 5405ABCDEF
6	certificateThumbprint	String	M/O	Certificate thumprint (certificate hash)
7	validFrom	Date	M/O	The date of certificate begin to be valid
8	validTo	Date	M/O	The date of certificate begin to be expired
9	contractExpiration	Date	M/O	The date of user's contract begin to be expired
10	issuerDN	String	M/O	Issuer Certifcate Distinguished Name (DN) ***E.g: CN=Mobile-ID Trusted Network, O=MOBILE-ID,C=VN***
11	certificate	String	M/O	The data of certificate in PEM format
12	sharedMode	int	M/O	Which sharing mode will be used in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE Default value is PRIVATE_MODE
13	authModeSupported	String[]	M/O	Specifies one of the authorization modes. ▪ "EXPLICIT/PIN": the authorization process is managed by the signature application, authentication method is passcode. ▪ "EXPLICIT/OTP-SMS": the authorization process is managed by the signature application, authentication method is otp sms. ▪ "EXPLICIT/OTP-EMAIL": the authorization process is managed by the signature application, authentication method is otp email.

SignCloudResp Attributes				
No	Name	Type	Require	Description
				▪ "IMPLICIT/CYBER-ID": the authorization process is managed by the remote service autonomously. Authentication factors are managed by the RSSP by interacting directly with the user, and not by the signature application. ▪ "IMPLICIT/BIP-CATTP": the authorization process is managed by Cyber-ID – a mobile application – which could interact to PKI USIM for signing. ▪ "EXPLICIT/OTP-MOBILE": the authorization process is managed by OTP Mobile Application. ▪ "OAUTH2": not implement yet.

2.7. setCertificateDetailForSignCloud

This method is allow to update the authorization email/mobileNo which is corresponding to the agreement in Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.7.1. Prototype

```
SignCloudResp setCertificateDetailForSignCloud(SignCloudReq signCloudReq);
```

2.7.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number

SignCloudReq Attributes				
No	Name	Type	Require	Description
4	mobileNo	String	O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication. In case of the Relying Party keep the customer information confidentially, it should be absence
5	email	String	O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence
6	authorizeCode	String	M	Authorize Code provided by customer. The customer provides the correct authorizeCode, the email/mobileNo could be updated.
7	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.8. prepareRenewCertificateForSignCloud

This method will extend the certificate expiration for an agreementUUID. You CANNOT change the certificate information with this method, the old information will be used for new certificate. But you can change the certificate profile.

[WORKFLOW WILL BE UPDATED SOON]

2.8.1. Prototype

```
SignCloudResp prepareRenewCertificateForSignCloud(SignCloudReq signCloudReq);
```

2.8.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudReq Attributes				
No	Name	Type	Require	Description
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	mobileNo	String	O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication. In case of the Relying Party keep the customer information confidentially, it should be absence If information is absence, the existing information will be used instead
5	email	String	O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence If this information is absence, the existing information will be used instead
6	certificateProfile	String	O	The profile of certificate will be used to renew. If this information is absence, the existing information will be used instead
7	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate enrollment process is completely finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties If this information is absence, the existing information will be used instead
8	keepOldKeysEnabled	Boolean	O	If you don't want to generate the new keypair, set this attribute to True. The default is False.

SignCloudReq Attributes				
No	Name	Type	Require	Description
9	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally. Remote Signing will response **SUCCESSFULLY**
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.9. prepareRevokeCertificateForSignCloud

This method will destroy an existing account on Remote Signing service and request to CA for preparing digital certificate revocation.

[WORKFLOW WILL BE UPDATED SOON]

2.9.1. Prototype

```
SignCloudResp prepareRevokeCertificateForSignCloud(SignCloudReq signCloudReq);
```

2.9.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate revocation process is completely finalized, the Remote Signing will send the details based on

SignCloudReq Attributes				
No	Name	Type	Require	Description
				the postbackUrl that already configured in RelyingParty properties Default value is **FALSE**
5	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally. Remote Signing will response **SUCCESSFULLY**
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.10. uploadFileForSignCloud

This method will upload file based on Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.10.1. Prototype

```
SignCloudResp uploadFileForSignCloud(SignCloudReq signCloudReq);
```

2.10.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	uploadingFileData	Byte[]	M	File binary that will be uploaded
5	mimeType	String	M	In case of PDF, mimeType is '**application/pdf**'
6	credentialData	CredentialData	M	Credential information used for client-server handshake

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	mimeType	String	M/O	In case of PDF file that be uploaded, mimeType is '**application/pdf**'
5	uploadedFileUUID	String	M/O	Uploaded file UUID that identified by our Remote Signing service

2.11. downloadFileForSignCloud

This method will download file that already signed by Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.11.1. Prototype

```
SignCloudResp downloadFileForSignCloud(SignCloudReq signCloudReq);
```

2.11.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	O	BillCode which is responded by authorizeCounterSigningForSignCloud Note: billCode and downloadingFileUUID cannot be both NULL or EMPTY
5	downloadingFileUUID	String	O	Downloading file UUID that identified by our Remote Signing service Note: billCode and downloadingFileUUID cannot be both NULL or EMPTY
6	mimeType	String	O	In case of PDF, mimeType is '**application/pdf**'
7	p2pEnabled	boolean	O	Default value is TRUE

SignCloudReq Attributes				
No	Name	Type	Require	Description
				Once this parameter is TRUE, RP should download directly from eSignloud service. Vice versa, it should be downloaded by DMS
8	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	downloadedFileData	Byte[]	M/O	File binary that be downloaded
5	mimeType	String	M/O	In case of PDF file that be downloaded, mimeType is '**application/pdf**'
6	downloadedFileUUID	String	M/O	Downloaded file UUID that identified by our Remote Signing service
7	dmsMetaData	String	M/O	DMS MetaData

2.12. changePasscodeForSignCloud

This method will change the current Passcode for authorize as Passcode method on the Remote Signing service

[WORKFLOW WILL BE UPDATED SOON]

2.12.1. Prototype

```
SignCloudResp changePasscodeForSignCloud(SignCloudReq signCloudReq);
```

2.12.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID

SignCloudReq Attributes				
No	Name	Type	Require	Description
4	currentPasscode	String	M	This is the current Passcode that current using for Remote Signing service
5	newPasscode	String	M	This is the new Passcode that will be used later. It should be noted the RelyingParty check new Passcode confirmation for ensure that customer owned the new Passcode
6	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	remainingCounter	int	M	Once changed Passcode successfully, this parameter will be the Max-Counter that already configured in the General Policy of Remote Signing service Once changed Passcode failed, this parameter will be decreased ONE value, if this parameter reached ZERO, our agreement in Remote Signing will be blocked in the duration that already configured. Once this duration ended, our agreement will be unblocked automatically and we continued to use with current Passcode

2.13. forgetPasscodeForSignCloud

This method will reset the current Passcode for authorize as Passcode method on the Remote Signing service , and the new Passcode will be created in the Remote Signing service and this parameter will be sent to customer based on the registration email

[WORKFLOW WILL BE UPDATED SOON]

2.13.1. Prototype

```
SignCloudResp forgetPasscodeForSignCloud(SignCloudReq signCloudReq);
```

2.13.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
3	notificationTemplate	String	M	Message contains Authorize Code will be sent to customer's Email. New Passcode is generated on Remote Signing and embedded into template. E.g: Your new Passcode: **{NewPasscode}**. New Passcode is valid within 30 days.
4	notificationSubject	String	M	This parameter is used as Email subject for forgot Passcode notification. It should be used in case of the Remote Signing will use owned SMTP to send OTP Email to customer.
5	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail. Normally, our message is **REQUEST ACCEPTED**
3	billCode	String	M	Receipt for each transaction
4	notificationMessage	String	M/O	Message contains randomly new Passcode will be sent to customer's Email. New Passcode is generated on Remote Signing and embedded into template. E.g: Your new Passcode: **{Passcode}**. New Passcode is valid within 30 days.

SignCloudResp Attributes				
No	Name	Type	Require	Description
5	notificationSubject	String	M/O	This one is used as Email subject for forget Passcode notification. It should be used in case of RelyingParty used their owned SMTP server for notify customer
6	authorizeCredential	String	M/O	In case of Authorize Code isn't sent by Remote Signing, Email of client/customer also is responded back to Finance/Insurance company and they use that information to send Authorize Code to their end-user.

2.14. authorizeSingletonSigningForSignCloud

This method is known as "Sign Response", that means client responds the Authorize Code to Remote Signing. Remote Signing will verify the received Authorize Code and response signed document (contract) whether it's correct. In case of the agreement don't own the certificate, the Remote Signing will enroll the corresponding ceritificate for business optimization and later will sign the document. This method is different from **authorizeCounterSigningForSignCloud** as only one signature will be applied, the counterpart's one won't be.

[WORKFLOW WILL BE UPDATED SOON]

2.14.1. Prototype

```
SignCloudResp authorizeSingletonSigningForSignCloud(SignCloudReq signCloudReq);
```

2.14.2. Parameters

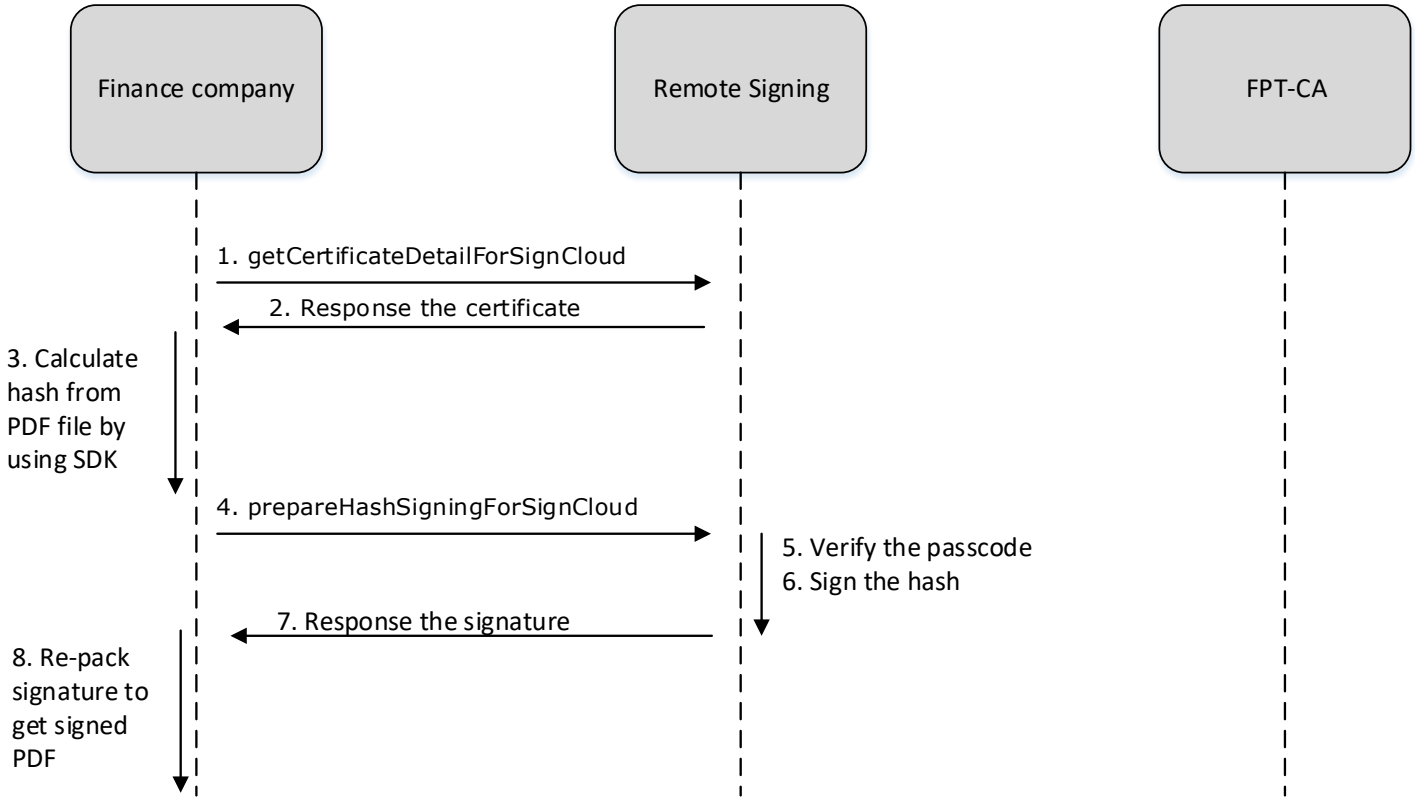
SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	billCode obtained from response of **prepareFileForSignCloud**
5	authorizeCode	String	M	Authorize Code provided by customer
6	messagingMode	int	O	Which messaging mode will be used

SignCloudReq Attributes				
No	Name	Type	Require	Description
				1: ASYNCHRONOUS_CLIENTSERVER (default) **2**: ASYNCHRONOUS_SERVERSERVER **3**: SYNCHRONOUS If default value 0 is used, the API downloadSignedFileForSignCloud will be used for getting the signed file
7	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	remainingCounter	int	M	Counter decreases once invalid Authorization Code provided. AgreementUUID will be blocked if counter goes to zero and it's automatically unblocked within 5 minutes. The default value for automatic unblock is 5 minutes, it could be reconfigured in the system
5	signedFileData	Byte[]	M/O	Signed document
6	mimeType	String	M/O	In case of PDF signer, mimeType is '**application/pdf**'
7	signedFileUUID	String	M/O	Signed file UUID that identified by our Remote Signing service
8	xmlSignature	String	M/O	XML Signature (XMLDSig or XML-XadDES)

2.15. prepareHashSigningForSignCloud

This method is known as "Sign Request", it means that client request to Remote Signing. Remote Signing will send back an Authorize Code via SMS or Email to authorize the client. This method also requires the hash value which needs to be signed.



2.15.1. Prototype

```
SignCloudResp prepareHashSigningForSignCloud(SignCloudReq signCloudReq);
```

2.15.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	mimeType	String	M	The available mime type for hash signing: application/sha1-binary application/sha256-binary application/sha384-binary application/sha512-binary
5	hash	String	M	Hash to be signed
6	hashAlgorithm	String	O	Hash algorithm. The default value is SHA256 The available mime type for hash signing:

SignCloudReq Attributes				
No	Name	Type	Require	Description
				SHA-1; SHA-256; SHA-384; SHA-512
7	encryption	String	O	Signature encryption type. The default value is RSA
8	certificateRequired	Boolean	O	If certificateRequired is TRUE. The signing certificate is also responded
9	notificationTemplate	String	M	Message contains Authorize Code will be sent to customer's Phone/Email. Authorize Code is generated on Remote Signing and embedded into template. E.g: Your authorize code: **{AuthorizeCode}**. Authorize Code is valid within 5 minutes.
10	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization. It should be used in case of the Remote Signing will use owned SMTP to send OTP Email to customer.
11	language	String	O	Possible value: VN, EN
12	authorizeMethod	int	O	Which method will be used to authorize to client. 1: OTP SMS (default) 2: OTP Email 3: OTP Mobile 4: Passcode 5: UAF *This attribute will be deprecated in the future and replaced by authMode*
13	authMode	String	RC	Specifies one of the authorization modes. ▪ "EXPLICIT/PIN": the authorization process is managed by the signature application, authentication method is passcode. ▪ "EXPLICIT/OTP-SMS": the authorization process is managed by the signature application, authentication method is otp sms.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				▪ "EXPLICIT/OTP-EMAIL": the authorization process is managed by the signature application, authentication method is otp email. ▪ "IMPLICIT/CYBER-ID": the authorization process is managed by the remote service autonomously. Authentication factors are managed by the RSSP by interacting directly with the user, and not by the signature application. ▪ "IMPLICIT/BIP-CATTP": the authorization process is managed by Cyber-ID – a mobile application – which could interact to PKI USIM for signing. ▪ "EXPLICIT/OTP-MOBILE": the authorization process is managed by OTP Mobile Application. ▪ "OAUTH2": not implement yet.
14	certificateRequired	Boolean	O	If you want signer certificate, set this attribute to True. The default is False
15	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M/O	Receipt for each transaction
4	notificationMessage	String	M/O	In case of Authorize Code isn't sent by Remote Signing, notification message which has already included Authorize Code will be sent back to Finance/Insurance company to send to their end-user.

SignCloudResp Attributes				
No	Name	Type	Require	Description
5	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization. It should be used in case of RelyingParty used their owned SMTP server for notify customer
6	authorizeCredential	String	M/O	In case of Authorize Code isn't sent by Remote Signing, Email/Mobile number of client/customer also is responded back to Finance/Insurance company and they use that information to send Authorize Code to their end-user.

2.16. authorizeHashSigningForSignCloud

This method is known as "Sign Response", that means client responds the Authorize Code to Remote Signing. Remote Signing will verify the received Authorize Code and responses PKCS#1 signature whether it's correct. In case of the agreement don't own the certificate, the Remote Signing will enroll the corresponding ceritificate for business optimization and later will sign the hash.

[WORKFLOW WILL BE UPDATED SOON]

2.16.1. Prototype

```
SignCloudResp authorizeHashSigningForSignCloud(SignCloudReq signCloudReq);
```

2.16.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	billCode obtained from response of **prepareHashSigningForSignCloud**
5	authorizeCode	String	M	Authorize Code provided by customer
6	messagingMode	int	O	Which messaging mode will be used **1**: ASYNCHRONOUS_CLIENTSERVER (default)

SignCloudReq Attributes				
No	Name	Type	Require	Description
				2: ASYNCHRONOUS_SERVERSERVER **3**: SYNCHRONOUS If default value 0 is used, the API getSignatureValueForSignCloud will be used for getting the signature
7	certificateRequired	Boolean	O	If you want signer certificate, set this attribute to True. The default is False
8	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	remainingCounter	int	M	Counter decreases once invalid Authorization Code provided. AgreementUUID will be blocked if counter goes to zero and it's automatically unblocked within 5 minutes. The default value for automatic unblock is 5 minutes, it could be reconfigured in the system
5	signatureValue	String	M/O	The PKCS#1 Signature as Base64 String will be sent to RelyingParty

2.17. getSignatureValueForSignCloud

This method will get signature value of corresponding the API authorizeHashSigningForSignCloud in Remote Signing service. In this case, messaging mode that current used is ASYCHRONOUS_CLIENTSERVER

[WORKFLOW WILL BE UPDATED SOON]

2.17.1. Prototype

```
SignCloudResp getSignatureValueForSignCloud(SignCloudReq signCloudReq);
```

2.17.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	Receipt for each transaction
5	certificateRequired	Boolean	O	If you want signer certificate, set this attribute to True. The default is False
6	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail. 2 cases we got - SIGNING PROCESSING - SIGNING FINALIZED
3	billCode	String	M	Receipt for each transaction
4	signatureValue	String	M/O	The PKCS#1 Signature as Base64 String will be sent to RelyingParty

2.18. assignCertificateForSignCloud

This method will create a new account on Remote Signing service and assign the existing certificate.

[WORKFLOW WILL BE UPDATED SOON]

2.18.1. Prototype

```
SignCloudResp assignCertificateForSignCloud(SignCloudReq signCloudReq);
```

2.18.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode

SignCloudReq Attributes				
No	Name	Type	Require	Description
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	certificateThumbprint	String	M	Certificate thumprint (certificate hash)
5	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.19. regenerateAuthorizationCodeForSignCloud

This function is used to regenerate the authorization code.

2.19.1. Prototype

```
SignCloudResp regenerateAuthorizationCodeForSignCloud(SignCloudReq signCloudReq);
```

2.19.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	notificationTemplate	String	M	Message contains Authorize Code will be sent to customer's Phone/Email.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				Authorize Code is generated on Remote Signing and embedded into template. E.g: Your authorize code: **{AuthorizeCode}**. Authorize Code is valid within 5 minutes.
5	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization. If OTP Email is used, this parameter is mandatory It should be used in case of the Remote Signing will use owned SMTP to send OTP Email to customer.
6	authorizeMethod	int	O	Which method will be used to authorize to client. 1: OTP SMS (default) 2: OTP Email 3: OTP Mobile 4: Passcode 5: UAF *This attribute will be deprecated in the future and replaced by authMode*
7	authMode	String	RC	Specifies one of the authorization modes. ▪ "EXPLICIT/OTP-SMS": the authorization process is managed by the signature application, authentication method is otp sms. ▪ "EXPLICIT/OTP-EMAIL": the authorization process is managed by the signature application, authentication method is otp email.

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally.

SignCloudResp Attributes				
No	Name	Type	Require	Description
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	notificationMessage	String	M/O	In case of Authorize Code isn't sent by Remote Signing, notification message which has already included Authorize Code will be sent back to Finance/Insurance company to send to their end-user.
5	notificationSubject	String	M/O	This parameter is used as Email subject for signing authorization. It should be used in case of RelyingParty used their owned SMTP server for notify customer
6	authorizeCredential	String	M/O	In case of Authorize Code isn't sent by Remote Signing, Email/Mobile number of client/customer also is responded back to Finance/Insurance company and they use that information to send Authorize Code to their end-user.

2.20. prepareChangeCertificateForSignCloud

This method helps to change the certificate information for a specific agreementUUID. You can re-provide the information in AgreementDetails object. The new certificate will be generated for that agreementUUID with the expiration date is the same as the old certificate. To be noticed that, you CANNOT change: personalID, passportID, taxID and budgetID, because these information used to be representative of an agreementUUID.

2.20.1. Prototype

```
SignCloudResp prepareChangeCertificateForSignCloud(SignCloudReq signCloudReq)
```

2.20.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode

SignCloudReq Attributes				
No	Name	Type	Require	Description
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	mobileNo	String	O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication. In case of the Relying Party keep the customer information confidentially, it should be absence
5	email	String	O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence
6	certificateProfile	String	O	The profile of certificate will be used to enroll. Once this parameter is absence, the default certificate profile is used based on RelyingParty properties
7	agreementDetails	AgreementDetails	M	Customer information. This one will be used as certificate information.
8	sharedMode	int	O	Which sharing mode will be used in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE Default value is PRIVATE_MODE. It's meant once Relying Party don't use this parameter, the PRIVATE_MODE is used instead
9	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response

SignCloudReq Attributes				
No	Name	Type	Require	Description
				REQUEST ACCEPTED immediately. Once the certificate enrollment process is completely finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties Default value is **FALSE**, Postback mechanism shouldn't be used
10	csrRequired	Boolean	O	If csrRequired set **TRUE**, Remote Signing service will generate keypair for customer and build the CSR for the response.
11	keepOldKeysEnabled	Boolean	O	If you don't want to generate the new keypair, set this attribute to True. The default is False.
12	revokeOldCertificateEnabled	Boolean	O	If you want the Remote Signing to revoke the old certificate, set it to True
13	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.

SignCloudResp Attributes				
No	Name	Type	Require	Description
				Normally. Remote Signing will response **REQUEST ACCEPTED**
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	csr	String	O	CSR will be responed if the csrRequired set **TRUE** in the request.

2.21. declineCertificateForSignCloud

Client can only call this method when the the agreement state is INITIALIZED. In practice, the client called prepareCertificateForSignCloud and waiting for approval from server, if the client want to decline this request, this function should be called.

2.21.1. Prototype

```
SignCloudResp declineCertificateForSignCloud(SignCloudReq signCloudReq)
```

2.21.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of

SignCloudReq Attributes				
No	Name	Type	Require	Description
				1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.22. disableCertificateForSignCloud

This function temporarily disable all sign/verify actions related to a agreementUUID.

2.22.1. Prototype

```
SignCloudResp disableCertificateForSignCloud(SignCloudReq signCloudReq)
```

2.22.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the

SignCloudReq Attributes				
No	Name	Type	Require	Description
				credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.23. enableCertificateForSignCloud

This function enable the sign/verify actions related to a agreementUUID after being disabled.

2.23.1. Prototype

```
SignCloudResp enableCertificateForSignCloud(SignCloudReq signCloudReq)
```

2.23.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudReq Attributes				
No	Name	Type	Require	Description
				Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

2.24. activateTSEDeviceForSignCloud

In this phase, RSSP will generate KAK and then respond to TSE device to store. The next phase is TSE device will generate KAK for itself and transfer KAK to RSSP.

2.24.1. Prototype

```
SignCloudResp activateTSEDeviceForSignCloud(SignCloudReq signCloudReq)
```

2.24.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.

SignCloudReq Attributes				
No	Name	Type	Require	Description
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	keyAuthorizationKey	String	M	The key authorization key (KAK) is generated by RSSP and transfer to TSE device. This value is in Base64 encode (could be decrypted if needed)

2.25. authorizeTSEForSignCloud

RP will call this API as the sign request. The authorization data will be sent to TSE device for authentication.

2.25.1. Prototype

```
SignCloudResp authorizeTSEForSignCloud(SignCloudReq signCloudReq)
```

2.25.2. Parameters

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	hash	String	O	Hash to be signed. For SCAL 1, it could be optional. Hash is in Hex string form
5	numSignatures	int	O	A number equal or higher to 1 representing the maximum number of signatures that can be created with this credential with a single authorization request.
6	expiresIn	int	M	The lifetime in seconds of the service access token. If omitted, the default expiration time is 3600 seconds (60 minutes).
7	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

4	authorizationRequest	String	M	The authorization request is presented as BASE64 encoding of JSON format that includes the attributes as below:
				{
				"authorizationNonce":"",
				"hash":"",
				"certificateThumbprint":"",
				"agreementUUID":"",
				"keyHandle":0
				}
				TSE device handles it by BASE64 decoding to get each value for processing.

2.26. signTSEForSignCloud

After receiving authorization data, TSE device will sign it by using stored KAK. The result signature will be sent through this API.

2.26.1. Prototype

```
SignCloudResp signTSEForSignCloud(SignCloudReq signCloudReq)
```

2.26.2. Parameter

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number
4	hash	String	M	Hash to be signed. For SCAL 1, it is mandatory and optional for SCAL 2.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				Hash is in Hex string form

5	signatureActivationData	String	M	This value is presented as BASE64 encode of a JSON data. The JSON contains 4 elements as below:
				{
				"authorizationSignature":"",
				"hash":"",
				"certificateThumbprint":"",
				"agreementUUID":"",
				"keyHandle":0,
				}
				authorizationSignature is created by TSE device using KAK and the signed data is authorizationNonce in authorizeTSEForSignCloud response

6	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction

SignCloudResp Attributes				
No	Name	Type	Require	Description
4	remainingCounter	int	M	Counter decreases once invalid Authorization Code provided. AgreementUUID will be blocked if counter goes to zero and it's automatically unblocked within 5 minutes. The default value for automatic unblock is 5 minutes, it could be reconfigured in the system
5	signatureValue	String	M/O	The PKCS#1 Signature as Base64 String will be sent to RelyingParty
6	remainingSignatures	int	M/O	The number of signatures could be created once authorization is made by calling authorizeTSEForSignCloud

2.27. requestAuthentication

Provide customer information such as:

- Full name
- Personal ID
- Address
- Mobile number (used for received SMS OTP)
- Both sides of scanned of identity document

After calling this function, customer will receive SMS OTP sent by Service Provider

2.27.1. Prototype

```
SignCloudResp requestAuthentication(SignCloudReq signCloudReq);
```

2.27.2. Parameter

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement number

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

SignCloudReq Attributes				
No	Name	Type	Require	Description
4	ownerUUID	String	O	System will find OWNER_ID based on ownerUUID and the newly registered certificate will be belonged to this owner. If this value is NULL, new owner will be created.
5	ownerUsername	String	O	ownerUsername and ownerPassword are both required not NULL to take effect. Newly created owner will be assigned this username/password. Error will be returned if the ownerUsername existed.
6	ownerPassword	String	O	
7	mobileNo	String	O	Mobile Number of client/customer. Known as contact address and used for SMS Authorize Code authentication. In case of the Relying Party keep the customer information confidentially, it should be absence
8	email	String	O	Email of client/customer and used for Email Authorize Code authentication In case of the Relying Party keep the customer information confidentially, it should be absence
9	certificateProfile	String	O	The profile of certificate will be used to enroll. Once this parameter is absence, the default certificate profile is used based on RelyingParty properties
10	promotion	int	O	The extra free day will be added for end-user certificate instead of only the day configured in certificate profile.
11	signingProfile	String	O	The profile decides the number of signing times the end-user can use. This value will be decreased once user performs signing.

SignCloudReq Attributes				
No	Name	Type	Require	Description
				User cannot sign if the signing counter is 0. If signing counter is -1, user's siging times is unlimited.
12	agreementDetails	AgreementDetails	M	Customer information. This one will be used as certificate information.
13	sharedMode	int	O	Which sharing mode will be used in the Remote Signing service. 1: PRIVATE_MODE 2: RP_SHARED_MODE 3: AGREEMENT_SHARED_MODE Default value is PRIVATE_MODE. It's meant once Relying Party don't use this parameter, the PRIVATE_MODE is used instead
14	postbackEnabled	Boolean	O	If postbackEnabled set **TRUE**, the Remote Signing service will response **REQUEST ACCEPTED** immediately. Once the certificate enrollment process is completely finalized, the Remote Signing will send the details based on the postbackUrl that already configured in RelyingParty properties Default value is **FALSE**, Postback mechanism shouldn't be used
15	csrRequired	Boolean	O	If csrRequired set **TRUE**, Remote Signing service will generate keypair for customer and build the CSR for the response.
16	credentialData	CredentialData	M	Credential information used for client-server handshake Cached feature is turn **ON/OFF**. Once the cached machine is **ON**, the Remote Signing just checked the credentialData in the handshake phase. Once credentialData is valid, our session will be valid in duration of 1 hour (this parameter

SignCloudReq Attributes				
No	Name	Type	Require	Description
				could be reconfigured in Remote Signing service). The Relying Party need to handshake again before the session is expired

In case of AgreementDetails, (*) is mandatory and (/*) is at least ONE in mandatory

AgreementDetails Attributes				
No	Name	Type	Require	Description
1	**personName (*)**	String	M/O	Name of customer
2	organization	String	M/O	Company name
3	organizationUnit	String	M/O	Department nam
4	title	String	M/O	Title of employee
5	email	String	M/O	Personal email
6	telephoneNumber	String	M/O	Personal phone number
7	**location (*)**	String	M	
8	**stateOrProvince (*)**	String	M	City or Province
9	country	String	M	Country (Just two letter) E.g: VN
10	**personalID (/*)**	String	M/O	Personal ID
	passportID (/*)	String	M/O	Personal Passport ID
11	taxID (/*)	String	M/O	Tax ID
	budgetID (/*)	String	M/O	Budget ID
12	applicationForm	Byte[]	O	Remote Signing Service application form
13	requestForm	Byte[]	O	Remote Signing Service request form for recovery, revocation …
14	authorizeLetter	Byte[]	O	Authorize Letter by Government/Corporate
15	photoIDCard	Byte[]	O	Scanned photo of ID Card that applied for personal certificate (In case only 1 file available)
16	photoFrontSideIDCard	Byte[]	O	Scanned front side of photo of ID Card that applied for personal certificate
17	photoBackSideIDCard	Byte[]	O	Scanned back side side of photo of ID Card that applied for personal certificate

AgreementDetails Attributes				
No	Name	Type	Require	Description
18	photoActivityDeclaration	Byte[]	O	Scanned photo of 'Initial Activity/Alteration Declation' that applied for corporate certificate

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result. Normally. Remote Signing will response **SUCCESSFULLY**
2	responseMessage	String	M	Message describes the result in detail.
3	billCode	String	M	Receipt for each transaction
4	certificateDN	String	M/O	Certificate Distinguished Name (DN) ***E.g: CN=Nguyen Van A, O=B Company,C=VN***
5	certificateSerialNumber	String	M/O	Certificate serial number E.g: 5405ABCDEF
6	certificateThumbprint	String	M/O	Certificate thumprint (certificate hash)
7	validFrom	Date	M/O	The date of certificate begin to be valid
8	validTo	Date	M/O	The date of certificate begin to be expired
9	issuerDN	String	M/O	Issuer Certifcate Distinguished Name (DN) ***E.g: CN=FPT Certification Authority, O=FPT-CA,C=VN***
10	certificate	String	M/O	The data of certificate in PEM format
11	certificateStateID	int	M/O	Certificate state ID • INITIALIZED = 2 • GENERATED = 3 • OPERATED = 4 • REVOKED = 5 • EXPIRED = 6 • DECLINED = 7 • RENEWED = 8 • RENEWED_EXPIRED = 9 • REVISED = 10 • RENEWED_KEEP_SN = 11

SignCloudResp Attributes				
No	Name	Type	Require	Description
				• BLOCKED = 12 REVISED_KEEP_SN = 13
12	signingCounter	int	M/O	How many times the certificate is used for signing. The default value is 1. For document signing this value should be greater than 1

CredentialData Attributes				
No	Name	Type	Require	Description
1	username	String	M	Username of relyingParty provided by Remote Signing
2	password	String	M	Password of relyingParty provided by Remote Signing
3	signature	String	M	Signature of relyingParty provided by Remote Signing
4	pkcs1Signature	String	M	Value is signed from client private key. Key is generated and provided by Remote Signing Value=username + password + signature + timestamp
5	timestamp	String	O	Current timestamp, format in yyyyddmmHHMMss or epoch time If timestamp is absence, the pkcs1Signature is fixed for all transactions

2.28. respondAuthentication

OTP verification will be performed in this function. Server will record the OTP validity, no any certificate issued in this function.

2.28.1. Prototype

```
SignCloudResp respondAuthentication(SignCloudReq signCloudReq);
```

2.28.2. Parameter

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode

SignCloudReq Attributes				
No	Name	Type	Require	Description
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	billCode obtained from response of **requestAuthentication**
5	authorizeCode	String	M	Authorize Code provided by customer
6	credentialData	CredentialData	M	Credential information used for client-server handshake

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail. Normally, this responses **SUCCESSFULLY**
3	billCode	String	M	Receipt for each transaction
4	remainingCounter	int	M	Counter decreases once invalid Authorization Code provided. AgreementUUID will be blocked if counter goes to zero and it's automatically unblocked within 5 minutes. The default value for automatic unblock is 5 minutes, it could be reconfigured in the system

2.29. approveAuthentication

Once this API is called, server will check the previous state of OTP authentication. If the OTP is valid, customer certificate will be generated and the document will be signed.

Document information should be provided in this API. Server also generates the PDF file based on that information.

The signed document could be returned by many ways: SFTP, FPT or synchronously in the response.

2.29.1. Prototype

```
SignCloudResp approveAuthentication(SignCloudReq signCloudReq);
```

2.29.2. Parameter

SignCloudReq Attributes				
No	Name	Type	Require	Description
1	relyingPartyBillCode	String	O	Relying Party's billcode
2	relyingParty	String	M	In case of many services are running on Remote Signing. RelyingParty is used to determine which client is calling to.
3	agreementUUID	String	M	Identify the client/customer/agreement ID
4	billCode	String	M	billCode obtained from response of **requestAuthentication**
6	messagingMode	int	O	Which messaging mode will be used **1**: ASYNCHRONOUS_CLIENTSERVER (default) **2**: ASYNCHRONOUS_SERVERSERVER **3**: SYNCHRONOUS If default value 0 is used, the API downloadSignedFileForSignCloud will be used for getting the signed file
7	credentialData	CredentialData	M	Credential information used for client-server handshake
8	templateId	String	M	Template to generate PDF
9	signCloudMetaData	signCloudMetaData	O	SignCloudMetaData contains additional signer properties which are used for PDF signature display. - Coordinate - Background image - Signer location - Signer reason - Text color In case of XML sigining, it will indicate which type should be applied XMLDSig, XML-XaDES, signature zone

SignCloudReq Attributes				
No	Name	Type	Require	Description
				Once this parameter is not used, Remote Signing should use the default value that already configured in the system In case of counter signing process, SignCloudMetaData included 2 MetaDatas for customer and Relying Party signer properties
10	documentAttributes	DocumentAttributes	M	Document content to be filled into template

SignCloudResp Attributes				
No	Name	Type	Require	Description
1	responseCode	int	M	Code describes the result.
2	responseMessage	String	M	Message describes the result in detail. Normally, this responses **SUCCESSFULLY**
3	billCode	String	M	Receipt for each transaction
4	signedFileData	Byte[]	M/O	File binary that be signed
5	mimeType	String	M/O	In case of PDF file that be downloaded, mimeType is '**application/pdf**'
6	signedFileUUID	String	M/O	Signed file UUID that identified by our Remote Signing service
7	multipleSignedFileData	List<MultipleSignedFileData>	M/O	In case of signing multiple files, the multiple signed files will be responded in this object.
8	xmlSignature	String	M/O	XML Signature (XMLDSig or XML-XadDES)

DocumentAttributes Attributes				
No	Name	Type	Require	Description
1	templateName	String	M	Template name
2	attributes	HashMap<String,String>	M	Document attributes are used to generate PDF

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

3.RESPONSE CODE AND MESSAGE

0	SUCCESSFULLY
1000	INVALID IP ADDRESS
1001	INVALID CREDENTIAL DATA
1002	INVALID PARAMS
1003	UNEXPECTED EXCEPTION
1004	INVALID AUTHORIZATION CODE
1005	AUTHORIZATION BLOCKED
1006	AUTHORIZATION CODE TIMEOUT
1007	REQUEST ACCEPTED
1008	AGREEEMENT NOT FOUND
1009	CERTIFICATE IS NOT ENROLLED
1010	AGREEMENT EXISTED
1011	UNSUPPORTED OPERATION
1012	ACCESS DENIED
1013	AGREEMENT NOT READY
1014	CERTIFICATE IS EXPIRED
1015	BILLCODE TIMEOUT DUE TO UNEXPECTED EXCEPTION WHILE SIGNING
1016	FILE IS BEING PROCESSED
1017	AGREEMENT REVOKED
1018	SUCCESSFULLY. YOUR PASSCODE NEED TO BE CHANGED
1019	FUNCTION ACCESS DENIED
1020	FAILED TO CHECK REVOCATION
1021	CERTIFICATE REVOKED
1022	CERTIFICATE UNKNOWN
1023	SHARED AGREEMENT NOT FOUND
1024	SHARED AGREEMENT INVALID SHARED MODE
1025	FILE NOT FOUND
1026	MULTIPLE CLIENT MACHINE YSING THE SAME AGREEMENT UUID
1027	SIGNING TRANSACTION CAN NOT BE COMMITED SINCE SIGNING ERROR
1028	CREDENTIAL IS EXPIRED
1029	EMAIL IS ALREADY REGISTERED
1030	DOCUMENT IS NOT SIGNED YET
1031	CERTIFICATE IS NOT ASSIGNED TO OWNER

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

1032	USERNAME IS ALREADY EXISTED
1033	SIGNING COUNTER IS EXCEEDED
1034	ENTERPRISE CODE IS EXISTED
1035	PERSONAL CODE IS ALREADY EXISTED
1036	CERTIFICATE ISSUANCE ERROR
1037	MOBILE APP SESSION IS EXPIRED. PLEASE AUTHORIZE AGAIN
1038	SAD IS INVALID
1039	NUMBER OF ALLOWED OTP GENERATION EXCEEDED
1040	INVALID AUTH METHOD
1041	THE DEFAULT PASSCODE MUST BE CHANGED
1042	AGREEMENT HAS NOT PERFORMED SIGNING YET
1043	IDENTITY DOCUMENT REQUIRED
1044	AUTHENTICATION FAILED
1045	FAILED TO SEND OTP
1046	SIGNER INFO REQUIRED FOR CHECKING
1047	INVALID SIGNER INFO PROVIDED
1048	NO CERTIFICATE FOUND
1049	NO KEY FOUND
1050	SIGNATURE POSITION NOT FOUND

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

APPENDIX A: CREDENTIAL DATA DETAILS

Create PKCS#1 signature in Java

```
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Enumeration;
-----------------------------------------------------------------------
KeyStore keystore = KeyStore.getInstance("PKCS12");
InputStream is = new FileInputStream("cloudfca.p12");
keystore.load(is, "12345678".toCharArray());

Enumeration<String> e = keystore.aliases();
String aliasName = "";
PrivateKey key = null;
while (e.hasMoreElements()) {
      aliasName = e.nextElement();
      key = (PrivateKey) keystore.getKey(aliasName, "12345678".toCharArray());
      if(key != null)
            break;
}
Signature sig = Signature.getInstance("SHA1withRSA");
sig.initSign(key);
sig.update("To be signed".getBytes());
String pkcs1Signature = DatatypeConverter.printBase64Binary(sig.sign());
```

Create PKCS#1 signature in Python

```
# easy_install.exe http://www.voidspace.org.uk/python/pycrypto-2.6.1/pycrypto-
2.6.1.win-amd64-py2.7.exe
# python get-pip.py
# python -m pip install --upgrade pip
# pip install pyopenssl
from Crypto.Signature import PKCS1_v1_5
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
from OpenSSL import crypto
from _helpers import _parse_pem_key
from _helpers import _to_bytes
from _helpers import _b64encode

p12 = crypto.load_pkcs12(open("cloudfca.p12", 'rb').read(), "12345678")
p12.get_certificate()     # (signed) certificate object
p12.get_privatekey()      # private key.
p12.get_ca_certificates() # ca chain.

pKeyPem = crypto.dump_privatekey(crypto.FILETYPE_PEM, p12.get_privatekey())

parsed_pem_key = _parse_pem_key(_to_bytes(pKeyPem))

message = 'To be signed'
key = RSA.importKey(parsed_pem_key)
h = SHA.new(message)
signer = PKCS1_v1_5.new(key)
signature = signer.sign(h)
print("Signature: "+_b64encode(signature))
```

Helper class

```
import base64
import json
import six

def _parse_pem_key(raw_key_input):
    """Identify and extract PEM keys.
    Determines whether the given key is in the format of PEM key, and extracts
    the relevant part of the key if it is.
    Args:
        raw_key_input: The contents of a private key file (either PEM or
                       PKCS12).
    Returns:
        string, The actual key if the contents are from a PEM file, or
        else None.
    """
    offset = raw_key_input.find(b'-----BEGIN ')
    if offset != -1:
        return raw_key_input[offset:]

def _to_bytes(value, encoding='ascii'):
    result = (value.encode(encoding)
              if isinstance(value, six.text_type) else value)
    if isinstance(result, six.binary_type):
        return result
    else:
        raise ValueError('%r could not be converted to bytes' % (value,))

def _b64encode(raw_bytes):
      raw_bytes = _to_bytes(raw_bytes, encoding='utf-8')
      return base64.b64encode(raw_bytes)
```

MOBILE-ID TECHNOLOGIES AND SERVICES JSC
19 Dang Tien Dong, An Phu Ward, District 2nd, Ho Chi Minh City, Vietnam
Tel: (84-28) 3636 6015- Fax: (84-28) 3636 6016 – Hotline: 1900 6884
info@mobile-id.vn – https://www.mobile-id.vn

APPENDIX B: SIGNCLOUD METADATA DETAILS

SignCloudMetaData has two attributes

- singletonSigning: Define the signature properties for customer
- counterSigning: Define the signature properties for organization

Both singletonSigning and counterSigning are Key-Value objects.

KEY	EXAMPLE VALUE	DESCRIPTION
PAGENO	First;Last;1;2;..	The page, signature will be put on. Possible value: 1; 2; 3…; First; Last
POSITIONIDENTIFIER	CHỮ KÝ KHÁCH HÀNG	Signature will be placed based on the detected string. Ex: "Customer signature" Server will find the coordinates of "Customer signature" and put the signature based on that coordinates.
RECTANGLEOFFSET	-10,120	This will be used with POSITIONIDENTIFIER. Let say POSITIONIDENTIFIER is "Customer signature" and the found coordinate is of "C" character. RECTANGLEOFFSET is the offset from "C" will help us correctly put the signature on.
RECTANGLESIZE	170,70	The size of rectangle signature
VISIBLESIGNATURE	True;False	True – Signature is visible False – Signature is invisible
VISUALSTATUS	True;False	True/False – Visiable/Hide the green stick validation
IMAGEANDTEXT	True;False	Showing signer info and image (logo) at the same time
TEXTDIRECTION		Used with IMAGEANDTEXT. Possible value • LEFTTORIGHT • RIGHTTOLEFT • TOPTOBOTTOM • BOTTOMTOTOP

		• OVERLAP
SHOWSIGNERINFO	True;False	Showing signer info – CN of certificate
SIGNERINFOPREFIX	Sign by:	The prefix of signer info. Ex: **Sign by:** <CN>
SHOWDATETIME	True;False	Showing signing datetime
DATETIMEPREFIX	Sign on:	The prefix of signing datetime. Ex: **Sign on:** <timestamp>
DATETIMEFORMAT	dd-MM-yyyy HH:mm:ss	Defining datetime format. Ex: dd-MM-yyyy HH:mm:ss
SHOWREASON	True;False	Showing signing reason
SIGNREASONPREFIX	Sign for:	The prefix of reason. Ex: **Sign for:** Approval
SHOWLOCATION	True;False	Showing signing location
LOCATION	HCM	Signing location. Ex: HCM
LOCATIONPREFIX	Sign at:	The prefix of location. Ex: **Sign location:** HCM
TEXTCOLOR	back;red	Text color. Possible value: yellow; blue; cyan; dark_gray; gray; green; light_gray; magenta; orange; pink; red; white
SHOWTELEPHONE	True;False	Showing telephone (telephone field of certificate)
TELEPHONEPREFIX	Điện thoại:	The prefix of telephone. Ex: **Điện thoại:** 090888888
SHOWENTERPRISEID	True;False	Showing taxID of organization (find MST: or MNS: in certificate)
ENTERPRISEIDPREFIX	MST	The prefix of taxID of organization. Ex: **MST:** 111111
SHOWPERSONALID	True;False	Showing personal ID (find CMND, CCCD or HC in certificate)
PERSONALIDPREFIX	CMND/CCCD	The prefix of personal ID. Ex: CMND: 222222
SHOWSERIALNUMBER	True;False	Showing certificate serial number
SERIALNUMBERPREFIX	SN:	The prefix of certificate serial number
SHOWORGANIZATION	True;False	Showing organization (O field of certificate)

ORGANIZATIONPREFIX	Tổ chức:	The prefix of organization
SHOWTITLE	True;False	Showing title (T field of certificate)
TITLEPREFIX	Chức vụ:	The prefix of title
ONLYCOUNTERSIGNENABLED	True;False	If True, only organization signature is applied.
COUNTERAGREEMENTUUID		Organization agreementUUID
COUNTERAGREEMENTPASSCODE		Organization passcode
COUNTERSIGNENABLED	True;False	If True, Organization and Customer signature are both applied.
SHOWISSUERINFO	True;False	Showing Issuer name of certificate
ISSUERINFOPREFIX		The prefix of issuer name
BACKGROUNDIMAGE	Base64 of image	Background image of signature field. Image will be resized to fix the signature rectangle.
PAGESFORINITIALSIGNATURE	all;1,2,3..	Initial signature
SHADOWSIGNATUREPROPERTIES	all	Multiple signatures fields for one time authentication
PDFPASSWORD	12345678	PDF protected password
LINESPACING	1.5	Space between lines
FONTSIZE	12	Font size
FONTSTYLE	• ARIAL • TAHOMA • TIMES • VERDANA • SCRIPTURE • HELVETICA	Font style