

CorplD API Specifications for e-services

Version: 1.0.4

June 2026

© The Government of the Hong Kong Special Administrative Region
of the People's Republic of China

The contents of this document remain the property of and may not be reproduced in whole
or in part without the express permission of the Government of
the Hong Kong Special Administrative Region of the People's Republic of China

Document Revision History

Ver. No	Date	Section Affected	Summary of Changes
1.0.0	31/03/2026		Initial release for e-service with CorpID
1.0.1	13/04/2026		Add website/app in same device
1.0.2	14/04/2026		Add App to App, update sequence diagram
1.0.3	30/04/2026		Updated Appendix A Profile fields Updated form pre-filling APIs request parameter Updated Anonymous form pre-filling workflow Updated getToken to return the "eServiceRole"
1.0.4	24/06/2026		Enhance Description of API detail Updated Appendix A to have "nonHKResidentFields" Update Digital Signing API to have "signType" and "requireAuthProof" Update Anonymous Digital Signing API to have CorpID get iAM Smart Signing Result Update Anonymous Digital Sign APIs response Update "iAMContent"

			and "iAMSecretKey" Update Obtain Profiles API response
--	--	--	--

Contents

CORPID API SPECIFICATIONS FOR E-SERVICES	1
1. OVERVIEW.....	7
1.1. INTRODUCTION OF API FUNCTIONS.....	7
1.2. CORPID PROFILES	8
2. SECURITY.....	9
2.1. HTTPS	9
2.2. CALLBACK VERIFICATION FROM CORPID SYSTEM.....	9
2.3. API DATA ENCRYPTION AND DECRYPTION	9
2.4. COMMON PARAMETERS	31
2.5. AUTHORISATION SCOPES.....	34
3. USE CASES AND SCENARIOS.....	35
3.1. AUTHENTICATION	35
3.2. OBTAIN PROFILES INFORMATION (FOR B/D AND GOVERNMENT BUREAU ONLY)	44
3.3. FORM PRE-FILLING WITH SERVICE LOGIN	45
3.4. FORM PRE-FILLING WITHOUT SERVICE LOGIN	48
3.5. DIGITAL SIGNING WITH SERVICE LOGIN.....	52
3.6. DIGITAL SIGNING WITHOUT SERVICE LOGIN.....	56
3.7. REQUEST TOKEN EXCHANGE.....	61
3.8. DOCUMENT SHARING WITH SERVICE LOGIN.....	62
4. API SPECIFICATION	66
4.1. AUTHENTICATION BY REQUESTING QR PAGE/ BROKER PAGE FOR ANONYMOUS REQUEST	66
4.2. GET ACCESS TOKEN.....	68
4.3. GET CORPORATION LIST	74
4.4. REQUEST TOKEN EXCHANGE FOR SWITCHING CORPORATION WITH UBI.....	77
4.5. OBTAIN PROFILES INFORMATION	83
4.6. REQUEST FORM PRE-FILLING.....	87
4.7. FORM PRE-FILLING CALLBACK.....	92
4.8. REQUEST ANONYMOUS FORM PRE-FILLING.....	94
4.9. CALLBACK WITH AUTHCODE FOR ANONYMOUS	98
4.10. GET ANONYMOUS FORM PRE-FILLING RESULT	100
4.11. REQUEST DIGITAL SIGNING	103

4.12.	DIGITAL SIGNING CALLBACK.....	110
4.13.	REQUEST PDF DIGITAL SIGNING.....	113
4.14.	PDF DIGITAL SIGNING CALLBACK	120
4.15.	ACKNOWLEDGES DIGITAL SIGNING RESULT	122
4.16.	REQUEST ANONYMOUS DIGITAL SIGNING	125
4.17.	REQUEST ANONYMOUS PDF DIGITAL SIGNING	131
4.18.	GET ANONYMOUS DIGITAL SIGNING RESULT	136
4.19.	GET ANONYMOUS PDF DIGITAL SIGNING RESULT	139
4.20.	REQUEST CORPORATION DIGITAL SIGNING (FOR INTEGRATED SIGNING)	142
4.21.	REQUEST CORPORATION PDF DIGITAL SIGNING (FOR INTEGRATED SIGNING)	146
4.22.	REQUEST DOCUMENT SHARING.....	150
4.23.	CALLBACK TO RECEIVE DOCUMENT SHARING	153
4.24.	REQUEST FORM PRE-FILLING (FOR APP E-SERVICE)	156
4.25.	REQUEST DIGITAL SIGNING (FOR APP E-SERVICE)	161
4.26.	REQUEST PDF DIGITAL SIGNING (FOR APP E-SERVICE)	168
4.27.	REQUEST DOCUMENT SHARING (FOR APP E-SERVICE)	174
4.28.	OPEN CORPID MINI APP FOR DOCUMENT SHARING	178
5.	APPENDIX A	180
5.1.	CORPPROFILEFIELDS SCHEMA.....	180
5.2.	NONHKRESIDENTFIELDS SCHEMA.....	187
5.3.	ECORPFIELDS SCHEMA.....	187
5.4.	“IAM SMART” PROFILE FIELD AND “E-ME” FIELD SCHEMA.....	189
6.	APPENDIX B	190
6.1.	SUPPORTED VALUE AT SOURCE PARAMETER.....	190
7.	APPENDIX C	191
7.1.	IDENTIFY DOCUMENT TYPE.....	191
8.	APPENDIX D	191
8.1.	RETURN CODE	191
8.2.	HTTP STATUS CODE.....	193
9.	APPENDIX E	195
9.1.	CORPORATE TYPE	195
9.2.	CORPORATE STATUS	196

1. OVERVIEW

1.1. INTRODUCTION OF API FUNCTIONS

Corpid functions are built in the form of RESTful Application Program Interface (API), which could be accessed by registered e-services. Corpid makes reference to OAuth 2.0 for authentication and authorisation amongst Corpid users, e-Services and the Corpid System. e-Services adopting Corpid are required to provide RESTful callback APIs to receive API responses from the Corpid System. Corpid APIs support the following functions:

- **Authentication**
e-Services can make use of the Authentication API provided by Corpid to verify the identity of the users in a simple and secure way. The API can be used in various scenarios, such as user login, membership system, booking system, etc.
- **Form Pre-Filling with Service Login**
e-Services can make use of the Profiles API to retrieve data of "corpProfileFields" and "eCorpFields" in streamline workflow. The API can be used in various scenarios, such as account opening, account linkup, form pre-filling, etc.
- **Form Pre-filling without Service Login (aka Anonymous Form Pre-Filling)**
e-Services can make use of the Anonymous Form Pre-filling API to request corporate information without prior Corpid authentication.
- **Digital Signing with Service Login**
e-Services can make use of Digital Signing API to enable Corpid user to complete online digital signing with legal backing after user authentication on e-Services using Corpid. It can be used in many cases, such as digital signing an online application form, a contract, or an agreement.
- **Digital Signing without Service Login (aka Anonymous Digital Signing)**
e-Services can make use of the Anonymous Digital Signing API to enable Corpid user to complete online digital signing without prior Corpid authentication.
- **Document Wallet**
The Corpid Platform's Document Wallet will store the digital licences and permits issued by the Government bureaux/departments, serving as a supplement to the physical documents. Corpid user can make use of this API to select and upload relevant document(s) to e-Services.

1.2. CORPID PROFILES

Corpid Profiles contain the corporate information provided by Corpid.

- “corpProfileFields”
The “corpProfileFields” consist of the account information of the corporation checked against various data sources during Corpid registration and on-going account maintenance.
- “nonHKResidentFields”
The “nonHKResidentFields” consist of the non-Hong Kong Resident user information that Corpid would obtain during the Corpid user onboarding, its sole information from the trusted party.
For Hong Kong Resident information, please request with iAM Smart’s “Profile Fields” and “e-Me Fields”.

In addition, the following information can be input by Corpid Users optionally:

- “eCorpFields”
In addition to the account information, Corpid user can input and update additional corporate information optionally for the purpose of online form pre-filling.

Details of the "corpProfileFields" and "eCorpFields" can be found in Appendix A.

2. SECURITY

2.1. HTTPS

E-service endpoint shall support TLS 1.2 or compatible for using CorpID APIs.

2.2. CALLBACK VERIFICATION FROM CORPID SYSTEM

CorpID System supports e-service to verify on the transport layer that the callback request it receives comes from CorpID System. If e-service chooses to verify, it can request the public key certificate of CorpID System which from the Recognized Certification Authorities in Hong Kong during the onboarding process and configures the certificate in the corresponding location of its HTTPS server to implement the client authentication function. Specific configurations vary depending on the HTTPS server or the reverse proxy used, and the corresponding administration manual needs to be consulted. For example, if using Nginx, e-service should set `ssl_client_certificate` and `ssl_verify_client` in its configuration file to authenticate client requests.

2.3. API DATA ENCRYPTION AND DECRYPTION

HTTPS will be used to secure communication channels between e-services and the CorpID System. However, there is still a chance that the data transferred by HTTPS could be eavesdropped or manipulated by malicious service providers in some special situations. To better protect the data in transit, an additional layer of data encryption will be applied to all APIs POST request (except the one that e-service request for getting symmetric encryption key). Callbacks from CorpID System will also be encrypted using the method described herein.

The overall idea of API data encryption is:

1. E-service requests for data encryption key from CorpID System;
2. CorpID System generates a symmetric data encryption key (CEK);
3. CorpID System stores the generated key and returns to the e-service who initiated the request in secure manner e.g. encrypted by the e-Service's public key KEK;
4. For subsequent interactions between e-service and CorpID System, business data will be encrypted by the generated key;
5. Both CorpID System and e-service will use the same key to decrypt the API data.
6. If the generated key is expired, e-service has to request a new key and repeat the above process.

2.3.1. Encryption and Decryption Workflow

The following diagram describes detailed workflow on how e-service can request data encryption key, perform encryption and decryption with the key, and request for another new encryption key if the existing key is expired:

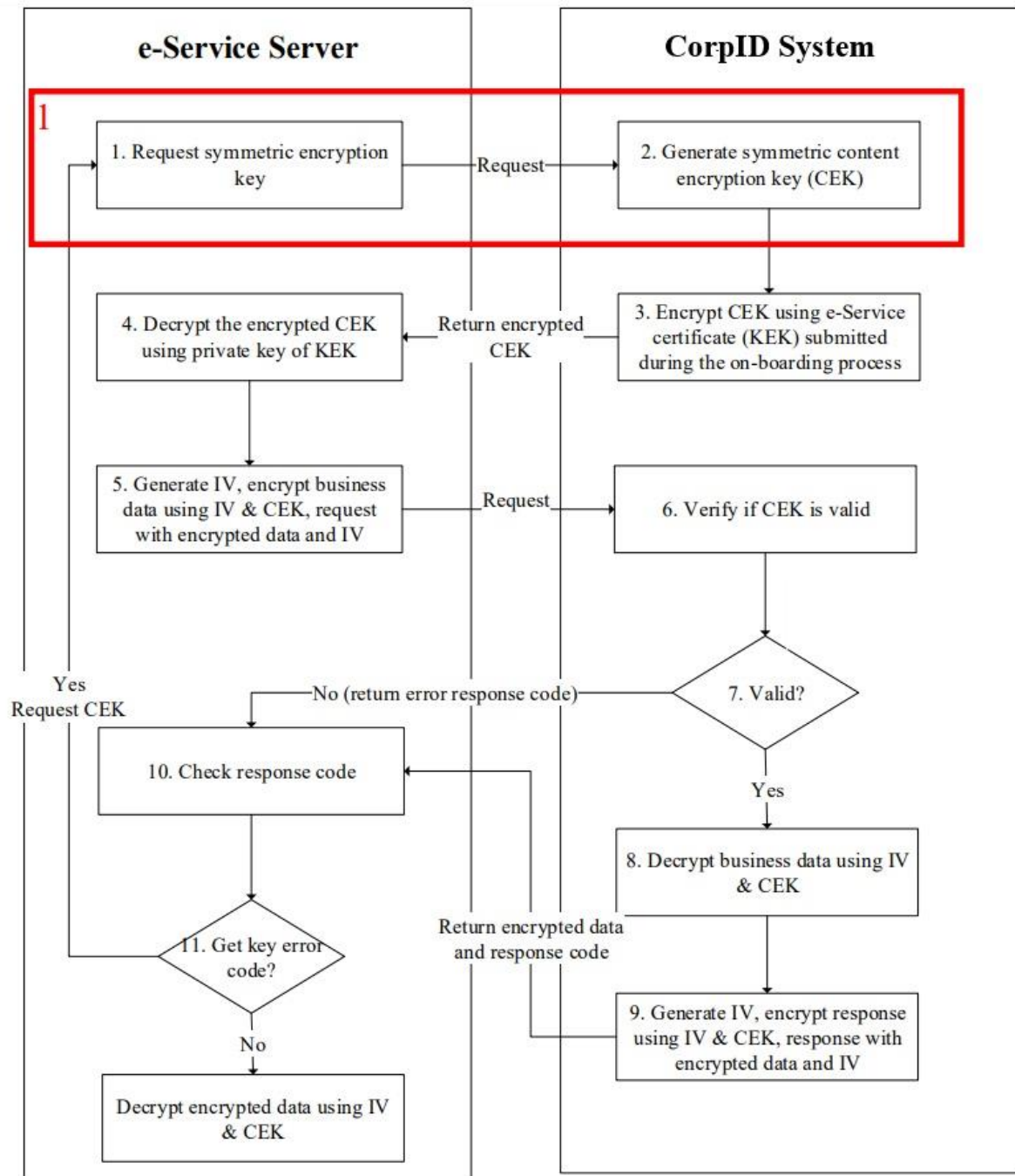


Figure - 1 Application of Content Encryption Key and Procedures of Encryption and Decryption (Perform by e-service)

The APIs for e-service to request Content Encryption Key are marked in red.

High Level Workflow Description of encrypting API request data using Content Encryption Key:

- During the onboarding process, e-service has to upload its certificate which was issued by Recognised Certification Authorities. The certificate contains a public key (denoted as KEK, the Key Encryption Key) that will be used in the following steps.
- CorpID System provides API to e-service for retrieving a symmetric key (denoted as CEK, the Content Encryption Key). Such key will be used to encrypt/decrypt subsequent API data. AES256 symmetric encryption algorithm will be adopted for encrypting the data. CEK will be expired after specified period (refer to parameter at 2.3.7.1) of generation.
- When CorpID System received request from e-service for retrieving CEK, CorpID System will employ the RSA algorithm and use e-service's KEK to encrypt the CEK. Then CorpID System will send the encrypted CEK to e-service. e-service can use the private key of the KEK to decrypt and obtain the symmetric key CEK.
- Before e-service calling APIs of CorpID System, e-service should check whether the symmetric data encryption key (CEK) exists and is valid. If e-service has no such CEK or the CEK is expired, then e-service should request a new CEK from CorpID System. The CEK will be used to encrypt subsequent API data until the key expires.
- Upon receiving the request from e-service, CorpID System will check whether e-service's data encryption key CEK exists and is still valid. If CEK does not exist or expired, error message will be returned to e-service. If CEK is still valid, the key will be used to decrypt the request data. Same CEK will also be used to encrypt respective response or callback data for returning to e-service.

High Level Workflow Description of Callback Encryption and Decryption:

- Before CorpID System initiates a callback, it will check if the data encryption key (CEK) of the corresponding e-service is still valid. If the CEK is expired or does not exist, CorpID System will regenerate a new data encryption key for that e-service.
- CorpID System uses the CEK to encrypt the callback body. Then such CEK will be encrypted by KEK of the e-service. The encrypted callback data and the encrypted CEK will be returned to e-service through the callback.

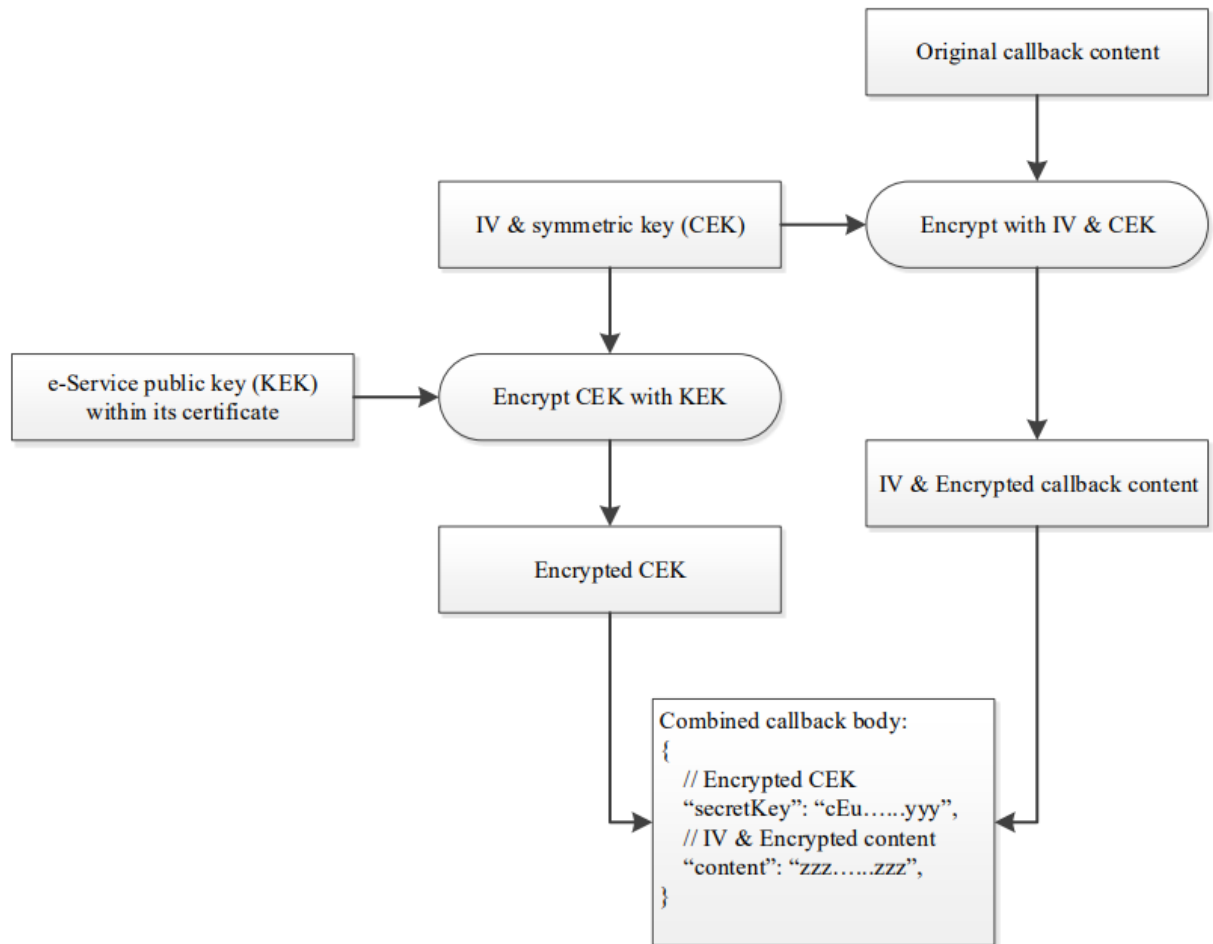


Figure - 2 Callback Encryption Process (Perform by CorpID System)

- After e-service receives the callback, e-service should use the private key of the KEK to decrypt the encrypted data encryption key (CEK). Then the decrypted data encryption key (CEK) can be used to decrypt the callback data.

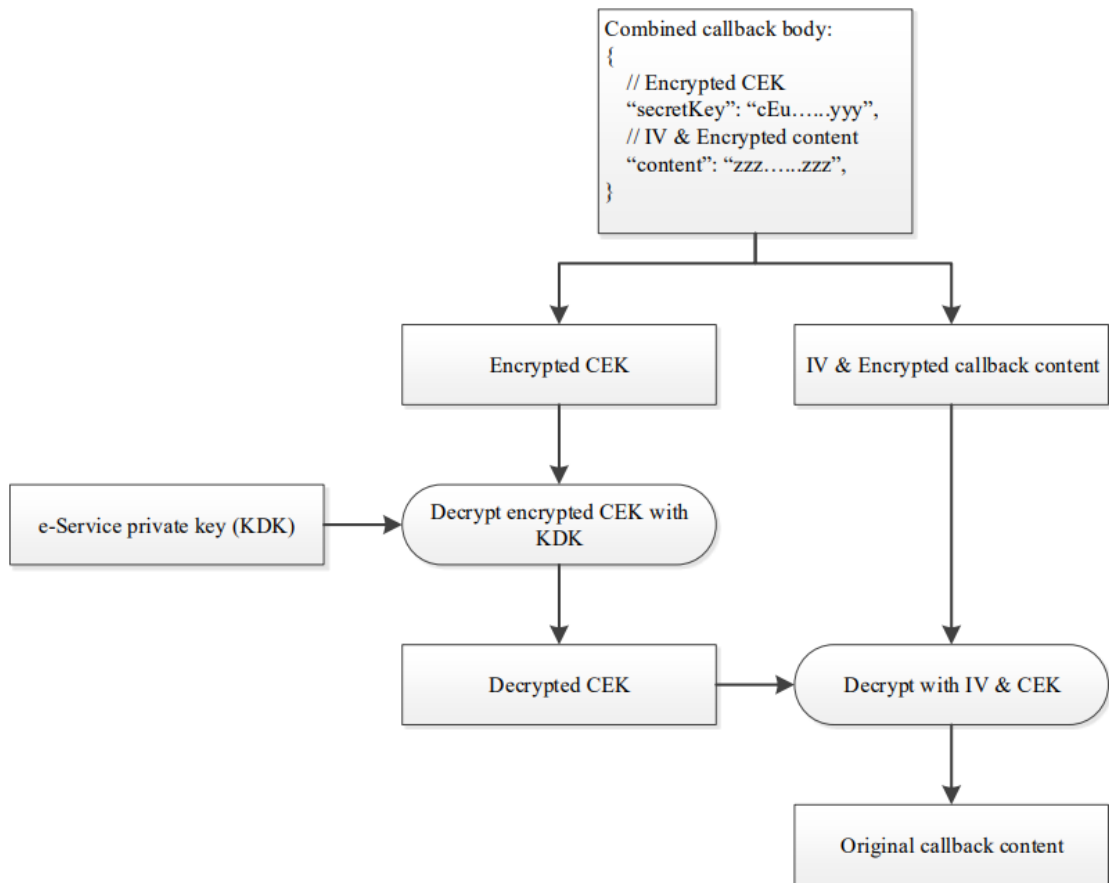


Figure - 3 Callback Decryption Process (Perform by e-service)

2.3.2. Data Encryption and Decryption for iAM Smart Data fields passed through CorpID Server

To better protect the data in transit, an additional layer of data encryption will be applied to fields passed between iAM Smart System to e-service server through CorpID API calls.

2.3.2.1. e-Service request to CorpID with iAM Smart Parameter

For iAM Smart integrated API, beside CorpID's "*cAccessToken*" and "*cOpenID*", e-service should also have the iAM Smart's "*accessToken*" and "*openID*" if the request related to iAM Smart information.

The iAM Smart's "*accessToken*" and "*openID*" should be input as below JSON format and used iAM Smart CEK encrypted before passed to CorpID and saved as "*iAMToken*":

Field Name	Data Type	Description	Values/Remark
accessToken	String	AccessToken value	
openID	String	Tokenised ID value	

```
// Line breaks are for legibility only.

// {

//   "accessToken": " b28a31e88bxxxxxxxx3d554c481d8da",

//   "openID": " 2d49358a23a1xxxxxxxxxxxx57506c2a5f462e60c87"

// }

// iv + ciphertext

"iAMToken":
"AAAADMBVmPKZZmm6/DBASQjG/0Ns11oBVKk8OcaQLsjim13XD/c1sZAkQfQQLiTFCxutL9TDCyV
xh1pUNQk//9VtPKXqmPbB1S/SgeNHYXhgbnQWvNW6Sx4pUcI440JbmUuU5JFWEnGVibJkmRGZ5QJ
RYXG89BI0bcjmrpMOnim0TKQ="
```

2.3.2.2. CorpID response / callback to e-Service with iAM Smart Content

For API response / callbacks from CorpID server, e-services should decrypt the field

“iAMContent” with iAM Smart CEK, and refer to iAM Smart API’s specified response JSON accordingly.

2.3.3. Proper Handling of Encryption and Decryption

E-service shall handle the following scenarios properly in their system design:

- Due to unexpected reasons, e-service may receive Encryption/Decryption Error Code. e-service shall perform retry of the “Request Symmetric Content Encryption Key” API specified in section 2.3.7.1 of this document to obtain the latest CEK and try to decrypt content with such CEK when it is received.
- Due to operational needs, the “expiresIn” response parameter of the “Request Symmetric Content Encryption Key” API specified in section 2.3.7.1 of this document may dynamically change, e-service shall calculate the expiry time of the CEK with “issueAt” and “expiresIn” response parameters.
- Under certain circumstances, CorpID API service may be unavailable for a certain period of time. The e-service shall prepare the fallback procedures (e.g. switch to paper mode, prevent users from accessing CorpID functions, etc.).
- E-service shall store the CEK to avoid unnecessary “Request Symmetric Content Encryption Key” API request (see section 2.3.7.1 of this document).
- Even a new KEK is configured for effective in a specified period, the existing CEK will still be returned through the “Request Symmetric Content Encryption Key” API specified in section 2.3.7.1 of this document and encrypted by the old KEK until the expiry of the existing CEK. To use the new KEK, e-service shall request a new CEK through the “Revoke Symmetric Content Encryption Key” and “Request Symmetric Content Encryption Key” APIs specified in sections 2.3.7.2 and 2.3.7.1 respectively of this document or to request a new CEK after its expiry through the same APIs.
- e-service shall also store the “iAM Smart” CEK to avoid unnecessary “Request Symmetric Content Encryption Key” in “iAM Smart” API request, to process the encryption of “iAMToken” in request APIs and perform retry of the “Request Symmetric Content Encryption Key” API in “iAM Smart”, “expiresIn” handling, and fallback procedures according to “iAM Smart” API Specification.

2.3.4. Encryption and Decryption Scope (From e-service perspective)

API Type	Parameter Type	Parameter Type/Field	Will Encrypt?	Encryption Rules
E-service request and response	Request parameters	request body	YES	1. Convert the request body into JSON string 2. Use "AES/GCM/NoPadding" to encrypt the byte array of the JSON string 3. Base64 encode the encrypted result and set the encoded string to the content field of the request body
		Field "iAMToken"	YES	1. Convert the field body into JSON string 2. Use "AES/GCM/NoPadding" to encrypt the byte array of the JSON string with "iAM Smart" CEK 3. Base64 encode the encrypted result and set the encoded string to the content field of the request body
		request header	NO	N/A
		url parameters	NO	N/A
	Response parameters	txID	NO	N/A
		code	NO	N/A
		msg	NO	N/A
		content	YES	1. Base64 decode the encrypted result and set the encoded string to the content field of the response body 2. Use "AES/GCM/NoPadding"

				to decrypt the byte array of the JSON string 3. Convert the content field of response body into JSON string
		Field "iAM"	YES	1. Base64 decode the encrypted result and set the encoded string to the content field of the response field 2. Use "AES/GCM/NoPadding" to decrypt the byte array of the JSON string with "iAM Smart" CEK 3. Convert the content field of response body into JSON string
CorpID callback	request parameters	request body	YES	1. Convert the content field of request body into JSON string 2. Using "AES/GCM/NoPadding" to encrypt the byte array of the JSON string 3. Base64 encode the encrypted result and set the encoded string to the content field of the request body 4. Encrypt the symmetric data encryption key (CEK) with the public key certificate (KEK). Base64 encode the encryption result and set the encoded string to the key field of the request body
		Field "iAMContent"	YES	1. Base64 decode the encrypted result and set the encoded string to the content field of the

				response field 2. Use "AES/GCM/NoPadding" to decrypt the byte array of the JSON string with "iAM Smart" CEK 3. Convert the content field of response body into JSON string
		request header	NO	N/A
		url parameters	NO	N/A
	response parameters	N/A	NO	N/A

2.3.5. Request and Callback Examples

- e-service calls CorpID API

Request Parameter

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/auth/getToken

// Request Headers

clientID: "edae2e2529ff46228af1e4d18c8405d1"

signatureMethod: "HmacSHA256"

signature: "5X42Y1B7MEd8Mm%2BonwjiQz9VCZkkrntADskXsYntavU%3D"

timestamp: 1557048906183

nonce: "e893647dc4204eb9b7b8eddd527b687c"

// Unencrypted Request Body

// Clear text before encryption

// {

//   "cAuthCode": "xxxa42e76bf4cb0846a68e6d83d6096",

//   "grantType": "authorization_code"

// }

// iv + ciphertext

{

  "content":

  "AAAAADMBVmPKZZmm6/DBASQjG/0Nsl1oBVKk8OcaQLsjiM13XD/clSZAkQfQQLiTFCxutL9TDCyV

xh1pUNQk//9VtPKXqmPbB1S/SgeNHYXhgbnQWvNW6Sx4pUcI440JbmUuU5JFWEnGViBjKmRGZ5QJ

RYXG89BI0bcjmrpMOnim0TKQ="

}
```

Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
```

```
"txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

"code": "M00000",

"msg": "SUCCESS",

// Clear text before encryption

// "content": {

//   "cAccessToken": "0ad186353c424c64897fcc00445c9ba1",

//   "tokenType": "Bearer",

//   "issueAt": 1557053922938,

//   "expiresIn": 14400000,

//   "cOpenID": "liR14%2BvX%2F5hSum5uf4ERczu0KcDnIJA5BM7FoM1ag9c%3D",

//   "lastModifiedDate": 1560849218006,

//   "scope": "corpidapi_auth corpidapi_formFilling"

// }

  "content":
  "AAADIF/FmVrY9df128k2fWjAFrRr29kTBVzfQTvJX+eN454TGl8X9SQ1DkY+W9ntIwmwXRz8fa
  jpk4XxYoM9702pFjK4JBzu6+0dCHo0TlNruMRD10nMstMthpT7O0hkouEWYmBnS8YJVpQz2ekaxT
  q0jWc0X0rjlB16Ir+mbs9h3qDEonbHSceDthfdKZsqLd67wfc9KwXVgVCMdr0vo3H4tdjYYfV1ut
  TLNdZbh09Z0C0Tnr64/7xZq5lZMnQ+ttEhksBeD9/zmPGVs77CzoWJihYW7S7dfZHAcfSIES2I
  bBeXnVb0Yg/wkN1FFEPcOSf6Dk04Yck/0iuN/iuXchGbrf5hz/mkhKdi2goxNt/vvU+N8Lk6U8Ld
  wZNmaAZLYXd3jL9ZHnUBkuygcHN8OyaxEkLlJOJ4FFPZpCf2O19aPyCxyJDK1gzp9+k+eD8wn3XH
  40ezYYUJrIqkW9+sS6KX3"

}
```

Example Success Response (For iAM Smart Integrated API)

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "SUCCESS",

// Clear text before encryption

// "content": {

//   "cAccessToken": "0ad186353c424c64897fcc00445c9ba1",
```

```
//  "tokenType": "Bearer",

//  "issueAt": 1557053922938,

//  "expiresIn": 14400000,

//  "cOpenID": "liR14%2BvX%2F5hSum5uf4ERczu0KcDnIJA5BM7FoMlag9c%3D",

//  "lastModifiedDate": 1560849218006,

//  "scope": "corpidapi_auth corpidapi_formFilling"

// }

  "content":
  "AAAAADIF/FmVrY9df128k2fwjAFrRr29kTBVZfQTvJX+eN454TGl8X9SQ1DkY+W9ntIwmwXRz8fa
  jpk4XxYoM9702pFjK4JBzu6+0dCHo0TlNruMRD10nMstMthpT7O0hkouEWYmBnS8YJVpQz2ekaxT
  q0jWc0X0rjlB16Ir+mbs9h3qDEonbHScEDthfdKZsqLd67wfc9KwXVgVCMdr0vo3H4tdjYYfV1ut
  TLNdZbh09Z0C0Tnr64/7xZq5lZMnQ+ttEhksBeD9/zmPGVs77CzoWJihYW7S7dfZHAcZfoSIeS2I
  bBeXnVb0Yg/wkN1FFEPcOSf6DkO4Yck/0iuN/iuXchGbrf5hz/mkhKdi2goxNt/vvU+N8Lk6U8Ld
  wZNmaAZLYXd3jL9ZHnUBkuygcHN8OyaxEkLlJOJ4FFPZpCf2O19aPyCxYJDK1gzp9+k+eD8wn3XH
  40ezYYUJrIqkW9+sS6KX3",

  "iAMContent":
  "DARVjJ6HKrm226aUDQ62A+N78SFFyQpOzzNd8vquVl0CW=4oixFFm8UpuYJzeUmQ8ESYJFeqvbv
  /Jg9MiCYR0RFkl1FFCxfvL7i8C/o=w9fx0SqZZ=0zWXSPedJAFWF4kji3Xd6OHAzeczJFgVlkLRKB
  UCHD+FNf9YCF9IKXf=M6xzJJSKqs1VgzL8wmm6MXDGjVHYiQbClZsmjEcFE+fdXKHnOWeV6ZhJ6Fy
  m2EB0j4BTckRnq3G+aiVLF+uh66FnHs9haZ5WrDUncVufc3SaVv=DuyzwaPV=rimuJ6qjCXoYZwe
  Xb4bDCJtdhbbZjOfOMQkTJxMwRADkNBqE9qUz6T4hU3lQiJdkdtrPbfSNwYSEQRnz+j2YAxfxVNj
  S=t7F+t55IK3+PN6Fp5MAwWpd2TEZrlUHPWjat2Z/4Lt=7auwTE6mgkkyC1Ajpg4hUfy1xbXym4Oz
  IHgDuX9QtLxUFAklNIDHJZz8=ExXZLpCDr//=FxLYptXdWl6CbqVHOyBQ"

}
```

Example error response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M30002",

  "msg": "content encryption key not exist or expired"

}
```

- CorpID calls back e-service

Request Parameter

```
// Line breaks are for legibility only.

POST

https://<rp_domain>/<rp_context>/<call_back_endpoint>

// Callback Body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "SUCCESS",

// Clear text before encryption

// "content": {

// "businessID": "2YotnFZFEjrlzCsicMWpAA",

// "state": "unesidkd",

// "hashCode": "tGzv3JOkF0XG5Qx2TlKWIA",

// "timestamp": 1556450176000,

// "signature": "nnoadisaufplanefhykdjf",

// "cert": "sdfGSDGsdfaGDEHfjsgQG.....GSGjljlkwjmh"

// }

// Base64 encoded encrypted CEK

  "secretKey":

  "XlUu98zV6/W4LPdfZZxzOVY62e4PHZ1e3BzENgA/ysO+rxmnBlqmYRUOqYKS8Fj/6EpuCA1jsnN

Y/NJWh8KBz/Kxfe03I1SuD/lHemzMahXLc9x58ecGU0q59VRNO+powUdPTYu0EPH8izh8GQ+Ht33

gic14GMmXuwjmqoo4TZl6h4hPGf6al6piQ9gucsJgFpIoP9genqXqcDCmNkhLoLUMrk5t/ZsA97A

P7DHAGh5RcUT3Qktwz+D1I7Nd4oWUS6yo8ci39epiYSrG9B06OZ/kVo7k6to1Q7KVoCKj3PsMJgC

UVfOQ0HP1YnH599Cu4eyULaa3rIpafpycPvqh9g==",

// iv + ciphertext

  "content":

  "AAAAFaP8/FuTcoymbq2hswpgHER9BUM5itaTKWo2/UwtG4wWRsMgri9bSMZ6uLxjKgmGxKtiR/

5N8JuggXlWJjoMJHp8wnLgSU4FFuGr5yNuGiLpW62fmf95GdF/ikInE5zkzT3hJRzTXbdTALeXWW

Pu/88WyZn1REGqSs6m2AdMY5pIhgia8bCHBxSIPBGuBtAujMII7Nlb6ocFwUL0QuaZqPF/hNJWt/

EI46YGHeGKiRB6KP6hXg6K5I9Sz77uQfO60PV5833fnvU"

}
```

- CorpID calls back e-service (For iAM Smart Integrated API)

Request Parameter

```
// Line breaks are for legibility only.

POST

https://<rp_domain>/<rp_context>/<call_back_endpoint>

// Callback Body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "SUCCESS",

// Clear text before encryption

// "content": {

// "businessID": "2YotnFZFEjrlzCsicMWpAA",

// "state": "unesidkd",

// "hashCode": "tGzv3JOkF0XG5Qx2TlKWIA",

// "timestamp": 1556450176000,

// "signature": "nnoadisaufplanefhykdjf",

// "cert": "sdfGSDGsdfaGDEHfjsgQG.....GSGjljlkwjmh"

// }

// Base64 encoded encrypted CEK

  "secretKey":

  "XlUu98zV6/W4LPDfZZxzOVY62e4PHZ1e3BzENgA/ysO+rxmnBlqmYRUOqYKS8Fj/6EpuCA1jsnN

Y/NJWh8KBz/Kxfe03I1SuD/lHemzMahXLc9x58ecGU0q59VRNO+powUdPTYu0EPH8izh8GQ+Ht33

gic14GMmXuwjmqoo4TZl6h4hPGf6al6piQ9gucsJgFpIoP9genqXqcDCmNkhLoLUMrk5t/ZsA97A

P7DHAGh5RcUT3Qktwz+D1I7Nd4oWUS6yo8ci39epiYSrG9B06OZ/kVo7k6to1Q7KVoCKj3PsMJgC

UVfOQ0HP1YnH599Cu4eyULaa3rIpafpycPvqh9g==",

// iv + ciphertext

  "content":

  "AAAAFaP8/FuTcoymbq2hswpgHER9BUM5itaTKWo2/UwtG4wWRsMgri9bSMZ6uLxjKgmGxKtiR/

5N8JuqgXlWJjoMJHp8wnLgSU4FFuGr5yNuGiLpW62fmf95GdF/ikInE5zkzT3hJRzTXbdTALeXWW

Pu/88WyZn1REGqSs6m2AdMY5pIhgia8bCHBxSIPBGuBtAujMII7Nlb6ocFwUL0QuaZqPF/hNJWt/

EI46YGHeGKiRB6KP6hXg6K5I9Sz77uQfO60PV5833fnvU",

// iAM Smart Base64 encoded encrypted CEK
```

```

    "iAMSecretKey":
    "wL6CEvM1lPf5W1LOep08zbdKYSarJnWz1htVElsyE1JYXbng+Sy19drVQQizWleuB1HhxzOJDR7
    UU7rAmYS3mam5drcQ+GT5kqL7JYoz2PEau4LMWEbZV6B4SDzbuWK0PTkMxFDWFHdquY1Oh7yRl9a
    ey478v6c53M5n3bWr/cYvjVRVZC08g2jUooXITyTSDaInOKPhw2S24fF1lP990JNBwLlBIQfS/Sq
    fBKJvs1NTZUM6ZQVz6NGx4lHF7R2CgGyZfFoSIfBM8ODED4hqyyMqFxe/jfvnTsjsct4cW8mqSzbC
    a51186EhgOpU4SbICYPjHeslQR1XhNj/L5MuQA==",

    // iv + ciphertext

    "iAMContent":
    "yXVCga4upxdlIksAkAjBo7mvSH/f/FoydNSEaC6yCmfvJmBboWkTzcqg+/xE11GjPz8E78encsU
    syVjbT6eQWcUtlDi0nbdLNheKf1pbxcN6jObYHtvbpodq3qsL3qTo0msA2Tp03BR325VZivdFjkU
    z0ixDM1NC9nhbldjt9+k="

}

```

2.3.6. Encryption and Decryption Reference Implementation

- Data Encryption

```

private static final String AES_GCM_NOPADDING = "AES/GCM/NoPadding";

/**
 * AES GCM encryption
 * @param contentByte - Bytes of content to be encrypted
 * @param key - CEK (AES 256)
 * @return
 */
public static String encrypt(byte[] contentByte, byte[] key) {
    SecretKeySpec skeySpec = new SecretKeySpec(key, AES);

    byte[] encrypted = null;

    try {
        SecureRandom = new SecureRandom();

        byte[] iv = new byte[12];

        // do not reuse iv with the same key

        secureRandom.nextBytes(iv);

        Cipher cipher = Cipher.getInstance(AES_GCM_NOPADDING);
    }
}

```

```

        GCMParameterSpec parameterSpec = new GCMParameterSpec(128, iv);

        cipher.init(Cipher.ENCRYPT_MODE, skeySpec, parameterSpec);

        encrypted = cipher.doFinal(contentByte);

        ByteBuffer byteBuffer = ByteBuffer.allocate(4 + iv.length +
encrypted.length);

        byteBuffer.putInt(iv.length);

        byteBuffer.put(iv);

        byteBuffer.put(encrypted);

        byte[] cipherMessage = byteBuffer.array();

        return Base64Util.base64Encode(cipherMessage);

    } catch (NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException | InvalidAlgorithmParameterException |
IllegalBlockSizeException | BadPaddingException e) {

        log.error("Encryption error: {}", e);

        throw new CorpIDException(e);

    }
}

```

- Data Decryption

```

private static final String AES_GCM_NOPADDING = "AES/GCM/NoPadding";

/**
 * AES GCM decryption
 * @param contentByte - Bytes of content to be decrypted
 * @param key - CEK (AES 256)
 * @return
 */
public static String decrypt(byte[] content, byte[] key) {

    SecretKeySpec skeySpec = new SecretKeySpec(key, AES);

    ByteBuffer byteBuffer = ByteBuffer.wrap(content);

    int ivLength = byteBuffer.getInt();

```

```
if(ivLength != 12) {

    throw new IllegalArgumentException("invalid iv length");

}

try {

    byte[] iv = new byte[ivLength];

    byteBuffer.get(iv);

    byte[] cipherText = new byte[byteBuffer.remaining()];

    byteBuffer.get(cipherText);


    Cipher cipher = Cipher.getInstance(AES_GCM_NOPADDING);

    GCMParameterSpec parameterSpec = new GCMParameterSpec(128, iv);

    cipher.init(Cipher.DECRYPT_MODE, sKeySpec, parameterSpec);

    byte[] original = cipher.doFinal(cipherText);

    return new String(original);

} catch (NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException | InvalidAlgorithmParameterException |
IllegalBlockSizeException | BadPaddingException e) {

    throw new CorpIDException(e);

}

}
```

2.3.7. API Encryption Details

2.3.7.1. Request Symmetric Content Encryption Key

RESTful API	https://<CorpID_domain>/api/eservice/v1/security/getKey
Request Type	POST
Service Version	1.0.0
Description	Each e-service must request a data encryption key to encrypt business data for every POST request when calling the CorpID API. This API allows e-service to request symmetric Content Encryption Key for encrypting business data using AES-256 algorithm. Key expiration time will also be specified in the API response.

2.3.7.1.1. Request Parameters

Common API Parameters described 2.4.1

2.3.7.1.2. Example Request

```
// Line breaks are for legibility only.

// How to generate shared common parameters is described in 2.4.1.

POST

https://<CorpID_domain>/api/eservice/v1/security/getKey

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"
```

2.3.7.1.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
secretKey	M	String	Base64-encoded symmetric data encryption key encrypted	

			by the certificate that e-service submitted to CorpID System	
pubKey	M	String	Base64-encoded public key of the certificate that is used to encrypt the symmetric data encryption key.	
issueAt	M	Long	secretKey issue time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
expiresIn	M	Long	Key expiration time, expressed in milliseconds.	

2.3.7.1.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "secretKey": "XXXXX",
    "pubKey": "XXXXX",
    "issueAt": 1556450176000,
    "expiresIn": 1556456176000
  }
}
```

2.3.7.1.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M99992",
```

```
"msg": "Authorization failed. Applicant is not permitted to submit the
request."

}
```

2.3.7.2. Revoke Symmetric Content Encryption Key

RESTful API	https://<CorpID_domain>/api/eservice/v1/security/revok eKey
Request Type	POST
Service Version	1.0.0
Description	This API allows e-service to revoke symmetric Content Encryption Key.

2.3.7.2.1. Request Parameters

Common API Parameters described 2.4.1

2.3.7.2.2. Example Request

```
// Line breaks are for legibility only.

// How to generate shared common parameters is described in 2.4.1.

POST

https://<CorpID_domain>/api/eservice/v1/security/revokeKey

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"
```

2.3.7.2.3. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "M00000",  
  
"msg": ""  
  
}
```

2.3.7.2.4. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2  
  
{  
  
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  
  "code": "M20006",  
  
  "msg": "signature verification failed"  
  
}
```

2.4. COMMON PARAMETERS

2.4.1. Request Parameters

There are two kinds of parameters in each API request, the common parameters and the specific business parameters.

Common Parameters

All of the common parameters should be included in every POST request that e-service sending to the CorpID System. The parameters should be put in the POST request headers (see examples in the following sections for details). For GET requests or callbacks, most of these common parameters are also required. Which of them should be used and where to put them will be described in details in every GET requests or callback.

Field Name	Mandatory/ Optional	Data Type	Description
clientID	M	String	E-service client identifier. The clientID will be assigned toe-service at the initial registration.
signatureMethod	M	String	signature algorithm: HmacSHA256
timestamp	M	Long	The timestamp is the request submit time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT. It is used to prevent replay attack. The value MUST be a positive integer and equal or greater than the timestamp used in previous requests.
nonce	M	String	A nonce is a random string, uniquely generated for each request by e-service. It is used to prevent replay attack. A nonce can be an ASCII string of any length less than or equal to 36 (UUID string length) as long as the uniqueness requirement is met.
signature	M	String	It is a signature of the submitted data. E-service uses <code>clientSecret</code> to sign the concatenated string of <code>clientID</code> , <code>signatureMethod</code> , <code>timestamp</code> , <code>nonce</code> , and <code>encrypted_request body</code>

			to get the signature. The pseudo code to generate the signature is shown at below.
--	--	--	--

Pseudo code for signature generation

```

message = clientID + signatureMethod + timestamp + nonce +
encrypted_request_body;

sha256_HMAC = Mac.getInstance("HmacSHA256");

secret_key = new SecretKeySpec(secret.getBytes(), "HmacSHA256");

sha256_HMAC.init(secret_key);

hash = Base64.encodeBase64String(sha256_HMAC.doFinal(message.getBytes()));

// If the hash needs to be URL encoded

signature = URLEncoder.encode(hash, "UTF-8");

```

Specific Business Parameters

Corpid System returns results based on the business request parameters. Those parameters are normally put in the request body as JSON data and submitted as POST request.

2.4.2. Response and Callback Format

Corpid System returns data in JSON format. The format is:

```

{

  // The txID is a unique identifier of response message.

  // The identifier will be used during troubleshooting

  "txID": "xxxxxxx",

  "code": "xxxxxxx", // return code

  "message": "xxxxxxx", //status message

  "content": {

    // the actual business data

  }

}

```

When the code is M00000, the service execution is successful, and the `content` field returned by the HTTPS request contains the actual business data.

Other than M00000, it indicates that the service execution is abnormal, and the `content` field will be empty or only contains required parameters, e.g. `businessID`.

For callback, if the data passed are in the URL and error occurred before sending the callback, the URL will contain the `cError_code` field to tell e-service the error code from CorpID and `error_code` field to tell e-service the error code from iAM Smart if any. If the data is in the callback body, the above JSON format will be used.

2.4.3. Return Code

Please refer to Appendix D

2.4.4. HTTP Status Code

Please refer to Appendix D

2.5. AUTHORISATION SCOPES

CorpID System makes reference to the OAuth 2.0 authentication framework to enable e-service to gain access to CorpID APIs, with CorpID user's authorisation using “iAM Smart” Mobile App and CorpID mini-program.

Determine Authorisation Scope

To access CorpID APIs, e-service should first determine the authorisation scope(s) required for the CorpID API(s) invoked for its business needs. e-service may refer to the table below for the authorisation mapping:

API	Scope Value
Authentication	corpidapi_auth
Token Exchange to switching corporation	corpidapi_auth
Form Pre-filling with Service Login (aka Profiles)	corpidapi_profiles
Form Pre-filling without Service Login (Anonymous Form Pre-Filling)	corpidapi_formFilling
Digital Signing with Service Login	corpidapi_sign
Digital Signing without Service Login (Anonymous Digital Signing)	corpidapi_sign
Document Wallet	corpidapi_docwallet

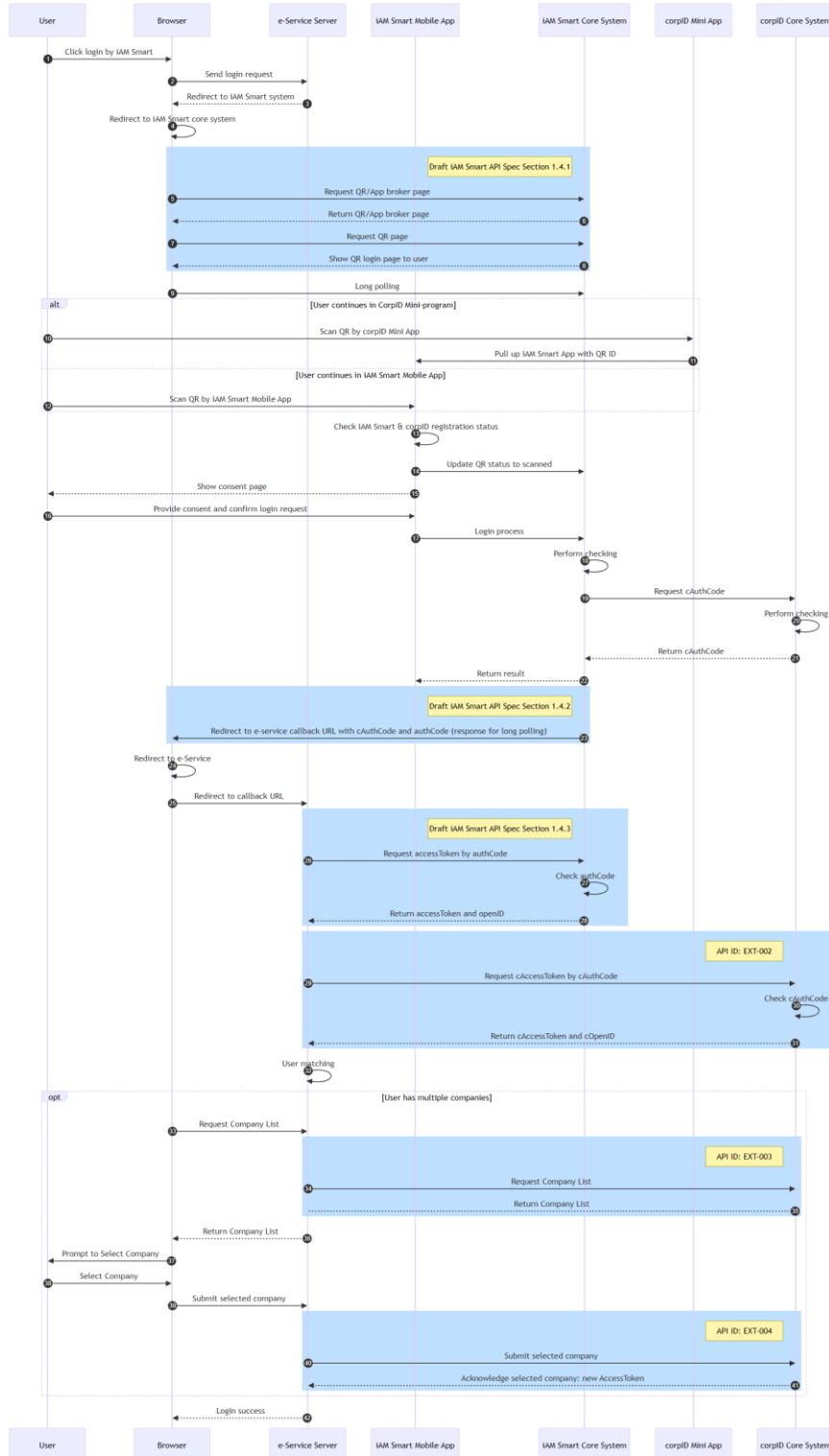
e-Service should combine all authorisation scope(s) required into a single request to get authorisation from CorpID user through CorpID System when CorpID user requests login to e-service using “iAM Smart” Mobile App and CorpID mini program. For example, authorisation scope “corpidapi_auth”, “corpidapi_formFilling” and “corpidapi_sign” are for CorpID authentication form pre-filling and digital signing respectively.

3. USE CASES AND SCENARIOS

3.1. AUTHENTICATION

3.1.1. Authentication with iAM Smart (e-service Website in Different Device)

CorpID Authentication (e-Service Website in Different Device)



API interactions between e-service servers and CorpID System are listed below. Technical

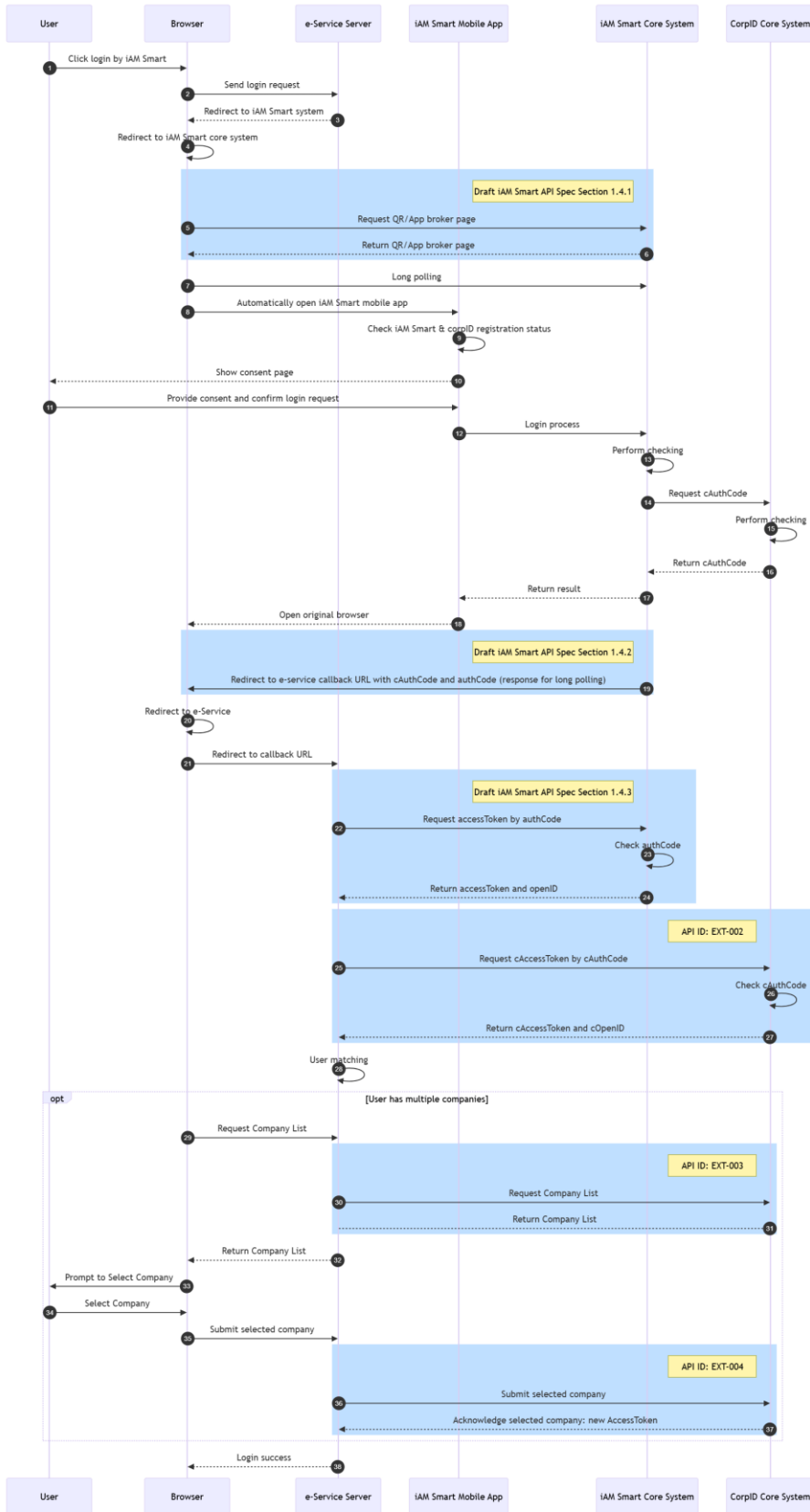
details of respective APIs can be found in the following sections.

No.	API Name	API Reference (Section)
1	Request QR/App Broker Page	Draft iAM Smart Document Section 1.4.1
2	Redirect to e-service callback URL	Draft iAM Smart Document Section 1.4.2
3	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
4	Get CorpID cAccessToken by cAuthCode	4.2
5	[Optional] Request Corporation List	4.3
6	[Optional] Request Token Exchange for Switching Corporation	4.4

3.1.2. Authentication with iAM Smart (e-service Website in Same Device)

3.1.2.1. Initiated from E-service Website

CorpID Authentication (e-Service Website in Same Device)



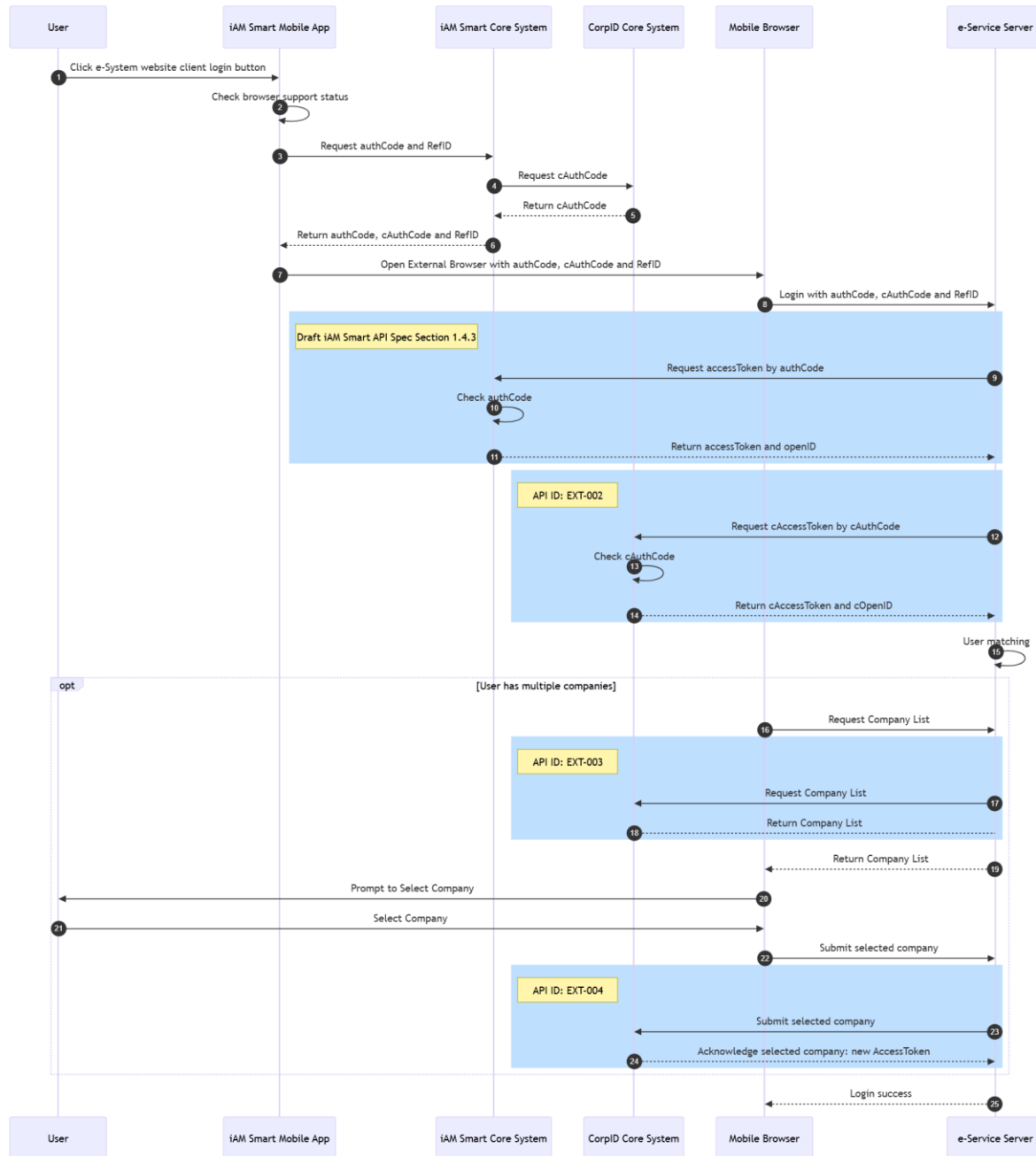
API interactions between e-service servers and CorpID System are listed below. Technical

details of respective APIs can be found in the following sections.

No.	API Name	API Reference (Section)
1	Request QR/App Broker Page	Draft iAM Smart Document Section 1.4.1
2	Redirect to e-service callback URL	Draft iAM Smart Document Section 1.4.2
3	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
4	Get CorpID cAccessToken by cAuthCode	4.2
5	[Optional] Request Corporation List	4.3
6	[Optional] Request Token Exchange for Switching Corporation	4.4

3.1.2.2. Initiated from “iAM Smart” mobile App (Direct Login v2)

CorpID Authentication (e-Service Website in Same Device)



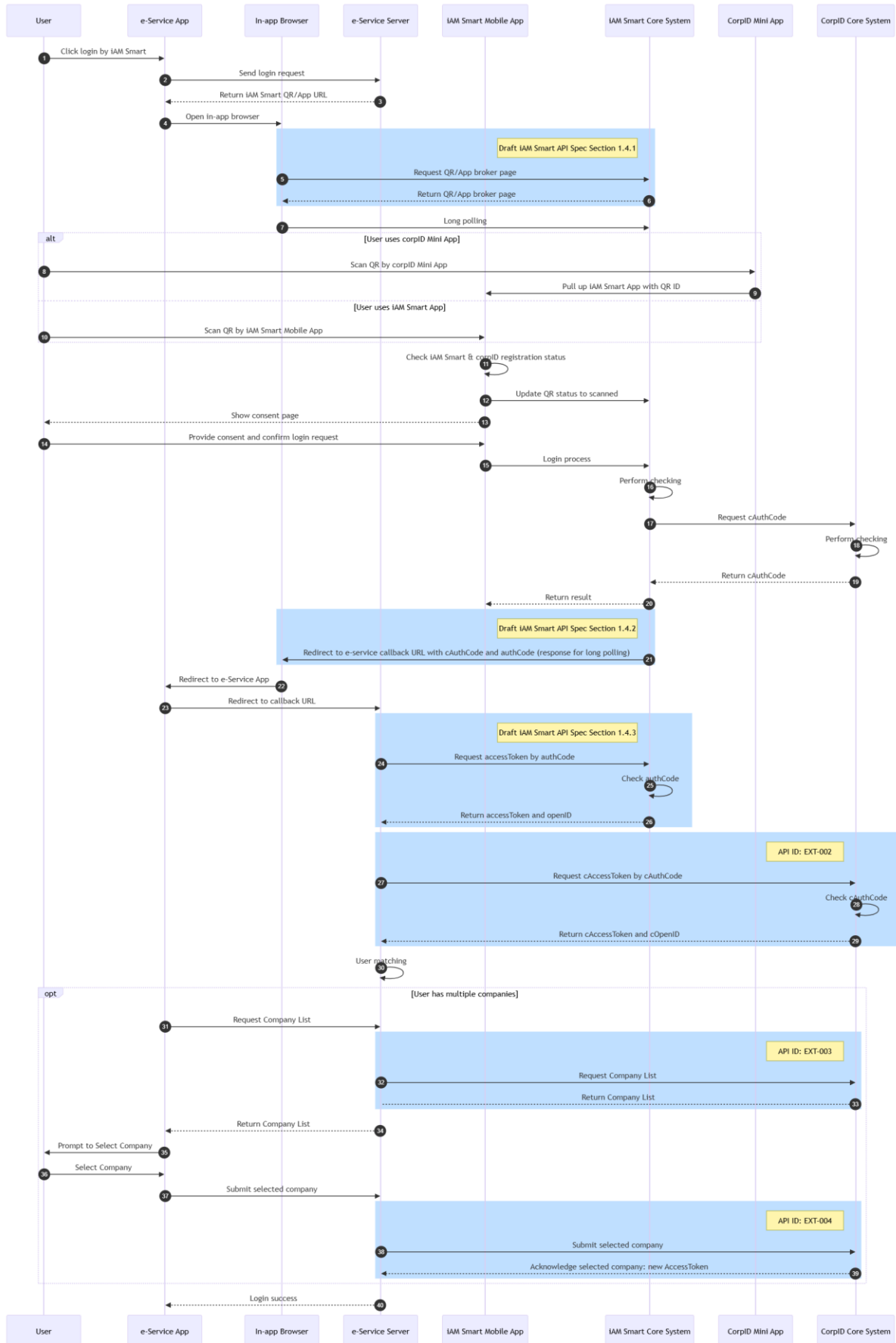
API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found in the following sections.

No.	API Name	API Reference (Section)
1	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 4.3
2	Get CorpID cAccessToken by cAuthCode	4.2
3	[Optional] Request Corporation List	4.3

4	[Optional] Request Token Exchange for Switching Corporation	4.4
---	---	-----

3.1.3. Authentication with iAM Smart (e-service App in Different Device)

Corpid Authentication (e-Service App in Different Device)



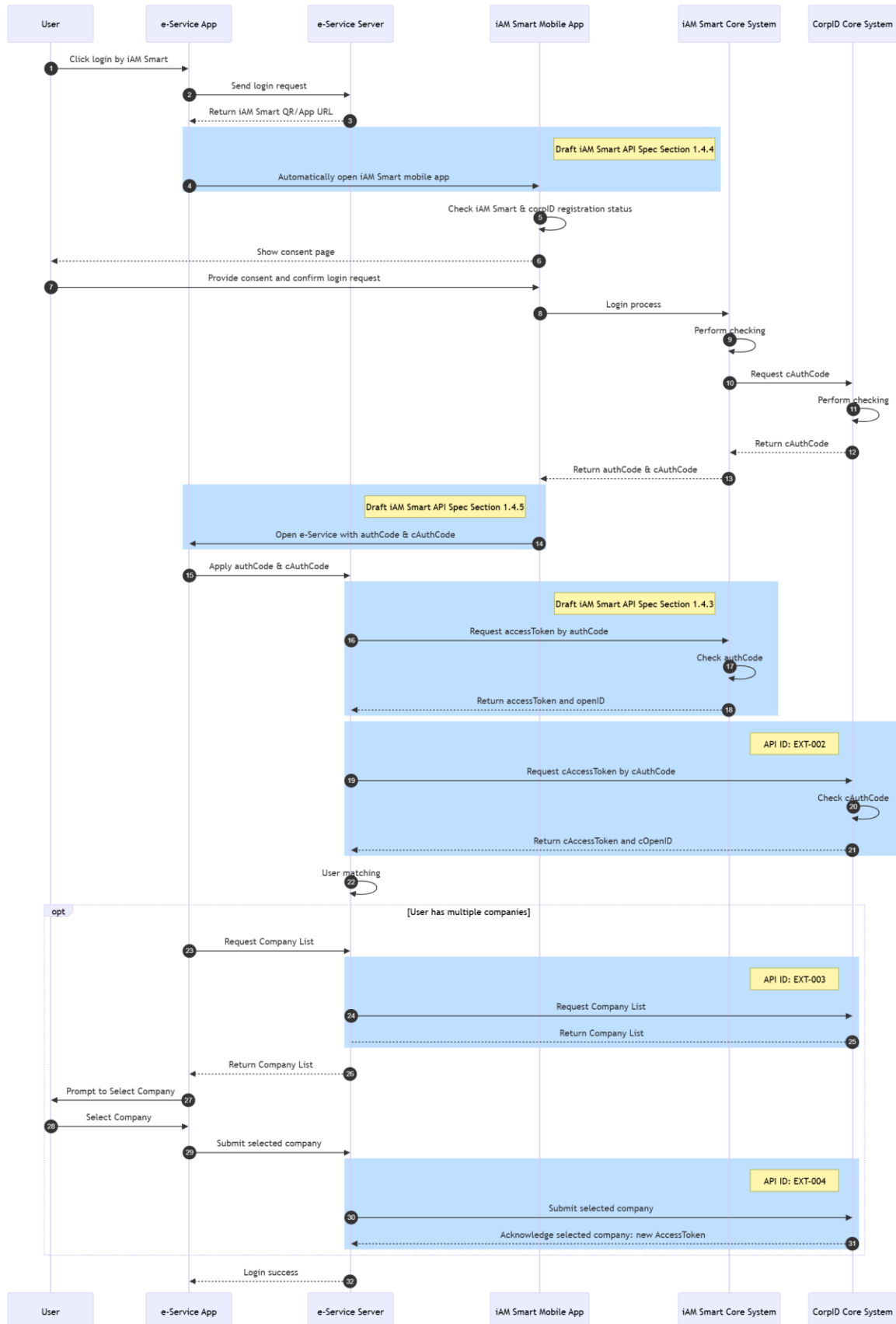
API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found in the following sections.

No.	API Name	API Reference (Section)
1	Request QR/App Broker Page	Draft iAM Smart Document Section 1.4.1
2	Redirect to e-service callback URL	Draft iAM Smart Document Section 1.4.2
3	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
4	Get CorpID cAccessToken by cAuthCode	4.2
5	[Optional] Request Corporation List	4.3
6	[Optional] Request Token Exchange for Switching Corporation	4.4

3.1.4. Authentication with iAM Smart (e-service App in Same Device)

3.1.4.1. Initiated from E-service App

CorPID Authentication (e-Service App in Same Device)

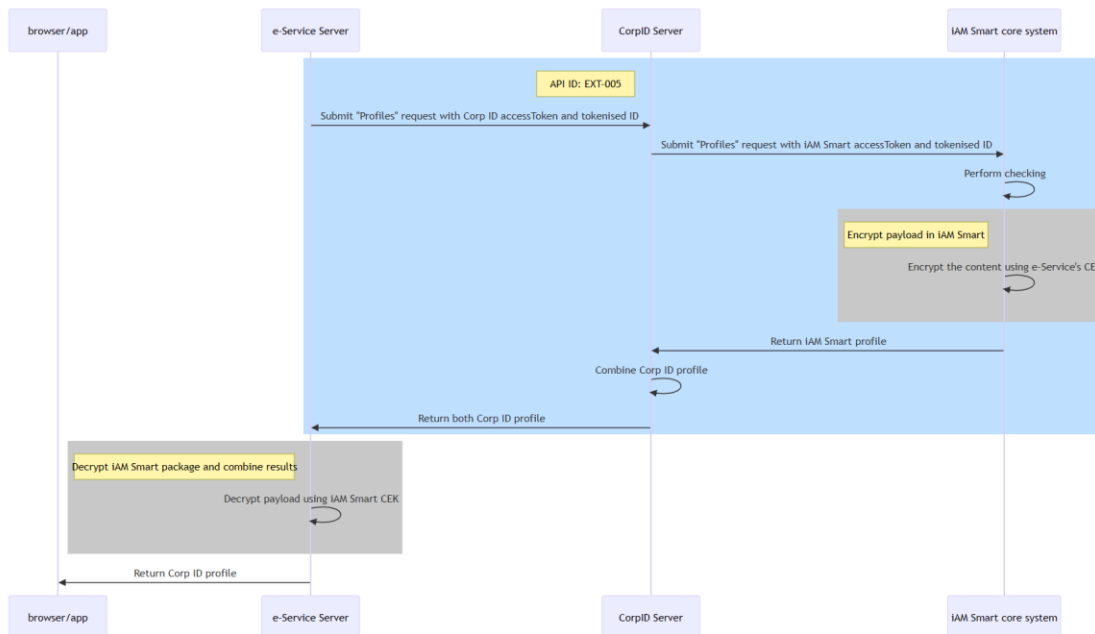


API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found in the following sections.

No.	API Name	API Reference (Section)
1	Open iAM Smart Mobile App for Authentication (appv2)	Draft iAM Smart Document Section 1.4.4
2	Call with cAuthCode to Online Service App	Draft iAM Smart Document Section 1.4.5
3	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
4	Get CorpID cAccessToken by cAuthCode	4.2
5	[Optional] Request Corporation List	4.3
6	[Optional] Request Token Exchange for Switching Corporation	4.4

3.2. OBTAIN PROFILES INFORMATION (FOR B/D AND GOVERNMENT BUREAU ONLY)

CorpID Profile



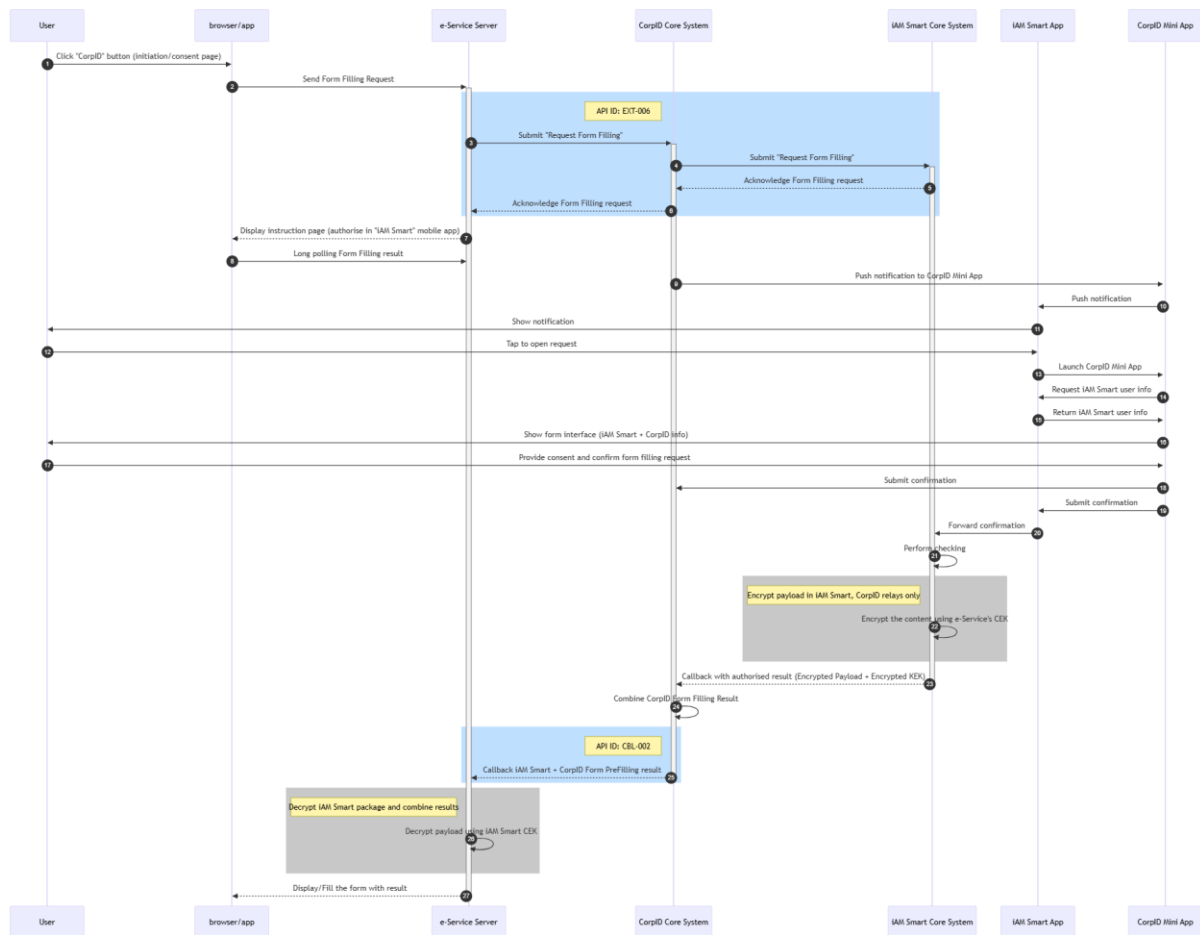
API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Obtain Profiles Information	4.5

3.3. FORM PRE-FILLING WITH SERVICE LOGIN

3.3.1. Form Pre-filling (e-service Website/App in Different Device)

CorpID Form Pre-filling (e-Service Website/App in Different Device)

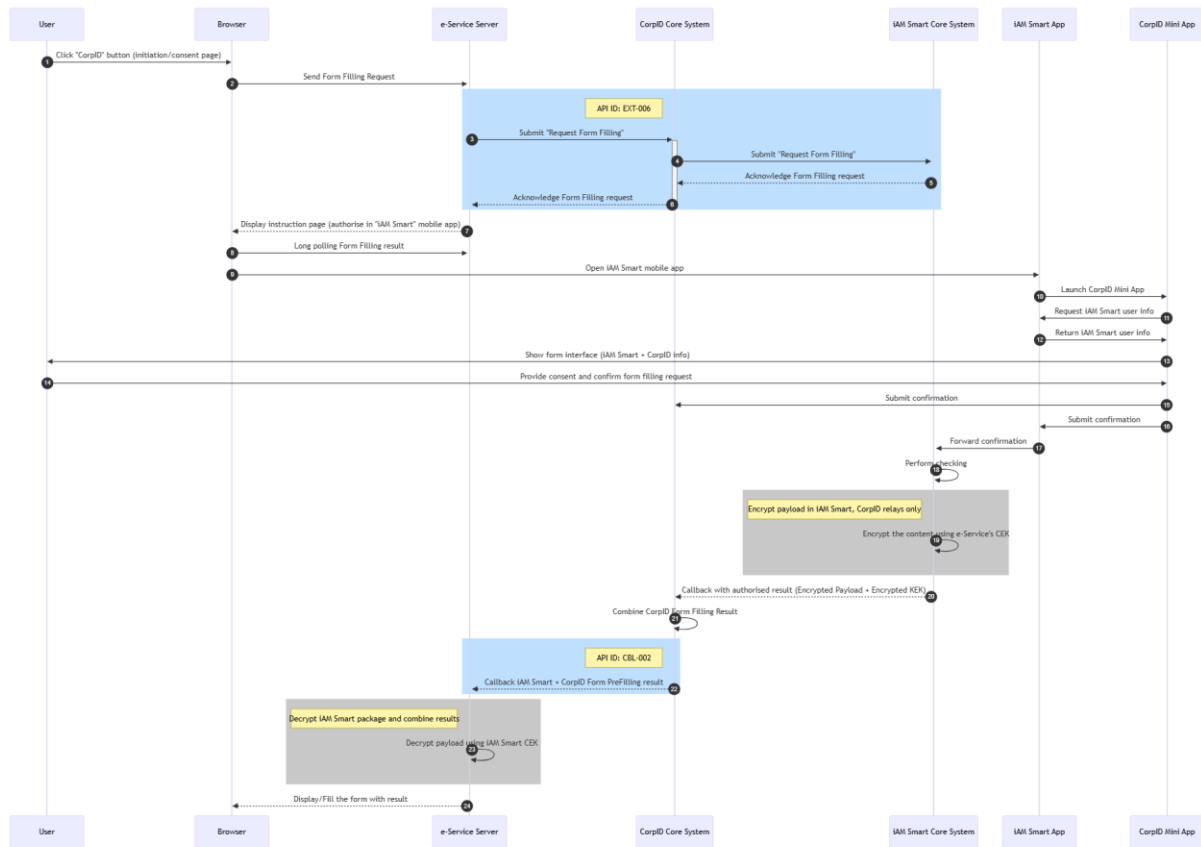


API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Form Pre-Filling	4.6
2	Form Pre-filling Callback	4.7

3.3.2. Form Pre-filling (e-service Website in Same Device)

CorPID Form Pre-filling (e-Service Website in Same Device)

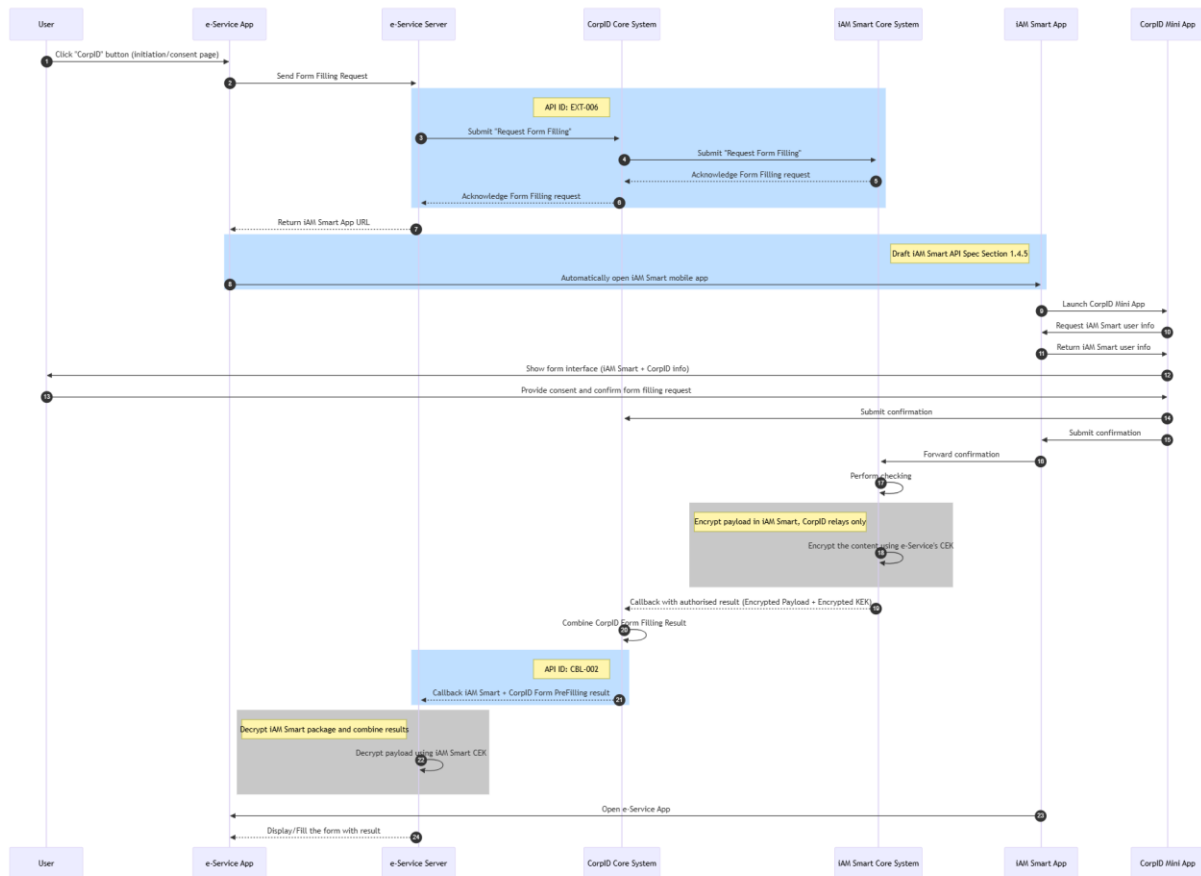


API interactions between e-service servers and CorPID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Form Pre-Filling	4.6
2	Form Pre-filling Callback	4.7

3.3.3. Form Pre-filling (e-service App in Same Device)

CorPID Form Pre-filling (e-Service App in Same Device)



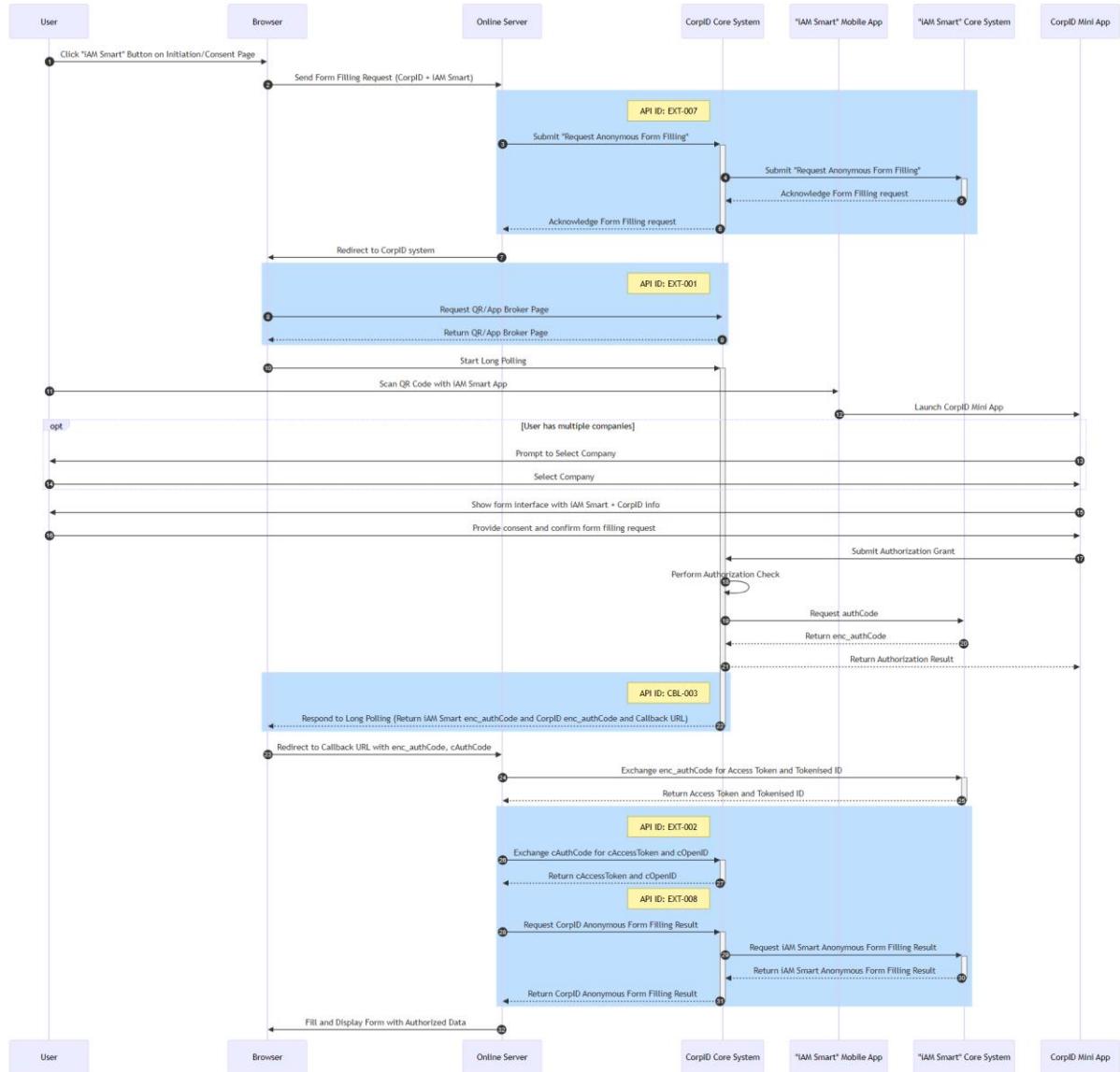
API interactions between e-service servers and CorPID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Form Pre-Filling	4.6
	Request Form Pre-Filling (for App e-Service)	4.24
2	Form Pre-filling Callback	4.7

3.4. FORM PRE-FILLING WITHOUT SERVICE LOGIN

3.4.1. Anonymous Form Pre-filling (e-service Website in Different Device)

CorpID Anonymous Form Filling (e-Service Website in Different Device)

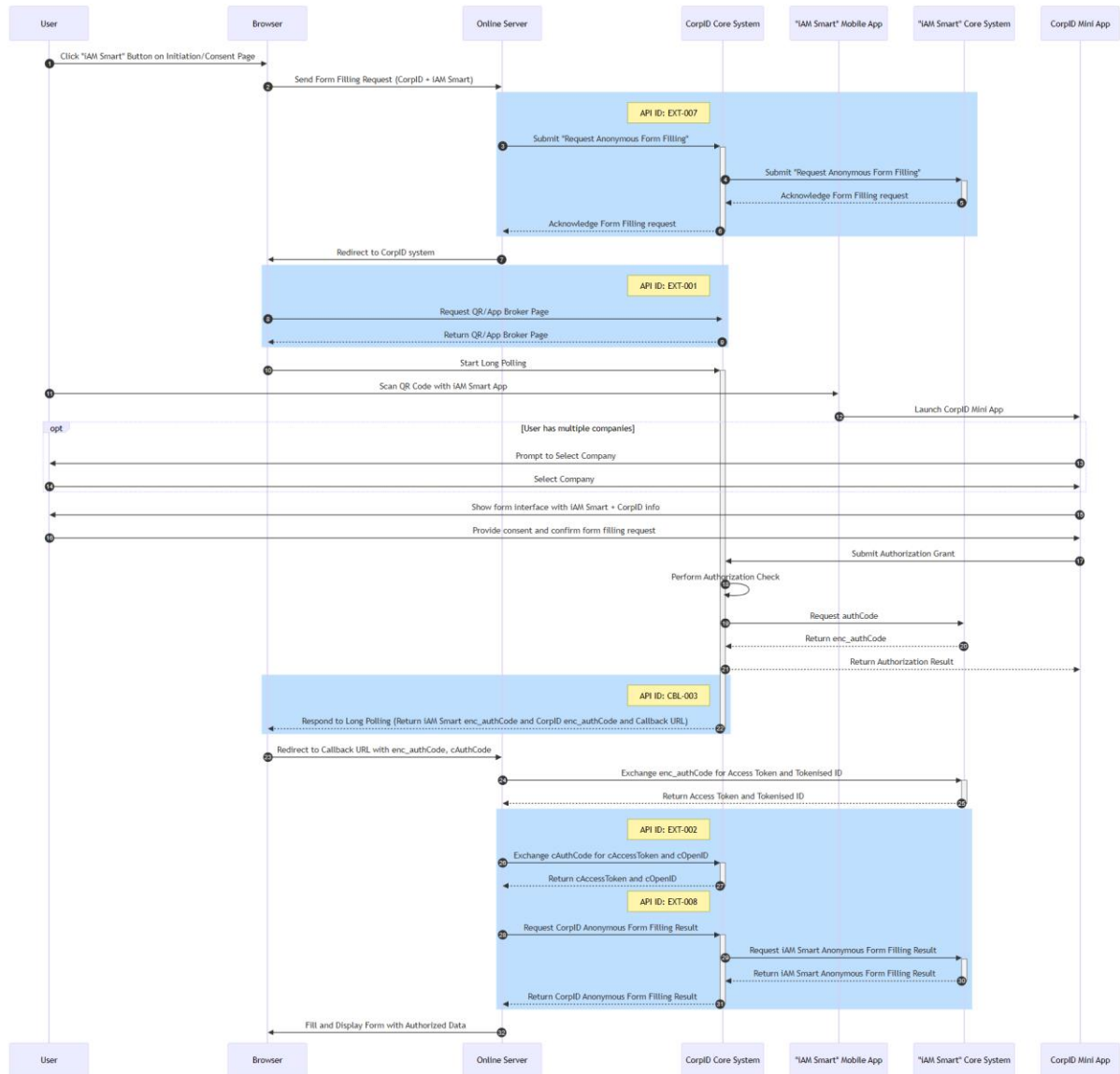


API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Form Pre-Filling	4.8
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get Access Token	4.2
5	Get Anonymous Form Pre-filling Result	4.10

3.4.2. Anonymous Form Pre-filling (e-service Website in Same Device)

CorPID Anonymous Form Filling (e-Service Website in Same Device)

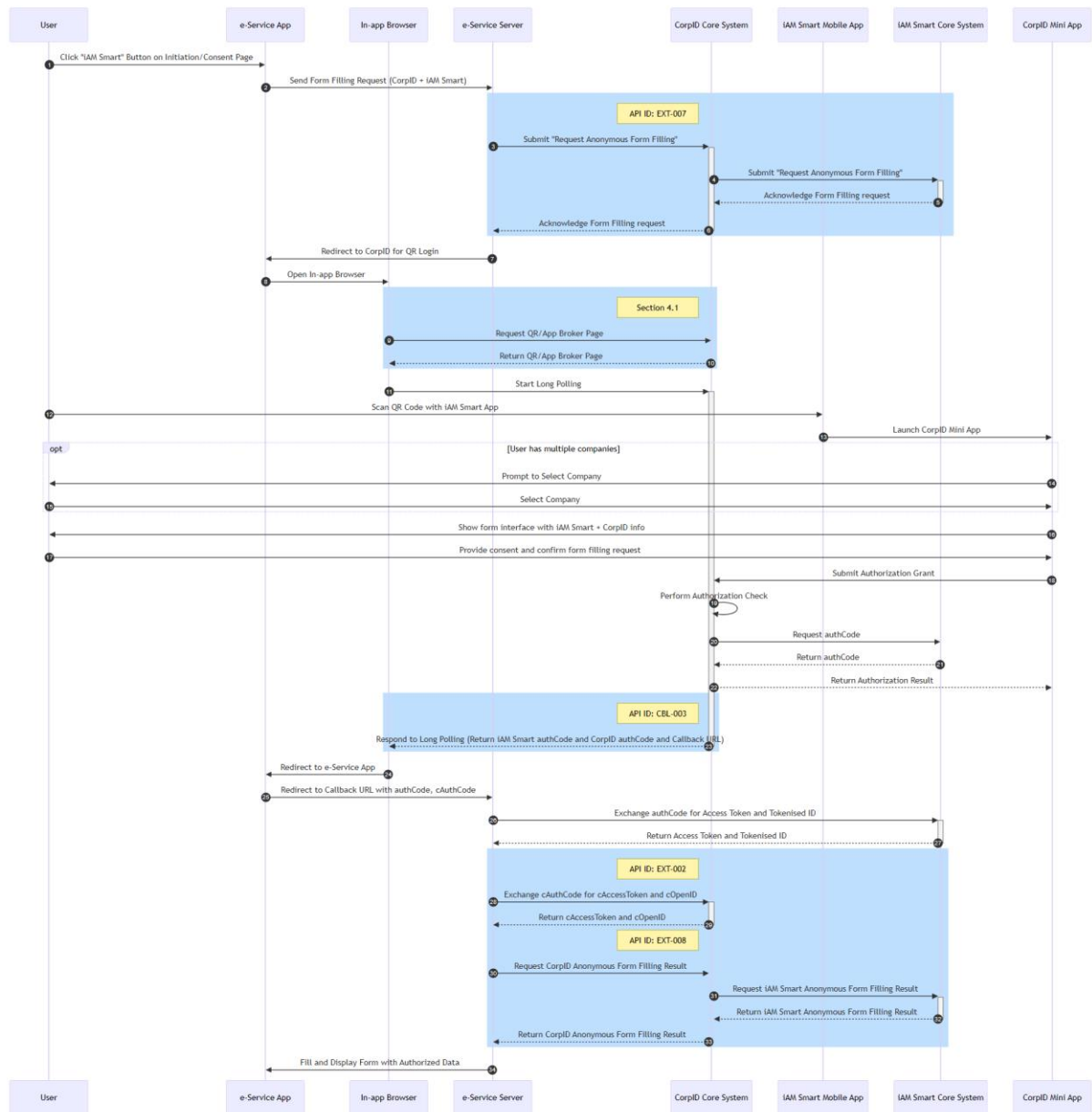


API interactions between e-service servers and CorPID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Form Pre-Filling	4.8
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get Access Token	4.2
5	Get Anonymous Form Pre-filling Result	4.10

3.4.3. Anonymous Form Pre-filling (e-service App in Different Device)

Corpid Anonymous Form Filling (e-Service App in Different Device)

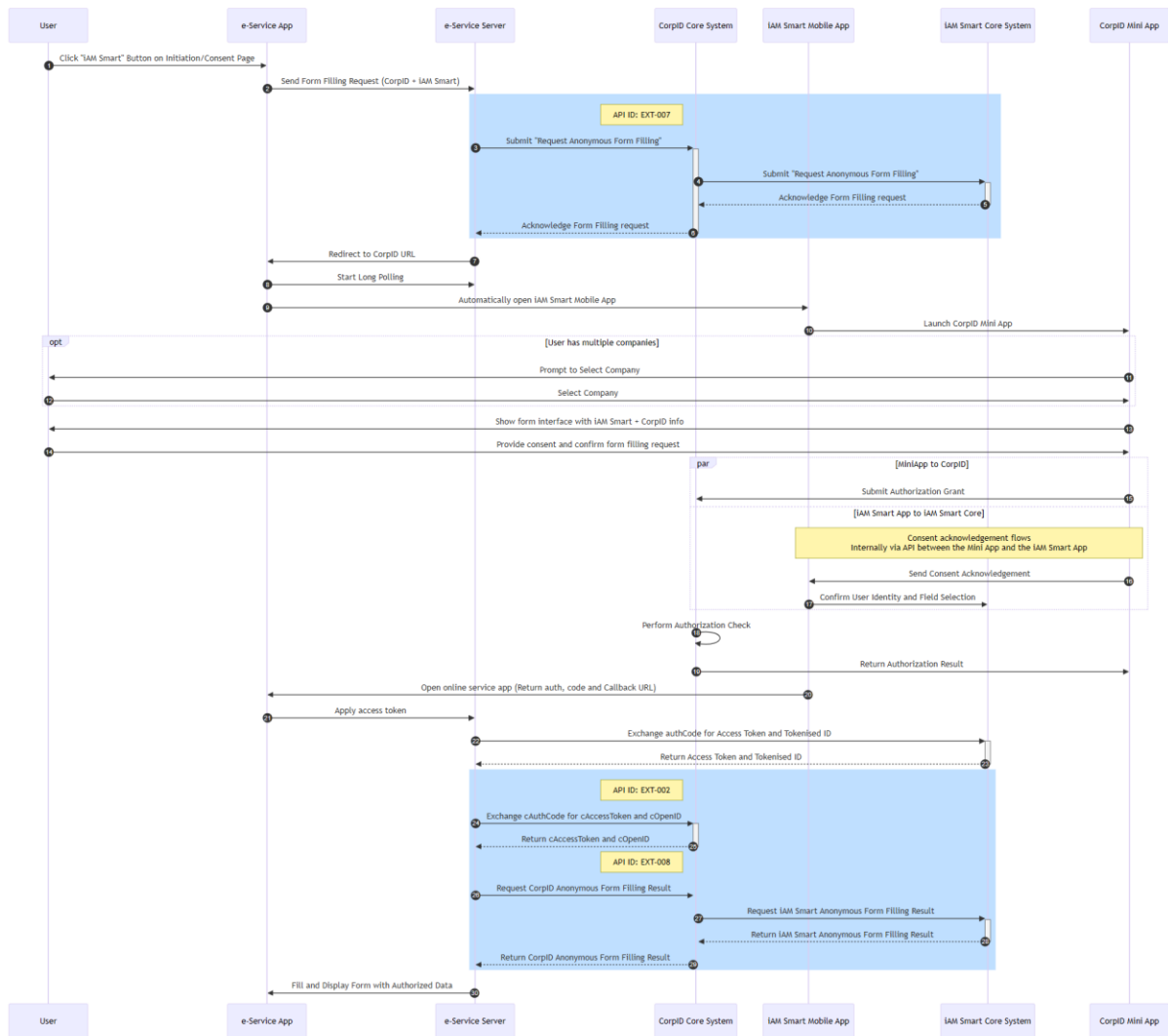


API interactions between e-service servers and Corpid System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Form Pre-Filling	4.8
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get Access Token	4.2
5	Get Anonymous Form Pre-filling Result	4.10

3.4.4. Anonymous Form Pre-filling (e-service App in Same Device)

CorPID Anonymous Form Filling (e-Service App in Different Device)



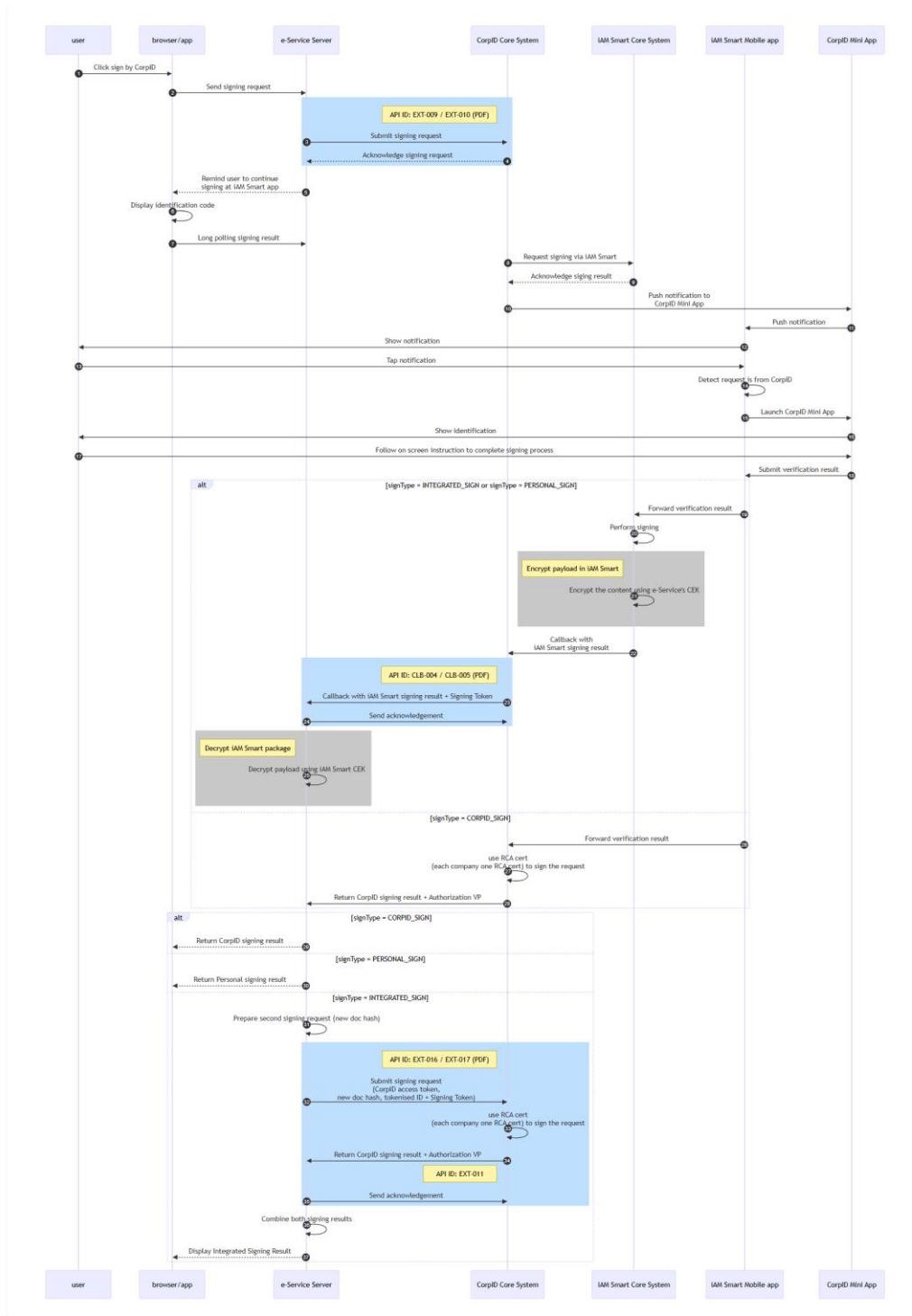
API interactions between e-service servers and CorPID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Form Pre-Filling	4.8
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get Access Token	4.2
5	Get Anonymous Form Pre-filling Result	4.10

3.5. DIGITAL SIGNING WITH SERVICE LOGIN

3.5.1. Digital Signing (e-service Website/App in Different Device)

CorpID Digital Signing (e-Service Website/App in Different Device)



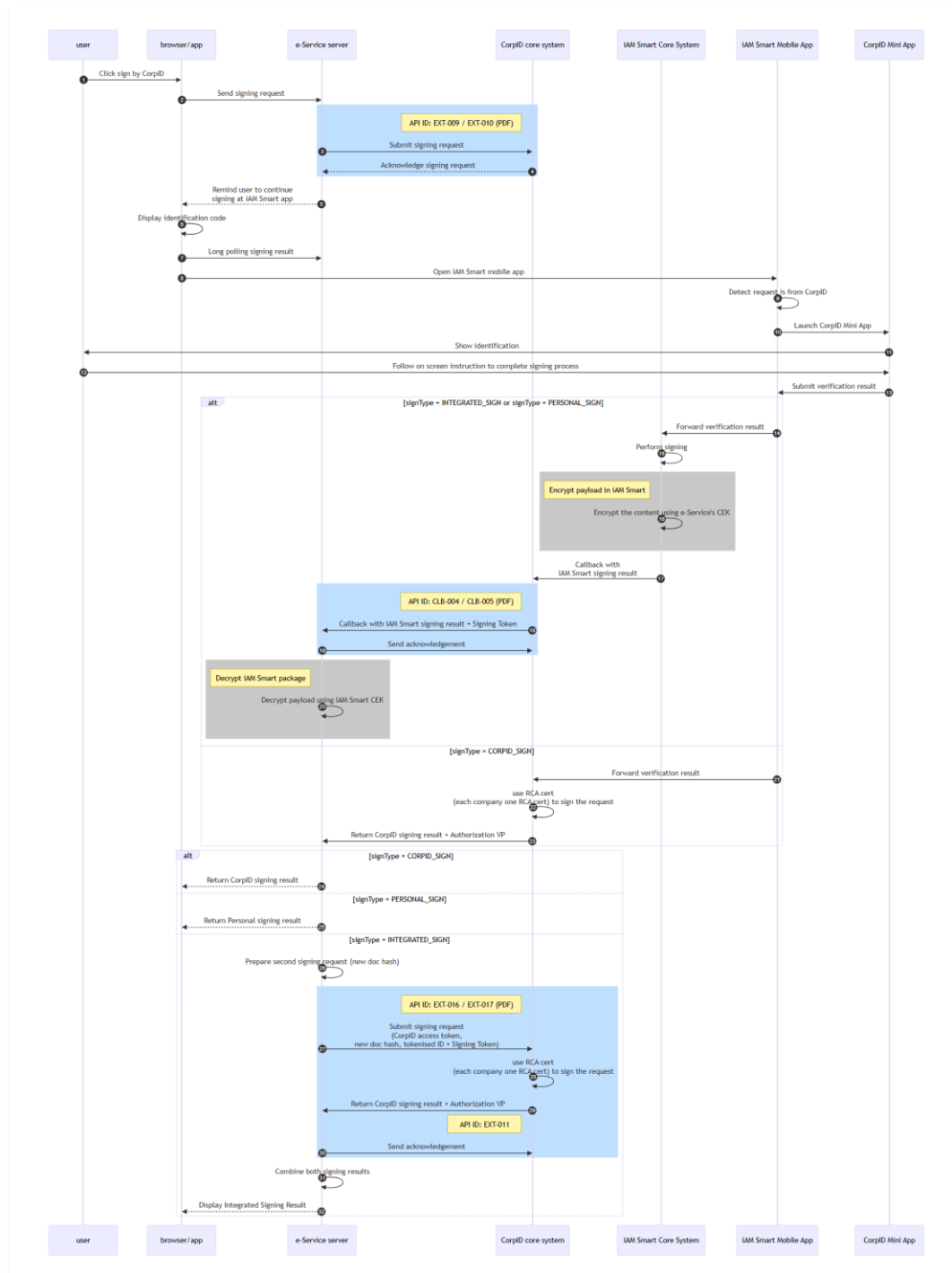
API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Digital Signing	4.11

	Request PDF Digital Signing	4.13
2	Digital Signing Callback	4.12
	PDF Digital Signing Callback	4.14
3	Request Corporation Digital Signing	4.20
	Request Corporation PDF Digital Signing	4.21
4	Acknowledges Digital Signing Result	4.15

3.5.2. Digital Signing (e-service Website/App in Same Device)

CorpiD Digital Signing (e-Service Website/App in Same Device)



API interactions between e-service servers and CorpiD System are listed below. Technical details of respective APIs can be found at the following sections.

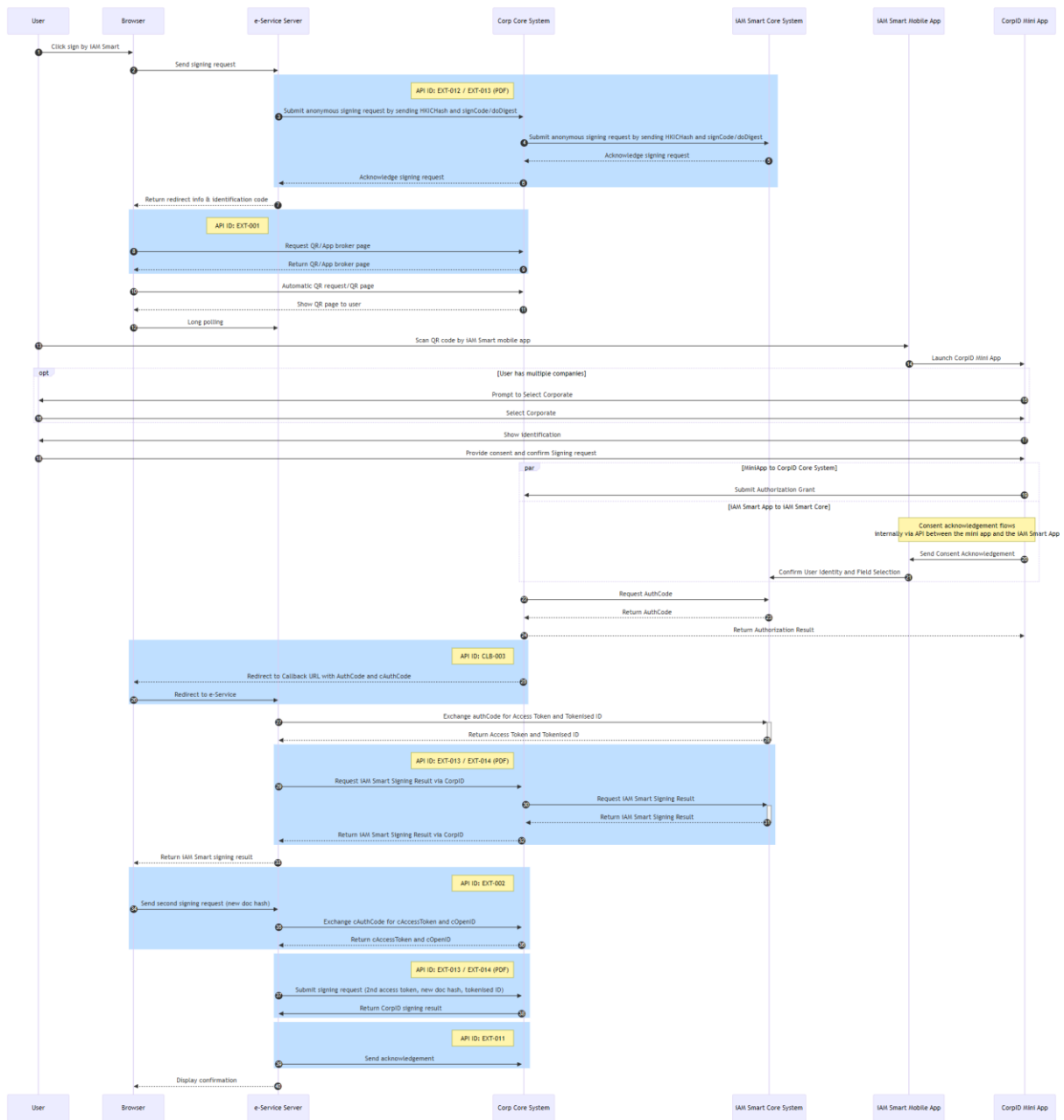
No.	API Name	API Reference (Section)
1	Request Digital Signing	4.11
	Request PDF Digital Signing	4.13
	Request Digital Signing (for App e-Service)	4.25

	Request PDF Digital Signing (for App e-Service)	4.26
2	Digital Signing Callback	4.12
	PDF Digital Signing Callback	4.14
3	Request Corporation Digital Signing	4.20
	Request Corporation PDF Digital Signing	4.21
	Request Corporation Digital Signing (for App e-Service)	4.20
	Request Corporation PDF Digital Signing (for App e-Service)	4.21
4	Acknowledges Digital Signing Result	4.15

3.6. DIGITAL SIGNING WITHOUT SERVICE LOGIN

3.6.1. Anonymous Digital Signing (e-service Website in Different Device)

CorpID Anonymous Digital Signing (e-Service Website in Different Device)



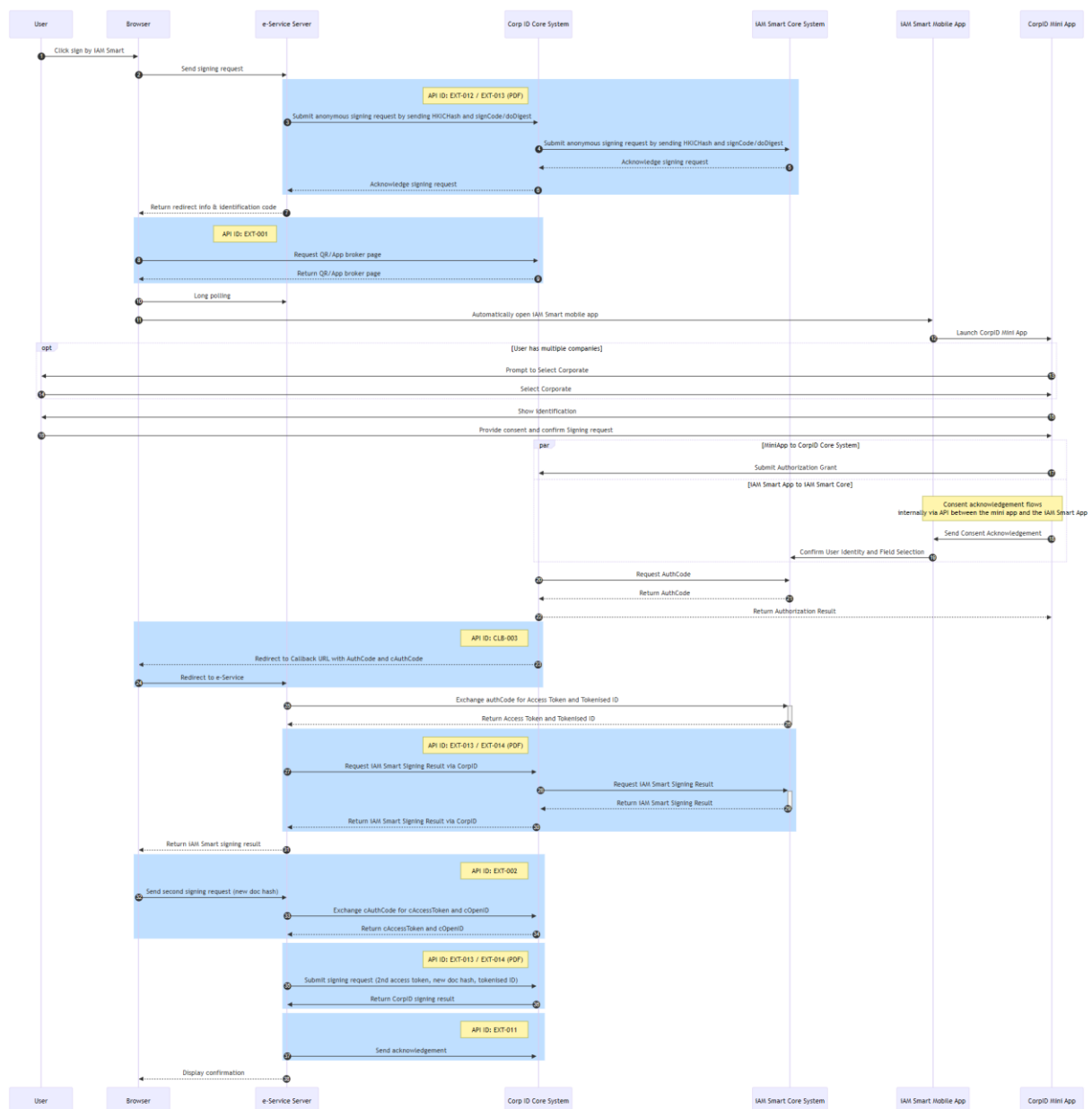
API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Digital Signing	4.16
	Request Anonymous PDF Digital Signing	4.17
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9

4	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
5	Get Anonymous Digital Signing Result (Personal Signing)	4.18
	Get Anonymous Digital Signing Result (PDF) (Personal Signing)	4.19
6	Get CorpID cAccessToken by cAuthCode	4.2
7	Get Anonymous Digital Signing Result (Company Chop)	4.20
	Get Anonymous Digital Signing Result (PDF) (Company Chop)	4.21
8	Acknowledges Digital Signing Result	4.15

3.6.2. Anonymous Digital Signing (e-service Website in Same Device)

CorpID Anonymous Digital Signing (e-Service Website in Same Device)

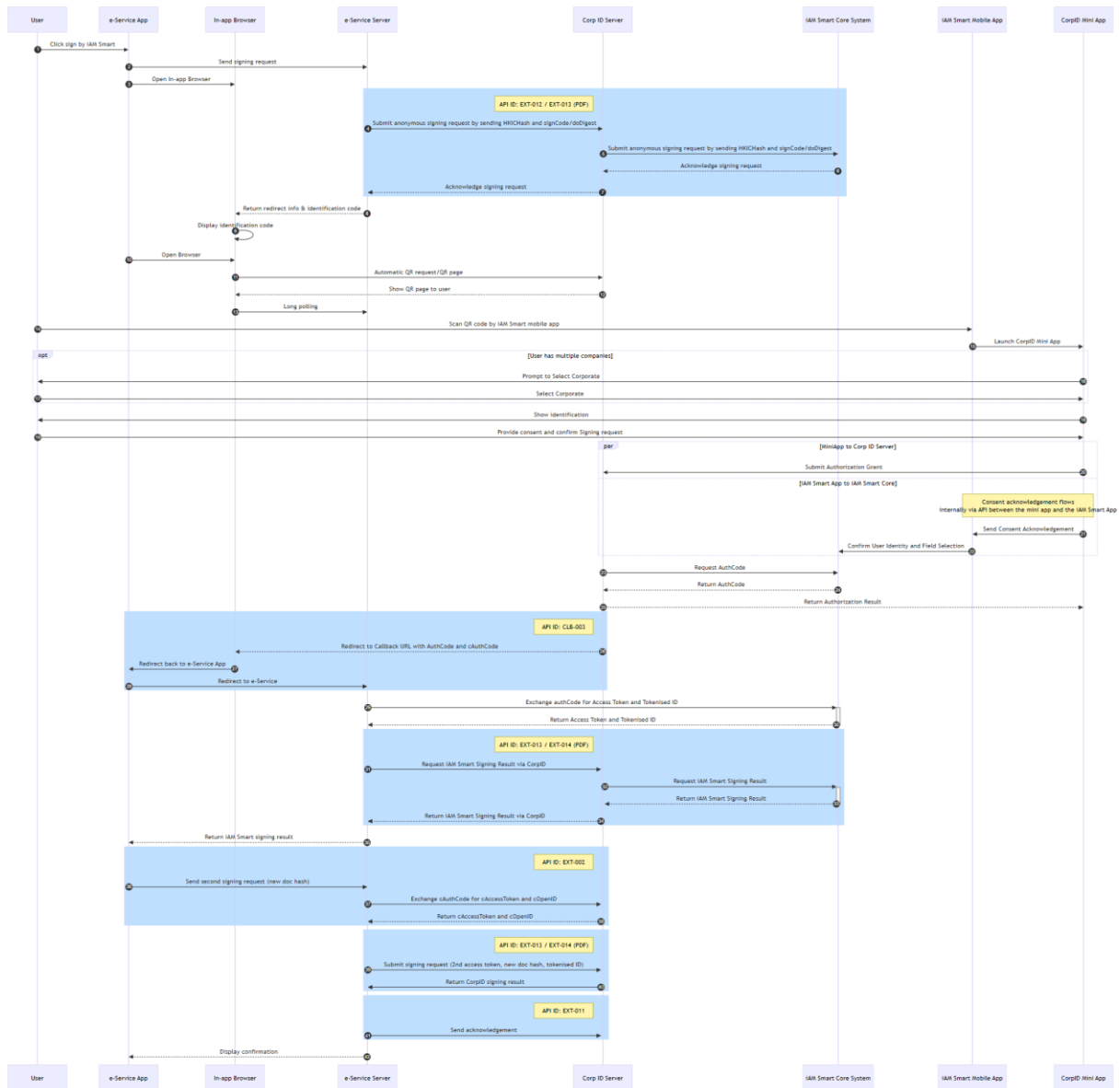


API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Digital Signing	4.16
	Request Anonymous PDF Digital Signing	4.17
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
5	Get Anonymous Digital Signing Result (Personal Signing)	4.18
	Get Anonymous Digital Signing Result (PDF) (Personal Signing)	4.19
6	Get CorpID cAccessToken by cAuthCode	4.2
7	Get Anonymous Digital Signing Result (Company Chop)	4.20
	Get Anonymous Digital Signing Result (PDF) (Company Chop)	4.21
8	Acknowledges Digital Signing Result	4.15

3.6.3. Anonymous Digital Signing (e-service App in Different Device)

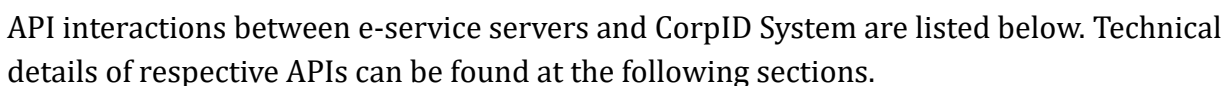
CorpID Anonymous Digital Signing (e-Service App in Different Device)



API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Anonymous Digital Signing	4.16
	Request Anonymous PDF Digital Signing	4.17
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
5	Get Anonymous Digital Signing Result (Personal Signing)	4.18

CorpID Anonymous Digital Signing (e-Service App in Same Device)



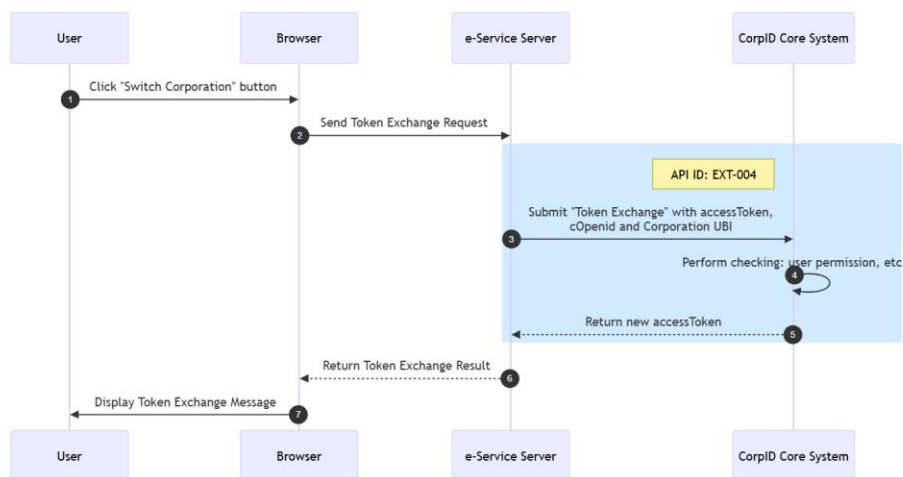
Version 1.0
Page 60

1	Request Anonymous Digital Signing	4.16
	Request Anonymous PDF Digital Signing	4.17
2	Authentication by requesting QR Page/ Broker Page	4.1
3	Callback with authCode to e-service Server	4.9
4	Get iAM Smart Access Token by authCode	Draft iAM Smart Document Section 1.4.3
5	Get Anonymous Digital Signing Result (Personal Signing)	4.18
	Get Anonymous Digital Signing Result (PDF) (Personal Signing)	4.19
6	Get CorpID cAccessToken by cAuthCode	4.2
7	Get Anonymous Digital Signing Result (Company Chop)	4.20
	Get Anonymous Digital Signing Result (PDF) (Company Chop)	4.21
8	Acknowledges Digital Signing Result	4.15

3.7. REQUEST TOKEN EXCHANGE

The sequence diagram below shows how an e-service website leverages the “CorpID” System to perform user token exchange with UBI for switching corporation.

CorpID Token Exchange



API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found in the following sections.

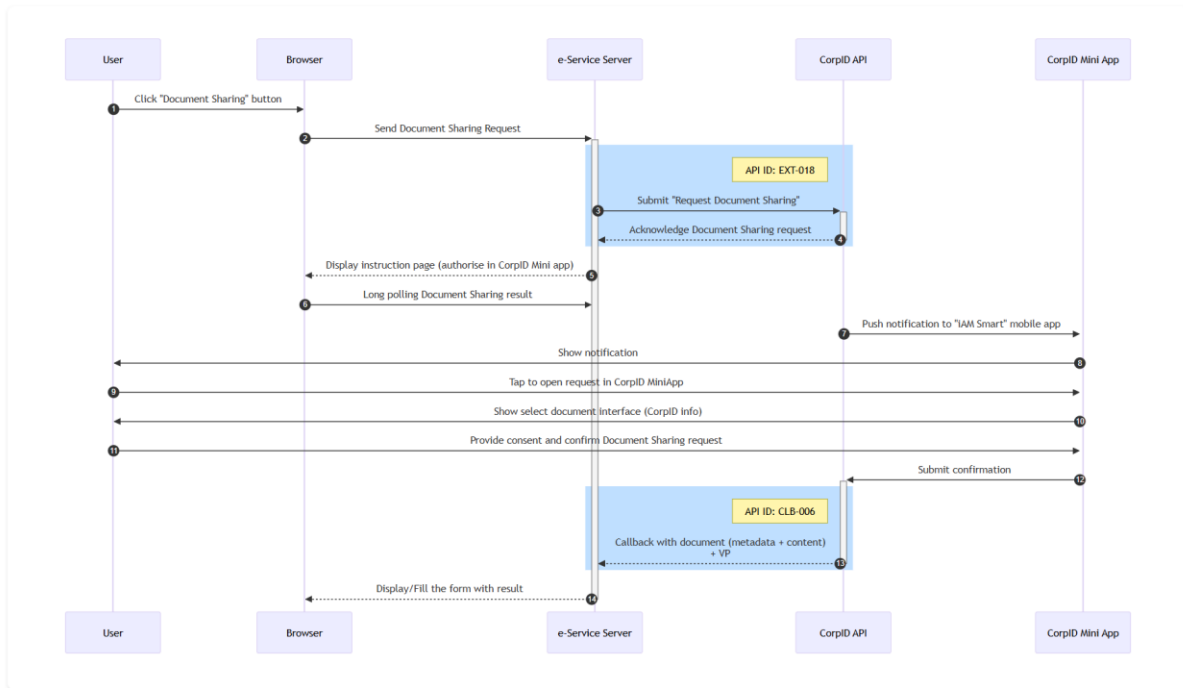
No.	API Name	API Reference (Section)
-----	----------	-------------------------

1	Request Token Exchange for Switching Corporation with UBI	4.4
---	---	-----

3.8. DOCUMENT SHARING WITH SERVICE LOGIN

3.8.1. Document Sharing (e-service Website in Different Device)

CorpiD Document Wallet (e-Service Website in Different Device)

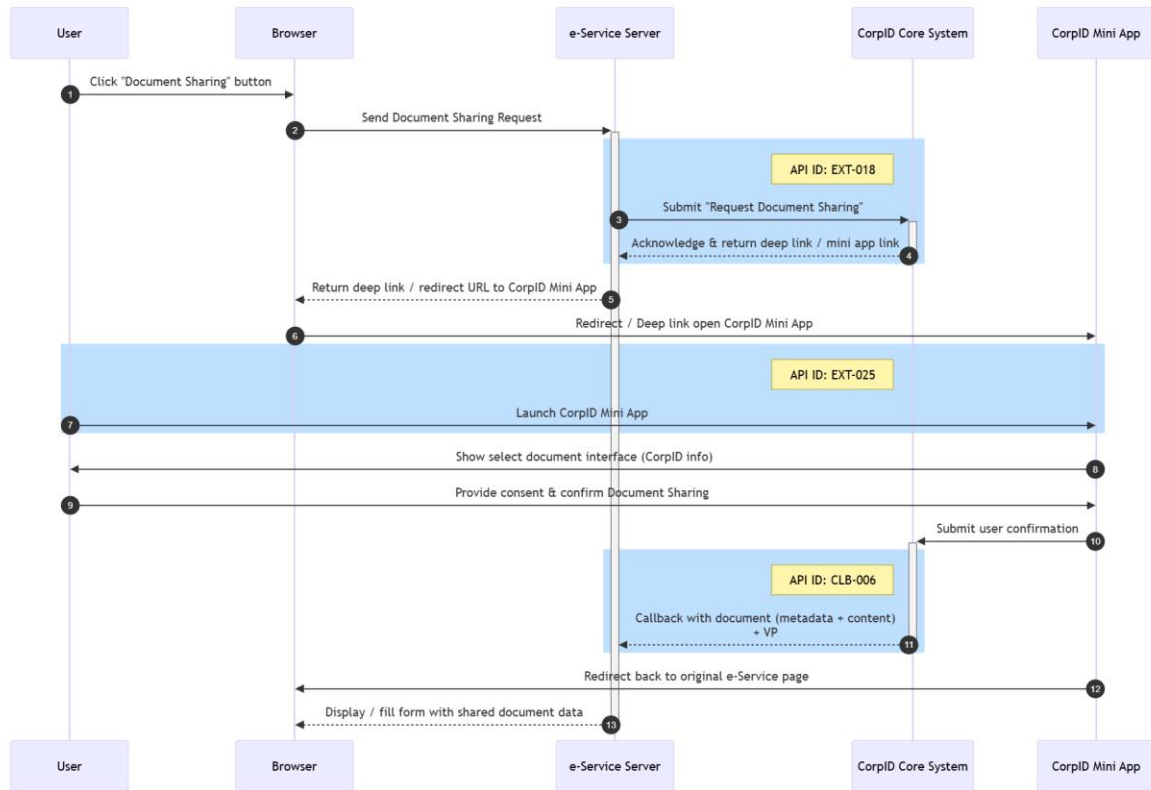


API interactions between e-service servers and CorpiD System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Document Sharing	4.22
2	Callback to Receive Document Sharing	4.23

3.8.2. Document Sharing (e-service Website in Same Device)

CorpID Document Wallet (e-Service Website in Same Device)

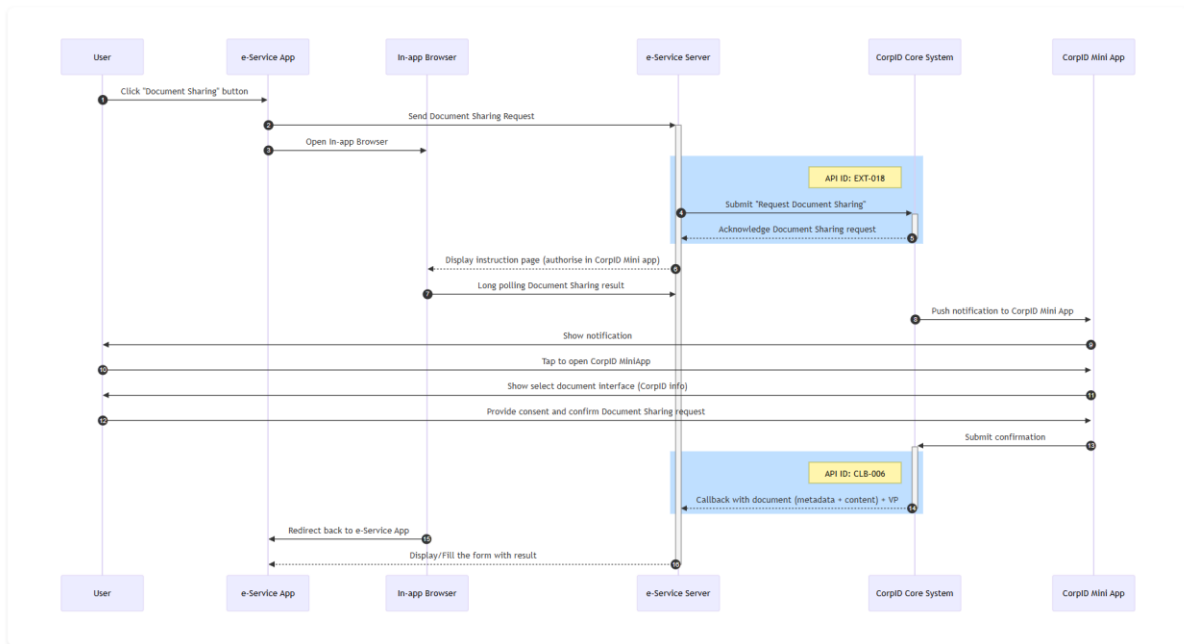


API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Document Sharing	4.22
2	Open CorpID Mini App for Document Sharing	4.28
3	Callback to Receive Document Sharing	4.23

3.8.3. Document Sharing (e-service App in Different Device)

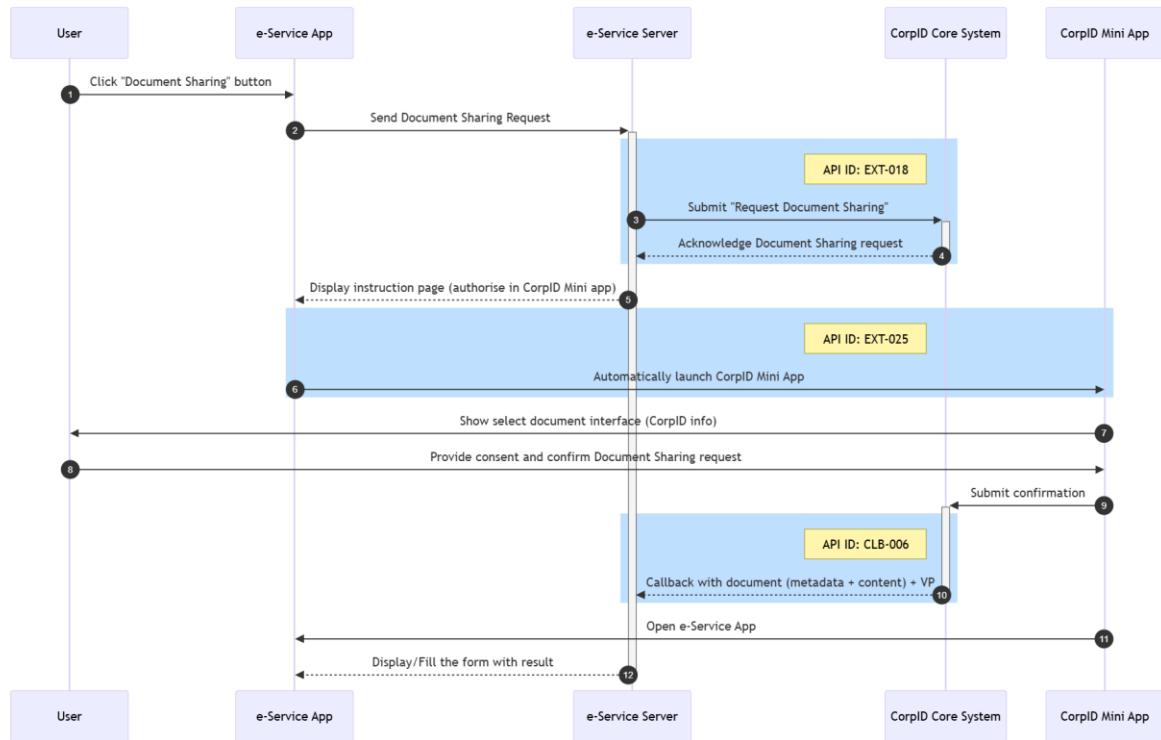
CorPID Document Wallet (e-Service App in Different Device)



API interactions between e-service servers and CorPID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Document Sharing	4.22
	Request Document Sharing (for App e-Service)	4.27
2	Callback to Receive Document Sharing	4.23

3.8.4. Document Sharing (e-service App in Same Device)

CorpID Document Wallet (e-Service App in Same Device)

API interactions between e-service servers and CorpID System are listed below. Technical details of respective APIs can be found at the following sections.

No.	API Name	API Reference (Section)
1	Request Document Sharing	4.22
	Request Document Sharing (for App e-Service)	4.27
2	Open CorpID Mini App for Document Sharing	4.28
3	Callback to Receive Document Sharing	4.23

4. API SPECIFICATION

4.1. AUTHENTICATION BY REQUESTING QR PAGE/ BROKER PAGE FOR ANONYMOUS REQUEST

RESTful API	https://<CorpID_domain>/api/eservice/v1/auth/getQR
Request Type	GET
Service Version	1.0.0
API ID	EXT-001
Description	e-service calls this API to get the QR/App broker page or QR page. After the user authorises anonymous request on the CorpID mini-program, the page will be redirected to the redirectURI with authCode and state parameters. If the user denies it, only the state parameter will be redirected.

4.1.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
clientID	M	String	e-service client identifier. The clientID will be assigned to e-service at the initial registration.	
responseType	M	String	The value MUST be set to code.	
source	M	String	Request initiator.	The supported source values can be found in Appendix B of this specification.
redirectURI	M	String	callback URI.	
scope	M	String	Authorisation Scope for “iAM Smart”, The value should be URL encoded.	The supported authorisation scope values refer to “iAM Smart” API Specification.
cScope	M	String	Authorisation Scope for CorpID, the value should be URL encoded.	The supported authorisation scope values refer to 2.5.

ticketID	M	String	Required in - anonymous form Pre-filling or; - anonymous digital signing;	
lang	0	String	If this parameter is not specified, zh-HK will be shown.	Language to display: - en-US - zh-HK - zh-CN
state	0	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
brokerPage	0	Boolean	If brokerPage is set to true, DeepLink will be leveraged to open the CorpID mini-program. This feature is useful for e-service supporting mobile web versions while triggering the CorpID mini-program or showing a QR page automatically. The default value is false.	

4.1.2. Example Request

```
// Line breaks are for legibility only.

https://<CorpID_domain>/api/eservice/v1/auth/getQR

?clientID=XXXXXX

&responseType=code

&source=Android_Chrome

&redirectURI=XXXXXX

&scope=XXXXXX

&cScope=XXXXXX

&lang=en-US

&state=XXXXXX

&ticketID=XXXXXX
```

4.2. GET ACCESS TOKEN

RESTful API	https://<CorpID_domain>/api/eservice/v1/auth/getToken
Request Type	POST
Service Version	1.0.0
API ID	EXT-002
Description	E-service uses this API to retrieve the access token and tokenised ID (cOpenID). An authorisation code is necessary during the process. The cAccessToken and cOpenID will be used to call corresponding CorpID services subsequently. If corpCount more than 1, e-service should call the get Corporation List, and request switch token once the user selects the exact corp profile user wanted to use.

4.2.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
cAuthCode	M	String	The authorisation code is received from the authorisation server. It can only be used once.	
grantType	M	String	the value MUST be set to	

			authorization_code.	
code_verifier	M (Conditional)	String	This is required for Direct Login (App). The PKCE code verifier received from CorpID Mini-Program during login request.	
isDirectLogin	M (Conditional)	Boolean	e-service must configure callback endpoint (different from the one for normal login) to receive the authorisation code ("cAuthCode") for Direct Login. E-service must submit the parameter "isDirectLogin" and set this value to "true" if the cAuthCode is received by the callback endpoint for Direct Login. If the cAuthCode for normal login or Direct Login is received from other callback endpoints, e-service can consider omitting this parameter or return a "false" value. The default value is "false".	

4.2.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/auth/getToken

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAuthCode": "xxxa42e76bf4cb0846a68e6d83d6096",

  "grantType": "authorization_code"

}
```

4.2.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	cAccessToken value For anonymous flows, it can only be used once.	
tokenType	M	String	Token type, support "Bearer" only	
issueAt	M	Long	The accessToken issue time is expressed in the number of milliseconds since January 1, 1970	

			00:00:00 GMT.	
expiresIn	M	Long	The accessToken expiry time is expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
cOpenID	M	String	Corpid tokenised ID, uniquely generated for each user of each e-service website or mobile application.	
cScope	M	String	The Corpid Authorisation scope of the token. Please refer to the corresponding section specified in each API function.	
corpCount	M	INT	The number of the Corp Profile Count, if more than 1, it means the user had more than 1 Corp Profile and e-Service could call the Get Corp List API for user to select the other Corp Profile(s).	
corpInfo	M (Conditional)	Object	Basic Information of "Corpid" User	
Details of "corpInfo"				
userRole	M	String	User role of the corporation	RP - Responsible Person ADM - Administrator AR - Authorised Representative USR - User
userPrivilege	M	String	Privileges of Corpid user at e-service The value of the parameter is expressed as a list of space-delimited, case-sensitive strings.	- user_access - user_companychop

eServiceRole	O	String	e-Service self-defined role in integrated portal. A Corporate Administrator may assign the user to the e-Service with this pre-defined role for e-Service to determine the user role in this e-Service.	
ubi	M	String	Unique Business Identifier	
corpNameEN	M	String	Corporation Name (English)	
corpNameCH	M	String	Corporation Name (Chinese)	
corpLstUpdT	M	Long	The last update time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
orgCorpUbi	O	String	Unique Business Identifier, of the commissioned corporation of the user, if any.	
orgCorpNameEN	O	String	Original Corporation Name (English), of the commissioned corporation of the user, if any.	
orgCorpNameCH	O	String	Original Corporation Name (Chinese), of the commissioned corporation of the user, if any.	
isSyncPending	M	Boolean	If True, e-Service should be aware the status and information of the Corp Profile may not be latest.	

4.2.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4

// The decrypted body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "M00000",

"msg": "",

"content": {

  "cAccessToken": "XXXXXX",

  "tokenType": "XXXXXX",

  "issueAt": XXXXX,

  "expiresIn": XXXXX,

  "cOpenID": "XXXXXX",

  "cScope": "XXXXXX",

  "corpCount ": 1,

  "corpInfo": {

    "userRole": "RP",

    "userPrivilege": "user_access user_companychop",

    "ubi": "XXXXXXXXXX"

    "corpNameEN": "XXXXXX",

    "corpNameCH": "XXXXXX",

    "corpLstUpdTs": "1764563715",

    "orgCorpUbi": "XXXXXX",

    "orgCorpNameEN": "XXXXXX",

    "orgCorpNameCH": "XXXXXX",

    "isSyncPending": False

  }

}

}
```

4.2.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4

// The decrypted body

{
```

```

"txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

"code": "M40004",

"msg": "cAuthCode not exist or expired"

}

```

4.3. GET CORPORATION LIST

RESTful API	https://<CorpID_domain>/api/eservice/v1/corp/corpList
Request Type	POST
Service Version	1.0.0
API ID	EXT-003
Description	This API allows e-service to get a list of all corporations of the logged-in user. The returned corporation list enables the e-service to present options for the user to select from when switching between corporations.

4.3.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	

4.3.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/corp/corpList

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX....."

}
```

4.3.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
corpList	M	Object Array	Array List of CorpID Profile the user has	
Details of "corpList"				
userRole	M	String	User role of the corporation	RP - Responsible Person ADM – Administrator AR – Authorised Representative USR – User
ubi	M	String	Unique Business	

			Identifier	
corpNameEN	M	String	Corporation Name (English)	
corpNameCH	M	String	Corporation Name (Chinese)	
corpLstUpdT	M	Long	Last update time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
orgCorpUbi	O	String	Unique Business Identifier, of the commissioned corporation of the user, if any.	
orgCorpNameEN	O	String	Original Corporation Name (English), of the commissioned corporation of the user, if any.	
orgCorpNameCH	O	String	Original Corporation Name (Chinese) , of the commissioned corporation of the user, if any.	

4.3.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4

// The decrypted body

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "corpList": [{
      "userRole": "RP",
```

```

    "ubi": "XXXXXXXX"

    "corpNameEN": "XXXXXX",

    "corpNameCH": "XXXXXX",

    "corpLstUpdT": "1764563715",

    "orgCorpUbi": "XXXXXX",

    "orgCorpNameEN": "XXXXXX",

    "orgCorpNameCH": "XXXXXX"

    },...]

}

}

```

4.3.5. Example Failed Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

    "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

    "code": "M20002",

    "msg": "empty parameter {cOpenID}"

}

```

4.4. REQUEST TOKEN EXCHANGE FOR SWITCHING CORPORATION WITH UBI

RESTful API	https://<CorpID_domain>/api/eservice/v1/corp/tokenExchange
Request Type	POST
Service Version	1.0.0
API ID	EXT-004
Description	This API allows e-service to request token exchange for switching corporation with “application/json” content type to CorpID System and return access token to the callbackURL.

4.4.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
cAccessToken	M	String	CorpID accessToken value. The token obtain from Anonymous flow could not be used in Request Token Exchange API	
cOpenID	M	String	CorpID tokenised ID value	
ubi	M	String	Unique Business Identifier For selecting the company without using the CorpID broker page.	

4.4.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/corp/tokenExchange

// Request Headers

"clientId": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAccessToken": "XXXXX.....",

  "cOpenID": "XXXXX.....",

  "ubi": "XXXXXXXXXX"

}
```

4.4.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	cAccessToken value As the anonymous flow would not able to use the Request Token Exchange API, this Token must be able to use more than once.	
tokenType	M	String	Token type, support "Bearer" only	
issueAt	M	Long	The accessToken issue time is expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
expiresIn	M	Long	The accessToken expiry time is expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
cOpenID	M	String	corpID tokenised ID, uniquely generated for each user of each e-service website or mobile application.	
cScope	M	String	The CorpID authorisation scope of the token. Please refer to the corresponding section specified in each API function.	
corpInfo	M (Conditional)	Object	Basic Information of "CorpID" User	
Details of "corpInfo"				
userRole	M	String	User role of the corporation	RP - Responsible Person ADM – Administrator AR – Authorised

				Representative USR – User
userPrivilege	M	String	Privileges of CorpID user at e-service The value of the parameter is expressed as a list of space-delimited, case-sensitive strings.	- user_access - user_companychop
eServiceRole	O	String	e-Service self-defined role in integrated portal. A Corporate Administrator may assign the user to the e-Service with this pre-defined role for e-Service to determine the user role in this e-Service.	
ubi	M	String	Unique Business Identifier	
corpNameEN	M	String	Corporation Name (English)	
corpNameCH	M	String	Corporation Name (Chinese)	
corpLstUpdT	M	Long	Last update time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
orgCorpUbi	O	String	Unique Business Identifier, of the commissioned corporation of the user, if any.	
orgCorpNameEN	O	String	Original Corporation Name (English), of the commissioned corporation of the user, if any.	
orgCorpNameCH	O	String	Original Corporation Name (Chinese) , of the commissioned	

			corporation of the user, if any.	
isSyncPending	M	Boolean	If True, e-Service should be aware the status and information of the Corp Profile may not be latest.	

4.4.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4

// The decrypted body

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "cAccessToken": "XXXXXX",
    "tokenType": "XXXXXX",
    "issueAt": XXXXX,
    "expiresIn": XXXXX,
    "cOpenID": "XXXXXX",
    "cScope": "XXXXXX",
    "corpInfo": {
      "userRole": "RP",
      "userPrivilege": "user_access user_companychop",
      "ubi": "XXXXXXXXXX"
      "corpNameEN": "XXXXXX",
      "corpNameCH": "XXXXXX",
      "corpLstUpdTs": "1764563715",
      "orgCorpUbi": "XXXXXX",
      "orgCorpNameEN": "XXXXXX",

```

```
    "orgCorpNameCH": "XXXXXX"

    "isSyncPending": False

  }

}

}
```

4.4.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M20002",

  "msg": "empty parameter {cOpenID}"

}
```

4.5. OBTAIN PROFILES INFORMATION

RESTful API	https://<CorpID_domain>/api/eservice/v1/profiles
Request Type	POST
Service Version	1.0.0
API ID	EXT-005
Description	E-service can use this API to request the CorpID corporate information ("corpProfileFields", "eCorpFields") and "iAM Smart" user account information ("profileFields") for user identity verification purpose and "e-ME" profile (include the "iAM Smart" user account information) ("eMEFields") for form pre-filling purpose.

4.5.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content Only for "iAM Smart" user	Refer to 2.3.2.1
corpProfileFields	M	String Array	Specify the "corpProfileFields" fields to be requested.	Refer to Appendix A
nonHKResidentFields	M	String Array	Specify the "nonHKResidentFields" fields to be requested.	Refer to Appendix A
authorizationToken	M	Boolean	Specify the "authorizationToken" to be requested.	default value as false
eCorpFields	M	String Array	Specify the "eCorpFields" fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	Array	Specify the "e-ME" fields to be requested.	Refer to "iAM Smart API Specifications"

	)		Only for “iAM Smart” user	
profileFields	M (Conditional)	Array	Specify the profile fields to be requested. Only for “iAM Smart” user	Refer to “iAM Smart API Specifications”

4.5.2. Example Request

```
// Line breaks are for legibility only.

// How to generate shared common parameters is described in Appendix A

POST
https://<CorpID_domain>/api/eservice/v1/profiles

// Request Headers

"clientID": "XXXXXX",

"signatureMethod": "HmacSHA256",

"signature": "XXXXXX",

"timestamp": XXXXX,

"nonce": "XXXXXX",

// Request Body

"content": {

  "businessID": "XXXXXX",

  "cAccessToken": "XXXXXX",

  "cOpenID": "XXXXXX",

  "iAMToken": "XXXXXX",

  "corpProfileFields": ["XXXXXX",XXXXXX],

  "nonHKResidentFields": ["XXXXXX",XXXXXX],

  "authorizationToken": False,

  "eCorpFields": ["XXXXXX","XXXXXX"],

  "profileFields": ["XXXXXX","XXXXXX"],

  "eMEFields": ["XXXXXX","XXXXXX"]
}
```

```
}

```

4.5.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
corp	M	Object	Information of CorpID account.	
iAMContent	O	Object	Information of "iAM Smart" user encrypted with iAM Smart CEK Only for "iAM Smart" user	
Details of "corp"				
Refer to Appendix A				

4.5.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "corp": {
      "ubi": "XXXXXXXX",
      "corpNameEN": "XXXXX",
      "corpNameTC": "XXXXX",
      "corpNameSC": "XXXXX",
      "corpAddr": "XXXXX",
      "corpTypeEN": "XXXXX",
      "corpTypeTC": "XXXXX",
      "corpTypeSC": "XXXXX",
      "corpTel": "XXXXX",
      "corpStatusEN": "XXXXX",
      "corpStatusTC": "XXXXX",
      "corpStatusSC": "XXXXX",

```

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {accessToken}"
}
```

4.6. REQUEST FORM PRE-FILLING

RESTful API	https://<CorpID_domain>/api/eservice/v1/formFilling/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-006
Description	e-service can use this API to request for form pre-filling. If the following eCorpFields and CorpProfileFields are not provided or are empty array, the HTTP code 200 with error code M20002 (empty parameter {eCorpFields, CorpProfileFields}) will be returned.

4.6.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content Only for "iAM Smart" user	Refer to 2.3.2.1
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
source	M	String	Request initiator.	The supported source values can be found in Appendix B of this specification.
redirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state	

			value will be returned toe-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
formName	0	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	0	String	The name of the form should be encoded in Unicode. Maximum Length: 20	
formDesc	0	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
callbackContentType	0	String	Specify the content type of the callback request that an e-service expects to receive in Section 4.7. The callbackContentType supports “application/json” and “multipart/formdata” values. The default value is “application/json”. If thee-service endpoint has a request size limit (Section 4.7), e-service may consider using “multipart/form-data” type.	

corpProfileFields	M	String Array	Specify the “corpProfileFields” fields to be requested.	Refer to Appendix A
nonHKResidentFields	M	String Array	Specify the “nonHKResidentFields” fields to be requested.	Refer to Appendix A
authorizationToken	M	Boolean	Specify the “authorizationToken” to be requested.	default value as false
eCorpFields	M	String Array	Specify the “eCorpFields” fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	String Array	Specify the “e-ME” fields to be requested. Only for “iAM Smart” user	Refer to “iAM Smart API Specification”
profileFields	M (Conditional)	String Array	Specify the profile fields to be requested. Only for “iAM Smart” user	Refer to “iAM Smart API Specification”

4.6.2. Example Request

```
// Line breaks are for legibility only.

// How to generate shared common parameters is described in Appendix A

POST

https://<CorpID_domain>/api/eservice/v1/formFilling/initiateRequest

// Request Headers

"clientID": "XXXXXX",

"signatureMethod": "HmacSHA256",

"signature": "XXXXXX",

"timestamp": XXXXX,

"nonce": "XXXXXX",

// Request Body

"content": {
```

```

"businessID": "XXXXXX",

"cAccessToken": "XXXXXX",

"cOpenID": "XXXXXX",

"iAMToken": "XXXXXX.....",

"source": "Android_Chrome",

"formName": "Example Account Registration Form",

"formNum": "APP0001",

"formDesc": "Example form description",

"callbackContentType": "multipart/form-data",

"redirectURI":
"https://<CorpID_domain>/api/eservice/v1/formFilling/callback",

"state": "XXXXXX",

"corpProfileFields": ["XXXXXX",XXXXXX"],

"nonHKResidentFields": ["XXXXXX",XXXXXX"],

"authorizationToken": False,

"eCorpFields": ["XXXXXX", "XXXXXX"],

"profileFields": ["XXXXXX", "XXXXXX"],

"eMEFields": ["XXXXXX", "XXXXXX"]

}

```

4.6.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
ticketID	M (Conditional)	String	TicketID is a unique identifier to trigger CorpID mini-program when authByQR is false.	
authByQR	M	Boolean	If user completes authentication by scanning QR code (i.e. different device), then "true" will be returned to e-services.	

4.6.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXX",
    "authByQR": false
  }
}
```

4.6.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {redirectURI}"
}
```

4.7. FORM PRE-FILLING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
API ID	CBL-002
Description	This callback allows CorpID System to pass Form Pre-filling information with “application/json” content type to service upon consent.

4.7.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. E-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
corp	M	Object	Information of CorpID account.	
iAMContent	O	Object	Information of “iAM Smart” user encrypted with iAM Smart CEK Only for “iAM Smart” user	Refer to “iAM Smart API Specification” ³ .5.1.3 Callback to Receive “iAM Smart” Profile callback parameters, except “businessID” and “state”
iAMSecretKey	O	String	Base64 encoded encrypted CEK from iAM Smart Only for “iAM Smart” user	Refer to 2.3.2.2
Details of “corp”				
Refer to Appendix A				

4.7.2. Example Callback

```
{
  "code": "M00000",
  "msg": "",
  "content": {
    "corp":{

    },
    "iAMContent": "xxxxx",
    "iAMSecretKey": "xxxxx"
  }
}
```

4.8. REQUEST ANONYMOUS FORM PRE-FILLING

RESTful API	https://<CorpID_domain>/api/eservice/v1/formFilling/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-007
Description	E-service can use this API to request anonymous form pre-filling. If the following corpProfileFields, eCorpFields are not provided or are empty array, the HTTP code 200 with error code M20002 (empty parameter {corpProfileFields, eCorpFields}) will be returned.

4.8.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
formName	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	O	String	The name of the form should be encoded in Unicode. Maximum Length: 20	
formDesc	O	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
corpProfileFields	M	String Array	Specify the "corpProfileFields" fields to be requested.	Refer to Appendix A
nonHKResidentFields	M	String	Specify the	Refer to Appendix A

		Array	"nonHKResidentFields" fields to be requested.	
authorizationToken	M	Boolean	Specify the "authorizationToken" to be requested.	default value as false
eCorpFields	M	String Array	Specify the "eCorpFields" fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	String Array	Specify the "e-ME" fields to be requested. Only for "iAM Smart" user	Refer to "iAM Smart API Specification"
profileFields	M (Conditional)	String Array	Specify the profile fields to be requested. Only for "iAM Smart" user	Refer to "iAM Smart API Specification"

4.8.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/formFilling/anonymous/initiateRequest

// Request Headers

"clientID": "XXXXXX",

"signatureMethod": "HmacSHA256",

"signature": "XXXXXX",

"timestamp": XXXXXX,

"nonce": "XXXXXX",

// Request Body

"content": {

    "businessID": "XXXXXX",

    "formName": "com.onlineservice.myapp",
```

```

"formNum": "callback_activity",

"formDesc": "callback_param",

"corpProfileFields": ["XXXXXX",XXXXXX"],

"nonHKResidentFields": ["XXXXXX",XXXXXX"],

"authorizationToken": False,

"eCorpFields": ["XXXXXX", "XXXXXX"],

"profileFields": ["XXXXXX", "XXXXXX"],

"eMEFields": ["XXXXXX", "XXXXXX"]

}

```

4.8.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
ticketID	M (Conditional)	String	TicketID is a unique identifier to trigger CorpID mini-program when authByQR is false.	

4.8.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {

    "ticketID": "XXXXXX"

  }

}

```

4.8.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {businessID}"
}
```

4.9. CALLBACK WITH AUTHCODE FOR ANONYMOUS

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	GET
API ID	CBL-003
Description	This callback is used to pass authCode to e-service Server. The URI must be registered on the self-service portal.

4.9.1. Callback Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
code	M (Conditional)	String	The authorisation code is generated by the “iAM Smart” System. The authorisation code will be expired 1 minute after issuance. e-service MUST NOT use the authorisation code more than once. An error message will be returned if an authorisation code is expired or re-used. For iAM Smart user only.	
cCode	M	String	The authorisation code is generated by the CorpID System. The authorisation code will be expired 1 minute after issuance. e-service MUST NOT use the authorisation code more than once. An error message will be returned if an authorisation code is expired or re-used.	
state	O	String	If the state parameter has been present in the request message,	

			the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
cError_code	M (Conditional)	String	Return error code when exception.	

4.9.2. Example URL Scheme

Allow

```
// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?businessID=b2c99aa83b0049e9ba370c5341681225

&code=0ad186353c424c64897fcc00445c9ba1

&cCode=0ad186353c424c64897fcc00445c9ba1

&state=eddd527b6
```

Deny

```
// Line breaks are for legibility only.

GET

https://<callback_endpoint>

?cError_code=M20001

&state=eddd527b6
```

4.10. GET ANONYMOUS FORM PRE-FILLING RESULT

RESTful API	https://<CorpID_domain>/api/eservice/v1/formFilling/anonymous/getResult
Request Type	POST
Service Version	1.0.0
API ID	EXT-008
Description	e-service can use this API to retrieve anonymous form pre-filling result.

4.10.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content If the form pre-filling request required iAM Smart data, e-Service shall pass the iAMToken	Refer to 2.3.2.1

4.10.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/formFilling/anonymous/getResult

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX....."

  "iAMToken": "XXXXXX.....",

}

}
```

4.10.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
corp	M	Object	Information of CorpID user	
iAMContent	O	Object	Information of "iAM Smart" user encrypted with iAM Smart CEK Only for "iAM Smart" user	Refer to "iAM Smart API Specification"3. 5.1.3 Callback to Receive "iAM Smart" Profile callback parameters, except "businessID" and "state"
Details of "corp"				

Refer to Appendix A

4.10.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2
// Decrypted Body
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "corp": {
      "ubi": "XXXXXXXX",
      ...
    }
  },
  "iAMContent": "xxxxxx"
}
```

4.10.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {accessToken}"
}
```

4.11. REQUEST DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/eservice/v1/signing/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-009
Description	e-service can use this API making request for digital signing by sending cAccessToken and cOpenID to CorpID system. e-service can use this API for integrated signing with iAM Smart or other remote signing service provider, and company chop only.

4.11.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
iAMToken	O	String	iAM Smart CEK encrypted token content Only for "iAM Smart" user	Refer to 2.3.2.1
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signType	M	String	To determine whether the signing request is for CorpID signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	INTEGRATED_SIGN: For integrated signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or

				remote signing service provider only
requireAuthProof	O	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic proof to verify the signer's identity and authority.	true: CorpID would response with the proof false: CorpID wouldn't response with the proof (Default)
source	M	String	Request initiator.	The supported source values can be found in Appendix B of this specification
redirectURI	M	String	Callback URL.	
state	O	String	If state parameter is presented in the request message, the same value will be returned to e-service during callback. It is used to prevent CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is	

			SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	O	String	If the iAMToken is not set, the HKICHash should not be passed. If HKICHash is not passed, the Signing request will not check the value Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-Service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name.	

			Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
suaMethod	O	JSON String	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-Service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for "iAM Smart" signing.	
suaWaitPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for "iAM Smart" signing.	

4.11.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/signing/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "iAMToken": "xxxxxx...",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "signType": "PERSONAL_SIGN",

  "requireAuthProof": false,

  "hashCode": "XXXXXX.....",

  "HKICHash": " XXXXX",

  "department": "XXXXXX",

  "serviceName": "XXXXXX",

  "documentName": "XXXXXX",

  "suaMethod": "{ \"unary\": [\"FR\"] }, { \"unary\": [\"NFC\"] }, { \"and\": [\"FR\", \"NFC\"] }",

  "suaWaitPeriod": 720

}
```

4.11.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

4.11.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

4.11.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
```

```
"msg": "empty parameter {hashCode}"  
  
}
```

4.12. DIGITAL SIGNING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
API ID	CLB-004
Description	This callback provides the digital signing result to the e-service. The payload structure depends on the signType specified in the initiation request: If signType is PERSONAL_SIGN or INTEGRATED_SIGN (iAM Smart / Remote Signing): The hashCode, timestamp, signature, and cert fields will be included in iAMContent. If signType is CORPID_SIGN (Company Chop Only): The response will include hashCode, timestamp, signature, and cert. The e-service should treat the signing request as failed if no callback is received within 18 minutes.

4.12.1. Callback Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
hashCode	O	String	The document hash submitted by e-service in the API request.	
timestamp	O	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only	

			when digital signing is successful.	
signature	0	String	Base64-encoded signature result string. CorpID System will provide this to e-service only when digital signing is successful.	
cert	0	String	Base64-encoded DER format certificate for the CorpID user. CorpID System will provide this to e-service only when the digital signing is successful.	
signToken	M (Conditional)	String	The signToken should be mapped to the signing request for corporation signing.	
iAMContent	0	Object	Object Information of "iAM Smart" user encrypted with iAM Smart CEK Only for "iAM Smart" user	Refer to "iAM Smart API Specification" 6.5.4 Callback to Receive Digital Signing Result" callback parameters, except "businessID" and "state"
iAMSecretKey	0	String	Base64 encoded encrypted CEK from iAM Smart Only for "iAM Smart" user	Refer to 2.3.2.1
authProof	0	String	The signing request authorization proof in Verifiable Presentation (VP) format	

4.12.2. Example Callback

```
// The descriptions of txID, code, and message are in Section 2.4.2

POST

https://<rp_domain>/<rp_context>/<callback_endpoint>

//Callback Body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "M00000",  
  
"msg": "",  
  
"content": {  
  
    "businessID": "XXXXX",  
  
    "state": "XXXXX",  
  
    "authProof": "XXXXX"  
  
},  
  
"iAMContent": "XXXXX",  
  
"iAMSecretKey": "XXXXX"  
  
}
```

4.13. REQUEST PDF DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/eservice/v1/pdfsigning/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-010
Description	e-service can use this API making request for digital signing by sending accessToken and openID to CorpID System. e-service can use this API for integrated signing with iAM Smart or other remote signing service provider, and company chop only.

4.13.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content Only for “iAM Smart” user	Refer to 2.3.2.1
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signType	M	String	To determine whether the signing request is for CorpID signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	INTEGRATED_SIGN: For integrated signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or remote signing

				service provider only
requireAuthProof	O	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic proof to verify the signer's identity and authority.	true: CorPID would response with the proof false: CorPID wouldn't response with the proof (Default)
source	M	String	Request initiator.	The supported source values can be found in Appendix B of this specification.
redirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to e-service during callback. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	O	String	If the iAMToken is not set, the HKICHash should not be passed.	

			Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
suaMethod	O	JSON String	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-Service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before.	

			This field only for “iAM Smart” signing.	
suaWaitPeriod	0	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for “iAM Smart” signing.	

4.13.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/pdfsigning/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "iAMToken": "xxx...",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "signType": "INTEGRATED_SIGN",

    "requireAuthProof": false,

  "source": "Android_Chrome",

  "redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",

  "state": "XXXXXX",

  "docDigest": "XXXXXX.....",

  "HKICHash": " XXXXX",

  "department": "XXXXXX",

  "serviceName": "XXXXXX",

  "documentName": "XXXXXX",

  "suaMethod":

  "{ \"unary\": [ \"FR\" ] }, { \"unary\": [ \"NFC\" ] }, { \"and\": [ \"FR\", \"NFC\" ] }",
```

```

"suaWaitPeriod": 720
}

```

4.13.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

4.13.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}

```

4.13.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {docDigest}"
}
```

4.14. PDF DIGITAL SIGNING CALLBACK

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
API ID	CLB-005
Description	This callback provides the digital signing result to the e-service. The payload structure depends on the signType specified in the initiation request: If signType is PERSONAL_SIGN or INTEGRATED_SIGN (iAM Smart / Remote Signing): The docDigest and pdfSignature fields will be included in iAMContent. If signType is CORPID_SIGN (Company Chop Only): The response will include docDigest and pdfSignature. The e-service should treat the signing request as failed if no callback is received within 18 minutes.

4.14.1. Callback Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. e-service can use the businessID to relate the callback message with the original request.	
state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by e-service and it should be a secure random value.	
docDigest	O	String	The pdf document digest submitted by e-service in the API request.	
pdfSignature	O	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains signer's certificate, signed hash value,	

			and the digital signing timestamp information. e-service can embed this value to the PDF document for future verification. "iAM Smart" System will provide this to e-service only when the digital signing is successful.	
signToken	M (Conditional)	String	signToken value for corporation signing	
iAMContent	M	Object	Information of "iAM Smart" user encrypted with iAM Smart CEK Only for "iAM Smart" user	Refer to "iAM Smart API Specification" 6.5.5 Callback to Receive PDF Digital Signing Result" callback parameters, except "businessID" and "state"
iAMSecretKey	O	String	Base64 encoded encrypted CEK from iAM Smart	
authProof	O	String	The signing request authorization proof in Verifiable Presentation (VP) format	

4.14.2. Example Callback

```
// The descriptions of txID, code, and message are in Section 2.4.2

POST

// For e-service

https://<rp_domain>/<rp_context>/<callback_endpoint>

//Callback Body

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {
```

```

    "businessID": "XXXXXX",

    "state": "XXXXXX",

    "signToken": "XXXXXX....."

  },

  "iAMContent": "XXXXXX.....",

  "iAMSecretKey": "XXXXXX"

}

```

4.15. ACKNOWLEDGES DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/eservice/v1/signing/ackResult
Request Type	POST
Service Version	1.0.0
API ID	EXT-011
Description	e-service calls this API to acknowledge CorpID System if the result of the digital signature is accepted or not.

4.15.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	
signingResult	M	String	"SR001": digital signature is accepted "SR002": digital signature is rejected "SR003": no digital signature was received	

4.15.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/signing/ackResult

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "signingResult": "SR001"

}
```

4.15.3. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": ""

}
```

4.15.4. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
```

```
"txID": "<T=938ffb193b4b4370b6c2584372c6a588>",  
  
"code": "M20002",  
  
"msg": "empty parameter {signingResult}"  
  
}
```

4.16. REQUEST ANONYMOUS DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/eservice/v1/signing/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-012
Description	e-service can use this API to initiate anonymous digital signing request by sending hashCode and HKICHash (for “iAM Smart” signing) to CorpID System.

4.16.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
signType	M	String	To determine whether the signing request is for CorpID signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	INTEGRATED_SIGN: For integrated signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or remote signing service provider only
requireAuthProof	0	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic	true: CorpID would response with the proof false: CorpID wouldn't response with the

			proof to verify the signer's identity and authority.	proof (Default)
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	M	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64	

			encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
suaMethod	O	JSON String	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-Service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for “iAM Smart” signing.	
suaWaitPeriod	O	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of	

			suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for "iAM Smart" signing.	
--	--	--	---	--

4.16.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/signing/anonymous/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body
{
  "businessID": "XXXXXX.....",
  "signType": "PERSONAL_SIGN",
  "requireAuthProof": false,
  "hashCode": "XXXXXX.....",
  "HKICHash": " XXXXX.....",
  "department": "XXXXXX",
  "serviceName": "XXXXXX",
  "documentName": "XXXXXX",
  "suaMethod":
  "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",
  "suaWaitPeriod": 720
}
```

```
}

```

4.16.3. Response Parameters

Field Name	Man dato ry/O ption al	Data Type	Description	Values/Remark
JSON Object				
ticketID	M	String	ticketID is a unique identifier that will be used to identify this request in the following steps of the digital signing workflow. The ticketID will be expired 12 minutes after issuance.	

4.16.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX"
  }
}
```

4.16.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```
"code": "M20002",  
  
"msg": "empty parameter {hashCode}"  
  
}
```

4.17. REQUEST ANONYMOUS PDF DIGITAL SIGNING

RESTful API	https://<CorpID_domain>/api/eservice/v1/pdfsigning/anonymous/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-013
Description	E-service can use this API making request for PDF digital signing by sending accessToken and openID to CorpID System.

4.17.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
signType	M	String	To determine whether the signing request is for CorpID signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	INTEGRATED_SIGN: For integrated signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or remote signing service provider only
requireAuthProof	O	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic proof to verify the signer's identity	true: CorpID would response with the proof false: CorpID wouldn't

			and authority.	response with the proof (Default)
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the CorpID user is matched with the HKICHash provided by e-service. e-service should convert the HKIC number into hash value using SHA256 before sending to CorpID System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the	

			user is going to sign. Maximum Length: 255	
suaMethod	0	JSON String	{"unary": ["FR"]}, {"unary": ["NFC"]}, {"and": ["FR", "NFC"]}. e-service should have been granted the requested step-up authentication method(s) in the e-Service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field is only for "iAM Smart" signing.	
suaWaitPeriod	0	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field is only for "iAM Smart" signing.	

4.17.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/pdfsinging/anonymous/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "signType": "PERSONAL_SIGN",

  "requireAuthProof": false,

  "docDigest": "XXXXXX.....",

  "HKICHash": " XXXXX.....",

  "department": "XXXXXX",

  "serviceName": "XXXXXX",

  "documentName": "XXXXXX",

  "suaMethod":

  "{ \"unary\": [\"FR\"], { \"unary\": [\"NFC\"], { \"and\": [\"FR\", \"NFC\"] } }",

  "suaWaitPeriod": 720

}
```

4.17.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
------------	------------------------	--------------	-------------	---------------

JSON Object				
ticketID	M (Conditional)	String	ticketID is a unique identifier that will be used to identify this request in the following steps of the digital signing workflow. The ticketID will be expired 12 minutes after issuance.	

4.17.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX"
  }
}
```

4.17.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {hashCode}"
}
```

4.18. GET ANONYMOUS DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/eservice/v1/signing/anonymous/getResult
Request Type	POST
Service Version	1.0.0
API ID	EXT-014
Description	e-service can use this API to retrieve anonymous digital signing result. If the request is getting from iAM Smart, the iAMToken must be passed and the cAccessToken and cOpenID should not be passed. If the request is getting from CorpID, the cAccessToken and cOpenID must be passed and the iAMToken should not be passed.

4.18.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	accessToken value	
cOpenID	M	String	Tokenised ID value	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content If the form pre-filling request required iAM Smart data, e-Service shall past the iAMToken	Refer to 2.3.2.1
businessID	M	String	The businessID used in EXT-012 Request Anonymous Digital Signing	

4.18.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/signing/anonymous/getResult

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "businessID": "XXXXXX.....",

}
```

4.18.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
hashCode	0	String	The document hash submitted by e-service in the API request.	
timestamp	0	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only when digital signing is successful.	
signature	0	String	Base64-encoded signature result string. CorpID System will provide this to e-service only when digital	

			signing is successful.	
cert	0	String	Base64-encoded DER format certificate for the CorpID user. CorpID System will provide this to e-service only when the digital signing is successful.	

4.18.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "hashCode": "XXXXXX",
    "timestamp": 1556450176000,
    "signature": "XXXXXX",
    "cert": "XXXXXX"
  }
}
```

4.18.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {accessToken}"
}
```

4.19. GET ANONYMOUS PDF DIGITAL SIGNING RESULT

RESTful API	https://<CorpID_domain>/api/eservice/v1/pdfsigning/anonymous/getResult
Request Type	POST
Service Version	1.0.0
API ID	EXT-015
Description	e-service can use this API to retrieve anonymous PDF digital signing result. If the request is getting from iAM Smart, the iAMToken must be passed and the cAccessToken and cOpenID should not be passed. If the request is getting from CorpID, the cAccessToken and cOpenID must be passed and the iAMToken should not be passed.

4.19.1. Request Parameters

Field Name	Mandatory/Optional	Data Type	Description	Values/Remark
JSON Object				
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content If the form pre-filling request required iAM Smart data, e-Service shall pass the iAMToken	Refer to 2.3.2.1
businessID	M	String	The businessID used in EXT-013 Request Anonymous PDF Digital Signing	

4.19.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/pdfsigning/anonymous/getResult

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "businessID": "XXXXXX.....",

}
```

4.19.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
docDigest	O	String	The pdf document digest submitted by e-service in the API request.	
pdfSignature	O	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains the signer's certificate, signed hash value, and the digital signing timestamp information. e-service can embed this value	

			to the PDF document for future verification. CorpID System will provide this to e-service only when the digital signing is successful.	
--	--	--	--	--

4.19.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "docDigest": "tGzv3JOkF0XG5Qx2TlKWIA.....",
    "pdfSignature": "nnoadisauflane fhykdj f....."
  }
}
```

4.19.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {accessToken}"
}
```

4.20. REQUEST CORPORATION DIGITAL SIGNING (FOR INTEGRATED SIGNING)

RESTful API	https://<CorpID_domain>/api/eservice/v1/signing/requestCorpSigning
Request Type	POST
Service Version	1.0.0
API ID	EXT-016
Description	e-service can use this API making request for Corporation digital signing as second signing by sending signToken and openID to CorpID system.

4.20.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signToken	M	String	signToken value	
hashCode	M	String	the document hash to be signed. e-service should compute the hash and send this hash to CorpID System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. e-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA. While using NONEwithRSA,	

			hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	

4.20.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/signing/requestCorpSigning

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "signToken": "XXXXXX.....",

  "hashCode": "XXXXXX.....",

  "sigAlgo": "XXXXXX.....",

  "department": "XXXXXX",

  "serviceName": "XXXXXX",

  "documentName": "XXXXXX"

}
```

4.20.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
hashCode	O	String	The document hash submitted by	

			e-service in the API request.	
timestamp	0	Long	Timestamp in milliseconds since January 1, 1970 00:00:00 GMT. CorpID System will provide this to e-service only when digital signing is successful.	
signature	0	String	Base64-encoded signature result string. CorpID System will provide this to e-service only when digital signing is successful.	
cert	0	String	Base64-encoded DER format certificate for the CorpID user. CorpID System will provide this to e-service only when the digital signing is successful.	

4.20.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "hashCode": "XXXXXX",
    "timestamp": 1556450176000,
    "signature": "XXXXXX",
    "cert": "XXXXXX"
  }
}
```

4.20.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
```

```

"code": "M20002",

"msg": "empty parameter {hashCode}"

}

```

4.21. REQUEST CORPORATION PDF DIGITAL SIGNING (FOR INTEGRATED SIGNING)

RESTful API	<a href="https://<CorpID_domain>/api/eservice/v1/pdfsingning/requestCorpSigning">https://<CorpID_domain>/api/eservice/v1/pdfsingning/requestCorpSigning
Request Type	POST
Service Version	1.0.0
API ID	EXT-017
Description	E-service can use this API making request for corporation PDF digital signing as second signing by sending signToken and openID to CorpID System.

4.21.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signToken	M	String	signToken value	
docDigest	M	String	The PDF document digest to be signed. e-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	

serviceName	M	String	The e-service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	

4.21.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/pdfsigning/requestCorpSigning

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "signToken": "XXXXXX.....",

  "docDigest": "XXXXXX.....",

  "department": "XXXXXX",

  "serviceName": "XXXXXX",

  "documentName": "XXXXXX"

}
```

4.21.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
docDigest	O	String	The pdf document digest submitted by e-service in the	

			API request.	
pdfSignature	0	String	Base64-encoded PKCS#7 object that is the actual PDF signature value. It contains the signer's certificate, signed hash value, and the digital signing timestamp information. e-service can embed this value to the PDF document for future verification. CorpID System will provide this to e-service only when the digital signing is successful.	

4.21.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "docDigest": "tGzv3JOkF0XG5Qx2TlKWIA.....",
    "pdfSignature": "nnoadisauflane fhykdj f....."
  }
}
```

4.21.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {docDigest}"
}
```

4.22. REQUEST DOCUMENT SHARING

RESTful API	https://<CorpID_domain>/api/eservice/v1/corp/requestDocSharing
Request Type	POST
Service Version	1.0.0
API ID	EXT-018
Description	e-service can use this API making request for document sharing sending cAccessToken, cOpenID and required document type to CorpID System. CorpID user can select the document(s) at Document Wallet in Mini-Program to share to e-service.

4.22.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request.e-service can use the businessID to relate the callback message with the original request.	
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
docType	O	String	The document type UUID For e-Service to specific which document type it want. For more than one document type required, use “,” to separate. If e-Service does not pass this parameter, CorpID would allow the CorpID user to select any of type of documents.	(Detail document list will be provided in separate document.)
docExpTs	O	Long	The document expiry timestamp in milliseconds since January 1, 1970	

			00:00:00 GMT.	
redirectURI	M	String	callback URL.	
state	M (Conditional)	String	If the state parameter is presented in the request message, the same state value will be returned to e-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by e-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	

4.22.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/eservice/v1/corp/requestDocSharing

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cOpenID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",
```

```

"docType": "XXXXXX",

"docExpTs": XXXXXX,

"redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",

"state": "XXXXXX"

}

```

4.22.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to e-service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger CorpID mini-program when authByQR is false. The value valid for 18 minutes.	

4.22.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {

    "ticketID": "XXXXXX",

    "authByQR": false

  }

}

```

```
}

```

4.22.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {cOpenID}"
}
```

4.23. CALLBACK TO RECEIVE DOCUMENT SHARING

RESTful API	https://<rp_domain>/<rp_context>/<callback_endpoint>
Request Type	POST
Service Version	1.0.0
API ID	CLB-006
Description	This callback will provide the document(s) from Document Wallet to e-service.

4.23.1. Callback Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for e-service to differentiate different request.e-service can use the businessID to relate the callback message with the original request.	

state	O	String	If the state parameter has been present in the request message, the exact value of state will be returned. It is used to prevent the CSRF attack. The value of state is defined by service and it should be a secure random value.	
docList	M	Object Array		
Details of “docList”				
docID	M	String	The ID of documents in Document Wallet	
docName	M	String	Documents name	
docDesc	M	String	Document description	
docCrtTs	M	Long	The create time of the documents in Document Wallet. The time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
docLstUpdTs	M	Long	The last update time of the documents in Document Wallet. The time expressed in the number of milliseconds since January 1, 1970 00:00:00 GMT.	
docType	M	String	The type of document e.g. License, Permit	
docContent	M	Object		
Details of “docContent”				
metadata	O	JSON String	Metadata of document	

file	0	String	Document file in Base64 encoded	
vc	0	String	Document in VC format	

4.23.2. Example Callback

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<rp_domain>/<rp_context>/<callback_endpoint>

// Callback Body

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXX.....",

  "state": "XXXXXX",

  "docList": [

    {

      "docID": "XXXXXX",

      "docName": "XXXXXX",

      "docDesc": "XXXXXX",

      "docCrtTs": "2025-01-01 15:00:00",

      "docLstUpdTs": "2025-01-01 15:00:00",

      "docType": "XXXXXX",
```

```

    "docContent": {
      "metadata": {
        "XXXXXX": "XXXXXX",
        "XXXXXX": "XXXXXX",
        "XXXXXX": "XXXXXX"
      },
      "file": "XXXXX..."
    }
  }, .....
]
}

```

4.24. REQUEST FORM PRE-FILLING (FOR APP E-SERVICE)

RESTful API	<a href="https://<CorpID_domain>/api/eservice/appv1/formFilling/initiateRequest">https://<CorpID_domain>/api/eservice/appv1/formFilling/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-019
Description	E-service (App) can use this API to request for form pre-filling. If the following eCorpFields and corpProfileFields are not provided or are empty array, the HTTP code 200 with error code M20002 (empty parameter {eCorpFields and corpProfileFields}) will be returned.

4.24.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for E-service to differentiate different request. It should be ASCII string with length less than or equal to 36 chars.	

iAMToken	M (Conditional)	String	iAM Smart CEK encrypted token content Only for “iAM Smart” user	
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
source	M	String	App_Link (iOS) or App_Package (Android)	
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to online service during the callback. It is used to prevent the CSRF attack. The value of state is defined by E-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
activityClass	M (Android only)	String	Activity class name of the E-service App for callback use.	
activityParams	O (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
callbackContentType	O	String	Specify the content type of the callback request that an online service expects to receive in Section 4.7. The callbackContentType supports “application/json” and	

			“multipart/formdata” values. The default value is “application/json”. If the online service endpoint has a request size limit (Section 4.7), online service may consider using “multipart/form-data” type.	
state	0	String	If state parameter is presented in the request message, the same state value will be returned to E-service during callback. It is used to prevent the CSRF attack. The value of state is defined by E-service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
formName	0	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
formNum	0	String	The name of the form should be encoded in Unicode. Maximum Length: 20	
formDesc	0	String	The name of the form should be encoded in Unicode. Maximum Length: 255	
corpProfileFields	M	String Array	Specify the “corpProfileFields” fields to be requested.	Refer to Appendix A
nonHKResidentFields	M	String Array	Specify the “nonHKResidentFields” fields to be requested.	Refer to Appendix A

authorizationToken	M	Boolean	Specify the "authorizationToken" to be requested.	default value as false
eCorpFields	M	String Array	Specify the "eCorpFields" fields to be requested.	Refer to Appendix A
eMEFields	M (Conditional)	String Array	Specify the "e-ME" fields to be requested. Only for "iAM Smart" user	Refer to "iAM Smart API Specification"
profileFields	M (Conditional)	String Array	Specify the profile fields to be requested. Only for "iAM Smart" user	Refer to "iAM Smart API Specification"

4.24.2. Example Request

```
// Line breaks are for legibility only.

// How to generate shared common parameters is described in Appendix A

POST

https://<CorpID_domain>/iam/api/appv1/formFilling/initiateRequest

// Request Headers

"clientId": "XXXXXX",

"signatureMethod": "HmacSHA256",

"signature": "XXXXXX",

"timestamp": XXXXX,

"nonce": "XXXXXX",

// Request Body

"content": {

  "businessID": "XXXXXX",

  "cAccessToken": "XXXXXX",

  "cOpenID": "XXXXXX",

  "iAMToken": "XXXXXX.....",

  "source": "App_Package",

  "packageName": "com.onlineservice.myapp",
```

```

"activityClass": "callback_activity",

"activityParams": "callback_param",

"redirectURI": "https://<rp_domain>/<rp_context>/<callback_endpoint>",

"state": "XXXXXX",

"CorpProfileFields": ["XXXXXX",XXXXXX"],

"eCorpFields": ["XXXXXX", "XXXXXX"],

"profileFields": ["XXXXXX", "XXXXXX"],

"eMEFields": ["XXXXXX", "XXXXXX"]

}

```

4.24.3. Response Parameters

Field Name	Mandatory /Optional	Data Type	Description	Values/Remark
JSON Object				
ticketID	M	String (Conditional)	TicketID is a unique identifier to trigger "Corpid" mini-program when authByQR is false.	
authByQR	M	Boolean	If user completes authentication by scanning QR code (i.e. different device), then "true" will be returned to E-service.	

4.24.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {

    "ticketID": "XXXXXX",

```

```

    "authByQR": false

  }

}

```

4.24.5. Example Failed Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M20002",

  "msg": "empty parameter {redirectURI}"

}

```

4.25. REQUEST DIGITAL SIGNING (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/eservice/appv1/signing/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-020
Description	E-service can use this API making request for digital signing by sending accessToken and openID to “CorpID” system.

4.25.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for online service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
iAMToken	O	String	iAM Smart CEK encrypted	Refer to 2.3.2.1

			token content Only for “iAM Smart” user	
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signType	M	String	To determine whether the signing request is for CorpID signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	INTEGRATED_SIGN: For integrated signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or remote signing service provider only
requireAuthProof	O	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic proof to verify the signer's identity and authority.	true: CorpID would response with the proof false: CorpID wouldn't response with the proof (Default)
source	M	String	App_Link (iOS) or App_Package (Android)	
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to online service during the callback. It is used to prevent the CSRF attack. The value of state is defined by E-service, and it should be a secure	

			random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
activityClass	M (Android only)	String	Activity class name of the E-service App for callback use.	
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to online service during callback. It is used to prevent the CSRF attack. The value of state is defined by online service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
hashCode	M	String	the document hash to be signed. E-service should compute the hash and send this hash to "Corpid" System for digital signing. The value should be Base64 encoded.	
sigAlgo	O	String	signature algorithm to be used. E-service can specify either SHA256withRSA or NONEwithRSA or SM3withSM2 or NONEwithSM2. The default value is SHA256withRSA.	

			While using NONEwithRSA, hashCode provided must be hashed with SHA256. While using NONEwithSM2, hashCode provided must be hashed with SM3.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	M (Conditional)	String	Signing request will only be processed when the hash of the HKIC number of the "CorpID" user is matched with the HKICHash provided by E-service. E-service should convert the HKIC number into hash value using SHA256 before sending to "CorpID" System. Only the identifier of HKIC number will be hashed, no check digit is needed e.g.A123456. The value should be Base64 encoded. Only require for "iAM Smart" signing.	
department	O	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The online service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	
suaMethod	O	JSON	{"unary": ["FR"]}, {"unary":	

		String	[{"NFC"}],{"and": [{"FR", "NFC"}]}. E-service should have been granted the requested step-up authentication method(s) in the e-Service Management Portal to use the "NFC", "FR" methods to verify the user. The API returns unsuccessful immediately if the first method fails. No matter if this parameter is specified, FIDO will be used to verify the user at the beginning as before. This field only for "iAM Smart" signing.	
suaWaitPeriod	0	Integer	If suaMethod is provided, this waiting period defines the minimum number of hours since the user account was created. If system defined waiting period is greater than suaWaitPeriod, greater value of system defined waiting period would be used. If suaWaitPeriod is greater than system defined waiting period, greater value of suaWaitPeriod would be used. System default value is zero. Maximum Value: 720 This field only for "iAM Smart" signing.	

4.25.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/iam/api/appv1/signing/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "signType": "PERSONAL_SIGN",

      "requireAuthProof": false,

  "iAMToken": "XXXXXX.....",

  "source": "App_Package",

  "packageName": "com.onlineservice.myapp",

  "activityClass": "callback_activity",

  "activityParams": "callback_param",

  "serverRedirectURI":
"https://<rp_domain>/<rp_context>/<callback_endpoint>",

  "state": "XXXXXX",

  "hashCode": "XXXXXX.....",

  "HKICHash": " XXXXX",

  "department": "XXXXXX",
```

```

"serviceName": "XXXXXX",

"documentName": "XXXXXX",

"suaMethod": "{ \"unary\": [\"FR\"], \"unary\": [\"NFC\"], \"and\": [\"FR\", \"NFC\"] }",

"suaWaitPeriod": 720

}

```

4.25.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to online service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger "Corpid" mini-program when authByQR is false. The value valid for 18 minutes.	

4.25.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {

    "ticketID": "XXXXXX",

    "authByQR": false

  }

}

```

4.25.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M20002",

  "msg": "empty parameter {hashCode}"

}
```

4.26. REQUEST PDF DIGITAL SIGNING (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/eservice/appv1/pdfsigning/initiateRequest
Request Type	POST
Service Version	1.0.0
API ID	EXT-021
Description	E-service can use this API making request for PDF digital signing by sending accessToken and openID to “CorpID” system.

4.26.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for online service to differentiate different request. It could be ASCII string with length less than or equal to 36 chars	
iAMToken	O	String	iAM Smart CEK encrypted token content Only for “iAM Smart” user	Refer to 2.3.2.1
cAccessToken	M	String	CorpID accessToken value	
cOpenID	M	String	CorpID tokenised ID value	
signType	M	String	To determine whether the signing request is for CorpID	INTEGRATED_SIG N: For integrated

			signing only, or requires integrated signing with iAM Smart or other remote signing service provider. If integrated signing is selected, CorpID will process the personal signing first, and then the e-service can proceed with the CorpID signing (refer to Section 4.20).	signing (both personal and company chop) CORPID_SIGN: For signing with company chop only PERSONAL_SIGN: For signing via iAM Smart or remote signing service provider only
requireAuthProof	O	Boolean	Specifies whether to generate a verifiable signing authorization proof for this request. Set to true if the e-service requires a cryptographic proof to verify the signer's identity and authority.	true: CorpID would response with the proof false: CorpID wouldn't response with the proof (Default)
source	M	String	App_Link (iOS) or App_Package (Android)	
clientRedirectURI	M (iOS only)	String	callback redirect URI. The value should be URL encoded and registered in the self-service portal.	
packageName	M (Android only)	String	If the state parameter is presented in the request message, the same state value will be returned to online service during the callback. It is used to prevent the CSRF attack. The value of state is defined by E-service, and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
activityClass	M (Android	String	Activity class name of the E-service App for callback use.	

	only)			
activityParams	M (Android only)	String	Optional params used during callback.	
serverRedirectURI	M	String	callback URI.	
state	O	String	If state parameter is presented in the request message, the same state value will be returned to online service during callback. It is used to prevent the CSRF attack. The value of state is defined by online service and it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphen are accepted.	
docDigest	M	String	The PDF document digest to be signed. E-service should compute the digest using the Adobe.PPKLite filter and the adbe.pkcs7.detached subfilter. The value should be Base64 encoded.	
ubi	M	String	Signing request will only be processed when the UBI number of the corporate is matched with the ubi provided by e-Service.	Plain text ubi of the corporate
HKICHash	O	String	Signing request will only be processed when the hash of the HKIC number of the "Corpid" user is matched with the HKICHash provided by E-service. E-service should convert the HKIC number into hash value using SHA256 before sending to "Corpid" System. Only the identifier of HKIC number will be hashed, no check digit is needed	

			e.g.A123456. The value should be Base64 encoded. Only require for “iAM Smart” signing	
department	0	String	The department that initiates the digital signing request. Maximum Length: 100	
serviceName	M	String	The online service name. Maximum Length: 255	
documentName	M	String	The document name that the user is going to sign. Maximum Length: 255	

4.26.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/iam/api/appv1/pdftsigning/initiateRequest

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "businessID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",

  "cOpenID": "XXXXXX.....",

  "iAMToken": "XXXXXX.....",

  "signType": "PERSONAL_SIGN",

      "requireAuthProof": false,

  "source": "App_Package",

  "packageName": "com.onlineservice.myapp",

  "activityClass": "callback_activity",

  "activityParams": "callback_param",

  "serverRedirectURI":
"https://<rp_domain>/<rp_context>/<callback_endpoint>",

  "state": "XXXXXX",

  "docDigest": "XXXXXX.....",

  "HKICHash": " XXXXX",

  "department": "XXXXXX",
```

```

"serviceName": "XXXXXX",

"documentName": "XXXXXX",

"suaMethod": "{ \"unary\": [\"FR\"], { \"unary\": [\"NFC\"] }, { \"and\": [\"FR\", \"NFC\"] } }",

"suaWaitPeriod": 720

}

```

4.26.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to online service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger "Corpid" mini-program when authByQR is false. The value valid for 18 minutes.	

4.26.4. Example Success Response

```

// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",

  "code": "M00000",

  "msg": "",

  "content": {

    "ticketID": "XXXXXX",

    "authByQR": false

  }

}

```

4.26.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{

  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",

  "code": "M20002",

  "msg": "empty parameter {docDigest}"

}
```

4.27. REQUEST DOCUMENT SHARING (FOR APP E-SERVICE)

RESTful API	https://<CorpID_domain>/api/eservice/v1/corp/requestDocSharing
Request Type	POST
Service Version	1.0.0
API ID	EXT-024
Description	E-service can use this API making request for document sharing sending accessToken, openID and required document type to “CorpID” System. "CorpID" user can select the document(s) at Document Wallet in Mini-Program to share to E-service.

4.27.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
businessID	M	String	businessID is a unique identifier for E-service to differentiate different request. E-service can use the businessID to relate the callback message with the original request.	
cAccessToken	M	String	accessToken value	
cOpenID	M	String	Tokenised ID value	
docType	O	String	The document type UUID	(Detail document list will be provided)

			For e-Service to specific which document type it want. For more than one document type required, use “,” to separate. If e-Service does not pass this parameter, CorpID would allow the CorpID user to select any of type of documents.	in separate document.)
docExpTs	O	String	The document expiry timestamp in milliseconds since January 1, 1970 00:00:00 GMT.	
serverRedirectURI	M	String	callback URI.	
source	M	String	App_Link (iOS) or App_Package (Android)	
clientRedirectURI	M (iOS only)	String	Callback redirect URI. The value should be URL encoded and registered in the self-service portal. Encoded and unencoded commas (“,”) are not accepted.	
packageName	M (Android only)	String	Package name of the E-service App for callback use.	
activityClass	M (Android only)	String	Activity class name of the E-service App for callback use.	
activityParams	O (Android only)	String	Optional params used during callback.	
state	M (Conditional)	String	If the state parameter is presented in the request message, the same state value will be returned to E-service during the callback. It is used to prevent the CSRF attack. The value of state is defined by E-service, and	

			it should be a secure random string of any length less than or equal to 36 (UUID string length). Only ASCII letters, numbers, underscore and hyphens are accepted.	
--	--	--	--	--

4.27.2. Example Request

```
// Line breaks are for legibility only.

// Please refer to Section 2.4 for generating shared common parameters/ header
format.

POST

https://<CorpID_domain>/api/v1/corp/requestDocSharing

// Request Headers

"clientID": "XXXXXX"

"signatureMethod": "HmacSHA256"

"signature": "XXXXXX"

"timestamp": XXXXX

"nonce": "XXXXXX"

// Unencrypted Request Body

{

  "cOpenID": "XXXXXX.....",

  "cAccessToken": "XXXXXX.....",

  "docType": "XXXXXX",

  "docExpTs": "XXXXXX",

  "serverRedirectURI":
  "https://<rp_domain>/<rp_context>/<callback_endpoint>",

  "source": "App_Package",

  "packageName": "com.onlineservice.myapp",

  "activityClass": "callback_activity",

  "activityParams": "callback_param",
```

```
"state": "XXXXXX",
}
```

4.27.3. Response Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
authByQR	M	Boolean	If the user completes authentication by scanning the QR code (i.e. different device), then "true" will be returned to online service.	
ticketID	M (Conditional)	String	ticketID is a unique identifier to trigger "Corpid" mini-program when authByQR is false. The value valid for 18 minutes.	

4.27.4. Example Success Response

```
// The descriptions of txID, code, and message are in Section 2.4.2
{
  "txID": "<T=938ffb193b4b4370b6c2584372c6a588>",
  "code": "M00000",
  "msg": "",
  "content": {
    "ticketID": "XXXXXX",
    "authByQR": false
  }
}
```

4.27.5. Example Failed Response

```
// The descriptions of txID, code, and message are in Section 2.4.2

{
  "txID": "<T=938fffb193b4b4370b6c2584372c6a588>",
  "code": "M20002",
  "msg": "empty parameter {openID}"
}
```

4.28. OPEN CORPID MINI APP FOR DOCUMENT SHARING

RESTful API	<Deeplink_URL>://<CorpID_domain>/document_share
Request Type	POST
Service Version	1.0.0
API ID	EXT-025
Description	The URL scheme makes use of deep linking to redirect users to specific document share in CorpID Mini App. The URL schemes are supported on iOS and Android versions of “CorpID” Mini-Program.

4.28.1. Request Parameters

Field Name	Mandatory/ Optional	Data Type	Description	Values/Remark
JSON Object				
ticketID	M	String	ticketID is a unique identifier provided by “CorpID” System. E-service retrieves ticketID while requesting the document share. It is an ASCII string with a length of less than or equal to 36 chars.	

4.28.2. Example Scheme

```
// Line breaks are for legibility only.

<Deeplink_URL>://<CorpID_domain>/document_share
```

?ticketID=bbb8aae57c104cda40c93843ad5e6db8

5. APPENDIX A

If the field is marked as “Adopt Common Schema”, they are defined in DPO common schema. More information about common schema can be found at https://www.digitalpolicy.gov.hk/en/our_work/data_governance/policies_standards/interoperability_framework/

5.1. corpProfileFields SCHEMA

Field	Mandatory/ Optional	Matching Source	Sub-Field	Type (Chi/Eng)	Max Length (Chi/Eng)	Description	Remark
ubi	M	CR/IRD		String	20	Unique Business Identifier	The CorpID Platform will verify the UBI against the matching source during the corporation's onboarding process.
did	M	N/A		String	300	Decentralised identifier	The CorpID Platform will generate a DID for the corporation during its onboarding process.
crn	O	CR		String	7	Company Registration Number	The CorpID Platform will associate the UBI with the CRN (if applicable) during the corporation's onboarding process.
duns	O	Dun & Bradstreet (D&B) via RCA		String	9	DUNS Number	The CorpID Platform will link the UBI with the DUNS (if available) from the RCA during the corporation's onboarding process.
lei	O	Global Legal		String	20	Legal Entity Identifier	The CorpID Platform will link the UBI with

		Entity Identifier Foundation (GLEIF) via RCA					the LEI (if available) from the RCA during the corporation's onboarding process.
type	M	CR/IRD		String	6	Corporation type code	Please refer to Appendix E for the list of possible values.
status	M	CR/IRD	statusCode	String	6	Corporation status code	Please refer to Appendix E for the list of possible values.
	O		engStatusPlaceholder	String[]	50	The english placeholder of the status description if any	For the status description in Appendix E with {0}, the {0} value would be filled in this fields
	O		chiStatusPlaceholder	String[]	50	The chinese placeholder of the status description if any	For the status description in Appendix E with {0}, the {0} value would be filled in this fields
engName	O	CR/IRD		String	150	English Name	Either engName, chiName, or both will be returned; at least one of these two fields is guaranteed to be present in the response. For the types "Partnership", "Individual" and "Body Unincorporate", the English name is verified against the IRD. For all other

							types, the English name is verified against the CR.
chiName	0	CR/IRD		String	50	Chinese Name	Either chiName, engName, or both will be returned; at least one of these two fields is guaranteed to be present in the response. For the types "Partnership", "Individual" and "Body Unincorporate", the Chinese name is verified against the IRD. For all other types, the Chinese name is verified against the CR.
engAddress	0	IRD/CR's Open Data ¹		String	300	English Registered Address	Either engAddress, chiAddress, or both will be returned; at least one of these two fields is guaranteed to be present in the response. For the types "Partnership", "Individual" and "Body

¹ <https://data.gov.hk/en-data/dataset/hk-cr-crdata-list-addr>

							Unincorporate", the English Registered Address is verified against the IRD. For all other types, the English Registered Address is verified against the open data.
chiAddress	0	IRD/CR's Open Data		String	100	Chinese Registered Address	Either chiAddress, engAddress, or both will be returned; at least one of these two fields is guaranteed to be present in the response. For the types "Partnership", "Individual" and "Body Unincorporate", the Chinese Registered Address is verified against the IRD. For all other types, the Chinese Registered Address is verified against the open data
commencementDate	0	CR/IRD		Date		Commencement date In YYYY-MM-DD format	For the types "Partnership", "Individual" and "Body Unincorporate", the commencement date refers to the BRC commencement date registered with the IRD. For all other

							types, the commencement date refers to the date of incorporation registered with the CR.
cessationDate	0	CR/IRD		Date		Cessation date of Place of Business in Hong Kong In YYYY-MM-DD format	For the types "Partnership", "Individual" and "Body Unincorporate", the cessation date refers to the cessation date registered with the IRD. For all other types, the cessation date refers to the date of cessation of dissolution / ceasing to exist in the CR.
annualReturnsDate	0	CR		Date		Submission date of annual returns In YYYY-MM-DD format	This refers to the submission/filing date of Annual Returns, as obtained from the Document Index Search of the CR.
brcExpiryDate	0	IRD		Date		Expiry Date of Business Registration Certificate In YYYY-MM-DD format	When a corporation imports the Business Registration Certificate (BRC) from the IRD to the Document Wallet of the CorpID Platform, the BRC expiry date

							will be extracted ¹ by the CorpID Platform.
crLstMatchTs	M	N/A		Long		The last matching time between the CorpID Platform and the CR, expressed in number of milliseconds since January 1, 1970 00:00:00 GMT.	
irdLstMatchTs	M	N/A		Long		The last matching time between the CorpID Platform and the IRD, expressed in number of milliseconds since January 1, 1970 00:00:00 GMT.	
openDataLstMatchTs	M	N/A		Long		The last matching time between the CorpID Platform and the open data, expressed in number of milliseconds since January 1, 1970 00:00:00 GMT.	
rcaLstMatchTs	M	N/A		Long		The last matching time between the CorpID Platform and the RCA, expressed in number of milliseconds since	

¹ The CorpID Platform will make every effort to parse the expiry date from the BRC. However, in the event of any discrepancies, the information on the BRC shall prevail.

						January 1, 1970 00:00:00 GMT.	
--	--	--	--	--	--	----------------------------------	--

5.2. nonHKResidentFields SCHEMA

Field	Mandatory / Optional	Matching Source	Sub-Field	Type (Chi/Eng)	Max Length (Chi/Eng)	Description	Remark
enName	M	CorpID	UnstructuredName	String	40	English Name in passport	
chName	O	CorpID	ChineseName	String	6	Chinese Name in passport	
email	M	CorpID		String	128	Email of the non HKID user	
countryCode	M	CorpID		String	3	Issuing state or organization (ISO 3166-1 alpha-3) of passport	
passport	M	CorpID		String	9	Passport	
rcaLstMatchTs	M	N/A		Long		The last matching time between the CorpID Platform and the RCA, expressed in number of milliseconds since January 1, 1970 00:00:00 GMT.	

5.3. eCorpFields SCHEMA

All fields are optional

Field	Mandatory / Optional	Matching Source	Sub-Field	Type (Chi/Eng)	Max Length (Chi/Eng)	Description	Remark
eCorpAddr	O	N/A		JSON		Corporate address	Four different tags exist in corpAddr and

							the returned value can be either one of the followings: ChiPremiseAddress, EngPremiseAddress, FreeFormatAddress, PostBoxAddress The common schema can be found at https://www.digitalpolicy.gov.hk/en/our_work/data_governance/policies_standards/interoperability_framework/
eCorpEmailAddr	0	N/A		String	128	Corporate email address	
eCorpTelNo	0	N/A	countryCode	String	3	Country code of telephone number	
	0	N/A	nationalDestinationCode	String	15	National destination code of telephone number	
	0	N/A	subscriberNumber	String	15	Subscriber number of telephone number	
	0	N/A	extensionNumber	String	7	Extension number of telephone number	
eCorpFaxNo	0	N/A	countryCode	String	3	Country code of FAX number	
	0	N/A	nationalDestinationCode	String	15	National destination code of FAX number	

	O	N/A	subscriberNumber	String	15	Subscriber number of FAX number	
eCorpWebsite	O	N/A		String	200	Corporate website URL	

5.4. “IAM SMART” PROFILE FIELD AND “E-ME” FIELD SCHEMA

(Please refer to “iAM Smart” API Specifications)

6. APPENDIX B

6.1. SUPPORTED VALUE AT SOURCE PARAMETER

CorpID System supports different types of browsers and e-service App calling methods. e-service should detect its user agent value for a browser use case and pass the respective source value to the CorpID System.

Platform	Supported Browser /e-service App calling method	Source
For e-service Web/App in Different Device and For e-service Web in Same Device		
Android Browser	Chrome	Android_Chrome
	Firefox	Android_Firefox
	Edge	Android_Edge
	Samsung built-in browser	Android_Samsung
	Huawei built-in browser	Android_Huawei
	Xiaomi built-in browser	Android_Xiaomi
	In-app browser	Android_IMS_InAppBrowser
iOS Browser	Safari	iOS_Safari
	Chrome	iOS_Chrome
	Firefox	iOS_Firefox
	Edge	iOS_Edge
	In-app browser	iOS_IMS_InAppBrowser
Desktop Browser	Chrome, IE, Edge, Firefox, Safari	PC_Browser

For e-service App in Same Device With CorpID App API v1		
e-service Mobile App	To be invoked by Universal Link (iOS)	App_Link
	To be invoked by App_Package (Android)	App_Package

7. APPENDIX C

7.1. IDENTIFY DOCUMENT TYPE

ID	Name in Form	
HKID	Hong Kong Identity Card	香港身份證
PASSPORT	Passport	護照

8. APPENDIX D

8.1. RETURN CODE

Response Code	Message/ Description
M00000	SUCCESS
Common Error Code	
M10001	The application is suspended. System should redirect to the “Service suspended” page.
M99990	Unhandled exception. System should redirect to the “Unhandled exception” page.
M99991	Cannot send the email notification. Please try again.
M99992	Authorization failed. Applicant is not permitted to submit the request.
S00001	Database error
C00001	Connection refused by backend API server (core)
M20000	unknown exception
M20001	parameter {%s} is missing
M20002	empty parameter {%s}
M20003	invalid parameter {%s}
M20004	duplicated request
M20005	unsupported signature method
M20006	signature verification failed
M20007	unsupported source
M20008	invalid e-service URL The callback URL provided by e-service is not registered in the ESP portal
M20009	accessToken not exist or expired

M20010	cOpenID not exist
M20011	duplicated businessID
M20012	insufficient permission for e-service / Forbidden e-service shall check the scope or API access control in the ESP portal
M20013	UBI mismatch e-service shall ensure that the ubi specified in the initial request is identical to the ubi associated with the cAccessToken obtained earlier. Any mismatch between the two will cause the request to be rejected.
M20015	scope mismatch e-service shall verify the scopes approved for the client id in ESP and the scopes configured in e-service catalogue.
Encryption/Decryption Error Code	
M30001	key encryption key not exist or expired
M30002	content encryption key not exist or expired
M30003	encryption exception
M30004	decryption exception
Authentication Error Code	
M40000	user cancelled authentication request
M40001	user rejected authentication request
M40002	failed to authenticate
M40003	authentication request timeout
M40004	cAuthCode not exist or expired
Profile Request Error Code	
M50001	user rejected profile request
M50002	failed to request profile
M50003	profile request timeout
Form Pre-filling Request Error Code	
M60000	user cancelled Form Pre-filling request
M60001	user rejected Form Pre-filling request
M60002	failed to request Form Pre-Filling
M60003	Form Pre-filling request timeout
Signing Request Error Code	

M70000	user cancelled signing request
M70001	user rejected signing request
M70002	failed to request signing
M70003	signing request timeout
M70004	user not allowed to sign
M70005	inconsistent HKIC number
M70006	failed to process signing acknowledgement
M77000	failed to sign
M77001	failed to sign with FR
M77002	failed to sign with NFC
M77003	device not supported for NFC
M77004	user is still in waiting period
Document Wallet Error Code	
M80000	user cancelled sharing request
M80001	user rejected signing request
M80002	failed to request sharing
M80003	Document sharing request timeout
M88001	failed to share

8.2. HTTP STATUS CODE

Status Code	Status description
200 OK	Request successful
302 Found	The requested resource resides temporarily under a different URI.
400 Bad Request	The request could not be understood by the server due to malformed syntax.
401 Unauthorised	The request has not been applied because it lacks valid authentication credentials for the target resource.
403 Forbidden	The server understood the request but is refusing to fulfil it.

404 Not Found	The server has not found anything matching the Request-URI.
413 Payload Too Large	The uploaded files are larger than 5MB or the request entity is larger than 10MB.
429 Too Many Requests	The user has sent too many requests in a given amount of time.
500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
503 Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

9. APPENDIX E

9.1. CORPORATE TYPE

Type Code	An ASCII string with length less than or equal to 200 chars	
	English	Chinese
C10100	Public company limited by shares	公眾股份有限公司
C10110	Private company limited by shares	私人股份有限公司
C10120	Company limited by guarantee	擔保有限公司
C10200	Public unlimited company	公眾無限公司
C10210	Private unlimited company	私人無限公司
C10300	Registered non-Hong Kong company	註冊非香港公司
C10400	Open-ended Fund Company	開放式基金型公司
C10500	Body corporate formed by Special Ordinance	根據特別條例成立的法人團
C10600	Limited Partnership Fund	有限合夥基金
C10610	Limited Partnership	有限責任合夥
C10700	Mutual Fund	互惠基金
C10800	Registered Trustees Incorporation	註冊受託人法團
C10900	Unregistered Company	非註冊公司
C11000	Unregistered Overseas corporation with charge(s) registered	曾登記押記的非註冊海外法團
C20100	Partnership	合夥
C20200	Individual	個人
C20300	Body Unincorporate	非屬法團的團體

9.2. CORPORATE STATUS

Status Code	An ASCII string with length less than or equal to 200 chars	
	English	Chinese
S10300	Ceased to exist as an entity after amalgamation	合併後不再是獨立的實體
S10310	Ceased place of business	已終止營業地點
S10320	Ceased place of business & dissolved	已終止營業地點及已告解散
S10330	Ceased Place of business & Ceased Registration	已終止營業地點及已終止註冊
S10340	Ceased Place of business, Ceased Registration & Dissolved	已終止營業地點、已終止註冊及已告解散
S10350	Ceased Registration	已終止註冊
S10360	Ceased Registration & Dissolved	已終止註冊及已告解散
S10370	Ceased Registration/ Dissolved	已終止註冊/已告解散
S10200	Dissolved	已告解散
S10100	Live	仍註冊
S10120	Live, Ceased dormancy on {0} ¹	仍註冊，於 {0} 恢復活動 ²
S10130	Live, Commenced dormancy on {0}	仍註冊，於 {0} 休止活動
S10400	Registration of Re-domiciled Company Revoked	經遷冊公司的註冊已遭撤銷
S20100	Live, with no cessation date	仍註冊，沒有終止註冊日期
S20200	Ceased Registration on {0}	已於 {0} 終止註冊

¹ The {0} should refer to the engStatusPlaceholder in Section 5.1

² The {0} should refer to the chiStatusPlaceholder in Section 5.1