Public document

# INTEGRATION GUIDE
for
The Relying parties with Cloud Qualified Electronic Signature service
provided by BORICA

Automated remote signing of electronic documents

CONTENTS

ACRONYMS

CA	Certification Authority
CRL	Certificate Revocation List
CQES	Cloud Qualified Electronic Signature
DN	Distinguished Name
EGN	Uniform civil number assigned to each Bulgarian citizen
LNC	Uniform civil number assigned to a foreigner living in Bulgaria
eIDAS	electronic Identification, Authentication and trust Services (EU Regulation 910/2014)
EU	European Union
HSM	Hardware Security Module
OCSP	On-line Certificate Status Protocol
PIN	Personal Identification Number
PKI	Public Key Infrastructure
QC	Qualified Certificate

For additional information related to this document, please contact the Provider at:

41 “Tsar Boris III” Blvd.
1612 Sofia
BORICA AD
Tel.: 0700 199 10
E-mail: info@borica.bg
Official Web site: www.b-trust.bg

1 Basic Scenario 1 – Automated remote signing of electronic documents

1.1 Step 1: Send consent request

The Relying party's signing application (CAS) specifies the recipient (sends information how to identify the client(and client certificate) of the consent request through the HEADER parameter rpToClientAuthorization. The following options are available:

- personalId: customer's national personal identifier(bulgarian EGN or personal identifier of foreigner(LNC)).
- certId: customer certificate's identifier. This identifier can be found in B-Trust MOBILE application - the second part of the number next to the name of the customer in CQES menu. For example if the information on the screen is IVAN IVANOV(11111-22222) then certId is 22222. CAS should request this information from the customer with B-Trust MOBILE.
- clientToken - customer's client token. In that case the customer is already registered in CAS enetering his profileId and OTP(Authorization code). This is done through /auth function (clientAuthUsingPOST) of this API which returns the client token as a result.

rpToClientAuthorization:

- personalId:put egn (Bulgarian national id)
- clientToken:TPC7416DC60EEEC8252E2531413010AD170
- certId:1234

The consent contains information about the organization and the role/title of the customer in that organisation for which the consent is given in order the CQES customer to sign documents using automatic remote signing. The consent signing request is send to the customer with rpToClientAuthorization header. As result callbackId and validity of the request are returned.

URL: https://cqes-rptest.b-trust.bg/signing-api/v2/arsign/consent
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
rpToClientAuthorization	certId:1234	true
accept	application/json	true
Content-Type	application/json	true

REQUEST BODY

{

"organisation": {

"orgIdentifier": 1234567890, // Legal entity organisation identifier (BULSTAT, EIK, VAT or PAC number)

"orgIdentifierType": "EIK", // Legal entity identifier type (BULSTAT, EIK, VAT or PAC)

"orgName": "BORICA" // Legal entity organisation name

},

"role": "HR manager", // The role of the customer(user) who gives the consent in the organization

"subject": "Signing leave applications", // Reason for automatic remote signing

"expirationTime": 90 // Specifies the validity period of the automatic remote signing acess token (in days between 90 and 365)

}

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 202

{

"data": {

"callbackId": "43ebb3d0-c936-4cc5-a557-7959b512c5ec", // Consent request ID. This ID is used to get the result of the consent with operation getConsent(/arsign/consent/{callbackId})

"validity": "2021-09-14T14:44:49.413+00:00" // End date of the consent validity request (till then relying party can pull the result of the consent)

},

"responseCode": "ACCEPTED", // Response code (status of the response)

"code": "ACCEPTED", // Response code (status of the response)

"message": "The request has been accepted." // Response message. The message can be localized with ‘Accept-language’ header

}

Status code 400

```
{
    "code": "BAD_REQUEST",
    "message": "The request could not be understood by the server due to malformed syntax (invalid request parameters)."
}
```

Status code 401

```
{
    "code": "UNAUTHORIZED",
    "message": "The request is unauthorized."
}
```

Status code 404

```
{
    "code": "NOT_FOUND",
    "message": "The server has not found the signed content."
}
```

Status code 500

```
{
    "code": "ERROR",
    "message": "Internal server error. The server encountered an unexpected condition which prevented it from fulfilling the request."
}
```

1.2 Step 2: CAS uses /arsign/consent/{callbackId} function (getConsent) to check the status of the requested consent and get the client's access token (using "Polling" mechanism)

URL: https://cqes-rptest.b-trust.bg/signing-api/v2/arsign/consent/{callbackId}
METHOD: GET
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD

Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true

REQUEST PARAMETERS

PATH:

- callbackId // Callback ID(request id) - result of the synchronous operation /arsign/consent request (sendConsentRequest)

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 206 Consent request is in progress

```
{
   "data": null,
   "responseCode": "IN_PROGRESS", //Response code (status of the response)
   "code": "IN_PROGRESS", //Response code (status of the response)
   "message": "Consent request is in progress." //Response message. The message can be
localized with 'Accept-language' header
}
```

Status code 200

```
{
   "data": {
      "accessToken": " b52b039d5054c855a1d18328ba00021c7d0d38354c669e
345c8c4a6512167e909a56664c4e805189933cd58dc5e9
cd97", //Access token that CAS(the Relying party) should use in order to sign documents automatic
      "expirationDate": "2021-12-13", //Expiration date of the consent. After this date a new consent
request should be send in order to continue automatic signing
      "consent": "JVBERi0xLjQKJeLjz9MKNSAwIG9iago8PC9GaW..." //Signed PDF
document(BASE64 encoded) with the customer's consent to sign automatic
   },
   "responseCode": "COMPLETED", //Response code (status of the response)
```

"code": "COMPLETED", //Response code (status of the response)

"message": "Sign request is completed." //Response message. The message can be localized with 'Accept-language' header

}

Status code 403

```
{
    "data": null,
    "code": "REJECTED",
    "message": "The request is rejected."
}
```

Status code 404

```
{
    "code": "NOT_FOUND",
    "message": "The server has not found the signed content."
}
```

1.3 Step 3: Send automated signing request and receive response using /arsign/sync operation (sendSyncSignRequest)

With the returned accessToken CAS sends automatic remote signing requests with /arsign/sync operation (sendSyncSignRequest). The requests are automatically signed and returned synchronous as a result of the operation. There is no need of customers confirmation in B-Trust MOBILE application

URL: https://cqes-rptest.b-trust.bg/signing-api/v2/arsign/sync
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
accessToken	b52b039d5054c855a1d18328ba00021c7d0d38354c669e345c8c4a6512167e909a56664c4e805189933cd58dc5e9cd97	true
accept	application/json	true
Content-Type	application/json	true

REQUEST BODY

```
{
  "contents": [ //A list with contents(DOCUMENT, DIGEST or TEXT) that should be signed
    {
        "confirmText": "Document for signing: test.xml", //This parameter is used in order to determine the dialog box(and the text in it) to confirm signing in B-Trust MOBILE
      "contentFormat": "BINARY_BASE64", //Type of the content(in the ‘data’ parameter) that will be signed (DIGEST, BINARY_BASE64 or TEXT)
      "data": "PHRlc3Q+CiAgIDx0aXRsZT7QotC10YHRgtC+0LIg0LTQvtC60YPQvNC10L3Rgjx0aXRsZT4KICA8Y29udGVudD7QotC10YHRgtC+0LLQviDRgdGK0LTRitGA0LbQsNC90LjQtTwvY29udGVudD4KPHRlc3Q+", //The content that will be signed. If the contentFormat parameter is BINARY_BASE64 or DIGEST, then this content(in data parameter) should be BASE64 encoded
      "fileName": "test.xml", //Name of the document(file) that will be signed
      "hashAlgorithm": "SHA256", //Signature digest algorithm (SHA256 or SHA512)
      "padesVisualSignature": false, //This parameter is used in order to specify if the signature in PDF signed file should be visualized in the signed file
      "signaturePosition": { //Specifies the signature position in PDF signed file
        "imageHeight": 100, //Sets a height of the signature field in PDF signed document
        "imageWidth": 100, //Sets a width of the signature field in PDF signed document
        "imageXAxis": 100, //Sets a upper left X coordinate of the signature field
        "imageYAxis": 100, //Sets a upper left Y coordinate of the signature field
        "pageNumber": 1 //Sets a page number where the signature field should be placed NOTE - the counting starts from 1 (one) for the first page of the PDF document
      },
      "signatureType": "XADES_BASELINE_LTA_ENVELOPING", //Signature type algorithm
      "toBeArchived": true //This parameter is used in order to specify that the signed documents should be archived in QLTPS(Qualified Long Term Preservation Service) archive
    }
  ]
}
```

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
   "data": {
      "signatures": [ //List with the signed documents
         {
            "status": "SIGNED",  //Request status
            "signature":   "PD94bWwgdmVyc2lvbj0iMS...",  //BASE64 encoded signature or signed
document.
            "signatureType": "XADES_BASELINE_LTA_ENVELOPING"  //Signature type(format) of the
signature or signed document
         }
      ]
   },
   "responseCode": "COMPLETED", //Response code (status of the response)
   "code": "COMPLETED", //Response code (status of the response)
   "message": "Sign request is completed."   //Response message. The message can be localized
with 'Accept-language' header
}
```

2 Optional Scenario 1 - Validate access token

URL: https://cqes-rptest.b-trust.bg/signing-api/v2/arsign/validate/token
METHOD: GET
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
Content-Type	application/json	true
accept	application/json	true
accessToken	b52b039d5054c855a1d18328ba00021c7d0d38354c669e	true

	345c8c4a6512167e909a56664c4e805189933cd58dc5e9cd97	

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
  "responseCode": "OK", //Response code (status of the response)
  "code": "OK", //Response code (status of the response)
  "message": "Valid access token." //Response message. The message can be localized with
‘Accept-language’ header
}
```

Status code 400

```
{
    "code": " BAD_REQUEST",
    "message": "The request could not be understood by the server due to malformed syntax (invalid
request parameters)”
}
```

Status code 401

```
{
    "code": "UNAUTHORIZED",
    "message": "The request is unauthorized "
}
```