OP Financial Group
Customer Instructions for Corporate Web Services Channel

October 2024

Content

1 Introduction

The Web Services channel (WS channel) is OP Financial Group's electronic data transmission channel for corporate and institutional customers. The channel enables the secure sending and reception of banking and insurance data, messages and orders.

Web Services is a connectivity solution – based on international standards – for machine-to-machine communication between banks and corporate customers. WS channel connection descriptions and specifications jointly issued by various bank consortia are available on Finance Finland's (the FFI's) website at www.finanssiala.fi.

This guide

- describes the operations which must be included in the software used by the customer.
- describes the procedures and practices related to the WS channel that are not covered by the joint message specifications issued by the banks.
- describes the operations and message descriptions of the WS channel and Certificate Service.
- provides advice on how to obtain and use the certificates required by OP Financial Group's WS channel.
- The document also includes instructions on software implementation, and sample data content and messages that can be used in such implementation. It does not describe the content of payment transfer data or account reporting, for which there are separate detailed descriptions.

Software suppliers are advised to follow the 'Information to software suppliers' page on op.fi (https://www.op.fi/corporate-customers/information-to-software-suppliers). The page provides general WS channel information and 'Service notice on payment transactions' updates on service interruptions.

The bank and customer make an agreement on use of the WS channel. In addition, a specific agreement is made on the use of digital services via the WS channel.

1.1 The Certificate Service of the Web Services channel

The Certificate Service generates and manages certificates used to verify WS channel signatures. It also updates and publishes certificate revocation information.

The WS channel verifies the integrity and authenticity of messages and application requests with XML Digital Signature Technology: by means of a digital signature. The recipient must verify their signature, which is used to ensure that the signed message or application request has not been modified after signing and can therefore be trusted.

The currently valid certificates used by the WS channel can be found on op.fi at Corporate customers > Payments and invoicing > Bank connection channel Web Services > Certificate service (www.op.fi/certificate-service).

1.2 Certificate registration and transfer keys

Due to the user rights conferred by a certificate, the customer must visit their bank to verify their identity, so that the certificate can be linked securely to the WS channel username. This first identity verification cannot be performed digitally.

To enable use of the WS channel, the customer's software must have a PKI key pair and a certificate issued by the WS channel's Certificate Service.

When an agreement has been made on use of the WS channel, the customer is issued with a transfer key for obtaining certificates. The WS channel username (10 characters) and the first part of the transfer key (eight characters) are presented in the WS channel agreement. The customer can choose to receive the second part (eight characters) of the transfer key by mobile phone (SMS), or have it sent by post to an address specified by the customer.

Once the customer has both parts of the transfer key (16 characters in most cases), they must enter both parts of the key and the WS channel username into the software and initiate generation of the certificate. The customer's software will send the certificate application request to the Certificate Service and receive the customer certificate in the response message.

1.3 Generation of the key pair

The customer is responsible for generating the key pair used in the WS channel. The bank does not participate in generating the key pair and cannot view or process it.

A key pair will be generated by the customer's software intended for this purpose. During key pair generation, the quality of the algorithm used by the software must be adequate and comply with good encryption practice. The key pair must be generated with an algorithm and method that ensure sufficient randomness. The key length must be 2,048 bits and the algorithm must be RSA; the message digest algorithm for the signature is sha256RSA.

1.4 Use of keys and certificates

Both the application request (ApplicationRequest) and the SOAP message must be signed separately via the WS channel.

The customer's software uses the customer's private key (the private part of the key pair) for the digital signature of messages and application requests in the WS channel.

Alongside the signature, the signing system must provide the certificate corresponding to the private key. The certificate contains the public key which the receiver uses to authenticate the signature. A party with access to the private key can transmit application requests and content to the bank via the WS channel. The bank executes this transfer, which is linked to the private key through the certificate, in the name of the customer.

The certificate is used for linking the public key and thus the key pair to the holder. On WS channel certificates, the Common Name information in the subject field of the certificate bearing the WS channel username serves as the holder's identifier.

The customer is responsible for safekeeping of the private (secret) key and for controlling its use. The private key must not be stored in unencrypted form, nor may its use be permitted without adequate authentication.

1.5 Certificate life cycle and renewal

Each customer certificate is valid for a maximum of two years and must be renewed before its expiry. The customer is responsible for ensuring that the certificate is renewed on time. The customer's software automatically ensures certificate renewal and can verify the certificate end date each time a certificate is used.

At the earliest, a valid certificate can be renewed 60 calendar days before its expiry. If a certificate expires before a new certificate is obtained, the customer must obtain new transfer keys from the bank.

A new key pair must be generated for the renewed certificate. If the customer's software submits a certificate application request for the key pair of a certificate already in use, the bank’s Certificate Service will not generate a new certificate but provide a copy of the previously generated one.

To obtain a new certificate while the prior one is active (within the renewal period of 60 days), new public and private keys must be generated and a Certificate Signing Request (CSR) must be created. If the same keys are used to create a CSR, it will still function as the original CSR and a copy of the prior certificate will be generated. For security reasons, new keys must be used.

A certificate renewal request is similar to requesting a new certificate, except that certificate renewal does not involve the use of a transfer key (CertApplicationRequest.transferKey) and the CertApplicationRequest is signed using the private key for which the username holds a valid certificate. Verification of a renewal request’s authenticity in the Certificate Service is based on the immediately prior certificate issued for the username. The prior certificate must be valid at the time of the request’s submission.

1.6 Submission of a certificate application request and certificate creation

Upon submitting a certificate application request, the customer’s software must check the SSL certificate of the Bank's Certificate Service issued for the domain wsk.op.fi. This is done to ensure that the certificate application request is genuinely routed to the bank's service.

The public key is used for creating a certificate application request in PKCS 10 format.

The subject field of the certificate application request must contain only two items of data:

- C=FI
- CN=[WS channel username, 10 characters]

The first certificate applied for with the username (the first certificate application request made without a certificate) is based on the registration performed at the bank: that is, on a transfer key. In such a case, the CertApplicationRequest.TransferKey element must be a 16-character transfer key and the CertApplicationRequest.CustomerId element must include a 10-character WS channel username. The last character in the transfer key is a verifier, by which the Customer’s software can locally verify that the transfer key has been entered correctly. This verifier is calculated using the Luhn modulo 10 algorithm. No signature is required for the CertApplicationRequest or SOAP message.

In the case of a valid certificate’s renewal (the certificate application request is based on a prior certificate), the CertApplicationRequest must be signed using the same key as that associated with the username of the prior certificate. The CertApplicationRequest.CustomerId element must include a 10-character WS channel username. No signature is required for the SOAP message.

If the customer’s software submits a certificate application request using a serial number, the element CertApplicationRequest.serialNumber must contain the serial number of the certificate. No signature is required for the CertApplicationRequest or SOAP message.

If the public key in the certificate application request is the same as that of a prior certificate for the same username, the bank’s response message will return the prior certificate corresponding to the public key, even if the prior certificate has expired. No error message will appear: the requesting software must detect that the copy relates to an expired certificate and that no new certificate was generated.

1.7 Revocation of a certificate, downloading and use of revocation information

If the customer suspects or is aware of unauthorised access to their private key, the customer must revoke the certificate without delay.

The customer can revoke its certificate by calling 010 252 8470. The 10-character WS channel username or the serial number of the certificate to be revoked is required for this. If a customer has revoked a certificate, the bank will not accept a signature generated by means of the secret key corresponding to the certificate in question.

The bank will publish a certificate revocation list (CRL), which contains the serial numbers of revoked certificates and the revocation reason codes.

The Certificate Service generates a CRL at least once a day and the list will be valid for three days at a time. It may also generate a new CRL when a certificate is revoked.

The CRL's addresses can be found in the trusted certificate's CRL Distribution Points field. The customer's system must download the CRL from the Certificate Service and check the revocation status of trusted certificates from the CRL (CA certificate and bank's service certificates). In practice, this means that the software must check the bank's service certificates contained in the response message.

A certificate can no longer be renewed after it has been revoked. To use a new certificate after a certificate revocation, the customer must re-register at the bank branch and submit a new certificate application request and new transfer key through the WS channel.

1.8 Definitions

Public key

The public part of a key pair used in asymmetric encryption in the public key infrastructure. Data encrypted with a public key can only be decrypted using the key pair's private key. When the holder of a public key is known, an electronic signature performed with the corresponding private key can be verified. The holder of a public key can be reliably identified with a certificate. Public keys are used for verifying signatures and performing encryption.

PKI

A public key infrastructure is a set of technical and administrative solutions used to create, manage, distribute, use, store and revoke public key certificates. Being based on a public key cipher suite, it also defines the controls and standards that Certification Authorities must comply with in their activities to ensure the compatibility, identifiability and availability of electronic certificates.

Transfer key

The authenticity of a certificate application request is verified with a transfer key, which the system includes in the request it sends.

Transport Layer Security

TLS is an encryption protocol used to protect the integrity and transfer of data between two applications.

Certificate

A certificate is a digital document used to link a public key and its holder's information with each other. It is signed by a Certificate Authority. The Certificate Authority’s signature verifies the information in question and the integrity of the certificate.

Certificate application request — A digital document sent by the customer system to the WS channel, containing the customer's public key and identifier. The bank’s Certificate Service generates a certificate based on the certificate application request and sends the certificate to the customer’s system in a response message.

XML signature — Technology used for verifying the authenticity and integrity of an XML document. This signature is made with a private key and authenticated with a public key.

Private key — The private part of the key pair used in a public key infrastructure for asymmetric encryption. A private key is unambiguously designated for a specific party and used to decrypt data encrypted with the key pair's public key.

2 Cipher suites

The customer must use the cipher suites listed below. They contain digest functions, digital signature algorithms and cipher suites.

Cipher suites and protocols protect sensitive, unclassified data.

Encryption software encrypts and decrypts transferred data. OP supports the encryption and decryption of data transferred using TLS version 1.2 or newer. The WS channel supports only TLS protocol 1.2 or newer. Data cannot be sent to OP with the old TLS 1.0 and TLS 1.1.

OP recommends that customers use at least the SHA-256 algorithm.

OP’s WS channel currently supports the following cipher suites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CB_

The suites are known as SHA-2 in accordance with their digest sizes in bits: SHA-256, SHA-384 and SHA-512.

3 Operations of the Web Services channel

The preferred mode of connection for the WSC is an SSL-protected HTTPS connection over the public internet. A digitally signed SOAP message forms the channel’s transport unit. This message contains the XML document ApplicationRequest, which is the actual service request. The ApplicationRequest contains the business content related to the service (e.g. payment transaction files) and is digitally signed.

The customer’s system transmits an application request and the WS channel immediately returns a response. Sent data content is saved on the bank’s side, pending processing. Processing may generate response data, which the customer’s system must download separately.

Using real-time services, the customer's software uploads the data to the bank and immediately receives the real-time service's final response in the response message.

3.1 Transmitting data to the bank

Upon transmission, the WS channel performs a format validation check on the sent data, rejecting any corrupted data. Then, the WS channel immediately returns an error message to the sending software, with error code 12 and the message 'Schema Validation Failed'.

Only one item can be transmitted at a time – one unit of data per message.

3.2 Downloading data from the bank

When downloading, the customer must specify the data for download, citing the data reference (FileReference). References are shown when the data is listed. The customer can then use the reference to download the data.

Only one item at a time can be downloaded.

The data is stored in the WS channel for three months and then deleted automatically. Deletion requires no action by the customer.

The customer may download the same data multiple times. The status of downloaded data is changed from 'NEW' to 'DLD', but the data remains viewable and downloadable.

3.3 Compression of content

We recommend the compression of all data content sent to the bank. In compliance with RFC 1952, the compression algorithm is GZIP. The original content is compressed prior to Base64 encoding and writing in the ApplicationRequest.Content element. The value of ApplicationRequest.compression must be 'true' after compression.

We also recommend compression when downloading content from the bank. Setting the value of ApplicationRequest.compression as 'true' in the download request will compress the downloadable content.

3.4 Real-time services

The WS channel currently provides the real-time services listed below.

Service's name/File type	Description
CustomerCreditTransferInitiantionV02 pain.001.001.02.xsd	C2B SEPA instant credit transfer
CustomerCreditTransferInitiantionV03 pain.001.001.03.xsd	C2B SEPA instant credit transfer
pain.001.001.02 TP4 PS01 and pain.001.001.03 TP4 PS01	C2B SEPA instant credit transfer
pain.001.001.02 TP4 PS01	POPS express transfer, schema version V02. The feedback is pain.002.001.02 TP4 PS01.
pain.001.001.03 TP4 PS01	POPS express transfer, schema version V03. The feedback is pain.002.001.03 TP4 PS01.
TP4 PS01	POPS express transfer
TP1 ES	Transfer between customer's own accounts
TP1 1SS	Account balance query
TP1 1VA	Currency account balance summary
TP1 2ST	Account transaction query

TP1 2SY	Account's extended balance summary
TP1 2KS	Cash pool account balance, debit transactions and credit transactions query
TP1 3ST	Account transactions query (Not yet downloaded account transactions for current day)
ORDER TU	Request for account statement regeneration

Real-time services utilise the uploadFile operation. A request is uploaded to the WS channel in the ApplicationRequest.content element, and the name/FileType of the real-time service is the ApplicationRequest.fileType, e.g. "TP1 1SS".

3.4.1 C2B SEPA instant credit transfer as a real-time service

The service's name/FileType is TP4 PS01.

CustomerCreditTransferInitiantionV02 pain.001.001.02.xsd or CustomerCreditTransferInitiantionV03 pain.001.001.03.xsd. This service can be used to make an individual, real-time SEPA instant credit transfer.

Real-time, individual SEPA instant credit transfers can be made to banks and payment service providers in Finland and elsewhere in the SEPA, provided that they have begun using the SEPA instant credit transfer service.

SEPA instant credit transfers sent as separate urgent payload content types (pain.001.001.02 TP4 PS01 or pain.001.001.03 TP4 PS01) are processed immediately and the sender immediately receives an online response message of the transfers (not in the case of a downloaded payload). Individual SEPA instant credit transfers do not involve due date processing and may be sent 24/7/365. Separate, real-time payments are never transmitted to POPS, but are always processed as SEPA instant credit transfers. If the recipient's bank is unable to process SEPA instant credit transfers, the payment will be rejected.

3.4.2 Express transfer (POPS)

The 'Real-time payment to another Finnish financial institution' service's name/FileType is TP4 PS01.

In the ApplicationRequest.Content element, the client software makes a Base64-encoded application request containing the following:

Name	Length	Description
Control command	11	”$$TP4 PS01 ”
Bank branch of payer	6	5nnnnn
Payer's account number	8	
Payer's name	30	
Bank branch of the payee	6	
Payee's account number	8	
Payee's name	30	
Amount to be transferred	14	presented with cents, see below
Currency unit code	1	1 euro
Due date	10	dd.mm.yyyy; blank until further notice
Reference	20	Starting zeros to be filled in
Message	140	
Paper receipt to the payer	1	"E", no receipt until further notice
Notification to the payee	1	0 no notification 1 phone 2 fax 9 other

Contact details of the payee	70	The contact details of the payee in connection with a notification; in all other cases blank
Timestamp	15	Yymmddmmssnnn, unique identifier
Message version	1	”1”
User key sequence number	1	0 ... 9
Check	16	not in use, leading zeros to be filled in

Sample request for express transfer. In the example, the space characters used in an actual request are represented by dots, to illustrate their number and position.

```
$$TP4.PS01.57803820021333Saku.Eeroila..................13934600001181Simo.Sammila..................0
0000000000001127.11.20110000000000000000001245.............................................................................
.....................................................................E0..........................................................................110727145700000
100000000000000000
```

Received input acknowledgement of express transfer request

An input acknowledgement of an urgent payment request comprises two files: the input acknowledgement file and the end-of-event file ($$EOF) for the OP transaction. Such an input acknowledgement can also be the $$ERROR error message returned by the OP service, e.g. PERMISSION ERROR or NO RESPONSE FROM HOST. The client software must allow for a response time that is longer than usual, of up to 120 seconds (the transaction may be processed in another financial institution). If the OP service does not return an input acknowledgement or returns the $$ERROR – NO RESPONSE FROM HOST error message, the client software must alert the customer to contact the bank or check the success status of the express transfer, by making a current day account statement request, for example. If a transaction corresponding to the express transfer is displayed on the account, this means that the transfer has been successfully executed.

The system calculates a unique MAC (Message Authentication Code) CheckSum for the input acknowledgement file, in compliance with the PATU standard. The CheckSum is calculated using the user key, from the start of the input acknowledgement file up to the 'CheckSum' field, in a similar manner as for all other PATU messages (ESI, SUO, VAR and PTE).

Name	Length	Description
Success status code	2	“00” Succeeded All other numerical values are errors and the explanation text provides the error reason, e.g. "REJECTED, INSUFFICIENT FUNDS".
Explanation text	80	Explanation text, in the language of the customer
Filing code	22	Included if operation succeeds; otherwise blank
Timestamp	15	Yymmddmmssnnn
Message version	1	”1”
User key sequence number	1	0 ... 9
Check	16	Not in use, zeros to be added

3.4.3 Real-time payment – credit transfer between customer's own accounts

The name and FileType of the ‘Real-time payment – credit transfer between customer's own accounts’ service is TP1 ES.

In the ApplicationRequest.Content element, the client software makes a Base64-encoded application request containing the following:

$$TP1 ES X vknro vtnro hknro htnro euroAmount message

where:

- X is the character X
- vknro is the branch to be debited, presented using 6 characters
- vtnro is the account number to be debited, presented using 8 characters
- hknro is the branch to be credited, presented using 6 characters
- htnro is the account number to be credited, presented using 8 characters
- euro amount is the amount to be transferred, presented in cents without a decimal separator, using max. 11 characters
- message is the message to be forwarded, presented using max. 70 characters in double quotes

Sample credit transfer where EUR 1,500 is transferred from account 500015-118 to account 500015-22228 with the message, “Sample credit transfer”:

- $$TP1 ES X 500015 10000018 500015 20002228 150000 “Credit transfer”

Response message to credit transfer application request

Name	Length	Description
Record sequence number	1	1
Response type	1	1=OK, other=error*
Reserved for future use	3	
Transaction branch	6	
Payment terminal code	2	
Transaction number	4	
Account holder	15	
Date	6	ddmmyy
Branch code debited	6	
Account number debited	8	
Balance of the account debited	11	presented with cents without decimal separator
Balance prefix	1	+/-
Branch code credited	6	
Account number credited	8	
Reserved for future use	12	
Amount in euros transferred	11	presented with cents without decimal separator
Prefix	1	+
Currency code	1	1=euro

3.4.4 Balance query

The name and FileType of the ‘Balance query’ service is TP1 1SS.

In the ApplicationRequest.Content element, the client software makes a Base64-encoded application request containing the following:

$$TP1 1SS BranchCode AccountNumber X

- the length of the branch code is 6 characters
- the length of the account number is 8 characters
- X is the character X.

The response to the balance query is returned in the ApplicationResponse.content element and has the following structure.

Name	Length	Description
Record sequence number	1	=1
Response type	1	1=OK, other=error*
Reserved for future use	3	

Transaction branch code	6	
Payment terminal code	2	
Transaction number	4	
Account holder	15	
Branch code	6	
Account number	8	
Date	6	ddmmyy
Balance	11	2 decimals
Balance prefix	1	+/-
Credit limit	11	2 decimals
Credit limit prefix	1	+/-
Funds available for withdrawal	11	2 decimals
Available funds prefix	1	+/-
Currency code	1	1=euro

3.4.5 Currency account balance summary

The name and FileType of the ‘Currency account balance summary’ service is TP1 1VA.

The bank connection software can use a control message to request a currency account balance summary.

$$TP1 1VA X account type currency code

where:

- X is the character X
- the account type is AV-KP (=OP currency account), MTA (=fixed-term OP currency account) or ALL (=all currency accounts)
- the currency code is the currency’s ISO code (e.g. USD) or ALL (=all currencies)

The response to a balance summary query comprises one or several records, only the last of which includes the currency code.

Name	Length	Description
Name of customer	15	
Number of accounts in this record	3	
Will more balance records be transmitted	1	0=no, 1=yes
Account (0-n)		
Branch code	6	
Account number	8	
Type of account	5	
Currency code	3	
Interest rate, %	6	4 decimals
Balances (3)		
Currency amount	13	2 decimals
Prefix	1	+/-
Entry date	6	ddmmyy
Equivalent value	13	2 decimals
Prefix	1	+/-
Average price	10	7 decimals
Number of amount	2	
Amount (0-n)		
Total, EUR	15	
Prefix	1	+/-
Entry date	6	ddmmyy

Currency code	1	1=euro This field is only included in the last record.

A value of 1 is given in the additional accounts field, if there are additional account records. The record includes a maximum of three records containing account data. Account summaries are presented by record; the final record does not therefore include a total of all balances in the message – they must be separately calculated for each record.

Each account record includes three balances, which present the balances of possible future entry dates. The entry field value of a missing balance is zero.

3.4.6 Account transaction query

The name and FileType of the 'Account transaction query' service is TP1 2ST.

The bank connection software may use a control message to request data on account transactions.

$$TP1 2ST BranchCode AccountNumber X

where:

- the length of the branch code is 6 characters
- the length of the account number is 8 characters
- X is the character X.

Transaction query response

Name	Length	Description
Record sequence number	1	=1
Response type	1	1=OK, other=error
Reserved for future use	3	
Transaction branch	6	
Payment terminal code	2	
Transaction number	4	
Account holder	15	
Branch code	6	
Account number	8	
Date	6	ddmmyy
Transactions (10)		
Transaction date	6	ddmmyy
Description	12	
Amount of money	11	2 decimals
Prefix	1	+/-
Balance	11	2 decimals
Prefix	1	+/-
Credit limit	11	2 decimals
Prefix	1	+/-
Preauthorisations	11	2 decimals
Prefix	1	+/-
Funds available for withdrawal	11	2 decimals
Prefix	1	+/-
Currency code	1	1=euro

3.4.7 Accounts' extended balance summary

The name and FileType of the 'Accounts' extended balance summary' service is TP1 2SY.

The bank connection software may use a control message to request an extended balance summary.

$$TP1 2SY

The balance summary response consists of one or more records.

Name	Length	Description
Name of customer	40	
Number of accounts in this record	3	
Will more balance records be transmitted	1	0=no, 1=yes
Account and balance (0-n)		
Branch code	6	
Account number	8	
Balance	13	11 integers + 2 decimals
Prefix	1	+/-
Funds available for withdrawal	13	11 integers + 2 decimals
Prefix	1	+/-
Interest rate	6	4 decimals
Balance date	8	yyyymmdd

The message's record length varies.

3.4.8 Cash pool account current day account statement query

The name and FileType of the 'Cash pool account balance, debit transactions and credit transactions query' service is TP1 2KS.

In the ApplicationRequest.Content element, the client software makes a Base64-encoded application request containing the following:

$$TP1 2KS BranchCode AccountNumber X

where:

- the length of the branch code is 6 characters
- the length of the account number is 8 characters
- X is the character X.

Response message to cash pool account current day account statement query

Name	Length	Description
Record sequence number	1	1
Response type	1	1=OK, other=error*
Reserved for future use	3	
Transaction branch	6	
Payment terminal code	2	
Transaction number	4	
Name of the account holder	15	
Account-holding branch code of the cash pool account	6	
Account number of the cash pool account	8	
Date	6	ddmmyy
Balance	13	2 decimals
Prefix	1	+/-
Current day debit transactions	13	2 decimals
Prefix	1	+/-
Current date credit transactions	13	2 decimals

Prefix	1	+/-
Currency code	1	1=euro

3.4.9 Current day transaction statement query

The name and FileType of the ‘Current day bank statement for transactions not yet downloaded’ service is TP1 3ST.

In the ApplicationRequest.Content element, the bank connection software makes a Base64-encoded application request containing the following:

$$TP1 3ST BranchCode AccountNumber X

where:

- the length of the branch code is 6 characters
- the length of the account number is 8 characters
- X is the character 1, if all transactions of the day, including those already downloaded, are requested; in all other cases, the service returns only new transactions, not yet downloaded transactions for the WS channel username (CustomerID).

Response message record descriptions
Records are separated from one another using the record separators. Each record ends with the 'carriage return' and 'line feed' symbols.

The basic record of the current day transaction statement

Field	Name	Format	Description
1	File identifier	AN1	S
2	Record identifier	AN2	00
3	Record length	N3	322
4	Version number	AN3	001
5	Account number	AN14	
6	Current day transaction statement no.	AN3	Blank
7	Query date		
	.1 Start date	N6	YYMMDD
	.2 End date	N6	YYMMDD
8	Generation time		
	.1 Current date	N6	YYMMDD
	.2 Time	N4	HHMM
9	Client ID	AN17	
10	Not in use	N6	
11	Not in use	AN19	
12	Not in use	N6	
13	Account currency code	AN3	ISO code
14	Account name	AN30	
15	Account limit	AN18	16 integers + 2 decimals
16	Account holder	AN35	
17	Bank’s name	AN40	
18	Not in use	AN40	
19	Not in use	AN30	
20	Not in use	AN30	
	TOTAL	322	

Field 4 indicates the software version used to generate the current day account statement.

Field 7 The start date and end date are identical: they are the query date.

Field 9 indicates the customer ID assigned by the bank to the account holder and the specifier, if applicable (the country code or standard code and specifier fields are initially blank).

- country code X(4) or .1 standard code X(4)
- customer ID X(8) .2 customer specifier X(10)
- customer specifier X(5) .3 customer specifier X(3)

Field 15 indicates the limit of a checking account with a credit line. No limit is associated with the account, if the field contains only zeros. The field indicates the limit of a sub-account under a cash pool account.

The basic record of a transaction

Field	Name	Format	Description
1	File identifier	AN1	S
2	Record identifier	AN2	10
3	Record length	N3	188
4	Transaction generation time	N6	HHMMSS
5	Original archiving code	AN18	
6	Entry date	N6	YYMMDD
7	Value date	N6	YYMMDD
8	Payment date	N6	YYMMDD
9	Transaction code	AN1	1, 2, 3, 4
10	Posting description .1 Code .2 Description	 AN3 AN35	
11	Transaction amount .1 Prefix .2 Amount	 AN1 N18	 16 integers + 2 decimals
12	Receipt code	AN1	E = itemisations to be excluded from the current day transaction statement
13	Transfer method	AN1	
14	Payee/Payer .1 Name .2 Source of name data	 AN35 AN1	 space character, A, J, or K
15	Payee's account .1 Account number .2 Account changed data	 AN14 AN1	 blank, *
16	Reference	AN20	
17	Form number	AN8	
18	Level code	AN1	0
	TOTAL	188	

Field 5 indicates the archiving code, assigned by the bank that generated the transaction and which can be used to trace the original payment order. The archiving code indicates the date on which the bank processed the payment order, and the bank branch or system involved.

VVKKPP XXXXXXXXXXXX^______________________________identifier
^____ date

The identifier of the archiving code is bank-specific. The first two characters indicate the code of the relevant bank group.

Field 9 contains the transaction code whose values are:

- 1 = credit to account
- 2 = debit to customer account
- 3 = correction of credit to account
- 4 = correction of debit to account

Note! Any correction of a correction must be either transaction type 1 (credit to account) or 2 (debit to account).

Field 10 contains a posting description that indicates the service through which, or describes how, the transaction is being posted in the account-holding bank. The primary purpose of the posting description code is to enable automated posting of the account transactions in the customer's bookkeeping. Identifying codes are assigned to transactions to be posted automatically, while generic codes are applied to all other transactions. The code values are common to all banks, but the description texts are bank-specific. In the corrections, the codes are used both for credit and debit transactions.

The values of the posting description code are:

- 700 = payment transfer service credit to customer account/debit to customer account
- 701 = recurring payment service credit to customer account/debit to customer account
- 702 = bill payment service debit to customer account
- 703 = payment terminal service credit to customer account
- 704 = direct debiting service/automatic payment service
- 705 = reference payment service credit to customer account
- 706 = payment service debit to customer account
- 710 = credit to customer account
- 720 = debit to customer account
- 721 = card payment debit to customer account
- 722 = checking account debit to customer account
- 723 = taxi/bus voucher debit to customer account
- 730 = collection of fee debit to customer account
- 740 = collection of interest debit to customer account
- 750 = payment of interest credit to customer account
- 760 = loan (including repayment, interest, and fee) debit to customer account
- 761 = loan repayment debit to customer account

Field 12 contains a receipt code, which indicates whether receipt information is to be provided on the bank statement, as a separate paper document, or as an itemisation of the individual transactions in binary form.

The receipt code values are:

- space character = the bank does not issue a paper receipt to the customer.
- E = an itemisation is linked to the transaction.
- P = the bank issues a paper receipt to the customer.

Field 13 contains a transfer method code, assigned by the bank which received the payment order, indicating how the payment order was transferred to the bank and where the original instruction is stored. In sorting situations, the transfer method is used to determine the party to be contacted for additional information on the transaction. Where the transfer method value is A, the sorting request is addressed to the initiator of the payment instruction. The account-holding bank must be contacted in all other cases.

The transfer method code values are:

- A = The customer has sent the payment in machine-readable format or has paid it via self-service. The original payment order is with the customer.

- J = The transaction is generated by the bank's system. The reason for its generation is available at the system sorting point indicated by the archiving code.
- K = The transaction is executed at a bank branch and saved by the bank’s employee. The payment order can be retrieved using the archiving code.

Field 14 contains the name of the counterparty to the transaction, where available. This information is not available for batch transactions. The name is either the name of the payee, in the case of an individual payer transaction, or the name of the payer in the case of an individual payee transaction. The source of the name information is included only in transactions where the ‘Payee/Payer' information is present, and indicates the source of the Payee/Payer name forwarded.

The values for the 'source of name' are:

- A = The name data originates from binary content submitted by the customer, or is saved by the customer through self-service.
- J = The name data is retrieved on the basis of the bank's register code.
- K = The name data is saved by a bank employee at a branch.

In a payer transaction, **Field 15** contains the payee's account number, included by the payer's bank upon the transfer of the transaction. The payer can use this data to check into which account the payment was made. The 'account changed' data is linked only to the payee's account number and indicates that the account originally provided by the payer has changed in the bank's system.

The values for the 'account changed' data are:

- space character = not changed
- * = changed

Additional record of a transaction

Field	Name	Format	Description
1	File identifier	AN1	S
2	Record identifier	AN2	11
3	Record length	N3	
4	Type of additional information	AN2	
5	Additional information	ANnnn	
	TOTAL	8+nnn	

The additional record of the transaction comprises the first part, common to all additional records, and the additional information, whose length varies according to the type of additional information.

Open-ended message, type = 00			
5.1	Message - 1	AN35	
5.2	Message - 2	AN35	
...	....		
5.12	Message - 12	AN35	
	TOTAL	Max. 420	

Number of transactions, type = 01			
5.1	Number of transactions	N8	
	TOTAL	8	

Billing transaction data, type = 02

5.1	Customer number	AN10	
5.2	Blank	AN1	
5.3	Invoice number	AN15	
5.4	Blank	AN1	
5.5	Date of invoice	AN6	YYMMDD
	TOTAL	33	

Card transaction data, type of additional information = 03			
5.1	Card number	AN19	
5.2	Blank	AN1	
5.4	Merchant's archiving reference	AN14	
	TOTAL	34	

Correction event data, type = 04			
5.1	The original archiving code for the transaction being corrected	AN18	
	TOTAL	18	

Currency transaction data, type of additional information = 05			
5.1	Equivalent value		
	.1 Prefix	AN1	
	.2 Amount	N18	16 integers + 2 decimals
5.2	Blank	AN1	
5.3	ISO currency code	AN3	
5.4	Blank	AN1	
5.5	Currency exchange rate	N11	4 integers + 7 decimals
5.6	Exchange rate reference	AN6	
	TOTAL	41	

Originator information, type = 06			
5.1	Originator information-1	AN35	
5.2	Originator information -2	AN35	
	TOTAL	70	

Additional information provided by the bank, type = 07			
5.1	Additional information-1	AN35	
5.2	Additional information-2	AN35	
...	....		
5.12	Additional information-12	AN35	
	TOTAL	Max. 420	

Payment purpose data, type = 08			
5.1	Payment purpose code	N3	
5.2	Blank	AN1	
5.3	Description of the payment purpose	AN31	
	TOTAL	35	

Name specifier data, type = 09			
5.1	Specifier of the payee/payer name	AN35	
	TOTAL	35	

Balance record

Field	Name	Format	Description
1	File identifier	AN1	S
2	Record identifier	AN2	40
3	Record length	N3	50
4	Query date	N6	YYMMDD
5	Balance at the time of query .1 Prefix .2 Amount	 AN1 N18	 16 integers + 2 decimals
6	Available funds .1 Prefix .2 Amount	 AN1 N18	 16 integers + 2 decimals
	TOTAL	50	

The notice record is forwarded to the customer only if the query fails or if the data is not up to date due to disruptions in the service.

Field	Name	Format	Description
1	File identifier	AN1	S
2	Record identifier	AN2	70
3	Record length	N3	
4	Bank group code	AN3	
5	Bulletin .1 Row-1 (e.g. cause of disturbance) 6 Row-6	 AN80 AN80	
	TOTAL	Max 489	

3.4.10 Request for account statement regeneration

The name/FileType of the ‘Request for account statement regeneration’ service is ORDER TU.

The request is sent in the following form:

$$ORDER TU StartDate EndDate BranchCode AccountNumber

where:

- the start date is the start date of the bank statement period presented as yyyymmdd
- the end date is the end date of the bank statement period presented as yyyymmdd
- the length of the branch code is 6 characters
- the length of the account number is 8 characters

If the request succeeds, the service returns the response code 00 OK. The bank statement is regenerated according to the bank statement generation schedule and can be downloaded on the following morning.

3.5 Listing of content

The customer’s system can download a listing of available content from the WS channel. The following search criteria can be used in the listing:

- The moment of saving content on the channel within a given period, delimited by the date.

- Content status information

    - for content sent by the customer
        - WPF – pending processing (‘Waiting for Processing’)
        - FWD – forwarded for further processing (‘Forwarded’)

    - for content downloadable by the customer
        - DLD – downloaded (‘Downloaded’)
        - NEW – not downloaded (‘New’)

- Content type – e.g. ‘pain.001.001.02’ or ‘pain.002.001.02’.

Content deleted by the customer, using the deleteFile operation, will not be shown in the listing.

The content sent to the bank by the customer and content available to the bank for download by the customer are shown on the content list. By applying appropriate filters to the getFileList operation, the customer’s software can select the content to be displayed in the list.

3.6 Deletion of content

The customer can use the deleteFile operation to delete any content they have sent to the bank. Deletion of content simply changes the status of the content from ‘WFP’ to ‘DEL’. Such a status change only denies entry of the content for further processing, but has no other consequences. Deleted content cannot be viewed by means of the getFileList operation.

Deletion of content is only feasible within the time slot from transmission to entry into processing. Content cannot be deleted once it is being processed.

The time during which the content is pending further processing in the WS channel varies in accordance with the service and the content type. For example, C2B payment transaction content is processed on business days at 2.30 and from 7.00 to 18.00 at half-hour intervals.

3.7 Administrator and authorisations

Authorisation related to payment transfer content is based on the Generator role for the WS channel username. The so-called administrator identifier is created from the CustomerID value entered in the WS channel agreement for the username in question, and from the location number which is a parameter for the username. This administrator identifier – the location – must be included in the allowed senders list or as an allowed receiver of downloadable content in the payment transfer agreement applicable to the processing and generation of content.

The administrator is the party entered in the payment transfer agreement as the allowed sender or receiver of the content. The administrator has a dedicated WC channel agreement, the related usernames and the certificates linked to the usernames.

3.8 Sample messages and application requests

3.8.1 Request message

A sample SOAP request message for the getFileList operation is presented below. The Base64-encoded content elements have been shortened and the omitted parts replaced with three dots for better readability.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
```

```
   <env:Header>
      <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
      wssecurity-secext-1.0.xsd" env:mustUnderstand="1">
         <wsse:BinarySecurityToken wsu:Id="bst_ag0md1SPzDjcLWHg" xmlns:wsu="http://docs.oasis-
         open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
         ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
         profile-1.0#X509v3" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
         200401-wss-soap-message-security-
         1.0#Base64Binary">MIIC9TCCA...z2nIv3xpHPU=</wsse:BinarySecurityToken>
         <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
            <dsig:SignedInfo>
               <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
               c14n#"/>
               <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
               <dsig:Reference URI="#Body_87p1SixC35qs3Lpk">
                  <dsig:Transforms>
                     <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <exc14n:InclusiveNamespaces
                        xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
                     </dsig:Transform>
                  </dsig:Transforms>
                  <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <dsig:DigestValue>ztKnhKXLpBQM/r3we3D0BdVeibE=</dsig:DigestValue>
               </dsig:Reference>
               <dsig:Reference URI="#Timestamp_MpXSne5nUJot8ltt">
                  <dsig:Transforms>
                     <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <exc14n:InclusiveNamespaces
                        xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
                     </dsig:Transform>
                  </dsig:Transforms>
                  <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <dsig:DigestValue>NRvpjFck2OEDAcgy0WxxV1WTz3w=</dsig:DigestValue>
               </dsig:Reference>
            </dsig:SignedInfo>
            <dsig:SignatureValue>UPzp6yAQ...6Od5+GRI0w==</dsig:SignatureValue>
            <dsig:KeyInfo>
               <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-
               open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
               xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
               utility-1.0.xsd" wsu:Id="str_2u1tu89DgKYG7uPe">
                  <wsse:Reference URI="#bst_ag0md1SPzDjcLWHg" ValueType="http://docs.oasis-
                  open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
               </wsse:SecurityTokenReference>
            </dsig:KeyInfo>
         </dsig:Signature>
         <wsu:Timestamp wsu:Id="Timestamp_MpXSne5nUJot8ltt" xmlns:wsu="http://docs.oasis-
         open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2011-08-16T11:42:28Z</wsu:Created>
            <wsu:Expires>2011-08-16T13:22:28Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </env:Header>
   <env:Body wsu:Id="Body_87p1SixC35qs3Lpk" xmlns:wsu="http://docs.oasis-
   open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <cor:downloadFileListin xmlns:cor="http://bxd.fi/CorporateFileService">
         <mod:RequestHeader xmlns:mod="http://model.bxd.fi">
            <mod:SenderId>1000000000</mod:SenderId>
            <mod:RequestId>1313494952760</mod:RequestId>
            <mod:Timestamp>2011-08-16T14:42:28.031+03:00</mod:Timestamp>
            <mod:Language>FI</mod:Language>
            <mod:UserAgent>OP Client</mod:UserAgent>
            <mod:ReceiverId>OKOYFIHH</mod:ReceiverId>
         </mod:RequestHeader>
         <mod:ApplicationRequest
         xmlns:mod="http://model.bxd.fi">PD94bWwg...ZXF1ZXN0Pg==</mod:ApplicationRequest>
      </cor:downloadFileListin>
   </env:Body>
</env:Envelope>
```

3.8.2 Response message

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <wsse:Security S:mustUnderstand="1">
         <wsu:Timestamp wsu:Id="_3" xmlns:ns11="http://docs.oasis-open.org/ws-sx/ws-
         secureconversation/200512" xmlns:ns10="http://www.w3.org/2003/05/soap-envelope">
            <wsu:Created>2011-08-16T11:42:29Z</wsu:Created>
            <wsu:Expires>2011-08-16T11:47:29Z</wsu:Expires>
         </wsu:Timestamp>
         <wsse:BinarySecurityToken wsu:Id="uuid_5ac774c6-d670-4168-be0f-084dcb8d92ac"
         EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
         security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-
         200401-wss-x509-token-profile-1.0#X509v3" xmlns:ns11="http://docs.oasis-open.org/ws-
         sx/ws-secureconversation/200512" xmlns:ns10="http://www.w3.org/2003/05/soap-
         envelope">MIID2DCC...iuycKgsL6euA==</wsse:BinarySecurityToken>
         <ds:Signature Id="_1" xmlns:ns11="http://docs.oasis-open.org/ws-sx/ws-
         secureconversation/200512" xmlns:ns10="http://www.w3.org/2003/05/soap-envelope">
            <ds:SignedInfo>
               <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
               c14n#"/>
               <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
               <ds:Reference URI="#_5002">
                  <ds:Transforms>
                     <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                  </ds:Transforms>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <ds:DigestValue>lkuQU09sgqWIp02wRR1BDxCrxyk=</ds:DigestValue>
               </ds:Reference>
               <ds:Reference URI="#_3">
                  <ds:Transforms>
                     <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                  </ds:Transforms>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <ds:DigestValue>dz7uPSeuk9tmjOU777o6/+GczFE=</ds:DigestValue>
               </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>BDV8Ctp...8rc0GX95w==</ds:SignatureValue>
            <ds:KeyInfo>
               <wsse:SecurityTokenReference>
                  <wsse:Reference ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-
                  200401-wss-x509-token-profile-1.0#X509v3" URI="#uuid_5ac774c6-d670-4168-be0f-
                  084dcb8d92ac"/>
               </wsse:SecurityTokenReference>
            </ds:KeyInfo>
         </ds:Signature>
      </wsse:Security>
   </S:Header>
   <S:Body wsu:Id="_5002">
      <ns2:downloadFileListout xmlns="http://model.bxd.fi"
      xmlns:ns2="http://bxd.fi/CorporateFileService">
         <ResponseHeader>
            <SenderId>1000000000</SenderId>
            <RequestId>1313494952760</RequestId>
            <Timestamp>2011-08-16T14:42:29.672+03:00</Timestamp>
            <ResponseCode>00</ResponseCode>
            <ResponseText>OK.</ResponseText>
            <ReceiverId>OKOYFIHH</ReceiverId>
         </ResponseHeader>
         <ApplicationResponse>PD94bWwgd...BvbnNlPg==</ApplicationResponse>
      </ns2:downloadFileListout>
   </S:Body>
</S:Envelope>
```

3.8.3 Application request getFileList

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ApplicationRequest xmlns="http://bxd.fi/xmldata/">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T09:48:31.177+03:00</Timestamp>
   <Status>NEW</Status>
   <Environment>TEST</Environment>
   <SoftwareId>soft</SoftwareId>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>sPNzEb+Mf5dchY5MTGq7GL1grEg=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>aIqreFNkxuy...nM4SXE8g==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIIC9TCCA...Iv3xpHPU=</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationRequest>
```

3.8.4 Application response getFileList

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationResponse xmlns="http://bxd.fi/xmldata/"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T09:48:33.668+03:00</Timestamp>
   <ResponseCode>00</ResponseCode>
   <ResponseText>OK.</ResponseText>
   <FileDescriptors>
      <FileDescriptor>
         <FileReference>5802</FileReference>
         <TargetId>MLP</TargetId>
         <ParentFileReference>5801</ParentFileReference>
         <FileType>pain.002.001.02</FileType>
         <FileTimestamp>2011-07-29T12:00:16.483+03:00</FileTimestamp>
         <Status>NEW</Status>
      </FileDescriptor>
      <FileDescriptor>
         <FileReference>5803</FileReference>
         <TargetId>MLP</TargetId>
         <ParentFileReference>5801</ParentFileReference>
         <FileType>pain.002.001.02</FileType>
         <FileTimestamp>2011-07-29T12:01:16.971+03:00</FileTimestamp>
         <Status>NEW</Status>
      </FileDescriptor>
   </FileDescriptors>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>LlpU5jyiDd5kO5FjJDIL7AWZyBQ=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>WKtQ1t8V1...LkGV9DMz0cQ==</SignatureValue>
      <KeyInfo>
         <X509Data>
```

```
            <X509Certificate>MIID1zCCAr...JKaoOlc5gLu</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationResponse>
```

3.8.5 Application request getFile

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationRequest xmlns="http://bxd.fi/xmldata/">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T13:01:46.911+03:00</Timestamp>
   <StartDate>2011-08-15+03:00</StartDate>
   <Environment>TEST</Environment>
   <FileReferences>
      <FileReference>5803</FileReference>
   </FileReferences>
   <Compression>true</Compression>
   <SoftwareId>soft</SoftwareId>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>OQA4fiudfd6KJKR0KINTsE9Fyxc=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>c2RzFUa...9VBAnMQ==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIIC9TC....v3xpHPU=</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationRequest>
```

3.8.6 Application response getFile

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationResponse xmlns="http://bxd.fi/xmldata/"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T13:01:49.591+03:00</Timestamp>
   <ResponseCode>00</ResponseCode>
   <ResponseText>OK.</ResponseText>
   <Compressed>true</Compressed>
   <CompressionMethod>RFC1952</CompressionMethod>
   <Content>H4sIAAAA...epSdAwAA</Content>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>gQf1Tmlhw7KdS7MT10L5yaTDmm4=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>bzS0Itu...U/y6jRg==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIID1zCCA...oOlc5gLu</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationResponse>
```

3.8.7 Application request uploadFile

```
<ApplicationRequest xmlns="http://bxd.fi/xmldata/">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T13:01:31.990+03:00</Timestamp>
   <Environment>TEST</Environment>
   <TargetId>target</TargetId>
   <Compression>true</Compression>
   <SoftwareId>soft</SoftwareId>
   <FileType>pain.001.001.02</FileType>
   <Content>H4sIAAA...KU0HAAA=</Content>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>o9/bmBaH58Phw01oiQS/ttrP/sY=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>NwNRa...dTtMMqvg==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIIC9TC...nIv3xpHPU=</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationRequest>
```

3.8.8 Application response uploadFile

An example of a validation error in the pain.001.001.02 content sent by the customer.

```
<ApplicationResponse xmlns="http://bxd.fi/xmldata/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:date="http://exslt.org/dates-and-times">
<CustomerId/>
<Timestamp>2018-03-16T17:14:38+02:00</Timestamp>
<ResponseCode>12</ResponseCode>
<ResponseText>Schema validation failed. - Tranid = 661232927</ResponseText>
<Compressed>false</Compressed>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
   <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <Reference URI="">
         <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
         </Transforms>
         <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <DigestValue>TlA6ACHFI9HVswrPCi6jhA10G14=</DigestValue>
      </Reference>
   </SignedInfo>
<SignatureValue>o9F1TZvdEFTeb09aBSf6TzGmCE/F09jd...S5YAiEGZtxvfR/FqO3i6u5P9VfK0cCy6czYqJs9Ew
==</SignatureValue>
   <KeyInfo>
      <X509Data>
         <X509Certificate>MIIGLzCCBBegAwIBAgIDKCf...POM88+Y+luwn7HmqB</X509Certificate>
         <X509IssuerSerial>
            <X509IssuerName>C=FI, CN=CUSTOMER TEST OP Services CA V2</X509IssuerName>
            <X509SerialNumber>2631673</X509SerialNumber>
         </X509IssuerSerial>
      </X509Data>
   </KeyInfo>
</Signature>
</ApplicationResponse>
```

The service returns the following message in the event of another type of schema error:

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationResponse xmlns="http://bxd.fi/xmldata/"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T13:01:34.851+03:00</Timestamp>
   <ResponseCode>12</ResponseCode>
   <ResponseText>Schemavalidation failed.</ResponseText>
   <FileType>pain.002.001.02</FileType>
   <Content>PD94bWw...dW1lbnQ+Cg==</Content>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>3GyOY2gXwgT7RFP8CIli4KQ5kcg=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>cBs4Lm...QvD1Q==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIID1zC...aoOlc5gLu</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationResponse>
```

In this error example, the Application Request.content element contains the following pain.002.001.02 data (in Base64-encoded form). For further information on the content and usage of payment feedback, see the separate customer instructions on C2B payments.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<pain.002.001.02>
<GrpHdr>
<MsgId>1313401940313</MsgId>
<CreDtTm>2011-08-15T12:52:20+03:00</CreDtTm>
</GrpHdr>
<OrgnlGrpInfAndSts>
<NtwkFileNm>1313401937067</NtwkFileNm>
<OrgnlMsgNmId>pain.001.001.02</OrgnlMsgNmId>
<GrpSts>RJCT</GrpSts>
<StsRsnInf>
<StsOrgtr>
<Id>
<OrgId>
<PrtryId>
<Id>1000000000</Id>
</PrtryId>
</OrgId>
</Id>
</StsOrgtr>
<StsRsn>
<Cd>NARR</Cd>
</StsRsn>
<AddtlStsRsnInf>pain.001.001.02 could not be processed, please verify structure.cvc-
datatype-valid.1.2.1: 'A1001.00' is n</AddtlStsRsnInf>
<AddtlStsRsnInf>ot a valid value for 'decimal'.cvc-complex-type.2.2: Element 'InstdAmt' must
have no element [children],</AddtlStsRsnInf>
<AddtlStsRsnInf>and the value must be valid.</AddtlStsRsnInf>
</StsRsnInf>
```

```
</OrgnlGrpInfAndSts>
</pain.002.001.02>
</Document>
```

3.8.9 Application request deleteFile

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationRequest xmlns="http://bxd.fi/xmldata/">
   <CustomerId>1000000000</CustomerId>
   <Timestamp>2011-08-15T09:53:53.778+03:00</Timestamp>
   <StartDate>2011-08-15+03:00</StartDate>
   <Environment>TEST</Environment>
   <FileReferences>
      <FileReference>6152</FileReference>
   </FileReferences>
   <SoftwareId>soft</SoftwareId>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
         20010315#WithComments"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
            <Transforms>
               <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>TsZYDgKXMO6/nfTlGGFGlHL43pI=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>dgUhp4b...qelFFvQ==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIIC9TCCAd2g...Iv3xpHPU=</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>
</ApplicationRequest>
```

3.8.10 Application response deleteFile

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplicationResponse xmlns="http://bxd.fi/xmldata/"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
     <CustomerId>1000000000</CustomerId>
     <Timestamp>2011-08-15T09:53:56.147+03:00</Timestamp>
     <ResponseCode>00</ResponseCode>
     <ResponseText>OK.</ResponseText>
     <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
          <SignedInfo>
               <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
               20010315#WithComments"/>
               <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
               <Reference URI="">
                    <Transforms>
                         <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
                         signature"/>
                    </Transforms>
                    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                    <DigestValue>F4NXYMUcrwJ83p92msZ48Jga7+c=</DigestValue>
               </Reference>
          </SignedInfo>
          <SignatureValue>OUjhFKVG...qL5xb4MQ==</SignatureValue>
          <KeyInfo>
               <X509Data>
                    <X509Certificate>MIID1zCC...aoOlc5gLu</X509Certificate>
               </X509Data>
          </KeyInfo>
     </Signature>
</ApplicationResponse>
```

4 Certificate service messages and service requests via the Web Services channel

4.1 The SHA1 certificate will be replaced with the SHA256 certificate

OP Financial Group will no longer support the SHA1 certificate and digital signature. These will be replaced with the SHA256 certificate.

The old SHA1 service will be closed down on 31 August 2025, after which customers must use the SHA256 algorithm. Content will not be transmitted through the Web Services channel from 1 September 2025, if the bank connection software uses the old certificate/TLS encryption protocol.

Customers must update their software with the SHA256 algorithm to enable ApplicationRequest and SOAPRequest operations.

- SignatureMethod Algorithm=http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- DigestMethod Algorithm=http://www.w3.org/2001/04/xmlenc#sha256

Correspondingly, response messages are signed using the SHA256 certificate and algorithm.

Addresses for the bank connection production environment:

- https://wsk.op.fi/services/OPCertificateServiceV2
- https://wsk.op.fi/services/CorporateFileServiceV2

4.2 Message descriptions for the Certificate Service

The structure of SOAP messages and the address of the Certificate Service are described in a WSDL file.

The SOAP message does not require a signature within the Certificate Service. Authenticity is verified by a signature at application request level only (CertApplicationRequest).

The WSDL file is available at:

- SHA1: https://wsk.op.fi/wsdl/MaksuliikeWS.xml.
- SHA256: https://wsk.op.fi/wsdl/MaksuliikeWSV2.xml

4.3 Application requests and schemas

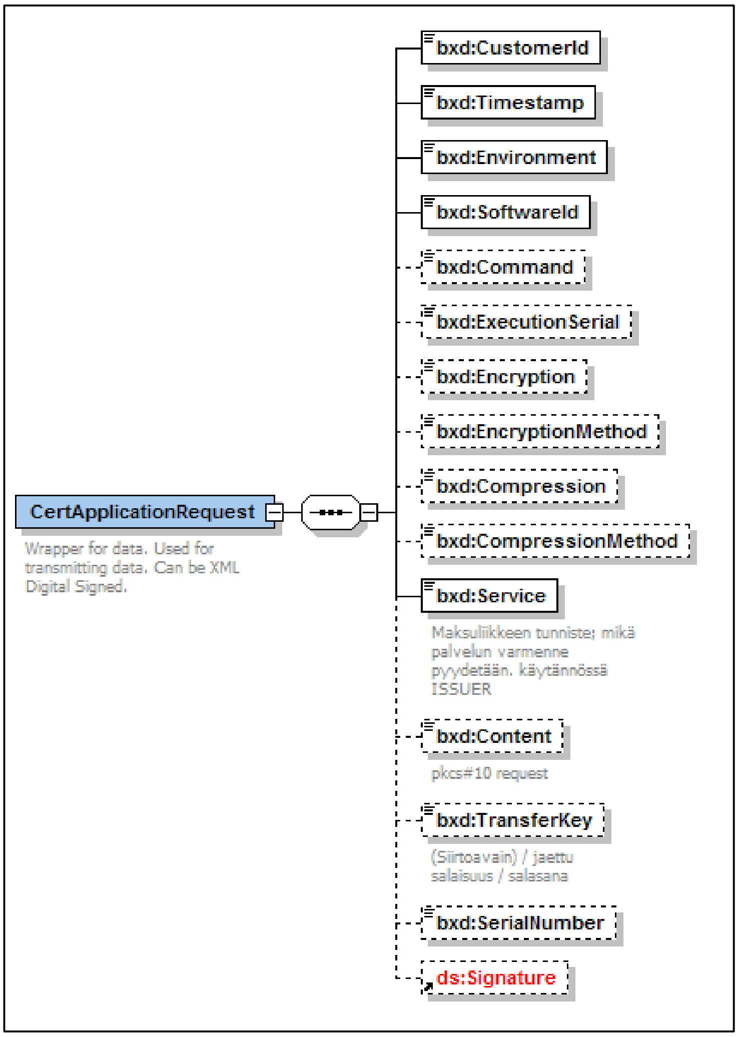
The XML Schema files describe the application request and application response wrapped in the message.

The WSDL for the Certificate Service is available at

- SHA1: https://wsk.op.fi/wsdl/MaksuliikeCertService.xml
- SHA256: https://wsk.op.fi/wsdl/MaksuliikeCertServiceV2.xml

The application request submitted by the customer is called the CertApplicationRequest and the application response returned by the bank is termed the CertApplicationResponse.

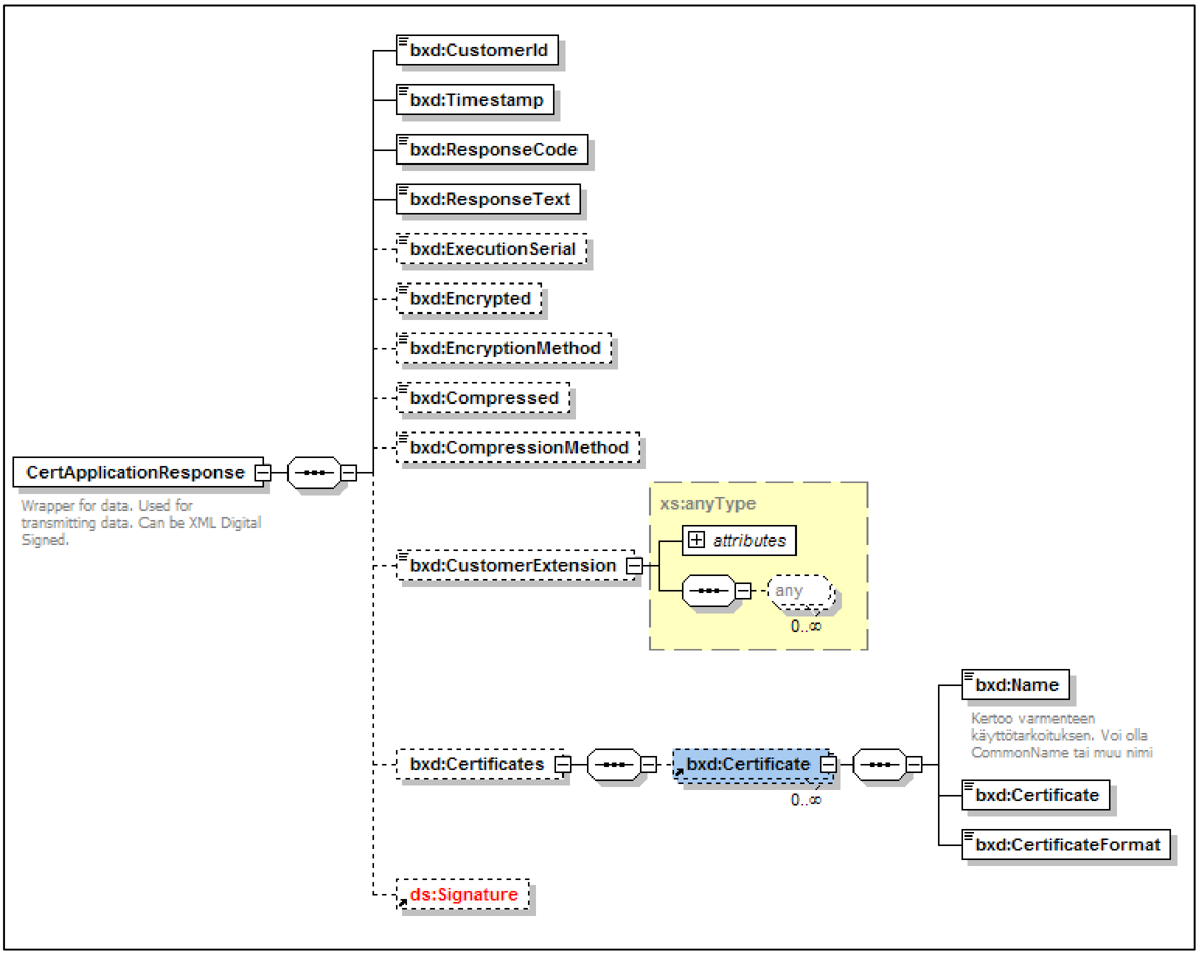
4.3.1 CertApplicationRequest



In the case of a certificate application request, the main elements to be entered in the application request are as follows:

- CustomerId – the WS channel username of the party requesting a certificate, 10 characters
- Content – certificate application request in PKCS10 format, Base64-encoded
- TransferKey – transfer key (16 characters) in the case of submission of the first certificate application request, under the username in question
- Signature – XML signature in the case of certificate renewal
- Mandatory information:
  - Timestamp – timestamp for the generation of the application request (in most cases, in support of sorting only)
  - Environment – in the production environment case, ‘PRODUCTION’ (otherwise, the request will be rejected)
  - SoftwareId – name of software submitting the application request (in most cases, in support of sorting only)
  - Service – MATU

4.3.2 CertApplicationResponse



4.4 Sample request for the Certificate Service

4.4.1 Request message

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Header/>
    <env:Body>
        <opc:getCertificatein xmlns:opc="http://mlp.op.fi/OPCertificateService">
            <opc:RequestHeader>
                <opc:SenderId>1000012222</opc:SenderId>
                <opc:RequestId>123</opc:RequestId>
                <opc:Timestamp>2010-01-26T14:32:43.800+02:00</opc:Timestamp>
            </opc:RequestHeader>
            <opc:ApplicationRequest>PD94bWwgdmVy... GlvblJlcXVlc3Q+</opc:ApplicationRequest>
        </opc:getCertificatein>
    </env:Body>
</env:Envelope>
```

4.4.2 Response message

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Header/>
    <env:Body>
        <opc:getCertificateout xmlns:opc="http://mlp.op.fi/OPCertificateService">
            <opc:ResponseHeader>
                <opc:SenderId>1000012222</opc:SenderId>
                <opc:RequestId>123</opc:RequestId>
                <opc:Timestamp>2010-01-26T14:32:45.909+02:00</opc:Timestamp>
                <opc:ResponseCode>00</opc:ResponseCode>
                <opc:ResponseText>OK.</opc:ResponseText>
            </opc:ResponseHeader>
            <opc:ApplicationResponse>PD94bWwgdmVyc2...
            W9uUmVzcG9uc2U+</opc:ApplicationResponse>
        </opc:getCertificateout>
    </env:Body>
</env:Envelope>
```

4.4.3 Application request for certificate renewal

The example shows an application request for certificate renewal made with username (CustomerID) 1000000047. The application request is signed, because identification and authentication are based on a valid certificate held by the same username.

```
<?xml version="1.0" encoding="UTF-8"?>
<CertApplicationRequest xmlns="http://op.fi/mlp/xmldata/">
    <CustomerId>1000000047</CustomerId>
    <Timestamp>2010-01-26T14:32:44.191+02:00</Timestamp>
    <Environment>TEST</Environment>
    <SoftwareId>soft</SoftwareId>
    <Compression>false</Compression>
    <Service>MATU</Service>
    <Content>MIICZzCCAU8CA... 3slAmKGflLvw==</Content>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
            20010315#WithComments"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <Reference URI="">
                <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
            signature"/>
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <DigestValue>i81y7OKgB8FBmOlv4gQWNtcCmLg=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>ZWSGuxU... gkZMGWA==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509Certificate>MIIDmjCCAoKg... Ct1jB0+UOw=</X509Certificate>
            </X509Data>
        </KeyInfo>
    </Signature>
</CertApplicationRequest>
```

4.4.4 Application response to certificate renewal request

```
<?xml version="1.0" encoding="UTF-8"?>
<xd:CertApplicationResponse xmlns:xd="http://op.fi/mlp/xmldata/">
    <xd:CustomerId>1000000047</xd:CustomerId>
    <xd:Timestamp>2010-01-26T14:32:51.808+02:00</xd:Timestamp>
    <xd:ResponseCode>00</xd:ResponseCode>
    <xd:ResponseText>OK.</xd:ResponseText>
    <xd:Certificates>
        <xd:Certificate>
            <xd:Name>CN=1000000047,C=FI</xd:Name>
            <xd:Certificate>MIICvTCCAa... Ne+0U19z3z25nFb</xd:Certificate>
            <xd:CertificateFormat>X509v3</xd:CertificateFormat>
        </xd:Certificate>
    </xd:Certificates>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
            20010315#WithComments"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <Reference URI="">
                <Transforms>
```

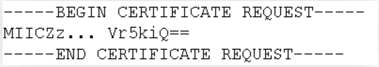
```
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
            signature"/>
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <DigestValue>ZdaOhjgcjfFb5aRwgMeWtlR5Oj0=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>PXPPXC... +TLjnO2g==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509Certificate>MIIDnDCCAo... A7xVA==</X509Certificate>
            </X509Data>
        </KeyInfo>
    </Signature>
</xd:CertApplicationResponse>
```

4.4.5 Certificate application request with a transfer key

```
<?xml version="1.0" encoding="UTF-8"?>
<CertApplicationRequest xmlns="http://op.fi/mlp/xmldata/">
    <CustomerId>1000010583</CustomerId>
    <Timestamp>2010-02-04T12:40:00.929+02:00</Timestamp>
    <Environment>TEST</Environment>
    <SoftwareId>software 1.01</SoftwareId>
    <Compression>false</Compression>
    <Service>MATU</Service>
    <Content>MIICZz... Vr5kiQ==</Content>
    <TransferKey>2251401483958635</TransferKey>
</CertApplicationRequest>
```

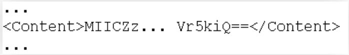
4.4.5.1 Detailed instructions on generation of 'getCertificate' request with transfer keys

a. The generation of a CSR (Certificate Signing Request) using a private and public keypair must look like this:



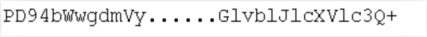
```
-----BEGIN CERTIFICATE REQUEST-----
MIICZz... Vr5kiQ==
-----END CERTIFICATE REQUEST-----
```

b. Start and End Tags from a generated CSR should be removed and the binary value should be inserted as shown in 4.4.5 Certificate Application request with a transfer key:



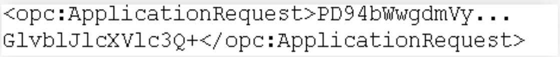
```
...
<Content>MIICZz... Vr5kiQ==</Content>
...
```

c. A Certificate Application request with a transfer key generated in accordance with section 4.4.5 should be base64 encoded to give the following values:



```
PD94bWwgdmVy......GlvblJlcXVlc3Q+
```

d. A Base64 encoded Application Request must be inserted into Request Message as shown in section 4.4.1.:



```
<opc:ApplicationRequest>PD94bWwgdmVy...
GlvblJlcXVlc3Q+</opc:ApplicationRequest>
```

e. The initial 'getCertificate' request using transfer keys is now ready to be sent to the bank for the downloading of a new certificate.

4.4.6 Application response to certificate application request with a transfer key

```
<?xml version="1.0" encoding="UTF-8"?>
<xd:CertApplicationResponse xmlns:xd="http://op.fi/mlp/xmldata/">
        <xd:CustomerId>1000010583</xd:CustomerId>
        <xd:Timestamp>2010-02-04T12:29:32.704+02:00</xd:Timestamp>
        <xd:ResponseCode>00</xd:ResponseCode>
        <xd:ResponseText>OK.</xd:ResponseText>
        <xd:Certificates>
                <xd:Certificate>
                        <xd:Name>CN=1000010583,C=FI</xd:Name>
                        <xd:Certificate>MIICvT... AssyGCD</xd:Certificate>
                        <xd:CertificateFormat>X509v3</xd:CertificateFormat>
                </xd:Certificate>
        </xd:Certificates>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
                <SignedInfo>
                        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
                        20010315#WithComments"/>
                        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                        <Reference URI="">
                                <Transforms>
                                        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
                        signature"/>
                                </Transforms>
                                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                                <DigestValue>pROjhxTaOs2FznVwOPhA7lbJYAE=</DigestValue>
                        </Reference>
                </SignedInfo>
                <SignatureValue>Kv0oDf... 9BU3Iw==</SignatureValue>
                <KeyInfo>
                        <X509Data>
                                <X509Certificate>MIIDn... xVA==</X509Certificate>
                        </X509Data>
                </KeyInfo>
        </Signature>
</xd:CertApplicationResponse>
```

4.4.7 Application request for certificate retrieval with a serial number

```
<?xml version="1.0" encoding="UTF-8"?>
<CertApplicationRequest xmlns="http://op.fi/mlp/xmldata/">
  <CustomerId>1000010583</CustomerId>
  <Timestamp>2010-02-04T12:53:55.325+02:00</Timestamp>
  <Environment>TEST</Environment>
  <SoftwareId>software 1.01</SoftwareId>
  <Compression>false</Compression>
  <Service>MATU</Service>
  <SerialNumber>442519889</SerialNumber>
</CertApplicationRequest>
```

4.4.8 Application response to certificate retrieval request with a serial number

```
<?xml version="1.0" encoding="UTF-8"?>
CertApplicationResponseDocument::<xd:CertApplicationResponse xmlns:xd="http://op.fi/mlp/xmldata/">
  <xd:CustomerId>1000010583</xd:CustomerId>
  <xd:Timestamp>2010-02-04T12:54:02.370+02:00</xd:Timestamp>
  <xd:ResponseCode>00</xd:ResponseCode>
  <xd:ResponseText>OK.</xd:ResponseText>
  <xd:Certificates>
    <xd:Certificate>
      <xd:Name>CN=1000010583,C=FI</xd:Name>
      <xd:Certificate>MIICvTC... AssyGCD</xd:Certificate>
      <xd:CertificateFormat>X509v3</xd:CertificateFormat>
    </xd:Certificate>
  </xd:Certificates>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
```

```
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
                  20010315#WithComments"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>fYSxDgACYGnJyt3R0Vg9aOLkdyk=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>O4vxL... n/th4DA==</SignatureValue>
    <KeyInfo>
      <X509Data>
        <X509Certificate>MIIDnD... 7xVA==</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
</xd:CertApplicationResponse>
```

4.4.9 Application request for retrieval of service certificates

```
<CertApplicationRequest xmlns="http://op.fi/mlp/xmldata/">
  <CustomerId>1000010522</CustomerId>
  <Timestamp>2010-02-04T12:59:35.727+02:00</Timestamp>
  <Environment>TEST</Environment>
  <SoftwareId>software 1.01</SoftwareId>
  <Service>MATU</Service>
</CertApplicationRequest>
```

4.4.10 Application response to retrieval request for service certificates

```
<?xml version="1.0" encoding="UTF-8"?>
<CertApplicationResponse xmlns="http://op.fi/mlp/xmldata/"xmlns:ns2=
"http://www.w3.org/2000/09/xmldsig#">
        <CustomerId>1000000047</CustomerId>
        <Timestamp>2018-0319T09:43:33.504+02:00</Timestamp>
        <ResponseCode>00</ResponseCode>
        ResponseText>OK.</ResponseText>
        <Certificates>
                <Certificate>
                        <Name>CN=CUSTOMER TEST OP-Pohjola Services CA, C=FI</Name>
                        <Certificate>MIIGIDCCBAigAwI...kVj8Sv1dNBrnd52LISFjx2wCXud</Certificate>
                        <CertificateFormat>X509v3</CertificateFormat>
                </Certificate>
                <Certificate>
                        <Name>CN=CUSTOMER TEST OP-Pohjola WS CA, C=FI</Name>
                        <Certificate>MIIGGjCCBAKgAwIBAgIDAT5bMA0G...dMwP+ujyr/EoHCNOrGcpAs</Certi
                        ficate>
                        <CertificateFormat>X509v3</CertificateFormat>
                </Certificate>
                <Certificate>
                        <Name>C=FI, CN=CUSTOMER TEST OP Services CA V2</Name>

                                <Certificate>MIIGGzCCBAOgAwIBAgIDKCJGMA0GCSqGS...3U+YS9431RzBqGk48uE5
                        KSxAcUZ
                        vLnc6372j0a7WsISQ==</Certificate>
                        <CertificateFormat>X509v3</CertificateFormat>
                </Certificate>
                <Certificate>
                        <Name>C=FI, CN=CUSTOMER TEST OP WS CA V2</Name>
                        <Certificate>MIIGFTCCA/2gAwIBAgIDKBo1M...tkoEmxWW1K8rootLAROAf+a
                        2K13wgSwOA==</Certificate>
                        <CertificateFormat>X509v3</CertificateFormat>
                </Certificate>
        </Certificates>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
            20010315#WithComments"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <Reference URI="">
                <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
            signature"/>
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <DigestValue>VyXRntiU4/X/h1GOGj0Tjtt7wlc=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>RR5AfAz0Rt7NPUQnnTJA0IuRUtZ9cQUIZRq0DN....sp
        ViIxA==</SignatureValue>
        <KeyInfo>
            <X509Data>
                <X509Certificate>MIIGKjCCBBKgA...HsHt8Os4G7ov7mhKYQ==</X509Certificate>
            </X509Data>
        </KeyInfo>
    </Signature>
</CertApplicationResponse>
```