Technical documentation
Pre-approved payments

Version: 1.0
Date: 15.09.2020
Issuer: SpareBank 1

Table of content

Revision history

Version	Name	Date	Description
1.0	SB1	15.09.2020	• New outline of the document, including change to chapter numbers. • On-boarding information added. • The API specification has been moved to this document. • Clarification made in chapter 1.3 • Appendix (3) added on ASiCManifest.
0.91	SB1	27.08.2020	Appendix 2 and 4: Added a missing field MessageIdentification to the format and example.
0.9	SB1	20.08.2020	Chapter 1.9: table of file types has been removed. (The only fil type to be sent in the ASiC container is PAIN001.) Appendix 2: - Added new DateTime field with the time of the Authentication in the file format. - updated **OrganisationIdentification** where two fields were missing. Appendix 4: Added DateTime in the example.
0.8	SpareBank 1 (SB1)	03.07.2020	First version

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to give a technical description of the solution for pre-approved payments and the requirements for setting up and using the solution by a Corporate Payment Initiator (CPI).

1.2 Target group

The target group for this document is CPIs (e.g. ERP vendors or corporates) who wants to send pre-approved payment files to SpareBank 1.

1.3 Scope, assumptions and limitations

The pre-approved payment file solution has the following scope:

- Receiving and handling of payment files (pain.001) and authorization files sent from ERP-system within an ASiC container.
- Other file types should be sent as before (without ASiC).
    - It is also possible to send cancellation files (camt.055) within an ASiC. In this case any approval file will be disregarded.
- Return files, e.g. pain.002 and camt.054, will *not* be sent by ASiC.
- Only approval data files with social security numbers will be supported
- If authorization fails for a payment, only this payment will be rejected, not the entire file.

1.4 Technical support

For any questions or support on this solution please contact erp@sparebank1.no.

2. HOW TO GET STARTED

This chapter describes the on-boarding process for the pre-approved payment files solution.

1. Fill in the on-boarding information and send it to erp@sparebank1.no. SB1 will then arrange for the necessary information to get to our service partner (TietoEVRY) and will return with an implementation guide where needed. A plan for test and go-live will also be established as part of the on-boarding process.

2. Sign the necessary product agreement. ("Avtale om tilknytning til sparebank1s betalingsinfrastruktur")

3. Implement a solution for strong authentication or signing by verified approvers according to the security requirements. ("Sikkerhetskrav - tilknytning betalingsinfrastruktur")
   - Contact SpareBank 1 before implementation to ensure a proper security level in the approval process and solution. This is done by a pre-approved payments revision process.

4. Implement the ASiC solution for packaging of payment files and approval information according to this guide.

5. (Optional) Implement the API function to check that a person has necessary authorization to perform a payment.

6. Implement storage of an audit trail of users approving payments. This information can later be requested by SpareBank 1 for use in any customer dispute.

7. Implement a way for a customer to choose whether to use pre-approved payments or not. This is to ensure that customers who want to make use of manual approval can continue to do so.

3. DESCRIPTION OF THE ASIC-CONTAINER

3.1 About SFTP with pre-approved payment files (ASiCe)

This document defines how ERP (or other third parties to bank customers) can deliver messages with pre-approval to SpareBank 1 via our service partner TietoEVRY. The service supports bank messages in (ISO20022) format and can be delivered with or without pre-approval (ASiCe).

3.2 Overview of document flow

The picture below shows how the service connects customers, ERP (or other third parties) and TietoEVRY.

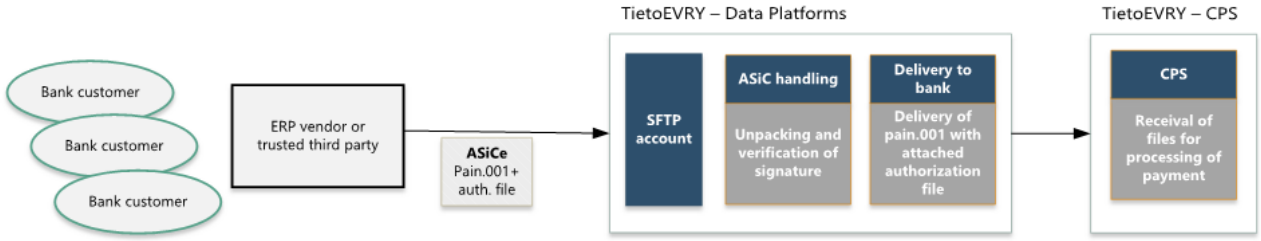


FIGURE 1 – DOCUMENT FLOW

3.3 Connecting to the service

The standard interface for connecting to our service is SFTP. The ERP/third party receives a user ID and a password to establish SFTP connection. The active party (ERP/third party or TietoEVRY) sends the data.

The service is available 24/7. Sending and receiving files is an automated process at TietoEVRY.
The file is considered received when TietoEVRYs SFTP server has announced that the file is received and transferred to the correct catalogue with a correct file name.

If the customer or TietoEVRY fails in the file transfer, new attempts will be made. TietoEVRY and the customer should check the catalogue for files on a regular basis. The customer and TietoEVRY should only send files according to a mutual agreement.

3.4 File and Format support

The solution supports ISO20022 formats that are either delivered with or without an ASiC container.
It is required that pre-approved payment files are delivered in a signed ASiC container with metadata, according to the definition in this document.

The ASiC container must have the file extension .asice.

3.5 Approval of payments in ERP system

The end user will approve payments in the ERP system.

The authentication or signing method used in the ERP system must be a SpareBank 1 approved two-factor mechanism. This will typically be a two-factor mobile authentication solution or a solution based on PKI – e.g. BankID authentication or signing.

The ERP system must store an audit trail that describes the approval performed by the user. This "package" will typically consist of log entries, results from authentication/signing and other context information that can be used as proof of user approval in a possible customer dispute. A reference to this package is put into the approver data file (in the field "AuthenticationReference") that is accompanying the payment file sent to the bank.

3.6 Creating the file package (ASiC file)

An Associated Signature Container (ASiC) will be used to package payment files and approval data to be sent to SpareBank 1.ASiC is basically a compressed (zip) file, that contains the data files, together with a standard ASiC manifest file and a digital signature (PKCS#7) that ensures that no forgery or tampering of the ASiC content can happen.

The digital signatures will be created using the private key in an enterprise signature.
The structure of the ASiC and the process of creating it is described in figure 1:

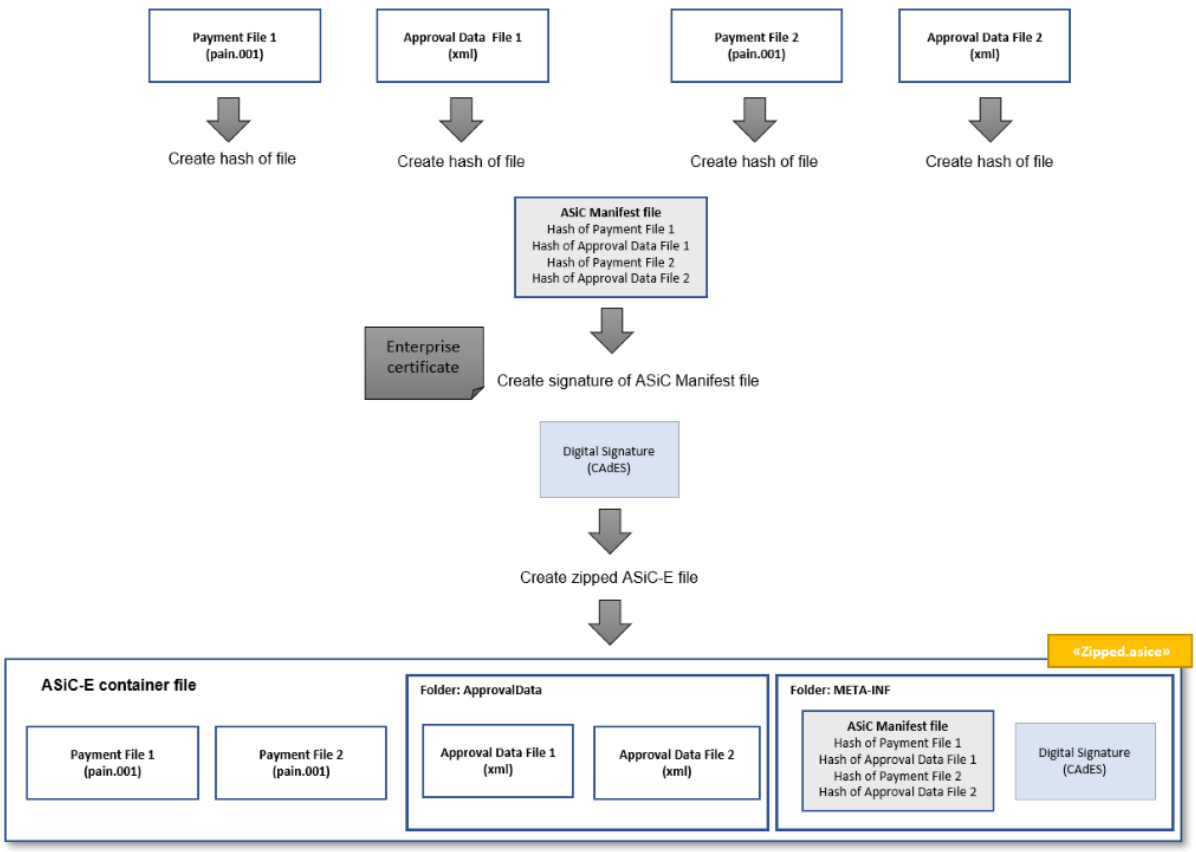


FIGURE 2 - STRUCTURE OF ASIC

3.7 File size

The maximum file size for documents distributed is 100 MB, unless otherwise explicitly agreed. Larger files must be divided into smaller files that do not exceed the limit specified.

3.8 References

ETSI TS 102 918 V1.3.1 - Associated Signature Containers
(ASiC): *https://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.03.01_60/ts_102918v010301p.pdf*

3.9 File naming standard

The file name is important for the service, and the name standard must be mutually agreed. The standard states that the file name must be unique, and that the agreed filename indicates which process the file should activate. It is not allowed to reuse old filenames and it is not allowed to use shift state as an only difference on the file name.
Recommended signs in the file name are letters a-z, A-Z, numbers, dot, hyphen and underscore. This is also recommended for file names inside a ZIP file. Note that space is not an allowed letter. The file name should not be longer than 100 signs.

File naming of the ASiC file must follow the following standard:
<status>_<ProviderId>_<unique>.asice
Where <status> is T for test or P for prod
<ProviderId> is the Id of the sender ("Tjenesteleverandør" TL) given by EVRY
<uniq> is a uniq part. Extension must be .asice

File naming of payment files of ASiC must use the following standard:

P.NOxxx.20180611154827-1R.asice

	Description
P	Indicator for prod, eller test. (P el. T)
xxxxx	The service providers internal ID in TietoEVRY
20180611154827-1R	Unique serial number (with timestamp) YYYYMMDDhhmmss-UnikId
.ascie	**Always use extension .ascie**

Service Provider

- Contact TietoEVRY to get the internal ID that you will use in the exchanges

Serial number

- Files sent to TietoEVRY must have unique serial number during a period of at least 24 hours
- Files sent from TietoEVRY will have unique serial number in the format: YYYYMMDDhhmmss-UnikId

File naming of ISO payment files within ASiC must use the following standard:
P.xxxxxxxxxx.0000.P001.20180611154827-1R.xxxx.Dxxx.DAT
P.xxxxxxxxxx.0000.C054.20180611154827-1R.xxxx.DAT

	Description
P	**Indicator for prod, or test (P el. T)**
xxxxxxxxxx	**The customers organization number in the bank. If 9 digits, 2 add 2 zeroes at start**
0000	**Bank id**
P001/C55	**Message type**
20180611154827-1R	**Unique serial number (with timestamp) YYYYMMDDhhmmss-UnikId**
xxxx	**The service providers internal ID in TietoEVRY**
Dxxx	**Divison number (optional)**
DAT	**Aways use extension .DAT**

Bank id

- Send bank id for which banks this service shall be available for to TietoEVRY, D-DS, for specifying which bank id to be used in the exchange

Serial number

- Files sent to TietoEVRY must have unique serial number during a period of at least 24 hours
- Files sent from TietoEVRY will have unique serial number in the format: YYYYMMDDhhmmss-UnikId

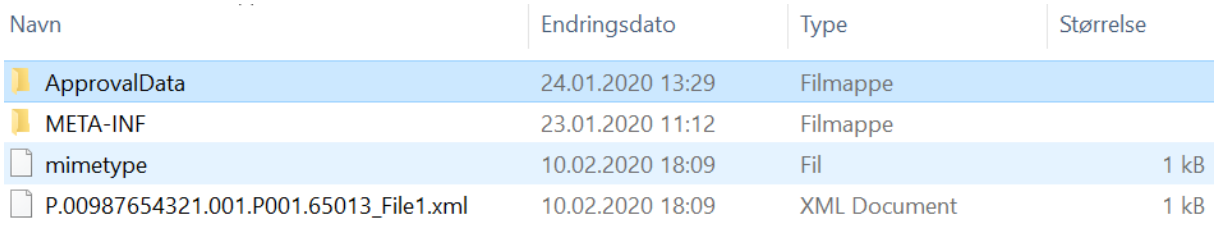
Service Provider

- Contact TietoEVRY to get the internal ID that you will use in the exchanges, should be the same as used in the asice

4. APPENDIX 1: ASIC CONTAINER STRUCTURE

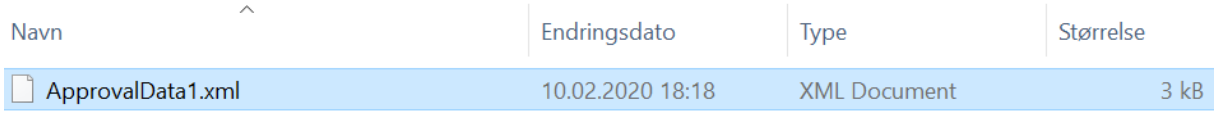
To sign an ISO20022 without touching the file itself, the use of ASIC containers has been advised.

ASIC structure:
META-INF/ASiCManifest.xml
META-INF/signature.p7f
ApprovalData/AnyFileName (e.g. ApprovalData*.xml)
mimetype
[T|P].<clientnumber>.<division number>.<filecode>.<your choice>.xml
(e.g. P.00987654321.001.P001.65013_File1.xml)
NB! Use Naming convention for payment, specified in Appendix 1 Fil Naming, for files inside the ASiC Container.
ASiC Container (root folder):



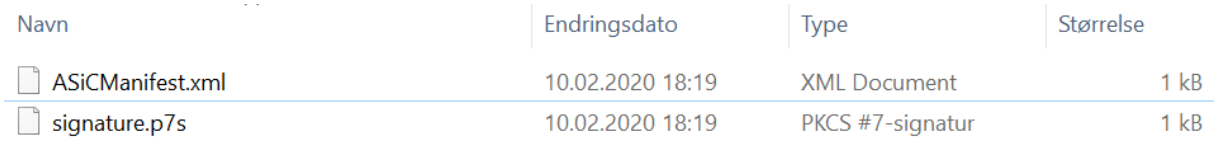
Navn	Endringsdato	Type	Størrelse
ApprovalData	24.01.2020 13:29	Filmappe	
META-INF	23.01.2020 11:12	Filmappe	
mimetype	10.02.2020 18:09	Fil	1 kB
P.00987654321.001.P001.65013_File1.xml	10.02.2020 18:09	XML Document	1 kB

ApprovalData folder:



Navn	Endringsdato	Type	Størrelse
ApprovalData1.xml	10.02.2020 18:18	XML Document	3 kB

META-INF folder:



Navn	Endringsdato	Type	Størrelse
ASiCManifest.xml	10.02.2020 18:19	XML Document	1 kB
signature.p7s	10.02.2020 18:19	PKCS #7-signatur	1 kB

5. APPENDIX 2: FIELD DESCRIPTION AUTHORIZATION FILE

Below is a description of the fields in the ApprovalData xml file:

Or	Tag	Level	Mult	Type	Comments
	<MessageReference format=”ISO20022”>	1	1	String	Reference to file in ASiC-E container.
	<MessageIdentification>	1	1	String	Reference as assigned by the instruction part. XML ISO 20022 = <MsgId>. Telepay = Sequence Number from startrecord (BETFOR00)
	<InitiatingParty>	1	1	Object	Party that initiates the ASiC container
	<Identification>	2	1	Object	
	<OrganisationIdentification>	3	1	Object	See "OrganisationIdentification"
	<PaymentInformation>	1	1..n	Object	
	<Debtor>	2	0..1	Object	
	<Identification>	3	0..1	Object	
	<OrganisationIdentification>	4	0..1	Object	See "OrganisationIdentification"
	<DebtorAccount>	2	1..n	Object	
	<Identification>	3	1	Object	
	<Other>	4	1	Object	
	<Identification>	5	1	Object	
	<SchemeName>	5	1	Object	
	<Code>	6	1	String	Used codes: • BBAN
	<PaymentTypeInformation>	2	1	Object	
	<CategoryPurpose>	3	1	Object	
	<Code>	4	1	String	Used codes: • OTHR • SALA

	<ApproverIdentification>	2	1..2	Object	First or First and second approver
	<Identification>	3	1	Object	If Code = SOSE then SSN of user. If Code = BANKID_SDO then file name reference.
	<PrivateIdentification>	4	1	Object	
	<Other>	5	1	Object	
	<Identification>	6	1	String	SSN or file name reference
	<SchemeName>	6	1	Object	
{Or	<Code>	7	1	String	Used codes: • SOSE • BANKID_SDO
Or}	<Proprietary>	7	1	String	NORWEGIAN_BANKID
	<AuthenticationInformation>	3	1	Object	
	<AuthenticationMethodVendor>	4	1	Object	
	<Name>	5	1	String	ERP name, BANKID (Vipps)
	<AuthenticationMethod>	4	1	Object	
	<Name>	5	1	String	ERP method, example ERP_2FA_mobile, BANKID_MOBILE
	<AuthenticationReference>	4	1	String	Reference for authentication
	<AuthenticationDateTime>	4	1	String	Reference for date/time

OrganisationIdentification

Tag	Level	Mult	Type	Comments
<Other>	1	0..n	Object	
<Identification>	2	1	String	
<SchemeName>	2	1	String	
<Code>	3	1	String	Type of identification for the initiating party, example CUST or BANK

6. APPENDIX 3: ASICMANIFEST

Structure of the ASiCManifest.

Message Item	Tag Name	Level	Comments
ASiCManifest	<ASiCManifest>	1	
Signature Reference	<SigReference>	2	URI="META-INF/signature.p7s" MimeType="application/x-pkcs7-signature"
Data Object Reference	<DataObjectReference>	2	
Digest Method	<DigestMethod>	3	Digest method algorithm
Digest Value	<DigestValue>	3	DigestValue contains the Base64 encoded result of applying the hash algorithm to the transformed resource(s) defined in the Data Object Referance element attributes.

Example of ASiC manifest file

```
<?xml version="1.0" encoding="utf-8"?>
<ASiCManifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://uri.etsi.org/02918/v1.2.1#"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
  <SigReference URI="META-INF/signature.p7s" MimeType="application/x-pkcs7-signature" />
  <DataObjectReference
URI="T.00xxxxxxxx.1801.P001.20200625121640_7b2957ea7ab8472990f7b24a6fe1ab72.NODLD.DAT"
MimeType="application/xml">
    <ns2:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ns2:DigestValue>uQ0fKJ35gqRWWnnoUsUthBamKE3QZ7oIKWjyKGCD/OA=</ns2:DigestValue>
  </DataObjectReference>
  <DataObjectReference
URI="ApprovalData\ApprovalData_T.00xxxxxxxx.1801.P001.20200625121640_7b2957ea7ab8472990f7b2
4a6fe1ab72.NODLD.DAT.xml" MimeType="application/xml">
    <ns2:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ns2:DigestValue>1twMh/BcfxkRWCFS4hjJJBKTJULpo7NDeGuh+elRLMo=</ns2:DigestValue>
  </DataObjectReference>
</ASiCManifest>
```

7. APPENDIX 4: EXAMPLE OF AUTHORIZATION FILE

```
<Document xsi:schemaLocation="urn:std:cps:xxxx:tech:xsd:yyyyapprovaldata.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:std:cps:xxxx:tech:xsd:yyyy">
  <MessageReference format="ISO20022"> pain001_paymentfile1.xml</MessageReference>
  <MessageIdentification>ee82a880dfc649248d306dcde5d1bb9a</MessageIdentification>
  <InitiatingParty>
    <Identification>
      <OrganisationIdentification>
        <Other>
          <Identification>987654321</Identification>
          <SchemeName>
            <Code>CUST</Code>
          </SchemeName>
        </Other>
        <Other>
          <Identification>MYDIV1</Identification>
          <SchemeName>
            <Code>BANK</Code>
          </SchemeName>
        </Other>
      </OrganisationIdentification>
    </Identification>
  </InitiatingParty>
  <PaymentInformation>
    <Debtor>
      <Identification>
        <OrganisationIdentification>
          <Other>
            <Identification>987654321</Identification>
            <SchemeName>
              <Code>CUST</Code>
            </SchemeName>
          </Other>
          <Other>
            <Identification>MYDIV2</Identification>
            <SchemeName>
              <Code>BANK</Code>
            </SchemeName>
          </Other>
        </OrganisationIdentification>
      </Identification>
    </Debtor>
    <DebtorAccount>
      <Identification>
        <Other>
          <Identification>96870529195</Identification>
          <SchemeName>
            <Code>BBAN</Code>
          </SchemeName>
        </Other>
      </Identification>
    </DebtorAccount>
    <PaymentTypeInformation>
```

```
        <CategoryPurpose>
          <Code>OTHR</Code> <!-- OPT: OTHR/SALA -->
        </CategoryPurpose>
      </PaymentTypeInformation>
      <ApproverIdentification>
        <Identification>
          <PrivateIdentification>
            <Other>
              <Identification>17018734591</Identification>
              <SchemeName>
                <Code>SOSE</Code>
              </SchemeName>
            </Other>
          </PrivateIdentification>
        </Identification>
        <AuthenticationInformation>
          <AuthenticationMethodVendor>
            <Name>ERP</Name>
          </AuthenticationMethodVendor>
          <AuthenticationMethod>
            <Name>ERP_2FA_MOBILE</Name>
          </AuthenticationMethod>
          <AuthenticationReference>1234-121212-1213</AuthenticationReference>
          <AuthenticationDateTime>2019-02-14T16:43:59</AuthenticationDateTime>
        </AuthenticationInformation>
      </ApproverIdentification>
    </PaymentInformation>
```

NB! Highlighted fields will occur twice if the debit account has double approval.

8. APPENDIX 5: EXAMPLE OF RETURN FILE

In general, there will be two cases of return files:

1. The ASiC container has error, but we can see the numbers of pain.001
2. No error in ASiC container, but some of the authorizations are rejected in the bank (CPS)

1. The ASiC container has error, but we can see the numbers of pain.001

pain.002 will be presented this way:

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.04">
 <CstmrPmtStsRpt>
  <GrpHdr>
   <MsgId>488679533.SFT.20200617203602170p01</MsgId>
   <CreDtTm>2020-06-17T20:36:10.394+02:00</CreDtTm>
  </GrpHdr>
  <OrgnlGrpInfAndSts>
   <OrgnlMsgId>0.MSG-20200617203533965</OrgnlMsgId>
   <OrgnlMsgNmId>pain.001.001.04</OrgnlMsgNmId>
   <GrpSts>RJCT</GrpSts>
   <StsRsnInf>
    <Rsn>
     <Cd>FF01</Cd>  <!-- File Format incomplete or invalid. -->
    </Rsn>
   </StsRsnInf>
   <StsRsnInf>
    <Rsn>
     <Cd>FF02</Cd>  <!-- Syntax error reason is provided as narrative information in the additional reason
information. -->
    </Rsn>
    <AddtlInf>**************************************************</AddtlInf>
    <AddtlInf>* Generic text describing the error situation   *</AddtlInf>
    <AddtlInf>* as precise as possible                        *</AddtlInf>
    <AddtlInf>**************************************************</AddtlInf>
   </StsRsnInf>
  </OrgnlGrpInfAndSts>
 </CstmrPmtStsRpt>
```

2. No error in ASiC container, but some of the authorizations are rejected in the bank (CPS)

Return code for rejected payments is AG08 with an explanation of reason in AddtlInf:

- <AddtlInf>Only one approver [id] accepted for account that requires two approvers<AddtlInf>
- <AddtlInf>Approver [id] not validated because AML-status not confirmed OK<AddtlInf>
- <AddtlInf>Approver [id] not validated because not granted approval right to debit account [id]<AddtlInf>

Example of pain.002 from a pain.001 with three PmtInf where the authorization for one fails (highlighted), and two PmtInf is OK.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
```

```
<CstmrPmtStsRpt>
  <GrpHdr>
    <MsgId>423316.400662.124677164.3</MsgId> <!-- Original value from PIN [423316.400662.124677164] -->
    <CreDtTm>2020-06-29T11:37:08.463+02:00</CreDtTm>
    <InitgPty>
      <Nm>Navn</Nm>
      <Id>
        <OrgId>
          <BICOrBEI>SPRONO22XXX</BICOrBEI>
          <Othr>
            <Id>9999999999</Id>
            <SchmeNm>
              <Cd>CUST</Cd>
            </SchmeNm>
          </Othr>
        </OrgId>
      </Id>
    </InitgPty>
  </GrpHdr>
  <OrgnlGrpInfAndSts>
    <OrgnlMsgId>9B0B4168DB744785AE762196E8F3BFE2</OrgnlMsgId>
    <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
  </OrgnlGrpInfAndSts>
  <OrgnlPmtInfAndSts>
    <OrgnlPmtInfId>BAA80DFDDA164913902F3EA8C3339D18</OrgnlPmtInfId>
    <TxInfAndSts>
      <OrgnlInstrId>0A97B1162B6E4DA2A4B140B9CBB08F88</OrgnlInstrId>
      <OrgnlEndToEndId>0A97B1162B6E4DA2A4B140B9CBB08F88</OrgnlEndToEndId>
      <TxSts>ACCP</TxSts>
      <StsRsnInf>
        <AddtlInf>181729023</AddtlInf>
      </StsRsnInf>
    </TxInfAndSts>
  </OrgnlPmtInfAndSts>
  <OrgnlPmtInfAndSts>
    <OrgnlPmtInfId>E85C49EEF4A94E99A20CE54CC27DB78E</OrgnlPmtInfId>
    <StsRsnInf>
      <Rsn>
        <Cd>AG08</Cd> <!-- Transaction failed due to invalid or missing user or access right. -->
      </Rsn>
      <AddtlInf>Generic text specifying why the authorization was rejected</AddtlInf>
    </StsRsnInf>
    <TxInfAndSts>
      <OrgnlInstrId>919110210211462D8B1933E4A915CAAB</OrgnlInstrId>
      <OrgnlEndToEndId>919110210211462D8B1933E4A915CAAB</OrgnlEndToEndId>
      <TxSts>RJCT</TxSts>
      <StsRsnInf>
        <AddtlInf>181729024</AddtlInf>
      </StsRsnInf>
    </TxInfAndSts>
  </OrgnlPmtInfAndSts>
  <OrgnlPmtInfAndSts>
    <OrgnlPmtInfId>BAA80DFDDA164913902F3Exxxxxxxx</OrgnlPmtInfId>
    <TxInfAndSts>
      <OrgnlInstrId>0A97B1162B6E4DA2A4B140B9Cxxxxxxxx</OrgnlInstrId>
      <OrgnlEndToEndId>0A97B1162B6E4DA2A4B140B9xxxxxxxx</OrgnlEndToEndId>
      <TxSts>ACCP</TxSts>
      <StsRsnInf>
        <AddtlInf>181729023</AddtlInf>
```

```
            </StsRsnInf>
          </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>
```

9. APPENDIX 6: AUTHORIZATION API

The API enables the CPI to check that a user has the necessary authorisation to perform a payment, according to access rights defined in CPS. The API also checks the AML status of the user.

9.1 Description of the API

Extract of data that should be part of the input to the API (AuthorisationRequest):

- publicID (SSN)
- accountList

Extract of data that will be returned by the API (AuthorisationResponse):

AuthorisationResponse:

- publicID (SSN)
- amlStatus (OK, NOT_VALIDATED, FAILED_NAME_CHECK)
- accountNumber
- authorisationStatus1 (OK, not OK)
- organisationId
- agreementIntId (for internal use in TE)
- customerIntId (for internal use in TE)
- numberOfApprovers (1,2)

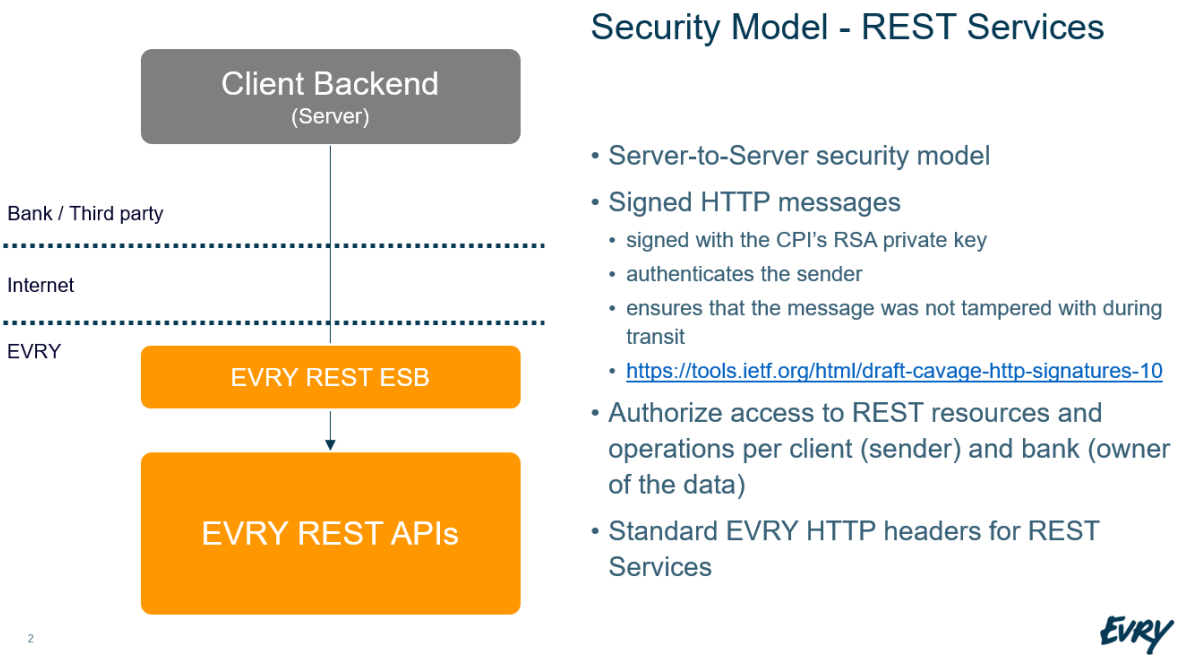
For a more detailed description of the API, please view the following attachments:

1. CRSAccountAuthorisationAPI_v_0.1.pdf
2. CRSAccountAuthorisationAPI_v_0.1.yaml

9.2 Access to the API

The security model for accessing TietoEVRY REST Services from third parties is based on signed HTTP messages according to the following standard: https://tools.ietf.org/html/draft-cavage-http-signatures-10

The security model is the same security model used by Application level signing of the PSD2 API, based on Berlin Group.



Minimum set HTTP of Headers (REST):

Header	Mandatory	Description
Signature	M	Used in accordance with https://datatracker.ietf.org/doc/draft-cavage-http-signatures/
Digest	M	Used in accordance with https://datatracker.ietf.org/doc/draft-cavage-http-signatures/ **Digest is mandatory for all requests containing body.**
Date	M	Standard HTTP Date Header
X-EVRY-DATAOWNERORGID	M	BankRegNum/BankID, 4 digits
X-EVRY-CLIENT-CLIENTNAME	M	Unique name identifying the requesting front or back-end application. For b2b, the certificates must be registered in the ESB on this name
X-EVRY-CLIENT-REQUESTID	M	Unique identification of request (e.g. used as log reference)
X-EVRY-CLIENT-ISMOBILE	O	If request origins from a mobile device, this is set to TRUE, else FALSE or not present
X-EVRY-USERID	O	The id of the user issuing this request if any

9.3 API certificate

The user of the service must provide TietoEVRY with an enterprise certificate containing the RSA-2048 bit public key that EVRY will use to validate received requests and associated key ID according to the specification.

The CPI can choose any issuer of the certificate, however, we strongly recommend a Qualified Certificate of Electronic Seals (QC eSeal) or an enterprise certificate.

The key must be RSA-2048 bit.

10. APPENDIX 7: ON-BOARDING INFORMATION

10.1 Signing up for the ASiC solution

1. Fill in the table below and send it to erp@sparebank1.no.

#	Question	Answer	Comment
1	Will you use the ASiC solution? (Y/N)		
2	Will you use the API to verify approvers? (Y/N)		
3	When do you plan to start testing? (Date)		
4	When will you be ready to go live with the solution? (Date)		

2. An SFTP account must be ordered if you do not already have one. To order an SFTP account please contact erp@sparebank1.no.
   In the SFTP account, use separate folders for test and production.

3. Acquire a valid enterprise certificate from a trusted issuer, e.g. Buypass or Commfides.
   Note that you will need separate certificates for test and production.

10.2 Access to the API

Fill in the table below to get access to the API and send it to erp@sparebank1.no.

SB1 will arrange for the necessary information to get to our service partner (TietoEVRY) and will return an implementation guide with further details.
Note that there will be a need for separate certificates for test and production so if ordering both, please make two copies of the table.

A copy of the certificate must also be provided as an attachment using a .pem format.

Required Information	To be filled in
Name of Company	
Company id	
Contact info (technical)	
Test or production	
New / renewal	
In case of renewal: Current ClientName (X-EVRY-CLIENT-CLIENTNAME) Current KeyID Date for removal of current certificate	