Contents

[1]

DeNederlandscheBank

EUROSYSTEEM

---

[1] Version 1.2.2, dated 2018-07-24

1 Delivery of data to DNB using Logius Digipoort

1.1 Introduction

DNB has opted for Logius as its partner for exchange of electronic messages. Logius is the organization officially appointed by the Dutch government for exchanging messages between businesses and governmental organizations. To successfully transmit messages to DNB, Logius has stipulated a number of requirements which are validated upon receipt of any message. After approval the message is relayed to the stipulated receiver which has been explicitly stated in the message header by the sender.
After DNB receives any message, a number of technical validations will be performed on the message as a whole, assuring the message is intended for DNB and the message complies with our regulations. After the delivery has passed technical validation the business data will be loaded into DNB databases and business validations will be executed. The log generated will be made available on the DLR (DNB portal) where the status of the transmitted message can be found.
If the first set of validations results in a technical error, the data will not be evaluated, the message is rejected and the reporting agent has submit a corrected message with Logius. The log on these errors will be available in the DLR, effectively both results will end up in the same log but if there is a technical error there will be no business evaluation.

1.2 Logius documentation

Logius general website: link (nl), link (en).
Logius specific documentation 'Digipoort Grote Berichten FTP 3.0' service: link (nl).
Logius XML schema for the message header is included in the documentation above.

1.3 Logius requirements

Every reporting agent will need a PKI-o certificate. These can be obtained from trusted service providers (TSP´s) that have been authorized by the Dutch government. A current list of acknowledged TSP's is published by Logius on their website.
Every reporting agent will also need a subscription with Logius. A subscription is recipient specific. Effectively there will be a folder on Logius' systems per recipient for every subscriber. If this is your first subscription you will receive a userid/pw for the (S)FTP connection.
Every message transmitted using the Logius service 'Digipoort Grote berichten FTP 3.0', needs to have its header set up in the correct way as specified in the Logius documentation. Only the XML node <aanleverkenmerk> in the header will have recipient (i.e. DNB) specific requirements.
The file itself is an uncompressed MIME multipart file[2] consisting of the Logius message header (XML) and a single ZIP file as its payload.

---

[2] IETF specification RFC2045: https://tools.ietf.org/html/rfc2045

1.3.1 Logius validations

After receiving of a message, Logius will validate:

- PKI-o validity;
- Subscription;
- XML validity of the message header;
- Existence of the recipient.

If any of these validations results in an error, the message will be rejected. If not, the message will be forwarded.

1.3.2 Logius particulars

Timestamps returned by Logius to the sender are in a different timezone than NL. The timestamps are in 'Zulu' time which is stated at the end of the full timestamp: 2018-02-09T02:39:21.842**Z.** Reporters that rely on a sequence in timestamps from their transmission and the response(s) from Logius will need to calculate the appropriate time.

1.4 DNB requirements

This section specifies the DNB requirements on the Logius message header specific node <aanleverkenmerk> reserved for recipients, the file naming and any payload requirement. Since some of the details used are message type specific, all DDA's[3] have a table specifying the message type dependent variables. This table looks like:

Variable	Value(s) to be used	Options
Logius issued message name		DNB_rapportages[4]
Reporter identifier		CRM, KvK, LEI, MDM, RIAD, RSIN
Data Delivery Code		
Data Delivery Agreement code		
Report Reference Date		Format: CCYY-MM-DD
Hashing method		Mandatory SHA-256 or Prohibited
Encryption method		AES
Data file types		CSV, PDF, JSON, XML, XBRL, SDMX

Data file types are always in UTF-8.

1.4.1 Logius XML header

The Logius documentation will guide you in setting up the Logius XML header in their message 'AanleverRequest'. DNB requires the XML header node <aanleverkenmerk> to contain:

- the ID of the reporting agent (example: NL001),
- optional[5]: the report reference date (example 2019-01-01) and
- the identifier of the report being transmitted; i.e. the DataDeliveryCode.

---

[3] DDA: Data Delivery Agreement (nl: Gegevensleveringsovereenkomst, GLO)

[4] For all deliveries the term DNB_rapportages has to be used as of April 18th 2019, until this date AnaCredit has to be used, even when the delivery DOES NOT concern AnaCredit (E.g. delivery for the Deposit Guarantee Scheme).

[5] For reporting obligations the report reference date is mandatory, for non-obligation deliveries this field is prohibited.

As these three data components exceed the allowed length of the node, their content is being shortened by applying the SHA-1 protocol. This has nothing to do with security, the protocol is used only to be able to fit the required data into the node Logius makes available.

The data components are semicolon separated (including an end semicolon) and can contain no spaces on the individual values. Letters are in upper case.

Depending on the type of report required, the reporter ID can be any RIAD, LEI, MDM or other code.

The report reference date notation is ISO-8601 compliant CCYY-MM-DD and only required if the report has reporting obligations for stipulated periods.

The DataDeliveryCode is set by each individual report type and is 14-18 positions uppercase.

Example (with reporting obligation):
NL001;2019-01-01;ZGRACRKANAXXXX;
Will result in SHA-1 value: B86EE31886470961615C0E28037942DD9EDA58BB
(upper/lower case in the SHA-1 value is irrelevant).

Example (without reporting obligation):
NL001;ZGRACRKANAXXXX;
Will result in SHA-1 value: 0B0F9F3BCBFC1A332E2B907397029CCFC0AFDED6

The result of this procedure is a XML file that will be the header part of the MIME file required by Logius.

1.4.2 Metadata file accompanying the data files

The container needs to be created by using a XML (meta) data file that describes the content of the container and the number of files required by the reporting obligation. In most cases this will be mean any number of CSV and or XML and or PDF files.

The metadata XML file is created according to its schema: dnbmeta.xsd in namespace https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01. Each of the nodes is specified in the schema.

The namespace prefix used to qualify the local names of the elements of the metadata file is ‘dnbmeta’.

Special attention is needed for the @attachment node. This Boolean is used to indicate whether a certain file in the compressed container is an attachment to any of the reported data files. This indicator is only used (set to ‘true’) for *voluntary* attachments. If such a voluntary file is allowed this will be indicated in the obligation on the DLR and made explicit in the DDA.

The name of the metadata XML file must be: dnbmetadata.xml

1.4.3 Hashing of individual data files

All files other than the dnbmetadata.xml may be hashed for checking their integrity upon reception. The hashing method is report type dependent the default being SHA-256. The file name and its hash are recorded in the dnbmetadata.xml file in no particular order.

1.4.4 Compressed container

The dnbmetadata.xml file and the data files themselves need to be compressed using (win)zip.
The compressed container may contain no folder structure and/or other *.zip files.
The name of the compressed container is at the discretion of the reporting agent if encryption is required but is restricted to the pattern [a-zA-Z0-9_-] (numbers and letters, underscore and hyphen).
If no encryption is required, the name of the zip container is a specified in chapter 1.4.8.

1.4.5 Option: encrypting the compressed container

Depending on the confidentiality of the submitted data, encryption of the data may be required. If the compressed container needs to be encrypted this will be indicated in the report type specific table (above), the default being AES-256. When encryption is required the reporter must generate an appropriate key and encrypt the ZIP container. The symmetric key and its IV value for decrypting by DNB must be stored in a text file named 'sleutel.xml' with no other content, based on the dnbencryption.xsd schema with namespace https://statistiek.dnb.nl/schemas/xml/a2aenc/2018-03-01 using *no namespace prefix*, and using http://www.w3.org/TR/2001/REC-xml-c14n-20010315 as the canocalization method because of the limited size. Be aware that the key length for AES-256 will be depending on the length of the key used for encrypting. Since the sleutel.xml accepts only base64 encoded values these lengths will be increased by a number of characters resulting in a length according to the formula: ceiling(4n/3;1), in which n is the length of your key in bytes and 1 is the significance of the ceiling function. Be aware that most encryption software generates key and IV values in HEX code whilst the sleutel.xml file expects base64 encoded strings.
The name of the encrypted container is derived from the compressed container, adding the extension '.pgp' which *does not* mean that the openpgp protocol is used.

Supported encryption methods

Name	Keysize	Blocksize	Mode	Padding
AES-256	256	128	CBC	PKSC#7
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]		
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	

1.4.6 Encrypting the reporter issued key used to encrypt the data

With the public X.509 key the 'sleutel.xml' file is encrypted (with RSA-4096 and OAEP-SHA-256 as padding)[6].
This results in a file named sleutel.xml.pgp.

---

[6] To enable the encryption a *.cer file is supplied bij DNB. Linux users will need to convert this file to a *.pem file.

1.4.7 Compressing two encrypted files

Both pgp files, the zipped and encrypted data and the sleutel.xml.pgp file, are compressed using (win)zip resulting in a file with the extension .zip. The name of the file is left at the discretion of the reporter.
The result of this procedure is a ZIP file that will be the payload of the MIME file required by Logius.

1.4.8 Naming the compressed container

For submissions made based on an obligation created by DNB, the name of the file is a concatenation of:
{ReporterIdentifier} underscore {ReportingPeriod} underscore {DataDeliveryCode} underscore {TransmissionDate} underscore {SequenceNumber} dot zip. E.g.:

`NL001_2019-01-01_ZGRACRKANAXXXX_2019-02-10_01.zip`

The sequence number is used for providing a uniquely identifier for multiple messages for the same period and always starts with 01 increasing with increments of 1 making 99 the maximum possible transmission for a single report.
For submissions without an obligation created by DNB, the name of the file is a concatenation of:
{ReporterIdentifier} underscore {DataDeliveryCode} underscore {TransmissionDateTime} dot zip. E.g.:

`NL001_ZGRACRKANAXXXX_2019-02-10-18-01-12.zip`

Where the DateTime is noted as ccyy-mm-dd-hh-mm-ss.

1.4.9 MIME file naming

The name of the file created for transmission to Logius, and ultimately DNB, with an obligation is a concatenation of:
{ReporterIdentifier} underscore {ReportingPeriod} underscore {DataDeliveryCode} underscore {TransmissionDate} underscore {SequenceNumber} dot mime. E.g.:

`NL001_2019-01-01_ZGRACRKANAXXXX_2019-02-10_01.mime`

The sequence number is used for providing a uniquely identifier for multiple messages for the same period and always starts with 01 increasing with increments of 1 making 99 the maximum possible transmission for a single report.
The name of the file created for transmission to Logius, and ultimately DNB, without an obligation is a concatenation of:
{ReporterIdentifier} underscore {DataDeliveryCode} underscore {TransmissionDate} underscore {SequenceNumber} dot mime. E.g.:

`NL001_ZGRACRKANAXXXX_2019-02-10-18-01-12.mime`

1.4.10 DNB validations

Upon reception DNB will perform the following validations:

- Is it a valid MIME file?

- Is the [MIME] header XML valid?
- Can we match the SHA-1 encrypted reporter-period-DDC <aanleverkenmerk> in our system?
- Is the [MIME] payload identical to a previous submission?
- Can we unzip the container?
- Can we decrypt the optional sleutel.xml.pgp?
- Can we decrypt the optional encrypted container with the supplied key?
- Can we unzip the container?
- Is the required dnbmetadata.xml file present?
- Can we match the data files submitted with the data files expected?
- Can we check the integrity of the data files submitted?
- Do the columns in the CSV files match the requirements?
- Do the reporter and reported period values in the <aanleverkenmerk> match the ones in the report, if appropriate?

Failing any of these validations will result in a technical error, meaning the reporting obligation has not been met. A log containing the error(s) will be made available in the DLR. The requirement for a correct report still has a status of 'open'.

Passing these validations means the report has been received and business validations will commence on the data. These validations may also lead to an error.

1.5 Delivery notifications from DNB to the reporting agents

Following submission, DNB sends two types of XML notifications to Logius for the reporting agent. These notifications are published in Logius Digipoort but are also available as status of the submission in the Digital Reporting Portal (DLR):

- *Delivery confirmation* (Logius name: AfleverResponse)

A notification from DNB[7] confirming that the transporter has delivered the data and that DNB has accepted the data for further processing.

- *Validation results*

A notification from DNB with the delivery's technical result, and, in the event of validation errors, any validation codes.

---

[7] Please note that this is not the same as the transporter's notice of receipt of the data.

1.6 Example of a Logius XML header (with DNB required details)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tns:aanleverRequest xmlns:tns="http://logius.nl/digipoort/koppelvlakservices/1.2/">
    <tns:kenmerk>optional: a reporter generated unique message ID</tns:kenmerk>
    <tns:berichtsoort>DNB_rapportages[8]</tns:berichtsoort>
    <tns:aanleverkenmerk>SHA-1 encrypted DNB code</tns:aanleverkenmerk>
    <tns:identiteitBelanghebbende>
        <tns:nummer>KvK nr sender as stated in PKI-o</tns:nummer>
        <tns:type>KvK</tns:type>
    </tns:identiteitBelanghebbende>
    <tns:rolBelanghebbende/>
    <tns:berichtInhoud>
        <tns:mimeType>application/zip</tns:mimeType>
        <tns:bestandsnaam>name of the data file .zip</tns:bestandsnaam>
        <tns:Reference id="Pl0" URI="cid:Payload-0">
            <tns:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

    <tns:DigestValue>8e975d4fdc9bab7dc17f03d980e8bb2d0b5373666c835e207e9cd7b800bb8bb7</tns:DigestVa
lue>
        </tns:Reference>
    </tns:berichtInhoud>
    <tns:berichtBijlagen/>
</tns:aanleverRequest>
```

---

[8] For all deliveries DNB_rapportages has to be used as of April 18th 2019, until this date AnaCredit has to be used, even when the delivery DOES NOT concern AnaCredit (E.g. delivery for the Deposit Guarantee Scheme).

1.7 Examples of dnbmetadata.xml files

AnaCredit

```
<?xml version="1.0" encoding="UTF-8"?>
<dnbmeta:metadata xmlns:dnbmeta="https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01">
	<dnbmeta:reportingObligation>
		<dnbmeta:reporterIdentifier IDtype="RIAD" value="NL148"/>
		<dnbmeta:reportReferenceDate>2019-03-31</dnbmeta:reportReferenceDate>
		<dnbmeta:dataDeliveryCode>ZGRACRKANAXXXX</dnbmeta:dataDeliveryCode>
		<dnbmeta:gloCode>DNB_STAT_ANACREDIT_GLO_M</dnbmeta:gloCode>
	</dnbmeta:reportingObligation>
	<dnbmeta:files>
		<dnbmeta:file name="address.csv" hashType="SHA-256" hash="..." attachment="false"/>
		...
	</dnbmeta:files>
</dnbmeta:metadata>
```

NDGS

```
<?xml version="1.0" encoding="UTF-8"?>
<dnbmeta:metadata xmlns:dnbmeta="https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01">
	<dnbmeta:reportingObligation>
		<dnbmeta:reporterIdentifier IDtype="RIAD" value="NL148"/>
		<dnbmeta:reportReferenceDate>2019-03-31</dnbmeta:reportReferenceDate>
		<dnbmeta:dataDeliveryCode>ZNDGSXASCVXXXGXX</dnbmeta:dataDeliveryCode>
		<dnbmeta:gloCode>DNB_STAT_DGSXSCVXX_GLO_K</dnbmeta:gloCode>
	</dnbmeta:reportingObligation>
	<dnbmeta:files>
		<dnbmeta:file name="entity.csv" hashType="SHA-256" hash="..." attachment="false"/>
		...
	</dnbmeta:files>
</dnbmeta:metadata>
```

No-obligation report

```
<?xml version="1.0" encoding="UTF-8"?>
<dnbmeta:metadata xmlns:dnbmeta="https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01">
	<dnbmeta:reportingObligation>
		<dnbmeta:reporterIdentifier IDtype="MDM" value="3451183"/>
		<dnbmeta:dataDeliveryCode>OCBSXXXOVHXXXXXX</dnbmeta:dataDeliveryCode>
		<dnbmeta:gloCode>DNB_STAT_SR_BB_DSD_GLO_Q</dnbmeta:gloCode>
	</dnbmeta:reportingObligation>
	<dnbmeta:files>
		<dnbmeta:file name="jaarwaarneming.csv" hashType="SHA-256" hash="..."
attachment="false"/>
		...
	</dnbmeta:files>
</dnbmeta:metadata>
```

1.8 Example of a Logius MIME file intended for DNB

```
MIME-Version: 1.0
Content-Type: multipart/related;
        boundary="----=_Part_0_1067040082.1401967407487"
        start="<metadatapart>"

------=_Part_0_1067040082.1401967407487
Content-Type: application/xml; charset=UTF-8; name="name of the data file without .zip .metadata.xml"
Content-Transfer-Encoding: binary
Content-ID: "<metadatapart>"

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tns:aanleverRequest xmlns:tns="http://logius.nl/digipoort/koppelvlakservices/1.2/">
        <tns:kenmerk>optional: a reporter generated unique message ID</tns:kenmerk>
        <tns:berichtsoort>DNB_rapportages[9]</tns:berichtsoort>
        <tns:aanleverkenmerk>SHA-1 encrypted DNB code</tns:aanleverkenmerk>
        <tns:identiteitBelanghebbende>
                <tns:nummer>KvK nr sender as stated in PKI-o</tns:nummer>
                <tns:type>KvK</tns:type>
        </tns:identiteitBelanghebbende>
        <tns:rolBelanghebbende>rapporteur</tns:rolBelanghebbende>
        <tns:berichtInhoud>
                <tns:mimeType>application/zip</tns:mimeType>
                <tns:bestandsnaam>name of the data file .zip</tns:bestandsnaam>
                <tns:Reference id="Pl0" URI="cid:Payload-0">
                        <tns:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

        <tns:DigestValue>8e975d4fdc9bab7dc17f03d980e8bb2d0b5373666c835e207e9cd7b800bb8bb7</tns:DigestValue>
                </tns:Reference>
        </tns:berichtInhoud>
        <tns:berichtBijlagen/>
</tns:aanleverRequest>

------=_Part_0_1067040082.1401967407487
Content-Type: application/zip; charset=UTF-8; name="name of the data file .zip"
Content-Transfer-Encoding: base64
Content-ID: <Payload-0>

TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQgdmljdG9yLCBjb25zZWN0ZXR1ciBhZGlwaXNpY2luZyBlbGl0IGVyaWMsIHNlZCBkbyBlaXVzbW9
kIHRlbXBvciBiYXJ0dXMgaW5jaWRpZHVudCB1dCBsYWJvcmUgZXQgZG9sb3JlIGFzdG9yZSBtYWduYSBhbGlxdWEuIFV0IGVuaW0gYWQgbWluaW
```

---

[9] For all deliveries DNB_rapportages has to be used as of April 18th 2019, until this date AnaCredit has to be used, even when the delivery DOES NOT concern AnaCredit (E.g. delivery for the Deposit Guarantee Scheme).

```
0gdmVuaWFtIG1hcm5peCwgcXVpcyBub3N0cnVkIGV4ZXJjaXRhdGlvbiB1bGxhbWNvIGxhYm9yaXMgbmlzaSB1dCBhbGlxdWlwIGV4IGVhIGNvb
W1vZG8gY29uc2VxdWF0LiBEdWlzIGF1dGUgaXJ1cmUgZG9sb3IgaW4gcmVwcmVoZW5kZXJpdCBpbiB2b2x1cHRhdGUgdmVsaXQgZXNzZSBjaWxs
dW0gZG9sb3JlIGV1IGZ1Z2lhdCBudWxsYSBwYXJpYXR1ci4gRXhjZXB0ZXVyIHNpbnQgb2NjYWVjYXQgY3VwaWRhdGF0IG5vbiBwcm9pZGVudCw
gc3VudCBpbiBjdWxwYSBxdWkgb2ZmaWNpYSBkZXNlcnVudCBtb2xsaXQgYW5pbSBpZCBlc3QgbGFib3J1bS4=
------=_Part_0_1067040082.1401967407487--
```

1.9 The dnbmeta.xsd schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dnbmeta="https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01"
targetNamespace="https://statistiek.dnb.nl/schemas/xml/a2ameta/2019-02-01" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.2">
        <xsd:element name="metadata" type="dnbmeta:metadataType"/>
        <xsd:complexType name="metadataType">
                <xsd:sequence>
                        <xsd:element name="reportingObligation" type="dnbmeta:reportingObligationType"
minOccurs="1" maxOccurs="1"/>
                        <xsd:element name="files" type="dnbmeta:filesType" minOccurs="1" maxOccurs="1"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="reportingObligationType">
                <xsd:sequence>
                        <xsd:element name="reporterIdentifier" type="dnbmeta:reporterIdentifierType"
minOccurs="1" maxOccurs="1"/>
                        <xsd:element name="reportReferenceDate" type="xsd:date" minOccurs="0" maxOccurs="1"/>
                        <xsd:element name="dataDeliveryCode" type="dnbmeta:ddcType" minOccurs="1"
maxOccurs="1"/>
                        <xsd:element name="gloCode" type="dnbmeta:gloType" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
                <xsd:assert test="if (substring(//dnbmeta:dataDeliveryCode,1,1) = 'O') then
(substring(//dnbmeta:dataDeliveryCode,7,1) = 'X') else true()"/>
                <xsd:assert test= "if (exists(//dnbmeta:reportReferenceDate)) then
substring(//dnbmeta:dataDeliveryCode,1,1) ne 'O' else true()"/>  </xsd:complexType>
        <xsd:complexType name="reporterIdentifierType">
                <xsd:attribute name="IDtype" type="dnbmeta:reporterIdentifierTypes" use="required"/>
                <xsd:attribute name="value" type="xsd:NCName" use="required"/>
        </xsd:complexType>
        <xsd:simpleType name="ddcType">
                <xsd:restriction base="xsd:string">
                        <xsd:pattern value="[DOZ]{1}[A-Z]{5}[ADHJKMWX]{1}[A-Z]{3}[A-Z]{3}[GPSX]{1}[A-Z]{0,2}[A-
Z]{0,2}">
                                <xsd:annotation>
                                        <xsd:documentation>
        Fraction 1: Type of delivery, (D)oorlevering, (Z)elfstandig , (O)ntvangst zonder verplichting
        Fraction 2: Reporting framework (5 pos)
        Fraction 3: Reporting period, (A)dhoc, (D)ag, (H)alfjaar, (J)aarlijks, (K)wartaal, (M)aand, (W)eek,
(X)other
        Fraction 4: Report name (3 pos)
        Fraction 5: Report sub name (3 pos)
```

```
    Fraction 6: Consolidation, (G)roep, (P)artieel, (S)olo, (X)other
    Fraction 7: Optional Variant indicator (2 pos)
    Fraction 8: Optional Complementary codes (2 pos)
    Total 14-18 digits
   </xsd:documentation>
                </xsd:annotation>
            </xsd:pattern>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="gloType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="DNB_[A-Z]{4}[_][A-Z]{9}_GLO_[KM]{1}">
                <xsd:annotation>
                    <xsd:documentation>
    Fraction 1: Supervisor, DNB
    Fraction 2: Department, STAT
    Fraction 3: Report, ANACREDIT LLD
    Fraction 4: Document, GLO
    Fraction 5: Reporting period, (K)wartaal, (M)aand
   </xsd:documentation>
                </xsd:annotation>
            </xsd:pattern>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="filesType">
        <xsd:sequence>
            <xsd:element name="file" minOccurs="1" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:attribute name="name" type="xsd:NCName" use="required"/>
                    <xsd:attribute name="hashType" type="dnbmeta:hashTypes"
use="required"/>
                    <xsd:attribute name="hash" type="dnbmeta:hashType" use="optional"/>
                    <xsd:attribute name="attachment" type="xsd:boolean" use="optional"
default="false"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="hashType">
        <xsd:union memberTypes="xsd:hexBinary xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="hashTypes">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="N.A."/>
```

```
                        <xsd:enumeration value="SHA-0"/>
                        <xsd:enumeration value="SHA-1"/>
                        <xsd:enumeration value="SHA-256"/>
                        <xsd:enumeration value="SHA-512"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name="reporterIdentifierTypes">
                <xsd:restriction base="xsd:token">
                        <xsd:enumeration value="CRM"/>
                        <xsd:enumeration value="KvK"/>
                        <xsd:enumeration value="LEI"/>
                        <xsd:enumeration value="MDM"/>
                        <xsd:enumeration value="RIAD"/>
                        <xsd:enumeration value="RSIN"/>
                </xsd:restriction>
        </xsd:simpleType>
</xsd:schema>
```

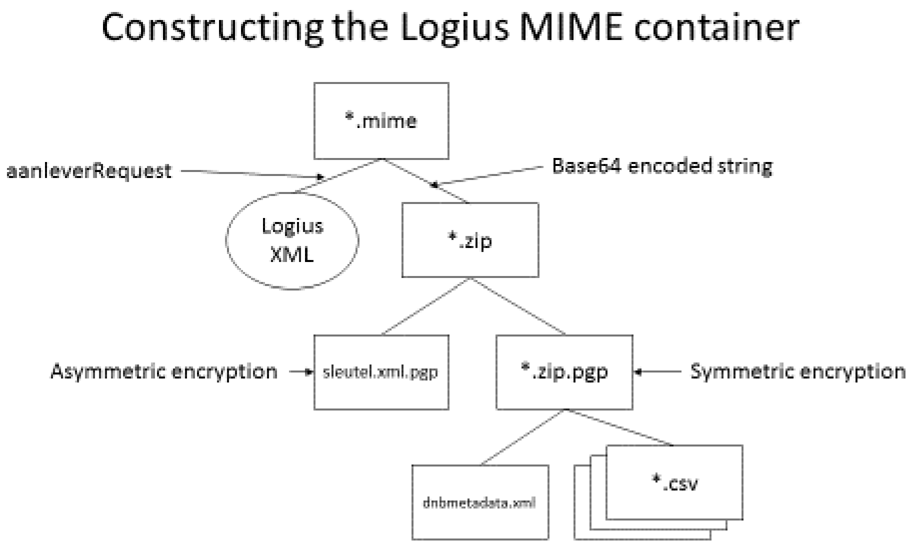
1.10 The dnbencryption.xsd schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dnbenc="https://statistiek.dnb.nl/schemas/xml/a2aenc/2018-03-01"
targetNamespace="https://statistiek.dnb.nl/schemas/xml/a2aenc/2018-03-01" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.1">
	<xsd:element name="parameters">
		<xsd:complexType>
			<xsd:attribute name="key" type="xsd:base64Binary" use="required" />
			<xsd:attribute name="IV" type="xsd:base64Binary" use="required" />
		</xsd:complexType>
	</xsd:element>
</xsd:schema>
```

1.11 Example of a sleutel.xml file

```
<?xml version="1.0" encoding="UTF-8"?><parameters
xmlns="https://statistiek.dnb.nl/schemas/xml/a2aenc/2018-03-01
key="WZe1ZogfXTps5pVLy1JDUUtjHmCcRWlN2zi/+S9+R3k=" IV="GmjELnKrzNkAN3kDMxuPvg=="/>
```

1.12 Graphical representation of files sent to Logius



CSV format as defined by DNB based on IETF RFC4180

1. Each record MUST BE located on a separate line;
2. Each line MUST BE delimited by a line break (CRLF);
3. The last record in the file MAY HAVE a line break;
4. There ~~MAY~~ MUST BE a header line appearing as the first line of the file[10];
5. Within the header and each record there ~~SHOULD~~ MUST BE one or more fields[11];
6. Fields MUST BE separated by a field separator: the ~~comma~~ semi-colon[12];
7. The last field of the header and each record MUST NOT have a field separator;
8. Each field ~~MAY~~ MUST BE enclosed by double quotes;
9. ~~If fields are not enclosed by double quotes, double quotes MUST NOT appear inside fields;~~
10. ~~Fields containing CRLF, double quotes and semi-colons SHOULD BE enclosed in double quotes;~~
11. ~~If fields are enclosed by double quotes, a double quote appearing inside a field MUST BE escaped by preceding it with another double quote.~~

Note: DNB uses “;” (three characters) internally as its field seperator

---

[10] More strict than the RFC4180
[11] More strict than the RFC4180
[12] Deviated from RFC4180, commas have a distinguishable meaning between the US/UK and Europe main land.