

Etere

a consistent system

ETERE TAPELESS WEB SERVICES: The best automated solution for content delivery.

Please note that this paper concerns a preliminary version of the product, therefore, it may be altered depending on the changes and/or additions performed to the final version.

Table of Contents

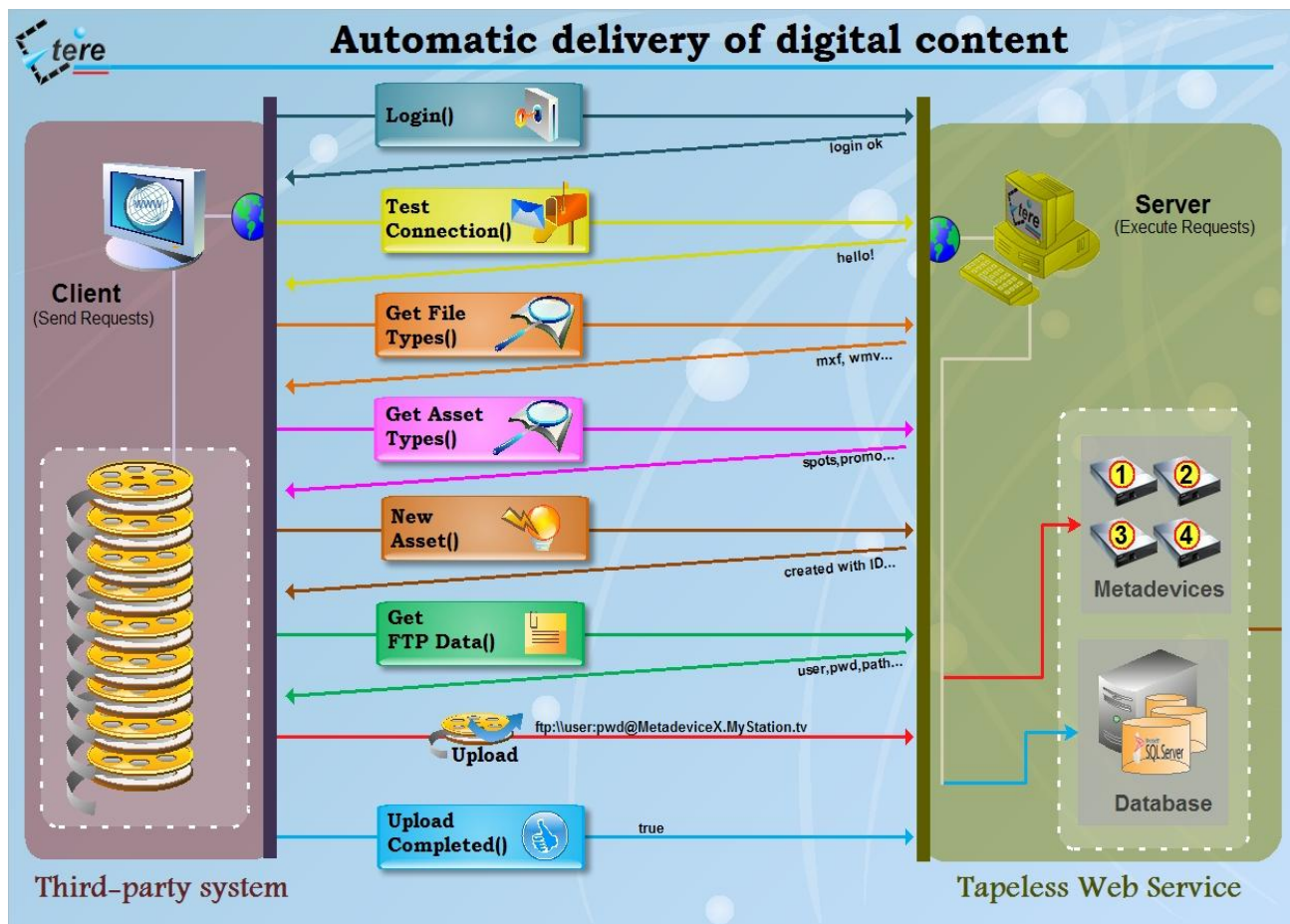
1. INTRODUCTION	3
2. DETAILED SOLUTION	4
2.1 SMPTE Metadata	5
2.2 FTP Upload	6
3. SUPPORTED FUNCTIONS.....	8
3.1 GetAssetTypes	9
3.2 GetFileTypes	9
3.3 GetFTPData	10
3.4 Login	10
3.5 NewAsset	11
3.6 NewAssetFromXmlFile	11
3.7 NewAssetFromXmlFileWithDescription	12
3.8 NewAssetFromXmlFileWithDescription	12
3.9 TestConnection	13
3.10 UploadCompleted	13
4. KEY BENEFITS.....	14

1. INTRODUCTION

Etere Tapeless Web Services is a web service which allows third-party applications and systems to upload files into an Etere system. **Etere Tapeless Web Services** makes available - via web - a series of functions (methods) that permits to consult the asset information present in the Etere system as well as to create the assets for which media needs to be uploaded.

The communication between the **server** and **clients** is performed through the sending of an XML message by the client who requests **certain information** (e.g. available asset types) or directly execute a **certain function** (e.g. create an asset).

All the functions supported by **Etere Tapeless Web Services** have been illustrated in the diagram below, thus indicating how to perform them sequentially and quickly:



This paper is aimed to provide complete information on the functioning of the system developed by Etere for the **automatic delivery of digital content** as well as on the **specific functions supported** by it. Moreover, a special chapter will enumerate and describe the benefits derived from the use of the web service treated in this document.

2. DETAILED SOLUTION

Etere Tapeless Web Service is the more efficient and secure solution developed by Etere for allowing stations to automatically and remotely receive digital content from other Non-Etere systems (e.g. production houses, commercial agencies, non-linear editors, etc.). The XML interface of the system allows remote clients to automatically upload files and metadata related to assets present in the Etere database, smartly integrating the upload procedure with the automatic triggering of workflows, before and after the upload completion.

Etere Tapeless Web Services enables authenticated users to deliver digital content and related metadata, providing them with all the data required for uploading files such as username, password and upload path. The overall content delivery functioning of **Etere Tapeless Web Services** is essentially performed through the following sequential steps:

	CLIENT	SERVER
1	The client performs the login into the system through the –unencrypted- authentication of its username and password .	In case of login success, the server will assign a session ID for the execution of operations related to the open session.
2	The client can request a list of the asset types and file types (codecs) supported for the uploading of digital content.	The server provides information concerning the asset types and file types (codecs) enabled for the uploading of content.
3	If the asset to be uploaded is not present in the Etere database, the client will be able to create it in order to upload it, specifying its type, codec and metadata (XML).	The server creates the asset according to the requested specifications, communicating to the client the ID with which the asset has been created within the system.
4	The client requests all connection information necessary for uploading a certain asset from a specific IP address .	The server provides unencrypted information about the upload path , the upload filename and the access credentials .
5	The client uploads the file in the given path using the provided access credentials.	The server remains waiting for the file to be uploaded.
6	Once the upload process finishes, the client communicates to the server the successful completion of the upload operation.	The server updates the presence of the uploaded file in the database and launches a predefined post-upload workflow .

2.1 SMPTE Metadata

As mentioned before, the delivery of content can be integrated with the Etere’s SMPTE metadata to be associated to uploaded assets. **Etere Tapeless Web Service** will be able to receive metadata contained in XML documents structured according to the following scheme:

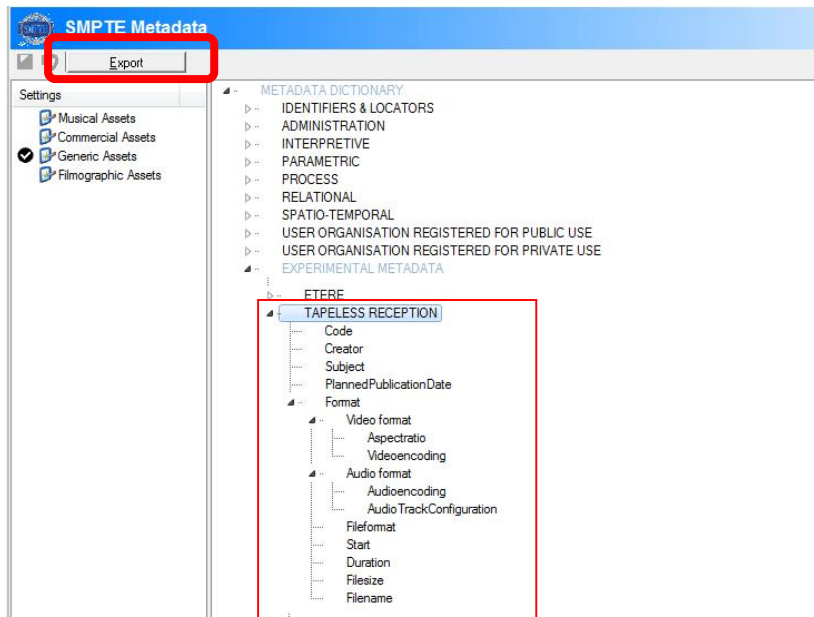
```
<?xml version="1.0" encoding="UTF-8"?>
-<coremetadata>
  <title item="0F.02.00.00.00.00.00.00">My title</title>
  <code item="0F.08.00.00.00.00.00.00"></code>
  <creator item="0F.06.00.00.00.00.00.00">My production house</creator>
  <subject item="0F.0D.00.00.00.00.00.00">My subject</subject>
  <type item="0F.14.00.00.00.00.00.00">My program type</type>
  <plannedpublicationdate
item="0F.0A.00.00.00.00.00.00">14/05/2012</plannedpublicationdate>
-<format>
  -<videofORMAT>
    <aspectratio item="04.01.01.01.01.00.00.00">16:9</aspectratio>
    <videoencoding item="0F.0E.00.00.00.00.00.00"></videoencoding>
  </videofORMAT>
  -<audioformat>
    <audioencoding item="0F.0F.00.00.00.00.00.00"></audioencoding>
    <audiotrackconfiguration
item="0F.10.00.00.00.00.00.00"></audiotrackconfiguration>
  </audioformat>
  <fileformat item="04.01.0B.01.00.00.00.00">WMV</fileformat>
  <start item="0F.11.00.00.00.00.00.00">25400</start>
  <duration item="07.02.02.01.02.05.00.00">115000</duration>
  <filesize item="0F.12.00.00.00.00.00.00"></filesize>
  <filename item="0F.13.00.00.00.00.00.00">mytest.wmv</filename>
</format>
</coremetadata>
```

The name of the **<tag>** represents the name of the SMPTE metadata to be inserted, the **“item”** attribute is the unique identifier of the SMPTE metadata and the **field value** is the proper content of the SMPTE metadata, just as shown in the following example:

NAME	IDENTIFIER	VALUE	NAME
<fileformat item="	04.01.0B.01.00.00.00.00">	WMV	</fileformat>

Supported metadata will be based on the items present in the **“Tapeless Reception”** SMPTE class, this data will be used for **mapping metadata** received from clients and inserting it into the Etere database. In case the received metadata field doesn’t exist, the specified fields will be either invalidated (i.e. not inserted) or validated (i.e. created within the system) depending on the server settings.

Moreover, all metadata specified under the **“Tapeless Reception”** SMPTE class will be available for export due to a function which permits to create an XML file containing all the – empty- SMPTE fields which are available in the system.



```
<?xml version="1.0" encoding="UTF-8"?>
<METADATA_DICTIONARY>
  <EXPERIMENTAL_METADATA>
    <TAPELESS_RECEPTION>
      <Title item="0F.02.00.00.00.00.00.00.00" />
      <Code item="0F.08.00.00.00.00.00.00.00" />
      <Creator item="0F.06.00.00.00.00.00.00.00" />
      <Subject item="0F.0D.00.00.00.00.00.00.00" />
      <Type item="0F.14.00.00.00.00.00.00.00" />
      <PlannedPublicationDate
item="0F.0A.00.00.00.00.00.00.00" />
      <Format>
        <Video_format>
          <Aspectratio item="04.01.01.01.01.00.00.00" />
          <Videocoding item="0F.0E.00.00.00.00.00.00.00" />
        </Video_format>
        <Audio_format>
          <Audioencoding item="0F.0F.00.00.00.00.00.00.00" />
          <AudioTrackConfiguration
item="0F.10.00.00.00.00.00.00.00" />
        </Audio_format>
        <Fileformat item="04.01.0B.01.00.00.00.00.00" />
        <Start item="0F.11.00.00.00.00.00.00.00" />
        <Duration item="07.02.02.01.02.05.00.00" />
        <Filesize item="0F.12.00.00.00.00.00.00.00" />
        <Filename item="0F.13.00.00.00.00.00.00.00" />
      </Format>
    </TAPELESS_RECEPTION>
  </EXPERIMENTAL_METADATA>
</METADATA_DICTIONARY>
```

2.2 FTP Upload

The server will provide all the data required by the requesting client for uploading media related to a specific asset via FTP, this information includes:

Data	Example
ID Metadevice	8
FTP address	ftp://195.120.230.35/Myserver/MyStorage
Username	User982
Password	2f4g58y2

Etere Tapeless Web Services: The best automated solution for content delivery

The abovementioned data will allow a client (**username** and **password**) to remotely upload a file into a certain device (**id metadvice**) through a protocol (**FTP address**) specified by the server for accessing the Etere media system.

2.3 Users management

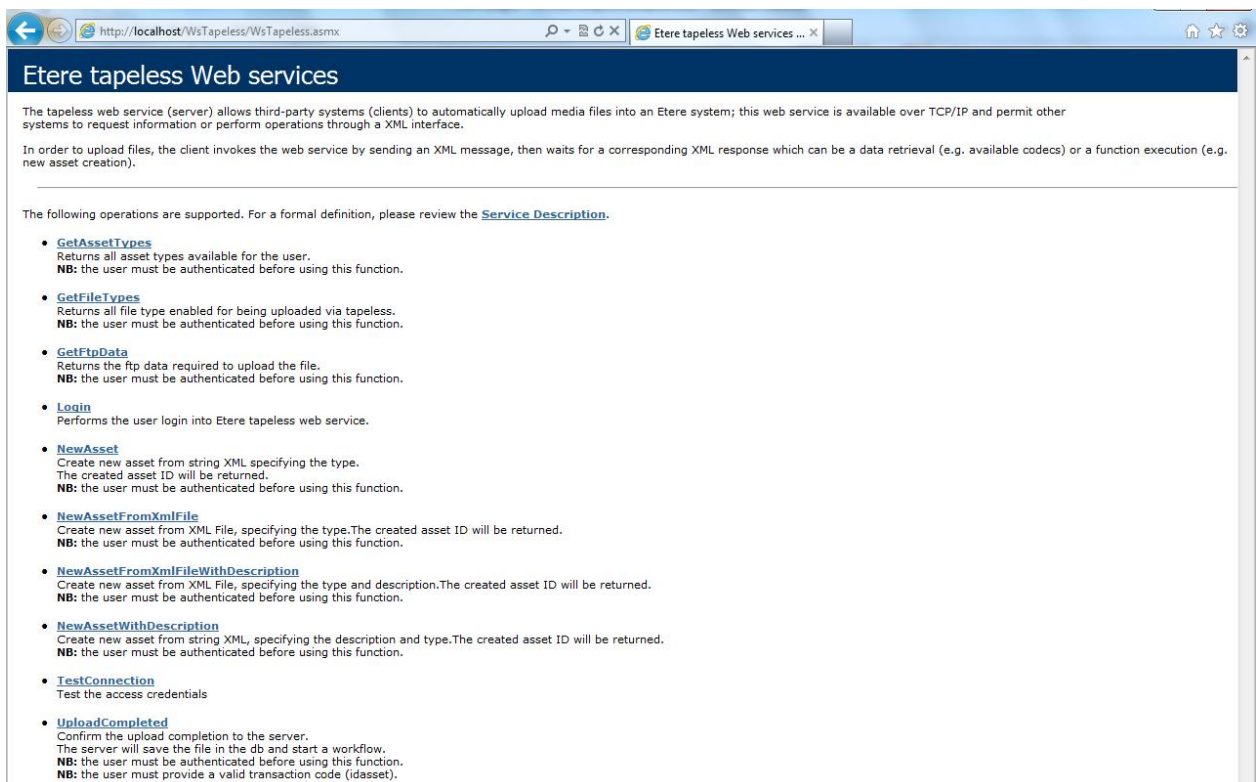
The web service authentication will be allowed only for those users configured in the Etere's Personal data module, said users must count with an enabled username and password credentials.

Please note that it won't be possible to access the web service using other user accounts out of the ones configured in the Personal data (e.g. SIS).

3. SUPPORTED FUNCTIONS

Etere Tapeless Web Service currently has more 10 functions for the consultancy and delivery of digital content. In future, it will be possible to increment this number of functions according to the system usage scenarios and requirements.

It is worth mentioning that Etere Tapeless Web Service can be accessed through any web browser, a simple interface will allow users to view and test all methods provided by the web service:



1. Access the web service: <http://myserver/WsTapeless/WsTapeless.asmx>.
2. The main page containing all supported functions will be displayed.
3. It will be possible to manually use a function just by clicking on its name (e.g. Login).
4. The page related to the selected function will be loaded, allowing the user to insert the parameters required for executing it (e.g. username and password).
5. Once parameters are filled, press the “Invoke” button to send a **request** message to the web server.
6. The web server will answer to the request with a **response message** (e.g. required upload data).

3.1 GetAssetTypes

This function requests a list of all the **asset types** enabled by the station for being uploaded via web.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAssetTypes xmlns="http://microsoft.com/webservices/" />
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfTipologie xmlns="http://microsoft.com/webservices/">
  <tipologie>
    <Idtype>int</Idtype>
    <Cod_type>string</Cod_type>
    <Description>string</Description>
  </tipologie>
</ArrayOfTipologie>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.2 GetFileTypes

This function requests a list of all the **file types** (codecs) enabled by the station for web upload. Codecs are enabled in Etere Configuration under the codec's "tapeless" property.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFileTypes xmlns="http://microsoft.com/webservices/" />
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfCodecs xmlns="http://microsoft.com/webservices/">
  <Codecs>
    <Id>int</Id>
    <Descr>string</Descr>
    <Type>string</Type>
  </Codecs>
</ArrayOfCodecs>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.3 GetFTPData

Richiede i dati necessari per il **caricamento via FTP** di un materiale da un certo indirizzo IP.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFtpdata xmlns="http://microsoft.com/webservices/">
      <AssetID>int</AssetID> <!--Codice univoco del materiale dentro il sistema -->
      <IpAddress>string</IpAddress>
    </GetFtpdata>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfFtpdata xmlns="http://microsoft.com/webservices/">
  <Ftpdata>
    <IdMetaDevice>int</IdMetaDevice>
    <Name>string</Name>
    <Ftpaddress>string</Ftpaddress>
    <UserName>string</UserName>
    <Password>string</Password>
  </Ftpdata>
</ArrayOfFtpdata>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.4 Login

This function authenticates the customer into the system, **enabling** its capabilities to apply requests and execute functions.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Login xmlns="http://microsoft.com/webservices/">
      <username>string</username>
      <password>string</password>
    </Login>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://microsoft.com/webservices/">string</string>
```

NB: A timeout of 20 minutes will determine the automatic closure of inactive *authenticated sessions* (not *upload sessions*), in order to keep their session active clients will be required to execute the "KeepAlive" method each "X" minutes to tell the server they are active.

3.5 NewAsset

This function creates a new asset within the Etere database based on the **type** and **metadata** specified by the client (through a XML string). Once created, the server will return to the client the ID of the asset in order to allow its subsequent uploading.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <NewAsset xmlns="http://microsoft.com/webservices/">
      <xmlMetadata>string</xmlMetadata>
      <assetType>string</assetType>
    </NewAsset>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://microsoft.com/webservices/">int</int>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.6 NewAssetFromXmlFile

This function creates a new asset within the Etere database based on the **type** and **metadata** specified by the client (through an XML file). Once created, the server will return to the client the ID of the asset in order to allow its subsequent uploading.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <NewAssetFromXmlFile xmlns="http://microsoft.com/webservices/">
      <xmlFile>base64Binary</xmlFile>
      <assetType>string</assetType>
    </NewAssetFromXmlFile>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://microsoft.com/webservices/">int</int>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.7 NewAssetFromXmlFileWithDescription

This function creates a new asset within the Etere database based on the **type**, **description** and **metadata** specified by the client (through a XML string). Once created, the server will return to the client the ID of the asset in order to allow its subsequent uploading.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <NewAssetFromXmlFileWithDescription xmlns="http://microsoft.com/webservices/">
      <xmlFile>base64Binary</xmlFile>
      <assetType>string</assetType>
      <assetDescription>string</assetDescription>
    </NewAssetFromXmlFileWithDescription>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://microsoft.com/webservices/">int</int>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.8 NewAssetFromXmlFileWithDescription

This function creates a new asset within the Etere database based on the **type**, **description** and **metadata** specified by the client (through an XML file). Once created, the server will return to the client the ID of the asset in order to allow its subsequent uploading.

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <NewAssetWithDescription xmlns="http://microsoft.com/webservices/">
      <xmlMetadata>string</xmlMetadata>
      <assetDescription>string</assetDescription>
      <assetType>string</assetType>
    </NewAssetWithDescription>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://microsoft.com/webservices/">int</int>
```

NB: The execution of this function requires the user to be currently logged into the system.

3.9 TestConnection

This function simply allows testing the access credentials of the client. Use this function to, for example, know if the client's authenticated session client is still valid or not (the session will be automatically closed after 20 minutes of inactivity).

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TestConnection xmlns="http://microsoft.com/webservices/" />
  </soap:Body>
</soap:Envelope> </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://microsoft.com/webservices/">string</string>
```

3.10 UploadCompleted

This function sends a confirmation to the server about the completion of a file upload related to a certain asset (its provided ID must be valid), in this manner, the server will be able to update the database inserting the uploaded file among the asset's linked media.

Moreover, the server will launch a post-upload workflow (e.g. proxy file generation, quality check, etc.).

CLIENT request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadCompleted xmlns="http://microsoft.com/webservices/">
      <idAsset>int</idAsset>
      <idMetadevice>int</idMetadevice>
      <fileTypeID>int</fileTypeID>
      <fileName>string</fileName>
    </UploadCompleted>
  </soap:Body>
</soap:Envelope>
```

SERVER response:

```
<?xml version="1.0" encoding="utf-8"?>
<boolean xmlns="http://microsoft.com/webservices/">boolean</boolean>
```

NB: The execution of this function requires the user to be currently logged into the system.

4. KEY BENEFITS

This paper has described the implementation of an **Etere Tapeless Web Services** solution able to automate the remote delivery by authenticated third-parties of digital content into an Etere system; this solution will allow the stations to achieve a large number of operational benefits and advantages derived from its usage and which can be condensed in the following points:

- ◆ **Integration**, a simple but secure interface enables stations to integrate any system (e.g. non-linear editing) with Etere.
- ◆ **Reliability**, operations launched via workflow before/after the upload operation allow to increment the reliability and efficiency of the system.
- ◆ **Flexibility**, access credentials can be periodically changed by the station as a protection policy to enhance the security of the web service and the global system.
- ◆ **Scalability**, capability of adding, replacing and removing metadevices (e.g. video servers) without altering the design and functioning of the overall system.
- ◆ **Distribution**, stations can entrust a quick and safe delivery to any client around the world via an internet connection.
- ◆ **Accuracy**, content upload into the system is automatically checked to reduce the risk of errors whether in files and metadata.
- ◆ **Security**, all clients must authenticate themselves into the system using the access credentials provided by the station, thus guaranteeing the security of the overall service.