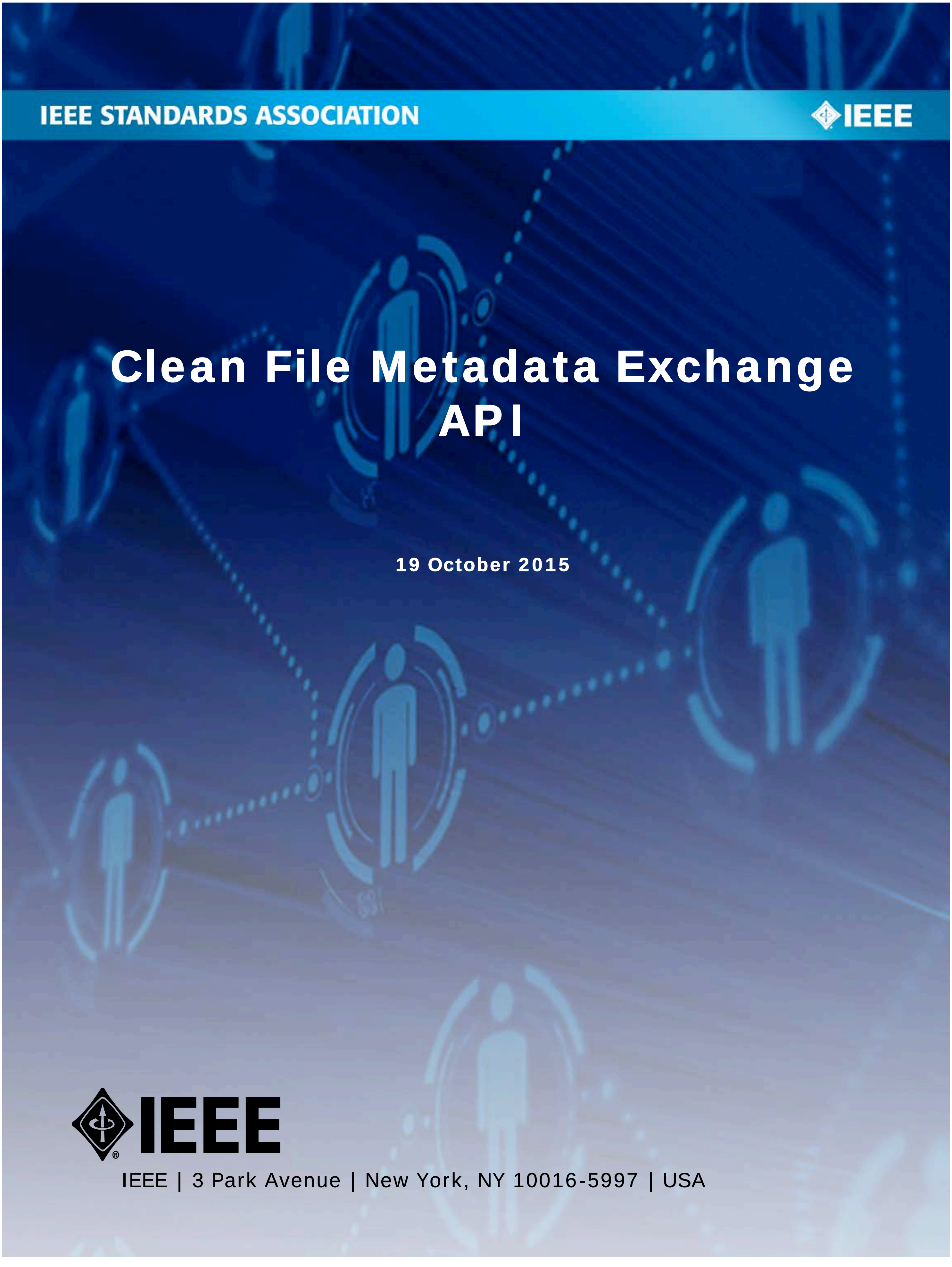

IEEE STANDARDS ASSOCIATION

Clean File Metadata Exchange API

19 October 2015

IEEE | 3 Park Avenue | New York, NY 10016-5997 | USA

CLEAN FILE METADATA EXCHANGE: API

Thomas Wegele
Avira Operations GmbH & Co.

Trademarks and Disclaimers

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

IEEE prohibits discrimination, harassment, and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

To order IEEE Press Publications, call 1-800-678-IEEE.
Find IEEE standards and standards-related product listings at: http://standards.ieee.org.

Notice and Disclaimer of Liability
Concerning the Use of IEEE-SA Industry Connections Documents

This IEEE Standards Association ("IEEE-SA") Industry Connections publication ("Work") is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the IEEE-SA Industry Connections activity that produced this Work. IEEE and the IEEE-SA Industry Connections activity members expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE-SA Industry Connections activity members disclaim any and all conditions relating to: results; and workmanlike effort. This IEEE-SA Industry Connections document is supplied "AS IS" and "WITH ALL FAULTS."

Although the IEEE-SA Industry Connections activity members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE OR IEEE-SA INDUSTRY CONNECTIONS ACTIVITY MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder. The policies and procedures under which this document was created can be viewed at http://standards.ieee.org/about/sasb/iccom/.

This Work is published with the understanding that IEEE and the IEEE-SA Industry Connections activity members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

CONTENTS

CLEAN FILE METADATA EXCHANGE: API

1. INTRODUCTION

The Clean File Metadata Exchange API is available as a Web service through HTTPS Post. The API authentication is done through HTTP authentication; any user with access to the API can call the HTTPS Post methods using the following scheme:

[protocol]://[user]:[password]@[APP-URI]/api/[method]

- Protocol: The protocol of the API, HTTPS
- User: The username of the user
- Password: The password of the user
- APP-URI: The URI of the application
- Method: The name of the method and the parameters

The following section contains a description of all available API methods. Fields that are marked with an asterisk (*) are mandatory and must be submitted; otherwise, the system will return an error message.

1.1 Submit metadata

a) Input parameters

Name	**Type**	**Description**
metadata*	POST - file	The input zip file of clean metadata.
response_type	GET|POST	The response type of the method by default is xml. It can be csv or xml.
*This parameter is required.		

b) Response parameters

Name	**Type**	**Description**
response	array	The response array of the method.
NOTE—The response is presented as the response_type parameter: csv or xml; xml by default.		

c) Response structure

Name	**Type**	**Description**
metadataid	integer	The unique ID stored in the system for the currently saved clean metadata.

1.2 getCompanies

a) Input parameters

Name	**Type**	**Description**
response_type	GET|POST	The response type of the method by default is xml. It can be csv or xml.

b) Result parameters

Name	Type	Description
companies	array	The array contains information regarding the companies available in the system.
NOTE—The response is presented as the response-type parameter: csv or xml; xml by default.		

c) Companies' element structure

Name	Type	Description
id	integer	The ID of the company
company	string	The name of the company
type	enum	3PSD, SSV or 3PSD & SSV

1.3 getLastId

The getLastId method will search the database for the most recently added clean metadata and return its metadataId and the date and time of the insertion.

a) Input parameters

Name	Type	Description
response_type	GET|POST	The response type of the method by default is xml. It can be csv or xml.

b) Result parameters

Name	Type	Description
metadataId	integer	The ID of the most recent clean metadata inserted into the system.
date	datetime	The date and time of the insertion of the metadata.
NOTE—The response is presented as the response-type parameter: csv or xml; xml by default.		

1.4 getMetadata

This method is used to get the XML for one or more submissions of clean metadata. It is supposed to have as an input parameter the startMetadataId, the ID of the first metadata archive that should be returned. It is obtained by the getLastId function, or by the submitMetadata method, and the amount of the objects that should be returned.

a) Input parameters

Name	**Type**	**Description**
startMetadataId *	GET|POST integer	The ID of the metaDataId.
amount	GET|POST integer	The amount of objects that should be returned; the default is 1.
*This parameter is required.		

b) Result

Name	**Type**	**Description**
metadata.zip	file	The zip file containing the xml files for the metadata found in the system.

1.5 listArchives

The listArchives method will show all the metadata archives available in the system.

a) Input parameters

Name	**Type**	**Description**
response type	GET|POST	The response type of the method by default is xml. It can be csv or xml.

b) Result parameters

Name	**Type**	**Description**
archives	array	The array contains information about the archives available in the system.
NOTE—The response is presented as the response-type parameter: csv or xml; xml by default.		

c) Archives element structure

Name	Type	Description
year	integer	The year of the archive
week	integer	The week number of the archive
metadata_no	integer	The number of metadata xml files existing in the archive
file	string	The filename and path of the archive
added	date	The date of the creation of the archive

1.6 downloadArchive

This method is used for downloading the archived zip file.

a) Input parameters

Name	Type	Description
file *	GET|POST string	The path and the filename of the archive that is requested to be downloaded
*This parameter is required.		

b) Result

Name	Type	Description
file.zip	file	The requested zip file that contains the meta data