GridOps Management Suite 3.10

File Claim Check Interface

Functional Specification

Document Version: 1.0

Updated: June, 2024

Table of Contents

Table of Figures

Table of Tables

Table of Documents

No table of figures entries found.

1. REFERENCES

#	Title	Description
1.	EcoStruxure GridOps Management Suite 3.10 File Monitoring Interface - Functional Specification	The document describes an out-of-the-box integration adapter designed for importing the files in the EcoStruxure GridOps by monitoring the predefined shared location.
2.	EcoStruxure GridOps Management Suite 3.10 Workforce Management Interface - Functional Specification[	The document describes an out-of-the-box integration adapter designed to be integrated with various workforce management systems.
3.	EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification	The document represents a set of common integration principles applied to all baseline integration adapters.
4.	EcoStruxure GridOps Management Suite 3.10 File Claim Check Interface	EcoStruxure GridOps Management Suite 3.10 File Claim Check Interface zip file contains essential configuration information, as well as web service definitions complemented with message examples.

2. ASSUMPTIONS AND PREREQUISITES

The File Claim Check Integration is developed under the following assumptions:

- External systems have possibility to transfer their data contained in a file to the EcoStruxure GridOps via predefined shared location. Format of the file depends on the type of system that is being integrated. To ensure that all file will be processed there should be time stamp within file name.
- To preserve the order of file processing, client must send files sequentially per session for same file type.
- On client's side send time out should be extended enough to ensure processing of the large files.
- File Claim Check Adapter supports configurable option for deleting file from (external) file share after file is successfully copied to EcoStruxure GridOps. This feature is disabled by default.
- Housekeeping of local file share is done by the File Claim Check Adapter.
- Housekeeping of remote (external) file share is done by the source system.
- Both SFTP and CIFS (SMB) file share types are supported.
- SFTP authentication can be done using username/password or private key file.
- Default encoding on SFTP server must be UTF-8.
- The File Claim Check Adapter supports SHA256 or SHA512 hash algorithm types – depends on configuration.
- The File Claim Check adapter supports configurable digital signature verification.
- Message payload is defined according to the FileNotification.xsd profile.

3. INTRODUCTION

EcoStruxure GridOps Management Suite is a family of solutions designed to help electric utilities in the operations and management of their grid. It is offered as EcoStruxure ADMS, EcoStruxure Grid Operation, EcoStruxure DERMS or EcoStruxure Energy Transmission Operation solutions, which share the same technology platform.

NOTE: The functionality described in this document applies to all solutions.

NOTE: Most images presented in this document are related to the EcoStruxure ADMS solution and should be used as an example. The images for other solutions may differ slightly.

Depending on the quantity of the data being exchanged between integrated systems, there can be two approaches of data transferring from the source to the destination system:

- In case of the small data quantity, the data is transferred as a payload within the message (mentioned use case is not covered by this document).
- In case of the large data quantity, the claim check pattern is used where the data is exported to a file and the target system is notified about the file location in the message. This process is described in following chapters.

When extract file of potentially large file size needs to be exchanged, it is more efficient to first prepare that file on the external shared file location (File Share) and then to send the message which contains the information that the file is ready and available at the specific location path.

The data can originate from different utility's systems like CIS, GIS, MDMS, etc. and it is intended to be imported in EcoStruxure GridOps. Depending on the business process, files can be exchanged between systems and/or the appropriate type of import can be invoked on the destination side. Depending on adapter configuration there are three possible functionalities:

- Copying data file(s) from the external file shared location to the local share location in DMZ system. Copied files can be imported by using functionality of FM adapter installed in Staging system. FM adapter located on Staging system monitors shared location in DMZ system, copies new file to local share on Staging system and invokes network import process (for additional information about the FM adapter, see *EcoStruxure GridOps Management Suite 3.10 File Monitoring Interface - Functional Specification* [1]). Results of the import process are extracts, changesets or some other data format that is understood by the EcoStruxure GridOps system.
- Copying data file(s) from the external file shared location to the local share location in DMZ system and uploading file as a multimedia attachment for following types of documents: work requests, switching plans, safety documents, incidents, incident hazards or trouble tickets.
- Copying data file(s) from the external file shared location to the local share location in DMZ and invoking corresponding service operation hosted on WFM adapter for crew model update (for additional information about the WFM adapter, see *EcoStruxure GridOps Management Suite 3.10 Workforce Management Interface - Functional Specification* [2]).

To support above mentioned functionalities, Claim Check Adapter is developed.

3.1. General Architecture

It is thoroughly described in the *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification* [3].

4. INTERFACE OVERVIEW

The File Claim Check Integration is implemented through the Claim Check Adapter component. The aforementioned adapter implements (hosts) the SOAP-based Web Service with the following operation:

- ReceiveFileNotificationService – used for receiving file notifications:
  - CreatedFileNotification operation

The following chapters provide more details regarding the interface listed above (Web Service) along with service operation, error handling scenarios, etc.

The use case diagram that represents common participants (actors) and users of the aforementioned interface in the File Notification Integration is presented in Figure 4.1.

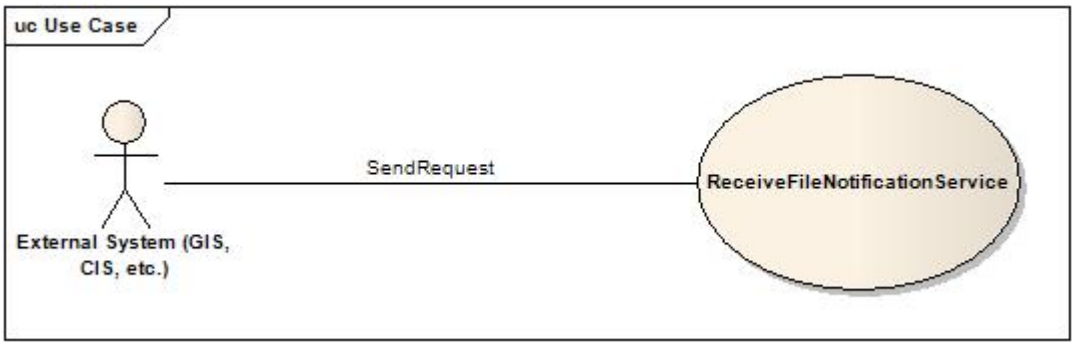

Figure 4.1 – The Claim Check Integration use case diagram

5. RECEIVEFILENOTIFICATION SERVICE

5.1. CreatedFileNotification Operation

5.1.1. Overview

As mentioned earlier, files containing data intended to be copied or imported are usually created in external systems. After the file has been created (write operation is finished), the external system (GIS, CIS, etc.) or process hosted on the middleware in most cases, creates the *CreatedFileNotificationEvent* object and sends it to the ReceiveFileNotification service.

The Claim Check Adapter component performs initial validation of the received message and transforms it into the request for import that the EcoStruxure GridOps can utilize. Along with this validation, the file is copied from the external file share (SFTP, etc.) to the local file share where it is accessible to the FM adapter from Staging system or it is used for multimedia upload or for crew model update process.

Depending on Claim Check adapter's configuration there are three possible functionalities:

- Copying file on predefined local file share location in DMZ system.
- Uploading file as multimedia attachment for following types of documents: work requests, switching plans, safety documents, incidents, incident hazards or trouble tickets.
- Supporting crew model update process by invoking corresponding service operation hosted on WFM adapter.

The *CreatedFileNotificationFault* is used for indication in cases when the message is invalid or in case of the communication issue.

The high-level sequence diagram is given in Figure 5.1., Figure 5.2. and Figure 5.3.

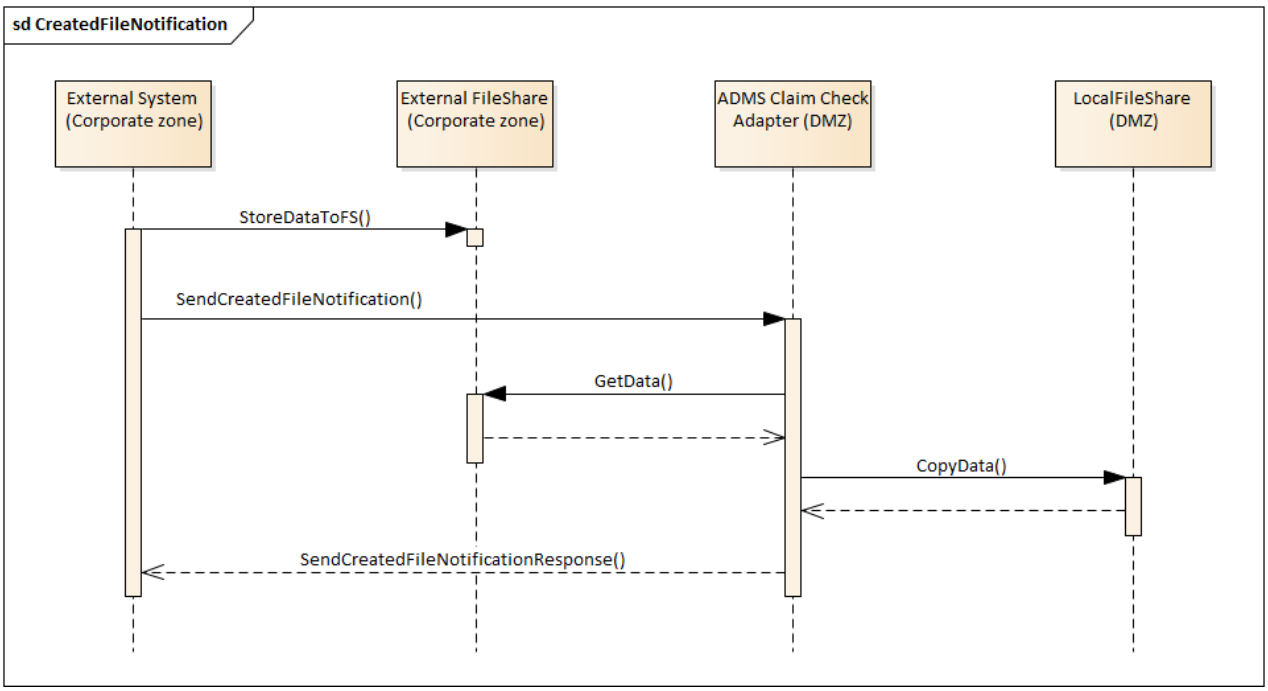

Figure 5.1 – The CreatedFileNotification copy operation execution

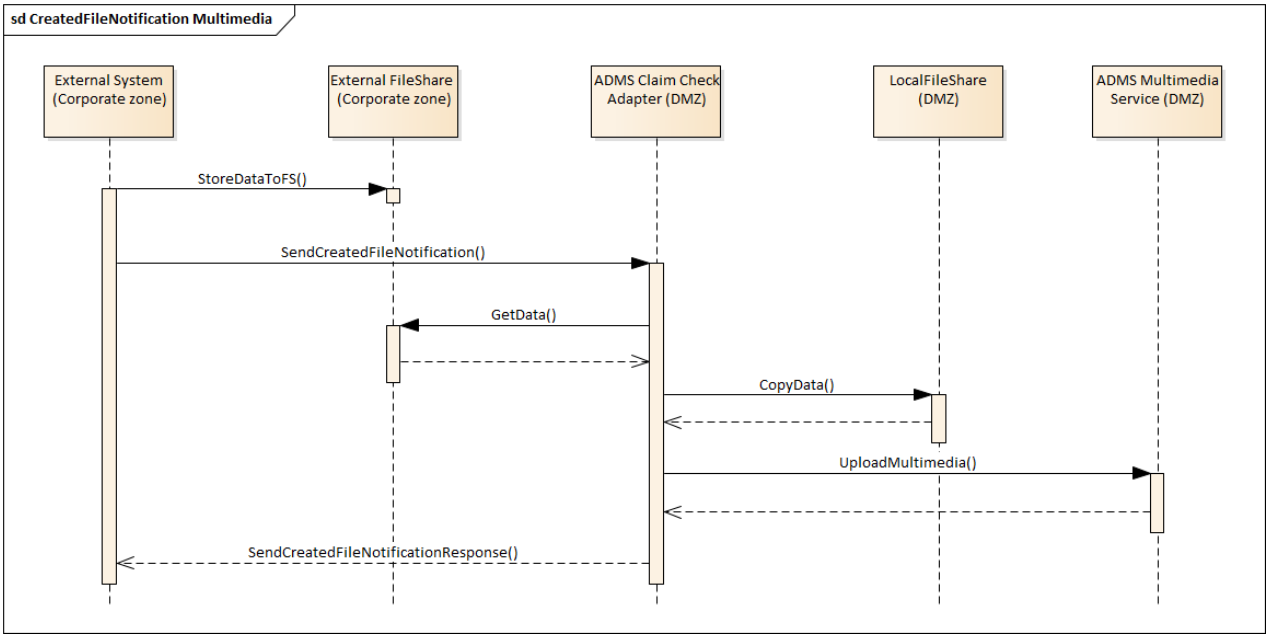

Figure 5.2 – The CreatedFileNotification uploading multimedia operation execution

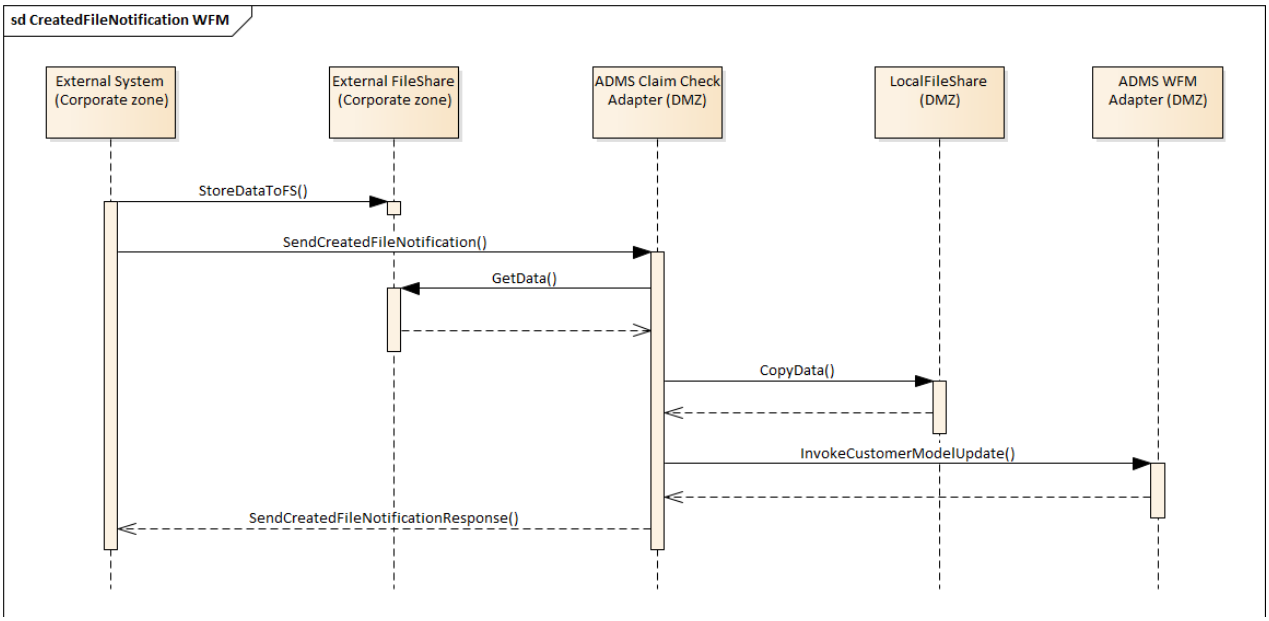

Figure 5.3 – The CreatedFileNotification crew model update operation execution

5.1.2. Use Cases

The list of possible use cases and corresponding faults is given in Table 5.1.

Table 5.1 – The CreatedFileNotification operation use cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Invalid Verb	Result	String	FAILED	External system sends request message with invalid Verb. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.9	
	Error.level	String	FATAL	
	Error.reason	String	InvalidVerb	
	Error.details	String	Invalid verb: {0}.	
Invalid Noun	Result	String	FAILED	External system sends request message with invalid Noun. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.5	
	Error.level	String	FATAL	
	Error.reason	String	InvalidNoun	
	Error.details	String	Invalid noun: {0}.	
Mandatory Element Missing	Result	String	FAILED	External system sends request message in which some of the mandatory elements are missing. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	1.8	
	Error.level	String	FATAL	
	Error.reason	String	InvalidMessage	
	Error.details	String	Received message is invalid against XSD schema. Reason: {0}.	
Unable to process the request	Result	String	FAILED	External system sends request message, but for some reason message processing fails due to various internal server error. Fault response message is sent by Claim Check Adapter.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	InternalServerError	
	Error.details	String	{0}.	
Created FileNotification – Invalid Object Type	Result	String	FAILED	
	Error.code	String	2.7	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.level	String	FATAL	External system sends CreatedFileNotification message with invalid object type. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.reason	String	InvalidObjectType	
	Error.details	String	Invalid Object Type: {0}.	
Created FileNotification – Copying file from the external system file share to the local file share	Result	String	OK	External system sends valid CreatedFileNotification message for copying file from the external system file share to the local file share. Response message is sent by Claim Check Adapter with OK result and event is created.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Created FileNotification – Crew model update successfully invoked	Result	String	OK	External system sends valid CreatedFileNotification message for starting crew model update process. After successfully invoking corresponding service operation hosted on WFM adapter, Claim Check Adapter sends response message to external system with OK result. Note: After file is successfully copied on share in DMZ system, adapter rises event.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Created FileNotification – Uploading multimedia attachment	Result	String	OK	External system sends valid CreatedFileNotification message for uploading multimedia attachment to work requests, switching plans, safety documents, incidents, incident hazards or trouble tickets. Response message is sent by Claim Check Adapter with OK result. Note: After file is successfully copied on share in DMZ system, adapter rises event.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Created FileNotification – URL is empty or invalid	Result	String	FAILED	External system sends CreatedFileNotification message with invalid (missing) URL. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	InvalidUrl	
	Error.details	String	File notification URL is invalid.	
Created FileNotification – Non-Existing File	Result	String	FAILED	External system sends CreatedFileNotification message with file location URL where specified file does not exist on the remote location. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	NonExistingFile	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.details	String	File notification URL does not exist on remote system.	
Created FileNotification – Format and URL mismatch	Result	String	FAILED	External system sends CreatedFileNotification message with different file format and URL file name. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.13	
	Error.level	String	FATAL	
	Error.reason	String	UrlFileNameAndFormatMismatch	
	Error.details	String	URL and format {0} mismatch in request message.	
Created FileNotification – Invalid File Format	Result	String	FAILED	External system sends CreatedFileNotification message with file format which differs from the file format specified in the URL field. Response message is sent by Claim Check Adapter with FAILED result and message is not processed. If format is not provided, response message is sent by Claim Check Adapter with OK result and message is processed.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	InvalidFileFormat	
	Error.details	String	File notification format {0} is invalid.	
Created FileNotification – Invalid File Notification Hash	Result	String	FAILED	External system sends CreatedFileNotification message where after verifying hash of the file Claim Check Adapter determines that it does not match to the one specified in the request message. Response message is sent by Claim Check Adapter with FAILED result and message is not processed.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	InvalidFileNotificationHash	
	Error.details	String	File notification hash {0} is invalid.	
Created FileNotification – External system file share is not available	Result	String	FAILED	External system sends CreatedFileNotification message for copying file or for multimedia upload. External system file share is unavailable. Claim Check Adapter cannot connect to file share in order to copy the file. Response message is sent by Claim Check Adapter with FAILED result.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	UnavailableFileShare	
	Error.details	String	Could not copy file: {0} from remote file share since it is unavailable.	
Created FileNotification – Failed to copy extract to local file share	Result	String	FAILED	External system sends CreatedFileNotification message for copying file of for multimedia upload. Local file share is unavailable. Claim Check Adapter tries to copy the file to a local file share but fails. Response message is sent by Claim Check Adapter with FAILED result.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	UnavailableLocalFileShare	
	Error.details	String	Could not copy file: {0} since LFS is unavailable.	

Use Case	Message Mapping			Action
	Property	Type	Value	
Created FileNotification – Request message for multimedia attachments does not contain Name attribute, or it is empty	Result	String	FAILED	External system sends CreatedFileNotification message for multimedia upload. Message contains object type **WorkRequest**, **SwitchingPlan, SafetyDocument, TroubleTicket, IncidentHazard** or **Incident** and does not contain Name attribute. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	MissingOrEmptyNameMultimediaFile	
	Error.details	String	File notification name is missing or empty.	
Created FileNotification – Object with specified ID not found	Result	String	FAILED	External system sends CreatedFileNotification message for multimedia upload. Message contains objectID which does not exist. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	InvalidObjectId	
	Error.details	String	Provided object id: {0} does not exist in ADMS	
Created FileNotification – Multimedia attachment exceeded maximum size	Result	String	FAILED	External system sends CreatedFileNotification message for multimedia upload of file that exceed maximum size. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	ExceededMaximumAttachmentSize	
	Error.details	String	Attachment is too big and can't be uploaded. Maximum attachment size is {0} bytes	
Created FileNotification – Multimedia service is unavailable	Result	String	FAILED	External system sends CreatedFileNotification message for uploading multimedia file attachment, but multimedia service is unavailable. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	UnavailableInternalService	
	Error.details	String	Could not invoke import for file since {0} service is unavailable.	
Created FileNotification – DMS_RealTime is unavailable	Result	String	FAILED	External system sends CreatedFileNotification message for uploading multimedia file attachment, but DMS_RealTime service is unavailable. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	UnavailableInternalService	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.details	String	Could not invoke import for file since {0} service is unavailable.	
Created FileNotification – WFM Adapter is unavailable	Result	String	FAILED	External system sends CreatedFileNotification message for update crew model, but WFM Adapter is unavailable. Claim Check Adapter cannot invoke service operation for crew model update, hosted on WFM Adapter since it is unavailable. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	5.3	
	Error.level	String	FATAL	
	Error.reason	String	UnavailableInternalService	
	Error.details	String	Could not invoke import for file since {0} service is unavailable.	
Created FileNotification – File already exist on local share (overwrite existing file is set to false in configuration)	Result	String	FAILED	External system sends CreatedFileNotification message for copying file from external system to EcoStruxure GridOps. Same file already exists on local share, and in adapter registry configuration file OverwriteExistingFileInLocalPath attribute is set to false. Response message is sent by Claim Check Adapter with FAILED result and message is discarded.
	Error.code	String	2.7	
	Error.level	String	FATAL	
	Error.reason	String	FileAlreadyExists	
	Error.details	String	File with name: {0} already exists on local share.	

6. MESSAGES

6.1. Common

6.1.1. Header

The header section is defined according to the IEC 61968-100. Currently, there are two required fields that must be populated:

- ***Verb*** – to identify a specific action to be taken. There is an enumerated set of valid verbs, where commonly used values include “get”, “create”, “change”, “cancel”, “close”, “execute” and “reply”. Within the event notification messages “past tense” verbs are used, which can include “created”, “changed”, “canceled”, “closed” and “executed”.
- ***Noun*** – to identify the subject of the action and/or the type of the payload, such as the FileNotification.

Field that can be optionally supplied include the following:

- Revision – to indicate the revision of the message definition. By default, this should be “1”.
- ReplayDetection – this is a complex element with a timestamp and a nonce used to guard against replay attacks. The timestamp is generated by the source system to indicate when the message was created. The nonce is a sequence number or randomly generated string (e.g. UUID) that would not be repeated by the source system for at least a day. This serves to improve encryption.
- ***Context*** – a string that can be used to identify the context of the message. This can help provide an application level guard against incorrect message consumption in configurations where there may be multiple system environments running over the same messaging infrastructure. Some example values are PRODUCTION, TESTING, STUDY and TRAINING.
- Timestamp – an ISO 8601 compliant string that identifies the time the message was sent. This is analogous to the JMSTimestamp provided by JMS. Either Zulu (‘Z’) time or time with a time zone offset may be used.
- Source – identifying the source of the message, which should be the name of the system or organization.
- AsyncReplyFlag – the Boolean data type (“true” or “false” values) that indicates whether a reply message will be sent asynchronously. By default, replies are assumed to be sent synchronously.
- ReplyAddress – the address to which replies should be sent. This is typically used for asynchronous replies. This should take the form of a URL, topic name or queue name. This is analogous to the JMSReplyTo field provided by JMS. This is ignored when using unidirectional integration patterns (e.g., AckRequired=false). If the reply address is a topic, the topic name should be prefixed by “topic”. If the reply address is a queue, the queue name should be prefixed by “queue”. If the reply address is a web service, the reply address should be a URL beginning with “http://” or “https:/”.
- AckRequired – the Boolean data type (“true” or “false” values) that indicates whether an acknowledgement is required. If false, this would indicate that a unidirectional integration pattern is being used for communicating transactional messages.

- User – a complex structure that identifies the user and associated organization. Should be supplied as it may be required for some interfaces, depending upon underlying implementations. This allows the UsersID string and optional the Organization string as sub-elements.
- MessageID – a string that uniquely identifies a message. Use of the UUID or sequence number is recommended. This is analogous to the JMSMessageID provided by JMS. A process should not issue two messages using the same MessageID value.
- CorrelationID – this is used to “link” messages together. This can be supplied on a request, so that the client can correlate a corresponding reply message. The server will place the incoming CorrelationID value as the CorrelationID on the outgoing reply. If not supplied on the request, the CorrelationID of the reply should be set to the value of the MessageID that was used on the request, if present. This is analogous to the use of the JMSCorrelationID provided by JMS. Given that the CorrelationID is used to ‘link’ messages together, it may be reused on more than one message. Use of a UUID or sequence number is recommended.
- Comment – any descriptive text, but shall never be used for any processing logic.
- Property – a complex type that allows the custom name/value pairs to be conveyed. The source and targets would need to agree upon usage. These are analogous to a Property as defined by JMS.
- Any – it can be used for custom extensions.

Figure 6.1 shows the graphical representation of the header field.

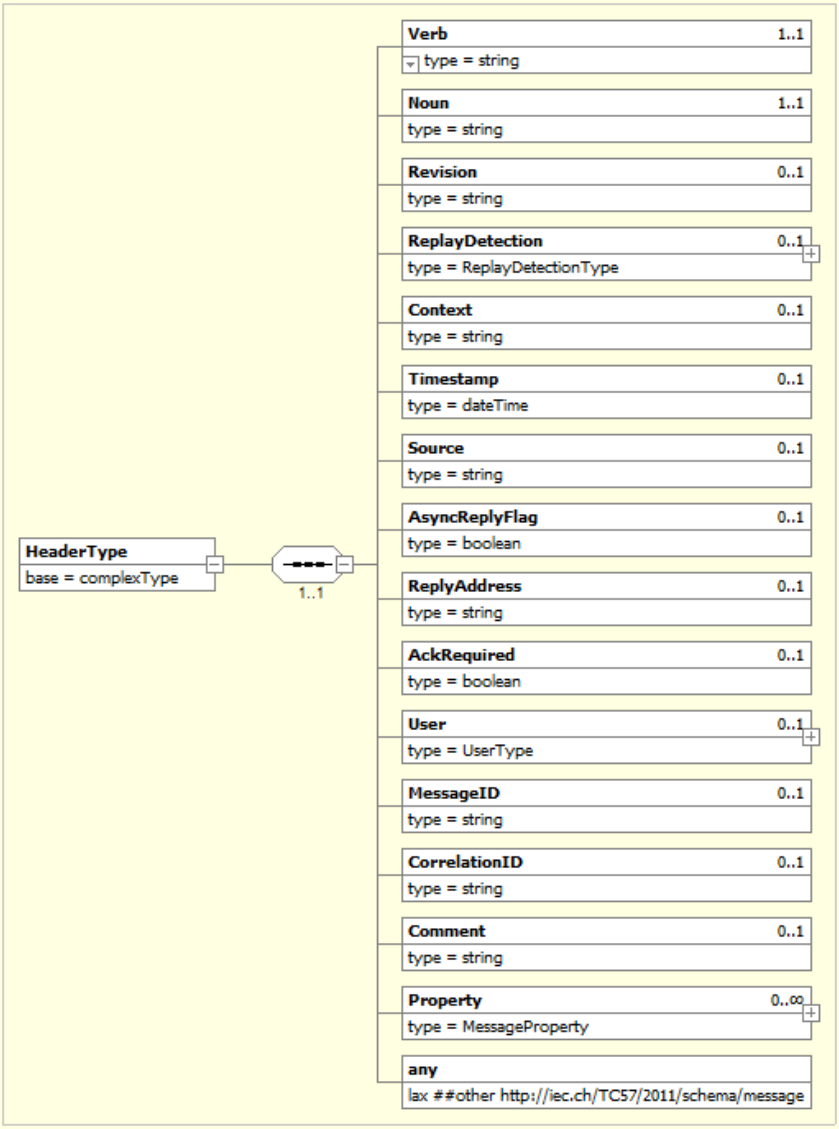

Figure 6.1 – The header field

6.1.2. Reply and Fault

The Reply.result value is an enumeration and would be populated in the following manner:

- "OK" – if there are no errors and all results have been returned. There is no requirement that a Reply.Error element be present.
- "PARTIAL" – if only a partial set of results has been returned, with or without errors. Existence of errors is indicated with one or more Reply.Error.code elements.
- “FAILED" – if no result can be returned due to one or more errors, indicated with one or more Reply.Error elements, each with a mandatory application level “code”.

If the result type is “PARTIAL” or “FAILED”, the **Error** field will be populated with the appropriate error description. The contents of the **Reply** and **Error** fields are presented in Figure 6.2.

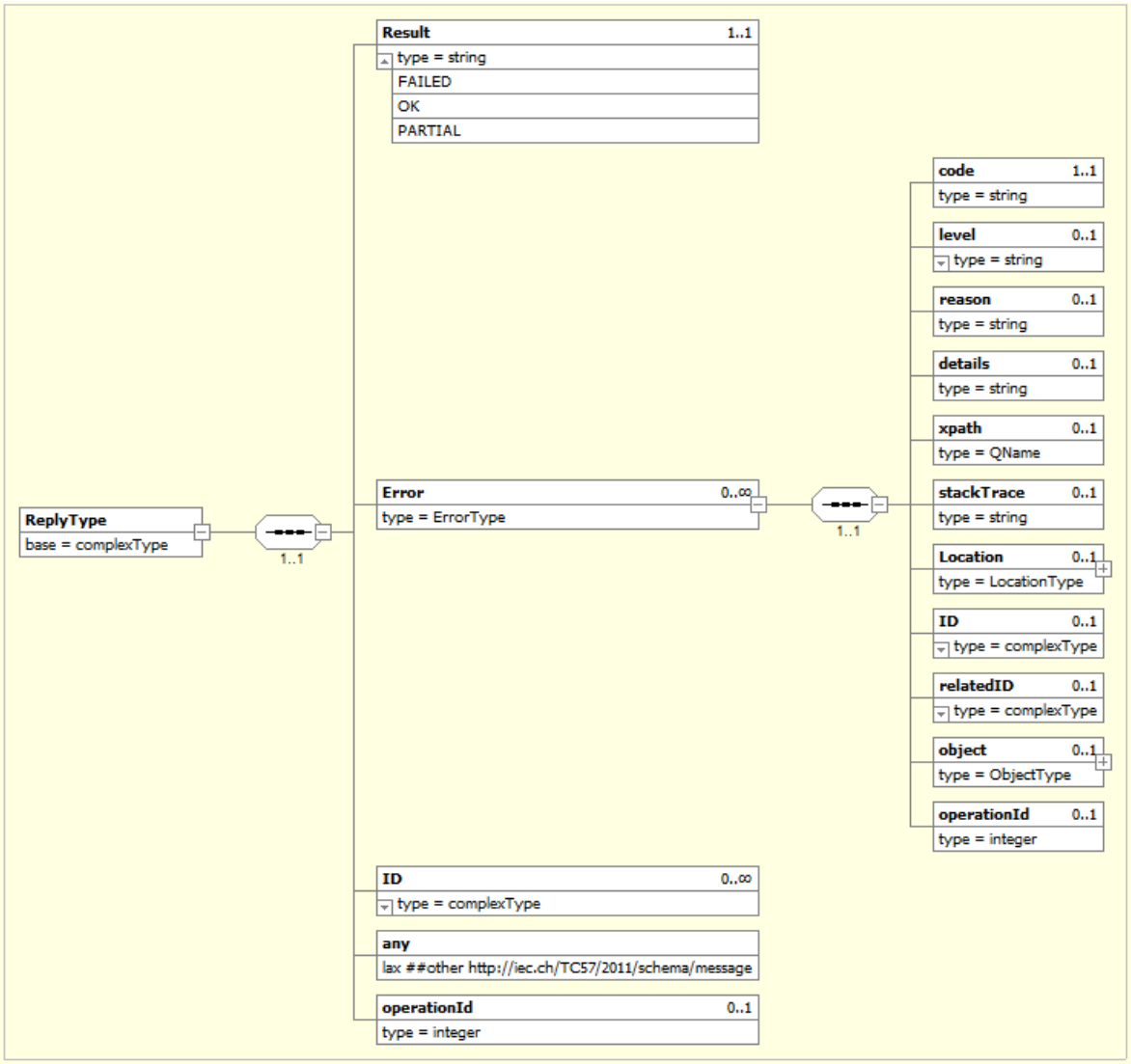

*Figure 6.2 – The **Reply** and **Error** field contents*

6.2. CreatedFileNotification Operation

The operation definition:

FileNotificationResponse CreatedFileNotification*(CreatedFileNotificationEvent)*

Both request and response messages contain the payload in form of the *FileNotification*.xsd schema which represents the CIM profile for file notification. The visual representation of the *FileNotification*.xsd schema is given in Figure 6.3.

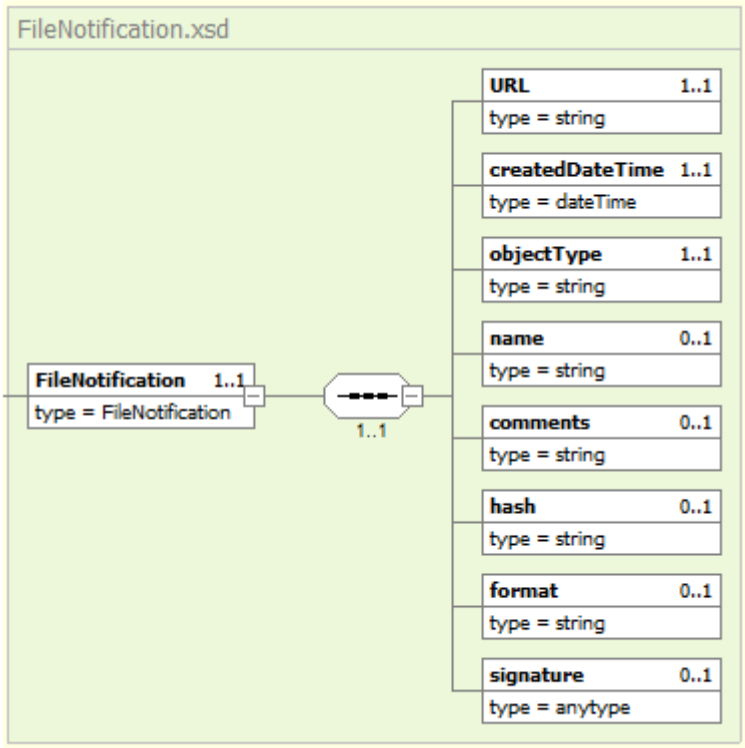

Figure 6.3 – FileNotification.xsd

6.2.1. Request

The *CreatedFileNotification* event message is defined according to the IEC 61968-100 and contains two following section:

- Header
- Payload

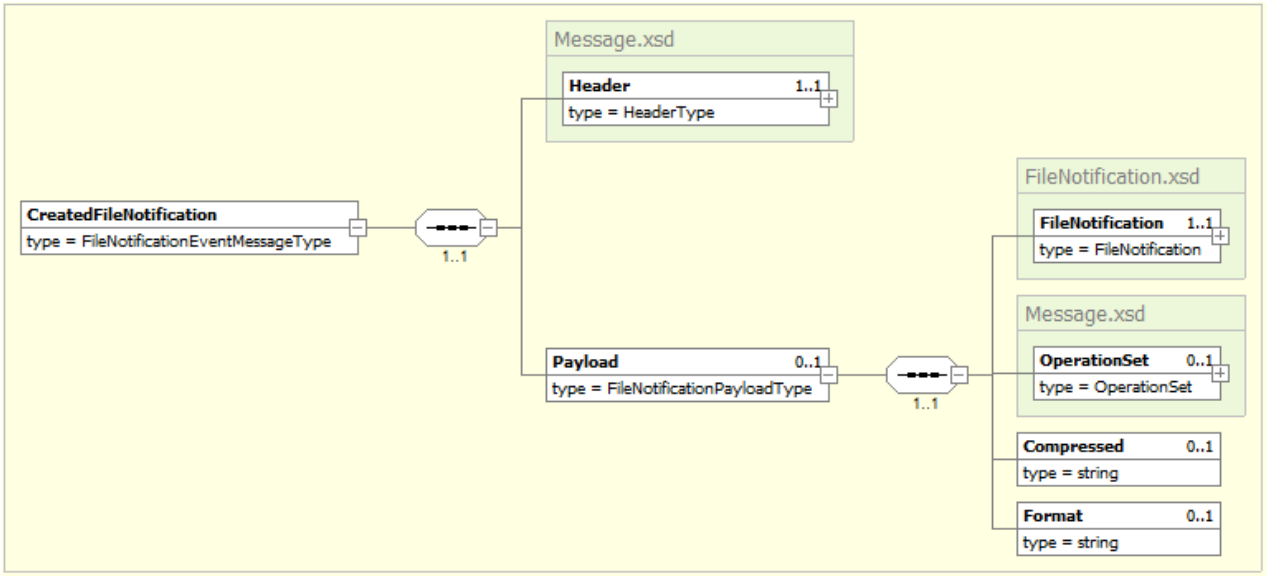

Figure 6.4 – The CreatedFileNotificationEvent message

The Payload field carries the CIM defined profile (*CreatedFileNotification.xsd*) for one or more import requests. Multiple import requests can be made if transferred file is archive (*zip*) containing multiple files of the same import type. Figure 6.5 depicts the *CreatedFileNotification* message.

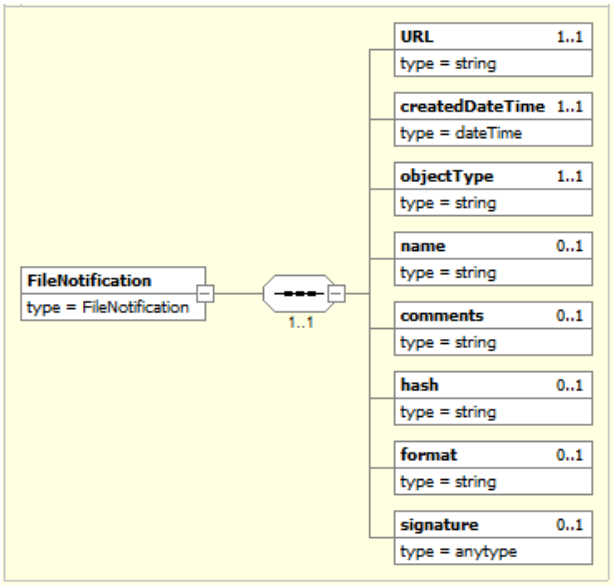

Figure 6.5 – FileNotification.xsd

Table 6.1 gives the description of entities that are a part of the *FileNotification.xsd*.

Table 6.1 – The FileNotification attributes description

CreatedFileNotification message			Description
Section	**Property**	**Type**	
Header	**Verb**	String	Identifier for a specific action to be taken. For this message, Verb is created.
Header	**Noun**	String	Identifier for the subject of the action and/or the type of the payload. For this message, Noun is FileNotification.
Header	Revision	String	Revision of CIM standard used. Default value is 2.0.
Header	**Timestamp**	DateTime	Timestamp when message was produced. Example: 2015-12-31T12:34:56+02:00
Header	Source	String	Source system or application that sends the message. For this message Source, can be: GIS, CIS, MDMS, etc.
Header	**MessageID**	String	Unique message ID to be used for tracking messages.
Header	**CorrelationID**	String	Same as message ID.
Request	URL	String	Location of the file on a FS, using a full path specification.
Request	CreatedDateTime	DateTime	Time of creation or last update
Request	ObjectType	String	File type: GIS, CIS, Landbase, LP, WorkRequest, SwitchingPlan, Incident, TroubleTicket, CrewModelUpdate etc. (List depends on the specific project requirements and can be configured).
Request	Name	String	Object identifier, to uniquely distinguish the object when appropriate, such as a file name, work request ID, switching plan ID, incident ID, trouble ticket ID, incident hazard ID, safety document ID.
Request	Comments	String	Comments describing file contents
Request	Hash	String	Hash of file contents using algorithm such as SHA256 or SHA512. Default one is SHA256.
Request	Format	String	File format, e.g. XML, CSV, ...
Request	Signature	Anytype	Not used. Signature is configured through registry configuration.

6.2.2. Response

After the import process is requested, the response is returned in form of the *FileNotificationResponse* message. The content of the response message is given in Figure 6.6.

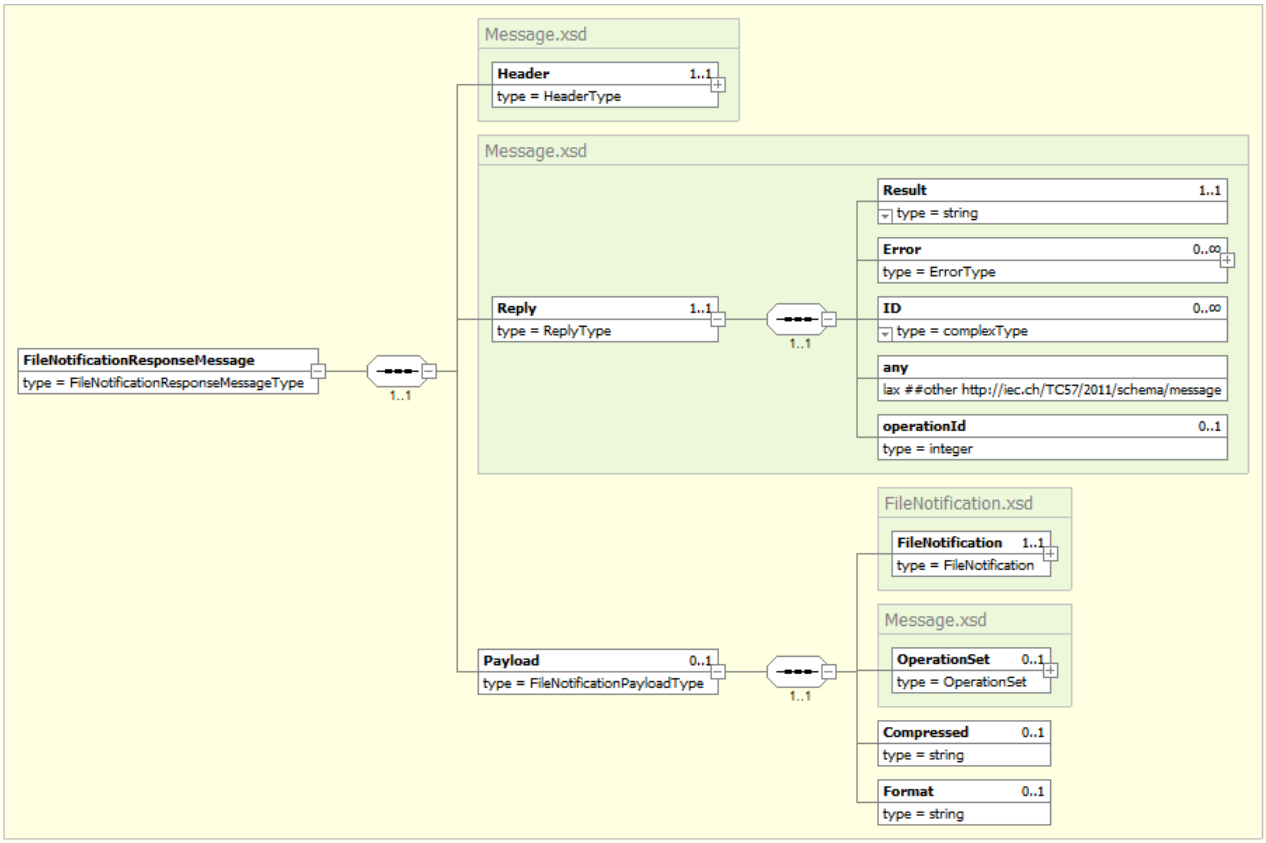

Figure 6.6 – The FileNotificationResponse message

6.2.3. Fault

The CreatedFileNotificationFault is depicted in Figure 6.7.

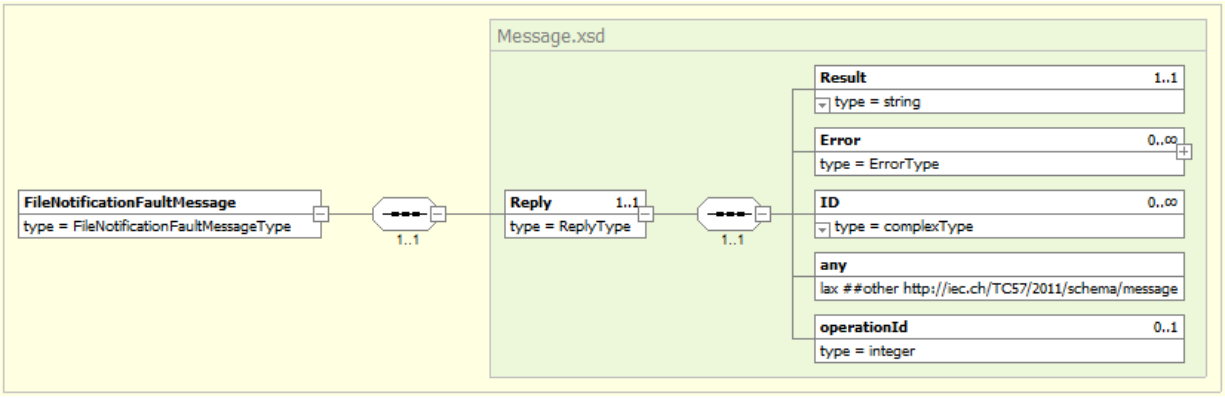

Figure 6.7 – The FileNotificationFault message

7. DEPLOYMENT

The Claim Check Adapter provides integration between the EcoStruxure GridOps and clients external applications such as: GIS, CIS, etc.

The deployment specification is provided in the following table:

Table 7.1 – The deployment specification

Deployment Specification	
Application	ClaimCheckAdapter
Critical process	No
OASyS service	OASyS DNA DMS_INTEGRATION Service
Servers	pdmz-int-1, pdmz-int-2, bdmz-int-1, bdmz-int-2
Zone	pdmz, bdmz
Installation Type	Product
Installation add-on name	Integration Adapters

Figure 7.1 depicts standard deployment configuration for all File Integration participants. If there is a need, the external (legacy) systems residing within the corporate environment of distribution utility can communicate with the File Integration interface through the ESB.

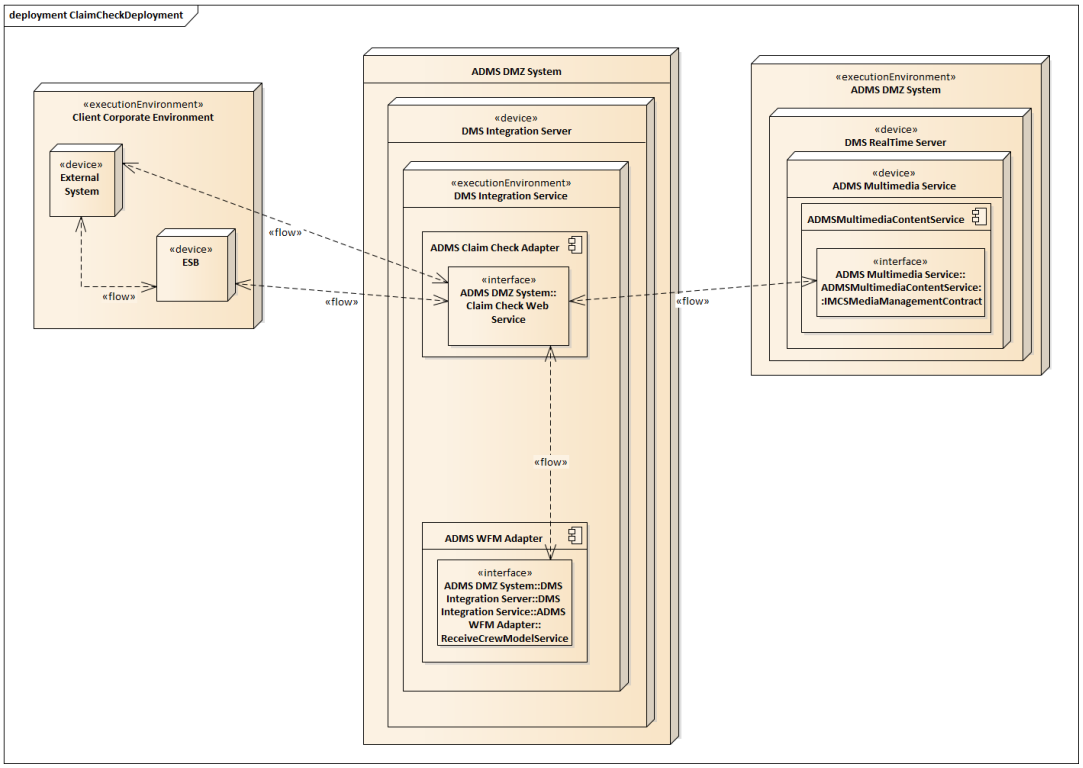

Figure 7.1 – The File Integration deployment diagram

8. INTERFACE CONFIGURATION

Claim Check adapter provides certain amount of configurability so that smaller adjustments in the functionality can be easily applied to the system, without interface down time. Such feature is provided through dedicated configuration files of the Claim Check adapter. Initially, following configuration files are used the adapter:

Table 8.1 – The configuration files specification

Name of the config file	Configuration File Description
AdapterClaimCheck	Registry configuration xml file
ErrorConfiguration_ClaimCheckAdapter	Error configuration xml file
AdapterClaimCheck_WebServiceConfiguration	Web service configuration xml file
FileTypeConfiguration_ClaimCheckAdapter	File type configuration xml file (contains settings for destination path, supported file formats)

For more details about adapters configuration files refer to the *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification* [3].

Detailed content of above-mentioned configuration files is provided within the *Configuration* folder in the *EcoStruxure GridOps Management Suite 3.10 File Claim Check Interface.zip* file [4].

9. APPENDIX

9.1. WSDL

The WSDL file, XSD schemas and sample messages defined according to the IEC 61968-100 for the *ReceiveFileNotificationService* is provided within the *Web Service Definitions* folder in the *EcoStruxure GridOps Management Suite 3.10 File Claim Check Interface.zip* file [4].

9.2. Message Examples

Message examples for several use cases are provided within the *Message Examples* folder in the *EcoStruxure GridOps Management Suite 3.10 File Claim Check Interface.zip* file [4].

9.3. File Naming Convention for Multimedia Attachments - Examples

All multimedia attachments are named as follows:

<ObjectId> + <"_"> + <ShortDescription> + "_" + < DateTime > + <FileFormat>", where

- <ObjectId> – custom id of Work request, switching plan, incident, incident hazard, safety document or trouble ticket.
- <ShortDescription> – short file description. Ex: CustomerReport.
- <DateTime> – is the UTC date and time that the file was created. Format is YYYYMMDDHHmmss,
- < FileFormat > – file format, e.g. pdf, jpg, etc.

Example: "SP000001_CustomerReport_201801202115.pdf".

Additionally, ObjectID must be specified within the *Name* attribute of the *CreatedFileNotification* message.

10. RELEASE NOTES

The following new features related to Product File Claim Check Interface were introduced in the software, starting from version 3.8.

10.1. Software Version 3.8.0

Feature	Description
File Monitoring and Claim Check interfaces	In order to enable easier deployment, configuration and maintenance, File Interface was separated into two interfaces: File Monitoring and Claim Check. Functionality was left intact, with small improvements related to more granular configuration of post copying actions.

10.2. Software Version 3.8 SP1

Feature	Description
Multimedia Attachments Upload	File Claim Check adapter supports upload of multimedia attachments for following types of documents: work requests, switching plans, safety documents, incidents, incident hazards and trouble tickets.
Increase Security	To increase security during file distribution within the ADMS system, the File Claim Check adapter is extended to support configurable digital signature verification.
Support Crew model update process	The Claim Check adapter supports import static crew model data.

10.3. Software Version 3.9

Feature	Description
File Claim Check Interface - Post-processing Support	File Claim Check adapter is enhanced with ability to perform simple, post-processing actions as is invoking some other services and configurable removing of old files on remote (SFTP/CIFS) location.

11. DEFINITIONS AND ABBREVIATIONS

Definition/Abbreviation	Description
ADMS	Advanced Distribution Management System (to be provided by Schneider Electric)
CIM	Common Information Model
CIS	Customer Information Service
DMD	Dynamic Mimic Diagram
DMZ	Access Service (DMZ) System
ESB	Enterprise Service Bus
LFS	Local File Share
FS	File Share
NIS	Network Import Service
WFM	Workforce Management
SFTP	SSH File Transfer Protocol
SOAP	Simple Object Access Protocol
WCF	Windows Communication Foundation
WS	Web Service
XML	Extensible Markup Language
XSD	XML Schema Definition