GridOps Management Suite 3.10

Redlining 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 Network Data Integration - Functional Specification	The document describes the Network Data Integration (NDI) module of EcoStruxure GridOps which represents a set of functionalities designed to facilitate the data migration as well as the sustained data integration between the most commonly encountered external data sources (e.g. GIS, EAM, CIS/CRM, MDMS) and EcoStruxure GridOps Network Model data repositories.
2.	EcoStruxure GridOps Management Suite 3.10 Model Management - Functional Specification	The document describes the general procedure of creating, verifying and distributing network model data and associated changesets within an EcoStruxure GridOps system.
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 Redlining Interface	EcoStruxure GridOps Management Suite 3.10 Redlining Interface zip file contains essential configuration information, as well as web service definitions complemented with message examples.

2. ASSUMPTIONS AND PREREQUISITES

Redlining (RLN) integration is designed with the following assumptions and prerequisites:

- Details about architecture, error handling and auditing, security are stated in the *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform – Functional specification* document [3].
- Message format exchanged between the EcoStruxure GridOps and client's system will be in accordance with the Redlining CIM Profile.
- Next to the SOAP-based web service, an out-of-the-box support for ArcFM Redlining Feature service is provided. Communication between Redlining Adapter and ArcFM Redlining Feature Service is performed using RESTful web services.
- Redlining integration is one-way integration where redline notes (map annotations, markups) will be published from EcoStruxure GridOps to the external system (typically GIS).
- Redline notes will be represented as a separate type of note.
- EcoStruxure GridOps users (operators, model managers, field crews, etc.) will be able to add/update/delete redline notes only from the geographic view.
- EcoStruxure GridOps users will be able to add redline note to an electrical and landbase equipment and on arbitrary position. Temporary elements are not included.
- EcoStruxure GridOps users will be able to add attachments of appropriate size and type to the redline note.
- Attachments will be transferred within the message payload as base64string.
- EcoStruxure GridOps users will be able to add redline notes in any production environment's systems (Real Time contexts in Core, Staging, and DMZ).
- All redline notes in production environment will be displayed in the Staging system (RealTime service in the Staging system) since redline notes from every production system (Core, DMZ) will be replicated to the Staging system.
- Staging redline notes are visible only to Staging users.
- DMZ and CORE redline notes are visible to all users.
- Redline notes created in DMZ and CORE systems can be removed only from the CORE system.
- Redline notes created in DMZ and CORE systems can be updated only from the CORE system.
- Redline notes created in the Staging system can be removed only from the Staging system.
- Redline notes created in the Staging system can be updated only from the Staging system.
- Redline notes will be preserved in the case when equipment, that has a redline note associated to it, is being deleted during model management process (for more details about the model management process please see *EcoStruxure GridOps Management Suite 3.10 Model Management - Functional Specification* [2]):
    - During equipment deletion, corresponding redline note will preserve element unique identifier (custom ID) and element's coordinates.
    - If graphical representation of the element is deleted, the corresponding redline note will be displayed in Note summary only. When graphical representation of the element is redrawn, redline note will be shown in the geographic view again.

- After the redline note is successfully propagated to the external system, an appropriate indication will be presented to the end user:
  - If for some reason the redline note is not sent successfully, it will be sent in the next cycle, when a new redline note is added/changed/deleted.
- Configurable coordinate conversion is supported. The coordinate system in which redline notes need to be sent must be agreed during the design workshops.

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.

The distribution networks are prone to changes, which occur daily by adding, changing or removing parts of the network. The source of these changes are typically corporate GIS systems. Changes in the GIS system, provided in form of feeder based CIM extracts, are delivered to the EcoStruxure GridOps. Such changes (CIM extracts) are imported, validated and applied to the network model. For more details on how the import process is executed, please see *EcoStruxure GridOps Management Suite 3.10 Network Data Integration - Functional Specification* [1]. During the conversion of CIM extracts to internal format, unexpected issues may occur due to invalid data according to CIM or EcoStruxure GridOps validation rules. Additionally, when the changeset is created and applied to the network model, the model manager user can preview it. By performing the visual inspection and validation, the model manager may detect some inconsistencies in the network. In order to correct those data inconsistencies in the source (GIS) as soon as possible, a model manager can send redline notes for specific elements in the network model. Redline notes can also be used by the control room operators to identify miss linked customers. Furthermore, during regular daily work in the field, field crews can use redline notes to notify GIS data entry personnel about potential corrections in the model.

By introducing redline notes, users can seamlessly mark features that require additional corrections and propagate such changes through a dedicated CIM (IEC 61968) compliant Redline (RLN) interface to any GIS system that supports the redlining concept.

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

Redline (RLN) interface is implemented through the RLN Adapter component. The aforementioned adapter implements SOAP based Web Service Client with appropriate set of operations:

- *SendRedlinesService* – used for sending Redline notes to the GIS system:
  - CreatedRedlines operation
  - ChangedRedlines operation
  - DeletedRedlines operation

The following chapters provide more details regarding Redline interface (web service client), appropriate operations, data mappings, error handling scenarios, etc.

The use case diagram that represents common participants (actors) and users of the Redline integration is given in Figure 4.1.

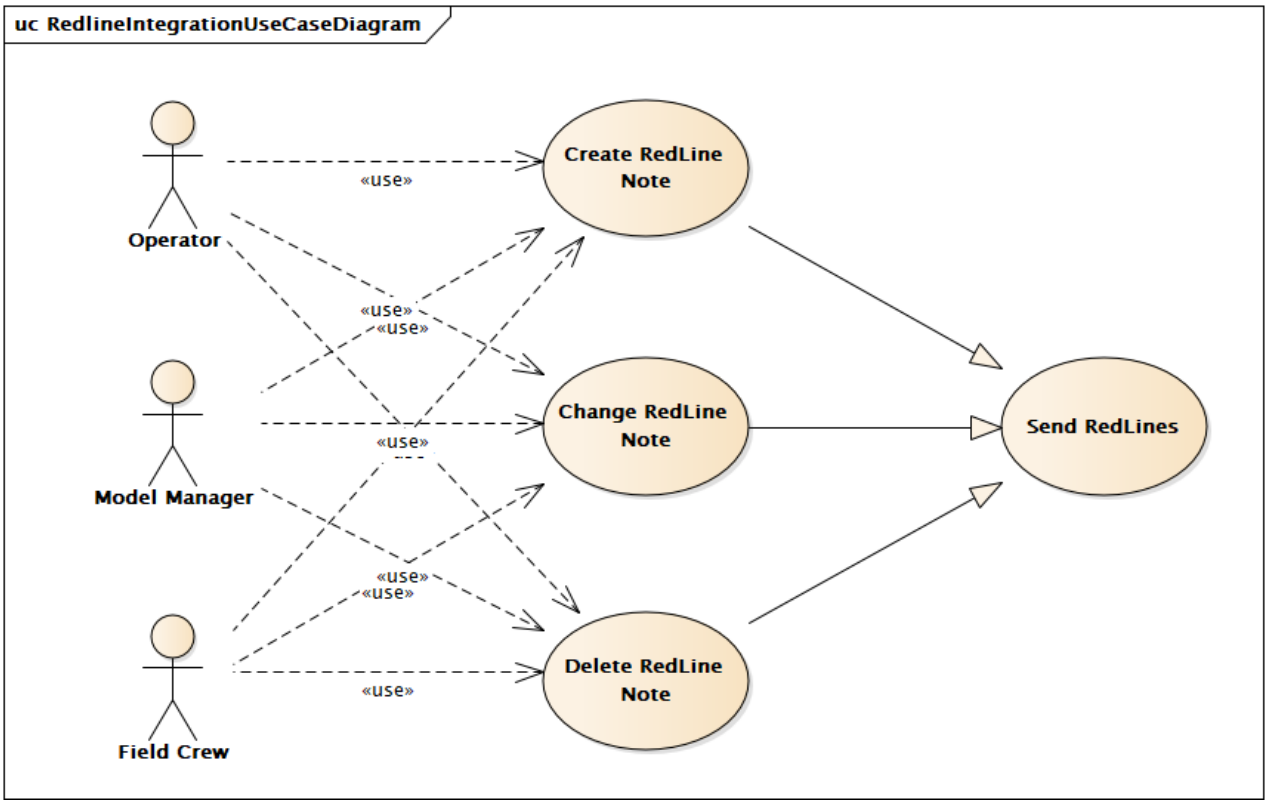

Figure 4.1 – Redline integration use case diagram

5. SENDREDLINESSERVICE

5.1. Overview

In case when additional correction needs to be performed on some electrical equipment or landbase element in the source system (typically GIS), EcoStruxure GridOps users (operators, model managers, field crews, etc.) have an ability to add redline note to the given equipment/element. During the redline note addition, users can specify comment describing the identified issue. Along with the comment, users can also provide appropriate attachment (e.g. picture) which will clarify requested correction in more detail. Every change (insert, update, delete) made to the redline note model will trigger appropriate action in RLN Adapter, which results in sending of redline message to dedicated web service (*SendRedlinesService*):

- *CreatedRedlines operation* – when redline note is created (added), RLN adapter receives a message and performs an initial data validation of the received data. If everything is valid, *CreatedRedlinesEvent* message is formed and corresponding operation invoked on the externally-hosted web service.
- *ChangedRedlines operation* – update of the existing redline note (comment or attachment) will result in another message publication to the RLN Adapter. Similar approach as above takes place; RLN adapter validates the received data and creates *ChangedRedlinesEvent* message. The created message is forwarded to the *SendRedlinesService* hosted on a client's side.
- *DeletedRedlines operation* – when certain correction is made in the source system or it is no longer necessary, users can remove the appropriate redline note. Upon removal, predefined message is published to the RLN adapter. The received message is then forwarded to the dedicated web service.

Source system for redline notes created in DMZ and Core system will be the CORE system. While for redline notes created in the Staging system, mentioned system will be the owner. Redline notes can be changed/deleted only in the system in which they were created. Change to the redline note model will be propagated to the RLN Adapter from the CORE or Staging systems. However, all redline notes from the DMZ and CORE systems will be replicated to the Staging system.

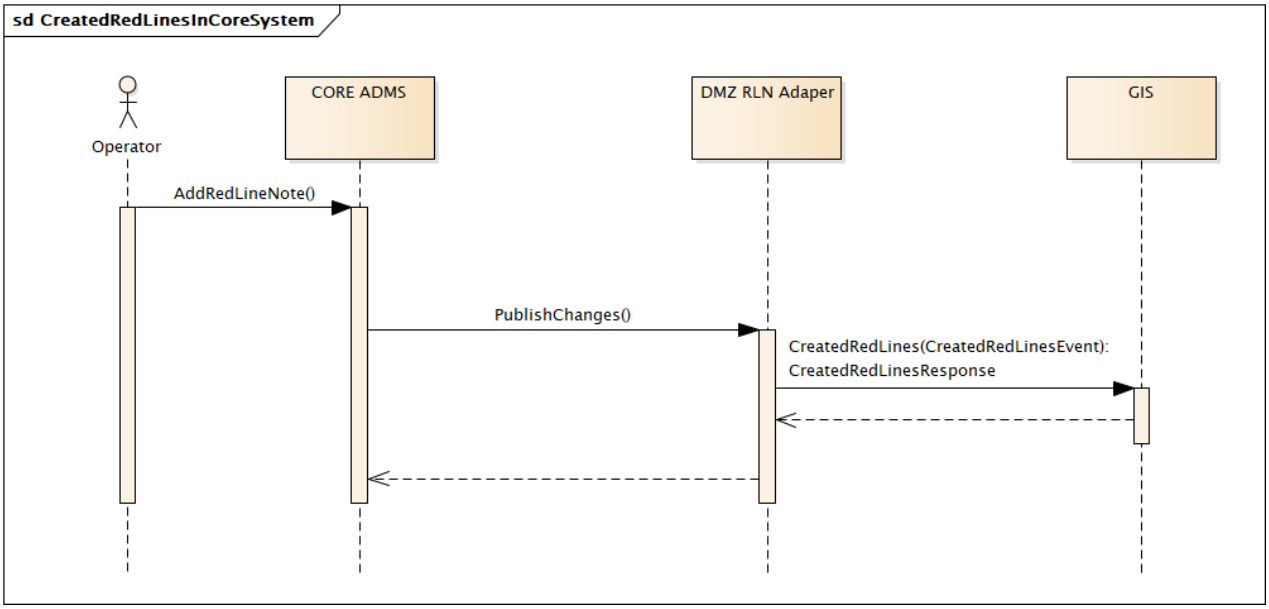

Figure 5.1 – Creation of a redline note in Core system

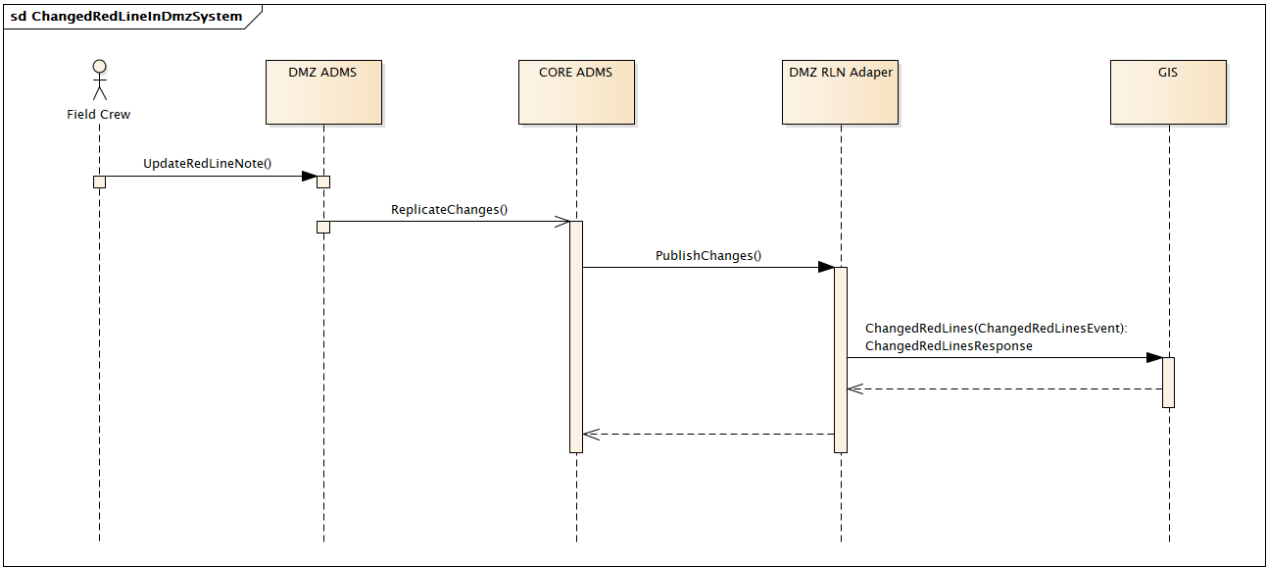

Figure 5.2 – Update of the redline note in DMZ system

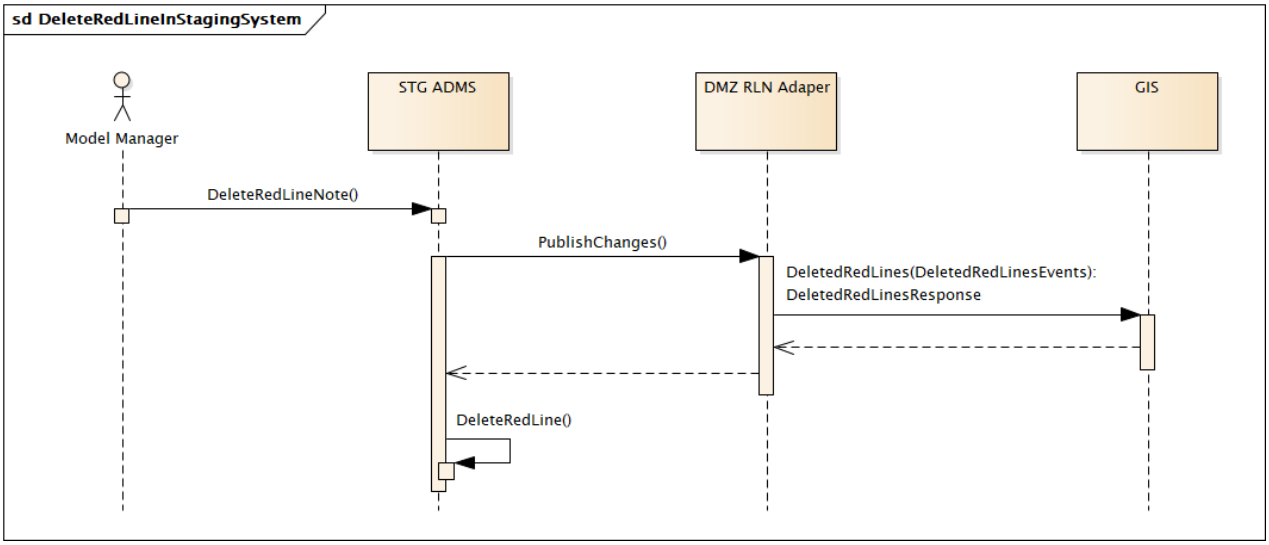

Figure 5.3 – Deletion of the redline note in Staging system

5.2. Use Cases

The list of possible use cases and corresponding faults is given in following table:

Table 5.1 – The SendRedlines web service's operation use cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Send Redlines – message is sent successfully	Result	String	OK	The redline note is created/changed/deleted in any of the EcoStruxure GridOps systems. Message is received, validated and forwarded by the RLN adapter to the external system. After the OK response is received from the external system, appropriate redline note is marked as sent.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Send Redlines – external system (web service) is unavailable	Result	String	N/A	The redline note is created/changed/deleted in any of the EcoStruxure GridOps systems. Message is received, validated and forwarded by the RLN adapter to the external system. However, if for some reason external system (web service) is unavailable or it cannot process the message, external system must return either fault message of failed response. Based on the fault message or failed response, RLN adapter will not mark that redline as sent. Next time, when the new redline note is created/changed/deleted, the system will send all the redline notes that were marked as not sent.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	

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”. Implementations should treat deprecated verbs “update” and “updated” as synonyms to “change” and “changed”.
- *Noun* – to identify the subject of the action and/or the type of the payload, such as the Redlines, etc.

The field that can be optionally supplied includes 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.

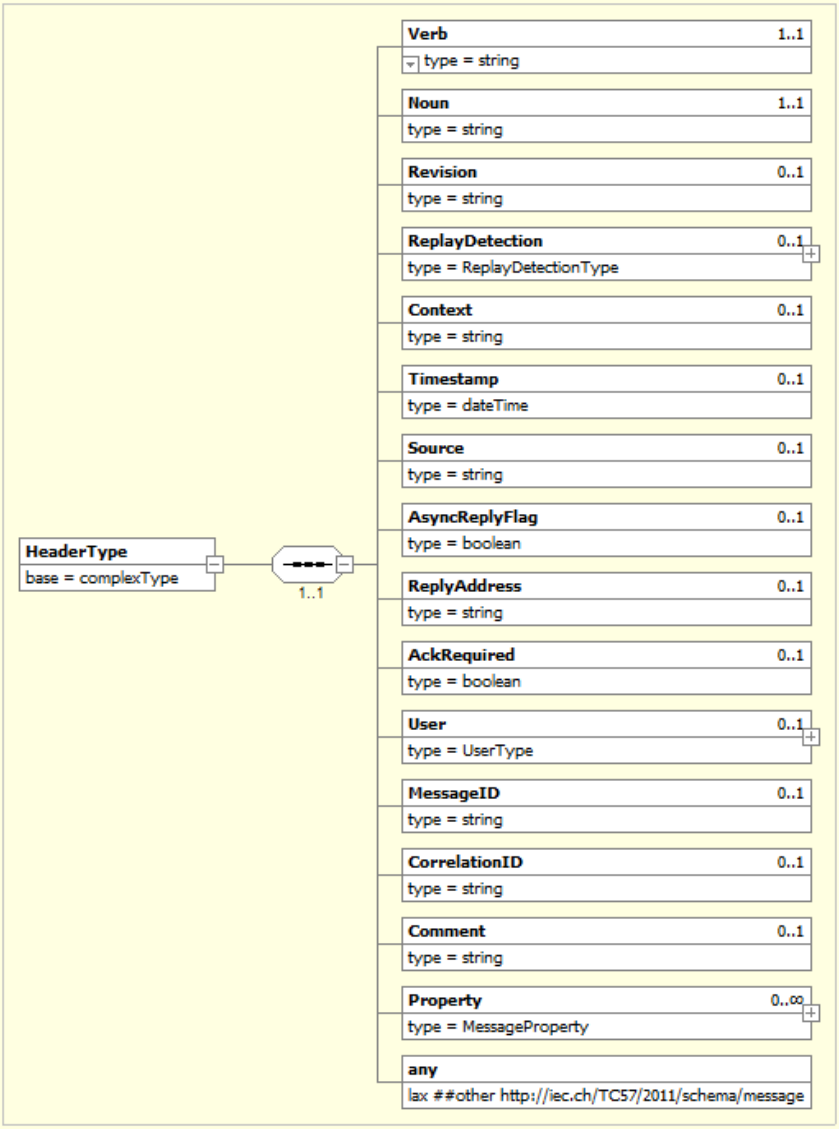

Figure 6.1 – The header field

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 the **Reply** and **Error** fields are presented in Figure 6.2.

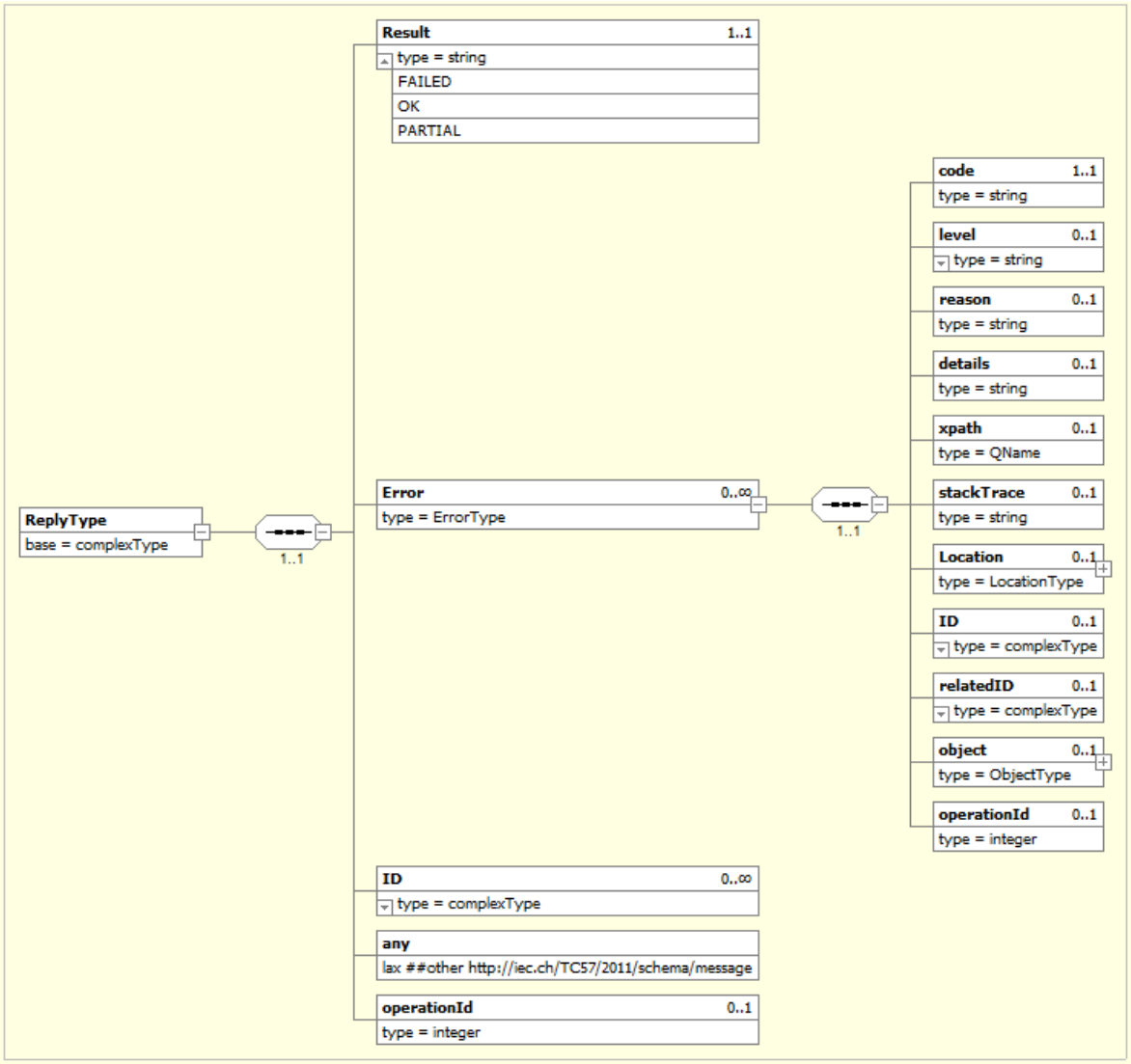

*Figure 6.2 – The **Reply** and **Error** field contents*

6.2. SendRedlinesService Operations

The operation definition:

CreatedRedlinesResponse CreatedRedlines(CreatedRedlinesEvent)

ChangedRedlinesResponse ChangedRedlines(ChangedRedlinesEvent)

DeletedRedlinesResponse DeletedRedlines(DeletedRedlinesEvent)

6.2.1. Request

6.2.1.1. SOAP

The *CreatedRedlines, ChangedRedlines* and *DeletedRedlines* event messages are defined according to the IEC 61968-100 and contain the following two sections:

- Header
- Payload

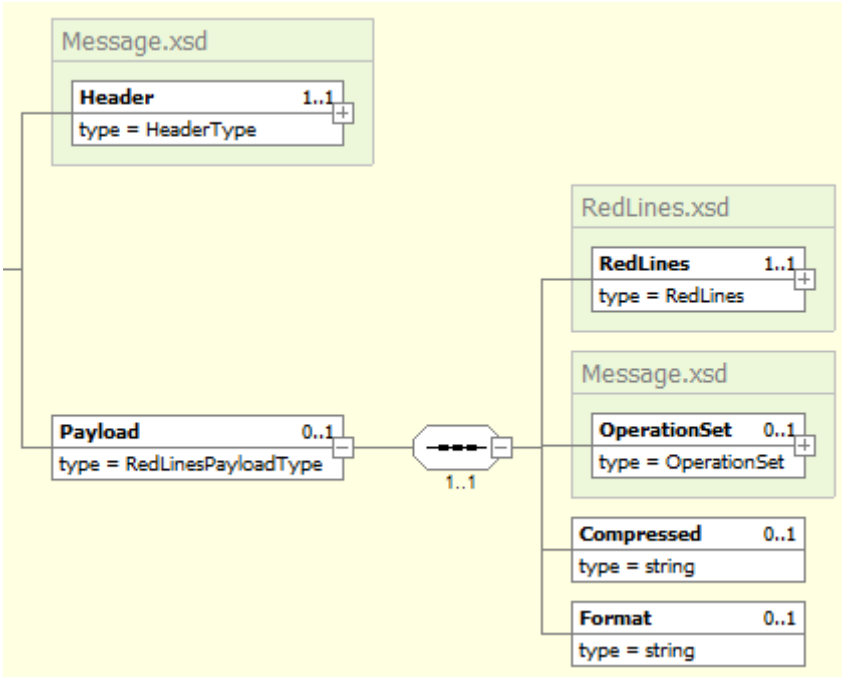

Figure 6.3 – RedLinesMessage.xsd

The Payload section of every message carries the CIM defined profile (*Redlines.xsd*). The visual representation of the *Redlines.xsd* schema is given in following figure.

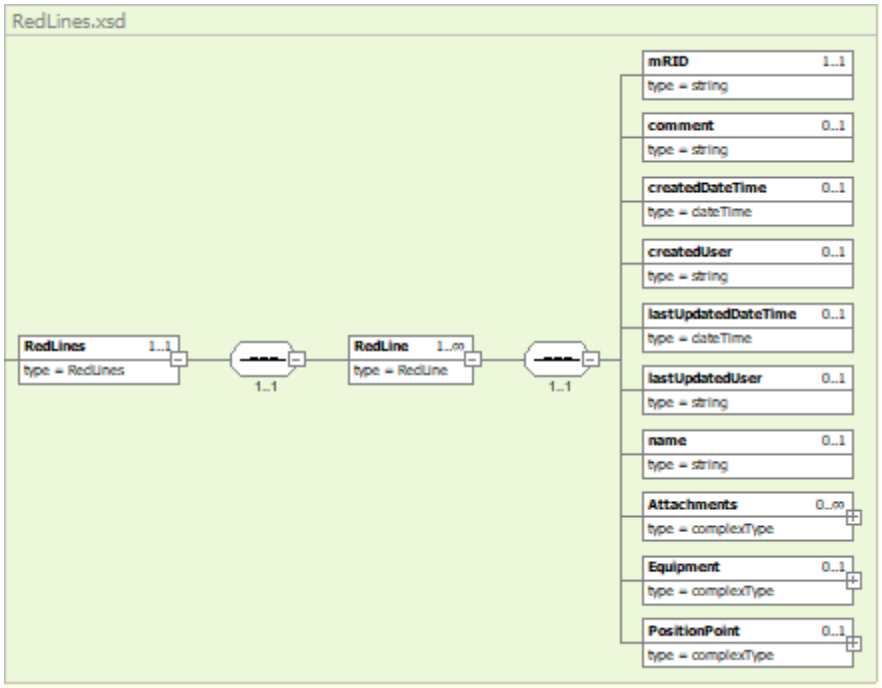

Figure 6.4 – RedLines.xsd

Following table defines the mapping between the *RedlinesMessage.xsd* and appropriate entities in the note model.

Table 6.1 – SOAP Redlines message → Note model mapping

Redlines message			Description	Note model	
Section	**Property**	**Type**		**Property**	**Type**
Header	**Verb**	String	The identifier for a specific action to be taken. For this message, the Verb is created/changed/deleted.	Populated by RLN Adapter	N/A
Header	**Noun**	String	The identifier for the subject of the action and/or the type of the payload. Noun is Redlines.	Populated by RLN Adapter	N/A
Header	**Timestamp**	DateTime	The timestamp when the message was produced. Example: 2015-12-31T12:34:56+02:00	Populated by RLN Adapter	N/A
Header	Source	String	The source system or application that sends the message. For this message, the Source is EcoStruxure GridOps .	Populated by RLN Adapter	N/A
Header	**MessageID**	String	The unique message ID to be used for tracking messages.	Populated by RLN Adapter	N/A
Header	CorrelationID	String	Correlation ID.	Populated by RLN Adapter	N/A
Payload	mRID	String	The unique identifier of the redline note.	CustomID	String
Payload	comment	String	Comment which describes necessary activity.	Comment	String
Payload	name	String	Optional name of the redline note.	N/A	N/A
Payload	createdDateTime	DateTime	Creation timestamp.	CreationTimestamp	DateTime
Payload	createdUser	String	System (Core, Staging, Dmz) user that created the redline.	CreationUserID	String
Payload	lastUpdateDateTime	DateTime	Last update timestamp.	Timestamp	DateTime
Payload	lastUpdateUser	String	System (Core, Staging, Dmz) user that updated/deleted the redline.	UserID	String
Payload	Attachment.content	String	Content serialized as base64string.	NoteAttachment	ByteArray
Payload	Attachment.format	String	Format of the attachment (xml, jpeg, png, etc.).	NoteAttachmentFormat	String
Payload	Attachment.name	String	Name of the attachment.	NoteAttachmentName	String
Payload	Equipment.mRID	String	CustomID (unique identifier) of the equipment to which the redline note was added.	NoteElementCustomID	String
Payload	PositionPoint.xPosition	String	X (longitude) coordinate of the redline.	NoteCoordinateX	Float
Payload	PositionPoint.yPosition	String	Y (latitude) coordinate of the redline.	NoteCoordinateY	Float

6.2.1.2. REST (JSON)

Following table defines the mapping between the *JSON message* and appropriate entities in the note model.

Table 6.2 – JSON Redlines message → Note model mapping

Redlines message		Description	Note model	
Property	Type		Property	Type
Name	String	Name of added element for which redline request is be sent	NoteElementName	String
Red Line ID	Guid	Redline unique identifier	Populated by RLN Adapter	String
Element Custom ID	String	CustomID (unique identifier) of the equipment to which the redline note was added.	NoteElementCustomID	String
location.X	Double	X (longitude) coordinate of the redline.	NoteCoordinateX	Float
location.Y	Double	Y (latitude) coordinate of the redline.	NoteCoordinateY	Float
Comment	String	Comment which describes necessary activity.	Comment	String

6.2.2. Response

After the redline is sent, the appropriate response is returned from the external system in the form of *RedlinesResponse* message. The content of the response message is given in Figure 6.3. Content of the *RedlinesResponse* message is given in the following table.

Table 6.3 – The Redlines Response message content

Redlines Response message			Description
Section	Property	Type	
Header	**Verb**	String	The identifier for a specific action to be taken. Verb is reply.
Header	**Noun**	String	The identifier for the subject of the action and/or the type of the payload. For this message, the Noun is Redlines.
Header	**Timestamp**	DateTime	The timestamp when the message was produced. Example: 2015-12-31T12:34:56+02:00
Header	Source	String	The source system or application that sends the message. For this message, the Source can be GIS, etc.
Header	**MessageID**	String	The unique message ID to be used for tracking messages.
Header	CorrelationID	String	Correlation ID that is equal to the request message ID.
Reply	**Result**	String	Returned as part of synchronous response. The valid values are: OK or FAILED.

6.2.3. Fault

The content of the *RedlinesFault* message is given in Figure 6.3.

7. DEPLOYMENT SPECIFICATION

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

The deployment specification is provided in the following table:

Table 7.1 – The deployment specification

Deployment Specification	
Application	RedlineAdapter
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

8. INTERFACE CONFIGURATION

RLN 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 RLN adapter.

Initially, following configuration files are used the adapter:

Table 8.1 – The configuration files specification

Name of the config file	Configuration File Description
AdapterRln	Registry configuration xml file
AdapterRedline_WebServiceConfiguration	Web service configuration xml file

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 Redlining Interface.zip* file [4].

9. APPENDIX

9.1. WSDL

The WSDL file, XSD schemas and sample messages defined according to the IEC 61968-100 standard for all RLN web services are provided within the *Web Service Definitions* folder in the *EcoStruxure GridOps Management Suite 3.10 Redlining 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 Redlining Interface.zip* file [4].

9.3. Operating with Redline Notes

The model manager (operator, filed crew, etc.) has the ability to create Redline Note on a given network element. This action is performed in the DMD application connected to RT context in DMZ, STG or Production. After creation, Redline Note will automatically be sent to external system (GIS), so that GIS personnel performs additional correction.

Redline note is created in following way:

1. In corresponding Network view user need to find desired network element, right click on it and select “Place Note” option, as highlighted in Figure 9.1.

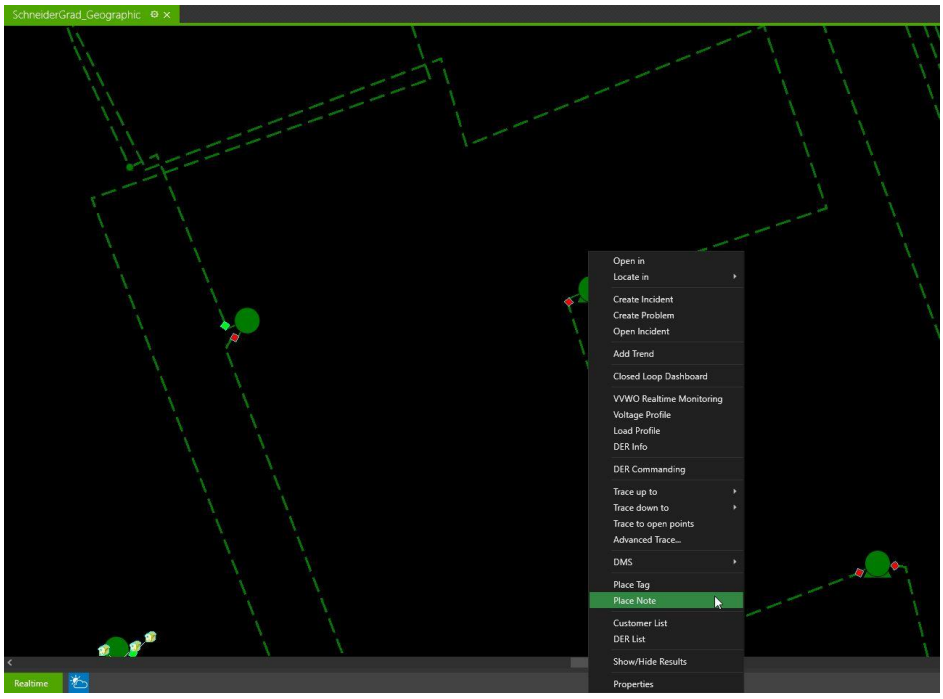

Figure 9.1 – Finding desired equipment on network view

2. As a result of the action a dedicated UI control is presented, dedicated for creating a note and associating it to a specified network element. User needs to select Redline Note option from the

dropdown menu, enter a short description in the comment section or provide appropriate attachment. By clicking on the OK button, Redline Note is created. Described process is presented on Figure 9.2.

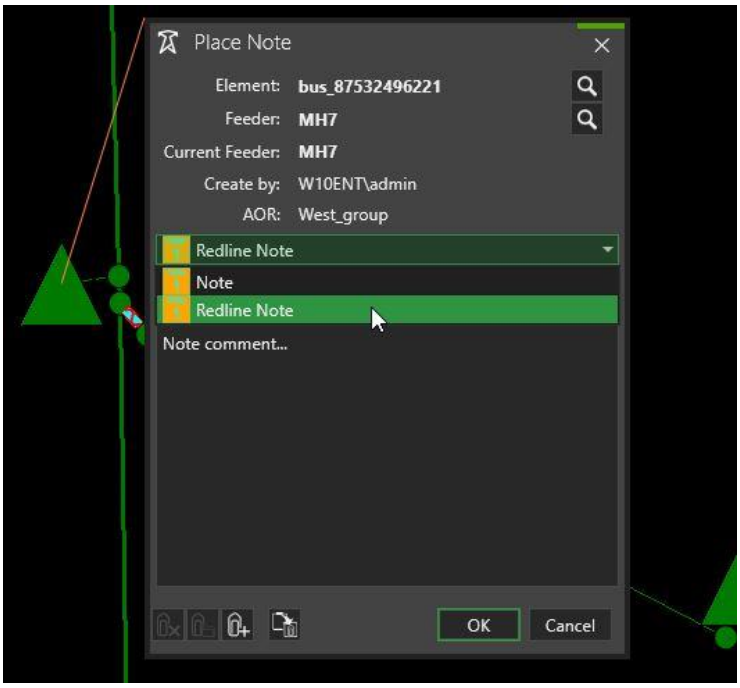

Figure 9.2 – Place Note UI control

3. Newly created redline note is visible on the network element as well as in the Note Summary window, as shown on Figure 9.3. Upon creation, RLN adapter receives this notification and performs initial data validation. If everything is valid, *CreatedRedlinesEvent* message is formed and sent to GIS system.

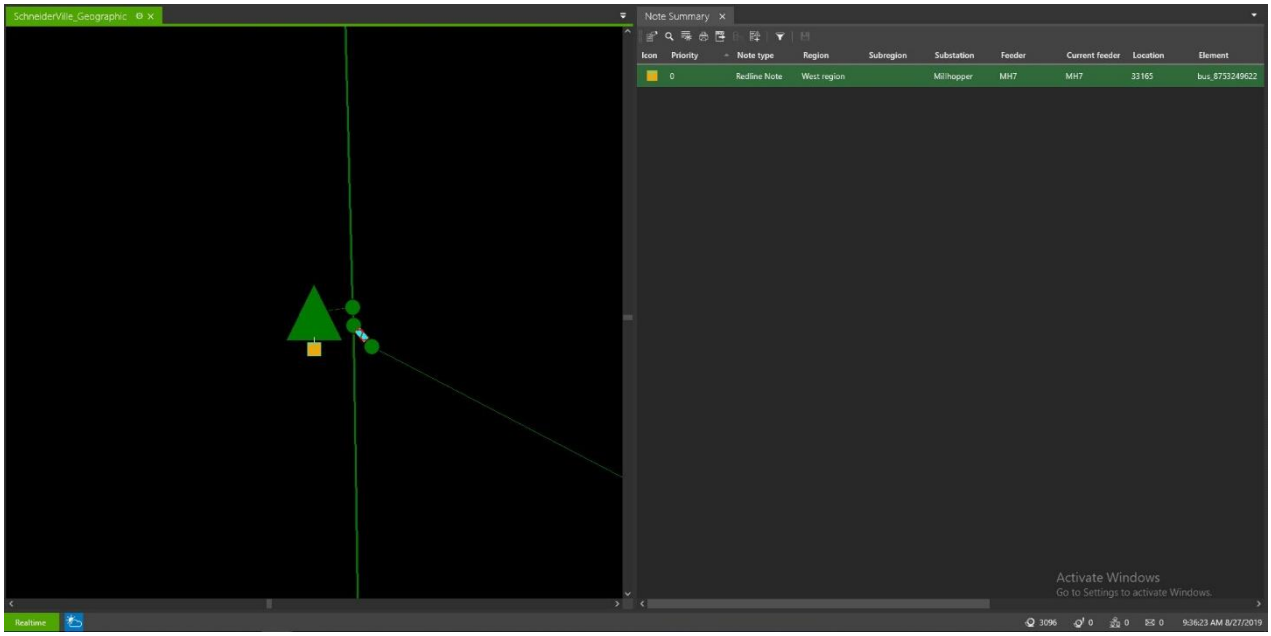

Figure 9.3 – Note summary view

Once added Redline note can also be modified by model manager, operator or filed crew:

1. Redline note can be modified by opening Note Detail window, updating comment or attachment (Figure 9.4) and clicking on OK button.

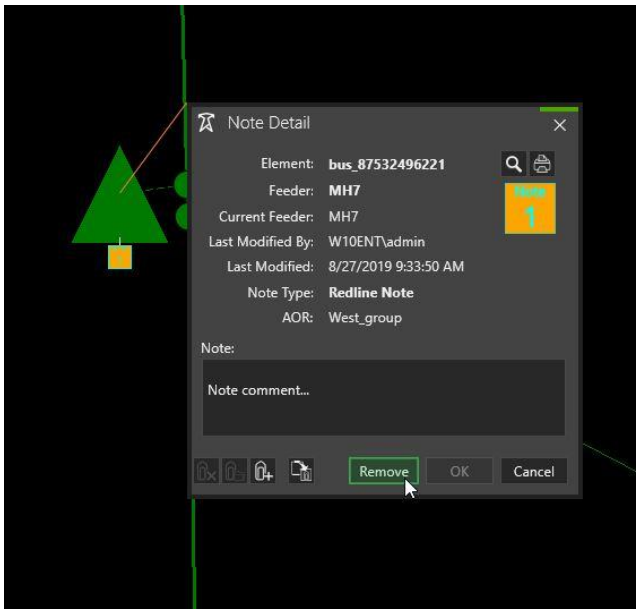

Figure 9.4 – Note Detail UI Control

2. After redline note is updated RLN adapter receives this notification and performs an initial data. If everything is valid, *ChangedRedlinesEvent* message is formed and sent to GIS system.

After certain correction is made in the GIS system or it is no longer necessary, users can remove the appropriate redline note:

1. Open Note Detail window, and click on Remove button (Figure 9.4).
2. Upon removal, predefined message is published to the RLN adapter. The received message is then forwarded to external system.

9.4. Business Use Cases

The following table lists just some of the business (end-2-end) use cases that can be implemented with the help of Redline interface:

Table 9.1 – Business use cases

Use Case Name	Use Case Description	Business Value
Invalid extract exported from GIS system	• Invalid GIS extract is prepared by GIS personal • Model manager imports previously prepared invalid extract • Model manager inspects import reports and locates part of the network for which feeder extract contains inconsistency. • In order to correct those data inconsistencies in the source (GIS) as soon as possible, the model manager places redline notes on specific elements in the network model. To provide more detail to GIS personnel model manager attaches report and writes appropriate description. • Redline note is sent through the dedicated CIM (IEC 61968) compliant Redline (RLN) interface to any GIS system that supports the redlining concept.	During the data transformation, unexpected issues may occur due to invalid data according to CIM validation rules. Using the Redline interface, precise and detailed information about detected issues can be provided to the GIS department. In that way, efficiency is increased, by easing communication between EcoStruxure GridOps and GIS personal.
Feeder changeset is rejected	• Incremental feeder extract that contains a deletion of an element on which there is attached TE is prepared by GIS personal. • Model manager imports previously prepared extract • Changeset goes to the REJECTED state. • The model manager inspects import reports and locates part of the network for which feeder extract contains inconsistency. • In order to correct those data inconsistencies in the source (GIS), the model manager places redline notes on elements to which a temporary element is attached to. To provide more detail to the GIS personnel model manager can attach a report and write appropriate description.	During the data transformation, unexpected issues may occur due to invalid data according to CIM validation rules. Using the Redline interface, precise and detailed information about detected issues can be provided to the GIS department. In that way, efficiency is increased, by easing communication between EcoStruxure GridOps and GIS personal.

Use Case Name	Use Case Description	Business Value
	• Redline note is sent through the dedicated CIM (IEC 61968) compliant Redline (RLN) interface to any GIS system that supports the redlining concept.	
During inspection on the field, Field crew detect inconsistence in network	• During a regular inspection on the field crew can detect some inconsistencies in the network, e.g. field crew identifies a device in the field, outside of the substation, that is different than what's on the map • Using FC (Field Client) application, the crew places Redline note on the disputed device on geographic view. The crew can attach an appropriate picture as an attachment on a redline note and writes an appropriate comment. • Redline note is visible in the control room • Redline note is sent through the dedicated CIM (IEC 61968) compliant Redline (RLN) interface to any GIS system that supports the redlining concept.	In a situation when crew detects some inconsistence in the field, during network model maintenance, model managers in EcoStruxure GridOps and GIS can be quickly informed by using redline functionality. Using FC (Field Client) application, the crew on the filed can place a redline note on the disputed device. The crew can attach an appropriate picture as an attachment on redline note and write an appropriate comment. This can improve situational awareness and enables prompt correction of data inconsistencies in the source (GIS).
Operator detects miss linked customers	• The operator gets a miss-linked call from a customer that is off but not included in the incident. The operator figures out that it is the wrong customer-transformer connection and adds this customer to the incident. • The customer is recorded as affected by the incident, but the crew discovers the customer has supply. It means customer is connected to the wrong transformer in the data model. The operator removes customer from the incident. • The crew places a redline note to the service location indicating that transformer connectivity needs to be corrected in GIS.	If any irregularities are detected on the network model after feeder import, GIS personal will be simply and quickly informed by using Redline functionalities. After network model inspection, Model manager detects some inconsistence in the network and places redline note on specific element. Redline note with precise and detailed information is propagate to GIS. This speed up correction process, reduces cost and unnecessary communication between EcoStruxure GridOps and GIS personal.

10. RELEASE NOTES

The following new features related to the RLN Interface were introduced in the software, starting from version 3.8 SP1

10.1. Software Version 3.8 SP1 HF2

Feature	Description
Redlining Integration – ArcFM RESTful Interface Support	Along with the SOAP-based web service, out-of-the-box support for Schneider Electric ArcFM Redlining Feature service is added. Communication between the Redlining Adapter and ArcFM Redlining Feature Service is performed using RESTful web services.

11. DEFINITIONS AND ABBREVIATIONS

Definition/Abbreviation	Description
ADMS	Advanced Distribution Management System
CIM	Common Information Model
Changeset	Changeset contains delta as a way to define what changes may be made to the EcoStruxure GridOps model, but it also contains additional information needed for correct handling of those changes. The changeset can be created as a result of the import procedure, or manually by using the Network Builder.
Core	Production Execution System (Zone)
DMZ	Demilitarized Zone
Extract	File (typically in CIM/XML format, in some cases .CSV) prepared by a source system to be used in EcoStruxure GridOps. The file is retrieved from the distributed location and stored locally. When the file is stored in CSRepo, the additional information about the file is associated: extract type (GIS feeder, patch, EMS, ESCADA, etc.), time of storing, current status. In the context of CSRepo, extract represents not only content of the retrieved file, but also all additional info.
GDA	Generic Data Access
GIS	Geographic Information System
SOAP	Simple Object Access Protocol
WCF	Windows Communication Foundation
WS	Web Service
XML	Extensible Markup Language