GridOps Management Suite 3.10

Outage Management Notification 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 Enterprise Integration Platform - Functional Specification	The document represents a set of common integration principles applied to all baseline integration adapters.
2.	EcoStruxure GridOps Management Suite 3.10 Outage Management - Functional Specification	The document describes the Outage Management component. The Outage Management functionality tracks all information about the power disturbances in the network and organizes the response to the disturbance into a user-friendly, efficient and safe workflow.
3.	EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface	EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface zip file contains essential configuration information, as well as web service definitions complemented with message examples.

2. ASSUMPTIONS

The Outage Notification integration is designed under the following assumptions:

- EcoStruxure GridOps system is the leading system responsible for entire life cycle of all outage management entities (trouble tickets, incidents, hazards, callbacks, etc.).
- Message exchange is supported utilizing publish/subscribe integration pattern.
- Since the outage management functionality is highly configurable, appropriate types (enumerations such as hazard codes, outage codes, incident codes, trouble codes, trouble ticket sources, callback results, etc.) that need to be exchanged between the external systems and EcoStruxure GridOps must be defined during design sessions.
- Configurable coordinate conversion is possible for entities which encompass coordinates (incidents, incident usage points, etc.).

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 the following solutions: EcoStruxure ADMS, EcoStruxure Grid Operation and EcoStruxure Energy Transmission Operation.

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.

In certain situations, EcoStruxure GridOps service consumers want to be notified in near real-time about changes being made to incident model. Such need may come from various regulatory requirements related to proactive communications and customer satisfaction.

The Outage Management Notification (ONT) Interface is designed to provide relevant information about changes in incident model to other systems, in near real-time. Such functionality enables utilities to support various business processes and day-to-day activities. Integration is designed “one-way”, which means that flow of data is always from EcoStruxure GridOps towards external system.

Changes in incident model data, to be sent to the external system, are divided into two separate operations. Data about incidents, and data about incident affected customers. Usually those changes are related to restoration time estimation, number of affected customers, incident state transitions, etc. Data are sent via invoking corresponding externally hosted web services.

3.1. General Architecture

Described in the *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification* [1].

4. INTERFACE OVERVIEW

The Outage Notification integration is implemented through the ONT Adapter component. The aforementioned adapter implements SOAP based Web Service Clients with appropriate set of operations:

- SendIncidents – used for sending incident related data to corresponding external web service:
  - CreatedIncidents operation
  - ChangedIncidents operation
  - DeletedIncidents operation
- SendIncidentUsagePoints – used for sending incident usage points to corresponding external web service:
  - ChangedIncidentUsagePoints operation
- SendIncidentTroubleTickets – used for sending incident trouble tickets to corresponding external web service:
  - ChangedIncidentTroubleTickets operation
- Integrity update – used for systems alignment. Uses existing service clients for invoking following external web service operations:
  - ChangedIncidents operation
  - ChangedIncidentUsagePoints operation
  - ChangedIncidentTroubleTickets operation

The following chapters provide more details regarding these interfaces (web service clients) and appropriate web service operations, data mappings (CIM Profiles → Outage Model), error handling scenarios etc.

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

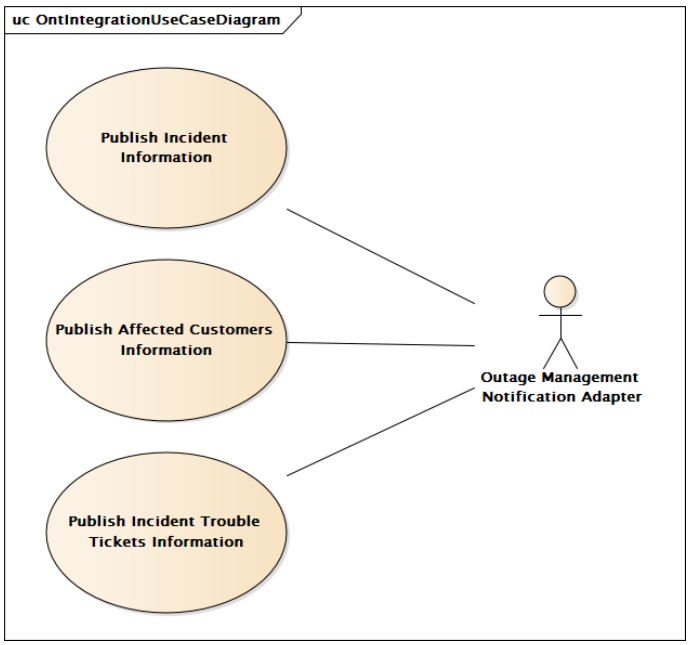

Figure 4.1 – The ONT Integration use case diagram

5. INTEGRATION INTERFACES

5.1. SendIncidents Service

5.1.1. Overview

ONT adapter is subscribed to changes of incident data. When one of the defined triggers occurs, adapter handles the publication and forms a request message based on the type of the operation that relates to incident creation, update, or deletion. ONT adapter performs initial data validation, based on the adapter configuration file, to determine whether the change is relevant and should be processed. When the publication does not meet the defined constraints requirement, adapter flags the publication as invalid and dumps it to adapter log file. Corresponding request message is later used to invoke external SendIncidents service.

- CreatedIncidents – when incident is created, adapter receives a message flagged with Insert operation type. If that message satisfies defined filtering criteria, described in Message Filtering, appropriate CreatedIncidentsEvent message is formed. Otherwise, first message with Update operation type that satisfies the filtering criteria will be treated as CreatedIncidentsEvent. Corresponding CreateIncidents operation on SendIncidents service is invoked upon message creation. CreatedIncidentsEvent message is the same as the other two, content-wise, it just uses a different envelope to differentiate the operation type. Sequence diagram is shown.
- ChangedIncidents – updating incident detail is common during the incident lifecycle. Properties like ETR, affected customers, status, etc. succumb often to changes. Those changes are often of interest, and the user wants to be notified when they do change. When the change occurs, ChangedIncidentsEvent message is formed and corresponding ChangedIncidents operation is invoked on SendIncidents service. ChangedIncidentsEvent message is the same as the other two, content-wise, it just uses a different envelope to differentiate the operation type. Sequence diagram is shown on Figure 5.2.
- DeletedIncidents – upon resolution, incident is closed and archived. When incident is archived, it is deleted from cache and stored in Operations database. Upon deletion, adapter component receives a notification and forms DeletedIncidentsEvent message. Corresponding DeletedIncidents operation is invoked on SendIncidents service. DeletedIncidentsEvent message is the same as the other two, content-wise, it just uses a different envelope to differentiate the operation type. Sequence diagram is shown on Figure 5.3.

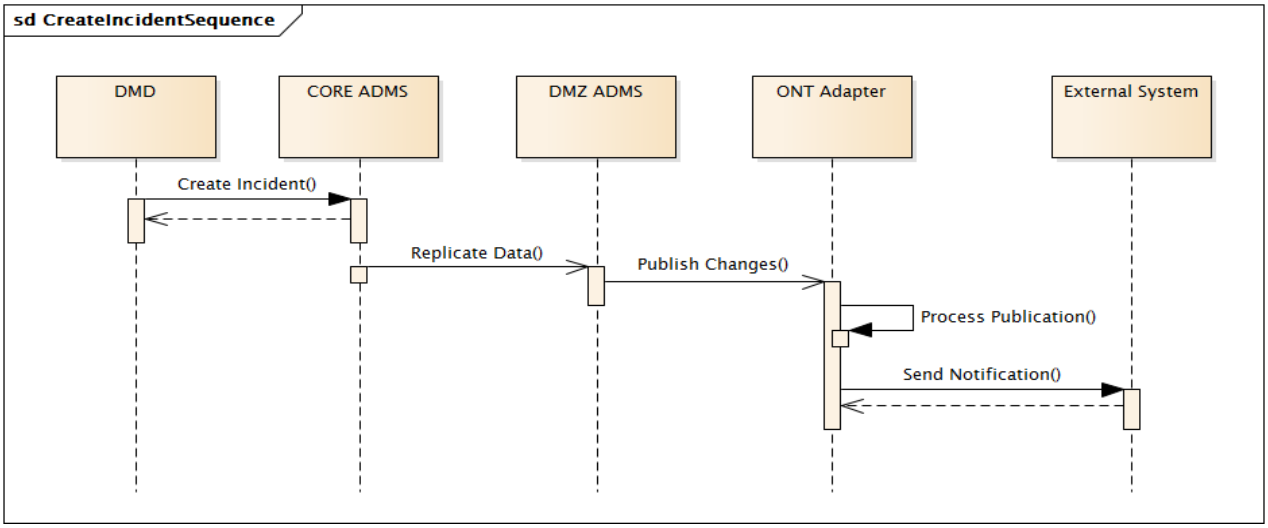

Figure 5.1 – Created Incidents - Sequence Diagram

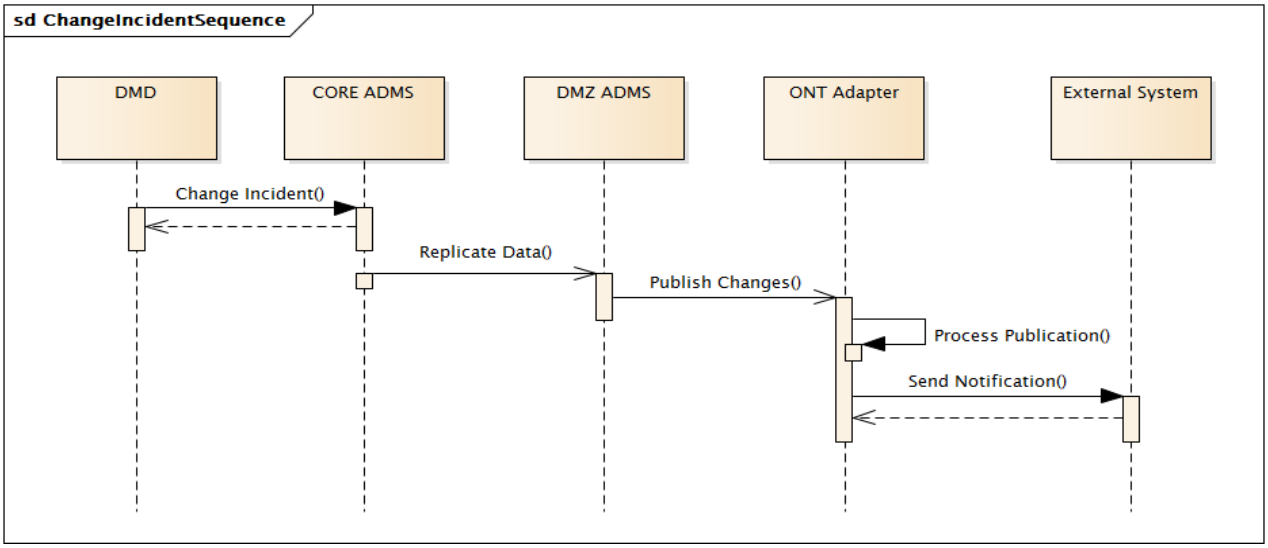

Figure 5.2 – Changed Incidents - Sequence Diagram

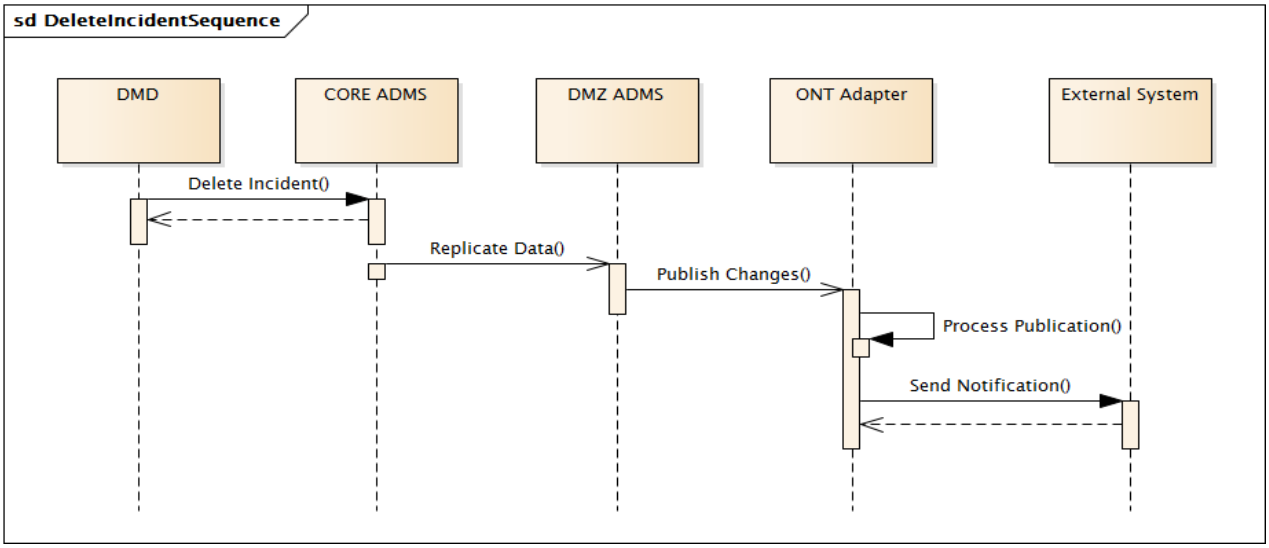

Figure 5.3 – Deleted Incidents - Sequence Diagram

In some occasion, a communication failure can occur between the external system and EcoStruxure GridOps. In those situations, adapter has an implemented queueing logic that preserves the message until communication with external system is restored. Queueing the messages preserves the time order.

5.1.2. Triggers

Adapter supports sending messages only for event-based triggers. Adapter listens to changes of incident model. When a change of interest occurs, adapter processes the received publication, queries for additional business process related data, and populates corresponding notification message. Following events will trigger creation of aforementioned messages.

- New incident is created – event triggers the CreatedIncidents operation.
- Existing incident is deleted – event triggers the DeletedIncidents operation.
- ChangedIncidents operation is invoked when one of the following incident attributes is changed:
  - Incident status – incident status transitions to a new one. Incident status trigger is highly configured within adapter configuration file. A list of supported incidents statuses is defined there.
  - Confirmation status – upon incident confirmation.
  - Problems – when a problem is added to an existing incident.
  - Comment – new note is added to an existing incident.
  - Resolution fields – upon incident resolution, resolution fields (cause, subcause, data problem, etc.) are modified to specify the source of the issue.
  - Incident Device – commonly changed during the Complex Incident Scenarios.
  - Affected customer count – commonly changed during the Complex Incident Scenarios.
  - Affected critical customer count – commonly changed during the Complex Incident Scenarios.
  - Unrestored customers count – commonly changed during the Complex Incident Scenarios.
  - Unrestored critical customers count – commonly changed during the Complex Incident Scenarios.
  - Estimated restoration time – commonly changed by crew on field.

- Actual restoration time – the timestamp when incident transitions to Restored.
- Crew assignments – when a crew is assigned to an existing incident.

It should be noted that triggers for ChangedIncidents operation are highly configurable within adapter configuration file. Abovementioned triggers are considered as default.

5.1.3. Use Cases

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

Table 5.1 – The SendIncidents service operations use cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Supported common use cases are described in				
Invalid Verb	Result	String	FAILED	ONT adapter sends the request message with invalid header Verb. Response message is sent from external system with FAILED result and the message is discarded. Valid verbs are: “created”, “changed”, “deleted”. Verbs must correlate the envelope that describes incident operation type.
	Error.code	String	1.2	
	Error.level	String	FATAL	
	Error.reason	String	InvalidVerb	
	Error.details	String	Invalid verb: {0}	
Invalid Noun	Result	String	FAILED	ONT adapter sends the request message with invalid header Noun. Response message is sent from external system with FAILED result and the message is discarded. Valid noun for incident operations is “Incidents”.
	Error.code	String	1.2	
	Error.level	String	FATAL	
	Error.reason	String	InvalidNoun	
	Error.details	String	Invalid Noun: {0}	
Mandatory Element Missing	Result	String	FAILED	ONT adapter sends request message in which some of the mandatory elements are missing. Response message is sent by external system 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 request	Result	String	FAILED	
	Error.code	String	5.3	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.level	String	FATAL	Adapter sends request message, but for some reason message processing fails in external system due to an internal server error. Fault response message is sent by external system component
	Error.reason	String	InternalServerError	
	Error.details	String	Unable to process the request message with ID: {0} - transaction attempted and failed	
ESB is not available	Result	String	N/A	EcoStruxure GridOps system sends request message to the external system but it failed due to unavailability.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Failover event	Result	String	N/A	EcoStruxure GridOps tries to send request messages to the external system which is unavailable. Adapter has a configurable retry mechanism and should retry sending in case of external system unavailability. In case a failover event occurs while adapter is in retry regime, STBY adapter instance resumes the where the formerly HOT adapter has stopped. All messages that were pending for sending, are transferred to new HOT adapter instance via DMS Replication service.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	

5.2. SendIncidentUsagePoints Service

5.2.1. Overview

During the process of incident resolution, affected customer count tends to change. Incidents can go through a partially restored phase, where a certain amount of customer get power back. Incident can be merged with another incident and naturally its affected customers also transition to that incident. Adapter is subscribed to all changes that correspond to affected customers, incident that affects them, their outage and restoration periods. Upon receiving the notification, adapter performs initial validation based on adapter configuration file and creates ChangedIncidentUsagePointsEvent message used for invoking ChangedIncidentUsagePoints operation on the external SendIncidentUsagePoints service.

ChangedIncidentUsagePointsEvent message payload is modeled so that it groups the affected customers to its related incident. Main entity within message payload is Incident. It contains a list of all related affected usage points with additional relevant information which includes outage periods, customer detail, etc. Messages are sent separately for each incident. In case where publications are processed for UsagePoints that are related to different incidents, more than one ChangedIncidentUsagePointsEvent message will be created and sent to corresponding external SendIncidentUsagePoints service. Incident constraints are defined in the adapter configuration file. Upon receiving the event, adapter queries for related incident data to determine whether the incident in question is supported for processing. When the publication does not meet the defined constraints requirement, adapter flags the publication as invalid and dumps it to adapter log file.

5.2.2. Triggers

Adapter supports sending messages only for event-based triggers. Adapter listens to changes of UsagePoints model. When a change of interest occurs, adapter processes the received publication, queries for additional business process related data, and populates corresponding notification message. Following events will trigger the creation of aforementioned message.

- Usage point incident reference change – this trigger follows three cases:
  - UsagePoint is referenced to a newly created incident.
  - UsagePoint incident reference is updated due to complex incident management scenarios.
  - UsagePoint is dereferenced from an incident.
- UsagePoint outage periods change (outage start and end times).
- ETR of the corresponding incident.

It should be noted that triggers for ChangedIncidentUsagePoints operation are highly configurable within adapter configuration file. Abovementioned triggers are considered as default.

5.2.3. Transferring Large Messages

Depending on the implementation and configuration of the SOAP web service, limitation of the message size may exist. In case of the large incidents, e.g. when outage is originated in the sub-transmission, number of the customers affected by an outage may be very large. This can result that the message with the affected customers becomes too large to transfer via the SOAP.

Outage Notification Adapter will divide large message into multiple smaller ones. Split will be performed when the number of the affected customers in the message exceeds the predefined data limit stated in the web service configuration, max message size parameter.

For the external system to be aware of the message number, message numeration will be introduced. Numeration will be added as a suffix to an existing MessageID attribute that is generated for each event. MessageID will be formatted in a following way:

```
<MessageID>1234567890_1/n</MessageID>
```

Where “n” represents the total number of segments after splitting.

5.2.4. Use Cases

The list of possible use cases and corresponding faults is given in Table 5.2.

Table 5.2 – The SendIncidentUsagePoints operation use cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Supported common use cases are described in				
Invalid Verb	Result	String	FAILED	Adapter sends request message with invalid Verb. Response message is sent by external system 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	Adapter sends request message with invalid Noun. Response message is sent by external system 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	Adapter sends request message in which some of the mandatory elements are missing. Response message is sent by external system 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}.	
	Result	String	FAILED	

Use Case	Message Mapping			Action
	Property	Type	Value	
Unable to process the request	Error.code	String	5.3	Adapter sends request message, but for some reason message processing fails in external system due to an internal server error. Fault response message is sent by external system component.
	Error.level	String	FATAL	
	Error.reason	String	InternalServerError	
	Error.details	String	Unable to process the request message with ID: {0} - transaction attempted and failed.	
ESB is not available	Result	String	N/A	EcoStruxure GridOps system sends request message to the external system but it failed due to unavailability.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Failover event	Result	String	N/A	EcoStruxure GridOps tries to send request messages to the external system which is unavailable. Adapter has a configurable retry mechanism, and should retry sending in case of external system unavailability. In case a failover event occurs while adapter is in retry regime, STBY adapter instance resumes where the formerly HOT adapter has stopped. All messages that were pending for sending, are transferred to new HOT adapter instance via DMS Replication service.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	

5.3. SendIncidentTroubleTickets Service

5.3.1. Overview

ONT adapter is subscribed to changes that correspond to trouble tickets data. Upon receiving the notification, adapter performs initial validation based on adapter configuration file and creates ChangedIncidentTroubleTickets event message used for invoking ChangedIncidentTroubleTickets operation on the external SendIncidentTroubleTickets service.

ChangedIncidentTroubleTicketsEvent message payload is modeled so that it groups the trouble tickets to its related incident. Main entity within the message payload is the Incident. It contains a list of all related trouble tickets with additional relevant information which includes id, caller detail, etc. Messages are sent separately for each incident. In case where publications are processed for TroubleTickets that are related to different incidents, more than one ChangedIncidentTroubleTicketsEvent message will be created and sent to corresponding external SendIncidentTroubleTickets service. Incident constraints are defined in the adapter configuration file. Upon receiving the event, adapter queries for related incident data to determines whether the incident in question is supported for processing. When the publication does not meet the defined constraints requirement, adapter flags the publication as invalid and dumps it to adapter log file.

5.3.2. Triggers

Adapter supports sending messages only for event-based triggers. Adapter listens to changes of PhoneCallEvent model. When a change of interest occurs, adapter processes the received publication, queries for additional business process related data, and populates corresponding notification message. Following events will trigger the creation of aforementioned message.

- PhoneCallEvent incident reference change – this trigger follows two cases:
  - PhoneCallEvent is referenced to a newly created incident.
  - PhoneCallEvent incident reference is updated due to complex incident management scenarios.

It should be noted that triggers for ChangedIncidentTroubleTickets operation are highly configurable within adapter configuration file. Abovementioned trigger is considered as default.

5.3.3. Transferring Large Messages

Depending on the implementation and configuration of the SOAP web service, limitation of the message size may exist. Since there is no limitation to a number of calls per customer, and in case of a large incident where numerous customers are submitting a trouble ticket, message with the incident trouble tickets can become too large to transfer via the SOAP request.

Outage Notification Adapter will divide large message into multiple smaller ones. Split will be performed when the number of the affected customers in the message exceeds the predefined data limit stated in the web service configuration, max message size parameter.

For the external system to be aware of the message number, message numeration will be introduced. Numeration will be added as a suffix to an existing MessageID attribute that is generated for each event. MessageID will be formatted in a following way:

```
<MessageID>1234567890_1/n</MessageID>
```

Where “n” represents the total number of segments after splitting.

5.3.4. Use Cases

The list of possible use cases and corresponding faults is given in Table 5.3.

Table 5.3 – SendIncidentTroubleTickets Operation Use Cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Supported common use cases are described in				
Invalid Verb	Result	String	FAILED	Adapter sends request message with invalid Verb. Response message is sent by external system 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	Adapter sends request message with invalid Noun. Response message is sent by external system 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	Adapter sends request message in which some of the mandatory elements are missing. Response message is sent by external system 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	
	Error.code	String	5.3	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.level	String	FATAL	Adapter sends request message, but for some reason message processing fails in external system due to an internal server error. Fault response message is sent by external system component.
	Error.reason	String	InternalServerError	
	Error.details	String	Unable to process the request message with ID: {0} - transaction attempted and failed.	
ESB is not available	Result	String	N/A	EcoStruxure GridOps system sends request message to the external system but it failed due to unavailability.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Failover event	Result	String	N/A	EcoStruxure GridOps tries to send request messages to the external system which is unavailable. Adapter has a configurable retry mechanism, and should retry sending in case of external system unavailability. In case a failover event occurs while adapter is in retry regime, STBY adapter instance resumes where the formerly HOT adapter has stopped. All messages that were pending for sending, are transferred to new HOT adapter instance via DMS Replication service.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	

5.4. Integrity Update

5.4.1. Overview

Integrity update is a process designed to align connected systems. It exists in order of making integration between ONT adapter and external system more reliable. There are possible failure scenarios where adapter is unable to process model change update, due to a number of reasons, and for that purpose, integrity update process is introduced. It's a scheduled process that takes a snapshot of a complete model covered within the scope of ONT adapter and notifies the external system about the current state.
Following messages are created during integrity update process:

- ChangedIncidentsEvent message – it differentiates 3 different use cases:
  - IntegrityUpdateStart event – in the moment when integrity update process is triggered, adapter creates ChangedIncidentsEvent message with an empty payload and "IntegrityUpdateStart" noun attribute within message header. It notifies the external system that integrity update process is about to start. When IntegrityUpdateStart event cannot be sent to external system, or when the external system returns FAILED response, integrity update process is aborted and will be executed in the next scheduled cycle.
  - Model alignment – upon receiving OK response for IntegrityUpdateStart event, ONT adapter queries ONT service in order to retrieve snapshot of the current state of incidents. Forms corresponding ChangedIncidentsEvent messages (each incident is sent separately, as described in chapter SendIncidents Service) and invokes ChangedIncidents operation on SendIncidents service. Only difference to separate ChangedIncidents operations is in message header. Header noun must be "IntegrityUpdate" in order to differentiate the integrity update process from regular model update notification.
  - IntegrityUpdateEnd event – upon exporting all integrity update related messages (incidents, usage points and trouble tickets), adapter finalizes the process with sending an additional ChangedIncidentsEvent message, representing IntegrityUpdateEnd event. IntegrityUpdateEnd event message is the same as IntegrityUpdateStart, only difference is in header noun attribute.
- ChangedIncidentUsagePointsEvent message – upon receiving OK Response for IntegrityUpdateStart event, ONT adapter queries OMS service in order to retrieve snapshot of the current state of incident usage points. Forms corresponding ChangedIncidentUsagePointsEvent messages (usage points are grouped for each incident, and sent separately, as described in chapter SendIncidentUsagePoints Service) and invokes ChangedIncidentUsagePoints operation on SendIncidentUsagePoints service. Only difference to separate ChangedIncidentUsagePoints operations is in message header. Header noun must be "IntegrityUpdate" in order to differentiate the integrity update process from regular model update notification.
- ChangedIncidentTroubleTicketsEvent message – upon receiving OK response for IntegrityUpdateStart event, ONT adapter queries OMS service in order to retrieve snapshot of the current state of incident trouble tickets. Forms corresponding ChangedIncidentTroubleTicketsEvent messages (trouble tickets are grouped for each incident, and sent separately, as described in chapter SendIncidentTroubleTickets Service) and invokes ChangedIncidentTroubleTickets operation on SendIncidentTroubleTickets service. Only difference to separate ChangedIncidentTroubleTickets

operations is in message header. Header noun must be “IntegrityUpdate” in order to differentiate the integrity update process from regular model update notification.

While integrity update is in progress, individual publications about model changes are processed and queued, but not sent to the external system. Upon integrity update finalization, all queued messages are sent to their corresponding web services.

Incident filtering that applies to all previously described interfaces, applies also for integrity update process. Only incidents that satisfy defined filtering criteria are queried and processed accordingly.

It should be noted that by default, integrity update is turned off in adapter registry configuration file.

5.4.2. Triggers

Adapter supports sending messages only for event-based triggers. When such a trigger occurs, adapter queries for all incident data, and their corresponding usage point and trouble ticket information. Supported integrity update triggers are:

- Adapter startup & failover events (Integration service startup/failover):
  - Defines the moment when any of the adapter pairs turns HOT.
- Scheduled event:
  - Periodic trigger that is configurable within adapter registry configuration file.

5.4.3. Transferring Large Messages

Since integrity update messages do not differentiate from usual ChangedIncidentTroubleTickets and ChangedIncidentUsagePoints (apart from message header), same rules apply for message splitting scenarios. In case where aforementioned messages exceed the defined message size limit, they will be split into segments and sent separately to their corresponding web services.

5.4.4. Use Cases

The list of possible use cases and corresponding faults is given in Table 5.4.

Table 5.4 - The IntegrityUpdate operation use cases

Use Case	Message Mapping			Action
	Property	Type	Value	
Supported common use cases are described in				
Invalid Verb	Result	String	FAILED	Adapter sends request message with invalid Verb. Response message is sent by external system with FAILED result and message is discarded. Defined verb for this process is “changed”.
	Error.code	String	2.9	
	Error.level	String	FATAL	
	Error.reason	String	InvalidVerb	
	Error.details	String	Invalid verb: {0}.	
Invalid Noun	Result	String	FAILED	Adapter sends request message with invalid Noun. Response message is sent by external system with FAILED result and message is discarded. Defined noun values for this process is “IntegrityUpdateStarted”, “IntegrityUpdate” and “IntegrityUpdateFinished”.
	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	Adapter sends request message in which some of the mandatory elements are missing. Response message is sent by external system 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	
	Error.code	String	5.3	

Use Case	Message Mapping			Action
	Property	Type	Value	
	Error.level	String	FATAL	Adapter sends request message, but for some reason message processing fails in external system due to an internal server error. Fault response message is sent by external system component.
	Error.reason	String	InternalServerError	
	Error.details	String	Unable to process the request message with ID: {0} - transaction attempted and failed.	
ESB is not available	Result	String	N/A	EcoStruxure GridOps system sends request message to the external system but it failed due to unavailability.
	Error.code	String	N/A	
	Error.level	String	N/A	
	Error.reason	String	N/A	
	Error.details	String	N/A	
Failover during integrity update process	Result	String	N/A	Upon receiving notification about the failover event, adapter which transitions to STBY state must abort IU process and stop sending messages. Adapter which transitions to HOT state reinitializes IU process and sends new “IntegrityUpdateStarted” event message. External system should disregard previously received model snapshot.
	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 standard. 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 Incidents, IncidentUsagePoints, IncidentTroubleTickets, IntegrityUpdate etc.

Field that can be optionally supplied include the following:

- Revision – to indicate the revision of the message definition. By default, this should be “2.0”.
- 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.

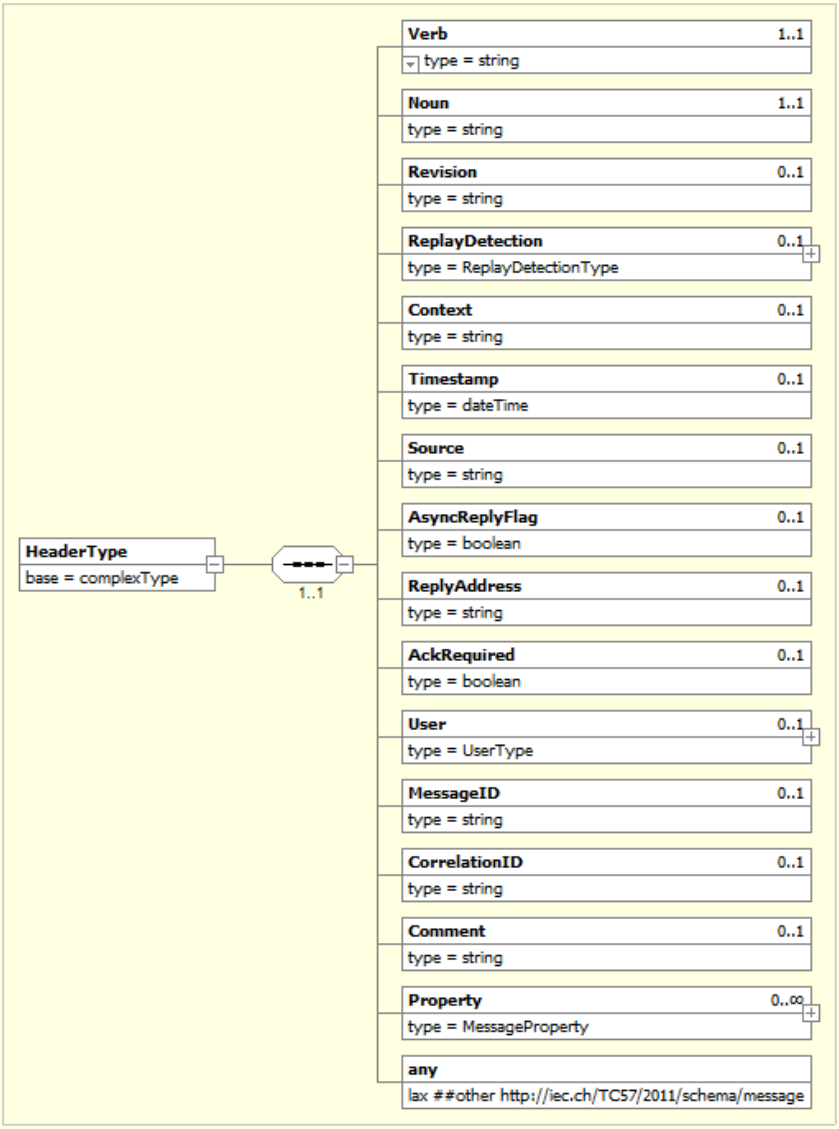

Figure 6.1 – The header field

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.

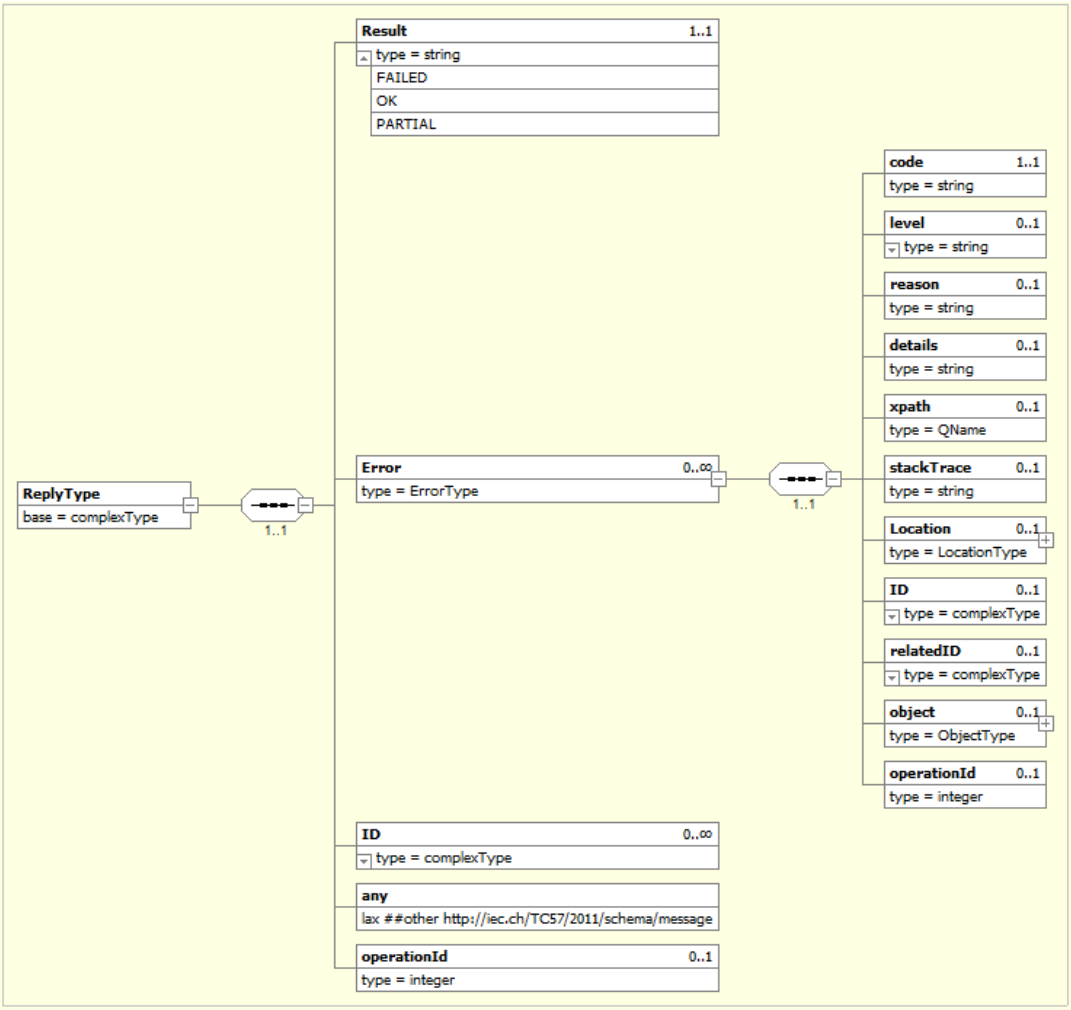

Figure 6.2 – The Reply and Error field contents

6.2. Incidents Operation Messages

The operation definitions:

CreatedIncidentsResponse CreatedIncidents (CreatedIncidentsEvent)

ChangedIncidentsResponse ChangedIncidents (ChangedIncidentsEvent)

DeletedIncidentsResponse DeletedIncidents (DeletedIncidentsEvent)

6.2.1. Events

6.2.1.1. CreatedIncidents

The CreatedIncidents event message is defined according to the IEC 61968-100 and contains the following two sections:

- Header
- Payload

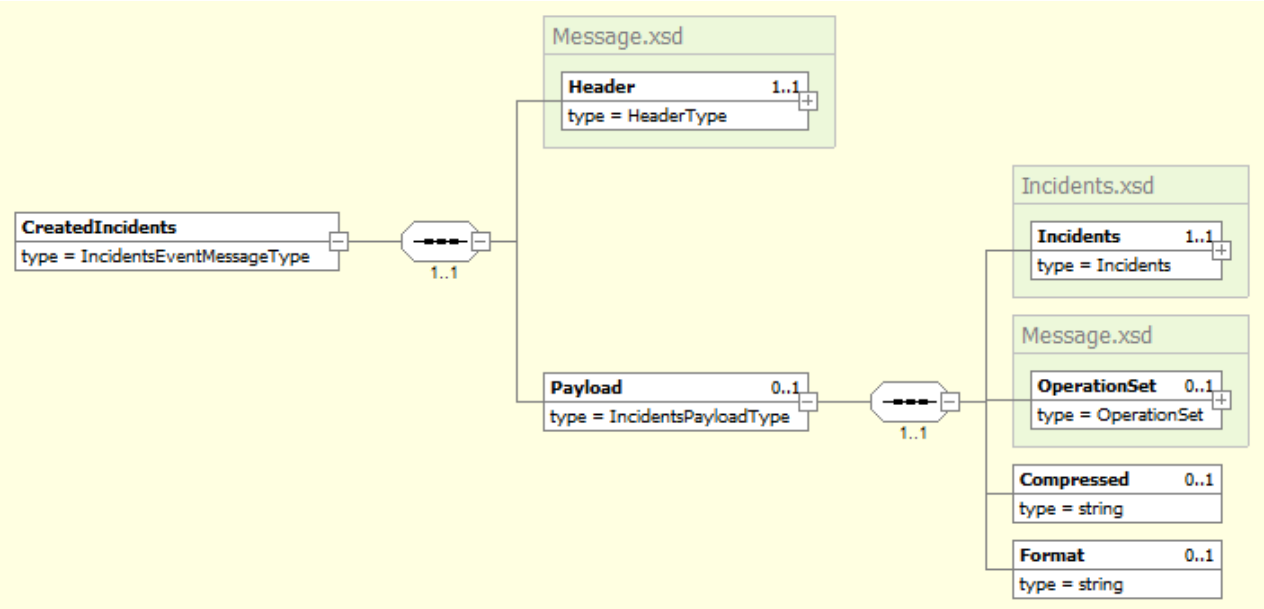

Figure 6.3 – The CreatedIncidentsEvent message

The Payload section carries the CIM defined profile (*Incidents.xsd*) for notification about recently created incidents. The visual representation of the *Incidents.xsd* schema is given in Figure 6.6.

6.2.1.2. ChangedIncidents

The ChangedIncidents event message is defined according to the IEC 61968-100 and contains the following two sections:

- Header
- Payload

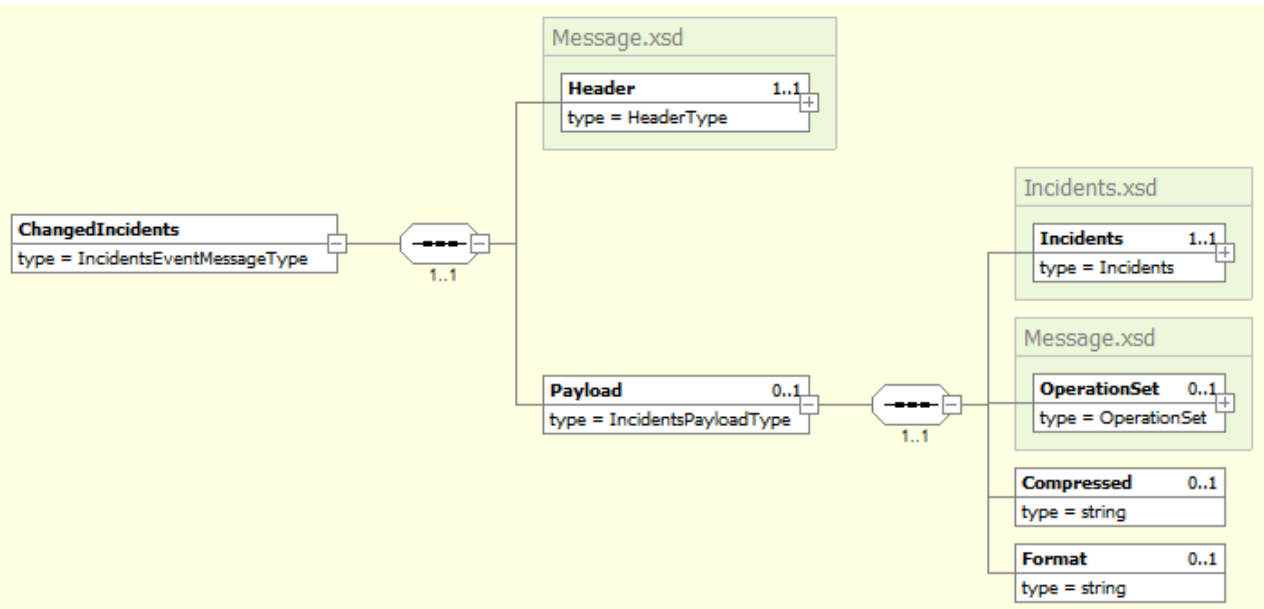

Figure 6.4 – The ChangedIncidentsEvent message

The Payload section carries the CIM defined profile (Incidents.xsd) for notification about recently changed incidents. The visual representation of the Incidents.xsd schema is given in Figure 6.6.

6.2.1.3. DeletedIncidents

The DeletedIncidents event message is defined according to the IEC 61968-100 and contains the following two sections:

- Header
- Payload

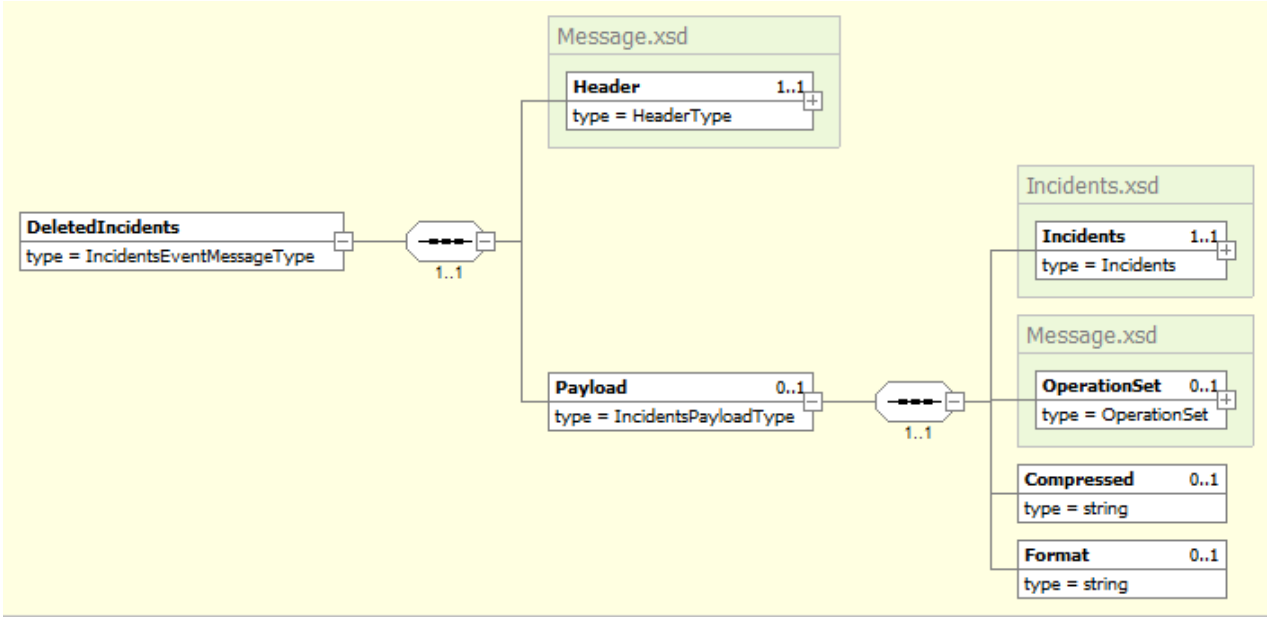

Figure 6.5 – The DeletedIncidentsEvent message

The Payload section carries the CIM defined profile (*Incidents.xsd*) for notification about recently deleted incidents. The visual representation of the *Incidents.xsd* schema is given in Figure 6.6.

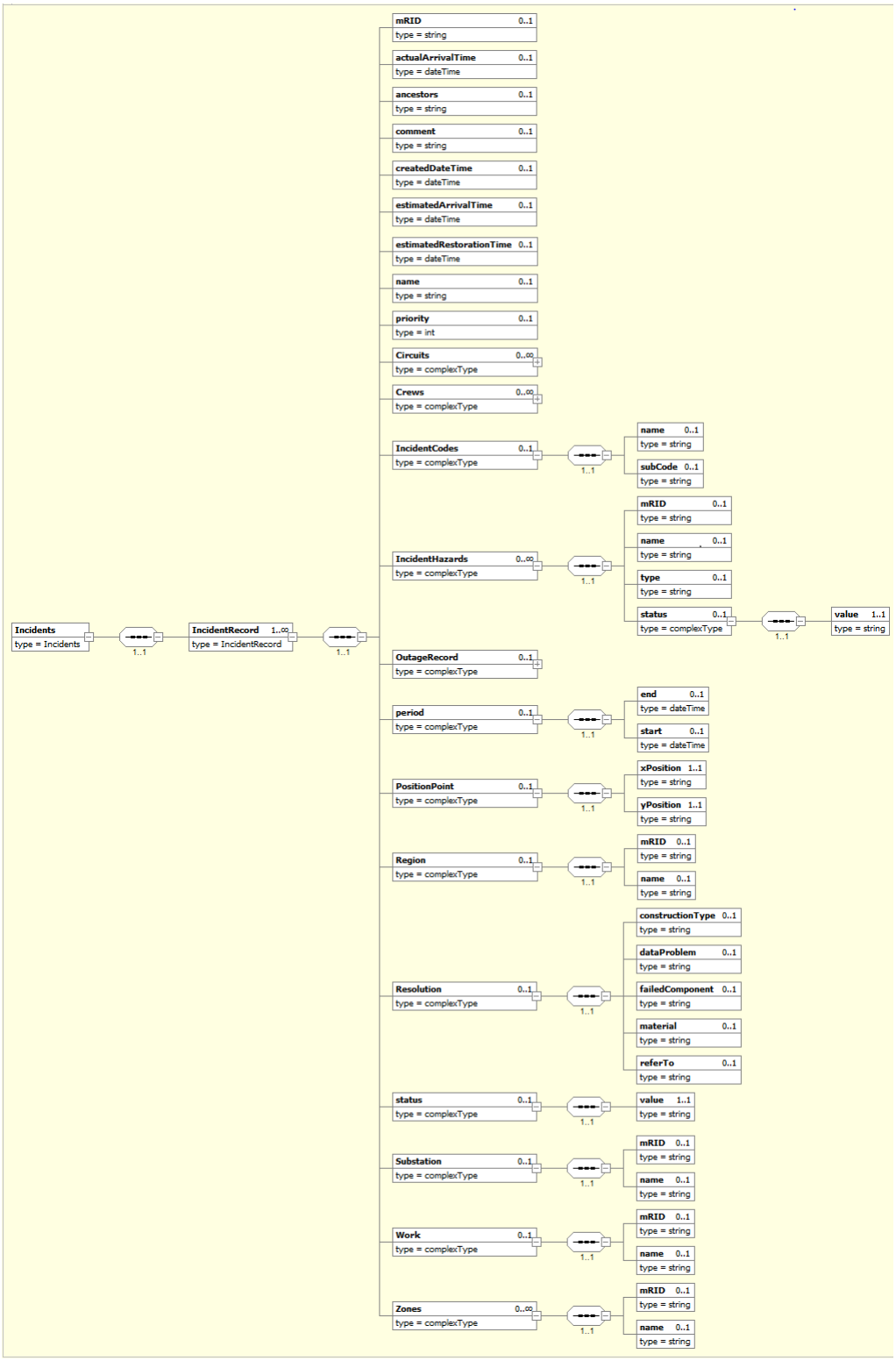

Figure 6.6 – Incidents.xsd

The incident contains the data about the actual outage present in the distribution network and all incident problems reported either through the trouble tickets (as hazards) or through the incident hazards interface. The aforementioned outage data are transferred within the OutageRecord object displayed in Figure 6.7.

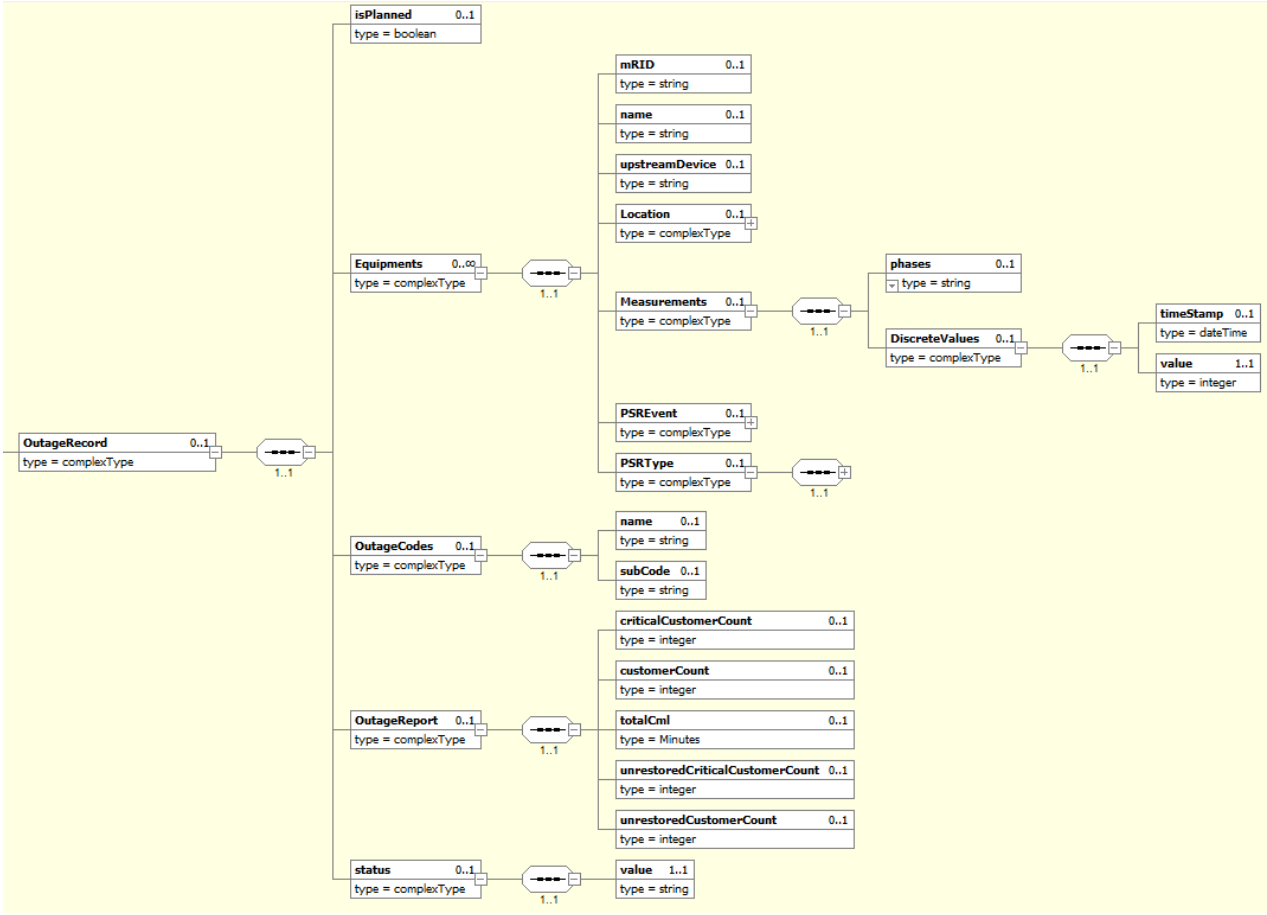

Figure 6.7 –The OutageRecord object

Table 6.1 defines the mapping between the payload of the IncidentsEvent message and the appropriate entities in the outage model.

Table 6.1 – The IncidentsEvent message → the outage model mapping

IncidentsEvent message			Description	Outage model		
Section	**Property**	**Type**		**Property**	**Type**	**Model Code**
Header	Verb	String	Identifier for a specific action to be taken. For this message, Verb can be created/changed/deleted.	Populated by ONT Adapter	N/A	N/A
Header	Noun	String	Identifier for the subject of the action and/or the type of the payload. For this message, Noun can be Incidents, IntegrityUpdateStart, IntegrityUpdate and IntegrityUpdateEnd.	Populated by ONT Adapter	N/A	N/A
Header	Revision	String	Revision of CIM standard used. Default value is 2.0.	Populated by ONT Adapter	N/A	N/A
Header	Timestamp	DateTime	Timestamp when message was produced. Example: 2015-12-31T12:34:56+02:00	Populated by ONT Adapter	N/A	N/A
Header	Source	String	Source system or application that sends the message. For this message, Source is EcoStruxure GridOps.	Populated by ONT Adapter	N/A	N/A
Header	MessageID	String	Unique message ID to be used for tracking messages.	Populated by ONT Adapter	N/A	N/A
Header	CorrelationID	String	Same as message ID.	Populated by ONT Adapter	N/A	N/A
Payload	mRID	String	Unique identifier of the incident.	CustomID	String	OMS_IMSOBJ_UID
Payload	actualArrivalTime	DateTime	Actual arrival time of first crew to the incident.	ata	DateTime	OMS_INCIDENT_ATA
Payload	ancestors	String	IDs of incidents that existed and were related to this incident. They were deleted and their details were transferred to this specific incident during complex incident management scenarios (merge, split, roll up, roll down, etc.). These incident	AncestryIDs	String	OMS_INCIDENT_ANCESTRY_UIDS

IncidentsEvent message			Description	Outage model		
Section	Property	Type		Property	Type	Model Code
			scenarios are explained in chapter Complex Incident Scenarios.			
Payload	estimatedArrivalTime	DateTime	Estimated arrival time of first crew to the incident.	eta	DateTime	OMS_INCIDENT_ETA
Payload	comment	String	Comments for incident entered by the operator or field crew.	NoteRefs	List<Long>	OMS_INCIDENT_NOTEREFS
Payload	createdDateTime	DateTime	Creation time of the incident.	CreateTime	DateTime	OMS_INCIDENT_CREATETIME
Payload	estimatedRestorationTime	DateTime	Estimated end time of the incident.	EstimatedEndTime	DateTime	OMS_INCIDENT_ESTIMATED_END_TIME
Payload	name	String	Optional name of the incident. Not used by EcoStruxure GridOps.	N/A	N/A	N/A
Payload	priority	Integer	Priority of an incident.	Priority	Integer	OMS_INCIDENT_PRIORITY
Payload	Circuits.name	List<Circuit>	List of feeders affected by the incident, concatenated with coma.	AffectedFeederNames	List<String>	OMS_INCIDENT_AFFECTED_FEEDER_NAMES
Payload	Crews.mRID	List<Crew>	IDs and Names of the crews assigned to the incident.	CrewRefs	List<Long>	OMS_INCIDENT_CREW_CUSTOMID
Payload	Crews. Assignments.status.mRID	String	Crew status	AssignmentStatus	String	OMS_INCIDENT_ASSIGMENT_STATUS_NAME
Payload	IncidentCodes.name	String	Determines the type of the incident.	IncidentTypeRef	Long	OMS_INCIDENT_TYPEREF
Payload	IncidentCodes.subCode	String	Determines the subtype of the incident.	IncidentSubTypeRef	Long	OMS_INCIDENT_SUBTYPEREF
Payload	IncidentHazards	List<Incident Hazard>	Information about all incident hazards associated to the incident. mRID, Name, Type (line down, gas leak, fire, etc) and status of each hazard.	IncidentProblemRef	String	OMS_INCIDENT_PROBLEMREFS
Payload	OutageRecord. isPlanned	Boolean	Indicate if outage is planned.	IsPlanned	Boolean	OMS_INCIDENT_IS_PLANNED
Payload	OutageRecord. OutageCodes.	String	Incident cause.	IncidentCauseRef	Long	OMS_INCIDENT_CAUSE_REF

IncidentsEvent message			Description	Outage model		
Section	Property	Type		Property	Type	Model Code
	name					
Payload	OutageRecord. OutageCodes. subCode	String	Incident subcause.	IncidentSubCasuseRef	Long	OMS_INCIDENT_SUBCAUSE_REF
Payload	OutageRecord. Equipments. mRID	String	Incident device unique identifier.	DeviceRefs	String	OMS_DEVICE_CUSTOM_ID
Payload	OutageRecord. Equipments. name	String	Device name.	DeviceRefs	String	OMS_DEVICE_NAME
Payload	OutageRecord. Equipments. upstreamDevice	String	Information about upstream device(s).	UpstreamDeviceRefs	String	OMS_DEVICE_UPSTREAM_DEVICES_NAMES
Payload	OutageRecord. Equipments. Location.direction	String	Location description.	DeviceLocation	String	OMS_DEVICE_LOCATION
Payload	OutageRecord. Equipments.Measurement. DiscreteValues.phases	String	Device phases.	DevicePhases	Enum	OMS_DEVICE_PHASES
Payload	OutageRecord. Equipments.Measurement. DiscreteValues.value	Integer	Device status. Example: Open – 1, Closed – 2.	DeviceRefs	Bool	OMS_DEVICE_IS_OPEN
Payload	OutageRecord. Equipments. PSRType.type	String	Custom Type.	DeviceType	String	OMS_DEVICE_TYPE
Payload	OutageRecord. OutageReport. criticalCustomerCount	Integer	Number of affected critical customers.	NumCriticalCustomers	Integer	OMS_INCIDENT_NUM_CRITICAL_CUSTOMERS

IncidentsEvent message			Description	Outage model		
Section	Property	Type		Property	Type	Model Code
Payload	OutageRecord. OutageReport. customerCount	Integer	Number of affected customers.	NumCustomers	Integer	OMS_INCIDENT_NUM_CUSTOMERS
Payload	OutageRecord. OutageReport. unrestoredCritical CustomerCount	Integer	Number of unrestored critical customers.	NumUnrestCriticalCustomer	Integer	OMS_INCIDENT_NUM_UNREST_CRITICAL_CUSTOMERS
Payload	OutageRecord. OutageReport. unrestoredCustomerCount	Integer	Number of unrestored customers.	NumUnrestCustomers	Integer	OMS_INCIDENT_NUM_UNREST_CUSTOMERS
Payload	OutageRecord. OutageReport. totalCml	Integer	Total customer minutes lost.	TotalCustInterruptionMinutes	Integer	OMS_INCIDENT_TOTAL_CUST_INTERRUPTION_DURATION
Payload	OutageRecord.status	String	Incident power status. Examples: NoOutage, Restored, NotRestored, PartiallyRestored.	IncidentPowerStatus	Enum	OMS_INCIDENT_POWER_STATUS
Payload	period.start	DateTime	Outage time of the incident.	OutageTime	DateTime	OMS_INCIDENT_OUTAGETIME
Payload	period.end	DateTime	Restoration time of an incident, set when incident is Completed or Canceled.	ActualEndTime	DateTime	OMS_INCIDENT_ACTUAL_END_TIME
Payload	PostionPoint.xPosition	String	X coordinate of the incident.	CoordinateX	Double	OMS_INCIDENT_COORDINATES_X
Payload	PostionPoint.yPosition	String	Y coordinate of the incident.	CoordinateY	Double	OMS_INCIDENT_COORDINATES_Y
Payload	Region.mRID	String	Unique identifier of the region to which incident belongs to	RegionRef	Long	OMS_INCIDENT_REGIONGID
Payload	Resolution. constructionType	String	Field for defining type of the construction on which work was performed.	ConstructionTypeRef	Long	OMS_INCIDENT_CONSTRUCTION_TYPE_REF
Payload	Resolution.dataProblem	String	Indication whether there was problem with data.	DataProblemRef	Long	OMS_INCIDENT_DATA_PROBLEM_REF

IncidentsEvent message			Description	Outage model		
Section	Property	Type		Property	Type	Model Code
Payload	Resolution. failedComponent	String	Field to define type of component on which was problem.	FailedComponentRef	Long	OMS_INCIDENT_FAILED_COMPONENT_REF
Payload	Resolution.material	String	Material used for repair.	MaterialRef	Long	OMS_INCIDENT_MATERIAL_REF
Payload	Resolution.referTo	String	Field used to set who was processing incident.	ReferToRef	Long	OMS_INCIDENT_REFER_TO_REF
Payload	status.value	String	Incident status. Examples: New, Dispatched, Damage Assessed, Field Completed, Closed, Archived, Cancelled.	IncidentStatusRef	Long	OMS_INCIDENT_STATUSREF
Payload	status.remark	String	Incident confirmation state (Confirmed, Unconfirmed)	IncidentConfirmedState	bool	OMS_INCIDENT_INC_CONFIRMED_STATE
Payload	Substation.mRID	String	Unique identifier of the substation.	SubstationRef	Long	OMS_INCIDENT_SUBSTATIONGID
Payload	Work.mRID	String	Unique identifier of a work plan assigned to incident.	WorkPlanGid	Long	OMS_INCIDENT_FOLLOW_UP_WORK PLAN_GIDS
Payload	Zones.mRID	String	Unique identifiers of AOR Groups	AORGroup	List<String>	OMS_INCIDENT_AORGROUP

6.2.2. Response

After incidents are successfully sent to external service, the synchronous response is returned in form of the IncidentsResponse message. The unique identifier of the incident along with its operation status is returned within the response message. The content is given in Figure 6.8.

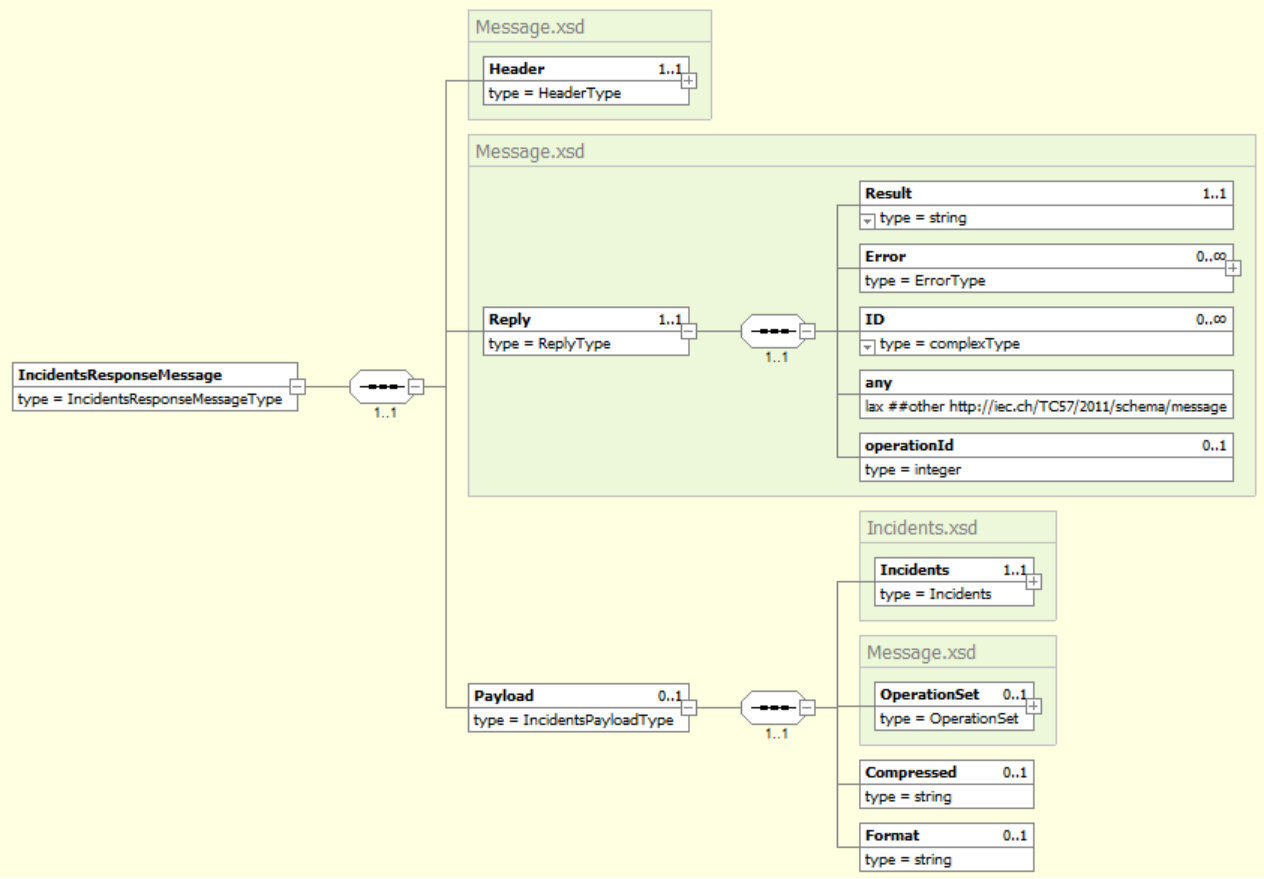

Figure 6.8 – The IncidentsResponse message

Table 6.2 defines the mapping between the *Incidents.xsd* object and the appropriate entities in the outage model for the response message.

Table 6.2 – The IncidentsResponse message → the outage model mapping

IncidentsResponse message			Description	Outage model		
Section	**Property**	**Type**		**Property**	**Type**	**Model Code**
Header	**Verb**	String	Identifier for a specific action to be taken. For this message, Verb is reply.	Populated by external system	N/A	N/A
Header	**Noun**	String	Identifier for the subject of the action and/or the type of the payload. For this message, Noun is Incidents.	Populated by external system	N/A	N/A
Header	Revision	String	Revision of CIM standard used. Default value is 2.0.	Populated by external system	N/A	N/A
Header	**Timestamp**	DateTime	Timestamp when message was produced. Example: 2015-12-31T12:34:56+02:00	Populated by external system	N/A	N/A
Header	Source	String	Source system or application that sends the message. For this message, Source is EcoStruxure GridOps.	Populated by external system	N/A	N/A
Header	**MessageID**	String	Unique message ID to be used for tracking messages.	Populated by external system	N/A	N/A
Header	**CorrelationID**	String	Same as correlation ID from request message.	Populated by external system	N/A	N/A
Reply	**Result**	String	Returned as part of synchronous response. Valid values are: OK, PARTIAL or FAILED.	Populated by external system	N/A	N/A

6.2.3. Fault

The IncidentsFault message is depicted in Figure 6.9.

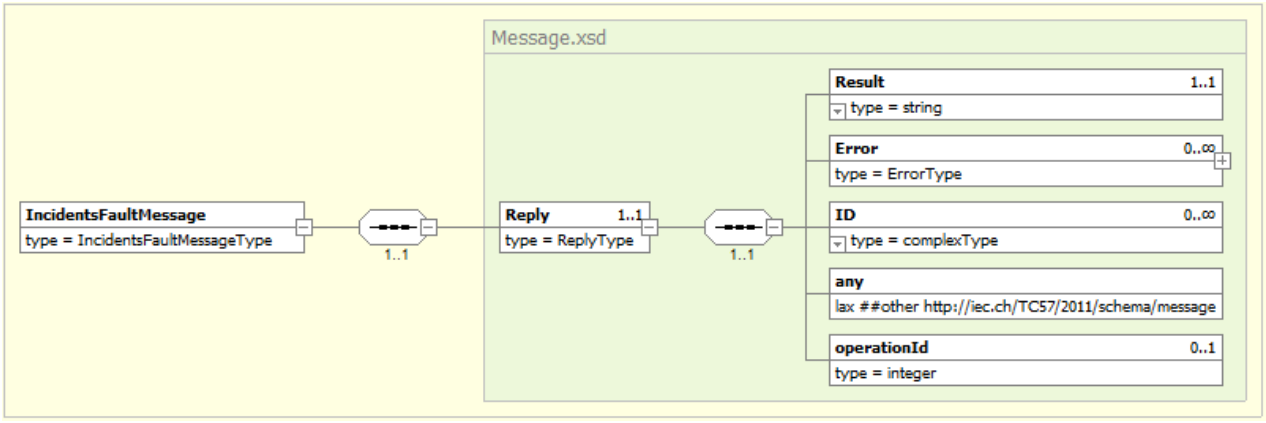

Figure 6.9 – The IncidentHazardsFault message

6.3. IncidentUsagePoints Operation Message

The operation definition:

ChangedIncidentUsagePointsResponse
ChangedIncidentUsagePoints(ChangedIncidentUsagePointsEvent)

6.3.1. Event

The ChangedIncidentUsagePoints *e*vent message is defined according to the IEC 61968-100 and contains the following two sections:

- Header
- Payload

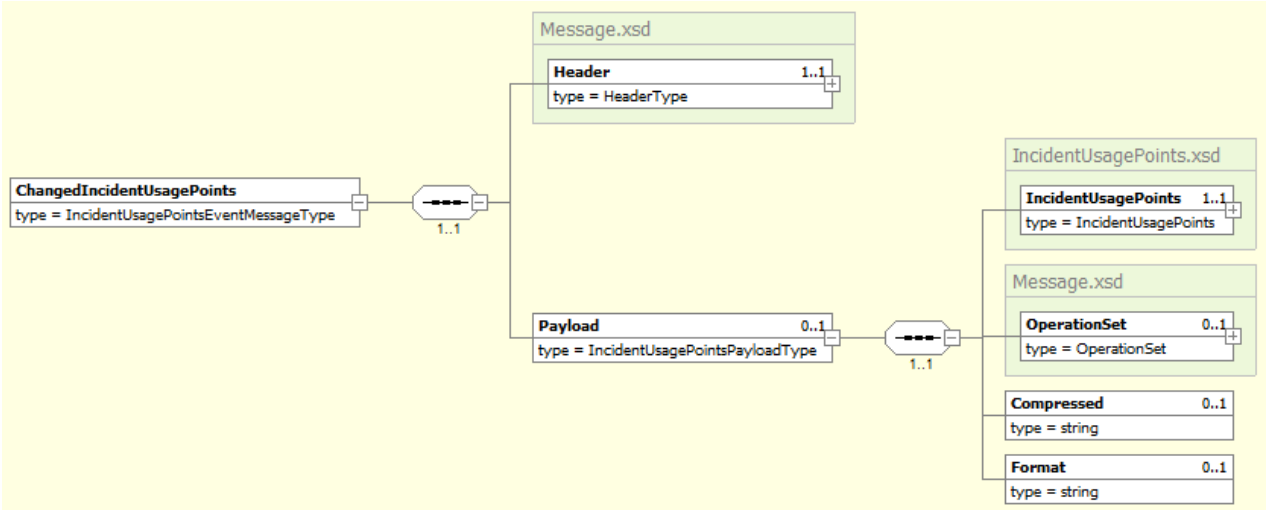

Figure 6.10 – The ChangedIncidentUsagePointsEvent message

The Payload section carries the CIM defined profile (*IncidentUsagePoints.xsd*) for update of one or several incident usage points. The visual representation of the *IncidentUsagePoints.xsd* schema is given in Figure 6.11.

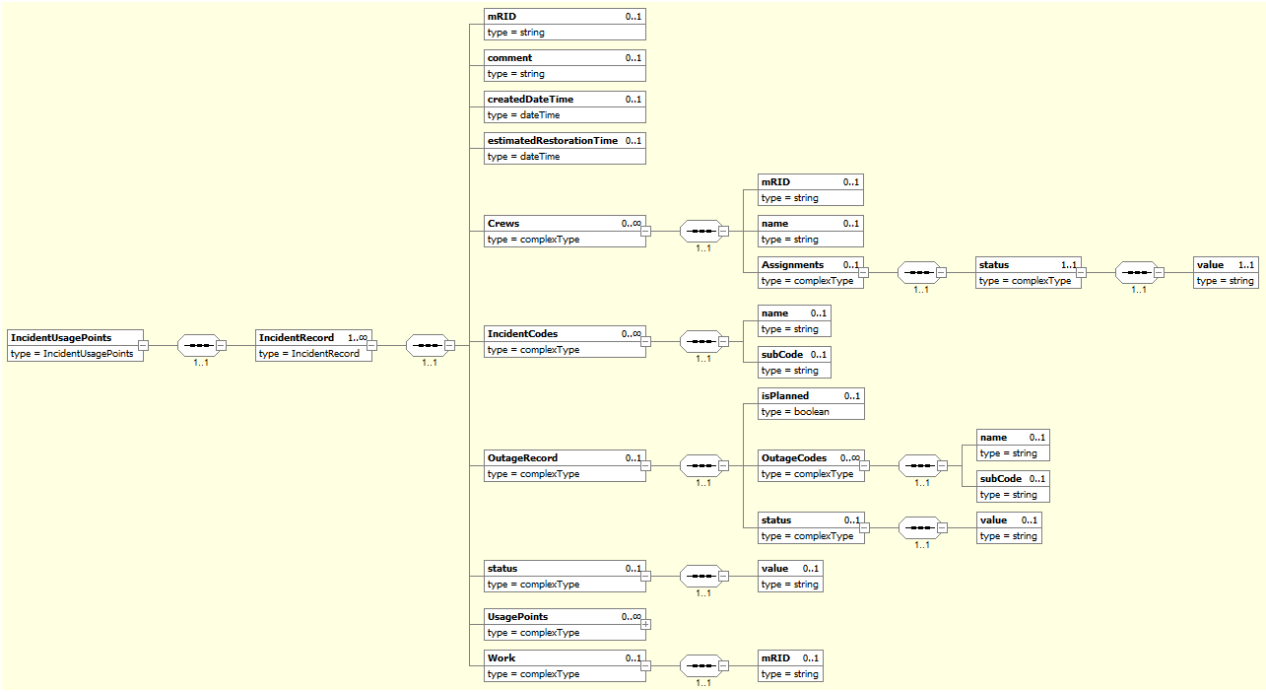

Figure 6.11 – The IncidentUsagePoints message

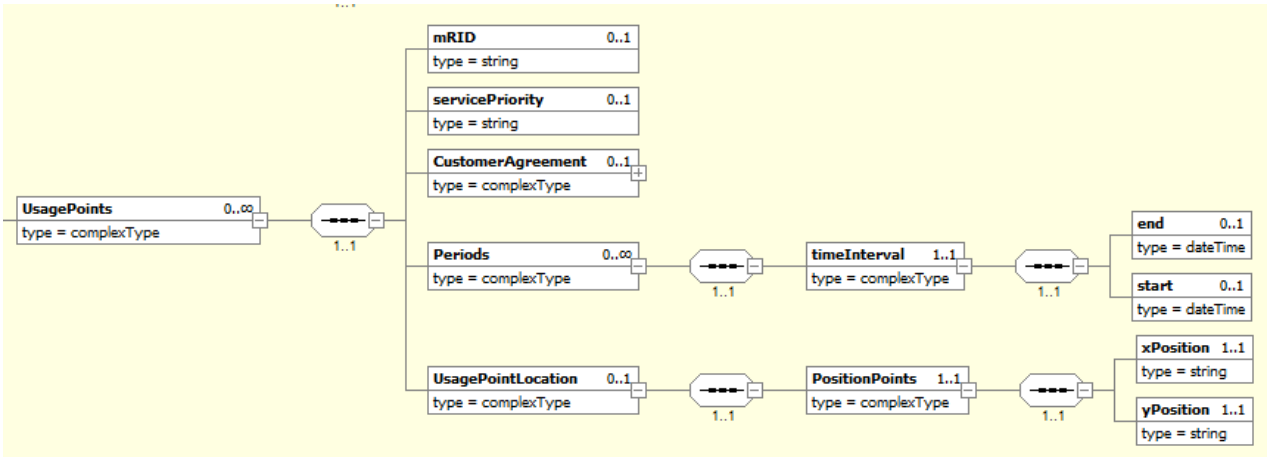

Figure 6.12 – UsagePoints object

Table 6.3 defines the mapping between the payload of the IncidentUsagePointsEvent message and the appropriate entities in the outage model.

Table 6.3 – The IncidentUsagePointsEvent message → the outage model mapping

IncidentUsagePoints.xsd			Description	Outage Model		
Section	**Property**	**Type**		**Property**	**Type**	**Model Code**
Header	Verb	String	Identifier for a specific action to be taken. For this message, Verb is changed.	Populated by ONT Adapter	N/A	N/A
Header	Noun	String	Identifier for the subject of the action and/or the type of the payload. For this message, Noun can be IncidentUsagePoints and IntegrityUpdate.	Populated by ONT Adapter	N/A	N/A
Header	Revision	String	Revision of CIM standard used. Default value is 2.0.	Populated by ONT Adapter	N/A	N/A
Header	Timestamp	DateTime	Timestamp when message was produced. Example: 2015-12-31T12:34:56+02:00	Populated by ONT Adapter	N/A	N/A
Header	Source	String	Source system or application that sends the message. For this message, Source is EcoStruxure GridOps.	Populated by ONT Adapter	N/A	N/A
Header	MessageID	String	Unique message ID to be used for tracking messages.	Populated by ONT Adapter	N/A	N/A
Header	CorrelationID	String	Same as message ID.	Populated by ONT Adapter	N/A	N/A
Payload	mRID	String	Unique identifier of the incident.	CustomID	String	OMS_IMSOBJ_UID
Patload	createdDateTime	DateTime	Incident creation time	Incident create time	DateTime	OMS_INCIDENT_CREATETIME
Payload	estimatedRestorationTime	DateTime	Estimated end time of the incident	EstimatedEndTime	int	OMS_INCIDENT_ESTIMATED_RESTORATION_TIME
Payload	OutageRecord. isPlanned	Boolean	Indicate if outage is planned	IsPlanned	Bool	OMS_INCIDENT_ISPLANNED

IncidentUsagePoints.xsd			Description	Outage Model		
Section	Property	Type		Property	Type	Model Code
Payload	OutageRecord.status.value	string	Incident power status	IncidentPowerStatus	enum	OMS_INCIDENT_POWER_STATUS
Payload	UsagePoint. mRID	String	Custom ID of the SDP that is affected or unrestored.	SdpCustomID	String	OMS_SDP_CUSTOMID
Payload	UsagePoint.ServicePriority	int	Service priority of the customer	SdpServicePriority	int	OMS_SDP_PRIORITY
Payload	UsagePoint. CustomerAgreement. Customer.ErpPerson. firstName	String	Name of the customer whose account is connected to this usage point. CustomerAgreement data is optional and its population is configured in the adapter configuration file.	AccountName	DateTime	OMS_SDP_ACCOUNT_NAME
Payload	UsagePoint. CustomerAgreement. Customer.ErpPerson. lastName	String	Last name of the customer whose account is connected to this usage point. CustomerAgreement data is optional and its population is configured in the adapter configuration file.	AccountLastName	String	OMS_SDP_ACCOUNT_LAST_NAME
Payload	Periods	TimeInterval	Start and End times of UsagePoint interruptions			OMS_SDP_OUTAGE_TIME & OMS_SDP_RESTORATION_TIME
Payload	Work. mRID	String	ID of a work order assigned to incident.	WorkOrderID	Long	OMS_INCIDENT_INCIDENT_FOLLOW_UP_ WORKPLAN_REFS

6.3.2. Response

After IncidentUsagePoints notification message is successfully sent to external service, the response is returned in form of the IncidentUsagePoints response message. The unique identifier of the incidents containing provided affected usage points along with operation status is returned within the response message (Table 6.3). The content is given in Figure 6.13.

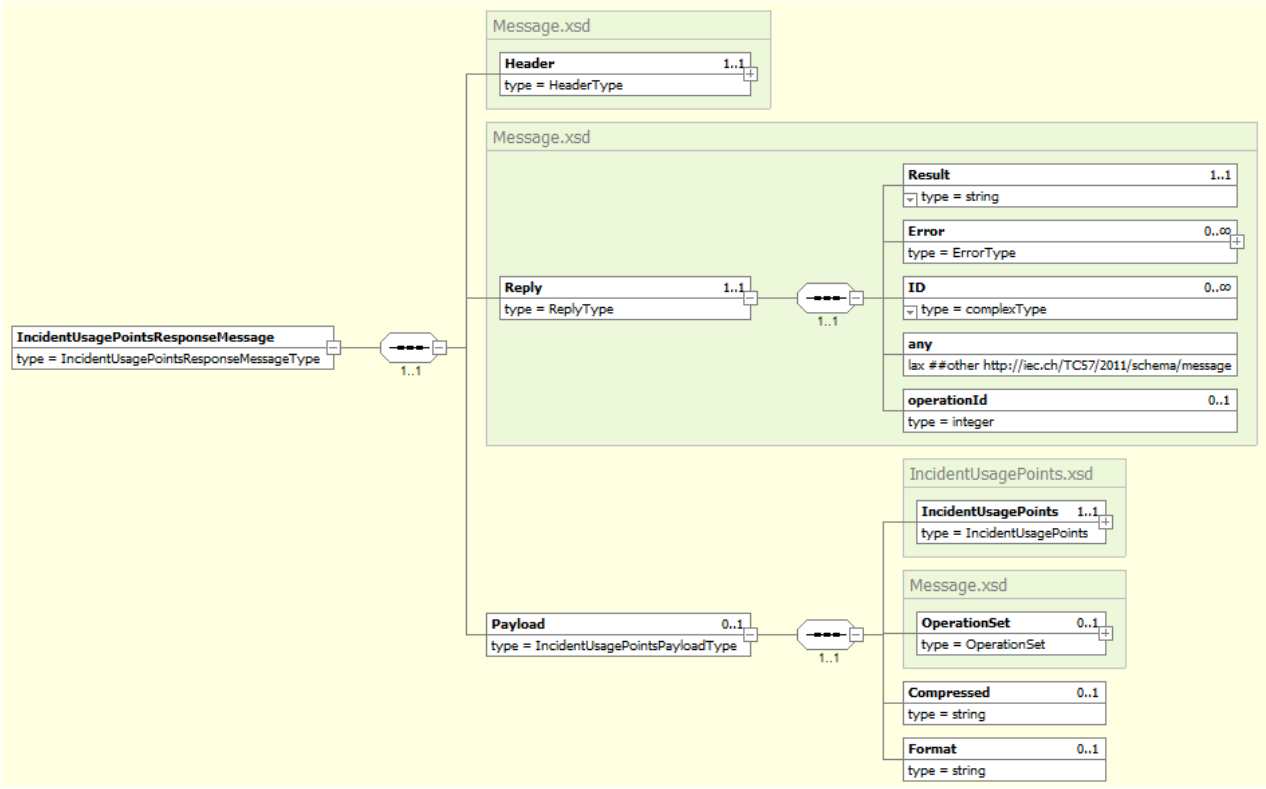

Figure 6.13 – IncidentUsagePointsResponseMessage

6.3.3. Fault

The IncidentUsagePointsFault message is depicted in Figure 6.14.

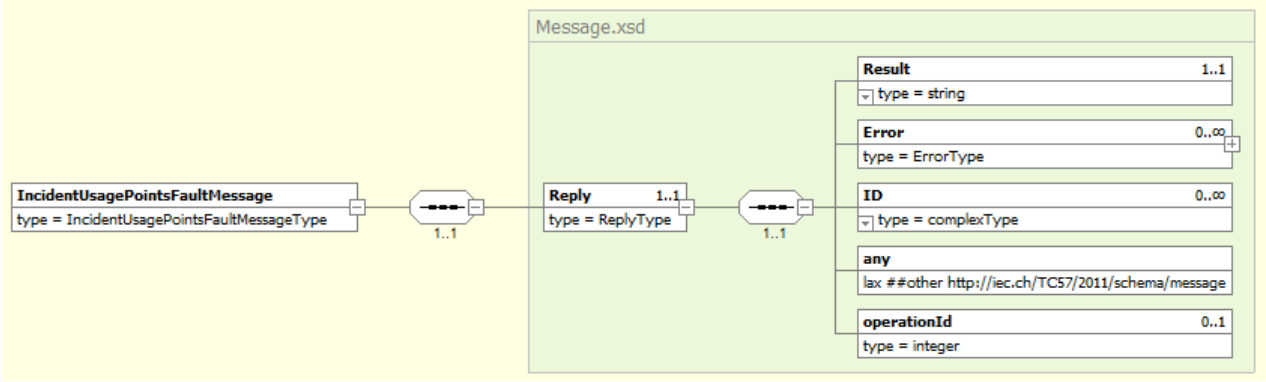

Figure 6.14 – IncidentUsagePointsFault message

6.4. IncidentTroubleTickets Operation Message

The operation definition:

ChangedIncidentTroubleTicketsResponse
ChangedIncidentTroubleTickets(ChangedIncidentTroubleTicketsEvent)

6.4.1. Event

The ChangedIncidentTroubleTickets *e*vent message is defined according to the IEC 61968-100 and contains the following two sections:

- Header
- Payload

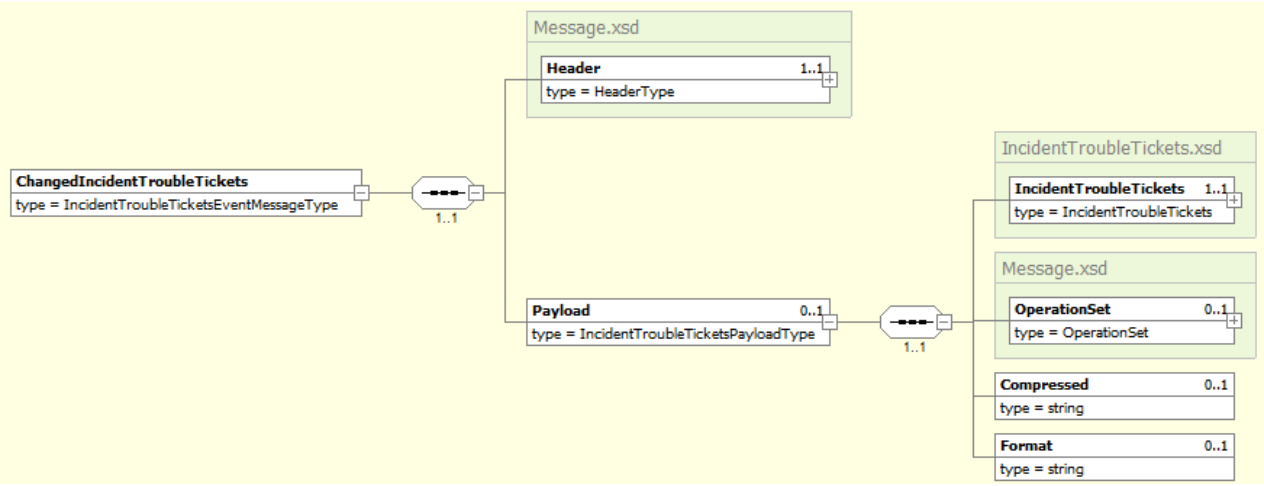

Figure 6.15 – The ChangedIncidentTroubleTicketsEvent message

The Payload section carries the CIM defined profile (*IncidentTroubleTickets.xsd*) for update of one or several incident trouble tickets. The visual representation of the *IncidentTroubleTickets.xsd* schema is given in Figure 6.11.

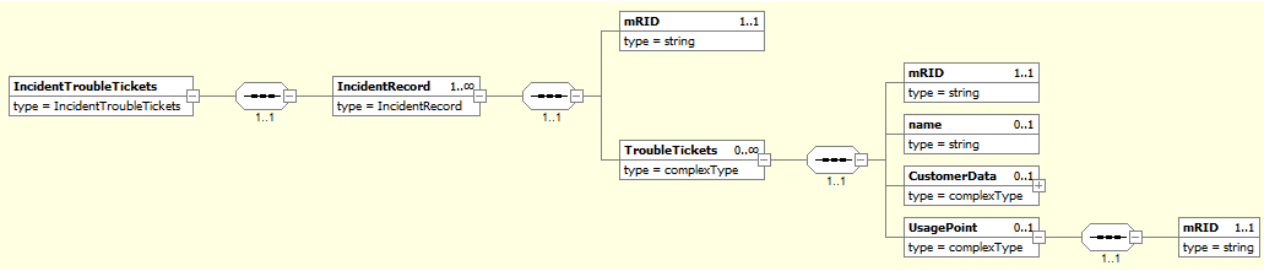

Figure 6.16 – The IncidentTroubleTickets message

Table 6.4 defines the mapping between the payload of the IncidentTroubleTicketsEvent message and the appropriate entities in the outage model.

Table 6.4 - The IncidentTroubleTicketsEvent message → the outage model mapping

IncidentUsagePoints.xsd			Description	Outage Model		
Section	**Property**	**Type**		**Property**	**Type**	**Model Code**
Header	Verb	String	Identifier for a specific action to be taken. For this message, Verb is changed.	Populated by ONT Adapter	N/A	N/A
Header	Noun	String	Identifier for the subject of the action and/or the type of the payload. For this message, Noun can be IncidentTroubleTickets and IntegrityUpdate.	Populated by ONT Adapter	N/A	N/A
Header	Revision	String	Revision of CIM standard used. Default value is 2.0.	Populated by ONT Adapter	N/A	N/A
Header	Timestamp	DateTime	Timestamp when message was produced. Example: 2015-12-31T12:34:56+02:00	Populated by ONT Adapter	N/A	N/A
Header	Source	String	Source system or application that sends the message. For this message, Source is EcoStruxure GridOps.	Populated by ONT Adapter	N/A	N/A
Header	MessageID	String	Unique message ID to be used for tracking messages.	Populated by ONT Adapter	N/A	N/A
Header	CorrelationID	String	Same as message ID.	Populated by ONT Adapter	N/A	N/A
Payload	mRID	String	Unique identifier of the incident.	CustomID	String	OMS_IMSOBJ_UID
Payload	TroubleTicket.mRID	String	Unique identifier of the call.	CustomID	String	OMS_IMSOBJ_UID
Payload	TroubleTicket.name	String	External system identifier of the call.	ExternalId	String	OMS_PHONE_CALL_EVENT_CUSTOMID
Payload	TroubleTicket.UsagePoint.mRID	String	Custom ID of the SDP that is affected or unrestored.	CustomId	String	OMS_SDP_CUSTOMID
Payload	TroubleTicket.CustomerData.ErpPerson.firstName	String	Name of the caller. CustomerData is optional and its population is configured in the adapter configuration file.	CallerName	String	OMS_CALLER_CONTACT_FIRST_NAME

IncidentUsagePoints.xsd			Description	Outage Model		
Section	Property	Type		Property	Type	Model Code
Payload	TroubleTicket. CustomerData.ErpPerson. lastName	String	Last name of the caller. CustomerData is optional and its population is configured in the adapter configuration file.	CallerLastName	String	OMS_CALLER_CONTACT_LAST_NAME
Payload	TroubleTicket. CustomerData.ErpPerson. electronicAddress.email1	String	Email of the caller. CustomerData is optional and its population is configured in the adapter configuration file.	ContactValue	String	OMS_CALLER_CONTACT_VALUE
Payload	TroubleTicket. CustomerData.ErpPerson. landlinePhone.localNumber	String	Landline phone number of the caller. CustomerData is optional and its population is configured in the adapter configuration file.	ContactValue	String	OMS_CALLER_CONTACT_VALUE
Payload	TroubleTicket. CustomerData.ErpPerson. mobilPhone.localNumber	String	Mobile phone number of the caller. CustomerData is optional and its population is configured in the adapter configuration file.	ContactValue	String	OMS_CALLER_CONTACT_VALUE

6.4.2. Response

After IncidentTroubleTickets notification message is successfully sent to external service, the response is returned in form of the IncidentTroubleTickets response message. The unique identifier of the incidents containing provided related trouble tickets along with operation status is returned within the response message (Table 6.3). The content is given in Figure 6.13.

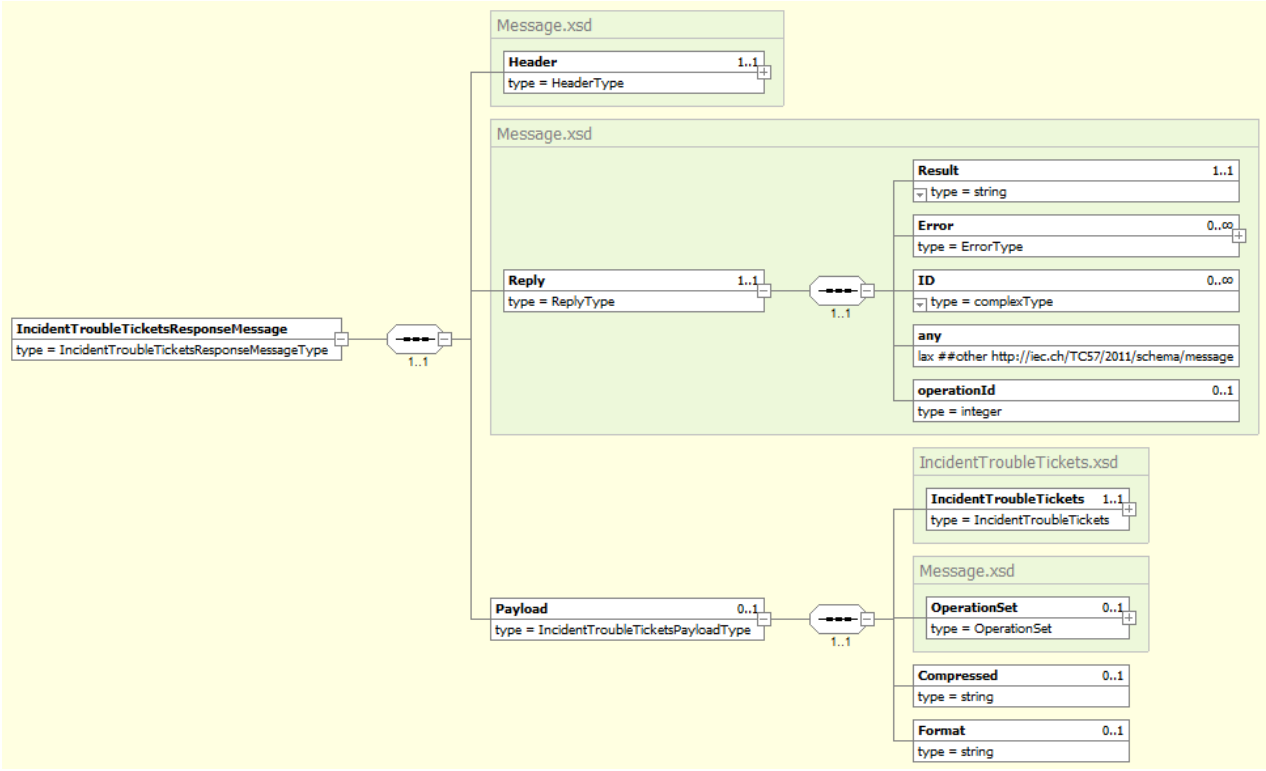

Figure 6.17 – The IncidentTroubleTicketsResponse message

6.4.3. Fault

The IncidentTroubleTicketsFault message is depicted in Figure 6.14.

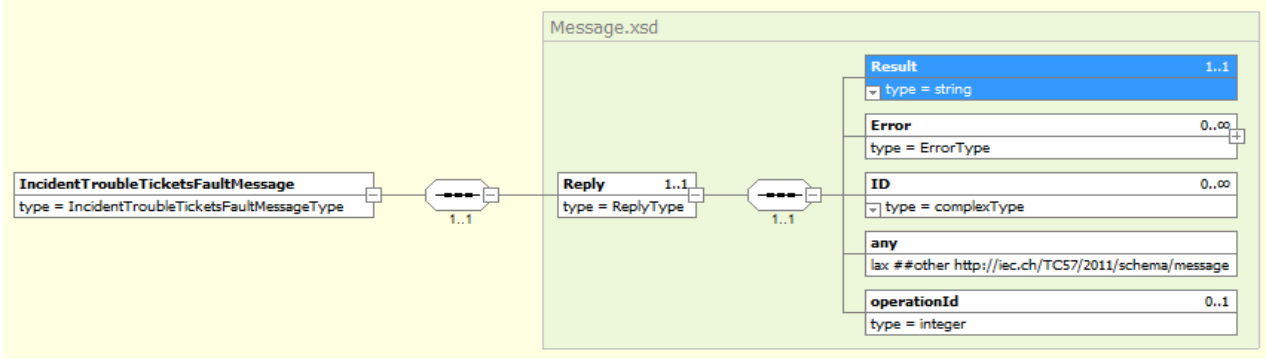

Figure 6.18 – The IncidentTroubleTicketsFault message

7. COMPLEX INCIDENT SCENARIOS

Some incident operations can cause the incident area and information to be divided, merged, or linked in some other way. These operations may result in multiple incident updates and the creation and/or deletion of some incidents. Complex incident operations explained in this chapter are:

- Manual or automatic roll up
- Manual roll down
- Manual merge
- Manual split
- Nested incidents

Following chapters describe these scenarios from the integration point of view. Each scenario is explained using an example and with the complete details of the integration messages which will be sent by the adapter.

These scenarios serve to assist external system in implementing their side of the integration. In the near real-life handling of incidents more complex situations can occur, where integration messages will be sent in a similar manner to those outlined in the examples. Since data is updated and retrieved by the adapter asynchronously, the order of the sent messages can be different than given in the examples. Schneider Electric recommendation is that client designs its systems and processes to not depend on the order in which messages are delivered by the Outage Notification Adapter.

Examples are shown on the simplified diagram of the electrical network. Network diagram legend is displayed in Figure 7.1.

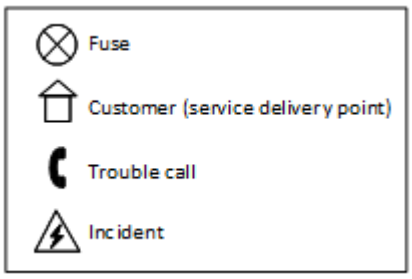

Figure 7.1 – The network diagram – the legend

NOTE: In the network model, customers are always connected to the service location entities. But for the simplicity of the display, it is assumed that one service location has only one customer connected, and both are presented using the single symbol of the house on the following diagrams.

7.1. Manual and Automatic Roll Up

When multiple incidents are located downstream from a common predicted device and it is determined that this device is the source of an outage, those incidents can be “rolled up” to that common device.

Roll up is available only for unconfirmed incidents.

The roll up can be done automatically by the prediction engine when conditions are met, or manually upon user request. The difference between an automatic and a manual roll up is the choice of a “parent incident”. Parent incident is the incident which will remain after the roll up, while other incidents are going to be deleted. In case of the manual roll up, the incident from which roll up was triggered is considered as the parent. On the other hand, in case of the automatic roll up, the parent incident cannot be determined with certainty. It may differ from case to case.

As a result of the roll up, multiple incidents are merged into the single parent incident. All trouble calls and the affected customers are transferred to that parent incident. The merged incident will maintain a reference to previously existing incidents—it will have a list of “ancestor incidents” with IDs of all incidents that are merged and removed. The list of ancestor IDs is going to be sent as a part of the integration messages, which can be used by the external systems.

The roll up will be explained in the following example where:

- Four trouble calls are created, each by a different customer (or service delivery point in EcoStruxure GridOps terminology).
- Two calls were located below low voltage transformer fuse “LV_Fuse_1” and the other below different low voltage transformer fuse “LV_Fuse_2”.
- As soon as calls are processed, each trouble call will initially create one incident.

This situation is displayed in Figure 7.2.

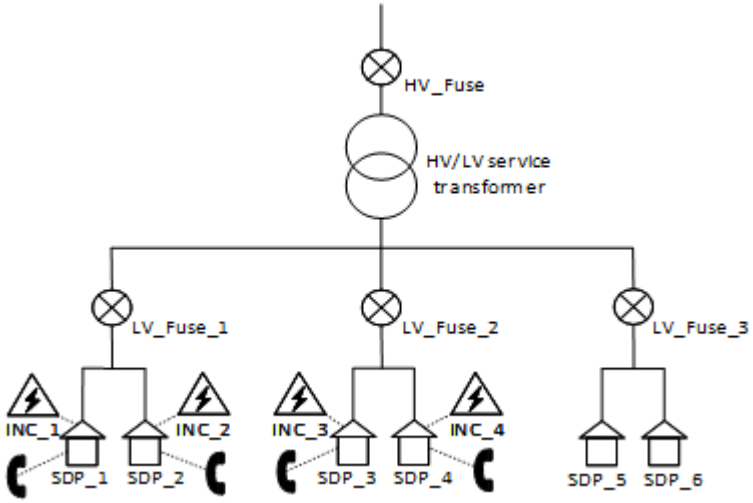

Figure 7.2 – The network diagram – the state before roll up

Following table contains list of actions related to the incident creation and ancestors populated for each incident.

Table 7.1 – Incident creation and ancestors

Incidents	Ancestor List	Action
INC_1	<empty>	Created
INC_2	<empty>	Created

Incidents	Ancestor List	Action
INC_3	<empty>	Created
INC_4	<empty>	Created

As a result of an incident creation, the following integration messages will be sent by the Outage Notification Adapter:

- 4 x CreatedIncidents messages, one per created incident.
- 4 x ChangedIncidentUsagePoints message, one per created incident.
- 4 x ChangedIncidentTroubleTickets message, one per created incident.

7.1.1. Automatic Roll-Up to the Low-Voltage Transformer Fuse

In this scenario, two pairs of incidents have the same common predicted device, which are the low voltage transformer fuses. If conditions defined by the prediction configuration are met, roll up to the first device rule will be applied automatically by the prediction engine. This is going to cause the merging of two incidents into a single incident. The actions will take place in the following sequence:

- Two incidents will roll up to upstream devices. For explanation purpose, we can say that those incidents are INC_1 and INC_3.
- Incident INC_1 and incident INC_3 devices will be updated to the new predicted devices: “LV_Fuse_1” and “LV_Fuse_2”.
- The prediction engine will populate new affected areas.
- INC_2 and INC_4 will be merged into the incidents INC_1 and INC_3, respectively.
- All calls, crew assignments and problems will be transferred from merged incidents to the parents.
- IDs of all incidents that are merged are recorded to parent incident into the ancestor incidents list.
- All merged incidents (INC_2 & INC_4) are deleted.

This situation is displayed in Figure 7.3.

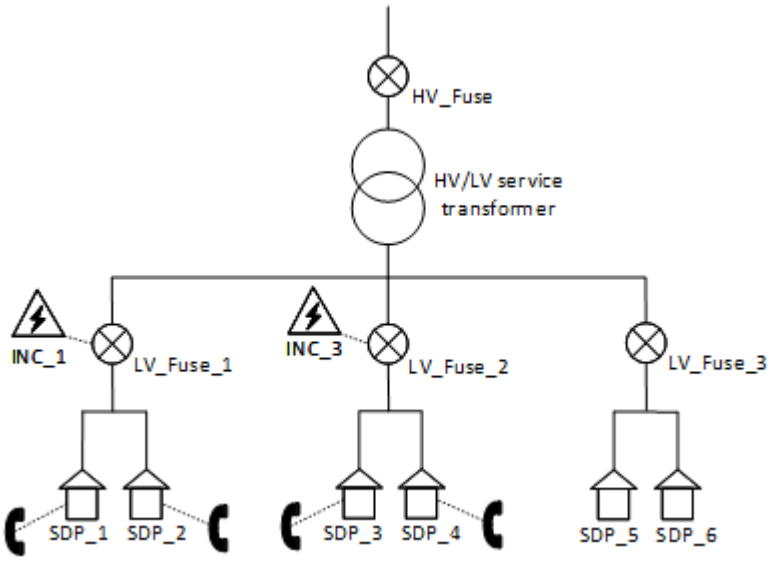

Figure 7.3 – Network diagram – State after automatic rollup

After the roll-up, incident INC_1 will be located on the low voltage transformer fuse “LV_Fuse_1”, while incident INC_3 will be located on the low voltage transformer fuse “LV_Fuse_2”. Roll up will cause a change of number of affected customers, but no customers will be lost from the affected area, just new ones may be added.

Following table contains list of actions related to the incident roll-up and ancestors populated for each incident.

Table 7.2 – Incident actions following automatic rollup

Incidents	Ancestor List	Action
INC_1	INC_2	Changed
INC_2	<empty>	Deleted
INC_3	INC_4	Changed
INC_4	<empty>	Deleted

Order of integration messages that will be sent to the external system in case of described roll up will be in accordance with the principle that first the remaining incidents are updated, and then the obsolete incidents are removed. Hence, following integration messages will be sent:

- 2 x ChangedIncidents messages for INC_1 and INC_3.
- 2 x ChangedIncidentUsagePoints messages for INC_1 and INC_3, with newly added customers.
- 2 x ChangedIncidentTroubleTickets messages for INC_1 and INC_3, with newly added calls.
- 2 x DeletedIncidents messages for INC_2 and INC_4.

7.1.2. Manual Roll-Up to the High-Voltage Transformer Fuse

As a next step, if it is determined that the device that caused the outage is further upstream the network, incidents can be manually rolled up.

The manual roll up of the incidents INC_1 will result in merging INC_3 into INC_1. The sequence of actions is the same as in the case of roll up to first predicted device described above. INC_1 is then updated with the new device which is high voltage transformer fuse “HV_Fuse”. Prediction engine will update incident area and references between affected customers and the incidents. After that, references of incidents are recorded to the parent incident and the merged incident is deleted. In this example, rolling up to the high voltage transformer fuse will cause the area affected by the incidents to widen and that newly affected customers are added to the incident.

The situation in the network after the second roll up displayed in Figure 7.4.

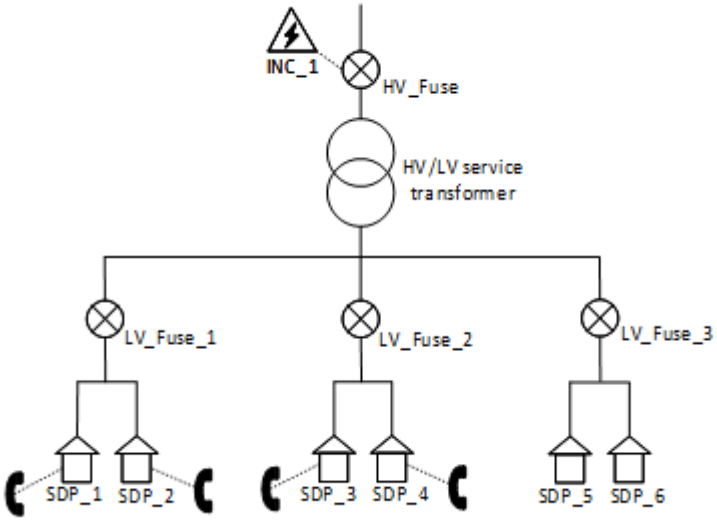

Figure 7.4 – Network diagram – State after manual roll up

Following table contains list of actions related to the incident roll-up to the “HV_Fuse” and ancestors populated for each incident.

Table 7.3 – Incident actions following manual rollup

Incidents	Ancestor List	Action
INC_1	INC_2, INC_3, INC_4	Changed
INC_3	INC_4	Deleted

These roll up will cause that following integration messages are sent:

- 1 x ChangedIncidents messages for INC_1.
- 1 x ChangedIncidentUsagePoints messages for INC_1, which will beside customers from the INC_3 include new affected customers (SDP_5 and SDP_6).
- 1 x ChangedIncidentTroubleTickets messages for INC_1, which will include calls from the INC_3.
- 1 x DeletedIncidents message for INC_3.

7.2. Manual Roll Down

When roll up of an incident is deemed to be incorrect, caused either by a premature automatic roll up or by user action, there is an option to “roll down” the incident. Roll down assumes that the incident will be moved to some device downstream. A protection device can be candidate for a new incident if there are calls submitted downstream of that protection device.

As a consequence of roll down, existing incident can either be:

- Split into two or more new incidents – in case there are at least two protection devices that are candidates for a new incident.

- Updated – in case that there is only one protection device downstream that is candidate for a new incident.

Roll down is available only for unconfirmed incidents.

Roll down of an incident is shown on an example, where at the start the incident is located on a high voltage transformer fuse “HV_Fuse”. Two calls are submitted downstream of the observed transformer, below two different low voltage transformer fuses “LV_Fuse_1” and “LV_Fuse_2”. Situation is presented in Figure 7.5.

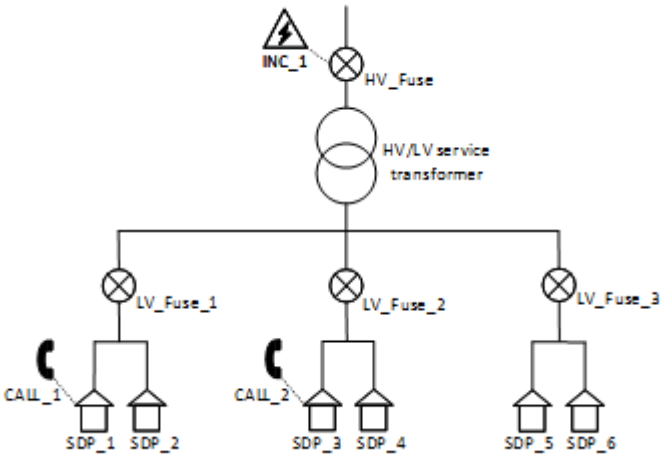

Figure 7.5 – Network diagram – State before roll down

As a result of incident creation, following integration messages will be sent by the adapter:

- 1 x CreatedIncidents message for INC_1.
- 1 x ChangedIncidentUsagePoints message, with all the affected customers downstream of the high voltage transformer fuse.
- 1 x ChangedIncidentTroubleTickets message, with all related incident calls.

If the dispatcher or the field crew concludes that devices that caused the outage are the low voltage fuses below the transformer, the incident can be rolled down to them. The result of the roll down action is that incident data is split is into two new incidents. Each of the incidents will be located at the low voltage transformer fuses where downstream calls have been submitted. All newly created incidents will have its own incident area.

After the roll down has been done, actions in EcoStruxure GridOps will be executed in the following sequence:

- Devices downstream from current device are detected and considered as candidates for incidents.
- New incident INC_2 and INC_3 are created for each detected device and that devices are set as incident devices: “LV_Fuse_1” and “LV_Fuse_2” respectively.
- Incident area is calculated for each incident independently.
- All calls and problems are set per new incident area. Crews that were associated to the parent incident are associated to all newly created incidents.
- Old incident INC_1 is deleted.

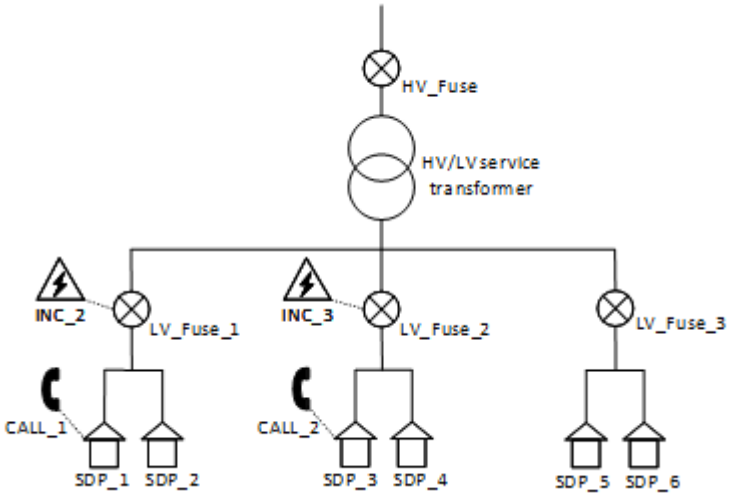

Figure 7.6 – Network diagram – State after roll down

Following table shows the operations and populated ancestor that are the result of splitting incident:

Table 7.4 – Incident actions following manual roll down (with incident split)

Incidents	Ancestor List	Action
INC_2	INC_1	Created
INC_3	INC_1	Created
INC_1	<empty>	Deleted

In this case, two new incidents are created and their list of ancestor incidents is populated with the INC_1. From the integration perspective, following messages are going to be sent by the adapter:

- 2 x CreatedIncidents messages for INC_2 and INC_3.
- 2 x ChangedIncidentUsagePoints message for INC_2 and INC_3.
- 2 x ChangedIncidentTroubleTickets message for INC_2 and INC_3.
- 1 x DeletedIncidents message for INC_1.

7.2.1. Roll Down Without Incident Split – Without Changes of the Affected Customers

When there is only one protection device downstream which is candidate for a predicted device, roll down actions simply result in an update of the original incident.

Figure 7.7 displays situation in the network before and after roll down, when the incident is just moved to the first device downstream. In this case, number of the affected customers stays the same, because incident affected area has not been changed.

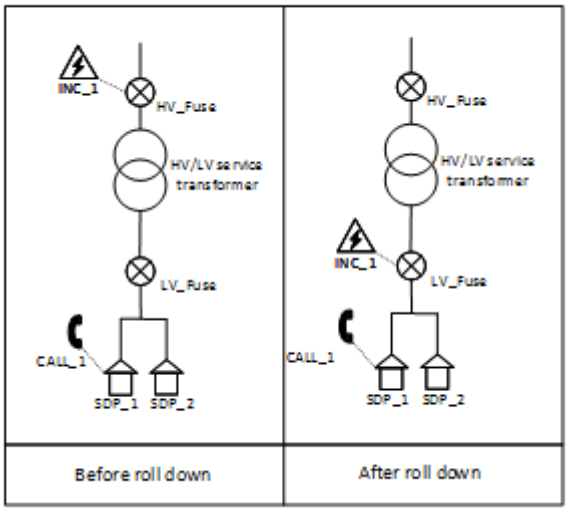

Figure 7.7 – Network diagram – States before and after the simple roll down action

Following table shows the performed operations after this kind of roll down.

Table 7.5 – Incident actions following manual roll down (without incident split)

Incidents	Ancestor List	Action
INC_1	<empty>	Changed

After the roll down, following integration message is sent by the adapter:

- 1 x ChangedIncidents messages for INC_1.

7.2.2. Roll Down Without Split – With Changes of the Affected Customers

Manual roll down operation may result in a decrease of the customers affected by the incident because affected area may become smaller after the action completion. The initial state is shown in Figure 7.8.

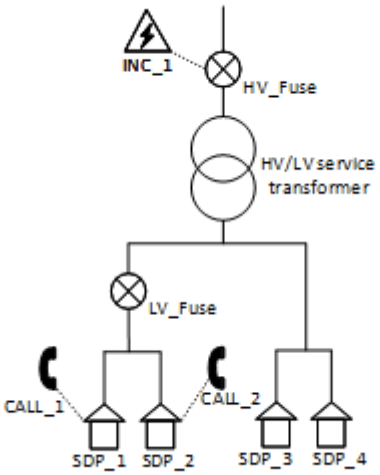

Figure 7.8 – Network diagram – State before simple roll down with customer changes

After the roll down is performed, incident is simply moved to the device downstream, which is low voltage transformer fuse “LV_Fuse”. But in this case, the number of the affected customers is decreased from 4 to 2 affected customers. This is displayed in Figure 7.9.

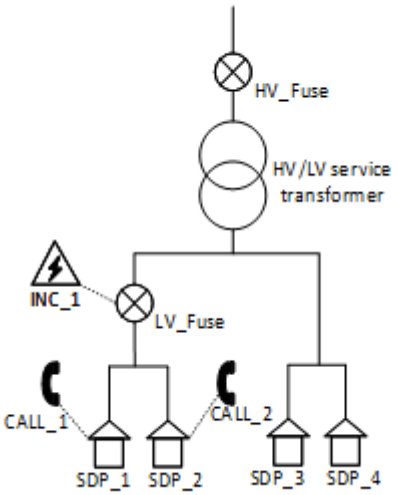

Figure 7.9 – Network diagram – State after simple roll down with customer changes

Following table shows actions that are consequence of the roll down.

Table 7.6 – Incident actions following manual roll down (without incident split)

Incidents	Ancestor List	Action
INC_1	<empty>	Changed

In this case of roll down, following integration messages are sent by the adapter:

- 1 x ChangedIncidents messages for INC_1.

- 1 x AffectedCustomers message for INC_1.

7.3. Manual Merge

In some cases, the dispatchers may want to handle multiple incidents through the same incident object. This can be achieved by merging two or more incidents into the single incident. This feature is used mainly in case of location-based calls and incidents, when the dispatcher wants to merge them to the existing incident, to have ability to use information from the merged incidents.

The manual merge is available for both unconfirmed and confirmed incidents.

In case of the unconfirmed incidents merge, only the calls are copied from one incident to the other and there are no other actions i.e. an update of incident areas, roll up, etc. The consequence of this type of merge, is that one incident is updated with the information from other incidents, while the other incident is deleted.

For demonstration of this feature, example is given in the figures below. Figure 7.10 displays the situation in the network before INC_1 and INC_2 are merged.

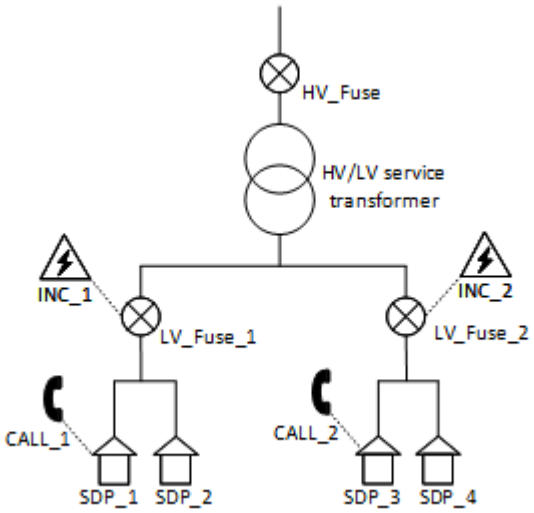

Figure 7.10 – Network diagram – State before the incident merge

7.3.1. Merge Unconfirmed Incidents

Figure 7.11 displays the situation of merging unconfirmed incidents, after the incident INC_2 is merge into the INC_1.

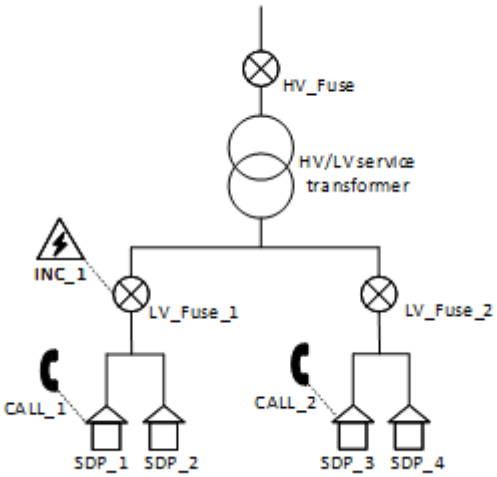

Figure 7.11 – Network diagram – State after merging unconfirmed incidents

In a case when the incidents INC_1 and INC_2 are unconfirmed, their merging will produce the following sequence of actions:

- All calls and problems are moved from the incident INC_2 to the INC_1.
- References between customers and incident INC_2 that is merged are deleted.
- INC_2 is added as ancestors for incident INC_1.
- Merged incident INC_2 is deleted.

Following table shows actions that are consequence of the incident merge.

Table 7.7 – Incident actions following merge of unconfirmed incidents

Incidents	Ancestor List	Action
INC_1	INC_2	Changed
INC_2	<empty>	Deleted

Order of integration messages that will be published in case of described manual merge of unconfirmed incidents will be in accordance with the principle that first the remaining incident is updated, and then the obsolete incident is removed:

- 1x ChangedIncidents messages for INC_1.
- 1x ChangedIncidentTroubleTickets for INC_1.
- 1x DeleteIncident message for INC_2.

7.3.2. Merge of Confirmed Incidents

From a user perspective, process for merging confirmed incidents is the same as the one described for unconfirmed incidents. Difference between these two cases are underlying incident model changes and integration messages that are sent. Figure 7.12 displays the situation of merging confirmed incidents, after the incident INC_2 is merged into the INC_1.

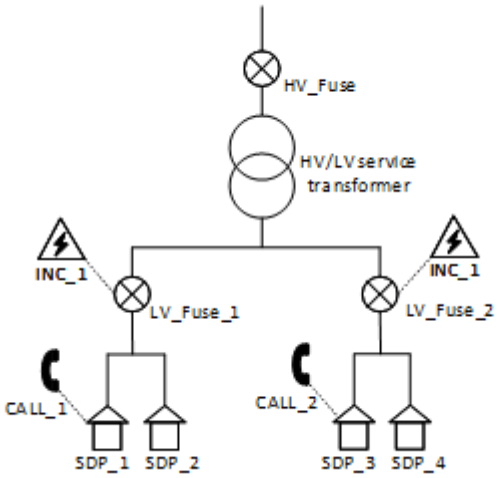

Figure 7.12 – Network diagram – State after merging confirmed incidents

In a case when incidents INC_1 and INC_2 are confirmed, their merging will produce the following sequence of actions:

- Device from merged incident INC_2 is added to the incident INC_1.
- Calls, crews and problems are moved from the incident INC_2 to the INC_1.
- References between affected customers and incident are updated for merged incident. Reference with merged incident are cleared and reference with incident that incident was merged into are obtained.
- INC_2 is added as ancestors for incident INC_1.
- Merged incident INC_2 is deleted.

As a result, after the merge one incident will have two devices, while the other incident that was merged is deleted. Also, all customers and incident area that were part of deleted incidents are now moved to the other incident.

Following table shows actions that are consequence of the incident merge.

Table 7.8 – Incident actions following merge of confirmed incidents

Incidents	Ancestor List	Action
INC_1	INC_2	Changed
INC_2	<empty>	Deleted

After the merge, following integration messages will be sent by the adapter:

- 1 x ChangedIncidents messages for INC_1.
- 1 x ChangedIncidentTroubleTickets message for INC_1.
- 1 x ChangedIncidentUsagePoints message for INC_1.
- 1 x DeletedIncidents messages for INC_2.

7.4. Manual Split

A manual split of a confirmed incident is the opposite operation from the manual merge.

In the beginning, assumption is that there is one incident with multiple devices associated. This can happen when multiple outage areas are manually or automatically merged, e.g. as a consequence of SCADA trip or switching program execution. In cases like this, there could be a need for dividing the incidents to the separate ones and for this purpose manual split feature can be used. Initial state is presented in Figure 7.13.

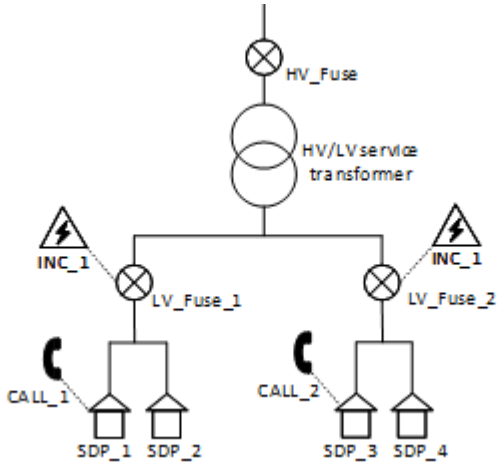

Figure 7.13 – Network diagram – State before the incident split

After the split operation is done, new incident is created and all data that was associated to area under one device (calls, problems, affected customers, etc.) is transferred to the new incident. Figure 7.14 displays the situation in the network after INC_1 is split, where “LV_Fuse_2” is selected as a device where to locate a new incident.

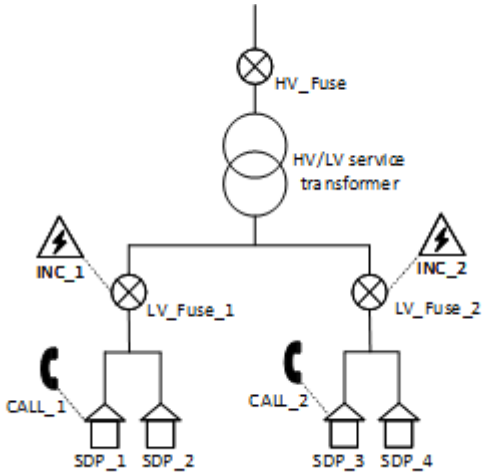

Figure 7.14 – Network diagram – State after the incident split

Splitting of the incident INC_1 is executed in the following order:

- New incident INC_2 is created.

- Selected device “LV_Fuse_2” is associated to the INC_2 and incident area is defined based on that device.
- All data that belongs to the newly obtained area is transferred from INC_1 into the INC_2.
- References between affected customers and incident are updated, so that all affected customers are under the incident area of the new incident and have reference to the new incident.
- Ancestor Incident list for INC_2 is updated.

Following table shows actions that are consequence of the incident splitting.

Table 7.9 – Incident actions following manual split

Incidents	Ancestor List	Action
INC_1	<empty>	Changed
INC_2	INC_1	Created

Order of integration messages that will be published in case of described splitting of confirmed incident will be in accordance with the principle that new incident is created first, and then that all others are updated:

- 1 x CreatedIncidents messages for INC_2.
- 1 x ChangedIncidents messages for INC _1.
- 2 x ChangedIncidentUsagePoints message for both INC_1 and INC_2.
- 1 x ChangedIncidentTroubleTickets message for INC_2.

8. DEPLOYMENT SPECIFICATION

Described in the *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification* [1].

The deployment specification is provided in the following table:

Table 8.1 – The deployment specification

Deployment Specification	
Application	AdapterONT
Critical process	Yes
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

9. INTERFACE CONFIGURATION

ONT 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 ONT adapter.

Table 9.1 – The configuration files specification

Name of the config file	Configuration File Description
AdapterOnt	Registry configuration xml file
OntFilterConfiguration	Filter configuration xml file
AdapterOnt_WebServiceConfiguration	Web service configuration xml file

Details about the structure and shared content of common interface configuration files are located in *EcoStruxure GridOps Management Suite 3.10 Enterprise Integration Platform - Functional Specification* [1].

Detailed content of aforementioned configuration files is provided within the *Configuration* folder in the *EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface.zip* file [3].

9.1. Message Filtering

Each publication message received must comply with configurable restraints which are defined in a related configuration xml file described in the *Interface Configuration Specification.xlsx* file provided within the *Configuration* folder in the *EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface.zip* file [3]. Notification message filtering is not a mandatory process. In cases where filtering is defined in the corresponding configuration file, a series of message filters must be performed on each publication. Only messages that satisfy all defined filter regulations will be processed and forwarded to external system. More details about the filtering configuration is located in the *Interface Configuration Specification.xlsx* file provided within the *Configuration* folder in the *EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface.zip* file [3].

Location of aforementioned filter configuration file is stated in registry adapter configuration xml.

If a restraint is not defined for a certain model attribute, it is considered that all values for that attribute are supported.

9.1.1. Out of the Box Message Filtering

Out of the box message filtering solution is defined to be flexible. Only restraints defined for the ONT adapter processes are made for type and subtype attributes of incident model.

Out of the box supported value for incident type is “unplanned”, and for incident subtype “sustained”.

10. APPENDIX

10.1. WSDL

The WSDL file, XSD schemas and sample messages defined according to the IEC 61968-100 for all ONT web services are provided within the *Web Service Definitions* folder in the *EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface.zip* file [3].

10.2. Message Examples

Message examples for several use cases are provided within the *Message Examples* folder in the *EcoStruxure GridOps Management Suite 3.10 Outage Management Notification Interface.zip* file [3].

11. RELEASE NOTES

The following new features related to the Product ONT Interfaces were introduced in the software, starting from version 3.8 SP1.

12. DEFINITIONS AND ABBREVIATIONS

Definition/Abbreviation	Description
ACK	Acknowledgement
ADMS	Advanced Distribution Management System
CC	Call Center
CIM	Common Information Model
CSR	Customer Service Representative
CTA	Call Taking Application
DMD	Dynamic Mimic Diagram
DMZ	Demilitarized Zone
EDS	External Dispatching System
ERS	External Reporting System
ESB	Enterprise Service Bus
FC	Field Client
IMS	Incident Management Service
IVR	Interactive Voice Response
NACK	Negative Acknowledgement
OMS	Outage Management System
OP	Outage Portal
SOAP	Simple Object Access Protocol
SDP	Service Delivery Point
WCF	Windows Communication Foundation
WS	Web Service
XML	Extensible Markup Language
XSD	XML Schema Definition