



EESSI - RINA National Information Exchange Interface (NIE)

Architecture & Design Specifications

Table of Contents

1	EESSI national information exchange overview	6
2	Common API concepts and features	8
2.1	User authentication	8
2.2	User authorization	8
2.3	Audit trail	8
2.4	Exception management	8
3	Functions and Use Cases	9
3.1	National information exchange function	9
3.1.1	Configure service	10
3.1.2	Load from national application	11
3.1.3	Store to national application	12
4	Behaviour	14
4.1	Sequence diagrams	14
4.1.1	Exchange case information with national application	15
4.1.2	Exchange document information with national application	15
4.1.3	Exchange notification & business' exception information with national application	17
4.2	State machine diagrams	17
4.2.1	Case state machine	17
4.2.2	Document sender state machine	18
4.2.3	Document receiver state machine	20
4.2.4	Subdocument sender state machine	20
4.2.5	Subdocument receiver state machine	21
5	Service contract	22
5.1	National information exchange client (NIE client)	22
5.1.1	hasCaseListener	22
5.1.2	hasDocumentListener	22
5.1.3	hasNotificationListener	23
5.1.4	fireCaseEvent	23
5.1.5	fireNotificationEvent	24
5.1.6	fireDocumentEvent	24
6	Data contract	25
6.1	Data model	25
6.2	EventType	25
6.3	CaseEventType	25
6.4	DocumentEventType	26
6.5	EventSource	26
6.6	PropertyType	27
6.7	BatchMetadata	27
7	Fault contract	28
8	Illustrative sample of a basic implementation	29

9	NIE configuration.....	30
10	Reference implementation of NIE (RINIE).....	31
10.1	RINIE behaviour	31
10.1.1	RINIE configuration.....	31
10.1.2	Cases processing.....	32
10.1.3	Documents processing.....	32
10.1.4	Notification processing.....	33
10.2	Service contract	39
10.2.1	CasesEvents	39
10.2.2	DocumentsEvents.....	39
10.3	Data contract.....	40
10.3.1	CaseEventType	40
10.3.2	DocumentEventType	41
10.3.3	PropertyType	41
10.3.4	RESCaseEvent	41
10.3.5	RESDocumentEvent.....	41
10.3.6	RESTNotificationEvent	42
10.4	RINIE events samples	42
10.4.1	CasesEvents	42
10.4.2	DocumentsEvents.....	44
10.4.3	NotificationsEvents	49

Document Control Information

Document Control	Value
Project Title	Electronic Exchange of Social Security Information (EESSI)
Document Title	EESSI – RINA - National Information Exchange Interface (NIE)
Document Category	Architecture & Design Specifications
Revision	-
Component Version	-
Last Publication Date Project Milestone	22/12/2020 EESSI-2020
Document Status	Final
Sensitivity (TLP) Distribution terms	Traffic Light Protocol (TLP) = "GREEN"  The distribution of this document is done strictly in line with the Traffic Light Protocol (TLP) established by the European Commission's note AC 790/15 REV for the EESSI project documentation. In line with the note AC 790/15 REV, this document is labelled as TLP = "Green". Therefore, it can be circulated widely within the EESSI community. However, the document or the information herein may not be published or posted on the Internet, nor released outside of the EESSI community.
Connected/Embedded Files	None
Authors	European Commission, DG EMPL F5, EESSI ARCH/RINA
Revised by	European Commission, DG EMPL F5, EESSI QA
Approved by	European Commission, DG EMPL F5, EESSI PM

Document history

Project Milestone /Released Component version	Changes/Corrections Description
EESSI 2019/ RINA 5.6.2	Update RINIE document template
	Updated port reference of RINIE service (page 35)
EESSI 2020/ RINA 6.2.1	Update the content to new RINA developments and architecture
	Replace the general introductory description to be aligned with "EESSI – RINA – Architecture Overview" (Chapter 1)
	Replace the NIE configuration/implementation screenshots with new ones from the New RINA Portal

1 EESSI national information exchange overview

Part of the National Institutions Domain, RINA (Reference Implementation for National Applications) consists in a collection of infrastructure and communication services, foundation, repository and publishing services, business, integration and user interface services which will provide for clerks and their organisations the tools to implement the International Protocol of Data Exchange based on Structured Electronic Documents in Social Security belonging to European Community Member States and Associated States.

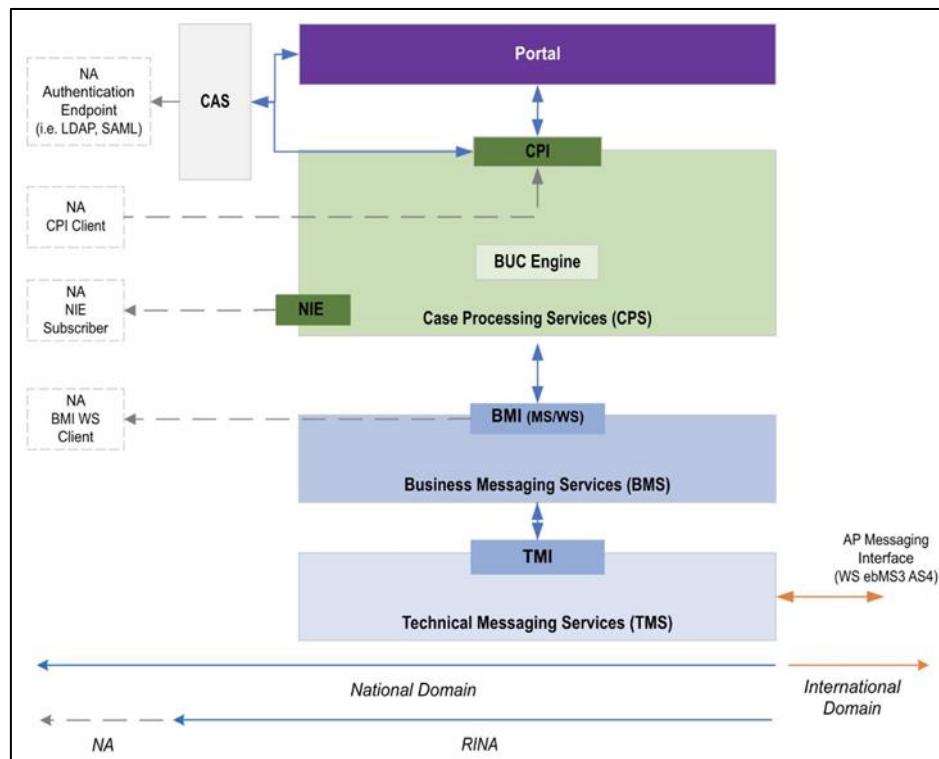


Figure 1 - Overview of RINA interfaces and layers

RINA is based on 4 layers which are built one on top of the other, plus the CAS module:

- *Technical Message Services (TMS)* - provide the mechanism for sending/receiving technical messages to the Access Point. It works based on the protocol ebMS3.0 - AS4 profile. It is an internal layer and cannot be used by third parties, being its unique client the RINA BMS.
- *Business Messaging Services (BMS)* - provide a reusable component for sending business messages to the Access Point via the TMS layer. These services provide the translation (business message to technical message and vice versa), transformation (converting IDs to GUIDs), validation and signing of the business messages for correct exchange with the EESSI environment, and the correct reception of messages from other institutions. It offers the BMI WS interface making available the integration of third party applications to the EESSI ecosystem.
- *Case Processing Services (CPS)* - provide a state full component built on top of the Business Messaging Services that manage cases in a structured manner taking care of all the issues regarding case flow, documents, notifications, user management and provides two interfaces for external access (NIE and CPI)

- *Portal* – provides an UI build on top of the Case Processing Services (CPS) having administration consoles and case Processing modules
- *CAS Authentication Server* – the SSO (Single Sign On) server providing the authentication to users/clients requesting to use a service

The scope of this document is to provide a specification document for the National Information Exchange Interface, part of the Case Processing Services (CMS) layer.

Here is a short description of how this document is structured

- Interface overview – this chapter, an introduction to what this interface is about, its place and why it is needed;
- Functions and use cases – a presentation of high level functions and use cases that drive this service component;
- Sequence diagrams – behavioural diagrams that describes the flow between components;
- Service interfaces – a description of the interfaces provided;
- Service contract – a description for all the methods exposed through the interfaces;
- Data contract – the data model used for this service;
- Fault contract – the exception model used for this service;
- Samples – some samples that shows how to use this library;
- NIE configuration – illustrative example of the configure events options;
- Reference implementation of NIE (RINIE) – a description of the reference implementation of the National Information Exchange which is done as the RESTful services.

This software component provides a JAVA interface that allows national applications to subscribe for specific events during the life cycle of a case or a document together with the notifications and business' exception associated to them. When that event happens, the national application is notified together with all the information details regarding the state of the case, or document, or with the notifications or business' exceptions associated with the former.

2 Common API concepts and features

2.1 User authentication

N/A

2.2 User authorization

N/A

2.3 Audit trail

As the National Information Exchange works in close collaboration with the Case Management Interface. Thus, all information related to its audit trail can be found in the *EESSI - RINA - Case Processing Interface (CPI)* document. Full information on RINA audit trail architecture is available in the *EESSI – RINA – Architecture Overview* document. Both documents are published and available at the RINA Architecture Documentation Confluence space:

<https://citnet.tech.ec.europa.eu/CITnet/confluence/display/EESSI/RINA++Architecture+Documentation>

2.4 Exception management

This section explains how the library manages unexpected behaviour.

When an error occurs during an API call the library will throw a Java exception that is defined in the Fault contract.

There are two generic types of exception that might occur

- Case processing exception – this kind of exception is thrown by the member states during the national application case handling when case processing cannot be done
- Document processing exception – this kind of exception is thrown by the member states during the national application document handling when document processing cannot be done

More information related on how RINA deals with Error and Exception management as well as its logging architecture can be found in the previously mentioned *EESSI – RINA – Architecture Overview* document

3 Functions and Use Cases

The national information exchange function is part of a broader set of national functions.

The national part of RINA supporting functions provide the system functionalities that will need to be provided at Member State level in order to complete the functioning of the RINA.

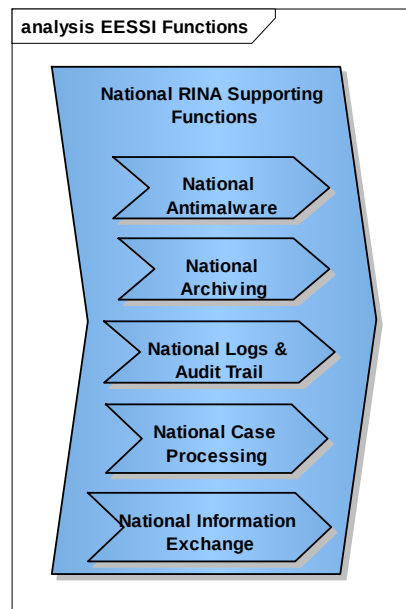


Figure 2. RINA Supporting Functions

This document specification is focused on the national information exchange function.

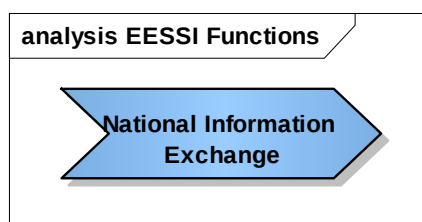


Figure 3. National Information Exchange Function

3.1 National information exchange function

The national information exchange function allows RINA to interact with national applications in order to exchange or receive information in the context of case management (e.g. for completing SEDs or exchanging SEDs with national systems).

The EESSI side is in charge to call the National Application for all the important events from the case or document (individual or in batch) lifecycle together with the notifications, business' exception generated and provide the relevant information (for example the actual SED part of a document).

It will be up to the MS to choose the way to implement the processing (for example store the SED along with other national useful information or give back an updated SED filled in with information that would be otherwise hard or cumbersome for a clerk to fill in manually)

The following figure shows the use case diagram for this function:

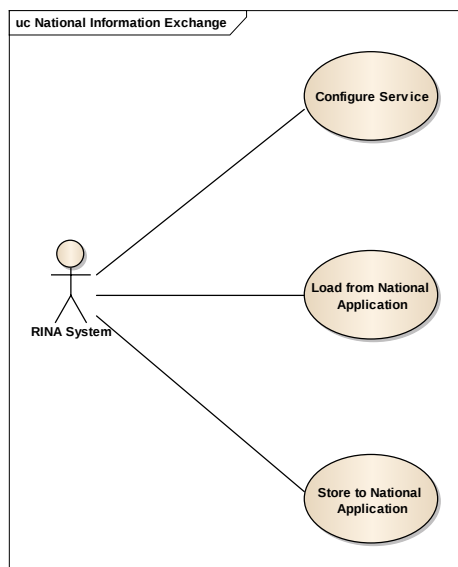


Figure 4. National information exchange use case

3.1.1 Configure service

Description

The user must be able to configure the national information exchange EESSI side component so that it can talk to any member state implementation.

- At configuration time, the RINA administrator uploads the member state implementation into RINA;
- The RINA system scans the library uploaded for the needed pieces to operate on and loads them all.

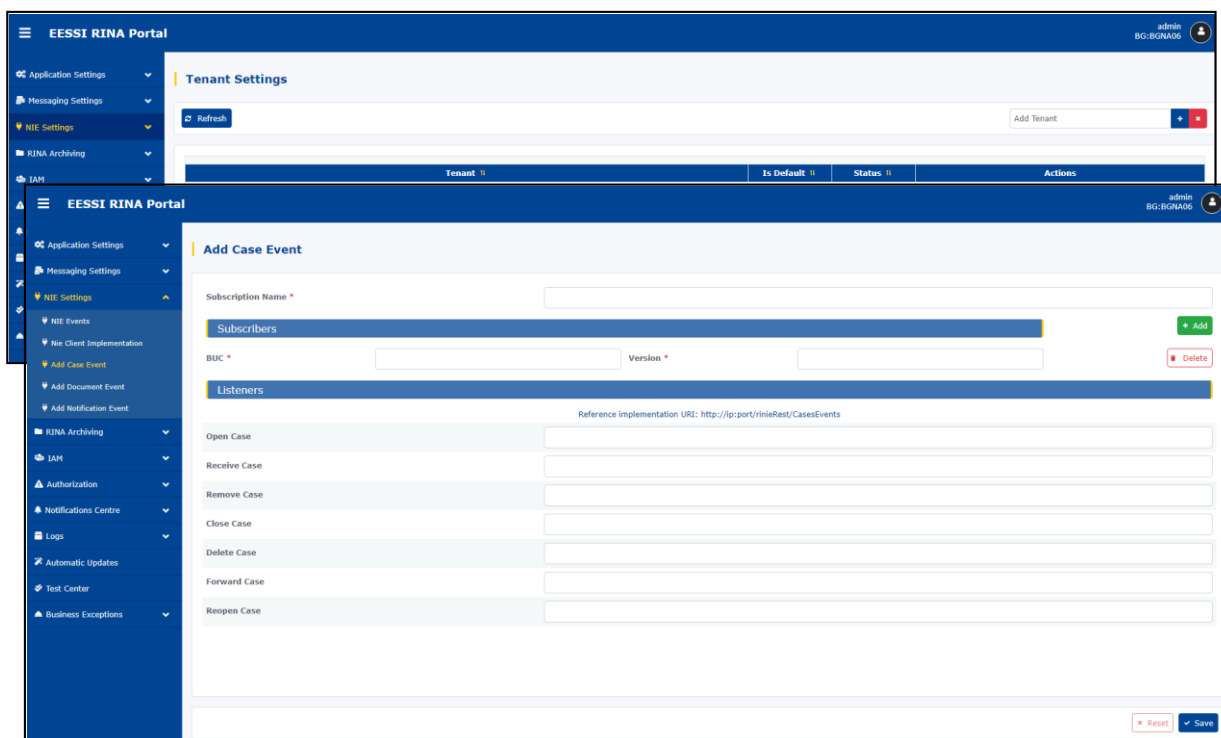


Figure 5 - Administrative Portal - NIE implementation configuration

Related to Figure 5, the Administrator has to upload the JAR file with the implementation. After this the class name which shall be instantiated shall be completed with the use of *Update the NIE settings* as described in the "*EESSI – RINA - Case Processing Interface (CPI)*" and "*EESSI – RINA - Case Processing (CPI) – Reference*" documents.

3.1.2 Load from national application

Description

The clerk may have a lot of information to fill in the SED before sending it, so there should be an automatically way to achieve this shortcoming of data re-entry.

In order to avoid redundancy of the information introduced by the clerk, which were already in the NA, at specified times (e.g., when a document has been initialised, before saving a document, etc.) the Guidance Service, triggers specific events by which the NIE is retrieving relevant information from NA and provide it to the Guidance Service for further use.

This use case is also useful for the batch documents. Through the import function a clerk can import a file, containing a batch of items (e.g. claims), and the individual items can be enriched automatically with needed / relevant information through NIE.

As an illustrative example the below use case covers this issue and describes how the EESSI system interacts with the National Application so that it can read the necessary information from the member state backend.

- The clerk prepares to send a new SED to another institution (for example he wants to send a pension P2000 SED);
- The clerk presses the "Create P2000" button ;
- The system builds the parameter context by putting the relevant information (user identification, case identification, correlation information, all the document metadata needed to track down the document);
- The system calls the national application implementation handing over the event source information (event just happened, business use case, SED type) and the context built at the previous step;
- The national application receives the event source and the information associated with the document and does whatever processing it needs to update the SED with all the fields it can automatically find in its backend system(s);
- The context is returned back to the RINA system and the SED updated with as many fields as the national application could find;
- The clerk may fill in more fields in the SED then sends the SED.

Support API

5.1 National information exchange client (NIE client)

3.1.3 Store to national application

Description

The member states have their own systems that need to be informed when specific events happen in the case management engine and a generic way of letting know about it should be supported.

The national applications should be aware about all important events from the case and document lifecycle including the notifications sent between the parties and the Business exceptions occurring during different transactions.

For the case scenario, the following events could be notified along with all the case information:

- When the case is opened;
- When a new case is received from the counter party;
- When the case is closed;
- When the case is deleted;
- When the case is forwarded to a counter party and removed from the local process owner.
- When a participant is removed from a case.

For the document scenario, the following events should be notified along with all the document information:

- When the SED has been initialized;
- When the SED is newly created;
- When the SED is updated;
- When the SED is sent;
- When the SED is deleted;
- When the SED is invalidated;
- When the SED is received.

Notifications and Business exceptions can be configured in the Notifications Centre and consists of the following type:

- Case SEDs Automatically Sent to Participant
- Approval for Sending Required for SED
- SED not matching Case
- A new SED Arrived
- Your Alarm Expired
- Request to Assign Case
- Request to Assign Case Rejected
- Request to Assign Accepted
- Case Unassigned
- Case Assigned

- An Updated SED Arrived
- SED Delivered
- A New Case Arrived
- SED Failed to be Delivered
- Case Automatically Closed
- Archiving Case Exception
- Archiving Case Problem
- Case Assignment Exception
- Business Exception - Case Forwarded
- Business Exception - Attachment failed antimalware checking
- Business Exception - Case Closed
- Business Exception - Unknown Cause
- Business Exception - Invalid Business Signature
- Business Exception - SED in Wrong Sequence
- Business Exception - SED update without Create
- Business Exception - Case Removed
- Business Exception - Case Missing

Additionally, for the batch document scenario, the following events should be notified along with all the document information:

- When a part of a batch SED has been initialized;
- When a part of a batch SED is newly created;
- When a part of a batch SED is updated;
- When a part of a batch SED is deleted;
- When a batch of parts of a batch SED is created;
- When a batch of parts of a batch SED is received.

Finally for different type of RINA's notifications and business' exceptions, the admin can choose if they want that these events should be notified along with the full notification's details.

Support API

5.1 National information exchange client (NIE client)

4 Behaviour

4.1 Sequence diagrams

The sequence diagrams show the interaction between the EESSI case management system and the national applications when exchanging information about a case or a document.

All the diagrams have the following internal components:

- Guidance Service – this component implements the national information exchange processing part and it does this by having direct contact with the case management system and with the national information exchange client;
- National Information Exchange Client – this component is used for registering national application clients interested in receiving notifications about various events that get fired in the case management engine during the life cycle of a case or a document;
- National Information Exchange API (ICD5) at the National Application – this component is actually the member state's national application that registers itself for case management event and handles (loads, store, all kind of processing) the events triggered in the case engine;

Next are presented the steps that need to be performed so that the national application receives a notification:

- The Guidance Service triggers the `hasCaseListener` / `hasDocumentListener` / `hasNotificationListener` methods which checks if there is any applicable situation for which needs to trigger the `fireCaseEvent` / `fireDocumentEvent` / `fireNotificationEvent`;
- For a case event, the list of business use cases is also provided so that notifications are sent for each case event - business use case combination;
- For a document event, the list of business use cases and the SED types are provided so that document notifications are sent for each document – business use case – SED type combination;
- For the notification event, the full notification's details are provided.
- When an event happens, the Guidance Service checks whether there is any listener registered. No case information context will be loaded if no listener is registered.

4.1.1 Exchange case information with national application

Diagram

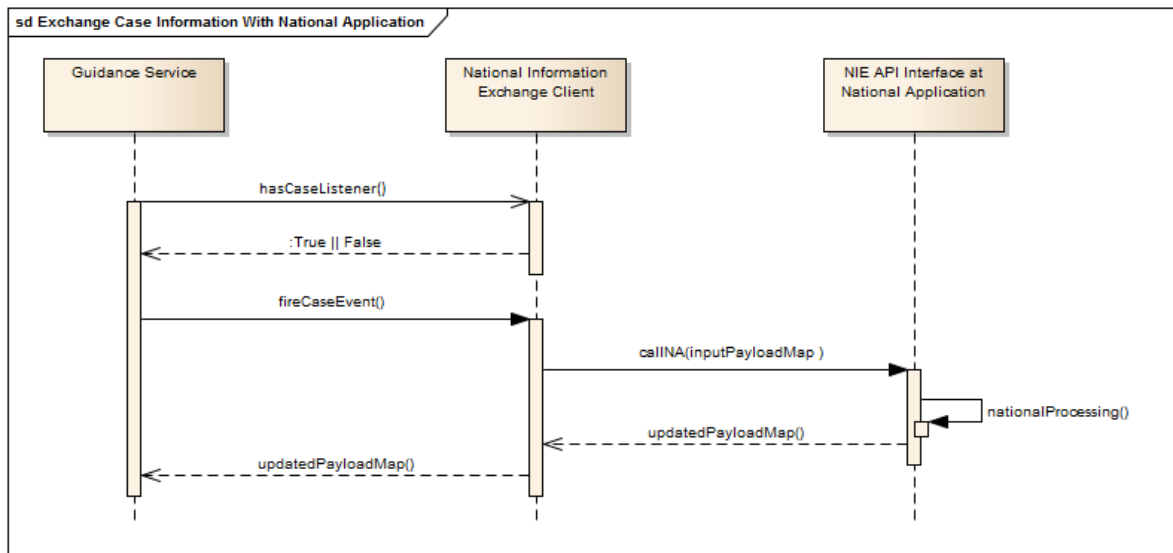


Figure 6. Sequence diagram - Exchange Case Information with National Application

Description

The sequence diagrams show the interaction between the EESSI case management system and the national applications when exchanging information about a case.

The following events can be triggered for this sequence diagram:

- When the case is opened;
- When the case is received from the counter party;
- When the case is closed;
- When the case is deleted;
- When the case is forwarded to a counter party and removed from the local process owner.

Once the event is triggered the Guidance Service uses the National Information Exchange Client to inform about the event source and the case context that is passed as an input payload map. For the case scenario, the payload map has only one entry, namely the RESTCase object that contains all the details needed for a case.

The National Information Exchange Client calls the NA, and provides the relevant information related to the current event and the NA needed data.

In the "Call" method, the national application does all the processing it needs to fulfil its requirements (store the case, users related to it, etc.). After all the member state processing is done it may return the updatedPayloadMap .

4.1.2 Exchange document information with national application

Diagram

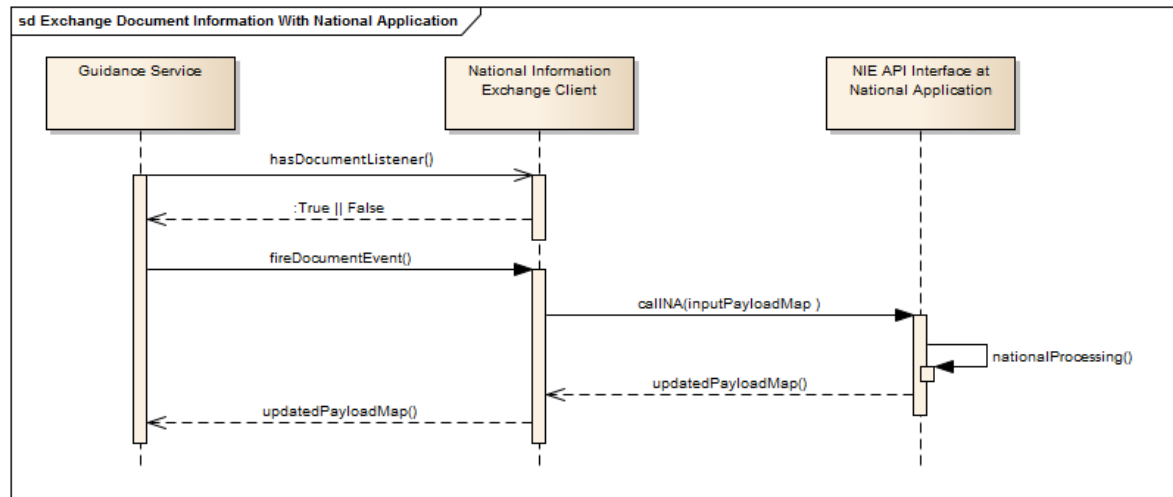


Figure 7. Sequence diagram - Exchange Document Information with National Application

Description

This sequence diagram shows the interaction between the EESSI case management system and the national applications when exchanging information about a document.

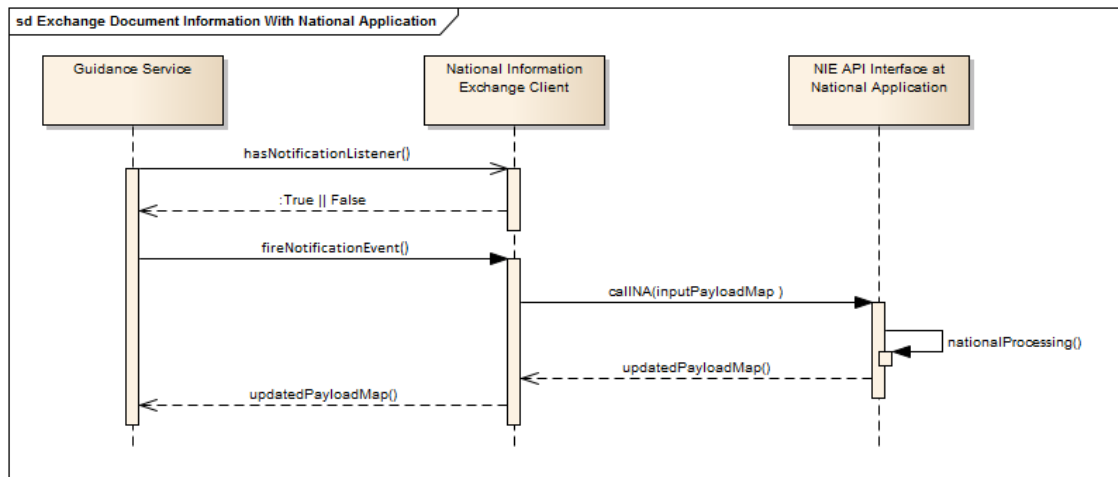
The following events can be triggered for this sequence diagram:

- When the document has been initialized;
- When the document is newly created;
- When the document is updated;
- When the document is sent;
- When the document is deleted;
- When the document is cancelled;
- When the document is received;
- When a part of a batch document has been initialized;
- When a part of a batch document is newly created;
- When a batch of parts of a batch document is created;
- When a part of a batch document is updated;
- When a part of a batch document is deleted;
- When a batch of parts of a batch document is received;

Once the event is triggered the Guidance Service uses the National Information Exchange Client to inform about the event source and the case context that is passed as an input payload map. For the document scenario, the payload map has two entries: the first entry is the Document Metadata that contains all the metadata for the document (what case is related to, the document type, who created, etc.) and the second entry is the actual SED payload.

The National Information Exchange Client calls the National Information Exchange API (ICD5) and provides the relevant information related to the current event and the NA needed data. In the method, the national application does all the processing it needs to fulfil its requirements (store the case, some users about it, etc.).

After all the member state processing is done it returns the updatedPayloadMap that in some cases (for example on initialized) should have lot of fields updated for the SED.



4.1.3 Exchange notification & business' exception information with national application

Depending on the option (generateNIE or not) chosen by the admin for different type of notifications & Business' exceptions in the Notification Centre, the system will share these details with the National Application.

4.2 State machine diagrams

The state machine diagrams presented below are also described in the EESSI Case Processing Interface (CPI) specification document (same chapter 4 Behaviour – state machines) as these views are important in understanding under what conditions information is exchanged with the national application.

4.2.1 Case state machine

The case state machine is used to keep track of the case state.

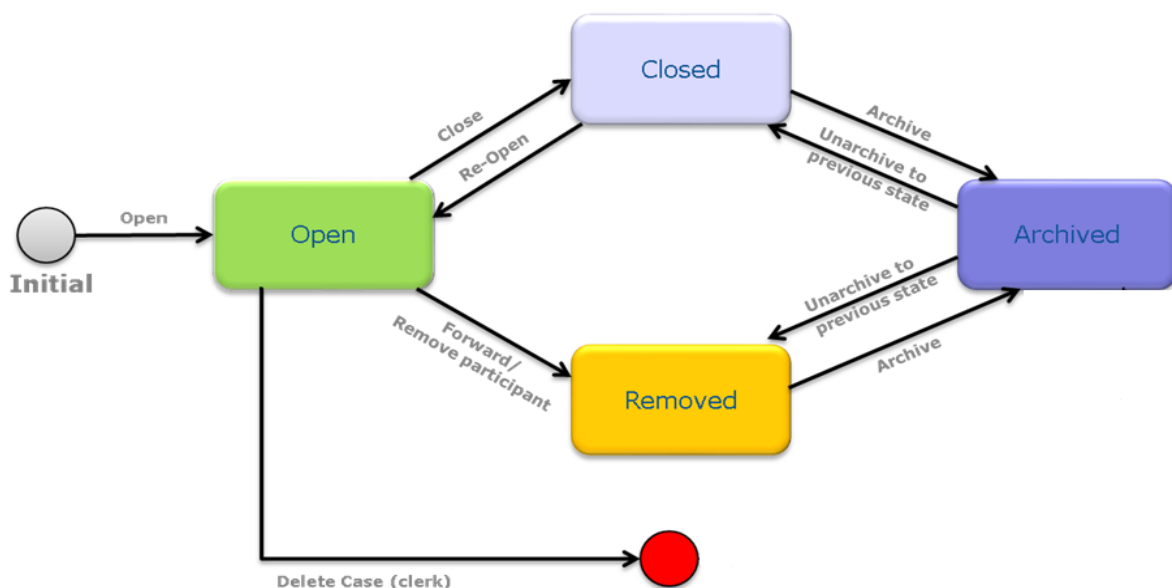


Figure 8. Case state machine

As illustrate in the Figure 8 above, during the transition a specific event is triggered so that case information at that moment is sent to the national application:

- **Open** – In this state the sender party creates the case and the receiver party opens the case for the first time. After all the work is done on this case the user may close the case. If other counter party should be responsible for this case from now on the user forwards the case. If the user is not interested anymore in the case he deletes the case. The case can be deleted if there are no SEDs exchanged;
- **Closed** - The user closed the case, from this state the case can be opened again; also un-archiving a case which was closed will bring the case from archived back to the closed status;
- **Removed** - The case has been forwarded to another counter party or participant has been removed and is no longer owned by the sender, therefore the case will be read-only;
- **Archived** – The case is archived either automatically following normal archiving process following case closing, case forwarding, or case removed participant (according to the defined timers for respective case's BUC Type) either manually by the clerk.

4.2.2 Document sender state machine

The document state machine is used to keep track of the document state on the sender side.

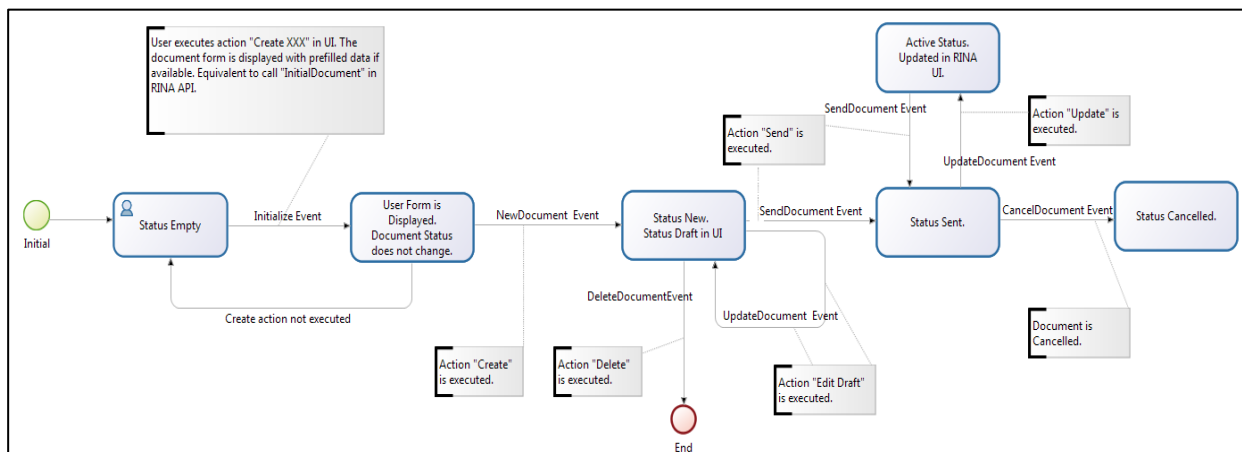


Figure 9. Document State Machine

Note: Being physically deleted in the state machine is not reflected in the state machine. This can be represented in UML with a terminated event; for the diagram clarity this was not represented.

As illustrate in the Figure 9 above, during the transition a specific event is triggered so that document information from that particular moment is exchanged with the national application:

- Empty

- This is the *Empty* state where the document metadata has just been initialized, *Create* action for the document is available and it has no content. Users with create rights can see the corresponding action *Create*.
- When **Create action** in the **UI** is triggered, **Initialise Document Event** described in the document event types - *DocumentEventType*. Triggering Create Action in UI does not change the status of the document, because it only retrieves the *initial document*. The *real* Create Action in RINA is only executed in UI if **save button is pressed**.
- By executing directly *RINA Create Action* by using RINA API, not portal, the event **Initialise Document Event** will not be triggered. The document content will be saved directly and event **NewDocument** will be triggered.
- **New**
 - When **save** button in the UI is pressed or the create action is executed directly using RINA API, the document will go to state **New**. In RINA Portal this status is displayed as **Draft**. Then the *NewDocument* event is triggered.
 - *NewDocument* event described in the document event types - *DocumentEventType* ;
 - Document in **New** state can be updated several times. In RINA portal the document status is displayed as **Draft** until the document is sent or deleted. In each of these updates an event *UpdateDocument* will be triggered. The document will not change status.
- **Deleted**
 - Before a document is sent, while in status New, it may be deleted. A deleted document is removed from the system and a new Create Action for same document is available. When a document is deleted a *DeleteDocument* event is triggered.
 - The *DeleteDocument* event described in the document event types - *DocumentEventType* ;
- **Sent**
 - When the **Send** action is executed, document is sent, from this state the document can be updated, cancelled or forwarded depending on the BUC. The document goes to status **Sent**. A *SendDocument* event is triggered.
 - The *SendDocument* event described in the document event types - *DocumentEventType* ;
- **Updated**
 - Once a document is in status *Sent*, depending on the BUC, may be updated. If it is updated the document will do to status **Active** in this status the document can be updated until document is cancelled. In RINA Portal the status will be displayed as **Updated**. An *UpdateDocument* event will be triggered every time the action **Update** is executed.
 - The *UpdateDocument* event described in the document event types - *DocumentEventType* ;
- **Cancelled**
 - A **Sent** document may be cancelled if AD_BUC_06_Subprocess-Invalidate_SED is required for that document. If the document is cancelled it goes to status **Cancelled** and event *CancelDocument* is triggered. This is the final state.
 - The *CancelDocument* event described in the document event types - *DocumentEventType*.

4.2.3 Document receiver state machine

The document state machine is used to keep track of the document state on the receiver side.



Figure 10. Document receiver state machine

As illustrated in the Figure 10 above, during the transition a specific event is triggered so that document information from that particular moment is exchanged with the national application:

- **Received**
 - When a document is received from another participant a *ReceiveDocumentEvent* is triggered.
 - The *ReceiveDocument* event described in the document event types - [DocumentEventType](#);
- **Updated**
 - Once a document is in status *Received*, depending on the BUC, new versions of the same document may be received. An *UpdateDocument* event will be triggered every time a newer version of a received document is received.
 - The *UpdateDocument* event described in the document event types - [DocumentEventType](#) ;
- **Cancelled**
 - A received document may be cancelled if AD_BUC_06_Subprocess-Invalidate_SED is required for that document. If the document is cancelled it goes to status **Cancelled** and event *CancelDocument* is triggered. This is the final state.
 - The *CancelDocument* event described in the document event types - [DocumentEventType](#).

4.2.4 Subdocument sender state machine

This applies only to *bulk* documents that may have attached subdocuments. Those for documents don't have a status and the associated NIE events depend on the action performed on them. Based on this actions:

- **Add Subdocument**
 - A new subdocument is added to a bulk document. An event *NewSubdocument* is triggered. Adding subdocuments is only allowed for documents in status **New**.
 - The *NewSubdocument* event described in the document event types - [DocumentEventType](#);

- **Update Subdocument**
 - Any update on an already added subdocument triggers a event *UpdateSubdocument* event.
 - The *NewSubdocument* event described in the document event types - [DocumentEventType](#);
- **Delete Subdocument**
 - When a subdocument is deleted an *DeleteSubdocument* event is triggered. Removing subdocuments is only allowed for documents in status **New**.
 - The *DeleteSubdocument* event described in the document event types - [DocumentEventType](#);

Warning: The action **Import Subdocuments** triggers a corresponding event, *NewSubdocument* if new or *UpdateSubdocument* if existing per each imported subdocument.

4.2.5 Subdocument receiver state machine

This applies only to *bulk* documents that may have attached subdocuments. Those for documents don't have a status and the associated NIE events depend on the action performed on them. Based on this actions:

- **Receiving a new Subdocument**
 - A new subdocument is added to a bulk document. An event *ReceiveDocumentBatch* is triggered.
 - The *ReceiveDocumentBatch* event described in the document event types - [DocumentEventType](#);

5 Service contract

5.1 National information exchange client (NIE client)

The NIE Client is used for receiving notifications about various events that get fired in the case management engine during the life cycle of a case or a document.

Before any call to the `fireCaseEvent`, `fireDocumentEvent` or `fireNotificationevent`, the `hasCaseListener`, the `hasDocumentListener` and the `hasNotificationListener` methods are triggered in order to monitor if there are any cases, documents or notifications to be updated.

At a new NIE client implementation it will be up to the MS to choose the way on how the NIE Client configurations will be managed.

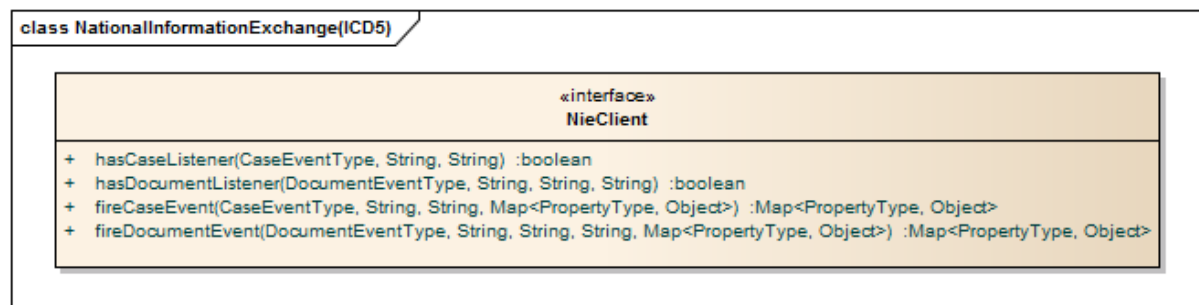


Figure 11. National Information Exchange interface

5.1.1 hasCaseListener

Name	hasCaseListener
Description	Checks whether there is a listener registered for a case business used case combination
Preconditions	N/A
Input Parameters	6.3<i>CaseEventType</i> – The case event type for which the listener is registered for <i>String</i> – The business use case for which the listener is registered for <i>String</i> – The version of business use case for which the listener is registered for
Output Parameters	N/A
Exceptions	N/A

5.1.2 hasDocumentListener

Name	hasDocumentListener
Description	Checks whether there is a listener registered for a document, business used case, SED type combination

Preconditions	N/A
Input Parameters	<i>DocumentEventType</i> – The document event type for which the listener is registered for <i>String</i> – The business use case for which the listener is registered for <i>String</i> – The version of business use case for which the listener is registered for <i>String</i> – The SED type for which the listener is registered for
Output Parameters	N/A
Exceptions	N/A

5.1.3 hasNotificationListener

Name	hasNotificationListener
Description	Checks whether there is a listener registered for notifications.
Preconditions	N/A
Input Parameters	<i>none</i>
Output Parameters	N/A
Exceptions	N/A

5.1.4 fireCaseEvent

Name	fireCaseEvent
Description	Checks whether there is a listener registered for a case type and business use case combination
Preconditions	N/A
Input Parameters	<i>CaseEventType</i>– The case event type for which the listener is registered for <i>String</i> – The business use case for which the listener is registered for <i>String</i> – The version of business use case for which the listener is registered for <i>Map<PropertyType, Object></i> –Payload map has only one entry, namely the RESTCase object that contains all the details needed for a case
Output Parameters	<i>Map<PropertyType, Object></i> – The updated payload map or the original input payload map if no updates were made

Exceptions	<i>CaseProcessingException</i> – Exception thrown during the case national application processing
-------------------	---

5.1.5 fireNotificationEvent

Name	fireNotificationEvent
Description	Checks whether there is a listener registered for a notifications event.
Preconditions	N/A
Input Parameters	<i>Map<PropertyType, Map<String, Object>)</i> –Payload map has only one entry, namely the notification object that contains all the full details needed for a notification.

5.1.6 fireDocumentEvent

Name	fireDocumentEvent
Description	Checks whether there is a listener registered for a document event, business use case, SED type combination
Preconditions	N/A
Input Parameters	<i>DocumentEventType</i> – The document event type for which the listener is registered for <i>String</i> – The business use case for which the listener is registered for <i>String</i> – The version of business use case for which the listener is registered for <i>String</i> – The SED type for which the listener is registered for <i>Map<PropertyType, Object></i> – Payload map has two entries: the first entry is the RESTDocument that contains all the metadata for the document (what case is related to, the document type, who created, etc.) and the second entry is the actual SED payload For the events where the SED payload is a batch of document parts, there is a 3rd map entry: the metadata of the batch, containing the batch size and the first part no (ordinal number). *Note: The payload needs to be populated according to the expected Schema described under the CDM artefacts document "EESSI_JSON_TEMPLATE_AND_SCHEMA (chapter 2: SCHEMA files)"
Output Parameters	<i>Map<PropertyType, Object></i> – The updated payload map or the original input payload map if no updates were made
Exceptions	<i>DocumentProcessingException</i> – Exception thrown during the document national application processing

6 Data contract

Additional details related to Data Contract used in NIE can be found in the Case Processing Interface specifications (i.e., RESTCase, RESTDocument, etc.)

6.1 Data model

Diagram

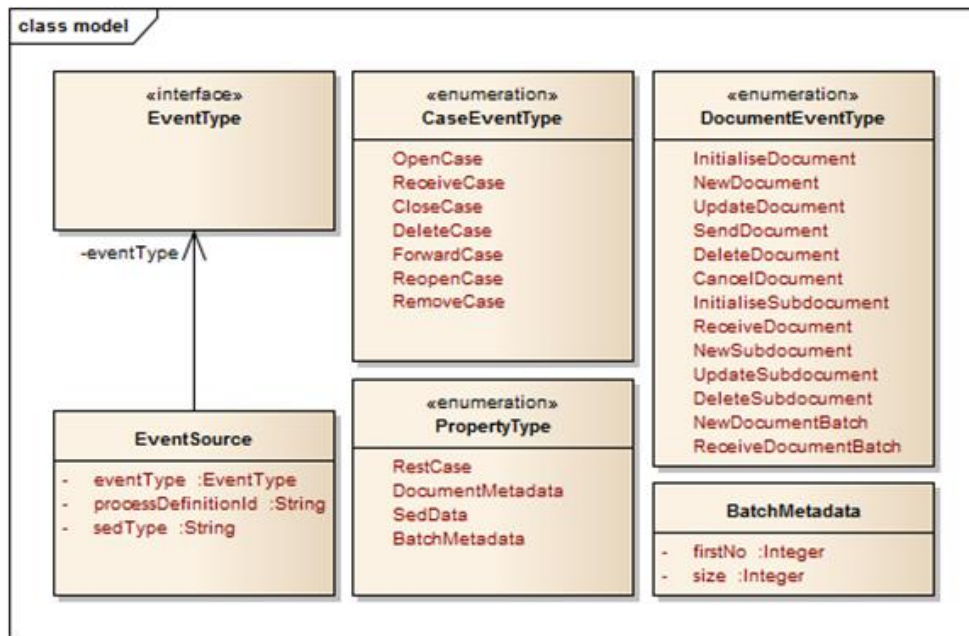


Figure 12. Data model

6.2 EventType

This is a marker interface for case and document event types

6.3 CaseEventType

CaseEventType tracks down the state where the case is in that moment. The case can be opened, received, closed or deleted.

Attribute	Description	Data Type
OpenCase	The case is opened as PO, unidirectional	CaseEventType
ReceiveCase	The case is received unidirectional	CaseEventType
RemoveCase	The case is removed, unidirectional	CaseEventType
CloseCase	The case is closed, unidirectional	CaseEventType
DeleteCase	The case is deleted, unidirectional	CaseEventType

ForwardCase	The case is forwarded to the counter party and the local process does not have ownership anymore, unidirectional	CaseEventType
ReopenCase	The case is reopen, unidirectional	CaseEventType

6.4 DocumentEventType

Attribute	Description	Data Type
InitialiseDocument	The document is initialized, bidirectional	DocumentEventType
NewDocument	The document is new, bidirectional	DocumentEventType
UpdateDocument	The document is updated, bidirectional	DocumentEventType
SendDocument	The document is sent, unidirectional	DocumentEventType
DeleteDocument	The document is deleted, unidirectional	DocumentEventType
CancelDocument	The document is cancelled, unidirectional	DocumentEventType
ReceiveDocument	The document is received, bidirectional	DocumentEventType
InitialiseSubdocument	The subdocument is initialized, bidirectional	DocumentEventType
NewSubdocument	The subdocument is created, bidirectional	DocumentEventType
UpdateSubdocument	The subdocument is updated, bidirectional	DocumentEventType
DeleteSubdocument	The subdocument is deleted, unidirectional	DocumentEventType
NewDocumentBatch	A batch of subdocuments is created, bidirectional	DocumentEventType
ReceiveDocumentBatch	A batch of subdocuments is received, bidirectional	DocumentEventType

6.5 EventSource

EventSource contains all the information needed for the client being notified to understand the source of the event fired: type of the event (case or document), the business use case and the SED type (only needed for the document event type):

Attribute	Description	Data Type
eventType	The type of the event either case or document	EventType

processDefinitionId	The business use case	String
sedType	The SED type	String

6.6 PropertyType

This enumeration shows what kind of object can be passed as input parameters :

Attribute	Description	Data Type
RestCase	The object type is RESTCase is contains the case details	PropertyType
DocumentMetadata	The object type is RESTDocument and contains document metadata details	PropertyType
SedData	The object type is String and contains the SED data payload in JSON format	PropertyType
BatchMetadata	The object type is BatchMetadata and contains metadata of the batch of document parts	PropertyType

6.7 BatchMetadata

BatchMetadata contains information about the batch of document parts: the number of the first subdocument in the batch and the batch size:

Attribute	Description	Data Type
firstNo	The ordinal number of the first subdocument in the batch	Integer
size	The number of subdocuments in this batch	Integer

7 Fault contract

We define the fault contract as the possible exceptions that might occur in the business messaging component.

Diagram

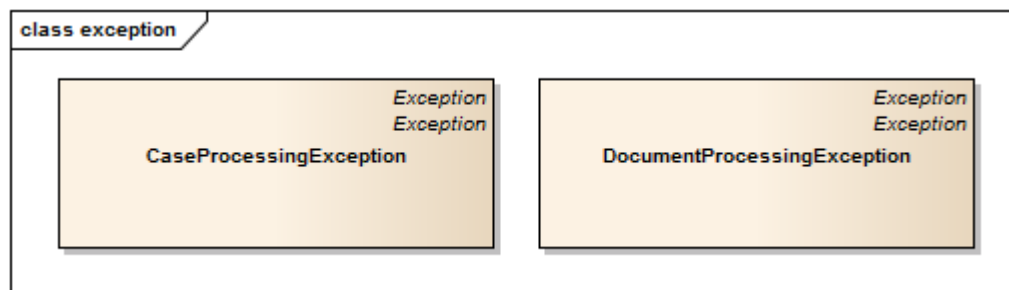


Figure 13. Fault Contract

CaseProcessingException

Exception that occurs during the execution of the case listener callback, this exception must be thrown by the client implementation in case of any error might happen during the case event processing.

DocumentProcessingException

Exception that occurs during the execution of the document listener callback, this exception must be thrown by the client implementation in case of any error might happen during the document event processing.

Note:

In cases when updatedPayloadMap are uncompliant, no faults are expected, this will be ignored and logged

8 Illustrative sample of a basic implementation

Depending on their needs, the national application implements the methods introduced in this document as they need. It is not necessary to address all possible events. The national application can choose to "listen" only for some combinations of events and input parameters and to ignore others.

A very basic implementation could use this pattern:

```
@Override
public boolean hasDocumentListener(DocumentEventType documentType, String buc, String version,
String document) {

    if (documentType.equals(DocumentEventType.INITIALISE_DOCUMENT)) {
if (buc.equals("P_BUC_01") && document.equals("P6000"))
return true;
if (buc.equals("P_BUC_01") && document.equals("P5000"))
return true;
return false;
}
// ...

return false;
}

@Override
public Map<String, Object> fireDocumentEvent(DocumentEventType documentEventType, String buc,
String version, String document, Map<PropertyType, Map<String, Object>> payload) throws
DocumentProcessingException {

    if (documentEventType.equals(DocumentEventType.INITIALISE_DOCUMENT) &&
buc.equals("P_BUC_01") && document.equals("P5000")) {

        Map<String, Object> data = payload.get(PropertyType.SED_DATA);
        .../* code to call the national application to store the SED data in it */
        String personId = .... /* code to read the person identification field/fields from received
sed data */
        Object naInformation = .... /* code to call the national application and to retrieve useful
data from it, related to the person with personId determined before */

        Map<String, Object> updatedPayload =
        .... /* code to fill in the SED with the data retrieved from the national application */

        return updatedPayload;
    }
    // ....
    return payload;
}
```

9 NIE configuration

The admin module of the portal allows also to configure the NIE events.

From the NIE Settings it can be added a new subscription to case event, document event or it can be deleted a subscription. Single or multiple cases or documents can be part of the same subscription.

If there is no subscription defined, no event will be fired/sent to the RINIE.

An example of the configured events is illustrated in the Figure 14 below.

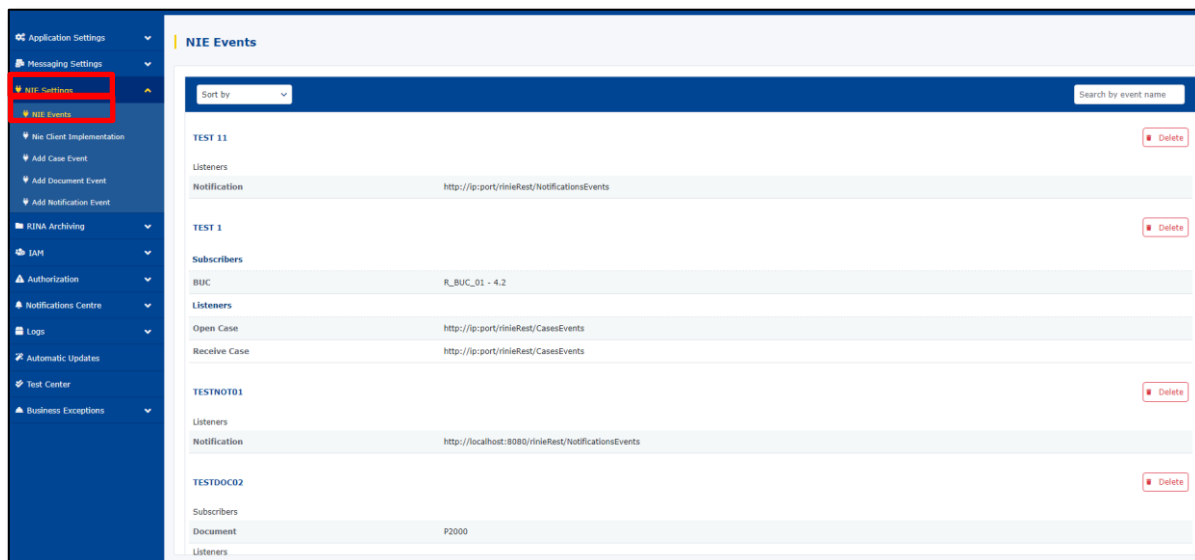


Figure 14. Illustrative sample of configured events

10 Reference implementation of NIE (RINIE)

The reference implementation of the National Information Exchange is done as the RESTful services.

RINIE purpose is to demonstrate how 3rd party applications may interact with RINA NIE service by using its API.

10.1 RINIE behaviour

10.1.1 RINIE configuration

By default RINIE is preconfigured to save all the incoming events to the root of the local tomcat home directory in folder called 'eessi' (`${catalina.home}/eessi`).

Of course, this configuration can be altered accordingly if needed.

The root path is configured into the RINIE project folder (`src/main/webapp/WEB-INF/spring/`) in a configuration file called `rinieRest.properties`

In the `rinieRest.properties` file, the `rinie.FilesStorageRoot` property is set to as follows:

(i.e. `"rinie.FilesStorageRoot=${catalina.home}/eessi"`).

When setting the NIE events in NIE Settings tab, you should noticed that the following URIs have been used as references:

<http://ip:port/rinieRest/CasesEvents> (for Case Events)

<http://ip:port/rinieRest/DocumentsEvents> (for Document Events)

<http://ip:port/rinieRest/NotificationsEvents> (for Notification Events)

These references URI can be configured for example for the documents' event as follows:

<http://127.0.0.1:8081/rinieRest/DocumentsEvents>

On specifying this URI for the Documents'events, RINIE will store the events under the current tomcat in a subfolder 'nie' of the root folder 'eessi'.

And you should be able to navigate to the 'eessi/nie' folder of the local tomcat to retrieve the events. For example:

C:\EESSI\REST\Tomcat\eessi\nie

Inside the 'nie' directory there are four RINIE specific sub-directories:

- **case_events**– used to store cases events;

- **document_events** – used to store documents events;
- **notification_events** – used to store the notifications;
- **templates** – used to store the documents templates (the JSON files that contain the SED with prefilled fields that need to be returned).

10.1.2 Cases processing

The case processing in RINIE is made following steps:

- Receive case event;
- The case content is extracted and stored as a new file in **case_events** sub-folder.

The name convention of new file mentioned above is:

EVENT_NAME + "_" + BUC_NAME + "_" + BUC_VERSION + "_" + CASE_ID ".json"

10.1.3 Documents processing

10.1.3.1 Intro

The scope of documents processing within RINIE is to demonstrate that the bidirectional events are working correctly.

As mentioned in the above section 4.1.2 - *Exchange document information with national application*, RINIE can modify the documents before returning them.

To simulate the NA message processing, RINIE implements its functionality by merging a predefined template with the SED, from a received message.

10.1.3.2 Use case scenario

This scenario is applicable for one of the following events:

- INITIALISE_DOCUMENT,
- NEW_DOCUMENT,
- UPDATE_DOCUMENT,
- SENT_DOCUMENT,
- DELETE_DOCUMENT,
- CANCEL_DOCUMENT,
- RECEIVE_DOCUMENT,
- INITIALISE_SUBDOCUMENT,
- NEW_SUBDOCUMENT,
- UPDATE_SUBDOCUMENT,
- DELETE_SUBDOCUMENT,
- NEW_DOCUMENT_BATCH,
- RECEIVE_DOCUMENT_BATCH.

Once the event is received in RINIE, the following processing steps are performed:

- A template file matching event + BUC + version + document type + PIN is searched. If found this is used to return content as reply to triggering this NIE event. I.E.: UPDATE_DOCUMENT_P_BUC_01_4.2_P5000_1234.json

- If not found, template file matching event + BUC + version + document type with NO PIN is searched. Thus, it is possible to initialize empty documents with no previous data, using Initialise_document event, i.e.: INITIALISE_DOCUMENT_P_BUC_01_4.2_P2000.json
- Always first will search with template including PIN number in the name. If not found, it will try to search the one without PIN in the name.
- If for the received event a corresponding template is found then, the SED content (within the received event) will be merged with the content from the aforementioned template;
Note: The template content has priority over the SED of event
- The SED of the document events is stored into document_events sub-directory;
- The RINIE document processing service will return the modified content which will be displayed on the portal.

Naming conventions

Document/SED files naming convention:

- EVENT_NAME + "_" + BUC_NAME + "_" + BUC_VERSION + "_" + DOCUMENT_NAME + "_" + CASE_ID ".json"

Template files naming conventions:

- EVENT_NAME + "_" + BUC_NAME + "_" + BUC_VERSION + "_" + DOCUMENT_NAME + "_" + PIN + ".json"
- EVENT_NAME + "_" + BUC_NAME + "_" + BUC_VERSION + "_" + DOCUMENT_NAME + ".json"

Illustrative examples of JSON templates' file names:

- INITIALISE_DOCUMENT_P_BUC_01_0.17_P5000_PIN156.json
- INITIALISE_DOCUMENT_P_BUC_01_0.17_P6000_PIN156.json
- INITIALISE_DOCUMENT_P_BUC_01_4.2_P5000.json
- NEW_DOCUMENT_P_BUC_01_4.2_P5000.json

10.1.4 Notification processing.

The notification processing in RINIE are as follows:

- Set the particular notification which needs to produce a NIE event in the Notification Centre;
- Receive a notification event.
- The notification content is extracted and stored as a new file in **notification_events** sub-folder.

The name convention of new file mentioned above is:

EVENT_NAME + "_" + CASE_ID + "_" + DATE_&_TIME_OF_NOTIFICATION.json

10.1.4.1 RINIE document templates illustrative samples

INITIALISE_DOCUMENT_P_BUC_01_0.17_P5000_PIN156.json

```
{
  "InsurancePeriods":{
    "informationInsurancePeriods":{
      "periods":{
        "total":{
          "months":"",
          "days":{
            "unit":"",
            "nb":""
          },
          "quarters":"",
          "years":"",
          "weeks":""
        },
        "occupation":"",
        "scheme":"",
        "infoCalculation":"",
        "contingency":"",
        "type":"",
        "period":{
          "startDate":"",
          "endDate":""
        },
        "country":""
      },
      "noInsurancePeriod":"false",
      "transmissionContext":""
    },
    "insuredPerson":{
      "addInfo":{
        "previousForenames":"",
        "nationality":"",
        "address":{
          "region":"Regiunea 1",
          "postalCode":"1000",
          "street":"ArtsLoi",
          "town":"Brussels",
          "country":""
        },
        "identityProtected":"",
        "previousFamilyNames":"",
        "previousForename":""
      },
      "personId":{
        "MinIDSetWithoutPIN-P":{
          "forenameofMother":"",
          "placeOfBirth":{
            "countryBirth":"",
            "townBirth":"",
            "regionBirth":""
          },
          "fatherFamilyNameAtBirth":"",
          "forenameofFather":"",
          "motherFamilyNameAtBirth":""
        },
        "familyName":"Ionescu",
        "gender":"M",
        "forenamesAtBirth":null,
        "birthDate":"2016-01-22T00:00:00Z",
        "familyNameAtBirth":null,
        "forenames":"Martin",
        "pinSendingInstitution":""
      }
    }
  }
}
```

```
        "MinIDSetWithPIN-P":{
            "pinReceivingInstitution":""
        }
    },
    "infop5000":"false",
    "pensionReceipt":"","
    "totalLengthPeriod":{
        "total":{
            "months":"","
            "days":{
                "unit":"","
                "nb":""
            },
            "quarters":"","
            "contingency":"","
            "years":"","
            "type":"","
            "weeks":""
        },
        "infoCalculation":{
            "months":"","
            "days":{
                "unit":"","
                "nb":""
            },
            "quarters":"","
            "infoCalcul":"","
            "years":"","
            "type":"","
            "weeks":""
        }
    },
    "flowID":{
        "relatedFlowID":""
    },
    "claimant":{
        "addInfo":{
            "nationality":"","
            "address":{
                "region":"","
                "postalCode":"","
                "street":"","
                "town":"","
                "country":""
            },
            "identityProtected":"false",
            "previousFamilyNames":"","
            "previousForename":""
        },
        "personId":{
            "MinIDSetWithoutPIN-P":{
                "placeOfBirth":{
                    "countryBirth":""
                }
            },
            "familyName":"","
            "gender":"","
            "forenamesAtBirth":"","
            "birthDate":"","
            "familyNameAtBirth":"","
            "forenames":"","
            "pinSendingInstitution":"","
            "MinIDSetWithPIN-P":{
                "pinReceivingInstitution":""
            }
        },
        "ifPin":{
            "institution":{
                "institutionCode":"","
                "pin":"","
                "institutionName":"","
                "country":""
            }
        }
    },
    "institutionConcerned":{
```

```
        "institutionIdentification":{
            "institutionCode":"","
            "pin":"","
            "institutionName":"","
            "countryCode":"","
            "caseNo":""
        }
    }
}
```

INITIALISE_DOCUMENT_P_BUC_01_0.17_P6000_PIN156.json

```
{
    "PensionDecision":{
        "insured":{
            "addInfo":{
                "previousForenames":"","
                "nationality":"","
                "address":{
                    "region":"South",
                    "postalCode":"1000",
                    "street":"Loi",
                    "town":"Gent",
                    "country":""
                },
                "identityProtected":"","
                "previousFamilyNames":""
            },
            "person":{
                "MinIDSetWithoutPIN-P":{
                    "forenameofMother":"","
                    "placeOfBirth":{
                        "countryBirth":"","
                        "townBirth":"","
                        "regionBirth":""
                    },
                    "fatherFamilyNameAtBirth":"","
                    "forenameofFather":"","
                    "motherFamilyNameAtBirth":""
                },
                "familyName":"Brad",
                "gender":"F",
                "forenamesAtBirth":null,
                "birthDate":"2000-01-22T00:00:00Z",
                "familyNameAtBirth":null,
                "forenames":"Joline",
                "pinSendingInstitution":"","
                "MinIDSetWithPIN-P":{
                    "pinReceivingInstitution":""
                }
            }
        },
        "additionalInformation":{
            "infoRightReview":{
                "timeLimit":"","
                "placeReview":""
            },
            "institutionDecision":{
                "institution":{
                    "institutionCode":"","
                    "pin":"","
                    "institutionName":"","
                    "caseNumber":"","
                    "country":""
                }
            },
            "pensionWithdrawal":"","
            "decisionAccordingReg":"false",
            "benefitsTypeA":"false",
            "refNb":"","
            "date":"","
            "otherInfo":"","
            "suspensionPension":""
        },
        "reduction":{
```

```
"reductionPension":{
  "otherBenefIncome":"","
  "reductionDifferentKind":"","
  "reductionSameKind":"","
  "overlappingPeriod":{
    "start":"","
    "end":""
  },
  "limitationRuleOverlappingArticle":""
},
"flowID":{
  "relatedFlowID":""
},
"decision":{
  "awardBenefitArticle":"","
  "rejectionReason":"","
  "pensionType":"","
  "startDayPension":"","
  "paymentDetails":{
    "paymentFrequency":"","
    "pensionAmount":{
      "amountAfterTaxesReduction":"","
      "supplementaryPensionAmount":"","
      "amountBasicPension":"","
      "amountBeforeTaxesReduction":""
    },
    "otherFrequency":"","
    "payableTo":"","
    "payableFrom":"","
    "currency":""
  },
  "decisionType":"","
  "definedTypePension":"","
  "partialSuspension":{
    "pensionSuspendedYes":{
      "amount":"","
      "reason":"","
      "currency":""
    },
    "pensionSuspended":""
  },
  "otherInsuranceBased":"","
  "voluntaryContributionsBased":"","
  "otherReason":"","
  "creditedPeriod":"","
  "supplement":"","
  "actualDatePensionPay":"","
  "additionalInformation":""
},
"claimant":{
  "addInfo":{
    "previousForenames":"","
    "nationality":"","
    "address":{
      "region":"","
      "postalCode":"","
      "street":"","
      "town":"","
      "country":""
    },
    "identityProtected":"false",
    "previousFamilyNames":""
  },
  "person":{
    "MinIDSetWithoutPIN-P":{
      "placeOfBirth":{
        "countryBirth":""
      }
    },
    "familyName":"","
    "gender":"","
    "forenamesAtBirth":"","
    "birthDate":"","
    "familyNameAtBirth":"","
    "forenames":"","
    "pinSendingInstitution":""
  }
}
```

```
        "MinIDSetWithPIN-P":{
            "pinReceivingInstitution":""
        }
    },
    "institutionConcerned":{
        "institutionIdentification":{
            "institutionCode":"","
            "pin":"","
            "institutionName":"","
            "countryCode":"","
            "caseNo":""
        }
    }
}
```

10.2 Service contract

On the server are published two RESTful services as illustrate in the Figure 15 below **Error! Reference source not found.:**

- POST /CasesEvents;
- POST /DocumentsEvents.

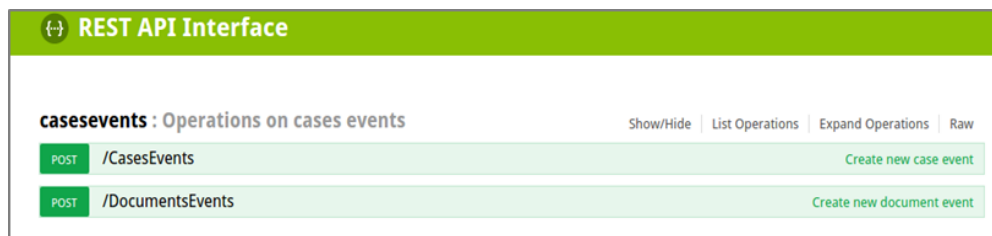


Figure 15. Rest API interface

10.2.1 CasesEvents

Post cases events on RINIE. The cases are saved on configured folder.

Request URI

- POST /CasesEvents.

Post Parameters

RESTCaseEvent

Request Sample

For details, please see: [10.4.1 CasesEvents](#)

Response (on Success)

HTTP Status Code	Description	Model
200	Returns the changed payload of the case	Map<String, Object>

Response (on Error)

HTTP Status Code	Reason	Description
500	Cannot return the payload of the case	RESTException

10.2.2 DocumentsEvents

Post documents events on RINIE. The documents payload are saved on configured folder.

Request URI

- POST /DocumentssEvents.

Post Parameters

RESTDdocumentEvent.

Request Sample

For details, please see: [10.4.2 DocumentsEvents](#)

Response (on Success)

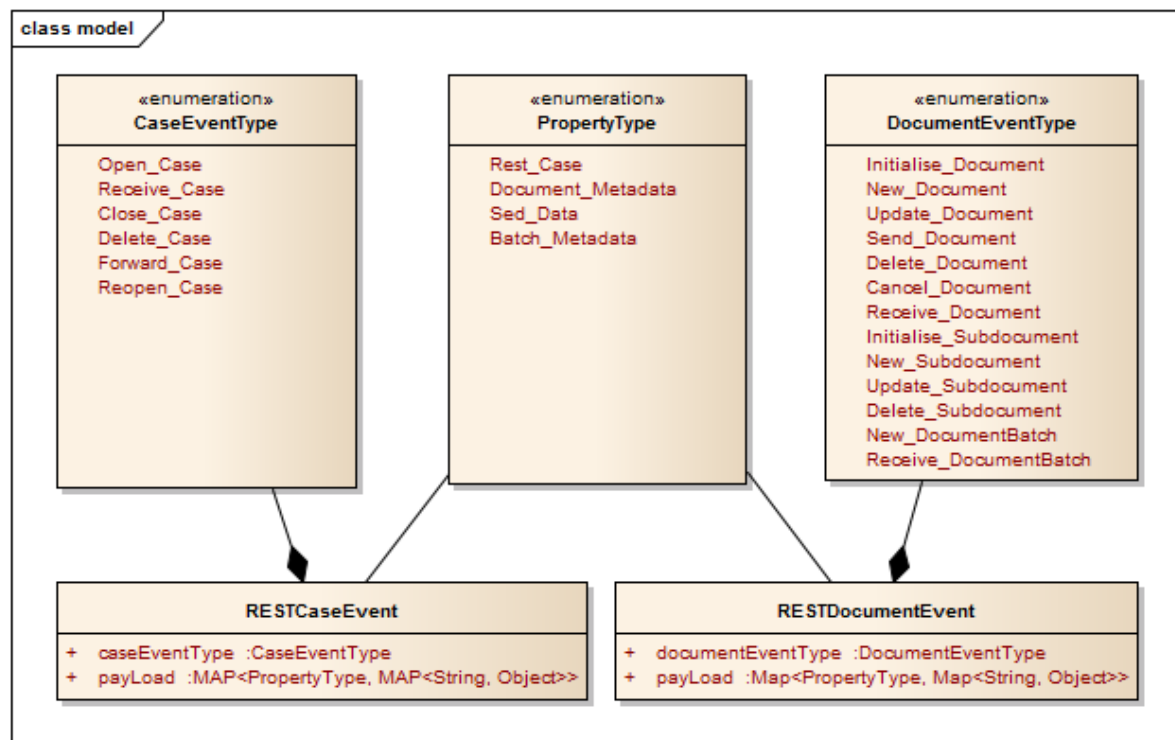
HTTP Status Code	Description	Model
200	Returns the changed SED payload of the document	Map<String, Object>

Response (on Error)

HTTP Status Code	Description	Model
500	Cannot return the SED payload of the document	RESTException

10.3 Data contract

The data transfer model is modelled as a UML class diagram as can be seen in the following figure and data classes described in the below sections.



10.3.1 CaseEventType

Case event type details

Attribute	Description	Data Type
OPEN_CASE	Open case event	CaseEventType
REMOVE_CASE	Remove case event	CaseEventType
CLOSE_CASE	Close case event	CaseEventType
DELETE_CASE	Delete case event	CaseEventType
FORWARD_CASE	Forward case event	CaseEventType
REOPEN_CASE	Reopen case event	CaseEventType

10.3.2 DocumentEventType

Document event type details

Attribute	Description	Data Type
INITIALISE_DOCUMENT	Initialise document event	DocumentEventType
NEW_DOCUMENT	New document event	DocumentEventType
UPDATE_DOCUMENT	Update document event	DocumentEventType
SENT_DOCUMENT	Send document event	DocumentEventType
DELETE_DOCUMENT	Delete document event	DocumentEventType
CANCEL_DOCUMENT	Cancel document event	DocumentEventType
RECEIVE_DOCUMENT	Receive document event	DocumentEventType
INITIALISE_SUBDOCUMENT	Initialise subdocument event	DocumentEventType
NEW_SUBDOCUMENT	New subdocument event	DocumentEventType
UPDATE_SUBDOCUMENT	Update subdocument event	DocumentEventType
DELETE_SUBDOCUMENT	Delete subdocument event	DocumentEventType
NEW_DOCUMENT_BATCH	New batch document event	DocumentEventType
RECEIVE_DOCUMENT_BATCH	Receive batch document event	DocumentEventType

10.3.3 PropertyType

Type of payload attached details

Attribute	Description	Data Type
REST_CASE	The payload attached is a case data.	PropertyType
DOCUMENT_METADATA	The payload attached is a document metadata	PropertyType
SED_DATA	The payload attached is a SED data	PropertyType
BATCH_METADATA	The payload attached is a batch metadata	PropertyType

10.3.4 RESTCaseEvent

Rest case event details

Attribute	Description	Data Type
caseEventType	The type of case event.	CaseEventType
payload	A map with one entry which is case data.	Map<PropertyType, Map<String, Object>>
userId	User performing the action that triggered the event. In case of system events user may be null or System	String

10.3.5 RESTDocumentEvent

Rest document event details

Attribute	Description	Data Type
documentEventType	The type of document event.	DocumentEventType
payload	Payload map has two entries: the first entry is the DocumentMetadata that contains all the metadata for the document (what case is related to, the document type, who created, etc.) and the second entry is the actual SED payload. There is a 3rd entry of batch metadata type for the batch document events.	Map<PropertyType, Map<String, Object>>
userId	User performing the action that triggered the event. In case of system events user may be null or System	String

10.3.6 RESTNotificationEvent

Attribute	Description	Data Type
payload	Payload map has one entry: this is the full notification.	Map<PropertyType, Map<String, Object>>

10.4 RINIE events samples

10.4.1 CasesEvents

Input parameter:

```
RESTCaseEvent{
  caseEventType=OpenCase,
  payLoad= {
    REST_CASE= {
      id=75,
      processDefinitionName=PO_P_BUC_01_v1.0,
      internationalId=645e2d0d4bcf450c927fd29f907f144c,
      sensitive=false,
      startDate=2018-02-07T12:42:04          Z,
      lastUpdate=2018-02-07T12:42:04        Z,
      searchMetadata= {
        caseId=75,
        flowType=P_BUC_01,
        status=open
      },
      participants= [
        ],
      status=open,
      subject= {
        },
      },
      whoami= {
        activeSince=2010-01-01T08:00:00.000    Z,
        registryNumber=,
        acronym=LOOPBACK,
        countryCode=BE,
        name=Public Sector,
        location=50.842443,
        4.367391,
        id=LOOPBACK
      },
      creator= {
        id=supervisor,
        name=supervisor,
        organisation= {
          activeSince=2010-01-01T08:00:00.000    Z,
          registryNumber=,
          acronym=LOOPBACK,
          countryCode=BE,
          name=Public Sector,
          location=50.842443,
          4.367391,
          id=LOOPBACK
        }
      },
      caseAssignment= {
        id=75,
        properties= {
          importance=1,
          criticality=1
        },
        actors= [
          {
            id=130,
            name=Auditor,
            userGroups= [
              {
                id=3,
                name=clerka,
```

```
        type=User
      },
      {
        id=1,
        name=supervisor,
        type=User
      }
    ]
  },
  {
    id=133,
    name=Authorized,
    userGroups= [
      {
        id=3,
        name=clerka,
        type=User
      },
      {
        id=1,
        name=supervisor,
        type=User
      }
    ]
  },
  {
    id=132,
    name=Medical,
    userGroups= [
      {
        id=3,
        name=clerka,
        type=User
      },
      {
        id=1,
        name=supervisor,
        type=User
      }
    ]
  },
  {
    id=129,
    name=Nonauthorized,
    userGroups= [
      {
        id=3,
        name=clerka,
        type=User
      },
      {
        id=1,
        name=supervisor,
        type=User
      }
    ]
  },
  {
    id=127,
    name=Supervisor,
    userGroups= [
      {
        id=3,
        name=clerka,
        type=User
      },
      {
        id=1,
        name=supervisor,
        type=User
      }
    ]
  },
  {
    id=134,
    name=System,
    userGroups= [
```

```
        {
            id=1,
            name=system,
            type=User
        }
    ],
},
{
    id=131,
    name=Viewer,
    userGroups=
    [
        {
            id=3,
            name=clerka,
            type=User
        },
        {
            id=1,
            name=supervisor,
            type=User
        }
    ]
},
{
    id=128,
    name=Vip,
    userGroups=
    [
        {
            id=3,
            name=clerka,
            type=User
        },
        {
            id=1,
            name=supervisor,
            type=User
        }
    ]
}
],
},
businessId=75
},
},
userId= supervisor
}
```

10.4.2 Documents Events

Input parameter:

```
{
  "documentEventType": "INITIALISE_DOCUMENT",
  "payload": {
    "DOCUMENT METADATA": {
      "internalId": "7_10c8179145cd47fb95db6f2524ae73d0",
      "ParentType": "P2000",
      "status": "sent",
      "direction": "OUT",
      "conversations": [
        {
          "id": "59e62ce14f5c4f4fb922029923089621",
          "userMessages": [
            {
              "id": "2aa30e14fa8e4b75af3fe96df695a8b2",
              "sender": {
                "id": "BENA07",
                "acronym": "BENA07",
                "location": "48.148744,14.892614",
                "name": "BENA07 institution",
                "activeSince": "2010-01-01T08:00:00.000Z",
                "countryCode": "BE",
                "registryNumber": ""
              },
              "isSent": true,
            }
          ]
        }
      ]
    }
  }
}
```

```
        "receiver": {
          "id": "BENA05",
          "acronym": "BENA05",
          "location": "46.548114,14.604011",
          "name": "BENA05 institution",
          "activeSince": "2010-01-01T08:00:00.000Z",
          "countryCode": "BE",
          "accessPoint": null,
          "registryNumber": ""
        }
      ],
      "participants": [
        {
          "organisation": {
            "id": "BENA07",
            "acronym": "BENA07",
            "location": "48.148744,14.892614",
            "name": "BENA07 institution",
            "activeSince": "2010-01-01T08:00:00.000Z",
            "countryCode": "BE",
            "registryNumber": ""
          },
          "role": "Sender"
        },
        {
          "organisation": {
            "id": "BENA05",
            "acronym": "BENA05",
            "location": "46.548114,14.604011",
            "name": "BENA05 institution",
            "activeSince": "2010-01-01T08:00:00.000Z",
            "countryCode": "BE",
            "accessPoint": null,
            "registryNumber": ""
          },
          "role": "Receiver"
        }
      ]
    },
    {
      "id": "0d9108a5fcd14b95a51b640282c5ff93",
      "participants": [
        {
          "role": "Sender",
          "organisation": {
            "id": "BENA07",
            "acronym": "BENA07",
            "location": "48.148744,14.892614",
            "name": "BENA07 institution",
            "countryCode": "BE",
            "activeSince": "2010-01-01T08:00:00.000Z",
            "registryNumber": ""
          }
        },
        {
          "role": "Receiver",
          "organisation": {
            "id": "BENA05",
            "acronym": "BENA05",
            "location": "46.548114,14.604011",
            "name": "BENA05 institution",
            "countryCode": "BE",
            "activeSince": "2010-01-01T08:00:00.000Z",
            "accessPoint": null,
            "registryNumber": ""
          }
        }
      ]
    }
  ],
  "type": "P2000",
  "creator": {
    "id": "2",
    "type": "User",
    "organisation": {
      "id": "BENA07",
```

```
        "acronym": "BENA07",
        "location": "48.148744,14.892614",
        "name": "BENA07 institution",
        "activeSince": "2010-01-01T08:00:00.000Z",
        "countryCode": "BE",
        "registryNumber": ""
    },
    "name": "supervisor"
},
"id": "10c8179145cd47fb95db6f2524ae73d0",
"versions": [
    {
        "id": "1",
        "date": "2015-12-02T10:04:23.492Z",
        "user": {
            "id": "2",
            "type": "User",
            "organisation": {
                "id": "BENA07",
                "acronym": "BENA07",
                "location": "48.148744,14.892614",
                "name": "BENA07 institution",
                "activeSince": "2010-01-01T08:00:00.000Z",
                "countryCode": "BE",
                "registryNumber": ""
            },
            "name": "supervisor"
        }
    }
],
"typeVersion": "3.2",
"caseId": "7",
"starter": true,
"attachments": null,
"displayName": "Old age pension claim",
"mimeType": "json",
"isFirstDocument": true
},
"SED_DATA": {
    "internalId": "7_10c8179145cd47fb95db6f2524ae73d0_1",
    "content": {
        "OldAgePensionClaim": {
            "insuredBenefitDetails": {
                "benefitDetails": {
                    "otherBenefit": "",
                    "benefitsPaymentEndDate": "",
                    "status": "",
                    "entitlementBenefitsStartDate": "",
                    "baseOfReceivedPension": {
                        "amount": "",
                        "pensionReceivedBasedOn": "",
                        "amountWork": ""
                    },
                    "entitlementBenefitsEndDate": "",
                    "benefits": "",
                    "amountInfo": {
                        "amount": "",
                        "paymentFrequency": "",
                        "otherFrequency": "",
                        "effective": "",
                        "currency": ""
                    },
                    "benefitsPaymentStartDate": "",
                    "institution": {
                        "institutionCode": "",
                        "pin": "",
                        "institutionName": "",
                        "caseNumber": "",
                        "country": ""
                    }
                }
            },
            "sedVer": "0",
            "children": {
                "childId": {
                    "addInfo": {
                        "relationInsured": "",

```

```
        "nationality": "",
        "deathDate": "",
        "otherChild": ""
    },
    "person": {
        "MinIDSetWithoutPIN-P": {
            "forenameofMother": "",
            "placeOfBirth": {
                "countryBirth": "",
                "townBirth": "",
                "regionBirth": ""
            },
            "fatherFamilyNameAtBirth": "",
            "forenameofFather": "",
            "motherFamilyNameAtBirth": ""
        },
        "familyName": "",
        "gender": "",
        "forenamesAtBirth": "",
        "birthDate": "",
        "familyNameAtBirth": "",
        "forenames": "",
        "pinSendingInstitution": ""
    }
},
"sedGVer": "0",
"infoGuardian": {
    "address": {
        "region": "",
        "postalCode": "",
        "contactInfo": {
            "phoneNo": "",
            "email": ""
        },
        "street": "",
        "town": "",
        "country": ""
    },
    "name": "",
    "grounds": ""
},
"misc": {
    "deductionsGrounds": "",
    "paymentBenefitsStartDateDesiredClaimant": "",
    "dateOfClaim": "",
    "decisionRecipient": "",
    "notClaimedYet": "",
    "deferment": {
        "defermentTime": "",
        "defermentInstitutionName": "",
        "defermentCountry": "",
        "defermentInstitutionCode": ""
    },
    "remarks": "",
    "otherAttach": "",
    "attachments": "",
    "otherRequest": ""
},
"institutionConcerned": {
    "institutionIdentification": {
        "institutionCode": "",
        "pin": "",
        "institutionName": "",
        "countryCode": "",
        "caseNo": ""
    }
},
"insuredPerson": {
    "addInfo": {
        "previousForenames": "",
        "nationality": "",
        "contactInfo": {
            "phoneNo": "",
            "email": ""
        },
        "address": {
```

```
        "region": "",
        "postalCode": "",
        "street": "",
        "town": "",
        "country": ""
    },
    "familyStatus": {
        "familyStatusDate": "",
        "familyStatusList": ""
    },
    "previousFamilyNames": ""
},
"person": {
    "MinIDSetWithoutPIN-P": {
        "forenameofMother": "",
        "placeOfBirth": {
            "countryBirth": "",
            "townBirth": "",
            "regionBirth": ""
        },
        "fatherFamilyNameAtBirth": "",
        "forenameofFather": "",
        "motherFamilyNameAtBirth": ""
    },
    "familyName": "7",
    "gender": "F",
    "forenamesAtBirth": "",
    "birthDate": "2015-12-01T23:00:00.000Z",
    "familyNameAtBirth": "",
    "forenames": "7",
    "pinSendingInstitution": "7"
},
},
"spouse": {
    "addInfo": {
        "previousForenames": "",
        "nationality": "",
        "previousFamilyNames": ""
    },
    "person": {
        "MinIDSetWithoutPIN-P": {
            "forenameofMother": "",
            "placeOfBirth": {
                "countryBirth": "",
                "townBirth": "",
                "regionBirth": ""
            },
            "fatherFamilyNameAtBirth": "",
            "forenameofFather": "",
            "motherFamilyNameAtBirth": ""
        },
        "familyName": "",
        "gender": "",
        "forenamesAtBirth": "",
        "birthDate": "",
        "familyNameAtBirth": "",
        "forenames": "",
        "pinSendingInstitution": ""
    },
    "spouseType": ""
},
"insuredEmploymentDetails": {
    "emplSelfEmplDetails": {
        "occupation": "",
        "startDateIntendedEmpl": "",
        "employment": "",
        "hoursPerWeek": "",
        "incomeAmount": {
            "amount": "",
            "paymentFrequency": "",
            "otherFrequency": "",
            "effective": "",
            "currency": ""
        },
        "endDateEmpl": "",
        "intendedRetirementDate": ""
    }
}
```

```
    },
    "sedPackage": "Pension",
    "flowID": {
        "relatedFlowID": "7"
    },
    "infoPayment": {
        "bankInfo": {
            "bankName": "",
            "bic": "",
            "bankAddress": {
                "region": "",
                "postalCode": "",
                "street": "",
                "town": "",
                "country": ""
            },
            "iban": ""
        },
        "payment": ""
    }
},
{id": "10c8179145cd47fb95db6f2524ae73d0",
"caseId": "7",
"versionId": "1"
}
},
userId= supervisor
}
```

10.4.3 Notifications Events

```
RESTNotificationEvent{

    payload= {
        {
            "severity": "warning",
            "creator": {
                "name": "System",
                "organisation": {
                    "activeSince": "2010-01-01T08:00:00.000Z",
                    "registryNumber": "",
                    "acronym": "LOOPBACK",
                    "countryCode": "BE",
                    "name": "Public Sector",
                    "location": "50.842443,4.367391",
                    "id": "LOOPBACK"
                },
                "id": "0",
                "type": "User"
            },
            "subject": null,
            "dueDate": "2018-03-18T10:16:46.227Z",
            "document": {
                "receiver": null,
                "name": "Old age pension claim",
                "type": "P2000"
            },
            "isRead": "false",
            "creationDate": "2018-03-15T10:16:46.227Z",
            "type": "case_arrived",
            "responsibleParties": [
                { "name": "clerkb",
                  "organisation": {
                      "activeSince": "2010-01-01T08:00:00.000Z",
                      "registryNumber": "", "acronym": "LOOPBACKB",
                      "countryCode": "BE",
                      "name": "PublicSector",
                      "location": "50.842443,4.367391",
                      "id": "LOOPBACKB"
                  },
                  "id": "5",
                }
            ]
        }
    }
}
```

```
        "type": "User"
      },
      { "name": "supervisorb",
        "organisation": {
          "activeSince": "2010-01-01T08:00:00.000Z",
          "registryNumber": "",
          "acronym": "LOOPBACKB",
          "countryCode": "BE",
          "name": "Public Sector",
          "location": "50.842443,4.367391",
          "id": "LOOPBACKB"
        },
        "id": "4", "type": "User"
      }
    ],
    "sourceType": "business",
    "lastUpdate": "2018-03-15T10:16:46.227Z",
    "caseId": "49",
    "id": "fd533826a9164fda9fda3d69fcdeec49",
    "status": "active"
  }
}
```