



e-GOV 3.0 – Documentation à destination des déclarants

Envoi par SFTP

2025 – 1^{re} version

TABLE DES MATIERES

| | |
|---|-----------|
| 1 – INTRODUCTION : PROJET E-GOV 3.0 | 3 |
| 2 – PRÉREQUIS AUX ÉCHANGES AVEC LA SÉCURITÉ SOCIALE | 4 |
| 2.1 – Enregistrement de l'accès | 4 |
| 2.2 – Protocole de communication | 5 |
| 2.2.1 – Protocole d'envoi | 5 |
| 2.2.2 – Protocole de réception..... | 6 |
| 3 – DESCRIPTION DES ÉCHANGES DÉCLARANT – ONSS..... | 7 |
| 3.1 – Description générale des échanges | 7 |
| 3.2 – Diagramme d'activités..... | 8 |
| 3.3 – Diagrammes de séquence..... | 9 |
| 3.3.1 – Scénario 1 : le groupe d'envoi est valide et tous les CloudEvents sont également valides..... | 9 |
| 3.3.2 – Scénario 2 : le groupe d'envoi est invalide | 9 |
| 3.3.3 – Scénario 3 : le groupe d'envoi est valide et un ou plusieurs CloudEvents sont invalides | 10 |
| 4 – DÉPÔT EFFECTUÉ PAR LE DÉCLARANT..... | 11 |
| 4.1 – Structure du nom des fichiers | 11 |
| 4.2 – Description des fichiers | 12 |
| 4.2.1 – Fichier de déclaration (FI) | 12 |
| 4.2.1.1 – Description technique d'un CloudEvent | 12 |
| 4.2.1.2 – Schéma JSON du fichier FI..... | 15 |
| 4.2.1.3 – Exemple de fichier FI..... | 16 |
| 4.2.2 – Fichier de signature (FS)..... | 16 |
| 4.2.3 – Fichier GO | 17 |

| | |
|--|-----------|
| 5 – RÉPONSES RETOURNÉES PAR L'ONSS | 18 |
| 5.1 – Structure du nom des fichiers | 18 |
| 5.2 – Description des fichiers | 19 |
| 5.2.1 – Fichier de réponses (FO) | 19 |
| 5.2.1.1 – Description des CloudEvents de réponse | 19 |
| 5.2.1.1.1 - ackEvent | 19 |
| 5.2.1.1.1.1 – Description technique | 19 |
| 5.2.1.1.1.2 – Schéma JSON | 23 |
| 5.2.1.1.1.3 – Exemple d'ackEvent positif..... | 24 |
| 5.2.1.1.1.4 – Exemple d'ackEvent négatif..... | 24 |
| 5.2.1.1.2 - problemEvent | 25 |
| 5.2.1.1.2.1 – Description technique | 25 |
| 5.2.1.1.2.2 – Schéma JSON | 29 |
| 5.2.1.1.2.3 – Exemple..... | 30 |
| 5.2.1.1.3 – businessFeedbackEvent..... | 30 |
| 5.2.1.2 – Schéma JSON du fichier FO | 31 |
| 5.2.1.3 – Exemple de fichier FO | 32 |
| 5.2.2 – Fichier de signature (FS)..... | 33 |
| 5.2.3 – Fichier GO | 33 |

1 – INTRODUCTION : PROJET E-GOV 3.0

L'objectif du projet e-Gov 3.0 est d'évoluer vers une couche centrale de données pour un traitement optimal de celles-ci. Cette nouvelle source centrale de données doit :

- fournir des services numériques basés sur les données nécessaires afin de régler les droits de Sécurité Sociale d'un citoyen ;
- être alimentée par les employeurs, leurs prestataires de services et les institutions à tous les niveaux ;
- être alimentée selon le principe du *only once*, ce qui signifie que les mêmes données ne doivent être saisies qu'une seule fois ;
- rendre l'information disponible le plus rapidement, afin de permettre des prises de décision immédiates et précises.

L'exemple de l'actuelle déclaration trimestrielle DmfA est explicite :

- aujourd'hui, les employeurs, et/ou leurs secrétariats sociaux, et/ou prestataires de services, communiquent trimestriellement les données de calcul des salaires. Ces données sont essentielles au calcul des droits des travailleurs ;
- demain, ces mêmes acteurs communiqueront en temps réel les évènements concernant un travailleur, dès leur survenance.

Dans le cadre du passage à e-Gov 3.0, la communication entre un déclarant (employeur ou son mandataire) et la Sécurité Sociale doit par conséquent être redéfinie. Ce document a pour but de décrire en particulier les échanges effectués par SFTP entre un déclarant et l'Office national de Sécurité sociale (ONSS), en spécifiant les inputs attendus de la part des déclarants et les réponses générées en retour par la Sécurité Sociale.

Ces échanges, inputs et outputs, seront basés sur le format CloudEvent, qui est une structure simple en JSON, de maximum 64KB, et ces échanges seront spécifiés par AsyncAPI, qui est un "contrat" similaire à ceux que l'on connaît en REST-API.

DOCUMENTATION COMPLÉMENTAIRE

- Projet e-Gov 3.0 : https://www.socialsecurity.be/site_fr/employer/infos/e-gov.htm
- Spécifications CloudEvent :
<https://github.com/cloudevents/spec/blob/main/cloudevents/spec.md>
- Spécifications AsyncAPI : <https://www.asyncapi.com/en>

2 – PRÉREQUIS AUX ÉCHANGES AVEC LA SÉCURITÉ SOCIALE

2.1 – ENREGISTREMENT DE L'ACCÈS

Le déclarant, pour échanger des données structurées avec la Sécurité Sociale, doit au préalable disposer d'un accès au [service en ligne "ChaMan - Gestion des canaux techniques"](#).

Le Gestionnaire d'Accès de l'entreprise a automatiquement accès à la plateforme. Pour les utilisateurs ordinaires, un Gestionnaire d'Accès doit leur attribuer un accès explicite à "ChaMan - Gestion des canaux techniques" dans le [service en ligne "Gestion des accès"](#).

Une fois connecté à ChaMan, le déclarant peut créer un canal SFTP, en suivant les étapes ci-dessous :

1. entrer les données de contact de la personne responsable de la gestion des canaux (un numéro d'expéditeur est alors attribué) ;
2. cliquer ensuite sur le bouton "Ajouter un compte FTP ou SFTP" et choisir le type SFTP ;
3. une fois sélectionné, la liste des champs associés s'affichera ;
4. entrer le nom du compte (un libellé indicatif) ;
5. choisir une ou plusieurs permissions pour ce compte. Dans le cadre d'e-Gov 3.0, de nouvelles permissions seront ajoutées ;
6. télécharger un certificat en spécifiant son nom ;
7. indiquer le nom d'utilisateur du compte SFTP et mettre la clé SSH publique ;
8. enfin, appuyer sur le bouton 'Valider' pour confirmer la création du compte SFTP.

Le canal SFTP de communication avec la Sécurité Sociale est créé.

Pour toute modification, il faut se référer au [guide d'utilisation de la plateforme ChaMan](#).

DOCUMENTATION COMPLÉMENTAIRE

- Créer un canal SFTP sur le portail de la Sécurité Sociale :
https://www.socialsecurity.be/site_fr/general/helpcentre/batch/document/pdf/step_5_sftp_F.pdf

2.2 – PROTOCOLE DE COMMUNICATION

La configuration et l'utilisation du canal SFTP du déclarant font l'objet d'une documentation technique renseignée ci-dessous.

Le répertoire d'échange du déclarant contient quatre sous-répertoires :

- IN
- INTEST
- OUT
- OUTTEST

DOCUMENTATION COMPLÉMENTAIRE

- Choisir un client SFTP :
https://www.socialsecurity.be/site_fr/general/helpcentre/batch/document/pdf/step_2_sftp_F.pdf
- Choisir un générateur de clés :
https://www.socialsecurity.be/site_fr/general/helpcentre/batch/document/pdf/step_3_sftp_F.pdf
- Créer une paire de clés :
https://www.socialsecurity.be/site_fr/general/helpcentre/batch/document/pdf/step_4_sftp_F.pdf
- Paramétriser le client SFTP :
https://www.socialsecurity.be/site_fr/general/helpcentre/batch/document/pdf/step_6_sftp_F.pdf

2.2.1 – Protocole d'envoi

Les répertoires IN et INTEST sont les répertoires dans lesquels les fichiers destinés à la Sécurité Sociale doivent être déposés par le déclarant.

Le répertoire IN reçoit les fichiers réels (dans l'environnement de production) tandis que le répertoire INTEST reçoit les fichiers de test (dans l'environnement de simulation).

Le détail des fichiers qui doivent être déposés est repris au point [4](#).

2.2.2 – Protocole de réception

Les répertoires OUT et OUTTEST sont utilisés pour communiquer les outputs générés par la Sécurité Sociale.

Le répertoire OUT reçoit les fichiers réels (dans l'environnement de production) tandis que le répertoire OUTTEST reçoit les fichiers de test (dans l'environnement de simulation).

Le but est que le déclarant récupère ses fichiers, pour ensuite vider le répertoire, afin de libérer de l'espace sur le serveur.

Le détail des fichiers retournés est repris au point [5](#).

3 – DESCRIPTION DES ÉCHANGES DÉCLARANT – ONSS

3.1 – DESCRIPTION GÉNÉRALE DES ÉCHANGES

Pour soumettre une déclaration, l'employeur ou son mandataire doit déposer dans le répertoire IN/INTEST de son canal SFTP un fichier d'input (fichier FI), sous forme de liste de CloudEvents. Ce fichier d'input doit être accompagné d'un fichier de signature et d'un fichier GO. Le contenu exact du dépôt est décrit au point [4](#).

L'ONSS détectera endéans une heure qu'un dépôt a été effectué et le prendra en charge. Une première vérification formelle sur l'envoi sera alors effectuée.

Si l'envoi est valide, un ackEvent positif sera créé par l'ONSS. En cas d'invalidité du dépôt, un ackEvent négatif sera généré et le traitement prendra fin.

En cas de succès du dépôt, l'ONSS continuera son traitement en procédant au découpage de la liste pour isoler chaque CloudEvent. Une série de vérifications supplémentaires, davantage orientées sur le contenu, sera effectuée sur chacun d'entre eux.

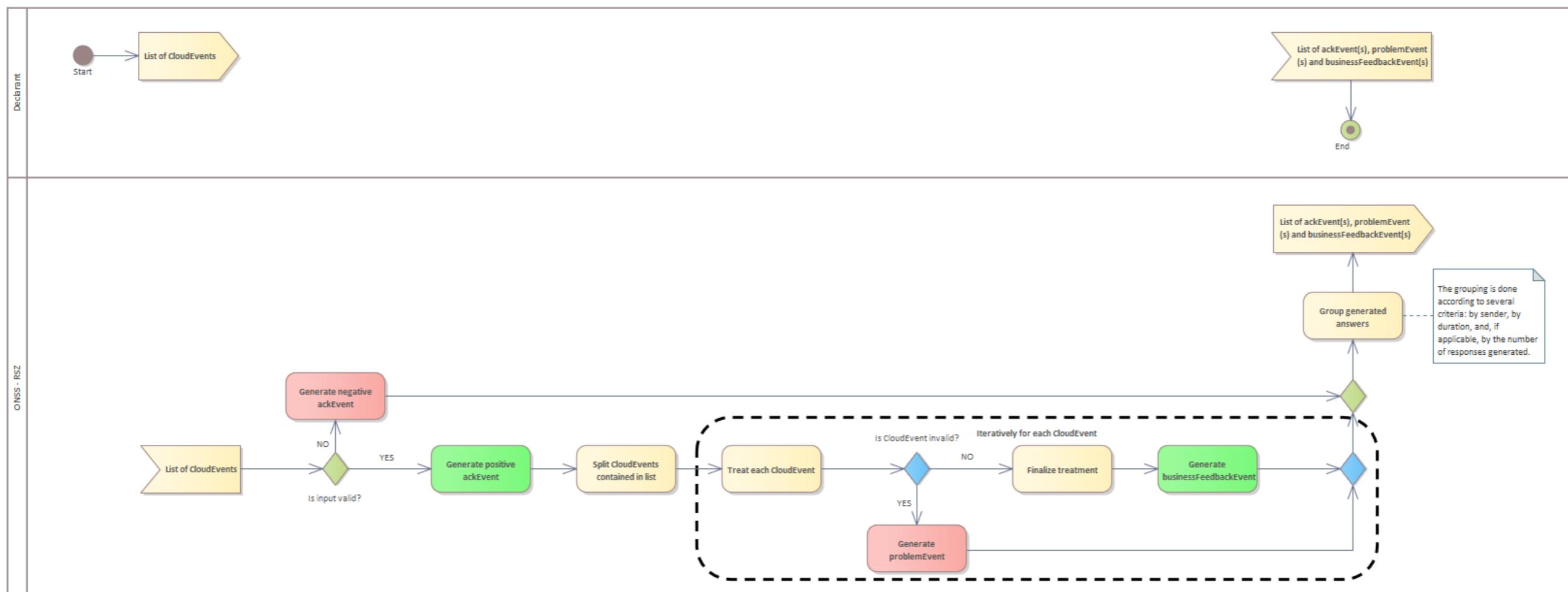
En cas d'anomalie sur un CloudEvent, un problemEvent sera généré et le traitement prendra fin pour le CloudEvent concerné. Parallèlement, le traitement des CloudEvents valides sera finalisé et un businessFeedbackEvent sera créé pour chacun d'entre eux.

Le contenu exact des réponses retournées par l'ONSS est décrit au point [5](#). Le contenu du businessFeedbackEvent, lui, est défini par les équipes business concernées.

Il est important de noter que toutes les réponses générées par l'ONSS seront regroupées dans un ou plusieurs fichier(s) d'output (fichier FO), déposé(s) sur le répertoire OUT/OUTTEST du déclarant. Le fichier d'output est donc une liste qui peut regrouper à la fois des ackEvents, des problemEvents et des businessFeedbackEvents.

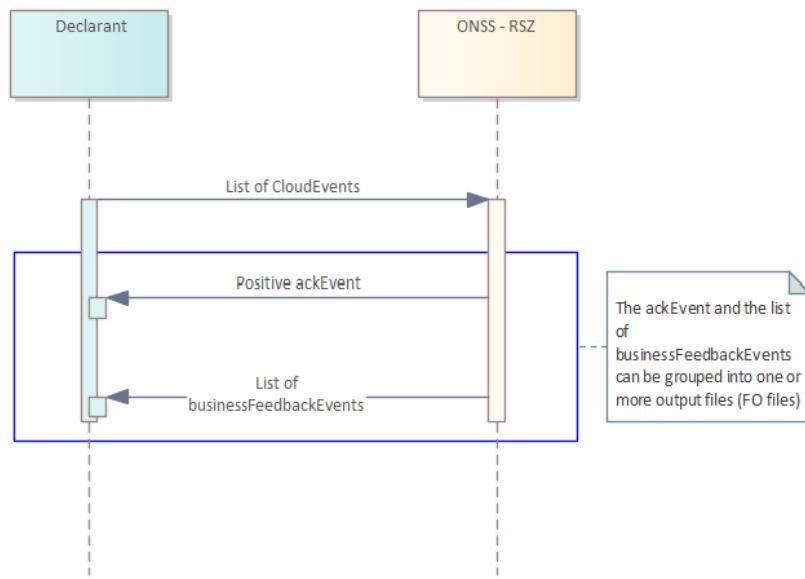
La logique CloudEvent fait que le contenu n'est plus corrélé à un contenant. Le fichier d'output retourné par l'ONSS ne sera donc pas nécessairement lié à un fichier d'input spécifique soumis par le déclarant.

3.2 – DIAGRAMME D'ACTIVITÉS

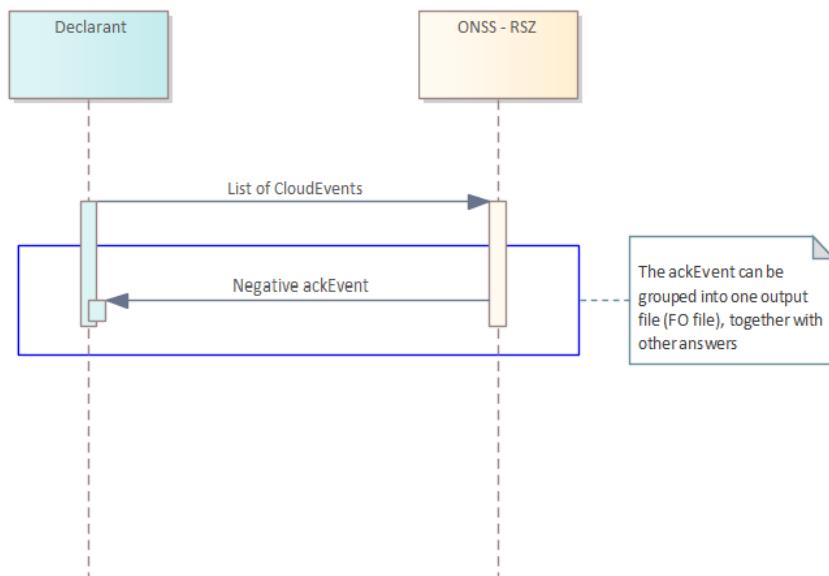


3.3 – DIAGRAMMES DE SÉQUENCE

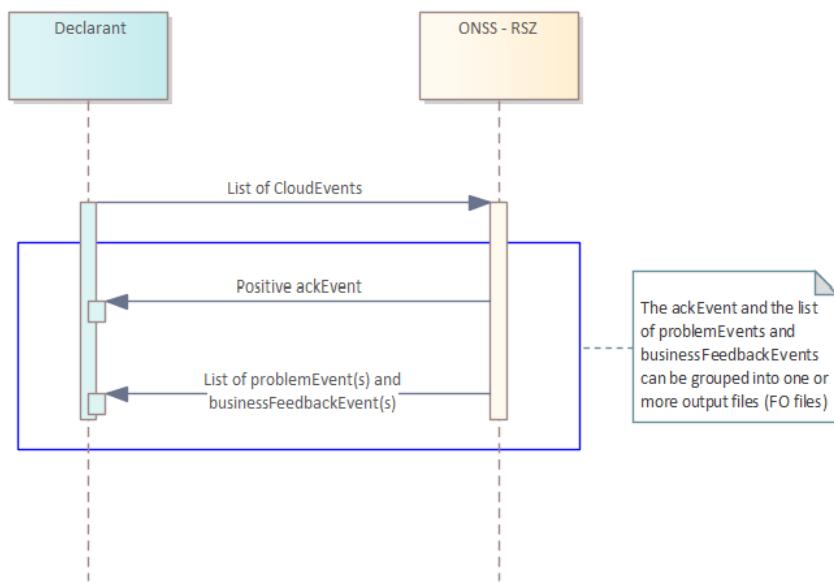
3.3.1 – Scénario 1 : le groupe d'envoi est valide et tous les CloudEvents sont également valides



3.3.2 – Scénario 2 : le groupe d'envoi est invalide



3.3.3 – Scénario 3 : le groupe d'envoi est valide et un ou plusieurs CloudEvents sont invalides



4 – DÉPÔT EFFECTUÉ PAR LE DÉCLARANT

Pour être correctement traité, le dépôt du déclarant doit se composer obligatoirement d'un fichier de déclaration (FI), accompagné d'un fichier de signature (FS) et d'un fichier GO.

| Fichiers attendus | Description | Format | Volume maximum |
|-------------------|---|--------|---|
| Fichier FI | Liste de 1 à n CloudEvents, chaque CloudEvent étant structuré par un header et un payload, dans lequel le déclarant insère ses données business. Les détails liés à la génération du fichier sont repris au point 4.2.1 | JSON | Le fichier doit avoir un volume maximum de 90MB non-compressé. La limite pourra être revue. Le nombre maximal de CloudEvents contenus dans le fichier doit être défini. Un cloudevent ne devrait pas avoir un volume supérieur à 64kb. |
| Fichier FS | Signature électronique en rapport avec le fichier de données FI. Les détails liés à la génération du fichier sont repris au point 4.2.2 | Base64 | - |
| Fichier GO | Fichier vide qui signale à l'ONSS que le fichier FI est prêt à être traité. Les détails liés à la génération du fichier sont repris au point 4.2.3 | - | - |

4.1 – STRUCTURE DU NOM DES FICHIERS

Le nom de chaque fichier doit respecter une nomenclature en cinq parties, chacune séparée par un point :

| Partie | Exemple | Valeur admissibles | Obligatoire | Description |
|--------|---------|--------------------|-------------|--|
| (1) | FI | FI / FS / GO | Obligatoire | Type de contenu |
| (2) | EVENT | EVENT | Obligatoire | Valeur fixe qui permet d'associer un fichier à un CloudEvent |

| | | | | |
|-----|--------------------------------------|--------|-------------|--|
| (3) | 000640 | | Obligatoire | Numéro d'expéditeur attribué au déclarant et connu par le service ChaMan (voir point 2.1) |
| (4) | 3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42 | UUID | Obligatoire | Identifiant unique du groupe d'envoi, c'est-à-dire du fichier d'input, accompagné de son fichier de signature et de son fichier GO |
| (5) | R | R T | Obligatoire | Environnement : R → fichiers réels (production) T → fichiers de test (simulation) |

EXEMPLE D'UN GROUPE D'ENVOI

- FI.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R
- FS.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R
- GO.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R

4.2 – DESCRIPTION DES FICHIERS

4.2.1 – Fichier de déclaration (FI)

Le fichier FI est une liste de cloudevents, encodée en JSON. Le fichier est structuré sous forme de tableau "messages", qui contient des objects cloudevents. Ceux-ci transportent les données business de la déclaration. Il faut noter que le contenu du fichier n'est pas lié à des données business spécifiques, et qu'il peut accueillir un mélange de CloudEvents de type différent.

Le fichier non-compressé doit avoir une taille maximale de 90MB. Cette limite pourra être revue au besoin. Le nombre maximal de CloudEvents doit encore être défini.

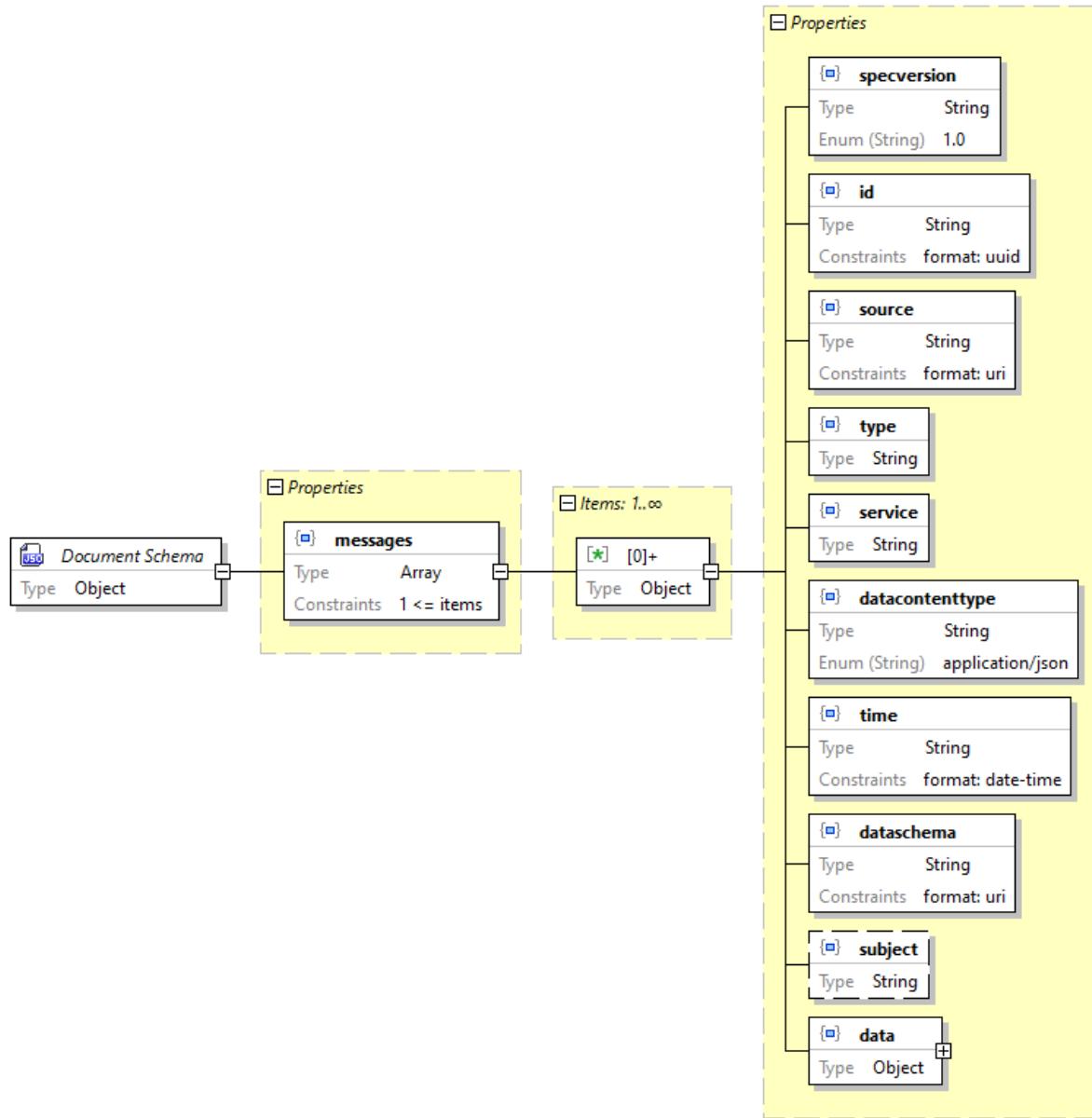
4.2.1.1 – Description technique d'un CloudEvent

Chaque CloudEvent est un object composé d'un header et d'un payload (bloc "data") dans lequel le déclarant insère ses données business. Un CloudEvent est structuré comme suit :

| Attribut | Format | Obligatoire | Exemple | Description |
|-----------------|---------------|--------------------|--|--|
| specversion | String | Obligatoire | 1.0 | La version de la spécification CloudEvents |
| id | UUID | Obligatoire | a81bc81b-dead-4e5d-abff-90865d1e13b1 | Identifie, pour une source donnée, un CloudEvent, de manière unique. L'id doit respecter les spécifications UUID |
| source | URI-reference | Obligatoire | urn:myprogram:expeditorId:0123456789 | Identifie le producteur. La valeur doit être entrée sous forme d'urn. La structure est libre mais l'urn doit se terminer par l'expeditorId (voir point 2.1) : <i>urn [:free value] : expeditorId : ChaManId</i> |
| type | String | Obligatoire | be.socialsecurity.services.salaryData.v1.salary.create | Cet attribut contient une valeur décrivant le type de CloudEvent. L'attribut doit suivre cette nomenclature : <i>serviceName .v majorVersion .resourceType[.childResourceType] .method[.suffix]</i> Les ressources et méthodes utilisées seront spécifiées par les équipes business, qui les auront préalablement communiquées à l'ONSS |
| service | String | Obligatoire | be.socialsecurity.services.salaryData.v1 | Identifie la description API qui définit les types de CloudEvent. L'attribut doit suivre cette nomenclature : <i>serviceName .v majorVersion</i> Le nom du service et la version seront spécifiés par les équipes business |

| | | | | |
|-----------------|----------------------|-------------|---------------------------|---|
| subject | String | Facultatif | | Identifie le sujet du CloudEvent dans le contexte du producteur (identifié par <i>source</i>). Utilisé lorsque la source à elle seule ne permet pas de discriminer un CloudEvent auquel un traitement spécifique doit s'appliquer (agit comme un filtre) |
| time | Timestamp (RFC 3339) | Obligatoire | 2025-08-20T12:10:13+02:00 | Timestamp |
| dataschema | URI | Obligatoire | salary-create-events.yaml | Identifie le schéma de validation des données business, dans le payload. Celui-ci doit être fourni par les équipes business concernées |
| datacontenttype | String (RFC 2046) | Obligatoire | application/json | Mimetype des données. Le mimetype doit toujours être <i>application/json</i> |
| data | Object | Obligatoire | | Payload (données business) |

4.2.1.2 – Schéma JSON du fichier FI



4.2.1.3 – Exemple de fichier FI

```
{  
    "messages": [  
        {  
            "specversion": "1.0",  
            "id": "a81bc81b-dead-4e5d-abff-90865d1e13b1",  
            "source": "urn:myprogram:expeditorId:0123456789",  
            "type": "be.socialsecurity.services.salaryData.v1.salary.create",  
            "service": "be.socialsecurity.services.salaryData.v1",  
            "datacontenttype": "application/json",  
            "time": "2025-08-20T12:10:13+02:00",  
            "dataschema": "salary-create-events.yaml",  
            "subject": "String",  
            "data": { -- declarant's business data --}  
        },  
        {  
            "specversion": "1.0",  
            "id": "7c6b2d9f-3c3a-4b6a-9f6a-8c6b8e1f4d2a",  
            "source": "urn:myprogram:expeditorId:0123456789",  
            "type": "be.socialsecurity.services.salaryData.v1.salary.create",  
            "service": "be.socialsecurity.services.salaryData.v1",  
            "datacontenttype": "application/json",  
            "time": "2025-08-20T12:10:15+02:00",  
            "dataschema": "salary-create-events.yaml",  
            "subject": "String",  
            "data": { -- declarant's business data --}  
        }  
    ]  
}
```

4.2.2 – Fichier de signature (FS)

Les documents envoyés vers le point de transfert de la Sécurité sociale doivent être signés numériquement par les déclarants. Chaque fichier FI de déclaration doit donc être accompagné du fichier FS correspondant.

Le fichier de signature est obligatoire dans l'environnement de production et dans l'environnement de simulation.

Le fichier de signature doit être créé avec le certificat qui a été renseigné lors de la création du canal SFTP.

Les certificats digitaux qualifiés suivants peuvent être utilisés :

- GlobalSign : PersonalSign 3 pro
- certificat de signature de la carte d'identité électronique

Le fichier de signature peut être créé avec OpenSSL, ou avec un programme d'une maison de soft, ou avec un programme développé par l'employeur ou son mandataire.

Il doit respecter les spécifications ci-dessous :

- utilisation du format de signature simple de type PKCS#7/CMS (signature binaire du fichier) ;
- l'algorithme de hachage utilisé par le processus de signature doit être de préférence SHA-256 ;
- la signature doit être basée sur un algorithme d'encryption de type RSA ou ECC (Elliptic curve) ;
- la taille des clés utilisées pour la signature doit être en accord avec les normes de sécurité actuelles (recommandations du NIST ou de l'ANSSI) ;
- la signature doit être encodée en base64. Ce base64 peut être encodé sur une seule ligne ou être formaté en colonne de 76 caractères (c'est-à-dire que le base64 sera représenté en plusieurs lignes de taille fixe). En cas de formatage en colonne, le caractère de retour à la ligne devra être un simple « Carriage Return » (CR – 0x0D) ;
- le certificat de signature utilisé devra avoir été certifié par l'une des autorités de certification acceptées par le point de transfert Sécurité sociale.

DOCUMENTATION COMPLÉMENTAIRE

- https://www.socialsecurity.be/site_fr/general/helpcentre/digital_sign/index.htm

4.2.3 – Fichier GO

Le fichier GO est un fichier vide. Il est le signal que l'expéditeur a placé ses fichiers et que le traitement de ces fichiers peut commencer. Il doit toujours être placé **après** les fichiers FI et FS comme dernier fichier dans le dossier IN et INTEST.

Pour créer un fichier GO, il faut ouvrir un fichier vide (par exemple un fichier texte) et le sauvegarder sous le nom de fichier correct.

5 – RÉPONSES RETOURNÉES PAR L'ONSS

Les réponses générées par l'ONSS sont communiquées dans un fichier d'output (fichier FO), accompagné d'un fichier de signature et d'un fichier GO.

5.1 – STRUCTURE DU NOM DES FICHIERS

Le nom de chaque fichier respecte une nomenclature en cinq parties, chacune séparée par un point :

| Partie | Exemple | Valeur admissibles | Obligatoire | Description |
|--------|--------------------------------------|--------------------|-------------|---|
| (1) | FO | FO / FS / GO | Obligatoire | Type de contenu |
| (2) | EVENT | EVENT | Obligatoire | Valeur fixe qui permet d'associer un fichier à un CloudEvent |
| (3) | 999999 | 999999 | Obligatoire | Numéro d'expéditeur de l'ONSS |
| (4) | e4c1d8f7-6a3b-4c92-9e5f-2b8a0d6c1f34 | UUID | Obligatoire | Identifiant unique du groupe d'envoi, c'est-à-dire du fichier d'output, accompagné de son fichier de signature et de son fichier GO |
| (5) | R | R T | Obligatoire | Environnement : R → fichiers réels (production) T → fichiers de test (simulation) |

EXEMPLE DE RÉPONSE

- FO.EVENT.999999.e4c1d8f7-6a3b-4c92-9e5f-2b8a0d6c1f34.R
- FS.EVENT.999999.e4c1d8f7-6a3b-4c92-9e5f-2b8a0d6c1f34.R
- GO.EVENT.999999.e4c1d8f7-6a3b-4c92-9e5f-2b8a0d6c1f34.R

5.2 – DESCRIPTION DES FICHIERS

5.2.1 – Fichier de réponses (FO)

Un fichier FO est une liste de CloudEvents et regroupe les différentes réponses de l'ONSS. Il peut potentiellement contenir un mélange d'ackEvents, de problemEvents et de businessFeedbackEvents.

Le fichier FO est structuré sous forme de tableau "messages", contenant les différents objects CloudEvents.

Chaque fichier FO est identifié par un UUID unique. Le fichier d'output ne répond pas à un fichier d'input spécifique et il n'y a pas de correspondance entre le UUID qui identifie le FI et celui qui identifie le FO.

La présente documentation a pour but de fournir la structure des réponses que le déclarant recevra. Le détail des valeurs qui pourront être admises dans les différents attributs sera communiquée ultérieurement.

5.2.1.1 – Description des CloudEvents de réponse

5.2.1.1.1 - ackEvent

En réponse à un groupe d'envoi, l'ONSS retourne un ackEvent, positif si le groupe d'envoi est valide et négatif si celui-ci est invalide.

La validation se fait au niveau des fichiers à ce stade-ci, pas au niveau des CloudEvents.

[5.2.1.1.1.1 – Description technique](#)

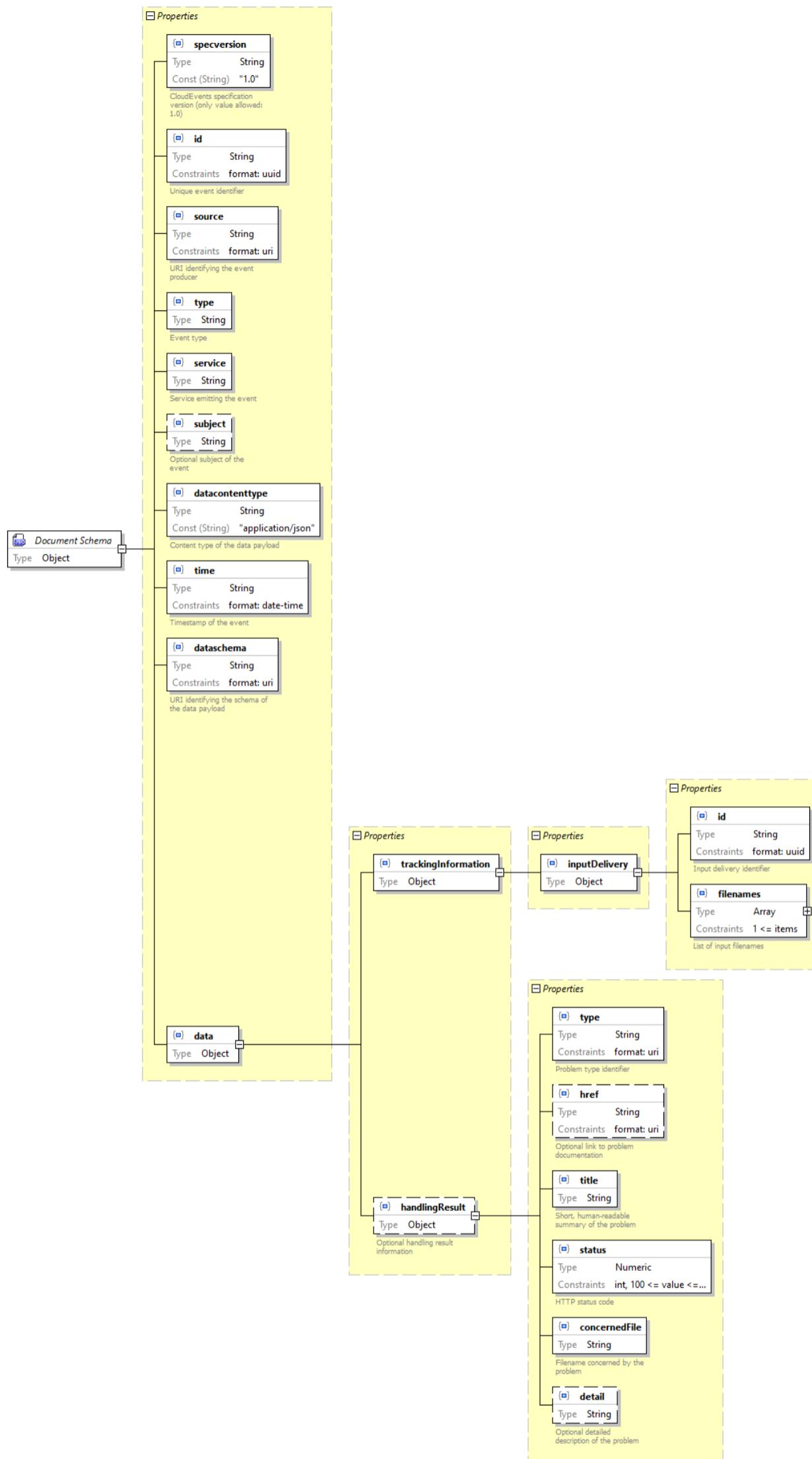
L'ackEvent est un CloudEvent, encodé en JSON. La structure de celui-ci est décrite dans le tableau ci-dessous.

| Attribut | Format | Obligatoire | Exemple | Description |
|-----------------|---------------|--------------------|---|---|
| specversion | String | Obligatoire | 1.0 | La version de la spécification CloudEvents |
| id | UUID | Obligatoire | db8fad51-f837-4afb-bfc1-08ea904aedcd | Identifie, pour une source donnée, un CloudEvent, de manière unique. Il s'agit d'un UUID généré par l'ONSS |
| source | URI-reference | Obligatoire | urn:be:socialsecurity:expeditorId:999999 | Identifie le producteur, ici l'ONSS |
| type | String | Obligatoire | be.socialsecurity.services. stream.outputgenerator.v1. file.notify.validated be.socialsecurity.services. stream.outputgenerator.v1. file.notify.rejected | Le type retourné par l'ONSS, mis sur <i>validated</i> , indique que l'envoi du déclarant a été accepté. Le type mis sur <i>rejected</i> indique que l'envoi a été refusé. Dans ce cas, le bloc <i>data</i> contient un object <i>handlingResult</i> (celui-ci n'est présent qu'en cas d'erreur) |
| service | String | Obligatoire | be.socialsecurity.services. stream.outputgenerator.v1 | Identifie la description API qui définit les types de CloudEvent |

| | | | | |
|-----------------------|-------------------------|-------------|--------------------------------------|---|
| subject | String | Facultatif | | Identifie le sujet du CloudEvent dans le contexte du producteur (identifié par <i>source</i>). Utilisé lorsque la source à elle seule ne permet pas de discriminer un CloudEvent auquel un traitement spécifique doit s'appliquer (agit comme un filtre) |
| time | Timestamp (RFC 3339) | Obligatoire | 2025-08-20T12:10:13+02:00 | Timestamp |
| dataschema | URI | Obligatoire | ack-1.0.yaml | Identifie le schéma de validation du payload. Celui-ci doit être fourni par l'ONSS |
| datacontenttype | String (RFC 2046) | Obligatoire | application/json | Mimetype des données |
| data | Object | Obligatoire | | Payload |
| --trackingInformation | Object | Obligatoire | | |
| ----inputDelivery | Object | Obligatoire | | |
| -----id | UUID | Obligatoire | 7c0ba8fe-28cd-4893-8fe2-b55e95517c86 | Identifiant attribué par l'ONSS à un groupe de fichiers reçus en input |

| | | | | |
|-------------------|---------|-------------|--|---|
| -----filenames | Array | Obligatoire | FI.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R, FS.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R, GO.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R | Liste des fichiers reçus |
| --handlingResult | Object | Facultatif | | Objet contenant le statut de l'envoi, selon le standard Belgif et s'approchant des openAPI's classiques. Uniquement présent en cas d'erreur |
| ----type | URI | Obligatoire | urn:problem-type:belgif:input-validation:invalidInput | URN identifiant le type d'erreur |
| ----href | URI | Facultatif | https://www.belgif.be/specification/rest/api-guide/issues/invalidInput.html | URL vers la documentation de l'erreur |
| ----title | String | Obligatoire | Unsupported Media Type | Libellé résumant le résultat du traitement |
| ----status | Integer | Obligatoire | 415 | Code reprenant le résultat du traitement |
| ----concernedFile | String | Obligatoire | FI.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R | Liste du fichier sur lequel une erreur a été rencontrée |
| ----detail | String | Facultatif | Content-Type 'application/xml' is not supported.Expected 'application/json' | Description plus détaillée du résultat du traitement |

5.2.1.1.2 – Schéma JSON



5.2.1.1.3 – Exemple d'ackEvent positif

```
{  
    "specversion": "1.0",  
    "id": "db8fad51-f837-4afb-bfc1-08ea904aedcd",  
    "source": "urn:be:socialsecurity:expeditorId:999999",  
    "type": "be.socialsecurity.services.stream.outputgenerator.v1.file.notify.validated",  
    "service": "be.socialsecurity.services.stream.outputgenerator.v1",  
    "datacontenttype": "application/json",  
    "time": "2025-08-20T12:10:13+02:00",  
    "dataschema": "ack-1.0.yaml",  
    "data": {  
        "trackingInformation": {  
            "inputDelivery": {  
                "id": "d4bfc0ef-bbbf-43fc-a626-d2d1f9521060",  
                "filenames": [  
                    "FI.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",  
                    "FS.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",  
                    "GO.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R"  
                ]  
            }  
        }  
    }  
}
```

5.2.1.1.4 – Exemple d'ackEvent négatif

```
{  
    "specversion": "1.0",  
    "id": "ebaef4b8-e82f-4037-9748-049c16dc35c3",  
    "source": "urn:be:socialsecurity:expeditorId:999999",  
    "type": "be.socialsecurity.services.stream.outputgenerator.v1.file.notify.rejected",  
    "service": "be.socialsecurity.services.stream.outputgenerator.v1",  
    "datacontenttype": "application/json",  
    "time": "2025-08-20T12:10:13+02:00",  
    "dataschema": "ack-1.0.yaml",  
    "data": {  
        "trackingInformation": {  
            "inputDelivery": {  
                "id": "7c0ba8fe-28cd-4893-8fe2-b55e95517c86",  
                "fileNames": [  
                    "FI.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R",  
                    "FS.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R",  
                    "GO.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R"  
                ]  
            }  
        },  
        "handlingResult": {  
            "type": "urn:problem-type:belgif:input-validation:invalidInput",  
            "href": "https://www.belgif.be/specification/rest/api-guide/issues/invalidInput.html",  
            "title": "Unsupported Media Type",  
            "status": 415,  
            "concernedFile": "FI.EVENT.000640.be8ed1e9-f3c1-416b-a770-f3871472e036.R",  
            "detail": "Content-Type 'application/xml' is not supported. Expected 'application/json'"  
        }  
    }  
}
```

5.2.1.1.2 - problemEvent

Un déclarant soumet un groupe d'envoi valide mais un ou plusieurs CloudEvents, passés dans le fichier de déclaration FI, sont invalides. L'ONSS retourne alors en réponse un ou plusieurs problemEvent(s). Chaque problemEvent se rapporte à un CloudEvent reçu en input.

5.2.1.1.2.1 – Description technique

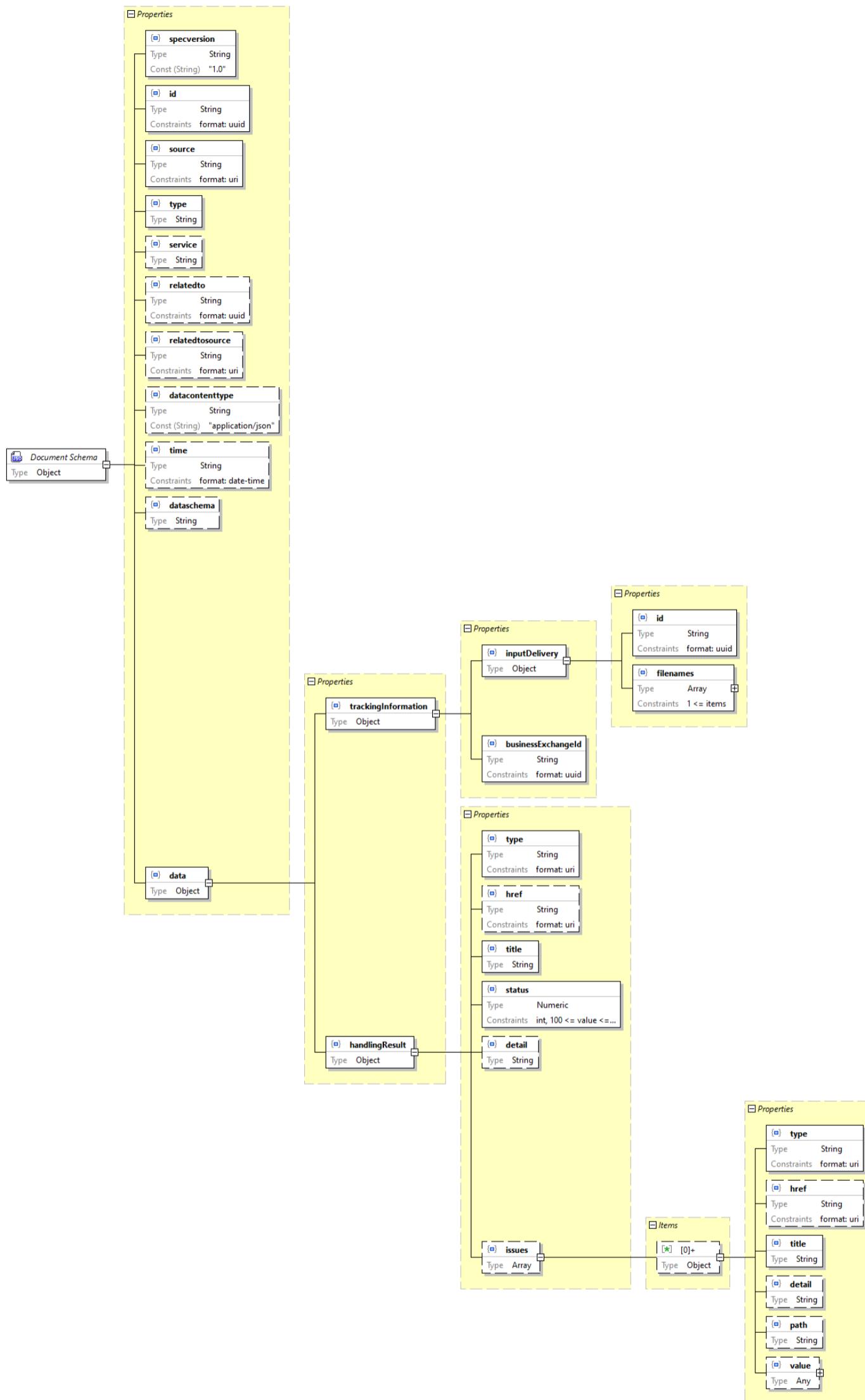
Un problemEvent est un CloudEvent, encodé en JSON. Il est structuré comme suit :

| Attribut | Format | Obligatoire | Exemple | Description |
|-----------------|---------------|-------------|--|---|
| specversion | String | Obligatoire | 1.0 | La version de la spécification CloudEvents |
| id | UUID | Obligatoire | db8fad51-f837-4afb-bfc1-08ea904aedcd | Identifie, pour une source donnée, un CloudEvent, de manière unique. Il s'agit d'un UUID généré par l'ONSS |
| source | URI-reference | Obligatoire | urn:be:socialsecurity:expeditorId:999999 | Identifie le producteur, ici l'ONSS |
| relatedto | UUID | Obligatoire | 8b1c78ee-289b-47c9-ac53-e09da7ab3a7e | Valeur de l'id du CloudEvent auquel le problemEvent se rapporte |
| relatedtosource | String | Obligatoire | urn:be:myprogram:expeditorId:0123456789 | Valeur de la source du CloudEvent auquel le problemEvent se rapporte |
| type | String | Obligatoire | be.socialsecurity.services.stream.outputgenerator.v1.businessEvent.notify.rejected | Le type retourné par l'ONSS, mis sur <i>rejected</i> , indique qu'un CloudEvent soumis par le déclarant a été refusé |
| service | String | Obligatoire | be.socialsecurity.services.stream.outputgenerator.v1 | Identifie la description API qui définit les types d'Events |
| subject | String | Facultatif | | Identifie le sujet du CloudEvent dans le contexte du producteur (identifié par <i>source</i>). Utilisé lorsque la source à elle seule ne permet pas de discriminer un CloudEvent auquel un traitement spécifique doit s'appliquer (agit comme un filtre) |

| | | | | |
|------------------------|----------------------|-------------|--|--|
| time | Timestamp (RFC 3339) | Obligatoire | 2025-08-20T12:10:13+02:00 | Timestamp |
| dataschema | URI | Obligatoire | problemEvent-1.0.yaml | Identifie le schéma de validation du payload. Celui-ci doit être fourni par l'ONSS |
| datacontenttype | String (RFC 2046) | Obligatoire | application/json | Mimetype des données |
| data | Object | Obligatoire | | Payload |
| --trackingInformation | Object | Obligatoire | | |
| ----inputDelivery | Object | Obligatoire | | |
| -----id | UUID | Obligatoire | db8fad51-f837-4afb-bfc1-08ea904aedcd | Identifiant attribué par l'ONSS à un groupe de fichiers reçus en input |
| -----filenames | Array | Obligatoire | FI.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R, FS.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R, GO.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R | Liste des fichiers reçus |
| ----businessExchangeId | UUID | Obligatoire | e3c6ad5b-11b2-4e12-8675-5f08abc03fa3 | Identifiant attribué par l'ONSS à un échange de CloudEvents |

| | | | | |
|------------------|---------|-------------|---|--|
| --handlingResult | Object | Obligatoire | | Objet contenant le statut du traitement, selon le standard Belgif et s'approchant des openAPI's classiques |
| ----type | URI | Obligatoire | urn:problem-type:belgif:badRequest | URN identifiant le type d'erreur |
| ----href | URI | Facultatif | https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html | URL vers la documentation de l'erreur |
| ----title | String | Obligatoire | Bad Request | Libellé résumant le résultat du traitement |
| ----status | Integer | Obligatoire | 400 | Code reprenant le résultat du traitement |
| ----detail | String | Facultatif | The input message is incorrect | Description plus détaillée du résultat du traitement |
| ----issues | Array | Facultatif | | Tableau détaillant les erreurs |
| -----type | URI | Obligatoire | urn:problem-type:belgif:input-validation:schemaViolation | URN identifiant le type d'erreur |
| -----href | URI | Facultatif | https://www.belgif.be/specification/rest/api-guide/issues/schemaViolation.html | URL vers la documentation de l'erreur |
| -----title | String | Obligatoire | Input isn't valid with respect to schema | Libellé résumant le résultat du traitement |
| -----detail | String | Facultatif | enterpriseNumber should be numeric | Description plus détaillée du résultat du traitement |
| -----path | String | Facultatif | \$[1].data.enterpriseNumber | JSON Path de l'attribut où se situe l'erreur |
| -----value | String | Facultatif | abc | Valeur qui a déclenché l'erreur |

5.2.1.1.2.2 – Schéma JSON



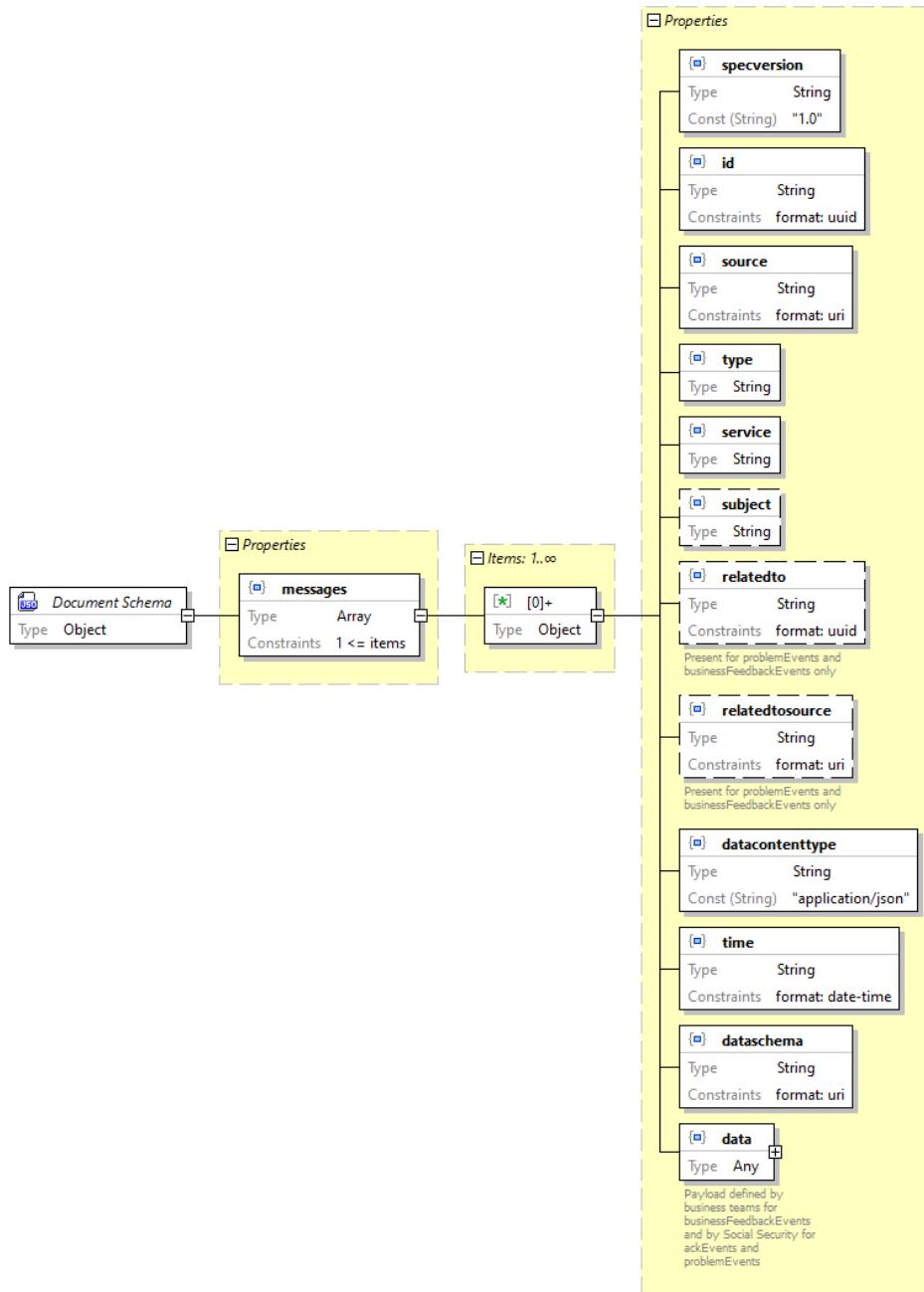
5.2.1.1.2.3 – Exemple

```
{  
    "specversion": "1.0",  
    "id": "db8fad51-f837-4afb-bfc1-08ea904aedcd",  
    "source": "urn:be:socialsecurity:expeditorId:9999999",  
    "relatedto": "8b1c78ee-289b-47c9-ac53-e09da7ab3a7e",  
    "relatedtosource": "urn:be:myprogram :expeditorId:0123456789",  
    "type": "be.socialsecurity.services.stream.outputgenerator.v1.businessEvent.notify.rejected",  
    "service": "be.socialsecurity.services.stream.outputgenerator.v1",  
    "datacontenttype": "application/json",  
    "time": "2025-08-20T12:10:13+02:00",  
    "dataschema": "problemEvent-1.0.yaml",  
    "data": {  
        "trackingInformation": {  
            "inputDelivery": {  
                "id": "db8fad51-f837-4afb-bfc1-08ea904aedcd",  
                "filenames": [  
                    "FI.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",  
                    "FS.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",  
                    "GO.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R"  
                ]  
            },  
            "businessExchangeId": "e3c6ad5b-11b2-4e12-8675-5f08abc03fa3"  
        },  
        "handlingResult": {  
            "type": "urn:problem-type:belgif:badRequest",  
            "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",  
            "title": "Bad Request",  
            "status": 400,  
            "detail": "The input message is incorrect",  
            "issues": [  
                {  
                    "type": "urn:problem-type:belgif:input-validation:schemaViolation",  
                    "href": "https://www.belgif.be/specification/rest/api-guide/issues/schemaViolation.html",  
                    "title": "Input isn't valid with respect to schema",  
                    "detail": "enterpriseNumber should be numeric",  
                    "path": "$.data.enterpriseNumber",  
                    "value": "abc"  
                },  
                {  
                    "type": "urn:problem-type:belgif:input-validation:unknownInput",  
                    "href": "https://www.belgif.be/specification/rest/api-guide/issues/unknownInput.html",  
                    "title": "Unknown input",  
                    "detail": "Input SSIN is unknown",  
                    "path": "$.data.boardMembers[0].ssin",  
                    "value": "12345678901"  
                }  
            ]  
        }  
    }  
}
```

5.2.1.1.3 – businessFeedbackEvent

Les spécifications du businessFeedbackEvent sont fournies par les équipes business concernées.

5.2.1.2 – Schéma JSON du fichier FO



5.2.1.3 – Exemple de fichier FO

```
{
  "messages": [
    {
      "specversion": "1.0",
      "id": "db8fad51-f837-4afb-bfc1-08ea904aedcd",
      "source": "urn:be:socialsecurity:expeditorId:999999",
      "relatedto": "8b1c78ee-289b-47c9-ac53-e09da7ab3a7e",
      "relatedtosource": "urn:be:myprogram:expeditorId:0123456789",
      "type": "be.socialsecurity.services.stream.outputgenerator.v1.businessEvent.notify.rejected",
      "service": "be.socialsecurity.services.stream.outputgenerator.v1",
      "datacontenttype": "application/json",
      "time": "2025-08-20T12:10:13+02:00",
      "dataschema": "problemEvent-1.0.yaml",
      "data": {
        "trackingInformation": {
          "inputDelivery": {
            "id": "db8fad51-f837-4afb-bfc1-08ea904aedcd",
            "filenames": [
              "FI.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",
              "FS.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R",
              "GO.EVENT.000640.3f2c8b9a-7e4d-4f1c-a6c2-9c6e5b8f1d42.R"
            ]
          },
          "businessExchangeId": "e3c6ad5b-11b2-4e12-8675-5f08abc03fa3"
        },
        "handlingResult": {
          "type": "urn:problem-type:belgif:badRequest",
          "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",
          "title": "Bad Request",
          "status": 400,
          "detail": "The input message is incorrect",
          "issues": [
            {
              "type": "urn:problem-type:belgif:input-validation:schemaViolation",
              "href": "https://www.belgif.be/specification/rest/api-guide/issues/schemaViolation.html",
              "title": "Input isn't valid with respect to schema",
              "detail": "enterpriseNumber should be numeric",
              "path": "$.data.enterpriseNumber",
              "value": "abc"
            },
            {
              "type": "urn:problem-type:belgif:input-validation:unknownInput",
              "href": "https://www.belgif.be/specification/rest/api-guide/issues/unknownInput.html",
              "title": "Unknown input",
              "detail": "Input SSIN is unknown",
              "path": "$.data.boardMembers[0].ssin",
              "value": "12345678901"
            }
          ]
        }
      }
    },
    {
      "specversion": "1.0",
      "id": "c1b6f6d9-6a1b-4c7c-9d3f-9b0a5c9d6c2a",
      "source": "urn:myprogram:expeditorId:999999",
      "relatedto": "8b1c78ee-289b-47c9-ac53-e09da7ab3a7e",
      "relatedtosource": "urn:be:myprogram:expeditorId:0123456789",
      "type": "be.socialsecurity.services.salaryData.v1.salary.notify.created",
      "service": "be.socialsecurity.services.salaryData.v1",
      "datacontenttype": "application/json",
      "time": "2025-08-20T12:10:15+02:00",
      "dataschema": "businessFeedbackEvent.yaml",
    }
  ]
}
```

```
        "subject": "String",
        "data": {-- business answer to declarant's business data --}
    }
}
```

5.2.2 – Fichier de signature (FS)

L'ONSS joindra à chaque fichier FO un fichier de signature (FS). Il s'agit d'un fichier encodé en base 64.

5.2.3 – Fichier GO

L'ONSS joindra également un fichier GO pour signaler au déclarant que la réponse a été chargée sur son répertoire et qu'elle est disponible à la consultation.

Après récupération des données, le déclarant doit nettoyer son répertoire.