Routeur Multipass

Documentation à destination des gestionnaires de consentements

// Version 2.01

**TABLE DES MATIÈRES**

[1. Version 3](#_Toc27133071)

[1.1 // Version 1.00 3](#_Toc27133072)

[1.2 // Version 2.00 3](#_Toc27133073)

[1.3 // Version 2.01 3](#_Toc27133074)

[2. Préambule 3](#_Toc27133075)

[3. L’ecosysteme multipass 3](#_Toc27133076)

[4. Les services 4](#_Toc27133077)

[4.1 // Le service d’authentification 4](#_Toc27133078)

[4.2 // Le service de vérification dES consentements 5](#_Toc27133079)

[a. Exposition en REST 5](#_Toc27133080)

[b. Exposition en SOAP 5](#_Toc27133081)

[c. Données en entrée 5](#_Toc27133082)

[d. Données en sortie 5](#_Toc27133083)

[4.3 // Le service de récupération des consentements 6](#_Toc27133084)

[a. Exposition en REST 6](#_Toc27133085)

[b. Exposition en SOAP 6](#_Toc27133086)

[c. Données en entrée 6](#_Toc27133087)

[d. Données en sortie 7](#_Toc27133088)

# Version

## Version 1.00

Version initiale du 16/04/19 par Julien CHARVET (OKTEO)

## Version 2.00

Version de diffusion du 17/06/19 par Bruno LAUGA (ARVALIS)

## Version 2.01

Version de diffusion du 13/12/19 par Béatrice BALVAY (Idele)

Correction mineure : Data**b**aseNumber au lieu de DataBaseNumber dans les interfaces Get et Check

# Préambule

Ce document a pour but de décrire les différents services qu’un gestionnaire de consentements devra exposer afin de pouvoir intégrer l’écosystème MULTIPASS, en particulier être interopérable avec le routeur.

MULTIPASS (2018-2020) est un projet de recherche financé par le Ministère Français en charge de l’Agriculture (CASDAR) et ses partenaires. Le projet vise à démontrer aux organisations professionnelles agricoles l’intérêt et la faisabilité d’un écosystème de gestion des consentements au travers de quelques cas d’usages concrets. Ces cas d’usage feront évoluer les présentes spécifications.

Les garanties apportées par les outils de l’écosystème MULTIPASS devraient rassurer les exploitants et favoriser les échanges de données de leurs exploitations. Ainsi, cela facilitera l’accès aux innovations actuelles et futures.

# L’ecosysteme multipass

Il existe déjà des solutions de gestion des consentements agricoles souvent conçues pour des besoins spécifiques. Ces gestionnaires de consentements multiples sont choisis librement par les acteurs de l’écosystème pour gérer leurs consentements. Les consentements sont donc enregistrés librement chez les gestionnaires avec pour seule contrainte d’enregistrer les informations prévues dans l’interface MULTIPASS.

L’outil au cœur de l’écosystème MULTIPASS est un routeur qui garantit l’interopérabilité des différents gestionnaires de consentements. Il permet un accès unifié aux outils de gestion des consentements, que ce soit pour en faire la liste (par ayant droit, par bénéficiaire, etc.) ou pour en vérifier la présence avant un échange de données. Pour cela, il connait et peut interroger les différents gestionnaires de consentements qui devront exposer des interfaces (API) similaires à celles du routeur MULTIPASS.

Il apporte en particulier à l’ayant droit une vision exhaustive de ses consentements (qui peuvent être répartis dans plusieurs systèmes de gestion des consentements), répondant aux attentes de transparence des exploitants agricoles. Il est également le point où un fournisseur de données peut vérifier si le consentement nécessaire à la fourniture de données existe, sans avoir besoin de connaître chez quel gestionnaire il est géré. Ces vérifications sont tracées.

Tous les acteurs autres que les ayant-droit doivent s’enregistrer sur le routeur avant de pouvoir l’utiliser, et un administrateur doit valider leur compte. Un ayant droit (ou son délégataire) n’interagit en effet pas avec le routeur. C’est le rôle du responsable d’enregistrement de permettre la saisie / modification d’un consentement. Seul l’ayant droit peut voir ses consentements une fois qu’il est authentifié. Le responsable d’enregistrement n’a pas à les voir. Cette sécurité est d’autant plus nécessaire quand ce responsable est aussi fournisseur de services (il ne doit pas voir si l’agriculteur travaille avec ses concurrents). Pour cela, soit celui-ci aura pris l’engagement contractuel en ce sens dans son contrat avec l’agriculteur, soit il sera engagé par l’adhésion à une charte soit il y sera obligé par le RGPD pour les données personnelles.

La gestion des référentiels est du ressort d’un administrateur qui s’appuie sur le routeur.

# Les services

Chaque gestionnaire de consentements enregistré dans le routeur Multipass devra exposer un WS de gestion des consentements composé d’au moins 4 services :

* Un service d’authentification
* Un service de vérification des consentements
* Un service de récupération des consentements
* Un service d’enregistrement des consentements, ce service est hors périmètre Routeur Multipass

Ce WS peut être exposé en REST ou en SOAP mais doit respecter les interfaces définies par le Routeur Multipass.

## Le service d’authentification

Le Gestionnaire de consentement devra exposer une méthode d’authentification prenant en paramètre un couple Login/Password passé dans le header de la requête http en tant que « Basic Auth » et renvoyant en sortie un jeton d’authentification.

Par défaut nous préconisons au gestionnaire d'utiliser son propre système d'authentification Oauth mais s'il considère que cette mise en place est trop lourde pour lui, il peut simplement faire le choix de reprendre les signatures des méthodes Oauth que nous avons décrites.

De même nous préconisons que la méthode renvoie un token JWT chiffré avec une clé asymétrique afin que le routeur Multipass puisse dans une future version vérifier que ce jeton ait bien été émis par le gestionnaire de consentements.

## Le service de vérification dES consentements

Le token JWT préalablement récupéré via le service d’authentification est passé dans le header de la requête.

Authorization : Bearer + Token JWT

### Exposition en REST

Le service doit être appelé en GET.

Les paramètres sont passés en tant que paramètres à la requête et sont les mêmes que pour le service CheckConsent du routeur Multipass

### Exposition en SOAP

Le service doit respecter l’interface telle que définie dans le fichier multipass.xsd

### Données en entrée

|  |  |  |
| --- | --- | --- |
| Paramètre | Commentaire | Obligatoire |
| Token | **Jeton d’authentification JWT (Dans le header de la requête)** | **Oui** |
| DatabaseNumber | **Identifiant du fournisseur de données, il correspond à un numéro de SIRET** | **Oui** |
| FarmerNumber | **Identifiant de l’ayant droit, il correspond à un numéro de SIRET** | **Oui** |
| SupplierNumber | **Identifiant du bénéficiaire du consentement, il correspond à un numéro de SIRET** | **Oui** |
| DataType | **Liste de familles de données sur lesquelles porte le consentement à vérifier** | **Oui** |
| UseCase | **Liste de catégories d’usage sur lesquelles porte le consentement à vérifier** | **Oui** |
| ConsentManagerNumber | **Identifiant du gestionnaire du consentement, il correspond à un numéro de SIRET** | **Non** |

### Données en sortie

|  |  |  |
| --- | --- | --- |
| Paramètre | Commentaire | Obligatoire |
| Result | **Résultat du service (Vrai/Faux)** | **Oui** |
| Anomaly | **Structure Anomalie présente si le résultat du service est Faux ou si présence d’une anomalie** | **Non** |
| Anomaly => ErrorCode | **Code erreur si le résultat est faux ou code anomalie** | **Oui** |
| Anomaly => ErrorDescription | **Libellé de l’erreur ou de l’anomalie** | **Oui** |
| ExistingConsent | **Vrai si le consentement existe, faux sinon. Cette donnée est présente uniquement si le résultat du service est Vrai** | **Non** |

## Le service de récupération des consentements

Le token JWT préalablement récupéré via le service d’authentification est passé dans le header de la requête.

Authorization : Bearer + Token JWT

### Exposition en REST

Le service doit être appelé en GET.

Les paramètres sont passés en tant que paramètres à la requête et sont les mêmes que pour le service GetConsent du routeur Multipass

### Exposition en SOAP

Le service doit respecter l’interface telle que définit dans le fichier multipass.xsd

### Données en entrée

|  |  |  |
| --- | --- | --- |
| Paramètre | Commentaire | Obligatoire |
| Token | **Jeton d’authentification JWT (Dans le header de la requête)** | **Oui** |
| FarmerNumber | **Identifiant de l’ayant droit, il correspond à un numéro de SIRET** | **Non** |
| SupplierNumber | **Identifiant du bénéficiaire du consentement, il correspond à un numéro de SIRET** | **Non** |
| DatabaseNumber | **Identifiant du fournisseur de données, il correspond à un numéro de SIRET** | **Non** |
| CollectorNumber | **Identifiant du responsable d’enregistrement, il correspond à un numéro de SIRET** | **Non** |
| DataType | **Famille de données sur laquelle porte le consentement à vérifier** | **Non** |
| UseCase | **Catégorie d’usage sur laquelle porte le consentement à vérifier** | **Non** |
| Active | **Si vrai, le service renvoie uniquement les consentements actifs à la date de la demande, sinon, il renvoie l’ensemble des consentements, indépendamment des dates de validité** | **Non**  **La valeur par défaut sera Vrai** |

### Données en sortie

|  |  |  |
| --- | --- | --- |
| Paramètre | Commentaire | Obligatoire |
| Result | **Résultat du service (Vrai/Faux)** | **Oui** |
| Anomaly | **Structure Anomalie présente si le résultat du service est Faux ou si présence d’une anomalie** | **Non** |
| Anomaly => ErrorCode | **Code erreur si le résultat est faux ou code anomalie** | **Oui** |
| Anomaly => ErrorDescription | **Libellé de l’erreur ou de l’anomalie** | **Oui** |
| ConsentsList | **Liste des consentements** | **Non** |

Description d’un consentement :

|  |  |  |
| --- | --- | --- |
| Paramètre | Commentaire | Obligatoire |
| FarmerNumber | **Identifiant de l’ayant droit, N° de SIRET** | **Oui** |
| SupplierNumber | **Identifiant du bénéficiaire, N° de SIRET** | **Oui** |
| DatabaseNumber | **Identifiant du fournisseur de données, N° de SIRET** | **Oui** |
| CollectorNumber | **Identifiant du responsable d’enregistrement, N° de SIRET** | **Oui** |
| DataType | **Famille de données sur laquelle porte le consentement** | **Oui** |
| UseCase | **Catégorie d’usage sur laquelle porte le consentement à vérifier** | **Non** |
| Restriction | **Zone de texte libre renseignée dans le cas où le consentement ne concerne pas toutes les données de l’ayant droit** | **Non** |
| ConsentBegin | **Date de début de validité du consentement** | **Oui** |
| ConsentEnd | **Date de fin de validité du consentement** | **Non** |
| UseCaseDescription | **Zone de texte libre expliquant l’usage du consentement** | **Non** |
| ContractFlag | **Vrai si le consentement découle d’un contrat** | **Oui** |
| ContractReference | **Référence du contrat si la donnée ContractFlag est égale à Vrai** | **Non** |
| AnonymizationFlag | **Vrai si les données doivent être anonymisées avant utilisation par le bénéficiaire** | **Oui** |
| ReversibilityFlag | **Vrai si les données doivent être détruites par le bénéficiaire à la clôture du consentement** | **Oui** |