Subscription

Manuel d'administration et de fonctionnement

MLAdvisory © 2026

Table des matières

Introduction 11
Organisation de ce manuel 11
Le menu de la console 11

Nouveautés 13
Novembre 2026 13

Prise en main de la console 15
Qui peut se connecter 15
L'organisation de l'écran 15
La barre supérieure 15
Le menu de gauche 16
L'accueil 16
Listes et formulaires 16
Le profil 17
Les paramètres 17
L'onglet Services 17
L'onglet Clé maître 17
Les mises à jour en direct 18
La langue 18
L'aide intégrée 18

Concepts clés 19
Vue d'ensemble 19
Client 19
Realm 19
Tenant 19
Utilisateur 20
Accès (utilisateur dans un tenant) 20
Rôle 20
Service 20
Fonctionnalité 21
Plan 21
Contrat 21
Les relations en tableaux 21

I **Gestion des tenants**

Clients et tenants . . . 25
L'arbre des clients et des tenants . . . 25
Rechercher et filtrer . . . 25
Les menus ··· . . . 25
Les contrats bientôt échus . . . 26
Créer ou modifier un client . . . 26
La page d'un client . . . 27
Le logo d'un client . . . 27
Supprimer un client . . . 27
Créer ou modifier un tenant . . . 28
Les identifiants de connexion . . . 28
La page d'un tenant . . . 28
Le statut d'un tenant . . . 29
La validité d'un tenant . . . 29
Suspendre, supprimer, restaurer . . . 29
Les membres d'un tenant . . . 30
Le stockage ALM . . . 30
Qui peut faire quoi . . . 31
Utilisateurs . . . 33
La liste des utilisateurs . . . 33
Filtres . . . 33
Actions . . . 33
Créer ou modifier un utilisateur . . . 33
La page d'un utilisateur . . . 35
Clients invités . . . 35
Désactiver et réactiver . . . 35
Les modifications de compte Keycloak . . . 35
Supprimer un utilisateur . . . 36
Importer des utilisateurs . . . 36
Accès aux tenants . . . 36
Rôle et plans . . . 37
Donner ou modifier un accès . . . 37
La page d'un accès . . . 37
Suspendre l'accès à un tenant . . . 37
Qui peut faire quoi . . . 38
Historique des accès . . . 39
La chronologie . . . 39
Les filtres . . . 39
Les types d'événements . . . 40
Qui a fait le changement . . . 40
La revue des accès . . . 41

Conservation . . . 41
Qui peut le voir . . . 41

Purge . . . 43
Tenants supprimés . . . 43
Vérifier un tenant . . . 43
Données orphelines . . . 44
Purger des données orphelines . . . 44
Tenants par défaut des services . . . 45
Qui peut le faire . . . 45

Console support . . . 47
Qui peut utiliser la console . . . 47
Ouvrir la console . . . 47
Choisir un tenant . . . 48
Les membres du tenant . . . 48
Demander l'accès . . . 49
Consulter en tant que . . . 49
Politique et autorisations actives . . . 49
Le journal support . . . 50
Terminer une consultation en cours . . . 50
Mises à jour en direct . . . 51

II Plateforme Reglab

Services . . . 55
La liste des services . . . 55
Créer un service . . . 55
Modifier un service . . . 55
La page d'un service . . . 55
Supprimer un service . . . 56

Plans . . . 57
La liste des plans . . . 57
Créer un plan . . . 57
La page d'un plan . . . 57
Modifier un plan . . . 58
Supprimer un plan . . . 58

Fonctionnalités . . . 59
La liste des fonctionnalités . . . 59
Créer une fonctionnalité . . . 59
La fonctionnalité base . . . 59

La page d'une fonctionnalité 60
Modifier une fonctionnalité 60
Supprimer une fonctionnalité 60

III **Configuration des services**

Collecteur 63
En-tête et bandeau de la dernière collecte 63
Publications 63
Retransmettre à la bibliothèque 64
Résumés pour le mail du jour 64
Sources 64
Collectes & planification 64
Planification 64
Comment le crawler parcourt 65
Dernières collectes 65
Flux 65
Ajouter ou modifier un flux 65
Supprimer un flux 66
Destinataires 66
Digests envoyés 66
Docling 67

Notifications 69
La liste des mails 69
Le détail d'un mail 69
Distribution 70
Renvoyer un mail 70
Les mails envoyés par subscription 71
Les rappels de fin de contrat 71

Librairie 73
Le parcours d'un document 73
Documents 73
Embeddings du chat 73
La liste des documents 74
Les actions sur un document 74
Téléversement 74
Téléverser des fichiers 75
Indexer depuis le web 75
Envoyer 75
Index de recherche 75

Référentiel 77
Cubes 77
Importer un cube 77
La liste des cubes 77

Sources . . . 77
La liste des sources . . . 77
Suivre ou modifier un cube . . . 78

IV **Supervision**

Activité Kafka DPM . . . 83
Lecture en direct . . . 83
Les totaux . . . 83
Les listeners . . . 83
Les partitions . . . 84

V **Fonctionnement**

Identités et realms . . . 87
Les realms . . . 87
Qui ouvre la console . . . 87
Ce que garde subscription, ce que garde Keycloak . . . 87
Un utilisateur : un realm et un identifiant . . . 88
Le realm d'un client . . . 88
La création et l'import . . . 88
La désactivation . . . 88
La suppression . . . 89
La dernière connexion . . . 89
Les recherches d'appartenance des services . . . 89
Paramètres . . . 90

Des plans aux droits . . . 91
Le principe . . . 91
Ce que la recherche répond . . . 91
Qui peut interroger subscription . . . 92
L'en-tête X-TENANT-ID et tenant-web-library . . . 92
Le cache des appartenances . . . 92
CIRCÉ (irrc) . . . 93
L'assistant . . . 93
L'événement UPDATE_PLAN . . . 93
Quand subscription envoie UPDATE_PLAN . . . 94
Ce que subscription ne signale pas . . . 94
Les délais en résumé . . . 95
Les noms de fonctionnalités lus . . . 95

Le cycle de vie d'un tenant . . . 97
Les statuts . . . 97
La validité . . . 98

Ce qui donne accès à un tenant . . . 98
La création . . . 98
Suspension et réactivation . . . 98
Suppression et délai d'attente . . . 98
La purge . . . 99
L'événement TENANT_DELETED . . . 99
Ce que chaque service purge . . . 99
Les messages UPDATE_PLAN . . . 100
Les e-mails aux membres . . . 100
Les relances de fin de contrat . . . 101
Les tenants de test . . . 101
Paramètres . . . 102

L'accès du support . . . 103
Vue d'ensemble . . . 103
Qui peut consulter qui . . . 103
La politique du tenant . . . 104
Les demandes d'accès . . . 104
Les autorisations . . . 105
La décision d'accès . . . 105
La session de consultation . . . 106
Comment les services reconnaissent une consultation . . . 107
Les motifs de fin . . . 107
Terminer une session . . . 108
Événements, journal et audit . . . 108
Mises à jour en direct et courriels . . . 109
Points d'accès de l'API . . . 110
Réglages de subscription . . . 110

La sécurité . . . 113
Les jetons acceptés . . . 113
Qui est qui . . . 113
Ce que l'API exige . . . 114
Appels des autres services . . . 114
Les credentials des tenants . . . 115
Les clés maîtres . . . 116
Faire tourner la clé maître . . . 116
Points d'accès publics et logos . . . 117
Vault et Consul . . . 117

Événements et appels . . . 119
Vue d'ensemble . . . 119
Les topics Kafka . . . 119

Les messages **120**
`UPDATE_PLAN` 120
`TENANT_DELETED` 120
`SEND_MAIL` 120
`NEW_PUBLICATION` et `NEW_COMMON_FILE` 120
`SUPPORT_REQUEST_CHANGED` 120
Les enveloppes **120**
Les garanties d'envoi **121**
Les mises à jour en direct de la console **122**
D'où viennent les événements 122
Le flux 122
Côté console 123
Les appels synchrones **123**

Glossaire **127**
Termes **127**

Introduction

Subscription est la console d'administration de la plateforme Reglab. C'est là que se déclarent les clients et leurs tenants, les utilisateurs et leurs accès, les services de la plateforme, leurs fonctionnalités et les plans qui les ouvrent. Le service du même nom répond aux autres services (CIRCÉ, HERMES, la librairie, l'assistant) quand ils vérifient qu'un utilisateur peut travailler dans un tenant.

Ce manuel décrit Subscription au 24 novembre 2026. Les changements récents sont listés dans les Nouveautés.

La console permet de :

- créer les clients et leurs tenants, suivre leur contrat et leur statut, jusqu'à leur suppression et leur purge ;
- gérer les utilisateurs, leur rôle et leurs plans dans chaque tenant, les importer et les désactiver ;
- déclarer les services de la plateforme, leurs fonctionnalités et les plans qui les regroupent ;
- suivre l'historique des accès et ouvrir, sur autorisation, une consultation de support dans un tenant ;
- configurer le collecteur, les notifications, les documents communs de la librairie et le référentiel ;
- superviser l'activité Kafka des traitements DPM.

> **Note**
>
> La console est réservée aux administrateurs de la plateforme : un compte portant le rôle `ROLE_ADMIN` dans un realm de la plateforme (`mla` par défaut). La **Console support** est aussi ouverte aux membres du support MLA et aux administrateurs d'un tenant. Voir Qui peut se connecter.

Organisation de ce manuel

Ce manuel s'adresse à deux publics :

- Les administrateurs de la plateforme, qui utilisent la console. La prise en main et les concepts clés viennent d'abord, puis une page par entrée du menu de gauche.
- Les développeurs et les exploitants, qui ont besoin de savoir ce qui se passe derrière les écrans. La partie **Fonctionnement** décrit les identités et les realms, la façon dont un plan devient un droit dans les autres services, le cycle de vie d'un tenant, l'accès support, la sécurité et les événements échangés.

Le glossaire reprend les termes de la console et de ce manuel.

Le menu de la console

Groupe	Entrées
Gestion des tenants	Clients et tenants, Utilisateurs, Historique des accès, Purge, Console support
Plateforme Reglab	Services, Plans, Fonctionnalités
Configuration des services	Collecteur, Notifications, Librairie, Référentiel
Supervision	Activité Kafka DPM

Un ordre de travail typique pour ouvrir la plateforme à un nouveau client :

1. Vérifiez que le service et le plan à vendre existent, avec les bonnes fonctionnalités.
2. Créez le client et son tenant, avec les dates de son contrat.
3. Créez ou importez ses utilisateurs, puis donnez à chacun un rôle et un plan par service dans le tenant.
4. Les services lisent ces droits à la connexion suivante : Des plans aux droits explique comment, et dans quel délai.

Nouveautés

Cette page liste, mois par mois et du plus récent au plus ancien, ce qui a changé dans la console Subscription et dans ce manuel. Chaque point renvoie à la section qui le décrit.

Novembre 2026

- **Première édition du manuel.** Les écrans de la console, une page par entrée du menu de gauche, et leur fonctionnement : identités et realms, droits ouverts par les plans dans les autres services, cycle de vie d'un tenant, accès support, sécurité et événements. Voir la prise en main et les concepts clés.

Prise en main de la console

Qui peut se connecter

La console subscription est réservée aux administrateurs de la plateforme : les comptes qui portent le rôle de realm `ROLE_ADMIN` dans un realm de la plateforme (`mla` par défaut). La connexion passe par la passerelle, qui envoie vers la page de connexion du realm `mla` ; Keycloak reprend la session déjà ouverte dans le navigateur quand il y en a une.

Une fois connecté, la console vérifie votre compte :

Situation	Ce qui s'affiche	Ce que vous pouvez faire
Votre compte n'a pas le rôle `ROLE_ADMIN`.	La page **Accès restreint** : « Subscription est réservé aux administrateurs MLAdvisory, et votre compte n'a pas ce rôle. »	**Se déconnecter**, ou contacter votre équipe support.
La connexion a été interrompue avant la fin (après une déconnexion, ou en quittant la page de connexion).	La page **La connexion n'a pas abouti**.	**Se reconnecter**.
La plateforme n'a pas répondu quand votre session a été vérifiée.	Une page d'erreur : « Le service est indisponible pour le moment. Réessayez dans un instant. »	Recharger la page un peu plus tard.

La page **Accès restreint** ne propose que le menu utilisateur avec **Se déconnecter**.

> **Note**
>
> La **Console support** a ses propres règles d'accès : des utilisateurs qui ne sont pas administrateurs de la plateforme peuvent l'ouvrir. Voir Console support.

Toutes les autres pages de la console, y compris le profil et les paramètres, demandent le rôle `ROLE_ADMIN`. Le service subscription applique la même règle à ses propres API : un administrateur de la plateforme est un jeton émis par un realm de la plateforme et qui porte le rôle `ROLE_ADMIN`, écrit exactement ainsi. Un rôle `ROLE_ADMIN` donné dans le realm d'un client ne suffit pas.

L'organisation de l'écran

L'écran se compose :

- à gauche, du logo et du nom **Subscription**, qui ramènent à l'accueil, puis du menu de gauche ;
- en haut, de la barre supérieure ;
- au centre, de la page ouverte ;
- en bas, d'un pied de page avec le logo, « MLA Solutions », le copyright et la version de la console.

La barre supérieure

De gauche à droite, la barre supérieure contient :

- le bouton chevron (**Afficher ou masquer le menu**) qui replie ou déplie le menu de gauche ;
- le **fil d'Ariane** : la page ouverte (**Accueil** sur l'accueil), puis l'action en cours (**Nouveau**, **Détail**, **Modifier**…). Ses éléments précédents sont des liens ;
- le menu d'aide (point d'interrogation), avec **Documentation en ligne** (voir L'aide intégrée) ;

- le choix de la langue, qui affiche le code de la langue en cours (`FR` ou `EN`) (voir La langue) ;
- le voyant des mises à jour en direct ;
- la photo de votre compte, ou une icône, et votre nom (à défaut votre e-mail). Ils ouvrent le menu **Profil**, **Paramètres** et **Se déconnecter**.

Le menu de gauche

Le menu est organisé en quatre groupes :

Groupe	Entrées
Gestion des tenants	**Clients et tenants**, **Utilisateurs**, **Historique des accès**, **Console support**, **Purge**
Plateforme Reglab	**Services**, **Plans**, **Fonctionnalités**
Configuration des services	**Collecteur**, **Notifications**, **Librairie**, **Référentiel**
Supervision	**Activité Kafka DPM**

Chaque entrée a sa page dans ce mode d'emploi :

- **Gestion des tenants** : Clients et tenants, Utilisateurs, Historique des accès, Console support, Purge ;
- **Plateforme Reglab** : Services, Plans, Fonctionnalités ;
- **Configuration des services** : Collecteur, Notifications, Librairie, Référentiel ;
- **Supervision** : Activité Kafka DPM.

L'accueil

L'accueil s'intitule **Gestion des contrats et souscriptions** : « Tenants, utilisateurs, plans et droits d'accès des clients de la plateforme. »

L'en-tête propose deux raccourcis :

- **Nouvel utilisateur** ouvre le formulaire de création d'un utilisateur (voir Utilisateurs) ;
- **Nouveau tenant** ouvre le formulaire de création d'un tenant (voir Clients et tenants).

La page présente ensuite :

- la partie **Gestion des tenants**, avec quatre tuiles : **Tenants**, **Utilisateurs**, **Services** et **Plans**. Chaque tuile affiche le nombre d'éléments et ouvre la liste correspondante. Le nombre de tenants ne compte ni les tenants supprimés ni les tenants temporaires (les tenants de test créés avec une durée de vie). Une tuile reste sans nombre quand sa liste n'a pas pu être lue ;
- la partie **Crawler** : l'état de la dernière collecte, le nombre de **Publications**, les **Nouvelles au dernier passage**, le **Prochain passage** (ou **Désactivé**) et la date de la dernière collecte. **Aller au crawler** ouvre la page du Collecteur ;
- la partie **Librairie**, avec la tuile **Documents communs** et leur nombre (voir Librairie) ;
- la partie **Référentiel**, avec la tuile **Cubes** (voir Référentiel).

Listes et formulaires

Les pages de liste de la console fonctionnent toutes de la même façon :

- la zone **Recherche globale…** filtre la liste au fil de la saisie ;
- les en-têtes de colonne marqués d'une flèche trient la liste ;
- la pagination, en haut de la liste, indique « *x* à *y* sur *n* éléments » et laisse choisir le nombre de lignes par page ;
- le menu ⋮ d'une ligne propose **Voir**, **Modifier** et **Supprimer** ;
- les cases à cocher sélectionnent des lignes ; le bouton **Supprimer** de la barre d'outils supprime la sélection, après confirmation.

Une suppression groupée envoie une demande par élément. Une fois toutes les réponses reçues, un message indique « *n* supprimé(s), *m* refusé(s) » et nomme les éléments refusés (« Refusé(s) : … »). La liste est ensuite relue.

Dans les formulaires :

- un champ obligatoire laissé vide affiche « Ce champ est obligatoire. » ;
- **Enregistrer** reste grisé tant que le formulaire n'est pas valide ;
- **Retour** quitte le formulaire sans enregistrer ;
- après un enregistrement réussi, un message de confirmation s'affiche en bas de l'écran ;
- quitter un formulaire modifié et non enregistré ouvre la fenêtre **Modifications non enregistrées** : « Vos modifications ne sont pas enregistrées. Quitter cette page et les perdre ? », avec **Quitter sans enregistrer** et **Rester sur la page**. Recharger ou fermer l'onglet déclenche l'avertissement du navigateur.

Les champs **Description** utilisent un éditeur de texte enrichi : titre, gras, italique, souligné, barré, listes, lien, citation, bloc de code et **Supprimer la mise en forme**.

Quand le serveur refuse une action, la page affiche un bandeau d'erreur qui en donne la raison.

Le profil

Profil, dans le menu sous votre nom, ouvre la page de votre compte.

- L'en-tête donne votre nom, votre identifiant et la mention « compte géré par l'annuaire de connexion » : le compte se modifie dans Keycloak, pas dans la console. Le bouton **Se déconnecter** termine la session.
- La carte **Identité** affiche **Prénom**, **Nom**, **E-mail** et **Rôles** (les rôles de votre compte, `ROLE_ADMIN` compris).
- La carte **Préférences** propose la **Langue de l'interface** (Français ou English) : « Appliquée tout de suite et retenue pour vos prochaines sessions. »

Les paramètres

Paramètres, dans le menu sous votre nom, ouvre une page à trois onglets : **Paramètres utilisateur**, **Services** et **Clé maître**.

L'onglet Services

L'onglet **Services** montre si les services de la plateforme répondent, et quelle version tourne.

- Le résumé indique « Les *n* services répondent » ou « *m* services sur *n* injoignables ». Chaque carte lit le `/actuator/info` du service via la passerelle. **Actualiser** relit toutes les cartes ; l'heure de la dernière lecture est affichée (« Lu à … »).
- Les cartes sont regroupées :
  - **Plateforme** : **Subscription console** (cette application), **Passerelle**, **Abonnements**, **Admin** ;
  - **Reporting réglementaire** : **IRRC**, **DPM**, **Moteur de reporting** ;
  - **Bibliothèque** : **Bibliothèque**, **Crawler** ;
  - **Services partagés** : **Notification**, **Référentiel**.
- Chaque carte affiche l'état **Répond**, **Injoignable** ou **Lecture**, puis la version, la date de construction (**Construit**) et le **Commit**, avec sa branche et un bouton pour copier l'identifiant du commit. Une carte **Injoignable** explique : « Pas de réponse via la passerelle : le service est arrêté, redémarre ou n'est pas déployé ici. »

L'onglet Clé maître

L'onglet **Clé maître** sert pendant une rotation de la clé maître de subscription, la clé qui chiffre les credentials des tenants. Il est en lecture seule : les clés elles-mêmes ne sont jamais affichées.

- Un bandeau résume la situation, par exemple « Tous les credentials de tenant sont sur la clé … », « La clé … ne chiffre plus rien » ou « La clé … chiffre encore *n* valeur(s) », avec la conduite à tenir.
- Le tableau liste chaque **Clé**, son **État** (**Courante**, **Encore utilisée**, **Peut être retirée**, **Absente de Vault**, ou **Format antérieur à v2 (clé 0)**) et le nombre de **Valeurs** qu'elle chiffre.
- **Actualiser** relit l'état.

La rotation et le retrait d'une clé se font dans Vault, en suivant la procédure `rotation-cle-maitre-subscription` du dépôt documentation.

Les mises à jour en direct

Tant que vous êtes connecté, la console garde ouvert un flux d'événements avec le service subscription. Les pages du Collecteur, de la Librairie et de la Console support se mettent à jour d'elles-mêmes quand un événement les concerne.

Un voyant, dans la barre supérieure, indique l'état du flux ; son info-bulle le dit en toutes lettres :

Voyant	Info-bulle	Signification
Vert	« Mises à jour en direct : connecté »	Le flux est ouvert.
Orange	« Mises à jour en direct : reconnexion en cours »	Le flux est coupé depuis plus de 5 secondes.
Rouge	« Mises à jour en direct interrompues — les données se rafraîchissent toutes les 30 s »	Le flux est coupé depuis plus d'une minute.

Une coupure plus courte ne change pas le voyant. Il n'apparaît pas tant que la première connexion n'est pas établie.

La langue

La console existe en français et en anglais. Changez de langue :

- avec le choix de la langue de la barre supérieure (**Français** ou **English**, la langue en cours est cochée) ;
- ou avec **Langue de l'interface** sur la page Profil.

Le changement s'applique tout de suite et subscription le retient pour vos prochaines sessions. Sans choix enregistré ni langue portée par votre compte, la console s'affiche en anglais.

L'aide intégrée

Documentation en ligne, dans le menu d'aide (point d'interrogation), ouvre à droite un panneau avec un guide court de la console, dans la langue en cours (en anglais à défaut). Quand le guide a plusieurs sections, un **Sommaire** permet d'y accéder. Si le guide ne peut pas être chargé, le panneau l'indique : « La documentation n'a pas pu être chargée. »

Concepts clés

Cette page présente les notions que la console manipule et la façon dont elles s'articulent. Les écrans qui les gèrent ont chacun leur page : Clients et tenants, Utilisateurs, Services, Plans et Fonctionnalités.

Vue d'ensemble

En une phrase : un **utilisateur** d'un **client** reçoit, dans un **tenant** de ce client, un **rôle** et des **plans** ; chaque plan appartient à un **service** et ouvre une sélection des **fonctionnalités** de ce service. Les services de la plateforme demandent à subscription, à chaque accès, ce que l'utilisateur détient dans le tenant (voir Des plans aux droits).

Deux familles d'objets se distinguent :

- le **catalogue**, commun à tous les clients : les services, leurs fonctionnalités et les plans (menu **Plateforme Reglab**) ;
- les **abonnés** : les clients, leurs tenants, leurs utilisateurs et les accès de ces utilisateurs aux tenants (menu **Gestion des tenants**).

Client

Un **client** est une organisation cliente de la plateforme. Il porte un nom (unique), une description, et, une fois qu'un administrateur les a fixés :

- un **Sous-domaine** : le label sur lequel répond le front du client ;
- un **Domaine (realm)** : le realm Keycloak où se connectent les utilisateurs du client (voir Realm).

Un client regroupe ses tenants et ses utilisateurs.

Realm

Un **realm** est un espace de comptes de Keycloak. Les realms acceptés par la plateforme sont fixés par la configuration ; parmi eux, les **realms de la plateforme** (`mla` et `shared` par défaut) sont ceux des administrateurs de la plateforme et des comptes de service.

Un nom d'utilisateur n'est unique qu'à l'intérieur de son realm : pour subscription, un utilisateur est le couple (realm, identifiant). Les services qui interrogent subscription envoient donc le realm qui a émis le jeton de l'utilisateur ; sans realm, subscription répond pour le realm par défaut de la plateforme.

Tenant

Un **tenant** est un environnement RegLab isolé appartenant à un client. Un client peut avoir plusieurs tenants. Un tenant porte un nom (unique), une description, et un identifiant public (`tenantId`) : c'est lui que les services reçoivent pour savoir dans quel tenant travaille un utilisateur.

Un tenant a un **statut** :

Statut	Signification
`ACTIVE` (actif)	Le tenant donne accès, dans sa période de validité.
`SUSPENDED` (suspendu)	Suspendu par un administrateur de la plateforme : tout est gardé, aucun accès, jusqu'à la réactivation.
`PENDING_DELETION` (suppression programmée)	Aucun accès ; restaurable jusqu'à sa date de purge, puis purgé.

Statut	Signification
`DELETED` (supprimé)	Purgé : les accès sont retirés, la ligne du tenant est gardée et son nom reste réservé. Ce statut est définitif.

Seul un tenant actif, dans sa période de validité, donne accès aux services. Le cycle de vie est décrit dans Clients et tenants et Le cycle de vie d'un tenant.

Utilisateur

Un **utilisateur** est un compte de la plateforme : un identifiant dans un realm, un e-mail, un prénom et un nom. Il appartient à un client, et un seul.

Un utilisateur peut rejoindre :

- les tenants de son propre client ;
- les tenants des **clients invités**, si un administrateur l'a rendu **Multi-client** et l'a déclaré invité de ces clients (pour un consultant ou un auditeur, par exemple).

Un utilisateur peut être désactivé : il n'a alors plus accès à aucun tenant. Voir Utilisateurs.

Accès (utilisateur dans un tenant)

Un **accès** est la place d'un utilisateur dans un tenant. Il se compose :

- d'un **rôle** dans le tenant, un seul, quels que soient ses plans (voir Rôle) ;
- d'une liste de **plans** : au plus un plan par service. Sans plan d'un service, ce service reste fermé à l'utilisateur dans ce tenant.

Un membre détient au moins un plan : retirer son dernier plan le retire du tenant.

Un accès peut être **suspendu** : le rôle et les plans restent, mais aucun service n'ouvre plus le tenant à l'utilisateur jusqu'à la réactivation.

Les accès se donnent depuis Clients et tenants ou depuis la liste des accès ; chaque changement est enregistré dans l'Historique des accès. Quand ses droits changent, l'utilisateur reçoit un e-mail, sauf dans un tenant temporaire.

Rôle

Deux rôles ne doivent pas être confondus :

Rôle	Où il se donne	Ce qu'il permet
Administrateur de la plateforme (`ROLE_ADMIN`)	Dans Keycloak, sur un realm de la plateforme	Utiliser la console subscription et ses API.
Rôle dans un tenant : **Membre** ou **Administrateur**	Dans l'accès de l'utilisateur au tenant	**Membre** : « Utilise les fonctionnalités de ses plans. » **Administrateur** : « Gère en plus les accès des autres utilisateurs au tenant. »

Le rôle dans un tenant ne vaut que pour ce tenant : un utilisateur peut être administrateur d'un tenant et membre d'un autre. Les services le lisent dans la réponse de subscription ; CIRCÉ, par exemple, fait d'un administrateur du tenant l'administrateur de ce tenant dans CIRCÉ.

Service

Un **service** est un produit RegLab (par exemple `irrc`, le service de CIRCÉ). Il porte un nom (unique) et une description. Chaque service a ses fonctionnalités et ses plans.

Le nom du service est celui que les applications demandent à subscription : CIRCÉ demande les plans du service `irrc`, l'assistant ceux des services `irrc`, `stress` et `library`. Voir Services.

Fonctionnalité

Une **fonctionnalité** est une capacité d'un service, désignée par un nom (unique dans son service) et décrite par une description. Les applications ouvrent un écran ou une action quand le plan de l'utilisateur porte la fonctionnalité qui correspond : `anacredit` ou `pilar3` dans CIRCÉ, `assistant` pour l'assistant, par exemple.

Chaque service a une fonctionnalité `base`, créée avec lui : la détenir fait de l'utilisateur un membre du service. Voir Fonctionnalités.

Plan

Un **plan** est une offre sur un service : il appartient à un service et ouvre une sélection de ses fonctionnalités. Un plan porte toujours la fonctionnalité `base` de son service, et deux plans d'un même service ne peuvent pas ouvrir exactement les mêmes fonctionnalités. Le nom d'un plan est unique sur toute la plateforme. Voir Plans.

Contrat

Le **contrat** d'un tenant est sa période de validité : une date de début (**Valide à partir du**) et une date de fin d'accès (**Fin d'accès le (exclu)**), chacune facultative. En dehors de cette période, les utilisateurs du tenant n'ont plus accès. Un tenant sans date de fin est sans échéance : l'accès ne s'arrête jamais de lui-même.

Avant la fin d'un contrat, subscription envoie des relances par e-mail à MLA et aux administrateurs du tenant. La page Clients et tenants montre les contrats bientôt échus et les relances envoyées.

Les relations en tableaux

Client et tenant

Un client	Un tenant
a zéro, un ou plusieurs tenants	appartient à un client

Client et utilisateur

Un client	Un utilisateur
a zéro, un ou plusieurs utilisateurs	appartient à un seul client (son client d'origine)
peut recevoir des utilisateurs invités d'autres clients	peut être invité d'autres clients s'il est **Multi-client**

Client et realm

Un client	Un realm
a au plus un realm, où se connectent tous ses utilisateurs	peut servir plusieurs clients

Utilisateur et tenant (l'accès)

Un utilisateur, dans un tenant	Contrainte
a un seul rôle	**Membre** ou **Administrateur**
détient au moins un plan	au plus un plan par service
peut être suspendu	rôle et plans gardés, aucun accès

Service, plan et fonctionnalité

Objet	Relation	Contrainte
Service ⟶ plan	un service a zéro, un ou plusieurs plans ; un plan appartient à un seul service	un service qui a des plans ne peut pas être supprimé
Service ⟶ fonctionnalité	un service a ses fonctionnalités, dont `base` ; une fonctionnalité appartient à un service	le nom d'une fonctionnalité est unique dans son service
Plan ↔ fonctionnalité	un plan ouvre plusieurs fonctionnalités ; une fonctionnalité peut être ouverte par plusieurs plans	un plan n'ouvre que des fonctionnalités de son service, `base` comprise ; deux plans d'un service n'ouvrent pas les mêmes
Plan ↔ accès	un plan peut être détenu par des utilisateurs dans plusieurs tenants	un plan détenu ne peut pas être supprimé

I

Gestion des tenants

Clients et tenants

Les clients et leurs tenants se trouvent dans le menu **Gestion des tenants › Clients et tenants**.

Un **client** est une organisation cliente. Un **tenant** est un environnement RegLab isolé qui appartient à un client ; un client peut en avoir plusieurs. Les utilisateurs reçoivent un rôle et des plans sur un tenant : voir Utilisateurs. Le cycle de vie d'un tenant (statuts, validité, suppression, purge) est expliqué en détail dans Le cycle de vie d'un tenant.

L'arbre des clients et des tenants

La page s'intitule **Tenants**. Son en-tête propose **Nouveau client** et **Nouveau tenant**.

Sous l'en-tête, le panneau **Contrats bientôt échus** et la ligne des tenants sans date de fin (voir Les contrats bientôt échus), puis l'arbre. Il a trois niveaux :

- un **client** par ligne de premier niveau, toujours déplié, avec son logo s'il en a un, son nom (lien vers la page du client), le nombre de ses tenants, son **Sous-domaine** et son **Domaine (realm)** s'ils sont renseignés. Un client sans tenant apparaît aussi. Les tenants sans client sont regroupés sous une dernière ligne **Sans client** ;
- sous chaque client, ses **tenants**, par ordre alphabétique : nom (lien vers la page du tenant), nombre de membres, pastilles d'état (voir Le statut d'un tenant), description, date de **Création** et de **Suppression** ;
- sous un tenant, ses **membres** : identifiant (lien vers la page de l'utilisateur), pastille **Suspendu** si son accès au tenant est suspendu, son rôle (**Membre** ou **Administrateur**) et ses plans (liens vers chaque plan).

Les membres d'un tenant sont lus quand vous le dépliez. Un tenant en affiche 20 au plus ; au-delà, une dernière ligne **Voir les N accès** ouvre la liste des accès filtrée sur ce tenant (voir Accès aux tenants).

La colonne **Suppression** donne la date de purge d'un tenant supprimé. Pour un tenant dont la suppression est programmée, elle affiche en gris, précédée d'une horloge, la date prévue de la purge (info-bulle « Purge prévue à cette date, restaurable d'ici là »).

La pagination porte sur les clients (« 1 à 10 sur N clients »).

Rechercher et filtrer

La barre au-dessus de l'arbre propose :

- **Recherche globale…** : cherche dans les noms, les descriptions et les noms de plans. Pendant une recherche, tous les tenants sont dépliés et montrent tous leurs membres, pour qu'aucune correspondance ne reste cachée ;
- **Choisir un client** : n'affiche que les clients choisis ;
- **Afficher les tenants supprimés (N)** : les tenants supprimés (purgés) sont masqués par défaut ; cochée, la case les affiche, grisés ;
- **Sans date de fin (N)** : n'affiche que les tenants actifs sans date de fin de validité, et les clients qui en ont. Chacun porte alors un bouton **Fixer une date de fin**, qui ouvre la fenêtre de validité.

Les menus ···

Le menu ··· d'un **client** propose **Voir**, **Modifier** et **Supprimer**. **Supprimer** est grisé tant que le client a des tenants (supprimés compris) ou des utilisateurs ; son info-bulle le dit : « Suppression impossible tant que le client a N tenant(s) et N utilisateur(s). »

Le menu ··· d'un **tenant** propose, selon son statut :

Entrée	Proposée quand le tenant est
Voir	toujours
Stockage ALM	toujours

Entrée	Proposée quand le tenant est
Modifier	non supprimé
Ajouter un utilisateur	actif ou suspendu
Suspendre	actif
Réactiver	suspendu
Validité	non supprimé
Programmer la suppression	actif ou suspendu
Restaurer	en attente de suppression

Ajouter un utilisateur ouvre le formulaire **Donner un accès** avec ce tenant et son client déjà choisis (voir Donner ou modifier un accès).

Le menu ··· d'un **membre** propose **Voir**, **Modifier**, **Suspendre l'accès** (ou **Réactiver l'accès**) et **Supprimer**. **Supprimer** retire l'utilisateur du tenant après la confirmation « Retirer X de T ? Son rôle et ses plans sur ce tenant sont supprimés, et X recevra un e-mail pour l'en informer. »

Les contrats bientôt échus

Le panneau **Contrats bientôt échus** liste les tenants actifs dont l'accès prend fin dans les 60 jours, et ceux échus depuis moins de 30 jours, le plus proche en premier. Il n'apparaît que s'il a au moins une ligne.

Chaque ligne affiche :

- le temps restant : « dans 12 jours », « dans 1 jour », « dernier jour », puis « échu aujourd'hui », « échu hier », « échu depuis N jours ». La couleur passe à l'orange sous 30 jours, au rouge sous 7 jours, au gris une fois échu ;
- le nom du tenant (lien vers sa page) et son client ;
- le dernier jour d'accès, « jusqu'au JJ/MM/AAAA inclus ». L'info-bulle donne l'instant exact où l'accès s'arrête (« L'accès s'arrête le …, exclu ») ;
- une jauge de la part restante des 60 jours ;
- les relances déjà envoyées pour cette date de fin : **J-60**, **J-30**, **J-7**, **échu**. L'info-bulle donne la date d'envoi et le nombre d'adresses ; une relance pâle, bordée de pointillés et marquée d'un tiret au lieu de l'enveloppe, a été sautée : une relance plus urgente est partie à sa place. Sans relance : « Aucune relance envoyée » ;
- le bouton **Prolonger**, qui ouvre la fenêtre de validité du tenant (voir La validité d'un tenant).

Sous le panneau, une ligne compte les tenants actifs sans date de fin : « N tenant(s) actif(s) sans date de fin : aucune relance ne préviendra de leur échéance. » **Les afficher** coche le filtre **Sans date de fin** ; **Afficher tous les tenants** le décoche.

Les relances elles-mêmes (qui les reçoit, et quand) sont décrites dans Les relances de fin de contrat.

Créer ou modifier un client

Nouveau client (en-tête de l'arbre) ouvre le formulaire **Nouveau client**. **Modifier** (menu ··· ou page du client) ouvre **Modifier le client**.

Champ	Obligatoire	Contrôles
Nom du client	oui	Unique : « Un client porte déjà ce nom ou ce sous-domaine. »
Sous-domaine	non	Un label en minuscules (lettres, chiffres, tirets), 63 caractères au plus, sans tiret au début ni à la fin. Ce qui est tapé passe en minuscules. Unique entre clients : « Un autre client utilise déjà ce sous-domaine. »

Champ	Obligatoire	Contrôles
Domaine (realm)	non	**Aucun realm** ou l'un des realms acceptés par la plate-forme.
Description	non	Texte mis en forme.
Logo	non	Seulement une fois le client créé (« Un logo pourra être ajouté une fois le client créé. »).

Tant que le sous-domaine est valide, le formulaire affiche l'**Adresse du front :** sur laquelle répond le front du client (par exemple `https://<sous-domaine>.irrc-int.…` sur l'environnement d'intégration). Sinon, il rappelle l'aide : « Facultatif : un label en minuscules (lettres, chiffres, tirets) sur lequel répond le front du client. »

Le **Domaine (realm)** est le realm Keycloak où se connectent les utilisateurs du client ; tous ses utilisateurs doivent déjà y être. Le serveur refuse un realm alors que des utilisateurs du client sont dans un autre : « Ce realm ne peut pas être choisi : des utilisateurs du client sont dans un autre realm. Déplacez-les d'abord. » Voir Le realm d'un client.

Enregistrer crée le client et ouvre sa page, ou enregistre la modification et revient à la page précédente. **Retour** quitte sans enregistrer ; si des valeurs ont été saisies, la fenêtre **Modifications non enregistrées** demande **Quitter sans enregistrer** ou **Rester sur la page**.

La page d'un client

La page d'un client s'ouvre depuis son nom dans l'arbre ou depuis **Voir**. Son titre est le nom du client, précédé de son logo. Les boutons sont **Retour**, **Nouveau tenant pour ce client** (le formulaire de tenant s'ouvre avec ce client choisi) et **Modifier**.

La page affiche :

- **Nom du client** et **Description** ;
- **Sous-domaine** : l'adresse du front du client, en lien, ou « Pas de sous-domaine » ;
- **Domaine (realm)** : le realm, ou « Aucun realm » ;
- **Logo** : le logo et ses boutons (voir ci-dessous) ;
- **Tenants** : le nom (lien vers le tenant) et la description de chacun, colonnes triables ; ou « Ce client n'a aucun tenant. » ;
- **Utilisateurs** : **Identifiant** (lien vers l'utilisateur), **Nom**, **Prénom**, **E-mail** ; ou « Ce client n'a encore aucun utilisateur. »

Le logo d'un client

Le logo se gère depuis la page du client ou son formulaire de modification :

1. **Choisir un logo** (ou **Remplacer** s'il y en a un) ouvre le choix de fichier : PNG, JPEG ou WebP, 512 Ko au plus. Un autre type ou un fichier trop lourd est refusé tout de suite : « Le logo doit être une image PNG, JPEG ou WebP. », « Le logo pèse N Ko, plus que les 512 Ko permis. »
2. Un aperçu s'affiche. **Envoyer** l'enregistre ; **Annuler** l'abandonne.
3. **Supprimer** retire le logo, après la confirmation « Supprimer le logo de X ? »

Le logo apparaît dans l'arbre des tenants et sur la page du client. Les applications le montrent aussi : les services et les membres d'un tenant lisent le logo du client de ce tenant, et la page de connexion d'un front qui répond sur le sous-domaine du client l'affiche avant la connexion.

Supprimer un client

Supprimer (menu ··· du client) demande « Supprimer le client X ? ». Le serveur refuse tant que le client a des tenants, supprimés compris, ou des utilisateurs. L'entrée est d'ailleurs grisée dans ce cas.

Créer ou modifier un tenant

Nouveau tenant (en-tête de l'arbre) ou **Nouveau tenant pour ce client** (page d'un client) ouvre le formulaire **Nouveau tenant**, « Un environnement RegLab isolé pour un client. ». **Modifier** ouvre **Modifier le tenant**, « Son nom, son client et sa description. »

Le formulaire a trois sections.

Identité : « Le nom sous lequel utilisateurs et services connaissent le tenant, et le client auquel il appartient. »

- **Nom** : obligatoire et unique, y compris parmi les tenants supprimés : « Un tenant porte déjà ce nom. »
- **Client** : obligatoire (**Choisir un client**).

Identifiants de connexion, à la création seulement : voir Les identifiants de connexion.

Description : texte mis en forme, « Affichée dans l'arbre des tenants. »

Enregistrer crée le tenant et ouvre sa page, ou enregistre la modification et revient à l'arbre. **Retour** revient à l'arbre ; des valeurs non enregistrées font apparaître la fenêtre **Modifications non enregistrées**.

Règles de la modification :

- Un tenant qui a des membres ne peut pas changer de client : le champ **Client** est verrouillé, avec l'aide « Ce tenant a N membre(s) : retirez-les avant de le rattacher à un autre client. » Le serveur, de son côté, refuse de rattacher un tenant à un client dont certains membres ne font pas partie et ne sont pas invités.
- Un tenant supprimé ne peut plus être modifié : le formulaire est grisé, avec le message « Ce tenant est supprimé : il ne peut plus être modifié. »

Un nouveau tenant est actif, sans limite de validité.

Les identifiants de connexion

La section **Identifiants de connexion** n'apparaît qu'à la création : « Choisis à la création : ce formulaire ne les modifie plus ensuite. »

- **Type** : **Mot de passe**, le seul type proposé.
- **Mot de passe** : rempli par un mot de passe généré par le serveur. **Générer** en tire un autre. Vous pouvez aussi en taper un ; le serveur refuse un mot de passe vide ou trop court.
- L'aide le rappelle : « Affiché maintenant seulement : copiez-le avant d'enregistrer. »

> ⚠ Avertissement
>
> Ce mot de passe est le secret de chiffrement du tenant. Aucun écran ne le montre plus après l'enregistrement, et aucun ne permet de le changer. Il est stocké chiffré ; seuls les services autorisés le lisent (voir Les identifiants d'un tenant).

La page d'un tenant

La page d'un tenant s'ouvre depuis son nom dans l'arbre, depuis **Voir** ou depuis le panneau des contrats.

L'en-tête affiche le lien **Tous les tenants**, le nom du tenant, la pastille **Client X** (lien vers le client) et les pastilles d'état. Les boutons sont, selon le statut : **Suspendre**, **Réactiver**, **Validité**, **Programmer la suppression**, **Restaurer** (voir Suspendre, supprimer, restaurer), puis toujours **Stockage ALM** et **Historique des accès** (l'historique filtré sur ce tenant, voir Historique des accès), et **Modifier** tant que le tenant n'est pas supprimé.

La carte **Définition** donne :

- **Nom du tenant** ;
- **Référence** : l'identifiant public du tenant, celui que connaissent les services ;
- **Nom du client** (lien) et **Description** ;
- **Valide à partir du** et **Fin d'accès le (exclu)**, ou « Sans limite » ;
- **Purge le**, pour un tenant dont la suppression est programmée ;
- **Supprimé le**, pour un tenant supprimé.

La carte **Contrat** donne :

- la date de fin, « jusqu'au JJ/MM/AAAA inclus », et le temps restant ; ou « Pas de date de fin », avec le rappel « Sans échéance : l'accès ne s'arrête jamais de lui-même et aucune relance de fin de contrat n'est envoyée. » ;
- **Relances envoyées** : chaque relance pour la date de fin actuelle, avec sa date et son nombre de destinataires, ou « sautée » ; ou « Aucune relance envoyée » ;
- **Dates de fin précédentes** : les relances envoyées pour les dates de fin qu'avait le tenant avant d'être prolongé.

La carte **Accès** liste les membres du tenant : voir Les membres d'un tenant.

Le statut d'un tenant

Un tenant est dans l'un de ces quatre statuts. Seul un tenant actif, dans sa période de validité, donne accès à ses utilisateurs.

Statut	Pastille	Info-bulle
Actif	aucune	
Suspendu	**Suspendu** (orange)	« Suspendu depuis le … : ses utilisateurs n'ont plus accès. »
En attente de suppression	**Suppression le … (N jours restants)**, puis **Suppression le … (dernier jour)** (rouge)	« Aucun accès ; restaurable jusqu'au …, puis purgé. »
Supprimé	**Supprimé** (rouge)	« Purgé le … : lecture seule. »

Un tenant actif ou suspendu porte aussi, selon sa validité :

- **Expiré** : « Validité terminée le … : ses utilisateurs n'ont plus accès. » ;
- **Pas encore valide** : « Validité à partir du … : aucun accès avant. »

La validité d'un tenant

Validité (menu ···, page du tenant), **Prolonger** (panneau des contrats) et **Fixer une date de fin** (filtre **Sans date de fin**) ouvrent la fenêtre **Validité de X** : « Ses utilisateurs n'ont accès que dans cette période. Laissez une date vide pour ne pas fixer de limite. »

- **Valide à partir du** : date et heure, ou vide (« Sans limite »).
- **Fin d'accès le (exclu)** : date et heure, ou vide. L'accès s'arrête à cet instant ; l'aide nomme le dernier jour d'accès qu'il donne : « L'accès s'arrête à cet instant : le dernier jour d'accès est le JJ/MM/AAAA, inclus. »

Si la fin ne vient pas après le début, le message « La fin doit venir après le début. » s'affiche et **Enregistrer** reste grisé. Après l'enregistrement : « Validité du tenant X mise à jour ». Le panneau des contrats et le compteur des tenants sans date de fin se mettent à jour.

La validité se change à tout statut sauf supprimé. Une nouvelle date de fin repart de zéro pour les relances : celles déjà envoyées restent sous **Dates de fin précédentes**.

Suspendre, supprimer, restaurer

Chaque action demande une confirmation qui nomme le tenant, puis affiche un message.

Action	Confirmation	Effet
Suspendre	« Suspendre le tenant X ? Ses utilisateurs perdent l'accès immédiatement. Tout est conservé jusqu'à sa réactivation. »	Statut suspendu. Les membres reçoivent un e-mail.
Réactiver	« Réactiver le tenant X ? Ses utilisateurs retrouvent l'accès immédiatement. »	Statut actif. Les membres reçoivent un e-mail, sauf ceux

Action	Confirmation	Effet
		dont l'accès personnel est suspendu.
Programmer la suppression	« Programmer la suppression du tenant X ? L'accès s'arrête dès maintenant pour tous ses utilisateurs. Ses données sont purgées au bout de 60 jours ; vous pouvez le restaurer jusqu'au … »	En attente de suppression, avec la date de purge. Les membres reçoivent un e-mail.
Restaurer	« Restaurer le tenant X ? Il retrouve le statut qu'il avait avant la programmation de sa suppression. »	Actif, ou suspendu s'il l'était avant. Les membres reçoivent un e-mail.

La suppression se programme que le tenant ait des membres ou non. Pendant les 60 jours d'attente, tout est conservé et le tenant reste restaurable. Ensuite, la purge retire ses membres, le marque supprimé et demande à chaque service d'effacer ses données : voir La purge. La page Purge permet de vérifier ce que chaque service garde encore.

> **i** Note
>
> Un tenant suspendu garde l'entrée **Ajouter un utilisateur**, mais le serveur n'y donne ni ne modifie aucun accès : « Ce tenant n'est pas actif : aucun accès ne peut y être donné ni modifié. » Retirer un utilisateur reste possible.

Les membres d'un tenant

La carte **Accès** de la page d'un tenant liste une ligne par utilisateur qui y a un rôle ou des plans :

- **Utilisateur** : l'identifiant (lien vers l'utilisateur), avec la pastille **Désactivé** si son compte est désactivé, et **Suspendu** si son accès à ce tenant est suspendu ;
- **Rôle** : **Membre** ou **Administrateur** ;
- **Plans** : un lien par plan.

Voir dans la liste des accès ouvre la liste des accès filtrée sur ce tenant ; **Donner un accès** ouvre le formulaire d'accès avec ce tenant choisi. Les rôles, les plans et la suspension d'un accès sont décrits dans Accès aux tenants.

Le stockage ALM

Stockage ALM (menu ··· ou page du tenant) ouvre la page **Stockage ALM** : « Ce qu'ALM garde dans S3 pour ce tenant, par nature. Ouvrez une nature pour voir ses fichiers ; ceux qui ne servent plus sont signalés. » La page ne fait que lire : elle ne supprime rien.

Le haut de la page donne la taille totale « dans N fichiers sous » le préfixe du tenant, le nombre de fichiers **inutilisés** et d'**orphelins**, et une barre de la taille par nature. Le tableau a une ligne par nature : **Nature**, **Fichiers**, **Taille**, **Plus ancien**, **Inutilisés**.

Nature	Contenu
Entrées de run	Ce que le run a reçu, gardé pour le rejouer
Résultats de run	La réponse du moteur, supprimée une fois enregistrée : devrait être quasi vide
Bases de flux	Un fichier DuckDB par run, gardé aussi longtemps que le run
Parquet des notebooks	Construit à la demande pour les notebooks, gardé 3 mois

Nature	Contenu
Pistes d'audit	Les pistes d'audit du reporting
Fichiers importés	Les fichiers importés dans ALM
Autres	Tout le reste sous le préfixe du tenant

Ouvrir une nature (**Voir les fichiers**) liste ses fichiers : **Clé**, **Statut**, **Taille**, **Modifié le**, **Run**. Les filtres sont **Tous**, **Inutilisés**, **Utilisés**, **Restants** et **Orphelins**, chacun avec son nombre. Les statuts :

- **Utilisé** : quelque chose dans ALM pointe dessus ;
- **Restant** : son run existe mais ne devrait plus l'avoir ;
- **Orphelin** : rien dans ALM ne pointe dessus.

Si ALM ne répond pas, la page affiche « Le stockage d'ALM n'a pas pu être lu » avec la raison (ALM sans cette liste, refus, ou pas de réponse) et **Réessayer**.

Qui peut faire quoi

Toute la console est réservée aux administrateurs de la plateforme (rôle `ROLE_ADMIN` dans un realm de la plateforme) : un autre compte voit la page **Accès restreint**. Le serveur applique la même règle à chaque appel de ces écrans. Voir Les droits et Qui ouvre la console.

Le serveur refuse en plus, quel que soit l'écran :

- toute modification d'un tenant supprimé ;
- la suspension ou la réactivation d'un tenant en attente de suppression : il faut d'abord le restaurer (« La suppression de ce tenant est programmée : restaurez-le d'abord. ») ;
- la restauration d'un tenant qui n'est pas en attente de suppression.

Utilisateurs

Les utilisateurs se trouvent dans le menu **Gestion des tenants › Utilisateurs**.

Un **utilisateur** est le compte d'une personne. Il appartient à un client, son **client d'origine**, et peut être invité d'autres clients. Il n'a accès à un tenant que par un **accès** : un rôle sur ce tenant et au moins un plan, un plan au plus par service. Ce qui relève de Keycloak (connexion, mot de passe, sessions) est expliqué dans Identités et realms.

La liste des utilisateurs

La page s'intitule **Utilisateurs** : « Les comptes de chaque client. Leurs accès aux tenants se donnent depuis l'arbre des tenants ou la liste des accès. » L'en-tête propose **Importer** (voir Importer des utilisateurs) et **Nouveau**.

Au-dessus de la liste peuvent apparaître les modifications de compte Keycloak en attente ou en échec : voir Les modifications de compte Keycloak.

La liste affiche pour chaque utilisateur :

- **Identifiant** : lien vers sa page, avec la pastille « Désactivé depuis le … » s'il est désactivé ;
- **Prénom**, **Nom**, **E-mail** ;
- **Client** : son client d'origine, suivi de « +N » s'il est invité de N autres clients (info-bulle « Invité de : … ») ;
- **Dernière connexion** : il y a combien de temps, la date et l'heure exactes en info-bulle ; ou « Jamais », en gris, si aucune connexion n'a été vue (« Aucune connexion vue depuis que les connexions sont enregistrées »).

Les colonnes se trient.

Filtres

- **Recherche globale…** : identifiant, prénom, nom, e-mail.
- **Choisir un client** : les utilisateurs des clients choisis.
- **Inactif depuis** : **Toute activité**, **Sans connexion depuis 30 jours**, **… 90 jours**, **… 180 jours**, **Jamais connecté**. Chaque choix indique combien d'utilisateurs il retient. Un filtre en jours ne retient que les utilisateurs dont une connexion a été vue, plus ancienne que ce délai.
- **Afficher les utilisateurs désactivés (N)** : les utilisateurs désactivés sont masqués par défaut ; l'interrupteur les affiche, grisés.

Avec un filtre en jours, le bouton **Sélectionner ces N** coche, sur toutes les pages, les utilisateurs actifs qu'il retient, prêts pour **Désactiver**. Il n'est pas proposé pour **Jamais connecté** : ne pas avoir de connexion vue peut seulement vouloir dire qu'aucune n'a eu lieu depuis le début de l'enregistrement des connexions.

Actions

Le menu ··· d'un utilisateur propose **Voir**, **Modifier**, **Désactiver** (ou **Réactiver**) et **Supprimer**. **Supprimer** est grisé tant que l'utilisateur a accès à un tenant ; l'info-bulle le dit : « Suppression impossible tant que l'utilisateur a accès à N tenant(s). »

Dès qu'un utilisateur est coché, le bouton **Désactiver** de la barre agit sur la sélection (voir Désactiver et réactiver). Sa flèche propose **Supprimer** pour la sélection (voir Supprimer un utilisateur).

Créer ou modifier un utilisateur

Nouveau ouvre **Nouvel utilisateur** ; **Modifier** ouvre **Modifier l'utilisateur**.

La section **Compte** : « L'identifiant de connexion du compte et le client auquel il appartient. »

Champ	Obligatoire	Contrôles et aide
Domaine (realm)	non	Les realms acceptés par la plateforme ; à la création, le realm par défaut est déjà choisi. Aide : « Le realm Keycloak avec lequel le compte se connecte ; garder la valeur par défaut sauf si l'utilisateur relève d'un autre realm. » En modification : « Ne changer le realm que pour corriger un compte enregistré sous le mauvais realm. »
Identifiant	oui	Unique dans son realm, vérifié pendant la saisie : « Un autre utilisateur de ce realm a déjà cet identifiant. »
Nom du client	oui	Le client d'origine. Il ne se change plus après la création.
Multi-client	non	« Peut être invité sur les tenants d'autres clients que le sien. Laisser décoché sauf pour un consultant ou un auditeur. »
Langue des mails	non	**Français**, **English**, ou vide (« Non renseignée ») : « La langue des mails envoyés à cet utilisateur. Non renseignée : les mails sont en français et en anglais. »

La section **Identité** : « Affichée dans les listes et sur la fiche de l'utilisateur. »

Champ	Obligatoire	Contrôles
Prénom	oui	
Nom	oui	
E-mail	oui	Format d'e-mail ; unique sur toute la plateforme, tous realms confondus : « Un autre utilisateur a déjà cet e-mail. »

Quand le client a un realm, le compte doit être dans ce realm : le serveur refuse un autre realm. Un utilisateur encore invité d'autres clients ne peut pas perdre **Multi-client** : la case est verrouillée, avec « Encore invité de N autre(s) client(s) : retirez-les depuis la page de l'utilisateur avant de décocher Multi-client. »

Enregistrer crée ou modifie l'utilisateur, puis revient à la page précédente. **Retour** quitte sans enregistrer, avec la fenêtre **Modifications non enregistrées** si des valeurs ont été saisies.

> **i** Note
>
> Créer un utilisateur ici crée sa fiche dans subscription, pas son compte Keycloak. Voir Ce que garde subscription, ce que garde Keycloak.

La page d'un utilisateur

L'en-tête affiche le lien **Tous les utilisateurs**, les initiales, le prénom et le nom (ou l'identifiant), puis l'identifiant et le client d'origine (lien). Les boutons sont **Historique des accès** (l'historique filtré sur cet utilisateur, voir Historique des accès) et **Modifier**.

La page a quatre cartes :

- **Compte** : **Actif** ou « Désactivé depuis le … », une phrase sur ce que cela veut dire, le bouton **Désactiver** ou **Réactiver**, et la **Dernière connexion** ;
- **Identité** : **Identifiant**, **Prénom**, **Nom**, **E-mail** (lien de messagerie), **Client**, **Multi-client** (**Oui** ou **Non**), **Langue des mails** ;
- **Clients invités** : voir Clients invités ;
- **Accès aux tenants** : une ligne par tenant où l'utilisateur a un rôle ou des plans, avec **Tenant** (lien), **Rôle**, **Plans**, et la pastille **Suspendu** si son accès y est suspendu. **Voir dans la liste des accès** ouvre la liste des accès filtrée sur cet utilisateur ; **Donner un accès** ouvre le formulaire d'accès avec lui choisi.

Clients invités

Un utilisateur ne rejoint normalement que les tenants de son client d'origine. Un utilisateur **Multi-client** peut être **invité** d'autres clients, un client à la fois, et rejoindre alors leurs tenants comme l'un de leurs utilisateurs.

La carte **Clients invités** de la page de l'utilisateur :

- sans **Multi-client**, dit « Seul un utilisateur multi-client peut être invité d'autres clients : cochez d'abord Multi-client dans le formulaire de modification. » ;
- avec **Multi-client**, liste les clients dont il est invité (ou « Invité d'aucun autre client pour l'instant. »), chacun avec **Retirer**, puis propose **Choisir un client** et **Ajouter**. Le choix exclut le client d'origine et les clients déjà invités.

Retirer demande : « Retirer X des invités de C ? X est retiré de tous les tenants de C et recevra un e-mail pour l'en informer. »

Désactiver et réactiver

Désactiver un utilisateur coupe tous ses accès d'un coup, sur chaque tenant de chaque client, sans les perdre.

Désactiver (menu ···, carte **Compte**, ou barre de la liste pour la sélection) ouvre une fenêtre qui détaille :

- **S'arrête maintenant** : « Tous les accès sont coupés, sur chaque tenant de chaque client » ; « Le compte Keycloak est désactivé sous une minute et ses sessions ouvertes prennent fin ; il n'est pas supprimé ».
- **Conservé** : « Les plans et rôles sur chaque tenant » ; « Réactiver rend chaque accès tel qu'il était ».

Après confirmation : « X désactivé : le compte Keycloak suit sous une minute ». Pour une sélection, la désactivation porte sur les utilisateurs actifs cochés, un appel chacun ; un message dit combien l'ont été et combien n'ont pas pu l'être.

Réactiver demande « Réactiver X ? Ses accès reviennent sur chaque tenant et son compte Keycloak est réactivé sous une minute. »

Un utilisateur désactivé reste dans la liste, marqué désactivé. Les écrans le signalent par la pastille **Désactivé** (accès d'un tenant, liste des accès). Le fonctionnement est décrit dans La désactivation.

Les modifications de compte Keycloak

Désactiver, réactiver ou supprimer un utilisateur demande une modification de son compte Keycloak, faite en arrière-plan dans la minute. Tant qu'il en reste, le haut de la liste des utilisateurs les montre, relu toutes les 30 secondes :

- en attente : « N modification(s) de compte Keycloak en attente : » suivi de l'identifiant de chaque compte (info-bulle : son realm) ;
- en échec : « N modification(s) de compte Keycloak en échec », « Retentée automatiquement ; le compte Keycloak reste en l'état jusqu'à ce qu'un essai réussisse. » Chaque ligne donne l'action (**Désactivation** ou **Réactivation**), l'utilisateur et son realm, le nombre d'essais en échec, la date du prochain essai et la dernière erreur.

Rien ne s'affiche quand il n'y en a aucune.

Supprimer un utilisateur

Supprimer (menu ···) demande « Supprimer l'utilisateur X ? ». Le serveur refuse tant que l'utilisateur a accès à un tenant : retirez d'abord ses accès.

Pour une sélection (flèche du bouton **Désactiver**, **Supprimer**), les utilisateurs qui ont encore des accès sont ignorés : « Supprimer N de ces M utilisateurs ? K ont encore des accès et seront ignorés. » Le message final donne le nombre de supprimés, d'ignorés et d'échecs.

La suppression retire la fiche de l'utilisateur et désactive son compte Keycloak, sans le supprimer.

Importer des utilisateurs

Importer ouvre la page **Importer des utilisateurs** : « Créez plusieurs comptes d'un coup à partir d'un classeur Excel. » Elle a trois étapes.

1. **Préparer le fichier** : « Partez du modèle : une ligne par utilisateur, les colonnes dans l'ordre du modèle. » **Télécharger le modèle (.xlsx)** télécharge `userTemplate.xlsx`. Ses colonnes sont `Username`, `Email`, `FirstName`, `LastName`, `ClientName` (le nom exact d'un client) et `Language` (`fr`, `en` ou vide).
2. **Déposer le fichier** : choisissez le **Domaine (realm)** où les comptes sont créés (le realm par défaut est déjà choisi), puis **Choisir un fichier**. Un fichier `.xls` ou `.xlsx` de 50 Mo au plus ; l'import démarre dès que le fichier est choisi.
3. **Vérifier le résultat** : « N utilisateur(s) créé(s). N refusé(s). », avec la durée. Les lignes refusées sont listées : **Ligne** (le numéro de la ligne dans le classeur), **Identifiant**, **Motif**.

Une ligne est refusée quand :

- une colonne obligatoire est vide (toutes sauf `Language`) : « La colonne X est vide. » ;
- l'e-mail n'est pas valide, ou est déjà utilisé ;
- la langue n'est ni `fr` ni `en` ;
- aucun client ne porte ce nom ;
- le realm choisi n'est pas celui du client, quand le client en a un ;
- l'identifiant est déjà pris dans ce realm.

Les autres lignes sont créées ; les lignes vides sont sautées. « Corrigez ces lignes dans le fichier et réimportez-le : les comptes déjà créés sont ignorés. » **Importer un autre fichier** revient à l'étape 2 ; **Retour aux utilisateurs** revient à la liste.

Accès aux tenants

La liste **Accès aux tenants** n'a pas d'entrée de menu. Elle s'ouvre par **Voir les N accès** dans l'arbre des tenants, **Voir dans la liste des accès** sur la page d'un tenant ou d'un utilisateur (filtrée sur lui), ou le lien **Tous les accès** de la page d'un accès. « Une ligne par utilisateur et tenant : un rôle, puis les plans auxquels il a droit. »

L'en-tête propose **Nouveau** (voir Donner ou modifier un accès). Chaque ligne affiche :

- un œil qui ouvre la page de l'accès ;
- **Nom du tenant** (lien vers le tenant) ;
- **Identifiant** (lien vers l'utilisateur), avec les pastilles **Désactivé** et **Suspendu** le cas échéant. Quand ni le compte ni l'accès ne sont coupés, le lien **Consulter en tant que** ouvre la console support sur ce tenant et cet utilisateur ;
- **Nom du plan** : ses plans sur ce tenant ;
- **Rôle**.

La barre propose la recherche globale, trois filtres (**Rechercher par tenant…**, **Rechercher par plan…**, **Rechercher par identifiant…**) et **Supprimer**, actif dès qu'une ligne est cochée : « Retirer ces N accès ? Chaque utilisateur perd son rôle et ses plans sur le tenant concerné et recevra un e-mail pour l'en informer. » Le menu ··· d'une ligne propose **Voir**, **Modifier**, **Suspendre l'accès** (ou **Réactiver l'accès**) et **Supprimer**.

Rôle et plans

Sur un tenant, un utilisateur a :

- un **rôle** : **Membre**, qui « Utilise les fonctionnalités de ses plans. », ou **Administrateur**, qui « Gère en plus les accès des autres utilisateurs au tenant. » ;
- au moins un **plan**, un plan au plus par service. Un service dont il n'a pas de plan lui reste fermé. Les plans sont décrits dans Plans.

Chaque changement d'accès est enregistré dans l'Historique des accès, et l'utilisateur reçoit un e-mail qui décrit ses nouveaux droits.

Donner ou modifier un accès

Le formulaire **Donner un accès** s'ouvre par **Nouveau** (liste des accès), **Ajouter un utilisateur** (menu d'un tenant, tenant déjà choisi) ou **Donner un accès** (page d'un tenant ou d'un utilisateur, l'un ou l'autre déjà choisi). **Modifier** ouvre **Modifier l'accès**.

Qui et où : « Un utilisateur du client, sur l'un de ses tenants. »

- **Client** (création seulement, obligatoire) : choisir le client remplit les deux listes suivantes.
- **Nom du tenant** : un tenant du client.
- **Identifiant** : un utilisateur du client, ou un invité du client.

Dès que le tenant et l'utilisateur sont choisis, le formulaire lit ce que l'utilisateur détient déjà sur ce tenant et affiche :

- **Rôle** : **Membre** ou **Administrateur** ;
- **Plans, un par service** : une liste par service, avec **Aucun plan** comme valeur vide. « Un plan au plus par service ; sans plan, le service reste fermé. »

Enregistrer n'est actif qu'avec au moins un plan. Pour un utilisateur déjà membre, retirer tous les plans affiche « Un membre détient au moins un plan. Pour retirer l'utilisateur du tenant, utilisez Retirer du tenant. » Si l'enregistrement retire des plans, une confirmation les nomme : « Retirer les plans X de l'utilisateur sur ce tenant ? » Le rôle et tous les plans sont enregistrés d'un coup.

En modification, le client, le tenant et l'utilisateur sont verrouillés. Pour un utilisateur déjà membre, le bouton **Retirer du tenant** retire son rôle et tous ses plans, après la confirmation « Retirer X de T ? … ».

Le serveur refuse :

- un tenant qui n'est pas actif : « Ce tenant n'est pas actif : aucun accès ne peut y être donné ni modifié. » ;
- un utilisateur qui n'appartient pas au client du tenant et n'en est pas invité ;
- deux plans du même service.

La validité du tenant n'est pas vérifiée : un accès peut se préparer avant le début du contrat.

La page d'un accès

La page d'un accès s'ouvre par l'œil ou **Voir**. L'en-tête affiche le lien **Tous les accès**, l'identifiant de l'utilisateur et, le cas échéant, la pastille **Suspendu**. Les boutons sont **Suspendre l'accès** (ou **Réactiver l'accès**), **Retirer du tenant** et **Modifier**.

La page affiche **Nom du tenant** (lien), **Identifiant** (lien), **Rôle**, **Plans** (ou « Aucun plan ») et **Accès** : **Actif**, ou « Accès suspendu depuis le … ». Après **Retirer du tenant**, la page revient à la liste des accès.

Suspendre l'accès à un tenant

Suspendre l'accès coupe l'accès d'un utilisateur à un seul tenant : « Suspendre l'accès de X à T ? Son rôle et ses plans restent, mais aucun service ne l'ouvre plus sur ce tenant jusqu'à la réactivation. »

Réactiver l'accès demande « Rendre à X son accès à T, avec son rôle et ses plans ? ». Le serveur ne réactive un accès que dans un tenant actif.

Ne pas confondre :

	Porte sur	Coupe
Suspendre l'accès	un utilisateur sur un tenant	cet accès seulement

	Porte sur	Coupe
Désactiver l'utilisateur	un utilisateur	tous ses accès, et son compte Keycloak
Suspendre le tenant	un tenant	l'accès de tous ses utilisateurs

Qui peut faire quoi

Ces écrans sont réservés aux administrateurs de la plateforme, comme toute la console ; le serveur applique la même règle. Voir Les droits. Le rôle **Administrateur** d'un tenant, donné ici, ne donne pas accès à la console.

Historique des accès

L'historique des accès se trouve dans le menu **Gestion des tenants › Historique des accès**.

« Chaque changement des droits d'accès d'un utilisateur, le plus récent en premier : qui a donné ou retiré quel rôle ou quel plan, à qui, sur quel tenant. L'export liste qui détient quoi aujourd'hui. » La page ne modifie rien : elle lit l'historique et exporte la revue des accès.

Elle s'ouvre aussi, déjà filtrée, depuis le bouton **Historique des accès** de la page d'un tenant ou d'un utilisateur.

La chronologie

Les événements sont groupés par jour, le plus récent en premier. Chaque jour porte son nom (« Aujourd'hui », « Hier »), sa date et son nombre d'événements. Au-dessus, le total : « N événement(s) · page X sur Y ».

Chaque événement affiche :

- l'heure, et il y a combien de temps ; l'info-bulle donne la date et l'heure à la seconde ;
- son **type**, dans une pastille colorée selon sa nature : un droit donné, un droit retiré, un droit suspendu ou refusé, un droit modifié (voir Les types d'événements) ;
- l'utilisateur concerné (lien vers sa page) et son realm ;
- le tenant, « sur T » (lien vers le tenant), ou « sur tous les tenants » pour une désactivation ou une réactivation de compte ; puis le client ;
- pour un changement de plan, le nom du service ; pour un changement de rôle, « rôle » ; puis l'avant et l'après, « aucun » d'un côté vide (par exemple **Standard → Premium**, ou aucun → **Membre**) ;
- la cause, quand le changement découle d'un autre : « parce que le tenant a été purgé », « parce que l'invitation sur le client a été retirée », « car l'utilisateur a été supprimé » ;
- l'auteur du changement (voir Qui a fait le changement) ;
- deux boutons : **L'historique de cet utilisateur seulement** et **L'historique de ce tenant seulement**, qui ajoutent ce filtre.

La page montre 50 événements à la fois ; **Plus récents** et **Plus anciens** changent de page. **Actualiser** relit l'historique.

Sans événement : « Aucun changement de droits d'accès enregistré pour l'instant. », ou, avec des filtres, « Aucun changement de droits d'accès ne correspond à ces filtres. » et le bouton **Effacer les filtres**.

Les filtres

Filtre	Valeur vide	Remarque
Client	**Tous les clients**	Choisir un client restreint la liste des tenants à ses tenants, et efface un tenant choisi d'un autre client.
Tenant	**Tous les tenants**	Chaque tenant montre son client.
Utilisateur	**Tous les utilisateurs**	Chaque utilisateur montre son realm.
Dates	**Toutes les dates**	Une période du premier au dernier jour, inclus. Elle s'applique une fois les deux jours choisis.

Filtre	Valeur vide	Remarque
Types	**Tous les événements**	Plusieurs types possibles : « N type(s) d'événement ».

Les filtres se combinent. **Effacer les filtres** les retire tous. Ils sont écrits dans l'adresse de la page : un lien copié rouvre le même historique filtré.

Les types d'événements

Type	Nature	Ce qui s'est passé
Membre ajouté	donné	L'utilisateur a reçu un rôle sur le tenant.
Membre retiré	retiré	L'utilisateur a perdu son rôle sur le tenant.
Rôle modifié	modifié	Son rôle sur le tenant a changé.
Plan attribué	donné	Il a reçu un plan d'un service.
Plan modifié	modifié	Son plan d'un service a été remplacé par un autre.
Plan retiré	retiré	Il a perdu son plan d'un service.
Accès suspendu	suspendu	Son accès au tenant a été suspendu.
Accès réactivé	donné	Son accès au tenant a été réactivé.
Invité du client	donné	Il est devenu invité d'un client.
Invitation retirée	retiré	Il n'est plus invité d'un client.
Compte désactivé	suspendu	Son compte a été désactivé, sur tous les tenants.
Compte réactivé	donné	Son compte a été réactivé.
Politique de support modifiée, **Accès support demandé**, **Accès support autorisé**, **Accès support retiré**, **Accès support refusé**, **Session support ouverte**, **Session support fermée**, **Session support refusée**	selon le cas	Les accès du support MLA à un tenant : voir L'accès du support.

Un même changement peut donner plusieurs événements : donner un accès avec un rôle et deux plans enregistre **Membre ajouté** et deux **Plan attribué**.

Qui a fait le changement

À droite de chaque événement, son auteur :

- **une personne** : ses initiales, son identifiant et son realm ;
- **un service** : l'identifiant du service qui a appelé ;
- **une tâche planifiée** : « Système ». C'est le cas de la purge d'un tenant, qui retire ses membres à la fin du délai d'attente.

La revue des accès

Exporter la revue des accès (xlsx) télécharge `access-review.xlsx` : qui détient quel plan sur quel tenant aujourd'hui. Le bouton indique sa portée : le client choisi dans le filtre, ou « tous les clients ». Les autres filtres ne s'appliquent pas à l'export.

Le classeur a une feuille, `Access review`, avec une ligne par plan détenu, et une ligne sans service pour un membre qui n'a aucun plan. Les tenants supprimés n'y figurent pas. Les colonnes :

Colonne	Contenu
`Client`, `Tenant`, `Tenant status`	Le tenant et son statut.
`Realm`, `Username`, `First name`, `Last name`, `Email`	L'utilisateur.
`Home client`, `Guest`	Son client d'origine, et s'il est invité du client du tenant (`yes` ou `no`).
`Role`, `Suspended since`	Son rôle sur le tenant, et la date de suspension de son accès.
`Service`, `Plan`, `Service held since`	Le plan détenu, et depuis quand l'utilisateur détient un plan de ce service sur ce tenant.
`Last login`	Sa dernière connexion vue.

La ligne d'en-tête est figée et filtrable.

Conservation

Chaque changement est enregistré au moment où il est fait, dans la même opération : un changement annulé n'en laisse aucune trace. L'historique ne se modifie pas ; il est purgé chaque nuit de ses événements de plus de 90 jours.

Les changements sur un tenant de test temporaire ne sont pas enregistrés (voir Les tenants de test).

Qui peut le voir

La page est réservée aux administrateurs de la plateforme, comme toute la console. Voir Les droits.

Purge

La page de purge se trouve dans le menu **Gestion des tenants › Purge**.

« Les tenants supprimés, et ce que chaque service en garde encore : un tenant est purgé quand MyReg, la bibliothèque et ALM n'en gardent plus rien. »

Quand le délai d'attente d'un tenant supprimé est écoulé, subscription le purge et demande à chaque service d'effacer ses données (voir La purge). Cette page sert à vérifier que chaque service l'a bien fait, et à effacer les données qu'aucune purge n'atteint.

Le haut de la page choisit la vue : **Tenants supprimés** ou **Données orphelines**.

Tenants supprimés

Cette vue liste les tenants supprimés, le plus récent en premier. Elle ne fait que lire : elle ne supprime ni ne restaure rien.

En haut, des tuiles comptent les tenants affichés, puis ceux **Purgé**, **Données restantes**, **Incomplet** et **Non vérifié**.

Pour chaque tenant, le tableau affiche :

- **Tenant** : son nom, la pastille **test** pour un tenant de test, et sa référence (l'identifiant que connaissent les services) ;
- **Client** ;
- **Supprimé** : il y a combien de temps (la date exacte en info-bulle) ;
- **Services** : une pastille par service, **MyReg**, **Bibliothèque** et **ALM**, qui montre son dernier résultat, avec le nombre de lignes restantes s'il en reste ;
- **Statut** : le verdict (voir Vérifier un tenant) ;
- une loupe, **Vérifier maintenant**.

La barre propose une recherche (« Nom, identifiant du tenant ou client ») et trois portées, chacune avec son nombre : **Tous**, **Tenants de test**, **Tenants clients**. Un tenant de test est un tenant dont le nom contient `playwright` : ce sont les tenants temporaires que créent les tests automatisés.

L'en-tête propose **Actualiser**, qui relit la liste, et **Vérifier les N affichés**, qui vérifie tous les tenants affichés, trois à la fois.

Vérifier un tenant

Vérifier maintenant demande à MyReg, à la bibliothèque et à ALM combien de lignes et d'objets stockés ils gardent encore pour ce tenant. Chaque réponse s'affiche dès qu'elle arrive.

Statut	Signification
Non vérifié	Rien n'a encore été demandé.
Vérification…	Au moins un service n'a pas encore répondu.
Purgé	Les trois services répondent qu'il ne reste rien.
Données restantes	Au moins un service garde encore quelque chose.
Incomplet	Aucun service ne garde rien, mais au moins un n'a pas pu répondre.

Une fois vérifié, la flèche de la ligne ouvre le **Détail par service** : pour chaque service, l'heure de la vérification et soit « Plus rien. », soit la liste de ce qui reste, table par table (la mention **stockage** distingue un stockage d'objets d'une table), le plus gros en premier, soit la raison pour laquelle il n'a pas répondu :

- « Ce service ne sait pas encore compter les données d'un tenant (endpoint absent). »
- « Ce service a refusé la demande : réservée aux administrateurs de la plateforme. »
- « Ce service n'a pas répondu. »

> **Note**
>
> Un tenant **Données restantes** juste après sa suppression peut simplement attendre que le service traite la demande de purge. Si le reste persiste, il faut regarder le service concerné : la purge d'un service qui échoue est retentée, puis mise de côté (voir Ce que chaque service purge).

Données orphelines

« Les données que les services gardent pour un identifiant de tenant que subscription ne connaît pas du tout, quel que soit son statut. Aucune purge ne les atteint : subscription n'a jamais supprimé un tel tenant. »

La vue demande à chaque service la liste des tenants dont il garde des données, et la compare aux tenants que connaît subscription, supprimés compris. Un identifiant que seuls des services connaissent est orphelin.

En haut, le nombre d'identifiants orphelins et le total : « N lignes et objets stockés au total ». **Actualiser** relit tout.

Le tableau a une ligne par identifiant orphelin, le plus gros en premier : **Identifiant du tenant** (avec « inconnu de subscription »), une colonne par service avec ses lignes (et ses objets stockés), un **Total**, et le bouton **Purger**. Dans une colonne de service, un bouclier marque l'identifiant que ce service tient pour son tenant par défaut (« tenant par défaut de ce service : protégé, jamais purgé d'ici »).

Si un service n'a pas pu donner sa liste, un avertissement le nomme avec la raison : « Certains services n'ont pas pu lister leurs tenants, la liste peut manquer leurs orphelins : » Sans orphelin : « Aucune donnée orpheline : chaque identifiant de tenant que gardent les services est connu de subscription. »

Purger des données orphelines

> **Avertissement**
>
> Cette purge est définitive : « Chaque ligne et chaque objet stocké ci-dessous sont supprimés définitivement, dans chaque service. Il n'y a pas d'annulation et aucune sauvegarde n'est faite. »

Purger ouvre la fenêtre **Purger des données orphelines**. Elle liste, par service, ce qui sera supprimé, et les services ignorés parce que l'identifiant est leur tenant par défaut.

1. Tapez l'identifiant exact du tenant dans **Tapez l'identifiant du tenant pour confirmer**.
2. **Purger définitivement** demande d'abord à subscription s'il connaît maintenant ce tenant. Si oui, rien n'est envoyé : « subscription connaît maintenant ce tenant : rien n'a été envoyé, rien n'a été supprimé. »
3. Sinon, chaque service qui garde quelque chose de cet identifiant le purge, l'un après l'autre. Sa réponse s'affiche sous son nom :

Réponse	Signification
« supprimé : N lignes, N objets stockés »	Fait.
« refusé : subscription connaît ce tenant (409) »	Le service a vérifié auprès de subscription, qui connaît ce tenant.
« refusé : c'est le tenant par défaut du service (409) »	Le service protège cet identifiant.
« refusé : le service n'a pas pu interroger subscription (503) »	Le service n'a pas pu vérifier ; rien n'est supprimé.
« indisponible : pas encore d'endpoint de purge (404) »	Ce service ne sait pas encore purger.
« refusé : réservé aux administrateurs de la plateforme »	Refus d'autorisation.

Réponse	Signification
« échec »	Autre erreur.

Une fois que chaque service a répondu : « Chaque service a répondu. La liste est relue à la fermeture. » **Fermer** relit la liste.

Tenants par défaut des services

« Un service qui tourne sans subscription garde toutes ses données sous son tenant par défaut, que subscription ne connaît pas. Les données de ce service ne sont jamais purgées d'ici ; un autre service qui garde des données du même identifiant peut le faire apparaître parmi les orphelins. »

Le cadre **Tenants par défaut des services**, sous le tableau, liste ces identifiants : de quel service ils sont le tenant par défaut, et ce que chaque service en garde. Ils ne sont jamais purgés dans le service qui les nomme ainsi.

Qui peut le faire

La page est réservée aux administrateurs de la plateforme, comme toute la console. Chaque service vérifie aussi, de son côté, que l'appel vient d'un administrateur de la plateforme. Voir Les droits.

Console support

La console support se trouve dans le menu **Gestion des tenants › Console support**. Elle permet au support MLA et aux administrateurs d'ouvrir CIRCÉ ou ALM « en tant que » un membre d'un tenant, en lecture seule, de demander l'autorisation de ce membre quand elle manque, de régler la politique d'accès du tenant et de suivre le journal de ces accès.

Le fonctionnement complet (politiques, autorisations, sessions, journal) est décrit dans L'accès du support. Ce que voit l'utilisateur ciblé est décrit dans le manuel de CIRCÉ, page « Accès du support ».

Qui peut utiliser la console

Compte	Tenants affichés	Ce qu'il peut faire
Administrateur plateforme (rôle `ROLE_ADMIN` du realm MLA)	tous les tenants, sauf les tenants temporaires des tests de bout en bout	consulter, demander l'accès, régler la politique, voir le journal, **Terminer** une consultation
Membre du support (rôle `SUPPORT` du realm MLA)	les mêmes	consulter, demander l'accès, voir le journal ; ni la politique ni **Terminer**
Administrateur d'un tenant	les seuls tenants dont il est administrateur actif	consulter les membres de ses tenants, demander l'accès, régler la politique, voir le journal de ses tenants, **Terminer** une consultation dans ses tenants

Un compte qui n'est dans aucun de ces cas voit le message **Aucun tenant administré accessible.**

> **Note**
>
> **Un administrateur de tenant n'a pas accès au reste de la console.** La console support est la seule page qui lui est ouverte : elle est placée hors de la mise en page d'administration. Il l'ouvre avec l'adresse de la page suivie de `?realm=<son realm>` : la connexion se fait alors dans le realm de son organisation au lieu du realm MLA. Sans ce paramètre, la connexion se fait dans le realm MLA.

Ouvrir la console

Trois chemins mènent à la console :

- le menu **Gestion des tenants › Console support** ;
- dans la liste des **Accès**, le lien **Consulter en tant que** à côté de chaque membre actif (ni désactivé, ni suspendu) : la console s'ouvre sur le tenant de ce membre ;
- le lien du courriel envoyé au demandeur quand l'utilisateur accepte ou refuse sa demande : la console s'ouvre sur le tenant concerné (si l'adresse de la console est configurée, sinon le lien mène à la page du produit).

La console occupe toute la page. Elle a trois colonnes :

- à gauche, la liste des tenants ;
- au centre, le tenant choisi : ses membres, puis, pour qui l'administre, sa politique et ses autorisations actives ;
- à droite, le **Journal support** du tenant.

En bas de la colonne de gauche figurent votre identifiant, le choix de la langue (**FR** / **EN**) et le lien **Retour**, qui ramène à l'accueil.

Choisir un tenant

La liste **Tenants** est triée par nom. Chaque tenant montre sa politique (**Sur demande**, **Toujours** ou **Jamais**) avec une pastille de couleur. Au-delà de cinq tenants, le champ **Filtrer les tenants…** filtre la liste par nom. Le premier tenant est sélectionné à l'ouverture ; le tenant choisi est repris dans l'adresse de la page.

L'en-tête du tenant choisi indique :

- son nom et le sous-domaine de son client ;
- la durée maximale des autorisations (« autorisations jusqu'à … h ») ;
- sa politique, et sous l'en-tête, ce qu'elle permet :

Politique	Texte affiché
Jamais	Le support ne peut ouvrir les écrans d'aucun membre. Les demandes sont refusées.
Sur demande	Chaque utilisateur autorise lui-même le support, pour une durée limitée, et peut révoquer.
Toujours	Le support MLA autorisé peut consulter les écrans des membres sans demander.

Le bouton **Actualiser** (flèche circulaire) relit le tenant.

Les membres du tenant

Le tableau **Membres** liste les membres du tenant, triés par identifiant, avec leur realm, leur **Produit** (`CIRCE` pour CIRCÉ, `ALM` pour HERMES ALM, ou les deux) et leur **État** vu depuis votre compte. Le champ **Chercher un utilisateur…** filtre par identifiant ou realm.

Au-dessus du tableau, quatre compteurs servent aussi de filtres (un clic filtre, un second clic retire le filtre) :

Compteur	Membres comptés
consultables maintenant	vous pouvez ouvrir leur session tout de suite
à faire autoriser	une autorisation de l'utilisateur est requise et vous ne l'avez pas encore demandée
demandes en attente	vous avez déjà envoyé une demande, sans réponse
non consultables	aucune action possible : votre propre compte, un membre sans produit, ou un refus pour une autre raison

La colonne **État** affiche **Demande en attente** pour une demande envoyée, sinon la raison donnée par subscription :

État	Signification
Autorisé	Vous pouvez consulter. Avec une autorisation personnelle, sa fin et le temps restant s'affichent (« Jusqu'au … »).
Autorisation requise	La politique est **Sur demande** et l'utilisateur ne vous a pas autorisé.
Politique : jamais	La politique du tenant interdit les consultations.
Votre propre compte	On ne se consulte pas soi-même.
Intervenant non autorisé	Votre compte ne peut pas consulter ce tenant (compte désactivé, ou administrateur de tenant qui n'administre pas ce tenant).

État	Signification
Membre inactif	L'utilisateur n'est pas un membre actif du tenant, ou le tenant n'est pas actif.
Compte protégé ou désactivé	Le compte cible est désactivé ou porte un rôle privilégié (administrateur plateforme, support, administration de l'annuaire).
Collaborateur MLA	Les comptes des collaborateurs MLA ne sont jamais consultés.
Annuaire indisponible	Keycloak n'a pas répondu : réessayez plus tard.

Quand un membre a les deux produits, un sélecteur `CIRCE` / `ALM` choisit celui qui sera ouvert.

Demander l'accès

Pour un membre en état **Autorisation requise**, cliquez sur **Demander l'accès**. Un formulaire s'ouvre sous sa ligne :

1. **Durée suggérée** : **1 heure**, **4 heures** (choix par défaut), **1 jour** ou **7 jours**. C'est une suggestion : l'utilisateur choisit lui-même la durée qu'il accorde.
2. **Motif** : ce que l'utilisateur doit savoir, par exemple l'écran, le rapport ou l'erreur à examiner (2 000 caractères au plus).
3. Cliquez sur **Envoyer la demande**, ou **Annuler**.

La console affiche **Demande enregistrée. L'utilisateur doit lui-même autoriser l'accès.** L'utilisateur reçoit un courriel « Demande d'accès support » avec un lien vers la page **Accès support** du produit choisi, et un bandeau dans l'application. Le bouton du membre devient **Demandé**, et son état **Demande en attente**.

Une demande reste valable sept jours. Renvoyer une demande au même utilisateur tant que la précédente est en attente ne crée pas de doublon.

> **i** Note
>
> Une demande ne donne aucun accès. Seul l'utilisateur peut l'accepter, et l'autorisation qu'il donne en réponse ne vaut que pour vous : un autre membre du support doit faire sa propre demande.

Consulter en tant que

Pour un membre **Autorisé**, cliquez sur **Consulter en tant que →**. Un nouvel onglet s'ouvre sur le produit choisi, connecté avec l'identité du membre, dans ce tenant seulement et en lecture seule. Votre session dans la console reste ouverte.

Si le navigateur bloque l'onglet, la console affiche **Autorisez l'ouverture du nouvel onglet, puis réessayez.**

Pendant la consultation, un bandeau rappelle « Vous consultez … en tant que … (lecture seule) » ; **Quitter** y met fin. Une action d'écriture affiche **Action impossible en consultation support.** La consultation se termine aussi d'elle-même : fin de l'autorisation, révocation, durée maximale, inactivité ; voir les motifs de fin.

Politique et autorisations actives

Cette partie n'apparaît qu'aux administrateurs plateforme et aux administrateurs du tenant.

Politique d'accès du support présente trois cartes : **Jamais**, **Sur demande** et **Toujours**. La politique en vigueur porte la mention **Actuelle**. Choisir une autre carte fait apparaître **Enregistrer la politique** et **Annuler**. Après l'enregistrement, la console affiche **Politique enregistrée.** et le changement entre au journal.

> ⚠ Avertissement
>
> Cet accès permet au support autorisé de consulter les écrans en lecture seule, dans ce tenant. Il contourne la connexion par l'annuaire d'entreprise de l'utilisateur et ses contrôles MFA ou d'accès conditionnel. **Toujours** donne donc cette capacité en permanence : à décider avec le client.

Autorisations actives liste les autorisations en cours dans le tenant : l'utilisateur qui l'a donnée, sa date de fin (« Jusqu'au … ») et le temps restant. Sans autorisation, la liste affiche **Aucune autorisation en cours.**

Le journal support

Le **Journal support** liste, du plus récent au plus ancien et groupés par jour, les événements d'accès du support dans le tenant, 50 par page (**Précédent**, **Suivant**) :

Événement	Libellé
ouverture d'une consultation	« … a ouvert … en tant que … »
fin d'une consultation	« Fin de la session de … en tant que … », avec son motif
ouverture refusée	« Refusé : … en tant que … », avec la raison
demande d'accès	« … a demandé l'accès à … »
autorisation	« … a autorisé le support »
révocation	« … a révoqué l'autorisation de … »
demande refusée par l'utilisateur	« … a refusé une demande d'accès »
changement de politique	« … a changé la politique : … → … »

La seconde ligne d'un événement précise selon le cas le motif de fin, la raison d'un refus, l'échéance (« → date ») et l'application ouverte. La fin d'une consultation indique aussi les actions que la passerelle a bloquées pendant la session, par exemple « 1 action bloquée (lecture seule) » ; les types sont **lecture seule**, **autre tenant** et **nouvelle consultation**.

Filtres ouvre le formulaire :

- **Intervenant** : l'identifiant du membre du support. Ce filtre retient les ouvertures, fins et refus de consultation de cet intervenant ;
- **Utilisateur** : un membre du tenant ;
- **Depuis** et **Jusqu'à** : la période.

Appliquer lance la recherche, **Effacer** remet les filtres à zéro. Le nombre de filtres actifs s'affiche sur le bouton **Filtres**.

Exporter télécharge le journal filtré au format CSV (`support-access.csv`), toutes pages comprises.

Un administrateur de tenant ne voit que le journal de ses tenants. Les événements sont conservés 90 jours par défaut (voir L'accès du support).

Terminer une consultation en cours

Dans le journal, chaque consultation en cours (ouverture sans fin enregistrée) porte un bouton **Terminer**. Il n'apparaît qu'aux administrateurs plateforme et aux administrateurs du tenant.

1. Cliquez sur **Terminer**.
2. Confirmez : « Terminer maintenant la consultation de … en tant que … ? », puis **Terminer** (ou **Annuler**).

La session de l'intervenant s'arrête sur-le-champ ; sa requête suivante affiche la page **Consultation terminée**. Le journal se recharge et la fin indique « terminée par » suivi de votre identifiant.

Messages en cas d'échec :

Message	Cause
Vous ne pouvez pas terminer cette consultation.	subscription a refusé : vous n'êtes ni administrateur plateforme ni administrateur actif de ce tenant, ou la session est déjà close, ou Keycloak n'a pas répondu.
Cette consultation est déjà terminée.	La passerelle ne trouve plus la session ouverte.
La consultation n'a pas pu être terminée. Réessayez.	Autre erreur, par exemple subscription injoignable : rien n'a été terminé.

Un membre du support qui n'a pas ces droits termine sa propre consultation avec **Quitter**, dans le bandeau de l'onglet consulté.

Mises à jour en direct

La console se recharge d'elle-même quand une demande est créée, acceptée ou refusée, ou qu'une autorisation est donnée ou révoquée dans le tenant affiché : membres, demandes, autorisations et journal sont relus sans vider la page ni fermer le formulaire en cours.

Quand une de vos demandes reçoit une réponse, ou qu'une autorisation que vous déteniez est révoquée, une notification s'affiche en bas de la page pendant dix secondes :

- « … a accepté votre demande (durée) » ;
- « … a refusé votre demande » ;
- « L'accès au compte de … a été révoqué ».

Si une mise à jour est manquée, **Actualiser** relit le tenant.

II Plateforme Reglab

Services

Les services se trouvent dans le menu **Plateforme Reglab › Services**. Un service est un produit RegLab : il porte ses fonctionnalités, que les plans ouvrent aux utilisateurs (voir Concepts clés).

Comme toute la console, cette page est réservée aux administrateurs de la plateforme (rôle `ROLE_ADMIN`).

La liste des services

La page **Services** s'ouvre sur la liste des services : « Les produits RegLab. Chacun porte ses fonctionnalités, que les plans ouvrent aux utilisateurs. »

La liste affiche pour chaque service :

- le **Nom du service**, qui ouvre sa page ;
- sa **Description** ;
- ses **Plans** : une pastille « *n* plan(s) », qui ouvre la liste des Plans, ou « Aucun plan ».

Les en-têtes **Nom du service** et **Description** trient la liste. La zone **Recherche globale…** cherche dans le nom et la description.

Le menu ⋮ d'un service propose **Voir**, **Modifier** et **Supprimer**. Les listes, la sélection et la suppression groupée fonctionnent comme partout dans la console (voir Listes et formulaires).

Créer un service

Cliquez sur **Nouveau**. Le formulaire **Nouveau service** s'ouvre : « Ce formulaire sert à créer un service. »

Renseignez :

- le **Nom du service** : obligatoire, et unique. Un nom déjà pris est refusé : « Un service porte déjà ce nom. » ;
- la **Description** : facultative, en texte enrichi.

Cliquez sur **Enregistrer**. Le message « Service créé » s'affiche et la liste des services revient.

À sa création, le service reçoit automatiquement sa fonctionnalité `base` (description « Access to the service ») : c'est elle qui fait d'un utilisateur un membre du service, et tous les plans du service la portent (voir La fonctionnalité base).

> ⚠ Avertissement
>
> **Le nom du service est celui que les applications demandent.** Pour savoir ce qu'un utilisateur détient, une application demande à subscription son plan d'un service désigné par son nom (comparé sans tenir compte des majuscules) : CIRCÉ demande `irrc`, l'assistant demande `irrc`, `stress` et `library`. Renommer un service, c'est donc fermer ce service aux utilisateurs des applications qui le demandent sous son ancien nom. Voir Des plans aux droits.

Modifier un service

Utilisez **Modifier** dans le menu ⋮ de la liste, ou le bouton **Modifier** de la page du service. Le formulaire **Modifier le service** propose les mêmes champs. **Enregistrer** affiche « Service modifié » et ramène à la liste.

La page d'un service

La page d'un service porte son nom en titre et affiche :

- le **Nom du service** et sa **Description** ;
- ses **Fonctionnalités**, par ordre alphabétique, chacune liée à sa page, ou « Aucune fonctionnalité » ;

- ses **Plans**, par ordre alphabétique, chacun lié à sa page, ou « Aucun plan ».

Les boutons **Retour** et **Modifier** se trouvent en haut à droite.

Supprimer un service

Utilisez **Supprimer** dans le menu ⋮, ou cochez des services et cliquez sur **Supprimer** dans la barre d'outils. Une confirmation est demandée :

- pour un service : « Supprimer le service *nom* ? Le serveur refuse tant qu'un plan en fait partie. » ;
- pour une sélection : « Supprimer ces *n* services ? Le serveur refuse ceux dont un plan fait partie. »

Un service qui a encore des plans ne peut pas être supprimé : « Ce service ne peut pas être supprimé : des plans l'utilisent. » Supprimez ou déplacez d'abord ses plans (voir Plans).

La suppression d'un service supprime aussi toutes ses fonctionnalités, `base` comprise.

Plans

Les plans se trouvent dans le menu **Plateforme Reglab › Plans**. Un plan appartient à un service et ouvre une sélection de ses fonctionnalités ; un utilisateur reçoit des plans dans un tenant, au plus un par service (voir Concepts clés).

Comme toute la console, cette page est réservée aux administrateurs de la plateforme (rôle `ROLE_ADMIN`).

La liste des plans

La page **Plans** s'ouvre sur la liste des plans : « Un plan appartient à un service et ouvre une sélection de ses fonctionnalités. »

La liste affiche pour chaque plan :

- le **Nom du plan**, qui ouvre sa page ;
- son **Service**, qui ouvre la page du service ;
- ses **Fonctionnalités** : « *n* fonctionnalité(s) ».

Les en-têtes **Nom du plan** et **Service** trient la liste. La zone **Recherche globale…** cherche dans le nom du plan et le nom de son service.

Le menu ⋮ d'un plan propose **Voir**, **Modifier** et **Supprimer**. Les listes, la sélection et la suppression groupée fonctionnent comme partout dans la console (voir Listes et formulaires).

Créer un plan

Cliquez sur **Nouveau**. Le formulaire **Nouveau plan** s'ouvre : « Un plan d'un service, et les fonctionnalités qu'il ouvre. »

Renseignez :

- le **Nom du plan** : obligatoire. Il est unique sur toute la plateforme, tous services confondus ;
- le **Service** (« Choisir un service ») : obligatoire ;
- les **Fonctionnalités** (« Choisir les fonctionnalités qu'il ouvre ») : au moins une. La liste reste grisée tant qu'aucun service n'est choisi, puis ne propose que les fonctionnalités de ce service. Choisir un autre service ne garde que les fonctionnalités cochées qui appartiennent au nouveau service.

Cliquez sur **Enregistrer**. Le message « Plan créé » s'affiche et la page précédente revient.

Le serveur vérifie le plan :

- la fonctionnalité `base` du service est ajoutée au plan si elle n'a pas été cochée : un plan la porte toujours ;
- un nom déjà pris est refusé : « Un plan porte déjà ce nom, ou ouvre déjà les mêmes fonctionnalités. » ;
- un plan qui ouvrirait exactement les mêmes fonctionnalités qu'un autre plan du même service est refusé : « Un autre plan de ce service ouvre déjà les mêmes fonctionnalités. » ;
- un plan n'ouvre que des fonctionnalités de son propre service.

La page d'un plan

La page d'un plan porte son nom en titre. Le lien **Tous les plans** ramène à la liste ; les boutons **Supprimer** et **Modifier** se trouvent en haut à droite.

La page présente trois cartes :

- **Définition** : le **Nom du plan** et son **Service**, lié à la page du service ;
- **Fonctionnalités** : le nombre de fonctionnalités, puis chacune avec son nom (lié à sa page) et sa description, par ordre alphabétique. Au-delà de cinq fonctionnalités, la zone **Filtrer les fonctionnalités** filtre la liste sur le nom. Un plan sans fonctionnalité affiche « Ce plan n'ouvre encore aucune fonctionnalité. » ;
- **Utilisé par** : les tenants dont des utilisateurs détiennent le plan, chacun lié à sa page avec « *n* utilisateur(s) », ou « Aucun tenant n'utilise ce plan. »

Modifier un plan

Utilisez **Modifier** dans le menu ⋮ de la liste, ou le bouton **Modifier** de la page du plan. Le formulaire **Modifier le plan** (« Son nom, son service et ses fonctionnalités. ») propose les mêmes champs et les mêmes contrôles qu'à la création. **Enregistrer** affiche « Plan modifié » et ramène à la page précédente.

Quand les fonctionnalités d'un plan changent, chaque utilisateur qui le détient reçoit un e-mail avec les fonctionnalités qu'il gagne ou qu'il perd.

> ⚠ Avertissement
>
> **Les applications voient le changement avec un délai.** Modifier un plan déjà détenu (ses fonctionnalités, son service ou son nom) ne prévient pas les applications : elles continuent d'appliquer ce qu'elles ont gardé en mémoire jusqu'à son expiration, jusqu'à 5 minutes pour CIRCÉ et une heure par défaut pour l'assistant quand il s'agit de retirer `assistant`. Voir Ce que subscription ne signale pas.

Supprimer un plan

Utilisez **Supprimer** dans le menu ⋮ ou sur la page du plan, ou cochez des plans et cliquez sur **Supprimer** dans la barre d'outils. Une confirmation est demandée :

- pour un plan : « Supprimer le plan *nom* ? Le serveur refuse tant que des utilisateurs le détiennent. » ;
- pour une sélection : « Supprimer ces *n* plans ? Le serveur refuse ceux que des utilisateurs détiennent encore. »

Un plan détenu par au moins un utilisateur, dans n'importe quel tenant, ne peut pas être supprimé : « Ce plan ne peut pas être supprimé : des utilisateurs le détiennent encore. » La carte **Utilisé par** de sa page montre les tenants concernés ; retirez d'abord le plan aux utilisateurs (voir Clients et tenants).

Fonctionnalités

Les fonctionnalités se trouvent dans le menu **Plateforme Reglab › Fonctionnalités**. Une fonctionnalité est une capacité d'un service ; les plans du service en ouvrent une sélection aux utilisateurs (voir Concepts clés).

Comme toute la console, cette page est réservée aux administrateurs de la plateforme (rôle `ROLE_ADMIN`).

La liste des fonctionnalités

La page **Fonctionnalités** s'ouvre sur le catalogue : « Le catalogue de fonctionnalités à partir duquel les plans sont construits. »

Les fonctionnalités sont regroupées par service, les services par ordre alphabétique ; une fonctionnalité rattachée à aucun service apparaît sous « Sans service ». La liste affiche pour chacune :

- le **Nom de la fonctionnalité**, qui ouvre sa page ;
- sa **Description**.

Les en-têtes **Nom de la fonctionnalité** et **Description** trient la liste. La zone **Recherche globale…** cherche dans le nom de la fonctionnalité et le nom de son service.

Le menu ⋮ d'une fonctionnalité propose **Voir**, **Modifier** et **Supprimer**. Les listes, la sélection et la suppression groupée fonctionnent comme partout dans la console (voir Listes et formulaires).

Créer une fonctionnalité

Cliquez sur **Nouveau**. Le formulaire **Nouvelle fonctionnalité** s'ouvre : « Ce formulaire sert à créer une fonctionnalité. »

Renseignez :

- le **Nom de la fonctionnalité** : obligatoire, et unique dans son service. Un nom déjà pris dans le service est refusé : « Une fonctionnalité porte déjà ce nom. » ;
- le **Service** (« Choisir un service ») : obligatoire ;
- la **Description** : facultative, en texte enrichi.

Cliquez sur **Enregistrer**. Le message « Fonctionnalité créée » s'affiche et la liste revient. La nouvelle fonctionnalité n'est ouverte à personne tant qu'aucun plan ne la porte : ajoutez-la ensuite aux plans voulus (voir Plans).

> ⚠ Avertissement
>
> **Le nom est un code.** Les applications ouvrent un écran ou une action quand le plan de l'utilisateur porte une fonctionnalité d'un nom précis, par exemple `anacredit`, `pilar3` ou `mcp` dans CIRCÉ, `assistant` pour l'assistant. Une fonctionnalité qu'aucune application ne lit n'ouvre rien ; renommer une fonctionnalité lue ferme ce qu'elle ouvrait. La liste des noms lus est dans Des plans aux droits.

La description sert aux personnes : quand les droits d'un utilisateur changent, l'e-mail qu'il reçoit désigne chaque fonctionnalité par sa description, ou par son nom quand elle n'en a pas.

La fonctionnalité base

Chaque service a une fonctionnalité `base`, créée avec le service (description « Access to the service »). Elle fait d'un utilisateur un membre du service : CIRCÉ, par exemple, ne compte comme membre qu'un utilisateur dont le plan porte `base`.

Elle est protégée :

- elle ne peut être ni supprimée, ni renommée, ni rattachée à un autre service : « La fonctionnalité de base d'un service ne peut être ni supprimée, ni renommée, ni déplacée. » ;

- tout plan du service la porte : le serveur l'ajoute à un plan qui ne l'a pas.

Elle disparaît seulement avec son service.

La page d'une fonctionnalité

La page d'une fonctionnalité porte son nom en titre et affiche son **Nom de la fonctionnalité**, son **Service** et sa **Description**. Les boutons **Retour** et **Modifier** se trouvent en haut à droite.

Modifier une fonctionnalité

Utilisez **Modifier** dans le menu ⋮ de la liste, ou le bouton **Modifier** de la page de la fonctionnalité. Le formulaire **Modifier la fonctionnalité** propose les mêmes champs. **Enregistrer** affiche « Fonctionnalité modifiée » et ramène à la liste.

Le serveur refuse :

- de rattacher à un autre service une fonctionnalité qu'un plan ouvre : « Cette fonctionnalité ne peut pas changer de service : des plans l'utilisent. » ;
- de renommer ou de déplacer la fonctionnalité `base` (voir La fonctionnalité base).

Supprimer une fonctionnalité

Utilisez **Supprimer** dans le menu ⋮, ou cochez des fonctionnalités et cliquez sur **Supprimer** dans la barre d'outils. Une confirmation est demandée :

- pour une fonctionnalité : « Supprimer la fonctionnalité *nom* ? Le serveur refuse tant qu'un plan l'ouvre. » ;
- pour une sélection : « Supprimer ces *n* fonctionnalités ? Le serveur refuse celles qu'un plan ouvre. »

Une fonctionnalité qu'un plan ouvre ne peut pas être supprimée : « Cette fonctionnalité ne peut pas être supprimée : des plans l'utilisent. » Retirez-la d'abord des plans (voir Plans). La fonctionnalité `base` d'un service ne peut pas être supprimée.

III Configuration des services

Collecteur

Le collecteur se trouve dans le menu **Configuration des services › Collecteur**.

Le collecteur (le service `crawler`) parcourt les sites et les flux RSS des régulateurs, y trouve les publications réglementaires et les transmet à la bibliothèque commune, celle que lisent tous les tenants (voir Librairie). Il envoie aussi chaque jour, à 13 h 00, un mail de synthèse, le digest, à une liste de destinataires.

La page s'intitule **Collecteur** : « Les publications réglementaires trouvées sur les sites et flux des régulateurs, et transmises à la bibliothèque commune de tous les tenants. »

Comme toute la console, elle est réservée aux administrateurs de la plateforme. Le collecteur vérifie lui-même que l'appelant en est un (voir Droits).

> **i** Note
>
> Quand le collecteur ne répond pas, chaque liste l'indique : « Le collecteur n'a pas répondu. Est-il démarré ? ».

En-tête et bandeau de la dernière collecte

Le bouton **Collecter maintenant** lance une collecte de toutes les sources activées. Pendant la collecte, il devient **Collecte en cours…** et reste désactivé. Le message « Collecte lancée. Cette page la suit jusqu'à la fin. » confirme le lancement ; si une collecte tourne déjà, la page répond « Une collecte est déjà en cours. ».

Sous l'en-tête, un bandeau résume la dernière collecte :

- son état (voir Dernières collectes) ;
- « Dernière collecte le … » et sa durée ;
- « … nouvelle(s) publication(s) » ;
- « Prochaine collecte le … », ou « Pas de collecte planifiée ».

La page suit les collectes en direct : le collecteur signale chaque changement d'une collecte, et la console relit la liste aussitôt (voir Événements et appels). Tant qu'une collecte tourne, la page relit aussi les collectes toutes les 30 secondes, au cas où les mises à jour en direct seraient coupées. Quand une collecte se termine, les publications et les sources sont relues.

La page compte sept onglets : **Publications**, **Sources**, **Collectes & planification**, **Flux**, **Destinataires**, **Digests envoyés** et **Docling**.

Publications

L'onglet **Publications (n)** liste les publications trouvées par le collecteur, par pages de 25, 50 ou 100.

Les filtres sont :

- la recherche (**Titre, éditeur, référence…**) ;
- l'état des publications vis-à-vis de la bibliothèque : **Toutes**, **Envoyées**, **Écartées**, **Non envoyées**, **Indexées par la bibliothèque**, **En échec dans la bibliothèque**, **En attente de la bibliothèque**.

Les colonnes sont :

- **Publication** : le titre (ou la référence), lien vers la publication, puis l'éditeur, le type et la référence ;
- **Publiée le** ;
- **Détectée le** : quand le collecteur l'a trouvée ;
- **Transmise à la bibliothèque** : la date d'envoi et ce qu'en a fait la bibliothèque, **Indexée** (la date en infobulle), **Échec** (la raison en infobulle) ou **En attente** (« La bibliothèque n'a pas encore dit si elle l'a indexée. ») ; une publication écartée porte **Écartées** avec la raison, une publication jamais envoyée **Non envoyées** ;
- **Résumé** : voir Résumés pour le mail du jour ;
- un bouton **Retransmettre à la bibliothèque**.

Sans publication, la liste affiche « Aucune publication trouvée pour l'instant : lancez une collecte, ou ajoutez des flux. ».

Retransmettre à la bibliothèque

Une publication, envoyée, écartée ou non envoyée, peut être transmise de nouveau à la bibliothèque :

- le bouton **Retransmettre à la bibliothèque** de sa ligne l'envoie aussitôt ;
- après avoir coché des lignes, **Retransmettre à la bibliothèque (n)** demande d'abord : « Retransmettre n publication(s) à la bibliothèque ? La bibliothèque télécharge chacune et l'indexe si elle ne l'a pas déjà. ».

Le résultat s'affiche au-dessus de la liste : « n publication(s) transmise(s) à la bibliothèque. », ou, quand une partie n'a pas pu être remise au courtier de messages, le nombre de publications en échec, à réessayer.

Résumés pour le mail du jour

La bibliothèque peut résumer, avec un modèle de langage, les publications qu'elle a indexées. Les résumés partent dans le digest de 13 h 00.

Un résumé se demande pour une publication indexée qui n'a jamais été résumée, ou dont le résumé a échoué. Sa case se coche dans la colonne **Résumé** ; la barre au-dessus de la liste propose :

- **Tout cocher (n)** : toutes les publications de la page qui peuvent être résumées ;
- **Décocher** ;
- **Lancer les résumés (n) · ~x tokens**, qui demande : « Faire résumer n publication(s) par la bibliothèque ? Environ x tokens en entrée, plus ceux de la réponse. ».

La colonne **Résumé** montre l'état de chaque résumé : **Non demandé**, **En cours**, **Résumée** (la date et le modèle en infobulle) ou **Échec** (la raison en infobulle). Elle affiche aussi les tokens estimés avant la demande, puis les tokens consommés en entrée et en sortie, et le coût en dollars quand la bibliothèque le donne. La barre totalise les résumés de la page.

Lire le résumé ouvre le résumé tel que le digest le reprend.

Tant qu'un résumé est en cours, la page relit l'état des résumés toutes les 10 secondes.

Sources

L'onglet **Sources (n)** liste toutes les sources que lit le collecteur : les sites des régulateurs (**Site**) et les flux RSS (**Flux**). « Une source désactivée est laissée hors des exécutions ; « Lancer » la parcourt seule, activée ou non. »

Les colonnes sont :

- **Source** : son nom, lien vers son site, et son type ;
- **Dernière exécution** : la date, puis le nombre de publications listées, ou **En échec** avec l'erreur ; « Pas encore lue » pour une source jamais lue ;
- **Dernier succès** ;
- **Parcourue** : l'interrupteur qui active ou désactive la source à partir de la prochaine collecte ;
- **Lancer** : parcourt cette seule source ; le bouton est désactivé pendant une collecte.

Actualiser relit la liste.

Collectes & planification

L'onglet **Collectes & planification** réunit la planification, les réglages de parcours et le journal des collectes.

Planification

- **Quand collecter (cron)** : six champs, « seconde minute heure jour-du-mois mois jour-de-la-semaine », lus dans le fuseau affiché. Par exemple `0 0 1 * * *` collecte chaque jour à 01:00.
- Quatre raccourcis remplissent le champ : **Chaque jour à 01:00** (`0 0 1 * * *`), **En semaine à 06:30** (`0 30 6 * * MON-FRI`), **Toutes les 6 heures** (`0 0 */6 * * *`), **Toutes les heures** (`0 0 * * * *`).

- **Collecter selon cette planification** : décochée, le collecteur ne se lance pas seul ; il faut alors utiliser **Collecter maintenant**.

Sous le formulaire s'affichent la prochaine collecte, et qui a modifié la planification et quand. **Enregistrer** n'est actif que si la planification a changé et que le cron compte six champs. Elle s'applique immédiatement.

Comment le crawler parcourt

Champ	Valeurs	Effet
Profondeur (mois)	1 à 120	Jusqu'où un parcours peut remonter ; une source s'arrête plus tôt, à une page de publications déjà connues.
Pause entre deux requêtes (ms)	0 à 10 000, par pas de 100	Entre deux requêtes au même site. À augmenter si un site se met à refuser le collecteur.
Délai d'attente (s)	1 à 300	Pour se connecter à un site et pour chaque lecture ; un site qui ne répond pas à temps met sa source en échec.

Les trois valeurs sont des nombres entiers. **Enregistrer** n'est actif que si une valeur a changé et que toutes sont dans leurs bornes. Les réglages s'appliquent dès la prochaine collecte.

Dernières collectes

Le tableau montre les 30 dernières collectes ; **Actualiser** le relit.

- **Début** ;
- **Lancée par** : **Démarrage du collecteur**, **Planification** ou **Collecter maintenant**, avec l'utilisateur ; une collecte limitée à une source porte « seulement « nom » » ;
- **État** : **En cours**, **Réussie**, **Échec partiel**, **Échouée** ou **Interrompue** ;
- **Durée** ;
- **Nouvelles publications**.

Une ligne se déplie et montre :

- **Sources en échec**, chacune avec son erreur ;
- **Ce que chaque source a coûté et rapporté** : par source, **Requêtes**, **Durée**, **Listées**, **Nouvelles**, puis ce qu'en a fait la bibliothèque, **Indexées**, **Échecs**, **En attente**. Une source qui a fait au moins 20 requêtes sans trouver de nouvelle publication porte **Rien pour beaucoup de requêtes**. Pour les collectes plus anciennes, qui n'ont pas mesuré ces chiffres, la page n'affiche que les **Publications listées par chaque source** ;
- **Publications trouvées** : les nouvelles publications de cette collecte, en liens.

Flux

L'onglet **Flux (n)** liste les flux RSS que le collecteur lit à chaque collecte, en plus des sites des régulateurs. Un flux se désactive dans l'onglet Sources ; il porte alors **Désactivé**. Une icône signale un flux qui a un filtre d'inclusion ou d'exclusion.

Les colonnes sont **Nom**, **Éditeur**, **Format** et **URL**, puis les boutons de modification et de suppression.

Ajouter ou modifier un flux

Ajouter un flux ouvre le formulaire :

- **Nom** (obligatoire) ;
- **Éditeur** (obligatoire), par exemple `ECB`, `BIS` ;
- **URL** (obligatoire), commençant par `http://` ou `https://` ;

- **Ne garder que les éléments correspondant à** (facultatif) ;
- **Écarter les éléments correspondant à** (facultatif).

Les deux filtres sont des expressions régulières, sans distinction de casse, de 1 000 caractères au plus. Chacune est cherchée dans le titre, dans chaque catégorie et dans la description. Un élément du flux est gardé quand la première est trouvée dans l'un d'eux (ou est vide) et que la seconde n'est trouvée dans aucun. Une expression que le collecteur refuse est signalée sous son champ.

Tester lit le flux une fois, sans l'enregistrer, et affiche le nombre d'éléments listés, combien les filtres en gardent, et combien des premiers sont des documents que la bibliothèque peut indexer. Suivent la liste des éléments **Gardés**, avec la raison de ceux que la bibliothèque écarterait, et celle des éléments **Exclus par les filtres**, avec la raison de chacun.

Enregistrer n'est actif que si les trois champs obligatoires sont remplis et l'URL valide. Un nouveau flux est lu dès la prochaine collecte.

Supprimer un flux

La suppression demande : « Supprimer le flux « nom » ? Les publications déjà trouvées restent dans la bibliothèque. ».

Destinataires

L'onglet **Destinataires (n)** gère qui reçoit le digest. « Le digest part chaque jour à 13 h 00 vers les destinataires activés, un mail chacun. »

> ⚠ Avertissement
>
> Les mails passent par le service de notification, qui n'accepte que certains domaines. Une adresse hors de ces domaines est refusée : envoyez-lui un mail de test pour la vérifier.

L'en-tête indique le nombre de destinataires activés sur le total, et propose **Actualiser** et **Ajouter un destinataire**.

Les colonnes sont :

- **Adresse e-mail** ;
- **Nom** ;
- **Régulateurs** : ceux dont le destinataire reçoit les publications, ou **Tous les régulateurs** ;
- **Reçoit le digest** : l'interrupteur qui l'active ou le désactive ;
- **Dernière modification** : la date et l'auteur ;
- les boutons **Modifier**, **Envoyer un mail de test** et **Supprimer**.

La suppression demande : « Retirer … des destinataires du digest ? ».

Le formulaire d'ajout ou de modification contient :

- **Adresse e-mail** (obligatoire) : une adresse valide, de 254 caractères au plus, absente de la liste (« Cette adresse est déjà dans la liste ») ;
- **Nom** (facultatif), 100 caractères au plus ;
- **Régulateurs** : **Tous les régulateurs**, « y compris ceux que le collecteur ajoutera », ou **Certains seulement**, choisis dans **Choisir les régulateurs** (au moins un). Dans ce second cas, le destinataire ne reçoit que les publications de ces régulateurs, et pas de mail les jours où il n'y en a aucune.

Digests envoyés

L'onglet **Digests envoyés** liste, jour par jour et du plus récent au plus ancien, chaque digest que le collecteur a envoyé, par pages de 20, 50 ou 100. Les colonnes sont **Jour**, **Mails**, **Publications** (et combien sont indexées), **Destinataires** et **Envoyé à**.

Pour chaque jour :

- **Ouvrir le digest** affiche les mails tels qu'ils sont partis ; quand le digest a été envoyé en plusieurs mails, un sélecteur permet de passer de l'un à l'autre ;
- **Renvoyer à une adresse** renvoie les mails tels qu'ils ont été envoyés, à cette seule adresse, qui n'a pas besoin d'être destinataire ; la liste des destinataires ne change pas. Quand le jour compte plusieurs

mails, **Mails** permet de choisir **Tous les mails du jour (n)** ou un seul. Une adresse hors des domaines autorisés par le service de notification ne reçoit rien ;
- **Copier le lien public** (quand le navigateur l'autorise) et **Ouvrir la page publique** donnent accès à la page publique du digest ; « Pas de page publique » sinon.

Aperçu du prochain digest montre le digest tel qu'il partirait maintenant : la date d'envoi, le nombre de publications et de destinataires, ou « Le digest est désactivé » avec le nombre de publications en attente. La bibliothèque peut encore indexer ou résumer des publications d'ici l'envoi ; **Actualiser** relit l'aperçu.

Docling

L'onglet **Docling** suit la file de conversion de la bibliothèque : docling-serve convertit les documents de la bibliothèque en Markdown, quelques-uns à la fois, et les autres attendent leur tour. La conversion concerne les documents communs et les documents des clients.

L'onglet affiche :

- l'état de docling-serve : **docling-serve répond**, avec le nombre maximal de conversions simultanées, ou **docling-serve ne répond pas** : les conversions sont alors suspendues et les documents en file gardent leur place ;
- **En file**, réparti entre documents communs et documents clients ;
- **Convertis** et **Échecs (sans Markdown)** dans la dernière heure ;
- **Durée moyenne** par document, sur la dernière heure ;
- **Fin estimée** de la file ;
- le tableau **En cours de conversion (n sur max)** : **Document**, **Fichier**, **Bibliothèque** (**Commune** ou **Client**), **Commencée à**, **Depuis**.

L'onglet se met à jour en direct, et relit la file toutes les 10 secondes. Si la bibliothèque ne répond pas, il garde les chiffres de la dernière lecture et le signale. Une bibliothèque qui ne publie pas encore sa file affiche « Pas encore disponible ».

La gestion des documents eux-mêmes (validation du Markdown, relance de docling) se fait dans Librairie.

Notifications

Les notifications se trouvent dans le menu **Configuration des services › Notifications**.

Les services de la plateforme n'envoient pas leurs mails eux-mêmes : ils les confient au service de notification (`notification`), qui les envoie par Azure Communication Services et en garde l'historique. La page **Mails envoyés** montre cet historique : « Les mails envoyés par la plateforme, les plus récents d'abord. Ils sont conservés 60 jours. ».

On y trouve donc les mails de tous les services qui passent par le service de notification, par exemple le digest du collecteur et les mails de subscription décrits plus bas.

Comme toute la console, la page est réservée aux administrateurs de la plateforme. Le service de notification vérifie lui-même qui lit l'historique et qui renvoie un mail : seuls les administrateurs de la plateforme peuvent renvoyer un mail.

La liste des mails

La liste est paginée par 25, 50 ou 100 mails.

Les filtres sont :

- la recherche (**Objet ou destinataire…**) ;
- le statut : **Tous**, **Envoyé**, **En cours**, **Échec**, **Refusé**.

Les colonnes sont :

- **Date** : quand le service de notification a reçu le mail ;
- **Statut** : le statut d'envoi et, dès qu'un rapport de distribution est arrivé, un second badge (voir Distribution) ;
- **Objet** ; un mail qui en renvoie un autre porte le lien **Renvoi**, qui ouvre l'original ;
- **Destinataires** : les trois premiers, puis « +n » ; l'infobulle les donne tous ;
- le nombre de pièces jointes (icône trombone) ;
- **Tentatives**.

Statut	Signification
Envoyé	Remis au service de messagerie.
En cours	En cours d'envoi, ou en attente d'une nouvelle tentative.
Échec	Abandonné après les nouvelles tentatives.
Refusé	Refusé sans envoi : un destinataire n'est pas autorisé.

> **Note**
>
> Le service de notification n'envoie qu'aux adresses des domaines qu'il autorise. Un seul destinataire hors de ces domaines fait refuser tout le mail : il apparaît alors **Refusé**.

Quand le service de notification ne répond pas, la liste affiche « Le service de notification n'a pas répondu. Est-il démarré ? ».

Le détail d'un mail

Un clic sur une ligne ouvre le mail :

- **Statut**, avec sa signification ;
- **Reçu** et **Envoyé** ;

- **Tentatives** ;
- **Destinataires**, tous ;
- **Renvoi de**, pour un mail qui en renvoie un autre : un lien ouvre l'original ;
- **Dernière erreur :**, quand l'envoi en a rencontré une ;
- **Distribution** (voir ci-dessous) ;
- **Pièces jointes** : chacune se télécharge ;
- **Contenu** : le corps du mail, affiché tel qu'il a été envoyé.

Un mail de plus de 60 jours n'est plus lisible : « Le mail n'a pas pu être lu. Il a peut-être plus de 60 jours, ou le service de notification n'a pas répondu. ».

Distribution

Envoyé signifie que le service de messagerie (Azure Communication Services) a accepté le mail. **Distribué** signifie que le serveur de messagerie du destinataire l'a pris, d'après le rapport reçu quelques minutes plus tard, destinataire par destinataire.

Le tableau **Distribution** donne, pour chaque destinataire, le **Statut**, la date de la **Tentative** et le **Message** du serveur.

Statut de distribution	Signification
Distribué	Le serveur de messagerie du destinataire a pris le mail.
Étendu	Remis à une liste de diffusion, qui le distribue à ses membres.
Inconnu	Le serveur de messagerie du destinataire n'a pas donné de réponse claire.
Supprimé	Non envoyé : l'adresse est sur la liste de suppression du service de messagerie (rejet ou désinscription antérieurs).
Filtré comme spam	Filtré comme spam en chemin.
En quarantaine	Mis en quarantaine par la messagerie du destinataire.
Rejeté	Refusé par le serveur de messagerie du destinataire (adresse inconnue, boîte pleine…).
Non distribué	Le service de messagerie n'a pas pu le distribuer.

Dans la liste, le badge de distribution résume tous les destinataires, ou le pire d'entre eux.

Sans rapport, le détail l'explique : pour un mail **Envoyé**, le rapport peut mettre quelques minutes à venir, et les mails envoyés avant l'arrivée des rapports n'en ont pas ; pour un autre mail, le service de messagerie ne l'a jamais accepté.

Renvoyer un mail

Le bouton **Renvoyer**, en bas du détail, envoie de nouveau le mail aux mêmes destinataires. Il demande d'abord : « Renvoyer « objet » aux mêmes n destinataire(s) ? Un nouveau mail est créé ; celui-ci reste dans l'historique. ».

Le nouveau mail s'ouvre ensuite et le résultat s'affiche : « Mail renvoyé : » suivi de son statut.

Le renvoi est refusé :

- pour un mail **En cours** : « Ce mail est encore en cours d'envoi : il pourra être renvoyé une fois envoyé, en échec ou refusé. » ;
- à un utilisateur qui n'est pas administrateur de la plateforme : « Vous n'êtes pas autorisé à renvoyer des mails. » ;
- pour un mail qui n'existe plus : « Ce mail n'existe plus : il a peut-être plus de 60 jours. ».

Les mails envoyés par subscription

subscription envoie lui-même quatre sortes de mails, par le service de notification. Ils partent une fois la modification enregistrée : une modification annulée n'envoie rien, et un envoi qui échoue est consigné dans les journaux du service sans annuler la modification.

Mail	Quand	Destinataires	Objet (en français)
Changement de droits	Le rôle d'un utilisateur dans un tenant, ou ses services et fonctionnalités, ont changé	L'utilisateur	« Vos droits ont changé – espace »
Statut du tenant	Le tenant est suspendu, mis en suppression, rouvert, ou sa suppression est annulée	Les membres du tenant qui ont une adresse e-mail ; à la réouverture, pas ceux dont l'accès personnel reste suspendu	« Accès suspendu », « Espace en cours de suppression », « Accès rétabli », « Suppression annulée »
Fin de contrat	La date de validité du tenant approche ou est passée	MLA, et les administrateurs du tenant	« Fin de contrat dans n jours », « Fin de contrat aujourd'hui », « Fin de contrat »
Accès support	Une demande d'accès support est faite, autorisée ou refusée	L'utilisateur concerné pour une demande ; le demandeur pour une autorisation ou un refus	« Demande d'accès support », « Accès support autorisé », « Demande d'accès support refusée »

Les mails de changement de droits, de statut et de fin de contrat sont dans la langue choisie par l'utilisateur, ou en français puis en anglais quand il n'en a choisi aucune ; MLA reçoit les siens dans la langue configurée pour lui. Les mails d'accès support sont en français ou en anglais, selon la langue du destinataire. Un tenant temporaire n'envoie pas de mail de changement de droits.

Les rappels de fin de contrat

Chaque jour (à 08:00, heure de Paris, par défaut), subscription cherche les tenants actifs dont la date de validité approche ou vient de passer, et envoie :

- à MLA, un seul mail à toutes ses adresses, 60, 30 et 7 jours avant la fin (par défaut) ;
- à chaque administrateur du tenant qui a encore accès et une adresse e-mail, 30 et 7 jours avant la fin (par défaut) ;
- aux deux, un mail une fois la date passée, dans les 7 jours qui suivent (par défaut).

Un rappel déjà envoyé ne l'est plus ; un rappel dont l'envoi échoue est retenté au passage suivant. Un tenant sans date de validité n'est jamais concerné.

> **Note**
>
> Ces rappels ne se règlent pas dans la console. L'heure, les seuils, les adresses et la langue des mails de MLA sont des propriétés de configuration du service (`application.contract-reminders.*`). Les contrats qui approchent de leur fin, et les rappels envoyés pour chacun, se consultent depuis les tenants (voir Tenants).

> **Avertissement**
>
> Sans les identifiants Kafka dédiés de subscription (`application.notification.kafka-username` et `kafka-password`), aucun de ces mails ne part ; le service le signale une fois au démarrage. Voir Événements et appels.

Librairie

La librairie se trouve dans le menu **Configuration des services › Librairie**.

La page **Documents communs** gère la bibliothèque commune du service `library` : « Réglementations et lignes directrices de la bibliothèque commune, que lit chaque tenant et où cherche le chat. ». Les documents des tenants eux-mêmes n'y figurent pas.

La bibliothèque commune reçoit des documents de deux façons : ceux que le collecteur trouve chez les régulateurs, et ceux qu'un administrateur ajoute ici.

Comme toute la console, la page est réservée aux administrateurs de la plateforme ; la bibliothèque et subscription vérifient eux-mêmes que l'appelant en est un.

La page compte trois onglets : **Documents**, **Téléversement** et **Index de recherche**.

Le parcours d'un document

Un document de la bibliothèque passe par ces étapes, que montre la colonne **Statut** de l'onglet **Documents** :

Statut	Ce qui se passe
En attente	Le document attend son tour dans la file de docling.
Conversion	docling le convertit en Markdown.
Markdown à valider	docling a produit un Markdown, qu'un administrateur doit relire et valider.
Sans Markdown	La conversion n'a produit aucun Markdown.
Vectorisation	La bibliothèque découpe le document en passages et les vectorise pour le chat.
Vectorisé	Le chat peut retrouver le document par le sens.
Échec de la vectorisation	La vectorisation a échoué.

Un document n'est vectorisé pour le chat qu'une fois son Markdown validé, ou, quand docling n'en a produit aucun, une fois demandée la vectorisation de son texte brut. La file de docling elle-même se suit dans l'onglet Docling du collecteur.

Documents

L'onglet **Documents** montre en haut les embeddings du chat, puis la liste des documents communs.

Embeddings du chat

« Chaque document est découpé en passages et vectorisé, pour que le chat le retrouve par le sens. La bibliothèque ne vectorise qu'à la demande, chaque vectorisation consommant des tokens du modèle. »

Le bloc compte les documents vectorisés sur le total, en cours, en échec, pas encore vectorisés, et ceux découpés avec une ancienne taille de passage. Il liste les échecs, avec leur erreur.

Trois boutons lancent une vectorisation en arrière-plan, chacun après une confirmation qui rappelle le coût en tokens :

- **Vectoriser n documents** : ceux jamais vectorisés ou en échec ;
- **Revectoriser les n documents obsolètes** (affiché quand il y en a) : ceux découpés avec une ancienne taille de passage, redécoupés à la taille actuelle ;
- **Revectoriser tous les documents communs**.

Ces boutons ne sont actifs qu'une fois que la bibliothèque a dit qu'aucune vectorisation n'est en cours, et que les embeddings sont activés dans l'environnement ; sinon le bloc affiche « Les embeddings sont désactivés dans cet environnement. ». Pendant une vectorisation, une barre de progression s'affiche.

La liste des documents

Les filtres sont :

- **Filtrer par titre ou nom de fichier** ;
- le statut (**Tous les statuts** par défaut) ;
- **À valider (n)** : les documents dont le Markdown attend une validation ;
- **Passages anciens (n)** (affiché quand il y en a) : les documents vectorisés avec une ancienne taille de passage.

Les colonnes sont **Titre**, **Fichier**, **Publié le**, **Statut** et **Chat** ; les quatre premières trient la liste. La colonne **Chat** indique **Vectorisé**, **Non vectorisé**, ou **Vectorisé · passages anciens**, l'infobulle donnant alors la taille de découpage d'origine.

Après avoir coché des lignes :

- **Supprimer la sélection (n)** ;
- **Valider la sélection (n)** : seuls les documents dont le Markdown attend une validation sont envoyés ; les autres sont laissés de côté, et le message le dit.

Relancer docling sur les documents sans Markdown (n) remet en file tous les documents **Sans Markdown**, après une confirmation : docling les convertit l'un après l'autre, ce qui peut prendre longtemps, et les fichiers qu'il ne sait pas lire sont laissés de côté.

La liste se met à jour en direct quand un document commun change, et se relit toutes les 30 secondes tant que des documents sont en attente, en conversion ou en vectorisation.

Les actions sur un document

Chaque ligne propose, selon le statut du document :

- **Aperçu** : ouvre le Markdown produit par docling, rendu ou en source (**Voir la source Markdown** / **Voir le rendu**). Si la conversion est encore en cours, ou n'a rien produit, l'aperçu le dit. Le bas de l'aperçu propose **Relancer docling**, **Vectoriser le texte brut** et **Valider** ;
- **Valider** (statut **Markdown à valider**) : la bibliothèque vectorise le document à partir de son Markdown ;
- **Vectoriser le texte brut** (statut **Sans Markdown**) : la bibliothèque vectorise le texte brut du fichier ;
- **Relancer**, un menu de trois actions :
  - **Relancer docling** : impossible tant que le document est en attente ou en conversion. Le Markdown actuel est remplacé, et le nouveau devra être validé ; une confirmation le rappelle, sauf pour un document sans Markdown ;
  - **Relancer l'indexation** : la bibliothèque relit le fichier et le remet dans l'index de recherche ; impossible tant que le document est en attente, en conversion ou en vectorisation ;
  - **Relancer la vectorisation** : pour un document **Vectorisé** ou **Échec de la vectorisation** ;
- **Supprimer**.

> ⚠ Avertissement
>
> La suppression est définitive et touche tous les tenants : « Supprimer « nom » de la bibliothèque commune ? Tous les tenants le perdent, et c'est définitif. ». Le document quitte la bibliothèque et le chat.

Téléversement

L'onglet **Téléversement** ajoute des documents à la bibliothèque commune : « La bibliothèque les indexe en arrière-plan. ».

Téléverser des fichiers

Glissez des fichiers dans la zone, ou cliquez sur **parcourez votre ordinateur**. Sont acceptés : `.pdf`, `.doc`, `.docx`, `.odt`, `.txt`, de 50 Mo au plus chacun, chaque nom une seule fois.

Chaque fichier s'affiche avec son **Format** et son **Statut** : **Prêt**, **Format non pris en charge**, **Trop volumineux** ou **Doublon**. Les fichiers refusés ne sont pas envoyés ; une poubelle retire un fichier de la liste.

Indexer depuis le web

Le champ **URL des documents, une par ligne** reçoit des adresses de pages ou de PDF ; la bibliothèque télécharge chacun et l'indexe. Les lignes vides sont ignorées ; une ligne qui n'est pas une URL `http(s)` (**Pas une URL http(s)**) ou qui répète une autre (**Doublon**) est écartée.

Envoyer

La barre du bas compte les fichiers et les URL prêts ; **Envoyer à la bibliothèque commune (n)** les envoie. Une barre de progression suit chaque fichier.

Ensuite, la page indique combien de documents sont partis, et propose **Ajouter d'autres documents** et **Aller au crawler**. Les envois en échec restent listés, avec leur raison, et **Réessayer** les renvoie ; les autres sont déjà envoyés.

> **i** Note
>
> Les fichiers vont directement à la bibliothèque : un fichier marqué **Envoyé** est entre ses mains. Les URL passent par subscription, qui les remet à la bibliothèque par Kafka : une URL envoyée est entre les mains du courtier de messages, et la bibliothèque la télécharge ensuite. Voir Événements et appels.

Index de recherche

« Si les recherches dans la bibliothèque manquent des documents qu'elle contient, reconstruisez son index de recherche depuis la base de données. Cela couvre les documents communs et internes. »

Reconstruire l'index de recherche demande confirmation : « La bibliothèque relit chaque document commun et interne. Cela peut prendre plusieurs minutes, et les recherches peuvent manquer des documents jusqu'à la fin. ».

Pendant la reconstruction, la page affiche « Reconstruction : décompte des documents… », puis le nombre de documents traités sur le total, avec une barre de progression. Une fois terminée, elle indique la date de la dernière reconstruction et le nombre de documents indexés, ou la raison de son échec. La bibliothèque refuse une seconde reconstruction tant qu'une première tourne.

Référentiel

Le référentiel se trouve dans le menu **Configuration des services › Référentiel**.

La page **Cubes** gère les données de référence du service `referential` : « Données de référence lues par tous les tenants dans RegLab : statistiques de comparaison de la BCE, de l'ABE et d'autres sources. ». Ces données sont rangées en cubes : un cube est un flux de données SDMX (agence, identifiant, version), avec sa structure (ses dimensions), ses listes de codes et ses observations.

Comme toute la console, la page est réservée aux administrateurs de la plateforme.

La page compte deux onglets : **Cubes** et **Sources**.

Cubes

Importer un cube

Déposez un bundle dans la zone **Importer un cube**, ou cliquez sur **choisissez un zip**.

Un bundle est un fichier `.zip` qui contient :

- `structure.json` : les listes de codes, la structure de données et le flux de données ;
- `data.csv` : les observations, au format SDMX-CSV.

Le bundle est importé d'un bloc : tout ou rien. Un fichier qui n'est pas un zip, ou qui dépasse 512 Mo (le maximum que lit le référentiel), est refusé avant l'envoi.

Une fois l'import réussi, la page indique le cube alimenté, le nombre de séries et d'observations, les listes de codes créées et complétées, et « Nouveau cube. » quand le cube n'existait pas. En cas d'échec, rien n'a changé : la page donne la raison et les lignes en erreur des fichiers (les 50 premières, puis leur nombre).

La liste des cubes

Cubes du référentiel (n) liste les cubes, triés par nom ; **Actualiser** relit la liste. La recherche (**Chercher un cube, un identifiant ou une dimension**) filtre la liste.

Les colonnes sont :

- **Cube** : son nom et son identifiant `agence:identifiant(version)` ;
- **Dimensions** : celles de sa structure, dans l'ordre de la clé de séries ;
- **Agence**.

Chaque ligne propose :

- **Réimporter** : choisissez un nouveau bundle ; la page demande « Remplacer cube par fichier ? Ses données, sa structure et ses listes de codes deviennent celles du bundle (une liste de codes partagée avec un autre cube ne fait que gagner des codes). Tout ou rien. », puis **Remplacer** ;
- **Supprimer** : la page demande « Supprimer cube et toutes ses données ? Sa structure et ses listes de codes partent aussi, sauf si un autre cube les partage. Une source du chargeur qui l'alimente le recharge en entier à sa prochaine mise à jour. », puis **Supprimer**.

Sources

L'onglet **Sources** montre les cubes que le chargeur du référentiel tient à jour seul : « la BCE chaque jour ; le Risk Dashboard, le Pillar 3 Data Hub, l'exercice de transparence et le test de résistance de l'ABE, la liste des IFM de la BCE et le Webstat de la Banque de France chaque semaine. ».

La liste des sources

L'en-tête propose **Actualiser**, **Suivre un cube** et **Tout mettre à jour** (désactivé pendant une mise à jour).

Les colonnes sont :

- **Cube** : la source, l'identifiant suivi, **En pause** pour une entrée désactivée, et ce que l'entrée charge (clé de séries et période de départ pour la BCE, modèle, fichier ou jeu de données pour les autres) ;
- **Dernière mise à jour** : son état (**En cours**, **Réussie**, **Échouée**), sa date, sa durée et son déclencheur (**planifiée** ou **à la main**), le message d'erreur d'un échec, et la date du dernier succès quand la dernière mise à jour n'a pas réussi ; « Jamais mis à jour » sinon ;
- **Chargé** : le nombre de séries et d'observations de la dernière mise à jour réussie, et le fichier chargé.

Source	Ce qu'une entrée charge
BCE	Un cube du portail de données de la BCE, selon sa clé de séries.
Annexe ABE	Le Risk Dashboard de l'ABE : un cube par onglet de l'annexe, ou les paramètres de risque de crédit.
ABE Pilier 3	Un modèle du Pilier 3 ; en pause tant que la licence des publications des établissements n'est pas levée.
ABE exercice de transparence	Un fichier de l'exercice de transparence européen.
ABE test de résistance	Un fichier du test de résistance européen.
BCE liste des IFM	Toute la liste des institutions financières monétaires.
Banque de France Webstat	Un jeu de données Webstat ; en pause tant qu'aucune clé d'API Webstat n'est définie.

Chaque ligne propose :

- **Mises à jour passées** : déplie l'historique des mises à jour de l'entrée ;
- **Charger un fichier d'annexe (xlsx)** : pour les entrées **Annexe ABE** seulement ; un classeur `.xlsx` de 64 Mo au plus ;
- **Modifier** (voir ci-dessous) ;
- **Ne plus suivre** : pour une entrée **BCE** seulement, après confirmation : « Arrêter de mettre à jour cube ? Les données déjà chargées restent dans le référentiel. » ;
- **Mettre à jour** : lance la mise à jour de cette seule entrée.

Les boutons de chargement, d'abandon et de mise à jour sont désactivés pendant une mise à jour de l'entrée. Tant qu'une mise à jour tourne, la liste se relit toutes les 3 secondes.

Suivre ou modifier un cube

Suivre un cube ouvre **Suivre un cube de la BCE**. Seules les entrées de la BCE se créent ici ; les autres sont livrées avec le référentiel.

Champ	Obligatoire	Règle
Agence	oui	`ECB` par défaut ; un identifiant SDMX d'agence.
Flux de données	oui	L'identifiant du flux, par exemple `SUP`.
Version	non	Vide : la dernière. Sinon des nombres séparés par des points, par exemple `1.0`.
Clé de séries	oui	Codes joints par `+`, dimensions par un point, une partie vide pour tout code, par exemple `Q.FR+DE.W0.....`.
Depuis la période (facultatif)	non	Par exemple `2015-Q1`.

Champ	Obligatoire	Règle
Mis à jour chaque jour	—	Désactivé, l'entrée est en pause.

Un champ invalide est mis en évidence (« Vérifiez les champs en évidence. ») avant tout envoi.

Modifier ouvre le même formulaire. Pour une entrée de la BCE, tous les champs se changent ; pour les autres, seul l'interrupteur de mise à jour (**Mis à jour chaque semaine**) se change, et le formulaire explique ce que charge l'entrée.

IV Supervision

Activité Kafka DPM

L'activité Kafka DPM se trouve dans le menu **Supervision › Activité Kafka DPM**.

Le service `dpm` reçoit son travail par Kafka : chacun de ses listeners lit un topic avec un ou plusieurs threads consommateurs. La page montre, en direct, « ce que fait chaque thread consommateur Kafka de dpm en ce moment, pour quel tenant, et les messages qui attendent derrière ». Elle sert à voir si un tenant en fait attendre un autre, ou si un message est bloqué.

La page ne modifie rien : elle lit un instantané que dpm publie (voir Événements et appels). Comme toute la console, elle est réservée aux administrateurs de la plateforme, et dpm le vérifie lui-même : s'il refuse l'accès, la page affiche **Accès refusé** (« dpm ne montre son activité Kafka qu'aux administrateurs de la plateforme. »).

Lecture en direct

La page relit dpm toutes les 5 secondes tant que l'onglet du navigateur est visible. L'en-tête indique :

- **En direct · toutes les 5 s** ;
- **En pause tant que l'onglet est masqué** : aucune lecture tant que l'onglet est en arrière-plan ;
- **Périmé** : la dernière lecture a échoué ; la page affiche alors le dernier instantané reçu, avec l'heure de l'échec ;
- **Arrêté** : dpm a refusé l'accès ; les lectures s'arrêtent jusqu'à un clic sur **Actualiser**.

L'heure de l'instantané (« Instantané dpm de … ») s'affiche à côté ; **Actualiser** relit aussitôt.

Quand dpm ne répond pas, la page l'explique : « Ce dpm ne publie pas encore son activité Kafka. », « dpm est injoignable. » ou « dpm a répondu par une erreur (…). ».

Les totaux

Trois tuiles résument l'instantané :

- **Threads occupés** : les threads occupés sur le total, sur tous les listeners ;
- **Tenants servis** : les tenants pour lesquels un thread travaille en ce moment ;
- **Messages en attente** : les messages qui attendent, et sur combien de partitions.

Les listeners

Chaque listener a son bloc, titré par son topic et son identifiant, avec le nombre de threads, d'occupés et de messages en attente. Une bande de couleur montre ses threads : chaque tenant a sa teinte, la même pour ses threads et pour les partitions qu'il occupe. Quand dpm n'a pas pu lire le retard d'un listener, le bloc affiche « Retard indisponible : … ».

Le tableau des threads a ces colonnes :

- **Thread** ;
- **État** : **Occupé** ou **Libre** ;
- **Tenant** : le nom du tenant, lu dans la liste des tenants de subscription, ou son identifiant quand il n'y figure pas ;
- **Message** : le type du message traité ;
- **Tâche** : le début de l'identifiant de la tâche ; un bouton copie l'identifiant complet ;
- **Étape** : où en est le traitement ;
- **Durée** : depuis quand le thread est sur ce message ; elle est signalée au-delà d'1 minute, puis comme bloquée au-delà de 5 minutes ;
- **Partition** : la partition et l'offset du message.

Un thread libre affiche « En attente d'un message » et les partitions qui lui sont attribuées.

Les partitions

Sous le tableau, chaque partition du topic indique :

- le nombre de messages en attente ;
- l'offset commité et l'offset de fin ;
- qui l'occupe : « en file derrière tenant sur thread » quand des messages attendent, « thread sur tenant » sinon, ou « Aucun message en cours ».

> Astuce
>
> Les messages en attente d'une partition sont en file derrière le message que traite le thread qui l'occupe. Une partition avec des messages en attente, occupée par un thread dont la **Durée** est signalée, montre donc un message long qui en retient d'autres, éventuellement d'autres tenants.

V Fonctionnement

Identités et realms

Cette page explique, pour les lecteurs techniques, comment subscription et Keycloak se partagent les identités : les realms, ce que chacun garde, et comment un utilisateur est créé, désactivé et supprimé. Les écrans sont décrits dans Utilisateurs.

Les realms

Un **realm** Keycloak est un annuaire de comptes, avec ses propres règles de connexion. La plateforme en accepte plusieurs :

- `security.jwt.allowed-realms` : les realms dont les services acceptent les jetons. Ce sont les seuls realms qu'un utilisateur ou un client peut recevoir ; `GET /api/v1/realms` les donne aux écrans, avec le realm par défaut.
- `application.identity.default-realm` (`mla` par défaut) : le realm d'un utilisateur créé sans realm, et celui d'une recherche d'appartenance qui n'en nomme pas. Il doit faire partie des realms acceptés, sinon subscription ne démarre pas.
- `security.jwt.platform-realms` (`[mla, shared]` par défaut, réglage de security-library) : les realms dont un jeton peut être celui d'un administrateur de la plateforme ou d'un service. Les administrateurs viennent de `mla`, les comptes de service de `shared`.

Qui ouvre la console

La console se connecte par la passerelle, par défaut sur le realm `mla`. Elle lit le compte connecté sur la passerelle (`/api/account`) et n'ouvre ses pages qu'à un compte qui porte l'autorité `ROLE_ADMIN` ; un autre compte voit la page **Accès restreint** (« Subscription est réservé aux administrateurs MLAdvisory, et votre compte n'a pas ce rôle. »), qui ne propose que **Se déconnecter**.

Le serveur fait sa propre vérification : un administrateur de la plateforme est un jeton émis par un realm de la plateforme et qui porte exactement le rôle de realm `ROLE_ADMIN` (`ReglabAuthorities.isPlatformAdmin`). Tout endpoint de subscription lui est réservé, sauf ceux que listent Les droits et La sécurité : les recherches des services, la lecture publique d'un client par son sous-domaine, la langue de l'utilisateur connecté, etc.

Le rôle **Administrateur** qu'un utilisateur a sur un tenant est une donnée de subscription, pas un rôle Keycloak : il ne donne pas accès à la console.

Ce que garde subscription, ce que garde Keycloak

subscription garde (`reglab_user` et les accès)	Keycloak garde (le compte du realm)
Le realm et l'identifiant du compte	Le compte lui-même, son mot de passe et sa façon de se connecter
Prénom, nom, e-mail	Le fait que le compte soit activé ou non
Le client d'origine, **Multi-client**, les clients invités	Les sessions ouvertes
La langue des mails (propriété `language`)	Les rôles de realm (`ROLE_ADMIN`, `SUPPORT`…)
La date de désactivation (`disabledSince`)	Les événements de connexion
La dernière connexion vue (`lastLoginAt`), recopiée de Keycloak	

subscription garde (`reglab_user` et les accès)	Keycloak garde (le compte du realm)
Les accès : rôle et plans sur chaque tenant, suspension	

subscription ne crée ni ne supprime jamais de compte Keycloak, et ne touche ni à son mot de passe ni à ses rôles. Il ne fait qu'une chose sur un compte : le désactiver (en fermant ses sessions) ou le réactiver (voir La désactivation). Il lit aussi les connexions des realms (voir La dernière connexion).

Un utilisateur : un realm et un identifiant

Un identifiant n'est unique qu'à l'intérieur de son realm : un utilisateur est la paire (realm, identifiant). La table `reglab_user` porte une contrainte d'unicité sur cette paire. L'e-mail, lui, est unique sur toute la plateforme.

Un jeton est rattaché à son utilisateur par son `preferred_username` et par le realm de son émetteur (`iss`, `…/realms/<realm>`). Les services passent ce realm à subscription dans leurs recherches d'appartenance ; un service qui n'en passe pas obtient le realm par défaut.

Le realm d'un client

Un client peut avoir un realm (`client.realm`) et un sous-domaine (`client.subdomain`). Ensemble, ils disent où se connectent ses utilisateurs :

- Le front d'un client répond sur `<sous-domaine>.irrc-<environnement>.…`. Avant toute connexion, il lit `GET /api/v1/public/clients/by-subdomain/{sous-domaine}`, ouvert à tous, qui donne le realm où se connecter, le nom du client et s'il a un logo. Cette lecture ne répond que pour un client qui a un sous-domaine et un realm encore accepté (404 sinon) ; les caches peuvent la garder 5 minutes.
- Le realm d'un client n'a pas de valeur par défaut : vide, il reste non renseigné.
- Les comptes des utilisateurs d'un client qui a un realm doivent être dans ce realm. subscription refuse de créer, de corriger ou d'importer un utilisateur dans un autre realm, et refuse de donner à un client un realm alors que certains de ses utilisateurs sont ailleurs (`error.client.usersOutsideRealm`). Les clients invités ne comptent pas : seul le client d'origine décide.
- Un realm n'est vérifié qu'à l'écriture. Retirer un realm de `security.jwt.allowed-realms` laisse ses clients pointer dessus : au démarrage, subscription journalise les clients dont le realm n'est plus accepté (`ClientRealmCheck`). La lecture publique ne les trouve plus, et leurs utilisateurs ne peuvent plus se connecter tant que le realm n'est pas de nouveau accepté ou que le client n'en a pas reçu un autre.

La création et l'import

Créer un utilisateur (`POST /api/v1/reglab-users`) ou en importer (`POST /api/v1/reglab-users/import`) crée sa fiche dans subscription, et rien dans Keycloak.

Le realm de l'utilisateur est, dans l'ordre : celui que la demande nomme, sinon celui de son client, sinon le realm par défaut. Il doit être accepté (400 sinon) et, quand le client en a un, être le sien. L'écran de création et l'import proposent le realm par défaut déjà choisi : pour un client qui a son propre realm, c'est celui-là qu'il faut choisir.

En modification, le realm peut être corrigé, pour un compte enregistré sous le mauvais realm, mais jamais vers un realm refusé. L'écran ne permet plus de changer le client d'origine après la création.

La désactivation

`PUT /api/v1/reglab-users/{id}/deactivation` désactive un utilisateur ; `DELETE` sur la même adresse le réactive. Dans la même transaction, la désactivation :

1. renseigne `disabledSince` : dès lors, les recherches d'appartenance ne trouvent plus l'utilisateur dans aucun tenant ;
2. garde ses rôles et ses plans, que la réactivation rend tels quels ;
3. envoie `UPDATE_PLAN` à irrc pour chacun de ses tenants ;
4. enregistre l'événement **Compte désactivé** dans l'historique des accès ;
5. ajoute une tâche de compte Keycloak `DISABLE` (`ENABLE` pour une réactivation) dans `keycloak_account_task`.

Désactiver un utilisateur déjà désactivé, ou réactiver un utilisateur actif, ne fait rien.

La tâche est faite par `KeycloakAccountJob`, chaque minute :

- Un compte n'a qu'une tâche : une nouvelle remplace celle en attente. Désactiver puis réactiver avant le passage de la tâche ne laisse que la réactivation.
- La tâche cherche le compte dans le realm de l'utilisateur par son identifiant, sinon par son e-mail.
- `DISABLE` désactive le compte et ferme ses sessions ouvertes ; `ENABLE` le réactive. Le compte n'est jamais supprimé.
- Sans compte correspondant (l'utilisateur ne s'est jamais connecté), la tâche est considérée comme faite.
- Deux comptes correspondants font échouer la tâche : elle attend qu'une personne règle le doublon.
- Une tâche qui échoue garde son erreur et est retentée après une attente qui double à chaque échec, d'une minute jusqu'à `application.keycloak-accounts.max-retry-delay` (6 heures).
- Chaque tâche est traitée dans sa propre transaction, sous verrou `FOR UPDATE SKIP LOCKED` : un seul réplica s'en charge.

Les tâches en attente et en échec s'affichent en haut de la liste des utilisateurs (`GET /api/v1/reglab-users/account-tasks`, voir Les modifications de compte Keycloak).

subscription appelle l'API d'administration de Keycloak avec, dans chaque realm, un client confidentiel `subscription-provisioning` (`application.keycloak-accounts.client-id`) dont le compte de service a les rôles `view-users` et `manage-users` de `realm-management`. Son secret, par realm, vient de Vault (`application.keycloak-accounts.client-secrets`). Un realm sans secret laisse ses tâches en échec. Sans `application.keycloak-accounts.server-url`, le job ne fait rien (un avertissement au démarrage) et les tâches attendent.

La suppression

`DELETE /api/v1/reglab-users/{id}` est refusé (403, `error.user.hasAccesses`) tant que l'utilisateur a un accès à un tenant. Sinon, dans la même transaction, la suppression :

- ajoute une tâche `DISABLE` pour son compte Keycloak, qui est désactivé mais pas supprimé ;
- enregistre **Compte désactivé**, avec la cause « car l'utilisateur a été supprimé », sauf s'il était déjà désactivé ;
- supprime sa fiche.

La dernière connexion

`LastLoginJob` lit toutes les 15 minutes (`application.keycloak-accounts.login-cron`) les événements `LOGIN` de chaque realm qui a un secret configuré, depuis sa dernière lecture. La première lecture d'un realm remonte à `login-history` (90 jours), dans la limite de ce que Keycloak garde.

- L'utilisateur est retrouvé par realm et identifiant, sinon par e-mail, sans tenir compte de la casse.
- Une connexion ouverte par une prise d'identité du support n'est pas une connexion de l'utilisateur : elle est ignorée.
- `lastLoginAt` ne fait qu'avancer.

Il faut que le realm enregistre ses événements (Realm settings › Events › Save events) et que le compte de service du client de provisionnement ait `view-events`. « Jamais » dans les écrans veut donc dire « aucune connexion vue depuis le début de l'enregistrement », pas forcément « jamais connecté ».

Les recherches d'appartenance des services

Les services demandent à subscription les accès d'un utilisateur :

- `GET /api/v1/user-tenants/{service}/{identifiant}?realm=` : ses accès pour un service. Un utilisateur final ne lit que les siens, dans le realm de son jeton ; les services du groupe `user-tenants` et les administrateurs de la plateforme lisent ceux de n'importe qui, dans le realm demandé ou le realm par défaut.
- `GET /api/v1/user-tenants/{tenantId}/{service}/{identifiant}?realm=` : son accès à un tenant pour un service, que les services vérifient avant de servir l'utilisateur.

Ces recherches ne renvoient que les accès en vigueur : voir Ce qui donne accès à un tenant. Les droits que portent les plans sont décrits dans Les droits.

Chaque utilisateur, quel qu'il soit, lit et change aussi sa propre langue de mails (`GET` et `PUT /api/v1/reglab-users/me/language`, `fr` ou `en`).

Paramètres

Propriété	Défaut	Rôle
`security.jwt.allowed-realms`		Realms acceptés.
`application.identity.default-realm`	`mla`	Realm par défaut.
`application.keycloak-accounts.server-url`		Adresse de Keycloak ; sans elle, aucune tâche n'est faite.
`application.keycloak-accounts.client-id`	`subscription-provisioning`	Client de provisionnement, le même dans chaque realm.
`application.keycloak-accounts.client-secrets`		Secret du client, par realm (Vault).
`application.keycloak-accounts.cron`	chaque minute	Passage des tâches de compte ; `-` le désactive.
`application.keycloak-accounts.max-retry-delay`	6 heures	Attente maximale entre deux essais d'une tâche.
`application.keycloak-accounts.timeout`	10 secondes	Durée maximale d'un appel à Keycloak.
`application.keycloak-accounts.login-cron`	toutes les 15 minutes	Lecture des connexions ; `-` la désactive.
`application.keycloak-accounts.login-history`	90 jours	Profondeur de la première lecture des connexions d'un realm.

Des plans aux droits

Cette page explique, pour les lecteurs techniques, comment les plans qu'un utilisateur détient dans un tenant deviennent des droits dans les autres services de la plateforme : la question que ces services posent à subscription, le jeton avec lequel ils la posent, la façon dont ils gardent la réponse en mémoire, et l'événement `UPDATE_PLAN` qui leur dit de l'oublier.

Le principe

subscription ne pousse pas les droits vers les services : chaque service les lui demande.

1. Un utilisateur appelle un service (CIRCÉ, l'assistant…) avec son jeton Keycloak et nomme le tenant dans l'en-tête `X-TENANT-ID`.
2. Le service demande à subscription l'accès de cet utilisateur, dans ce tenant, pour ce service : `GET /api/v1/user-tenants/{tenantId}/{serviceName}/{userName}?realm=…`, avec son propre jeton de service.
3. subscription répond le plan que l'utilisateur détient pour ce service, avec ses fonctionnalités, et son rôle dans le tenant ; ou `404` s'il n'en détient aucun qui donne accès.
4. Le service décide : l'utilisateur est membre si le plan porte la fonctionnalité attendue (`base` pour CIRCÉ, `assistant` pour l'assistant) ; chaque écran ou action vérifie ensuite sa propre fonctionnalité.
5. Le service garde la réponse en mémoire quelques minutes. Quand un droit change, subscription publie `UPDATE_PLAN` pour que les services l'oublient.

Ce que la recherche répond

Le point d'accès `GET /api/v1/user-tenants/{tenantId}/{serviceName}/{userName}` cherche l'accès de l'utilisateur au tenant pour le service :

- `{tenantId}` est l'identifiant public du tenant (`tenantId`), le même que celui de l'en-tête `X-TENANT-ID` ;
- `{serviceName}` est le nom du service, comparé sans tenir compte des majuscules (`irrc`, `stress`, `library`…) ;
- `{userName}` est l'identifiant de l'utilisateur, et le paramètre `realm` le realm qui a émis son jeton. Sans `realm`, subscription prend le realm par défaut (`application.identity.default-realm`) ; un realm que subscription n'accepte pas est une erreur `400`.

subscription ne répond un accès que si toutes ces conditions sont réunies :

Condition	Sinon
L'utilisateur détient un plan du service dans le tenant.	`404`
Son accès au tenant n'est pas suspendu.	`404`
Son compte n'est pas désactivé.	`404`
Le tenant est `ACTIVE`.	`404`
L'instant présent est dans la période de validité du tenant (début inclus, fin exclue).	`404`

La validité est comparée au moment de chaque recherche : un contrat qui arrive à échéance ferme l'accès sans aucune action ni aucun message.

La réponse contient l'identifiant de l'accès, l'utilisateur, le tenant (sans son statut ni ses dates), le **plan** avec ses **fonctionnalités**, et le **rôle** de l'utilisateur dans le tenant (`MEMBER` ou `ADMIN`). Comme un utilisateur détient au plus un plan par service dans un tenant, il y a au plus une réponse.

Un second point d'accès, `GET /api/v1/user-tenants/{serviceName}/{userName}?realm=…`, renvoie les accès de l'utilisateur au service dans tous les tenants qui donnent accès, avec les mêmes conditions (`404` quand il n'y en a aucun). CIRCÉ s'en sert pour lister les tenants qu'un utilisateur peut nommer.

Qui peut interroger subscription

Les services interrogent subscription avec leur propre jeton : un jeton *client credentials* émis par un realm de la plateforme (`mla` ou `shared` par défaut, `security.jwt.platform-realms`). Un jeton émis par le realm d'un client n'est jamais un jeton de service.

Point d'accès	Appelants admis
`GET /api/v1/user-tenants/{tenantId}/{serviceName}/{userName}`	les administrateurs de la plateforme ; les services du groupe `user-tenants`
`GET /api/v1/user-tenants/{serviceName}/{userName}`	les mêmes, et l'utilisateur lui-même, pour son propre nom et dans son propre realm

Un service fait partie du groupe `user-tenants` quand le client Keycloak de son jeton est listé dans `security.service-callers.groups.user-tenants`. Tant que `security.service-callers.legacy-shared-client` vaut `true` (la valeur par défaut), le client partagé `internal_service_client` est admis aussi.

Côté services :

- CIRCÉ (`irrc`) appelle subscription par un client Feign nommé `subscription`, trouvé par la découverte de services ;
- l'assistant appelle subscription avec le `ServiceTokenClient` de security-library (`application.s2s.*`), dont les identifiants viennent du secret applicatif partagé de Vault.

L'en-tête X-TENANT-ID et tenant-web-library

Un appel à un service multi-tenant nomme son tenant dans l'en-tête `X-TENANT-ID`, qui porte l'identifiant public du tenant. La bibliothèque partagée tenant-web-library fournit le filtre qui le lit et vérifie l'appartenance de l'appelant ; l'assistant l'utilise. CIRCÉ a son propre filtre, qui suit les mêmes règles (voir CIRCÉ).

Le filtre de tenant-web-library (`TenantBindingFilter`) s'exécute après la chaîne Spring Security, sur les chemins couverts (`/api/**` par défaut, moins les chemins exclus) :

1. Il lit `X-TENANT-ID`. Un en-tête absent ou vide est refusé en `400` (`error.tenant.missing`), une valeur qui n'est pas un UUID canonique aussi (`error.tenant.invalid`). En mode mono-tenant, l'en-tête est ignoré et le tenant configuré est utilisé.
2. Il demande au service si l'appelant est membre du tenant (`TenantMembershipVerifier`, que chaque service fournit). Un appelant anonyme ou non membre reçoit un `403` (`error.tenant.forbidden`), identique pour un tenant inconnu et un tenant interdit. Une recherche qui échoue donne un `500` (`error.tenant.lookup-failed`) et ne laisse rien passer. Un appelant que le service déclare de confiance (`TrustedCallerPolicy`) passe sans vérification ; par défaut, personne.
3. Il exécute la requête dans le tenant, qu'il inscrit aussi dans le contexte des logs.

Sur un chemin déclaré à tenant facultatif (`optional-tenant-paths`), une requête sans en-tête s'exécute sans tenant.

Le cache des appartenances

Le filtre garde les réponses dans un cache (`TenantMembershipCache`), par tenant, realm et nom d'utilisateur :

Propriété	Défaut	Rôle
`reglab.tenant.membership-cache.positive-ttl`	`5m`	Durée pendant laquelle une appartenance est crue : ce qui borne l'accès gardé par un utilisateur retiré si l'éviction est perdue.

Propriété	Défaut	Rôle
`reglab.tenant.membership-cache.negative-ttl`	`30s`	Durée pendant laquelle un refus est cru : ce qu'un utilisateur qui vient de recevoir un accès peut avoir à attendre.
`reglab.tenant.membership-cache.maximum-size`	`10000`	Nombre maximal de réponses gardées.

Les appels simultanés pour la même clé attendent une seule recherche ; une erreur n'est jamais gardée. Le service vide le cache avec `evict(tenant)`, `evict(tenant, callerName)` ou `evictAll()`, en particulier à la réception d'`UPDATE_PLAN`.

CIRCÉ (irrc)

CIRCÉ demande les accès du service `irrc`.

- **Appartenance.** Un utilisateur est membre d'un tenant si subscription lui répond un plan `irrc` qui porte la fonctionnalité `base`.
- **Administrateur du tenant.** Le rôle `ADMIN` de la réponse fait de l'utilisateur l'administrateur de ce tenant dans CIRCÉ, quel que soit son rôle dans ses autres tenants.
- **Fonctionnalités.** Les écrans liés à une offre demandent en plus une fonctionnalité du plan ; un membre dont le plan ne la porte pas reçoit un `403` (voir Les noms de fonctionnalités lus).
- **Cache.** CIRCÉ garde chaque réponse 5 minutes, et un refus 30 secondes ; ces durées sont fixées dans son code (`CacheConfiguration`). Le cache est propre à chaque instance.
- **Comptes de service.** Les services du groupe `internal` (sur les routes `/api/v1/internal/**` seulement) passent sans recherche.

L'assistant

L'assistant s'ouvre à un utilisateur dans un tenant si son plan, dans ce tenant, d'un des services de `application.subscription.plan-services` (`irrc`, `stress` et `library` par défaut) porte la fonctionnalité `assistant` (comparée sans tenir compte des majuscules). Il n'offre les outils d'une application que si le plan de cette application ouvre l'assistant.

- L'assistant pose la question pour chacun de ces services, avec le realm du jeton de l'utilisateur ; un jeton qui ne nomme aucun realm ne fait pas un membre.
- Il garde les fonctionnalités de chaque plan, par tenant, realm, utilisateur et service : pendant `membership-cache-ttl` (une heure par défaut) quand elles ouvrent l'assistant, pendant le `negative-ttl` de tenant-web-library (30 secondes par défaut) sinon.
- Le filtre de tenant-web-library garde en plus sa réponse : une appartenance pendant `membership-cache-ttl`, un refus pendant le `negative-ttl`.
- Aucun appelant n'est de confiance : l'assistant agit toujours pour un utilisateur.

L'événement UPDATE_PLAN

Quand un droit change, subscription publie un message sur le topic Kafka `subscription_irrc_topic` :

```
{"tenantId": "<identifiant public du tenant>", "messageType": "UPDATE_PLAN",
"schemaVersion": 1}
```

- Le message nomme un tenant, jamais un utilisateur ; sa clé Kafka est le `tenantId`, ce qui garde dans l'ordre les messages d'un même tenant. Les champs sont lus par leur nom : ils font partie du contrat entre services.
- Il part dans la transaction du changement, une fois le changement écrit, et subscription attend l'accusé de réception du broker (délai `application.kafka.send-timeout`). Un envoi qui échoue annule le changement et fait échouer la demande : un retrait de droit n'est jamais enregistré sans que les services soient prévenus.
- Le message part juste avant la validation de la transaction : une recherche faite dans ces quelques millisecondes lit encore l'ancien droit et le garde jusqu'à son expiration.

À la réception :

- **CIRCÉ** (`MembershipConsumer`) oublie les appartenances et les refus gardés pour ce tenant ; un message sans tenant lisible vide tout le cache. Chaque instance lit tous les messages, dans un groupe de consommateurs qui lui est propre (`irrc-membership-<uuid>`).
- **L'assistant** (`MembershipConsumer`) oublie les plans gardés pour ce tenant, puis les appartenances du filtre ; un message sans tenant lisible vide les deux caches. Chaque instance a son groupe (`assistant-membership-<uuid>`) et commence au dernier message ; un message illisible part dans `subscription_irrc_topic.DLT`.

Une instance sans broker, ou qui n'a pas encore rejoint son groupe, ne lit rien : seule l'expiration des caches s'applique alors.

Quand subscription envoie UPDATE_PLAN

Changement	Origine dans la console
Le rôle et les plans d'un utilisateur dans un tenant sont remplacés (`PUT /api/v1/tenants/{id}/members/{userId}`)	le formulaire d'accès d'un utilisateur à un tenant
Le plan ou le rôle d'un accès change (`PUT /api/v1/user-tenants/{id}`)	—
Un plan est retiré à un utilisateur (`DELETE /api/v1/user-tenants/{id}`) ; le dernier plan retire l'utilisateur du tenant	—
Un utilisateur est retiré d'un tenant (`DELETE /api/v1/tenants/{id}/members/{userId}`)	**Retirer du tenant**
Un utilisateur cesse d'être invité d'un client : il est retiré de tous les tenants de ce client	la carte **Clients invités** d'un utilisateur
L'accès d'un utilisateur à un tenant est suspendu ou réactivé	**Suspendre l'accès**, **Réactiver l'accès**
Un utilisateur est désactivé ou réactivé : un message par tenant où il détient un plan	Utilisateurs
Le statut ou la validité d'un tenant change : suspension, réactivation, période de validité, suppression programmée, restauration, purge	Clients et tenants

À la purge, subscription publie en plus `TENANT_DELETED` sur `tenant_lifecycle_topic` (voir Le cycle de vie d'un tenant).

Rien n'est envoyé quand le changement ne change rien : suspendre un accès déjà suspendu, réactiver un accès actif, retirer d'un tenant un utilisateur qui n'y détient rien.

Ce que subscription ne signale pas

Dans ces cas, aucun `UPDATE_PLAN` ne part : seule l'expiration des caches des services s'applique.

Changement	Effet dans les services
La définition d'un plan change : ses fonctionnalités, son service, son nom (page Plans)	Les services appliquent l'ancienne définition jusqu'à l'expiration de leur cache : jusqu'à 5 minutes pour CIRCÉ ; pour l'assistant, `assistant` retiré d'un plan reste ouvert jusqu'à `membership-cache-ttl` (une heure par défaut). Les détenteurs du plan reçoivent un e-mail.
Une fonctionnalité est renommée (page Fonctionnalités)	Comme pour un plan modifié.

Changement	Effet dans les services
Un accès est créé seul (`POST /api/v1/user-tenants`)	Le droit gagné attend l'expiration du refus gardé : jusqu'à 30 secondes. Le formulaire d'accès de la console passe par le remplacement, qui envoie le message.

> **Note**
>
> Un retrait de droit qui passe par la console est signalé ; un gain de droit qui ne l'est pas n'attend que l'expiration d'un refus (30 secondes). C'est une modification du catalogue (plans, fonctionnalités) qui peut laisser un droit retiré actif le plus longtemps.

Les délais en résumé

Situation	CIRCÉ	Assistant
Droit retiré, `UPDATE_PLAN` reçu	à la requête suivante	à la requête suivante
Droit retiré, `UPDATE_PLAN` perdu	5 minutes au plus	`membership-cache-ttl` au plus (une heure par défaut)
Droit donné après un refus gardé, sans `UPDATE_PLAN`	30 secondes au plus	`negative-ttl` au plus (30 secondes par défaut)
Contrat du tenant arrivé à échéance	dès l'expiration de la réponse gardée (5 minutes au plus)	dès l'expiration de la réponse gardée

Les durées de CIRCÉ sont fixées dans son code ; celles de l'assistant sont des propriétés, servies par Consul.

Les noms de fonctionnalités lus

Une fonctionnalité n'ouvre quelque chose que si un service lit son nom. Les noms lus aujourd'hui :

Fonctionnalité	Plan du service	Ce qu'elle ouvre
`base`	tout service	L'appartenance au service ; dans CIRCÉ, l'accès au tenant.
`anacredit`	`irrc`	Les écrans AnaCredit de CIRCÉ.
`pilar3`	`irrc`	Les écrans Pilier 3 de CIRCÉ.
`ste`	`irrc`	Les packages STE de la BCE (SREP) et leurs contrôles.
`stress`	`irrc`	Les stress tests de CIRCÉ.
`generate-dpm`	`irrc`	La génération de packages XBRL-CSV à partir d'un classeur.
`mcp`	`irrc`	Les outils MCP de CIRCÉ pour un client MCP.
`assistant`	`irrc`	Les outils MCP de CIRCÉ quand c'est l'assistant qui appelle pour l'utilisateur.
`assistant`	`irrc`, `stress` ou `library`	L'assistant lui-même, et les outils de cette application dans l'assistant.

Les noms sont comparés tels quels par CIRCÉ, sans tenir compte des majuscules par l'assistant.

Le cycle de vie d'un tenant

Cette page explique, pour les lecteurs techniques, la vie d'un tenant dans subscription, de sa création à sa purge, et ce que chaque service en fait. Les écrans sont décrits dans Clients et tenants et Purge.

Les statuts

Un tenant (`Tenant`, table `tenant`) a un statut, `TenantStatus` :

Statut	Sens	Accès
`ACTIVE`	Le statut normal.	Oui, dans la fenêtre de validité.
`SUSPENDED`	Suspendu par un administrateur de la plateforme : tout est conservé.	Non, jusqu'à la réactivation.
`PENDING_DELETION`	Suppression demandée : tout est conservé, le tenant reste restaurable jusqu'à sa date de purge.	Non.
`DELETED`	Purgé : membres retirés ; la ligne et l'identifiant de connexion sont gardés, le nom et l'identifiant restent réservés. Statut final.	Non.

Les transitions :

Depuis	Action	Vers	Endpoint
`ACTIVE`	Suspendre	`SUSPENDED`	`PUT /api/v1/tenants/{id}/suspension`
`SUSPENDED`	Réactiver	`ACTIVE`	`DELETE /api/v1/tenants/{id}/suspension`
`ACTIVE`, `SUSPENDED`	Programmer la suppression	`PENDING_DELETION`	`DELETE /api/v1/tenants/{id}` (202)
`PENDING_DELETION`	Restaurer	`SUSPENDED` s'il l'était avant, sinon `ACTIVE`	`POST /api/v1/tenants/{id}/restore`
`PENDING_DELETION`	Purge, une fois la date passée	`DELETED`	`TenantPurgeJob`

Le serveur répond 409 à une action que le statut ne permet pas : suspendre ou réactiver un tenant `PENDING_DELETION` (`error.tenant.pendingDeletion`), restaurer un tenant qui n'est pas `PENDING_DELETION` (`error.tenant.notPendingDeletion`), modifier quoi que ce soit d'un tenant `DELETED` (`error.tenant.deleted`). Suspendre un tenant déjà suspendu, réactiver un tenant actif, ou programmer la suppression d'un tenant déjà en attente ne change rien ; dans ce dernier cas, il garde sa date de purge.

La date de suspension (`suspendedSince`) survit à une demande de suppression : c'est elle qui permet à la restauration de rendre le statut `SUSPENDED`.

La validité

Indépendamment du statut, un tenant a une fenêtre de validité, `validFrom` (inclus) et `validUntil` (exclu). Une borne vide est sans limite. `PUT /api/v1/tenants/{id}/validity` remplace la fenêtre entière ; une fin qui ne vient pas après le début est refusée (400). La validité se change à tout statut sauf `DELETED`.

Le statut reste `ACTIVE` quand la fenêtre est finie ou pas encore commencée : c'est la lecture des accès qui compare l'instant présent à la fenêtre, à chaque lecture. Rien ne se passe donc à l'instant où un contrat échoit, sinon que plus aucun accès n'est trouvé.

Le dernier jour d'accès est le jour, à l'heure de Paris, de l'instant qui précède `validUntil`. Un `validUntil` au 1er janvier à 00:00 donne donc le 31 décembre comme dernier jour.

Ce qui donne accès à un tenant

Les services demandent à subscription les accès d'un utilisateur (voir Les recherches d'appartenance des services). Un accès n'y figure que si, au moment de la lecture :

- le tenant est `ACTIVE` ;
- l'instant est dans sa fenêtre de validité ;
- l'accès de l'utilisateur à ce tenant n'est pas suspendu ;
- l'utilisateur n'est pas désactivé.

Donner ou modifier un accès n'est permis que dans un tenant `ACTIVE` (409 `error.tenant.inactive` sinon) ; retirer un accès reste permis à tout statut. La fenêtre de validité n'est pas regardée : un accès peut se préparer avant le début du contrat.

La création

`POST /api/v1/tenants` crée le tenant :

- avec un identifiant public tiré au hasard, `tenantId` : c'est la **Référence** des écrans, celle que stockent les services, distincte de la clé primaire de subscription ;
- avec un nom unique, y compris parmi les tenants `DELETED` ;
- rattaché à un client, obligatoire ;
- au statut `ACTIVE`, sans fenêtre de validité ;
- avec son identifiant de connexion (`TenantCredential`).

L'identifiant de connexion est un mot de passe, le secret de chiffrement du tenant. Le seul type accepté est `PASSWORD` : `KEY` est refusé (400), aucun service ne chiffrant encore avec une paire de clés. Le mot de passe ne peut être vide ni plus court que `application.credential.minimum-password-length`. subscription lui ajoute un sel tiré au hasard. Le mot de passe et le sel sont chiffrés en base avec la clé maître courante ; un redémarrage rechiffre avec la clé courante les valeurs restées sur une ancienne clé (`TenantCredentialReencryption`).

`GET /api/v1/security/password/generate` fournit le mot de passe généré que propose le formulaire. Aucun écran ne relit l'identifiant de connexion : seul `GET /api/v1/tenant-credentials?tenantId=` le sert, aux services du groupe `tenant-credentials` (irrc) et à eux seuls, jamais au jeton d'un utilisateur, administrateurs compris. Chaque lecture est journalisée avec l'appelant et le tenant, jamais le secret. Il reste lisible après la purge.

Suspension et réactivation

La suspension coupe l'accès de tous les utilisateurs du tenant et ne retire rien. La réactivation le rend. Les deux préviennent irrc (voir Les messages UPDATE_PLAN) et écrivent aux membres (voir Les e-mails aux membres).

Suppression et délai d'attente

Programmer la suppression passe le tenant en `PENDING_DELETION`, que le tenant ait des membres ou non :

- `deletionRequestedAt` reçoit l'instant de la demande ;
- `deletionScheduledAt` reçoit la date de purge : la demande plus le délai d'attente, `application.tenant-lifecycle.deletion-grace-period`, 60 jours par défaut.

Pendant ce délai, aucun accès n'est trouvé, tout est conservé et **Restaurer** rend au tenant son statut d'avant. La restauration efface les deux dates.

> **Note**
>
> Les écrans annoncent « 60 jours » en dur, dans la confirmation et dans le calcul de la date de restauration. Si le délai d'attente est configuré autrement, la date affichée dans la confirmation sera fausse ; la date réelle est celle que la pastille et la page du tenant affichent ensuite, lue du serveur.

La purge

`TenantPurgeJob` tourne toutes les 15 minutes (`application.tenant-lifecycle.purge-cron`). Il purge les tenants dus : `PENDING_DELETION` dont la date de purge est passée, ou tenants de test dont la durée de vie est écoulée (voir Les tenants de test).

Chaque tenant est purgé dans sa propre transaction, sous un verrou de ligne `FOR UPDATE SKIP LOCKED` : de plusieurs réplicas, un seul purge un tenant donné, et un tenant restauré entre-temps est sauté. La purge :

1. supprime tous les accès du tenant (`user_tenant`) et ses membres (`tenant_member`) ;
2. enregistre chaque droit retiré dans l'historique des accès, avec la cause « parce que le tenant a été purgé » (`TENANT_PURGED`) ;
3. passe le tenant en `DELETED` avec `deletedAt` ;
4. envoie `UPDATE_PLAN` à irrc ;
5. envoie `TENANT_DELETED` sur `tenant_lifecycle_topic`.

Chaque envoi attend l'accusé de réception du broker. Un envoi qui échoue annule toute la purge du tenant, retentée au passage suivant. La purge n'écrit à personne : les membres ont été prévenus quand la suppression a été programmée.

Le tenant `DELETED` garde sa ligne et son identifiant de connexion ; son nom et son `tenantId` restent réservés.

L'événement TENANT_DELETED

`TENANT_DELETED` est publié sur `tenant_lifecycle_topic`, avec le `tenantId` comme clé, pour que les messages d'un même tenant restent dans l'ordre :

```
{ "tenantId": "12359bb8-b120-11ed-afa1-0242ac120005", "messageType": "TENANT_DELETED",
  "deletedAt": "2026-11-04T03:30:00.123456Z", "schemaVersion": 1 }
```

- `tenantId` est l'identifiant public du tenant, jamais la clé primaire de subscription.
- Le message est envoyé une fois, quand la purge se valide, mais livré au moins une fois : un consommateur doit être idempotent.
- Un consommateur ignore un `messageType` qu'il ne connaît pas.
- Le message est envoyé avant que la transaction de purge ne soit validée : pendant quelques millisecondes, un service qui lit le statut peut encore voir le tenant non `DELETED`.

Un service peut vérifier le statut d'un tenant par son identifiant public avec `GET /api/v1/tenant-status?tenantId=`, réservé aux services du groupe `tenant-status` et aux administrateurs de la plateforme. La réponse donne `status`, `deletedAt` et `purgeDue` (le tenant serait purgé maintenant). Un `tenantId` inconnu répond 404 : le tenant n'a jamais existé, ou c'était un tenant de test effacé après sa purge.

Ce que chaque service purge

Service	Ce qu'il supprime à la réception de `TENANT_DELETED`	Particularités
irrc (MyReg)	Les objets du tenant dans S3, puis les lignes de toutes les tables qui ont une colonne `tenant_id` (lues dans le cata-	Une transaction pour les lignes. Un seul réplica purge ; un échec est retenté, puis mis dans

Service	Ce qu'il supprime à la réception de `TENANT_DELETED`	Particularités
	logue, enfants d'abord), puis ce que les caches de la réplique en gardent.	`tenant_lifecycle_topic.DLT`. Aucun tenant n'est exempté.
library (bibliothèque)	Les documents propres au tenant et leurs fichiers, tâches, résumés, marqueurs de conversion docling, ses passages dans l'index, et ses utilisateurs. La bibliothèque commune et les autres tenants ne sont pas touchés.	Par lots, chacun validé à part. Échec retenté, puis `tenant_lifecycle_topic.DLT`.
ALM (stress, risk-app)	Toutes les lignes et tous les objets du tenant ; chaque réplica vide aussi ses caches locaux du tenant.	Ne purge qu'après avoir vérifié auprès de subscription que le tenant est `DELETED` (`GET /api/v1/tenant-status`). Ne purge jamais son tenant par défaut. Actif seulement quand ALM tourne avec subscription.

Chaque service purge de façon idempotente : un message rejoué ne trouve plus rien. La page Purge interroge ces trois services pour vérifier qu'ils ne gardent plus rien d'un tenant supprimé, et permet d'effacer les données de tenants que subscription ne connaît pas.

Les messages UPDATE_PLAN

Chaque changement de statut ou de validité, comme chaque changement d'accès, envoie à irrc un message `UPDATE_PLAN` sur `subscription_irrc_topic`, avec le `tenantId` comme clé. irrc vide alors son cache des accès de ce tenant et les relit chez subscription.

Le changement est écrit, puis le message est envoyé dans la même transaction et son accusé de réception attendu : un envoi qui échoue annule le changement et fait échouer la requête. Une révocation n'est donc jamais validée sans qu'irrc en soit prévenu. Une validité inchangée n'envoie rien.

Les e-mails aux membres

Les membres d'un tenant (un rôle ou un plan, et une adresse e-mail) reçoivent un e-mail quand son statut change :

Événement	E-mail
Suspension	L'accès s'arrête.
Suppression programmée	Plus d'accès ; données purgées à la date de purge sauf restauration.
Réactivation, ou restauration vers `ACTIVE`	L'accès revient. Un membre dont l'accès personnel est suspendu n'en reçoit pas.
Restauration vers `SUSPENDED`	La suppression est annulée ; l'accès reste suspendu.

Une modification de validité et la purge n'écrivent à personne. Chaque e-mail est dans la langue de l'utilisateur, ou en français puis en anglais s'il n'en a pas choisi. Les e-mails partent après la validation du changement, par le service de notification (`SEND_MAIL` sur `notification_topic`) : un changement annulé n'écrit à personne, et un envoi qui échoue est journalisé sans annuler le changement.

Les relances de fin de contrat

`ContractReminderJob` tourne chaque jour à 08:00, heure de Paris (`application.contract-reminders.cron`). Il ne regarde que les tenants `ACTIVE` qui ont un `validUntil` : un tenant sans fin n'est jamais relancé.

Destinataire	Seuils par défaut	Réglage
MLA (`support.reglab@mladvisory.com` par défaut), un seul e-mail à toutes ses adresses, dans sa langue (`fr` par défaut)	60, 30 et 7 jours avant le dernier jour	`mla-recipients`, `mla-language`, `mla-thresholds-days`
Les administrateurs du tenant : rôle administrateur, accès non suspendu, compte non désactivé, une adresse e-mail. Chacun dans sa langue, ou en français puis en anglais.	30 et 7 jours	`admin-thresholds-days`
Les deux	Fin du contrat (« échu »), jusqu'à 7 jours après	`ended-lookback`

Un seuil de N jours est dû à partir du jour où il reste N jours avant le dernier jour d'accès. Parmi les seuils dus et pas encore traités, seul le plus urgent est envoyé ; les autres sont enregistrés comme sautés. Un passage en retard, ou une validité fixée près de sa fin, n'envoie donc qu'un e-mail. Un contrat échu depuis plus de 7 jours n'est plus relancé, pour que la première exécution n'écrive pas à propos de contrats échus depuis longtemps.

Chaque relance envoyée ou sautée est enregistrée (`contract_reminder` : tenant, `validUntil`, seuil, date, nombre de destinataires) et n'est jamais renvoyée pour le même `validUntil`. Prolonger un tenant change son `validUntil` : les relances repartent pour la nouvelle date, et les anciennes restent visibles sous **Dates de fin précédentes**. Chaque tenant est traité dans sa propre transaction, sous verrou `FOR UPDATE SKIP LOCKED` ; un e-mail qui échoue annule les enregistrements du tenant, retentés le lendemain.

Le panneau **Contrats bientôt échus** lit `GET /api/v1/tenants/contract-endings?days=60` (les tenants `ACTIVE` dont la fin tombe dans les 60 jours ou est passée depuis moins de 30) ; la carte **Contrat** d'un tenant lit `GET /api/v1/tenants/{id}/contract-reminders`.

Les tenants de test

Les tests de bout en bout créent et suppriment leurs propres tenants par l'API. Un tenant dont le nom contient `playwright` (`application.tenant-lifecycle.test-tenant-marker`, sans tenir compte de la casse) est un tenant de test, et peut recevoir :

- à la création, une durée de vie, `expiresAfter` (durée ISO-8601, par exemple `PT1H`) : la purge le purge à son terme, même sans demande de suppression. Un tel tenant est dit temporaire ;
- à la suppression, un préavis plus court que le délai d'attente, `noticePeriod`.

Ces durées vont de `minimum-notice-period` (1 heure) au délai d'attente ; tout autre tenant qui en demande une est refusé (400).

Un tenant temporaire ne laisse pas de trace : ses changements d'accès ne sont ni enregistrés dans l'historique ni envoyés par e-mail. Deux jours après sa purge (`temporary-tenant-retention`), la purge l'efface entièrement, ligne et identifiant de connexion compris. Un service qui reçoit son `TENANT_DELETED` plus tard ne le trouve plus (404) ; ses données éventuelles deviennent alors orphelines, effaçables depuis la page Purge.

Paramètres

Propriété	Défaut	Rôle
`application.tenant-lifecycle.deletion-grace-period`	60 jours	Délai d'attente avant la purge.
`application.tenant-lifecycle.purge-cron`	toutes les 15 minutes	Passage de la purge ; - la désactive.
`application.tenant-lifecycle.minimum-notice-period`	1 heure	Préavis minimum d'un tenant de test.
`application.tenant-lifecycle.test-tenant-marker`	`playwright`	Marque d'un tenant de test dans son nom.
`application.tenant-lifecycle.temporary-tenant-retention`	2 jours	Délai avant l'effacement d'un tenant temporaire purgé.
`application.contract-reminders.*`	voir ci-dessus	Relances de fin de contrat.
`application.kafka.send-timeout`	10 secondes	Attente maximale de l'accusé de réception d'un envoi Kafka.

L’accès du support

Cette page décrit de bout en bout l’accès du support aux comptes des clients : qui peut consulter qui, comment le client autorise, comment la consultation est ouverte, limitée et terminée, et ce qui est enregistré. Elle s’adresse aux lecteurs techniques. L’usage de la console est décrit dans Console support ; le point de vue de l’utilisateur ciblé, dans le manuel de CIRCÉ, page « Accès du support ».

Vue d’ensemble

Une consultation support ouvre CIRCÉ ou HERMES ALM avec l’identité applicative d’un membre choisi, dans un seul tenant, en lecture seule. Le support voit les données et peut télécharger les exports accessibles à ce membre ; il ne voit pas son navigateur en direct.

Cinq acteurs y prennent part :

Acteur	Rôle
subscription	tient les politiques, les demandes et les autorisations ; décide si une consultation est permise ; décide qui peut en terminer une ; tient le journal
la passerelle (gateway)	ouvre la session de consultation, impose le tenant et la lecture seule, revérifie l’autorisation pendant la session, signale ouvertures, fins et refus à subscription
Keycloak	les identités ; la passerelle y obtient la session du membre ciblé ; subscription y relit les rôles de l’intervenant et de la cible
CIRCÉ et ALM	la page **Accès support** et le bandeau où l’utilisateur répond aux demandes et autorise
la console support	la page de subscription-front d’où le support et les administrateurs agissent

Le parcours type, en politique **Sur demande** :

1. Le support demande l’accès depuis la console ; l’utilisateur reçoit un courriel et un bandeau.
2. L’utilisateur autorise depuis sa propre session, pour une durée qu’il choisit.
3. Le support clique sur **Consulter en tant que** ; la passerelle demande à subscription si c’est permis, puis ouvre un onglet connecté avec l’identité de l’utilisateur.
4. La passerelle revérifie l’autorisation pendant la session et la termine au premier motif de fin.
5. Chaque étape est enregistrée dans le journal support.

> ⚠ Avertissement
>
> **Cet accès contourne l’authentification auprès de l’IdP d’entreprise de l’utilisateur cible, son MFA et ses règles d’accès conditionnel.** Le support s’authentifie avec son propre compte ; la passerelle obtient ensuite une session de la cible auprès de Keycloak.

Qui peut consulter qui

L’intervenant est l’une de ces deux personnes :

- un membre du support MLA : un compte du realm `mla`, actif dans Keycloak, qui porte le rôle de realm `SUPPORT` ou `ROLE_ADMIN` (rôles effectifs, composites et groupes compris). Il peut intervenir dans tous les tenants ;
- un administrateur du tenant : un membre actif de ce tenant, de rôle `ADMIN`. Il ne peut intervenir que dans ce tenant.

subscription lit l'intervenant dans Keycloak à chaque décision, jamais dans les rôles que lui transmet la passerelle : un compte désactivé ou privé de son rôle perd l'accès à la vérification suivante.

La cible doit réunir toutes ces conditions :

- membre actif du tenant : compte non désactivé, tenant au statut actif et dans sa période de validité, un rôle dans le tenant, accès non suspendu ;
- pas un collaborateur MLA : un utilisateur sans client, ou dont le client a un sous-domaine listé dans `application.support-access.staff-clients` (`mla` par défaut), n'est jamais consulté, quel que soit son realm ;
- dans Keycloak : un compte trouvé une seule fois, actif, qui n'est pas un compte de service (`service-account-…`) et ne porte aucun rôle privilégié. La liste vient de security-library (`ImpersonationTargets`), partagée avec la passerelle : les rôles de realm `ROLE_ADMIN`, `SUPPORT` et `admin`, et tout rôle du client `realm-management`. subscription lit les rôles effectifs ; la passerelle revérifie sur les jetons émis pour la consultation ;
- pas l'intervenant lui-même, et pas depuis une consultation déjà ouverte (pas de chaînage).

La politique du tenant

Chaque tenant a une politique d'accès du support :

Politique	Valeur	Effet
Sur demande	`ON_REQUEST`	Valeur initiale de tout tenant. Chaque utilisateur autorise lui-même le support, pour une durée limitée.
Toujours	`ALWAYS`	Aucune autorisation individuelle n'est requise. Les autres conditions s'appliquent toujours.
Jamais	`NEVER`	Toute consultation est refusée, même si une autorisation individuelle existe.

La politique est changée par un administrateur plateforme ou un administrateur actif du tenant (`PUT /api/v1/support-access/tenants/{tenantId}/policy`). Chaque changement écrit un événement `SUPPORT_POLICY_CHANGED` avec la politique avant et après.

Les demandes d'accès

Une demande (`POST /api/v1/support-access/requests`) nomme le tenant, la cible, une durée suggérée, un motif facultatif (2 000 caractères au plus) et le produit à ouvrir. subscription l'accepte seulement si la décision d'accès pour cet intervenant et cette cible est `CONSENT_REQUIRED` ou `CONSENT_FOR_ANOTHER_REQUESTER` : une autorisation manque. Dans tous les autres cas, elle répond 403 avec la raison (ou « Access already authorized »).

Règles :

- la durée suggérée va de 15 minutes à `application.support-access.max-grant-duration` ;
- le produit doit avoir une page dans `application.support-access.front-urls` ; `{subdomain}` y est remplacé par le sous-domaine du client du tenant ;
- une demande vit sept jours. Une demande du même intervenant à la même cible encore en attente est renvoyée telle quelle ; une demande expirée est close et remplacée ;
- l'adresse électronique et la langue de l'intervenant sont lues dans son jeton, pour lui écrire ensuite.

Après l'enregistrement, subscription écrit `SUPPORT_ACCESS_REQUESTED` au journal et, une fois la transaction validée, envoie à la cible le courriel « Demande d'accès support », avec un lien vers la page **Accès support** du produit.

Un utilisateur ne voit que les demandes qui lui sont adressées, et un intervenant que les siennes (`GET /api/v1/support-access/requests?tenantId=`).

Refus. La cible peut refuser une demande (`POST /api/v1/support-access/requests/{requestId}/decline`, réponse 204). Seule la cible, membre actif du tenant, le peut (403 sinon) ; une demande inconnue, déjà traitée ou expirée répond 404. La demande est close et marquée refusée, le journal reçoit `SUPPORT_ACCESS_DECLINED`, et l'intervenant reçoit le courriel « Demande d'accès support refusée ». Il peut redemander ensuite.

Les autorisations

Une autorisation est donnée par l'utilisateur lui-même, depuis sa propre session authentifiée (`POST /api/v1/support-access/tenants/{tenantId}/grant`). Une demande reçue, ou son courriel, ne donne aucun accès.

- **Conditions.** L'utilisateur est membre actif du tenant, et la politique est **Sur demande** (403 sinon).
- **Durée.** La fin est fixée par l'utilisateur, entre 15 minutes et `application.support-access.max-grant-duration` (30 jours par défaut) à partir de maintenant.
- **Portée.** Une autorisation donnée en réponse à une demande ne laisse entrer que l'intervenant qui l'a faite (son identifiant et son realm). Une autorisation donnée sans demande laisse entrer tout le support.
- **Coexistence.** Un utilisateur peut avoir plusieurs autorisations en cours : une par intervenant, plus une pour tout le support. Une nouvelle autorisation ne remplace que celle de même portée ; la remplacée est révoquée (`SUPPORT_ACCESS_REVOKED`).
- **Demandes résolues.** Une autorisation donnée sur demande clôt cette demande seulement ; une autorisation pour tout le support clôt toutes les demandes en attente adressées à l'utilisateur. Chaque intervenant dont la demande est close reçoit le courriel « Accès support autorisé », avec un lien vers la console (`application.support-access.console-url`, sinon la page du produit).
- **Révocation. Révoquer maintenant** (`DELETE /api/v1/support-access/tenants/{tenantId}/grant`) retire toutes les autorisations de l'utilisateur dans ce tenant, avec un `SUPPORT_ACCESS_REVOKED` par autorisation.

Chaque autorisation écrit `SUPPORT_ACCESS_GRANTED`, avec sa fin et l'intervenant auquel elle est liée.

La décision d'accès

La passerelle demande à subscription si un intervenant peut consulter une cible (`POST /api/v1/support-access/check`) à l'ouverture et pendant la session. La console et la création d'une demande utilisent la même décision. subscription vérifie dans cet ordre et répond par la première raison qui s'applique :

Raison	Cause
`CHAINING`	L'appel vient déjà d'une consultation.
`INVALID_IDENTITY`	La cible ou l'intervenant n'est pas nommé.
`SELF`	L'intervenant est la cible.
`TENANT_NOT_FOUND`	Tenant inconnu.
`TARGET_NOT_ACTIVE_MEMBER`	La cible n'est pas membre actif du tenant.
`TARGET_IS_MLA_STAFF`	La cible est un collaborateur MLA.
`IDENTITY_PROVIDER_UNAVAILABLE`	Keycloak n'a pas répondu, à la lecture de l'intervenant ou de la cible.
`ACTOR_NOT_AUTHORIZED`	L'intervenant est désactivé dans Keycloak, ou n'est ni membre du support MLA ni administrateur actif du tenant.

Raison	Cause
`TARGET_PRIVILEGED_OR_DISABLED`	La cible est désactivée, introuvable, compte de service ou porte un rôle privilégié dans Keycloak.
`POLICY_NEVER`	La politique du tenant est **Jamais**.
`ALLOWED`	Politique **Toujours** : permis, sans date de fin.
`CONSENT_REQUIRED`	Politique **Sur demande** et aucune autorisation en cours.
`CONSENT_FOR_ANOTHER_REQUESTER`	Des autorisations existent, mais aucune pour cet intervenant ni pour tout le support.
`ALLOWED`	Une autorisation couvre l'intervenant : permis jusqu'à sa fin.

La réponse porte `allowed`, `until` (la fin de l'autorisation retenue, vide en politique **Toujours**), `reason` et le nom du tenant.

La session de consultation

Ouverture. Consulter en tant que envoie à la passerelle (`POST /api/impersonation`) le tenant, la cible et l'adresse du produit. La passerelle demande la décision à subscription, puis renvoie une adresse d'entrée à usage unique (`…/impersonation/enter?ticket=…`) ; la console n'ouvre son onglet que sur cette adresse de la passerelle. La passerelle ouvre alors la session de la cible auprès de Keycloak, avec le client `gateway-impersonation` du realm de la cible, et signale l'ouverture à subscription (`IMPERSONATION_STARTED`, avec une référence de session).

Les jetons sont rangés par session web de la passerelle : une consultation ne remplace ni ne ferme la session que l'utilisateur ciblé a lui-même ouverte.

Limites pendant la consultation (passerelle) :

- le tenant est imposé : une requête vers un autre tenant est refusée ;
- les méthodes d'écriture sont refusées, sauf une liste explicite de requêtes POST de consultation ou d'export : celles de `application.impersonation.read-only-requests` et celles que les services publient eux-mêmes dans Consul, sous `support/read-only-requests/<service>`. `application.impersonation.denied-requests` l'emporte toujours ;
- l'autorisation est revérifiée auprès de subscription au plus une fois toutes les 60 secondes, lors d'une requête de la session. Ce délai n'est pas configurable : une révocation peut donc prendre jusqu'à 60 secondes pour couper une session ouverte ;
- la fin de l'autorisation, connue à chaque vérification, est comparée à chaque requête ; une autorisation plus longue donnée ensuite ne prolonge pas la session ;
- si subscription ou Keycloak ne répond pas, chaque requête reçoit 503 pendant le délai de grâce ; au-delà, la session se termine ;
- après 5 lancements refusés en 10 minutes depuis la même session de console, la passerelle répond 429 sans interroger subscription.

Les requêtes refusées pendant la session sont comptées par type (lecture seule, autre tenant, nouvelle consultation) et transmises avec la fin.

Réglages de la passerelle (Consul) :

Propriété	Défaut	Rôle
`application.impersonation.max-session-duration`	`PT4H`	durée maximale d'une consultation
`application.impersonation.unavailable-grace`	`PT5M`	délai de grâce quand subscription ou Keycloak ne répond pas
`application.impersonation.sweep-interval`	`PT1M`	intervalle du balayage des sessions inactives ou échues

Propriété	Défaut	Rôle
`application.impersonation.max-refusals`	`5`	lancements refusés avant 429
`application.impersonation.refusal-window`	`PT10M`	fenêtre de comptage des refus
`application.impersonation.published-requests-refresh`	`PT30S`	relecture des requêtes publiées dans Consul
`application.session.valkey.enabled`	`false`	sessions web et état des consultations dans Valkey, partagés entre instances

Comment les services reconnaissent une consultation

Les services reçoivent le jeton de la cible, obtenu par la passerelle auprès de Keycloak : ils appliquent les permissions et l'isolation de la cible comme pour elle-même. Rien dans ce jeton ne donne de droit supplémentaire.

La passerelle ajoute à chaque requête de la consultation un marqueur de lecture seule, l'en-tête `X-SUPPORT-READ-ONLY: true`. Ce marqueur ne sert qu'à supprimer des écritures, il n'accorde aucun accès ni rôle :

- CIRCÉ (irrc) ne crée pas de profil et ne synchronise pas les informations du jeton lors d'une lecture ;
- la bibliothèque neutralise de même la synchronisation de profil ;
- ALM ne crée ni ne met à jour la fiche d'activité de la cible.

Les journaux techniques, les événements d'audit et les caches de consultation peuvent toujours être alimentés.

subscription, de son côté, refuse toute nouvelle consultation demandée depuis une consultation : la passerelle le lui signale dans la vérification (`impersonated`), ce qui donne la raison `CHAINING`.

Les motifs de fin

Chaque fin de session est un événement `IMPERSONATION_ENDED` avec son motif :

Motif	Libellé de la console	Signification
`EXIT`	quittée par l'intervenant	Le support a cliqué **Quitter** dans le bandeau.
`EXPIRED`	autorisation expirée	La fin de l'autorisation est passée.
`REVOKED`	autorisation révoquée	subscription a refusé lors d'une revérification : révocation, changement de politique, cible ou intervenant plus autorisé.
`MAX_DURATION`	durée maximale atteinte	La session a atteint `max-session-duration`.
`LOGOUT`	déconnexion	Le support s'est déconnecté, ou sa propre session a disparu.
`IDLE`	session inactive	La session est restée sans requête jusqu'à son expiration dans le magasin de sessions de la passerelle.

Motif	Libellé de la console	Signification
`UNAVAILABLE`	service indisponible trop longtemps	subscription ou Keycloak est resté injoignable au-delà du délai de grâce.
`UNKNOWN`	fin non signalée	La passerelle n'a jamais signalé la fin ; subscription l'écrit elle-même (voir plus bas).
`TERMINATED`	terminée par …	Un administrateur plateforme ou un administrateur du tenant l'a terminée depuis la console.

Sessions orphelines. Avec le magasin en mémoire, un redémarrage de la passerelle perd ses sessions, et leur fin n'est jamais signalée. Toutes les cinq minutes (`application.support-access.orphan-session-cron`), subscription écrit une fin de motif `UNKNOWN`, au nom de `system`, pour chaque ouverture plus ancienne que `application.support-access.orphan-session-after` (5 heures par défaut) restée sans fin. Cette fin est datée de l'ouverture plus ce délai : une borne haute, pas l'heure réelle. Une ouverture qui porte une référence de session n'est close que par une fin de même référence. Le délai doit rester supérieur à la durée maximale de session de la passerelle.

Terminer une session

Le bouton **Terminer** de la console appelle la passerelle (`POST /api/impersonation/sessions/{sessionRef}/terminate`). Avant de fermer quoi que ce soit, la passerelle demande l'autorisation à subscription (`POST /api/v1/support-access/sessions/{sessionRef}/termination`, avec l'identifiant et le realm de l'appelant).

subscription accorde (`ALLOWED`) si l'appelant est administrateur actif du tenant de la session, ou s'il est un compte actif du realm `mla` portant `ROLE_ADMIN`, lu dans Keycloak. Elle refuse sinon :

Raison	Cause
`INVALID_IDENTITY`	Appelant non nommé.
`NOT_FOUND`	Aucune ouverture ne porte cette référence.
`ALREADY_ENDED`	La session a déjà une fin enregistrée.
`NOT_AUTHORIZED`	L'appelant n'a pas ce droit. Un membre du support ne peut pas terminer la session d'un autre.
`IDENTITY_PROVIDER_UNAVAILABLE`	Keycloak n'a pas pu dire si l'appelant est administrateur plateforme.

Si c'est accordé, la passerelle ferme la session Keycloak de la cible, invalide la session web de la consultation et écrit la fin (`TERMINATED`, avec `terminatedBy` et `terminatedByRealm`). La requête suivante de l'intervenant reçoit 401 et son onglet affiche **Consultation terminée**.

> **Note**
>
> **Plusieurs instances de passerelle.** Avec le magasin de sessions en mémoire (défaut), une seule instance tient une session : une terminaison reçue par une autre instance répond 404. Avec Valkey (`application.session.valkey.enabled`), toutes les instances voient les mêmes sessions et une seule ferme chaque session.

Événements, journal et audit

Les événements de l'accès du support sont des lignes de la table `access_event`, la même que l'historique des accès :

Événement	Écrit par	Quand
`SUPPORT_POLICY_CHANGED`	subscription	changement de politique (avant, après)
`SUPPORT_ACCESS_REQUESTED`	subscription	demande créée
`SUPPORT_ACCESS_GRANTED`	subscription	autorisation donnée
`SUPPORT_ACCESS_REVOKED`	subscription	autorisation révoquée ou remplacée
`SUPPORT_ACCESS_DECLINED`	subscription	demande refusée par la cible
`IMPERSONATION_STARTED`	la passerelle	ouverture d'une consultation
`IMPERSONATION_ENDED`	la passerelle ; subscription pour `UNKNOWN`	fin d'une consultation
`IMPERSONATION_REFUSED`	la passerelle	ouverture refusée, avec la raison de subscription

La passerelle envoie ses événements à `POST /api/v1/support-access/sessions/events` (`SupportSessionEvent`). subscription les contrôle avant de les écrire : type parmi les trois `IMPERSONATION_*`, motif de fin connu, compteurs de requêtes refusées sur une fin seulement, référence de session faite de lettres, chiffres, _ et - (64 caractères au plus), et `terminatedBy` présent si et seulement si le motif est `TERMINATED`. Un tenant ou une cible inconnue répond 404.

Chaque ligne garde, selon l'événement : la date, l'acteur réel et son realm, la cible et son realm, le tenant et son client, l'intervenant (sur les événements de session seulement), la fin de validité, l'application ouverte, le motif de fin ou de refus, les requêtes refusées par type, la référence de session, l'auteur d'une terminaison, les politiques avant et après, et l'intervenant auquel une autorisation est liée. Les noms sont copiés dans la ligne : un changement ou une suppression de compte ne les efface pas.

- **Refus dédoublonnés.** Un `IMPERSONATION_REFUSED` identique (même tenant, cible, intervenant et raison) n'est enregistré qu'une fois par `application.support-access.refusal-deduplication-window` (5 minutes par défaut).
- **Append-only.** Un déclencheur de la base refuse toute modification et toute suppression de `access_event`, sauf la purge de rétention.
- **Rétention.** Chaque nuit (03:40, heure de Paris, `application.access-event.purge-cron`), les événements plus anciens que `application.access-event.retention` (90 jours par défaut) sont supprimés. Voir aussi Purge.

Le journal support (`GET /api/v1/support-access/journal`) ne lit que les événements `SUPPORT_*` et `IMPERSONATION_*`. Pour un tenant donné, il est ouvert au support MLA et aux administrateurs de ce tenant ; sans tenant, au support MLA seul. Il marque comme en cours (`open`) chaque ouverture dont la référence de session n'a pas de fin. Les pages font 500 lignes au plus.

L'export (`GET /api/v1/support-access/journal/export`) produit `support-access.csv`, en UTF-8 avec BOM, avec les colonnes `date`, `event`, `actor`, `actorRealm`, `user`, `userRealm`, `tenant`, `front`, `validUntil`, `endReason`, `refusalReason`, `refusedReadOnly`, `refusedTenant`, `refusedChaining`, `policyBefore`, `policyAfter`, `sessionRef`, `terminatedBy`, `terminatedByRealm`. Une cellule qui commence par `=`, `+`, `-`, `@`, une tabulation ou un retour chariot est précédée d'une apostrophe, pour qu'un tableur ne l'exécute pas comme une formule.

Mises à jour en direct et courriels

Après chaque demande créée, autorisation donnée, révocation ou refus, et une fois la transaction validée, subscription publie `SUPPORT_REQUEST_CHANGED` sur le topic Kafka `ui_events_topic`, avec l'identifiant public du tenant, le realm et l'identifiant de l'utilisateur ciblé. Un envoi raté est journalisé et abandonné : les fronts relisent l'état à intervalles réguliers.

- CIRCÉ (irrc) et ALM (stress) relaient le message à leurs navigateurs : le bandeau de demande et la pastille de l'utilisateur se rechargent aussitôt.
- subscription consomme aussi ce topic, chaque instance dans un groupe à elle, et le relaie à la console sur son flux `GET /api/v1/events`, sous le nom `SupportAccessChanged`. Les administrateurs plateforme

reçoivent l'identifiant du tenant ; les autres utilisateurs de la console (support, administrateurs de tenant) reçoivent l'événement sans aucun champ, et leur console relit le tenant affiché.

Les courriels (« Demande d'accès support », « Accès support autorisé », « Demande d'accès support refusée ») sont confiés au service de notification par un message `SEND_MAIL` sur `notification_topic`, après validation de la transaction. Un courriel qui ne peut pas être construit (page du produit ou sous-domaine du client non configurés) est abandonné et journalisé, sans annuler l'autorisation ni le refus.

Points d'accès de l'API

Tous sous `/api/v1/support-access` :

Point d'accès	Appelant
`GET /tenants`	utilisateur de la console : le support voit tous les tenants, un administrateur de tenant les siens
`GET /tenants/{tenantId}/members`	support MLA, ou administrateur du tenant
`GET /tenants/{tenantId}`	état du tenant : support MLA, ou membre actif
`PUT /tenants/{tenantId}/policy`	administrateur plateforme, ou administrateur du tenant
`GET /tenants/{tenantId}/grants`	support MLA, ou administrateur du tenant
`POST /tenants/{tenantId}/grant`, `DELETE /tenants/{tenantId}/grant`	l'utilisateur lui-même, membre actif
`POST /requests`, `GET /requests?tenantId=`	intervenant ; cible pour la lecture
`POST /requests/{requestId}/decline`	la cible de la demande
`GET /journal`, `GET /journal/export`	support MLA ; administrateur du tenant pour son tenant
`POST /check`	la passerelle seule
`POST /sessions/events`	la passerelle seule
`POST /sessions/{sessionRef}/termination`	la passerelle seule

« La passerelle seule » : un jeton du realm `shared`, émis au client `svc-gateway`, membre du groupe d'appelants `support-access` (voir La sécurité).

Réglages de subscription

Propriété	Défaut	Rôle
`application.support-access.max-grant-duration`	`P30D`	durée maximale d'une autorisation et d'une durée suggérée
`application.support-access.console-url`	aucun	adresse de la console dans les courriels envoyés à l'intervenant ; à défaut, la page du produit
`application.support-access.front-urls.<produit>`	aucun	page **Accès support** de chaque produit, `{subdomain}` remplacé par le sous-domaine du client
`application.support-access.staff-clients`	`[mla]`	sous-domaines des clients dont les utilisateurs sont des collaborateurs MLA

Propriété	Défaut	Rôle
`application.support-access.orphan-session-after`	`PT5H`	âge d'une ouverture sans fin avant la fin `UNKNOWN` ; supérieur à la durée maximale de session
`application.support-access.orphan-session-cron`	`0 */5 * * * *`	planification des sessions orphelines (`-` la désactive)
`application.support-access.refusal-deduplication-window`	`PT5M`	fenêtre de dédoublonnage des refus
`application.access-event.retention`	`P90D`	conservation des événements d'accès

La sécurité

Cette page décrit comment subscription protège ce qu'elle détient et qui peut l'appeler : les jetons acceptés, ce que chaque point d'accès exige, les appels des autres services, les credentials des tenants et leurs clés maîtres, les points d'accès publics, et où subscription lit ses secrets. Elle s'adresse aux lecteurs techniques (développement, exploitation). Aucune valeur secrète n'y figure, seulement des noms de propriétés et de chemins.

Les jetons acceptés

subscription est un serveur de ressources OAuth2 : chaque appel porte un jeton d'accès Keycloak (JWT), vérifié par le décodeur de security-library (`CustomNimbusJwtDecoder`).

- **Realms acceptés.** `security.jwt.allowed-realms` liste les realms dont les jetons sont acceptés. L'émetteur (`iss`) du jeton doit être exactement `<url Keycloak>/realms/<realm>` pour l'un d'eux ; tout autre jeton est refusé avant de contacter Keycloak. Sans cette propriété, le service ne démarre pas.
- **Signature.** Seuls les jetons signés RS256 sont lus ; un jeton chiffré ou non signé reçoit 401. Les clés publiques viennent du point JWKS du realm.
- **Realm de l'appelant.** subscription lit le realm de l'utilisateur dans l'émetteur du jeton, et son identifiant dans la claim `preferred_username`. Un utilisateur est le couple (realm, identifiant) : un identifiant n'est unique qu'à l'intérieur de son realm.

Le navigateur ne détient pas de jeton : il a une session auprès de la passerelle, qui joint le jeton de l'utilisateur à chaque requête vers subscription.

Qui est qui

subscription distingue ces appelants, toujours d'après le jeton :

Appelant	Reconnu à
Administrateur plateforme	jeton d'un realm plateforme (`security.jwt.platform-realms`, `[mla, shared]` par défaut) portant exactement le rôle `ROLE_ADMIN` dans `access_roles` ou `realm_access.roles`
Membre du support MLA	administrateur plateforme, ou jeton du realm `mla` portant le rôle `SUPPORT` dans `access_roles` ou `realm_access.roles`
Utilisateur	son realm et son `preferred_username` ; ses droits dans un tenant (membre actif, administrateur) sont lus dans la base de subscription
Service de la plateforme	jeton d'un realm plateforme émis à un client Keycloak listé dans un groupe d'appelants (`security.service-callers.groups.<groupe>`)
Compte de service partagé	jeton d'un realm plateforme émis au client `internal_service_client`
La passerelle, pour l'accès du support	jeton du realm `shared` émis au client `svc-gateway`, membre du groupe `support-access`

Un rôle `ROLE_ADMIN` porté dans le realm d'un client ne fait pas un administrateur plateforme.

Ce que l'API exige

La console d'administration est réservée aux administrateurs plateforme. subscription applique la première règle qui correspond à la requête :

Requête	Qui peut l'appeler
`OPTIONS`, toute adresse	tout le monde
`GET /actuator`, `/actuator/health/**`, `/actuator/info`	tout le monde (sondes Kubernetes et Consul)
le reste de `/actuator/**`	les services du groupe `actuator` seulement (Spring Boot Admin), jamais un jeton d'utilisateur, administrateurs plateforme compris
`GET /v3/api-docs/**`, `/swagger-ui/**`	tout le monde ; servis seulement là où Consul les active (en local et en intégration)
`GET /api/v1/user-tenants/{service}/{utilisateur}`	administrateur plateforme, services du groupe `user-tenants`, ou l'utilisateur lui-même pour son propre realm
`GET /api/v1/tenant-credentials`	services du groupe `tenant-credentials` seulement, jamais un jeton d'utilisateur, administrateurs plateforme compris
`GET /api/v1/user-tenants/{tenant}/{service}/{utilisateur}`	administrateur plateforme, ou services du groupe `user-tenants`
`GET /api/v1/tenant-status`	administrateur plateforme, ou services du groupe `tenant-status`
`GET /api/v1/public/clients/by-subdomain/*` et `…/logo`	tout le monde, sans jeton
`GET /api/v1/tenants/{tenantId}/logo`	tout appelant authentifié, puis : compte de service, administrateur plateforme, ou membre actif du tenant
`/api/v1/reglab-users/me/language`	tout jeton qui nomme un utilisateur ; chacun lit et règle sa propre langue
`GET /api/v1/events` (flux des mises à jour en direct)	administrateur plateforme, ou utilisateur de la console support
`/api/v1/support-access/check`, `/sessions/events`, `/sessions/*/termination`	la passerelle seule
`GET /api/v1/access-events`	membre du support MLA ; hors administrateur plateforme, limité aux événements de l'accès du support
`/api/v1/support-access/**` (le reste)	tout appelant authentifié ; chaque opération vérifie ensuite ses droits (voir L'accès du support)
tout le reste, dont `/api/v1/master-keys` et `/api/v1/security/password/generate`	administrateur plateforme

Une page d'erreur d'une requête déjà autorisée (un 404, par exemple) reste une erreur et ne devient pas un 403.

Appels des autres services

Les services de la plateforme appellent subscription avec leur propre jeton, obtenu auprès de Keycloak par le flux *client credentials*. security-library le fournit (`ServiceTokenClient`) à partir des

clés `application.s2s.*` du service : `application.s2s.client-id`, `application.s2s.client-secret`, `application.s2s.token-endpoint` et `application.s2s.grant-type`. Le jeton est renouvelé avant son expiration (`application.s2s.expiration-margin`, 60 secondes par défaut). Un service qui a son propre client Keycloak (`svc-irrc`, `svc-stress`…) range ses clés dans son chemin Vault, qui l'emporte sur celles du client partagé de `secret/application`.

subscription accorde chaque groupe de points d'accès aux clients listés dans sa configuration Consul, `security.service-callers.groups.<groupe>` :

Groupe	Points d'accès	Clients dans la pile locale
`tenant-credentials`	`GET /api/v1/tenant-credentials`	`svc-irrc`, `svc-stress`, `svc-risk-engine`
`user-tenants`	`GET /api/v1/user-tenants/{service}/{utilisateur}` et `/{tenant}/{service}/{utilisateur}`	`svc-irrc`, `svc-library`, `svc-stress`, `svc-risk-engine`, `svc-assistant`
`tenant-status`	`GET /api/v1/tenant-status`	`svc-stress`
`support-access`	les points d'accès de la passerelle pour l'accès du support	`svc-gateway`
`actuator`	`/actuator/**` au-delà de health et info	`svc-admin`

- Tant que `security.service-callers.legacy-shared-client` vaut `true` (défaut), le client partagé `internal_service_client` est admis dans chaque groupe. La première admission par groupe est journalisée, pour repérer les services qui l'utilisent encore. Les points d'accès de la passerelle pour l'accès du support exigent en plus le client `svc-gateway` du realm `shared` : le client partagé n'y entre pas.
- Un changement de ces groupes dans Consul est pris en compte sans redémarrage, au rafraîchissement suivant de la configuration (environ une minute). Une valeur invalide (groupe vide) est refusée et la valeur en cours est gardée ; un redémarrage échouerait en revanche sur elle.
- Les groupes réellement configurés dans les environnements déployés se lisent dans leur Consul, `config/subscription/data`.

Appels sortants de subscription. Le code de subscription n'appelle pas les autres services en HTTP. Il appelle :

- l'API d'administration de Keycloak, dans chaque realm, avec le client `subscription-provisioning` (`application.keycloak-accounts.client-id`) et le secret de ce realm (`application.keycloak-accounts.client-secrets.<realm>`). Il s'en sert pour désactiver ou réactiver des comptes, lire les connexions, et relire les rôles de l'intervenant et de la cible de l'accès du support ;
- Kafka, avec l'utilisateur de `spring.kafka.*` ; les courriels partent sur `notification_topic` avec un utilisateur dédié (`application.notification.kafka-username`, `application.notification.kafka-password`), seul autorisé à écrire ce topic. Sans ces deux propriétés, aucun courriel n'est envoyé.

Les credentials des tenants

Chaque tenant a un credential, choisi à sa création (**Identifiants de connexion** dans le formulaire du tenant) et que le formulaire ne modifie plus ensuite. CIRCÉ (irrc) s'en sert pour chiffrer les données du tenant.

- **Type.** Seul **Mot de passe** (`PASSWORD`) est accepté : un credential **Clé** (`KEY`) est refusé, aucun service ne chiffrant encore avec une paire de clés.
- **Mot de passe.** Il doit atteindre `application.credential.minimum-password-length` caractères. **Générer** en propose un (`GET /api/v1/security/password/generate`) ; il n'est affiché qu'à ce moment. subscription tire un sel aléatoire à la création.
- **Stockage.** Les colonnes `password`, `salt`, `private_key` et `public_key` de `tenant_credential` sont chiffrées avec la clé maître courante (voir plus bas).
- **Lecture.** `GET /api/v1/tenant-credentials?tenantId=` rend le credential en clair aux seuls services du groupe `tenant-credentials`. Aucun front ne le lit, et un jeton d'administrateur plateforme est refusé : une session d'administrateur volée ne donne pas les clés. Chaque lecture est journalisée avec le client

appelant, son realm, le tenant et le résultat, jamais le secret (`Tenant credential read: tenant=…, caller=…, realm=…, found=…`).

Les clés maîtres

subscription chiffre les credentials des tenants avec ses propres clés maîtres (`EncryptionService`) :

- chaque valeur est chiffrée en AES-GCM (format v2 de `VersionedAesGcmEngine`) ; elle commence par l'octet de version puis l'identifiant de la clé qui l'a chiffrée (0 à 255) ;
- chaque valeur est liée à la table `tenant_credential` : une valeur déplacée vers un autre usage du même secret est rejetée ;
- chaque clé est dérivée de sa phrase de passe et de son sel (PBKDF2, 600 000 itérations), une fois, au premier usage.

Les clés sont des propriétés :

Propriété	Rôle
`application.credential.encrypt-password`, `application.credential.encrypt-salt`	clé 0 ; elle lit aussi les valeurs écrites avant le format v2
`application.credential.keys.<id>.password`, `application.credential.keys.<id>.salt`	les autres clés, `<id>` de 1 à 255 ; une phrase de passe trop faible empêche le démarrage
`application.credential.current-key-id`	la clé de toutes les nouvelles écritures (0 par défaut) ; si elle n'est pas configurée, subscription ne démarre pas

Rechiffrement au démarrage. À chaque démarrage, subscription rechiffre avec la clé courante toutes les valeurs qui ne sont pas sur elle (`TenantCredentialReencryption`). Un verrou de la base limite ce travail à une seule instance. Une valeur qui échoue est journalisée et laissée telle quelle, pour le démarrage suivant. Le journal indique ensuite si toutes les valeurs sont sur la clé courante, ou lesquelles restent sur une autre clé.

L'onglet Clé maître. La page **Paramètres**, onglet **Clé maître**, montre le même état sans lire les journaux (`GET /api/v1/master-keys`, administrateurs plateforme). Les clés elles-mêmes ne sont jamais affichées, et l'écran ne modifie rien : chaque action se fait dans Vault. **Actualiser** relit l'état.

Le bandeau donne le verdict, du plus grave au moins grave :

Bandeau	Signification	Que faire
« … valeur(s) sur la clé …, absente de Vault »	des valeurs nomment une clé qui n'est plus configurée : elles sont illisibles	remettre la clé dans Vault sous le même identifiant, puis redémarrer subscription
« La clé … chiffre encore … valeur(s) »	une ancienne clé sert encore	la garder dans Vault ; redémarrer subscription pour rechiffrer
« La clé … ne chiffre plus rien »	l'ancienne clé ne sert plus	la retirer de Vault ; en garder une copie tant qu'une sauvegarde antérieure à la rotation existe
« Tous les credentials de tenant sont sur la clé … »	une seule clé configurée	rien

Le tableau donne, pour chaque clé, son **État** (**Courante**, **Encore utilisée**, **Peut être retirée**, **Absente de Vault**) et son nombre de **Valeurs**, plus une ligne **Format antérieur à v2 (clé 0)** s'il en reste.

Faire tourner la clé maître

La rotation est manuelle, dans Vault ; subscription ne lit ses clés qu'au démarrage, donc chaque étape se termine par un redémarrage progressif de subscription :

1. Ajouter la nouvelle clé (`application.credential.keys.<id>.password` et `.salt`) sans changer `current-key-id`, puis redémarrer. Tous les pods connaissent alors la clé avant qu'elle serve.

2. Rendre la nouvelle clé courante (`application.credential.current-key-id`), puis redémarrer : les valeurs sont rechiffrées au démarrage. L'onglet **Clé maître** doit dire que l'ancienne clé ne chiffre plus rien.
3. Retirer l'ancienne clé de Vault, puis redémarrer.
4. Garder l'ancienne clé dans le coffre de l'équipe tant qu'une sauvegarde de la base antérieure à la rotation existe.

La procédure détaillée, vérifications et retour arrière compris, est le runbook `rotation-cle-maitre-subscription` du dépôt de documentation. La rotation change la clé maître seulement : les credentials des tenants ne changent pas, et les données de CIRCÉ restent lisibles.

Points d'accès publics et logos

Avant toute connexion, un front lit le client que nomme son nom d'hôte :

- `GET /api/v1/public/clients/by-subdomain/{sous-domaine}` rend le realm où se connecter, le nom du client et s'il a un logo ;
- `GET /api/v1/public/clients/by-subdomain/{sous-domaine}/logo` rend son logo.

Ces deux points d'accès sont ouverts sans jeton. Ils ne répondent que pour un client qui a un sous-domaine et un realm accepté, et n'en disent pas plus ; sinon, 404. Leur réponse peut être gardée 5 minutes par les caches partagés.

Le logo du client d'un tenant (`GET /api/v1/tenants/{tenantId}/logo`) est réservé au compte de service, aux administrateurs plateforme et aux membres actifs du tenant ; un autre appelant est refusé avant de savoir si le tenant existe. Sa réponse est privée, gardée une heure, puis revalidée par ETag. Seuls les administrateurs plateforme déposent, lisent par client et retirent un logo (`/api/v1/clients/{id}/logo`).

Vault et Consul

subscription importe sa configuration de Consul, puis de Vault (`spring.config.import` : `consul:` puis `optional:vault://`). Une clé de Vault l'emporte sur une clé de même nom dans Consul. Consul est obligatoire : un agent injoignable arrête le démarrage. Vault est marqué `optional` pour le seul cas où il est désactivé ; un Vault injoignable reste soumis au réglage *fail-fast*.

Source	Chemin	Contenu
Consul	`config/application/data`, puis `config/subscription/data` (YAML)	la configuration : `security.jwt.*`, `security.service-callers.*`, `application.support-access.*`, Kafka, etc.
Vault, moteur KV `secret`	`secret/application`	les clés partagées par tous les services : utilisateur Kafka partagé, client de service partagé (`application.s2s.*`), `keycloak.server.url`
Vault	`secret/subscription`	les secrets de subscription : connexion à la base, utilisateur Kafka, clés maîtres (`application.credential.*`), secrets des clients `subscription-provisioning` (`application.keycloak-accounts.client-secrets.<realm>`)

La connexion se règle par variables d'environnement : `VAULT_URI`, `VAULT_TOKEN`, `VAULT_KV_BACKEND` (`secret` par défaut), `VAULT_AUTH_METHOD` (`token` par défaut ; AppRole et Kubernetes sont possibles avec `VAULT_APPROLE_ROLE_ID`, `VAULT_APPROLE_SECRET_ID`, `VAULT_K8S_ROLE`, `VAULT_K8S_AUTH_PATH`), et pour Consul `CONSUL_HOST`, `CONSUL_PORT`, `CONSUL_TOKEN`.

Événements et appels

Cette page, destinée aux lecteurs techniques, décrit comment subscription et sa console échangent avec les autres services de la plateforme :

- les messages Kafka que le service subscription écrit et lit ;
- les mises à jour en direct que subscription pousse à la console ;
- les appels HTTP synchrones, ceux de la console et ceux du service.

Elle décrit le code des branches `dev` de subscription et de subscription-front ; les consommateurs des autres services sont repris de la page d'architecture des messages Kafka de la documentation de la plateforme.

Vue d'ensemble

- subscription **écrit** sur cinq topics : `subscription_irrc_topic`, `tenant_lifecycle_topic`, `notification_topic`, `library_topic` et `ui_events_topic`. Il ne **lit** que `ui_events_topic`, pour relayer les changements à la console.
- La console (subscription-front) appelle **directement**, depuis le navigateur et par la passerelle, chaque service qu'elle administre : crawler, library, notification, referential, dpm. subscription n'est pas un intermédiaire, sauf pour les URL ajoutées à la bibliothèque commune.
- Le service subscription n'appelle aucun de ces services en HTTP : il ne joint library et notification que par Kafka. Son seul appel synchrone sortant est l'API d'administration de Keycloak.

Les topics Kafka

Topic	subscription	Types de message	Clé	Lu par
`subscription_irrc_topic`	écrit	`UPDATE_PLAN`	`tenantId`	irrc, assistant, risk-app
`tenant_lifecycle_topic`	écrit	`TENANT_DELETED`	`tenantId`	irrc, library, assistant, risk-app
`notification_topic`	écrit	`SEND_MAIL`	l'utilisateur, le tenant ou la demande d'accès support concernés	notification
`library_topic`	écrit	`NEW_PUBLICATION`, `NEW_COMMON_FILE`	`link` ou `fileName`	library
`ui_events_topic`	écrit et lit	écrit `SUPPORT_REQUEST_CHANGED` ; lit `CRAWLER_RUN_CHANGED`, `LIBRARY_DOCUMENT_CHANGED`, `LIBRARY_EMBEDDINGS_CHANGED`, `LIBRARY_INDEX_CHANGED`, `DOCLING_QUEUE_CHANGED`, `SUPPORT_REQUEST_CHANGED`	`tenantId`	subscription, irrc, risk-app

D'autres services écrivent aussi sur certains de ces topics : crawler sur `library_topic`, `notification_topic` et `ui_events_topic`, library sur `ui_events_topic`, irrc et risk-app sur `notification_topic`.

Chaque topic a son producteur dans le paquet `bus` : `Producer` (`subscription_irrc_topic`), `TenantLifecycleProducer`, `NotificationProducer`, `LibraryProducer` et `UiEventProducer` ; le lecteur est `UiEventsConsumer`.

Les messages

UPDATE_PLAN

Les appartenances d'un tenant ont changé : les plans ou le rôle d'un utilisateur dans ce tenant, un utilisateur retiré, suspendu ou rétabli, un utilisateur désactivé ou réactivé, ou le statut ou la validité du tenant. Le message nomme le tenant, jamais l'utilisateur ; il n'a pas de `content`.

irrc, assistant et risk-app oublient alors ce qu'ils gardent en cache pour ce tenant (tous les tenants quand le message n'en nomme pas un lisible) ; irrc relit ensuite les appartenances auprès de subscription. Voir Cycle de vie des tenants et Droits.

TENANT_DELETED

Le délai de grâce d'un tenant supprimé est écoulé et la purge l'a marqué supprimé : chaque service efface ce qu'il garde pour ce tenant. Le message est envoyé une fois, au moment où la purge est validée, et peut arriver plus d'une fois : chaque lecteur doit pouvoir le traiter à nouveau sans dommage. `tenantId` est l'identifiant public du tenant (celui que les services stockent), jamais la clé primaire de subscription. Voir Purge.

SEND_MAIL

Un mail prêt à partir : subscription produit lui-même le HTML, à partir des modèles `templates/mail/*.ftlh` et des textes `i18n/mail*.properties`. Le contenu porte `subject`, `htmlContent` et `recipients`, sans pièce jointe. Le service notification l'envoie, l'inscrit dans son historique et refuse le mail entier si un destinataire est hors des domaines autorisés. Les mails eux-mêmes sont décrits dans Notifications.

NEW_PUBLICATION et NEW_COMMON_FILE

Des documents pour la bibliothèque commune, envoyés par un administrateur de la plateforme :

- `NEW_PUBLICATION` : une URL, un message par URL, envoyé par `POST /api/v1/common-documents/urls` (`CommonDocumentResource`). Le contenu ne porte que `link` ; library télécharge le document et l'indexe. Le collecteur envoie le même type de message pour ses publications ;
- `NEW_COMMON_FILE` : un fichier entier, encodé en base64 dans le message (`fileName`, `contentType`, `content`, `metadata`, toujours vide), envoyé par `POST /api/v1/common-documents/upload`. La console n'utilise pas ce point d'entrée : elle téléverse les fichiers directement chez library (voir Les appels synchrones).

SUPPORT_REQUEST_CHANGED

Une demande ou une autorisation d'accès support d'un utilisateur a changé : ses applications rechargent leur bandeau de support sans attendre leur relecture périodique. Le contenu porte `tenantId`, `realm` et `username`. subscription le lit aussi, pour la console support. Voir Accès support.

Les enveloppes

Chaque topic a son enveloppe JSON. Toutes portent `schemaVersion` à `1` ; les lecteurs lisent les champs par leur nom, si bien qu'un champ ajouté ne les gêne pas.

Champ	`subscription_irrc_topic`	`tenant_lifecycle_topic`	`notification_topic`	`library_topic`	`ui_events_topic`
`messageType`	`UPDATE_PLAN`	`TENANT_DELETED`	`SEND_MAIL`	`NEW_PUBLICATION`, `NEW_COMMON_FILE`	le type d'événement
`schemaVersion`	`1`	`1`	`1`	`1`	`1`
`tenantId`	l'identifiant public du tenant	l'identifiant public du tenant	—	—	— (dans `content` quand il y en a un)
`content`	—	—	le mail	le document	l'événement

Champ	subscription_irrevocable_topic	tenant_lifecycle_topic	notification_topic	library_topic	ui_events_topic
`deletedAt`	—	l'instant de la purge	—	—	—

Classes : `GenericProducerMessage`, `TenantLifecycleMessage`, `NotificationMessage` (avec `MailMessage`), `LibraryMessage` (avec `PublicationContent` ou `CommonFileContent`) et `UiEventMessage` (avec `SupportRequestChangedContent`), dans `bus/message`.

```
{"tenantId":"11111111-1111-1111-1111-111111111111","messageType":"UPDATE_PLAN","schemaVersion":1}
```

```
{"tenantId":"12359bb8-b120-11ed-afa1-0242ac120005","messageType":"TENANT_DELETED",
 "deletedAt":"2026-11-04T03:30:00.123456Z","schemaVersion":1}
```

```
{"messageType":"NEW_PUBLICATION","schemaVersion":1,"content":{"link":"https://www.
example.com/doc.pdf"}}
```

```
{"messageType":"SUPPORT_REQUEST_CHANGED",
                                                    "content":{"tenantId":"12359bb8-
b120-11ed-afa1-0242ac120005","realm":"bpi","username":"s.bassin"},
 "schemaVersion":1}
```

> ⚠ Avertissement
>
> Les noms des types de message et les champs des enveloppes forment le contrat entre les services. Un type renommé ou retiré ne fait échouer aucune compilation : il casse la lecture chez les autres services, à l'exécution. Tout changement se décide avec la liste des lecteurs à livrer en même temps.

Les garanties d'envoi

Chaque producteur attend l'accusé de réception du courtier, au plus `application.kafka.send-timeout` (10 secondes par défaut, la valeur de chaque environnement venant de Consul). Ce qui se passe en cas d'échec dépend du message :

Message	Moment de l'envoi	En cas d'échec
`UPDATE_PLAN`	Dans la transaction de la modification, après son écriture	La modification est annulée et la requête échoue : un retrait de droits n'est jamais enregistré sans qu'irrc le sache.
`TENANT_DELETED`	Dans la transaction de la purge, après son `UPDATE_PLAN`	La purge est annulée.
`SEND_MAIL`	Après la validation de la modification	L'échec est consigné dans les journaux ; la modification reste. Pour les rappels de fin de contrat, l'échec annule l'enregistrement du rappel, retenté au passage suivant.
`NEW_PUBLICATION`, `NEW_COMMON_FILE`	Pendant la requête de l'administrateur	La requête échoue, avec l'erreur `error.messageTooLarge` pour un message trop gros, `error.unknown` sinon.
`SUPPORT_REQUEST_CHANGED`	Après la validation de la modification	L'échec est consigné et le message abandonné : les applica-

Message	Moment de l'envoi	En cas d'échec
		tions le rattrapent à leur relecture périodique.

Le courtier ne laisse écrire `notification_topic` qu'à des utilisateurs Kafka dédiés. `NotificationProducer` se connecte donc avec ses propres identifiants, `application.notification.kafka-username` et `application.notification.kafka-password` ; sans eux, il le signale une fois au démarrage et aucun mail ne part. Les autres topics utilisent l'utilisateur Kafka commun.

Un fichier téléversé voyage entier dans un message : les envois de fichiers sont limités à 50 Mo, et la taille des messages du producteur à 75 000 000 octets (un fichier de 50 Mo pèse environ 67 Mo une fois encodé en base64).

Les mises à jour en direct de la console

La console ouvre, par onglet du navigateur, un flux d'événements serveur (SSE) sur subscription : `GET /api/v1/events` (`text/event-stream`, classes `EventResource` et `SseHub` du paquet `sse`). Les événements sont des invalidations : ils disent « telle chose a changé, relisez-la » ; la page concernée relit alors ses données par l'API, qui vérifie les droits.

D'où viennent les événements

`UiEventsConsumer` lit `ui_events_topic` et transforme chaque message qu'il connaît en événement pour le navigateur. Chaque instance de subscription lit le topic dans un groupe qui lui est propre (`subscription-ui-events-<uuid>`), à partir des derniers messages : chaque instance voit tous les événements et les relaie aux navigateurs qu'elle sert. Rien n'est retenté ni mis de côté : un message illisible est ignoré. La propriété `application.kafka.ui-events.auto-startup` permet de ne pas démarrer ce lecteur.

Message Kafka	Événement envoyé (`nature`)	Données	Écran qui relit
`CRAWLER_RUN_CHANGED`	`CrawlerRunChanged`	`runId`	Collecteur, les collectes
`LIBRARY_DOCUMENT_CHANGED` (documents communs seulement)	`LibraryDocumentChanged`	`documentId`	Librairie, la liste des documents
`LIBRARY_EMBEDDINGS_CHANGED`	`LibraryEmbeddingsChanged`	—	Librairie, les embeddings
`LIBRARY_INDEX_CHANGED`	`LibraryIndexChanged`	—	Librairie, l'index de recherche
`DOCLING_QUEUE_CHANGED`	`DoclingQueueChanged`	—	Collecteur, l'onglet Docling
`SUPPORT_REQUEST_CHANGED`	`SupportAccessChanged`	`tenantId` (administrateurs de la plateforme seulement)	Console support

Un identifiant à `null` signifie que plusieurs éléments ont changé : la page relit toute la liste. Les messages de type inconnu, et ceux des documents propres aux tenants, ne sont pas relayés.

Le flux

- Chaque trame est un événement `message` dont les données sont un objet JSON `{"nature": …, …}`.
- La première trame d'un flux est `{"nature":"Connected"}` : la console sait alors qu'elle a pu manquer des événements et relit les pages ouvertes.
- Un commentaire `ping` part toutes les 20 secondes, bien en deçà du délai de lecture de 60 secondes de l'ingress.
- Un flux est fermé au bout de 30 minutes ; le navigateur se reconnecte.
- Un même événement reçu plusieurs fois en moins de 500 millisecondes ne part qu'une fois.

- Un envoi qui échoue ferme ce flux seulement.

Deux publics reçoivent le flux :

- les administrateurs de la plateforme reçoivent tous les événements, avec leurs identifiants ;
- les autres utilisateurs de la console support (membres du support, administrateurs de tenant) ne reçoivent que `SupportAccessChanged`, sans le tenant, pour qu'un flux ne leur révèle jamais un tenant qu'ils n'administrent pas.

Côté console

La console diffuse chaque événement sous le nom `event.<nature>` ; chaque page écoute ceux qui la concernent et `event.Connected`. Après une coupure, elle retente la connexion, avec au plus 60 secondes entre deux essais, y compris après un refus 401 ou 403.

Un point dans la barre du haut montre l'état du flux : vert, « Mises à jour en direct : connecté » ; orange après 5 secondes de coupure, « Mises à jour en direct : reconnexion en cours » ; rouge après une minute, « Mises à jour en direct interrompues — les données se rafraîchissent toutes les 30 s ».

Les pages ne comptent pas que sur le flux : elles relisent aussi d'elles-mêmes tant qu'un traitement est en cours.

Écran	Relecture périodique
Collecteur, collectes	toutes les 30 s tant qu'une collecte tourne
Collecteur, résumés	toutes les 10 s tant qu'un résumé est en cours
Collecteur, Docling	toutes les 10 s tant que l'onglet est ouvert
Librairie, documents	toutes les 30 s tant qu'un document est en attente, en conversion ou en vectorisation
Librairie, embeddings et index de recherche	toutes les 30 s tant qu'un traitement tourne
Référentiel, sources	toutes les 3 s tant qu'une mise à jour tourne (sans flux)
Activité Kafka DPM	toutes les 5 s tant que l'onglet est visible (sans flux)

Les appels synchrones

La console appelle chaque service depuis le navigateur, par la passerelle : l'adresse est celle de la passerelle, suivie du nom du service et du chemin, et l'appel porte le jeton de l'utilisateur. Le service appelé décide de ce que l'utilisateur peut faire : crawler et notification, par exemple, réservent leur API aux administrateurs de la plateforme et au compte de service interne, et notification réserve le renvoi d'un mail aux seuls administrateurs.

Appelant → service	Points d'entrée	Écran
console → crawler	`/api/v1/publications` (liste, `POST /send-to-library`) ; `/api/v1/feeds` (liste, création, modification, suppression, `POST /test`) ; `/api/v1/crawler/sources` (liste, `PUT /{id}`) ; `/api/v1/crawler/settings` et `/api/v1/crawler/schedule` (lecture, `PUT`) ; `/api/v1/crawler/runs` (liste, `POST` avec `source` facultatif, `/{id}`, `/{id}/publications`) ; `/api/v1/digest-recipients` (liste, création, modification, suppression, `/sources`, `POST /`	Collecteur

Appelant ⟶ service	Points d'entrée	Écran
	`{id}/test`) ; `/api/v1/digests` (liste, `/next`, `/{jour}`, `POST /{jour}/resend`)	
console ⟶ library	`/api/v1/crawler-summaries` (lecture par `documentIds`, `POST`) ; `/api/v1/documents/markdown/queue`	Collecteur, résumés et Docling
console ⟶ library	`/api/v1/documents/common` (liste, `DELETE /{id}`, `POST` d'une liste d'identifiants pour supprimer, `POST /upload`, `GET /{id}/markdown`, `POST /{id}/markdown/retry`, `POST /markdown/retry-missing`, `POST /{id}/embeddings`, `POST /embeddings/selection`, `POST /{id}/indexing`) ; `/api/v1/documents/common/embeddings` (état, `POST`) ; `/api/v1/documents/index/rebuild` (état, `POST`)	Librairie
console ⟶ subscription	`POST /api/v1/common-documents/urls`	Librairie, URL
console ⟶ notification	`/api/v1/mails` (liste, `/{id}`, `/{id}/attachments/{pièce}`, `POST /{id}/resend`)	Notifications
console ⟶ referential	`/api/v1/structure/dataflow`, `/api/v1/structure/datastructure` ; `/api/v1/bundles` (`POST` d'un zip, `PUT` et `DELETE /{agence}/{id}/{version}`) ; `/api/v1/loader/cubes` (liste, `POST`, `PUT` et `DELETE /{id}`, `/{id}/runs`, `POST /{id}/refresh`, `POST /{id}/upload`) ; `POST /api/v1/loader/refresh`	Référentiel
console ⟶ dpm	`GET /api/v1/admin/kafka-work`	Activité Kafka DPM
console ⟶ subscription	`GET /api/v1/tenants` (noms des tenants) ; `GET /api/v1/events` (flux)	Activité Kafka DPM ; toutes les pages
subscription ⟶ Keycloak	API d'administration de chaque realm, par son client de service (`subscription-provisioning` par défaut) : désactiver ou réactiver un compte (la désactivation ferme ses sessions), lire les connexions et les usurpations d'identité, vérifier un compte avant un accès support	Identités, Accès support

> **Note**
>
> Les URL de la bibliothèque commune passent par subscription, qui les met sur `library_topic` ; les fichiers, eux, vont directement à library. Une URL envoyée est donc entre les mains du courtier, un fichier envoyé entre celles de library.

Glossaire

Les termes de la console et de ce manuel, par ordre alphabétique, avec la page qui les décrit.

Termes

Terme	Définition	Voir
Accès	La place d'un utilisateur dans un tenant : un rôle et au plus un plan par service.	Utilisateurs
Accès suspendu	Accès dont le rôle et les plans sont conservés, mais qu'aucun service n'accepte plus.	Utilisateurs
Activité Kafka DPM	La vue en direct de ce que fait chaque consommateur Kafka du service dpm, et de ce qui attend derrière lui.	Activité Kafka DPM
Administrateur de la plateforme	Un compte d'un realm de la plateforme (`mla`, `shared`) portant le rôle `ROLE_ADMIN` ; le seul qui ouvre la console.	Prise en main
Autorisation (consentement)	La permission, limitée dans le temps, qu'un utilisateur donne depuis sa propre session pour une consultation de support.	L'accès du support
Bibliothèque commune	Les documents du service library que tous les tenants lisent et que l'assistant consulte.	Librairie
Clé maître	Clé avec laquelle subscription chiffre les credentials des tenants ; sa rotation se fait dans Vault.	Sécurité
Client	Une organisation cliente ; elle regroupe des tenants et des utilisateurs, et peut avoir un sous-domaine et un realm Keycloak.	Concepts clés
Client d'origine	Le client auquel un utilisateur appartient.	Utilisateurs
Client invité	Un autre client dont un utilisateur multi-client peut rejoindre les tenants.	Utilisateurs
Collecte	Un passage du collecteur sur ses sources, planifié, au démarrage ou lancé à la main.	Collecteur

Terme	Définition	Voir
Collecteur	Le service qui lit les sites et les flux RSS des régulateurs et transmet ce qu'il trouve à la bibliothèque commune.	Collecteur
Compte protégé	Compte qu'une consultation de support ne peut jamais emprunter (administrateurs, support, gestion du realm).	L'accès du support
Consultation (session de support)	Une session en lecture seule ouverte par la passerelle avec l'identité d'un membre, dans un tenant.	L'accès du support
Console support	La page où le support MLA et les administrateurs voient les membres d'un tenant, demandent un accès, fixent la politique et lisent le journal.	Console support
Contrat	Les dates de validité d'un tenant et les relances envoyées avant sa fin.	Clients et tenants
Credential de tenant	Le mot de passe de chiffrement d'un tenant, conservé chiffré par subscription, choisi à la création et jamais réaffiché.	Clients et tenants
Délai d'attente	La période (60 jours par défaut) pendant laquelle un tenant supprimé peut être restauré, avant sa purge.	Cycle de vie d'un tenant
Demande d'accès	La demande d'autorisation qu'un membre du support adresse à un utilisateur ; elle ne donne aucun accès par elle-même.	Console support
Désactiver un utilisateur	Couper tous ses accès et désactiver son compte Keycloak, en conservant ses rôles et ses plans.	Utilisateurs
Digest	Le résumé quotidien des nouvelles publications, envoyé par mail à chaque destinataire activé.	Collecteur
Document commun	Un document de la bibliothèque commune, ajouté par le collecteur ou par un administrateur.	Librairie
Enveloppe	L'objet JSON qui entoure un message Kafka (`messageType`, `schemaVersion`, `tenantId`, `content`…).	Événements

Terme	Définition	Voir
Fonctionnalité	Une capacité nommée d'un service, dont les applications vérifient le nom.	Fonctionnalités
Fonctionnalité `base`	Créée avec chaque service, elle signifie l'appartenance au service ; on ne peut ni la supprimer ni la renommer.	Fonctionnalités
Historique des accès	La trace de chaque changement de droits d'accès, conservée 90 jours.	Historique des accès
Intervenant	La personne qui ouvre une consultation : un membre du support MLA ou un administrateur du tenant.	L'accès du support
Jeton de service	Le jeton client-credentials qu'un service obtient pour appeler un autre service.	Sécurité
Journal support	Les événements d'accès du support dans un tenant, filtrables et exportables en CSV.	Console support
Mises à jour en direct	Le flux d'événements qui dit aux pages de la console de relire leurs données.	Prise en main
Multi-client	Un utilisateur qui peut être invité dans les tenants d'autres clients.	Utilisateurs
Plan	Une offre sur un service, qui ouvre une sélection de ses fonctionnalités (`base` toujours comprise).	Plans
Politique d'accès du support	La règle d'un tenant pour les consultations : sur demande, toujours ou jamais.	Console support
Purge	Le traitement qui passe un tenant arrivé au bout de son délai d'attente à `DELETED`, retire ses membres et envoie `TENANT_DELETED`.	Purge
Realm	Un annuaire de comptes Keycloak ; un utilisateur est le couple (realm, identifiant).	Identités et realms
Référence d'un tenant	L'identifiant public du tenant (`tenantId`), celui que les services conservent.	Clients et tenants
Référentiel	Le service des données de référence (cubes SDMX, annexes de l'ABE).	Référentiel

Terme	Définition	Voir
Relance de fin de contrat	Un mail envoyé avant ou à la fin de la validité d'un tenant.	Notifications
Revue des accès	L'export xlsx de qui détient quel plan sur quel tenant aujourd'hui.	Utilisateurs
Rôle dans le tenant	Membre ou Administrateur, valable dans ce tenant seulement ; l'administrateur gère aussi les accès des autres.	Concepts clés
Service	Un produit de la plateforme (`irrc`, `stress`, `library`…), que les applications retrouvent par son nom.	Services
Statut d'un tenant	`ACTIVE`, `SUSPENDED`, `PENDING_DELETION` ou `DELETED` ; seul `ACTIVE` donne accès.	Cycle de vie d'un tenant
`TENANT_DELETED`	Le message Kafka, sur `tenant_lifecycle_topic`, qui demande à chaque service de purger les données d'un tenant.	Cycle de vie d'un tenant
Tenant	Un environnement Reglab isolé d'un client, connu des services par son `tenantId`.	Concepts clés
`UPDATE_PLAN`	Le message Kafka, sur `subscription_irrc_topic`, qui dit aux services d'oublier les appartenances d'un tenant qu'ils ont en cache.	Des plans aux droits
Validité	La période (début incluse, fin exclue) en dehors de laquelle un tenant ne donne aucun accès.	Clients et tenants
Vectorisation	Le découpage d'un document en passages et leur plongement, pour que l'assistant le trouve par le sens.	Librairie
`X-TENANT-ID`	L'en-tête qui nomme le tenant d'une requête par son identifiant public.	Des plans aux droits