

Optimisation des transactions de consultation DMP

Version : V0.3 | Date : 26/05/2026

1. Objet

Dans le cadre de l'optimisation des transactions de consultation DMP, l'objectif de ce document est de décrire un guide pour les implémenteurs.

2. Définitions

2.1 DMP

2.1.1 Transactions utilisées

Les transactions utilisées dans ce flux sont des requêtes **IHE ITI-18 Stored Query** (TD3.1) et **IHE ITI-43 Retrieve Document Set** (TD3.2).

| Transaction | Type | Description |
|---------------------------------|----------------------------|---|
| <code>FindDocuments</code> | TD3.1 — Stored Query | Recherche de documents par critères (statut, date d'acte, type). Retourne les métadonnées XDS. |
| <code>FindSubmissionSets</code> | TD3.1 — Stored Query | Recherche des lots de soumission depuis une date donnée (<code>\$XDSSubmissionSetSubmissionTimeFrom</code>). |
| <code>GetAssociations</code> | TD3.1 — Stored Query | Retourne toutes les associations liées aux objets XDS passés en paramètre (par <code>entryUUID</code>). |

| Transaction | Type | Description |
|--|----------------------------|---|
| <code>GetDocumentsAndAssociations</code> | TD3.1 — Stored Query | Retourne les documents et leurs associations (RPLC, HasMember) à partir d'une liste d' <code>entryUUID</code> . Appelé par batch de 20. |
| <code>RetrieveDocumentSet</code> | TD3.2 — Retrieve | Télécharge le contenu binaire des documents à partir d'une liste d' <code>uniqueId</code> . |

Paramètres utilisés

| Paramètre | Transaction | Description |
|---|--|--|
| <code>availabilityStatus:urn:oasis:names:tc:ebxml-regrep:StatusType:Approved</code> | <code>FindDocuments</code> | Filtre sur les documents courants uniquement |
| <code>XSDocumentEntryServiceStartTimeFrom</code> | <code>FindDocuments</code> | Date de début de l'acte médical (filtrage sur 2 ans) (UTC) |
| <code>\$XSDocumentEntryTypeCode</code> (optionnel, multiple) | <code>FindDocuments</code> | Liste des typeCode à filtrer (optionnel) |
| <code>\$XDSSubmissionSetSubmissionTimeFrom</code> | <code>FindSubmissionSets</code> | Filtre sur la date (UTC) d'ajout dans le DMP à partir de laquelle on souhaite récupérer les nouveaux documents : renseigné avec la date de la dernière connexion du professionnel au DMP |
| <code>entryUUID</code> | <code>GetAssociations</code> , <code>GetDocumentsAndAssociations</code> | Identifiant technique affecté par le SI DMP |

2.1.2 Identifiants des documents utilisés

| Identifiant | Description |
|------------------------------|---|
| <code>uniqueId</code> | Identifiant logique globalement unique, généré par le logiciel. Présent dans le CDA et les métadonnées XDS. |
| <code>entryUUID</code> | Identifiant technique affecté par le SI DMP à une version de métadonnées d'un document. Change à chaque nouvelle version du document (remplacement) et à chaque modification de métadonnées (masquage/démasquage aux PS, remise en visibilité patient ou RL). |
| <code>logicalId</code> (lid) | Identifiant technique affecté par le SI DMP à un document (identique pour toutes ses versions de métadonnées). Change à chaque nouvelle version du document (remplacement) mais reste identique à chaque modification de métadonnées. |

2.2 Variables logiciel

| Variable | Type | Description |
|--------------------------------|----------------|--|
| <code>dateAppelDMP</code> | DateTime (UTC) | Date et heure de la dernière connexion au DMP pour ce patient. Initialisée à la première venue, mise à jour à chaque venue suivante. |
| <code>typeDMP</code> | Liste de codes | Liste des <code>typeCode</code> de documents DMP. Préférence de l'utilisateur. |
| <code>localDocumentsDMP</code> | Map | Index local des documents déjà stockés dans le LPS. Contient pour chaque document son dernier <code>entryUUID</code> , son <code>logicalId</code> et son <code>uniqueId</code> . |

3. Description du processus

Le processus se divise en deux branches selon qu'il s'agit de la première ouverture du dossier patient dans le logiciel ou des accès suivants.

3.1 Première ouverture

L'objectif est de détecter les documents présents dans le DMP qui ne sont pas encore connus du système.

3.1.1 Phase 1 — Recherche des documents

Le LPS enregistre la date et heure courante dans `dateAppelDMP`. Cette date servira de référence lors des accès suivants.

Une requête **FindDocuments** est envoyée au DMP pour récupérer les documents courants de moins de 2 ans.

Notes :

1. La première recherche documentaire automatique dans un DMP doit obligatoirement se faire sur une période maximale de 2 ans.
2. Uniquement sur un statut "courant" (`urn:oasis:names:tc:ebxml-regrep:StatusType:Approved`).
3. Dans le cas où le PS a des préférences de type de document renseigné (typeDMP) : sur une liste de type de documents particuliers. Le LPS doit indiquer que la recherche est filtrée sur ces types de document uniquement. Le PS doit pouvoir étendre sa recherche à d'autres types de document ou désactiver ce filtre s'il ne trouve pas les document recherchés.
4. Point d'attention sur le document particulier « Historique de vaccination » qui peut dans la majorité des cas avoir une date de début d'acte (plus ancienne vaccination) inférieure à 2 ans (vaccination COVID notamment). Un LPS doit pouvoir proposer au PS une recherche spécifique et manuelle sur ce typeCode sans limitation de date (si le document pas présent dans la première recherche automatique).

3.1.2 Phase 2 — Traitement (interne logiciel)

Note : Le même document CDA peut être transmis via MSS et déposé sur le DMP. Le `uniqueId` étant identique dans les deux cas, le LPS doit systématiquement vérifier qu'un document n'est pas déjà présent en local avant tout import.

Pour chaque document retourné, le système vérifie que son `uniqueId` n'est pas déjà présent dans les documents en local.

Les documents dont le `uniqueId` correspond à des documents déjà présents en local (documents reçus par MSS, ...) sont ajoutés à `localDocumentsDMP` (`entryUUID`, `logicalId`, `uniqueId`).

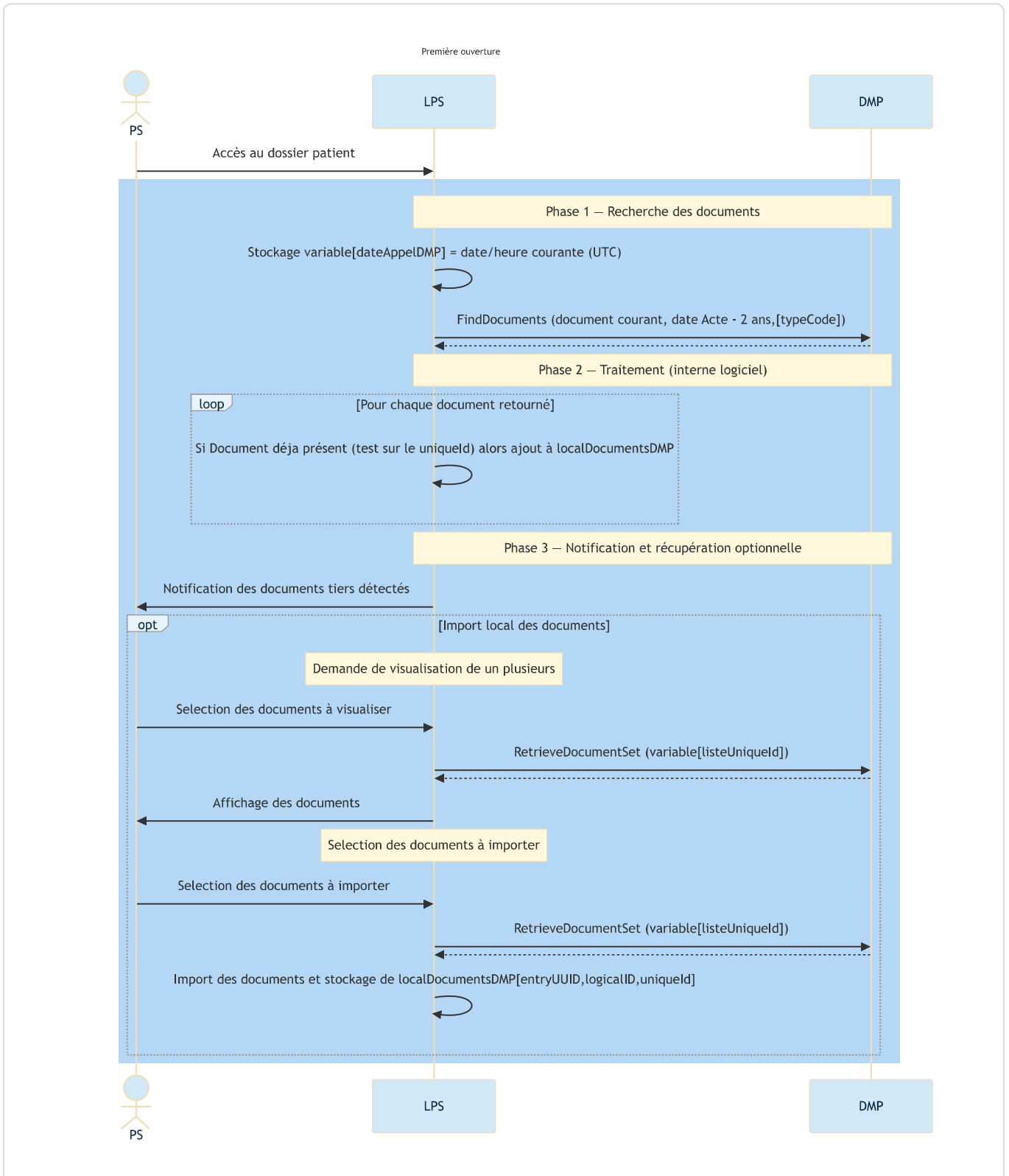
3.1.3 Phase 3 — Notification et récupération optionnelle

Le PS est notifié du nombre de documents tiers détectés.

Le PS peut visualiser un ou plusieurs documents via `RetrieveDocumentSet`.

Si le PS souhaite importer des documents DMP en local, le LPS appelle `RetrieveDocumentSet` et stocke les documents en local avec leurs identifiants (`entryUUID`, `uniqueId`, `logicalId`) dans `localDocumentsDMP`.

3.1.4 Logigramme



3.2 Accès suivants

L'objectif est de détecter uniquement les changements survenus depuis la dernière connexion :

- Nouveaux documents
- Remplacements de documents ou documents ayant changé de confidentialité

3.2.1 Phase 1 — Recherche des nouveaux lots

`dateAppeIDMP` est mis à jour avec la date/heure courante (UTC).

Le système appelle `FindSubmissionSets` (`$XDSSubmissionSetSubmissionTimeFrom` = `dateAppeIDMP`) pour récupérer tous les lots soumis depuis la dernière connexion.

Il appelle ensuite `GetAssociations` sur les `entryUUID` des lots retournés, puis filtre les associations de type **HasMember** pour extraire les `entryUUID` des documents concernés.

3.2.2 Phase 2 — Récupération des métadonnées

Les documents sont récupérés par lot de 20 via `GetDocumentsAndAssociations`, qui retourne les métadonnées et les associations (notamment `RPLC`). Les « résultats finaux » sont constitués de l'agrégat des résultats des différents lots éventuels.

Note : Rendre paramétrable le nombre de documents pouvant être récupéré via la query `GetDocumentsAndAssociations` (par défaut 20).

3.2.3 Phase 3 — Analyse des résultats finaux

Pour chaque document, ignorer ceux qui sont au statut **Deprecated** (il s'agit d'une version antérieure du document déjà remplacée par une plus récente également retournée dans les résultats, ou d'une ancienne version de métadonnée obsolète). Deux cas sont ensuite traités :

- **Cas A — Document remplacé** : une association **RPLC** existe et son `targetObject` correspond à un document déjà en local (`localDocumentsDMP[entryUUID]`). Le document local est marqué comme remplacé. Si aucun document "Non Deprecated" n'est trouvé directement dans la liste des nouveaux documents (cas où le nouveau document est également lui-même remplacé), la chaîne RPLC doit être remontée (via les associations RPLC et les `sourceObject/targetObject`) pour trouver la version la plus récente du document qui n'est pas Deprecated et qui se trouve dans la liste des nouveaux documents.
- **Cas B — Nouveau document ou mise à jour de métadonnées** :
 - Si le `logicalId` du document récupéré correspond à un document déjà en local (`localDocumentsDMP[logicalId]`), le document a subi une modification de métadonnées par un autre acteur (masquage/démasquage au PS, remise en visibilité patient ou représentant légal). Le système met à jour le code de confidentialité en local et en informe le PS. Il met à jour les références du document avec ses nouveaux identifiants (nouvel `entryUUID` , `uniqueId` inchangé, `logicalId` inchangé).
 - Sinon, il s'agit d'un **nouveau document** : statut non `Deprecated` (`Approved` ou `Archived`), sans association RPLC pointant vers un document local. Il est ajouté à `localDocumentsDMP` .

Note : Lors de l'analyse des résultats et avant tout import, le LPS doit vérifier que le document n'est pas déjà présent en local via son `uniqueId` . Les documents dont le `uniqueId` correspond à des documents déjà présents en local (documents reçus par MSS, ...) sont ajoutés à `localDocumentsDMP` (`entryUUID`, `logicalId`, `uniqueId`).

3.2.4 Phase 4 — Mise à jour et notification

Le PS est notifié des documents nouveaux ou remplacés.

Il peut ensuite les consulter via `RetrieveDocumentSet` .

Si le PS souhaite importer des documents DMP en local, le LPS appelle `RetrieveDocumentSet` et stocke les documents en local avec leurs identifiants (`entryUUID` , `uniqueId` , `logicalId`) dans `localDocumentsDMP` .

3.2.5 Logigramme

