Interopérabilité du notariat

API Interop

Révisions du document

1. 14/12/18 : Version originale (RL).
1.1 04/09/19 : Ajouts d'informations contextuelles (RL).
1.2 06/10/19 : Mise en page, référence juridique (JM).
1.3 13/10/19 : Ajouts du lien de comparants, et détail de la gestion des droits (RL).
1.4 13/10/19 : Relecture et ajout d'une conclusion (JM)
1.5 30/03/20 : Mis à jour suite retours d'implémentation (RL)

Contacts

NS-SOFT : Romain Lagrange (Lead Dev) romain.lagrange@notasolutions.com 06.85.87.78.16
NS-SOFT : Jacques Massa (Gérant) jacques.massa@notasolutions.com 06.15.38.63.67

Table des matières

Introduction

Ce document a pour but de servir de référence pour l'élaboration de normes à implémenter par les partenaires techniques autour du métier du notariat.

Ces normes s'appliquent autant aux éditeurs de LRA qu'aux legaltech.

Ce document est une première version proposée par NS-SOFT, et pourra être enrichi en collaboration avec d'autres partenaires et encadré par le CSN.

CONTEXTE

Le groupe de travail « interopérabilité » du CSN propose un format de données normalisé pour échanger des informations entre les différents systèmes informatiques du notariat.

Ce format décrit les dossiers, les actes, les biens immobiliers, les personnes physiques et morales, et les autres pièces électroniques. Ce document ne reprend pas ces descriptions. Il est inspiré des formats Télé@cte et ce veut complet dans ses descriptions. Un schéma simplifié est fourni en 4/.

Néanmoins, ce format de données, ne suffit pas à proposer une expérience « fluide » à l'utilisateur clerc ou notaire, qui est contraint à passer par des opérations d'importations et d'exportations de fichiers pour transférer ses données entre ses divers logiciels.

Nous proposons ci-après une méthode pour normaliser la façon dont ces données peuvent être échangées par les différents acteurs tout en utilisant ce format de données, à travers un ensemble de méthodes informatiques (API Interop) devant couvrir l'essentiel des besoins.

Par ailleurs, comme certains logiciels sont installés sur le réseau local de l'étude, ce document décrit un moyen standardisé de communication avec ces systèmes.

Intérêts pour les parties

POUR LA PROFESSION

Cela permet de mettre en œuvre l'interopérabilité requise, et de proposer aux éditeurs un cadre fonctionnel et technique leur permettant d'être en conformité avec les obligations de la charte du CSN et du code de la propriété intellectuelle :

- L'article 16 du décret n°2005-973 du 10 août 2005 relatif aux actes établis par les notaires qui prévoit que « Les systèmes de communication d'informations mis en œuvre par les notaires doivent être interopérables avec ceux des autres notaires et des organismes auxquels ils doivent transmettre des données ».
- L'article 4 intitulé « Travail collaboratif et concurrence saine et loyale » de la Charte du Conseil supérieur du Notariat qui impose à ses signataires, dans le dessein de favoriser l'interopérabilité des systèmes, de « mettre leurs compétences au service de l'innovation et de favoriser entre eux des échanges ouverts et collaboratifs, afin de favoriser au mieux de leurs possibilités respectives le développement des services et des technologies associées » ;
- Le caractère d'ordre public de l'exception d'interopérabilité des logiciels prévue par l'article L. 122-6-1 du Code de la propriété intellectuelle.

POUR LES CLERCS ET NOTAIRES

Permettre à chaque utilisateur d'accéder à ses données à partir de tous les logiciels dont il a acquis les droits sans être contraint par un hypothétique partenariat entre la legaltech et de son éditeur Logiciel de Rédaction d'Actes (LRA) ou au bon vouloir de ce dernier.

L'utilisateur conserve le privilège d'accorder un droit d'accès à ses propres données pour un éditeur et un service qu'il a lui-même choisi, sans dépendre exclusivement de l'éditeur de LRA. Pour rappel,

l'éditeur du LRA n'est pas le propriétaire des données contenues et insérées par l'utilisateur dans son LRA.

Pour les éditeurs de Logiciel de Rédaction d'Actes (LRA)

Garantir l'interopérabilité de son logiciel avec les autres logiciels respectant les mêmes engagements fixés par la charte du CSN et le cadre légal.

L'intégration de nouveaux partenaires est facilitée par la description fonctionnelle, technique et normalisée des API Interop.

Enfin, l'accès offert sera maitrisé par l'éditeur, sécurisé par ses soins et auditable. L'éditeur pourra y implémenter ses règles métiers afin de mieux encadrer l'interopérabilité.

Pour les Legaltechs

Permettre à leurs clients clercs et notaires une interopérabilité pérenne et normalisée entre leur logiciel et le logiciel métier de l'étude, évitant les doubles-saisies et les nombreux risques d'erreurs.

Le collaborateur de l'étude pourra, depuis le logiciel de la Legaltech, rechercher son dossier du LRA et importer les éléments et documents dans le dossier de la Legaltech.

Inversement, une fois le dossier Legaltech complet, le collaborateur pourra le déverser dans son LRA.

Principes de l'API Interop

L'API Interop permet d'effectuer toutes les fonctions courantes de gestion de dossier : rechercher, lire, créer, modifier, les éléments définis dans la norme : dossier, clients, immeuble, actes, documents.

Ces fonctions étant implémentées par chaque logiciel, chaque partie peut donc implémenter ses règles de gestion en toute autonomie.

l'API Interop se présente sous forme d'un service http(s) REST respectant **OPEN API Specification 3.0** et hébergé sur l'instance du serveur, qu'il soit Cloud ou Local afin de garantir l'étendue la plus large possible des cas d'utilisations.

AUTHENTIFICATION

L'utilisateur connecté est forcément un utilisateur de l'étude. Il n'est pas question ici de donner un accès direct entre systèmes sans l'accord explicite de l'utilisateur.

Le serveur REST doit imposer une authentification de l'utilisateur de l'étude. Nous proposons le protocole ouvert : Oauth car il permet de répondre aux besoins suivants :

- La fenêtre d'authentification est gérée par l'éditeur du système interrogé, de manière à ne pas avoir à donner au partenaire la possibilité de stocker un mot de passe attaquable.
- Le jeton authentifie un utilisateur du système interrogé, dont les données chiffrées sont à la discrétion de l'implémentation.
- Le jeton (refresh_token dans le cas de Oauth) n'expire pas, ou son expiration longue est repoussée à chaque interrogation

Néanmoins, le protocole Oauth est conçu pour contenir les informations de droits d'accès encodées, ce qui nous semble inapproprié étant donné la granularité que nous souhaitons mettre en place plus bas. Il peut donc être préférable de stocker le jeton coté serveur, afin de maintenir l'ensemble des droits associés au jeton.

Oauth permet un accès direct entre systèmes si un accord est passé entre deux éditeurs, il suffit alors de fournir un jeton pour un compte système, avec un rôle permettant d'outrepasser la gestion des droits.

GESTION DES DROITS

Par défaut, l'application doit obligatoirement s'authentifier pour effectuer une requête ; si elle n'est pas authentifiée le service retourne HTTP 401, avec l'adresse de la page web permettant à l'utilisateur de donner le droit au service d'accéder à la ressource demandée.

La page web affiche cette page web en « popup ». L'utilisateur se connecte éventuellement, puis peut autoriser ou non l'opération demandée par l'application, pour une durée limitée ou non.

La « popup » redirige vers le service Oauth du service appelant, avec le jeton d'accès pour enregistrement.

La page web réitère alors la demande de ressource, avec le jeton d'authentification ainsi obtenu.

Pour chaque nouvelle ressource demandée, la même demande d'autorisation pourra être faite. La restriction d'accès au système est donc définie par l'utilisateur.

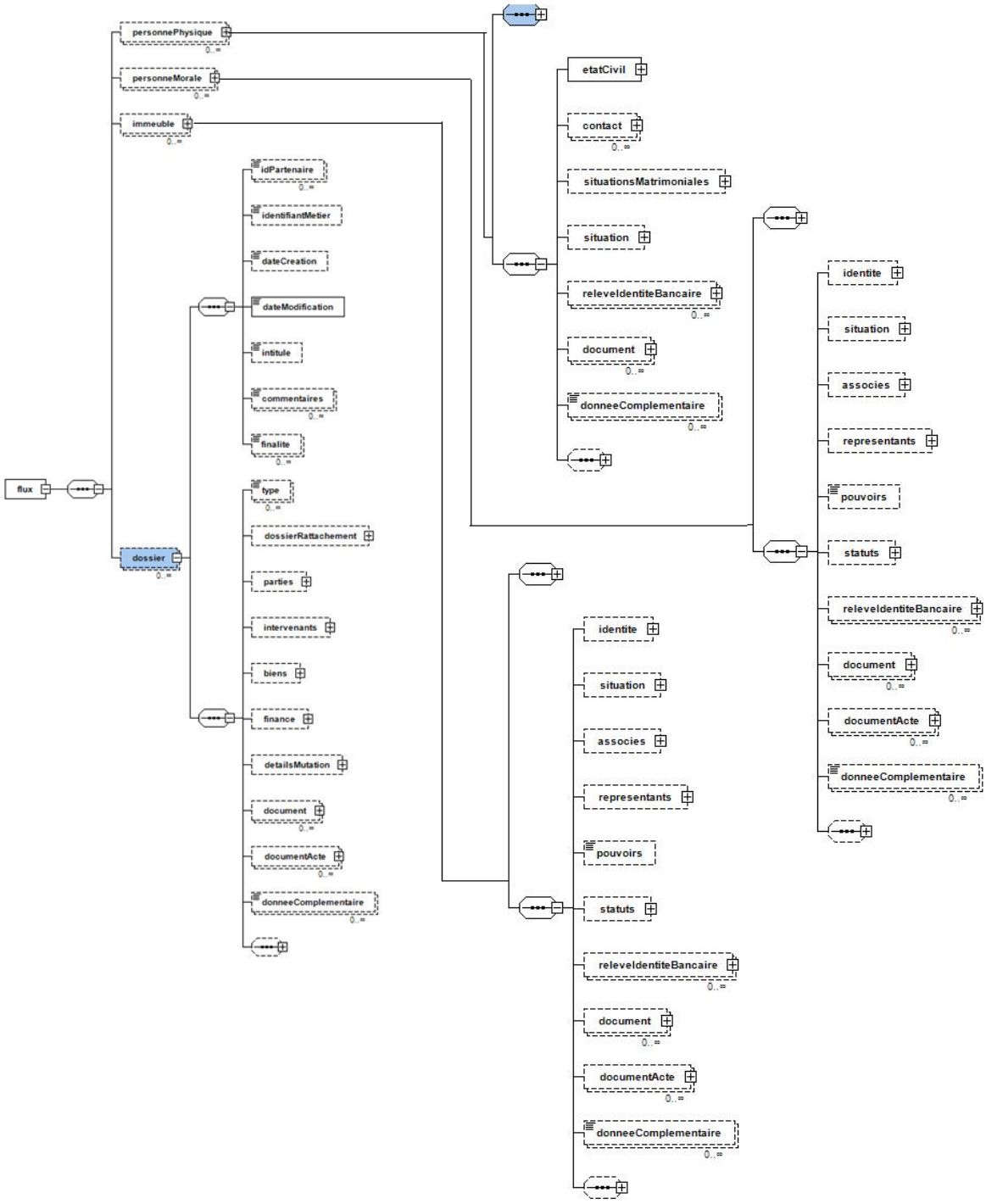
La gestion des droits n'est pas normée et laissée libre à l'implémentation. Nous avons néanmoins analysé cette partie, dont l'explication est disponible au chapitre dédié.

Schéma simplifié des données

Ce schéma est susceptible d'évoluer avec les itérations du groupe de travail Interopérabilité.

Consulter les schémas XSD pour avoir l'ensemble du détail.

Sur le principe, un flux contient un ou plusieurs dossiers, ainsi que les fiches clients et immeubles auquel le dossier fait référence. Les métadonnées des fichiers sont fournies dans les propriétés « documentActe » et « document », mais -en l'état- pas le binaire du fichier.



Formats de retour

Généralités

Afin de faciliter l'implémentation par les tiers, les implémentations doivent respecter l'OPEN API SPECIFICATION 3.0 : https://github.com/OAI/OpenAPI-Specification et devrait idéalement proposer une interface web (de type swagger) afin de pouvoir tester les accès.

Pagination

Les méthodes de recherches peuvent retourner un grand nombre de résultats. Il est donc important de pouvoir effectuer de la pagination des données retournées afin de naviguer facilement dans le jeu de résultats.

Pour ce faire les informations minimales suivantes sont utilisées :

- Entrée/sortie : page, taillePage,
- Sortie seule : total.

Les paramètres d'entrées sont optionnels, signifiant des valeurs par défaut (exemple page=1, taillePage=30).

- REQUEST interop/1.0/recherche/dossiers/test?page=1&taillePage=30
- RESPONSE { page :1, taillePage:30, total :65, elements :[] }

La liste paginée est dénommée ListePaginee<TYPE> dans le reste du document.

Objet de Transfert de Données (DTO)

Ces objets sont ceux de la norme. Consultez les XSD pour une vue complète, sinon un schéma simplifié est disponible plus haut. Certains points ne sont pas traité par cette norme. Notamment :

Pas de distinction entre pièces (annexes), courriers et sous-produit. Il faut pouvoir les différencier par leur propriété « nature » du type Document, qui se présente sous la forme d'un Enum.

La norme ne décrit pas non plus la notion de « classeurs » ou « tag » du dossier des LRA comme pour iNot ou Signature, permettant de trier la masse de documents du dossier.

La notion de dossier et sous-dossier est absente de la norme. Nous recommandons à tout les LRA disposant de cette organisation, de donner directement accès au sous-dossier en ignorant la notion de dossier. Cette recommandation est succeptible de changer avec les évolutions de la norme.

Pour les recherches, le type final est résumé aux données de FicheType dans les réponses. Les données présentes dans cet objet permettent de déterminer comment interroger le détail de l'objet.

Codes erreurs

Les erreurs sont signalées par un état HTTP, accompagné d'un DTO json ou xml au format RFC 7807 « ProblemDetails ».

Exemple :

```
{   "title": "Accès non autorisé",
    "detail": "L'utilisateur n'a pas autorisé la recherche de dossiers.",
    "instance": "/interop/autorisations/dossiers/recherche?token=123456"
}
```

Cas d'erreurs normés :

- 400 : si les paramètres fournis sont incorrects, un code HTTP 400 sera émis : mauvaise requête.
- 401 : Si l'utilisateur n'a pas autorisé l'accès à la ressource, émission d'un code HTTP 401 : unauthorized. Dans le DTO de retour, la propriété instance DOIT contenir l'url pour demander l'autorisation d'accès à la ressource (voir exemple ci -dessous).
- 401 : Si l'utilisateur n'a pas encore configuré son LRA, login/password du LRA, ou demandé un accès OAUTH, émission d'un code HTTP 401. Idem, le DTO de retour DOIT contenir l'url pour permettra à l'utilisateur d'accéder à la configuration ou à l'écran OAUTH d'authentification.
- 403 : si l'utilisateur n'a pas le droit d'accès à la ressource dans le LRA, HTTP 403 : forbidden.
- 404 : Si l'objet demandé n'a pas été trouvé, code HTTP 404 : not foud. NB : les recherches doivent retourner une liste vide si aucun résultat (ce n'est pas une erreur).
- 500 : Les erreurs non gérées côté serveur donnent lieu à l'émission d'un code HTTP 500 : internal error, si possible en fournissant un message et une stacktrace afin de faciliter le dialogue entre les services informatiques de l'appelant et de l'appelé.
- 501 : Si la fonction existe mais n'est pas implémentée pour le service demandé, émission d'un code HTTP 501 : not implemented

Liste des méthodes

RÉSUMÉ

Les méthodes proposées permettent de rechercher, lire, créer, mettre à jour et supprimer les objets de la norme.

Fichier swagger.json correspondant aux méthodes ci-dessous au format OpenApi :


swagger.json

RECHERCHES

Lorsqu'aucun critère n'est passé, la méthode retournera les derniers éléments consultés par l'utilisateur connecté, ou à défaut les derniers éléments modifiés dans le système.

- GET interop/1.0/recherche/dossiers/{critères} : ListePaginee<FicheType>
- GET interop/1.0/recherche/actes/{critères} : ListePaginee<FicheType>
- GET interop/1.0/recherche/personnes/{critères} : ListePaginee<FicheType>
- GET interop/1.0/recherche/immeubles/{critères} : ListePaginee <FicheType>

Le critère est une phrase saisie par l'utilisateur. Il est attendu que chacun des mots de cette phrase soit utilisé pour filtrer l'index de cette table. Exemple : « monnom monprenom » interroge « where champIndex like '%monnom%' and champIndex like '%monprenom%' »

Pour rappels pour les LRA utilisant une notion de dossier et sous-dossier, comme l'interop ne fournit pas cette abstraction il est recommandé de fournir directement des sous-dossiers dans la recherche de dossiers.

OBJETS MÉTIER

Détail d'objet métier

- GET interop/1.0/objet/flux/{iddossier} : flux fournit le détail d'un dossier et tous ses éléments
- GET interop/1.0/objet/dossier/{id} : dossier

NB : Dans un but d'optimisation de performances, le dossier peut être demandé sans l'intégralité de ses éléments. Paramètres optionnels : ?avecActes=false&avecDocuments=false&avecFinances=false

- GET interop/1.0/objet/personne/{id} : personnephysique/morale/contact
- GET interop/1.0/objet/immeuble/{id} : Immeuble

NB : Idem pour ces éléments, paramètre optionnel : &avecDocuments=false pour obtenir d'éventuels documents enfants

- GET interop/1.0/objet/dossier/{id}/acte/{id} : métadonnée de l'acte DocumentActe
- GET interop/1.0/objet/dossier/{id}/document/{iddo} : metadonnées du DocumentSimple
- GET interop/1.0/objet/personne/{id}/document/{iddoc} : idem
- GET interop/1.0/objet/immeuble/{id}/document/{iddoc} : idem

Création d'objets

Les mêmes méthodes que pour le détail GET d'objet métier, via un POST avec le DTO en corps.

L'API Interop doit retourner l'id de l'objet créé, en corps de la réponse.

- POST interop/1.0/objet/flux : créer l'ensemble d'un coup
- POST interop/1.0/objet/dossier
- POST interop/1.0/objet/personne/physique
- POST interop/1.0/objet/personne/morale
- POST interop/1.0/objet/personne/contact
- POST interop/1.0/objet/immeuble

L'appelant à la responsabilité de rechercher les doublons avant de créer un nouvel objet ici.

NB : L'ajout d'actes et documents est traité à part. Voir Gestion des fichiers.

Suppression d'objet (optionnel)

Mêmes méthodes que GET d'objet métier, avec une commande DELETE.

Le support de cette commande serait optionnel dans la norme, car il est compréhensible que les éditeurs ne souhaitent pas courir le risque de suppression d'objets via un programme externe.

En tout état de cause, la suppression serait dans tous les cas généralement bloquée par des règles métiers une fois un acte signé.

Modification d'objet

Étant donné que les DTO de la norme seront sans doute incomplets par rapport aux détails des objets dans les systèmes des intervenants, le remplacement « tel quel » est exclus.

Les commandes seront donc traitées en PATCH, qui ne modifierons que les données fournies.

Trois formats proches sont exploitables, qu'il faudra départager. Vous avez ci-dessous les avantages et inconvénients de chacun.

- PATCH + ContentType=application/json (ou xml)
  - Les éléments non fournis ou non modifiés seront NULL
  - Il n'est pas possible de vider une propriété déjà présente
  - Très simple à implémenter
- PATCH + ContentType=application/merge-patch+json (ou xml)
  - Le corps sera exprimé en RFC 7396 reprenant la forme du dto mais dont les nœuds et attributs non touchés sont supprimés
  - Requiert une implémentation particulière
  - Il n'est pas possible de vider une propriété déjà présente
- PATCH + ContentType=application/json-patch+json (ou xml-patch+xml)
  - Le corps sera exprimé en RFC 6902 (json) ou 5261 (xml), qui décrit des différentiels
  - Requiert une implémentation particulière
  - Ce format supporte la suppression de propriétés déjà présentes

Le support de PUT (remplacement intégral) serait optionnel, et n'aurait d'utilité que pour les cas des plateformes ne faisant qu'échanger des objets de la norme sans traitement (Échange entre confrère, stockage temporaire).

- Un PUT doit avoir en corps le dto intégral.

Voici les commandes PATCH et PUT :

- interop/1.0/objet/dossier/{id}
- interop/1.0/objet/personne/{id}
- interop/1.0/objet/immeuble/{id}
- interop/1.0/objet/dossier/{id}/acte/{idacte}
- interop/1.0/objet/dossier/{id}/document/{iddoc}
- interop/1.0/objet/personne/{id}/document/{iddoc}
- interop/1.0/objet/immeuble/{id}/document/{iddoc}

NB : pour tout ce qui concerne les actes, les LRA ne devraient autoriser les modifications que dans les actes libres et les fiches signature sans rédaction, qui ne sont pas déjà signées. Ceci pour sauvegarder la capacité à teleacter d'une part, et à garantir l'intégrité des pièces une fois la fiche signée.

LIER DES CLIENTS ET DES IMMEUBLES AU DOSSIER

Ceci permettra au service appelant, d'ajouter au dossier le comparant tout juste créé ou identifié

Création

- POST interop/1.0/lier/dossier/{id}/personne/{idpers}/partie (PartieType)
- POST interop/1.0/lier/dossier/{id}/personne/{idpers}/intervenant (IntervenantType)
- POST interop/1.0/lier/dossier/{id}/immeuble/{idpers}

NB : selon les systèmes, le role/type n'est pas déterminé au niveau du dossier mais au niveau de l'acte, et ne pourra donc pas être pris en compte.

NB : ceci ne couvre pas l'information de présence à la comparution, ou a la complexité liée à l'état marital (présence conjoint). Il sera nécessaire que l'utilisateur vérifie le lien créé par le système.

Suppression

- DELETE interop/1.0/lier/dossier/{id}/personne/{idpers}/partie ?type={type}
- DELETE interop/1.0/lier/dossier/{id}/personne/{idpers}/intervenant ?role={role}
- DELETE interop/1.0/lier/dossier/{id}/immeuble/{idpers}

Le type ou role est optionnel, et sert à départager le lien dans l'hypothèse ou une même partie intervient plusieurs fois à l'acte.

GESTION DE FICHIERS

Téléchargement de documents

Doit permettre le chargement d'un document d'un dossier, d'une fiche client, immeuble, etc.

- GET interop/1.0/telecharger/dossier/{id}/acte/{iddoc}
    - Paramètre optionnel : ?type={auto|projet|minute|copieaae|copieavecannexe}
    - Valeur par défaut pour auto => copieaae si dispo, sinon minute, sinon projet
- GET interop/1.0/telecharger/dossier/{id}/document/{iddoc}
- GET interop/1.0/telecharger/personne/{id}/document/{iddoc}
- GET interop/1.0/telecharger/immeuble/{id}/document/{iddoc}

Le corps de la réponse contiendra le fichier binaire, avec en header le nom du fichier : voir RFC 1806 ou 2616-19.5.1 : Content-Disposition: attachment; filename="fname.ext"

Chargement de documents

Le corps de la requête « multipart form/data » contient les données permettant de reconstruire un DocumentSimple ou DocumentActe.

Pour le transfert du binaire, il peut soit être fourni en « file » de la requête http, soit être transmis sous la forme d'une adresse URL pour que le service récupère le fichier à cette adresse.

- nomFichier
- fichier (type « file » de la requête http)
- adresse (adresse ou télécharger le fichier, accessible anonymement)
- mimetype
- checksum (optionnel, sha256)
- idPartenaire (id dans le système source)
- Pour les actes, la propriété supplémentaire « nature » du DocumentActe

Les commandes POST retournent en corps de la réponse, l'id de l'objet créé.

Si le checksum calculé côté serveur ne correspond pas au checksum passé , une erreur HTTP400 avec la description de l'erreur sera retournée et aucune création n'aura lieu.

Créer :

- POST interop/1.0/telecharger/dossier/{id} : ajout d'un document au dossier
- POST interop/1.0/telecharger/dossier/{id}/acte/{idacte} : ajout d'un document à l'acte
- POST interop/1.0/telecharger/dossier/{id}/document/{iddoc} : ajout d'un doc au doc (exemple du courrier word signé et envoyé). Si non supporté, ajouter la pièce au dossier.
- POST interop/1.0/telecharger/dossier/{id}/acte : ajout d'un acte libre/fiche signature
- POST interop/1.0/telecharger/personne/{id} : ajout d'un doc au client
- POST interop/1.0/telecharger/immeuble/{id} : ajouter d'un doc au bien

Ecraser :

- PUT interop/1.0/telecharger/dossier/{id}/document/{iddoc} : maj d'un doc d'un dossier
- PUT interop/1.0/telecharger/personne/{id}/document/{iddoc} : maj d'un doc du client
- PUT interop/1.0/telecharger/immeuble/{id}/document/{iddoc} : maj d'un doc du bien
- PUT interop/1.0/telecharger/dossier/{id}/acte/{idacte} : met à jour un acte libre/fiche sign

Note pour les annexes :

Elles doivent si-possible garantir la conformité PDFA/1B imposée par le MICEN.
Le logiciel de rédaction d'acte est libre d'effectuer automatiquement la transformation en PDFA1/B si le document n'est pas signé numériquement.

Notes pour les actes :

Les rédactions d'actes doivent pouvoir refuser la modification d'actes selon leur règles internes, notamment dans le but de maintenir la capacité à télé@cter.

Lors d'une création :

- Si document WORD : la rédaction d'acte doit être capable de reforger un projet d'acte sur un modèle sans trame « acte libre » à partir du document Word fournit.
- Si document PDF : c'est une « Fiche signature sans rédaction » qui devrait être créée (process du notaire en concours en charge des formalités et non de la rédaction). Le PDF correspond alors à la minute signée dans l'autre étude.
- La création d'acte devrait échouer si un autre type de fichier est fourni.

Gestion des droits

La gestion des droits ne fait pas l'objet de la norme, et relève de la gestion interne à l'implémentation. Néanmoins, une analyse est proposée ci-dessous :

La granularité des droits se fait sur deux dimensions :

- Droit global sur un type (Dossier, Personne, Immeuble)
- Droit spécifique sur un objet en particulier

Voici la liste des niveaux d'accès identifiés

- Aucun (par défaut)
- Recherche (droit global uniquement)
- Créer (droit global uniquement)
- Lire les métadonnées de l'objet
- Lire les documents
- Lire aussi les actes
- Ajouter et modifier des fichiers
- Lier des personnes et immeubles au dossier
- Modifier les métadonnées
- Supprimer

NB : Les droits peuvent être limités dans le temps.
NB : les droits sont cumulatifs : le droit de modification donne tous les droits précédents.

Par exemple :

- L'utilisateur X autorise à l'application A le droit de rechercher des dossiers
    - /interop/autorisations/dossiers/recherche?token={refreshtoken}
- X autorise à A le droit d'ajouter des fichiers au dossier N jusqu'à la date D
    - /interop/autorisations/dossier/{N}/ajouter?token={refreshtoken}

NB : afin de faciliter la navigation, la page autorisant l'accès au dossier devrait aussi définir automatiquement les mêmes droits pour les fiches clients et immeubles associées.

Points remontés par les partenaires

Représentants fiscaux

Les représentants fiscaux souhaitent pouvoir envoyer la PVI pré-remplie, prête à être Télé@ctée. Le schéma de données ne prenant pas en compte ce cas, les fonctions suivantes sont proposées :

- GET interop/1.0/telecharger/dossier/{id}/pvi/{iddoc} : télécharger la pvi
- PUT interop/1.0/telecharger/dossier/{id}/pvi/{iddoc} : modifier la pvi
- POST interop/1.0/telecharger/dossier/{id}/pvi : créer la pvi (document de nature pvi + fichier)

Le corps contiendrait le fichier XML au format Tele@cte.

Le PUT n'est autorisé que si la PVI n'a pas été déposée.

La PVI apparaitrait par ailleurs comme un document de l'objet dossier.

A la création par cette méthode, le projet de PVI est créé dans la rédaction d'acte. Le formaliste devra vérifier, puis procéder à l'opération Télé@cte.

Outils de planifications

Les éditeurs proposant aux notaires un service « à la doctolib » de prise de rendez-vous, souhaitent pouvoir synchroniser leur calendrier avec celui du notaire inclus dans le logiciel de rédaction d'acte.

Ce point est assez délicat, car la gestion des évènements récurrents est d'une grande complexité, néanmoins bien traités par certaines normes (ical : rfc 5545, mais aussi MS Exchange)

NeoNotario, NotaStart, izilaw, notre-notaire, et il semble l'ADSN souhaitent se positionner.

Conditions d'accès à l'API Interop

Ce point est à définir par le CSN, et ne fait l'objet ici que d'une suggestion.

Dans l'idée, nous proposons que le CSN gère la liste des legaltech labellisées et autorisées à se connecter aux API Interop, afin de ne pas laisser ce choix au bon vouloir de chaque éditeur. Cela permet aussi de centraliser cette gestion, sans avoir besoin de synchroniser tous les éditeurs.

Cette décision est du ressort du CSN et peut être débattu, le cas échéant, au sein du groupe de travail interopérabilité

Voici un ensemble de propositions :

- Le CSN génère une clef privé/publique, et communique librement la clef publique auprès des éditeurs souhaitant implémenter la partie serveur.
- Le CSN génère des clés API pour les éditeurs souhaitant implémenter la partie client, sous réserve de labellisation. Cette clé API est générée par chiffrement du nom du service, par la clef privée.
- Le CSN gère par ailleurs une liste de révocation de ces API Key.
- Les serveurs d'interopérabilité vérifient l'API Key avec la clef publique et le nom du service passé en header des requêtes avant d'autoriser l'appel. Ils mettent aussi à jour périodiquement la liste de révocation du CSN.

Accès aux serveurs LRA locaux

Les API REST sont facilement accessibles en javascript. Il est donc possible d'exécuter le code de synchronisation depuis le navigateur de l'utilisateur, qui a donc accès aux plateformes « cloud » ainsi qu'aux serveur locaux (pour autant qu'il en connaisse l'adresse).

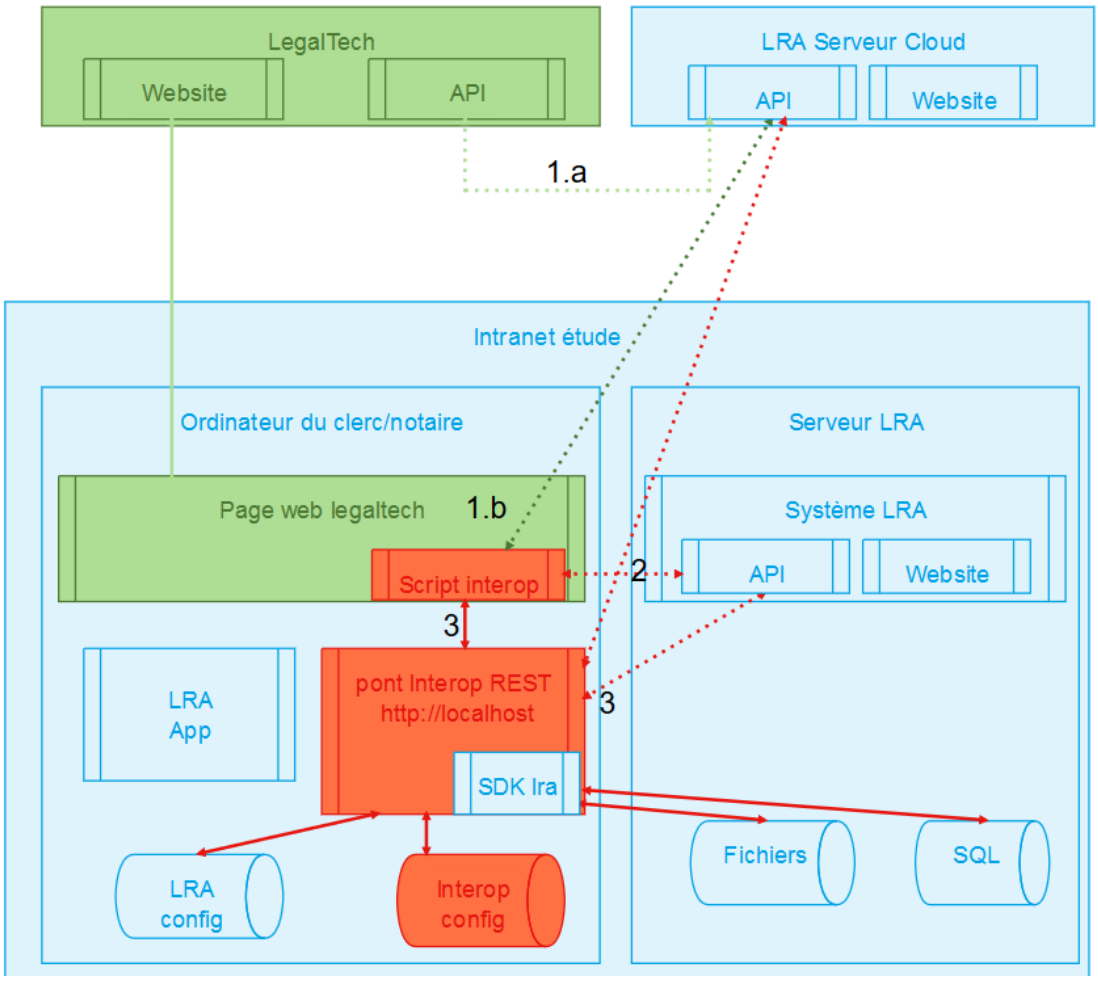
Alternativement, un programme local répondant en web sur l'interface « localhost », pourra faire le « pont » avec les API REST ou le SDK du LRA demandé, qu'il soit local ou réseau.

Le « pont » peut être implémenté par chaque éditeur de LRA, ou peut être un service unique créé avec le concours des éditeurs. Il serait homologué par le CSN et déployable par les legaltech à la demande et passera par les mêmes contraintes d'authentification de l'utilisateur.

Le "pont" est un petit exécutable résident en mémoire, qui s'occupe de récupérer l'autorisation de l'utilisateur, puis de traduire les requêtes de la page web de la legaltech exprimée selon la norme, en requête compatibles avec le LRA en place.

Accès synchrones

Pour les accès synchrones, voici un diagramme évoquant les possibilités :



Depuis la page web :
1.a : l'étude a un LRA cloud : la legaltech accède aux API du serveur Cloud si autorisée
1.b : l'étude a un LRA cloud : le script d'interop interroge directement l'API Cloud si dispo

2. L'étude a un LRA local avec API : le pont retourne l'adresse du serveur local, que le script d'interop interrogera ensuite directement
3 L'étude a un LRA local sans API : le script d'interop communique avec le "pont" installé localement, et propose son installation si le "pont" ne réponds pas. Le pont communique avec l'api non standard, ou directement avec fichiers et données sql, via le SDK fournit par l'éditeur du LRA.

Ceci est un optimum technique (on fonctionne le plus directement possible). Il est possible de passer systématiquement par le "pont" (3.), qui de façon transparente, passera par le canal le plus adapté en fonction du LRA.

ACCÈS ASYNCHRONES

Une simple plateforme hébergeant un zip avec le flux et ses fichiers, associée à une page web capable d'appeler les API REST semble suffisante. L'adresse de la page avec l'id (guid) du flux sera envoyée par un autre vecteur (email), et le flux supprimé au bout de X jours ne nécessiterait pas d'authentification.

L'accès à cette page déclenchera l'intégration via le script présent sur la page web. On reviendra alors sur la mécanique synchrone indiquée ci-dessous.

- Alternativement, le flux peut être envoyée en PJ d'un mail (attention à la taille), et intégré par le plugin Outlook du LRA ou via un webhook pour les legaltech.

Conclusion

Afin de favoriser et permettre l'interopérabilité entre les différents logiciels utilisés par les Notaires et leur collaboratrices-teurs une norme de fichiers a été définie au sein d'un groupe de travail animé par le CSN. C'est un bon point de départ, mais nous pensons que cela ne sera pas suffisant tant les scénarios d'interopérabilités sont divers et plus ou moins complexes.

Dans ce document, nous proposons d'aller plus loin et de permettre une interopérabilité fonctionnelle basée sur un jeu d'API REST normalisées.

Cette initiative, si l'on souhaite l'adhésion de tous les acteurs éditeurs du Notariat, ne peut être le fait d'une seule société. Nous pensons, qu'a l'instar des travaux sur la norme d'Interop des fichiers, ce projet doit être organisé et encadré par le CSN.