

ARCHITECTURE TECHNIQUE

Version : 1.1

Date : 7 Mars 2022

Rédigé par : BELABDOUN / DURAND / FONTENIL / MENUDE / TOURE

Relu par : BELABDOUN / DURAND / FONTENIL / MENUDE / TOURE

Approuvé par : BARDET MAGALI

Signature :

MISES À JOUR

Version	Date	Modifications réalisées
0.1	30/10/21	Création
0.2	15/11/21	Mise à niveau
0.3	11/12/2021	Modification retour de la cliente
1.0	26/01/22	Version rendue pour la revue de projet.
1.1	07/03/22	Mise à niveau (suite à la modif STB)

1. Objet

Le but de ce document est de décrire les solutions techniques conçues pour répondre aux exigences définies dans la spécification technique de besoin. Pour rappel, notre projet est de mettre en place un outil pour authentifier des documents divers à l'aide de la technologie 2d-doc, puis de pouvoir vérifier l'authenticité du document de manière pédagogique. De plus, le besoin de lire et d'authentifier des pass sanitaires a été émis par la cliente.

Après réunion avec la cliente, les exigences les plus importantes, selon elle, sont d'avoir un outil qui a pour but d'expliquer la technologie 2d-doc de façon pédagogique et la lecture de pass sanitaire. De plus, la production de documents sur notre projet devra être assez complète pour permettre la continuité du projet. Autre point, la mise en place d'un serveur web devra donner lieu à un document pédagogique, qui permettra aux étudiants de créer à leur tour un serveur web sécurisé.

Ce document servira donc à présenter l'architecture technique de notre projet. Pour cela, nous allons dans un premier temps présenter la structure statique de l'application, puis nous décrirons les composants et sous-composants qui vont constituer l'interface graphique des différents outils présentés dans la STB.

2. Documents applicables et de référence

Pour la réalisation de ce document, les documents suivants seront utilisés :

- La STB (Spécification Technique de Besoin) a permis de mettre en place des règles pour chaque fonctionnalité du site qui ont apporté de nombreuses réponses à la création de ce document.
- La documentation du standard 2D-Doc est accessible au lien suivant :

<https://ants.gouv.fr/nos-missions/les-solutions-numeriques/2d-doc>

3. Terminologie et sigles utilisés

TSL (Trusted Service List) : Annuaire contenant les autorités de certifications associées à leur(s) certificat(s).

VM (Virtual Machine) : un environnement entièrement virtualisé qui fonctionne sur une machine physique. Elle exécute son propre système d'exploitation (OS) et bénéficie des mêmes équipements qu'une machine physique. Elle nous permettra d'héberger un serveur (serveur de base de données, serveur Apache, ...).

OS (Operating System) : un ensemble de programmes chargé d'établir une relation entre les différentes ressources matérielles, les applications et l'utilisateur.

AC (Autorité de certification) : un tiers de confiance situé à la base de la chaîne de certification électronique. C'est elle qui délivre et gère les certificats numériques utilisés pour sécuriser les échanges dématérialisés et garantir l'identité des émetteurs.

BL (Blacklist) : Liste contenant les DataMatrix considérés comme non valide (ex : DataMatrix frauduleux).

DGC (Digital Green Certificate) : Protocole européen pour la gestion des pass sanitaires sous format QR Code.

Logiciel tiers utilisé :

LucidChart : pour la réalisation du diagramme de cas d'utilisation.

Web Sequence Diagrams : pour la réalisation des diagrammes de séquence pour chaque cas d'utilisation.

4. Configuration requise

- Accès à l'outil :

PC : Windows, Linux, MacOS

Navigateur : Chrome, Firefox, Safari, Opera

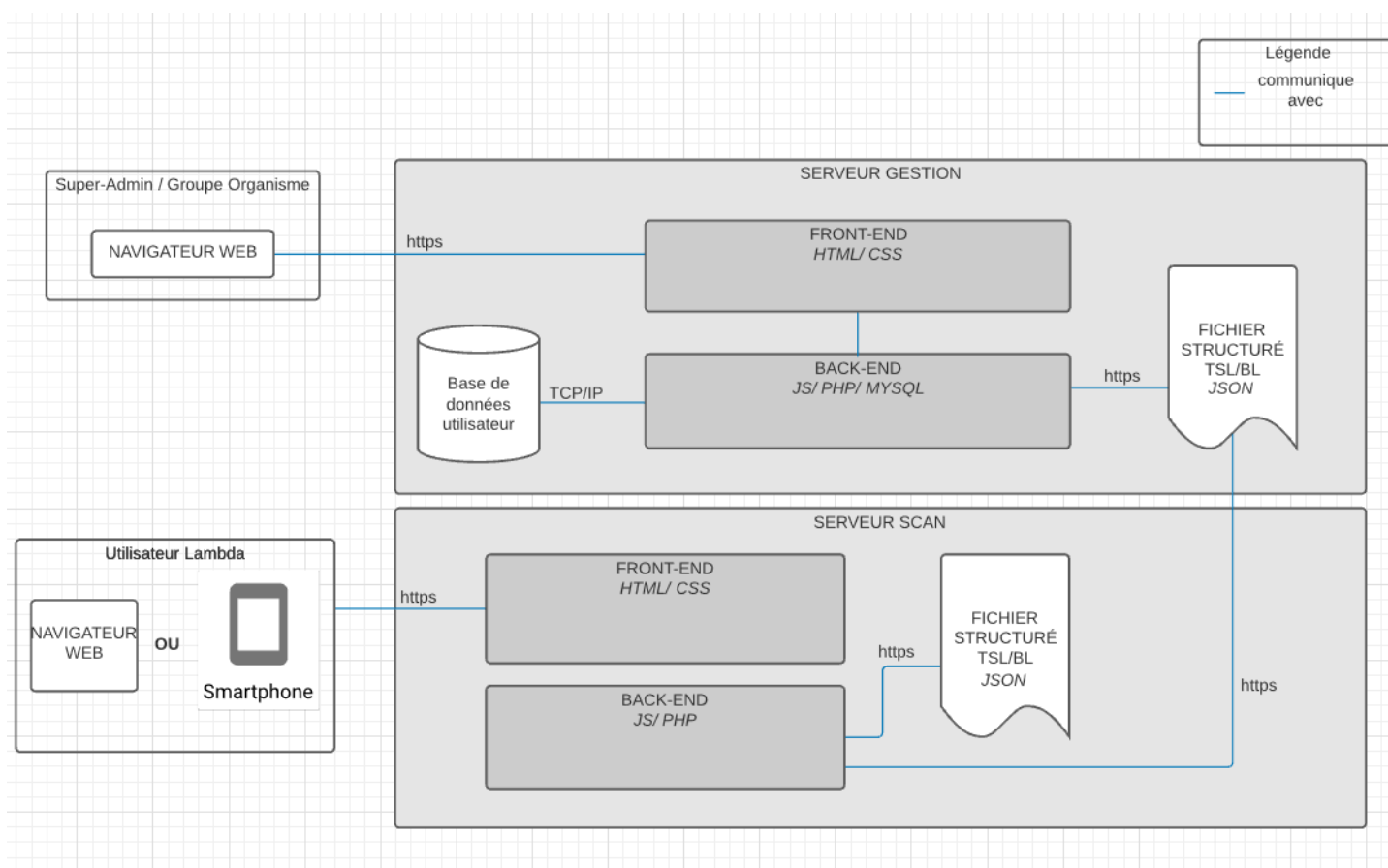
Smartphone : Android 5 ou supérieur

- Hébergement de l'outil :

Notre projet demande 2 Machines Virtuelle avec un serveur Apache sur chacune et un serveur de base de données sur l'une des deux. Ces machines ont été réservées au préalable et sont disponibles pour le développement.

5. Architecture statique

5.1. Structure



5.2. Description des constituants logiciels :

Description du constituant “Serveur Scan” (C0-1)

Le premier serveur est celui qui permet l'accès à notre solution de scan et aussi le site web qui lui est associé. Ce serveur contiendra un fichier structuré (type JSON, XML, ...) défini ci-dessous. Ce dernier devra se mettre à jour de manière automatisée en faisant une requête pour récupérer le contenu dans le serveur de gestion ou lors d'une modification des sources sur le dépôt Git. De plus, il devra être robuste (sécurisé). Ce premier serveur mettra donc à disposition une zone d'hébergement pour l'interface utilisateur et l'outil de scan avec lequel elle communique. Dans les faits, ce composant sera une machine virtuelle avec un serveur Apache d'installé. Nous pourrons communiquer avec cette dernière par connexion SSH pour toute manipulation interne. La mise en place de ce serveur, de manière qu'il remplisse les critères exigés (redéploiement automatique après une mise à jour du dépôt, serveur robuste, ...) demandera alors un effort conséquent lors de la réalisation de ce projet.

Description du constituant “Serveur Gestion” (C0-2)

Le deuxième serveur, quant à lui, permet l'accès à notre outil de création et également à l'interface web qui lui est associé. C'est ce serveur qui contiendra notre base de données MySQL défini ci-dessous. Il contiendra en outre un fichier structuré de même type que celui du serveur de scan et pourra être modifié avec une interface. Il devra lui aussi se mettre à jour de manière automatisée lors d'une modification des sources sur le dépôt Git et être robuste (sécurisé). Ce deuxième serveur mettra à disposition une nouvelle zone d'hébergement pour l'interface réservée aux organismes et les outils de création de DataMatrix et de gestion de nos solutions. Ce composant sera, lui aussi, une machine virtuelle avec un serveur Apache d'installé. On retrouvera aussi un serveur de base de données. Nous pourrons communiquer avec cette dernière par connexion SSH pour toute manipulation interne. La mise en place de ce serveur, de manière qu'il remplisse les critères exigés (redéploiement automatique après une mise à jour du dépôt, serveur robuste, ...) demandera une fois de plus un effort conséquent dans la réalisation du projet.

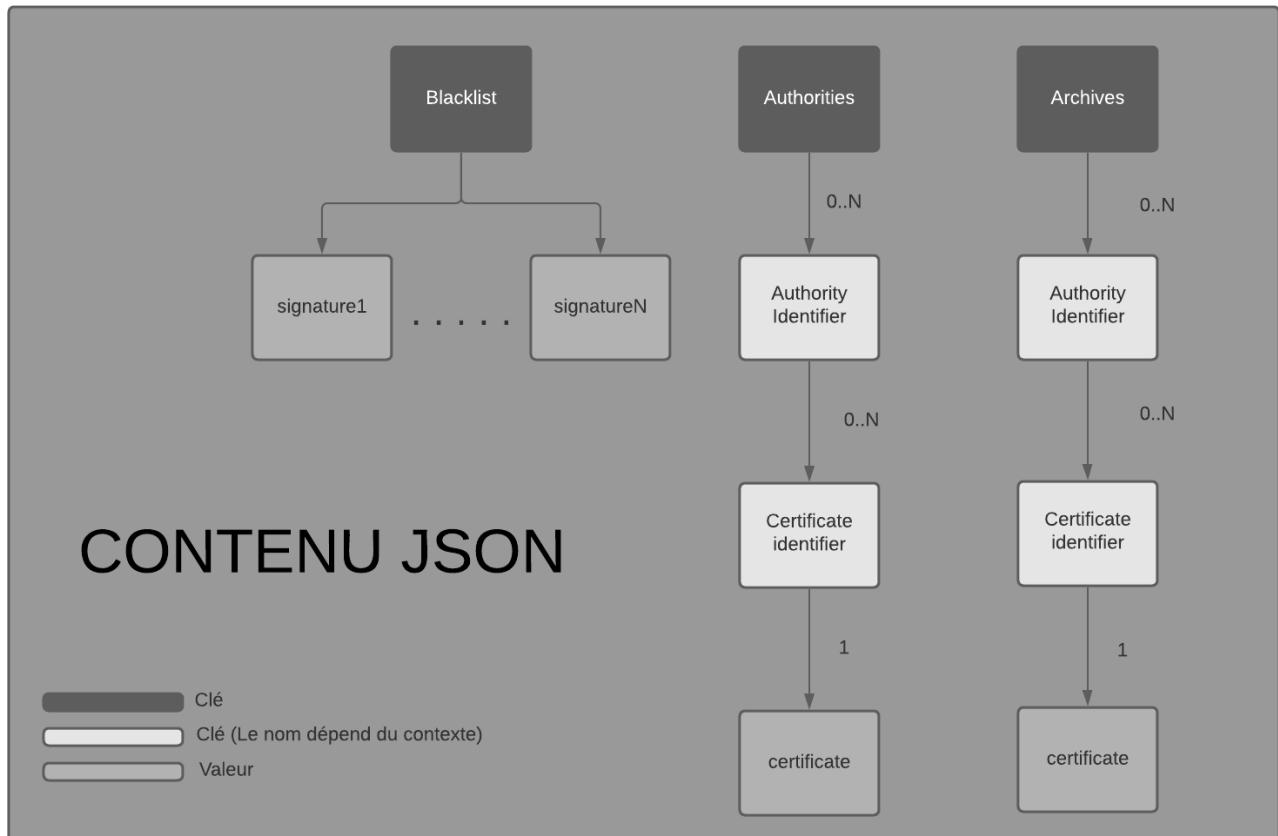
Serveur Scan :

Description du constituant “Fichier structuré” (C1-0)

Le fichier structuré est un élément clé dans le fonctionnement de l'outil de scan. Ce dernier permet d'offrir un moyen de stockage pour la Blacklist ainsi que les autorités de certifications (nom de l'autorité et ses certificats associés). Ce genre de système de stockage (JSON, XML) permet une recherche rapide des informations. Il permettra de gérer *nos autorités de certifications et notre Blacklist*¹. Ce fichier devra être maintenu à jour pour le bon fonctionnement de notre service. Cette mise à jour sera effectuée de manière automatique grâce à l'outil de gestion du serveur 2 (constituant

¹ Les certificats/Blacklists des autres autorités sont accessibles par URL via un navigateur web.

C2-2). Avec cette mise à jour, on autorise l'utilisation de l'application mobile en mode hors-ligne.



Description du constituant “Interface utilisateur de scan” (C1-1)

Ce constituant permet d’offrir une interface web pour l’utilisation de l’outil de scan de DataMatrix. Celui-ci permettra d’afficher la validité d’un DataMatrix utilisant le protocole 2D-Doc et d’un pass sanitaire puis d’afficher son contenu de manière pédagogique. Il y aura également la possibilité d’éditer les données brutes récupérées dans une zone de texte éditable. Cette zone sera affichée au moment du scan d’un DataMatrix. Ce dernier sera régénéré après édition des données. Les langages utilisés seront les langages standard du web (HTML, CSS, JS, PHP). Lors du développement, les sources seront stockées sur notre dépôt Git et développées avec des éditeurs de code comme VS Code ou encore WebStorm.

Description du constituant “Back-End de traitement de DataMatrix” (C1-2)

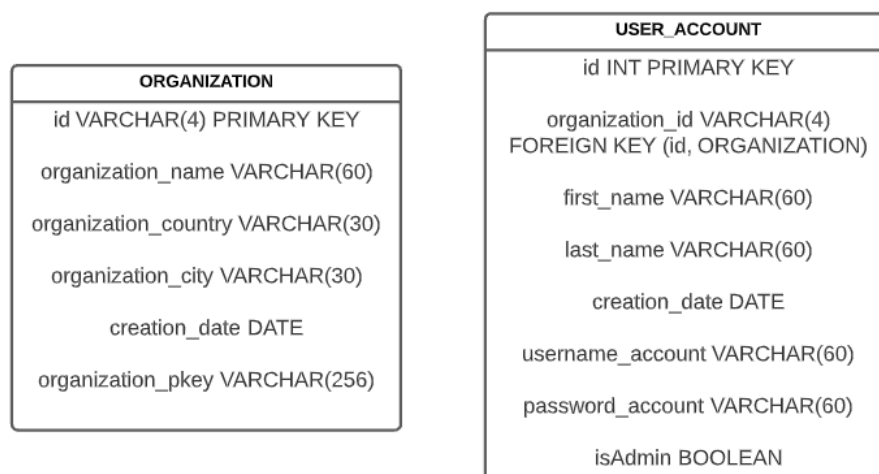
Le Back-End du serveur de scan va permettre d’effectuer le traitement des DataMatrix que l’utilisateur veut vérifier. Ce dernier doit être capable de vérifier la validité d’un DataMatrix (respectant le protocole 2D-Doc) et d’en extraire les données à l’aide du fichier structuré. Un langage serveur

type Python ou Node.JS comportant une bibliothèque² manipulant des DataMatrix sera utilisé pour cette partie du projet. Lors du développement, les sources seront stockées sur notre dépôt Git et développées avec des éditeurs de code comme VS Code ou encore WebStorm. L'importance de ce mécanisme étant un point central de notre projet, un effort important devra être mis en place.

Serveur Gestion :

Description du constituant “Base de Données d’authentification” (C2-0)

La base de données permettra de stocker les différents comptes, celui du super-administrateur, celui des administrateurs organisme et pour finir les différents comptes d'utilisateur. Cette base donnée permet de référencer nos organismes et les différents comptes associés pour pouvoir se connecter et accéder à l'outil de création/gestion. Une base MySQL a été choisie pour ce constituant. On pourra configurer celle-ci avec un outil type PHPMyAdmin.



Description du constituant “Interface utilisateur de gestion” (C2-1)

Ce constituant permet d'offrir une interface web pour l'utilisation de l'outil de création de DataMatrix ainsi que la partie gestion par le Super-Administrateur. Celui-ci permettra aux organismes de certifier des documents en y apposant un DataMatrix contenant les données clefs du document et également de gérer nos services comme la révocation de DataMatrix (ou Certificat). Les langages utilisés seront les langages standard du web (HTML, CSS, JS, PHP). Lors du développement, les sources seront stockées sur notre dépôt Git et développées avec des éditeurs de code comme VS Code ou encore WebStorm.

² Bibliothèque de lecture : [mebjas/html5-qrcode](#)
Bibliothèque pour la création : [datalog/datamatrix-svg](#)

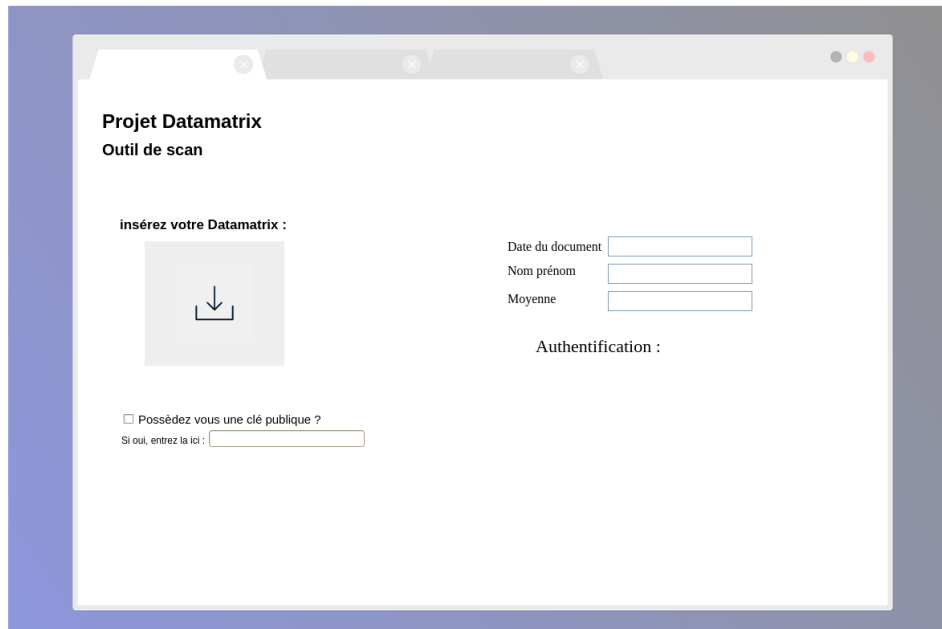
Description du constituant “Back-End de gestion” (C2-2)

Le Back-End du serveur de gestion va permettre la création d'un DataMatrix (protocole 2D-Doc) contenant les données du document et aussi toute la partie gestion. La partie gestion recevra les demandes de modification de la partie Front-End (C2-1) et devra modifier le fichier structuré en conséquence. Un langage serveur type Python ou Node.JS comportant une bibliothèque manipulant des DataMatrix sera utilisé pour cette partie du projet. Lors du développement, les sources seront stockées sur notre dépôt Git et développées avec des éditeurs de code comme VS Code ou encore WebStorm. L'importance de ce mécanisme étant un autre point central de notre projet, un effort important devra être mis en place.

5.3. Composant Front-End du serveur 1 (C1-0)

- Sous-composant “scan_vide” (C1-0-1)

Ce sous-composant permet à l'utilisateur lambda qui y accède, de charger ou scanner son DataMatrix/Pass Sanitaire afin de s'authentifier. L'utilisateur pourrait tenter d'insérer un autre type de code barres qu'un DataMatrix ou un QR Code (un MaxiCode par exemple). On s'assurera donc qu'il s'agit bien d'un DataMatrix (resp. QR Code) et ainsi on vérifiera que le contenu du DataMatrix (resp. QR Code) est conforme au standard 2D-Doc (resp. DGC) et ne contient pas des informations dangereuses comme la chaîne de caractère EICAR (chaîne destinée à tester le bon fonctionnement des logiciels antivirus mais qui peut être utilisée afin de casser le fonctionnement de certains lecteurs de passe sanitaire).



- Sous-composant “scan_rempli” (C1-0-2)

Ce sous-composant permet à l'utilisateur lambda qui a chargé son DataMatrix de voir si son DataMatrix/Pass Sanitaire est valide ou non ainsi que d'avoir accès à une description du fonctionnement de son DataMatrix/Pass Sanitaire. Une fois que le contenu a été validé, on affiche les informations contenues dans ce dernier, tel que la date d'émission du Datamatrix, le nom de la personne et la moyenne. Une attention particulière sera portée sur les entrées de façon à ce que


l'utilisateur ne puisse pas écrire n'importe quoi dans les champs. Pour ce faire, du code JavaScript autorisant seulement un certain type de donnée ou encodant l'information avant de l'afficher sera présent.

Outil Scan

Projet Datamatrix
Outil de scan

insérez votre Datamatrix :

Date du document: 20/11/21 Moyenne: 11.78
Nom prénom: Monsieur Soda

Authentification :  **Datamatrix Valide**

DONNÉES BRUTE CONTENU DANS LE DATAMATRIX
exemple :
DC02FR000001125E125B0126FR247500010MME/SPECIMEN/NATA
CHA 22145 AVENUE DES SPECIMENS 54LDD5F7JD4JEFR6WZ
YVZVB2JZXP2B73SP7WUTN5N44P3GESXW75JZUZD5FM3G4URA
J6IKDSSUB66Y3QWQIEH2G46QOAGWH7YHJWQ

Explication du fonctionnement

	donnée lu	donnée exprimé pour vous	Explication
Date	125E	15 novembre 2012	Date d'émission du document indiquée par le nombre de jours en hexadécimal depuis le 1er janvier 2000. Ici 125E représente 4702, on ajoute donc 4702 jours au 1er janvier 2000 et on obtient le 15 novembre 2012

5.4. Composant Front-End du serveur 2 (C2-1)

- Sous-composant "Authentification" (C2-1-1)

Ce sous-composant permet aux utilisateurs Super-Administrateur, Administrateur-organisme, Utilisateur-organisme d'avoir accès à leurs outils respectif (2DMatrix-Manager et 2DMatrix-Create) en s'identifiant grâce à un identifiant et un mot de passe. Ce sous-composant sera robuste face aux injections SQL qui peuvent être utilisées pour par exemple se connecter à un autre rôle. Celui-ci prendra donc en compte les tentatives de connexions malicieuses.

Authentification

Projet Datamatrix
Outil de scan

Identifiez -vous !

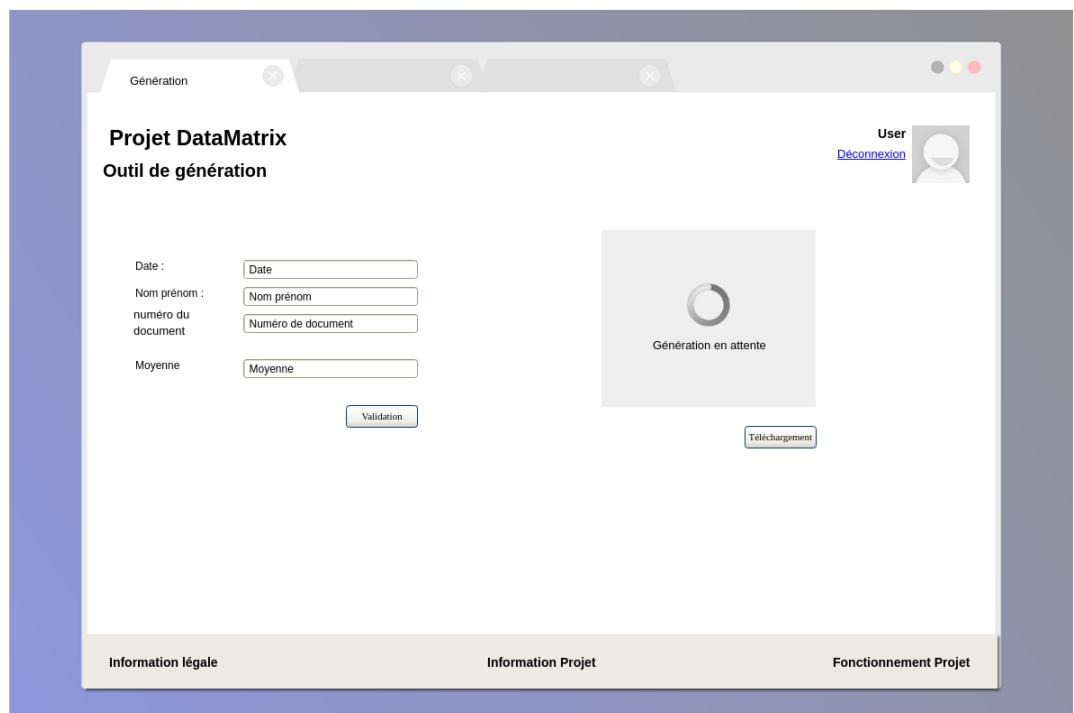
Identifiant

Mot de passe

Connection

- Sous-composant "Génération" (C2-1-2)

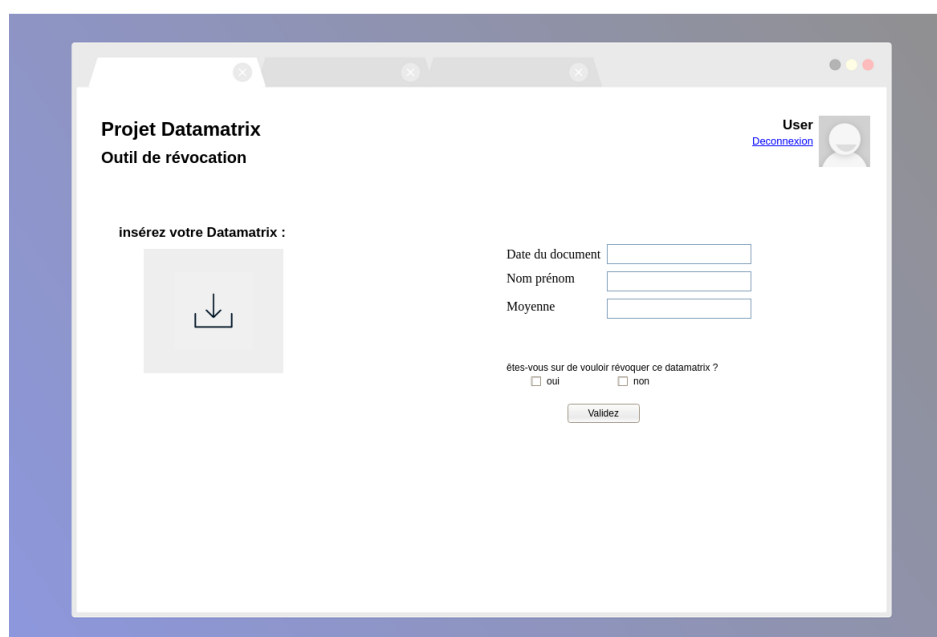
Ce sous-composant permet aux Utilisateurs-organisme de pouvoir créer un DataMatrix afin de valider un document. L'utilisateur devra fournir les informations nécessaires tel que le nom et la moyenne. Même chose qu'avec le sous-composant "scan_rempli", on s'assurera que l'utilisateur-organisme ne rentre pas n'importe quoi dans le DataMatrix, ce qui pourrait compromettre la sécurité.



The screenshot shows a web browser window titled 'Génération'. The main heading is 'Projet DataMatrix' with the subtitle 'Outil de génération'. In the top right corner, there is a user profile icon and the text 'User' with a 'Déconnexion' link. The form contains four input fields: 'Date', 'Nom prénom', 'Numéro de document', and 'Moyenne'. Below these fields is a 'Validation' button. To the right of the form is a large grey box with a circular loading indicator and the text 'Génération en attente'. Below this box is a 'Téléchargement' button. At the bottom of the page, there is a footer with three links: 'Information légale', 'Information Projet', and 'Fonctionnement Projet'.

- Sous-composant "Révocation 2D" (C2-1-3)

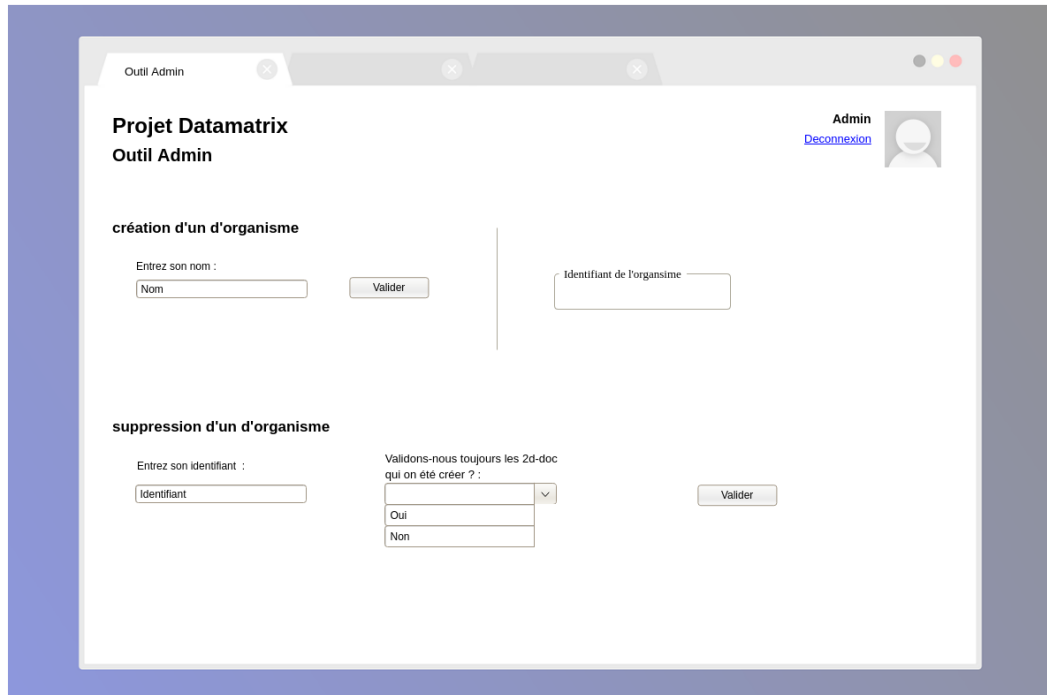
Ce sous-composant permet aux utilisateurs d'un organisme de révoquer un DataMatrix en y insérant ce dernier. Comme les sous-composants précédents, on s'assurera que l'utilisateur-organisme rentre des informations valides et on s'assurera une nouvelle fois que la donnée insérée soit un DataMatrix valide.



The screenshot shows a web browser window with the title 'Projet Datamatrix' and subtitle 'Outil de révocation'. In the top right corner, there is a user profile icon labeled 'User' with a 'Déconnexion' link. The main content area is divided into two sections. On the left, under the heading 'insérez votre Datamatrix :', there is a large square button with a downward arrow icon. On the right, there are three input fields labeled 'Date du document', 'Nom prénom', and 'Moyenne'. Below these fields, a confirmation question is displayed: 'êtes-vous sûr de vouloir révoquer ce datamatrix ?' with two radio button options, 'oui' and 'non'. At the bottom right of the form, there is a 'Validez' button.

- Sous-composant "Gérer_Org" (C2-1-4)

Ce sous-composant permet à un super-administrateur de créer, supprimer ou révoquer un organisme. On vérifiera à chaque fois les entrées pour que la sécurité du serveur ne soit pas compromise.



Outil Admin

Projet Datamatrix
Outil Admin

Admin
[Deconnexion](#)

création d'un d'organisme

Entrez son nom :

Nom Valider

Identifiant de l'organisme

suppression d'un d'organisme

Entrez son identifiant :

Identifiant

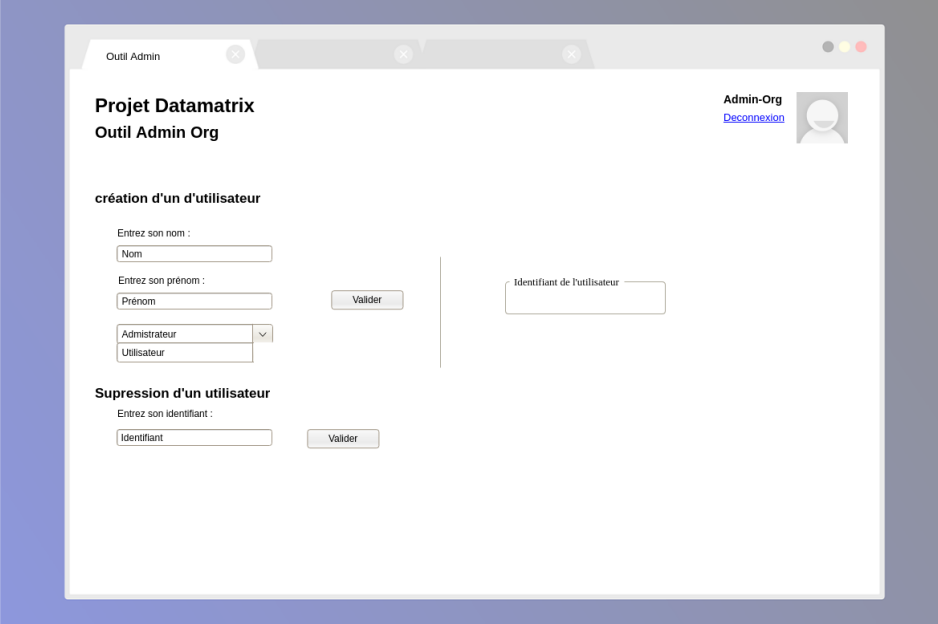
Validons-nous toujours les 2d-doc qui on été créer ? :

Oui
Non

Valider

- Sous-composant "Ajout_userOrg" (C2-1-5)

Ce sous-composant permet à un administrateur d'organisme de créer ou supprimer un utilisateur appartenant à son organisme. Les mêmes étapes que le sous-composant précédent seront mises en place vis-à-vis de la sécurité.



5.5. Justifications techniques

HTML/CSS : Pour notre interface graphique (web), nous utiliserons ces deux langages de balisage qui sont tous les deux des standards pour la création de page web. HTML (HyperText Markup Language) est un langage de balisage nous permettant de créer des pages web et CSS (Cascading Style Sheets) est un langage permettant de faire une mise en forme des pages HTML.

MySQL/PHP : Nous avons besoin d'une base de données pour obtenir les informations, nous utiliserons donc MySQL en tant que système de gestion de base de données relationnelles. MySQL a été choisi, car nous n'aurons pas besoin de base de données utilisant le modèle objet-relationnel, il est alors plus pertinent de choisir MySQL qu'Oracle ou PostgreSQL. Pour PHP, il a été choisi car il s'agit d'un langage standard qui nous permettra de faire la liaison entre notre interface et notre base de données.

MySQL est un système open source de gestion de base de données relationnelles qui permet de sauvegarder et manipuler des données. En ce qui concerne PHP (HyperText PreProcessor), il s'agit d'un langage de programmation conçu pour le développement d'applications web et peut être intégré dans le contenu HTML.

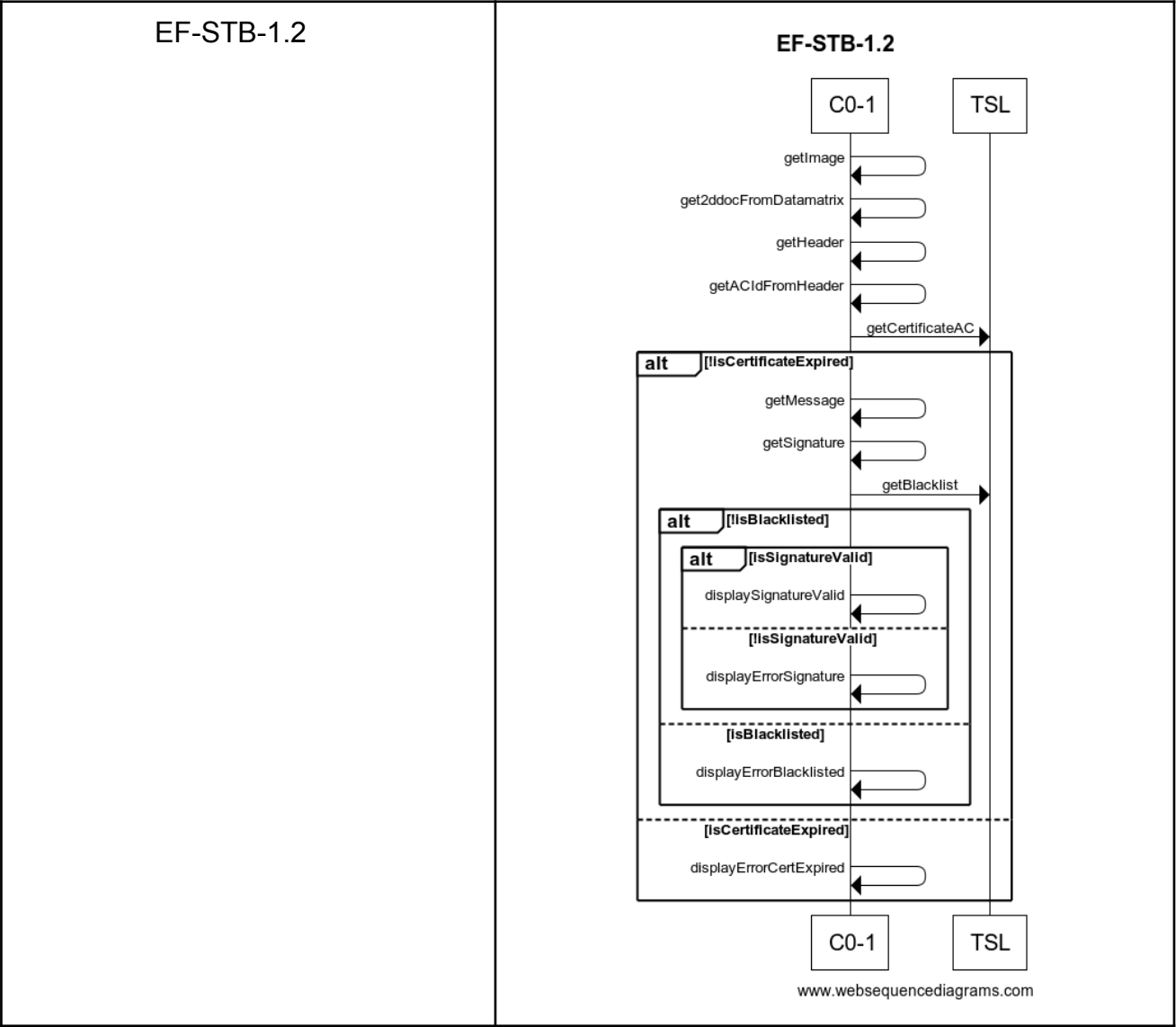
JSON : Nous allons avoir besoin d'une Blacklist, JSON (JavaScript Object Notation) a été choisi pour simplement représenter les informations de manière structurée permettant ainsi de rendre la

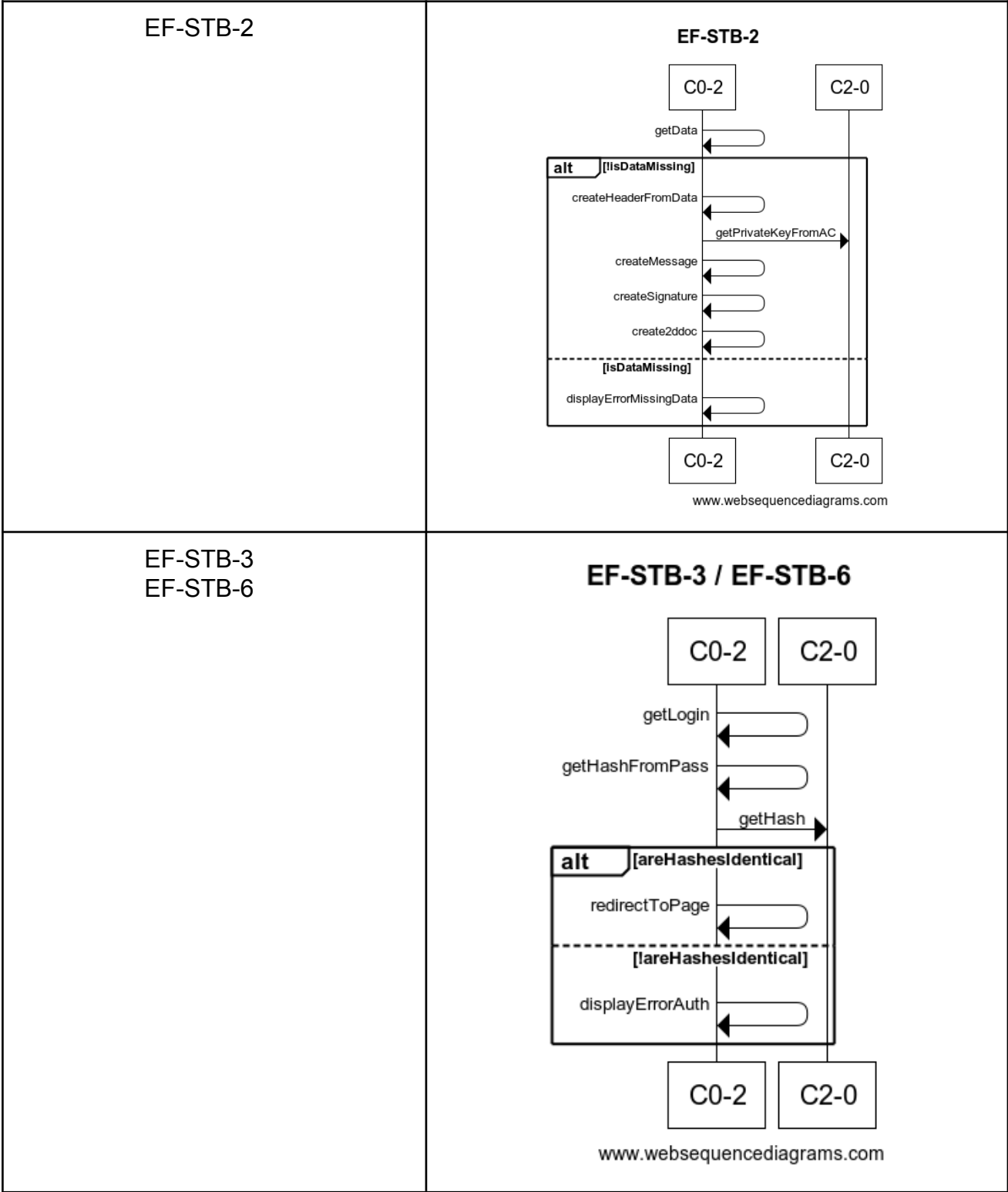
structure avec les données disponible en mode hors connexion. La structure étant la même que la structure d'un objet en Javascript, il sera plus simple de manipuler les données avec un fichier JSON qu'avec un fichier XML.

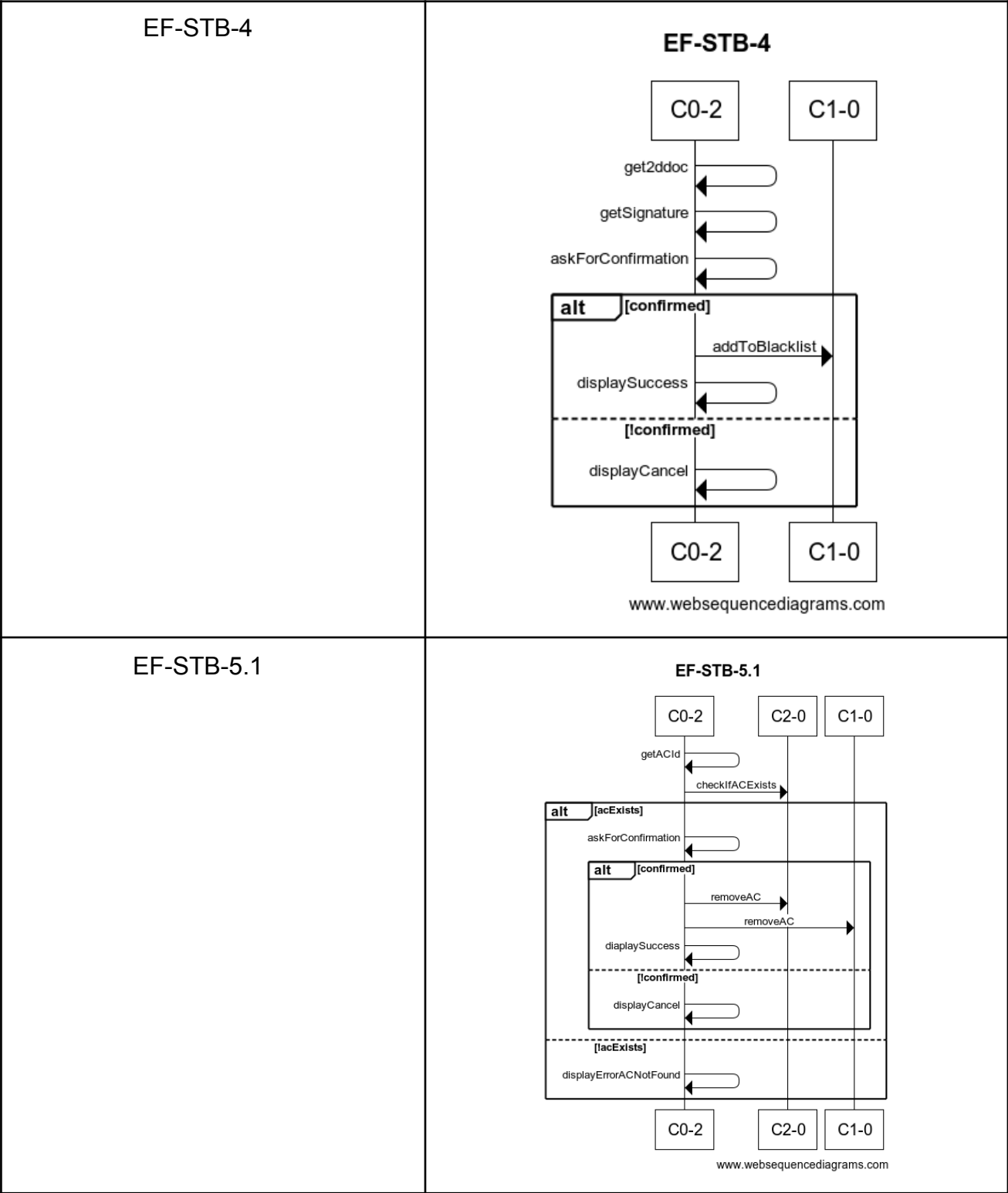
JavaScript : Nous utilisons JavaScript qui est un langage de programmation permettant de créer du contenu dynamique sur une page web. JavaScript sera donc utilisé pour générer et lire le DataMatrix. Il existe différentes bibliothèques en JavaScript qui sont capables de lire des codes-barres comme des DataMatrix ou des QR Codes, nous en utiliserons de ce fait une parmi les disponibles.

Fonctionnement dynamique

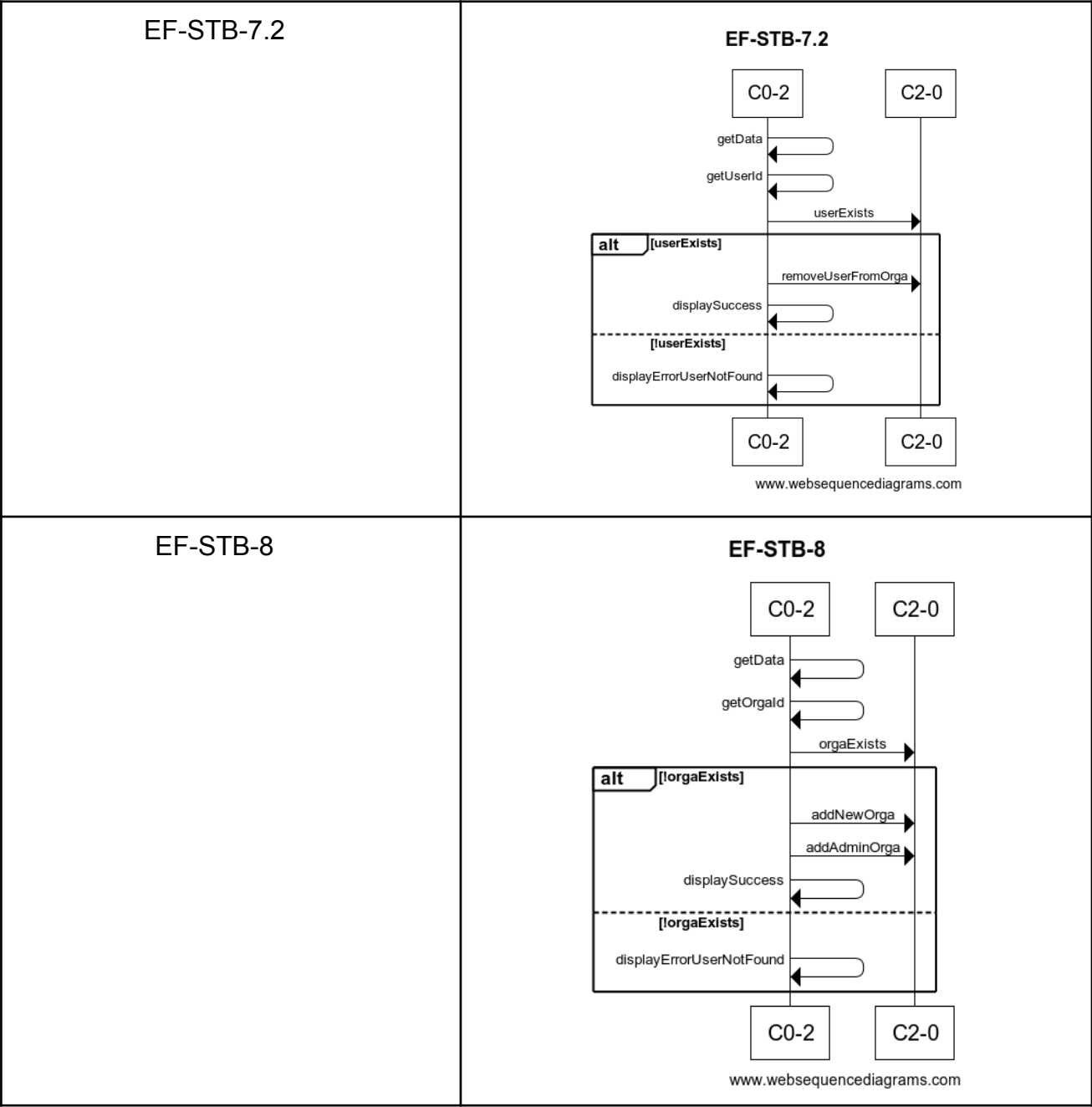
Référence de la fonctionnalité	Diagramme de séquence
EF-STB-1.1	<div> <p>EF-STB-1.1</p> <pre> sequenceDiagram participant C0-1 participant C1-0 C0-1->>C0-1: getImage C0-1->>C0-1: get2ddocFromDatamatrix C0-1->>C0-1: getHeader C0-1->>C0-1: getACIdFromHeader C0-1->>C1-0: getCertificateAC alt [!IsCertificateExpired] C0-1->>C0-1: getMessage C0-1->>C0-1: getSignature C0-1->>C1-0: getBlacklist alt [!IsBlacklisted] alt [!IsSignatureValid] C0-1->>C0-1: displaySignatureValid else [!IsSignatureValid] C0-1->>C0-1: displayErrorSignature end else [!IsBlacklisted] C0-1->>C0-1: displayErrorBlacklisted end else [!IsCertificateExpired] C0-1->>C0-1: displayErrorCertExpired end end </pre> <p>www.websequencediagrams.com</p> </div>

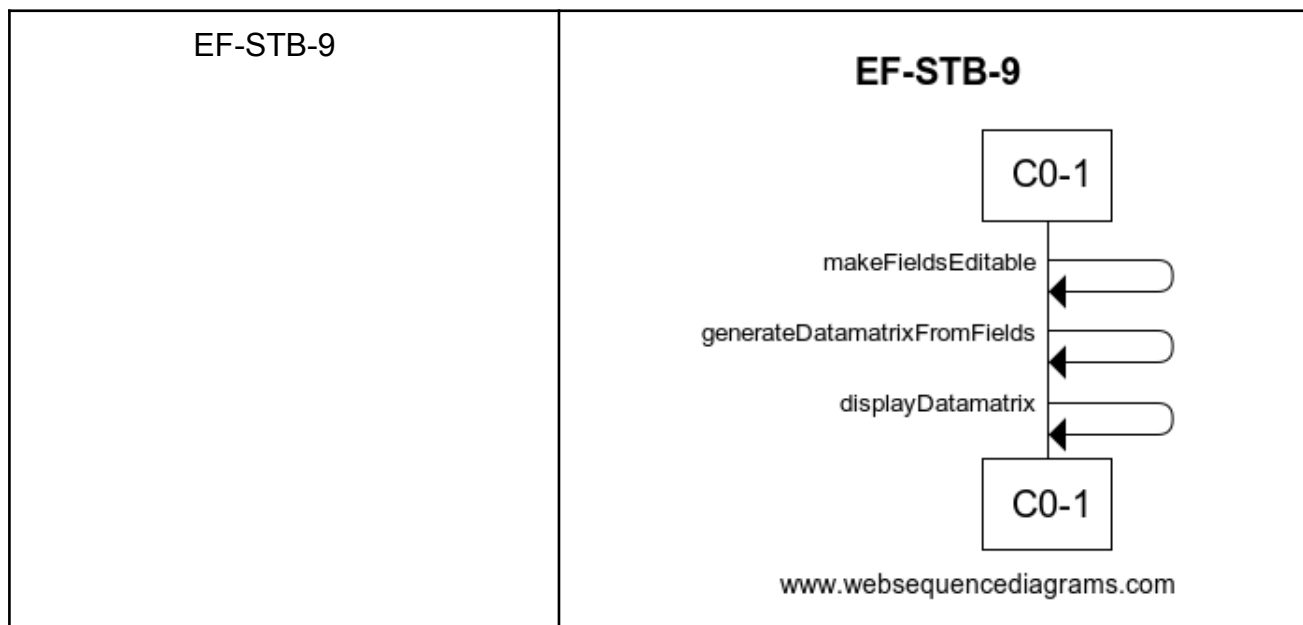






EF-STB-5.2	<div>EF-STB-5.2</div> <pre>sequenceDiagram participant C0-2 participant C2-0 C0-2->>C0-2: getACId C0-2->>C2-0: checkIfACEExists alt [acExists] C0-2->>C0-2: askForConfirmation alt [confirmed] C0-2->>C2-0: removeAC C0-2->>C0-2: displaySuccess else [!confirmed] C0-2->>C0-2: displayCancel end else [!acExists] C0-2->>C0-2: displayErrorACNotFound end</pre> <p>www.websequencediagrams.com</p>
EF-STB-7.1	<div>EF-STB-7.1</div> <pre>sequenceDiagram participant C0-2 participant C2-0 C0-2->>C0-2: getData C0-2->>C0-2: getUser C0-2->>C2-0: userExists alt [!userExists] C0-2->>C0-2: getHashPass C0-2->>C2-0: addUserToOrga C0-2->>C0-2: displaySuccess else [userExists] C0-2->>C0-2: displayErrorUserExists end</pre> <p>www.websequencediagrams.com</p>





6. Traçabilité

Référence de la fonctionnalité	Référence du composant
EF-STB-1.1	C1-0-1 & C1-0-2
EF-STB-1.2	C1-0-1 & C1-0-2
EF-STB-2	C2-1-2
EF-STB-3	C2-1-1
EF-STB-4	C2-1-3
EF-STB-5.1	C2-1-4
EF-STB-5.2	C2-1-4
EF-STB-6	C2-1-1
EF-STB-7.1	C2-1-5
EF-STB-7.2	C2-1-5
EF-STB-8	C2-1-4
EF-STB-9	C1-1

7. Exigences de la spécification technique de besoins

ID Exigence	Actions
EO-STB-01	Le super-administrateur possèdera un certificat lui permettant de se connecter sans mot de passe ni identifiant.
EQ-STB-01	Les données dans le DataMatrix seront chiffrées avec un chiffrement asymétrique pour masquer ses informations. Seul 2DMatrix-Scan sera en mesure de déchiffrer son contenu.
EQ-STB-02	L'utilisateur lambda aura la possibilité de modifier les données brutes fournies par l'outil de scan. Une fois régénéré, il pourra vérifier l'invalidité de ce nouveau Datamatrix (resp. pass sanitaire) de par le fait que la signature n'est pas modifiée.
EQ-STB-03	Lors du scan d'un DataMatrix/QRCode, il sera possible d'afficher ou de masquer les données présentes dans le DataMatrix/QRCode.
EQ-STB-04	Notre base de données contiendra les informations d'authentification des utilisateurs/administrateurs organismes avec le stockage des mots de passe de manière hachée.
EQ-STB-05	Un intérêt sera porté sur les possibilités d'attaques sur nos serveurs. Nous éviterons les failles (Injection SQL, XSS).
EQ-STB-06	La base de données devra être bien configurée en suivant les recommandations de la documentation MySQL.
EQ-STB-07	Les bibliothèques utilisées ne devront émettre aucune requête externe et devront pouvoir fonctionner en local. Notre code ne doit pas contenir d'inclusion de code venant de l'extérieur. (récupération d'un script par HTTP)
ER-STB-01	Un document détaillant la mise en place des serveurs sera réalisé au moment de leur déploiement. Il expliquera également les principaux risques et vulnérabilités d'un serveur web et comment s'en protéger.
ER-STB-02	L'outil de scan devra pouvoir lire les pass sanitaires et indiquer s'ils sont valides.

ER-STB-03	L'utilisateur lambda pourra modifier les règles de validation d'un pass sanitaire pour comprendre de manière ludique qu'un même pass sanitaire peut être ou non valide en fonction de la manière dont les données lues sont interprétées. (nombre de doses requises, date de péremption du pass, ...)
-----------	---