



29 janvier 2021
Bastien ANTOINE - Lucas DUMESTRE

Encadrant: Andrien LEBRE

Projet 3A : Keep Control of Your Data

Rapport de projet



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Table des matières



TABLE DES MATIÈRES

1	Contexte du projet	5
1.1	Intérêt croissant pour la protection des données	6
1.1.1	Intérêt de l'Europe avec le RGPD	6
1.1.2	Les projets Solid et IPFS	7
1.2	Solution fonctionnelle proposée	7
1.2.1	Dissociation du stockage et de l'utilisation	8
1.2.2	Acteur intermédiaire	8
2	Réalisation	8
2.1	Détail de la solution technique	8
2.1.1	Principe Interception-Stockage-Envoi	9
2.2	Développements réalisés	9
2.3	Création et utilisation de balises HTML personnalisées	10
2.3.1	Intérêt de l'utilisation de nouvelles balises	10
2.3.2	Création et mise en place	11
2.3.3	Utilisation des balises dans l'application de démonstration	11
2.4	Fonctionnement de l'extension	12
2.4.1	Principes généraux d'une extension pour navigateur	12
2.4.2	Parcours et affichage des données client distantes	12
2.4.3	Mise en place du principe ISE	14
2.5	Développement d'applications de démonstration	14
2.5.1	Application « à la Instagram »	14
2.5.2	Application d'hébergement de fichiers	15
2.5.3	Intérêt du développement en interne des applications de démonstration	16
3	Limites et problèmes rencontrés	16
3.1	Problèmes lors de l'implémentation des balises HTML	16
3.2	Limites de la solution liées à l'extension	17
3.2.1	Manque de documentation et effet tunnel	17
3.2.2	Utilisation de l'AJAX synchrone déprécié	17
3.2.3	Reconstruction du formulaire	18
3.3	Problèmes potentiels de sécurité	18
3.3.1	Problème de confiance	18
3.3.2	Le chiffrement des données	19
3.4	Problèmes de performances	19
4	Poursuites envisagées	19
4.1	Le cas des vidéos	19
4.2	L'API de l'application de stockage de démonstration	20
4.3	L'accès au grand public	21
4.3.1	<i>InterPlanetary File System</i>	21
4.3.2	Le projet Solid	22
4.3.3	L'inconvénient d'un nouveau protocole de communication	22
4.4	Proposition d'un standard	23
5	Organisation du projet	23
5.1	Un projet de recherche et d'anticipation	23
5.2	Points réguliers avec notre encadrant	23
5.3	L'évolution de la planification à travers les roadmaps	24
5.3.1	Comparaison du planifié et de l'effectif	24
5.3.2	Planification future	25
6	Conclusion	26

1 Contexte du projet

Chaque jour plus de 100 millions de photos et vidéos sont publiés sur Instagram dans le monde¹. Cela représente près de 1200 photos et vidéos mis en ligne à chaque seconde. De même, sur YouTube 720 000 heures de contenu sont ajoutés sur la plateforme chaque jour² de part le monde, ce qui fait plus de 8 heures ajoutées à chaque seconde. Si l'on s'intéresse à Facebook, on apprend qu'il comptait au premier trimestre 2020 2,7 milliards d'utilisateurs qui utilisaient le réseau au moins une fois par mois, soit environ 34% de la population mondiale, et près 1,8 milliards d'utilisateurs quotidiens³.

Ces quelques chiffres sont considérables, et il est même parfois un peu compliqué d'en saisir l'ampleur, mais ils montrent bien l'importance qu'ont pris ces différents réseaux aujourd'hui. Ainsi la moindre décision qu'ils prennent affecte des dizaines, si ce n'est centaines, de millions, voire milliards, de personnes. Par leur base d'utilisateurs immenses ces plateformes constituent aujourd'hui des bases de données d'informations personnelles absolument colossales que leurs utilisateurs leur livrent, consciemment ou non d'ailleurs.

Ces données personnelles, bien que par définition privées, sont à la base du business model de ces réseaux qui les utilisent, la plus part du temps, pour permettre à des annonceurs de diffuser des publicités ciblées aux utilisateurs des plateformes. C'est *en échange* de ces données que les plateformes sont entièrement gratuites pour les utilisateurs. Bien que parfaitement légales, ces pratiques peuvent poser des problèmes d'ordre éthique, légal, ou bien pratique. Nous nous intéresserons qu'au dernier point, les deux premiers n'entrant pas dans le cadre de ce projet.

Comme ces données sont un des éléments de base du business model de ces géants, il est aisément compréhensible qu'ils soient réticents à donner la possibilité à leurs utilisateurs de savoir ce qui est réellement collecté, voire même de leur donner la possibilité de supprimer partiellement ou totalement leurs informations personnelles. Néanmoins ces plateformes ont tout de même mis en place ces systèmes, que ça soit à cause de contraintes légales, par peur de mauvaise publicité, ou bien même par désir de respect de la vie privée de leurs utilisateurs⁴.

On peut par exemple citer le lancement à l'été 2019 de *Off Facebook Activity*, un outil proposé par Facebook qui permet à un utilisateur de visualiser les sites qu'il a consulté et qui ont partagé des informations avec le réseau, et de supprimer ces données. Il y a seulement une nuance : la présentation de cet outil indique clairement qu'il ne s'agit pas de suppression des données, mais plutôt de dissociation : « - *Si je dissocie mon activité de mon compte, sera-t-elle supprimée ? - Les informations que vous dissociez ne seront plus associées à votre compte. Elles pourront toutefois toujours être utilisées sans être associées à un utilisateur particulier* »⁵.

Malgré ces systèmes de contrôles mis à disposition, et qui permettent théoriquement à tout utilisateur d'une plateforme le contrôle de ses données personnelles, on est en droit de se demander ce qu'il en est réellement dans les faits. Comment un utilisateur qui effectue des choix de confidentialité sur Facebook, Twitter ou encore Instagram est assuré que la plateforme les a bien suivis ? Comment accorder une réelle confiance pour le respect de

1. <https://www.blogdumoderateur.com/chiffres-instagram>

2. <https://www.blogdumoderateur.com/chiffres-youtube>

3. <https://www.blogdumoderateur.com/chiffres-facebook>

4. On est tout de même en droit pour ce point de se demander s'il ne s'agit pas uniquement de messages de façades qui ne reflètent pas réellement le fond de la philosophie de l'entreprise.

5. <https://www.facebook.com/off-facebook-activity>

nos choix, quand ceux-ci sont proposés par la même plateforme qui en sera impactée ? L'utilisateur se retrouve finalement dans une situation où il est dépendant d'une plateforme qui est à la fois juge et partie.

1.1 Intérêt croissant pour la protection des données

Depuis quelques temps à travers le monde nous pouvons observer une tendance croissante en ce qui concerne la protection des données des utilisateurs. Des textes sont promulgués au niveau local, national ou international, comme dans le cas de l'Union européenne avec le RGPD.

1.1.1 Intérêt de l'Europe avec le RGPD

À partir de janvier 2012 la Commission européenne propose une refonte globale des règles pour la protection des données personnelles avec pour objectif de « *redonner aux citoyens le contrôle de leurs données personnelles, tout en simplifiant l'environnement réglementaire des entreprises* ».

Ainsi le 14 avril 2016 le *Règlement général sur la protection des données* est adopté, il est applicable à partir du 25 mai 2018. Cette mise à jour du règlement précédent vise à renforcer et unifier la protection des données des citoyens des pays membres. Il responsabilise aussi les acteurs réalisant un traitement sur les données personnelles de leurs utilisateurs, notamment en augmentant les pouvoirs des autorités de contrôle comme la CNIL⁶ en France.

Ce règlement contient plusieurs règles explicites permettant aux citoyens un meilleur contrôle sur leurs données :

- Les services doivent obtenir le consentement « *explicite* » et « *positif* » de leurs visiteurs : ici les utilisateurs doivent avoir plus de contrôle sur leurs données collectées et leurs utilisations, notamment via l'acceptation des cookies d'un site internet. Dans un autre exemple il n'est plus possible dans un formulaire de contact ou d'adhésion que la case « j'accepte de recevoir la newsletter » soit validée par défaut.
- Le droit à l'effacement : chaque citoyen concerné peut obtenir d'une entreprise qui collecte ses données personnelles, l'effacement de celles-ci.
- Le droit à la portabilité des données personnelles : chaque utilisateur peut demander à un service qui collecte ses données personnelles l'extraction de celles-ci.

1.1.1.1. Version californienne

En juin 2018, la législature d'État de la Californie aux États-Unis promulgue la « *California Consumer Privacy Act* ». Ce texte de loi, semblable au RGPD, protège et donne plus de contrôle aux citoyens de l'état sur leurs données personnelles⁷. Entrée en application deux ans plus tard, en janvier 2020, elle a entre temps inspiré neuf autres états américains à mettre en place des réglementations similaires. Des discussions sont en cours pour porter ces réglementations au niveau fédéral, ce qui les rendraient applicables au 330 millions de résidents des États-Unis.

6. Commission Nationale de l'Informatique et des Libertés

7. <https://www.cnet.com/news/ccpa-is-here-californias-privacy-law-gives-you-new-rights/>

1.1.2 Les projets Solid et IPFS

1.1.2.1. *Solid et Inrupt menés par Tim Berners-Lee*

Dans le contexte grandissant de volonté de contrôle des données personnelles, il est quasiment évident que les problématiques que nous avons soulevé ici soient apparues aux yeux de d'autres auparavant. C'est par exemple le cas de Tim Berners-Lee. Dans une lettre ouverte publiée en mars 2017⁸, il indique que la perte de contrôle des données personnelles fait partir des 3 grands problèmes qui empêchent « *le Web [d']atteindre son véritable potentiel* ». En octobre 2018, il crée une startup, Inrupt⁹, afin de servir de plateforme de développement commercial au projet Solid qu'il développe depuis 2015.

Le projet Solid¹⁰ est un projet open source qui vise à promouvoir et mettre en place une nouvelle forme de web décentralisé, où chaque utilisateur aurait le contrôle de ses données personnelles, stockées dans des PODS¹¹, et pourrait ainsi contrôler qui ou quel service aurait accès à quelles données.

1.1.2.2. *InterPlanetary File System*

Le projet IPFS, acronyme de *InterPlanetary File System*, est un protocole de communication pair-à-pair conçu pour la transmission de fichiers. Ainsi, par définition, ce réseau est décentralisé.

La connexion à ce réseau se fait très simplement, via le téléchargement et l'installation sur l'ordinateur d'un logiciel client. Il est également possible d'installer une extension pour navigateurs web qui permet une utilisation plus fluide du réseau IPFS directement dans le navigateur internet.

1.2 Solution fonctionnelle proposée

Ainsi nous avons la confirmation d'un intérêt grandissant porté par différents ensembles de citoyens et de leurs représentations à travers le monde.

Cependant les services tel que les réseaux sociaux restent toujours en possessions des données de leurs utilisateurs tant que ces derniers n'ont pas demandé leur suppression. Les utilisateurs n'ont alors pas de réelle assurance que leurs choix concernant leurs données soient respectés. C'est dans l'objectif de résoudre ce problème que notre solution prend place.

Le projet que nous menons s'inscrit en complément des actions portées par les gouvernements et collectivités. Il permet de rendre le réel contrôle de leurs données aux utilisateurs.

Pour cela notre solution repose sur deux points clés :

- La séparation des acteurs hébergeant les données personnelles de ceux les utilisant,
- La mise en place d'un acteur intermédiaire permettant de faire le lien entre le service hébergeur de données et celui qui les utilise.

8. <https://webfoundation.org/2017/03/web-turns-28-letter/>

9. <https://inrupt.com>

10. <https://solidproject.org>

11. *Personal Online Data Stores*

1.2.1 Dissociation du stockage et de l'utilisation

Cette séparation des services et des ressources de leurs utilisateurs permettrait que les ressources postées sur internet par un utilisateur ne soient jamais transmises aux services en question, mais uniquement mises à disposition des utilisateurs de ces derniers. Ainsi les utilisateurs garderaient la possession réelle de ces ressources et pourraient les rendre indisponibles à n'importe quel moment.

1.2.2 Acteur intermédiaire

Lorsque les utilisateurs se rendent sur leur site internet, un acteur intermédiaire doit réaliser la récupération et l'affichage des ressources qui n'ont pas été stockées et fournies par le service. Il doit aussi avoir pour rôle de permettre le stockage des ressources du client sur un espace personnel.

2 Réalisation

La réalisation d'un prototype de démonstration de notre solution a débuté en octobre 2020. Nous avons commencé les spécifications, et par la suite le développement.

Nous avons commencé par définir, en collaboration avec notre encadrant, un cahier des charges. Ce dernier a pu évoluer au cours du projet. Dès le départ il contenait plusieurs contraintes principales :

- Notre solution devait pouvoir fonctionner sur n'importe quel service distant, que celui-ci soit prévu pour ou non.
- Son utilisation devait être la plus transparente possible pour l'utilisateur, il devait pouvoir garder ses habitudes.

En se basant sur ces conditions, notre cahier des charges et notre solution fonctionnelle attendue, nous avons réalisé le plan de notre solution technique

2.1 Détail de la solution technique

Nous devons permettre aux clients de pouvoir stocker leurs données personnelles à l'extérieur des services qu'ils utilisent. Nous avons identifié trois formes de données : les textes, les images et les vidéos.

Ainsi si nous prenons l'exemple d'un fil d'actualité sur LinkedIn (voir figure 1 page 9), tous les textes postés par les utilisateurs, afin d'accompagner un de leur post ou de répondre à un de leur contact, mais aussi toutes les images postées peuvent être externalisées. Dans le contexte de notre solution, LinkedIn fournirait un document décrivant la structure HTML de la page, mais chaque élément, sauvegardé sur les stockages client, serait identifié pour être récupéré et affiché par le navigateur.

Chaque élément contient un hyperlien permettant alors à l'acteur intermédiaire de récupérer la ressource distante et de l'afficher dans la page.

Dans le cas d'un ajout sur un site, notre acteur intermédiaire intercepte l'envoi afin de le modifier selon le principe d'Interception-Stockage-Envoi (principe ISE).

2. RÉALISATION

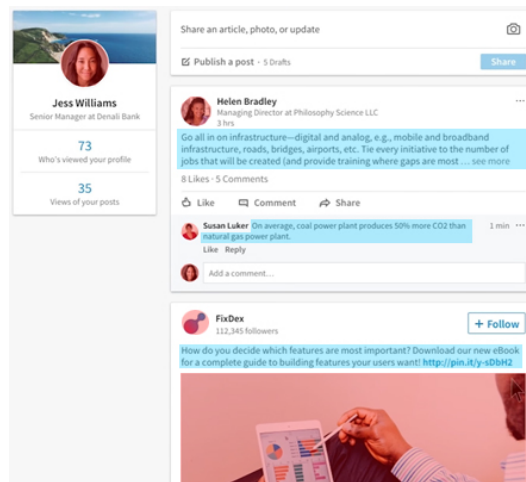


FIGURE 1 – Exemple de ressources sur LinkedIn

2.1.1 Principe Interception-Stockage-Envoi

Ce principe permet d'intercepter et de stocker les ressources que l'utilisateur souhaite envoyer à l'application qu'il utilise. Après interception des données à envoyer, le système mettant en place le principe Interception-Stockage-Envoi (ISE) doit :

1. Récupérer chaque ressource à stocker sur l'espace personnel de l'utilisateur.
2. Pour chacune de ces ressources, l'envoyer au serveur de stockage à travers une requête HTTP simple. Le serveur de stockage la met à disposition et crée un hyperlien qui permet d'y accéder.
3. Récupérer les hyperliens créés et liés aux ressources envoyées.
4. Modifier les ressources interceptées :
 - Dans le cas d'un service conçu pour respecter notre principe, celui-ci s'attend à recevoir des hyperliens, nous pouvons alors remplacer les ressources par les hyperliens générés.
 - Dans le cas d'un service qui n'est pas conçu pour respecter notre principe, celui-ci s'attend à recevoir les ressources demandées. Par exemple dans le cas où il attend une image nous devons lui fournir une image. Ainsi dans ce cas, des ressources factices contenant les hyperliens doivent être générées.

2.2 Développements réalisés

La partie de réalisation du projet repose sur deux parties :

- Le développement d'une extension pour navigateur permettant de réaliser le rôle de l'acteur intermédiaire.
- La conception de plusieurs services permettant de réaliser la démonstration de notre solution et de ses principes :
 - Un service « à la Instagram » émulant les principes de base du réseau
 - Une application de stockage des ressources client indépendante

2. RÉALISATION

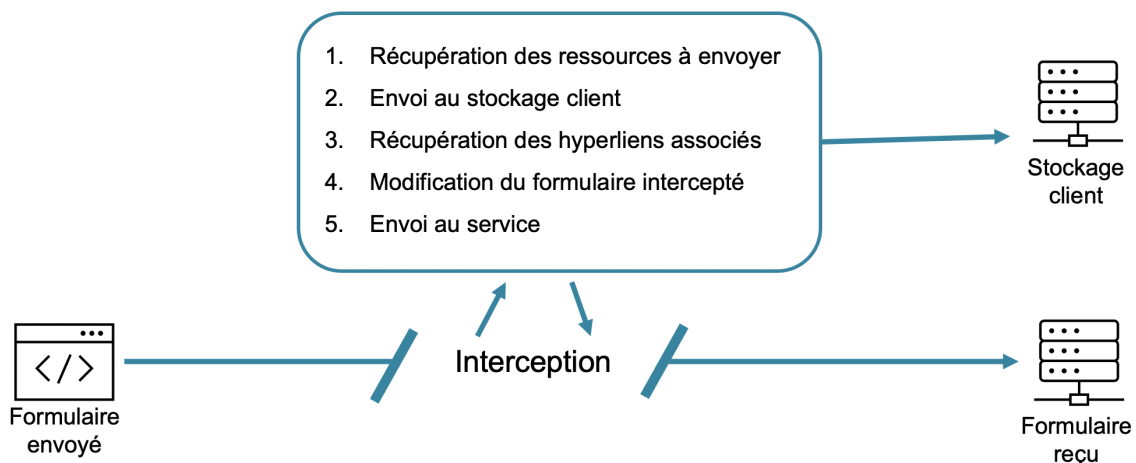


FIGURE 2 – Principe Interception-Stockage-Envoi

2.3 Création et utilisation de balises HTML personnalisées

Afin de pouvoir récupérer les liens des ressources et les afficher, il nous fallait trouver un moyen fiable et connu à la fois de l'application « à la Instagram » et de l'extension pour que l'un sache comment communiquer les liens des ressources à l'autre.

2.3.1 Intérêt de l'utilisation de nouvelles balises

Instinctivement, nous avons pensé à l'inclure dans un attribut `data-src`¹² des balises HTML, et indiquer une classe CSS particulière, `trusted-storage` par exemple, pour que l'extension puisse facilement récupérer toutes les balises HTML qui possèdent cette classe, et du coup récupérer le lien qui leur est associé.

Cependant cette idée soulève un problème. En utilisant une classe CSS en particulier pour identifier, nous limitons les classes à dispositions du service¹³, ce qui fait que, soit le service est au courant de cette limitation, soit il ne le sait pas et dans ce cas risque théoriquement des problèmes techniques s'il vient à utiliser cette même classe CSS pour un autre usage.

Pour éviter ce problème théorique, nous avons décidé de définir de nouvelles balises HTML spécifiquement pour l'usage que nous recherchons. Ainsi nous avons créé les trois balises suivantes :

- `e-txt` : pour des ressources distantes de type texte
- `e-img` : pour des ressources distantes de type image
- `e-vid` : pour des ressources distantes de type vidéo

L'utilisation de trois balises HTML distinctes pour pour chaque type de ressource, provient la manière dont ces dernières sont affichés dans une page web. Ainsi, avec une balise par type de ressource, nous pouvons contrôler séparément leur affichage.

12. On aurait pu tout simplement utiliser l'attribut `src`, comme c'est déjà le cas pour les images et vidéos. Mais justement puisque c'est déjà le cas pour ces ressources, utiliser cet attribut avec ces balises reviendrait à ne pas complètement mettre en place notre système.

13. Dans la pratique ça n'en fait qu'une de moins sur l'infinité à disposition...

Avec ces balises, il suffit au service d'inclure le lien de la ressource distante dans l'attribut `src` de la balise du bon type, et l'extension pourra facilement récupérer et utiliser ce lien pour afficher la ressource.

2.3.2 Création et mise en place

Afin de permettre la création et l'utilisation de balises HTML personnalisées, le WHATWG¹⁴, propose des outils pour permettre à des développeurs de décrire les nouvelles balises qu'ils souhaitent pouvoir utiliser dans leurs documents, ainsi que le comportement qu'elles doivent adopter. La création doit passer par la déclaration de la balise et de son comportement en Javascript, via l'appel à `CustomElementRegistry.define()`¹⁵. Une fois cet appel effectué, il est possible d'utiliser sans problème la nouvelle balise dans la page HTML où elle est déclarée.

L'utilisation d'Angular en tant que framework pour notre front-end a permis de simplifier la création de ces nouvelles balises `e-txt`, `e-img` et `e-vid`. En effet l'API standard du framework fournit une fonction `createCustomElement`¹⁶ qui permet de transformer un composant¹⁷ Angular en classe Javascript décrivant un comportement que l'on peut ensuite utiliser pour définir la nouvelle balise.

Nos nouvelles balises n'ont aucun comportement en particulier, puisqu'elles ne sont là que pour communiquer les liens des ressources distantes à l'extension pour navigateur, il a donc simplement fallu créer trois composants vides, un pour chaque balise, et les utiliser pour pouvoir déclarer correctement nos balises `e-txt`, `e-img` et `e-vid`.

2.3.3 Utilisation des balises dans l'application de démonstration

Dans un premier temps, après la mise en place de ces balises, nous les avons utilisées directement dans notre code ce qui a permis d'avoir un code de l'extension très générique. Cependant les problèmes rencontrés liés à l'extension ont fait apparaître que ces balises ne pouvaient pas être utilisées dans un premier temps. De plus leur utilisation n'était pas compatible avec notre souhait d'une solution qui pourrait s'adapter à tous les sites possibles, même ceux non conçus pour notre principe.

En effet comme décrit auparavant, chaque site qui souhaite pouvoir utiliser ces balises doit au préalable les déclarer. Il est ainsi aisément compréhensible que l'immense majorité des sites ne pourraient pas les utiliser, étant donné qu'elles ne seraient pas déclarées chez eux. De plus, comme nous imposerions notre système à des sites qui ne les connaissent pas, ces derniers ne pourraient pas utiliser les balises.

Ainsi après ces réflexions, nous avons fait le choix de trouver une autre méthode afin de pouvoir stocker et récupérer le lien chez les services utilisateurs des données.

Il est néanmoins toujours prévu de mettre en place ces balises, afin de rendre cette tâche de transmission et récupération des liens plus aisée, dans l'ébauche de standard que nous proposons en poursuite du projet (détaillé dans la section 4.4 page 23).

14. Le *Web Hypertext Application Technology Working Group* s'occupe de l'écriture, entre autre, du standard HTML. Officiellement chargé par le W3C de l'écriture de ce standard depuis fin mai 2019, il est composé en grande partie d'Apple, Google, Mozilla et Microsoft (voir https://en.wikipedia.org/wiki/HTML5#W3C_and_WATWG_conflict).

15. Voir : <https://developer.mozilla.org/en-US/docs/Web/API/CustomElementRegistry/define>.

16. Voir : <https://angular.io/api/elements/createCustomElement>

17. Les composants sont une des briques élémentaire du fonctionnement d'Angular.

2.4 Fonctionnement de l'extension

Dans le cas de notre prototype, l'acteur intermédiaire prend la forme d'une extension pour navigateur. Cette dernière implémente le principe ISE, ainsi que les fonctionnalités de récupération des ressources.

2.4.1 Principes généraux d'une extension pour navigateur

Une extension pour navigateur permet d'interagir avec l'ensemble des pages web chargées par ce dernier. En fonction des droits donnés à l'extension, elle peut utiliser certaines API du navigateur. Le développement d'une extension est semblable au développement d'un script pour un site internet. Les deux se font en JavaScript et peuvent utiliser des bibliothèques externes comme jQuery.

La principale différence est l'existence de deux types de script au sein des extensions : les `background-script` et les `content-script`.

- Les `background-script` sont exécutés dans le background du navigateur une seule fois et n'interagissent pas avec les pages HTML chargées.
- Les `content-script` sont exécutés dans le contexte de chaque page HTML chargée, ils peuvent lire les détails des pages et y effectuer des modifications.

2.4.2 Parcours et affichage des données client distantes

À la réception d'une page HTML, notre script Javascript, défini comme `content-script` de notre extension, parcourt le document reçu à la recherche d'informations de ressources distantes.

Ces informations ont pris plusieurs formes au cours du projet :

- Dans un premier temps elles ont été incluses au sein de nouvelles balises que nous avons défini (voir section 2.3 page 10).
- Dans un second temps, afin de proposer un prototype plus générique (fonctionnel sur tous les sites conçus ou non pour notre standard), ces informations sont incluses dans des ressources factices fournies par l'application visitée. Cette version plus générique est en cours de développement et sera réalisée au cours du mois de février 2020.

Dans le premier cas décrit, à la réception d'un document HTML l'extension parcourt la page à la recherche des balises `e-txt`, `e-img` et `e-vid`. Pour chaque balise récupérée, elle la traite afin de récupérer la ressource stockée par l'utilisateur et puis réalise son affichage :

- Cas d'un `e-txt` : Avec l'hyperlien présent dans la balise, l'extension réalise une requête HTTP afin de récupérer le contenu présent à l'adresse obtenue. Une fois le contenu texte récupéré, celui-ci est injecté dans le document HTML à la place de la balise `e-txt`.
- Cas d'un `e-img` : L'extension récupère simplement le lien présent dans la balise et l'utilise pour la création d'une nouvelle balise `img` qui remplace alors la balise `e-img`.
- Cas d'un `e-vid` : L'extension récupère tous les attributs présents dans la balise et les injecte dans deux nouvelles balises. La première `video` garde les informations de mise en forme définies pour notre élément, la seconde `source`, balise enfant de `video`, contient l'hyperlien de notre ressource distante. Cet ensemble de balises remplace notre balise `e-vid`. Le détail d'une balise `video` peut être trouvé à la figure 4 page 13

2. RÉALISATION

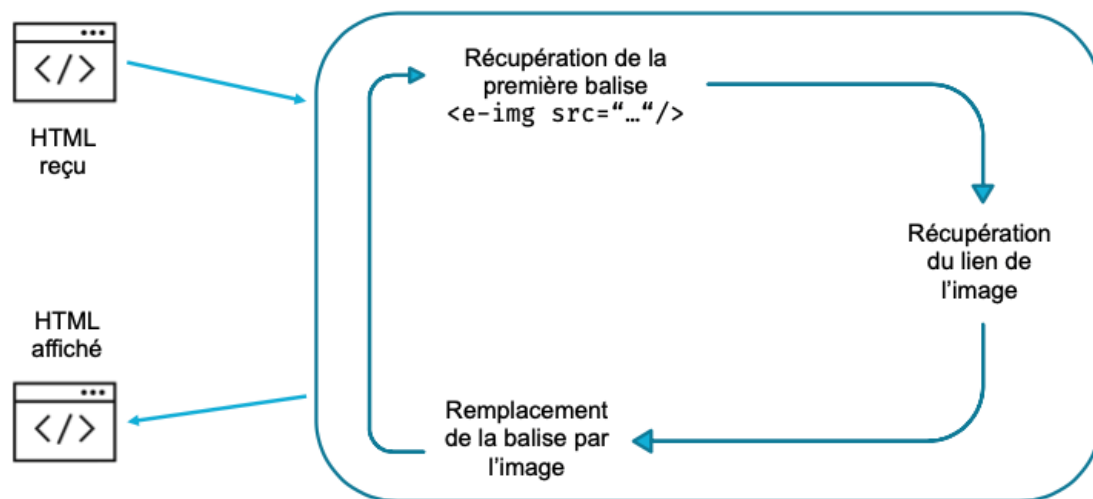


FIGURE 3 – Schéma de fonctionnement de l'extension à la réception d'une page HTML

```
<video>
  <source src="ma_video.mp4" type="video/mp4"/>
  <source src="ma_video.webm" type="video/webm"/>
</video>
```

FIGURE 4 – Détail d'une balise vidéo comprenant plusieurs sources possibles

2.4.2.1. Protection du fonctionnement des sites

Les développeurs mettent en place un graphisme et un dynamisme qui est propre à leur site, ainsi nous devons de conserver la construction de la page lors de l'intégration des ressources utilisateurs.

Pour cela la création de balise personnalisées comme décrite en partie 2.3 page 10 nous a permis de ne pas impacter d'autres attributs tel que les `class`, `id`, `type`, `name`... afin de définir les ressources sauvegardées sur les stockages client. Ainsi lors du remplacement des balises, les attributs non utilisés pour la récupération des ressources sont gardés et réinjectés dans le nouvel élément (voir figure 5 page 13). Cela permet ainsi d'afficher la page HTML comme pensée par les concepteurs du site.

```
<e-img src="image.jpg" alt="This is an image" class="myClass" id="myId"></e-img>
```

(a) Exemple d'une balise `e-img` utilisée avec des des attributs `alt`, `class` et `id`

```

```

(b) Exemple d'une balise `img` utilisée avec des des attributs `alt`, `class` et `id`

FIGURE 5 – Remplacement d'une balise `e-img` par une balise `img` avec conservation des attributs définis

2.4.3 Mise en place du principe ISE

Le fonctionnement du principe ISE au sein de l'extension respecte les règles décrites dans la section 2.1.1 page 9.

Il commence lorsque l'utilisateur complète un formulaire au sein d'une page HTML :

- Préalablement au remplissage du formulaire, lors du chargement de la page web le script place des déclencheurs pour détecter leur envoi.
- Une fois que l'utilisateur a complété et envoie son formulaire, l'extension l'interrompt en l'interceptant. Pour cela elle utilise la méthode `.preventDefault()` permettant de stopper la soumission standard du formulaire.
- Par la suite elle récupère le contenu de chaque ressource à stocker. Actuellement ces ressources ont des `input` identifiés avec la classe `self-storage`, dans notre standard proposé nous souhaiterions que cela soit un attribut à part entière (voir section 4.4 page 23).
 - Dans le cas d'un texte, il n'est pas possible d'envoyer le texte tel quel, puisque le stockage client s'attend à recevoir un fichier, et non du texte. Ainsi après avoir récupéré le contenu sous la forme d'une chaîne de caractères, le script la transforme en fichier en créant un nouvel élément de type `Blob`.
- Ces ressources sous la forme de fichier sont alors envoyées au service de stockage client en utilisant des requêtes AJAX synchrone. Pour chacune des ressources l'hyperlien correspondant renvoyé par le service de stockage est récupéré.
- Les ressources définies par l'utilisateur sont ensuite remplacées par les hyperliens reçus. Le script enlève enfin le déclencheur sur la soumission de formulaire et soumet le formulaire pour réaliser son envoi.

Ainsi nous avons pu développer l'extension en respectant le principe ISE que nous avons défini. L'interception et le stockage dans le stockage client fonctionnent correctement. Cependant nous faisons face à un problème lors de la dernière étape de reconstruction du formulaire, il sera détaillé plus tard dans ce document.

Ce problème nous a conduit à revoir la façon dont nous transmettons les hyperliens au service.

2.5 Développement d'applications de démonstration

Afin de pouvoir tester notre prototype, nous avons développé et mis en place deux applications : une application « à la *Instagram* » utilisée comme équivalent d'un réseau social, et une application utilisée pour l'hébergement des ressources que l'utilisateur partage.

2.5.1 Application « à la *Instagram* »

Cette application reprend le principe de base d'Instagram où un utilisateur peut créer un post, représenté par une image, et d'autres utilisateurs peuvent rajouter des commentaires à des posts. Cette application est conçue autour d'une API tournant sous Python avec Flask, et d'un front-end en Javascript développé avec Angular.

L'API est conçue comme une API RESTful et permet de gérer les posts et commentaires, via 3 endpoints :

- La gestion des posts s'effectue via l'endpoint `/posts` par des opérations classiques CRUD et leurs méthodes HTTP respectives.
- La gestion des commentaires s'effectue via deux endpoints, selon les opérations à effectuer :
 - `/posts/{id}/comments` : permet la création d'un commentaire associé à un post identifié par `id` (cet endpoint n'accepte que des requêtes en POST, puisqu'il ne réalise que la création de commentaires)
 - `/comments/{id}` : permet la mise à jour et la suppression d'un commentaire identifié par son `id`, via des requêtes avec les méthodes HTTP associées

À noter qu'il n'est pas possible de récupérer les informations d'un commentaire tout seul (une requête `GET@/comments/{id}` retourne une erreur HTTP 405 Method not allowed). Par soucis de simplicité, nous ne permettons pas à un utilisateur d'ajouter un commentaire en réponse à un autre commentaire. Ainsi ils sont nécessairement liés à un post, et ne sont donc accessibles qu'en récupérant les informations du post qui leur est associé (via une requête `GET@/posts`).

La documentation détaillée de l'API, incluant les données attendues et leur format, ainsi que les retours, est disponible sur le dépôt GitHub du projet ¹⁸.

Côté front-end, le projet Angular en place permet à un utilisateur de visualiser les différents posts présents sur le service, ainsi que les commentaires qui peuvent être associés. Il est également possible de créer un nouveau post, d'en modifier, ou bien d'en supprimer un, en fournissant un lien vers une ressource le cas échéant. Les mêmes actions sont possibles pour les commentaires.

2.5.2 Application d'hébergement de fichiers

Afin de pouvoir tester l'implémentation du principe ISE mis en place par l'extension (voir 2.4.3 page 14), il nous fallait un espace de stockage en ligne qui exposait une API, afin de pouvoir envoyer et récupérer des ressources de manière automatique, via l'extension, ou bien même directement le navigateur.

Trois possibilités s'offraient à nous : utiliser un stockage en ligne déjà existant proposant une API, mettre en place nous même un service open source similaire, ou bien créer ce service de zéro.

2.5.2.1 Services de stockages déjà existant

Parmi les plateformes qui proposent un service de stockage accessible via une API, les plus importantes sont AWS S3 ¹⁹, par Amazon, et Blob Azure ²⁰, par Microsoft. Ces deux solutions sont très similaires (voire même identiques pour nos besoins, modulo les potentielles différences de structure des API proposés), et proposent du stockage de gros volume à bas coup. Néanmoins, *bas coût* ne veut pas dire gratuit, et les solutions proposées par AWS et Azure ne sont pas réputées pour être open source.

Nous avons donc par la suite exploré les solutions proposées en open source et gratuites, si possible en self hosted. Les deux principales dont notre encadrant nous a parlé

18. <https://github.com/bastantoine/KeepYourDataUnderControl/blob/main/SPECS/API.md>

19. <https://aws.amazon.com/fr/s3/>

20. <https://docs.microsoft.com/fr-fr/azure/storage/blobs/storage-blobs-introduction>

sont IPFS²¹ et OpenStack Swift²². Le premier permet la mise en place d'un réseau décentralisé, pair-à-pair, de partage de fichiers, mais ne semblait pas réellement proposer d'API qui pouvait convenir à nos besoins. Le second repose sur l'écosystème OpenStack, et nécessitait l'installation de composants annexes pour fonctionner correctement. De plus ces deux systèmes sont conçus pour des environnements professionnels, et donc répondent à des besoins de rapidité et de fiabilité bien plus élevés que ce que nous avons besoin pour nos tests.

2.5.2.2. Développement d'un service stockage

Nous avons donc fait le choix de développer notre propre solution. Le service que nous avons développé est une simple API tournant sous Python avec Flask et connectée à une base de données MongoDB.

Cette API permet le stockage de fichiers, quels qu'ils soient, en les envoyant via des requêtes `POST@/`. Lors du stockage, un identifiant unique (UUID) lié au fichier est généré et est envoyé en retour au client. Cet UUID, ainsi que le nom du fichier qui lui est associé, sont stockés dans la base de données.

Lorsqu'un client souhaite accéder à un fichier, il a simplement à effectuer une requête `GET@/{uuid}` qui renvoie le fichier associé.

2.5.3 Intérêt du développement en interne des applications de démo

Les quelques fonctionnalités que nous avons implémenté dans notre application « à la Instagram » ne constituent qu'une toute petite portion de l'ensemble des fonctionnalités disponibles sur le vrai Instagram. On peut citer entre autre l'authentification des utilisateurs et l'utilisation de pseudonymes. Nous n'avons pas jugé utile d'ajouter d'autres fonctionnalités que celles strictement nécessaires pour démontrer notre prototype puisque, ne rentrant pas dans le cadre de notre projet, il n'y avait que peu d'intérêt à passer du temps pour les implémenter. De plus ces fonctionnalités auraient été de potentielles sources supplémentaires de problèmes, compliquant ainsi l'investigation de ceux que nous avons pu rencontrer.

En développant ces applications de démonstration nous même, nous avons le contrôle entier sur chaîne de test et ainsi il était plus aisé d'identifier d'où pouvait provenir les problèmes. Que ça soit dans l'extension pour navigateur ou les applications, il était possible (théoriquement du moins) de les résoudre, puisque nous contrôlions le code source.

3 Limites et problèmes rencontrés

3.1 Problèmes lors de l'implémentation des balises HTML

La création et l'utilisation des balises personnalisées, telles que décrites dans la section 2.3 page 10 a posé quelques soucis, notamment à cause du framework utilisé pour développer le front-end de l'application « à la Instagram ».

En effet Angular est conçu pour être très strict sur le respect des standards web en place. Nous ne pouvions pas simplement utiliser nos balises `e-txt`, `e-img` et `e-vid` dans les

21. <https://ipfs.io>

22. <https://docs.openstack.org/swift/latest/>

pages HTML utilisées par Angular sans les déclarer correctement au préalable. En effet, lors de la compilation des fichiers sources pour utilisation dans un environnement de production, nous obtenions des erreurs qui nous empêchaient d'aller plus loin si leur déclaration était manquante.

Fort heureusement, comme indiqué précédemment, le WHATWG propose des outils afin de résoudre ce problème, outils que nous avons pu utiliser. Néanmoins nous n'avons pas appris de suite l'existence de ces outils, et nous avons dû nous résoudre à utiliser des classes CSS (voir 2.3.1 page 10) pour la démonstration que nous avons faite lors de la première revue de projet, le 19 novembre 2020.

3.2 Limites de la solution liées à l'extension

Le développement de l'extension a posé plusieurs problèmes au cours du projet, la plus part ont été résolus et ne seront donc pas détaillés ici. Cependant dans la majorité des cas, les investigations nécessaires ont été conséquentes.

3.2.1 Manque de documentation et effet tunnel

Un manque de documentation lié à la spécificité des problèmes rencontrés peut expliquer la longueur de ces investigations. En effet une grande partie des obstacles que nous rencontrions était liée à des fonctionnalités spécifiques au sein de Javascript et du fonctionnement des API mise à disposition pour les extensions du navigateur Google Chrome. Qui plus est les réponses trouvées lors de nos recherches étaient souvent données pour des utilisations au sein de script inclut au sein de pages HTML et non de `content-script` d'extension. Les quelques différences sur les scripts d'extension empêchaient alors d'appliquer à notre projet les solutions trouvées.

Nous devons alors tester différentes approches, en analysant régulièrement les retours et les contenus des variables afin de corriger nos problèmes et de rendre fonctionnelles nos nouvelles fonctionnalités. Un inconvénient de cette approche est que nous ne sommes pas sûr que les solutions apportées soient optimales.

De plus ces investigations coûteuses nous ont souvent conduit vers des effets tunnels, où il était difficile de quantifier la charge de résolution des problèmes ni même s'ils étaient résolubles.

3.2.2 Utilisation de l'AJAX synchrone déprécié

Dans le processus de fonctionnement de l'extension, celle-ci doit attendre que l'envoi des fichiers soit terminé et que tous les hyperliens associés soient récupérés avant de modifier le formulaire et de l'envoyer au service. Nous avons alors été confronté à un problème lors de l'envoi de la requête AJAX au service de stockage.

Notre but était de réaliser les envois multiples au service de stockage en parallèle afin de minimiser le temps nécessaire à l'envoi de l'ensemble, puis de passer modifier du formulaire. Cependant si nous effectuions nos envois de manière asynchrone, il nous était impossible d'attendre leur finalisations pour récupérer les liens associés aux ressources, avant de modifier le formulaire.

L'utilisation de l'AJAX en synchrone permet de résoudre ce problème car le script attendra alors la réponse de cette requête avant d'exécuter la suite de notre algorithme.

Cela pose néanmoins deux problèmes :

1. Le temps nécessaire pour l'envoi de toutes les ressources les unes après les autres est bien évidemment plus long que si elles étaient envoyées en parallèle
2. L'utilisation d'AJAX en synchrone est déprécié au sein de jQuery, nous risquons que son utilisation ne soit plus possible dans une future mise à jour de la librairie.

3.2.3 Reconstruction du formulaire

La reconstruction des formulaires au cours de l'implémentation du processus ISE (voir section 2.4.3 page 14) pose un problème principal. Dans le cas d'une ressource image, stockée sur le serveur de stockage client, nous devons remplacer au sein du formulaire son champ de type `file`, avec en valeur l'image, par l'hyperlien retourné par le service de stockage, qui est de type `text`.

Bien que n'ayant pas d'erreur apparente lors de ce remplacement, en modifiant correctement le type d'entrée, et même si la valeur contenue dans le champ du formulaire par la suite était bien notre hyperlien, le service de démonstration « à la Instagram » ne recevait pas l'hyperlien redéfini, mais le nom du fichier que l'utilisateur avait tenté d'envoyer.

N'ayant pas réussi à trouver de solution afin de gérer cet envoi d'hyperlien, nous avons décidé, après concertation avec notre encadrant, de réaliser dans un premier temps la solution plus générique. Cette dernière permet d'imposer notre solution à tout les services en leur envoyant les liens de nos ressources distantes au sein de ressources factices que nous générerons.

3.3 Problèmes potentiels de sécurité

La cybersécurité n'étant pas notre domaine principal de compétences, ces enjeux de sécurité n'ont pas été réellement au cœur de nos réflexions dans le développement de ce projet. Néanmoins nous avons pu identifier quelques points sensibles, notamment après discussion avec un ami ingénieur en cybersécurité.

3.3.1 Problème de confiance

Les premiers points que nous avons relevé concernent la confiance qu'a un utilisateur qui utilise notre système. Comment peut-il être assuré que les données qui sont interceptées par l'extension sont bien envoyées uniquement sur son espace de stockage personnel ? Comment peut-il être assuré qu'elles ne sont pas envoyées autre part ?

Il y a plusieurs arguments qui peuvent répondre à ces interrogations. Tout d'abord nous prévoyons de diffuser le code en open source²³. Ainsi toute personne avec des connaissances techniques suffisante pourra voir et comprendre le fonctionnement de l'extension. De plus ça ne serait pas du tout dans l'esprit de notre projet, alors que nous cherchons à redonner aux internautes le contrôle réel de leur données, de ne pas respecter leur choix et de mettre en place une collecte de leurs données par le biais de notre extension.

23. Techniquement il l'est déjà, dans un dépôt GitHub, simplement que le lien de ce dépôt n'est pas communiqué. D'ailleurs : <https://github.com/bastantoine/KeepYourDataUnderControl>

3.3.2 Le chiffrement des données

En sécurité la multiplication des acteurs intermédiaires conduit souvent à une augmentation des risques et des failles. Dans notre cas, en rajoutant un intermédiaire qui envoie les données utilisateur sur un stockage externe, avant de transmettre un lien au service les ayant demandé, on augmente les risques de fuite de données, notamment par l'interception de celles-ci. Des solutions existent cependant pour pallier à ce problème, notamment l'utilisation du HTTPS qui permet des communications chiffrées.

Il pourrait être possible de mettre en place un filtre dans l'extension qui, par défaut, ne permet pas les envois de ressources à des stockages qui ne mettent pas en place de chiffrement. L'utilisateur aurait tout de même la possibilité de passer outre ce filtre, mais il serait averti des risques potentiels qu'il prend.

3.4 Problèmes de performances

À l'heure actuelle, l'un des grands points du développement logiciel est l'optimisation et la recherche de performances. Bien que cet enjeu soit relativement moins important dans le cas du développement de services web, ce point a tout de même une certaine importance, surtout pour des services comme les réseaux sociaux qui doivent être conçus pour supporter plusieurs millions, si ce n'est plus, d'utilisateurs en même temps.

Des techniques existent et sont mises en place, comme la réplication des services sur différents serveurs et la mise en place de load balancer capables de répartir les requêtes vers les différents services. Il existe des réseaux de diffusion de contenu²⁴ conçus pour mettre la mise à disposition très rapide et très performante de ressources. On peut parler aussi de polices de mise en cache qui peuvent être mises en place, afin de sauvegarder le résultat d'une requête relativement tôt dans une chaîne de communication afin de pouvoir répondre plus rapidement à cette même requête, sans avoir à remonter l'entièreté de la chaîne.

Toutes ces pratiques visant à rendre les services plus performants, et la navigation des internautes plus rapides, seraient moins efficaces, voire inutile pour certains, dans le cas d'une décentralisation des ressources personnelles. Tout le monde n'aurait pas les connaissances, ni la possibilité de mettre en place un système de cache, par exemple, sur le stockage de sa box personnelle. Ainsi les performances des pages web pourraient s'en retrouver affectées.

4 Poursuites envisagées

4.1 Le cas des vidéos

Lors du développement de notre prototype, très peu a été mis en place concernant des ressources de type vidéo, mis à part la création de la balise `e-vid`, et un remplacement par une autre balises HTML. Lors des tests, rien n'a été mis en place pour s'assurer que ça fonctionnait correctement.

24. *Content Delivery Network*, CDN en anglais, voir https://fr.wikipedia.org/wiki/Réseau_de_diffusion_de_contenu

La raison est liée à la complexité de la gestion des vidéos dans des pages web. Ces fichiers peuvent avoir plusieurs formats différents, dépendant du codec²⁵ et du conteneur²⁶ utilisés. Ainsi, selon les navigateurs utilisés et leurs versions, ces différents formats ne pourront pas tous être utilisés²⁷. Pour palier à ce problème, le standard HTML offre la possibilité aux concepteurs de sites de fournir plusieurs versions sous différents formats²⁸. Ainsi le navigateur du client aura *simplement* à choisir la version qu'il sait correctement lire.

Cette possibilité de fournir plusieurs types de vidéos différents rajoute une complexité que nous n'arrivons pas à estimer. Elle a également soulevé des questions auxquelles nous ne sachions pas répondre. Comment gérer le cas où le service utilisateur de la vidéo la convertie en différents formats pour la rendre accessible au plus grand nombre ?

C'est à cause de cette complexité, et parce que nous estimions que tenter de la résoudre aurait pris un temps non négligeable, que nous avons décidé assez tôt dans le projet de laisser ce point de côté. Il s'agit néanmoins d'un point très important qui serait à explorer dans un futur du projet, tant la place des vidéos sur internet est importante.

4.2 L'API de l'application de stockage de démonstration

Lors de la conception et du développement de l'application de stockage utilisée pour la démonstration (voir 2.5.2 page 15), il a fallu faire des choix quand à l'architecture et au fonctionnement de son API. Ces choix ont été faits afin d'avoir quelque chose de très simple, et fonctionnel pour nos tests.

Tout au long de notre projet, nous avons cherché à développer un prototype qui serait le plus générique possible, afin que le strict minimum de modifications, voire aucune, soit à faire pour pouvoir l'utiliser sur un site. Néanmoins certains choix ont influencé des implémentations. On peut notamment citer la conception de l'API de l'application de stockage de fichier qui a influé l'implémentation du principe ISE dans l'extension. Ainsi, comme l'extension est dépendante de la structure de l'API du stockage, tout changement sur cette dernière, voire même un changement de stockage utilisé implique nécessairement des changements dans l'extension.

Lors de nos recherches sur les services de stockage en ligne qui fournissent une API (voir 2.5.2.1 page 15) ; nous avons trouvé AWS S3 et Azure Blob pour le domaine du IaaS²⁹, et OpenStack Swift pour le monde de l'open source. Il paraît probable que, si notre solution venait à être démocratisée, ces acteurs seraient parmi les plus utilisés.

Ainsi, afin de permettre une meilleure intégration de ces différents acteurs, et une transition entre la plus fluide possible pour les utilisateurs, nous avons envisagé deux possibilités qui pourraient être développée :

- La première consiste à intégrer directement dans l'extension les différentes architectures des services de stockages les plus courants. Ainsi, avec un simple paramétrage

25. Un codec vidéo est un algorithme particulier qui compresse une source vidéo afin principalement d'en réduire le poids. Voir cet article très détaillé qui passe en revue les principaux codecs qui existent, leurs caractéristiques principales et leurs différences : https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs

26. Pour une source vidéo, le conteneur est tout simplement son format de fichier. Il dépend notamment du codec utilisé. Par exemple le codec AVC (H.264) pourra être utilisé pour des conteneurs 3GP, MP4 ou WebM, tandis que le codec HEVC (H.265) ne pourra être utilisé qu'avec du MP4.

27. Pour un tableau de compatibilité des codecs selon les navigateurs, voir : https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Containers#index_of_media_container_formats_file_types

28. Voir <https://developer.mozilla.org/fr/docs/Web/HTML/Element/video>

29. Infrastructure as a Service

de la part de l'utilisateur, l'extension saurait immédiatement quel format d'API utiliser

- La seconde consiste à définir un standard auquel répondrait les différents services de stockage et qui définirait une architecture d'API commune à tous. Ainsi tout service de stockage qui voudrait se rendre compatible avec notre système aurait *simplement* à implémenter ce standard.

4.3 L'accès au grand public

Comme indiqué au début de ce rapport, le souci de la protection des données personnelles en ligne est un problème qui concerne tout le monde, et pas uniquement les personnes ayant un certain bagage qui leur permettrait de comprendre les points techniques de fonctionnement.

Ainsi la mise en place d'un service de stockage personnel, l'un des points clés du fonctionnement proposé dans ce projet, doit pouvoir être réalisé par n'importe qui, quelque soit ses connaissances techniques. Nous sommes bien conscient que le code que nous avons produit ne sera pas utilisable par tout le monde, ne serait-ce que par les moyens nécessaires pour utiliser l'application de stockage que nous avons développé³⁰.

Ainsi il serait nécessaire de trouver une solution relativement facile d'accès pour une personne dont le développement logiciel n'est pas le domaine de compétences, et qui pourrait être mise en place aisément. Parmi les solutions qui existent actuellement, on peut citer IPFS et Solid.

4.3.1 *InterPlanetary File System*

Comme indiqué dans la section 1.1.2 page 7, l'IPFS, acronyme de *InterPlanetary File System*, est un protocole de communication pair-à-pair conçu pour la transmission de fichiers. Avec application implémentant ce protocole, lorsqu'un utilisateur se connecte sur le réseau et demande à accéder à une ressource d'un autre membre du réseau, celle-ci est récupérée chez son propriétaire réel, ou du moins sur un hébergement qu'il contrôle. On voit facilement que ce réseau est décentralisé, et répond donc bien à notre souhait de découplage de l'hébergement et de l'utilisation des ressources personnelles.

La connexion à ce réseau se fait via l'utilisation sur l'ordinateur d'un logiciel (voir figure 6 page 22). À partir de ce moment, l'utilisateur est connecté au réseau, et peut partager et accéder aux fichiers présents. Le projet IPFS propose également une extension pour navigateurs web³¹ qui permet d'utiliser de manière plus transparente le réseau IPFS via un navigateur. Entre autre, cette extension permet l'accès à des fichiers via des adresses utilisant le protocole `ipfs://`.

Le projet IPFS propose aussi des outils plus orientés techniques, comme une interface en ligne de commandes, ou IPFS Cluster, à destination des serveurs et permettant de mettre en place des clusters fonctionnant avec IPFS.

30. Avec l'utilisation de conteneurs Docker, le déploiement est relativement simple et peut se faire sur un bon nombre de plateformes, mais néanmoins il est nécessaire d'avoir un certain niveau de connaissances techniques pour savoir comment le faire.

31. Disponible pour Mozilla Firefox et Google Chrome : <https://github.com/ipfs-shipyard/ipfs-companion>

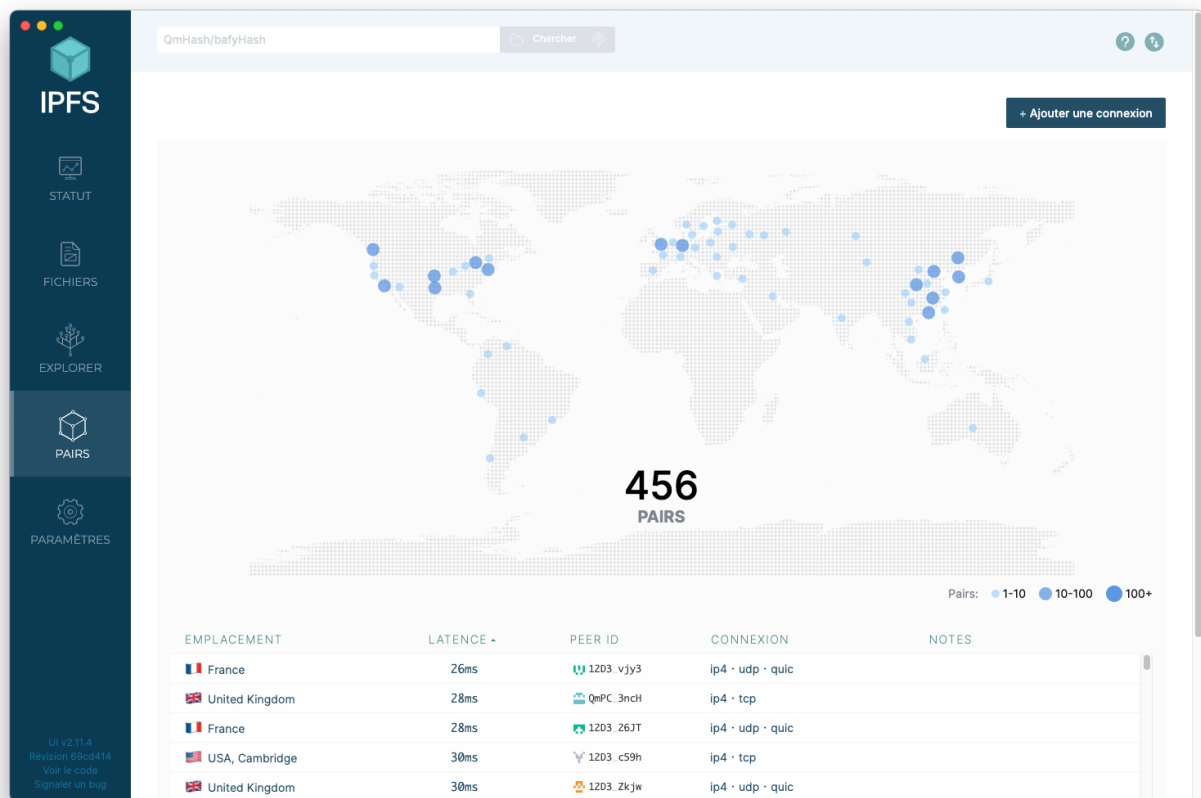


FIGURE 6 – Capture d’écran d’une fenêtre du logiciel pour ordinateur permettant de se connecter au réseau IPFS - La fenêtre montre le nombre de pairs que le logiciel a pu détecter.

4.3.2 Le projet Solid

Comme indiqué au début de ce document, dans la section 1.1.2.1 page 7, le projet Solid vise à mettre en place une nouvelle forme de web décentralisé. Dans cette version du web, chaque utilisateur stocke ses données personnelles dans un espace sécurisé appelé PODS³². Par la suite, il va pouvoir choisir les utilisateurs ou services qui ont accès à ses données. Cet accès se fait via le protocole Solid³³.

À travers la société Inrupt, qui sert de plateforme de développement commercial au projet Inrupt, des outils sont proposés pour accélérer la diffusion du projet. Il y a notamment la possibilité pour des entreprises d’avoir un serveur Solid qui leur permet d’accéder aux données de leurs clients contenus dans leurs PODS personnels. Inrupt propose aussi des outils à destination des développeurs, comme des bibliothèques Javascript pour interagir avec des PODS.

4.3.3 L’inconvénient d’un nouveau protocole de communication

Un des inconvénients pratiques du projet IPFS est la mise en place d’un nouveau protocole. En effet, depuis ses débuts, internet se base sur le protocole HTTP pour fonctionner et

32. *Personal Online Data Stores*

33. <https://solid.github.io/specification/protocol>

permettre la communication de données via des hyperliens. La mise en place d'un nouveau protocole nécessiterait des changements au coeur même des navigateurs. C'est pour cela qu'il est nécessaire, pour le moment du moins, d'installer une extension pour navigateur afin de les rendre compatibles.

Le projet Solid permet de palier à ce problème. En effet, le projet propose un nouveau protocole, mais celui-ci s'appuie sur le protocole HTTP pour communiquer. Ainsi la transition entre un web centralisé, tel qu'il est actuellement, et un web décentralisé est plus aisée.

4.4 Proposition d'un standard

Ce projet reste actuellement limité par l'ampleur de son déploiement possible. En effet un utilisateur intéressé doit installer et utiliser un navigateur particulier ainsi qu'une extension pour celui-ci, et installer un service de stockage compatible.

Ainsi nous souhaitons proposer un standard utilisable par tous :

- Nos nouvelles balises HTML seraient définies dans le standard HTML et seraient ainsi utilisables sur tous les services souhaitant mettre en place cette solution.
- Des services importants seraient basés sur ce standard et s'attendraient ainsi à stocker des liens vers les ressources et non plus les ressources elles mêmes.
- L'acteur intermédiaire implémentant le principe ISE serait directement intégré aux différents navigateurs pour une utilisation plus transparente. Ils incluraient aussi l'algorithme de récupération des ressources distantes définies dans les balises `e-txt`, `e-img` et `e-vid`.

5 Organisation du projet

5.1 Un projet de recherche et d'anticipation

Ce projet ayant une grande part de recherche ou d'anticipation, son organisation n'a pas été facile, en particulier à cause des différentes problématiques listées au niveau du développement de l'extension, détaillées dans la partie 3.2 page 17.

L'orientation de ce projet rendit difficile la quantification de charge et dans les cas de blocages lié à une investigation sur un problème rencontré, l'impossibilité de date de résolution fixe et dans certains cas de savoir si une résolution sera possible.

5.2 Points réguliers avec notre encadrant

Tout au long du projet nous avons pu mener des points de synchronisation avec notre encadrant M. Adrien LEBRE. Ces points nous ont permis d'échanger sur le projet afin d'obtenir un avis extérieur.

Cet avis plus haut niveau nous a permis d'éviter certains effets tunnels en explorant plusieurs voies lorsque nous étions bloqués sur un problème.

5.3 L'évolution de la planification à travers les roadmaps

Avant le mois de novembre 2020 nous n'avions pas de planification complètement détaillée. Le projet était encore en grande voie de spécification, une roadmap était alors difficile à établir.

Suite à la première revue de projet du 19 novembre 2020, nous avons établi une feuille de développement allant de décembre 2020 à mi-janvier 2021.

5.3.1 Comparaison du planifié et de l'effectif pour de décembre 2020 à janvier 2021

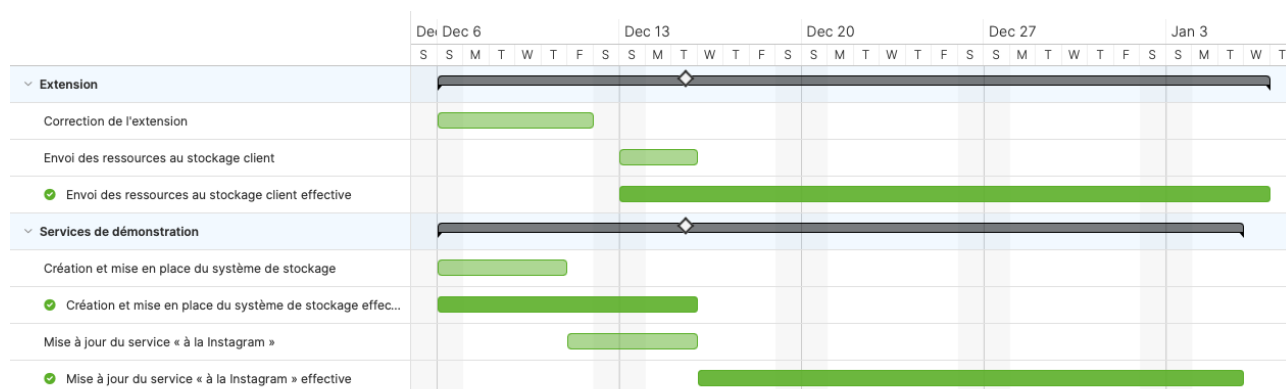


FIGURE 7 – Roadmap prévue de début décembre à mi-janvier - Les tâches ✓ correspondent aux tâches réalisées.

La roadmap contient plusieurs tâches que nous avons ciblées pour réalisation :

➤ Des améliorations sur l'extension :

- Correction des défauts de la version présentée en revue de projet pour qu'elle fonctionne au rechargement d'une page internet :

Ici l'extension ne fonctionnait pas lors du rafraîchissements des pages web. Ainsi les balises HTML `e-txt`, `e-img` et `e-vid` n'étaient pas traitées et les ressources distantes spécifiées n'étaient alors pas récupérées.

Ce point n'empêchant pas l'extension de fonctionner et de réaliser des démonstrations du prototype (il fallait que nous sortions et revenions sur le site de démonstration), il a été retiré des cibles de développement pour nous concentrer sur l'ajout de fonctionnalités dans l'extension.

- Envoi des ressources directement au stockage client par l'extension :

Cet objectif a eu une semaine de retard, modulo la période de fin d'année. Ce retard s'explique principalement par l'investigation qui fut nécessaire afin de rendre le principe ISE fonctionnel.

➤ Des modifications et des développement sur les applications de démonstration :

- Création d'un système de stockage client :

Comme indiqué dans la section 2.5.2 page 15, lorsque nous avons dû mettre en place un service de stockage pour les besoins de démonstration, nous pensions initialement pouvoir utiliser un des services open source que nous avons listé

(IPFS ou OpenStack Swift), mais il s'est avéré que ceux-ci, soit ne convenaient pas, soit étaient trop complexes à mettre en place.

Nous avons donc décidé de développer nous même un service qui pourrait répondre à nos besoins. Bien que simple, il a tout de même fallu se mettre d'accord sur la structure de l'API à mettre en place, afin de ne pas avoir de problème par la suite.

- Modification de l'application « à la *Instagram* » pour revenir à un fonctionnement classique

Pour la présentation de la revue de projet du 19 novembre 2020, notre application « à la *Instagram* » demandait aux visiteurs de rentrer eux mêmes les liens vers leurs ressources. À ce moment, l'extension n'intégrait pas encore le principe ISE. Pour la suite, et pour pouvoir tester l'implémentation de ce principe, il nous fallait modifier l'application de démonstration, pour revenir à un comportement normal, c'est-à-dire où l'utilisateur doit fournir une image et non plus un lien vers cette image.

Le décalage qui s'est produit entre la date prévue et la date effective d'achèvement de cette tâche, est lié à sa facilité. En effet, nous savions dès le départ qu'elle serait très simple à réaliser. Ainsi nous avons préféré la décaler pour nous concentrer sur d'autres tâches.

5.3.2 Planification future

Dans les semaines qui suivent la remise de ce rapport nous continuerons à travailler sur le projet. Ainsi nous avons actuellement une roadmap des tâches à réaliser entre la fin du mois de janvier et le mois de février 2021.

Ces tâches concernent à nouveau les deux parties du développement du prototype :

➤ Sur le service de démonstration :

- Modification du service afin qu'il reçoive des images et non plus des liens. Nos images intégreront alors un lien vers nos ressources distantes stockées sur l'hébergement de l'utilisateur (estimée pour une semaine de développement, achevée fin janvier 2021).

➤ Sur l'extension :

- Recherche sur la manière d'envoi d'un lien à un service qui s'attend à recevoir un autre type de ressource :

Ici nous voulons trouver la manière optimale de création de nos ressources factices afin que ces dernières stockent les liens des ressources distantes. Au niveau des images nous imaginons la création dynamique de QR codes contenant ce lien. Les ressources factices correspondront alors au type demandé par le service visité mais ne contiendront qu'un lien vers nos ressources réelles stockées sur le stockage client.

Nous estimons que cette étape sera achevée fin janvier 2021

- Recherche sur les moyens de récupération du lien par l'extension :

Dans cette étape, nous souhaitons trouver la manière optimale de récupérer les liens stockés au sein des ressources factices.

Cette étape devrait être finalisée au cours de la première semaine de février 2021

- Mise en place de ces solutions à l'intérieur de l'extension appliquant le principe système interception-stockage-envoi :

Dans cette dernière étape, nous mettrons en place ce qui aura été défini dans les deux étapes précédentes afin de réaliser la création et l'envoi de ressources factices de la façon la plus transparente possible pour l'utilisateur.

Ce développement est prévu finalisé pour milieu février 2021.

6 Conclusion

On voit apparaître depuis quelques années un intérêt de plus en plus fort pour meilleur respect de la vie privée sur internet ainsi qu'un contrôle plus fort de l'utilisation des données personnelles. Cet intérêt se traduit par des actions au niveau des états, comme le RGPD au sein de l'Union européenne, ou bien la « *California Consumer Privacy Act* » au niveau de l'état de Californie aux États-Unis. Des actions plus concrètes, visant à changer le modèle existant voient également le jour depuis quelques temps, comme le projet Solid ou le protocole IPFS.

Les projets qui souhaitent développer un nouveau modèle de réseau web sont dépendantes de leur généralisation et de leur acceptation par le grand public, ainsi que par les applications internet utilisatrices des données personnelles. Notre projet se place dans ce cadre, mais vise à imposer à ces services un modèle de dissociation entre l'hébergement des données personnelles et leur utilisation.

Nous avons commencé par définir les principes importants que nous souhaitons mettre en place, en particulier la dissociation hébergement-utilisation, la mise en place d'un acteur intermédiaire et le principe interception-stockage-envoi. Ensuite nous avons réalisé un prototype se composant d'une extension pour navigateur web chargée de réaliser le rôle d'intermédiaire, et d'applications utilisées pour démontrer nos principes.

Ces développements ont fait apparaître des limitations liées aux principes que nous avons mis en place. Il peut s'agir de problèmes techniques liés à l'implémentation, par exemple l'utilisation de requêtes AJAX synchrones, ou comme de problèmes plus généraux, notamment des problèmes potentiels de sécurité ou de performances.

La partie théorique de ce projet couvrant un large champ, nous avons dû effectuer des suppositions lors de notre développement, afin d'obtenir un prototype fonctionnel démontrant les principes généraux proposés. Ainsi les choix que nous avons pris pour acquies lors de notre développement seraient à explorer plus en détail, dans le futur du projet.

Ce projet se plaçant dans un cadre scolaire, nous consacrons aussi une partie de ce rapport sur l'organisation du travail, notamment avec une comparaison entre la roadmap planifiée de décembre 2020 à janvier 2021 et celle effective, ainsi qu'une explication des différences.



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

IMT Atlantique Bretagne - Pays de la Loire - www.imt-atlantique.fr

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 03
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes
4, rue Alfred Kastler - La Chan-
trerie
CS 20722
44307 Nantes Cedex 03
T +33 (0)2 51 85 81 00
F +33 (0)2 51 85 81 99

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné
Cedex
T +33 (0)2 99 12 70 00
F +33 (0)2 99 12 70 08