SCIENCE

Gestion sécurisée de données personnelles

Nicolas Anciaux [1] et Benjamin Nguyen [2]

« *Make yourselves sheep, and the wolves will eat you.* »
BENJAMIN FRANKLIN

1. Introduction

En quelques années, nous sommes entrés dans une ère de génération massive des données personnelles créées par les individus, leurs équipements digitaux (smartphones, équipements d'auto-mesure, compteurs électriques intelligents) ou mises à disposition par les organisations (banques, administrations, centres médicaux, etc.). L'ensemble de ces données constitue la vie numérique (souvent privée) de l'individu, décrivant ses déplacements, sa consommation, ses relations, son état médical, social, financier, ses comportements, ses préoccupations, etc.

Ces données constituent une manne pour l'économie. La valeur boursière des entreprises dont le modèle d'affaire est basé sur l'exploitation des données personnelles en témoigne. Deux milliards de dollars par an sont dépensés aux États-Unis dans l'achat d'informations personnelles [17]. Le Forum économique mondial assimile les données personnelles à un « nouveau pétrole » [31] et certaines initiatives politiques les assimilent à une ressource et envisagent des manières de les taxer pour qu'elles n'échappent pas à la TVA [12].

Afin de pouvoir bénéficier de leurs propres données, les individus utilisent des applications en ligne qui rendent ces données exploitables et accessibles. Le respect de la vie privée des personnes est alors délégué à l'application, de manière

1. Inria Paris-Rocquencourt.
2. INSA Centre Val de Loire.

contractuelle, mais sans garantie tangible pour l'individu. Les données personnelles sont ainsi parfois exploitées de façon très peu transparentes à des fins secondaires, pour répondre aux exigences des modèles d'affaire ayant cours. Des passe-droits peuvent être accordés pour satisfaire les requêtes d'un gouvernement ou d'un partenaire industriel. Et tous ces usages peuvent être menés de façon unilatérale par le gestionnaire des données, sans l'assentiment de l'individu concerné, ou suivant des chartes de confidentialité parfois floues, changeantes, et présumées acceptées par celui-ci. Les systèmes manquent souvent aussi d'ouverture pour l'usager vis-à-vis de ses propres données, et tout désengagement est souvent difficile. Lorsqu'il est pratiqué, c'est souvent au risque de perdre ses données. Par ailleurs, la centralisation des données personnelles conduit à des divulgations accidentelles et à des attaques informatiques répétées impactant de très grands volumes de données. Ainsi, l'utilisateur, dépossédé de tout moyen de contrôle, ne peut ni éviter, ni même souvent connaître les usages indésirables qui pourraient être faits de ses propres données.

La situation actuelle est donc très contestable du point de vue du respect de la vie privée et cela commence à avoir un impact sur l'économie. La révélation de l'observation du Cloud par la NSA pourrait conduire à des pertes économiques pour les acteurs du Cloud américain estimées entre 22 et 180 milliards de dollars selon les analystes [10, 27]. IBM implante des serveurs hors des États-Unis [21], pour satisfaire ses clients pour qui la localisation géographique (et les passe-droits associés) prend de l'importance. Le Forum économique mondial lui-même plaide pour un contrôle accru des usagers sur leurs données [32].

Un consensus économique, politique et social émerge actuellement pour parvenir à un modèle plus respectueux de la vie privée. Les grands acteurs économiques travaillent à cet objectif, comme en attestent les travaux du groupe « *Trustworthy Computing* » de Microsoft [3] visant à améliorer la confiance dans les serveurs. De plus, certains industriels sont réticents à une exploitation généralisée des données personnelles qu'ils ont en charge dans le modèle actuel du Web et soutiennent les approches visant à tirer parti de l'explosion actuelle de l'utilisation des données personnelles digitales, tout en préservant l'intimité des usagers. Par exemple, EDF qui se voit comme un tiers de confiance pour les données de consommation électrique, intègre actuellement des technologies de protection de la vie privée dans ses compteurs électriques intelligents. Nous assistons aussi à une prise de conscience du monde politique en Europe qui s'exprime notamment par le biais du renforcement [4] de la Directive 95/46/CE [13] pour devenir un règlement [26], édictant ainsi les principes légaux de respect de la vie privée numérique dans l'Union. Une décision de la

3. Voir `http://www.microsoft.com/en-us/twc/default.aspx`

4. CNIL. Règlement européen et surveillance des citoyens : avancées au Parlement européen. `http://www.cnil.fr/linstitution/actualite/article/article/reglement-europeen-et-surveillance-des-citoyens-avancees-au-parlement-europeen/`

Cours européenne de mai 2014 [5], dite « affaire Google Spain », contraignant Google à offrir aux usagers un outil de droit à l'oubli va aussi clairement dans le sens d'un durcissement de la législation ; peut-être même trop, car cette décision conduit à de nombreuses autres questions, comme en particulier la légitimité qu'aurait une entreprise de décider de ce qu'elle déréférence ou pas, ou encore le droit à l'information et au principe d'archivage. Enfin, les représentants de la société civile et de nombreux usagers restent attachés aux principes de respect de la vie privée numérique. Contrairement à l'idée reçue, les plus jeunes ne sont pas moins préoccupés que leurs aînés par le respect de leur intimité [8] et sont plus nombreux aujourd'hui à modifier les paramètres de confidentialité de leur smartphone [6] et à bloquer des applications perçues comme trop invasives [22]. De plus, ils se tournent vers des moyens de communication plus éphémères comme *Snapchat* dont le principe est d'envoyer des photos qui disparaissent quelques secondes après avoir été visualisées. Notons au passage le fait que *Snapchat* a été attaqué en justice car il était avéré que les données n'étaient pas véritablement effacées [7]...

Ainsi, les données personnelles doivent être manipulées sous un contrôle accru des individus de manière à rétablir la confiance nécessaire. Il s'agit donc de garantir les principes fondamentaux du respect de la vie privée : consentement des individus pour la finalité du traitement, collecte et rétention de données limitées, exposition minimale des données à des tiers, droit d'ouverture sur les données et audit des usages.

Malheureusement, il n'y a pas encore de solution technique satisfaisante [8]. Les deux approches actuelles consistent soit à améliorer la confiance que l'on peut mettre dans les serveurs, soit à introduire des serveurs personnels chargés de la gestion des données de leur propriétaire. La première approche ne permet pas de résoudre les problèmes intrinsèques aux approches centralisées : attaques sophistiquées qui, si elles réussissent, compromettent de grands volumes de données et, surtout, le modèle centralisé est basé sur la délégation et la confiance dans l'opérateur gérant les données. Les approches décentralisées, si elles ne font plus une hypothèse de confiance, doivent sacrifier certaines fonctionnalités et usages innovants sans toutefois pouvoir garantir une grande sécurité en contrepartie.

---

5. Google Spain SL et Google Inc. contre Agencia Española de Protección de Datos (AEPD) et Mario Costeja González, voir `http://curia.europa.eu/juris/liste.jsf?num=C-131/12`

6. Byron Acohido. *Snowden effect : Young people now care about privacy.* USA Today, 18 Nov. 2013.

7. Elizabeth Dwoskin and Brent Kendall. *Snapchat Settles FTC Charges*. The Wall Street Journal, 8 May 2014.

8. Nous parlons ici de solutions techniques. Naturellement, le droit a également sa place, par la confiance qu'il peut instaurer entre un utilisateur (client) d'un système, et l'infrastructure (fournisseur de services). Toutefois, nous estimons qu'il n'est pas satisfaisant de faire reposer l'intégralité de la confiance simplement sur le droit, et ainsi dans tous nos travaux nous considérons un attaquant de type semi-honnête (dit aussi honnête mais curieux), qui observe volontiers toutes les informations auxquelles il peut avoir accès.

Les travaux entrepris dans l'équipe Inria SMIS (*Secured and Mobile Information Systems*) préfigurent une nouvelle approche de gestion des données personnelles que nous appelons le « Web Personnel Sécurisé », où les individus régulent leurs données personnelles depuis des composants personnels sécurisés, permettant de réaliser des serveurs personnels de données *sécurisés* [1]. Les fonctionnalités principales des solutions centralisées sont préservées : durabilité, disponibilité, partage des données. Mais l'exploitation des données se fait avec l'assentiment du propriétaire, qui régule les usages au travers des autorisations qu'il donne, et dispose de fortes garanties de non contournement de ses directives. Outre les problèmes techniques, le thème de recherche du Web Personnel Sécurisé touche de nombreux enjeux économiques, juridiques, ou sociologiques. Sans prétendre étudier ces enjeux, nous essayons d'y confronter nos solutions techniques. Cela passe par une stratégie de validation de nos propositions, des démonstrations aux industriels, des discussions avec des chercheurs d'autres disciplines (droit, économie, etc.) et des essais sur le terrain impliquant des usagers.

Dans cet article, nous présentons certains travaux de l'équipe SMIS les plus en lien avec le nouveau modèle du Web Personnel Sécurisé. Nous introduisons d'abord en section 2 une famille d'architectures radicalement différente de celle du Web actuel où l'individu exerce un contrôle sur ses données personnelles depuis des composants personnels sécurisés situés aux extrémités du réseau, avec de fortes garanties de non contournement de ses directives. Puis, nous présentons en section 3 deux types de problèmes de recherche sous-jacents à ces architectures : d'une part, la problématique de gestion de données embarquées sur du matériel sécurisé, et d'autre part, la problématique de calcul distribué sécurisé mettant en jeu de grandes populations de serveurs personnels. Nous illustrons cette approche en section 4 avec un exemple d'expérimentation terrain, le « Dossier Médico-Social Partagé » (DMSP), qui exploite ces concepts. Enfin, nous concluons en section 5 avec une discussion sur l'adoptabilité de cette approche.

2. Architectures

Comme cela a été décrit en introduction, un large consensus existe aujourd'hui sur la nécessité de renforcer le contrôle des individus sur la gestion de leurs données personnelles, très insuffisant dans le modèle actuel du Web.

Deux approches principales sont considérées actuellement. La première, suivie par la plupart des grands éditeurs de systèmes de gestion de bases de données (IBM, Microsoft, Oracle, etc.), consiste à améliorer la confiance que les usagers placent dans le système en implantant dans les serveurs de nouvelles mesures renforçant le respect de la vie privée. IBM propose le concept de SGBD hippocratiques [5], rendant le serveur de données responsable de l'application des principes législatifs relatifs à la gestion de données personnelles (consentement, finalité, exposition limitée,

collecte et rétention minimum, audit, etc.). Microsoft propose d'introduire le concept de serveurs de confiance (« *Trustworthy Computing* ») pour promouvoir l'implantation de mesures de sécurité accrues sur les serveurs [11] : sécurisation matérielle apportée par les modules TPM (*Trusted Platform Manager*), réduction du nombre de personnes ayant des droits administrateurs et implantation de principes attenants au respect de la vie privée et de structures de contrôle assermentant les systèmes. De plus, la plupart des grands éditeurs de bases de données ont intégré ces dernières années de nouvelles fonctionnalités liées à la sécurité : chiffrement transparent, possibilité de masquer des données et de brouiller le contenu de certains éléments sensibles des résultats de requêtes SQL autorisées, etc. TrustedDB [6, 7] propose même d'équiper les serveurs de dispositifs matériels sécurisés et impliqués dans l'exécution de manière à garantir un très haut niveau de sécurité. Ces approchent démontrent la nécessité de prendre en compte le respect de la vie privée, de réduire les risques de fuites de données et d'attaques côté serveur. Mais elles ne permettent pas de résoudre les deux problèmes intrinsèques à toute approche serveur : (1) une fuite de données ou une attaque menée avec succès compromet de très grands volumes de données et (2) le modèle étant basé sur la délégation, il peut conduire à des passe-droits et à des usages secondaires indésirables pour les usagers, incontrôlables par ces derniers.

La seconde approche consiste à offrir aux individus des serveurs personnels (réels ou virtuels) pour gérer leurs données de façon décentralisée. Cette approche est prometteuse car elle répond bien aux deux limites intrinsèques de l'approche serveur. Le projet FreedomBox [9] est l'un des pionniers à proposer une plateforme logicielle adaptée à l'architecture d'un *plug computer* (par exemple un Raspberry Pi) permettant aux individus de communiquer de façon anonyme et indépendante du réseau Internet classique. Les approches basées sur des serveurs personnels se démocratisent actuellement et de nombreux projets et startups proposent des solutions à destination du grand public, comme openPDS [10] [23], CozyCloud [11], Younity [12], Lima [13], OwnCloud [14], Tonido [15], Seafile [16], SparkleShare [17], etc. D'autres exemples sont discutés dans [24] et une critique principale est formulée : la difficulté de garantir à l'usager une protection contre les accès dérobés (matériels ou logiciels) potentiels et de lui garantir l'usage qui sera fait de ses données une fois celles-ci transmises hors de

9. FreedomBox : `http://freedomboxfoundation.org/`
10. OpenPDS@MIT : `http://openpds.media.mit.edu/`
11. CozyCloud : `https://www.cozycloud.cc`
12. Younity : `http://getyounity.com/`
13. Lima : `https://meetlima.com/`
14. OwnCloud : `https://owncloud.org/`
15. Tonido : `http://www.tonido.com/`
16. SeaFile : `http://seafile.com/en/home/`
17. SparkleShare : `http://sparkleshare.org/`

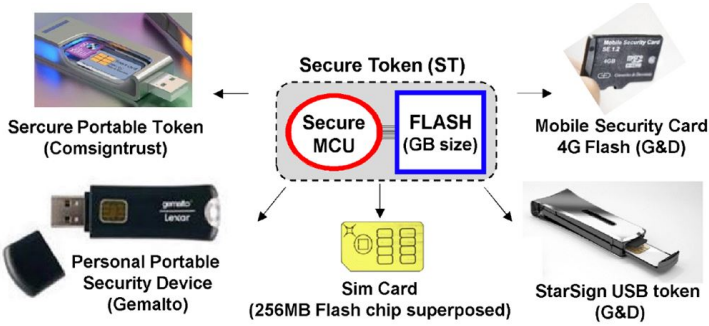

FIGURE 1. Exemples de dispositifs personnels et sécurisés existants.

son serveur personnel. Mais à notre connaissance aucune de ces approches, représentatives de ce que l'on pourrait baptiser le « Web Personnel », n'intègre de composant sécurisé matériellement. Notre vision se distingue donc nettement et pourrait être une préfiguration d'un « Web Personnel Sécurisé » où l'usager pourrait avoir de fortes garanties sur ses propres données et, n'ayant pas lui-même tous les droits sur son propre serveur, pourrait offrir de solides garanties à ceux qui interagissent avec lui.

2.1. Approche

Notre approche est basée sur l'émergence de nouveaux dispositifs matériels sécurisés qui fleurissent actuellement sous différentes formes allant, selon le contexte applicatif, de cartes SIM multimédia nouvelle génération aux clés USB ou cartes SD sécurisées, badges d'authentification ou cartes communicantes (voir Figure 1). Ces dispositifs sont présentés sous des noms différents tels que « *Smart USB Token* » [14], « *Mobile Security Card* [18] » pour Giesecke & Devrient [15], « *Personal Portable Security Device* [19] » pour Gemalto et Lexar, ou « *Secure Portable Token* » [1]. Ils sont fondés sur une architecture commune combinant une puce sécurisée matériellement (ou microcontrôleur sécurisé) avec une mémoire de stockage persistante de grande capacité de type flash NAND.

Notre objectif est donc d'embarquer des composants logiciels permettant de collecter, stocker et partager les données personnelles d'un individu, avec des garanties

18. Les produits « Mobile Security Card » de Giesecke & Devrient combinent une puce sécurisée et une mémoire de stockage de masse de type flash NAND dans une carte microSD. Voir : `http://www.gi-de.com/en/products_and_solutions/products/strong_authentication/Mobile-Security-Card-31488.jsp`

19. Voir à titre d'exemples les produits « Smart Guardian » `http://cardps.com/product/gemalto-smart-guardian` et « Smart Enterprise Guardian » `http://cardps.com/product/gemalto-smart-enterprise-guardian`

tangibles de non contournement. Le dispositif offre un très haut niveau de sécurité : (1) l'attaquant a l'obligation d'être (physiquement) en contact avec le dispositif pour l'attaquer, (2) le dispositif hérite de la sécurité matérielle de la puce sécurisée qui le protège contre les attaques par canaux auxiliaires, (3) le code embarqué est ouvert (*open source*) et peut être prouvé formellement ou certifié par la communauté, ce qui le protège contre les attaques logicielles, (4) la simplicité du serveur lui permet d'être auto administré, ce qui prévient la possibilité d'une attaque de l'administrateur, (5) le ratio coût/bénéfice d'une attaque comparé à un serveur classique est augmenté par les trois premiers points et par le fait qu'une attaque réussie ne compromet que les données d'un seul individu, et (6) le porteur lui-même n'a pas directement accès aux données embarquées, ce qui garantit que les données obtenues provenant d'autres usagers ne seront pas compromises.

Au-delà d'un simple répertoire sécurisé de documents personnels, nous souhaitons permettre le développement de nouvelles applications orientées données et donner la possibilité à des applications existantes d'interroger le dispositif, ce qui nécessite d'organiser les documents de manière structurée, consistante et interrogeable. Nous souhaitons aussi permettre à l'usager de contrôler les règles de partage des données et lui offrir des garanties tangibles de non contournement de ces règles. Ces objectifs combinés nous conduisent à définir un véritable serveur de données, personnel et sécurisé. Les avantages d'un tel serveur sont les suivants : (1) offrir les fonctionnalités principales d'un moteur de base de données (structuration des données, contrôle d'accès, facilités d'interrogation et transactions) et être interopérable avec des sources de données existantes et avec les autres usagers, (2) permettre à l'utilisateur de contrôler le partage de ses propres données (quelles données, avec qui, pour combien de temps, à quelles fins) et garantir les principes de respect de la vie privée (consentement, collecte et rétention minimum, audit) pour ses propres données et celles appartenant à d'autres, et (3) garantir à l'usager un très haut niveau de sécurité et lui offrir un accès déconnecté aux données qu'il ne pourrait obtenir avec un serveur classique.

L'architecture initiale, *Secure Personal Data Servers*, que nous avons proposée dans [1], se base sur une hypothèse de monde fermé et très organisé. Elle s'adapte à certains scénarios d'usage et sert de base à l'application PlugDB/DMSP [20] présentée en Section 4. Nous avons ensuite conçu une version plus ouverte de l'architecture, adaptée à la gestion de l'ensemble des données produites autour d'un individu ou d'un domicile (documents personnels, mais aussi traces de consommation électrique, données issues de capteurs domotiques, traces GPS, etc.), présentée dans [2] et baptisée *Trusted Cells*. Nous envisageons une troisième version de cette architecture, dite *Folk-IS*, dont la particularité est d'être adaptée aux pays les moins avancés (PMA) et caractérisée par une absence d'infrastructure (faible couverture réseau, pas

20. https://project.inria.fr/plugdb/

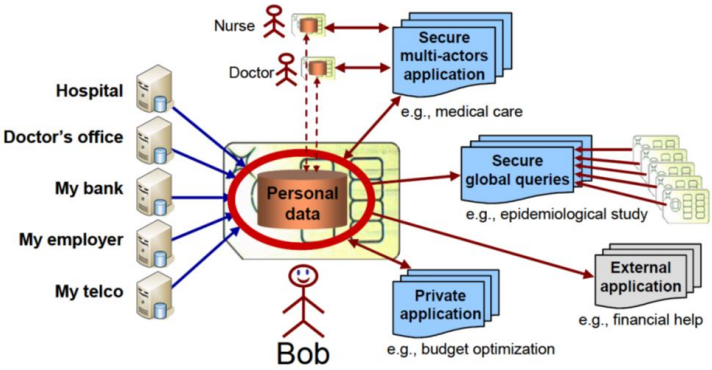

FIGURE 2. L'approche « Serveur Personnel de Données ».

de serveurs centraux, pas d'autorité de certification, etc.), présentée dans [3] et dont nous ne parlerons pas ici. Ces architectures soulèvent des problèmes de recherche pour la communauté base de données dont les sections 3 et 4 sont des exemples. Ces architectures, toutes basées sur l'introduction d'un serveur personnel et sécurisé, préfigurent la voie d'un Web Personnel Sécurisé.

2.2. Architecture « Serveur Personnel de Données » (Secure Personal Data Servers) [1]

L'architecture « Serveur Personnel de Données » (SPD) définit une infrastructure permettant de mettre en œuvre la vision illustrée Figure 2. Les données de l'utilisateur, Bob, sont produites par différentes sources et transmises à son SPD qui peut ensuite répondre aux requêtes d'applications privées (servant les intérêts de Bob), partagées (accédant aux données de Bob depuis d'autres SPD), globales (interrogeant l'ensemble des SPD de façon anonyme) ou externes (accédant aux données de Bob sans SPD).

Le SPD seul ne peut offrir toutes les fonctionnalités (de type base de données) désirées. Nous introduisons donc dans l'architecture un serveur de support responsable d'assurer la durabilité des données et de stocker les messages envoyés à destination des SPD. Ce serveur est honnête mais curieux (il effectue correctement la tâche demandée mais cherche à obtenir de l'information confidentielle), ce qui est l'hypothèse habituelle pour un service de stockage. Les données transmises au serveur de support sont donc chiffrées. Nous parlons ainsi d'*architecture asymétrique*, puisque d'une part nous avons les SPD, de faible puissance, souvent déconnectés, mais très sécurisés, et d'autre part une infrastructure de type Cloud, disponible 24/7, mais dans laquelle notre confiance est toute relative.

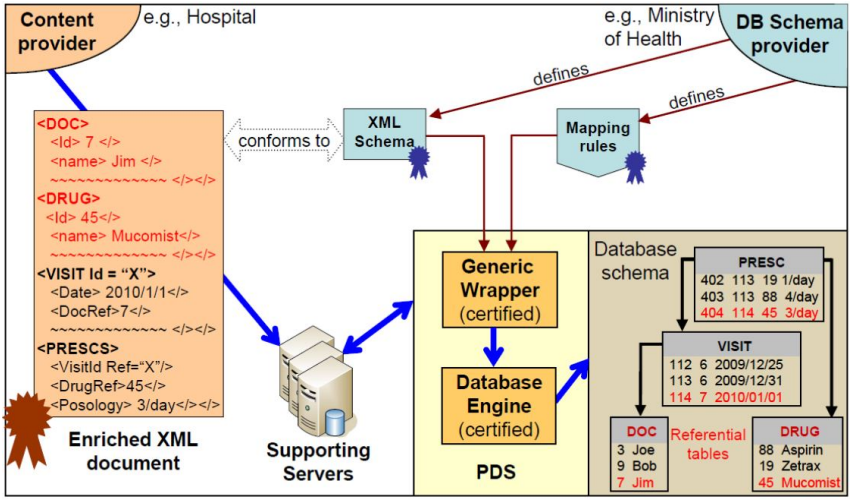

FIGURE 3. Insertion d'un document dans la base de données du SPD.

Pour que les SPD puissent interagir avec les applications, les documents stockés doivent être représentés de manière structurée et interrogeable. Dans cette proposition nous supposons que des schémas de base de données sont définis par des *fournisseurs de schémas* (des agences gouvernementales comme le ministère de la santé ou des consortiums privés comme un groupement de banques) pour chaque domaine d'application. Des *fournisseurs de contenu* fournissent des documents, en XML, suivant un format standard (comme HL7 pour des documents de santé) ou défini par un fournisseur de schéma. Nous considérons que chaque document (p. ex. une prescription médicale) est enrichi de toutes les références nécessaires (p. ex. les informations relatives au médecin qui a établi la prescription). Le document peut ainsi être posté vers un SPD destinataire via le serveur de support, puis téléchargé et transformé par le SPD destinataire en un ensemble de tuples de la base grâce à des règles de transformation fournies par le fournisseur de schéma. Ces règles sont déclaratives et vérifiables. La Figure 3 illustre la transformation d'une prescription médicale transmise par un hôpital, enrichie des références aux médecins et aux médicaments. Nous supposons que la base embarquée est relationnelle mais ce choix n'a pas d'impact sur l'architecture globale.

Les applications sont développées par des *fournisseurs d'applications*, sur les schémas publiés par les fournisseurs de schémas. À chaque application correspond un ensemble de *règles de collecte* spécifiant le sous-ensemble des documents requis pour son bon fonctionnement. Ces règles sont exprimées au niveau des documents

pour être comprises par les usagers et sont transposées au niveau de la base de données pour être évaluées comme des règles de contrôle d'accès.

L'usager exerce un contrôle sur l'usage qui est fait de ses données en acceptant ou refusant les applications (contrairement à une application serveur classique, il peut changer d'application sans perdre ses données), il consent à ce qu'un tiers (un médecin) puisse utiliser son dossier en lui délivrant (physiquement) son SPD, qui peut identifier le médecin comme tel et limite ses droits grâce à une politique d'accès prédéfinie par le fournisseur de schéma (qui fixe une politique d'accès au schéma conforme à la législation pour les différentes catégories de professionnels) ou le fournisseur d'applications. Le porteur du SPD peut aussi définir ses propres règles de masquage sur les documents de la base (et ainsi cacher des documents à certains acteurs). De plus, pour les données personnelles exportées vers un autre SPD, le « donneur » fixe des règles de divulgation minimum (durée de rétention, droits de dissémination), garde la possibilité de supprimer une donnée transmise à tout moment, et définit des règles d'audit à appliquer par le SPD destinataire pour vérifier l'usage qui est fait de ses données. Les données sont publiées auprès du serveur de support et les règles spécifiées par le donneur seront garanties par le ou les SPD destinataires.

Le serveur de support offre une zone de stockage (de données chiffrées) et un service d'horodatage (les SPD ne sont pas équipés d'horloge). Les communications sont asynchrones entre les SPD (ils sont le plus souvent déconnectés) et un service de durabilité (les SPD s'envoient des messages à eux-mêmes) permet la restauration des données d'un SPD perdu à partir d'une passe-phrase connue du porteur.

La sécurité de l'architecture repose sur la sécurité matérielle du SPD, la certification du code embarqué, la ratification de règles déclaratives (règles de transformation, règles de collecte et règles de masquage) et le chiffrement de toute donnée externalisée vers le serveur de support. De plus l'anonymat des SPD se connectant au serveur de support doit être assuré au risque de révéler de l'information sensible (le volume de données transmis à un médecin peut révéler une pathologie). Les SPD intègrent un protocole rendant des communications anonymes. La certification ne concerne que certaines parties du code embarqué indiquées sur la Figure 4.

2.3. Architecture à base de « cellules de confiance » (Trusted Cells) [2]

Les limites de l'approche « Serveur Personnel de Données » sont liées au partage nécessairement très asynchrone, car les SPD sont la plupart du temps déconnectés, et aux limites imposées sur les applications qui sont embarquées dans le SDP et doivent s'adapter à de très faibles ressources. L'architecture « cellules de confiance », présentée dans cette section, repousse ces limites et permet d'envisager des usages plus généraux.

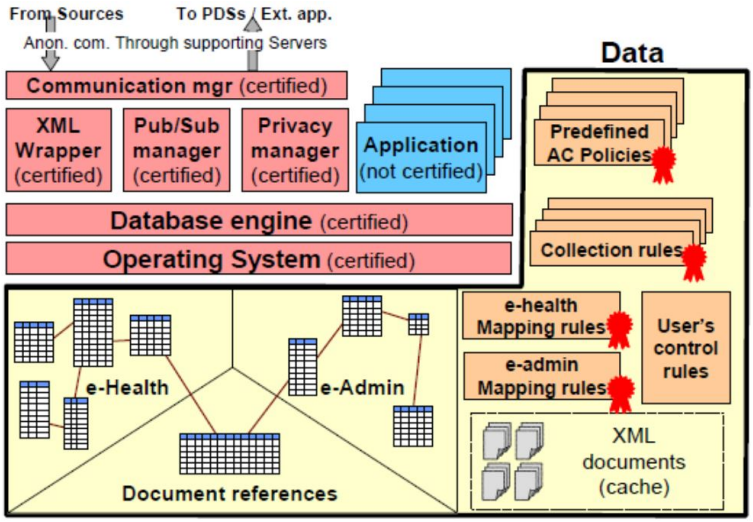

FIGURE 4. Logiciel générique du PDS, applications et bases de données.

Cette architecture se base sur les avancées récentes en matière de matériel sécurisé. AMD incorpore des processeurs de type ARM TrustZone [21] dans ses chips visant le marché des smartphones, tablettes, boîtiers décodeurs et ordinateurs portables. Les processeurs TrustZone disposent d'un 33^{e} bit (matériel) sur le bus, servant à séparer matériellement les instructions provenant d'une zone « riche » du système (ouverte et dans laquelle s'exécutent les applications) d'une zone « sécurisé » (exécutant des modules de code sécurisés). TrustZone donne la possibilité de sécuriser les périphériques (p. ex. une partie de la RAM, les ressources d'entrées sorties comme le clavier, l'écran ou les dispositifs de stockage externe comme la carte micro SD) de manière à les rendre accessibles depuis la zone sécurisée en les isolant de la zone riche. Des entreprises comme NVidia, Sierraware ou Genode Labs permettent de rendre TrustZone utilisable depuis Linux or FreeRTOS, et Xilinx ou Trustonic proposent des plateformes matérielles de développement d'applications TrustZone [16].

Cette évolution nous permet d'envisager une architecture dans laquelle de nombreux dispositifs personnels seraient constamment connectés et dotés de sécurité matérielle (Figure 5). Toute donnée personnelle produite par l'espace personnel d'un utilisateur (son domicile, sa voiture, sa tablette ou son smartphone) pourrait être acheminée vers la cellule de confiance principale (fixe), par exemple intégrée ou

21. http://www.arm.com/products/processors/technologies/trustzone.php

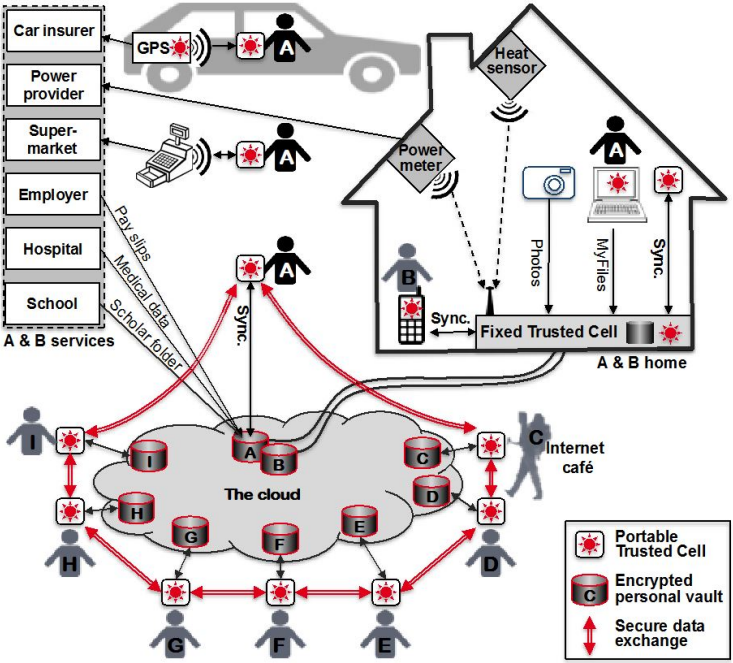

FIGURE 5. **A**lice **(A)** et **B**ob **(B)** sont équipés de cellules de confiance (« *trusted cells* ») fixes et mobiles collectant des données depuis différentes sources et les synchronisant (chiffrées) avec un espace digital personnel sur le Cloud. Tous les utilisateurs équipés de cellules de confiance peuvent partager de façon sécurisée leurs données chiffrées via le Cloud.

connectée à la boîte Internet du domicile. De même, des sources de données présentes dans la maison (compteur électrique intelligent, appareils domotiques) pourraient nourrir cette cellule de confiance principale.

Le SPD n'est pas pour autant absent de l'architecture. Il offre une sécurité matérielle indispensable contre les attaques physiques (et notamment les attaques du propriétaire de la cellule de confiance) que n'offre pas TrustZone. Nous voyons donc le SPD comme un composant bas niveau dans cette architecture, intégré dans la cellule fixe et utilisé pour stocker les clés de chiffrement donnant accès aux données et évaluer les droits d'accès, alors que les ressources moins sécurisées (TrustZone) servent à exécuter des applications utilisant les données et à implanter des contrôles sur l'usage fait des données par ces applications.

2.4. Exploitation de ces architectures

Ces différentes architectures montrent que des alternatives au modèle du Web actuel (fortement dépendant d'acteurs centralisant les données, et en qui il faut avoir confiance) peuvent être envisagées. La confiance est ici réintroduite par l'utilisation d'un dispositif sécurisé. Toutefois, ce dispositif, d'une très grande résistance aux attaques, est caractérisé par de très fortes contraintes. Nous discutons dans la prochaine section de la problématique de réaliser un SGBD sur la base d'un tel dispositif et de la problématique d'effectuer des requêtes distribuées sur une telle architecture.

3. Quelques problèmes de recherche autour des SPD

Le thème de nos recherches tourne autour de la gestion de données en utilisant un SPD, et en particulier la possibilité de permettre d'exécuter des requêtes de type SQL sur les données stockées dans les SPD.

Il faut donc d'une part permettre la gestion *locale* des données, et donc pouvoir embarquer un SGBD relationnel pouvant offrir au moins les opérations de base telles que : sélections, projections, jointures sur clé et calculs d'agrégats. Le moteur embarqué peut ainsi produire différentes vues des données personnelles avec de très fortes garanties de sécurité héritées de la sécurité physique offerte par la puce. Les données sont stockées dans une mémoire flash NAND externe interfacée par un bus avec la puce. Un stockage à distance est possible sur le Cloud et, dans ce cas, le composant personnel gère des métadonnées (liens, attributs décrivant les données, mots clés, tags, clés de chiffrement, etc.) décrivant les données externes chiffrées (les clés de chiffrement restant confinées dans le composant personnel) et le partage de documents peut être établi en partageant ces métadonnées sous le contrôle du propriétaire des données. Notons que le volume de données/métadonnées embarquées dans le composant personnel peut être important dès lors qu'il s'agit de gérer l'ensemble de l'histoire digitale d'un individu.

D'autre part, on peut considérer l'union de toutes les données des utilisateurs présentes sur les SPD comme une unique base de données distribuée, partitionnée horizontalement. Sur chaque partition, le propriétaire des données peut décider du contrôle d'accès qu'il souhaite appliquer. Considérons des requêtes lancées par un organisme de statistiques, ou un laboratoire en épidémiologie, souhaitant par exemple calculer le revenu moyen par commune en France. Ceci peut être calculé par une requête SQL du style (où `MesDonnees` représente la table distribuée qui nous intéresse) :

```
SELECT AVG(revenu), Code_Insee FROM MesDonnees
GROUP BY Code_Insee
```

En effet, chaque individu ne souhaite pas divulguer son revenu, il est donc hors de question de publier les données individuelles précises. Toutefois, les résultats de la

requête d'agrégat globale peuvent être considérés comme suffisamment anonymes. Nous avons étudié comment exécuter de manière sécurisée de telles requêtes, c'est-à-dire comment réussir à ne produire que le résultat final de la requête sans divulguer les valeurs atomiques présentes dans le serveur local.

3.1. Interrogation locale de données : concevoir un SGBD embarqué

Concevoir un tel SGBD embarqué pose des difficultés techniques liées aux fortes contraintes de la puce sécurisée et de la mémoire flash NAND. D'une part, le microcontrôleur de la puce dispose de très peu de RAM (au plus quelques dizaines de Ko) et cette quantité augmente très faiblement depuis des années [22]. D'autre part, le module de flash NAND a de très mauvaises performances d'accès lorsqu'on le soumet à de petites écritures aléatoires. En effet, ce type de mémoire supporte très mal les écritures aléatoires de petite taille. La mémoire est en effet divisée en blocs contenant chacun des pages (p. ex. 64), elles-mêmes découpées en secteurs. La granularité d'écriture est la page (ou le secteur) et les écritures doivent être réalisées séquentiellement dans un même bloc. Une page ne peut être réécrite sans que le bloc contenant cette page n'ait été effacé au préalable. Le nombre d'effacements possibles de chaque bloc est borné (p. ex. 10^4 effacements). Habituellement ces contraintes sont masquées par un module de traduction optimisant l'accès à la mémoire flash qui intègre une couche de traduction d'adresses permettant de réaliser les mises à jour hors place, un ramasse-miettes pour réclamer les blocs dont les données sont devenues obsolètes et un mécanisme de nivellement de l'usure des blocs permettant un effacement équitable. De nombreux travaux (voir [18]) proposent des améliorations du module de traduction. Cependant, avec très peu de RAM, le module de traduction ne peut pas masquer les contraintes de la mémoire flash de manière efficace. De plus, ce module n'est pas dans l'enceinte sécurisée de la puce, ce qui impose de chiffrer les données écrites dans cette mémoire.

Lorsqu'il s'agit de gérer de grands volumes de données, ces contraintes sont difficiles à résoudre car elles mènent à des techniques antagonistes : évaluer des requêtes base de données avec très peu de RAM conduit à indexer massivement les données pour obtenir des performances acceptables, or la maintenance de ces index lors des insertions et mises à jour induit de très nombreuses écritures aléatoires de petite taille.

Les produits SGBD embarqués existants, tels SQLite ou BerkeleyDB, ainsi que les versions légères des SGBD du commerce, comme IDM DB2 Everyplace ou Oracle Database Mobile Server, visent des plateformes personnelles (smartphone ou set-top-box) relativement puissantes. Ces solutions sont clairement inadaptées aux contraintes du dispositif que nous considérons. Certains autres SGBD de l'état

22. Les ressources de la puce cohabitent sur le même dé de silicium, dont la taille doit être réduite pour apporter la sécurité matérielle désirée. Or, la RAM a une faible densité, ce qui conduit les industriels à favoriser les autres composants (notamment la quantité de mémoire stable).

de l'art considèrent spécifiquement les microcontrôleurs sécurisés [25, 9], mais ne considèrent pas de mémoire flash externe. Ils sont adaptés à de faibles volumes de données stockées dans des mémoires internes (technologie flash NOR ou EEPROM) avec des caractéristiques très différentes de la mémoire flash NAND (notamment en termes de granularité des accès). Les techniques pensées dans ce contexte ne peuvent être transposées à notre cas.

Nous avons proposé dans [4] une solution, appelée MiloDB (*Massively Indexed Log-Only Database*), qui répond aux objectifs contradictoires suivants : (1) indexer massivement la base de données, (2) produire exclusivement des écritures séquentielles en mémoire flash et (3) consommer une toute petite quantité de RAM indépendante de la taille de la base de données. Nous proposons pour cela d'organiser toute la base de données (les données, index, tampons, journaux transactionnels, etc.) dans des structures de données purement séquentielles que nous appelons des *containers séquentiels* (CS). Un CS satisfait trois conditions : (1) son contenu est écrit séquentiellement dans les blocs de mémoire flash qui lui sont alloués (une fois écrites, les pages ne sont jamais modifiées ni déplacées) ; (2) de nouveaux blocs peuvent être alloués pour étendre le CS ; (3) un CS est libéré entièrement lorsqu'il devient obsolète (pas de libération partielle de l'espace du CS).

L'intérêt d'adopter une stratégie purement séquentielle est d'éviter les écritures aléatoires par définition. Cependant les traitements appliqués sur les structures séquentielles ne passeront pas à l'échelle. Pour permettre la gestion de grands volumes de données, la base de données séquentielle initiale devra être réorganisée de manière itérative dans de nouvelles structures plus performantes. Cette nouvelle organisation doit être elle aussi produite dans des CS pour satisfaire l'objectif de départ.

MiloDB est le premier serveur embarqué dans un microcontrôleur sécurisé à même de considérer de grands volumes de données stockées dans une mémoire flash NAND et supportant l'ensemble de l'algèbre relationnelle. Le haut degré de sécurité est obtenu grâce aux trois propriétés suivantes : (1) la protection matérielle du microcontrôleur ; (2) l'embarquement du code et son évaluation, permettant de n'externaliser que des résultats autorisés sur les données ; et (3) le stockage chiffré des données dans la mémoire flash NAND.

3.2. Requêtes SQL distribuées [28, 29, 30]

Intéressons-nous maintenant à des requêtes calculant des agrégats sur les données présentes dans les SPD et pouvant être récupérées localement par une requête SQL. En effet, ces requêtes présentent une grande difficulté de calcul sécurisé, puisqu'elles passent par des calculs ensemblistes sur les résultats intermédiaires envoyés par les SPD aux serveurs de support (dite infrastructure de support ou SSI par la suite). En effet, les SPD ont en général une RAM, une puissance de calcul et une connectivité très limitées. Outre l'intérêt intrinsèque de l'augmentation de sécurité par la distribution du calcul, il n'est pas raisonnable pour des raisons techniques

d'imaginer un protocole où un seul SPD téléchargerait toutes les données et effectuerait le calcul entier de l'agrégation. Cependant, l'infrastructure de support ne peut guère aider dans le processus de calcul puisque 1) elle n'est pas autorisée à déchiffrer les résultats intermédiaires et 2) elle ne peut pas regrouper les données selon les valeurs chiffrées des attributs de regroupement (notés $A_G = \{G_i\}$) sans acquérir une certaine connaissance sur la distribution des valeurs sur ces attributs. Le point (2) pose problème puisque la connaissance de la distribution de A_G ouvre la porte à des attaques de fréquence [19]. Dans le cas extrême où A_G contiendrait à la fois des quasi-identifiants [23] et des valeurs sensibles, le chaînage d'attributs serait immédiat. Enfin, l'entité posant la requête (le *querier*) ne peut pas aider non plus, puisqu'elle n'a le droit que de voir le résultat final et non les données atomiques ou les agrégats partiels.

Pour résoudre ce problème, nous avons proposé un protocole d'agrégation générique entre des SPD et une SSI. Ce protocole procède en quatre étapes :

(1) **Collection.** Chaque SPD participant à la requête envoie son propre résultat (*i.e.* des tuples *chiffrés* vérifiant la clause `WHERE`) à la SSI.

(2) **Construction.** La SSI construit des partitions de données chiffrées, dont la taille est calibrée selon la puissance de calcul d'un SPD. Elle envoie ces partitions à des SPD connectés, qui peuvent être différents des SPD impliqués dans la phase 1.

(3) **Agrégation.** Les SPD déchiffrent ces partitions, calculent des agrégats partiels (i.e. agrègent les données d'une partition selon les groupes définis dans la clause `GROUP BY`) et retournent ces résultats intermédiaires à la SSI. Si l'agrégat correspond à l'agrégat final, l'opération de filtrage `HAVING` est exécutée par le PDS.

(4) **Fusion.** La SSI reçoit les données et poursuit de deux manières différentes : soit le résultat obtenu est le résultat final, auquel cas il est envoyé au *querier*, soit plusieurs résultats d'agrégation partiels ont été récupérés et dans ce cas les étapes 2 et 3 sont exécutées de manière itérative.

Ce protocole est suffisamment générique pour gérer des requêtes de la forme suivante, où les fonctions d'agrégat utilisées sont les plus courantes, comme celles présentées par Locher [20], c'est-à-dire distributives, algébriques ou holistiques (voir [33] pour une définition précise) :

```
SELECT <attribut(s) et/ou fonction d'agrégat>
FROM <SPDs>
[WHERE <condition(s)>]
```

23. Un quasi-identifiant est un ensemble d'attributs permettant, pour certaines valeurs spécifiques, d'identifier de manière unique un ou plusieurs individus.

```
[GROUP BY <attribut(s) de partitionnement>]
[HAVING <condition de filtrage>]
SELECT maladie, AVG(age)
FROM PDS-YVELINES WHERE age > 18
GROUP BY maladie
HAVING COUNT(DISTINCT NSS) > 10
```

Exemple. L'ensemble cible est l'ensemble des SPD des Yvelines, regroupés dans une table distribuée virtuelle PDS-YVELINES, et la condition de sélection locale sur l'âge concerne les personnes de plus de 18 ans. Des résultats agrégés sont produits lorsqu'il y a au moins 10 individus dans une ville donnée qui ont une maladie donnée (critère de protection de type 10-anonymat), et donc toutes les maladies « rares » n'apparaîtront plus après la dernière phase 3.

Nos travaux ont porté, entre autres, sur diverses variantes de ce protocole générique, selon le type de chiffrement utilisé, comment la SSI construit les partitions, et quel type d'information est révélé à la SSI. Chaque solution à ses avantages et inconvénients, et peut donc être utile dans un cas spécifique. Nous avons proposé trois types de solutions : *agrégation sécurisée*, *perturbation par bruitage*, et *équipartition des données*. Nous invitons le lecteur intéressé par le code de chaque algorithme, l'étude de ses propriétés de sécurité et ses performances, à lire [28].

3.3. Pistes de recherche et ouverture

Les pistes de recherche que nous explorons concernent à la fois la possibilité d'effectuer des opérations de plus en plus complexes dans l'enceinte sécurisée (par exemple de manière à n'externaliser que des données agrégées/calculées mais pas les données de base) et permettre d'autres types de calculs sécurisés distribués utilisant les données présentes dans les SPD, comme par exemple du *Map/Reduce*, ou encore des algorithmes de graphes appliqués à des réseaux sociaux. Dans la section suivante, nous donnons l'exemple de l'expérimentation terrain DMSP.

4. L'exemple de PlugDB/DMSP

Le Dossier Médico-Social Partagé (DMSP) est une collaboration incluant (entre autres) le Conseil général des Yvelines 78, qui a permis le déploiement et la validation de nos technologies dans le cadre d'une expérimentation sur le terrain. En effet, le DMSP exploite la technologie PlugDB[20] qui implémente les architectures et algorithmes que nous avons brièvement évoqués dans les sections 2 et 3.

4.1. Cadre et motivation de l'expérimentation

Le vieillissement de la population impose d'améliorer le suivi sanitaire des personnes dépendantes à domicile. Dans ce contexte, des informations médicales, sociales et administratives doivent être échangées entre les acteurs intervenant dans la prise en charge (médecins, aides ménagères, aides-soignants, assistantes sociales, auxiliaires de vie, kinésithérapeutes, etc.). Cette coordination passe naturellement par un accès à ces données au chevet du patient ou lorsque celui-ci se rend en consultation dans le cadre de son suivi médico-social et également à distance hors de la présence du patient (par exemple pour des prises de décision par le praticien interrogé au téléphone).

S'agissant de données de santé ou de données sociales, chacun des intervenants doit avoir des droits d'accès différenciés aux données. La personne suivie, avec l'aide de son médecin traitant ou de son entourage, doit pouvoir consentir (ou non) à ce que certains professionnels jouent un rôle sur son dossier. De plus le patient doit pouvoir masquer certaines données particulièrement sensibles avec l'aide de son médecin traitant. Ceci permet de faire face à des situations humainement complexes, comme par exemple un patient se sachant en fin de vie et ne voulant pas le dévoiler pour des raisons humaines et/ou financières, ou désirant ne pas révéler une pathologie à ses proches.

Les différents intervenants du circuit médico-social disposent, en règle générale, de leurs propres logiciels informatiques : logiciel de cabinet médical, de service hospitalier, logiciel infirmier, de coordination gérontologique, etc. Il serait donc souhaitable que les données des dossiers patients puissent se synchroniser avec ces outils préexistants afin d'éviter les doubles saisies.

On peut également imaginer, même si cela n'a pas été testé dans l'expérimentation actuelle, que des données personnelles soient extraites, anonymisées et analysées, dans le cadre d'une étude épidémiologique.

4.2. Les dossiers médicaux existants

Concernant la gestion des dossiers médicaux, trois approches principales se distinguent. La première consiste à interconnecter des systèmes autonomes préexistants dans une infrastructure régionale ou nationale avec un contrôle central minimal, selon l'exemple danois (*Medcom*) ou nord-américain (*eHealth Exchange, NHIN*). Une deuxième approche renforce l'intégration grâce à des index (ou des résumés de données) centralisés, à l'image du projet néerlandais (relancé en 2013 après avoir été abandonné pour la défiance qu'il inspirait aux patients) ou du projet autrichien (ELGA). La troisième approche est totalement centralisée, comme en témoignent le système VistA développé aux USA et le projet DMP national français.

Dans la pratique, la coordination des soins à domicile s'effectue souvent aux travers d'un dossier papier conservé au domicile des personnes suivies. Par exemple,

l'ALDS a mis au point un « Dossier Médical Commun » papier, permettant aux intervenants de reporter les faits importants du suivi des personnes dépendantes. Un intercalaire est disponible dans ce dossier pour chaque type d'intervenant lui permettant de consigner les faits marquants survenus et de les partager avec les autres intervenants. Une feuille générale « tableau de bord » permet aussi de porter toute indication significative.

Quelques solutions informatisées de coordination pour la prise en charge à domicile apparaissent. Par exemple, la société Arcan [24] (groupe Chèque Déjeuner) propose des solutions logicielles mobiles pour la coordination de soins à domicile et lance actuellement une application pour Windows phone. La solution Globule [25] présente le même type de fonctionnalités. Ce genre d'application centralise l'information de coordination sur un serveur central et la rend accessible depuis des applications mobiles. Ces applications nécessitent en général un accès réseau pour avoir accès au dossier centralisé. Certaines permettent parfois de continuer à fonctionner en mode déconnecté, permettant par exemple de saisir certaines données à domicile même hors de toute couverture réseau (les données seront synchronisées sur le serveur dès que le mobile retrouvera un accès réseau). Ces solutions ont un défaut majeur : elles nécessitent que tous les professionnels gravitant autour du patient s'équipent d'un même outil logiciel. Dans la pratique, les structures qui interviennent autour d'un même patient sont très nombreuses (aide sociale diligentée par la mairie, le département, une société privée, acteurs médicaux et sociaux provenant de divers organismes de coordination gérontologique, de cliniques, installés en cabinet, etc.) et toutes ne peuvent s'équiper d'un même outil. Ce type de solution conduit donc à des dossiers partiels (beaucoup moins complets qu'un dossier papier conservé au domicile du patient) et conduit à des doubles saisies (de nombreux organismes ayant déjà leur propre outil informatique). De plus, ces solutions ne remplissent pas les critères de sécurité habituels en matière de gestion de données de santé (identification forte impossible depuis les smartphones qui ne sont pas dotés d'un lecteur de carte CPS, fonctions de respect de la vie privée moins évidentes car il s'agit avant tout d'un dossier de professionnel auquel les patients et l'entourage n'ont pas accès). Enfin ces systèmes peuvent conduire à une défiance de certains intervenants (et même des personnes suivies ou de leurs proches) qui peuvent se sentir en situation de surveillance.

4.3. Approche

L'approche DMSP utilise l'architecture « Serveur Personnel de Données ». Elle consiste à équiper chaque patient d'un SPD embarquant son dossier médico-social, capable d'authentifier chacun des intervenants, de lui donner accès à la vue du dossier correspondant à sa pratique et de se synchroniser sans connexion Internet avec

24. Voir : `http://www.arcan.fr/`
25. Voir : `http://www.globule.net/fr/index.html`

un serveur distant permettant la sauvegarde du dossier et sa synchronisation avec différents logiciels ou chaînes de traitement externes.

Pour permettre la synchronisation entre le SPD et le serveur central sans connexion Internet, nous utilisons le matériel des intervenants (leur lecteur de carte CPS ou leur tablette/smartphone) pour convoyer des paquets chiffrés entre le SPD et le serveur central accessible en zone connectée. Pour permettre l'interopérabilité avec des logiciels ou chaînes de traitement externes, des connecteurs adaptés peuvent être créés en partenariat avec certains éditeurs logiciels. Plus généralement, le SPD peut produire un fichier, suivant un standard établi, que tout logiciel de coordination pourrait à terme reconnaître et savoir intégrer, à l'image de ce qui se pratique aux États-Unis dans le cadre de l'initiative « Blue Button [26] ».

Le patient reste maître de ses données et est dépositaire de l'ensemble de son dossier. La complétude est acquise de fait car tous les intervenants nourrissent le même dossier, celui du patient. Ce dernier peut protéger la confidentialité de son dossier en habilitant certains professionnels à y accéder à distance (l'habilitation est automatique lorsque le professionnel se rend au chevet du patient) et il peut masquer certaines données perçues comme très sensibles avec l'aide de son médecin traitant. La puce sécurisée lui garantit que les droits d'accès associés à chaque intervenant par les autorités sanitaires et sociales, ainsi que ses propres règles de masquage et habilitations, ne pourront pas être contournées.

4.4. Résultats

Nous avons conçu deux plateformes logicielles tournant sur du matériel différent et supportant l'application DMSP. Nous avons aussi conçu un nouveau modèle de masquage, appelé « EBAC » (pour « *Event-Based Access Control* »), que nous avons implémenté sur le schéma de la base de données de l'application DMSP.

La plateforme initiale est basée sur trois éléments : (1) un serveur personnel (SPD) intégré dans une carte SIM de nouvelle génération sur laquelle est superposée une mémoire de type flash NAND (256 MB) intégrée dans un châssis USB ; (2) un porte badge intelligent, lecteur de carte CPS (pour les médecins) et CPA (pour les intervenant sociaux) et qui contient aussi une carte SIM à grande mémoire permettant de convoyer les fichiers de synchronisation entre les serveurs personnels et le serveur central ; (3) un serveur central contenant une réplique (chiffrée) des données du dossier permettant d'alimenter les différentes chaînes de traitement et de régénérer le contenu du dossier en cas de perte du serveur personnel.

26. Initiative permettant au patient de télécharger ses données médicales, stockées par des centres médicaux ou des applications médicales, dans un format texte interprétable par toutes les applications estampillées « Blue Button ». Voir : `http://en.wikipedia.org/wiki/The_Blue_Button`

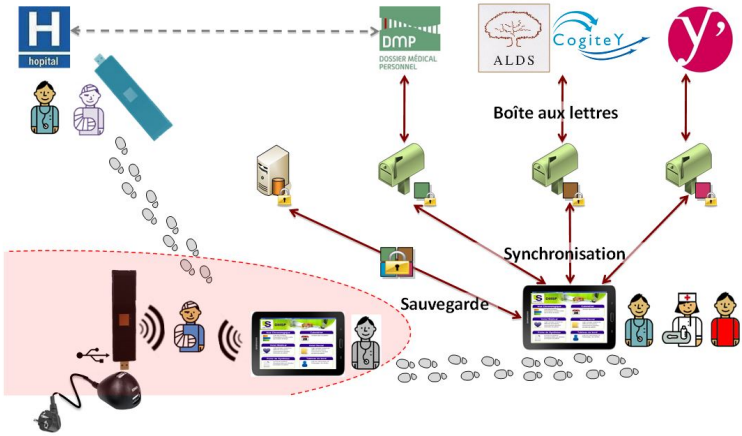

FIGURE 6. Architecture actuelle de la plate-forme PlugDB.

Expérimentation terrain.

Sur la base de cette plateforme, nous avons conduit une expérimentation terrain sur le territoire des Yvelines, pendant 18 mois (de mi-2011 à fin 2012) auprès de 40 patients et 80 professionnels. Les patients et professionnels ont été équipés de mini-PC (modèles *EeePC*) capables de jouer l'application DMSP depuis un navigateur Web, s'interfaçant indifféremment avec le serveur central ou le SPD. Les conclusions sont prometteuses (amélioration de la communication entre praticiens et entre métiers, évolution des pratiques de soins, suppression de la double saisie) mais certains utilisateurs se sont heurtés à une inadéquation de certains éléments de la technologie (trop d'éléments matériels différents à connecter au domicile, obligation d'installer des pilotes spécifiques Gemalto sur les *EeePC* compliquant le déploiement et ayant généré des incidents techniques répétés).

Plateforme logicielle et matérielle actuelle.

Nous avons élaboré une deuxième plateforme (voir Figure 6), corrigeant les défauts de la première. D'abord, nous avons remplacé le SPD de Gemalto par un composant matériel de fonctionnalité équivalente, réalisable par toute PME spécialisée en électronique. Nous avons porté notre moteur de gestion de données sur ce nouveau matériel et avons pu rendre le SPD complètement *plug and play* sur tablettes et smartphones Android. Nous avons aussi doté le SPD d'un module Bluetooth et des primitives nécessaires à un usage sans fil. Nous y avons intégré un lecteur d'empreinte digitale afin de s'affranchir de l'usage des cartes CPS/CPA tout en permettant une authentification forte des professionnels. Dans sa dernière version, le dispositif

a également été complété par un microphone, sans réel impact sur le coût du dispositif, qui permet une modalité d'interaction nouvelle pour stocker des données, souvent pratique dans le cas médical.

Au niveau global, le dossier du patient peut être archivé sur un service de sauvegarde, de manière à pouvoir restaurer le contenu du serveur en cas de perte, sans déperdition de sécurité (les clés de chiffrement sont conservées sous le contrôle du patient). Un mécanisme de synchronisation profite de la venue à domicile des professionnels pour synchroniser le dossier du patient avec le service de sauvegarde et avec certains silos de données distants (Conseil général, coordinations gérontologiques, etc.) dans les deux sens, via des boîtes aux lettres numériques sécurisées. Enfin, le patient peut emporter son serveur personnel avec lui lorsqu'il se rend en consultation.

5. Discussion et conclusion

Le modèle du Web actuel, en ce qui concerne la gestion de données personnelles, ne propose qu'un seul mode de fonctionnement : nos données sont stockées et exploitées par des grandes multinationales auxquelles nous devons faire confiance. Les travaux menés depuis plusieurs années au sein de l'équipe SMIS montrent au contraire qu'une autre voie est possible. Nous avons en effet réalisé un système de gestion de données personnelles offrant les mêmes fonctionnalités qu'un système centralisé, et avons validé le système sur le terrain dans le cadre d'une application traitant des données médico-sociales. Nous espérons que ces travaux puissent constituer les prémisses d'un véritable « Web Personnel Sécurisé ».

Néanmoins, il est légitime de s'interroger sur la capacité qu'auront les utilisateurs à s'émanciper des fournisseurs de services actuels. Il convient de séparer deux problèmes : (i) *l'auto-hébergement*, c'est-à-dire la gestion « physique » des données par l'utilisateur, sur un terminal qu'il contrôlerait et (ii) la gestion « logique » de son patrimoine numérique (p. ex. la définition de règles de contrôle d'accès et d'usage). Les compétences nécessaires de la part de l'utilisateur sont différentes.

L'*auto-hébergement* correspond à un choix délibéré de l'utilisateur de sortir des systèmes mis en place par les grands acteurs du Web. L'utilisateur devra installer et configurer sa propre machine, installer un serveur Web, les applications, configurer le réseau, les DNS, etc. Il faut donc reconnaître qu'à l'heure actuelle, il faut de solides connaissances techniques pour pouvoir réaliser son auto-hébergement. Une approche basée sur l'auto-hébergement complet paraît donc difficilement imaginable, tout du moins dans un futur proche. En revanche, l'approche que nous préconisons ne regroupe pas l'auto-hébergement de toutes les applications, mais simplement d'une brique particulière qui correspond à la gestion des données. De plus, l'un des critères de conception de la partie SGBD du SPD était qu'il ne nécessite aucune intervention d'un utilisateur qualifié pour administrer la base. Ainsi, on peut considérer le SPD

comme une brique matérielle simple, exhibant une caractéristique *plug and play*, qui permet de sécuriser le stockage et le traitement local des données. Toutefois, l'approche que nous considérons travaille de concert avec une *infrastructure de support* puissante et disponible 24/7 (p. ex. Cloud) qu'on suppose *honnête mais curieuse*, c'est-à-dire remplissant son contrat de service, mais ne se privant pas de regarder les données qui y transitent. La solution du SPD ne prône donc pas une approche aussi radicale que l'auto-hébergement complet des données. Par exemple, on pourrait imaginer que le SPD soit branché dans la prise USB de la box Internet d'un utilisateur.

Une autre critique faite à l'auto-hébergement réside dans le fait qu'il est de la responsabilité de l'utilisateur de garantir la disponibilité, la sécurité et la durabilité des données : en d'autres termes, que se passe-t-il si le disque dur comportant toutes les données de l'utilisateur plante ou, dans le cas du *token*, que ce dernier soit détruit ou volé ? C'est là que la synergie entre *token* sécurisé et infrastructure de support intervient : la durabilité des données est garantie par l'infrastructure, toutefois les données sauvegardées sont chiffrées. Il est donc simple de récupérer les données sur le Cloud suite à une panne, sous réserve qu'on soit capable de se rappeler du mot de passe utilisé pour chiffrer les données.

L'auto-hébergement n'est pas la seule alternative envisagée pour le stockage et le traitement de données personnelles : l'une des propositions à la mode est celle d'un Cloud « souverain », c'est-à-dire qui serait garanti par un acteur légitime comme l'État. Notons que cette suggestion est très française, les américains étant bien plus méfiants par rapport à l'État que par rapport à des sociétés privées. Par ailleurs cette proposition ne résout en rien la problématique de la confiance, qui résiderait ici sur un contrat ou une loi sans garantie tangible que le contrat soit suivi, par opposition à l'utilisation de matériels ou techniques où la sécurité est prouvée.

Le problème de la compétence de l'utilisateur moyen se pose indépendamment du fait d'auto-héberger ses données. De plus, cette question sur la capacité qu'aurait l'utilisateur à être en mesure de gérer lui-même son patrimoine numérique est plus subjective. Certes, il paraît actuellement irréaliste de penser que chaque individu serait capable d'écrire des règles de contrôle d'accès complexes sur l'ensemble de ses données personnelles. D'ailleurs, même un individu relativement éclairé aurait du mal à estimer le risque qu'il court en publiant telle ou telle information, qui pourrait être recoupée avec d'autres ou utilisée pour déduire des informations sensibles via des techniques de fouille de données. Néanmoins, la demande de contrôle des utilisateurs va croissant : les diverses affaires sur la publication de communications personnelles lors de changement de politiques de confidentialité de Facebook ont illustré le fait que les utilisateurs veulent contrôler l'accès à leurs données ; en même temps, ce contrôle doit pouvoir être effectué simplement. D'ailleurs, le thème de la confidentialité des données est actuellement un sujet de recherche important en IHM, avec des travaux comme [34] qui font des suggestions automatiques aux

utilisateurs lorsqu'ils semblent vouloir « en dévoiler trop » sur eux-mêmes sur Facebook. Il nous semble toutefois que cette question dépasse largement les questions informatiques et est intriquée avec des considérations juridiques, économiques, sociétales et politiques qui ouvrent la voie à de nombreuses recherches adoptant une approche pluridisciplinaire, comme ce qui peut être fait au sein de l'Institut de la société numérique [27] de Paris-Saclay.

Bibliographie

[1] T. Allard, N. Anciaux, L. Bouganim, Y. Guo, L. Le Folgoc, B. Nguyen, Pucheral P. , Ray I. , Ray I., and Yin S. Secure personal data servers : a vision paper. *36th International Conference on Very Large Data Bases (VLDB)*, pp. 25-35, 2010.

[2] N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, P.Pucheral, I. S. Popa. Trusted Cells : A Sea Change for Personal Data Services. *6th Conference on Innovative Database Research (CIDR)*, 4 p., 2013.

[3] N. Anciaux, L. Bouganim, T. Delot, S. Ilarri, L. Kloul, N. Mitton, P. Pucheral. Opportunistic data services in least developed countries : benefits, challenges and feasibility issues. *SIGMOD Record*, Vol. 43, n° 1, pp. 52-63, 2014.

[4] N. Anciaux, L. Bouganim, P. Pucheral, Y. Guo, L. Le Folgoc. MiloDB : a Personal, Secure and Portable Database Machine. *Distributed and Parallel Databases (DAPD)*, Vol. 32, n° 1, pp. 37-63, 2014.

[5] R. Agrawal, J. Kiernan, R. Srikant, Y. X. Hippocratic Databases. *International Conference on Very Large Data Bases (VLDB)*, pp. 143-154, 2002.

[6] S. Bajaj, R. Sion : TrustedDB : a trusted hardware based database with privacy and data confidentiality. *SIGMOD Conference* 2011 : 205-216

[7] S. Bajaj, R. Sion. TrustedDB : A Trusted Hardware-Based Database with Privacy and Data Confidentiality. *IEEE Transactions on Knowledge and Data Engineering*, 26(3), 752-765, 2014.

[8] G. Blank, G. Bolsover, E. Dubois. A New Privacy Paradox. Global Cyber Security Capacity Centre, Draft Working Paper, 2014.

[9] Bolchini C., Salice F., Schreiber F., Tanca L. Logical and Physical Design Issues for Smart Card Databases, *TOIS*, 2003.

[10] D. Castro, D. How much will PRISM cost the US cloud computing industry. *The Information Technology and Innovation Foundation*. Jan. 2013.

[11] S. Charney. Trustworthy Computing Next. Microsoft, white paper, 2012.

[12] P. Collin, N. Colin. Mission d'expertise sur la fiscalité de l'économie numérique. Ministère des Finances et de l'Économie. Rapport au Ministre de l'économie et des finances, au Ministre du redressement productif, au Ministre délégué chargé du budget et à la Ministre déléguée chargée des petites et moyennes entreprises, de l'innovation et de l'économie numérique. Jan. 2013.

[13] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data. *Official Journal of the EC*, 23, 1995.

[14] Eurosmart. Smart USB token. White paper, Eurosmart, 2008.

[15] Giesecke & Devrient. StarSign® Mobile Security Card SE 1.2, *Datasheet*, 2014.

27. http://digitalsocietyinstitute.com/fr/

[16] J. González and P. Bonnet. Towards an open framework leveraging a trusted execution environment. In *Cyberspace Safety and Security*. Springer, 2013

[17] F. Khatibloo. Personal Identity Management - Preparing For A World Of Consumer-Managed Data. *Forrester Report*, Sept. 30, 2011.

[18] Koltsidas I., Viglas S. D. Data management over flash memory, *SIGMOD*, 2011.

[19] Liu H., Wang H., Chen Y. (2010). Ensuring Data Storage Security against Frequency-based Attacks in Wireless Networks. DCOSS California, p. 201-215.

[20] Locher T. (2009). Foundations of Aggregation and Synchronization in Distributed Systems. ETH Zurich, ISBN 978-3-86628-254-4.

[21] Claire Cain Miller. Revelations of N.S.A. Spying Cost U.S. Tech Companies. *The New York Times*, 21 Mars 2014.

[22] M. Madden, A. Lenhart, S. Cortesi, U. Gasser. Teens and Mobile Apps Privacy. Pew Internet and American Life Project. Août 2013.

[23] Y.A. de Montjoye, E. Shmueli, S.S. Wang, A.A. Pentland. openPDS : Protecting the Privacy of Metadata through SafeAnswers. *PloS one*, 9(7), 2014.

[24] A. Narayanan, V. Toubiana, S. Barocas, H. Nissenbaum, D. Boneh, D. A critical look at decentralized personal data architectures. arXiv preprint arXiv :1202.4503, 2012.

[25] P. Pucheral, L. Bouganim, P. Valduriez, C. Bobineau, PicoDBMS : Scaling down Database Techniques for the Smartcard, *Very Large Data Bases Journal, VLDBJ*, 10(2-3), 2001. Special issue on the best papers from VLDB'2000.

[26] Résolution législative du Parlement européen du 12 mars 2014 sur la proposition de règlement du Parlement européen et du Conseil relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données (règlement général sur la protection des données). 12 mars 2014.

[27] J. Staten. The Cost of PRISM Will Be Larger Than ITIF Projects. Forrester blog. Août 2013.

[28] To Q.C., Nguyen B., Pucheral P. Privacy-Preserving Query Execution using a Decentralized Architecture and Tamper Resistant Hardware. In *EDBT*, Athens, 2014.

[29] To, Q., C., Nguyen, B., Pucheral, P. SQL/AA : Executing SQL on an Asymmetric Architecture, in *40th International Conference on Very Large Databases* (PVLDB), vol. 7(13) : 1625-1628, 2014.

[30] To, Q., C., Nguyen, B., Pucheral, P. Exécution Sécurisée de Requêtes avec Agrégats sur des Données Distribuées, in *Ingénierie des Systèmes d'Information* (ISI), 19(4) : 119-144, 2014.

[31] The World Economic Forum. Personal Data : The Emergence of a New Asset Class. Nov. 2011.

[32] The World Economic Forum. Rethinking Personal Data : Strengthening Trust. May 2012.

[33] Wang H., Lakshmanan V. S. Efficient secure query evaluation over encrypted XML databases. *VLDB*. Seoul, p. 127-138, 2006.

[34] Wang Y., Leon P., G., Acquisti A., Cranor L. F., Forget A., Sadeh N. M. A field trial of privacy nudges for Facebook, in *ACM Conference on Human Factors in Computing Systems* (CHI), 2367-2376, 2014.