Gestion de données personnelles respectueuse de la vie privée

Nicolas Anciaux

▶ **To cite this version:**

Nicolas Anciaux. Gestion de données personnelles respectueuse de la vie privée. Base de données [cs.DB]. Université de Versailles Saint-Quentin-en-Yvelines, 2014. tel-01104999v2

HAL Id: tel-01104999
https://hal.science/tel-01104999v2

Submitted on 11 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gestion de données personnelles respectueuse de la vie privée

Nicolas ANCIAUX

Rapport scientifique pour l'obtention de l'

HABILITATION A DIRIGER LES RECHERCHES EN INFORMATIQUE

Université de Versailles Saint-Quentin-en-Yvelines

Soutenance le 9 décembre 2014

Jury

Rapporteurs :	**Serge ABITEBOUL**	**(Directeur de Recherche, Inria, & Prof., ENS Cachan)**
	Ernesto DAMIANI	**(Prof., Université de Milan)**
	David GROSS-AMBLARD	**(Prof., Université de Rennes 1)**
Membres :	**Philippe AIGRAIN**	**(Dr. HDR, La Quadrature du Net, CEO Sopinspace)**
	Philippe PUCHERAL	**(Prof., U. de Versailles St-Quentin-en-Yvelines)**
Tuteur :	**Luc BOUGANIM**	**(Directeur de Recherche, Inria)**

« *(...) il ne faut pas faire de l'histoire pour l'histoire, on peut le faire, mais c'est comme faire de la philosophie pour la philosophie, de la musique pour la musique, ou de faire de l'art pour l'art, c'est s'installer dans une position d'esthète; il faut faire de l'histoire pour faire de telle sorte que notre présent et notre futur soient possibles autrement que ce qu'a été le passé, bien sûr, ça n'aurait aucun sens de faire l'archéologie du passé, si ça ne permettait pas l'architecture du futur, voire l'architecture du présent* ».

Michel Onfray

Contre histoire de la philosophie, Saison 12 : Pensée post-nazie

Le principe Eichman de notre monde

Analyse d'un petit texte de Gunter Anders: "Nos fils d'Eichman"

Podcast: http://www.franceculture.fr/player/reecouter?play=4870664

La citation se trouve à 54 minutes 28 secondes

Entendu à la radio, mercredi 28 aout à 20h55, soir où j'ai terminé l'écriture de mon manuscrit, et qui m'est apparue pertinente pour certains de nos travaux de recherche en informatique.

Préliminaire

La majorité des résultats présentés dans ce document sont le fruit de collaborations, comme en atteste les publications sur lesquelles ils reposent. Je voudrais donc exprimer toute ma reconnaissance à mes co-auteurs et à tous ceux et toutes celles qui ont contribué à ces travaux.

Table des matières

Introduction

Ce manuscrit présente certains de mes résultats de recherche, relatifs à l'élaboration d'un nouveau modèle de gestion des données personnelles, plus respectueux de la vie privée. Les travaux présentés sont volontairement introduits de manière informelle, espérant les rendre accessibles à un non spécialiste. Ce chapitre d'introduction présente la motivation de ces travaux, les axes de recherche considérés, résume mon parcours scientifique, les projets dans lesquels je m'implique, et introduit le plan du document.

1. Motivation des travaux de recherche

En très peu de temps, 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 Etats-Unis dans l'achat d'informations personnelles [Kha11]. Le Forum Economique Mondial assimile les données personnelles à un « nouveau pétrole » [WEF11], 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 [CoC13].

Les individus, pour pouvoir bénéficier de leurs propres données, passent par des applications en ligne qui rendent ces données exploitables et accessibles. Le respect de la vie privée des individus est alors délégué à l'application, de manière 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 ou pratiqué au risque de perdre ses données. D'autre part, 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 connaitre, 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 impacter 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 [Cas13, Sta13]. IBM implante des serveurs hors des Etats-Unis [Mil14], pour satisfaire ses clients pour qui la localisation géographique (et les passe-droits associés) prend de l'importance. Le Forum Economique Mondial lui-même plaide pour un contrôle accru des usagers sur leurs données [WEF12].

Un consensus économique, politique et social émerge actuellement, pour parvenir à un modèle plus respectueux de la vie privée. Les grands acteurs économiques travaillent dans cet objectif, comme en attestent les travaux du groupe « Trustworthy Computing » de Microsoft[1] 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[2] de la Directive 95/46/CE [Res14, Dir95] qui édicte les principes légaux de respect de la vie privée numérique dans l'Union. Une décision de

---

[1] Voir http://www.microsoft.com/en-us/twc/default.aspx

[2] 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/

la Cours Européenne de mars dernier[3], contraignant Google à offrir aux usagers un outil de droit à l'oubli, va aussi dans le sens d'un durcissement. 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 ainés par le respect de leur intimité [BBD14], et sont plus nombreux aujourd'hui à modifier les paramètres de confidentialité de leur smartphone[4] et à bloquer des applications perçues comme trop invasives [MLC+13]. De plus, ils se tournent vers des moyens de communication plus éphémères, comme Snapchat qui permet d'envoyer des photos qui disparaissent quelques secondes après avoir été visualisées.

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.

Mais il n'y a pas encore de solution technique satisfaisante. Les deux approches actuelles consistent à améliorer la confiance que l'on peut mettre dans les serveurs, ou à introduire des serveurs personnels en charge de la gestion des données de leur propriétaire. La première approche ne permet pas de résoudre les problèmes intrinsèques aux approches centralisées (attaques sophistiquées, modèle basé sur la délégation), et les approches décentralisées sacrifient les fonctionnalités et usages innovants sans toutefois garantir une très grande sécurité.

Nous introduisons une nouvelle approche, que nous appelons le « Web Personnel Sécurisé », où les individus régulent leurs données personnelles depuis des composants personnels sécurisés. Les fonctionnalités principales des solutions centralisées doivent être préservées: durabilité, disponibilité, partage des données. Mais l'exploitation des données se fait avec

---

[3] Le Monde.fr, 13 mai 2014, « Droit à l'oubli : Google débouté par la justice européenne », par Martin Untersinger. http://www.lemonde.fr/technologies/article/2014/05/13/droit-a-l-oubli-google-deboute-par-la-justice-europeenne_4415804_651865.html

[4] Snowden effect: Young people now care about privacy. By Byron Acohido, USA Today, 18 Nov. 2013.

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.

2. Travaux de recherche

Mes travaux de recherche s'intègrent dans les deux axes de recherche sous-jacents au modèle du Web personnel sécurisé. Il s'agit d'une part (***Axe 1***), d'embarquer des techniques de gestion de données dans les composants personnels sécurisés pour en faire de véritables serveurs personnels de données, et d'autre part (***Axe 2***), de définir et mettre en œuvre de nouveaux modèles de gestion de données respectueux de la vie privée, régulant le partage, la collecte, la rétention, et l'usage des données personnelles, et d'intégrer ces modèles dans une architecture suivant une approche Privacy-by-Design, offrant de fortes garanties de non contournement à l'utilisateur.

Axe 1. Concernant la gestion de données embarquées, nous nous concentrons sur des composants sécurisés matériellement, notamment sur des dispositifs dotés de microcontrôleurs sécurisés et disposant d'une mémoire Flash (grande capacité de stockage). Le problème est de concevoir des algorithmes de gestion de données et des structures accélératrices, de façon adaptée aux fortes contraintes du composant: très faible quantité de RAM; caractéristiques techniques des mémoires Flash induisant des performances particulières d'accès en lecture/écriture. Mes contributions sur le sujet portent sur la conception d'un système de gestion de bases de données (SGBD) relationnel embarqué [ABP+14, ABG+10, AAB+07, SAB+07, AAB+09].

Axe 2. Concernant la définition de procédés de gestion de données respectueux de la vie privée, certains de mes travaux ont porté sur la gestion de données dans une base de données intégrant des limites de rétention suivant les objectifs des traitements [HFA09, ABH+08a, ABH+08b, ABH+08c, HAF+06], et ont été conduits dans le cadre d'une coopération avec l'Université de Twente (Pays-Bas). Les contributions les plus récentes se concentrent sur la définition d'architectures Privacy-by-Design reposant sur des puces sécurisées [ABD+14a, ABD+14b, ABB+13, ANP13, AAB+10a], sur des procédés d'exposition minimum lors d'interactions avec des processus externes de prise de décision personnalisée [ABN+15, ABN+13, ABN+12, ANV12], et sur de nouveaux modèles permettant aux individus de réguler le partage de leurs données [AAB+10b, AAB+09, AAB+10c].

Ces deux axes de recherche sont très complémentaires : la gestion ubiquitaire de données personnelles introduit de nouveaux problèmes de préservation de la vie privée, et la gestion de données embarquées dans des composants sécurisés permet d'envisager de nouveaux modèles de sécurisation de bases de données.

Outre les problèmes techniques, ce thème de recherche touche de nombreux enjeux économiques, juridiques, ou sociologiques. Nous ne prétendons pas étudier ces enjeux, mais nous essayons de confronter nos solutions techniques à ces enjeux. 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. Ainsi, le prototypage, l'expérimentation, et les discussions multi disciplinaires font partie intégrante de mon activité de recherche.

3. Parcours

J'obtiens ma thèse de doctorat [Anc04] de l'Université de Versailles Saint-Quentin en Yvelines fin 2004, sous la direction de Philippe Pucheral, Professeur à l'Université de Versailles Saint-Quentin-en-Yvelines. J'aborde dans ma thèse l'étude de l'environnement carte à puce sous l'angle de la gestion de données, l'évaluation de requête en environnement contraint, les techniques de co-design permettant de calibrer à la fois les ressources matérielles de la puce (notamment la RAM) à une application donnée, et inversement, les structures de données et les opérateurs internes à la quantité de RAM disponible [ABP+01, ABP03b], pour parvenir à la définition de bancs d'essais pour SGBD embarqués [ABP+08b]. Des contacts avec Bull CP8 puis Axalto (maintenant intégré à Gemalto) me permettent alors d'avoir accès à des prototypes de carte à puce avancés, dans lesquels j'ai pu porter mon implémentation de PicoDBMS [ABP+01], le premier SGBD complet (supportant l'algèbre relationnelle) tournant sur carte à puce. Ce code a également permis à Axalto de tester son système d'exploitation embarqué, et l'a conduit à appliquer certaines optimisations.

J'obtiens en 2005 un poste de chercheur post-doctorant dans l'équipe bases de données du département d'informatique et du CTIT (Center for Telematics and Information Technology) de l'Université de Twente dirigé par Peter Apers, Professeur à l'Université de Twente. Ma

recherche s'inscrit alors dans le cadre du projet national NWO-VIDI-2005 intitulé "*Context-Aware Data Management Towards Ambient Intelligence*" conduit par Ling Feng, maintenant Professeur à l'Université Tsinghua en Chine. Mon rôle était d'étudier la préservation de l'intimité des usagers évoluant dans un environnement d'intelligence ambiante, doté de nombreux dispositifs communicants. Plus particulièrement, j'ai proposé des procédés de dégradation progressive et automatique des données personnelles, fondés sur l'hypothèse que les objectifs à long terme des applications peuvent être atteints en utilisant des données plus générales (moins précises) que leurs objectifs à court terme. J'ai encadré le stage d'Harold van Heerde sur cette thématique, puis sa thèse (de 2006 à 2010), en collaboration avec Maarten Fokkinga, enseignant chercheur à l'Université de Twente, sous la co-direction de Peter Apers et de Philippe Pucheral. Nos travaux menés dans le cadre de cette thèse ont donné lieu aux publications [HFA09, ABH+08a, ABH+08b, ABH+08c, HAF+06].

Ayant pris pleinement conscience de l'importance sociétale croissante des problèmes liés au respect de la vie privée, et souhaitant me consacrer pleinement à leur étude, je postule au concours de Chargé de Recherche INRIA, et suis recruté en 2006 dans le projet SMIS[5]. A mon arrivée nous montons les projets PlugDB et CG78/DMSP, et j'encadre les travaux menés par les ingénieurs et doctorants dans le cadre de ces projets. Nous abordons alors le problème de la gestion sécurisée de données embarquées et cherchons à concevoir un serveur de données sécurisé. Nous nous intéressons d'abord avec Dennis Shasha, Professeur à l'Université de New York, au problème des traitements distribués entre une base de données embarquée et confidentielle (statique, en lecture seule) et une base de données externe et publique [AAB+07, SAB+07, AAB+09]. Nous démarrons les thèses de Yanli Guo (2008-2011) et de Lionel le Folgoc (2009-2012), que je co-encadre sous la direction de Luc Bouganim, sur la conception d'un moteur de gestion de données embarqué dans un microcontrôleur sécurisé relié à une mémoire Flash externe de grande capacité (Go). Nous jetons ensuite les bases d'une première architecture de gestion de données personnelles respectueuse de la vie privée, conçue dans une approche Privacy-by-Design, en collaboration avec Indrajit et Indrakshi Ray, Professeurs à l'Université de Colorado. Cette étude bénéficie de l'expérience développée dans le cadre des projets PlugDB et CG78/DMSP, qui nous sert de cas d'usage. Les travaux sur le Serveur

[5] Secured and Mobile Information Systems, équipe commune INRIA-UVSQ-CNRS. http://www-smis.inria.fr/

Personnel de Données embarqué donnent lieu aux publications [ABP+14, ABG+10], la première version d'une architecture Privacy-by-Design est présentée à la conférence VLDB'10 [AAB+10a]. Nous appliquons aussi ces travaux au cas des données de santé [AAB+10b, AAB+10c, ABB+08a, ABB+08b]. Nous cherchons aussi à étendre nos techniques de gestion de données embarquées, pensées au départ pour le modèle relationnel, à d'autres modèles de données. En 2012, nous lançons la thèse de Saliha Lallali sur cette thématique. Nous cherchons à généraliser les techniques de gestion de données embarquées, pour couvrir au moins le cas de l'indexation de documents.

Nous nous posons aussi la question du contrôle des données hors du serveur personnel. En 2011, nous démarrons avec Michalis Varzirgiannis, Professeur à l'Ecole Polytechnique, une étude sur le problème de l'exposition du minimum de données personnelles lorsqu'un serveur personnel interagit avec l'extérieur. Ces travaux donnent lieu aux publications [ABN+15, ABN+13, ABN+12, ANV12]. Nous lançons en parallèle une autre étude, en collaboration avec Philippe Bonnet, Professeur à l'ITU au Danemark, sur le contrôle d'usage des données manipulées en dehors du serveur personnel. Une architecture préliminaire a été présentée à CIDR'13 [ABB+13]. Depuis fin 2013, nous envisageons un usage du serveur personnel adapté au contexte des Pays les Moins Avancés, pouvant fonctionner sans aucune infrastructure (réseau, PKI, etc.). J'ai présenté notre vision de ce type d'usage à VLDB'14 [ABD+14a]. Nous étudions actuellement avec le LIRIMA (laboratoire Inria en Afrique) la possibilité de travailler conjointement sur cette base.

4. Implication dans des projets

Les projets auxquels je participe sont tous positionnés sur des thématiques liées à la gestion de données respectueuse de la vie privée. Certains de ces projets sont conduits avec des partenaires d'autres disciplines (juristes, économistes, et sciences humaines et sociales), des partenaires industriels et des représentants de la société civile, pour nous permettre de bien appréhender les aspects transverses liés à la dimension sociétale de notre thématique.

Projet CG78/DMSP (Département des Yvelines, depuis 2007)

Coordinateur: Philippe Pucheral et Nicolas Anciaux (SMIS).

Mon rôle: coordination du projet, référent technique du projet.

Partenaires: Inria, Université de Versailles, Santeos (Atos Origin), Gemalto, ALDS (coordination gérontologique médicale), et COGITEY (coordination sociale).

Objectif: Le projet a pour objectif de concevoir un dossier médico-social mobile et sécurisé facilitant les soins au domicile de personnes dépendantes, et d'expérimenter la solution sur le territoire des Yvelines. Le projet a déjà donné lieu à 3 conventions : 2007-2010 (élaboration de la technologie), 2011-2012 (expérimentation terrain) et 2013-2014 (évolution de la technologie). Au niveau technique, le projet implique la conception et la mise en œuvre d'un serveur personnel de données sur un composant matériel combinant un microcontrôleur sécurisé (type carte à puce) et une grande quantité de la mémoire FLASH dans un format carte SIM, ainsi que la conception et le développement des services attenants de synchronisation et de restauration du serveur embarqué. L'expérimentation terrain s'est déroulée sur 18 mois en 2011/2012, auprès d'une centaine de patients et professionnels médicaux sociaux sur le territoire des Yvelines. Un retour d'expérience a conduit à des adaptations importantes (matérielles et logicielles) du composant personnel sécurisé, nous amenant à faire fabriquer nous-même un nouveau composant doté d'une interface Bluetooth et d'un lecteur d'empreinte digitale. L'ARS Ile de France réalise actuellement un audit de la solution développée dans le projet, afin d'envisager un déploiement plus large (résultat de l'audit prévu pour fin 2014). Une vidéo décrit la solution et une démonstration est disponible. La version actuelle du composant personnel s'interface avec n'importe quel Smartphone ou tablette Android équipé d'un port USB ou du Bluetooth.

Projet CityLab@Inria (Inria Project Lab, depuis juin 2014)

https://citylab.inria.fr/

Coordinateur: Valérie Issarny (Inria@Sillicon Valley & Arles-Mimove).

Mon rôle : Responsable pour le partenaire SMIS.

Partenaires: Arles-Mimove, Clime, Dice, Fun, Myriads, OAK, SMIS, Urbanet et Willow.

Objectif: Le projet étudie les solutions ICT pour la ville intelligente dans un objectif de soutenabilité sociale (et environnementale). J'ai participé à la rédaction de la proposition de projet, et y représente l'équipe SMIS. Notre implication a pour but d'envisager des architectures Privacy-by-Design garantissant la vie privée des citoyens, dans un contexte où ils sont producteurs de données [ABB+14]. Nous nous intéressons en particulier à la capture

de données sociales, produites par les usagers depuis leur smartphone, dans un environnement dit de "social sensing".

ISN (Idex Paris Saclay, depuis dec. 2013)

http://digitalsocietyinstitute.com/

Coordinateurs du pôle: Fabrice Le Guel (ADIS) et Benjamin Nguyen (SMIS).

Mon rôle : Membre du pôle « Vie privée et identité numérique » et responsable pour SMIS du projet PEPS PAIP.

Partenaires: GRACE/LIX, COMETE/LIX, DANTE, CERDI, SAMOVAR, SMIS, RITM.

Objectif: L'Institut de la Société Numérique (ISN) adopte une approche interdisciplinaire, entre disciplines informatiques et sciences humaines économiques et sociales, pour étudier certains défis sociétaux inhérents à la société numérique. Deux pôles ont été lancés: le premier sur le thème de la co-évolution homme/machine, le second sur celui de la vie privée et l'identité numérique dans lequel SMIS est impliqué. Nous avons notamment lancé un projet PEPS financé par le CNRS impliquant les partenaires du pôle, dans lequel nous évaluons, sous forme expérimentale, l'impact sur les usagers de solutions de gestion de données personnelles où l'individu possède (physiquement) le serveur qui régit la dissémination de ses données, par rapport aux solutions centralisées classiques.

Projet KISS (ANR INS, Dec. 2011 – Dec. 2015)

https://project.inria.fr/kiss/

Coordinateur: Philippe Pucheral (SMIS).

Mon rôle : Responsable de la tâche sur l'exposition minimum de données.

Partenaires: Conseil Général des Yvelines, CryptoExpert, Gemalto, Inria (SMIS & SECRET), LIRIS, PRiSM.

Objectif: Le projet vise à produire une alternative crédible à la centralisation systématique des données personnelles sur des serveurs tiers, ouvrant la voie à de nouvelles solutions suivant une approche Privacy-by-Design pour la gestion des données personnelles. L'idée soutenue dans KISS est d'embarquer dans des composants personnels de confiance, des modules logiciels capables d'acquérir, de stocker et de gérer différentes formes d'informations personnelles (ex. bulletins de salaires, factures, données bancaires, médicales,

traces de géolocalisation) selon les applications, et d'en réguler la dissémination [APP+12]. Ces modules logiciels forment un serveur personnel de données, capable de s'interfacer avec des services externes, mais restant sous le contrôle de son propriétaire. Je suis responsable dans ce projet des travaux cherchant à limiter au minimum les données à exposer à des services externes. Nous avons proposé des procédés et algorithmes adaptés à certains scénarios applicatifs validés avec le Conseil Général des Yvelines dans le cadre de la demande d'aide sociale.

Projet DEMOTIS (ANR-ARPEGE, 2009 – 2012)

http://www.demotis.org/

Coordinateur: Philippe Aigrain (Sopinspace).

Mon rôle : Responsable scientifique pour les équipes Inria.

Partenaires : CECOJI, Inria (CACAO, SECRET, SMIS), Sopinspace.

Objectif : Le projet DEMOTIS (Définir, Évaluer et MOdéliser les Technologies de l'Information de Santé) vise à éclairer les limitations et compromis réciproques que l'intrication des domaines juridiques et informatiques impose à la conception d'infrastructures en charge du Dossier Médical Personnalisé (DMP) et celles des dossiers des réseaux de soins liés à certaines affections (SIDA, cancer). Les deux volets du projet, juridique (droit de la santé, des données personnelles ou de la propriété intellectuelle) et informatique (sécurité des bases de données, techniques cryptographiques utilisées pour les protéger, ou anonymisation de données) ont été abordés de manière conjointe par les partenaires.

Projet PlugDB (ANR RNTL, 2007 – 2010)

Coordinateur: Philippe Pucheral (SMIS).

Mon rôle : Responsable de la coordination technique.

Partenaires : ALDS (coordination gérontologique médicale), Gemalto, Inria SMIS, PRiSM, Santeos (filiale d'Atos).

Objectif : Conception d'un serveur personnel de données sur un nouveau composant matériel combinant un microcontrôleur sécurisé (type carte à puce) et une grande quantité de la mémoire FLASH (Go) dans un châssis USB. La solution doit offrir une alternative à la

centralisation des données plus respectueuse de la vie privée, tout en restaurant les propriétés classiques d'un serveur central.

Projet CADMAI (NWO VIDI, 2005 – 2010)

Grant individuel attribué au Professeur Ling Feng (Université de Twente, Pays-Bas)

Mon rôle : Responsable de la tâche relative à la protection de la vie privée.

Objectif : Le projet CADMAI (Context-Aware Data Management Towards Ambient Intelligence) étudie les problèmes liés à la conception et à la mise en œuvre de techniques de gestion de données dans un contexte d'intelligence ambiante. Ling Feng a imaginé le projet, et formé une équipe de cinq doctorants et d'un chercheur post-doctorant. C'est dans le cadre de ce projet que j'ai réalisé mon post-doctorat. Je me suis principalement intéressé à l'intimité que peuvent avoir les individus dans un tel environnement, et ai initié une étude sur la dégradation progressive des données personnelles. J'ai encadré en 2005/2006 le stage de Master d'Harold van Heerde sur la thématique qu'il a ensuite poursuivi en thèse.

5. Plan du document

Ce manuscrit présente un sous-ensemble de mes travaux de recherche. Il est organisé en trois volets liés à l'étude du modèle du Web Personnel Sécurisé présenté en introduction: *architecture* (chapitre 1), *contributions techniques* sous-jacentes (chapitres 2 et 3), et *application* (chapitre 4). Les deux premiers volets reposent sur les articles scientifiques annexés au document, et le troisième décrit une application emblématique de ce que les techniques présentées dans ce manuscrit peuvent apporter et montre la faisabilité de l'approche. Le manuscrit est conçu comme un guide de lecture, accessible aux non spécialistes, présentant de façon informelle les travaux scientifiques placés en annexe. Le contenu de chacun des chapitres est résumé ci-dessous.

Chapitre 1 : ***Architecture.*** Nous introduisons dans ce chapitre 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, tout en continuant à bénéficier des mêmes fonctionnalités qu'une solution centralisée. Le chapitre introduit trois architectures représentatives de différents contextes. Chacune génère des

problèmes scientifiques pour la communauté base de données, dont certains sont étudiés dans les chapitres 2 et 3. Chacune des architectures présentée ici repose sur une publication annexée au document : l'architecture « Serveur Personnel de données » décrite dans la Section 3.1 repose sur [AAB+10a] (Annexe A), l'architecture à base de « Cellules de Confiance » décrite en Section 3.2 repose sur [ABB+13] (Annexe B), et l'architecture « Folk-IS » décrite en Section 3.3 repose sur [ABD+14a] (Annexe C).

Chapitre 2 : ***Serveur Personnel de Données.*** Ce chapitre se concentre sur la conception du Serveur Personnel de Données (SPD) embarqué formant le cœur des architectures présentées au chapitre 1. Nous considérons un dispositif, qui, à l'image de nouveaux objets personnels qui fleurissent actuellement (carte SIM à grande capacité, capteur grande mémoire, carte SD sécurisée, etc.), combine un microcontrôleur sécurisé (type carte à puce) et une mémoire Flash de grande capacité (Go). Nous présentons dans ce chapitre les contraintes posées par cet environnement, leur impact sur la conception d'un SGBD relationnel embarqué dans le dispositif, et les solutions que nous proposons. Ce chapitre repose sur la publication [ABP+14] (Annexe D).

Chapitre 3 : ***Exposition Minimum.*** Ce chapitre montre comment l'introduction d'un SPD permet à un usager de réduire la dissémination de ses données au minimum lorsqu'il interagit avec un service externe. Nous nous plaçons dans le cas de la collecte de données via des formulaires, telle qu'elle est pratiquée par les organisations (aide sociale, banques, etc.) souhaitant ajuster leur offre à la situation spécifique d'un demandeur. Nous montrons en quoi l'approche actuelle, qui détermine a priori les données à divulguer, n'est pas minimale. Nous décrivons notre solution, basée sur une modélisation des objectifs du demandeur sous forme de règles de collecte, confrontées aux données du demandeur dans son SPD, pour permettre cette minimisation. Ce chapitre repose sur la publication [ANV12a] (Annexe E).

Chapitre 4 : ***Application DMSP.*** Plusieurs applications respectueuses de la vie privée ont été développées par l'équipe SMIS. Cette section se focalise sur l'une d'entre elles, fondatrice pour l'équipe et particulièrement représentative de notre approche: l'application « Dossier Médico-Social Personnel » (DMSP). Elle repose sur une architecture très proche de l'architecture « Serveur de Données Personnel » présentée Section 3.1, Chapitre 1. L'application a été

développée pour le Conseil Général des Yvelines, et a donné lieu à une expérimentation terrain. Ce chapitre présente, au travers de cette application, la faisabilité de notre approche, l'intérêt du modèle de gestion de données que nous proposons, et les résultats principaux direct et indirects liés à cette application.

Chapitre 1

Architecture

Notre approche consiste à introduire un serveur personnel sécurisé matériellement, permettant à l'individu d'exercer un contrôle sur ses données tout en préservant les avantages des solutions centralisées : durabilité, disponibilité et partage des données. Ce chapitre motive nos contributions architecturales, introduit trois propositions reposant sur les publications VLDB'10 [AAB+10a] présentée en Annexe A, CIDR'13 [ABB+13] en Annexe B, et VLDB'14 [ABD+14a] en Annexe C, et conclut en résumant les problèmes scientifiques sous-jacents abordés dans les chapitres suivants, et en résumant nos résultats les plus significatifs.

1. Motivation et état de l'art

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 [AKS+02], 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 (« Trustworhty Computing ») pour promouvoir l'implantation de mesures de sécurité accrues sur les serveurs [Cha12] : sécurisation matérielle apportée par les modules TPM, 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 [BaS11,

BaS14] 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 conduite avec succès conduit à compromettre 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[6] 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[7] [MSW+14], CozyCloud[8], Younity[9], Lima[10], OwnCloud[11], Tonido[12], Seafile[13], SparkleShare[14], etc. D'autres exemples sont discutés dans [NTB+12], et une critique principale est formulée : la difficulté de garantir à l'usager une protection contre les accès dérobés (matériel ou logiciels) potentiels et de lui garantir l'usage qui sera fait de ses données une fois celles-ci transmises hors de 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 des garanties à ceux qui interagissent avec lui.

---

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

2. 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 » [Eur08], « Mobile Security Card[15] » pour Giesecke & Devrient [GiD14], « Personal Portable Security Device[16] » pour Gemalto et Lexar, ou « Secure Portable Token » [AAB+10a]. 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.

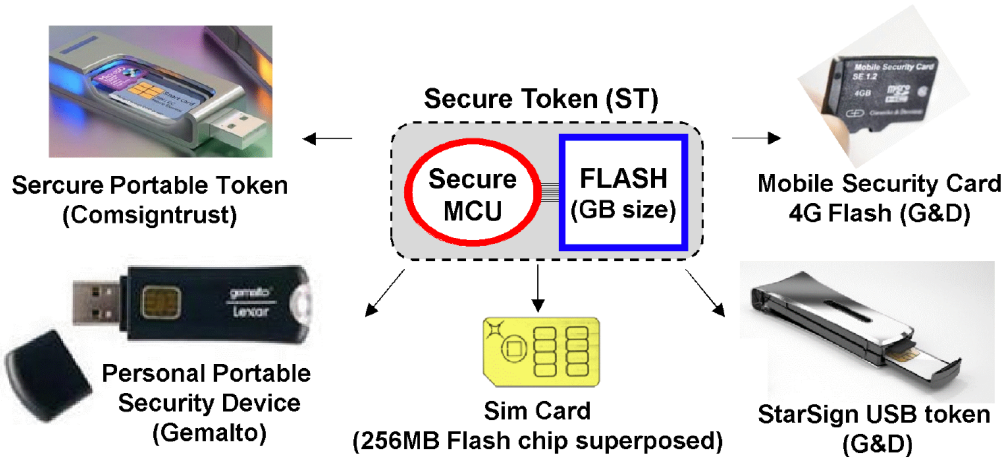


Figure 1. *Exemples de dispositifs personnels et sécurisés existants*

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

[15] 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

[16] 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

(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.

3. Contributions

L'architecture initiale que nous avons proposée 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 DMSP présentée Chapitre 4. Nous présentons ensuite 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.). Enfin, nous présentons une troisième version de cette

architecture, adaptée aux Pays les Moins Avancés, caractérisée par une absence d'infrastructure (faible couverture réseau, pas de serveurs centraux, pas d'autorité de certification, etc.). Ces architectures soulèvent des problèmes de recherche pour la communauté base de données, dont certains font l'objet des chapitres suivants (Chapitre 3 et 4).

3.1. Architecture « Serveur Personnel de Données » (Annexe A, [AAB+10a])

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).

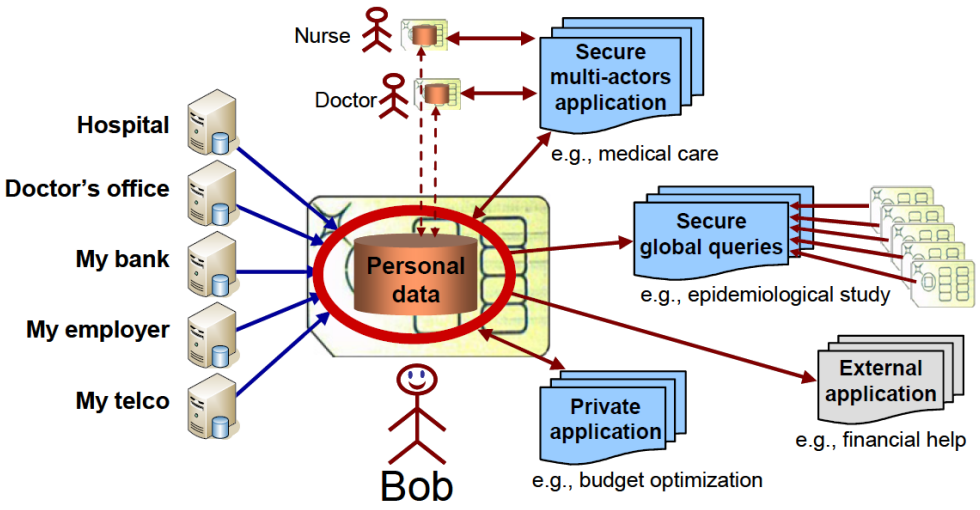


Figure 2. *L'approche "Serveur Personnel de Données".*

Le SPD seul ne peut offrir toutes les fonctionnalités 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.

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 (ex. une prescription médicale) est enrichi de toutes les références nécessaires (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.

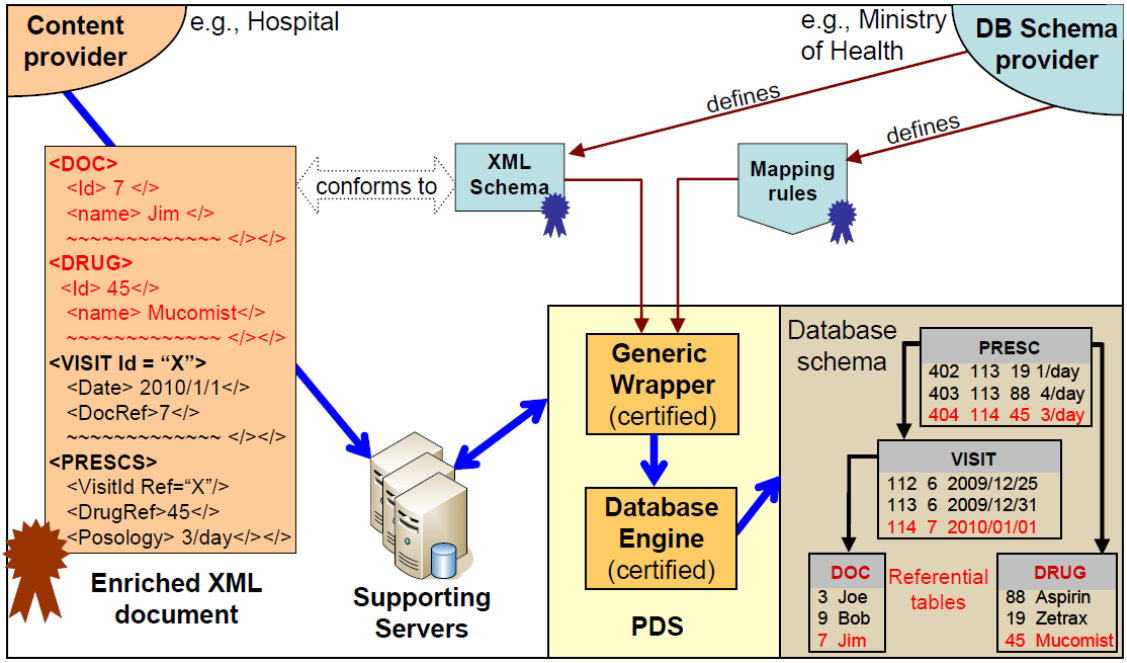


Figure 3. *Insertion d'un document dans la base de données du SPD*

Les applications sont développées par des Fournisseurs d'Applications, sur les schémas publiés par les Fournisseurs de Schémas. A chaque application correspond un ensemble de *règles de collectes* 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'Application. 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 les données transmises 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 de restaurer les données d'un SPD perdu à partir d'une passe-phrase connue du porteur.

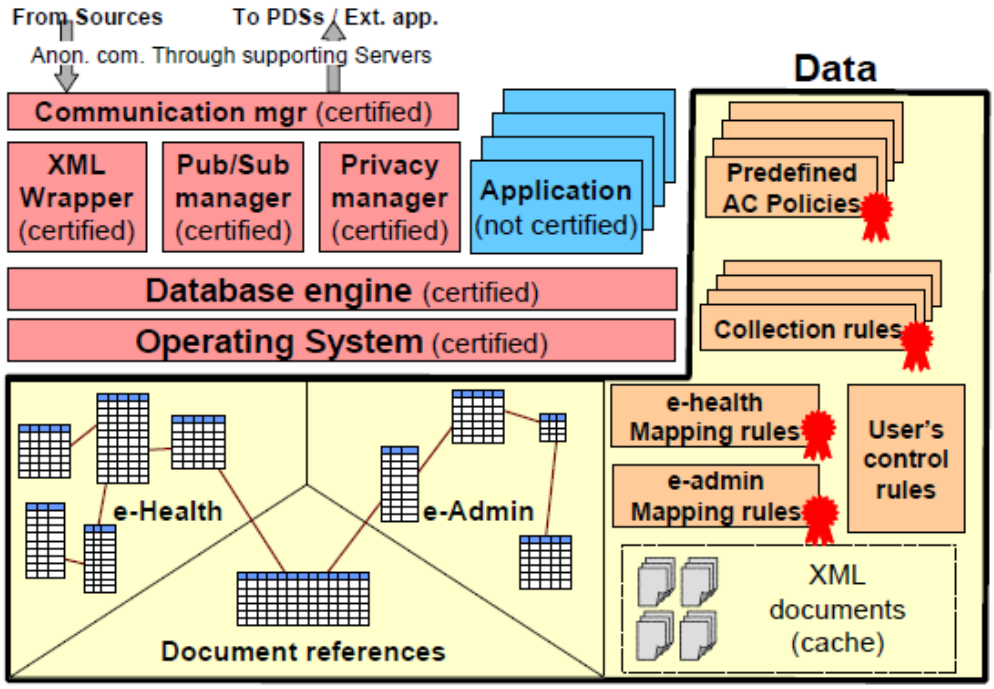


Figure 4. *Logiciel générique du PDS, applications et bases de données*

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 collectes 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.

3.2. Architecture à base de « cellules de confiance » (Annexe B, [ABB+13])

Les limites de l'approche « Serveur de Données Personnel » 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.

Cette architecture se base sur les avancées récentes en matière de matériel sécurisé. AMD incorpore des processeurs de type ARM Trust Zone[17] dans ses chips visant le marché des smart phones, tablettes, boitiers décodeur et ordinateurs portables. Les processeurs TrustZone disposent d'un 33ème 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 (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 [GoB14].

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 connectée à la box 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 contre les attaque 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.

[17] http://www.arm.com/products/processors/technologies/trustzone.php

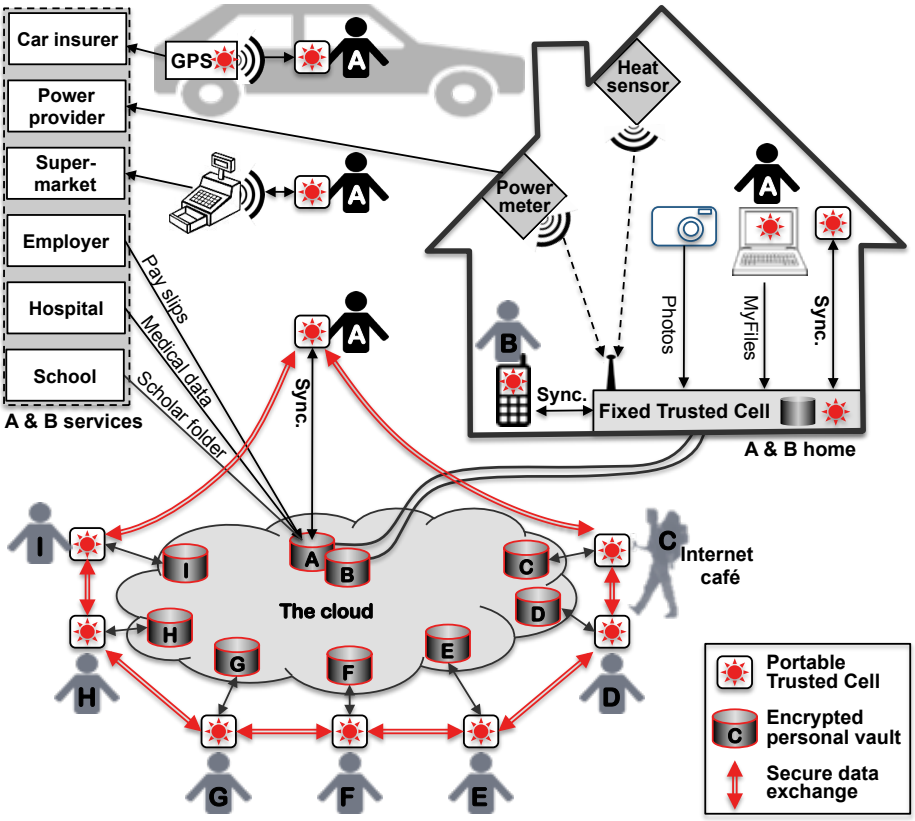


Figure 5. *Alice **(A)** et **B**ob **(B)** sont équipés de cellules de confiance (« trusted cells ») fixes et mobiles, collectant des données depuis différentes sources, et les synchronisant (chiffrées) avec un espace digital personnel sur le Cloud. Tous les utilisateurs équipés de cellules de confiance,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

3.3. Architecture « Folk-IS » (Annexe C, [ABD+14a])

L'approche Folk-IS s'inspire de l'architecture initiale « Serveur Personnel de Données », appliquée dans le cadre des Pays les Moins Avancés (PMA). L'objectif est de fournir aux habitants un dossier numérique personnel (médical, scolaire, etc.) et la possibilité de communiquer entre eux (messages vocaux, email, etc.), et de permettre aux organisations (administrations, acteurs médicaux, ONGs, etc.) d'établir un outil d'échange des données avec les habitants (recensement de population, détection d'épidémie, programmes culturels, suivi médical et sanitaire, etc).

Le déploiement de services orientés données dans les pays les moins avancés se heurte au manque de moyens de communication (couverture 3G partielle et à coût prohibitif), et plus généralement au manque d'infrastructure technique, économique, juridique, politique et organisationnelle. Après discussions avec plusieurs ONG et acteurs locaux, nous avons identifié les besoins suivants en matière de solutions TIC: (1) le manque de sécurité institutionnelle (peu de lois protégeant les individus et peu de recours en cas d'atteinte) et de moyens d'identification (habitants sans carte d'identité) imposent à la solution TIC de garantir par elle-même la sécurité et les principes de respect de la vie privée ; (2) la solution doit présenter des bénéfices personnels immédiats car les acteurs économiques et politiques locaux n'ont pas la capacité d'imposer une solution sur le terrain ; (3) la solution doit être auto-suffisante, c'est-à-dire ne pas reposer sur une amélioration hypothétique des infrastructures ; et (4) le coût par usager doit être très faible et le déploiement de la solution doit pouvoir se faire de manière incrémentale (sans investissement lourd au départ), tout en générant une source de revenus pour de nouveaux emplois locaux.

Nous proposons une solution centrée sur les habitants, basée sur l'introduction d'un composant individuel, portable, à faible coût et sécurisé, appelé *Folk-node*, capable de gérer des données personnelles et de transmettre des messages. Il s'agirait d'un composant à bas coût intégrant au minimum une puce sécurisée, une carte mémoire Flash, et un lecteur d'empreinte digitale (voir la Figure 6). Les acteurs pourront accéder aux e-services (dossiers médicaux, scolaires, services personnels de messagerie) en connectant leur *Folk-node* (sans écran ni clavier) à un terminal (avec écran et clavier). Les terminaux seront détenus par des travailleurs itinérants (des personnes rémunérées pour se rendre dans les villages et partager leur terminal avec les habitants) et des acteurs locaux (enseignants, médecins, etc.). Pour permettre les échanges de données sans couverture internet, les terminaux et *Folk-nodes* seront équipés de services réseaux de routage géographique. Les messages chiffrés seront acheminés de manière transparente lors des interactions *Folk-nodes*/terminaux, en utilisant les déplacements des habitants, pour parvenir de proche en proche jusqu'au destinataire, qui seul pourra accéder au contenu du message. Les délais de transmission pourront être importants (plusieurs jours), mais de nombreuses applications restent compatibles avec ce type d'asynchronisme. Sans infrastructure, les déplacements des habitants et des travailleurs itinérants peuvent assurer seuls

la mise en place globale des e-services, et le système peut profiter de tout élément d'infrastructure réseau existant pour diminuer la latence des échanges.

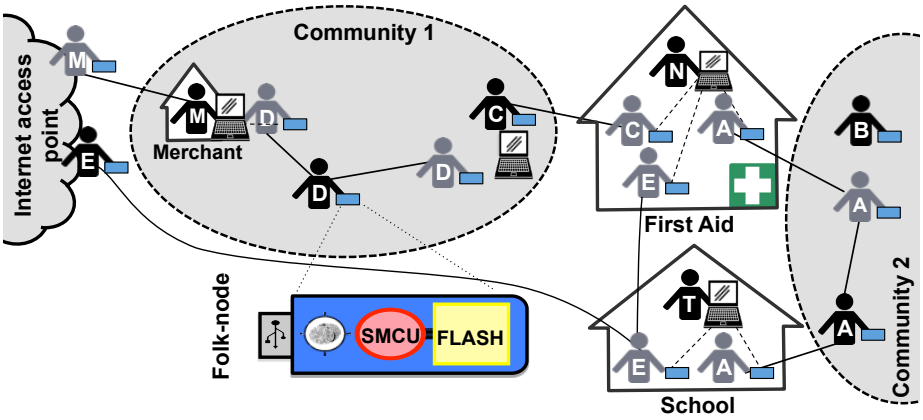


Figure 6. *Deux communautés rurales, leurs habitants (icones noires) et leur déplacement (icones grises), une école et une infirmerie accessibles aux deux communautés, et un point d'accès Internet. Les interactions Folk-nodes/terminaux sont représentées en pointillés. Un message transmis de A à B peut suivre différents chemins (par exemple : A → T → E → Internet), selon les déplacements des acteurs et leurs interactions avec des terminaux.*

4. Conclusion et résultats

Ces différentes architectures montrent que des alternatives au modèle du Web actuel peuvent être envisagées. De nombreux défis scientifiques pour la communauté base de données sont sous-jacents à ces architectures. Ils sont présentés dans les annexes A, B et C du document. Certains fondent mes travaux de recherche actuels et futurs (chapitre 5). Deux de ces défis seront abordés dans les chapitres suivants (Chapitre 3 et 4), et sont transverses aux trois architectures :

Moteur de gestion de données embarqué dans des puces sécurisées. Il s'agit d'embarquer dans des puces sécurisées des fonctionnalités de gestion de données assurant notamment le stockage des données, leur interrogation des données, et l'évaluation de règles d'accès et de politiques de confidentialités. Le défi scientifique associé est décliné selon l'architecture cible dans les annexes (voir Annexe A, Section 5; Annexe B, Section 4, paragraphe « Secure private

store »; Annexe C, Section 3.1). L'étude de ce problème est présentée dans le chapitre 2 et dans l'Annexe D.

Contrôle et sécurisation de la dissémination des données de l'utilisateur. Ce verrou scientifique recoupe plusieurs aspects qui se posent de façon différente selon l'architecture cible (voir Annexe A, Section 4.3; Annexe B, Section 4, paragraphes « Secure sharing », « Controlled collection of sensed data » et « Secure usage and accountability »; et Annexe C, Section 3.3). Le problème est abordé au Chapitre 3, sous l'angle de la collecte minimum d'informations auprès d'un usager qui interagit avec un service externe.

De nombreuses autres perspectives de recherche sont décrites dans les Annexes A, B et C. Je souhaite investiguer plus particulièrement l'une d'entre elles, liée la gestion de données sans infrastructure dans le cadre de l'architecture Folk-IS. Cette perspective est plus détaillée dans la Section 2 du Chapitre « Conclusion et Perspectives ».

Les résultats les plus marquants, ainsi que certaines activités de dissémination, relatifs à ces travaux sur les architectures, sont récapitulés ci-dessous.

Articles scientifiques. L'architecture « SPD » initiale a fait l'objet d'un article VLDB'10 [AAB+10a]. La vision étendue aux cellules de confiance a été présentée à CIDR'13 [ABB+13], et le contexte des Pays les Moins Avancés est considéré dans un papier vision VLDB'14 [ABD+14a] et dans un papier SIGMOD Record [ABD+14b]. L'approche « Serveur Personnel Sécurisé » a donné lieu à deux tutoriaux à MDM'13 [ANP13] et à EDBT'14 [ANP14].

Thèses. J'ai co-encadré les thèses de Yanli Guo [Guo11] et Lionel Le Folgoc [Fol12], qui ont toutes deux contribué au design de l'architecture « SPD ». Avec Yanli, nous avons étudié des techniques cryptographiques adaptées aux structures de la base de données embarquée, et des protocoles cryptographiques permettant d'assurer les communications asynchrones et anonymes entre les SPD via le serveur de support. Avec Lionel, nous avons étudié des techniques de journalisation des mises à jour pour les données du SPD, efficaces en mémoire Flash et permettant d'assurer l'atomicité des transactions. Ces résultats font l'objet d'une partie de leur thèse respective, et ont contribué aux papiers [AAB+10a] sur l'architecture.

Plateformes matérielles et logicielles. Nous avons conçu une plateforme logicielle représentative de l'architecture à base de « cellules de confiance » dans le cadre du projet ANR KISS. L'architecture intègre un agent logiciel implantant une API permettant d'interfacer des applications externes avec le SPD. L'agent tourne à la fois sur systèmes Windows et Linux (PC classique et Raspberry Pi). J'ai participé à la conception de cette API, en ai supervisé une partie des développements, et ai participé à l'élaboration des tutoriaux de prise en main de la plateforme. Nous travaillons aussi en lien avec CozyCloud à la réalisation d'une plateforme de « Web personnel sécurisée » en cherchant à combiner la solution offerte par CozyCloud à la nôtre. La thèse de Paul Tran Van, dont j'ai co-encadré le stage de Master, va démarrer sur un contrat Cifre et aura pour objectif d'étudier la façon d'interfacer un composant personnel et sécurisé au « data system » de CozyCloud pour offrir de fortes garanties sur le partage et les usages des données personnelles gérées dans une instance Cozy.

Dissémination. Notre objectif est de disséminer notre plateforme sous forme de logiciel libre et de matériel libre au travers de l'enseignement. La diffusons a démarré en 2014 auprès des étudiants de l'ENSIIE, dans le cadre du cours « Architectures Privacy-by-Design » que nous avons monté pour cela. L'ENSIIE a fait l'acquisition du matériel nécessaire pour mettre en place l'architecture (composants personnels sécurisés, environnement de développement et sondes matérielles de debug). Nous espérons ainsi sensibiliser les étudiants à la problématique du respect de la vie privée et leur permettre de contribuer à développer des applications innovantes et de nouveaux usages sur cette base. En 2015 nous allons disséminer cette plateforme auprès des étudiants de l'Université de Versailles St-Quentin par le biais du FabLab[18] de l'UVSQ et auprès de élèves ingénieurs de l'INSA Bourges. Nous envisageons aussi une diffusion similaire auprès des élèves ingénieurs de l'INSA Lyon. Concernant la dissémination dans la communauté scientifique, un article ERCIM News décrit une déclinaison de notre architecture telle qu'étudiée dans le cadre du projet ANR KISS [APP+12], et un autre décline cette architecture pour la ville intelligente dans le cadre du projet CityLab@Inria [ABB+14] présentée aussi à Futur en Seine 2014 [Anc14b]. Ce type d'architecture suscite aussi un intérêt pour les acteurs du domaine publicitaire, qui se voient confisquer les données

[18] http://www.fondaterra.com/projet/fablab/

personnelles par les grands acteurs du Web, et qui m'ont invité à leur en faire une présentation au séminaire BIG-DATA'14 [Anc14].

Chapitre 2

Serveur Personnel de Données

Ce chapitre se concentre sur le serveur personnel de données formant le cœur des architectures décentralisées présentées au chapitre précédent et vues comme une préfiguration du Web personnel sécurisé. Notre solution consiste à embarquer le code et les données dans un composant sécurisé, combinant une puce ayant un niveau de sécurité matérielle élevé (comme dans une carte bancaire) avec une mémoire Flash de grande capacité (Go). De nouveaux dispositifs personnels combinant ces deux composantes fleurissent actuellement, sous différents facteurs de forme, en fonction de leur usage applicatif. Notre objectif est ici de concevoir un moteur de gestion de données relationnel embarqué dans ce type de dispositifs. Ce chapitre résume les contraintes techniques de ces dispositifs et leur influence sur la conception du SGBD embarqué, puis présente nos contributions et résultats les plus significatifs. Les contributions techniques de ce chapitre reposent sur la publication VLDB'10 [ABP+14] présentée en Annexe D.

1. Motivation

L'idée d'utiliser un serveur personnel, propriété de l'individu, pour gérer le patrimoine numérique de celui-ci, sous son contrôle effectif, est actuellement soutenue par de nombreux projets et startups. La plupart des initiatives allant dans ce sens se base sur des plateformes personnelles classiques (ordinateur personnel, tablette, smartphone, plug computers, etc.) pour jouer le rôle de serveur personnel. Dans le projet SMIS, nous considérons des plateformes sécurisées matériellement, afin de garantir à l'usager que les règles de partage et de dissémination de ses données personnelles ne pourront être contournées. Les plateformes compatibles avec cette approche combinent un microcontrôleur sécurisé avec la grande capacité de stockage des mémoires de type NAND Flash (voir Chapitre 1, Section 2).

Nous cherchons dans un premier temps à embarquer un SGBD relationnel dans ce type de dispositif, permettant de stocker, indexer et interroger les données en SQL, en supportant au moins les opérations de base : 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 NAND Flash 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. 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.

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[19]. D'autre part, le module de Flash NAND a de très mauvaises performances d'accès lorsqu'on le soumet à de petites écritures aléatoires. 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.

2. Etat de l'art et formulation du problème

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 de l'art considèrent spécifiquement les microcontrôleurs sécurisés [PBV+01, BSS+03], mais ne considèrent pas de mémoire Flash

[19] 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).

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 Flash NAND (notamment en termes de granularité des accès). Les techniques pensées dans ce contexte ne peuvent être transposées à notre cas.

Certaines techniques classiques en bases de données permettent d'obtenir de bonnes performances lorsque les requêtes SQL nécessitent d'évaluer des jointures ou des calculs d'agrégats sur de grands volumes de données par rapport à la quantité de RAM disponible pour effectuer le calcul. Mais, les performances des algorithmes classiques de jointure (par boucle imbriquée, par tri-fusion, par hachage de Grace ou hybride) se détériorent rapidement[20] lorsque la taille du plus petit argument de la jointure dépasse la taille de la RAM [HCL+97]. Des algorithmes plus récents comme le « Jive Join » et le « Slam Join » utilisent des indices de jointure [LiR99], mais nécessitent tout de même une taille de RAM de l'ordre de la racine carrée de la taille de la plus petite des tables impliquée dans la jointure. Dans notre contexte, le ratio entre la taille de la RAM et celle des tables est si petit que la seule solution est de considérer un modèle très fortement indexé, où toutes les jointures (au moins sur clé) sont pré calculées, comme dans le cas d'un entrepôt de données. Par exemple, pour évaluer une jointure en étoile impliquant une très grande table des Faits, les entrepôts de données indexent habituellement la table des Faits sur toutes ses clés étrangères, ce qui revient à pré calculer la jointure avec toutes les tables Dimensions, et sur tous les attributs de ces tables participant à la requête [Sun99, Wei02]. Cependant, une indexation aussi massive nécessite un très grand nombre d'écritures aléatoires de petite taille lors des insertions des données, afin de maintenir les index à jour, ce qui dans notre contexte induirait un coût inacceptable lié aux écritures en Flash NAND.

Les problèmes de la gestion de données en mémoire Flash NAND et de sa couche de traduction ont fait l'objet de nombreux travaux de la communauté bases de données. 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 (ex. 64), elles-mêmes découpées en secteurs. La granularité d'écriture est la page (ou le secteur), et les écritures doivent être réalisées

[20] Avec 10Ko de RAM, joindre des tables de 100Ko par boucle imbriquée conduit à lire la seconde table 10 fois (par bloc), puis la troisième 100 fois, etc.

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 (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 miette 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 [KoV11]) 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 Flash de manière efficace. Dans notre contexte, le microcontrôleur sécurisé peut disposer d'un accès direct au composant Flash (s'il est soudé au dispositif) ou d'un accès via une couche de traduction (si le composant Flash est intégré dans une carte SD ou microSD). Dans le premier cas, les techniques de traduction efficaces consomment de la RAM, ce qui proscrit leur utilisation dans notre contexte. Dans le second cas, les écritures aléatoires par page (ou secteur) sont beaucoup plus coûteuses que les écritures séquentielles (les temps d'écritures aléatoires par page ou secteur de 100 à 1000 fois plus coûteux[21] qu'en séquentiel[22]).

De nombreuses études récentes proposent des solutions au problème du stockage et de l'indexation en mémoire Flash NAND. Les index classiques, comme l'arbre B+, se comportent très mal lorsqu'ils sont implantés en mémoire Flash au-dessus d'une couche de traduction [WCK03]. Les solutions actuelles adaptent en général l'arbre B+ en journalisant les mises à jour pour les intégrer par batch dans l'arbre et minimiser le nombre d'écritures aléatoires en Flash. Le journal des mises à jour doit être indexé en RAM pour assurer des performances satisfaisantes. Les différentes propositions diffèrent par la façon de gérer le journal et l'index en RAM, et par l'impact de cette gestion sur la fréquence de l'intégration du contenu du journal dans l'arbre B+. Pour obtenir un gain important sur le temps d'écriture en Flash, le journal doit être intégré très peu fréquemment dans l'arbre B+, mais cela conduit à consommer plus de

[21] Des tests sur 20 cartes SD récentes montrent les écritures aléatoires sont en moyenne 1300 fois plus couteuses que les écritures séquentielles [SmR]. Nous obtenons des ratios de l'ordre de 100 à 1000 sur les cartes SD et microSD dont nous disposons dans l'équipe.

[22] Cela n'est pas le cas des disques Flash SSDs, qui disposent d'une quantité de RAM interne relativement large (ex. 16 MB), et peuvent offrir un coût d'écriture aléatoire proche de celui d'une écriture séquentielle.

RAM. Inversement, minimiser la consommation RAM conduit à intégrer le journal à l'arbre B+ plus fréquemment, et donc à plus d'écritures aléatoires. Avec la très faible quantité de RAM disponible dans un microcontrôleur, la fréquence d'intégration des mises à jour est de fait élevée, donnant un gain minime sur les écritures aléatoires.

D'autre travaux proposent des index basés sur des structures séquentielles [Arg03, MOP+00, OCG+96], inspirés des systèmes de gestion de fichiers organisés sous forme de journaux [RoO92] ou de fichiers différentiels [SeL76], où une zone séquentielle sert à stocker les nouveaux enregistrements, qui seront intégrés à terme dans le système de gestion données principal. Ces systèmes utilisent aussi de grands tampons en RAM, incompatibles avec nos contraintes. Plus récemment, le système Hyder a été proposé [BRD11] pour gérer sous forme séquentielle une base de données clé valeur en mémoire Flash. Ce système utilise un (seul) arbre binaire pour retrouver l'enregistrement correspondant à une clé donnée. A chaque mise à jour de l'arbre, au lieu de modifier l'arbre en place, le chemin complet depuis la racine jusqu'à l'élément inséré ou modifié est réécrit. Mais cette technique n'est pas adaptée à un système massivement indexé, car les arbres binaires sont adaptés aux index sur clés uniques, mais pas aux index secondaires. D'autres propositions de systèmes clé valeur en Flash, comme SkimpyStash [DeS11], LogBase [VWA+12] ou SILT [LFA11], organisent les paires clé valeur dans des journaux pour éviter les écritures aléatoires, et maintiennent en mémoire (RAM) des index d'une taille proportionnelle à celle de la base de données (au minimum 1 octet par enregistrement), ce qui est incompatible avec les contraintes d'un microcontrôleur.

Pour conclure, concevoir un SGBD, avec des performances acceptables, sur des Go de données, avec une toute petite RAM et une grande mémoire Flash NAND, reste un problème ouvert. Les solutions actuelles sont contradictoires, nécessitant à d'indexer massivement la base de données (petite RAM), avec pour conséquence d'engendrer de très nombreuses mises à jours aléatoires pour maintenir les index (coûteuses en Flash), sauf à consommer plus de RAM pour amortir ces coûts. Notre objectif est de proposer des techniques permettant de rompre ce cercle vicieux.

3. Approche

Notre solution doit répondre aux objectifs contradictoires suivants : (1) indexer massivement la base de données, (2) produire exclusivement des écritures séquentielles en 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 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 devant être elle aussi produite dans des CS pour satisfaire l'objectif de départ.

4. Contributions [ABP+14]

Nos contributions se situent (1) au niveau de l'évaluation de requêtes (sélection, projection, jointure, calculs d'agrégats) avec une petite quantité de RAM bornée, (2) au niveau de l'organisation de la base de données sous forme de CS notamment pour gérer les tampons, l'atomicité de la base de données, les mises à jour, et les index de sélection et de jointure, et (3) au niveau de la protection cryptographique des données contenues dans la Flash (qui n'est pas protégée matériellement comme l'est le microcontrôleur). Les sections suivantes résument les deux premières contributions, la troisième est décrite dans [ABP+14] et les détails sont donnés dans [Guo11].

4.1. Stratégie d'évaluation de requête avec une petite RAM

Avec une très faible quantité de RAM par rapport au volume de données à traiter, nous devons considérer des index généralisés de sélection et de jointure [ABB+09, PBV+01, Sun99, Wei02] qui capturent toutes les relations directes et transitives entre les tuples. Sur cette base, un index généralisé peut être défini avec deux types d'index. Le premier type d'index, que nous

appelons *TJoin* (pour Jointure Transitive), pré calcule la jointure naturelle d'une table référençant d'autres tables (directement ou transitivement). Ainsi, un index TJoin construit sur la table T_i vers la tables T_j, noté $I_{Ti \rightarrow Tj}$, associe à chaque tuple t de la table T_i le tuple t' de la table T_j référencé par t (pour lequel il existe dans la base un chemin de jointures sur clés de t vers t'). Le second type d'index, que nous appelons *TSelect* (pour Sélection Transitive), pré calcule une sélection sur les valeurs d'un attribut d'une table référencée (directement ou transitivement) par une autre table. Un index TSelect construit sur la colonne $T_j.A$ d'une table T_j vers une autre table T_i qui référence T_j, noté $I_{Tj.A \rightarrow Ti}$, associe à chaque valeur v de la colonne $T_j.A$ tous les tuples de T_i qui référencent un tuple de T_j pour lequel la valeur de A est v. Les résultats doivent être triés pour permettre les unions et intersections des listes sans consommer de RAM (par simple fusion de listes triées).

Par exemple, sur le schéma de la base de données considéré sur la Figure 7, les références entre les tables sont indiquées par les flèches en italique : la table T_0 référence directement T_1 et T_3 et référence transitivement T_2, T_4 and T_5. Dans cet exemple, 8 index TJoin sont créés, représentés par les flèches pleines de la Figure 7.a. Les flèches pleines de la Figure 7.b présentent les index TSelect, construits sur les attributs a, b, c, d, e, et f des tables T_0 à T_5, considérant qu'un (seul) attribut de chaque table T_i est indexé. Pour l'attribut c de la table T_2, nous créons donc 3 index TSelect: $I_{T2.c \rightarrow T2}$, $I_{T2.c \rightarrow T1}$, $I_{T2.c \rightarrow T0}$.

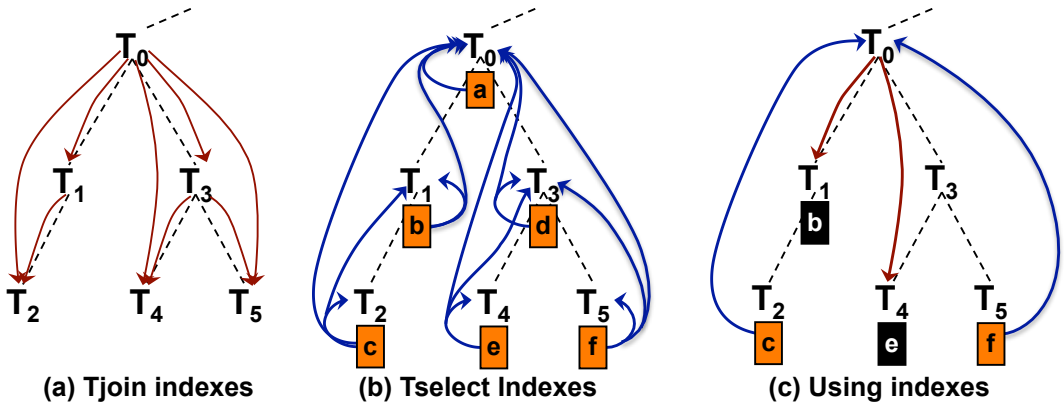


Figure 7. *Exemple d'un schéma d'indexation massif et son utilisation.*

Intuitivement, une requête impliquant des sélections, projections et jointures (SPJ) est évaluée en : (1) traversant les index TSelect construits sur les attributs impliqués dans une

sélection (l'index $I_{Tj.A \rightarrow Ti}$ est impliqué si la requête contient un prédicat de la forme $T_j.A$ et si T_i est la table commune de tous les index TSelect impliqués dans la requête), (2) fusionnant les ensembles triés d'identifiants de tuples de T_i en pipeline (en appliquant une intersection et/ou une union); et (3) traversant les index TJoin nécessaires pour projeter les tuples résultats. La Figure 3.c montre le procédé appliqué pour évaluer une requête joignant toutes les tables, avec le prédicat de sélection ($T_2.c = v_1$ *and* $T_5.f = v_2$) et projetant les attributs $T_1.b$ and $T_4.e$. Cela conduit à : (1) accéder à $I_{T2.c \rightarrow T0}(v_1) \rightarrow S_1$ et $I_{T5.f \rightarrow T0}(v_2) \rightarrow S_2$, (2) faire l'intersection des ensembles triés S_1 et S_2 en pipeline, et (3) utiliser les index $I_{T0 \rightarrow T1}$ et $I_{T0 \rightarrow T4}$ pour retrouver les tuples résultats et projeter les attributs $T_1.b$ et $T_4.e$.

Les requêtes avec une clause de groupement sont plus difficiles à évaluer car la consommation RAM est liée au nombre de groupes résultats. Dans ce cas, le résultat de la partie SPJ de la requête est stocké dans un CS temporaire, puis toute la RAM est utilisée pour calculer les agrégats en plusieurs itérations sur ce CS, produisant une fraction du résultat à chaque itération. Nous proposons diverses stratégies d'optimisation pour ce traitement dans [ABB+09].

4.2. Organisation séquentielle de la base de données

Tampons et atomicité. Les tuples des tables de la base peuvent facilement être organisés séquentiellement, sous forme d'un ensemble de CS nommé ↓DATA, en adoptant une organisation des tables soit en ligne, soit en colonne. L'insertion de nouveaux enregistrements produit des données à ajouter (séquentiellement) aux CS de ↓DATA. Des tampons doivent être utilisés pour stocker les mises à jour à grain fin (potentiellement plus petites que la taille d'une page Flash), jusqu'à obtenir une page pleine à ajouter à un CS donné. Ils sont eux aussi organisés sous forme d'un ensemble de CS, nommé ↓ BUF, et sont utilisés non seulement comme tampon pour tous les autres CS de la base, mais servent aussi à garantir l'atomicité des mises à jour. La bonne gestion des tampons et de l'atomicité de la base de données, afin de minimiser le nombre d'écritures global dans les LC, a fait l'objet d'une partie du travail de thèse de Lionel Le Folgoc [Fol12].

Mises à jour. Les mises à jour (respectivement, les suppressions) d'enregistrements ne peuvent être reportées directement dans les CS (par définition), mais sont journalisées dans un

ensemble de CS dédiés, nommé ↓UPD (resp. ↓DEL). Pour gérer les mises à jour, les anciennes et nouvelles valeurs des attributs mis à jour sont journalisées dans ↓MAJ. Lors de l'exécution des requêtes, le moteur vérifie dans ↓UPD si des mises à jour peuvent impacter leur résultat. Si une mise à jour correspond à un prédicat de requête, la requête est compensée en éliminant les faux positifs (enregistrements qualifiés pour le prédicat de la requête sur leur ancienne valeur mais pas sur la nouvelle) et en intégrant les faux négatifs (enregistrements qualifiés sur leur nouvelle valeur mais pas sur l'ancienne). Ensuite, ↓UPD et ↓DEL sont vérifiés à nouveau lors de l'étape de projection des résultats, afin de projeter les nouvelles valeurs et de retirer du résultat les enregistrements supprimés. Le surcoût est minimisé en indexant les structures ↓UPD et ↓DEL en Flash, et en maintenant en RAM des résumés partiels de ces structures sous forme de filtres de Bloom. La stratégie proposée est simple, mais certains détails d'implémentation plus arides sont exposés dans [Fol12].

Index séquentiels. Bien que certaines structures d'indexation existantes soient naturellement compatibles avec la notion de CS (ex. l'index bitmap), la plupart des index ne l'est pas (ex. les structures arborescentes et celles basées sur du hachage), car l'insertion de nouvelles données génèrerait des mises à jours dans des nœuds ou paquets de hachage existants. Nous avons donc proposé de nouvelles formes d'index séquentiels, compatibles avec la notion de CS, et applicables aux index TSelect et TJoin nécessaires à notre stratégie d'évaluation de requête avec une petite RAM. Ces structures sont décrites en détail dans [ABP+14]. Elles sont dans la suite notées ↓IND.

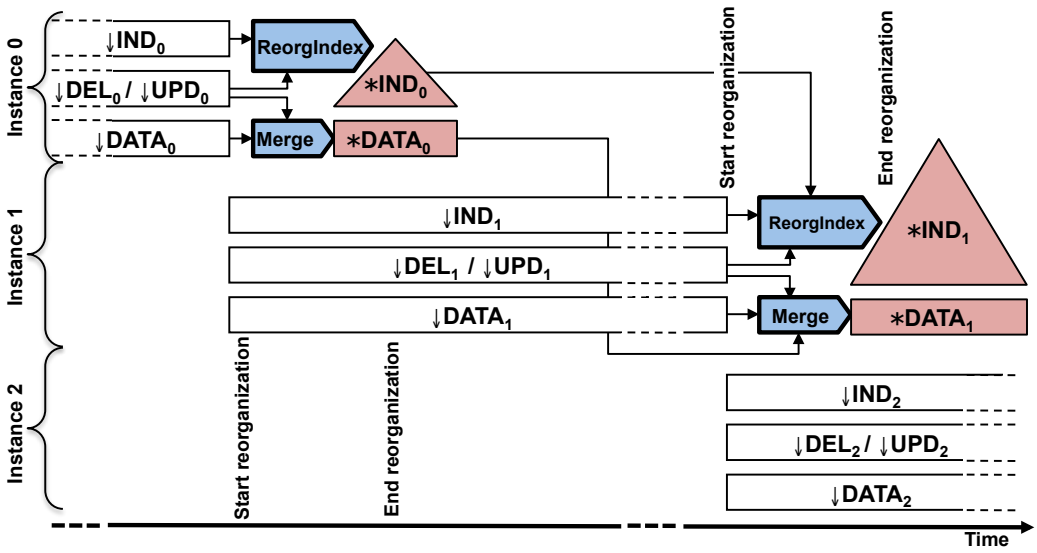


Figure 8. *Le processus de réorganisation.*

Passage à l'échelle. Pour gérer efficacement de grands volumes de données, nous transformons les structures séquentielles de la base de données initiale, notées ↓DB, en une base de données dite **optimale*, générée elle aussi séquentiellement, notée *DB. La base *DB est dite **optimale* dans le sens où elle permet d'obtenir des performances d'interrogation aussi bonnes que si elle avait été construite avec les méthodes de l'état de l'art en ignorant les contraintes de mise à jour de la Flash. Par exemple, la réorganisation de ↓DB en *DB peut donner un ensemble de tables dans lesquelles toutes les modifications et suppressions sont intégrées, et un ensemble d'index organisés à base de structures arborescentes ou basées sur du hachage. Notons que le processus de réorganisation que nous proposons est indépendant du schéma de stockage et d'indexation sous-jacent. Avant toute réorganisation, nous avons $\downarrow DB_0$ = ($\downarrow BUF_0$, $\downarrow DATA_0$, $\downarrow IND_0$, $\downarrow UPD_0$, $\downarrow DEL_0$), l'indice 0 représentant un compteur incrémenté à la fin de chaque réorganisation. Lorsque $\downarrow DB_0$ atteint une limite (en termes de taille ou de performance en interrogation), il faut construire $*DB_0$. La réorganisation déclenche trois actions décrites Figure 8 : (1) le contenu de $\downarrow BUF_0$ est intégré dans les CS cibles de $\downarrow DATA_0$, $\downarrow IND_0$, $\downarrow UPD_0$ et $\downarrow DEL_0$, (2) $\downarrow DB_1$ est alloué et toutes les nouvelles insertions, modifications et suppressions sont redirigées vers $\downarrow DB_1$, et (3) $\downarrow DB_0$ (mise en lecture seule) est réorganisée en $*DB_0$, composée de $*DATA_0$ et $*IND_0$. $*DATA_0$ est construit en intégrant dans $\downarrow DATA_0$ toutes les modifications et suppressions journalisées dans $\downarrow UPD_0$ et $\downarrow DEL_0$ (opération *Merge*). $*IND_0$

est la réorganisation **optimale* de $\downarrow UPD_0$, intégrant les éléments de $\downarrow UPD_0$ et $\downarrow DEL_0$ (opération *ReorgIndex,* cette opération de réorganisation des index est décrite dans [ABP+14]). La base de données est alors composée de $\downarrow DB_1$ (utilisé pour stocker les nouvelles données) et de $*DB_0$ composé de $*DATA_0$ et $*IND_0$. A la fin du processus de réorganisation (lorsque $*DATA_0$ and $*IND_0$ sont construits complètement), tous les CS de $\downarrow DB_0$ sont libérés. La taille de $\downarrow DB_1$ augmente ensuite jusqu'à atteindre une limite qui déclenchera la prochaine phase de réorganisation. La réorganisation est ainsi un processus itératif, très différent des approches journalisant les mises à jour pour ensuite les intégrer par batch, qui produisent toujours des écritures aléatoires.

5. Conclusion et résultats

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 3 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. Cette étude a généré des résultats scientifiques, logiciels, et sert de base à des actions de dissémination, les plus marquants étant récapitulés ci-dessous.

Articles scientifiques. Certains éléments du modèle d'exécution de requêtes embarqué ont été d'abord introduits dans un contexte plus simple, où seules certaines colonnes sensibles étaient embarquées et étaient non modifiables, les autres colonnes restant stockées sur un serveur public. Cette étude a fait l'objet des publications SIGMOD'07 [ABB+07], DAPD'09 [ABB+09] et d'une démonstration à VLDB'07 [SAB+07]. L'étude présentée dans ce chapitre a consisté à embarquer toute la base dans le composant personnel, en supportant les mises à jour, ce qui a fait l'objet d'une démonstration à SIGMOD'10 [ABG+10] et d'un article journal DAPD'14 [ABP+14].

Thèses. Les thèses de Yanli Guo [Guo11], Lionel Le Folgoc [Fol12], et Saliha Lallali (en cours) considèrent la conception de techniques de gestion de données pour un serveur

embarqué. Avec Yanli, nous avons proposé des techniques de protection cryptographique des données et des index adaptées au stockage en Flash et aux motifs d'accès et structures du SGBD embarqué. Avec Lionel, nous nous sommes concentrés sur l'organisation des tampons et sur l'atomicité transactionnelle. Leurs travaux de thèse ont contribué aux publications [ABG+10, ABP+14]. Avec Saliha, nous étudions actuellement la généralisation des techniques proposées pour le relationnel à d'autres modèles (clé-valeur, recherche de documents, etc.).

Logiciels. Le logiciel embarqué, nommé *PlugDB-engine*, est développé sur la base du design présenté ici. Il a fait l'objet de trois dépôts successifs à l'Agence de Protection des Programmes [ABP+08, ABP+09a, ABP+11], et un nouveau dépôt est en cours pour fin 2014. J'en pilote les développements, qui impliquent depuis 2007 des doctorants et ingénieurs de l'équipe SMIS. Ce logiciel est utilisé dans de nombreuses activités de l'équipe décrites dans ce document : l'application DMSP (voir Chapitre 4), les projets CG78/DMSP, PlugDB et KISS (voir Introduction, Section 4), le module SIPD1 et 2 à l'ENSIIE (Chapitre 1, Section 4), etc.

Dissémination et transfert. Un contrat de transfert du logiciel PlugDB-engine est actuellement en cours de signature avec la startup CozyCloud, qui propose une solution de Cloud personnel. Le transfert vise à interfacer le « data system » de CozyCloud avec PlugDB-engine, de manière à maintenir des fichiers chiffrés dans l'instance Cozy de l'usager, et les clés de chiffrement correspondantes dans PlugDB-engine, associées à des métadonnées décrivant les fichiers et sur lesquels définir des règles d'accès. En couplant la solution Cozy avec PlugDB-engine, nous souhaitons offrir aux usagers une première version d'un « Web Personnel Sécurisé » décrite en introduction du manuscrit.

Nos perspectives de recherche consistent à embarquer toujours plus de fonctionnalités de gestion de données dans le serveur personnel, pour pouvoir réaliser en local le plus de calculs possibles et n'externaliser que les résultats de ces calculs, et pour être en mesure d'évaluer des règles d'accès plus riches dans l'enceinte sécurisée. Ces perspectives sont détaillées dans la Section 2 du Chapitre « Conclusion et Perspectives ».

Chapitre 3
Exposition Minimum

Ce chapitre montre comment le serveur personnel d'un usager peut permettre une minimisation de l'exposition des données d'un individu lors d'une interaction avec un service externe. Nous nous plaçons dans le cadre de la collecte de données via des formulaires, telle qu'elle est pratiquée par les organisations (aide sociale, banques, assurances, administration, etc.) souhaitant ajuster leur offre à la situation spécifique d'un demandeur. Ce chapitre introduit le contexte et le mode opératoire actuel, puis présente nos contributions techniques dans le cas de décisions modélisables par des classifieurs multi-labels (ensembles d'arbres de décision), et conclut par un résumé de nos résultats principaux. Les contributions techniques du chapitre reposent sur la publication [ANV12a] présentée en Annexe E.

1. Contexte

La directive *EU95/46/CE* [Dir95] fait référence en Europe concernant la protection des données à caractère personnel, et prône l'application de principes de collecte (et de rétention) minimum des données personnelles. Leur application permet notamment de réduire les effets d'une fuite de données postérieure à la collecte (causée par une attaque, une négligence, ou un usage détourné de la finalité de la collecte). Cette directive stipule que les données collectées doivent être « ***non excessives*** *au regard des finalités pour lesquelles elles sont collectées* ». Suite au scandale PRISM révélant les accès de la NSA aux données des usagers collectées par Google, Facebook, Microsoft, Apple, etc., la Commission Européenne procède actuellement à une refonte de cette directive. Le projet actuel [Res14], adopté en première lecture le 12 mars 2014, renforce ces principes en stipulant que les données personnelles doivent être « *limitées au* ***minimum*** *nécessaire au regard des finalités pour lesquelles elles sont traitées* ».

L'application du principe de collecte minimum participe aussi à réduire les coûts pour les organisations. Le traitement des données et leur archivage nécessitent parfois des ressources importantes et l'intervention d'employés. Les coûts engendrés dépendent grandement de la quantité d'information à traiter. Par exemple, les demandes d'aide sociale sollicitées auprès des Conseils Généraux se font au travers de formulaires très complets, comme le formulaire GEVA (« Guide d'EVAluation » des besoins de compensation de la personne dépendantes), découpés

en plusieurs volets (médical, social, ressources financières, entourage, etc.) et comportant des dizaines de pages et des centaines de champs. Les demandes sont nombreuses (60.000 par an pour le seul Conseil Général des Yvelines) et impliquent beaucoup d'employés (160 personnes au CG78) chargés de vérifier l'information renseignée, de calibrer l'aide à apporter, puis d'archiver les demandes (rejetées et acceptées) pour pouvoir justifier la décision en cas de contestation, en attester la nature non discriminante, et permettre des audits réguliers.

Dans la pratique, les formulaires à renseigner sont construits par les organisations en faisant l'union de tous les attributs pouvant avoir un impact sur la décision finale d'offre de service. Pourtant, pour un individu donné, seul un sous ensemble de l'information est pertinent. Par exemple, une personne dépendante pourrait bénéficier d'une aide financière pour employer une aide journalière à domicile dans les cas suivants: (i) sa pension de retraite annuelle est inférieure à 30.000€ et elle a plus de 80 ans, (ii) sa pension de retraite est inférieure à 10.000€ quel que soit son âge, ou (iii) son score Groupe Iso-Ressources (GIR) indiquant son niveau de dépendance (sur une échelle de 1 à 6) est supérieur à 2. Pour une personne p_1 = [*pension* = 25.000, *age* = 81, *GIR* = 1] l'ensemble minimum à produire serait [*pension*, *age*]. Pour une personne p_2 = [*pension* = 40.000, *age* = 60, *GIR* = 3], il suffirait de produire [*GIR*]. Cet exemple simpliste montre que le contenu minimum du formulaire ne peut être produit a priori, mais dépend des valeurs des attributs demandés pour la personne concernée.

Notre objectif est de proposer des techniques permettant la minimisation des données à renseigner dans le formulaire, et respectant les hypothèses suivantes : (1) l'offre de service obtenue à partir du formulaire minimisé doit être identique à celle qui aurait été obtenue à partir du formulaire complet et les valeur des attributs du demandeur justifiant l'offre de service doivent pouvoir être connues (et archivées) par l'organisation ; (2) le processus de décision ne doit pas être exposé au demandeur car il peut révéler le modèle d'affaire (« business model ») de l'organisation (prêts bancaires, contrats d'assurance) ou inciter à la fraude (falsification de certaines valeurs d'attributs pour obtenir des bénéfices supplémentaires) ; notre solution doit aussi être (3) adaptée à des formulaires de grande taille, rencontrés couramment dans le cas de l'attribution d'aides sociales, ou lors des interactions avec les banques, les compagnies

d'assurance, ou administration fiscale ; enfin (4) le principe proposé doit pouvoir convenir pour un large spectre de processus de décision.

2. Etat de l'art

La transposition de principes légaux dans les systèmes informatiques a été la base de nombreux travaux au cours de la dernière décennie. Des exemples emblématiques incluent la plateforme P3P [CLM+02], les langages de définition de politiques de vie privée comme EPAL [AHK+03], ou encore les bases de données Hippocratiques [AKS+02]. La technologie P3P permet de mettre en évidence des problèmes d'incompatibilité de politiques de confidentialité entre un individu et un service, mais ne permet pas de choisir un sous ensemble des données à exposer. Les langages de définition de politiques de confidentialité qui ont été proposés, comme EPAL, XACML [Mos05] ou WSPL [And04], n'ont pas non plus été introduits dans l'objectif de minimiser la collecte des données. En revanche, l'architecture d'une base de données Hippocratique repose sur dix principes de respect de la vie privée tirés de la législation, qui incluent bien la collecte limitée des données. Un SGBD Hippocratique limite la collecte d'informations en associant à chaque objectif de traitement l'ensemble des attributs requis pour atteindre cet objectif. Toutefois, cette solution fait l'hypothèse que les données utiles et inutiles au traitement pour atteindre l'objectif peuvent être déterminées en amont de la collecte. Comme nous l'avons montré Section 1, c'est peut-être vrai dans certains cas simples, mais cela n'est en général pas le cas des processus de décision complexes.

Un domaine reposant sur des techniques proches de celles utilisées dans notre étude est le domaine de la négociation automatique de confiance, notamment dans le contexte des modèles de contrôle d'accès ouverts, où les décisions d'accès résultent d'une confrontation entre les politiques de contrôle d'accès des parties et les valeurs d'attributs qui les caractérisent. Un petit nombre de travaux suit une approche de collecte minimale, nommée dans ce contexte *exposition minimale* [ADF+12, CCK+05, YFA+08]. Ces travaux minimisent le nombre de valeurs d'attributs à exposer par les parties lors de la phase de négociation automatique, pour permettre la prise de décision (donner ou pas l'accès à une ressource). Toutefois le problème et les solutions sont différents pour deux raisons essentielles. Tout d'abord, les processus de prise de décision que nous considérons sont plus complexes que ceux sous-jacents au contrôle d'accès.

Les règles de collecte que nous considérons modélisent des classifieurs multi labels (ensemble d'arbres de décision), car de nombreuses dimensions peuvent être considérées (ex. pour une demande de prêt bancaire : taux plus faible, durée plus longue, assurance réduite, etc.) dont chacune peut impacter l'offre finale proposée à l'utilisateur. D'autre part, dans notre contexte la prise de décision nécessite de très grandes quantités de données personnelles (les formulaires contiennent souvent des centaines de champs), tandis que dans le domaine du contrôle d'accès, seules quelques autorisations sont prises en compte (ex. jusqu'à 35 dans les évaluations de performance présentées dans [ADF+12]). Ces travaux ne peuvent donc pas être réutilisés dans notre contexte, puisqu'ils ne sont pas pertinents en termes d'expressivité et de passage à l'échelle.

3. Approche

Le procédé actuel pour calibrer une offre de service à la situation particulière du demandeur, illustré Figure 9, se déroule selon les étapes suivantes : (1) le demandeur récupère le formulaire d'application vierge correspondant à sa demande ; (2) il le renseigne conformément à sa situation personnelle et le transmet à l'organisme concerné ; (3) l'organisme vérifie la validité des informations transmises (en utilisant des preuves de provenance ou en croisant les données avec des informations dont elle dispose) et détermine l'offre de service correspondante. L'organisme (4) archive le formulaire et la décision correspondante, et (5) transmet la décision à l'intéressé.

Notre approche pour réduire le formulaire tout en restant conforme à cette pratique, est basée sur trois ingrédients : des règles de collectes modélisant le processus de décision de l'organisme, les données du demandeur modélisées de manière adaptée aux règles de collectes, et une métrique d'exposition des données permettant d'évaluer la quantité d'informations personnelles présentes dans un formulaire donné. De plus, notre solution doit aussi permettre de confronter les règles de collectes et les données du demandeur sans divulguer ni les règles au demandeur, ni les données (n'apparaissant pas dans le résultat) à l'organisme. Pour cela, un algorithme de minimisation de l'exposition doit déterminer parmi les données du demandeur, le sous ensemble minimum permettant d'offrir au demandeur tous les avantages auxquels il a droit, et cette minimisation doit se faire dans l'enceinte sécurisée du serveur personnel.

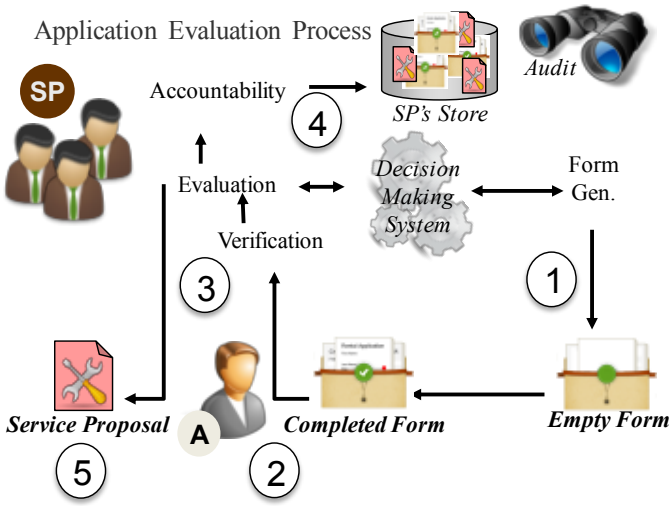


Figure 9. *Architecture classique de collecte de données par formulaire.*

4. Contributions [ANV12a]

Nos contributions se situent au niveau : (1) de l'architecture à collecte minimum à base de serveur personnel sécurisé ; (2) de la modélisation du problème passant par une représentation des règles de collecte, des données de l'utilisateur et de la métrique d'exposition permettant d'appliquer des algorithmes calculant le contenu minimum du formulaire pour le demandeur ; et (3) de la résolution du problème et de la validation de nos solutions sur des cas d'usage réels. Les sections suivantes résument chacune de ces contributions principales.

Architecture à collecte minimum. L'architecture à base de serveur personnel que nous considérons est présentée Figure 10. Deux étapes additionnelles sont introduites au procédé actuel, les autres étapes restant inchangées. D'abord, des règles de collecte sont construites de manière à refléter (tout ou partie) du processus de décision. Ces règles sont transmises au demandeur avec le formulaire d'application. La transmission des règles vers le serveur personnel du demandeur passe par un canal sécurisé classique (type SSH) pour éviter que le demandeur ne puisse accéder aux règles en clair. Ensuite, les règles de collecte et les données du demandeur sont confrontées dans l'enceinte sécurisée du serveur personnel (à laquelle n'a accès aucune des parties), en appliquant un algorithme de réduction de l'exposition des données à renseigner dans le formulaire.

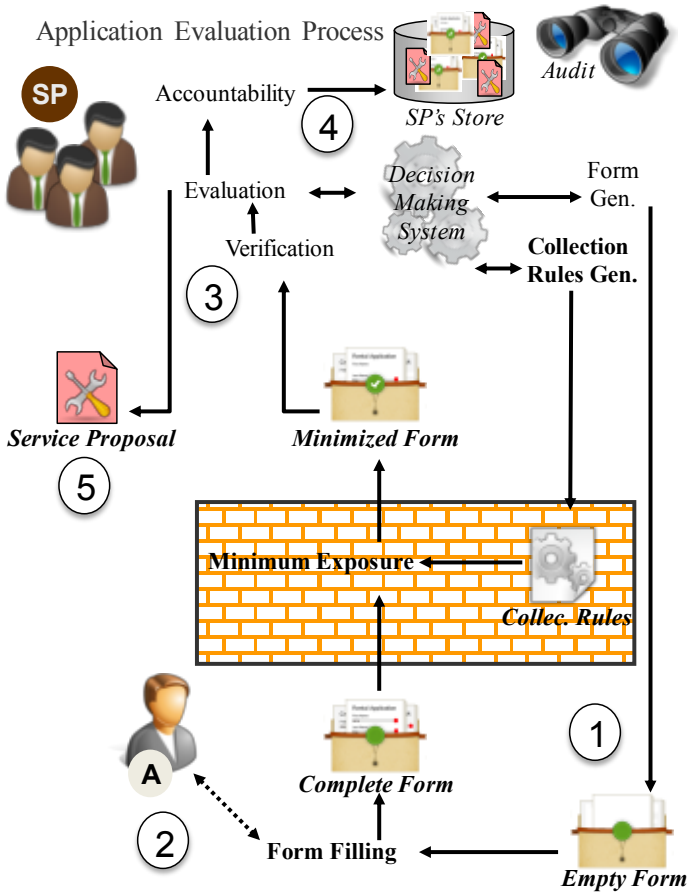


Figure 10. *Architecture à collecte minimum.*

Modélisation du problème. La modélisation du problème que nous proposons a juste pour but de démontrer l'applicabilité de l'approche dans certains cas d'usage réels. Nous considérons que les données personnelles d'un individu sont représentées par un ensemble de couples (attribut, valeur). Comme métrique d'exposition, nous considérons qu'une valeur est attribuée à chaque couple dénotant sa sensibilité[23], et nous calculons l'exposition d'un ensemble de couples en faisant la somme des valeurs de sensibilité des couples de l'ensemble[24]. Pour simplifier

[23] Une bonne métrique d'exposition de l'ensemble des couples présents dans un formulaire devrait permettre de satisfaire à la fois les exigences de l'usager (respect de sa vie privée) et l'objectif de l'organisme (réduire les coûts de traitement du formulaire), mais c'est probablement un problème de recherche en soi, dépassant l'objectif que nous nous donnons ici.

[24] Notons que certaines métriques existantes mesurant la perte d'information créées dans le cadre de travaux sur l'anonymisation de données, comme la *distorsion minimale* [XiT06] ou *ILoss* [YFA+08], peuvent elles aussi être supportées dans notre formalisme.

l'explication, nous considérerons dans la suite que tous les couples (attribut, valeur) ont une valeur de sensibilité de 1, ce qui revient à évaluer la sensibilité d'un formulaire comme étant le nombre d'attributs différents renseignés. Concernant les règles de collecte, nous exprimons chaque bénéfice potentiel sous la forme d'un ensemble de disjonctions de conjonctions de prédicats sur les attributs (forme normale disjonctive) permettant d'attribuer ce bénéfice au demandeur. Ce formalisme permet de représenter les décisions basées sur des règles logiques, des arbres de décision et des forêts d'arbres de décision, représentant ainsi de nombreux classifieurs binaires, multi-classes et multi-labels [TsK07]. Par exemple, la règle de collecte permettant d'obtenir une aide financière donnée en exemple en introduction du chapitre, peut s'écrire :

$aide\_financière$: ($pension < 30.000 \wedge age > 80$) $\vee$ $pension < 10.000 \vee GIR > 2$

Un ensemble de couples (attribut, valeur), lorsqu'ils sont exposés dans le formulaire, valident donc certaines règles de collecte, qui déterminent les bénéfices à attribuer au demandeur. Par exemple, le couple (*GIR*, 3) valide le prédicat $GIR > 2$ et donc la règle de collecte *aide_financière*. Le problème, dit de *n-exposition*, que nous considérons est ainsi le suivant : d'après un ensemble de règles de collecte R et un ensemble de données personnelles $P=\{(a,v)\}$, P est n-exposable pour R si et seulement s'il existe un sous ensemble de $m \leq n$ éléments dans P qui valide toutes les règles de R. Minimiser le formulaire revient à trouver l'ensemble n-exposable ayant la valeur minimum de n.

Complexité du problème. Le problème de *n-exposition* est NP-complet et le problème d'optimisation associé (trouver la valeur minimum de n) est NP-difficile. En effet, le problème du *MIN-SAT pondéré*, connu pour être NP-complet, est réductible au problème de *n-exposition*. Le problème du *MIN-SAT pondéré* est formulé comme suit dans [5] : soit un entier n, un ensemble $\{P_{j,k}\}$ de variables booléennes, une formule logique sous forme normale conjonctive $F = \wedge_j (\vee_k P_{j,k})$ sur $\{P_{j,k}\}$, et une fonction pondérée positive $w: \{P_{j,k}\} \rightarrow R^+$, trouver un assignement T de valeurs de vérité pour $\{P_{j,k}\}$ tel que F soit satisfaite et $w(T) = \Sigma_{j,k}\, w(P_{j,k}) \times T(P_{j,k})$ soit inférieure à n. Pour un individu donné, en ne considérant que l'ensemble R des règles de collecte satisfaites par les données de cet individu et réduites aux « conjonctions satisfaites », le problème de n-exposition peut être formulé de façon similaire. La satisfaction de l'ensemble R

des règles de collecte satisfaites pour l'individu est représenté par une formule logique $F' = \wedge_j (\vee_k (\wedge_l P_{j,k,l})$ où les $P_{j,k,l}$ sont des variables booléennes dont la valeur est *vraie* si le couple (attribut, valeur) validant le prédicat correspondant de la règle de collecte est exposé par l'individu, et *fausse* s'il n'est pas exposé. La fonction F, exprimée dans le cas du *MIN-SAT pondéré*, se ramène à la fonction F' de la n-exposition, en prenant l=1 dans F' (cas d'une fonction F' qui représenterait uniquement des règles de collecte purement disjonctives). Les résultats de complexité obtenus pour le *MIN-SAT pondéré* s'appliquent donc aussi à la n-exposition. Notamment, le problème du *MIN-SAT pondéré* est NP-complet et le problème d'optimisation relatif est NP-difficile [Co71, Ka72]. De plus, le problème d'optimisation du *MIN-SAT pondéré à valeurs positives* n'est pas dans la classe APX [AAG+98] (on ne peut pas trouver d'algorithme d'approximation de la solution fonctionnant en temps polynomial et dont l'approximation serait bornée par une constante). Ces résultats s'appliquent également à la n-exposition.

Algorithmes de résolution. Le problème de n-exposition, représenté sous forme logique, peut être résolu de manière exacte en utilisant un solveur de programmation en nombres entiers existant. Nous avons utilisé le solveur COUENNE [BLM+09], distribué en open source par le projet COIN-OR[25], qui est l'un des plus connus et efficaces parmi les solveurs gratuits adaptés à notre problème. Dans notre architecture, un tel solveur ne peut toutefois pas être envisagé car, d'une part il ne trouve la solution dans un temps acceptable que pour de petites instances du problème (formulaires de quelques dizaines de champs), et d'autre part il est beaucoup trop consommateur de ressources pour pouvoir être embarqué dans l'environnement contraint du serveur personnel sécurisé[26]. Nous avons donc proposé des algorithmes de résolution heuristiques embarqués dans le serveur personnel.

Utiliser le solveur COUENNE nécessite d'exprimer le problème d'optimisation de n-exposition sous forme d'un programme en nombres entiers. Nous avons choisi le langage de programmation AMLP [FGK02], un langage de modélisation algébrique de problèmes d'optimisation, supporté par COUENNE. Le problème de n-exposition exprimé sous forme de

[25] Voir http://www.coin-or.org/Couenne/

[26] Notons que faire tourner le solveur hors de l'enceinte sécurisée conduirait à divulguer les règles de collecte en clair hors du matériel sécurisé, ce qui serait contraire aux hypothèses que nous nous sommes fixées.

problème SAT peut être transposé en AMPL en utilisant une variable binaire par couple (attribut, valeur) impliqué, en exprimant une contrainte pour chaque règle de collecte, et en choisissant comme fonction objectif la somme des variables binaires du problème. Un exemple simplifié est donné Figure 11.

Règles de collectes :

r_1: $(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$ $\Rightarrow c_1$
r_2: $(p_5 \wedge p_6 \wedge p_7) \vee (p_4 \wedge p_8 \wedge p_9)$ $\Rightarrow c_2$
r_3: $(p_1 \wedge p_6 \wedge p_7) \vee (p_2 \wedge p_4 \wedge p_{10})$ $\Rightarrow c_3$
r_4: $(p_2 \wedge p_5 \wedge p_6 \wedge p_7) \vee (p_1 \wedge p_4 \wedge p_8 \wedge p_9)$ $\Rightarrow c_4$

***avec* :**

p_1: *pension* < 30.000, p_2: *age* > 80,
p_3: *tutelle* = 1, p_4: *GIR* > 2,
p_5: *vit_seul* = 1, p_6: *entourage* = 0,
p_7: *mobilité* = 0.5, p_8: *traitement* = 1,
p_9: *isolement* > 0.5, p_{10}: *plain_pied* = 1.

c_1=*aide_financière*, c_2=*assistante_journalière*, c_3=*adaptation_lmogement*, c_4=*portage_repas*.

Formulaire à renseigner :

Pension, *age*, *tutelle*, *GIR*, *vit_seul*, *entourage*, *mobilité*, *traitement*, *isolement*, *plain_pied*.

Données de l'utilisateur :

(*pension, 25.000*), (*age, 83*),
(*tutelle*, *1*), (*GIR*, *3*),
(*vit_seul*, *1*), (*entourage*, *0*),
(*mobilité*, *0.5*), (*traitement*, *1*),
(*isolement*, *1*), (*plain_pied*, *1*).

Programme AMPL :

```
var b1 binary; ... var b10 binary;
minimize EX:
b1+b2+b3+b4+b5+b6+b7+b8+b9+b10;
subject to
r1: b1*b2 + b3*b4 >= 1;
r2: b5*b6*b7 + b4*b8*b9 >= 1;
r3: b1*b6*b7 + b2*b4*b10 >= 1;
r4: b2*b5*b6*b7 + b1*b4*b8*b9 >= 1;
```

Solution minimale :

(*pension, 25.000*), (*age, 83*),
∅ ∅,
(*vit_seul*, *1*), (*entourage*, *0*),
(*mobilité*, *0.5*), ∅,
∅, ∅.

Figure 11. *Exemple de problème de n-exposition et formulation AMPL.*

Nous avons implanté plusieurs algorithmes donnant une solution approchée de façon adaptée aux contraintes du serveur personnel. L'algorithme appelé RAND* choisit plusieurs solutions de façon aléatoire et conserve la meilleure. Pour trouver une solution, cet algorithme tire au hasard pour chaque règle de collecte l'une des conjonctions de prédicats formant la règle, et expose les couples (attribut, valeur) validant les prédicats impliqués dans cette conjonction. La solution validant l'ensemble des règles est l'union des couples choisis pour chaque règle. Le procédé est répété, et la meilleure solution est conservée d'une exécution sur l'autre. D'autres algorithmes ont été testés, basés sur la méthode du recuit simulé ou sur des heuristiques adaptées aux scénarios réels que nous avons étudiés. L'heuristique qui donne globalement les meilleurs résultats sur les scénarios réels testés sélectionne pour chaque règle la ou les conjonctions contenant le moins de prédicats et dont la validation minimise le nombre de prédicats à valider dans les règles restant à traiter. La complexité de l'algorithme est supérieure

à celle de RAND*, mais pour faire une comparaison équitable nous avons calibré le nombre de solutions considérées par RAND* de manière à donner aux deux algorithmes le même temps d'exécution.

Résultats expérimentaux. L'objectif des mesures est de pouvoir valider l'approche et de quantifier les gains obtenus sur la réduction de l'exposition des données dans des scénarios réels. Le cas d'usage réel que nous avons le plus investigué concerne l'attribution d'aide sociale par les Conseils Généraux sur la base du formulaire GEVA. Divers bénéfices sont attribués au demandeur à partir de ce formulaire, et couvrent différentes dimensions: aide financière, médicale, para médicale, aide humaine (ménagère, repas, compagnie), adaptation du logement, etc. Nous avons modélisé les décisions sous forme de règles de collectes, en partenariat avec le Conseil Général des Yvelines. Les décisions se basent à la fois sur la régulation en vigueur, et sur des choix propres aux décisionnaires du département. Les critères de décision ne doivent pas être explicitement présentés aux usagers, pour éviter d'inciter à la fraude et parce que la décision finale d'attribution de l'aide reste souveraine. Nous avons identifié, en interaction avec les services du Conseil Général des Yvelines, 63 règles de collecte, impliquant 440 prédicats. Parallèlement, nous avons construit des classifieurs multi-labels sur deux jeux de données réels publics : ENRON (un ensemble d'emails devenus publics suite au scandale ENRON) et MEDICAL (rendu public par le département de radiologie d'un hôpital du Cincinnati). Ces jeux de données sont sans rapport avec le cas des formulaires d'application qui nous intéressent, mais les règles de collecte obtenues à partir de ces données suivent une topologie différente de celle que obtenue à partir du formulaire GEVA et nous permettent de tester les algorithmes sur d'autres topologies. Nous avons appliqué différentes techniques de résolution (exactes, heuristiques) et avons tiré les conclusions suivantes : (1) la réduction d'exposition est (presque toujours) conséquente avec les algorithmes heuristiques, apportant une réduction d'exposition des données renseignées dans le formulaire variant de 30% à 80% selon la topologie du problème ; (2) la résolution exacte n'est pas souvent possible (elle prend plusieurs heures dès que le nombre d'entrées dépasse 50 à 100 selon les topologies) ; (3) l'algorithme heuristique présenté ci-dessus, exécuté dans le microcontrôleur du serveur personnel, donne des résultat approchés satisfaisants, proches de ceux de COUENNE (quelques points d'écart sur le pourcentage de réduction, pour les instances sur lesquelles la résolution exacte est possible), et meilleurs que ceux obtenus avec l'algorithme aléatoire RAND* à temps d'exécution comparable.

5. Conclusion et résultats

Ces travaux nous permettent de démontrer l'intérêt du serveur personnel dans le contexte de la collecte minimum de données. Les règles de collecte et les données peuvent être confrontées dans l'enceinte sécurisée du microcontrôleur, préservant ainsi la confidentialité des règles et des données. Les algorithmes heuristiques embarqués présentent de bons résultats sur les cas réels, en termes de réduction d'exposition, et de temps d'exécution (moins d'une minute pour les instances les plus grandes). De nombreux processus de prise de décisions peuvent être couverts par le formalisme que nous proposons. Les résultats scientifiques, logiciels, et les actions de dissémination les plus marquants sont récapitulés ci-dessous.

Articles scientifiques. Nous avons introduit le concept d'exposition minimum comme interprétation stricte du principe légal de collecte limitée dans [ANV12a, ANV13]. Nous avons conduit une étude expérimentale utilisant des données réelles et des classifieurs multi-label pour démontrer l'applicabilité des techniques d'exposition minimum [ABN+15]. Une démonstration de techniques d'exposition minimum embarquées dans un composant à microcontrôleur, et leur application au cas de l'attribution d'aides sociales, a été présentée à EDBT'13 [ABN+13].

Logiciel. Le prototype MinExp-Card a été développé dans le cadre du projet ANR KISS, en lien avec le Conseil Général des Yvelines. Le prototype implante les techniques heuristiques d'exposition minimum sur une carte de développement STMicroelectronics STM32L152-EVAL (microcontrôleur ARM Cortex-M3, RISC 32 bit avec 16KB de RAM et 128KB de stockage persistant). Ce prototype démontre que ces techniques peuvent être portées sur des cartes à puce bon marché (quelques euros la carte) ou sur les serveurs personnels sécurisés considérés dans nos travaux.

Dissémination. Les travaux sur l'exposition minimum ont été présentés dans le cadre de Workshop [ANV11, ABN+12]. Nous avons aussi écrit un article dans le magazine « Tangente » [AnB14], destiné aux élèves de lycées, dans lequel nous présentons le problème de l'exposition minimum, et proposons au lecteur un défi sur un jeu de règles de collectes inspiré de celui construit dans le cas de l'aide sociale.

Un article de conclusion, identifiant certaines attaques à l'exposition minimum et proposant des contremesures, est actuellement en préparation. En effet, la connaissance des règles de collecte, de l'objectif de l'algorithme, ou de l'algorithme lui-même, peut conduire l'organisme destinataire du formulaire, ou un autre tiers, à inférer un certain nombre de données personnelles du demandeur non présentes dans le formulaire minimisé. Ces attaques peuvent être prises en compte. D'une part, les algorithmes proposés peuvent intégrer un calcul d'inférence qui complètera les formulaires minimisés avec les données inférées, afin de ne pas fausser le calcul de la mesure d'exposition. D'autre part, nous espérons pouvoir produire de nouveaux algorithmes de n-exposition, capables de considérer leur propre inférence à l'exécution, pour produire des résultats offrant une réduction plus importante.

Dans nos travaux futurs, nous chercherons à élargir le principe de l'exposition minimum au contexte du contrôle d'usage. Cette piste de recherche est décrite plus en détail dans la Section 3 du Chapitre « Conclusion et Perspectives ».

Chapitre 4
Application DMSP

Les sections précédentes mentionnent certaines des applications qui ont motivé les travaux de recherche présentés. L'application « Dossier Médico-Social Personnel » DMSP est particulièrement emblématique pour le projet SMIS, car elle repose sur l'architecture « Serveur de Données Personnel » et a été expérimentée sur le terrain. Elle donne un poids supplémentaire à nos arguments vers un « Web Personnel Sécurisé » auprès de la communauté non scientifique, industrielle et du grand public. Nous coordonnons le projet CG78/DMSP avec Philippe Pucheral, et j'en pilote les développements depuis 2007. Ce chapitre présente la motivation de l'application, l'état de l'art, l'approche suivie et les principaux résultats obtenus.

1. Motivation

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, aide-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 coordinations gérontologiques, etc. Il serait donc souhaitable que les données des dossiers patients puissent se synchroniser avec ces outils pré existants afin d'éviter les doubles saisies.

2. Etat de l'art

Concernant la gestion des dossiers médicaux, trois approches principales se distinguent. La première consiste à interconnecter des systèmes autonomes pré existants 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 témoigne 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 ce dossier pour chaque type d'intervenant, lui permettant de consigner les faits marquants survenus et devant être partagés 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[27] (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[28] présente le même type de

[27] Voir : http://www.arcan.fr/

[28] Voir : http://www.globule.net/fr/index.html

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 complet 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.

3. Approche

Notre approche s'inspire de l'architecture « Serveur Personnel de Données » (voir Chapitre 1, Section 4.1). Elle consiste à équiper chaque patient d'un SPD, embarquant son dossier médico-social, capable d'authentifier chacun des intervenants et de lui donner accès à la vue du dossier correspondant à sa pratique, et de se synchroniser sans connexion internet avec un serveur distant permettant la sauvegarde du dossier, et sa synchronisation avec différents logiciels ou chaines 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), qui convoient des paquets chiffrés entre le SPD et le serveur central accessible en zone connectée. Pour permettre l'interopérabilité avec des logiciels ou chaines 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 reconnaitre et savoir intégrer, à l'image de ce qui se pratique aux Etats-Unis dans le cadre de l'initiative « Blue Button[29] ».

Le patient reste maitre de ses données, et centralise 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. 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. Nous avons réalisé une expérimentation terrain, réalisé de nombreuses démonstrations, disséminé ces résultats auprès d'industriels et de juristes spécialisés dans la gestion de données personnelles de santé. L'application DMSP sert aussi de cas d'usage à plusieurs de nos articles scientifiques.

[29] 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

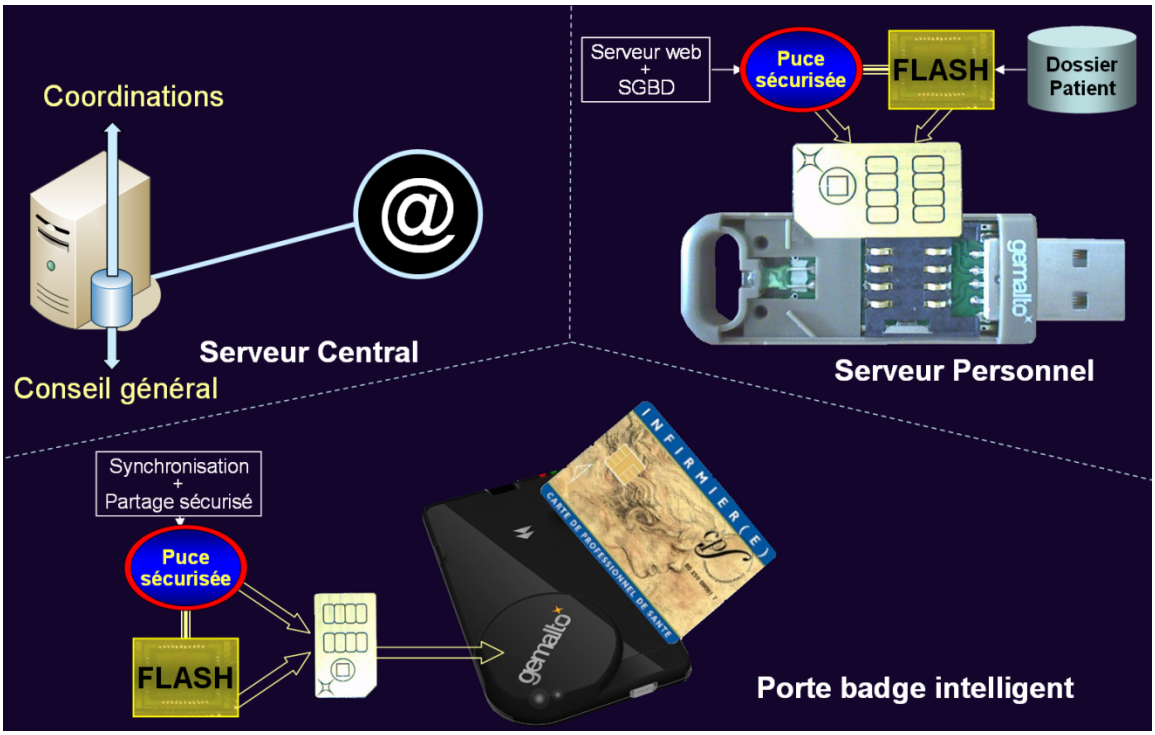


Figure 12. *Architecture initiale de la plate-forme PlugDB*

Plateforme logicielle et matérielle initiale. La plateforme initiale est basés sur trois éléments (voir Figure 12) : (1) un Serveur Personnel (SPD) intégré dans une carte SIM de nouvelle génération, sur laquelle est superposée une mémoire de type NAND Flash (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 chaines de traitement et de régénérer le contenu du dossier en cas de perte du serveur personnel. Nos contributions logicielles à cette plateforme se sont portées sur le développement du SGBD relationnel embarqué, la conception de modules annexes permettant de pré-compiler des requêtes SQL pour les applications, les spécifications et l'implémentation du processus de synchronisation SPD/Serveur central, la conception du pilote JDBC embarqué (il n'y a pas de standard JDBC pour Javacard, le sous-ensemble de Java pour l'embarqué), et son optimisation pour le système d'exploitation CIGALE (OS Java natif, JVM embarquée) de Gemalto. Nous avons développé cette plateforme avec les ingénieurs, doctorants

et chercheurs de l'équipe SMIS de 2007 à 2010, et en interaction avec des ingénieurs de chez Santeos (Amaury Willemand et Solène Martin) et de chez Gemalto (Jean-Jacques Vandewalle, Laurent Lagocento, Olivier Potonier, Patrick Enjolras et Laurent Boulard).

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 eePC) capables de jouer l'application DMSP depuis un navigateur Web, s'interfaçant indifféremment avec le serveur central ou le SPD. Un bilan de l'usage de l'application a été réalisé par Clarisse Chalard, étudiante en Sciences Sociales à l'UVSQ. Les conclusions étaient prometteuses (amélioration de la communication entre praticiens et autres métiers, évolution des pratiques de soins, suppression de la double-saisie), mais se heurtaient 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 eePC compliquant le déploiement et ayant généré des incidents techniques répétés).

Plateforme logicielle et matérielle actuelle. Nous avons élaboré une deuxième plateforme, corrigeant les défauts de la première. D'abord, nous avons substitué le SPD de Gemalto par un composant matériel de fonctionnalité équivalente, réalisable par toute PME spécialisées en électronique. Le composant a été conçu sous notre direction par la société Zed Electronics[30]. L'objectif était de maîtriser l'OS (nous utilisons maintenant FreeRTOS[31]) et la technologie hardware pour être en mesure de la faire évoluer en fonction des besoins de l'application plutôt que d'utiliser la solution fermée de Gemalto. Nous avons porté notre moteur de gestion de données sur ce nouveau matériel, ce qui nous a conduit à modifier les structures de données (buffers, journaux transactionnels) stockées en mémoire NOR (nouvelle technologie de NOR), à s'interfacer avec un nouvel OS, et à intégrer des fichiers pilotes adaptés à nos périphériques. Nous avons ainsi pu rendre le SPD complètement Plug-and-Play sur tablettes et smartphone

[30] ZED est une société d'électronique spécialisée dans la conception de prototypes et dans la fabrication de petites séries. Voir http://www.zeus.fr/

[31] Système d'exploitation embarqué et open source pour microcontrôleur le plus utilisé du marché. Voir http://www.freertos.org/

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. Un microphone pourrait aussi être intégré sans impact sur le coût du dispositif.

EBAC, un modèle de droits d'accès basé sur des évènements sémantiques. Cette étude adresse le problème du recueil du consentement d'un usager sur une politique d'accès, dans le domaine de la santé. Les données de santé sont régies par des politiques d'accès si complexes qu'il est inenvisageable d'obtenir un consentement éclairé des patients, comme cela est requis par la loi. La difficulté réside à la fois dans le très grand nombre de règles mises en œuvre et dans la complexité intrinsèque des données médicales. Nous avons proposé le modèle EBAC (« *Event-Based Access Control* ») pour aider le patient à réguler les données sensibles de son dossier en partant d'une sémantique connue. Ainsi, les données sont regroupées sous forme d'épisodes ou d'événements (« avortement », « dépression nerveuse 2012 »). Le patient choisit les intervenants médicaux qui peuvent ou non jouer un rôle sur ces événements, et précise les modalités d'échange d'informations entre ces intervenants. EBAC est appliqué comme un modèle de masquage dont les règles ont priorité sur celles définies dans la politique de contrôle d'accès générale. Les bases du modèle EBAC ont été publiées dans des revues et chapitres de livres [AAB+09, AAB+10a, AAB+10b]. Une version simplifiée du modèle a été utilisée pour l'expérimentation terrain de l'application DMSP.

Articles scientifiques. Outre les publications [AAB+09, AAB+10a, AAB+10b] liées au modèle EBAC, certaines publications scientifiques décrivent l'application DMSP [ABB+08a, ABB+08b].

Dissémination. La plateforme initiale a été démontrée dans une douzaine d'événements nationaux et internationaux dont Javaone [AnV09] (15000 participants) et Futur en Seine [Anc13a]. Nos solutions ont fait l'objet d'une audition publique à l'Assemblée Nationale en 2009 de Philippe Pucheral, responsable du projet SMIS, dans le cadre de la relance du dossier médical personnel (DMP) national. L'expérimentation terrain conduite dans les Yvelines a fait l'objet de nombreuses brèves publiées par le Conseil Général. De plus, le cas d'usage DMSP a servi de base à de nombreuses discussions entre juristes et informaticiens lors du projet multi

disciplinaire DEMOTIS, visant à étudier les compromis techniques et juridiques sous-jacents à la conception des infrastructures en charge du Dossier Médical Personnalisé (DMP).

5. Conclusion

L'application DMSP est très appréciée des professionnels de santé et des acteurs sociaux. Elle répond bien à leurs besoins, et présente des garanties de sécurité inégalées permettant de garantir le respect de la vie privée des patients. La solution est implantable sur plateforme générique et n'importe quel assembleur de composant électronique peut la fabriquer pour quelques dizaines d'euros. L'application arrive à un niveau de maturité qui nous permet d'envisager une industrialisation. Un audit de la solution est actuellement conduit par l'ARS île de France, afin d'étudier la possibilité d'une expérimentation plus large, préfigurant une industrialisation. Les enjeux du domaine de la santé sont particulièrement complexes, et l'avenir de DMSP dépend de nombreux facteurs que nous ne maitrisons pas. Cependant, dans une réponse[32] publiée dans le journal Officiel du Sénat daté du 21/08/2014, Marisol Touraine, Ministre de la Santé, constate que seuls 473 493 dossiers DMP ont été créés en France, la plupart étant vides ou ne contenant qu'un seul document, et a décidé de « *recentrer le DMP, renommé dossier médical partagé, sur les patients atteints de maladies chroniques ainsi que sur les personnes âgées, en particulier dans le cadre des expérimentations personnes âgées en risque de perte d'autonomie (PAERPA), qui justifient prioritairement d'une prise en charge pluriprofessionnelle coordonnée* ». Cette décision, nous l'espérons, nous aidera à envisager des expérimentations plus larges, dans lequel l'outil DMSP pourrait être vu comme facilitateur dans l'alimentation du DMP.

Enfin, l'expérience accumulée dans le cadre de DMSP nous a permis de mettre au point un nouveau composant dont nous maitrisons le matériel et le système d'exploitation. Saliha Lalalli et Cuong To, doctorants dans l'équipe, réalisent actuellement leurs développements et validations sur ce composant. Athanassia Katsouraki, elle aussi doctorante dans l'équipe, monte actuellement une expérimentation sur l'acceptabilité de ce type de composant et sur l'impact en

[32] http://www.senat.fr/basile/visio.do?id=qSEQ120901761&idtable=q289377|q259785|q263126|q258878&_c=%22paerpa%22&rch=qs&de=20110826&au=20140826&dp=3+ans&radio=dp&aff=sep&tri=p&off=0&afd=ppr&afd=ppl&afd=pjl&afd=cvn

termes de perception de sécurité des utilisateurs, en collaboration avec des chercheurs en économie expérimentale du laboratoire ADIS dans le cadre du projet ISN. Tout ceci découle en grande partie de notre implication collective dans l'élaboration de cette application.

Conclusion et perspectives

Cette section conclue le document et présente certains des éléments de mon programme de recherche, les plus en ligne avec les travaux présentés dans les chapitres précédents.

1. Conclusion générale

Les approches centralisées de gestion des données personnelles posent vis-à-vis du respect de la vie privée des problèmes intrinsèques, liés à un très faible ratio coût/bénéfice des attaques perpétrées, et au modèle sous-jacent basé sur une délégation de la gestion des données, et conduisant à des usages hors de contrôle du propriétaire des données, voire indésirables.

Dans ce document, nous avons présenté différentes architectures, alternatives ou complémentaires au modèle du Web actuel, qui dans différents contextes permettent de garantir une gestion de données plus respectueuse de la vie privée. Notre approche repose sur l'introduction d'un Serveur Personnel de Données (SPD) sécurisé matériellement, permettant à l'individu d'exercer un contrôle sur ses données personnelles, leur partage et leur dissémination, avec un niveau de garantie inégalé, hérité de la sécurisé physique offerte par le SPD. Ces différentes architectures permettent d'envisager l'émergence d'un nouveau modèle, que nous appelons le « Web Personnel Sécurisé ».

L'étude de ces architectures nous a conduit à proposer différentes contributions scientifiques relatives à leur mise en œuvre. Deux d'entre elles sont présentées plus en détail dans ce manuscrit. La première concerne la conception d'un SGBD embarqué, compatible avec les fortes contraintes du dispositif. Le design du SGBD se base sur des structures d'indexation massives permettant d'évaluer efficacement les requêtes avec très peu de RAM, générées séquentiellement pour éviter les mises à jour aléatoires en mémoire NAND Flash, et réorganisées de façon itérative pour supporter de grands volumes de données. La seconde contribution présentée est centrée sur le problème de la minimisation de l'exposition de données transmises à des tiers, dans le cas d'interactions avec des services en ligne au travers de formulaires. D'autres contributions parfois évoquées dans ce manuscrit n'y ont pas été présentées. Elles concernent par exemple la dégradation progressive des données personnelles [HFA09, ABH+08a, ABH+08b, ABH+08c, HAF+06], l'exécution de requêtes croisant des données privées embarquées avec des données externes publiques sans fuite d'information

privée [AAB+07, SAB+07, AAB+09] ou de nouveaux modèles permettant aux individus de réguler le partage de leurs données personnelles [AAB+10b, AAB+09, AAB+10c]. Les bénéfices potentiels de notre approche ont été illustrés au travers de l'application DMSP.

Mon programme de recherche des prochaines années vise à contribuer à l'établissement d'un « Web personnel sécurisé ». Pour disséminer cette vision, nous envisageons de proposer une version en logiciel libre et matériel libre de nos solutions, permettant un usage et un développement communautaire. La dissémination de cette plateforme commence par une mise à disposition auprès des étudiants de différentes universités et écoles d'ingénieurs. Nous travaillons aussi en lien avec la startup CozyCloud qui propose une plateforme libre de « Web personnel », afin d'y intégrer de plus fortes garanties grâce à nos technologies.

La suite de ce chapitre présente certaines perspectives de recherches, les plus en lien avec notre vision du « Web personnel sécurisé ». Ces perspectives sont structurées en trois axes : (1) l'embarquement de nouvelles fonctionnalités de gestion de données embarqué sous-jacentes au partage sécurisé des données, (2) l'étude de modèles de contrôle d'usage permettant de sécuriser les traitements effectués sur les données personnelles et qui ne peuvent être embarqués, et (3) la gestion de données personnelles en l'absence d'infrastructure. Ces trois axes sont nécessaires et complémentaires en vue de la réalisation d'un « Web personnel sécurisé », le troisième axe étant particulièrement pertinent dans le cas des Pays les Moins Avancés.

2. Gestion de données embarquées pour sécuriser les contrôles d'accès

Dans le contexte du Web personnel sécurisé, garantir le partage des données suivant des règles d'accès établies nécessite d'évaluer dans le composant sécurisé certaines opérations de gestion de données. Par exemple, pour une base de données relationnelle, c'est le calcul des vues autorisées qui doit être embarqué. Pour d'autres modèles de données (documents, clé valeur, etc.) et d'autres modèles de contrôle d'accès (expressions portant sur des tags, modèles de contrôle d'accès basés attributs, etc.), il est nécessaire de pouvoir embarquer d'autres techniques de gestion de données. Nous envisageons de définir des méthodes de conceptions permettant une transposition systématique dans l'environnement embarqué des techniques de gestion de données sous-jacentes à l'évaluation de règles d'accès. Nous illustrons ici cette problématique sur le cas de la gestion de collections de documents accessibles au travers de fonctions de recherche d'information sur le contenu.

Motivation. Dans le contexte du Web personnel sécurisé, les documents relatifs à un individu (mails, fichiers divers, administratifs, médicaux, bancaires, etc.) sont régulés depuis le serveur personnel. Des droits sont donnés aux applications en fonction des tags apposés sur les fichiers. Certaines applications (gestion des mails, gestion de fichiers, etc.) nécessitent un accès aux fichiers au travers d'un moteur de recherche d'information. Le moteur de recherche doit être embarqué pour éviter d'exposer hors de l'enceinte sécurisée l'index inversé permettant de réaliser les recherches (il contient des informations sur le contenu de chaque document indexé). Lors d'une recherche, seuls les documents qualifiés par la recherche et compatibles avec les règles d'accès établies doivent être produits en résultat.

Les techniques de recherche d'information pour l'embarqué se justifient aussi dans d'autres contextes. En effet, de nombreux objets intelligents de notre environnement sont maintenant dotés de capacités (locales) de collecte, de stockage et de traitements de données. Ils offrent des interfaces de recherche d'information pour accéder aux données [YGM08]. Certains objets domotiques maintiennent une description de leur environnement [YSM+08], par exemple, certaines librairies offrent la possibilité aux lecteurs d'interroger directement les rayons pour retrouver les livres les plus pertinents. D'autres exemples justifient la conception de moteurs de recherche embarqués dans des capteurs pour retrouver les objets pertinents de leur environnement ou dans des appareils photo (ou capteurs photos) pour rechercher des images sur des tags.

Tous cela justifie la conception d'un moteur de recherche, pouvant être vu comme une généralisation de Google Desktop ou de Spotlight, pour l'embarqué. Dans le contexte du Web personnel sécurisé, ce moteur devra être capable de traiter de grandes collections de documents et d'évaluer des droits d'accès basés sur des tags associés à ces documents.

Etat de l'art. Les moteurs de recherche ont été largement étudiés dans la littérature (voir [ZoM06] pour un état de l'art). Ils se basent sur des index inversés, et sur une fonction de classement (par exemple basée sur une métrique de type *tf-idf*[33]), pour retrouver les documents les plus pertinents pour une recherche. Chaque document est associé à un ensemble de termes (décrivant le contenu du document) pondérés (caractérisant l'importance du terme dans le

[33] « Term Frequency-Inverse Document Frequency ». Métrique traditionnelle en recherche d'information permettant de classer des documents selon leur pertinence, en fonction de la fréquence des termes de la recherche dans le document, et de l'inverse de la fréquence de ces termes dans la collection complète.

document). Pour les documents textuels, les termes sont les mots composant le document, et leur poids est la fréquence du mot dans le document. Un index inversé, organisé sous forme d'arbre, associe à chaque terme la liste des documents contenant ce terme et le poids du terme dans le document. Les requêtes sont évaluées de la manière suivante : (1) l'index inversé est accédé pour chaque terme de la requête, et renvoie les listes de couples (identifiant du document, poids) correspondantes, (2) un container est alloué en RAM pour chaque identifiant de document retourné par l'index et stocke les poids correspondants pour chaque terme de la recherche, (3) le score *tf-idf* est calculé pour chaque document/container, puis (4) les documents sont triés par score et les *k* plus grands sont produits en résultat. Cette stratégie ne peut pas être appliquée dans le contexte embarqué car, d'une part, l'index inversé ne peut pas être maintenu en Flash NAND efficacement sous forme arborescente (le temps d'insertion de nouveaux documents serait inacceptable), et d'autre part, la quantité de RAM disponible est très insuffisante pour permettre l'allocation d'un container par document.

Certains travaux de recherche se sont intéressés à concevoir des techniques de recherche d'information pour l'embarqué [TSW+08, TSW+10, WTL10, YGM08]. Elles sont basées sur une organisation séquentielle de l'index inversé, mais ne fonctionnent qu'avec un petit nombre de documents (quelques centaines) et ne passent pas à l'échelle (la performance des recherches est linéaire avec le nombre de documents indexés). D'autre part, elles ne permettent pas de supporter la suppression de documents.

Approche. La difficulté du problème vient des contraintes conflictuelles entre une RAM très limitée et une mémoire Flash NAND supportant très mal les petites mises à jour aléatoires. Notre approche se fonde sur trois principes de conception: (1) l'index inversé doit être composé de partitions écrites séquentiellement et jamais modifiées, (2) les requêtes doivent être évaluées sur les différentes partitions dans une RAM bornée avec un coût du même ordre de magnitude que le coût équivalent sur une structure optimale (ex. un index inversé sous forme d'arbre), et (3) pour limiter le nombre total de partitions celles-ci doivent être fusionnées en RAM bornée sans générer (trop) d'écritures aléatoires. Nous cherchons à définir à partir de ces principes un nouvel index inversé adapté à la recherche d'information sur de grands volumes de données (centaines de milliers de documents) et implantant des contrôles d'accès. Dans un deuxième temps, nous chercherons à affiner les principes de conception et à couvrir les bases de données clé/valeur et les modèles de contrôle d'accès associés. Plus généralement, nous essaierons de délimiter le champ d'application de nos principes de design de manière à permettre une

transposition systématique dans l'environnement embarqué des techniques de gestion de données sous-jacentes à l'évaluation de règles d'accès.

3. Contrôle d'usage

Certains traitements orientés données complexes ne peuvent être embarqués dans l'enceinte sécurisée d'un serveur personnel, car ils consomment beaucoup trop de ressources (calcul de profil, de recommandations, etc.). Mais les exécuter hors du serveur personnel nécessite d'externaliser de grands volumes de données personnelles nécessaires au traitement, ce qui peut remettre en cause la confidentialité des données. Il est donc nécessaire d'étudier et de proposer des techniques permettant de contrôler l'usage que certaines applications font des données, au risque de rendre l'approche Web personnel sécurisé caduque pour les traitements de données complexes. Cette étude consiste à étendre la sphère de sécurité, qui se limite essentiellement jusqu'ici à garantir des règles de contrôles d'accès, pour garantir du contrôle d'usage[34]. Par contrôle d'usage, nous désignons la possibilité de contrôler ce qu'une application fait des données en limitant les effets de bord de cette application, c'est-à-dire toute exploitation ou fuite de données qui pourrait conduire à un usage indésirable des données.

Nous avons choisi d'étudier ce problème dans le contexte concret et particulièrement représentatif de la gestion de données issues de compteurs électriques intelligents. Les données de consommation électriques produites à domicile nécessitent en effet des traitements complexes, par exemple de désagrégation, pour être exploitées. Nous illustrons ici la problématique du contrôle d'usage sur ce cas particulier.

Motivation. Dans les prochaines années, de nombreux foyers en Europe seront équipés de compteurs intelligents. Ces compteurs produisent une trace détaillée de la consommation électrique d'un logement. L'objectif de cette technologie est d'une part de permettre aux distributeurs d'énergie de mieux observer le réseau pour faire de l'équilibrage de charge, et d'autre part de permettre l'éclosion de nouveaux services pour les usagers, basés sur l'exploitation de leurs données de consommation d'énergie (conseils conduisant à des

[34] Notons que le terme contrôle d'usage est parfois utilisé pour désigner des modèles de contrôle d'accès sophistiqués, prenant par exemple en compte le contexte de la requête (heure, type de machine cliente, adresse IP, etc.). Il n'est pas entendu ici dans ce sens.

économies d'énergie, diagnostic en cas de panne d'un appareil électrique, jeux conduisant à faire baisser sa consommation, etc.). Le risque d'atteinte à la vie privée est réel puisque la signature énergétique révèle l'activité précise des habitants du logement (ex. [MWB11]). La CNIL Européenne signale aussi ces dangers[35]. Pour pouvoir utiliser ces données tout en limitant les risques, il faut offrir aux développeurs d'applications et aux usagers une plateforme permettant de fournir des garanties quant à l'usage effectif que les applications font des données.

Etat de l'art. L'usage par le propriétaire des données issues de son compteur intelligent n'est pas régulé, et permet à tout citoyen Européen d'utiliser des applications en ligne offrant un certains services sur la base de ces données. Aux Etats-Unis, l'initiative « Green Button[36] », à l'image du « Blue Button » pour les données de santé, permet aux individus de télécharger leurs données de consommation d'énergie auprès de leur fournisseur dans un format textuel compréhensible et documenté, utilisable par de nombreuses applications en ligne (notamment toutes les applications estampillées « Green Button »). Une telle exploitation des données n'offre pas de contrôle à l'usager sur la façon dont ses données sont utilisées ou disséminées.

Certaines solutions de l'état de l'art prennent en compte le problème du respect de la vie privée pour les données issues de compteurs intelligents. Certaines propositions visent à intégrer dans le compteur des fonctions capables de facturer l'usager en supportant différentes politiques tarifaires sans externaliser les données de consommation électrique. Notamment, des protocoles sécurisés ont été proposés pour couvrir les usages principaux du fournisseur d'électricité, sans exposer les données les données du compteur, tout en ayant une preuve d'authenticité du résultat [RiD11]. D'autres techniques permettent de réaliser des calculs d'agrégats provenant d'un ensemble de compteurs en se basant sur des techniques de chiffrement homomorphe (ex. [LLL10]). Ces travaux s'intéressent uniquement à des usages spécifiques des données de

[35] Voir le communiqué de presse du 11 juin 2012 du Contrôleur européen de la protection de données (EDPS), intitulé « Compteurs intelligents: le profilage des consommateurs permettra de suivre bien plus que leur consommation d'énergie si des limites claires ne sont pas établies ». Extrait du communiqué: l'introduction d'un compteur intelligent permet potentiellement de « *suivre ce que les membres d'un ménage font dans l'intimité de leurs maisons, s'ils sont en vacances ou au travail, si l'un d'eux utilise un dispositif médical spécifique ou un moniteur pour bébé, comment ils aiment passer leur temps libre, etc.* »

[36] Voir : http://www.nist.gov/smartgrid/greenbutton.cfm

compteurs, nécessaires aux autorités publiques ou aux organismes intervenant dans la distribution d'électricité et dans la facturation. Mais ils ne sont pas adaptés aux nouvelles applications personnelles possibles, en plein essor, permettant à l'usager d'analyser et de partager ses données, et d'améliorer sa propre consommation énergétique.

Approche. Les opérations à appliquer à la trace de consommation électrique d'un usager sont consommatrices de ressources, et brassent de très grands volumes de données (potentiellement toute la trace). C'est le cas de l'opération de désagrégation, qui apporte une sémantique aux données de consommation (ex. une vue des appareils électriques en fonctionnement). La désagrégation nécessite d'identifier la signature électrique de chaque appareil dans la trace, calcul qui ne peut pas être évalué dans l'environnement contraint du serveur embarqué. Il est pourtant nécessaire de supporter ce type d'opération, qui précède à tout usage applicatif des données. Notre objectif est donc de déporter ce type de calcul hors de l'enceinte sécurisée, tout en garantissant à l'individu que l'opération s'exécute sans effet de bord indésirable (c'est-à-dire sans fuite des données qu'elle manipule). Ce modèle de contrôle d'usage pourra ensuite être étendu aux autres applications orientées données.

Pour contrôler ces effets de bord, nous envisageons d'isoler l'environnement d'exécution de certains traitements (de certains périphériques, et des autres traitements applicatifs) grâce à des techniques de « sandboxing ». Cela nécessite d'implanter une infrastructure telle qu'illustrée sur la Figure 13. Le compteur alimente un serveur personnel sécurisé capable de communiquer (en Wifi, Bluetooth ou USB) avec l'infrastructure d'exécution des applications et d'appliquer des fonctions utilisateurs orientées données trop complexes pour être embarquées.

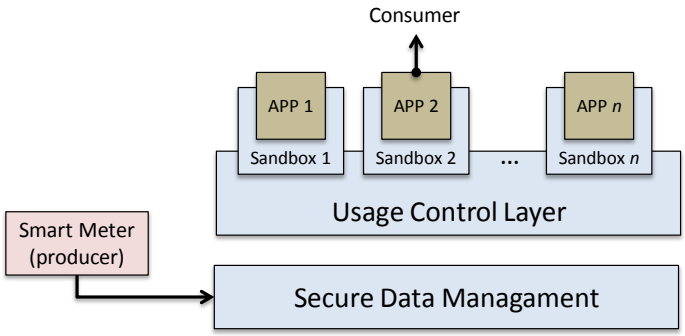


Figure 13. *Contrôle d'usage basé sur le principe du bac à sable (sandboxing).*

Tout traitement orienté données externe peut occasionner une fuite de données (effet de bord), soit via un canal de transmission auxiliaire (accès à un périphérique réseau, utilisation du système de fichier), soit via son résultat qui peut être non conforme et intégrer des informations supplémentaires. Pour éviter le premier type de fuite, nous envisageons de permettre un découpage des applications de manière à (1) isoler les traitements faisant des accès intensifs aux données (par exemple pour désagréger la trace énergétique) du reste de l'application ; (2) réguler le flux de données entre les compartiments isolés de l'application grâce au serveur personnel ; et (3) calibrer correctement les autorisations et les accès aux périphériques nécessaires à chacun des compartiments de l'application. L'isolation peut être obtenue grâce à Trustzone (voir Chapitre 1, Section 4.1) ou avec un hyperviseur classique (ex. Xen, pour lequel l'isolation peut être renforcée par du matériel sécurisé). Pour éviter le second type de fuite (via le résultat du calcul), des techniques de tests devront être définies (techniques collaboratives entre usagers, statistiques, à base de jeux d'entrées/sorties connues, etc.).

L'étude de cette perspective de recherche a tout juste démarré dans le cadre d'une collaboration avec Philippe Bonnet, Professeur à l'ITU au Danemark, qui a passé une année au sein du projet SMIS en 2013/2014.

4. Gestion de données sans infrastructure

D'après de nombreuses études [ITU11, CBP+12], les technologies de l'information constituent pour les Pays les Moins Avancés un élément facilitateur de développement économique pouvant améliorer la situation dans des domaines importants comme l'éducation, la santé ou la finance. Nous pensons qu'un serveur personnel de données permettrait d'offrir des services orientés données dans certains de ces différents domaines. Cependant, le contexte des Pays les Moins Avancés nous conduit à considérer une approche sans infrastructure (voir Chapitre 1, Section 4.3), très redondante et robuste par construction, ne reposant sur aucune autorité centrale pour fonctionner, où les communications sont asynchrones et opportunistes et où les coûts globaux du système sont directement proportionnels à la taille de la population visée. Les problèmes à résoudre pour établir un serveur personnel dans ce contexte concernent par exemple le déploiement d'applications, la modélisation de données unifiées, la vérification des identités, les règles de partage des données, les requêtes distribuées, tout cela sans reposer

sur aucune infrastructure centrale et avec une gestion opportuniste des communications. Le problème de l'authentification et du contrôle d'accès sans autorité centrale est détaillé ci-dessous à titre d'illustration.

Motivation. Les techniques de vérification des identités se basent sur des infrastructures à clé publique (PKI). Elles sont généralement centralisées, et reposent sur l'existence d'autorités de certification, qui sont les seules à être de confiance dans l'infrastructure. La vérification des identités s'organise alors sous forme hiérarchique à partir de ces autorités. Dans un contexte où le tissu économique, politique, associatif, la société civile, sont fortement représentés et bien structurés, les autorités de confiance peuvent être établies par consensus. C'est indirectement sur l'existence de toute cette structuration que repose la confiance. Nous considérons que ce modèle n'est pas transposable dans le contexte des PMA, qui se caractérisent par l'absence de structuration.

Etat de l'art. Certaines techniques de vérification des identités, alternatives et décentralisées, ont été proposées, comme PGP. Ces techniques reposent sur la notion de toile de confiance. Dans les approches basées sur le standard OpenPGP, chaque individu détient une clé publique, une clé privée, et un trousseau de clés permettant de stocker des clés et des signatures, et peut certifier l'identité d'un autre individu. Un individu A peut alors certifier l'identité d'un individu B par signature (A signe la clé publique de B avec sa clé privée). La vérification des identités repose sur le niveau de confiance que chacun attribue aux autres individus. Ainsi, si Alice fait confiance à Bob (dans le fait qu'il identifie correctement les individus qu'il certifie), les clés signées par Bob pourront être considérées comme valides par Alice. Des niveaux de confiance gradués peuvent aussi être attribués aux utilisateurs, et la validité d'une identité pourra alors être obtenue si suffisamment d'individus de niveau de confiance intermédiaire ont certifié cette identité. Ces techniques constituent un point de départ intéressant, mais elles ne peuvent pas être transposées directement dans notre contexte sans infrastructure. En effet, des serveurs sont nécessaires pour stocker et distribuer les clés et les signatures. L'approche suppose aussi qu'un nombre d'individus suffisamment important participe au système (des « signing parties » doivent être organisées au départ pour que les individus s'authentifient les uns les autres). Enfin, cette approche ne résout pas le problème de la propagation de règles d'accès.

Approche. Une caractéristique sous-jacente de l'architecture Folk-IS est de déporter les outils de contrôle aux extrémités du réseau, dans du matériel sécurisé, et nécessitant des interactions en face-à-face entre les individus. Cela peut permettre d'établir des sphères de confiance locales. Par exemple, une organisation locale (ex. une ONG de suivi sanitaire) pourrait produire des applications, un modèle de données, des informations d'identifications ad-hoc et des politiques de contrôle d'accès, et les pousser sur le réseau Folk-IS (l'injection des données pourrait être conduite localement, par des interactions physiques entre des individus et les acteurs de terrain de l'ONG). Nous espérons ainsi pouvoir jeter les bases d'un nouveau mode de gestion de données, semi-centralisé, dans lequel chaque *Folk-node* pourrait garantir (1) que les données produites par les acteurs d'une organisation donnée ne peuvent fuiter hors de l'organisation, et (2) que la vie privée des individus participant au système est toujours respectée.

Nous sommes actuellement en train de monter une proposition de projet européen répondant à l'appel H2020 ICT39[37], en collaboration avec notamment l'équipe IDASCO du LIRIMA au Cameroun.

[37] http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/9094-ict-39-2015.html

Bibliographie

[AAB+09] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Seamless Access to Healthcare Folders with Strong Privacy Guarantees. Special issue of the Journal of Healthcare Delivery Reform Initiatives, Vol. 1, n°4, pp. 82-107, 2009.

[AAB+10a] 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.

[AAB+10b] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Chap. IX: Trustworthiness of Pervasive Healthcare Folders. Book chapter of Pervasive and Smart Technologies for Healthcare: Ubiquitous Methodologies and Tools, A. Coronato, G. De Pietro (editors), Information Science Reference, pp. 1-24, 2010.

[AAB+10c] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Concilier ubiquité et sécurité des données médicales. Les Cahiers du CRID « Les technologies au service des droits, opportunités, défis, limites », D. Le Métayer (Editor), Vol. 32, Jan. 2010.

[AAG+98] Alimonti, P., Ausiello, G., Giovaniello, L., , and Protasi, M. On the Complexity of approximating weighted satis_ability problems. Tech. rep., Universit_a di Roma, 1998.

[ABB+07] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha: GhostDB: Querying Visible and Hidden Data without Leaks. 26th ACM International Conference on Management of Data (ACM SIGMOD), Beijing, China, June 2007.

[ABB+08a] N. Anciaux, M. Berthelot, L. Braconnier, L. Bouganim, M. De la Blache, G. Gardarin, P. Kesmarszky, S. Lartigue, J-F. Navarre, P. Pucheral, J-J. Vandewalle, K. Zeitouni. A Tamper-Resistant and Portable Healthcare Folder. International Journal of Telemedicine and Applications (IJTA), Vol. 2008, 9 pages, 2008.

[ABB+08b] N. Anciaux, M. Benzine, L. Bouganim, K. Jacquemin, P. Pucheral, S. Yin. Restoring the Patient Control over her Medical History. 21th IEEE Int. Symposium on Computer-Based Medical Systems (IEEE CBMS), Jyväskylä, Finland, June 2008.

[ABB+09] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha. Revelation on Demand. Distributed and Parallel Database Journal (DAPD), Vol. 25, n°1-2, pp. 5-28, 2009.

[ABB+13] 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.

[ABB+14] N. Anciaux, P. Bonnet, L. Bouganim, P. Pucheral. Trusted Cells: Ensuring Privacy for the Citizens of Smart Cities. ERCIM News, Vol. 98, 2014.

[ABD+13] N. Anciaux, L. Bouganim, T. Delot, S. Ilarri, L. Kloul, N. Mitton, P. Pucheral. Folk-IS: Opportunistic Data Services in Least Developed Countries. 36th International Conference on Very Large Data Bases (PVLDB), Vol. 7(5), Vision Paper, pp. 425-428, 2014.

[ABD+14] 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.

[ABG+10] N. Anciaux, L. Bouganim, Y. Guo, P. Pucheral, J.-J. Vandewalle, S. Yin. Pluggable Personal Data Servers. 29th ACM International Conference on Management of Data (ACM SIGMOD), demo. Paper, Indianapolis, Indiana, Jun. 2010.

[ABH+08a] N. Anciaux, L. Bouganim, H. van Heerde, P. Pucheral, P. M. G. Apers. Data Degradation: Making Private Data Less Sensitive Over Time. 17th ACM International Conference on Information and Knowledge Management (ACM CIKM), short paper, Napa Valley, USA, to appear, Oct. 2008.

[ABH+08b] N. Anciaux, L. Bouganim, H. van Heerde, P. Pucheral, P. M. G. Apers. InstantDB: Enforcing Timely Degradation of Sensitive Data. 24th International Conference on Data Engineering (ICDE), short paper, Cancun, Mexico, Apr. 2008.

[ABH+08c] N. Anciaux, L. Bouganim, H. van Heerde, P. Pucheral, P. M. G. Apers. Dégradation progressive et irréversible des données. 24èmes journées Bases de Données Avancées (BDA), Oct. 2008.

[ABN+12] N. Anciaux, D. Boutara, B. Nguyen, M; Vazirgiannis. Limiting Data Exposure in Multi-Label Classification Processes. In International Workshop on Privacy-AwaRe Intelligent Systems (PARIS2012), 2012.

[ABN+13] N. Anciaux, W. Bezza, B. Nguyen, M. Vazirgiannis. MinExp-Card : Limiting Data Collection Using a Smart Card. 16th International Conference on Extending Database Technology (EDBT), demo paper, pp. 753-756, 2013.

[ABN+15] N. Anciaux, D. Boutara, B. Nguyen, M. Vazirgiannis. Limiting Data Exposure in Multi-Label Classification Processes. Fundamenta Informaticae, to appear in 2015.

[ABP+01] N. Anciaux, C. Bobineau, L. Bouganim, P. Pucheral, P. Valduriez, ''PicoDBMS: Validation and Experience. 27th International Conference on Very Large Data Bases (VLDB), demo. paper, Roma, September 2001.

[ABP+07] N. Anciaux, L. Bouganim, P. Pucheral, 'Future Trends in Secure Chip Data Management', IEEE Data Engineering Bulletin (IEEE DEB), Vol. 30, n°3, 2007.

[ABP+08a] N. Anciaux, L. Bouganim, P. Pucheral, K. Jacquemin, S. Yin, D. Shasha, C. Salperwyck, M. Benzine. Software: PlugDB-engine version 1, registered at the 'Agence pour la Protection des Programmes (APP)' under the reference IDDN.FR.001.280004.000.S.C.2008.0000.10000, July 2008.

[ABP+08b] N. Anciaux, L. Bouganim, P. Pucheral, P. Valduriez. DiSC: Benchmarking Secure Chip DBMS. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), Vol. 20, n° 10, pp. 1363-1377, 2008.

[ABP+08c] N. Anciaux, L. Bouganim, P. Pucheral. SGBD embarqué dans une puce : retour d'expérience. Technique et Science Informatiques (TSI), Vol. 27, n°1-2, 2008.

[ABP+09] N. Anciaux, L. Bouganim, P. Pucheral, S. Yin, M. Benzine, K. Jacquemin, D. Shasha, C. Salperwyck, M. El Kholy. Software: PlugDB-engine version 2, registered at the 'Agence pour la Protection des Programmes (APP)' under the reference IDDN.FR.001.280004.000.S.C.2008.0000.10000, April 2009

[ABP+11] N. Anciaux, L. Bouganim, P. Pucheral, S. Yin, Yanli Guo, K. Jacquemin. Software: PlugDB-engine version 3, registered at the 'Agence pour la Protection des Programmes (APP)' under the reference IDDN.FR.001.280004.000.S.C.2008.0000.10000, Nov. 2011

[ABP+14] 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.

[ABP03a] N. Anciaux, L. Bouganim, P. Pucheral. Database Components on Chip. ERCIM News, Vol. 54, July 2003.

[ABP03b] N. Anciaux, L. Bouganim, P. Pucheral. Memory Requirements for Query Execution in Highly Constrained Devices. 29th International Conference on Very Large Data Bases (VLDB), Berlin, September 2003.

[ABP06] N. Anciaux, L. Bouganim, P. Pucheral. Data confidentiality: to which extent cryptography and secured hardware can help. Annals of telecom, Vol. 61, n°3-4, 2006.

[ABP09] N. Anciaux, L. Bouganim, P. Pucheral. A Hardware Approach for Trusted Access and Usage Control. Book chapter of the Handbook of Research on Secure Multimedia Distribution, S. Lian, Y. Zhang (editors), Information Science Reference, pp. 157-179, 2009.

[ADF+12] Ardagna, C.A., De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., and Samarati, P. Minimising Disclosure of Client Information in Credential-Based Interactions. Int. Journal of Information Privacy, Security and Integrity, 1(2), pp. 205-233, 2012.

[AGP10] Sécurité des bases de données. N. Anciaux, D. Gross-Amblard, P. Pucheral, R. Thion. Ecole de printemps « MASSES DE DONNEES DISTRIBUEES », Ecole de Physique de Houches, du 16 au 21 mai 2010.

[AGS+09] Agrawal, D., Ganesan, D., Sitaraman, R., Diao, Y. and Singh, S. Lazy-adaptive tree: An optimized index structure for flash devices. Proc. of the VLDB, 2(1):361-372, 2009.

[AHK+03] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. Enterprise privacy authorization language 1.2 (EPAL 1.2). W3C Member Submission, 2003.

[AKS+02] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu: Hippocratic Databases. International Conference on Very Large Data Bases (VLDB), pp. 143-154, 2002.

[AnB06] N. Anciaux, L. Bouganim. Data Management in Embedded Smart Devices. Tutorial donné à la Smart University, co-organisée avec la 7ème édition de la conférence internationale e-smart. Sept. 2006.

[AnB14] N. Anciaux, B. Nguyen. Limiter la collecte des données personnelles, un problème juridique NP-difficile. Magazine Tangente, numéro hors-série n°52 bibliothèque, Mathématiques & Informatique, 2014.

[Anc04] Thèse de doctorat de l'Université de Versailles St-Quentin-en-Yvelines, '*Systèmes de gestion de base de données embarqués dans une puce électronique*'. Déc. 2004. Rapporteurs: Patrick Valduriez (Directeur de Recherche, Inria), Michael J. Franklin (Professeur, Univ. Berkeley). Examinateurs : Philippe Bonnet (Professeur, DIKU), Jean-Claude Marchetaux (Ingénieur de Recherche, Gemalto). Directeur : Philippe Pucheral (Professeur, UVSQ). Co-encadrant : Luc Bouganim (Directeur de Recherche, Inira).

[Anc10] N. Anciaux. Dossier Médico-Social Portable et Sécurisé. Présentation et démonstration. Les Industries du Numérique pour la Santé, RII, in conjunction with the Connectathon, 2010, Cité Mondiale, Bordeaux.

[Anc11] N. Anciaux. Dossier Médico-Social Portable et Sécurisé. Présentation et démonstration. Les sciences du numérique au service de la santé à domicile et de l'autonomie, RII, 2010, Espaces Cap 15, Paris.

[Anc13a] N. Anciaux. Gestion de données personelles respetueuse de la vie privée. Presentation et démonstration, Futur en Seine, Archipelle des projets, 2013.

[Anc13b] N. Anciaux. Une nouvelle approche de la protection de nos données. Interview, MyScienceWork news, par Abby Tabor, 2013.

[Anc14] N. Anciaux. Vers un modèle de gestion des données respectueux de la vie privée : application à la collecte limitée d'informations personnelles. Seminaire IREP "BIG DATA", 2014.

[Anc15] N. Anciaux. Garantir la confidentialité des données personnelles. Futur en Seine 2014, Répondre aux défis des smart cities, 2014.

[And04] Anderson, A.H. An Introduction to the Web Services Policy Language (WSPL). In Proceedings of the POLICY Workshop, 2004.

[ANP13] N. Anciaux, B. Nguyen, I. S. Popa. Personal Data Management with Secure Hardware : The advantage of Keeping you Data at Hand. 14th International Conference on Mobile Data Management (MDM), Advanced Seminar, pp 1-2, 2013.

[ANP14] N. Anciaux, B. Nguyen, I. S. Popa. Tutorial: Managing Personal Data with Strong Privacy Guarantees. 17th International Conference on Extending Database Technology (EDBT), Tutorial, pp. 672-673, 2014.

[AnV09] Demonstration of electronic Health Records (eHR) on Java Card™ 3.0 Technology. Nicolas Anciaux (Inria) and Jean-Jacques Vandewalle (Gemalto). BOF-4576, CS Advanced Based Devices, JavaOne Conference, San Francisco, USA, Jun. 2009.

[ANV11] N. Anciaux, B. Nguyen, M. Vazirgiannis. Minimum Exposure - A New Approach for Limited Data Collection. Invited talk, Digiteo workshop on Web Mining, 2011, Telecom ParisTech, organized by M. Vazirgiannis and P. Senellart.

[ANV12a] N. Anciaux, B. Nguyen, M. Vazirgiannis. Limiting Data Collection in Application Forms : A real-case Application of a Founding Privacy Principle. 10th Conference on Privacy, Security and Trust (PST), 8p., 2012.

[ANV12b] N. Anciaux, B. Nguyen, M. Vazirgiannis. The Minimum Exposure Project: Limiting Data Collection in Online Forms. ERCIM News, Vol. 90, 2012.

[ANV13] N. Anciaux, B. Nguyen, M. Vazirgiannis M. Exposition minimum de données pour des applications à base de classifieurs. Ingénierie des Systèmes d'Information, Vol. 18, n°4, pp. 59-85, 2013.

[APP+12] N. Anciaux, J.M. Petit, P. Pucheral, K. Zeitouni. Personal Data Server: Keeping Sensitive Data under the Indiviual's Control. ERCIM News, Vol. 90, 2012.

[Arg03] Arge L., "The Buffer Tree: A Technique for Designing Batched External Data Structures", Algorithmica, 2003.

[BaS11] S. Bajaj, R. Sion: TrustedDB: a trusted hardware based database with privacy and data confidentiality. SIGMOD Conference 2011: 205-216

[BaS14] Bajaj, S., & Sion, R. (2014). TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality. Knowledge and Data Engineering, IEEE Transactions on, 26(3), 752-765.

[BBD14] G. Blank, G. Bolsover, E. Dubois. A New Privacy Paradox. Global Cyber Security Capacity Centre, Draft Working Paper, 2014.

[BLM+09] Belotti, P., Lee, J., Liberti, L., Margot, F., Wachter, A. Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods and Software 24, 4-5 (2009).

[BRD11] Bernstein P., Reid C., Das S., "Hyder - A Transactional Record Manager for Shared Flash," CIDR, 2011.

[BSS+03] Bolchini C., Salice F., Schreiber F., Tanca L., "Logical and Physical Design Issues for Smart Card Databases," TOIS, 2003.

[Cas13] D. Castro, D. How much will PRISM cost the US cloud computing industry. The Information Technology and Innovation Foundation. Jan. 2013.

[CBP+12] Coceres, R., Belding, E. M., Parikh, T. S., and Subramanian, L. 2012. Information and Communication Technologies for Development - Guest Editors' Introduction. IEEE Pervasive Computing, 11(3).

[CCK+05] Chen, W., Clarke, L., Kurose, J., and Towsley, D. Optimizing cost-sensitive trust-negotiation protocols. IEEE Computer and Communications Societies (INFOCOM), 2005.

[Cha12] S. Charney. Trustworthy Computing Next. Microsoft, white paper, 2012.

[CLM+02] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002.

[CoC13] P. Collin, N. Colin. Mission d'expertise sur la fiscalité de l'économie numérique. Ministère des Finances et de l'Economie. 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.

[Coo71] Cook, S. A. The complexity of theorem-proving procedures. In Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (1971).

[DDJ+03] E. Damiani, S. De Capitani Vimercati, S. Jajodia, S. Paraboschi, P. Samarati, 'Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs', *ACM Conference on Computer and Communications Security (CCS)*, 2003.

[DDP+02] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, 'A Fine-Grained Access Control System for XML Documents', *ACM Transactions on Information and System Security (ACM TISSEC)*, (5)2, 2002.

[DeS11] Debnath B., Sengupta S., Li J., "SkimpyStash: RAM Space Skimpy Key-Value Store on Flash," SIGMOD, 2011.

[DGM+07] Diao, Y., Ganesan, D., Mathur, G., and Shenoy, P. J. Rethinking data management for storage-centric sensor networks. In CIDR, pp. 22–31, 2007.

[Dir95] 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.

[EhJ07] J. H. Ehlers and S. A. Jassim. Wavelet library for constrained devices. volume 6579, pages 65790P–65790P–11, 2007.

[Eur85] European Directive 95/46/EC, 'Protection of individuals with regard the processing of personal data', Official Journal L 281, 1985.

[Euro08] Eurosmart. Smart USB token. White paper, Eurosmart, 2008.

[FGK02] Fourer, R., Gay, D. M., and Kernighan, B. W. AMPL : A Modeling Language for Mathematical Programming, second edition. Duxbury Press, 2002.

[GiD14] Giesecke & Devrient. StarSign® Mobile Security Card SE 1.2, Datasheet, 2014.

[GoB14] J. González and P. Bonnet. Towards an open framework leveraging a trusted execution environment. In Cyberspace Safety and Security. Springer, 2013

[HAF+06] H.J.W. van Heerde, N. Anciaux, L. Feng, P. Apers. Balancing Smartness and Privacy for the Ambient Intelligence. First European Conference on Smart Sensing and Context (EuroSSC), Lecture Notes in Computer Science 4272 springer 2006, Enschede, The Netherlands, Oct. 2006

[HCL+97] Haas L. M., Carey M. J., Livny M., Shukla A., "Seeking the truth about ad hoc join costs," VLDB Journal, 1997.

[HFA09] H. van Heerde, M. Fokkinga, N. Anciaux. A Framework to Balance Privacy and Data Usability Using Data Degradation. IEEE International Conference on Computational Science and Engineering (CSE), Los Alamitos, CA, USA, 2009.

[IBM03] IBM corporation, 'IBM Data Encryption for IMS and DB2 Databases v. 1.1', 2003. http://www-306.ibm.com/software/data/db2imstools/html/ibmdataencryp.html.

[ITU11] ITU. 2011. The Role of ICT in Advancing Growth in Least Developed Countries – Trends, Challenges and Opportunities. http://www.itu.int/pub/D-LDC-ICTLDC.2011

[Kar72] Karp, R. M. Reducibility among combinatorial problems. In Complexity of Computer Computations (1972), pp . 85-103.

[Kha11] F. Khatibloo. Personal Identity Management - Preparing For A World Of Consumer-Managed Data. Forrester Report, Sept. 30, 2011.

[KoV11] Koltsidas I., Viglas S. D., "Data management over flash memory," SIGMOD, 2011.

[LFA11] Lim H., Fan B., Andersen D., Kaminsky M., "SILT: a memory -efficient, high-performance key-value store", SOSP, 2011.

[LiR99] Li, Z., and Ross, K. A., "Fast joins using join indices", VLDB Journal, 1999.

[LLL10] Li, F., Luo, B., & Liu, P. (2010, October). Secure information aggregation for smart grids using homomorphic encryption. In Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on (pp. 327-332). IEEE.

[Mer90] R. Merkle, 'A Certified Digital Signature', *Advances in Cryptology (Crypto'89)*, LNCS, vol.435, Springer--Verlag, 1990.

[MHV+08] 'Les Yvelines, acteurs et partenaires de la Recherche et Développement' lors de la Convention d'Affaires 'Les RDV Carnot, Palais des Congrès, Versailles. Table ronde animée par : Christian Beley, Sous-Directeur Pôle économique CG78, Frédérick Becquet, Chargé de mission R&D CG78. Participants : Pr. Luc Montagnier, PDG Nanectis Biotechnologies, Yan Haentjens, PDG Vectrawave, Jean-Pierre Arragon, Directeur Portfolio Management Continental Automotive, et Nicolas Anciaux, Chargé de recherche à Inria Paris-Rocquencourt. Mars 2008.

[Mil14] Claire Cain Miller. Revelations of N.S.A. Spying Cost U.S. Tech Companies. The New York Times, 21 Mars 2014.

[MLC+13] M. Madden, A. Lenhart, S. Cortesi, U. Gasser. Teens and Mobile Apps Privacy. Pew Internet and American Life Project. Août 2013.

[MOP+00] Muth P., O'Neil P., Pick A., Weikum G., "The LHAM log-structured history data access method", VLDB Journal, 2000.

[Mos05] Moses, T. Extensible access control markup language (xacml) version 2.0. Oasis Standard, 2005.

[MSW+14] 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.

[MWB11] A. D. K. Mulligan, L. Wang, and A. J. Burstein, "Final Project Report Privacy in the Smart Grid: An Information Flow Analysis," CIEE Report, 2011.

[NTB+12] 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.

[OCG+96] O'Neil P., Cheng E., Gawlick D., O'Neil E., "The log-structured merge-tree (LSM-tree)", Acta Informatica, 1996.

[PBV+01] 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.

[Pri74] The Privacy Act, 5 U.S.C.§552a, 1974. http://www.usdoj.gov/04foia/privstat.htm

[Res14] 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.

[RiD11] Rial, A., & Danezis, G. (2011, October). Privacy-preserving smart metering. In Proceedings of the 10th annual ACM workshop on Privacy in the electronic society (pp. 49-60). ACM.

[RoO92] Rosenblum M., Ousterhout J., "The Design and Implementation of a Log-Structured File System", ACM TOCS, 1992.

[SAB+07] C. Salperwyck, N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha. GhostDB: Hiding Data from Prying Eyes. 33th International Conference on Very Large Data Bases (VLDB), demo. paper, Vienna, Austria, Sept. 2007.

[Sam01] Samarati, P. Protecting respondents' identities in microdata release. IEEE TKDE, 13(6), 2001.

[SeL76] Severance D., Lohman G., "Differential files: their application to the maintenance of large databases". ACM TODS, 1976

[SmR] Schmid P., Roos A., "SDXC/SDHC Memory Cards, Rounded Up And Benchmarked", http://tinyurl.com/tom-sdxc

[Sta13] J. Staten. The Cost of PRISM Will Be Larger Than ITIF Projects. Forrester blog. Août 2013.

[Sun99] Sundaresan P., "General Key Index", US. Patent n° 5870747, 1999.

[TNP14] To, Q.-C., Nguyen, B., and Pucheral, P.: Privacy-Preserving Query Execution using a Decentralized Architecture and Tamper Resistant Hardware. EDBT 2014: 487-498.

[TsK07] Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM), 3(3), 1-13.

[TSW+08] C. C. Tan, B. Sheng, H. Wang, and Q. Li. Microsearch: When search engines meet small devices. In Pervasive Computing, volume 5013 of Lecture Notes in Computer Science, pages 93–110. Springer Berlin Heidelberg, 2008.

[TSW+10] C. Tan, B. Sheng, H. Wang, and Q. Li. Microsearch: A search engine for embedded devices used in pervasive computing. ACM Trans. Embed. Comput. Syst., 9(4):43:1–43:29, Apr. 2010.

[VaA09] Demonstration of Electronic Health Records (EHR) on Java Card 3.0 Based Devices. Jean-Jacques Vandewalle, Research Engineer GEMALTO, Nicolas Anciaux, Researcher (Inria). Smart Event 10th Edition, World e-ID 2009, Sophia-Antipolis, sept. 2009.

[Val87] P. Valduriez, 'Join Indices', *ACM Transactions on Database Systems (ACM TODS)*, 12(2), 1987.

[Vin02] R. Vingralek, 'Gnatdb: A small-footprint, secure database system', *28th International Conference on Very Large Data Bases (VLDB)*, August 2002.

[VMS02] R. Vingralek, U. Maheshwari, W. Shapiro, 'TDB: A Database System for Digital Rights Management', *8th International Conference on Extending Database Technology (EDBT)*, March 2002.

[VWA+12] Vo, H. T., Wang, S., Agrawal, D., Chen, G., & Ooi, B. C. (2012). LogBase: a scalable log-structured database system in the cloud. Proceedings of the VLDB Endowment, 5(10), pp. 1004-1015, 2012.

[War10] Fading data could improve privacy. By M. Ward, BBC News. 16 June 2010. http://www.bbc.co.uk/news/10324209

[WCK03] Wu C., Chang L., Kuo T., "An Efficient B-Tree Layer for Flash-Memory Storage Systems," RTCSA, 2003.

[WEF11] The World Economic Forum. Personal Data: The Emergence of a New Asset Class. Nov. 2011.

[WEF12] The World Economic Forum. Rethinking Personal Data: Strengthening Trust. May 2012.

[Wei02] Weininger, A., "Efficient execution of joins in a star schema", SIGMOD, 2002.

[WTL10] H. Wang, C. C. Tan, and Q. Li. Snoogle: A search engine for pervasive environments. Parallel and Distributed Systems, IEEE Transactions on, 21(8):1188–1202, Aug 2010.

[XiT06] Xiao, X., and Tao, Y. Personalized privacy preservation. In Proceedings of ACM SIGMOD, 2006.

[YFA+08] Yao, D., Frikken, K.B., Atallah, M.J., and Tamassia, R. Private information: To reveal or not to reveal. In ACM TISSEC, 12(1), 2008

[YGM08] Yan, T., Ganesan, D., and Manmatha, R. Distributed image search in camera sensor networks. In Proc. of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys'08, pp. 155–168, 2008.

[YPM09] Yin S., Pucheral P., Meng X., "A Sequential Indexing Scheme for Flash-based embedded systems," EDBT, 2009.

[YSM+08] Yap, K.-K., Srinivasan, V., and Motani, M. Max: Wide area human-centric search of the physical world. ACM Transactions on Sensor Networks, 4(4):26:1-34, 2008.

[ZoM06] Zobel, J. and Moffat, A. Inverted files for text search engines. ACM Computing Survey, 38(2), 2006.

Annexe A.

Secure Personal Data Servers: a Vision Paper

Tristan Allard, Nicolas Anciaux, Luc Bouganim, Yanli Guo, Philippe Pucheral, Benjamin Nguyen, Lionel Le Folgoc, Indrajit Ray, Indrakshi Ray, Shaoyi Yin

Proceedings of the VLDB Endowment (PVLDB), Volume 3(1), pp. 25-35, 2010.

Secure Personal Data Servers: a Vision Paper

Tristan Allard[*,**], Nicolas Anciaux[*], Luc Bouganim[*], Yanli Guo[*], Lionel Le Folgoc[*], Benjamin Nguyen[*,**], Philippe Pucheral[*,**], Indrajit Ray[***], Indrakshi Ray[***], Shaoyi Yin[*]

* INRIA Paris-Rocquencourt
Le Chesnay, France
<Fname.Lname>@inria.fr

** PRISM Laboratory
Univ. of Versailles, France
<Fname.Lname>@prism.uvsq.fr

*** Colorado State University
Fort Collins, CO, USA
{indrajit,iray}@cs.colostate.edu

ABSTRACT

An increasing amount of personal data is automatically gathered and stored on servers by administrations, hospitals, insurance companies, etc. Citizen themselves often count on internet companies to store their data and make them reliable and highly available through the internet. However, these benefits must be weighed against privacy risks incurred by centralization. This paper suggests a radically different way of considering the management of personal data. It builds upon the emergence of new portable and secure devices combining the security of smart cards and the storage capacity of NAND Flash chips. By embedding a full-fledged Personal Data Server in such devices, user control of how her sensitive data is shared by others (by whom, for how long, according to which rule, for which purpose) can be fully reestablished and convincingly enforced. To give sense to this vision, Personal Data Servers must be able to interoperate with external servers and must provide traditional database services like durability, availability, query facilities, transactions. This paper proposes an initial design for the Personal Data Server approach, identifies the main technical challenges associated with it and sketches preliminary solutions. We expect that this paper will open exciting perspectives for future database research.

1. INTRODUCTION

The number of information systems continuously gathering personal data on servers is escalating at a tremendous pace. Electronic Health Record systems used today in most advanced countries, vehicle tracking systems used to compute insurance premium, and soon carbon tax, travelers tracking systems used by public transportation companies, systems implementing e-administration procedures (scholarship folders, identity cards, social security covers, pension funds, income taxes, …) are illustrative but not exclusive examples. Citizens have no way to opt-out of these applications because governments, public agencies or companies that regulate our daily life require them.

In the meantime, administrations and companies deliver an increasing amount of digitized personal data to the user (salary forms, insurance forms, invoices, phone call sheets, banking statements, etc). Primary copies of this data are kept by the information system that produced the data and secondary copies are delivered to the user for her personal use. While nothing dictates this, these secondary copies often also end up in servers for user's convenience. Indeed, the user expects her data to be resilient to failure, available through the internet 24/7 and easily manageable (i.e., organized, queryable, sharable). Many internet companies provide precisely this service to everyone, sometimes for free, and without requiring any computer expertise from the end user.

All these situations put together result in accumulating a complete digital history of citizens in servers. The benefits of centralizing personal data are unquestionable in terms of data completeness, failure resiliency, high availability and even consistency of security policies. But these benefits must be weighed carefully against the privacy risks of centralization. There are many examples of privacy violations arising from negligence, abusive use, internal attacks, external attacks, and even the most secured servers are not spared (see Annex B).

This paper draws a radically different vision of the management of personal data. This vision builds upon the emergence of new hardware devices called Secure Portable Tokens (SPT for short). Whatever their form factor (SIM card, secure USB stick, wireless secure dongle), SPTs combine tamper resistant smart card microcontrollers with large storage capacity NAND Flash chips. This unprecedented conjunction of portability, secure processing and Gigabytes-sized storage holds the promise of a real breakthrough in the secure management of personal data. The idea promoted in this paper is to embed, in such devices, software components capable of acquiring, storing and managing securely personal data. However, our approach does not amount to a simple secure repository of personal documents. The ambition is, first, to allow the development of new, powerful, user-centric applications and to serve data requests from existing server-based applications managing personal data, thus requiring a well organized, structured, consistent and queryable representation of these documents[1]. Second, we want to provide the user with a friendly control over the sharing conditions related to her data and with tangible guarantees about the enforcement of these conditions. These two objectives lead to the definition of a real secure and portable Personal Data Server (PDS for short). With appropriate infrastructure, PDSs enable the vision depicted by Figure 1. Bob's personal data, delivered by different sources, is sent to his PDS which can then serve data requests from private applications (serving Bob's interest), secure multi-actors applications (accessed through actors' PDS) and external applications. Bob's PDS can also take part in secure global processing.

What can be precisely expected from a PDS is the following:

- To provide the main functionalities of a database engine (data description and structuring, access control, query facilities, and transactions) to help developing user-centric applications. Embedding the database engine in the SPT ensures that only authorized data are delivered to the querier's terminal.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Articles from this volume were presented at The 36th International Conference on Very Large Data Bases, September 13-17, 2010, Singapore.
Proceedings of the VLDB Endowment, Vol. 3, No. 1
© 2010 VLDB Endowment 2150-8097/10/09... $10.00

[1] Simple document repositories, even if they integrate keyword search facilities cannot meet these requirements. The lack of data structure has been considered as one of the major reasons (with security) that explain the failure of the first French national EHR System [15].

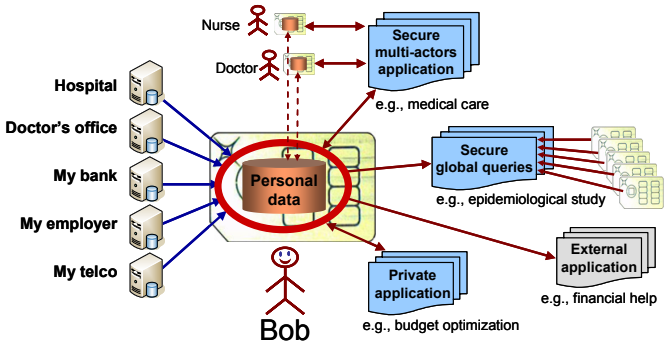


Figure 1. The Personal Data Server approach

- To be interoperable (1) with existing data sources to allow the acquisition (and rendering) of personal data, and (2) with other PDSs to allow secure data sharing protocols among them.
- To reestablish the control of the user on how her personal data is shared with others (what data, with whom, for how long, for which purpose). In other words, a PDS must give the ability to enforce privacy principles (e.g., consent, limited collection, limited retention, audit) [3] for all data it stores and for all data it accesses from other PDSs.
- To inherit the portability and tamper-resistance of the device embedding it, thereby providing disconnected facilities and an enforcement of security rules stronger, yet more flexible, than those of a traditional server.

Converting the PDS vision into reality introduces important challenges. First, the SPT, central element of the approach, exhibits strong hardware constraints. Traditional core database techniques (storage and indexing, query and transaction processing) need to be fully revisited to design an embedded database engine that provides acceptable performance. Second, to enforce security rules in a PDS-based information system, atypical distributed protocols combining a large number of highly secure but low power PDSs with a powerful but unsecured server infrastructure must be devised. Third, the PDS approach aims at helping every individual to better protect her privacy. The way to control how data is shared and protected must therefore be highly intuitive and simple. Our hope is that this paper will open various and exciting research directions for the database community.

The rest of the paper is organized as follows. Section 2 presents the characteristics of a SPT and derives from it the problem statement associated with the implementation of the PDS vision. Section 3 illustrates the PDS vision through different scenarios. We sketch out an initial design for the PDS architecture in Section 4, present a set of technical challenges in Sections 5 to 7 and provide concluding remarks in Section 8. Additional material is provided in Annex (preliminary prototypes prefiguring the PDS vision and protocols) to further convince the reader that the PDS vision is not pure utopia.

2. PROBLEM STATEMENT

We introduce the hardware characteristics of SPT, then present the hypothesis related to the security of PDSs and of the infrastructure surrounding them and finally state the problem related to the implementation of the PDS approach.

Hardware characteristics of SPTs: SPTs are emerging today in a wide variety of form factors ranging from SIM cards to various forms of pluggable secure tokens. Whatever the form factor, SPTs share several hardware commonalities. Their microcontroller is typically equipped with a 32 bit RISC processor (clocked at about 50 MHz today), memory modules composed of ROM, static RAM (about 64KB), a small internal stable storage (about 1MB of NOR Flash) and security modules providing the tamper-resistance. The microcontroller is connected by a bus to a large external mass storage (Gigabytes of NAND Flash). However, this mass storage does not benefit from the microcontroller tamper resistance. SPTs can communicate with the outside world through various standards (e.g., USB2.0, Bluetooth, 802.11). Figure 2 shows typical examples of SPTs but this paper makes no assumption on the form factor.

Hardware progresses are fairly slow in the secure chip domain because the size of the market (billions of units), and the requirement for high tamper-resistance leads to adopting cheap and proven technologies [13]. Nonetheless, SPT manufacturers forecast a regular increase of the CPU power, stable storage capacity and the support of high communication throughputs (up to 480 Mb/s). RAM will unfortunately remain a scarce resource in the foreseeable future due to its poor density. Indeed, the smaller the silicon die, the more difficult it is to snoop or tamper with its processing, but RAM competes with CPU, ROM and NOR in the same silicon die.

In summary, a SPT can be seen as a low power but very cheap (a few dollars), highly portable, highly secure computer with reasonable storage capacity for personal use.

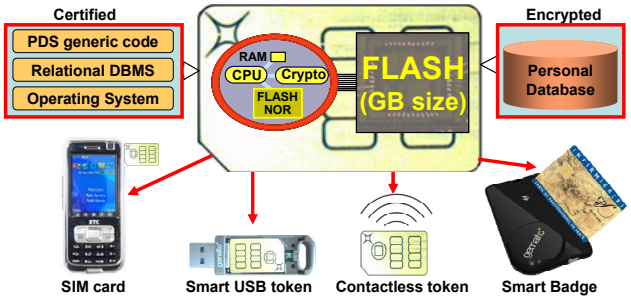


Figure 2: Secure Portable Token and embedded PDS

PDS security: the level of trust which can be put in the PDS comes from the following factors:

1. The PDS software inherits the tamper resistance of the SPT making hardware and side-channel attacks highly difficult.
2. The basic software (operating system, database engine and PDS generic tools), called hereafter *PDS core*, can be certified according to the Common Criteria, making software attacks also highly difficult.
3. The PDS core can be made auto-administered thanks to its simplicity, in contrast to its traditional multi-user server counterpart. Hence, DBA attacks are also precluded.
4. Compared to a traditional server, the ratio Cost/Benefit of an attack is increased by observations 1 and 2 and by the fact that a successful attack compromises only the data of a single individual.
5. Even the PDS holder cannot directly access the data stored locally. After authentication (e.g., by a pin code), she only gets the data according to her privileges.

Unfortunately, a PDS cannot provide all the required database functionalities (e.g., durability, if the PDS is lost or destroyed, availability when the PDS is disconnected, global queries involving data from several PDSs) without resorting to external servers, called hereafter *Supporting Servers*.

Supporting Servers: we assume that Supporting Servers are *Honest* but *Curious*, a common security assumption regarding Storage Service Providers. This means that they correctly provide the services that are expected from them (typically serve store, retrieve, and delete data requests) but they may try to breach confidentiality of any data that is stored locally.

Therefore, implementing the PDS approach requires solving the problem stated below:

- To revisit the main database techniques to make the PDS core compliant with the SPT hardware constraints.
- To reestablish the traditional functions of a central server (durability, availability, global queries) in a secure way using Honest but Curious Supporting Servers.
- To provide the user with intuitive tools and underlying mechanisms helping her to control how her personal data is shared.

The next two sections assume that the problem stated above can be solved. Then Section 5, 6 and 7 discuss the technical challenges associated to each dimension of this problem.

3. MOTIVATING EXAMPLES

3.1 Healthcare scenario

Alice carries her electronic healthcare folder (along with other information) on a PDS. She has an account on *e-Store*, a Supporting Server provider. She downloaded in her PDS, from the Ministry of Health, a predefined healthcare database schema, an application to exploit it, and an access control policy defining the privileges attached to each role (physician, nurse, etc). Alice may manage the role assignment by herself or activate specific user policies predefined by e.g., a patient association. When she visits Bob, a new physician, she is free to provide her SPT or not, depending on her willingness to let Bob physically access it (this is a rough but effective way to control the sharing of her data, as with a paper-based folder). In the positive case, Bob plugs Alice's PDS on his terminal, authenticates to the PDS server with his physician credentials, queries and updates Alice's folder through his local Web browser, according to the physician's privileges.

Bob prescribes a blood test to Alice. The test results is sent to Alice by the medical lab in an encrypted form, through e-Store acting here as a secure mailbox. The document is downloaded from e-Store and wrapped by Alice's PDS to feed the embedded database. If this document contains information Alice would like to keep secret, she simply masks this document so that it remains hidden from any user querying the database except her. The lab keeps track of this medical act for administrative purposes but does not need anymore to keep a copy of its medical content. If Alice loses her PDS, its tamper-resistance renders potential attacks harmless. She will then recover her folder from an encrypted archive stored by e-Store using, e.g., a pass-phrase.

Alice suffers from a long-term sickness and must receive care at home. Any practitioner can interact at home with Alice's PDS thanks to his netbook, tablet PC or PDA without need for an Internet connection. To improve care coordination, Bob convinces Alice to make part of her folder available 24/7, during a one month period, to him and to Mary, a specialist physician. A1ice uploads the requested part of her folder encrypted on e-Store. The secret key is exchanged with Bob's and Mary's PDSs in order for them to be able to download Alice's data on their own PDS and query it. While Alice's data is now replicated on Bob's and Mary's PDSs, Bob and Mary cannot perform actions on the replica exceeding their privileges and this replica will be destroyed after a one month period because their PDS will enforce these controls. Bob and Mary's actions are recorded by their own PDSs and sent back to Alice through e-Store for audit purpose. To make this sharing scenario possible, patients and practitioners are all assumed to be equipped with PDSs and these PDSs are assumed to share a compliant database schema. As shown in Annex A, which presents a field experimentation, this assumption is realistic in several practical situations.

Finally, if the Ministry of Health decides to compute statistics or to build an anonymized dataset from a cohort of patients, the targeted PDSs will perform the processing and deliver the final result while preventing any leakage of sensitive data or identifying information.

3.2 Vehicle tracking scenario

John, a traveling salesman, drives a car from his company during working hours and shares his personal car with Cathy, his daughter. Both have a PDS that they plug in the car to register all their personal trips. Several applications are interested in the registered GPS locations. John's insurance company adapts the insurance fee according to different criteria (e.g., the distance traveled, type of road used, and speed). Cathy will probably pay more than her father because she lacks enough driving experience. The Treasury is also interested by this information to compute John's carbon tax according to similar criteria, though the computation rules are different. Finally, John's company would also like to track John's moves to organize his rounds better. GPS raw data is obviously highly private. Fortunately, John's PDS externalizes only the relevant aggregated values to each application. In other words, each application is granted access to a particular view of the data registered in John's database.

3.3 BestLoan.com & BudgetOptim scenarios

Alice needs a loan to buy an apartment. She would like to find the best rates for her loan and, thus, relies on the service of *BestLoan.com* (BL for short), a mortgage broker. To assess Alice's financial situation, BL needs to get access to sensitive information from Alice's PDS such as salary, bank statements and tax information. Alice's data can be securely shared with Donald, a BL employee, as follows: (1) Alice opts in for the BL application and downloads the security policy associated to it in her PDS, (2) Donald authenticates to Alice's PDS with his credentials embedded in his own PDS and requests the required data, (3) Alice agrees to share this data with Donald for a specified duration (e.g., two weeks), (4) finally Donald downloads the data in his PDS, all this by exchanging messages and data through the e-Store Supporting Servers. Donald cannot perform actions on Alice's data exceeding their privileges or the retention period fixed by Alice because his PDS will preclude these actions. If Alice distrusts Donald, she can audit his activity and can at any moment opt out of the BL application (with the effect of deleting Alice's data in Donald's PDS), all this again by exchanging messages through the e-Store.

Alice now wants to optimize her budget and thus opts in for the BudgetOptim application (BO for short). BO runs locally on Alice's PDS with a GUI running on the terminal. BO accesses details of Alice's invoices, telecom bills, etc. in order to suggest more advantageous services according to her consuming profile. With BO application, Alice does not share data with anybody. This last scenario is typical of many private applications that can process personal data (e.g., diet advices, tax minimization, pension simulation, vaccine reminders, etc.).

3.4 Positioning

Compared to an approach where all personal data is gathered on traditional servers, the benefit provided by PDS is fourfold. First, the PDS holder is his own Database Service Provider. Hence, abusive uses by the Database Service Provider are precluded. Second, the PDS provides the holder with tangible elements of trust which cannot be provided by any traditional server (see factors 1 to 4 in Section 2). Third, privacy principles (e.g., limited retention, audit) can be enforced for the data externalized by the

holder provided the recipient of this data is another PDS. Fourth, the holder's data remains available in disconnected mode.

However, alternatives to the traditional server exist. The Hippocratic database approach [3] has been precisely designed to protect personal data thanks to principles like purpose specification, consent, limited collection, limited retention, audit, safety, etc. Part of our architectural ideas has been inspired by this work. But the Hippocratic database approach provides tangible guarantees only if the server can be fully trusted. In this respect, the PDS approach can be seen as a fully distributed implementation of a Hippocratic database where the founding Hippocratic principles can be definitely enforced.

The Database as a Service approach (DAS) [18] is another option. Here data is stored encrypted on the server and is decrypted at the client side, making server attacks harmless. This time, the DAS approach makes sense only if all clients can be trusted; the PDS provides a way to make the clients trusted.

Statistical databases [1] and data anonymization [14] are both motivated by the desire to compute statistics or to mine data without compromising sensitive information about individuals. Both require trusting the server, either to perform query restriction or data perturbation in the former case, or to produce the anonymized data set in the latter case. Though orthogonal to the PDS approach, these concerns still exist in the PDS context and must be addressed adequately.

4. PDS GLOBAL ARCHITECTURE

As mentioned in the introduction, PDS is not a simple secure repository of personal documents but rather provides a well organized, structured, consistent and queryable representation of these documents for serving applications requests. The difficulty to achieve this objective comes notably from the variety of data sources and applications targeted by PDS. This section presents an initial design of the PDS architecture.

4.1 Personal database

The personal database is assumed to be composed of a small set of database schemas, typically one per application domain. We make no assumption on the granularity of application domains but e-health and e-administration are illustrative examples of domains. Database schemas are defined by *DB Schema Providers*. Depending on the domain, a DB Schema Provider can be a government agency (e.g., Ministry of Health) or a private consortium (e.g., a group of banks and insurances).

Content Providers are external information systems that deliver personal data (e.g., blood test, salary form), encoded in XML. We make the simplifying assumption that each XML document conforms to one XML schema defined by a standardization organization (e.g., HL7) or by a DB Schema Provider (e.g., the Ministry of Health). To allow building a consistent and structured view of a set of related documents, an XML document (e.g., a prescription) is enriched with all referential data required to fill the embedded database accurately (e.g., detailed data related to the doctor who wrote the prescription and to the drug prescribed). Hence, the data contained in different documents related to a given doctor or drug can be easily queried and cross documents processing becomes possible (e.g., get the list of current medical treatments or compute average blood pressure during the last month). Then the enriched document is pushed in an encrypted form to the recipient PDS through Supporting Servers (see section 4.4 for a description of Supporting Servers). The recipient PDS downloads the XML document and wraps it into a set of records thanks to *mapping rules* provided by DB Schema Providers[2]. Mapping rules are declarative and interpreted by a generic wrapper, a certified component of the PDS core (see Section 4.5 for a deeper discussion on certification). The benefit of declarative mapping rules is not only that it simplifies the work of the DB Schema Provider but primarily that the safety of these rules can be controlled.

Figure 3 illustrates the wrapping of a prescription, enriched with doctor and drug referentials sent from a hospital. In this figure, we assume that the embedded database is relational but the choice of the database model (relational, XML, hybrid) has little impact in the global architecture. The document conforms to an XML schema for healthcare, and is wrapped into four tables (two of them being referentials) from the healthcare database schema. As shown in Figure 4, not all documents are wrapped and integrated in the database. Some documents (e.g., an X-ray image) can stay encrypted in the Supporting Servers and simply be referenced by the embedded database.

Note that problems incurred by the existence of several standards in a given application domain and the problems of data redundancy when database schemas overlap are orthogonal to the PDS approach and are not tackled in this paper.

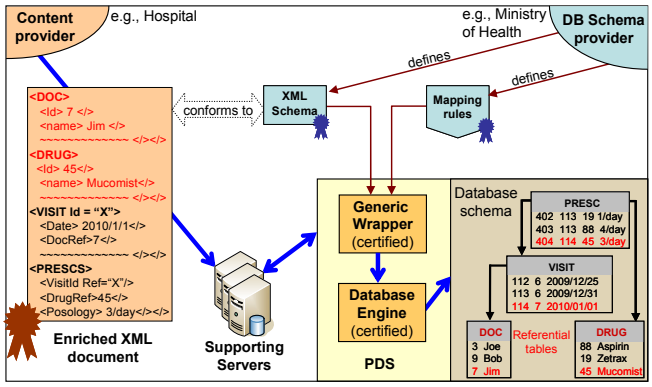


Figure 3: Wrapping a document into the PDS database

4.2 Applications

Applications are developed by *Application Providers* (e.g., BestLoan.com). They are defined on top of the published DB schema(s) of interest and can use all database functionalities provided by the embedded DBMS (i.e., DDL and DML statements). Each application defines a set of *collection rules* specifying the subset of documents required to accomplish its purpose (e.g., the five most recent salary forms are required by BestLoan.com). These rules are expressed at the document level to make them meaningful to the PDS holder (helping him to opt in or opt out of this application) and are mapped at the database level to be enforced similarly to access control rules. Applications can run locally (on the holder's PDS with a GUI on a terminal), on another user's PDS (e.g., on the doctor's one) or on an external server sending queries to the holder's PDS (e.g., the Treasury server computing the holder's carbon tax). While most applications are assumed to perform only selection queries, insertion of new documents is not precluded (e.g., a treatment prescribed at home by the doctor). An updating application will play the same role as a content provider and the insertion will follow the same process.

4.3 User Control

The prime way for the PDS holder to control the usage of her data is to opt-in/out of applications and to decide situations where she

[2] The mapping rules are related to the transcription of XML documents into a structured database and are required even with an XML database.

physically delivers her PDS to another individual (e.g., a doctor). Assuming that the PDS holder's consent has been given, the actions that any individual can perform are regulated by a predefined access control policy. This policy can either be defined by the DB schema provider (e.g., the Ministry of Health fixes a RBAC policy stating the privileges of each category of professionals according to current legislation) or be defined by the Application Provider and be ratified by a consumer protection association or the legislator.

Predefined access control policies are usually far too complex to be understandable by the PDS holder (e.g., the RBAC matrix regulating the use of the French EHR contains more than 400 entries). It is therefore mandatory to provide the PDS holder with simple tools to protect her sensitive data following her wish. A first way consists in managing the privileges through a simple GUI, as illustrated in the healthcare scenario. A second way is to give the user the ability to mask documents in the database. The records corresponding to a masked document are no longer considered at query execution time, except if the query is issued by the PDS holder herself (through an application). To make this process intuitive, the DB Schema Provider can predefine masking rules (e.g., hide documents by doctor, pathology, time period, etc.) exploiting the expressive power of the DBMS language and easily selectable by the user through a GUI.

The PDS holder (called hereafter the donor) can also impose privacy preserving rules whenever data leaves her PDS to enter another PDS. This sharing is required when a donor's data must be made available while her PDS is disconnected (see the healthcare scenario). This sharing must be ruled by the following principles:

- *Minimal exposure:* in a nominal use, only the results of authorized queries are externalized by a PDS and raw data always remains confined in the PDS. When donor's raw data is made available to others, this must be done in such a way that minimal data (limited collection principle) is exchanged during a minimal duration (limited retention principle) and with the minimum number of recipient PDS (need-to-know principle) to accomplish the purpose of this externalization.
- *Secure delete:* if the donor decides to delete a document before the retention period expires, all replicas of the corresponding raw data hosted by the recipient PDSs must be deleted.
- *Audit:* the donor must have the ability to audit the actions performed by all recipient PDSs on replicas.

Minimal exposure can be implemented by a Secure Publish/Subscribe mechanism working as follows. The raw data to be exchanged (published) is the records belonging to the database view computed over the data targeted by the purpose of the sharing, by intersecting the collection rules of the application, the predefined access control rules applied to the subscribers and the donor's masking rules. The donor publishes these records in an encrypted form on the Supporting Servers. The recipient PDSs subscribe to this data and receive the decryption key once the publisher has accepted the subscription. If the content of the view evolves in the publisher PDS (e.g., because new documents have been inserted), the update is pushed to the subscriber PDSs. We assume that publisher and subscriber PDSs have a compatible database schema (e.g., doctors and patients share a uniform healthcare DB schema).

In the following, we denote by *user's control rules* all rules which can be fixed by the PDS holder herself to protect her privacy, namely masking rules, retention rules and audit rules. User's control rules are enforced by all PDSs, both on the PDS holder's data and on the data downloaded after a subscription.

4.4 Supporting Servers

Supporting Servers Providers provide storage (for encrypted data) and timestamp services to implement the functions that PDSs cannot provide on their own, namely:

- *Asynchronous communication*: since PDSs are often disconnected, documents, shared data and messages must be exchanged asynchronously between Content Providers and PDSs and between PDSs themselves through a storage area.
- *Durability*: the embedded database must be recovered in case of PDS loss or destruction. The PDS holder's personal data can be recovered from the documents sent by Content Providers through the Supporting Servers (assuming these documents are not destroyed). Data downloaded from other PDSs can be recovered from the data published in the Supporting Servers (assuming their retention limit has not been reached). Other data (user's control rules definition, metadata built by applications, etc.) must be saved explicitly by the embedded DBMS on the Supporting Servers (e.g., by sending a message to itself).
- *Global processing:* a temporary storage area is required to implement processing combining data from multiple PDSs. Statistical queries and data anonymization are examples of such processing.
- *Timestamping*: the SPT hardware platform is not equipped with an internal clock since it takes electrical power from the terminal it is plugged in. Hence, a secure time server is required to implement auditing and limited retention.

4.5 Security

The security of the architecture lies in (1) the tamper-resistance of the SPT platform, (2) the certification of the embedded code (and ratification of declarative rules), and (3) the encryption of any data externalized in the Supporting Servers.

Regarding encryption, the security of data embedded in a given PDS is considered comparable to the security of the same data stored encrypted in the Supporting Servers as long as the key remains confined to this PDS.

Even if any data stored in the Supporting Servers is encrypted, the identity of the users downloading and uploading this data must be obfuscated. Indeed, spying communications could lead to disclosure of sensitive information (e.g., the volume of data sent by a hospital may reveal a heavy pathology). The Supporting Servers provide the storage required to make the communication asynchronous and the PDS themselves integrate a protocol making these communications anonymous.

The certification does not apply to all parts of the embedded code. Typically, assuming the certification of all embedded applications is unrealistic. Figure 4 shows the elements for which certification is mandatory, namely: (1) the core software (operating system, database engine), (2) the generic XML wrapper, (3) the communication manager, (4) the Publish/Subscribe manager and (5) the privacy manager enforcing the user's control rules. Implementing these software pieces and certifying them is the responsibility of the *PDS Providers* (e.g., a SPT manufacturer like Gemalto). Declarative rules need also to be ratified to prove their conformance to a public specification. This data is: (1) the mapping rules consumed by the wrapper, (2) the predefined access control rules, the predefined masking rules and the collection rules enforced by the DBMS. The documents themselves are assumed to be signed to prove their authenticity.

Trusting the predefined access control policies requires being able to authenticate all users. Depending on the application domains,

PKI infrastructures already serve this purpose. For example, in France, all healthcare professionals have a certificate embedded in a smart card containing their identity and role (a strong authentication is mandatory to access any server hosting healthcare data). In the same spirit, several countries are developing infrastructures based on smart cards or on software certificate to allow any citizen to authenticate electronically (e.g., IdéNum in France).

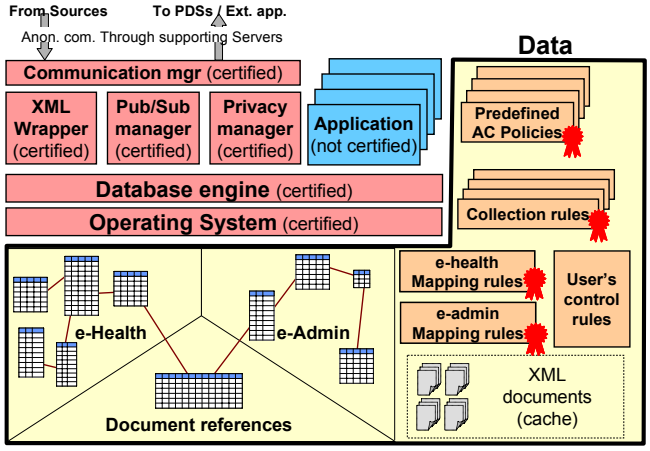


Figure 4: PDS generic software, application, and database

5. EMBEDDED DATA MANAGEMENT

The SPT hardware constraints presented in Section 2 introduce three main technical challenges discussed above.

Computing queries on Gigabytes of data without relying on external resources

Database queries must be executed on Gigabytes of data with Kilobytes of RAM. Join is the most RAM demanding operation. It is usually not supported in tiny RAM devices (e.g., sensors) while it is a central operator in the PDS context. The performance of "Last resort" join algorithms (block nested loop, sort-merge, Grace hash, hybrid hash) quickly deteriorates when the smallest join argument exceeds the RAM size [17]. Jive join and Slam join use join indices [20] but both require that the RAM size is of the order of the square root of the size of the smaller argument. In the PDS context, swapping data in the terminal or in the local NAND Flash is precluded due (1) to the dramatic amount of swapping required considering the ratio between the RAM size and the size of the data to be joined and (2) to the cost of encryption (only the microcontroller is trusted).

Consequently, the unique solution is to resort to a highly indexed model where all (key) joins are precomputed. In [7], we already proposed, in a relational context, a multiway join index called Subtree Key Table and a Climbing Index allowing to speed up selections at the leaves of a join tree. Combined together, these indexes allow selecting tuples in any table, reaching any other table in the join path in a single step. Queries can then be executed in a pure pipeline fashion without consuming RAM or producing intermediate results. This work must be considered as a first step towards the definition of indexing models and query execution techniques dedicated to tiny RAM devices.

Efficient atomic storage and indexing model in NAND Flash

NAND Flash chips exhibit uncommon characteristics: (1) reads and writes are done at a page granularity, but writes are more costly than reads, (2) a page cannot be rewritten without erasing the complete block containing it, which is a costly operation, (3) writes must be done sequentially within a block and (4) a block wears out after about 10^5 repeated write/erase cycles. A main consequence of this is that random writes can be up to order(s) of magnitude more costly than sequential writes [10]. Combining these constraints with the RAM limit makes the storage and indexing problem very challenging.

Regular indexing techniques (e.g., B+Tree) are poorly adapted to NAND Flash because of the high number of random writes they incur [23]. All improvements (e.g., BFTL [23], Lazy-Adaptive Tree [2]) rely on the idea to defer index updates using a log (or Flash-resident cascaded buffers) and batch them to decrease the number of writes. The side effect is a higher RAM consumption (to index the log or to implement write-coalescing of buffers) and a waste of Flash memory space.

A suggested alternative is to try to organize the whole database in a pure sequential way to take advantage of the update pattern of PDS (massive insertions, almost no updates, and few deletes) and of the Flash characteristics. The benefit of sequentiality is in minimizing the need for buffering and caching (thereby saving RAM), in avoiding random writes and in greatly simplifying transaction atomicity because only a set of high watermarks have to be maintained to determine whether NAND Flash blocks contain dirty data or not. Updates and deletes are not reported on the database. Rather, they are kept in a sequential list, the updated pages are marked and their up-to-date image is rebuilt on the fly when the page is loaded in RAM, in a way inspired by [19]. The index problem is more complex since even sequential insertions generate random updates in the index. In [25], we suggested a Flash-aware indexing technique, called PBFilter, which organizes also the index sequentially and speeds-up lookups thanks to partitioned Bloom filters. However, this strategy does not scale for GB of data. To tackle this problem, we are investigating a solution where the database is stratified so that the indexing strategy can change among strata without incurring a dramatic number of rewrites. We feel that designing storage and indexing techniques combining the Flash constraints and the embedded constraints (RAM limitation, optimal Flash usage) deserve a great interest considering the increasing diversity of Flash-based devices.

Enforcing local data confidentiality and integrity

The NAND Flash being not protected by the tamper-resistance of the microcontroller, cryptographic techniques must be used to protect the database footprint against confidentiality and integrity attacks. Indeed, integrity attacks make sense because the PDS holder herself can try to tamper the database (e.g., she could perform a replay attack to be refunded several times for the same drug or try to change an access control rule or the content of an administrative document, e.g., a diploma).

A primary concern in the PDS context is the granularity of the traditional encryption and hashing algorithms (e.g., 128 bits for AES and 512 bits for SHA). As explained above, the PDS query execution engine must rely on a highly indexed model, thereby generating very fine grain random accesses (in the order of the size of a pointer). Solutions to this problem can be: (1) designing encryption and hashing techniques for fine grain accesses [21] compatible with the SPT's resources, (2) designing clustering techniques so that relevant data are contiguous, in the spirit of the PAX models [4] and (3) encrypting the data in such a way that lookups can be done without decrypting the data. The idea here is different from order-preserving encryption or privacy-homomorphism. Roughly speaking, the idea is to exploit the sequentiality of the database to encrypt the data according to their insertion order (hence data having the same clear text get a different cipher text) but equality tests on the cipher text remain

possible if they take this order into account. Version management, required to detect replay attacks, is another complex issue. Maintaining a version number for each page in secure storage (i.e., in the NOR of the microcontroller) is unrealistic considering the small size of the NOR and the fact that it is primarily dedicated to the storage of application code. TEC-Tree [12] overcomes this problem by organizing secret information as a tree. However, it incurs update propagation in the tree, which badly adapts to NAND Flash. Again, our expectation is that the sequential organization of the database can lead to smarter version management techniques. Hence PDS introduces specific interesting challenges in terms of cryptographic techniques applied to database management.

6. DURABILITY, AVAILABILITY AND GLOBAL PROCESSING

Durability and Availability

Honest but Curious Supporting Servers are assumed to correctly store, retrieve and delete data requests on an unbounded storage area in a durable and highly available way. PDSs capitalize on this to implement higher level secure functions.

Anonymous communications between Content Providers and PDSs and between PDSs themselves can be implemented through the Supporting Servers using an anonymizing network like Tor [11], based on the Onion-routing protocol [16]. The anonymizing network provides a virtual circuit C from the source to the Supporting Servers. Thus, the latter can send data back to the source without knowing its identity, following the *return circuit* C^{-1} encoded in the initial message (this is called *Reply Onions* [16]). An interesting challenge is to use the secure microcontrollers of SPTs to increase the security of anonymous protocols, having SPTs as entry or exit point for the anonymous route.

Recipient PDSs must be able to retrieve messages or data sent to them. Although communications are anonymous, the difficulty lies in selecting the relevant message/data without disclosing any information that could help the Supporting Servers to infer PDS identity. A protocol tagging messages with anonymous markers is proposed to this end. The delete request is trickier to implement. First, the physical image of the targeted data should be destroyed by the Supporting Servers (e.g., for cleaning purpose) only if the requesting PDS can exhibit an anonymous proof of legitimacy for this request. Second, the deletion must be effective even if an attacker spies all messages sent to the Supporting Servers and records them. Hence, there is no other solution than removing definitely the access to some data (i.e., by removing the way to decrypt it) even if its image has been stolen and cannot be physically destroyed. To tackle this problem, we defined a protocol based on the Diffie-Hellman key agreement. Note that secure deletion is also a prerequisite to enforce masking and limited retention. Assuming Supporting Servers guarantee the durability of all messages/data sent to them (except those legitimately destroyed), the log enabling PDSs to recover after a crash or a loss comes for free. Finally, enforcing audit requires a protocol guaranteeing that audit logs are produced and delivered despite unpredictable disconnections of the subscriber and the publisher PDSs. An initial version of the main protocols is given in Annex C.

Global processing

Executing global processing over a set of autonomous trusted PDSs connecting to Honest but Curious Supporting Servers leads to unusual computations in order (1) to tackle the unpredictable nature of PDS connections and (2) to preserve PDS holders' privacy. We illustrate this through examples on relational data.

The Ministry of Health would like to prevent a pandemic. It executes a continuous-like query on each PDS that connects to the Supporting Server in order to select individuals having a given set of symptoms. If more than p individuals living in the same region are at risk, they are encouraged to go to a hospital. However, the patients consent to this form of dynamic queries only if their anonymity is guaranteed. The query can then be of the form 'SELECT *pseudonym, city* FROM *any PDS* WHERE *symptom* IN *(x,y,z)*' where pseudonym and city are sent to the querier in the clear through the Supporting Servers. If threshold p is reached, the querier sends messages back tagged with the pseudonym of the individual at risk to the Supporting Servers. Thanks to anonymous communication, a PDS holder can get the outcome of the query for herself without revealing her identity. Interesting issues lie in the organization of the continuous querying protocol, in the classification of the queries which can be managed in this manner and in the conditions to preserve anonymity (i.e., anonymity could be breached if successive queries succeed in recomposing the association between quasi-identifiers and sensitive attributes).

Statistical databases [1] aim at answering aggregate queries (e.g., "SELECT *AVG(IQ)* FROM ... WHERE *Age=10* AND *Diagnosis='Dyspraxia'*") without compromising sensitive information about individuals. Examples of disclosure control techniques include analyzing the query trail to prevent compromising overlaps between successive queries and/or perturbing the result without affecting the global distribution [24]. An interesting feature of the PDS context is that successive aggregates are computed over a fluctuating population of PDSs (due to the unpredictable nature of PDS connections), making inference among runs harder and influencing the design of disclosure control algorithms accordingly.

Privacy Preserving Data Publishing is another form of global processing aimed at publishing a set of micro-data while protecting the identity of individuals. The traditional process is composed of three phases: data collection, computation of sanitization rules based on the collected data and finally data sanitization. The challenge here is to design a distributed protocol that (1) allows the publisher (through the Supporting Servers) to collect enough data from the targeted PDSs to compute the sanitization rule, and then (2) delegates the sanitization process itself to the PDSs (so that raw data is never exposed) while providing them a way to control the safety of the sanitization rules. We suggested a preliminary solution [5] for a sanitization algorithm preventing record linkages through k-anonymity [22]. Much work remains to be done to prevent from other types of linkages (e.g., attribute linkage prevention through l-diversity [14]).

7. USER CONTROL

Enforcing user's control rules, namely masking, limited retention and audit and combining them with application's collection rules introduce a set of interesting problems described below:

Impedance mismatch between documents and databases

While predefined access control rules (e.g., RBAC matrix published by an application or by the DB Schema Provider) and queries issued by applications are expressed at the database level (e.g., in SQL), user's control rules as well as application's collection rules are expressed over documents to be meaningful for the end-user. Conversely, for audit purposes, accesses are recorded at the database level but must be delivered to the end-user at document level in order to interpret them. Consequently, *translation structures* must be integrated in the PDS to store document-to-record and record-to-document links.

The query engine must integrate these links in the query evaluation in order to compute a result compliant with the application's collection rules, the predefined access control rules and the user's masking rules. The evaluation can be as follows. When a document D (e.g., a medical prescription) is inserted in the database, the records created at wrapping time reference D in the database (records related to referentials like doctors and drugs are not concerned). Let S_c be the set of documents targeted by the collection rules of application A and S_m be the set of documents targeted by the user's masking rules. When A queries the database, the query result includes the document references for each selected record r and this result is post-filtered to keep only the records satisfying ($r \in S_c \wedge r \notin S_m$). Post-filtering can be implemented efficiently in RAM constrained environments using Bloom filters [9].

When a delete request is issued for D or when D reaches its retention limit, it must be removed from the database. The translation structures are used to identify all records related to this document. This includes the records referencing D either directly (e.g., prescription elements) or transitively (i.e., the referential data like the doctor who does the prescription and the drug prescribed). The presence of referential data in a personal database is sensitive and the related records must be removed as well. The difficulty lies in the fact that referential data may be shared by other documents. A garbage collector algorithm [3] must be designed to tackle this problem. The deletion of the targeted records can be logical (following the marking process sketched in Section 5) or physical, the latter case being more costly due to the Flash constraints.

Propagating user's control rules to other PDSs

If data has been uploaded on the Supporting Servers by a publisher PDS and downloaded by a subscriber PDS, the user's control rules defined by the publisher must be propagated to the subscriber. Being able to implement the mechanisms presented above on the subscriber PDS requires sending the user's control rules and the translation structures along with the data and forwarding to the subscriber any masking and delete operation performed on the fly by the publisher. Hence, the effect of user's control rules will be the same independently of the location of the data and of the number of replica.

8. CONCLUDING REMARKS

Our belief and hope is that the emergence of new hardware devices combining portability, secure processing and Gigabytes-sized storage will revolution the way people think about management and protection of personal data. The vision proposed in this paper of a secure and portable Personal Data Server is a first contribution in this direction. We have presented an initial design for this vision and have identified important technical challenges related to it. Moreover, Annex A presents an experiment in the healthcare field which prefigures the PDS approach and gives some confidence about the feasibility of converting the PDS vision into reality.

Simplifying assumptions have been made, other solutions could have been envisioned to tackle the identified challenges and new challenges could also have been identified by enlarging the PDS vision. We have considered a highly structured vision of the personal database to support rich applications and we have made strong security assumptions by considering that the PDS is the main element of trust in the architecture. These two options can be debated and reconsidered, opening the way for other exciting research work.

Acknowledgments: The authors wish to thank Anne Canteaut and Philippe Bonnet for their helpful comments on this paper.

[3] Storing reference counters is badly adapted to the Flash update constraints. An option can be to recompute counters dynamically.

9. REFERENCES

[1] Adam, N. R. and Worthmann, J. C. Security-control methods for statistical databases: a comparative study. ACM Comput. Surv., 1989.

[2] Agrawal, D., Ganesan, D., Sitaraman R., Diao Y. and Singh S. Lazy-Adaptive Tree: An Optimized Index Structure for Flash Devices. VLDB, 2009.

[3] Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y. Hippocratic Databases. VLDB, 2002.

[4] Ailamaki, A., DeWitt, D.J. and Hill, M. D. Data page layouts for relational databases on deep memory hierarchies. The VLDB Journal, 2002.

[5] Allard, T., Nguyen, B. and Pucheral, P. Safe Anonymization of Data Hosted in Smart Tokens, PRiSM Technical Report n° 526, 2010.

[6] Allard, T., Anciaux, N., Bouganim, L., Pucheral, P., Thion, R. Trustworthiness of Pervasive Healthcare Folders, Pervasive and Smart Technologies for Healthcare, Information Science Reference, 2009.

[7] Anciaux, N., Benzine, M., Bouganim, L., Pucheral, P. and Shasha, D. GhostDB: Querying Visible and Hidden data without leaks. ACM SIGMOD, 2007.

[8] Anciaux, N., Bouganim, L., Guo, Y., Pucheral, P., Vandewalle J-J. and Yin, S. Pluggable Personal Data Servers. ACM SIGMOD, 2010.

[9] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970.

[10] Bouganim, L., Jónsson, B. Þ. and Bonnet P. uFLIP: Understanding Flash IO Patterns. CIDR, 2009.

[11] Dingledine, R., N. Mathewson, and Syverson P. Tor: The Second-Generation Onion Router. USENIX, 2004.

[12] Elbaz, R., Champagne, D., Lee, R. B., Torres, L., Sassatelli G. and Guillemin P. TEC-Tree: A Low-Cost, Parallelizable Tree for Efficient Defense Against Memory Replay Attacks. CHES, 2007.

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

[14] Fung, B. C. M., Wang K., Chen R. and Yu P. S. Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys, 2010. To appear.

[15] Gagneux, M. Recommandations de la mission de relance du projet de DMP. http://www.sante-jeunesse-sports.gouv.fr/IMG/pdf/Rapport_DMP_mission_Gagneux.pdf (in French).

[16] Goldschlag, D., M. Reed, and Syverson P. Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM, 1999.

[17] Haas, L. M., Carey, M. J., Livny, M. and Shukla, A. Seeking the truth about ad hoc join costs. VLDB Journal, 1997.

[18] Hacıgümüş, H., Iyer, B., and Mehrotra, S. Providing Database as a Service. ICDE, 2002.

[19] Lee, S. and Moon, B. Design of flash-based DBMS: an in-page logging approach. ACM SIGMOD, 2007.

[20] Li, Z. and Ross, K. A. Fast joins using join indices. VLDB Journal, 1999.

[21] Robshaw, M., Billet, O. New Stream Cipher Designs - The eSTREAM Finalists, LNCS 4986, 2008

[22] Sweeney, L. k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst, 2002.

[23] Wu, C., Chang, L., and Kuo, T. An Efficient B-Tree Layer for Flash-Memory Storage Systems. RTCSA, 2003.

[24] Xiao, X. and Tao, Y. Output perturbation with query relaxation. VLDB, 2008.

[25] Yin, S., Pucheral, P. and Meng, X. A Sequential Indexing Scheme for flash-based embedded systems. EDBT, 2009.

Annex A: The DMSP experimental project

This annex presents an experimental project of secure and portable medical-social folder (DMSP in French) [6]. Its goal is to improve the coordination of medical and social cares while giving the control back to the patient over how her data is accessed and shared. The DMSP project fits well the PDS vision but is less ambitious and general. It constitutes, however, a first real-life experience bringing some insights on the benefits and feasibility of managing highly sensitive personal data on Secure Portable Tokens. The goal of this annex is precisely to present these insights exemplifying the PDS approach on a real-case study.

The DMSP project is funded by the Yvelines District and is led by INRIA (the French National Research Institute in Computer Sciences). It involves the University of Versailles, SANTEOS (the provider hosting the French National EHR system), Gemalto (the smart card world leader), ALDS and COGITEY (two gerontology networks), the project being targeted to elderly people.

The ageing of population makes the organization of home care a crucial issue and requires sharing medical and social information between different participants (doctors, nurses, social workers, home helpers and family circle) at the patient's bedside. Server-based EHR solutions are inadequate because (1) the access to the folder is conditioned by the existence of a high speed and secure internet connection at any place and any time; and (2) they fail in providing ultimate security guarantees to the patients, a fundamental concern for patients facing complex human situations (diagnosis of terminal illness, addictions, financial difficulties, etc).

The goal of the DMSP project was precisely to address these concerns. The solution adopted is a simplified and mono-application instance of the PDS architecture.

Patients are equipped with SPTs embedding a personal server to manage their medical-social folder. The form factor of patient's SPT is a USB token. The French law imposes that all professionals strongly authenticate to any server containing medical data thanks to a Health Professional Smart Card. This led Gemalto to develop a specific smart badge (see figure 5) acting both as a smart card reader and as a SPT used for synchronization purpose.

A central server achieves the durability and the availability of the patient's folders (without risk of privacy breach, as discussed next), and imports/exports data from/to the gerontology networks information systems. However, few elderly patients have an internet connection at home. Hence the SPTs of professionals are used to carry synchronization data between the central server and the patient's SPTs thereby implementing a "pedestrian network", the latency of which is linked to the frequency of visits at home.

The patients' folder includes social information such as financial resources or scores measuring possible lack of autonomy, as well as medical data like diagnosis, treatments, and evolution of medical metrics (e.g., weight, blood pressure, cholesterol, etc.).

Database schema

The data stored in DMSP has been modeled in a highly structured way to allow expressing powerful queries and access control rules in the application. For instance, diagnosis and prescriptions produced by the professionals are all wrapped into relational tables, e.g., Professional, Visit, Prescription, and Drug tables. The wrapping principle is similar to the one illustrated in Figure 3 though the wrappers used in DMSP are not generic. The contents of Professional and Drug are referential data shared by several documents.

Transaction atomicity and durability

Transaction atomicity is required for inserting documents, e.g., while inserting a diagnosis and associated prescriptions. Also, processing synchronization files issued from the central server requires atomicity. Transaction durability for updates performed at the patient's bedside is ensured only when the synchronization file reaches the central server through the "pedestrian network".

Access control

Predefined access control rules have been set in conformance with the healthcare RBAC matrix edited by the French government. Authorized views are expressed by select-project-join-agg queries. For example, nurses are granted access to the current medical prescriptions (not to the complete history) to be able to administer the treatment. Health professional cannot access raw social data, and vice-versa, but a reduced set of aggregates is allowed. Different levels of authentications are supported, i.e., strong authentication for professional, login/password for family circle, no authentication for occasional visitor with highly restricted access (access to emergency contacts and to a dashboard to notify some events).

User control

The patient can regulate her privacy by (1) selecting professionals that can access her folder (respecting the predefined access control policy); (2) share part of her folder securely within a restricted trusted circle of professionals; and (3) mask sensitive documents (e.g., the one produced by a given doctor, during a given period of time, for a given pathology). These features were required to make the DMSP acceptable for patients; otherwise, the initiative may have been perceived as an additional transgression of their privacy, equivalent to publishing their former "paper-based" folder on the internet. Consequently, part of the patient's data stored on the central server is encrypted and the cryptographic keys remain confined in the Patient's SPT and in the SPTs of the member of her trusted circle (i.e., in the spirit of the PDS publish/subscribe mechanism but less general and dynamic than this protocol).

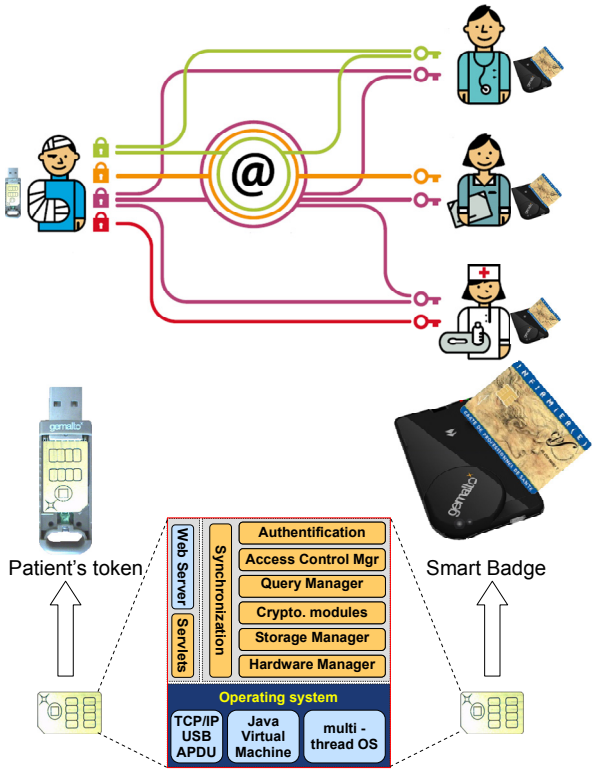


Figure 5. The DMSP project

Queries and application

The SPTs embed a local Web server, Servlets implementing the DMSP application, a JDBC driver, and the DBMS engine. The DMSP application issues SQL queries to feed the GUI and the embedded query engine computes these queries over views (access controls) and takes into account masking rules. The query and part of the application logic is processed inside the SPT microcontroller (DBMS engine and Servlets). The presentation of the results and additional computations (e.g., plotting curves of cholesterol rates) are done by applets using JavaScript.

An experiment in the field has begun since the end of 2009 and involves 25 practitioners and 100 elderly patients. Both the DMSP application and the DBMS internals will be demonstrated at ACM Sigmod 2010 [8], focusing on some elements mentioned in this paper (e.g., PBFilters, SKT and climbing index).

In summary, the DMSP project can be considered as a prefiguration of the PDS vision, with several restrictions in terms of genericity and dynamicity. However, it already demonstrates the interest of this vision in terms of privacy preservation. The experiment in the field will last during the next 18 months and more ambitious extensions are already foreseen. Typically, the French government has adopted, very recently, a law allowing triggering experiments of healthcare folder managed on secure USB tokens[4] for long term illnesses[5]. This could open future opportunities, relying on the DMSP experience and building on the PDS vision.

Additional information on the DMSP project can be found at http://www-smis.inria.fr/~DMSP/home.php

Annex B: Large scale privacy violations

An example of recent privacy violations caused by negligence:

- The *National Archives and Records Administration* (NARA) is investigating on the loss of a hard drive containing more than 70 millions of veterans' records (social security numbers, dates of birth, names…). The failing hard drive was first sent for repair; however, as the task was too complicated, it was outsourced as-is to another enterprise to be recycled. (DataLossDB, 05 October 2009).

An example of abusive usage of data:

- Attackers who managed to gather hundreds of British medical folders announced that they would sell them £ 4 per unit. They said that several customers – marketing, insurance offices – were interested to sell targeted products to vulnerable people. These medical folders came from a private hospital whose documents were outsourced to an enterprise (which used an Indian subcontractor) in order to be digitized. (The Daily Mail Online, 19 October 2009).

An example of data breaches caused by internal attackers:

- One of the largest reported data breach caused by a malicious insider occurred in 2004 at America Online when 92 million email addresses for 32 million subscribers were sold to spammers. (DataLossDB, Open Security Foundation).

An example of external attacks:

- 30 000 patients of UCSD's Moores Cancer Center have been notified that their personal data - names, dates of birth, medical record number, diagnosis and treatment dates dating back to 2004 – have been leaked after a hacker breached the center's data servers. (Sign on San Diego, 15 July 2009).

Even the most secured servers are not spared:

- 1 600 soldiers have been notified that some personal information, including their names, e-mail messages, phone numbers, home addresses, awards received, ranks, gender, ethnicity and dates of deployment on the field have been breached after an U.S. Army database has been penetrated by unauthorized users. (Federal Computer Week, 12 March 2009).
- A recent computer intrusion that forced the FBI to shut down its computer network and disrupted FBI operations for about 48 hours was traced to an e-mail containing malicious code that originated in China, according to FBI officials. (The Washington Times, 18 June 2009).

Examples of the demands from users for more control:

- 60% of the American people can be considered as privacy pragmatists: they have strong feelings about privacy and are very concerned to protect themselves from the abuse or misuse of their personal information by companies or government agencies. (Alan Westin, Harris Privacy Survey, 2003). Notably, 43% of the people consider that the privacy risk incurred by EHR systems outweighs the excepted benefit. (Harris/Westin survey, 'Privacy and EHR Systems', 2006).
- In the Netherlands, privacy and access concerns are major arguments for the postponement of the national EHR (The International Council on Medical & Care Compunetics, 2009). In particular, the lack of security measures limiting data access for service providers and the loss of control on their own data has been identified as a main reason for citizens to opt-out of the system (within 2 months over 330.000 persons opted-out).

Annex C: Communication protocols

This annex describes the main protocols necessary for anonymously and asynchronously exchanging messages, and supporting deletion. Table 1 introduces a set of notations used in the protocols.

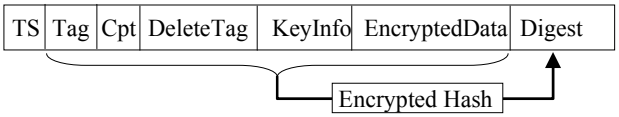
Table 1: List of Symbols used in Protocols

$E_X^{pub}[M]$	Encryption of message M with the public key of entity X.
$E_k[M]$	Encryption of message M with secret key k
$M_1 \| M_2$	Concatenation of messages M_1 and M_2
$H[M]$	Cryptographic hash of message M
$rand_k()$	A pseudorandom number generator using a secret key k, specific to each actor (PDSs or Content Providers)
$ID(X)$	Publicly known identifier of entity X
TS	Timestamp generated by a time server
N	Null value

Messages sent and stored on Supporting Servers have the following structure:

[4] See: http://www.assemblee-nationale.fr/13/dossiers/dossier_medical_cle_USB.asp

[5] Considering long term illnesses is of utmost interest since they concern 15% of the population but amount to 65% of the total medical expenses (www.assemblee-nationale.fr/13/rapports/r2347.asp)



TS is a timestamp acquired by the supporting server thanks to a secure time server and added to the message to allow filtering out the messages a recipient PDS already received. *Tag* is an anonymous marker allowing a receiver PDS to retrieve its messages on the Supporting Servers. *Cpt* is a counter associated to each sender/receiver pair (or to each marker), incremented by the sender and used by the receiver to check the correctness of the message ordering (not shown in the protocol). *DeleteTag* is a proof of legitimacy for the delete operation, as explained next. *KeyInfo* is a session key used to produce the *EncryptedData* field, itself encrypted with the public key of the receiver. *EncryptedData* is the actual content of the message. Finally, *Digest* is a hash of the previous fields, encrypted with the session key of *KeyInfo* and is used to check the integrity of the message (not shown in the protocols).

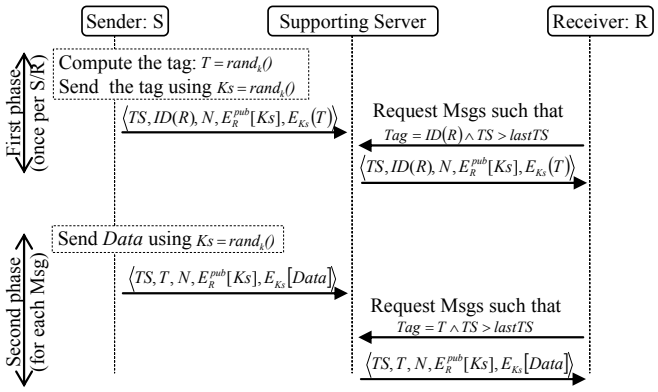


Figure 6. Communication using markers

Message marking and retrieval from Supporting Servers

The protocol to establish anonymous markers works in two phases (see Figure 6). In the first phase, the sender computes a tag *T* (which will be used to tag the next messages) thanks to the pseudorandom number generator. The computed tag *T* is transmitted encrypted with the session key *Ks*, itself encrypted with the public key of the receiver. This first message between a sender and a receiver is itself tagged with the public identifier of the receiver *ID(R)*. Note that, while the receiver identifier is transmitted in clear-text in this first message, it does not disclose sensitive information because (1) the sender is anonymous and (2) for a sender/receiver pair there is only one message of that kind. Hence, an attacker could only count the number of entities who established a communication with a given PDS.

In the second phase, data is exchanged using the defined marker *T*, the timestamp *TS*, and the session key *Ks* encrypted with the public key of the receiver. Note that the reuse of markers with timestamps allows a passive observer to determine that new data items are shared possibly between the same sender and the receiver. Since all communications are anonymous this information cannot be exploited further to link a particular data item to one specific sender or receiver. However, this information could be hidden by changing the marker periodically, transmitting the new marker in the last message using the current marker.

Deletion with proof of legitimacy

A proof of legitimacy is required to guarantee that only the PDS which produces a data can delete it. Audit data is a special case where the PDS which is granted permission to delete some audit data (i.e., the publisher) is actually not the PDS which produces it (i.e., the subscriber). We illustrate below the protocol used when the delete right is delegated to the receiver. The protocol when the sender keeps the delete right can be deduced easily. The idea is based on cryptographic hash functions preimage resistance property. The sender computes a random value called *Delete Proof* or *DP* and applies a cryptographic hash, thus obtaining *DT*, the *Delete Tag*. To transmit the delete right to the receiver, the sender simply adds *DP* to the data before encrypting it. When the receiver receives the message, it extracts *DP* and stores it. At delete time, the receiver sends a delete request, sending *DP* to the Supporting Server. Since given the hash value *DT*, it is computationally infeasible to find *DP*, such that *DT = H(DP)* (pre-image resistance property), the Supporting Server knows that the delete request was sent by an authorized PDS.

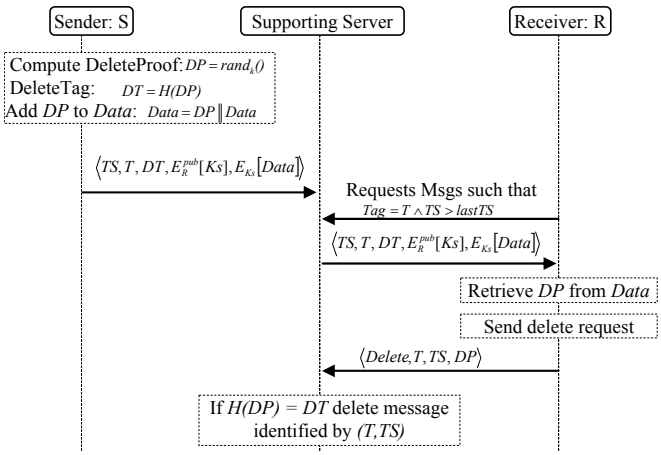


Figure 7. Deletion with proof of legitimacy for the receiver

Secure deletion

All data stored in the Supporting Servers have been carried by messages. Hence deleting a data on the Supporting Servers amounts to deleting the corresponding message. Since the communications may be spied by an attacker and the messages copied, there is no other solution for enforcing the deletion than removing permanently the access to this message. This can be implemented as follows. The sender and the receiver establish a secret key using the Diffie-Hellman key agreement protocol and use it to encrypt the message (thus do not fill the KeyInfo field). When, e.g., the sender decides to delete the message, he destroys his partial secret and sends a message to the receiver requiring deletion of his partial secret. Even if an attacker tampers one of the SPT after the deletion occurs, he cannot recover the message. This idea is simple but the protocol to implement it is more complex due to the fact that each party must be able to recover this message (assuming it has not been yet deleted) in case of a SPT failure (i.e., to ensure the durability property).

Annexe B.

Trusted Cells: a Sea Change for Personal Data Services

Nicolas Anciaux, Philippe Bonnet, Luc Bouganim, Benjamin Nguyen, Philippe Pucheral, Iulian S. Popa

Conference on Innovative Database Reasearch (CIDR), 4 pages, 2013.

Trusted Cells: A Sea Change for Personal Data Services

Nicolas Anciaux[1, 2], Philippe Bonnet[3], Luc Bouganim[1, 2],
Benjamin Nguyen[1, 2], Iulian Sandu Popa[1, 2], Philippe Pucheral[1, 2]

[1] INRIA Paris-Rocquencourt
Le Chesnay, France
<Fname.Lname>@inria.fr

[2] PRISM Laboratory
Univ. of Versailles, France
<Fname.Lname>@prism.uvsq.fr

[3] IT University of Copenhagen
Copenhagen, Denmark
phbo@itu.dk

ABSTRACT

How do you keep a secret about your personal life in an age where your daughter's glasses record and share everything she senses, your wallet records and shares your financial transactions, and your set-top box records and shares your family's energy consumption? Your personal data has become a prime asset for many companies around the Internet, but can you avoid -- or even detect -- abusive usage? Today, there is a wide consensus that individuals should have increased control on how their personal data is collected, managed and shared. Yet there is no appropriate technical solution to implement such personal data services: centralized solutions sacrifice security for innovative applications, while decentralized solutions sacrifice innovative applications for security. In this paper, we argue that the advent of secure hardware in all personal IT devices, at the edges of the Internet, could trigger a sea change. We propose the vision of *trusted cells*: personal data servers running on secure smart phones, set-top boxes, secure portable tokens or smart cards to form a global, decentralized data platform that provides security yet enables innovative applications. We motivate our approach, describe the trusted cells architecture and define a range of challenges for future research.

1. INTRODUCTION

With the convergence of mobile communications, sensors and online social networks technologies, we are witnessing an exponential increase in the creation and consumption of personal data. Paper-based interactions (e.g., banking, health), analog processes (e.g., photography, resource metering) or mechanical interactions (e.g., as simple as opening a door) are now sources of digital data linked to one or several individuals. They represent an unprecedented potential for applications and business.

Until now, the enthusiasm for new opportunities has thwarted privacy concerns. Nevertheless, the risk of a backlash is growing as new devices and new services bring us closer to the dystopias described in the science fiction literature. This risk is well documented and the nature of the solution is consensual: it is necessary to increase the control that individuals have over their personal data [11,9,12]. The World Economic Forum even formulates the need for a data platform that allows individuals *to manage the collection, usage and sharing of data in different contexts and for different types and sensitivities of data* [13].

Unfortunately, none of the solutions available today can be used to implement this vision. Centralized solutions, including emerging cloud-based personal data vaults management platforms[1], trade security and protection for innovative services. At best, such approaches formulate sound privacy policies, but none of them propose mechanisms to automatically enforce these policies [1]. Even TrustedDB [3], which proposes tamper-resistant hardware to secure outsourced centralized databases, does not solve the two intrinsic problems of centralized approaches. First, users get exposed to sudden changes in privacy policies. Second, users are exposed to sophisticated attacks, whose cost-benefit is high on a centralized database.

Decentralized solutions are promising because they do not exhibit these intrinsic limitations. However, existing decentralized solutions sacrifice functionality or usability for security. Many examples are discussed in [8]. Other examples include the PDS vision [2] or the FreedomBox [4]. In PDS, a personal data server is embedded in a tamper-resistant portable token to hold the personal data of a user, but the sharing of data is cumbersome (since the tokens are mostly disconnected) and the range of personal services is limited (since the tokens have extreme resource constraints). FreedomBox aims at providing a software platform that interconnects groups of individuals that trust each other, thus drastically limiting the range of services it can support.

We argue that the advent of secure hardware embedded in all forms of personal devices, at the edges of the Internet, will trigger a sea change. Recently, AMD announced that it will incorporate a secure Trust Zone-based[2] ARM processor on its chips to be included into smart phones, set-top boxes and laptops. Such secure tamper-resistant microcontrollers provide tangible security guarantees in the context of well-known environments[3]. We can now imagine that whenever you take a picture, your smart phone securely contacts the personal services of all individuals in the frame of the picture, and automatically blurs the face of those who request it. We can also imagine that the GPS tracker in your son's car gives him detailed turn-by-turn guidance, but hides those details to local government, only delivering road-pricing results.

In this paper, we propose the vision of **trusted cells**, i.e., personal data servers running on secure devices to form a decentralized data platform. We illustrate how trusted cells can be used in the context of an application scenario, describe the trusted cells architecture and discuss requirements and challenges for future research.

This article is published under a Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), which permits distribution and reproduction in any medium as well allowing derivative works, provided that you attribute the original work to the author(s) and CIDR 2013.
6th Biennial Conference on Innovative Data Systems Research (CIDR '13) January 6-9, 2013, Asilomar, California, USA.

[1] These include Personal (http://www.personal.com), My personal vault (http://www.mypersonalvault.com), or Mydex (http://www.mydex.org).
[2] http://www.arm.com/products/processors/technologies/trustzone.php
[3] The adoption of a standard API for secure micro-controllers [5] and the availability of an open source embedded secure operating system based on it (Open Virtualization) now enable higher level services.

2. MOTIVATION

Alice lives in France with Bob and their two children. Their house is now one of the 35 million households equipped with a Linky power meter. The power meter reports once a day to the distribution company, a certified time series of readings for verification, billing and network operation [6]. Alice and Bob have installed an energy butler app on their secure home gateway, a trusted cell managing all smart appliances in their home and storing their data. That award-winning app relies on external feeds from their utility and local weather prediction, as well as a feed of readings received every second from the Linky[4], to control their heat pump and the charge of their electrical vehicle. This app minimizes overall load on the distribution network and saves them 30% on their bill. In addition, Alice is engaged in a social game (a follow-up to simpleEnergy.com) where she competes with some friends on their energy savings, reducing consumption by 20%.

At the 1Hz granularity provided by the Linky, most electrical appliances have a distinctive energy signature. It is thus possible to infer from the power meter data which activities Alice and Bob are involved in at specific points in time [7]. How do Alice and Bob configure the home gateway trusted cell to preserve privacy while preserving the benefit of their applications? They have a shared account on this trusted cell. Bob, Alice and their children have agreed that they do not want to fully disclose all their activities to each other. They rather have access to 15 min aggregates via a visualization app – at that granularity one cannot detect specific activities, but it is still possible to infer a daily routine. At the same time, daily statistics feed their social game, monthly statistics are delivered to the distribution company and time series at required granularity are securely exchanged with other trusted cells in their neighborhood to achieve consumption peak load shaving.

None of this data leaves the trusted cell application unless it is accessed via a predefined set of aggregate queries. The trusted cell guarantees that no malware can tamper with the data. If the trusted cell gets stolen, an elaborate attack would need to be mounted to break the secure hardware and get access to their personal data.

This scenario can be easily transposed to different types of personal data like GPS traces, Internet traces, mobile phone data, bills, pay slips, photos as well as health, administrative or scholar records. We classify the data that could be managed with trusted cells, based on how and who actually produces it:

(1) *Data produced by smart sensors* installed by companies in the user's home (e.g., power-meter, heat sensor) or in the user's environment (e.g., user's car GPS tracking box for a PAYD application) on which the user has full or shared ownership, externalizing aggregated data. Users may opt-in for small-scale sharing (e.g., local traffic optimization) or larger-scale sharing (e.g., social games or traffic optimization).

(2) *Data produced or inferred by external systems* (e.g., purchase receipt obtained by near field communication or medical data sent by the hospital or labs). Small-scale sharing allows the user to optimize her buying habits or to compare her medical treatment with people having the same disease. Larger-scale sharing brings public health insights (e.g., epidemiological study cross-analyzing diseases and alimentation).

(3) *Data authored by the user herself* (e.g., a photo, a mail, a document) on which she has complete ownership. Small-scale sharing benefit is obvious here. Larger-scale sharing of partial data (e.g., photo location only, number of exchanged mails) is undoubtedly a source of precious information (e.g., most interesting places on Google maps).

3. TRUSTED CELLS ARCHITECTURE

What personal data services actually run on a trusted cell? How do these services allow a user to control whom she shares her secrets with? How do applications access these services? What kind of guarantees do trusted cells offer about the security of the data they manage? We obviously do not aim at answering those questions fully in this paper. Our goal here is to draw the contours of an architecture based on *Trusted Cell*s interconnected via an *Untrusted Infrastructure*.

Trusted Cells: A trusted cell implements a client-side reference monitor [10] on top of secure hardware. At a minimum, the hardware must guarantee a clear separation between secure and non-secure software. We abstract a Trusted Cell as (1) a Trusted Execution Environment, (2) a tamper-resistant memory where cryptographic secrets are stored, (3) an optional and potentially untrusted mass storage and (4) communication facilities. Physically, a trusted cell can either be a stand-alone hardware device (e.g., a smart token) or be embedded in an existing device (e.g., a smartphone based on ARM's TrustZone architecture).

The very high security provided by trusted cells comes from a combination of factors: (1) the obligation to physically be in contact with the device to attack it, (2) the tamper-resistance of (part of) its processing and storage units making hardware and side-channel attacks highly difficult, (3) the certification of the hardware and software platform, or the openness of the code, making software attacks (e.g., Trojan) also highly difficult, (4) the capacity to be auto-administered, contrary to high-end multi-user servers, avoiding insider (i.e., DBA) attacks, and (5) the impossibility even for the trusted cell owner to directly access the data stored locally or spy the local computing (she must authenticate and only gets data according to her privileges).

In terms of functionality, a full-fledged trusted cell should be able to (1) acquire data and synchronize it with the user's digital space, (2) extract metadata, index it and provide query facilities on it, (3) cryptographically protect data against confidentiality and integrity attacks, (4) enforce access and usage control rules, (5) make all access and usage actions accountable, (6) participate to computations distributed among trusted cells. Basic (e.g., sensor-based) trusted cells may implement a subset of this.

Untrusted infrastructure: The infrastructure provides the storage, computing and communication services, which expand the resources of a single trusted cell and form the glue between trusted cells. By definition, the infrastructure does not benefit from the hardware security of the trusted cell and is therefore considered untrusted. We consider that the infrastructure is implemented by a Cloud-based service provider[5].

In terms of functionality, the untrusted infrastructure is assumed to: (1) ensure a highly available and resilient store for all data outsourced by trusted cells, (2) provide communication facilities

[4] In France, such a short-range radio link is a requirement from the regulation authorities. In other countries, the data from a smart meter might not be directly accessible. In the US for example, the Green Button initiative allows customers to obtain online the smart meter data collected by their utility (http://www.greenbuttondata.org/)

[5] A P2P infrastructure among trusted cells could be envisioned but would raise many technical issues of limited interest for this article.

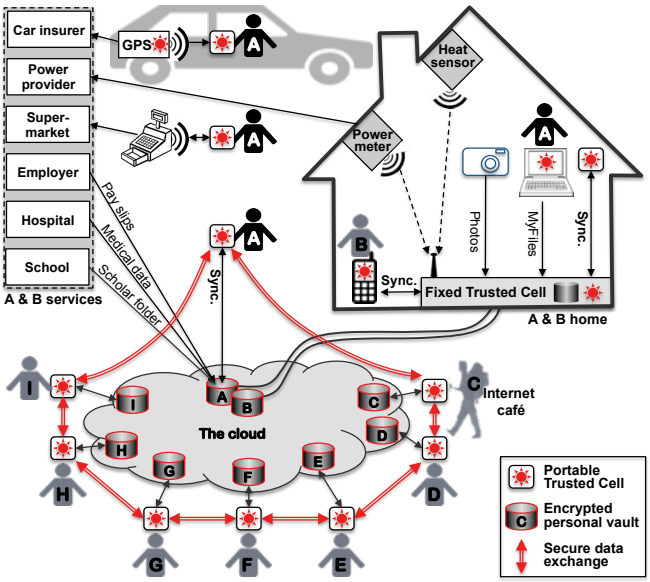


Figure 1: *Alice **(A)** and Bob **(B)** are equipped with fixed and portable trusted cells, acquiring data from several data sources, synchronizing with their encrypted personal digital space on the cloud. Charlie **(C)** is travelling around the world and can securely access all his data from any (unsecure) terminal thanks to his portable trusted cell. All users equipped with trusted cells can securely share their encrypted data through the cloud.*

among cells and (3) participate to distributed computations (e.g., store intermediate results), provided this participation can be guaranteed harmless by security checks implemented at the trusted cells side.

Figure 1 illustrates how trusted cells and the untrusted infrastructure can collaborate to implement scenarios meeting the privacy requirements stated above.

Threat model: In our context, the primary adversary is the infrastructure. The infrastructure may deviate from the protocols it is expected to implement with the objective to breach the confidentiality of the outsourced data. Integrity attacks (e.g., on data related to access control) must also be deterred since they may lead to subsequent confidentiality leaks. The infrastructure is assumed trying to cheat only if it cannot be convicted as an adversary by any trusted cell. Indeed, revealing a data leak (or a denial of service) in a public place would cause irreversible political/financial/legal damage to the service provider. Such adversaries are usually called malicious adversary having weakly malicious intents [14]. Trusted cells are themselves presumably trusted. However, even secure hardware can be breached, though at very high cost, so that one cannot exclude with certainty that a very small number of trusted cells be compromised. Hence, the trusted cells' cryptographic secrets must be managed in such a way that a successful attack on a (small set of) trusted cell cannot degenerate in breaking class attack. This is of utmost importance considering also that an individual succeeding in breaking her trusted cell could have effective malicious intents.

4. REQUIREMENTS AND CHALLENGES

We identify five major requirements for the user to actually control how the data entering her personal digital space is collected, protected, shared and finally used.

Controlled collection of sensed data: The targeted user(s) should be the unique recipient(s) of raw sensed data and would accept externalizing only aggregates by opting in/out for selected applications/services.

At home, the power meter continuously pushes raw measurements to Alice's and Bob's trusted cell gateway, while a certified aggregated time series is sent to the power supplier company and aggregates for a social game are pushed to the Cloud every day. Similarly, the tracking box installed on Alice's car is a trusted cell delivering aggregated GPS data to her insurer and raw data to her trusted cell smartphone that she will synchronize with her personal space for further use when back home. Hence, adding a trusted cell to a sensor, allows defining e.g., the frequency and or precision of the data that should be externalized, thus leading to a *trusted source* both for the user (in terms of privacy preservation) and the provider (in terms of certification of the output data).

Related challenges: Co-design is a primary issue to allow the definition of affordable sensor-based trusted cells. Low-cost is indeed a prerequisite to the generalization of trusted sources, capable of securely filtering and aggregating stream-based spatio-temporal data with tiny hardware resources. Some trusted sources being weakly connected to the Internet; asynchrony problems must also be addressed. Finally, the combination of data streams from multiple sources, each being separately harmless, may generate new privacy risks that must be carefully tackled.

Secure private store: All data must be made highly available, resilient to failure and protected against confidentiality and integrity attacks. Accessing this data from any terminal, including those outside the user's ownership sphere (e.g., internet café), should leave no trace of the access.

Cryptographic techniques (i.e., encryption, hashing, signatures) are used to protect trusted cell's data, keeping cryptographic keys in their tamper-resistant memory. The data is then stored in the Cloud and potentially cached in the trusted cell local mass storage. At a minimum, trusted cells keep locally extended metadata: access information, indexes, keywords, and cryptographic keys. Metadata should be sufficient to allow performing queries before accessing the Cloud to retrieve the data of interest. Cryptographic keys never leave the trusted cells tamper-resistant memory. Hence a trusted cell can be used to get securely data from any (untrusted) terminal it is connected with.

Related challenges: Designing an intuitive HCI for managing this bunch of heterogeneous personal data (data modeling, data integration, querying) is a major challenge. Besides, a significant amount of data and metadata is likely to be embedded in some trusted cells and may need to be queried efficiently. While it does not seem a major issue in powerful trusted cells (e.g., a smart phone), it appears much more challenging when facing low-end hardware devices like secure tokens (e.g., a microcontroller with tiny RAM, connected to NAND Flash chips or SD cards, possibly with energy consumption constraints). Whatever their complexity, trusted cells should also be designed to support self-tuning, self-diagnosis and self-healing to minimize the management burden put on the trusted cell owner.

Secure sharing: The user can decide to keep her data private or share it with other users or group of users under certain conditions (e.g., time, location). Under which model the access control policies are actually defined is an open issue, but not the main concern of this paper. However, we insist that the user must get a proof of legitimacy for the credentials exposed by the participants of a data exchange and must trust the evaluation of the exchange conditions (if any).

Practically, sharing data means sharing the associated metadata (so that the recipient user can get the referenced data in the Cloud), the cryptographic keys (so that her trusted cell can decrypt them) and the sticky policy (so that her trusted cell can enforce the expected access control rules). Hence, thanks to its security properties, including the protection against illegitimate actions of the recipient user, the recipient trusted cell can enforce all the conditions appearing in the access control rules (user's credential, contextual conditions).

Related challenges: Again, an intuitive HCI for defining the access control policies and simple modes of operation must be devised. The trusted cells themselves may be a source of simplification (e.g., integration of biometric sensors to automatically authenticate users, automatic production of certified credentials safely computed on a trusted cell, definition of default policies by trusted third parties – e.g., citizen associations – which could be automatically selected depending on a computed individual's profile). Also, secret management is at the heart of any sharing protocol between trusted cells (i.e., at this level a secret is a cryptographic key) and must be carefully designed (e.g., class-breaking attacks must be prevented, master secrets must be restorable in case of crash/loss of a trusted cell).

Secure usage and accountability: Usage control usually refers to $UCON_{ABC}$ [8]: obligations (actions a subject must take before or while it holds a right), conditions (environmental or system-oriented decision factors), and mutability (decisions based on previous usage)[6]. Again, defining appropriate usage control policies for trusted cell applications is an open issue.

Similarly to access control rules, usage control rules can be implemented as sticky policies so that they are made cryptographically inseparable from the data to be protected. Hence usage control rules will be enforced by any trusted cell downloading data and cannot be bypassed by the recipient user. Regarding accountability, the recipient trusted cell can maintain an audit log, encrypt it and push it on the Cloud to the destination of the originator trusted cell.

Related challenges: Many challenges are common with secure sharing. However, trusted cells hold the promise of new usages and new usage controls. For example, trusted cells could be parameterized so that any personal data produced by a trusted source linked to an individual A and referencing individual B be submitted for approbation to B's trusted cell before being integrated to A's digital space.

Shared Commons: Privacy has also a collective dimension in the sense that preserving one's privacy should not hinder societal benefits (e.g., census, epidemiologic releases, global queries). A trusted cell user is thus expected to participate to global treatments assuming her data suffers appropriate transformations (e.g, anonymization, output perturbation) depending on the trustworthiness of the recipient(s) and the expected usage of the data/query. When data needs to be transformed before being delivered, the recipient trusted cell implements the transformation on its own if possible (e.g., filtering, local data perturbation) or in collaboration with other trusted cells if the transformation requires a collective action (e.g., anonymization, global data perturbation). In the latter case, the computation may be implemented in a pure Secure Multi-Party fashion or may require the participation of the untrusted infrastructure (e.g., to store intermediate results).

[6] For instance, a photo could be accessed ten times (mutability), in the course of 2012 (condition), informing the owner of the precise access date (obligation).

Related challenges: Such large scale computations may lead to atypical distributed protocols combining security and performance requirements in an asymmetric context made on one side of a very large number of highly secure, low power and weakly available trusted cells and on the other side of a highly powerful, highly available but untrusted infrastructure. Hence, the trusted cells architecture can be seen as a massive untrusted interconnection of trusted co-processors.

5. CONCLUSION

We proposed the trusted cell architecture, a vision reconciling individual's privacy with innovative acquisition and sharing of personal data. This vision is based on the premise of ubiquitous and open secure hardware. Trusted cells enforce access and usage control at the edges of the Internet, and thus constitute a sea change with respect to personal data management. This vision undoubtedly opens a set of exciting challenges that must be explored by the database community.

6. ACKNOWLEDGEMENTS

This work has been partially funded by the French ANR KISS project.

7. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu: Hippocratic Databases. VLDB 2002: 143-154

[2] T. Allard et al.: Secure Personal Data Servers: a Vision Paper. PVLDB 3(1): 25-35 (2010)

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

[4] FreedomBox: http://freedomboxfoundation.org/.

[5] Global Platform Device Technology. Trusted Execution Environment Internal API Specification. Version 1.0. December 2011.

[6] S. Katzenbeisser and K. Kursawe, Privacy and Security in Smart Energy Grids, *Dagstuhl Seminar 1151*, 2011

[7] H. Lam. A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signatures,. IEEE Transactions on Consumer Electronics, 2007.

[8] A. Narayanan, V. Toubiana, S. Barocas, H. Nissenbaum, D. Boneh: A Critical Look at Decentralized Personal Data Architectures CoRR abs/1202.4503: (2012)

[9] H. Nissenbaum, Privacy in context: Technology, policy, and the integrity of social life,"*Stanford Law Books*, 2010.

[10] J. Park and R. Sandhu, "The $UCON_{ABC}$ usage control model," *ACM Trans Inormationf System Security*, vol. 7, no. 1, pp. 128-174, 2004.

[11] A. Pentland et al. Personal Data: The Emergence of a New Asset Class. World Economic Forum. January 2011.

[12] S. Petronio, Unpacking the paradoxes of privacy in CMC relationships: The challenges of blogging and relational communication on the internet, *In Computer-mediated communication in Personal Relationships*, 2011.

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

[14] N. Zhang, W. Zhao: Distributed privacy preserving information sharing. VLDB 2005.

Annexe C.

Folk-IS: Opportunistic Data Services in Least Developed Countries

Nicolas Anciaux, Luc Bouganim, Thierry Delot, Sergio Ilarri, Leïla Kloul,
Nathalie Mitton, Philippe Pucheral

Proceedings of the VLDB Endowment (PVLDB), Volume 7(5), pp. 425-428, 2014.

Folk-IS: Opportunistic Data Services in Least Developed Countries

N. Anciaux[1, 2], L. Bouganim[1, 2], T. Delot[3,1], S. Ilarri[4], L. Kloul[2], N. Mitton[1], P. Pucheral[1, 2]

[1] INRIA, France *fname.lname@inria.fr*
[2] PRISM, UVSQ, France *fname.lname@prism.uvsq.fr*
[3] LAMIH, UVHC, France *Thierry.Delot@inria.fr*
[4] Univ. of Zaragoza, Spain *silarri@unizar.es*

ABSTRACT

According to a wide range of studies, IT should become a key facilitator in establishing primary education, reducing mortality and supporting commercial initiatives in Least Developed Countries (LDCs). The main barrier to the development of IT services in these regions is not only the lack of communication facilities, but also the lack of consistent information systems, security procedures, economic and legal support, as well as political commitment. In this paper, we propose the vision of an infrastructureless data platform well suited for the development of innovative IT services in LDCs. We propose a participatory approach, where each individual implements a small subset of a complete information system thanks to highly secure, portable and low-cost personal devices as well as opportunistic networking, without the need of any form of infrastructure. We review the technical challenges that are specific to this approach.

1. INTRODUCTION

As a citizen of a developed country, Alice receives most personal data electronically (e.g., salary forms, banking statements, medical records) and stores them in the cloud where service providers deliver a myriad of digital services in the context of healthcare, education, and business. But what if Alice lives on the opposite side of the digital divide? Least Developed Countries (LDCs) are those countries that meet United Nations (UN) criteria in terms of poverty, human resource weaknesses and economic vulnerability. These countries definitely lack IT infrastructures. Nevertheless, many reports (e.g., [7, 9, 12]) emphasize that IT is called to play a catalytic role in LDCs, helping to establish primary education, reduce mortality and boost individual commercial initiatives.

While 60% of the population in LDCs is already covered by a mobile cellular signal, only 0.5% has a mobile broadband subscription and a 3G service is offered only in at most 25% of the LDCs, very often at a prohibitive cost [9]. Hence, mobile phones in these areas are primarily feature phones used for voice and SMSs, not for data-driven applications. According to many analysts, this situation cannot evolve rapidly due to a combination of technical, economical and organizational barriers [9]. A few pioneering proposals [11, 14] tried to overcome these limitations by mounting mobile access points on vehicles to transport data from one place to another. Google project Loon is a more ambitious attempt to connect people in rural and remote areas and bring people back online after disasters thanks to a network of high altitude balloons. However, it is unclear if such a network infrastructure is durable, since current balloons can only function for a few weeks. For its part, Internet.org promotes low cost wireless handsets, compression of Web pages, and data caching on the edge of the network, to reduce data transfer. These different initiatives demonstrate the growing interest, as well as the high difficulty, to integrate LDCs in global IT networks.

But low connectivity is not the ultimate and unique barrier. Several solutions have considered the use of mobile phones (and text-messages data transfer) to address specific issues like tracking vaccine cold chains [6], improving agriculture [13], or easing administrative procedures [10]. Despite their undisputed interest, these initiatives remain confined to specific applications and their generalization faces many obstacles: lack of global information system infrastructures, maintenance of the system, security concerns, etc. Hence, we consider as characteristics of LDCs not only the lack of communication facilities, but also the lack of consistent information systems, security procedures, economic and legal support, as well as political commitment.

According to Non-Governmental Organizations (NGOs), four main requirements must be met to build a practical technical solution: (1) **privacy protection**: a major prerequisite due to local opaque practices and the lack of any security infrastructure (coercive laws, secured servers, trusted authorities, etc.), leading to a self-enforcement of privacy principles; (2) **immediate personal benefit**: that should be provided to each user because of the lack of strong economical or political incentives to impose the solution; (3) **self-sufficiency**: the solution must not rely on a hypothetic improvement of the existing software and hardware infrastructure; and (4) **very low deployment cost**: the usual scale being a few dollars per user. Besides this, **users' empowerment** is crucial to make the solution sustainable in LDCs, and its maintenance should ideally generate a **source of revenue for new local jobs**.

In [1], we proposed the vision of *trusted cells*, a data platform for personal data services where the shared infrastructure (e.g., the cloud) is untrusted, while personal devices (e.g., smart phones) are trusted execution environments. In this paper, we revisit this vision to the context of LDCs. We propose, *Folk-enabled Information System (Folk-IS),* a new paradigm based on a fully decentralized and participatory approach, where each individual implements a small subset of a complete information system without the need for infrastructure. As trusted cells, Folk-IS builds upon the emergence of highly secure, portable, low-cost storage and computing devices, called hereafter *Smart Tokens*. Here, however, the focus is on the low-cost of ownership, deployment and maintenance, and on the absence of a networked infrastructure. With Folk-IS, and thanks to smart tokens, people will transparently and opportunistically perform data management and networking tasks as they physically move, so that IT services are truly delivered by the crowd.

We do not argue that Folk-IS is the ultimate solution. The future of IT in LDC will probably be multiform, the problem being important and complex enough to leave room for complementary initiatives. Folk-IS has the salient characteristics to enable a

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/3.0/. Obtain permission prior to any use beyond those covered by the license. Contact copyright holder by emailing info@vldb.org. Articles from this volume were invited to present their results at the 40th International Conference on Very Large Data Bases, September 1st - 5th 2014, Hangzhou, China.
Proceedings of the VLDB Endowment, Vol. 7, No. 5
Copyright 2014 VLDB Endowment 2150-8097/14/01

smooth and incremental deployment of an information system in a purely infrastructureless context while taking advantage of existing elements of infrastructure, if any, to improve its own behavior. This paper shows that this paradigm technically makes sense, conforms to the requirements mentioned above, and opens important and exciting research challenges.

2. ARCHITECTURE AND SCENARIOS

The main Folk-IS elements are the following ones:

Smart Tokens: Smart Tokens appear today in various form factors and may have different hardware characteristics. In the Folk-IS context, Smart Tokens embed at least: *(1)* enough stable storage to host the complete digital environment of its holder, *(2)* enough tamper-resistant computing resources to run a server managing the data and enforcing access control rules, and *(3)* a biometric sensor to authenticate users (e.g., a fingerprint reader, well adapted to illiterate people). Smart Tokens require also input/output capabilities to interact with users and communication facilities (e.g., short-range communications) to exchange messages.

Shared devices: To meet low-cost requirements, a Smart Token may inherit part of its functionalities from the terminals it connects to. While storage and security resources cannot be delegated to shared devices without compromising data availability and introducing the risk of class-breaking attacks, the I/O capabilities of shared devices (e.g., screen, keyboard, etc.) could be naturally used. Similarly, shared devices can act as a relay when connected to the Smart Tokens, compensating their lack of communication facilities. Shared devices are either made publicly available in specific places or owned by local workers.

Figure 1 shows concrete examples of Smart Tokens and shared devices (e.g., off-the-shelf PCs and tablets or specific devices containing only validated software and hardware). The *Basic* Smart Token embeds only mandatory resources, namely a Flash stable storage, a Secure microcontroller (SMCU) [4], and a fingerprint reader. It achieves the lowest cost and the highest robustness at the expense of needing a shared device to be used. The left-end side of the figure shows a real product that we used in a field experiment [2], provided by Gemalto for a few dollars. The *Self-powered* Smart Token is a bit more expensive but includes also a speaker and a microphone for basic user interactions, a battery, solar cells, and wireless communication to enable message exchanges without the need of shared devices.

Folk-IS Personal Node (Folk-node): a Folk-node is associated with each individual and refers to the combination of a Smart Token and the embedded software components required to manage the holder's data, act as a network node, and enforce the security of the whole system. Based on Folk-nodes, we can set up an ad hoc delay-tolerant-network [5], called in the following *Folk-enabled Network (Folk-Net),* enabling the transfer of messages among Folk-nodes, and from Folk-nodes to Internet access points, without requiring any existing communication infrastructure. In Folk-Net, communications are opportunistic by nature, and follow a carry-and-forward protocol.

Folk-enabled Information System (Folk-IS): based on the existence of Folk-nodes, we can devise a system implementing the three main functions of an information system as follows:

- *Communication management:* by carrying and routing data, each Folk-node acts as an active node in the Folk-Net. Assuming that each Folk-node maintains a history of its moves (e.g., by gathering the GPS coordinates of all the devices it connects to), it can forward messages to encountered Folk-nodes whose moving profile best matches the recipient's location, thus providing a much better resource utilization of the Folk-Net than a basic flooding protocol.

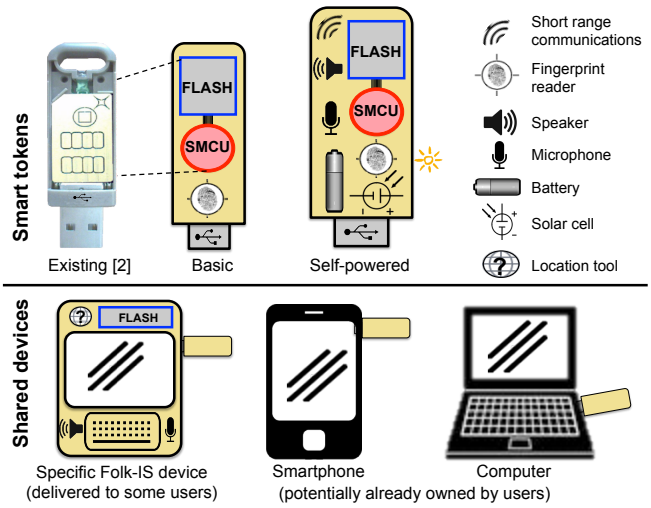


Figure 1. Smart Tokens and shared devices

- *Data management:* each individual centralizes in his Folk-node all his personal data (e.g., medical records, administrative documents, credentials). Each time the holder interacts with a data source (e.g., by physically meeting a data provider, like a doctor, or by receiving a document through the Folk-Net), the application simply inserts these data in a local Folk-node database. From the information system viewpoint, the global database is the "union" of all these local databases. Each Folk-node is assumed to participate in common services, like processing global queries (broadcasted through the Folk-Net) and ensuring data durability (thanks to data replication among Folk-nodes).
- *Access control & authentication:* each Folk-node controls the access to the data it hosts and strongly authenticates the requesters. Performing this control on the user's side is the ultimate solution to increase the holder's confidence, enforce his consent, and minimize the benefit/cost ratio of an attack. Indeed, the complexity of attacking the system is increased by the Smart Token tamper-resistance and by the obligation to be physically in contact with it to attack it. In parallel, the benefit of the attack is limited to disclosing/corrupting the data of a single individual. Note that the holder himself must authenticate and does not have all the privileges on his own Folk-node (e.g., he cannot tamper his own medical data to prescribe himself new drugs). In addition, messages carried by a Smart Token or replicas from other individuals stored locally are encrypted with keys never available to that Smart Token.

Hence, the complete system (data storage and network facilities) progressively deploys itself as Folk-nodes are distributed to the participants. Folk-IS is by construction highly redundant and robust, it does not require any central administration, and its global cost is proportional to the scale of the targeted population. As discussed next, some Folk-IS functionalities could be delegated to specific workers (e.g., people renting terminals or acting as postmen) in order to improve the quality of service while creating a new local economic model (e.g., like people renting their cell phones in LDCs [3]). If the infrastructure is partially available, the Folk-IS quality of service also improves accordingly, decreasing the network latency by shortening the route to the nearest Internet access point.

Figure 2 illustrates the Folk-IS mode of operation. It shows two rural communities, residents (black icons with letters) and their possible moves (grey icons), a *school* and a *first aid* room used by

residents from both communities, and an Internet access point. Data exchanges through terminals are represented by dashed lines (e.g., the Folk-node of person A is used from the terminal of nurse N in the infirmary), and short-range data exchanges are represented by pink halos (e.g., between A and B in *Community 2*). E serves as a *netman* (i.e., a postman conveying digital messages) and carries data when travelling from place to place (e.g., the school, the infirmary, and the Internet access point) thanks to his Folk-node. Different routing paths to transmit a message from A to the Internet are represented, e.g., the path ABTE: A → B → T (teacher) →E (netman) → Internet. These paths benefit from the physical mobility of people and their devices.

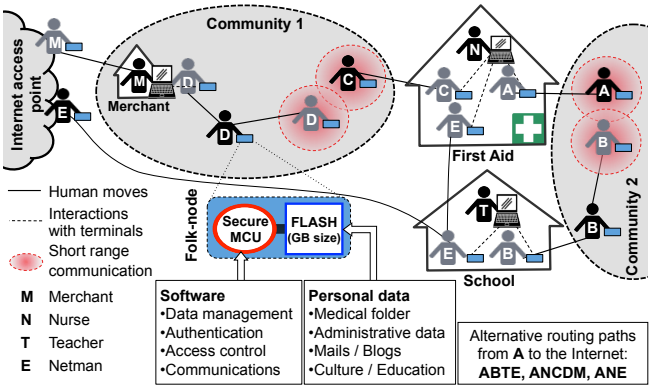


Figure 2. Folk-IS: mode of operation

Based on this mode of operation, several concrete scenarios can be envisioned. For illustration, we show an example below.

Healthcare scenario: In most LDCs, healthcare is provided by local nurses/doctors working in first aid rooms or intervening in rural communities during drought periods or health programs. They only possess very basic medicines and equipment. Needless to say, they do not benefit from Electronic Health Records, which are already highly difficult to organize in developed countries. Even paper-based medical records are too complex to maintain because many people have no identity document to link them to their records and people move according to seasons and drought periods. By using a Folk-node, each patient owns his complete and up-to-date medical folder. Without any Internet connection, a nurse can access the fingerprint-authenticated patient's medical folder, and append diagnosis and treatment information. She may request medical advice from a distant clinic, by transmitting documents (e.g., a picture of the patient's injury) from the patient's Folk-node through the Folk-Net. She can notify the patient a few days later, still through the Folk-Net, in case a serious problem has been detected. Global queries can also be broadcasted through the Folk-Net to conduct epidemiological studies or maintain health indicators on populations unreachable so far. Hence, each Folk-node acts as a smart Personally Controlled Electronic Health Records.

Generalization: Many usual scenarios can be transposed to rural communities thanks to Folk-IS, provided that they accommodate high-latency data exchanges: scholar folders for children, e-administration procedures (e.g., drought warnings, recording births and deaths), personal data applications (e.g., email, social or networking services), etc. Folk-IS also paves the way to new practices like sharing pieces of cultural heritage (e.g., traditional songs, beliefs and religion, etc.) with the outside world. This may generate a new economic model where every actor is paid according to a specific Digital Rights Management (DRM) model. Similarly, people playing the role of a *netman* can be paid according to the volume of transported data and distance covered. In both cases, the exact contribution of each actor can be recorded and certified by his tamper-resistant Folk-node.

3. DATABASES CHALLENGES

Several challenges of Folk-IS are at the crossroad of Computer Science, Economics and Social Science. Issues linked to economic models are crucial for the adoption of the system, and the problems linked to the low level of education, lack of training, specific beliefs and norms of potential users need to be addressed. In the following, we focus solely on database challenges.

(1) Co-design of the embedded data management engine. A primary role of a Folk-node is to ensure secure storage and sharing of all data forming the holder's digital environment: user authentication, secure data storage, query evaluation, and access control enforcement. Designing these components is very challenging due to the inherent hardware constraints of smart tokens, namely the tiny RAM of the secure microcontroller, the NAND flash memory that badly supports random writes, and the need to protect the confidentiality and integrity of the stored data. These constraints lead to contradictory objectives: executing queries with a tiny RAM entails indexing massively the embedded database, while index updates generate fine-grained random writes, and then unacceptable NAND flash write costs and cryptographic overhead. In addition, small hardware variations (i.e., varying the precise characteristics of each element) may greatly impact their cost, energy consumption, and performance. Existing embedded and lightweight DBMS products target devices far more powerful than smart tokens, and state-of-the-art research solutions do not explicitly tackle these issues. Initial work has been done to handle classical smart token constraints [2], but the challenge here is to co-design and implement the best storage, indexing and query engine to reduce the overall cost of the device and its energy consumption, and provide a high level of flexibility.

(2) Folk-Net routing and mobility prediction protocols. The Folk-Net routing protocol aims at delivering the data exchanged among Folk-nodes, workers' devices, and remote Web servers. Given the lack of infrastructure and the highly dynamic nature of Folk-Net, the routing protocol has to rely on opportunistic communications. Thus, the choice of the best candidate(s) to carry the encrypted data towards their destination is a primary concern. Another concern is the energy limitations, which have to be taken into account in the design of the routing scheme. Geographic routing protocols are interesting candidates to avoid flooding the network, as they do not need to maintain routing tables and work nearly stateless. However, existing georouting protocols can hardly be considered because they rely on the exact locations of individual nodes. In our context, the location of the device must be approximated from interactions with a small subset of localized Folk-nodes, shared devices, or fixed nodes. New routing protocols mixing geolocation approximation, mobility prediction strategies, and social interactions (e.g., [8]), definitely deserve to be studied to select the best "data mules".

(3) Application, data model and access control deployment. Folk-IS enables many important applications, like healthcare and e-administration, some of them novel by their purpose (e.g., sharing pieces of cultural heritage) or by their target (e.g., epidemiological studies on populations unreachable so far). The infrastructureless context introduces new challenges with respect to application deployment, unified data modeling, identity verification, access control, and user's consent. Typically, data standardization, central application stores, or central authorities

identifying people, delivering certificates and enforcing access control rules, cannot be assumed. Conversely, a salient feature of Folk-IS is the ability to push the control at the edge of the network, that is to say (1) within each tamper-resistant Smart Token, and (2) through face-to-face interactions between users (a de-facto user's consent). This may enable the reestablishment of local spheres of trust (e.g., NGO producing applications, data models, home-made identification information, access control policies, and pushing them at the Folk-node level through Folk-Net). This paves the way to a semi-decentralized way of managing and deploying applications, each Folk-node guaranteeing that (i) data produced by a given organization will not leak outside that organization, and (ii) the data owners' privacy is always respected.

(4) Evaluation of global queries on a population of Folk-nodes. Traditional techniques used to process queries in distributed databases or in peer-to-peer networks are not suitable, as they assume a good knowledge about the data location in the network. Here, data will be dynamically distributed over a network of nodes that may be accessible or not at a certain moment, and the carrier of a replica of a data item may change at any time. Moreover, we cannot assume that all data relevant to a given query will always be available and retrievable in a reasonable time period. So, we should assume the possibility of approximate answers, decide how to identify the relevant data sources without overloading the network (e.g., based on spatial conditions), how to route queries and results to their recipients (while preserving autonomy) using the physical mobility of nodes, and how to determine when a query and its associated routing tasks have to be finished. Moreover, new types of queries could be of interest in this context, requiring new query processing techniques; for example, reachability queries [15] could be used to study the possibility of propagation of a disease by analyzing past trajectories.

(5) Structuration and calibration of the Folk-IS architecture. Folk-IS is built on a large number of highly-secure but seldom-available Folk-nodes (basic or self-powered) on the one side, and on less secure but more accessible and powerful shared devices on the other side. This unusual asymmetric architecture requires deeply rethinking the overall organization of an information system. This means distributing software resources on the hardware elements of the architecture, such that local and global applications can run and provide results with acceptable performance, security and resiliency. This also means associating different responsibilities to different devices (e.g., *super-nodes*), specific roles to humans (e.g., *netmen* to reduce latency), and designing new distributed protocols for global functionalities like query evaluation or data durability. For the latter, data replication is required because Folk-nodes may be lost, stolen, broken, etc. Data could be replicated based on the user mobility profiles. Confidentiality of replicas can be achieved through encryption, leading to the problem of choosing the most adequate set of Folk-nodes to hold a copy of the keys or of key shares (e.g., based on trust or similar mobility profiles). Finally, new simulation models are needed to calibrate IT resources according to the target applications and the local habits of residents. The objective is to determine suitable network topologies (given a certain density of Folk-nodes, communication frequency, etc.) with viable incremental deployments. Those simulation models will also help quantify the expected gain in terms of social and economic sustainability, which is key for their long-term adoption

4. CONCLUSION

As mentioned in [3], time has come for research works, not only commercial initiatives, addressing the expectations of 80% of the population living outside developed countries. The *Folk-IS* paradigm combines low-cost secure devices, embedded software components and opportunistic communications, to meet fundamental requirements of LDCs. The promise of the solution is to guarantee high privacy standards at very low cost (a few dollars per user), granting users access to innovative personal services with the ability to benefit from future infrastructure improvements. The maintenance and performance improvement of the system can be a source of empowerment with new local jobs, crucial to make the solution sustainable. We have presented an initial architecture and identified important challenges, which pave the way to exciting future works for our research community.

Acknowledgment: We warmly thank Léo Dayan, scientific director of APREIS NGO and of the Nomadic World University for Sustainable Development, and Pascale Pollack, director of E-NEXUS NGO, for precious discussions and feedback on the IT requirements for LDCs.

5. REFERENCES

[1] N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, I. S. Popa, P. Pucheral, “Trusted Cells: A Sea Change for Personal Data Services”. CIDR, 2013.

[2] N. Anciaux, L. Bouganim, Y. Guo, P. Pucheral, J.-J. Vandewalle, S. Yin, “Pluggable Personal Data Servers”, ACM SIGMOD, 2010.

[3] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, K. Fall. “The Case for Technology in Developing Regions”, Computer 38(6), 2005.

[4] D. Bursky, “Secure Microcontrollers Keep Data Safe”, PRN Engineering Services, http://tinyurl.com/secureMCU, 2012.

[5] Y. Cao, Z. Sun, “Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges”, Communications Surveys & Tutorials, IEEE 15(2), 2013.

[6] R. Chaudhri, G. Borriello,R. J. Anderson, “Monitoring Vaccine Cold Chains in Developing Countries”, IEEE PerCom, 11(3), 2012.

[7] R. Coceres, E. M. Belding, T. S. Parikh, L. Subramanian, “Information and Communication Technologies for Development - Guest Editors' Introduction”, IEEE PerCom, 11(3), 2012.

[8] L. Gao, M. Li, A. Bonti, W. Zhou, and S. Yu, “Multidimensional Routing Protocol in Human-Associated Delay-Tolerant Networks”, IEEE TMC, 12(11), 2013

[9] ITU, “The Role of ICT in Advancing Growth in Least Developed Countries – Trends, Challenges and Opportunities”, 2011.

[10] V. Ndou, “E-Government for Developing Countries: Opportunities and Challenges”, EJISDC volume 18, 2004.

[11] A. Pentland, R. Fletcher, A. Hasson, “DakNet: Rethinking Connectivity in Developing Nations”, IEEE Computer, 37(1), 2004.

[12] G. Rossi, S. Murugesan, N. Godbole, “IT in Emerging Markets”, IT Professional 14(4), 2012.

[13] H. Sahilu, A. Villafiorita, K. Weldemariam, M. Belachew, A. Zewge, “Designing Distributed Agricultural Information Services for Developing Countries”, ACM DEV, 2012.

[14] A. Seth, D. Kroeker, M. A. Zaharia, S. Guo, S. Keshav, “Low-Cost Communication for Rural Internet Kiosks Using Mechanical Backhaul”, ACM MOBICOM, 2006.

[15] H. Shirani-Mehr, F. B. Kashani, C. Shahabi, “Efficient Reachability Query Evaluation in Large Spatiotemporal Contact Datasets”, PVLDB, 5(9), 2012.

Annexe D.

MILo-DB: a personal, secure and portable database machine

Nicolas Anciaux, Luc Bouganim, Philippe Pucheral, Yanli Guo, Lionel Le Folgoc, Shaoyi Yin

Distributed and Parallel Databases (DAPD), Volume 32(1), pp. 37-63, 2014.

Distrib Parallel Databases
DOI 10.1007/s10619-012-7119-x

MILo-DB: a personal, secure and portable database machine

Nicolas Anciaux · Luc Bouganim · Philippe Pucheral · Yanli Guo · Lionel Le Folgoc · Shaoyi Yin

© Springer Science+Business Media New York 2013

Abstract Mass-storage secure portable tokens are emerging and provide a real breakthrough in the management of sensitive data. They can embed personal data and/or metadata referencing documents stored encrypted in the Cloud and can manage them under holder's control. Mass on-board storage requires efficient embedded database techniques. These techniques are however very challenging to design due to a combination of conflicting NAND Flash constraints and scarce RAM constraint, disqualifying known state of the art solutions. To tackle this challenge, we proposes a log-only based storage organization and an appropriate indexing scheme, which (1) produce only sequential writes compatible with the Flash constraints and (2) consume a tiny amount of RAM, independent of the database size. We show the effectiveness of this approach through a comprehensive performance study.

Communicated by Elena Ferrari.

N. Anciaux (✉) · L. Bouganim · P. Pucheral · Y. Guo · L. Le Folgoc
INRIA Paris-Rocquencourt, Le Chesnay, France
e-mail: Nicolas.Anciaux@inria.fr

L. Bouganim
e-mail: Luc.Bouganim@inria.fr

P. Pucheral
e-mail: Philippe.Pucheral@inria.fr

Y. Guo
e-mail: Yanli.Guo@inria.fr

L. Le Folgoc
e-mail: Lionel.LeFolgoc@inria.fr

N. Anciaux · L. Bouganim · P. Pucheral · Y. Guo · L. Le Folgoc
PRISM Laboratory, University of Versailles Saint-Quentin-En-Yvelines, Versailles, France

S. Yin
University of Cergy Pontoise, Cergy-Pontoise, France
e-mail: Shaoyi.Yin@u-cergy.fr

Published online: 16 January 2013

Keywords Embedded database · Secure and privacy aware data management · Secure chip · Flash memory · Tiny RAM · Log-only database structure

1 Introduction

As any citizen today, Alice receives salary forms, invoices, banking statements, etc., through the Internet. In her everyday life, she continuously interacts with several electronic devices (at home, at work, at hospital, while driving or shopping, taking photos, etc.) acquiring and producing large volumes of personal data. Alice would like to store this mass of data securely, share it with friends and relatives and benefit from new, powerful, user-centric applications (e.g., budget optimization, pay-per-use, health supervision, e-administration procedures and others Personal Information Management applications). Storing this data in a well organized, structured and queryable *personal database* [3, 17] is mandatory to take full advantage of these applications. Recently, KuppingerCole,[1] a leading security analyst company promotes the idea of a "*Life Management Platform*", a "*new approach for privacy-aware sharing of sensitive information, without the risk of losing control of that information*". Several projects and startups (e.g., QiY.com, Personal.com, Project VRM[2]) are pursuing this same objective.

But are there effective technological solutions matching this laudable objective? Any solution where Alice's data is gathered on her own computer would be very weak in terms of availability, fault tolerance and privacy protection against various forms of attacks. Solutions based on third party storage providers (e.g., in the Cloud) nicely answer the availability and fault tolerance requirements but the price to pay for Alice is loosing the control on her data.[3] This fear is fed by recurrent news on privacy violations resulting from negligence, abusive use and internal or external attacks (see e.g., DataLossDB.org). Should Alice follow the FreedomBox initiative [25] which promotes the idea of a small and cheap data server that each individual can plug on her Internet gateway? She will avoid any risk linked to the centralization of her data on remote servers. But what are the real security and availability guarantees provided by her plug computer?

The Personal Data Server vision [3] promotes an idea similar to FreedomBox, that is providing a fully decentralized infrastructure where personal data remains under holder's control, but with stronger guarantees. It builds upon the emergence of new devices combining the tamper resistance of a secure microcontroller [12] with the storage capacity of NAND Flash chips. Tamper-resistance provides tangible security guarantees missing to traditional plug computers. Availability is provided by replicating data in remote encrypted archive. These devices have different form factors (e.g.,

[1] http://www.kuppingercole.com/.

[2] http://cyber.law.harvard.edu/projectvrm/Main_Page.

[3] A recent Microsoft survey states that "*58 percent of the public and 86 percent of business leaders are excited about the possibilities of cloud computing. But more than 90 percent of them are worried about security and privacy of their data as it rests in the cloud*" http://news.cnet.com/8301-1009_3-10437844-83.html.

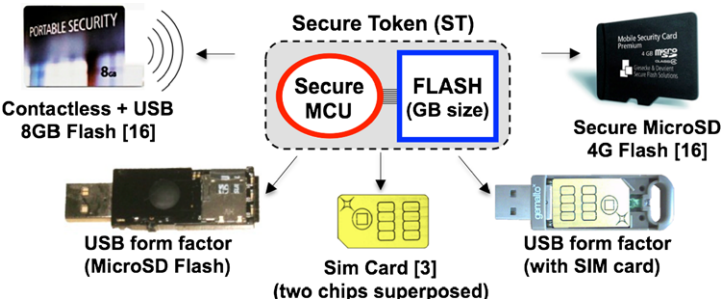


Fig. 1 Different form factors for Secure Tokens (STs)

SIM card, USB token, Secure MicroSD [18]) and names (e.g., Personal Portable Security Device [20], Smart USB Token [16], Portable Security Token [18] or Secure Portable Token [3]).

Despite this diversity (see Fig. 1), Secure Tokens (STs) share similar characteristics (low-power, cheap, highly portable, highly secure), holding the promise of a real breakthrough in the management of personal data.

Capitalizing on the Personal Data Server vision, we could devise an effective secure *Life Management Platform* where personal data is stored either locally (in the ST Flash memory) or remotely (e.g., in the Cloud). In the latter case, ST manages only metadata (description and location attributes, keywords, encryption keys, etc.) of encrypted documents stored remotely. Document sharing can be obtained by sharing the corresponding metadata under the ST holder's control, thereby providing ultimate security and privacy to cloud data [1]. The amount of data/metadata to be managed by the ST can be rather huge, embracing potentially the complete digital history of an individual. The heart of the Personal Data Server vision is then a ST turned into a personal database machine able to manage and share personal data under the control of its owner, with strong security guarantees inherited from tamper resistant hardware. More precisely, turning this vision into reality requires solving a core technical challenge, that is designing a DBMS engine embedded in a ST providing a least basic storage, indexing and query facilities with acceptable performance. This challenge holds whatever be the external resources the ST may be connected to (network, cloud, local host, etc.) since this is key to securely manage and share personal data.

Why is this challenging? STs provide unquestionable benefits (tamper-resistance, robustness, portability, low cost, low energy consumption), but have inherent hardware constraints. A ST is primarily made of a secure microcontroller (SMCU) connected to an external (i.e., unsecure) mass storage NAND Flash memory. Relatively speaking, SMCU have a powerful architecture (e.g., a 32-bit RISC processor, clocked at about 50 MHz, a cryptographic co-processor and internal stable storage up to 1 MB to store the embedded code and sensitive metadata). Actually, the main ST constraints are twofold. First, SMCU has a tiny RAM (at most 64 KB today). According to SMCU manufacturers, RAM will unfortunately remain a scarce resource in the foreseeable future due to its very poor density. Indeed, the smaller the silicon die, the more difficult it is to snoop or tamper with its processing. Hence, the RAM capacity increases much slowly than the stable storage capacity, worsening the

ratio RAM/stable storage over time. Second, the NAND Flash stable store (usually connected by a bus to the SMCU) badly support random writes and cannot benefit from the SMCU tamper resistance. This leads to contradictory objectives: executing queries with acceptable performance with a tiny RAM entails indexing massively the database, while index updates generate fine-grain random writes, then unacceptable NAND Flash write costs and cryptographic overhead. The goal is thus to design an embedded DBMS engine that accommodates the tiny RAM constraints and manage large volume of data stored, in an encrypted form on Flash, without generating any random write.

The contribution of this paper is threefold:

(1) We show that state of the art database technologies cannot tackle the conjunction of ST hardware constraints and propose a *log-only* based database organization, producing only sequential writes matching these constraints.
(2) We show that massive indexing is mandatory and propose the design of indexes compatible with the log-only approach.
(3) We show how to combine these contributions to build a complete embedded DBMS engine, and notably, how to efficiently ensure its security. We then evaluate the performance of a resulting prototype named MILo-DB (Massively Indexed Log-only DB).

The rest of the paper is organized as follows. Section 2 analyzes the related works and states the problem to be solved. Section 3 gives a global overview of the log-only approach. Section 4 is devoted to the design of our indexed scheme and Sect. 5 to the way scalability is achieved. Section 6 sketches the remaining elements of the MILo-DB engine and Sect. 7 assesses the performance of this engine. Finally, Sect. 8 concludes.

2 Related work and problem formulation

This section briefly reviews works done in the embedded DBMS context. Then, it concentrates on works related to the main dimensions of the problem tackled in this paper, namely RAM conscious indexing schemes, Flash constraints management, Flash-based indexing schemes and log-structured indexing schemes. It concludes by stating the contradictory objectives pursued by these different approaches and formulates the main challenge to be solved.

Embedded DBMSs Embedded DBMS products (e.g., SQLite, BerkeleyDB) and light versions of popular DBMSs (e.g., DB2 Everyplace, Oracle Database Mobile Server) target small but relatively powerful devices (e.g., PDA, smart phone, set top box), far from the constrained hardware considered in this paper. Typically, they address neither the limitations of SMCUs (in particular the very tiny RAM) nor the specificities of NAND Flash. Proposals dedicated to databases embedded on SMCUs [11, 28] consider small databases stored in the SMCU internal stable storage—hundreds of kilobytes—and rely on NOR Flash or EEPROM technologies, both having a different behavior compared to NAND Flash (e.g., byte access granularity).

Massive indexing schemes The performance of SQL queries involving Joins and/or Aggregate computations declines sharply when the ratio between the RAM size and the size of the data to be processed falls below a certain threshold. In particular, "last resort" Join algorithms (block nested loop, sort-merge, Grace hash, hybrid hash) quickly deteriorate when the smallest join argument exceeds the RAM size [19]. To give a rough idea, with 10 KB of RAM, joining tables of 100 KB each using the block nested loop Join algorithm[4] may lead to read the second table ten times, the third table one hundred times, etc., which would be far inefficient. More recent algorithms like Jive join and Slam join use join indices [22] but both require that the RAM size is of the order of the square root of the size of the smaller table. In our context, the ratio between the RAM size and the tables' size is so small that the unique solution is to resort to a highly indexed model where all (key) joins are precomputed, as already devised in the Data Warehouse (DW) context. To deal with Star queries involving very large Fact tables (hundreds of GB), DW systems usually index the Fact table on all its foreign keys to precompute the joins with all Dimension tables, and on all Dimension attributes participating in queries [32, 34]. However, the consequence of massively indexing the database is generating a huge amount of fine-grain random writes at insertion time to update the indexes, in turn resulting in an unacceptable write cost in NAND Flash.

NAND flash behavior and flash translation layers NAND Flash memory is badly adapted to fine-grain data (re)writes. Memory is divided into blocks, each block containing (e.g., 64) pages themselves divided into (e.g., 4) sectors. The write granularity is the page (or sector) and pages must be written sequentially within a block. A page cannot be rewritten without erasing the complete block containing it and a block wears out after about 10^4 repeated write/erase cycles. The technology trend is to increase the density of Flash (e.g., MLC vs. SLC), thereby ever worsening these constraints. To tackle Flash constraints, updates are usually managed out of place with the following side effects: (1) *a Translation Layer (TL)* is introduced to ensure the address invariance at the price of traversing/updating indirection tables, (2) *a Garbage Collector (GC)* is required to reclaim stale data and may generate moves of valid pages before reclaiming a non empty block and (3) *a Wear Leveling mechanism (WL)* is required to guarantee that blocks are erased evenly. A large body of work (see [20]) has focused on the design of Flash Translation Layers (FTLs), i.e., the conjunction of TL, GC and WL. However, with scarce RAM, FTL cannot hide NAND Flash constraints without large performance degradation.[5] Typically, random writes are two or three orders of magnitude more costly than sequential writes on SD cards.[6] In addition, FTL are black box firmwares with behaviors difficult to predict and optimize. For all these reasons, we argue that delegating the optimization of the Flash usage to

[4]Bloc nested loop Join is often the only Join algorithm provided in embedded DBMS products (e.g., for SQLite see http://www.sqlite.org/optoverview.html).

[5]This is not the case in high-end SSDs which can use relatively large RAM (e.g., 16 MB) to handle those constraints.

[6]Tests on 20 recent SD cards have shown that random writes are in average 1300 times more costly than sequential writes (min 130×, max 5350×) [30].

a FTL does not make sense in our context. Our design considers that the SMCU has direct access to the NAND Flash[7] but can accommodate Flash access through FTL with minimal extension.

Indexing techniques for NAND flash Many studies address the problem of storage and indexing in NAND Flash. Conventional indexes like B+-Tree perform poorly on top of FTL [35]. Most of the recent proposals [2, 9, 21, 35] adapt the traditional B+-Tree by relying on a Flash resident log to delay the index updates. When the log is large enough, the updates are committed into the B+-Tree in a batch mode, to amortize the Flash write cost. The log must be indexed in RAM to ensure performance. The different proposals vary in the way the log and the RAM index are managed, and in the impact it has on the commit frequency. To amortize the write cost by large factors, the log is seldom committed, leading to consume more RAM. Conversely, limiting the RAM size means increasing the commit frequency, thus generating more random writes. RAM consumption and random write cost are thus conflicting parameters. Under the RAM limitations of SMCUs the commit frequency becomes de facto very high and the gain on random writes vanishes. PBFilter [36] is an indexing scheme specifically designed for Flash storage in the embedded context. It organizes the index in a sequential way thereby avoiding random writes. The index is made of a key list compacted using a Bloom filter summary, which can further be partitioned. This leads to good lookup times with very few RAM. However, PBFilter is designed for primary keys. With secondary keys the Bloom filter summary becomes non selective and mainly useless (the complete set of keys has to be accessed). This makes PBFilter of little interest for implementing massive index schemes, since most indexes are secondary keys.

Log-structured indexes Many proposals for log-structured indexes [7, 26, 27] are based on principles from log-structured file systems [29] (also used in the design of FTLs to hide NAND Flash constraints) or older principles like *Differential File* [31], an overflow area storing new records, merged with the main data file at some future point. These proposals differ in the way indexes are built or maintained but they always make use of relatively large buffers in RAM incompatible with our constraints. More recently, [8] proposed Hyder, a log-structured, multiversion key-value database, stored in flash memory, and shared over the network. Hyder makes use of a single binary balanced tree index to find any version of any tuple corresponding to a given key. The binary tree is not updated in place, the path from the inserted or updated node being rewritten up to the root. Unfortunately, this technique cannot be used to implement massive indexing schemes (binary trees are not adequate to index non unique keys). Still in the key-value store context, SkimpyStash [14], LogBase [33] and SILT [23] organizes key-value pairs in a log structure to exploit sequential writes and maintain some form of in-memory (RAM) indexing with a size proportional to the database size, thus consuming too much RAM for a SMCU (at least 1 Byte per record).

[7] A first layer (the Hardware Adaptation Level) of the controller software manages Low Level Drivers (LLD), Error Correction (ECC) and Bad Block Management (BBM). The second layer is the FTL, and it can be bypassed on most platforms.

Problem formulation To conclude, the challenge tackled in this paper lies in the combination of a tiny working memory (RAM) with a huge NAND Flash mass storage badly accommodating random writes. Executing queries with acceptable performance on gigabytes of data with a tiny RAM entails indexing massively the database. The consequence is generating a huge amount of fine-grain random writes at insertion time to update the indexes, which in turn results in an unacceptable write cost in NAND Flash. Conversely, known solutions to decrease the amount of random writes in Flash require a significant amount of RAM. A vicious circle is then established and lets little hope to build an embedded DBMS engine by assembling state of the art solutions. The objective of our work is to break this circle.

3 Log-only database organization

To tackle the problem stated above, we propose a new database organization called *Log-Only* trying to reconcile three a priori contradictory objectives: (1) massively indexing the database, (2) producing only sequential writes in Flash and (3) consuming a tiny amount of RAM, independent of the database size. To this end, we propose to organize the complete database (raw data but also indexes, buffers and more generally all data structures managed by the DBMS engine) into *Log Containers*. Log containers are purely sequential persistent recipients.

Log Container (LC) A LC is a data structure satisfying three conditions: (1) its content is written sequentially within the Flash block(s) allocated to it (i.e., pages already written are never updated nor moved); (2) blocks can be dynamically added to a LC to expand it; (3) a LC is fully reclaimed when obsolete (no partial garbage collection occurs).

The net effect of organizing the complete database into Log Containers is to avoid random writes by definition. Hence, the dramatic overhead of random writes in Flash is avoided.[8] However, processing sequential structures do not scale well also by definition. Hence, to make the database scalable, the initial sequential database must be iteratively reorganized into more efficient data structures. The resulting data structures must be produced in Log Containers as well to preserve the initial objective. The way a sequential database can be initially produced and then reorganized according to log-only constraints is sketched below.

3.1 Log-only initial database

Base data Natural log-only solutions can be easily devised to organize the base data (*tables*). Typically, a table can be stored as a sequence of rows in a Row Store scheme or as a set of sequences of attribute values in a Column Store one. Adding new base data is direct in both cases. ↓DATA (↓ stands for log-only) denotes the LCs dedicated to base data.

[8] Moreover, the Flash Translation Layer becomes useless (thereby saving translation costs) and the garbage collection and wear leveling mechanism can be greatly simplified.

Indexes While particular index structures like *bitmap indexes* easily comply with the LC definition, most classical indexes (e.g., tree-based or hash-based) are proscribed. Indeed, inserting new base data would generate random node/bucket updates. In Sect. 4, we propose new forms of indexes compatible with LCs to speed up joins and selections. We denote by ↓IND the LCs dedicated to such indexes.

Updates and deletes When updating or deleting base data, directly reporting modifications in ↓DATA would violate the LC definition. Instead, updates and deletes are logged in dedicated LCs, respectively named ↓UPD and ↓DEL. To manage updates, the old and new attribute values of each updated tuple are logged in ↓UPD. At query execution time, ↓UPD is checked to see whether its content may modify the query result. First, if a logged value matches a query predicate, the query is adjusted to eliminate false positives (i.e., tuples matching the query based on their old value but not on their new value) and to integrate false negatives (i.e., tuples matching the query based on their new value but not on their old value). Second, ↓UPD and ↓DEL are also checked at projection time, to project up-to-date values and to remove deleted tuples from the query result. An important remark is that ↓DEL only records the tuples that have been explicitly deleted (i.e., it does not records cascaded deletes, that is deletes of referencing tuples). This does not strongly impact query execution since join indexes are created anyway and can be accessed efficiently when evaluating a query (this access is even mandatory for join queries as explained in the last paragraphs of Sect. 4.1). Overheads are minimized by indexing ↓UPD and ↓DEL on Flash (similarly as indexing a table), and building dedicated data structures in RAM to avoid accessing Flash for each result tuple.[9]

Buffers and transactions ↓DATA, ↓IND, ↓UPD, ↓DEL being made of fine grain elements (tuples, attribute values, pointers, or index entries), inserting data into LCs without buffering would lead to waste a lot of space in Flash. The objective of buffering is to transform fine-grain writes (e.g., attributes, pointers) to coarse-grain writes (a full flash page) in LCs. To this end, we implement buffers themselves by means of LCs denoted by ↓BUF. Roughly speaking, fine-grain elements are gathered into buffers until a full page of those elements can be built. The way ↓BUF is actually managed and the way transaction atomicity (to undo dirty insertions into LCs) is enforced is more deeply discussed in Sect. 6.

The log-only database is denoted by ↓DB and is thus composed of the LCs of buffers, base data, indexes, update and delete logs.

The log-only database principle leads to a robust and simple design, compliant with all ST constraints. However, such a design scales badly since ↓IND cannot compete with classical indexes (e.g., B+-Tree), and the accumulation over time of elements in ↓UPD and ↓DEL will unavoidably degrade query performance. There is a scalability limit (in terms of ↓DATA, ↓IND, ↓UPD and ↓DEL size) beyond which performance expectation will be violated, this limit being application dependent.

[9] While the strategy for handling deletes and updates is rather simple, the details on query compensation is a bit tricky and cannot be included in the paper due to size constraint.

3.2 Log-only reorganizations of the database

To tackle this scalability issue, we reorganize ↓DB into another **optimal* log-only database denoted by *DB. By **optimal*, we mean that the performance provided by *DB is at least as good as if *DB were built from scratch by a state of the art method ignoring Flash constraints. For example, the reorganization of ↓DB into *DB can result in a set of tables where all updates and deletes are integrated and which are indexed by tree-based or hash-based indexes. Note that the reorganization process presented below is independent of the indexing scheme selected for *DB.

For the sake of clarity, we introduce a reorganization counter, named *instance,* added in subscript in the notations. Before any reorganization occurs, the initial log-only database is

$$\downarrow DB_0 = (\downarrow BUF_0, \downarrow DATA_0, \downarrow IND_0, \downarrow UPD_0, \downarrow DEL_0).$$

When the scalability limit of $\downarrow DB_0$ is reached, $*DB_0$ needs to be built. The reorganization process triggers 3 actions.

$\downarrow BUF_0$ elements are flushed into their target LC (i.e., $\downarrow DATA_0$, $\downarrow IND_0$, $\downarrow UPD_0$, $\downarrow DEL_0$).

All new insertions, updates and deletes are directed to $\downarrow DB_1 = (\downarrow BUF_1, \downarrow DATA_1, \downarrow IND_1, \downarrow UPD_1, \downarrow DEL_1)$ until the next reorganization phase ($\downarrow DB_1$ is initially empty).

$\downarrow DB_0$ (which is then frozen) is reorganized into $*DB_0$, composed of $*DATA_0$ and $*IND_0$.$*DATA_0$ is built by merging $\downarrow DATA_0$ with all updates and deletes registered in $\downarrow UPD_0$ and $\downarrow DEL_0$ (*Merge* operation). $*IND_0$ is the **optimal* reorganization of $\downarrow IND_0$, including modifications stored in $\downarrow UPD_0$ and $\downarrow DEL_0$ (*ReorgIndex* operation).

The database is then composed of $\downarrow DB_1$ (receiving new insertions, updates and deletes) and of $*DB_0$, itself composed of $*DATA_0$ and $*IND_0$. When the reorganization process terminates (i.e., $*DATA_0$ and $*IND_0$ are completely built), all LCs from $\downarrow DB_0$ can be reclaimed. $\downarrow DB_1$ keeps growing until the next reorganization phase. The next time the scalability limit is reached, a new reorganization occurs. Reorganization is then an iterative mechanism summarized in Fig. 2 (without ↓BUF for the sake of clarity).

$$
\begin{array}{l}
\textbf{\textit{Reorg}}(\downarrow \textbf{\textit{DB}}_i) \rightarrow *\textbf{\textit{DB}}_i \\
\quad \downarrow DB_{i+1} = \varnothing \\
\quad Flush(\downarrow BUF_i) in \downarrow DATA_i, \downarrow IND_i, \downarrow UPD_i, \downarrow DEL_i \\
\quad Reclaim(\downarrow BUF_i) \\
\quad if\ (i > 0) \\
\qquad *DATA_i = Merge(*DATA_{i-1}, \downarrow DATA_i, \downarrow UPD_i, \downarrow DEL_i) \\
\qquad *IND_i = ReorgIndex(*IND_{i-1}, \downarrow IND_i, \downarrow UPD_i, \downarrow DEL_i) \\
\qquad Reclaim(*DATA_{i-1}, *IND_{i-1}) \\
\quad else \\
\qquad *DATA_i = Merge(NULL, \downarrow DATA_i, \downarrow UPD_i, \downarrow DEL_i) \\
\qquad *IND_i = ReorgIndex(NULL, \downarrow IND_i, \downarrow UPD_i, \downarrow DEL_i) \\
\qquad Reclaim(\downarrow DATA_i, \downarrow IND_i, \downarrow UPD_i, \downarrow DEL_i)
\end{array}
$$

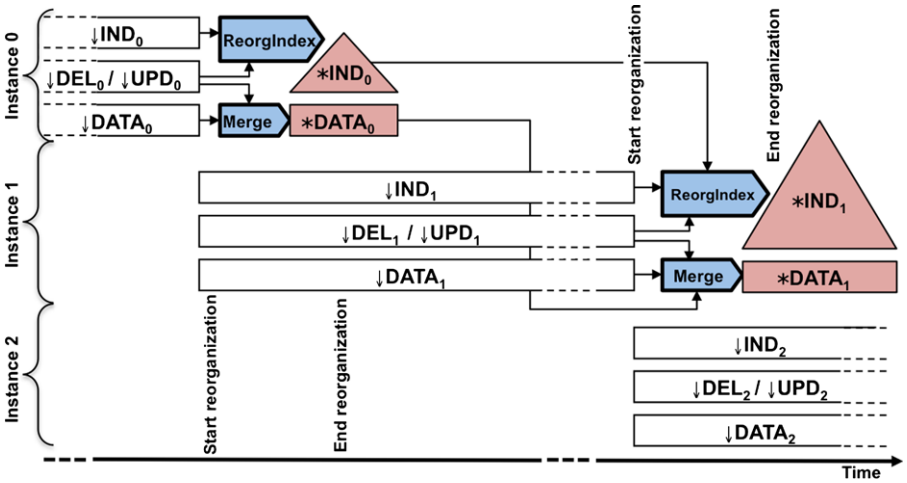


Fig. 2 The log-only reorganization process

Remark that reorganization is very different in spirit from batch approaches deferring updates thanks to a log. Indeed, deferred updates produce random rewrites while log-only reorganizations produce only sequential writes into LCs. The price to pay is however a complete reconstruction of $*DB_i$ at each reorganization.

4 Log-only indexes

In this section, we discuss the types of indexes that are required in a massively indexed context, and propose a design for constructing them (both ↓IND and *IND) in a way compliant with the log-only constraints. We concentrate on indexing schemes for simple Select/Project/Join/Aggregate queries. We consider this enough to validate the approach in the settings presented in the introduction and let the study of more complex queries for future work.

4.1 Which indexes?

Let us first consider Select/Project/Join queries (SPJ). The very small ratio between RAM and database size leads to use generalized selection and join indexes [5, 28, 32, 34]. These indexes are called generalized in the sense that they capture the transitive relationships which may exist between tuples of different tables. In other words, we say that a tuple t of table T_i *references* a tuple t' of table T_j (denoted by $t \rightarrow t'$), if t is linked directly or transitively to t' by a join path on foreign/primary keys (i.e., starting with a foreign key of t and ending with t' primary key). On this basis, generalized indexes can be defined as follows:

- **Index $I_{T_i \rightarrow T_j}$ (TJoin)**: *For a given tuple t of T_i, index $I_{T_i \rightarrow T_j}$ returns the identifiers of all tuples of T_j referenced by t*: $I_{T_i \rightarrow T_j}(t \in T_i) = \{t'.id / t' \in T_j$ *and* $t \rightarrow t'\}$.

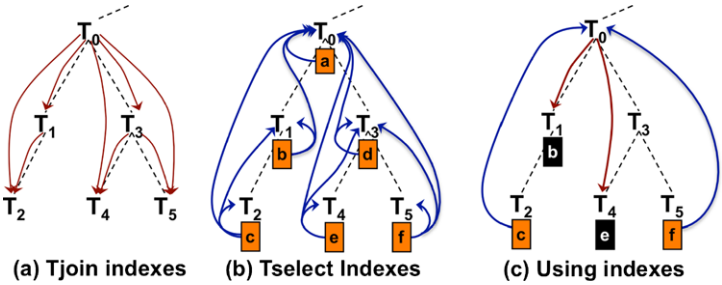


Fig. 3 The massive indexation scheme and its usage

This index is actually a one-way generalized join index, precomputing natural joins from referencing to referenced tables. Let us call this class of indexes TJoin (Transitive Join).

- **Index $\mathbf{I_{T_j.A \to T_i}}$ (TSelect)**: *Given a value v from the domain of attribute $\mathrm{T}_j.\mathrm{A}$, this index returns the identifiers of each tuple t of T_i such that t references a tuple t' of T_j where $t'.\mathrm{A} = v$*: $\mathrm{I}_{\mathrm{T}_j.\mathrm{A} \to \mathrm{T}_i}\ (v \in dom(\mathrm{T}_j.\mathrm{A})) = \{t.id / t \in \mathrm{T}_i, t' \in \mathrm{T}_j$ *and* $t \to t'$ *and* $t'.\mathrm{A} = v\}$.

Note that the result must be ordered on $t.id$ to allow merging (by unions or intersections) sorted lists of t.id with no RAM. In the particular case where $i = j$, $t = t'$ and this index acts as a regular selection index (in other words, each tuple references itself). If $i \neq j$, it implements a selection in a referencing table on an attribute from a referenced table (just as indexing a Fact table on attributes of a dimension table). Let us call this class of indexes TSelect (Transitive Select).

For instance, on the partial database schema presented in Fig. 3a, T_0 references directly T_1 and T_3 and transitively T_2, T_4 and T_5. 8 TJoin indexes are created in this example. Figure 3b shows TSelect indexes, considering that a single attribute in each table T_i is indexed. For instance, for attribute c of table T_2, we create 3 such indexes: $\mathrm{I}_{\mathrm{T}_2.c \to \mathrm{T}_2}$, $\mathrm{I}_{\mathrm{T}_2.c \to \mathrm{T}_1}$, $\mathrm{I}_{\mathrm{T}_2.c \to \mathrm{T}_0}$.

More generally, to allow an efficient computation of all SPJ queries with a tiny RAM, we consider a massive indexing scheme in the same line as [5, 28, 32, 34]. This scheme is composed by (1) a TJoin index for each $(\mathrm{T}_i, \mathrm{T}_j)$ such that T_i references T_j[10] and (2) a TSelect index for each attribute potentially involved in selection predicates and for each $(\mathrm{T}_i, \mathrm{T}_j)$ such that T_i references T_j.

Intuitively, SPJ queries are executed by (1) traversing the relevant TSelect indexes, i.e., the $\mathrm{I}_{\mathrm{T}_j.\mathrm{A} \to \mathrm{T}_i}$ indexes corresponding to all predicates of the form $\mathrm{T}_j.\mathrm{A}$ in the query and such that T_i is a common table for all indexes, (2) merging the sorted sets of T_i tuple identifiers in pipeline (by intersection and/or union); and (3) traversing the relevant TJoin indexes to project the resulting tuples. For instance, Fig. 3c shows the strategy for computing a query which joins all tables, evaluates the selection predicates ($\mathrm{T}_2.c = v1$ and $\mathrm{T}_5.f = v2$) and projects attributes $\mathrm{T}_1.b$ and $\mathrm{T}_4.e$. This leads to: (1) lookup in $\mathrm{I}_{\mathrm{T}_2.c \to \mathrm{T}_0}(v1) \to S1$ and $\mathrm{I}_{\mathrm{T}_5.f \to \mathrm{T}_0}(v2) \to S2$, (2) intersect sorted sets

[10] For the sake of clarity, we make here the assumption that at most one join path exists between two tables (e.g., a snowflake schema). The indexing scheme can be extended trivially to the multiple paths case.

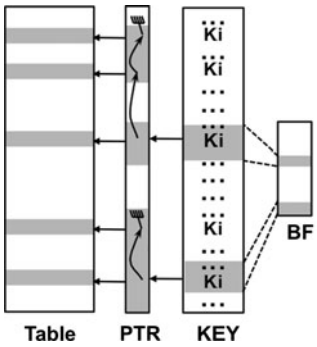


Fig. 4 TSelect index design in ↓DB

$S1$ and $S2$ in pipeline (3) use indexes $I_{T_0 \to T_1}$ and $I_{T_0 \to T_4}$ to find resulting tuples and project attributes $T_1.b$ and $T_4.e$.

Queries with Group By clauses and aggregates are more complex to execute since the RAM consumption is linked with the number of groups in the results. In that case, the result of the SPJ part of the query is stored in a temporary Log Container. Then the whole RAM is used to perform the aggregation in several steps by computing a fraction of the result at each iteration. Several strategies have been proposed in [5] in another context and can be applied directly here.

4.2 TJoin index design

For a given table T_i, a set of TJoin indexes must be created, one for each referenced table. All those indexes can be stored in the same Log Container to save IOs when inserting tuples into T_i since they are filled exactly at the same rate. Note that storing these indexes separately would bring marginal IOs savings at query time. Indeed, TJoin indexes are generally not accessed sequentially: they are accessed after evaluating the selection predicates using TSelect indexes which return sets of tuples identifiers. Hence, the set of TJoin indexes of table T_i can be represented as a single regular table with as many columns as they are tables referenced by T_i. Such as TJoin table is naturally ordered on T_i tuple identifiers, following the insertion order in T_i. Access to this table is direct, using T_i tuple identifiers as positions. At insertion time, the direct references are part of the inserted tuple (foreign keys), the transitive ones being retrieved by accessing the TJoin indexes of referenced tables (e.g., in Fig. 3a., the set $I_{T_0 \to T_j}$ is filled by accessing $I_{T_1 \to T_2}$, $I_{T_3 \to T_4}$ and $I_{T_3 \to T_5}$). Since TJoin indexes behave as normal tables, they are stored as regular tables.

4.3 TSelect index design

As pictured in Figs. 4 and 5, TSelect indexes have a more complex structure to comply with the log-only constraints. We first describe the general organization of these indexes, and then explain their usage (lookup and maintenance operations). Considering a TSelect index I, we denote by ↓I the log-only initial structure of this index and by *I its **optimal* reorganization.

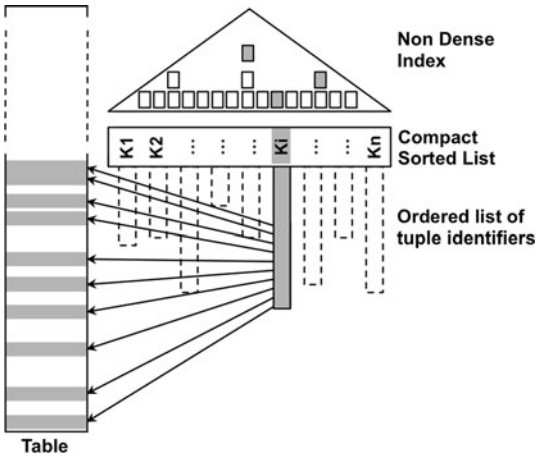


Fig. 5 TSelect index design in *DB

4.3.1 Organization of TSelect indexes in ↓DB

Three Log Containers are used to materialize an index $I_{T_j.A \to T_i}$. A first container called KEY is used to sequentially store, for each new tuple t of T_i, the T_j.A value of the referenced tuple (this key can be efficiently obtained by accessing the appropriate $I_{T_i \to T_j}$ index using the tuple identifier of t). Hence, KEY is a dense index which must be sequentially scanned when looking up to a key. To decrease this cost, we store a compressed representation of KEY, obtained using *Bloom filters* [10], in a second container called BF. A Bloom filter represents a set of values in a compact way and allows probabilistic membership queries with no false negatives and a very low rate of false positive.[11] One new Bloom filter is built for each new Flash page of KEY and is sequentially appended into BF. At lookup time BF is sequentially scanned and a page of KEY is accessed only in case of a match in a Bloom filter. The false positive rate being rather low, the IO cost is roughly decreased by the compression factor of Bloom Filters. To further reduce the lookup cost, index entries sharing the same key value are chained by pointers in a third container, called PTR (to comply with the log-only constraints, this chaining is done backward). Hence, only the head of the chain corresponding to the searched value is done sequentially through BF then KEY. Then, a direct access to PTR is performed to access the pointer chain and follow it to retrieve all other matching tuples' identifiers.

The benefit incurred by PTR comes at the expense of maintaining the pointer chains. Each time a new tuple is inserted, the head of the chain corresponding to its key must be found to be able to expand the chain. While this search is improved by BF, it may become expensive when inserting a tuple having a "rare" key (i.e., a value with few occurrences in KEY). To bound this overhead, the chain is broken if this head is not found after a given threshold (e.g., after having scanned n BF pages) and the new key is simply inserted with a preceding pointer set to NULL. At lookup time,

[11] For example, the false positive rate using 4 hash functions and allocating 16 bits per value is 0.24 % [10]. Hence, Bloom Filters provide a very flexible way to trade space with performance.

when a NULL pointer is encountered, the algorithm switches back to BF, skipping the n next pages of BF, and continues searching the next key occurrence using BF and KEY. A positive consequence of this scheme is that the pointers can be encoded on a smaller number of bits (linked to the number of keys summarized by n pages of BF) thereby reducing the access cost of PTR.

*4.3.2 Organization of TSelect indexes in *DB*

The construction of *I takes advantage of three properties: (1) *I is never updated and can therefore rely on more space and time efficient data structures than traditional B^+-Tree (100 % of space occupancy can be reached compared to about 66 % in B-Tree nodes); (2) at the time the reorganization occurs, all the data items to be indexed are known; (3) the whole RAM can be dedicated to index construction since reorganization can be stopped/resumed with low overhead (see Sect. 5). The constraint however is to build *I in such a way that the lists of tuple identifiers associated to the index entries are kept sorted on the tuples insertion order. This requirement is mandatory to be able to merge efficiently lists of identifiers from multiple indexes. The resulting reorganized index structure is depicted in Fig. 5. A first Log Container contains the sets of compact *Ordered Lists of tuple Identifiers* built for each index key K_i. A second container stores the set of index entries, represented as a *Compact Sorted List*. Finally, the last container stores a compact *Non-Dense Index*, made of the highest key of each page of the compact sorted list in a first set of Flash pages, itself indexed recursively up to a root page. The index is built from the leaves to the root so that no pointers are required in the non-dense index. Reorganized indexes are then more compact and efficient than their traditional B^+-Tree counterpart. Note that the non-dense index and compact sorted list can be shared between several indexes indexing the same key (T_j.A).

*4.3.3 Combining indexes in ↓DB and *DB*

Tuples of $*DB_{i-1}$ tuples always precede tuples of $\downarrow DB_i$ in their insertion order, and so are the values of their identifiers. To produce ordered tuple identifiers,[12] the lookup operation thus combines the results from $\downarrow I_i$ and $*I_{i-1}$ by a simple concatenation of the lists of matching tuples (see Fig. 6).

To conclude, the proposed index structure allows an efficient retrieval of the ordered list of identifiers with a bounded insertion cost. It answers all requirements of TSelect indexes while satisfying the log-only constraints assuming reorganization can be done efficiently.

5 Reorganization process

In our approach, reorganizing a database sums up to processing the *Merge* and *ReorgIndex* operations as introduced in Sect. 3.2. We focus here on the complex *ReorgIndex* operation since *Merge* is rather trivial (basically, checking in pipeline each

[12]The result tuple identifiers are produced in reversed order, from the most recently inserted to the least recent inserted, which is suitable with the definition of Tselect Index $I_{T_j.a \rightarrow T_i}$ given in Sect. 4.1.

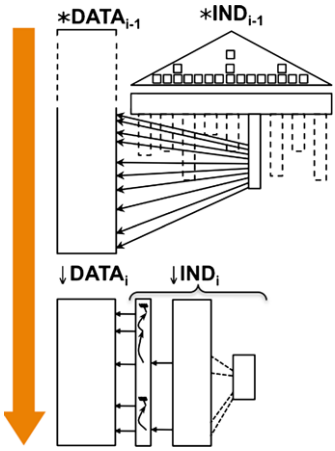


Fig. 6 Combining Indexes in ↓DB and *DB

tuple in ↓DATA and *DATA using summaries of ↓UPD and ↓DEL in RAM to avoid accessing it on Flash).

Different structures are involved in index reorganization (see Fig. 2). Let us first analyze their organization to infer an adequate reorganization strategy matching the ST constraints.

- *IND contains a set of TSelect indexes[13] sorted on the index key, each key being linked to an *ordered list of tuple identifiers* (see Sect. 4.3), thus each index is sorted on (*Key*, *ID*) where *Key* are index keys and *ID* are tuple identifiers.
- ↓IND contains a set of TSelect indexes, each sorted on tuple identifiers since it is built sequentially.
- ↓UPD and ↓DEL have not specific order since elements are added sequentially when updates and deletes are performed.

These structures being stored in different orders, the reorganization process appears to be complex, especially with large data structures and scarce RAM. Note that after some initial reorganizations, the size of *IND is much larger than ↓IND. In addition, the size of ↓DEL and ↓UPD is expected to be small relatively to ↓IND (i.e., much less updates and deletes than insertions in a traditional database). The idea is thus to adapt the ordering of ↓BUF, ↓UPD and ↓DEL to the one of *IND such that they can be merged in pipeline (i.e., without RAM consumption) to build the new *IND (denoted *newIND in this section, to avoid indices for better readability).

- *Adapting ↓IND*: Each $\downarrow I_{T_j.A \rightarrow T_i}$ must be sorted on (*Key*, *ID*) such that it will be sorted as the corresponding $*I_{T_j.A \rightarrow T_i}$. Sorting can be done by sort merge using the whole RAM and a temporary LC reclaimed at the end of the process.
- *Adapting ↓DEL*: To avoid querying ↓DEL for each tuple of ↓IND and *IND (a simple but highly costly solution), we compute beforehand the impact of deleted tuples on all TSelect indexes. This requires one query per deleted tuple but the impact on performance is limited given the reduced size of ↓DEL. Once ↓DEL is

[13] Since TJoin indexes behave as normal tables, they are handled in the same way by the *Merge* operations, and thus, are not discussed here.

correctly *expanded* (i.e., integrates cascade deletes and impact on all indexes using $I_{T_j.ID \rightarrow T_i}$), the result is sorted on (#*Index, Key*, *ID*) where #*Index* is the identifier of TSelect Indexes $I_{T_j.A \rightarrow T_i}$. This is described in the pseudo-code below:

Adapt(↓DEL) → D

1. *Cascade*(↓*DEL*) → *D* // one query per deleted tuple to find cascade deletes*
2. *Project_on_all_indexed_attributes*(*D**) → *D*π // using* $I_{T_j}.ID \rightarrow T_i$
3. *Sort*(*D* ∗ *π*) *on* (#*Index*, *Key*, *ID*) → *D*

- *Adapting* ↓*UPD* : ↓UPD roughly follows the same processing as ↓DEL with two main differences. First, we need to build two structures, ↓UPD_FP for false positives, and ↓UPD_FN for false negatives. Second, ↓UPD impacts a more reduced number of indexes than ↓DEL (i.e., only those concerning the updated attributes) making this expansion less costly. As for ↓DEL, after expansion, the result is sorted on (#*Index, Key*, *ID*).

Adapt(↓UPD) → (U_FP, U_FN)

1. *Cascade*(↓*UPD*) → *U*// one query per updated tuple to compute the impact // on all TSelect indexes*
2. *Sort*(*U**) *on* (#*Index*, *OldValue*, *ID*) → *U_FP // False positive i.e. OldValue // should be removed from indexes*
3. *Sort*(*U**) *on* (#*Index*, *NewValue*, *ID*) → *U_FN // False negative i.e. NewValue // should be added to indexes*

Once all structures have the same order, we can compute *newIND as described in the pseudo-code below:

ReorgIndex(*IND, ↓IND, ↓UPD, ↓DEL) → *newIND

4. *Adapt*(↓*DEL*) → *D*
5. *Adapt*(↓*UPD*) → (*U_FP*, *U_FN*)
6. *For each* $I_{T_j.A \rightarrow T_i}$ *index*
7. *Adapt*($\downarrow I_{T_j.A \rightarrow T_i}$) → *S // Sort on* (*Key*, $T_i.ID$)
8. *Fusion*($*I_{T_j.A \rightarrow T_i}$, *S*, *D*, *U_FP*, *U_FN*) → $*newI_{T_j.A \rightarrow T_i}$

The *Fusion* operation merges in pipeline $*I_{T_j.A \rightarrow T_i}$ with S (the sorted version of $\downarrow I_{T_j.A \rightarrow T_i}$), removing deleted entries (D), false positives (U_FP) and adding false negatives (U_FN).

The reorganization is a long process (see Sect. 7.5) and thus should be designed such that it can be stopped and resumed with minimal overhead if the database is queried while a reorganization process is in progress. Actually, the reorganization process combines three types of operations:

- *Pipelined operations*: These operations (e.g, Fusion) can be stopped at very low cost since we only need to store the current state of the operation (i.e., pointers on all operands of the operation). A single page IO need be done (in an operation log).
- *Queries*: ↓UPD and ↓DEL *adaptation* triggers a set of queries to build their expanded versions. Each individual query is very simple (basically following TJoin indexes and projecting some individual values) and can then be completed before stopping the reorganization process.

- *Sort operations*: If queries are triggered when a sort operation is about to be performed (e.g., RAM has been loaded with data to be sorted), we can either abort this operation or complete it (i.e., sort the data and write it back to Flash in temporary LCs), thus delaying queries. The scarce RAM is, this time, an advantage. Indeed sorting 64 KB of data can be done in few ms, as well as storing the result on Flash (about 10 ms for 64 KB).

Thus, we can easily stop and resume the reorganization process, completing any atomic operation and logging the current state of the reorganization process in a single Flash IO. As a consequence, queries can be run while reorganization is incomplete. Note that this is not a problem because LCs from the previous database instance are reclaimed only at the end of the process and can then serve to answer queries.

The reorganization cost is related to the size of ↓DB, to the number of deletes and updates performed since last reorganization, and to the size of *DB. The essential features of the reorganization are: (1) the reorganized part of the database is read and written only once; (2) the process can be interrupted then resumed, without hindering queries on the database; (3) the process is by nature failure-resistant (e.g., sudden power loss) since no element of ↓DB is reclaimed before the process completion;[14] (4) since each index is reorganized independently, queries that may be triggered during reorganization can take advantage of already reorganized indexes.

6 Buffers, transaction management, security

The preceding sections have presented (1) the log-only approach underlying MILo-DB, (2) a massive indexing scheme based on generalized indexes and its log-only implementation and (3) the reorganization process ensuring the scalability of the whole. These contributions form the core of the MILo-DB design. This section gives some insights on the main other elements of this engine, namely the buffer management, the transaction management and the crypto-protection of the database. Taken together, these elements form a complete embedded DBMS engine capable of addressing the requirements of the targeted use-cases.

6.1 Buffer management

↓DB structures are made of fine grain elements (tuples, attribute values or index entries). NAND Flash constraints however impose writing a new page for each new element if insertions are done one by one into those structures. This leads to waste a lot of space and to decrease query performance. Indeed, the density of a Log Container determines the efficiency of scanning it. Buffering strategies are then required to transform fine-grain writes in Flash to coarse-grain writes. Each new element targeting a given Log Container *LC* is first gathered into buffers until the set of buffered elements for this LC can fill a complete Flash page, which is then flushed.

[14]The work required to recover from a crash during the reorganization can be minimized by storing on Flash the current state of each operation and analyzing this log at recovery time.

Buffers cannot reside in RAM because of its tiny size, and also because we cannot assume electrical autonomy and no failure.[15] Hence, buffers must be saved in NAND Flash and the density of buffer pages depends on the transactions activity. To increase buffer density (and therefore save writes), elements targeting different LCs are buffered together if they are filled and flushed synchronously. For example, let us consider the Log Containers PTR, KEY and BF used in ↓IND. For a given table T_i, all indexes of type $I_{T_j.A \rightarrow T_i}$ built over that table T_i can be buffered together. A new element is inserted into their respective PTR and KEY Log Containers at each insertion in table T_i, and a new entry is inserted into their respective BF Log Containers synchronously each time a KEY page is filled. In Fig. 3b, this means that, at least, indexes $I_{T_1.b \rightarrow T_0}$, $I_{T_2.c \rightarrow T_0}$, $I_{T_3.d \rightarrow T_0}$, $I_{T_4.e \rightarrow T_0}$, and $I_{T_5.f \rightarrow T_0}$ can be buffered together.

Remark also that buffers must be organized themselves as LCs to comply with the log-only constraints. We manage each buffer as a sliding window within its Log Container, using *Start* and an *End* markers to identify its active part (i.e., the part not yet flushed).

6.2 Transaction management

Regarding transaction atomicity, we restrict ourselves to a single transaction at a time (which makes sense in our context). Rolling-back a transaction, whatever the reason, imposes undoing all dirty insertions to the Log Containers of ↓BUF. To avoid the presence of dirty data in Log Containers, only committed elements of ↓BUF are flushed in their target structure as soon as a full Flash page can be built. So, transaction atomicity impacts only the ↓BUF management. In addition to the *Start* and *End* markers of ↓BUF, a *Dirty* marker is needed to distinguish between committed and dirty pages. Rolling-back insertions leads (1) to copy after *End* the elements belonging to the window [*Start*, *Dirty*] containing the committed but unflushed elements, and (2) to reset the markers ($Dirty = Dirty - Start + End$; $Start = End$; $End = Dirty$) thereby discarding dirty elements.

6.3 Cryptographic protections

The NAND Flash being not protected by the tamper-resistance of the secure MCU, cryptographic techniques are required to protect the database footprint against confidentiality and integrity attacks. Indeed, attacks can be conducted by a pirate (if the ST is stolen) or by the ST holder herself (to modify administrative forms, to forge medical prescriptions, etc.). The database must thus be encrypted and protected against four forms of tampering: *modification*, *substitution* (replace valid data by other valid data), *deletion* and *replay* of data items (i.e., replacing a data item by an old valid version).

Confidentiality attacks: All data items are encrypted separately using AES. To prevent statistical attacks on encrypted data, all instances of a same value are encrypted differently (see below).

[15] Actually, it is worth managing a very small buffer (e.g., 1 page) in RAM to buffer several insertions of the same transaction.

Modification, substitution, deletion attacks: Illicit *modifications* are traditionally detected by computing a Message Authentication Code (MAC: a keyed-cryptographic hash [24]) of each data item. The item address is incorporated in the MAC to prevent *substitution*. Finally, *deletions* are detected by checking the integrity of the container of a data item (e.g., the page containing it).

Replay attacks: Replay attacks are trickier to tackle. Traditionally, detecting *replay* attacks imposes including a version number in the MAC computation of each data item and storing the association between a data item and its version in the secure internal storage of the ST. If any data item may have any version number, maintaining these numbers requires either a very large secure memory, unavailable on STs, or maintaining a hierarchy of versions in NAND Flash, storing only the root in secure memory (thereby inducing huge update costs). The log-only database organization proposed in this paper greatly simplifies the problem. Obsolete LCs can only be reclaimed during database reorganization and reallocated to next database instances. Indeed when reorganization starts, all LCs of the current database are frozen. Hence, checking version can be done by storing only the current database instance number in the secure internal storage of the ST and by including this number in the MAC computation of each data item.

Thus, protecting the database is done using state of the art cryptographic techniques. At query execution, each accessed data item is decrypted and its integrity is checked. The integrity of the query result is guaranteed since the evaluation starts by metadata stored in the secure MCU (secure root). The challenge is however to minimize the cryptographic overhead by (1) adapting the granularity of encryption primitives to the granularity of the data items accessed and (2) proposing cryptographic conscious page organization.

Granularity of encryption primitives: LCs are written sequentially at flash page granularity, pushing for computing the MAC at that granularity to minimize the storage overhead. LCs are however mainly read randomly at a very small granularity, e.g., when following pointers in indexes or when accessing attribute values. In those cases, computing the MAC of a full page to check the integrity of a small chunk is clearly sub-optimal. For small granularity data, we use Block-Level Added Redundancy Explicit Authentication (AREA) [15], an authenticated encryption mode working as follows. A nonce is concatenated to small granularity plaintext data, and then potentially padded, to obtain an encryption block. After decryption, the nonce must match its initial value. Otherwise, at least one bit of the data or the nonce has been modified. The security of this construction is based on (1) the diffusion property of the block level encryption functions, (2) the uniqueness of the nonce, (3) the nonce size. Using 16 bytes block encryption functions such as AES [24], a 6 bytes nonce and 10 bytes of payload data, the chance of an attacker to correctly build a fake data is $1/2^{48}$. In our scheme, the nonce is composed of the data address (4 bytes) concatenated with the version information. Thus, the nonce is guaranteed to be unique in the whole database life. We use AREA for small granularity data, (e.g., pointers, small keys, small attributes) generally accessed randomly. For instance, PTR is encrypted using AREA, while the ordered list of tuple identifiers in *IND uses a classical MAC since the whole list is always accessed.

Cryptographic conscious page organization: Since Flash memory can only be read and written by page, the content of each page can be reorganized before being flushed

in Flash to minimize cryptographic costs. This idea can be exploited in several structures. Typically, the structure KEY in ↓IND is always processed with the same access pattern. If a page of KEY is accessed, this means that a match exists in BF for the searched key k_i and for this page. While searching k_i within the page, to avoid decrypting the complete list of keys, we encrypt the searched key k_i and look up its encrypted representation. However, identical keys must have different encrypted representations to prevent any statistical attack. To avoid duplicate values inside the same page we store only the last occurrence of each key in the page, the previous occurrences being obtained using PTR. To avoid duplicate values in different pages, we encrypt each key using an initialization vector (IV) [24] depending on the page and on the version. At lookup time, if k_i is found in the targeted page, its integrity must be checked. In the rare cases where k_i is not found (BF false positive), the integrity of the complete list of keys must be checked. To optimize both cases, we store in the page the MAC of each key in addition to the MAC of the whole set of keys.

7 Performance evaluation

Developing embedded software for SMCU is a complex process, done in two phases: (1) development and validation on simulators, (2) adaptation and flashing on SMCU. Our current prototype is in phase 1 and runs on a software simulator of the target hardware platform: a SMCU equipped with a 50 MHz CPU, 64 KB of RAM and 1 MB of NOR Flash, connected to a 4 GB external NAND Flash. This simulator is IO and crypto-accurate, i.e., it computes exactly the number of page read/write, block erase operations and cryptographic operations (by type), by far the most costly operations. In order to provide performance results in seconds, we calibrate[16] the output of this simulation using a set of micro-benchmarks and with performance measurements done on the ST, using a previous prototype named PlugDB. PlugDB has already reached phase 2 (i.e., runs on a real hardware platform), has been demonstrated [6] and is being experimented in the field in an experimental project of secure and portable medical-social folder [4]. While PlugDB is simpler than MILo-DB, it shares enough commonalities with MILo-DB design to allow this calibration.

Let us note that we cannot run existing lite or embedded existing DBMS systems in a SMCU because they cannot adapt to its tiny RAM (most products could even not execute within a SMCU). DBMS systems designed for SMCUs would match the RAM constraint but they all consider eXecute In Place (XIP) memories (e.g., EEPROM in [28], NOR Flash in [11]) enabling bit/word access granularity and updates in place. This is incompatible with NAND Flash which exhibits block/page access granularity and forbids updates in place.

7.1 Insertion cost and tuple lifecycle

This section assesses the benefit of the log-only strategy in terms of write I/Os to the NAND Flash.

[16] This calibration is important to take into account aspects that cannot be captured by the simulator (e.g., synchronizations problems when accessing the Flash memory). It impacts negatively the performance shown here roughly by a factor of 1.4.

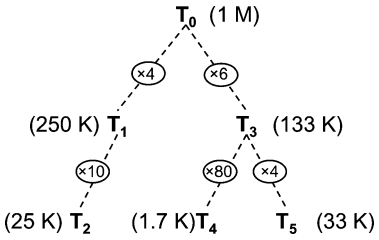


Fig. 7 Synthetic database schema and tables cardinalities

We consider a synthetic database with 6 tables, T_0 to T_5 (see Fig. 7). Each of the 6 tables has 5 indexed attributes ID, Dup10, Dup100, MS1, MS10. ID is the tuple identifier, Dup10, DUP100, MS1 and MS10 are all CHAR(10), populated such that exact match selection retrieves respectively 10 tuples, 100 tuples, 1 % and 10 % of the table. Including the required foreign keys and other non-indexed attributes, the tuple size reaches 160 bytes. The tables are populated uniformly. We build a massively indexed schema holding 64 TSelect indexes, from which 29 for T_0 (5 from each sub-table + 4 defined at T_0 level) and 8 TJoin indexes (stored in 3 dedicated tables associated to T_0, T_1 and T_3).

Let us first study the insertion cost of a single tuple. The number of impacted indexes depends on the table, and so is the insertion cost. Insertion cost varies between 1.5 ms for T_2 to about 8 ms for T_0, and therefore does not constitute a bottleneck even for massively indexed tables. To illustrate the benefit with respect to state of the art techniques, we used the performance numbers of the 20 SD cards tested in [30] and evaluated the insertion cost of one tuple with indexes built as classical B^+-Trees on top of a FTL. Averaging the SD cards performance, this cost varies between 0.8 s (T_2) to 4.2 s (T_0). The minimal insertion cost (best SD card, table T_2) is 42 ms while the maximal cost (worst SD card, table T_0) is 15 s. Moreover, these numbers obviously do not include any technique for managing versions (to avoid replay attacks).

7.2 Tuple lifecycle and scalability limit

The numbers presented above must be put in perspective with the cost incurred by future reorganizations. Thus, we compare the write IOs induced by the insertion of a single tuple during the complete lifecycle of the database (i.e., through all reorganizations) with the same operation over a FTL without reorganization.

Considering the final size of the database (1 M tuples in T_0) and a scalability limit such that $\downarrow$DB = 100 K (i.e., 100 K tuples in T_0), the total number of IOs induced by a single tuple is between 4 and 5, depending on the table. This number rises between 12 and 17 with $\downarrow$DB = 10 K (the smaller the scalability limit, the higher the number of reorganizations). The reason for these surprisingly small numbers is twofold: (1) during the initial insertion in $\downarrow$DB, buffers factorize the write cost of all elements inserted synchronously (e.g., attributes and entries of indexes of the same table, see Sect. 6.1); (2) the reorganization incurs the rewriting of the complete database once, the part of this cost attributed to each tuple being proportional to its size (including indexes). In our evaluation, this size is at maximum 300 bytes, leading to about 1/7 IO per tuple for each reorganization. In contrast, an insertion through a FTL would produce between 12 (T_2) and 31 (T_0) random IOs, under the favorable hypothesis that

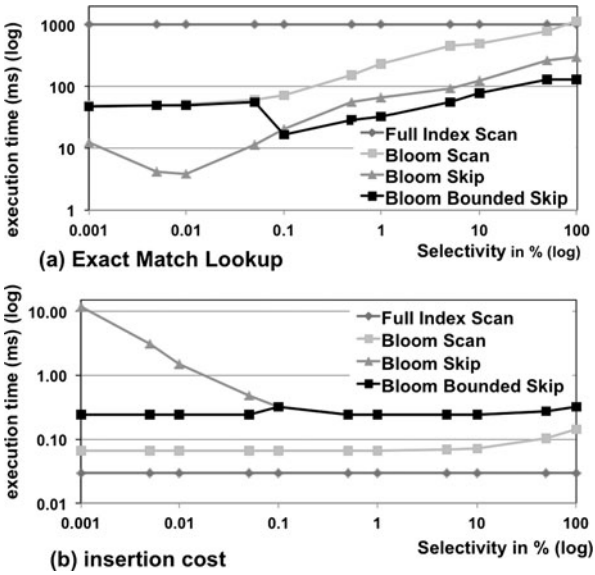


Fig. 8 Performance of (unreorganized) log-only indexes

each insertion in a B^+-Tree index generates a single IO. Each of these IO in turn generates p physical writes, where p is the write amplification factor of the considered FTL.[17] Thus, MILo-DB not only speeds-up the insertion cost at insertion time but also reduces the *total write cost*, thus reducing energy consumption and maximizing the NAND flash lifetime.

7.3 Performance of log-only indexes

We now focus on the cost of exact match lookups and insertions for the log-only indexes, before any reorganization, detailed in Sect. 4. Contrary to conventional databases, indexes can be beneficial even with very low selectivity; indeed, random or sequential reads on NAND Flash (with no FTL) have the same performance. Therefore, it makes sense to vary selectivity up to 100 %. To this end, we built a single table of 300 K records, stored in ↓DB, with 11 attributes, populated with a varying number of distinct values (3, 15, 30, ..., up to 300 K), uniformly distributed.

Figure 8a reports the results for exact match index lookups and Fig. 8b for index insertions (in one single index). For insertions, we considered the worst case (i.e., inserting a single tuple, then committing, thereby limiting the benefit of buffering).

We compare the index proposed in Sect. 4 with 3 other log-only indexes: *Full Index Scan* has neither PTR nor BF and is thus limited to a full scan of the KEY LC. It is used as a baseline. *Bloom Scan* has no PTR but can use the BF to avoid the full scan of KEY LC. Bloom Scan can be seen as a trivial extension of PBFilter [36] for secondary indexes. *Bloom Skip* has PTR, BUF and KEY but does not bound

[17] Note that giving a value of p for simple Flash devices like SD cards is difficult since FTL code is proprietary. It is however necessarily rather large because of their reduced cache capabilities. This is confirmed by the ratio between sequential and random writes (between 130 and 5350! [30])

the insertion cost (i.e., the pointer chain is never interrupted). Finally, we name our proposed index *Bloom Bounded Skip*.

As expected, *Full Index Scan* has a very good insertion cost and a very bad lookup performance (full scan whatever the selectivity). *Bloom Scan* reaches pretty good insertion costs but lookups do not scale well with low selectivity predicates (the whole BF and KEY need to be scanned). Conversely, *Bloom Skip* performs better in terms of lookup but induces very high chaining costs when the inserted value is infrequent (BF needs to be scanned to find the previous occurrence of the inserted value). *Bloom Bounded Skip* appears as the best compromise, with a bounded insertion cost (with selectivity higher than 0.1 %, the pointer chain is broken), and very good lookup costs. With low selectivity predicates, it even outperforms any other index, including *Bloom Skip,* because pointers accessed in PTR are smaller in size.

7.4 Performance of the overall system

This section analyzes the behavior of the whole system considering the query cost of several types of queries, the impact of updates and deletes and the reorganization cost.

We first focus on the query cost and run a set of 18 queries (see Table 1): 12 queries, termed $Mono_i$, involve an exact match selection predicate on a single table on attribute ID, DUP10 or DUP100, joins this table up to T_0 and projects one attribute per table. 3 queries, termed $Multi_i$, involve 2 or 3 exact match predicates on MS1 or MS10. Finally, 3 queries, termed $Range_{i,}$, involve a range predicate on T_5.DUP100. This attribute is therefore indexed using an adequate bitmap encoding as proposed in [13] enabling range predicate evaluation using a Full Index Scan. We measured the query response time with 3 settings: (1) the database has just been reorganized and thus $\downarrow$DB $= \emptyset$; (2) $\downarrow$DB $=$ 10 K; (3) $\downarrow$DB $=$ 100 K. Figure 9 presents the measurements for the 18 queries, ordered by the number of resulting tuples (X axis). We split the graph in two in order to have different scales for response time (0–400 ms, 0–10 s).

For selective queries (1–799 results), selection cost is relatively important with large $\downarrow$DB (100 K) while with $\downarrow$DB $=$ 10 K the response time is very near the reorganized one. Considering several predicates (Multi1) increases this cost, as expected. Finally, the cost of Mono1 is almost zero because it retrieves a T_0 having a specific ID, which is, in our setting, the tuple's physical address.

For less selective queries (1 K–60 K results), the cost is dominated by the projection. Consequently, the $\downarrow$DB size has little influence.

Regarding updates and deletes, measurements on $\downarrow$DB $=$ 10 K and $\downarrow$DB $=$ 100 K have been done after having randomly deleted 3000 tuples (with cascade delete option) and having updated 3000 tuples (uniformly distributed on DUP10, DUP100, MS1, MS10 and A1 attributes of each table). We observed little influence of updates and deletes on performance, because they are evenly distributed. Considering more focused updates on the queried attribute value would lead to larger degradations, which stay however rather limited thanks to the indexation of $\downarrow$UPD.

Table 1 Algebraic expression of the 18 queries

Name	NbRes	Algebraic expression
Mono1	1	$\pi_{T_0.A_1}(\sigma_{ID=500000}(T_0))$
Mono2	8	$\pi_{T_0.A_1,T_3.A_1}(T_0 \bowtie \sigma_{ID=10000}(T_3))$
Mono3	10	$\pi_{T_0.A_1}(\sigma_{DUP_{10}=\text{'VAL\_50000'}}(T_0))$
Mono4	30	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{ID=5000}(T_5))$
Mono5	80	$\pi_{T_0.A_1,T_3.A_1}(T_0 \bowtie \sigma_{DUP_{10}=\text{'VAL\_1000'}}(T_3))$
Mono6	100	$\pi_{T_0.A_1}(\sigma_{DUP_{100}=\text{'VAL\_5000'}}(T_0))$
Multi1	100	$\pi_{T_0.A_1,T_1.A_1,T_2.A_1,T_3.A_1,T_4.A_1,T_5.A_1}(T_0 \bowtie T_1 \bowtie \sigma_{MS_1=\text{'VAL\_50'}}(T_2) \bowtie T_3 \bowtie T_4 \bowtie \sigma_{MS_1=\text{'VAL\_50'}}(T_5))$
Mono7	300	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{10}=\text{'VAL\_500'}}(T_5))$
Mono8	599	$\pi_{T_0.A_1,T_3.A_1,T_4.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{ID=1000}(T_4))$
Mono9	799	$\pi_{T_0.A_1,T_3.A_1}(T_0 \bowtie \sigma_{DUP_{100}=\text{'VAL\_1000'}}(T_3))$
Multi2	998	$\pi_{T_0.A_1,T_1.A_1,T_2.A_1,T_3.A_1,T_4.A_1,T_5.A_1}(T_0 \bowtie T_1 \bowtie \sigma_{MS_{10}=\text{'VAL\_5'}}(T_2) \bowtie T_3 \bowtie \sigma_{MS_{10}=\text{'VAL\_5'}}(T_4) \bowtie \sigma_{MS_{10}=\text{'VAL\_5'}}(T_5))$
Mono10	2997	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{100}=\text{'VAL\_50'}}(T_5))$
Range1	2997	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{100}>\text{'VAL\_99'}}(T_5))$
Mono11	5995	$\pi_{T_0.A_1,T_3.A_1,T_4.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{10}=\text{'VAL\_100'}}(T_4))$
Multi3	9971	$\pi_{T_0.A_1,T_1.A_1,T_2.A_1,T_3.A_1,T_4.A_1,T_5.A_1}(T_0 \bowtie T_1 \bowtie \sigma_{MS_1=\text{'VAL\_50'}}(T_2) \bowtie T_3 \bowtie T_4 \bowtie \sigma_{MS_1=\text{'VAL\_50'}}(T_5))$
Range2	14955	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{100}>\text{'VAL\_95'}}(T_5))$
Range3	29912	$\pi_{T_0.A_1,T_3.A_1,T_5.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{100}>\text{'VAL\_90'}}(T_5))$
Mono12	59881	$\pi_{T_0.A_1,T_3.A_1,T_4.A_1}(T_0 \bowtie T_3 \bowtie \sigma_{DUP_{100}=\text{'VAL\_10'}}(T_4))$

7.5 Reorganization cost and scalability limit

Let us now consider the reorganization cost. We consider a fixed size for *DB (900 K tuples in T_0) and vary the size of ↓DB (varying the T_0 table from 10 K to 100 K tuples, other tables growing accordingly). The reorganization cost varies linearly from 6.2 min (for 10 K) to 7.7 min (for 100 K) (see Fig. 10a). It is worth noting that reorganization does not block queries (it can be efficiently stopped and resumed). The reorganization cost results from (1) reorganizing ↓IND, ↓DEL and ↓UPD and (2) reading $*DB_i$ and rewriting $*DB_{i+1}$. The Flash "consumption", i.e., the quantity of Flash memory written during one reorganization, varies between 0.5 GB (for 10 K) and 0.9 GB (for 100 K) (see Fig. 10b). However, 100 reorganizations are required to reach 1 M tuples in T_0 with ↓DB of 10 K, while only 10 are necessary with ↓DB of 100 K. In any case, the Flash lifetime does not appear to be a problem.

As a final remark, note that the FTL approach, already disqualified due to its write behavior, is not a good option in terms of query performance either. Indeed, except for highly selective (then cheap) queries where the FTL approach can perform slightly better than the log-only approach, the FTL overhead incurred by traversing translation tables makes the performance of less selective (therefore costly) queries much worse than with MILo-DB. Indeed, the high number of IOs generated at projection time dominates the cost of these queries.

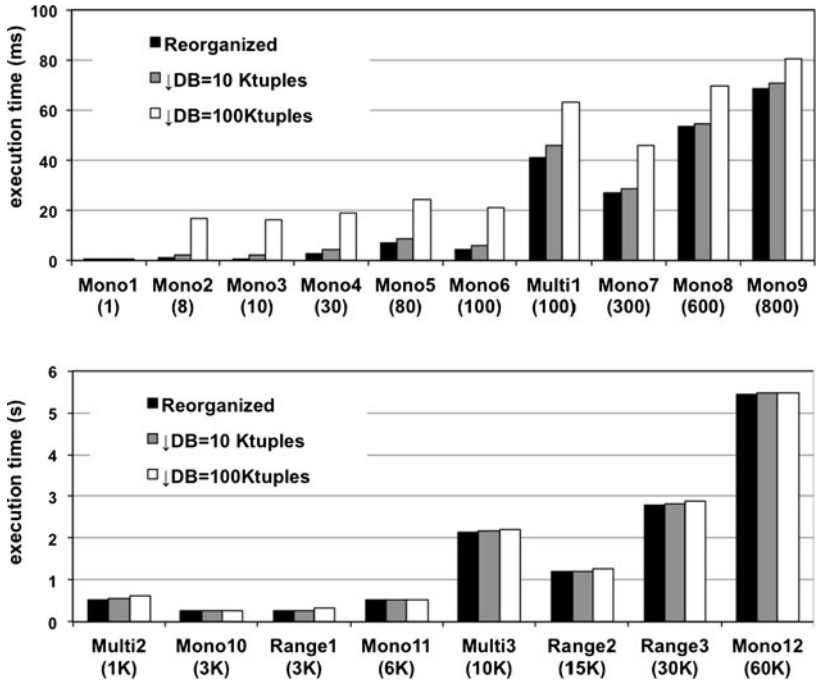


Fig. 9 Performance of 18 queries with different settings

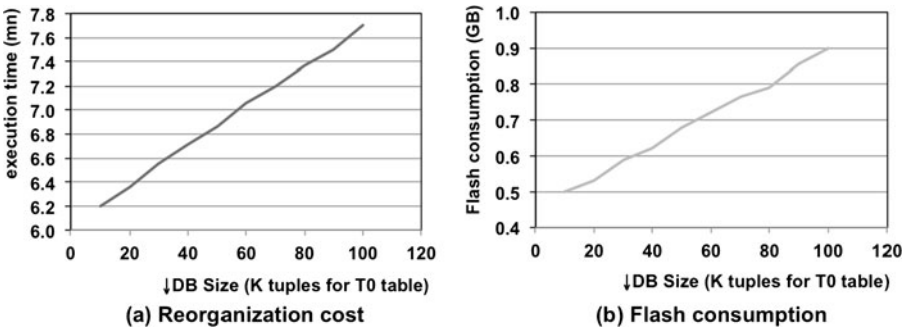


Fig. 10 Reorganization cost and flash consumption

8 Conclusion

This paper proposed a new approach based on log-only data structures to tackle the conflicting NAND Flash/tiny RAM constraints inherent to Secure Tokens. It has shown the effectiveness of the approach to build a complete embedded DBMS engine called MILo-DB. MILo-DB is well adapted to data insertions, even at high rate, even with massive indexing, and supports gracefully a reasonable amount of updates and deletes. It offers an efficient evaluation of relational queries with a tiny RAM on gigabyte sized datasets and can protect these datasets against any form of confiden-

tiality and integrity attacks. Hence, MILo-DB perfectly matches the requirements of a personal secure server where data (either stored locally or in the Cloud) needs to be managed under holder's control.

MILo-DB is however not adapted to all contexts. For instance, high updates/deletes rates would trigger too frequent reorganizations, long transactions and complex queries (e.g., involving ad-hoc joins or nested queries, etc.) are not yet supported. We currently investigate part of those issues but some of them are inherent to the approach. We are also adapting the design to cope with a very small number of Log Containers (typically less than 4) to efficiently address contexts where the FTL cannot be bypassed (e.g., SD cards).

This paper has shown that managing a crypto-protected gigabyte sized database within a secure token is not pure utopia. Beyond the interest for PIM applications, the major technical contribution is being able to manage a complete database without generating any random write. This result may have a wider applicability, in every context where random writes are detrimental in terms of I/O cost, energy consumption, space occupancy or memory lifetime.

Acknowledgements This work has been partially funded by the French ANR KISS project under grant No. ANR-11-INSE-0005. The authors also wish to thank Philippe Bonnet for his accurate comments on early versions of this paper.

References

1. Agrawal, D., Abbadi, A.E., Wang, S.: Secure data management in the cloud. In: DNIS (2011)
2. Agrawal, D., Ganesan, D., Sitaraman, R., Diao, Y., Singh, S.: Lazy-adaptive tree: an optimized index structure for flash devices. In: PVLDB (2009)
3. Allard, T., Anciaux, N., Bouganim, L., Guo, Y., Le Folgoc, L., Nguyen, B., Pucheral, P., Ray, I., Ray, I., Yin, S.: Secure personal data servers: a vision paper. In: PVLDB (2010)
4. Allard, T., Anciaux, N., Bouganim, L., Pucheral, P., Thion, R.: Trustworthiness of pervasive healthcare folders. In: Pervasive and Smart Technologies for Healthcare, Information Science Reference (2009)
5. Anciaux, N., Benzine, M., Bouganim, L., Pucheral, P., Shasha, D.: Revelation on demand. In: DAPD (2009)
6. Anciaux, N., Bouganim, L., Guo, Y., Pucheral, P., Vandewalle, J.J., Yin, S.: Pluggable personal data servers. In: SIGMOD (2010)
7. Arge, L.: The buffer tree: a technique for designing batched external data structures. Algorithmica (2003)
8. Bernstein, P., Reid, C., Das, S.: Hyder—a transactional record manager for shared flash. In: CIDR (2011)
9. Bityutskiy, A.B.: JFFS3 design issues. Tech. report (2005)
10. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM (1970)
11. Bolchini, C., Salice, F., Schreiber, F., Tanca, L.: Logical and physical design issues for smart card databases. In: TOIS (2003)
12. Bursky, D.: Secure microcontrollers keep data safe. PRN engineering services (2012). http://tinyurl.com/secureMCU
13. Chan, C.Y., Ioannidis, Y.E.: An efficient bitmap encoding scheme for selection queries. In: SIGMOD (1999)
14. Debnath, B., Sengupta, S., Li, J.: SkimpyStash: RAM space skimpy key-value store on flash. In: SIGMOD (2011)
15. Elbaz, R., Champagne, D., Lee, R.B., Torres, L., Sassatelli, G., Guillemin, P.: TEC-tree: a low-cost, parallelizable tree for efficient defense against memory replay attacks. In: CHES (2007)
16. Eurosmart: Smart USB token. White paper (2008)

17. Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: a personal database for everything. Commun. ACM **49**(1) (2006)
18. Giesecke devrient: portable security token. http://www.gd-sfs.com/portable-security-token
19. Haas, L.M., Carey, M.J., Livny, M., Shukla, A.: Seeking the truth about ad hoc join costs. VLDB J. (1997)
20. Bonnet, P., Bouganim, L., Koltsidas, I., Viglas, S.D.: System co-design and date management for flash devices. In: PVLDB (2011)
21. Li, Y., He, B., Yang, R.J., Luo, Q., Yi, K.: Tree indexing on solid state drives. In: PVLDB (2010)
22. Li, Z., Ross, K.A.: Fast joins using join indices. VLDB J. (1999)
23. Lim, H., Fan, B., Andersen, D., Kaminsky, M.: SILT: a memory-efficient, high-performance key-value store. In: SOSP (2011)
24. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A., Rivest, R.L.: Handbook of Applied Cryptography. CRC Press, Boca Raton (2001)
25. Moglen, E.: FreedomBox. http://freedomboxfoundation.org
26. Muth, P., O'Neil, P., Pick, A., Weikum, G.: The LHAM log-structured history data access method. VLDB J. (2000)
27. O'Neil, P., Cheng, E., Gawlick, D., O'Neil, E.: The log-structured merge-tree (LSM-tree). Acta Inform. (1996)
28. Pucheral, P., Bouganim, L., Valduriez, P., Bobineau, C.: PicoDBMS: scaling down database techniques for the smart card. VLDB J. (2001)
29. Rosenblum, M., Ousterhout, J.: The design and implementation of a log-structured file system. ACM Trans. Comput. Sci. (1992)
30. Schmid, P., Roos, A.: SDXC/SDHC memory cards, rounded up and benchmarked. http://tinyurl.com/tom-sdxc
31. Severance, D., Lohman, G.: Differential files: their application to the maintenance of large databases. ACM Trans. Database Syst. (1976)
32. Sundaresan, P.: General key indexes. US Patent No. 5870747 (1999)
33. Vo, H.T., Wang, S., Agrawal, D., Chen, G., Ooi, B.C.: LogBase: scalable log-structured storage system for write-heavy environments. Technical report (2012)
34. Weininger, A.: Efficient execution of joins in a star schema. In: SIGMOD (2002)
35. Wu, C., Chang, L., Kuo, T.: An efficient b-tree layer for flash-memory storage systems. In: RTCSA (2003)
36. Yin, S., Pucheral, P., Meng, X.: A sequential indexing scheme for flash-based embedded systems. In: EDBT (2009)

Annexe E.

Limiting Data Collection in Application Forms

A real-case Application of a Founding Privacy Principle

Nicolas Anciaux, Benjamin Nguyen, Michalis Vazirgiannis

International Conference on Privacy, Security and Trust (PST), pp. 59-66, 2012

Limiting Data Collection in Application Forms

A real-case application of a Founding Privacy Principle

Nicolas Anciaux[1,2]
[1] INRIA, Le Chesnay, France
[2] U. de Versailles St-Quentin, Versailles, France

Benjamin Nguyen[1,2]
[2] U. de Versailles St-Quentin, Versailles, France

Michalis Vazirgiannis[3,4]
[3] Athens U. of Economics & Business, Athens, Greece
[4] LIX, Ecole Polytechnique, Palaiseau, France

Abstract— **Application forms are often used by companies and administrations to collect personal data about applicants and tailor services to their specific situation. For example, taxes rates, social care, or personal loans, are usually calibrated based on a set of personal data collected through application forms. In the eyes of privacy laws and directives, the set of personal data collected to achieve a service must be restricted to the minimum necessary. This reduces the impact of data breaches both in the interest of service providers and applicants. In this article, we study the problem of limiting data collection in those application forms, used to collect data and subsequently feed decision making processes. In practice, the set of data collected is far excessive because application forms are filled in without any means to know what data will really impact the decision. To overcome this problem, we propose a reverse approach, where the set of strictly required data items to fill in the application form can be computed on the user's side. We formalize the underlying NP Hard optimization problem, propose algorithms to compute a solution, and validate them with experiments. Our proposal leads to a significant reduction of the quantity of personal data filled in application forms while still reaching the same decision.**

Privacy principle; Limited collection; Automated form filling.

I. INTRODUCTION

A massive digitalization of personal information is currently underway. Individuals are receiving an ever increasing amount of important documents in digital form (financial, professional, medical, relative to insurance, administrative, linked to daily consumption, etc.), issued by their employers, banks, insurances companies, civil authorities, hospitals, schools, ISP, telcos, etc. In parallel, secured online personal stores are emerging. The domain of the personal cloud is flourishing, and a recent report forecasts a $12 billion market[1]. Alternative offers propose storage facilities on the user's side with extended privacy controls, like for example Personal Data Servers [2] or Plug Servers (e.g., FreedomBox[2]).

This thriving market attests a reality: official documents are continuously accumulated and treasured by their owner. The reason is simple: legal obligations require them to be kept (e.g., 1 year for bank statements) and these documents are used as evidence when performing subsequent administrative tasks (e.g., paying taxes) or applying to services (e.g., bank loans).

In this paper, we consider the interaction between an applicant and a service provider, where the service provider requests personal information about the applicant to select the appropriate best offer. Such interactions occur whenever services are calibrated to adapt to the particular situation of each user. For example, the characteristics of a personal loan (rate, duration, insurance fee…) are defined based on decision making processes which use personal information such as income, employment, title deeds, personal references, forms of collateral, medical records, past lines of credits, etc. To cite other examples, contracting an insurance (health, car, job protection, etc.), social assistance, tax refund, or more generally any kind of personal information describing one's specific situation, to customize the offer that is made.

The necessity of evaluating the particular situation of an applicant is unquestionable and is in the interest of both the service provider and the customer. However, the requested set of personal information must be restricted to the minimum for two main reasons. First, the privacy of the applicant must be protected. Privacy legislations worldwide [13], [19] have enacted the *Limited Data Collection* (*LDC*) principle to this end, stating that collected sets of personal data must be strictly restricted to the minimum necessary to achieve the goal the user consents to. Second, the cost of potential information leakage must be reduced. Indeed, all too often personal data ends up being disclosed by negligence or hack. In 2011, the Open Security Foundation[3] reported more that a thousand data loss incidents affecting more than a hundred millions records. This is a financial disaster for the companies in charge of the data. A recent study [20] estimates the cost of data breaches at an average $7.2million per incident. Indeed, data breach laws enacted in many countries including 46 US states and the EU, compel companies to notify data owners in the event of data breaches, assist the victims in minimizing the impact of the data leak (e.g., canceling their credit card if the number has been disclosed) and often incur financial compensations. Security companies provide online breach cost calculators[4] to draw attention to this phenomenon: the more data exposed, the greater the cost in the event of a data breach.

The target of this paper is to restrict the set of information users have to expose to service providers, in accordance with the *LDC* privacy principle, and without impacting the evaluation of the decision making processes.

[1] The Personal Cloud: Transforming Personal Computing, Mobile, And Web Markets, Frank Gillett, a Forrester report, June 2011.
[2] See http://freedomboxfoundation.org/
[3] See http://www.datalossdb.org/reports
[4] See http://databreachcalculator.com.sapin.arvixe.com/

This is a difficult problem. In practice, the data useful or useless to make the decision cannot be distinguished *a priori* (at collection time). Such an assumption only holds for very simple cases, e.g., when ordering online, the address of the customer is mandatory to deliver the purchased items. However, in a general decision making system it does not hold. What data is useful to come to the decision of lowering the rate of the loan proposed to a user? Not only does the information harvesting depend on the purpose, it also depends on the data itself. Consider a reduction of rate based on either the salary or the assets of an individual. Revealing her income of *$30.000* if her age is below *25* may be enough. But an income of *$50.000* would suffice, regardless of age. Maybe *both* income and age values are useless if sufficient assets (e.g., greater than *$100.000*) can be justified. For a user with values u_1=[*income=$35.000*, *age=21*, *assets=$10.000*] the minimum data set would be [*income*, *age*]. For a user with u_2=[*income=$40.000*, *age=35*, *assets=$250.000*] it would be [*assets*]. Hence, a bank cannot specify a minimum set of attributes needed to make its decision since this decision depends on looking at the entire attributes available. Fixing the data to be collected *a priori* inevitably leads to over-estimating the data to be collected.

The common procedure when a server has to evaluate a decision making process is thus to request the users to fill in application forms which cover all the information which *may turn out to be of use* at some point in the decision process. This obviously does not comply with the *LDC* principle, since service providers collect personal data which may *not* impact the final decision to be taken. Our study focuses on the strict compliance with the *LDC* principle in this context. Our approach is based on a reverse implementation of the traditional *LDC* strategy, where users are given enough knowledge about the underlying decision making process to determine locally the minimum set of data items to fill in to achieve the expected service with maximum benefit.

To the best of our knowledge, all existing techniques addressing *LDC* principle fix the data to be collected a priori, leading to collect too much data. A few recent works in the domain of credential based access control can be viewed as vanguards in the application of the *LDC* principle. However, the underlying techniques cannot be used to solve our problem, mainly because of scalability issues. We give details about the positioning of our work in Section VI.

The contribution we make in this paper is threefold:

(i) we formalize the *Minimum Exposure* approach in the case of a decision making process;
(ii) we state the underlying optimization problem and study its complexity; and
(iii) since the problem is NP-hard we propose approximation algorithms and validate them with experiments.

The paper is organized as follows. Section II gives the general scenario, and presents the running example. In Section III, we state the *Minimum Exposure* optimization problem and study its complexity. Several algorithms are introduced in Section IV, and validated in Section V. Section VI discusses related works and Section VII concludes.

II. SCENARIO FOR MINIMUM EXPOSURE

A. General Scenario

We consider the general scenario depicted in Figure 1 which involves three main parties: Data Producers, Users, and Service Providers. **Data Producers** act as data sources. They include for example banks, employers, hospitals, or administrations. The information they deliver to users can be signed to prove integrity and origin (e.g., salary forms, bank records history, tax receipts, etc.). **Users** store the documents they receive in their personal spaces. We make no hypothesis on users' personal space, which could be their own PC, cloud storage, or secure devices, etc. **Service Providers** may include banks or insurances companies, but also public welfares or administrations. They propose services, which may include bank loans, health insurance or social benefits, which require users' personal information to evaluate the decision making processes and calibrate the offer made to the applicant.

In practice, Service providers issue application forms to collect the data which may impact their decision. Huge amounts of data items may be requested, depending on the context. For example, loan applications may include mortgage application forms, which commonly collect hundreds of personal data items [5], and social care applications require equivalently large forms to be filled in.

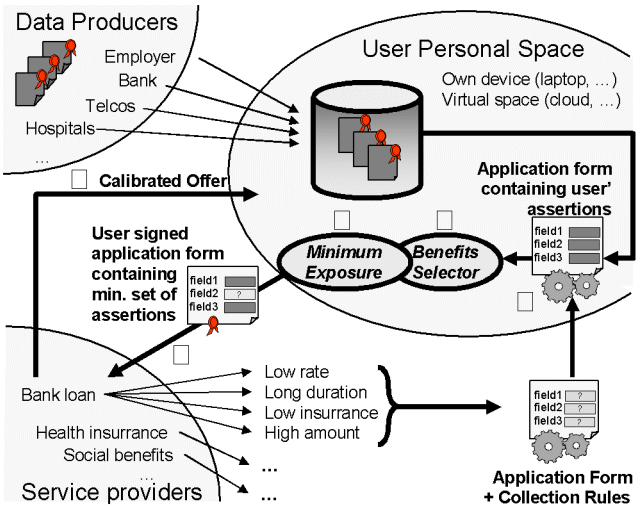


Figure 1. General architecture enabling Minimum Exposure.

We promote in this paper a new approach where the server has to provide both the application form and a set of data *collection rules*. Those collection rules enable the users to select among the data items requested in the application form the minimum required set to be filled in. We call *Minimum Exposure* (*ME*) the process which identifies the minimum subset of assertions to be exposed by a User to a Service Provider to trigger the desired service with the set of advantages she can (and wants) to obtain. *ME* requires

[5] See for example the mortgage application form of Nationwide Building Society (the largest building society in the world) as a good representative: http://www.nationwide.co.uk/nr/rdonlyres/a48ffc87-7e29-4ea6-b24d-2720746c5d9e/0/m1inov06.pdf

confronting the set of assertions that can be made by the User, with the advantages associated with the set of collection rules describing the information requested by the Service Provider.

The execution of *ME* must take place on the user's side or on any trusted third party, to comply with the *LDC* principle. Indeed, the system in charge of running *ME* needs to collect more data than the minimum subset computed by *ME*.

The general scenario pictured on Figure 1 is as follows: when a user wants to apply to a service, she downloads the application forms and the collection rules provided by the service provider and fills in the form given the documents she owns, uses the *Benefit Selector* to locally compute the advantages she can obtain based on the completed application form and the collection rules, and selects among these advantages the ones she desires to obtain, runs a *Minimum Exposure* process to compute the minimum set of useful assertions (i.e., data items provided) in the application form to obtain the service with the selected advantages, validates and signs the application form with this minimum set of assertions and sends it to the service provider. The service provider can run its decision processes based on the content of the form and calibrate the offer made to the applicant. The declaration of the applicant is then stored by the service provider (the delay depends on the context). After point the applicant can be asked to provide certified documents to prove the assertions provided in her form. Remark that in many cases the verification phase only occurs unlikely, e.g, medical declarations linked to car or loan insurance are often only checked in the event of a claim, income taxe returns in the case of a tax audit, etc.

B. Setting

1) Collection Rules

The collection rules describe the information required by the service provider and the advantages associated with it.

Decision making processes are generally based on *white box* mechanisms (i.e., comprehensible by humans and justifiable) and are thus public. This is the case of administrative applications (e.g., tax services, social and health care where the decision process is either publicly documented or considered as common knowledge), and law abiding commercial applications. Indeed, as shown in [15], a white box requirement is imposed by law and/or for user acceptance of the process in many domains. For example, laws such as the US "Equal Credit Opportunity Act" impose white box for credit scoring[6]. This is the same for many medical systems. Recent studies like [7] even transform "black box" decision models like SVM or neural networks into white box ones.

In addition, decision making rules may be complex in practice, e.g., loans are granted based on decision trees, SVM or neural networks [12]. The collection rules must be expressive enough to successfully reflect the decision making process of the service provider. In this paper, we consider sets of collection rules, each one being modeled using disjunctions of conjunctions of constraints on attribute-value pairs. This is a comprehensible rule-based model, which is very expressive since it covers the widely used decision tree model [17], as well as forests of decision trees (i.e., covering multi-dimensional decisions). For example, an organization offering loans may include a dimension in its decision making rules to favor families and young students by subsidizing a part of the loan as a *Non Interest Loan* (*NILo*), expressed by the following rule:

$$NILo: (married=true \wedge children>0) \vee (age<30 \wedge Edu='Univ')$$

We assume that no-one can force users to transmit assertions. The only penalty is to prevent them from obtaining advantages. Therefore, rules must be *positive*, in the sense that it is beneficial for a user to trigger them. This is not a limitation of the model since rules leading to constraints that prevent the grant of services (called *negative* rules) can be constructed by integrating the negation of the rule into the collection rule set. For example, if the *NILo* mentioned above is *not* granted to people with a police record ($police\_record='YES' \Rightarrow \neg NILo$), the rule can be written:

$$NILo: (married=true \wedge children>0 \wedge police\_record='NO') \vee (age<30 \wedge Edu='Univ' \wedge police\_record='NO')$$

2) Users' Assertions and Documents

We distinguish two types of documents: assertion documents (simply called assertions) used to fill in application forms; and official documents signed by data producers and kept by users to be shown if service providers request proving assertion validity. Application forms are signed as a whole by applicants and official documents are signed by data producers at various granularities. In case of verification of an assertion, e.g., $income = \$50.000$, the user may disclose an official document containing this value, e.g., the income tax receipt.

For the sake of simplicity and without lack of generality, we consider in this paper that each assertion is an inseparable (*attribute, value*). This is the format in which application forms generally require personal information. We show in [3] that our model can also be extended to *attribute* θ *value* assertions, with θ the comparator $<, \leq, =, \neq, \geq$, or $>$, leading to expose even less information with respect to collection rules. Regarding official documents, most are currently signed as a whole, but there is no technical difficulty to sign (*attribute, value*) pairs or *attribute* θ *value* separately as show in [8]. Thus, we also consider that each assertion d can be proven by a given official documents d' without leaking more information than d. We provide in [3] a discussion when this hypothesis is not fulfilled.

3) Metrics to Evaluate the Degree of Exposure

The minimization of the set of assertions resulting from the application of the *ME* algorithm can be appreciated in terms of reduction of the data (assertions) exposed, harmful to both user (in terms of privacy) and service provider (in terms of financial cost). We consider on the one hand that the privacy harm associated with a dataset is proportional to the usefulness of that dataset, and on the other hand that the financial cost of a data breach for service providers is directly proportional to the quantity of exposed data. The financial cost for service providers is determined by two dominant factors [20]. First, the *ex-post response* represents 20% of the cost. It includes the actions taken by the company to provide assistance to the

[6] See http://www.ftc.gov/bcp/edu/pubs/consumer/credit/cre15.shtm

victims in the necessary procedures conducted to minimize the harm: the greater the exposure, the greater the harm. Second, *lost business* (50% of data breach cost) is the direct consequence of the negative publicity associated with the data breach incident headings: the greater the exposure, the worse the publicity.

These components of the breach costs are thus tightly linked with information loss, both for users and service providers. Many information loss metrics exist (e.g., *minimal distortion* [21], [22] or *ILoss* [23]). They all associate an exposure value to each dataset item independently; our approach can be used with any such metric.

C. Running Example

We introduce here a loan scenario, used as a running example (see Table I) throughout the paper. The example has deliberately been simplified, since real loan application forms may include mortgage/medical forms that collect hundreds of personal data items.

An institution proposes, to any applicant, personal loans of *$5.000* at *10%* rate with *1* year duration and a *$50* per month insurance cost for job loss protection. But, a higher loan of *$10.000* can be offered to wealthy customers fulfilling the following requirement:

$$(income{>}\$30.000 \wedge assets{>}\$100.000) \vee (collateral{>}\$50.000 \wedge life\_insurance{=}'yes')$$

This leads to the first collection rule r_1 given in Table I. Collection rule r_2 enables obtaining a loan granted at only *5%* rate for families and low risk factor young people; collection rule r_3 expresses that loans can be granted for an extended duration of *2* years to high revenues families and to low risk people; and rule r_4 states that the insurance cost for job loss protection can be proposed with a *30%* discount to rich families and promising young workers. Those collection rules are made of a disjunction of conjunction of predicates p_i of the form *attribute* θ *value* with θ the comparator <, ≤, =, ≠, ≥, or >. They are given with the corresponding application form in table I.

The user can assert that she is married, 25 years old, with one child, a $35.000 year income, a university degree, $5.000 goods as collateral, an income tax at 11.5% rate, a life insurance. She also claimed only $250 last year. This information is summarized by a set of *attribute = value* assertion termed as_1 to as_{10} in Table I such that $as_i \Rightarrow p_i$. This user could then activate the complete set of advantages c_1 to c_4. The *ME* algorithm has to identify the minimum set of assertions allowing this.

III. THE MINIMUM EXPOSURE PROBLEM

This section first states the *Minimum Exposure* problem more formally and studies its complexity.

A. Problem Statement

We denote by $|S|$ the cardinality of a set S. We introduce below the other required definitions, and then state the problem. We illustrate the notions using the example in Table I.

TABLE I. FORM, RULES AND ASSERTIONS FOR THE LOAN SCENARIO.

Collection rules:

r_1: $(p_1 \wedge p_2) \vee (p_3 \wedge p_4) \Rightarrow c_1$

r_2: $(p_5 \wedge p_6 \wedge p_7) \vee (p_4 \wedge p_8 \wedge p_9) \Rightarrow c_2$

r_3: $(p_1 \wedge p_6 \wedge p_7) \vee (p_2 \wedge p_4 \wedge p_{10}) \Rightarrow c_3$

r_4: $(p_2 \wedge p_5 \wedge p_6 \wedge p_7) \vee (p_1 \wedge p_4 \wedge p_8 \wedge p_9) \Rightarrow c_4$

with p_1: *year_inc>$30.000*, p_2: *assets>$100.000*,
p_3: *collateral>$1.000*, p_4: *life_insurance='yes'*,
p_5: *tax_rate>10%*, p_6: *married=true*,
p_7: *children>0*, p_8: *edu='university'*,
p_9: *age<30*, p_{10}: *insurance_claims<$5.000*.

and c_1=*high_loan*, c_2=*5%_rate*,
c_3=*long_loan*, c_4=*low_insurance*.

Application form: year_inc?, collateral?, tax_rate?, children?, age?, assets?, life_insurance?, married?, edu?, insurance_claims?

User's assertions:

as_1:	*year_inc=$35.000*,	as_2:	*assets=$150.000*,
as_3:	*collateral=$5.000*,	as_4:	*life_insurance='yes'*,
as_5:	*tax_rate=11.5%*,	as_6:	*married=true*,
as_7:	*children=1*,	as_8:	*edu='univ'*,
as_9:	*age=25*,	as_{10}:	*insurance_claims=$250*.

1) Definitions

Attributes. Let $A = \{a_i\}$ represent a finite set of attributes. Each attribute a_i has an associated domain $dom(a_i)$.

Classes. Let $C = \{c_j\}$ represent a finite set of Boolean variables, interpreted as *positive* classes to which users can belong. If c_j=*true* for a given user, this means she can obtain the advantage associated with c_j.

Predicates. We call *predicate over A* any expression of the form $a\theta v$ where $a \in A$, $v \in dom(a)$ and $\theta \in \{=, <, >, \leq, \geq, \neq\}$.

Example: p_1: *year_inc>$30.000* is a predicate.

Assertions. Let as_i represent an assertion composed of a single equality predicate over A. We denote by $Data_u = \{as_i\}$ the set of assertions a given user u can state truthfully (i.e. she owns a signed document proving this assertion). We say that an assertion as_i *proves* a predicate p if $as_i \Rightarrow p$.

Atomic Rules. An atomic rule leading to class c_j, denoted by $atom_j$ is a conjunction of predicates such that $atom_j$=*true* ⇒ c_j=*true*. Since there are usually several atomic rules leading to a class c_j we write $atom_{j,k}$ using k to distinguish them.

Example: $atom_{1,1}$: (*year_inc>$30.000* ∧ *assets>$100.000*) and $atom_{1,2}$: (*collateral>$50.000* ∧ *life_insurance='yes'*) are two atomic rules leading to class c_1.

We say that a set of assertions $Data_u = \{as_i\}$ proves an atomic rule $atom_{j,k} = \bigwedge_m q_{j,k,m}$ where $q_{j,k,m}$ is a predicate over A, if and only if $\forall j,k,m \; \exists \, i : as_i \Rightarrow q_{j,k,m}$ and *uniquely proves* $atom_{j,k}$ if and only if $\forall j,k,m \; \exists! \, i : as_i \Rightarrow q_{j,k,m}$.

Example: $data_u = \{as_1, as_2, as_3, as_4\}$ uniquely proves atomic rules $atom_{1,1}$ and $atom_{1,2}$.

Collection Rules. A collection rule r_j is a disjunction of atomic rules leading to class c_j. More formally: r_j: $\bigvee_k atom_{j,k}$. If a signed set of assertions $Data_u$ proves an atomic rule $atom_{j,k}$ then we say that $Data_u$ *proves* r_j, which means that user u can benefit from the advantage associated with c_j (obviously, r_j=*true* ⇒ c_j=*true*).

Example: r_1: ($year\_inc>\$30.000 \wedge assets>\$100.000$) $\vee$ ($collateral>\$50.000 \wedge life\_insurance='yes'$) is a collection rule leading to class *high_loan*.

In what follows, we write $r_j = \vee_k (\wedge_m \ q_{j,k,m})$ where $q_{j,k,m}$ is a predicate over A. Considering r_1 in the previous example, we have $q_{1,1,1}$: $year\_inc>\$30.000$, $q_{1,1,2}$: $assets>\$100.000$, $q_{1,2,1}$: $collateral>\$50.000$ and $q_{1,2,2}$: $life\_insurance='yes'$.

Rule Set. Let $R = \{r_j\}$ represent a set of $|C|$ *collection rules*, one for each class c_j. If $Data_u$ (uniquely) proves all the rules of R then we say that $Data_u$ (uniquely) proves R.

Rule Set Boolean Formula. Since only one (verifiable) assertion uniquely proves a given predicate used in the rules, deciding whether $Data_u$ proves the rule set R is *equivalent* to testing the truth-value of a Boolean formula $E_R=\wedge_j(\vee_k(\wedge_m b_{f(j,k,m)}))$ called Rule Set Boolean Formula associated to R, where $f(j,k,m)$ is a function of domain $[1;|A|]$ defined by $[f(j,k,m)=i$ such that $as_i \Rightarrow q_{j,k,m}]$ and $b_{f(j,k,m)}$ is a Boolean variable which is *true* if $as_{f(j,k,m)}$ is exposed and *false* otherwise. Note that if we consider the truth assignment that sets all values $b_{f(j,k,m)}$ to *true*, then E_R=*true* $\Leftrightarrow$ $Data_u$ *proves* R.

Example: Table II illustrates a Rule Set Boolean Formula based on R defined in Table I.

Exposure metric. Let $B=\{b_i\}$ represent a set of Boolean variables. Let T_B represent a truth assignment of these variables such that b_i=*true* $\Leftrightarrow$ as_i is disclosed to the service provider. We note $\mathbf{EX}(T_B)$ a function representing the exposure of the associated assertions set disclosed. Exposure is proportional to financial cost for service providers, and privacy harm for users.

Example: The function $\mathbf{EX}(T_B) = |\{ b_x \in B: T_B(b_x)=true \}|$ that counts the number of assertions disclosed can be used as an exposure metric. Note that any metric that is invariant over time, given a truth assignment T_B can be used. In particular, this includes information loss metrics, which can be assumed proportional to **EX**. Henceforth, if an assertion is disclosed, we say that it is *exposed*.

Boolean Minimum Exposure Problem. We can now define the Minimum Exposure decision problem of a set of assertions $Data_u$ with regards to a rule set R and an exposure metric **EX**. Note that we suppose that $Data_u$ proves R. Should this not be the case, we would simply use R' the subset of rules of R proven by $Data_u$. Our goal is to find a truth assignment T_B of the Boolean variables associated to the disclosure of the assertions minimizing their exposure computed using the above exposure metric.

The *Boolean ME* decision problem:
Given a rule set R, $Data_u=\{as_x\}$ a set of q assertions that uniquely prove R, B a set of Boolean variables $B=\{b_1,..,b_q\}$ such that b_x=*true* $\Leftrightarrow$ as_x is exposed, $E_R=\wedge_j (\vee_k (\wedge_m \ b_{j,k,m}))$ where $\forall j,k,m \ b_{j,k,m} \in B$ the rule set formula associated to R, and the exposure function **EX,** $Data_u$ is *n-exposable* with regards to R if and only if there exists a truth assignment T_B of B such that $\mathbf{EX}(T_B) \leq n$ and E_R is *true*.

We study the related optimization problem, whose goal is to minimize n.

TABLE II. RULE SET BOOLEAN FORMULA FOR THE LOAN SCENARIO

$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$ such that:
$\forall i \in [1;10]$, b_i=*true* $\Leftrightarrow$ as_i *is exposed.*
The Rule Set Boolean Formula E_R is as follows:
$E_R = \ ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge \ ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge \ ((b_1 \wedge b_6 \wedge b_7) \vee (b_2 \wedge b_4 \wedge b_{10}))$
$\wedge \ ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$
Suppose that the user can only truthfully state assertions 1-9 we prune out all the classes and atomic rules that can not be proven:
$E_R = \ ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge \ ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge \ ((b_1 \wedge b_6 \wedge b_7)$
$\wedge \ ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$

TABLE III. ALGORITHM NOTATIONS USING THE LOAN SCENARIO

$D=|Data_u|=10$; $C=4$;
$B[]$ is an array of Booleans of size D such that:
$\forall i \in [1;10]$, $B[i]$=*true* $\Leftrightarrow$ as_i is *exposed*
$R[]$ is an array of C collection rules;
$R[j].atom[]$ for $j \in [1;4]$ are arrays of *2* atomic rules; $R[j].atom[k].b[]$ with $j \in [1;4]$, $k \in [1;2]$ are arrays of references to $B[i]$ elements. We denote by $*B[i]$ a reference to $B[i]$. $R[j].atom[k].b[m]$are set as follows:
$R[1].atom[1].b[1] \leftarrow *B[1]$; $R[1].atom[1].b[2] \leftarrow *B[2]$;
(…) $R[2].atom[2].b[2] \leftarrow *B[8]$; $R[2].atom[2].b[3] \leftarrow *B[9]$; (…)
$R[4].atom[2].b[3] \leftarrow *B[8]$; $R[4].atom[2].b[4] \leftarrow *B[9]$;

B. Complexity Results

The *ME* problem defined above is NP-Hard (proof is a reduction to the min weighted SAT problem omitted due to lack of space, and can be found in [3]). In addition, we show in [3] that the *ME* optimization problem is not in APX[7], and has a differential approximation[8] ratio of 0-DAPX[9]. This is a negative complexity result in the sense that it shows that the problem is difficult and that polynomial approximation algorithms will provide bad approximation guarantees in the worst case. In Section IV, we examine the problem by (experimentally) exploring the domain where it is possible to provide an exact resolution using a state of the art solver. When such a resolution is too long to compute, we rely on polynomial approximation algorithms.

IV. SOLUTIONS OF THE *ME* PROBLEM

In this section, we provide exact and approximation algorithms to compute a solution of the *ME* problem. For the exact resolution, we use a *Binary Integer Programming* (BIP) state of the art solver. For the approximate resolution, we propose a naïve random algorithm, a simulated annealing based meta-heuristics algorithm, and a specific heuristic algorithm.

[7] The APX class is the set of NP optimization problems that allow polynomial-time approximation algorithms with an *approximation ratio* bounded by a constant.

[8] Given an instance I of an optimization problem, and a feasible solution S of I, we denote $m(I,S)$ the value of solution S, $opt(I)$ the value of an optimal solution of I and $W(I)$ the value of a worst solution of I. The *differential approximation ratio* of S is defined by $DR(I,S)=abs((m(I,S)-W(S))/(opt(I)-W(I))$. The traditional approximation ratio for a minimization problem is simply defined by $m(I,S)/opt(I)$.

[9] 0-DAPX is the class of NP optimisation problems for which all polynomial approximation algorithms have a *differential approximation ratio* of 0.

In all algorithms, we consider a Boolean formula E_R constructed as explained in Section III using a rule set R composed of a set of C collection rules associated with classes (or benefits) that the user can (and wants) to claim, and where each atomic rule can be proven using her assertions. Atomic rules that cannot be proven are removed via *Benefits Selector* (see Figure 1, step □) before constructing R. The size of $Data_u$, i.e., the set of assertions related to the rule set, is noted D.

T_B is a truth assignment function to $Data_u$ that we implement as an array of Booleans with the semantics $B[i]=true \Leftrightarrow as_i$ *is exposed*. The rule set is represented as an array *R[]* of C collection rules, each collection rule *R[i]* being an array *atom[]* of atomic rules, each atomic rule *R[i].atom[j]* being an array *b[]* of references to the elements of B (see example in Table III). Note that E_R is *true* when each collection rule *R[i]* has at least one atomic rule where all referenced Boolean elements are *true*.

A. Exact Resolution (BIP model)

We propose to use a state of the art BIP solver, generally termed as *Mixed Integer Non-Linear Program* (MINLP) solver, to produce an exact result. We have chosen the popular and open source *COUENNE* solver [9] to this respect.

In order to use a MINLP solver, an instance of the problem must be written as a MINLP program. This is a direct transformation where each assertion corresponds to a Boolean variable, where the objective function is simply the sum of all the variables, and in which we express one *non-linear* constraint per collection rule r_j: $\Sigma_k \Pi_m as_{j,k,m} \geq 1$

The running example presented in Section II.C can be expressed by the following program, written in *AMPL* [14].

```
var b1 binary; ... var b10 binary;
minimize EX:
b1+b2+b3+b4+b5+b6+b7+b8+b9+b10;
subject to
r1: b1*b2 + b3*b4 >= 1;
r2: b5*b6*b7 + b4*b8*b9 >= 1;
r3: b1*b6*b7 + b2*b4*b10 >= 1;
r4: b2*b5*b6*b7 + b1*b4*b8*b9 >= 1;
```

The program is then fed to the BIP solver. As shown in Section V, the range of parameters for which the BIP solver computes the solution in an acceptable time (under 2h) is small.

B. Approximate Solutions (Polynomial Time)

We need to revert to a polynomial time approximation in order to compute results for the instances of the problem that cannot be tackled within reasonable time by the solver. We propose three algorithms: a naïve fully random algorithm called *RAND**, a simulated annealing meta-heuristics based algorithm called *SA**, and an algorithm called *HME* using a heuristic specially designed for the *ME* problem. These algorithms are non deterministic, therefore they can be run many times and the best solution is kept. However, they produce their first result in linear or polynomial time, depending on the algorithm. We discuss the complexity of the algorithms on a single run, to compare their speed. To compare their quality, we run the longest algorithm (*HME*) once, and we execute the other algorithms (*RAND** and *SA**) as many times as necessary, until they run out of processor time.

1) Fully random algorithm (RAND*)

The fully random algorithm *RAND** is based on a random choice of rules, and serves as a baseline. *RAND** randomly chooses one atomic rule for each collection rule and sets to *true* the value of each Boolean in B that this atomic rules refers to. Since each class is covered, the corresponding set of assertions determined by the truth assignment T_B is a solution to the *ME* problem instance. The result is the solution found within the allocated time limit for which **EX** is minimum (best result).

2) Simulated Annealing Algorithm (SA*)

Meta-heuristics are used in optimization problems in order to guide the algorithm towards better solutions, instead of simply randomly selecting them. We consider here simulated annealing [16] and introduce the *SA** algorithm to serve as a representative for meta-heuristic guided algorithms. Before each run, *RAND** is executed once to provide a starting solution, to feed *SA**. This solution is improved by *SA**, and if given enough processor time, *SA** can restart. The pseudo code of *SA** can be found in [3]. *Both RAND** and *SA** algorithm provide a solution in polynomial (linear) time of complexity.

3) The HME Algorithm

The Heuristic for Minimum Exposure (*HME*) algorithm that we propose uses a specific heuristic for the *ME* problem. The heuristic lies in the computation of *score[i]* the score of the i^{th} Boolean entry in B, using the function *fix*(B). This function computes a lower bound of the value of **EX**, by computing the number of predicates that can no longer be set to *false* for the given B. For instance, suppose that *B[i]=false* (i.e., as_i is not exposed). All the atomic rules referring to *B[i]* cannot be proven anymore. This leads to the fact that **EX** will be greater (or equal) to the value of the cardinality of the set of predicates in the atomic rules that are the only ones left to prove a given class. Using the running example (see Section II.C), we illustrate the computation of *fix(B)* in Table IV for each Boolean entry at each step of the algorithm. Let us briefly see how *score[1]* and *score[3]* are computed for the first step. If *B[1]=false*, then we have to prove collection rules *R[1]*, *R[3]*, *R[4]* using respectively atomic rules *R[1].atom[2]*, *R[3].atom[2]*, *R[4].atom[1]* (i.e., which means setting to *true* the *7* Booleans *B[2]*, *B[3]*, *B[4]*, *B[5]*, *B[6]*, *B[7]*, *B[10]*), leading to *score[1]=7*. If *B[3]=false*, this means proving *R[1]* using *R[1].atom[1]* (i.e., set to *true* the *2* Booleans *B[1]*, *B[2]*), therefore *score[3]=2*. We show in grey the lowest score, which means a truth assignment set to *false* in next steps, indicated by the symbol –. Assertions for which the score is denoted by ∞ are those for which the final truth assignment is set to true. The final result is here *B=[B[1]=true, B[2]=true, B[3]=false, B[4]=false, B[5]=true, B[6]=true, B[7]=true, B[8]=false, B[9]=false, B[10]=false]* which happens to be the minimal value of **EX** on this instance of the problem.

We see that the cost of *HME* algorithm is proportional to $O(\text{COST}_{\text{FIX}} \times D^2)$, where COST_{FIX} is the cost of computing the *fix* function. More precisely, in our implementation, $COST_{FIX} = O(|R| \times d_C \times d_{QD})$, where $|R|$ is the number of collection rules, d_C is the number of atomic rules per collection rule and d_{QD} is the number of predicates per atomic rule.

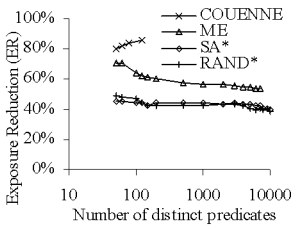


Figure 2. *ER* varying the number of distinct predicates.

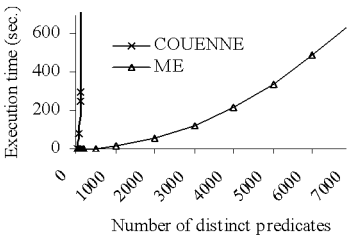


Figure 3. Execution time varying the number of distinct predicates.

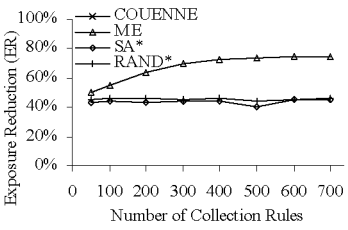


Figure 4. *ER* varying the number of collection rules.

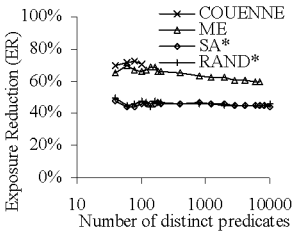


Figure 5. *ER* varying the number of predicates & collection rules

HME algorithm

```
Input:  E_R a Rule Set Boolean Formula
Ouput:  B a truth assignment of the assertions that
        proves R
1.   for i = 1 to D do
2.     B[i] ← true
3.   endfor
4.   while (exists i such that: B[i]=true and
           if B[i] is set to false then E_R (B) remains true)
5.     for i = 1 to D do
6.       score[i] ← ∞
7.     endfor
8.     forall i such that B[i] = true do
9.       B[i] ← false
10.      if E_R (B)=true then // E_R (B) is true iff B proves R
11.        score[i] ← fix(B)
12.      endif
13.      B[i] ← true
14.    endforall
15.    m ← i such that score[i] is minimum
16.    B[m] ← false
17.  endwhile
18.  return B
```

The intuition behind the heuristic is to successively get rid of the assertions which require keeping the least number of other assertions (such that all benefits are preserved) among the remaining ones. This heuristic is particularly relevant when the number of atoms per collection rule is small. Our performance evaluation confirms this scope. Note that if this number increases then *HME* tends towards *RAND**.

We show in Section V that the *HME* algorithm provides very good results in terms of quality of approximation, while maintaining reasonable computational complexity. We show in [3] that the *HME* algorithm can be extended to assertions with *attribute* θ *value* predicates, with $\theta \in \{<, \leq, =, \neq, \geq, >\}$.

TABLE IV. EXECUTION OF THE *HME* ALGORITHM

Steps	*B[1]*	*B[2]*	*B[3]*	*B[4]*	*B[5]*	*B[6]*	*B[7]*	*B[8]*	*B[9]*	*B[10]*
1: score[i]	7	7	**2**	5	4	6	6	4	4	3
2: score [i]	∞	∞	–	5	5	6	6	5	5	**4**
3: score [i]	∞	∞	–	**5**	7	∞	∞	5	5	–
4: score [i]	∞	∞	–	–	∞	∞	∞	**5**	5	–
5: score [i]	∞	∞	–	–	∞	∞	∞	–	**5**	–
Final B[i]	*true*	*true*	*false*	*false*	*true*	*true*	*true*	*false*	*false*	*false*

V. EXPERIMENTS

In this section, due to lack of space we only briefly present an experimental validation of our approach on synthetic data. Experiments were conducted on a HP workstation with 3.1GHz Intel CPU and 8GB RAM running Java1.6 (x64). The *COUENNE* solver was run on the same machine. Algorithms, data and BIP model generator code are available at http://project.inria.fr/minexp/

Results concern both scalability and quality. The quality is measured by computing the reduction of the set of exposed assertions in the application form, as follows:

Exposure Reduction: $\boldsymbol{ER}(T_B)=1-\mathbf{EX}(T_B)/|B|$,.

There are many parameters to the problem (number of collection rules, atomic rules per collection rule, number of distinct predicates/assertions, etc.), but many are linked. We fix the number of predicates per atomic rule (assuming that a user can prove all predicates through assertions, this is equivalent to fixing $Data_u$) and the number of atomic rules per collection rule to 4. This corresponds to the values of real decision trees extracted from neural networks used for credit scoring process [7]. To analyze exposure reduction and scalability, we generate rule sets varying the number of rules $|R|$ and the number of distinct predicates D in these rules (which is equivalent to the number of assertions related to the rule set). The number of atomic rules $|Q|$ varies accordingly since $|Q|=|R|\times 4$. We set a time limit of 2 hours for the exact solver and of 10 minutes for the approximation algorithms. The results are presented in Figures 2 to 5.

We draw three main conclusions from these experiments. First of all, the exposure reduction is important even with very simple algorithms (see RAND*), ranging from 30% to 80%, and is on average of 70% in the area of applicability of the exact solver. This means that on average only 30% of a user's data items is sent when using *ME* compared to the traditional case. Second, the scope of the exact solution is limited, and therefore the use of approximation algorithms is unavoidable. Third, *HME* provides the best results of the approximation algorithms, outperforming them by about 10%, and scales in polynomial time with regards to D.

VI. RELATED WORK

The transposition of legal privacy principles into privacy aware systems has fostered many studies. Emblematic examples include the P3P Platform for Privacy Preferences [11], privacy policy languages like EPAL [6] and Hippocratic databases [1]. P3P highlights conflicting policies, but it offers no means to calibrate the data exposed by a user and achieve *Limited Data Collection* (LDC). Many other policy languages have been proposed for different application scenarios, like EPAL [6], XACML [18] or WSPL [4], but to the best of our knowledge, no language has been introduced with *LDC* in mind. Another emblematic study deals with Hippocratic databases [1]. The architecture of a Hippocratic database is

based on ten guiding privacy principles including *LDC*. It addresses *LDC* by maintaining the set of attributes that are required for achieving each declared purpose. However, this solution assumes useful and useless data for a given purpose can be distinguished at the time of the data collection. As mentioned in the introduction, this assumption only holds for simple cases, but not in general decision making processes.

Existing works closer to our study can be found in the area of automated trust negotiation and credential based access control, where access decisions are based on the gradual confrontation of an access control policy with a set of credentials. A few number of works including [5, 10, 24] can be considered as following a *minimum exposure* approach. All those works minimize the privacy leak of a set of personal data items (credentials) while enabling a given decision to be made (the grant or deny access decision). However, the problem and solutions are different from ours for two founding reasons. First, the decision making processes that we consider are more complex than access control. The collection rules in *ME* can model sets of decision trees classifiers: several dimensions can be considered (e.g., lower credit rate, longer duration, lower cost of insurance, larger portion of 0% loan, etc.) each one potentially impacting the final offer made to the applicant. Second, in our context, the decision making process requires by nature a huge amounts of personal data (e.g., to obtain a loan offer customers are asked to fill in forms with hundreds to thousands fields), while in access control only a few credentials are considered (e.g., up to 35 in [5]). The results of these works can therefore not be used in our context, because they fall short on both expressivity and scalability requirement.

VII. CONCLUSION

In this article, we have introduced the *Minimum Exposure* approach and the related *ME* problem. We have shown how it can be expressed in the form of a Boolean minimum weighted satisfiability problem. We have studied the scope of applicability of general operational research solutions, using a state of the art MINLP solver. For cases where an exact resolution was not applicable, we have proposed several algorithms to compute an approximation of the solution. In all cases, we have shown that the exposure reduction that can be achieved compared with traditional implementations of limited data collection is around 50% in the average. These benefits are not only interesting for the user, whose privacy is less exposed, but also for the service providers who can limit their losses in the event of a data breach. Our hope is to open a new avenue for interesting applications of the minimum exposure principle introduced in this paper.

VIII. ACKNOWLEDGMENTS

This work was supported by the KISS grant ANR-11-INSE-0005, by the DIGITEO LeTeVoNe grant, and by the INRIA Collaborative Action on the Protection of Privacy in the Information Society (CAPPRIS) grant.

REFERENCES

[1] Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. Hippocratic databases. In *Proceedings of VLDB*, 2002.

[2] Allard, T., Anciaux, N., Bouganim, L., Guo, Y., Le Folgoc, L., Nguyen, B., Pucheral, P., Ray, I., Ray, I., and Yin, S. Secure Personal Data Servers: a Vision Paper. In *VLDB Endowment*, *3*(1), 2010.

[3] Anciaux, N., Nguyen, B., and Vazirgiannis, M. Minimum Exposure in classification scenarios. INRIA Research Report, 2012. Available at http://www-smis.inria.fr/~anciaux/MinExp/

[4] Anderson, A.H. An Introduction to the Web Services Policy Language (WSPL). In *Proceedings of the POLICY Workshop*, 2004.

[5] Ardagna, C.A., De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., and Samarati, P. Minimising Disclosure of Client Information in Credential-Based Interactions. *Int. Journal of Information Privacy, Security and Integrity*, *1*(2/3), to appear in 2012.

[6] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. Enterprise privacy authorization language 1.2 (EPAL 1.2). W3C Member Submission, 2003.

[7] Baesens, B., Setiono, R., Mues, C. and Vanthienen, J. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, *49*(3), 2003.

[8] Bauer D, Blough D, and Cash D. Minimal information disclosure with efficiently verifiable credentials. *Digital Identity Management*, 2008.

[9] Belotti, P., Lee, J., Liberti, L., Margot, F., and Wachter, A. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software, 24*(4-5), 2009.

[10] Chen, W., Clarke, L., Kurose, J., and Towsley, D. Optimizing cost-sensitive trust-negotiation protocols. *IEEE Computer and Communications Societies* (INFOCOM), 2005.

[11] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002.

[12] Crook, J.N., Edelman, D.B., and Thomas, L.C. Recent developments in consumer credit risk assessment. *Euro. J. of Op. Research*, *183*(3), 2007.

[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] Fourer, R., Gay, D.M., and Kernighan, B.W. A Modeling Language for Mathematical Programming. *Management Science*, *36*, 1990.

[15] Huysmans, J., Baesens, B., and Vanthienen, J. Using rule extraction to improve the comprehensibility of predictive models. Open Access publications from Katholieke Universiteit Leuven, 2007.

[16] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. Optimization by Simulated Annealing. *Science*, *220*(4598), 1983.

[17] Mitchell, T. *Machine Learning*. McGraw-Hill, 1997.

[18] Moses, T. Extensible access control markup language (xacml) version 2.0. Oasis Standard, 2005.

[19] OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 23rd Sept. 1980.

[20] Ponemon Institute, LLC. 2010 Annual Study: U.S. Cost of a Data Breach. 2011.

[21] Samarati, P. Protecting respondents' identities in microdata release. *IEEE TKDE*, *13*(6), 2001.

[22] Sweeney, L. k-Anonymity: a model for protecting privacy. *Int. Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*, 2002.

[23] Xiao, X., and Tao, Y. Personalized privacy preservation. In *Proceedings of ACM SIGMOD*, 2006.

[24] Yao, D., Frikken, K.B., Atallah, M.J., and Tamassia, R. Private information: To reveal or not to reveal. In *ACM TISSEC*, *12*(1), 2008.

Annexe F.

Curriculum Vitae – Nicolas Anciaux

Nicolas ANCIAUX

Chargé de Recherche, Inria Paris-Rocquencourt, Equipe-Projet SMIS.

Né le 30 mars 1975 à Antibes
Nationalité française
Marié, 2 enfants

Inria Paris-Rocquencourt, Domaine de Voluceau
78153 Le Chesnay Cedex
tél.: 01.39.63.56.35 - fax : 01.39.63.55.96
email: Nicolas.Anciaux@inria.fr
http://www-smis.inria.fr/~anciaux/

1 CURSUS UNIVERSITAIRE

2001-2004 : Thèse de doctorat de l'Université de Versailles/St-Quentin (UVSQ).
Intitulée *'Systèmes de gestion de base de données embarqués dans une puce électronique'*. Cette thèse, financée par une bourse MENRT, a été conduite dans l'équipe *Mobilité et Confidentialité des Données* du laboratoire CNRS PRiSM, fut soutenue le 17 déc. 2004 et obtenue avec la mention 'Très honorable avec les félicitations du jury'.

2000-2001 : D.E.A. Méthodes Informatiques des Systèmes Industriels de l'UVSQ.
Obtenu avec mention 'Bien'. Stage: *'PicoDBMS, a DBMS for the smartcard'*.

1996-1999 : Etudes d'Ingénieur. Institut Supérieur d'Electronique de Paris (ISEP).

1993-1996 : Math sup / Maths spé option M. Lycée Wallon, Valenciennes.

1993 : Baccalauréat, série C. Obtenu avec mention 'Bien'.

2 CURSUS PROFESSIONNEL

Depuis 04/06 : Chargé de Recherche – Equipe-Projet SMIS, Inria Paris-Rocquencourt.
http://www-smis.inria.fr/
Recruté comme 2ème classe, devenu 1ère classe le 01/04/2008, titulaire de la prime d'excellence scientifique depuis 2010.

03/05 – 04/06 : Chercheur Post-Doctorant – Database group, University of Twente, Pays-Bas.
http://db.cs.utwente.nl/

09/03 - 03/05 : ATER – Université de Versailles/St-Quentin.

09/00 – 09/03 : Doctorant (bourse MENRT) – Université de Versailles/St-Quentin.
Moniteur – Université de Versailles/St-Quentin.

09/98 - 03/99 : Stagiaire – Server Technologies Division, Oracle Corp., Redwood Shores, USA.
J'ai conçu et implanté une infrastructure permettant de tester les optimisations implantées dans le moteur d'exécution parallèle d'Oracle 8i, sous la direction de Bruce Chang. Il s'agissait de programmer des procédures Oracle PL/SQL de test et d'introduire dans le noyau d'Oracle 8i écrit en Langage C les procédures de trace nécessaires à la vérification.

3 ACTIVITES DE RECHERCHE

3.1 Thème de recherche

Motivation des travaux de recherche. Les individus, pour pouvoir bénéficier de leurs propres données personnelles, passent par des applications en ligne qui rendent ces données exploitables et accessibles. Le respect de la vie privée des individus est alors délégué à l'application. Les données sont parfois exploitées à des fins secondaires, des passe-droits peuvent être accordés et la centralisation des données conduit à des attaques répétées impactant de très grands volumes de données. L'utilisateur est ainsi dépossédé de tout moyen de contrôle, ne peut ni éviter, ni même souvent connaitre, les usages indésirables faits de ses propres données. Cette situation est contestable. Un consensus économique, politique et social émerge actuellement, pour parvenir à un modèle plus respectueux de la vie privée. Mais il n'y a pas encore de

solution technique satisfaisante. Les deux approches actuelles consistent à améliorer la confiance que l'on peut mettre dans les serveurs, ou à introduire des serveurs personnels en charge de la gestion des données de leur propriétaire. La première ne permet pas de résoudre les problèmes intrinsèques aux solutions centralisées (attaques sophistiquées, modèle basé sur la délégation) et la seconde sacrifie souvent les fonctionnalités et usages innovants (le partage) sans garantir une très grande sécurité.

Nous introduisons une nouvelle voie, que nous appelons le « Web Personnel Sécurisé », où les individus régulent leurs données personnelles depuis des composants personnels et sécurisés. Les fonctionnalités principales des solutions centralisées peuvent être préservées (durabilité, disponibilité, partage des données) et l'exploitation des données se fait avec l'assentiment du propriétaire qui dispose de fortes garanties de non contournement de ses directives.

Travaux de recherche. Mes travaux de recherche s'intègrent dans les deux axes de recherche sous-jacents au modèle du Web personnel sécurisé. Il s'agit d'une part (*Axe 1*), d'embarquer des techniques de gestion de données dans les composants personnels sécurisés pour en faire de véritables serveurs personnels de données, et d'autre part (*Axe 2*), de définir et mettre en œuvre de nouveaux modèles de gestion de données respectueux de la vie privée, régulant le partage, la collecte, la rétention, et l'usage des données personnelles, et d'intégrer ces modèles dans une architecture suivant une approche *Privacy-by-Design* offrant de fortes garanties de non contournement à l'utilisateur.

Axe 1. Concernant la gestion de données embarquées, nous nous concentrons sur des composants sécurisés matériellement, notamment sur des dispositifs dotés de microcontrôleurs sécurisés et disposant d'une mémoire Flash (grande capacité de stockage). Le problème est de concevoir des algorithmes de gestion de données et des structures accélératrices, de façon adaptée aux fortes contraintes du composant: très faible quantité de RAM; caractéristiques techniques des mémoires Flash induisant des performances particulières d'accès en lecture/écriture. Mes contributions sur le sujet portent sur la conception de techniques de gestion de bases de données embarquées et les performances du système dans une logique de co-conception [RI9, RI6, RI5, RI3, CI10, CI5, CI4, CI2].

Axe 2. Concernant la définition de procédés de gestion de données respectueux de la vie privée, certains de mes travaux ont porté sur la gestion de données dans une base de données intégrant des limites de rétention suivant les objectifs des traitements [CI9, CI8, CI6, CI3] et ont été conduits dans le cadre d'une coopération avec l'Université de Twente (Pays-Bas). Les contributions les plus récentes se concentrent sur la définition d'architectures Privacy-by-Design reposant sur des puces sécurisées [RI10, CI18, CI17, CI16, CI14, CI11], sur des procédés d'exposition minimum lors d'interactions avec des processus externes de prise de décision personnalisée [RI8, CI15, CI13, CI12] et sur de nouveaux modèles permettant aux individus de réguler le partage de leurs données [CL2, RI7, RN2].

Ces deux axes de recherche sont très complémentaires : la gestion ubiquitaire de données personnelles introduit de nouveaux problèmes de préservation de la vie privée, et la gestion de données embarquées dans des composants sécurisés permet d'envisager de nouveaux modèles de sécurisation de bases de données. Outre les problèmes techniques, ce thème de recherche touche de nombreux enjeux économiques, juridiques, ou sociologiques. Nous ne prétendons pas étudier ces enjeux, mais nous essayons de confronter nos solutions techniques à ces enjeux. 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. Ainsi, le prototypage, l'expérimentation, et les discussions multi disciplinaires font partie intégrante de mon activité de recherche.

3.2 Encadrement de doctorants

Co-encadrement de la thèse de Saliha Lallali (depuis oct. 2012).

Part encadrement : 40% (avec Iulian Sandu Popa et Philippe Pucheral)

Sujet: L'objectif de la thèse de Saliha est de concevoir un module de contrôle d'accès embarqué dans un composant personnel et sécurisé, permettant d'évaluer des vues sur des ensembles de documents, fondées sur le paradigme NoSQL. Nous développons pour l'embarqué, une nouvelle classe de techniques d'indexation de documents basées sur des index inversés. L'index doit être capable de journaliser les mises à jour dans des structures écrites séquentiellement, afin d'éviter les modifications en place trop coûteuses en mémoire Flash. Le stockage dans ces journaux conduisant inévitablement à de mauvaises performances lors des recherches, l'index doit être périodiquement réorganisé de manière à fusionner les journaux avec l'index inversé. En parallèle, nous cherchons à caractériser les différents

modèles de contrôle d'accès que nous pourrions couvrir avec ces techniques d'interrogation, notamment parmi les modèles associant des tags aux documents.

Résultats: Nous avons conçu et implémenté un nouveau modèle d'indexation pouvant être utilisé pour implanter un index inversé et réaliser des recherches top-k, capable d'indexer de l'ordre du million de document en embarqué. Un article présentant cette technologie est en cours de préparation.

Pilotage du développement de PlugDB-engine (depuis sept. 2007).

Sujet: PlugDB-engine est un SGBD relationnel embarqué dans un nouveau type de dispositif personnel et sécurisé. Le logiciel recouvre le moteur du SGBD et des modules complémentaires permettant de pré-compiler sur PC les requêtes SQL des applications, de communiquer avec le SGBD à partir d'un programme Java via JDBC, et de synchroniser les données locales avec un serveur central. Je joue le rôle d'architecte et de coordinateur de ces développements, qui impliquent les doctorants et ingénieurs successifs de l'équipe: Yanli Guo, Lionel Le Folgoc, Mehdi Benzine, Tristan Allard, Shaoyi Yin, et actuellement Saliha Lallali (doctorants) et Christophe Salperwick, Kevin Jacquemin, Maggy El-Kholy, Alexei Troussov et actuellement Quentin Lefebvre et Aydogan Ersoz (ingénieurs dans l'équipe).

Résultats: PlugDB-engine a été présenté dans une douzaine d'événements nationaux and internationaux majeurs. Cette action rassemble l'équipe autour d'un élément fédérateur, génère des échanges techniques avec Gemalto et la société ZED Electronics (assembleur de matériel électronique) qui nous sont indispensables, et permet d'offrir une visibilité à l'équipe, notamment de par sa valorisation dans le projet CG78/DMSP.

Co-encadrement de la thèse de Lionel Le Folgoc (2009 - 2012).

Part encadrement : 30% (avec Luc Bouganim)

Sujet: Le sujet de Lionel s'inscrit à la fois dans l'étude du serveur de données personnel et sécurisé, et dans les projet CG78/DMSP. Plus précisément, il s'agit de concevoir des mécanismes de bufferisation efficaces en mémoire FLASH, d'assurer l'atomicité transactionnelle des mises à jour, la durabilité des données (restauration en cas de panne ou de vol du composant personnel, sans perte de sécurité), et leur disponibilité (auprès des autres usagers ou de tiers extérieurs, tout en respectant le contrôle d'accès, d'usage et les procédures d'audit).

Résultats: Des résultats ont été obtenus sur la bufferisation en FLASH et l'atomicité transactionnelle. Ces résultats ont été intégrés aux publications [RI9, CI11].

Co-encadrement de la thèse de Yanli Guo (2008 - 2011).

Part encadrement : 20% (avec Luc Bouganim et Philippe Pucheral)

Sujet: Le sujet de Yanli s'inscrit lui aussi dans le cadre du serveur de données personnel et sécurisé. Il adresse le problème de la protection cryptographique (chiffrement, hachage cryptographique, etc.) de la mémoire FLASH, qui n'est pas protégée matériellement. La difficulté est de combiner les techniques cryptographiques avec les techniques d'exécution de requêtes de manière efficace, sécurisée, et adaptée aux contraintes de la puce.

Résultats: Des résultats préliminaires ont été obtenus sur la protection cryptographique de structures triées (indexes hiérarchiques) et séquentielles en FLASH, très adaptées à l'exécution de requête (minimisant les opérations cryptographiques). Ces résultats ont été intégrés aux publications [RI9, CI11, CI10].

Co-encadrement de la thèse de Harold van Heerde (2006 - 2010).

Part encadrement : 50% (avec Maarten Fokkinga, Peter Apers et Philippe Pucheral)

Sujet: Le sujet de la thèse d'Harold porte sur la conception et l'évaluation d'un modèle de conservation limitée des données personnelles. Le modèle est basé sur la dégradation progressive est irréversible des données, faisant l'hypothèse que les données anciennes sont moins utiles et peuvent être oubliées sans nuire à l'usage qui en est fait. Des procédures systèmes (réécriture, chiffrement et effacement de la clé) permettent de garantir que les données ont physiquement disparu. Le travail de thèse était supervisé au départ avec Peter Apers, professeur à l'Université de Twente, directeur de l'équipe base de données (Database Group) et du CTIT (Center for Telematics and Information Technology). Lors de mon départ de l'Université de Twente pour Inria, la thèse a pris la forme d'une cotutelle, co-encadrée par Maarten Fokkinga (université de Twente) et moi-même, et supervisée par Peter Apers et Philippe Pucheral (co-directeurs de thèse).

Résultats: Les résultats de la thèse sont à la base des publications [CI9, CI8, CI6, CI3]. L'approche défendue a eu une visibilité auprès du grand public et a fait l'objet d'un article sur BBC News [D8].

3.3 Implication dans des projets

Projet CG78/DMSP (Département des Yvelines, depuis 2007)

Coordinateurs: Philippe Pucheral et Nicolas Anciaux (SMIS).

Mon rôle: coordination du projet, responsable technique du projet.

Partenaires: Inria, Université de Versailles, Santeos (Atos Origin), Gemalto, ALDS (coordination gérontologique médicale), et COGITEY (coordination sociale).

Objectif: Le projet a pour objectif de concevoir un dossier médico-social mobile et sécurisé facilitant les soins au domicile de personnes dépendantes, et d'expérimenter la solution sur le territoire des Yvelines. Le projet a déjà donné lieu à 3 conventions : 2007-2010 (élaboration de la technologie), 2011-2012 (expérimentation terrain) et 2013-2014 (évolution de la technologie). Au niveau technique, le projet implique la conception et la mise en œuvre d'un serveur personnel de données sur un composant matériel combinant un microcontrôleur sécurisé (type carte à puce) et une grande quantité de la mémoire FLASH dans un format carte SIM, ainsi que la conception et le développement des services attenants de synchronisation et de restauration du serveur embarqué. L'expérimentation terrain s'est déroulée sur 18 mois en 2011/2012, auprès d'une centaine de patients et professionnels médicaux sociaux sur le territoire des Yvelines. Un retour d'expérience a conduit à des adaptations importantes du composant personnel sécurisé (matérielles et logicielles), nous amenant à faire fabriquer nous-même un nouveau composant doté d'une interface Bluetooth et un lecteur d'empreinte digitale. L'ARS Ile de France réalise actuellement un audit de la solution développée dans le projet afin d'envisager un déploiement plus large (résultat de l'audit prévu pour fin 2014). Une vidéo décrit la solution et une démonstration est disponible. La version actuelle du composant personnel s'interface avec n'imposte quel Smartphone ou tablette Android équipé d'un port USB ou du Bluetooth.

Projet CityLab@Inria (Inria Project Lab, depuis juin 2014)

https://citylab.inria.fr/

Coordinateur: Valérie Issarny (Inria@Sillicon Valley & Arles-Mimove).

Mon rôle : Responsable pour le partenaire SMIS.

Partenaires: Arles-Mimove, Clime, Dice, Fun, Myriads, OAK, SMIS, Urbanet et Willow.

Objectif: Le projet étudie les solutions ICT pour la ville intelligente dans un objectif de soutenabilité sociale (et environnementale). J'ai eu un rôle important dans la rédaction de la proposition de projet, et y représente l'équipe SMIS. Notre implication a pour but d'envisager des architectures Privacy-by-Design garantissant la vie privée des citoyens, dans un contexte où ils sont producteurs de données [ABB+14]. Nous nous intéressons en particulier la capture de données sociales, produites par les usagers depuis leur smartphone, dans un environnement dit de "social sensing".

ISN (Idex Paris Saclay, depuis dec. 2013)

http://digitalsocietyinstitute.com/

Coordinateurs du pôle: Fabrice Le Guel (ADIS) et Benjamin Nguyen (SMIS).

Mon rôle : Membre du pôle Vie privée et identité numérique et responsable pour SMIS du projet PEPS PAIP.

Partenaires: GRACE/LIX, COMETE/LIX, DANTE, CERDI, SAMOVAR, SMIS, RITM.

Objectif: L'Institut de la Société Numérique (ISN) adopte une approche interdisciplinaire, entre disciplines informatiques et sciences humaines économiques et sociales, pour étudier certains défis sociétaux inhérents à la société numérique. Deux pôles ont été lancés: le premier sur le thème de la co-évolution homme/machine, le second sur celui de la vie privée et l'identité numérique. Je m'implique actuellement dans le pôle 'vie privée et identité numérique'. Nous avons notamment lancé le projet PEPS PAIP financé par le CNRS et impliquant les partenaires du pôle, dans lequel nous évaluons sous forme expérimentale, l'impact sur les usagers de solutions de gestion de données personnelles où l'individu possède (physiquement) ses données ainsi que le serveur qui en régit la dissémination, par rapport aux solutions centralisées classiques.

Projet KISS (ANR INS, Dec. 2011 – Dec. 2015 (à vérifier))

https://project.inria.fr/kiss/

Coordinateur: Philippe Pucheral (SMIS).

Mon rôle : Responsable de la tâche sur l'exposition minimum de données.

Partenaires: Conseil Général 78, CryptoExpert, Gemalto, SMIS, LIRIS, SECRET, UVSQ.

Objectif: Le projet vise à produire une alternative crédible à la centralisation systématique des données personnelles sur des serveurs tiers, et ouvrir la voie à de nouvelles solutions suivant une approche Privacy-by-Design pour la gestion des données personnelles. L'idée soutenue dans KISS est d'embarquer dans des composants personnels de confiance, des modules logiciels capables d'acquérir, de stocker et de gérer différentes formes d'information personnelle (ex. bulletins de salaires, factures, données bancaires, médicales, traces de géolocalisation) selon les applications, et d'en réguler la dissémination [APP+12]. Ces modules logiciels forment un serveur de données personnel, capable de s'interfacer avec des services externes, tout en restant sous le contrôle de son propriétaire. Je suis responsable dans ce projet des travaux cherchant à limiter au minimum les données à exposer à des services externes. Nous avons proposé des procédés et algorithmes adaptés à certains scénarios applicatifs validés avec le Conseil Général des Yvelines dans le cadre de la demande d'aide sociale.

Projet DEMOTIS (ANR-ARPEGE, 2009 – 2012)

http://www.demotis.org/

Coordinateur: Philippe Aigrain (Sopinspace).

Mon rôle : Responsable scientifique pour les équipes Inria.

Partenaires : CECOJI, Inria (équipes CACAO, SECRET, SMIS), Sopinspace.

Objectif : Le projet DEMOTIS (Définir, Évaluer et MOdéliser les Technologies de l'Information de Santé) vise à éclairer les limitations et compromis réciproques que l'intrication des domaines juridiques et informatiques impose à la conception d'infrastructures en charge du Dossier Médical Personnalisé (DMP) et celles des dossiers des réseaux de soins liés à certaines affections (SIDA, cancer). Les deux volets du projet, juridique (droit de la santé, des données personnelles ou de la propriété intellectuelle, par exemple) et informatique (sécurité des bases de données et techniques cryptographiques utilisées pour les protéger) ont été abordés de manière conjointe par les partenaires.

Projet PlugDB (ANR RNTL, 2007 – 2010)

Coordinateur: Philippe Pucheral (SMIS).

Mon rôle : Responsable de la coordination technique.

Partenaires : Inria (SMIS), Univ. de Versailles, Santeos (Atos Origin), Gemalto, ALDS (coordination gérontologique médicale).

Objectif : Conception d'un serveur personnel de données sur un nouveau composant matériel combinant un microcontrôleur sécurisé (type carte à puce) et une grande quantité de la mémoire FLASH (Go) dans une clé USB. La solution doit offrir une alternative à la centralisation des données plus respectueuse de la vie privée, tout en restaurant les propriétés classiques d'un serveur central.

Projet CADMAI (NWO VIDI, 2005 – 2010)

Grant individuel attribué au Prof. Ling Feng (Université de Twente, Pays-Bas)

Mon rôle : Responsable de la tâche relative à la protection de la vie privée.

Objectif : Le projet CADMAI (Context-Aware Data Management Towards Ambient Intelligence) adresse le problème de la conception et la mise en œuvre de techniques de gestion de données dans un contexte d'intelligence ambiante. Le Prof. Ling Feng a formé une équipe de cinq doctorants. C'est dans le cadre de ce projet que j'ai réalisé mon post-doctorat à l'Université de Twente. Je me suis principalement intéressé à l'intimité que peuvent avoir les individus dans un tel environnement, et ai initié l'étude sur la dégradation progressive des données personnelles dans ce contexte. J'ai encadré en 2005/2006 le stage de Master d'Harold van Heerde sur la thématique, qui a ensuite poursuivi en thèse.

3.4 Animation Scientifique

Membre de comités d'organisation

2013: Co-organisation de la 4ème édition de l'Atelier Protection de la Vie Privée (APVP'13)
2012: Président des démonstrations pour la conférence "Bases de Données Avancées" (BDA'12)

Membre de comités de sélection et d'expertise

2013: Université Paris Dauphine, poste de Maître de Conférence n°0230.
2012: Expert pour l'ANR, Programme Blanc JCJC SIMI 2 - Science informatique et applications.
2010: Université Paris Sud & Chaire INRIA/UPS, poste de Maître de Conférence n°27MCF0653.

Membre de la Commission du Développent Technologique (CDT), Inria Paris-Roc. (depuis 2013)

La CDT est en charge de (1) sélectionner les équipes Inria bénéficiant de supports ingénieurs, valider les candidats identifiés par l'équipe, évaluer le renouvellement des supports, et (2) affecter les ingénieurs du SED aux équipe Inria nécessitant un support spécifique.

Membre de comités de programme (CP) et comités éditoriaux (CE)

2014: Conf. Nat. Bases de Données Avancées BDA'14 (CP demo)
2013: Conf. Nat. Bases de Données Avancées BDA'13 (CP & session chair)
2012: Conf. Nat. Bases de Données Avancées BDA'12 (CP demo),
Techniques et Sciences Informatiques TSI (CE)
2011: Int. Conf. on Data Engineering ICDE'11 (CP),
Techniques et Sciences Informatiques TSI (CE)
2010: Int. Conf. on Data Engineering ICDE'10 (CP demo),
Techniques et Sciences Informatiques TSI (CE)
2009: Int. Conf. on Management of Data SIGMOD'09 (CP demo),
Techniques et Sciences Informatiques TSI (CE)
2008 Int. Conf. on Management of Data SIGMOD'09 (CP),
Int. Conf. on Data Engineering ICDE'08 (CP),
Int. Conf. on Extending Database Technology EDBT'08 (CP demo),
Techniques et Sciences Informatiques TSI (CE)

3.5 Développement de Prototypes et Applications.

PlugDB-Engine. Système de Gestion de Bases de Données (SGBD) relationnel embarqué dans un composant personnel dotée d'un microcontrôleur sécurisé (type carte à puce) relié à une mémoire FLASH de grande capacité. Le moteur du SGBD s'exécute dans le microcontrôleur et les données sont stockées (sous protection cryptographique) en mémoire FLASH. Des modules complémentaires permettent de pré-compiler des requêtes SQL pour les applications, de communiquer avec le SGBD à partir d'un programme Java/Android, et de synchroniser les données locales avec un serveur central. Ce prototype est développé dans le cadre du projet CG78/DMSP. Il a fait l'objet de dépôts successifs à l'Agence pour la Protection de Programmes (APP) depuis 2008, a été démontré dans une douzaine d'événements nationaux et internationaux dont Javaone'09 [D4] (événement industriel important regroupant près de 15000 participants), SIGMOD'10 [CI10], Futur en Seine 2013, et a fait l'objet d'une audition publique à l'Assemblée Nationale en 2009 dans le cadre de la relance du dossier médical personnel. Le prototype a été expérimenté sur le terrain dans les Yvelines. L'ARS Ile de France réalise actuellement un audit de l'application en vue d'une expérimentation plus large.

Mon rôle : Conception, pilotage des développements, coordination technique avec les partenaires.

MinExp-Card. Des formulaires de demande sont souvent utilisés pour collecter des données personnelles sur les postulants, pour ensuite pouvoir ajuster les services offerts (e.g., prêt bancaires, aide sociale) à leur situation spécifique. L'ensemble de ces données doit être réduit à son strict minimum en vue du traitement ultérieur, comme l'impose le principe de collecte limitée inscrit dans les textes internationaux sur la vie privée, mais aussi pour réduire les coûts (traitement des dossiers, archivage, fuites) pour les fournisseurs de services. Nous avons conçu un prototype sur carte à puce basé sur nos travaux de recherche [RI8, CI13, CI12], visant les services d'aide sociale fournis par

les Conseils Généraux. Le prototype a notamment été démontré à EDBT'13 [CI15] et a fait l'objet d'une présentation à l'IREP [D15].

Mon rôle : Conception et développement du prototype.

InstantDB. Système de gestion de données implantant des mécanismes d'oubli progressif. Notre activité quotidienne laisse des traces digitales dans un nombre croissant de bases de données (sites Web commerciaux, fournisseurs de services Internet, moteurs de recherche, etc.). Ces traces font souvent l'objet de divulgations accidentelles, de piratage, ou d'interrogations abusives. Personne n'est à l'abri car une situation singulière (comme la recherche d'un emploi ou une demande de crédit) peut rendre un historique, a priori quelconque, soudainement intéressant. InstantDB permet d'explorer les potentialités d'un système de gestion de traces digitales incorporant des mécanismes d'oubli progressif. Le modèle de dégradation considéré comme point de départ est décrit dans [CI8, CI6].

Mon rôle : Conception et participation à la réalisation.

GhostDB. Prototype permettant d'interroger une base dont certaines colonnes sensibles sont embarquées dans une clé USB dotée d'un microcontrôleur sécurisé [CI5, CI4]. La première version du prototype [CI5] permet de traiter des requêtes simples (sélections, projections et jointures) et tourne sur un simulateur logiciel comptabilisant les accès à la FLASH. Une analyse approfondie, considérant les calculs d'agrégats, réalisée sur un composant réel fourni par Gemalto, a été publiée ensuite [RI6].

Mon rôle : Conception et participation à la réalisation.

4 ACTIVITES D'ENSEIGNEMENT

J'assure depuis plusieurs années des enseignements dans des écoles d'ingénieur (ENSTA, Telecom Paristech) et dans des filières de formation par l'apprentissage (CFA UVSQ/UPMC/AFTI). Ces cours, outre l'enseignement des problématiques générales des bases de données, me permettent de sensibiliser les étudiants aux problèmes et procédés liés à la gestion de données respectueuse de la vie privée. De plus, je recrute chaque année un étudiant de l'ENSTA en stage sur ces thématiques. L'un d'entre eux s'apprête à poursuivre en thèse dans l'équipe.

Nous avons en outre monté cette année à l'ENSIIE la filière "Systèmes d'information Privacy-by-Design", avec un module de cours et un module de projet. Le module de projet permet aux étudiants de proposer et de valider de stratégies permettant de garantir certains principes inhérent au respect de la vie privée, sur la base de composants matériels personnels et sécurisés que nous faisons fabriquer. Nous envisageons à terme de proposer du matériel et des briques logicielles aux écoles sous la forme d'open hardware / open software.

4.1 Bases de données

Mots clefs: Objectifs des SGBD, modèle et algèbre relationnels, langage SQL, architecture des SGBD, programmation SQL, contraintes d'intégrités, stockage et indexation, évaluation de requête, transactions, sécurité.

2015:	Bases de données avancées, module C1.3, 3ème année cycle ingénieur	ENSTA	21h/an
2015:	Bases de données, module IN206, 2ème année cycle ingénieur	ENSTA	21h/an
2007 – 2014:	Stage de bases de données, module IN410, Mastère ASI	ENSTA	21h/an
2009 – 2013:	Bases de données, module IN206, 2ème année cycle ingénieur	ENSTA	21h/an
2009 – 2014:	Bases de données, filière IRS	CFA UVSQ/AFTI	30h/an
2006 – 2010:	Master Analyse des Systèmes Stratégiques	UVSQ	36h/an
2006 – 2008:	Bases de données avancées, 3ème année cycle ingénieur	ISTY	36h/an

4.2 Sécurité des Bases de données

Mots clefs: Architecture des systèmes orientés données, sécurité des SGBD, contrôles d'accès, protection cryptographique de bases de données, anonymisation, tolérance aux pannes, gestion de données et vie privée.

2013 – 2014:	Systèmes d'information Privacy-by-Design, modules SIP1&2	ENSIIE	15h/an
2013 – 2014:	Interventions en sécurité des BD, module INF345	Telecom Paristech	9h/an
2011 – 2014:	Sécurité des bases de données, filière MSI	CFA UPMC/AFTI	30h/an

5 SELECTION DE PUBLICATIONS

5.1 Chapitres de Livre

[CL2] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Trustworthiness of Pervasive Healthcare Folders. Book chapter of Pervasive and Smart Technologies for Healthcare: Ubiquitous Methodologies and Tools, A. Coronato, G. De Pietro (editors), Information Science Reference, pp. 1-24, 2010.

[CL1] N. Anciaux, L. Bouganim, P. Pucheral. A Hardware Approach for Trusted Access and Usage Control. Book chapter of the Handbook of Research on Secure Multimedia Distribution, S. Lian, Y. Zhang (editors), Information Science Reference, 2009.

5.2 Revues Internationales

[RI10] 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.

[RI9] 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.

[RI8] N. Anciaux, D. Boutara, B. Nguyen, M. Vazirgiannis. Limiting Data Exposure in Multi-Label Classification Processes. Fundamenta Informaticae, to appear.

[RI7] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Seamless Access to Healthcare Folders with Strong Privacy Guarantees. Special issue of the Journal of Healthcare Delivery Reform Initiatives, 2010.

[RI6] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha. Revelation on Demand. Distributed and Parallel Database Journal (DAPD), Vol. 25, n°1-2, pp. 5-28, 2009.

[RI5] N. Anciaux, L. Bouganim, P. Pucheral, P. Valduriez. DiSC: Benchmarking Secure Chip DBMS. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), Vol. 20, n° 10, pp. 1363-1377, 2008.

[RI4] N. Anciaux, M. Berthelot, L. Braconnier, L. Bouganim, M. De la Blache, G. Gardarin, P. Kesmarszky, S. Lartigue, J-F. Navarre, P. Pucheral, J-J. Vandewalle, K. Zeitouni. A Tamper-Resistant and Portable Healthcare Folder. International Journal of Telemedicine and Applications (IJTA), Vol. 2008, 2008.

[RI3] N. Anciaux, L. Bouganim, P. Pucheral, 'Future Trends in Secure Chip Data Management', IEEE Data Engineering Bulletin (IEEE DEB), Vol. 30, n°3, 2007.

[RI2] N. Anciaux, L. Bouganim, P. Pucheral. Data confidentiality: to which extent cryptography and secured hardware can help. Annals of telecom, Vol. 61, n°3-4, 2006.

5.3 Revues Nationales

[RN3] N. Anciaux, B. Nguyen, M. Vazirgiannis M. Exposition minimum de données pour des applications à base de classifieurs. Ingénierie des Systèmes d'Information, Vol. 18, n°4, pp. 59-85, 2013.

[RN2] T. Allard, N. Anciaux, L. Bouganim, P. Pucheral, R. Thion. Concilier ubiquité et sécurité des données médicales. Les Cahiers du CRID « Les technologies au service des droits, opportunités, défis, limites », D. Le Métayer (Editor) Bruylant, Vol. 32, Jan. 2010.

[RN1] N. Anciaux, L. Bouganim, P. Pucheral. SGBD embarqué dans une puce : retour d'expérience. Technique et Science Informatiques (TSI), Vol. 27, n°1-2, 2008.

5.4 Conférences Internationales

[CI18] N. Anciaux, L. Bouganim, T. Delot, S. Ilarri, L. Kloul, N. Mitton, P. Pucheral. Folk-IS: Opportunistic Data Services in Least Developed Countries. 36th International Conference on Very Large Data Bases (PVLDB), Vol. 7(5), Vision Paper, pp. 425-428, 2014.

[CI17] N. Anciaux, B. Nguyen, I. S. Popa. Tutorial: Managing Personal Data with Strong Privacy Guarantees. 17th International Conference on Extending Database Technology (EDBT), Tutorial, pp. 672-673, 2014.

[CI16] N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, P.Pucheral, I. S. Popa. Trusted Cells : A Sea Change for Personnal Data Services. 6th Conference on Innovative Database Research (CIDR), 2013.

[CI15] N. Anciaux, W. Bezza, B. Nguyen, M. Vazirgiannis. MinExp-Card : Limiting Data Collection Using a Smart Card. 16th International Conference on Extending Database Technology (EDBT), demo paper, pp. 753-756, 2013.

[CI14] N. Anciaux, B. Nguyen, I. S. Popa. Personal Data Management with Secure Hardware : The advantage of Keeping you Data at Hand. 14th International Conference on Mobile Data Management (MDM), Advanced Seminar, pp 1-2, 2013.

[CI13] N. Anciaux, D. Boutara, B. Nguyen, M; Vazirgiannis. Limiting Data Exposure in Multi-Label Classification Processes. In International Workshop on Privacy-AwaRe Intelligent Systems (PARIS2012), 2012.

[CI12] N. Anciaux, B. Nguyen, M. Vazirgiannis. Limiting Data Collection in Application Forms : A real-case Application of a Founding Privacy Principle. 10th Conference on Privacy, Security and Trust (PST), 8p., 2012.

[CI11] 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.

[CI10] N. Anciaux, L. Bouganim, Y. Guo, P. Pucheral, J.-J. Vandewalle, S. Yin. Pluggable Personal Data Servers. 29th ACM International Conference on Management of Data (ACM SIGMOD), demo. Paper, Indianapolis, Indiana, Jun. 2010.

[CI9] H. van Heerde, M. Fokkinga, N. Anciaux. A Framework to Balance Privacy and Data Usability Using Data Degradation. IEEE International Conference on Computational Science and Engineering (CSE), Los Alamitos, CA, USA, 2009.

[CI8] N. Anciaux, L. Bouganim, H. van Heerde, P. Pucheral, P. M. G. Apers. Data Degradation: Making Private Data Less Sensitive Over Time. 17th ACM International Conference on Information and Knowledge Management (ACM CIKM), short paper, Napa Valley, USA, to appear, Oct. 2008.

[CI7] N. Anciaux, M. Benzine, L. Bouganim, K. Jacquemin, P. Pucheral, S. Yin. Restoring the Patient Control over her Medical History. 21th IEEE Int. Symposium on Computer-Based Medical Systems (IEEE CBMS), Jyväskylä, Finland, June 2008.

[CI6] N. Anciaux, L. Bouganim, H. van Heerde, P. Pucheral, P. M. G. Apers. InstantDB: Enforcing Timely Degradation of Sensitive Data. 24th International Conference on Data Engineering (ICDE), short paper, Cancun, Mexico, Apr. 2008.

[CI5] C. Salperwyck, N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha. GhostDB: Hiding Data from Prying Eyes. 33th International Conference on Very Large Data Bases (VLDB), demo. paper, Vienna, Austria, Sept. 2007.

[CI4] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha: GhostDB: Querying Visible and Hidden Data without Leaks. 26th ACM International Conference on Management of Data (ACM SIGMOD), Beijing, China, June 2007.

[CI3] H.J.W. Van Heerde, N. Anciaux, L. Feng, P. Apers. Balancing Smartness and Privacy for the Ambient Intelligence. First European Conference on Smart Sensing and Context (EuroSSC), Lecture Notes in Computer Science 4272 springer 2006, Enschede, The Netherlands, Oct. 2006

[CI2] N. Anciaux, L. Bouganim, P. Pucheral: 'Memory Requirements for Query Execution in Highly Constrained Devices. 29th International Conference on Very Large Data Bases (VLDB), Berlin, September 2003.

[CI1] N. Anciaux, C. Bobineau, L. Bouganim, P. Pucheral, P. Valduriez, ''PicoDBMS: Validation and Experience. 27th International Conference on Very Large Data Bases (VLDB), demo. paper, Roma, September 2001.

5.5 Dissémination et vulgarisation

[D17] N. Anciaux, B. Nguyen. Limiter la collecte des données personnelles, un problème juridique NP-difficile. Tangente Hors-série n°52, Mathématiques & Informatique, 2014.

[D16] N. Anciaux, P. Bonnet, L. Bouganim, P. Pucheral. Trusted Cells: Ensuring Privacy to for the Citizens of Smart Cities. ERCIM News, Vol. 98, 2014.

[D15] N. Anciaux. Garantir la confidentialité des données personnelles. Futur en Seine 2014, Répondre aux défis des smart cities, 2014. (slides)

[D14] N. Anciaux. Une nouvelle approche de la protection de nos données. Interview, MyScienceWork news, par Abby Tabor, 2013. (article) (video)

[D13] N. Anciaux. Vers un modèle de gestion des données respectueux de la vie privée : application à la collecte limitée d'informations personnelles. Seminaire IREP "BIG DATA", 2014. (slides)

[D12] N. Anciaux. Gestion de données personelles respetueuse de la vie privée. Presentation et démonstration, Futur en Seine, Archipelle des projets, 2013. (slides)

[D11] N. Anciaux, B. Nguyen, M. Vazirgiannis. The Minimum Exposure Project: Limiting Data Collection in Online Forms. ERCIM News, Vol. 90, 2012.

[D10] N. Anciaux, J.M. Petit, P. Pucheral, K. Zeitouni. Personal Data Server: Keeping Sensitive Data under the Indiviual's Control. ERCIM News, Vol. 90, 2012.

[D9] N. Anciaux, B. Nguyen, M. Vazirgiannis. Minimum Exposure - A New Approach for Limited Data Collection. Invited talk, Digiteo workshop on Web Mining, 2011, Telecom ParisTech, organized by M. Vazirgiannis and P. Senellart. (slides)

[D8] N. Anciaux. Dossier Médico-Social Portable et Sécurisé. Présentation et démonstration, Les Industries du Numérique pour la Santé, RII, in conjunction with the Connectathon, 2010, Cité Mondiale, Bordeaux. (video)

[D7] Fading data could improve privacy, By Mark Ward Technology correspondent, BBC News. 16 June 2010. http://www.bbc.co.uk/news/10324209

[D6] Sécurité des bases de données. N. Anciaux, D. Gross-Amblard, P. Pucheral, R. Thion. Ecole de printemps « MASSES DE DONNEES DISTRIBUEES », Ecole de Physique de Houches, du 16 au 21 mai 2010.

[D5] Demonstration of electronic Health Records (eHR) on Java Card™ 3.0 Technology. Nicolas Anciaux (Inria) and Jean-Jacques Vandewalle (Gemalto). BOF-4576, CS Advanced Based Devices, JavaOne Conference, San Francisco, USA, Jun. 2009. (slides)

[D4] Demonstration of Electronic Health Records (EHR) on Java Card 3.0 Based Devices. Jean-Jacques Vandewalle, Research Engineer GEMALTO, Nicolas Anciaux, Researcher (Inria). Smart Event 10th Edition, World e-ID 2009, Sophia-Antipolis, sept. 2009.

[D3] Participation à la Table Ronde 'Les Yvelines, acteurs et partenaires de la Recherche et Développement' lors de la Convention d'Affaires 'Les RDV Carnot. au Palais des Congrès de Versailles. Animée par : Christian Beley, Sous-Directeur Pôle économique CG78, Frédérick Becquet, Chargé de mission R&D CG78. Participants : Pr. Luc Montagnier, PDG Nanectis Biotechnologies, Yan Haentjens, PDG Vectrawave, Jean-Pierre Arragon, Directeur Portfolio Management Continental Automotive, et Nicolas Anciaux, Chargé de recherche à Inria Paris-Rocquencourt. Mars 2008.

[D2] Participation à la Smart University. N. Anciaux, L. Bouganim. Data Management in Embedded Smart Devices. Tutorial donné à la Smart University, co-organisée avec la 7ème édition de la conférence internationale e-smart. Sept. 2006.

[D1] N. Anciaux, L. Bouganim, P. Pucheral. Database Components on Chip. ERCIM News, Vol. 54, July 2003.

5.6 Thèse

[Th] Thèse de doctorat de l'Univ. de Versailles/St-Quentin, *'Systèmes de gestion de base de données embarqués dans une puce électronique'*. Mention très honorable avec félicitations. Déc. 2004.

Rapporteurs	Patrick Valduriez,	Directeur de Recherche - Inria, Rocquencourt
	Michael J. Franklin,	Professeur - Berkeley, San Francisco
Examinateurs	Philippe Bonnet,	Professeur - DIKU, Copenhague
	Jean-Claude Marchetaux,	Ingénieur de Recherche - Gemalto, Meudon
Directeur	Philippe Pucheral,	Professeur - PRISM, Versailles
Co-encadrant	Luc Bouganim,	Directeur de Recherche - Inria, Rocquencourt