données et algorithmes
pour une ville intelligente et durable

Séminaire SystemX
18 octobre 2018

Protection de la vie privée : potentiel et paradoxe du Cloud Personnel

Philippe Pucheral
UVSQ / DAVID & INRIA Saclay, équipe PETRUS

2 mots sur les enjeux de la protection des données personnelles

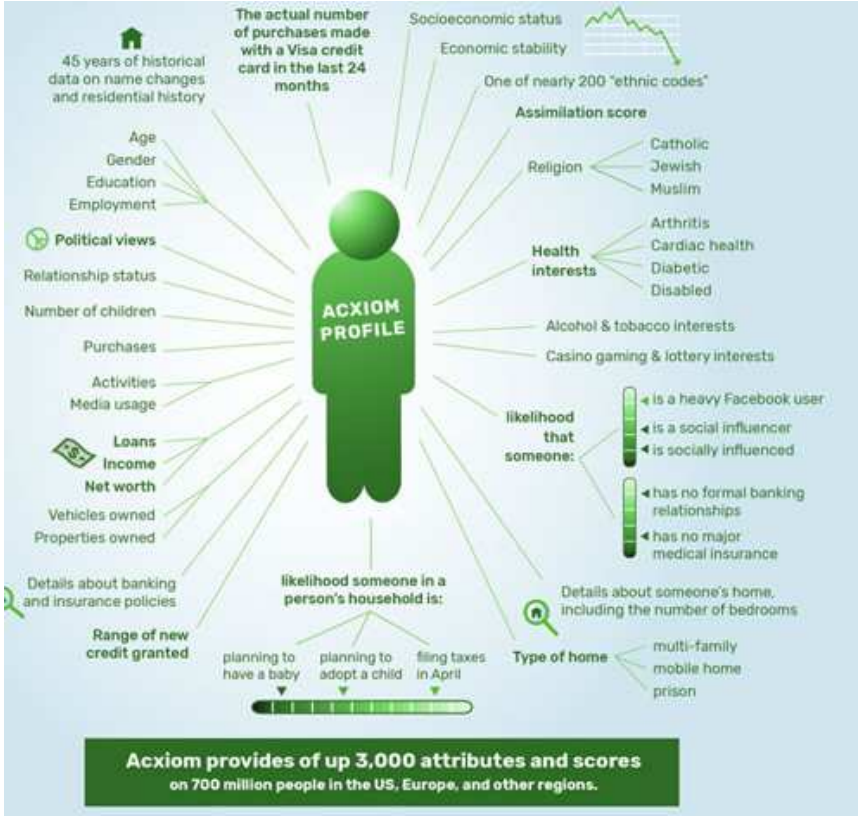
Entre enjeux économiques …

- Risque généralisé d'intermédiation

… et sociétaux

- Discrimination dans l'offre de services (ex: Allianz conduite connectée, YouDrive…)
- Uniformisation des comportements induits par le profilage (Google vs Qwant)
- Surveillance entre particuliers (ex : Intelius)
- Fuites massives : Yahoo, Equifax …

Exacerbé par la collecte silencieuse de données via les objets connectés



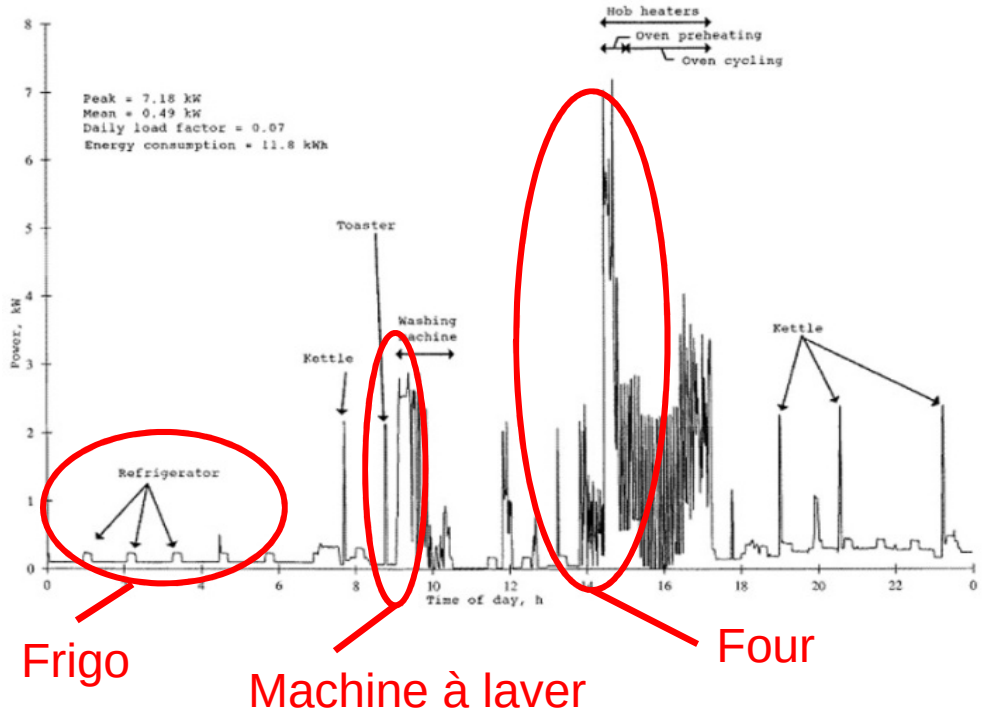
Données personnelles et IoT

80 Milliards d'objets connectés en 2020 (IDATE)

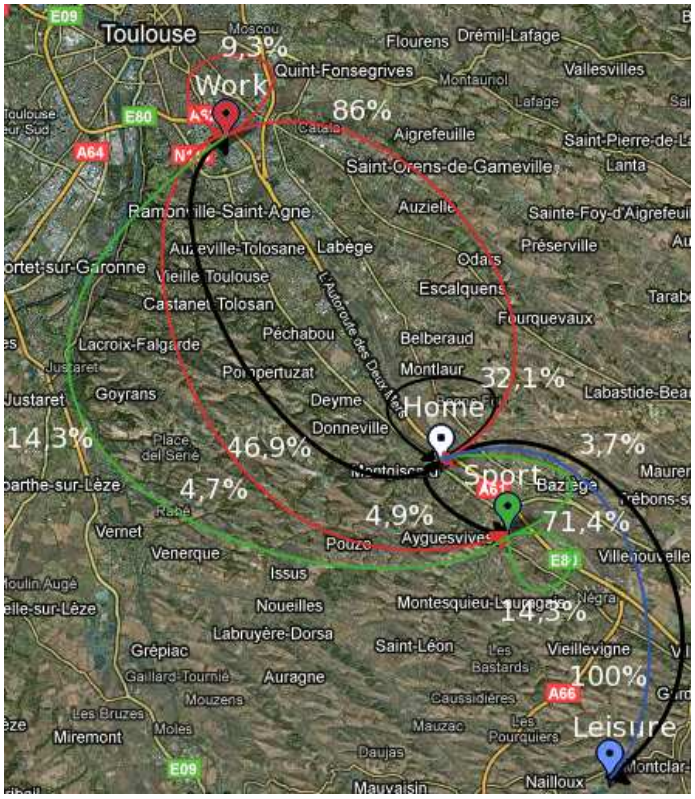
- Une manne pour l'analyste mais un risque patent pour la vie privée …

Doit-on réellement tout centraliser ?

Courbe de charge de smart meter



Trace de geolocalisation



Réglement Général Européen sur la Protection des Données (RGPD)

Quelques principes clé du RGPD

- Transparence et Consentement explicite et éclairé quant à la finalité de la collecte et du traitement des données
- Responsabilité (Accountability)
  - Privacy-by-Design, Sécurité, Data Protection Officer, Registre, Notification des incidents …
- Droit des personnes
  - Droit à l'oubli, Droit d'accès, **Droit à la portabilité (art. 20)**…

Loi Lemaire pour une République Numérique

- Portabilité et récupération des données (art. 48), droit à l'auto-hébergement (art. 41)

Consacre le droit des individus à récupérer une copie de leurs données personnelles et à les gérer par eux-même

Exemples d'initiatives

Récupération de données médicales et énergétiques - USA

Blue Button Download My Data

Green Button Connect My Data

Données généralistes (énergie, crédit, smartphone …) - UK

MIDATA

« initiative gouvernementale regroupant industriels et associations de consommateurs »

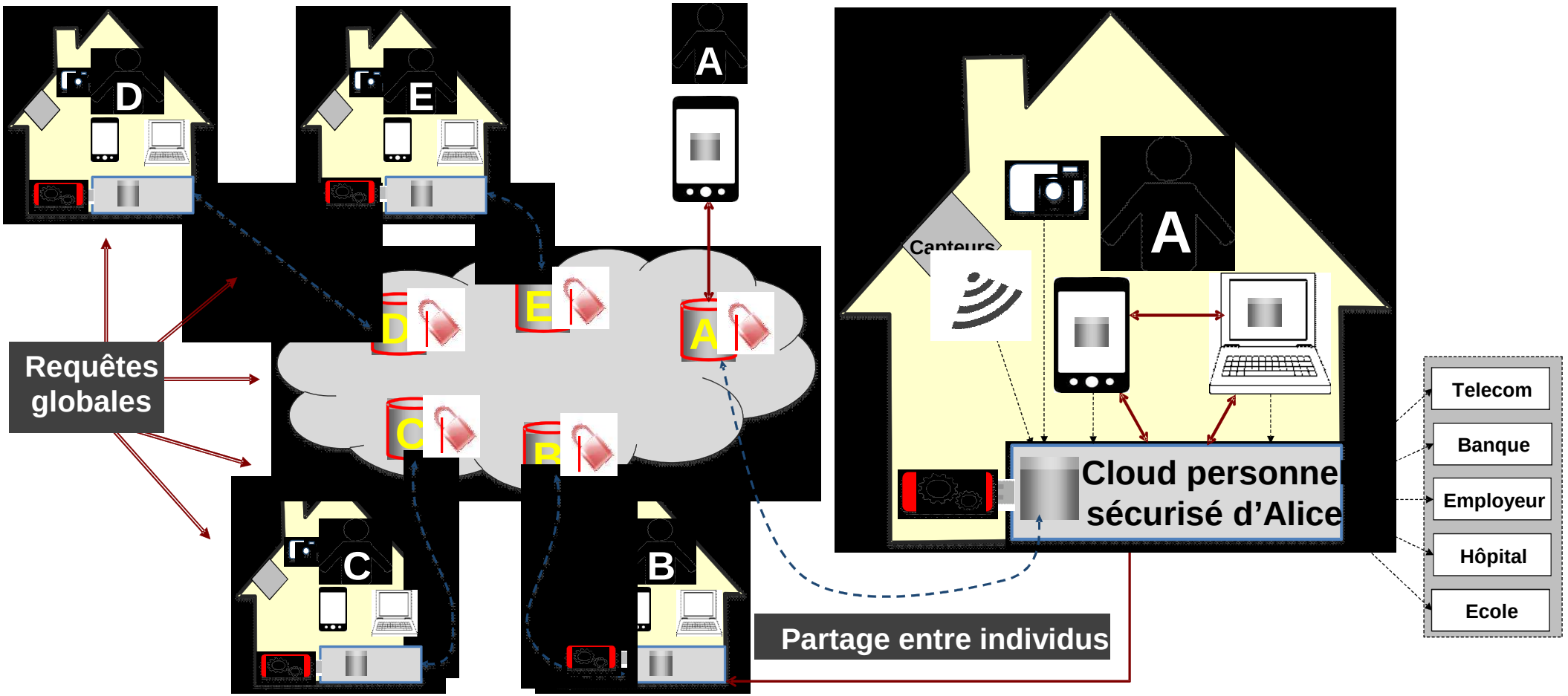
Self-Data en France

MES INFOS

« La production, l'exploitation et le partage de données personnelles par les individus, sous leur contrôle et à leurs propres fins »

Cloud Personnel/PDMS (Personal Data Mgt Syst)

- Capter, stocker et gérer les données à la source, sous le contrôle de l'individu, pour mieux maîtriser leur cycle de vie et unifier les silos d'information



Que peut-on faire avec ?

Organiser l'ensemble de son patrimoine numérique et bénéficier d'outils puissants de recherche, de visualisation et de partage

- Par mots-clés, par facettes, visualisation de masses de données historisées

Personal Big Data

- Croiser des données auparavant en silos
  - Santé : quantified-self × alimentation × examens biologiques × …
  - Optimisation du budget : tickets de caisse × factures × relevés bancaires × …
  - Optimisation du temps : usage apps smartphone × trajets × télé × …

Big Personal Data

- Croiser des données de multiples individus (ex: études épidémiologiques, sociologiques, économiques, deep learning) en préservant la vie privée de chacun

Et beaucoup à inventer …

Faut-il y croire ?

Hippocratic DB [VLDB'02], Personal Data Server [VLDB'10], OpenPDS [PLOS'14] …

- Des succès d'estime (académiques)

En 2015, Serge Abiteboul y croit ☺

- *Managing your digital life with a Personal information management system*, Com. of the ACM

En 2018, Tim Berners Lee aussi ☺ ☺

"…. **the web has evolved into an engine of inequity and division …** ***Solid is how we evolve the web in order to restore balance - by giving every one of us complete control over data, in a revolutionary way …"*** **https://www.inrupt.com/blog/one-small-step-for-the-web**

Ainsi que de nombreuses startups, y compris en France : CozyCloud, Helixee, Lima …

Quelques challenges autour du cycle de vie des données personnelles

- Collecte automatique des données (scrappers, IoT …)
- Intégration/nettoyage des données
- Indexation sémantique des données
- Analyse de données Privacy-by-Design
- Destruction des données / droit à l'oubli

- Et la sécurité !

Quelques challenges liés à la sécurité / privacy

Modèles de contrôle d'accès et d'usage

- Adapté à l'utilisateur lambda *(empowerment, audit, ease of use)*

Chiffrement des données

- Adapté à l'IoT, adapté à l'archivage

Anonymisation

- Compromis protection/utilité, décentralisation, personnalisation, differencial privacy

Calculs distribués

- Privacy-Preserving Big Personal Data

Principes architecturaux

- Paradoxe du Cloud Personnel faisant suite au *Privacy Paradox*

Diversité des architectures de Cloud Personnel/PDMS

PDMS ⇒ tout notre patrimoine numérique au même endroit

Hébergement en ligne traditionnel

- BitsAbout.me, Meeco, CozyCloud …
- Trust model : confiance dans l'hébergeur, le PDMS provider, les Apps

Hébergement Zero-Knowledge

- SpyderOak, MyDex, Sync …
- Trust model : confiance dans l'environnement client et les Apps

Auto-hébergement

- SW (OpenPDS, DataBox …), HW(Helixee, Lima, CloudLocker …), Secure HW (PDS, Trusted Cells …)
- Trust model : confiance dans le PDMS provider et dans le SW ou HW ou secure HW

Paradoxe du Cloud Personnel

Souveraineté accrue ⟹ surface d'exposition accrue

Définir de nouvelles propriétés de sécurité capturant le cycle de vie des données dans un PDMS

Cycle de vie des données

- Collecte/stockage-restauration/exploitation/partage/destruction

Des différences avec un environnement Cloud traditionnel

- Utilisateur ‘naif’ vs. administrateurs experts (DBA, DSA)
- Chiffrement des données + archive avec clé issue de password vs. HSM
- Riche écosystème d’Apps untrusted vs. Applications trusted et établies
- Confiance du *Querier* dans les données consultées ?
- Politique de partage adhoc vs. modèles RBAC/MAC administrés de façon centralisée

Définir de nouvelles propriétés de sécurité capturant le cycle de vie des données dans un PDMS [IS'18]

Propriétés requises pour un *Extensible&Secure PDMS (ES-PDMS)*

Piped Data Collection

- Eviter la fuite des credentials
- Confiner le code de collecte à l'insertion de données dans le PDMS

Mutual Data at Rest Protection

- Prévenir les attaques de confidentialité et d'intégrité sur les données stockées et l'archive, y compris de la part du propriétaire de PDMS (*owner*)
- Le secret protégeant l'archive doit être reconstructible
- Ce secret ne peut être reconstruit que par un ES-PDMS, pas même par l'owner

Définir de nouvelles propriétés de sécurité capturant le cycle de vie des données dans un PDMS (con't)

Bilaterally Trusted Personal Computation

- Uniquement les données requises pour le calcul sont accédées
- Seul le résultat final est exposé au *querier*
- L'exécution produit une trace d'audit accessible à *l'owner*
- Le querier doit recevoir une preuve que le calcul a été effectué avec le bon code, sur les bonnes données

Mutually Trusted Collective Computation

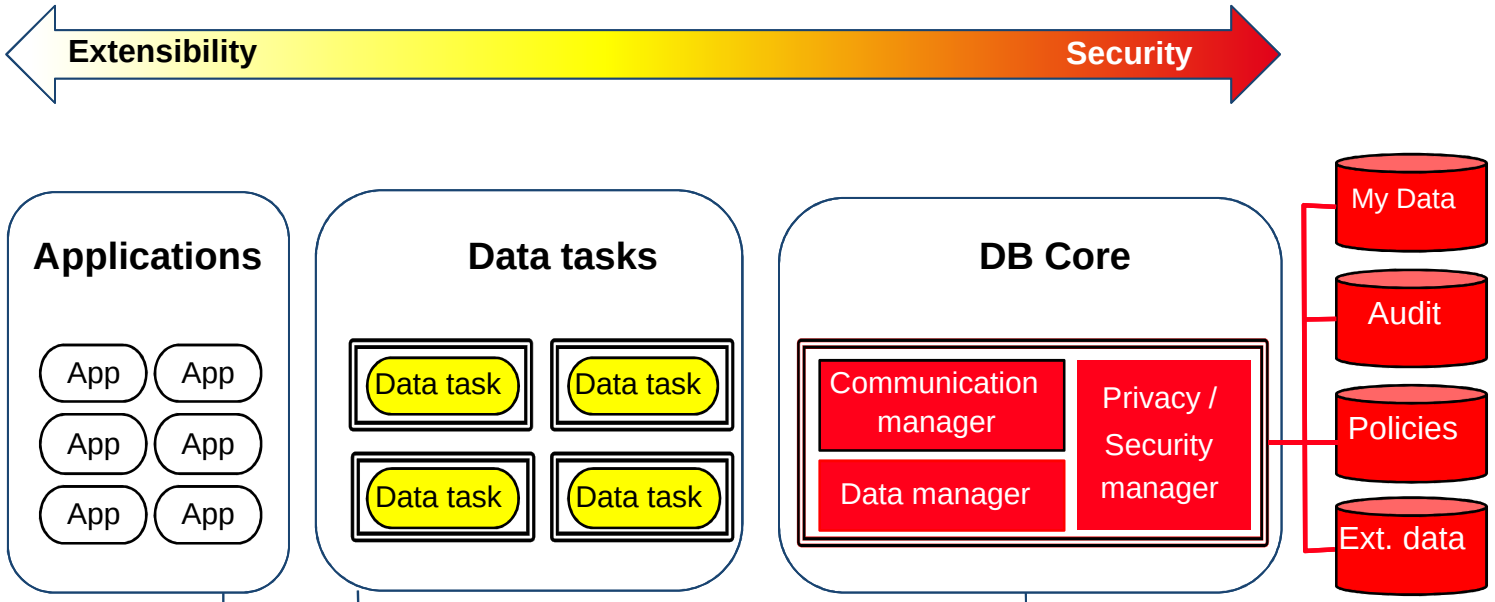
- Transposition des conditions précédentes à un calcul distribué sur n participants

Controlled data dissemination

- L'intégrité et la confidentialité des interactions entre l'owner et son PDMS sont garanties (notamment concernant la définition de la politique de partage)
- Les règles de dissémination énoncées sont incontournablement appliquées par le PDMS

Problématique architecturale [IS'18]

Difficulté : Satisfaction des propriétés de sécurité en environnement ouvert

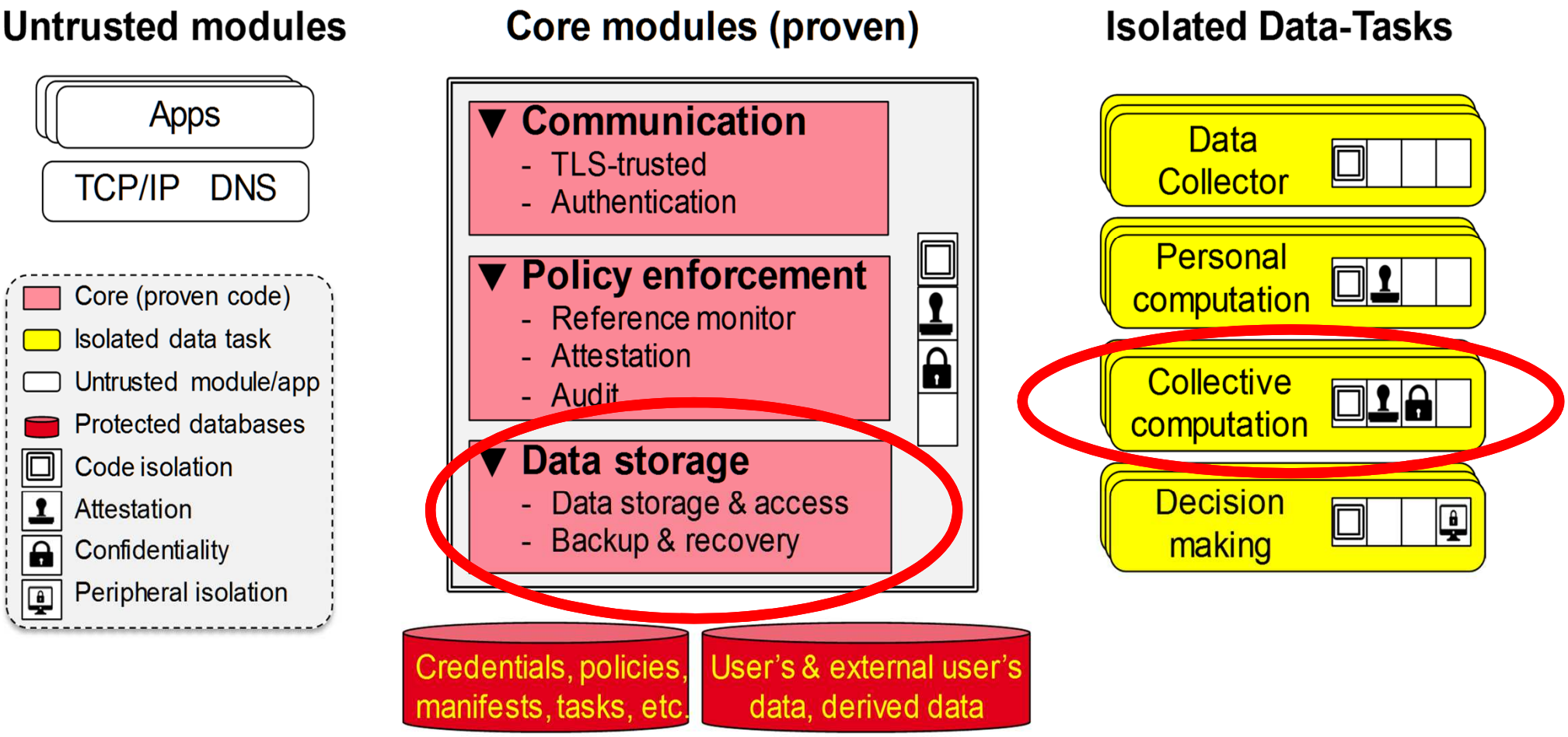


Code quelconque
Pas d'hyp. de confiance

Code extensible
Env. d'exec isolé
(**T**rusted **E**xec **E**nv – **TEE**)
Ex: SGX, TrustZone …

Code fermé, 'minimal', prouvé
Env. d'exec sécurisé
(**S**ecure **E**lement – **SE**)

Problématique architecturale (con't)



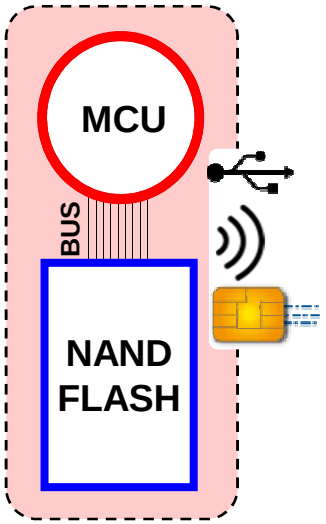
Rapide focus sur : Data Storage & Access

Evaluation de requêtes dans le Secure Core (Secure Element)

- Contrôle d'accès assertionnel = autoriser des traitements sur des Raw Data sensibles sans les dévoiler (équiv. Vues relationnelles)
- Exemple = calcul d'agrégat, sous-ensemble de valeurs répondant à une qualification
- Un enjeu de *privacy* et de *soutenabilité*

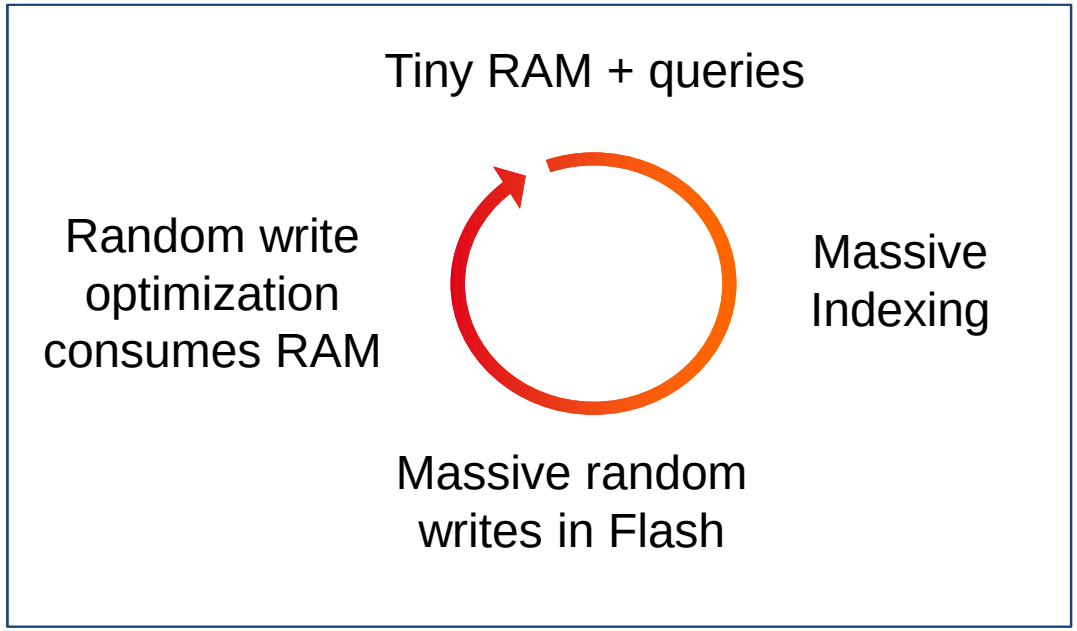
Difficulté : Gestion de données en environnement fortement contraint

- Microcontrôleur
  - RAM très faible (5KB ~ 128KB) et ratio RAM/Stable storage décroissant
  - Encore plus faible dans les SE
- Mémoire stable NAND Flash (GB-sized)
  - erase-before-rewrite, limited erase cycles, FTL unpredictability, cost of random (re)writes



Data Storage & Access (con't)

Une combinaison de contraintes HW conflictuelles (relative à la gestion de données)



Sequential indexing
+
Pipelined query execution

Nécessite de revoir en profondeur les stratégies d'indexation et d'évaluation de requêtes BD

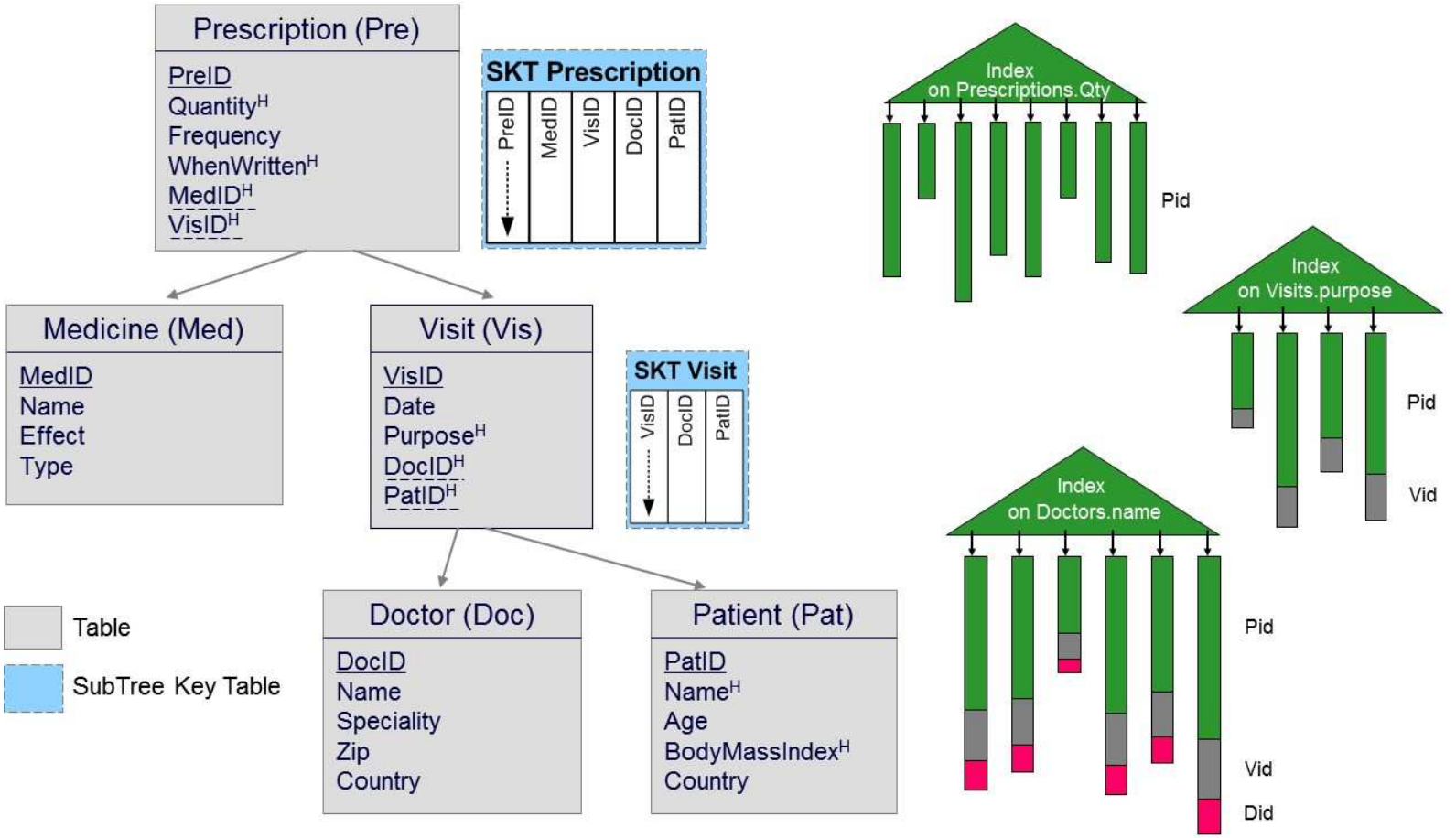
Application à des requêtes SQL-like [DAPD'14]

Index purement séquentiels

Hyp : schéma de BD arborescent

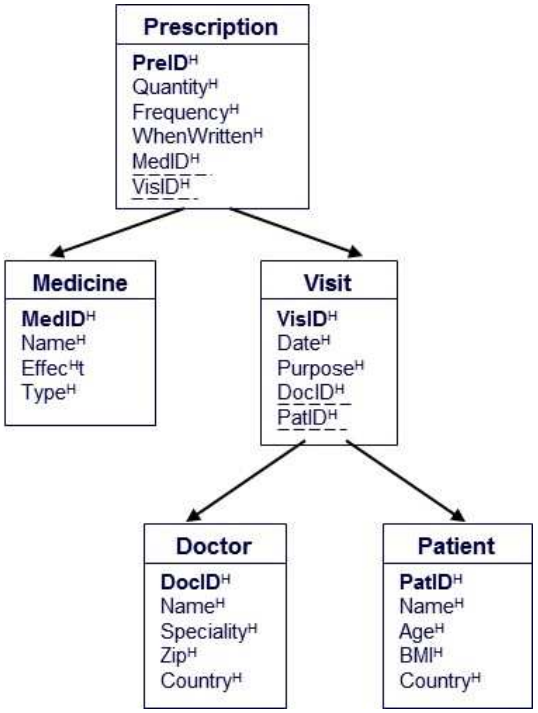
SKT : Subtree Key Table (index de jointure généralisé)

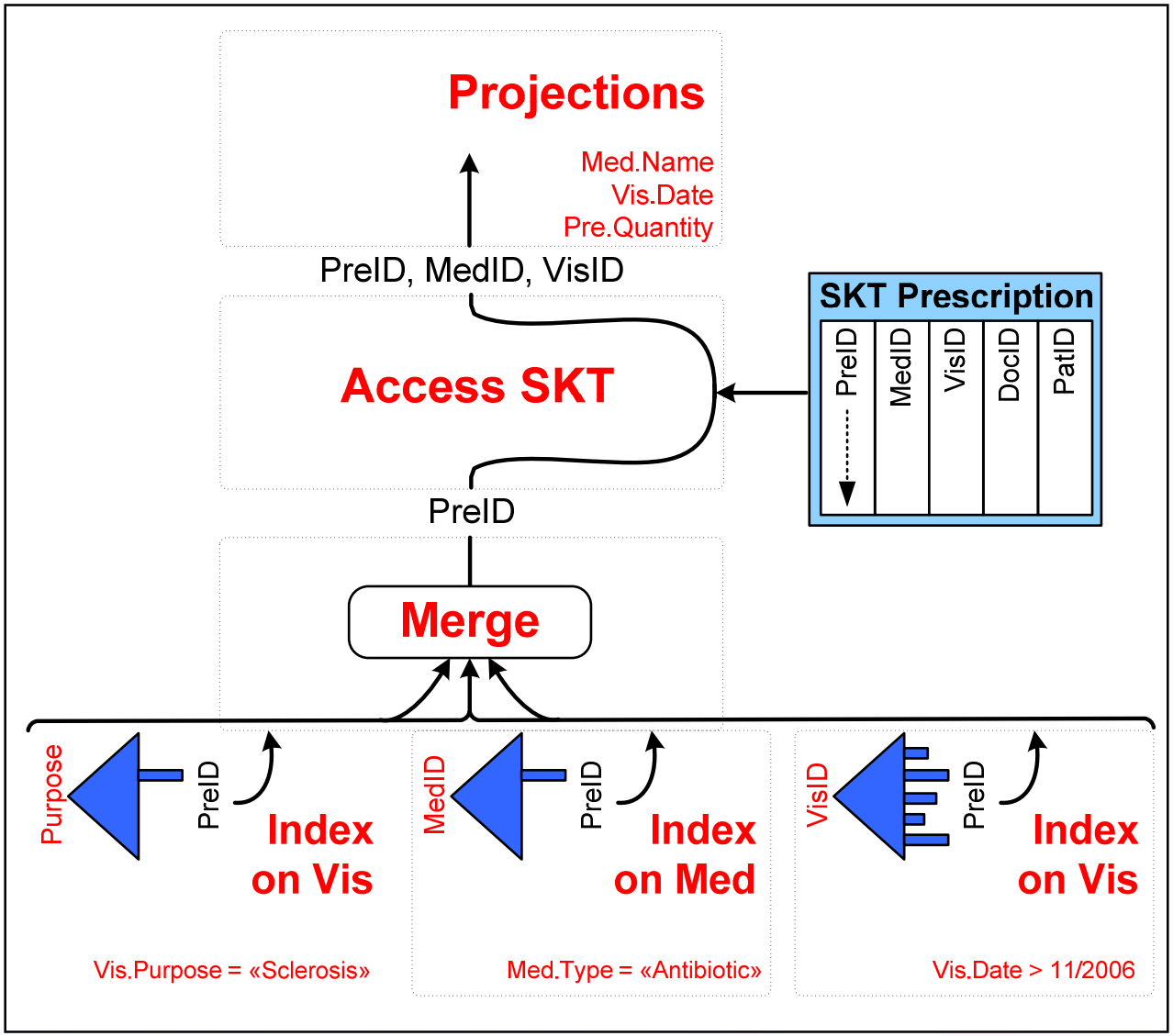
Climbing Index : multi-tables, basé sur des Bloom filters



Execution de requête 100% pipeline

```
SELECT  Med.Name, Pre.Quantity, Vis.Date
FROM    Medicine Med, Prescription Pre, Visit Vis
WHERE   Vis.Date > 11/2006
    AND Vis.Purpose = “Sclerosis”
    AND Med.Type = “Antibiotic”
    AND Med.MedID = Pre.MedID
    AND Vis.VisID = Pre.VisID;
```





Généralisation : règles de design [IS'17, VLDB'15]

(1) Write-once partitioning

- Découper l'index en *n* partitions successives de taille bornée (RAM) et jamais modifiées après leur écriture en Flash

(2) Linear pipelining

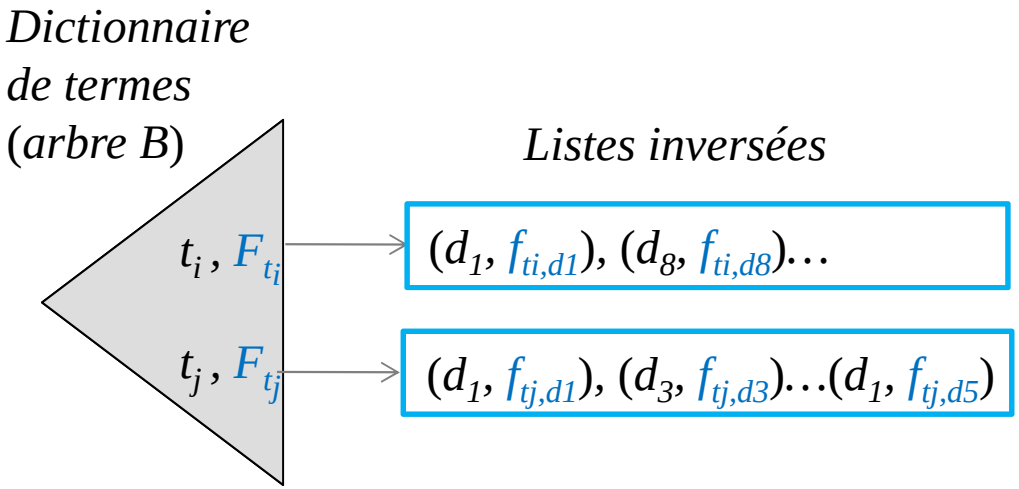
- Définir un algorithme d'évaluation pipeline (en RAM bornée et à 'coût linéaire') sur l'ensemble des partitions

(3) Background linear merging

- Proposer un algorithme de fusion des partitions (pipeline) pour limiter leur nombre (log.) et passer à l'échelle

Requêtes top-k [IS'17, EDBT'16, VLDB'15]

- **Méthodes BD classiques**
  - Index inversé (modifié 'en place')

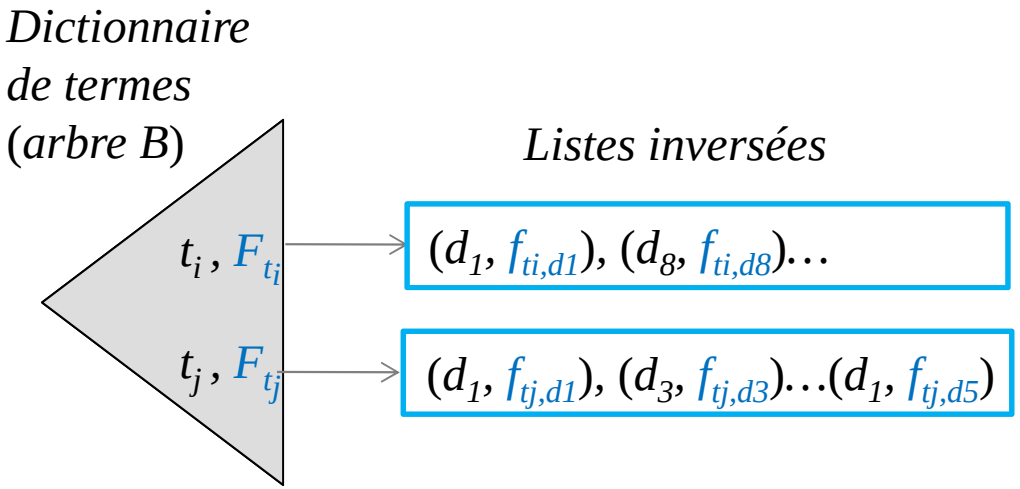


  - Evaluation d'une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

Requêtes top-k

- **Méthodes BD classiques**
  - Index inversé (modifié ‘en place’)



  - Evaluation d’une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

- **Design du composant**

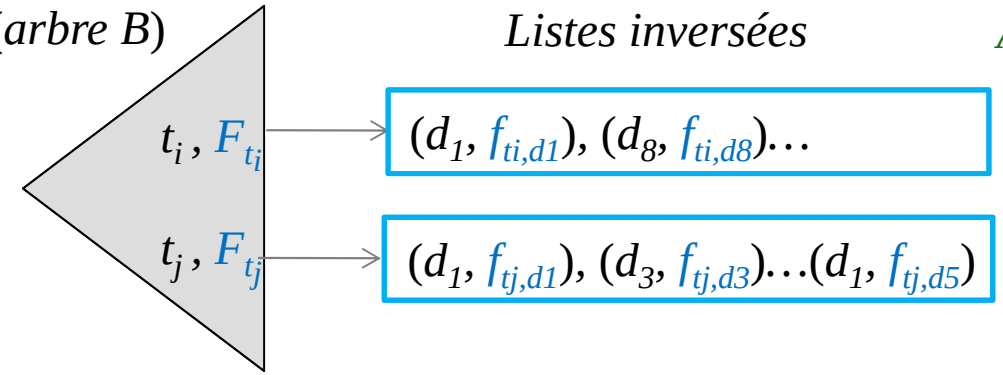
  (1) Write-once partitioning

Requêtes top-k

- **Méthodes BD classiques**
  - Index inversé (modifié ‘en place’)

Dictionnaire de termes (arbre B)

Listes inversées



  - Evaluation d’une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

- **Design du composant**
  - **(1) Write-once partitioning**
  - **(2) Linear pipelining**

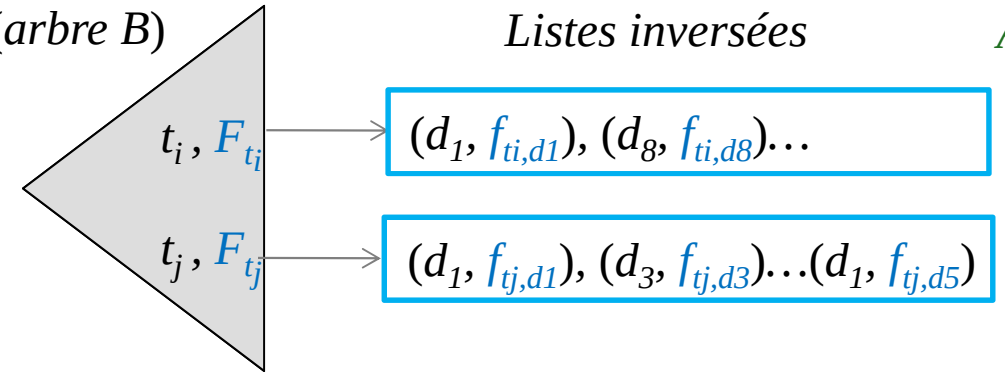
Algorithme RAM bornée, traverse 1 fois chaque partition

Requêtes top-k

- **Méthodes BD classiques**
  - Index inversé (modifié ‘en place’)

Dictionnaire de termes (arbre B)

Listes inversées

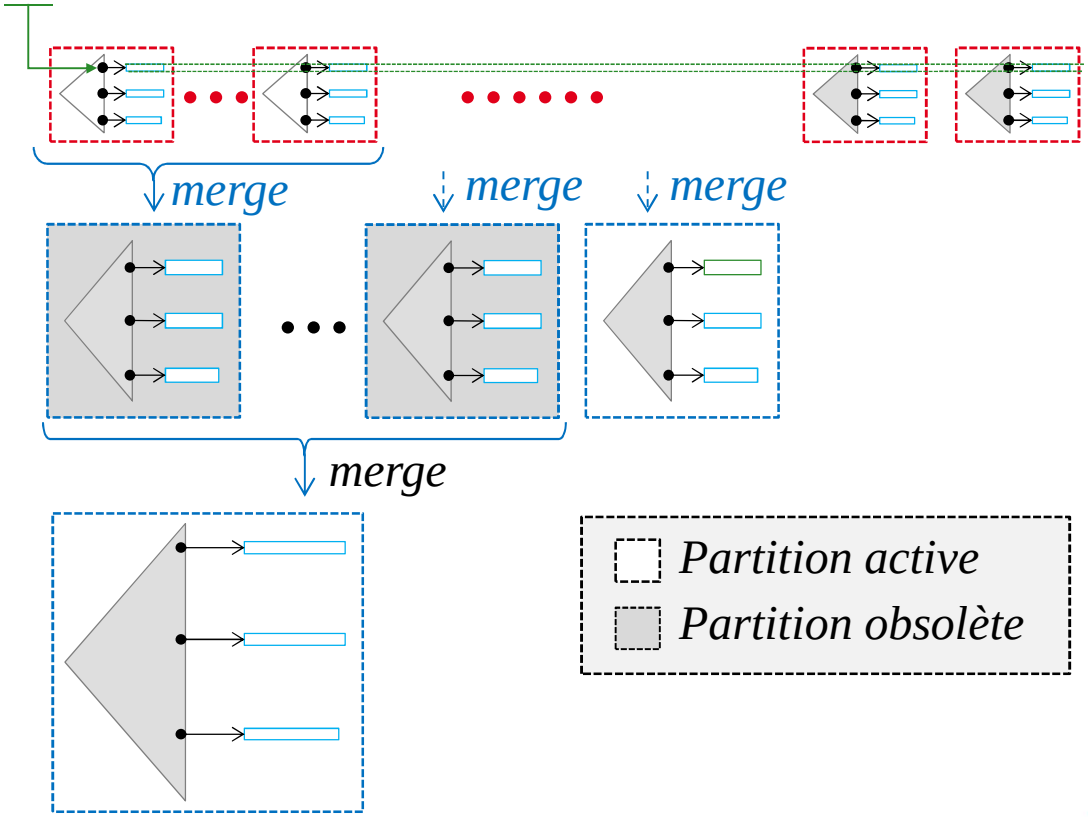


  - Evaluation d’une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

- **Design du composant**
  - **(1) Write-once partitioning**
  - **(2) Linear pipelining**
  - **(3) Background linear merging**

Algorithme RAM bornée, traverse 1 fois chaque partition

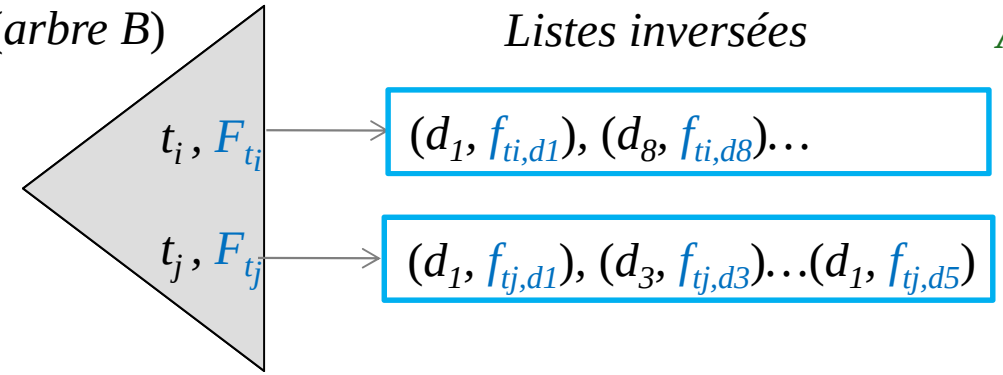


Requêtes top-k

- **Méthodes BD classiques**
  - Index inversé (modifié ‘en place’)

Dictionnaire de termes (arbre B)

Listes inversées

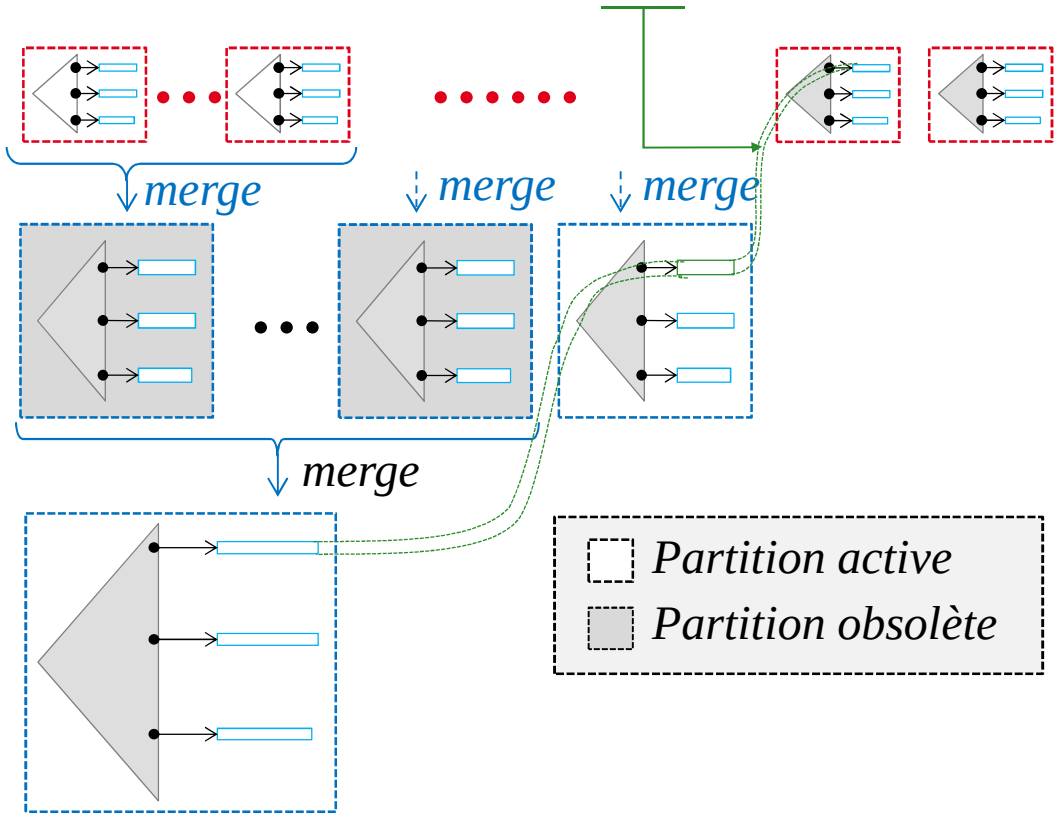


  - Evaluation d’une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

- **Design du composant**
  - **(1) Write-once partitioning**
  - **(2) Linear pipelining**
  - **(3) Background linear merging**

Algorithme RAM bornée, traverse 1 fois chaque partition

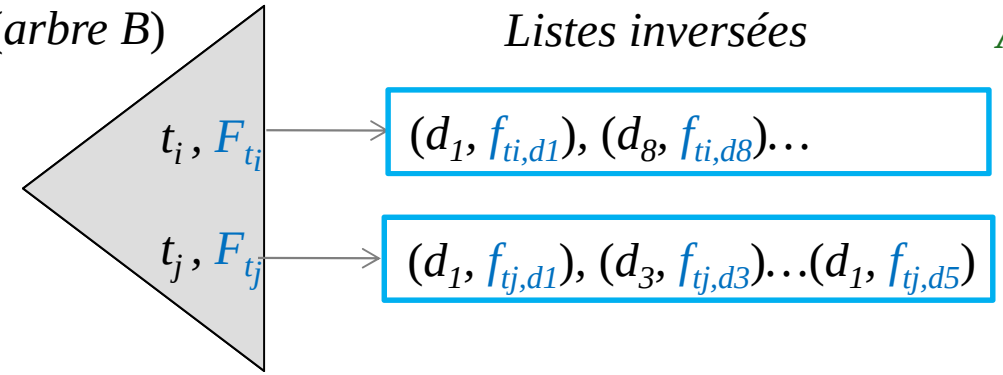


Requêtes top-k

- **Méthodes BD classiques**
  - Index inversé (modifié ‘en place’)

Dictionnaire de termes (arbre B)

Listes inversées

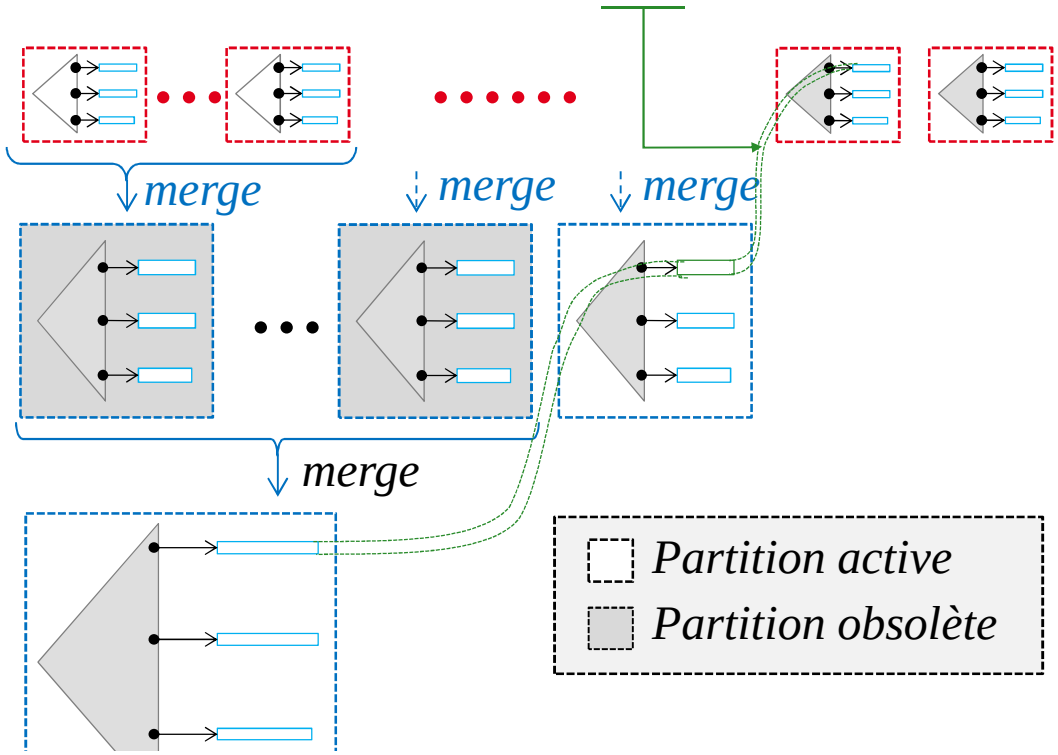


  - Evaluation d’une fonction score
  - (matérialisation en RAM)

$$\text{TF-IDF(d)} = \sum_{\{t_i\} \in Q} \left(f_{t_i,d} * \text{Log}(N / F_{t_i}) \right)$$

- **Design du composant**
  - **(1) Write-once partitioning**
  - **(2) Linear pipelining**
  - **(3) Background linear merging**

Algorithme RAM bornée, traverse 1 fois chaque partition



➔ **Composant capable d’indexer des millions d’entrées**

Rapide focus sur : Collective Computations

Objectif

- Calculs distribués quelconques sur données réelles, large échelle et assurant que
  (1) consentement, limited collection et limited retention sont respectés,
  (2) seul le résultat du calcul est dévoilé à une tierce partie,
  (3) le querier a la garantie d'un résultat calculé honnêtement
- Protocoles SMC et Gossip répondent au point (2) mais ne satisfont pas la combinaison large échelle / généricité
- Differential privacy, homomorphic encryption ne répondent pas à ce problème
- Trusted central server (e.g., TrustedDB …) contredit l'hypothèse de décentralisation

Hypothèses

- Chaque participant satisfait *a priori* les propriétés de sécurité du PDMS
- Majorité de participants Honest-but-Curious et quelques participants Malicious (side-channel attacks sur TEE)

Calcul décentralisé de requêtes Agrégat avec GroupBy [EDBT'14, TODS'17], extension au MapReduce [CoopIS'15]

Trust model : asymmetric architecture

- Tous les participants sont *Honest* (SE) *(même si leur owner est malicious)*
- Une Supporting Server Infrastructure (SSI) *Honest-but-Curious*

Principe

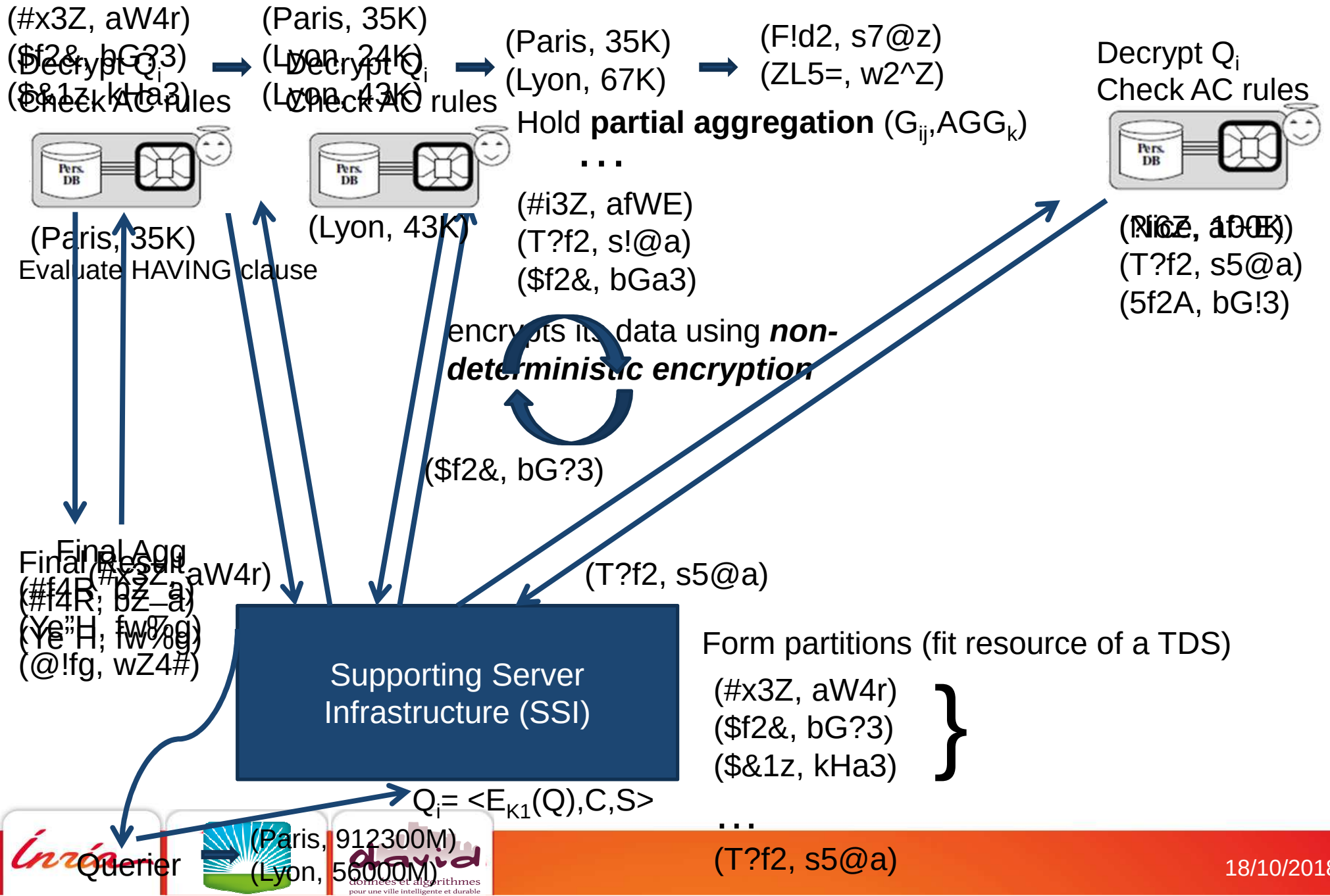
- Chaque participant contribue avec ses données chiffrées pendant la phase de collecte
- Le SSI forme des partitions de ces données qui sont ensuite déchiffrées par les participants pendant la phase de calcul (n rounds)
- Contre-mesures pour minimiser le nombre de rounds sans que le flot de données chiffrées ne puisse subir d'attaque statistique (noise-based, histogram-based)

Limite

- Partage d'un secret protégé par le SE (tamper-resistant) de chaque participant

Secure Aggregation

Q: SELECT SUM(*Salary*) GROUP BY *City*
HAVING SUM(Salary) > 50M



Towards a more general solution

Trust model

- Untrusted user devices and infrastructure
- Large set of trusted TEEs, small set of corrupted TEEs (side channel attacks)
- Trusted computation code (no restriction on the code itself)

Security goals

- Code safety : compliance with consent / limited data collection & data retention
- Mutually Trusted Collective Computation : querier and participants get the guarantee that the genuine code has been computed on genuine data
- Resilience to attacks and failures

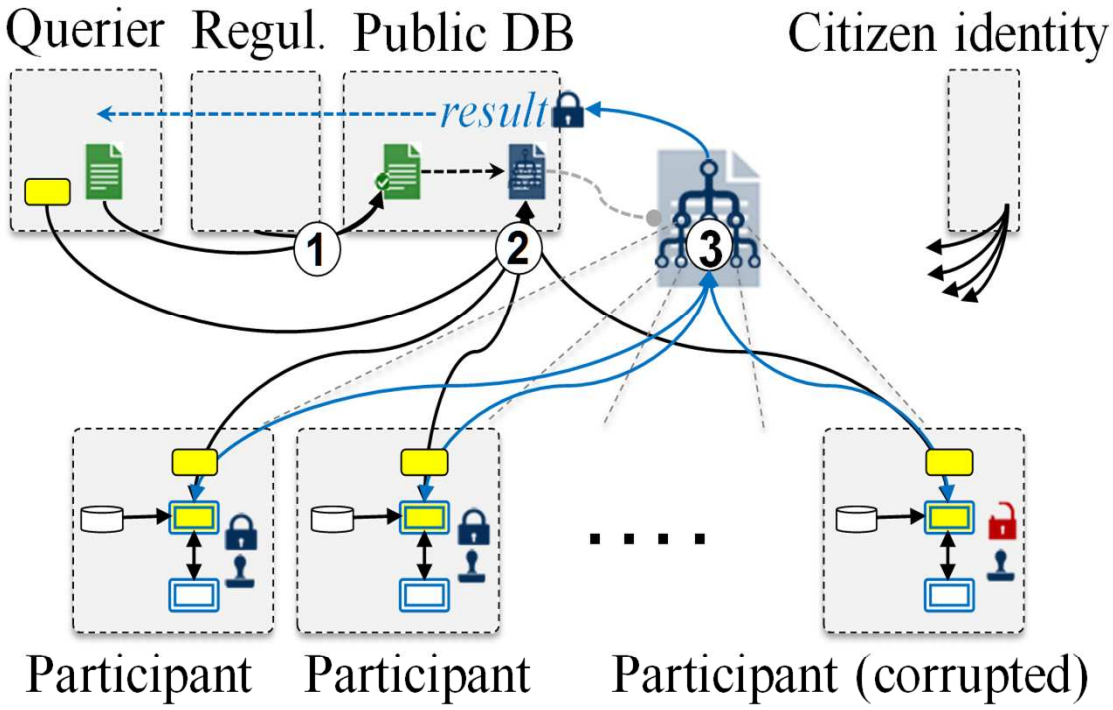
Principle

- Based on manifests run inside SGX enclaves

Global picture

Legend:

- Actors (citizen/legal person)
- Manifest (spec.)
- Manifest (exec.)
- TEE protocol driver
- TEE computation code
- Untrusted protocol driver
- PDMS (personal data)
- Attestation
- Confidentiality
- Corrupted TEE



① Querier specifies a manifest and publish it; Regul. certifies the manifest

② Participants consent to manifest; Participants/querier build executable manifest

③ Participants execute manifest (query plan); Querier retrieves results (encrypted)

Counter-measures must guarantee

- Locally checkable validity of executions (by each participant)
- Provably random executions

Equipe PETRUS : PErsonal and TRUSted Cloud UVSQ / DAVID & Inria Saclay

Un objectif

- Concevoir une plateforme PDMS répondant au Paradoxe du Cloud Personnel

4 axes de recherche

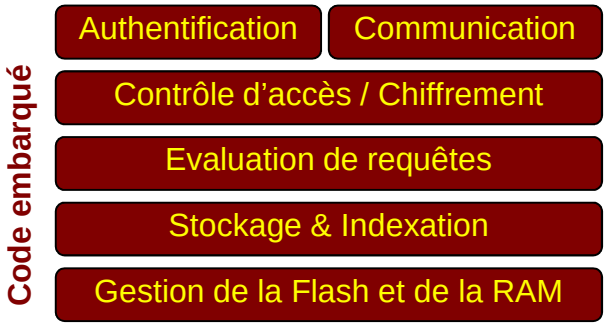
- Architectures pour le Cloud Personnel
- Modèles d'administration et mise en oeuvre sécurisée des politiques
- Protocoles décentralisés de gestion de données personnelles
- Implications juridiques, économiques et sociétales

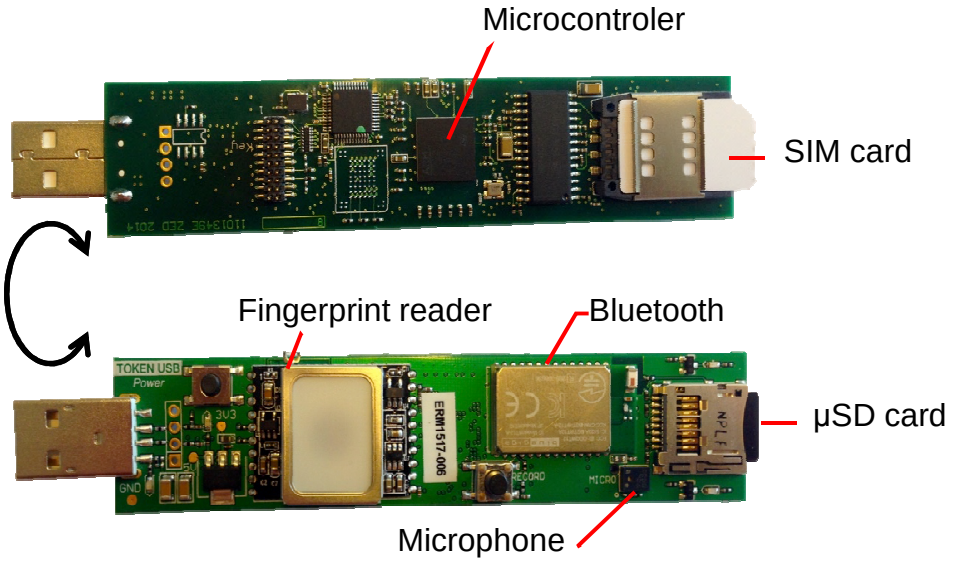
Une volonté d'expérimentation et de transfert

- Une plateforme concrète : PlugDB
- Un IILab (Inria-UVSQ-Hippocad) : OwnCare

PlugDB : un serveur personnel sécurisé

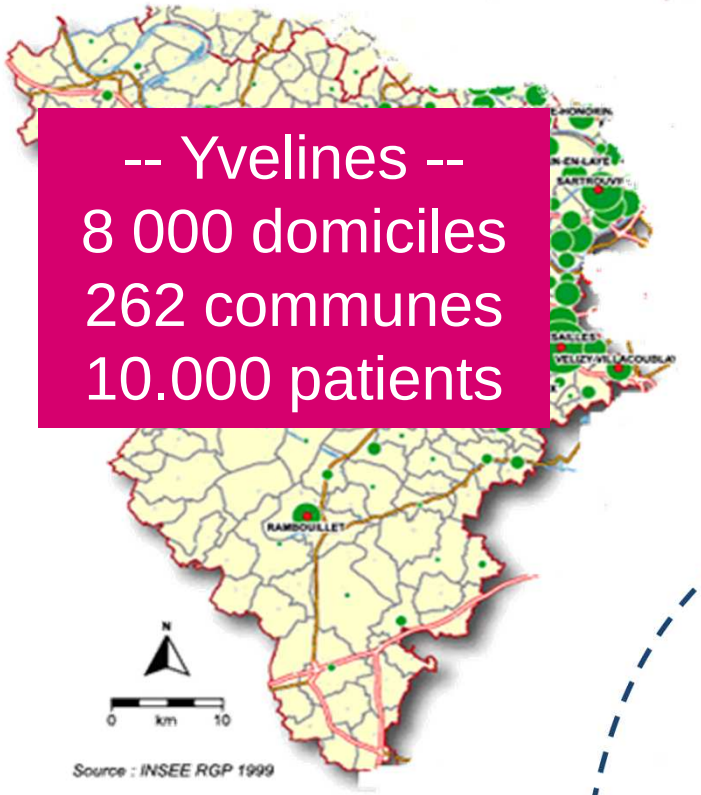
- **Stockage de données embarquées, indexation, requêtes, droits d'accès, chiffrement, transactions**
- **Moteur BD embarquable dans un microcontrôleur sécurisé**
  - Verrou: tiny RAM + NAND Flash + requêtes





Inria Innovation Lab OwnCare
& application DomYcile (CD78/Hippocad/Petrus)

Déploiement d'un dossier médico-social personnel sur le département des Yvelines



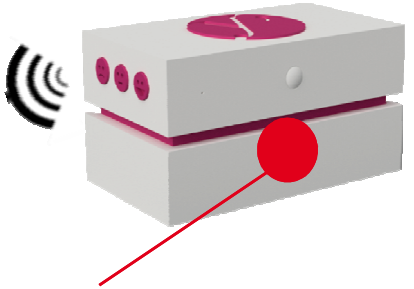
Objectif 1

Contrôle d'effectivité

Objectif 2

Coordination des soins et des prestations sociales

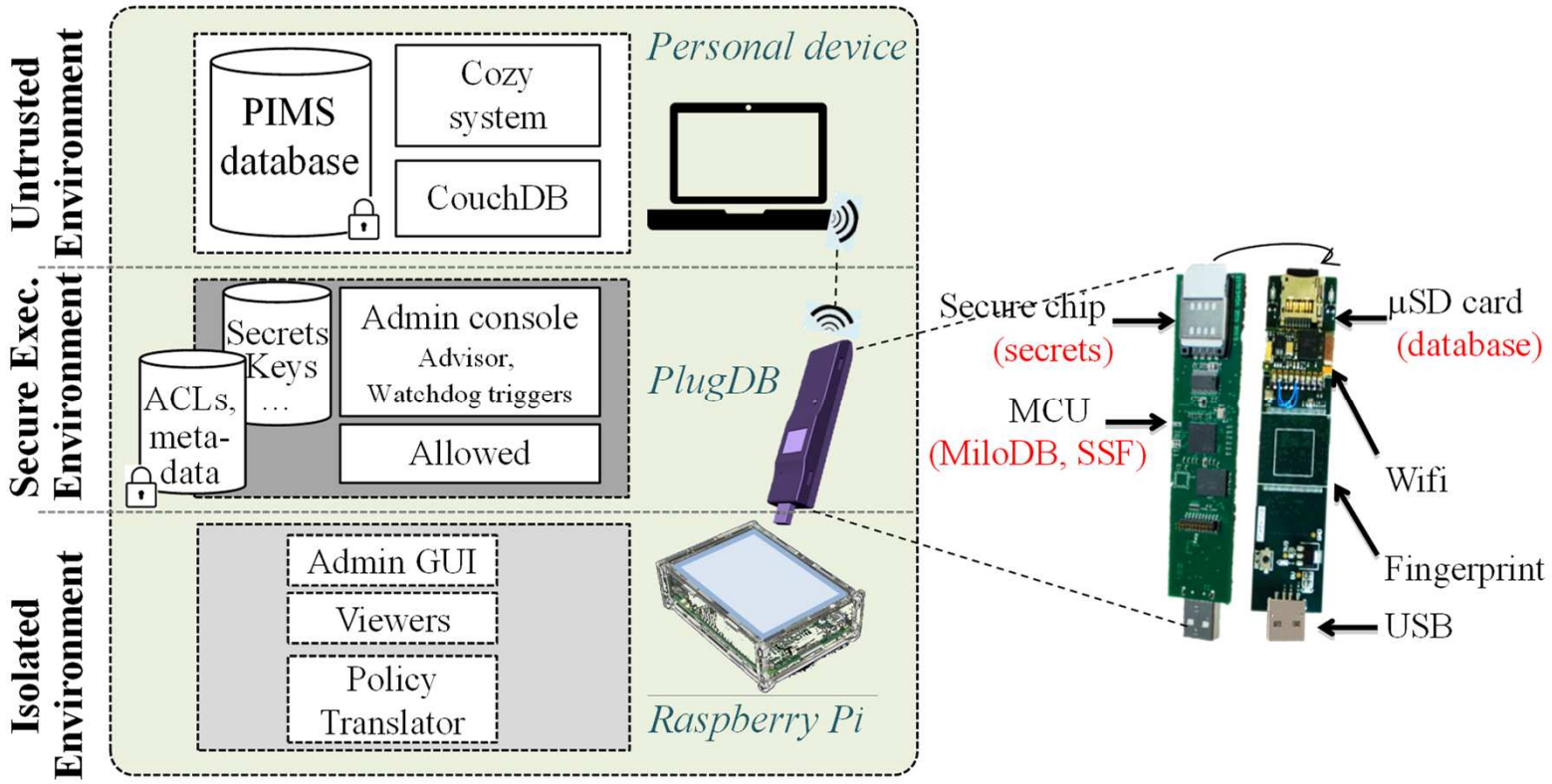
Sigfox



PlugDB inside

Cozy sécurisé (Cozy Cloud / Petrus)

- Cozy Cloud : leader français du Cloud personnel
- Contributions : sécurisation des données stockées, modèle ad-hoc de contrôle d'accès et d'administration des politiques de partage



Le Cloud Personnel: sujet interdisciplinaire par excellence

Entre …

- **informaticiens** : challenges à la croisée des SI, de l'IA, de la cryptographie, des algorithmes distribués, des architectures logicielles et matérielles
- **… économistes :** autour des modèles économiques liés à l'exploitation des données perso.
- **… et juristes :** autour des principes de responsabilité, de privacy-by-design, droit à l'oubli …

De multiples questions ouvertes par le cloud personnel

- Comment éviter un effet boomerang de la souveraineté (redéfinition des responsabilités) ?
- Quel statut pour les données hébergées relatives à d'autres individus ?
  - 98% of MIT students are happy to trade their friends' email addresses for free pizza... (National Bureau of Economic Research)
- La monétisation des données personnelles comme incitation à la protection ?
- Les données personnelles considérées comme moyen de paiement d'un service ?
- Jusqu'où ouvrir le droit à la récupération des (méta-)données ?
  - Motif de collecte, période de rétention, transferts …
- Cloud personnel choisi ou imposé ?
- …

données et algorithmes
pour une ville intelligente et durable

Questions ?

PETRUS Project - team

UVSQ / DAVID & INRIA Saclay

https://www.inria.fr/equipes/petrus

https://project.inria.fr/plugdb