Managing Personal Data with Strong Privacy Guarantees

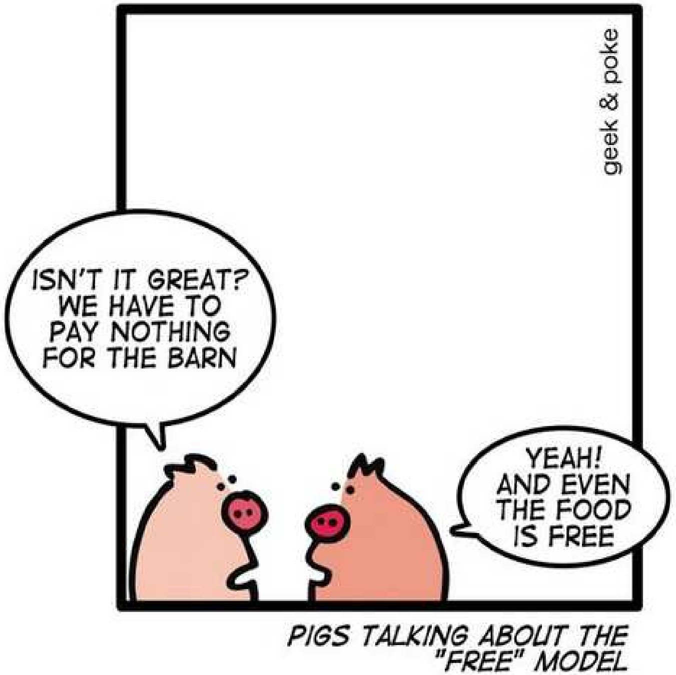
Nicolas Anciaux, Benjamin Nguyen & Iulian Sandu Popa
INRIA Paris-Rocquencourt & University of Versailles St-Quentin

EDBT'13 Tutorial
25th March 2014

An era of massive generation of (personal) data

Data sources have turned digital

- Analog processes
  - e.g., silver photography
- Paper interactions
  - e.g., banking, administration
- Mechanical interactions
  - e.g., opening a door
- Communications
  - e.g., email, SMS, MMS, Skype



Good news: it's free… ☺

All this information is stored in data centers

- 112 new emails per day → Mail servers
- 65 SMS sent per day → Telcos
- 800 pages of social data → Social networks
- Web searches, list of purchases → Google, Amazon

1- WHY?

2- Is this a problem?

“Personal data is the new oil” (World Eco. Forum)

Is this good news ?

- **$2 billions a year spend by US companies on third-party information about individuals (Source: Forrester Report)**

- **$44.25 is the estimated return on $1 invested in email marketing (Source: Direct Marketers Association)**
NB: ERoI is around $20 in the oil production industry…

- **Companies managing personal data boast impressive market values**
  Facebook: value / #accounts ≈ $50
  Google: $38 billion business sells ads based on how people search the Web
  Amazon (knows purchase intent), mail order systems companies (gmail), loyalty programs (supermarkets), banks & insurrance, employement market (linkedIn, viadeo), travel & transportation (voyages-sncf), the « love » market (meetic), etc.

We are sitting on valuable oil fields…
but we have left them unguarded

How do the new oil producers behave?

They offer to exploit our oil fields for free

… and can know all about us

They offer *free* services to us

… which do not cost that much to run

They provide *real* services (not advertised) to their *paying* customers

… which cover the costs of the services and yield healthy returns

e.g. advertisement and profiling, location tracking and spying, …

They process our personal data

… within sophisticated *data refineries*

… REGARDLESS OF PEOPLE'S PRIVACY !

It's the business model !

A privacy preserving alternative to extreme centralization?

The current Web model is fully centralized

Intrinsic problem #1: personal data is exposed to sophisticated attacks

- High benefits to successful hack
- One person negligence may affect millions

Intrinsic problem #2: personal data is hostage of sudden privacy changes

- Centralised administration of data means *delegation of control*
- Regular changes: application (and business) evolution, mergers and acquisition, based on polls (e.g., Facebook 2012)

Increasing security is only a partial solution since it does not solve those intrinsic limitations

- E.g., TrustedDB [BS12] proposes tamper-resistant hardware to secure outsourced centralized databases.

After all, is privacy really required

Privacy is an old-fashioned concept

- Because young people expose personal life online more likely than adults
- “privacy is no longer the social norm” (M. Zuckerberg)

Great untruth for sociologists

- Household is the adult’s private sphere, for a teen the online sphere is private
- 2013: less young daily users, while adults daily users keeps increasing

“When your mom, grandmother, auntie and all the rest of your older family members joined Facebook, it’s time to find another social media outlet to congregate.” – Teenager

Privacy has become essential

Spying impact: for companies, the place where content is stored is essential

- Companies plan to quit US clouds, estimated losses \$35-180billions (ITIF/Forrester)

“Snowden effect”: young people are more likely to manage privacy settings [Harris, Pew], and turn to ephemeral communication means (Snapchat)

Towards a new web model: trusted companies (banks) give back their data to the users, startups (Cozy@Mozilla) offer personal HW for a personal cloud !

Alternative solutions?

For the World Economic Forum (WEF) it would be:

- **"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"**

Alternative privacy preserving technical solutions are flourishing

- **E.g., Freedombox, projectVRM, Personal data servers…**

Goal of this presentation

- **Investigate solutions based on decentralization & user centric principles**
- **See how to preserve functionalities for users, and for third parties**

Outline of the tutorial

PART I. Decentralized architectures

Review of privacy-oriented decentralized solutions

Interesting attempts or a panacea ?

Abstract architecture with secure hardware

A see change ?

PART II. Resource constrained data management

Review of data management techniques for constrained HW

…needed to regulate data sharing from the edges of the Internet

PART III. Global processing

Review of existing solutions

Distributed processing on the asymmetric architecture

PERSPECTIVES. A view of expected instances

PRiSM Lab. - UMR 8144

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

informatics mathematics

PART I

Decentralized Architectures

Decentralized Architectures

Part I: Outline

Review of privacy-preserving decentralized solutions

- Infomediaries
- Vendor Relationship Management
- FreedomBox
- Decentralized Social Networks

Personal Data Server (PDS) architecture

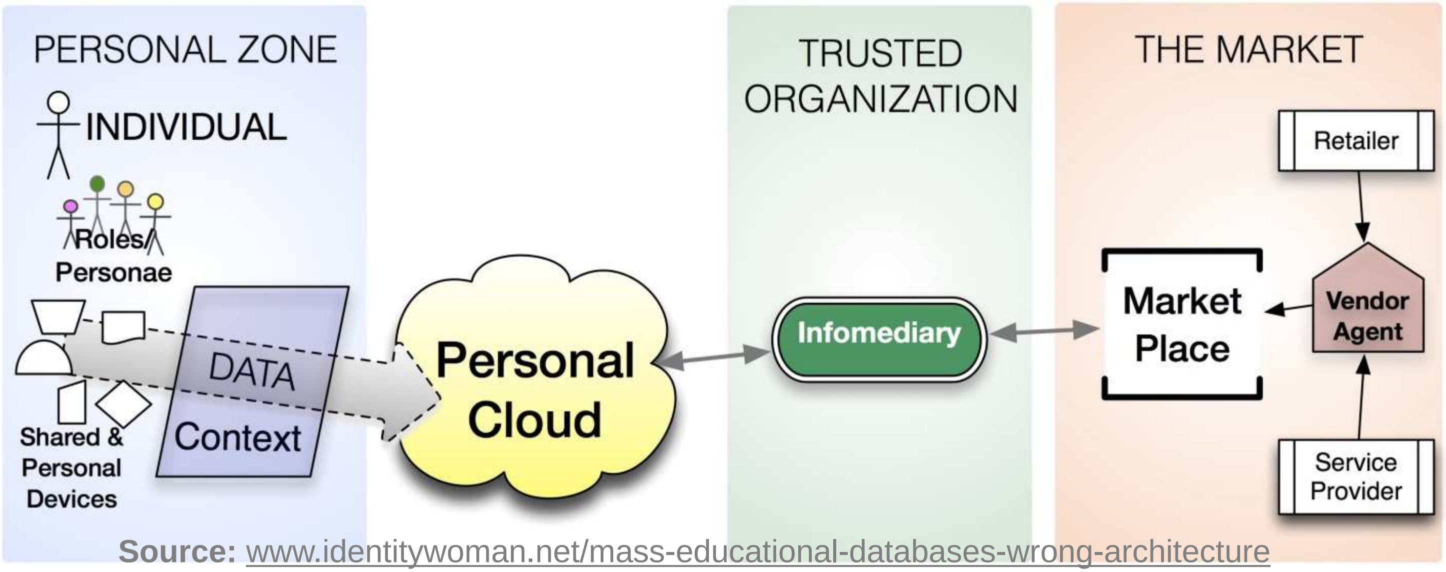
- A trusted, secure and decentralized architecture for personal data management

Infomediaries (since late 1990)

Infomediary: trusted third party helping consumers to take control over the personal information used by marketers

- Personal information is the property of individuals, not of the one who gathers it
- Personal data has value → provide users with means to monetize and profit from their information profiles
- Trust: separate the control over personal data from the service provider

AllAdvantage, Bynamite, Mydex, Adnostic, Lumeria, …



Source: www.identitywoman.net/mass-educational-databases-wrong-architecture

Vendor Relationship Management (VRM, *projectvrm.org*, since 2006)

VRM: software tools for customers to provide them independence from vendors

VRM is a software implementation of an infomediary

Observations

- **No privacy implemented in the Internet, which mainly works as a Master-Slave system**
- **Customer Relationship Management (CRM), 14billion$ market in 2013, but the customers are not involved**
  - "*Big Data is turning into Big Brother*" (Washington Post)

(Some of) VRM principles

- **Give the customer independence and a way to engage**
- **Specify your own terms of service**
- **Be able to gather, examine and control the use of your own data**

VRM tools to do all that either on your own or with the help of a "fourth party" (a third-party that works for you)

- **a dozen of open source and commercial development projects in 2012 (Privowny, Mydex, …)**

FreedomBox
(*freedomboxfoundation.org/*, since 2010)

Personal plug servers running open software to regain privacy and control

- Return the Internet to its intended P2P architecture (dehierarchicalization)
- Keep your data in your home

Base hardware requirements

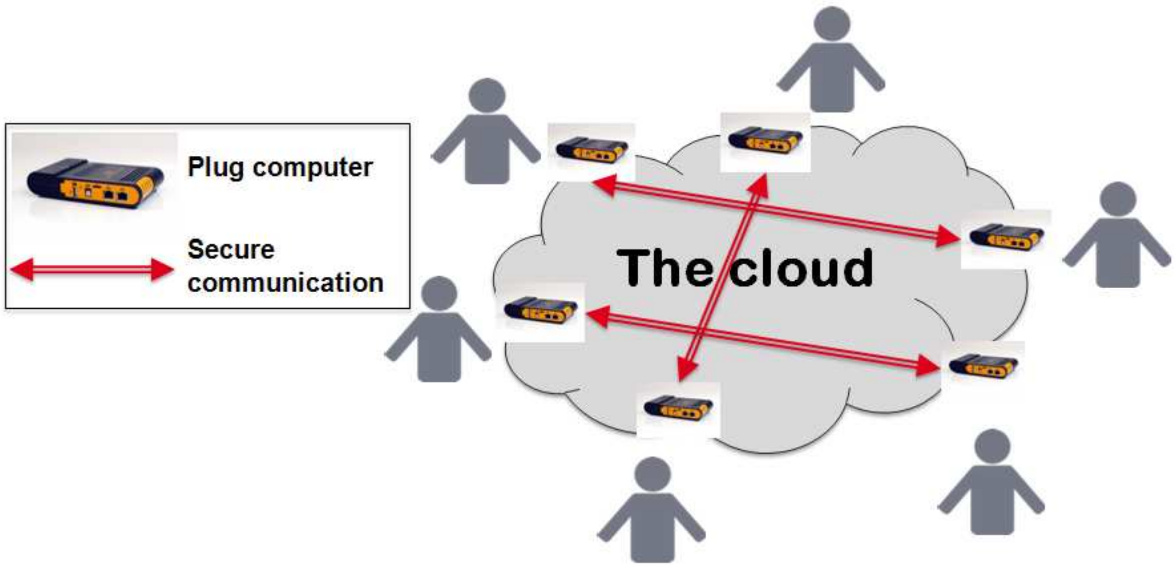
- Cheap (around 30$ for a plug server)
- Power consumption < 15W
- RAM > 256MB, Flash storage for file system > 512MB
- Communication interfaces: network, serial, JTAG
- Storage interfaces: SATA, USB, SD
- Noise level < 20dB

FreedomBox

Software stack covering a wide range of applications:

- Secure and anonymous communications
- Distributed Social Networks
- Personal Cloud
- VRM

Trust: secure and anonymous communications, open software, distribution



Decentralized Social Networks (DSN)

Distributed SN (P2P) or Federated SN (interoperable client-server implementations)

Main challenges of privacy-preserving DSN

- Secure message hosting
- Secure and anonymous message transfer

Message hosting

- Encryption and distributed hash table (Lotusnet, PeerSoN), encryption and trusted contacts (Safebook)
- Attribute-based encryption for fine-grained access control (Persona)
- Self-hosting (FreedomBox)

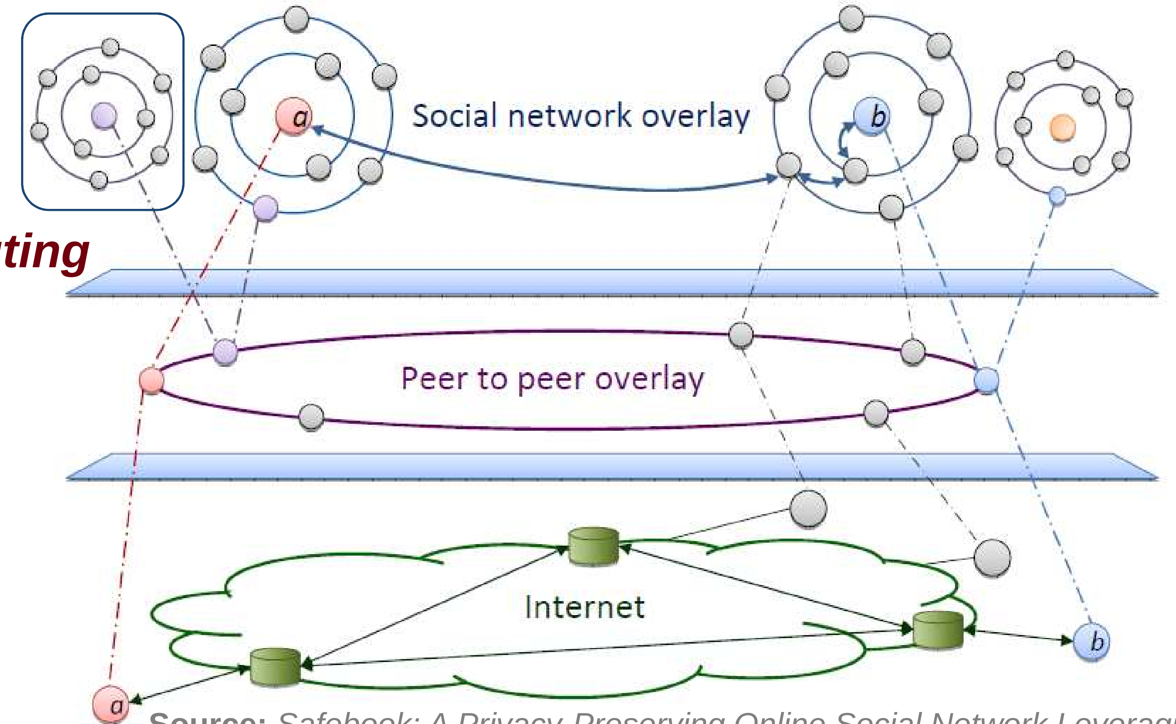
Message transfer in DSNs

Message transfer: communication privacy optimized on the social graph and physical network topology

- **Hop-by-hop encryption among trusted users (Freenet)**
- **Anonymous routing (Safebook, FreedomBox)**

Matryoshka

Anonymous routing in Safebook



Source: *Safebook: A Privacy-Preserving Online Social Network Leveraging on Real-Life Trust*

Diaspora* DSN

Diaspora* (https://joindiaspora.com/, since 2010, more than 400 thousand users in 2013, cf. Wikipedia): appeared as a response to the many privacy issues engendered by Facebook/Google

"*...our distributed design means no big corporation will ever control Diaspora. Diaspora* will never sell your social life to advertisers, and you won't have to conform to someone's arbitrary rules or look over your shoulder before you speak.*"

Trust: distribution, open software, users own their data

Summary of Distributed Solutions

Common main objective: privacy-preserving services

Different types of decentralized architectures

- Three-tier architecture (Infomediary)
- Two-tier architecture (VRM)
- P2P (FreedomBox, Decentralized Social Networks)
- Hybrid architecture (Decentralized Social Networks, Personal Cloud-FreedomBox, Personal Data Store)

Built on common principles

- User-centricity and trust (transparency, security, control)

Critique of Decentralized Approaches

The Good: do not exhibit the intrinsic limitations of centralized solutions (privacy, security, etc…)

The Bad: yet, they've generally known little success (the privacy paradox)

… and the Challenging: raise important, but interesting challenges

Economic: viable business models compatible with privacy

Technical: design a secure Personal Data Server

1 - Secure storage of personal data (i.e., local requirements)

2 - Provide the same level of functionality, responsiveness and availability as a centralized solution (i.e., global requirements)

1. Secure storage with a Personal Data Server

Secure storage under user's control

- **Data must be made highly available, resilient to failure and protected against confidentiality and integrity attacks**
- **Cryptographic keys must be secured and only accessible by the user**
- **Accessing data from anywhere without privacy breaches**

Data integration/aggregation

- **Aggregate user's data in a single location: better usage, privacy, value**
- **Personal data is heterogeneous**
  - Structured/unstructured data, text, images, sound, video …
  - Records of transactions, clickstream data, bookmarks, bills, profiles, projects, preferences …
- **Data modeling, data integration, querying**

Privacy policy definition

- **Intuitive, simple ways for users to define access control rules**

Existing attempts of a Personal Data Server

- Many recent initiatives (Mydex, the Locker Project, Pixeom, Personal.com, data.fm, Qiy Foundation, …)
  - Personal data stores, personal data lockers/vaults, personal cloud

- Focus on secure storage and data aggregation
  - Managed locally by the user (The Locker Project) or outsourced to a trusted third party (Mydex, Personal.com)
  - Federate data from different sources (The Locker Project)

Weaknesses of exiting solutions

Important security breaches related to the data storage

- **Data is stored encrypted in the Cloud (Mydex, Personal.com)**
  - But the cryptographic keys are under the control of the service provider
- **Data is stored locally by the users on their personal computers (The Locker Project) or plug server (Pixeom, Freedombox)**
  - Raises several problems related to security, durability and availability

Many functionalities required to obtain a complete Personal Data Ecosystem are not provided

- **E.g., Global querying, anonymous data publishing, secure sharing, secure usage and accountability**

2. Required global functionalities of a Personal Data Server

Global querying

- **Personal data is essential to the development of societal related applications (smart cities, transport, energy, healthcare …)**
- **Transparently query many PDSs as with a centralized database**

Anonymous data publishing

- **PDS must allow users to anonymously participate in global treatments**

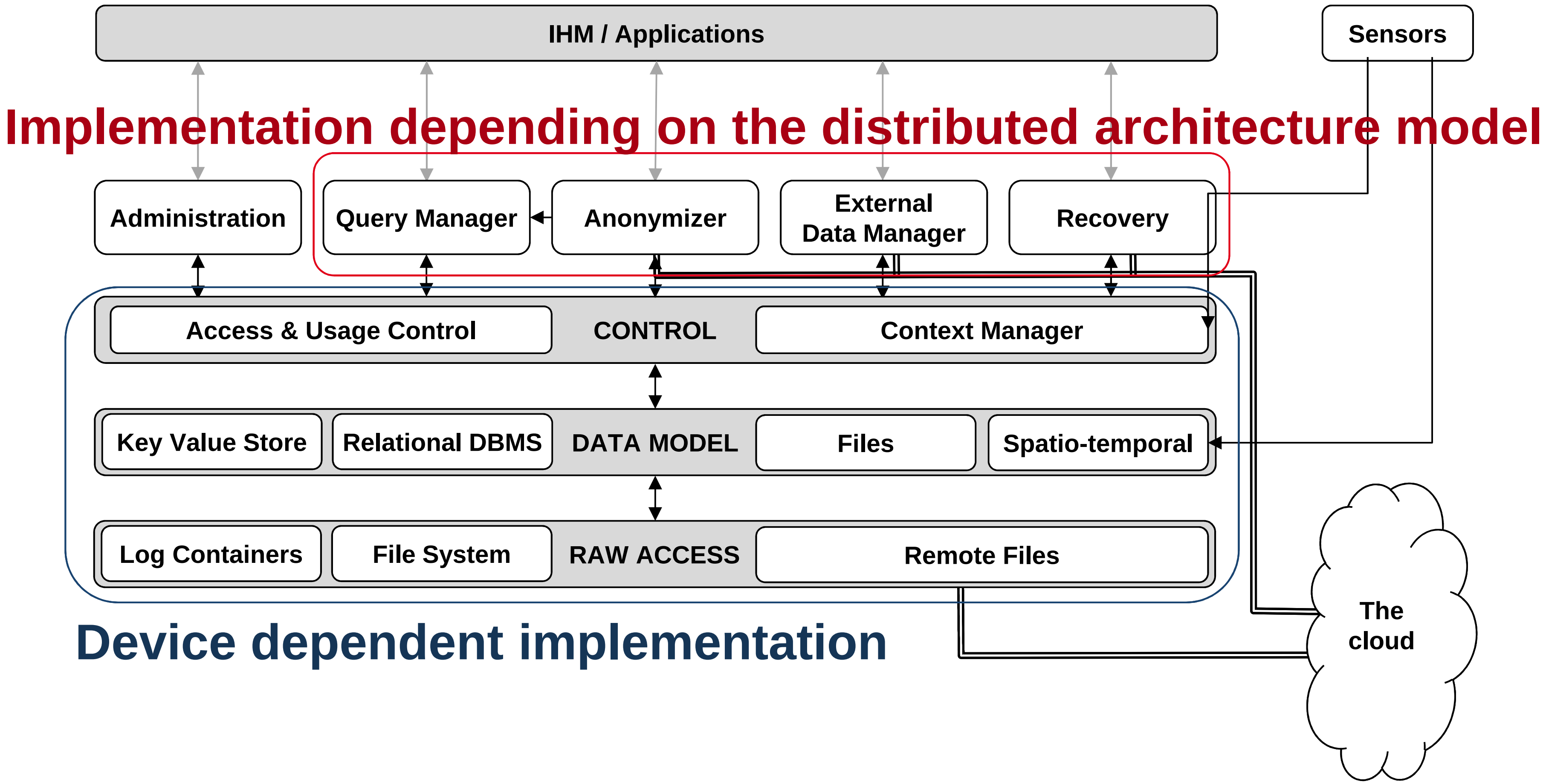
Distributed secure sharing

- **Users must get a proof of legitimacy for the credentials exposed by the participants of a data exchange**

Secure usage and accountability

- **Users must not loose control over their data through data sharing**
  - KuppingerCole, a security analyst company promotes ***Life Management Platforms*** "*a new approach for privacy-aware sharing of sensitive information, without the risk of loosing control of that information*"
- **Privacy principles must be enforced for the externalized data**

Personal Data Server: complete functional architecture



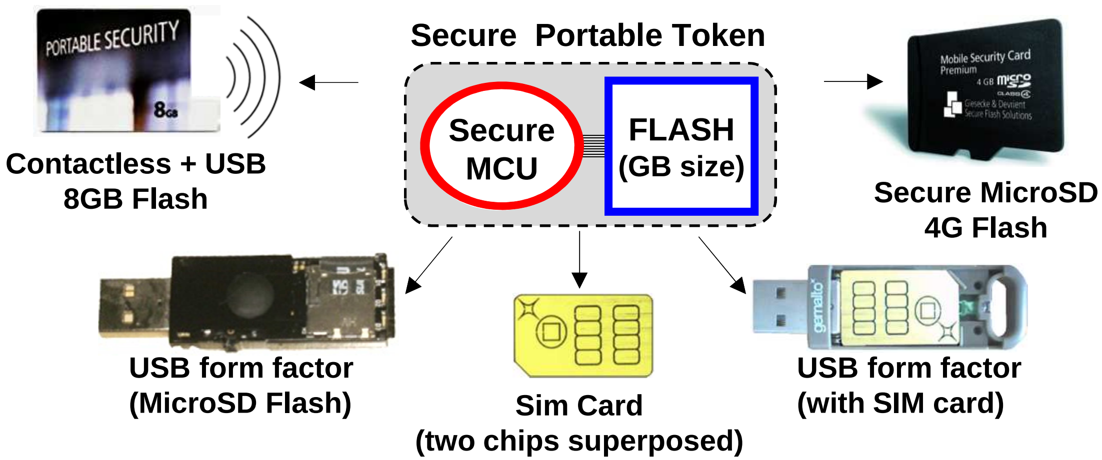
How to enforce the security of the PDS architecture

Advent of secure hardware at the edges of the Internet

- **Secure portable tokens: Secure MCU + Flash storage**

A sea change for personal data services

- **Offer privacy guarantees (>> Trust)**



Why trust personal secure HW solutions?

Users store their own data → minimize abusive usage

Self (user) managed platform → no DBA attack

Tamper-resistance + certified code/secure execution + single user
→ ratio cost/benefit of an attack is very high

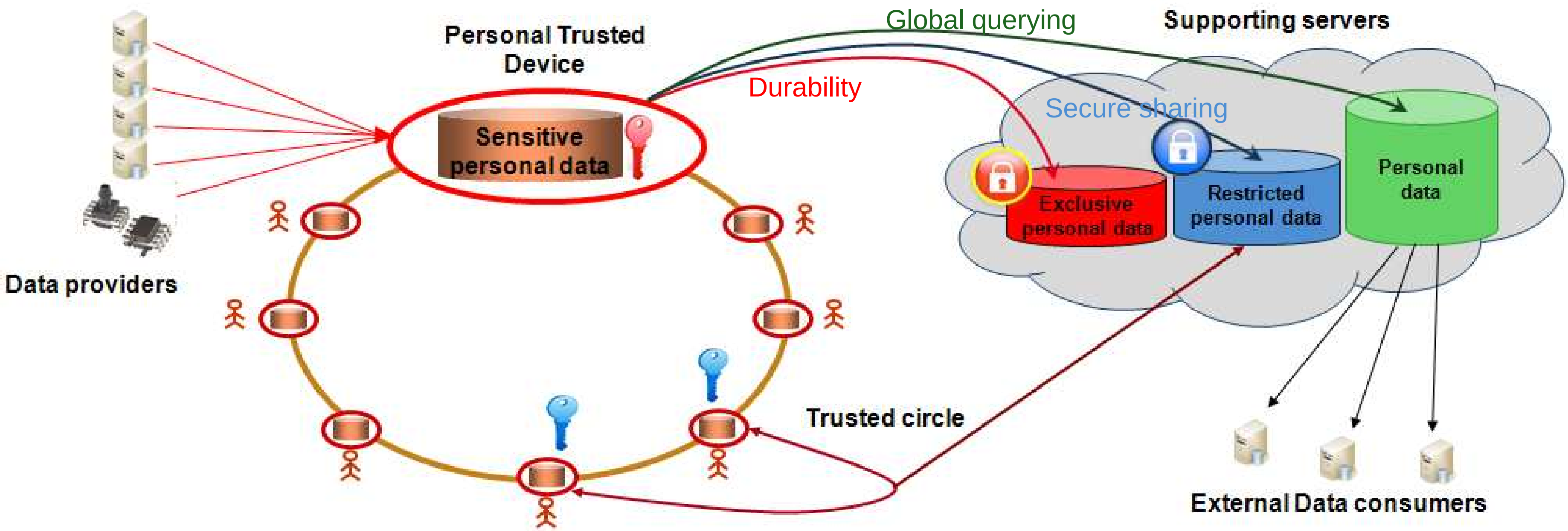
Enforce privacy principles for externalized (shared) data provided the recipient of the data is another PDS

Observation: a user does not have all the privileges over the data in her PDS

Global PDS Architectures: a spectrum of solutions

PDS asymmetric architecture

- **Built on Secure Portable Tokens**
- **Challenges**
  - Embedded data management (Part II of the tutorial)
  - Global querying (Part III of the tutorial)



LOW POWER & AVAILABILITY
HIGH TRUST

ASYMMETRIC

HIGH POWER & AVAILABILITY
LOW / NO TRUST

Present other configurations of global architectures in the Conclusion

PART II

Resource Constrained Data Management

… to regulate data sharing from the edge of the Internet

Resource constrained data management

Goal: manage personal data at the extremity of the Internet

- **Within sensors collecting data, in secure & personal user devices**
- **Potentially large data collections**
  - e-mails, medical records, official forms (admin., bank…), digital histories of interactions with e-services (Amazon, Telcos…) or physical systems (transport, smart homes, …)
- **Query functionalities must be embedded to compute authorized results**

Outline

- **Target hardware platforms**
- **Problem statement**
- **The general framework to solve the problem**
- **Representative proposals: search engine & SQL queries**

Target hardware

Sensors equipped with flash memory cards

Personal & secure devices

Personal memory devices in which a secure chip is implanted

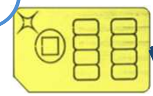
1 
USB MicroSD reader

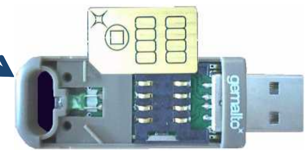
2 

Secure MicroSD 4GB Flash

3 
Contactless + USB 8GB Flash

Secure devices on which a GB flash chip is superposed

4 
Sim Card



Common architecture

Microcontroller

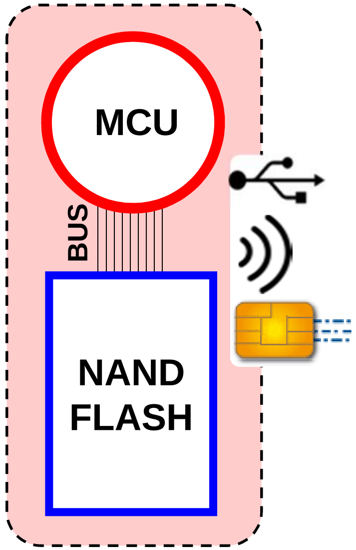
Low cost (sensors)

Tamper resistance [SC02]

- Miniaturization, protective layers (carrying signal),
- Multi-Layering (hide sensitive lines),
- Sensors (light/temp/power/freq.)
- ⇒ **prevent the chip from physical attacks**

GBs of memory

NAND FLASH (dense, robust, low cost)



Severe hardware constraints
… with a strong impact on data management

Microcontrollers

Small RAM (<128 KB) $\Longrightarrow$ Favor pipeline query evaluation

RAM is not dense $\Rightarrow$ (many) indexes

Security is linked with size

NAND FLASH

High cost of random writes $\Longrightarrow$ Data structures and strategies…

Pages are erased before write … must avoid random writes

Erase by Block vs. write by Page

How do existing techniques deal with these constraints ?

Existing Techniques

Light & embedded versions of DBMS products

- **e.g., SQLite, BerkeleyDB, DB2 Everyplace, …**
- **Target small but powerful devices (e.g., smart phones, set top boxes)**
- $\Rightarrow$ **Not compliant with very small RAM & not adapted to NAND Flash**

FLASH aware versions of traditional database indexes

- **BTree adaptation: BFTL [TECS07], LATree [VLDB09], FDTree [VLDB10]**
  - Store index updates in a **Flash resident log**, itself **indexed in RAM**
  - Updates are committed to the BTree in a batch mode (amortize write cost)
  - Small RAM $\Rightarrow$ Small index in RAM $\Rightarrow$ High commit frequency $\Rightarrow$ Low gains
- $\Rightarrow$ **Not compliant with very small RAM**

Existing Techniques (cont.)

Flash aware implementations of key-value stores

- **SkimpyStash [SIG11], LogBase [VLDB12], SILT [SOSP11]**
  - A log structure in FLASH is used to store the key-value pairs
  - An index is maintained in RAM to index that log (~1B per key-value pair)
- ⇒ **Incompatible with small RAM**

Data management techniques for MCUs

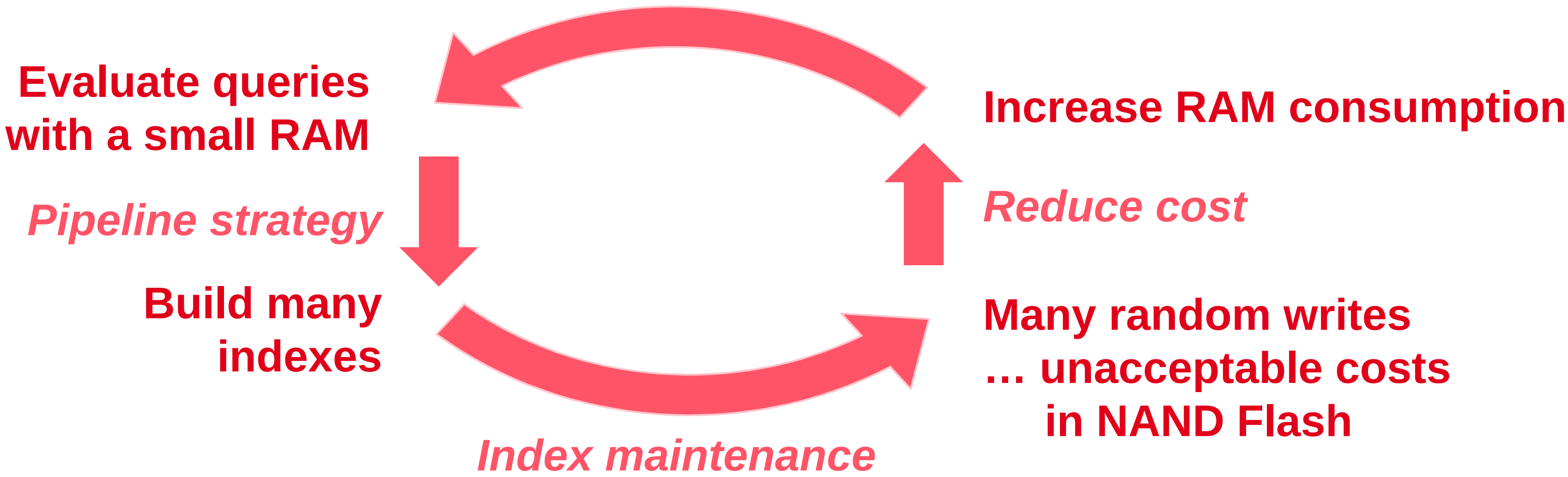
- **Proposals consider small amounts of (internal) memory**
  - PicoDBMS [VLDBJ01], VSDB [TOIS03], HybridStore [WSN13]
    - **Exploit byte writes accesses (EEPROM, NOR) specific to certain kinds of MCUs**
- **Recent proposals consider large Flash memory**

Details next

- **RDBMS: GhostDB [SIG07], PBFilter [IS12], MiloDB [DAPD14]**
- **Search engines: MAX [TSN08], Snoogle [TPDS10], Microsearch [TECS10]**

Problem statement

Problem : execute queries with a very small RAM
on large volumes of data stored in NAND FLASH



How do recent works resolve the problem ?

General (implicit) framework to solve the problem

1- Design index structures enabling pipeline query evaluation

2- Organize them into sequential structures (Logs)

Log structures satisfy Flash constraints

Pages are written sequentially (and never updated nor moved)

…. random write are avoided by construction

Allocation & de-allocation are made on large grains (Flash block basis)

…. partial garbage collection never occurs (avoids costly GC)

3- Provide scalability by reorganizing the Logs structures

Transform the sequential indexes into more efficient data structures

… the transformation itself must only use log structures

How do recent works implement this methodology?

First illustration: embedded search engines

Answer IR queries

For a set of query keywords, produce the *N* most relevant documents (according to a weight function like TF-IDF)

$$\text{TF-IDF(doc)} = \sum_{\{ki\}\ \text{query keywords}} \left(\text{weight}_{ti,doc} \times \text{Log}(|\{doc\}| \ / \ |\{\text{doc containing } t_i\}|) \right)$$

Inverted index

Stores triples (keyword, docid, weight)

Used at query time to retrieve all triples containing a query keyword

Search algorithm

The inverted index is accessed for each query keyword

In RAM: one container is allocated per retrieved docid… ⟵ *too much!*

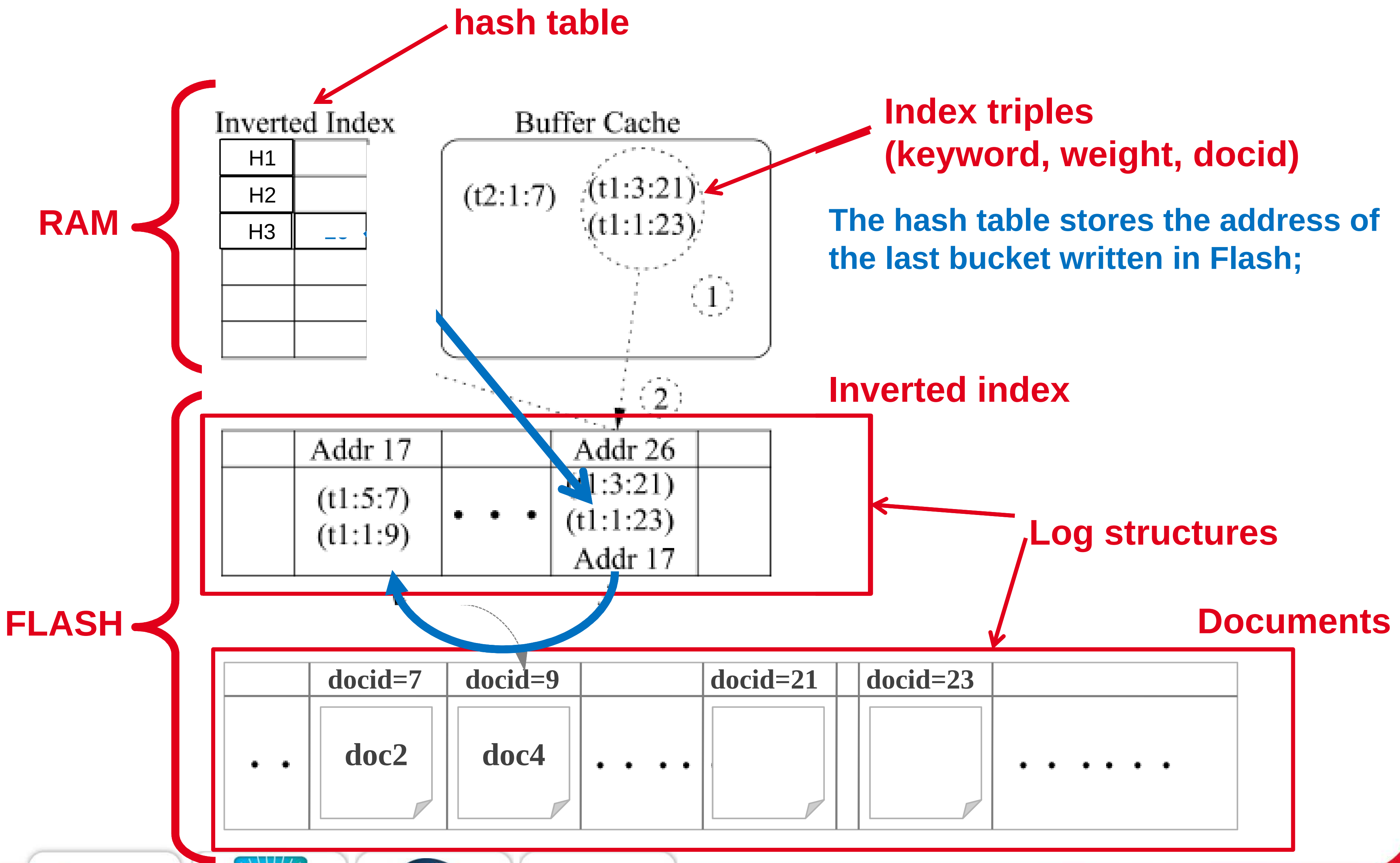
…used to aggregate the triples for one docid, and compute its TFIDF

The N documents with the highest scores are returned

How to store the index sequentially? How to search in pipeline?

Tan et al. [TECS10]

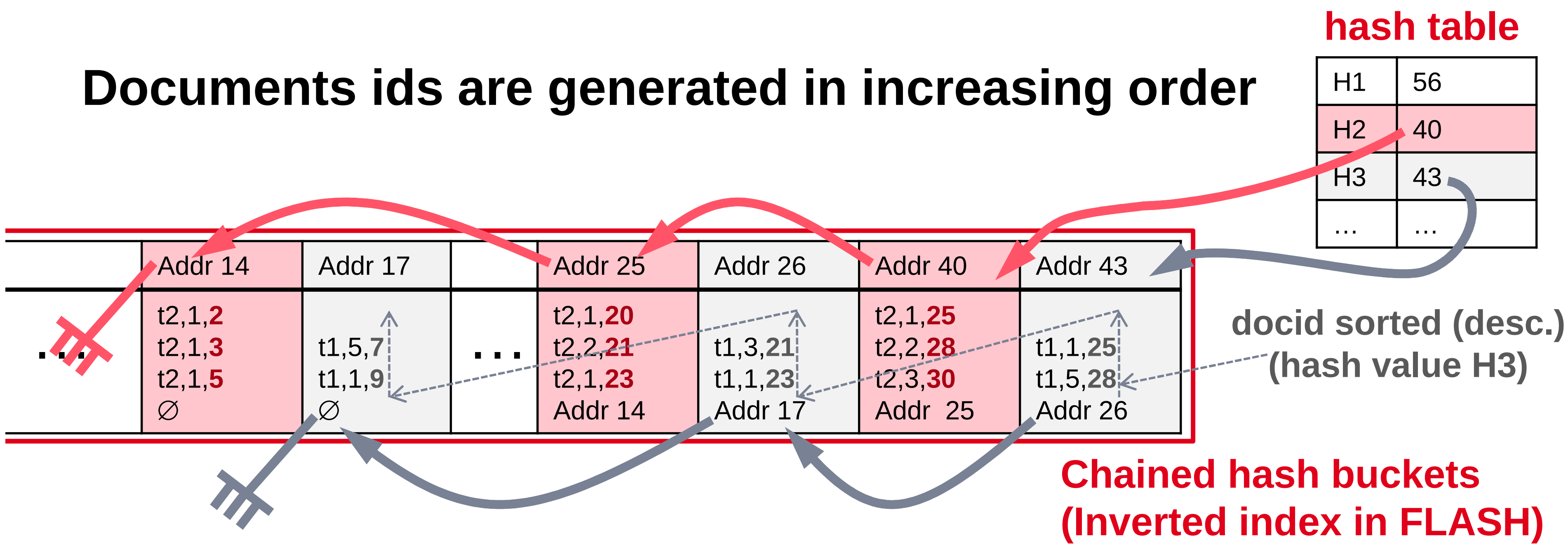
How to store the inverted index sequentially ?



Tan et al. [TECS10]

How to evaluate search queries in pipeline?

Documents ids are generated in increasing order



The query is computing in pipeline using a merge operation

Requires 1 page in RAM per hash list (per query keyword)

The triples are scanned, and “merged” on docids

$\Rightarrow$ Triples with an equal docid arrive in RAM at the same time…

… and the TF-IDF score of each docid can be computed in pipeline

The N docids with the highest score are kept in RAM

Second illustration: embedded relational database

SQL queries

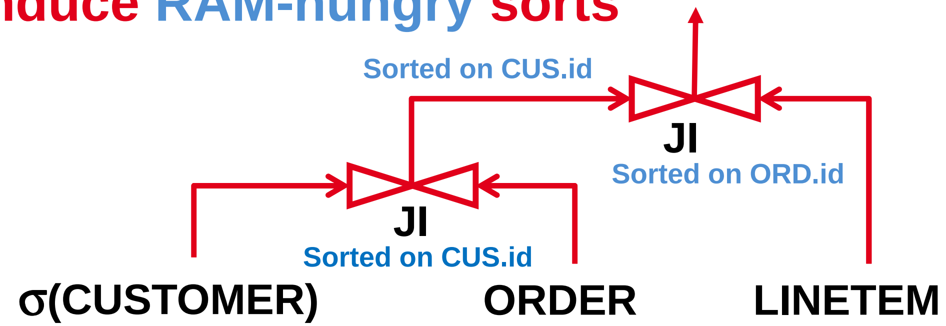
- Evaluate selections, projections, joins

Selection and join indexes

- Q1: How to store such indexes in log structures?
- Q2: How to make it scale?

Join algorithms consume lots of RAM

- Join indices could be a solution…
  - … but consecutive joins induce RAM-hungry sorts



- Q3: How to compute select-project-joins queries in pipeline?

How to build an index in log structures?

Log1: «Keys» (vertical partition)

Stores the index key, filled at tuple insertion

Log2: «Bloom Filters»

1 BF build for each page in «Keys»

BF is a probabilistic summary (~2B/key)

Retrieve CUSTOMER.CITY=‘Lyon’

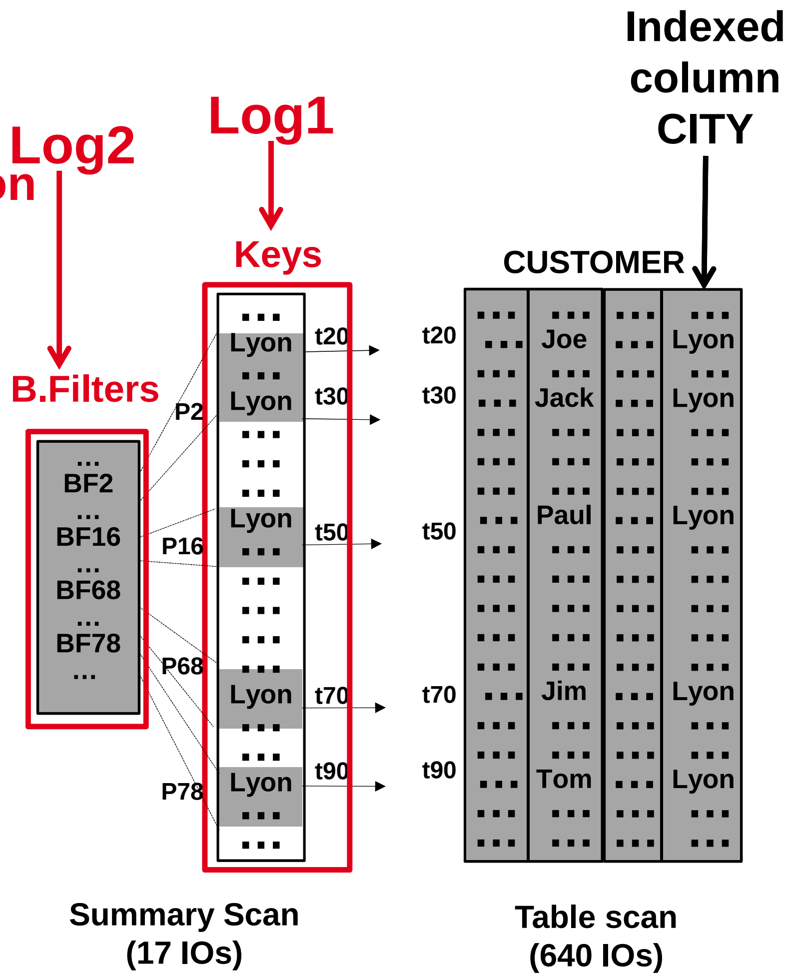
Scan of «Bloom Filters»

For each BF : if ‘Lyon’ ∈ BF

Negative ⇒ ignore it

Positive ⇒ access 1 page of «Keys»

search ‘Lyon’ & return tuples pointers

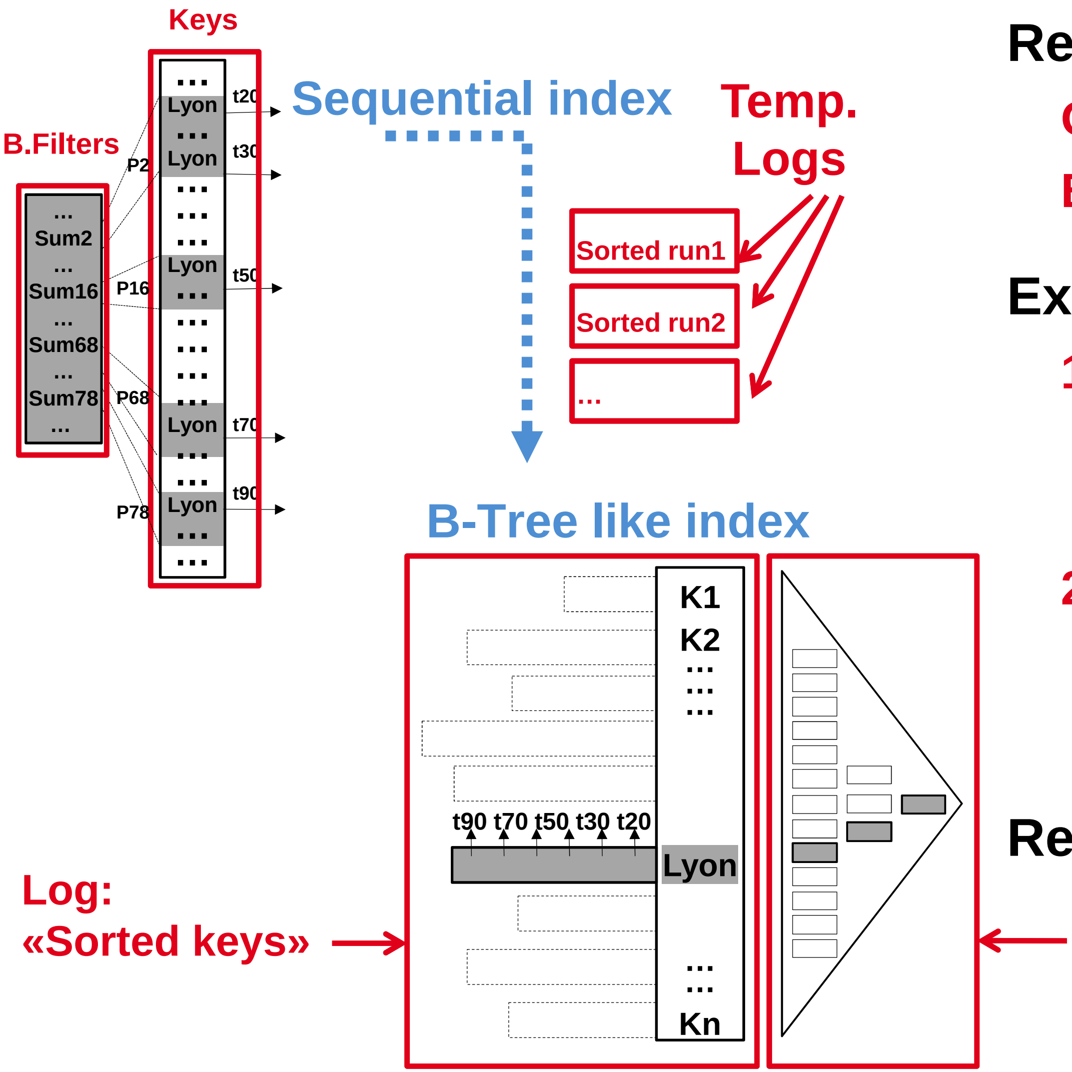


Efficient search: |Log2| I/O + 1 IO/result

… but how to achieve scalability?

Scalability ⇒ timely reorganize the index …to transform it into a more efficient index

[DAPD14]



Reorganization process:

- Only uses log structures
- Background / interruptible

Ex: Sequential index → B-Tree like

1) Sort the (key, pointer) pairs
   - → Temp. logs (sorted "runs")
   - → result written seq.: «Sorted Keys»
2) Build a key hierarchy
   - → No need of temporary Logs
   - → result is written seq.: «Tree»

Result: efficient B-Tree like index

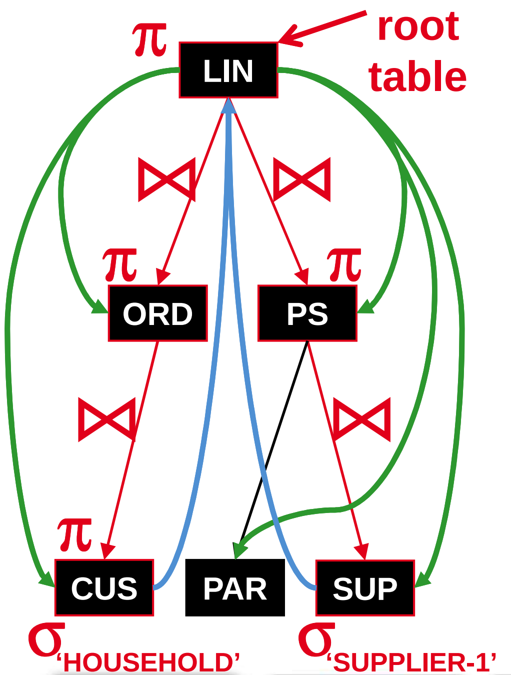
… how to evaluate SQL queries in pipeline?

[SIG07, DAPD14]

How to evaluate SQL queries in pipeline ?

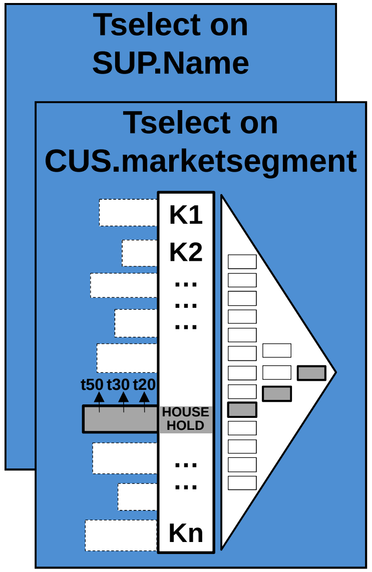
```
SELECT  CUS.*, ORD.*, LIN.*, PARTSUP.*
FROM    CUSTOMER CUS, ORDER ORD, LINETEM LIN, PARTSUP PS, SUPPLIER SUP
WHERE   CUS.CUSkey = ORD.CUSkey AND ORD.ORDkey = LIN.ORDkey AND
        LIN.PSkey = PS.PSkey AND PS.SUPkey = SUP.SUPkey AND
        CUS.Mktsegment = 'HOUSEHOLD' AND SUP.Name = 'SUPPLIER-1'
```

TPCD like schema



Tselect Indexes

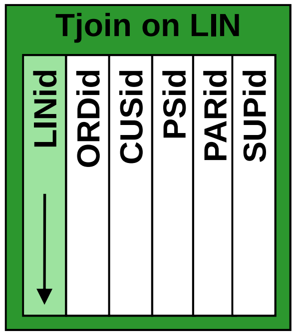
Each key of the index contains the rowids of the root table refering to that key



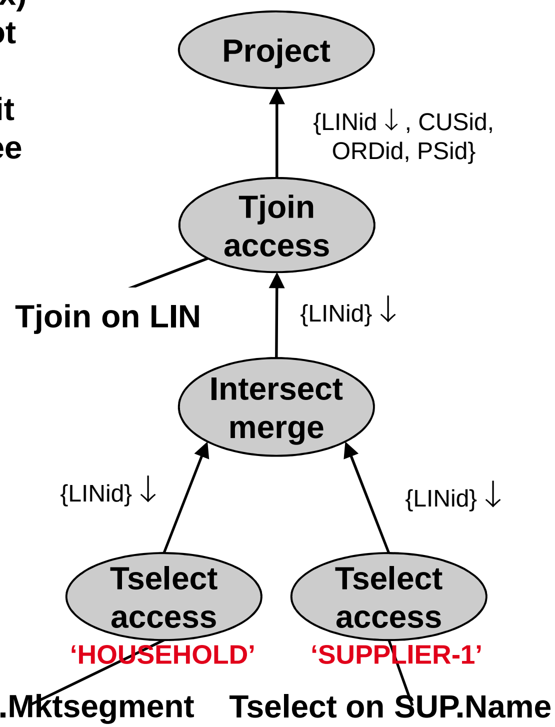
NB: Tselect returns sorted row ids!

Tjoin Index

(generalized join index) each rowid of the root table contains the rowids of the tuples it refers to in the subtree



Execution Plan



Conclusion

Encouraging results

- **Efficient search engines**
- **Efficient SQL queries**

Remaining challenges

- **Extend the principles to other data models**
  - XML, time series, spatial-temporal data, noSQL & key-value stores, etc.
- **A general co-design approach is still missing**
  - How to calibrate the HW (RAM) to data oriented treatments ?
  - How to adapt to dynamic variations of the HW parameters ?

PRiSM Lab. - UMR 8144

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

PART III : SECURE GLOBAL COMPUTATIONS

The example of Secure computation of Privacy Preserving Data Publishing Algorithms using Tokens

Secure Global Computation and SQL

PART III: OUTLINE

Problem Statement

Current Solutions to Secure Global Computation

- ***Generic Approach***
- ***Toolkits for Secure Computation***
- ***Using Trusted Hardware to Achieve Generic Computation***

Taking on SQL Aggregate Queries

Perspectives

Secure Global Computation on PDSs

PROBLEM STATEMENT:

How to perform global computations on the asymmetric architecture? (i.e. *using data from many/all PDSs*)

- SQL (aggregate) queries
- Privacy Preserving Data Publishing
- Data Mining
- …

The « classical » problem of Secure Global Computation (e.g., SMC) is more general and makes no trust assumption.

An overview to Secure Global Computations

Several approaches are possible to securely perform global computations:

1. Use only an untrusted server/cloud/P2P and use generic (and costly) algorithms. (e.g. Secure Multi-Party Computation [Yao82, GMW87, CKL06], fully homomorphic encryption [Gent09]) →**Problem = COST**

2. Use only an untrusted server/cloud/P2P and develop a specific algorithm for each specific class of queries or applications. (e.g. DataMining Toolkit [CKV+02]) →**Problem = GENERICITY**

3. ***Introduce a tangible element of trust, through the use of a trusted component and develop a generic methodology to execute any centralized algorithm in this context. ([Katz07, GIS+10, AAB+10]) → Problem = TRUST***

CURRENT SOLUTIONS TO SECURE GLOBAL QUERYING

Generic Secure Multi-Party Computation (SMC)

Truly Generic SMC is exponential in the number of inputs and therefore does not scale. See [Yao82, Yao86].

Other solutions such as [GMW87] do not provide specific generics to compute a solution (i.e. they need a zero-knowledge proof to work).

COST!

- Cost is unpractical : the resolution of the *millionaire problem* proposed in ’82 is proportional to the size of the values compared.
- Generalization to *n* different parties requires taking into account cheating (extra cost).
- [CKL06] have shown that in fact if there is not an honest majority, then only trivial functions can be computed.

There are (more or less) complicated cryptographic protocols.
Protocols are generic in the sense that they compute values of mathematical functions.
Protocols are *far too costly*.

Homomorphic Encryption Example

Homomorphic Encryption is a characteristic of several crypto-systems such as RSA, Paillier, ElGamal, etc.

Example : **Consider RSA. Given the RSA public key (e, m), the encryption of a message x is given by :**

E(p)=p^e mod m

The homomorphic property is :

$E(p_1) \times E(p_2) = p_1\hat{}e \times p_2\hat{}e \bmod m = (p_1 \times p_2)\hat{}e \bmod m = E(p_1 \times p_2)$

Fully **Homomorphic Encrytion means that *all ring* operators are homomorphic (this means + and x).**

Fully Homomorphic Encryption

Why is this a solution ?

- Any program with bounded input can be transformed into a Boolean circuit
- Any circuit can be transformed into a polynomial modulo 2
- Secure computation of a polynomial equates to securely computing any program
- To securely compute a polynomial, it is necessary and sufficient to securely compute + and x operations.

Definition :

We say that E is a fully homomorphic encryption from ({0,1}, +, x) to (D, ⊕, ⊗) if for all c_1, c_2 in D, such that $c_1=E(p_1)$ and $c_2=E(p_2)$

$$E^{-1}(c_1) \oplus E^{-1}(c_2) = p_1+p_2$$

$$E^{-1}(c_1) \otimes E^{-1}(c_2) = p_1 \times p_2$$

Or more generally $E^{-1}(f_D(c_1,\ldots,c_n)) = f_{\{0,1\}}(p_1,\ldots,p_n)$

COST!

A first result was proposed using ideal lattice cryptography in [Gent09], and has been a hot topic since.

The cost to have good security is (incredibly) high.

TOOLKITS FOR SECURE COMPUTATIONS

Data Mining Toolkit

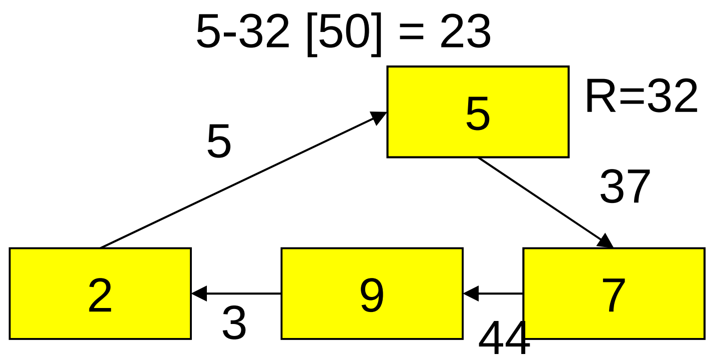
Toolkit for Data Mining : [CKV+02] Primitives :

- **Secure Sum,**
- **Secure Set Union,**
- **Secure Size of Set Intersection,**
- **Scalar Product.**

Can compute : Association Rules, Clusters. (Also : efficiency drops when some participants are dishonest).

Not usable for other applications

(such as SQL or PPDP)



Secure Sum Primitive

USING TRUSTED HARDWARE TO ACHIEVE GENERIC GLOBAL COMPUTATIONS

A new trend : SMC Using Tokens

The general idea when using Secure Hardware : Use *cheap secure hardware* to obtain substancial complexity class gains with SMC algorithms.

- Using tokens/smart-cards to improve the speed of computations [JKSS10]
- New foundations of SMC [Katz07, GIS+10]
- Limited to Secure Intersect (Oblivious Search) [HL08, FPS+11]

→ The primitives used are not « data intensive » primitives. Complex processing using tokens is a new topic !

→ These processes involve *initializing and sending* one or more smart cards. (SPTS would be an alternative).

→ Smart cards cannot compute everything themselves (this is *not* introducing a trusted third party)

COOL ☺ but PDSs are cooler !

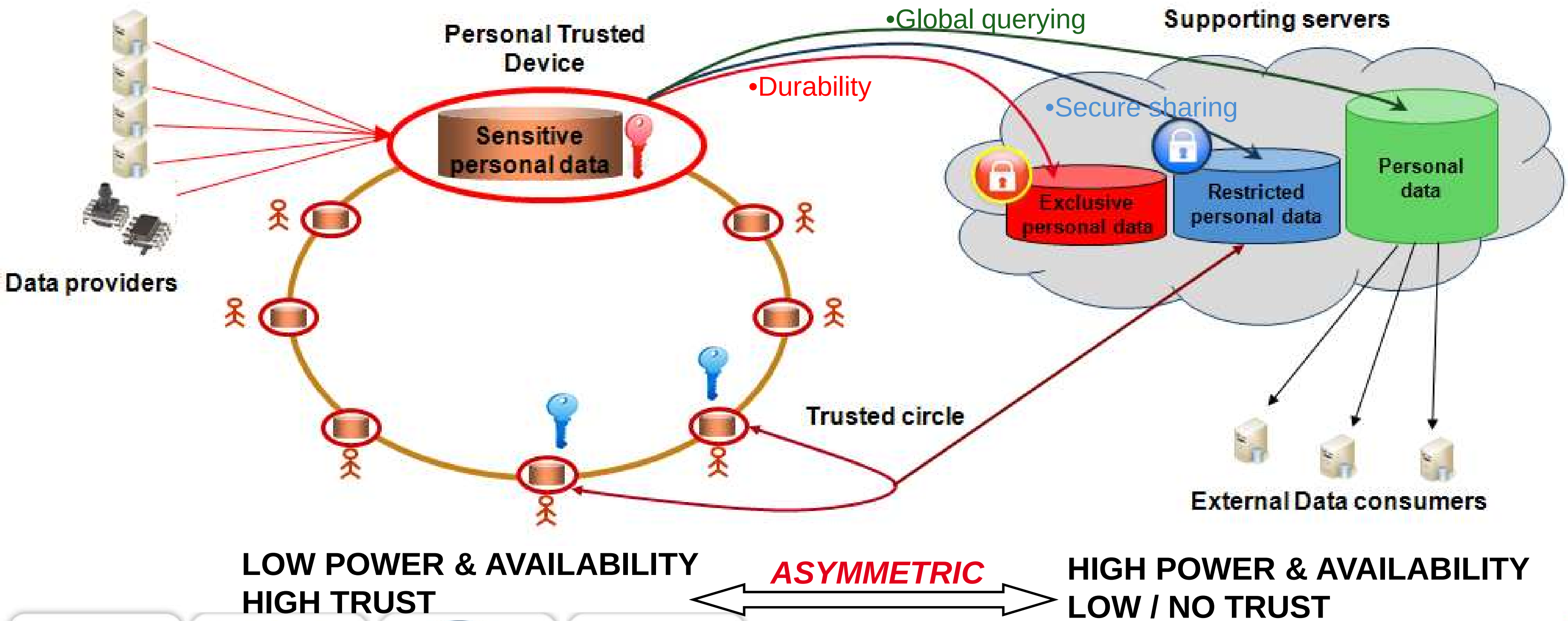
So, what's new ?

We have not one, but *many* elements of trust

- **Low powered, highly disconnected**
- **Trust between the elements, distributed computing is possible (*à la* cloud)**

Data is located *within* the elements of trust

- **Taking the device offline is a *physical* enforcement of AC**
- **Completeness of queries makes no sense**



EXAMPLE

Taking on SQL queries…

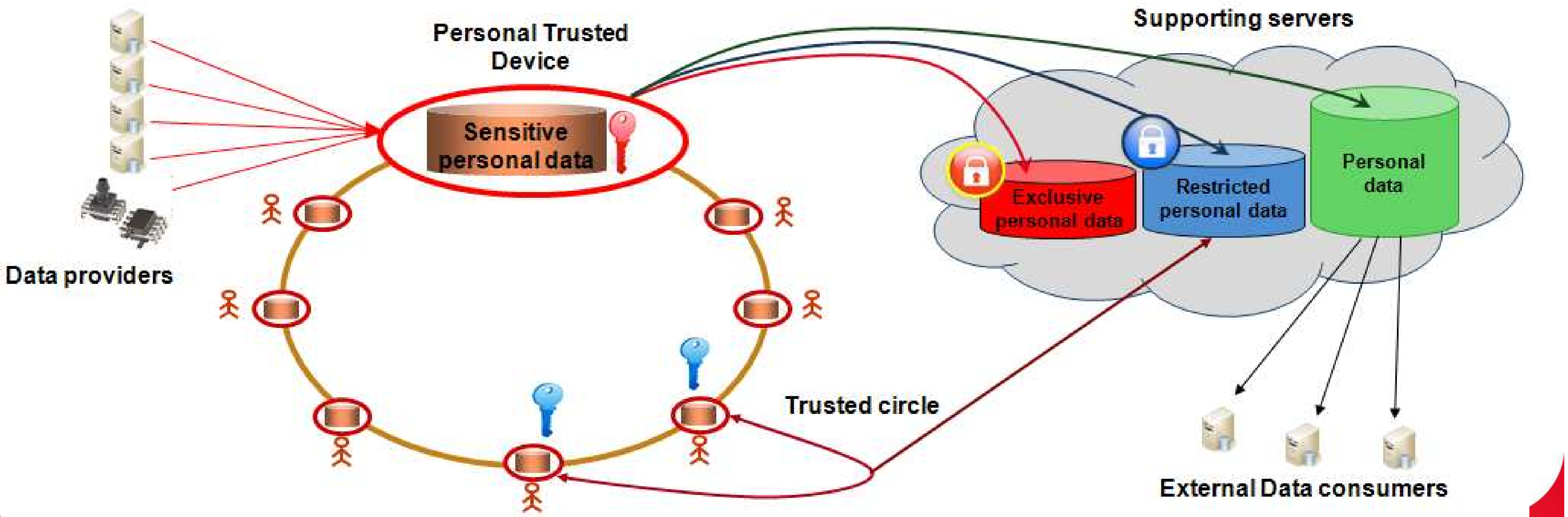
(or more generally aggregation operations)

…using Secure Portable Tokens

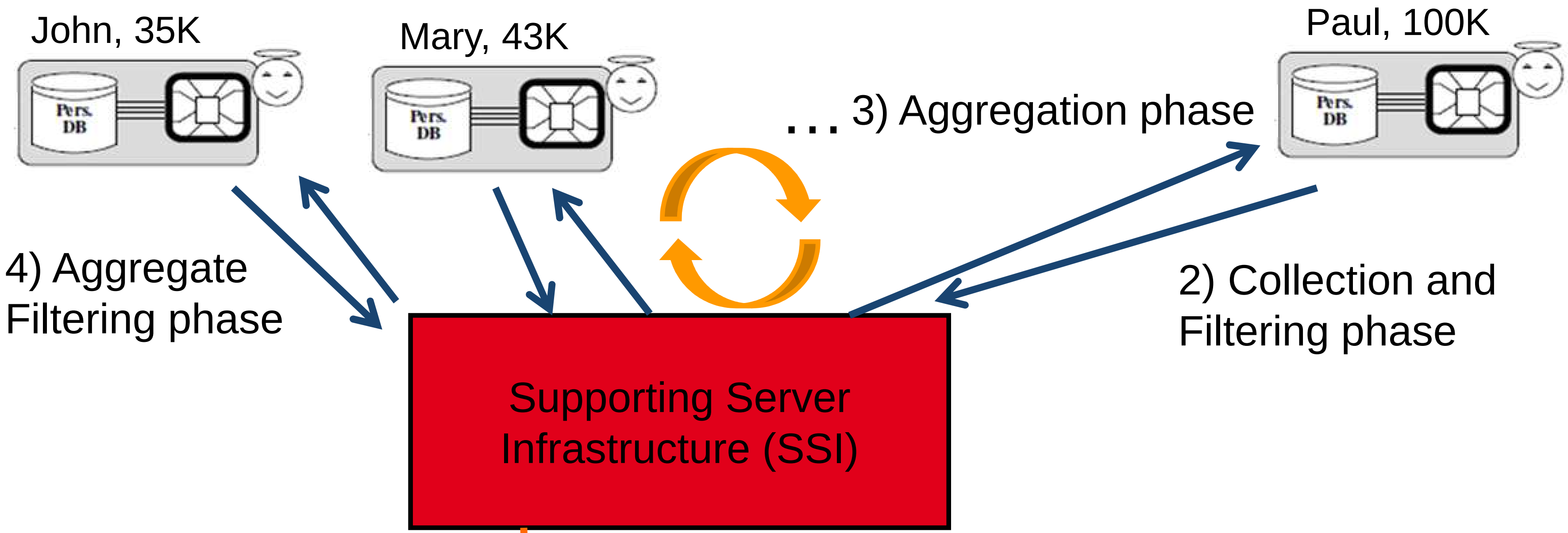
THREAT MODEL:

PDS can be :	**Infrastructure (SSI) can be :**
Unbreakable (honest)	Honest but curious (Semi-honest)
Broken (Weakly Malicious)	Weakly-Malicious (Covert Adversary = does not want to be detected)

A. HBC + Unbreakable → “simple protocols” presented here ([TNP14])
B. WM + Broken → Must be prevented ! (via **security primitives**) see [ANP13]



Solution Overview



SELECT age, AVG(salary)
FROM user
WHERE town = “Orsay”
GROUP BY age
HAVING MIN(salary) > 0
SIZE

1) Query

SELECT <attribute(s) and/or **aggregate function(s)**>
FROM <Table(s) / SPTs>
[WHERE <condition(s)>]
[GROUP BY <grouping attribute(s)>]
[HAVING <grouping condition(s)>]
[SIZE <size condition(s)>];

Stop condition: max #tuples or max time

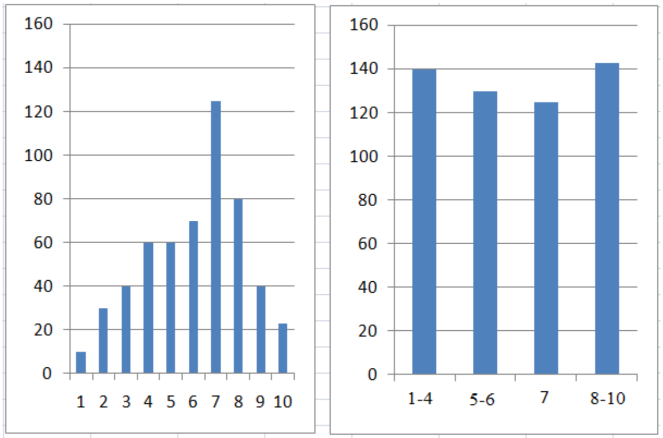
Proposed Solutions [TNP14] → *EDBT'14 Privacy Research Session 13 (Thursday 14h)*

Solutions vary depending on which kind of encryption is used, how the SSI constructs the partitions, and what information is revealed to the SSI.

- Secure aggregation solution (based on **non deterministic** encryption)

- Noise-based solutions (based on **deterministic encryption and fake tuples**)
  - **random (white) noise**
  - **noise controlled by the complementary domain**

- Histogram-based solutions (based on Hacigumus' **equidepth histogram** approach)



Conclusion of secure global computations with PDSs

What do we have now?

- Data mining toolkit [CKV+02]
- Generic protocol to solve SQL and SQL aggregate queries [TNP14] .
- This generic protocol can be used in many different contexts, such as Privacy Preserving Data Publishing [ANP13].
- These protocols support Honest-but-Curious and Malicious adversaries (detection and deterrence).

Are these solutions sufficient?

- Other types of queries (No-SQL) could also be supported
  - The difficult part will often be the aggregate part.
  - /!\ Graph based queries (private secure network queries) have an inherent difficulty because the security must be assured all along a path.

PRiSM Lab. - UMR 8144

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

PERSPECTIVES

Instances of alternative global architectures relying on secure hardware

Personal Social-Medical Folder (Field experiment)

- A personal folder available at home to ease care coordination
- Each patient owns her medical-social folder in a secure token
- The folder is archived (encrypted) on a central server
- Local and central copies are synchronized without Internet connection

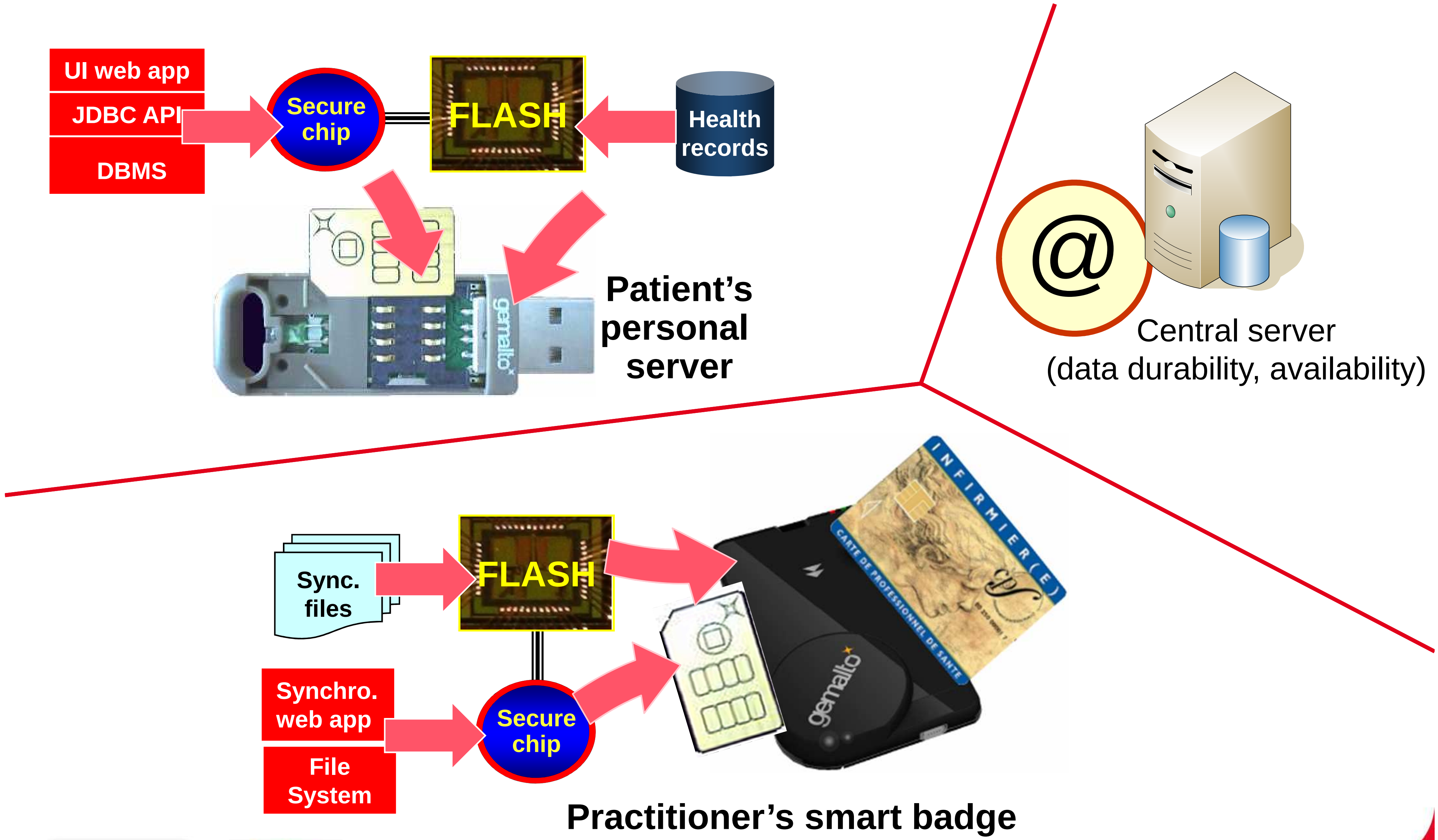
Folk-enabled Information Systems

- Enable personal data services in the Least Developed Countries
- No infrastructure required, a delay tolerant network is established

Trusted Cells

- Regulate personal data produced around an individual, at home
- Using the cloud as a storage service for encrypted data

Personal social-medical folder: architecture elements

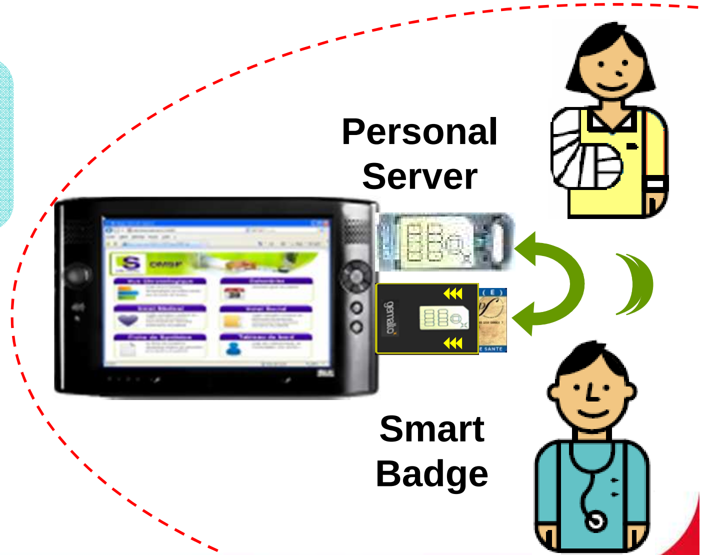


Availability at patient's home

EHR on a personal server

Access from a browser by patient's visitors (doctors & social workers, family…)

Disconnected access to Personal Servers (patient)



Care coordination between practitioners

EHRs on a central server

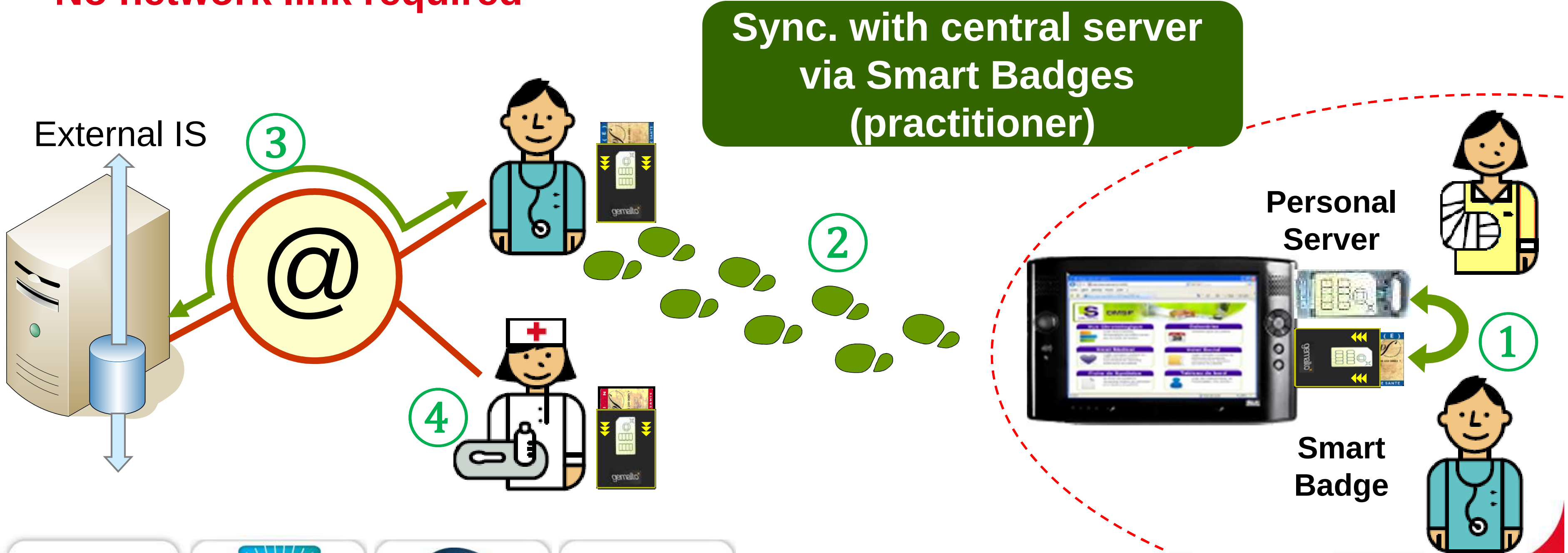
Web access & exchange

Sync. via Smart Badges

- No data re-entered
- No network link required

EHR on a personal server

Access from a browser by patient's visitors (doctors & social workers, family…)

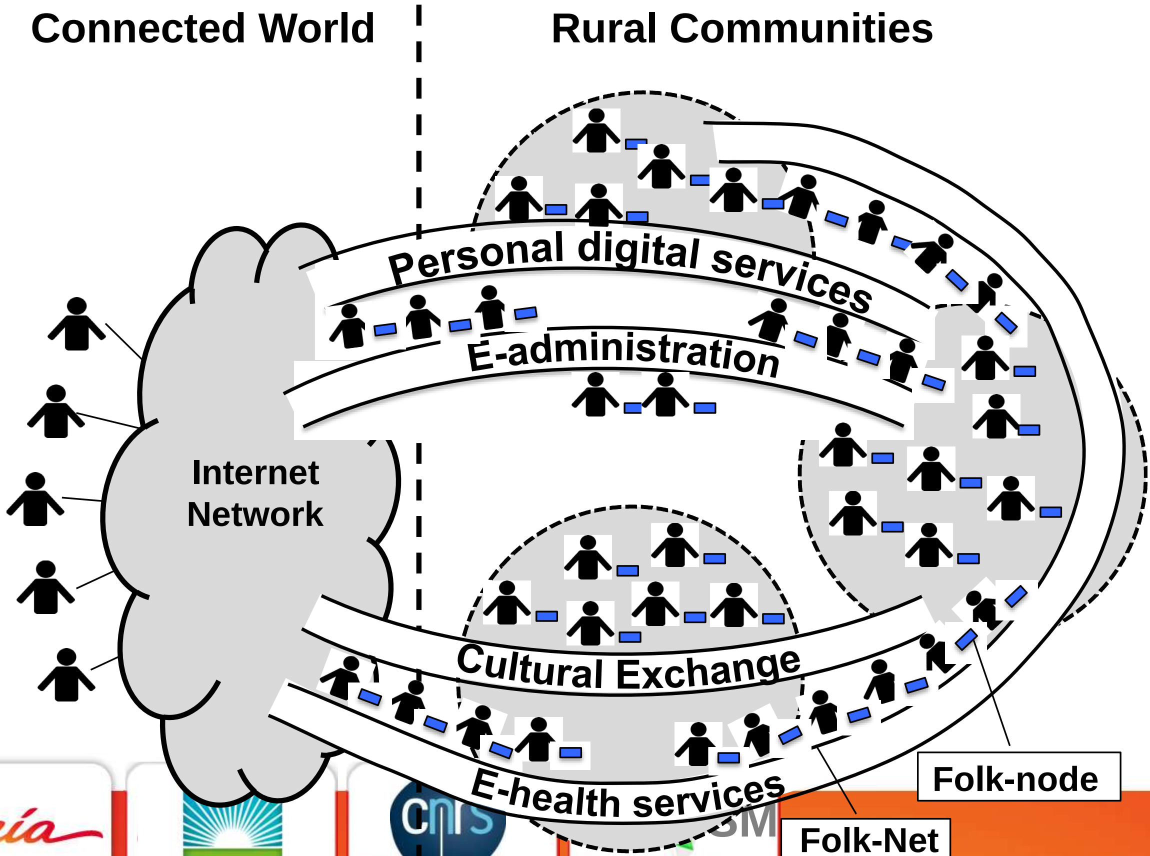


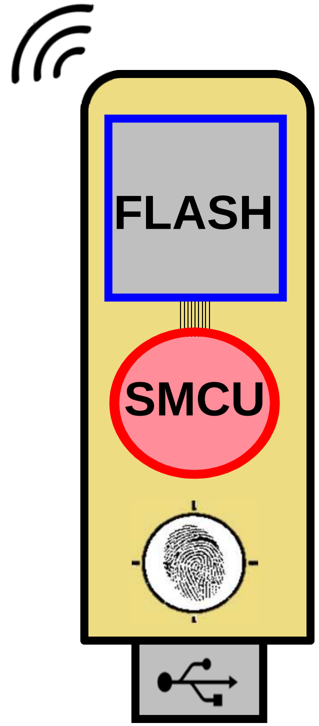
Folk-enabled Information Systems (Folk-IS)

1: Privacy: Lack of security infrastructure (coercive laws, secured servers, trusted authorities, …) leading to a self-enforcement of privacy principles

2: Self-sufficiency: must not rely on an hypothetic improvement of the existing software and hardware infrastructure

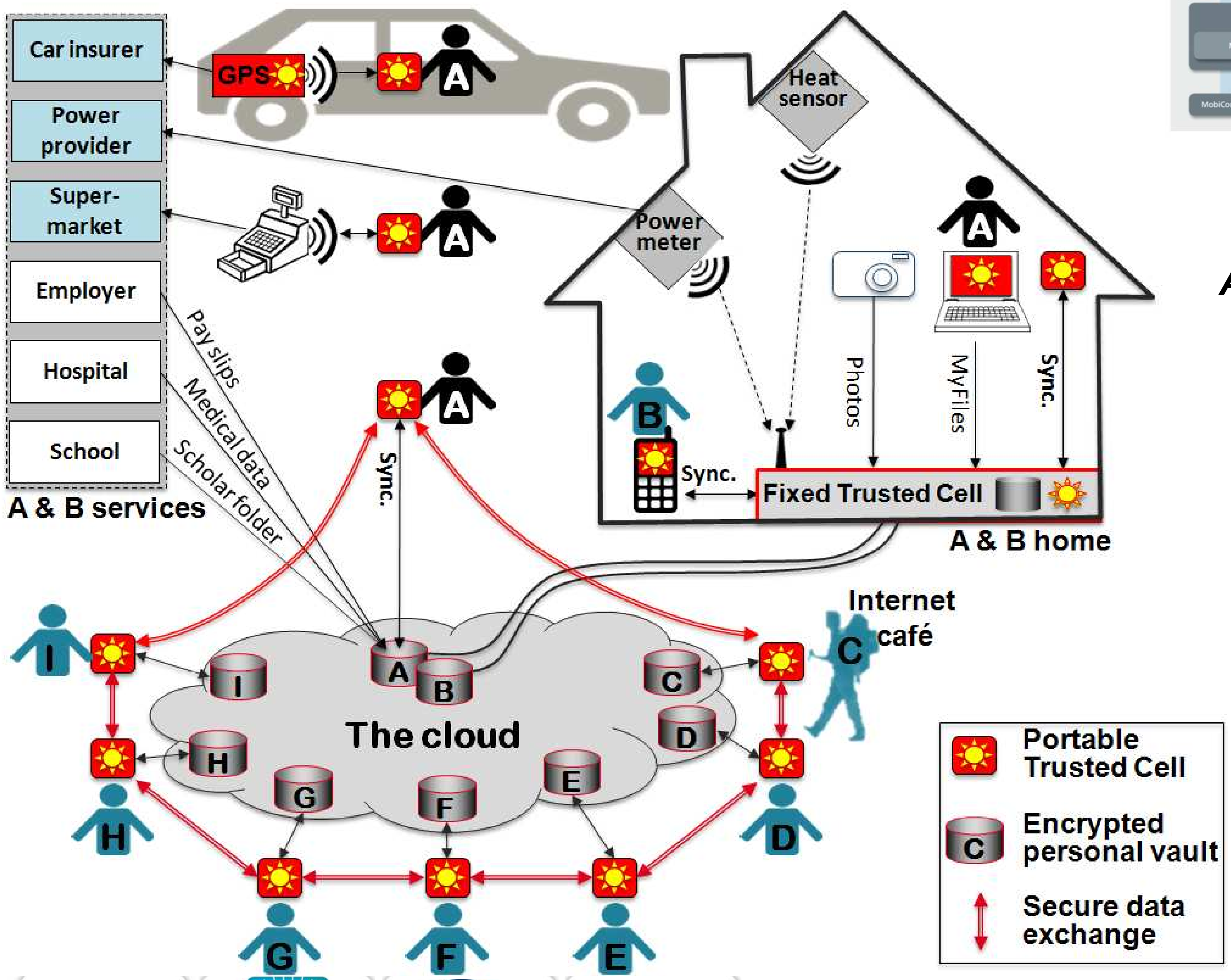
3: Very low and incremental deployment cost: the usual scale being a few dollars per user, without any large initial investments.

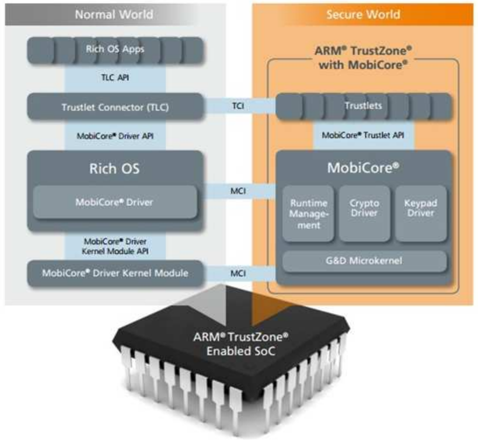




Trusted Cells Vision Architecture

(credit: Gi-De)





ARM Trust Zone

PRiSM Lab. - UMR 8144

THANK YOU

PRiSM Lab. - UMR 8144

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

REFERENCES

PART I: Distributed architecture (1/3)

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

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

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

J. Catlett. Panel on infomediaries and negotiated privacy techniques. In Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions, CFP '00, pages 155–156, New York, NY, USA, 2000

Mass-Educational Databases = Wrong Architecture, www.identitywoman.net/mass-educational-databases-wrong-architecture

VRM project, http://blogs.law.harvard.edu/vrm/projects/

A. Mitchell, I. Henderson, and D. Searls. Reinventing direct marketing — with vrm inside. Journal of Direct Data and Digital Marketing Practice, 10(1):3–15, 2008

FreedomBox: http://freedomboxfoundation.org/

Wikipedia. Freedombox, Vendor Relationship Management, Distributed Social Networks

PART I: Distributed architecture (2/3)

L. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. IEEE Communications Magazine, 47(12):94–101, 2009

L. M. Aiello and G. Ruffo. Lotusnet: tunable privacy for distributed online social network services. Computer Communications, In Press, 2010

I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. Protecting free with freenet. Internet Computing IEEE, 6(February):40–49, 2002

Diaspora*, https://joindiaspora.com/

R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An online social network with user-defined privacy. Computer, 39(4):135–146, 2009

S. Buchegger, D. Schioberg, L. H. Vu, and A. Datta. PeerSoN: P2P Social Networking - Early Experiences and Insights. In Proceedings of the Second ACM Workshop on Social Network Systems Social Network Systems 2009, co-located with Eurosys 2009, Nurnberg, Germany, March 31 2009

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

M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan. Personal Data Vaults: a locus of control for personal data streams. 2010

PART I: Distributed architecture (3/3)

Mydex, http://mydex.org/

Mydex. The case for personal information empowerment : The rise of the personal data store, 2010

The Locker Project, http://lockerproject.org/

Qiy Foundation, www.qiyfoundation.org/

Personal, www.personal.com

KuppingerCole, http://www.kuppingercole.com/report/advisorylifemanagementplatforms7060813412

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

Giesecke & Devrient, "Portable Security Token", http://www.gd-sfs.com/portable-security-token

Eurosmart. Smart USB token. White paper, Eurosmart, 2008, (10p)

ARM-TrustZone, http://www.arm.com/products/processors/technologies/trustzone.php

N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, I. Sandu Popa, P. Pucheral. Trusted Cells: A Sea Change for Personnal Data Services, in "6th Biennal Conference on Innovative Database Research (CIDR)", Asilomar, États-Unis, 2013

PR SM

PART II: Resource constrained data management (1/4)

Smart card security

[SC02] Witteman, M. (2002). Advances in smartcard security. Information Security Bulletin, 7(2002), 11-22.

Flash aware indexes

[TECS07] Wu, C. H., Kuo, T. W., & Chang, L. P. (2007). An efficient B-tree layer implementation for flash-memory storage systems. ACM Transactions on Embedded Computing Systems (TECS), 6(3), 19.

[VLDB09] Agrawal, D., Ganesan, D., Sitaraman, R., Diao, Y., & Singh, S. (2009). Lazy-adaptive tree: An optimized index structure for flash devices. Proceedings of the VLDB Endowment, 2(1), 361-372.

[VLDB10] Li, Y., He, B., Yang, R. J., Luo, Q., & Yi, K. (2010). Tree indexing on solid state drives. Proceedings of the VLDB Endowment, 3(1-2), 1195-1206.

PART II: Resource constrained data management (2/4)

Flash aware key-value stores

[SIG11] Debnath, B., Sengupta, S., & Li, J. (2011, June). SkimpyStash: RAM space skimpy key-value store on flash-based storage. In Proceedings of the 2011 international conference on Management of data (pp. 25-36). ACM.

[VLDB12] 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), 1004-1015.

[SOSP11] Lim, H., Fan, B., Andersen, D. G., & Kaminsky, M. (2011, October). SILT: A memory-efficient, high-performance key-value store. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 1-13). ACM.

PART II: Resource constrained data management (3/4)

DBMS on-chip

[VLDBJ01] Pucheral, P., Bouganim, L., Valduriez, P., & Bobineau, C. (2001). PicoDBMS: Scaling down database techniques for the smartcard. The VLDB Journal, 10(2-3), 120-132.

[TOIS03] Bolchini, C., Salice, F., Schreiber, F. A., & Tanca, L. (2003). Logical and physical design issues for smart card databases. ACM Transactions on Information Systems (TOIS), 21(3), 254-285.

[SIG07] Anciaux, N., Benzine, M., Bouganim, L., Pucheral, P., & Shasha, D. (2007, June). GhostDB: querying visible and hidden data without leaks. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (pp. 677-688). ACM.

[IS12] Yin, S., & Pucheral, P. (2012). PBFilter: A flash-based indexing scheme for embedded systems. Information Systems.

PART II: Resource constrained data management (4/4)

DBMS on-chip (cont.)

[DAPD14] Anciaux, N., Bouganim, L., Pucheral, P., Guo, Y., Le Folgoc, L., & Yin, S. (2013). MILo-DB: a personal, secure and portable database machine. Distributed and Parallel Databases, 1-27.

Search engines on-chip

[TSN08] Yap, K. K., Srinivasan, V., & Motani, M. (2008). Max: Wide area human-centric search of the physical world. ACM Transactions on Sensor Networks (TOSN), 4(4), 26.

[TPDS10] Wang, H., Tan, C. C., & Li, Q. (2010). Snoogle: A search engine for pervasive environments. Parallel and Distributed Systems, IEEE Transactions on, 21(8), 1188-1202.

[TECS10] Tan, C. C., Sheng, B., Wang, H., & Li, Q. (2010). Microsearch: A search engine for embedded devices used in pervasive computing. ACM Transactions on Embedded Computing Systems (TECS), 9(4).

PART III: references (uncomplete)

[ANP13] Allard, T., Nguyen, N., Pucheral, P.: MetaP: Revisiting Privacy-Preserving Data Publishing using Secure Devices, in DAPD, 55p, to appear.

[CKV+02] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.Y.: Tools for privacy preserving distributed data mining. SIGKDD Explor. Newsl., vol. 4, pages 28-34, ACM, New York, NY, USA, (2002)

[FPS+11] Fischlin, M., Pinkas, B., Sadeghi, A-R., Schneider, T., Visconti, I.: Secure set intersection with untrusted hardware tokens. In CT-RSA, (2011).

[Gent09] Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In STOC, (2009)

[GIS+10] Goyal, V., Ishai, Y., Sahai, A., Venkatesan R., Wadia, A.: Founding Cryptography on Tamper-Proof Hardware Tokens. Theory of Cryptography, pp 308-326, (2010)

[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play ANY mental game. In ACM STOC, pp 218-229, New York, NY, USA, (1987)

[HILM02] Hacigumus, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL over encrypted data in database service provider model. ACM SIGMOD, pp. 216-227. Wisconsin (2002)

[HIM04] Hacigumus, H., Iyer, B. R., Mehrotra, S.: Efficient execution of aggregation queries over encrypted relational databases. DASFAA, pp. 125-136. Korea (2004)

[HL08] Hazay, C., Lindell, Y.: Constructions of truly practical secure protocols using standard smartcards. In ACM CCS, New York, NY, USA (2008)

PART III: references

[JKSS10] Jarvinen, K., Kolesnikov, V., Sadeghi A-R., Schneider, T.: Embedded SFE:Offloading Server and Net-work Using Hardware Tokens. In Financial Cryptography and Data Security (2010)

[Katz07] Katz, J.:Universally Composable Multi-party Computation Using Tamper-Proof Hardware. In Advances in Cryptology, EUROCRYPT '07, pp 115-128, (2007)

[Yao82] Yao, A.C.: Protocols for secure computations. In Annual Symposium on Foundations of Computer Science, FOCS, pp 160-164, Washington, DC, USA, (1982)

[Yao86] Yao, A.C.: How to generate and exchange secrets. In Annual Symposium on Foundations of Computer Science, FOCS, pp 162-167, Washington, DC, USA, (1986)