Tutorial: Trusted Execution Environments and Intel SGX - a few basic notions and usages

Antoine Boutet, Iulian Sandu Popa

RESSI

Neuvy-sur-Barangeon, 10 mai 2023

Antoine Boutet (PRIVATICS@Inria Rhone-Alpes)
Iulian Sandu Popa (PETRUS@Inria de Saclay/UVSQ)

Trusted Execution Environments and Intel SGX

a few basic notions and usages

PEPR Cybersécurité – IPoP

FRANCE 2030
STRATEGIE NATIONALE CYBERSECURITE
PROGRAMME ET EQUIPEMENTS PRIORITAIRES DE RECHERCHE CYBERSECURITE
anr

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

From Secure Elements to Trusted Execution Environments (TEEs)

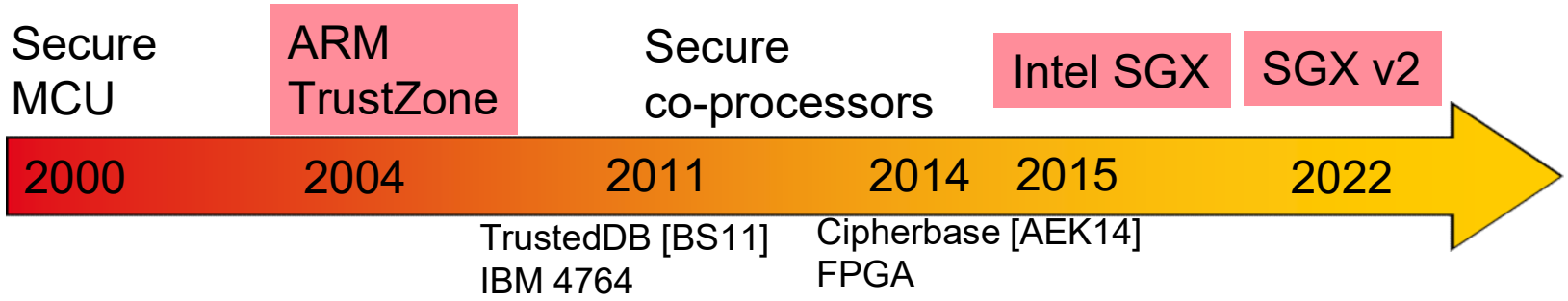
From secure elements, TPM, HSM, etc. [ANS14]

Smart cards or TPM (in smartphones, PCs, home boxes)

… to: Trusted Execution Environments (TEEs) [ABP+19]

Specialized HW: ARM TrustZone, Intel SGX, AMD platform security, etc.

Everywhere : Smartphones & PCs



Promise: HW level isolation and attestation

Isolation:

- Code executed within a TEE safe from external observation/tampering (OS, user)

Attestation:

- Ability to give a certificate that result produced by a specific piece of code running within TEE

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

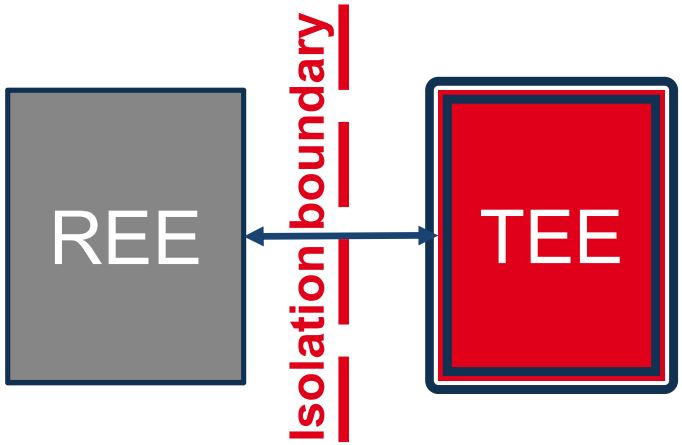
- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Secure data management in TEEs

Common general architecture (for existing basic TEEs, secure co-CPUs/FPGAs, recent TEEs-Intel SGX): trusted vs. untrusted memory space



What to look for in details?

- HW architecture: inherent limitations of the HW (e.g., SCPU clock, size of the secure RAM, bandwidth between secure/unsecure worlds…)
- SW architecture: which modules run inside the secure HW => Objective: minimize the Trusted Computing Base (TCB) vs. efficiency (REE/TEE context switching)
- Security guarantees: access pattern leak vs. oblivious query processing

Adversary: untrusted, curious and controls the system

- Assumption: TEE isolation cannot be bypassed by an attacker controlling the system

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

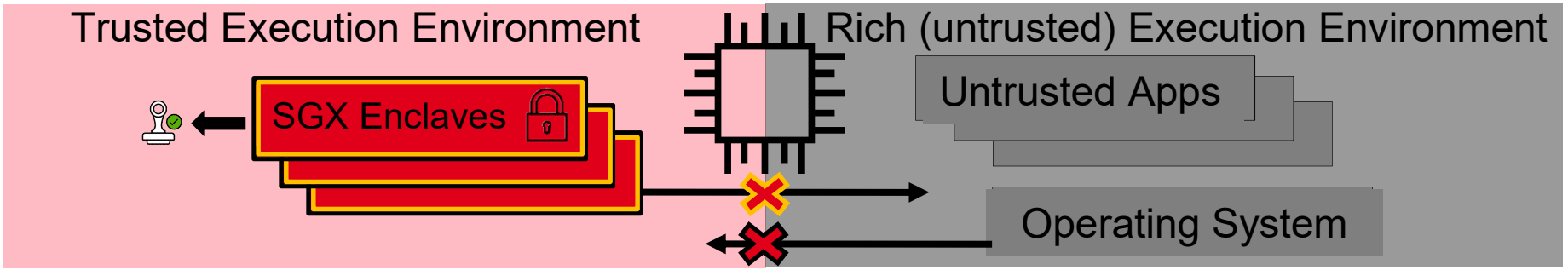
PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Intel Software Guard eXtensions (SGX)



- **Enclaving**
- **Attestation**
- **Confidentiality**

Insured by SGX

- + Recent works with **Sandboxing**

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Efficient data processing with TEEs

Modern HW, e.g. Intel SGX, democratize the access to trusted execution technologies

- Main CPU chip offers TEE capabilities through enclaves (special CPU mode enabled via new instructions) => ubiquitous access to TEE and strong (HW) integration between REE/TEE
- Yet, performance considerations remain critical for minimizing the enclave related overheads

Main overhead sources with SGX enclaves [WAK18] [PVC18]

- Memory encryption and integrity checking: unavoidable but low overhead
- Enclave transitions (ECALL/OCALL): high overhead
- Enclave paging (related to a limited enclave size): high overhead

It requires carefully redesigning (data-oriented) apps

Efficient data processing with TEEs – EnclaveDB [PVC18]

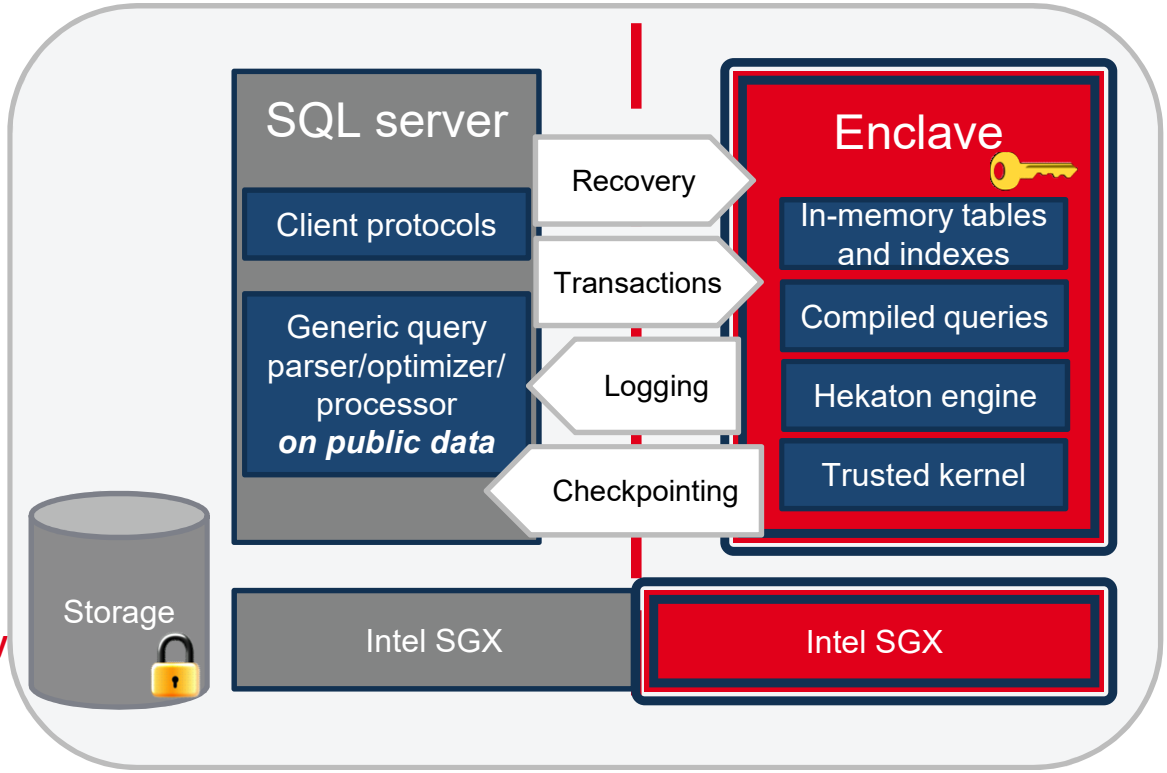
High performance DB engine…
with security using Intel SGX

Important assumption: <u>all sensitive data loaded in enclave memory</u>

- No need for expensive SW encryption/integrity checks
- In-memory enclave data minimizes the leakage of sensitive information
- Also minimizes the number of costly IN/OUT enclave transitions
- Smaller TCB (Heckaton engine) using precompiled procedures

Focus on secure and efficient DB logging and recovery

- Efficient protocol for checking integrity and freshness of the DB log
- Low overhead (~40%) compared with classical industry in-memory DBs



Efficient data processing with TEEs – Indexing/KVS

HardIDX [FBB+18]: secure and efficient B-tree indexing using SGX

- Leverage SGX enclaves to secure outsourced data searches while maintaining high query performance
- Several order of magnitude lower query processing time than with traditional compared with the best known searchable encryption schemes…
- … with similar level of confidentiality protection

eLSM [TCL+19]: authenticated KVS with TEE enclaves

- Focuses on optimizing update-oriented workloads…
- … and ensuring query authenticity: integrity, completeness and freshness
- Modifies the classical LSM-tree to cope with SGX enclave constraints

Both HardIDX and eLSM leak the access patterns

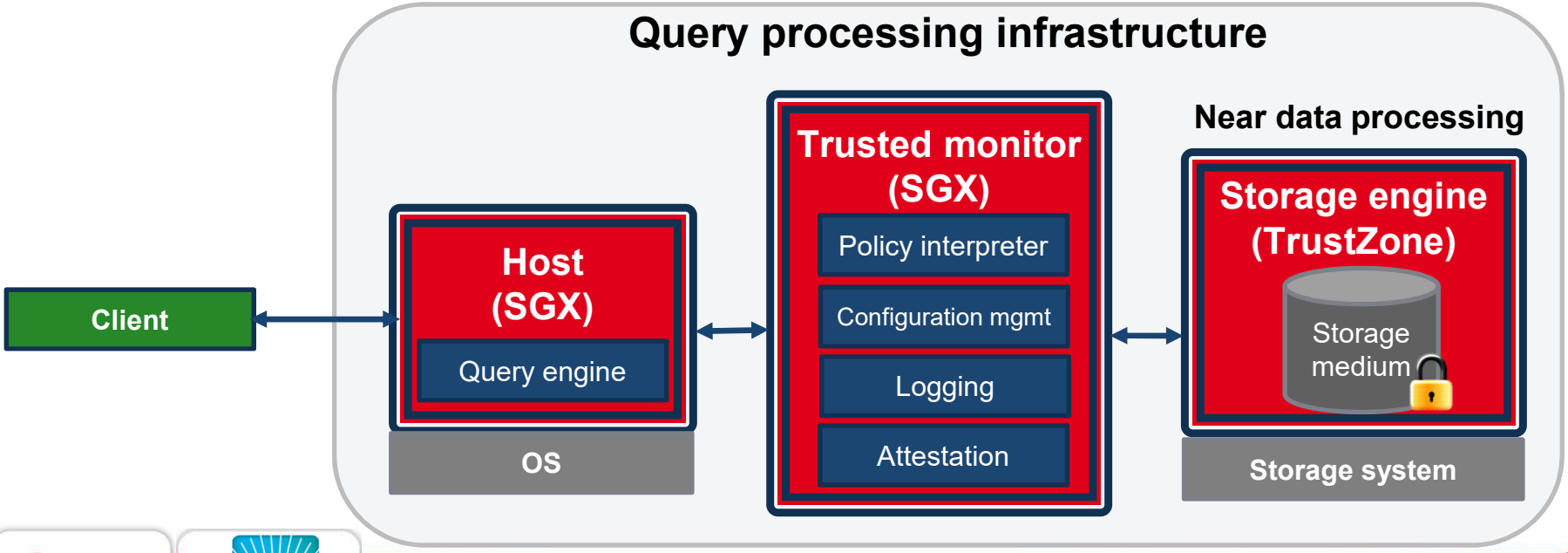
IronSafe [UCB+22]

IronSafe computing storage architecture (CSA)

- Efficient data-oriented query processing leveraging near data processing
- TEE-based policy enforcement and compliance

Heterogeneous TEEs for host (Intel SGX) and storage (ARM TrustZone)

- Combine execution across TEEs
- Offload computations closer to data (TrustZone-based storage) for performance gains
- Trusted monitor (SGX enclave) to ensure the integrity/authenticity of all components through attestation



Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Security limitations of TEEs

TEEs do not protect accesses outside the secure enclave

- **Loading everything inside the enclave is not always an option**
- **Known side channel attacks with Intel SGX: OS can observe the enclave data accesses at the granularity of pages**

Access patterns in the workflow can reveal information (e.g., order, frequency distribution) for disk/memory resident data

Example:

1. Query Alice's age
2. Query average age of people who voted for X
3. If record retrieved in 1 is also retrieved in 2, Alice voted for X

Oblivious query processing

Objective: make sure memory access patterns are data independent (except for query input/output size) [AK13]

Ensures that the only leakage from a query is the size of input/output, even if the adversary observes memory

i.e., semantic security for queries

Relevant here: adversary is assumed to control all memory external to secure hardware

ORAM (Opaque [ZDB+17]), ObliDB [EZ17], Oblix [MPC18], Path ORAM[HH21]…

What if enclaved code cannot be trusted?

Problem: TEEs do not ensure that malicious code cannot voluntarily leak data

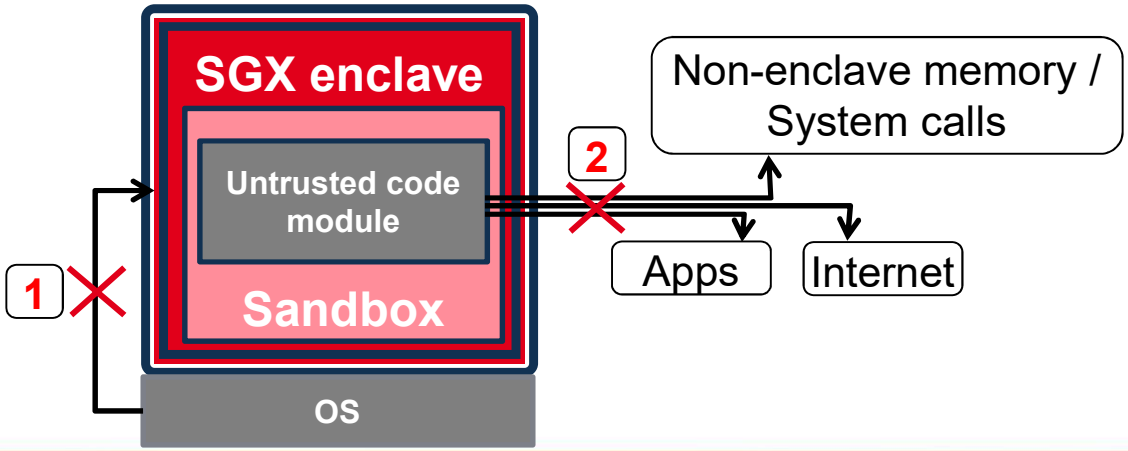
Confining untrusted code through double isolation

1. **Execute code inside of an enclave untrusted platform cannot access private enclave-memory data**
2. **Restrict accessible memory of the untrusted module execution with a sandbox module cannot copy secrets to non-enclave memory**

Ryoan [HZX18] sandboxing

Based on Google's Native Client: can only address module memory, intercepts syscalls…

Similar alternative sandboxing solutions: SGXJail [WSG19], SGX-LKL [PML+19], DBT [CSS+22]

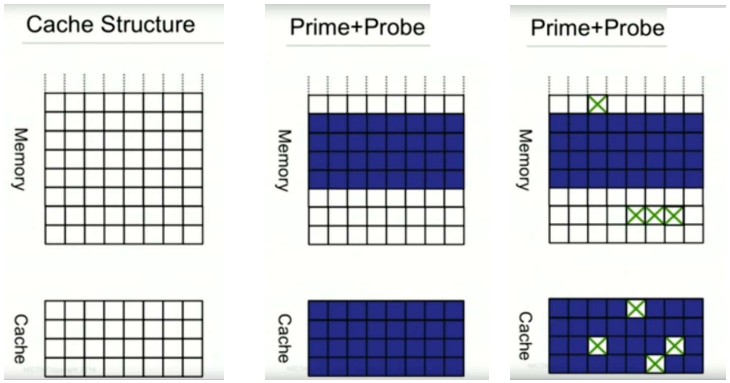


Side-channel attacks on SGX

Malware Guard Extension: abusing Intel SGX to conceal cache attacks

M Schwarz, S Weiser, D Gruss, C Maurice, S Mangard

Cybersecurity 3 (1), 2



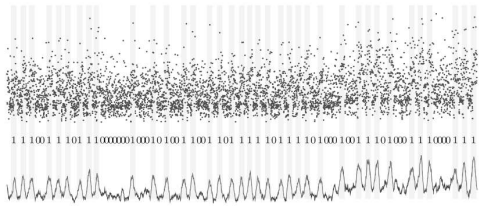


Fig. 7 On top is one trace's raw measurement over 4000000cycles. The peaks in the resampled trace on the bottom clearly indicate '1's

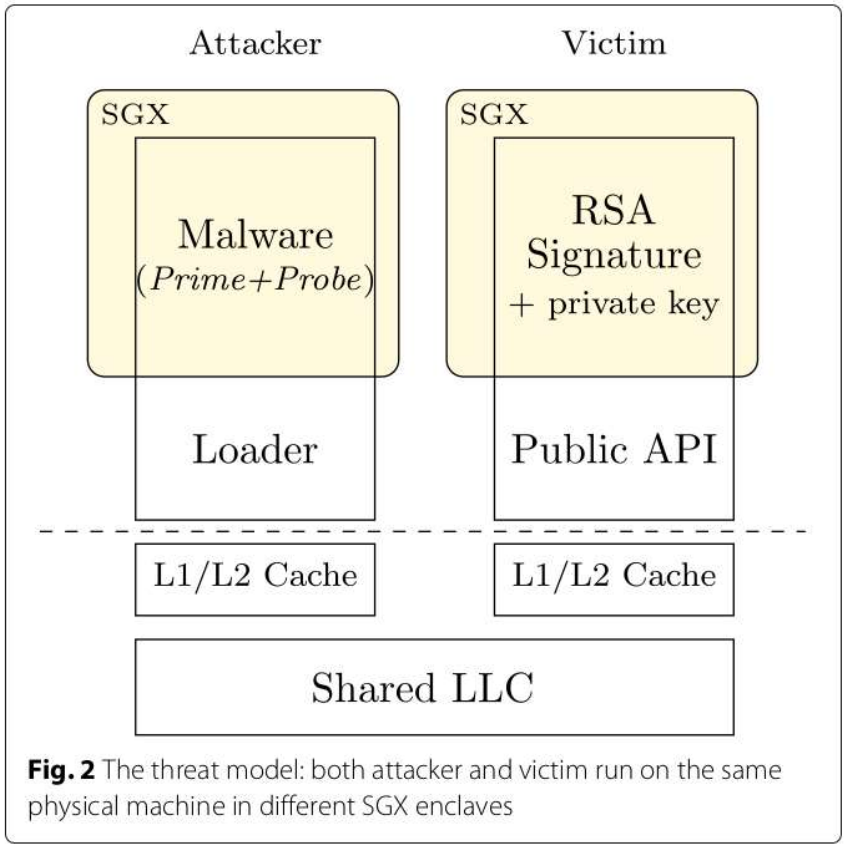


Fig. 2 The threat model: both attacker and victim run on the same physical machine in different SGX enclaves

Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

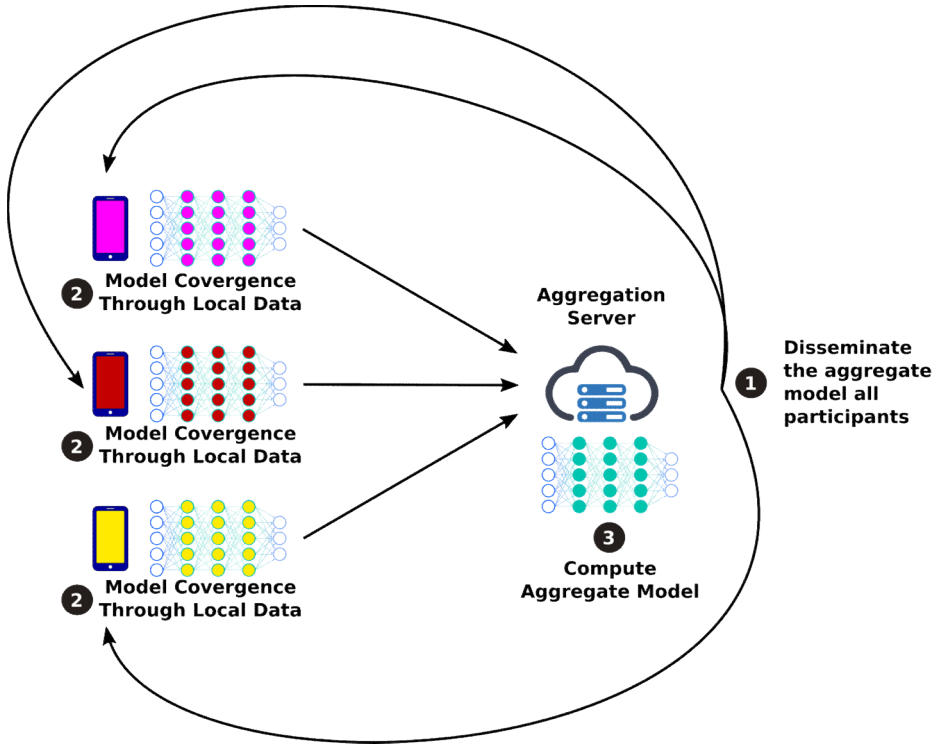
- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

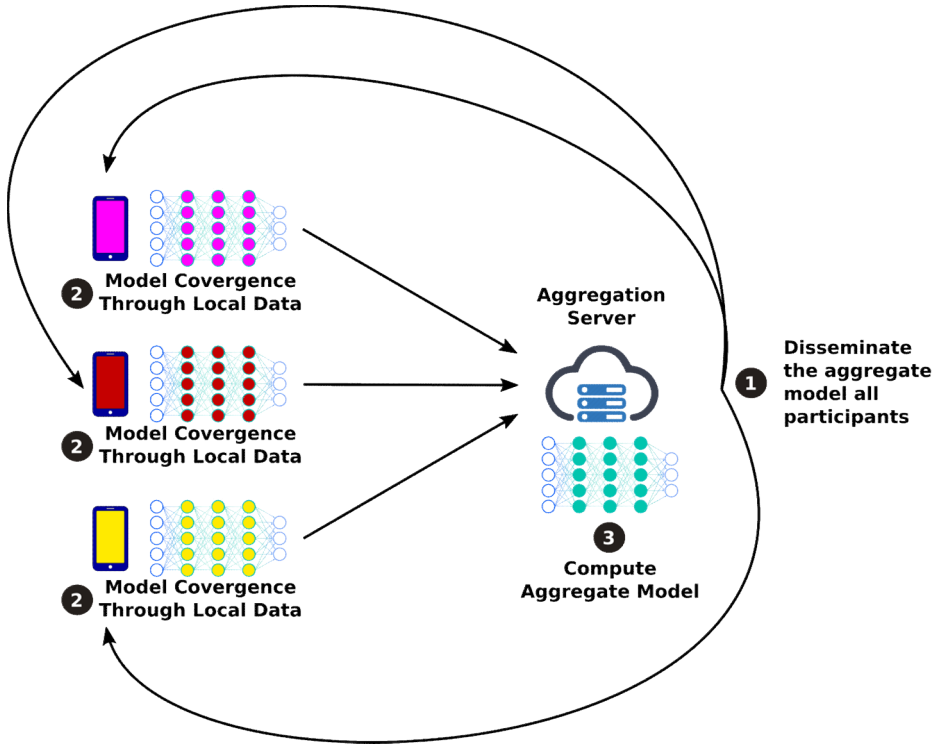
Tutorial introducing SGX

Federated Learning



Tutorial introducing SGX

Federated Learning



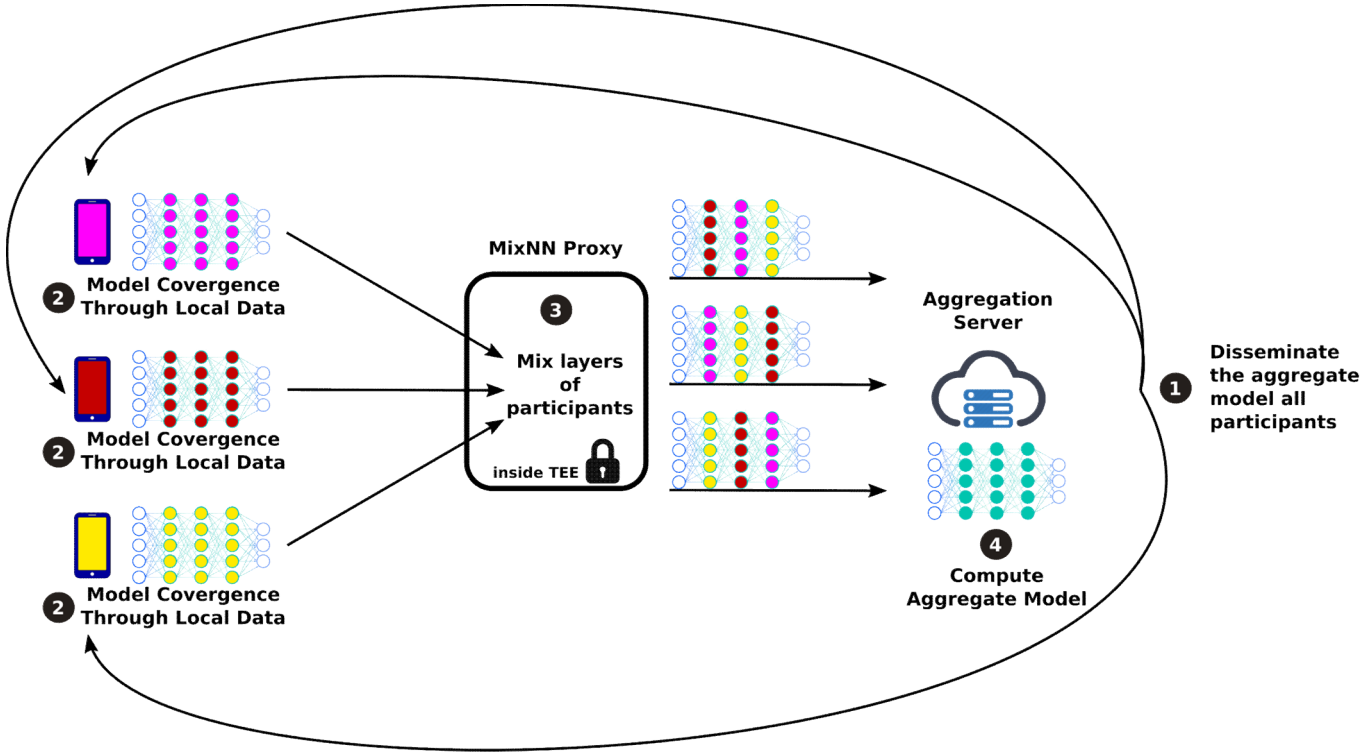
- Limitations
- Countermeasures
  - Perturbation (e.g., Differential Privacy [1])

    ➔ Drastically reduces accuracy
  - Crypto (e.g., Secure Aggregation [2])

    ➔ Overhead

Tutorial introducing SGX

Federated Learning scheme using SGX: MixNN [1]



➔ **Improve the privacy (at low cost) without compromising the utility**

[1] MixNN : protection of federated learning against inference attacks by mixing neural network layers, T Lebrun et al., Middleware 2022

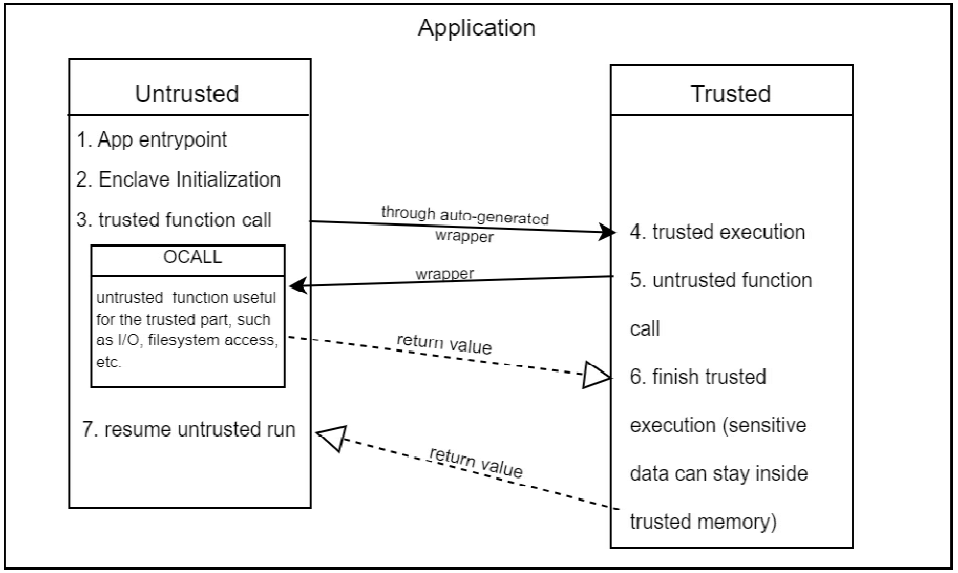
Tutorial introducing SGX

Installation

- Guide à travers l'installation des drivers & SDK SGX + OpenSSL SGX

Bases

- Architecture de base d'un projet SGX
- Initialisation d'une enclave
- Principe OCALL & ECALL : permet de faire un "context-switch" entre la partie trusté & non-trusté
- Comment faire transiter des données & stockage de données sécurisé pour l'enclave



Tutorial introducing SGX

Crypto

- Utilisation de la librairie de crypto OpenSSL : génération d'une paire de clés RSA et l'implémentation d'un chiffrement / déchiffrement basique
- Principe de "Sealing", chiffrement d'une données qui peut être enregistrée sur disque afin d'être réutilisée entre plusieurs run d'une enclave, ici chiffrement de la clé privée

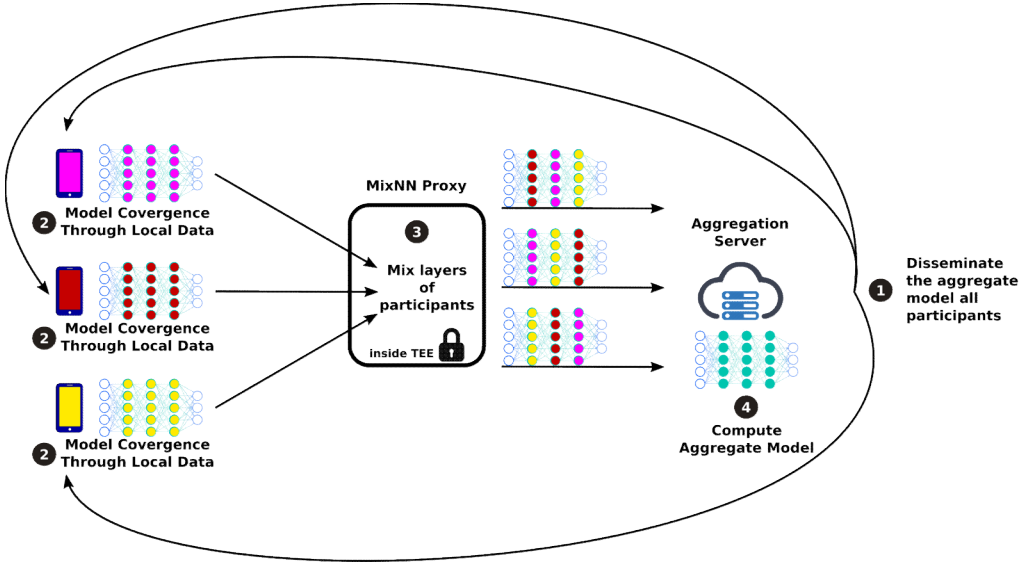
Multi Threading

- Mise en place du multi-threading à l'intérieur de l'enclave
- Synchronisation entre les différents threads

Tutorial introducing SGX

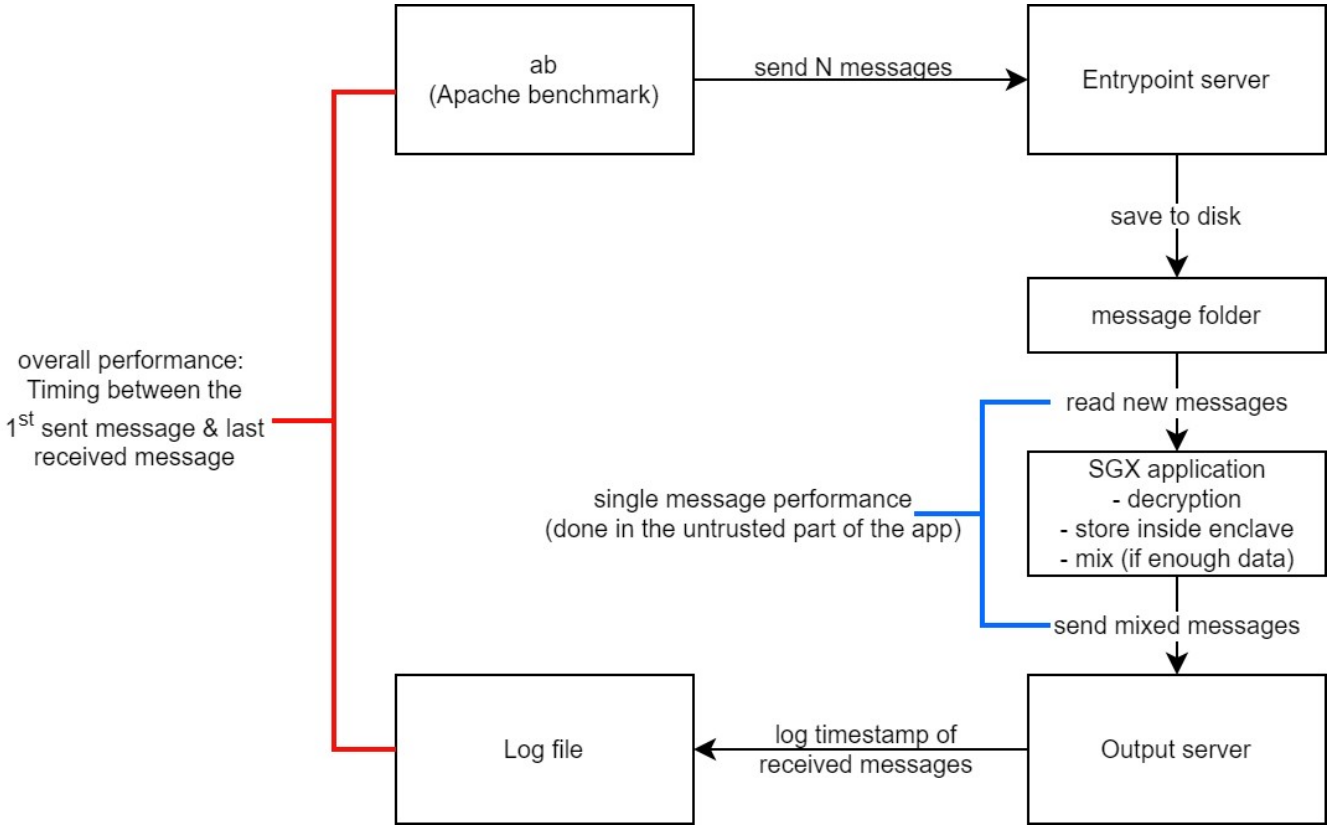
Implémentation MixNN

- Génération d'une paire de clés et partage de la clé publique
- Déchiffrent des messages en entrée et stockage de leur contenue à l'intérieur de l'enclave
- Une fois un seuil de message atteint, mixage du contenu des messages et sortie de l'enclave avec transmission des messages mixés au serveur
- Les messages sont simplement des tableaux d'entiers, ce qui simplifie le stockage dans l'enclave et les structures de données avec lesquels le mixage doit être fait



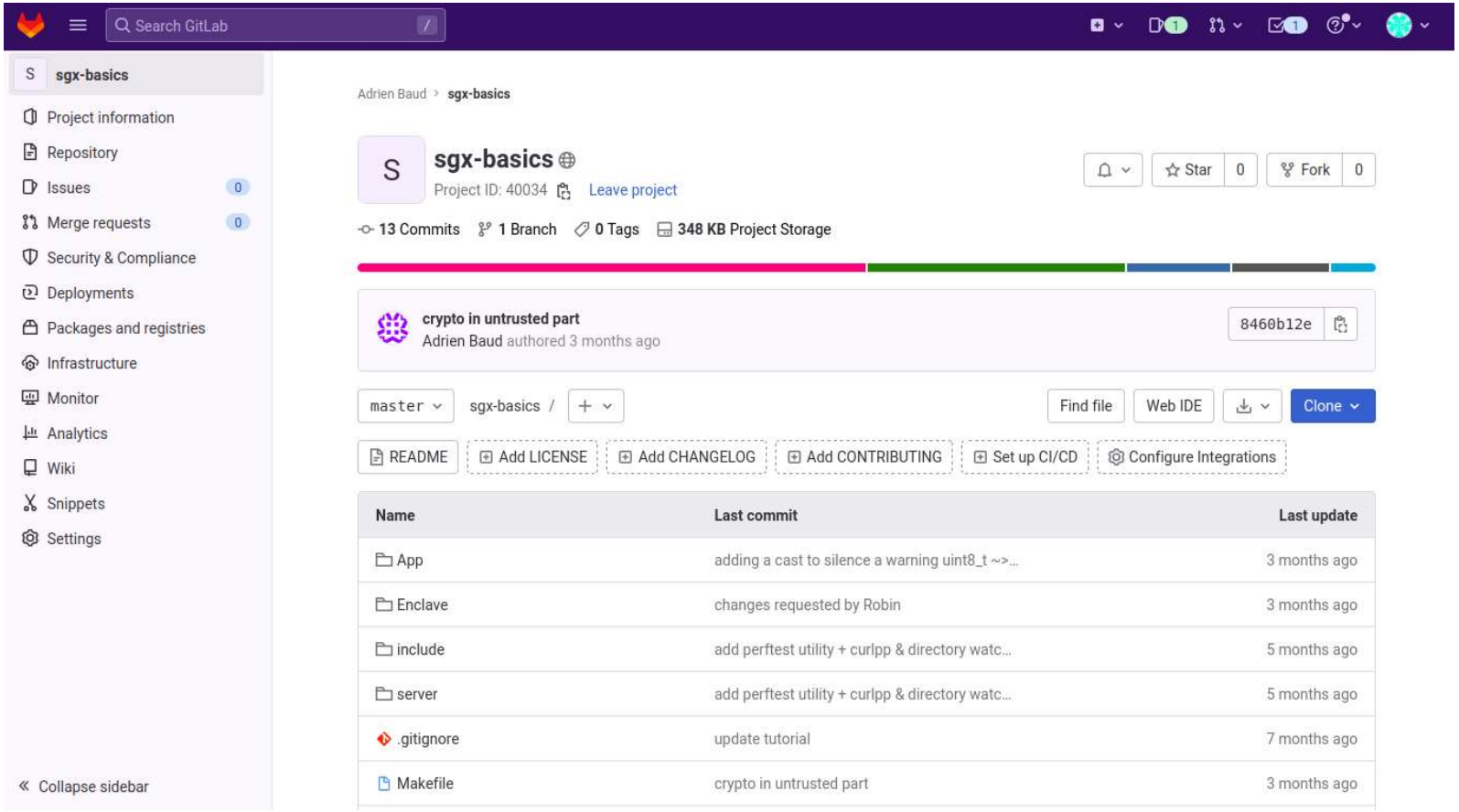
Tutorial introducing SGX

Évaluation des performances



Tutorial introducing SGX

https://gitlab.inria.fr/abaud/sgx-basics



Overview

PART I – Introduction to Trusted Execution Environments (TEE)

- Historical view of TEEs
- Promises and architecture of TEEs
- Intel Software Guard eXtensions (SGX)
- Efficient data processing with SGX
- Limitations of TEEs and attacks on TEEs (SGX)

PART II – Tutorial introducing SGX

- Federated Learning scheme using SGX

PART III – Use-case

- Secure and Extensive Personal Data Management Systems with SGX

Current trend: return personal data to the individuals to enable individual agency

Act I: the right to Data portability

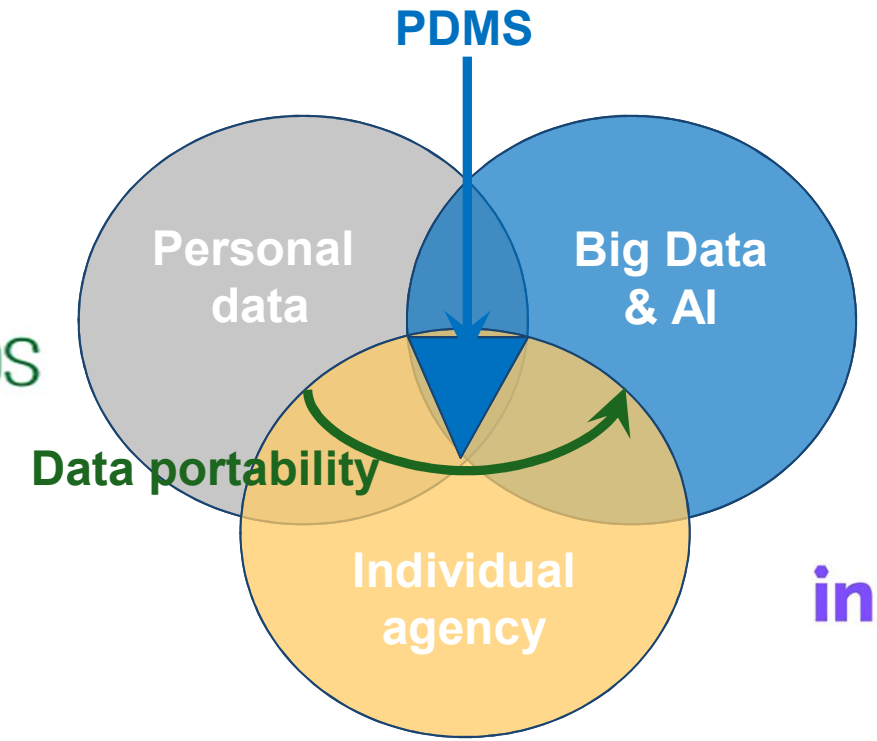
… the right to retrieve its own data

Blue Button Download My Data

Green Button Connect My Data

MesINFOS

General Data Protection Regulation



Act II: Personal Data Mg^t Systems (PDMS)

… the tool to manage its own data

cozy.io

digi.me

DATABOX

BITSABOUT.ME

inrupt Tim Berners-Lee

Personal Data Management Systems

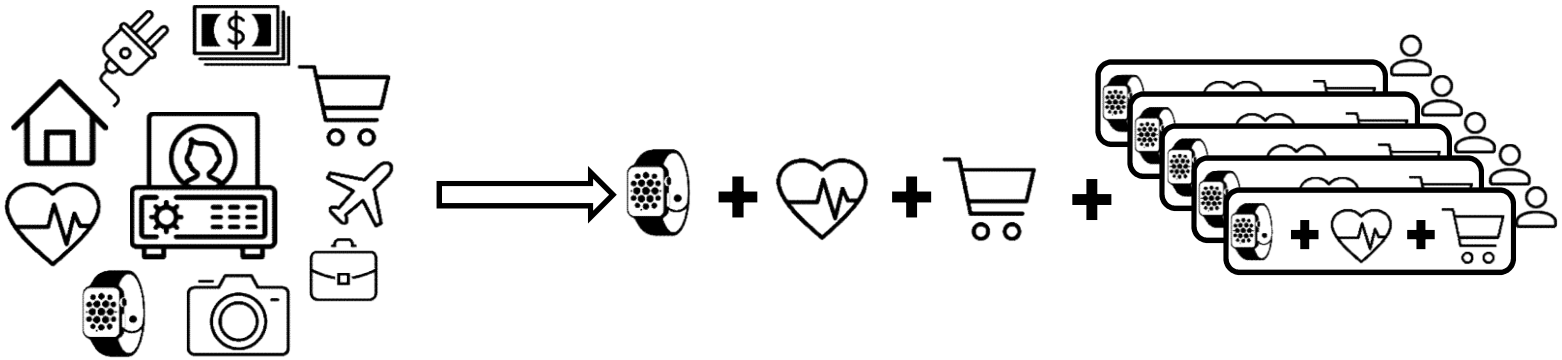
Solutions to gather the digital life of an individual

More and more solutions available (Cozy Cloud, Digi.me, Personal Infomediaries . . .)

Boosted by regulations such as Data Portability (GDPR)

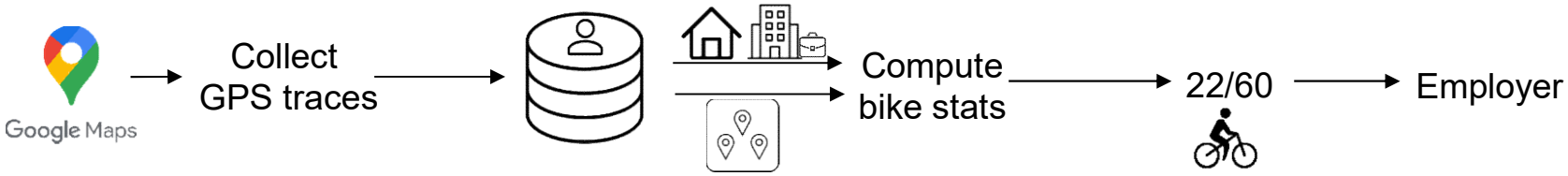
Crosses data from multiple data sources and/or multiple users

Promising paradigm : computation comes to data



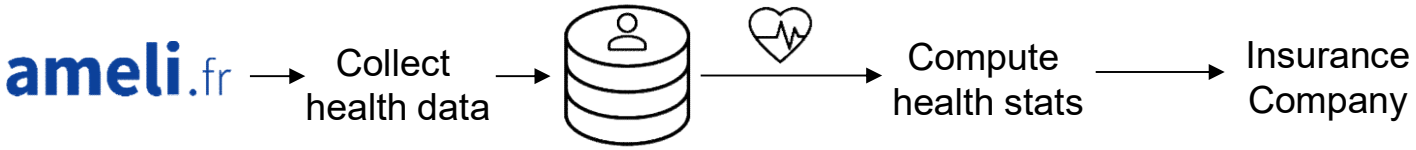
Examples of PDMS related use-cases

Companies want to reward employees commuting by bike



Green Bonus workflow

Citizens wants to get a quotation from a new health insurance company

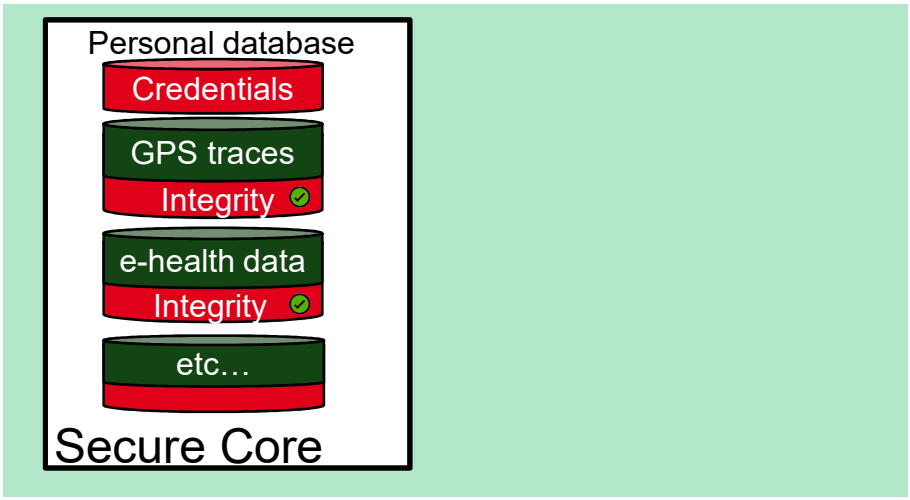


Health insurance workflow

Such use-cases require

- **Privacy**
- **Verifiability**
- **Extensibility**

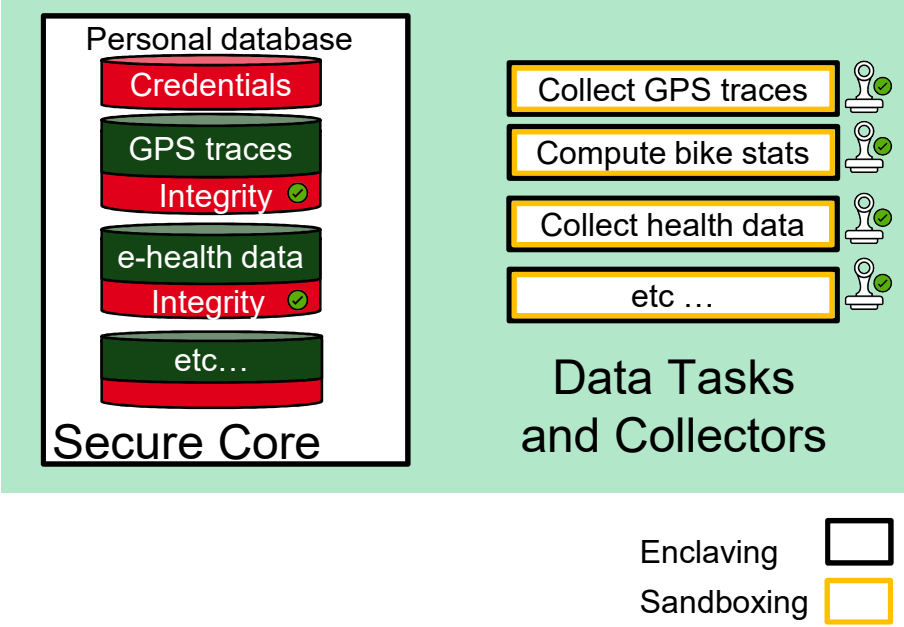
PDMS architecture leveraging SGX enclaves



Elements & Properties

- **one Secure Core**
  - Trusted Computing Base
  - Minimal and Inextensible
  - Enclaved

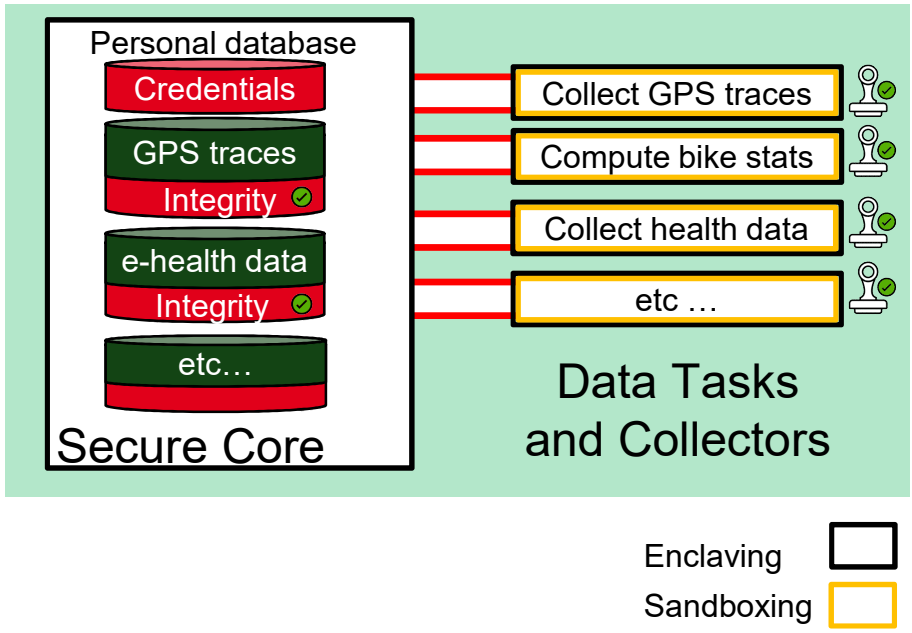
PDMS architecture leveraging SGX enclaves



Elements & Properties

- **one Secure Core**
  - Trusted Computing Base
  - Minimal and Inextensible
  - Enclaved
- **and Data Tasks/Collectors**
  - Code extensions
  - Enclaved
  - Sandboxed

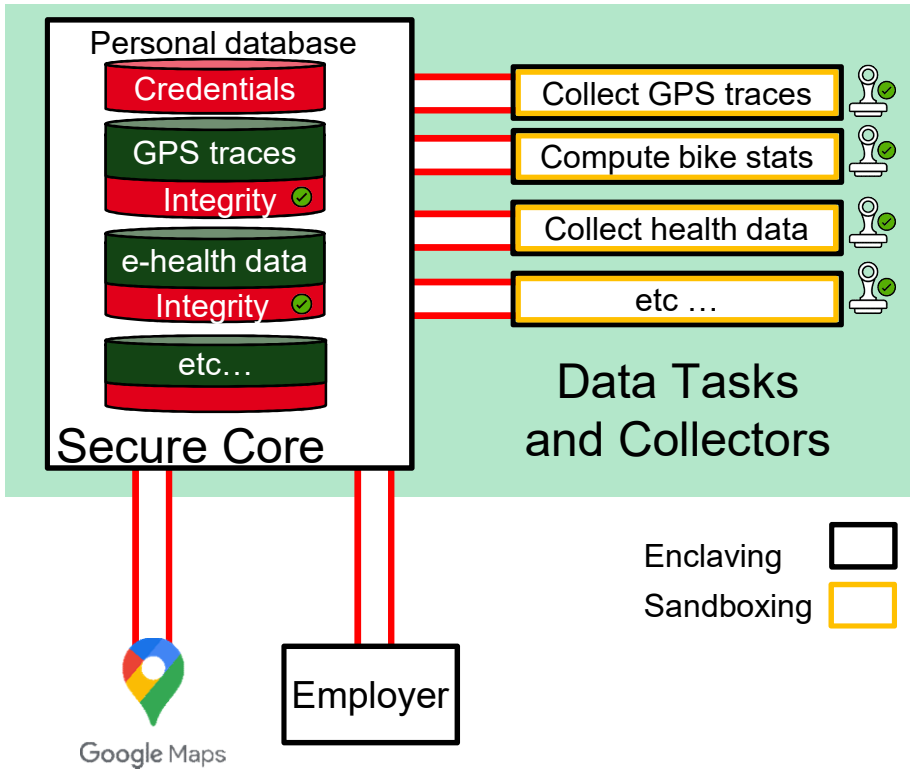
PDMS architecture leveraging SGX enclaves



Elements & Properties

- **one Secure Core**
  - Trusted Computing Base
  - Minimal and Inextensible
  - Enclaved
- **and Data Tasks/Collectors**
  - Code extensions
  - Enclaved
  - Sandboxed
- **Secure communications**

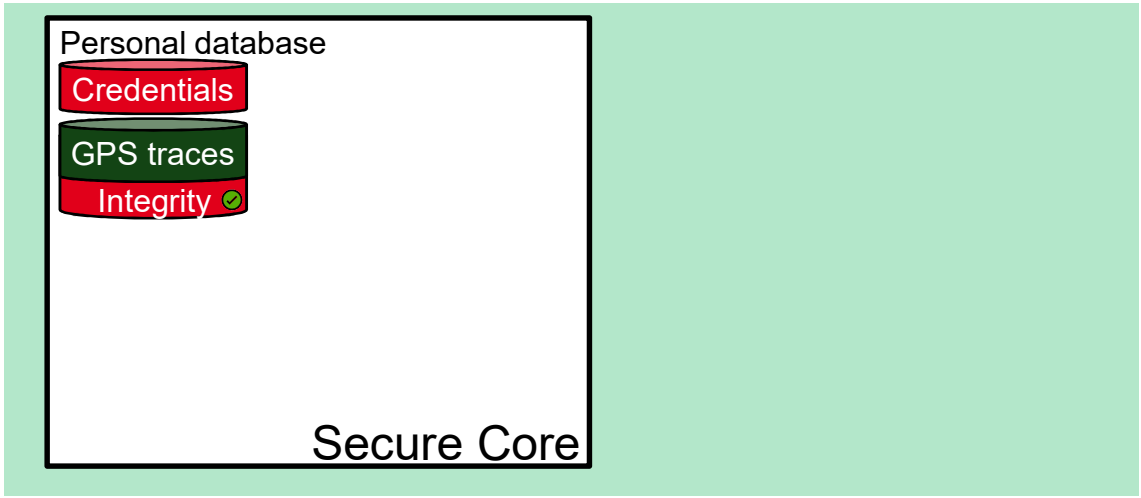
PDMS architecture leveraging SGX enclaves



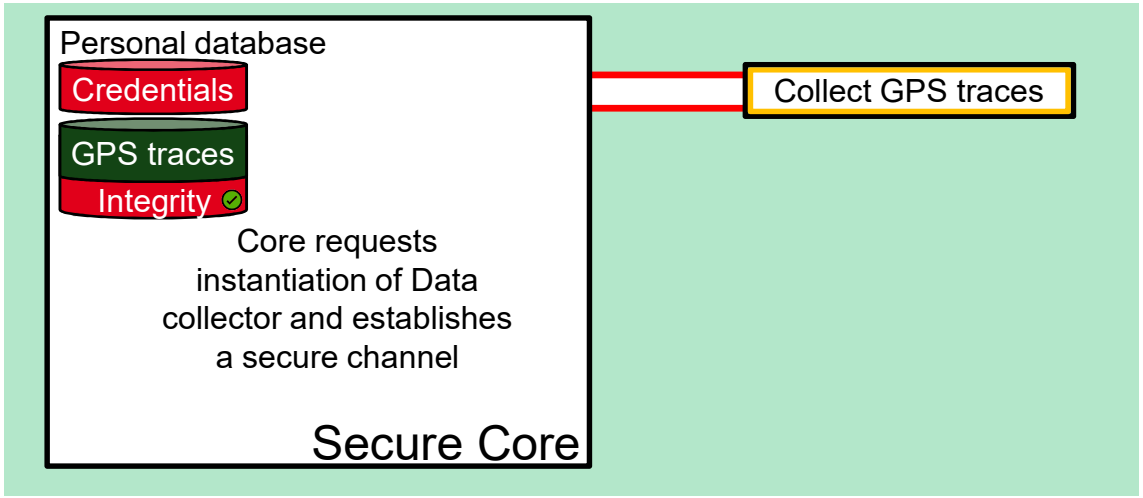
Elements & Properties

- **one Secure Core**
  - Trusted Computing Base
  - Minimal and Inextensible
  - Enclaved
- **and Data Tasks/Collectors**
  - Code extensions
  - Enclaved
  - Sandboxed
- **Secure communications**
- **Core as proxy**

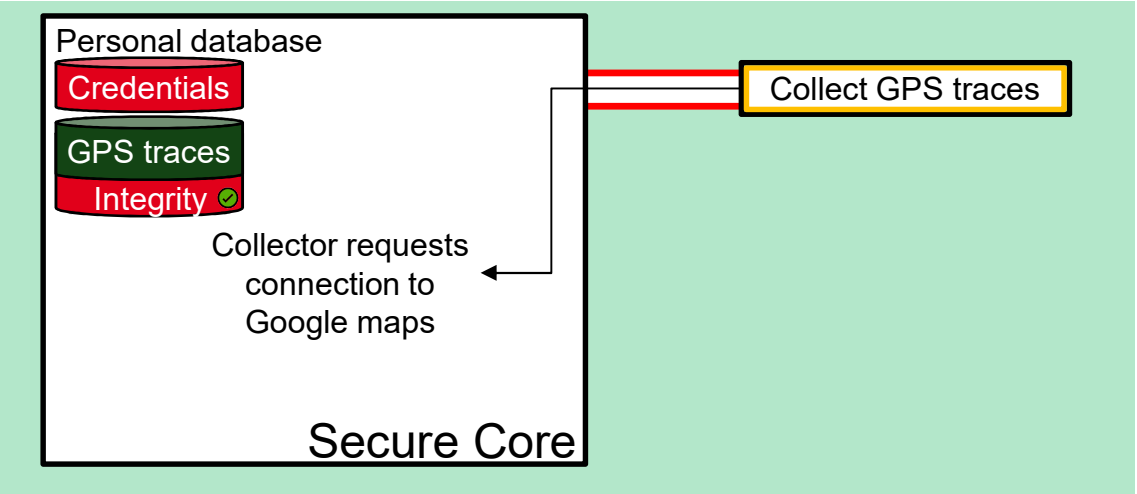
Data collection



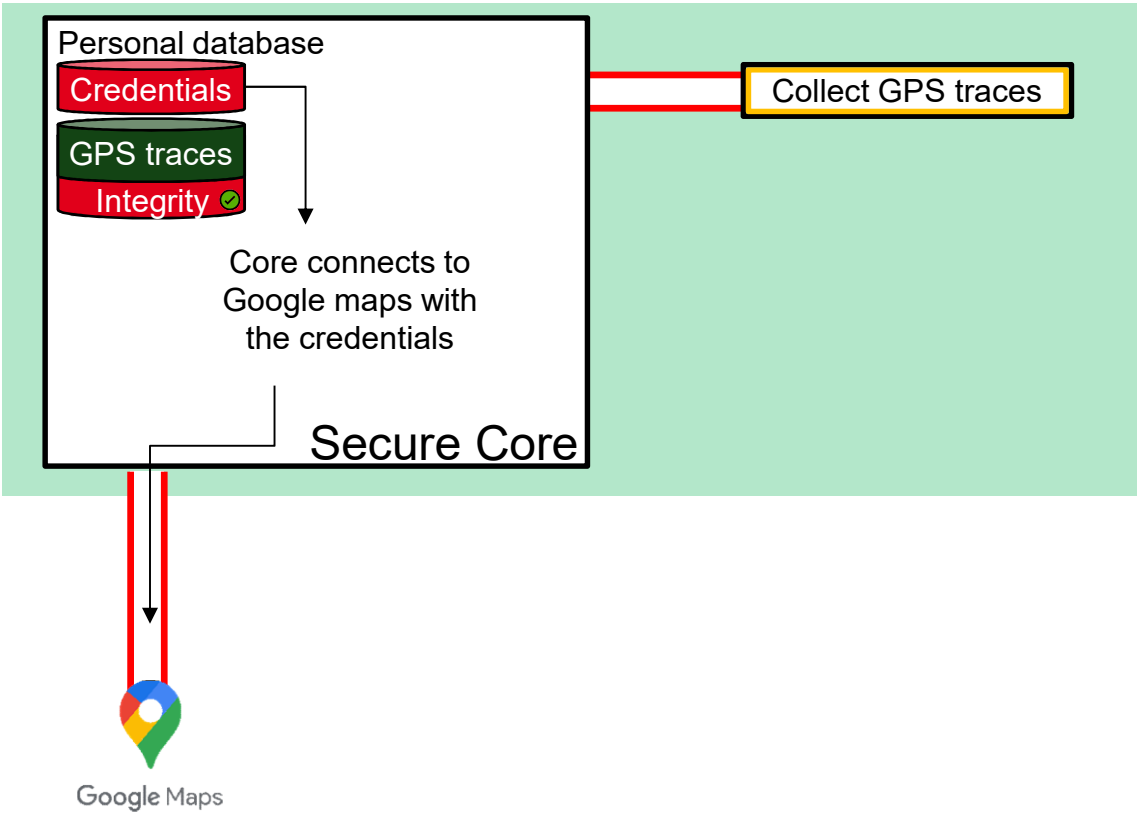
Data collection



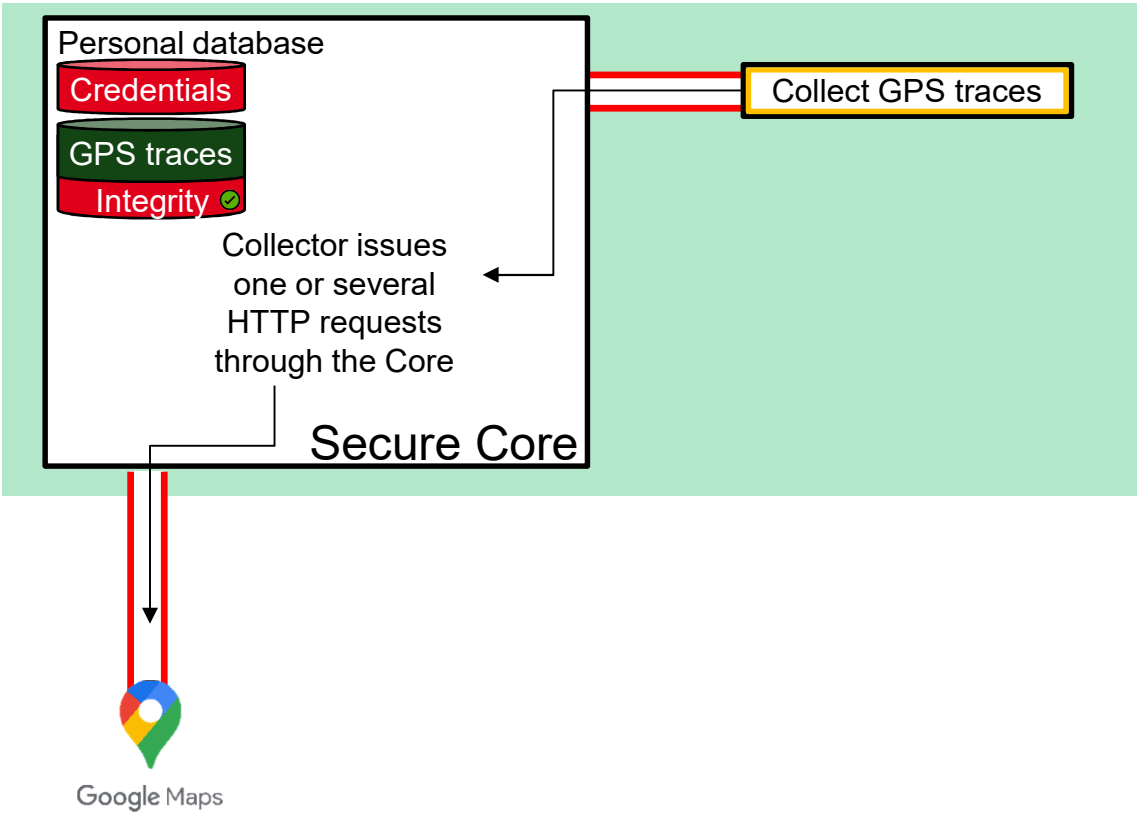
Data collection



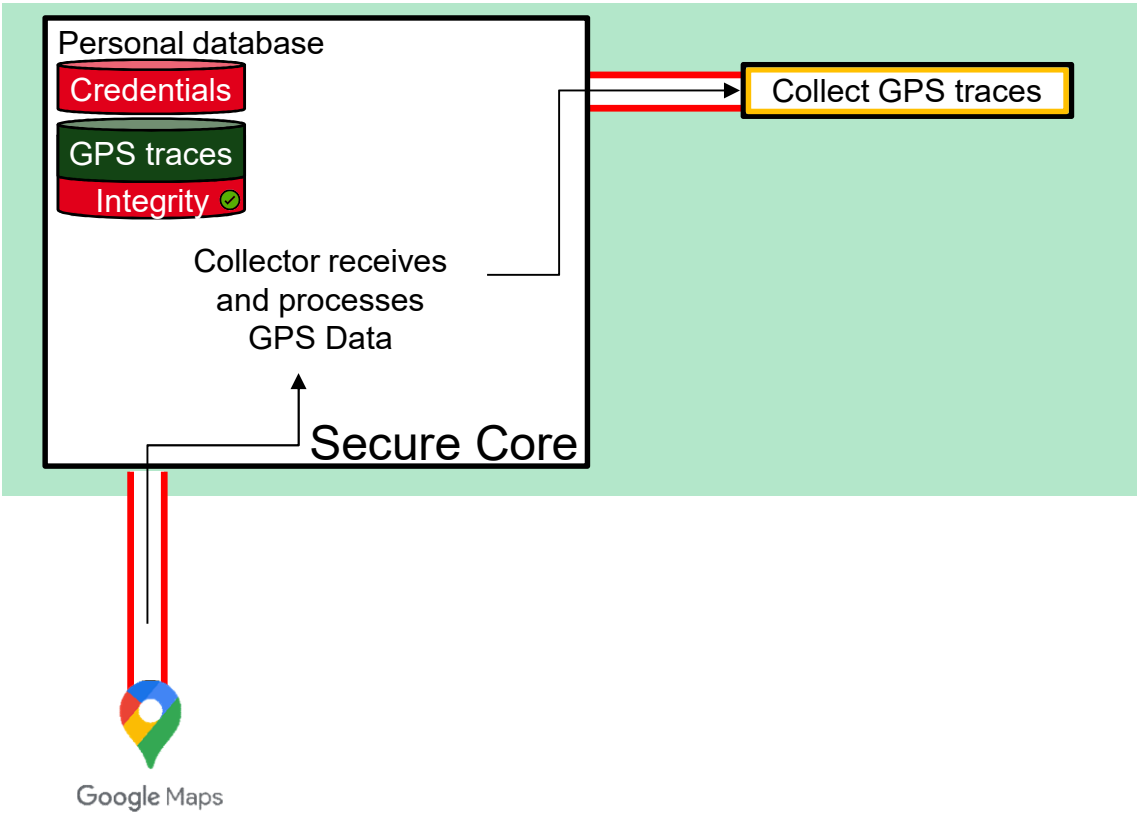
Data collection



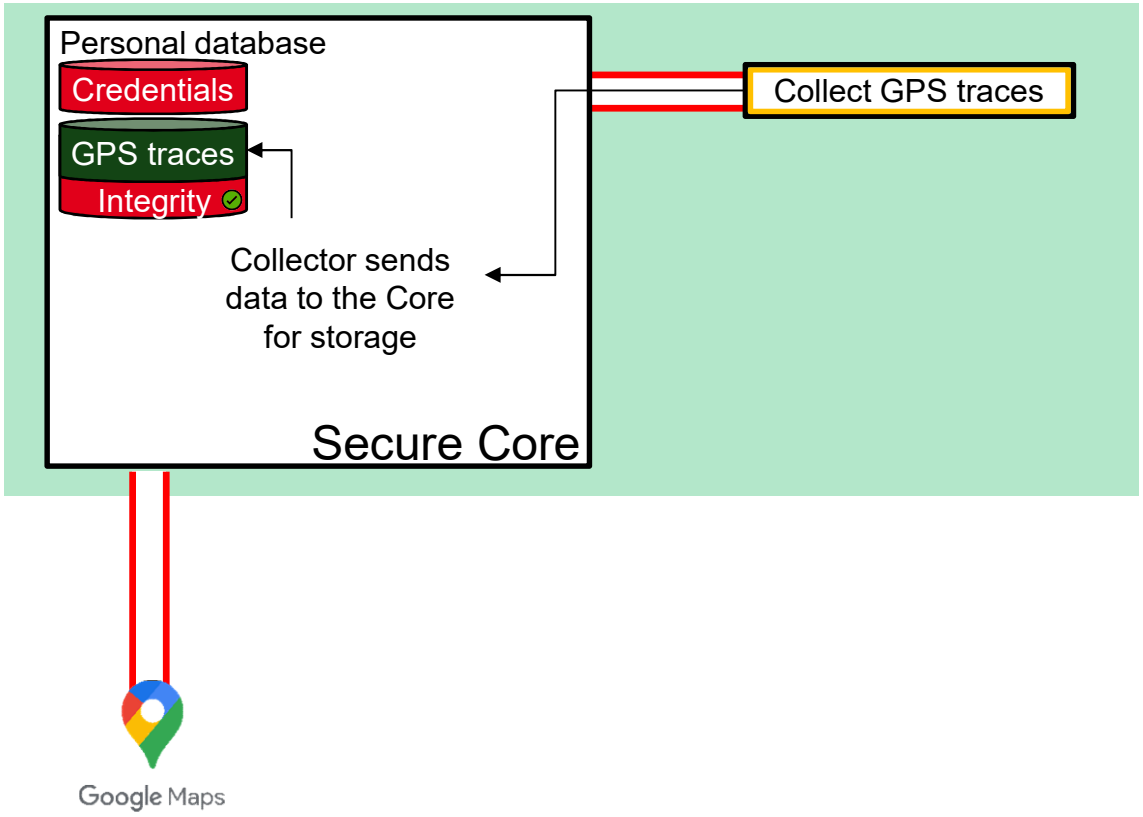
Data collection



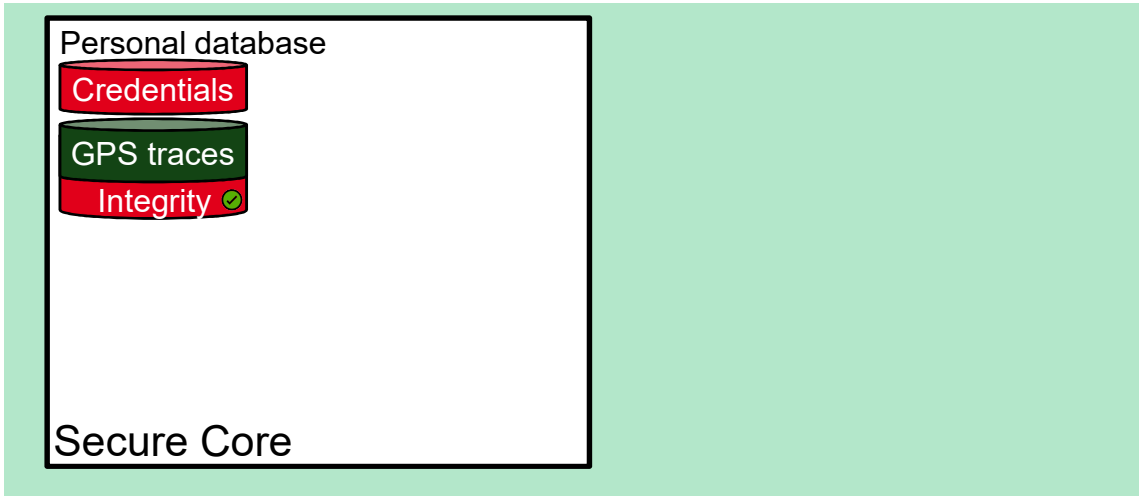
Data collection



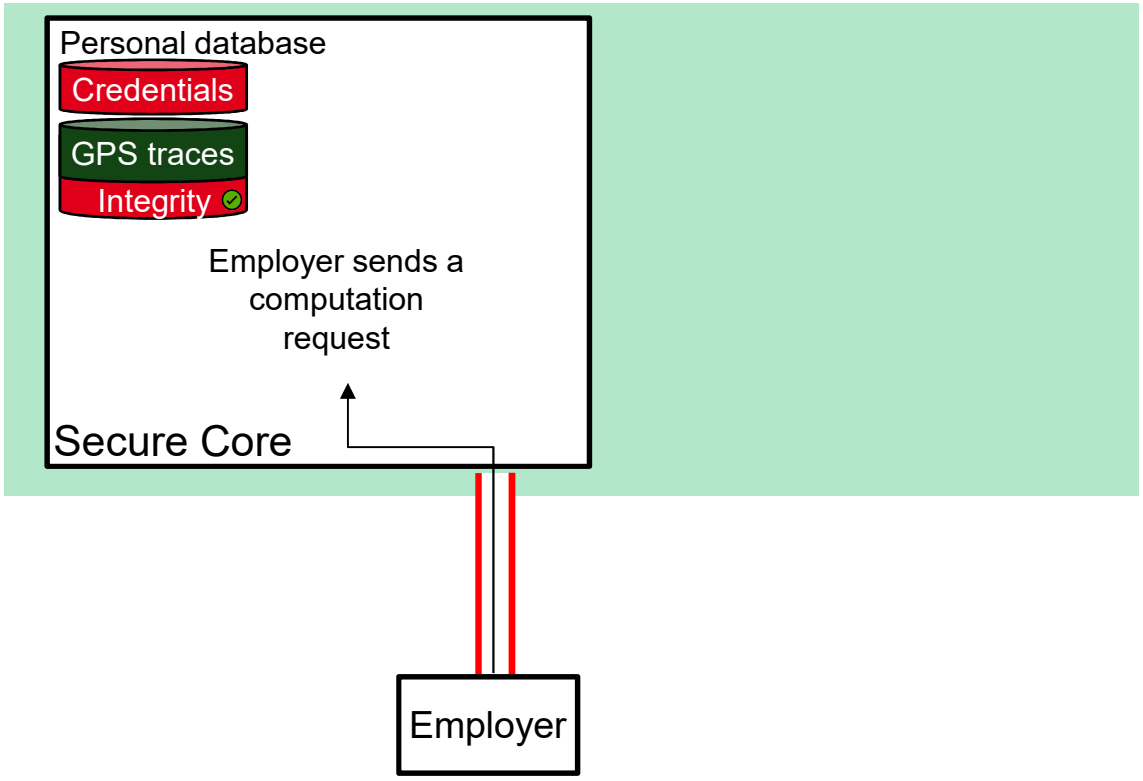
Data collection



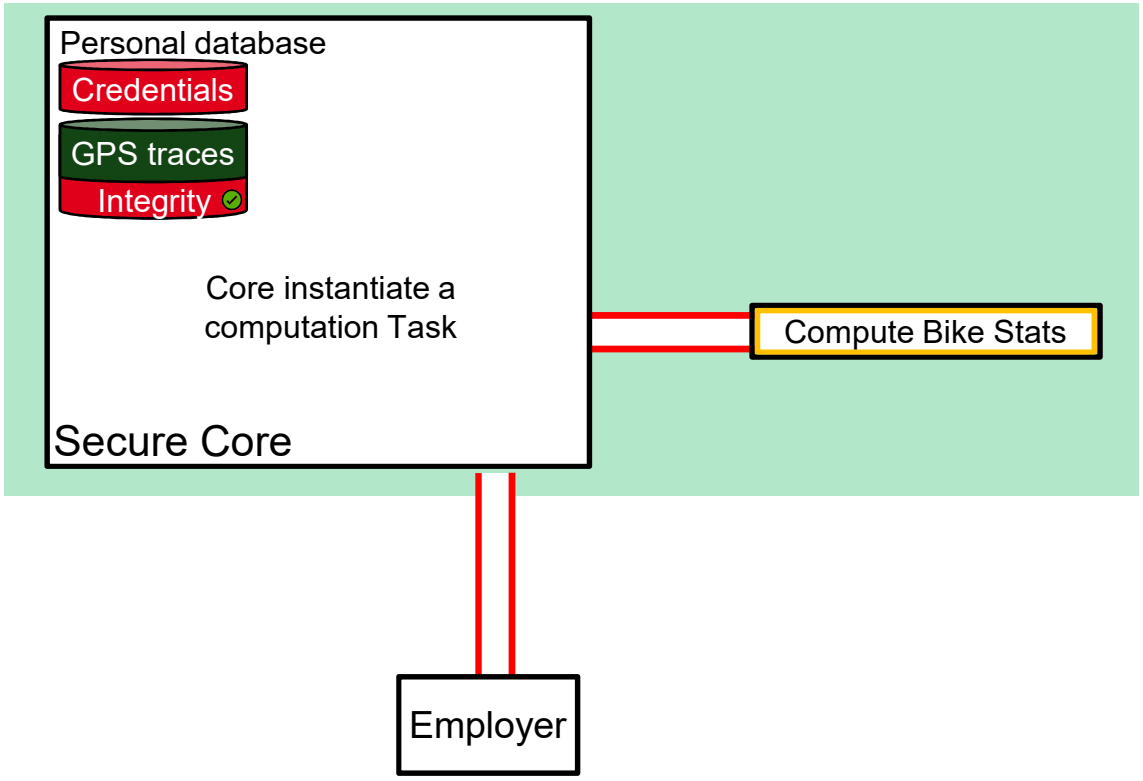
Data computation



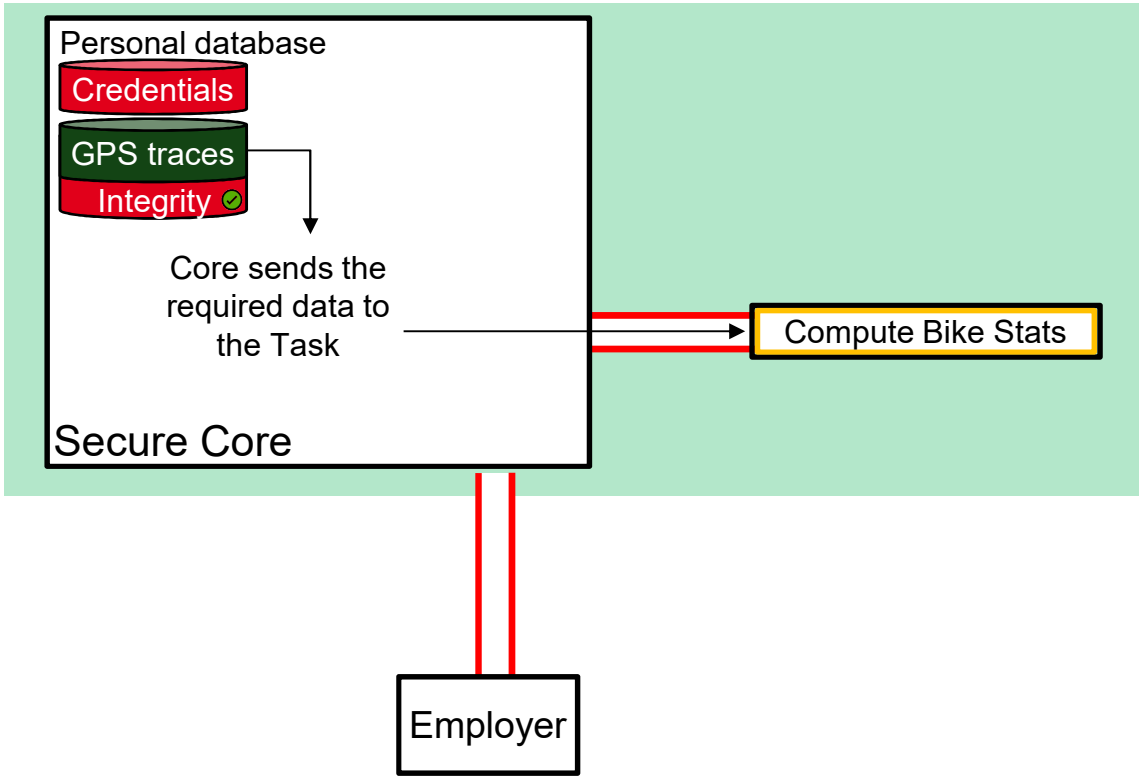
Data computation



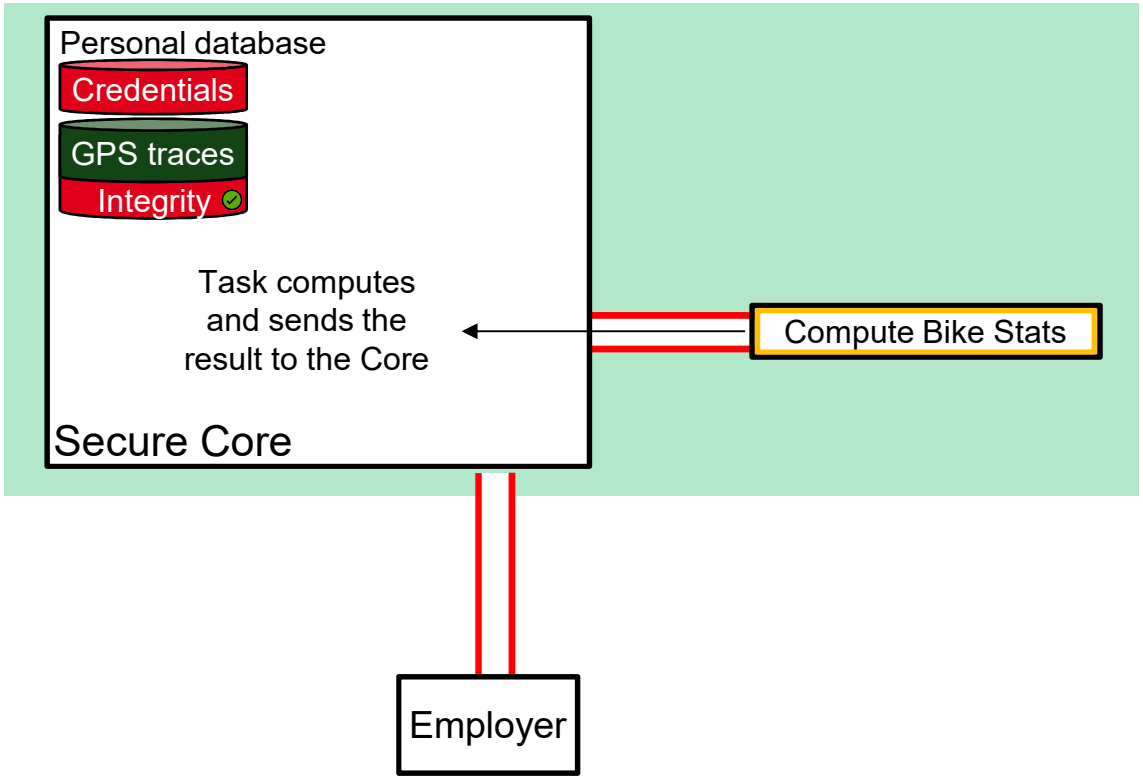
Data computation



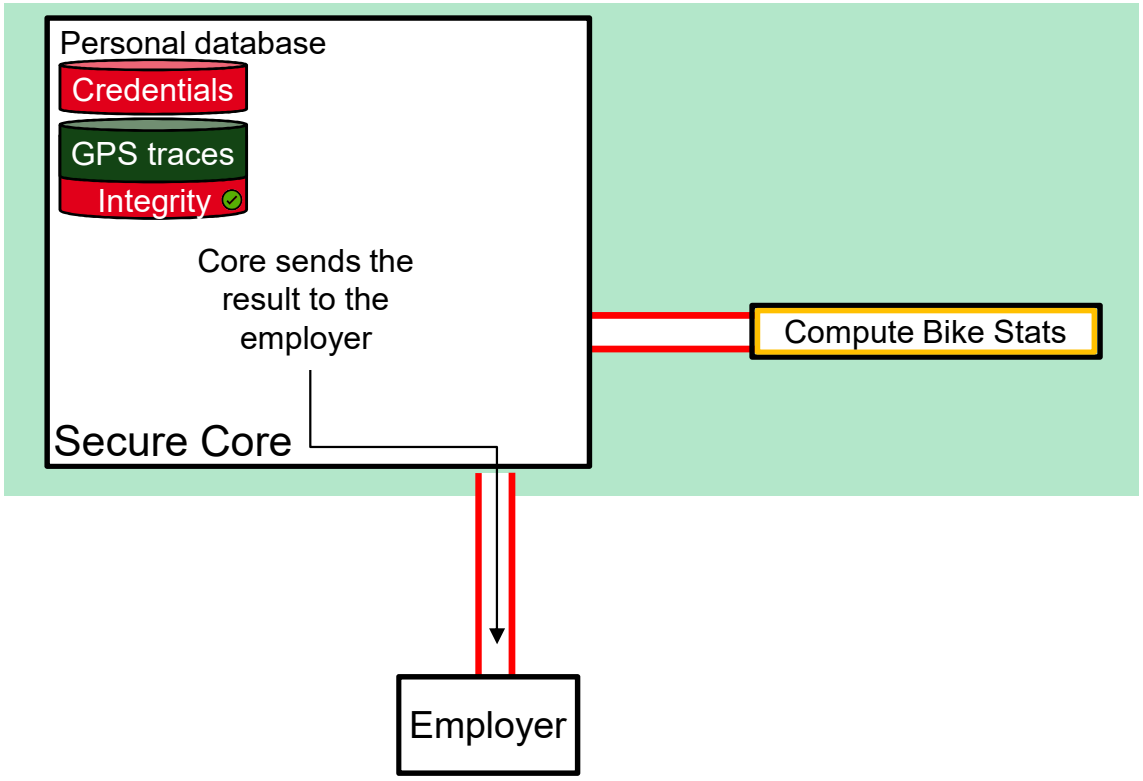
Data computation



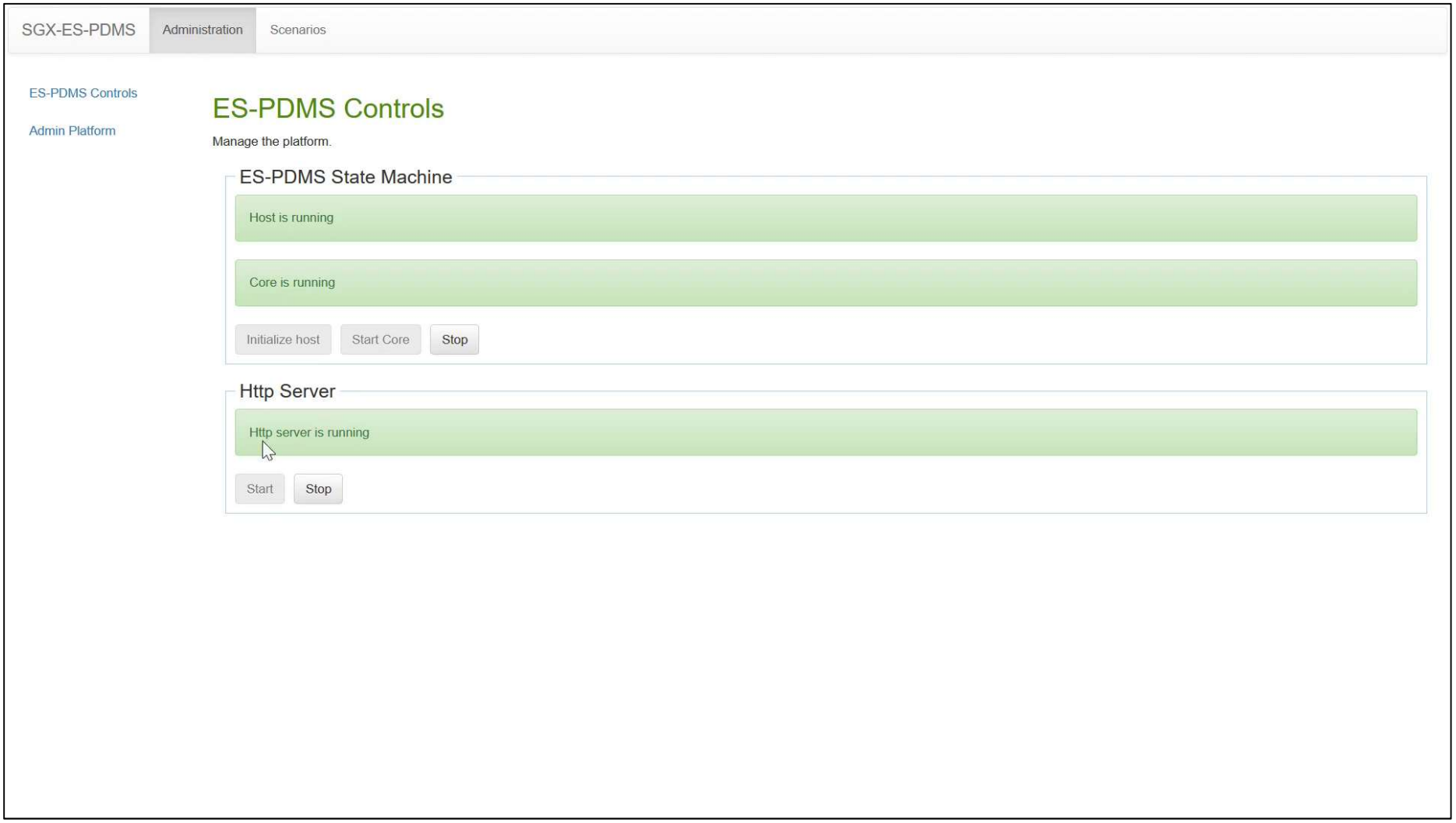
Data computation



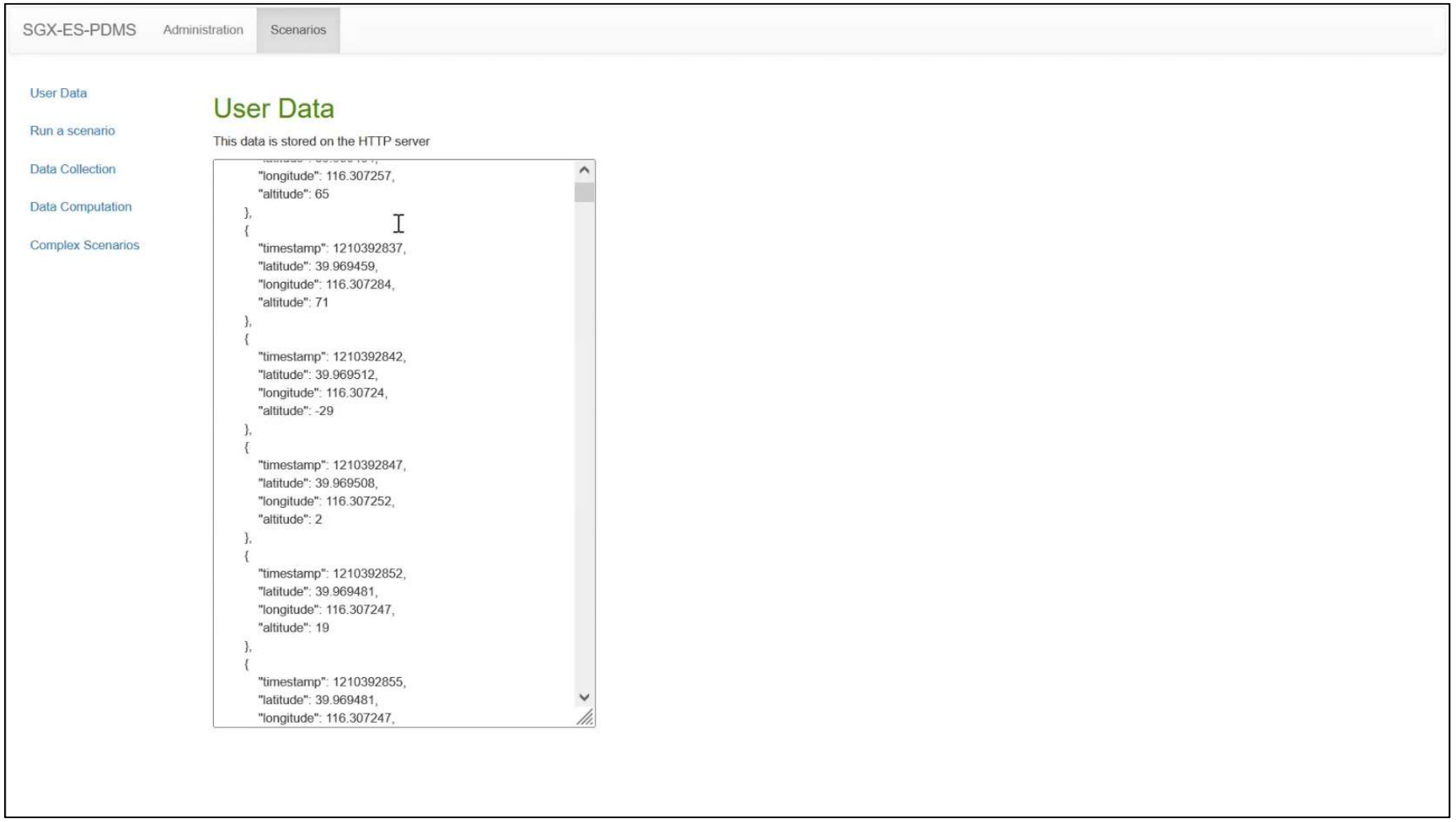
Data computation



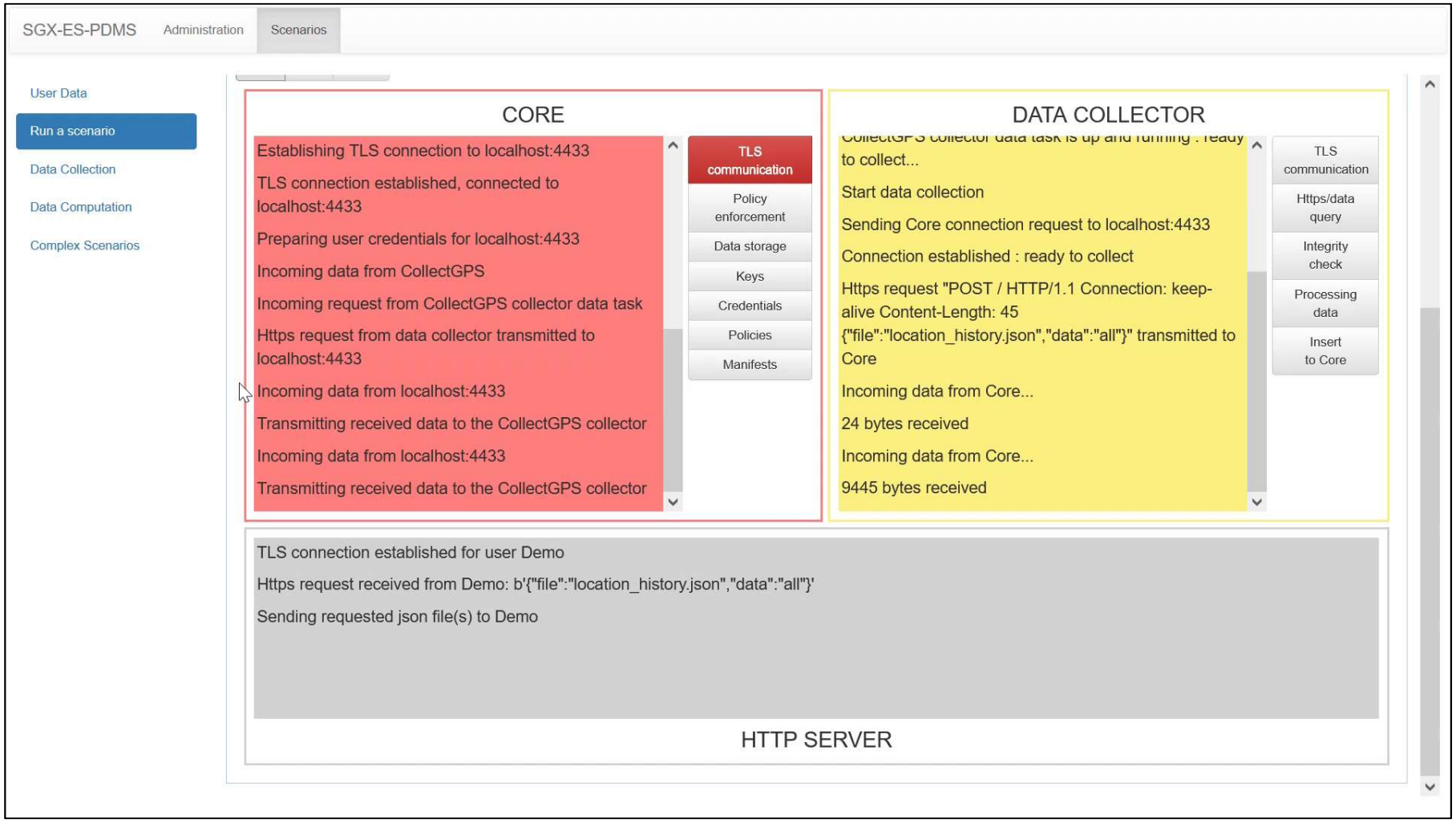
Prototype PDMS on SGX project.inria.fr/espdms



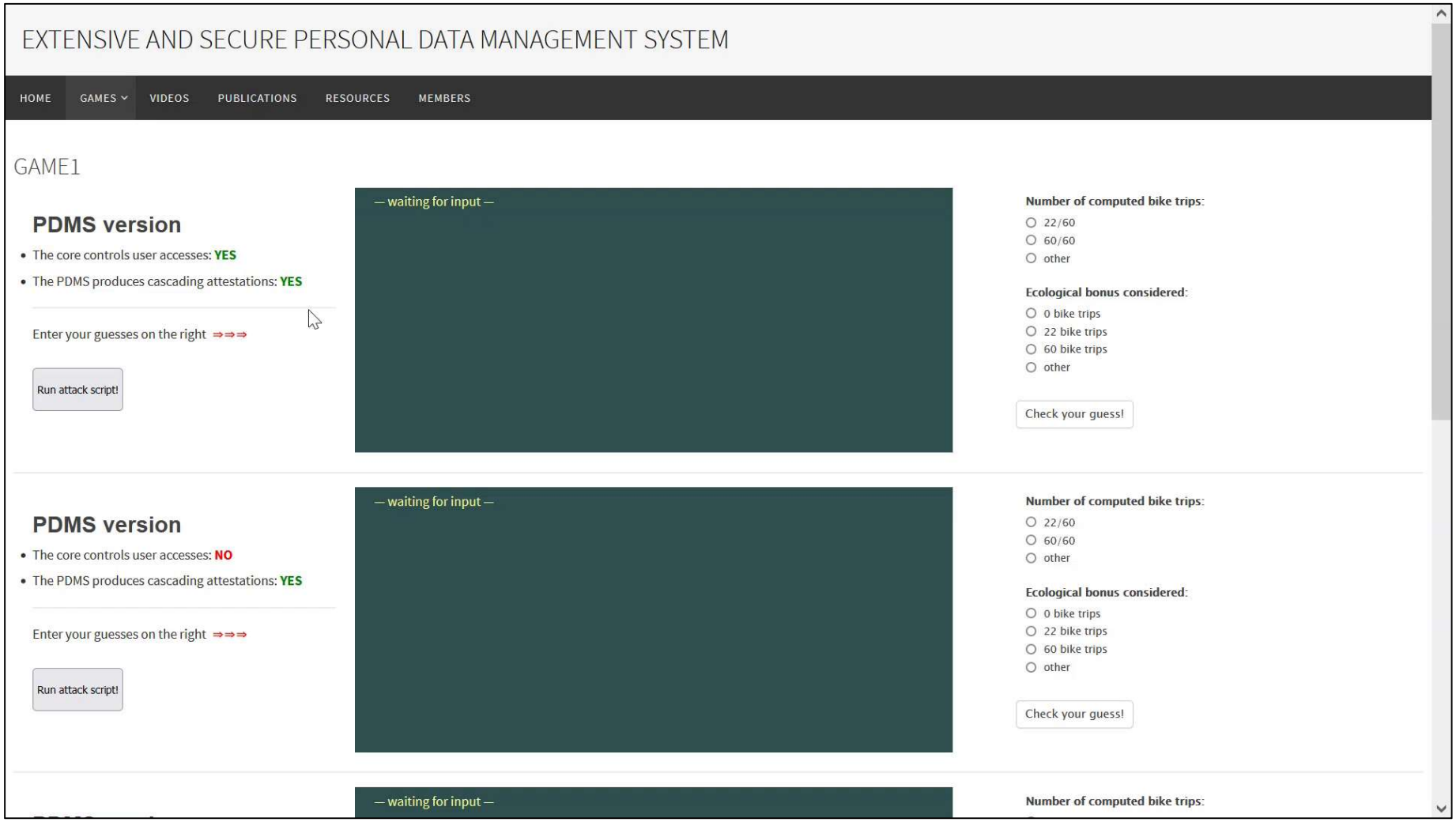
Prototype PDMS on SGX project.inria.fr/espdms



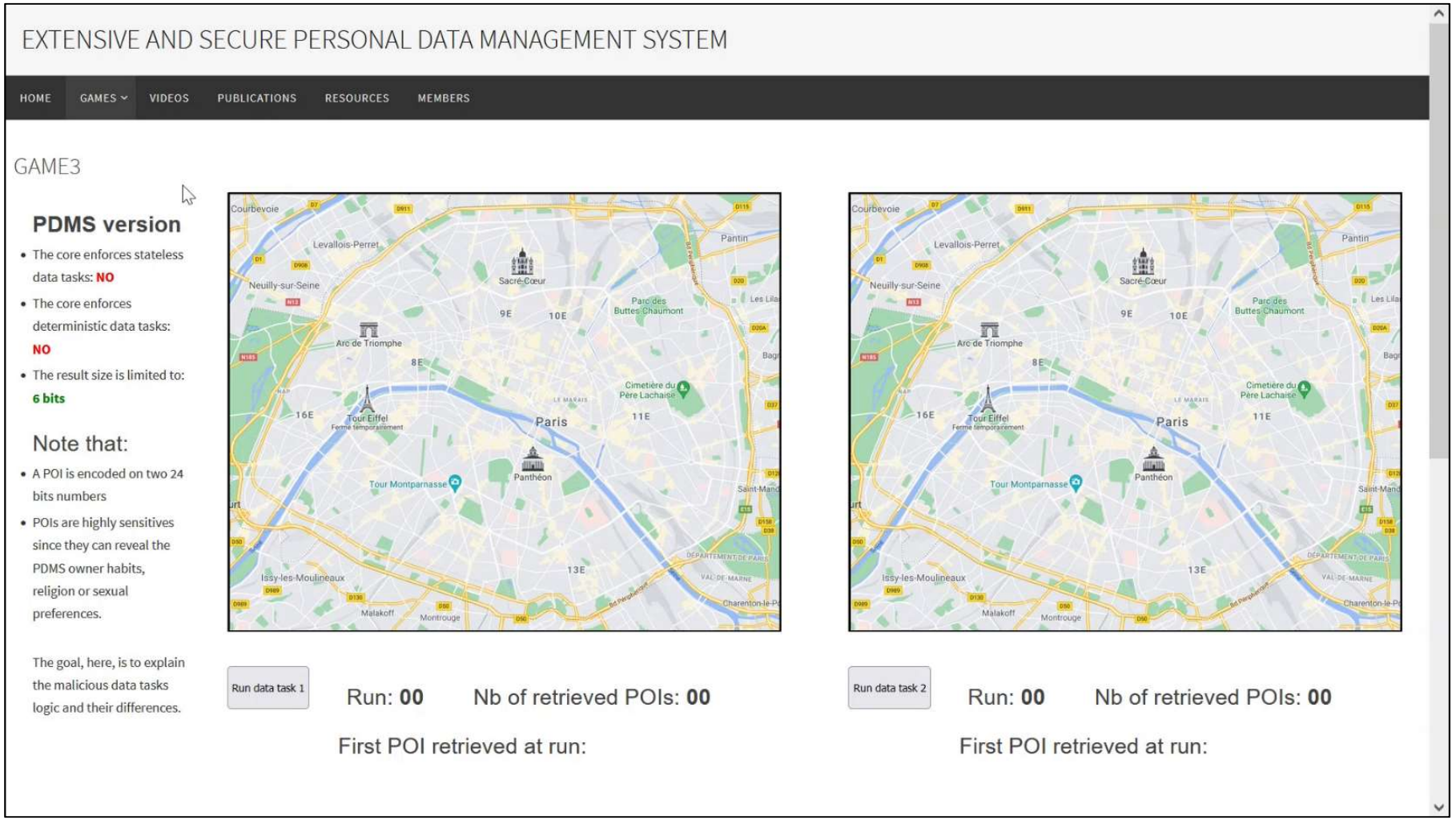
Prototype PDMS on SGX project.inria.fr/espdms



Prototype PDMS on SGX project.inria.fr/espdms



Prototype PDMS on SGX project.inria.fr/espdms



Thanks !

Questions ?

Inria

References (1)

[AAB+10] T. Allard, N. Anciaux, L. Bouganim, Y. Guo, L. L. Folgoc, B. Nguyen, P. Pucheral, I. Ray, S. Yin. Secure personal data servers: a vision paper. PVLDB, 3(1), 25-35, 2010.

[ABB+19] N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, P. Pucheral, I. S. Popa, G. Scerri. Personal data management systems: The security and functionality standpoint. Inf. Syst., 80:13–35, 2019.

[ABD+19] M. Acosta, T. Berners-Lee, A. Dimou, J. Domingue, L-D. Ibá, K. Janowicz, M-E. Vidal, A. Zaveri: The FAIR TRADE Framework for Assessing Decentralised Data Solutions. WWW 2019

[ABP+14] N. Anciaux, L. Bouganim, P. Pucheral, Y. Guo, L. L. Folgoc, S. Yin. Milo-DB: a personal, secure and portable database machine. Distributed and Parallel Databases, 32(1):37–63, 2014.

[ABP+19] N. Anciaux, L. Bouganim, P. Pucheral, I. Sandu Popa, G. Scerri: Personal Database Security and Trusted Execution Environments: A Tutorial at the Crossroads. Proc. VLDB Endow. 2019

[AEJ+15] A. Arasu, K. Eguro, M. Joglekar, R. Kaushik, D. Kossmann, R. Ramamurthy: Transaction processing on confidential data using cipherbase. ICDE 2015: 435-446

[AEK+14] A. Arasu, K. Eguro, R. Kaushik, R. Ramamurthy: Querying encrypted data. SIGMOD Conference 2014: 1259-1261

[AK13] A. Arasu, R. Kaushik: Oblivious Query Processing. ICDT 2014.

[ALS+15] N. Anciaux, S. Lallali, I. Sandu Popa, P. Pucheral: A Scalable Search Engine for Mass Storage Smart Objects. PVLDB 8(9): 910-921 (2015)

[ANS13] N. Anciaux, B. Nguyen, I. Sandu Popa: Personal Data Management with Secure Hardware: How to Keep Your Data at Hand. MDM (2) 2013: 1-2

[ANS14] N. Anciaux, B. Nguyen, I. Sandu Popa: Tutorial: Managing Personal Data with Strong Privacy Guarantees. EDBT 2014: 672-673

References (2)

[BBB+17] R. Bahmani, M. Barbosa, F. Brasser, B. Portela, A.-R. Sadeghi, G. Scerri, B. Warinschi: Secure Multiparty Computation from SGX. Financial Cryptography 2017: 477-497

[BEE+17] J. Bater, G. Elliott, C. Eggen, S. Goel, A. Kho, J. Rogers: SMCQL: secure querying for federated databases. PVLDB 2017

[BGC+18] V. Bindschaedler, P. Grubbs, D. Cash, T. Ristenpart, V. Shmatikov: The tao of inference in privacy-protected databases. PVLDB 2018

[BPS+16] M. Barbosa, B. Portela, G. Scerri, B. Warinschi: Foundations of Hardware-Based Attested Computation and Application to SGX. EuroS&P 2016: 245-260

[BS11] S. Bajaj, R. Sion: TrustedDB: a trusted hardware-based database with privacy and data confidentiality. SIGMOD Conference 2011: 205-216

[CSS+22] Jinhua Cui, Shweta Shinde, Satyaki Sen, Prateek Saxena, Pinghai Yuan: Dynamic Binary Translation for SGX Enclaves. ACM Trans. Priv. Secur. 25(4): 32:1-32:40 (2022)

[DSC+15] T. T. A. Dinh, P. Saxena, E. Chang, B. C. Ooi, C. Zhang: M2R: Enabling Stronger Privacy in MapReduce Computation. USENIX Security 2015

[EZ17] S. Eskandarian, M. Zaharia: An oblivious general-purpose SQL database for the cloud. CoRR, abs/1710.00458, 2017

[FBB+18] B. Fuhry, R. Bahmani, F. Brasser, F. Hahn, F. Kerschbaum, A.-R. Sadeghi: HardIDX: Practical and secure index with SGX in a malicious environment. Journal of Computer Security 26(5): 677-706 (2018)

[HH21] Z. Han and H. Hu. 2021. ProDB: A memory-secure database using hardware enclave and practical oblivious RAM. Information Systems 96 (2021).

[HZX18] T. Hunt, Z. Zhu, Y. Xu, S. Peter, E. Witchel: Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data. ACM Trans. Comput. Syst. 35(4): 13:1-13:32 (2018)

References (3)

[LAP+19] R. Ladjel, N. Anciaux, P. Pucheral, G. Scerri. Trustworthy Distributed Computations on Personal Data Using Trusted Execution Environments. TrustCom, 2019.

[LAS+17] S. Lallali, N. Anciaux, I. Sandu Popa, P. Pucheral: Supporting secure keyword search in the personal cloud. Inf. Syst. 72: 1-26 (2017)

[LSB19a] J. Loudet, I. Sandu Popa, L. Bouganim: SEP2P: Secure and Efficient P2P Personal Data Processing. EDBT 2019.

[LSB19b] J. Loudet, I. Sandu-Popa, L. Bouganim. DISPERS: Securing Highly Distributed Queries on Personal Data Management Systems. PVLDB 2019

[LWG+13] S. Lee, E.L. Wong, D. Goel, M. Dahlin, V. Shmatikov, πbox: A platform for privacy-preserving apps, in: NSDI, 2013.

[MPC+18] P. Mishra, R. Poddar, J. Chen, A. Chiesa, R. A. Popa: Oblix: An Efficient Oblivious Search Index. S&P 2018.

[MSW+14] Y-A. de Montjoye, E. Shmueli, SS. Wang, AS. Pentland: OpenPDS: Protecting the Privacy of Metadata through SafeAnswers. PLoS ONE 9(7) 2014

[MZC+16] R. Mortier, J. Zhao, J. Crowcroft, L. Wang, Q. Li, H. Haddadi, Y. Amar, A. Crabtree, J. Colley, T. Lodge, T. Brown, D. McAuley, C. Greenhalgh: Personal Data Management with the Databox: What's Inside the Box? ACM CoNEXT Cloud-Assisted Networking workshop, 2016

[OCF+15] O. Ohrimenko, M. Costa, C. Fournet, C. Gkantsidis, M. Kohlweiss, D.Sharma: Observing and Preventing Leakage in MapReduce. CCS 2015.

References (4)

[OSF+16] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, M. Costa: Oblivious Multi-Party Machine Learning on Trusted Processors. USENIX Security 2016.

[PGF+17] R. Pires, D. Gavril, P. Felber, E. Onica, M. Pasin: A lightweight MapReduce framework for secure processing with SGX. CCGrid 2017

[PML+19] C. Priebe, D. Muthukumaran, J. Lind, H. Zhu, S. Cui, V. A. Sartakov, P. R. Pietzuch: SGX-LKL: Securing the Host OS Interface for Trusted Execution. CoRR abs/1908.11143 (2019)

[PVC18] C. Priebe, K. Vaswani, M. Costa: EnclaveDB: A Secure Database Using SGX. IEEE Symposium on Security and Privacy 2018: 264-278

[RHM19] L. Roche, J. M. Hendrickx, Y-A. de Montjoye: Estimating the success of re-identifications in incomplete datasets using generative models. Nature Communications 2019

[SCF+15] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, M. Russinovich: VC3: Trustworthy Data Analytics in the Cloud Using SGX. S&P 2015

[TAP17] P. Tran-Van, N. Anciaux, P. Pucheral: SWYSWYK: A Privacy-by-Design Paradigm for Personal Information Management Systems. ISD 2017

[TCL+19] Y. Tang, J. Chen, K. Li, J. Xu, Q. Zhang: Authenticated Key-Value Stores with Hardware Enclaves. CoRR abs/1904.12068 (2019)

[UCB+22] H. Unnibhavi, D. M. Cerdeira, A. Barbalace, N. Santos, P. Bhatotia: Secure and Policy-Compliant Query Processing on Heterogeneous Computational Storage Architectures. Sigmod 2022

[WAK18] N. Weichbrodt, P.-L. Aublin, R. Kapitza: SGX-perf: A Performance Analysis Tool for Intel SGX Enclaves. Middleware 2018

[WMSG19] S. Weiser, L. Mayr, M. Schwarz, Daniel Gruss: SGXJail: Defeating Enclave Malware via Confinement. RAID 2019.

[ZDB+17] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, I. Stoica. Opaque: An oblivious and encrypted distributed analytics platform. NSDI 2017