Tutorial Outline

PART I. Personal Data Management Systems (PDMS)

Review of functionalities & addressed privacy threats

Individual's PDMS vs (corporate) DBMS and main properties to achieve

PART II. TEE-based Data Management

The promises of Trusted Execution Environments (TEEs)

A review of privacy-preserving data management using TEEs

PART III. Bridging the Gap between PDMS and TEEs

Towards a reference logical PDMS architecture

Reusing building blocks from the existing TEE-based solutions

A quick view of remaining challenges

Definition of an Extensive and Secure PDMS (ES-PDMS)

Extensive & Secure

Extensive	Secure
provides the expected set of functionalities to cover the complete data life-cycle	*and is compliant with their respective security properties counterparts,*
data collection,	*piped data collection,*
storage and recovery,	*mutual data at rest protection,*
cross-computations,	*bilaterally trusted personal computation,*
collective computations,	*mutually trusted collective computation,*
data dissemination,	*controlled data dissemination.*

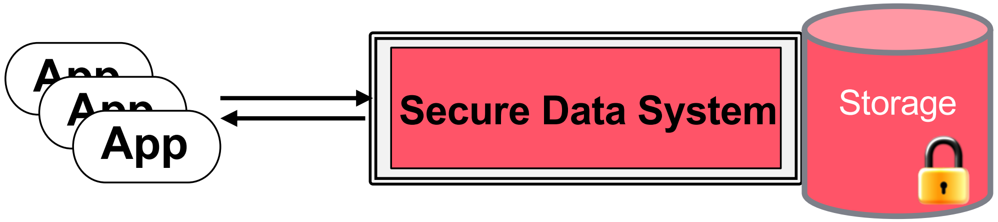
ES-PDMS problem ➔ typical *tension extensibility vs. security*

- **TEEs are a prime opportunity to alleviate this tension**
- **… but one needs to consider the *atypical context* of the Personal Cloud**
  - **Layman PDMS owner (and no DBA/DSA)**
  - **Completely open and hence untrusted system environment**
  - **Apps 'move' to data**
  - **Large scale collective computations among many individuals unknown to each other**
  - **Unclear legal responsibility of the PDMS owner w.r.t. third party data**
  - **…**

Logical Architecture : Three Layers [ABB+19]

ES-PDMS problem ➔ tension extensibility – security

- In practice : interesting processing is App-specific ➔ privacy violations are App-specific…
- ➔ How to avoid data leaks and launch advanced data processing?



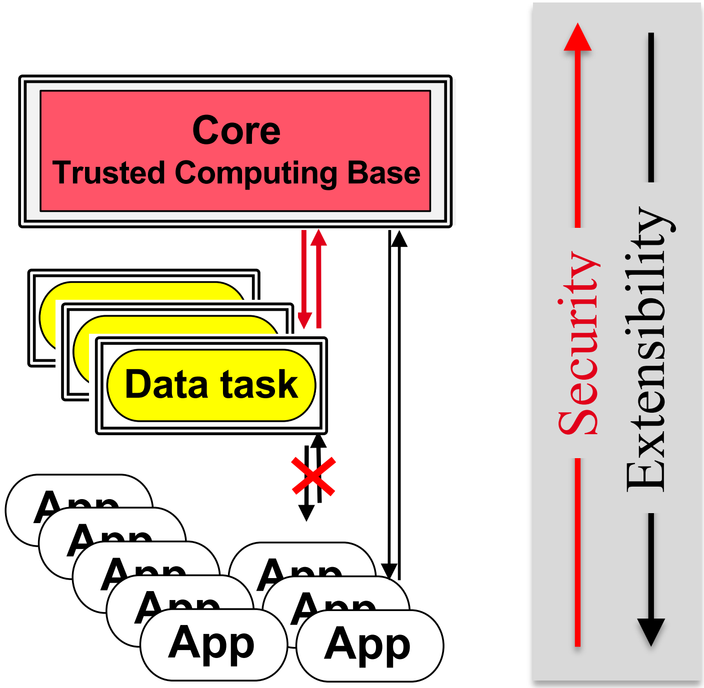
Logical Architecture : Three Layers [ABB+19]

ES-PDMS problem ➔ tension extensibility – security

- In practice : interesting processing is App-specific ➔ privacy violations are App-specific…
- ➔ How to avoid data leaks and launch advanced data pr

Three-layer architecture

- **Core (limited and secure)**
  - Trusted Computing Base (TCB) – small and (ideally) proven
  - Data Storage, Policy enforcement, Communication
- **Data tasks (advanced and isolated/sandboxed)**
  - Untrusted code – potentially large
  - Deal with (complex) app specific data management
- **Applications (Apps)**
  - No trust assumptions can be made (today)
  - Manipulate results (but not raw data)



Logical Architecture : Three Layers [ABB+19]

ES-PDMS problem ➔ tension extensibility – security

In practice : interesting processing is App-specific ➔ privacy violations are App-specific…

➔ How to avoid data leaks and launch advanced data pro…

Reuse building blocks from the existing TEE-based solutions to implement the core
- e.g., EnclaveDB[PVC18], HardIDX[FBB+18], secure KVS[TCL+19], Oblix[MPC+18]…
- e.g., for secure and efficient DB logging/recovery, base query processing and indexing, protection against side-channels attacks…

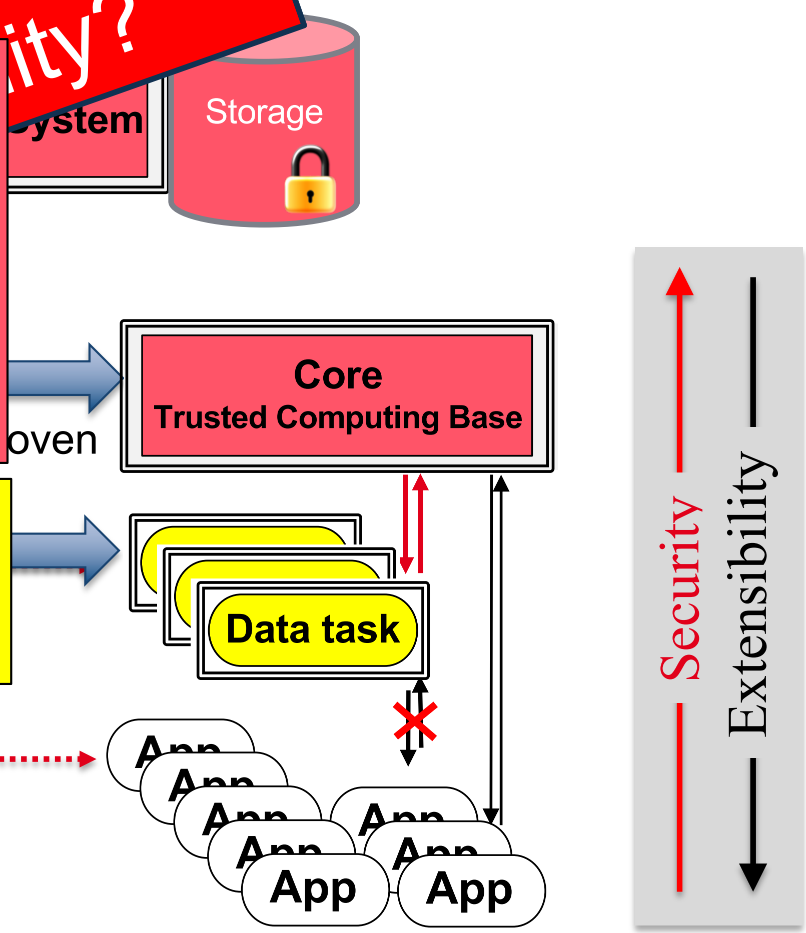
Reuse building blocks from the existing TEE-based solutions to secure the Data tasks (see next)

Deal with (complex) app specific data management

Applications (Apps)

No trust assumptions can be made (today)

Manipulate results (but not raw data)



Satisfying *bilaterally trusted data computations* property

Assumptions

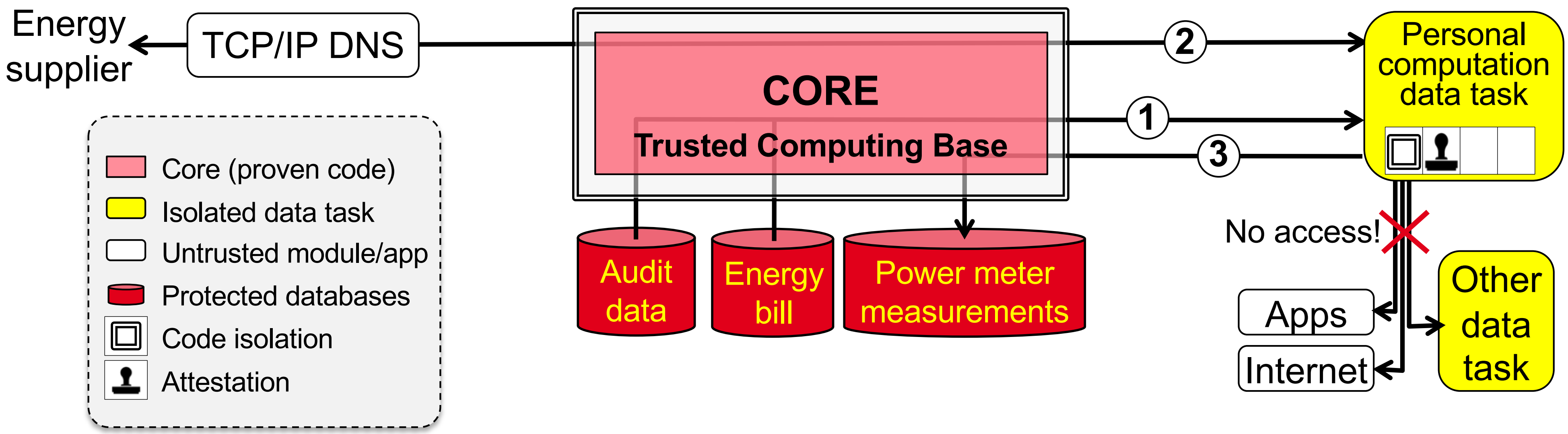
Execute any arbitrarily complex but untrusted computation code with access to some (large amounts of) PDMS raw data

Requirements

Computation code only accesses required raw data, only the result is shared and attested

➔ Manifest: collection rules + computation code + 3rd party accessing the result

➔ Data task runs computation code (▢, attestation) + result declassification by the Core



Satisfying *bilaterally trusted data computations* property

Assumptions

Execute any arbitrarily complex but untrusted computation code with access to some (large amounts of) PDMS raw data

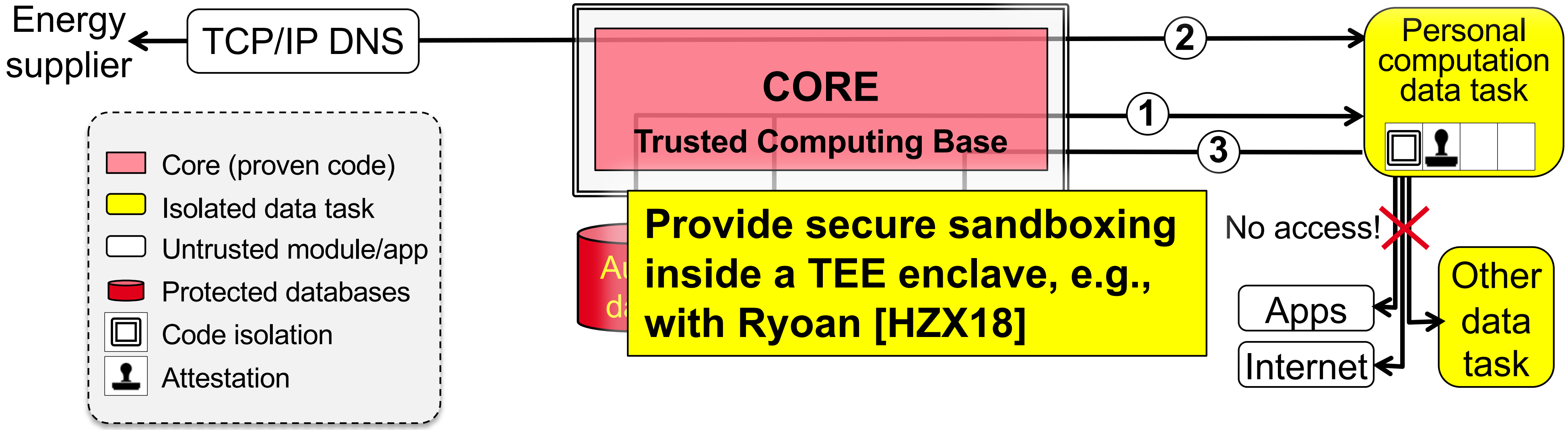
Requirements

Computation code only accesses required raw data, only th

➔ Manifest: collection rules + computation code + 3rd part

➔ Data task runs computation code (▢, attestation) + result decla

Control several data tasks used simultaneously in a computation, e.g., such as in πBox[LWG+13]



Provide secure sandboxing inside a TEE enclave, e.g., with Ryoan [HZX18]

Remaining Challenges: Architectural level

Design a minimal and provable Core engine

- Minimal (in code size & complexity) set of modules
- Algebra of operators that cannot be delegated to Data tasks
- Support different data models vs. deal with data models/optimizations at Data task level?
- Design provably secure components

Establish a continuum of solutions (modular approach)

- How to map our logical architecture on different physical ones?
- How to avoid (re-)proving Core modules on different HW targets (SE, TrustZone, SGX…)?

Remaining Challenges: Architectural level (cont.)

- **Full (both ways) Data task isolation ➔ *stateless data tasks***
  - TEEs offer code/data confidentiality inside the enclave but assume code is trusted!
  - In the PDMS context, the Data tasks are untrusted ➔ require another level of isolation
  - Enforcing reverse isolation requires additional in-enclave sandboxing (e.g., Ryoan [HZX18]) ➔ Stateless data tasks, i.e., no access to persistent storage and one-shot at input data
- **Joint (asymmetrical) data management between Core and Data tasks ➔ *require redesigning traditional data access methods***
  - Core can only support basic, generic data management operations (e.g., in the spirit of a KVS) and has access to storage
  - Data tasks implement advanced data processing techniques but require to map them to the Core API for persistence

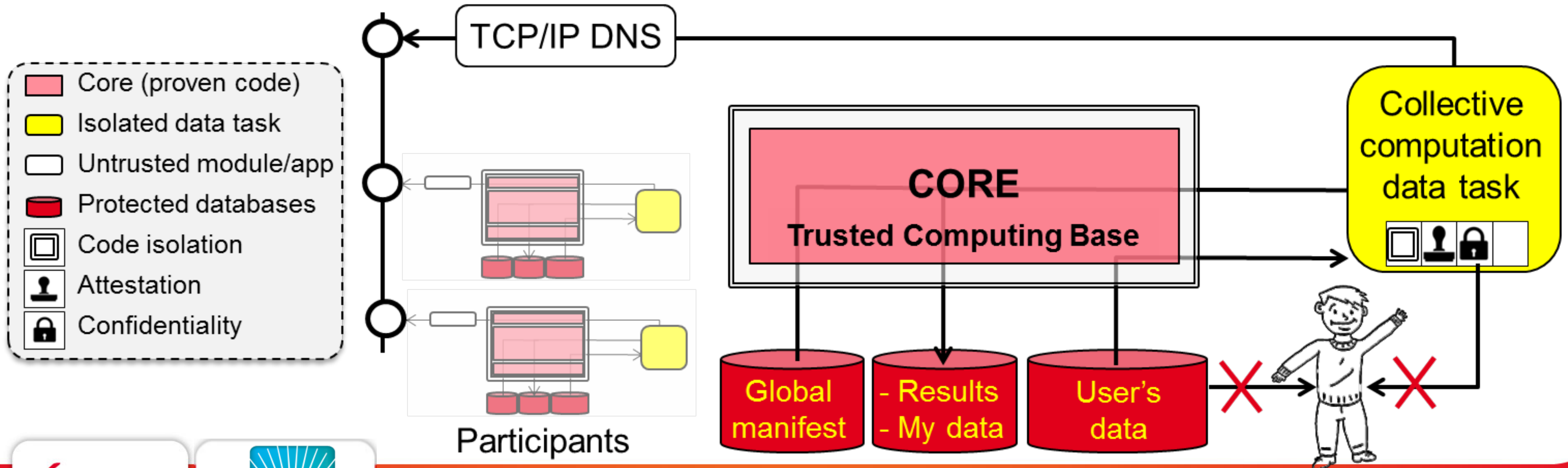
Satisfying *mutually trusted collective computations* property

Assumptions

- Distributed computation code is arbitrary (and untrusted), with access to *n* PDMS raw data
- The number of participants can be huge
- Confidentiality (🔒) of a (small) subset of PDMS can be broken by their owners (colluding)

Requirements

- Computation code only accesses needed raw data, only the result is shared, it is attested
  - ➔ Global Manifest: collection rules + computation code + 3rd party accessing the result
  - ➔ *Local validity*: each participant checks *locally* that *all* others behave honestly
  - ➔ *Global integrity*: certify through attestation the output of each data task for incremental integrity checks
  - ➔ *Side-channel attacks resiliency*: (1) circumscribe leakage and (2) prevent targeting



Building blocks from [LAP+19, LSB19a]

Satisfying *mutually trusted collective computations* property

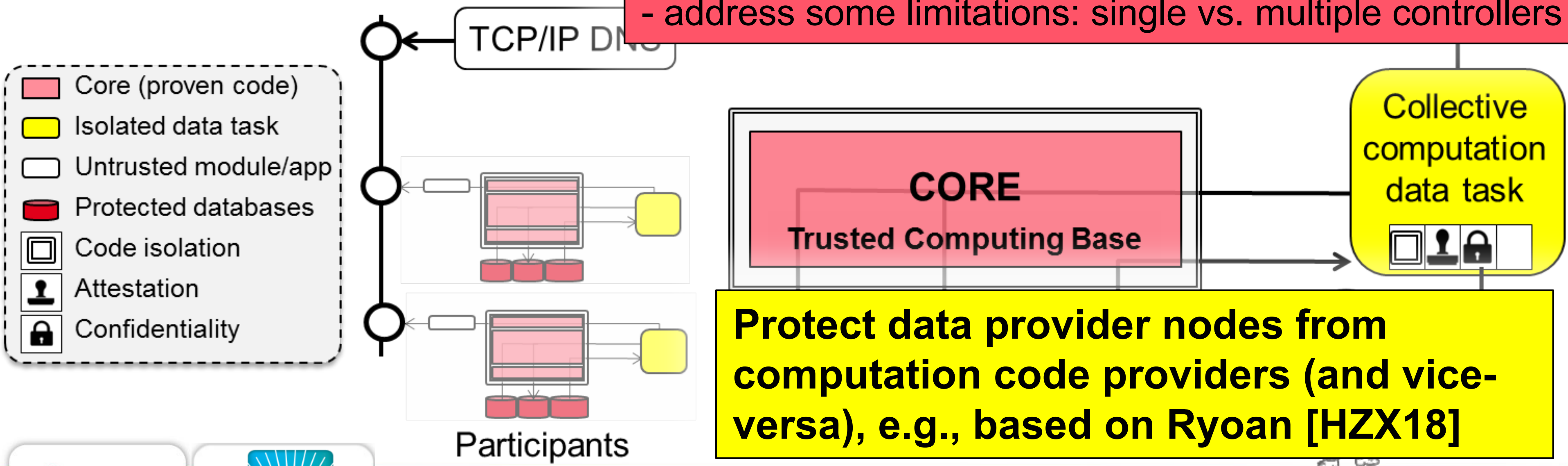
Assumptions

- Distributed computation code is arbitrary (and untrusted), with access to *n* PDMS raw data
- The number of participants can be huge
- Confidentiality (🔒) of a (small) subset of PDMS can be broken by their owners (colluding)

Requirements

- Computation code only accesses needed raw data, only the result is shared, it is attested
  - ➔ Global Manifest: collection rules + computation code + 3rd party accessing the result
  - ➔ *Local validity*: each participant
  - ➔ *Global integrity*: certify through
    integrity checks
  - ➔ *Side-channel attacks resiliency*

Reusable building blocks to achieve integrity and confidentiality from multiple TEEs, e.g., VC3, M2R…
- address some limitations: single vs. multiple controllers



Protect data provider nodes from computation code providers (and vice-versa), e.g., based on Ryoan [HZX18]

Remaining Challenges: Distributed data management level

Guarantees against confidentiality attacks from covert (colluding) adversaries in large PDMS networks*

- Which network overlay to allow for efficient, scalable and <u>secure</u> node selection and distributed data indexing?
- How to efficiently prevent a (large colluding) attacker to influence a query execution?
- How to efficiently execute a query while minimizing the private data leakage risks?
- **DISPERS: Securing Highly Distributed Queries on Personal Data Management Systems [LSB19b]*

Leakage through communication patterns

- Making the dataflow of a distributed computation data independent is costly (broadcasts…)
- Find appropriate, efficient solutions to anonymize the communications between peers (e.g., in-network proxies, differentially private data exchanges, mixer nodes, …)

Confidentiality guarantees on the final result in a *fully distributed setting*

- Make sure that the final result is 'sufficiently' anonymous (e.g., the selected set of participants is sufficiently large and divers)
- Integrate privacy guarantees as part of the manifest (e.g., to let the nodes decide if the exposure risk is compliant with the user required privacy level)

Other remaining challenges: Administration level

Make the net effects of sharing policy viewable and easy to adjust

- How to integrate (faithful) visualization tools in Data tasks with peripheral isolation?
- How to validate net effects of decisions when the set of permissions is huge (e.g., [LWG+13])?
- How to identify suspicious grants (first step [TAP17])?
- Visualize the complete life-cycle of personal data while it may undergo transformations?
- How to visualize time or location-based contextual rules?

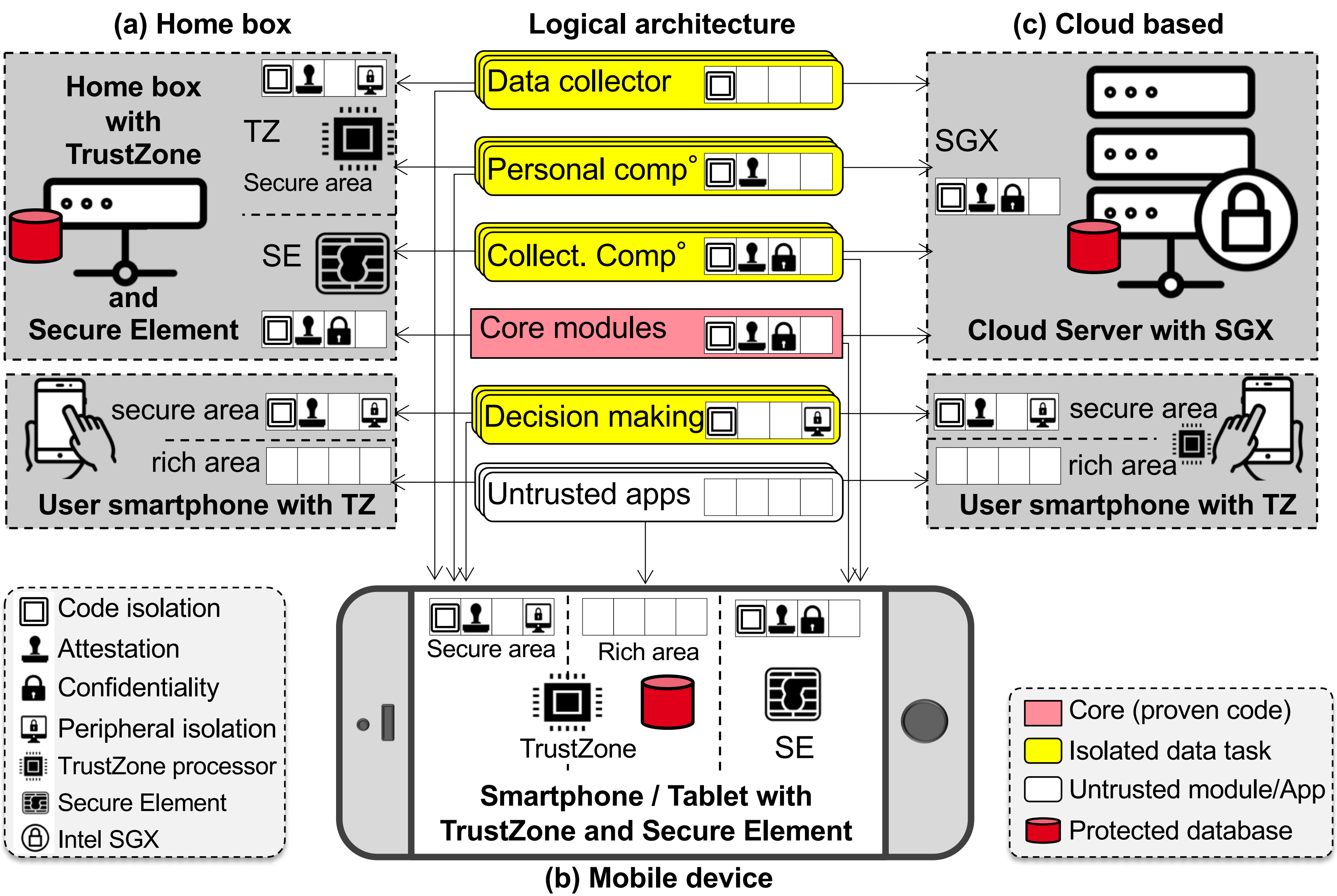
Design a trusted reference monitor to enforce ‘effects’ rather than ‘decisions’

- Define a model to capture the effects of policies in a simple/basic logic
- Visualization tools need to be executed as Data tasks providing *peripherals isolation*
- Materialize / maintain these effects in an efficient way

Design ‘zero-knowledge’ grants

- Model side-channels leaking personal data within (acceptable) sets of permission
- Design countermeasure, e.g., based on replayed enclaves on a “what if” basis

Which possible concrete PDMS instances ?



Conclusion

Personal Data Management Systems arrive at a rapid pace and provide us with novel tools to manage our own personal data

- But current existing solution lead to various & irreconcilable choices

There is stringent need for an Extensible & Secure PDMS (ES-PDMS)

- I.e., provide at least the minimum set of functionalities to cover the complete data life-cycle with trustworthy security guarantees

TEEs offer a great opportunity to reach the ES-PDMS goal

- But we need to look more closely at the specificities of the PDMS context

A first major step is to arrive at a reference logical architecture based on TEE

- A way out of the dilemma between the delegation of personal data to providers
- … and the myth of the owner's self-capacity to secure her own data

But the journey has only began and many challenges remain to be addressed!

… and the most critical ones primarily concern our research community

Conclusion

Personal Data Management Systems arrive at a rapid pace and provide us with novel tools to manage our own personal data

- But current existing solution lead to various & irreconcilable choices

There is stringent need for an Extensible & Secure PDMS (ES-PDMS)

- I.e., provide at least the minimum set of functionalities to cover the complete data life-cycle with trustworthy security guarantees

TEEs offer a great opportunity to reach the ES-PDMS goal

- But we need to look more closely at the specificities of the PDMS context

A first major step is to arrive at a reference logical architecture based on TEE

- A way out of the dilemma between the delegation of personal data to providers
- … and the myth of the owner's self-capacity to secure her own data

But the journey has only began and many challenges remain to be addressed!

… and the most critical ones primarily concern our research community

Conclusion

Personal Data Management Systems arrive at a rapid pace and provide us with novel tools to manage our own personal data

- But current existing solution lead to various & irreconcilable choi[illegible]

There is stringent need for an Extensible & Se[illegible]

- I.e., provide at least the minimum set of fun[illegible]plete data life-cycle with trustworthy security gu[illegible]

TEEs offer a great op[illegible]S-PDMS goal

- But we need to [illegible]ecificities of the PDMS context

A fi[illegible]ve at a reference logical architecture based on TEE

- A w[illegible]ma between the delegation of personal data to providers
- … an[illegible]th of the owner's self-capacity to secure her own data

Let's make it happen!

But the journey has only began and many challenges remain to be addressed!

… and the most critical ones primarily concern our research community

Thanks !

Questions ?

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.

[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

[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)

[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

[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)

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

[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