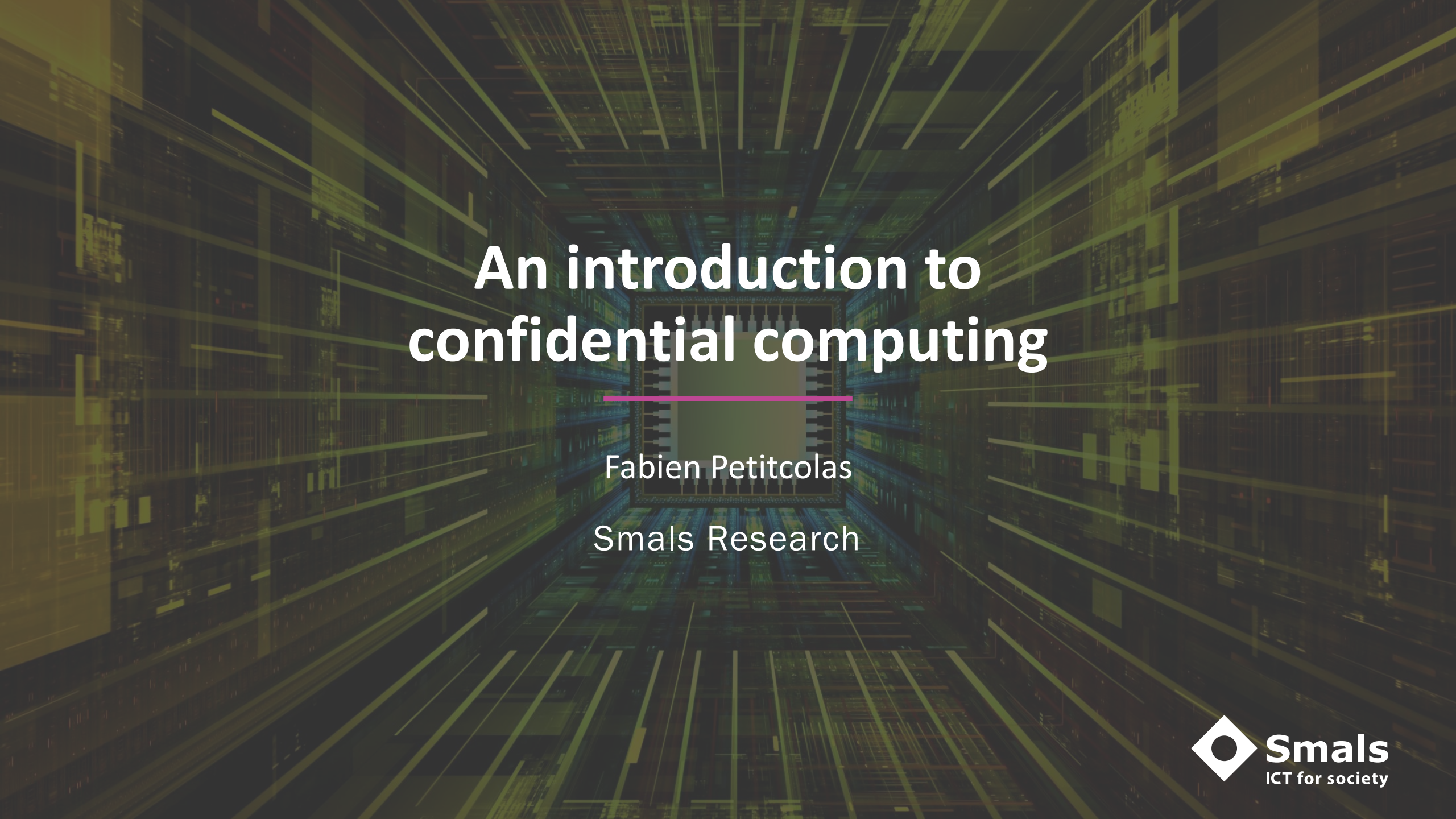


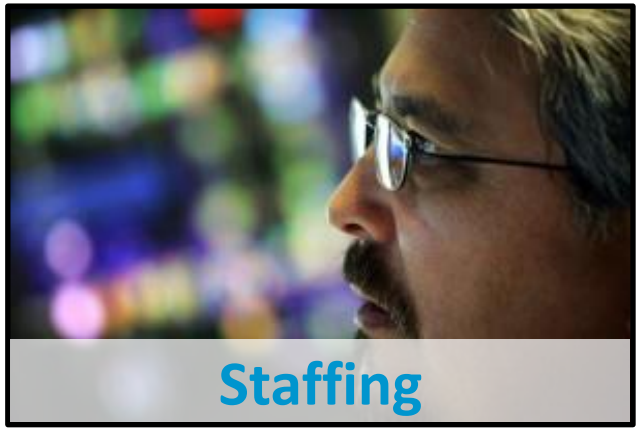
An introduction to confidential computing

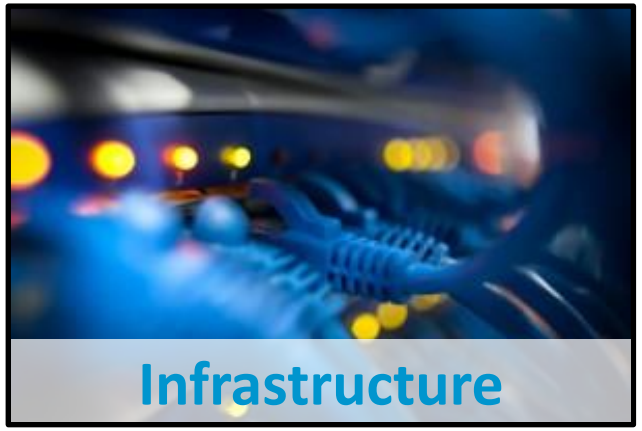
Fabien Petitcolas

Smals Research

Smals
ICT for society

Smals – Support for e-Government





g-cloud

16 YEARS!
top EMPLOYER BELGIË BELGIQUE BELGIUM 2024
CERTIFIED EXCELLENCE IN EMPLOYEE CONDITIONS

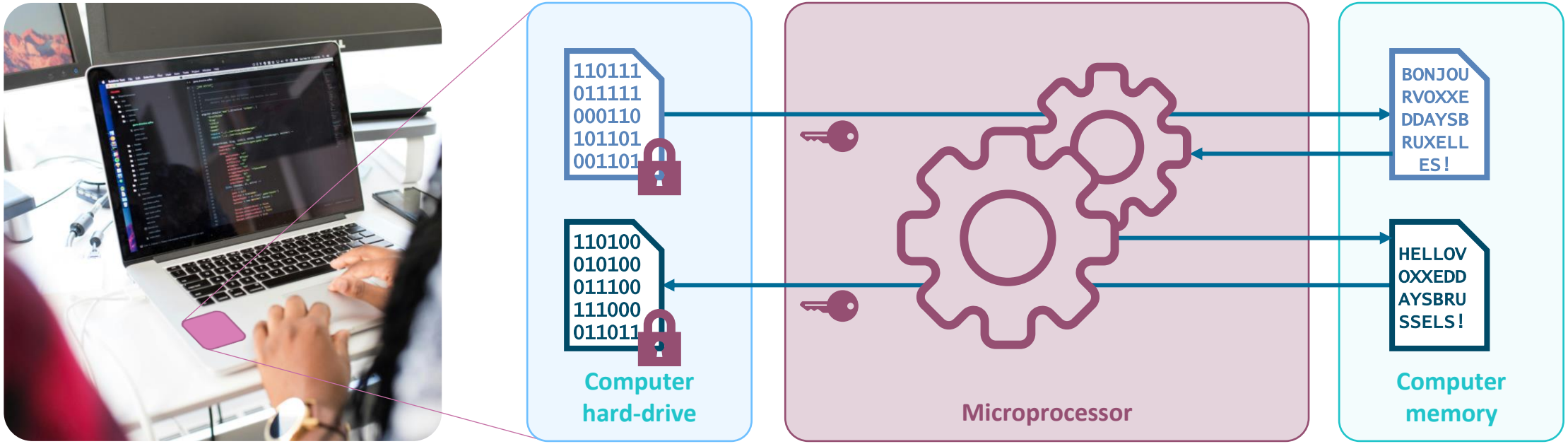
www.smals.be

Smals
ICT for society

Agenda

- Secure remote computation
- Homomorphic encryption
- Secure multiparty computation
- Trusted execution environments (TEE)
- Comparison of maturity
- Secured processors for TEE
- Conclusions and recommendations

Traditional computation on encrypted data



Smals
ICT for society

Basic remote computation

Concern: confidentiality, privacy (GDPR), sovereignty, etc. of data “in use”?

Not entirely trusted

Client

Computation request

Results of computation

Shared computing infrastructure

Smals
ICT for society

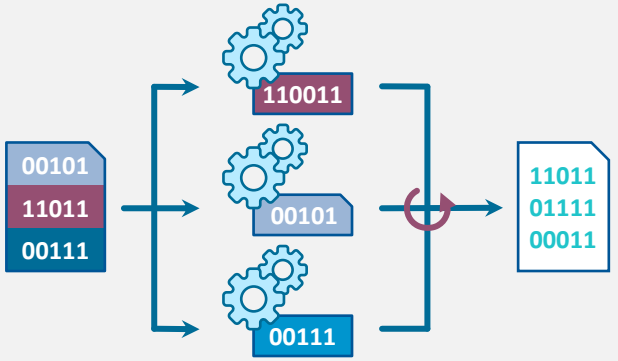
Main techniques for trusted remote computation

Aim: move the infrastructure provider outside of the trust boundary

Compute directly on encrypted data

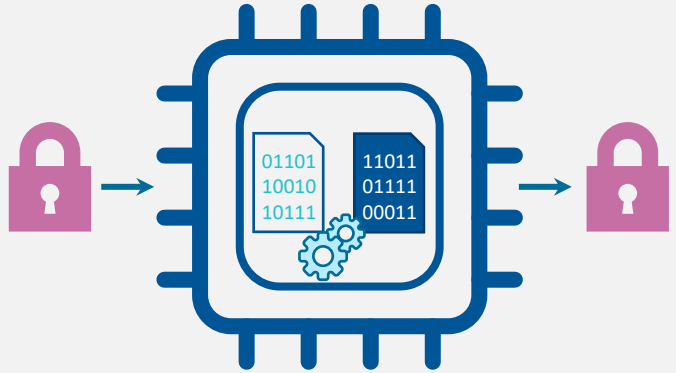
Homomorphic encryption (HE)

Split data and share computation between parties



Secure multiparty computation (MPC)

Compute inside “enclave”



Trusted execution environments (TEE)

Smals
ICT for society

Homomorphic encryption (HE)

Smals
ICT for society

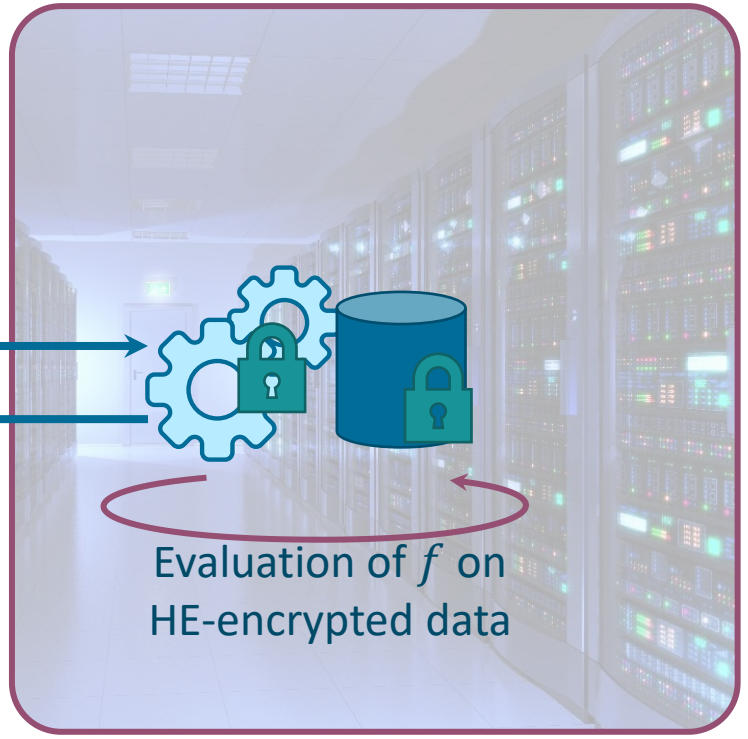
Homomorphic encryption: schematic overview

Client

Shared computing infrastructure

HE-encrypted data
function to evaluate (f)

HE-encrypted results

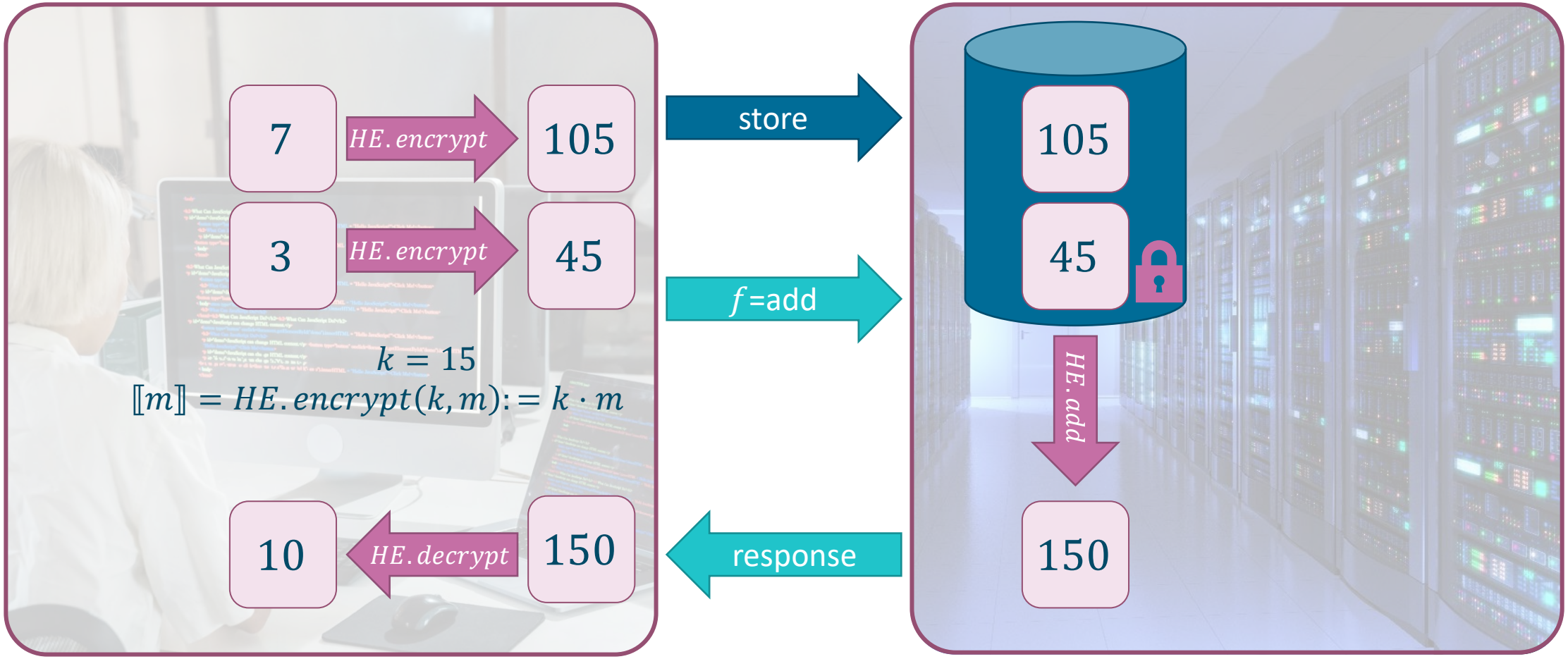


Depending on allowed complexity of f:

- Partial homomorphic encryption (PHE)
- Somewhat homomorphic encryption (SWHE)
- Fully homomorphic encryption (FHE)

Smals
ICT for society

Homomorphic encryption: trivial example



Advantages and limits of homomorphic encryption

Pros

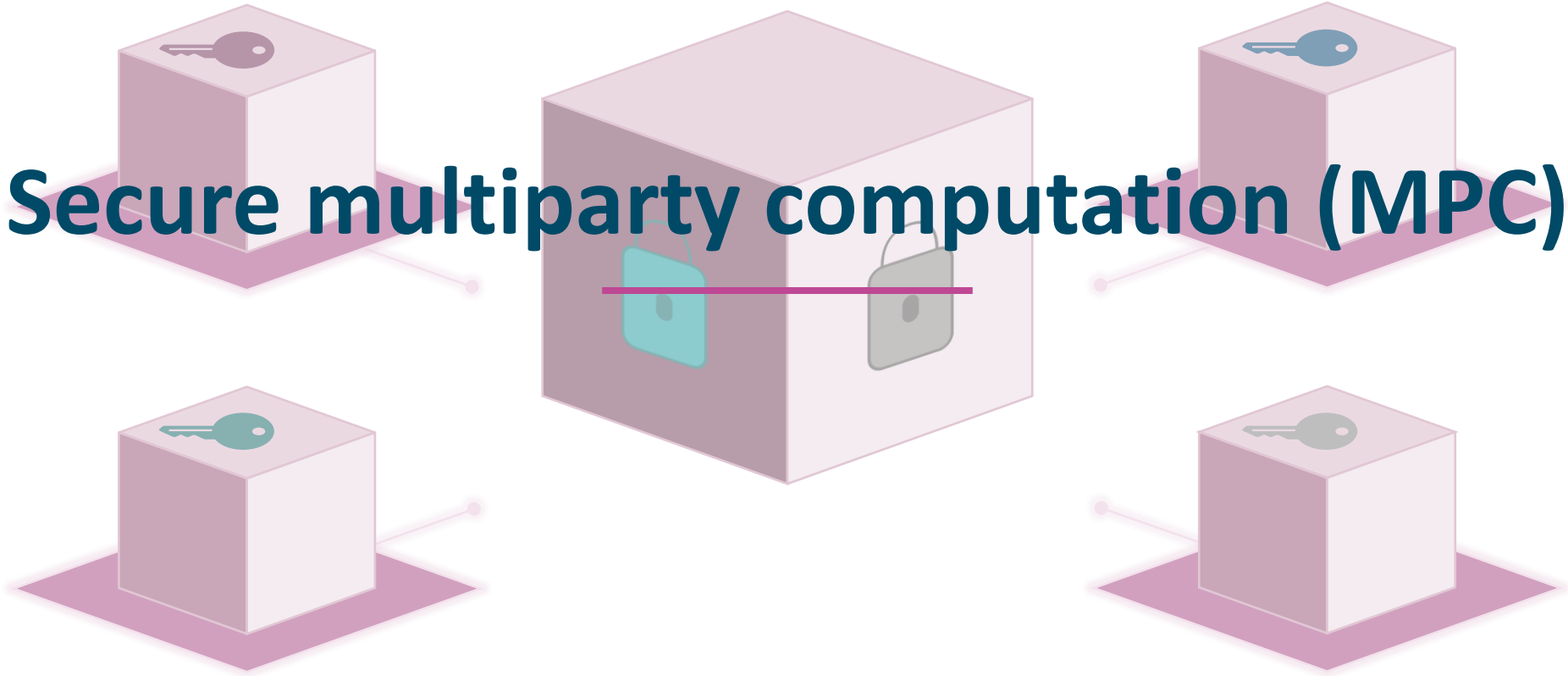
- Compute directly on encrypted data
- Security based on strong mathematical evidence under well defined assumptions
- Does not need special hardware
- Some schemes robust to post-quantum attacks
- Active research area

Cons

- Cryptography expert required to build protocol based on HE
- Computation not guaranteed
  - e.g., cannot check if $[\![m_1]\!] \oplus [\![m_2]\!]$ or $[\![m_1]\!] \ominus [\![m_2]\!]$ was computed
- High overhead:
  - Engineering cost: complex parametrisation, substantial changes required in application
  - Storage and bandwidth: large message expansion
  - Relatively low performance

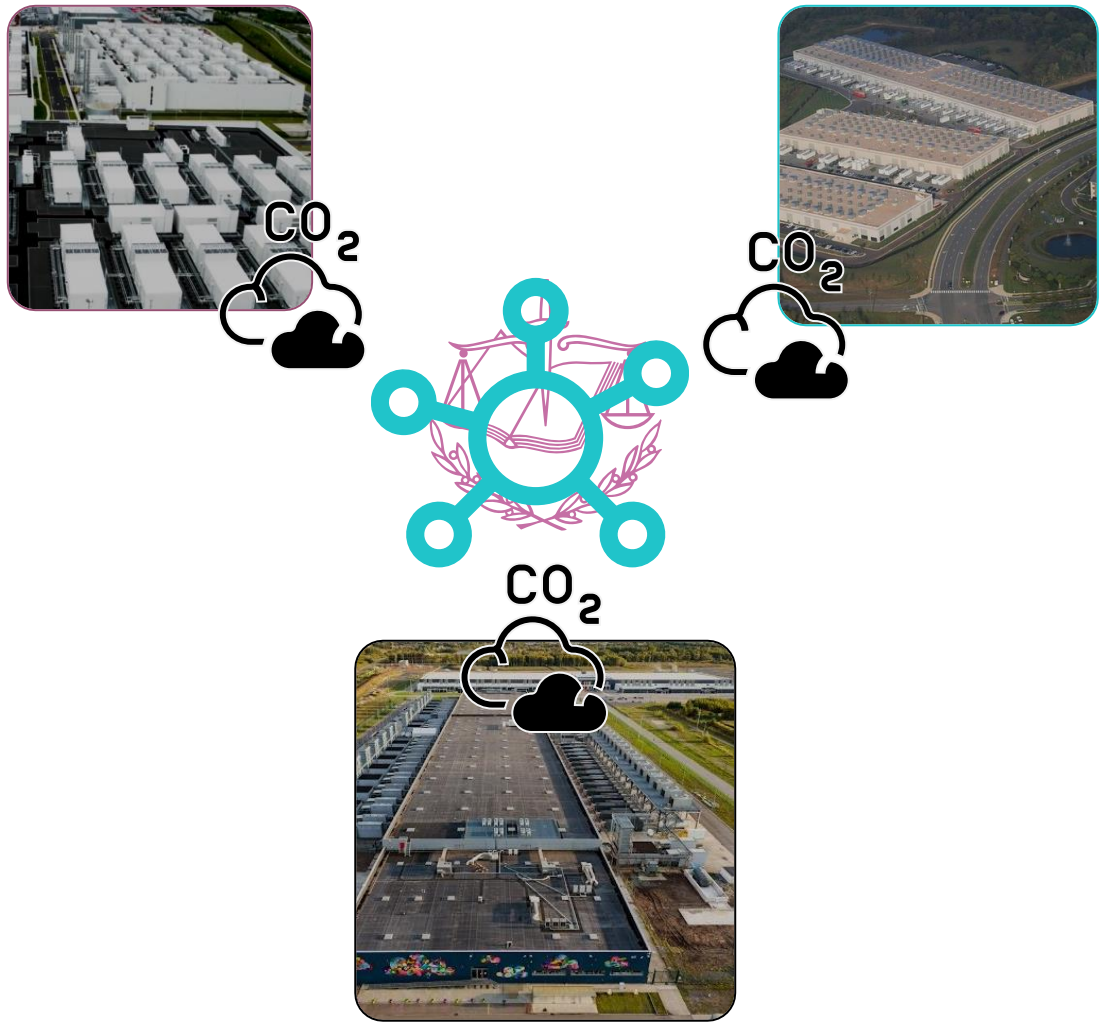
Smals
ICT for society

Secure multiparty computation (MPC)



Smals
ICT for society

MPC problem example



Three computing companies want to know which one of them has the lowest carbon footprint without revealing their respective values

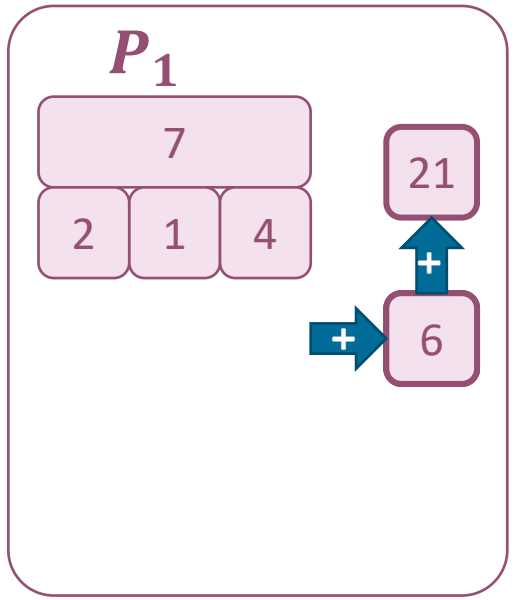
- Solutions:
  - Use a trusted party (hard to find)
  - Use multiparty computation (MPC): it enables mutually distrusting parties to compute an arbitrary function on their inputs.

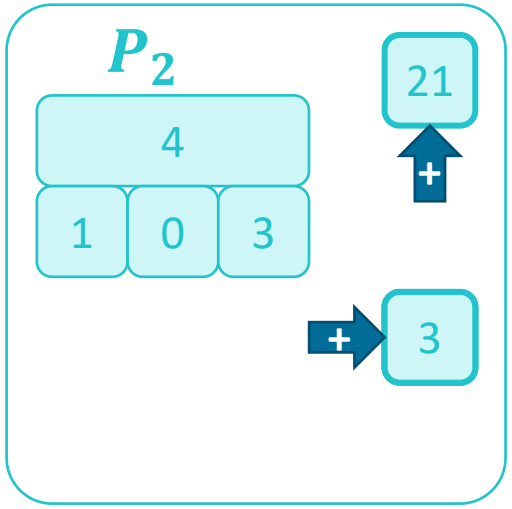
Smals
ICT for society

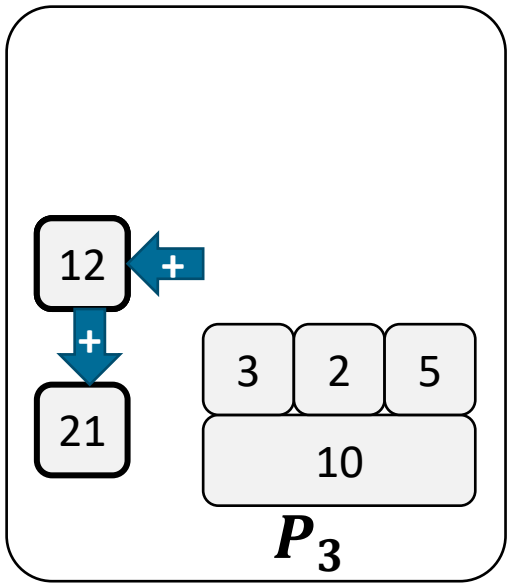
MPC problem and *a* solution

- **Problem**: n parties $P_1, \dots, P_n$ each have a secret input x_i and want to evaluate a function f in such a way that:
  - only the value $z = f(x_1, \dots, x_n)$ is learned
  - and nothing else is learned about $x_1, \dots, x_n$.
- **MPC solution example**:
  - Use arithmetic circuits to break down f into a composition of addition $(+)$ and multiplications $(\times)$
  - Each party follows a specific protocol:
    - Split input data into pieces and share pieces with other parties
    - Apply additions and multiplications on data shares locally (or with minimal interaction between parties)
    - Recombine partial results to get final result

MPC in action: simple addition example







1. Split input:
$x \rightarrow (x_1, x_2, x_3)$ s.t. $x_1 + x_2 + x_3 = x$

2. Share input

3. Local addition

4. Share partial results

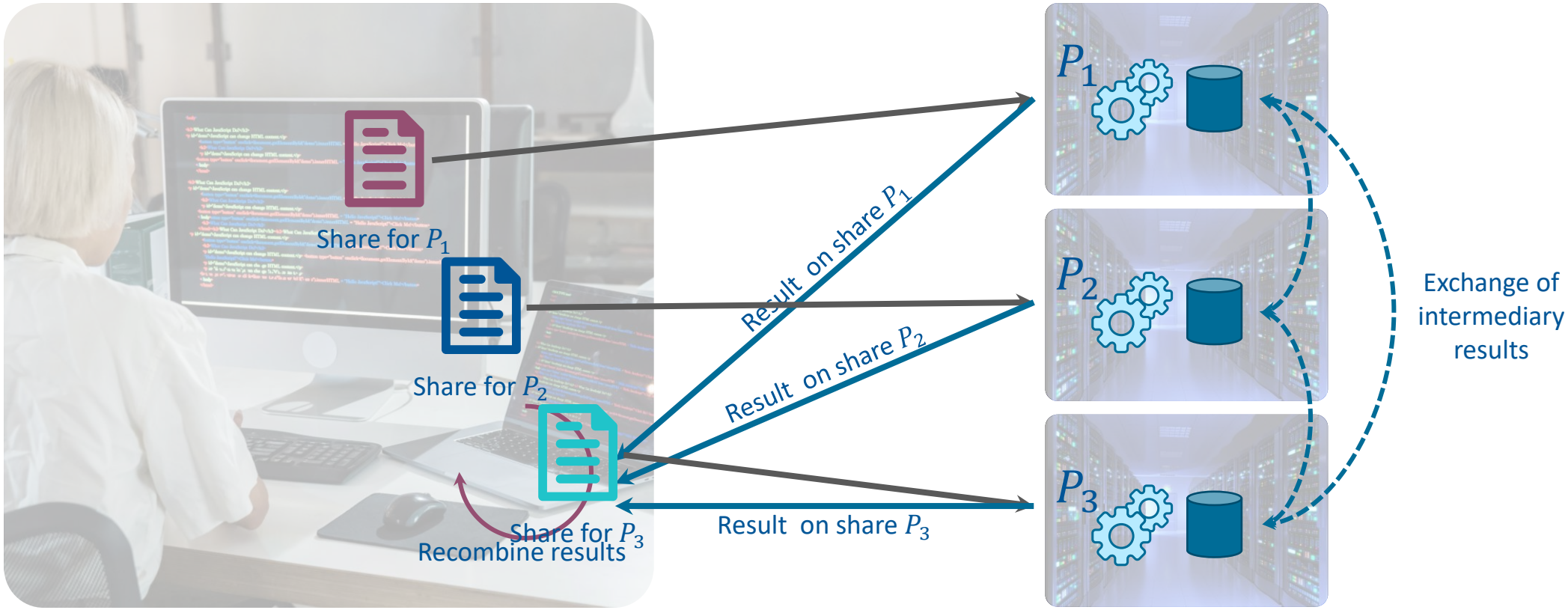
5. Local addition of final result

Smals
ICT for society

MPC deployment example

Client

Shared computing infrastructures



Smals
ICT for society

Advantages and limits of MPC

Pros

- Enable collaboration between untrusting parties
- Does not require central trusted party
- Cryptographic-based security
- Does not require special hardware
- Active research area

Cons

- Complexity of formally verifying protocol
- Complex setup and management
- Software rewriting required with highly specialised client-server software
- High communication cost between parties
- Small number of applications in production

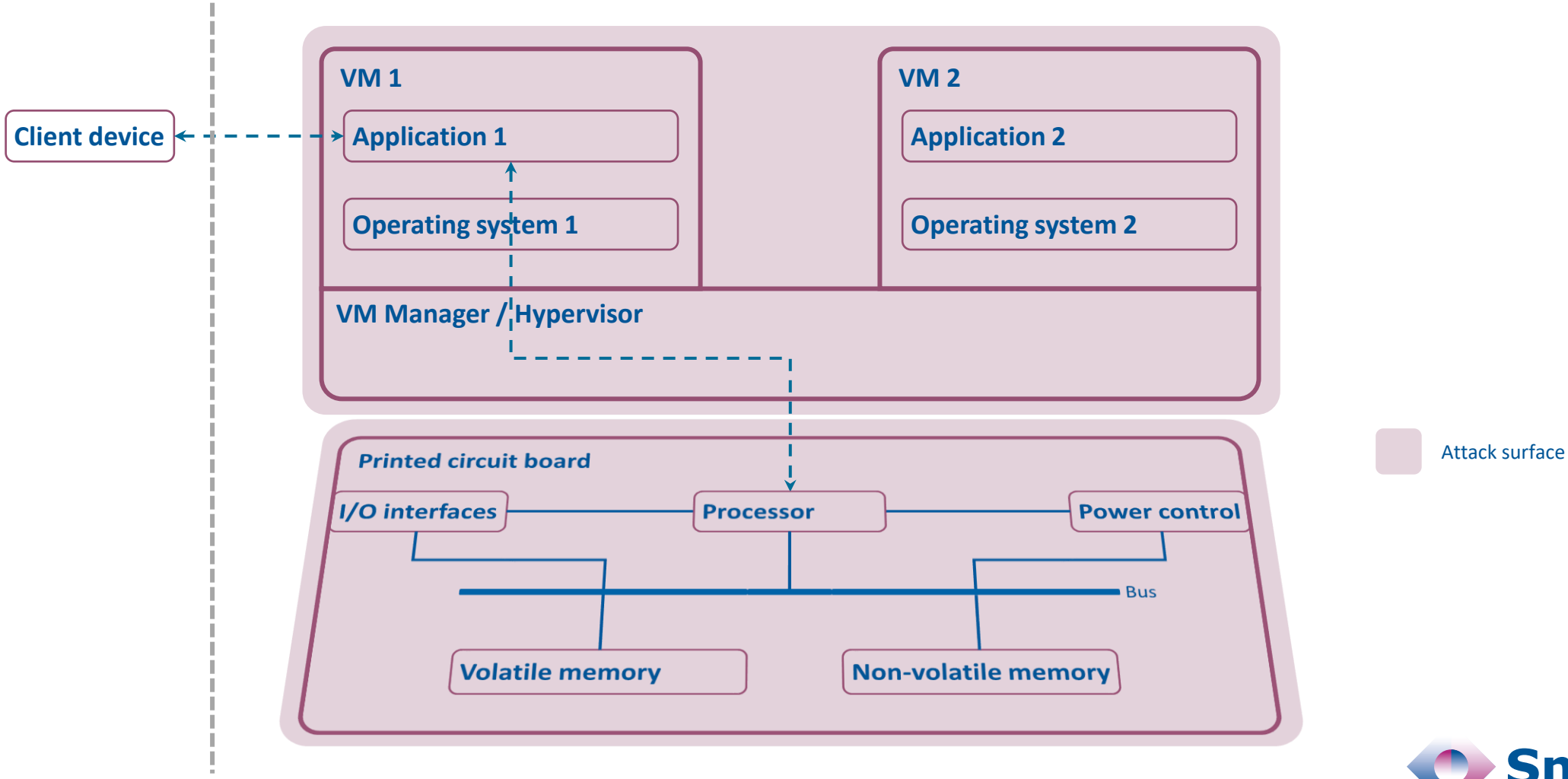
Trusted execution environments (TEE)

Smals
ICT for society

Hardware-based isolated execution

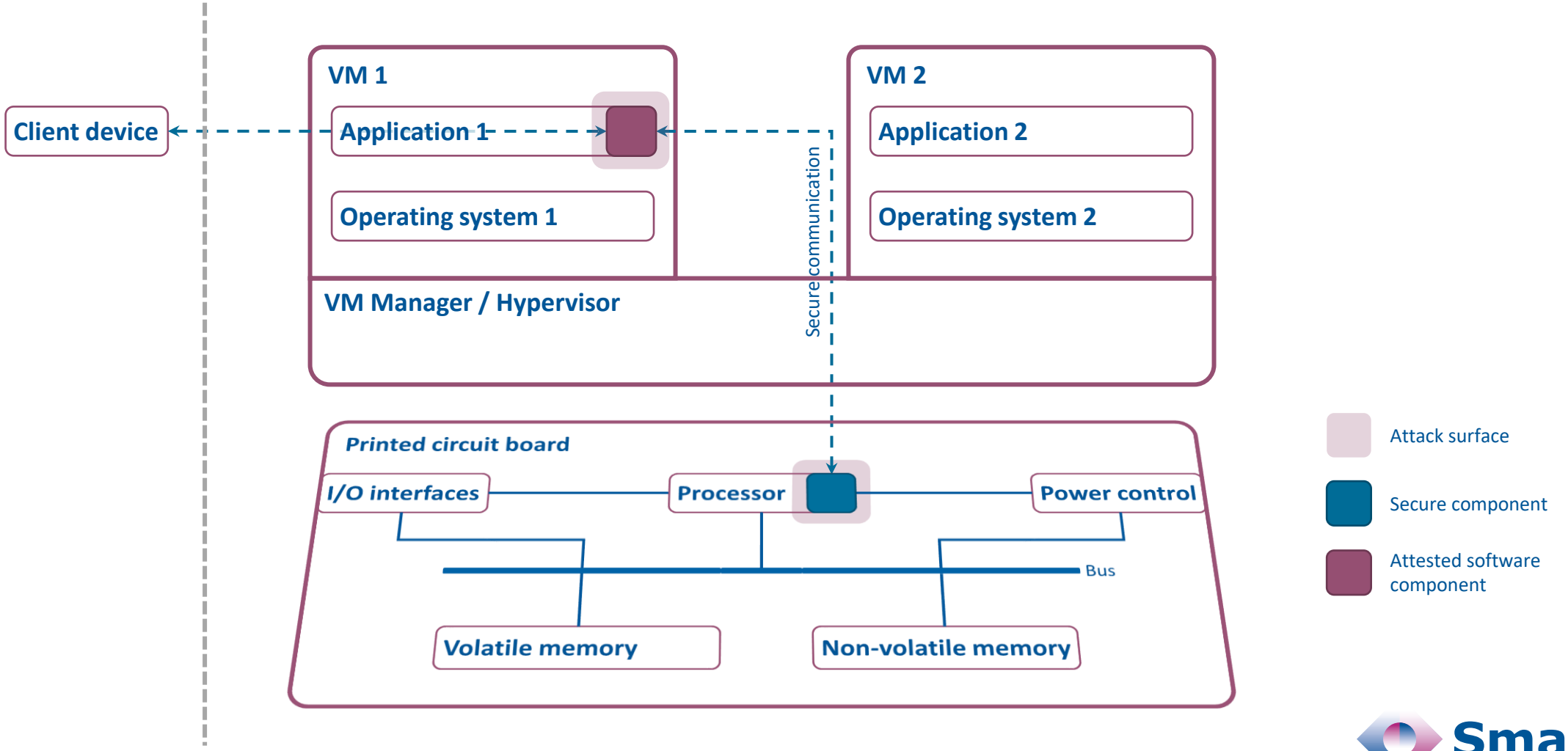
- **Technical goal**: better protect software processes from each other by creating isolated environments enforced by the hardware layer
- **Rational**: a system cannot be secure if its lowest layer (the hardware) is not
- **Requirements** for TEE:
  - Hardware root of trust to hold platform secrets
  - Reserved encrypted memory for trusted code and data
  - Encryption of all input/outputs
  - Evidence of authenticity and integrity

Generic architecture



Smals
ICT for society

Possible generic architecture with secure hardware



Smals
ICT for society

Verifying integrity of the system

Secure boot

From machine power-on to known secure state:

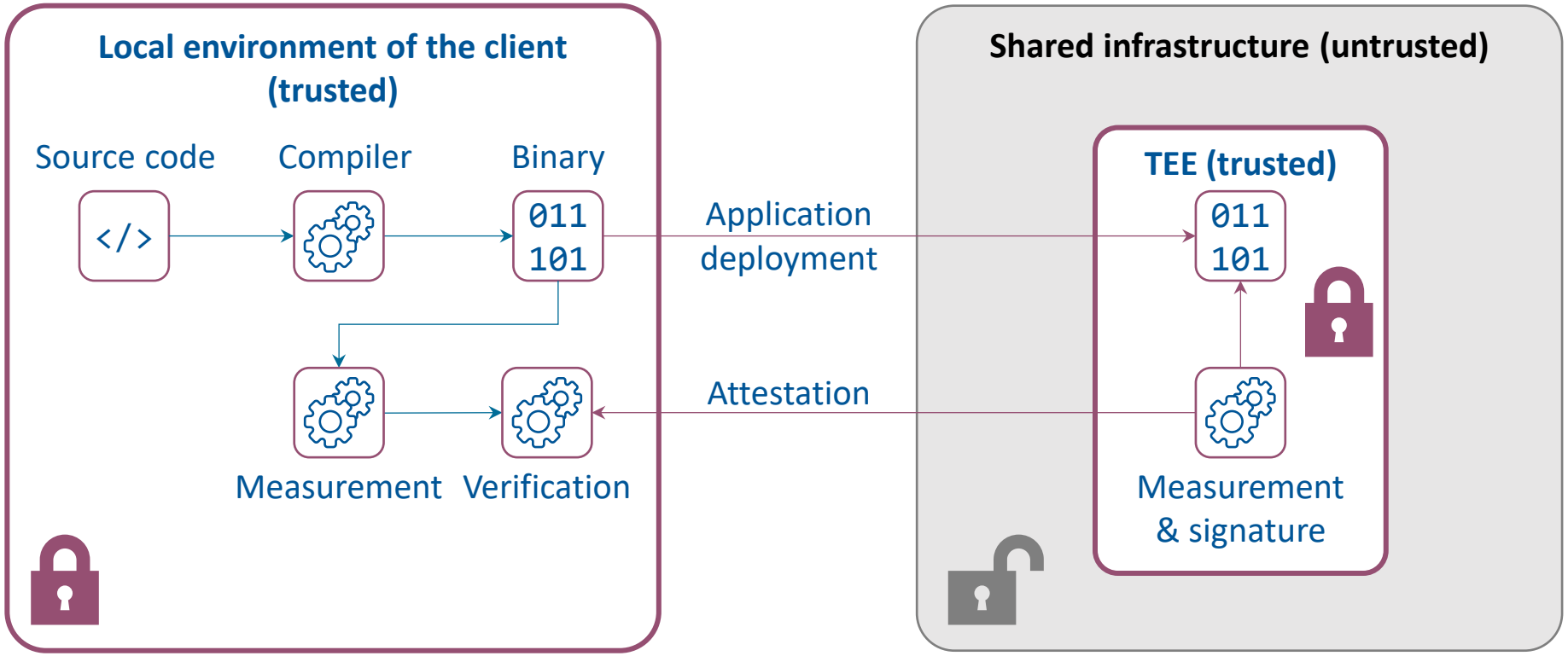
- Chain of trust from hardware to operating system software
- Each higher piece of firmware and software corresponds to what is expected by the lower component

Attestation

Signed evidence about remote system and state of software executing on it:

- Application runs on the expected hardware
- Executed binary is the expected application's binary
  - Should also correspond to the expected code

Attestation example



- Can be used to establish secret key with trusted application
- Can be used in security policies

Smals ICT for society

Advantages and limits of TEE

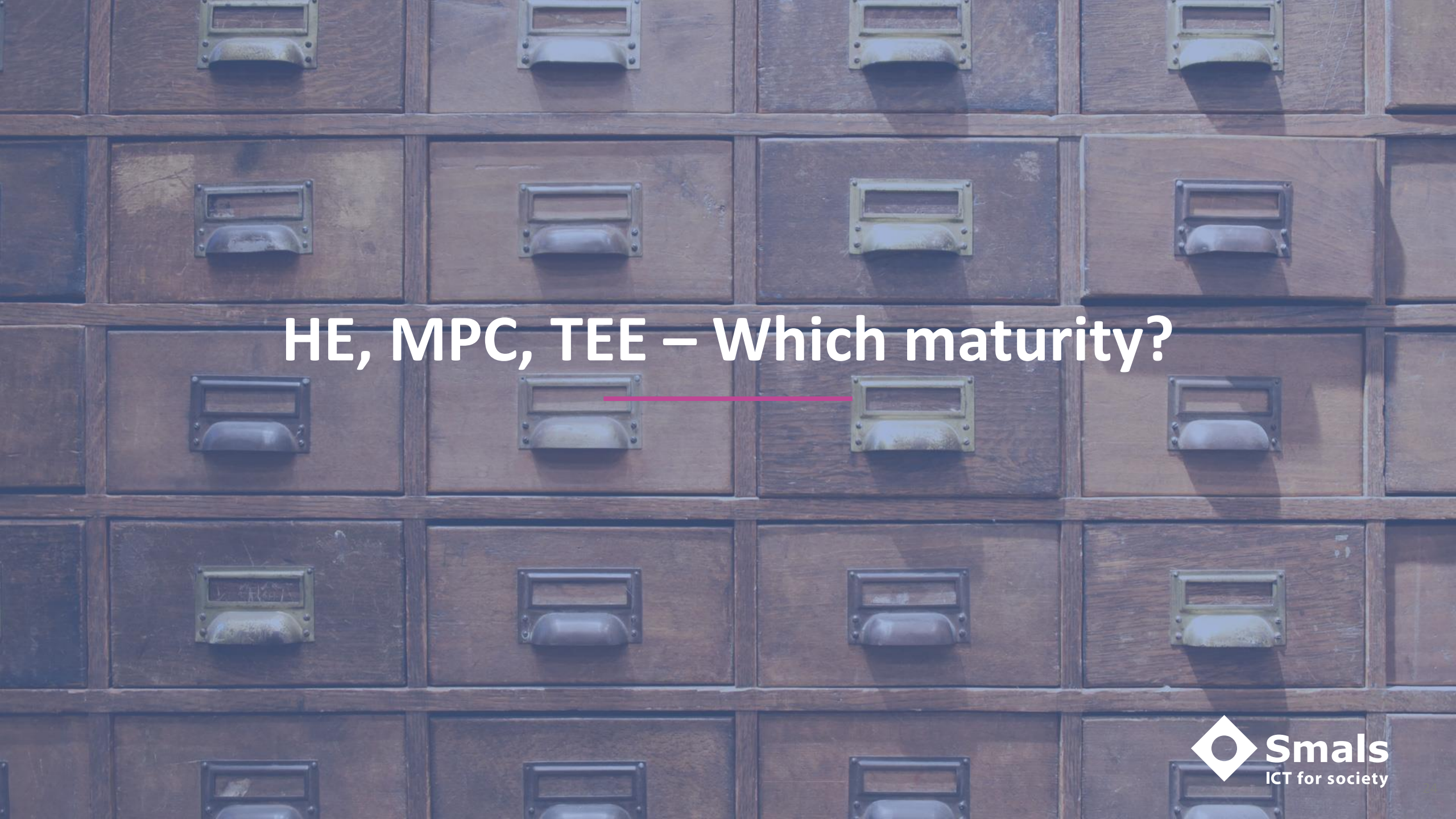
Pros

- Hardware-based security (trust the hardware manufacturer instead of the infrastructure provider)
- Available from main infrastructure providers
- Relatively simple application migration (containers, VM) compared to MPC and HE

Cons

- Requires specialised hardware
- Vulnerable to some physical and timing attacks
- Different abstractions could lead to vendor lock-in (no standard)
- May be impossible to verify attestation fully independently

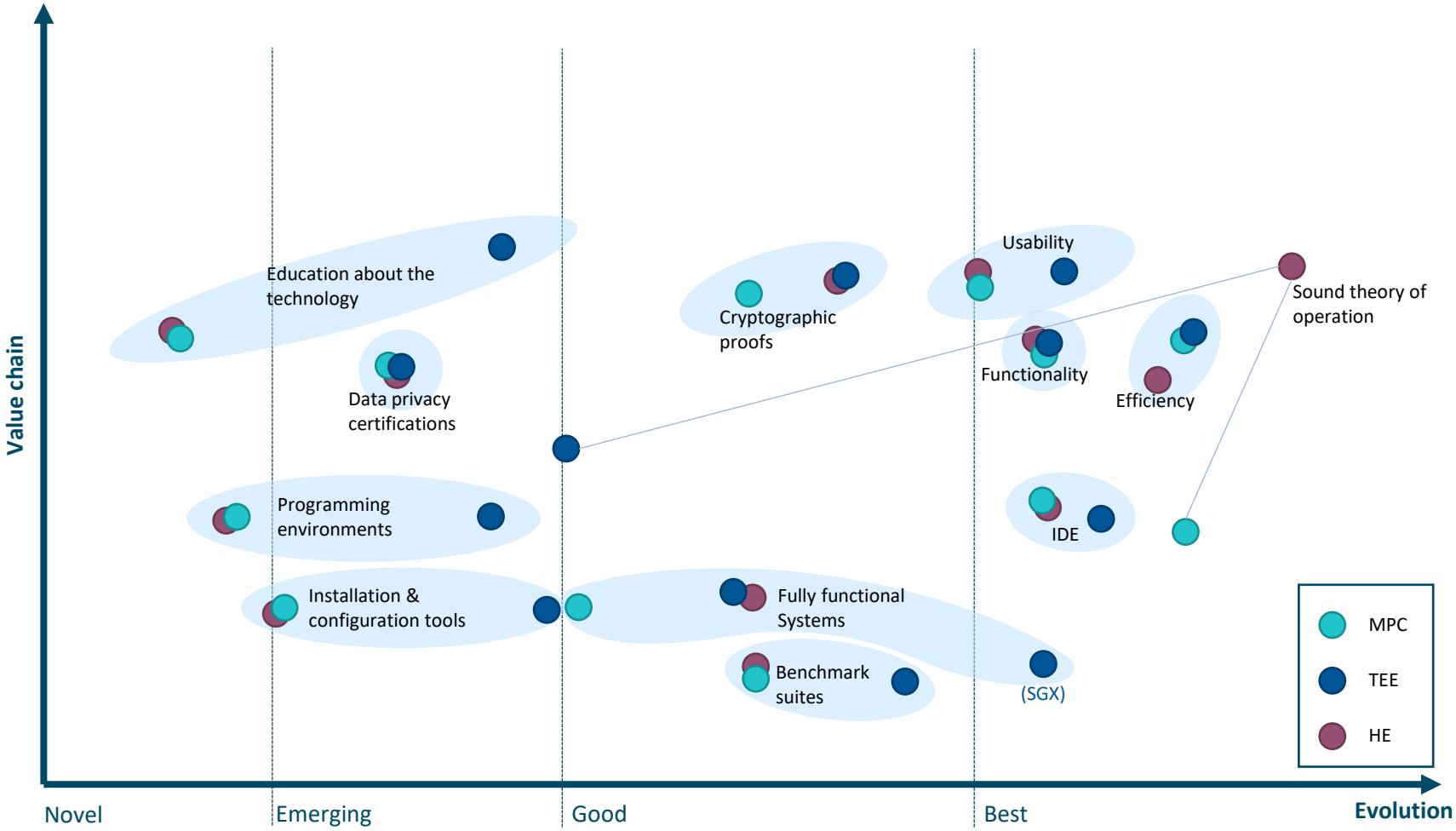
Smals
ICT for society



HE, MPC, TEE – Which maturity?

Smals
ICT for society

Maturity of confidential computing techniques



Source: « UN Handbook for Privacy-Preserving Computation Techniques », 2023.

https://unstats.un.org/bigdata/task-teams/privacy/UN%20Handbook%20for%20Privacy-Preserving%20Techniques.pdf

Smals ICT for society

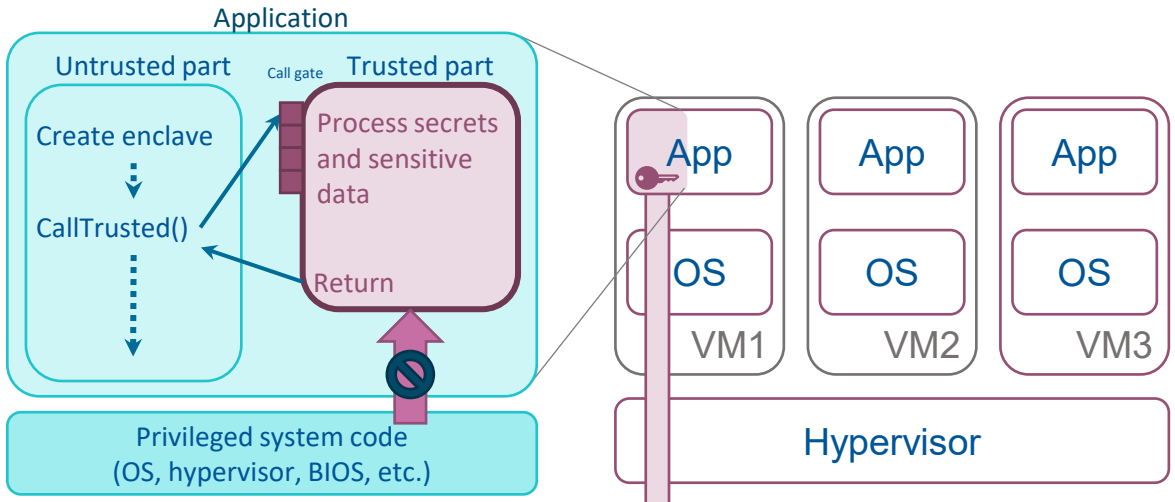
TEE-based market offer

AMD SEV-SNP, Intel SGX / TDX

Smals
ICT for society

Two main types of hardware-based isolation

Intel SGX



AMD SEV-SNP, Intel TDX

Attestation mechanism specific to AMD or Intel (based on secret fused inside processor)

Untrusted element

Trusted element

Intel Software Guard Extension (SGX)

Intel Trust Domain Extensions (TDX)

AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP)

Smals ICT for society

Conclusions and recommendations

Smals
ICT for society

General remarks

- HE and MPC not mature enough and limited to niche applications
- TEE provide improved level of protection for computing infrastructures and applications, thanks to:
  - Better process isolation
  - Hardware memory encryption
  - Secure boot
  - Remote attestation

Remarks on the market offer

- Main infrastructure offers:
  - Azure offers the most varied solution based on Intel and AMD
  - AWS' offer is different in nature (Nitro architecture) – with small offer based on AMD SEV-SNP
  - Google's offer appears* less mature than the offers from AWS and Microsoft
- Unresolved trust issue: client still needs to fully trust the infrastructure provider in practice
- Uncertain overall performance impact due to added complexity

* At time of study during first half of 2023

Recommendations

- **Attestation**:
  - Ability to verify TEE content independently from infrastructure provider (e.g., should be signed by hardware manufacturer)
- **Transparency**:
  - Ability to verify source code of any software in trusted computing base (TCB)
- **Key management**:
  - Ability to import own keys on dedicated hardware (minimum)
  - Better: manage keys externally
- **Training**:
  - Provide specific training for analysts, architects, and developers
- **Holistic view**:
  - Consider security of the system as a whole

https://www.smalsresearch.be/

Fabien.Petitcolas@smals.be

Smals
ICT for society