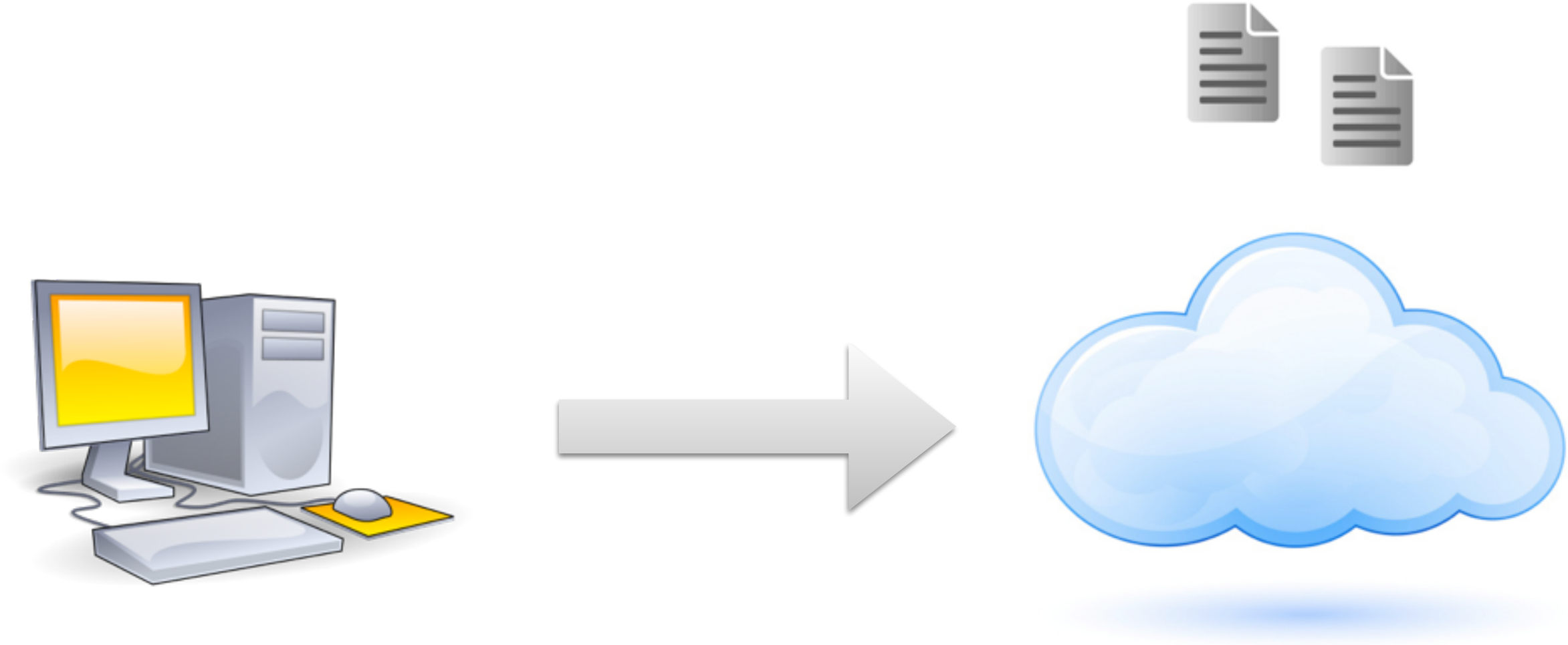
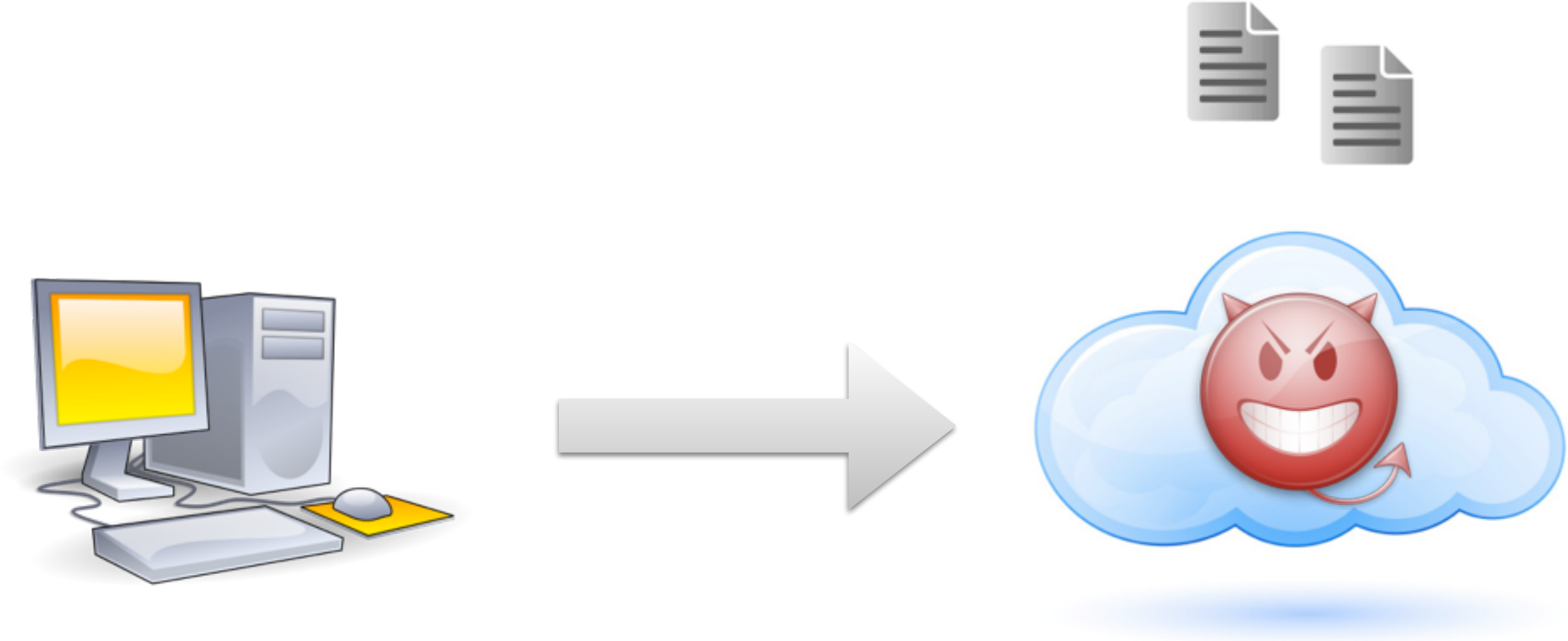
Shielding Applications from an Untrusted Cloud with Haven

Wenting Zheng





Theory

Searchable encryption

OPE

Oblivious RAM

…

Systems

CryptDB: queries on encrypted database

Pinnochio, Pantry: verifiable computation

VC3: secure MapReduce

……

Theory

Searchable encryption

OPE

Oblivious RAM

…

Systems

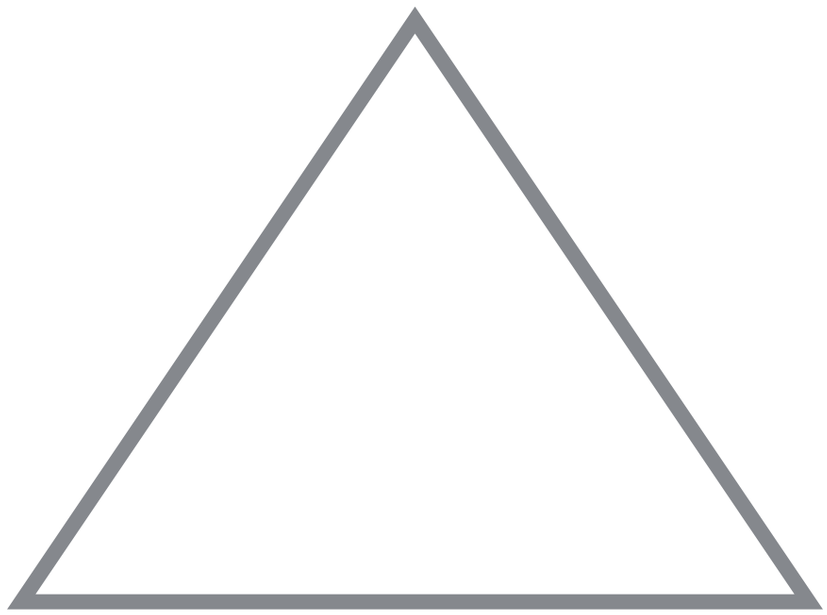
CryptDB: queries on encrypted database

Pinnochio, Pantry: verifiable computation

VC3: secure MapReduce

……

Strong Security



Functionality

Performance

Haven: shielded execution on the cloud

Goal: Secure, private execution of unmodified applications in an untrusted cloud on commodity hardware

Haven: shielded execution on the cloud

Goal: **Secure, private execution** of unmodified applications **in an untrusted cloud** on commodity hardware

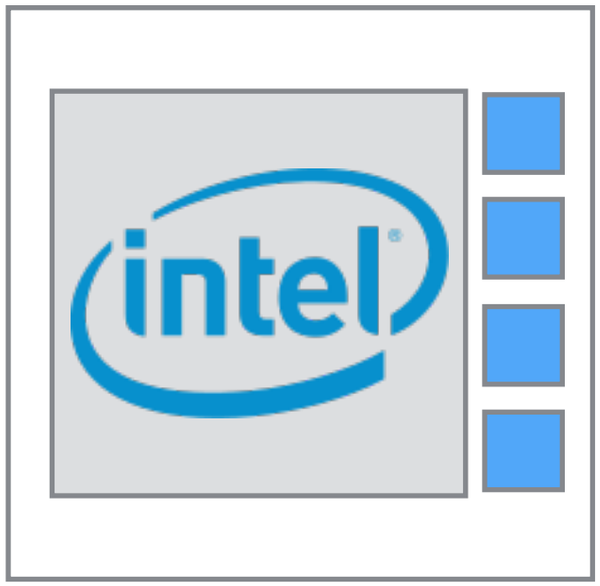
Haven: shielded execution on the cloud

Goal: Secure, private execution of
unmodified applications
in an untrusted cloud
on commodity hardware

Threat Model

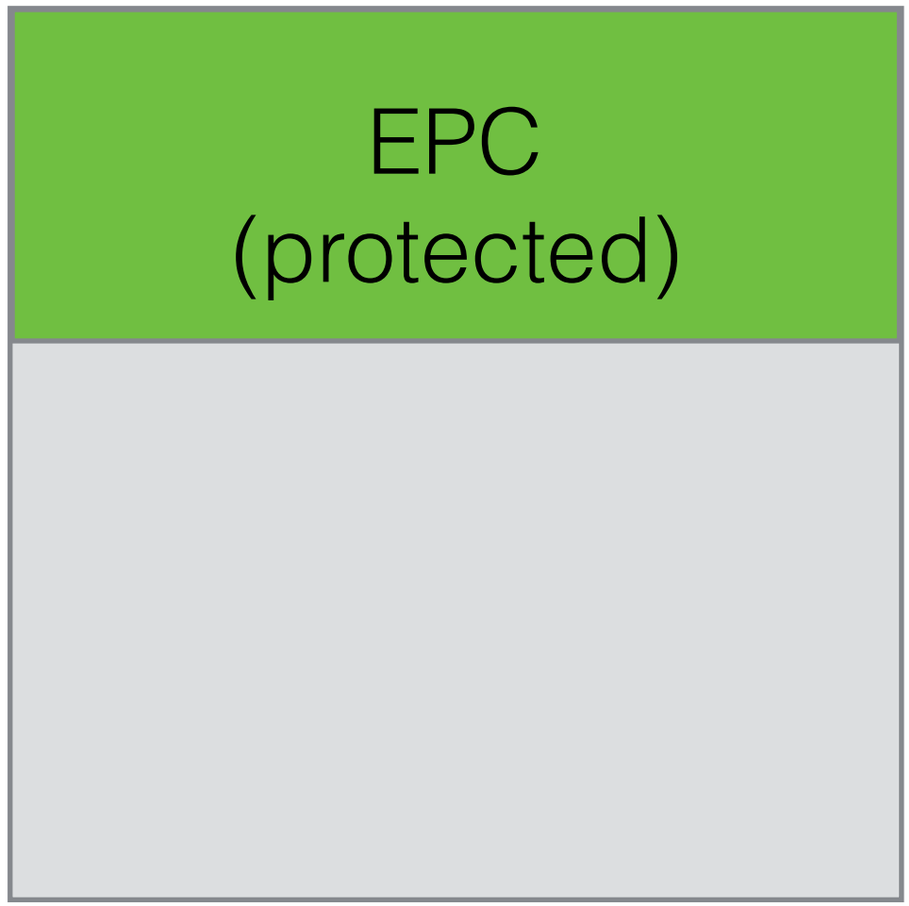
- Haven
  - Powerful adversary: controls most of hardware, and all of software
  - Hardware
    - Processor is not compromised
    - Encrypted memory (strong encryption, no rollback
  - Software
    - Controls OS, hypervisor, etc
- Does not prevent side channel attacks, DoS

SGX: Software Guard Extensions



Trusted CPU

RAM



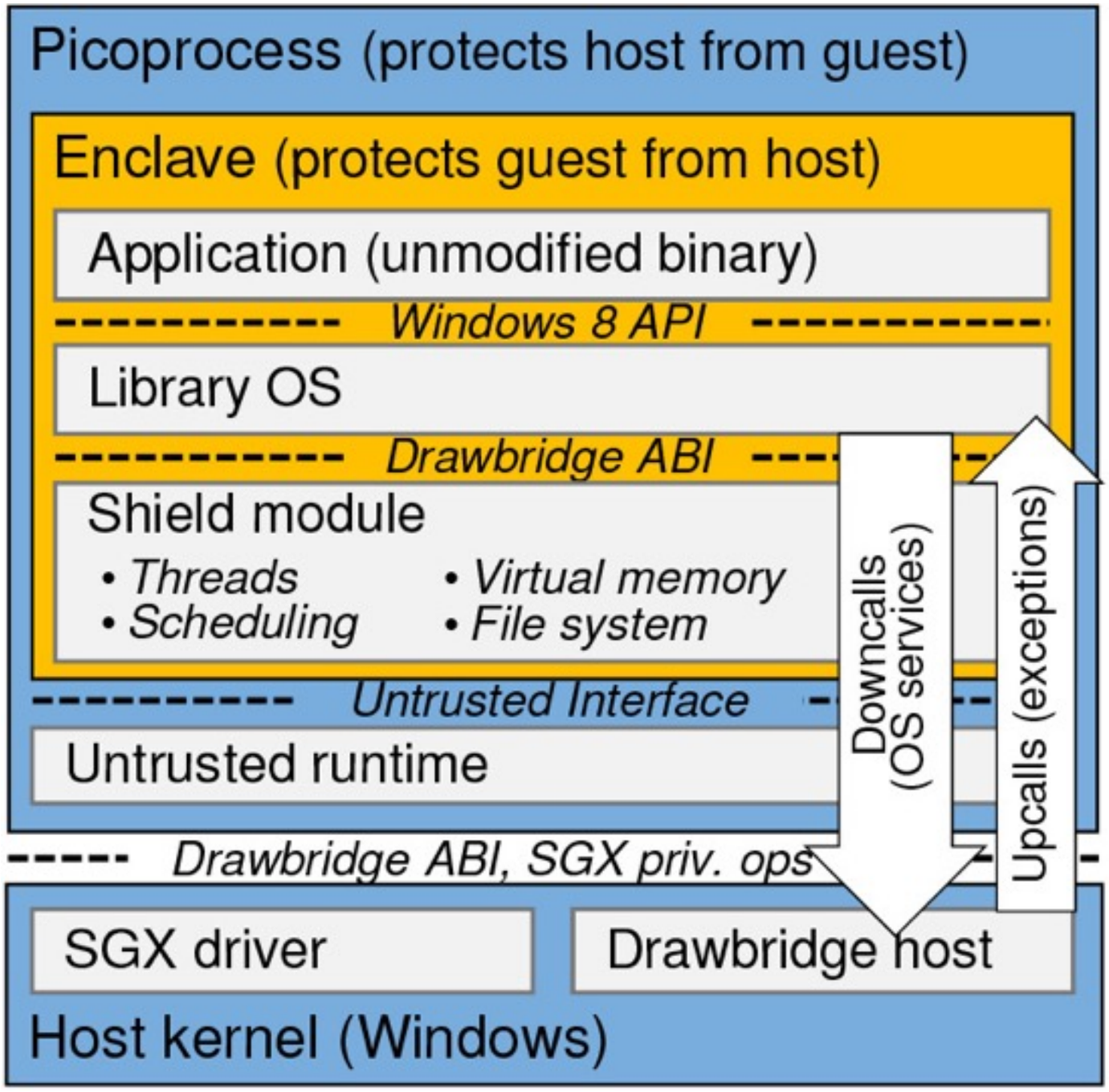
Code run in enclave is secure!

Threat Model

- Haven
  - Powerful adversary: controls most of hardware, and all of software
  - Hardware
    - Processor is not compromised
    - Encrypted memory (strong encryption, no rollback
  - Software
    - Controls OS, hypervisor, etc
- Does not prevent side channel attacks, DoS

- CryptDB
  - Supports data confidentiality
    - Guarantees users' data confidentiality *if* they are not logged in during an attack
  - Assumes DBMS server is not compromised

Design


Picoprocess (protects host from guest)
Enclave (protects guest from host)
Application (unmodified binary)
Windows 8 API
Library OS
Drawbridge ABI
Shield module
• Threads
• Scheduling
• Virtual memory
• File system
Untrusted Interface
Untrusted runtime
Downcalls (OS services)
Upcalls (exceptions)
Drawbridge ABI, SGX priv. ops
SGX driver
Drawbridge host
Host kernel (Windows)

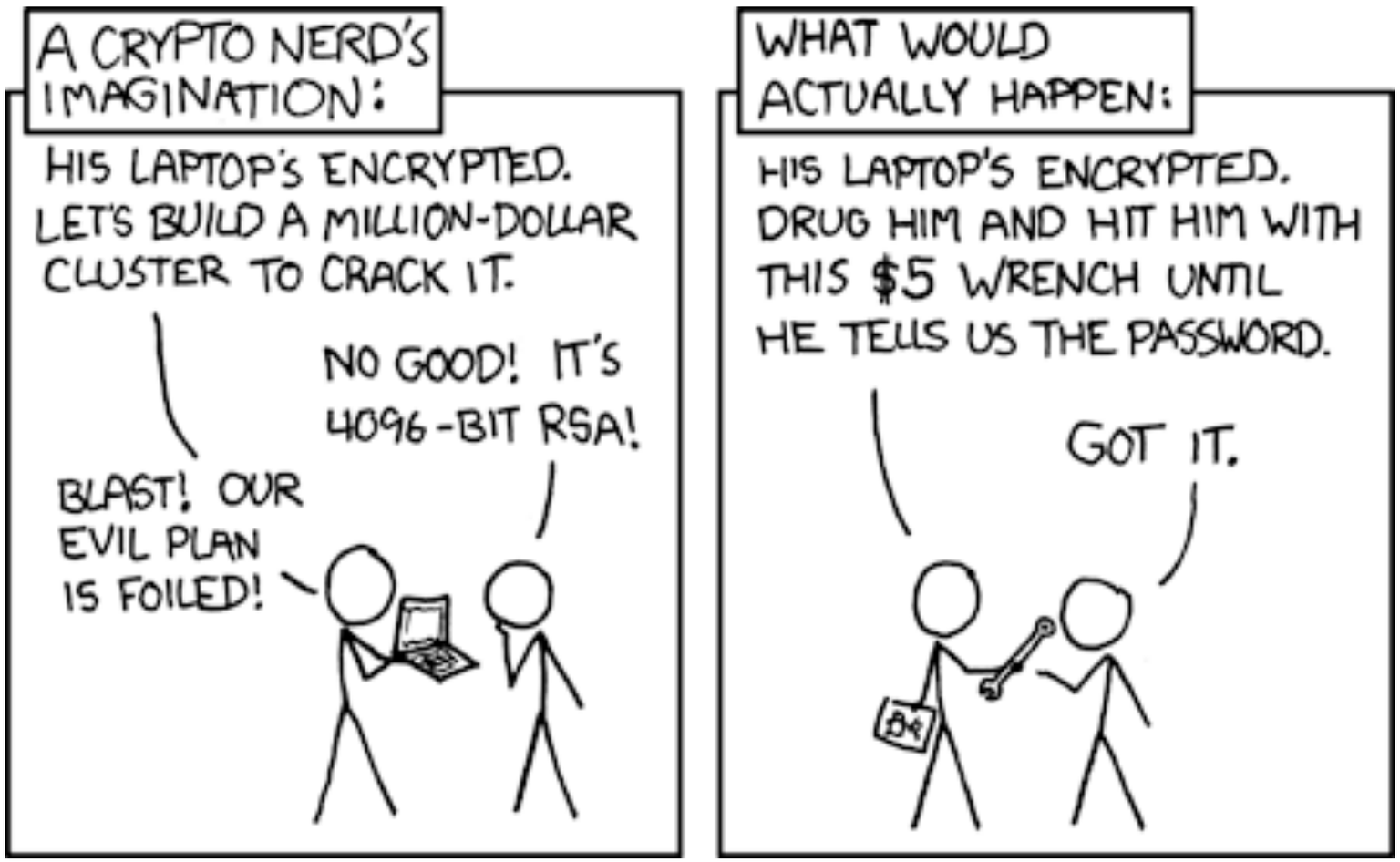
Design

- Small interface between trusted domain and untrusted domain

Design

- Small interface between trusted domain and untrusted domain
- Unmodified binaries —> a lot of code in an enclave —> large TCB
- VC3 “recent work proposes loading a library variant of Window 8 together with an application into SGX… results in a TCB that is larger than VC3’s by several orders of magnitude”

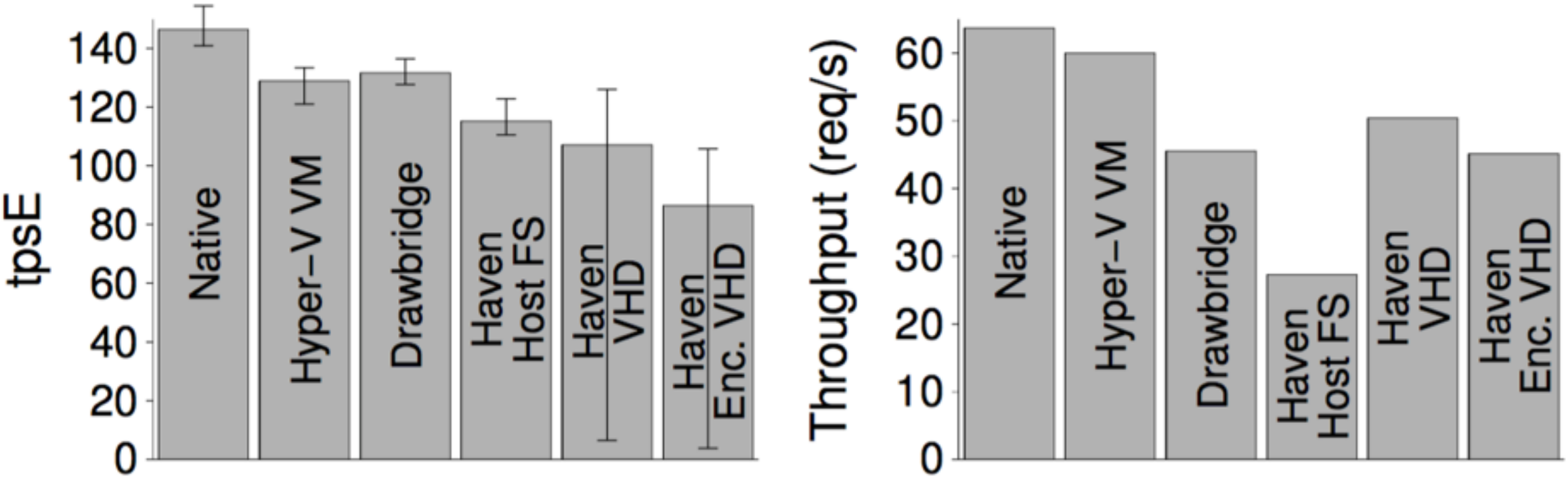
Security


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Security

- DoS
- Storage: rollback attacks
- Side channels: memory
- Systems issues
  - what happens when an enclave terminates?
  - what happens when there is a failure?

Performance



(a) SQL Server, TPC-E (b) MediaWiki on Apache

Figure 4: Performance breakdown

What’s the catch?

- Reliance on specific hardware

What’s the catch?

- Reliance on specific hardware
- Should we trust Intel?

Did NSA Put a Secret Backdoor in New Encryption Standard?

“We cannot trust” Intel and Via’s chip-based crypto, FreeBSD developers say

Following NSA leaks from Snowden, engineers lose faith in hardware randomness.

Discussion

- Is SGX “the answer”? Do we need fancy crypto?
- Is Intel trustworthy?
- How much code in each enclave? Is smaller TCB better?
- How to deal with side channels?
- Strong security, functionality, performance: can we achieve all three?
  - inherent tension between security and performance?