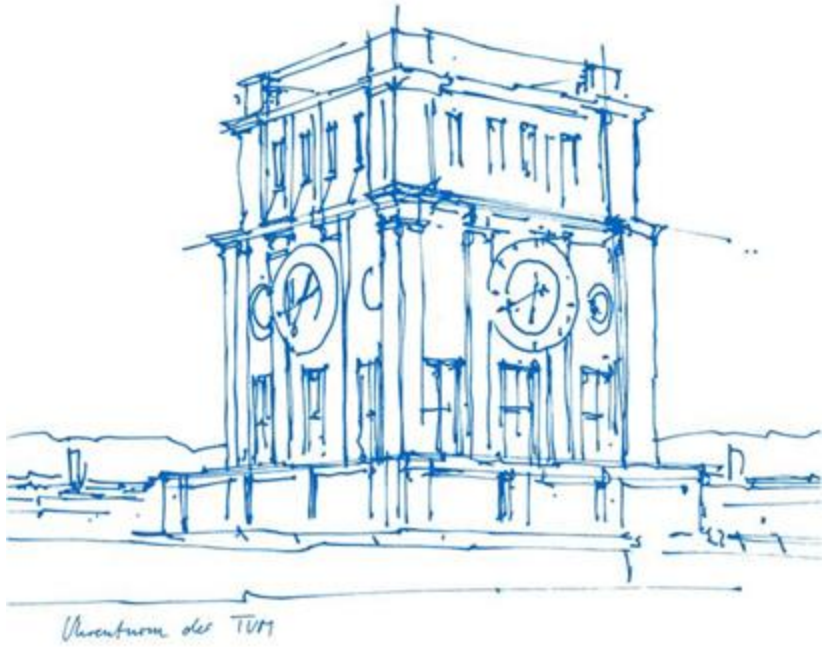
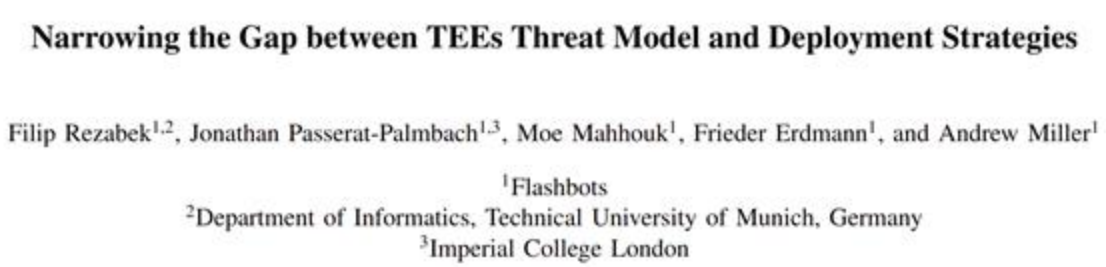
Narrowing the Gap between TEEs Threat Model and Deployment Strategies

Filip Rezabek
frezabek@net.in.tum.de

Jonathan Passerat-Palmbach
Moe Mahhouk
Frieder Erdmann
Andrew Miller





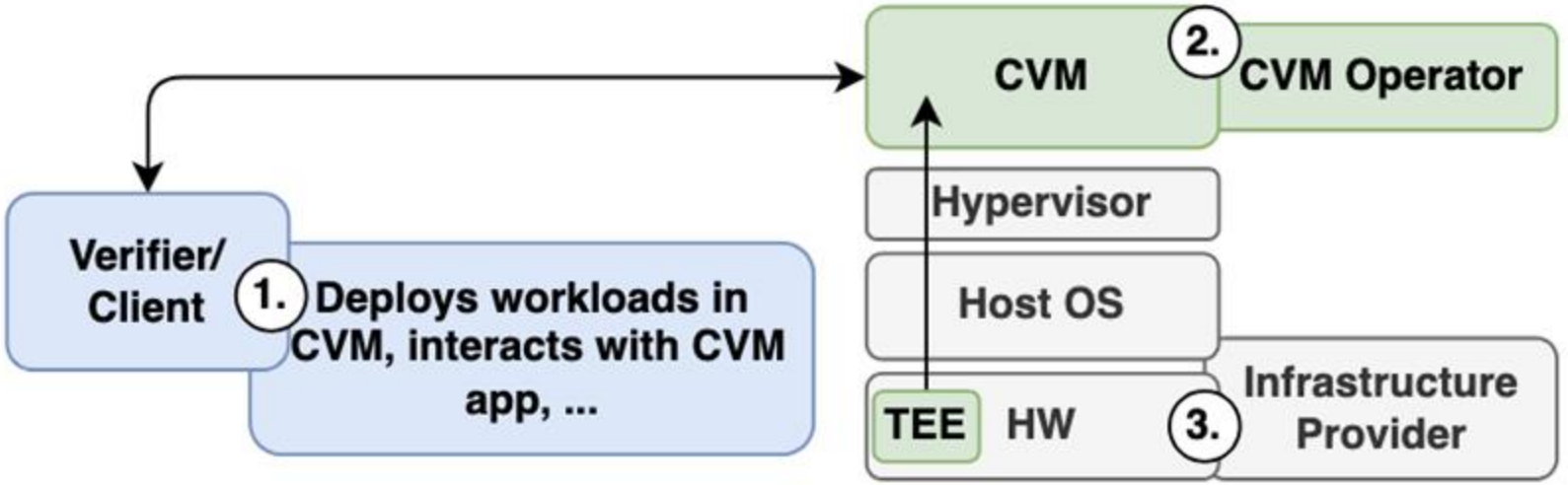
Narrowing the Gap between TEEs Threat Model and Deployment Strategies

Filip Rezabek[1,2], Jonathan Passerat-Palmbach[1,3], Moe Mahhouk[1], Frieder Erdmann[1], and Andrew Miller[1]

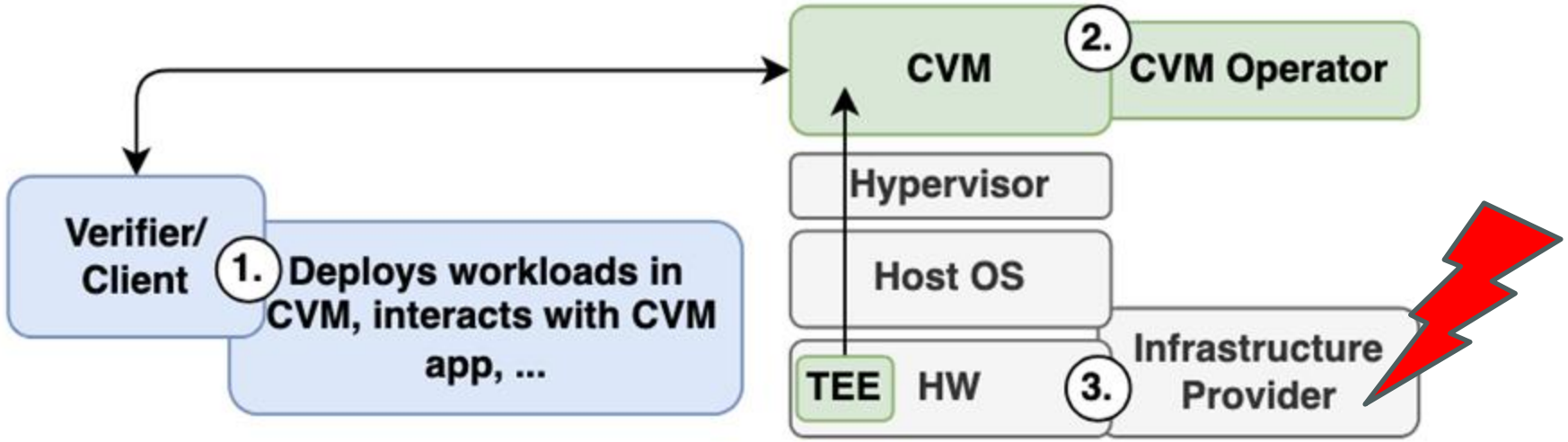
[1]Flashbots
[2]Department of Informatics, Technical University of Munich, Germany
[3]Imperial College London

Flashbots

Motivation
Setting

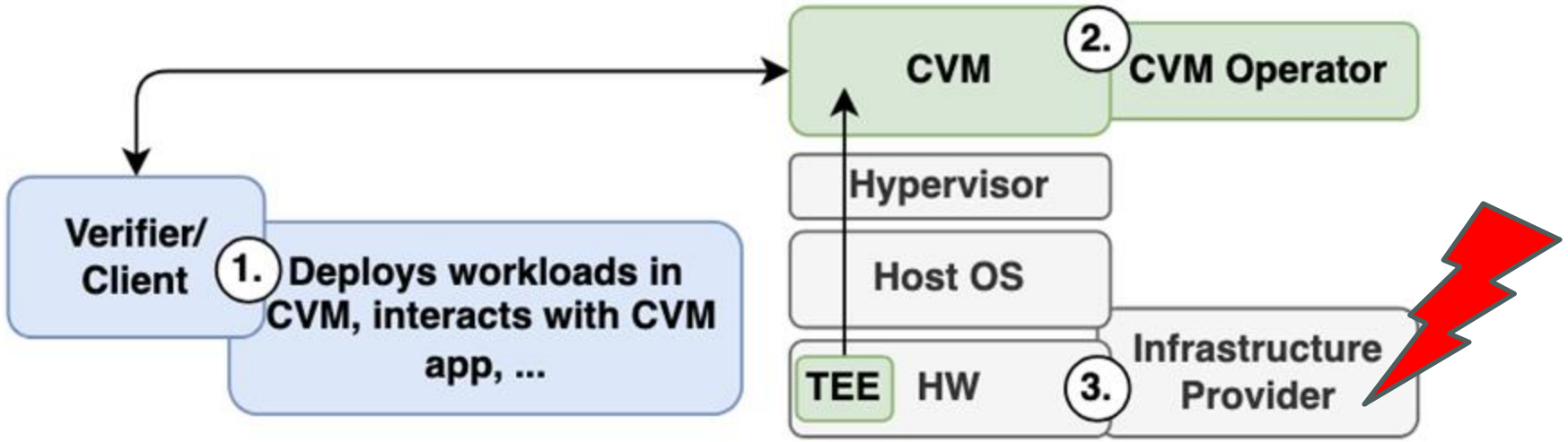


Motivation
Setting



Sophisticated hardware layer attacks are possible

Motivation
Setting



Problematic, especially in malicious setting of blockchain (Maximal Extractable Value, …)

- Compromising a **TEE**, could lead to large financial losses
- Not limited to blockchains - AI model&data, …

Motivation
Problem Definition

Trust assumptions are now on the host/operator being honest, to ensure **physical security**

However, current **TEE attestation flows** do not provide guarantees they operate in a given infrastructure

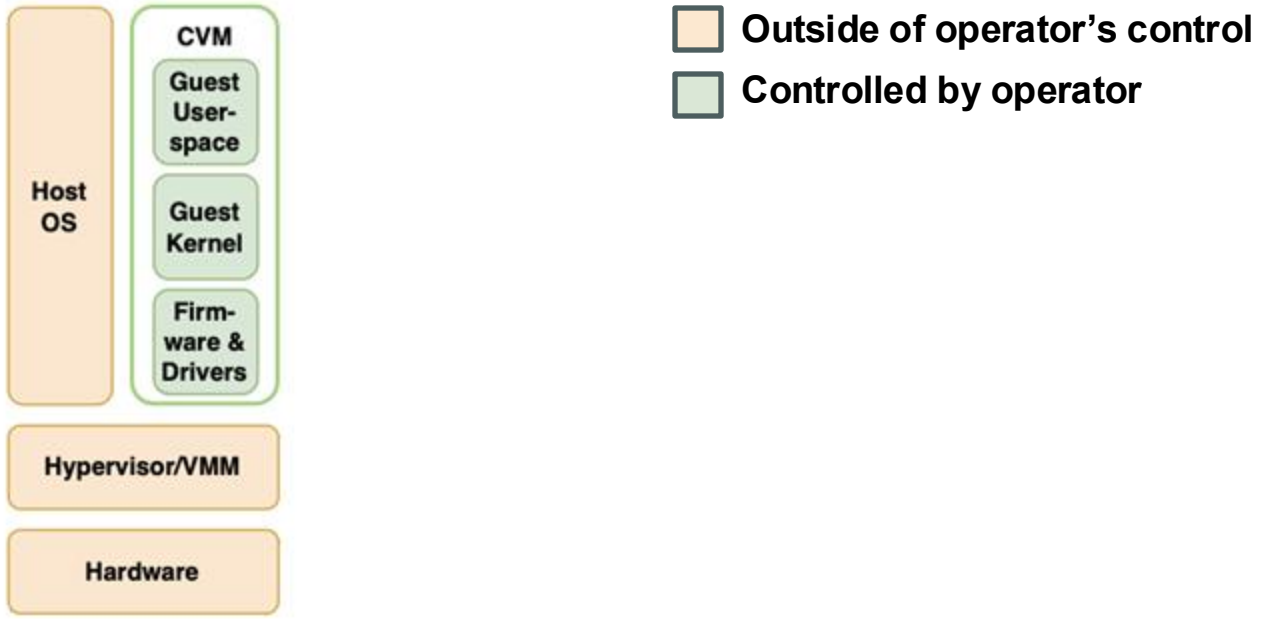
→ Provide **assurance that CVM** runs in the respective (trusted) infrastructure

How do we extend the attestation flow so the CVM runs on the respective infrastructure?

Background
Deployments

Two **deployments**:

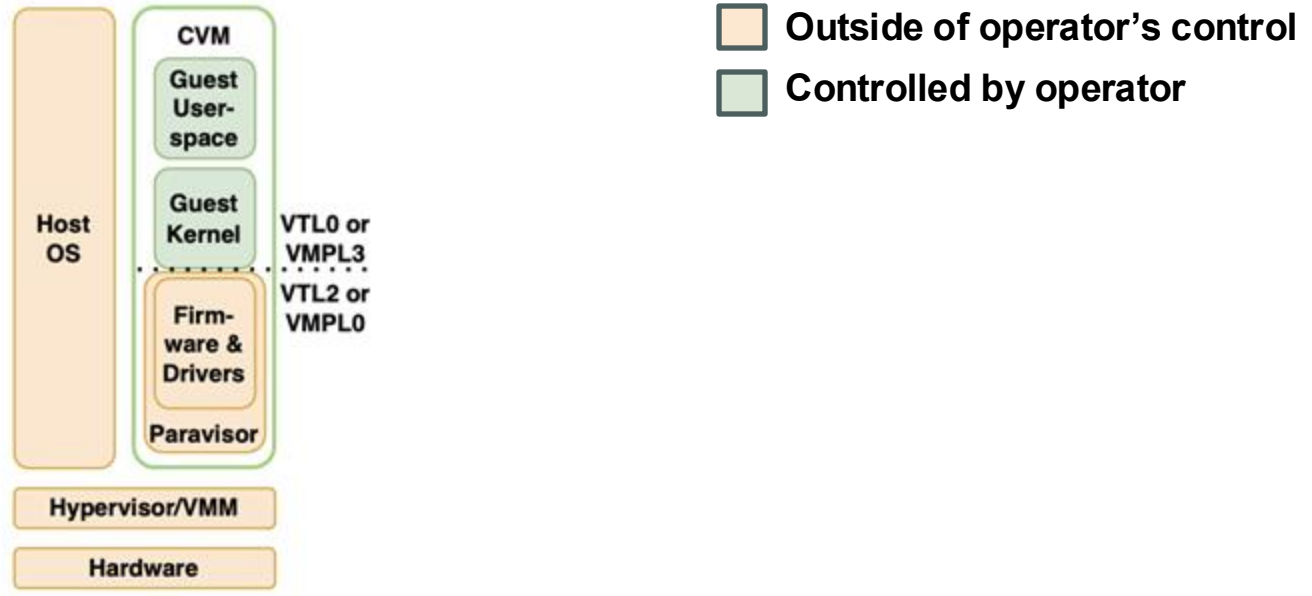
1. **Bare/CVM** flow



Background
Deployments

Two **deployments**:

1. **Bare/CVM** flow
2. **Paravirtualization** flow
   - Virtual Trust Level (VTL) for Intel or Virtual Machine Protection Level (VMPL) for AMD
     - Direction towards open source OpenHCL and COCONUT



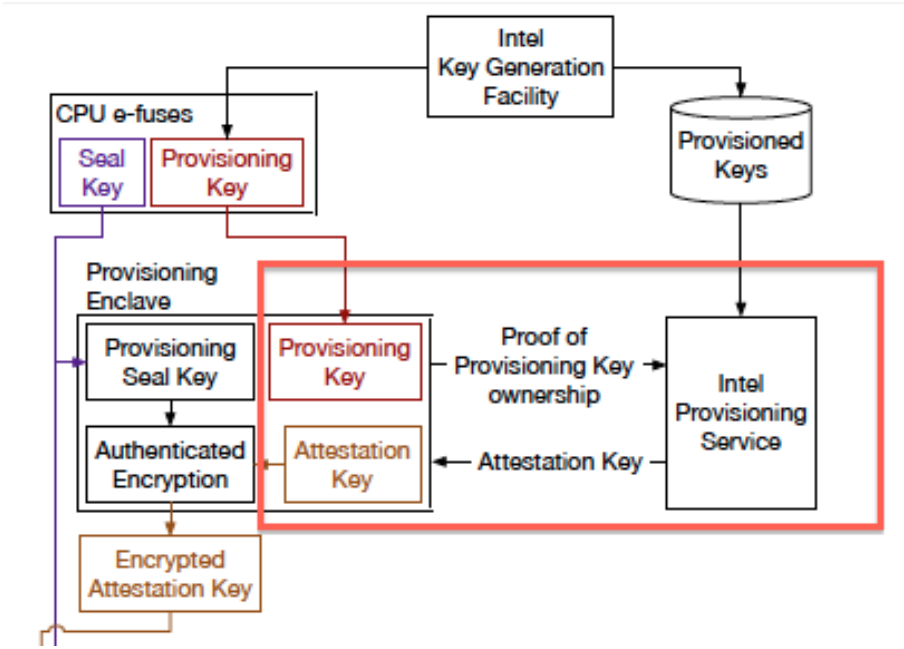
Background
Intel TDX

Intel TDX relies on two Intel SGX enclaves as a part of its attestation flow

- PCE (Provisioning Certificate Enclave)
- TDQE (Trust Domain Quoting Enclave)

In collaboration with **Intel**, receive an attestation key

As a part of the flow, provide PPID → unique identifier of the platform



Design
Use PPID?

We can rely on **PPID**

- Is part of the attestation flow
- **Unique** per platform
- **Fixed** for the platform

Design
Use PPID?

We can rely on **PPID**

- Is part of the attestation flow
- **Unique** per platform
- **Fixed** for the platform

What about the **binding** to the infrastructure provider?

Design
Use PPID?

We can rely on **PPID**

- Is part of the attestation flow
- **Unique** per platform
- **Fixed** for the platform

What about the **binding** to the infrastructure provider?

- **Provider** can create a database storing the values
- **Verifier** can query the database and receive True/False as an output

Design
Use PPID – Challenges

Introduces some limitations:

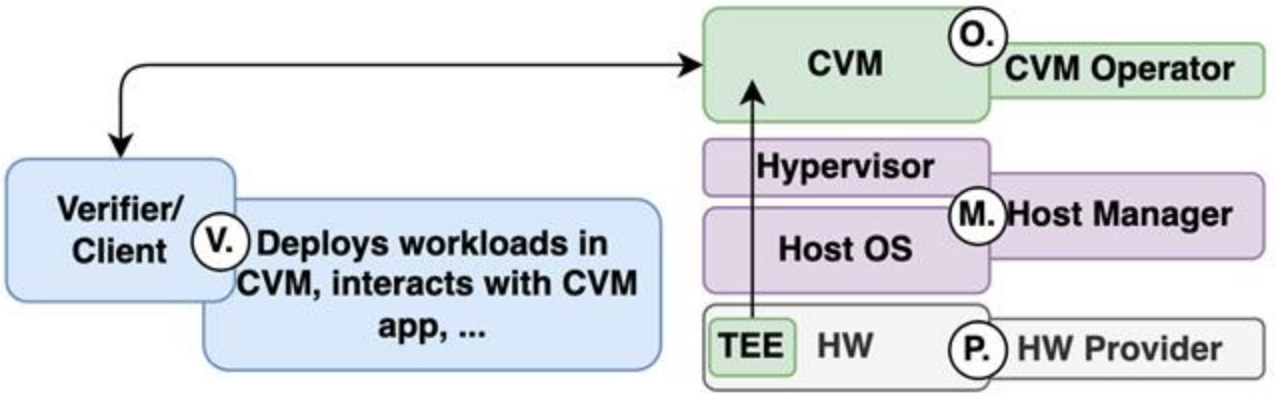
- **Relies** on provider to share those information
  - Might vary across **cloud** providers
  - Extension to many parties, otherwise hard to integrate
- Limited **visibility** for the case of different **CVMs** on the same node
- Other **TEE** implementations
- **Bare** metal deployments?

Extension to Bare Metal

Future directions

Two scenarios:

1. **Confidential Virtual Machine (CVM)** in cloud
2. **Bare** metal in cloud

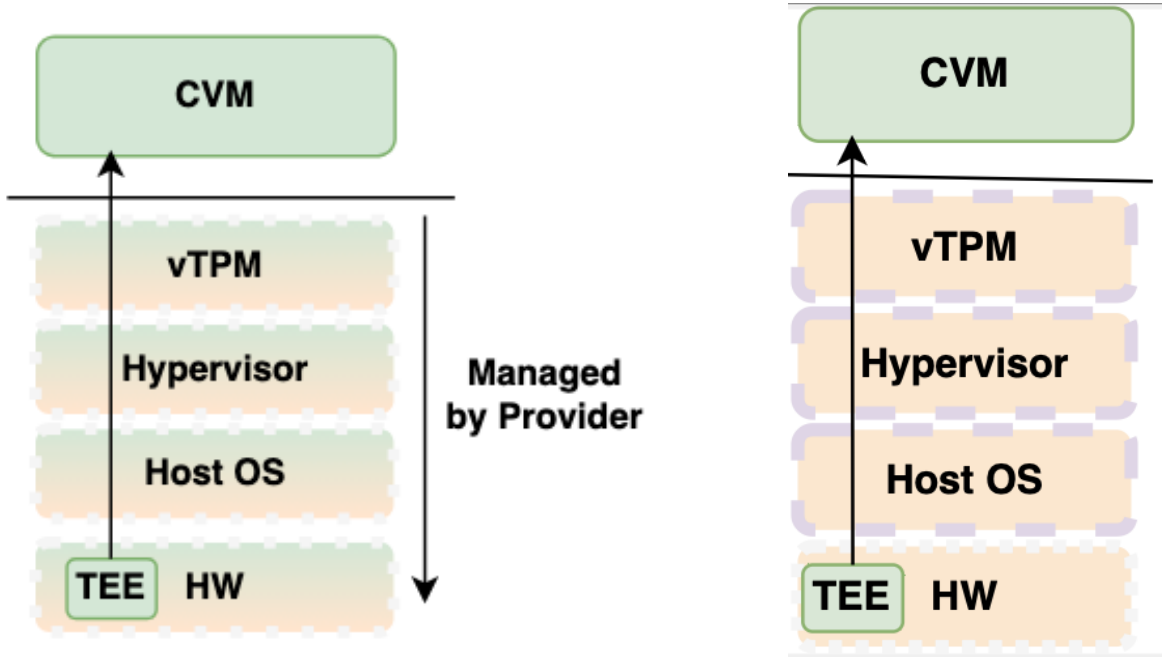


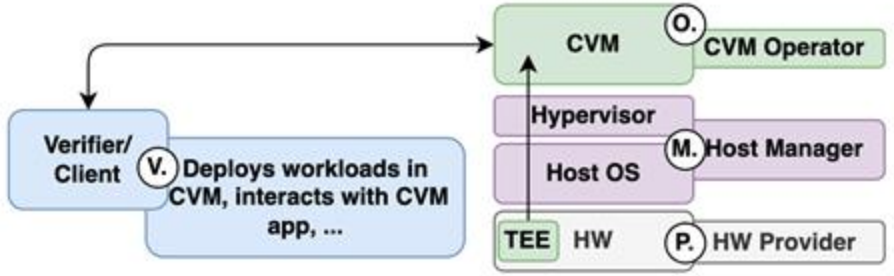
Extension to Bare Metal

Future directions

Two scenarios:

1. **Confidential Virtual Machine (CVM)** in cloud
2. **Bare** metal in cloud





Summary
Overview

Identification of the **gap** between attestation and threat model

Suggestion to strengthen it using **PPID** (or similar for AMD)

Poses **several** challenges

Future work should expand on the **bare metal** and less involvement of the **provider**

Summary
Overview

Identification of the **gap** between attestation and threat model

Suggestion to strengthen it using **PPID** (or similar for AMD)

Poses **several** challenges

Future work should expand on the **bare metal** and less involvement of the **provider**

Thank you!

frezabek@net.in.tum.de