duet: Combining a Trustworthy Controller with a Confidential Computing Environment

Istemi Ekin Akkus, Ivica Rimac 08.07.2024

7th Workshop on System Software for Trusted Execution (SysTEX 2024)

NOKIA BELL LABS

Trusted Execution Environments (TEEs)

Confidentiality and integrity protection

- Confidential Virtual Machines
  - AMD SEV-SNP, Intel TDX
- Application Enclaves
  - Intel SGX, ARM TrustZone

- Protect data in-use
  - Confidentiality of data during computation
- Enable remote attestation
  - Integrity of code computing over data
- Base root of trust on hardware
  - No trust in the cloud provider or operator

Verifiable Computation

Verifiable Computation with TEEs – A Comparison

Confidential Virtual Machines (CVMs) run entire operating systems in the TEE

Pros

- Lift and shift with no change to applications
- Flexible and easy-to-use
- Access to confidential GPUs (e.g., NVIDIA H-100)

Cons

- Remote attestation integrity only for initial state
- Owner has full control (i.e., “insider threat”)

Owner Verifiable Computation

Application Enclaves run only certain parts of the application in the TEE

Pros

- Minimized trusted computing base
- Transparent integrity of code via remote attestation (i.e., no changes after start)

Cons

- Requires re-architecting and splitting applications into "trusted" and "untrusted" code
- No access to specialized hardware

3rd Party Verifiable Computation

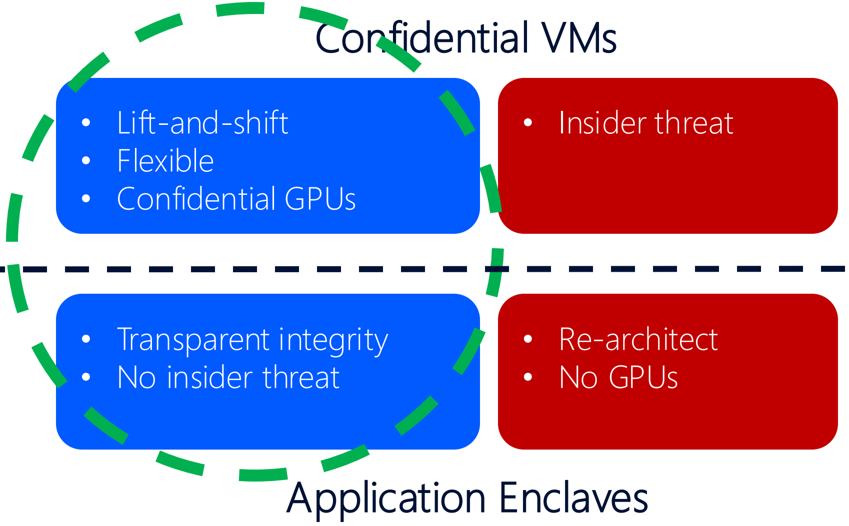
NOKIA BELL LABS

Verifiable Computation with TEEs – A Guideline

Q1. Whose data is it?

Q2. Who is doing the computation?

- Same entity:
  - "Owner self-run": use confidential VMs for its advantages
- Not the same entity:
  - "as-a-Service provider": use application enclaves to ensure trust for users



Goal

To combine

the **transparent integrity** of application enclaves

with

the **flexibility and ease-of-use** of confidential VMs

Agenda

- Motivation
- Background & Assumptions
  - CVM integrity
  - Actors
  - Threat Model
- duet Overview
- Prototype Implementation

CVM Integrity Checks

- Attestation report covers only the firmware
  - But not the rest of boot binaries
- Measured boot approaches with other boot binaries
  - QEMU and OVMF patches: kernel, kernel command line, initRAMdisk
  - vTPM as a Secure VM Service Module (SVSM)

Gives only the initial CVM state integrity

- Good for "Owner verifiability"
- Not good for "3rd party verifiability"

- "Read-only disk" hosting the confidentiality-offering service
  - Revelio [Middleware'23]
  - Confidential Containers (CoCo)

Gives runtime integrity

- Good for "3rd party verifiability"
- Not good for "flexibility" and "maintenance"

NOKIA BELL LABS

Actors

Service Owner

Offers a service with confidentiality protections

- “Confidential/Private Machine Learning”

Service User

Uses the confidentiality-offering service

- Has confidential/private assets but does not want to run the service

Infrastructure Provider

Supplies the underlying hardware infrastructure with up-to-date TEEs

- No other interaction with service owner and user

Threat Model & Assumptions

- Attacks on TEE hardware out-of-scope
- Infrastructure provider not interfering with disk integrity
  - Can be combined with integrity protection solutions (done once per OS image)
- Publicly available OS and software packages
  - Well-known Ubuntu image, Ubuntu packages
- Non-public software packages are from well-known sources
  - NVIDIA GPU drivers signed by NVIDIA
- Publicly available confidentiality-offering service code

Agenda

- Motivation
- Background & Assumptions
- duet Overview
  - High-level overview
  - Service Owner workflow
  - Service User workflow
- Prototype Implementation

duet Overview

Approach in a nutshell

Goals

- Flexibility and ease-of-use for service owner
  - Need CVMs for specialized hardware like confidential GPUs
  - Need updates to the OS packages and service for new functionality
- CVM's runtime integrity
  - Service owner == CVM owner
  - How do we prevent the service owner from installing anything malicious after the initial boot?

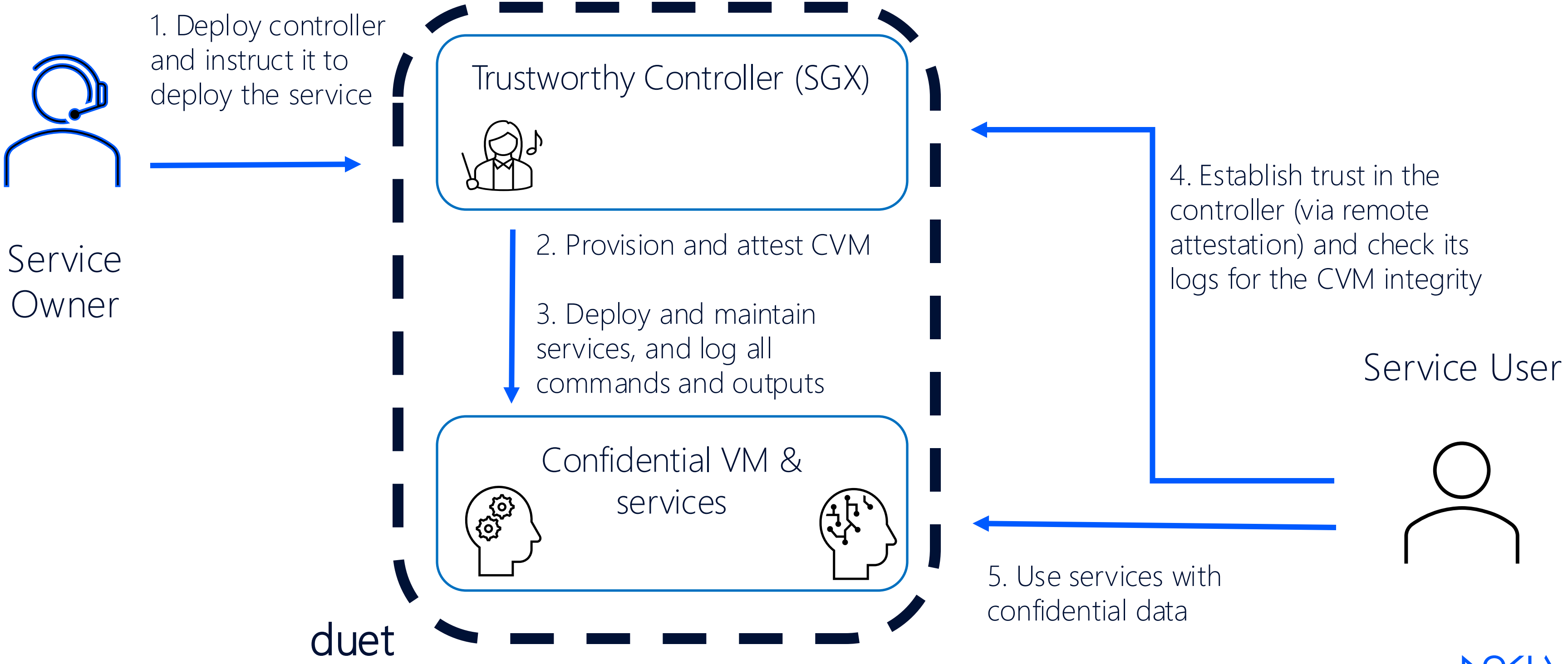
Idea

1. An application enclave gives transparent runtime integrity
   - ➢ SGX MRENCLAVE value

2. Let the application enclave run a controller application to "own" the CVM
   - Service owner =/= CVM owner

duet

Combining best of both worlds



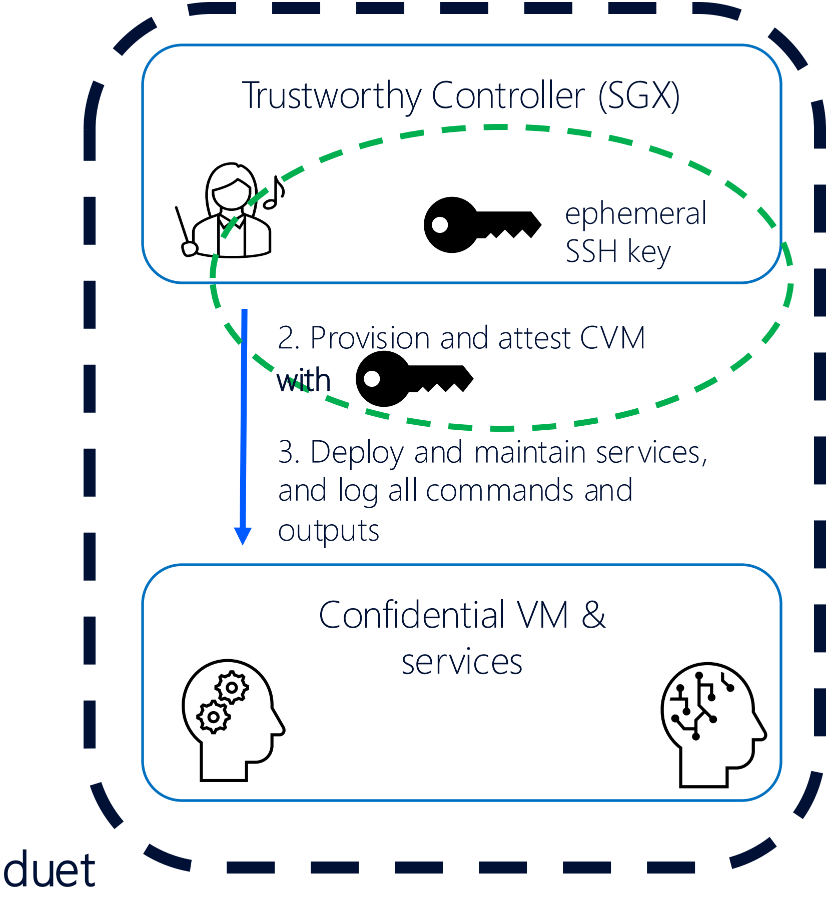
Owning the CVM

Trustworthy controller operation

- Provisioning resources for the CVM
    - From cloud provider or on-premise server
- CVM access only possible by the controller
    - Controller generates an ephemeral SSH key while provisioning the CVM; other login options are disabled

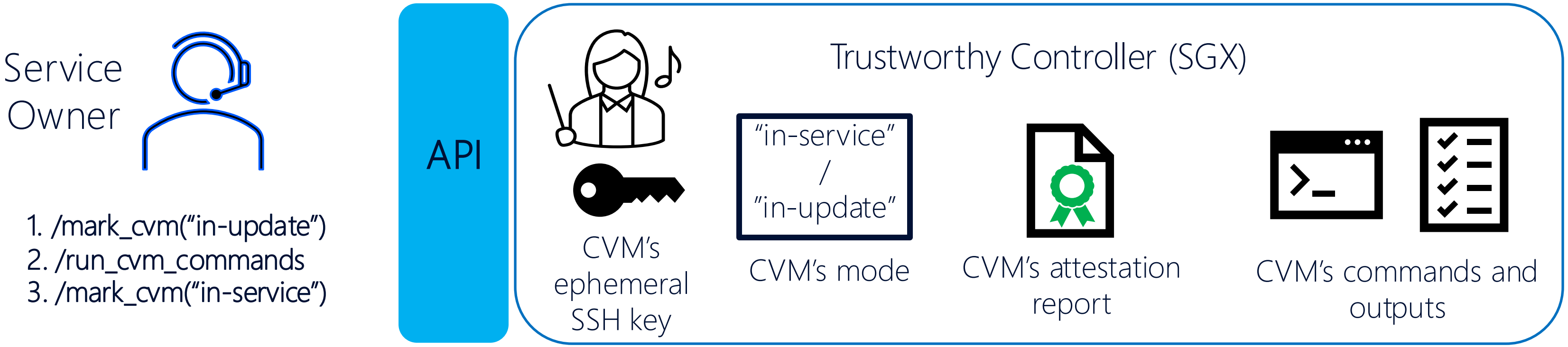
The SSH key stays inside the SGX enclave [confidential]

➢ Controller has full control of CVM
- Deploying and maintaining the service possible via the controller API

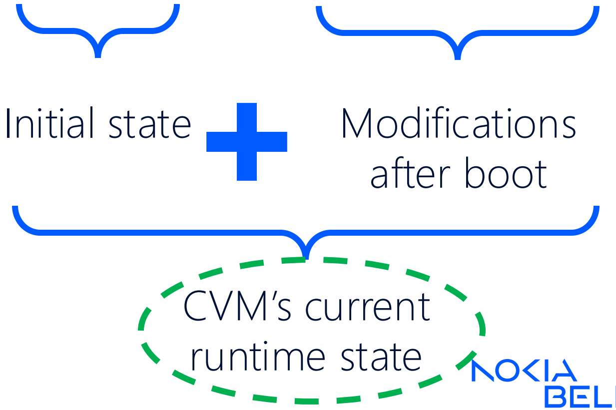


Deploying and Maintaining the Confidential Service

Service owner workflow

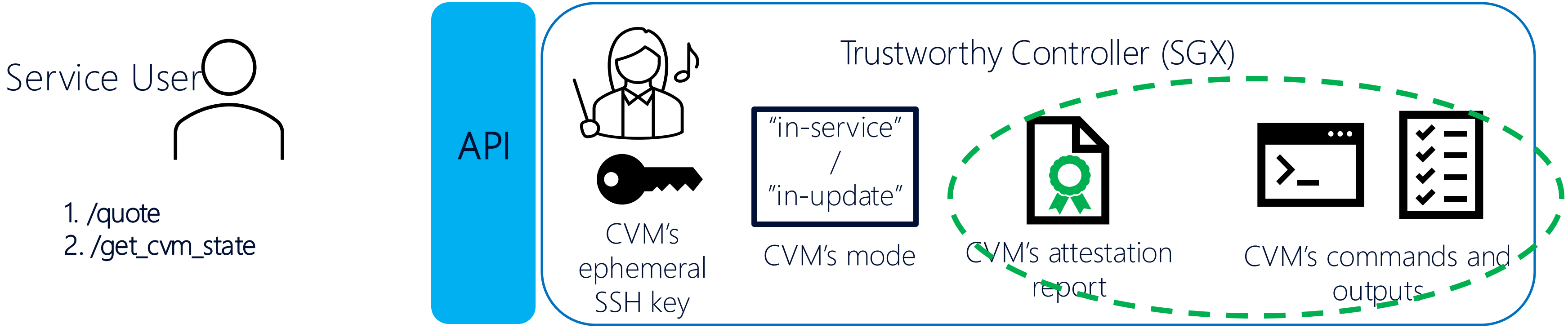


- Service owner can manage packages and service via controller interface
- Controller logs all commands, applies them to the CVM and logs all outputs
- Controller can reject commands if certain conditions are not satisfied



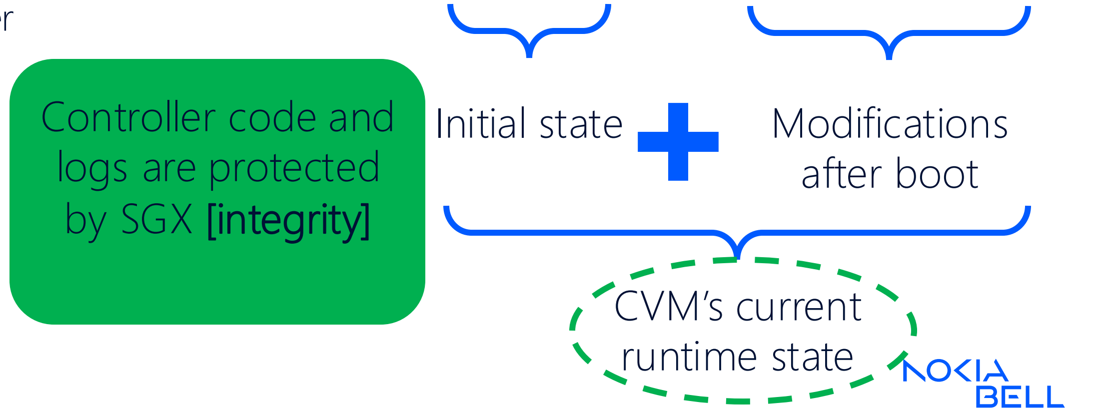
Confidential Service Operation

Checking the runtime integrity of the CVM by service users



- Service users establish trust in the controller via SGX remote attestation
- Users check CVM's runtime integrity
  - Obtain "current CVM state"
  - Check attestation report, logs and outputs

➢ If all good, share confidential/private data with confidential service



Agenda

- Motivation
- Background & Assumptions
- duet Overview
- Prototype Implementation
  - Controller & API client

NOKIA BELL LABS

Prototype Implementation

- Generic controller implementation
  - ~1K lines of Python code + gramine libOS for containerization
  - Controller API client with ~425 lines of Python code
- Initial set of CVM commands for attestation report
  - With Microsoft Azure Attestation
- Provider-specific provisioning commands
  - AMD SEV-SNP and Intel TDX on Microsoft Azure
- Standard Ubuntu 22.04 as CVM image
  - Designated for confidential computing by Canonical

Trustworthy Controller (SGX)

Limitations & Future Work

- Microsoft Attestation Service and firmware closed source
  - No direct access to the TEE
  - Other providers allow access to the TEE hardware
- Disk integrity assumed after initial boot
  - SNPGuard may help
  - Protection would be applied once per OS image; not per deployed service

NOKIA BELL LABS

Summary

- Alternative approach for verifying CVM runtime integrity
  - An application enclave as the CVM owner/controller
  - Transparent integrity checks for 3rd party verifiability
  - Flexibility and ease-of-use for service owner

Confidential VMs

- Lift-and-shift
- Flexible
- Confidential GPUs

- Transparent integrity
- No insider threat

Application Enclaves

- Source code available: https://github.com/Nokia-Bell-Labs/tee-duet

NOKIA BELL LABS

NOKIA
BELL
LABS