サーバーレス機密コンテナ

Serverless Confidential Containers

Carlos Segarra, Peter Pietzuch (Imperial College London)
Yoshiaki Sato (Waseda University)
Pierre-Louis Aublin (IIJ Research laboratory)

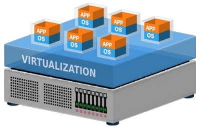
2025 年 5 月 19 日
IIJ Research Lab Seminars

Acknowledgements

This **talk** is heavily **based on** the following paper:

Segarra, C., Feldman-Fitzthum, T., Buono, D., & Pietzuch, P.
Serverless confidential containers: Challenges and opportunities.
In *Proceedings of the 2nd Workshop on SErverless Systems, Applications and Methodologies* (SESAME), April 2024.

Cloud Computing Layers

		 User Maintenance	 Speed	 Scalability	 Price
Serverless		– – – –	++++	++++	¥
Container		+	+++	++	¥¥
Virtualization		++	++	++	¥¥¥
Bare Metal		++++	– – – –	– – – –	¥¥¥¥

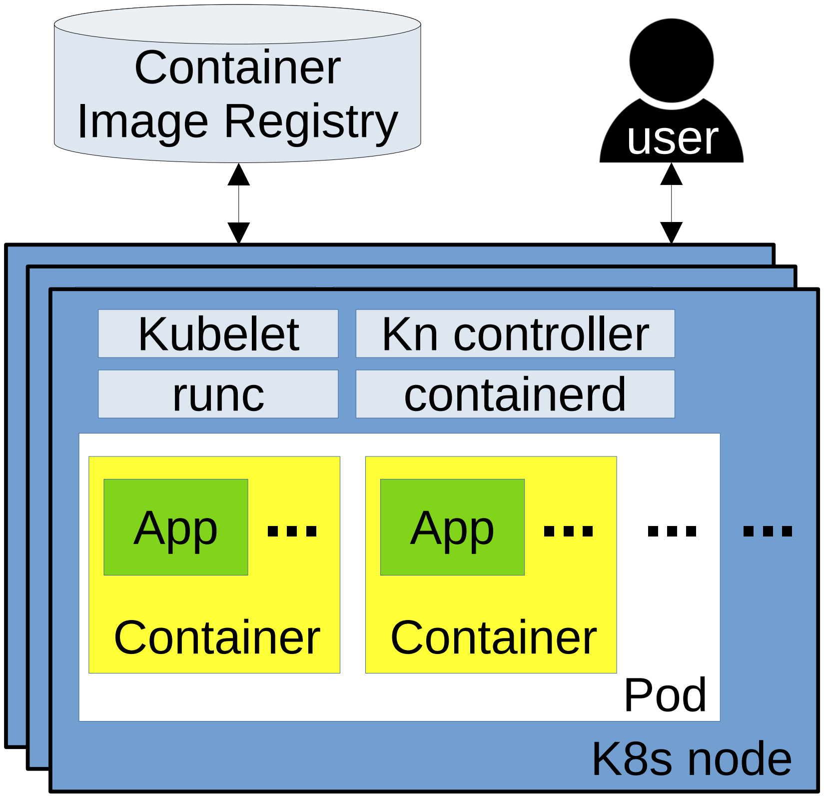
Serverless Behind the Scenes

- **Developer writes** and **deploys application** in the cloud
- **Cloud** provider **provides** entire **infrastructure**
  - Application runtime, datastore, authentication, monitoring, scaling, …
- Cloud provider can **colocate** multiple **users**

- Multiple **Frameworks**
  - OpenFAAS https://www.openfaas.com/
  - Apache OpenWhisk https://openwhisk.apache.org/
  - **Knative** https://knative.dev/docs/

Knative Serverless Computing Framework

- Built on top of Kubernetes (==K8s)
- Components:
  - **Container Image Registry**: manages container images (contain application code)
  - **Kubelet**: 1 per worker node, manages containers
  - **Pod**: basic scheduling/isolation unit; 1+ containers on same node
  - **Kn controller**: control-plane pod
  - **containerd**: container runtime
  - **runc**: tool to execute a container

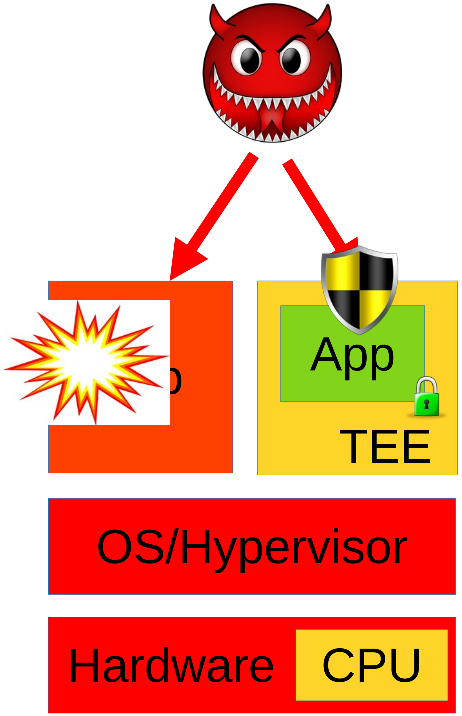


The Need for Confidential Serverless Computing

- **Serverless Use-cases**
  - Image processing
  - Machine learning
  - Medical research
  - Transaction fraud detection
  - ...
- **Confidentiality concerns**
  - Sensitive/private user data
  - Security attacks
  - Government-sponsored mass surveillance programs
  - Data protection regulations (e.g., GDPR)
- **Issues**
  - Malicious/compromised cloud provider or user can access it

Confidential Computing

- **Protects** data **confidentiality** and **integrity**
- Assumes **strong attacker**
  - **Controls hardware** and **software**, including OS/hypervisor
- Augments CPU with **secure component**
  - **Trusted Execution Environment (TEE)**
- Implementations
  - Intel **SGX, TDX**
  - AMD SEV, **SEV-SNP**
  - ARM TrustZone, **CCA**
  - RISC-V Keystone, PENGLAI



Confidential Computing Attestation

Problem

- How to make sure the **correct code** is executed?
- How to make sure the code is executed on the **correct hardware**?

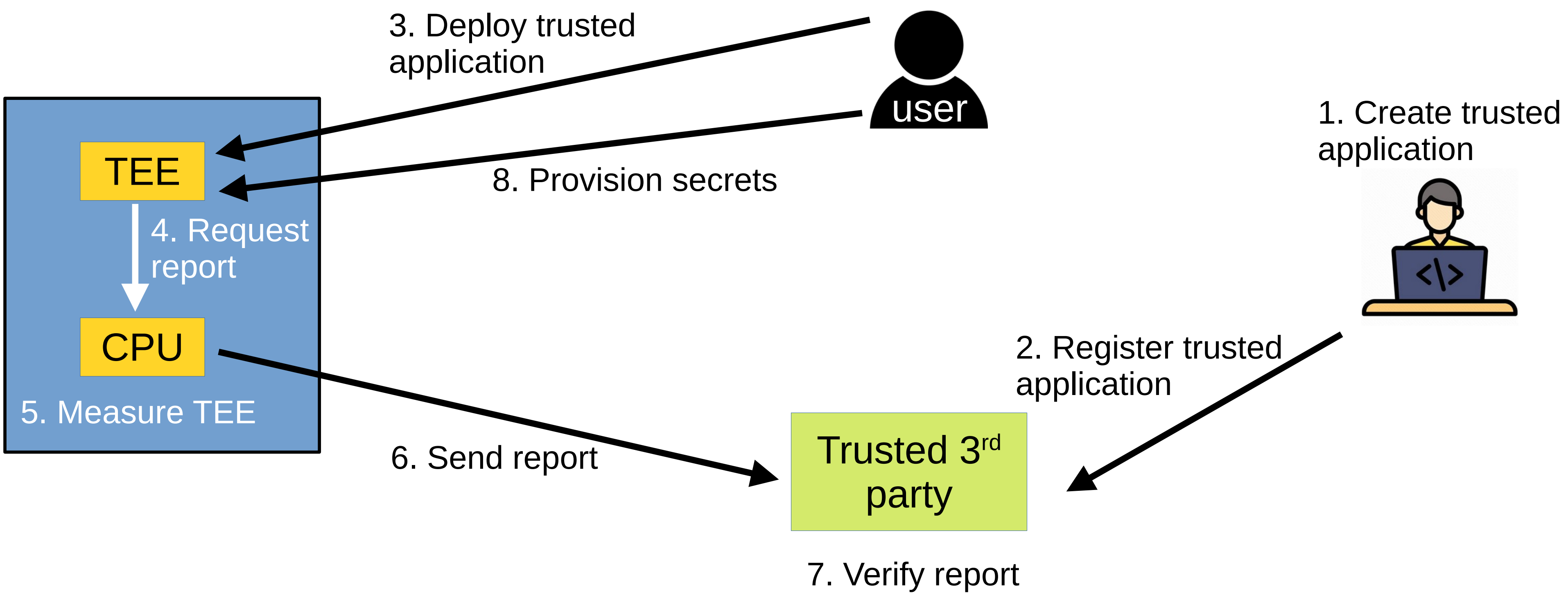
Solution: **attestation**

- CPU augmented with cryptographic **keys**
- Involves **Trusted 3rd party**

Process in a nutshell

- **CPU** produces **signed report**
- **3rd party checks report** validity

Confidential Computing Attestation Process

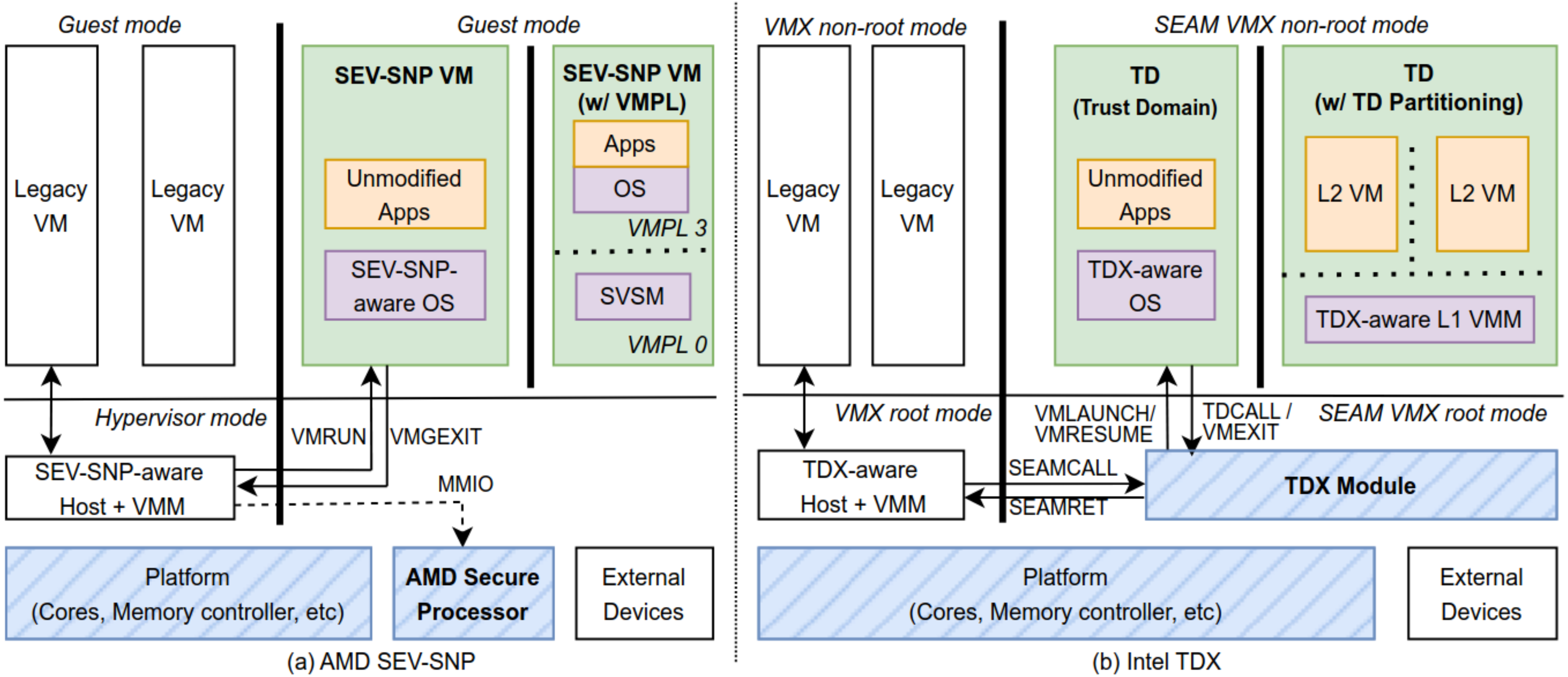


Two flavours of TEEs

- **Enclave-based**
  - Executes only **part of application** inside **TEE**
  - Need to **modify application**
  - **Small** Trusted Computing Base **(TCB)**
  - Intel SGX, ARM TrustZone

- **VM-based (cVM)**
  - Executes **entire VM** inside **TEE**
  - **No** application **modification**
  - **Large TCB**
  - Intel TDX, AMD SEV-SNP

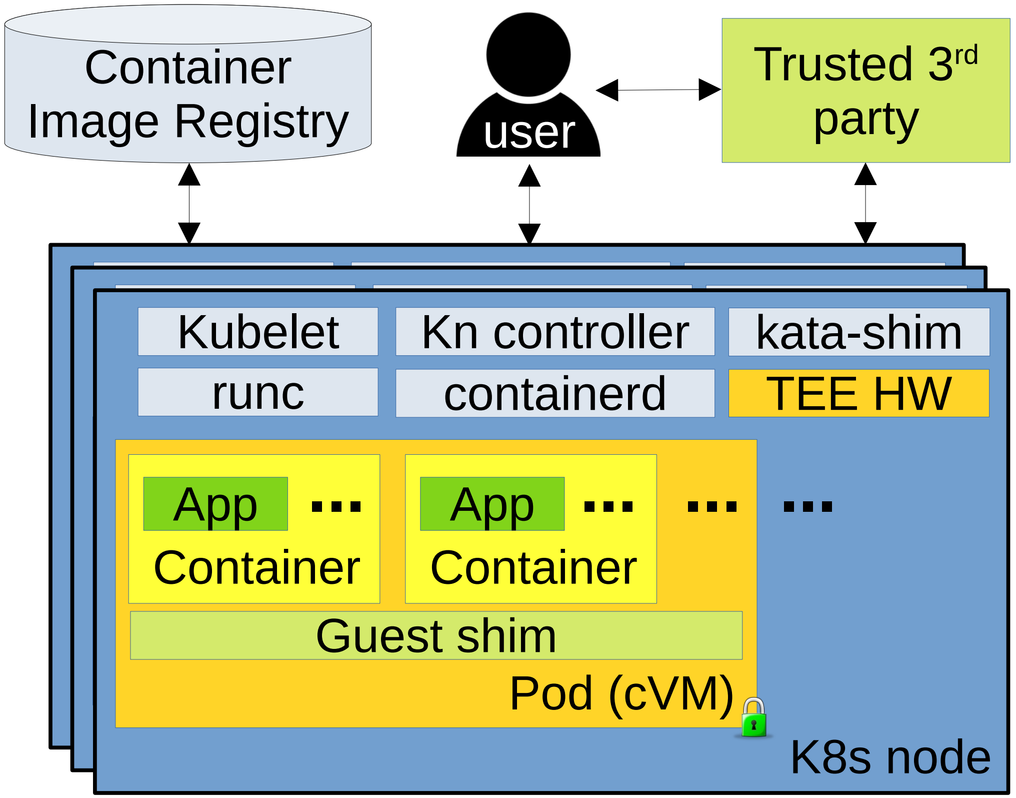
cVM TEEs



Misono et al. "**Confidential VMs Explained: An Empirical Analysis of AMD SEV-SNP and Intel TDX.**"
Proceedings of the ACM on Measurement and Analysis of Computing Systems 2024 Vol. 8 No. 3 Article 36

CC-Knative: Serverless Confidential Containers

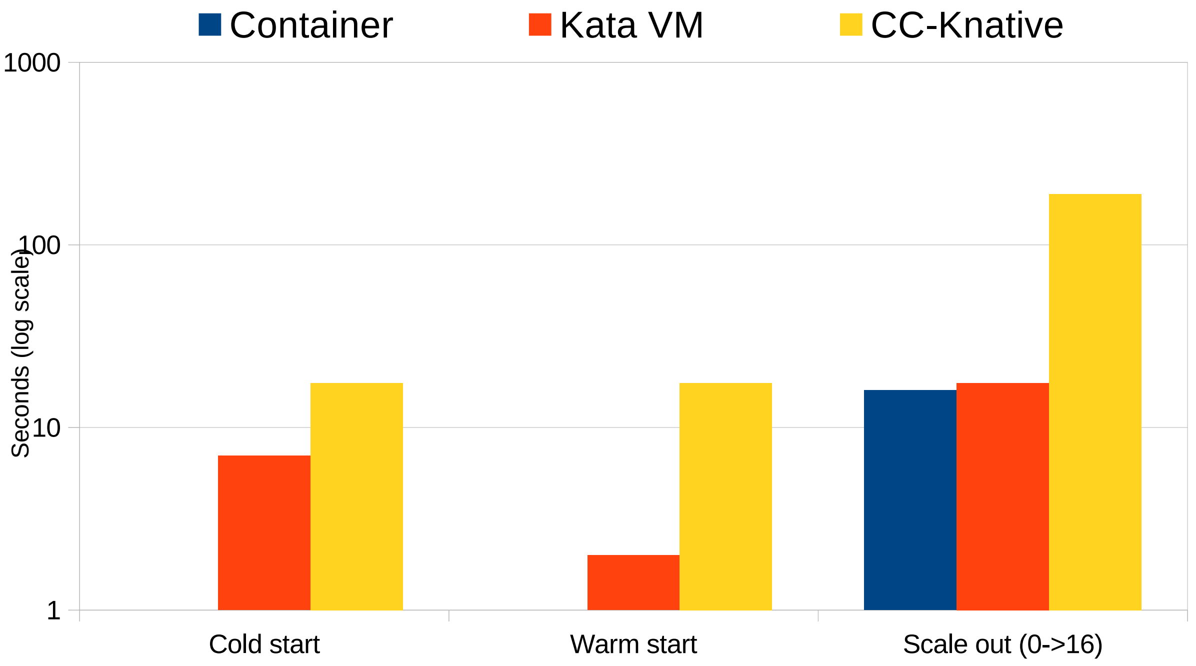
- Additional components:
  - **TEE** hardware
  - **Trusted 3rd party**: attestation & secrets provisioning
  - **Pod** is a **cVM**: provides confidentiality/isolation
  - **Kata containers**: "The speed of containers, the security of VMs"
  - Kata shim: executes containers inside cVM
  - Guest shim: host kernel, Kata endpoint, ...



Performance Evaluation Settings

- **Set-up**
  - Single-node: AMD EPYC 7763 CPU, 128 cores (SEV)
  - Kubernetes 1.28.2; CoCo 0.7.0, Knative 1.11.0
  - “Hello world” Knative service in Python (Knative demo)
- **Metrics**
  - Cold start
  - Warm start
  - Scale-up latency
- **Systems**
  - **runc**: default in Knative; uses containers
  - **kata**: executes each pod in a VM
  - **CC-Knative**: kata + attestation + encryption

Performance Evaluation Summary



CC-Knative overhead compared to containers:

- Cold start: 2.5x
- Warm start: 8.5x
- Scale out: 11x

Overheads

- **Cold start**
  - Guest VM pins all memory (specific to AMD SEV)
  - VM image decryption
- **Warm start**
  - Not possible to reuse VM for security reasons
  - Not possible to pre-warm VMs with AMD SEV
- **Scale up**
  - Pulling images throttled down by remote registry
  - Relying on local registry faster, but less secure (cloud provider controls it)

Proposed Optimisations

- 2 categories
  - VM **creation**
  - VM **provisioning**
- **Optimise image deployment**
  - One part confidential (never cached locally)
  - One part non-confidential (cached locally)
- **Pre-warm the VM**
  - Start VMs in advance
  - Problems:
    - Max #cVM limited by hardware
    - Attestation

- **Cache cVMs** with **Trusted Monitor**
  - Rely on **virtual TPM**
    - Tamper-resistant secure crypto-processor
  - Trusted monitor executes inside cVM at higher privilege than application
  - Attestation via vTPM
  - **Confidential Restore**: restore cVM to clean state for re-use

Future Works

- Port to AMD SEV-SNP and Intel **TDX**
  - Done
- Implement the **optimisations**
- Implement the vTPM and attestation
- In-depth **evaluation**
  - Multiple nodes setting
  - ...

Conclusion

- **Serverless** Computing **needs Confidential** Computing
- We propose **CC-Knative**: **Knative** serverless framework port to **cVMs**
- Stay tuned!
  - https://github.com/sc2-sys

code

contact