

サーバーレス機密コンテナ

Serverless Confidential Containers

English slides



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

urlr.me/srJf7T

2025 年 5 月 19 日
IIJ Research Lab Seminars

謝辞

この発表は主に以下の論文に基づいています：

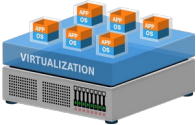
Segarra, C., Feldman-Fitzthum, T., Buono, D., & Pietzuch, P.

Serverless confidential containers: Challenges and opportunities.

In Proceedings of the 2nd Workshop on SErverless Systems, Applications and Methodologies (SESAME), April 2024.



クラウドコンピューティングレイヤー

				
	ユーザーメンテナンス	速度	スケーラビリティ	価格
サーバーレス 	— — — —	++++	++++	¥
コンテナ 	+	+++	++	¥ ¥
仮想化 	++	++	++	¥ ¥ ¥
ベアメタル 	++++	— — — —	— — — —	¥ ¥ ¥ ¥



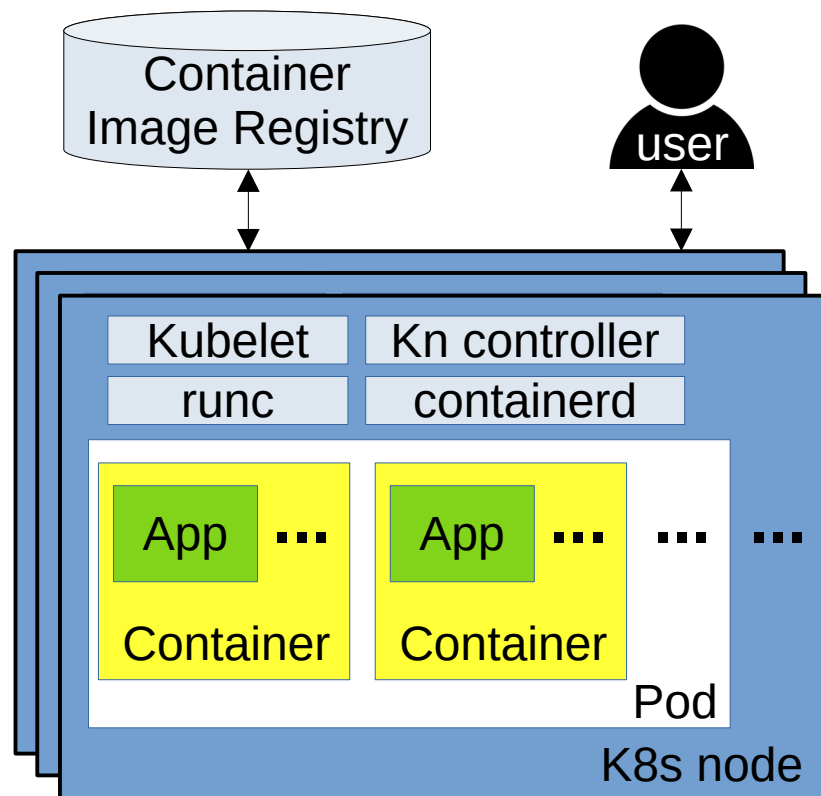
サーバーレスコンピューティングとは

- 開発者はクラウド上でアプリケーションを開発・デプロイします
- クラウドプロバイダーはインフラストラクチャ全体を提供します
 - アプリケーションランタイム、データストア、認証、監視、スケーリング。。。。
- クラウドプロバイダーは複数のユーザーを同じ場所に配置できます
- 複数のフレームワーク
 - OpenFAAS <https://www.openfaas.com/>
 - Apache OpenWhisk <https://openwhisk.apache.org/>
 - **Knative** <https://knative.dev/docs/>



Knative サーバーレス コンピューティング フレームワーク

- Kubernetes (==K8s) 上に構築
- コンポーネント :
 - **Container Image Registry**: コンテナイメージ (アプリケーションコードを含む) を管理
 - **Kubelet**: ワーカーノードごとに1つ、コンテナを管理
 - **Pod**: 基本的なスケジューリング / 分離ユニット。同一ノードに1つ以上のコンテナを配置
 - **Kn controller**: コントロールプレーンポッド
 - **containerd**: コンテナランタイム
 - **runc**: コンテナを実行するツール



機密性の高いサーバーレスコンピューティングの必要性

- **サーバーレスユースケース**

- 画像処理
- 機械学習
- 医療研究
- 取引不正検知
- . . .

- **機密性に関する懸念**

- ユーザーデータは機密情報 / 個人情報です
- セキュリティ攻撃
- 政府主導の大規模監視プログラム
- データ保護規制（例：GDPR）

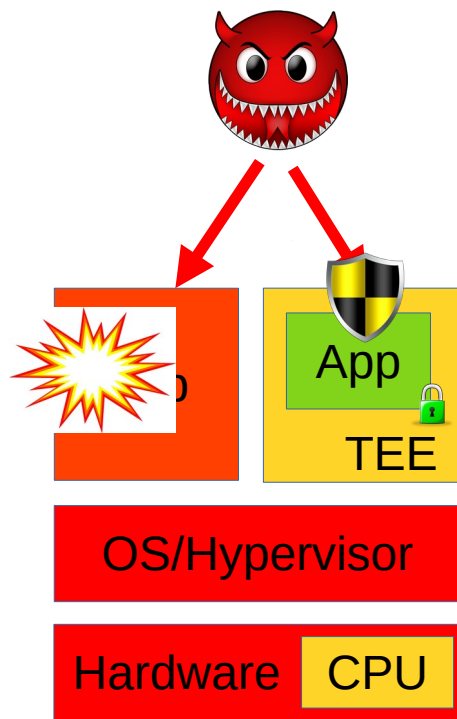
- **問題**

- 悪意のある / 侵害されたクラウドプロバイダーまたはユーザーがアクセスできる可能性があります



機密コンピューティング

- データの機密性と整合性を保護
- 強力な攻撃者を想定
 - OS/ ハイパーバイザーを含むハードウェアとソフトウェアを制御
- セキュアコンポーネントで CPU を拡張
 - **Trusted Execution Environment (TEE)**
- 実装
 - Intel **SGX**, **TDX**
 - AMD **SEV**, **SEV-SNP**
 - ARM TrustZone, **CCA**
 - RISC-V Keystone, PENGGLAI



機密コンピューティングの認証

問題

- 正しいコードが実行されていることをどのように確認するか？
- コードが正しいハードウェア上で実行されていることをどのように確認するか？

解決策：認証

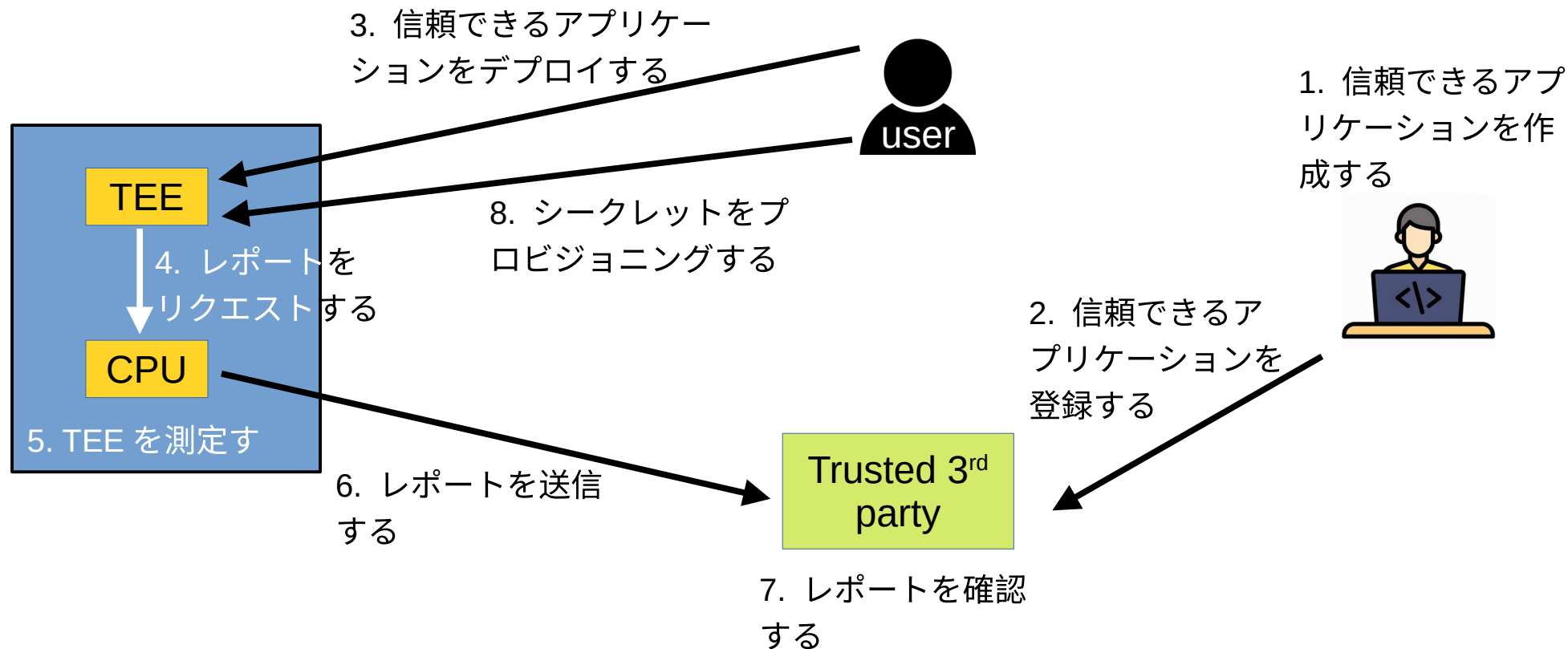
- CPU に暗号鍵を追加
- 信頼できる第三者機関が関与

プロセスの概要

- CPU が署名付きレポートを作成
- 第三者機関がレポートの妥当性を確認



機密コンピューティングの認証プロセス



TEE には 2 つの種類があります

- エンクレーブベース

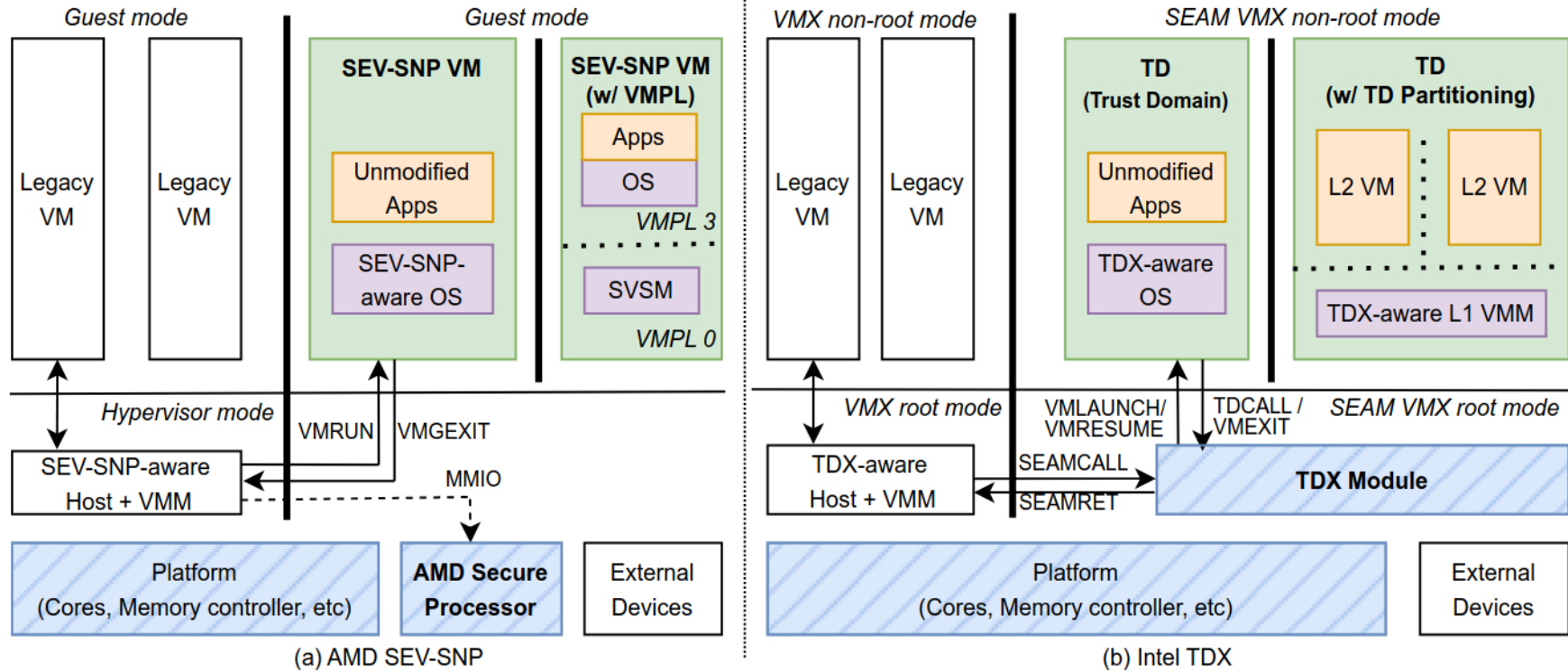
- TEE 内でアプリケーションの一部のみを実行
- アプリケーションの**変更が必要**
- **小規模な Trusted Computing Base (TCB)**
- Intel SGX, ARM TrustZone

- VM ベース (cVM)

- TEE 内で VM 全体を実行
- アプリケーションの**変更は不要**
- **大規模な TCB**
- Intel TDX, AMD SEV-SNP



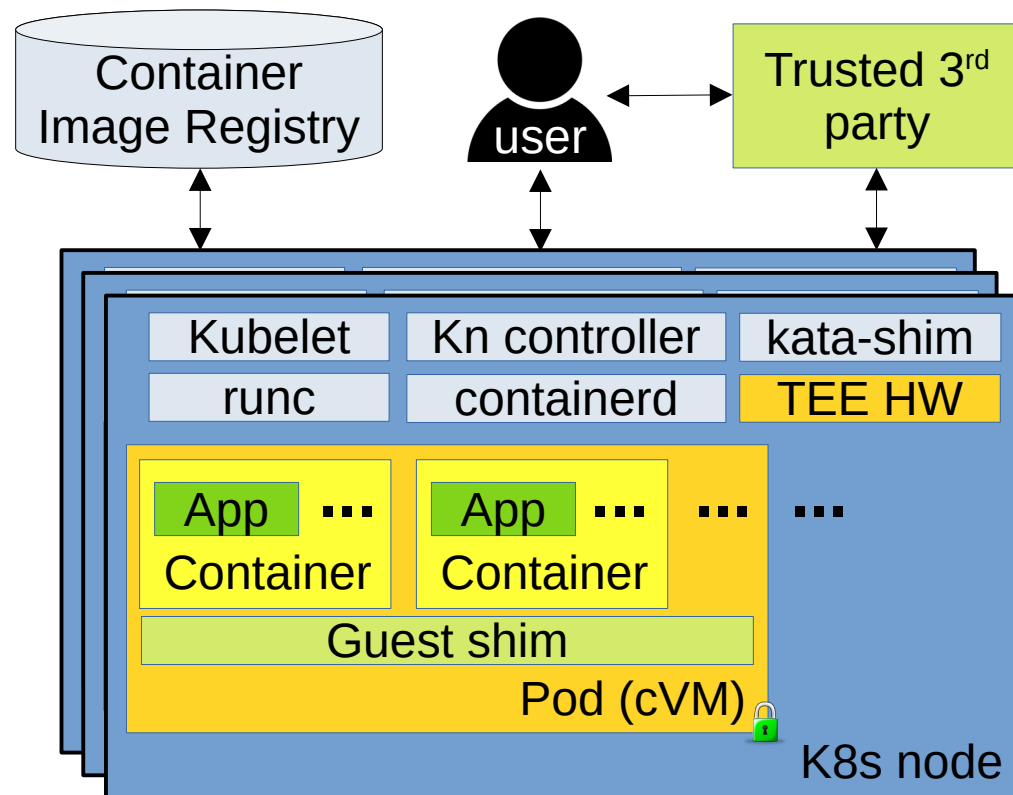
cVM TEEs



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: サーバーレス機密コンテナ

- 追加コンポーネント：
 - TEE ハードウェア
 - Trusted 3rd party**: 構成証明とシークレットのプロビジョニング
 - Pod** は **m**: 機密性 / 分離を提供
 - Kata コンテナ**: 「コンテナのスピード、VM のセキュリティ」
 - Kata shim: cVM 内でコンテナを実行
 - Guest shim: ホストカーネル、Kata エンドポイントなど



パフォーマンス評価設定

- セットアップ

- シングルノード : AMD EPYC 7763 CPU、128 コア (SEV)
- Kubernetes 1.28.2、CoCo 0.7.0、Knative 1.11.0
- Python で実行した「Hello world」 Knative サービス (Knative デモ)

- メトリクス

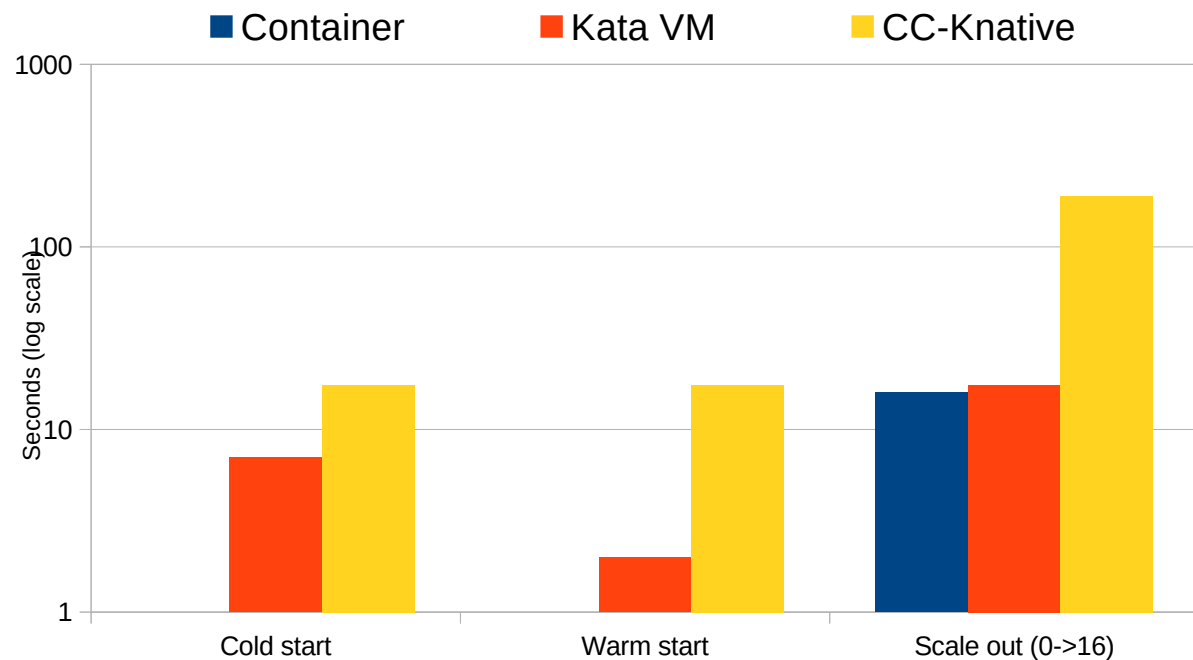
- コールドスタート
- ウォームスタート
- スケールアップレイテンシ

- システム

- **runc**: Knative のデフォルト。コンテナを使用
- **kata**: 各ポッドを VM で実行
- **CC-Knative**: kata + 構成証明 + 暗号化



パフォーマンス評価の概要



コンテナと比較した CC-Knative のオーバーヘッド:

コールドスタート: 2.5 倍

ウォームスタート: 8.5 倍

スケールアウト: 11 倍



諸経費

- **コールドスタート**

- ゲスト VM はすべてのメモリを固定します（AMD SEV に特有）
- VM イメージの復号

- **ウォームスタート**

- セキュリティ上の理由から VM の再利用はできません
- AMD SEV では VM を事前にウォームアップすることはできません

- **スケールアップ**

- リモートレジストリによってイメージのプルが抑制されます
- ローカルレジストリを利用するため高速ですが、セキュリティは低くなります（クラウドプロバイダーが管理しています）



提案された最適化

- 2つのカテゴリ
 - VM 作成
 - VM プロビジョニング
- **イメージの展開を最適化**
 - 一部は機密情報（ローカルにキャッシュされない）
 - 一部は非機密情報（ローカルにキャッシュされる）
- **VM の事前ウォームアップ**
 - VM を事前に起動
 - 問題点：
 - cVM の最大台数はハードウェアによって制限される
 - 構成証明
- **Trusted Monitor による VM のキャッシュ**
 - 仮想 TPM の利用
 - 耐タンパ性を備えたセキュアな暗号プロセッサ
 - Trusted Monitor は cVM 内でアプリケーションよりも高い権限で実行される
 - vTPM による構成証明
 - **機密情報の復元**: cVM をクリーンな状態に復元して再利用



今後の作品

- AMD SEV-SNP および Intel TDX への移植
 - 完了
- 最適化の実装
- 仮想 TPM、認証
- 詳細評価
 - 複数ノード設定
 - ...



結論

- サーバーレスコンピューティングには機密コンピューティングが必要
- CC-Knative を提案します: Knative サーバーレスフレームワークを cVM に移植
- 今後の展開にご期待ください
 - <https://github.com/sc2-sys>

コード



連絡先

