SUSTech Southern University of Science and Technology

Teecert Labs

Revolutionizing System Support:

Supporting Firecracker Virtualization for Jinux Platform

Group project
Ruixiang JIANG – Report & PPT
Wenqian YAN - Presentation

Outline

Teecert Labs

SUSTech Southern University of Science and Technology

Firecracker is an open-source virtualization technology developed by Amazon Web Services (AWS), tailored for the modern cloud computing landscape.

At the same time, Jinux is a secure, fast, and general-purpose OS kernel, written in Rust and providing Linux-compatible ABI.

Our propose is to support Firecracker virtualization for Jinux platform.

Catalogue

Teecert Labs

SUSTech Southern University of Science and Technology

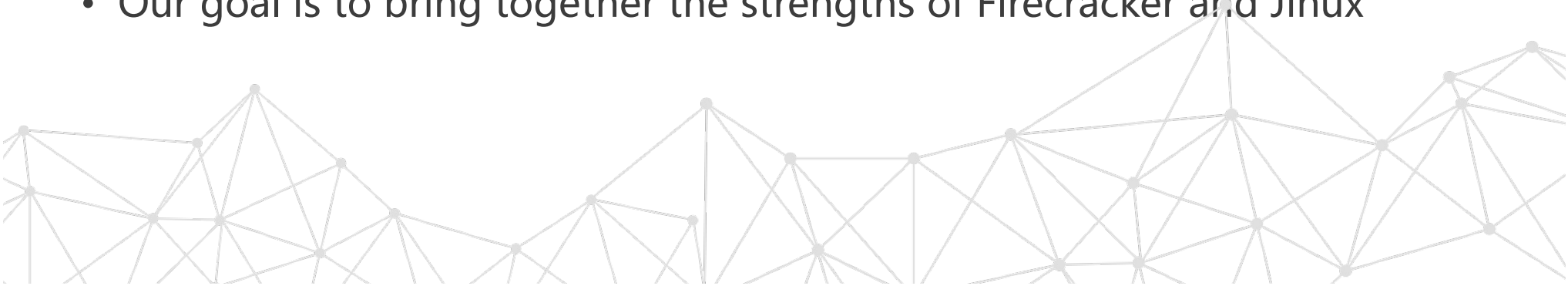
SUSTech Southern University of Science and Technology

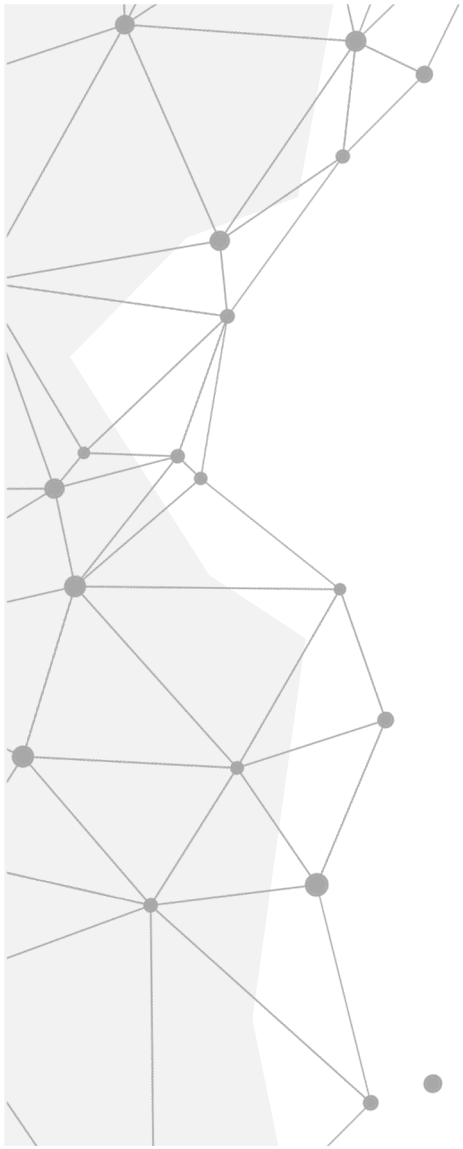
01

Introduction

Introduction

- Firecracker, an open-source virtualization technology, enables efficient virtualization and rapid deployment of virtual machines.
- Jinux, written in Rust, is an exceptionally secure and high-performance operating system kernel.
- Our goal is to bring together the strengths of Firecracker and Jinux

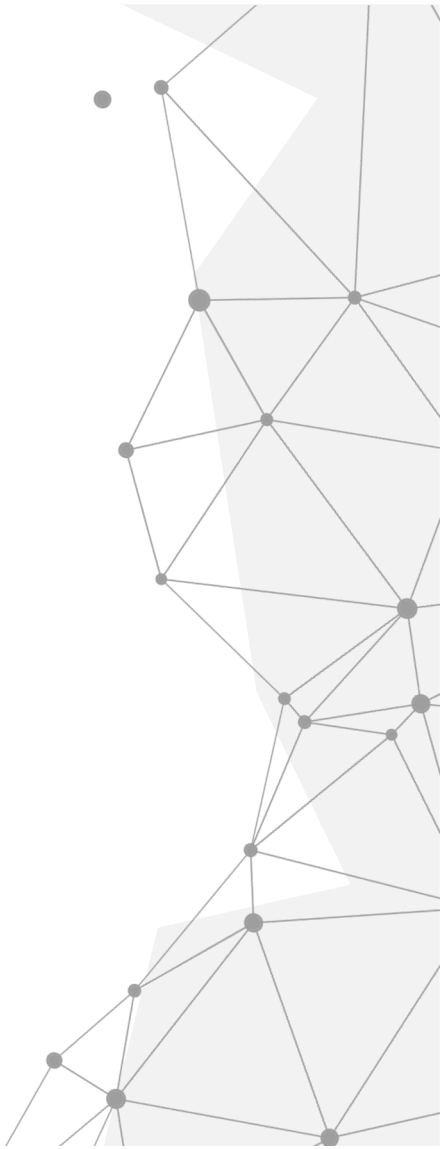




SUSTech Southern University of Science and Technology

02

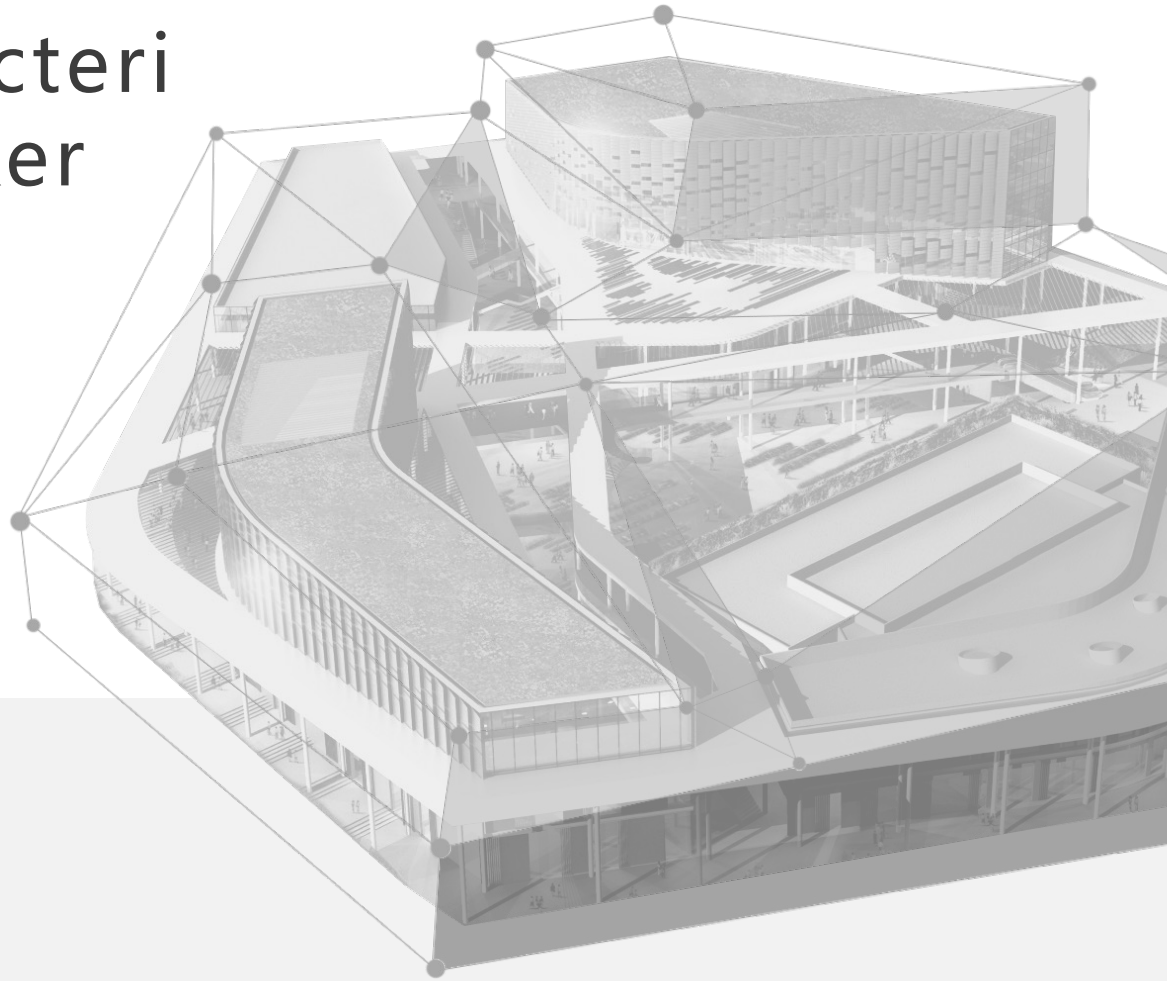
Firecracker



Distinctive Characteristics of Firecracker

Demand is fast and efficient virtualization.

Firecracker, introduced by AWS, offers a unique approach to meet the demand.



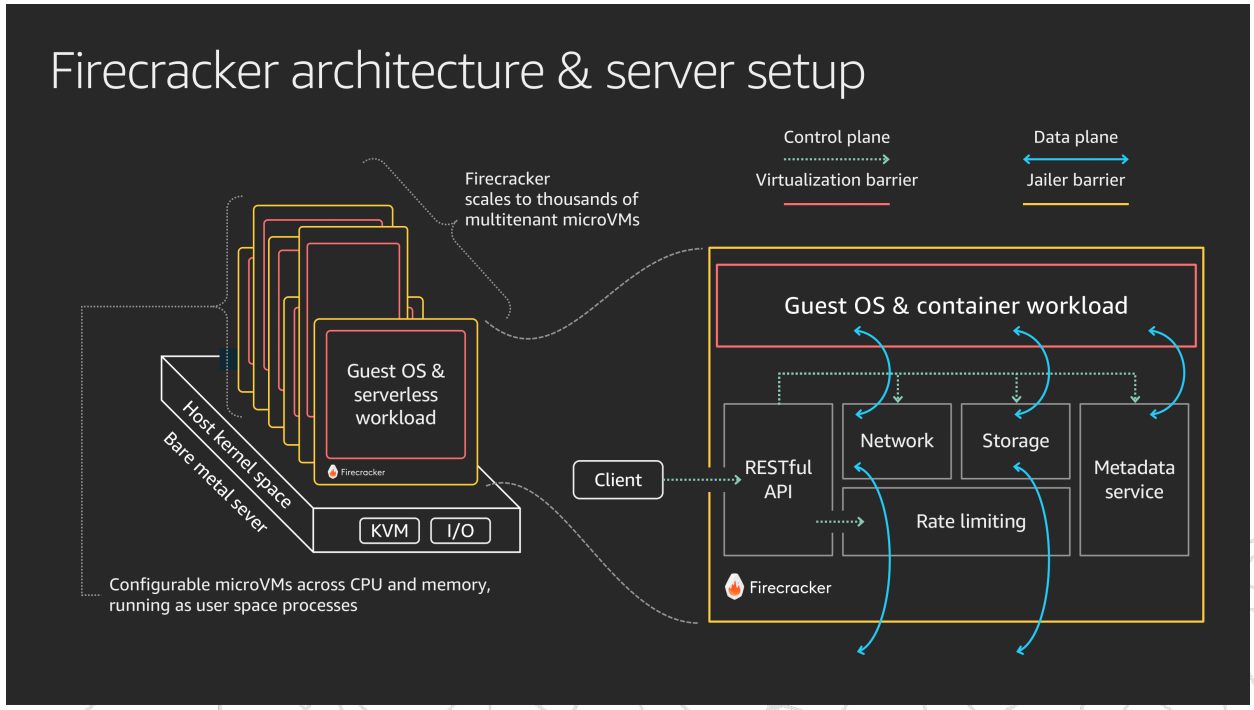
SUSTech Southern University of Science and Technology

Teecert Labs

Firecracker

Rapid Boot Time

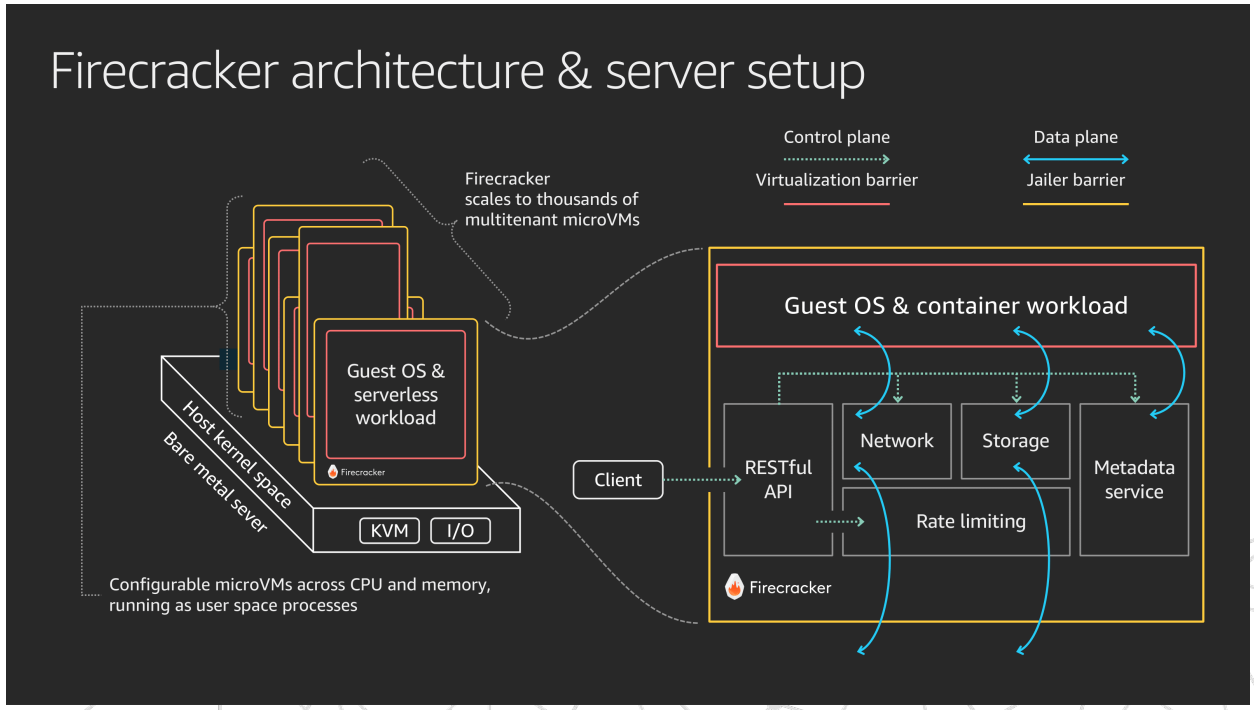
- MicroVM Architecture
- Firecracker Kernel
- Just-In-Time Initialization
- Pre-Allocated Resources
- Reduced Emulation Overhead
- Single-Purpose VMs



Firecracker

Resource Efficiency

- Low Memory Footprint
- Multi-Tenant Isolation
- Resource C-group Management
- Efficient Component Initialization



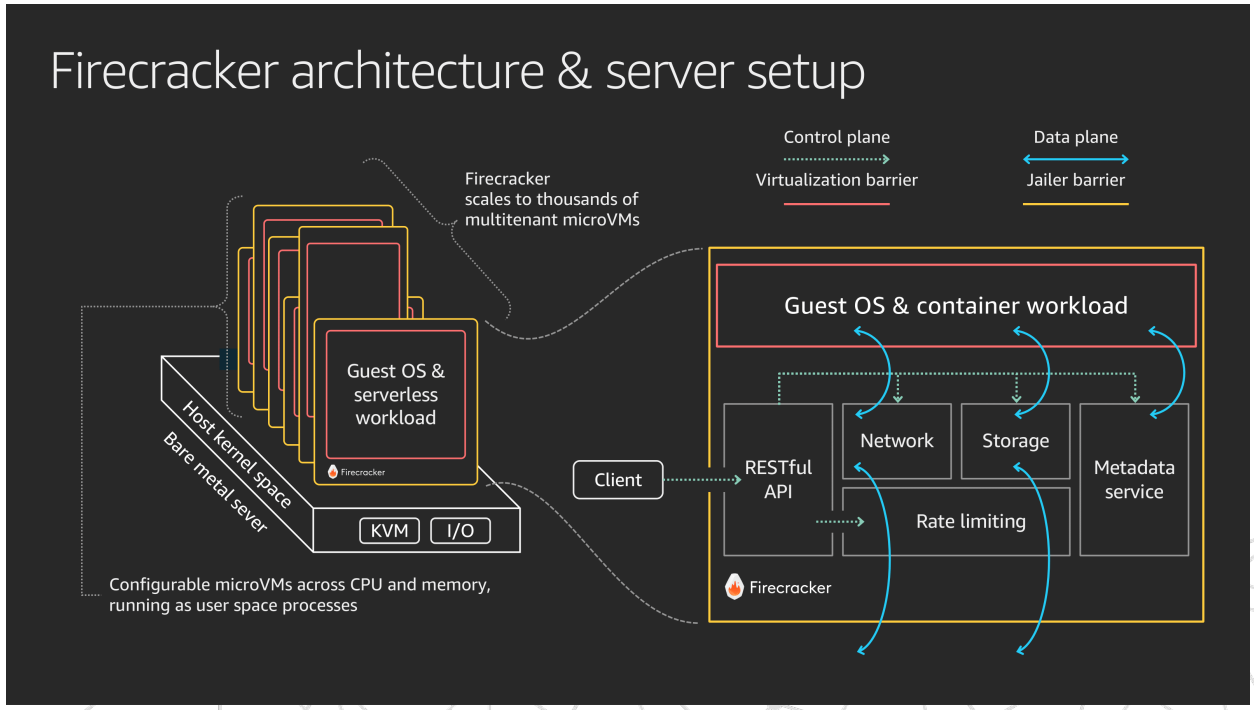
Firecracker

Teecert Labs

SUSTech Southern University of Science and Technology

High security

- Isolation
- Minimal Attack Surface
- Customizable Security Policies
- Monitoring and Auditing



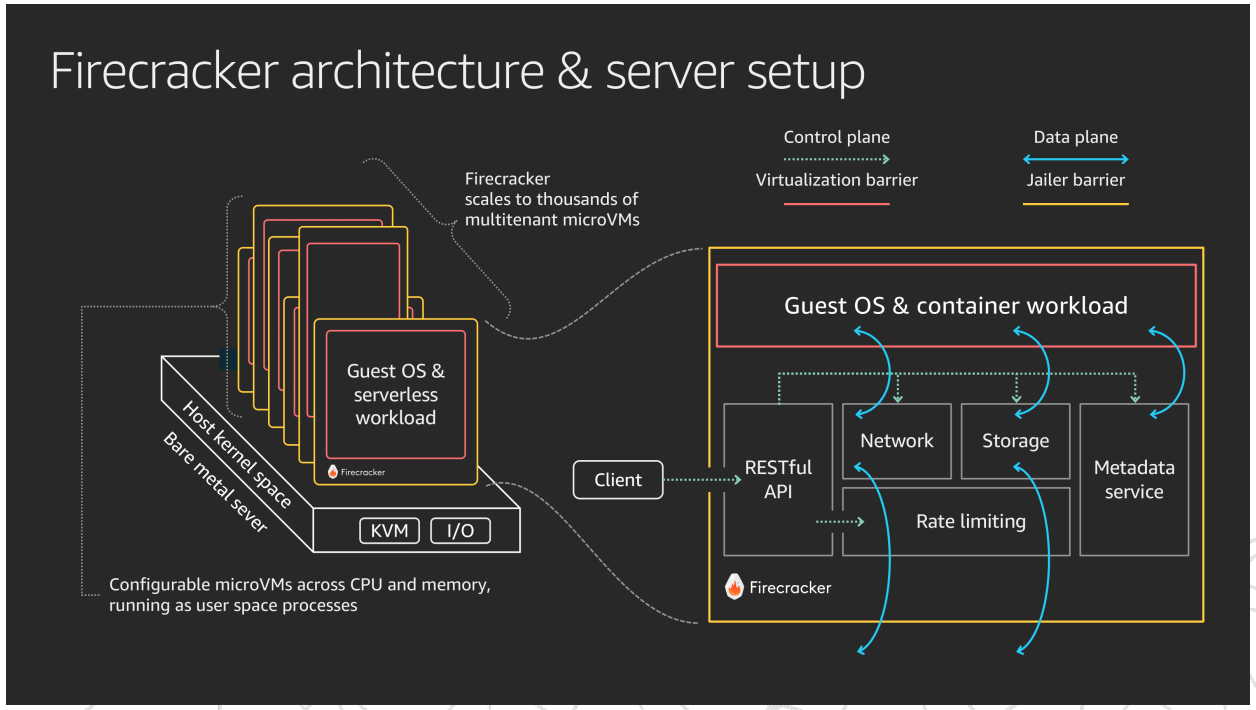
Firecracker

Teecert Labs

SUSTech Southern University of Science and Technology

Container Support

- Docker
- Kubernetes

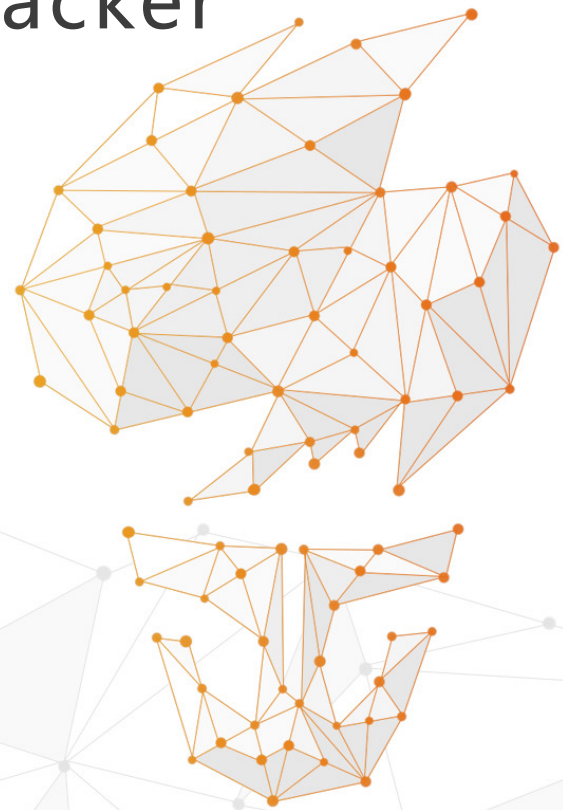


Distinctive Characteristics of Firecracker

Teecert Labs

SUSTech Southern University of Science and Technology

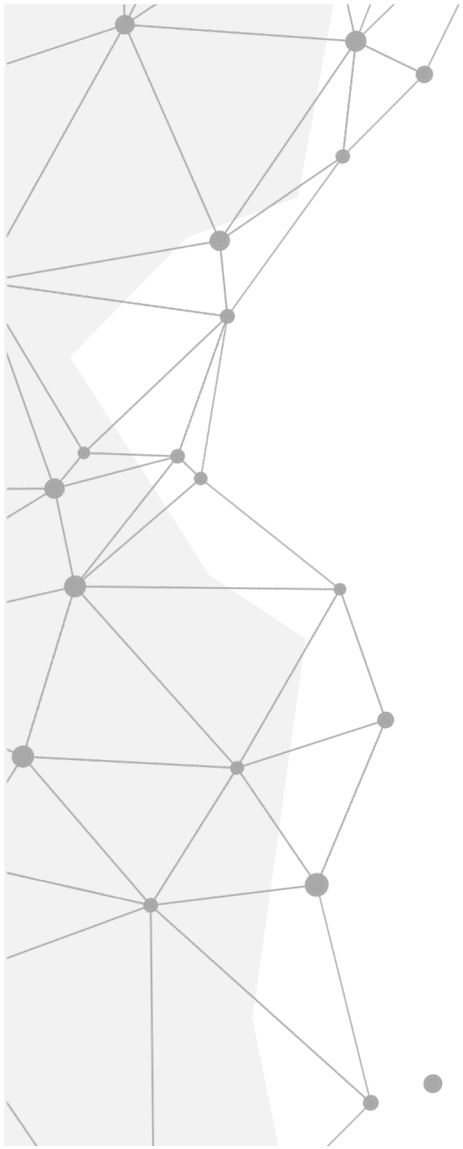
Rapid boot time



Collaboration, innovation, and transparency

Resource efficiency

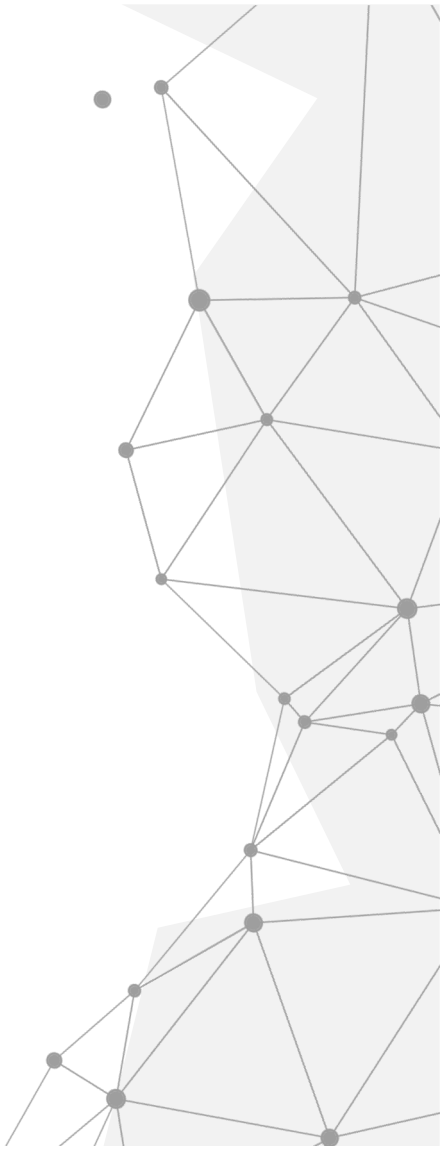
Seamless integration with container technologies



SUSTech Southern University of Science and Technology

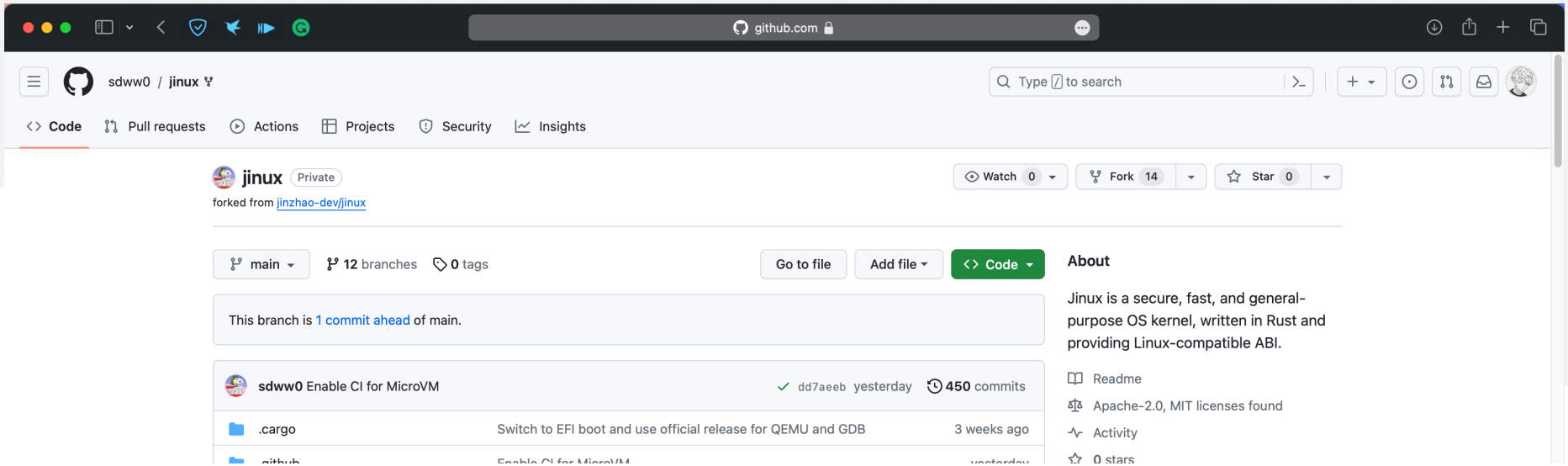
03

Jinux



SUSTech Southern University of Science and Technology

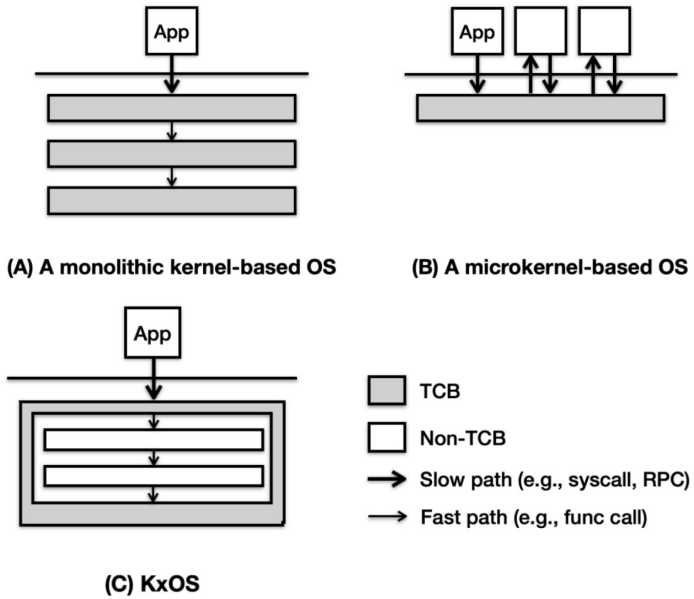
Teecert Labs



Jinux

Rust, Least privilege, High performance standards

Why Jinux distinctive



Architectural Comparison

The architectural level:

- Single address space
- Restricting unsafe code

The component level

- Containing safe Rust code
- Be governed by Jinux Component System

The object level

All kernel resources are accessed through capabilities

Why Jinux distinctive

1

Security

Least privilege principle

Everything is a capability

2

Virtualization

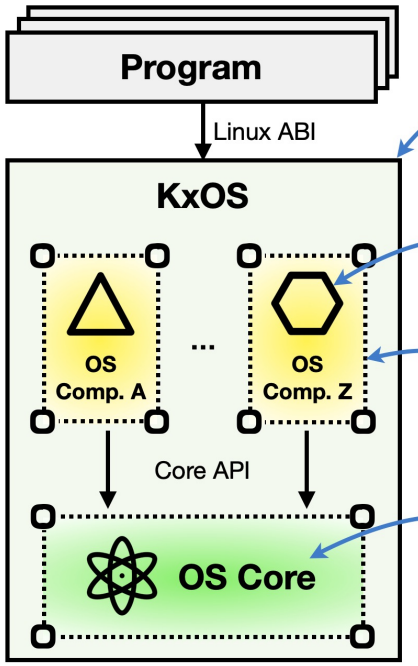
Trustworthy OS-level virtualization platform

3

APIs

Regular OS kernel space

Library OS in user space



Single address space

Like a monolithic kernel, KxOS allows *efficient* communication between all parts of OS.

Runnable in user space

As in an anykernel, OS components can be developed and tested in user space.

Least privileged principle

OS components are access controlled with capabilities, making the OS more reliable.

Small and well-defined TCB

Like a microkernel for KxOS, the OS core exposes the hardware capabilities in a safe and portable fashion. It largely frees the rest of the OS from writing *unsafe* Rust code.

SUSTech Southern University of Science and Technology

SUSTech Southern University of Science and Technology

04

Our work

Teecert Labs

SUSTech Southern University of Science and Technology

Firecracker Integration

Libraries, APIs, and configuration options

MicroVM Configuration

Adjust them to align with Firecracker's requirements

Device Emulation

Add emulated devices to the Jinux environment to ensure proper functionality

Security Considerations

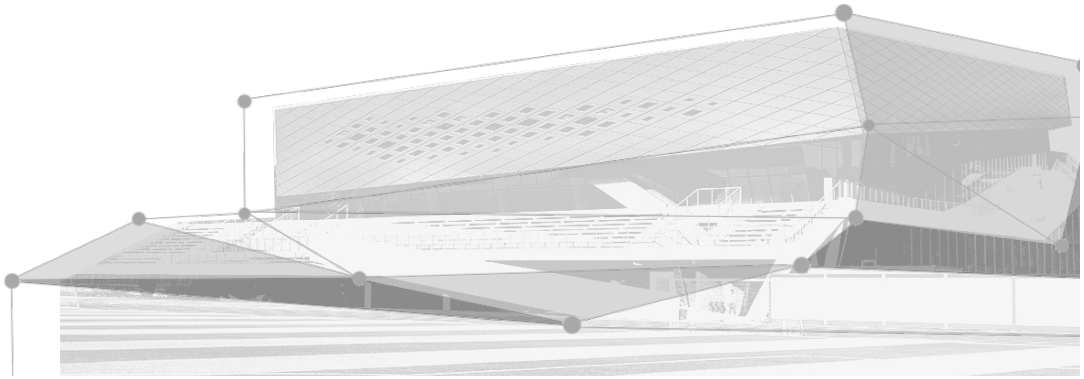
Enhancement Jinux codebase to meet Firecracker's stringent security standards

Several challenges

- Codebase Compatibility
- Device Emulation
- Security Enhancements

Our work

Modify the Jinux codebase to support execution within the Firecracker



Teecert Labs

SUSTech Southern University of Science and Technology

Thanks

[1] "Firecracker open-source innovation", DevelopersIO. [Online]. Available: https://dev.classmethod.jp/articles/reinvent2019-opn402

[2] "Jinux", Github. [Online]. Available: https://github.com/sdww0/jinux

[3] "Introduction", Github. [Online]. Available: https://github.com/sdww0/jinux/tree/main/docs/src