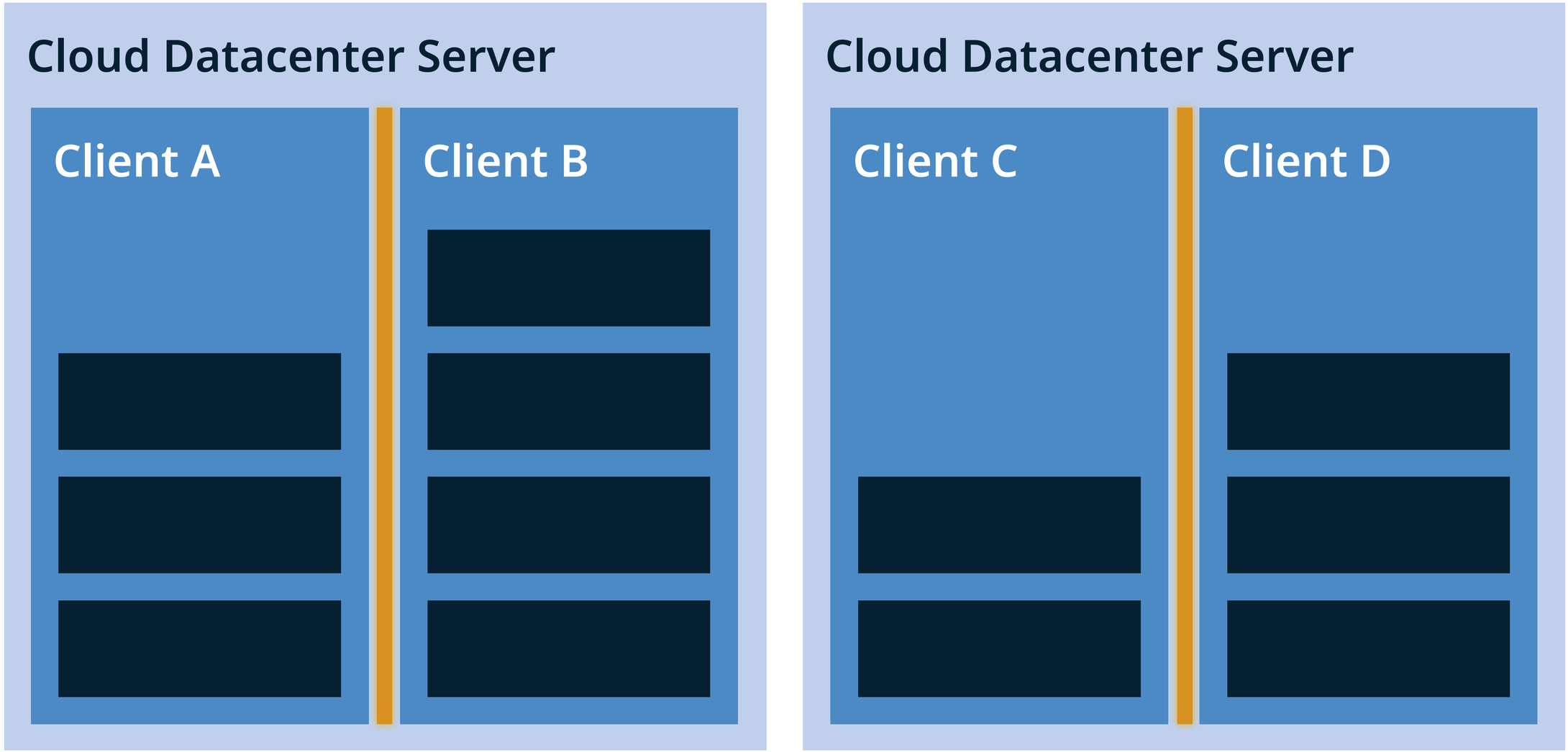
barkhauseninstitut.org

barkhausen institut

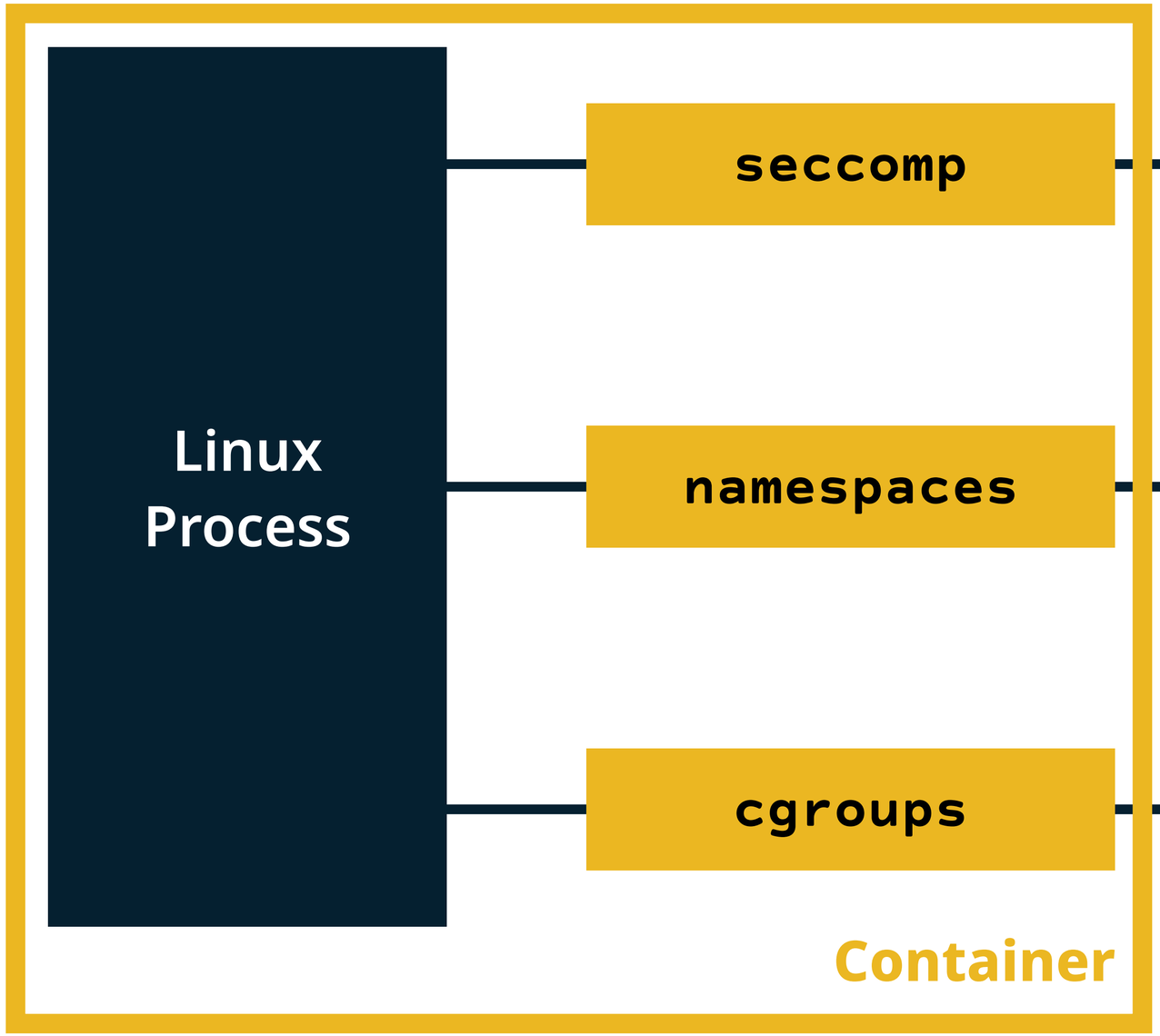
A Perfect Fit? – Towards Containers on Microkernels

Till Miemietz, Viktor Reusch, Matthias Hille, Max Kurze, Adam Lackorzynski, Michael Roitzsch, Hermann Härtig

Container Origin Story



The Weaknesses of Process Isolation



large kernel API available via system calls

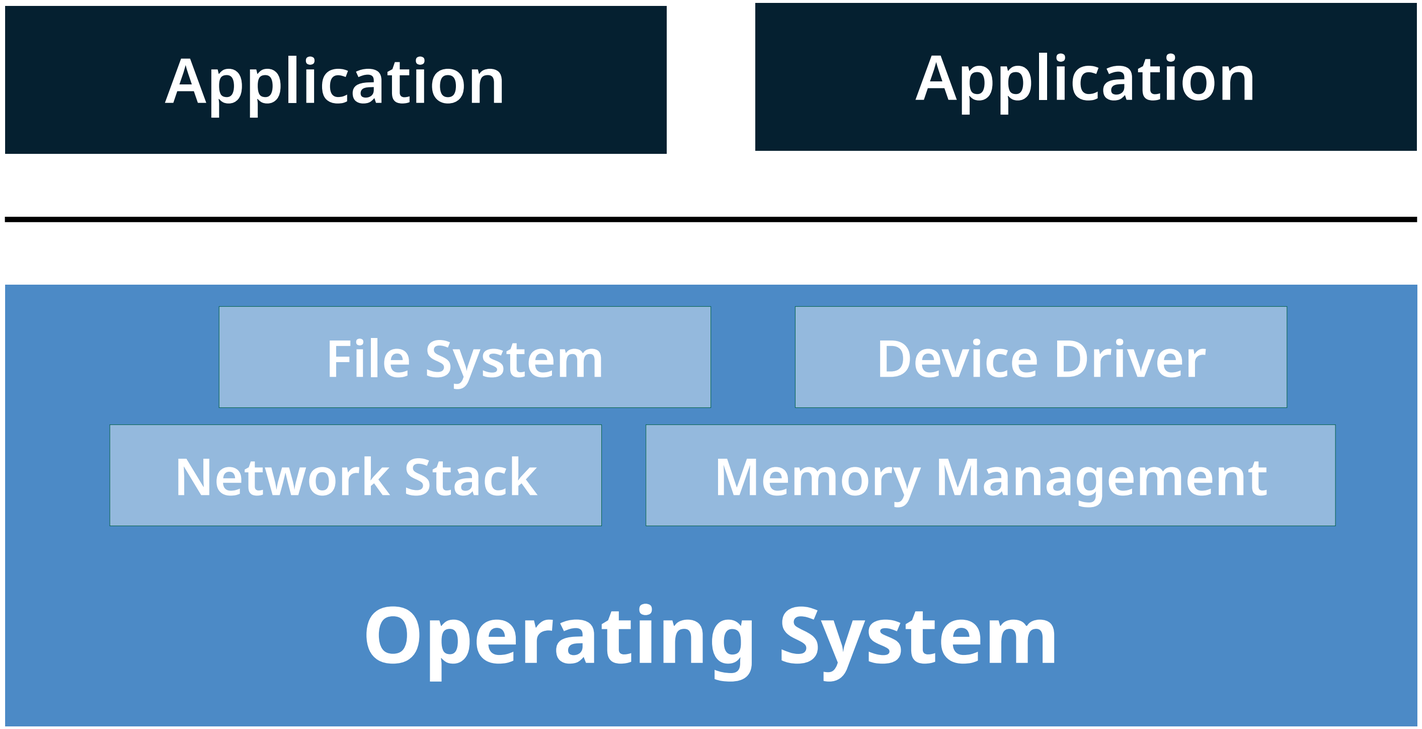
resources of other clients globally visible

resource consumption impact across clients

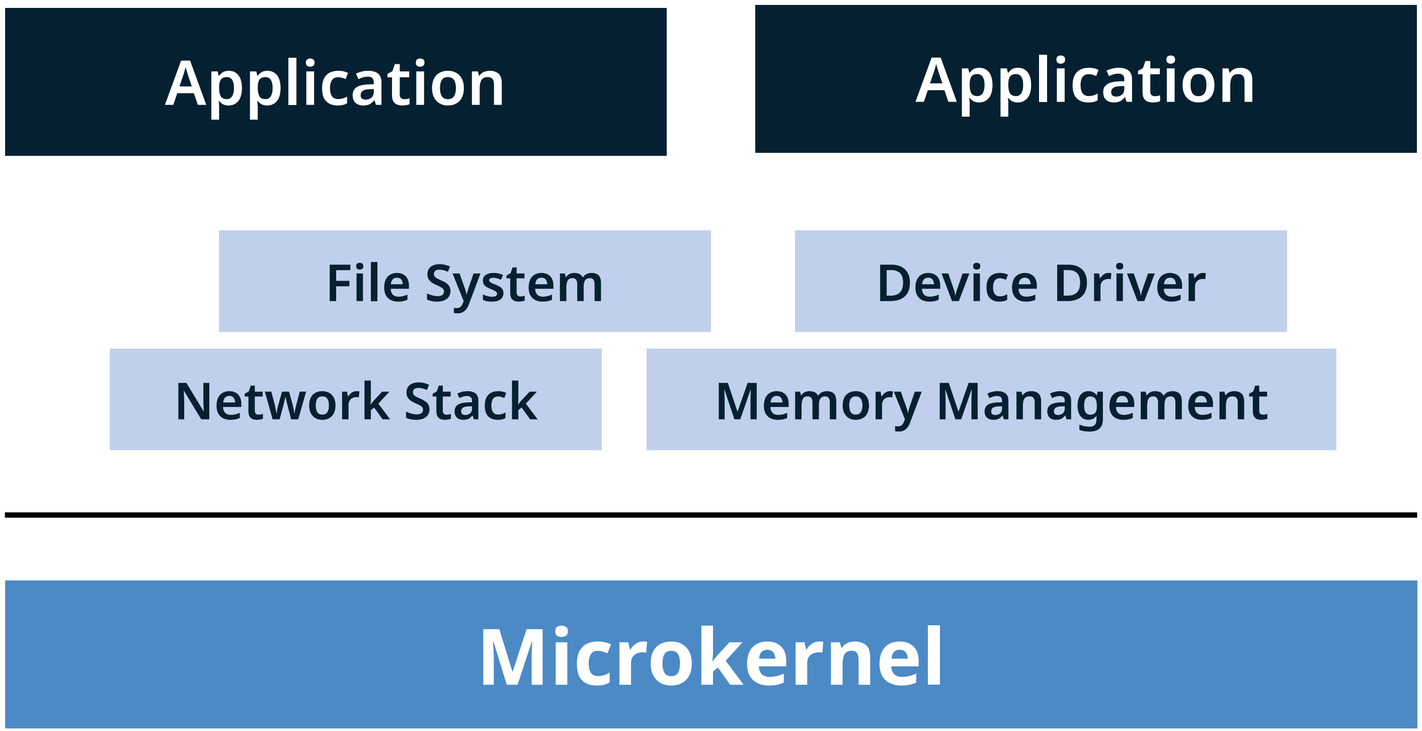
Complexity is the Enemy of Security

- **retroactively strengthening** a weakly isolated primitive feels wrong
- `seccomp`, `namespaces`, `cgroups` are **complex** Linux kernel subsystems
- bugs in these subsystems lead to **exploitable** security problems

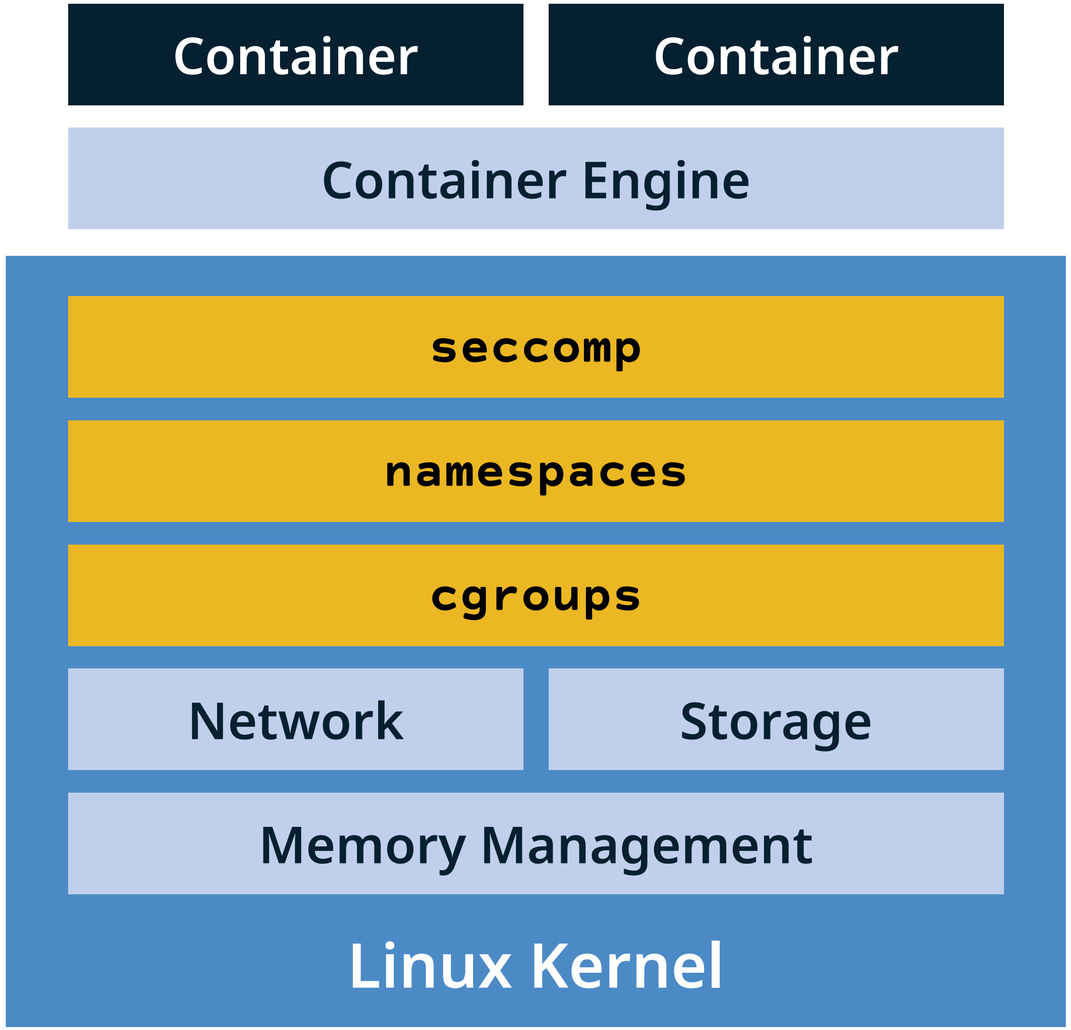
Classical Monolithic OS Design

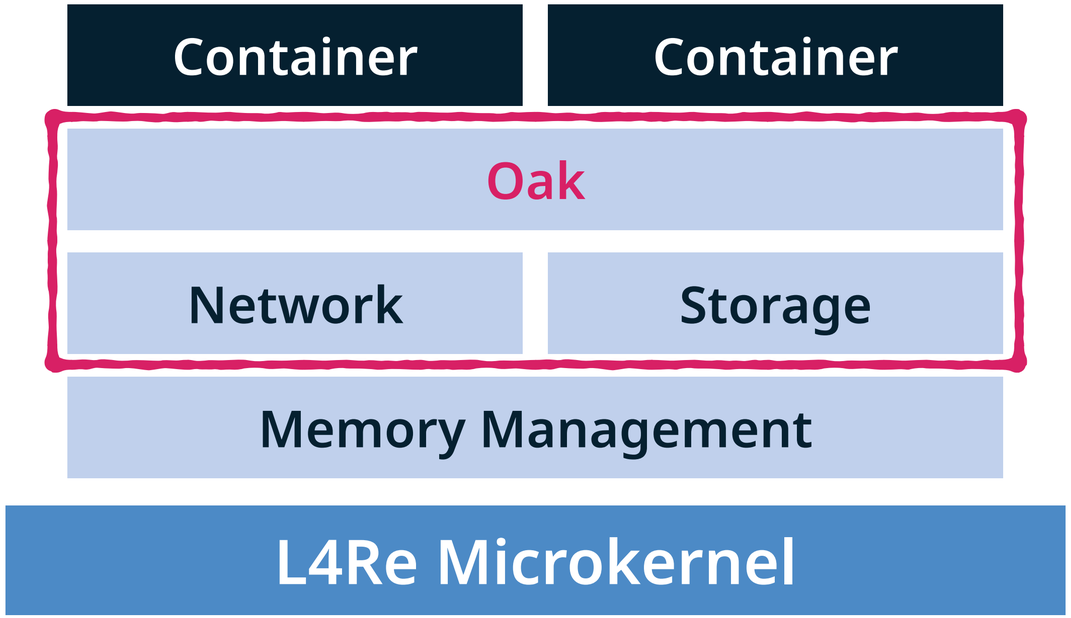


The Microkernel Idea



Containers on a Microkernel

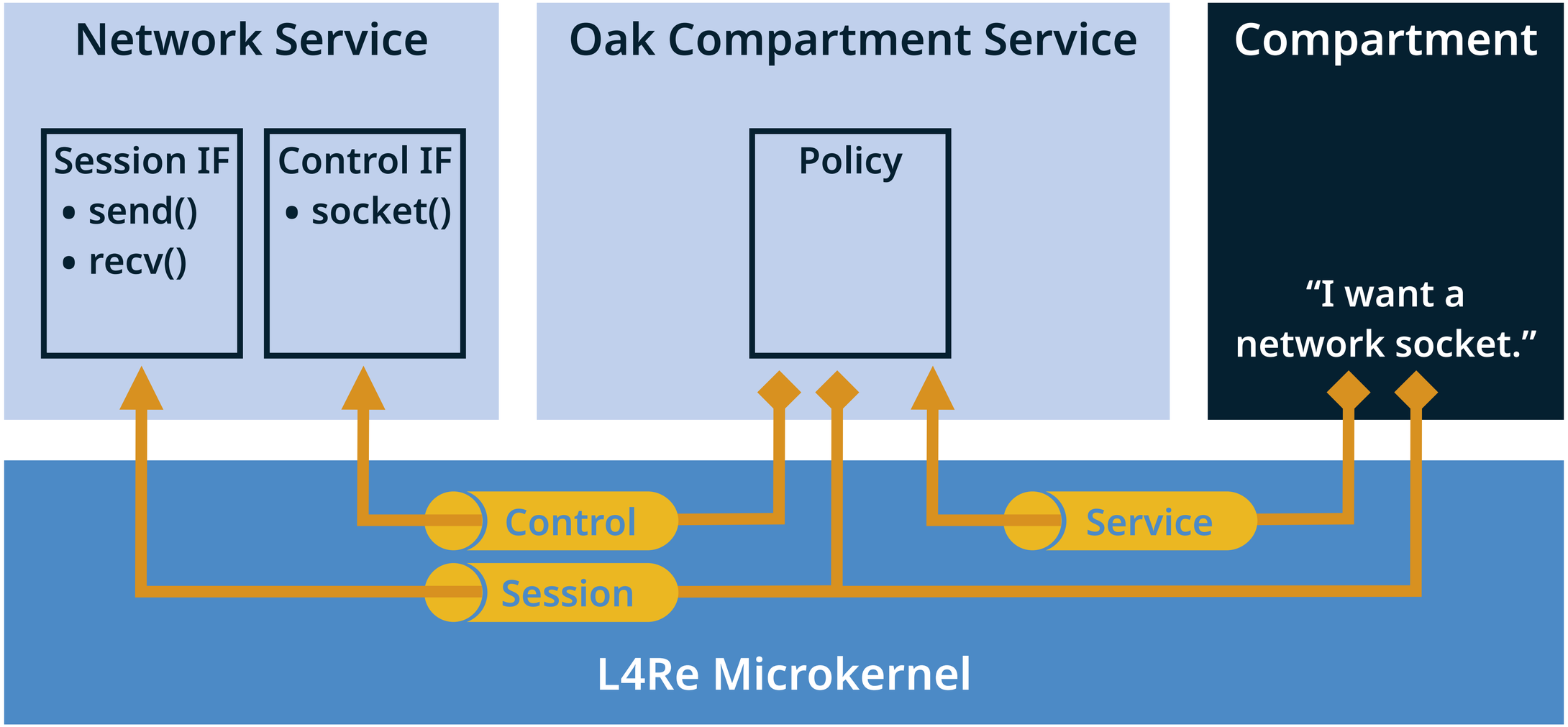




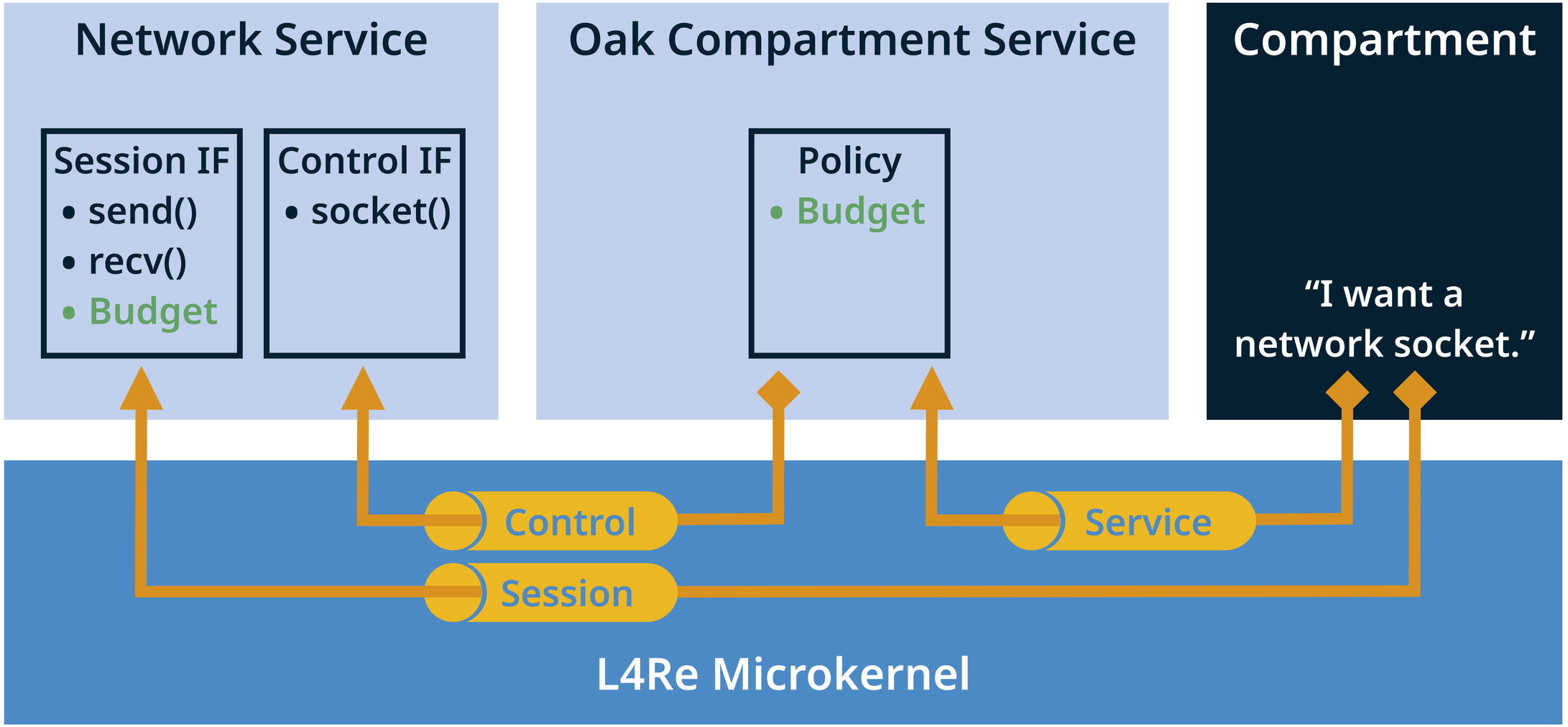
seccomp on a Microkernel

This page intentionally left blank

namespaces on a Microkernel



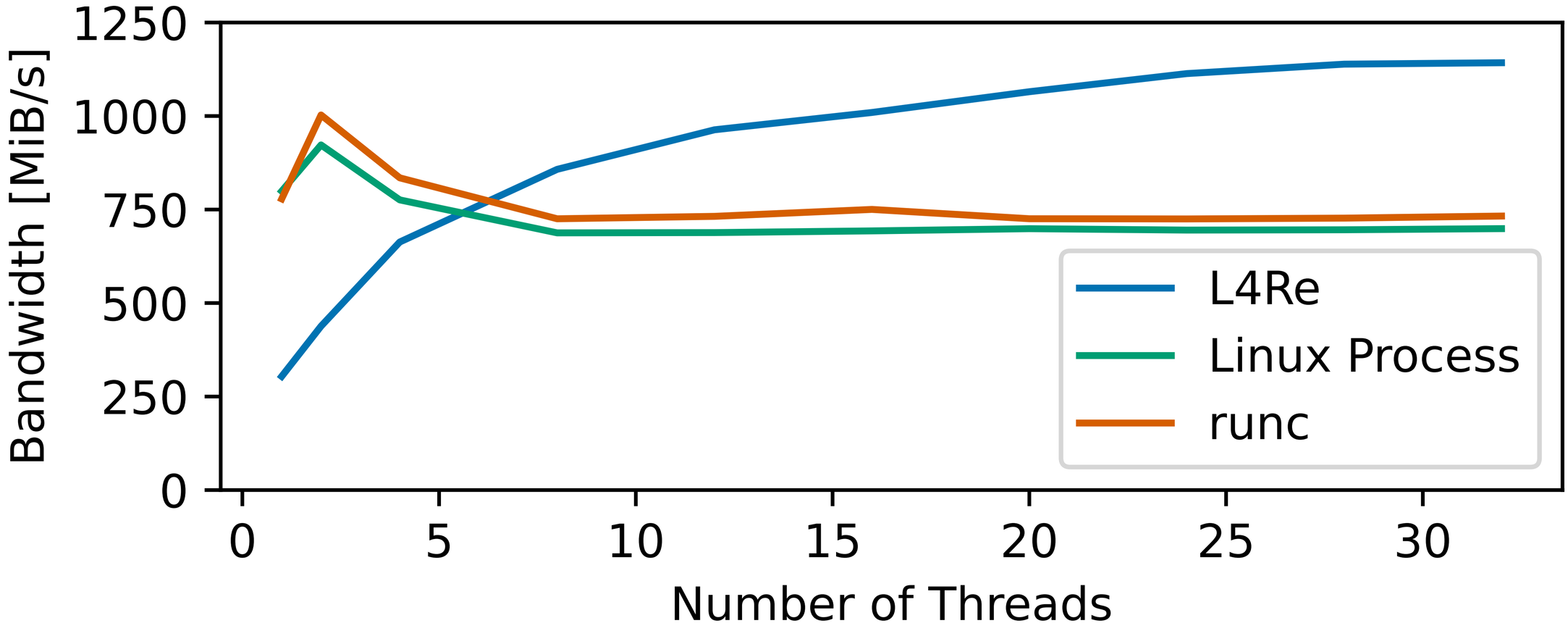
cgroups on a Microkernel



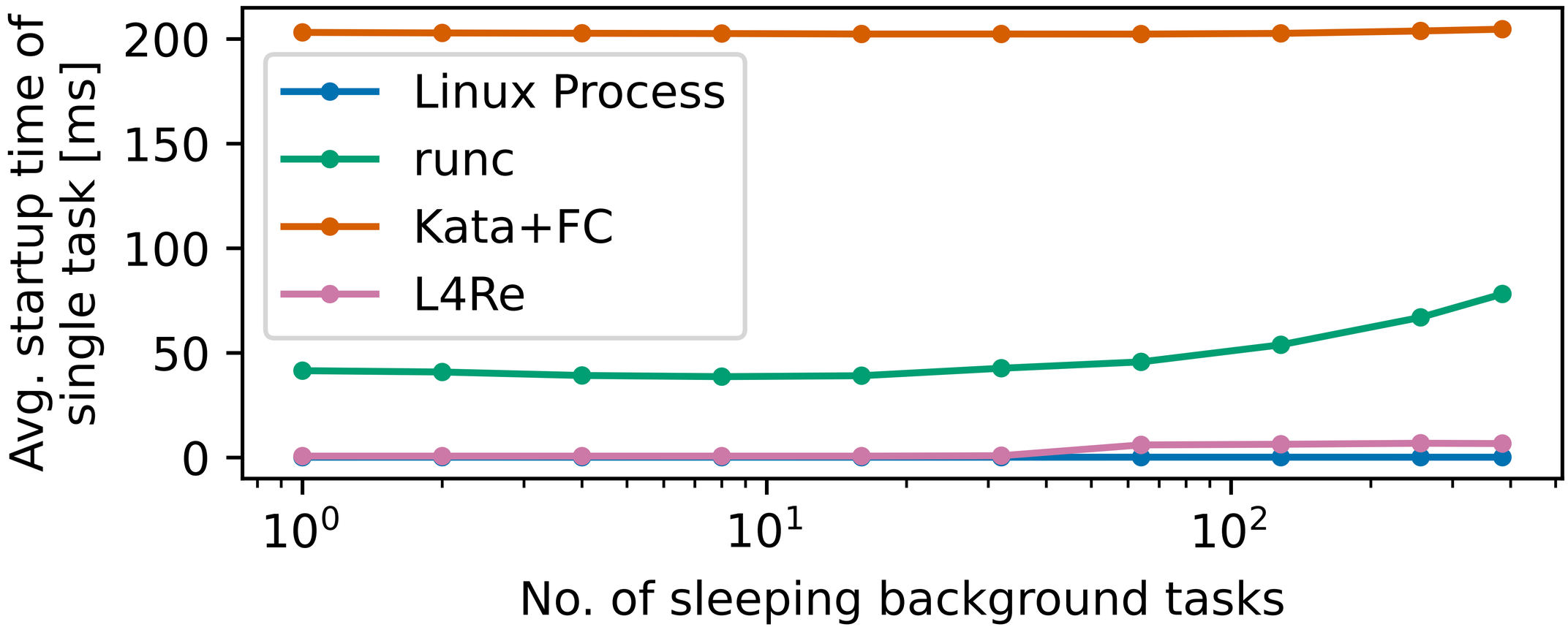
Evaluation

- implemented **Oak, network, and storage** services on L4Re microkernel
- Linux baselines
  - **processes:** fastest option on Linux, weak isolation
  - **runc containers:** isolation based on `seccomp`, `namespaces`, `cgroups`
  - **Kata containers + Firecracker:** virtualization-based isolation
- dual-socket Intel Xeon Platinum 8358 servers, 500 GiB DRAM, 10G ethernet

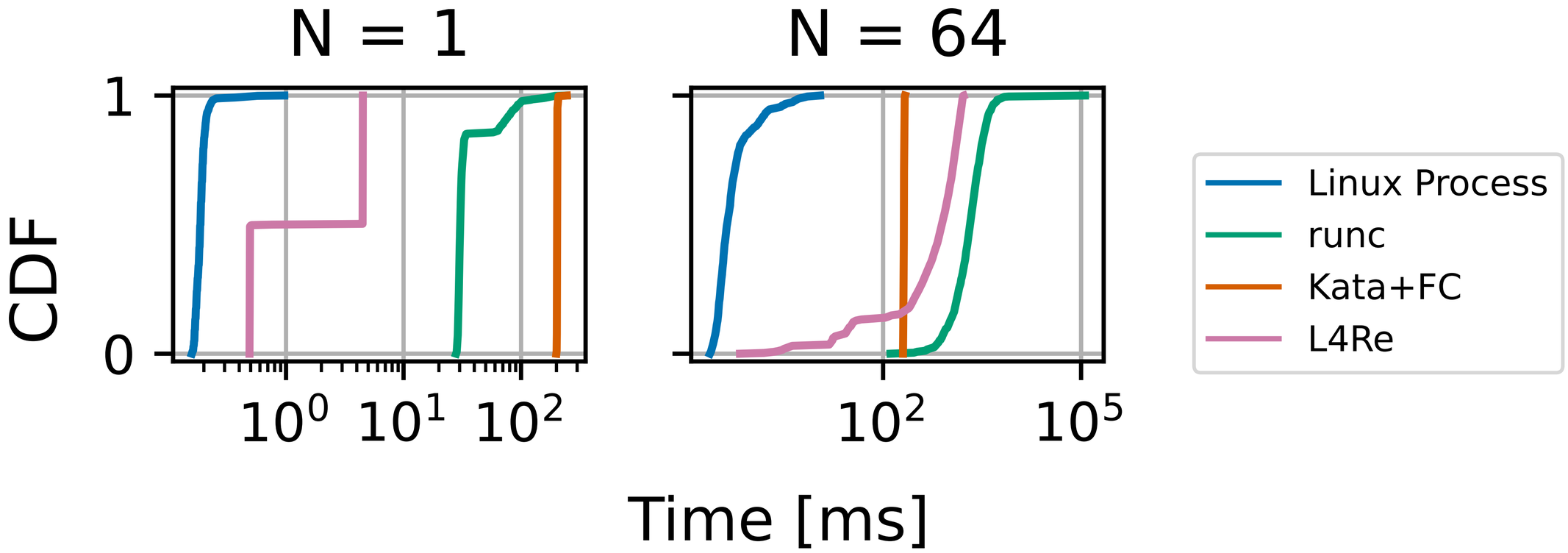
Network Performance



Container Startup Latency: Single Launch



Container Startup Latency: Parallel Launch



Summary

Oak implements secure container isolation for trustworthy clouds

- Linux processes need **additional restriction** to provide container isolation
- mechanisms have shown **security vulnerabilities**
- **microkernels fully isolate** processes by default
- **Oak:** secure containers on a microkernel-based system
- **competitive performance** for network IO and container startup