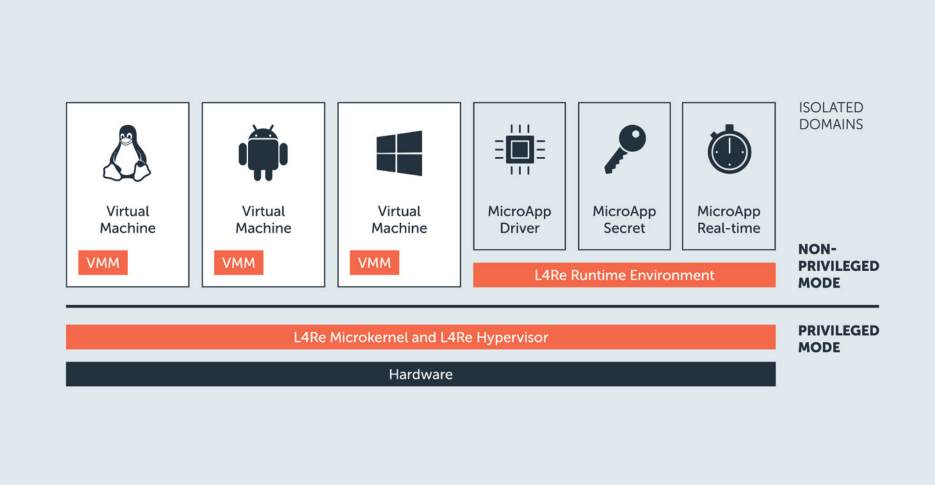
KERNKONZEPT



- **+ Open-source microkernel-based operating system and hypervisor**
- **+ Security by design – not as an afterthought**
- **+ L4Re Microkernel with less than 30,000 lines of code – Certification ready**
- **+ L4Re Runtime Environment for developing trusted, minimal applications**
- **+ Flexible and scalable: automotive, IoT, cloud, government**
- **+ Principle of least authority through object capability-based access control**
- **+ Highly scalable multiprocessor support**
- **+ ARM, x86, and MIPS, in 64-bit and 32-bit modes**
- **+ Hardware virtualization on ARM, x86, and MIPS, capable of running in ARM TrustZone**
- **+ Multi-personality OS: native L4Re applications, unmodified Linux, Windows and legacy OS guests, paravirtualized Linux, OpenBSD, and FreeRTOS systems**
- **+ coreboot support**

L4Re OPERATING SYSTEM AND L4RE HYPERVISOR

BUILDING TRUST AND SECURITY

Today’s systems are increasingly packed with features. They are also increasingly exposed to threats such as malware, data breaches, and spying. At Kernkonzept, we believe that a proper system design will alleviate the increasing threats in the digital world.

We develop and maintain the open-source L4Re Operating System Framework. The microkernel-based Operating System Framework is built on the principle of a minimal trusted computing base: Minimize your application’s potential for failure and attacks by modularization and by reducing its dependencies. Isolate components in secure compartments and virtual machines. Reuse your legacy systems as untrusted components.

USE CASES

Consolidation: Multiple systems on one device – The L4Re Operating System Framework offers a platform for integrating multiple systems or applications on one device. Subsystems and applications do not have to trust each other and are separated into isolated compartments. The L4Re Operating System Framework provides both temporal and spatial isolation, supporting both security-critical and real-time applications.

Minimal trusted computing base
The component architecture of the L4Re Operating System Framework allows reducing your critical application’s trusted computing base (TCB) to just those components that are actually used. Native L4Re MicroApps have a TCB that is several orders of magnitude smaller than with conventional systems such as Linux or Windows. Nonetheless, legacy OSes and software stacks can be reused by moving them off the critical path into an isolated virtual machine.

Secure virtualization – The L4Re Operating System Framework provides both paravirtualization and hardware-assisted virtualization on all supported platforms. Each virtual machine (VM) uses its private virtual machine monitor (VMM), and only this VMM contributes to the VM’s trusted computing base. Several different VMM implementations for different use cases are available.

TECHNOLOGY

The L4Re Microkernel serves as the heart of L4Re Operating System Framework and functions as a hypervisor, separation kernel, and real-time microkernel. It implements only those mechanisms that need to reside in the CPU’s privileged mode: address spaces, threads, and inter-process communication.

All other operating system components, including all device drivers and access policies, are implemented in user-mode application programs or are encapsulated in virtual machines.

- Third-generation, minimal, security-centric, real-time capable microkernel
- Supports both static and dynamic system designs
- Supports open, yet confined, systems with user-installed apps
- Access control and uniform resource access with kernel-protected object capabilities
- Device drivers and VMMs are untrusted user-level components
- Open source code builds trust, allows custom development, and eases evaluations

Dedicated VMM for each virtual machine – The L4Re Operating System Framework‘s virtual machine monitor (VMM) enables hardware-accelerated VMs on all supported CPU architectures. The VMM provides a minimal execution environment for unmodified guest OSes, and it is not part of the TCB of trusted components.

- Runs unmodified guest operating systems
- Enables native Linux device drivers with device pass-through
- Virtio-compatible high-performance virtual interfaces for secure interaction with other L4Re components and other VMs

L4Re Runtime Environment – The L4Re Runtime Environment is used to develop native trusted L4Re MicroApps with extremely small trusted computing bases.

- C++11 or higher / C programming environment
- Native L4Re MicroApps development with libstdc++, pthreads, libc, and a subset of POSIX interfaces
- Uniform, seamless and secure naming of, and access to, microkernel and runtime resources through object capabilities
- Rich set of high-level abstractions for flexible system design
- Secure GUI
- Secure flexible platform management
- Flexible system startup
- Support for third-party libraries and systems available on request
- Staged boot

Development environment for native MicroApps

- Seamless development using native Linux compilers and tools (GCC >=6.0)
- Fast and flexible development cycles with, Qemu, KVM, or VirtualBox
- Extensive kernel debugger for in-depth view into microkernel and system status
- Supported by Lauterbach and iSystems debuggers

Get the L4Re Operating System Framework

The L4Re Microkernel, L4Re Runtime Environment, VMM, and other resources are all licensed under the GPLv2. Download at github: *github.com/kernkonzept*

L4Re is a trademark or registered trademark of Kernkonzept in certain countries. Other names may be trademarks of their respective owners. This document serves informational purposes only. Subject to change without notice. Not warranted to be error-free, nor subject to any warranties. Kernkonzept disclaims any liabilities, and no contractual obligations are formed by this document.