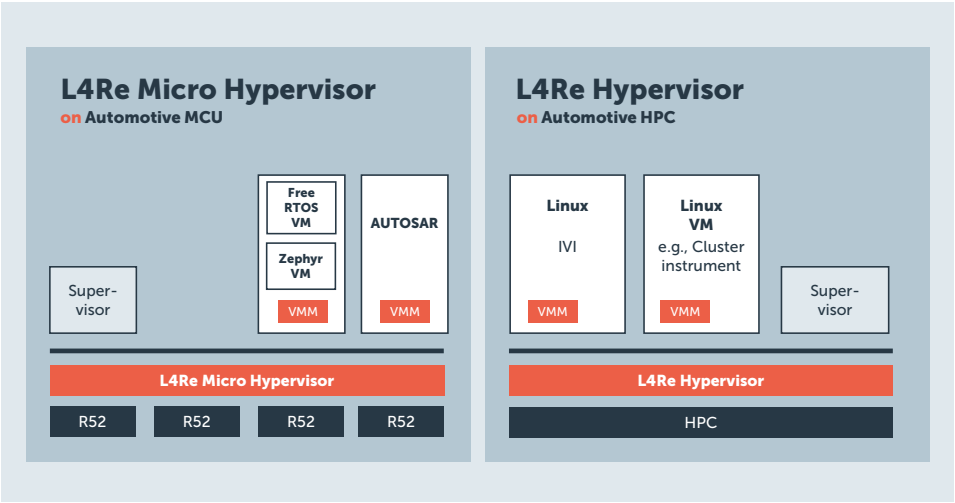
KERNKONZEPT



+ **Open-source, micro-kernel-based operating system and hypervisor family**

+ **Security by design – not as an afterthought**

+ **L4Re Microkernel and Hypervisor with only 30,000 lines of code – certification-ready and certified**

+ **L4Re Core Foundation for developing trusted, minimal applications**

+ **Flexible: applicable for automotive, IoT, cloud, government**

+ **Scales from embedded to cloud**

+ **Principle of least authority through object-capability-based access control**

+ **HIghly scalable multi-processor support**

+ **Arm v7/8-A, Arm v8-R, x86, MIPS, RISC-V, running in 64/32-bit mode**

+ **Hardware virtualization on Arm, x86, MIPS, RISC-V, running in Arm TrustZone**

+ **Multi-personality OS: native L4Re applications, unmodified guest OSs from FreeRTOS, Zephyr to Linux and OpenBSD, classic AUTOSAR**

+ **Coreboot support**

L4Re OPERATING SYSTEM
L4Re HYPERVISOR FAMILY

BUILDING SECURITY, TRUST, AND SAFETY

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 in a proper system design to alleviate the increasing threats in the digital world.

We develop and maintain the open source L4Re Operating System and Hypervisor 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 (VMs). Reuse your legacy systems as untrusted components.

USE CASES

ECU Consolidation: Multiple ECUs on one device

The L4Re System offers a platform for integrating multiple logical ECUs or other software-defined functions into a more capable zonal controller. The L4Re Hypervisor allows the reuse of existing ECU software stacks almost unchanged in VMs and provides freedom of interference between safety-related functions. It ensures temporal and spatial separation for safety and realtime-critical applications.

Minimal trusted computing base

The component architecture of the L4Re Operating System and Hypervisor Framework allows reducing your critical application's trusted computing base (TCB) to just those components that are actually used. Native L4Re Micro-Apps have a TCB that is infinitely smaller than that of conventional systems such as Linux or Windows. Nonetheless, legacy OSes and software stacks can still be used by moving them off the critical path into an isolated VM.

Secure virtualization

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

TECHNOLOGY

The L4Re Microkernel and L4Re Hypervisor is the heart of the L4Re Operating System Framework. It functions as a hypervisor, separation kernel, and real-time microkernel. It features a modular design which provides fault isolation. Critical functionality is moved out of the kernel into encapsulated user-mode applications which are subject to the object-capability-based access control mechanism.

KERNKONZEPT

GEHEIM

NATO
SECRET

CC
EAL 4+

ASIL B *

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

- Third-generation, minimal, security-centric, realtime-capable microkernel
- 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 userlevel components
- Open-source code for trust, custom development, easier certification
- L4Re Hypervisor Family provides full flexibility from embedded to cloud

Dedicated VMM for each virtual machine

L4Re OS 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 OSs, it is not part of the TCB of trusted components.

- Runs unmodified guest operating systems: e.g. FreeRTOS, Zephyr, Linux, OpenBSD
- Enables native Linux device drivers with device pass-through
- Virtio-compatible high-performance virtual interfaces for secure interactions with other L4Re components and other VMs

L4Re Core Foundation

The L4Re Runtime Environment is used to develop native trusted L4Re Micro-Apps with extremely small trusted computing bases.

- C/C++11 or higher / C programming environment
- Native L4Re Micro-Apps 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 and safe platform management
- Multiple boot methods and boot loaders supported
- Support for libraries and third-party systems available on request
- Staged boot

Development SDK for native Micro-Apps

- Seamless development with native Linux compilers and tools (GCC ≥ 9, LLVM)
- Rapid development cycles with QEMU, KVM, VirtualBox; no hardware needed
- Extensive kernel debugger for in-depth view into microkernel and system status
- Supported by Lauterbach and iSystems debuggers

Get the L4Re Operating System and Hypervisor Framework

The L4Re Microkernel, L4Re Hypervisor, L4Re Micro Hypervisor for MPU-protected processors, L4Re Runtime Environment, VMM and other resources are all licensed under the open-source MIT license.

Download: Get acquainted with L4Re on *l4re.org* and *github.com/kernkonzept*

Kernkonzept GmbH
Buchenstraße 16b
01097 Dresden, Germany
+49 351 41 888 611
info@kernkonzept.com
www.kernkonzept.com

* for customer product