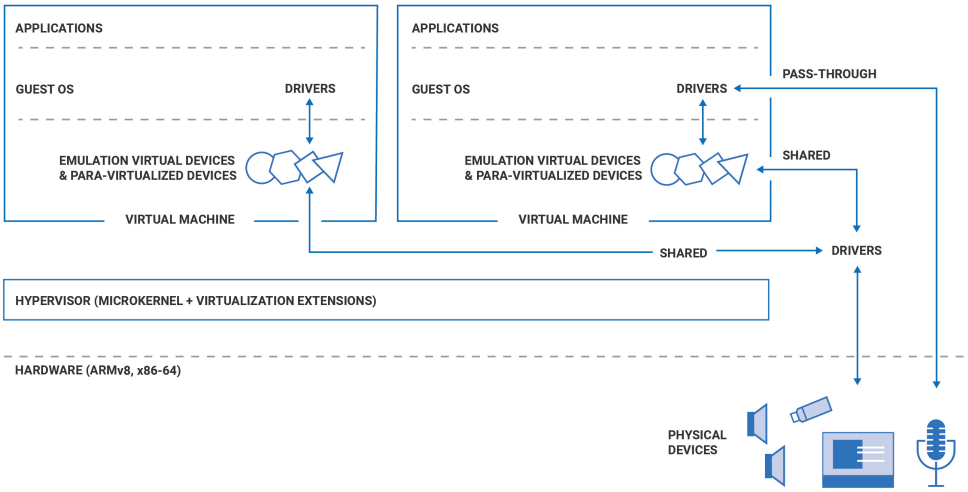
BlackBerry | QNX

Product Brief

QNX Hypervisor



Overview of a virtualized system using QNX Hypervisor with virtual, para-virtualized and pass-through devices

Consolidate Diverse Systems on a Single SoC

With the QNX® Hypervisor you can consolidate multiple systems with diverse OSs and different reliability and security requirements onto a single System on a Chip (SoC). You can, for example, build a safety-critical system certified to standards such as IEC 61508 and ISO 26262 that includes one or more safety-certified virtual machines. These virtual machines can contain non-safety guest systems and run alongside safety-certified software components executing mission-critical work.

Whatever your overall system requirements, with the QNX Hypervisor you'll be able to implement the features you need, on the OSs you prefer, all the while reducing system power consumption and the need for thermal dissipation, and—especially—reducing both initial development costs and long-term cost of ownership.

Proven Reliability

The QNX Hypervisor implements virtualization extensions to the QNX Neutrino® Real-time Operating System (RTOS) microkernel, so when you design a system with the QNX Hypervisor, you are building on a foundation whose reliability and performance has been proven over more than four decades in hundreds of millions of mission-critical systems.

Move Legacy Code to the Latest Hardware

With the QNX Hypervisor you can run legacy code on the latest hardware, simply by running it in an appropriately configured hypervisor virtual machine.

Innovate with Android & Linux

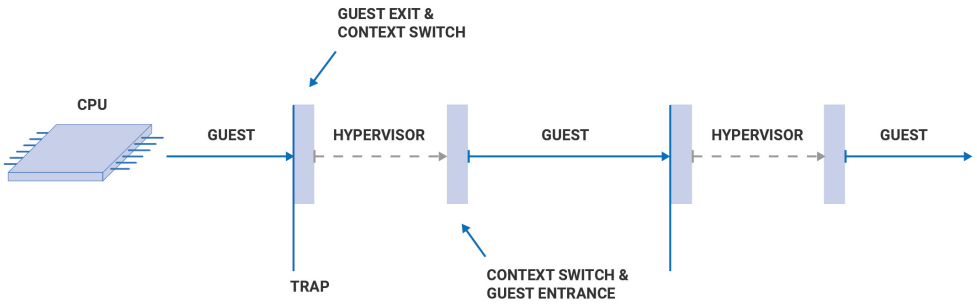
With guest OSs safely and securely contained in QNX Hypervisor virtual machines, you can innovate with Android and Linux systems free of concerns that they might jeopardize the integrity of your overall system.

Ensure System Isolation

Thanks to its microkernel architecture, the QNX Hypervisor protects itself and your system from both internal faults and outside interference, including from guests in its virtual machines.

When configuring the virtual machines, you assign the guests' memory allocations and access to devices, both physical and virtual, and the hypervisor enforces these configurations. If a guest attempts to overstep boundaries you set, either through error or malicious intent, the QNX Hypervisor prevents the guest from completing the action.

The QNX Hypervisor includes the SMMU manager service (SMMUMAN), which works with your hardware System Memory Management Units (SMMUs) to ensure that Direct Memory Access (DMA) devices are contained; they can't access memory outside the bounds you program into the SMMUs.



A Lahav Line illustrating how in a QNX Hypervisor system execution alternates between the hypervisor and its guests

Contain DMA Devices

The SMMUMAN service works with board IOMMU/SMMUs to program and enforce the memory regions DMA devices can access.

Versatile Virtualization Model

The QNX Hypervisor leverages the latest ARMv8 and x86-64 hardware virtualization extensions to provide a supremely versatile virtualization solution. With the QNX Hypervisor you can run your OSs and their applications as guest systems in virtual machines, limiting your hypervisor to handling events and exceptions. Just as easily, you can run one or more guest systems in virtual machines, while also implementing a full system, including resource managers, drivers and applications, directly on the hypervisor.

Whichever model you select as most appropriate for your project, the QNX Hypervisor ensures that the guest systems and the hypervisor itself are contained and isolated from the other systems on the SoC. In short, you can choose the virtualization model that best suits your needs for reliability, performance and security—and your budget.

Share the SoC and Devices

Not only can diverse OSs share the same SoC, they can share physical devices, reducing product hardware costs.

QNX Hypervisor Virtual Machines

A QNX Hypervisor virtual machine is configured much as a physical board is assembled, with memory and devices assigned to the guests or to the hypervisor itself, as required by your implementation. Devices can be physical devices (including pass-through devices), or virtual devices, including emulation and para-virtualized devices from our virtual device library.

Leverage QNX Neutrino RTOS Technologies

Since virtual machines are QNX Hypervisor processes, you can use our adaptive partitioning technology to guarantee the hypervisor and its guests the physical CPU time they need without wasting precious cycles. And because virtual CPUs are threads in the virtual machine processes, you can use our symmetrical and bound multiprocessing (SMP, BMP) technology to pin virtual CPUs to physical CPUs, or allow them to migrate, as best suits your requirements.

Develop Your Own Virtual Devices

The QNX Hypervisor includes a virtual device developer's API reference and a virtual device developer's guide, complete with examples of virtual device source code you can use as models for developing your own virtual devices, including para-virtualized devices designed and built to the VirtIO standards.

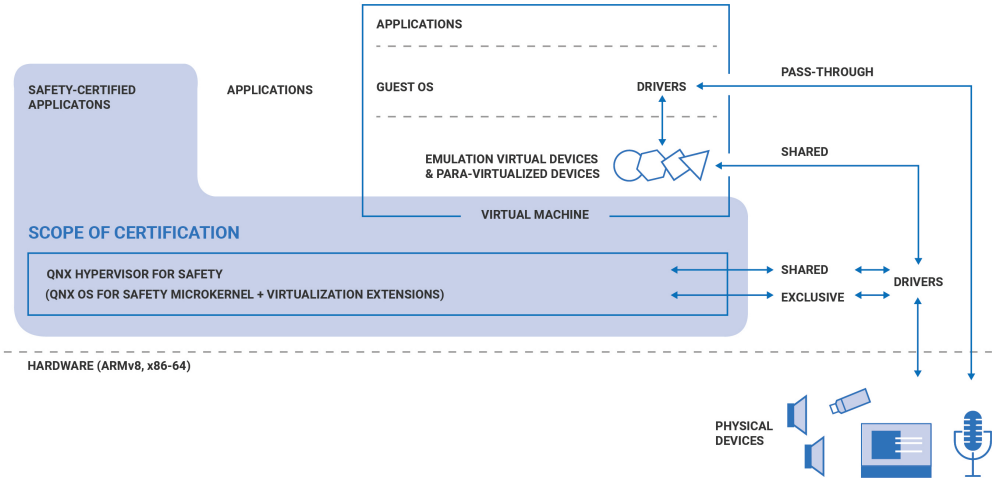
Familiar QNX Neutrino RTOS API

If you're familiar with the QNX Neutrino RTOS you'll require no ramp-up time to begin your hypervisor development work: the QNX Hypervisor is fully API-compatible with the QNX Neutrino RTOS API.

You'll be able to develop non-safety as well as safety-critical applications on the same foundations, and of course you'll be able to continue working in the QNX Software Development Platform's POSIX-compliant environment and using the QNX Momentics® Tool Suite.

QNX Hypervisor for Safety

The QNX Hypervisor for Safety is the safety-certified variant of the QNX Hypervisor. Pre-certified by TÜV Rheinland to IEC 61508 SIL 3 and ISO 26262 ASIL D, it will streamline and speed your system's safety certification. With non-safety OSs (Android, Linux) contained in QNX Hypervisor for Safety virtual machines, you can focus your certification efforts and funds on certifying only your safety-critical components.



A QNX Hypervisor for Safety configuration with a guest contained in a virtual machine, while safety-critical applications run directly on the hypervisor

Virtualization API

The QNX Hypervisor includes a virtualization API so you can develop your own, custom virtual devices.

This API is POSIX-compliant. If you're familiar with Linux you'll have no trouble working with the QNX Hypervisor.

BlackBerry QNX Professional Services

If you're uncertain about how to see your virtualized system through to market, our virtualization, security and safety experts can assist you at every stage—from design to certification.

The QNX Hypervisor at a Glance

Guest OS Support

Run unmodified guests:

- Linux
- Android
- QNX Neutrino RTOS
- QNX OS for Safety

Processor Support

- 64-bit support for the latest ARMv8 and x86-64 SoCs

Security

- Access control lists (ACLs)
- Mandatory Access Control (process privileges)
- Isolation and separation of guests
- No root mode vulnerable to malicious exploitation

Configuration

- Supports thin (event handler) implementation or full-featured OS implementation
- Guest access to Advanced Configuration and Power Interface (ACPI) tables and Flattened Device Trees (FDTs)
- Easily-modified, cascading configuration files

Networking

- Guest-to-guest
- Guest-to-hypervisor
- Guest-to-world

Memory Sharing

- Guest-to-guest
- Guest-to-hypervisor

Devices

- Assignable to a guest or to the hypervisor
- Configurable device sharing
- Pass-through physical devices
- Extensive library of virtual devices for ARMv8 and x86-64 platforms
- Para-virtualized devices build to VirtIO specifications
- Support for custom virtual devices

Development

- Fully-documented virtualization API
- Virtual device examples with source code
- VirtIO para-virtualization support

Documentation

- User's Guide (includes virtualization basics, virtual machine and device configuration, debugging and performance tuning)
- Virtual Device Developer's Guide
- Virtual Device Developer's API Reference

QNX Hypervisor for Safety

- QNX OS for Safety plus QNX Hypervisor virtualization extensions and virtual machines
- Certified to IEC 61508 SIL 3 and ISO 26262 ASIL D
- Certified libc, libm (mathlibs), and SMMU, hypervisor and virtual device interfaces
- Certified SMMU manager (SMMUMAN)
- Safety Manual
- Hazard and Risk Analysis
- Safety Case
- Release Notes

Related Products

QNX Neutrino RTOS

Not building a system that needs certification? The QNX Neutrino Real-time Operating System (RTOS) powers hundreds of millions of embedded systems in every industry where reliability matters, including automotive, medical devices, robotics, transportation and industrial automation.

QNX OS for Safety

Don't need a hypervisor system, but need a safety-certified system? The QNX OS for Safety is certified by TÜV Rheinland to IEC 61508 SIL 3, ISO 26262 ASIL D, and IEC 62304 Class C, so you can focus your talents and efforts on developing the systems your customers need.

QNX Momentics Tool Suite

Work with a mix of languages (e.g., C, C++ and Python), and develop for multiple SoC architectures (ARM and x86) simultaneously in a familiar Eclipse-based environment.

QNX Black Channel Communications Technology

Concerned about data integrity? Certified by TÜV Rheinland to ISO 26262 ASIL D, QNX Black Channel Communications Technology helps ensure the safety of your system's data communication.

BlackBerry QNX Professional Services

We've helped thousands of clients build safe, secure and reliable systems on the QNX OSs. BlackBerry® QNX system architects and engineers are here to guide you through the complex process of aligning software, hardware and processes to achieve your project goals.

Safety Services

We offer functional safety training, consulting, custom development, root cause analysis and troubleshooting, and system-level optimization and onsite services across a range of industries and systems. Let us help you with your certification journey.

Virtualization Assessment

If you built your prototype on Linux or another OS but don't know how to proceed with virtualization, we will help you better understand the effort and resources required to port your prototype or project to a QNX Hypervisor system.

About BlackBerry QNX

BlackBerry QNX is a leading supplier of safe, secure, and trusted operating systems, middleware, development tools, and engineering services for mission-critical embedded systems. BlackBerry QNX helps customers develop and deliver complex and connected next generation systems on time. Their technology is trusted in more than 215 million vehicles and hundreds of millions of embedded systems in medical, industrial automation, energy, and defense and aerospace markets. Founded in 1980, BlackBerry QNX is headquartered in Ottawa, Canada and was acquired by BlackBerry in 2010.

For more information, visit blackberry.qnx.com and follow @QNX_News.

MKT20-BB20 | 201023