



Open Kernel Labs™

Be open. Be safe.

No Compromises: The Truth about Security, Reliability, and Virtualization for Linux Devices

Rob McCammon

Vice President of Product Management

March 25, 2008

Who We Are



Leading provider of high-performance embedded virtualization and secure componentization technology

- A provider of software products and related services
- Leading innovator in microkernel technology and its applications
 - 10+ year track record and experience across 7 HW architectures
- A commercialization vehicle for NICTA, Australia's Centre of Excellence in ICT research
- Close affiliation with University of New South Wales (UNSW)
- The provider of a software solution that is deployed in 10s of millions of end-user devices
- A company founded in 2006 with headquarters in the US and engineering in Sydney, Australia

Where We Are



**Open
Kernel
Labs™**

*Be open.
Be safe.*

HTC S730



Where will we be this afternoon at 3:00?



→ Texas Land and Cattle
Steakhouse

→ 3945 Central Expressway

Open Kernel Labs and NICTA



- NICTA: National Centre of Excellence for ICT research
 - Created and funded by Australian federal government (\$500MM)
 - Shareholders include leading universities, state governments
- ERTOS: Embedded, Real-Time & Operating Systems
 - One of the largest research programs in NICTA
- Open Kernel Labs (OK): commercial spinout from ERTOS
 - Independent, venture-backed US company
- Close collaboration between OK and NICTA
 - OK: professional services and product development
 - NICTA: research; results licensed to OK; shareholder
 - Gernot Heiser is CTO of OK and ERTOS Program Director
 - Opportunity to transfer people together with technology

Embedded Systems Today



*Be open.
Be safe.*

→ Trends

- Increasing complexity (HW, SW, more functions)
- Increasing connectivity and openness
- New applications, services, management needs

→ Device Requirements

- High performance
- Constrained resources
- Reliability, Safety, Security

→ Development Challenges

- Time to market
- Development and BOM cost
- Quality assurance

→ How can you stay ahead?



Cost of Complexity

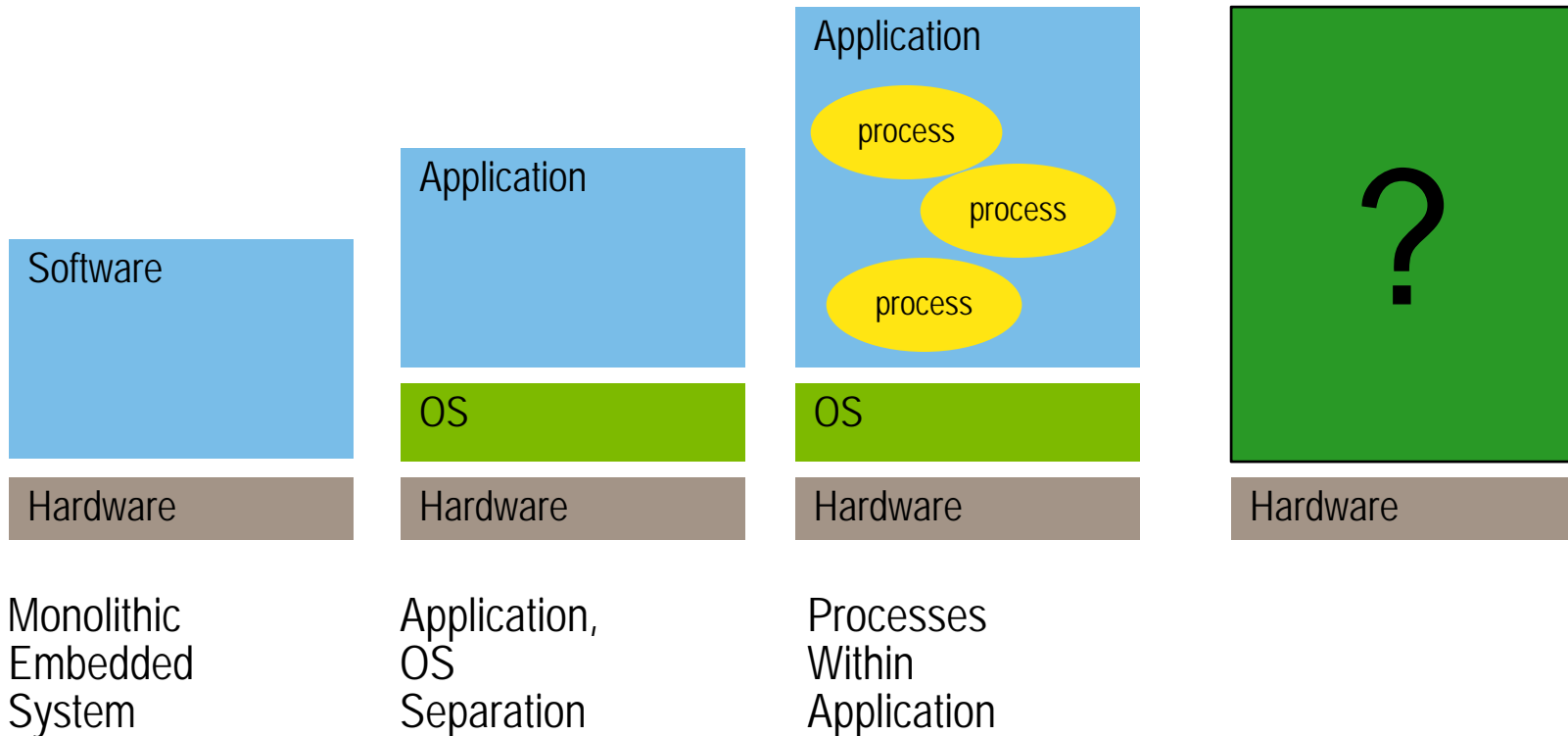


- Increased likelihood of SW defects
- Increased security risks
- Larger development teams
- More integration effort, less value added development
- Increased development time and risk

- Are we near the end of the era of increasing complexity ?

- How are you coping with this complexity now?
- How much longer will your current methods work?

Embedded System Software Has Evolved



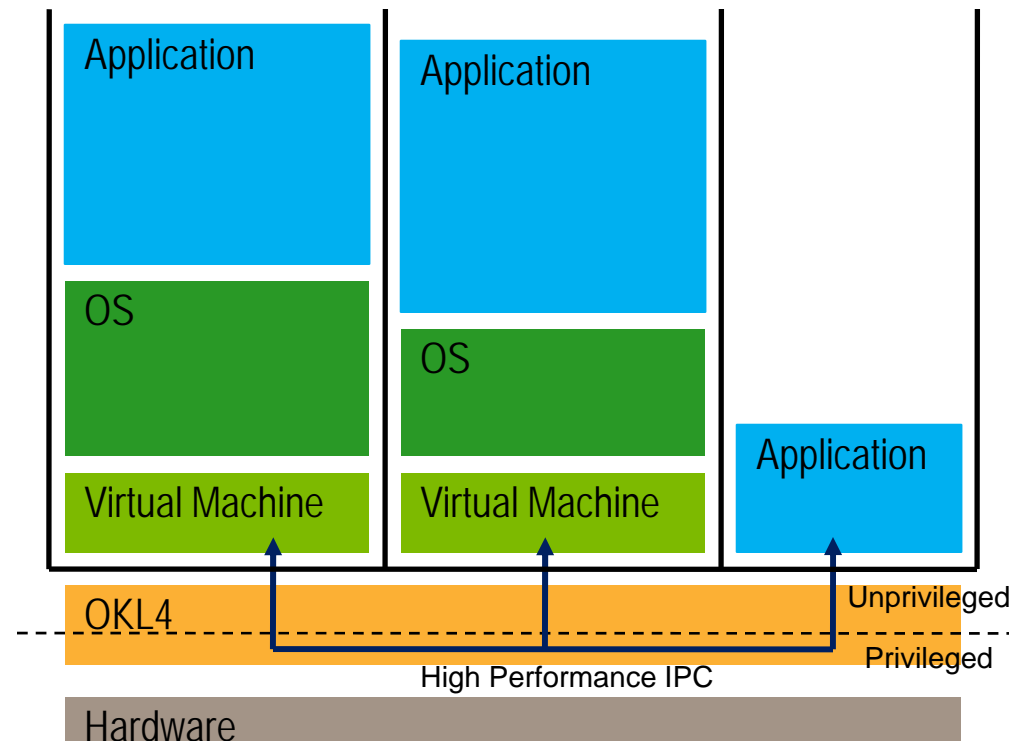
OKL4 Solution Overview



*Be open.
Be safe.*

Addresses the challenges of complexity through flexible partitioning of software into isolated subsystems

- Trustworthy Virtualization
- Secure Componentization
- Microkernel Based
- Freely available source code



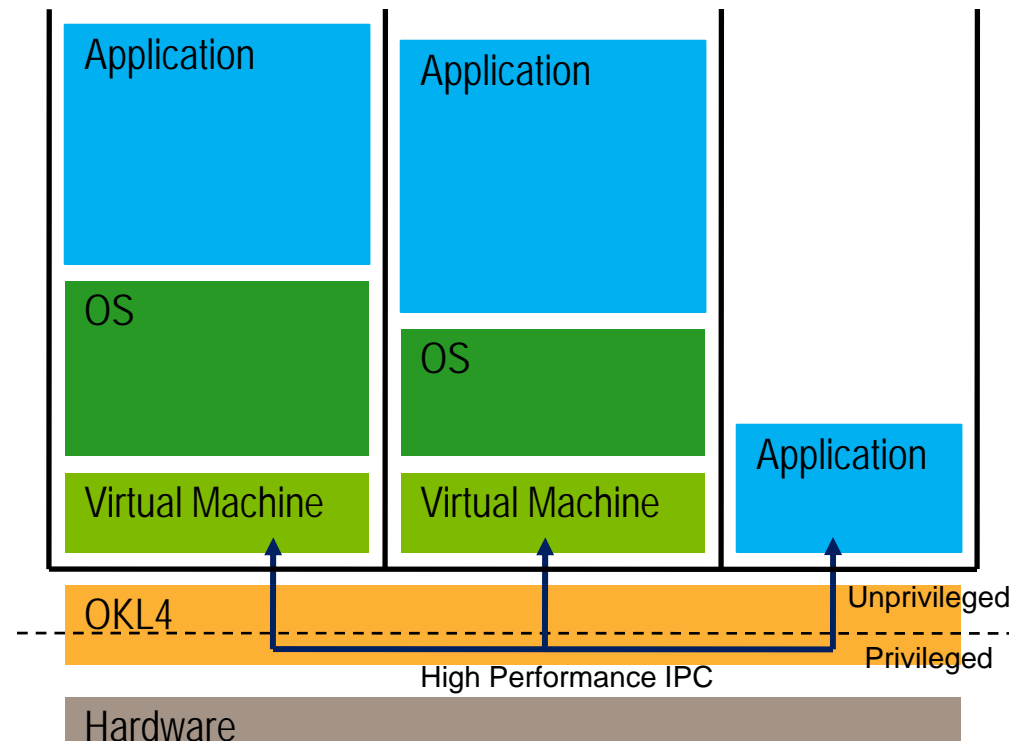
Trustworthy Virtualization



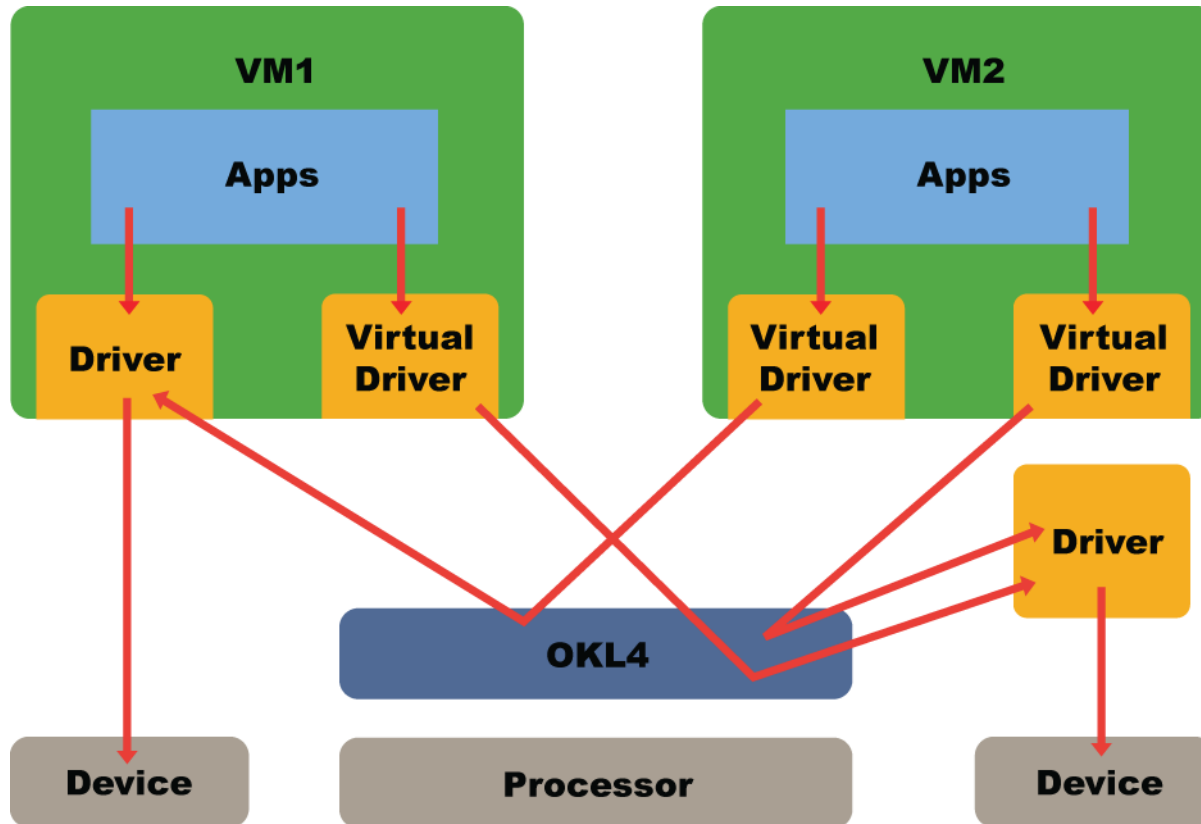
*Be open.
Be safe.*

Attributes of OKL4 make trustworthy virtualization possible

- All guests de-privileged
- All guests isolated
- Microkernel Based
 - Minimal Privileged Code
- Isolated, sharable, devices
- Facilities for VM interaction
 - High performance IPC
 - Shared memory
 - Device sharing



Device Sharing with OKL4

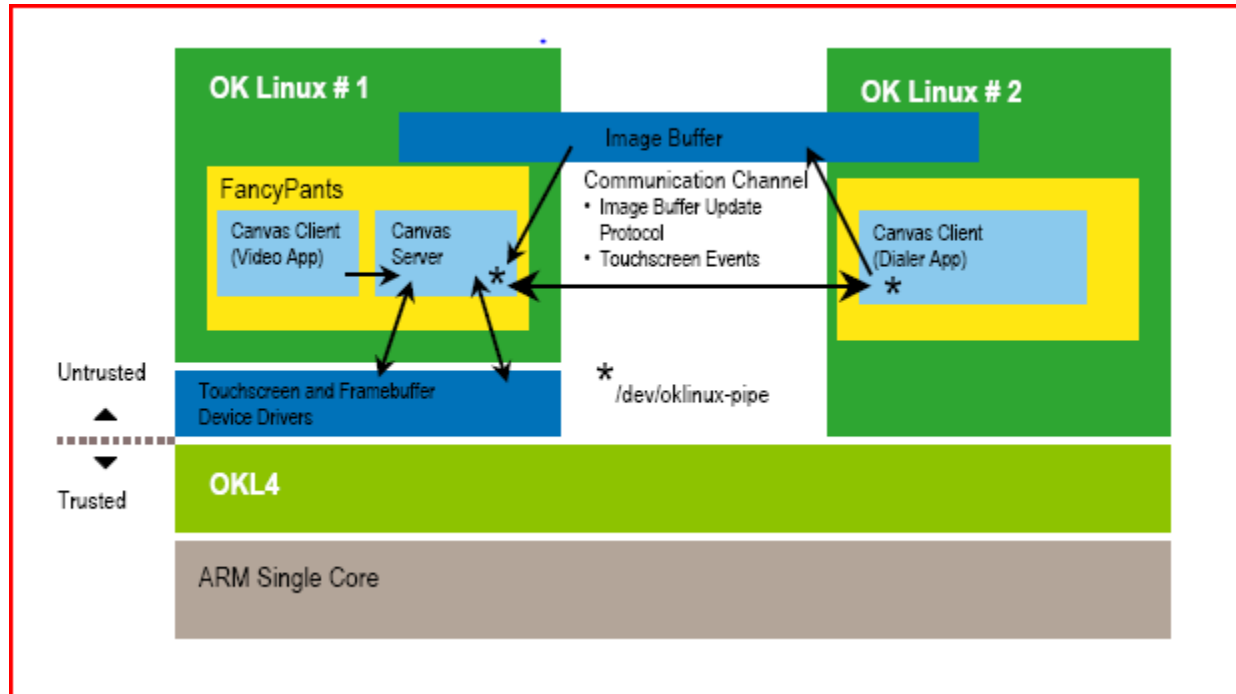


OpenMoko Demo Example

Multiple Linux VMs with OKL4



*Be open.
Be safe.*



OKL4 Performance Linux Macrobenchmarks



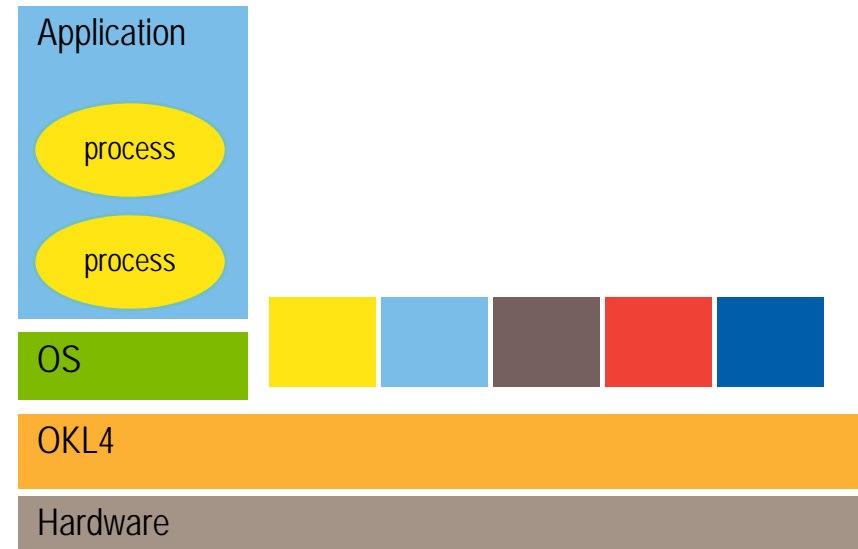
ReAIM Benchmark	Native	Virtualised	Ratio
1 Task	45.2	43.6	0.96
2 Tasks	23.6	22.6	0.96
3 Tasks	15.8	15.3	0.97

Native vs virtualised Linux on PXA255 @ 400MHz

Secure Componentization with OKL4 Offers Increased Reliability and Security



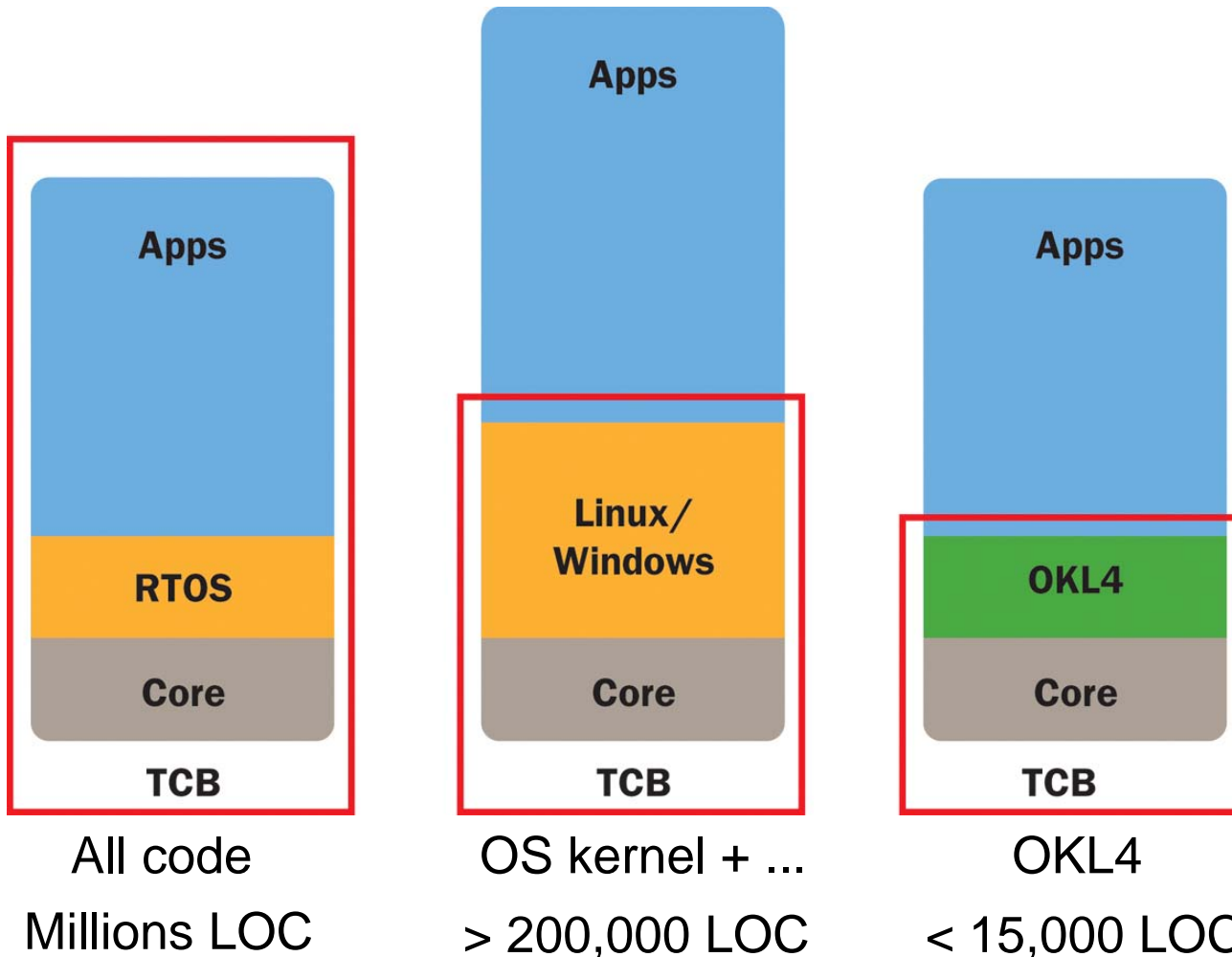
- Enables incremental migration towards componentization
- Simpler , isolated components easier to develop
- Improved fault isolation
- Reduced trusted computing base for more applications
- More granular authorization for security purposes added in a future release



Trusted Computing Base Comparison



*Be open.
Be safe.*

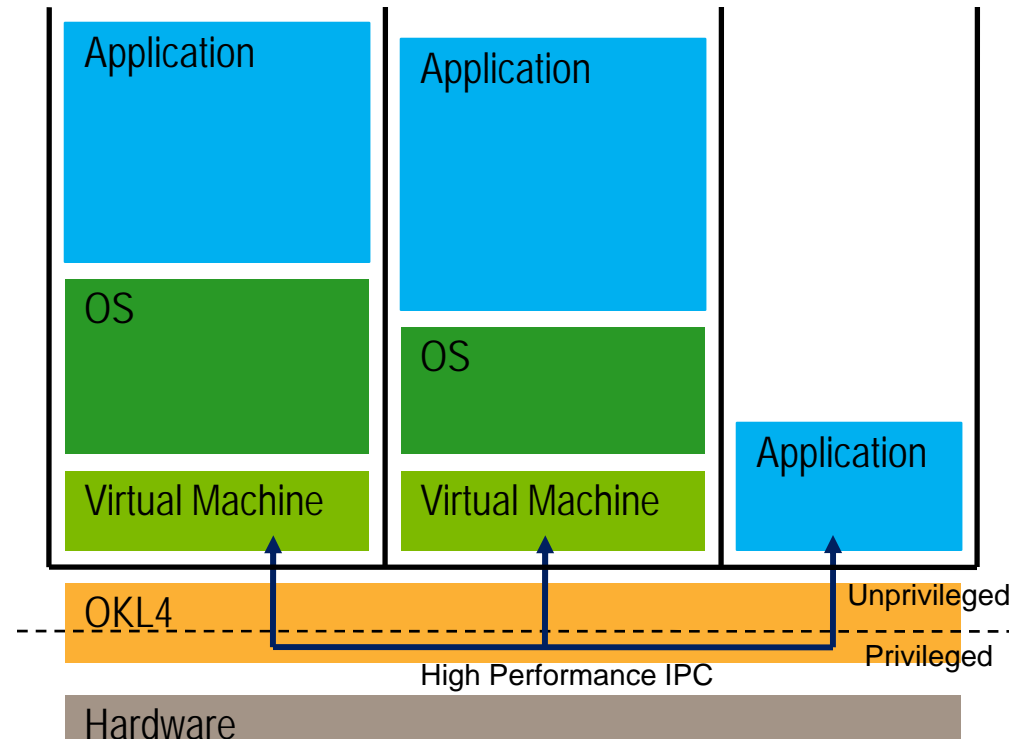


OKL4 Use Case Examples



Addresses the challenges of complexity through flexible partitioning of software into isolated subsystems

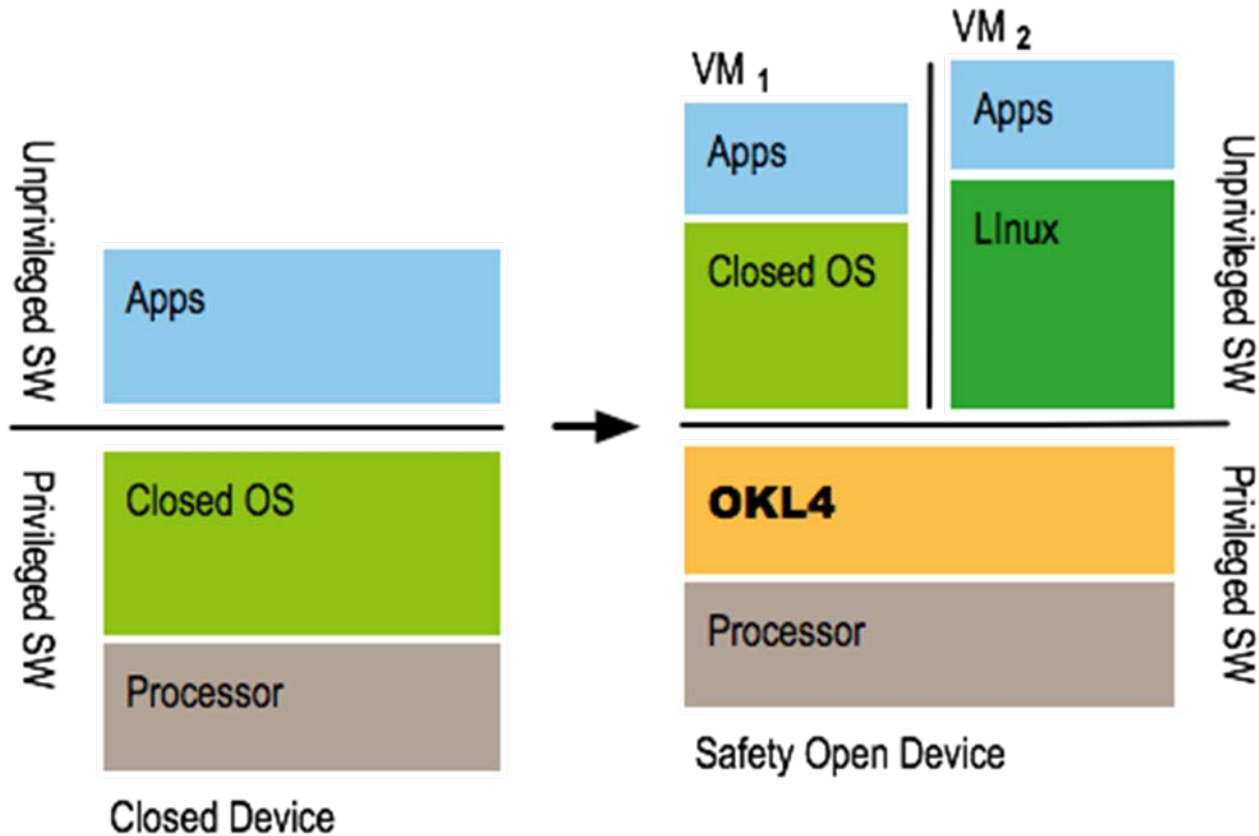
- Add Linux to a device
- Improve security
- Componentization
- Complimentary OS design
- Enabling reuse
- Processor Consolidation



Adding Linux To A Device



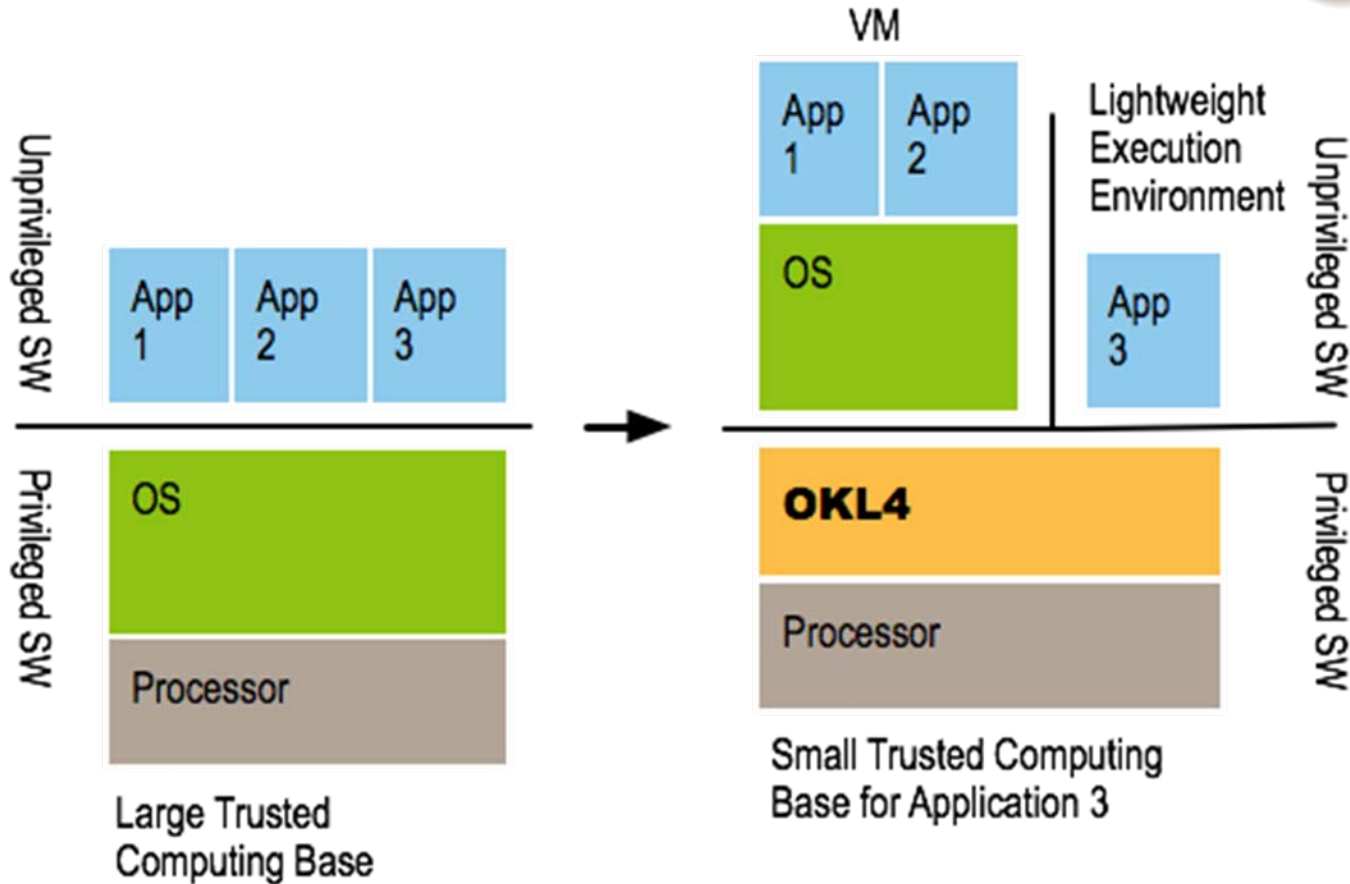
*Be open.
Be safe.*



Improving Security



Be open.
Be safe.

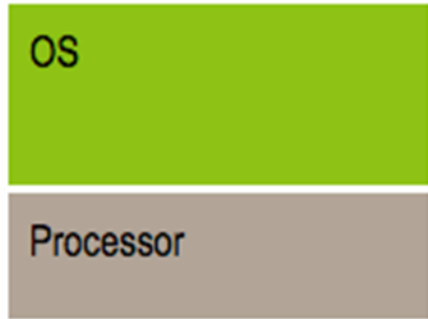


Componentization



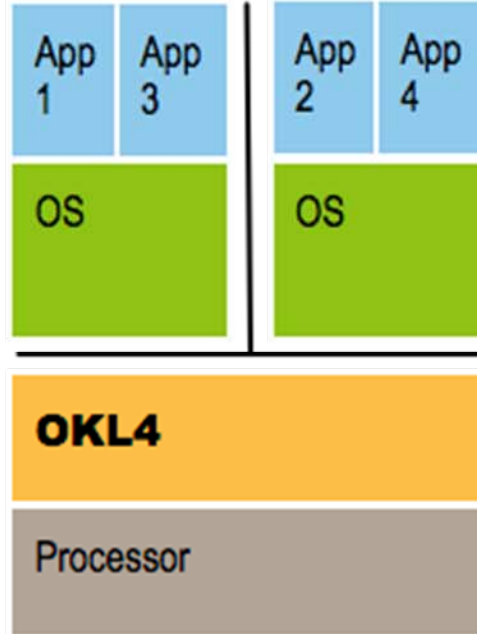
Be open.
Be safe.

Unprivileged SW



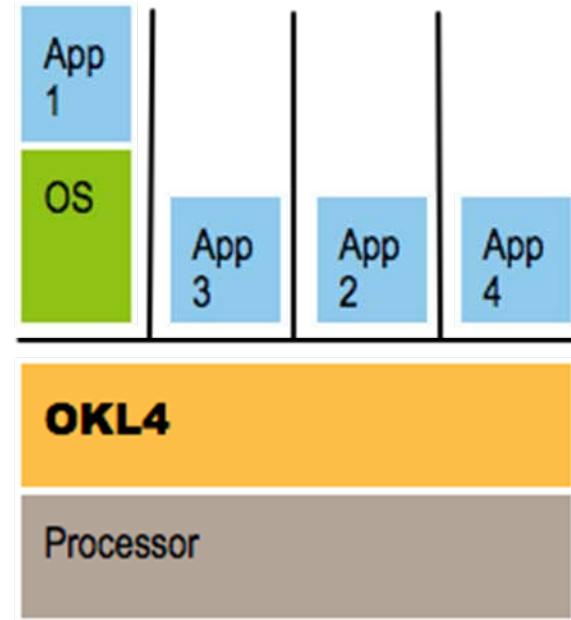
Monolithic Architecture

VM 1 VM 2



Coarse Grained Partitioning with Virtualization

VM



Finer Grained Partitioning using OKL4 Execution Environments

Lightweight Execution Environments

Unprivileged SW

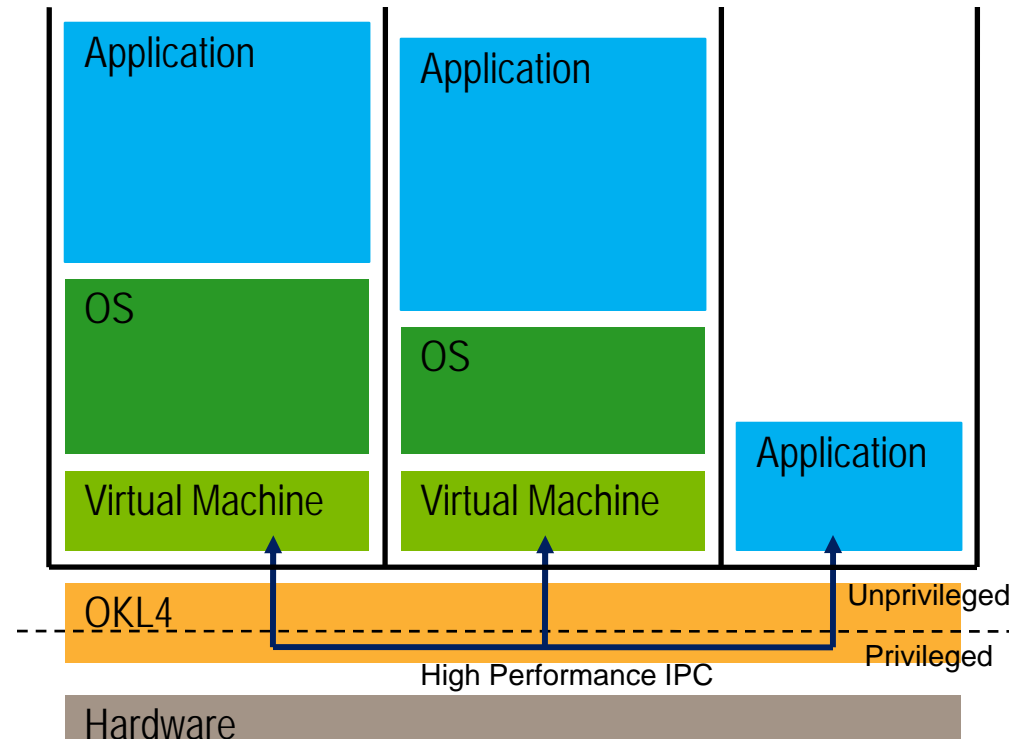
Privileged SW

Future, Research-Enabled, Enhancements to OKL4



Unparalleled trustworthiness for OKL4

- Authorization of component communication
- Comprehensive worst case timing analysis
- OKL4 formal verification
 - Design of security properties
 - Defect free C code implementation
 - Mathematically proven



Meet the Challenges of Today's Embedded Systems with OKL4



- Use OKL4 to satisfy your virtualization requirements
- Partition complex software into less complex subsystems
 - Virtual machines for course grained partitioning
 - Lightweight OKL4 native components for finer grained partitioning
- Improve security by going beyond address space isolation to a smaller TCB for critical components and the use of controlled communication
- Benefit from OKL4 evolution into a system software platform with unparalleled trustworthiness
 - Deep knowledge of worst case execution timing
 - Mathematically proven correctness

Questions?



- Join us for more detailed discussion
- 3:00
 - Texas Land and Cattle Steakhouse
 - 3945 Central Expressway



*Be open.
Be safe.*

Questions?