



Open Kernel Labs™

Be open. Be safe.

VIRTUALIZATION AND COMPONENTIZATION IN EMBEDDED SYSTEMS

And how it will change the way you engineer

Josh Matthews

Field Application Engineer

October 27, 2008

Agenda



*Be open.
Be safe.*

- **The changes in the market**
 - **And how they affect you**
- Virtualization and Componentization
 - And how it can help!
- The OKL4 Approach
 - Engineering change with Secure HyperCell™ Technology
- Open Kernel Labs

It's Getting Hot in Here



*Be open.
Be safe.*

- Mobile and CE marketplace is changing
 - Massive growth = significant shift in consumer expectations
- User-customizable / open source app environments
 - The mobile as the new PC = relinquishing control over the device
 - How to guarantee a consistent user experience?
- Unprecedented security requirements
 - Financial transactions, sensitive data, enterprise networks
 - How to guarantee security? How to minimize the TCB?
- Unprecedented reliability requirements
 - The device is the cornerstone of the digital life
 - How to guarantee an always-on experience?



It's Getting Hot in Here



*Be open.
Be safe.*

- Responsiveness from go to “whoa”
 - Doing more, doing it faster
 - And doing it with less!
- The great platform wars
 - OpenMoko, LiMo, Android
 - Unclear which platform will be the consumer choice
- Legacy migration
 - Leveraging the massive investment in base-band stacks, legacy RTOS's
- Time to market, hardware consolidation, BOM cost
 - Get it out there
 - Make it smaller
 - Sell it for less



Agenda



*Be open.
Be safe.*

- The changes in the market
 - And how they affect you
- **Virtualization and Componentization**
 - **And how it can help!**
- The OKL4 Approach
 - Engineering change with Secure HyperCell™ Technology
- Open Kernel Labs

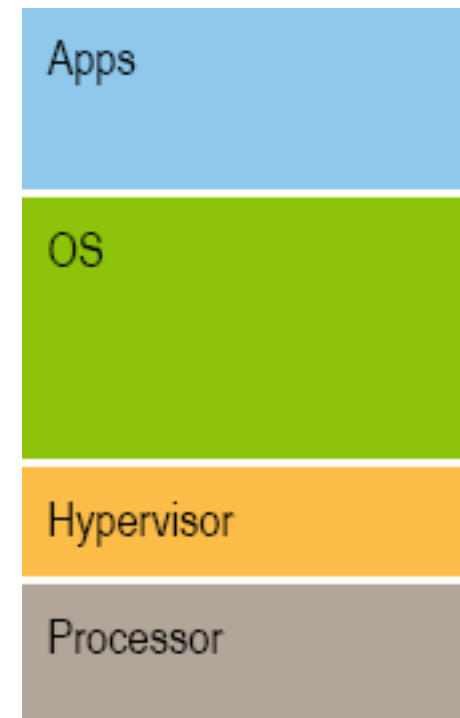
Engineering Change: Virtualization



*Be open.
Be safe.*

- A software environment on which programs, including operating systems, can run as if on bare hardware
 - An efficient, isolated duplicate of the real machine
 - VMM/Hypervisor: the software layer that provides the VM environment

- 3 essential characteristics:
 - An environment that is essentially identical with the original machine
 - Programs run with, at worst, minor decreases in speed
 - The VMM is in complete control of system resources



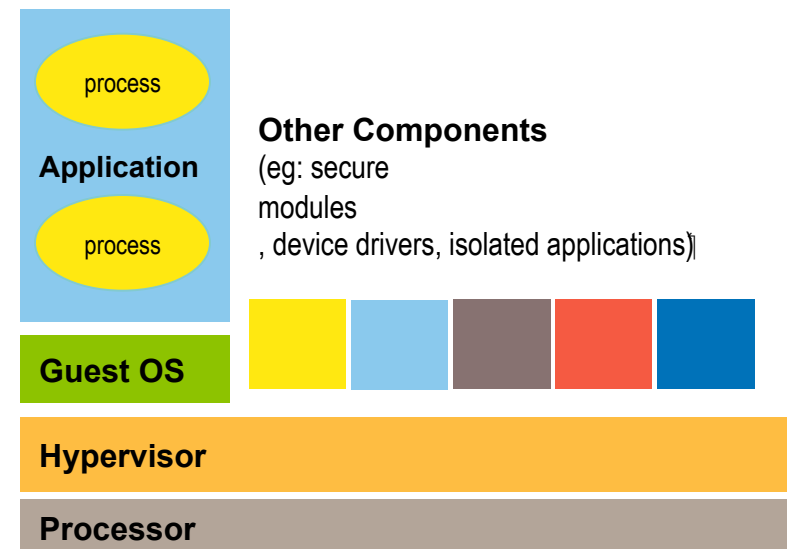
Engineering Change: Componentization



*Be open.
Be safe.*

- Segment and distribute large and complex software into a number of simpler, isolated components
 - With smaller trusted computing bases!
 - Isolated from one another
 - Secured

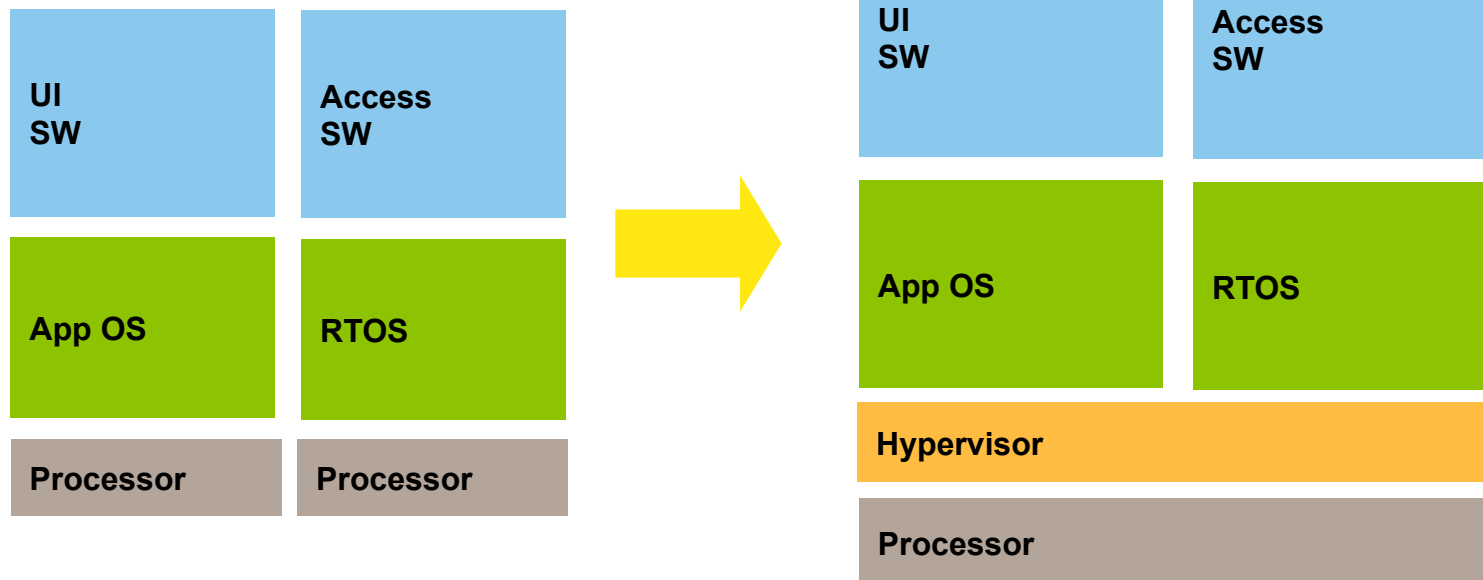
- 3 characteristics:
 - Components are isolated from one another in protected execution environments.
 - Components can be constructed with a choice of granularity
 - Decomposition can be done incrementally



Benefit: Processor Consolidation and Multiple Concurrent Operating Systems



- Various subsystems with competing aims
- RTOS for base-band
 - High Level OS for applications
 - Run both!
 - On the one processor!

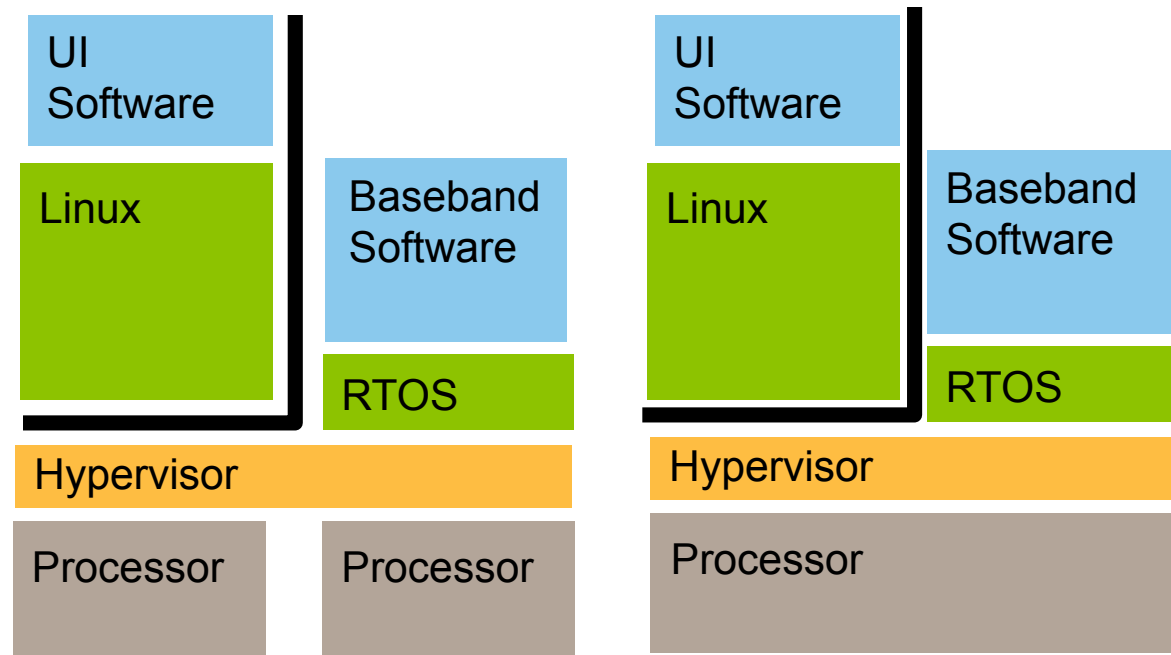


Benefit: Software Architecture Abstraction



Be open.
Be safe.

- Support for *product series*
 - Range of related products with varying capabilities
- Same low-level software for high- and medium-end devices
 - Single underlying API to program against
 - Reduced development time
 - Freedom in design

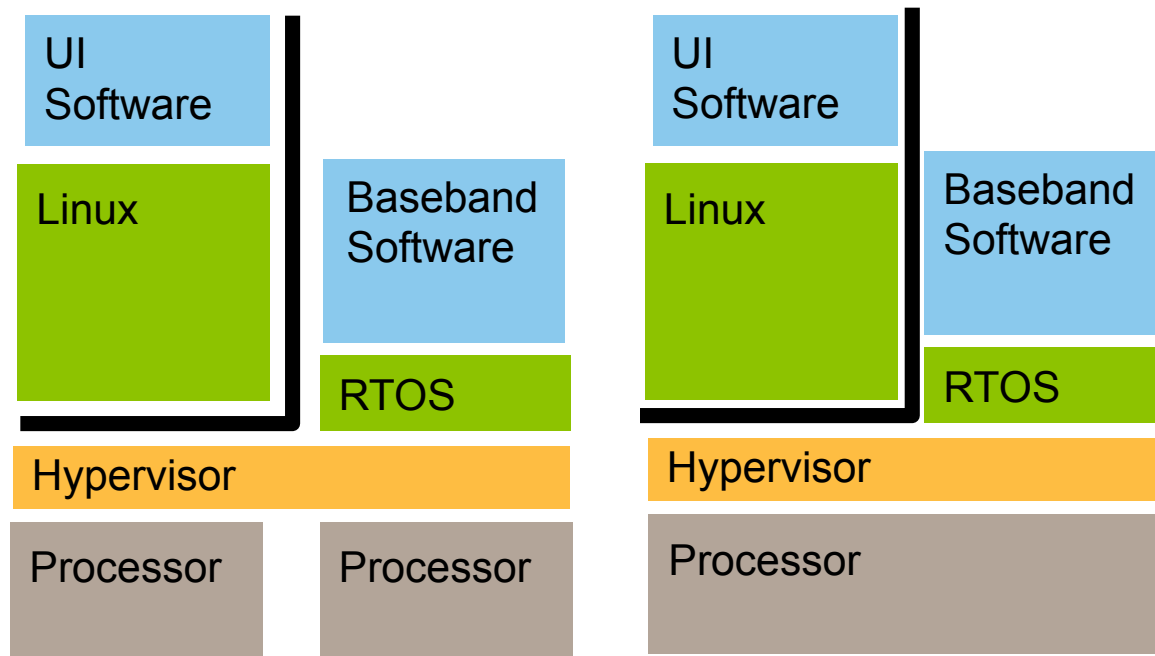


Benefit: Dynamic Processor Allocation



*Be open.
Be safe.*

- Allocate share of base-band processor to application OS
 - Provide extra CPU power during high-load periods (media play)
 - Better processor utilization
 - Higher performance with lower-end hardware
 - HW cost reduction

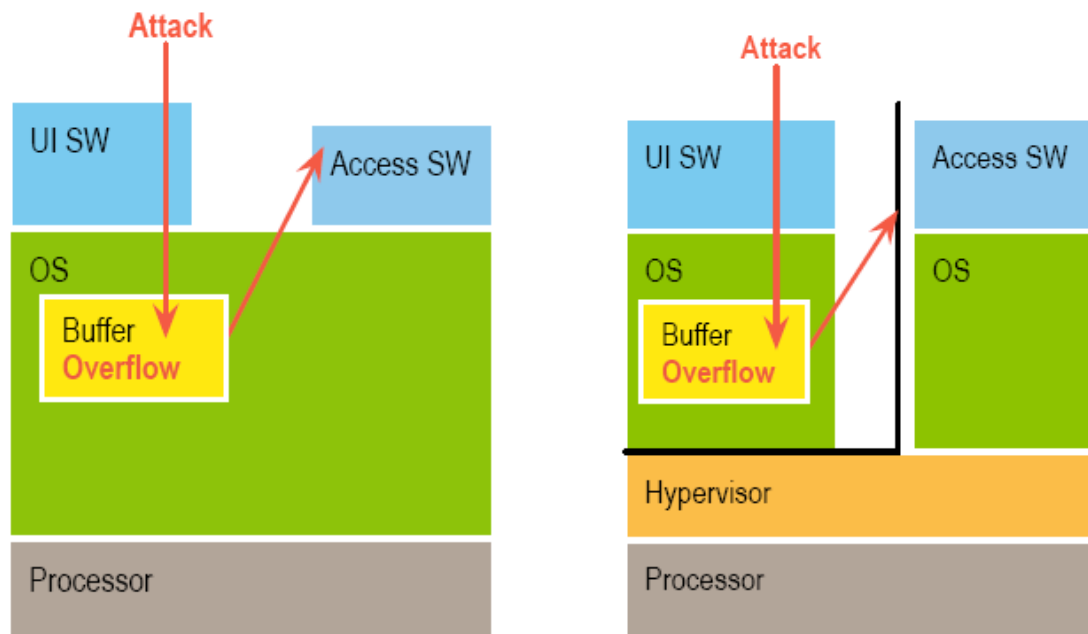


Benefit: Real Security



Be open.
Be safe.

- Complete isolation of components / VMs
 - Attack of a component cannot impact other sub-systems
 - All components are *deprivileged*
 - Without virtualization, high level OS – 100kloc – executes privileged



Benefit: Reliability, Fault Detection, Recovery



*Be open.
Be safe.*

- Complete isolation of components / VMs
 - Failure of a component cannot impact other sub-systems
 - Components can be automatically shut down and restarted on fault detection
 - Provides seamless user experience

- Particularly important for Device Drivers
 - Highest density of bugs in drivers
 - Most common cause of OS failure and system instability
 - Solution: deprive and componentize device drivers!

Benefit: System Management and Update



*Be open.
Be safe.*

- System software no longer static
 - But how to update without impacting usability?
 - Emerging technologies: OMA DM, FOTA for remote management

- Components are a deployment vehicle!
 - Package system software in a virtualized component
 - Hypervisor can receive over network and:
 - Install into executing system
 - Update executing component

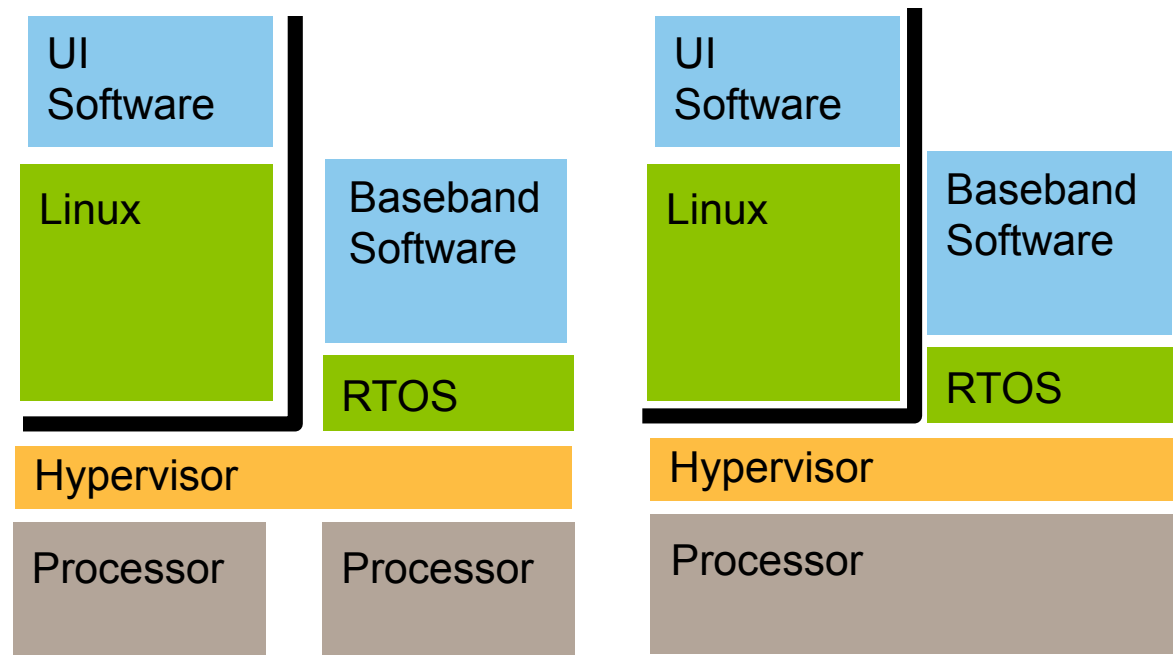
Benefit: Certification Re-use



*Be open.
Be safe.*

- Phones need to be certified to comply with communication standards
 - Any change that (potentially) affects comms needs re-certification
 - UI part of system changes frequently

- Encapsulation of UI
 - Provided by VM
 - Avoids need for costly recertification

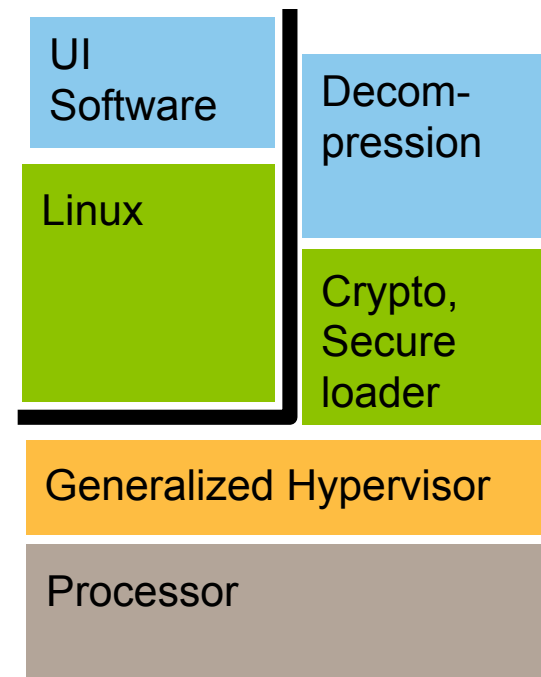


Benefit: IP Protection



*Be open.
Be safe.*

- High level OS on same system as highly valuable IP
 - Example: highly-efficient, proprietary compression algorithm deployed alongside Linux
- Operates in hostile environment
 - Reverse engineering of algorithms
- Need highly-trustworthy component that
 - Loads code from Flash into on-chip RAM
 - Decrypts code
 - Runs code protected from interference

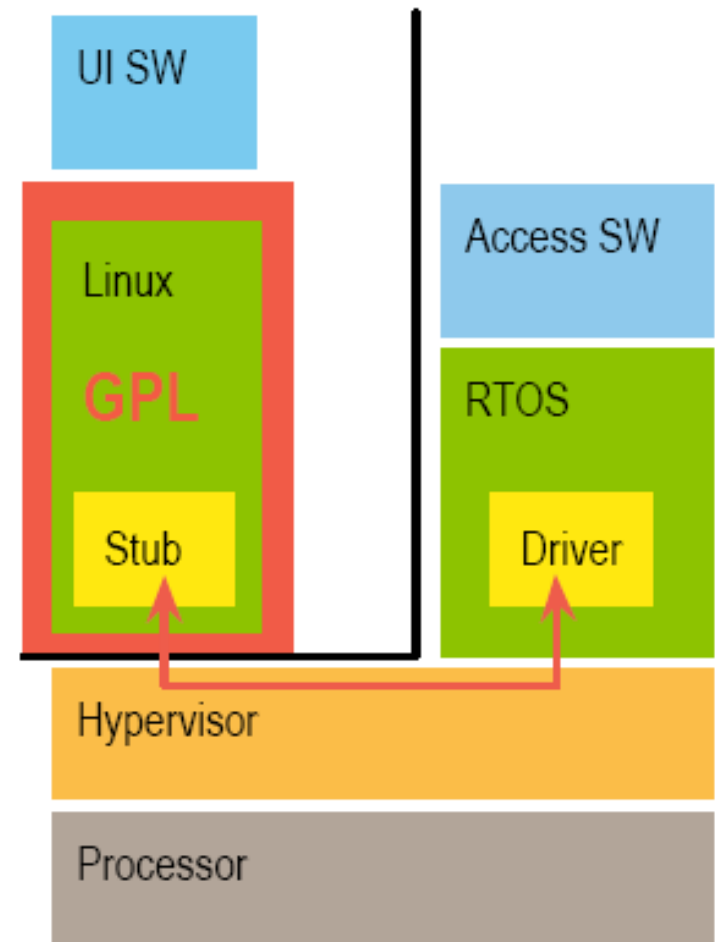


Benefit: License Separation



*Be open.
Be safe.*

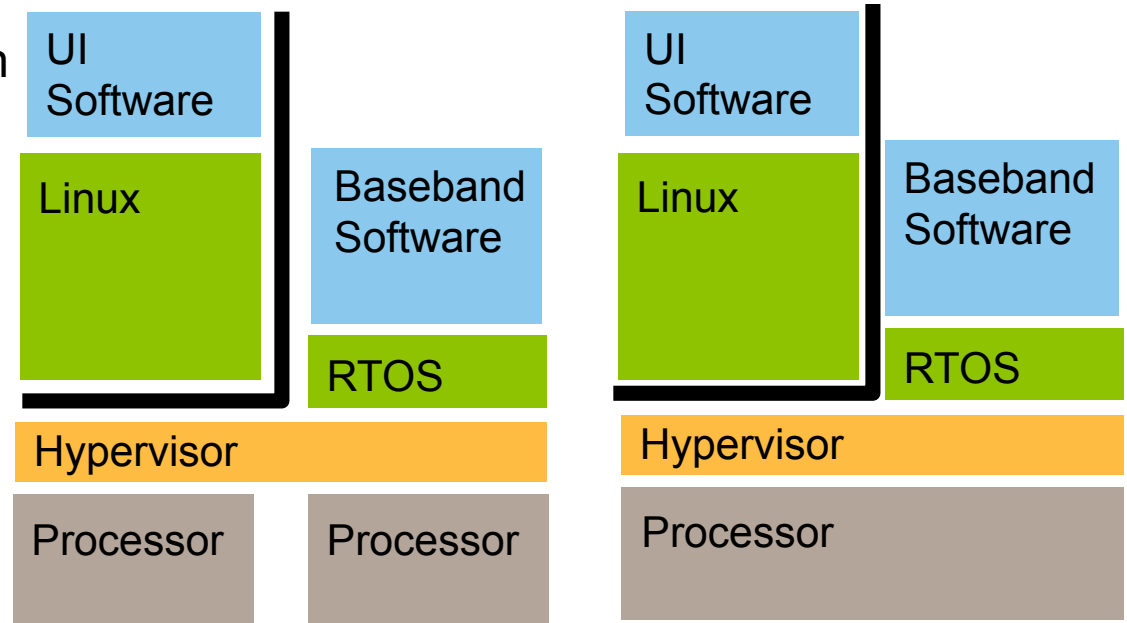
- GPL'd OS has many benefits
 - Royalty free
 - Independence from vendors
 - Strong and vibrant development community
 - Widespread deployment
 - Large ecosystem
- GPL often incompatible with proprietary licensing
- Separate via VMs



Use Case: Open Phone with User Configured OS



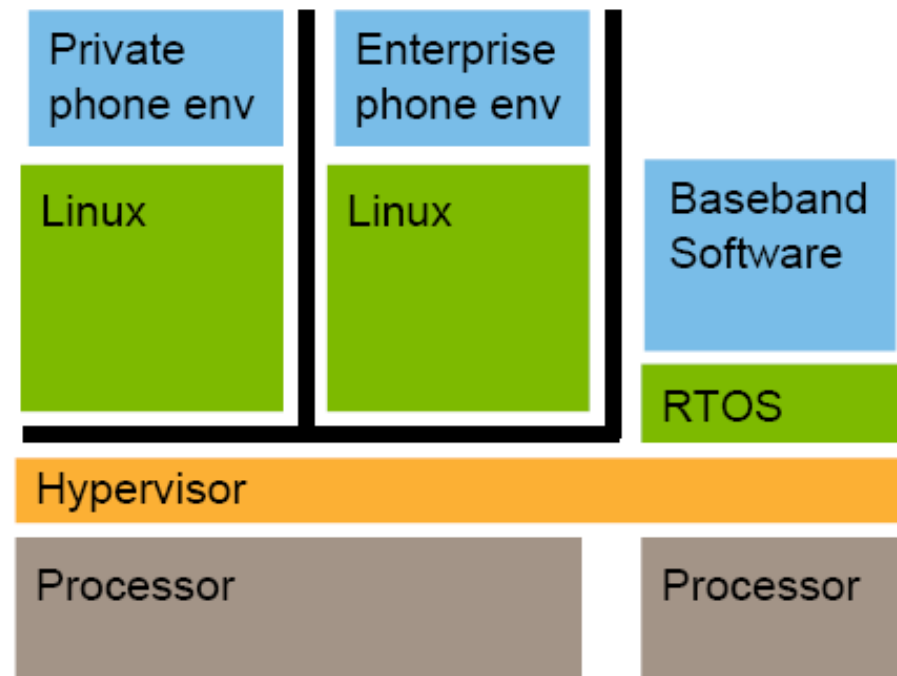
- Give users control over the application environment
 - Perfect match for Linux
- Requires strong encapsulation of application environment
 - Without undermining performance!



Use Case: Phone with Private and Enterprise Environment



- Two Environments:
 - Work phone environment integrated with enterprise IT system
 - Private phone environment contains sensitive personal data
- Mutual distrust between the environments
 - Strong isolation needed



Hey, don't we already have solutions?!



*Be open.
Be safe.*

- “Virtualization” is thrown around a lot
 - And it's great in enterprise markets
 - But embedded has unique constraints that these solutions fail to address
- Complexity
 - Course-grained VM's don't help
- Integration
 - Objective in enterprise server market is *de-integration*
- Device Sharing
 - Devices need to be componentized, not owned by a VM
- Security
 - Fine-grained control over security policies required
- TCB Size
 - TCB typically includes entire guest OS in enterprise virtualization!
- Performance
 - Severe resource limitation
- Conclusion: you need an effective combination of virtualization and componentization, purposely engineered in a solution for embedded

Agenda



*Be open.
Be safe.*

- The changes in the market
 - And how they affect you
- Virtualization and Componentization
 - And how it can help!
- **The OKL4 Approach**
 - **Engineering change with Secure HyperCell™ Technology**
- Open Kernel Labs

OKL4: The Embedded Hypervisor

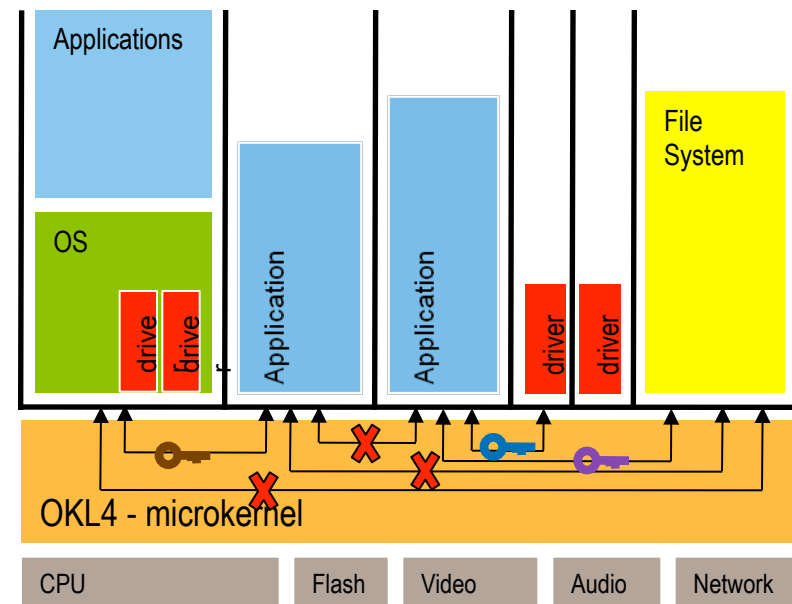


Be open.
Be safe.

Secure HyperCell™ Technology

- Goes well beyond the classical hypervisor model
- Enables virtualization and componentization
 - VM = OS plus its applications in a cell
 - Lightweight execution environments
 - Drivers
 - HW enforced isolation between cells
- Control over communication between cells
 - Required for mandatory access control
- Fast context switching and high performance inter-cell communication
- Highly trustworthy privileged code
 - Small., clean, and open source

User level “cells” contain components in HW-enforced address space isolation



Privileged mode software limited to the OKL4 microkernel

OKL4 Microkernel Technology: Ideally Addresses the Requirements of an Embedded Hypervisor



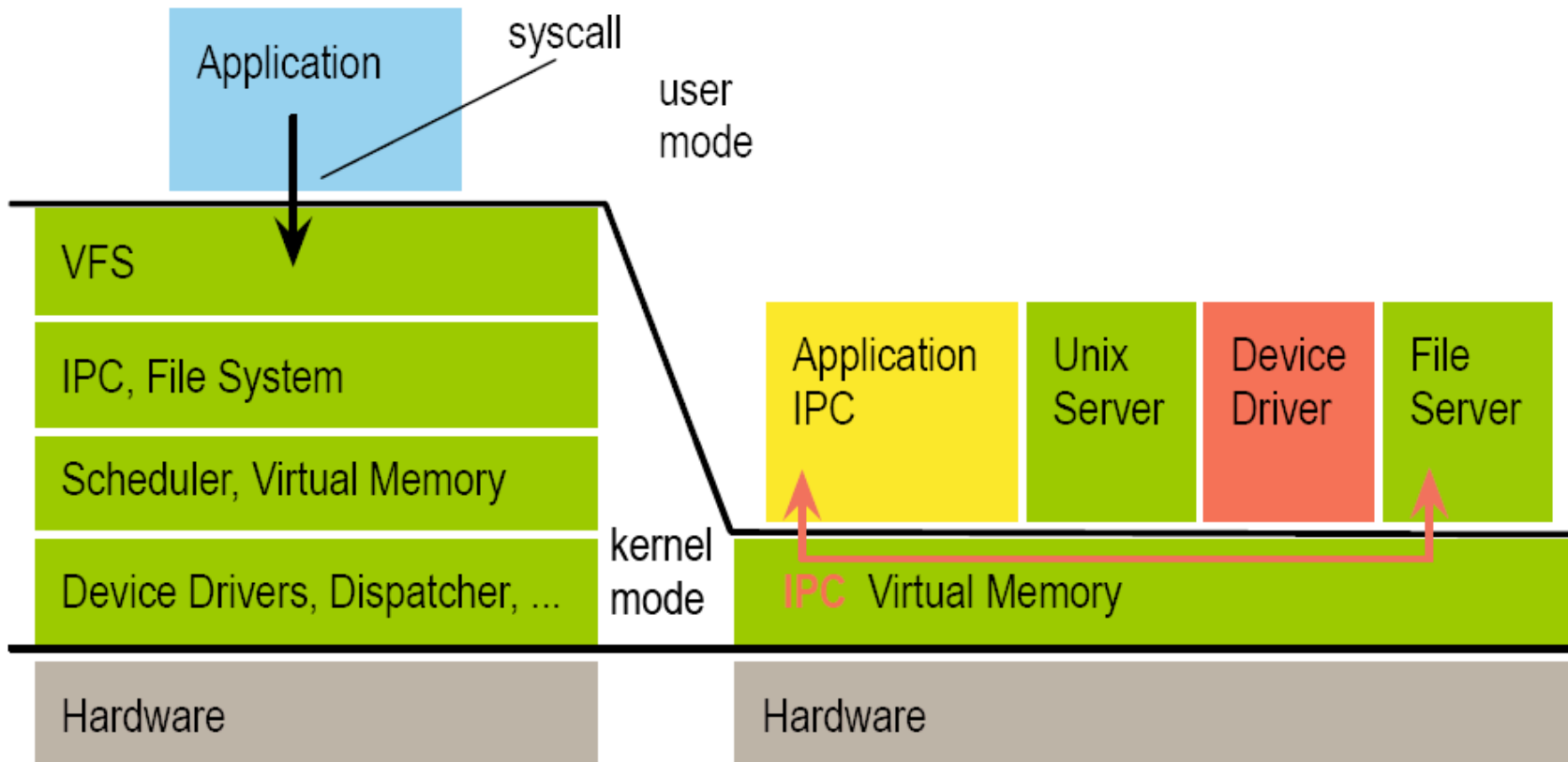
*Be open.
Be safe.*

- Small kernel providing core functionality
 - 10 kLOC, just enough for required mechanisms, no policy
 - No other code running in privileged mode, not even device drivers
 - Provide mechanisms for building arbitrary systems on top
 - Services can have 15 kLOC TCB
- Uses MMU to isolate address spaces
- Fast message-passing IPC operation
 - Performance close to the hardware limit
- Shared memory for bulk memory transfer
- 10-year track record of high-performance Linux virtualization
- Suitable as a basis for general operating systems
- Powerful and sophisticated security mechanisms
- *Open source!*

OKL4 Microkernel Technology



*Be open.
Be safe.*



OKL4 Performance



*Be open.
Be safe.*

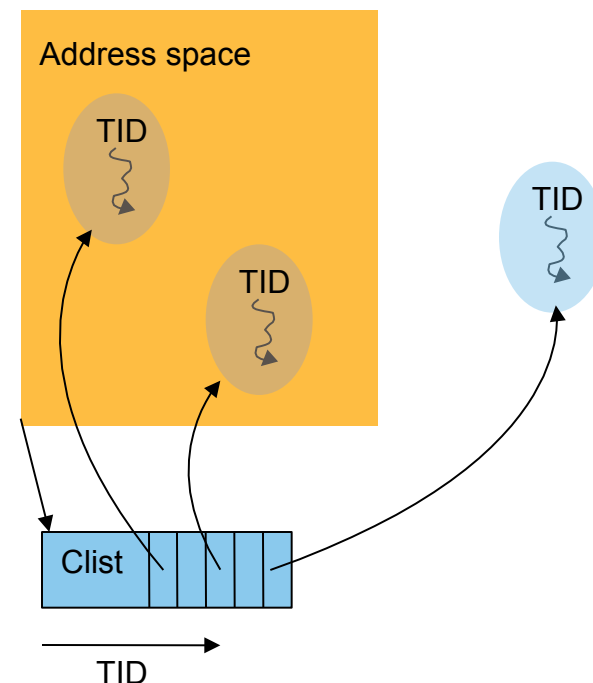
- OKL4 and OK Linux features OK Fast Address Space Switching (FASS) Technology
 - By taking advantage of ARM domains and FCSE extensions
 - “Faster than native” virtualized Linux context switching performance on ARMv5
- *Unmatched IPC performance* operating as the backbone for interrupt delivery
 - OKL4 IPC implemented in assembly “fastpaths” to achieve zero-copy overhead
 - Low latency interrupt handling (best-case and worst-case)
- OKL4 high-performance inter-cell communication
 - Communication channels built over shared memory and lightweight OKL4 IPC
 - Allows for highly cooperative yet modular systems without sacrificing performance
- OKL4 microkernel design provides *strong real-time guarantees*
 - By providing lightweight primitives backed by short kernel execution paths
 - Enables consolidation of real-time and non-real-time system components on a single-core solution

OKL4 Resource and Security Management



Be open.
Be safe.

- Resources controlled by kernel-protected *capabilities*
- Capability conveys privilege
- Efficient resource delegation by providing set of caps
- Subject to system-wide security and resource-management policies defined by system designer
- Static resource partitioning
- *Secure HyperCell™ technology* provides security management framework

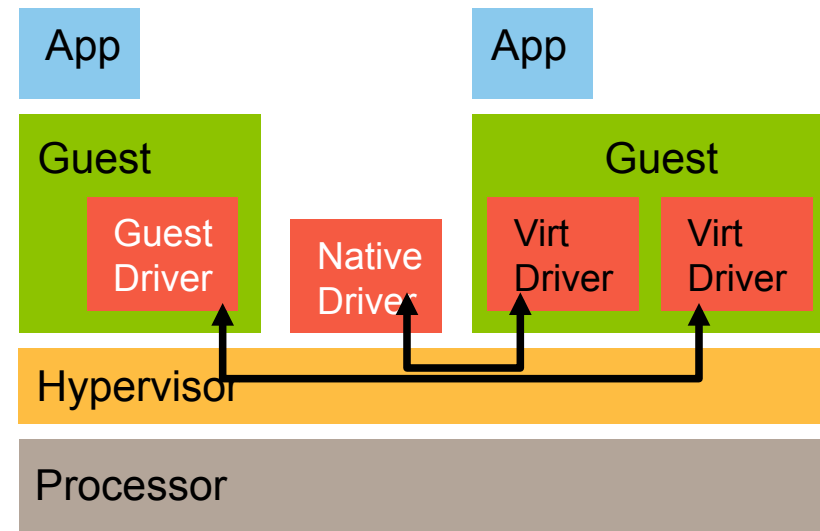


OKL4 Embedded Hypervisor Device Drivers



*Be open.
Be safe.*

- Simple embedded systems
 - Drivers owned by OS
 - Drivers privileged
- With OKL4 embedded hypervisor
 - Guest OS or OKL4 native drivers
 - User level drivers
 - Device sharing mechanisms
- Benefits of OKL4 native drivers
 - Smaller TCB
 - Fault isolation
 - Control access using caps
 - Improves separation of trusted and un-trusted subsystems



OKL4 Lightweight Execution Environments



Be open.
Be safe.

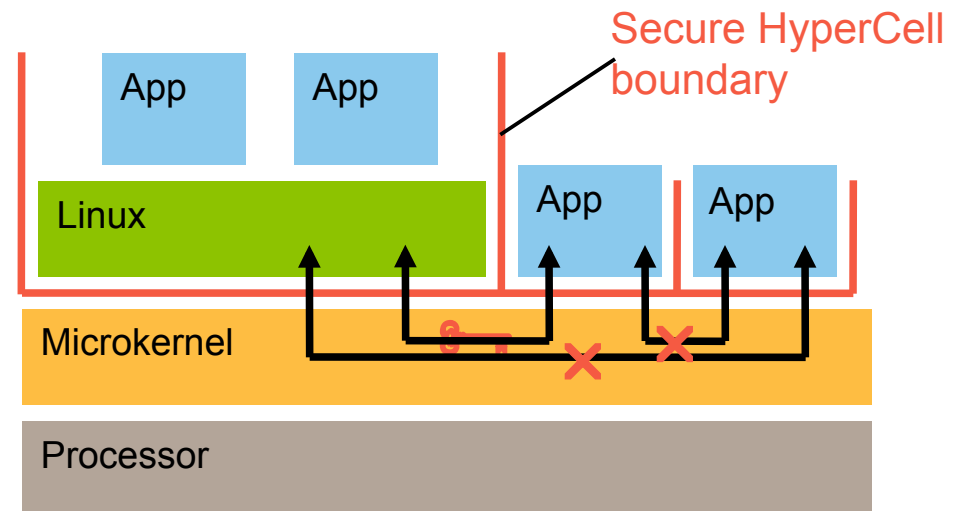
- Conventional virtualization solutions only support running complete operating systems (Linux, WinCE, ...) in a virtual machine
- OKL4 supports execution of *lightweight execution environments* (LWEE)
 - A LWEE executes an application
 - for system service directly on OKL4 without inclusion of a complete operating system
 - Critical services software can be hosted directly on top of the OKL4 microkernel
 - to reduce their *trusted computing base* (code they depend on for correct functionality)
 - Lower cost of cells without a guest OS allows practical application of finer grained isolation (i.e. More cells than what would be possible with vanilla VMs)
 - Separate security-sensitive and proprietary software from an

Minimized Trusted Computing Base



Be open.
Be safe.

- The **trusted computing base** (TCB) of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security, in the sense that bugs occurring inside the TCB might jeopardize the security properties of the entire system.
- The careful design and implementation of a system's trusted computing base is paramount to its overall security.
- Identify security-critical apps
 - crypto services
 - e-cash...
- Run native on OKL4 in their own cell
- Security policy defines allowed communication channels
- *TrustZone-like isolation*, but more general
- TCB: <15kLOC, much smaller than for a Linux application



Fast Deployment



Be open.
Be safe.

- OKL4 technology supports *true virtualization* over mere operating system co-location
 - Isolation of guest operating systems is enforced and therefore modifications to one virtual machine do not impact other virtual machines
 - Small changes to a particular virtual machine do not mandate re-qualification of the entire system

- OKL4 provides a lightweight POSIX library implemented directly over microkernel primitives

- Eases migration of existing software components to OKL4 and allows for reuse of

third-party POSIX conformant software (e.g. GNOME Mobile Platform, LWIP, .29)

- OKL4 System Builder Tool constructs a system image

Multi-Processing Support



Be open.
Be safe.

- OKL4 supports UP, SMT and SMP configurations
 - On real hardware including proprietary 6-way multi-processing systems, and quad-core Intel-based machines
 - Running real work loads ranging from RTOS's to consumer desktop OSes
- OKL4 provides a flexible multi-processing model to deliver *maximum* performance and *scalability* on a given hardware platform
 - Kernel mechanisms for multi-processing are selected to match a given set of hardware characteristics (interrupt + memory latencies, synchronization costs)

OKL

4 multi-processing support extracts full potential for parallelism from hardware

→ OKL4 features an innovative $O(1)$ scheduler designed with multi-processing in mind 30

- Supports priority-based load balancing across execution units
- Minimizes inter-processor communication and sharing of data to ensure scalability

Some Other Benefits of OKL4 Embedded Hypervisor Use



*Be open.
Be safe.*

- Reduce deployment cost by
 - Consolidating physically separate systems onto shared processing resources to eliminate unused capacity
 - Implementing a more modular and therefore more maintainable software architecture
- Reduce time to market
 - Replace OS porting with VM integration when adding new applications
 - Reuse legacy SW components in their legacy OS environment “as-is” alongside a new application OS
- Reduce development effort and improve return on software development investment by
 - Reuse one implementation of a critical function in multiple products with different hardware and operating system components
 - Deploy the same software architecture on single and multicore systems
 - Applies either over time or across products at a given time

Agenda



*Be open.
Be safe.*

- The changes in the market
 - And how they affect you
- Virtualization and Componentization
 - And how it can help!
- The OKL4 Approach
 - Engineering change with Secure HyperCell™ Technology
- **Open Kernel Labs**

Who We Are



*Be open.
Be safe.*

- The best choice for an embedded hypervisor solution
- Founded in 2006 with headquarters in the US and engineering in Sydney, Australia
- With ownership of technology resulting from a 10+ year program of research and development on microkernel technology across 7 HW architectures
- Offering it's customers unique access to the fruits of continuing research
 - A commercialization vehicle for NICTA, Australia's Centre of Excellence in ICT research and an ongoing partnership with University of New South Wales (UNSW)
- The provider of an embedded hypervisor solution that is already deployed in 100+ million of end-user devices

Transfer of NICTA Research Outcomes Into OK Labs Products



*Be open.
Be safe.*

- Military-strength security and isolation
 - Based on further development of capability-based access control
 - Kernel resources also subject to system's resource management policies
 - Mathematical isolation proofs
- Mathematical proof of implementation correctness
 - Kernel proven free of security-relevant bugs
 - Possible due to small size of privileged-mode code base
- Strict analysis of real-time properties
 - Provide real guarantees for worst-case latencies

Deployed in 100+ Million Devices!



*Be open.
Be safe.*

→ Mobile Phones

- 2006- Toshiba W47T CDMA phone selling in Japan
- 2007- 3G phones from HTC, LG, shipping since July
- 2008- Samsung SPH-m800 Instinct™ and HTC Dream (G1) with Android
- Linux phones later this year



→ Products in other industry verticals in pipeline

- HD IP-TV set top box announced, to ship this year
- More to come



Thank You



*Be open.
Be safe.*



Open Kernel Labs™

Be open. Be safe.

Josh Matthews
Field Application Engineer

Open Kernel Labs
jmatthews@ok-labs.com