Open Kernel Labs™

Be open. Be safe.

Next-Generation Embedded Operating Systems

Gernot Heiser

Founder and CTO, *Open Kernel Labs*

Professor of Operating Systems, *UNSW*

Program Leader, *NICTA*

August 2007

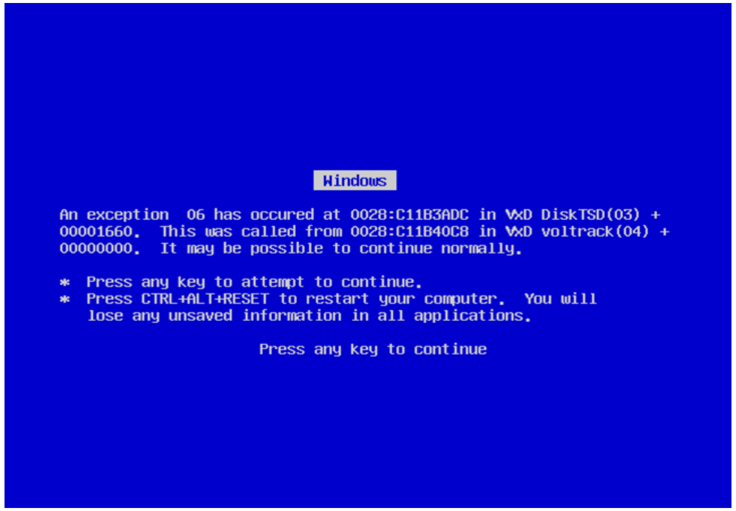
Embedded Systems are Everywhere

Let's think about the implications...

Lessons from Desktop Systems

Desktop Computers Suck

- They crash
- They get cracked
- They get infected



```
                              Windows

 An exception  06 has occured at 0028:C11B3ADC in VxD DiskTSD(03) +
 00001660.  This was called from 0028:C11B40C8 in VxD voltrack(04) +
 00000000.  It may be possible to continue normally.

 *  Press any key to attempt to continue.
 *  Press CTRL+ALT+RESET to restart your computer.  You will
    lose any unsaved information in all applications.

                          Press any key to continue
```

How about embedded systems?

Wireless Everywhere!

- Bank accounts
  - → Is someone monitoring your financial transactions?
  - → Is someone taking money out of your account?
- Automobiles
  - → Is someone changing your engine settings?
  - → Is someone manipulating your breaks?
- Health cards
  - → Is someone accessing your medical history?
  - → Is someone changing your medication?
- Your home
  - → Is someone watching you at home?
  - → Is someone entering while you are away?

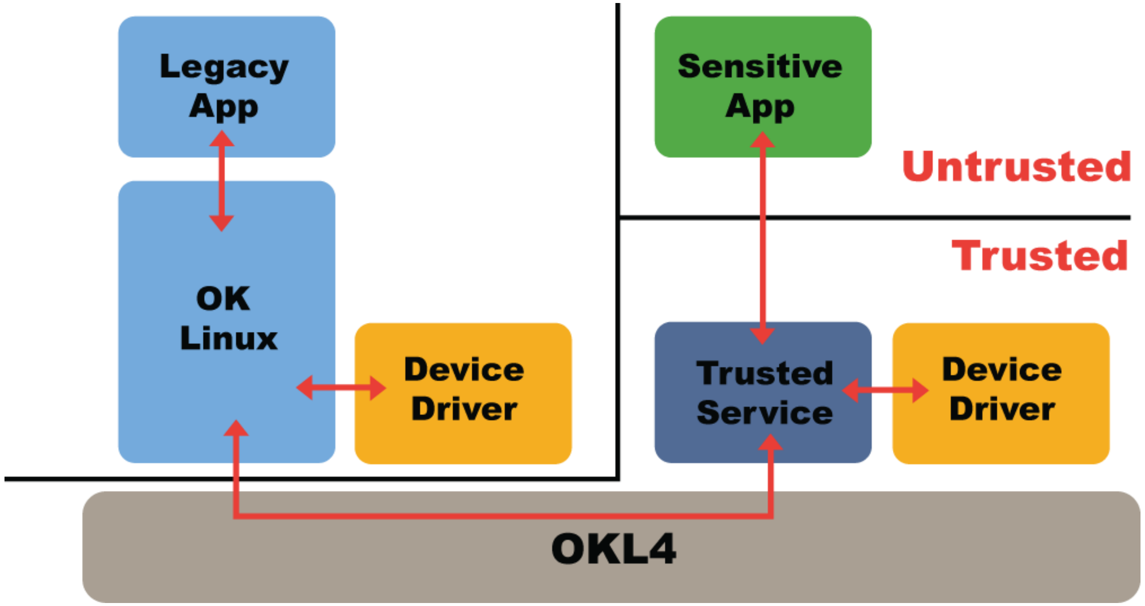
Computer Unreliability — Why?

- Complexity is the arch-enemy of reliability
  - Complex systems are impossible to understand completely
  - Complex systems are faulty
- Software systems are incredibly complex
  - Smartphones have 5-7 M lines of code (LOC)
  - Cars contain Gigabytes of software
  - Future systems will be even more complex
- Software is buggy
  - Good-quality software has about 1 bug per 1,000 LOC
  - Bug count grows super-linearly in code size
  - Systems have thousands and thousands of bugs

Reliable Systems — How?

- Need a high-performance microkernel
- Need certainty it provides *right mechanisms*
- Need certainty its *implementation is correct*
- Need *credible timing model*
- Need *software-engineering infrastructure*

OKL4: Embedded OS and Virtualization



- Small OKL4 micro-kernel (10 kLOC)
  - unbeaten IPC performance
  - native real-time programming env
- Virtualization for standard high-level OS API (Linux)
  - Full binary compatibility
- Developed at UNSW and NICTA, spun out into startup
- Open Kernel Labs markets and continues development
- Joint venture of Open Kernel Labs and NICTA
  - develop next-generation technology based on OKL4

OKL4 Commercial Deployment

- Shipped by QUALCOMM on their latest chipsets
- First OKL4 phone on the market: Toshiba W47T
  - → on sale in Japan since late 2006
- More handsets to hit market in next 12 months
  - → US, Korean manufacturers
- Products in other industry verticals in pipeline

Reliable Systems — How?

- Need a high-performance microkernel
  - → This exists: OKL4
- Need certainty it provides *right mechanisms*
  - → can support secure systems (encapsulation etc)
- Need certainty its *implementation is correct*
- Need *credible timing model*
- Need *software-engineering infrastructure*

Reliable Systems — How?

- Need a high-performance microkernel
  - → This exists: OKL4
- Need certainty it provides *right mechanisms*
  - → can support secure systems (encapsulation etc)
  - → NICTA project seL4
- Need certainty its *implementation is correct*
  - → implementation matches specification
- Need *credible timing model*
- Need *software-engineering infrastructure*

Reliable Systems — How?

- Need a high-performance microkernel
  - → This exists: OKL4
- Need certainty it provides *right mechanisms*
  - → can support secure systems (encapsulation etc)
  - → NICTA project seL4
- Need certainty its *implementation is correct*
  - → implementation matches specification
  - → NICTA project L4.verified
- Need *credible timing model*
  - → actual worst-case latencies, based on sound methodology

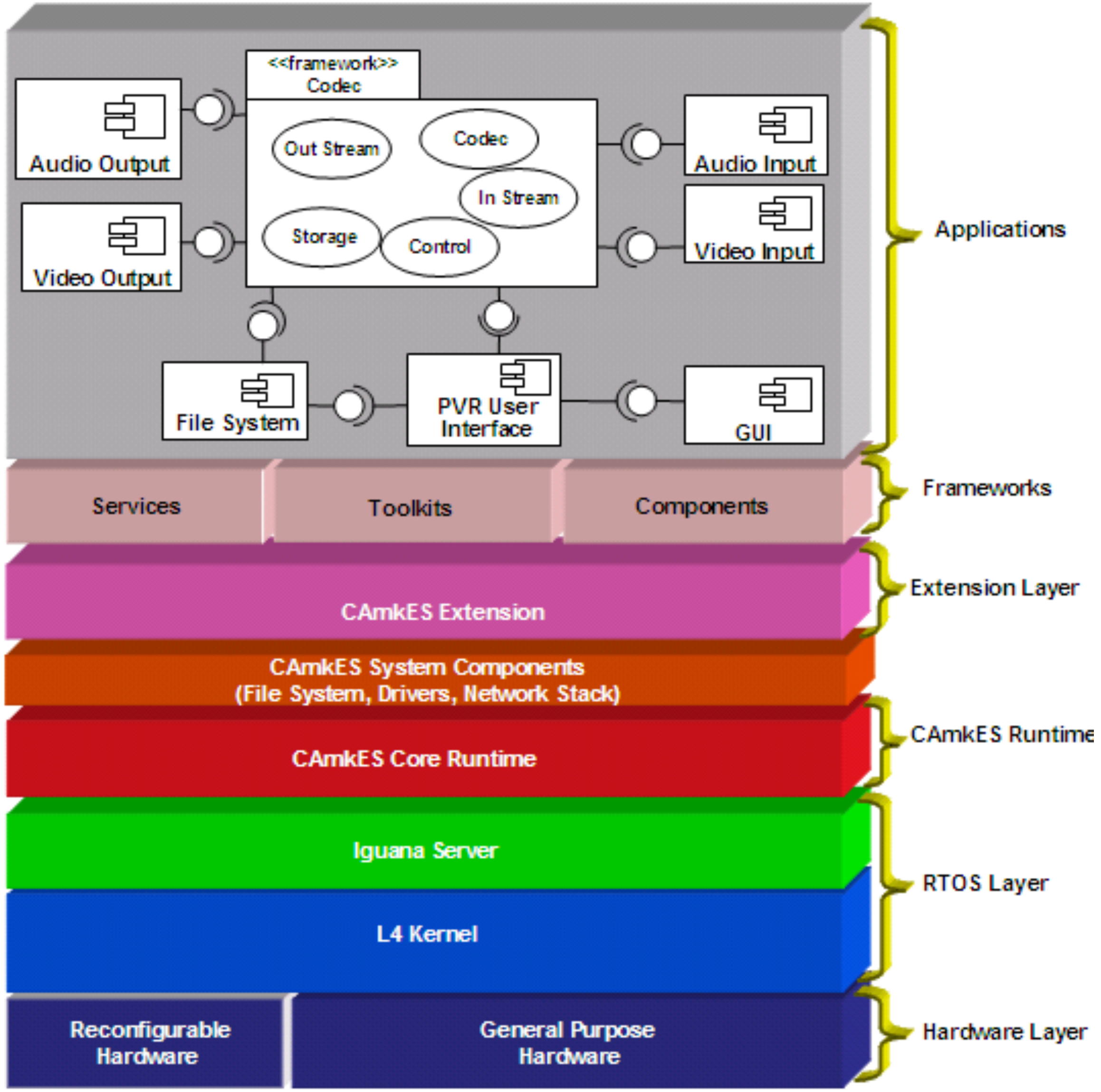
- Need *software-engineering infrastructure*

Reliable Systems — How?

- Need a high-performance microkernel
  - → This exists: OKL4
- Need certainty it provides *right mechanisms*
  - → can support secure systems (encapsulation etc)
  - → NICTA project seL4
- Need certainty its *implementation is correct*
  - → implementation matches specification
  - → NICTA project L4.verified
- Need *credible timing model*
  - → actual worst-case latencies, based on sound methodology
  - → NICTA project Potoroo
- Need *software-engineering infrastructure*
  - → support for building large and complex systems

CamkES Project: Component Architecture

- Software-Engineering framework for OKL4
  - → *support highly modular systems*
  - → *components encapsulated by kernel*
- Designed for embedded systems
  - → *very lightweight*
  - → *no overhead for unused features (eg. dynamic components)*



Reliable Systems — How?

- Need a high-performance microkernel
  - → This exists: OKL4
- Need certainty it provides *right mechanisms*
  - → can support secure systems (encapsulation etc)
  - → NICTA project seL4
- Need certainty its *implementation is correct*
  - → implementation matches specification
  - → NICTA project L4.verified
- Need *credible timing model*
  - → actual worst-case latencies, based on sound methodology
  - → NICTA project Potoroo
- Need *software-engineering infrastructure*
  - → support for building large and complex systems
  - → NICTA project CAmkES

Next-Generation Embedded Operating Systems

- Need to be ultra-reliable
  - → based on microkernels
  - → provably-secure mechanisms
  - → provably-correct implementation
  - → credible timing models
- Need to be highly componentised
  - → components protected by microkernel address spaces
  - → can isolate faults, support run-time upgrades
  - → can prove correctness of components, or at least confinement of faults
- NICTA/OK Partnership will deliver this
  - → core technology OKL4 already on market and deployed on products
  - → research agenda for next generation completed next year
  - → commercial availability within 2-3 years

Open Kernel Labs™

Be open. Be safe.

Gernot Heiser
Founder and CTO

Open Kernel Labs
t +61 28306 0550
gernot@ok-labs.com