NICTA

Towards an OS Platform for Truly Dependable Real-Time Systems

Gernot Heiser
NICTA and University of New South Wales, Sydney

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners

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

What’s Next?

Complexity Threatens Dependability

- Massive functionality ⇒ huge software stacks
  - Expensive recalls of CE devices

- Increasing usability requirements
  - Wearable or implanted medical devices
  - Patient-operated
  - GUIs next to life-critical functionality

- On-going integration of critical and entertainment functions
  - Automotive infotainment and engine control

Safety Issues Are Real!

Malicious remote operation of car

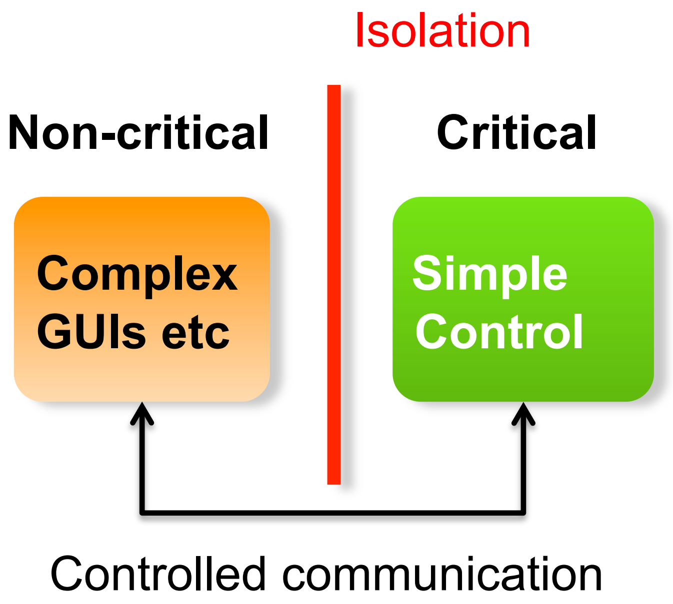
Malicious remote control of pacemaker

Root Cause: Complexity

NICTA

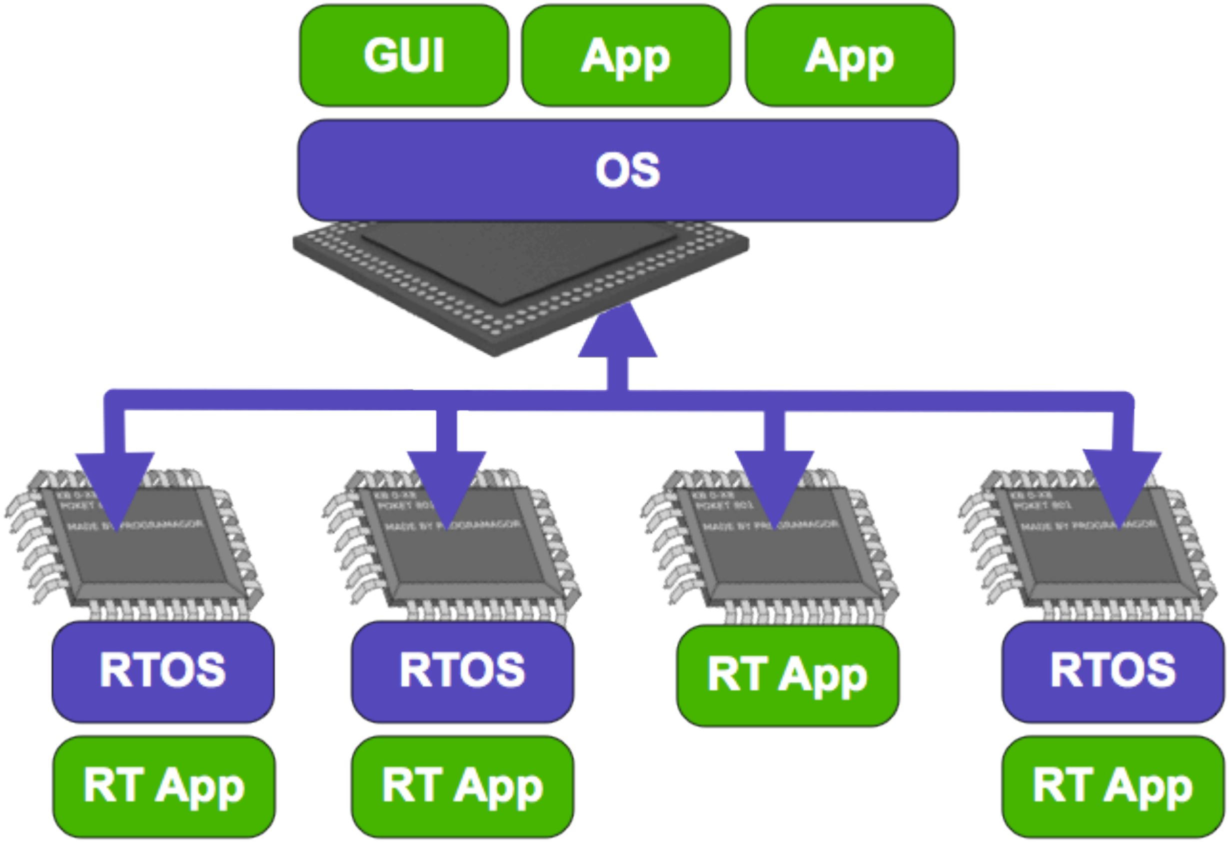
- Complexity of critical devices will continue to grow
  - Critical systems with millions of lines of code (LOC)
- We need to learn to ensure *dependability* despite complexity
  - Need to *guarantee* dependability
- Correctness guarantees for MLOCs unfeasible

- Key to solution: *isolation*
  - … with controlled communication



Isolation: Physical

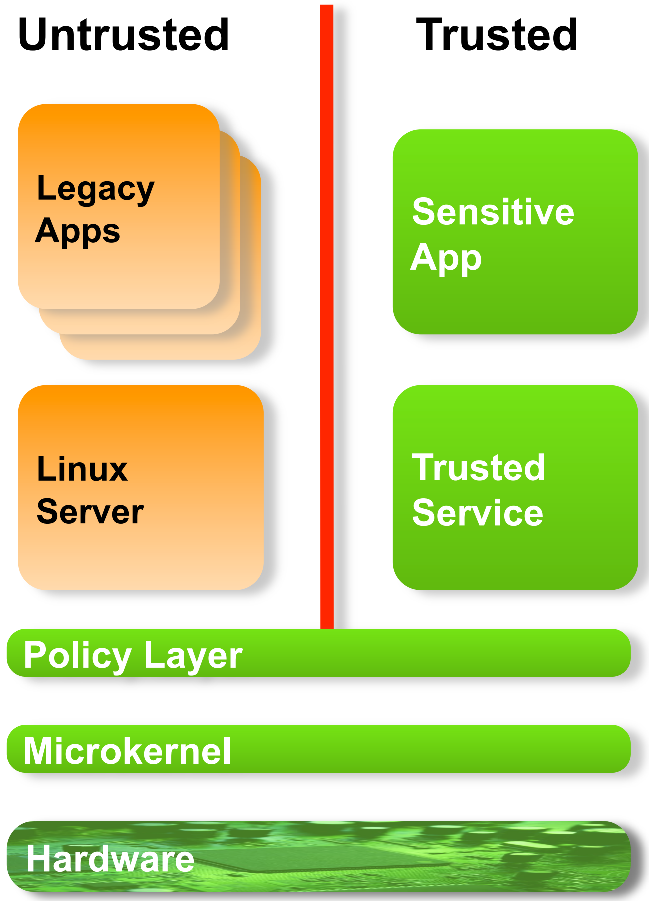
Dedicated CPUs for critical tasks



Cost: Space, costly interconnects, poor use of hardware

Isolation: Logical

- Protect critical components by sandboxing complex components
- Provide tightly-controlled communication channels
- *Trustworthy microkernel* provides general mechanisms to enforce isolation
- Policy layer defines access rights
- Microkernel becomes core of trusted computing base
  - System trustworthiness only as good as microkernel



Isolation Requirements

To guarantee dependability, following must be guaranteed:

- Isolation infrastructure impact must be specified
  - To allow reason about operation of isolated critical instances
- Isolation infrastructure must behave as specified
  - Functional correctness
  - Bounded and know worst-case latencies
- Isolation infrastructure must provide actual isolation
  - Integrity guarantees
  - Temporal isolation

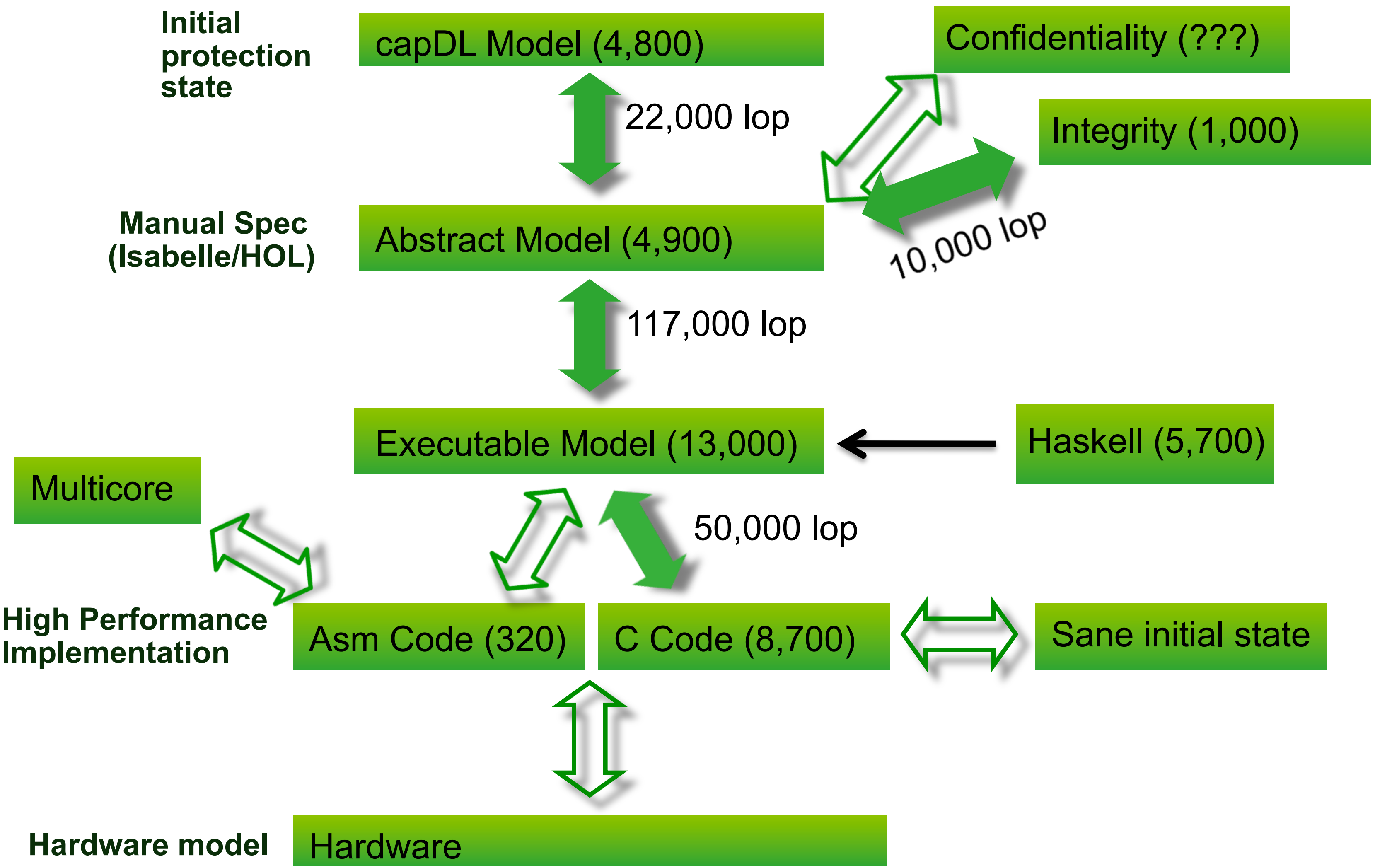
NICTA Trustworthy Systems Agenda

NICTA

1. **Ensure microkernel (seL4) dependability**
   - Formal specification of functionality
   - Proof of functional correctness of implementation
   - Proof of safety/security properties
   - WCET guarantees

2. **Lift microkernel guarantees to whole system**
   - Use kernel correctness and integrity to guarantee critical functionality
   - Ensure correctness of balance of trusted computing base
   - Prove dependability of complete system

Kernel Functional Verification



Kernel Worst-Case Execution Time

Issues for WCET analysis of seL4

- Need knowledge of worst-case interrupt-latency
  - Longest non-preemptible path + IRQ delivery cost
  - seL4 runs with interrupts disabled
    - System calls in well-designed microkernel are short!
    - Strategic preemption points in long-running operations
    - Optimal average-case performance with reasonable worst-case
- Applications also need to know cost of system calls
  - Need WCET analysis of *all* possible code paths

Kernel Worst-Case Execution Time

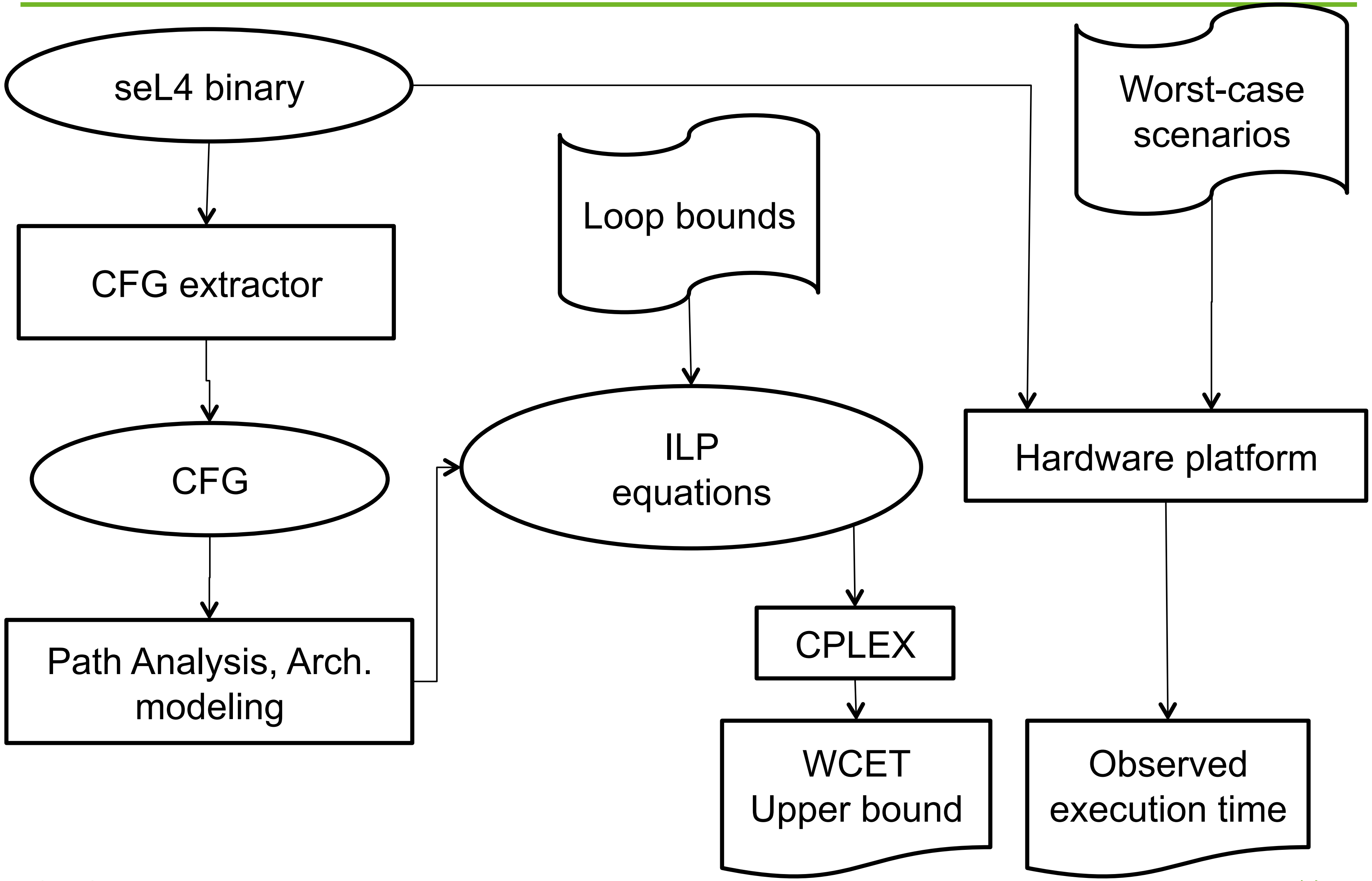
Challenges for WCET analysis of OS kernels in general:

- Kernel code notoriously unstructured
- Low-level system-specific instructions
- Context-switching
- Assembly code

seL4-specific advantages:

- (Relatively) structured design (evolved from Haskell prototype)
- Event-based kernel (single kernel stack)
- Small (as far as operating systems go!)
- No function pointers in C
- Preemption points are explicit and preserve code structure
- Memory allocation performed in userspace

WCET analysis process



Evaluation platform

- OMAP3-based BeagleBoard-xM
  - ARM Cortex-A8 @ 800 MHz
  - 128 MB memory
  - 32KB 4-way set-associative L1 instruction cache
  - Disabled data cache
    - Cache analysis did not scale
  - Disabled branch predictors
    - Pipeline model too simple
  - Modeled single-issue pipeline
    - A8 is dual-issue

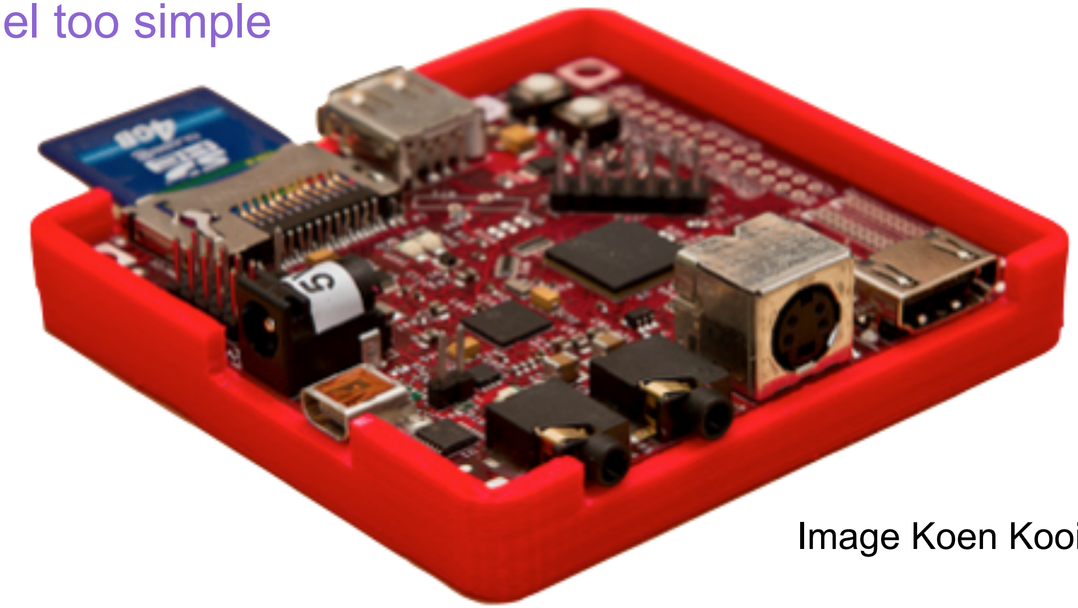
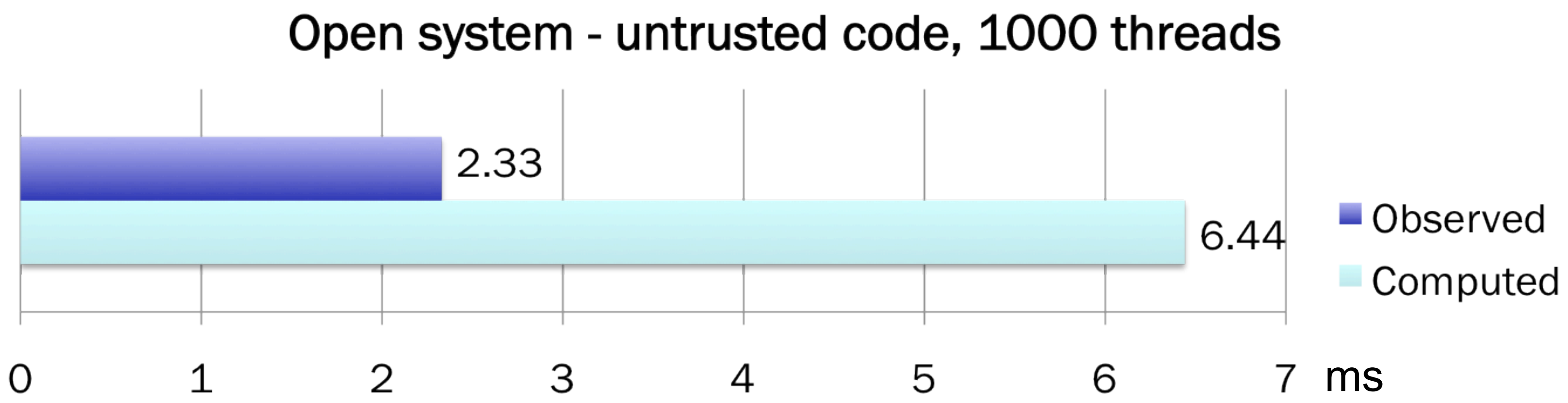


Image Koen Kooi CC-SA 2.0

Early Days…

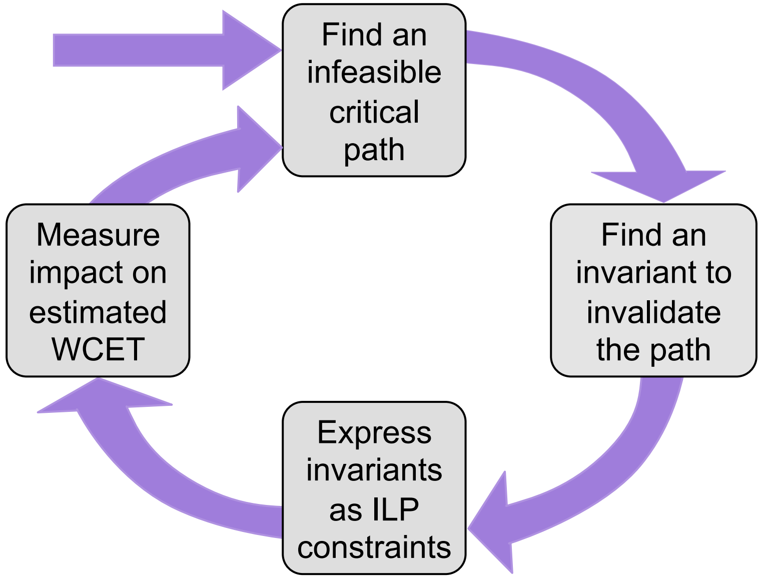


Open system - untrusted code, 1000 threads
2.33
6.44
Observed
Computed
0
1
2
3
4
5
6
7
ms

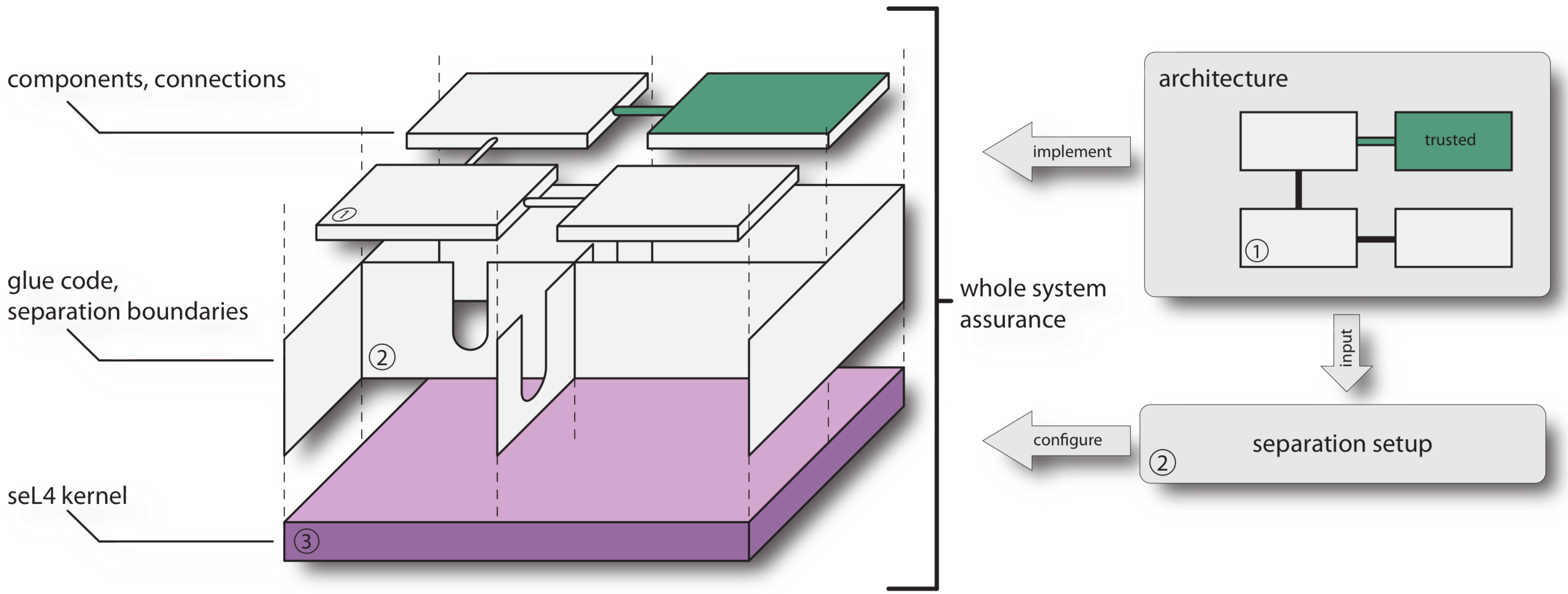
Closed system
0.1182
0.8473
Observed
Computed
0
0.2
0.4
0.6
0.8
1
ms

Improve WCET

- Analysis helps placing preemption points
  - Will be able to reduce WCET by 1–2 orders of magnitude
- Knowledge about seL4 can eliminate many paths
  - Invariants proved during verification
  - E.g. loop iteration counts, non-interference
  - Can easily prove new invariants
- Power-of-2 alignment of kernel objects constrain cache layout
  - May make D-cache analysis feasible
- Improved pipeline modelling
  - May have practical approach for complex pipelines
- Aim: IRQ WCET < 10 μs

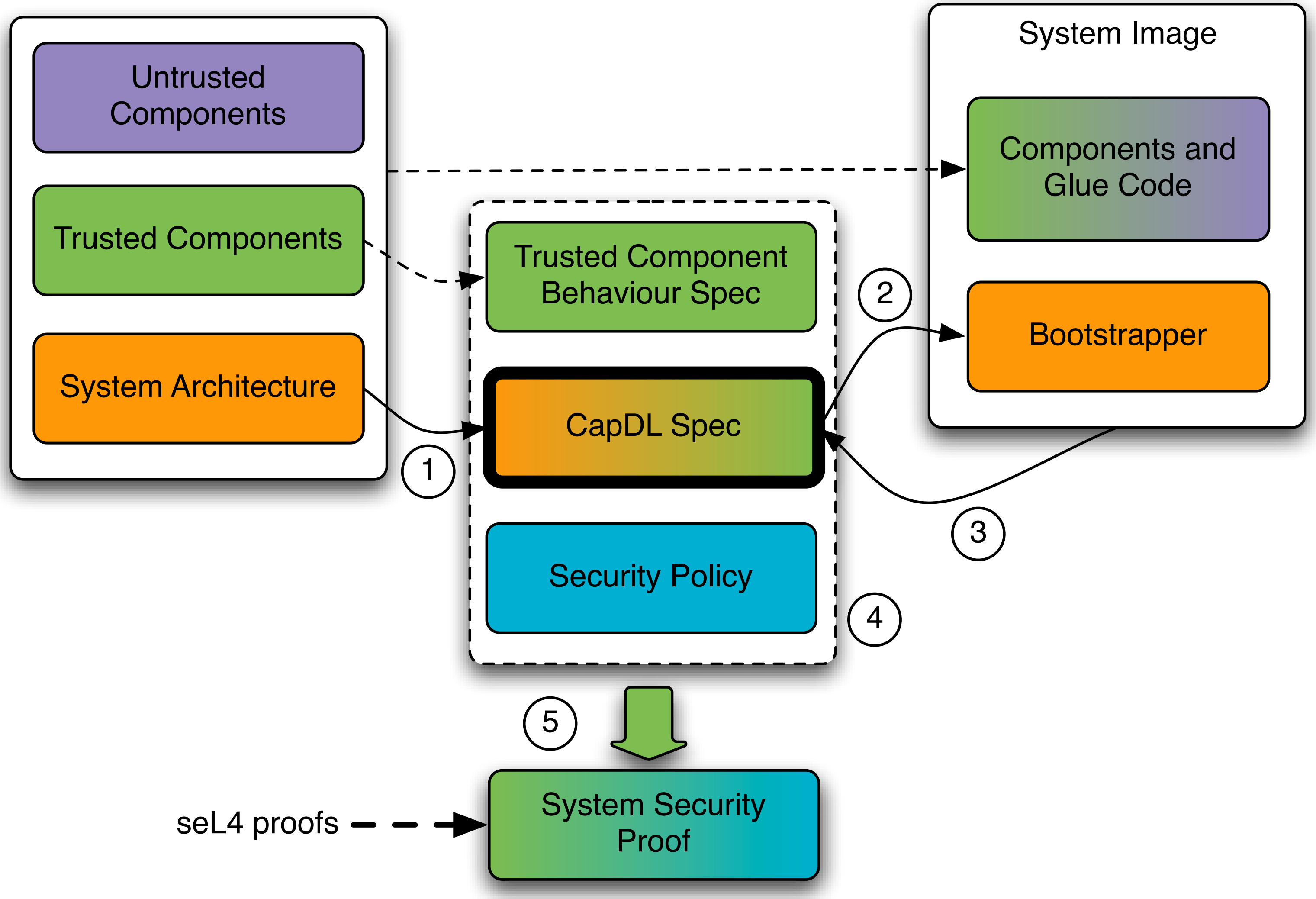


Full-System Guarantees



- Build system with minimal TCB
- Formalize and prove security properties about architecture
- Prove correctness of trusted components
- Prove correctness of setup
- Prove temporal properties (isolation, WCET, …)
- Maintain performance

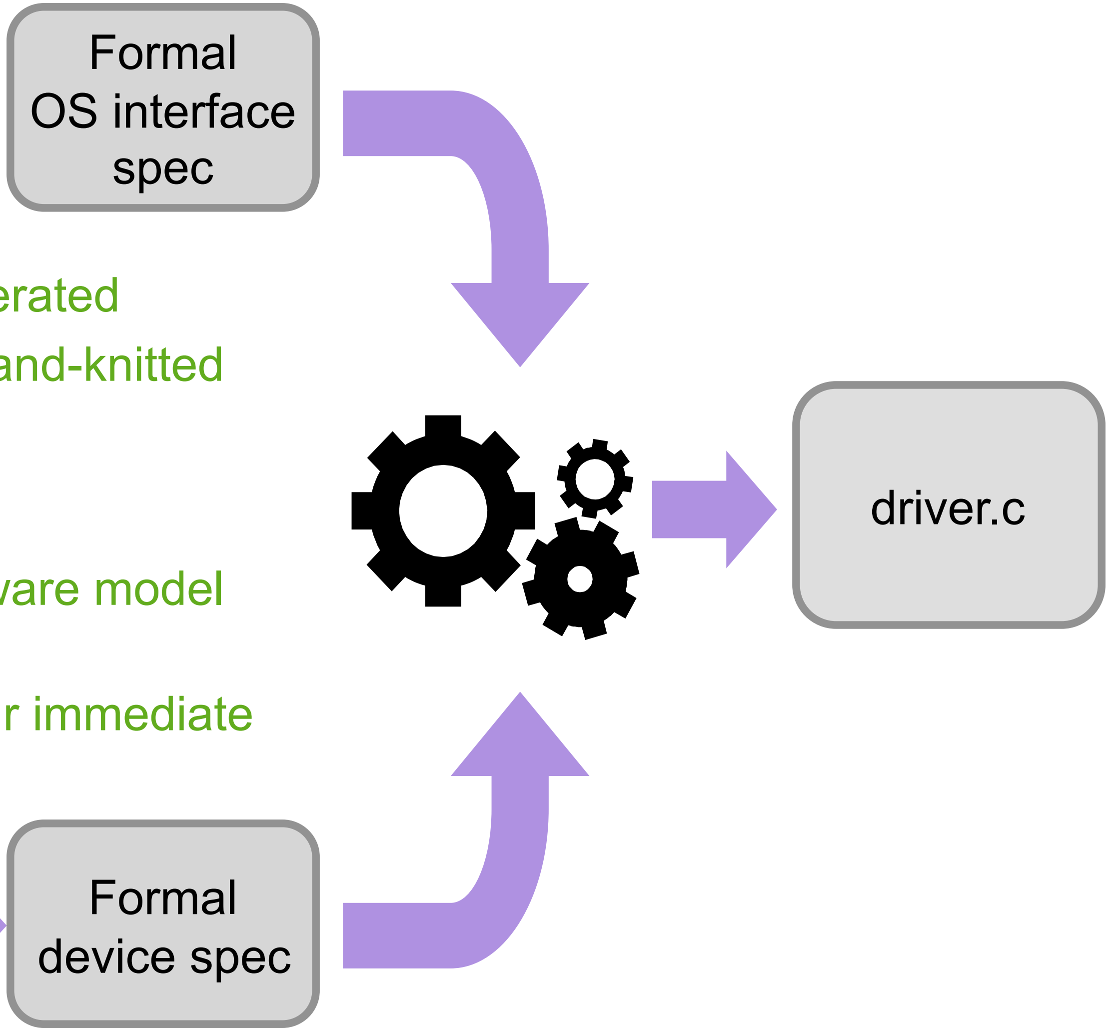
Specifying Access Control

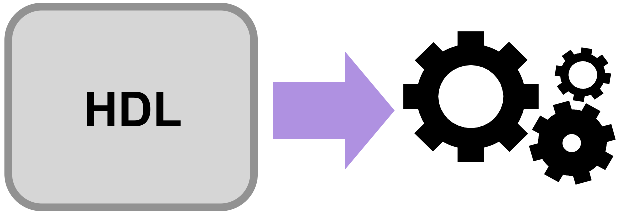


Device Drivers: Correct By Construction

NICTA

- Correct driver synthesis
  - given model of driver interface, basic behaviour, and hardware
  - driver is automatically generated
  - performance as good as hand-knitted
- Challenge: device spec
- Vision:
  - automatically extract hardware model from HDL description
  - potential impact beyond our immediate agenda





Complex Yet Dependable Systems?

- A first step has been taken: seL4 is a dependable base
  - Proof of functional correctness, integrity
  - Feasibility of WCET analysis
- Progress on full-system properties
  - capDL refinement + integrity
- Much remains to be done
  - Missing bits in kernel verification
  - Verification of large TCB components
    - Synthesis beats manual verification
    - Driver synthesis results encouraging
  - Overall system guarantees

mailto:gernot@nicta.com.au

Google: “ertos”