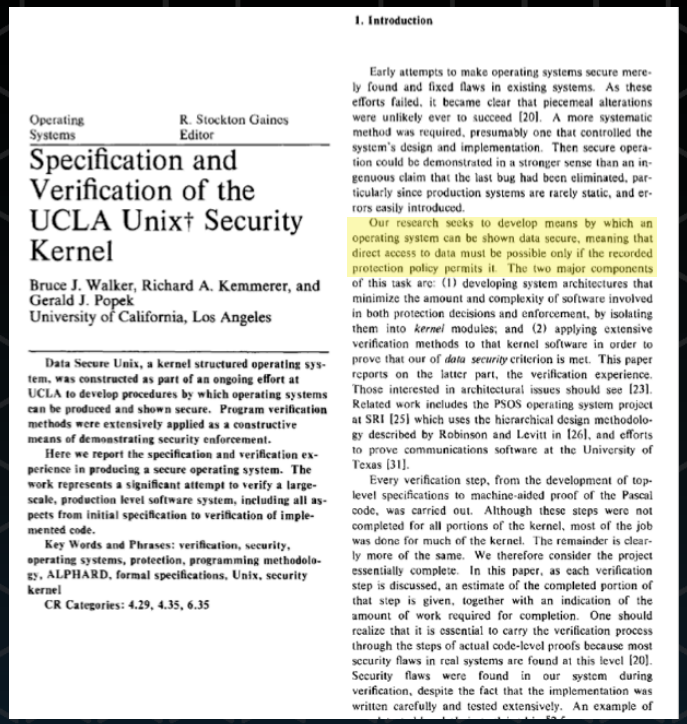

CSIRO
DATA 61
Verified seL4 on Secure RISC-V Processors
… and Other News in seL4 Land
Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
• LCA, Gold Coast, QLD, 2020-01-15
https://trustworthy.systems
seL4

What is seL4?

CSIRO DATA 61

A 30-Year Dream



Operating Systems — R. Stockton Gaines, Editor

Specification and Verification of the UCLA Unix† Security Kernel

Bruce J. Walker, Richard A. Kemmerer, and Gerald J. Popek
University of California, Los Angeles

Data Secure Unix, a kernel structured operating system, was constructed as part of an ongoing effort at UCLA to develop procedures by which operating systems can be produced and shown secure. Program verification methods were extensively applied as a constructive means of demonstrating security enforcement.

Here we report the specification and verification experience in producing a secure operating system. The work represents a significant attempt to verify a large-scale, production level software system, including all aspects from initial specification to verification of implemented code.

Key Words and Phrases: verification, security, operating systems, protection, programming methodology, ALPHARD, formal specifications, Unix, security kernel

CR Categories: 4.29, 4.35, 6.35

1. Introduction

Early attempts to make operating systems secure merely found and fixed flaws in existing systems. As these efforts failed, it became clear that piecemeal alterations were unlikely ever to succeed [20]. A more systematic method was required, presumably one that controlled the system's design and implementation. Then secure operation could be demonstrated in a stronger sense than an ingenuous claim that the last bug had been eliminated, particularly since production systems are rarely static, and errors easily introduced.

Our research seeks to develop means by which an operating system can be shown data secure, meaning that direct access to data must be possible only if the recorded protection policy permits it. The two major components of this task are: (1) developing system architectures that minimize the amount and complexity of software involved in both protection decisions and enforcement, by isolating them into *kernel* modules; and (2) applying extensive verification methods to that kernel software in order to prove that our of *data security* criterion is met. This paper reports on the latter part, the verification experience. Those interested in architectural issues should see [23]. Related work includes the PSOS operating system project at SRI [25] which uses the hierarchical design methodology described by Robinson and Levitt in [26], and efforts to prove communications software at the University of Texas [31].

Every verification step, from the development of top-level specifications to machine-aided proof of the Pascal code, was carried out. Although these steps were not completed for all portions of the kernel, most of the job was done for much of the kernel. The remainder is clearly more of the same. We therefore consider the project essentially complete. In this paper, as each verification step is discussed, an estimate of the completed portion of that step is given, together with an indication of the amount of work required for completion. One should realize that it is essential to carry the verification process through the steps of actual code-level proofs because most security flaws in real systems are found at this level [20]. Security flaws were found in our system during verification, despite the fact that the implementation was written carefully and tested extensively. An example of

> Our research seeks to develop means by which an operating system can be shown data secure, meaning that direct access to data must be possible only if the recorded protection policy permits it. The two major components

Communications of the ACM	February 1980 Volume 23 Number 2

seL4: The Dream Come True!

The world's **first** operating-system kernel with **provable** security enforcement

World's most advanced mixed-criticality OS

Open Source

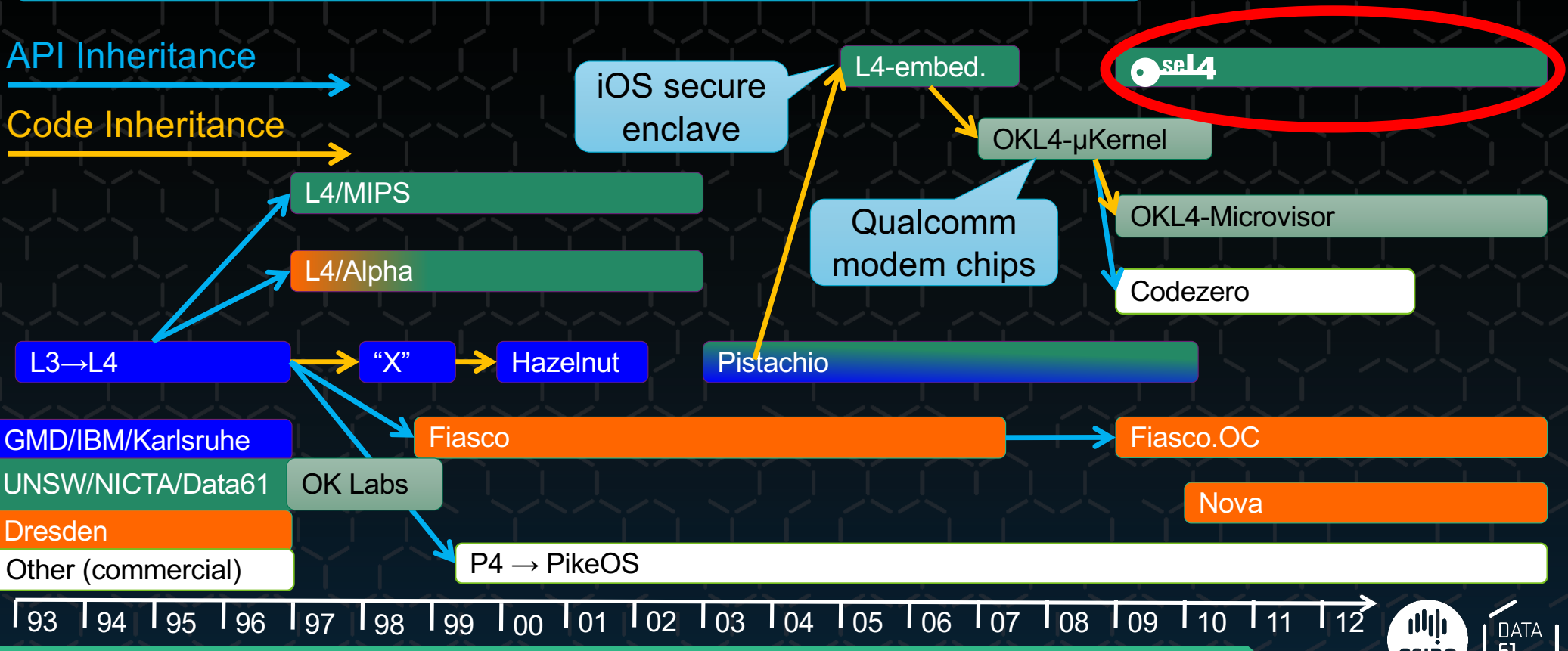
The world's **only** protected-mode OS with complete, sound timeliness analysis

The world's **fastest** microkernel, designed for **real-world** use

CSIRO DATA 61

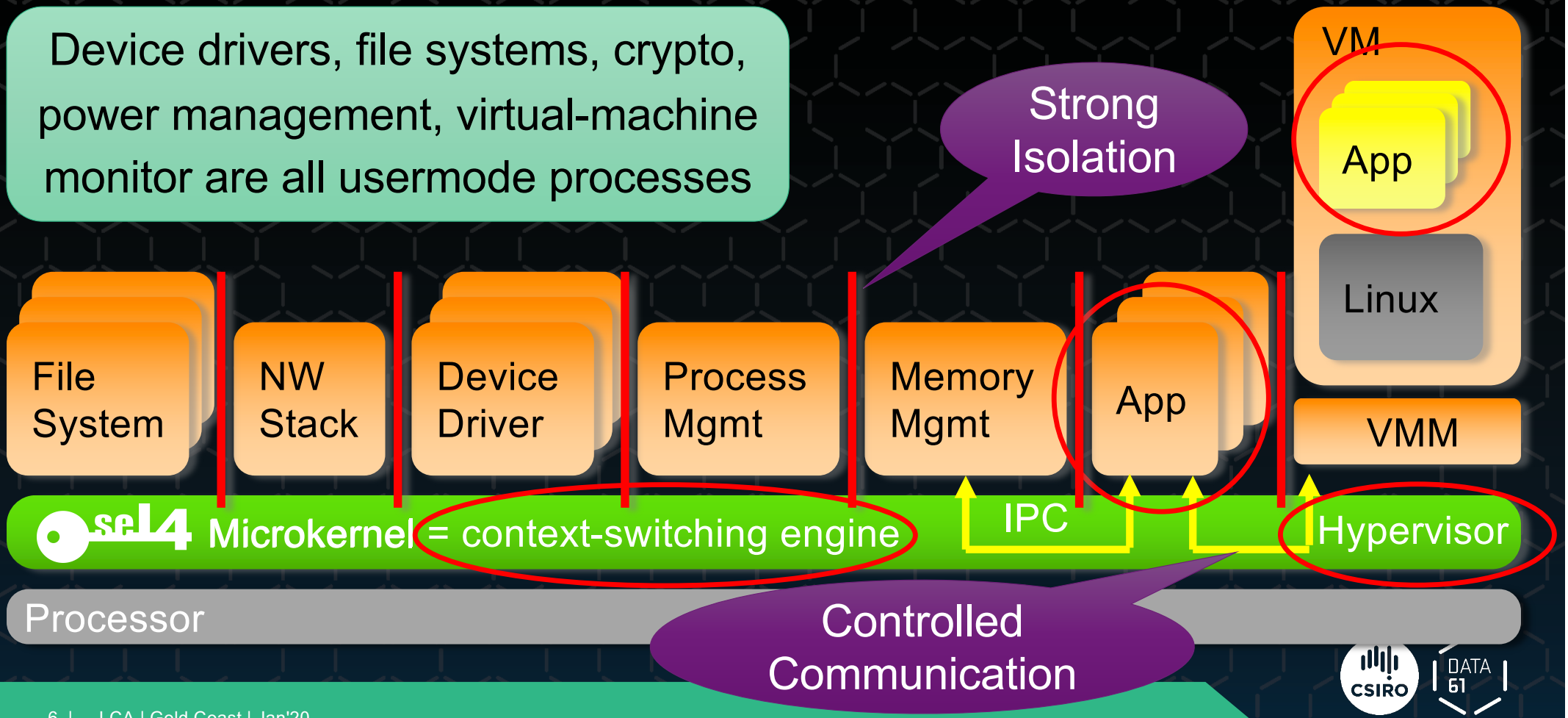
L4: 25 Years High-Performance Microkernels

seL4: The latest member of the L4 microkernel family



A Microkernel is not an OS

Device drivers, file systems, crypto, power management, virtual-machine monitor are all usermode processes



Core Mechanism: Object Capability

Capability = Access Token:
Prima-facie evidence of privilege

Eg. thread, address space

Obj reference

Access rights

Eg. read, write, send, execute…

Capabilities provide:

- Fine-grained access control
- Reasoning about information flow

CSIRO DATA 61

Microkernel: seL4 vs Linux Licensing

Valuable IP

Your code (applications)

any…

Your system services

any…

TS system services:
libs, drivers, file systems,
NW stacks, tools, …

BSD

Platform port:
- timer
- serial
- HW init

GPL v2

Boiler plate

TS kernel source

GPL v2

Your code (applications)

any…

kernel.org source
- core kernel
- device drivers
- file systems
- NW stacks
- …

GPL v2

Your system services

GPL v2

Platform port:
- device drivers
- HW init

GPL v2

Boiler plate

Details: https://microkerneldude.wordpress.com

CSIRO DATA 61

Military-Strength Security

Unmanned Little Bird (ULB)

DARPA HACMS: Retrofit existing system!

Autonomous trucks

Secure Comms Dongle

Cross-Domain Desktop Compositor

Verification

CSIRO DATA 61

World's Most Secure OS: Arm v7

Confidentiality

Integrity

Availability

Proof

Proof

Proof

Model enforces security

Abstract Model

Functional correctness: C code only behaves as specified

Proof

Translation validation: Binary retains C-code semantics

C Imple-mentation

Proof

Sound worst-case execution time bound

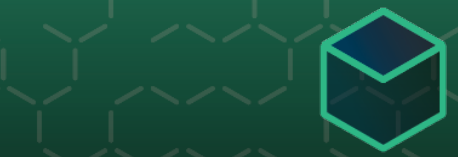
Binary code

Limitations (work in progress):

- Kernel initialisation not yet verified
- MMU & caches modelled abstractly
- Timing channels not ruled out

CSIRO DATA 61

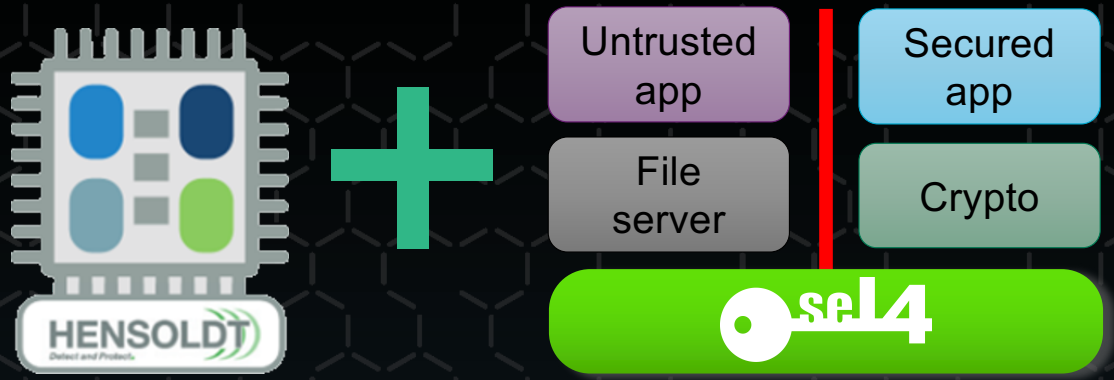
seL4 on RISC-V



CSIRO DATA 61

Background: HENSOLD Cyber

Disclosure: I have an interest in HENSOLDT Cyber



Munich-based startup

- Secure RISC-V processor
- Based on open-source Ariane
- Supply chain secured through logic encryption
- Secure OS based on seL4
- Targets defence, industrial control, critint, automotive

Performance on RV64

Message-passing round-trip latency in cycles

Not yet fully optimised!

Arch	x86 32b	x86 64b	Arm 32b	Arm 64b	RISC-V 64b
Intra address space	427	565	625	752	690
Inter address space	752	1041	625	752	1006

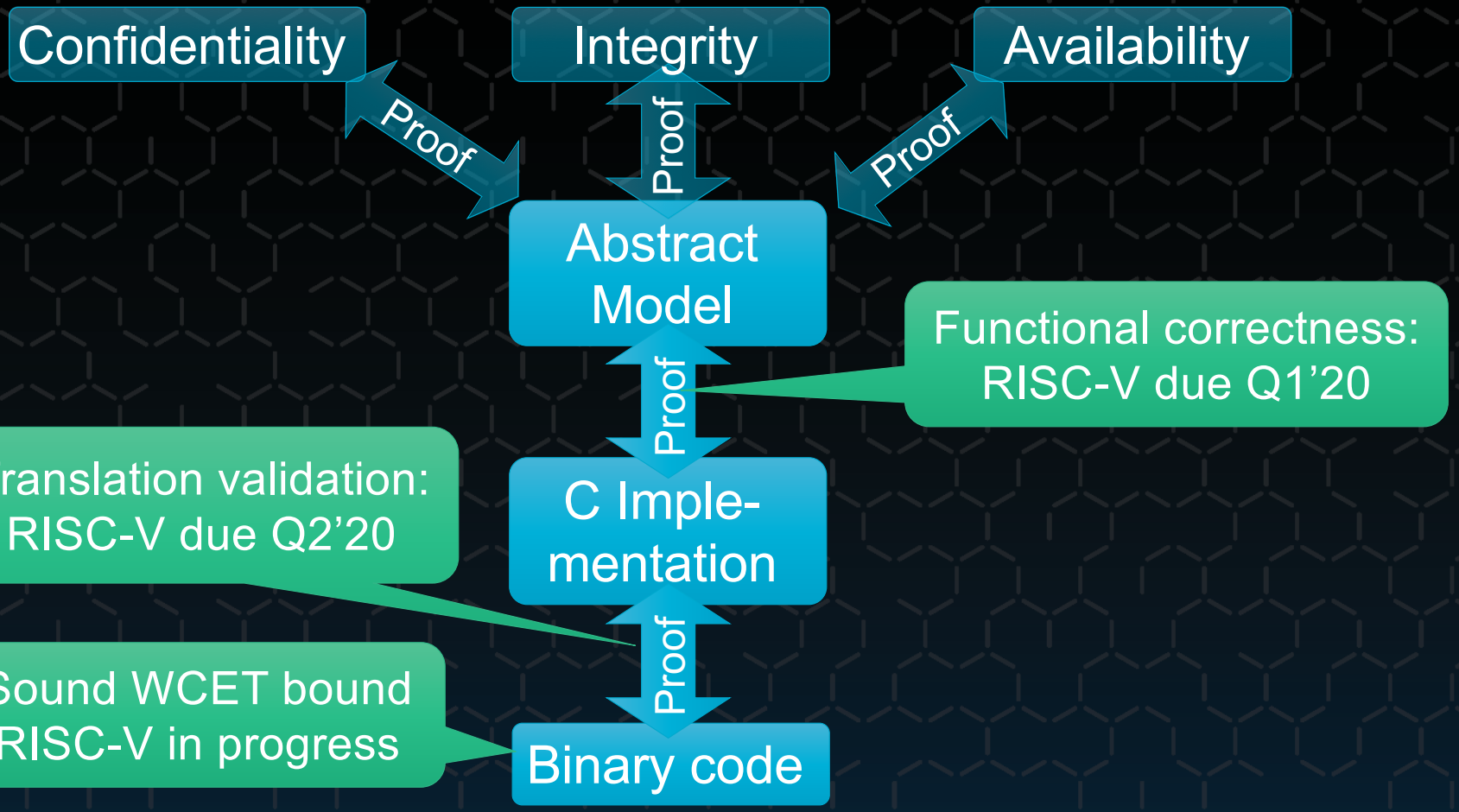
Spectre-workaround disabled (else much more expensive)

No ASIDS on **HiFive Unleashed**, else inter-AS would be same as intra-AS

Hypervisor extensions (draft spec 0.5) supported in branch

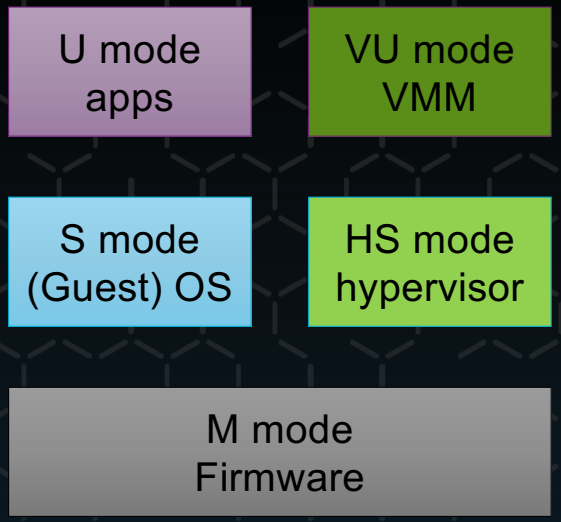
CSIRO DATA 61

Verification: RISC-V Status



Experience with RISC-V Architecture

- Kernel port straightforward:
  - simple and clean RISC architecture
- Verification benefitted from cleanness
  - … but some challenges from less typing in page tables
- Hypervisor (draft) extensions even simpler
- M (machine) mode makes firmware explicit
  - configures HW, delegates to S (supervisor) mode
  - emulates features not implemented in HW
  - should be verified
- Extensibility of ISA could be a concern
  - could undermine portability
- Formal ISA spec is great!



CSIRO DATA 61

LCA'18 Refresher: Time Capabilities

Classical thread attributes

- Priority
- Time slice

Not runnable if null

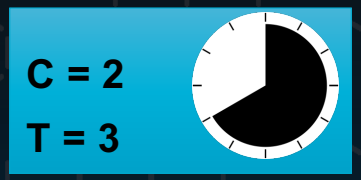
New thread attributes

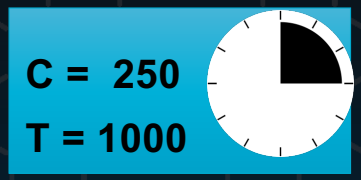
- Priority
- Scheduling context capability

Capability for time

Limits CPU access!

Scheduling context object
- T: period
- C: budget (≤ T)

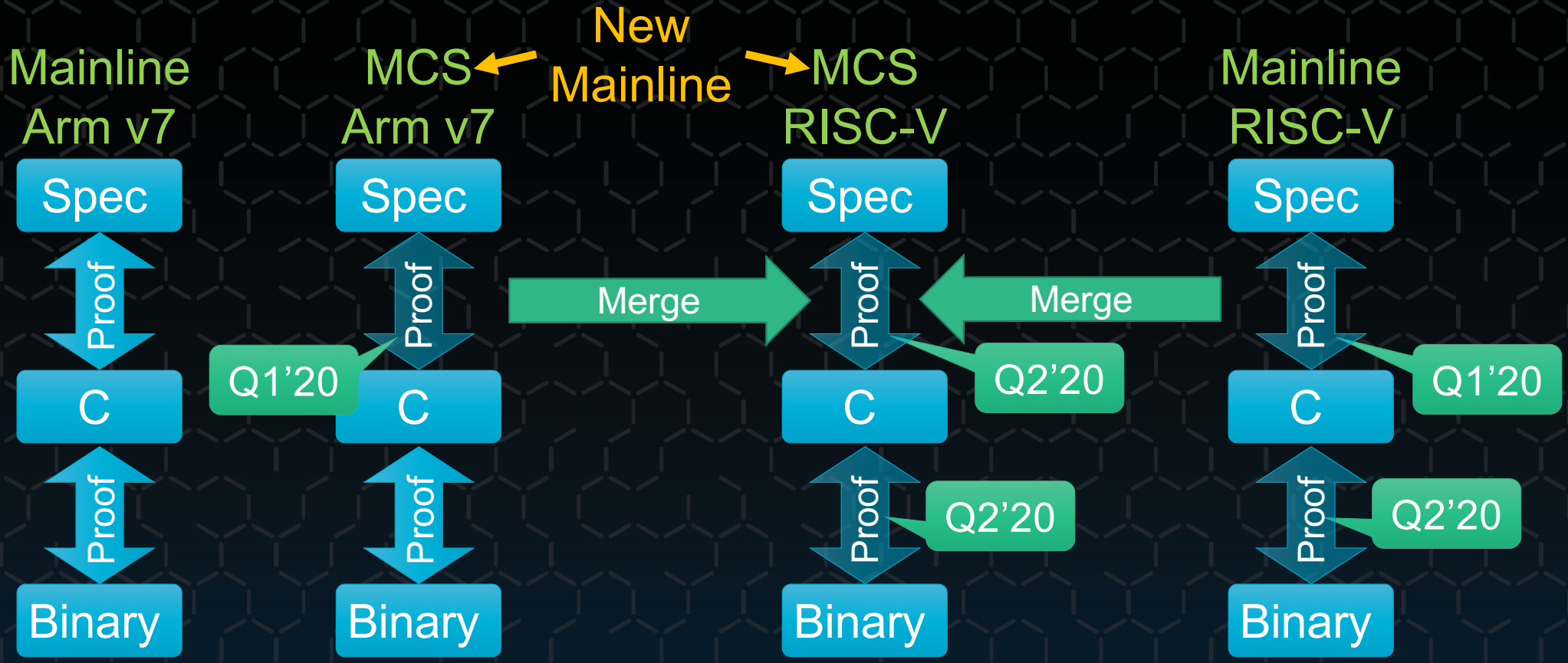




Enables reasoning about time and temporal isolation for mixed-criticality systems

CSIRO DATA 61

Time Caps (MCS) Kernel Verification



Research: Time Protection

CSIRO DATA 61

Threats

=

Speculation

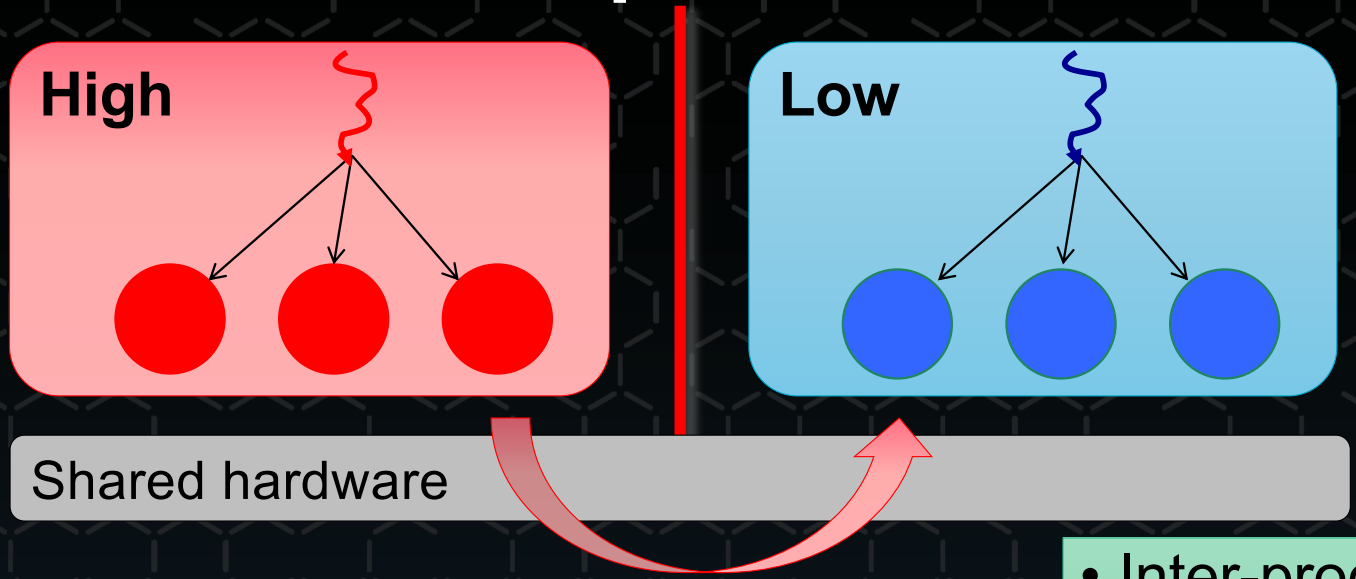
An “unknown unknown” until recently

+

A “known unknown” for decades

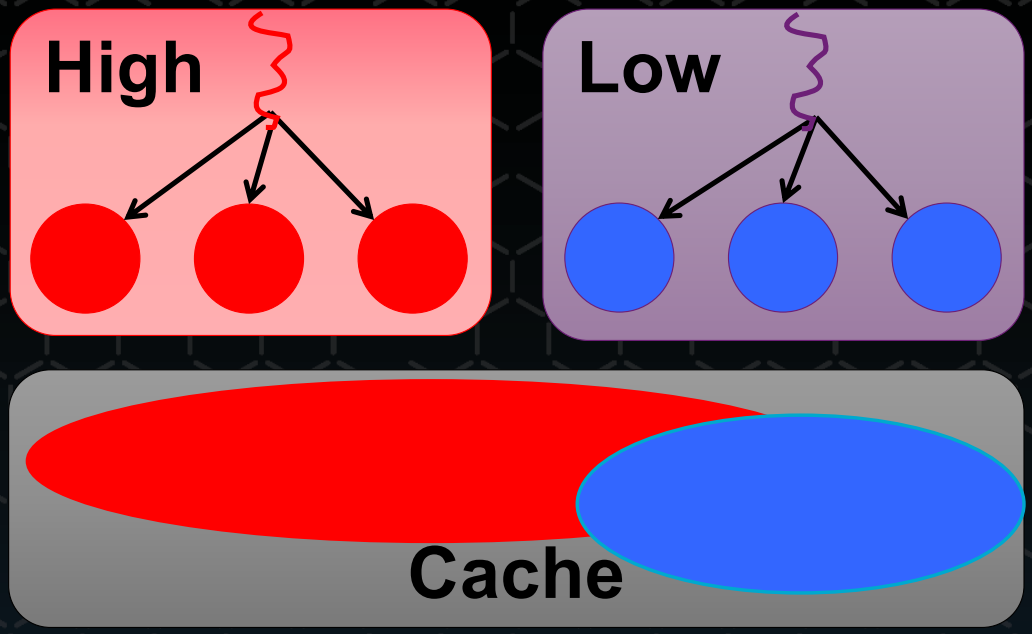
Microarchitectural Timing Channel

Cause: Competition for HW Resources



- Inter-process interference
- Competing access to micro-architectural features
- Hidden by the HW-SW contract!

Sharing: Stateful Hardware



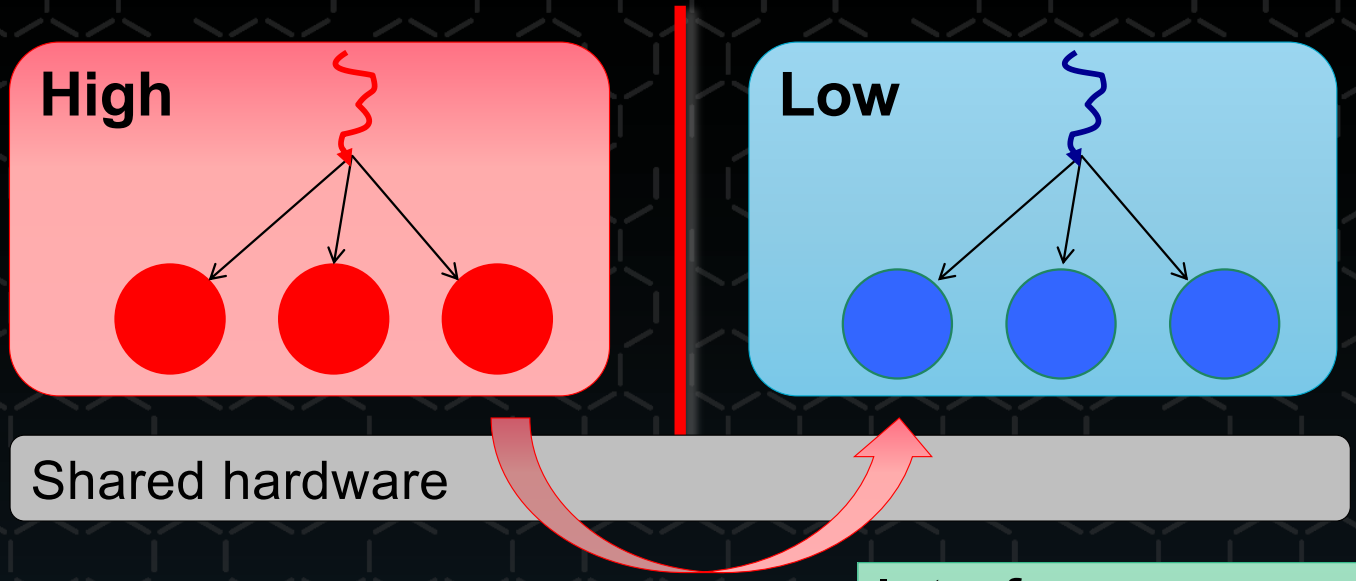
HW is *capacity-limited*

- Interference during
  - concurrent access
  - time-shared access
- Collisions reveal addresses
- *Usable as side channel*

Any state-holding microarchitectural feature:

- cache, branch predictor, pre-fetcher state machine

Time Protection: Prevent Interference

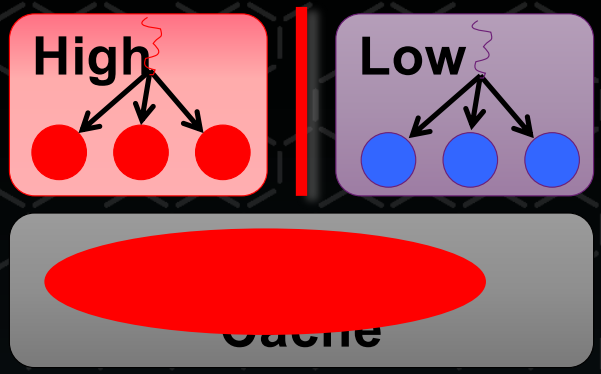


Affect execution speed

Interference results from sharing
⇒ Partition hardware:

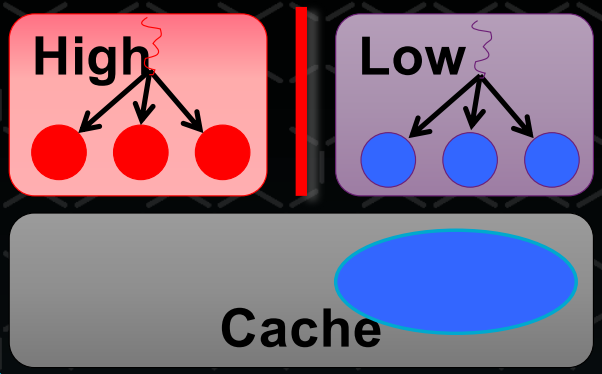
- spatially
- temporally (time shared)

Time Protection: Partition Hardware

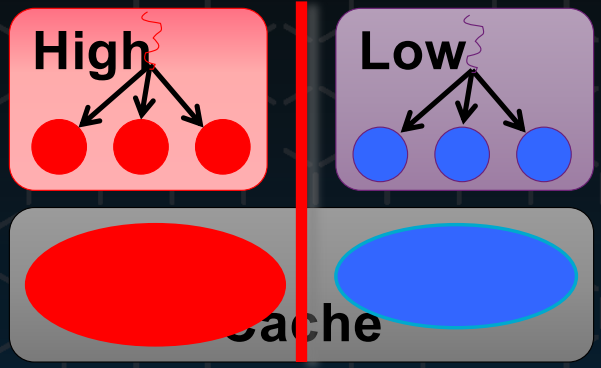


Temporally partition

Flush



Spatially partition



Need both!

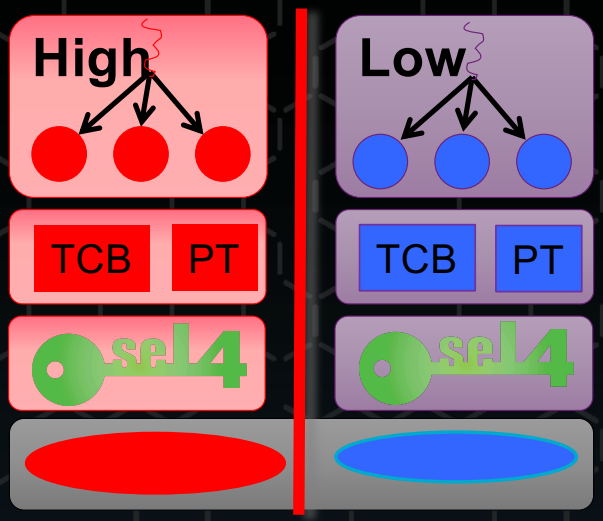
Cannot spatially partition on-core caches (L1, TLB, branch predictor, pre-fetchers)

- virtually-indexed
- OS cannot control

Flushing useless for concurrent access

- HW threads
- cores

Spatially Partition: Cache Colouring



- Partitions get frames of disjoint colours
- seL4: userland supplies kernel memory ⇒ colouring userland colours dynamic kernel memory
- Per-partition kernel image to colour kernel

[Ge et al. EuroSys'19]

Temporal Partitioning: Flush on Switch

Must remove any history dependence!

1. T_0 = current_time()
2. Switch user context
3. Flush on-core state
4. Touch all shared data needed for return
5. while (T_0+WCET < current_time()) ;
6. Reprogram timer
7. return

Latency depends on prior execution!

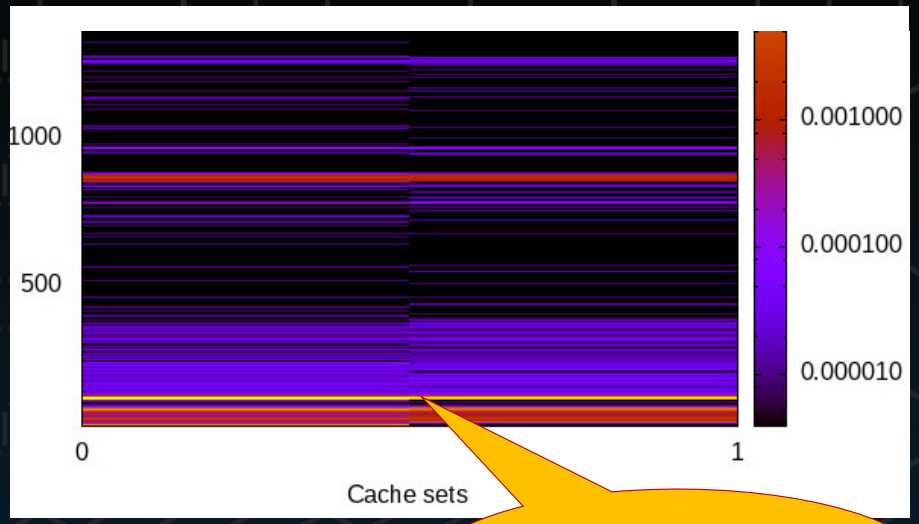
Time padding to Remove dependency

Ensure deterministic execution

Challenge: Broken Hardware

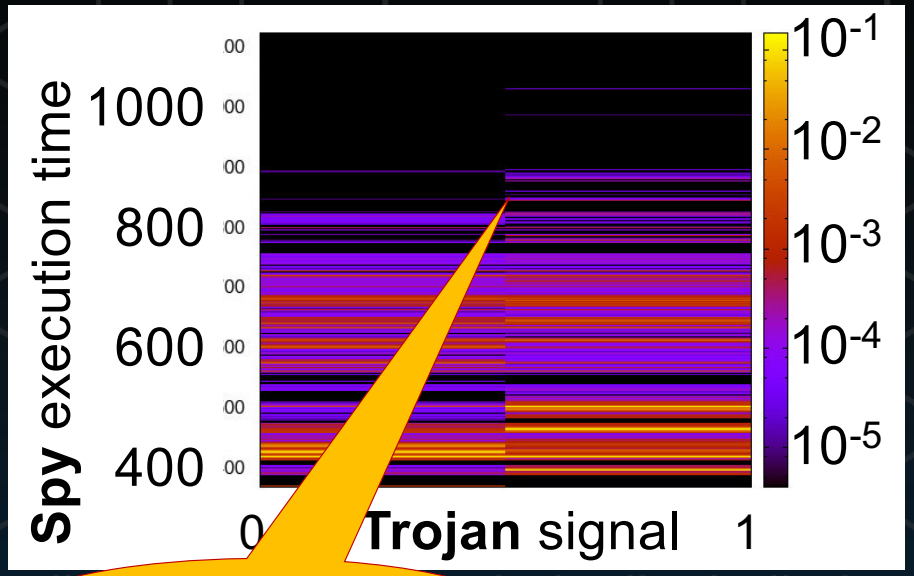
- Systematic study of COTS hardware (Intel and Arm) [Ge et al, APSys'18]:
  - contemporary processors hold state that cannot be reset

Intel branch history buffer



Small channel!

HiSilicon A53 branch history buffer



Channel!

Way Out: New HW-SW Contract!

ISA is purely functional contract, abstracts too much away

New contract (augmented ISA):

All shared HW resources must be spatially or temporally partitionable by OS

[Ge et al, APSys'18]

RISC-V to the rescue:
Strong commitment to making it happen!

Community/ Ecosystem

CSIRO DATA 61

Experience with RISC-V Foundation

Security Standing Committee

- Invited me on
- Very receptive and supportive
- Committed to making RISC-V "most secure architcture"
- Facilitated engagement with Privspec TC (now Standing Committee)

Privileged Spec Tech Committee

- Hypervisor-extension feedback well received
  - Easy engagement
  - Constructive proposal from TC chair addressing our issues
- Time-protection slow to get traction
  - Now good engagement, hopefully progress soon

- Open but skeptical
- They need to manage conflicting ideas
- Keen to get "most secure arch" recognition

CSIRO DATA 61

We Are Creating the seL4 Foundation!

Aims:

- Provide a neutral entity for coordinating & enhancing seL4 ecosystem
- Grow adoption of seL4
- Improve (organisational and individual) community participation & cooperation
  - Developers
  - Adopters
- Develop / standardise seL4 system
  - kernel & proofs
  - libraries, services, tools
- Protect and promote the seL4 brand
  - prevent reputational damage from using modified seL4 (verification invalidated)
- Provide platform for pooling funds for critical “big-ticket” items (verification)

CSIRO DATA 61

Foundation Structure

seL4 Foundation

seL4 Board

seL4 Fund Charter

seL4 Directed Fund $$

LF Projects LLC

seL4 Series LLC

seL4 TM

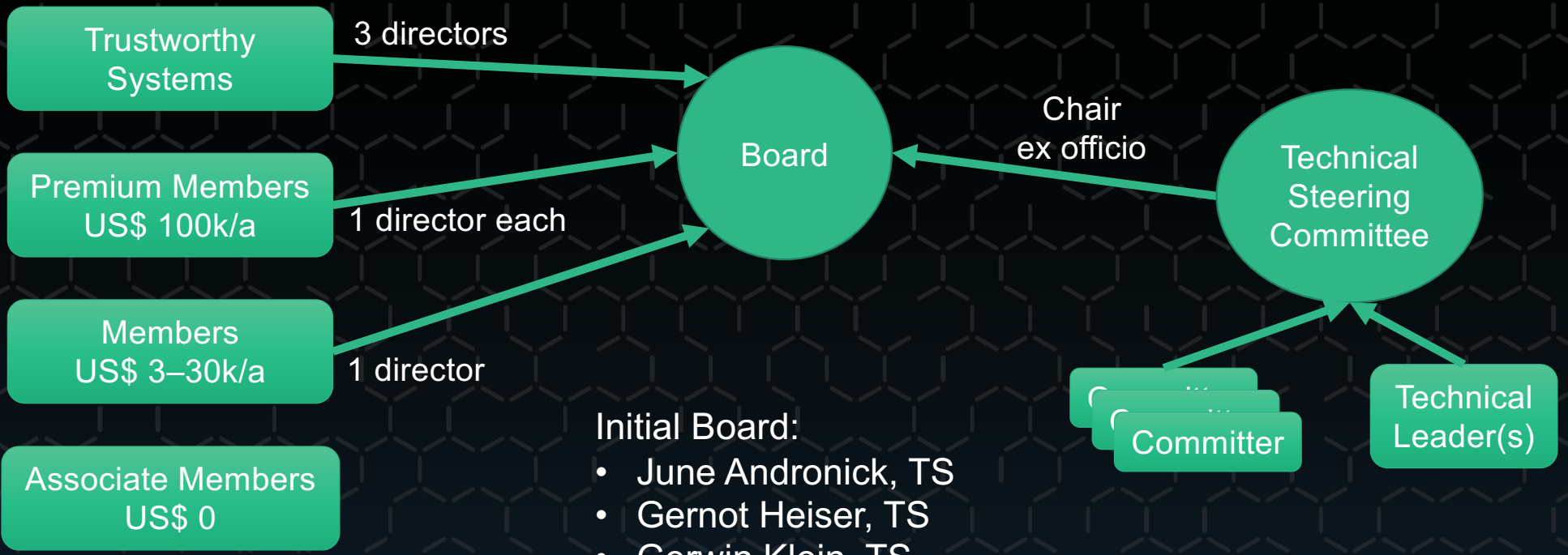
https://sel4.systems

seL4 Technical Charter

Technical Project

Contributor

Membership (Subject to Minor Change)

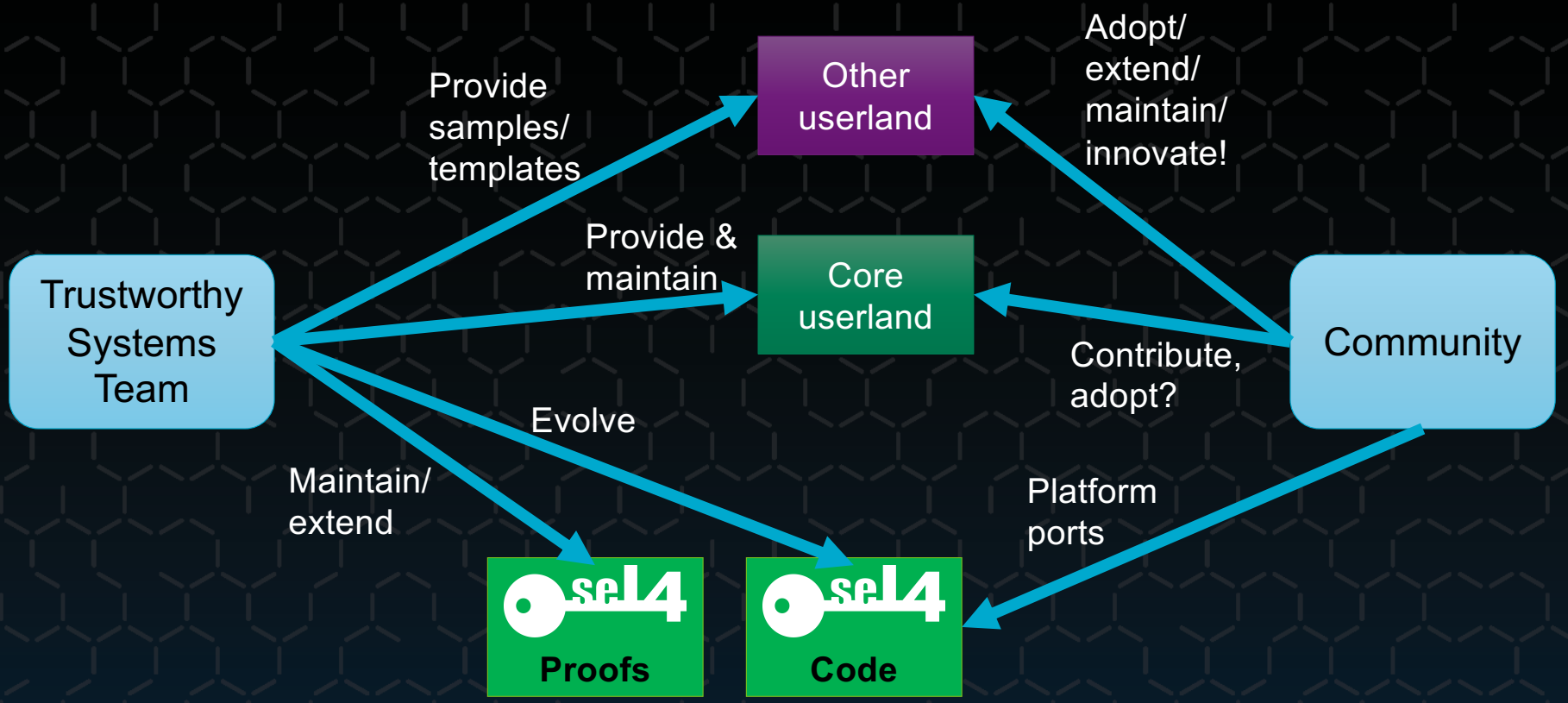


Note: members must be financial members of Linux Foundation!

Initial Board:

- June Andronick, TS
- Gernot Heiser, TS
- Gerwin Klein, TS
- John Launchbury, Galois (ex DARPA)
- Sascha Kegreiß, HENSOLDT Cyber
- Daniel Potts, Ghost Locomotion

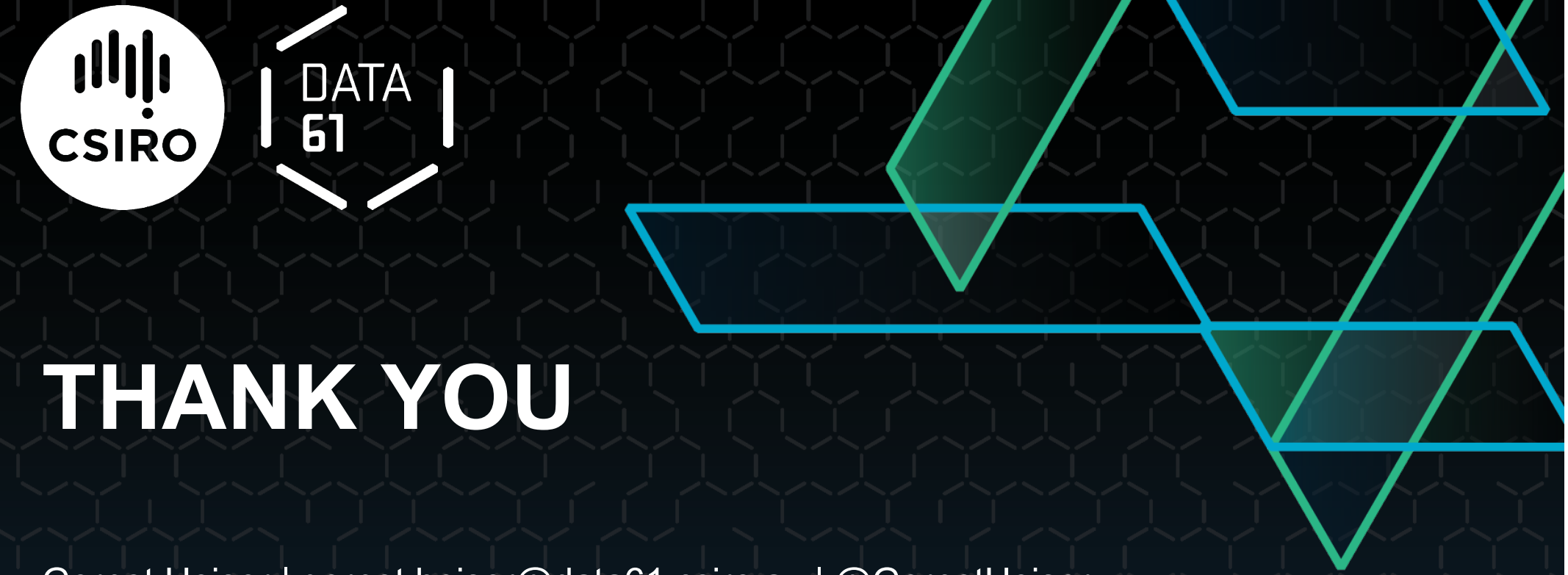
Community Engagement



Foundation Status

- Legal docs (fund charter & technical charter) submitted to Linux Foundation
  - just received their feedback
- Trademark ready for transfer to Foundation
- Initial board appointed
- Interim web site shows structure and “Principles” document
  - legal docs will be there once approved by LF
- Hopefully days away from being able to set up members
  - Mail foundation@sel4.systems if you’re interested in joining!

https://sel4.systems/Foundation



THANK YOU

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser

- LCA'20, Gold Coast, QLD, 2020-01-15

https://trustworthy.systems