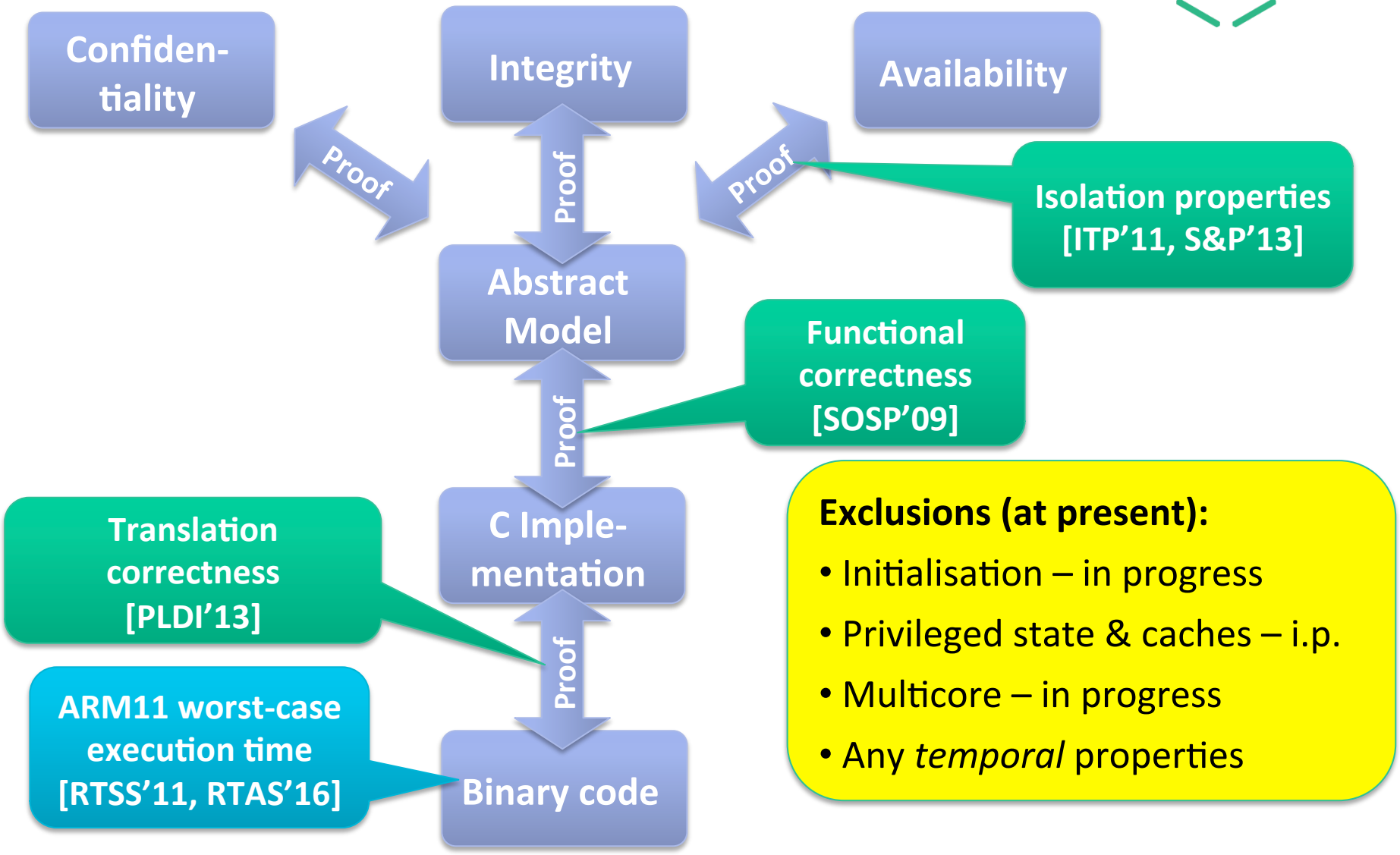
Security vs Manufacturers
A Losing Battle

Gernot Heiser
gernot.heiser@data61.csiro.au | @GernotHeiser
IFIP WG10.4, January 2017

https://trustworthy.systems

CSIRO

Spatial Isolation: Solved



Exclusions (at present):

- Initialisation – in progress
- Privileged state & caches – i.p.
- Multicore – in progress
- Any *temporal* properties

Critical Systems: DARPA HACMS

Boeing Unmanned Little Bird

Retrofit existing system!

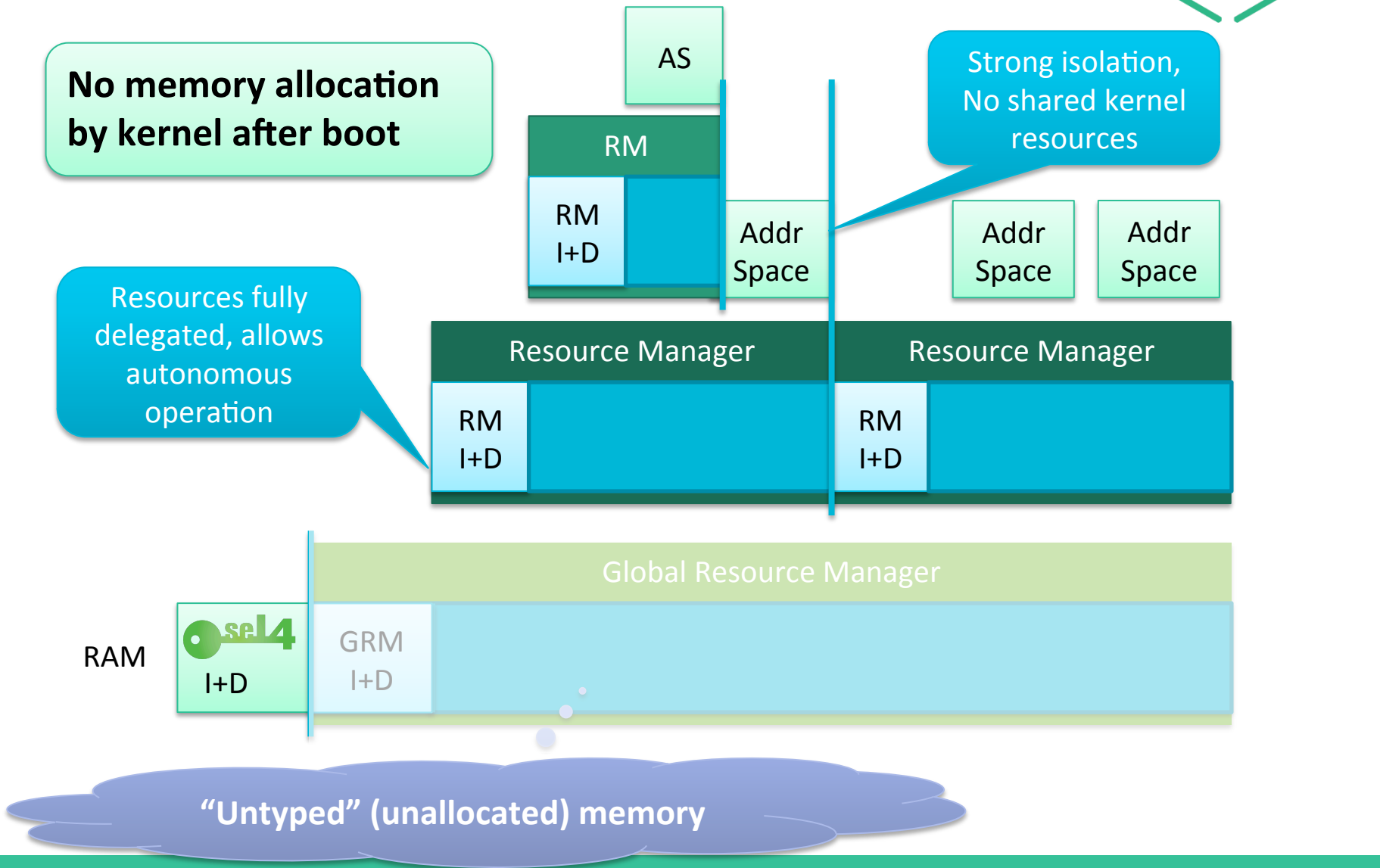
US Army Autonomous Trucks

SMACCMcopter Research Vehicle

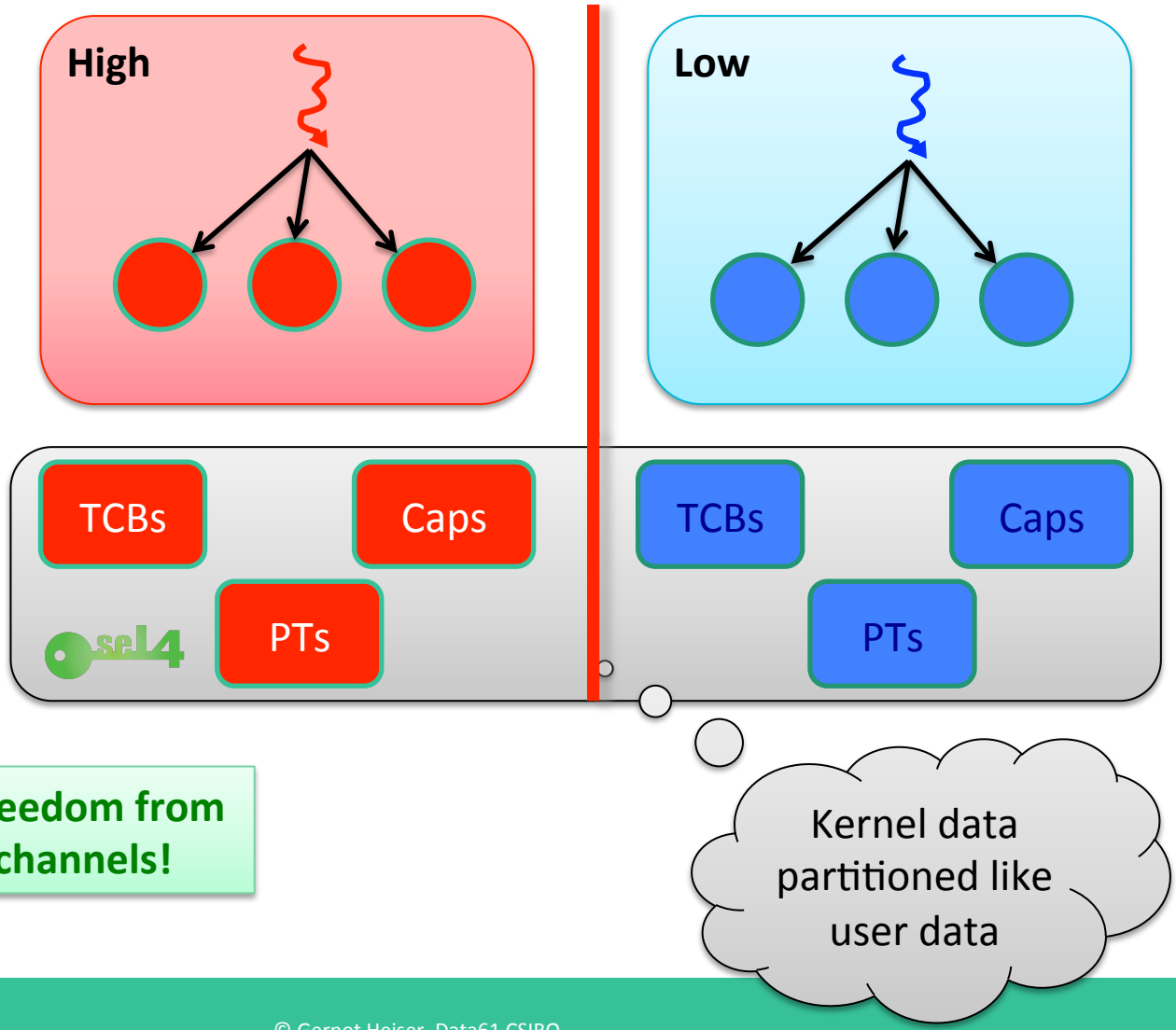
Develop technology

TARDEC GVRbot

Design for Isolation



Isolation Goes Deep



Provable freedom from storage channels!

Time – The Final Frontier

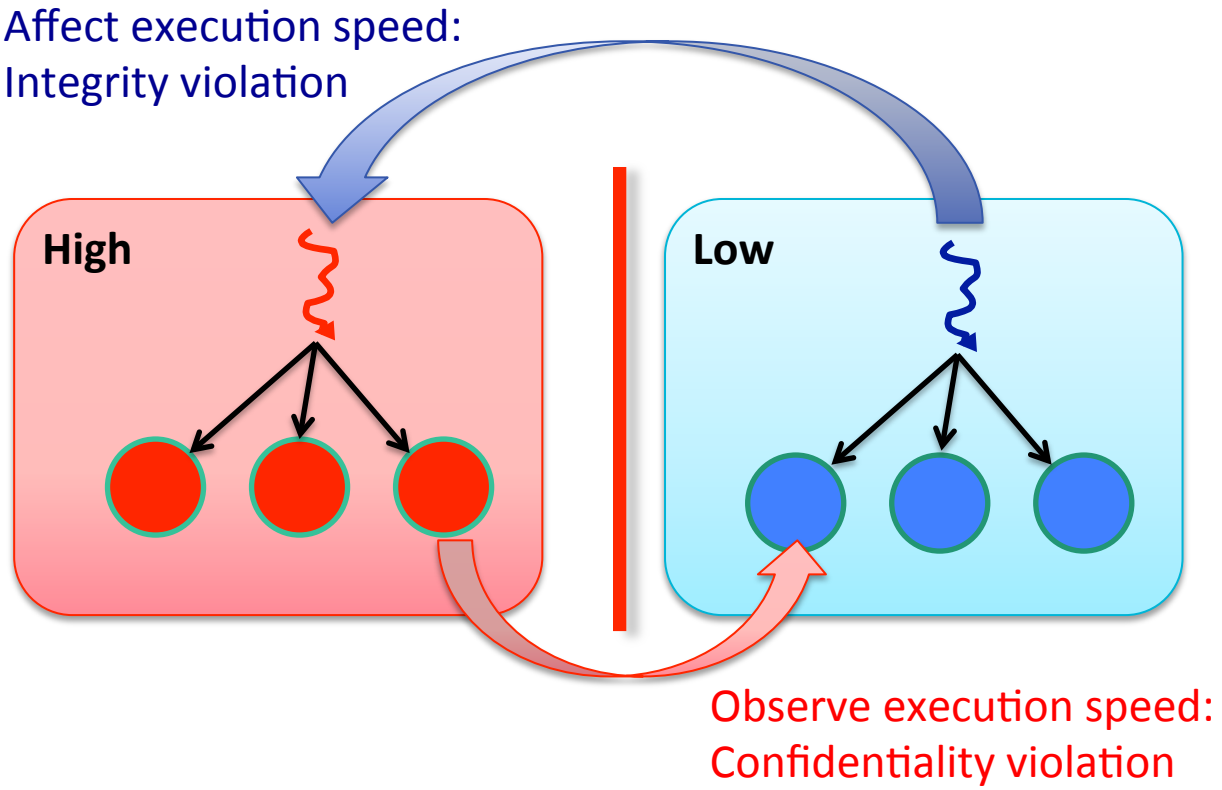
Two Aspects of Temporal Isolation

Safety: Timeliness

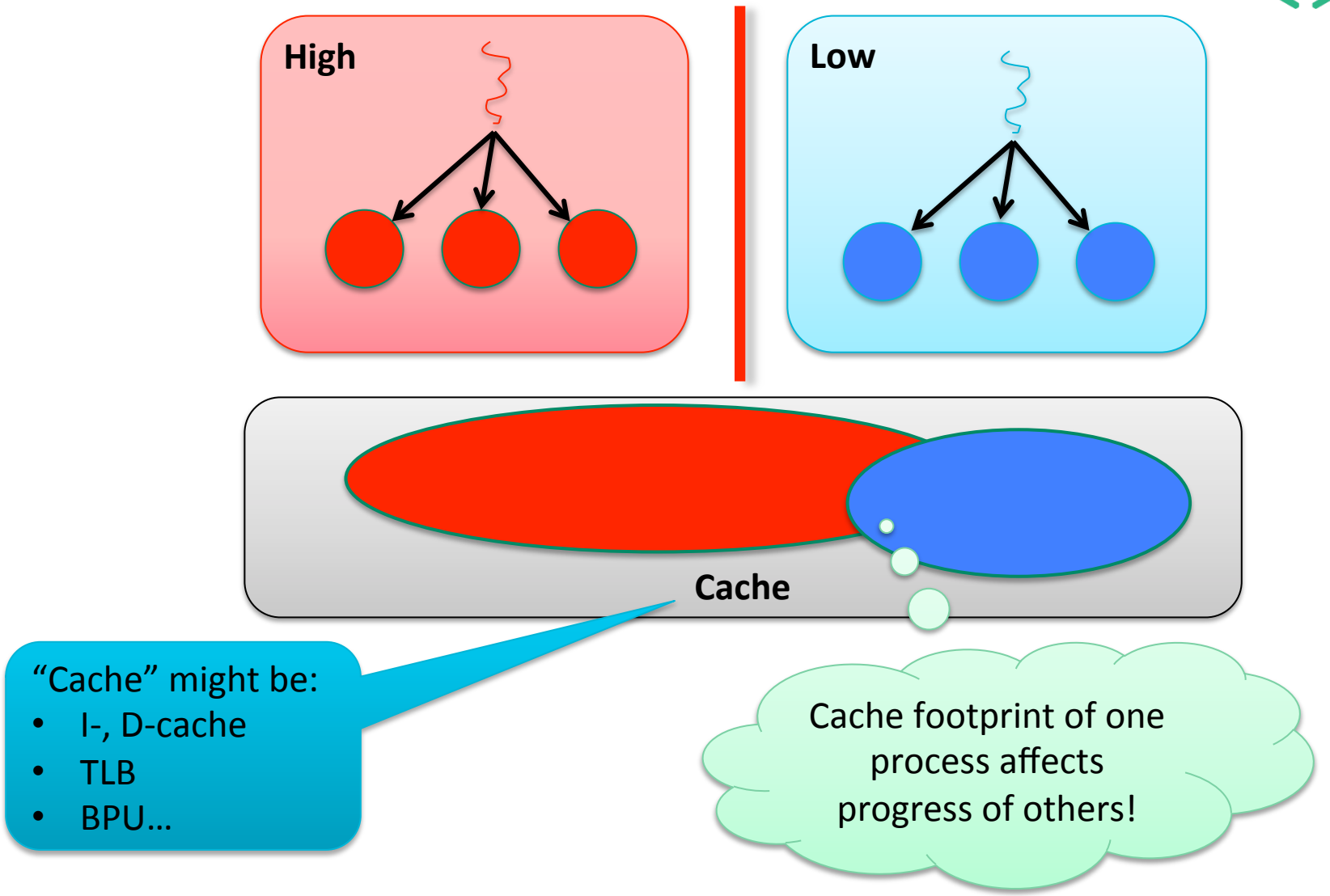
- *Bound* execution interference

Security: Confidentiality

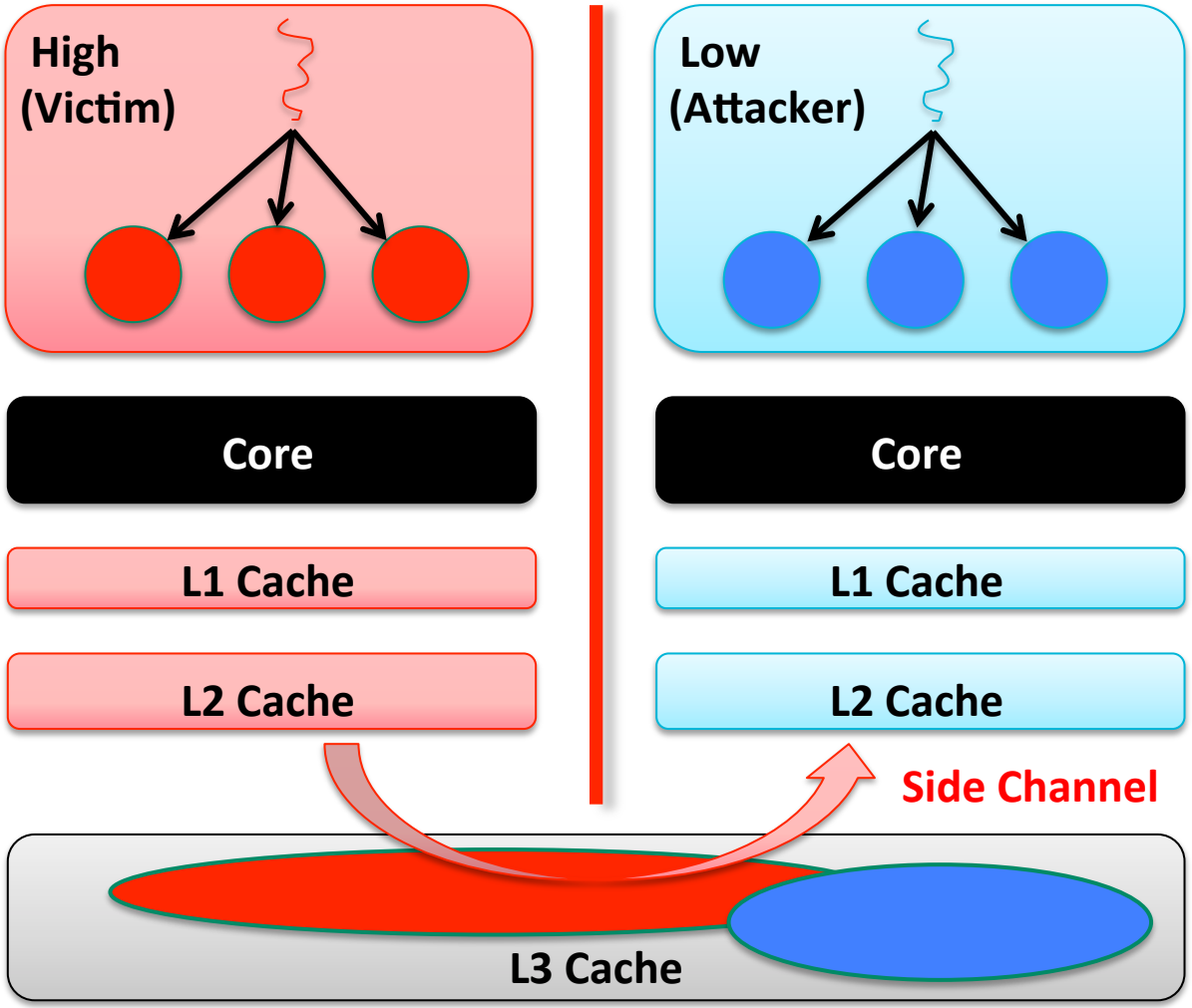
- *Prevent* leakage via timing channels



Timing Channels

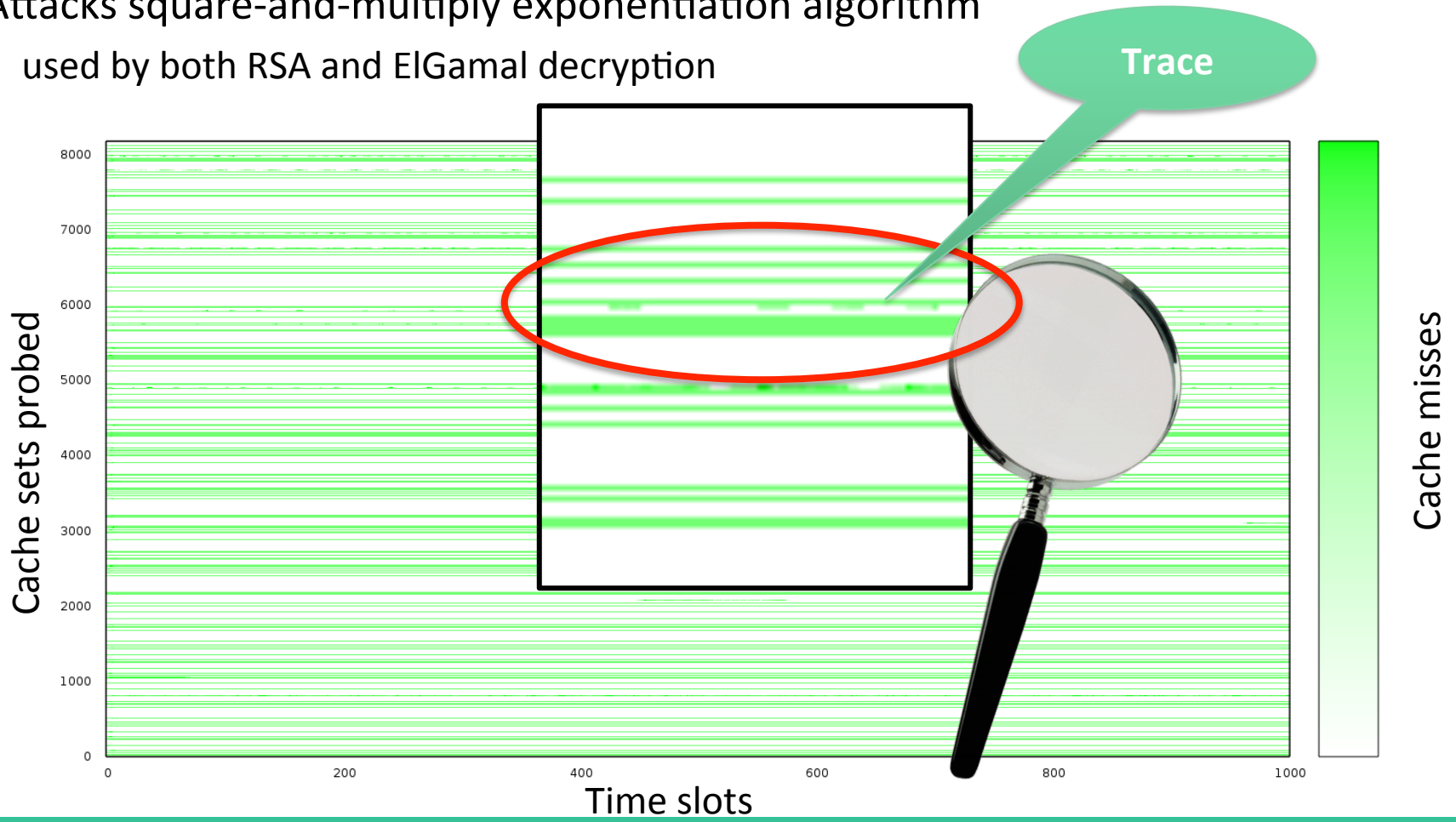


Cloud Scenario: Cross-Core LLC Attack
Side-channel attack through last-level cache

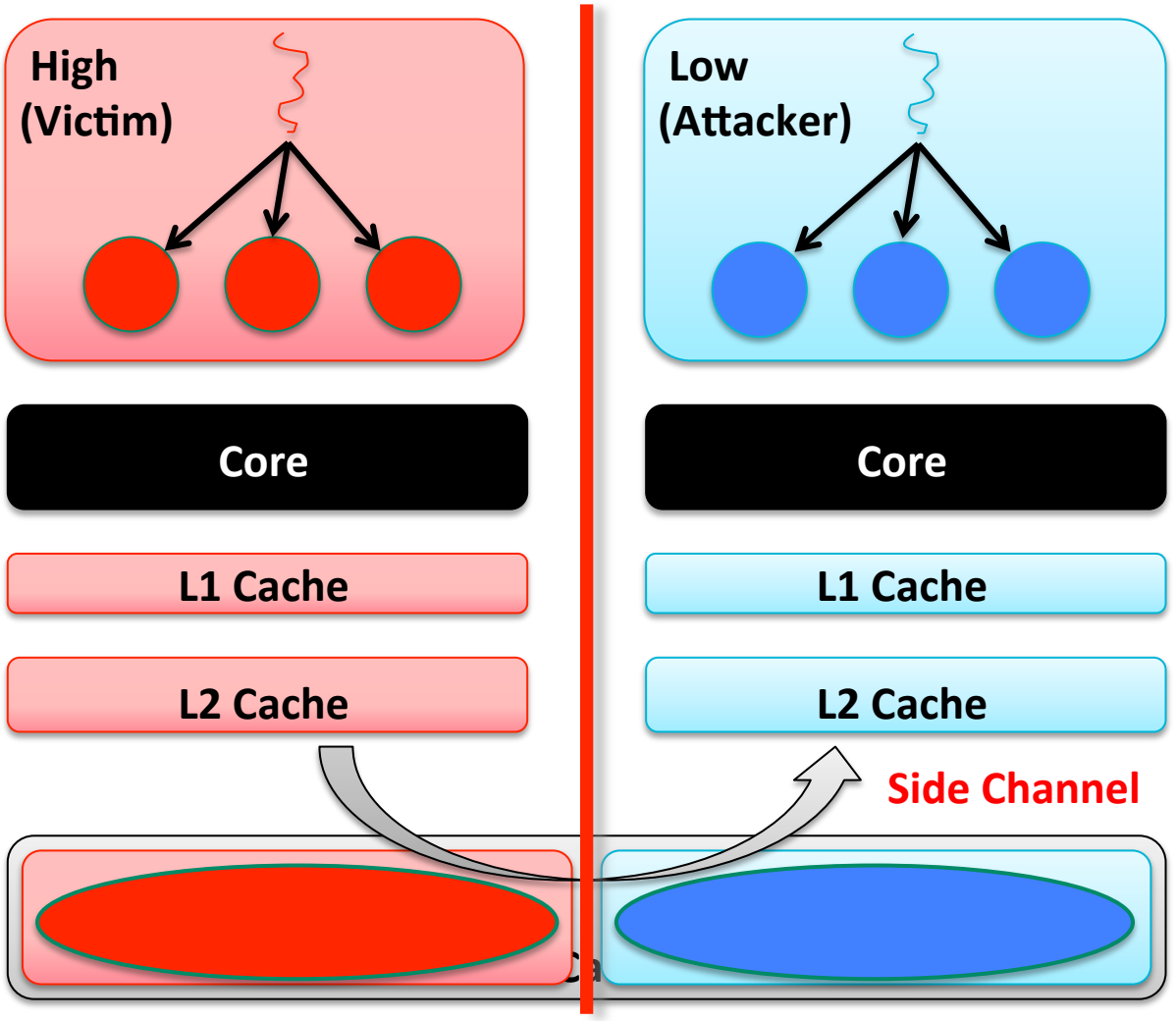


Cross-Core LLC Timing Side Channel [Liu et al, Oakland'15]

- Prime + probe technique, cross-core, cross-VM
- Attacks square-and-multiply exponentiation algorithm
  - used by both RSA and ElGamal decryption

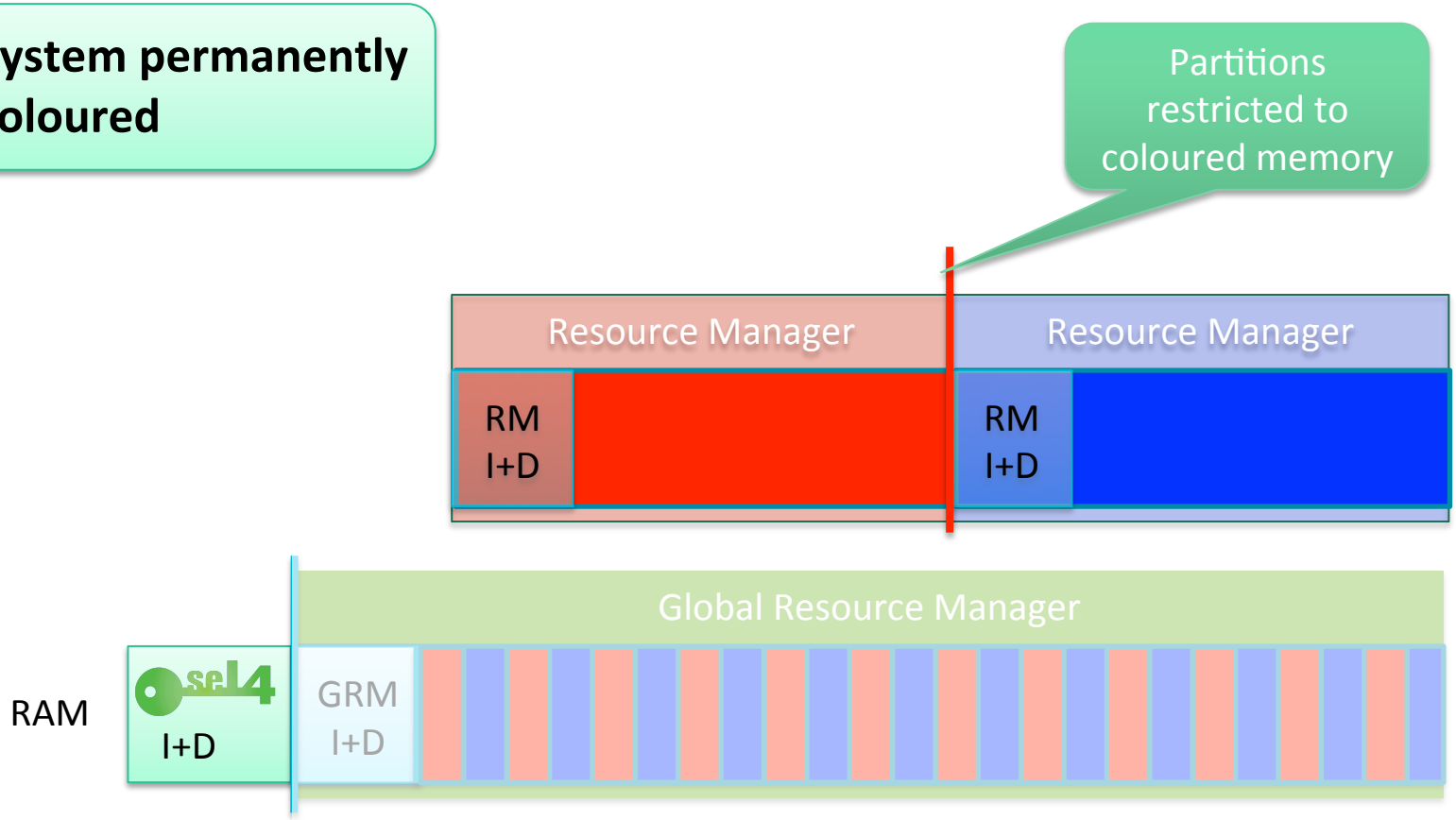


Mitigation: Partition Cache (Colouring)



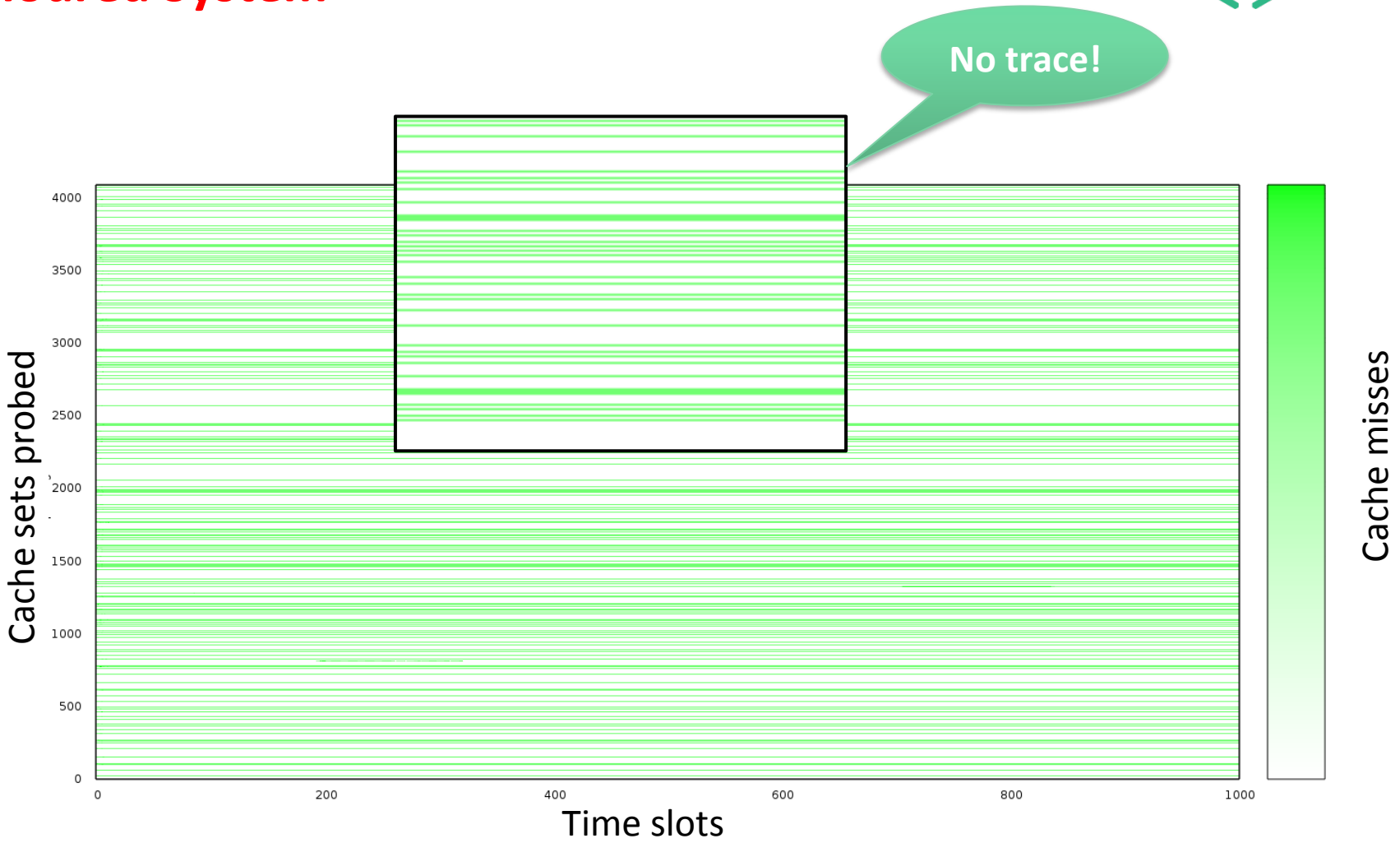
Colouring the System is Easy

System permanently coloured



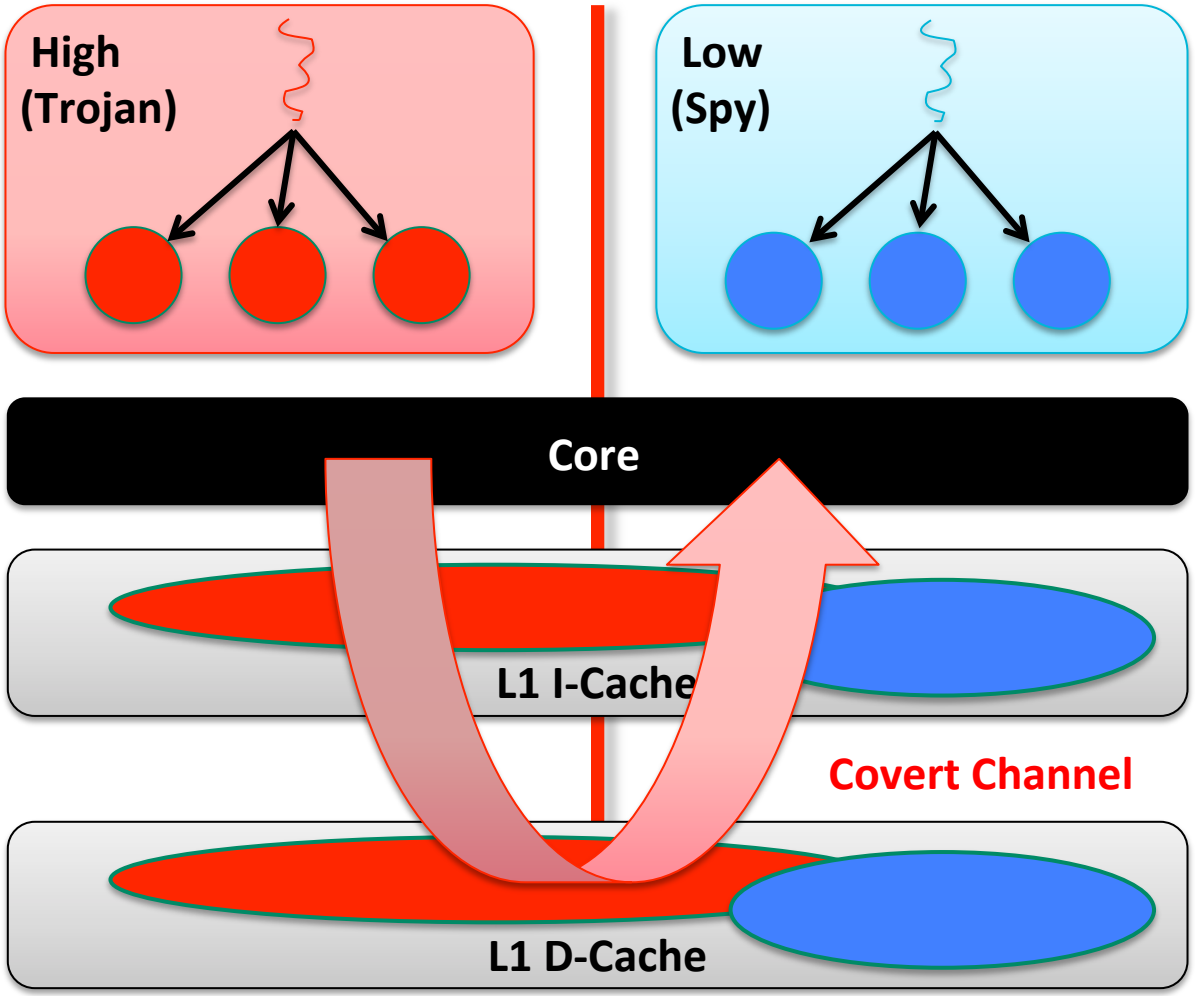
LLC Timing Side Channel Attack on seL4

Coloured System

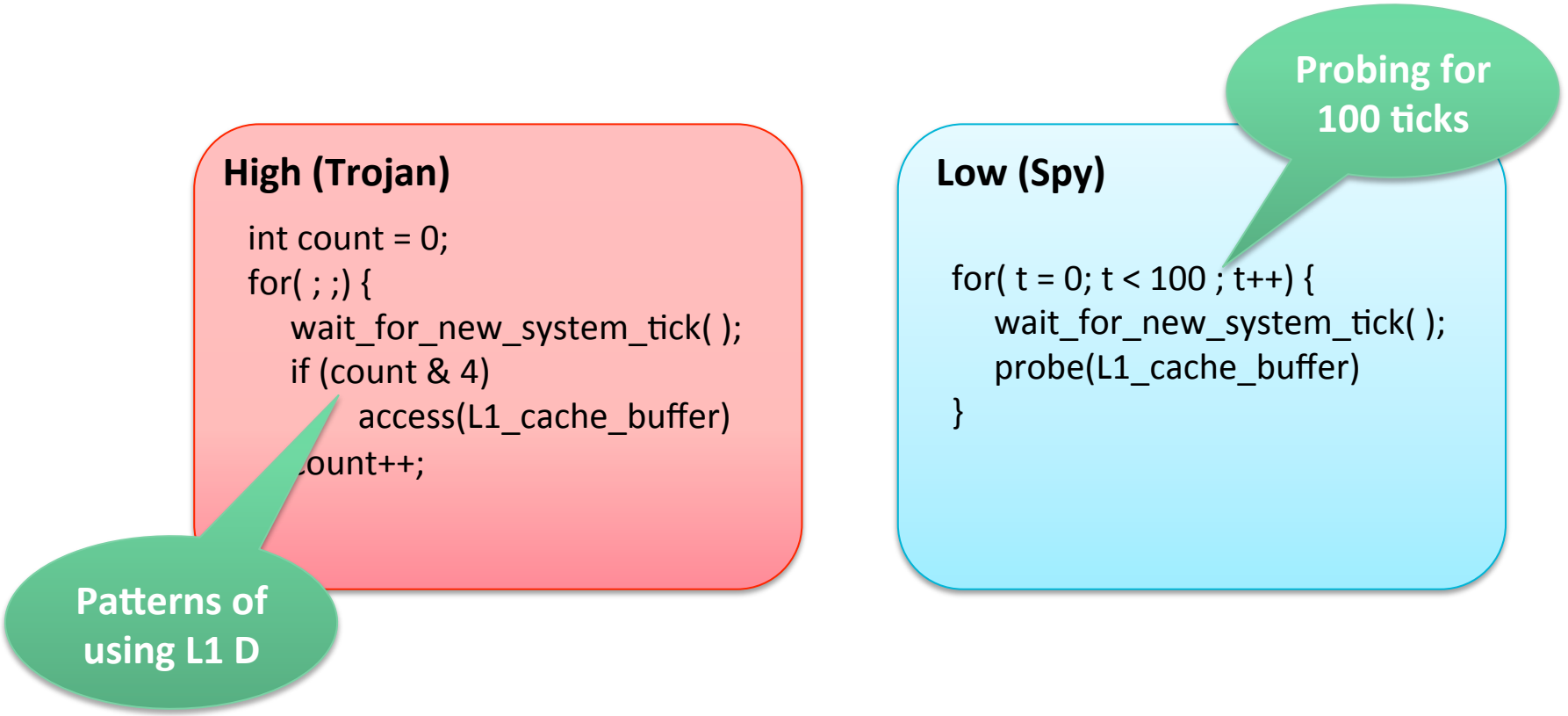


Covert-Channel Scenario: Intra-Core L1 Attack

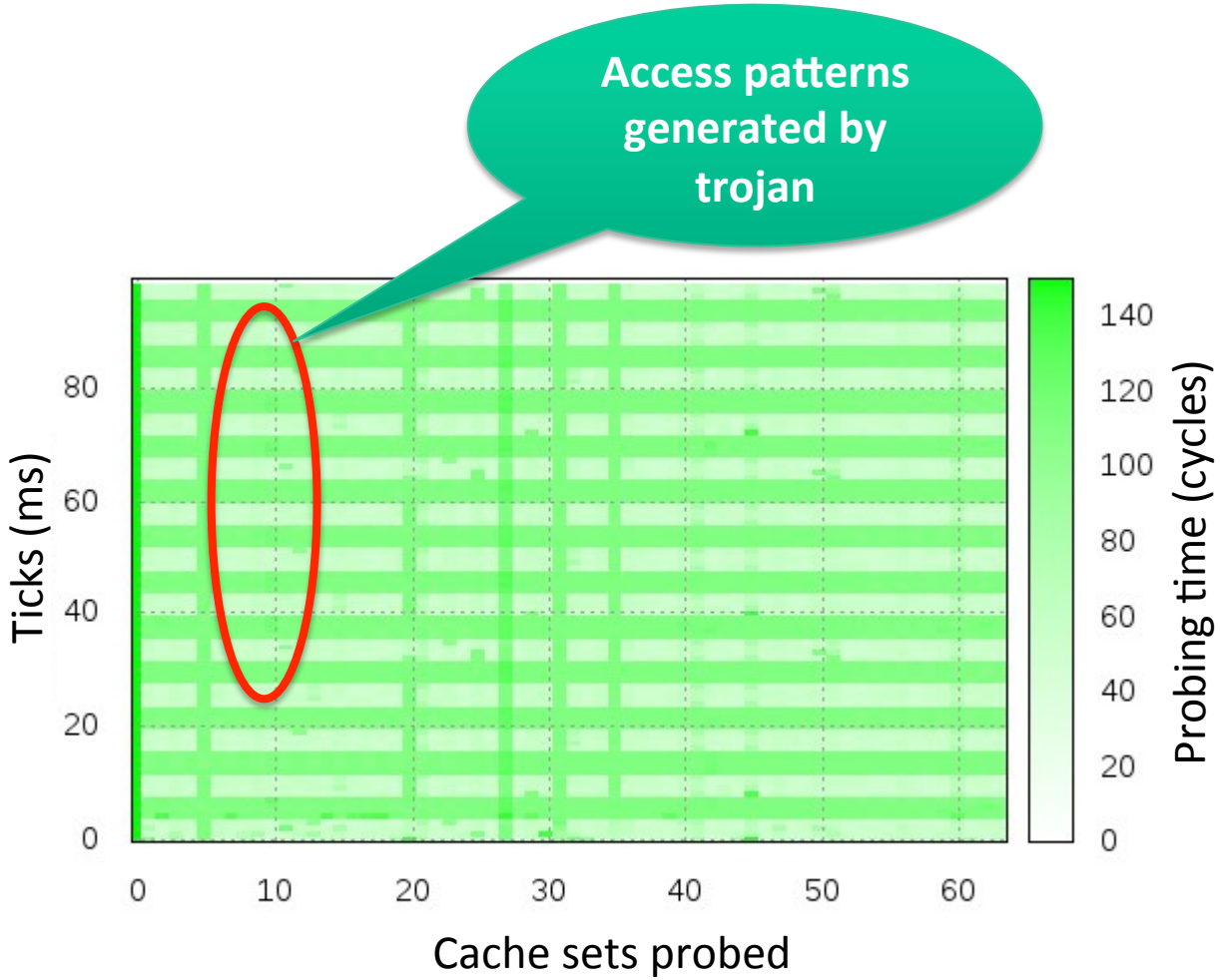
Covert-channel attack on time-shared core



L1 D-Cache Covert Channel



L1-D Cache Covert Channel
Spy observations on ARM A9



Challenge: L1 Cache Cannot Be Coloured

- L1 is virtually addressed ⇒ layout is not under OS control
- On most processors, L1 is too small (single colour)
- Even if it could be partitioned, performance cost would be high

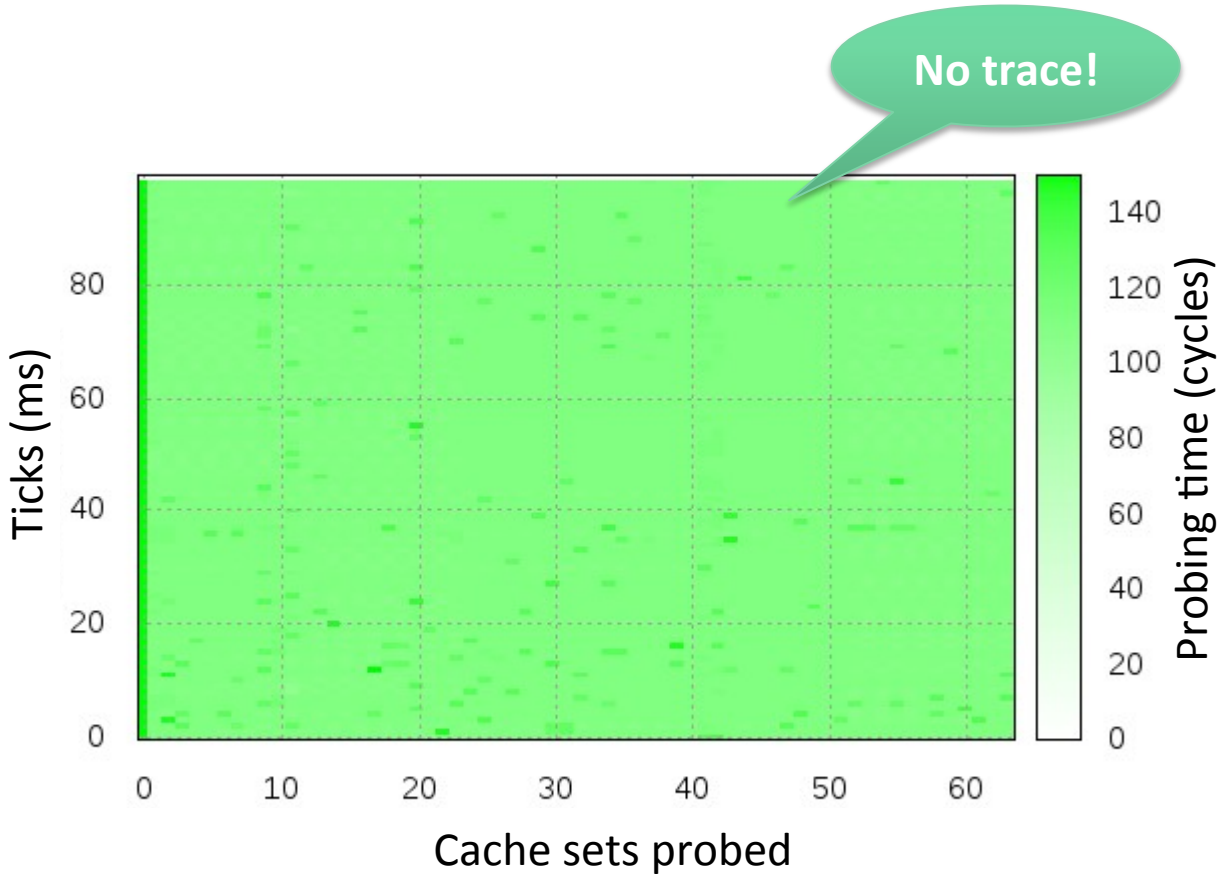
Solution: Flush L1 on context switch

- Direct cost low (1–2 µs)
- Indirect cost negligible:
  - Done only done on partition switch (≥ 1ms)
  - No hot data in L1 anyway after two partition switches
- Pain on x86: no selective L1 flush instruction
  - Need to flush by walking a buffer

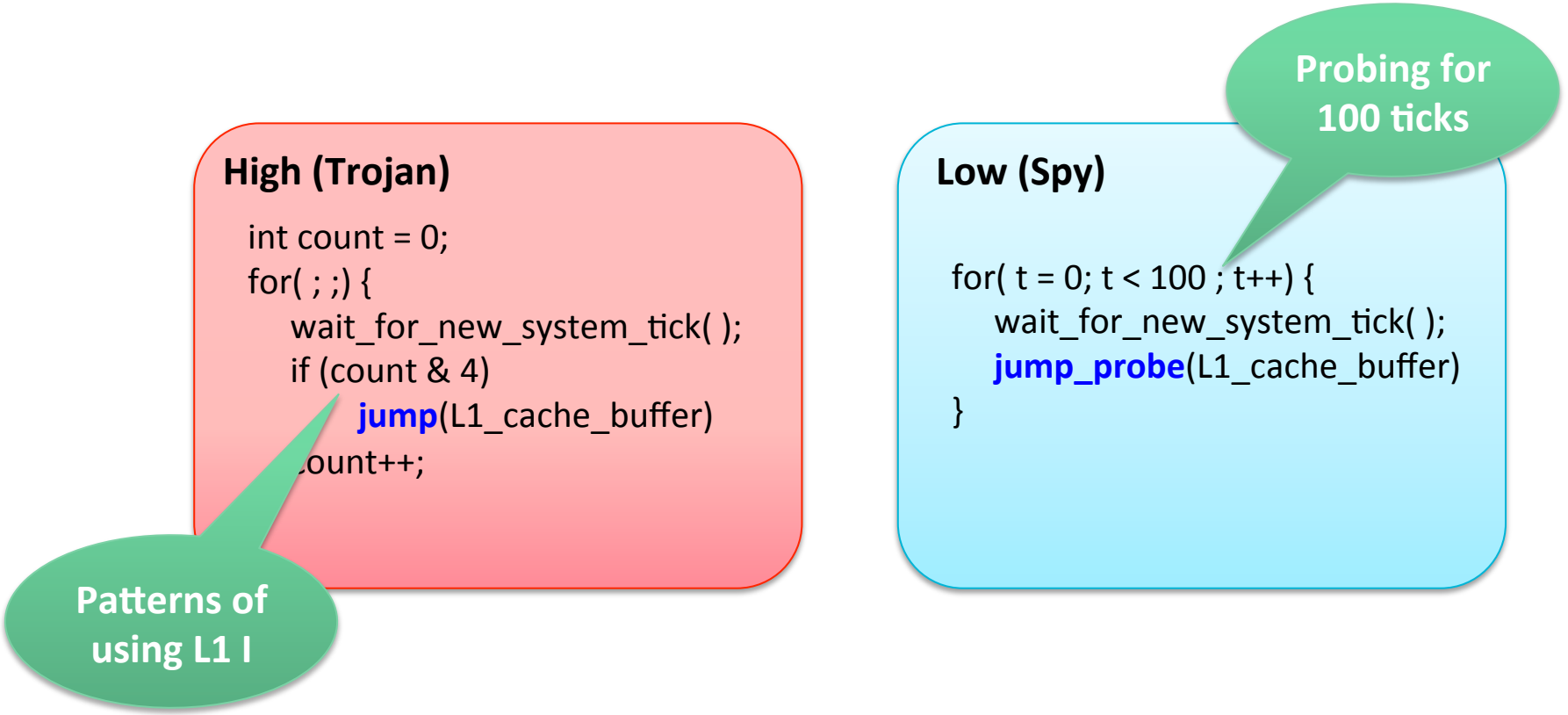
Unsafe: makes assumptions on line-replacement policy of cache

L1-D Cache Covert Channel

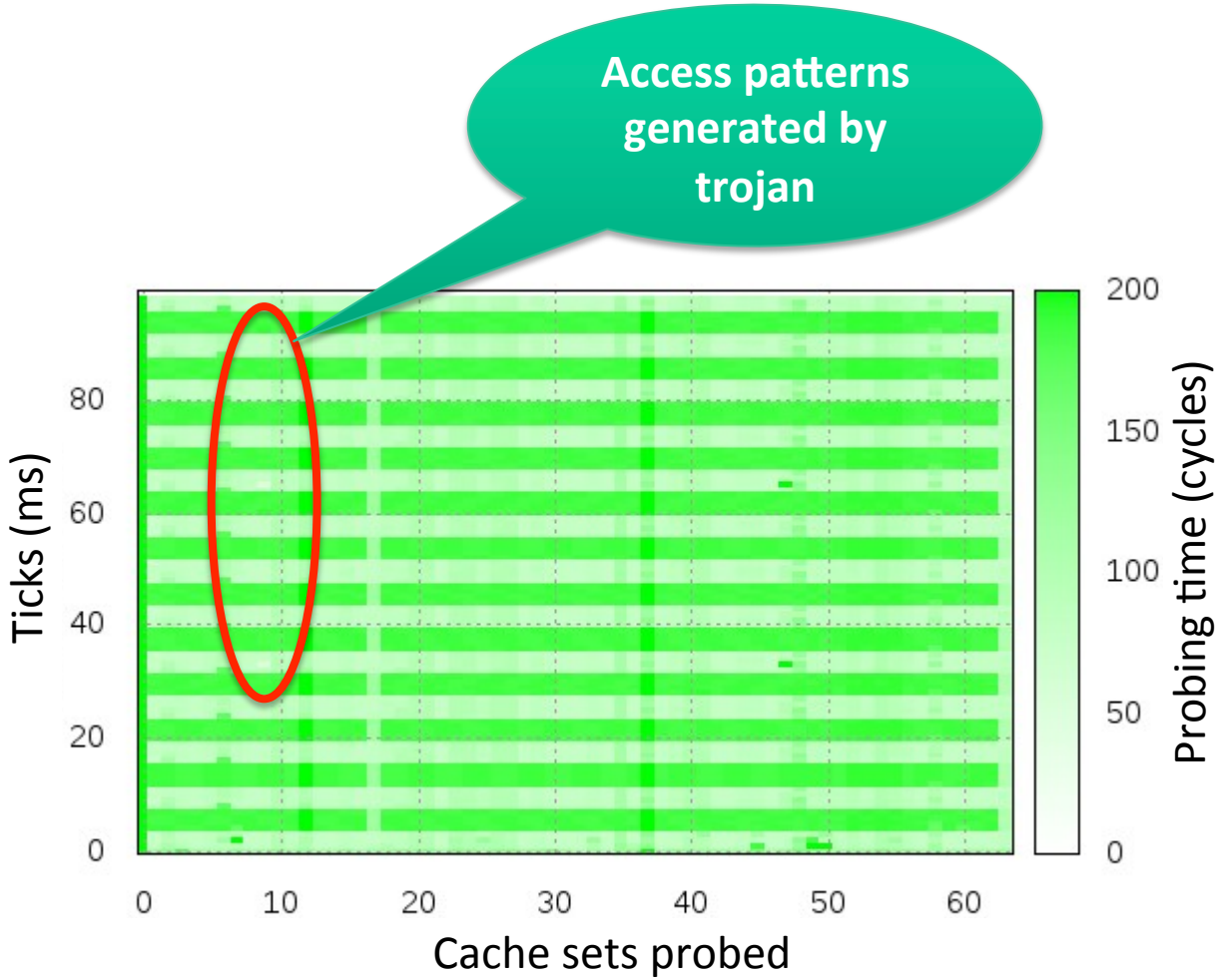
Spy observations with L1 flush on ARM A9



L1 I-Cache Covert Channel



L1-I Cache Covert Channel
Spy observations on ARM A9

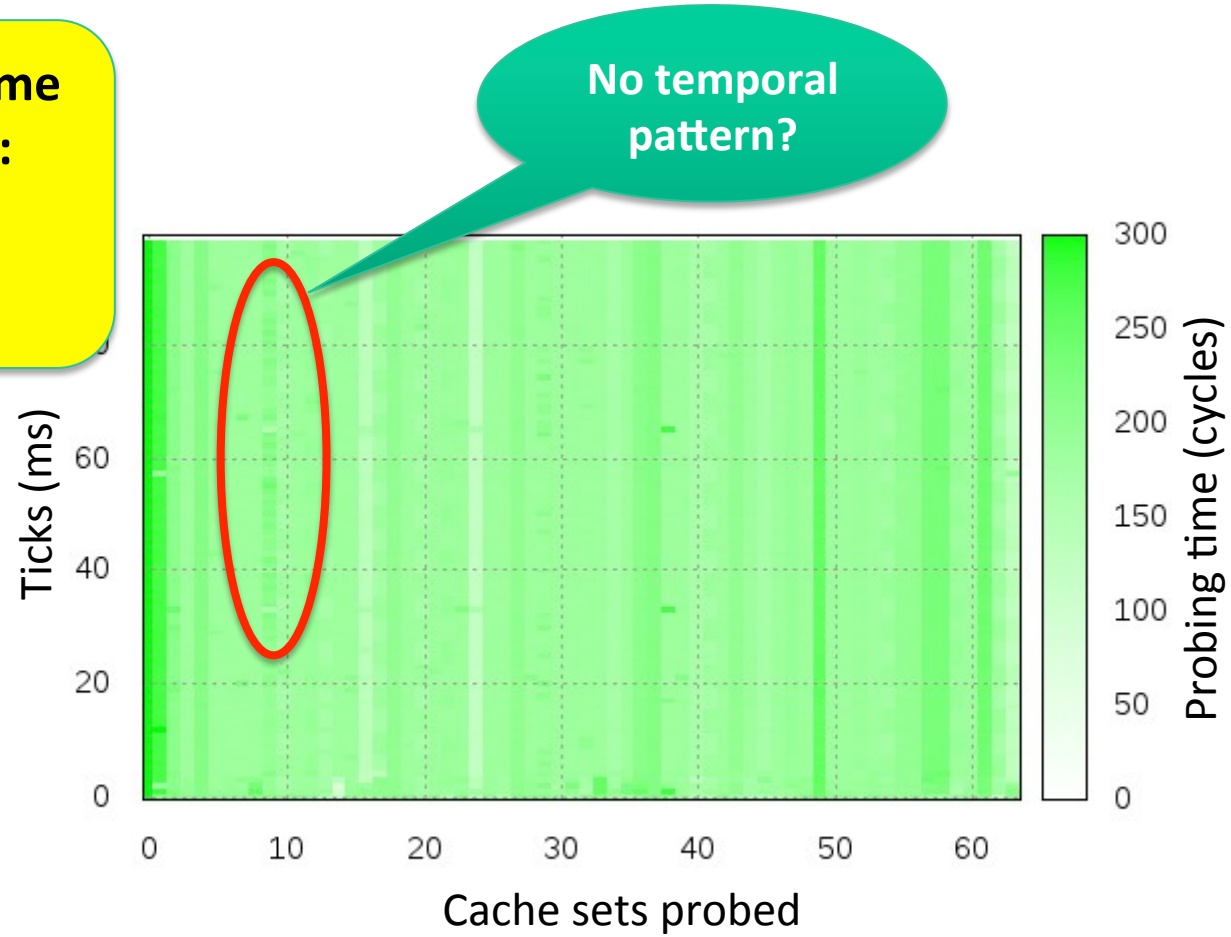


L1 I-Cache Covert Channel

Spy observations with flush on ARM A9

Can play same games with:

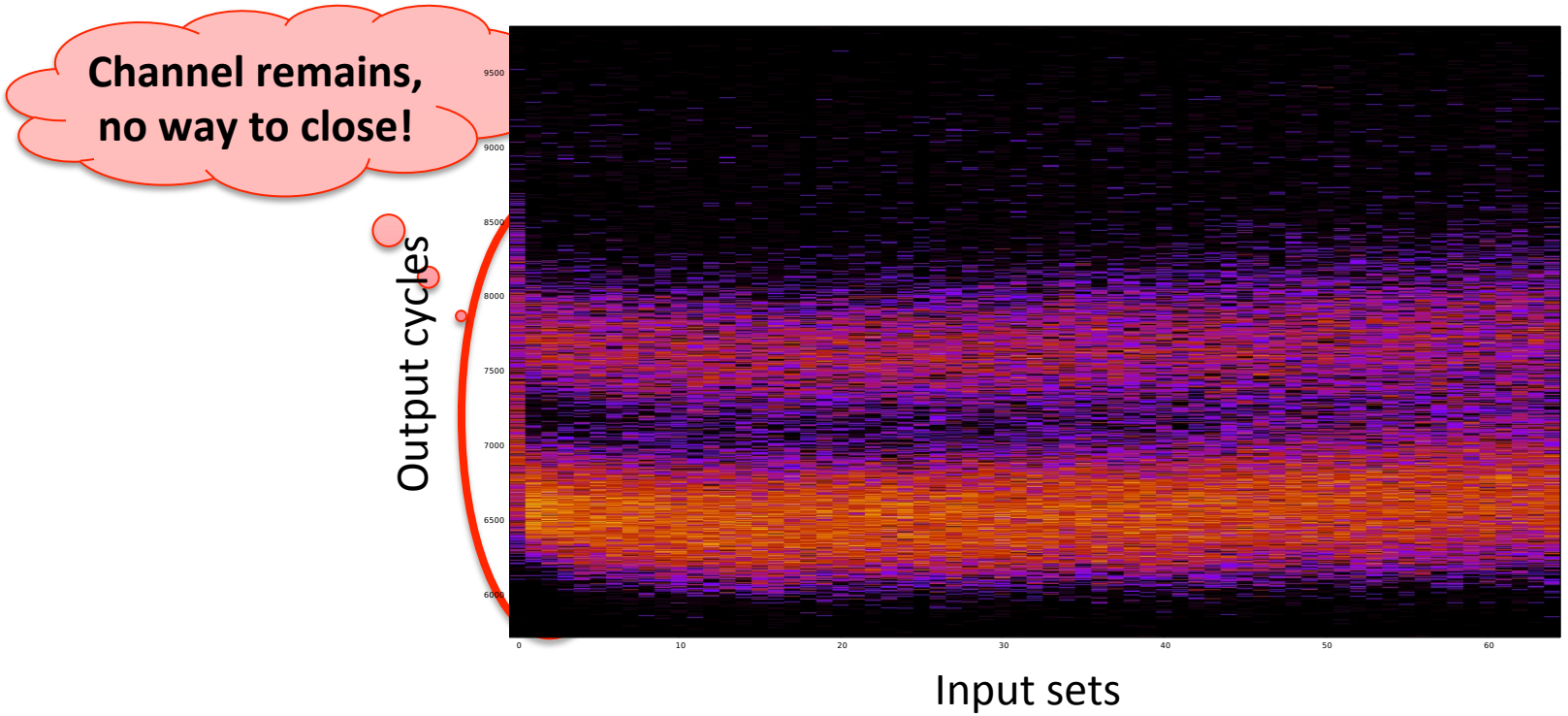
- TLB
- BPU



Recent Intel Processor: Skylake I-Cache

Approach:

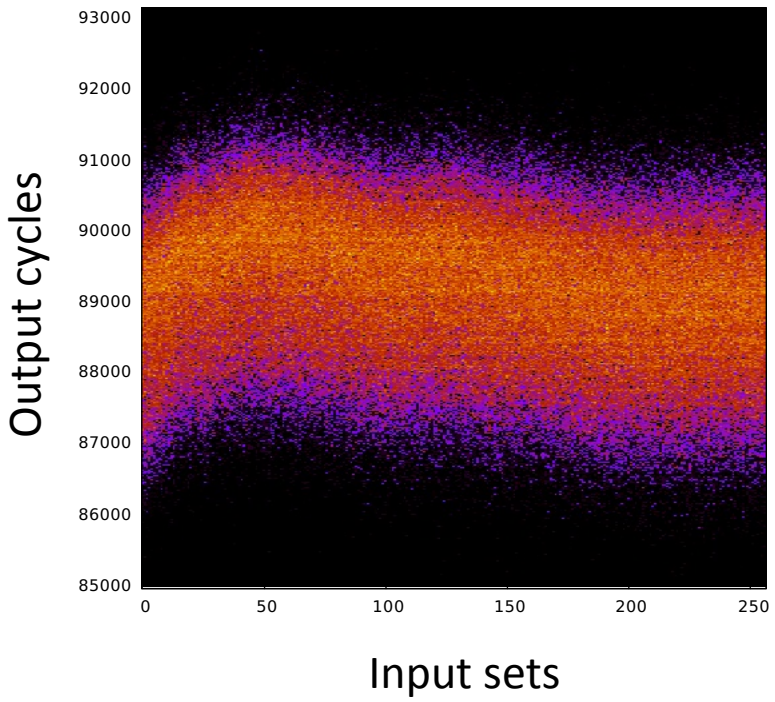
- Do everything the hardware lets you (expensive!)
- Flush L1, L2, L3, TLB, BPU
- Disable prefetcher



Recent ARM Processor: A53 I-Cache

Approach:

- Do everything the hardware lets you (expensive!)
- Flush L1, L2, L3, TLB, BPU
- Disable prefetcher



Discussion

Security seems a losing game!

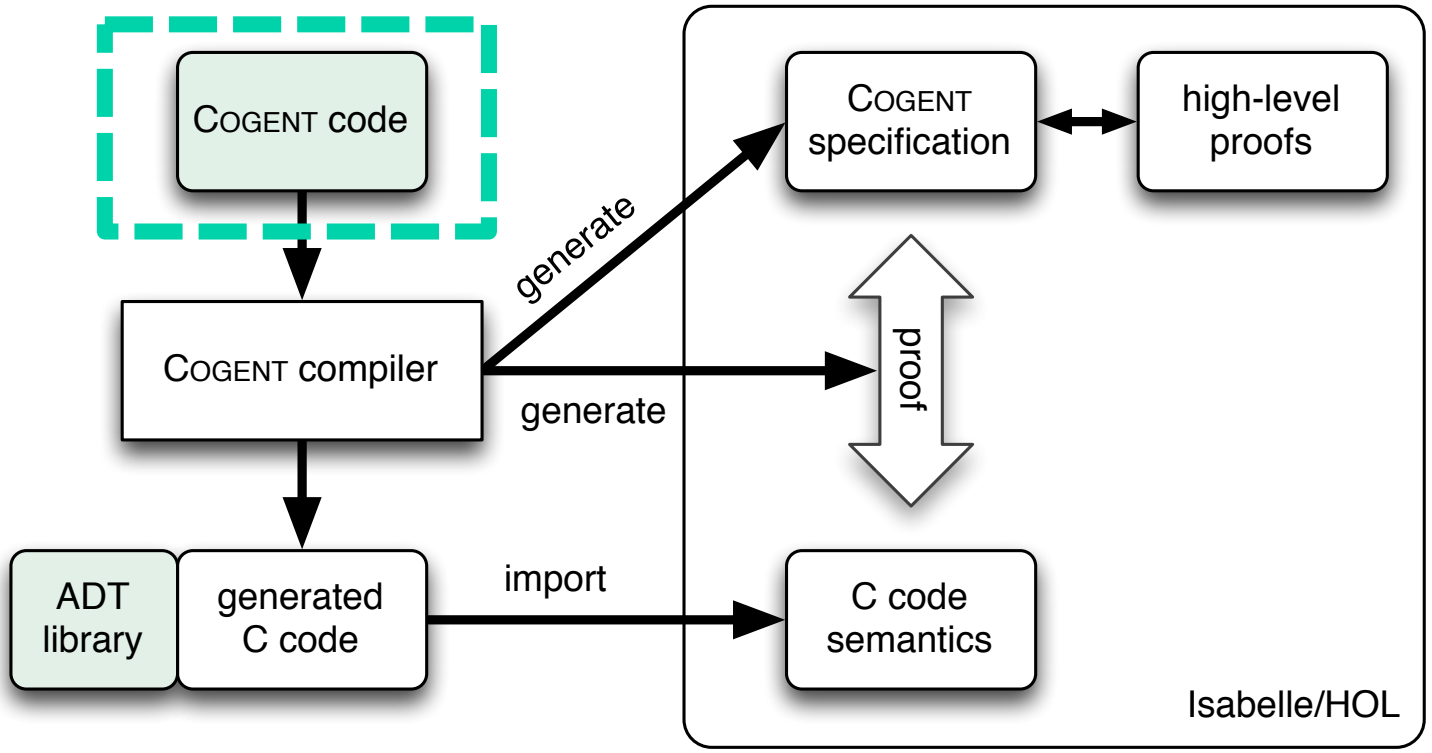
- The ISA is no longer a sufficient contact for security
  - Software designers are left to second-guess micro-architecture
  - Failures are predictable (and demonstrable)
  - The ISA over-abstracts hardware
- Security requires an extended ISA:
  - Must explicitly identify all shared hardware state
  - All shared state must be either partitionable or flushable
  - Architecture must provide explicit flush of all non-partitionable state

Need new hardware-software contract!

On a Different Note

Cogent workflow

- Cogent: purely functional memory-safe language



Thank you

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

https://trustworthy.systems

CSIRO