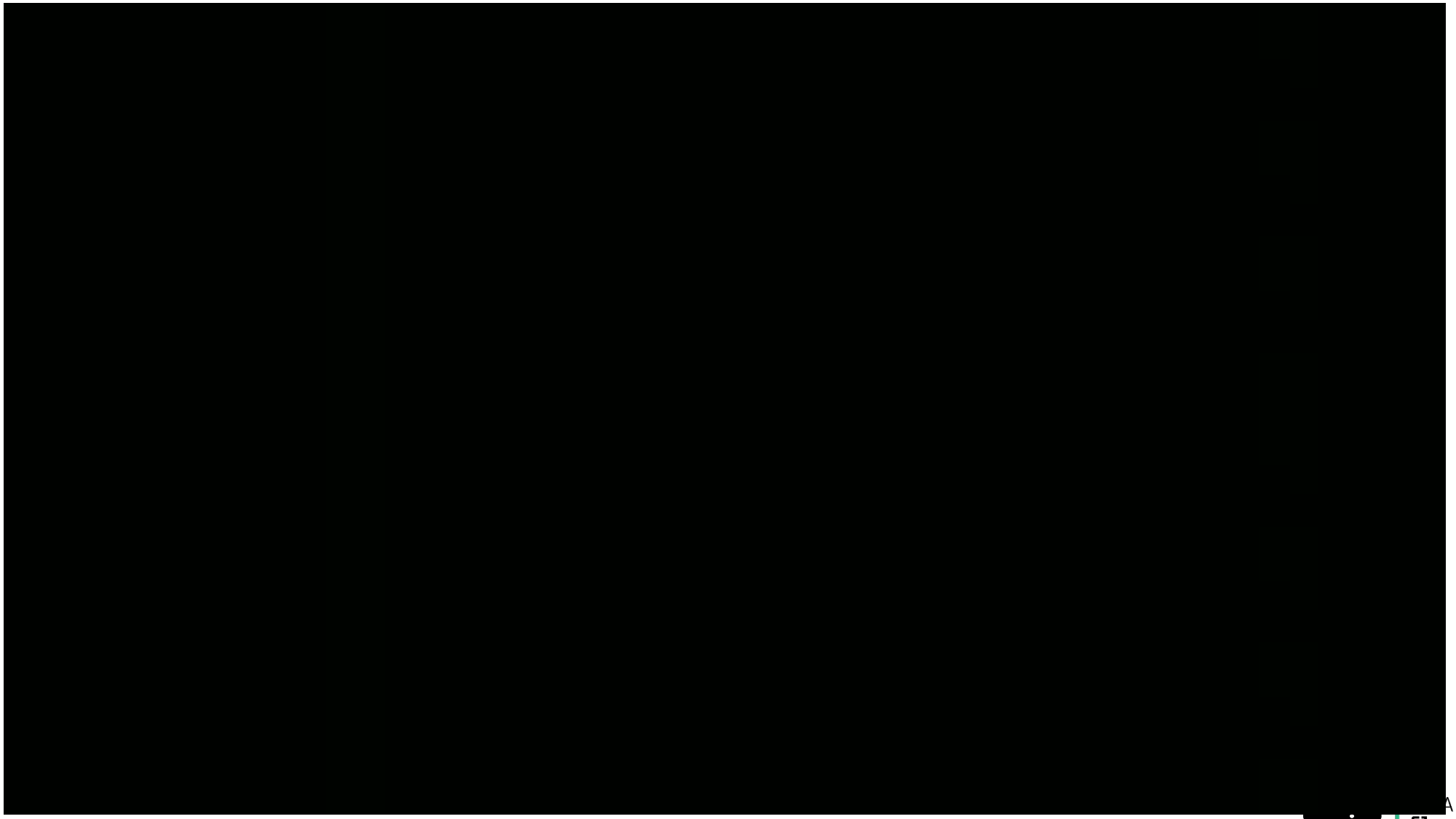
What's New in the World of seL4?

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

- FOSDEM'19

https://sel4.systems



Military-Grade Autonomous System?

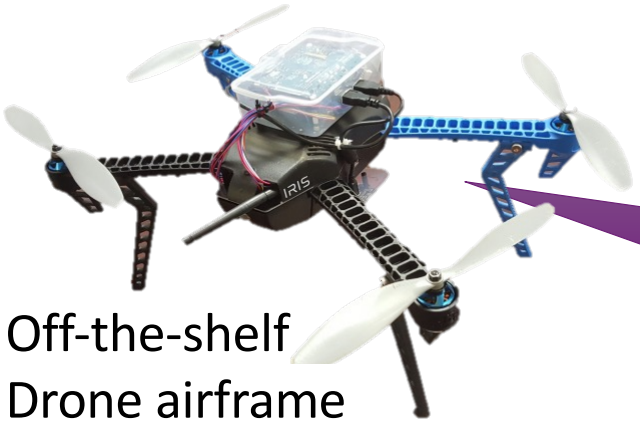
Hacked within 2 weeks by professional pen-testers!

DARPA HACMS: Protected Autonomy

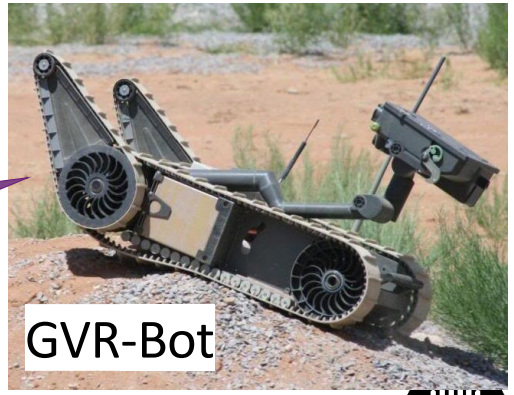

Unmanned Little Bird (ULB)

Retrofit existing system!


Autonomous trucks


Off-the-shelf Drone airframe

Develop technology


GVR-Bot

Outline

1. What's seL4
2. Technical update
   - Mixed-criticality scheduling
   - Security by architecture
   - Cogent: high-assurance components
3. Community & ecosystem status

The seL4 Microkernel

CSIRO DATA 61

World's Most Secure/Safe OS

seL4: The world's only operating-system kernel with provable security enforcement

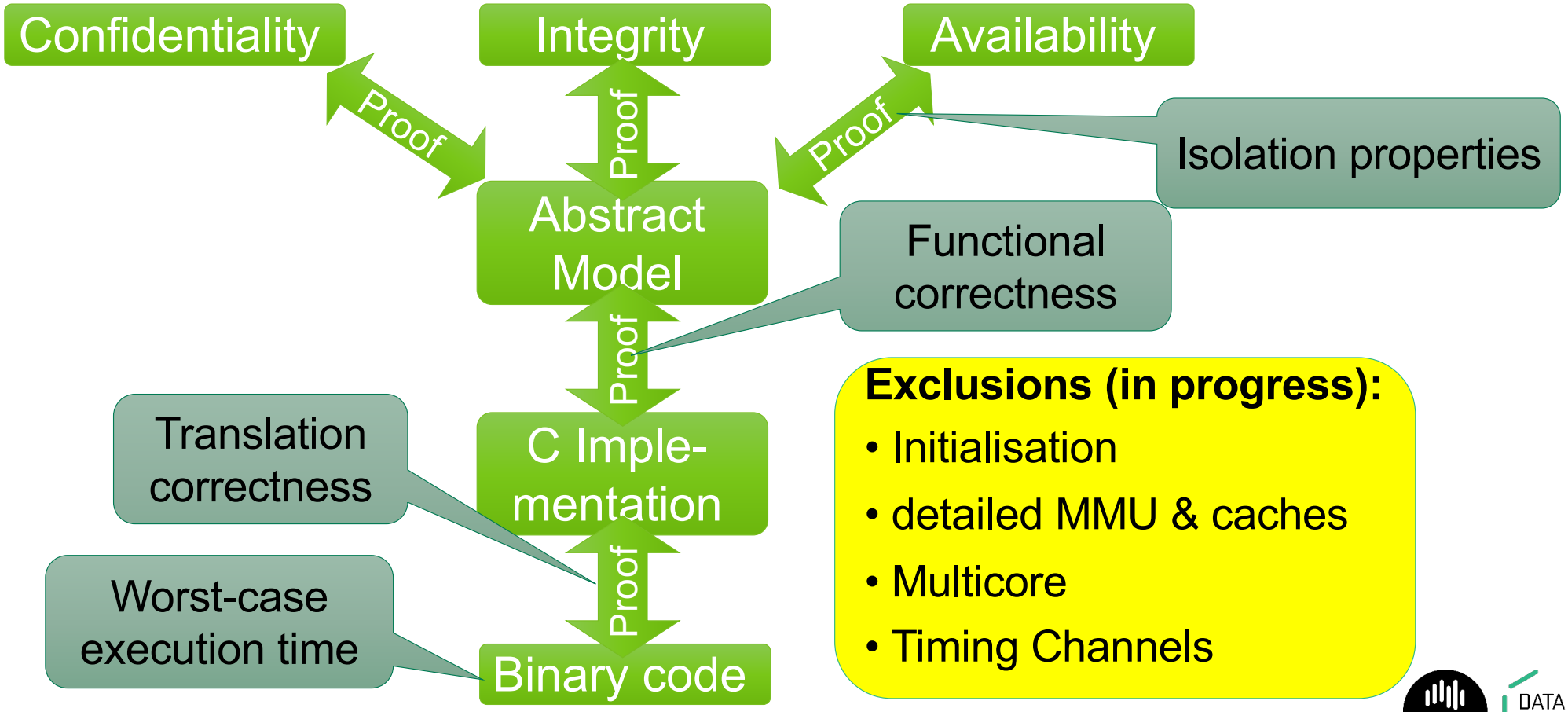
Open Source

seL4: The world's only protected-mode OS with complete, sound timeliness analysis

seL4: The world's fastest microkernel

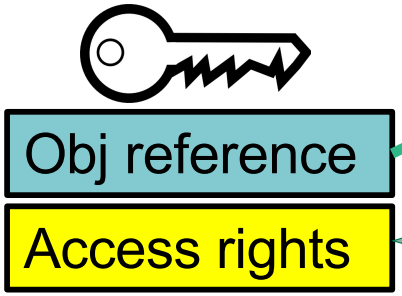
CSIRO DATA 61

Assurance Proof Chain



Core Mechanism: Capability

Capability = Access Token:
Prima-facie evidence of privilege



Eg. thread, address space

Eg. read, write, send, execute…

Capabilities provide:
- Fine-grained access control
- Reasoning about information flow

Any system call is invoking a capability:

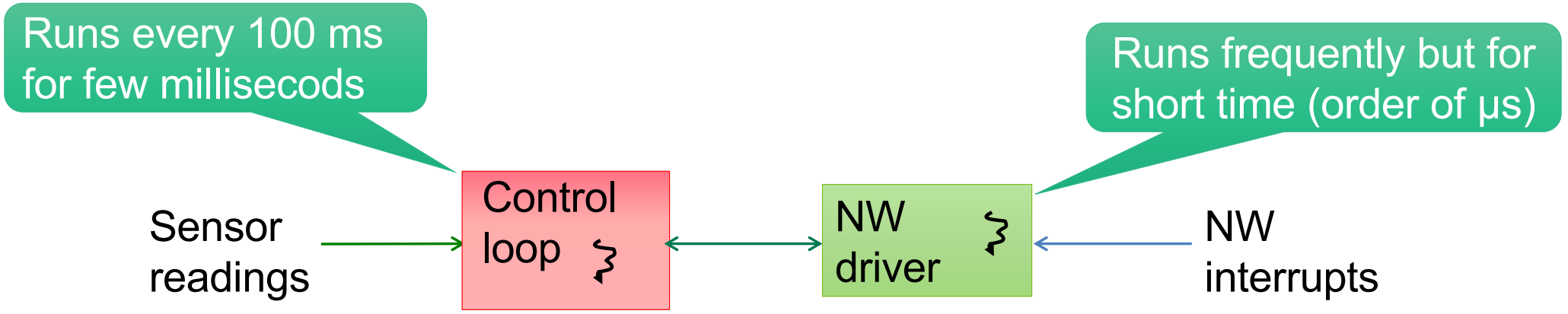
```
err = method( cap, args );
```

Mixed-Criticality Scheduling: Enforcing Temporal Safety

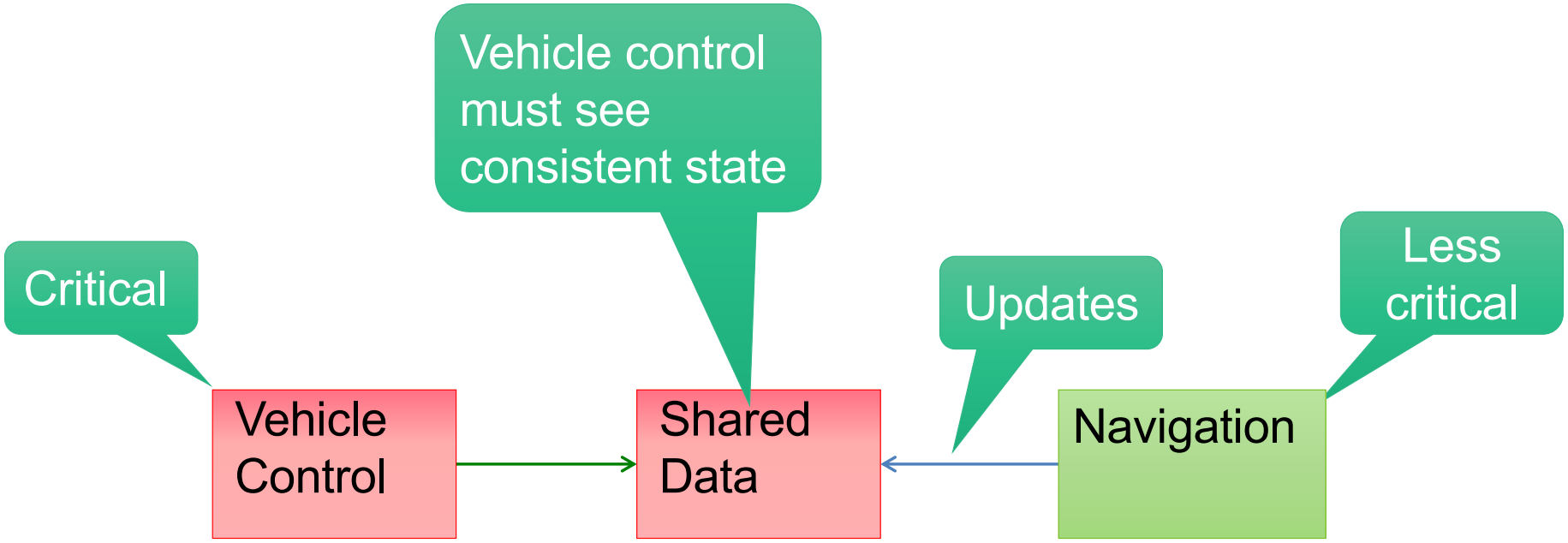
Integration Challenge: Mixed Criticality

NW driver must preempt control loop

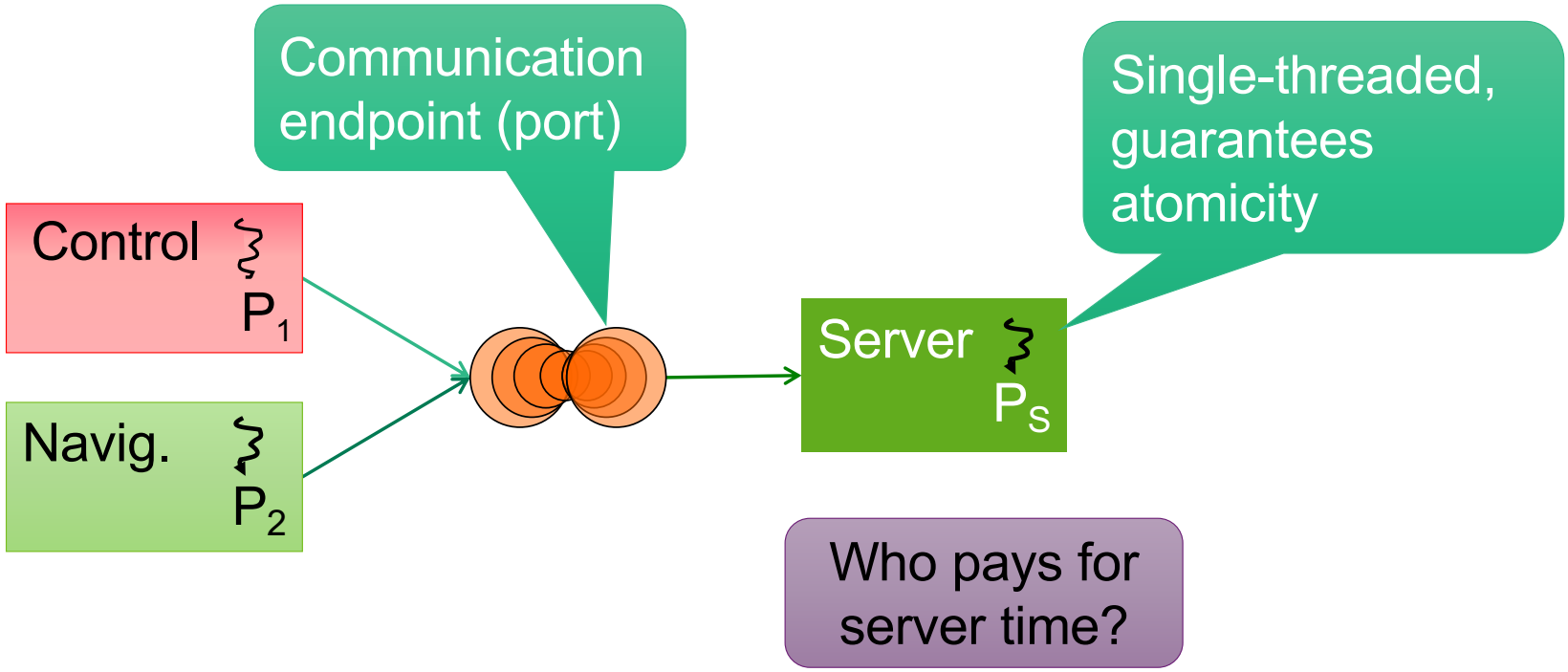
- … to avoid packet loss
- Driver must run at high prio
- Driver must be trusted not to monopolise CPU



Integration Challenge: Sharing



Sharing Through *Resource Server*



Scheduling Contexts: Time Caps

Classical thread attributes

- Priority
- Time slice

Not runnable if null

New thread attributes

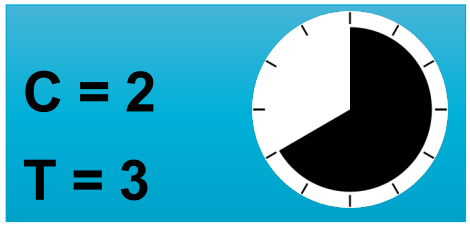
- Priority
- Scheduling context capability

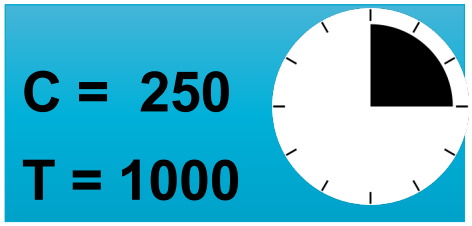
Limits CPU access!

Scheduling context object

- T: period
- C: budget (≤ T)

Capability for time

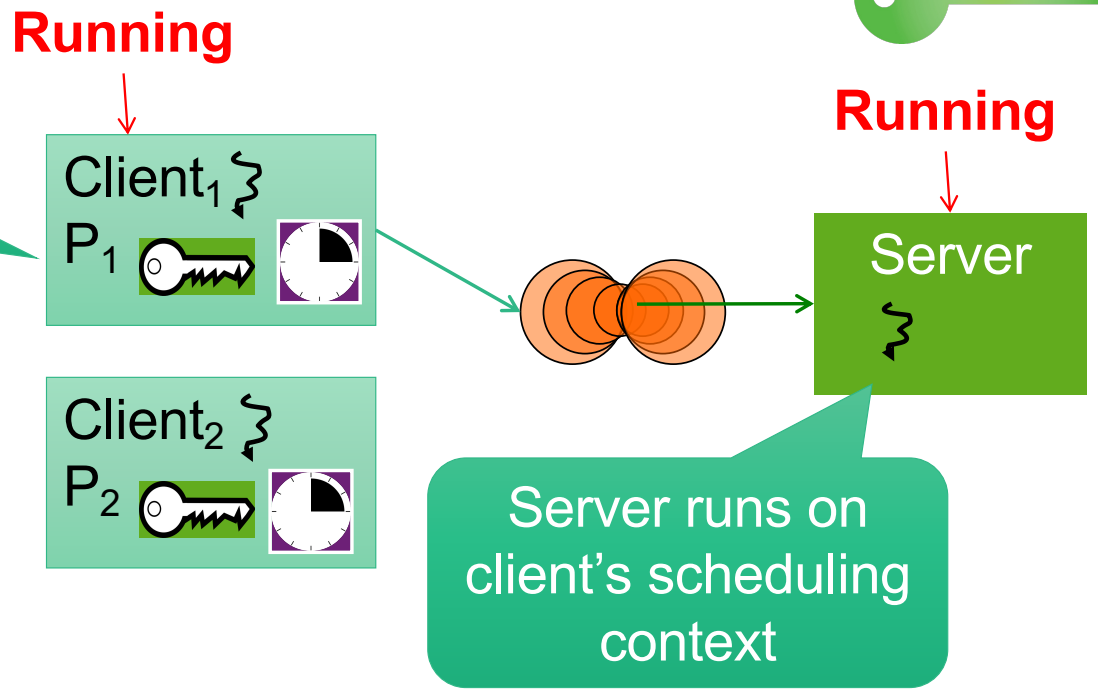




Shared Server

Will become mainline once verified

Client is charged for server's time



Server runs on client's scheduling context

- First* real-time system with capabilities for time
- Only high-assurance supporting mixed criticality with high utilisation

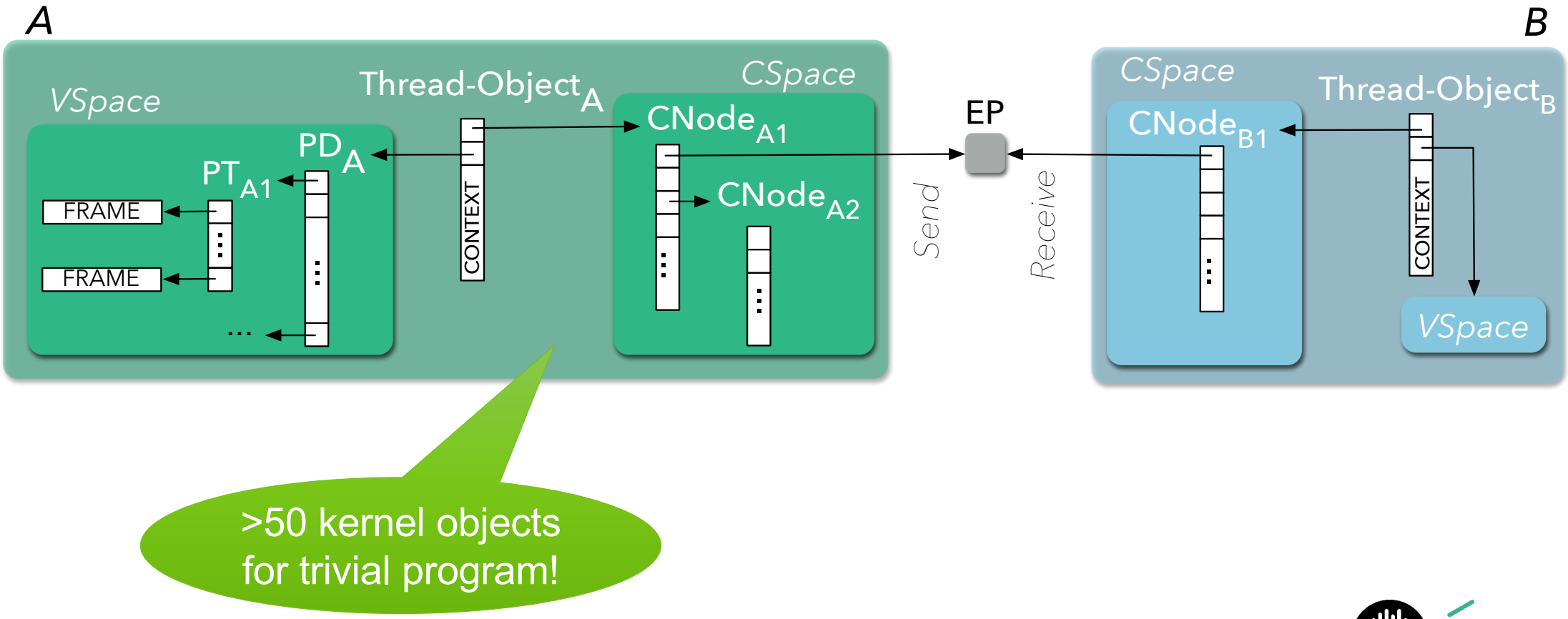
*Concurrent work with GWU

Scheduling-context capabilities: a principled, light-weight OS mechanism for managing time [Lyons et al, EuroSys'18]

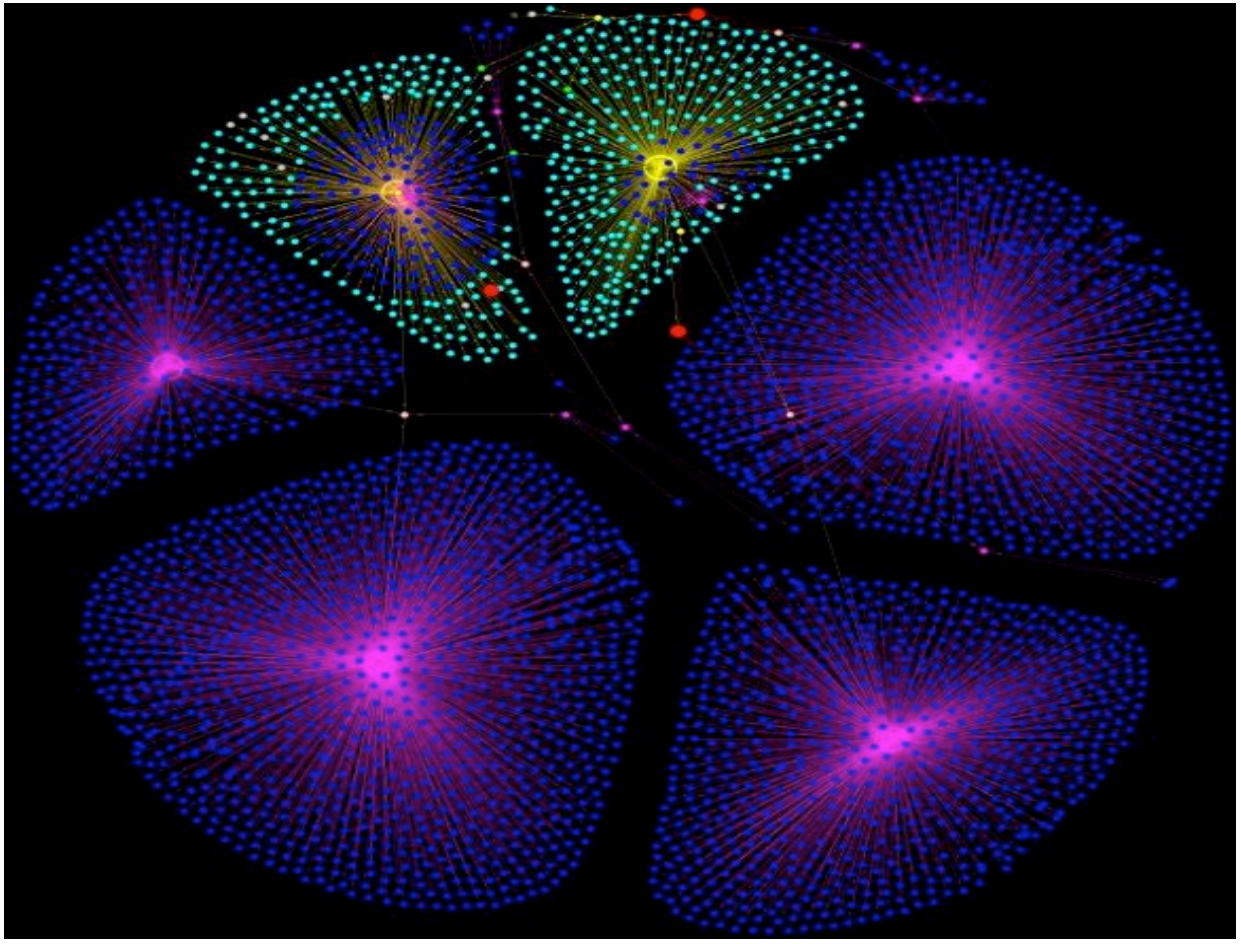
Security Enforcement by Architecture

CSIRO DATA 61

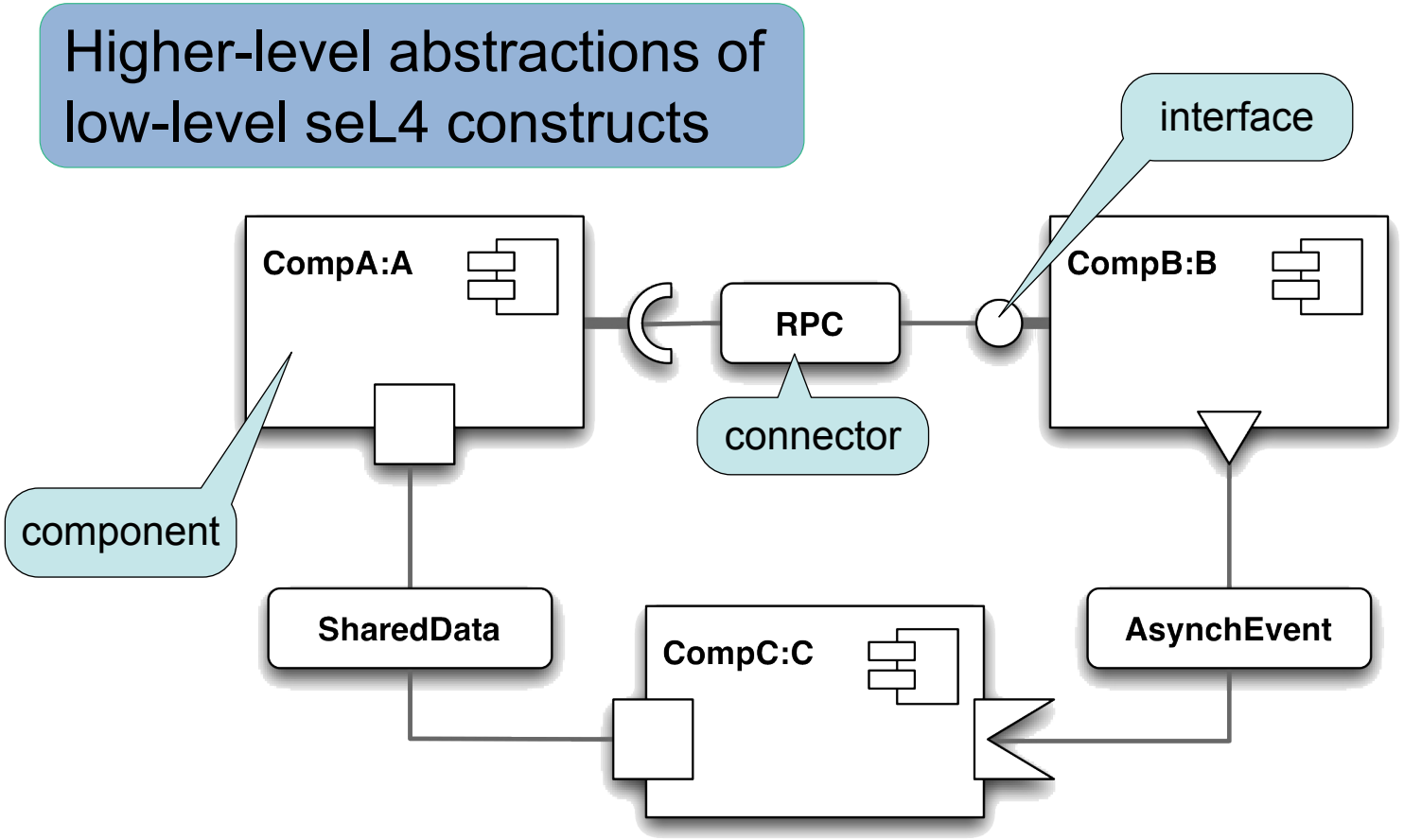
Issue: Primitives are Low-Level



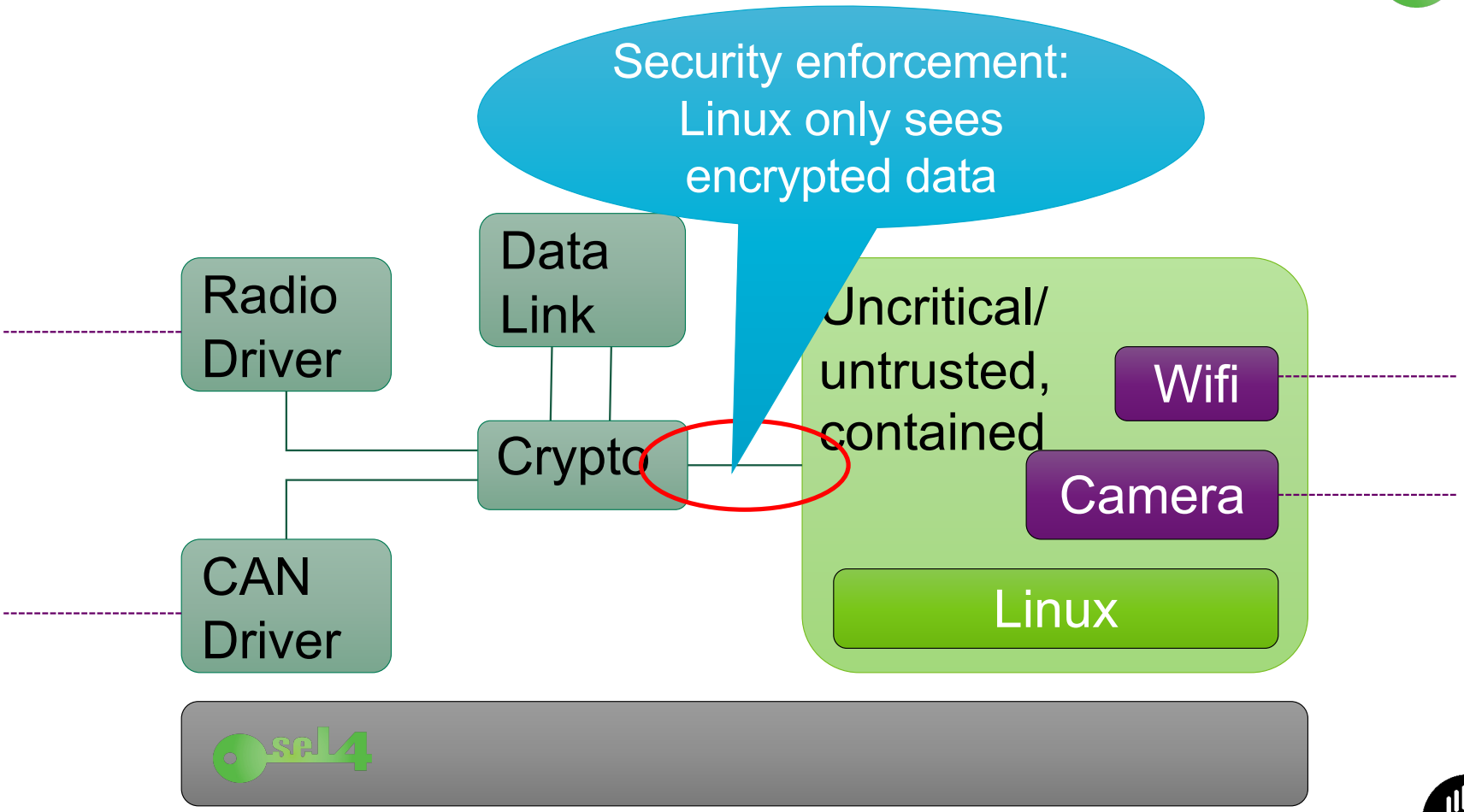
Non-Trivial But Simple System



Component Middleware: CAmkES

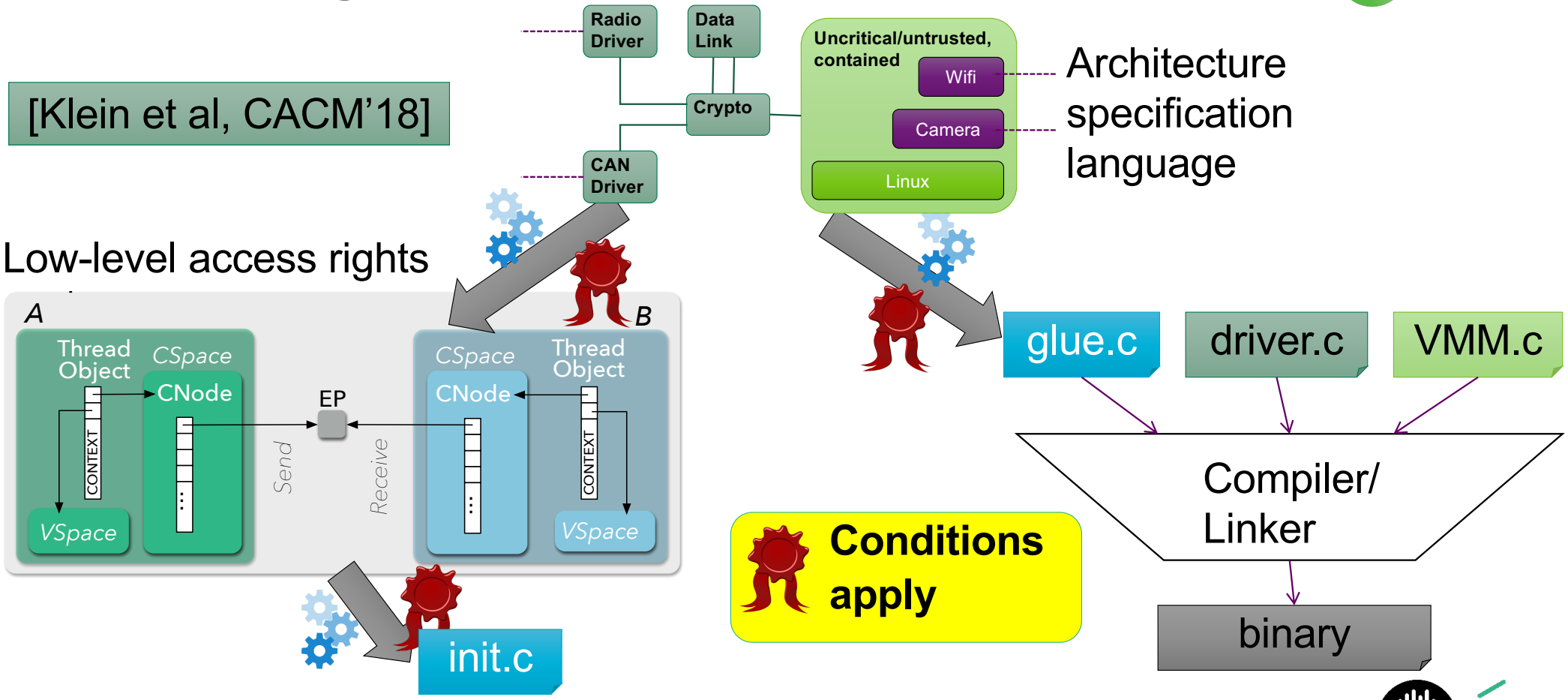


Simplified UAV Architecture

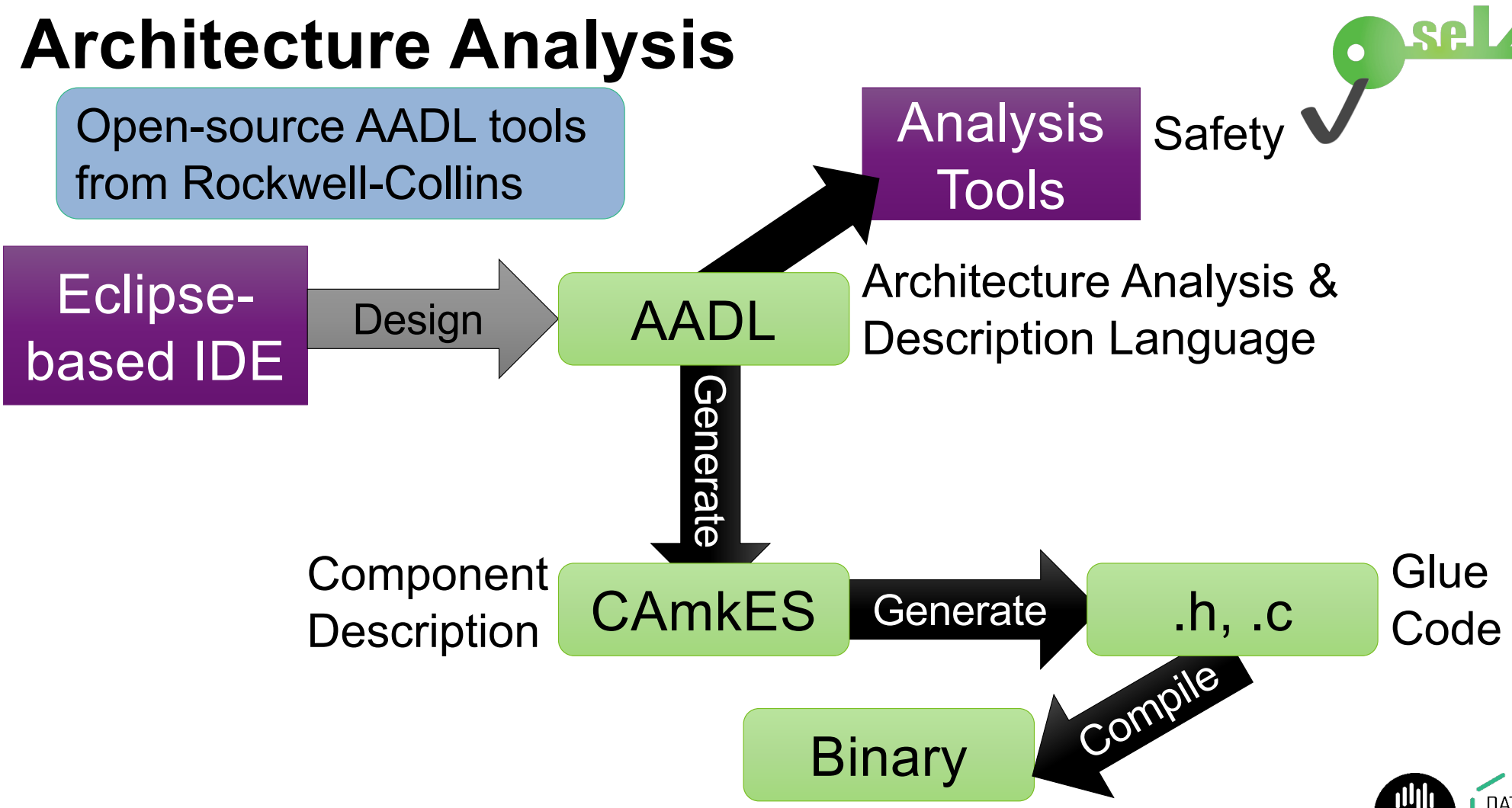


Enforcing the Architecture

[Klein et al, CACM'18]



Architecture Analysis

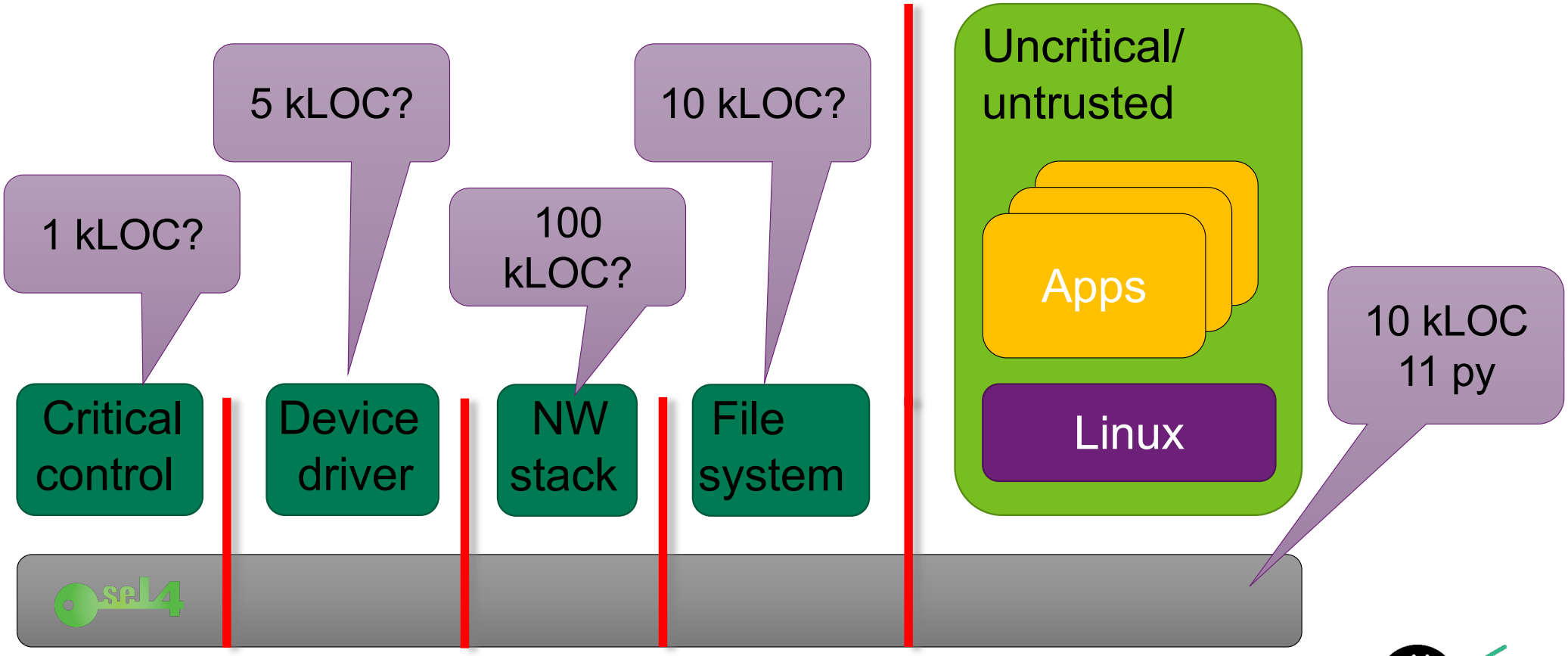


High Assurance Code Beyond the Kernel

seL4

CSIRO DATA 61

Beyond the Kernel

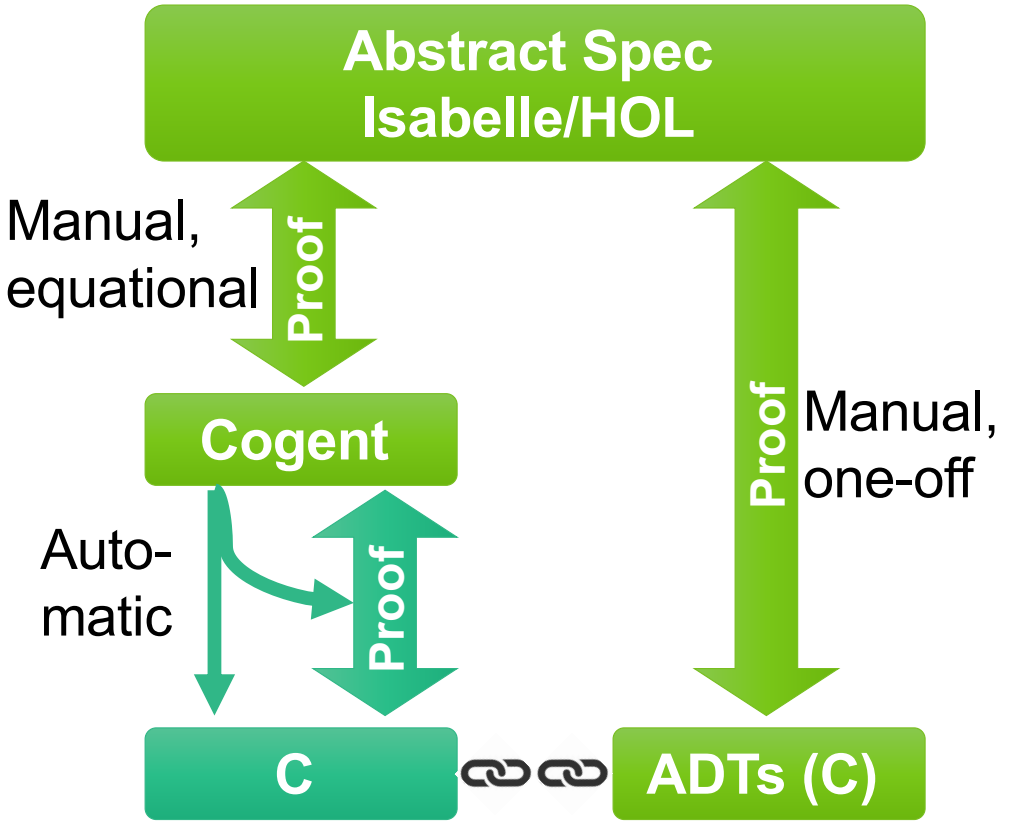


Cogent: Code & Proof Co-Generation

Aim: Reduce cost of verified systems code

- Restricted, purely functional *systems* language
- Type- and memory safe, not managed
- Turing incomplete
- Case-studies: BilbyFs, ext2, F2FS, VFAT

[O'Connor et al, ICFP'16; Amani et al, ASPLOS'16]

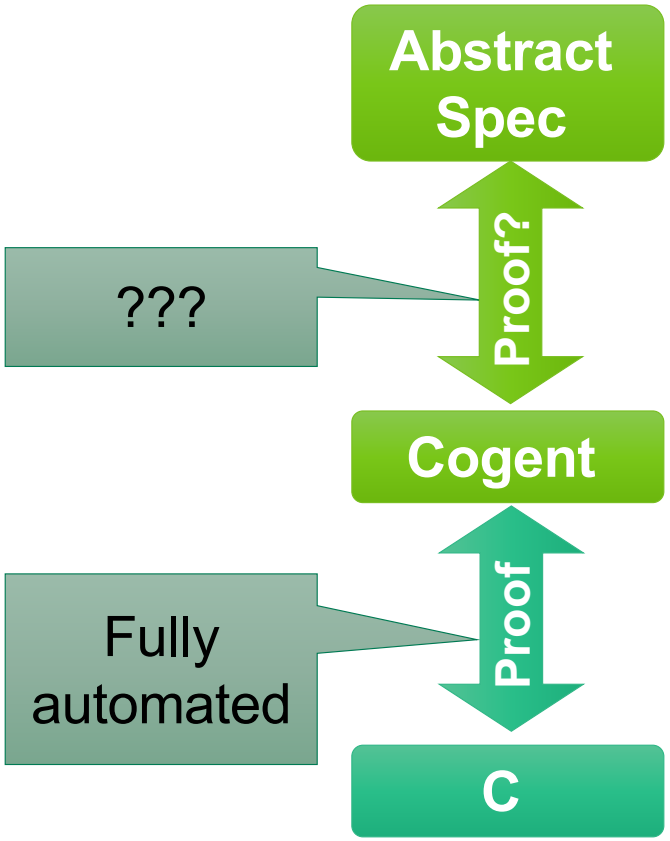


Manual Proof Effort

BilbyFS functions	Effort	Isabelle LoP	Cogent SLoC	Cost $/SLoC	LoP/ SLOC
isync() iget() library	9.25 pm	13,000	1,350	150	10
sync()-specific	3.75 pm	5,700	300	260	19
iget()-specific	1 pm	1,800	200	100	9
seL4	**12 py**	**180,000**	**8,700 C**	**350**	**20**

BilbyFS: 4,200 LoC Cogent

Dependable And Affordable?



Dependability-cost tradeoff:

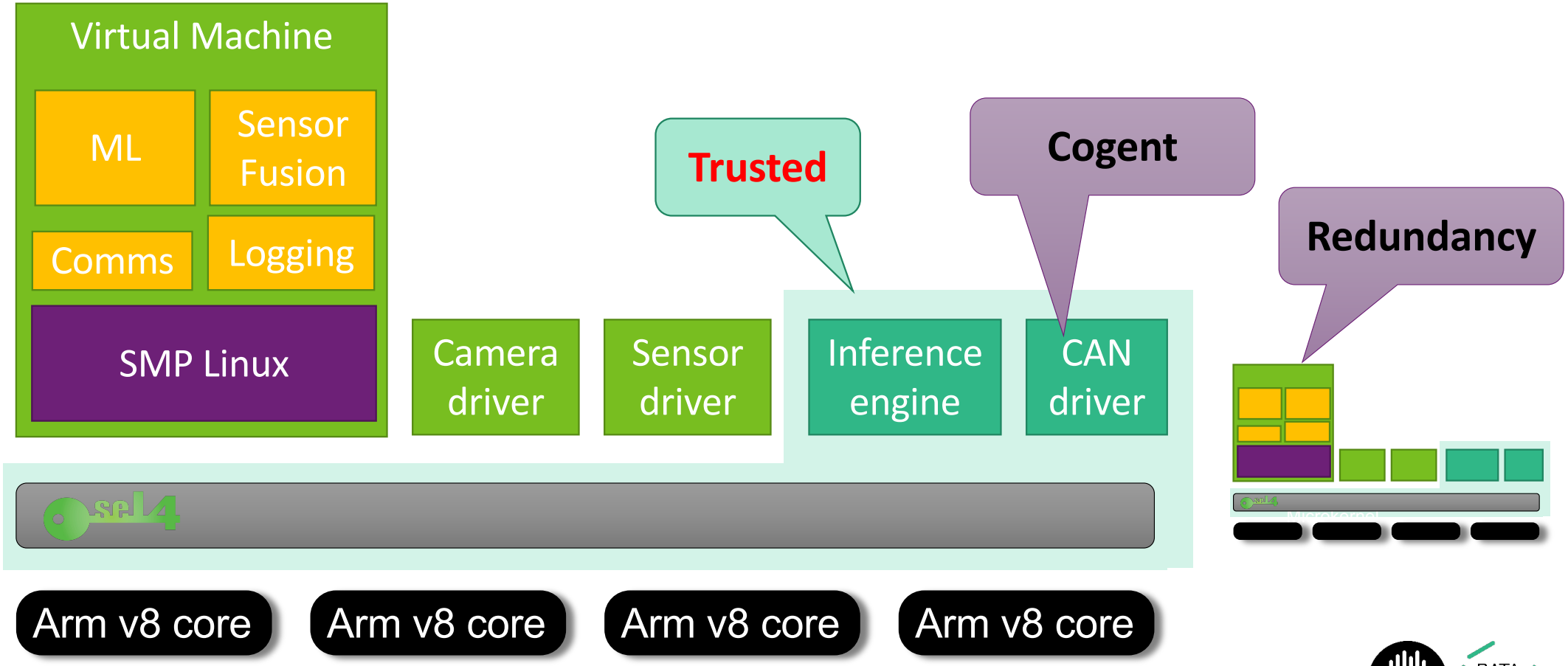
- Reduced faults through safe language
- Property-based testing (QuickCheck)
- Model checking
- Full functional correctness proof

Spec reuse!

Work in progress:

- Language expressiveness
- Reduce boiler-plate code
- Network stacks
- Device drivers

Application to Autonomous Cars



Community & Ecosystem

Status: The Good News

- Growing user base, active developer mailing list
- Heavy support from US government (DoD, DHS)
  - "US seL4 Center of Excellence"
- High-assurance evaluation
  - Certified for defence use & in deployed in UK, AU
- Commercial funding for accelerating development
  - multicore verification
  - RISC-V verification
  - Cogent
- Being designed into many systems
  - Moving beyond defence: medical, industrial control, autonomous cars

Downside: Ecosystem

Very spartanic

- Few components (drivers, file systems, network stacks)
- Components are poorly documented, hard to find, unsupported
- Shortage of tools
- Insufficient community engagement

Main reason: We're too busy

- Large number of projects
- Can't hire quickly enough
- Responsive but not proactive

Improving Engagement

- Recently added 6 OS engineers (plus 4 student casuals)
- Engaging with CoE
  - improving tools, contributing components, ports
- Documenting what is there, support status
  - Encourage community adoption & new contributions
- Provide visibility for contributors

Improving Transparency

Done:

- Started RFC process for major changes
  https://sel4kernel.atlassian.net/secure/Dashboard.jspa

Soon:

- Opening up kernel+platform Jira
- Opening up discourse group
- Public roadmap

Please talk to us!

Trustworthy Systems Team

Thank you!

Gernot Heiser

gernot.heiser@data61.csiro.au

@GernotHeiser

Security is no excuse for poor performance!

https://sel4.systems