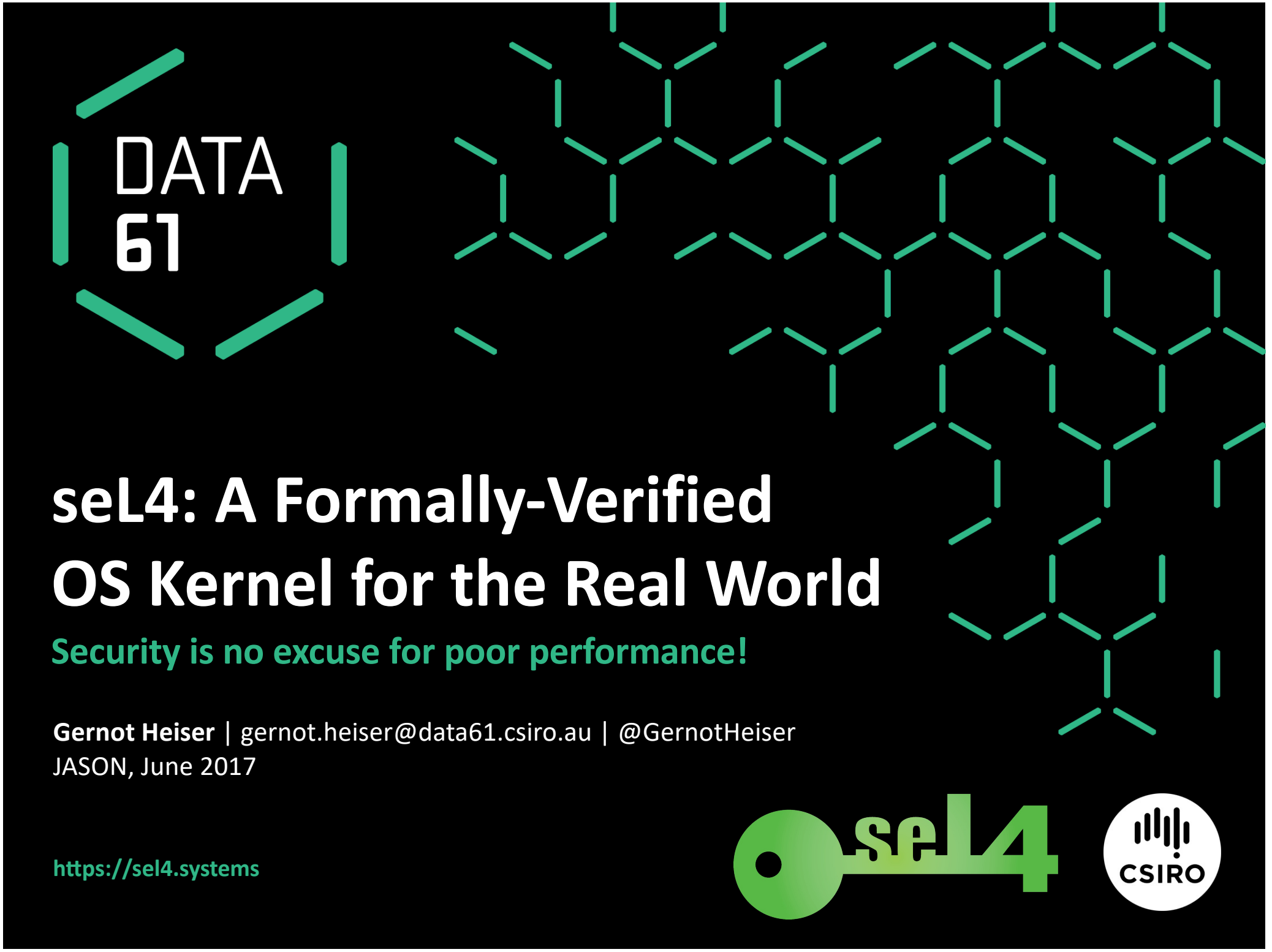


seL4: A Formally-Verified OS Kernel for the Real World

Security is no excuse for poor performance!

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
JASON, June 2017

https://sel4.systems

Formal Methods vs Systems

What is seL4?

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

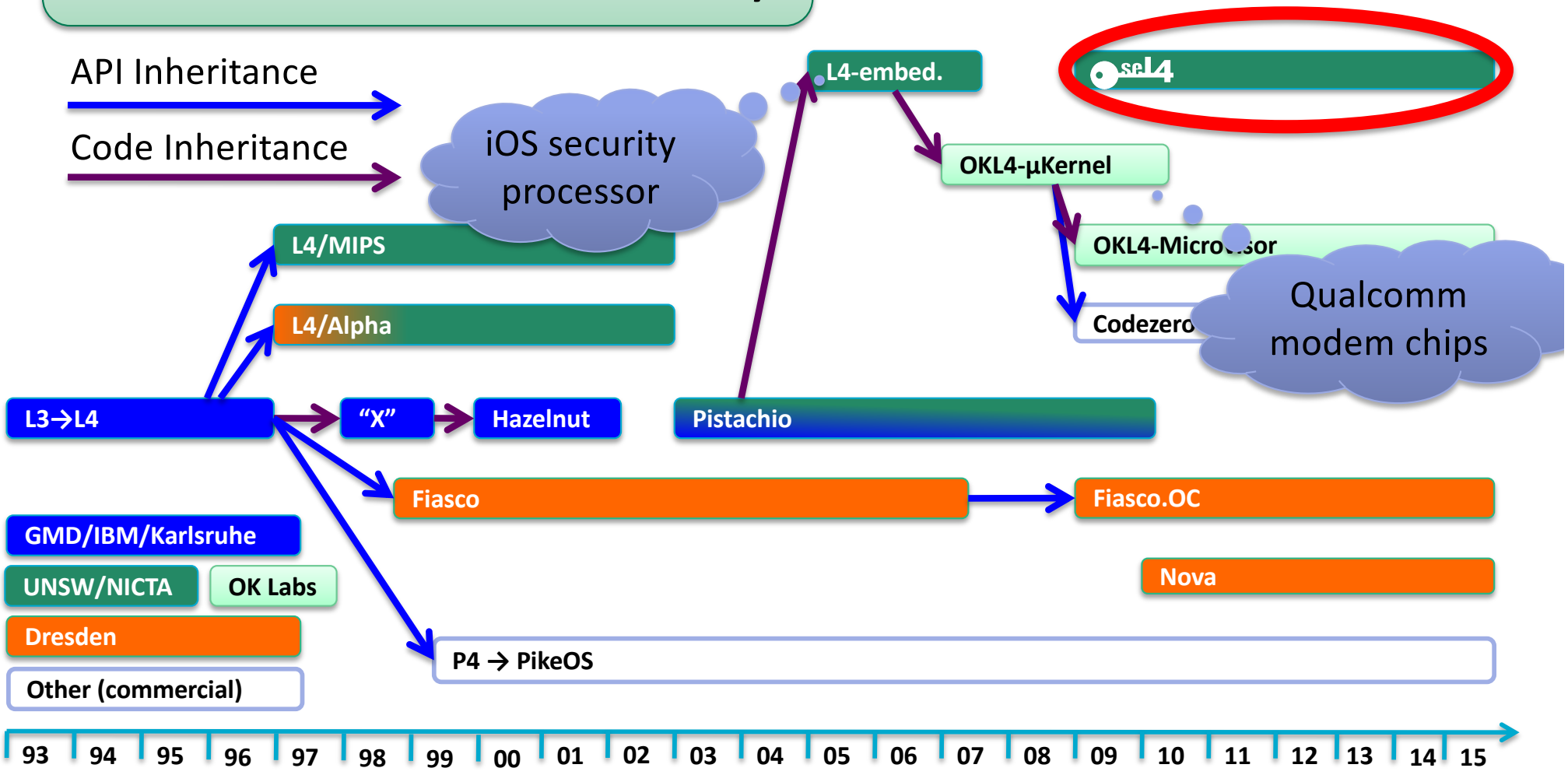
Open Source

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

seL4: The world's fastest microkernel

20+ Years of L4 Microkernel R&D

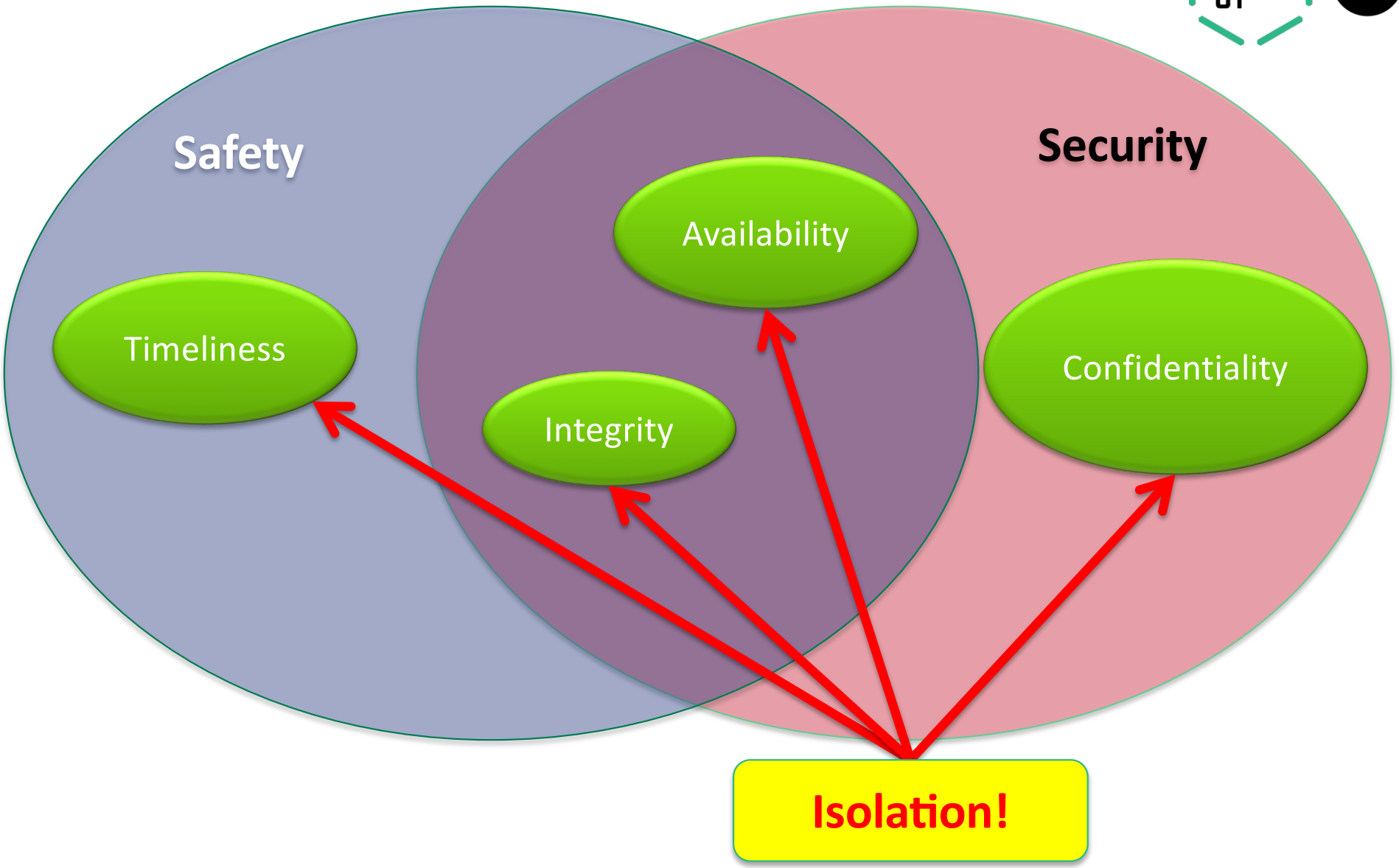
seL4: The latest (and most advanced) member of the L4 microkernel family



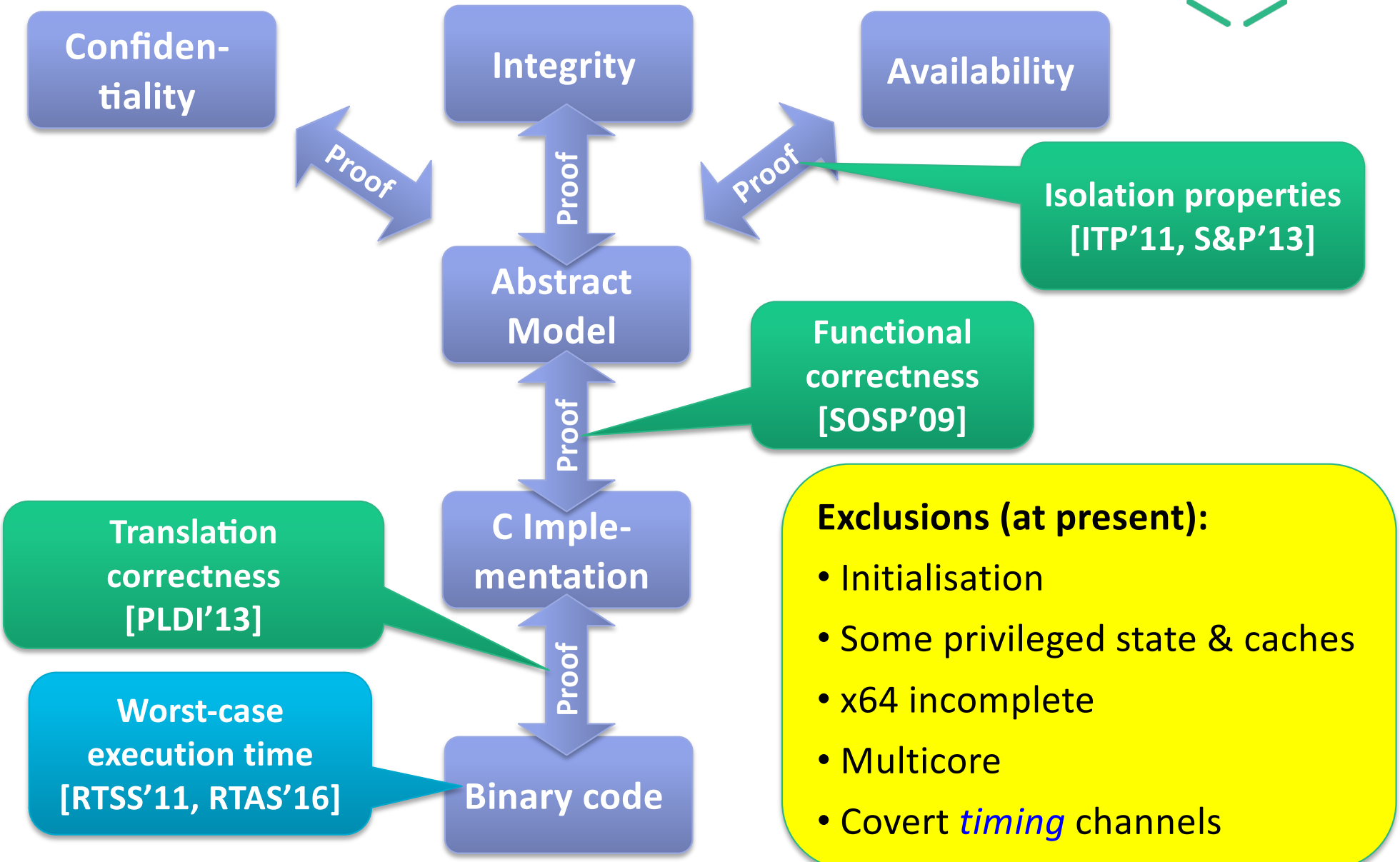
L4 IPC Performance over 20 Years

Name	Year	Processor	MHz	Cycles	µs
Original	1993	i486	50	250	5.00
Original	1997	Pentium	160	121	0.75
L4/MIPS	**1997**	**R4700**	**100**	**86**	**0.86**
L4/Alpha	**1997**	**21064**	**433**	**45**	**0.10**
Hazelnut	2002	Pentium 4	1,400	2,000	1.38
Pistachio	**2005**	**Itanium**	**1,500**	**36**	**0.02**
OKL4	**2007**	**XScale 255**	**400**	**151**	**0.64**
NOVA	2010	i7 Bloomfield (32-bit)	2,660	288	0.11
seL4	**2017**	**i7 Skylake (32-bit)**	**3,400**	**203**	**0.06**
seL4	**2017**	**I7 Skylake (64-bit)**	**3,400**	**138**	**0.04**
seL4	**2017**	**Cortex A53**	**1,200**	**225**	**0.19**

Security and Safety



Proof of Security and Safety



Exclusions (at present):

- Initialisation
- Some privileged state & caches
- x64 incomplete
- Multicore
- Covert *timing* channels

Capability-Based Access Control

Cap = Access Token:
Prima-facie evidence of privilege

Obj reference

Access rights

Object

Eg. thread, address space

Eg. read, write, send, execute…

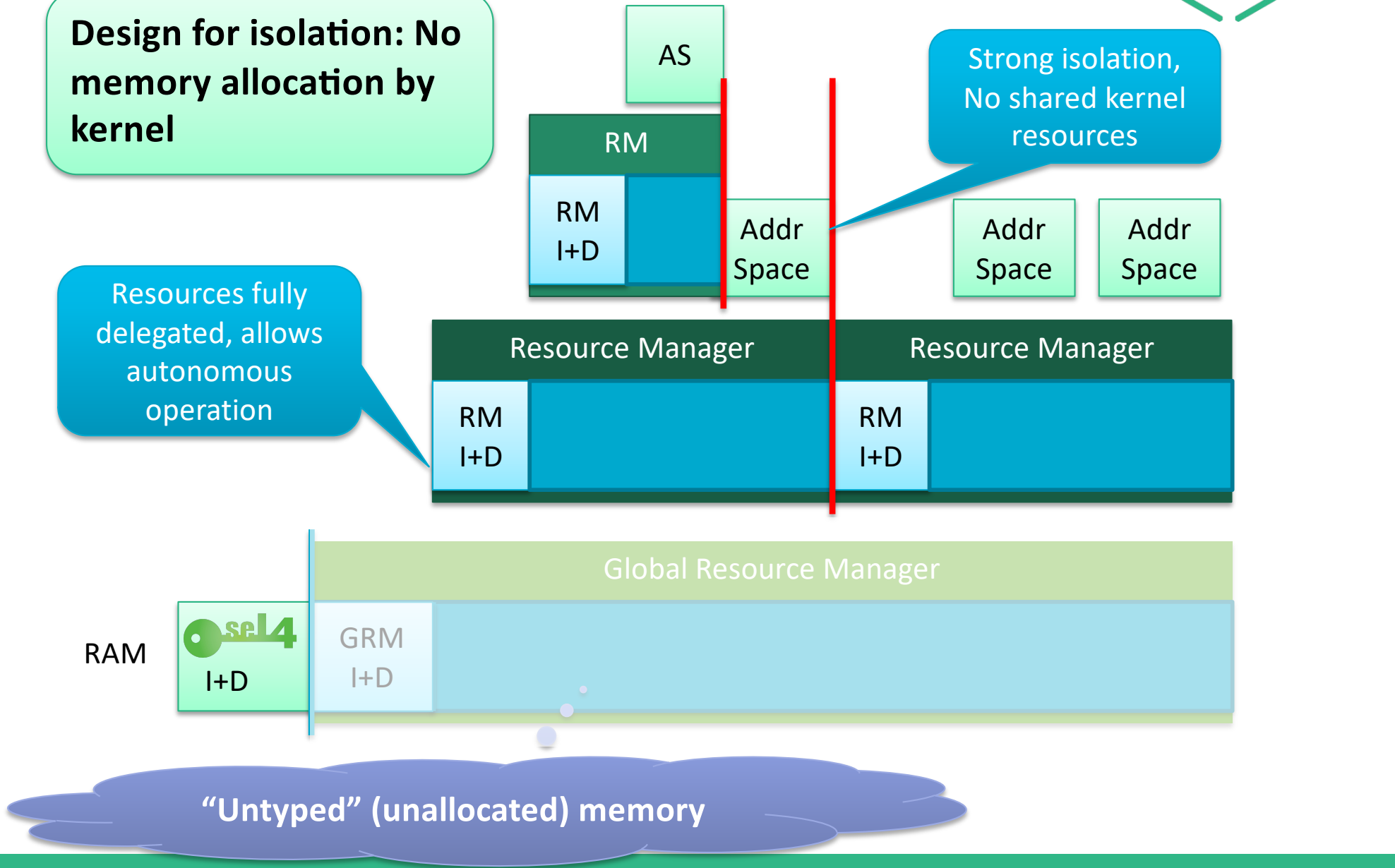
Capabilities provide
- Fine-grained access control
- Reasoning about information flow

Any system call is invoking a capability:

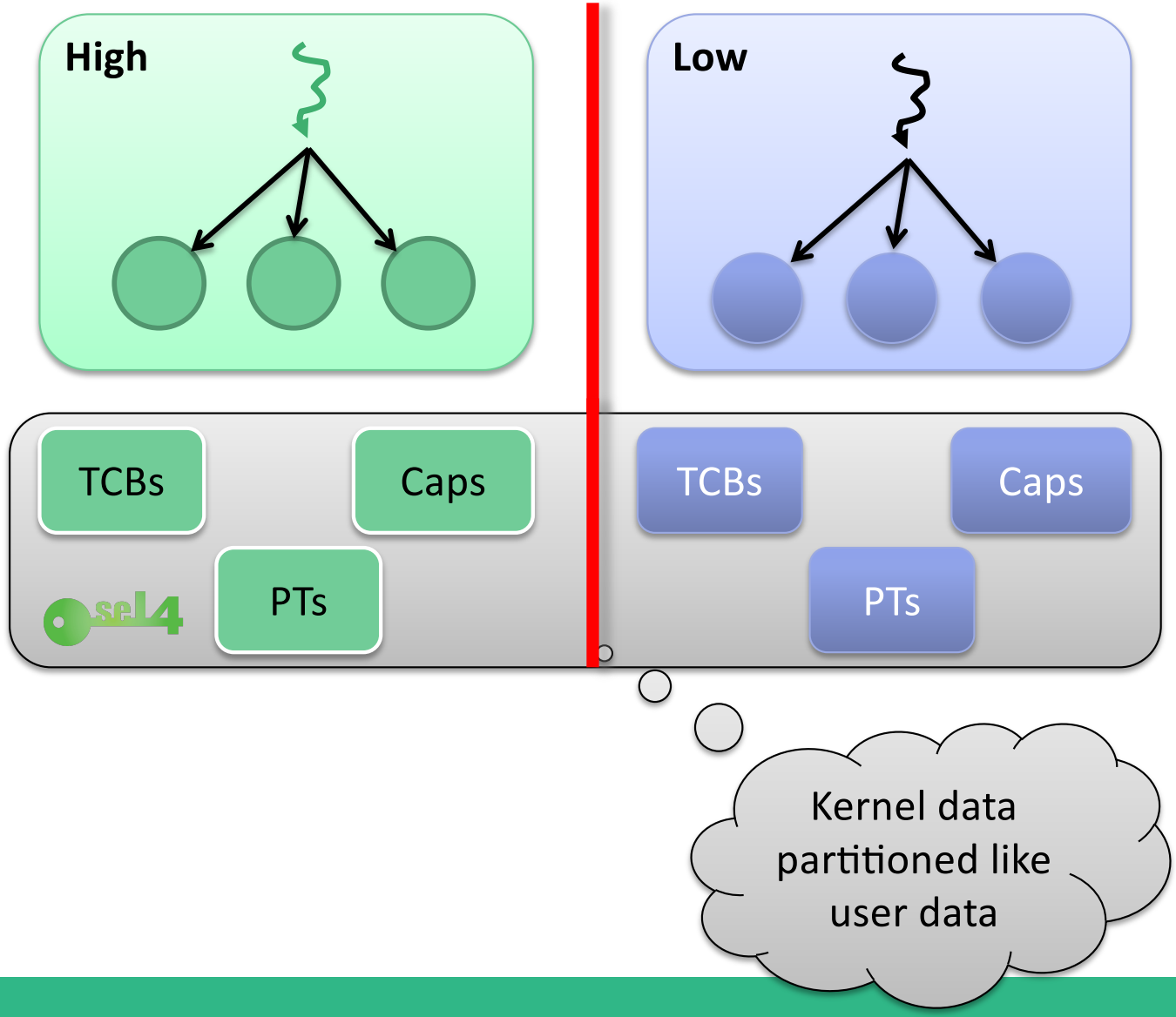
```
err = method( cap, args );
```

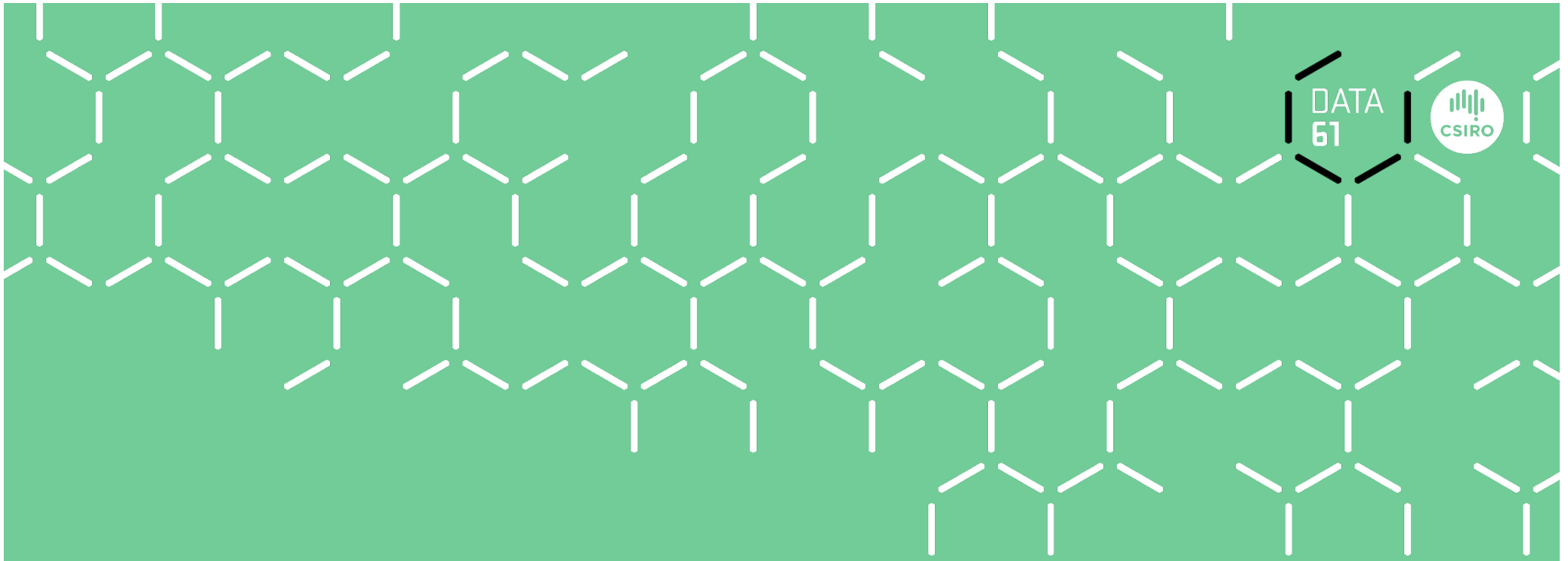
What's Different to Other Microkernels?

Design for isolation: No memory allocation by kernel



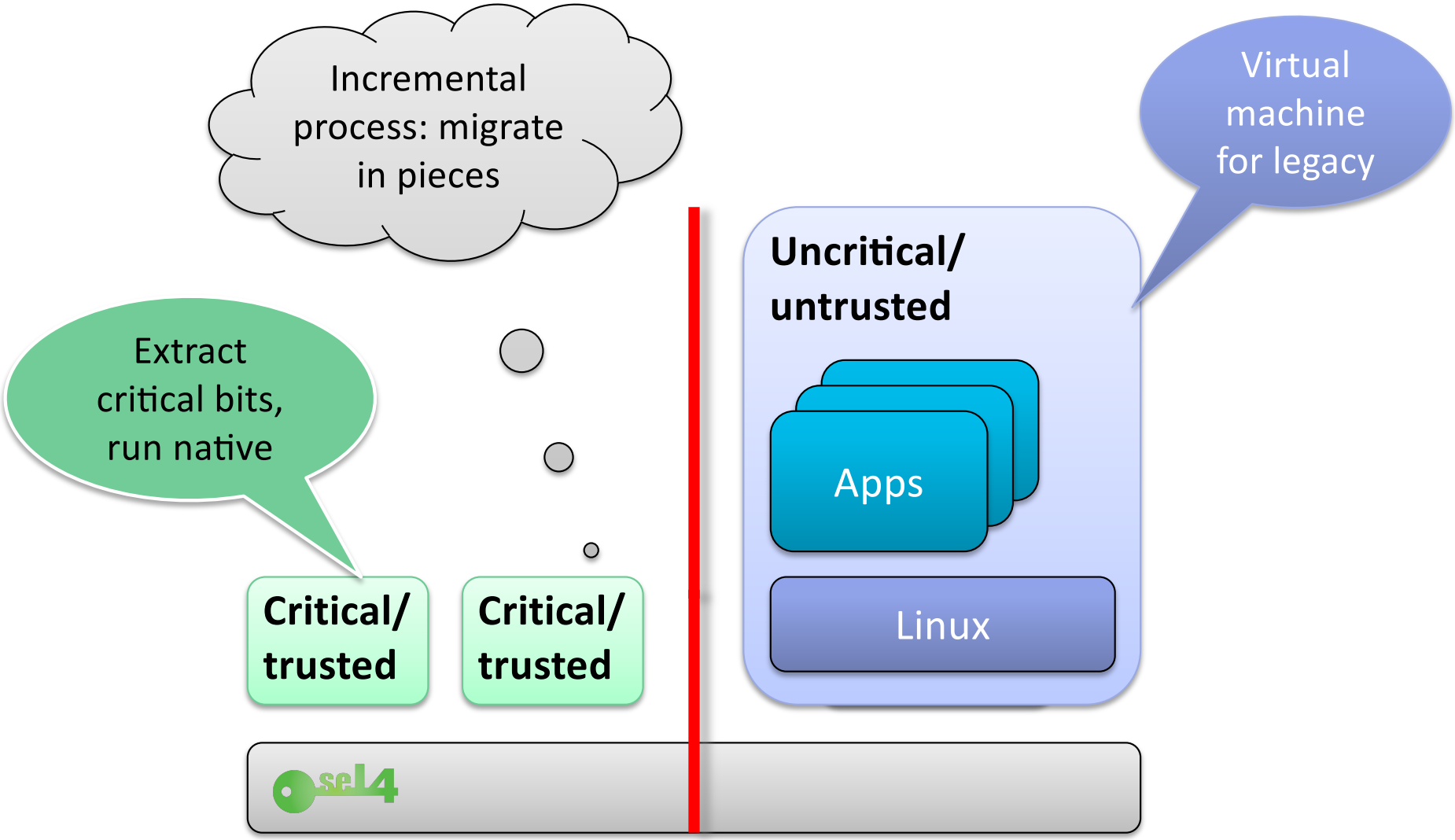
seL4 Isolation Goes Deep



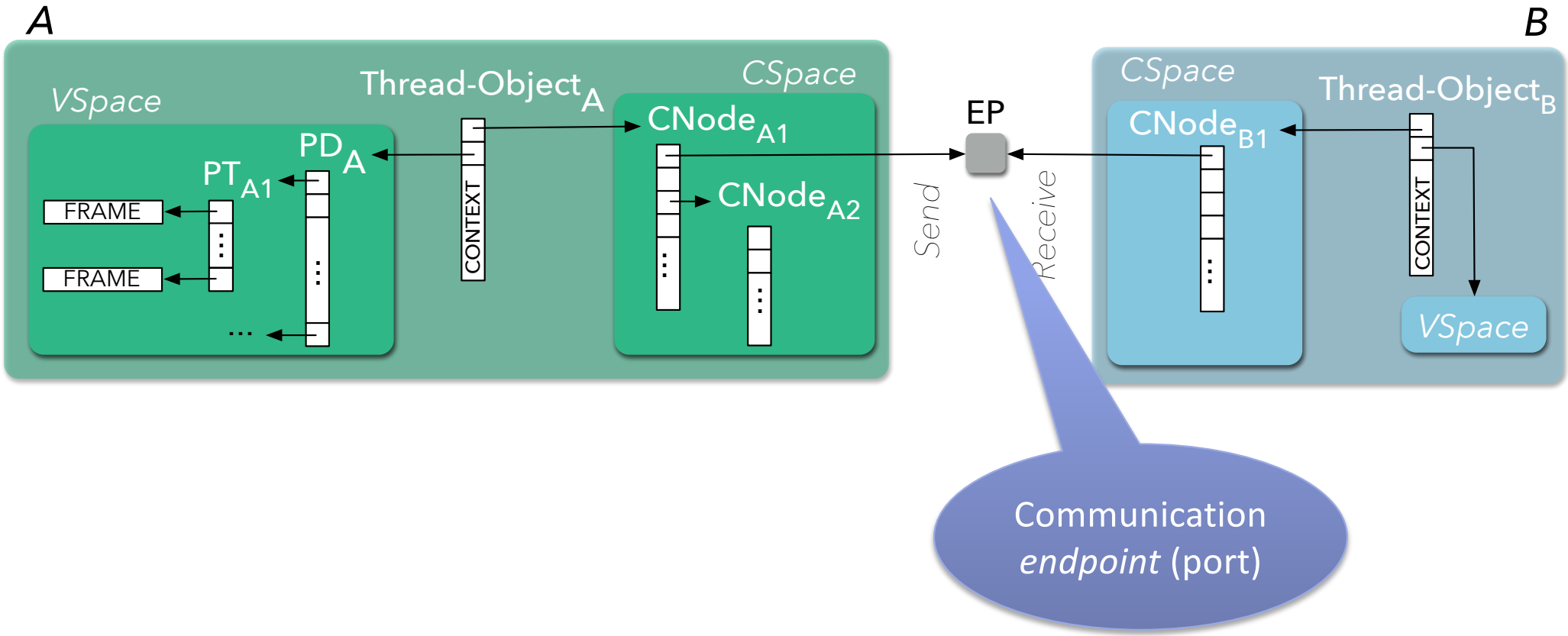


Building Trustworthy Systems

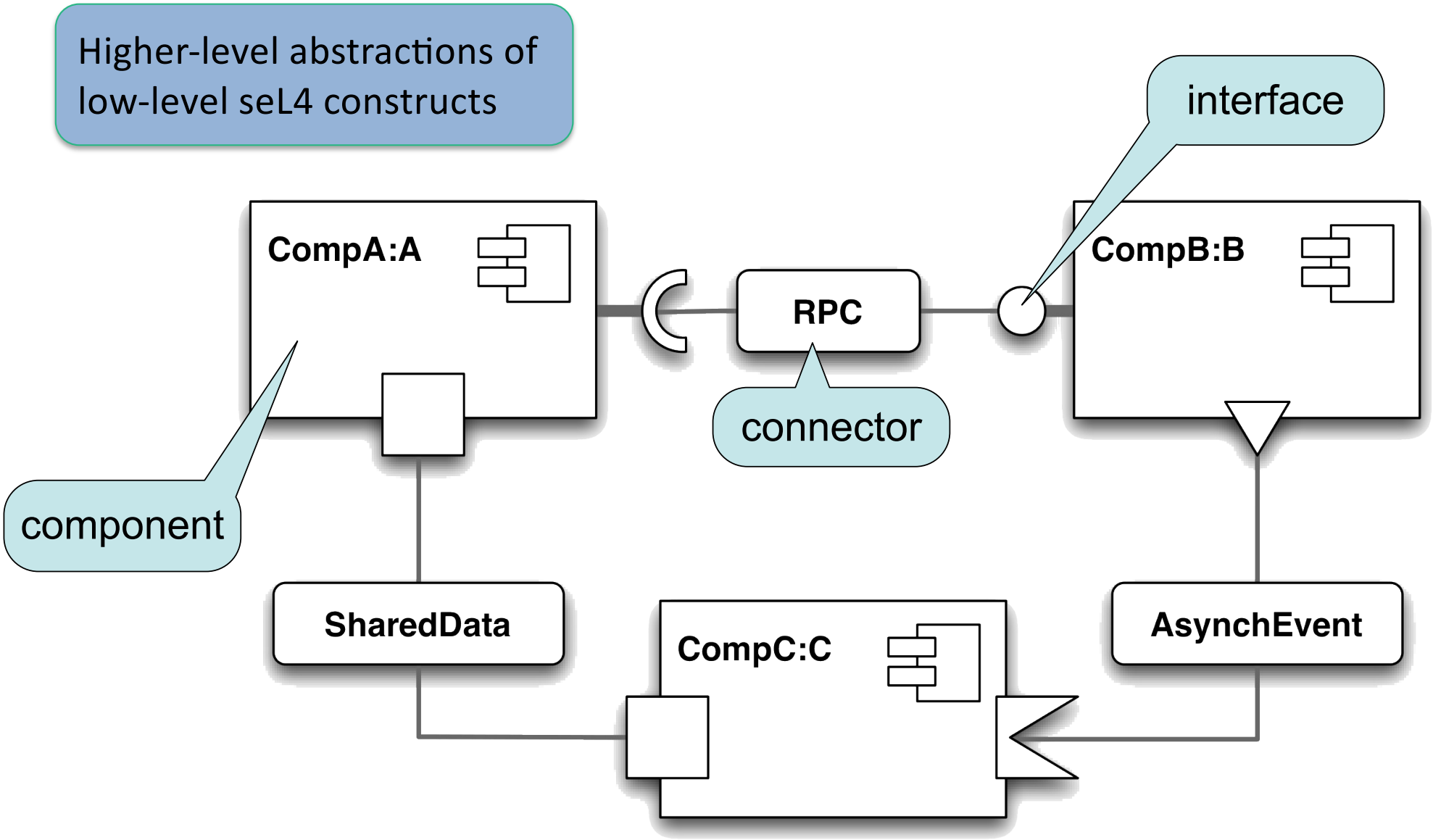
Security by Architecture



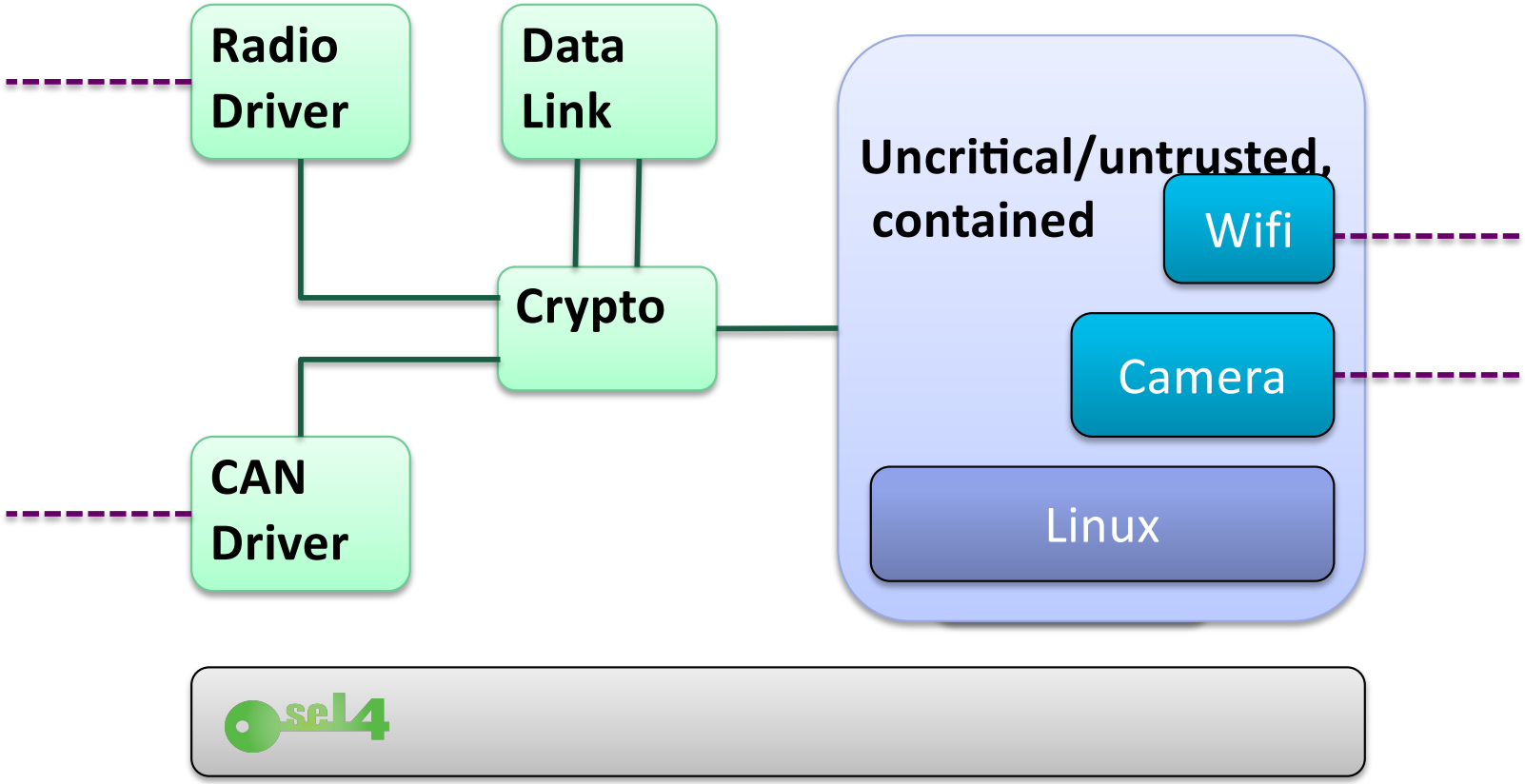
Example: Communicating Processes



Component Middleware: CAmkES

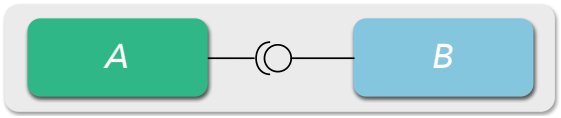


Example: Simplified HACMS UAV

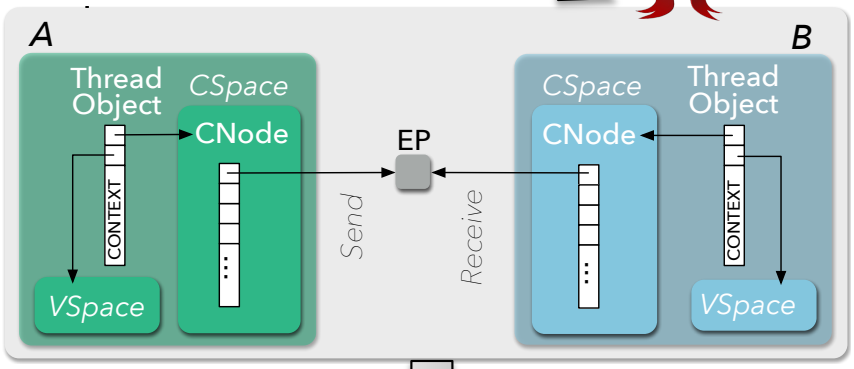


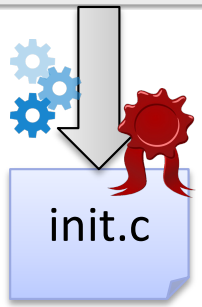
Automating the Abstraction

CAmkES: Architecture spec



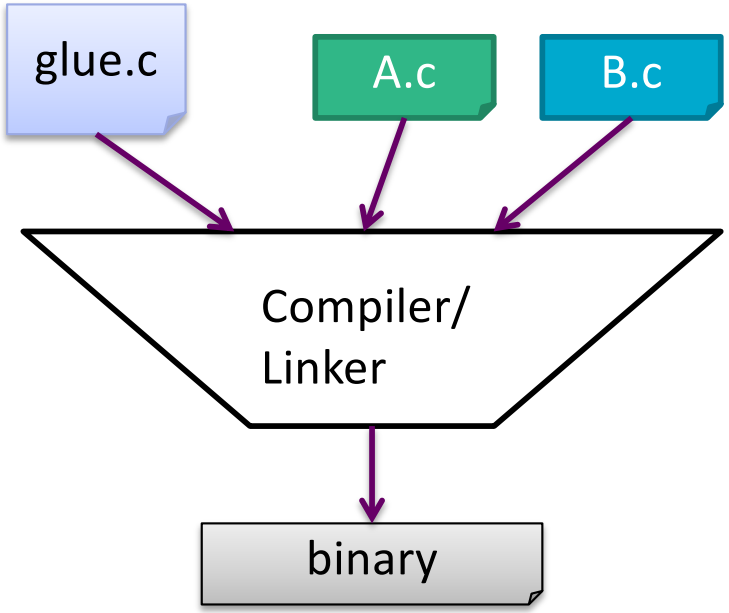
capDL: Authority spec





Limitations (at present):

- Glue code verification on subset of connectors
- Initialisation verified on model-level only



Thank you!