The Open-Source seL4 Kernel

Military-Grade Security Through Mathematics

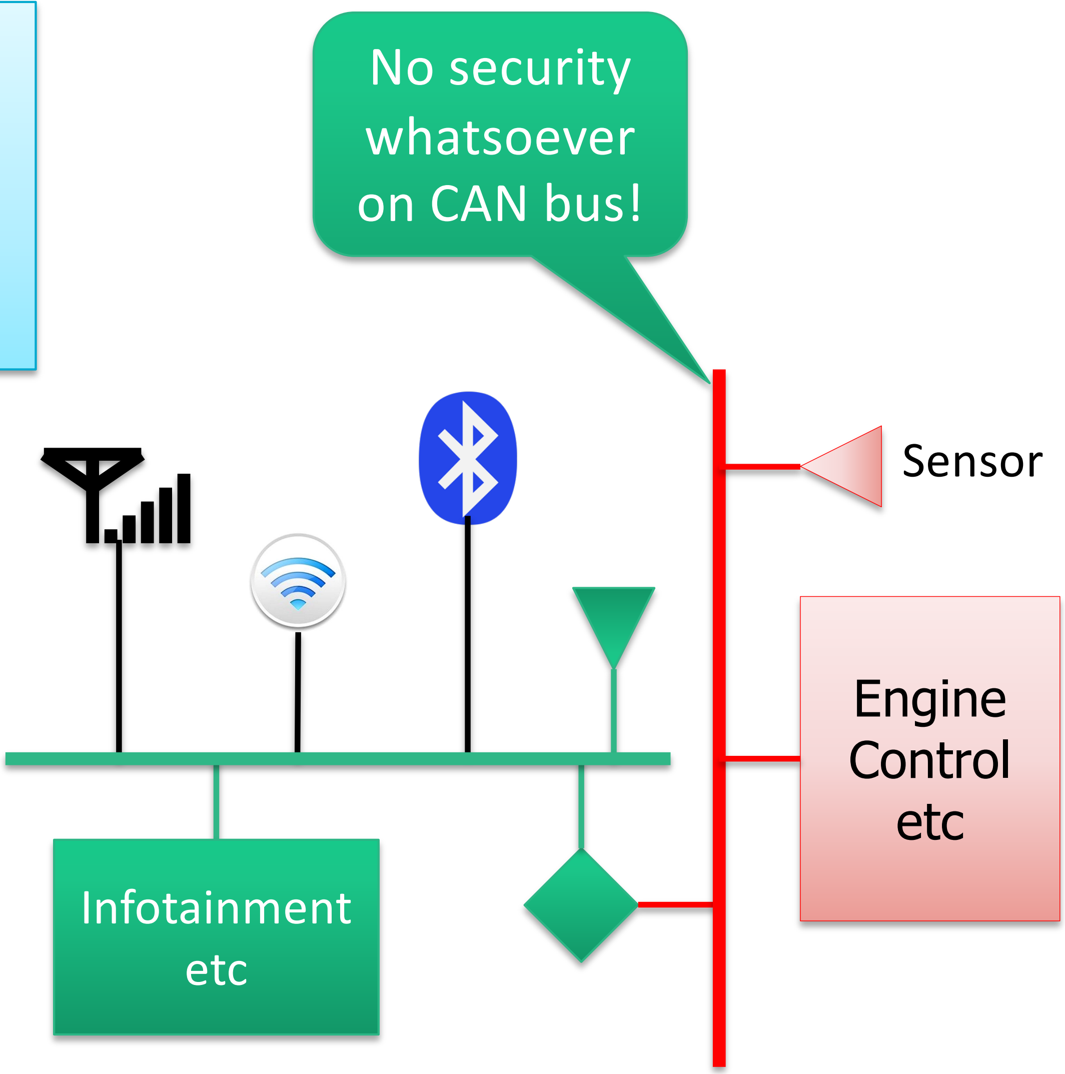
Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
Trustworthy Systems | Data61

Linaro Connect SFO’17

https://sel4.systems

Car Hacking – What's Behind?

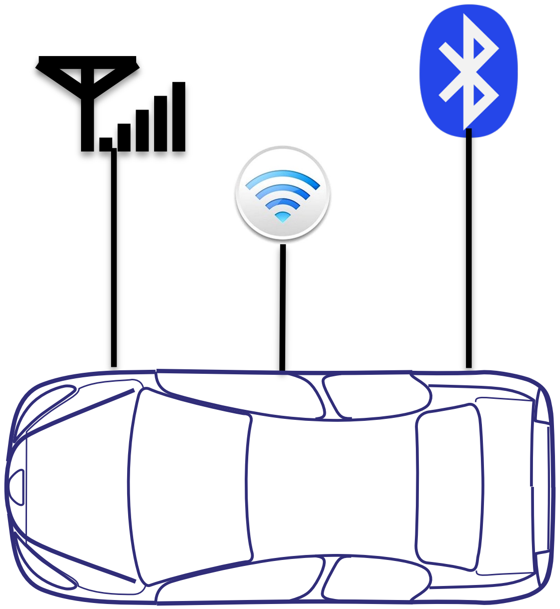
Networking for:
- Entertainment
- Connected car
- Safety (tire pressure…)
- Maintenance (OTA upgrades)



Challenge of Networking

Networking creates remote attack opportunities

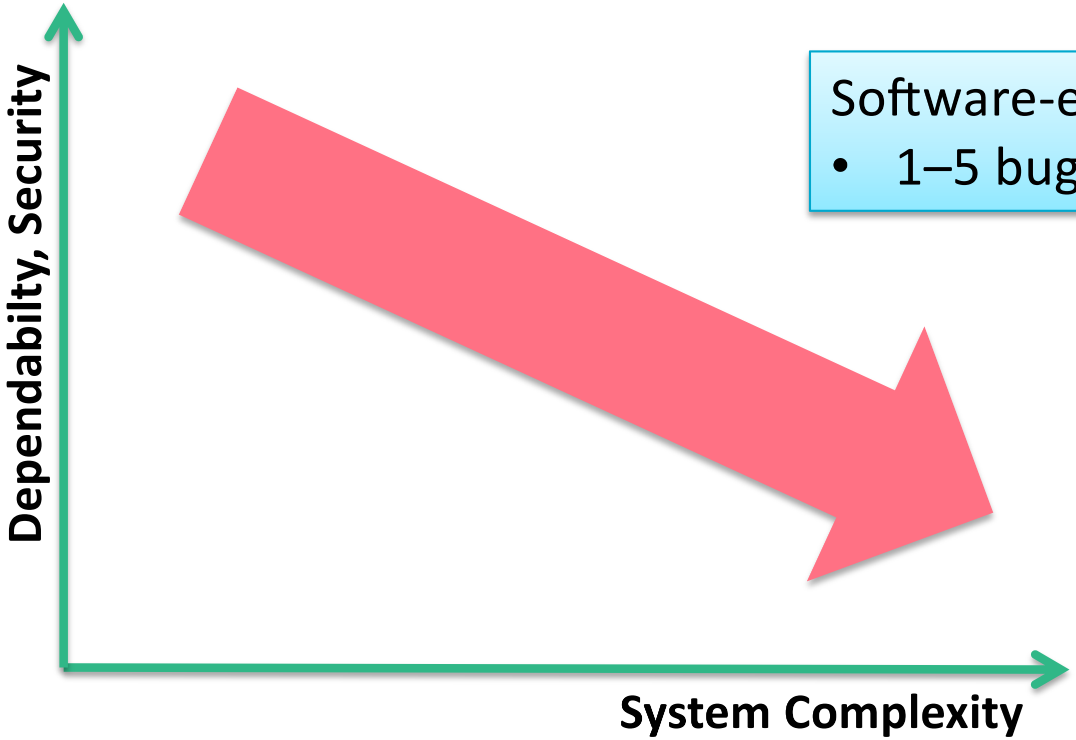
- from passengers (wifi, Bluetooth)
- from nearby cars (wifi, Bluetooth) – drive-by shooting, spread of viruses
- from anywhere (cellular)



Attack vectors:

- Insecure protocols
- Reusing crypto keys
- Software vulnerabilities

Software Vulnerabilities



Software-engineering rule of thumb:

- 1–5 bugs per 1,000 lines of ***quality*** code

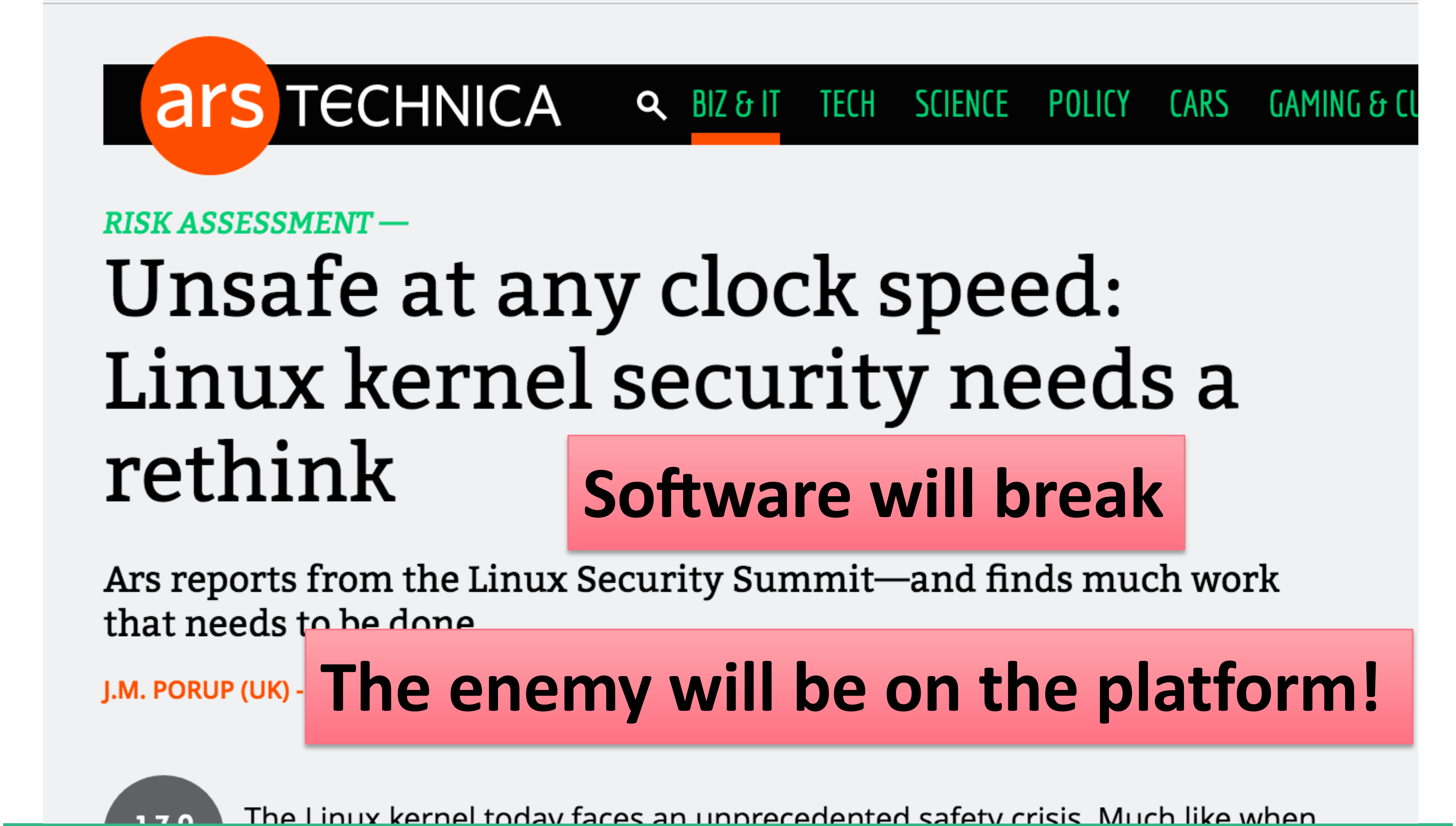
**Bluetooth protocol stack:
Multiple 100,000 lines**

**Linux kernel:
Tens of millions lines**

Complexity Drivers

- Features/functionality
- Legacy reuse

Linux “Security”



RISK ASSESSMENT —

Unsafe at any clock speed: Linux kernel security needs a rethink

Ars reports from the Linux Security Summit—and finds much work that needs to be done

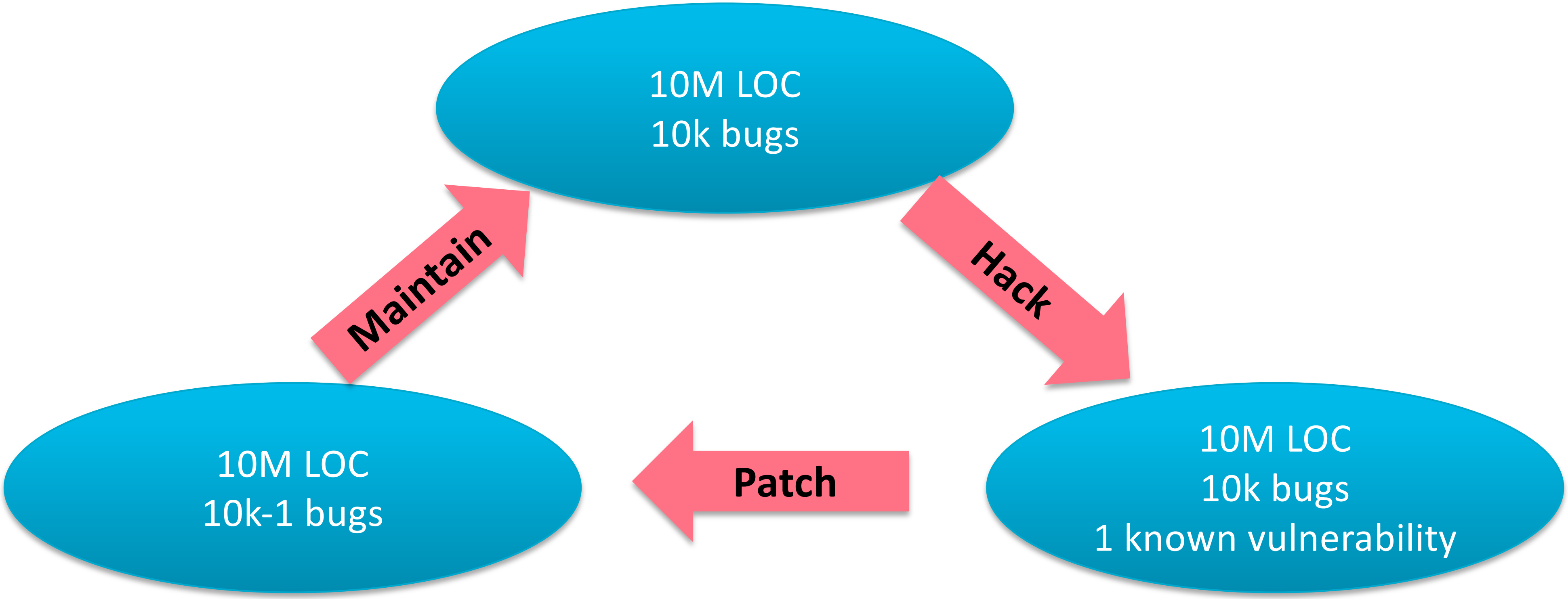
J.M. PORUP (UK) -

The Linux kernel today faces an unprecedented safety crisis. Much like when

Software will break

The enemy will be on the platform!

OK, So Let's Patch Regularly

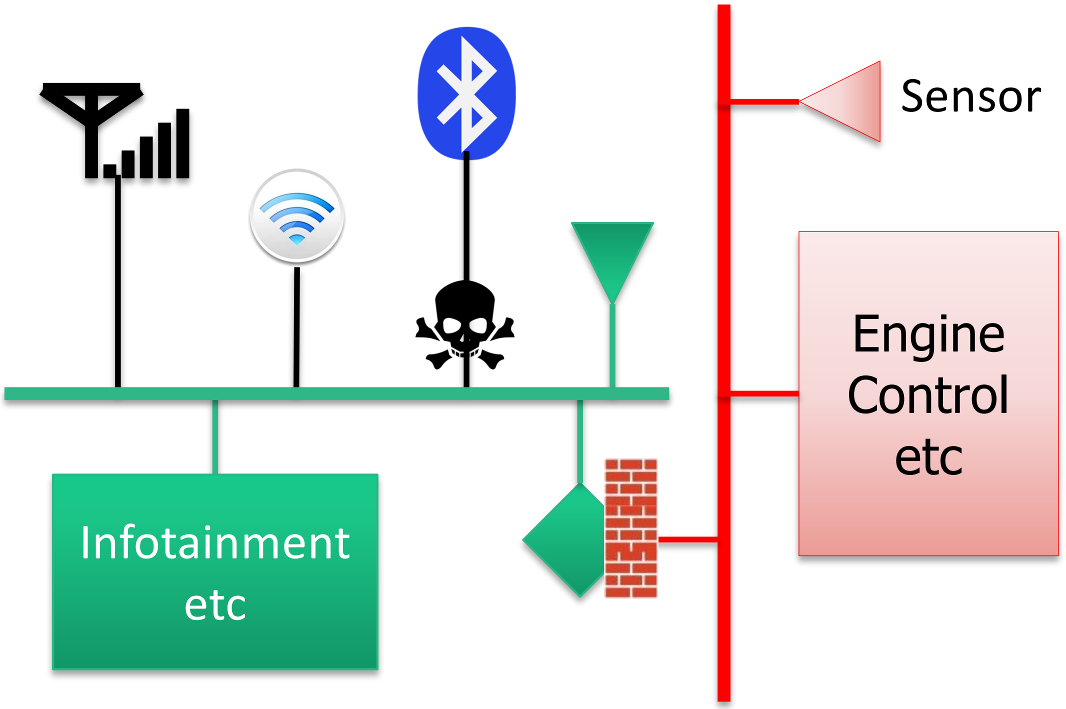


Patch-and-Pray: A losing proposition

So, Let’s Use Firewalls!

- Imposes overhead (SWaP) or
- Runs on vulnerable OS ⇒ worthless if OS compromised
- Even more code – may *increase* attack surface
- No help for valid messages that trigger bugs in software

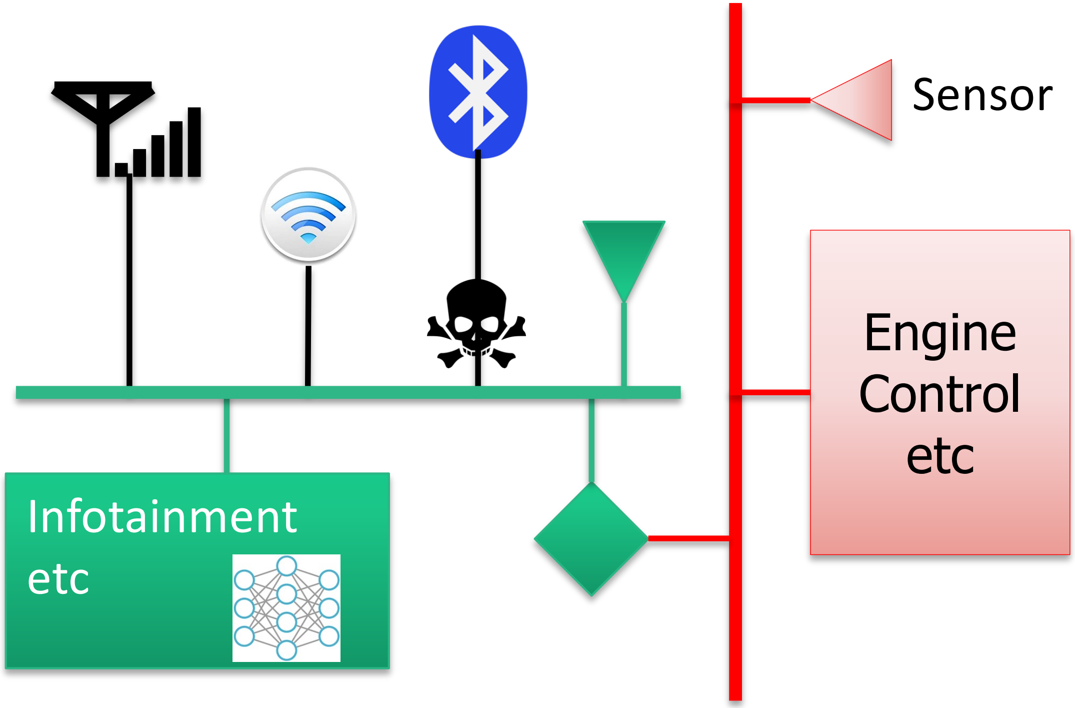
Firewalls treat symptoms,
not causes of problems!



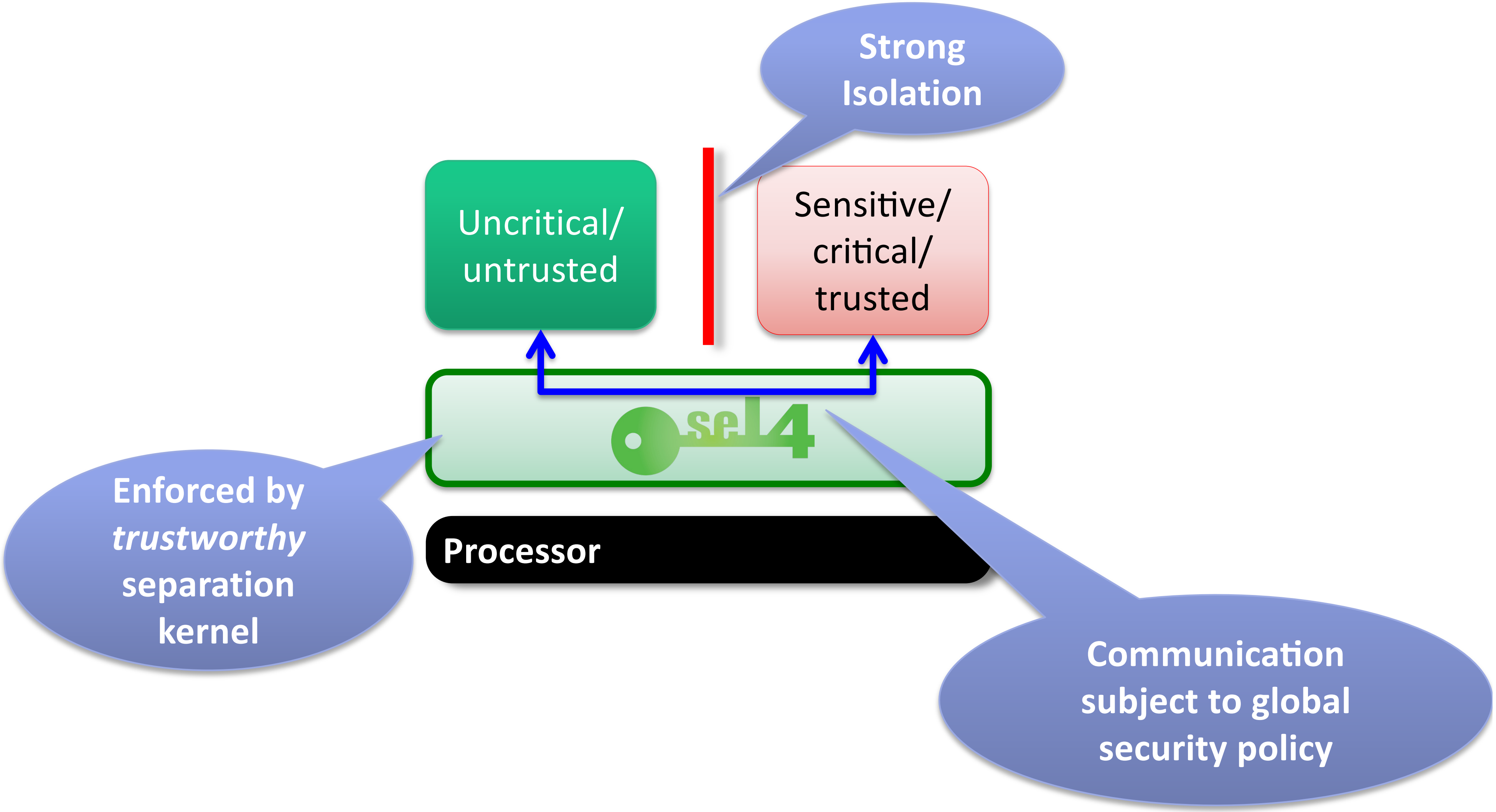
Let’s Use AI to Detect Compromise!

- Runs on vulnerable OS ⇒ worthless if OS compromised
- Even more code – may *increase* attack surface
- Can only detect that system is **already compromised**

Intrusion detection: admission of defeat



Fundamental Security Requirement: Isolation



Enforced by *trustworthy* separation kernel

Communication subject to global security policy

Trustworthiness: Can We Rely on Isolation?

A system is **trustworthy** if and only if:

- it behaves **exactly** as it is specified,
- in a **timely** manner,
- while ensuring **secure** execution

***Claim*:**

A system must be considered *untrustworthy* unless *proved* otherwise!

Corollary [with apologies to Dijkstra]:

Testing, code inspection, etc. can only show *lack of trustworthiness*!

seL4 Provably Secure Operating System

~10,000 lines of C and ASM code

⬇

Small attack surface,

Amenable to full verification

Non-kernel code can only access resources and communication channels if explicitly authorised with a per-object access token (capability)

□

- **Confined damage**
- **Least privilege**

Small,

fast,

capability-based,

operating system kernel

World's fastest (5–10X faster) operating system designed for security and safety

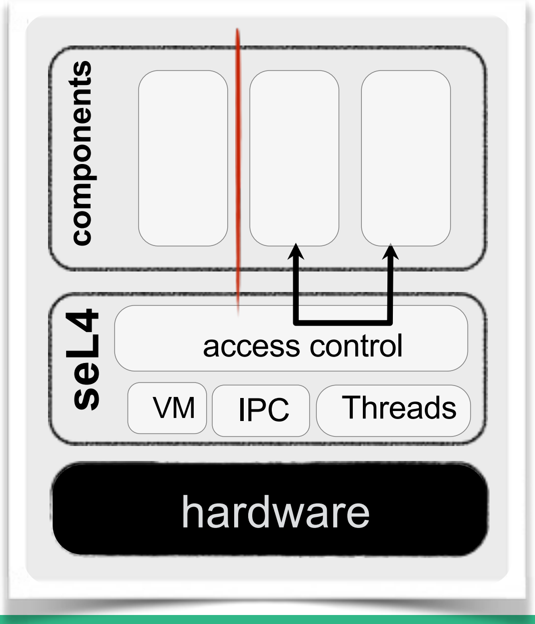
⬇

Suitable for real-world deployment

Code that runs in privileged mode of the hardware

□

Most critical part

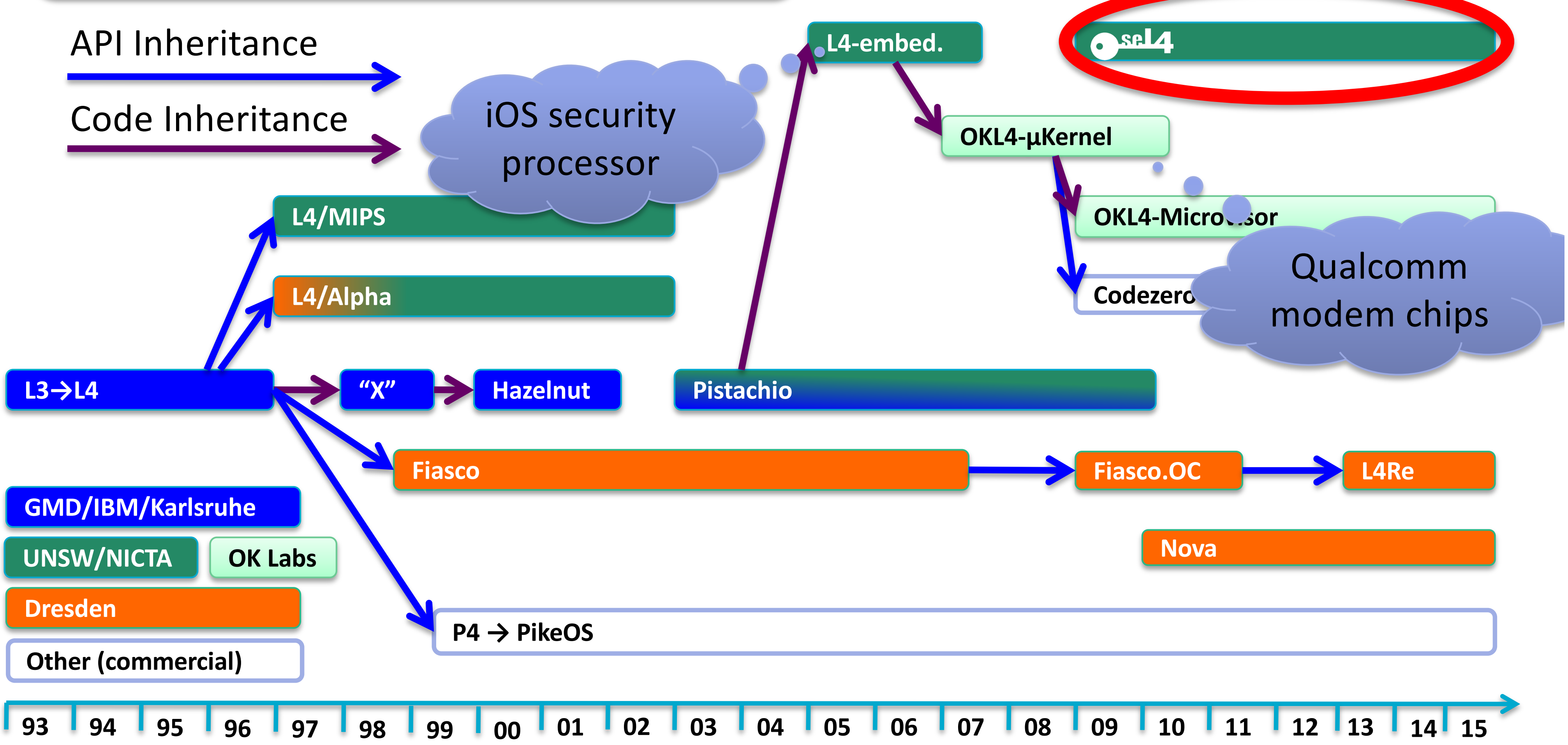


Unprivileged mode

Privileged mode

20+ Years of L4 Microkernel R&D

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



API Inheritance

Code Inheritance

iOS security processor

Qualcomm modem chips

L3→L4

L4/MIPS

L4/Alpha

"X"

Hazelnut

Pistachio

L4-embed.

seL4

OKL4-µKernel

OKL4-Micro..sor

Codezero

Fiasco

Fiasco.OC

L4Re

Nova

P4 → PikeOS

GMD/IBM/Karlsruhe

UNSW/NICTA

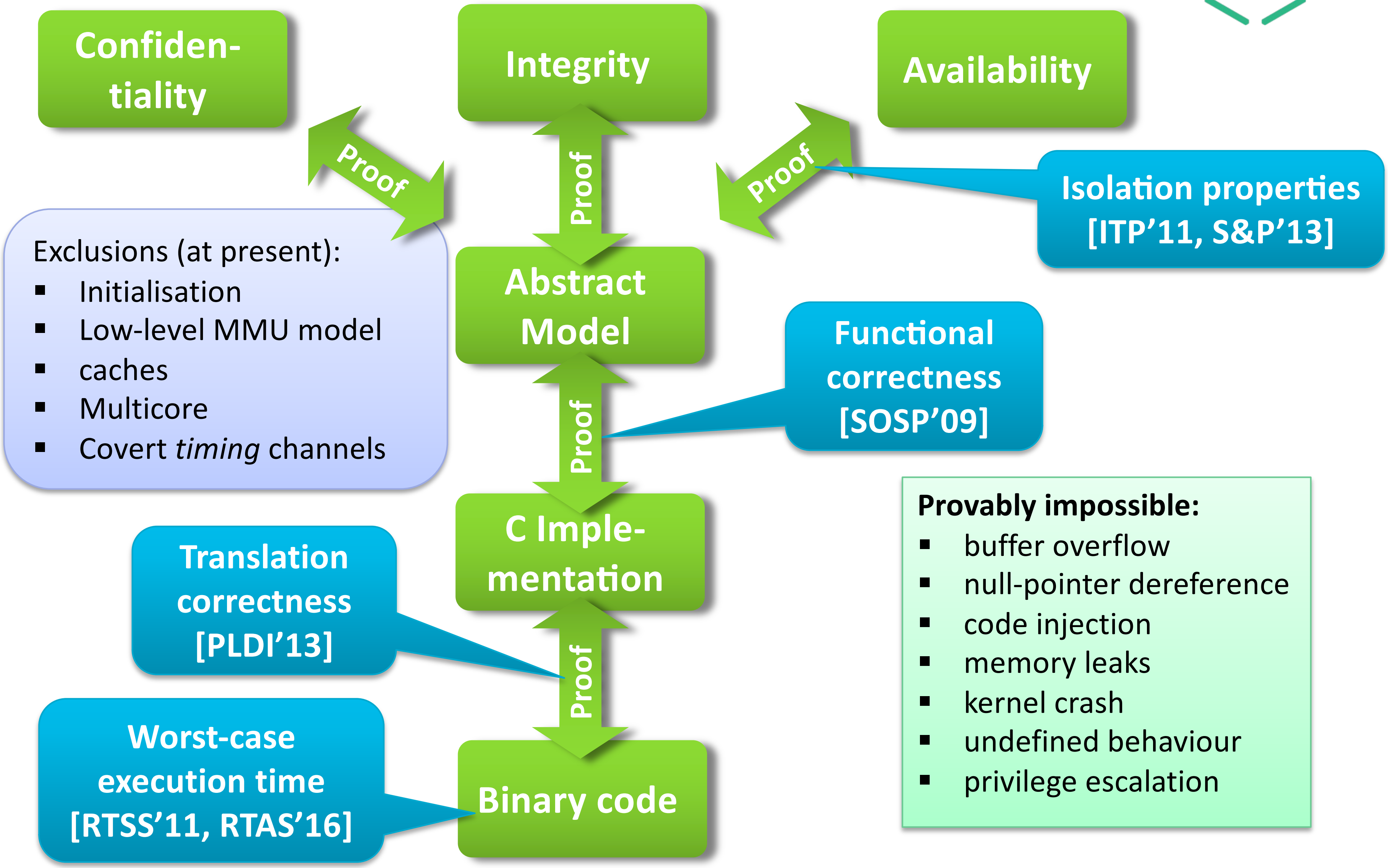
OK Labs

Dresden

Other (commercial)

93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

Proving Trustworthiness of seL4


Confiden-tiality
Integrity
Availability
Proof
Proof
Proof
Isolation properties [ITP’11, S&P’13]
Abstract Model
Proof
Functional correctness [SOSP’09]
C Imple-mentation
Proof
Translation correctness [PLDI’13]
Binary code
Worst-case execution time [RTSS’11, RTAS’16]

Exclusions (at present):

- Initialisation
- Low-level MMU model
- caches
- Multicore
- Covert *timing* channels

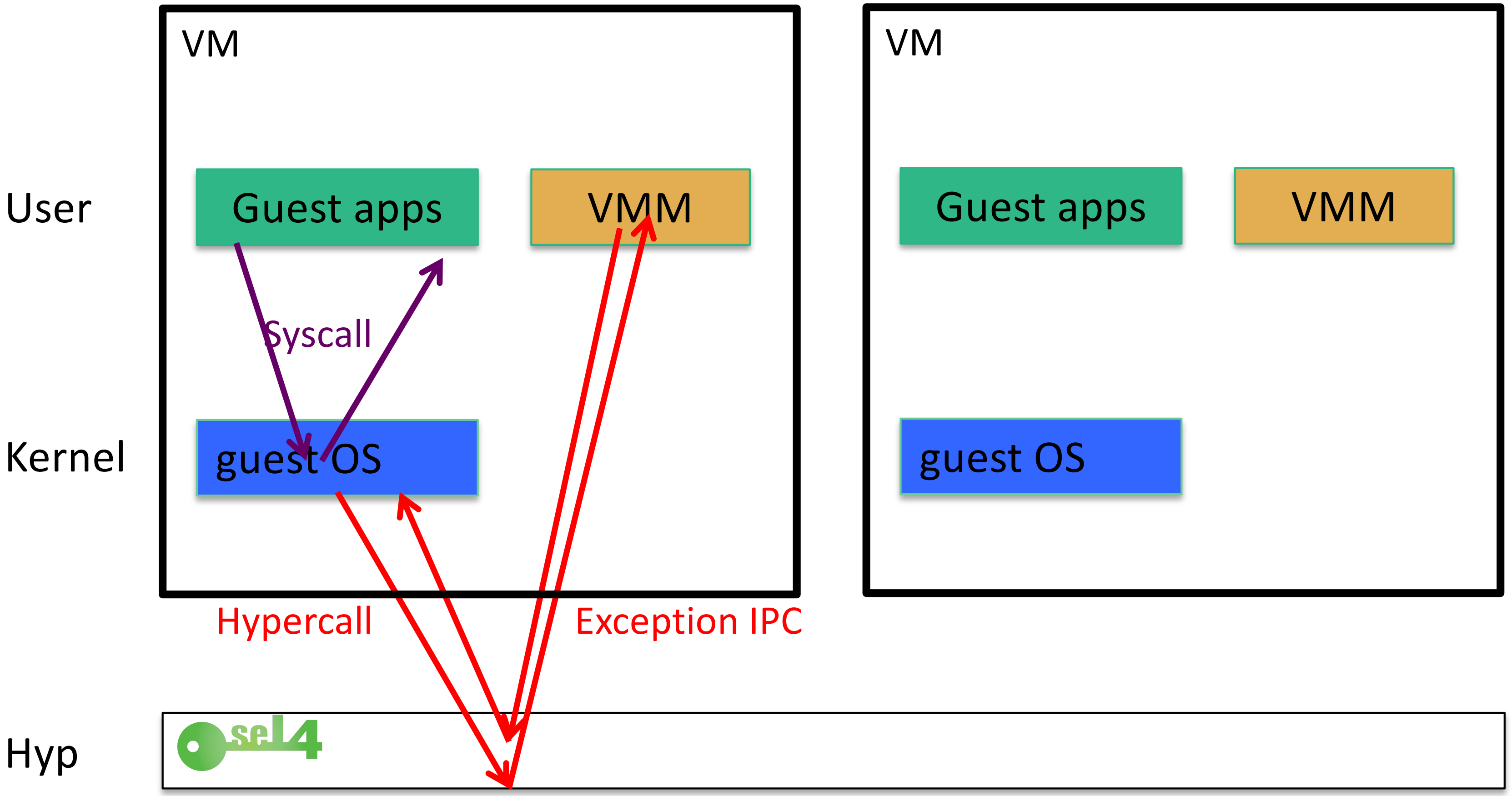
Provably impossible:

- buffer overflow
- null-pointer dereference
- code injection
- memory leaks
- kernel crash
- undefined behaviour
- privilege escalation

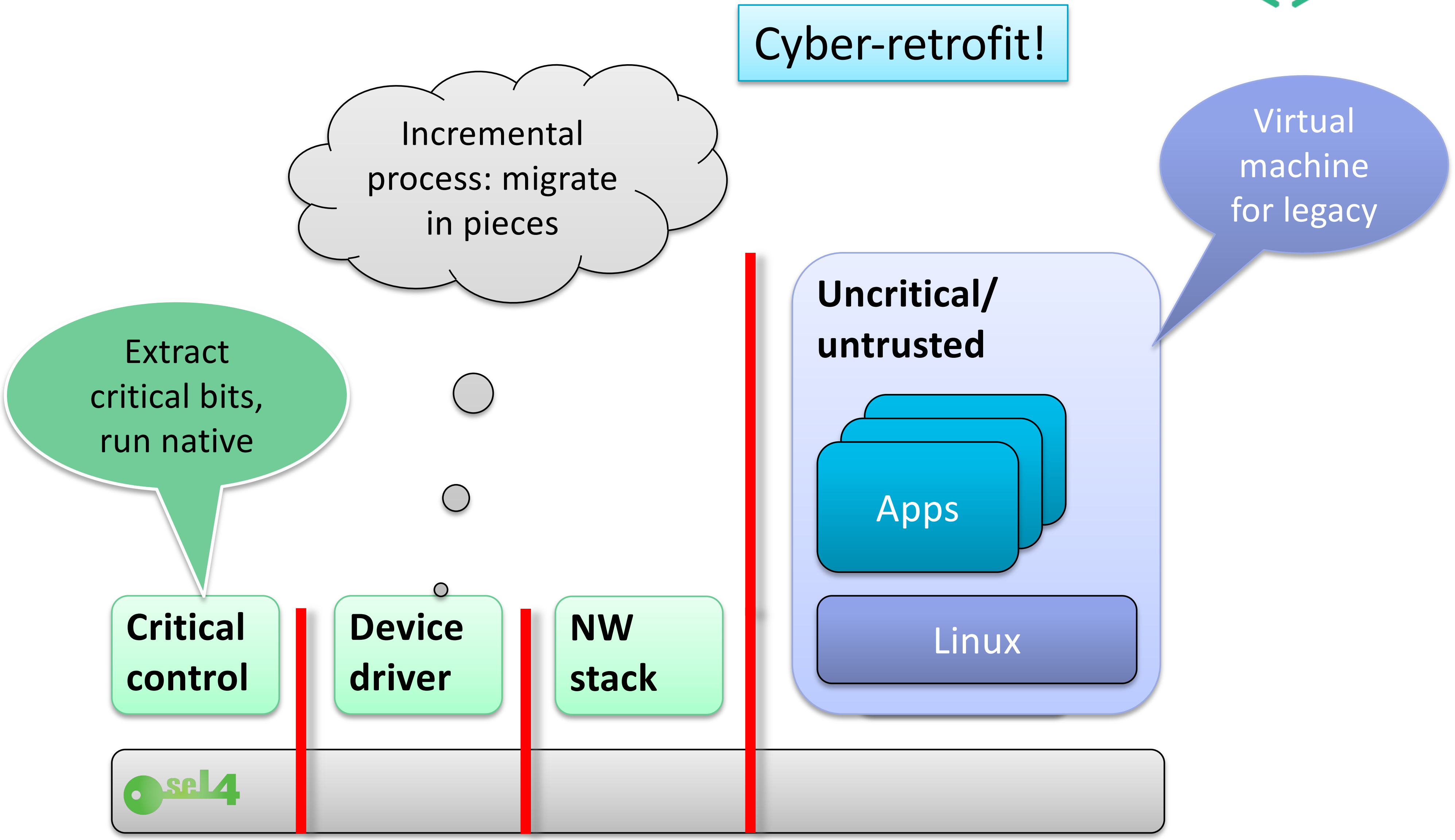
How Does seL4 Compare?

Feature	seL4	Other hypervisors, RTOSes, separation kernels
Performance	Fastest	2–10 × slower
Functional correctness	Proved	No Guarantee
Isolation	Proved	No Guarantee
Worst-case latency bounds	Sound & complete	Estimates only
Storage channel freedom	Proved	No Guarantee
Timing channel prevention	Low overhead	None or High Overhead
Mixed-criticality support	Fully supported, high utilisation	Limited, resource-wastive

Virtualisation



Security by Architecture


Cyber-retrofit!
Incremental process: migrate in pieces
Virtual machine for legacy
Extract critical bits, run native
Uncritical/ untrusted
Apps
Linux
Critical control
Device driver
NW stack
seL4

Real-World Example: DARPA HACMS

Retrofit existing system!

Boeing Unmanned Little Bird

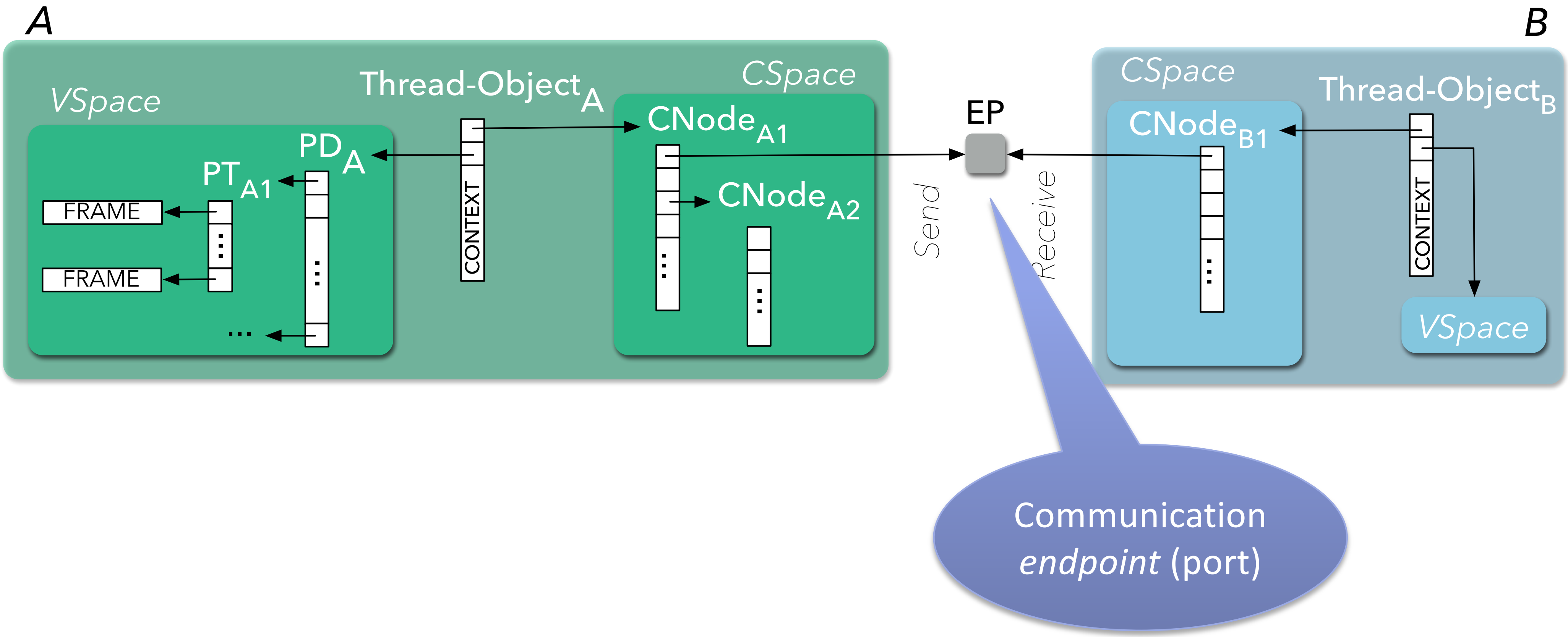
US Army Autonomous Trucks

SMACCMcopter Research Vehicle

Develop technology

TARDEC GVR-Bot

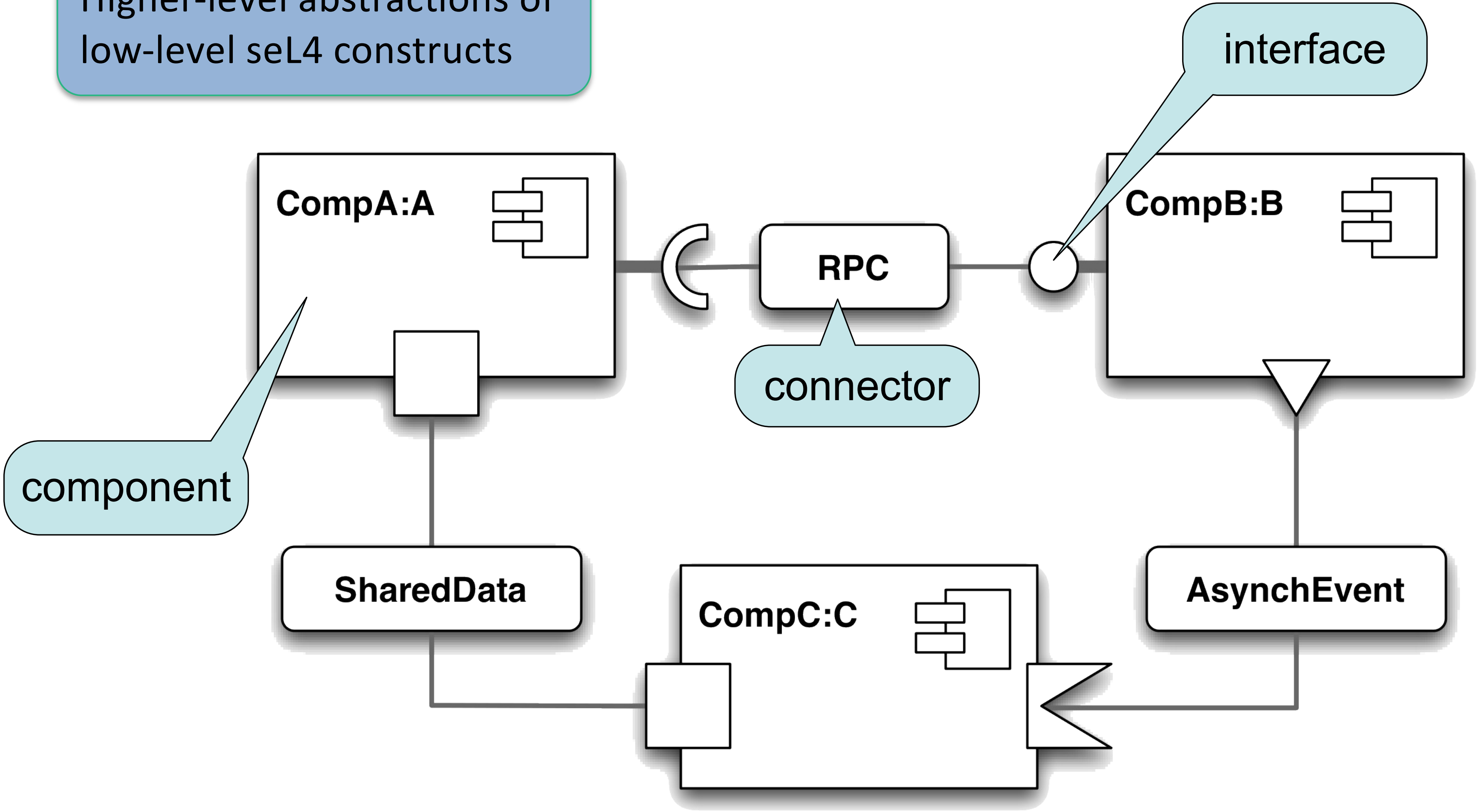
Example: Communicating Processes



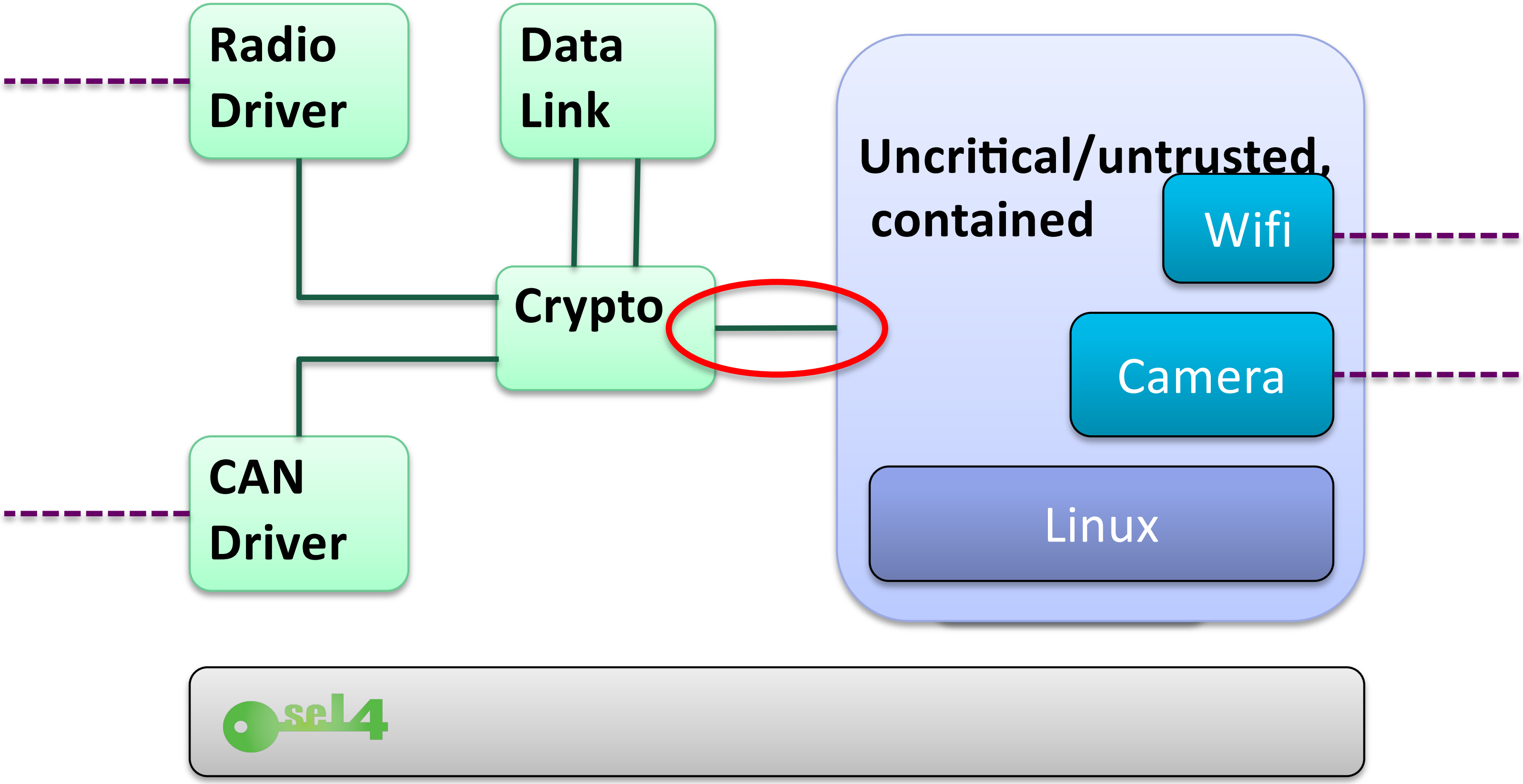
Communication *endpoint* (port)

Component Middleware: CAmkES

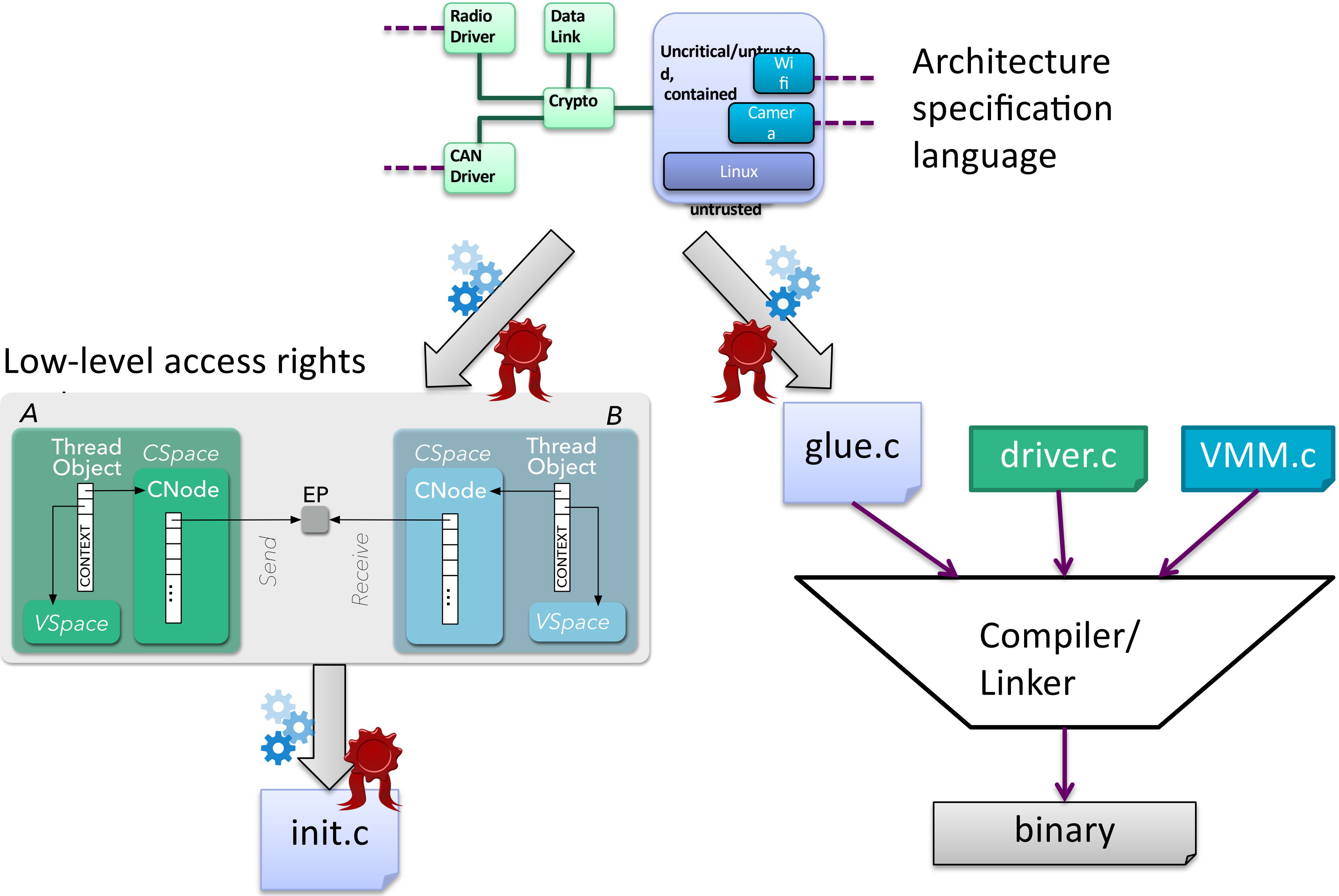
Higher-level abstractions of low-level seL4 constructs



Example: Simplified HACMS UAV



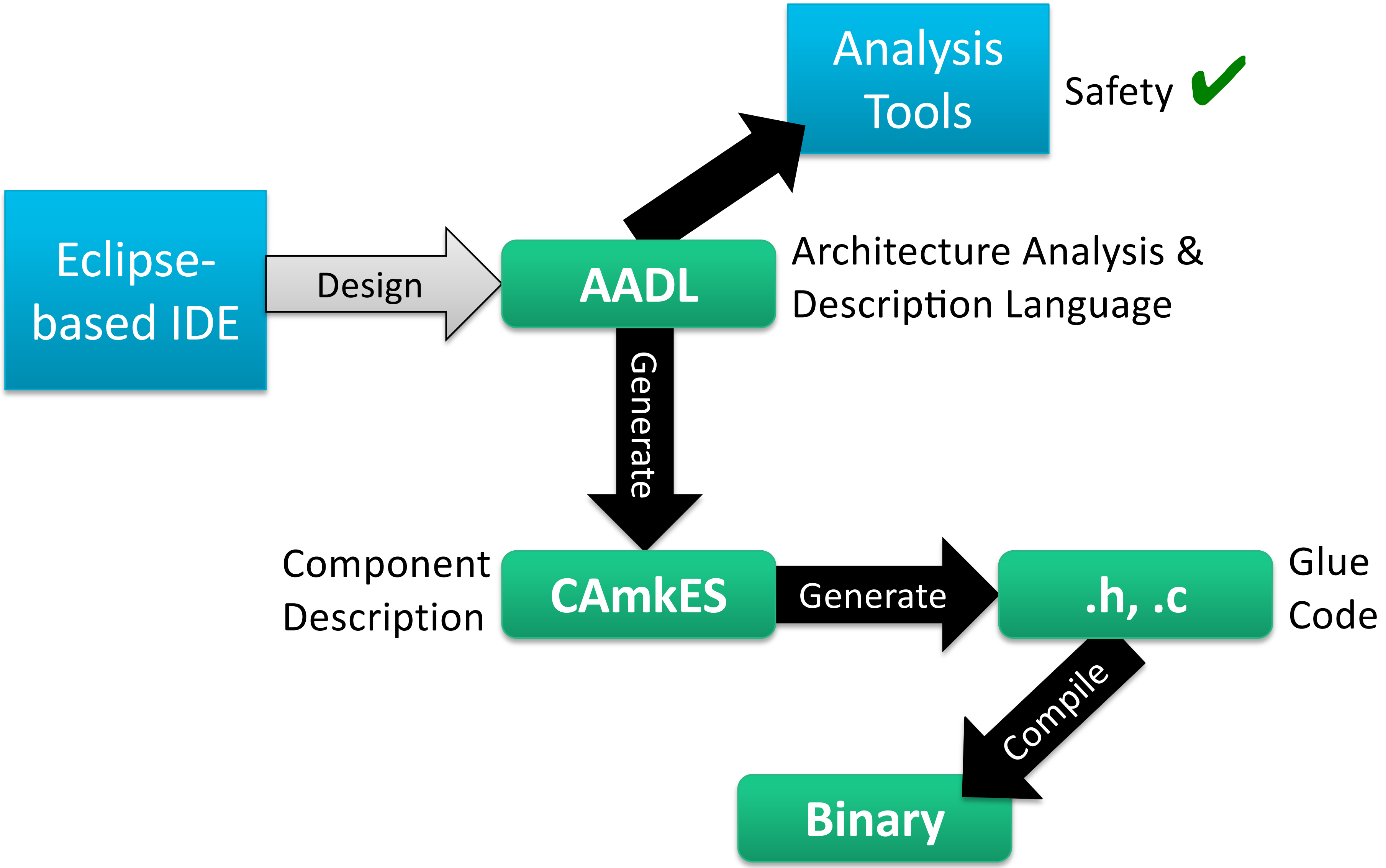
Enforcing the Architecture



Architecture specification language

Low-level access rights

Open-Source Architecture Analysis



Safety ✓

Architecture Analysis & Description Language

Component Description

Glue Code

Military-Grade Security

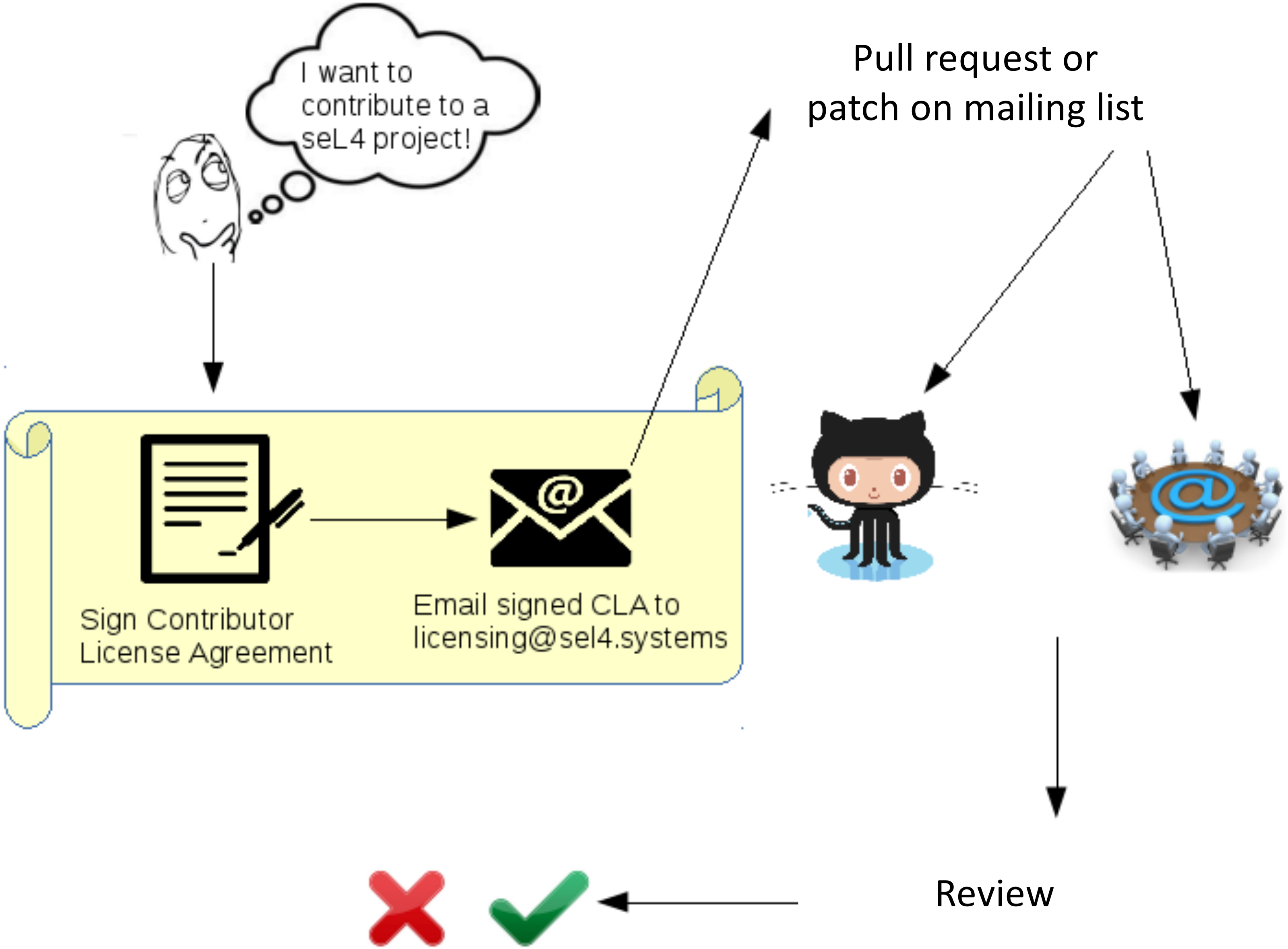
Cross-Domain Desktop Compositor

Multi-level secure terminal

- Successful defence trial in AU
- Evaluated in US, UK, CA
- Formal security evaluation soon

Pen10.com.au crypto communication device undergoing formal security evaluation in UK

Contributions



I want to contribute to a seL4 project!

Sign Contributor License Agreement

Email signed CLA to licensing@sel4.systems

Pull request or patch on mailing list

Review

Thank you

Robin Randhawa

Please check out https://sel4.systems

Military-Grade Security for You!

Security is no excuse for poor performance!

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

Linaro Connect SFO’17

http://sel4.systems