School of Computer Science & Engineering

Trustworthy Systems Group

Can We Make Trustworthy Systems a Reality?

Gernot Heiser
UNSW Sydney & seL4 Foundation
@GernotHeiser

Cyberattacks Are Everywhere

BITSIGHT

Report Shows Cyber Attacks on Cloud Services Have Doubled

Cyberattacks on Automated Vehicles Rise by 99%: Report

By **CISOMAG** - June 9, 2020

Cyber Attacks That Target Electrical Devices and Equipment: What Engineers Should Know

February 10, 2020 by Ikimi .O

News / World

'Most serious cyberattack of the Ukraine war': Tens of thousands modems crippled

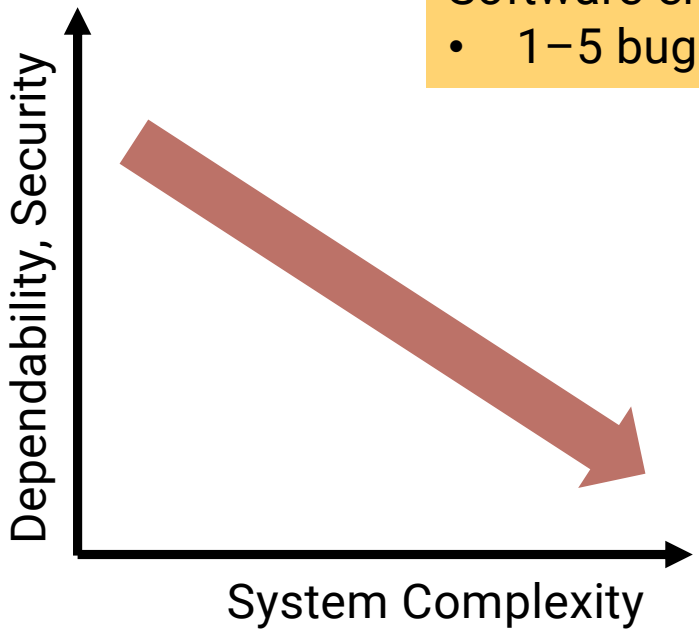
AP By Associated Press | 5:38pm Mar 31, 2022

Increasingly used by
- organised crime
- state actors

NEWS | February 7, 2022

Ransomware attack on Swissport causes delay at Zurich Airport

Core Problem: Complexity

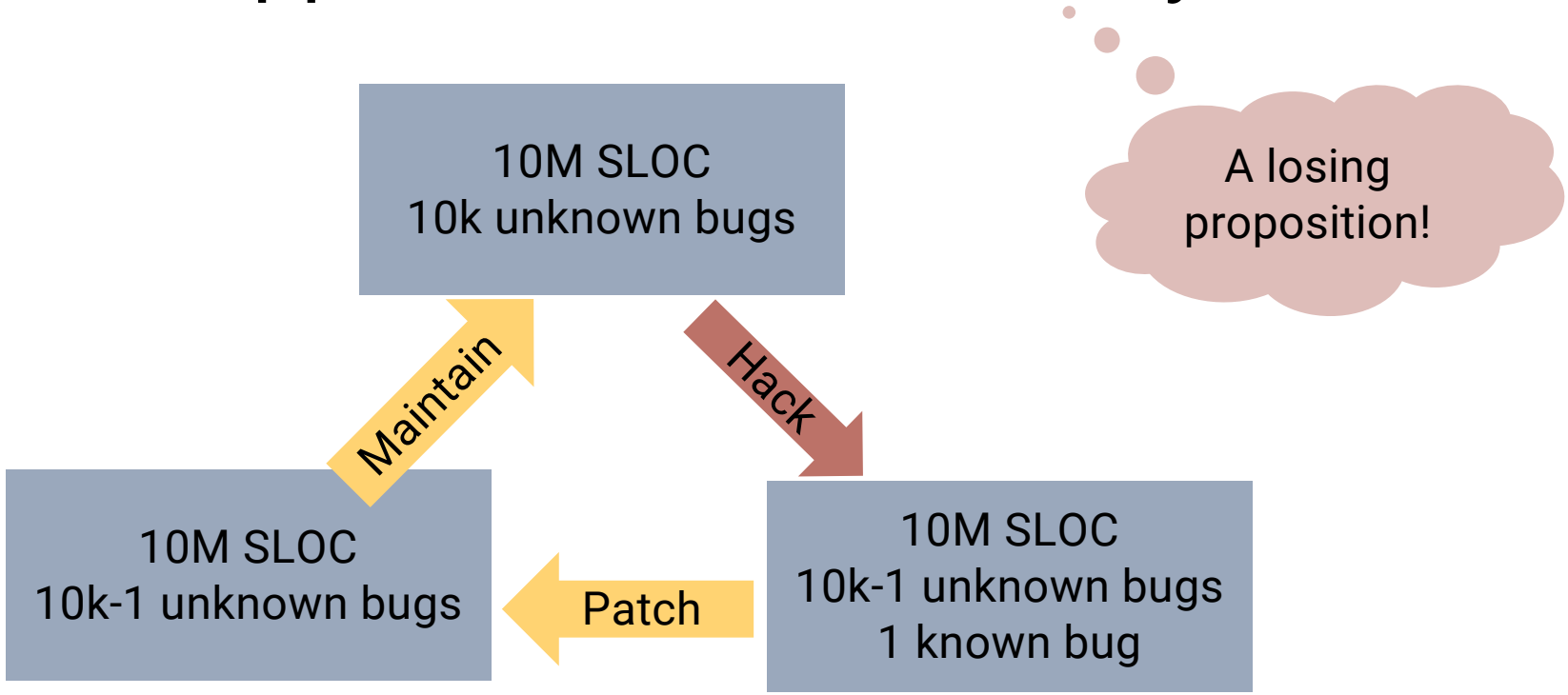


Software-engineering rule of thumb:
- 1–5 bugs per 1,000 lines of quality code

Bluetooth protocol stack:
100s kSLOC

Linux/Windows kernel:
10s MSLOC

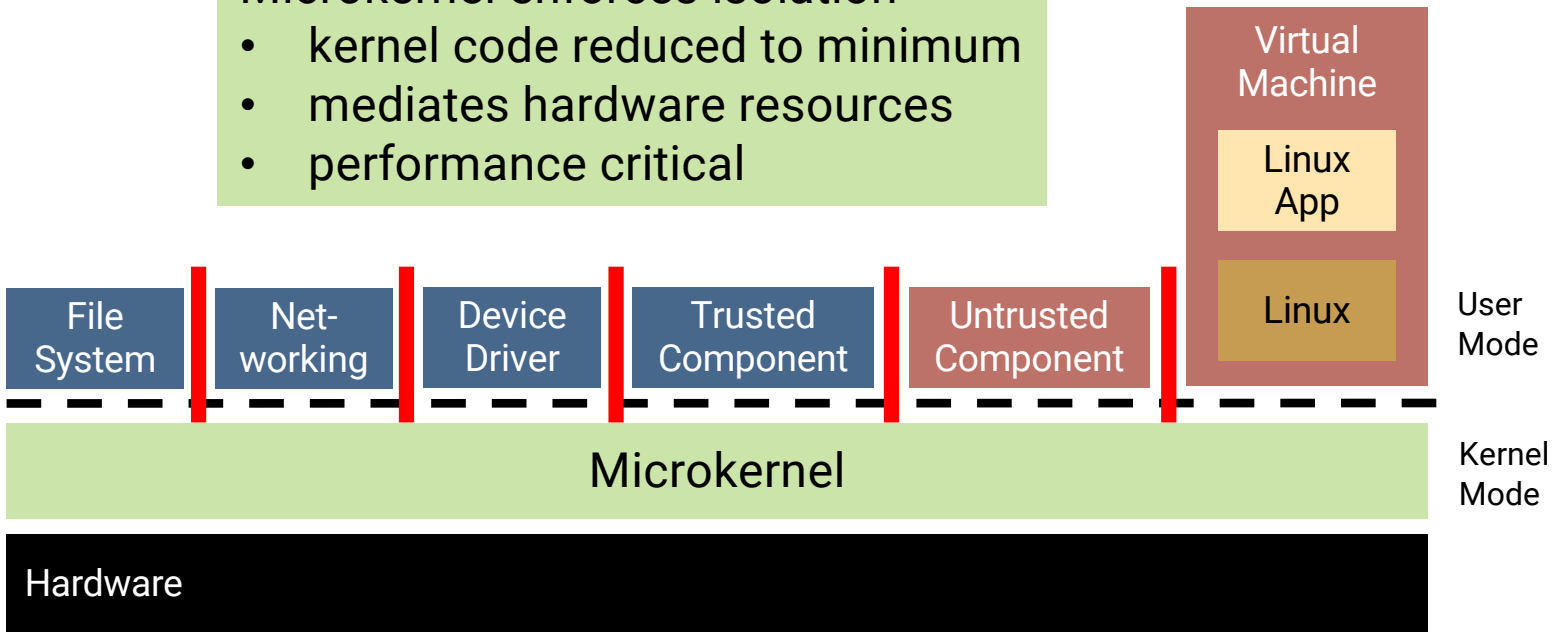
Standard Approach: Patch-and-Pray



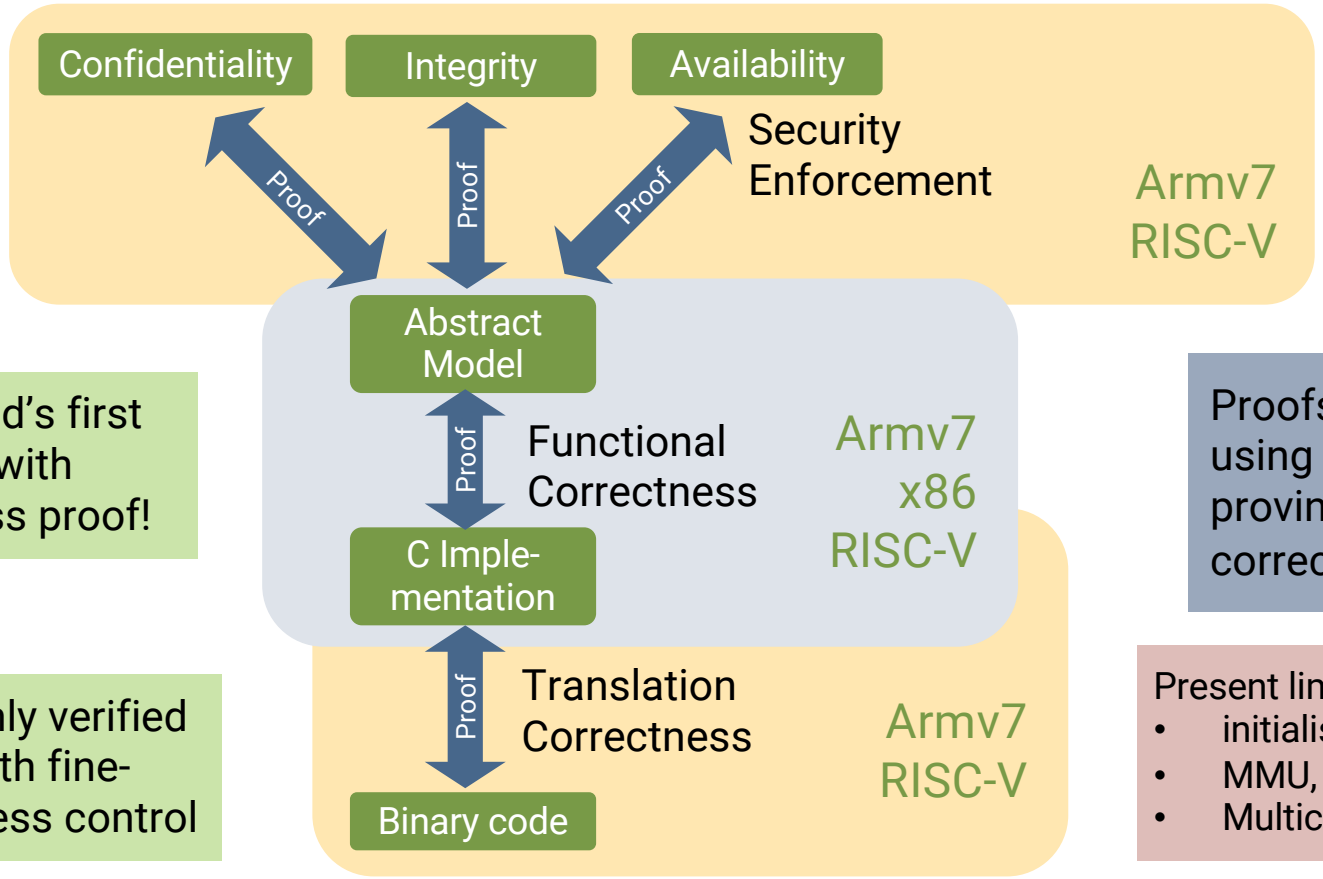
Solution 1: Minimise Trusted Computing Base

Modularisation: Separate components
- operating-system services
- applications

Microkernel enforces isolation
- kernel code reduced to minimum
- mediates hardware resources
- performance critical



Solution 2: Mathematical Proof



AArch64 in progress

seL4: World's first OS kernel with correctness proof!

Proofs are machine-checked, using interactive theorem proving (translation correctness fully automated)

seL4: Still only verified OS kernel with fine-grained access control

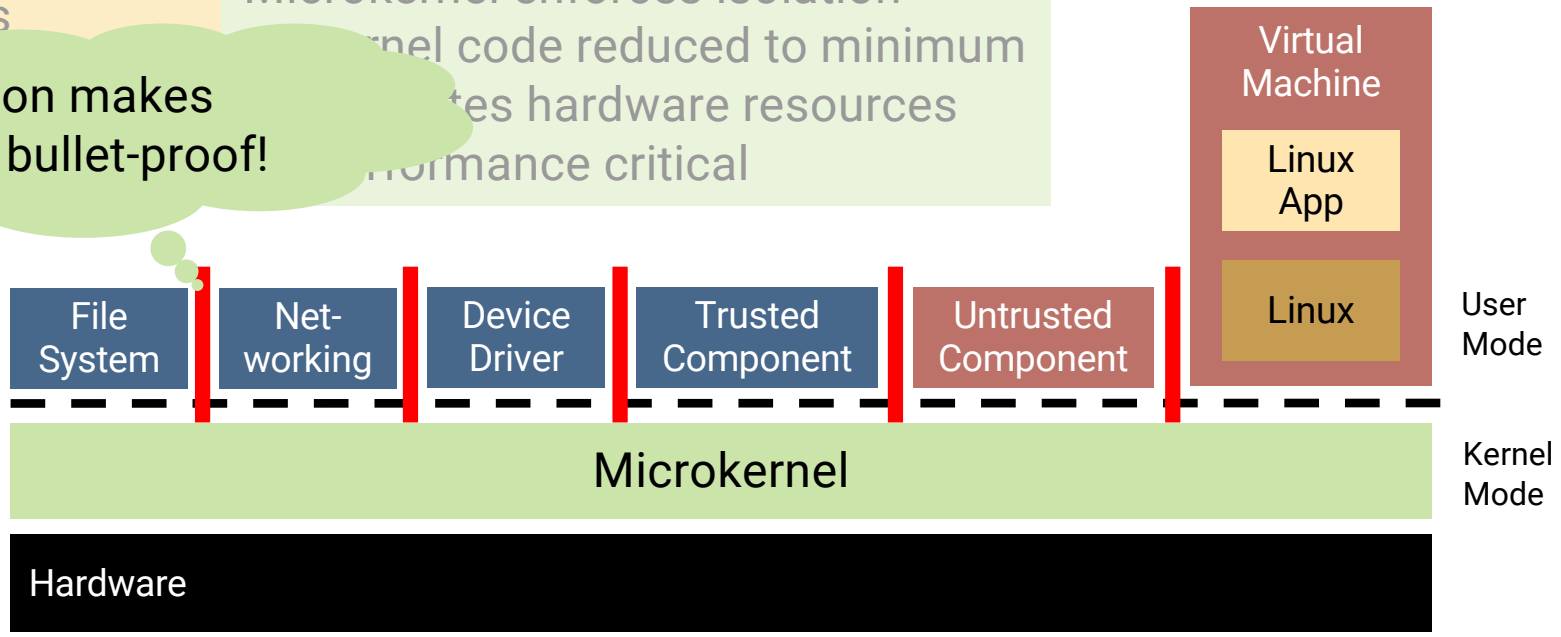
Present limitations
- initialisation code not verified
- MMU, caches modelled abstractly
- Multicore not yet verified

Solution 1: Minimise Trusted Computing Base

Modularisation: Separate components
- operating-system servi
- applications

Microkernel enforces isolation
nel code reduced to minimum
es hardware resources
rmance critical

Verification makes isolation bullet-proof!



Security Is No Excuse For Bad Performance!

Latency (in cycles) of a round-trip cross-address-space IPC on x64

Source	seL4	Fiasco.OC	Zircon
Mi et al, 2019	986	2717	8157
Gu et al, 2020	1450	3057	8151
seL4.systems, Jun'22	767	N/A	N/A

World's fastest microkernel!

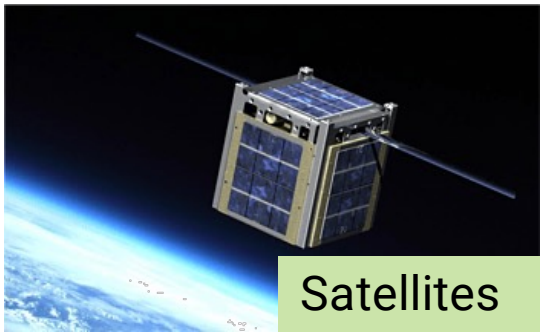
Within 10% of hardware limit!

Sources:

- Zeyu Mi, Dingji Li, Zihan Yang, Xinran Wang, Haibo Chen: "SkyBridge: Fast and Secure Inter-Process Communication for Microkernels", EuroSys, April 2020
- Jinyu Gu, Xinyue Wu, Wentai Li, Nian Liu, Zeyu Mi, Yubin Xia, Haibo Chen: "Harmonizing Performance and Isolation in Microkernels with Efficient Intra-kernel Isolation and Communication", Usenix ATC, June 2020
- seL4 Performance, https://sel4.systems/About/Performance/, accessed 2020-11-08

Made For Real-World Use

Autonomous vehicles

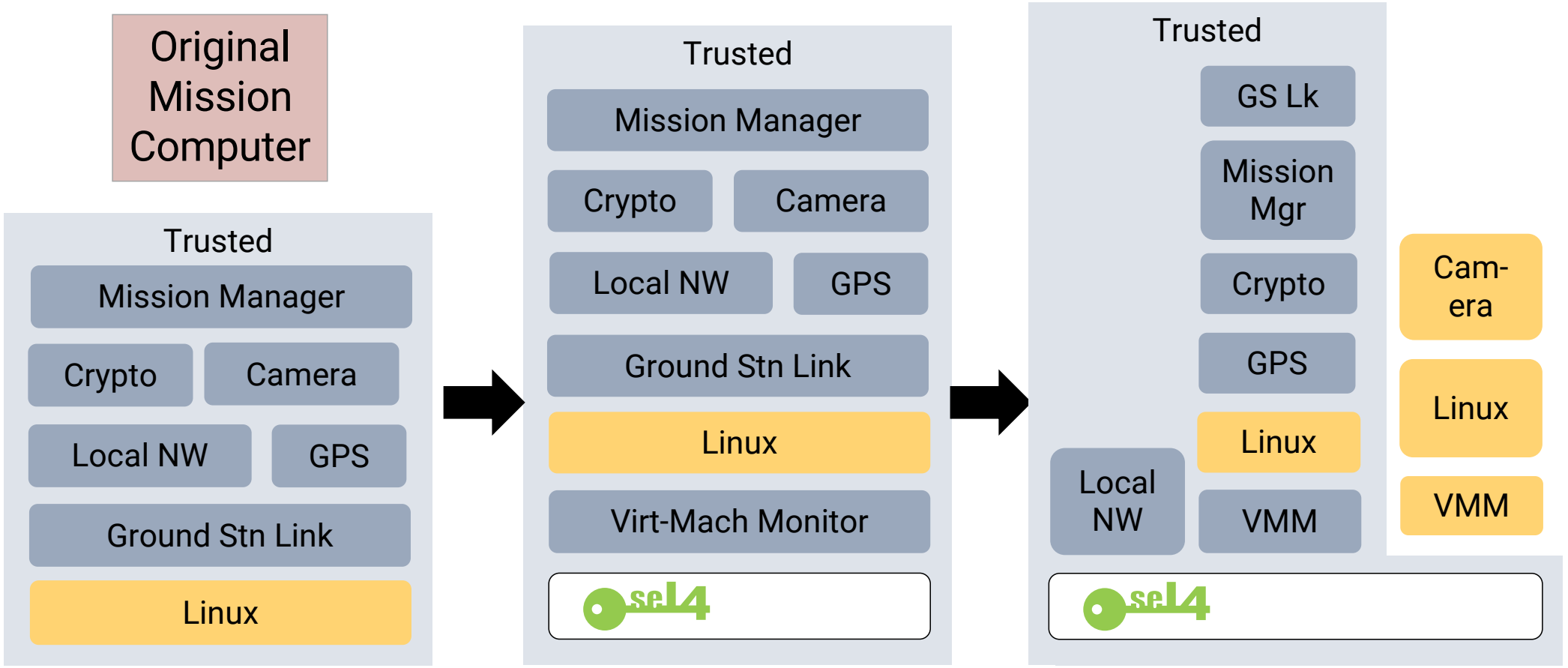


Satellites

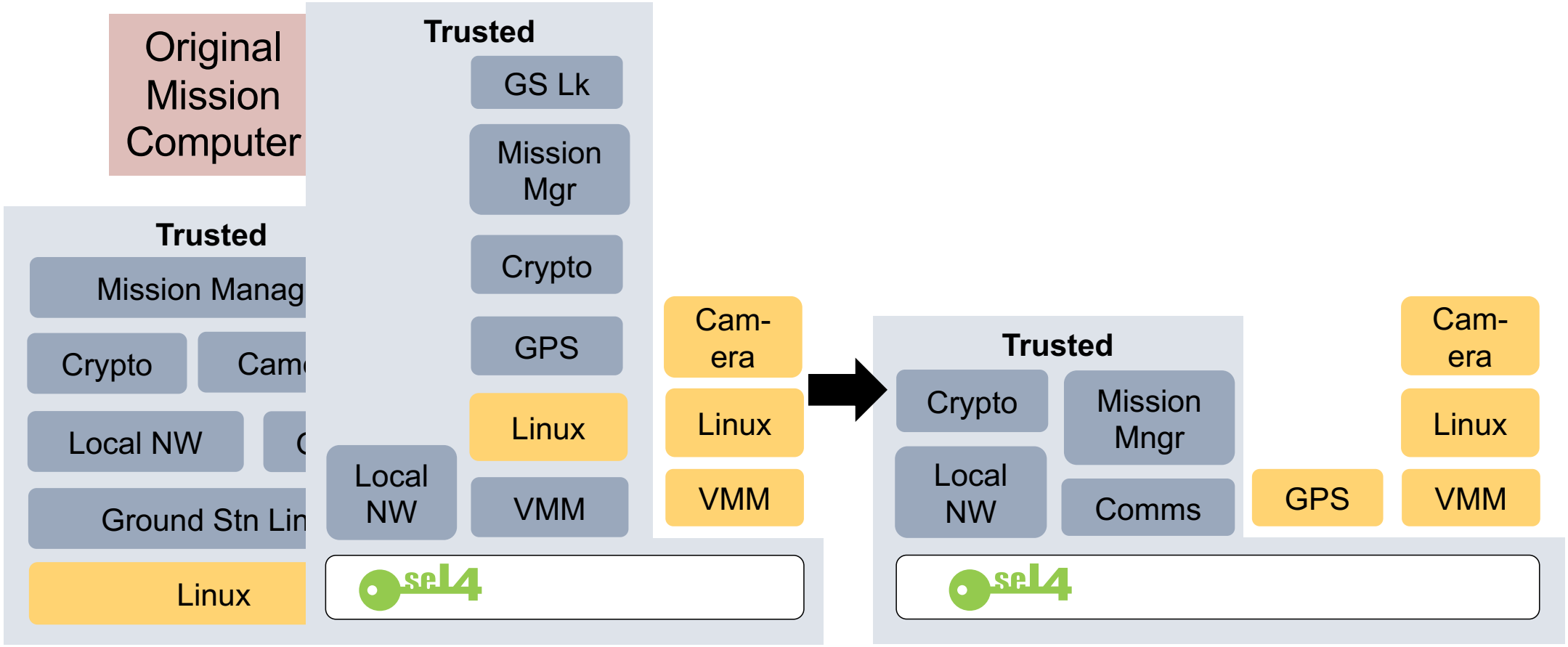
Secure communication device
In use in AU, UK defence forces

Laot: Critical infrastructure protection

DARPA HACMS: Incremental Cyber Retrofit



DARPA HACMS: Incremental Cyber Retrofit

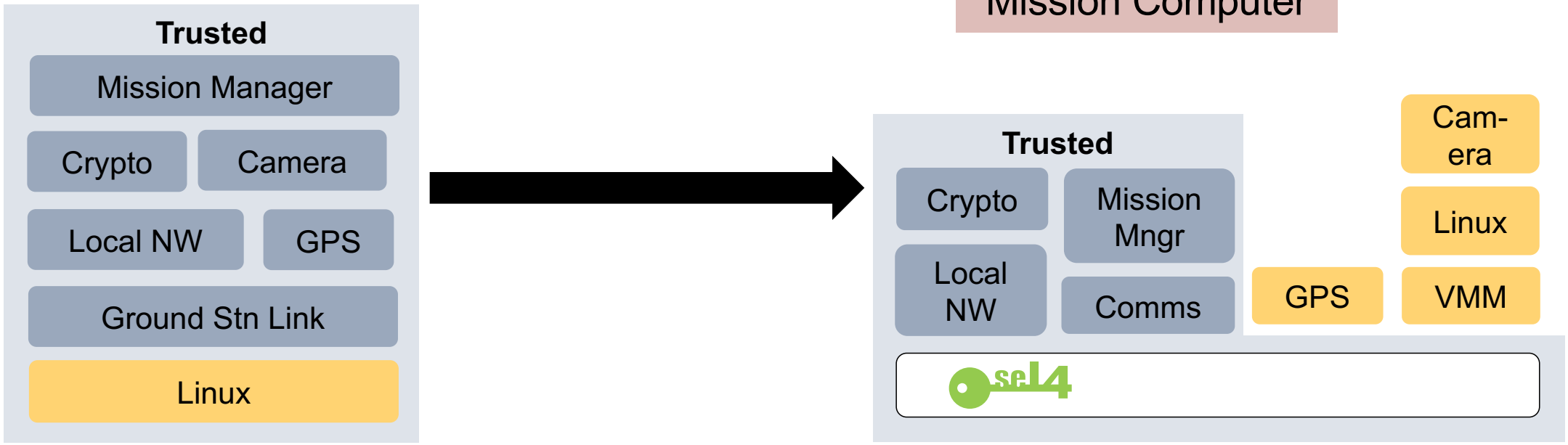


DARPA HACMS: Incremental Cyber Retrofit

Original Mission Computer

[Klein et al, CACM, Oct'18]

Cyber-secure Mission Computer



World's Most Secure Drone

← Tweet

DARPA
@DARPA

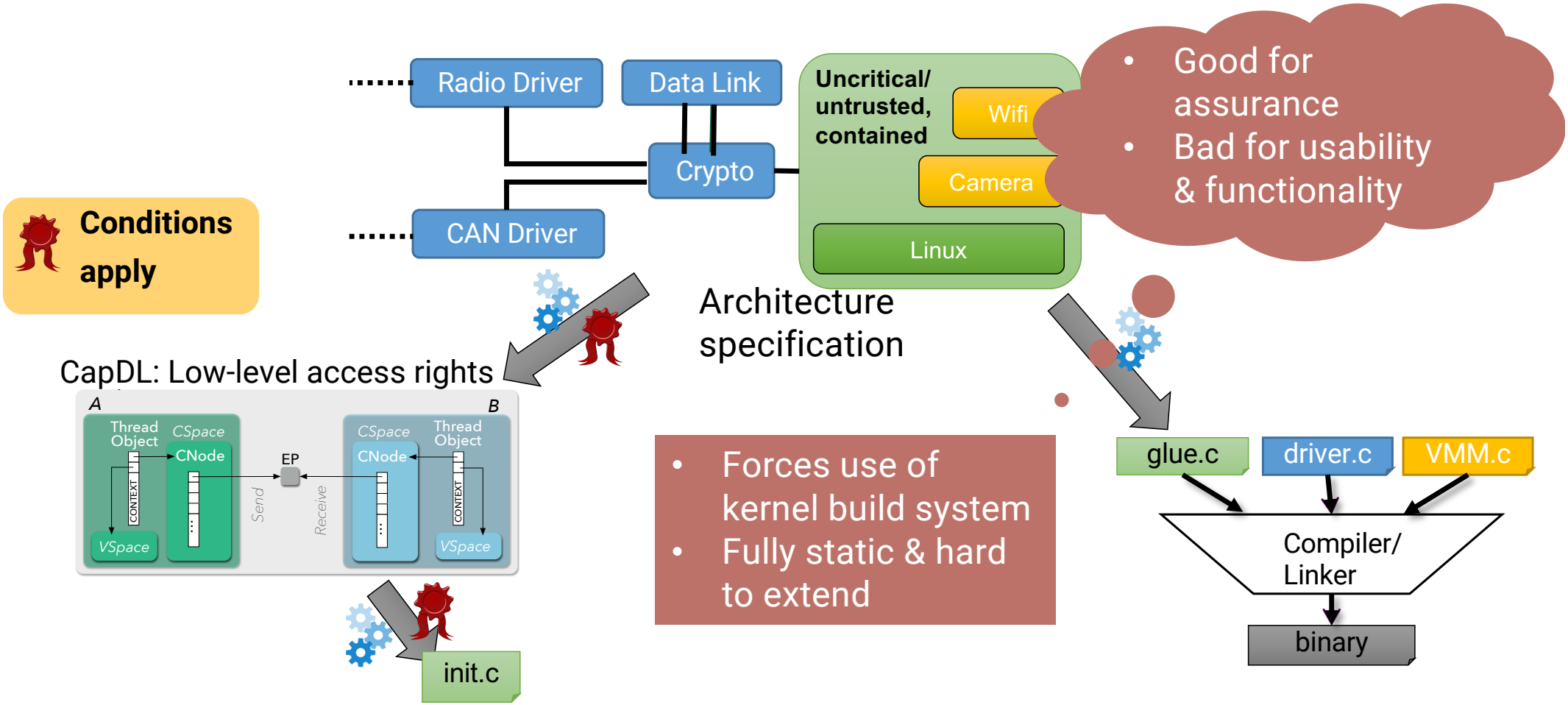
We brought a hackable quadcopter with defenses built on our HACMS program to @defcon #AerospaceVillage. As program manager @raymondrichards reports, many attempts to breakthrough were made but none were successful. Formal methods FTW!

So, Why Isn’t seL4 Everywhere By Now?

- Usability
- Functionality: Native services
- Trustworthiness: More than the kernel
- Embedded vs general-purpose

Usability

Recommended Framework: CAmkES



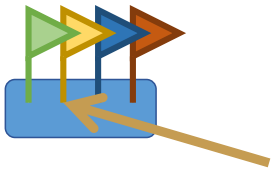
New Framework: seL4 Core Platform

Small OS for IoT, cyber-physical and other embedded use cases

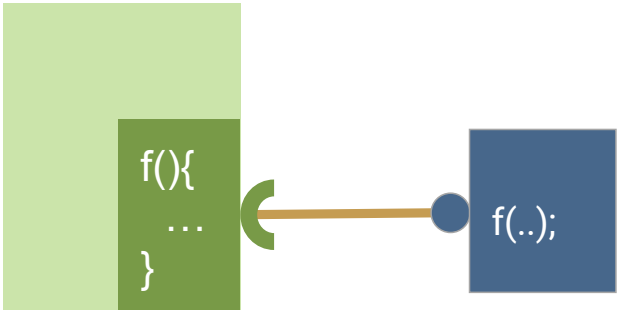
- Leverage seL4-enforced isolation for strong security/safety
- Retain seL4's superior performance
- Support "correct" use of seL4 mechanisms by default
- Ease development and deployment
  - SDK, integrate with build system of your choice
- Retain near-minimal trusted computing base (TCB)
- Be amenable to formal verification of the TCB

seL4CP Abstractions

- Thin wrapper of seL4 abstractions
- Encourage “correct” use of seL4



Notification



Protected Procedure Call (PPC)

Communication Channel (CC)

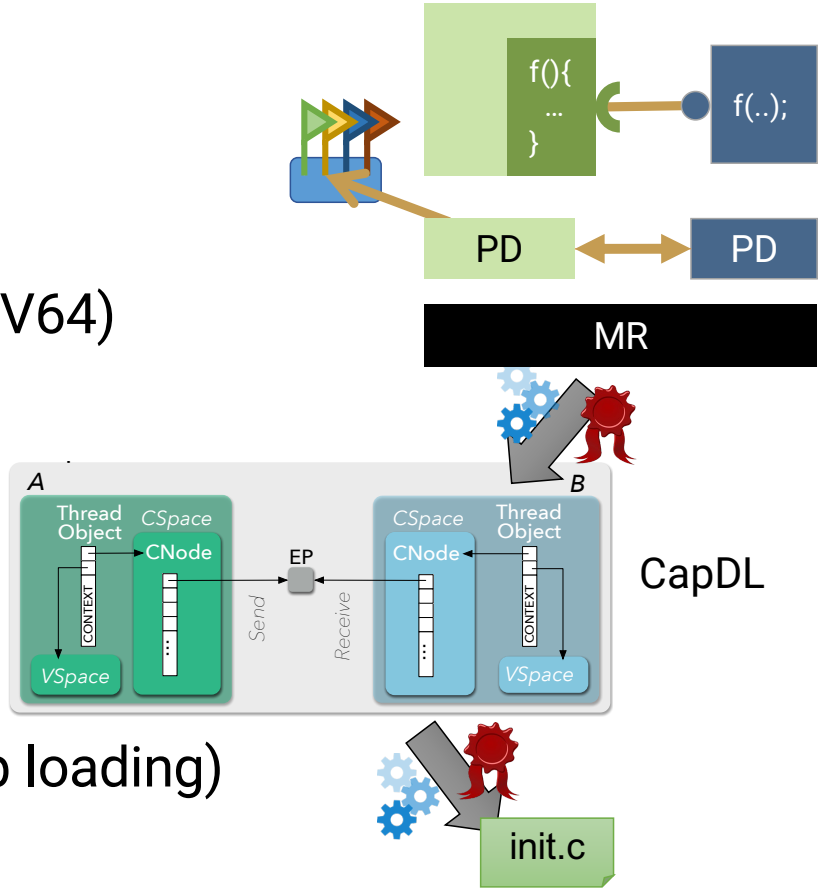
Protection Domain (PD)

Protection Domain (PD)

Memory Region (MR)

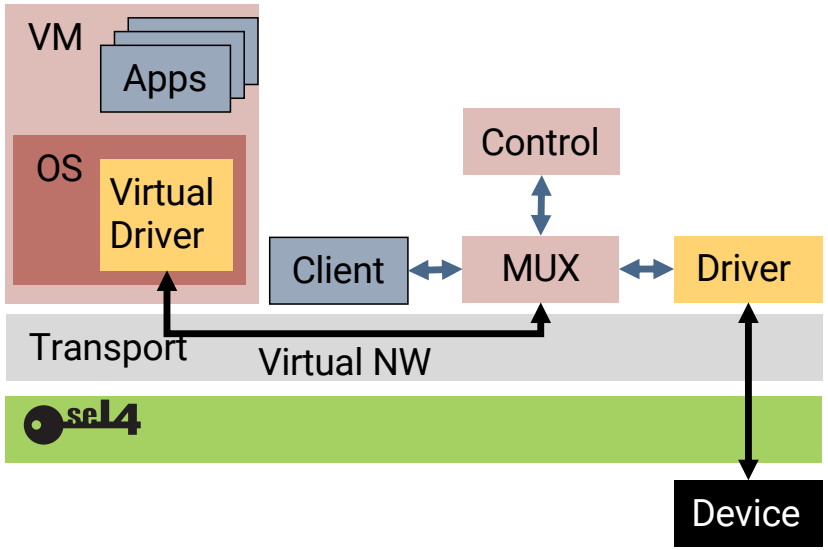
seL4CP Status

- Used in products (AArch64-based)
- Platform and ISA ports in progress (x64, RV64)
- Virtualisation support in progress
- Dynamic features prototype:
  - fault handlers
  - start/stop protection domains
  - re-initialise protection domains
  - empty protection domains (for late app loading)
- Verified mapping to CapDL in progress
- Push-button verification of CapDL under investigation



Functionality: Native Services

Key Component: Device Driver Framework



Aim:

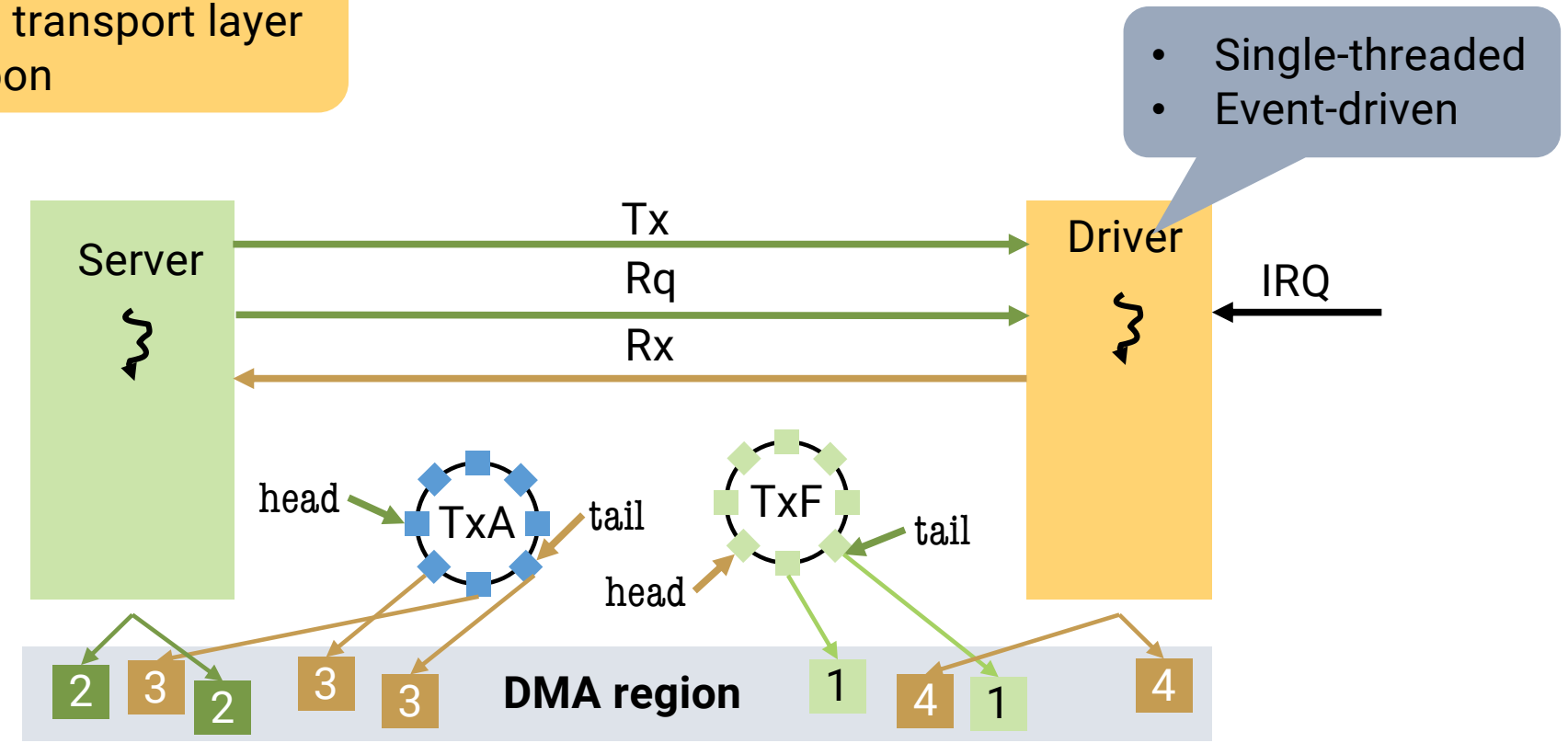
- Secure, low-overhead sharing of devices between components
- Defined interfaces to guide driver writers

Approach:

- Zero-copy transport layer
- Standard interfaces, VirtIO
- Re-use Linux drivers in per-device VM
- Investigate verifying MUX, Controller

Low-Overhead Transport

Status:
- Optimising transport layer
- Release soon

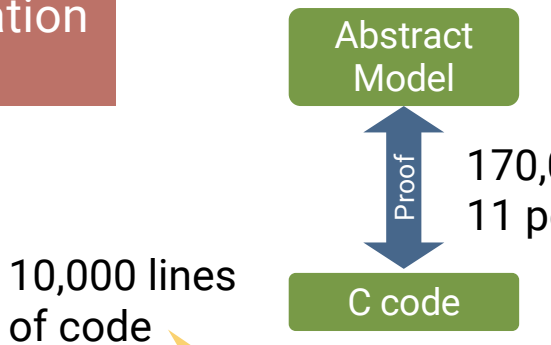


Trustworthiness

More than the kernel

Cost of Verification?

Verifying code not written for verification is infeasible!



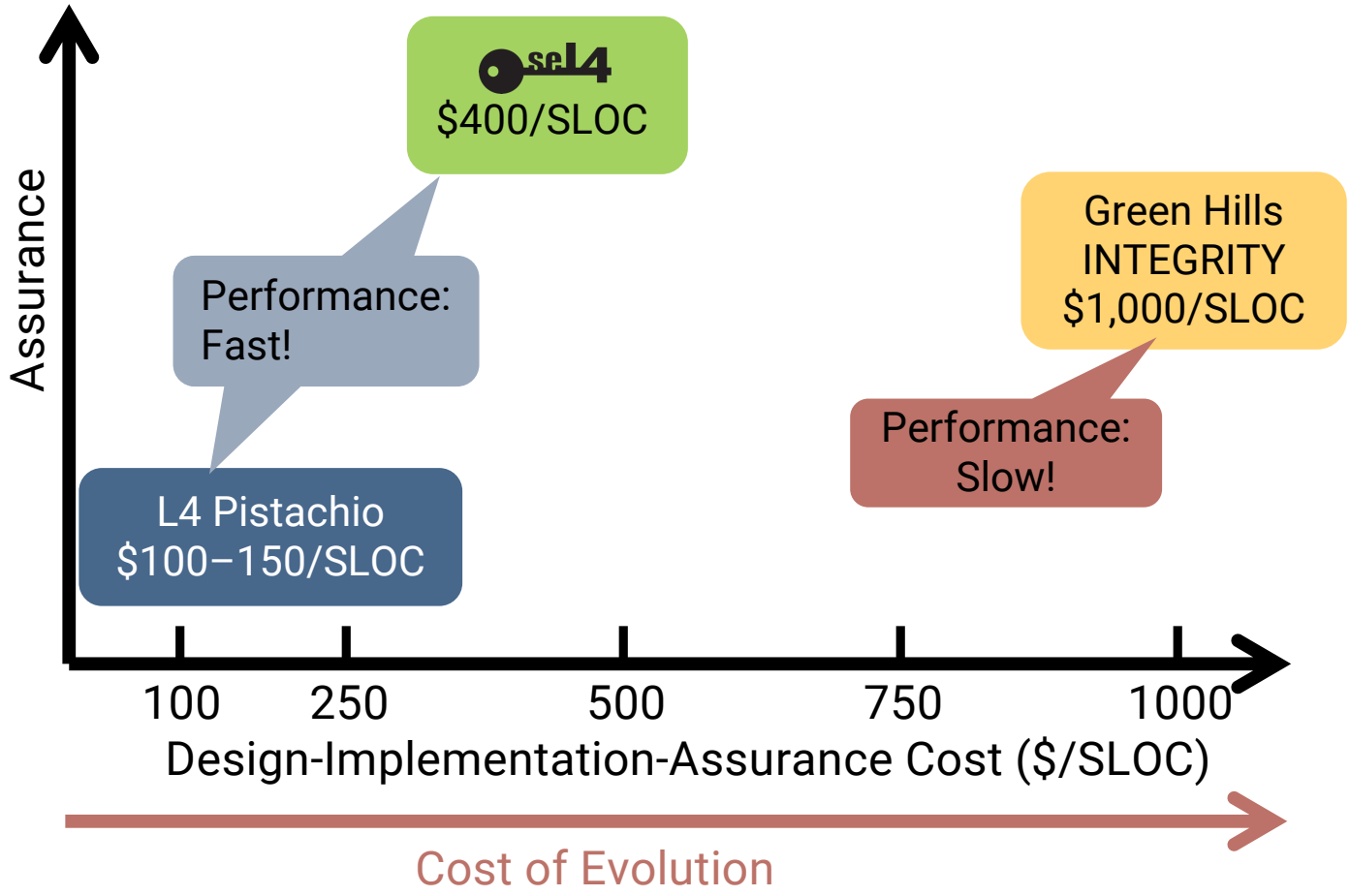
170,000 lines of proof
11 person years

10,000 lines of code

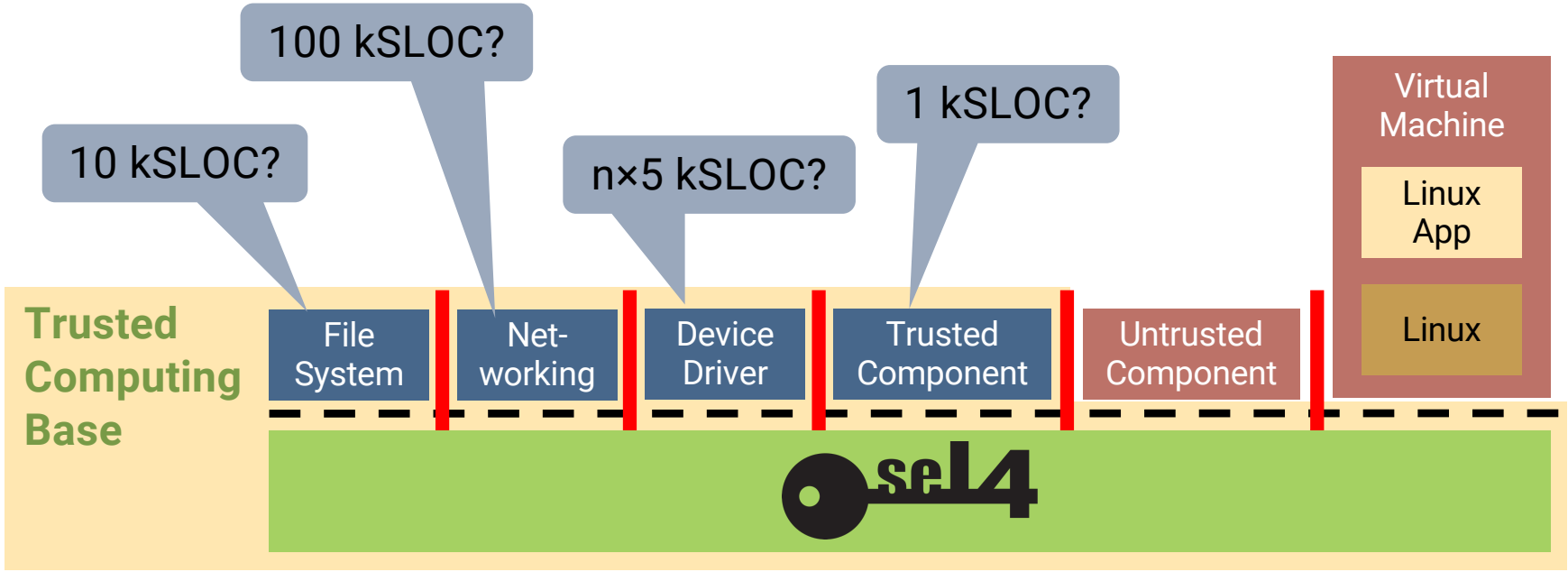
Designed and implemented for verification!

Complete seL4 proof base now ≫ 1,000,000 lines!

Verification Cost in Context

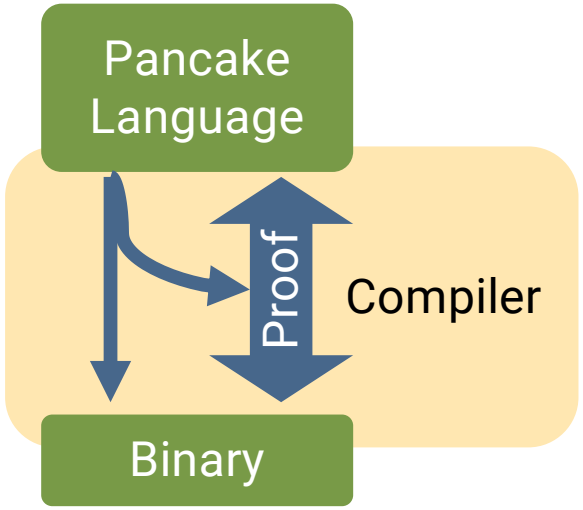


Beyond the Kernel



Reducing Cost of Verified Systems Software

Aim: Reduce cost of verified systems code



Idea:

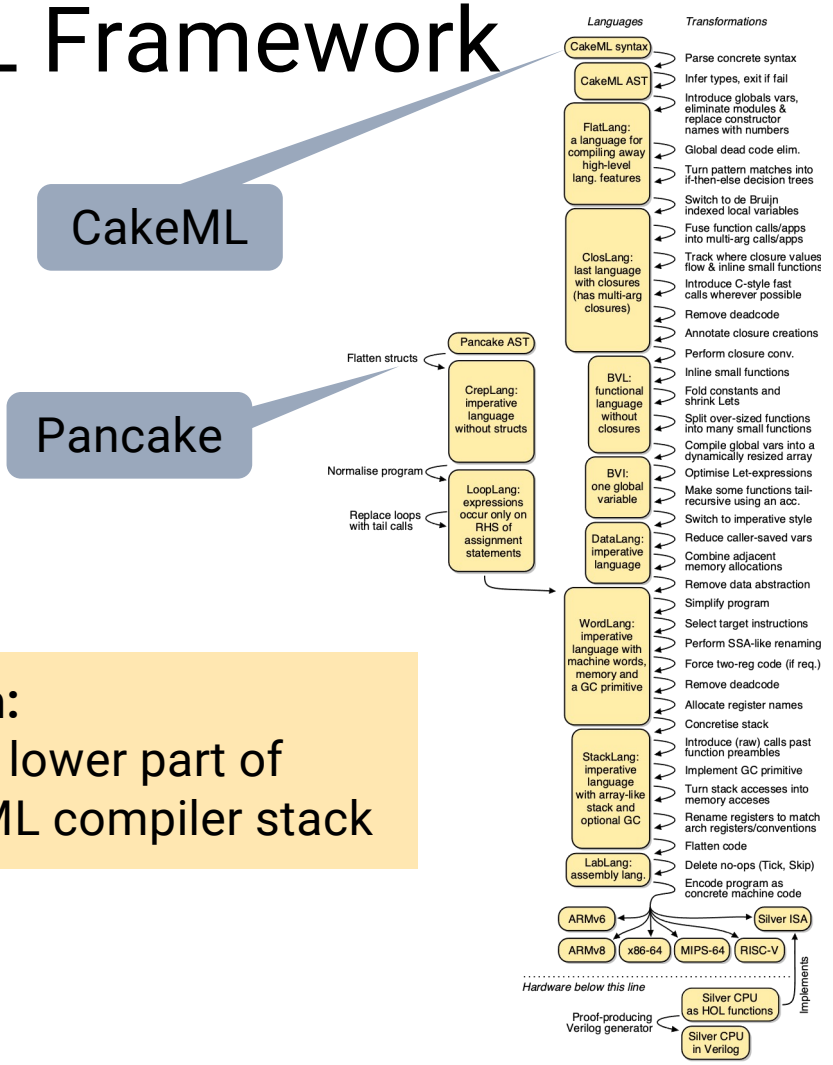
- Use low-level but safe systems language with certifying compiler
- Gives many proof obligations for free

Systems language:

- memory safe
- not managed (no garbage collector)
- low-level (obvious translation)
- interfacing to hardware
- minimal run time

Approach: Re-Use CakeML Framework

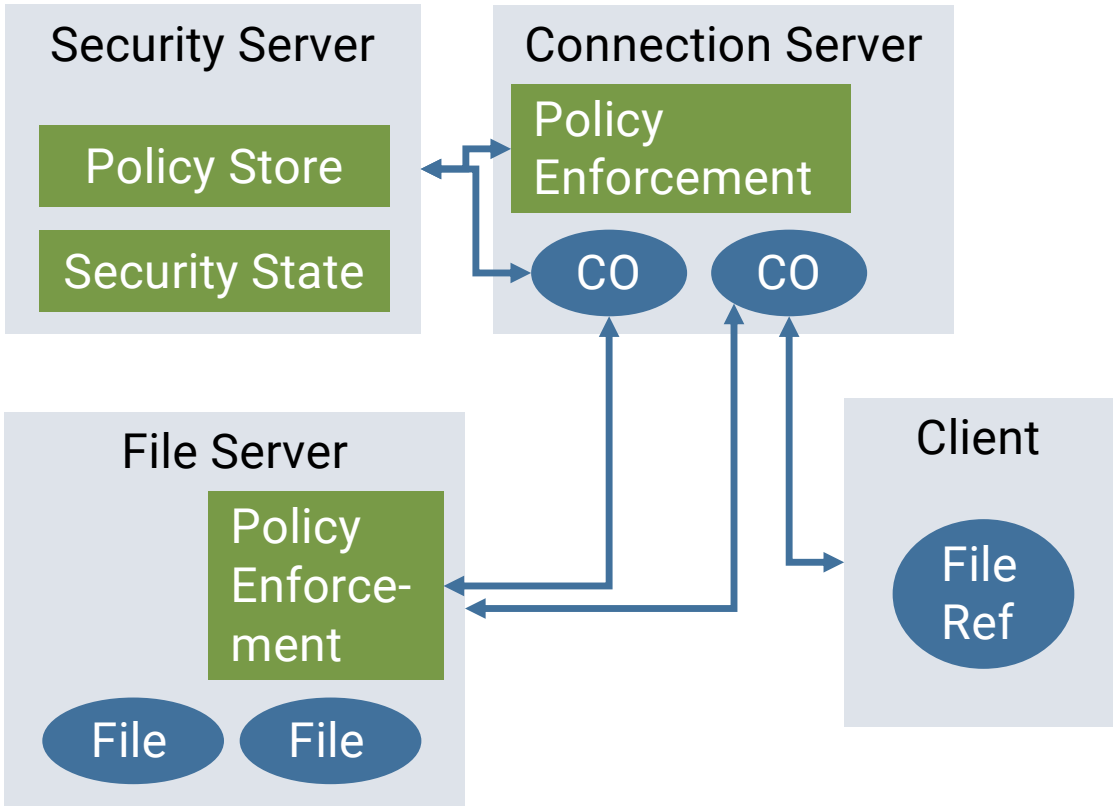
CakeML:
- functional language
- type & memory safe
- managed (garbage collector)
- high-level, abstract machine
- verified run time
- verified compiler
- mature system
- active ecosystem



Approach:
- re-use lower part of CakeML compiler stack

Secure General-Purpose OS?

Secure, General-Purpose OS



Aim: General-purpose OS that provably enforces a security policy

Requires:

- mandatory policy enforcement
- policy diversity
- minimal TCB
- low-overhead enforcement

Preventing Timing Channels – Provably

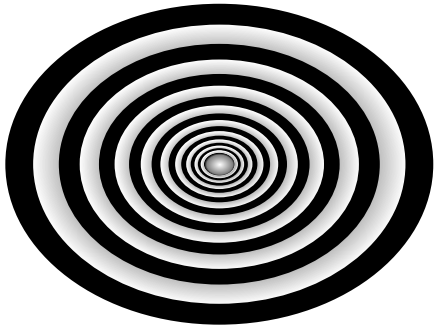
What is Spectre?

SPECTRE

=

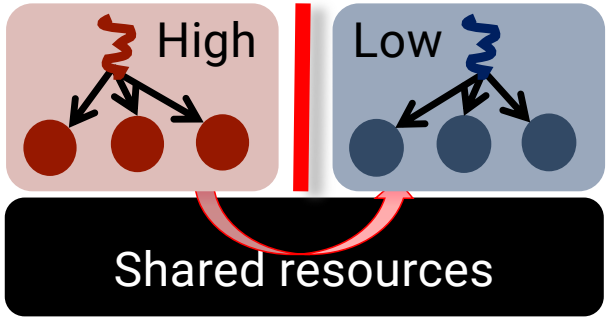
Speculation

+



Microarchitectural timing channel

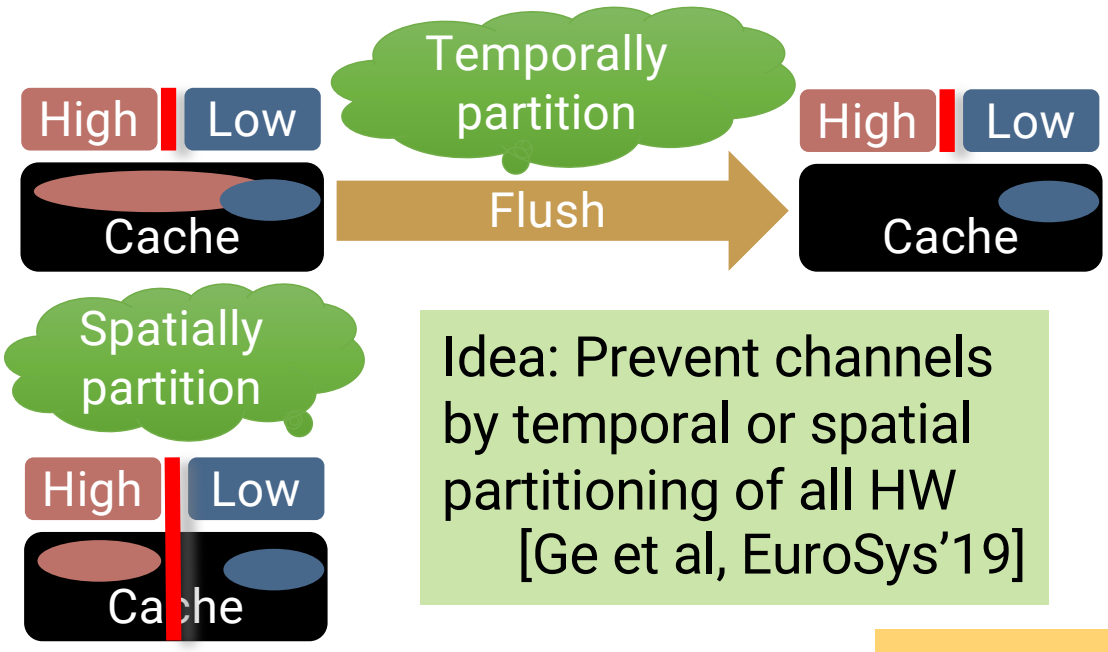
Microarchitectural Timing Channels



Contention for shared hardware resources affects execution speed, leading to timing channels

Standard approach: more patch&pray

Time Protection: Timing-Channel Prevention



Aim: Provably prevent information flow through micro-architectural timing channels

Idea: Prevent channels by temporal or spatial partitioning of all HW
[Ge et al, EuroSys'19]

Status:
1. Specified isolation property
2. Proved enforcement on high-level model
3. Now working on connecting to seL4 proofs

Summary

- seL4 *is* usable for real-world systems – but more functionality needed
- Usability should (hopefully) be addressed with the Core Platform
- seL4 Device Driver Framework will support I/O and device sharing
  - … including per-device Linux driver VMs
- We *think* Pancake will enable verified drivers
- We're about 1 year away from proving timing-channels prevention

Defining the state of the art in trustworthy operating systems since 2009

Thanks To Our Sponsors!

Australian Government

Department of Defence

neutrality

Technology Innovation Institute

seL4 FOUNDATION

DARPA

Australian Government

Australian Research Council

in association with

National Cyber Security Centre

The seL4 Foundation

Premium Members

Cyber

HENSOLDT

UNSW SYDNEY

地平线 Horizon Robotics

jumptrading

Li Auto

NIO

General Members

Adventium LABS

Cog

DORNERWORKS

GHOST

KRYO

LOTUS

penten

Proofcraft

Raytheon Technologies

SECOND STATE

xcalibyte

Associate Members

ETH zürich

KANSAS STATE UNIVERSITY

in association with National Cyber Security Centre

RISC-V

TUM