School of Computer Science & Engineering

Trustworthy Systems Group

Intelligent Vehicle Security Needs a Verified Operating System

Gernot Heiser
UNSW and seL4 Foundation
gernot@unsw.edu.au

Car Hacking Danger Is Likely Closer Than You Think

A Detroit Free Press report shows there were 150 automotive cybersecurity incidents in 2019 alone.

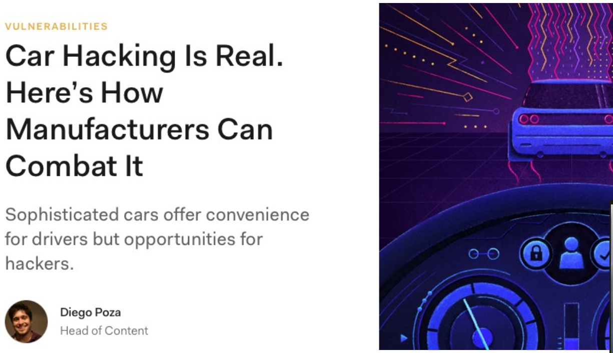
BY SEBASTIAN BLANCO PUBLISHED: SEP 4, 2021

NATIONAL

Nearly 400 car crashes in 11 months involved automated tech, companies tell regulators

June 15, 2022 · 1:26 PM ET

By The Associated Press



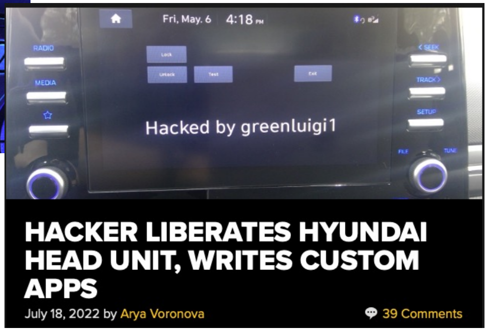
VULNERABILITIES

Car Hacking Is Real. Here's How Manufacturers Can Combat It

Sophisticated cars offer convenience for drivers but opportunities for hackers.

Diego Poza

Head of Content



HACKER LIBERATES HYUNDAI HEAD UNIT, WRITES CUSTOM APPS

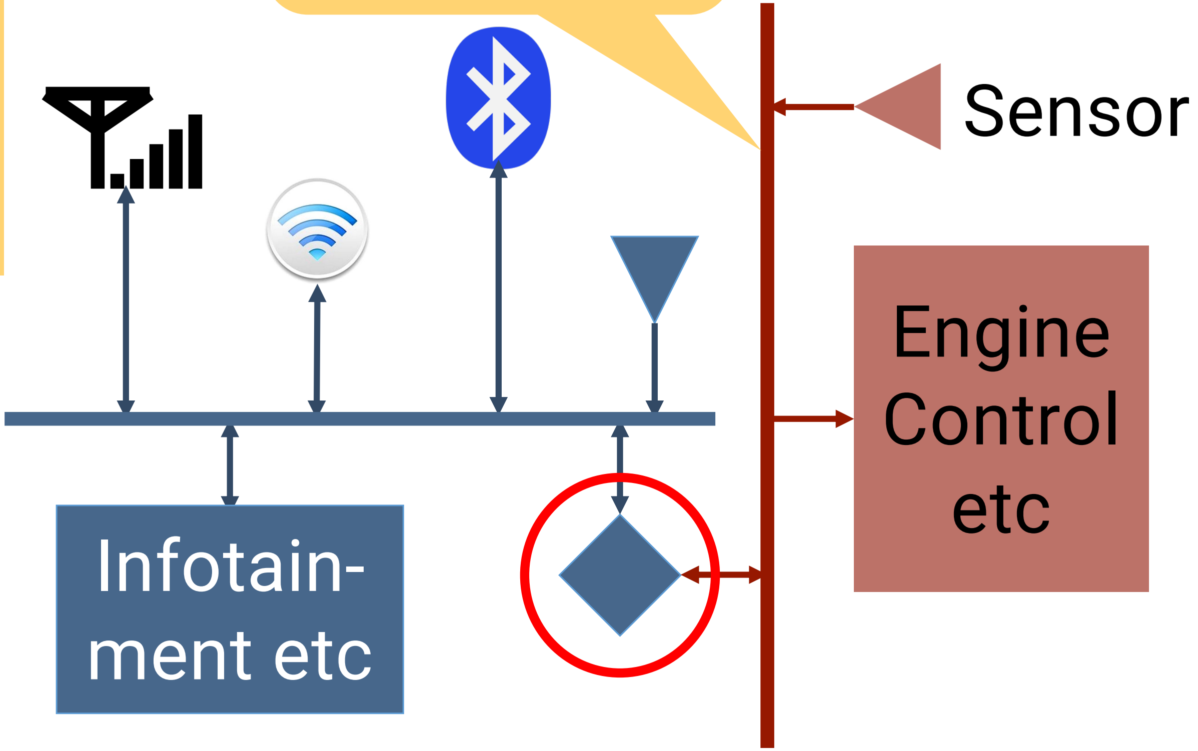
July 18, 2022 by Arya Voronova 39 Comments

Traditional Cars Are Expos

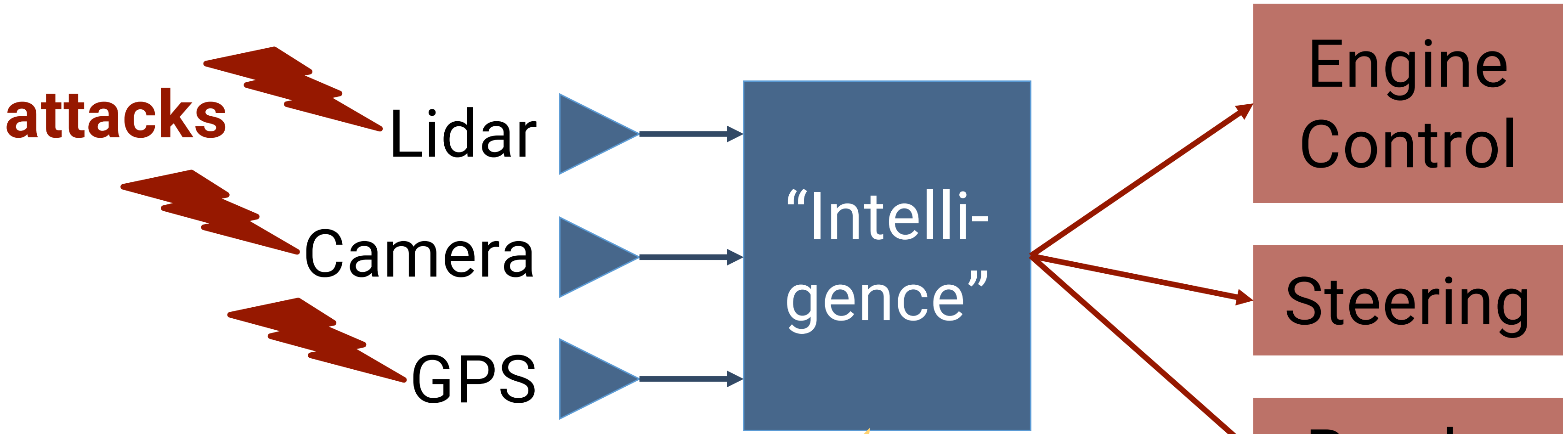
Networking for:

- Entertainment
- Driver information
- Safety (tyre pressure…)
- Maintenance (OTA upgrades)

No security whatsoever on CAN bus!



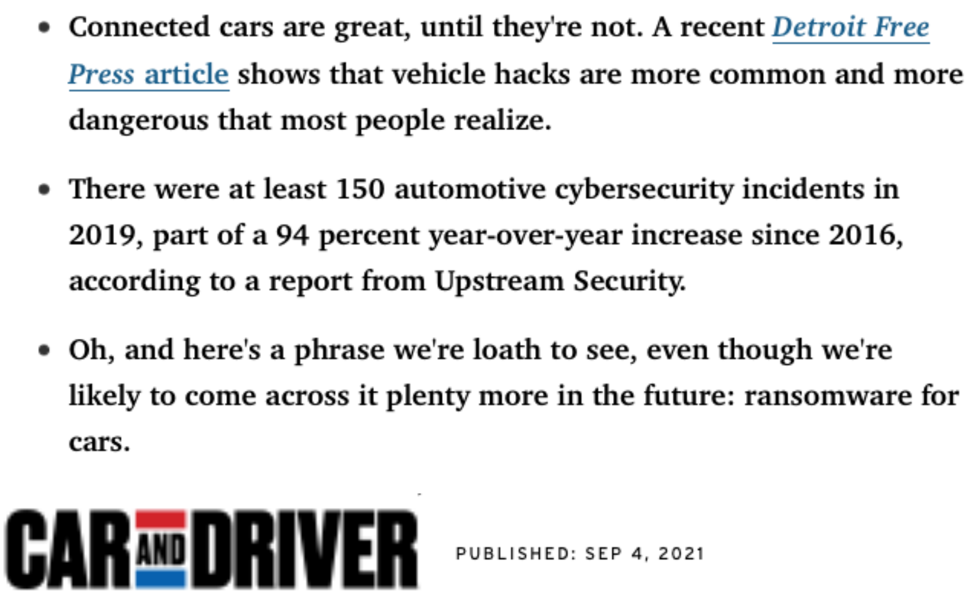
Intelligent Vehicles: Hacker's Paradise!



Attack by:
- spoofing signals
- exploiting bugs

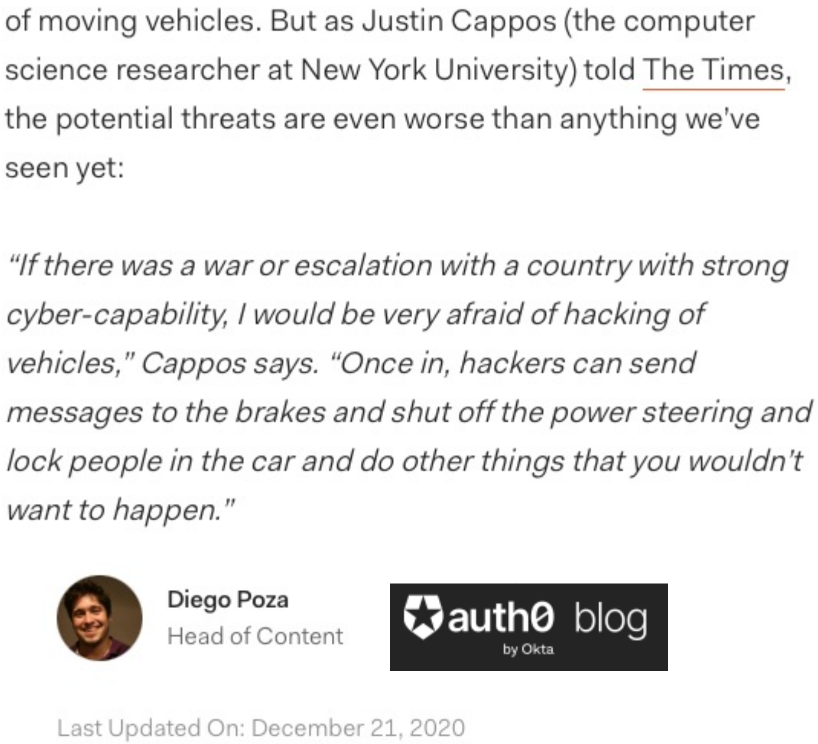
Highly complex!

Who Cares?



- Connected cars are great, until they're not. A recent *Detroit Free Press article* shows that vehicle hacks are more common and more dangerous that most people realize.
- There were at least 150 automotive cybersecurity incidents in 2019, part of a 94 percent year-over-year increase since 2016, according to a report from Upstream Security.
- Oh, and here's a phrase we're loath to see, even though we're likely to come across it plenty more in the future: ransomware for cars.

CAR AND DRIVER PUBLISHED: SEP 4, 2021



of moving vehicles. But as Justin Cappos (the computer science researcher at New York University) told The Times, the potential threats are even worse than anything we've seen yet:

"If there was a war or escalation with a country with strong cyber-capability, I would be very afraid of hacking of vehicles," Cappos says. "Once in, hackers can send messages to the brakes and shut off the power steering and lock people in the car and do other things that you wouldn't want to happen."

Diego Poza
Head of Content

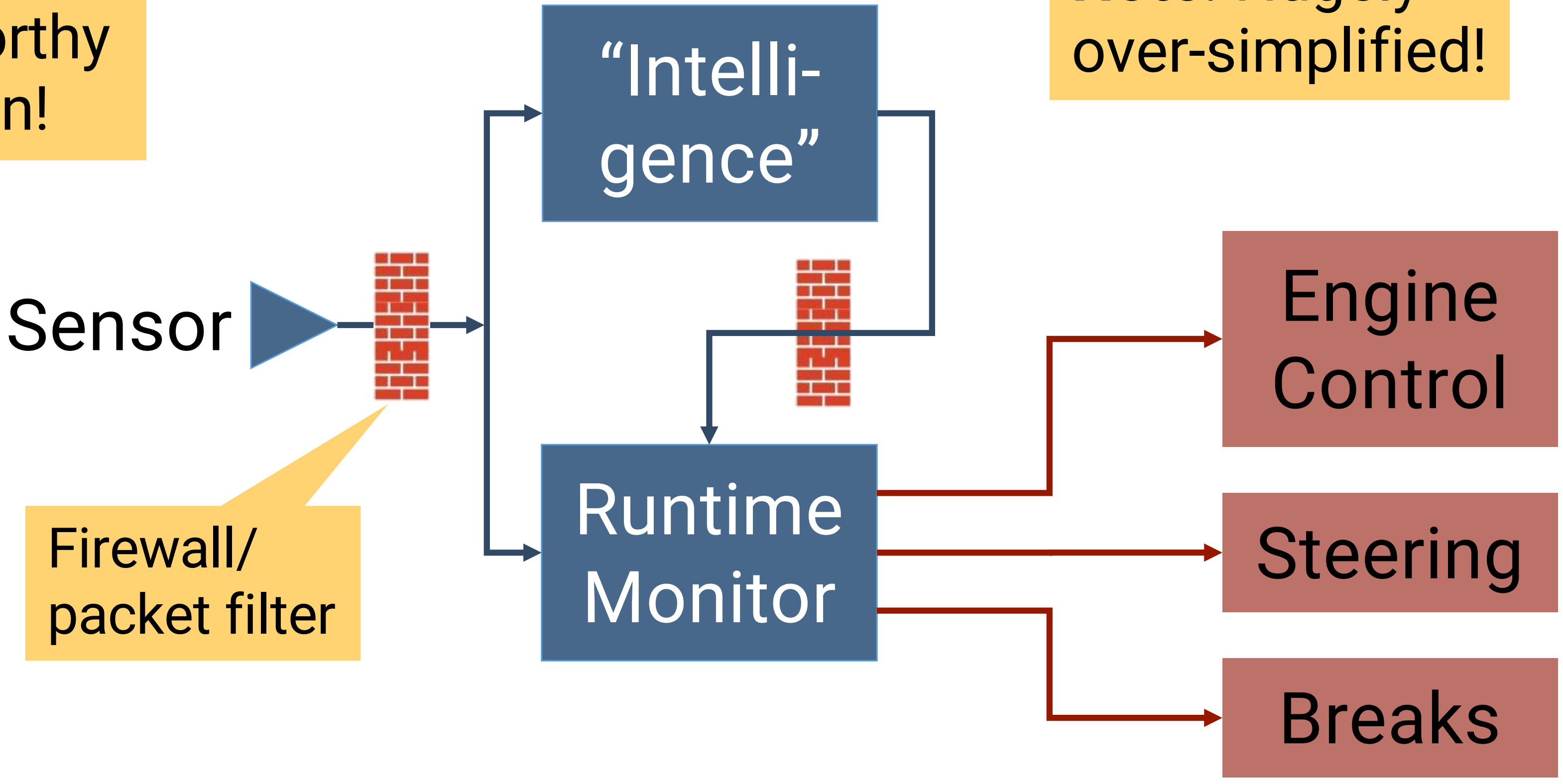
auth0 blog by Okta

Last Updated On: December 21, 2020

How Can We Protect Intelligent Vehicles?

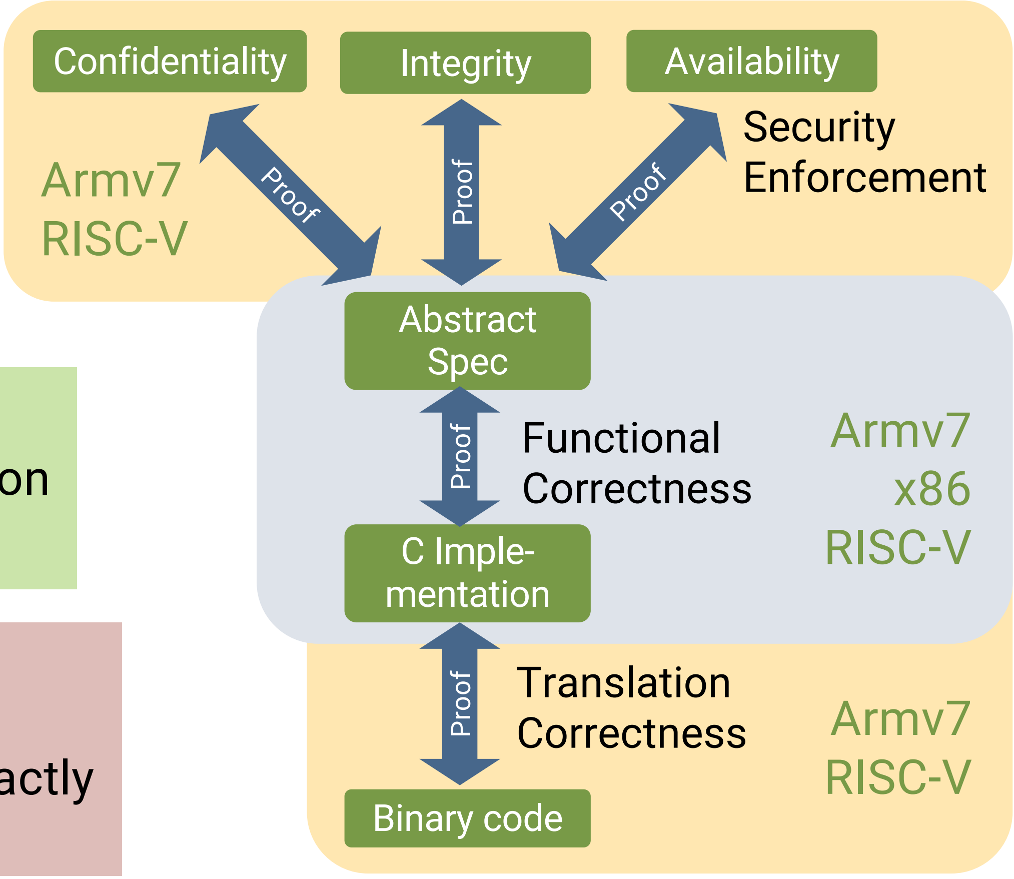
Requires trustworthy isolation!

Note: Hugely over-simplified!



Firewall/ packet filter

seL4 Foundation for Truly Secure Systems



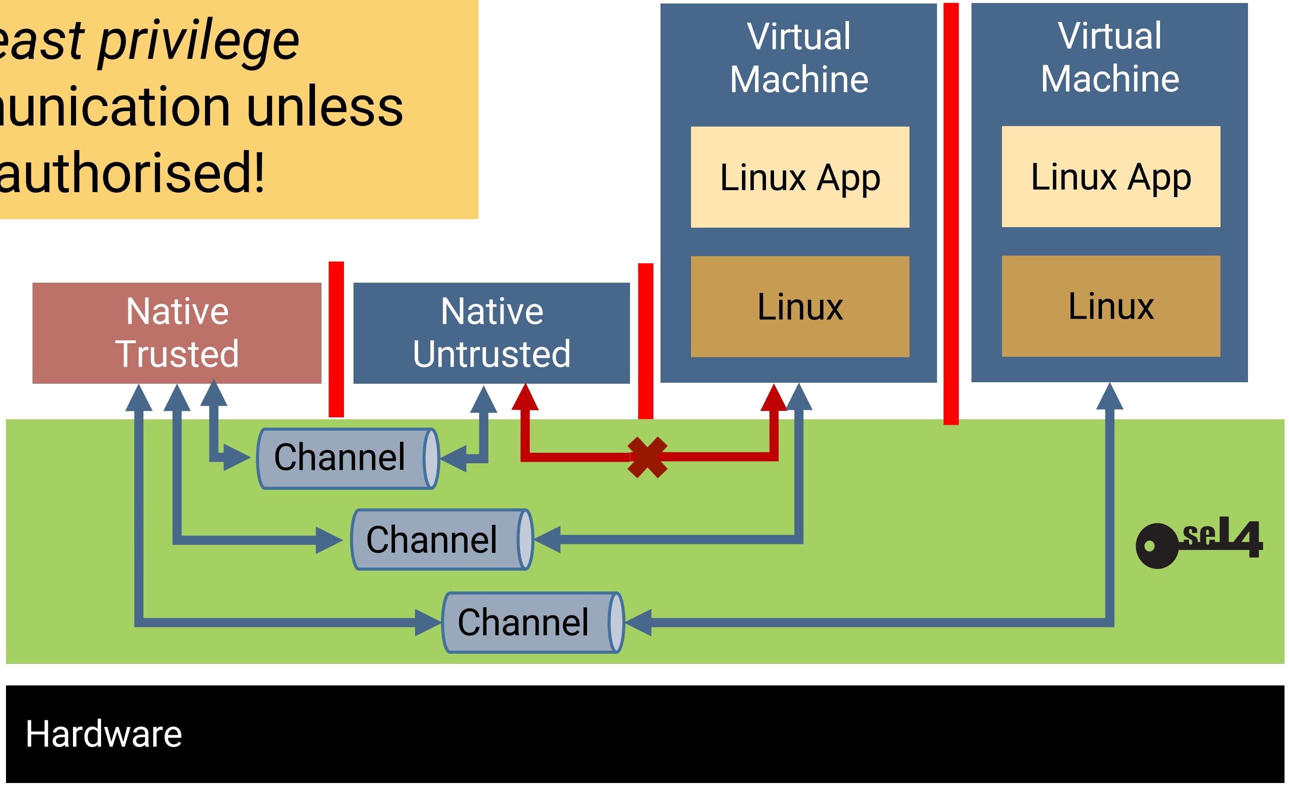
- Comprehensive formal verification
- Capabilities for fine-grained protection
- World’s fastest microkernel

Present limitations

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

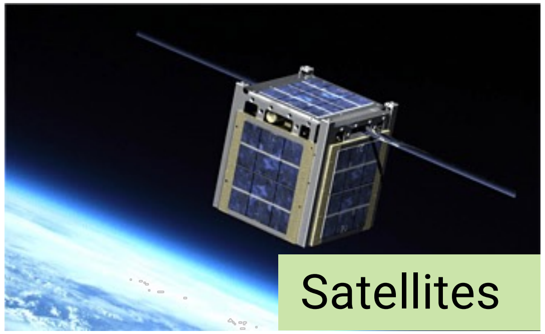
Capabilities: Fine-Grained Protection

- Enforce *least privilege*
- No communication unless explicitly authorised!



Made For Real-World Use

Autonomous vehicles

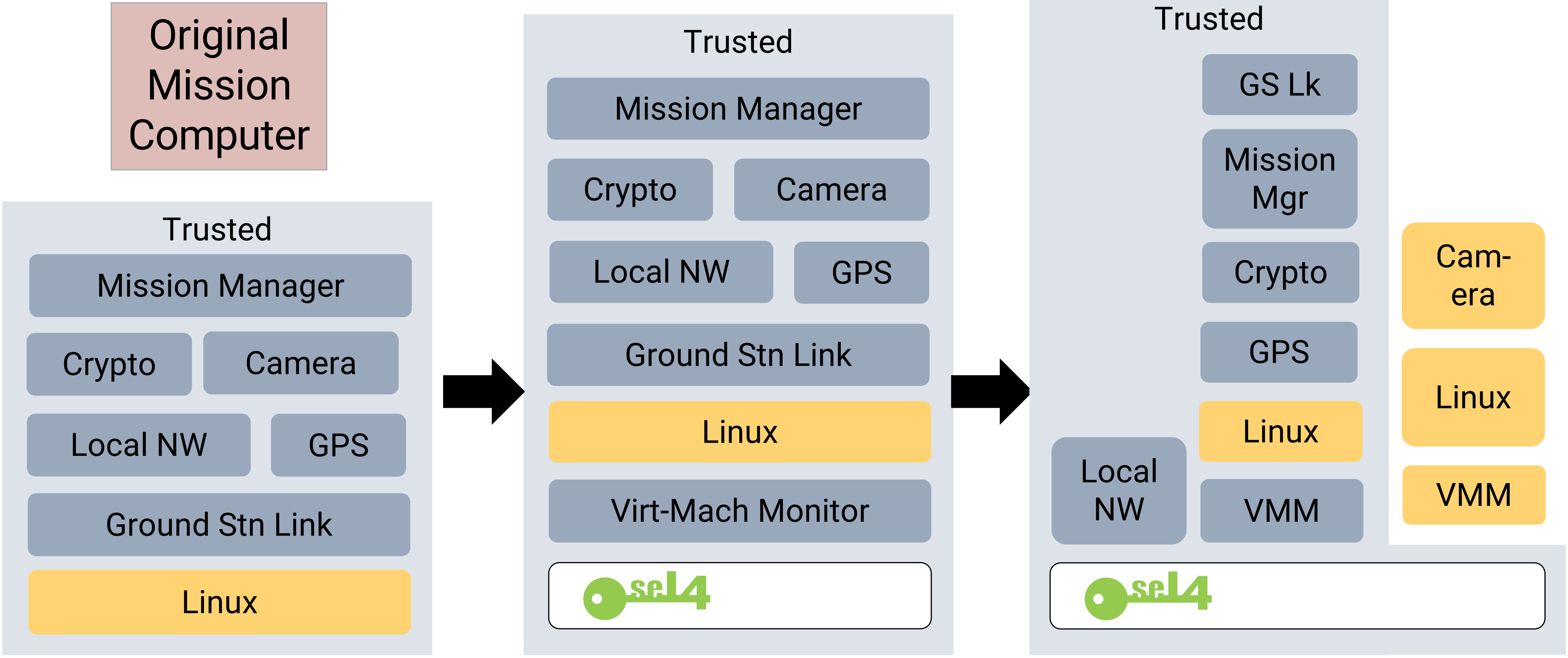


Satellites

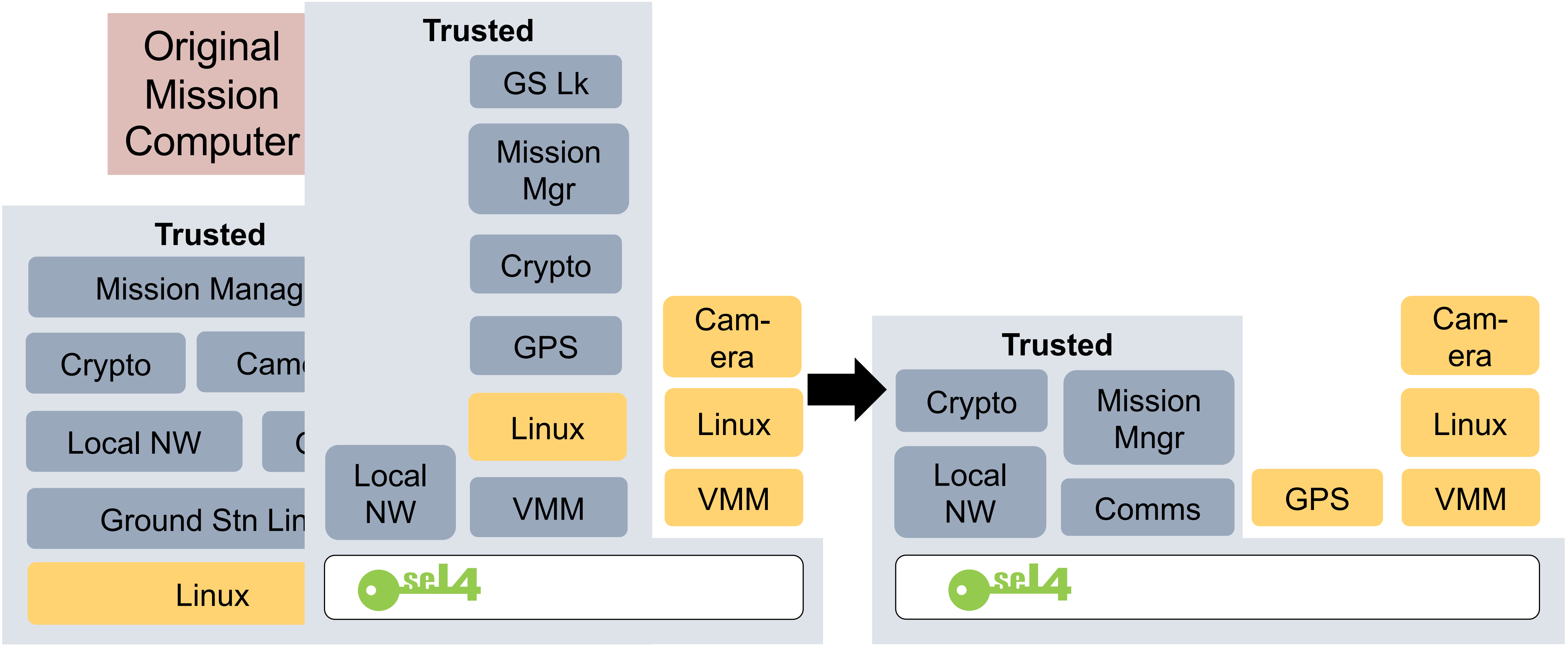
Secure communication device
In use in multuiple defence forces

Laot: Critical infrastructure protection

DARPA HACMS: Incremental Cyber Retrofit

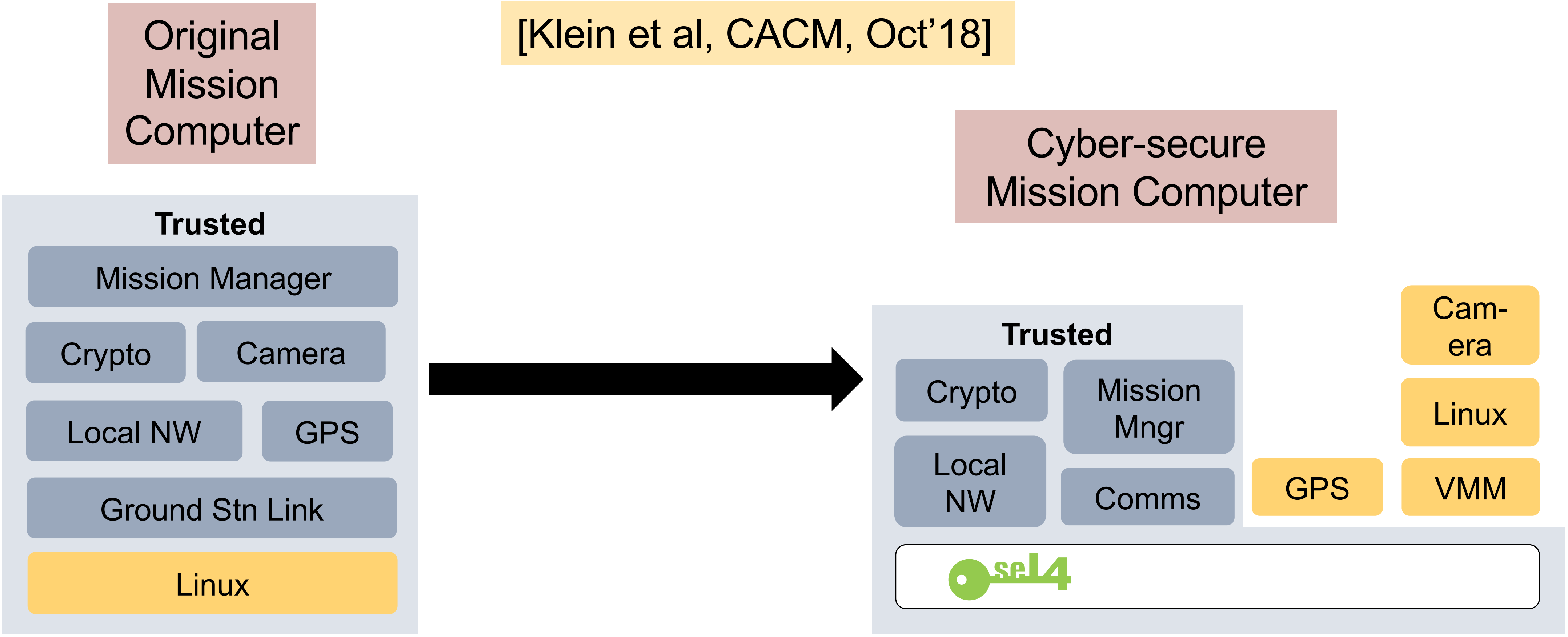


DARPA HACMS: Incremental Cyber Retrofit

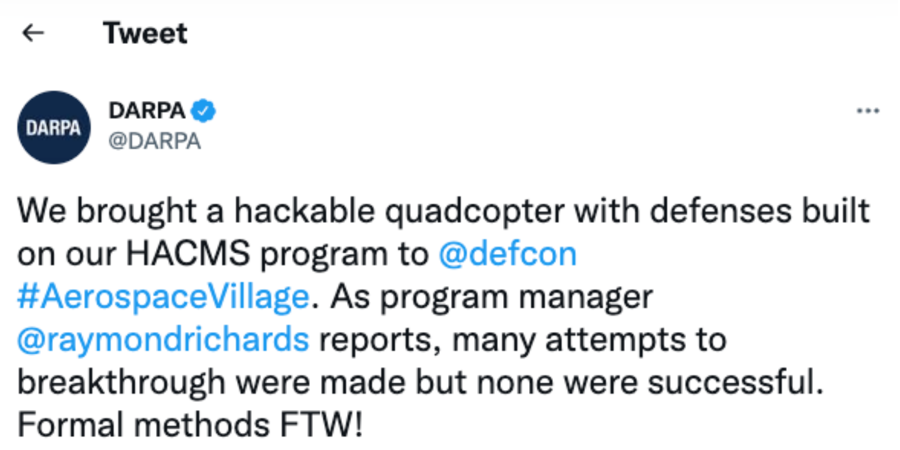


DARPA HACMS: Incremental Cyber Retrofit

[Klein et al, CACM, Oct'18]



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!

seL4 Needs an OS

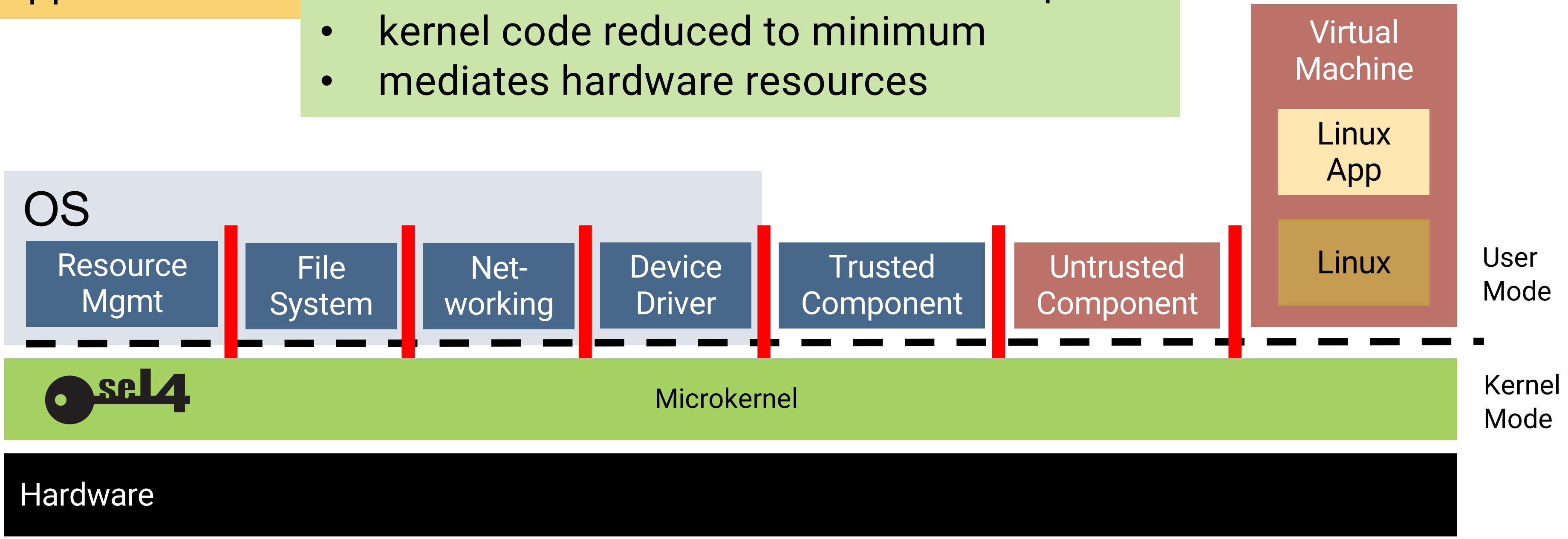
Microkernel Is Not An OS

Modularisation: Separate components

- operating-system services
- device drivers
- applications

Microkernel enforces isolation – bullet-proof

- kernel code reduced to minimum
- mediates hardware resources



Build a performant OS from Scratch?

Yes – if we strictly observe some fundamental principles: KISS

- Fine-grained modularity, strong *separation of concerns*
- Least privilege
- *Radical Simplicity™*: provide *only* the features needed
- Swappable, *use-case specific policy* (rather than universal policy)

Software engineering 101

Reason about security

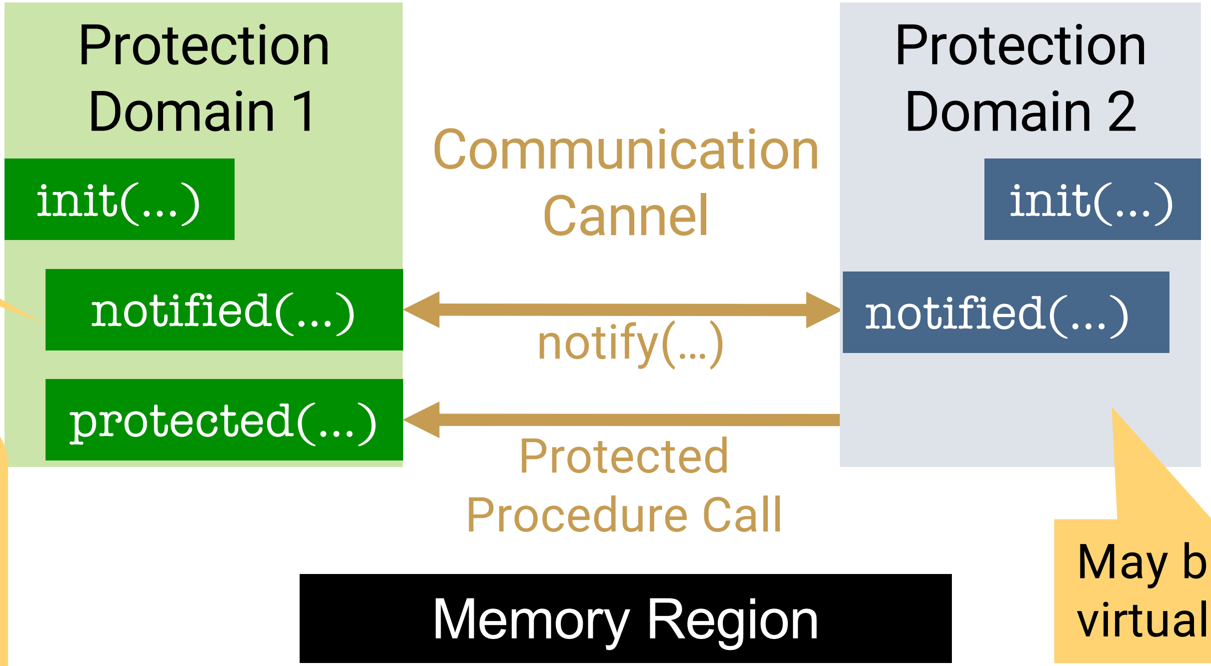
KISS

"Universal" policies are complex, always have pathological cases

Also limit scope:
- Cyberphysical systems
- IoT systems

OS Framework: The seL4 Core Platform

Simple, single-threaded event-driven

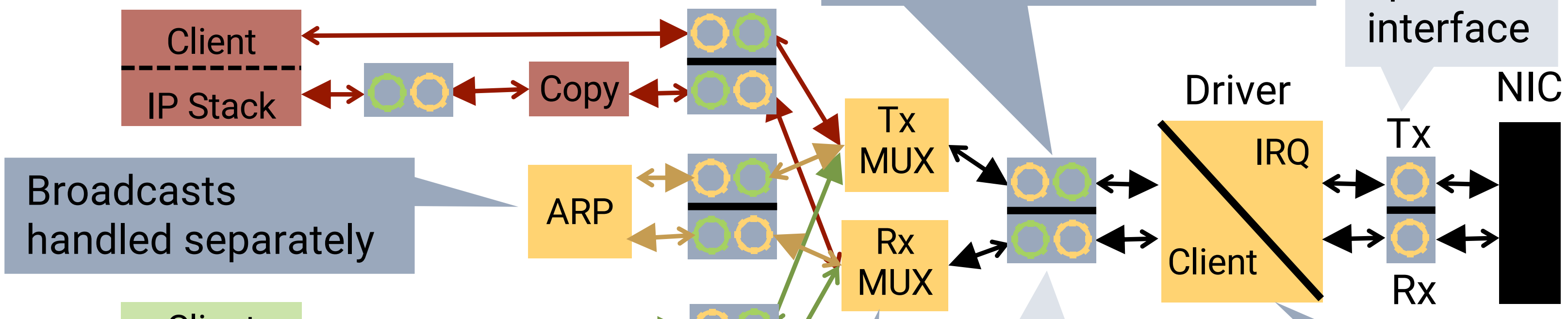


May be a virtual machine

- Minimal abstractions
- Thin wrapper of seL4
- Encourage “correct” use of seL4 primitives
- Aimed at IoT/cyberphysical

Apply KISS Principles to OS

Example: Networking



Large number of extremely simple components

Zero-copy, lock-free, asynchronous shared-memory communication

HW-specific interface

Broadcasts handled separately

HW-independent interface

Device driver only translates HW-specific to HW-independent interface

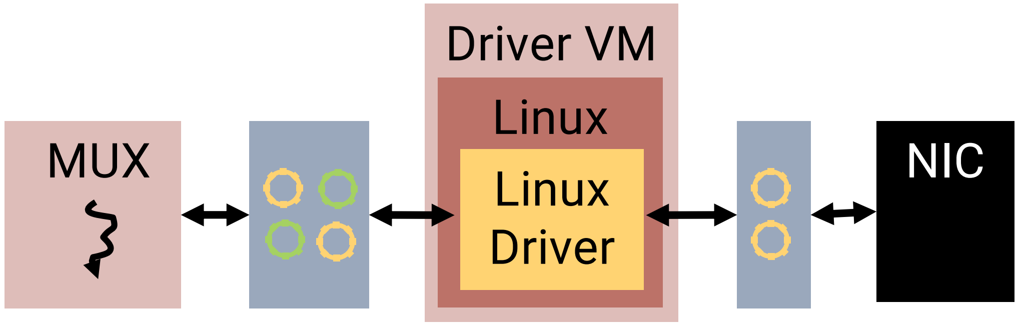
IP stack is client library

Copy where needed for security only

Multiplexers share device between clients

Legacy Drivers?

Can use Linux drivers wrapped into individual driver VM



Can This Work?

Comparison to Linux

Linux:
- NW driver: 4k lines
- NW system total: 1M lines

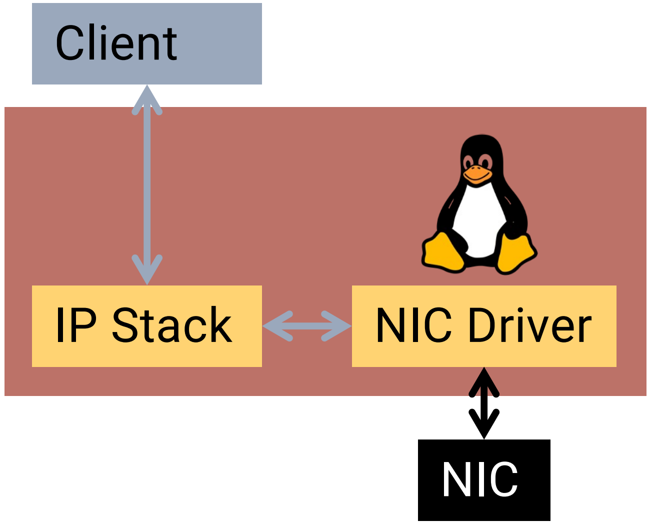
Performance?

KISS design:
- NW driver: 700 lines
- MUX: 400 lines
- Copier: 200 lines
- IP stack: much simpler, client library
- shared NW system total < 2,000 lines

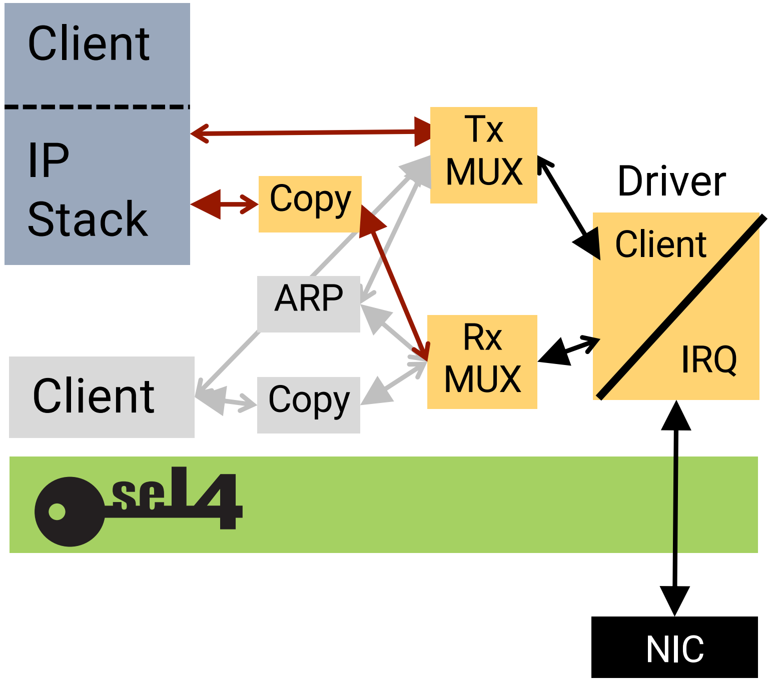
Written by second-year student!

Evaluation Setup

2 context switches per packet

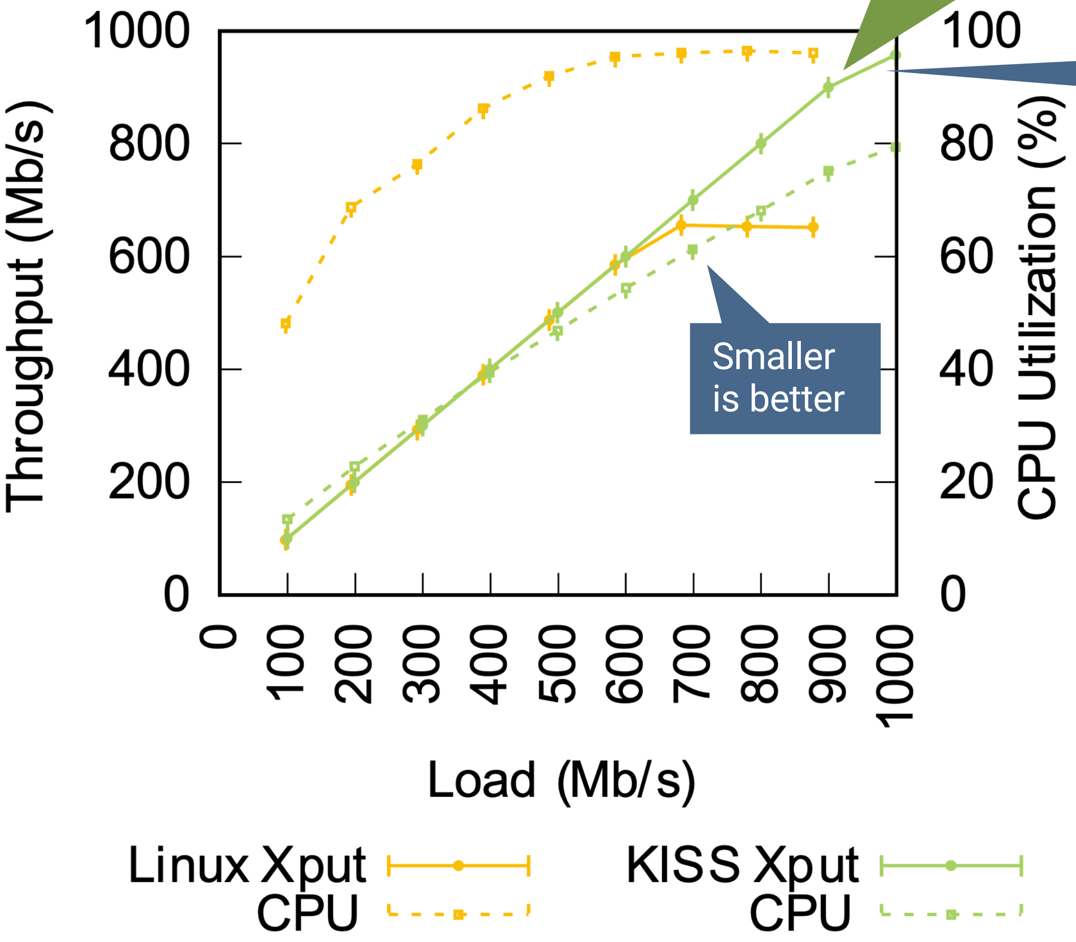


10 context switches per packet



Achieved Performance

- Gigabit Ethernet
- single core



Simplicity wins!

Bigger is better

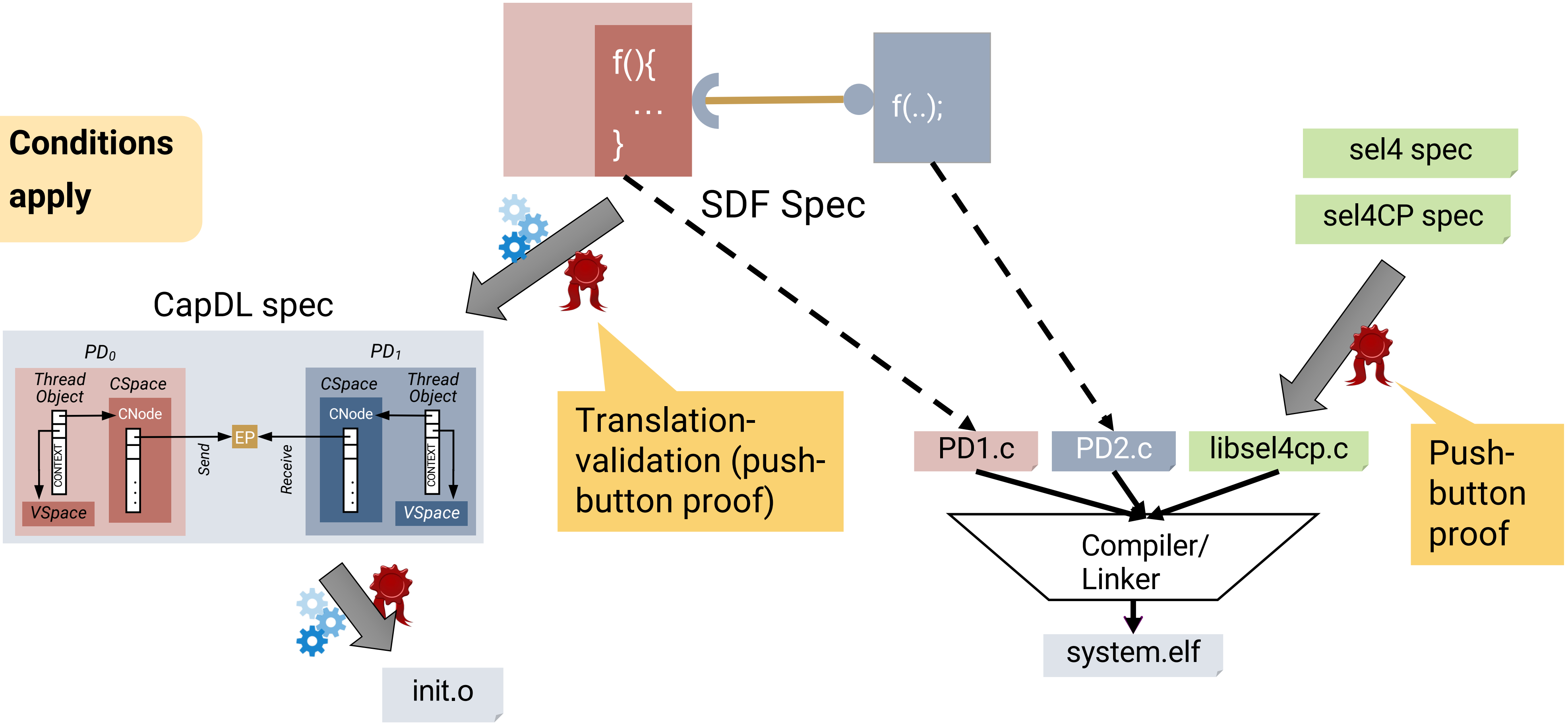
Smaller is better

Core take-away: We can build a performant OS this way!

Can We Verify It?

seL4CP Verification

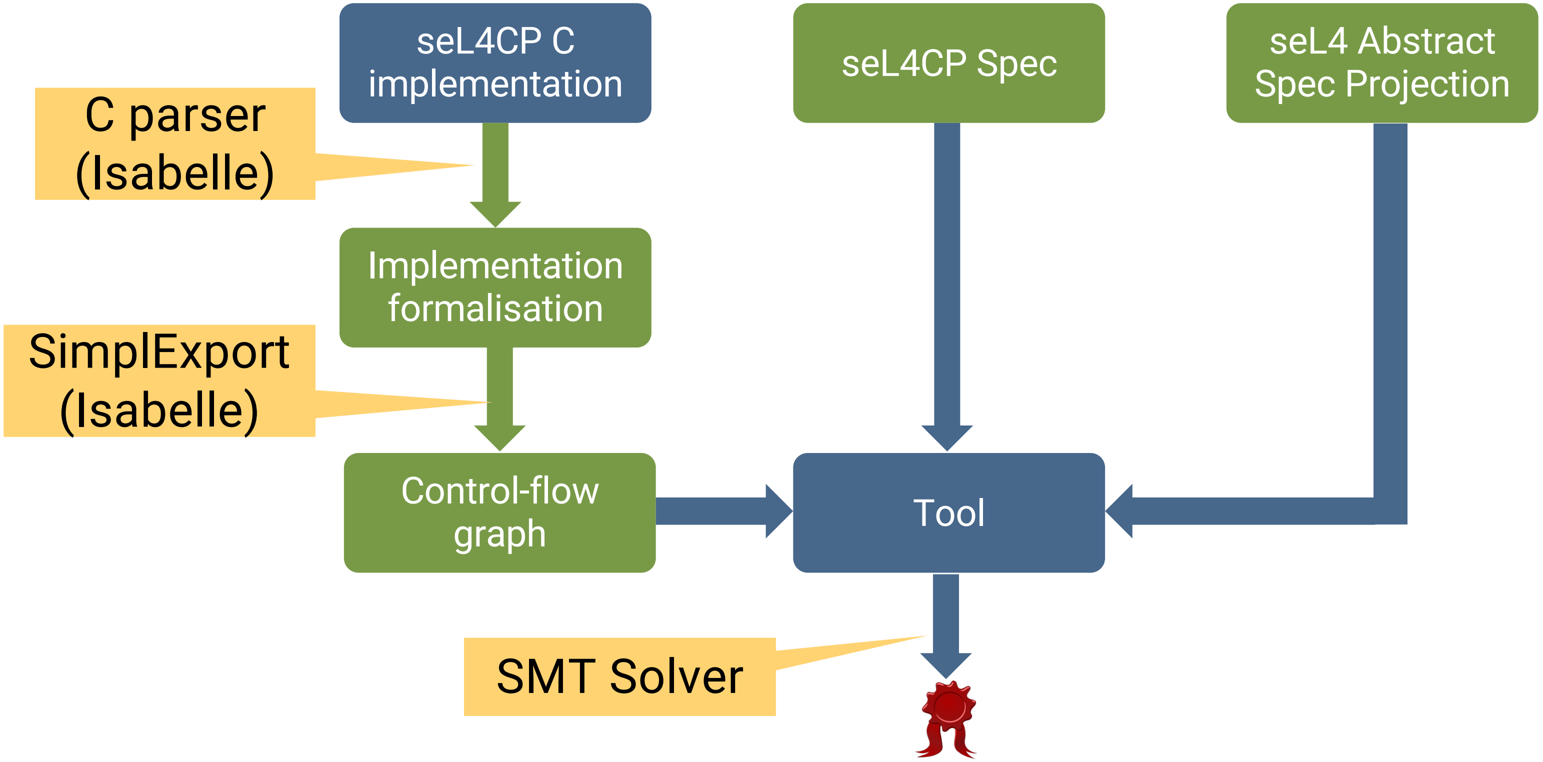
Conditions apply



Translation-validation (push-button proof)

Push-button proof

seL4CP Verification: libsel4cp



s4L4CP Verification in Context

Linux:

- NW driver: 4k lines
- NW system total: 1M lines

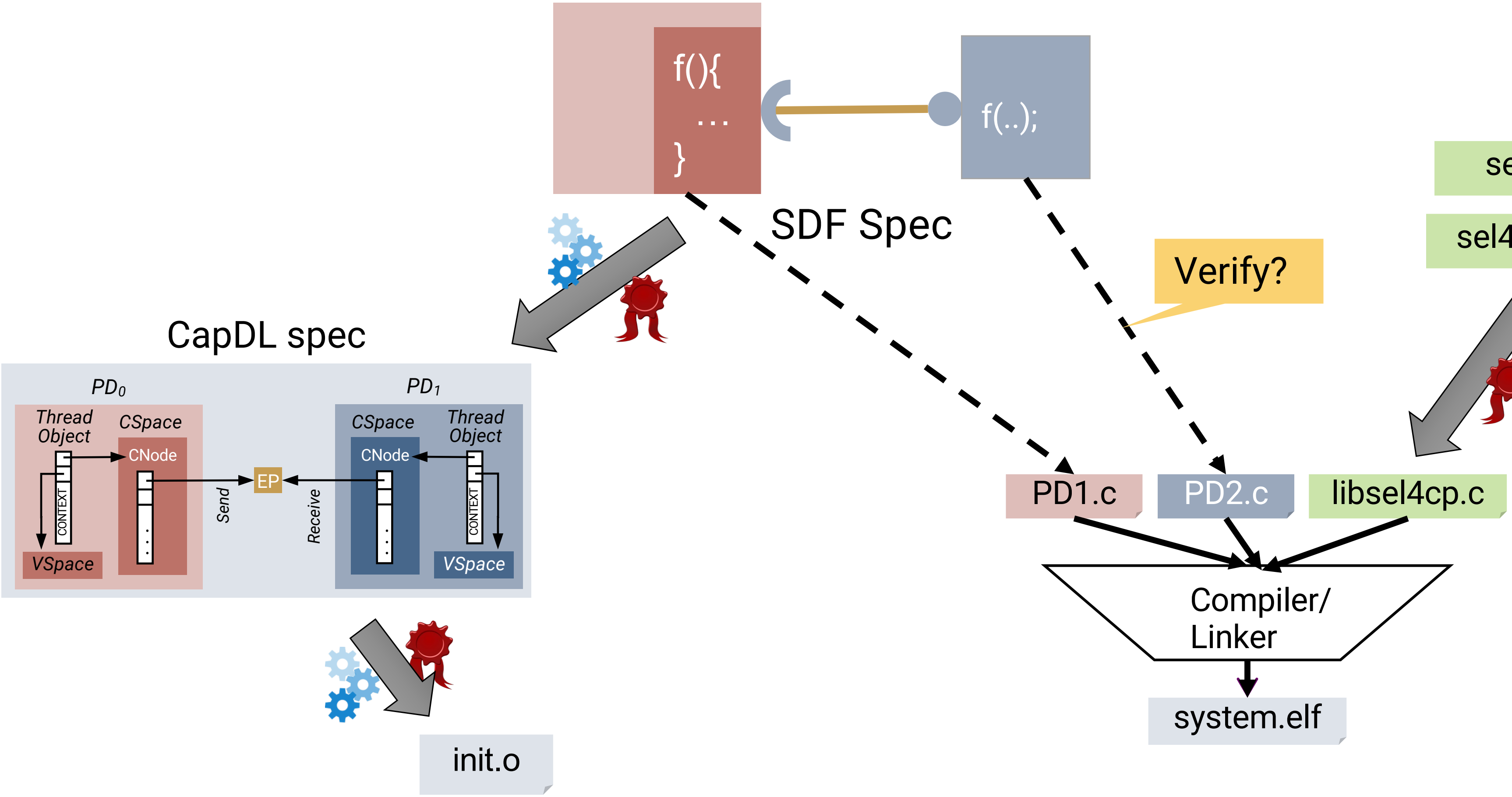
KISS design:

- NW driver: 700 lines
- MUX: 400 lines
- Copier: 200 lines
- IP stack: much simpler, client library
- shared NW system total: < 2,000 lines

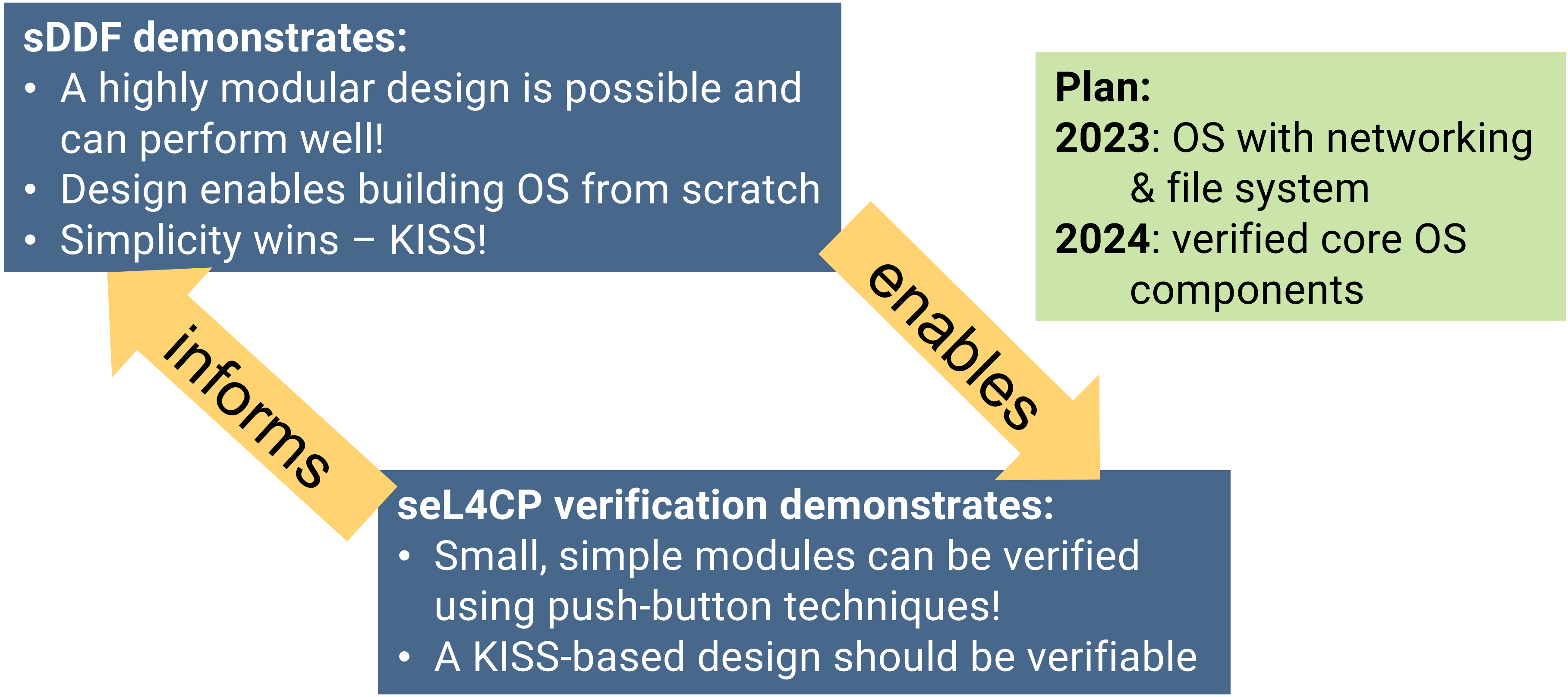
seL4CP:

- libsel4cp: 280 lines

seL4CP Verification



Stepping Back: sDDF and CP Verification



sDDF demonstrates:
- A highly modular design is possible and can perform well!
- Design enables building OS from scratch
- Simplicity wins – KISS!

Plan:
2023: OS with networking & file system
2024: verified core OS components

seL4CP verification demonstrates:
- Small, simple modules can be verified using push-button techniques!
- A KISS-based design should be verifiable

Security is no excuse for bad performance!

https://trustworthy.systems