School of Computer Science & Engineering

Trustworthy Systems Group

Can We Put The “S” Into IoT?

Gernot Heiser, Lucy Parker, Peter Chubb, Ivan Velickovic, Ben Leslie
gernot@unsw.edu.au

Securing IoT

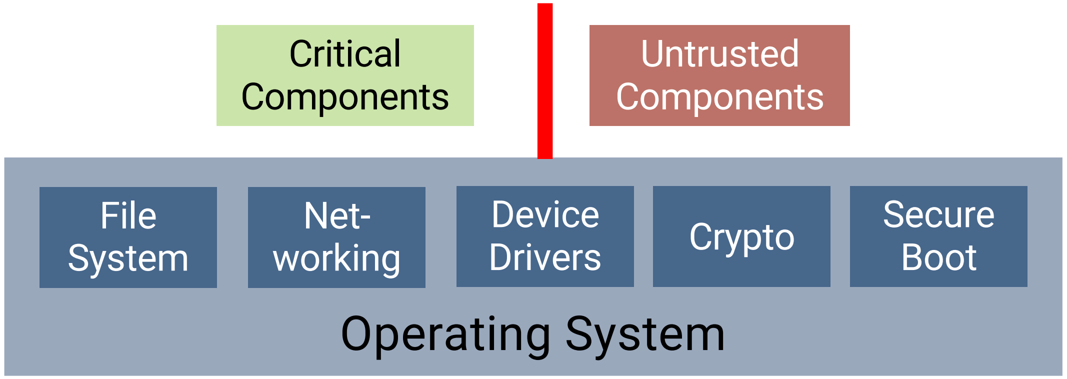
What’s Needed To Secure IoT Systems?

IoT Systems:
- limited mission-critical functionality
- incorporate much legacy code
- limited developer expertise
- operate unattended for a long time
- fleets of large number of devices

Not trustworthy

Must be easy to develop

SW updates costly

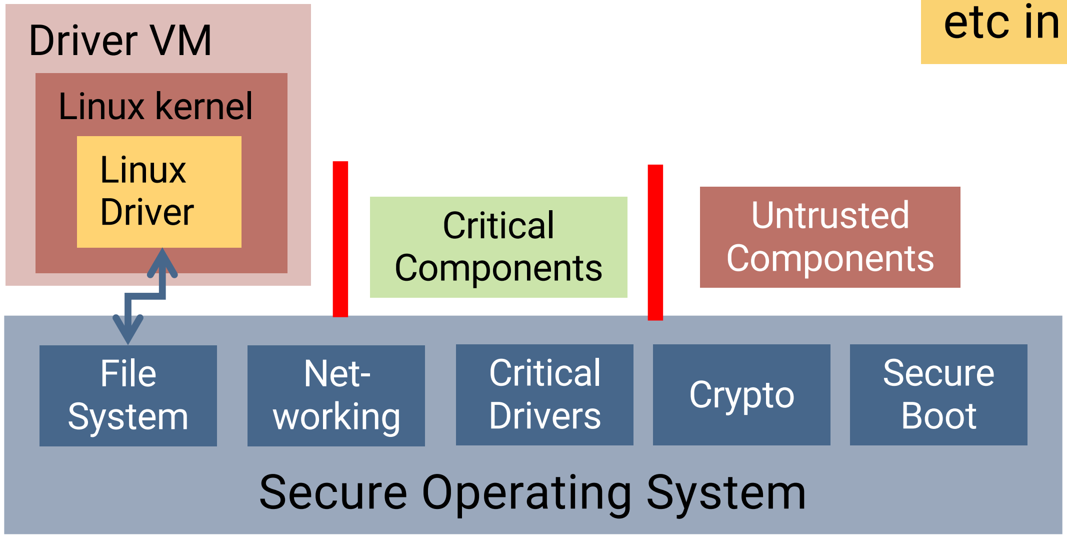


OS requirements:
- developer friendly
- hard to compromise
- enforce internal isolation
- secure virtualisation

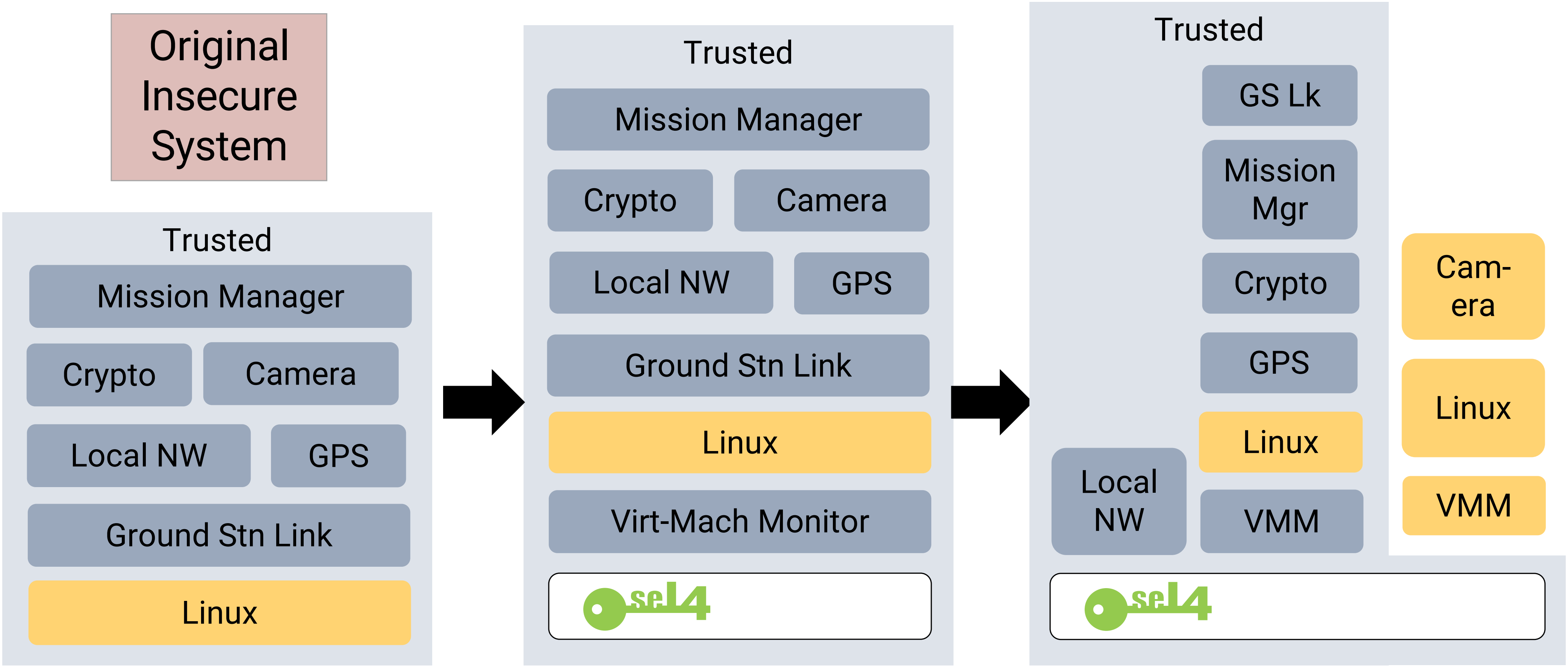
for legacy support

Virtualisation in IoT: Legacy Re-Use

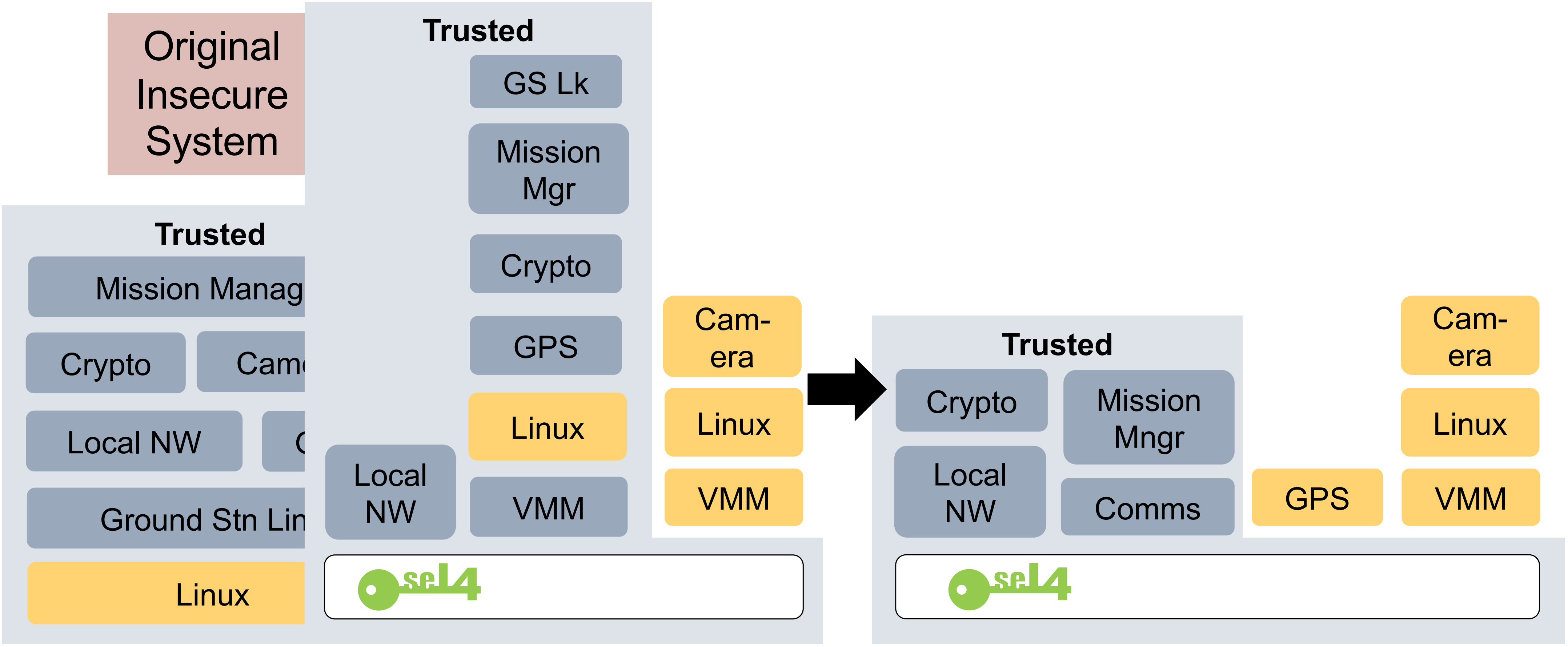
Enable re-use of unmodified legacy drivers, file systems, etc in deployed systems



Virtualisation: Incremental Cyber Retrofit



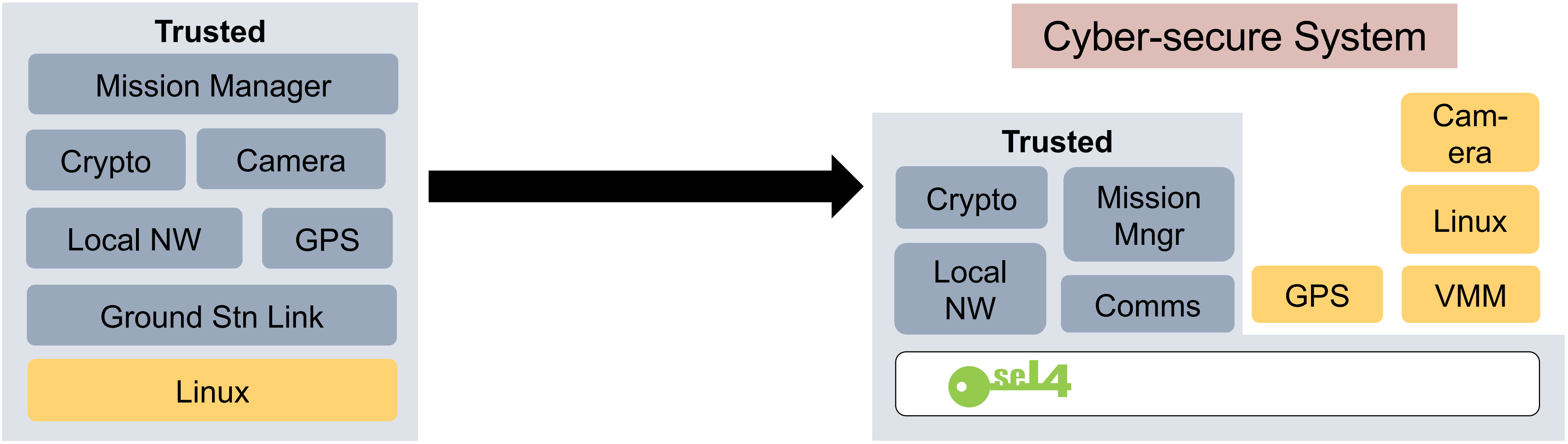
Virtualisation: Incremental Cyber Retrofit



Virtualisation: Incremental Cyber Retrofit

[Klein et al, CACM, Oct'18]

Original Insecure System



Trusted

Mission Manager

Crypto

Camera

Local NW

GPS

Ground Stn Link

Linux

Cyber-secure System

Trusted

Crypto

Mission Mngr

Local NW

Comms

GPS

Cam-era

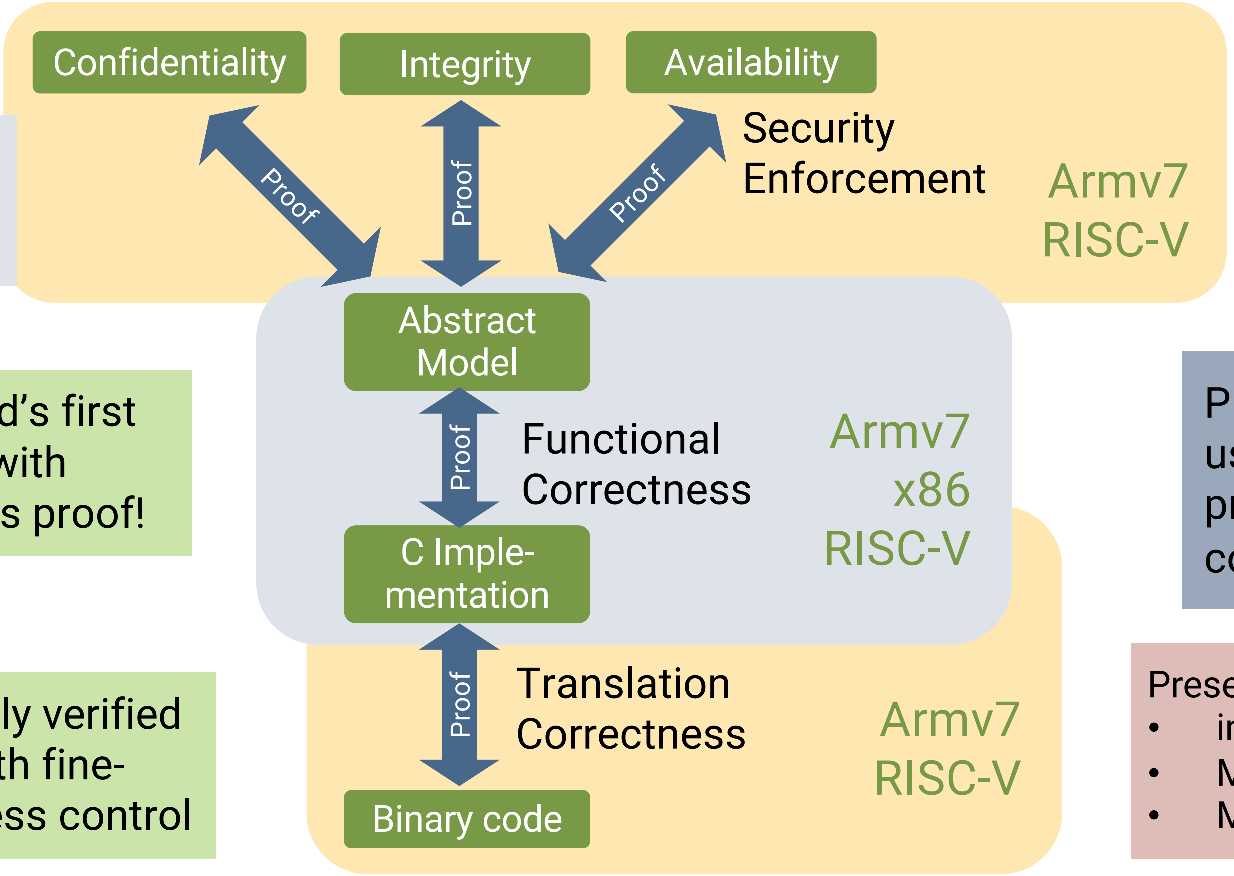
Linux

VMM

seL4

Secure OS For IoT

Foundation: Verified seL4 Microkernel



AArch64 in progress

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

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

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

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

Microkernel Is Not An OS

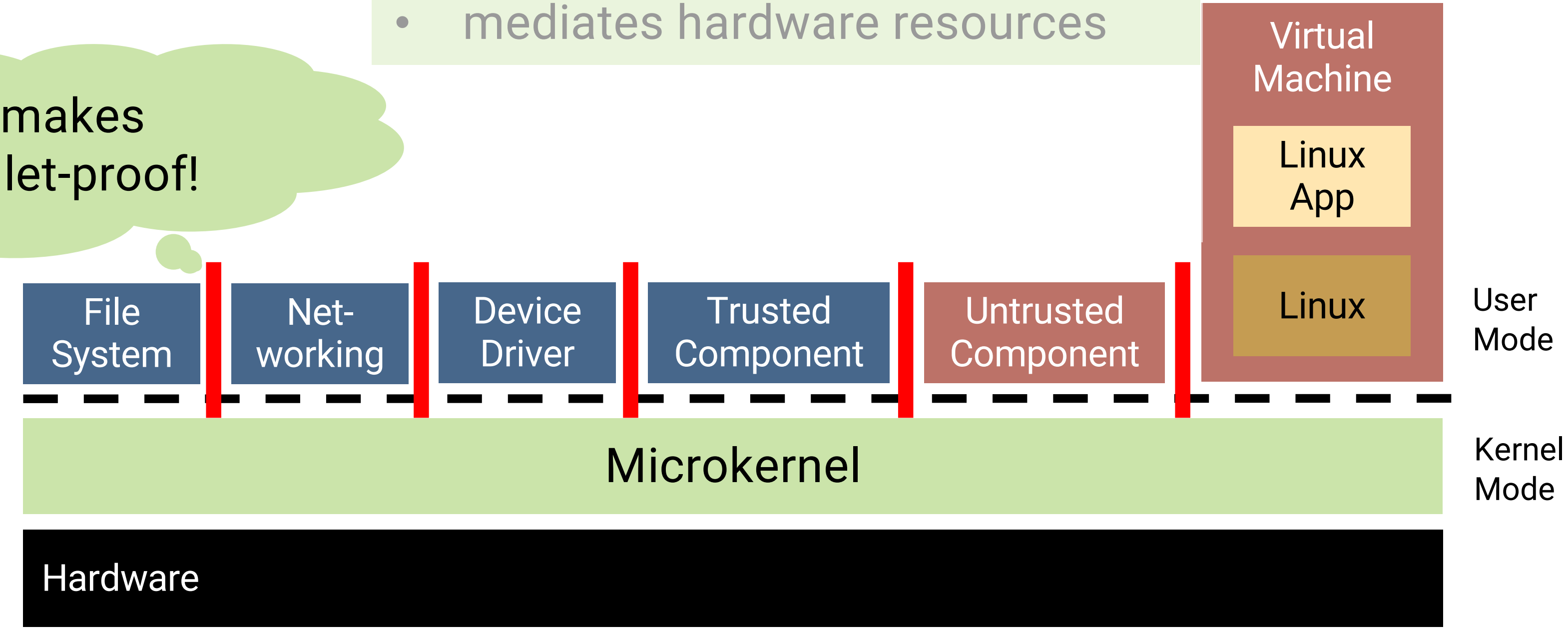
Modularisation: Separate components

- operating-system services
- applications

Microkernel enforces isolation

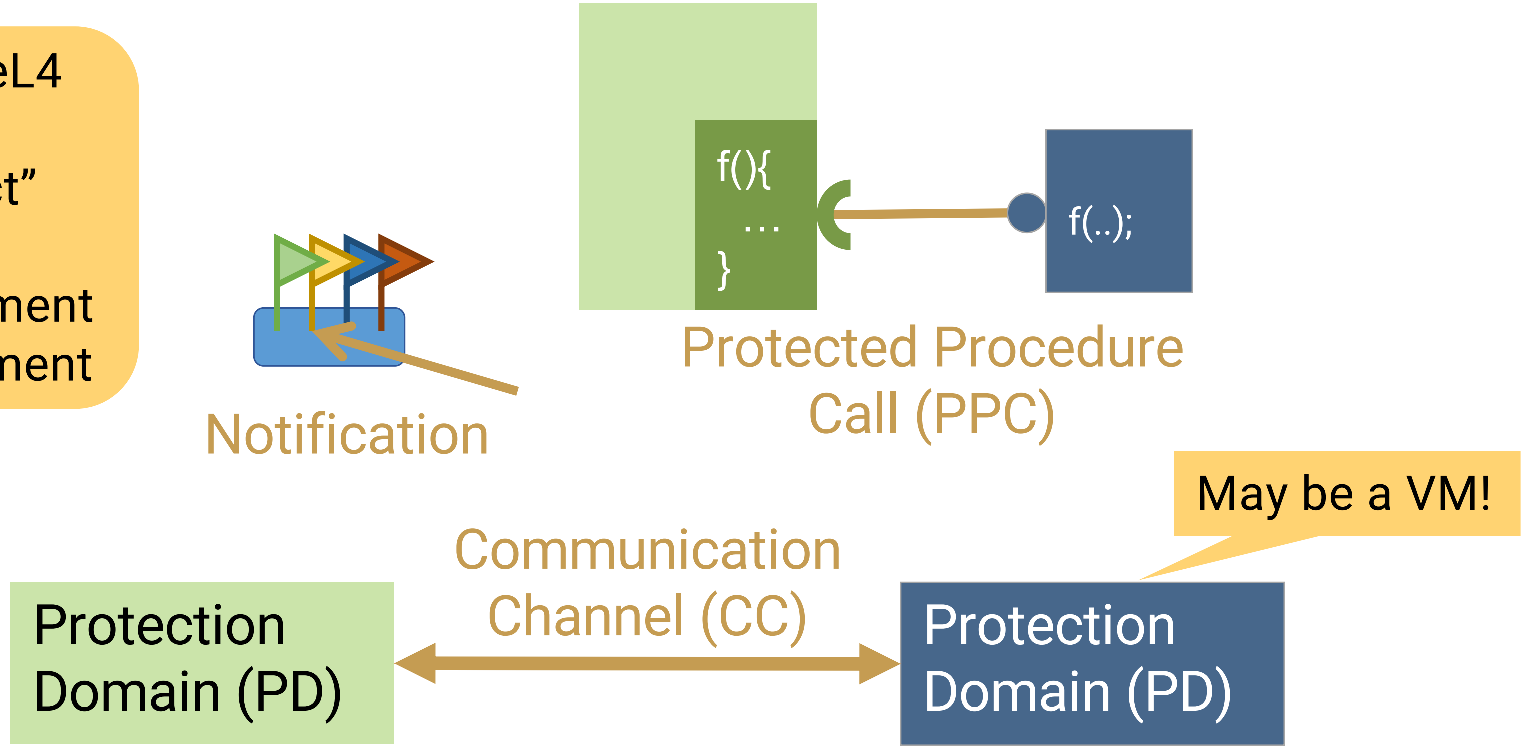
- kernel code reduced to minimum
- mediates hardware resources

Verification makes isolation bullet-proof!



OS Framework: seL4 Core Platform

- Thin wrapper of seL4 abstractions
- Encourage "correct" use of seL4
- Software development kit eases development



Notification

f(){
…
}

f(..);

Protected Procedure Call (PPC)

May be a VM!

Communication Channel (CC)

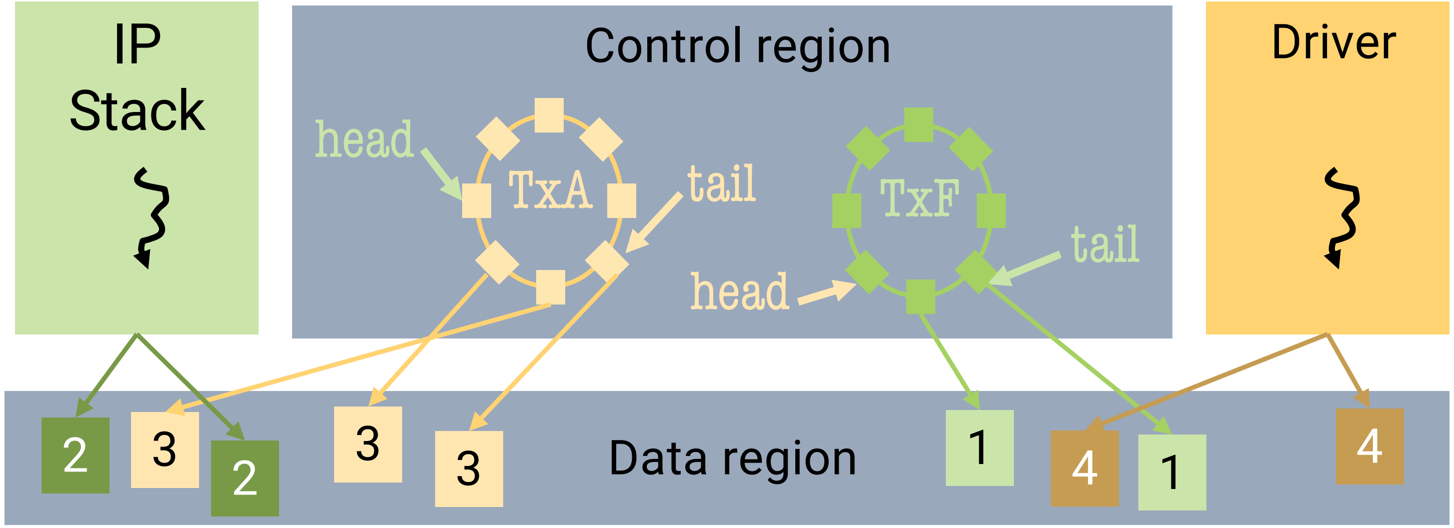
Protection Domain (PD)

Protection Domain (PD)

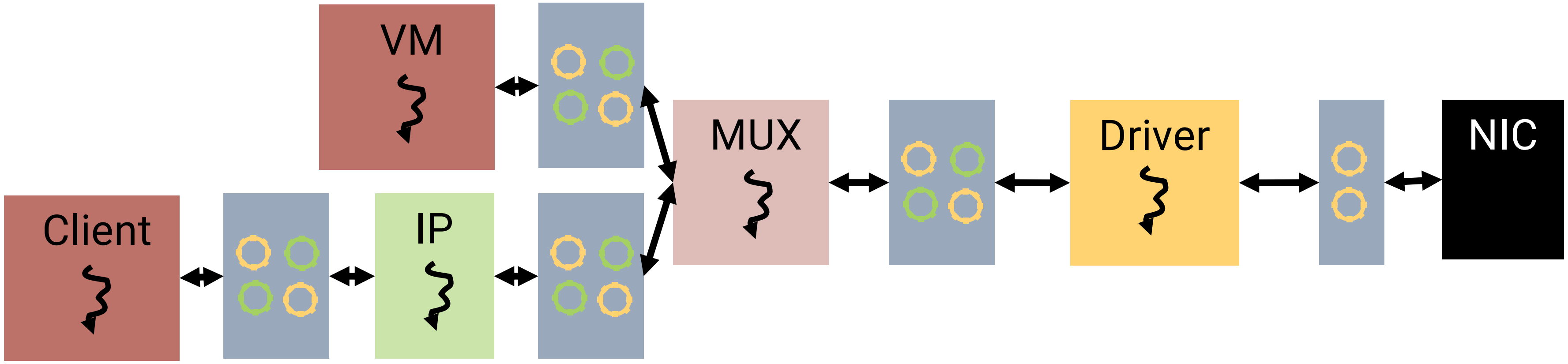
Memory Region (MR)

seL4 Device Driver Framework

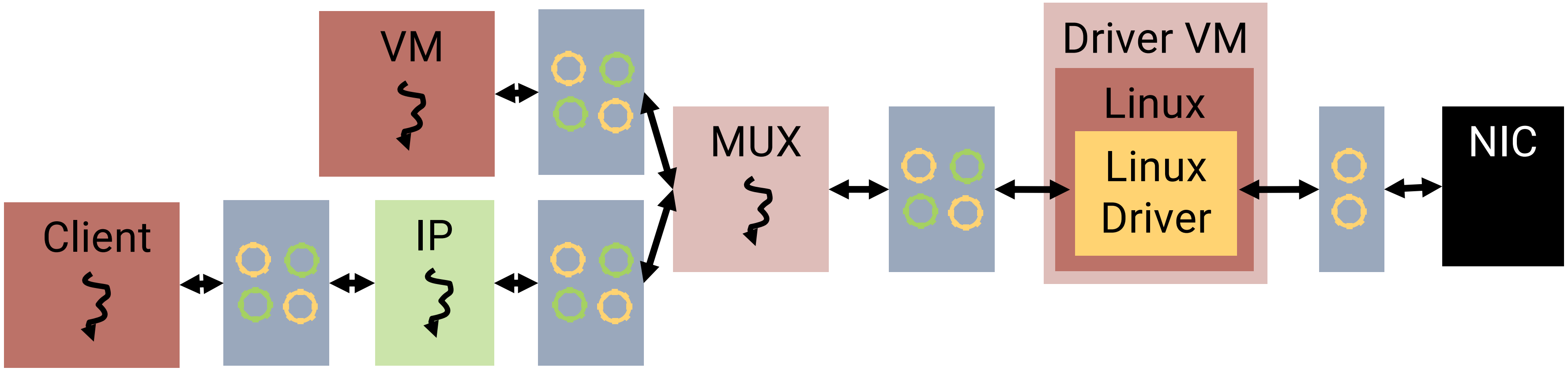
- Lightweight
- Simple, event-based, single-threaded drivers
- Asynchronous, zero-copy transport layer



Device Sharing

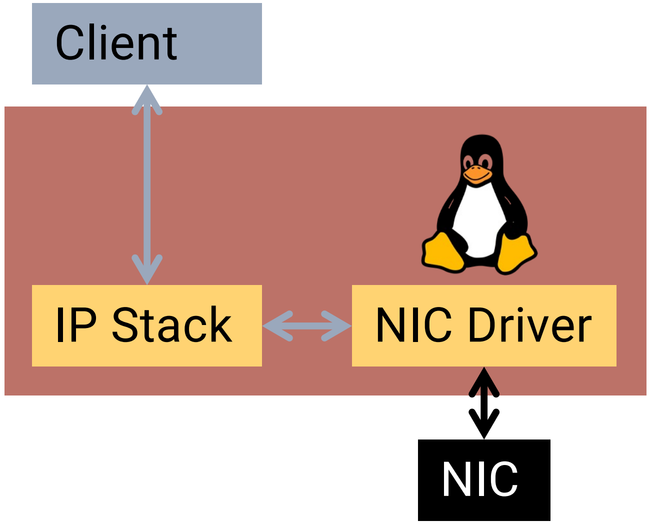


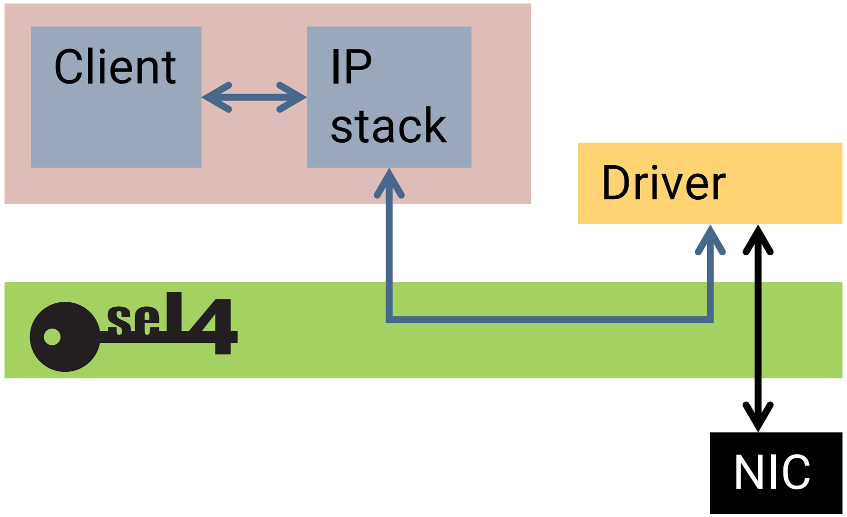
Device Sharing with Legacy Re-Use



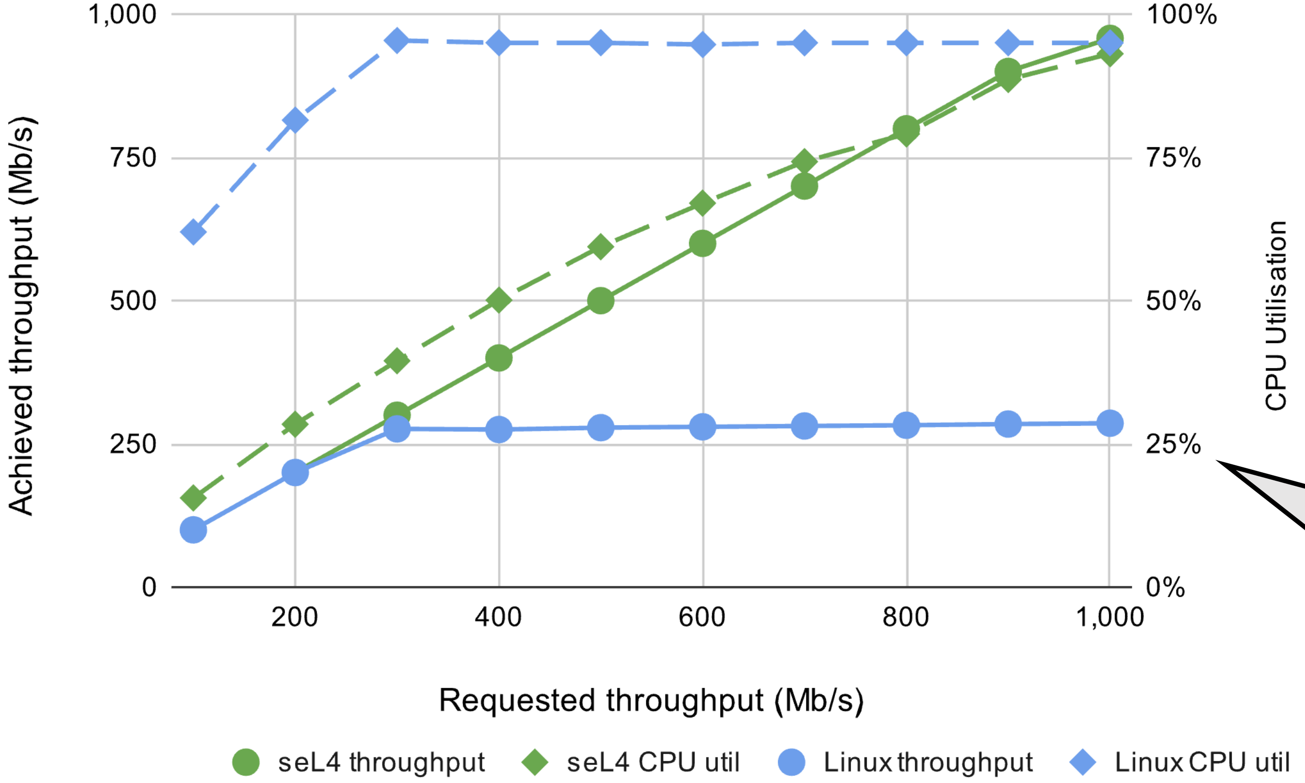
Does It Perform?

Evaluation Setup





seL4 vs Linux Networking Performance



3000 SLOC

vs

< 500 SLOC

Summary

- **seL4 is a rock-solid base for secure IoT**
  - … due to formal correctness & isolation proofs
- **The seL4 Core Platform makes seL4 easy to use**
  - Software development kit (SDK) for easy deployment
  - Simple abstractions, map onto “correct” usage of seL4
  - Virtual machines enable legacy re-use and incremental cyber retrofit
- **Highly modularised design with seL4-enforced module boundaries**
  - … provides security-by-design
- **Excellent performance despite modularisation**
  - … if well-designed
  - Significantly outperforms Linux on network performance

Defining the state of the art in trustworthy systems since 2009