School of Computer Science & Engineering

Trustworthy Systems Group

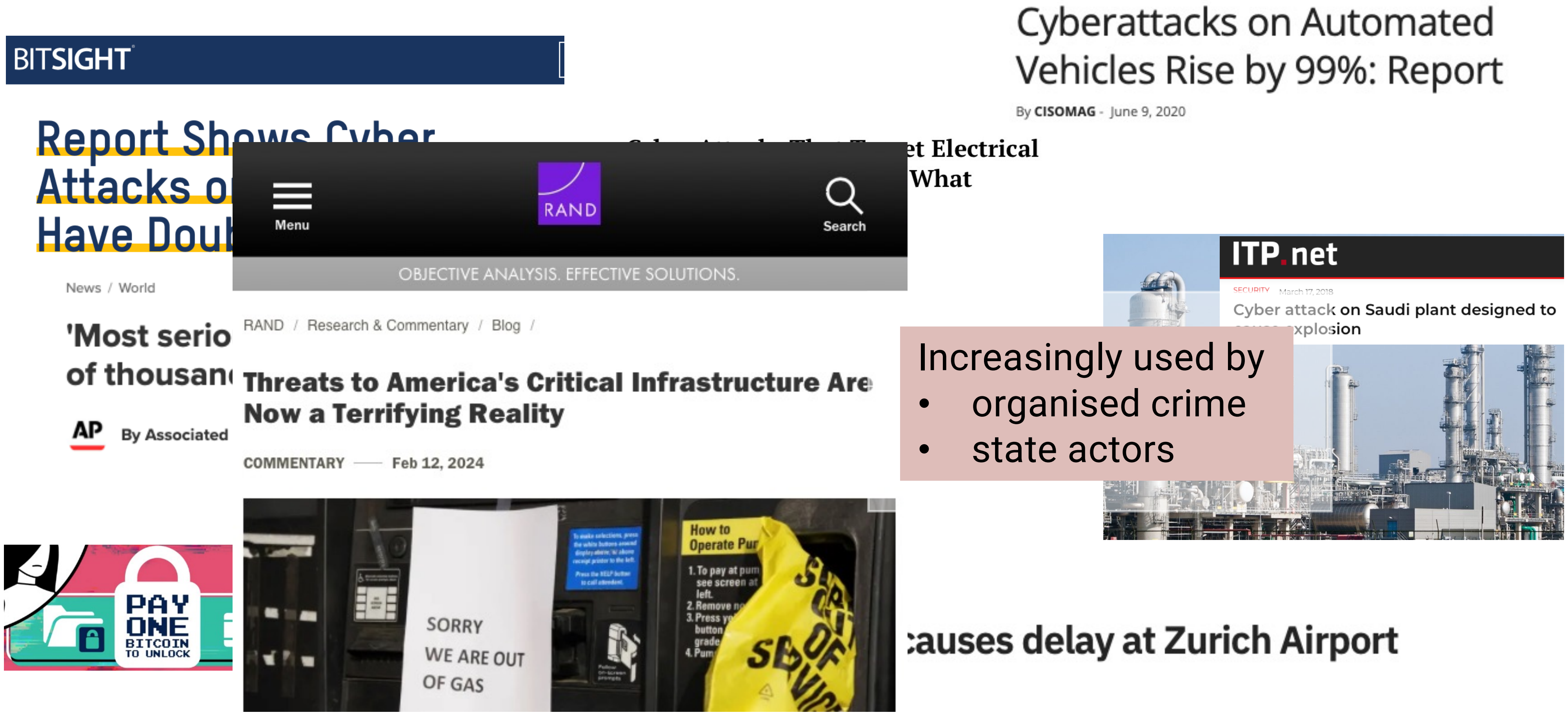
LionsOS

A Highly Dependable Operating System for Cyberphysical Systems

Gernot Heiser

gernot@unsw.edu.au
@gernot@discuss.systems
https://microkerneldude.org/

Cyberattacks Are Everywhere



BITSIGHT

Report Shows Cyber
Attacks o
Have Doub

News / World

'Most serio
of thousan

AP By Associated

Cyberattacks on Automated Vehicles Rise by 99%: Report

By CISOMAG - June 9, 2020

et Electrical
What

RAND

OBJECTIVE ANALYSIS. EFFECTIVE SOLUTIONS.

RAND / Research & Commentary / Blog /

Threats to America's Critical Infrastructure Are Now a Terrifying Reality

COMMENTARY —— Feb 12, 2024

ITP.net

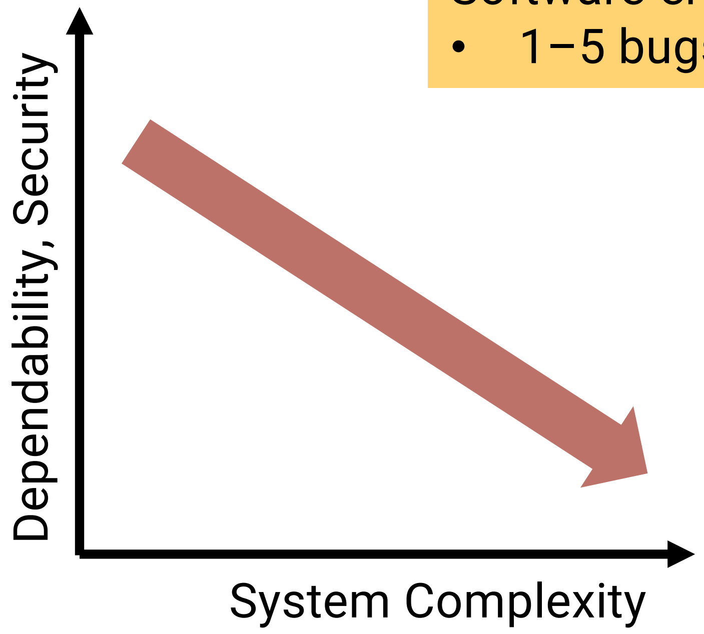
SECURITY March 17, 2018

Cyber attack on Saudi plant designed to
xplosion

:auses delay at Zurich Airport

Increasingly used by
- organised crime
- state actors

Core Problem: Complexity



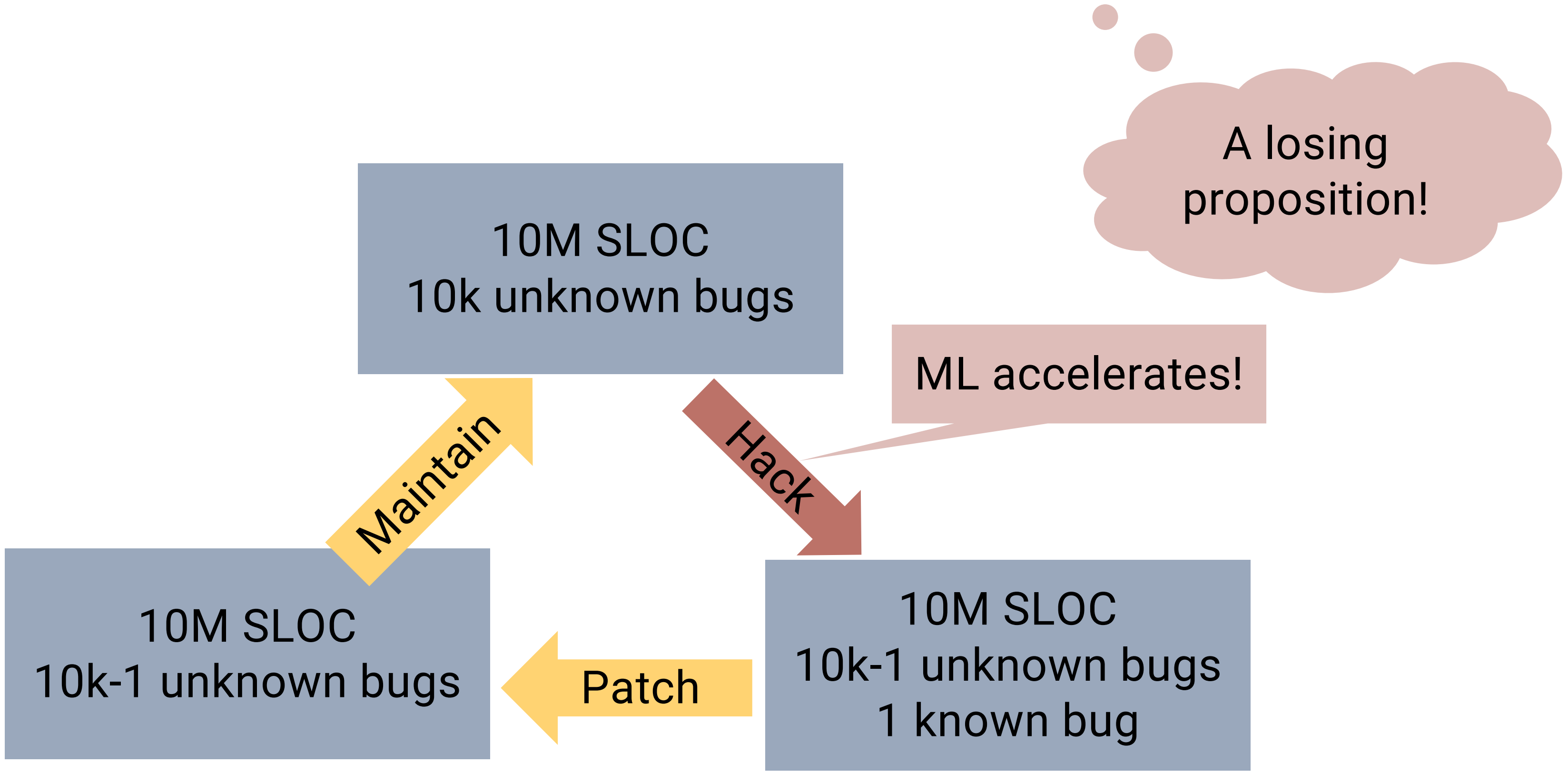
Software-engineering rule of thumb:

- 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



We Need To Do Better!

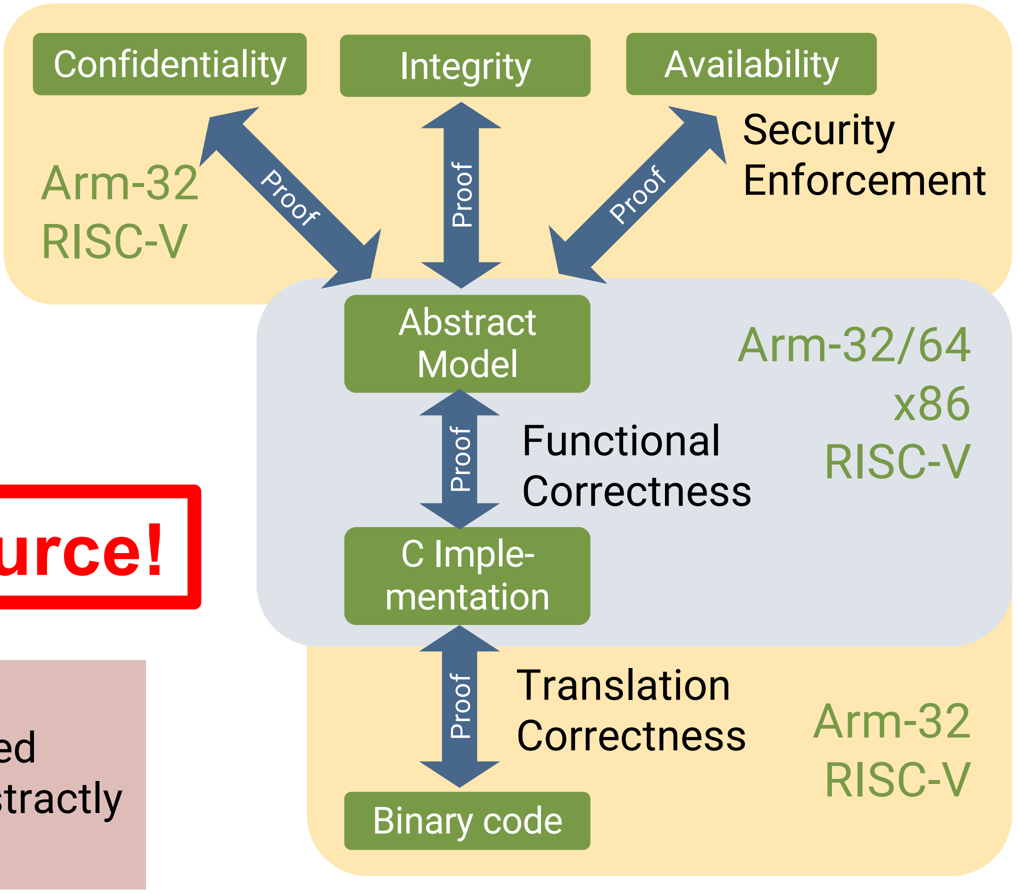
We need principled solutions based on solid foundations!

Mathematically Proved Secure

- Comprehensive formal verification
- Only verified OS with fine-grained protection (capabilities)
- Only protected-mode RTOS with sound and compete WCET analysis
- World's fastest microkernel

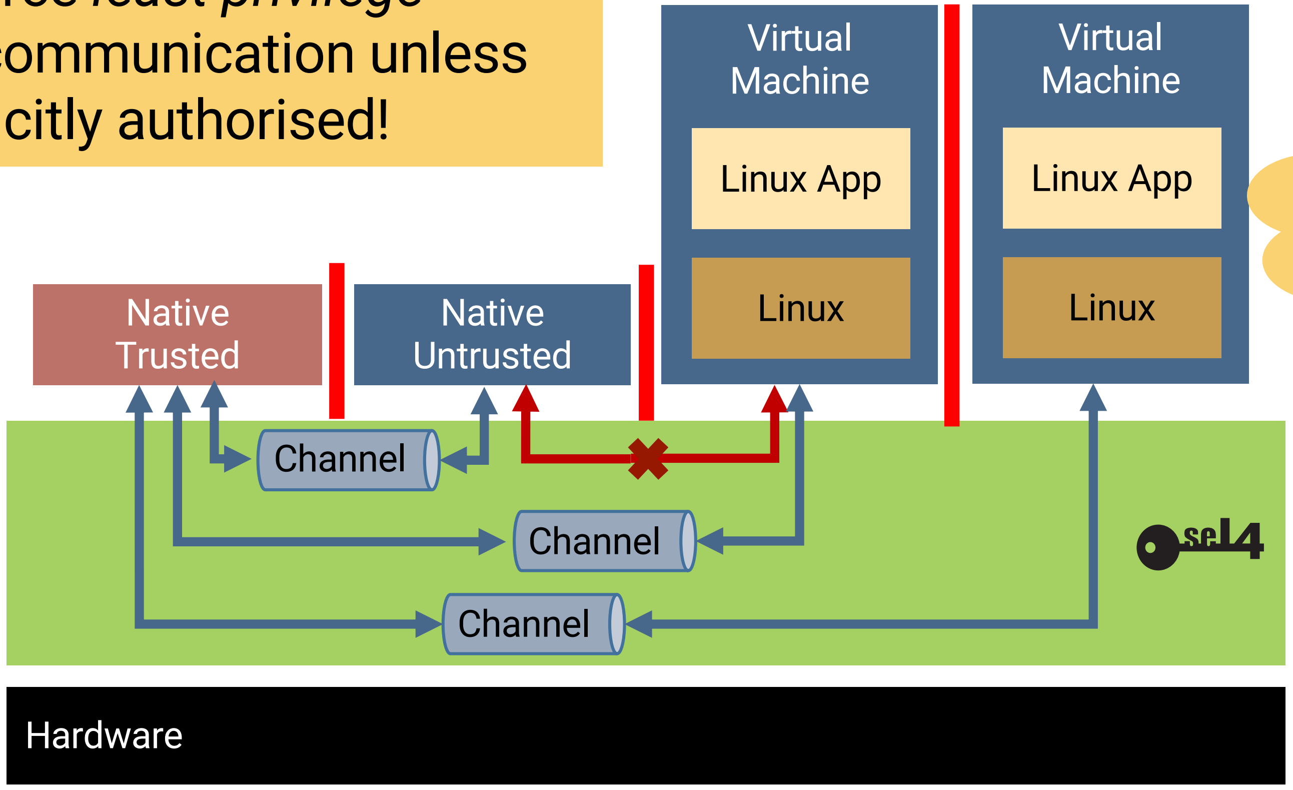
Open Source!

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!



No capabilities? You're not serious about security!

The Benchmark for Performance

Round-trip cross-address-space IPC on 64-bit Intel Skylake

	seL4	Fiasco.OC aka L4Re	Google Zircon
Latency (cycles)	986	2717	8157
Mandatory HW cost* (cycles)	790	790	790
Overhead absolute (cycles)	196	1972	7367
Overhead relative	25%	240%	930%

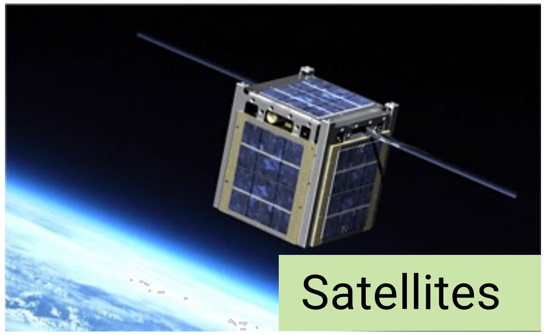
Smaller is better

World's fastest microkernel!

***:** The Cost of `SYCALL` + 2 × `SWAPGS` + `SYSRET` = 395 cycles, times 2 for round-trip
Source:
Zeyu Mi, Dingji Li, Zihan Yang, Xinran Wang, Haibo Chen: "SkyBridge: Fast and Secure Inter-Process Communication for Microkernels", EuroSys, April 2019

Used in Real-World Systems

Autonomous vehicles

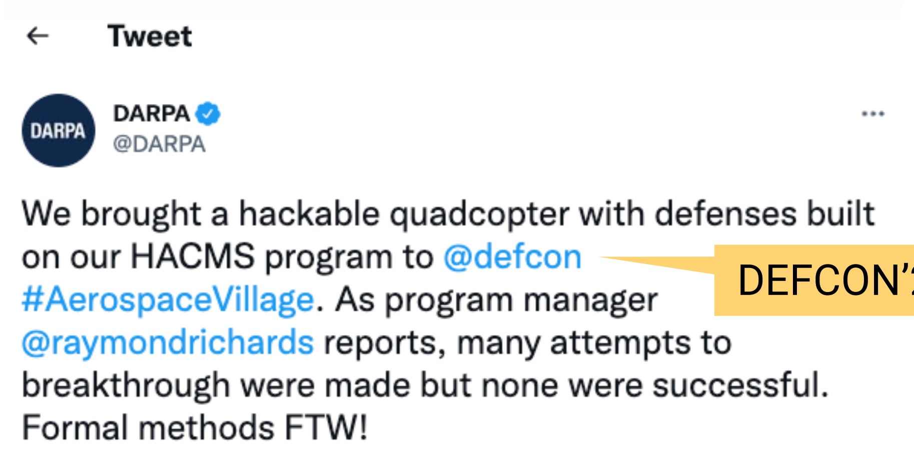

Satellites

Critical infrastructure protection

Secure communication device
In use in multiple defense forces

Cars

“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!

DEFCON’22

Why Aren’t We Done Yet?

A Microkernel

Microkernel:
- OS code that must execute in privileged mode
- Everything else belongs in user mode servers
- Servers are subject to the microkernel's security enforcement!

Consequence:
- Small: 10 kLOC
- Only fundamental, policy-free mechanisms
- No application-oriented services/abstractions
- **BYO file system, memory manager, device drivers**

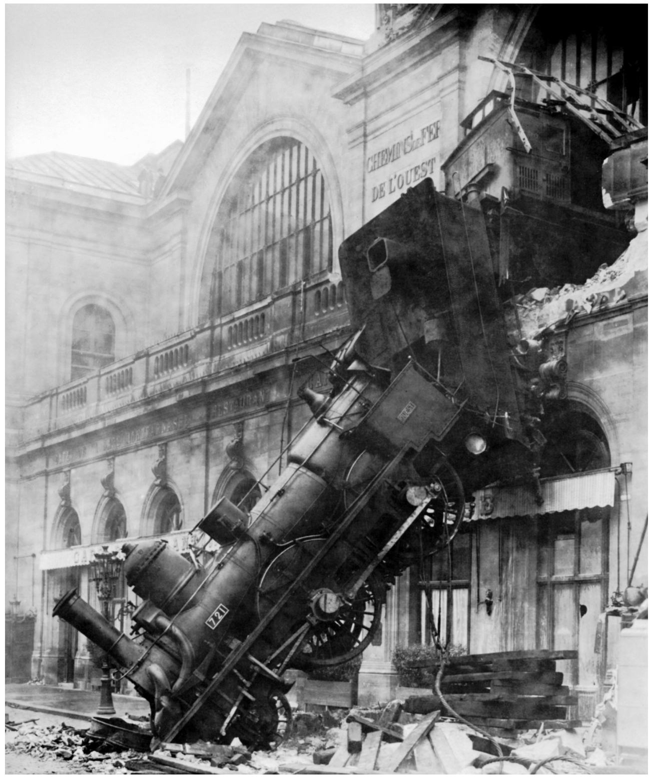
seL4 Experience of the First 10+ Years

seL4's assurance and power is unrivalled, but:

- good design of seL4-based systems requires deep expertise
- a secure microkernel doesn't guarantee a secure system

seL4 needs an OS that:

- provides "usual" OS services
- is easy to use
- is performant
- **is secure**



Enter LionsOS

Stop The Train Wrecks!

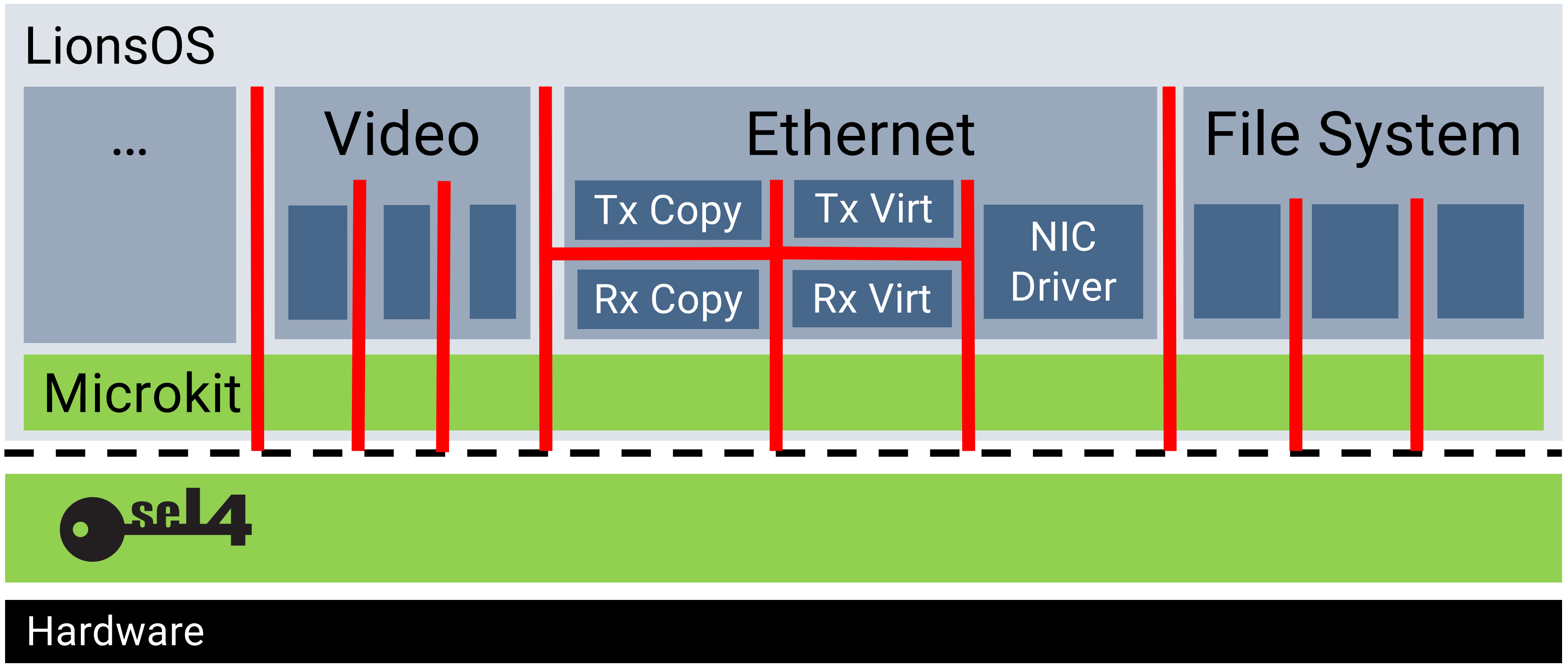
Lions OS: Secure, Fast, Adaptable

Aim 1: *Practical, easy-to-use, open-source* OS for wide range of embedded/IoT/cyberphysical use cases

Aim 2: *Best-performing* microkernel-based OS ever

Aim 3: *Provably secure* OS

LionsOS: Highly Modular OS



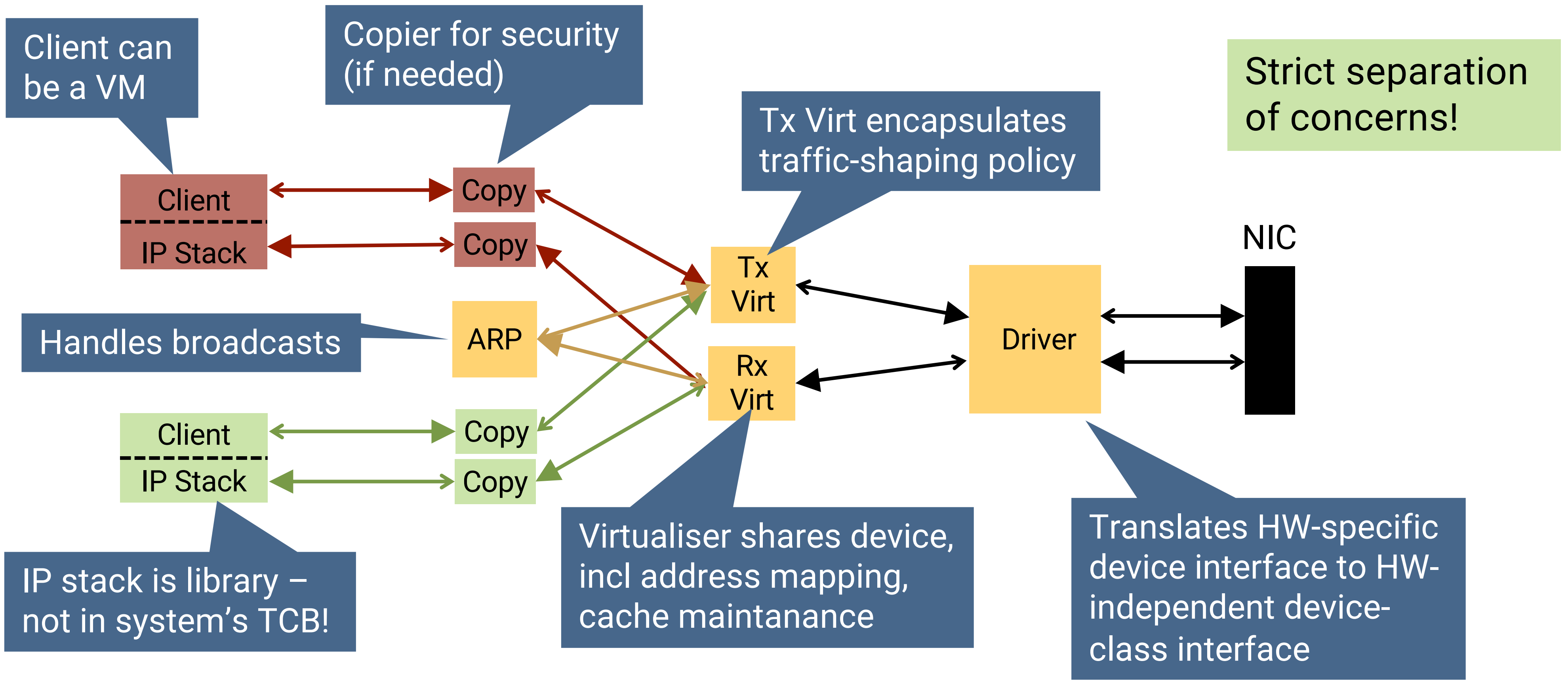
LionsOS Design Principle: KISS

Keep it simple, stupid!
- Strong separation of concerns
- Simplest implementation possible
- Least privilege

Implications:
- Use-case-specific instead of “universal” policies
- Use-case diversity by swapping policy modules!

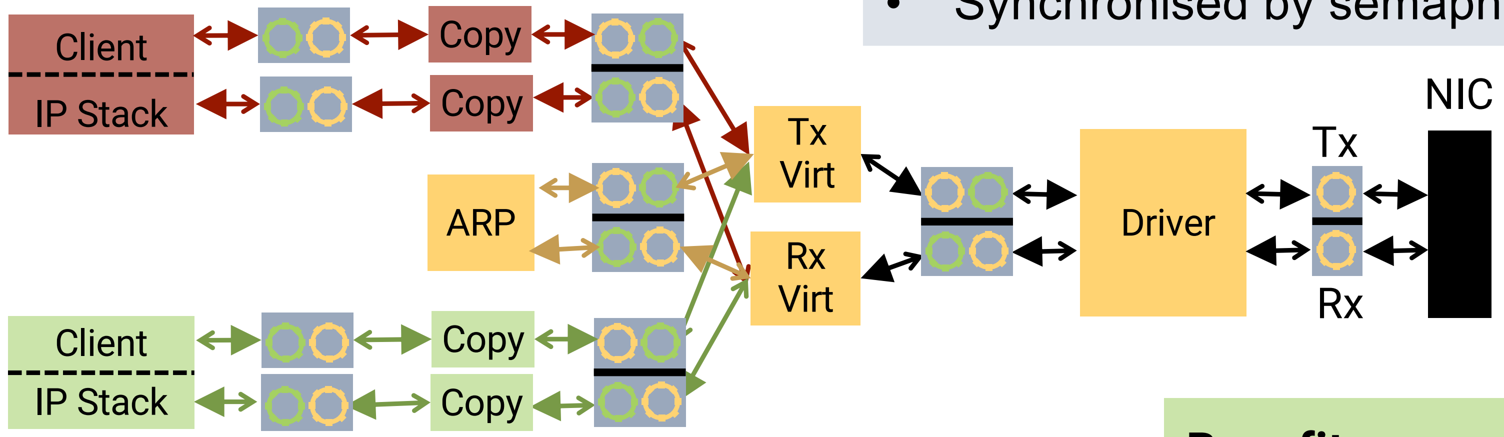
Simple programming model:
- “Microkit” abstraction layer
- Event-driven programming model
- Static architecture
- Location-transparent components

Example: Networking Subsystem


Client can be a VM
Copier for security (if needed)
Tx Virt encapsulates traffic-shaping policy
Strict separation of concerns!
Client
IP Stack
Copy
Copy
Tx Virt
NIC
Handles broadcasts
ARP
Driver
Rx Virt
Client
IP Stack
Copy
Copy
IP stack is library – not in system's TCB!
Virtualiser shares device, incl address mapping, cache maintanance
Translates HW-specific device interface to HW-independent device-class interface

Networking Detail

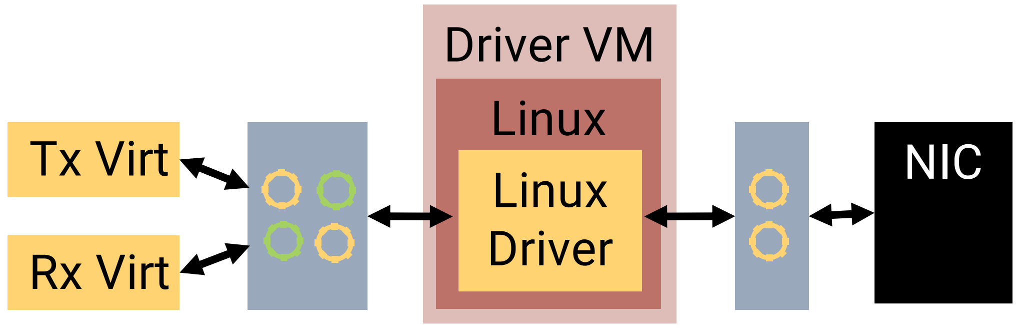
Zero-copy communication:
- Lock-free, single-producer, single-consumer, bounded queues
- Synchronised by semaphores



Benefits:
- simple components
- location transparency
- suitable for verification

Legacy Re-use: Driver VMs

Can re-use unmodified Linux drivers:
- Transparently use driver VM instead of native driver
- develop Lions-OS components on Linux



Comparison to Linux on i.MX8M (Armv8)

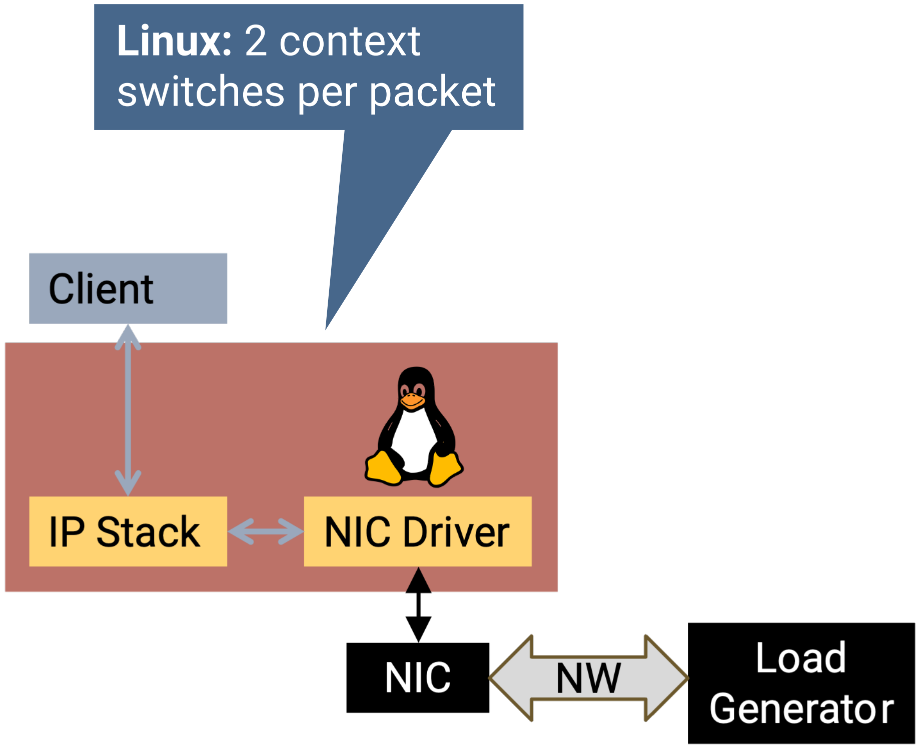
Linux:
- NW driver: 4k lines
- NW system total: 1M lines

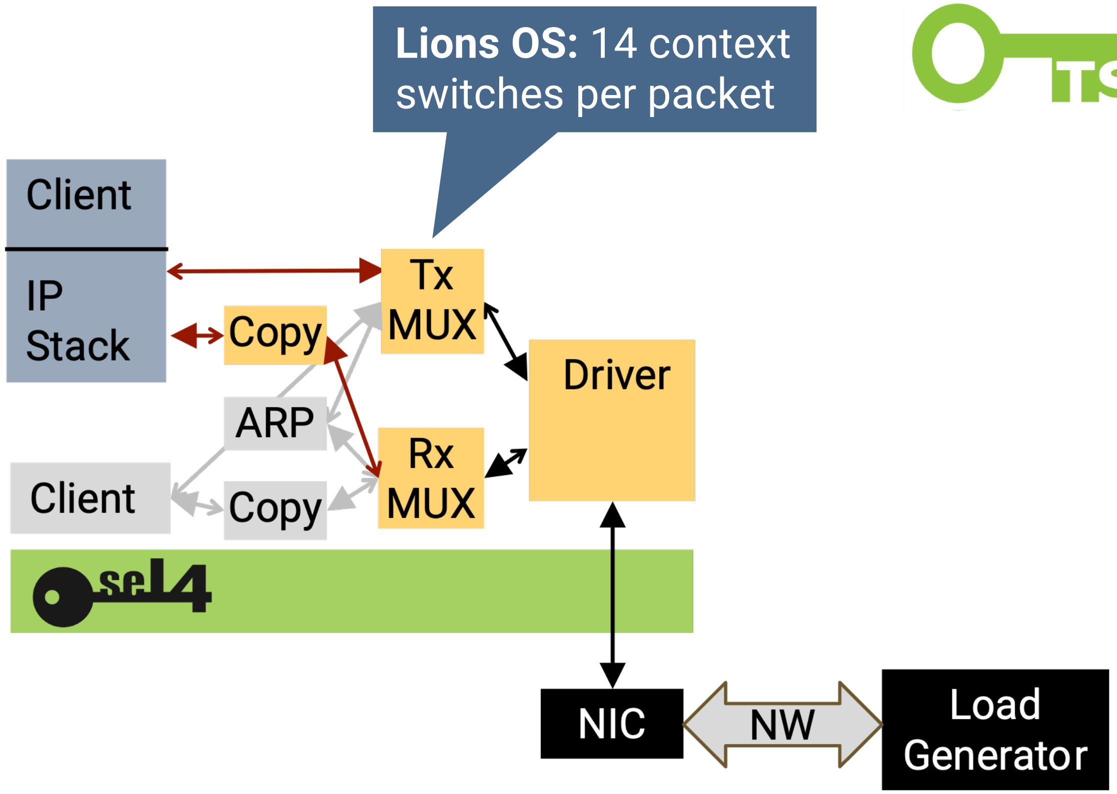
Performance?

Written by second-year student!

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

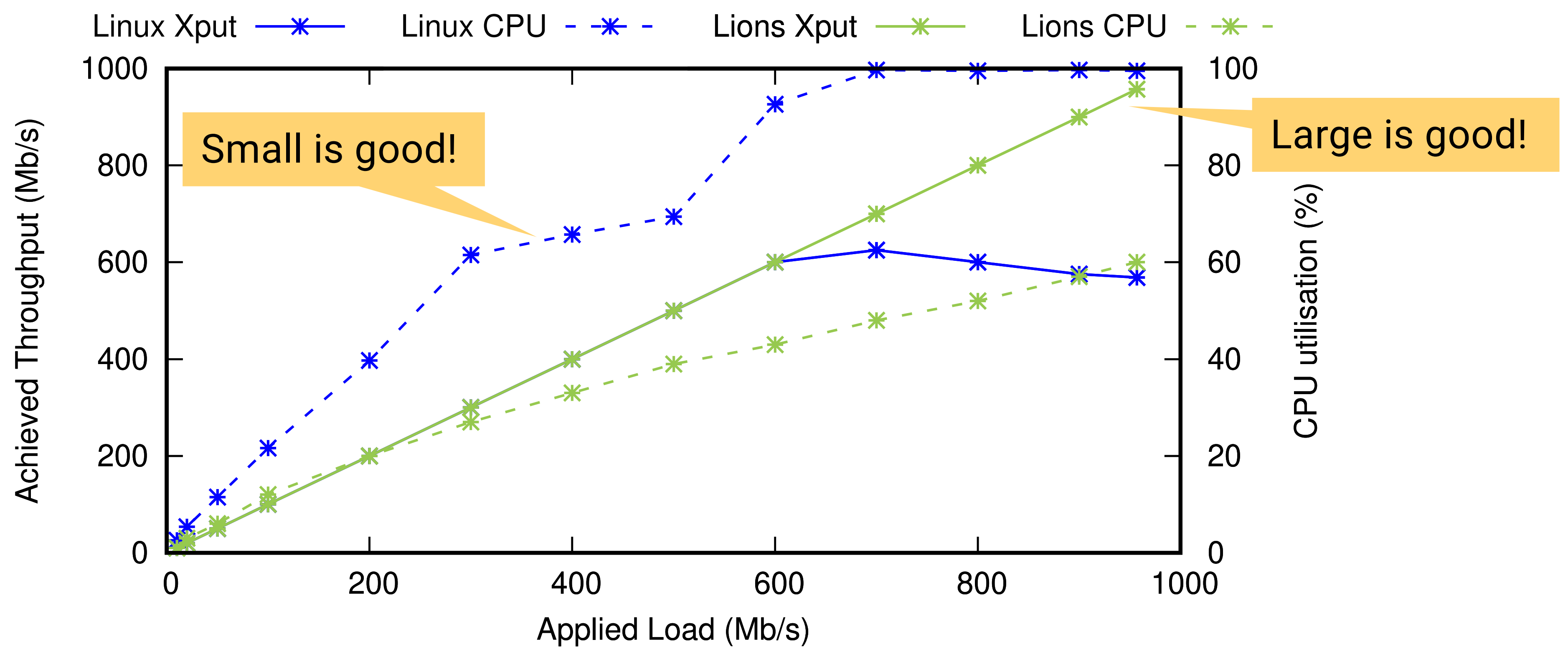
Evaluation Setup





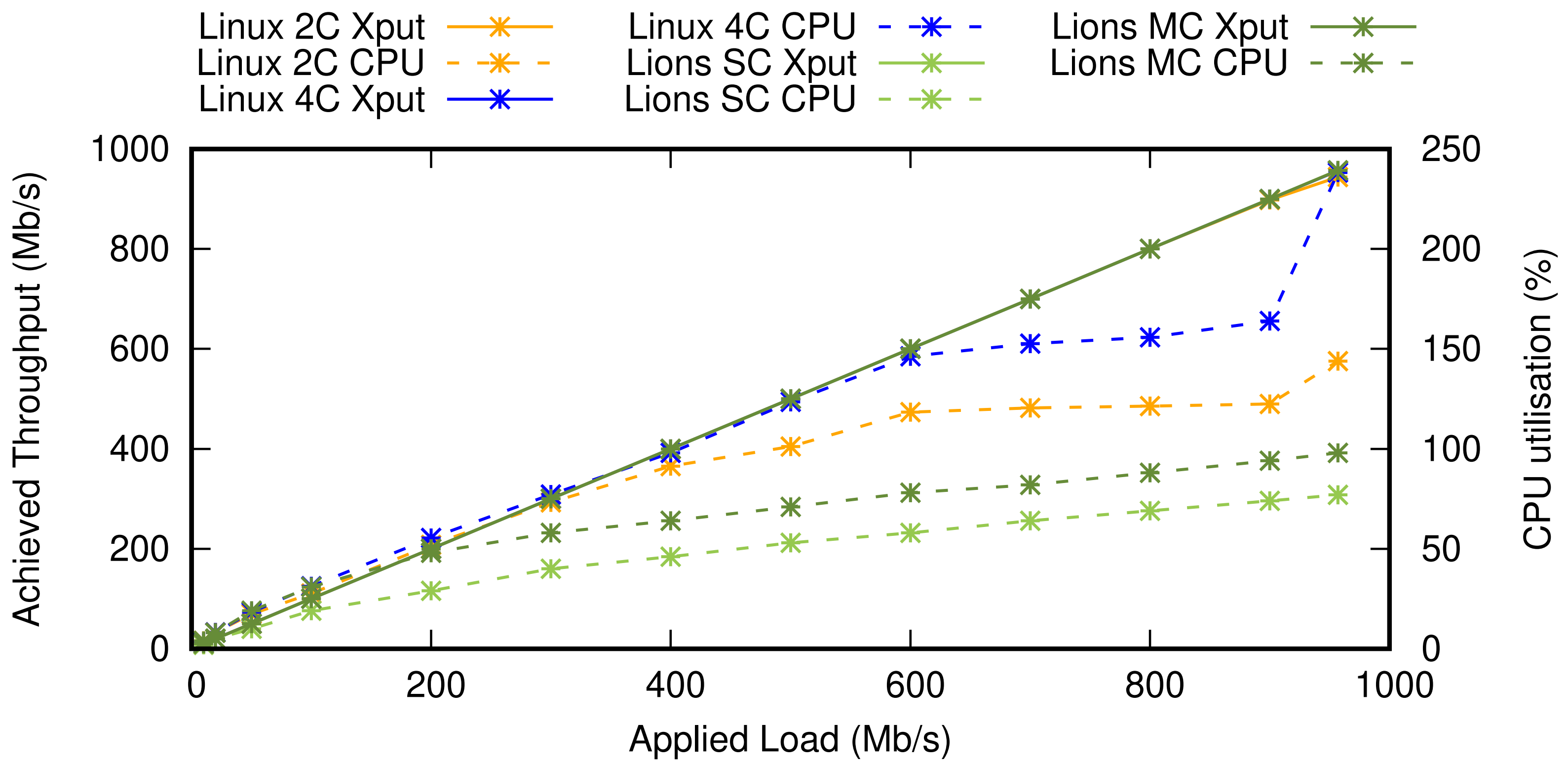
- External load generator
- Measures throughput, latency
- Client echoes packets

Performance: i.MX8M, 1Gb/s Ethernet



Single-core configuration

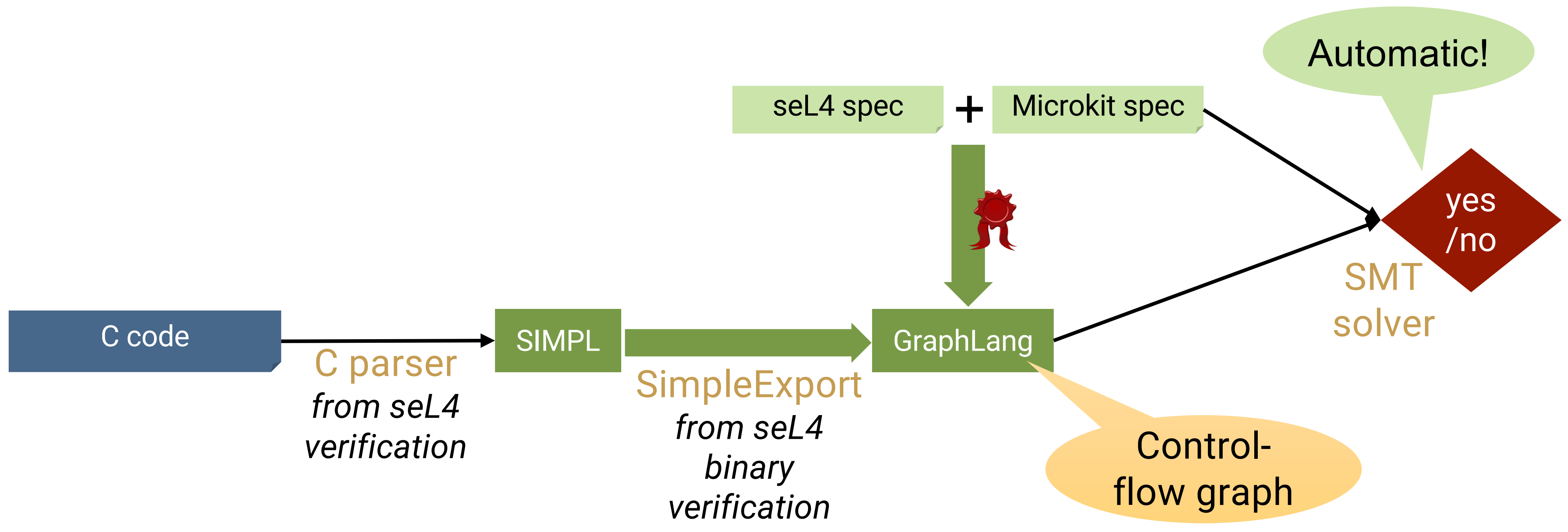
Performance: i.MX8M, 1Gb/s Ethernet



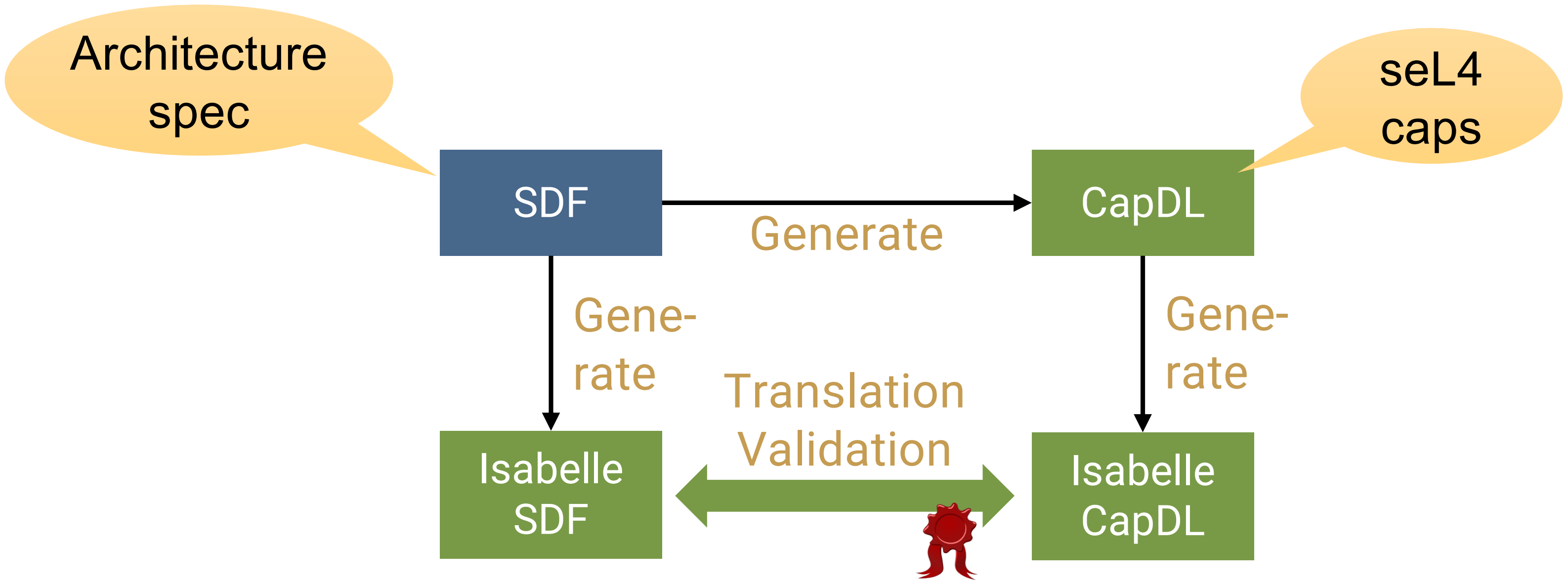
Multicore configuration

How About Verification?

Verifying the Microkit: libmicrokit



Verifying the Microkit: System Initialisation



Verifying LionsOS

- Microkit programming model:
  - simple event handlers
  - strictly sequential code

Very little time spent on debugging component logic

Suitable for SMT solvers

- Fine-grained modularity:
  - concurrency by distribution, "tamed" concurrency
  - complex signalling protocols

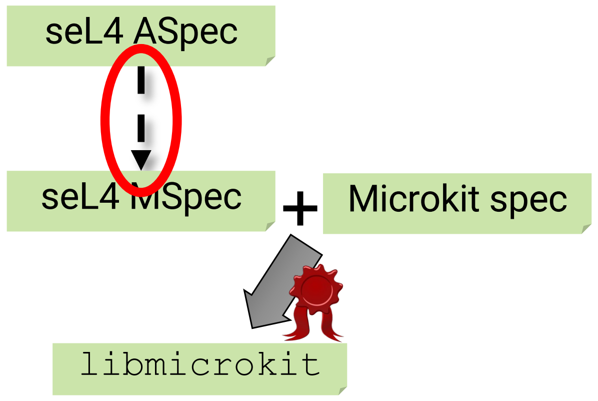
Protocol bugs are mostly performance problems

Ideal for model checking!

Automatic proofs!

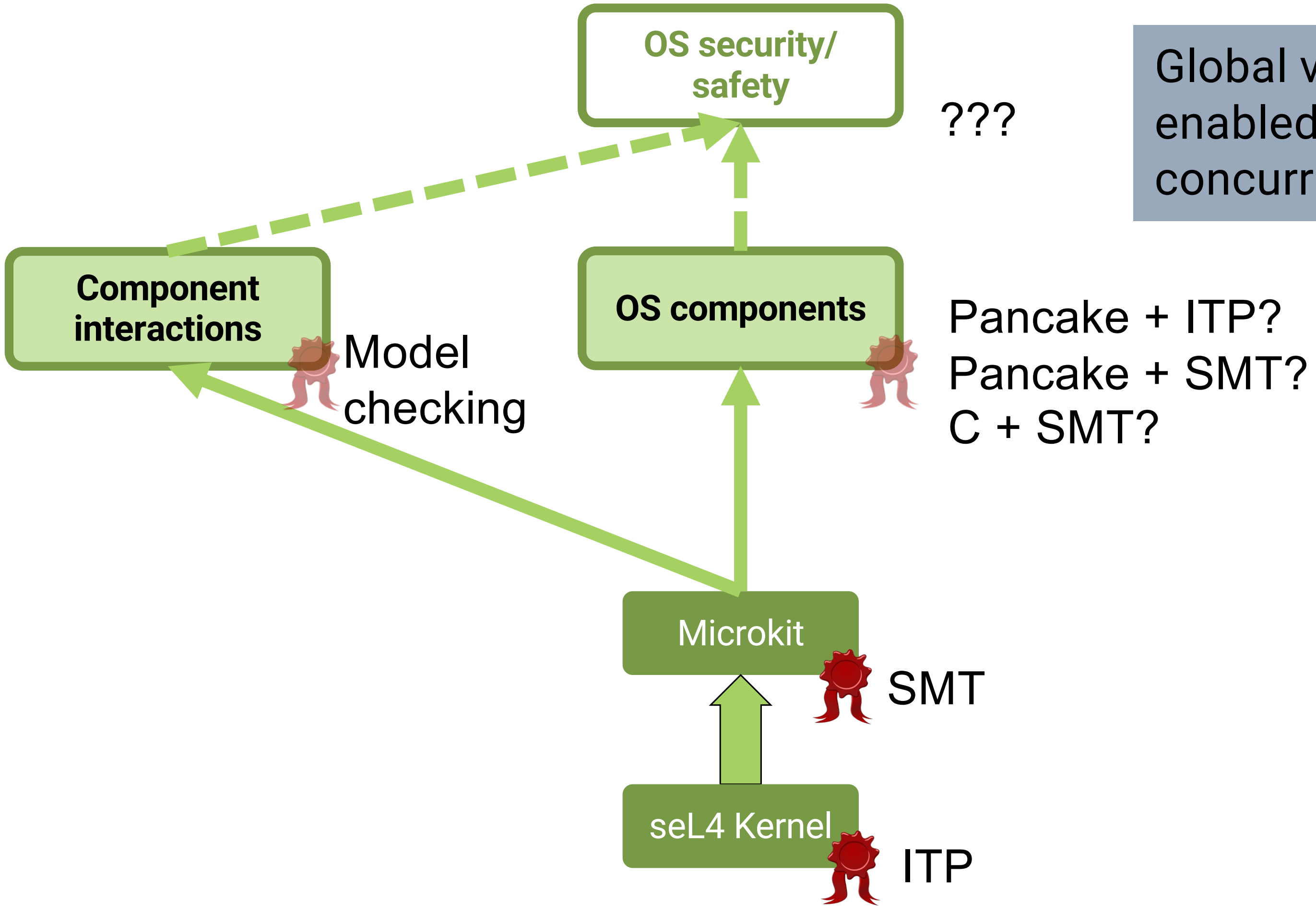
Lions OS Verification Status

- Network-layer protocols *automatically* proved deadlock-free
  - eliminated multiple performance bugs
  - verification supports aggressive optimisation!
- One component (copier) *automatically* verified with SMT solver
  - functional correctness (subject to correctness of neighbours)
  - confident can prove global properties
- Exploring refinement proof of MSpec



Challenge:
- Proving global security properties!

Aim: Verified LionsOS

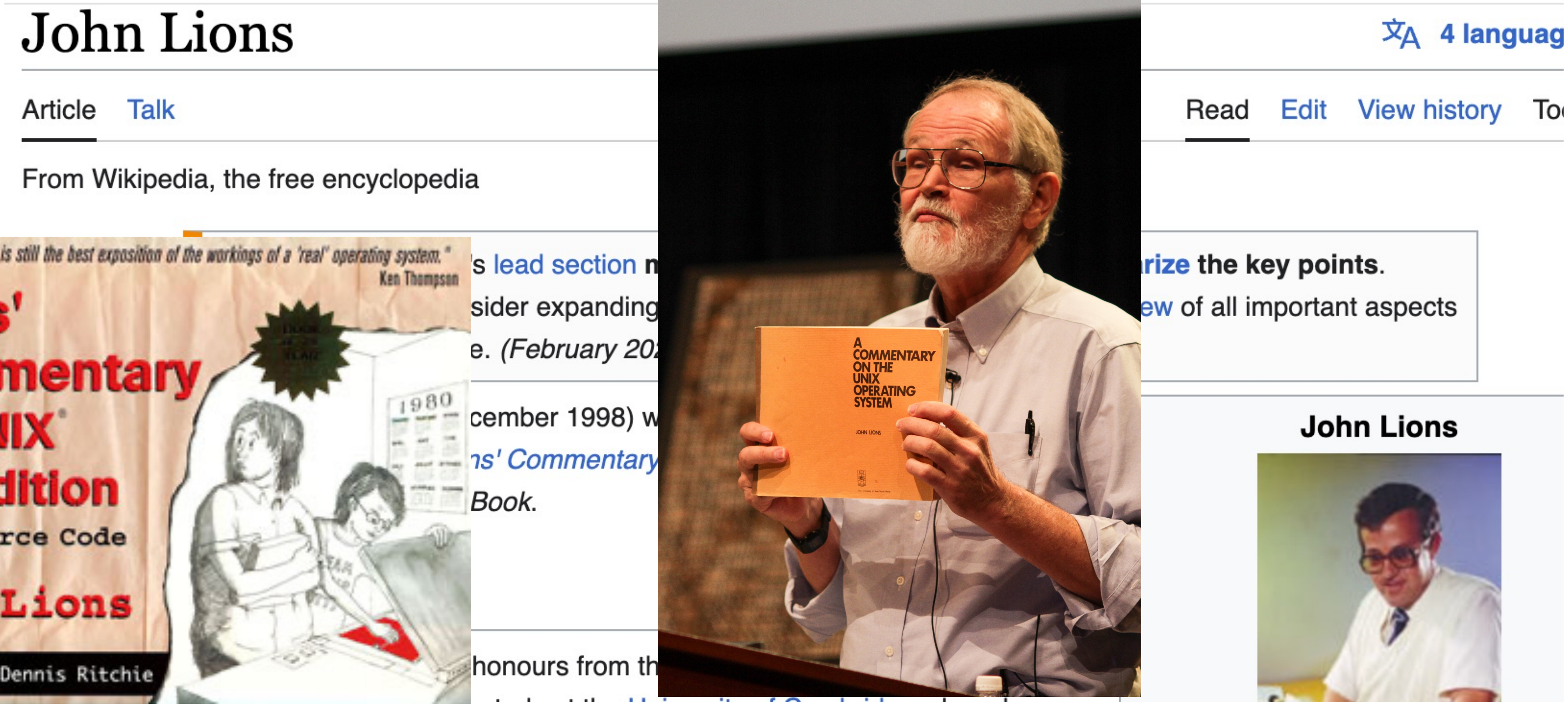


Global verification enabled by “tamed” concurrency

Summary: LionsOS

What's In a Name?

- https://en.wikipedia.org/wiki/John_Lions



John Lions

4 languag

Article Talk

Read Edit View history To

From Wikipedia, the free encyclopedia

's lead section m

sider expanding

e. (February 20

rize the key points.

ew of all important aspects

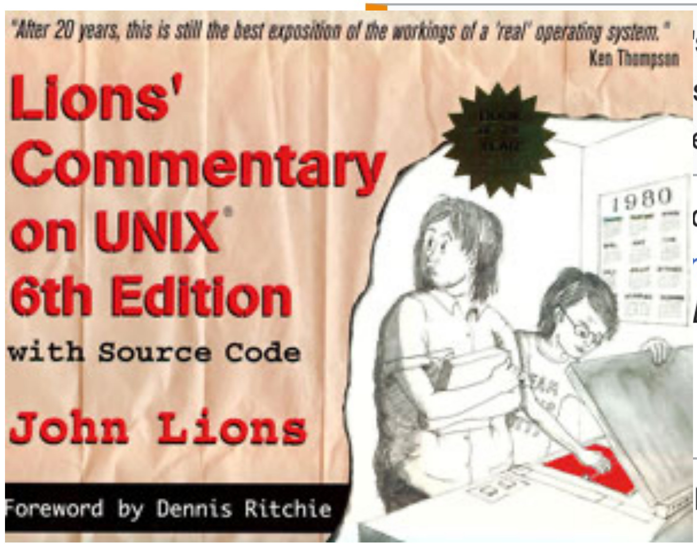
cember 1998) w

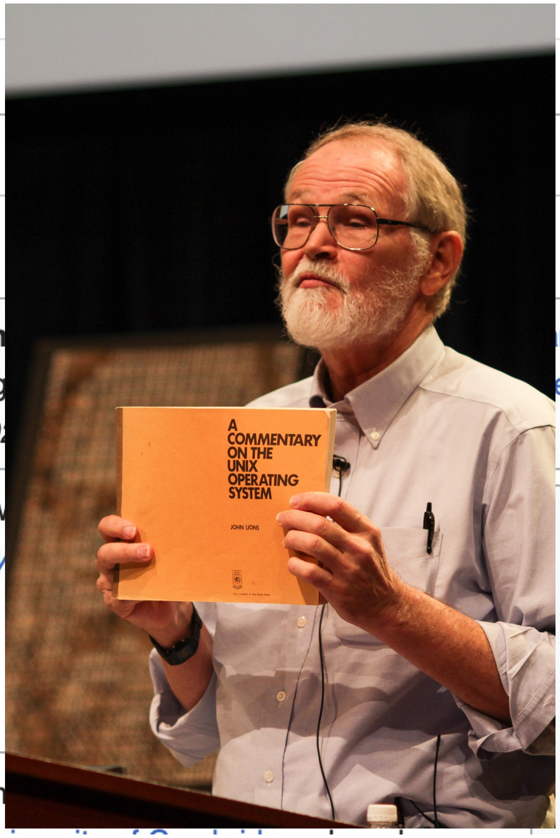
ns' Commentary

Book.

honours from th

John Lions


"After 20 years, this is still the best exposition of the workings of a 'real' operating system." Ken Thompson
Lions' Commentary on UNIX 6th Edition with Source Code
John Lions
Foreword by Dennis Ritchie
1980



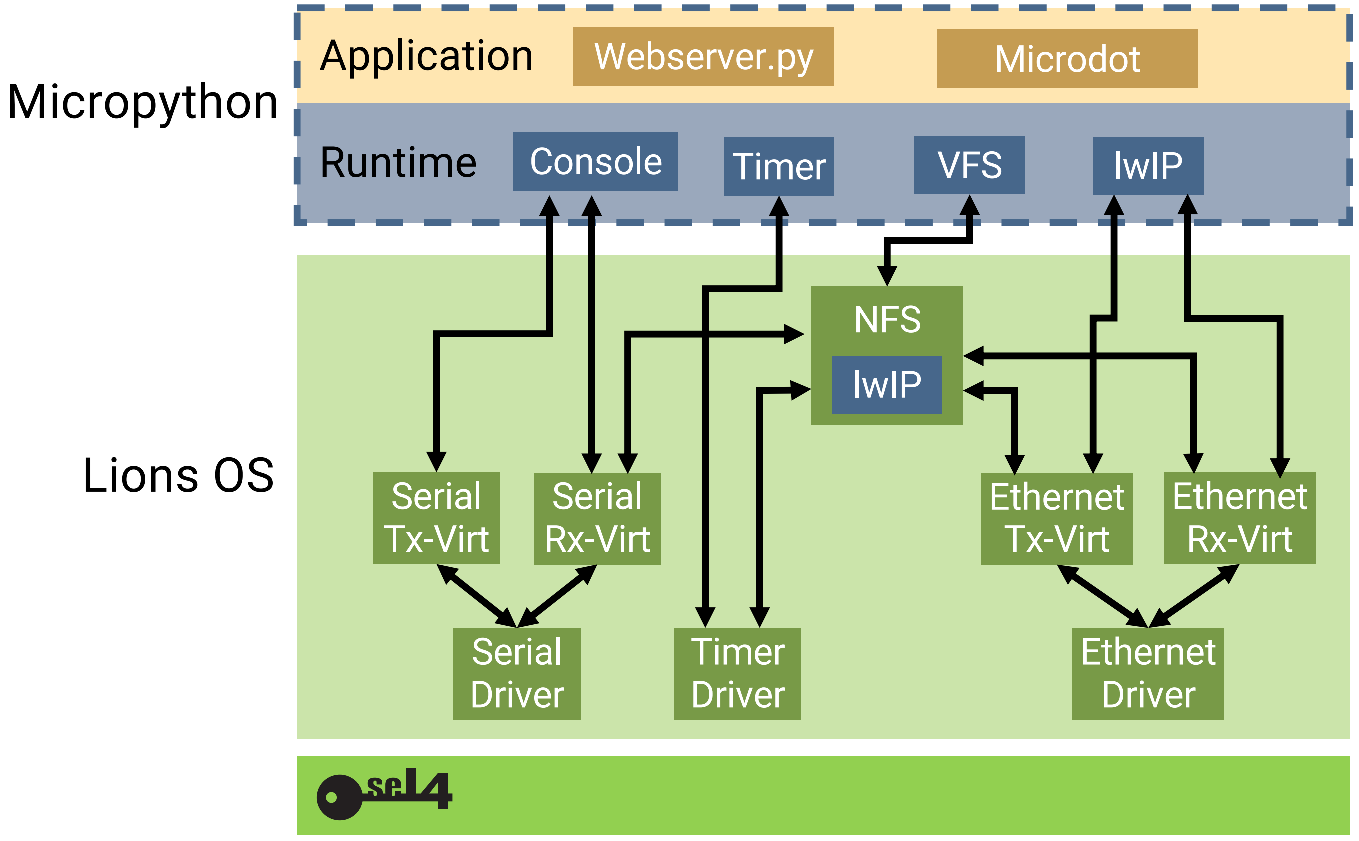
LionsOS Release 0.2 (July'24)

- Native serial, Ethernet, I2C drivers
- Native NFS client, Python interpreter (MicroPython)
- Native components in Rust supported
- Native web server (in Python)
- Driver VMs: graphics, touch screen, audio
- Next release (0.3 – Oct'24): Native file system

Overview:	https://trustworthy.systems/projects/LionsOS/
Docs:	https://lionsos.org/
Source:	https://github.com/au-ts/lionsos/
License:	2-clause BSD

Open Source!

Web Server Based on LionsOS



https://beta.sel4.systems/

LionsOS Support

- NIO America
- DARPA PROVERS program collaborating with Collins Aerospace
- Cyberagentur (Germany) EvIT program collaborating with PlanV, U Gothenburg

Foundations (Microkit, device driver framework) supported by:

in association with National Cyber Security Centre

Technology Innovation Institute

Security is no excuse for bad performance!

https://trustworthy.systems