NO SAFETY WITHOUT SECURITY

Cybersecurity for Autonomous Vehicles

Gernot Heiser FTSA FIEEE FACM

Scientia Professor & John Lions Chair, UNSW Sydney

Chief Research Scientist, Data61, CSIRO

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

Cybersecurity: 1st Class Safety Issue

ATSE

Fundamental rules of cyber space:

1. **The internet is a hostile environment**
2. **Anything that is internet-connected *can* be attacked**
3. **Anything that *can* be attacked *will* be attacked**

Examples:

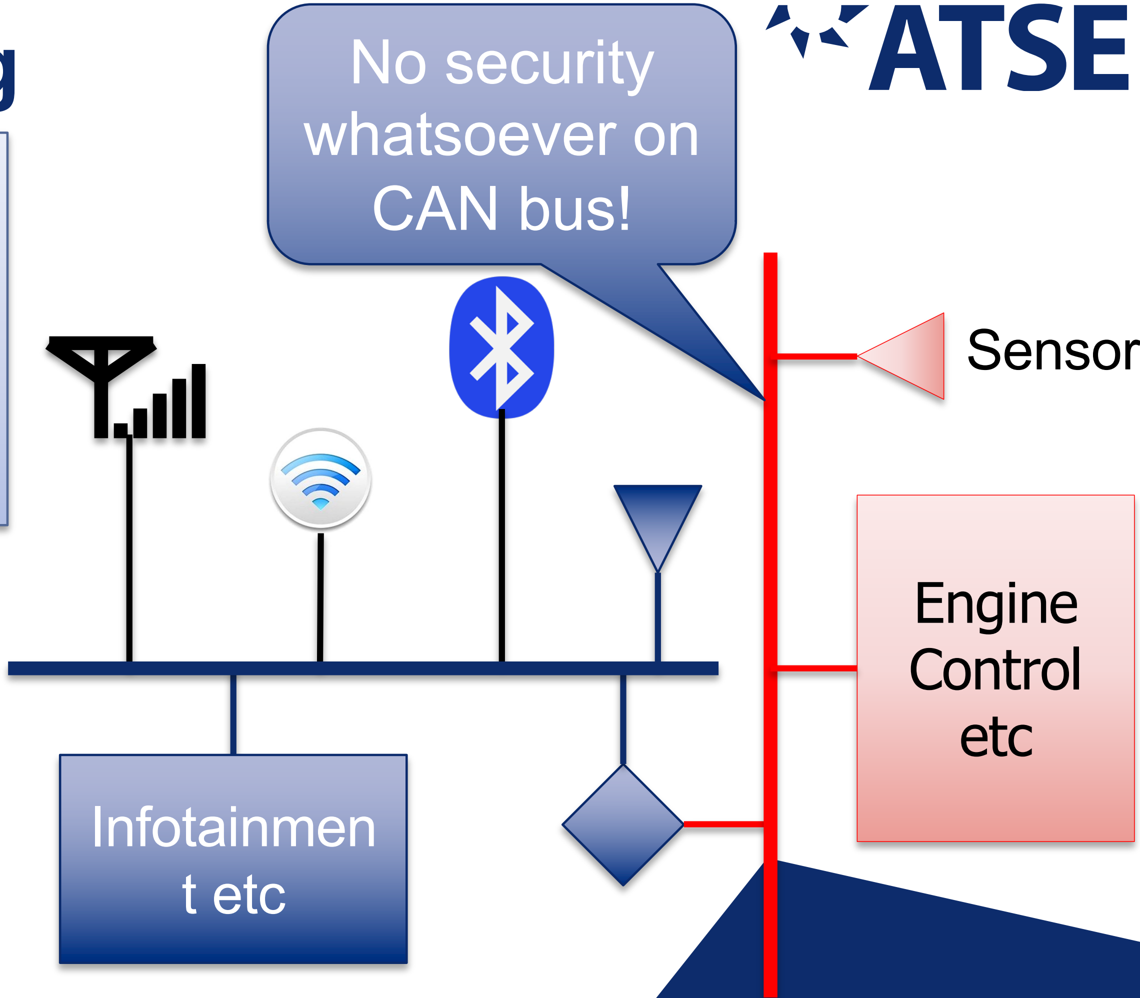
- **Cars**
- **Trains**
- **Aircraft**

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

Example Car Hacking

Networking for:

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



ATSE

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

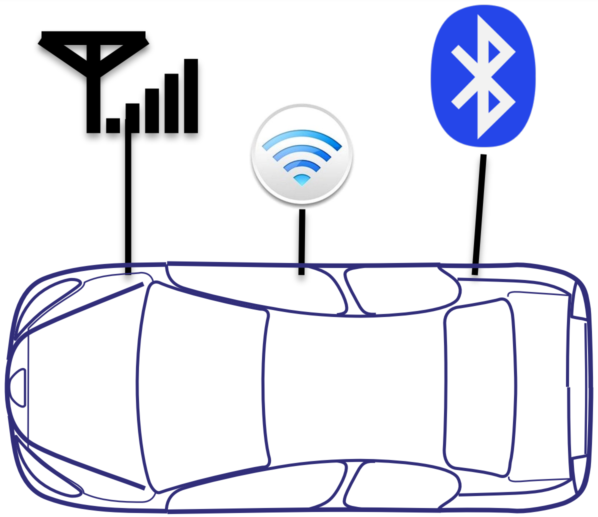
Challenge of Networking

ATSE

Networking creates remote attack opportunities

- from passengers (wifi, Bluetooth)
- from nearby cars (wifi, Bluetooth) – incl infected ones!
- from anywhere (cellular)

BlueBorne



Attack vectors:

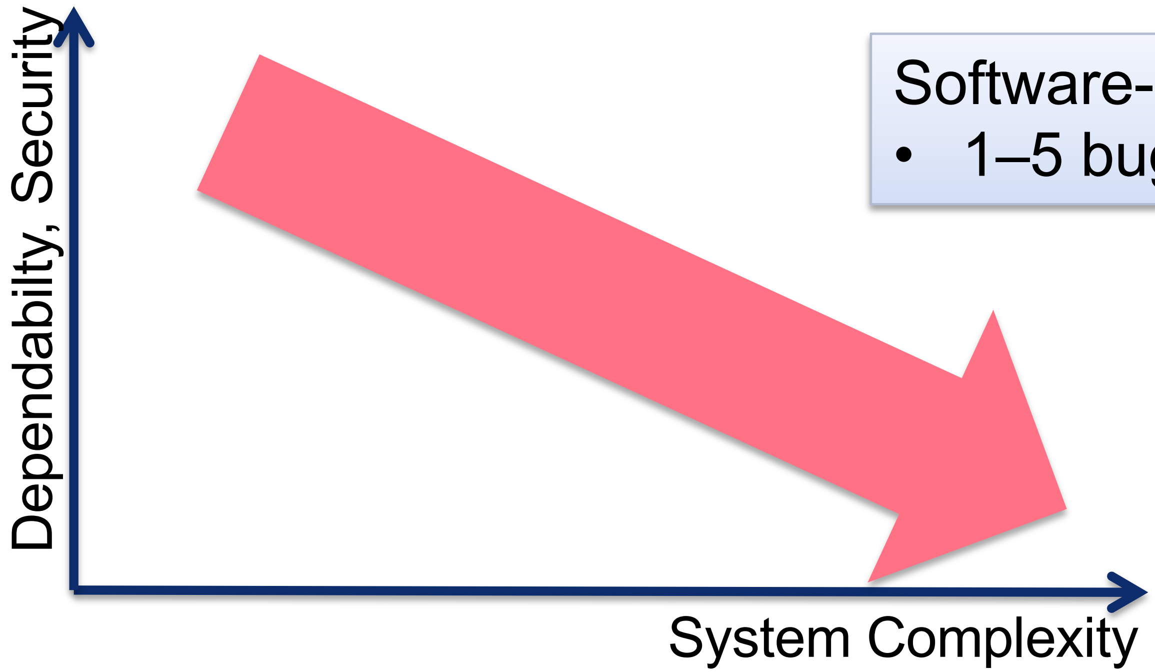
- Insecure protocols
- Reusing crypto keys
- Software vulnerabilities

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

WHY ARE SYSTEMS SO VULNERABLE?

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

Failure Reason #1: Complexity



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

Bluetooth protocol stack:
Multiple 100,000 lines

Linux/Windows kernel:
Tens of millions lines

Complexity Drivers
- Features/functionality
- Legacy reuse

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Operating-System "Security"

ars TECHNICA — BIZ & IT · TECH · SCIENCE · POLICY · CARS · GAMIN

RISK ASSESSMENT —

Windows is no better!

Unsafe at any clock speed: Linux kernel secu[...] rethink[...]

Software will break

The enemy will be on the platform!

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

Autonomy Increases Complexity

ATSE

Requires new functionality

- Core autonomy functionality
- Computer vision
- Sensor fusion, from vastly increased number of sensors
- Collision avoidance

Increased integration of automotive control with external world

Much increased attack surface!

Failure Reason #2: Care Factor

ATSE

Developer priorities

1. Features/functionality
2. Cost
3. Time to market
4. …
5. ...
6. ...
7. ...

....

....

999. Security

Developer expertise

1. Undergraduate programming
2. Application domain
3. Maybe hardware
4. …
5. ...
6. ...
7. ...

....

....

999. Security

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

Failure Reason #3: Security ≠ Safety

ATSE

Classic safety thinking (eg automotive, avionics, electrical):

- Failures are *random*
- Failure rates can be kept *very low* through systematic process
- Multiple failures are *independent*

No safety without security!

Reality of software security weaknesses:

- Failure is *deterministic*
- Failure rates are *high*
- Attackers *systematically combine* multiple vulnerabilities
- **Classic safety approaches do not work against cyber attacks!**

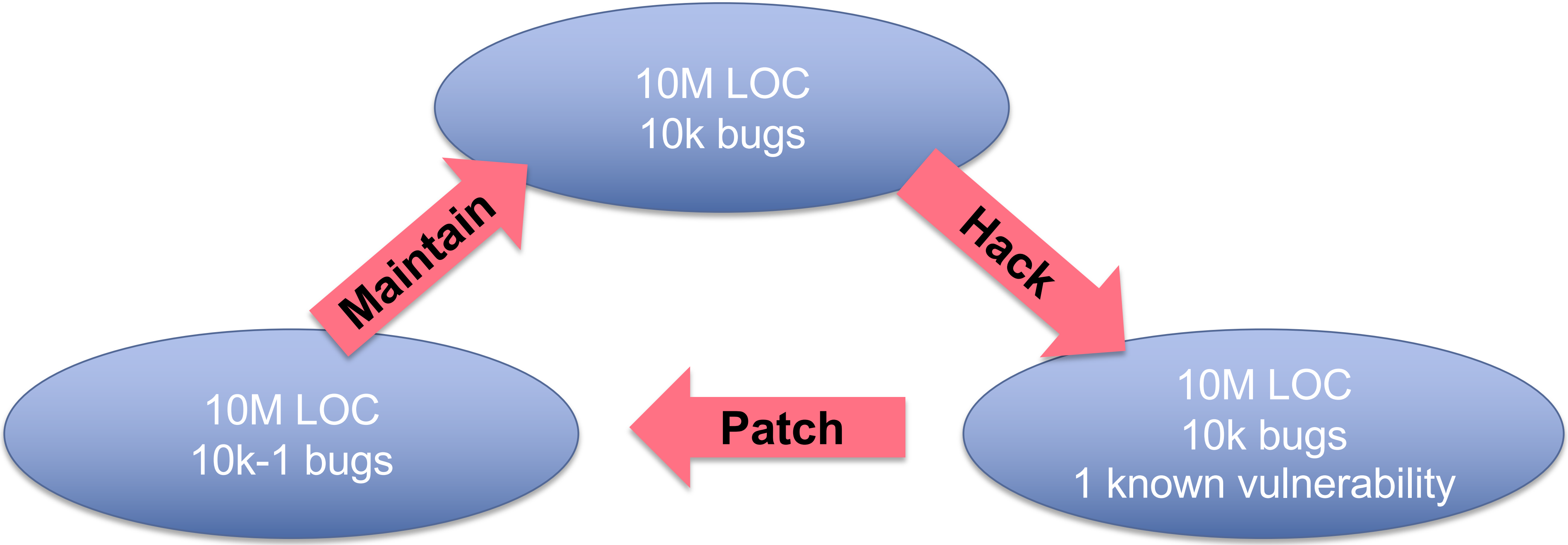
AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

ATSE

STANDARD DEFENCES

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

OK, So Let's Patch Regularly



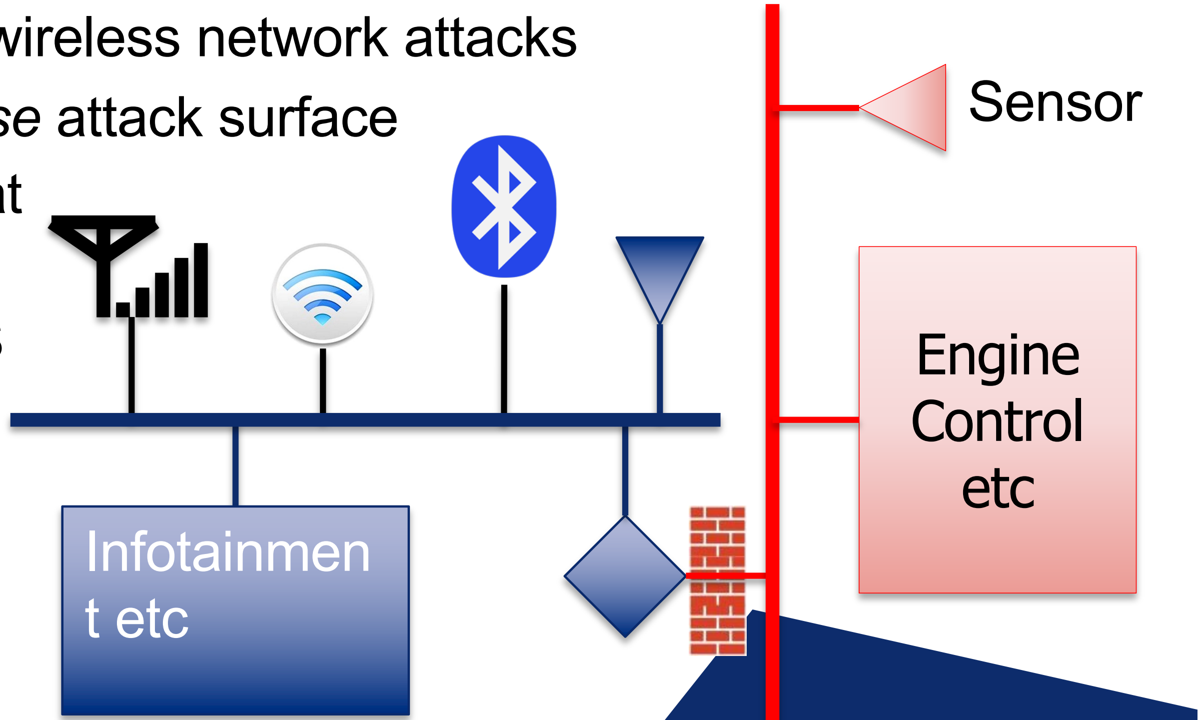
Patch-and-Pray: A losing proposition

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

So, Let’s Use Firewalls!

ATSE

- Imposes overhead (SWaP)
- Doesn’t protect against edge, wireless network attacks
- Even more code – may *increase* attack surface
- No help for valid messages that trigger bugs in software
- Firewall runs on vulnerable OS



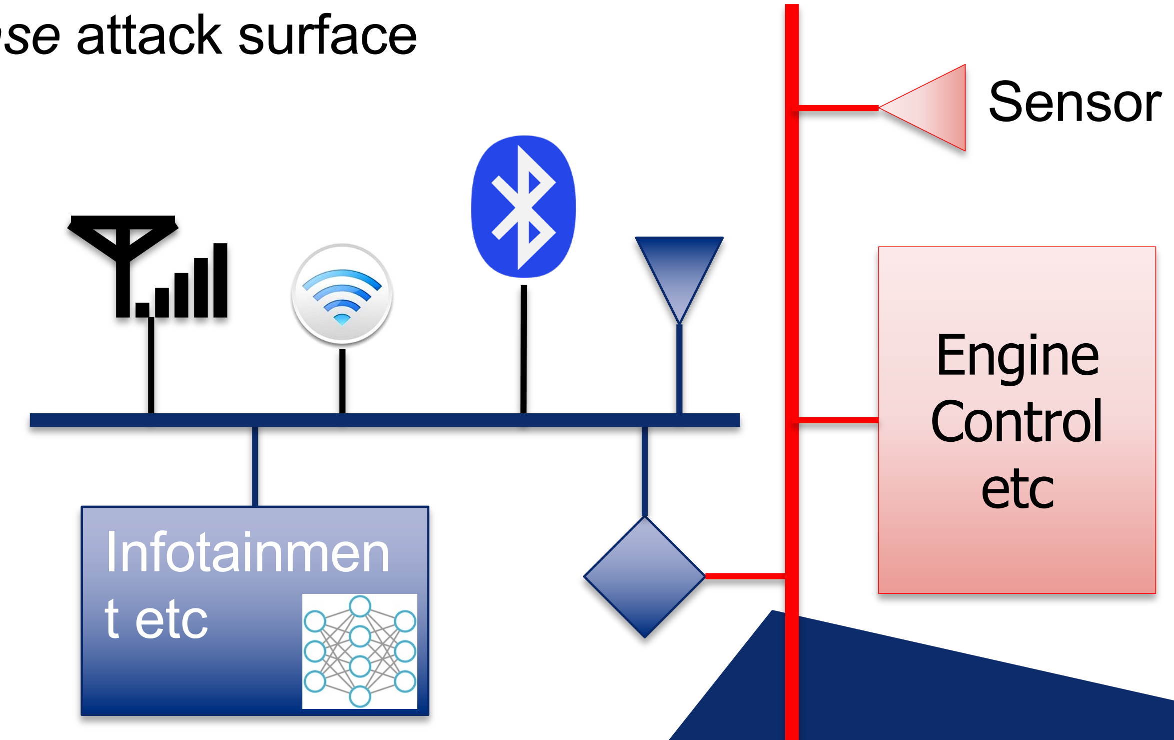
Firewalls treat symptoms, not causes of problems, are just another arms race!

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Let's Use AI to Detect Compromise!

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

Intrusion detection – admission of defeat

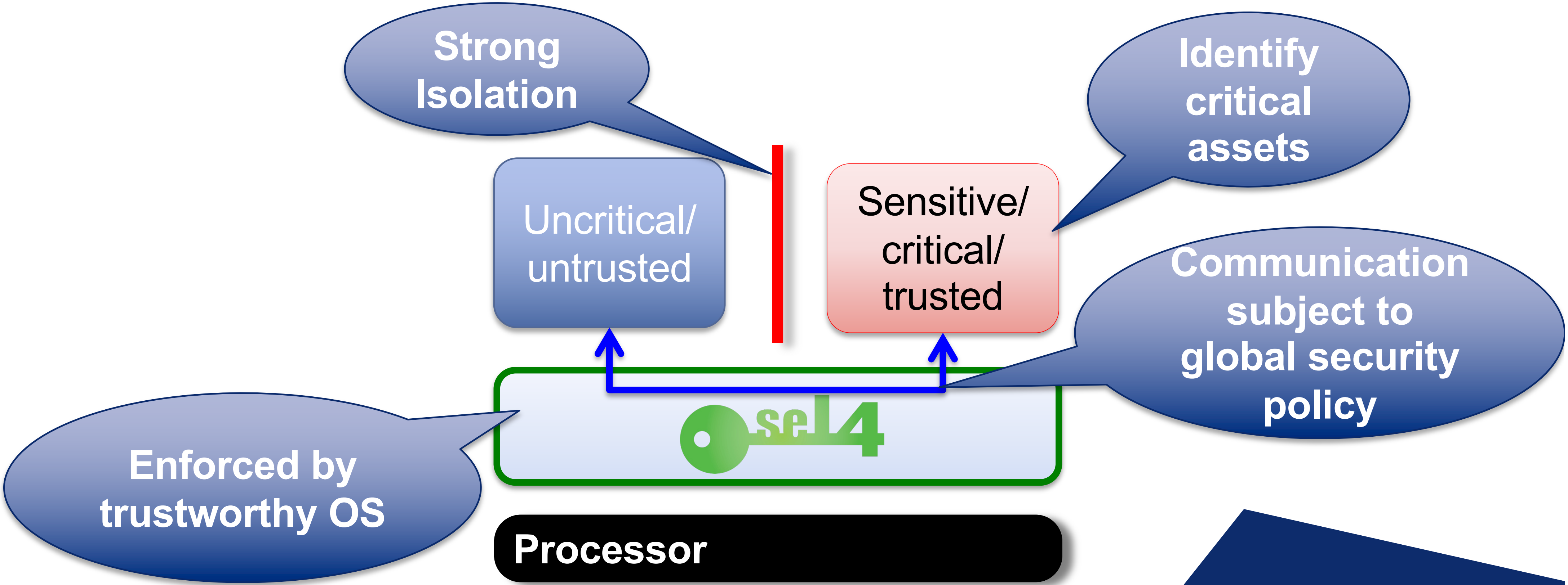


AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

WHAT IS NEEDED?

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Fundamental Requirement: Isolation



AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Trustworthiness: Can We Rely on Isolation?

ATSE

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

- it behaves **exactly** as it is specified,
- in a **timely** manner, and
- 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*!

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

seL4 Microkernel: We Have Proof!

Confidentiality | Integrity | Availability

Proof | Proof | Proof

Isolation

Abstract Model

Proof — Functional correctness

C Implementation

Proof — Translation correctness

Binary code

Provably impossible:

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

Exclusions (in progress):

- Initialisation
- Privileged state & caches
- Multicore
- Temporal isolation

How Does seL4 Compare?

ATSE

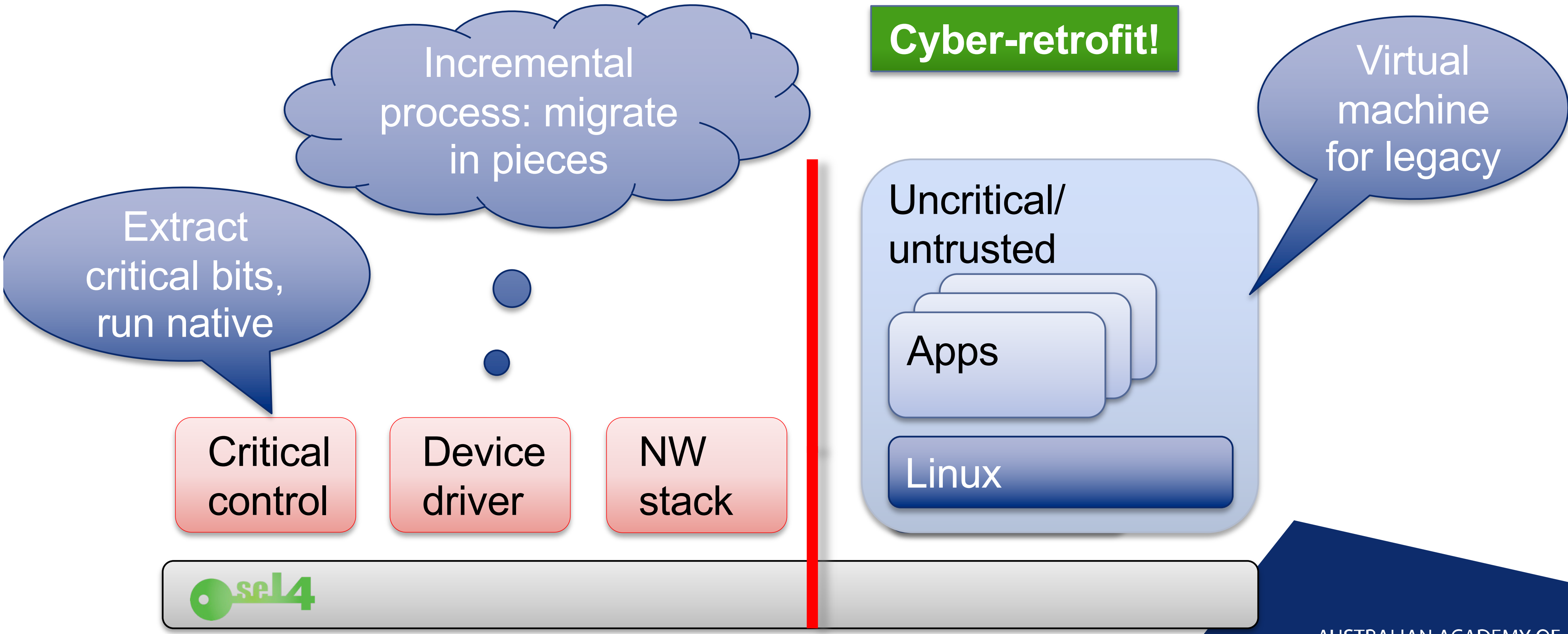
Feature	seL4	Others
Performance	Fastest	5–10 × slower
Impl. bugs	Provably none	No guarantee
Isolation	Proved	No guarantee
Latency bounds	Sound and complete	Estimates only
Storage channels	Provably none	No guarantee
Timing channels	Low-overhead prevention	No story or high overhead
Mixed criticality	Supported with high utilisation	None or resource-wastive

“World’s most verified kernel”

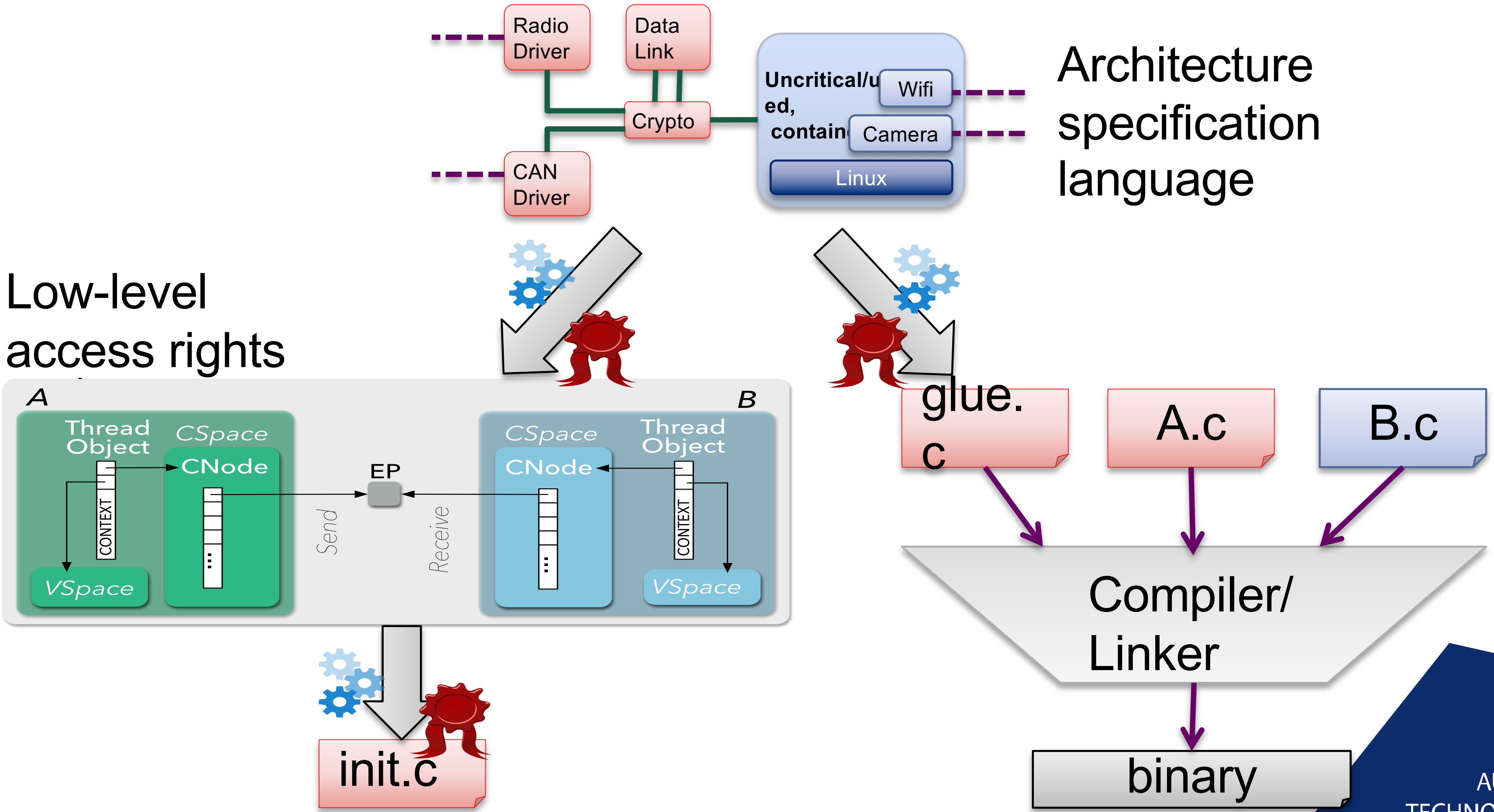
“Software you can depend on, data access you can trust”

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Security by Architecture



Enforcing the Architecture



Military-Grade Security

ATSE

Cross-Domain Desktop Compositor

Boeing Unmanned Helicopter

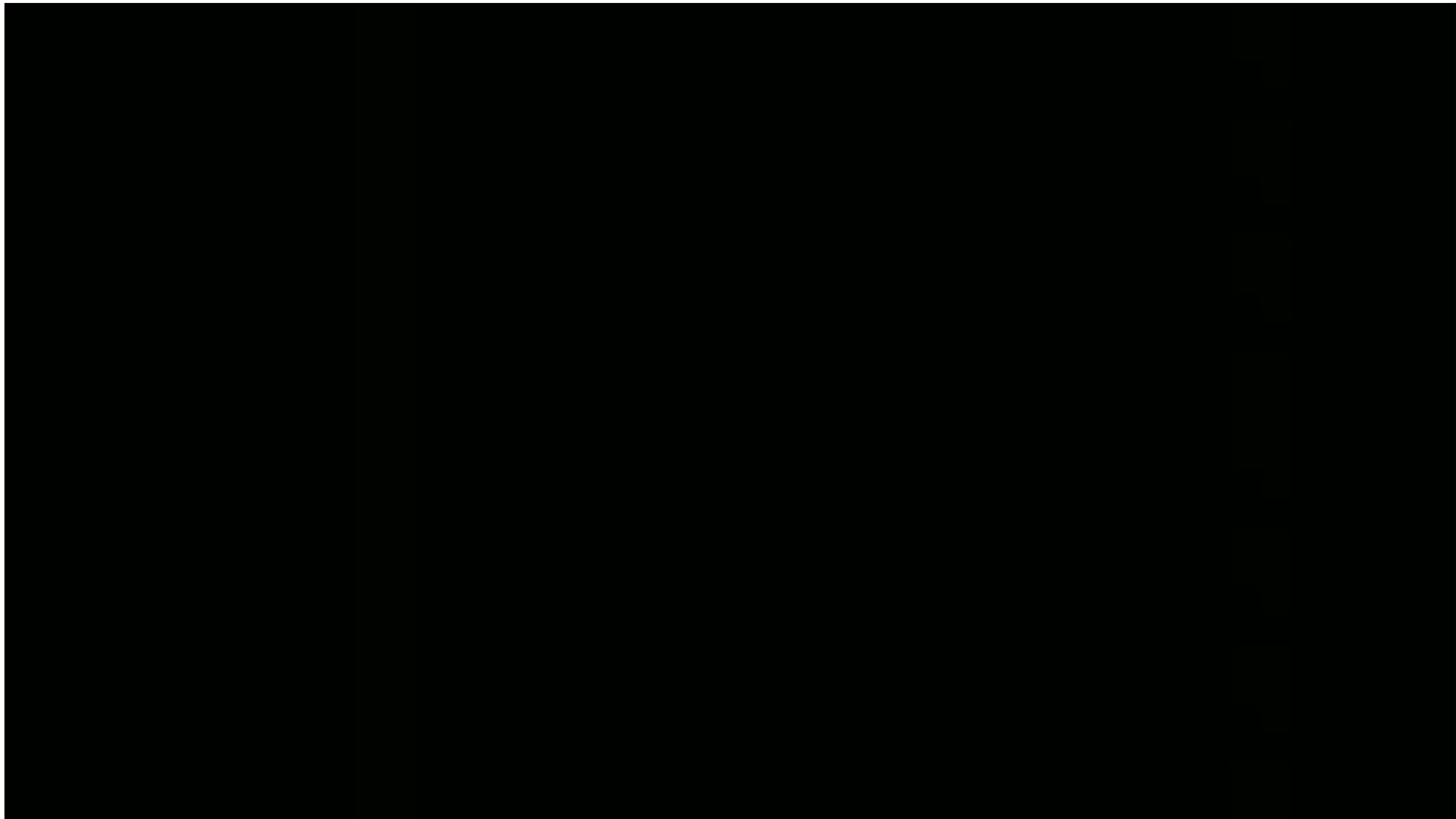
Crypto Stick

US Army Autonomous Trucks

AUSTRALIAN ACADEMY OF TECHNOLOGY AND ENGINEERING

Trustworthy Software At Work

ATSE



AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING

ATSE

https://trustworthy.systems

AUSTRALIAN ACADEMY OF
TECHNOLOGY AND ENGINEERING