seL4:
verified OS kernel
protects against cyber attacks

June Andronick & the Trustworthy Systems Group

January 2018

http://trustworthy.systems/


We offer three kinds of service:
GOOD - CHEAP - FAST
You can pick any two
GOOD service CHEAP won't be FAST
GOOD service FAST won't be CHEAP
FAST service CHEAP won't be GOOD

We offer:

VERIFIED *and* FAST

We offer:
VERIFIED *and* FAST

We're adding:
***and* CHEAP**

We offer:
VERIFIED *and* FAST

We're adding:
***and* CHEAP**

We need ***all 3*** to get
DEPLOYED

Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Approach:

- ➔ minimal & verified TCB
- ➔ ecosystem: seL4&co

Deployment

- ➔ projects
- ➔ community!

Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Approach:

- ➔ minimal & verified TCB
- ➔ ecosystem: seL4&co

Deployment

- ➔ projects
- ➔ community!






Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Approach:

- ➔ minimal & verified TCB
- ➔ ecosystem: seL4&co

Deployment

- ➔ projects
- ➔ community!

cheaper ➔ proofs for free

relevant ➔ more features

scale ➔ proof engineering





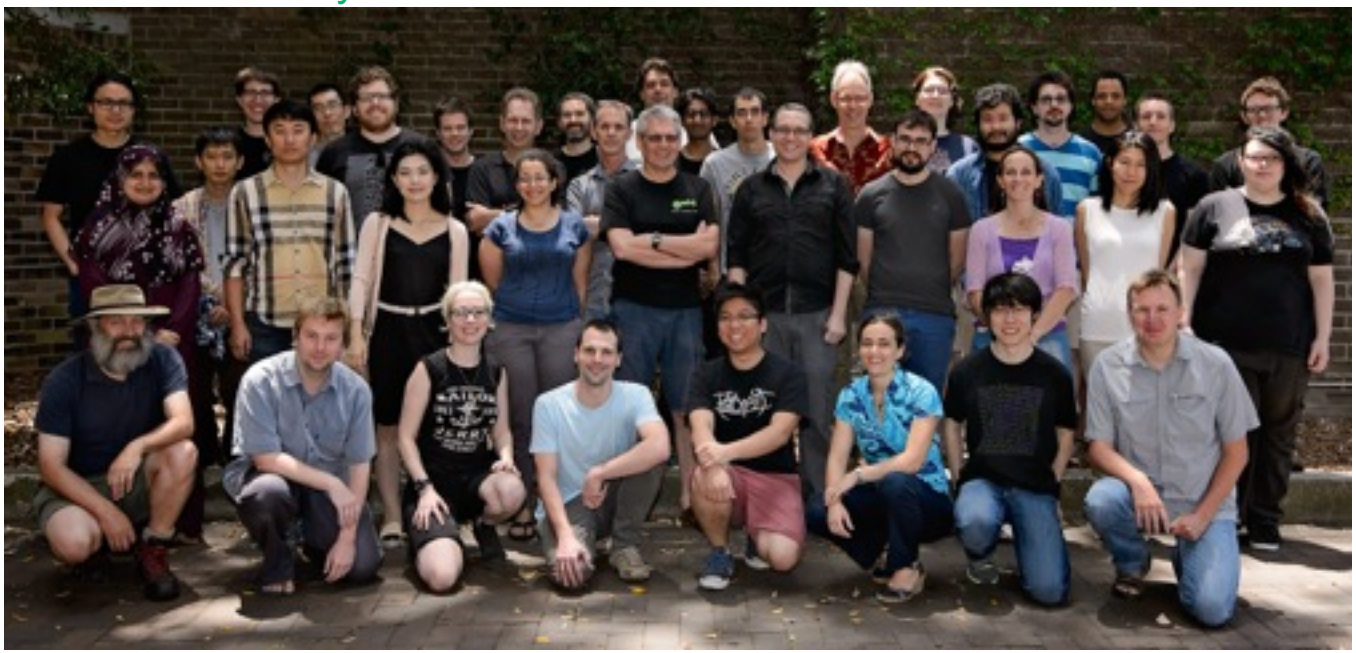

TS @ Data61

https://trustworthy.systems/

The Trustworthy Systems group is a set of people with a mission

experts in formal methods, operating systems, programming languages, security

provide the world with deployable, truly trustworthy software systems



TS @ Data61

https://trustworthy.systems/

The Trustworthy Systems group is a set of people with a mission

experts in formal methods, operating systems, programming languages, security

provide the world with deployable, truly trustworthy software systems

Key differentiator:

- combination of expertises
- combination of research and engineering
- critical mass

Key differentiator

- strength of mathematical proof, to highest standards
- high performance for real-world impact and deployment

Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Approach:

- ➔ minimal & verified TCB
- ➔ ecosystem: seL4&co

Deployment

- ➔ projects
- ➔ community!

cheaper

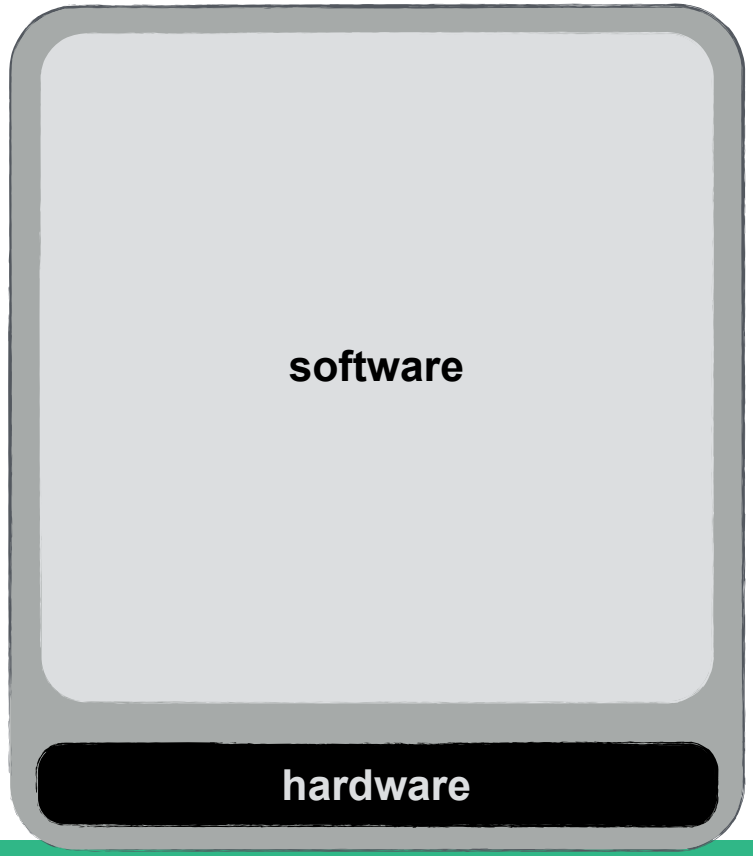
relevant

scale





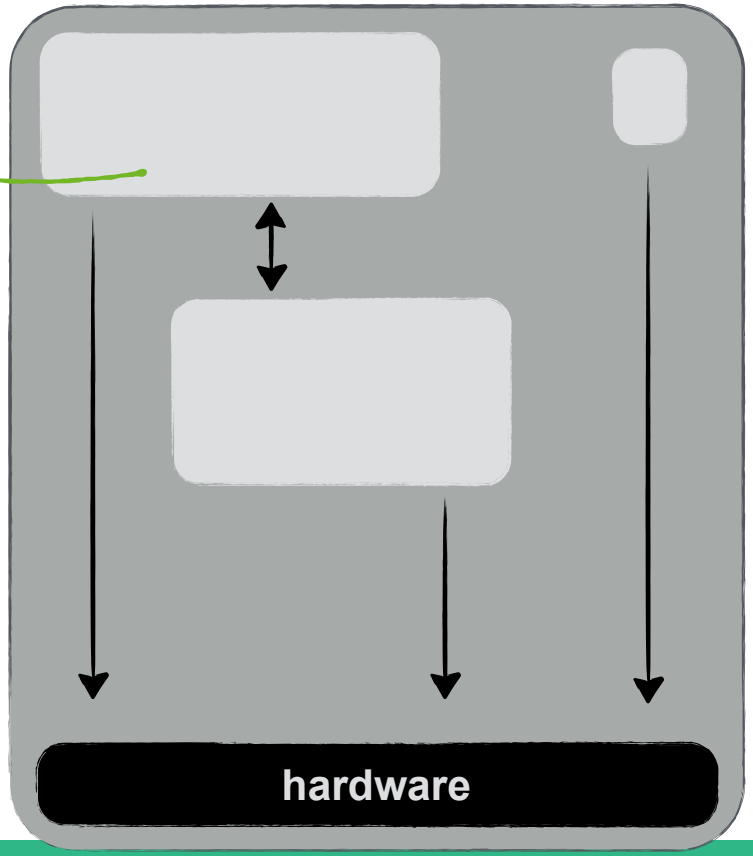

Our approach



Our approach

Componentized architecture

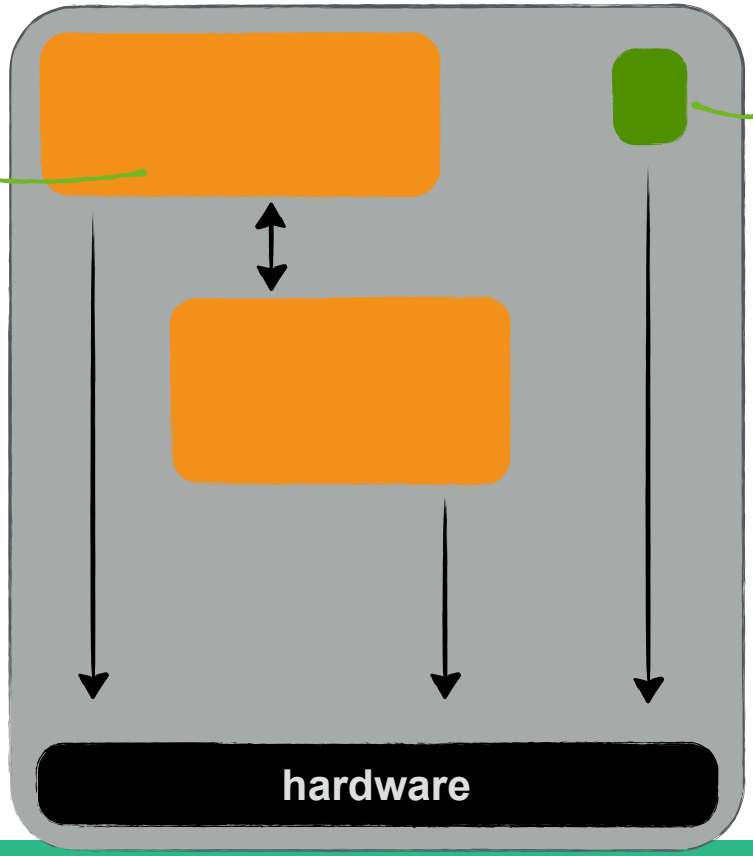
careful design



Our approach

Componentized architecture

careful design isolating trusted and untrusted part of the system



Minimal TCB (Trusted Computing Base)

limited number of trusted components

critical/trusted

uncritical/untrusted

Our approach

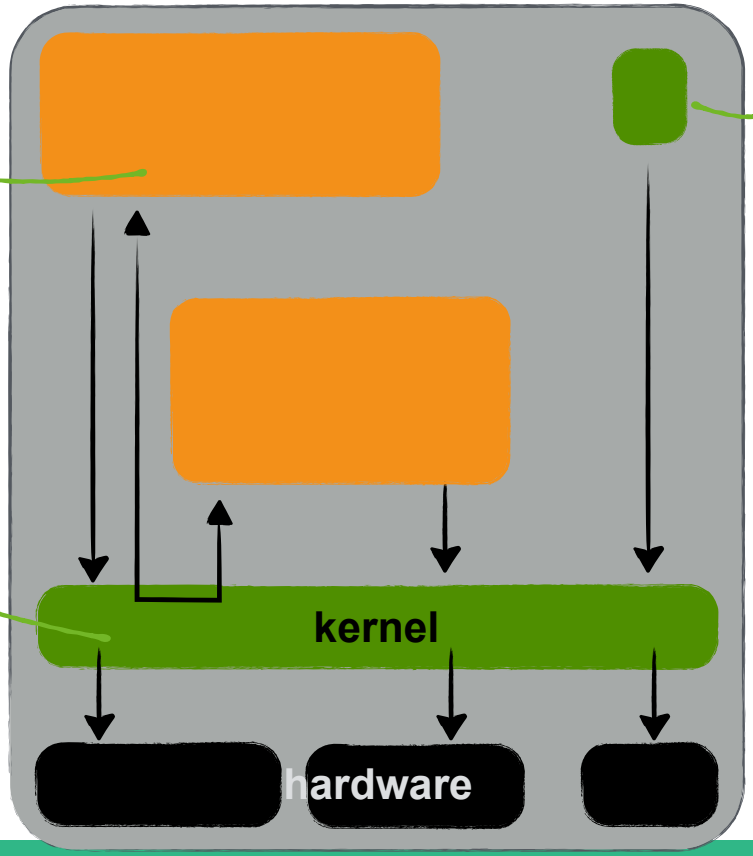
componentized architecture, with minimal TCB, on a trustworthy foundation

Componentized architecture

careful design isolating trusted and untrusted part of the system

Kernel-based system

enforcing access control and isolation



Minimal TCB (Trusted Computing Base)

limited number of trusted components

Our approach

componentized architecture, with minimal TCB, on a trustworthy foundation

Componentized architecture

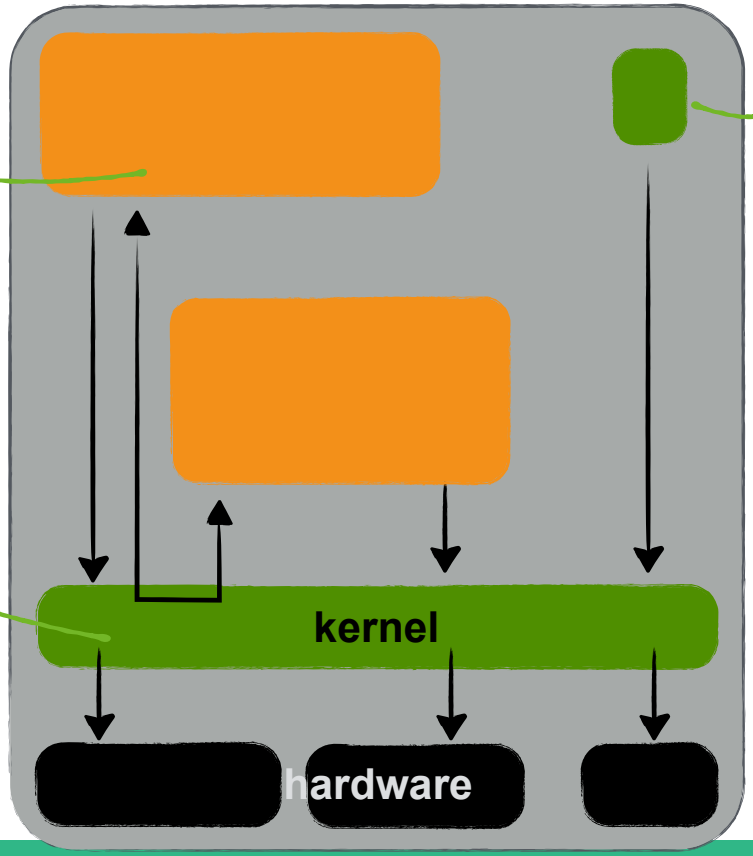
careful design isolating trusted and untrusted part of the system

VERIFIED

Kernel-based system

enforcing access control and isolation

VERIFIED



Minimal TCB (Trusted Computing Base)

limited number of trusted components

VERIFIED

Our approach

componentized architecture, with minimal TCB, on a trustworthy foundation

→ seL4 & family (CAmkES, etc)

Componentized architecture

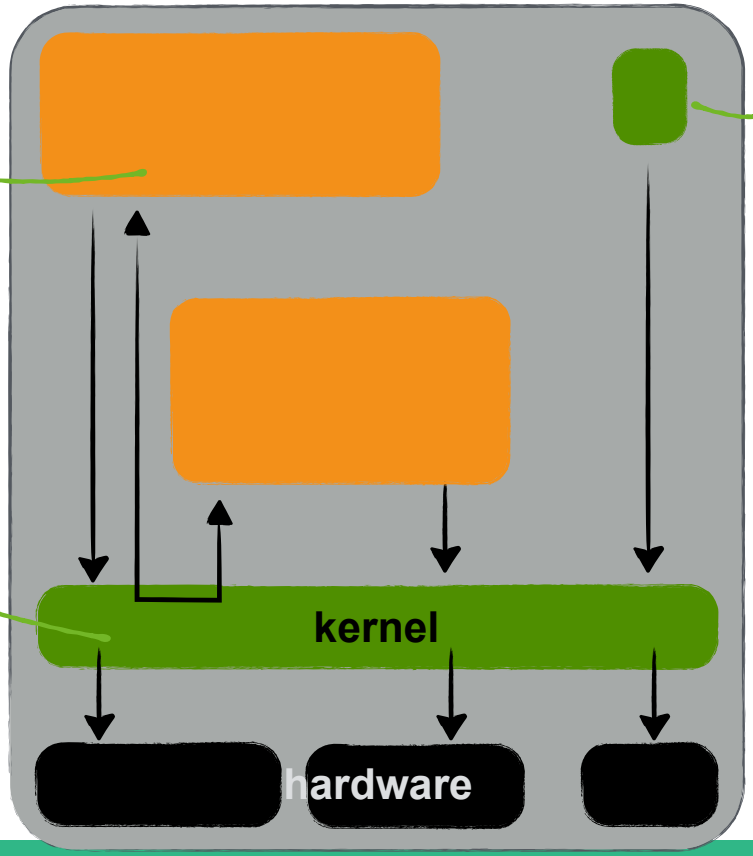
careful design isolating trusted and untrusted part of the system

VERIFIED

Kernel-based system

enforcing access control and isolation

VERIFIED



Minimal TCB (Trusted Computing Base)

limited number of trusted components

VERIFIED

seL4 in 1 slide

Small,

fast,

capability-based,

operating system kernel

seL4 in 1 slide

seL4
Security. Performance. Proof.

Small,

fast,

capability-based,

operating system kernel

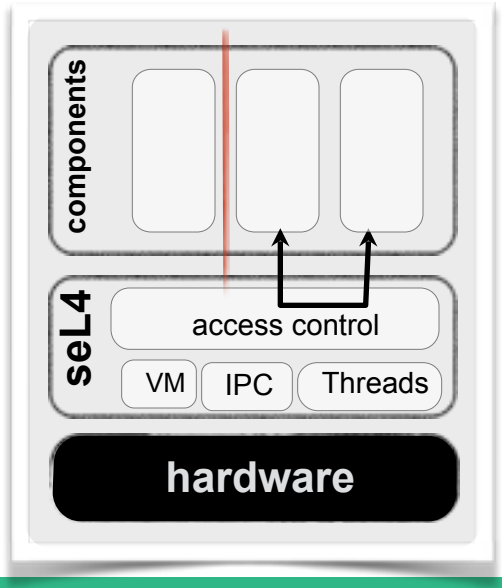
Code that runs in privileged mode of the hardware

⬇

Most critical part

Unprivileged mode

Privileged mode



seL4 in 1 slide

~10,000 lines of C and ASM code

⬇

Small attack surface,
More amenable to full verification

seL4
Security. Performance. Proof.

Small,

fast,

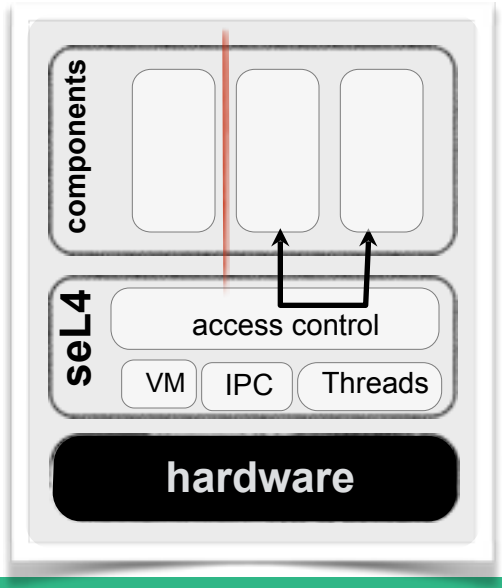
capability-based,

operating system kernel

Code that runs in privileged mode of the hardware

⬇

Most critical part



seL4 in 1 slide

seL4
Security. Performance. Proof.

~10,000 lines of C and ASM code
↓
Small attack surface,
More amenable to full verification

Small,

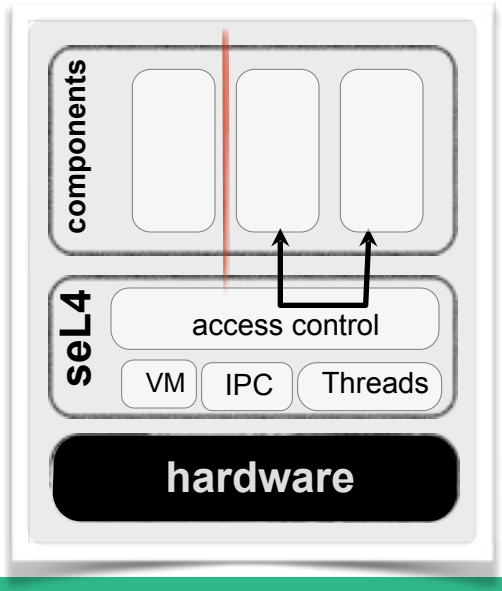
fast,

capability-based,

operating system kernel

Unprivileged code needs special permission (capability) to access resources and communicate
↓
Kernel can confine damage from attacks in unprivileged code

Code that runs in privileged mode of the hardware
↓
Most critical part



Unprivileged mode

Privileged mode

seL4 in 1 slide

seL4
Security. Performance. Proof.

~10,000 lines of C and ASM code
↓
Small attack surface,
More amenable to full verification

Small,

fast,

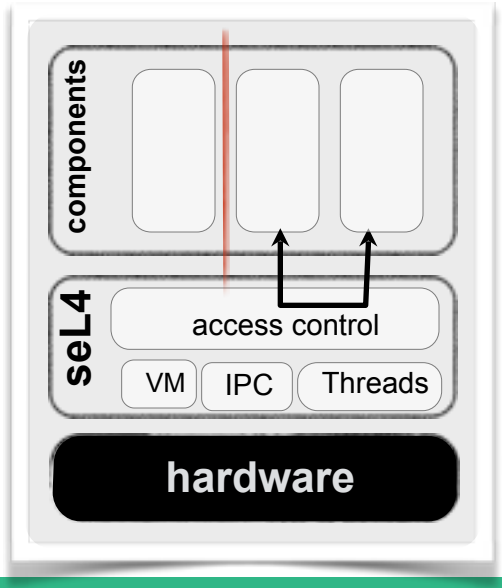
capability-based,

operating system kernel

World's fastest (5–10 faster) operating system designed for security/safety
↓
Suitable for real-world deployment

Unprivileged code needs special permission (capability) to access resources and communicate
↓
Kernel can confine damage from attacks in unprivileged code

Code that runs in privileged mode of the hardware
↓
Most critical part



seL4 in 1 slide

seL4
Security. Performance. Proof.

Small,
fast,
capability-based,
operating system kernel

~10,000 lines of C and ASM code
⬇
Small attack surface,
More amenable to full verification

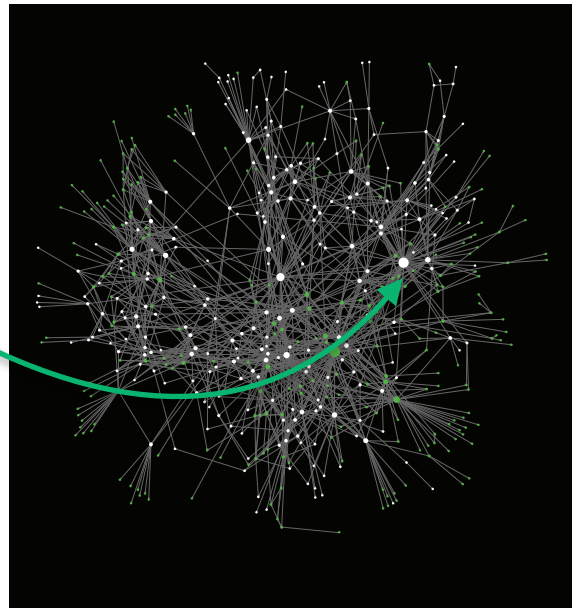
Unprivileged code needs special permission (capability) to access resources and communicate
⬇
Kernel can confine damage from attacks in unprivileged code

World's fastest (5–10 faster) operating system designed for security/safety
⬇
Suitable for real-world deployment

Code that runs in privileged mode of the hardware
⬇
Most critical part

```
void kernel_call ()
 {
    …
    …
    …
 }
```

(>500 functions)



What makes seL4 truly unique?

“World’s most verified kernel”

What makes seL4 truly unique?

"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**

What makes seL4 truly unique?

"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**

What it means:

Execution of kernel always defined:

- no null pointer de-reference
- no buffer overflows
- no code injection
- no memory leaks/out of kernel memory
- no div by zero, no undefined behavior
- no undefined execution
- no infinite loops/recursion

Even stronger:

- all the possible behaviours of the binary conform to spec
- security policies are enforced

Assumptions:

- Correct assembly code
- Correct hardware behaviours
- Correct hardware management (TLB and caches)
- Correct boot code
- DMA off or trusted
- Secure configuration

What makes seL4 truly unique?

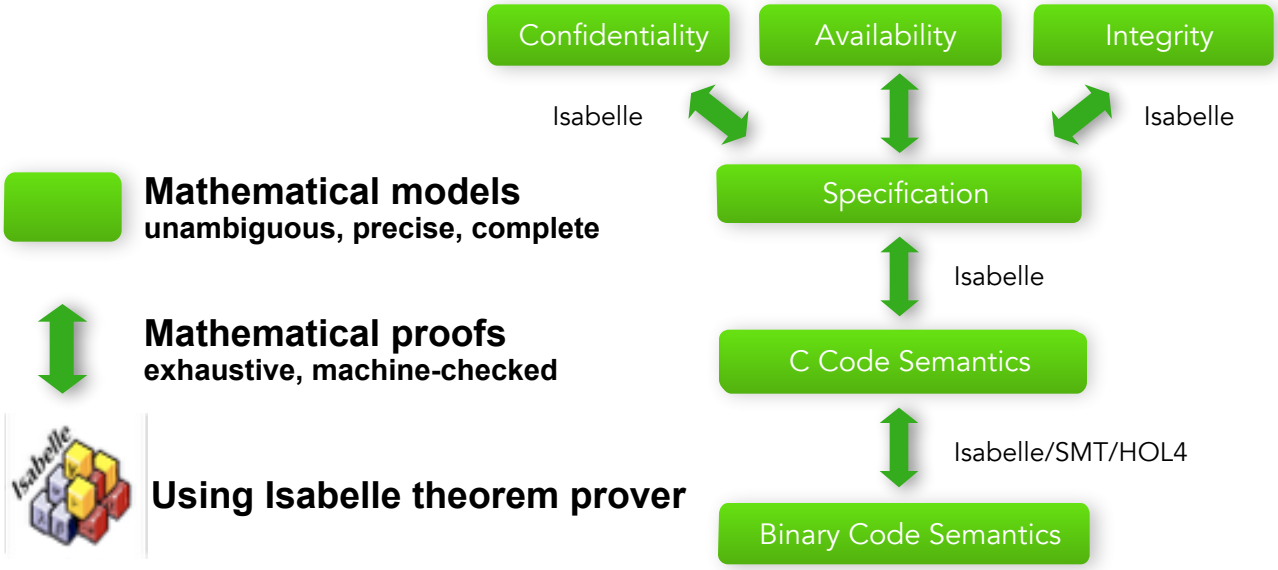
seL4
Security. Performance. Proof.

"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**



What makes seL4 truly unique?

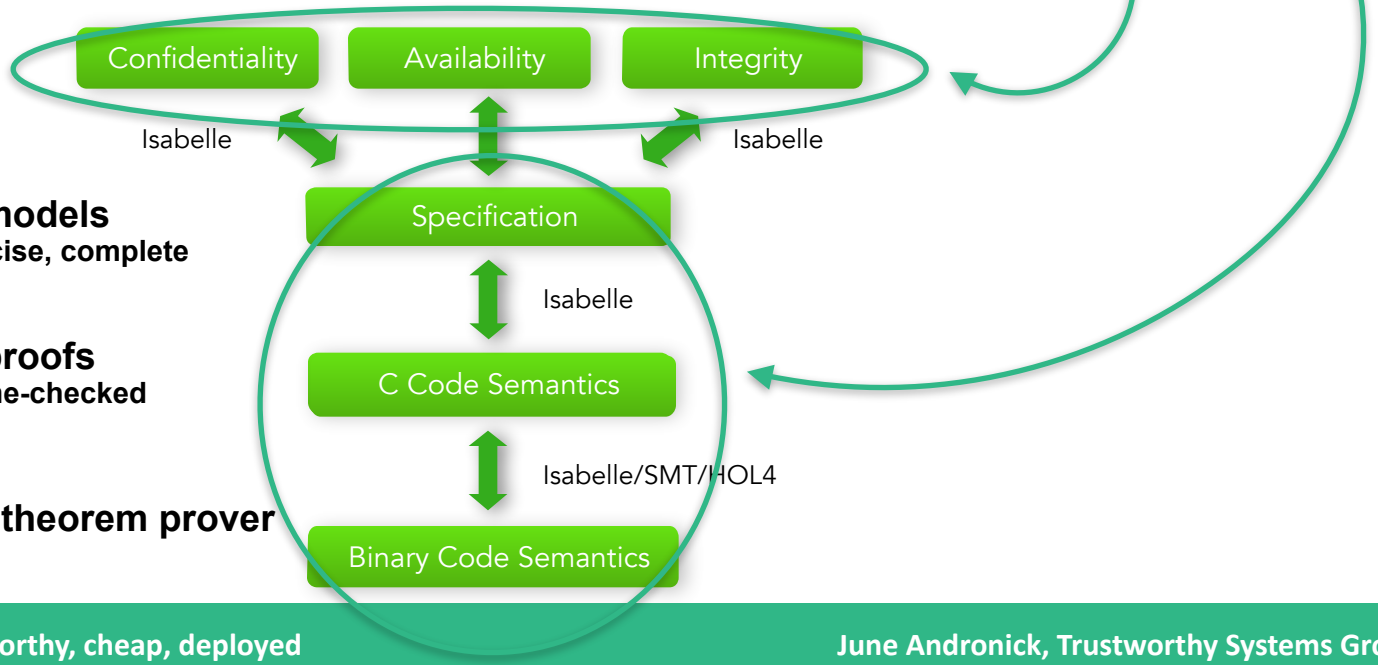
seL4
Security. Performance. Proof.

"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**



Mathematical models
unambiguous, precise, complete

Mathematical proofs
exhaustive, machine-checked

Using Isabelle theorem prover

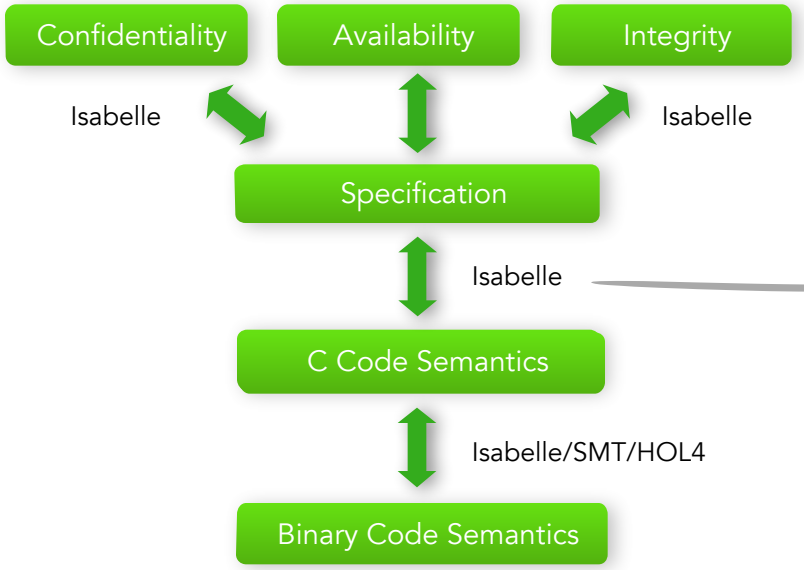
What makes seL4 truly unique?

"World's most verified kernel"

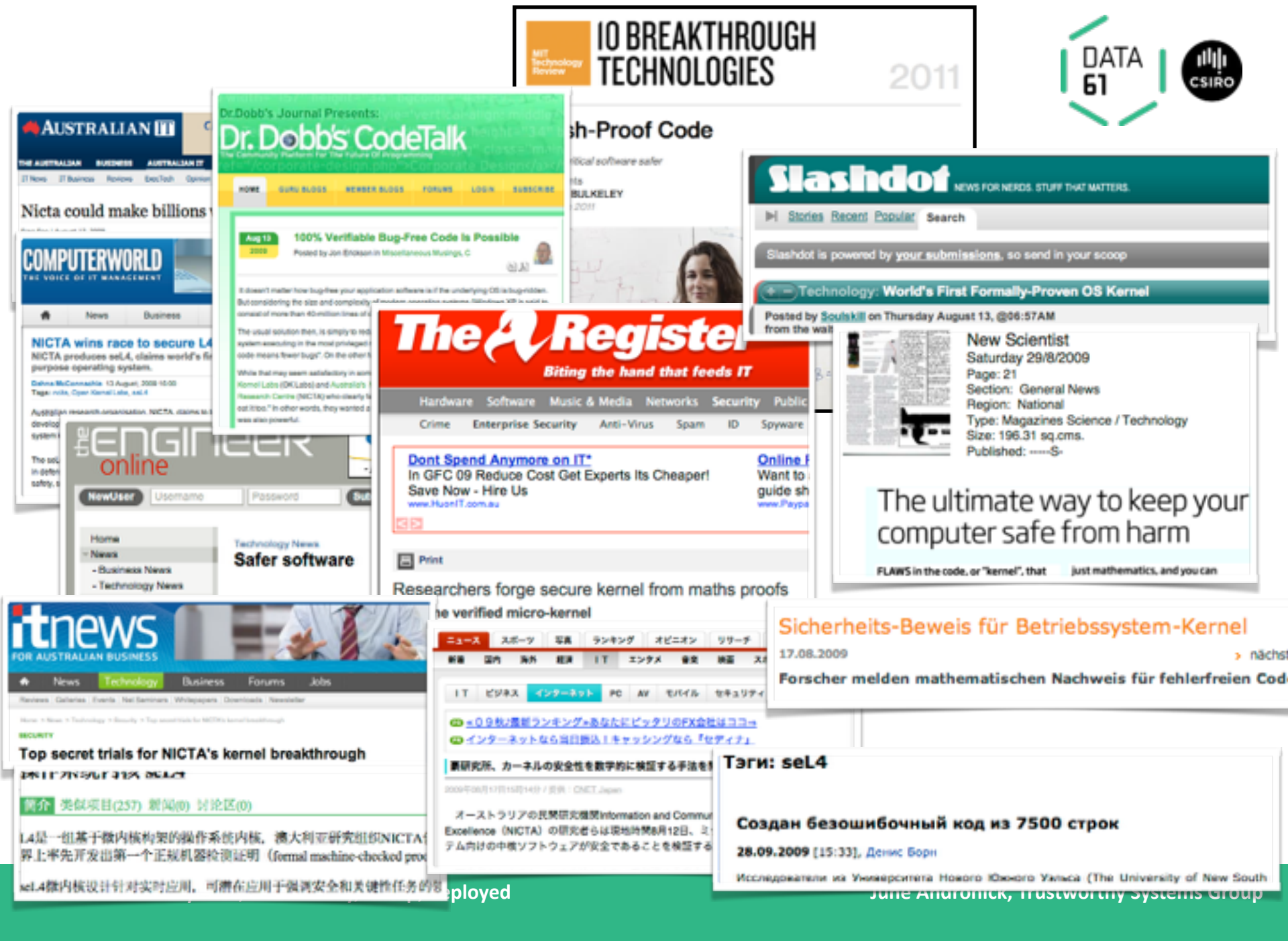
Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**



2009



eployed

June Andronick, Trustworthy Systems Group

What makes seL4 truly unique?

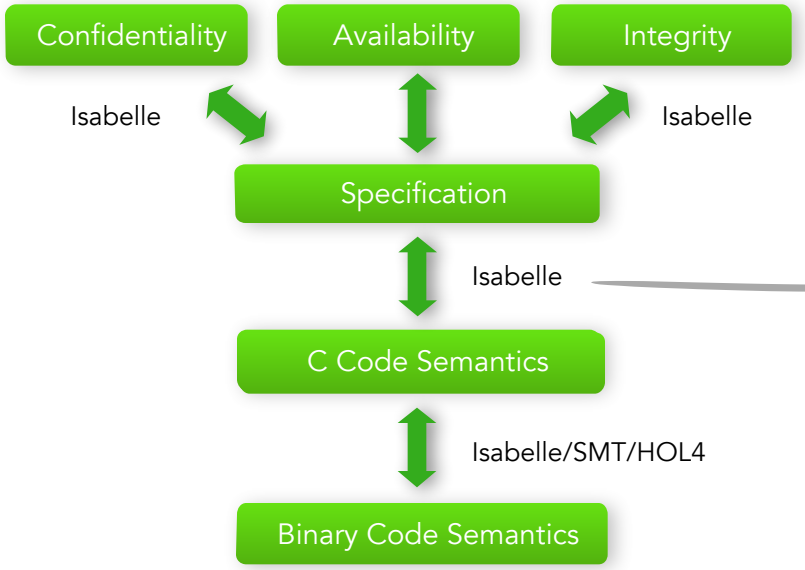
seL4
Security. Performance. Proof.

"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**



2009

What makes seL4 truly unique?

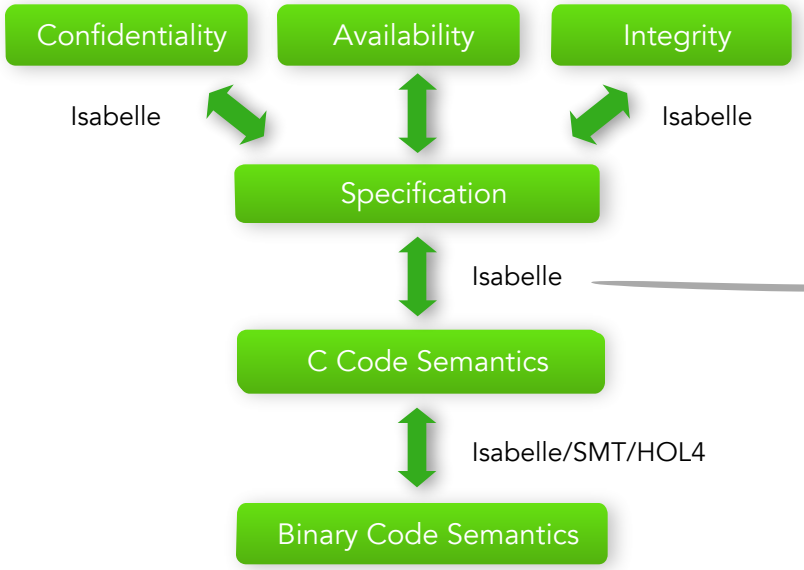
"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**

2012



2009

What makes seL4 truly unique?

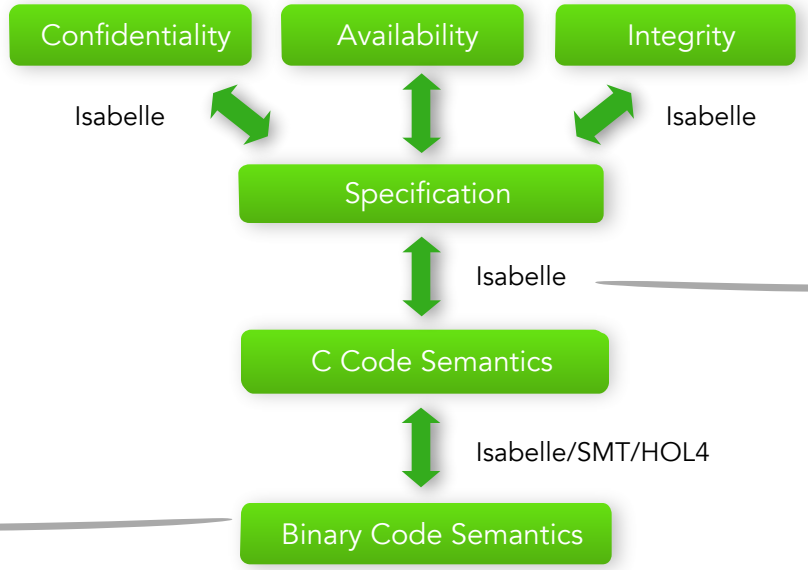
"World's most verified kernel"

Mathematical proof that code is **correct** w.r.t. specification,

Mathematical proof that it enforces strong **security** properties,

Proved safe upper bounds on their **worst-case execution times**

2012



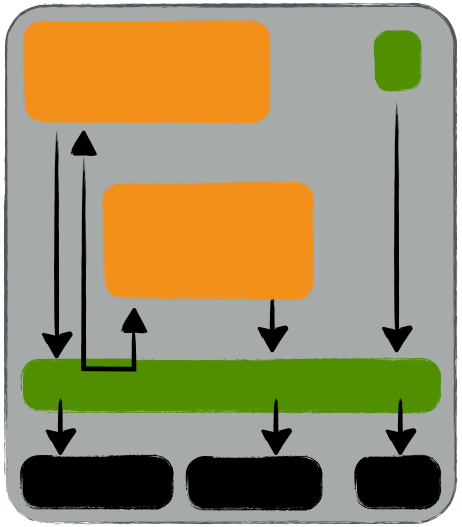
2009

2013

Building systems

seL4
Security. Performance. Proof.

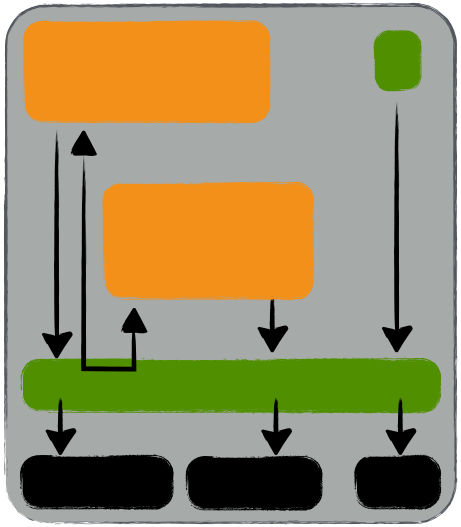
Key: proved **isolation**

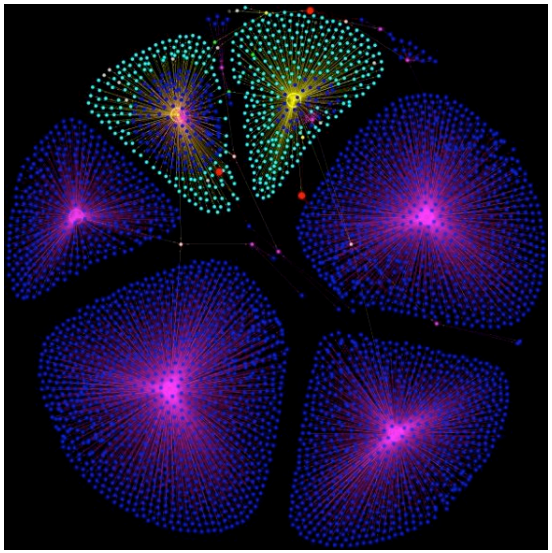


Building systems

seL4
Security. Performance. Proof.

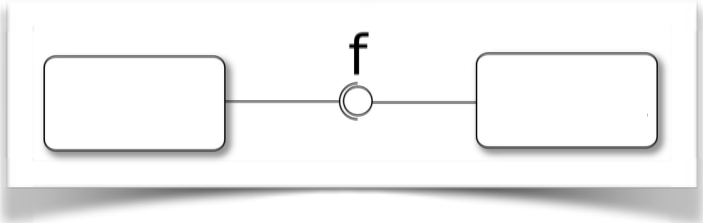
Key: proved **isolation**

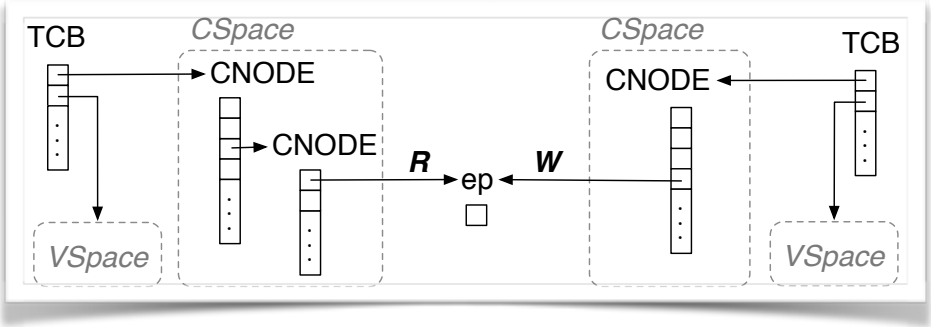




Building systems - component platform

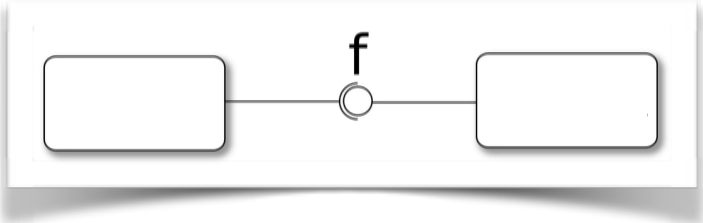
CAmkES





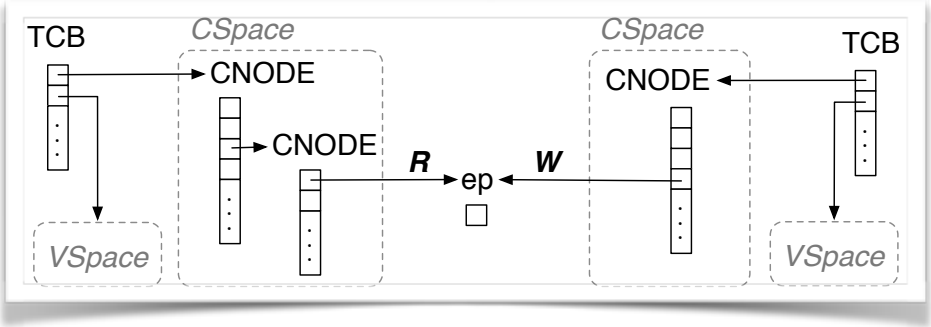
Building systems - component platform

CAmkES

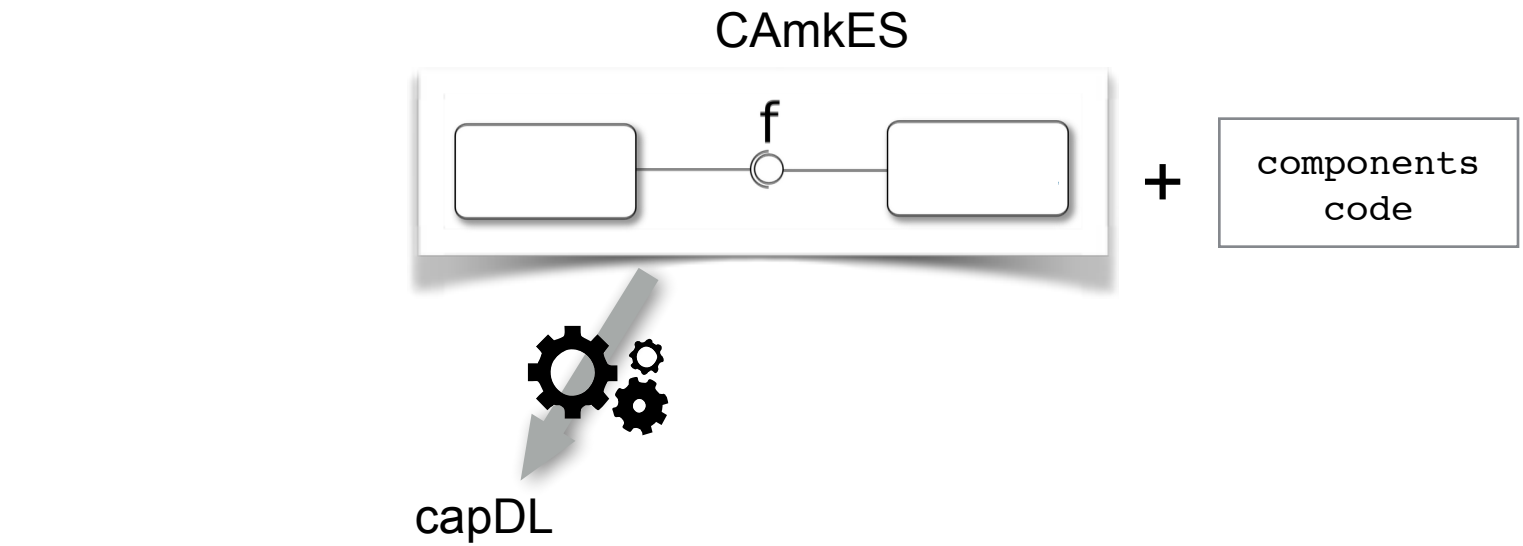


\+

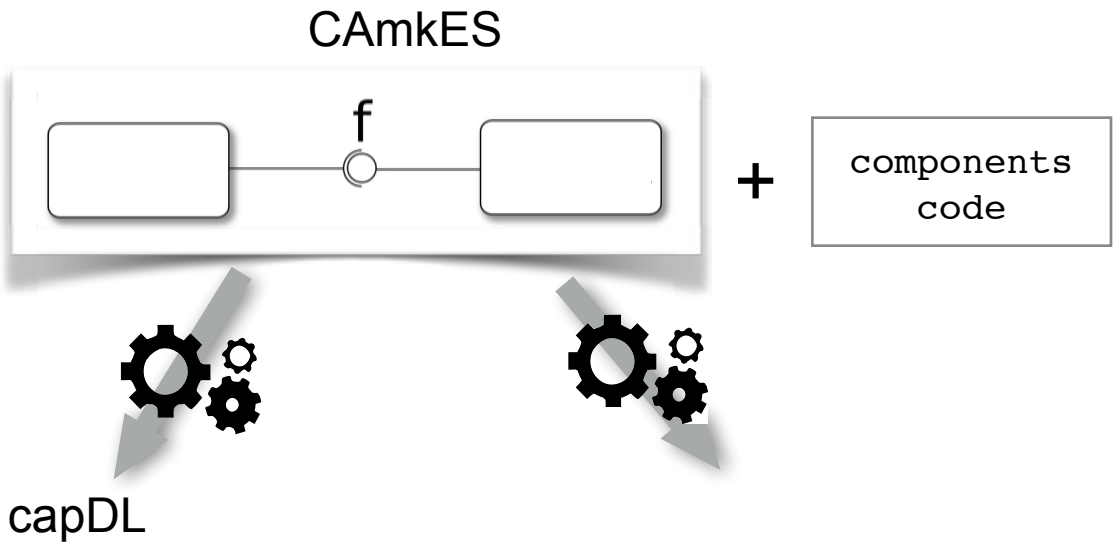
```
components
code
```



Building systems - component platform

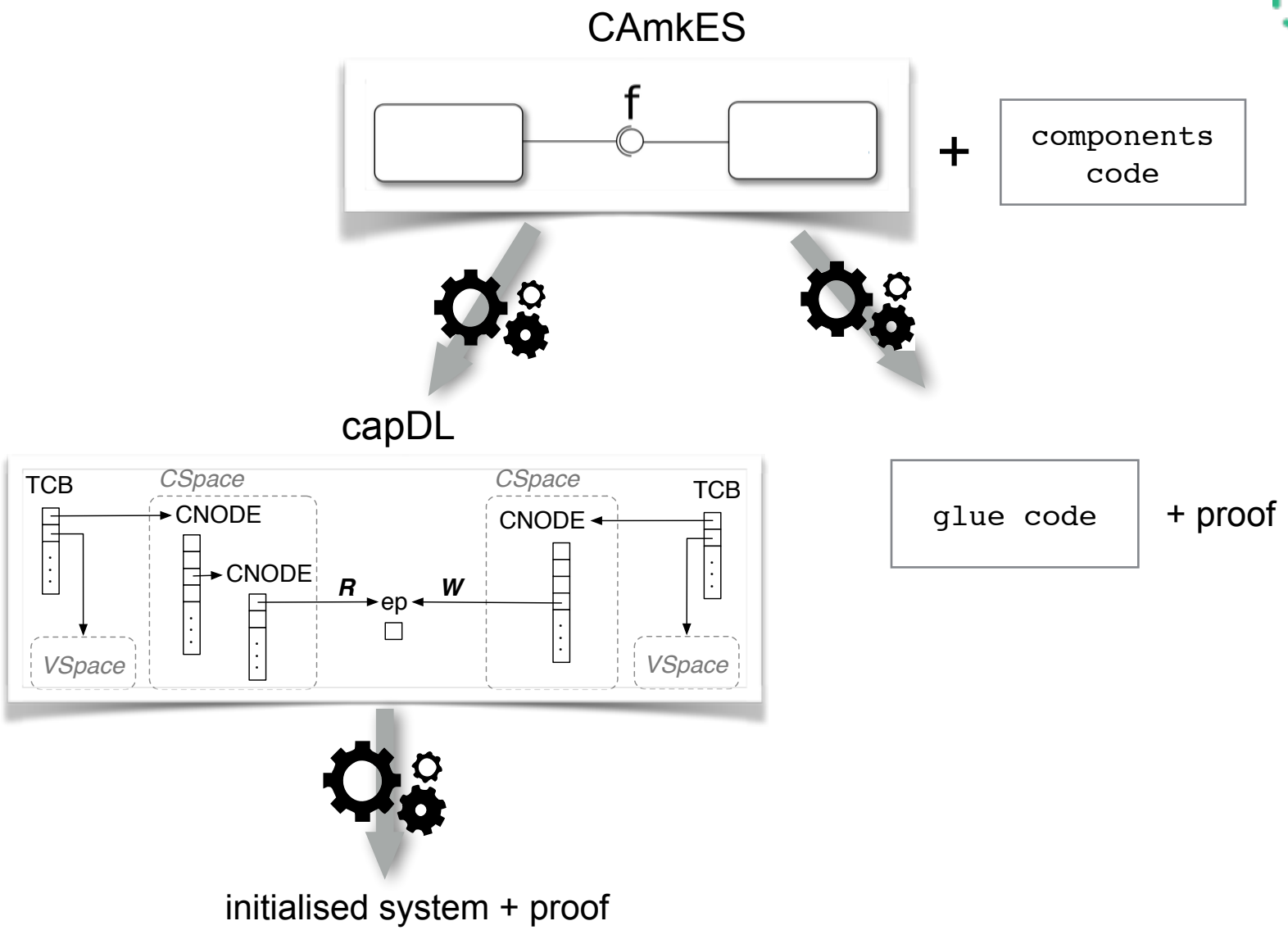


Building systems - component platform

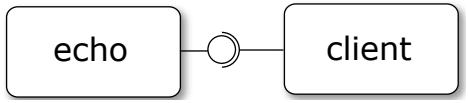


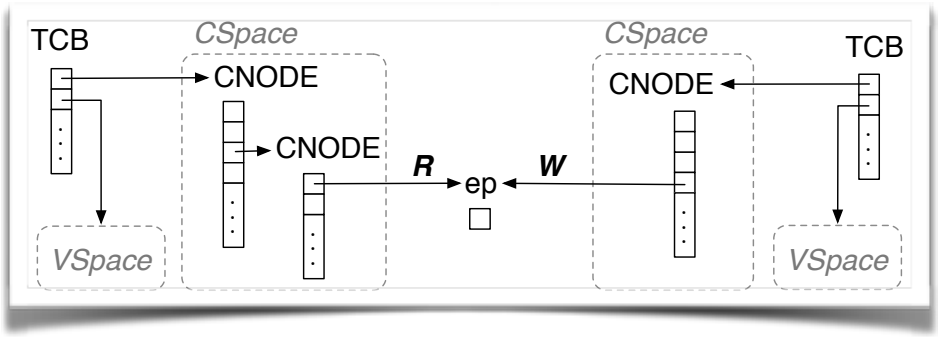
glue code + proof

Building systems - component platform

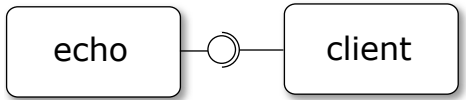


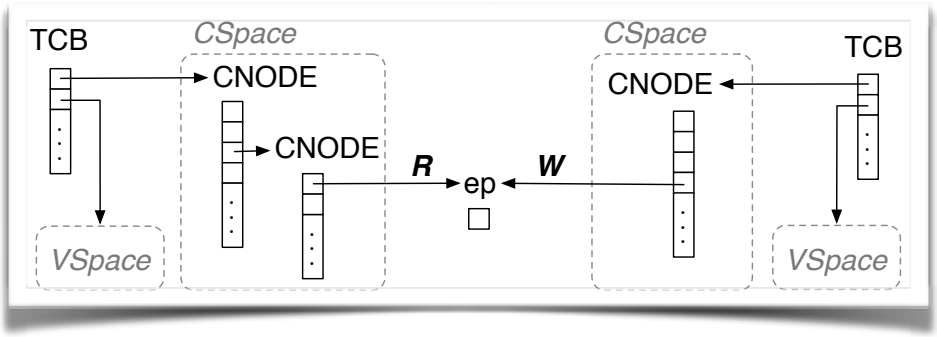
Building systems - initialisation

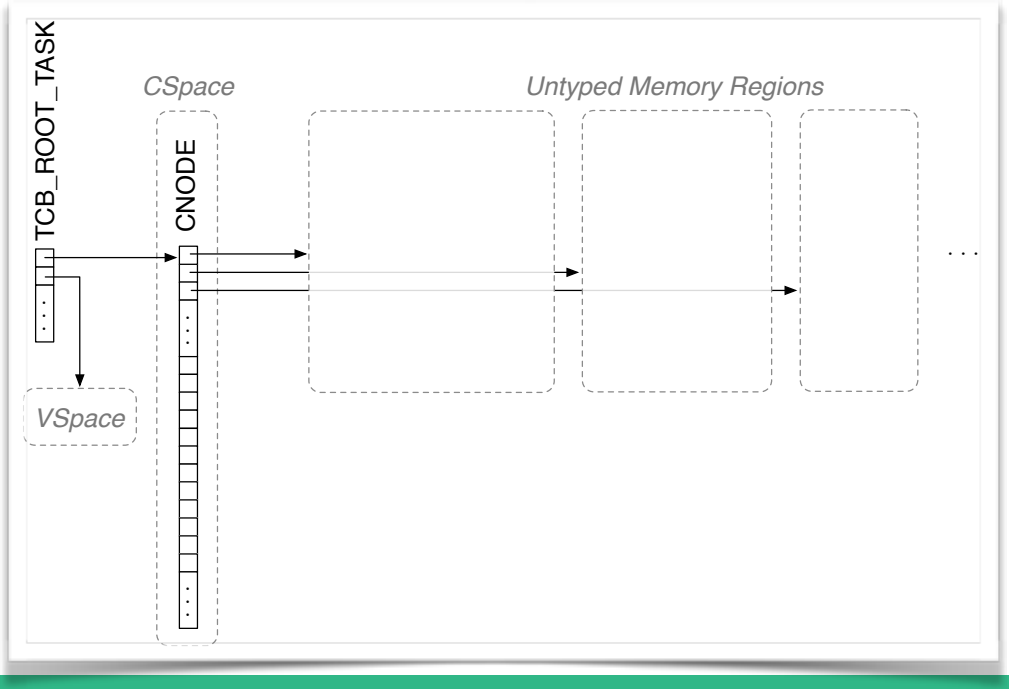




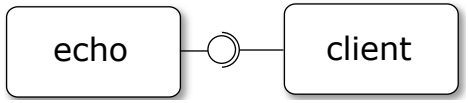
Building systems - initialisation

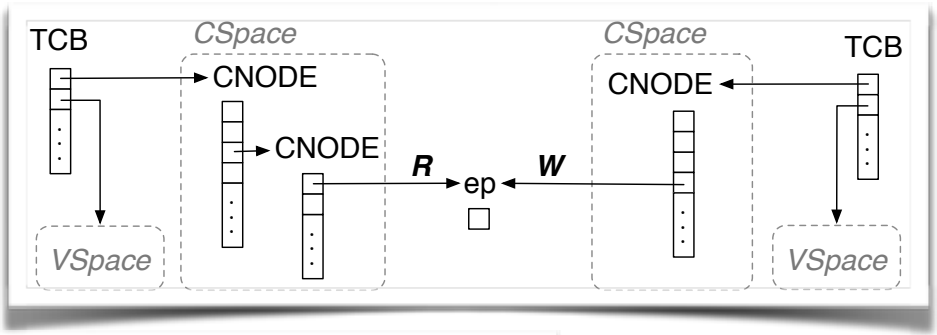


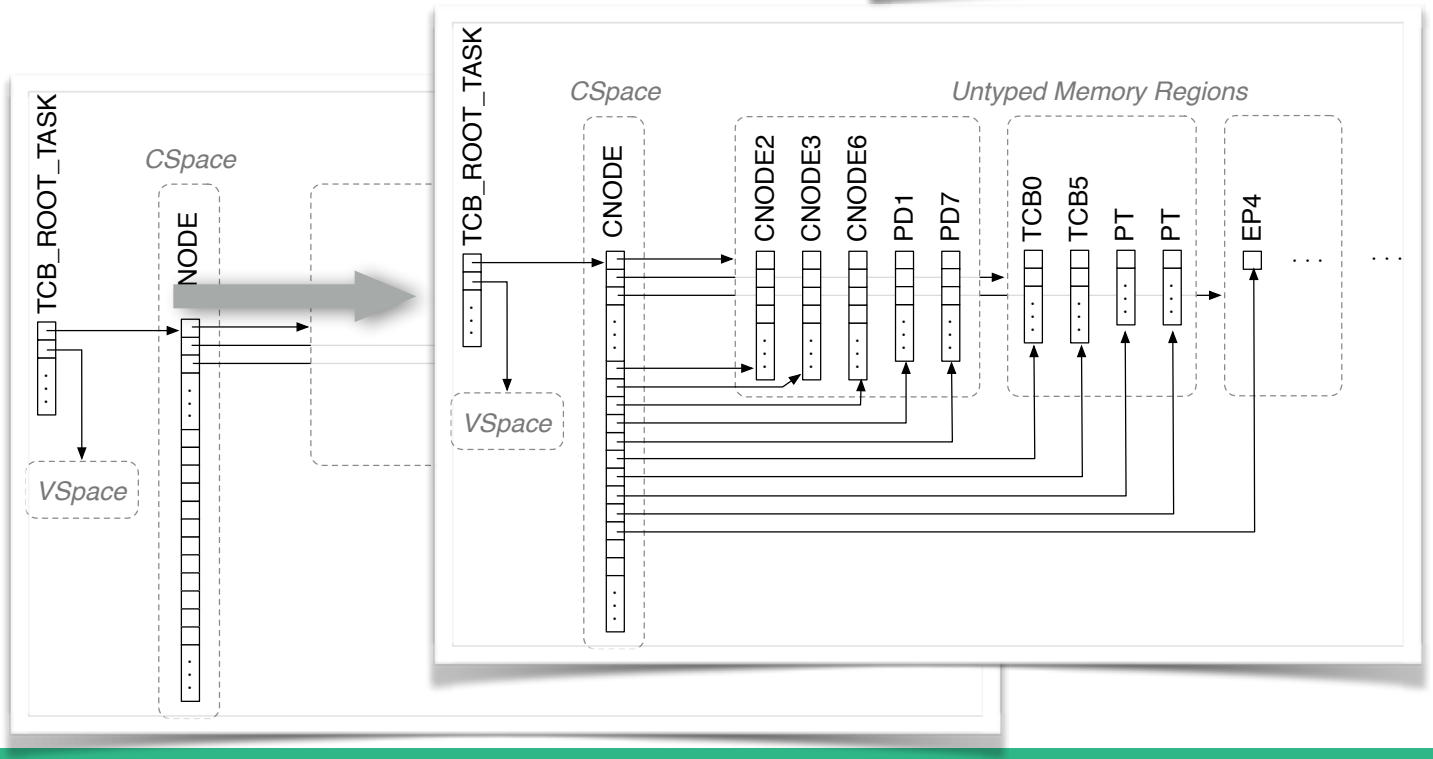




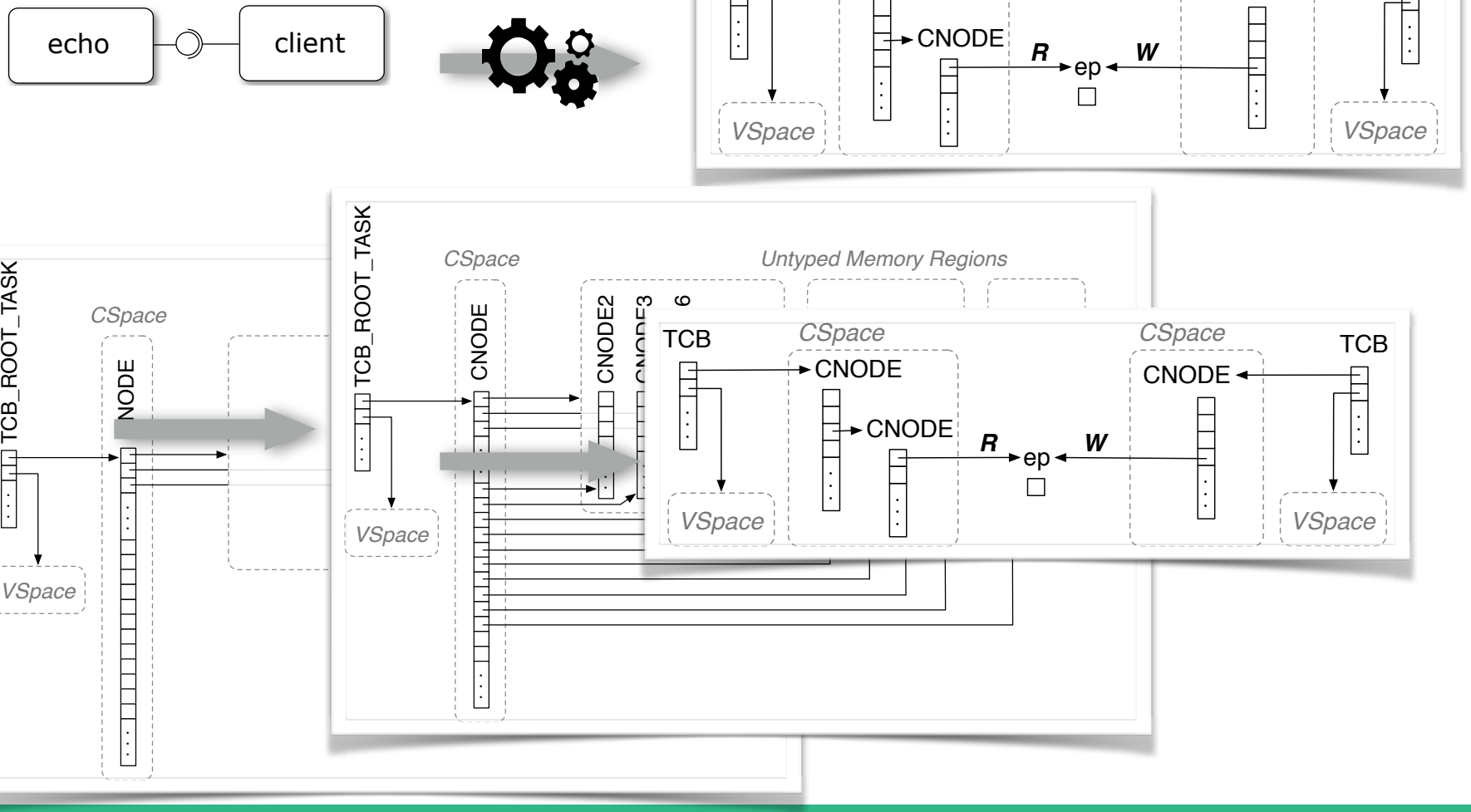
Building systems - initialisation



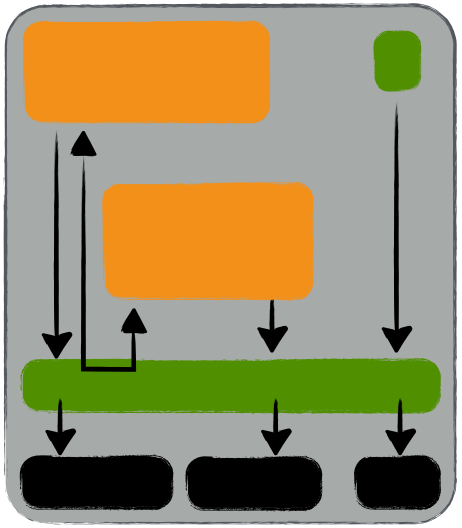




Building systems - initialisation



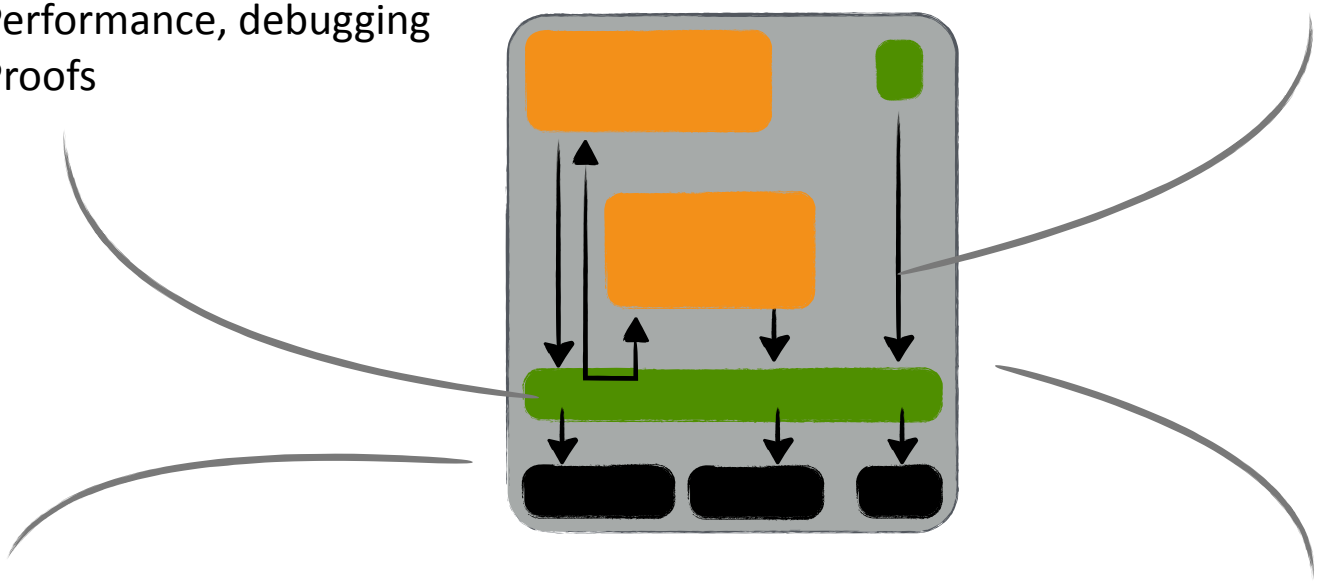
seL4 & family: an ecosystem



seL4 & family: an ecosystem

seL4 Kernel
- Platform ports
- Performance, debugging
- Proofs

seL4 Platform
- Libraries, CAmkES, Driver framework…
- OS system components, VMM…



Development support
- seL4test, continuous integration
- Debugging
- Benchmarking

Support
- Documentation
- Community support

Deployment: in DARPA HACMS project

➔ Verified software does protect against cyber attacks

Deployment: in DARPA HACMS project

➔ Verified software does protect against cyber attacks

Deployment: in DARPA HACMS project

➔ Verified software does protect against cyber attacks

Quadcopter

Land robot

Unmanned Helicopter

Autonomous Trucks

DARPA Rockwell Collins galois DATA 61 CSIRO University of Minnesota BOEING

Deployment: in DARPA HACMS project

➔ Verified software does protect against cyber attacks

Quadcopter

Land robot

Unmanned Helicopter

Autonomous Trucks

DARPA Rockwell Collins galois DATA 61 CSIRO University of Minnesota BOEING

Deployment: in DARPA HACMS project

➔ Verified software does protect against cyber attacks

Quadcopter

Land robot

Unmanned Helicopter

Autonomous Trucks

DARPA Rockwell Collins galois DATA 61 CSIRO University of Minnesota BOEING

Deployment: in DARPA HACMS project

→ componentisation of unmanned air vehicles

Quadcopter

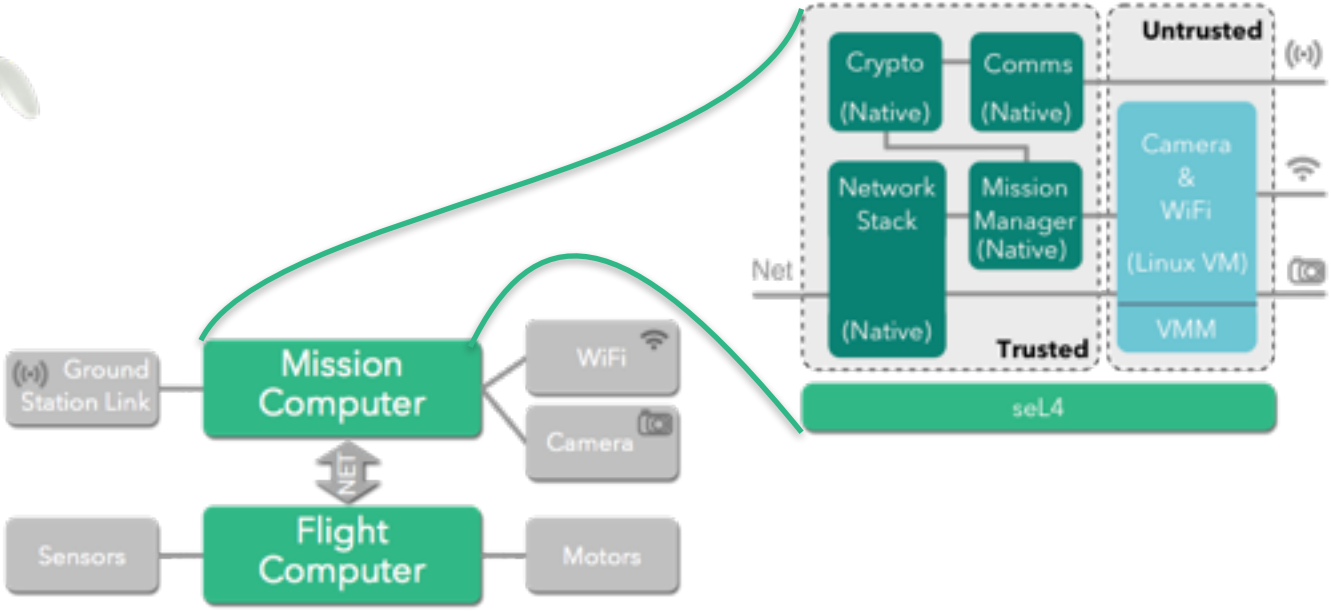
Unmanned Helicopter

Deployment: in DARPA HACMS project

→ componentisation of unmanned air vehicles

Quadcopter

Unmanned Helicopter

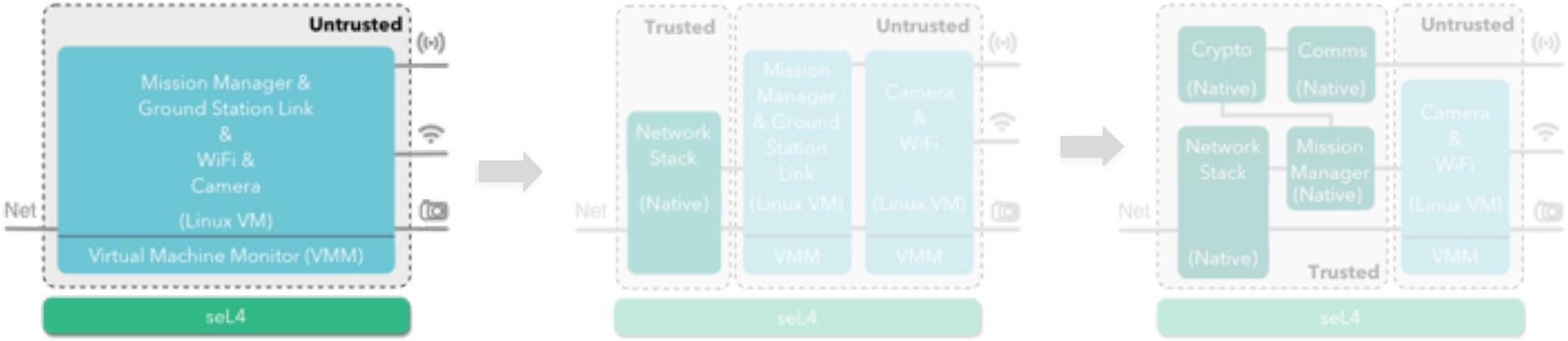


seL4 inside!

Security. Performance. Proof.

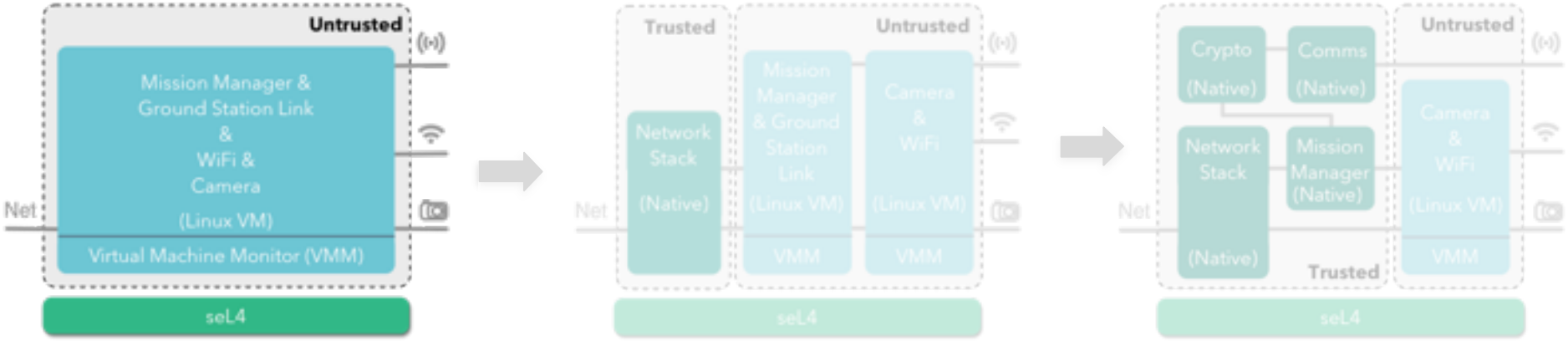
Deployment: in DARPA HACMS project

→ Retrofitting a system to be high-assurance



Deployment: in DARPA HACMS project

➔ Retrofitting a system to be high-assurance



First put all of the existing software inside a VM running on seL4

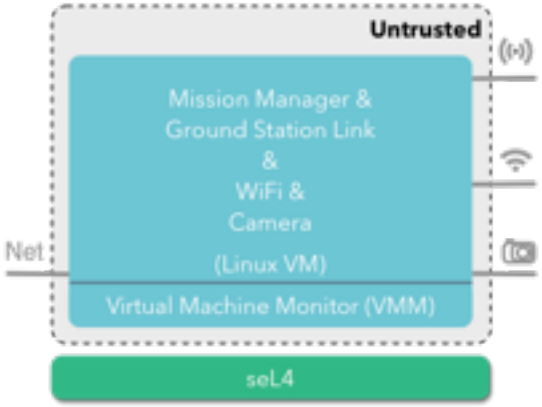
⬇

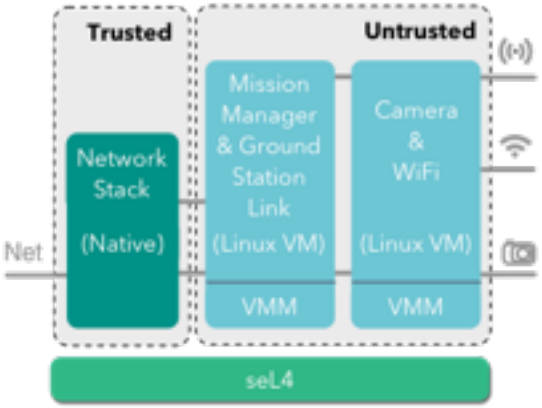
No security benefit yet,

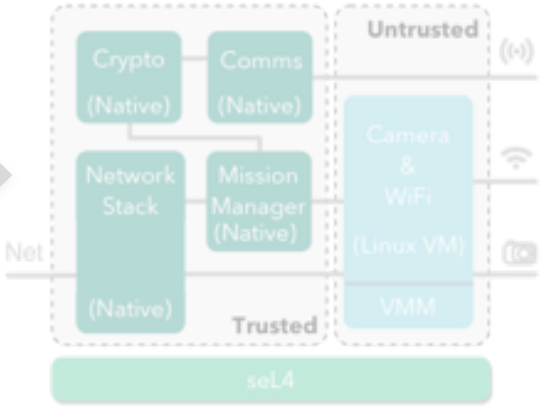
simply showing that seL4 runs on the target platform and that all the software can run virtualised

Deployment: in DARPA HACMS project

➔ Retrofitting a system to be high-assurance







First put all of the existing software inside a VM running on seL4

⬇

No security benefit yet,

simply showing that seL4 runs on the target platform and that all the software can run virtualised

Then start pulling **some** trusted components out of the VM to run natively on seL4

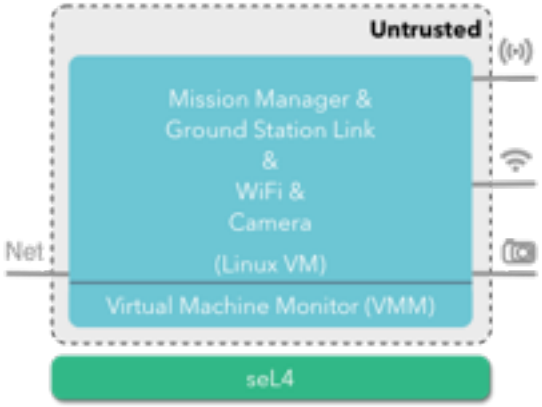
⬇

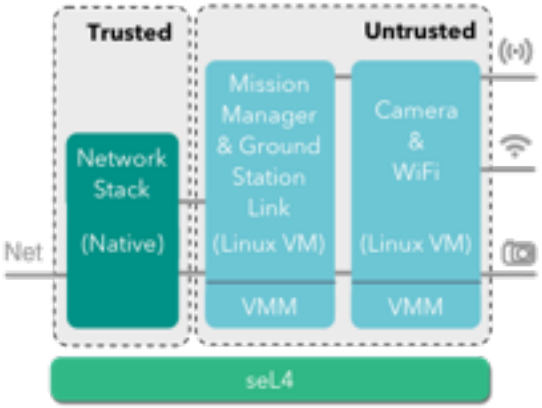
Some security benefit:

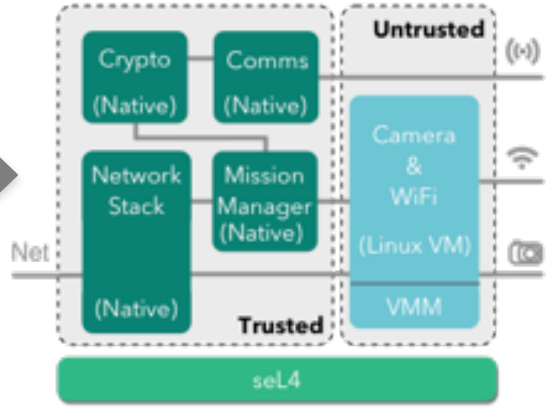
compromise in VM cannot propagate to trusted component

Deployment: in DARPA HACMS project

➔ Retrofitting a system to be high-assurance







First put all of the existing software inside a VM running on seL4

⬇

No security benefit yet,

simply showing that seL4 runs on the target platform and that all the software can run virtualised

Then start pulling **some** trusted components out of the VM to run natively on seL4

⬇

Some security benefit:

compromise in VM cannot propagate to trusted component

Full security architecture, with **all** trusted components running as a seL4 component

⬇

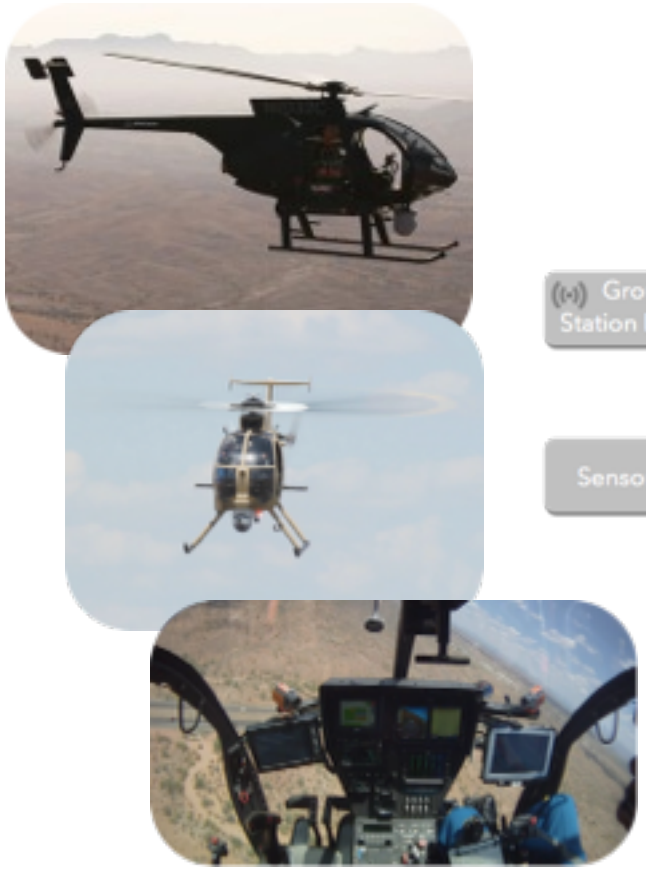
Important security benefit:

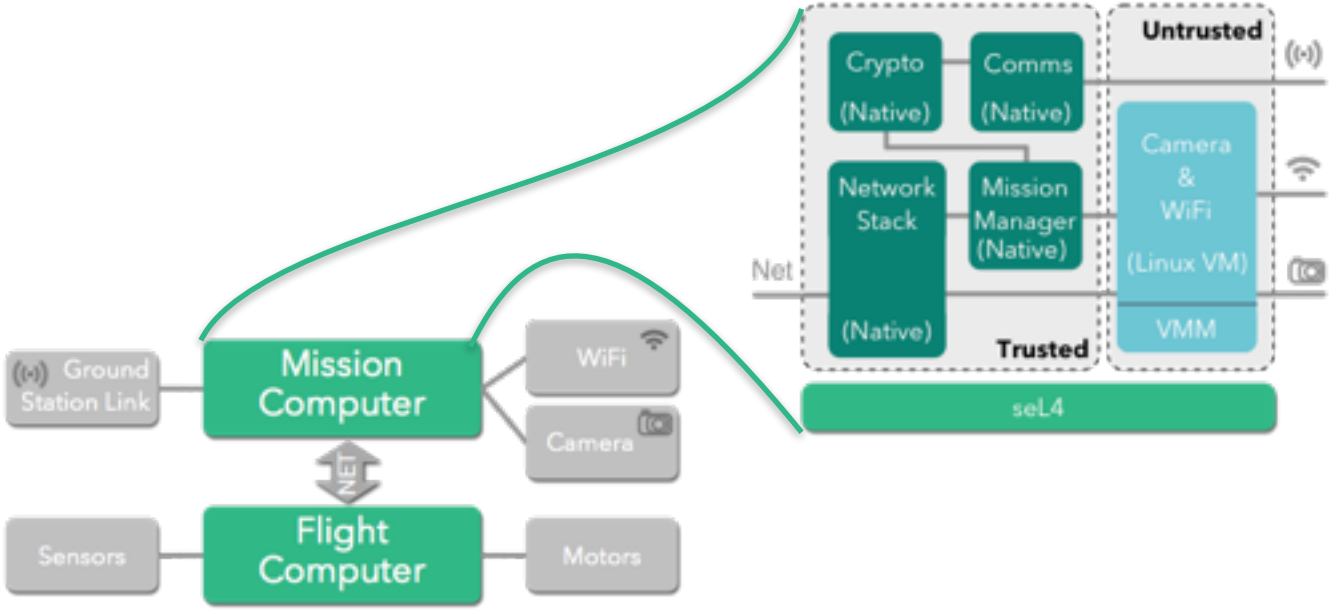
All components run isolated in a container, only the VM is still vulnerable

Deployment: in DARPA HACMS project

➔ ***Red-Team*** *could take control over the camera and Linux VM.*

➔ *Red-team could* ***not*** *send malicious commands to flight computer.*





seL4 inside!

Security. Performance. Proof.

Deployment - beyond

HACMS




Deployment - beyond

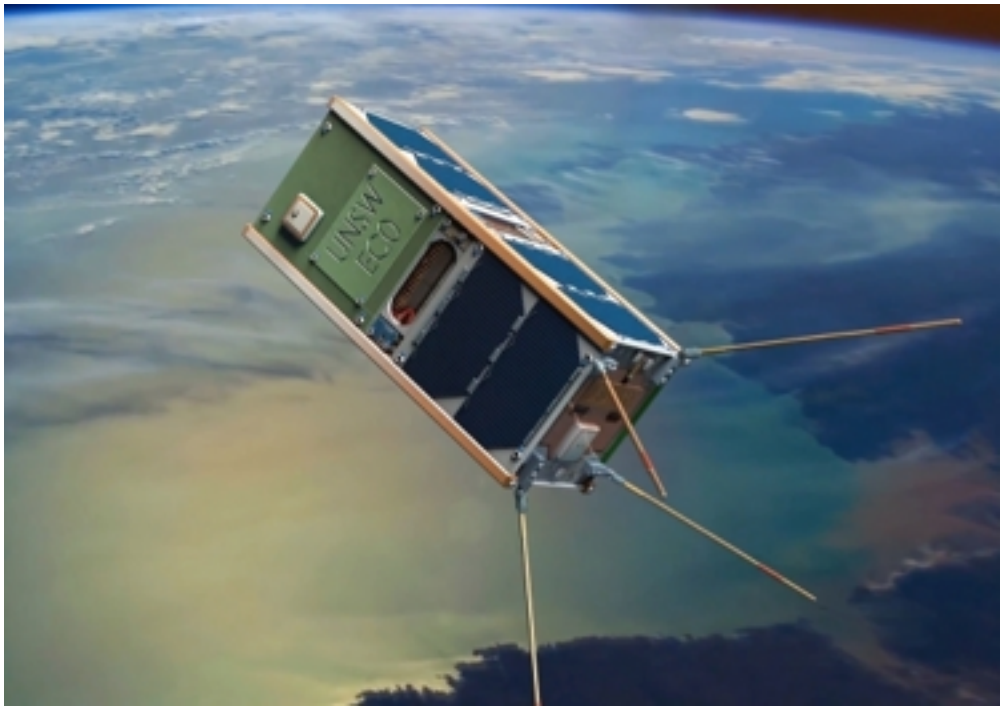
HACMS				QB50
				

Deployment - beyond

HACMS	QB50







UNSW-ECO CUBESAT
WAS LAUNCHED FROM
CAPE CANAVERAL
ON APRIL 19TH 2017
AT APPROXIMATELY
1:11AM SYDNEY TIME

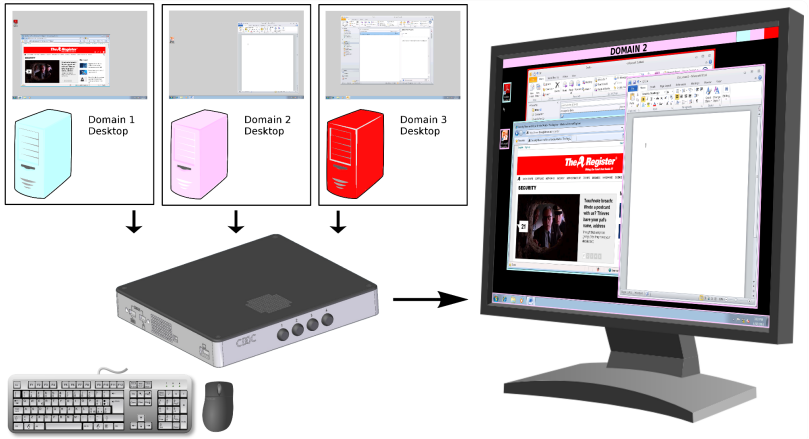
Deployment - beyond

HACMS				QB50	CDDC
					

Deployment - beyond

HACMS				QB50	CDDC
					

CDDC: Cross-Domain Desktop Compositor



3 state iAwards!
2 national iAwards!

Deployment - beyond

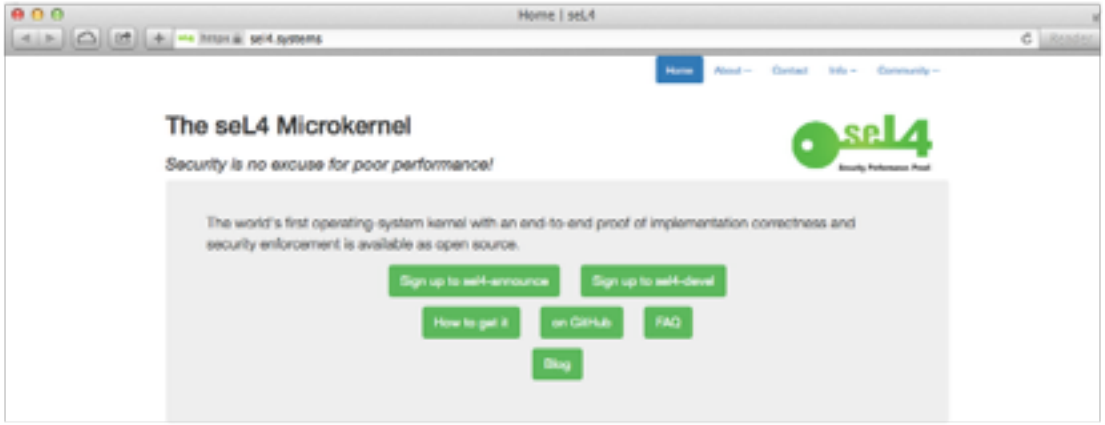
HACMS	QB50	CDDC






OPEN SOURCE!

https://github.com/sel4	Mailing list, IRC	Website, Wiki, Blog	Developer days



Deployment - beyond

HACMS	QB50	CDDC






OPEN SOURCE!

https://github.com/sel4	Mailing list, IRC	Website, Wiki, Blog	Developer days

SBIR+community

Deployment - beyond

HACMS	QB50	CDDC






OPEN SOURCE!

https://github.com/sel4	Mailing list, IRC	Website, Wiki, Blog	Developer days

SBIR+community

helmets	satellites	submarines	wireless storage	communications dongle	…

Deployment - beyond

HACMS	QB50	CDDC






OPEN SOURCE!

- https://github.com/sel4
- Mailing list, IRC
- Website, Wiki, Blog
- Developer days

SBIR+community

- helmets
- satellites
- submarines
- wireless storage
- communications dongle
- …

interest from

- IoT
- Automotive
- Defence
- …

Take away

Making verified software a
reality
in real-world systems

Approach:

➔ minimal & verified TCB

➔ ecosystem: seL4&co

Deployment

➔ projects

➔ community!






Take away

Making verified software a
reality
in real-world systems

Approach:

- ➔ minimal & verified TCB
- ➔ ecosystem: seL4&co

Deployment

- ➔ projects
- ➔ community!

Repeat?






Overview

Making verified software a **reality** in real-world systems

Remaining challenges to **mainstream** verified software

Approach:

cheaper ➔ proofs for free

relevant ➔ more features

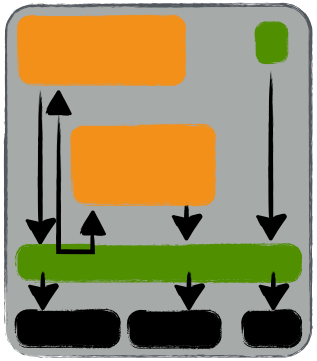
Deployment

scale ➔ proof engineering

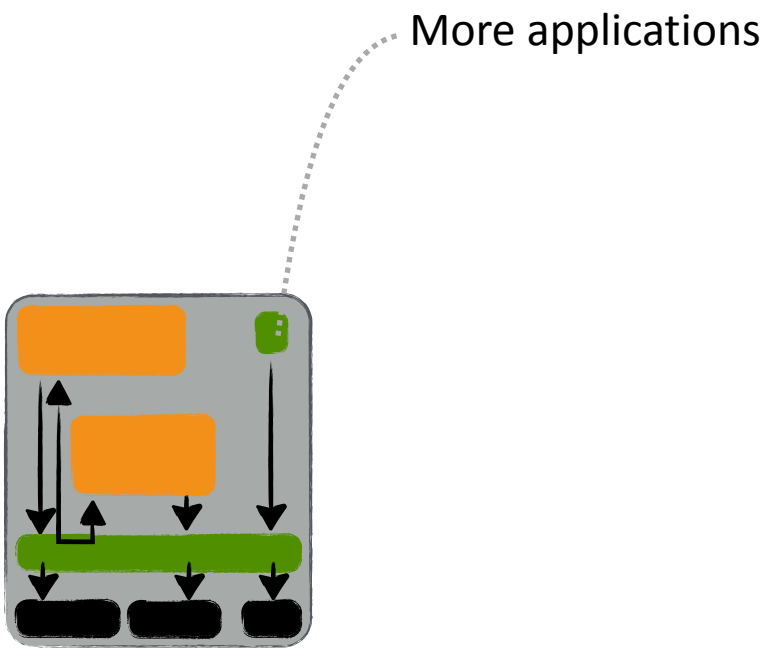





Now/Next



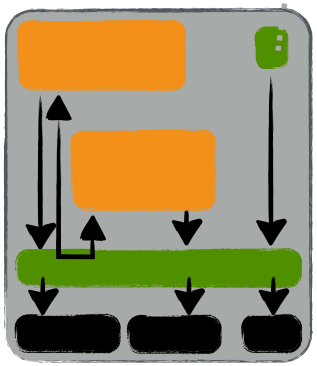
Now/Next



Now/Next

Cheaper

More applications

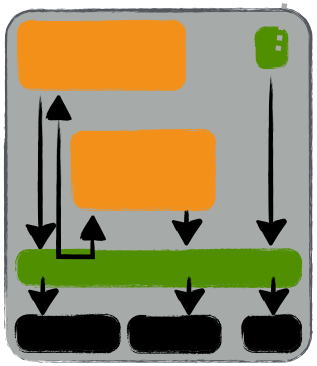


Now/Next

Cheaper

More applications

high-level languages



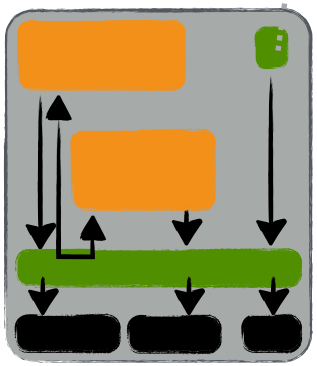
Now/Next

Cheaper

More applications

high-level languages

Cogent, CakeML, mVM



Now/Next

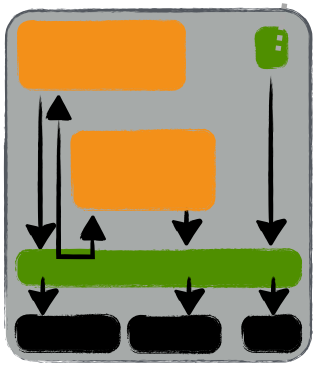
More features/guarantees

Cheaper

More applications

high-level languages

Cogent, CakeML, mVM

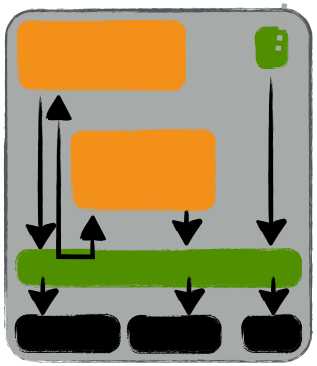


Now/Next

More features/guarantees
Real-time, multicore

Cheaper

More applications
high-level languages
Cogent, CakeML, mVM

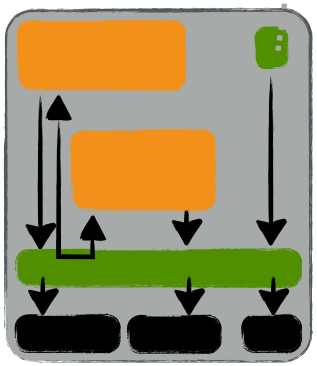


Now/Next

More features/guarantees
- Real-time, multicore
- Side-channels, WCET

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



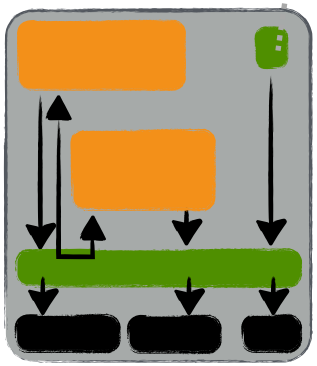
Now/Next

Relevant

More features/guarantees
- Real-time, multicore
- Side-channels, WCET

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



Now/Next

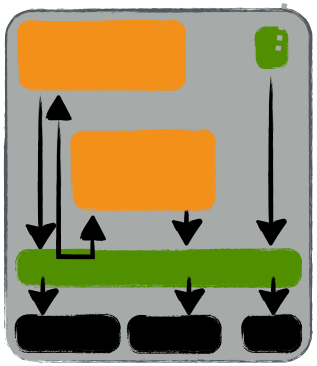
Relevant

More features/guarantees
- Real-time, multicore
- Side-channels, WCET

More usability
- platform support,
- platforms ports

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM

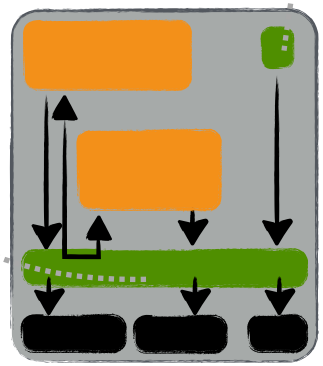


Now/Next

Relevant

More features/guarantees
- Real-time, multicore
- Side-channels, WCET

More usability
- platform support,
- platforms ports



Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM

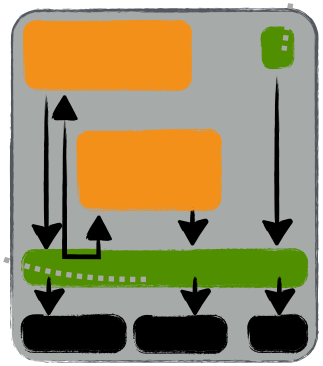
Now/Next

Relevant

More features/guarantees
- Real-time, multicore +verification!
- Side-channels, WCET

More usability
- platform support, platforms ports

+verification!



Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM

Now/Next

Relevant

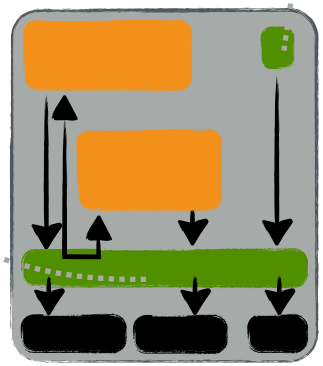
More features/guarantees
- Real-time, multicore +verification!
- Side-channels, WCET

More usability
- platform support, platforms ports

+verification!

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



Proof engineering!

Now/Next

Relevant

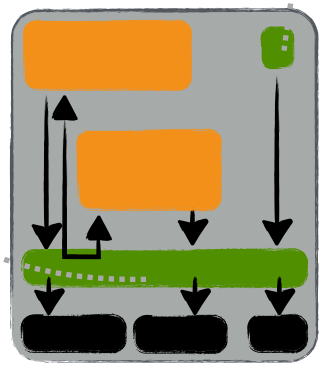
More features/guarantees
- Real-time, multicore +verification!
- Side-channels, WCET

More usability
- platform support, platforms ports

+verification!

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



Proof engineering!
- proof platform,
- proof development,
- proof maintenance

Now/Next

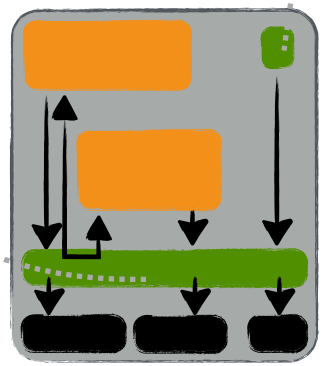
Relevant

More features/guarantees
- Real-time, multicore +verification!
- Side-channels, WCET

More usability
- platform support, platforms ports
  - +verification!

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



Scalable

Proof engineering!
- proof platform,
  proof development,
  proof maintenance

Now/Next

https://sel4.systems/Info/Roadmap/

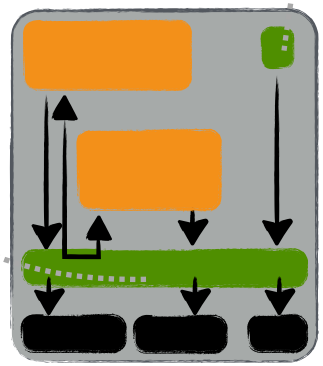
Relevant

More features/guarantees
- Real-time, multicore +verification!
- Side-channels, WCET

More usability
- platform support, platforms ports
- +verification!

Cheaper

More applications
- high-level languages
  - Cogent, CakeML, mVM



Scalable

Proof engineering!
- proof platform,
- proof development,
- proof maintenance

Take-away message

We have produced **verified technology**
that is **high-performance**
and is now **in use** in various systems in the world

We aim to **radically change** the way the world builds secure systems,
and **mainstream** verified software
by increasing **automation**, **proof engineering**, **community support**

Take-away message

We have produced **verified technology**
that is **high-performance**
and is now **in use** in various systems in the world

We aim to **radically change** the way the world builds secure systems,
and **mainstream** verified software
by increasing **automation**, **proof engineering**, **community support**

VERIFIED *and* FAST *and* CHEAP *and* DEPLOYED

DATA 61 CSIRO

TS @ Data61

https://trustworthy.systems/