Experiments on automation of formal verification of devices at the binary level

Thomas Lacroix

INSA Lyon
Soutenance de PFE (Option R&D)

19/06/2019

Section 1

Motivation

Table of Contents

HOL4

Proof-producing analysis

Machine-checked proofs

Table of Contents

Security critical systems

Privacy

- Smartphones
- Smart TVs

Security

- Hospital equipment
- Traffic control systems
- Power plants

Security critical systems - vulnerable

https://www.wired.com/2014/04/hospital-equipment-vulnerable/
It's Insanely Easy to Hack Hospital Equipment

https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
Hackers Remotely Kill a Jeep on the Highway—With Me in It

Security critical systems - vulnerable

Vulnerabilities come because of:

- Increased surface of attack (more and more features, codebases explode in size)
- Connected to networks $\rightarrow$ remote attacks

Secure operating systems

[1] https://sel4.systems/Info/FAQ/proof.pml

Secure operating systems

Formal proof[1]:

- The binary code correctly implements its **abstract specification**.
- The specification guarantees **integrity** and **confidentiality**.

[1] https://sel4.systems/Info/FAQ/proof.pml

Secure operating systems

Formal proof[1]:

- The binary code correctly implements its **abstract specification**.
- The specification guarantees **integrity** and **confidentiality**.

- **Integrity**: data cannot be *changed* without permission.
- **Confidentiality**: data cannot be *read* without permission.

[1] https://sel4.systems/Info/FAQ/proof.pml

Secure operating systems

Proof assumptions[2]:

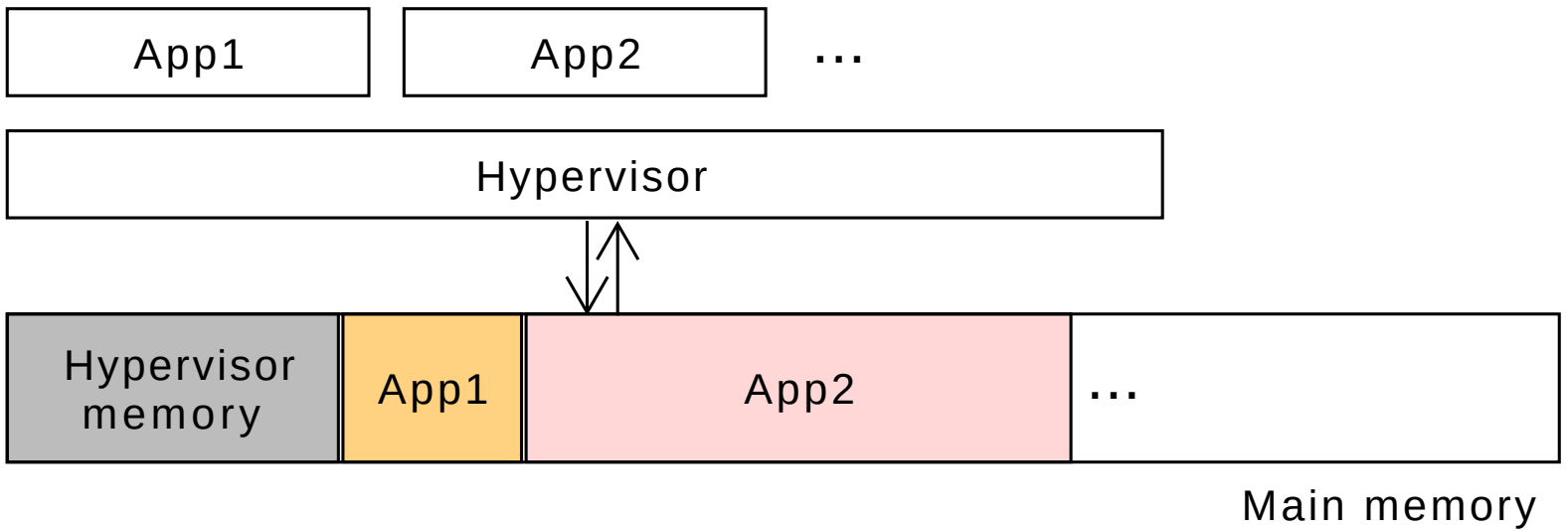
[2] https://docs.sel4.systems/FrequentlyAskedQuestions#is-sel4-proven-secure

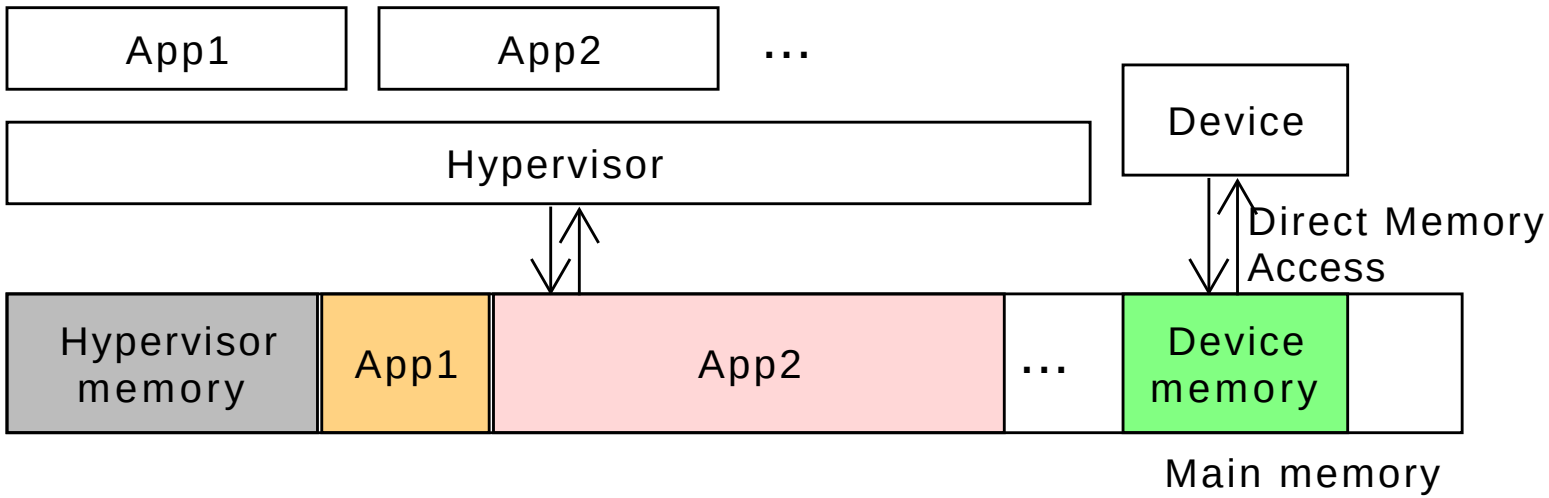
Secure operating systems

Proof assumptions[2]:

- Use of Direct Memory Access (DMA) is excluded, or only allowed for **trusted drivers that have to be formally verified by the user**.

[2] https://docs.sel4.systems/FrequentlyAskedQuestions#is-sel4-proven-secure


App1
App2
...
Hypervisor
Hypervisor memory
App1
App2
...
Main memory


App1
App2
...
Device
Hypervisor
Direct Memory Access
Hypervisor memory
App1
App2
...
Device memory
Main memory

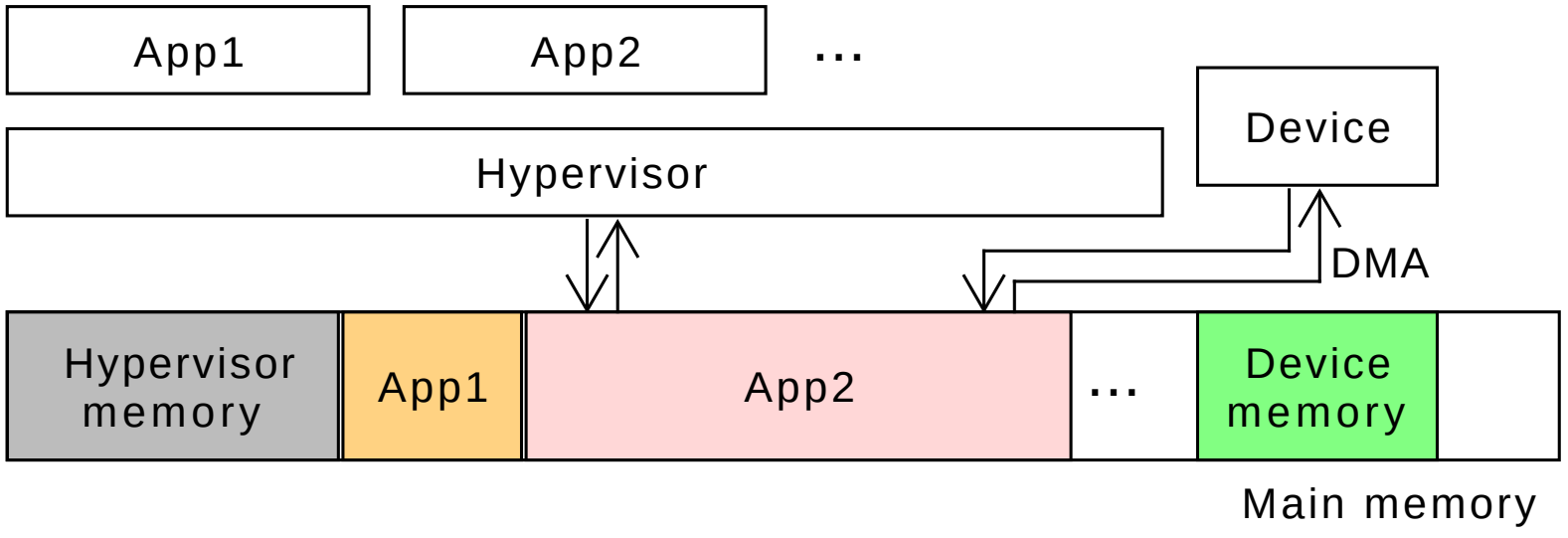

App1
App2
...
Device
Hypervisor
DMA
Hypervisor memory
App1
App2
...
Device memory
Main memory

Table of Contents

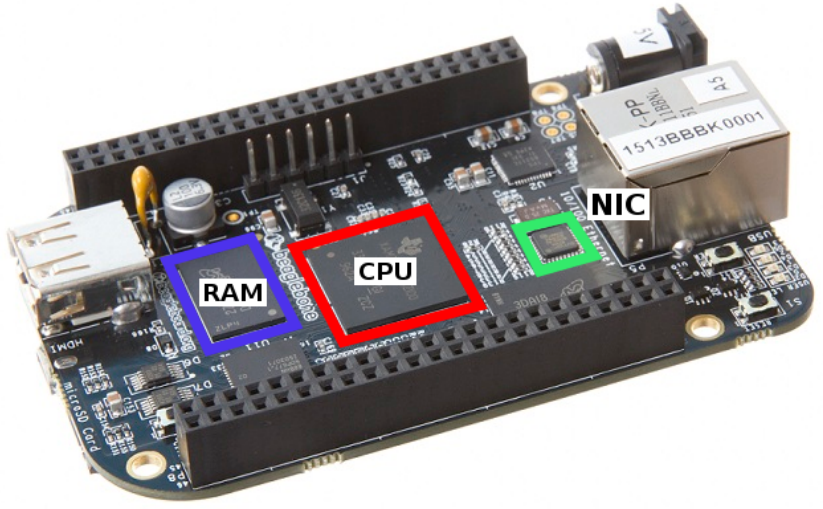
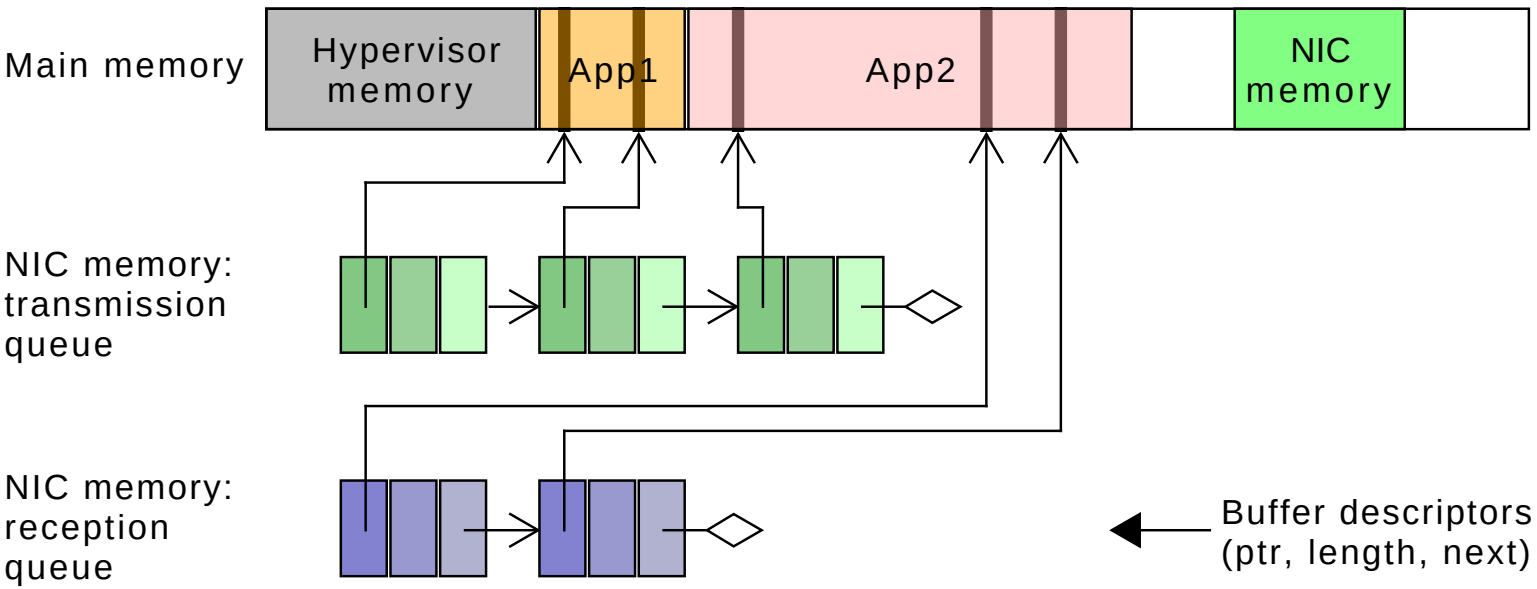
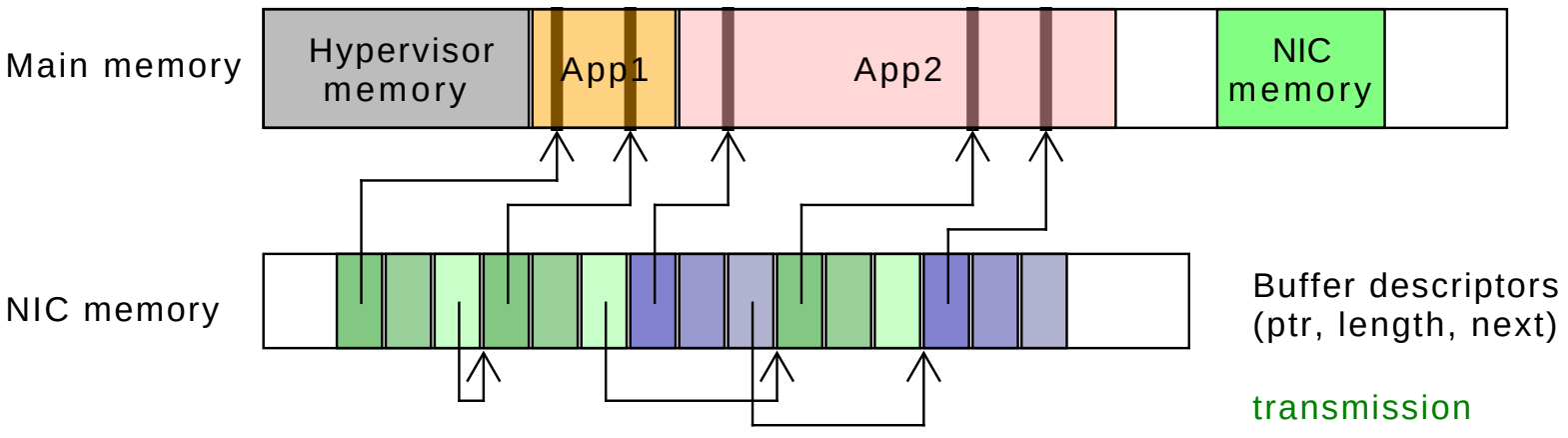
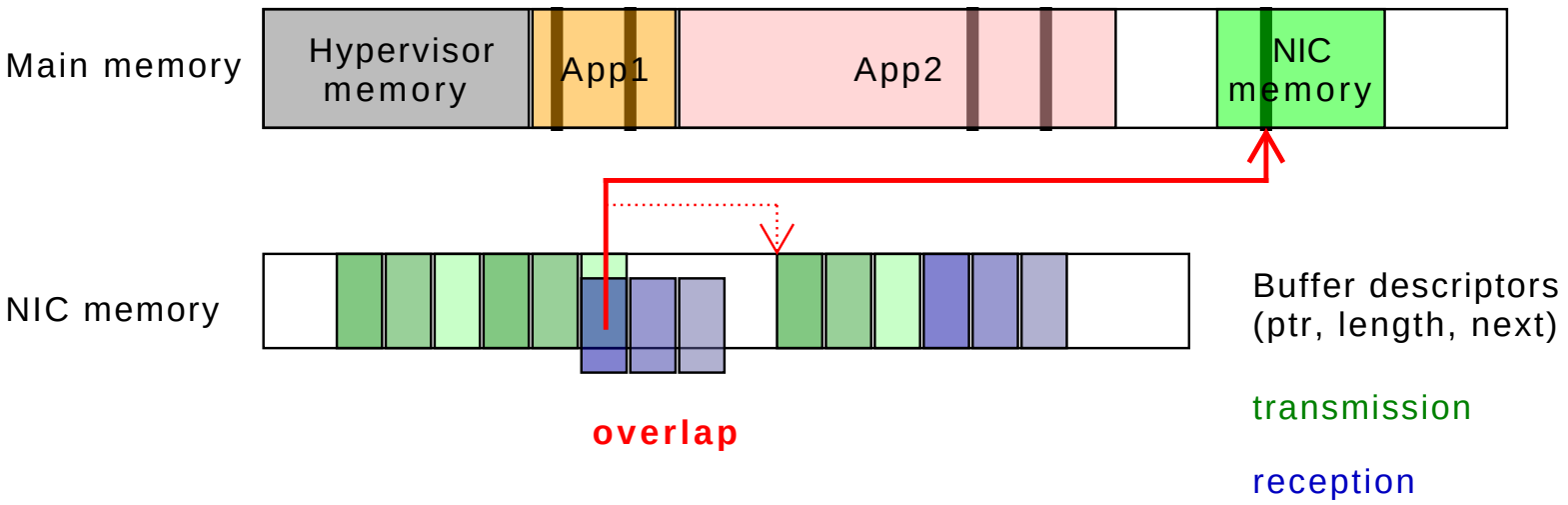


Figure: BeagleBone Black.


Main memory
Hypervisor memory
App1
App2
NIC memory
NIC memory: transmission queue
NIC memory: reception queue
Buffer descriptors (ptr, length, next)


Main memory
Hypervisor memory
App1
App2
NIC memory
NIC memory
Buffer descriptors
(ptr, length, next)
transmission
reception


Main memory
Hypervisor memory
App1
App2
NIC memory
NIC memory
overlap
Buffer descriptors (ptr, length, next)
transmission
reception

Research question

Can we apply traditional software verification techniques and tools to show security properties of hardware devices?

Section 2

Non proof-producing verification

Table of Contents

Title

Table of Contents

Title

Section 3

Proof-producing verification

Table of Contents

Title

Section 4

Conclusion

Questions

References I