Sovereign Smartphone

Srdjan Čapkun
ETH Zurich

(with Friederike Groschupp, Ivan Puddu, Moritz Schneider, Mark Kuhne, Shweta Shinde)

Google / Apple Ecosystems

- Most users locked into Android and iOS ecosystem

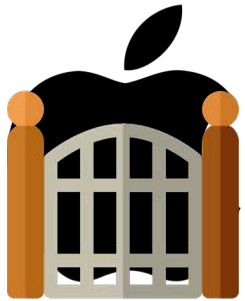
Google / Apple Ecosystems

- Most users locked into Android and iOS ecosystem
- Android is ‘nominally’ open source
- But many apps rely on closed source google services

Google / Apple Ecosystems

- Most users locked into Android and iOS ecosystem
- Android is ‘nominally’ open source
- But many apps rely on closed source google services

- => Locked into ecosystems with gatekeepers



‘Centralized’ Ecosystems: Top Notch Security

- Usability
- Performance
- Centralized APIs
- **Security Features**
- Rich app ecosystem
- …

So… is there a problem?

A security expert found that Apple's latest iPhone can still track your location data, even if you toggle it off for every app

Google tracks your movements, like it or not

So… is there a problem?

Fortnite Creator Sues Apple and Google After Ban From App Stores

So… is there a problem?

Europe

Google, Apple remove Navalny app from stores as Russian elections begin

Reuters

So… is there a problem?

Apple plans to scan US iPhones for child sexual abuse images

Security researchers fear neuralMatch system could be misused to spy on citizens

Bugs in our Pockets: The Risks of Client-Side Scanning

Hal Abelson Ross Anderson Steven M. Bellovin
Josh Benaloh Matt Blaze Jon Callas Whitfield Diffie
Susan Landau Peter G. Neumann Ronald L. Rivest
Jeffrey I. Schiller Bruce Schneier Vanessa Teague
Carmela Troncoso

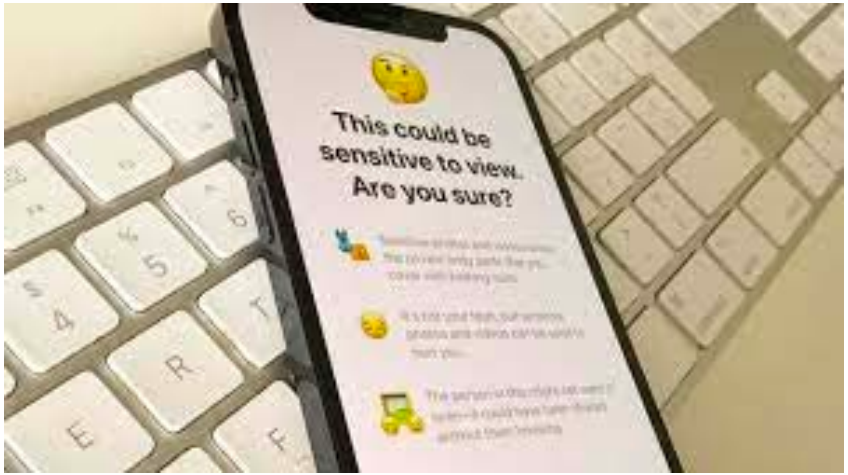
October 15, 2021

Executive Summary

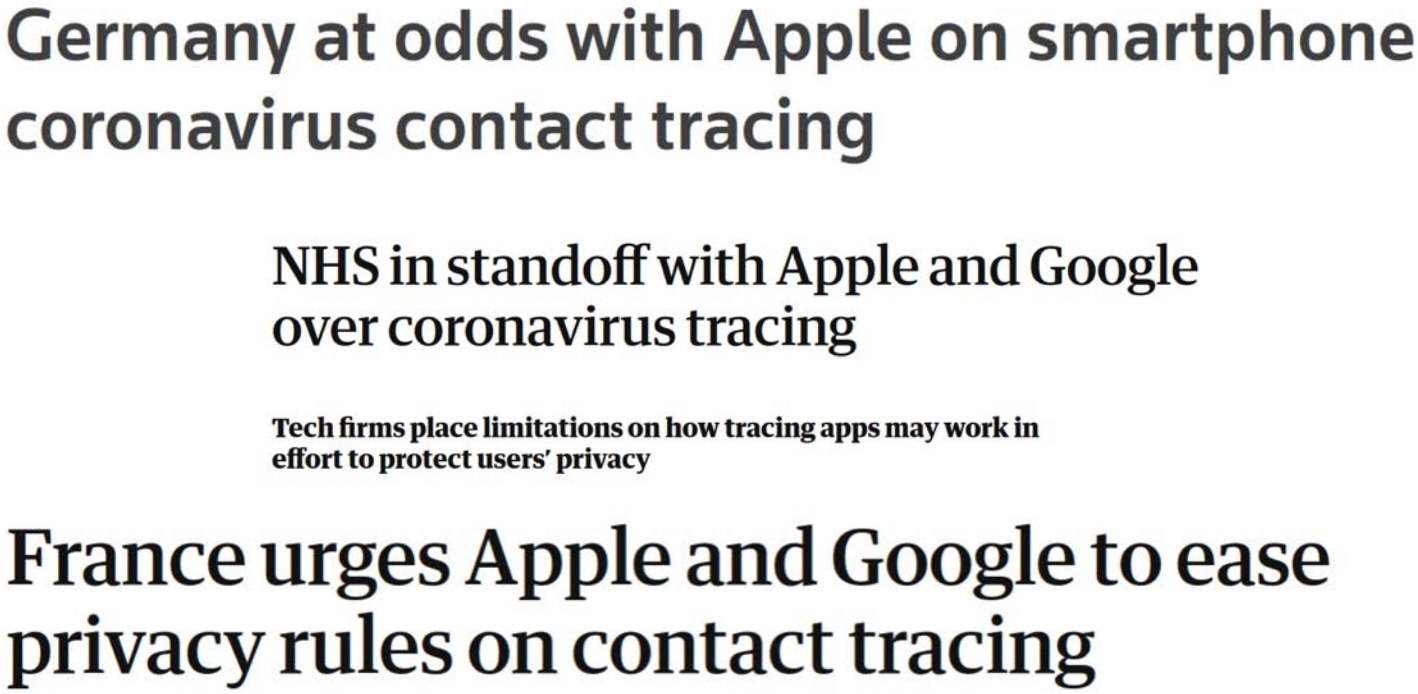
Our increasing reliance on digital technology for personal, economic, and government affairs has made it essential to secure the communications and devices of private citizens, businesses, and governments. This has led to pervasive use of cryptography across society. Despite its evident advantages, law enforcement and national security agencies have argued that the spread of cryptography has hindered access to evidence and intelligence. Some in industry and government now advocate a new technology to access targeted data: *client-side scanning* (CSS). Instead of weakening encryption or providing law enforcement with backdoor keys to decrypt communications, CSS would enable on-device analysis of data in the clear. If targeted information were detected, its existence and, potentially, its source, would be revealed to the agencies; otherwise, little or no information would leave the client device. Its proponents claim that CSS is a solution to the encryption versus public safety debate: it offers privacy—in the sense of unimpeded end-to-end encryption—and the ability to successfully investigate serious crime.

In this report, we argue that CSS neither guarantees efficacious crime prevention nor prevents surveillance. Indeed, the effect is the opposite. CSS by its nature creates serious security and privacy risks for all society while the assistance it can provide for law enforcement is at best problematic. There are multiple ways in which client-side scanning can fail, can be evaded, and can be abused.

Its proponents want CSS to be installed on all devices, rather than installed



So… is there a problem?



Germany at odds with Apple on smartphone coronavirus contact tracing

NHS in standoff with Apple and Google over coronavirus tracing

Tech firms place limitations on how tracing apps may work in effort to protect users' privacy

France urges Apple and Google to ease privacy rules on contact tracing

Contact Tracing and Apple/Google …

VIEWPOINT

Whose Smartphone Is It?

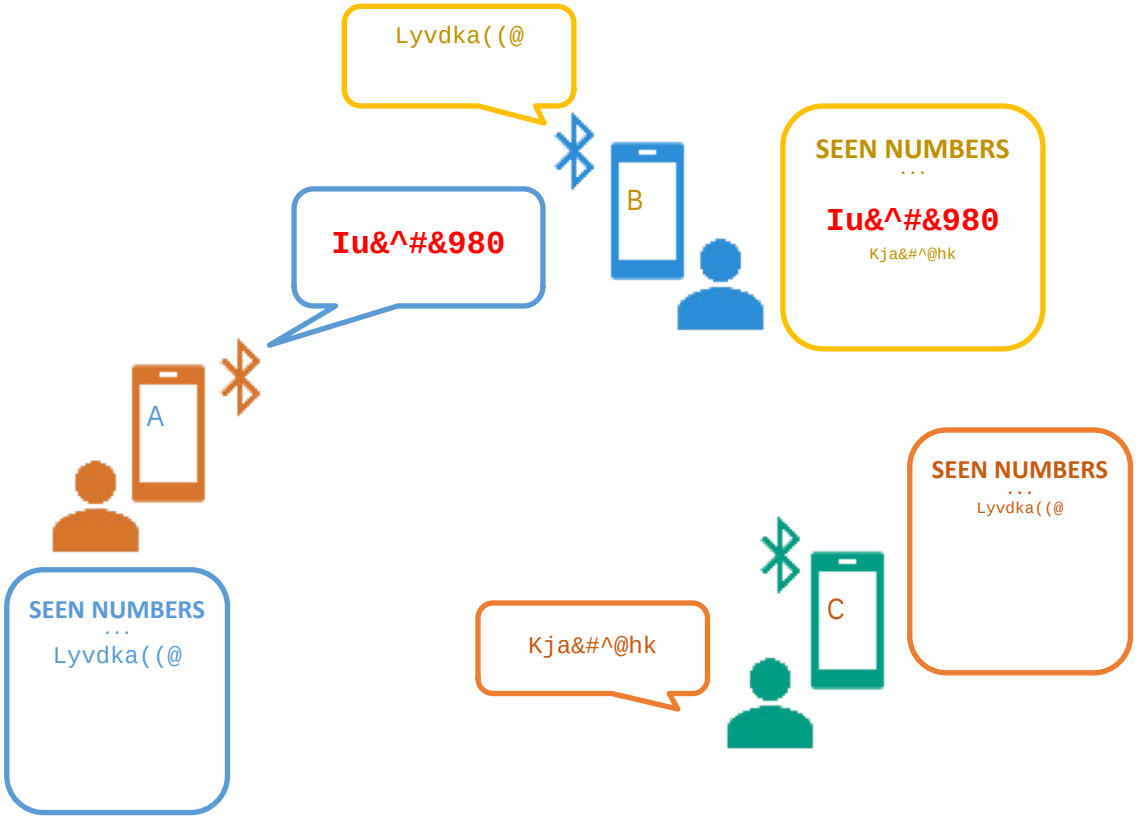
By James R. Larus
Communications of the ACM, September 2021, Vol. 64 No. 9, Pages 41-42
10.1145/3454007
Comments (1)

VIEW AS: SHARE:

On May 27, 2020, in the French National Assembly, Cédric O, th French Secretary of State for Digital Economy, forcibly expresse his government's frustration with Apple and Google in terms more appropriate to a cold war confrontation between superpowers. He noted that France and the U.K. were the two European countries building COVID-19 contact-tracing apps without these tech giants' assistance. These countries were also the only two European countries with nuclear weapons, the "acme of national sovereignty."[a]

The frustration of a modern state, unable to respond to the most severe public health crisis in a century because of two private companies' decisions, should give us all pause. Apple and Google

https://github.com/DP-3T



Google

But apps cannot access these identifiers!
(to protect users from evi)

Privacy-Preserving Contact Tracing

Motivation

Users should be in control of the phone.

What about…?

- Rooting / Jailbreaking
- Unlocking the bootloader, Open-source OSes
- Sideloading

What about…?

No Google apps or services

GrapheneOS will never include either Google Play services or another implementation of Google services like microG. It's possible to install Play services as a set of fully sandboxed apps without special privileges via our sandboxed Play services compatibility layer. See the FAQ section for more details on our plans for filling in the gaps from not shipping Play services and Google apps.

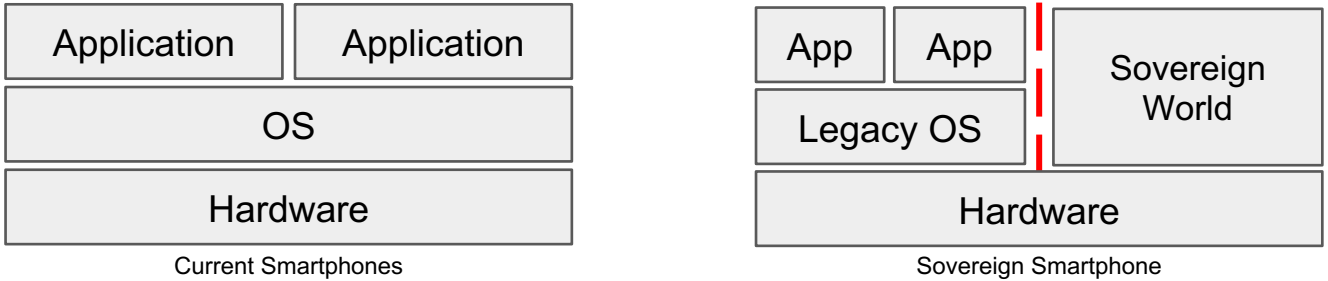
Motivation

Users should be in control of the phone.

But …

- **Allow legacy OS to protect apps and their ecosystem**
- Users should be able to protect 'Sovereign‘ (user, bare-metal) App/OS
- 'Common‘ trusted part as small as possible (who watches the watcher?)
  Best if no need to watch the watcher.

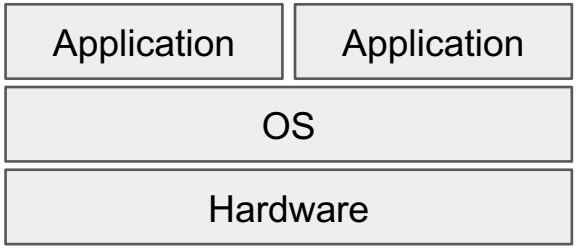
Ideally …



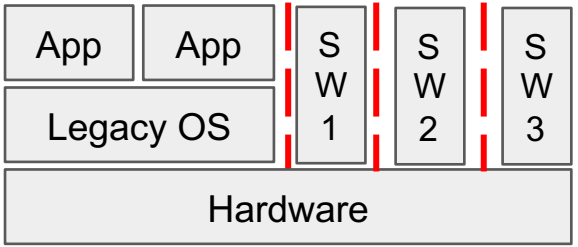
- Two isolated *worlds*: *legacy* and *sovereign world*
- Both worlds are equally privileged*
- User can assign resources to the worlds

*One world in charge of scheduling

Ideally …



Current Smartphones



Sovereign Smartphone

- Two isolated *worlds*: *legacy* and *sovereign world*
- Both worlds are equally privileged*
- User can assign resources to the worlds

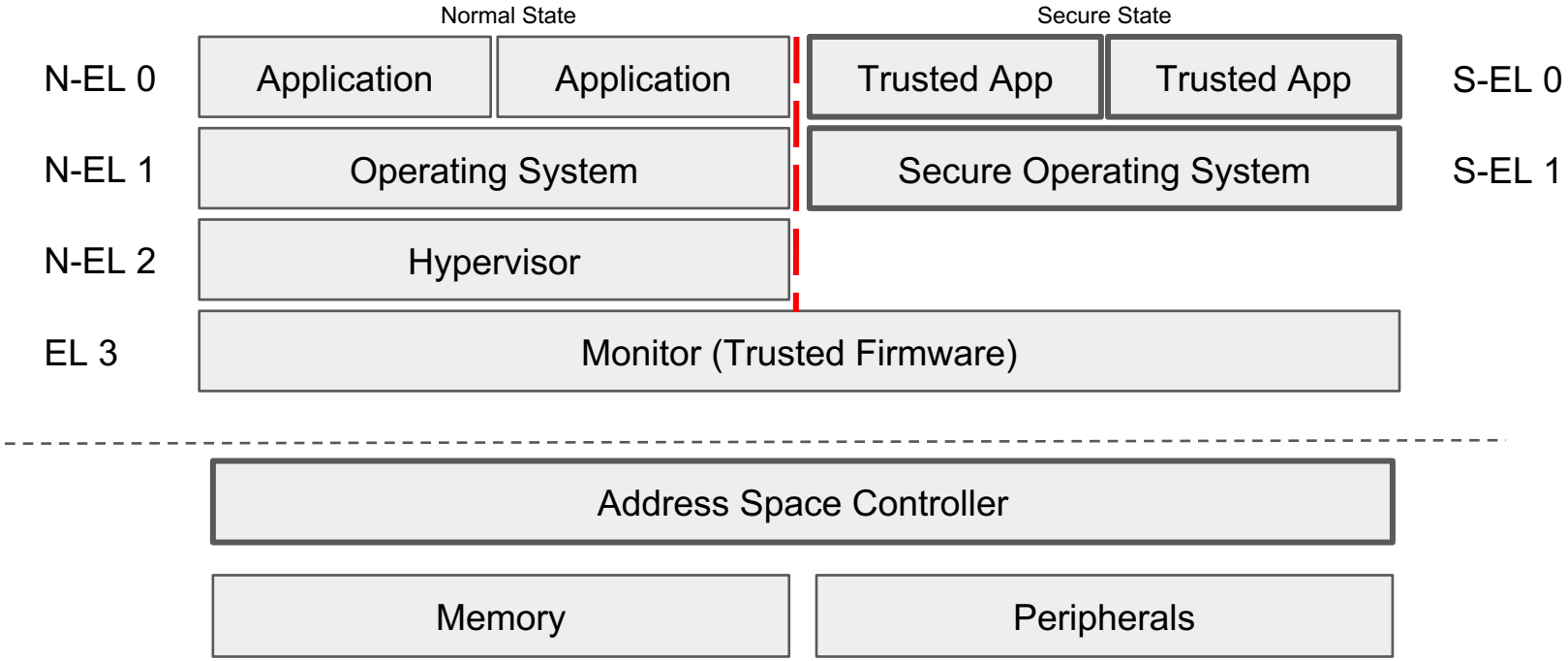
*One world needs to be in charge of scheduling

Hypervisor?

- Who will maintain the hypervisor?
- If not e.g., Apple / Google, how would they protect their OS / ecosystem?
- If Apple / Google then no point of having a hypervisor.

- We need a simple and small TCB that allows the coexistence of sovereign and legacy worlds. Ideally an ‘immutable’ TCB.

ARM TrustZone?



ARM TrustZone?

- Secure state is strictly more privileged than normal state
- Who maintains the secure OS?
- How do we protect legacy OS against from the secure OS?

- However, Trusted Firmware is trusted and enables context switching, communication, and memory isolation between normal and secure state

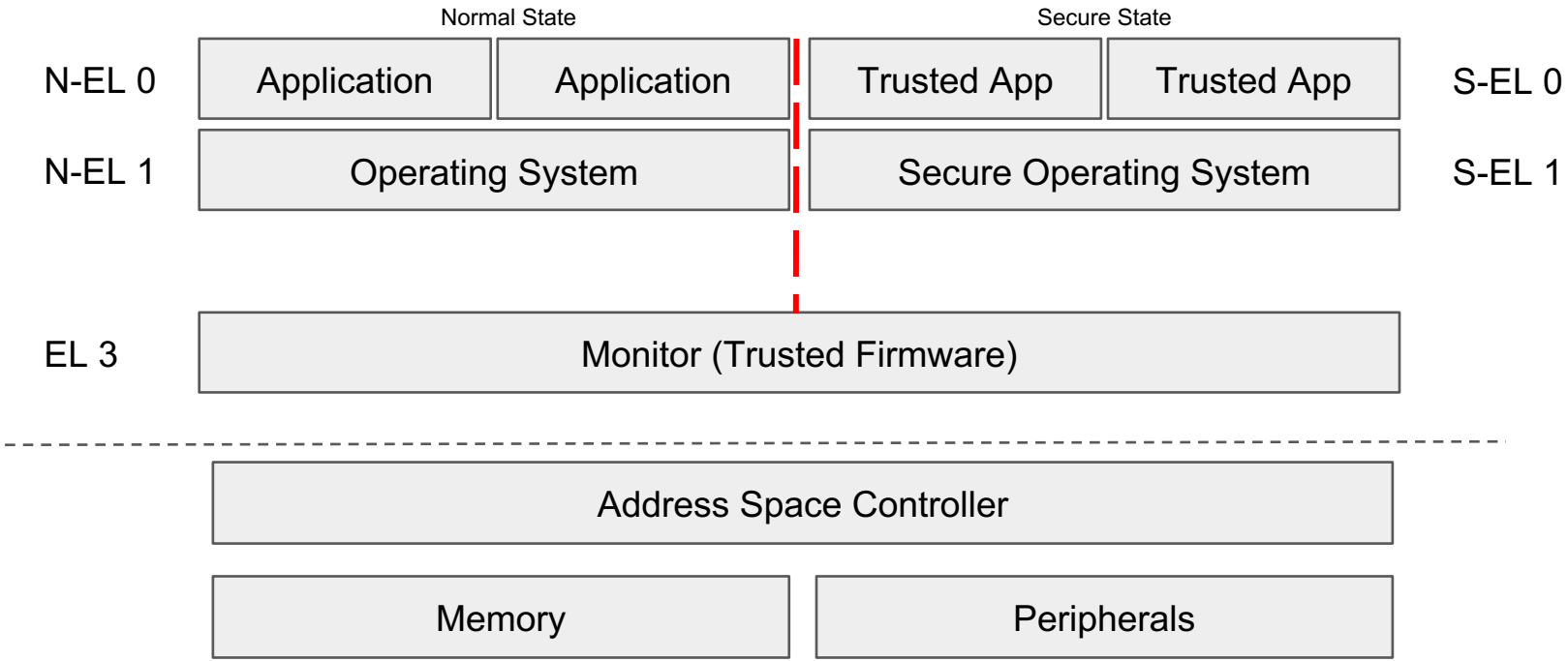
Some Initial Thoughts on “Sovereign Phone“ Design

- Extend Trusted Firmware with support for multiple isolated worlds in the normal state
- Address Space Controller can be used for memory and ***peripheral isolation***, configuration per-core and adapted on world switches
- ***Manage without inspection.***
- Hopefully small changes to legacy OSs.

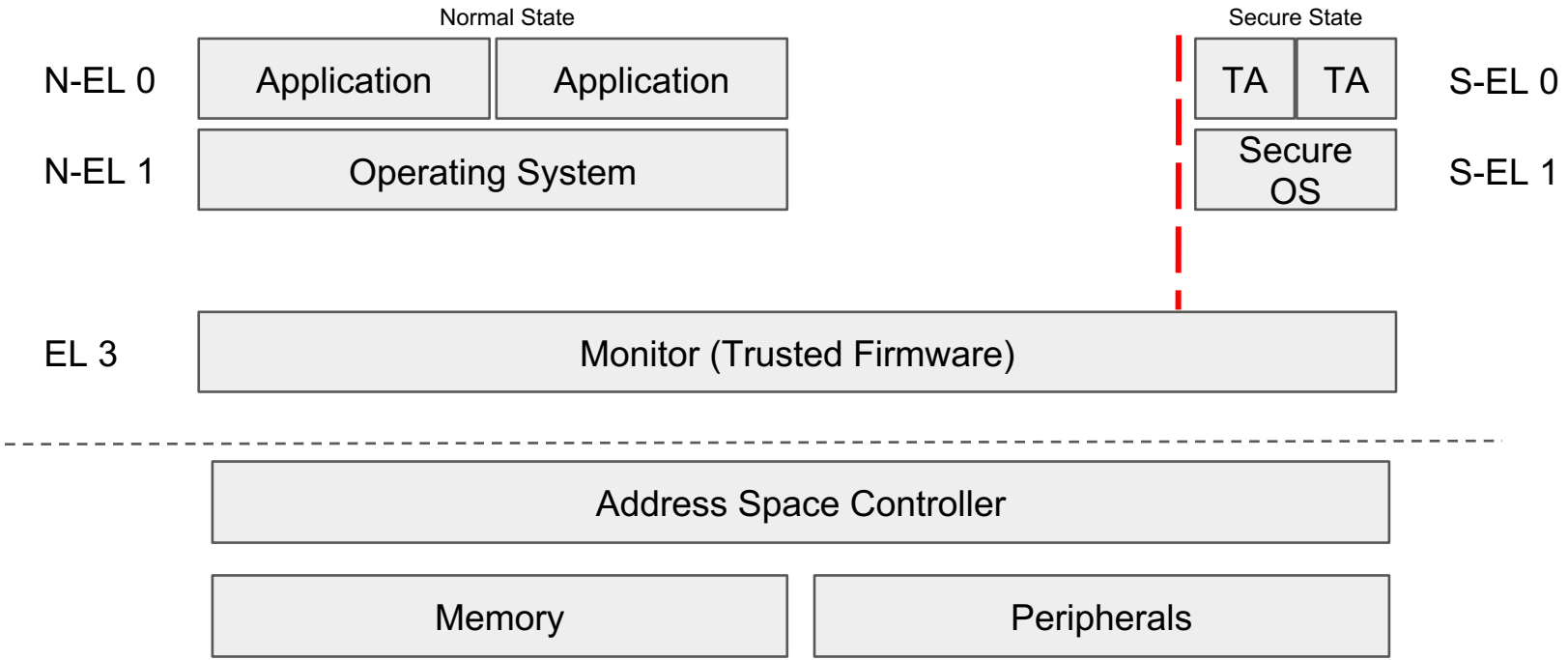
- Related Work: TrustICE [1], Sanctuary [2], Keystone [3] (RISC-V)

[1] Sun, He, et al. "Trustice: Hardware-assisted isolated computing environments on mobile devices." *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2015.
[2] Brasser, Ferdinand, et al. "SANCTUARY: ARMing TrustZone with User-space Enclaves." *NDSS*. 2019.
[3] Lee, Dayeol, et al. "Keystone: An open framework for architecting trusted execution environments." *Proceedings of the Fifteenth European Conference on Computer Systems*. 2020.

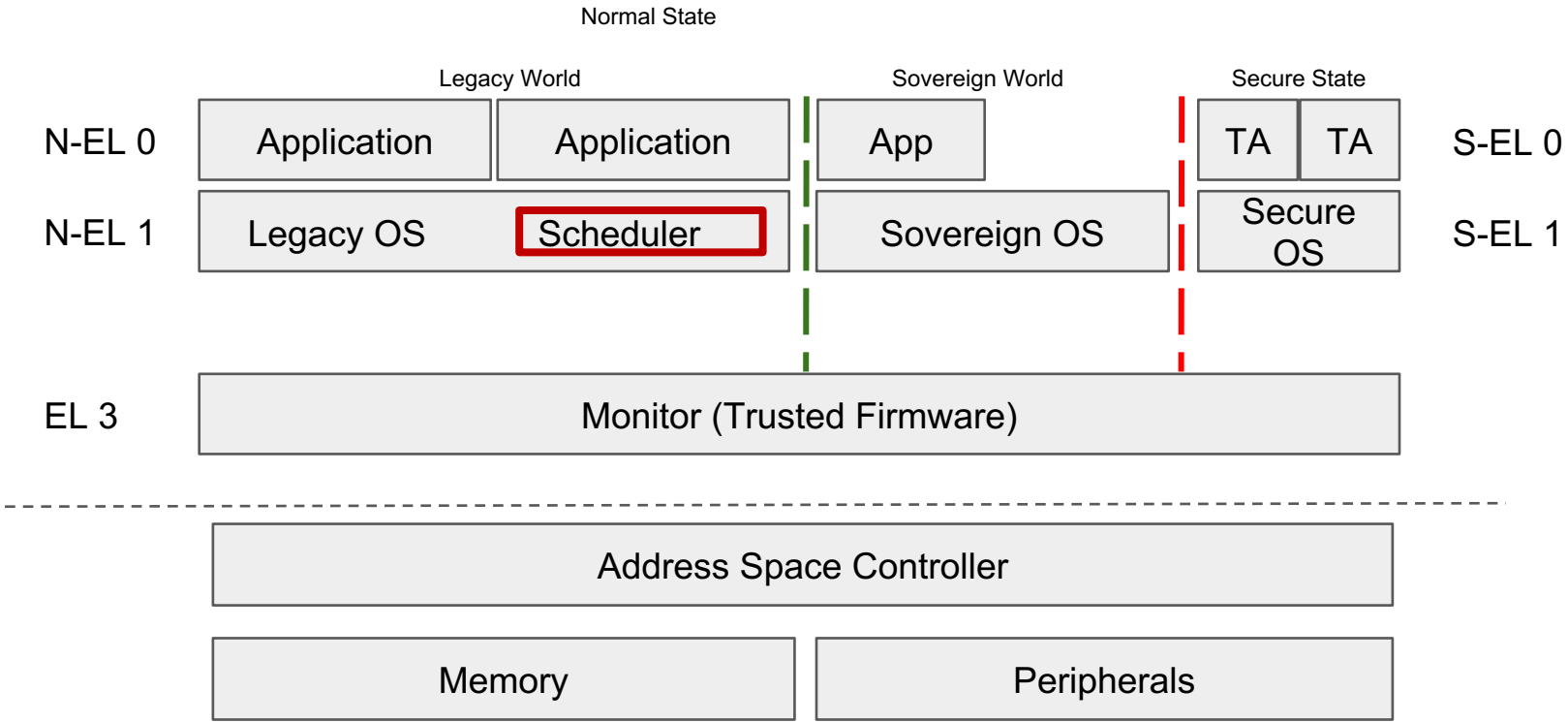
enforced in HW (NS bit)



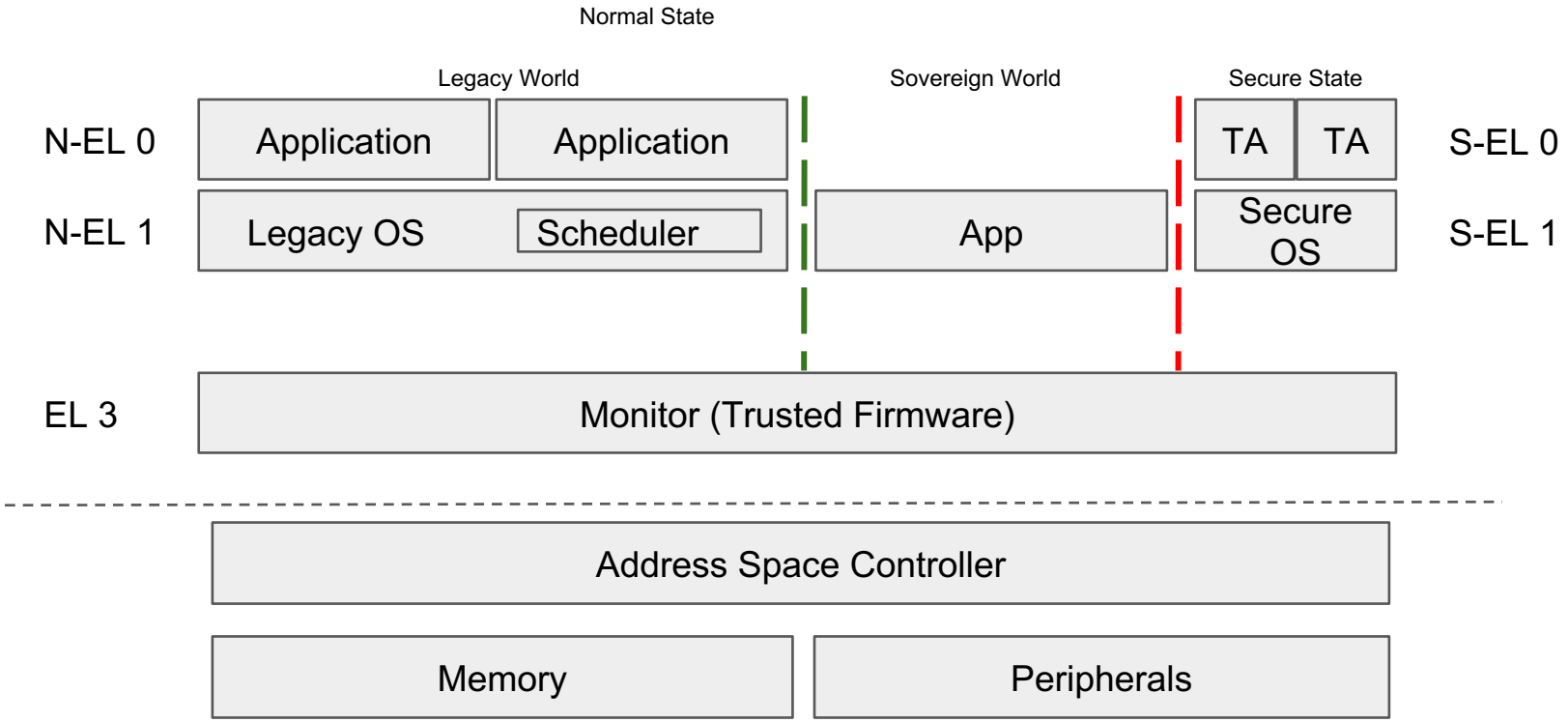
enforced in HW (NS bit)



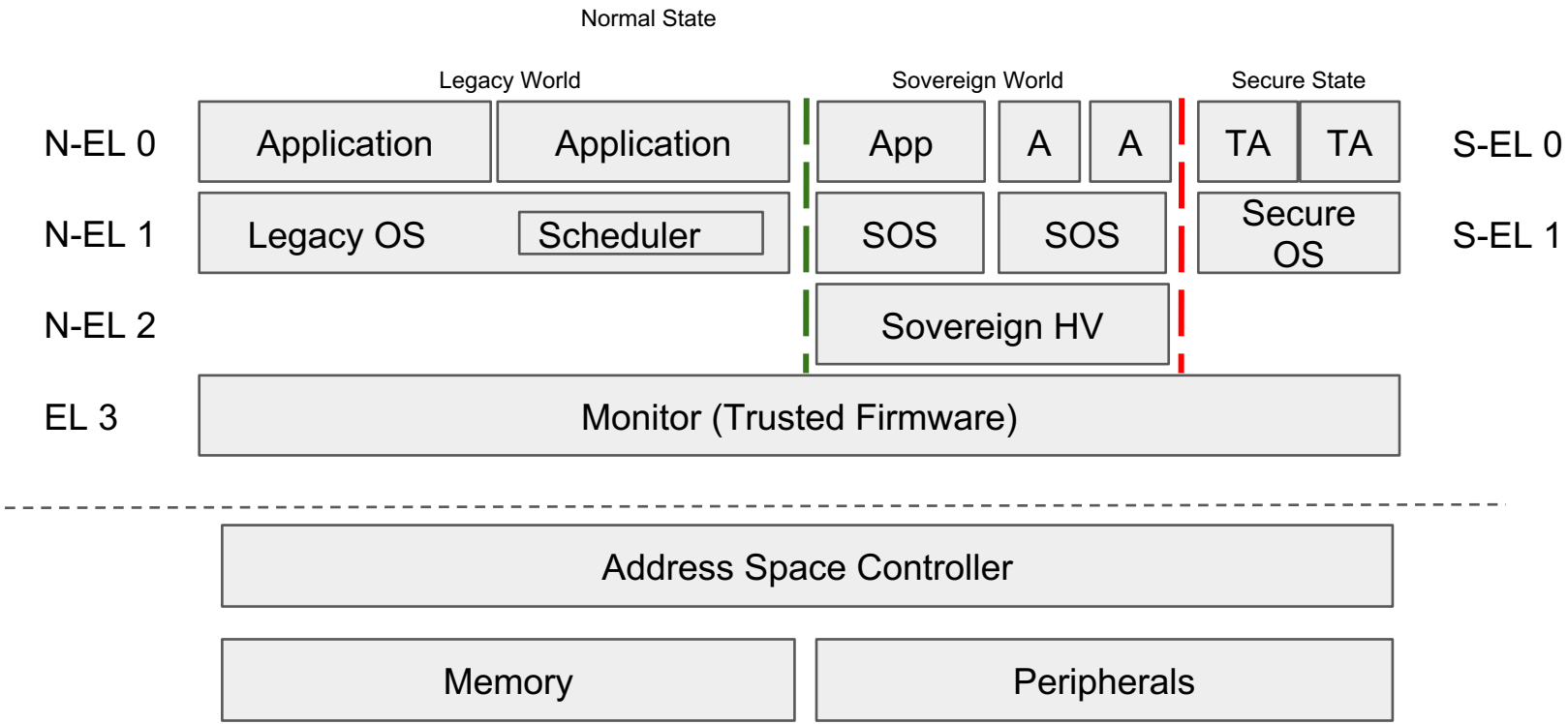
enforced in HW (NS bit) enforced by TF



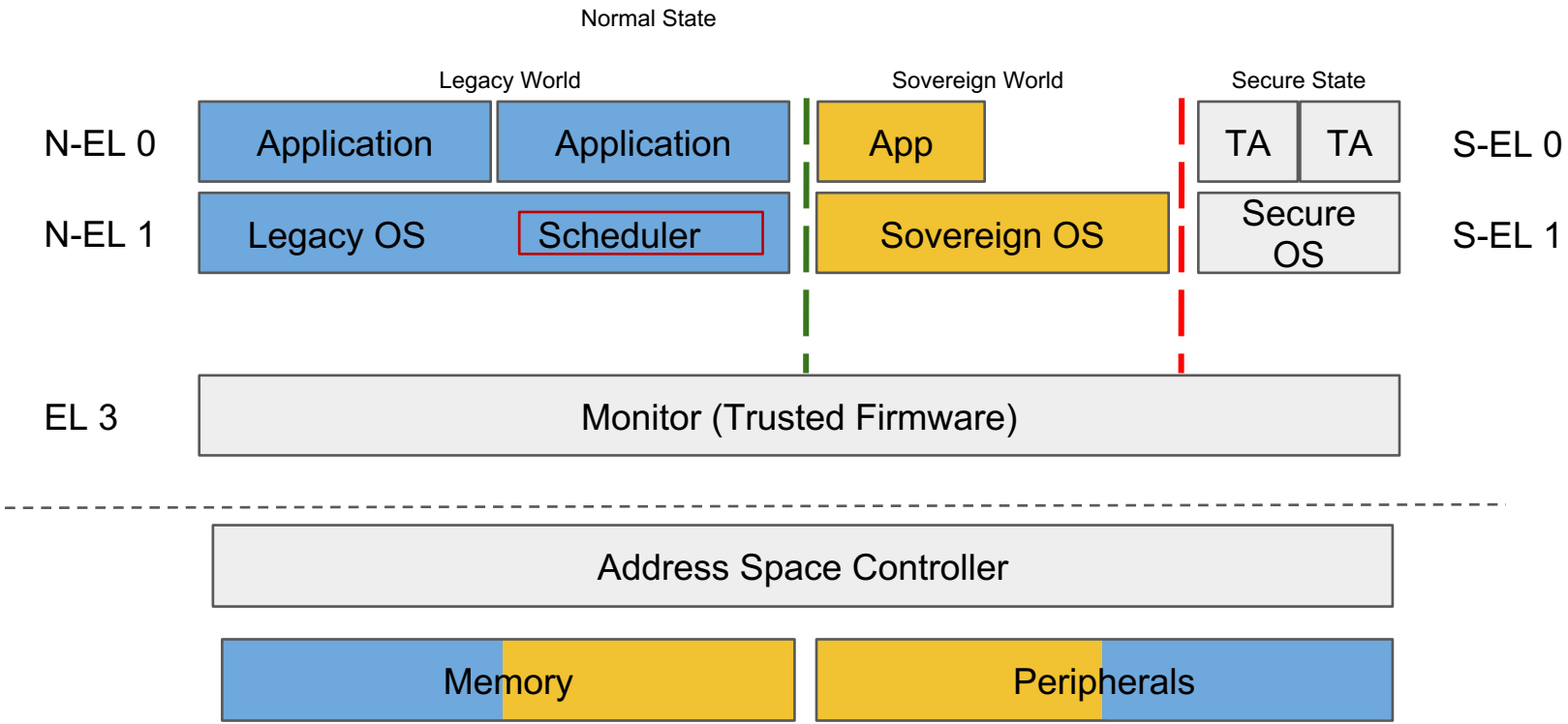
enforced in HW (NS bit) enforced by TF



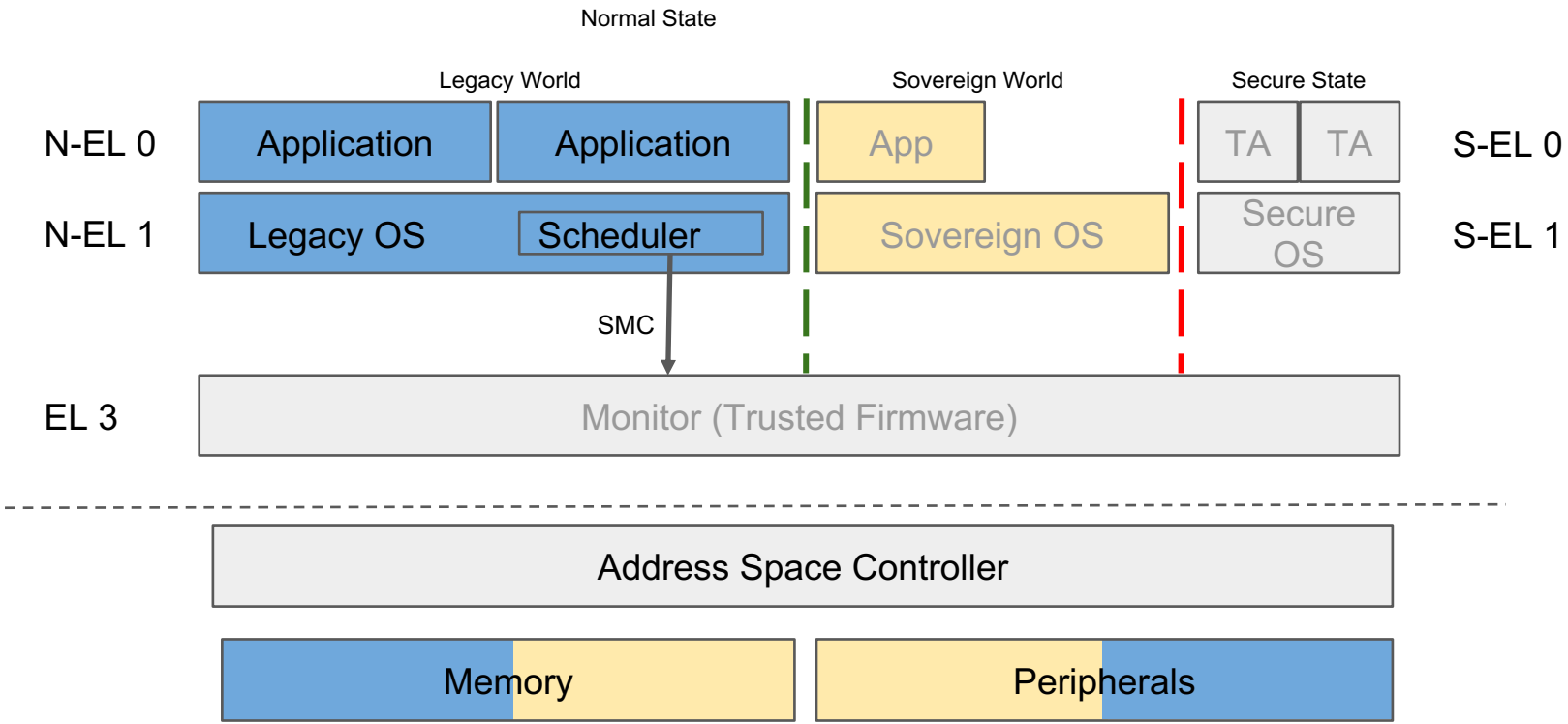
enforced in HW (NS bit) enforced by TF



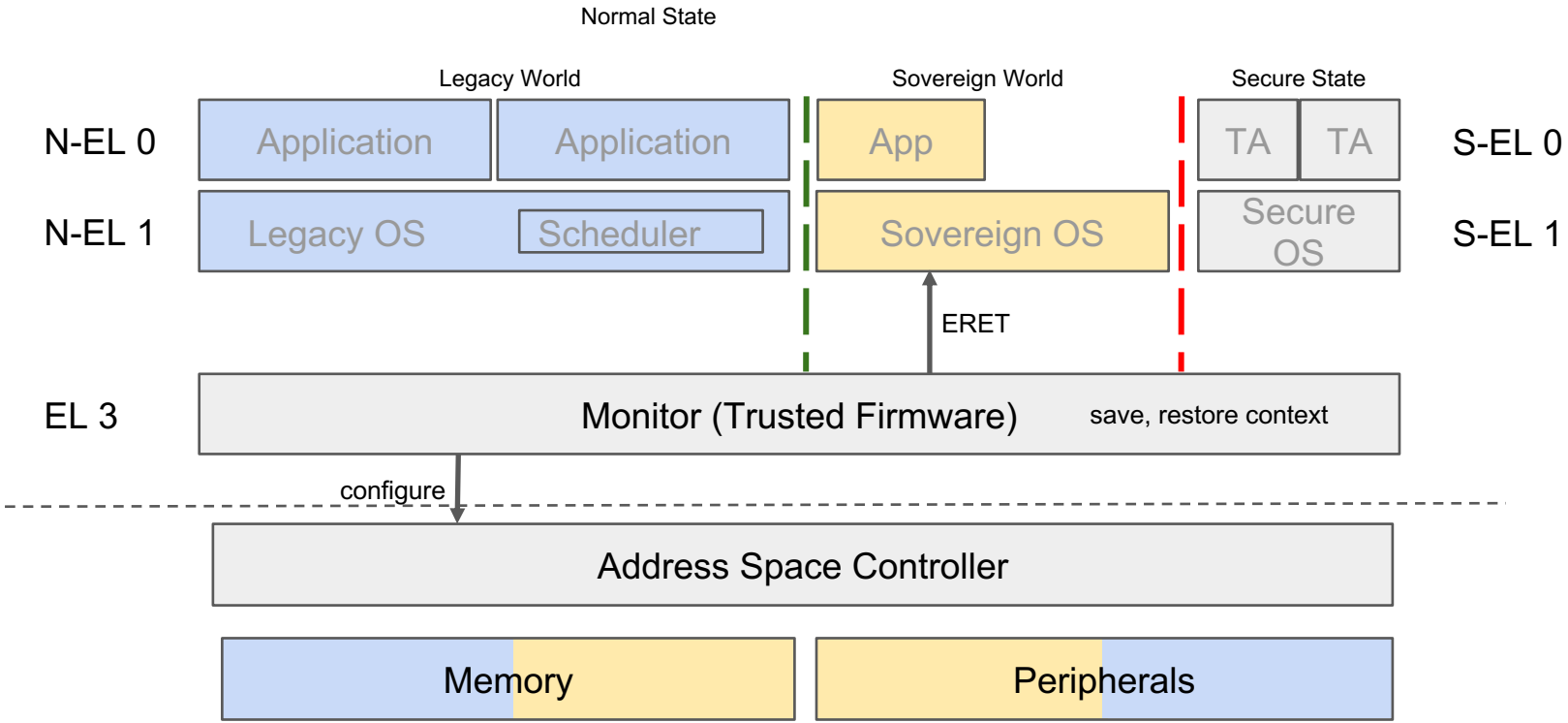
enforced in HW (NS bit) enforced by TF



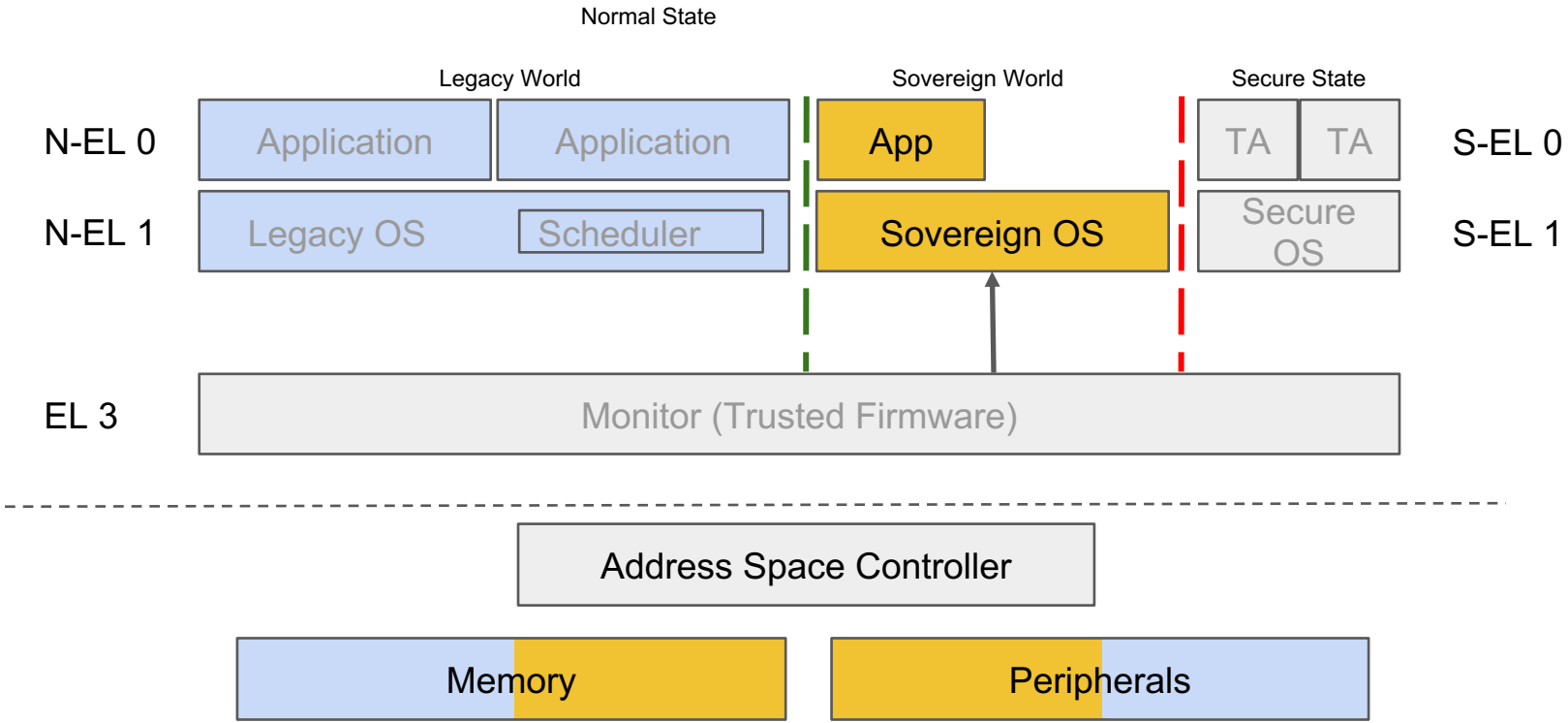
enforced in HW (NS bit) enforced by TF



enforced in HW (NS bit) enforced by TF



enforced in HW (NS bit) enforced by TF

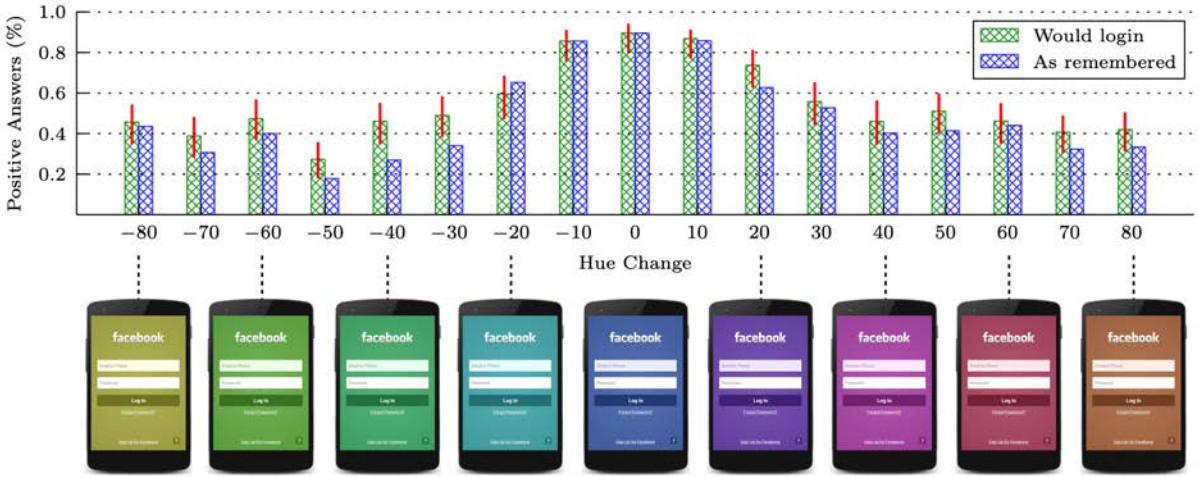


Considerations

- Common TCB: Hardware, boot stages up until Trusted Firmware, Trusted Firmware runtime, (Code run in secure state)
- Additional TCB for legacy OS: additions to Trusted Firmware
- Legacy OS is responsible for scheduling and can DoS the sovereign world
- Timer interrupts need to be handled by legacy world, even when sovereign world is running
- User has full control over hardware: can configure Address Space Controller and install sovereign world code

UI issues

- User needs to be aware of the Sovereign vs Legacy world.
- Indicators / hardware switches
- ‘Attestation’ to the user



Current Status

- Implementation for aarch64 on qemu, until now:
  - “sovereign” state in Trusted Firmware
  - SMC handler in Trusted Firmware that switches context
  - Sovereign binary loaded statically in memory
  - Sovereign World preemptable
  - …

Conclusion

- A balance needs to be found between the services provided by the legacy OS companies and the needs of users to control their phones.
- ‘Manage without inspection’ is a key functionality.
- *Why would legacy OS companies support this?*

arxiv.org/abs/2102.02743

Sovereign Smartphone:
To Enjoy Freedom We Have to Control Our Phones

Friederike Groschupp Moritz Schneider Ivan Puddu Shweta Shinde Srdjan Capkun
ETH Zurich