Sanctuary: ARMing TrustZone with User-space Enclaves

Ferdinand Brasser, David Gens, Patrick Jauernig, Ahmad-Reza Sadeghi, Emmanuel Stapf
Technische Universität Darmstadt

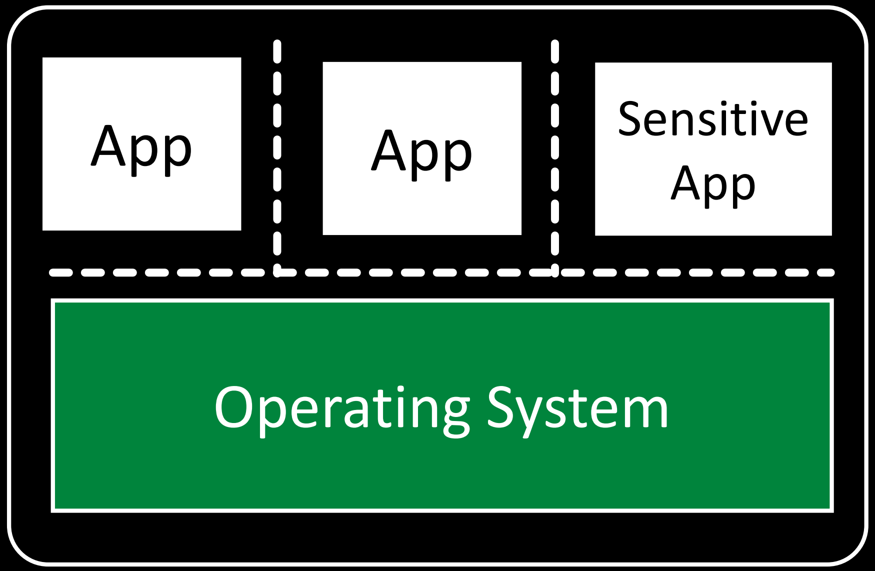
intel | Research Institute for Collaborative Autonomous and Resilient Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Research Problem

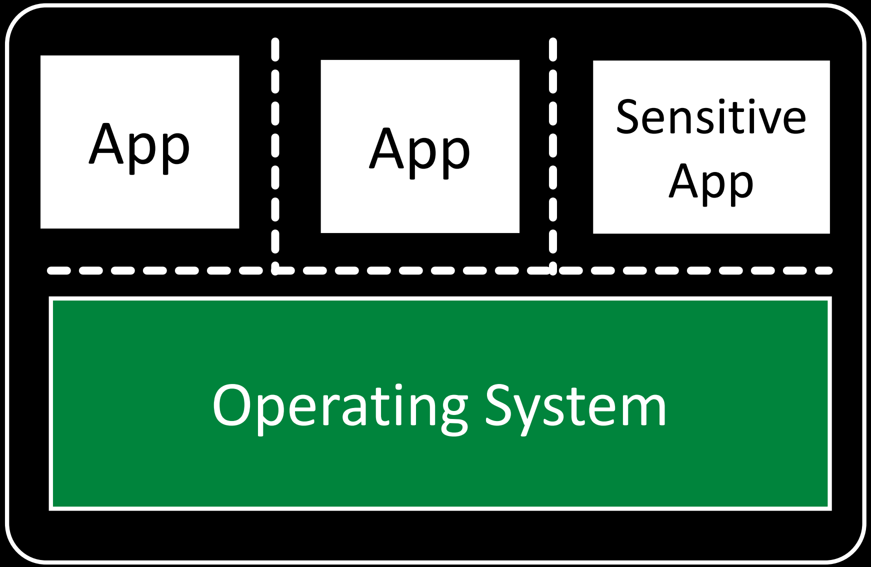
How can we construct an ecosystem of mutually distrusted enclaves on mobile devices?

Commodity OS



Trusted Untrusted Compromised

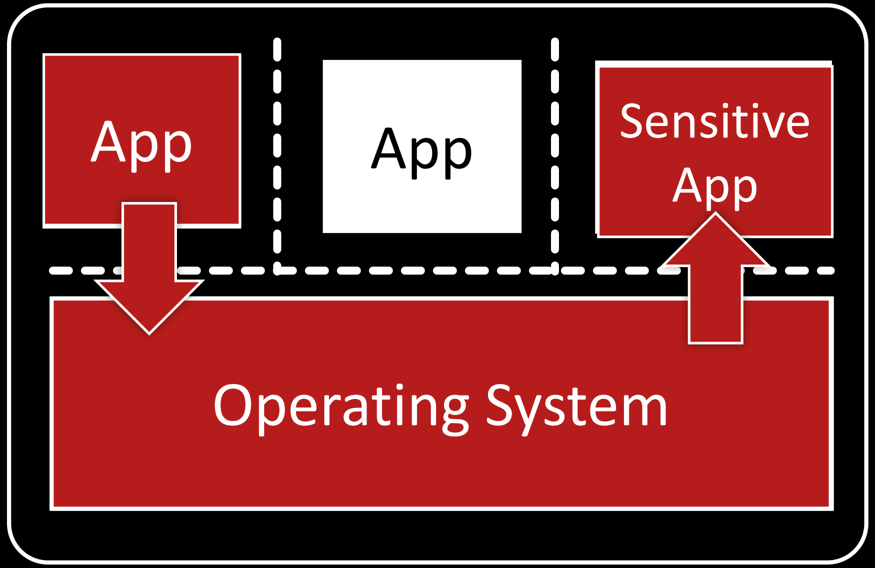
Commodity OS



- OS large attack surface
- Insufficient sensitive app protection

Trusted Untrusted Compromised

Commodity OS

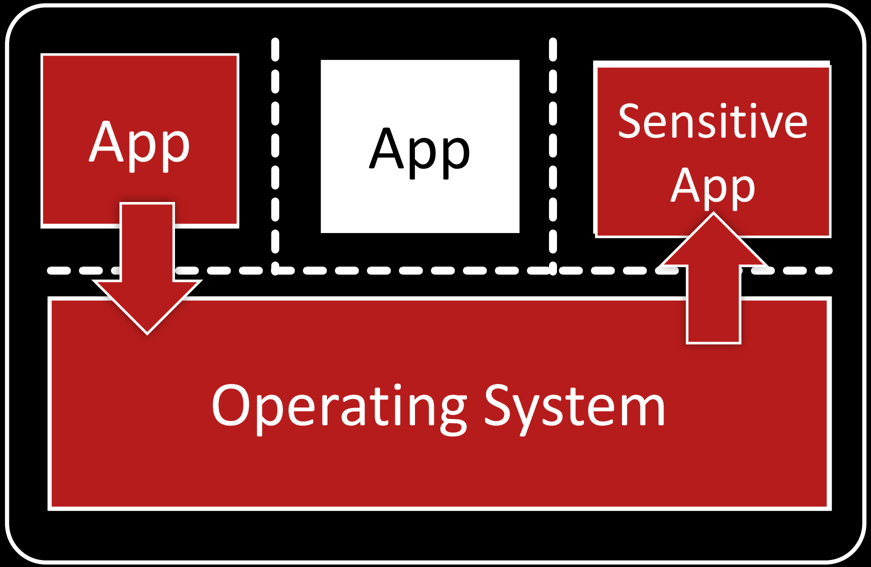


- OS large attack surface
- Insufficient sensitive app protection

Trusted Untrusted Compromised

Commodity OS

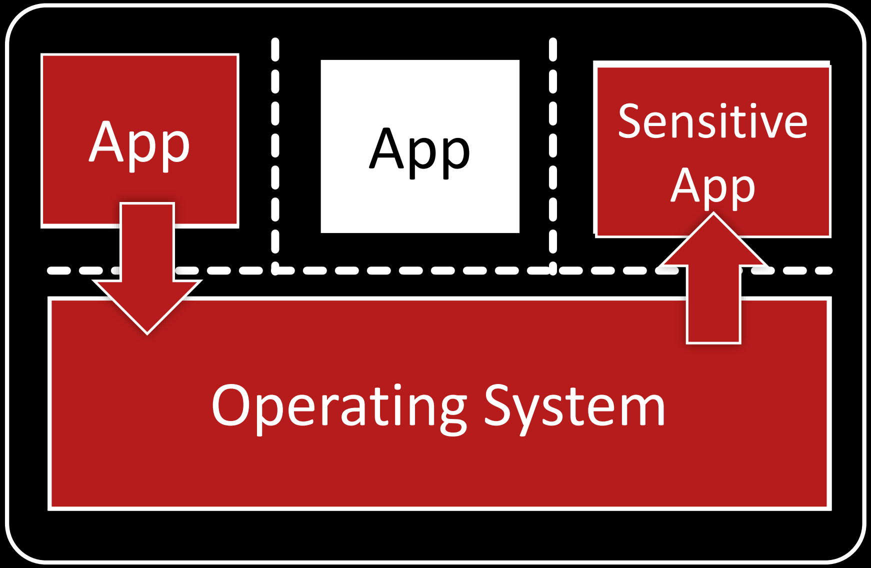
ARM TrustZone



- OS large attack surface
- Insufficient sensitive app protection

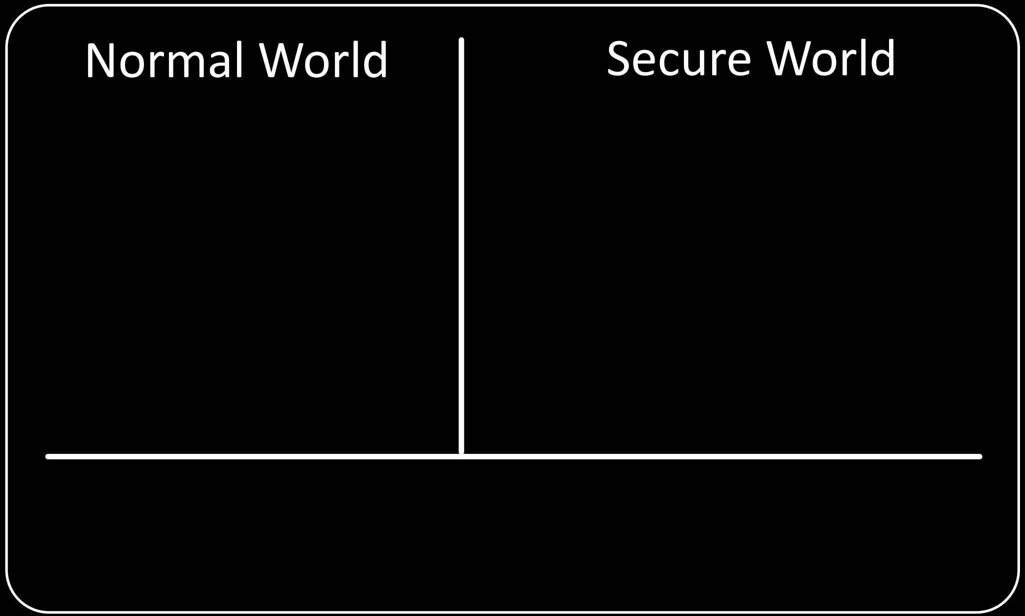
Trusted Untrusted Compromised

Commodity OS



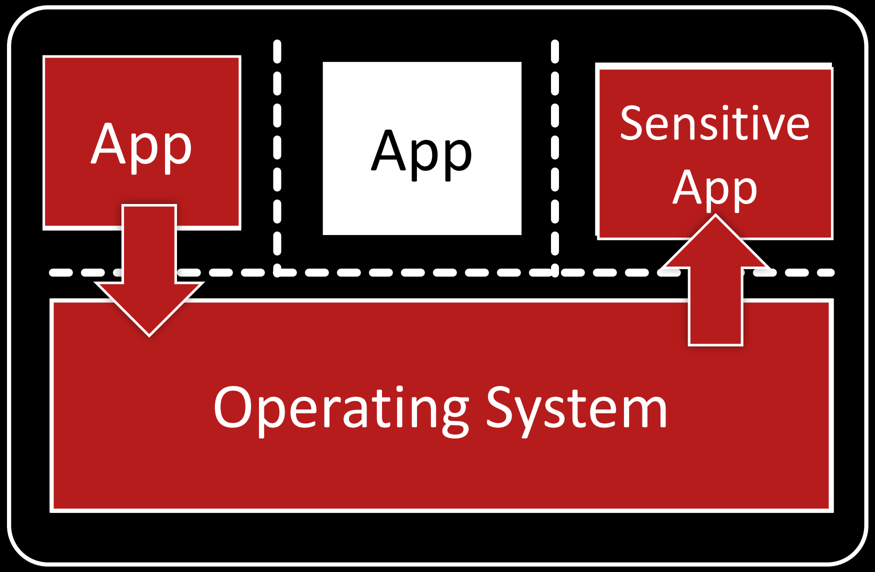
- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone



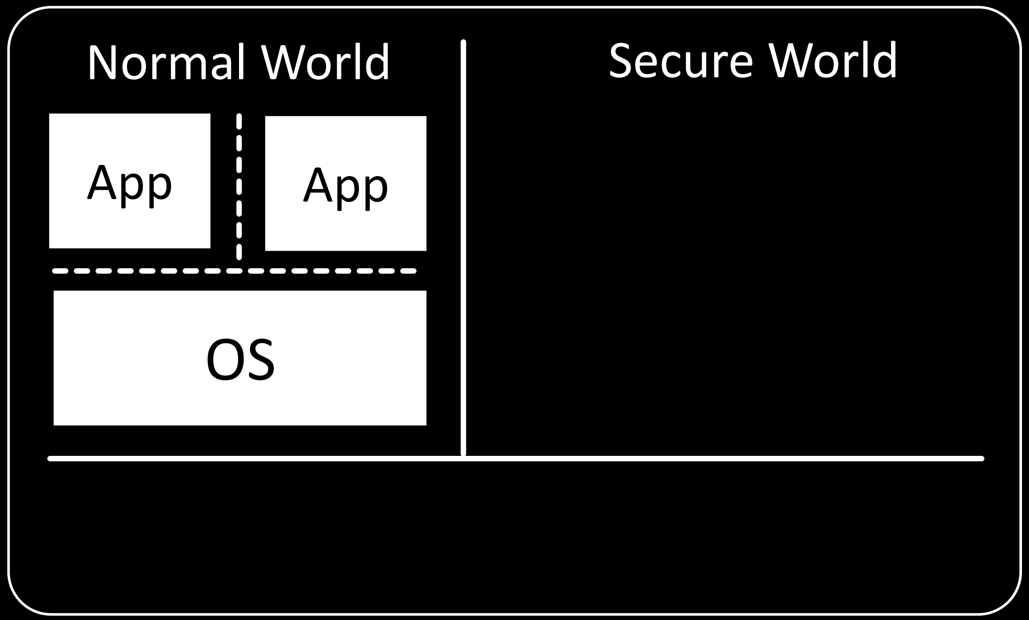
Trusted Untrusted Compromised

Commodity OS



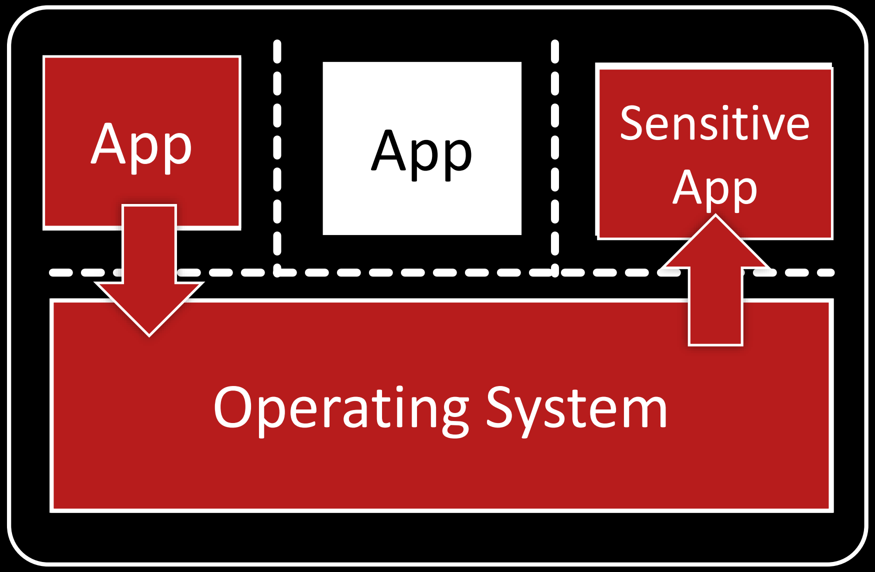
- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone



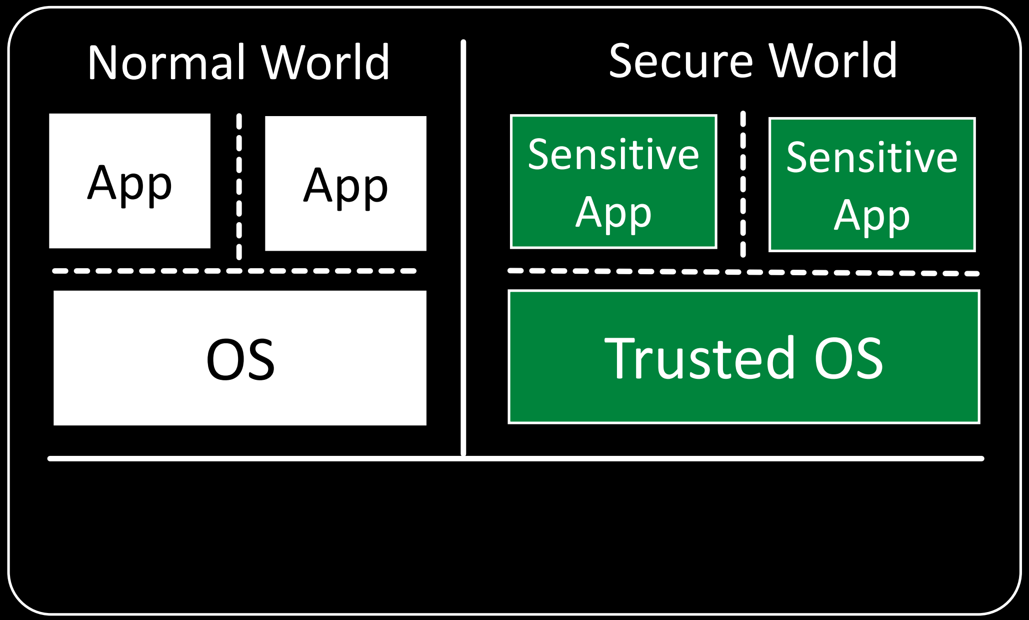
Trusted Untrusted Compromised

Commodity OS



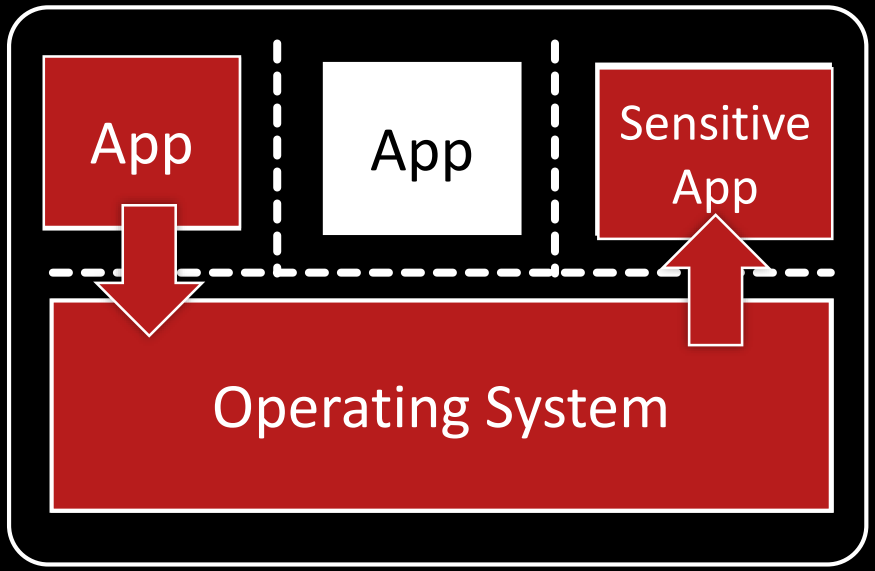
- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone



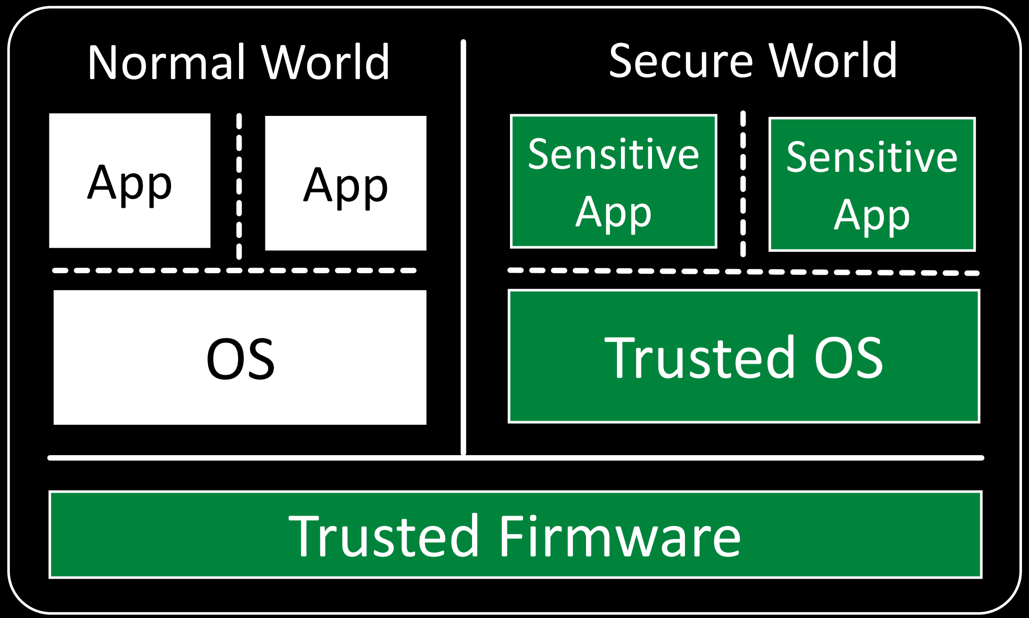
Trusted Untrusted Compromised

Commodity OS



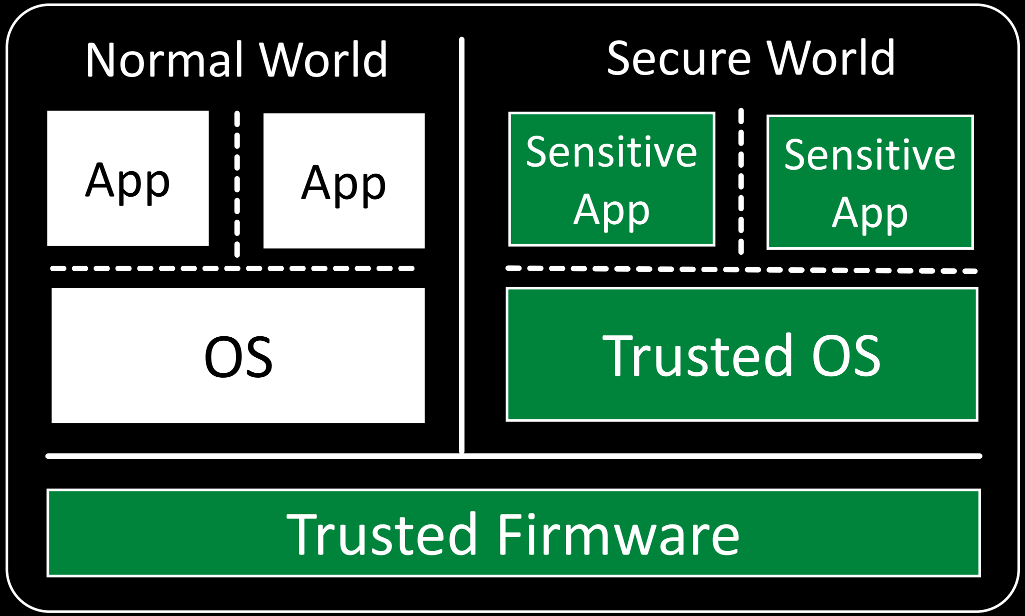
- OS large attack surface
- Insufficient sensitive app protection

ARM TrustZone



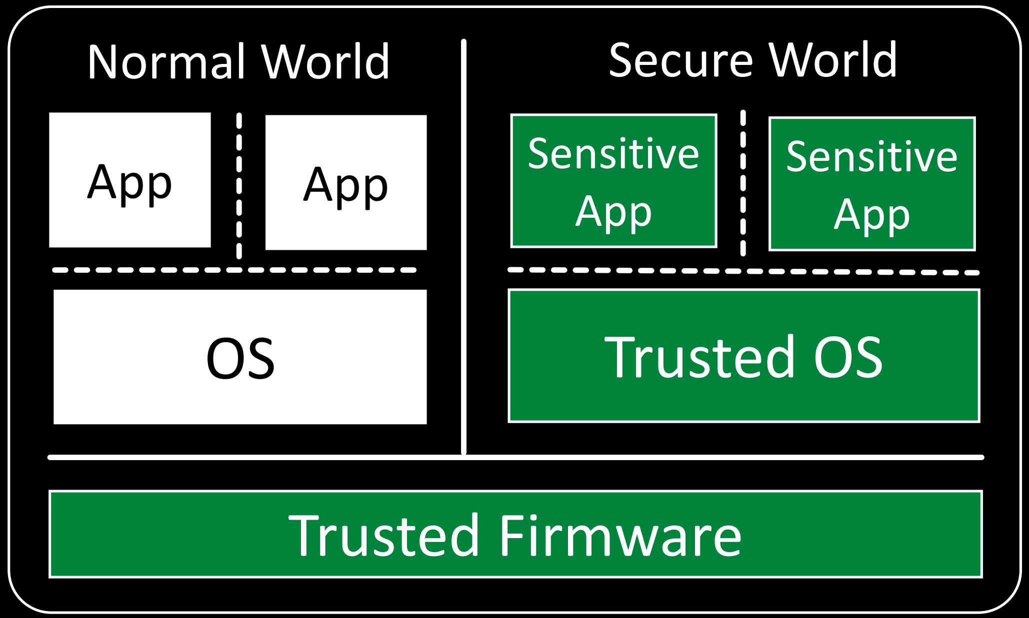
Trusted Untrusted Compromised

ARM TrustZone
The Solution?



Trusted Untrusted Compromised

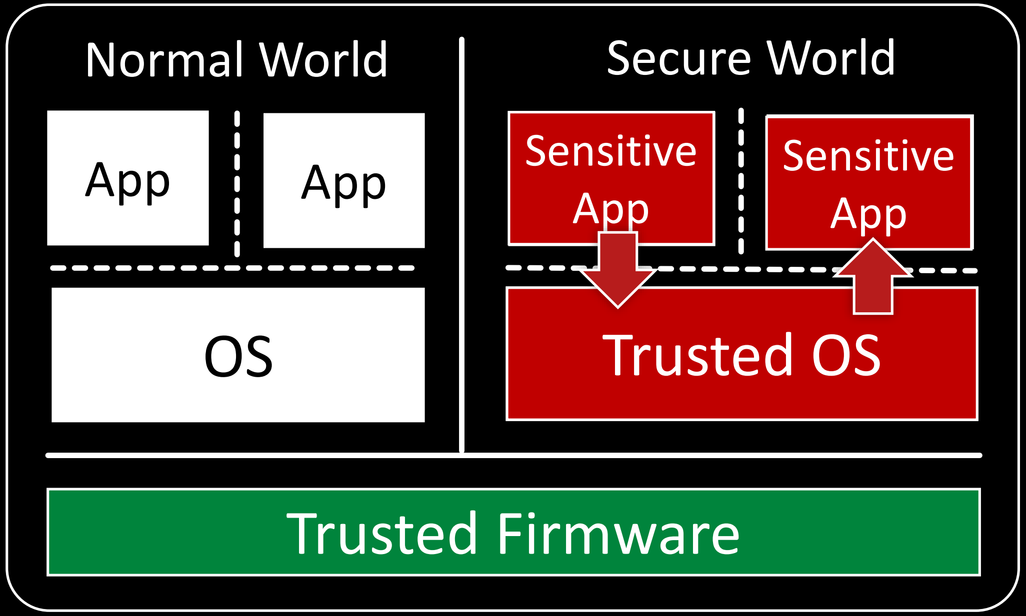
ARM TrustZone
The Solution?



- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Trusted Untrusted Compromised

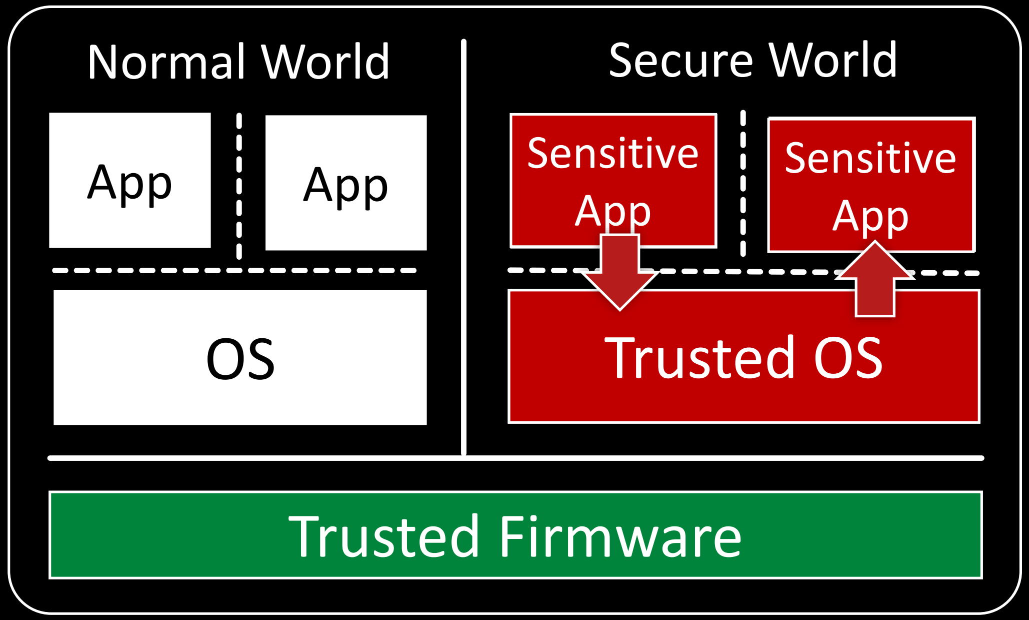
ARM TrustZone
The Solution?



- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

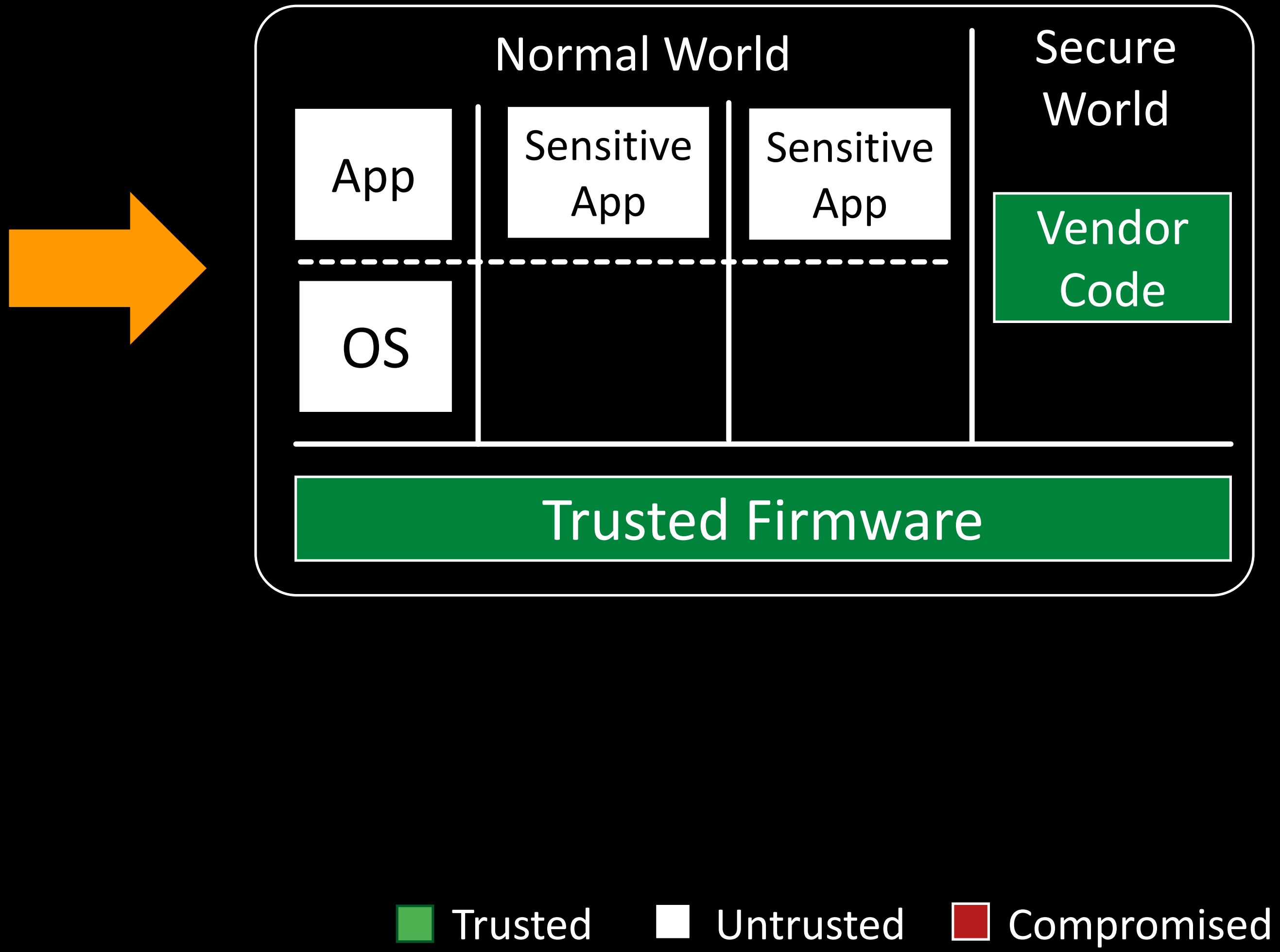
Trusted Untrusted Compromised

ARM TrustZone
The Solution?

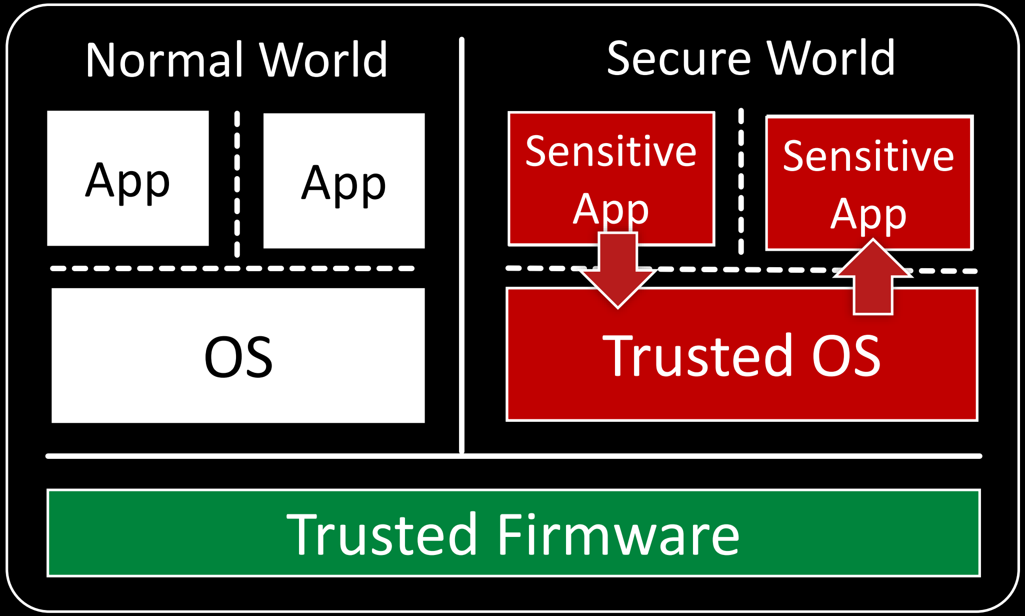


- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Sanctuary
Multiple Security Domains

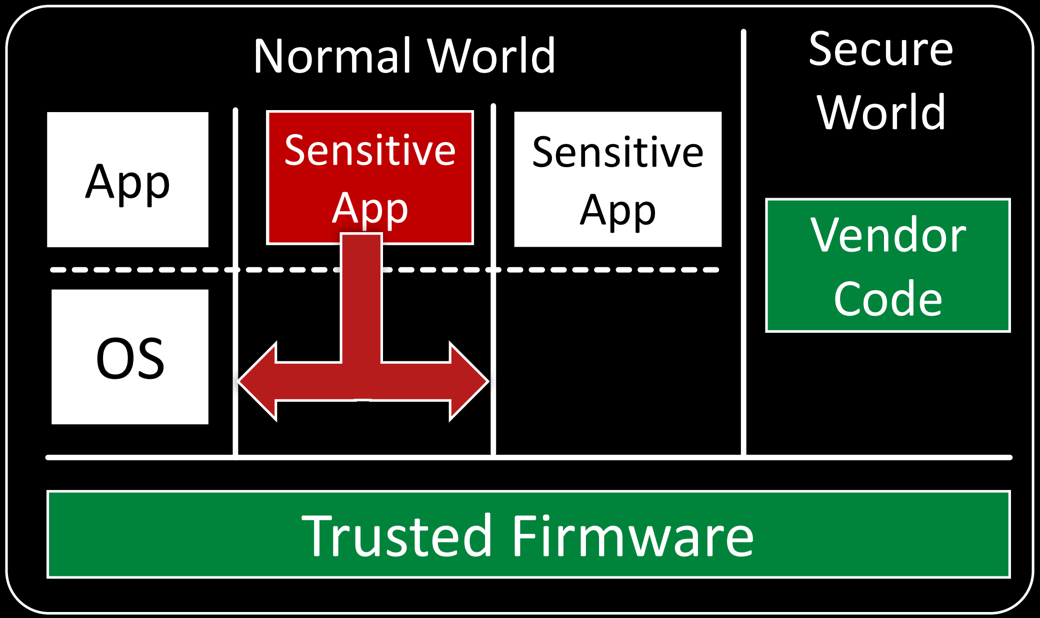


ARM TrustZone
The Solution?



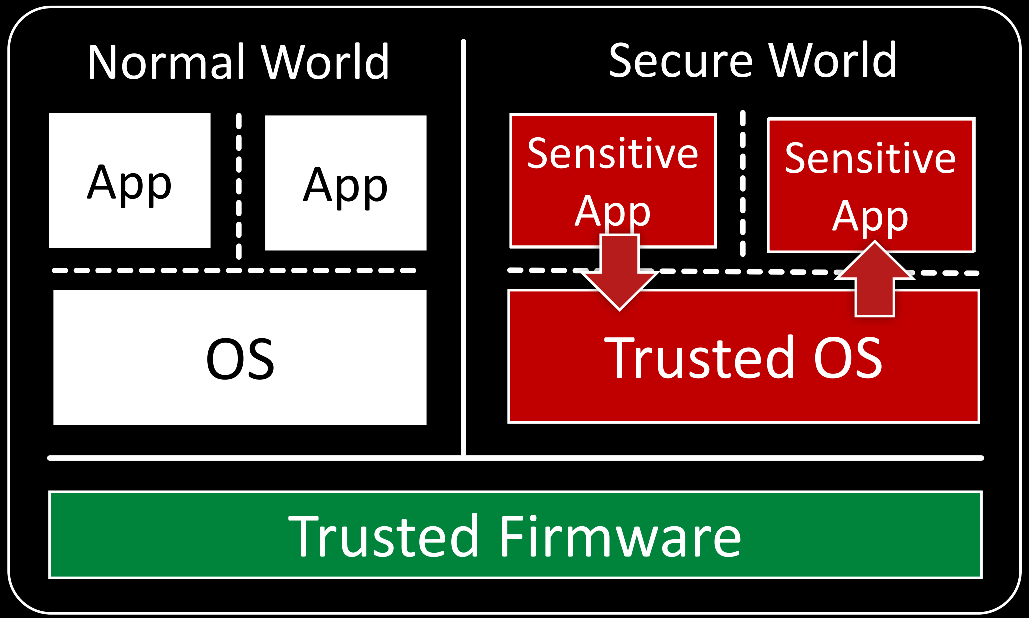
- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Sanctuary
Multiple Security Domains



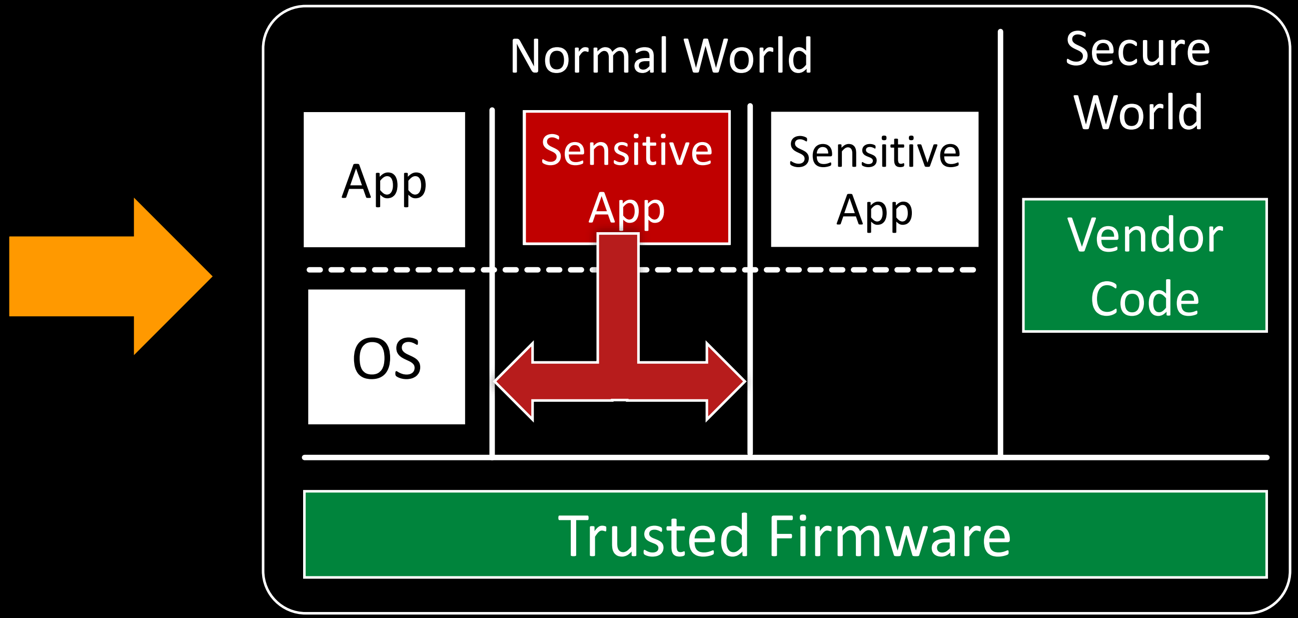
Trusted Untrusted Compromised

ARM TrustZone
The Solution?



- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

Sanctuary
Multiple Security Domains



- Sensitive apps can remain untrusted
- Make TrustZone protection available to third parties

Trusted Untrusted Compromised

ARM TrustZone
The Solution?

Sanctuary
Multiple Security Domains

Challenges

- No new hardware components
- No replacement of existing code bases (e.g. Trusted OS)
- Only minor impact on commodity OS

Norma

App

Trusted Firmware

Secure World

Vendor Code

Trusted Firmware

- Trusted OS still large attack surface
- High costs to protect apps with TrustZone
- Main problem: Single security domain

- Sensitive apps can remain untrusted
- Make TrustZone protection available to third parties

Trusted · Untrusted · Compromised

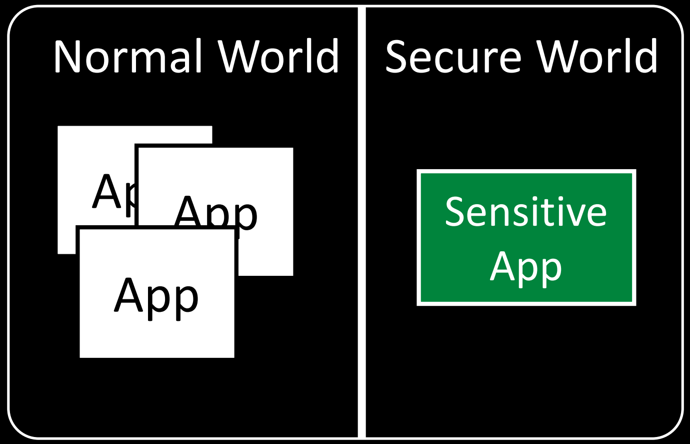
Adversary Model

- Adversary can
  - compromise normal-world software at all privilege levels
  - run malicious sensitive apps
- Adversary cannot
  - compromise TrustZone (trust anchor)
  - perform physical attacks

Related Work

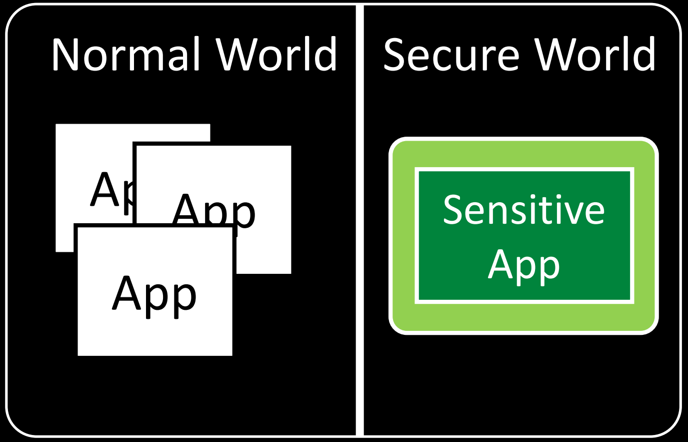
Existing Research Proposals

Software Approaches
Utilizing TrustZone



Existing Research Proposals

Software Approaches
Utilizing TrustZone



Improve isolation of sensitive apps without add. HW features

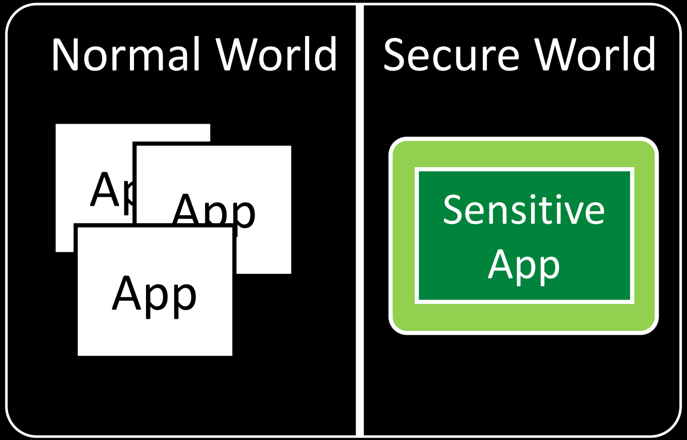
among others,

[Ferraiuolo et al., SOSP 2017]

[Sun et al., DSN 2015]

Existing Research Proposals

Software Approaches Utilizing TrustZone



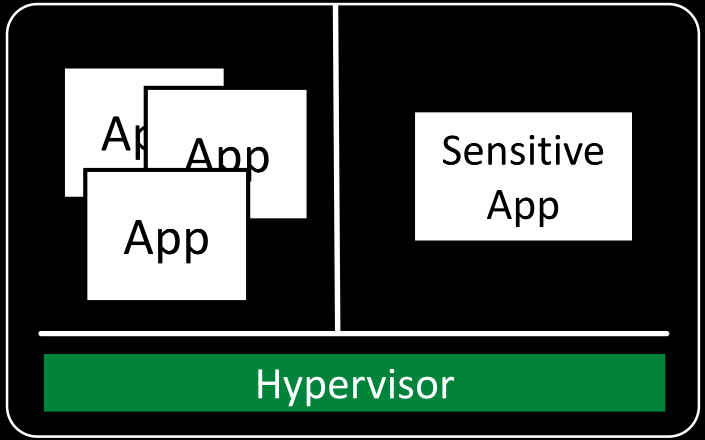
Improve isolation of sensitive apps without add. HW features

among others,

[Ferraiuolo et al., SOSP 2017]

[Sun et al., DSN 2015]

Hypervisor-based Isolation Utilizing TrustZone



Use virtualization to protect sensitive apps.
Use TrustZone as trust anchor

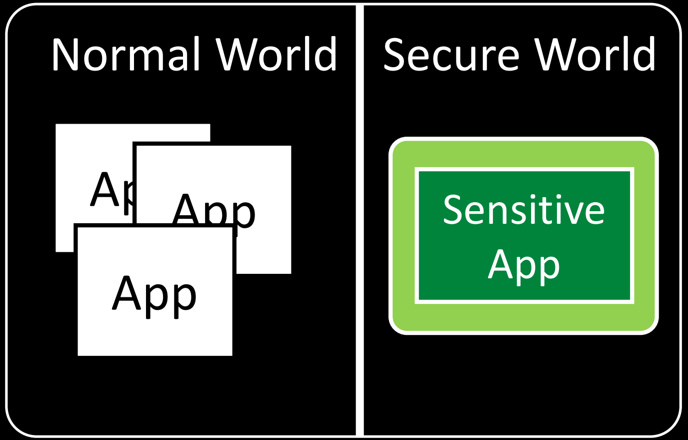
among others,

[Hua et al., USENIX 2017]

[Cho et al., USENIX 2016]

Existing Research Proposals

Software Approaches Utilizing TrustZone



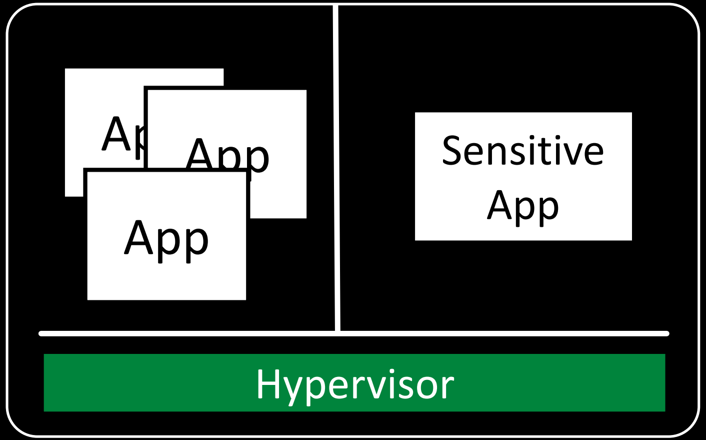
Improve isolation of sensitive apps without add. HW features

among others,

[Ferraiuolo et al., SOSP 2017]

[Sun et al., DSN 2015]

Hypervisor-based Isolation Utilizing TrustZone



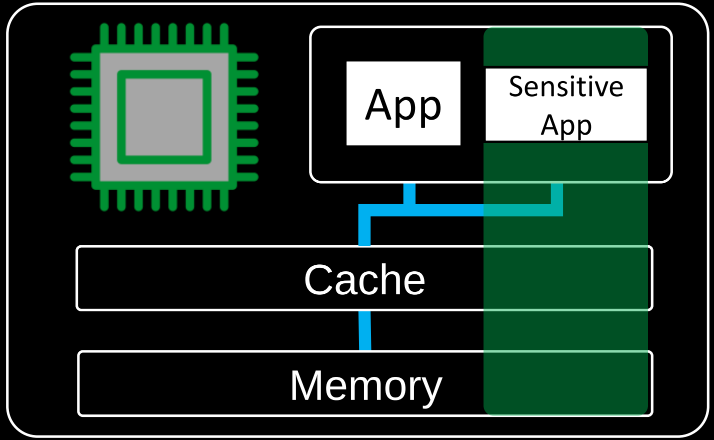
Use virtualization to protect sensitive apps.
Use TrustZone as trust anchor

among others,

[Hua et al., USENIX 2017]

[Cho et al., USENIX 2016]

Architectural Modifications



Overcome TrustZone's shortcomings with own HW

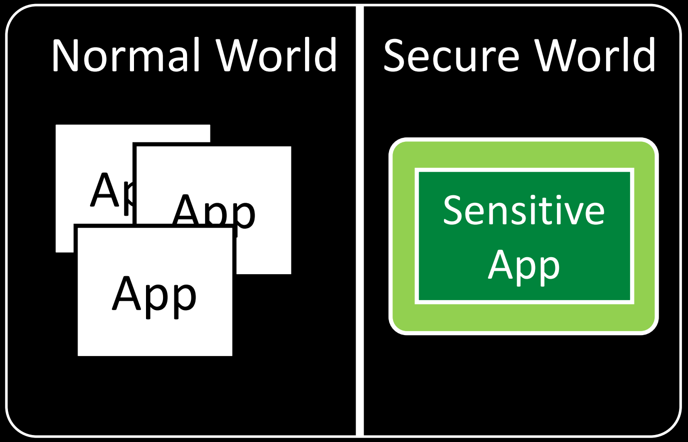
among others,

[Costan et al., USENIX 2016]

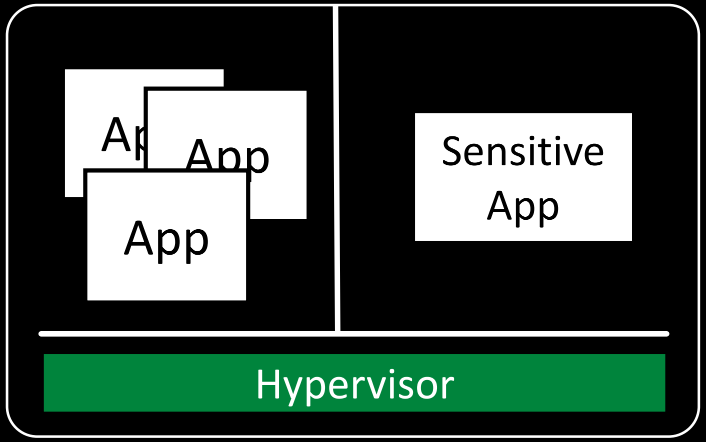
[Evtyushkin et al., MICRO 2014]

Existing Research Proposals - Problems

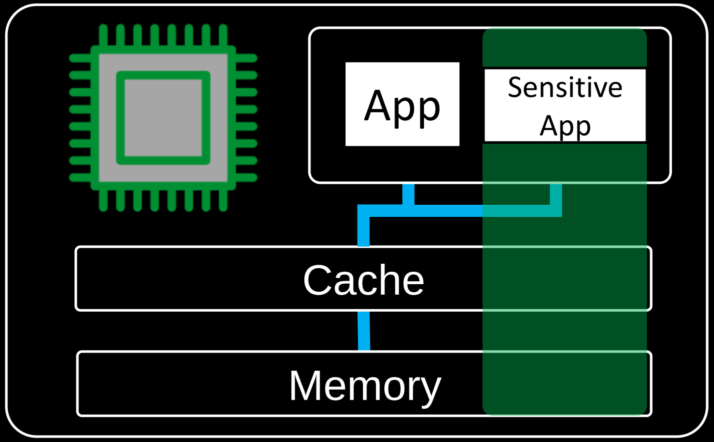
Software Approaches Utilizing TrustZone



Hypervisor-based Isolation Utilizing TrustZone



Architectural Modifications



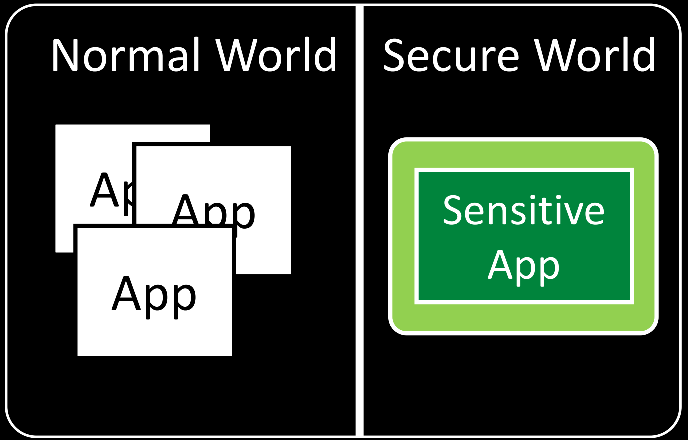
Replaces existing code base

Hypervisor blocked

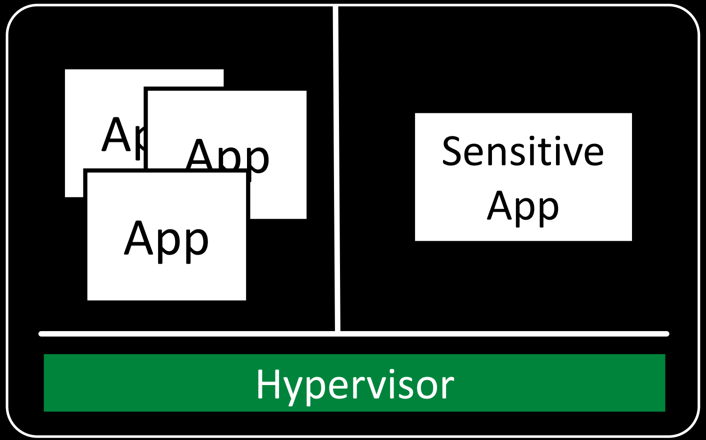
New hardware design

Existing Research Proposals - Problems

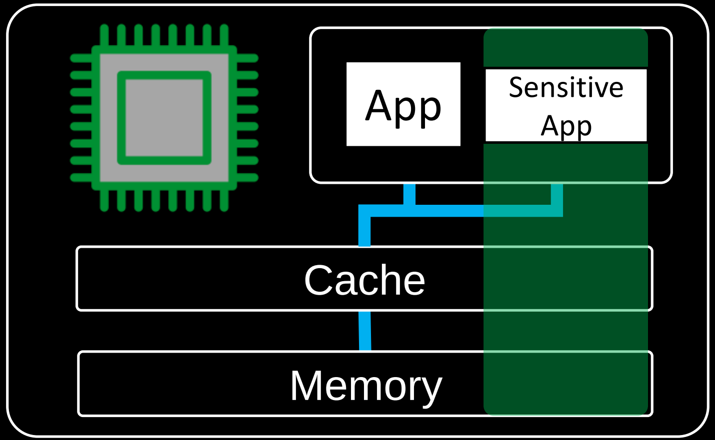
Software Approaches Utilizing TrustZone



Hypervisor-based Isolation Utilizing TrustZone



Architectural Modifications



Replaces existing code base

OS needs to be suspended

No support for multi-core environments

Hypervisor blocked

Slows down commodity OS

Additional TCB component

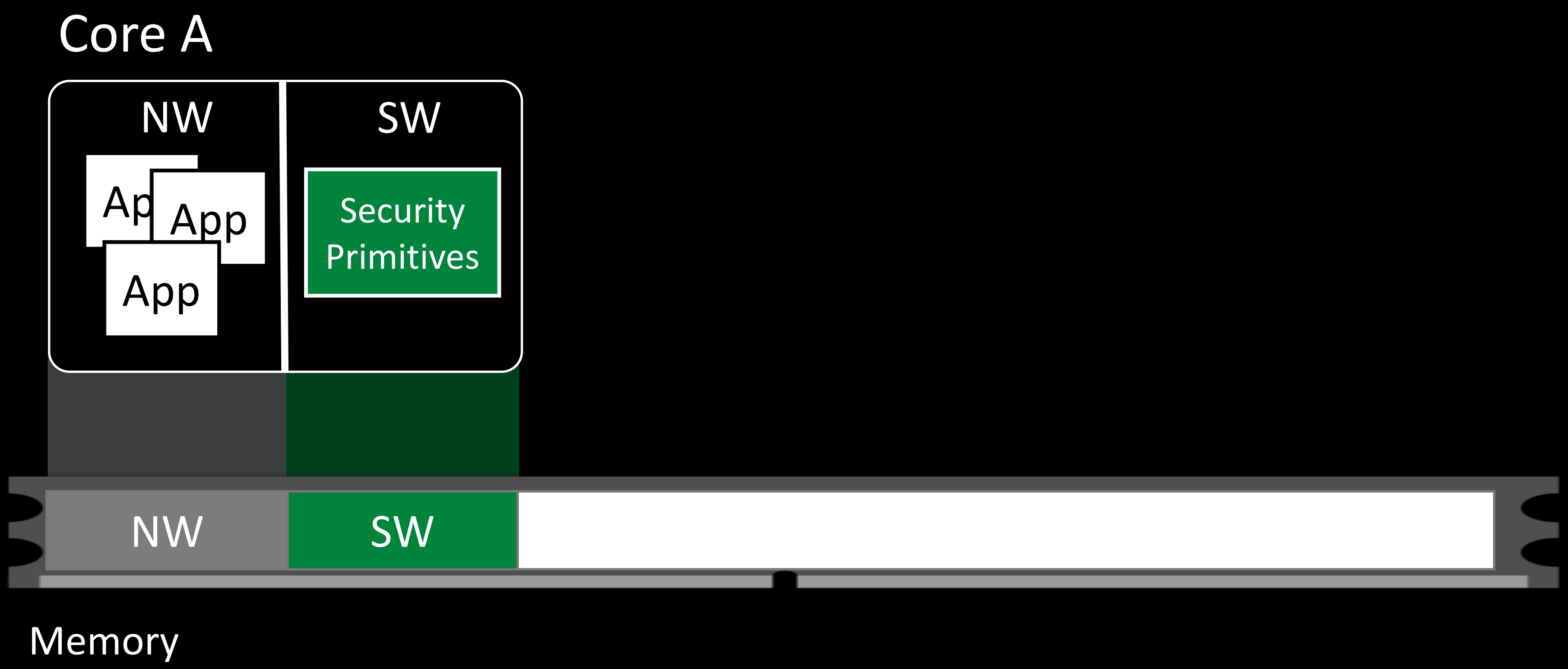
Additional HW for DMA access control

New hardware design

High deployment costs

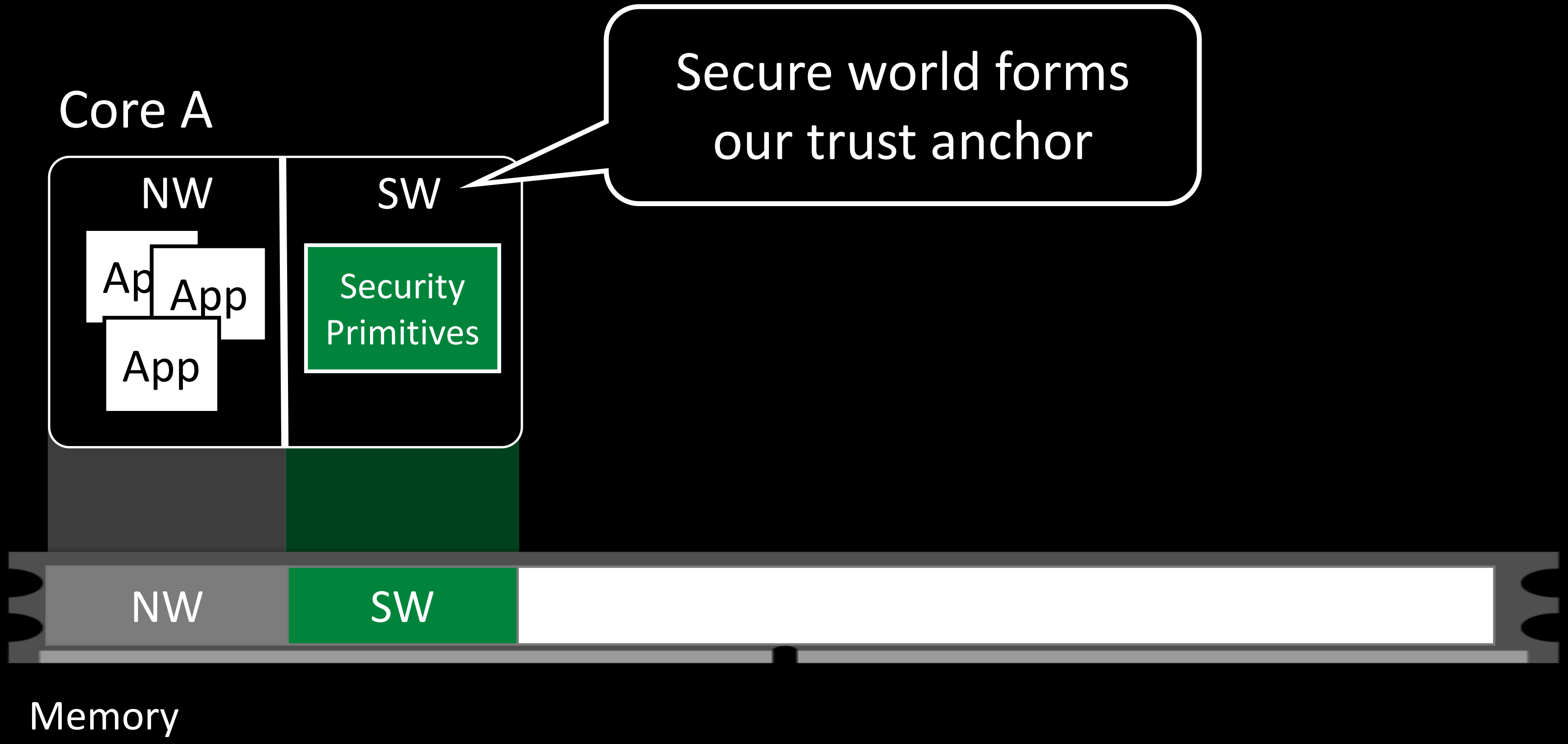
Low adoption by industry

Sanctuary provides Multi-Domain Isolation



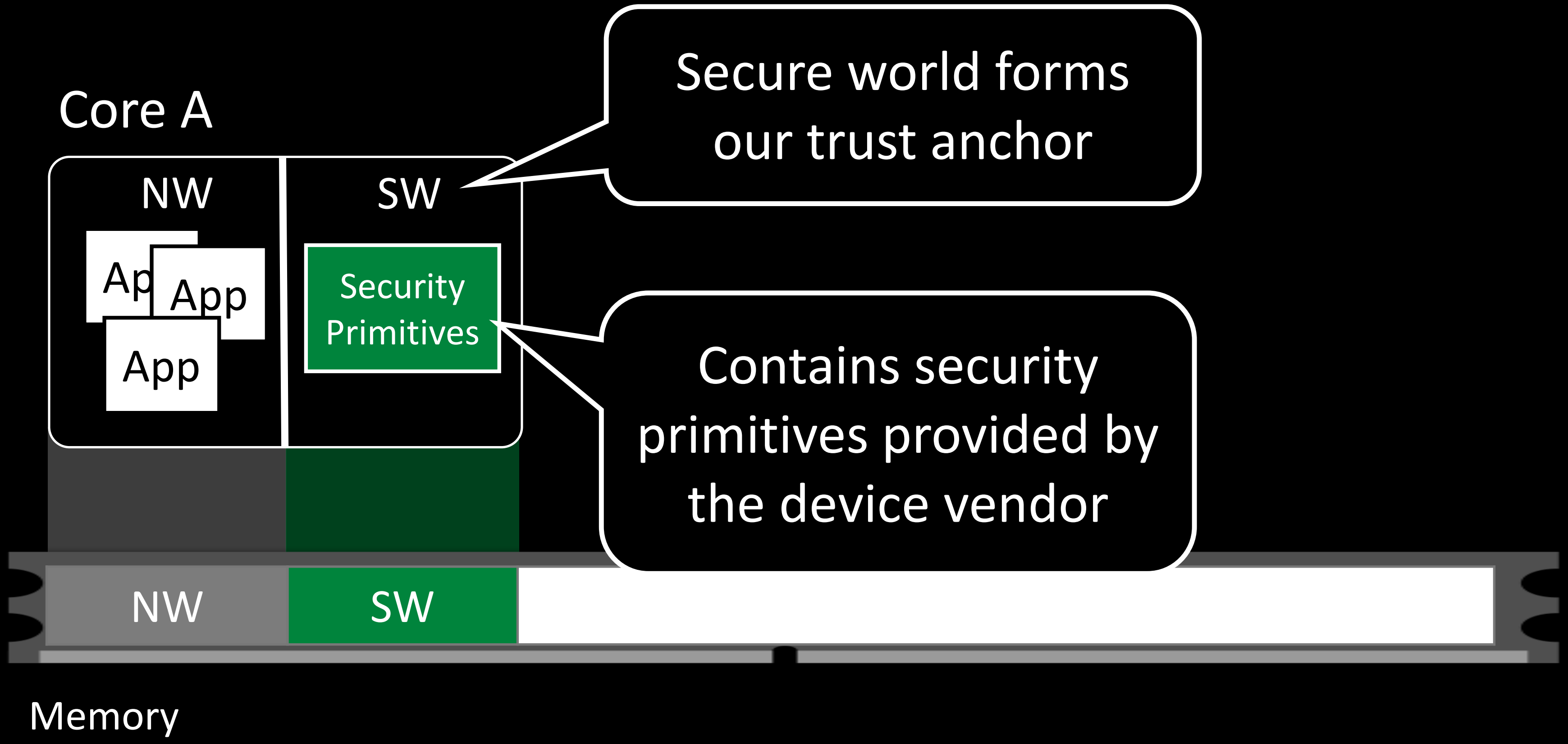
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



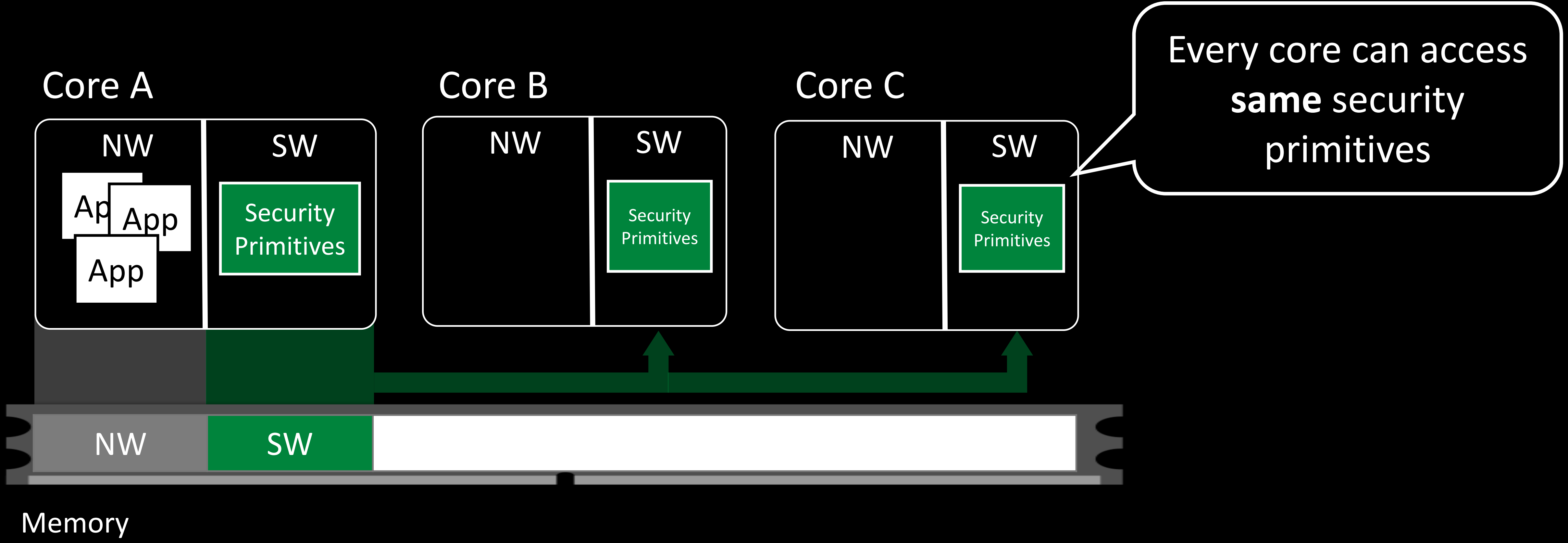
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



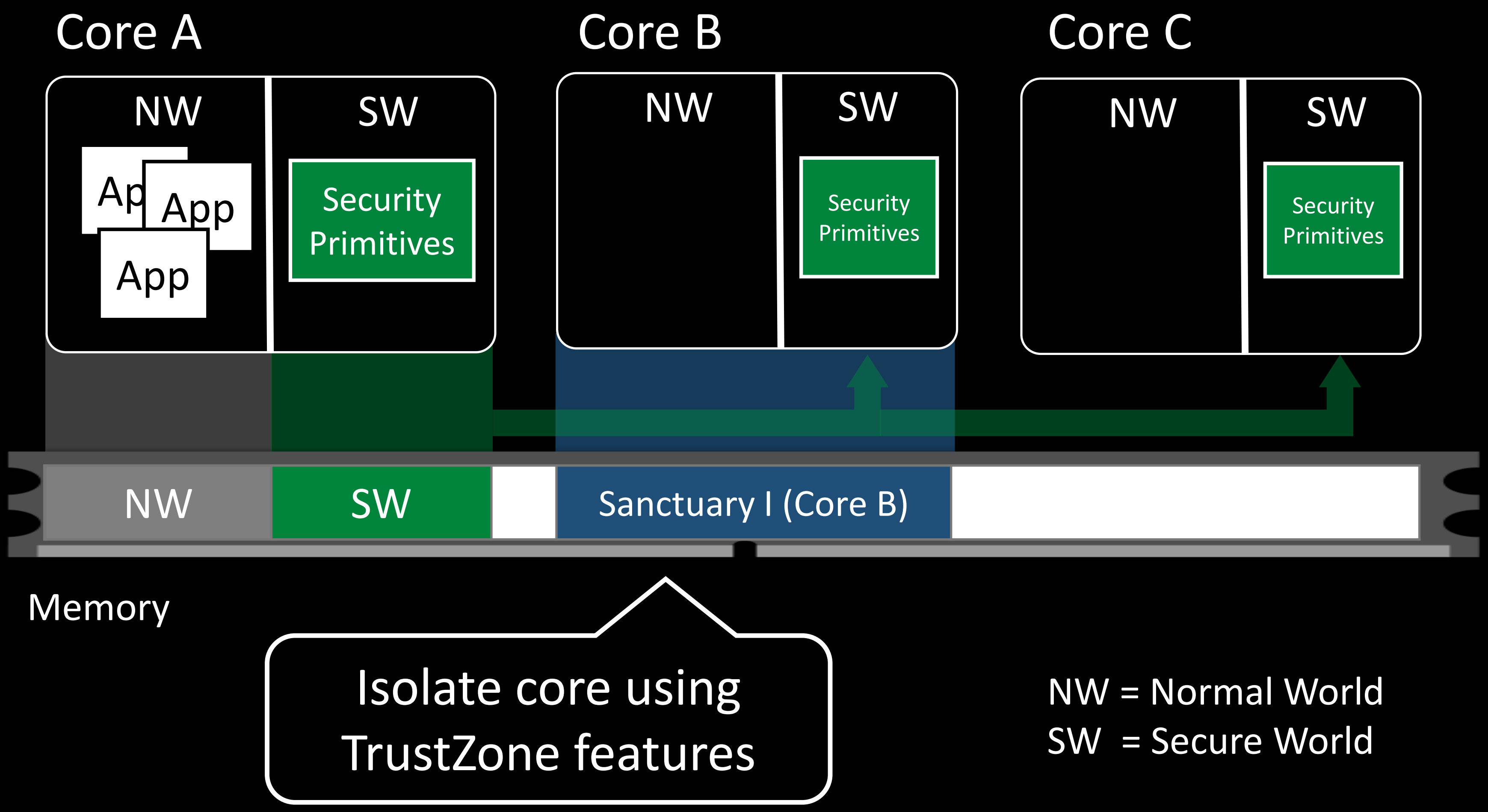
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



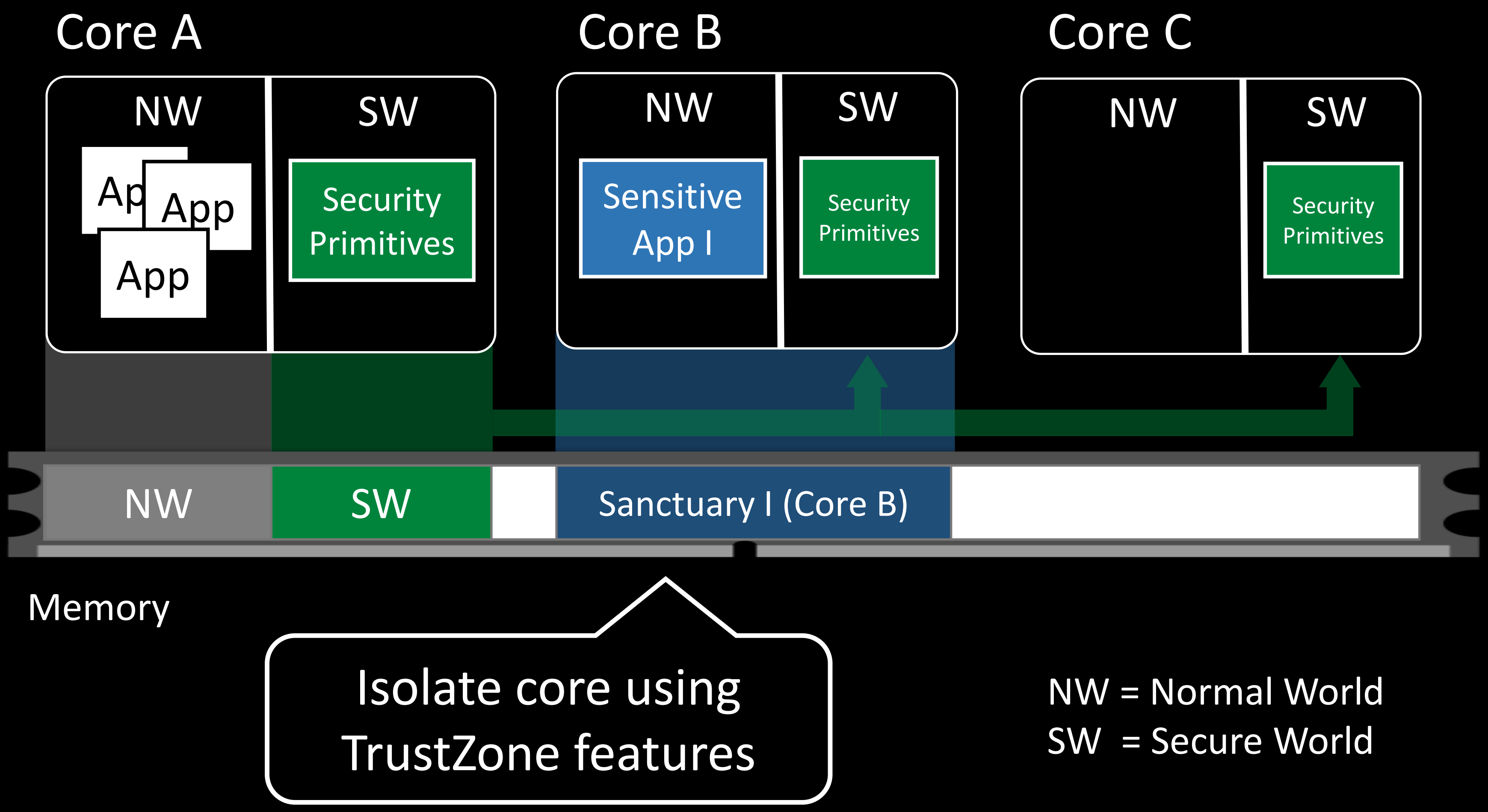
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



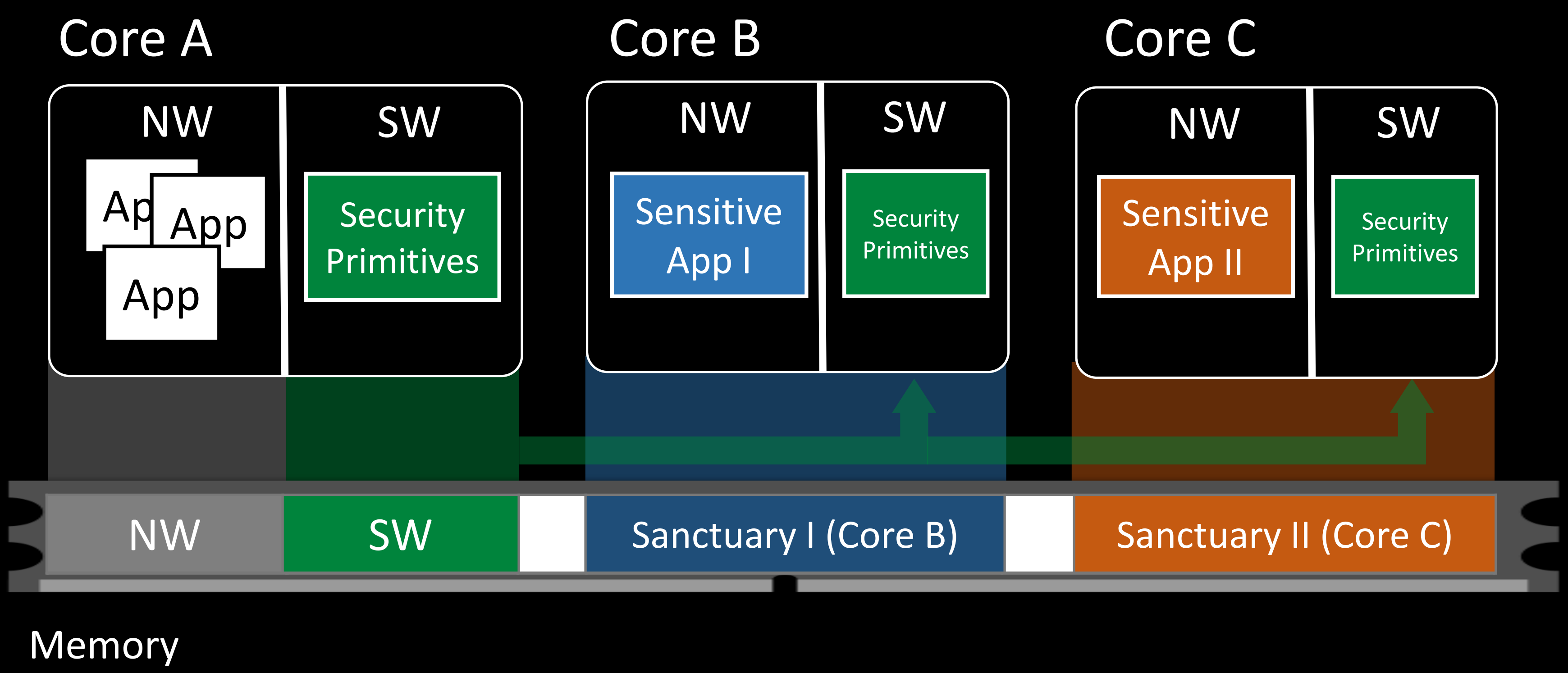
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



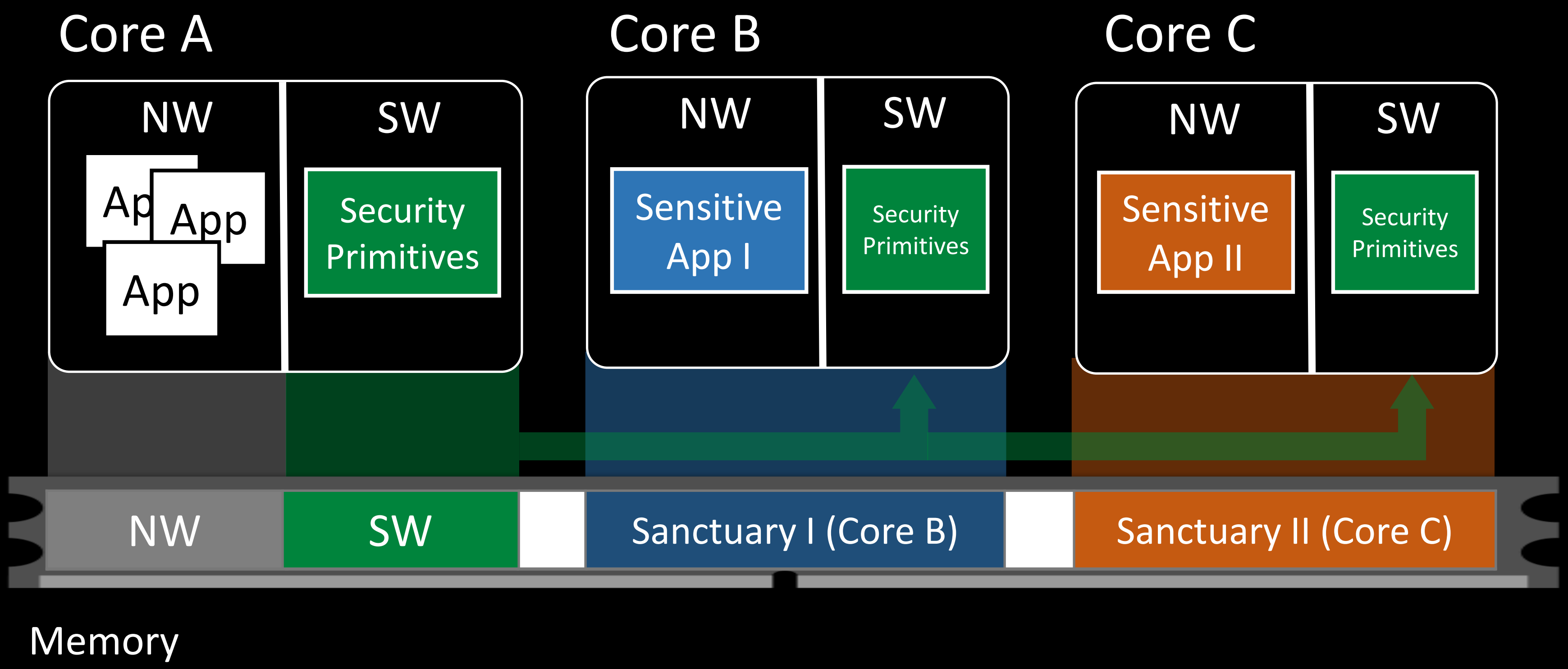
NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



NW = Normal World
SW = Secure World

Sanctuary provides Multi-Domain Isolation



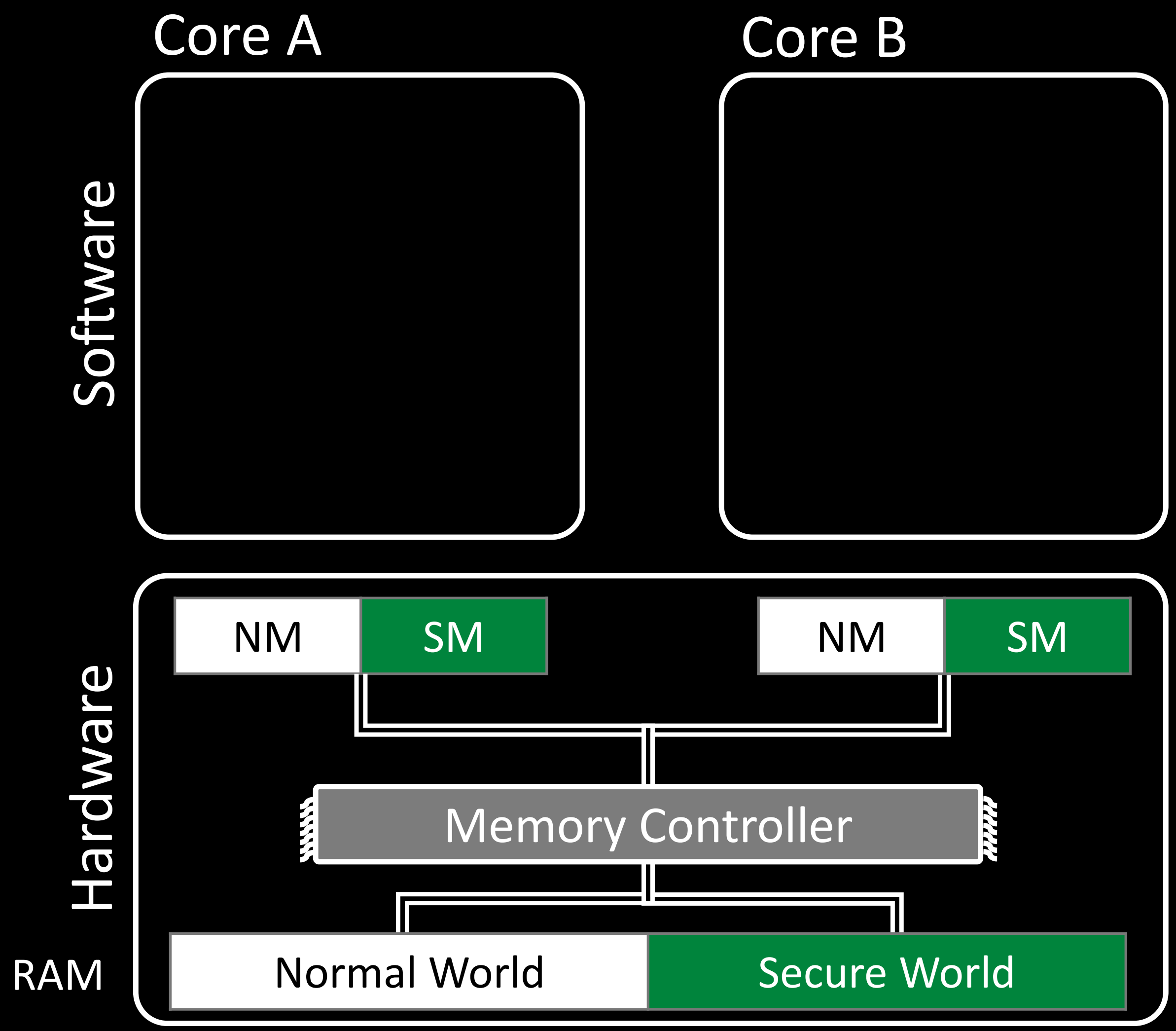
NW = Normal World
SW = Secure World

Challenges revisited

- Only using TrustZone features.
  No new HW design
- Vendor can keep existing code in SW
- No heavy influence on commodity OS

Technical Details of Sanctuary

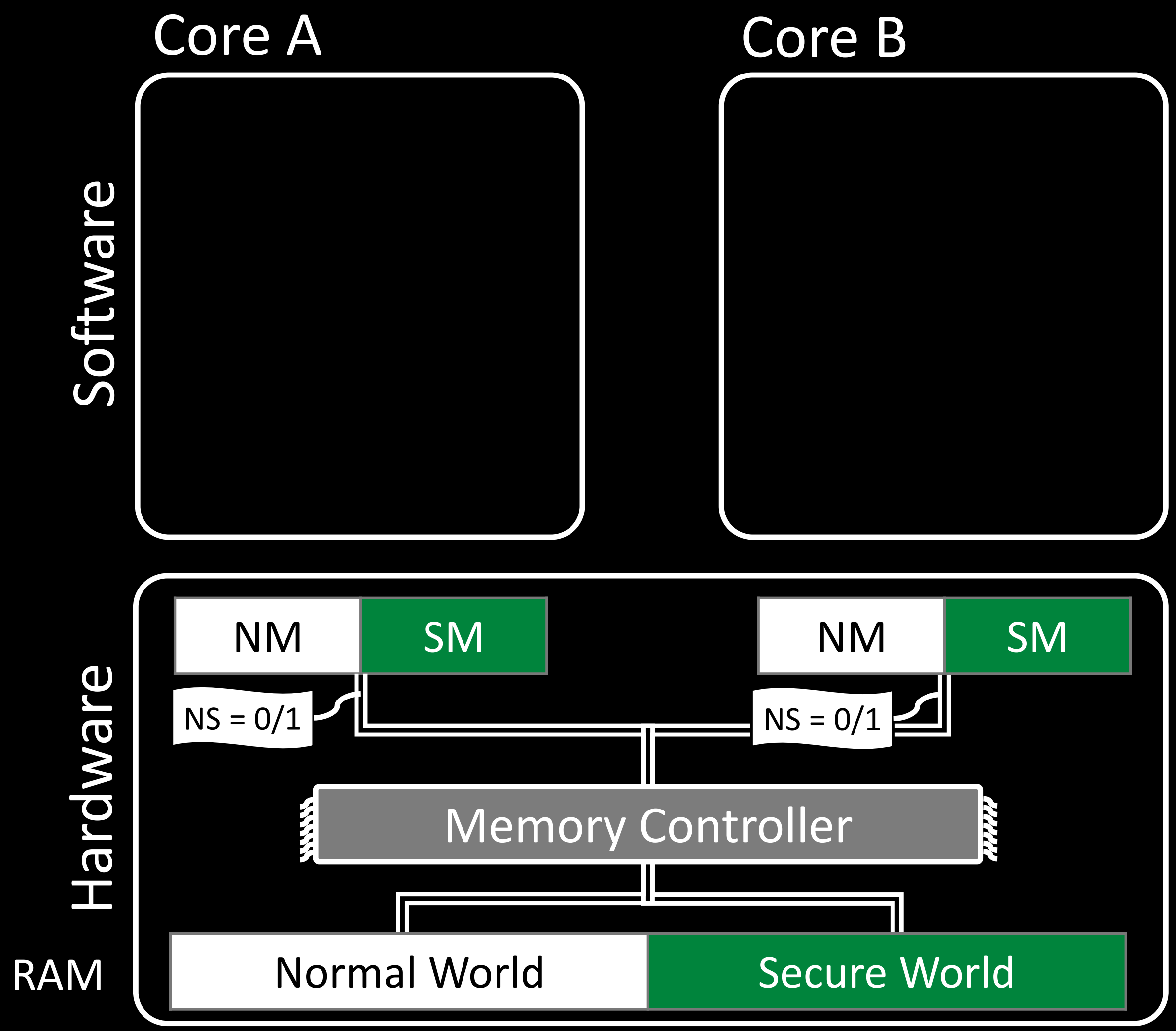
Going Beyond TrustZone …



NM = Normal Mode

SM = Secure Mode

Going Beyond TrustZone …

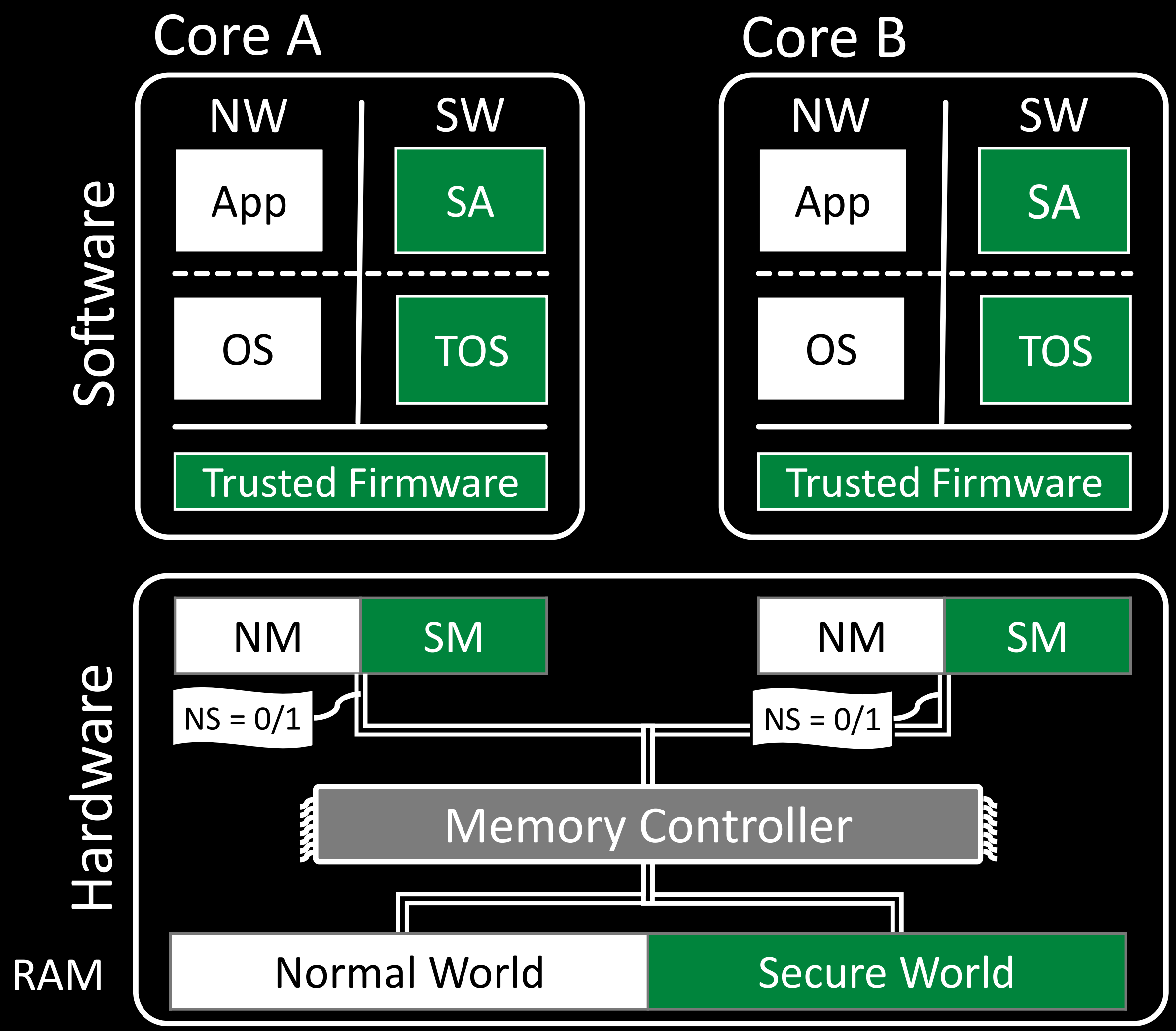


NM = Normal Mode

SM = Secure Mode

NS = Non-Secure

Going Beyond TrustZone …



SA = Sensitive App

TOS = Trusted OS

NW = Normal World

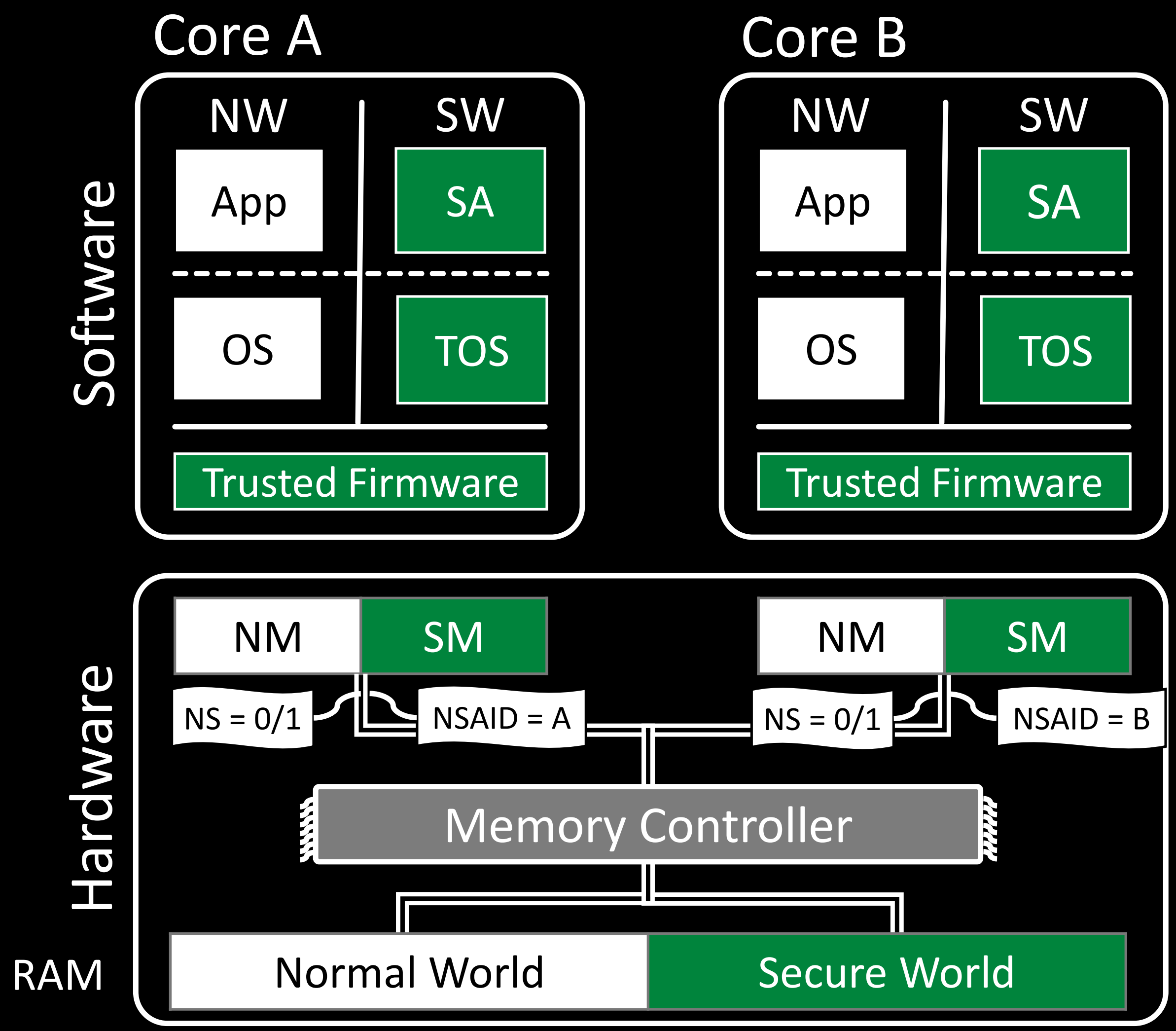
SW = Secure World

NM = Normal Mode

SM = Secure Mode

NS = Non-Secure

Going Beyond TrustZone …



SA = Sensitive App

TOS = Trusted OS

NW = Normal World

SW = Secure World

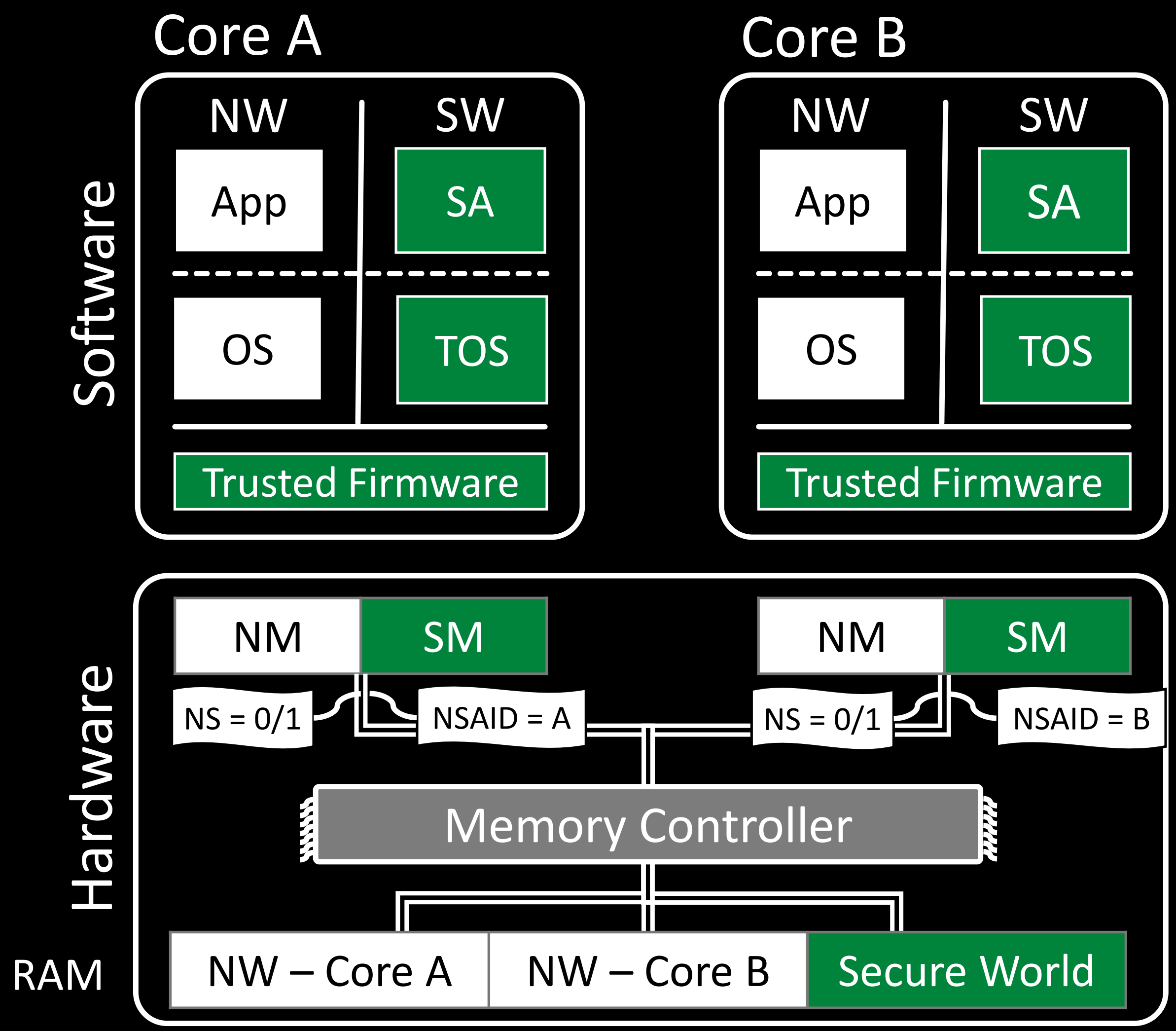
NM = Normal Mode

SM = Secure Mode

NS = Non-Secure

NSAID = Non-Secure Access ID

Going Beyond TrustZone …



SA = Sensitive App

TOS = Trusted OS

NW = Normal World

SW = Secure World

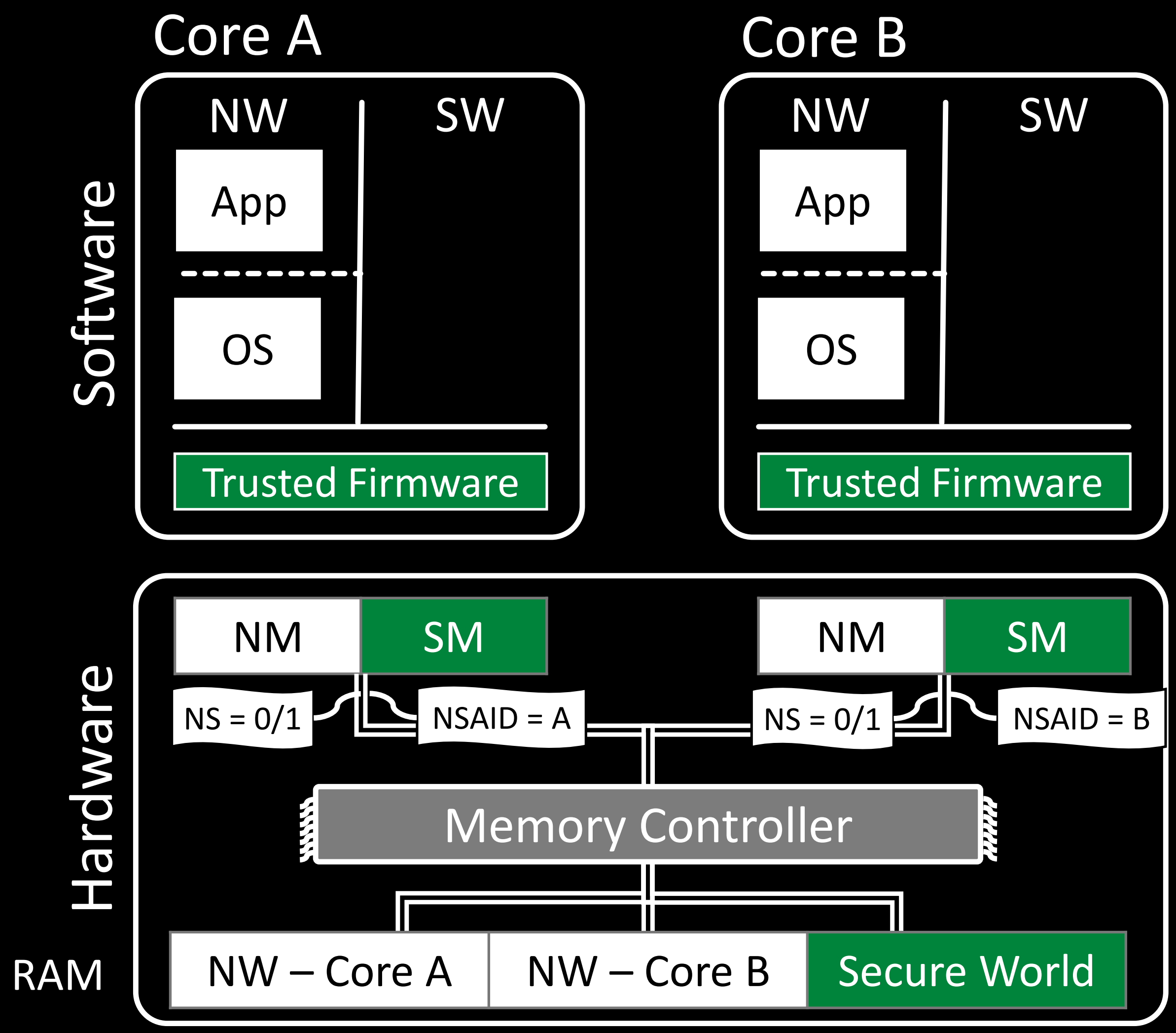
NM = Normal Mode

SM = Secure Mode

NS = Non-Secure

NSAID = Non-Secure Access ID

Going Beyond TrustZone …



SA = Sensitive App

TOS = Trusted OS

NW = Normal World

SW = Secure World

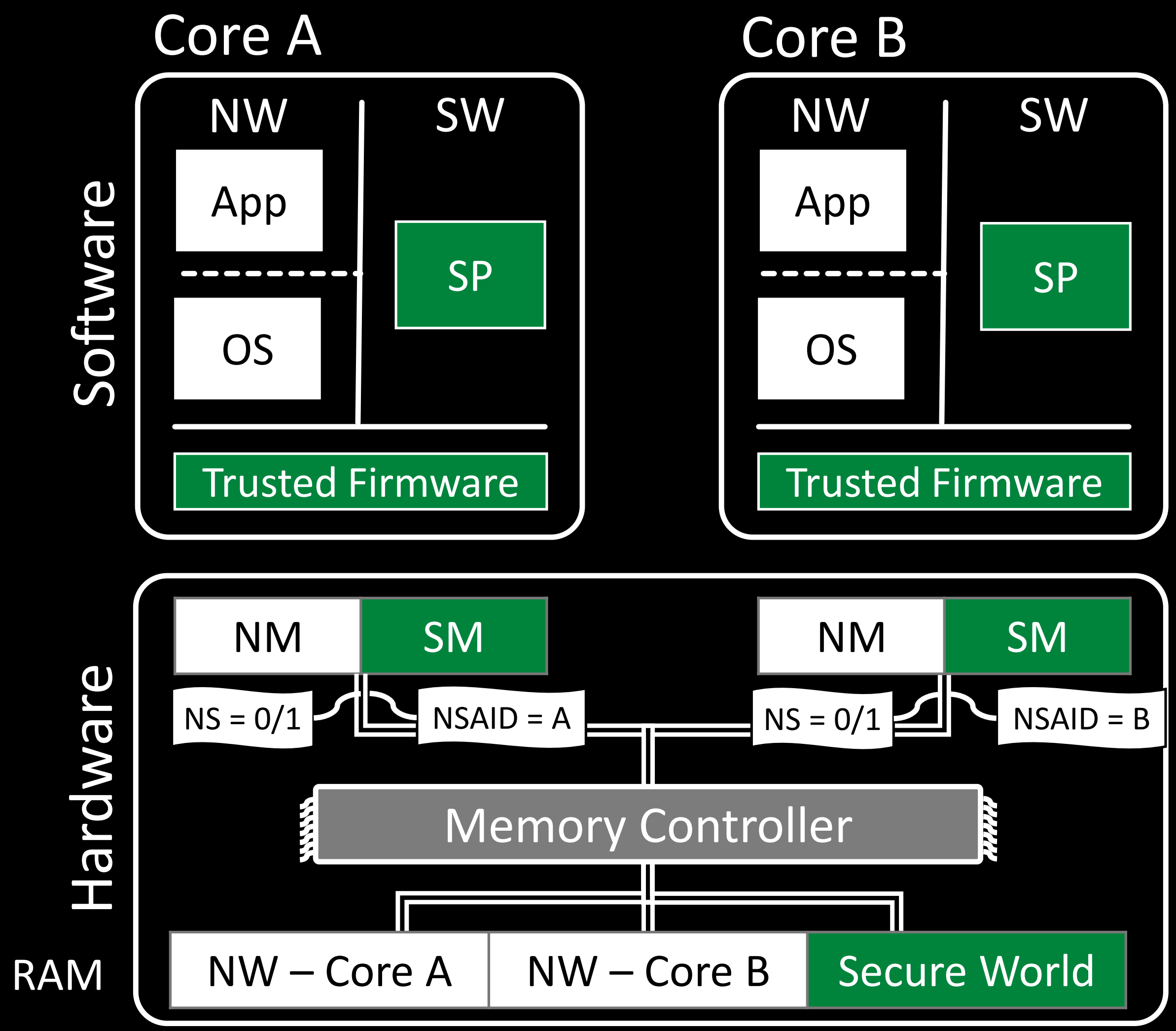
NM = Normal Mode

SM = Secure Mode

NS = Non-Secure

NSAID = Non-Secure Access ID

Going Beyond TrustZone …



SP = Security Primitives

SA = Sensitive App

TOS = Trusted OS

NW = Normal World

SW = Secure World

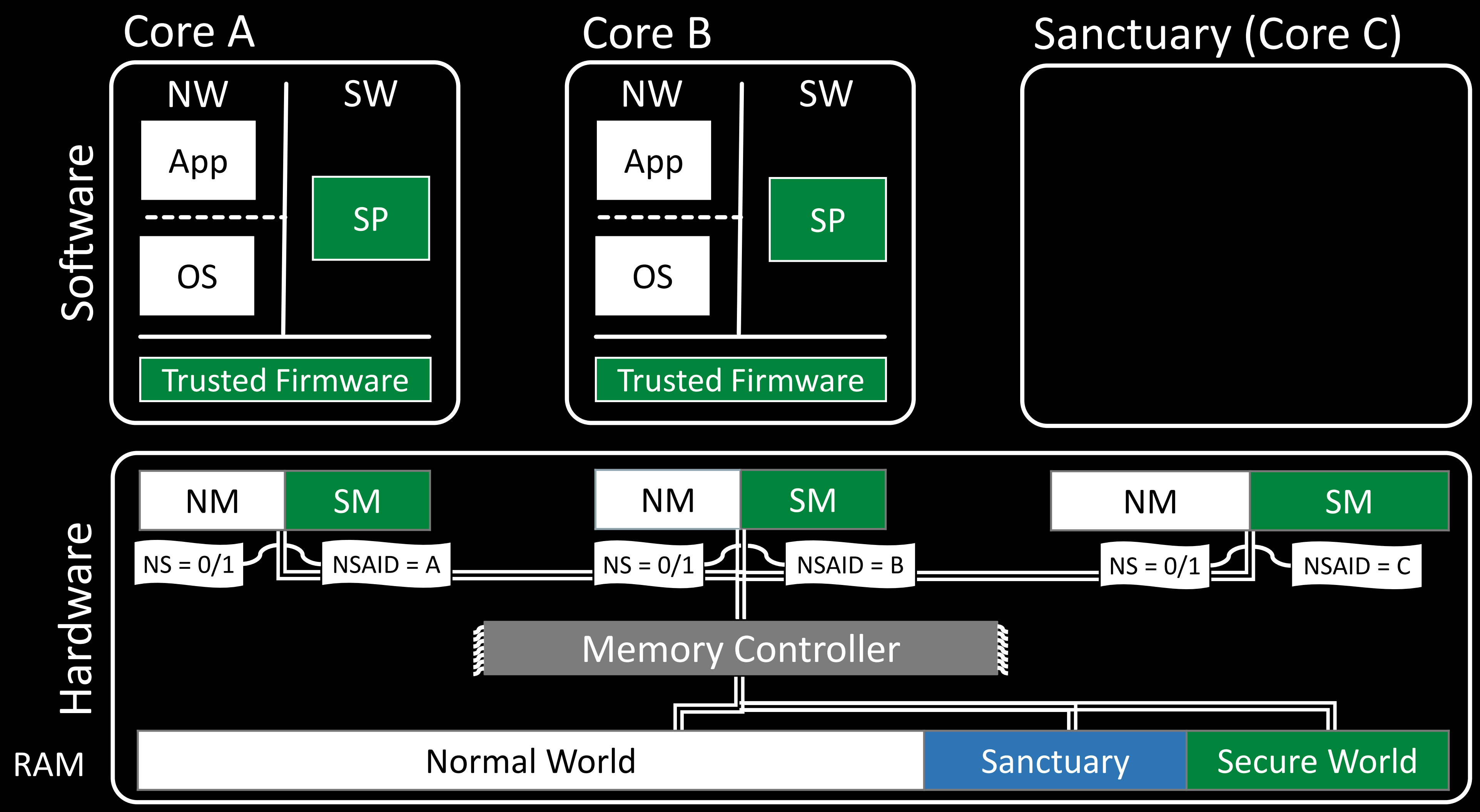
NM = Normal Mode

SM = Secure Mode

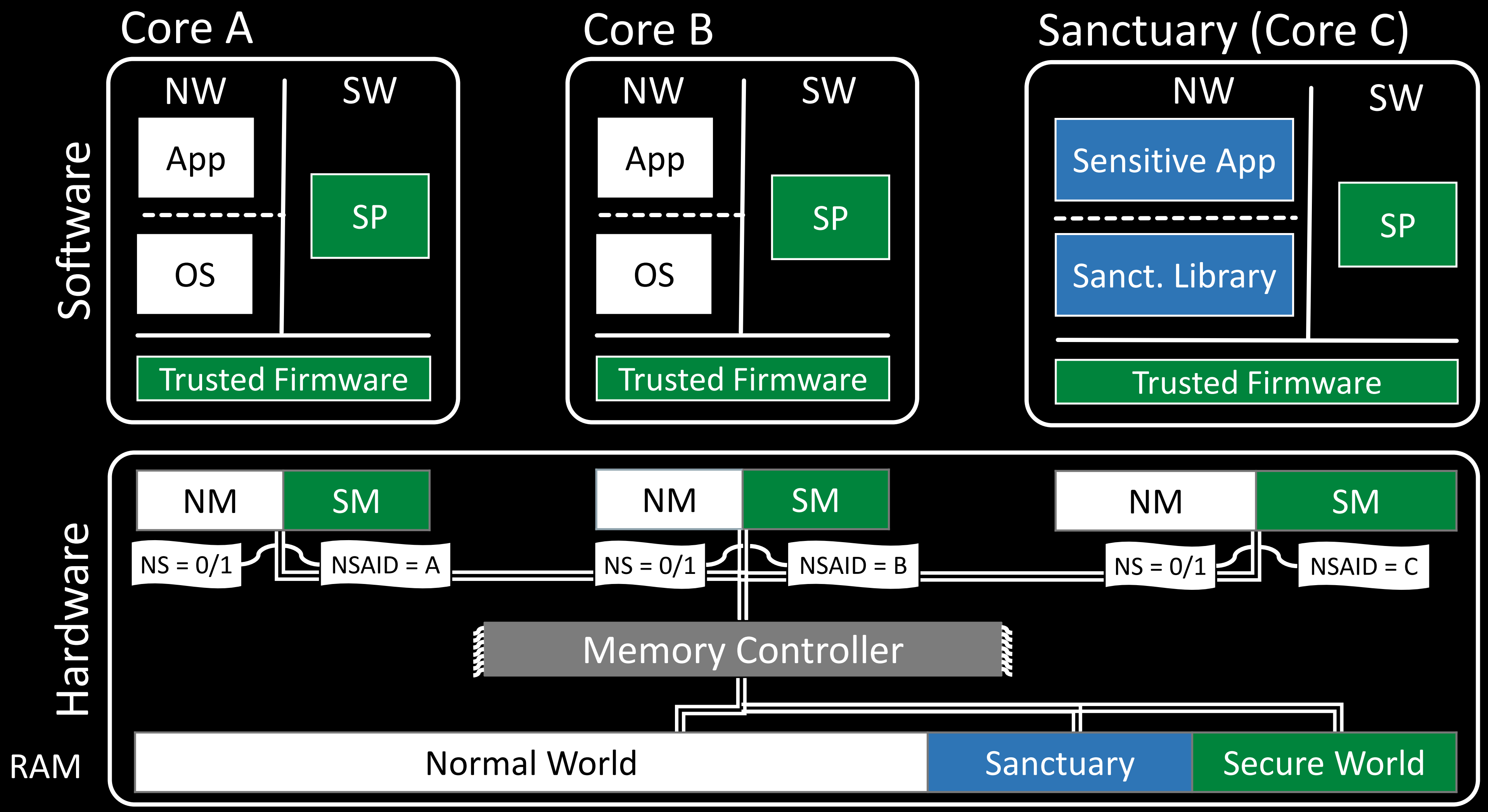
NS = Non-Secure

NSAID = Non-Secure Access ID

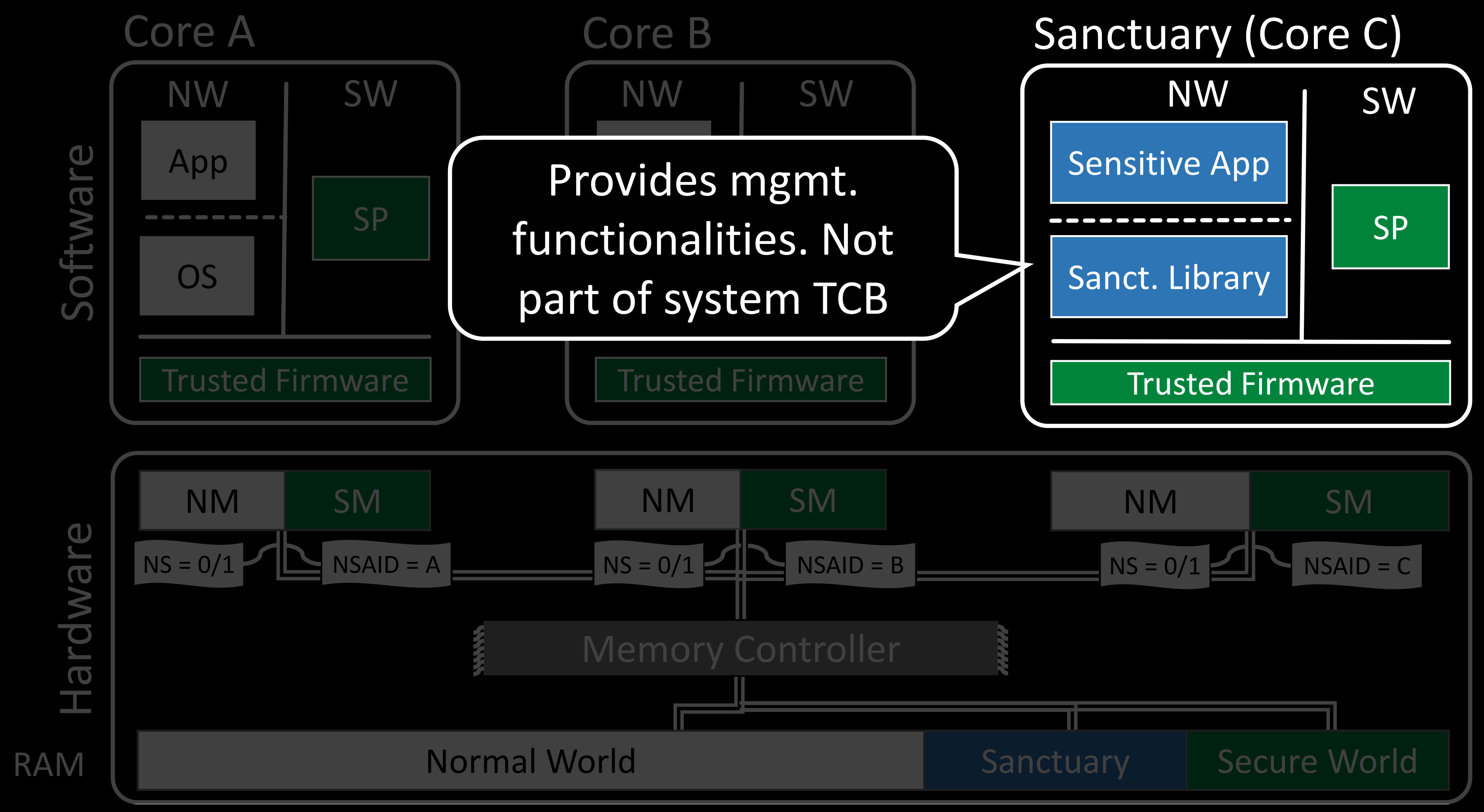
Going Beyond TrustZone …



Going Beyond TrustZone ...



Going Beyond TrustZone ...



Evaluation of Sanctuary PoC

Security Considerations

- ✓ Protection from compromised OS
  before, after and during runtime of sensitive app
- ✓ Protection from malicious sensitive app
  sensitive app isolated from OS and other sensitive apps
- ✓ Protection from cache-side channel attacks
  flush exclusive caches, exclude sensitive apps from shared caches

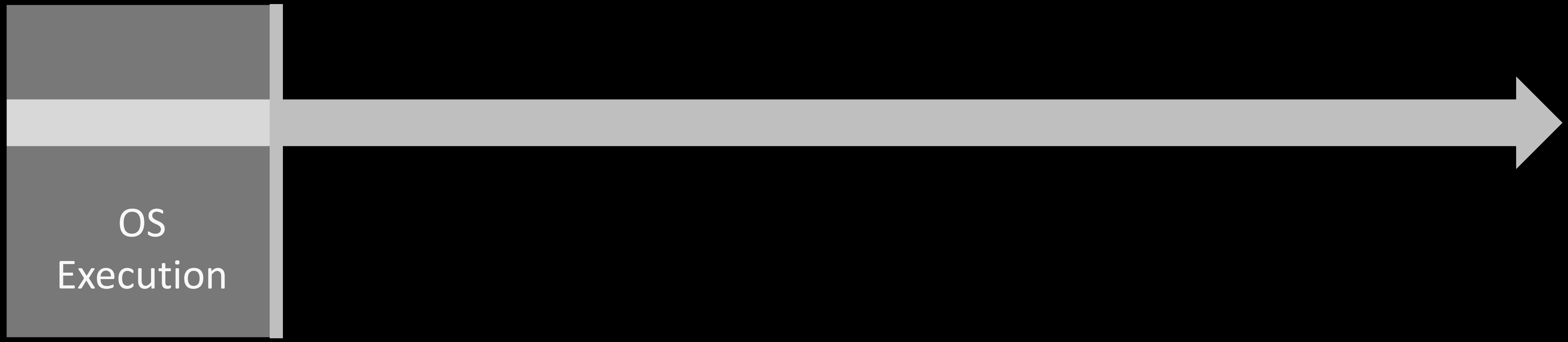
Performance Evaluation

Performance Evaluation

Sanctuary Life Cycle

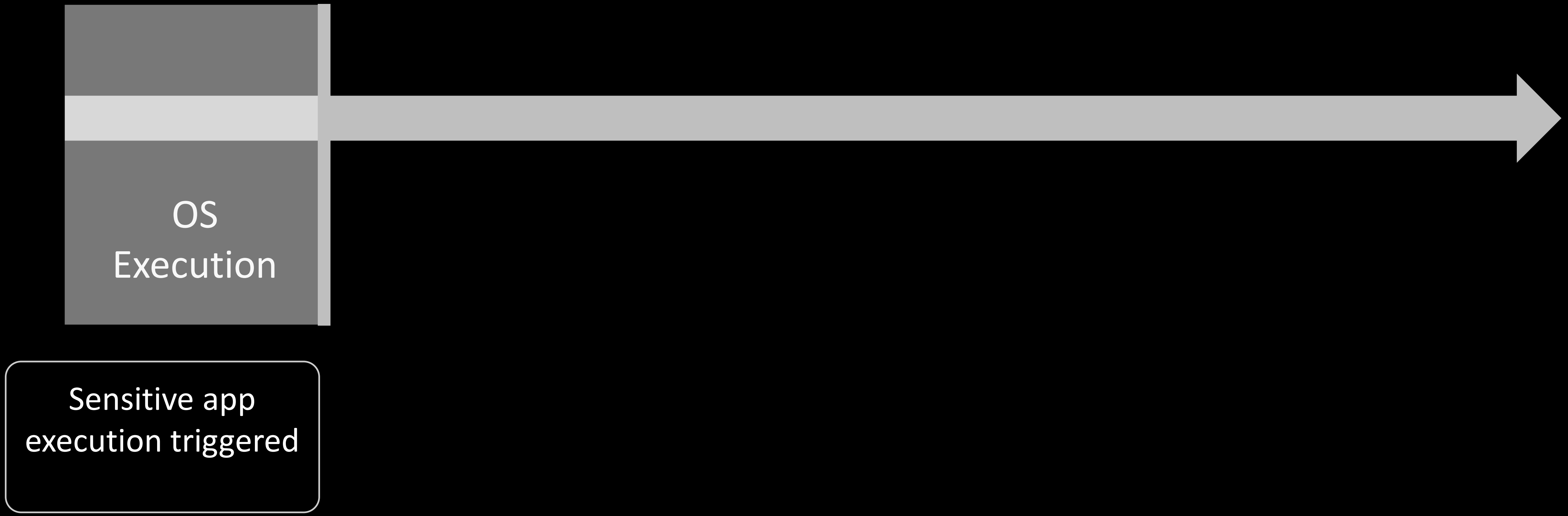
Performance Evaluation

Sanctuary Life Cycle

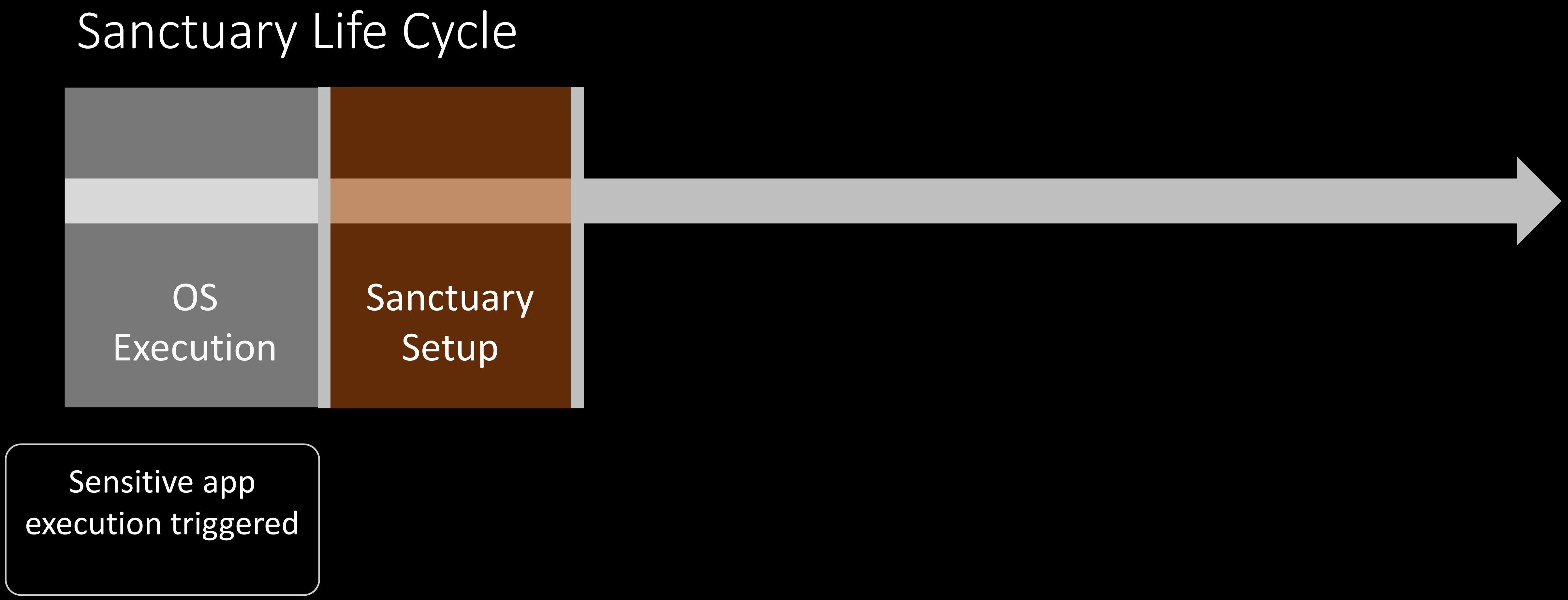


Performance Evaluation

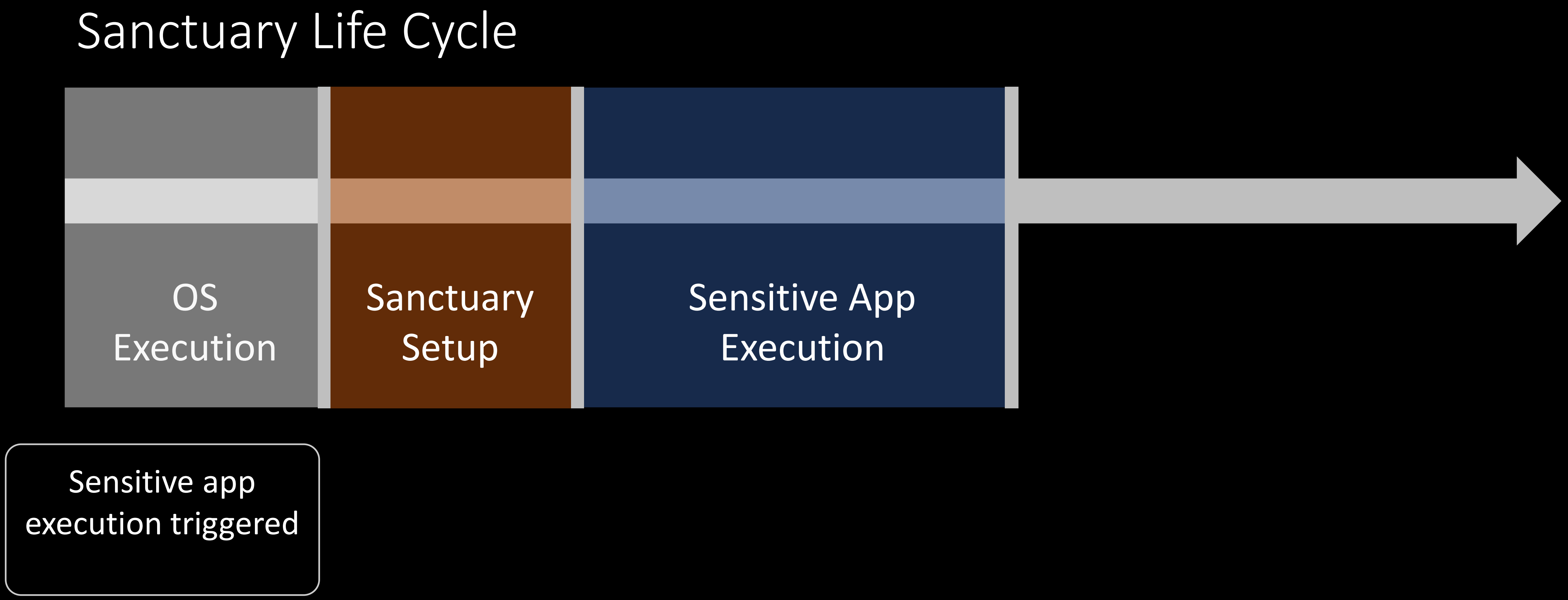
Sanctuary Life Cycle



Performance Evaluation



Performance Evaluation



Performance Evaluation

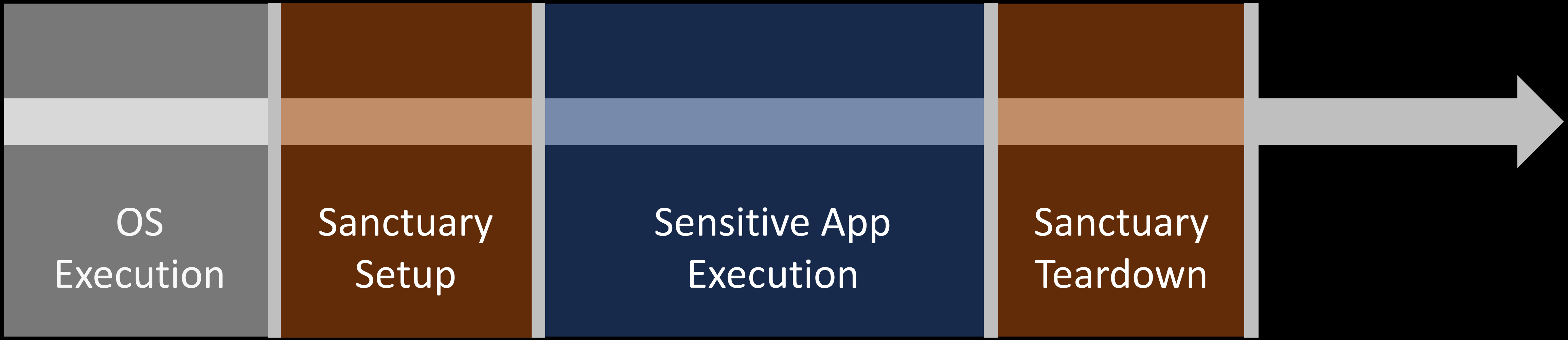
Sanctuary Life Cycle

Sensitive app execution triggered

OS still runs in parallel on other cores

Performance Evaluation

Sanctuary Life Cycle

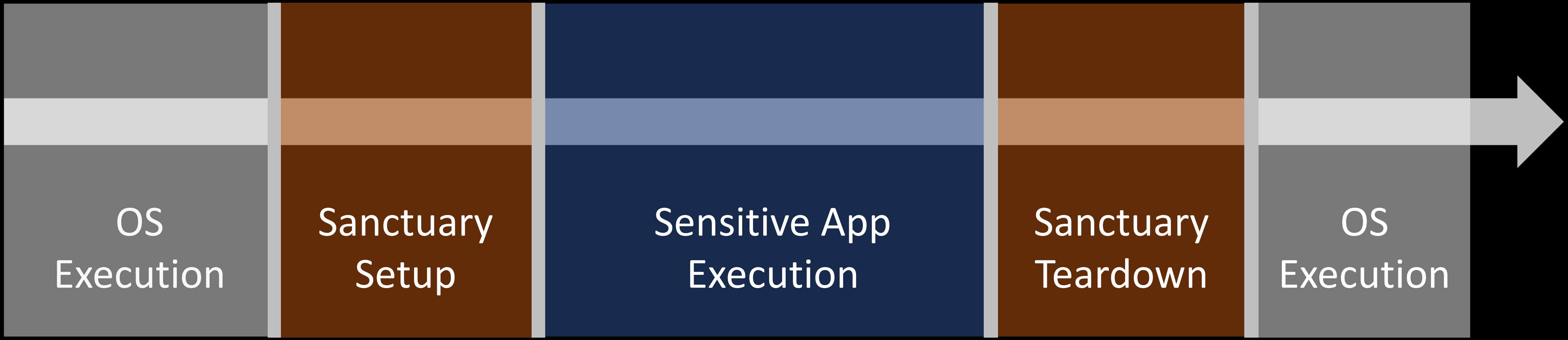


Sensitive app execution triggered

OS still runs in parallel on other cores

Performance Evaluation

Sanctuary Life Cycle

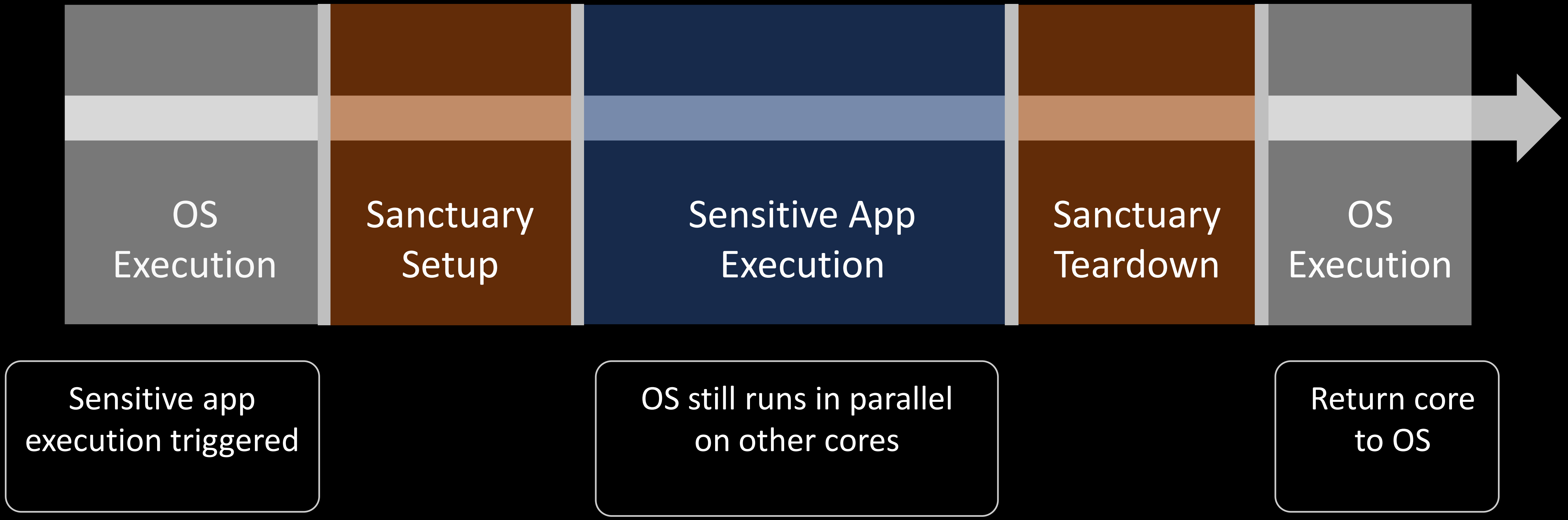


Sensitive app execution triggered

OS still runs in parallel on other cores

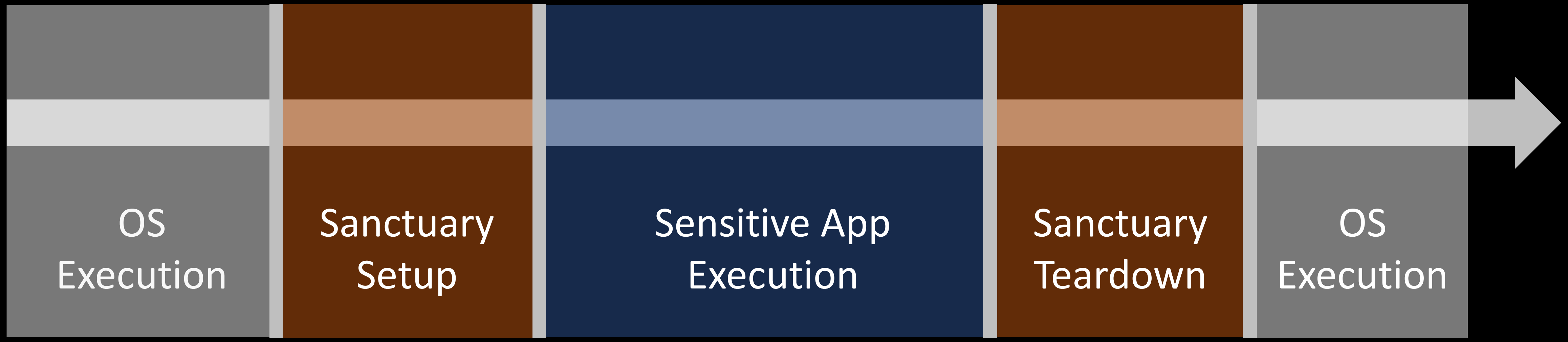
Performance Evaluation

Sanctuary Life Cycle



Performance Evaluation

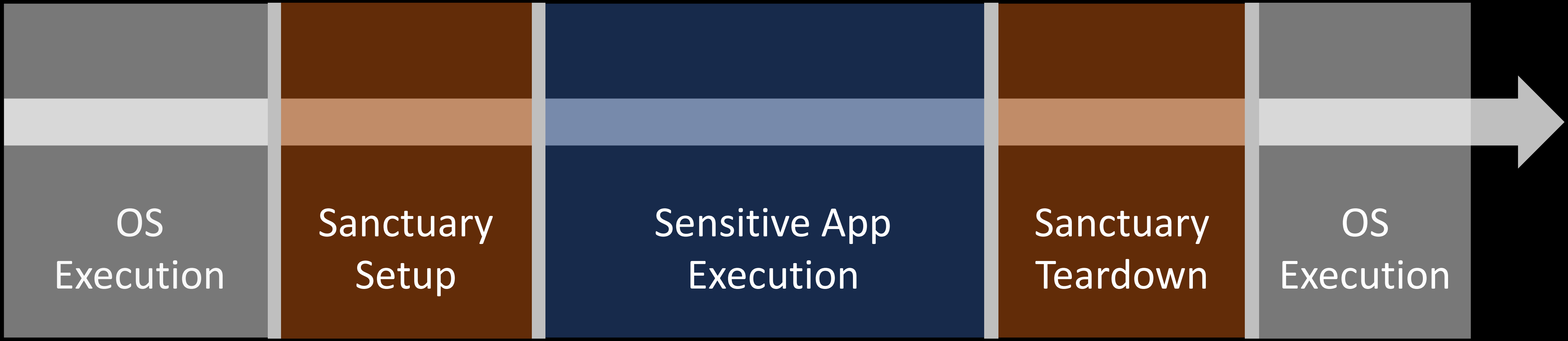
Sanctuary Life Cycle



Sensitive app execution triggered	**440 ms** w/o shared cache	OS still runs in parallel on other cores	**110 ms** w/o shared cache	Return core to OS

Performance Evaluation

Sanctuary Life Cycle



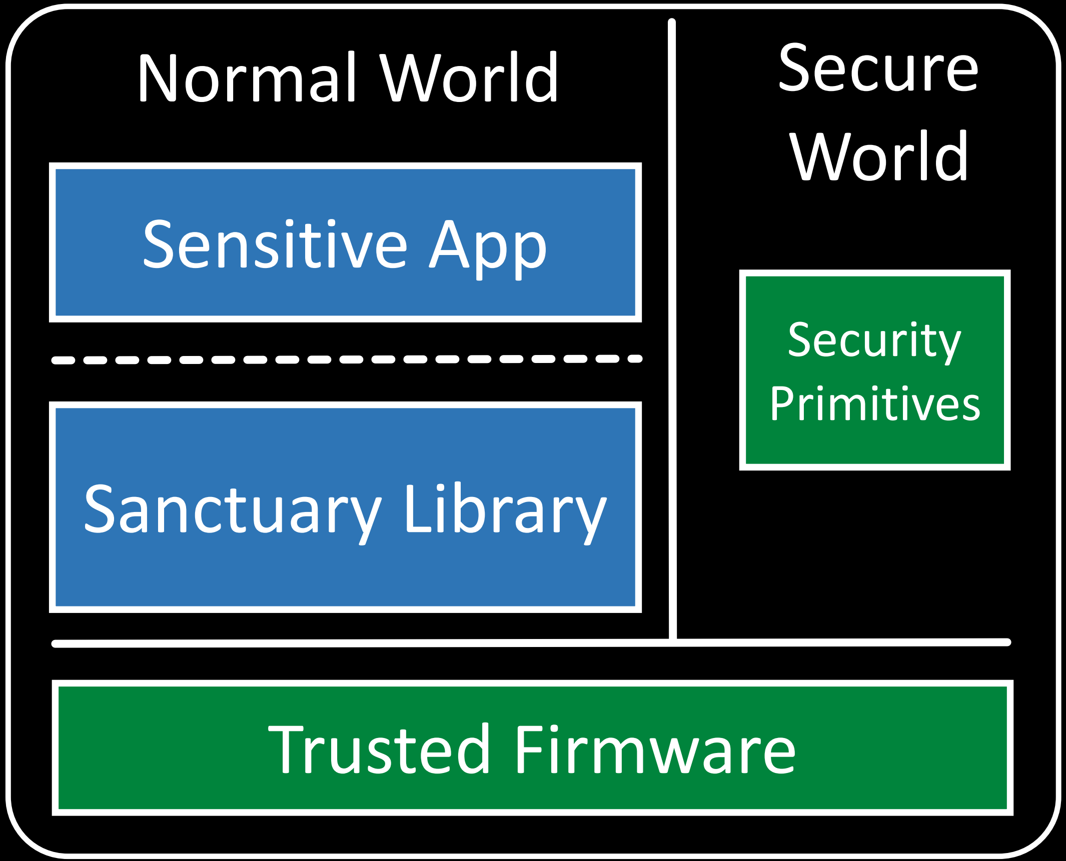
Sensitive app execution triggered

<u>Provision OTP Key:</u>
1.2 s (w/o shared cache)

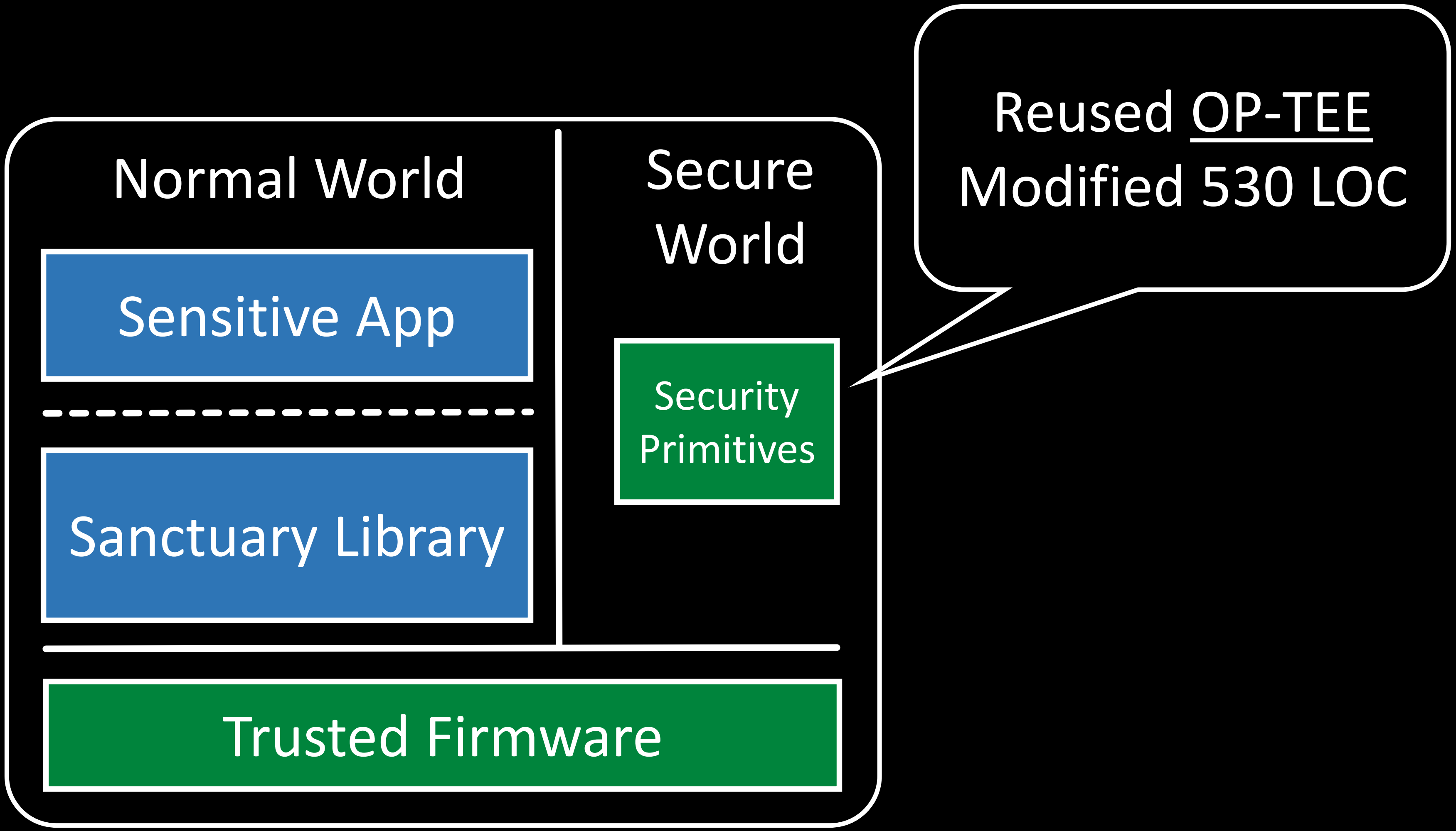
<u>Display OTP:</u>
600 ms (w/o shared cache)

Return core to OS

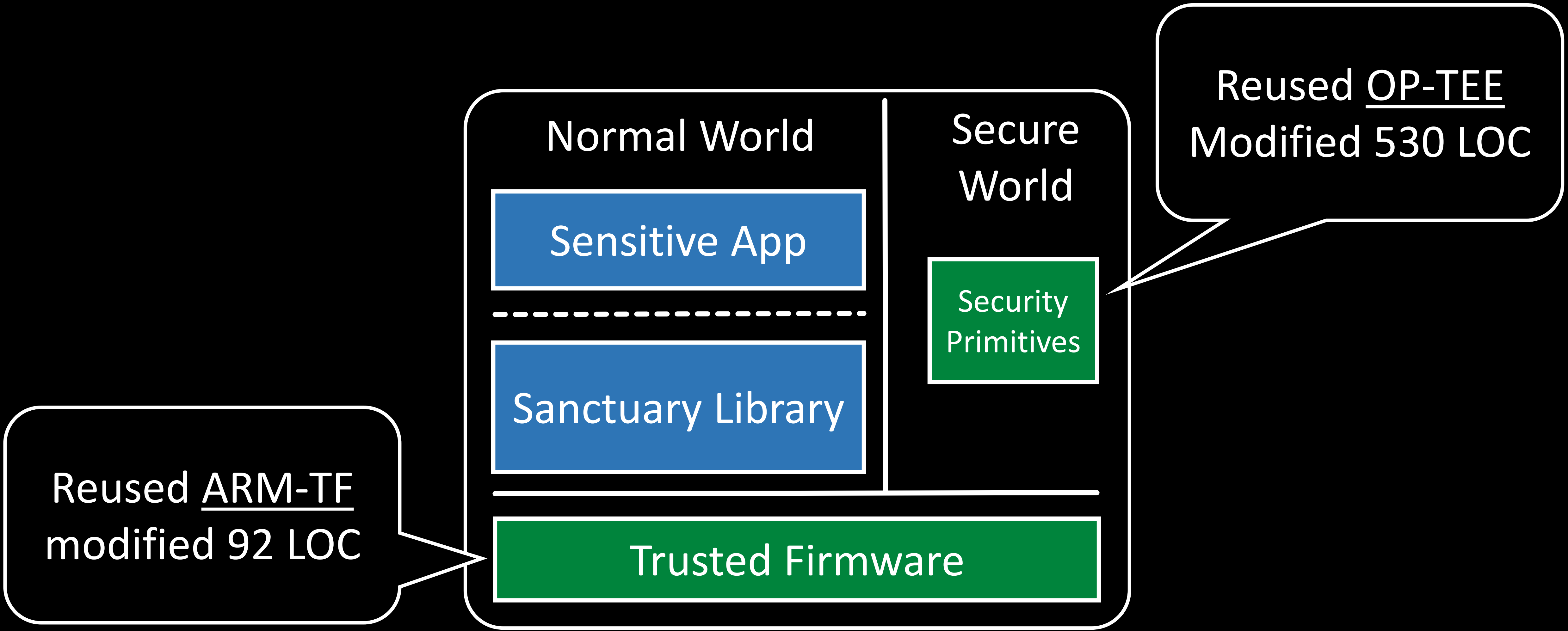
Code Modifications



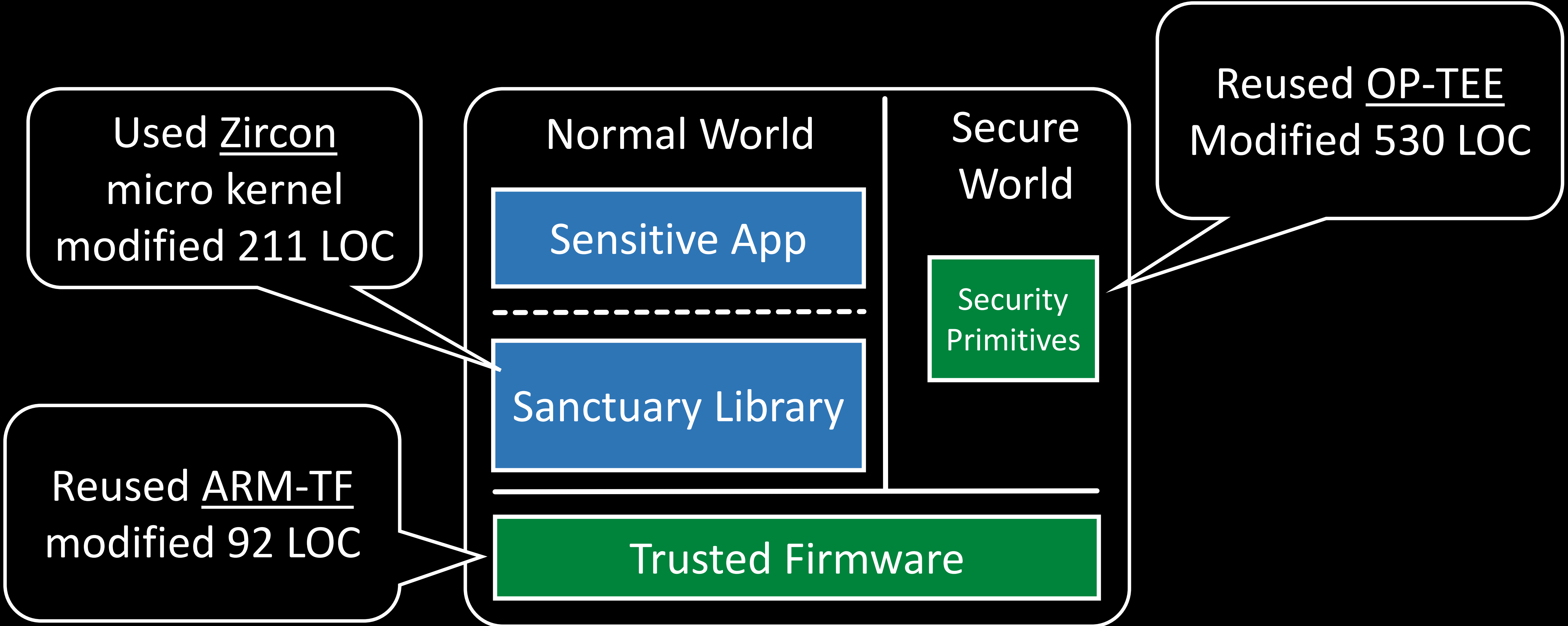
Code Modifications



Code Modifications



Code Modifications



Conclusion

- Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices

Conclusion

- Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  - ✔ Sanctuary does not introduce new HW designs

Conclusion

- Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  - ✔ Sanctuary does not introduce new HW designs
  - ✔ Sanctuary does not replace existing code bases

Conclusion

- Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  - ✓ Sanctuary does not introduce new HW designs
  - ✓ Sanctuary does not replace existing code bases
  - ✓ Sanctuary does not impact the commodity OS heavily

Conclusion

- Sanctuary provides an ecosystem of mutually distrusted enclaves on ARM devices
  - ✓ Sanctuary does not introduce new HW designs
  - ✓ Sanctuary does not replace existing code bases
  - ✓ Sanctuary does not impact the commodity OS heavily
- Current Work
  - ➢ Implement further use cases (IP protection of ML algorithms, digital car key)
  - ➢ Sanctuary for RISC-V

Questions ?

emmanuel.stapf@trust.tu-darmstadt.de