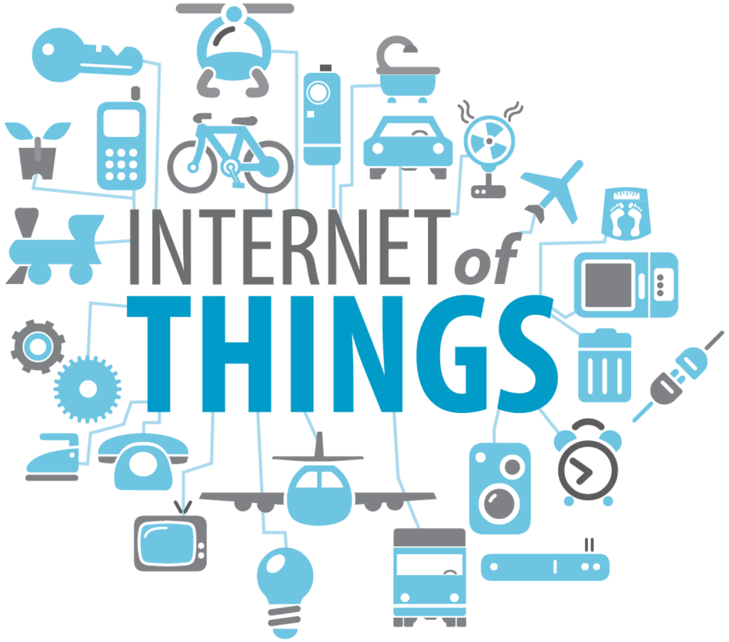
BenchIoT: A Security Benchmark for The Internet of Things

Naif Almakhdhub, Abraham Clements, Mathias Payer, and Saurabh Bagchi

Internet of Things

- **The number of IoT devices is expected to exceed 20 billion by 2020.**

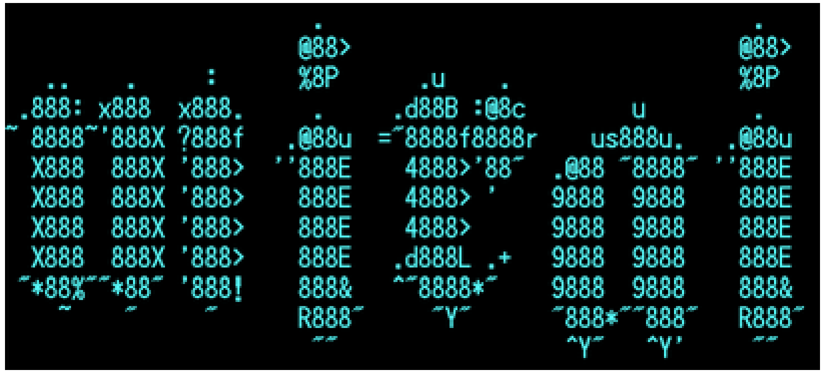
- **Many will be microcontroller based systems (IoT-µCs).**
  - Run single static binary image directly on the hardware.
  - Can be with/without an OS (bare-metal).
  - Direct access to peripherals and processor.
  - Small memory.



- **Examples:**
  - WiFi System on Chip
  - Cyber-physical systems
  - UAVs

Internet of Things Security

- In 2016, one of the largest DDoS attack to date was caused by IoT devices[1].



```
                       .                               .
                      @88>                            @88>
   ..    .     :      %8P      .u    .                %8P
 .888: x888  x888.     .    .d88B :@8c         u       .
~ 8888~'888X ?888f   .@88u =~8888f8888r    us888u.   .@88u
  X888  888X '888>  ''888E   4888>'88~  .@88 ~8888~ ''888E
  X888  888X '888>    888E   4888>  '    9888 9888    888E
  X888  888X '888>    888E   4888>       9888 9888    888E
  X888  888X '888>    888E  .d888L .+    9888 9888    888E
 ~*88%~~*88~ '888!    888&  ^~8888*~     9888 9888    888&
    ~    ~     ~      R888~    ~Y~      ~888*~~888~   R888~
                       ~~                ^Y~   ^Y'     ~~
```

- In 2017, Google's Project Zero used a vulnerable WiFi SoC to gain control of the application processor on smart phones[2].

[1] https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/
[2] https://googleprojectzero.blogspot.co.uk/2017/04/over-air-exploiting-broadcoms-wi-fi_4.html

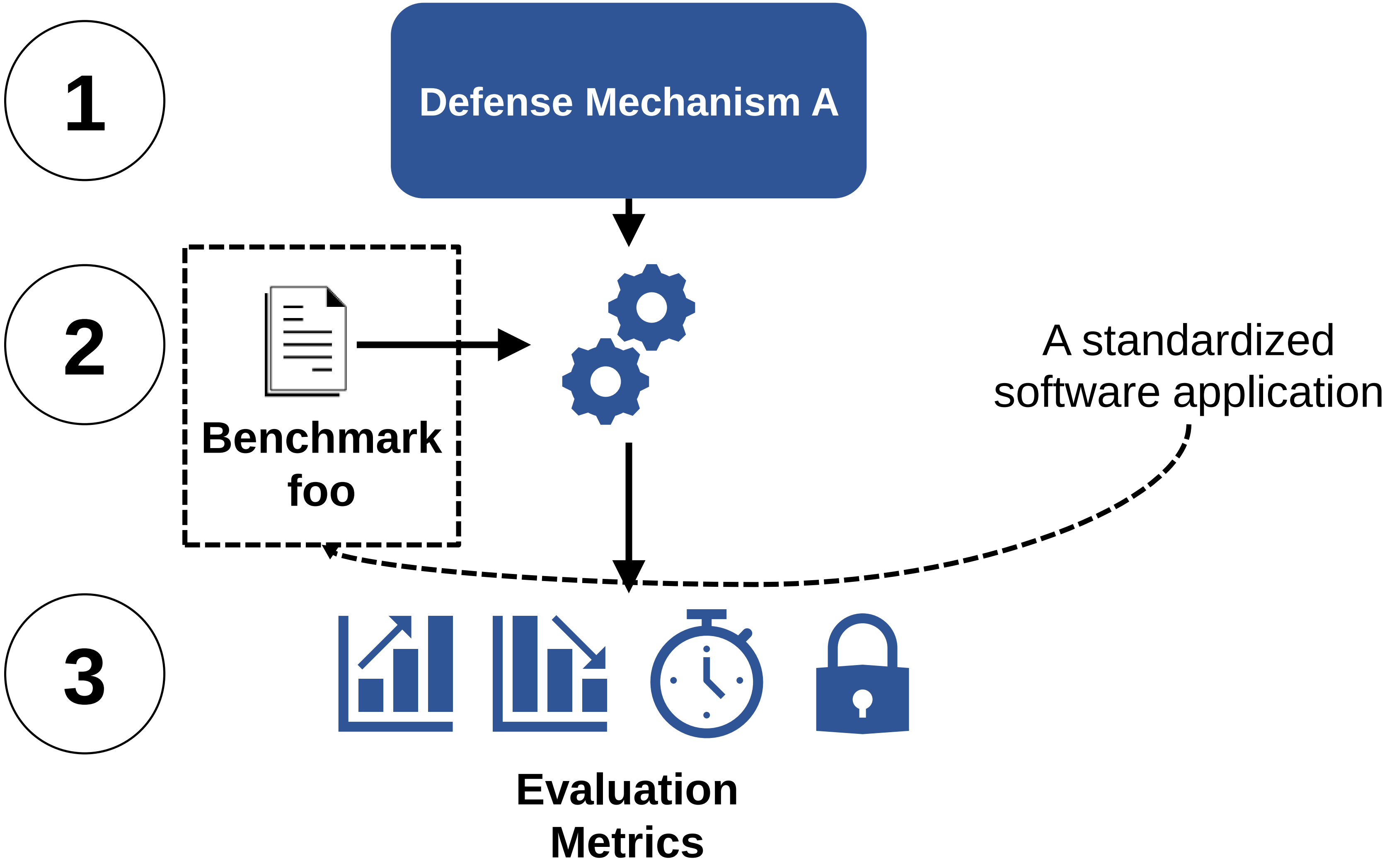
Evaluation in Current IoT Defenses

- **Multiple defenses have been proposed.**
  - TyTan[DAC15], TrustLite[EurSys14], C-FLAT [CCS16], nesCheck[AsiaCCS17], SCFP[EuroS&P18], LiteHAX[ICCAD18] CFI CaRE [RAID17], ACES[SEC18], MINION [NDSS18], EPOXY [S&P17]
- **How are they evaluated?**
  - Ad-hoc evaluation.

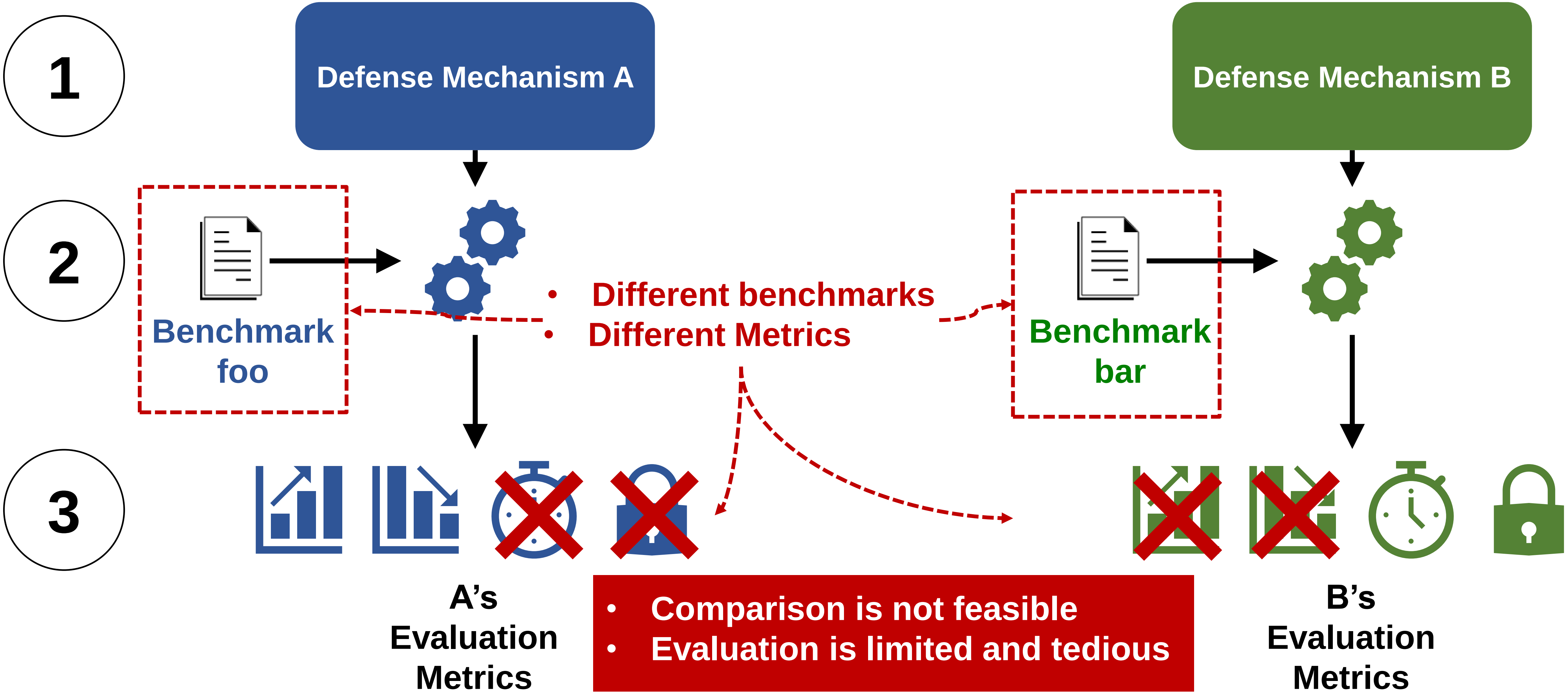
Defense	**Evaluation Type**	
	Benchmark	**Case Study**
TyTan		✓
TrustLite		✓
C-FLAT		✓
nesCheck		✓
SCFP	Dhrystone[1]	✓
LiteHAX	CoreMark[2]	✓
CFI CaRE	Dhrystone[1]	✓
ACES		✓
Minion		✓
EPOXY	BEEBS[3]	✓

[1] R. P. Weicker, “Dhrystone: a synthetic systems programming benchmark,” Communications of the ACM, vol. 27, no. 10, pp. 1013–1030, 1984
[2] EEMBC, “Coremark - industry-standard benchmarks for embedded systems,” http://www.eembc.org/coremark.
[3] J. Pallister, S. J. Hollis, and J. Bennett, “BEEBS: open benchmarks for energy measurements on embedded platforms,” CoRR, vol. abs/1308.5174, 2013.[Online]. Available: http://arxiv.org/abs/1308.5174

IoT-µCs Evaluation (Ideally)



IoT-µCs Evaluation (Reality)



1

2

3

Defense Mechanism A

Defense Mechanism B

Benchmark foo

Benchmark bar

- Different benchmarks
- Different Metrics

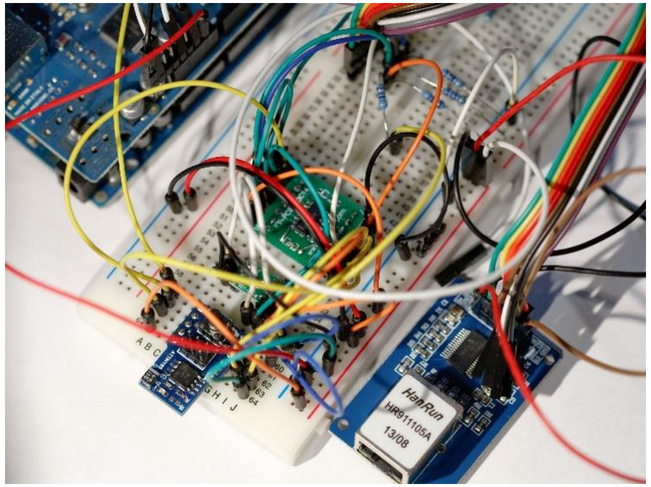
A's Evaluation Metrics

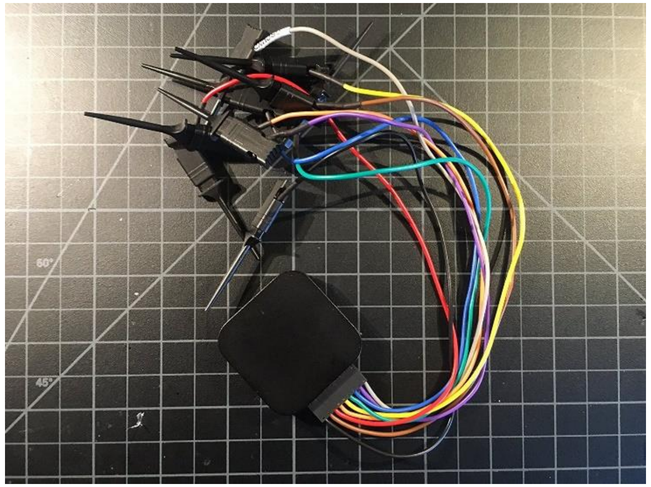
B's Evaluation Metrics

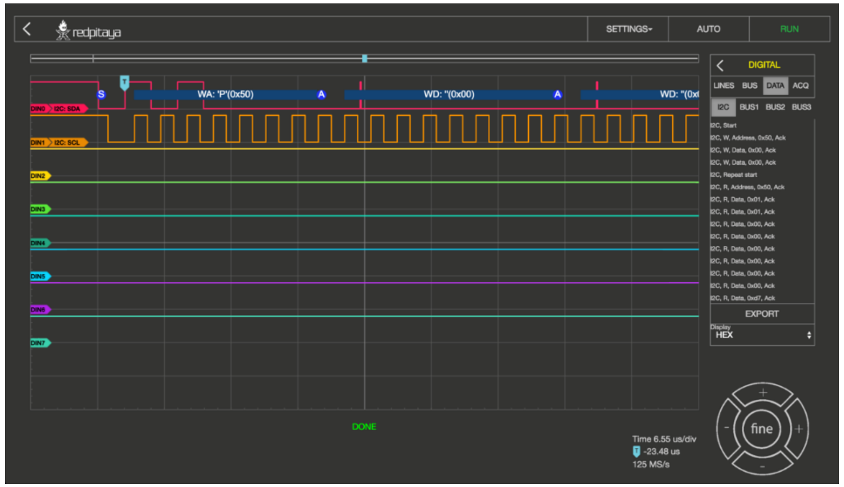
- Comparison is not feasible
- Evaluation is limited and tedious

Why not use Existing Benchmark?

- **Current benchmarks are rigid and simplistic.**
  - Many are just one file with simple application.
  - Metrics are limited and cumbersome to collect.
  - Hardware dependent.
- **Do not use peripherals.**
- **No network connectivity.**







```
I2C, Start
I2C, W, Address, 0x50, Ack
I2C, W, Data, 0x00, Ack
I2C, W, Data, 0x00, Ack
I2C, Repeat start
I2C, R, Address, 0x50, Ack
I2C, R, Data, 0x01, Ack
I2C, R, Data, 0x01, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0x00, Ack
I2C, R, Data, 0xd7, Ack
```

Display: HEX

Time 6.55 us/div
-23.48 us
125 MS/s

Proposed Solution: BenchIoT

- **BenchIoT provides a suite of benchmark applications and an evaluation framework.**

- **A realistic set of *IoT* benchmarks.**
  - Mimics common IoT characteristics, e.g., tight coupling with sensors and actuators.
  - Works for both with/without an OS.

- **Our evaluation framework is versatile and portable.**
  - A software based approach.
  - Can collect metrics related to security and resource usage.

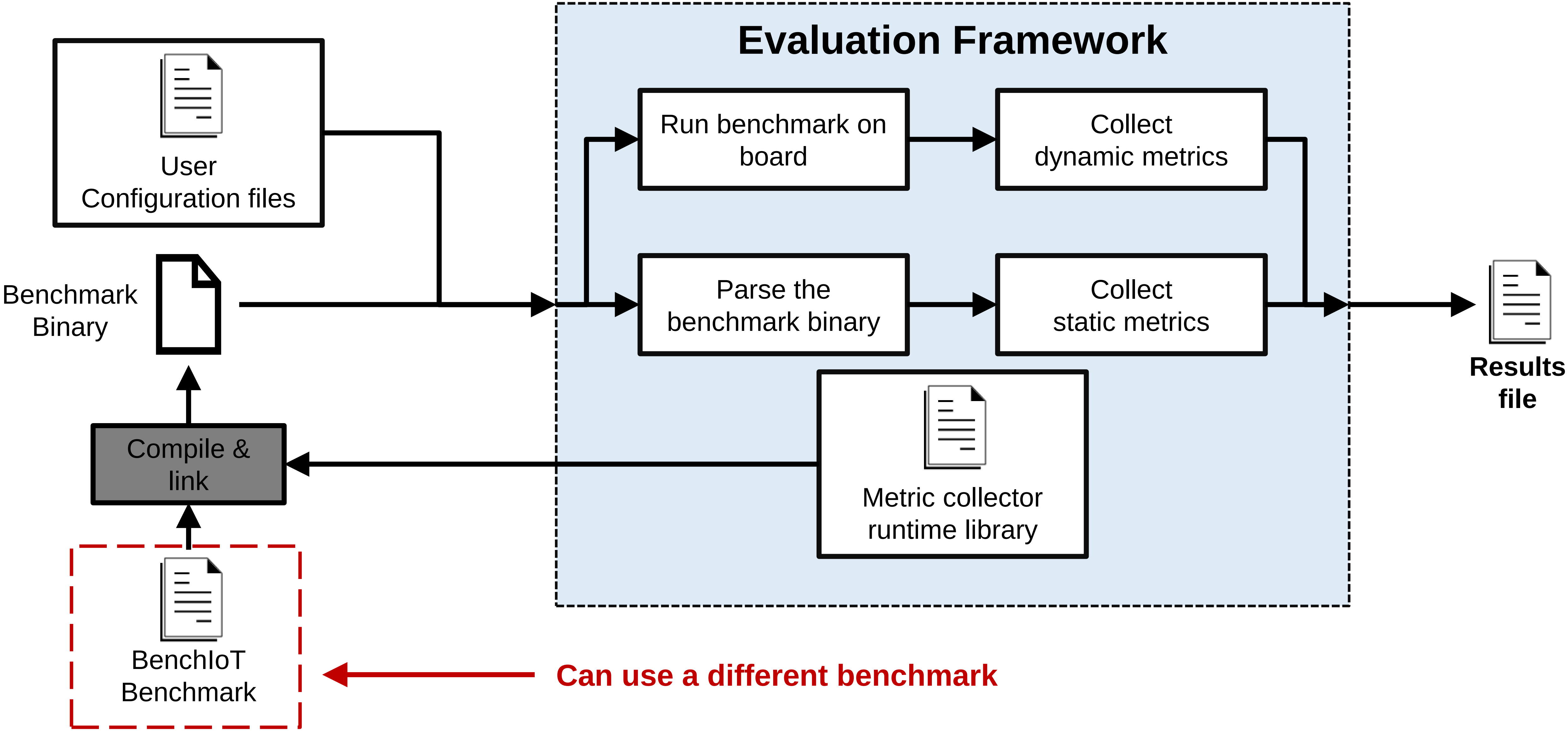
- **Targeted Architecture: ARMv7-M (Cortex-M3,4, and 7 processors).**

Comparison Between BenchIoT and Other Benchmarks

Benchmark	Task Type			Network Connectivity	Peripherals
	Sense	Compute	Actuate		
BEEBS [2]		✓			
Dhrystone [1]		✓			
CoreMark [3]		✓			
IoTMark [4]	✓	✓		Partially (Bluetooth only)	Only I^{2}C
SecureMark [5]		✓			
BenchIoT	✓	✓	✓	✓	✓

[1] R. P. Weicker, “Dhrystone: a synthetic systems programming benchmark,” Communications of the ACM, vol. 27, no. 10, pp. 1013–1030, 1984
[2] J. Pallister, S. J. Hollis, and J. Bennett, “BEEBS: open benchmarks for energy measurements on embedded platforms,” CoRR, vol. abs/1308.5174, 2013.[Online]. Available: http://arxiv.org/abs/1308.5174
[3] EEMBC, “Coremark - industry-standard benchmarks for embedded systems,” http://www.eembc.org/coremark
[4] EEMBC, “Coremark - industry-standard benchmarks for embedded systems,” http://www.eembc.org/iotmark
[5] EEMBC, “Coremark - industry-standard benchmarks for embedded systems,” http://www.eembc.org/ securemark

BenchIoT: Overview

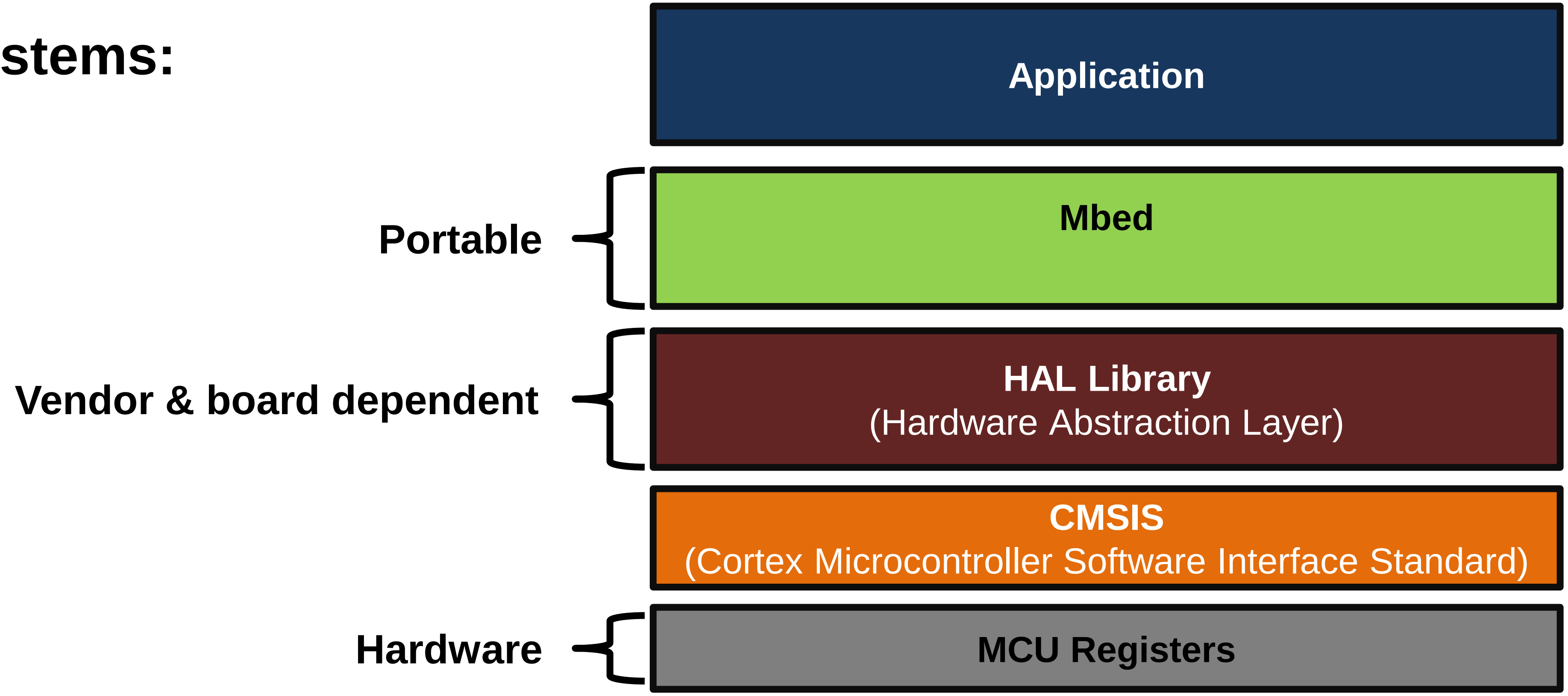


Can use a different benchmark

BenchIoT Design Feature: (1) Hardware agnostic

- **Applications often depend on the underlying vendor & board.**
  - Memory is mapped differently on each board.
  - Peripherals are different across boards.

- **For Operating systems:**
  - Mbed OS(C++)



BenchIoT Design Feature: (2) Reproducibility

- **Applications are event driven.**
  - Example: User enters a pin.
  - Problem: This is inconsistent (e.g., variable timing).

- **Solution: Trigger interrupt from software.**
  - Creates deterministic timing.
  - Allows controlling the benchmarking execution.

BenchIoT Design Feature: (2) Reproducibility

Normal application

```
/* Pseudocode */
void benchmark(void){
    do_some_computation();
    ...
    ...
    wait_for_user_input();
    read_user_input();
    ...

}
```

This is not deterministic

BenchIoT

Deterministic

```
/* Pseudocode */
void benchmark(void){
    do_some_computation();
    ...
    ...
    trigger_interrupt();
    ...
    read_user_input();
    ...

}
```

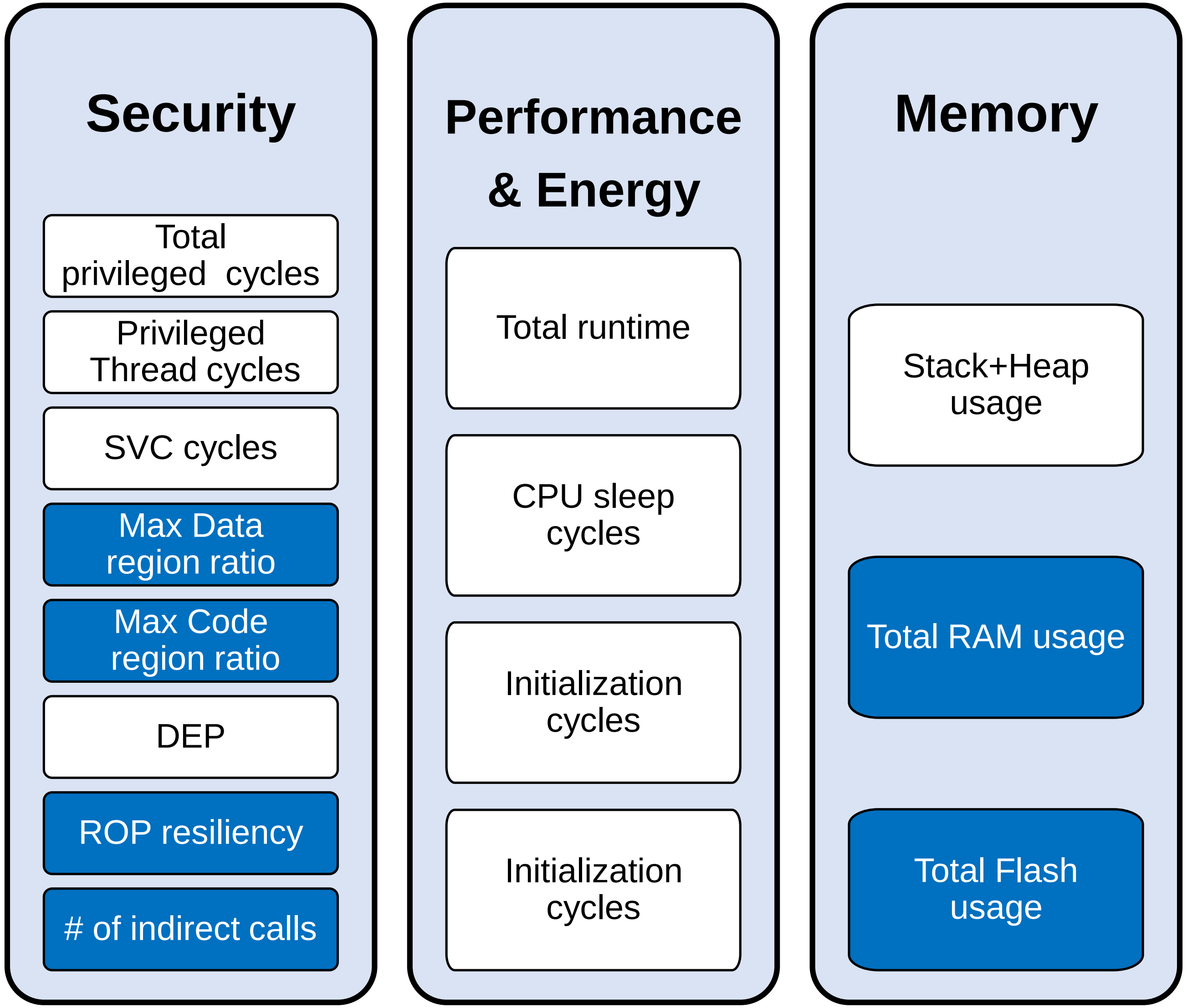
BenchIoT Design Feature: (3) Metrics

- **Allows for measurement of 4 classes of metrics: Security, performance, energy, and memory.**

BenchIoT Design Feature: (3) Metrics

🟦 : Static metric

□ : Dynamic metric



Security

- Total privileged cycles
- Privileged Thread cycles
- SVC cycles
- Max Data region ratio
- Max Code region ratio
- DEP
- ROP resiliency
- # of indirect calls

Performance & Energy

- Total runtime
- CPU sleep cycles
- Initialization cycles
- Initialization cycles

Memory

- Stack+Heap usage
- Total RAM usage
- Total Flash usage

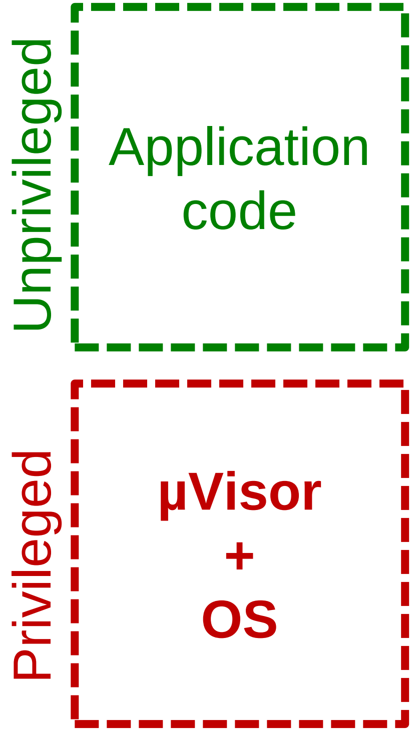
Set of Benchmark Applications

Benchmark	Task Type			Peripheral
	Sense	Compute	Actuate	
Smart Light	✓	✓	✓	Low-power Timer, GPIO, Real-time clock
Smart Thermostat	✓	✓	✓	ADC, Display, GPIO, uSD card
Smart Locker		✓	✓	Serial (UART),Display, uSD Card , Real-time clock
Firmware Updater		✓	✓	Flash in-application programming
Connected Display		✓	✓	Display, uSD Card

- **Boards without non-common peripherals can still run the benchmark.**

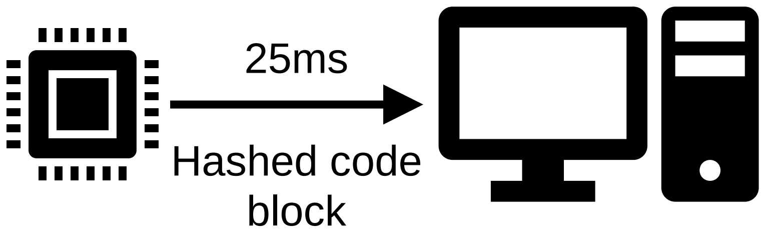
BenchIoT Evaluation: Defense Mechanisms

ARM’s Mbed-µVisor



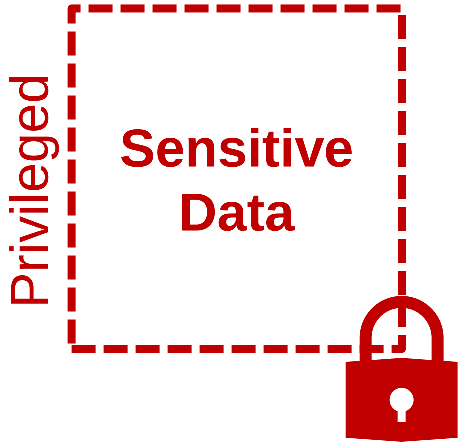
- A hypervisor that enforces the principle of least privilege.

Remote Attestation (RA)



- Verifies the integrity of the code present on the device.
- Uses a real-time task that runs in a separate thread.
- Isolates its code in a secure privileged region.

Data Integrity (DI)



- Isolates sensitive data to a secure privileged region.
- Disables the secure region after the data is accessed.

BenchIoT Evaluation: Defense Mechanisms

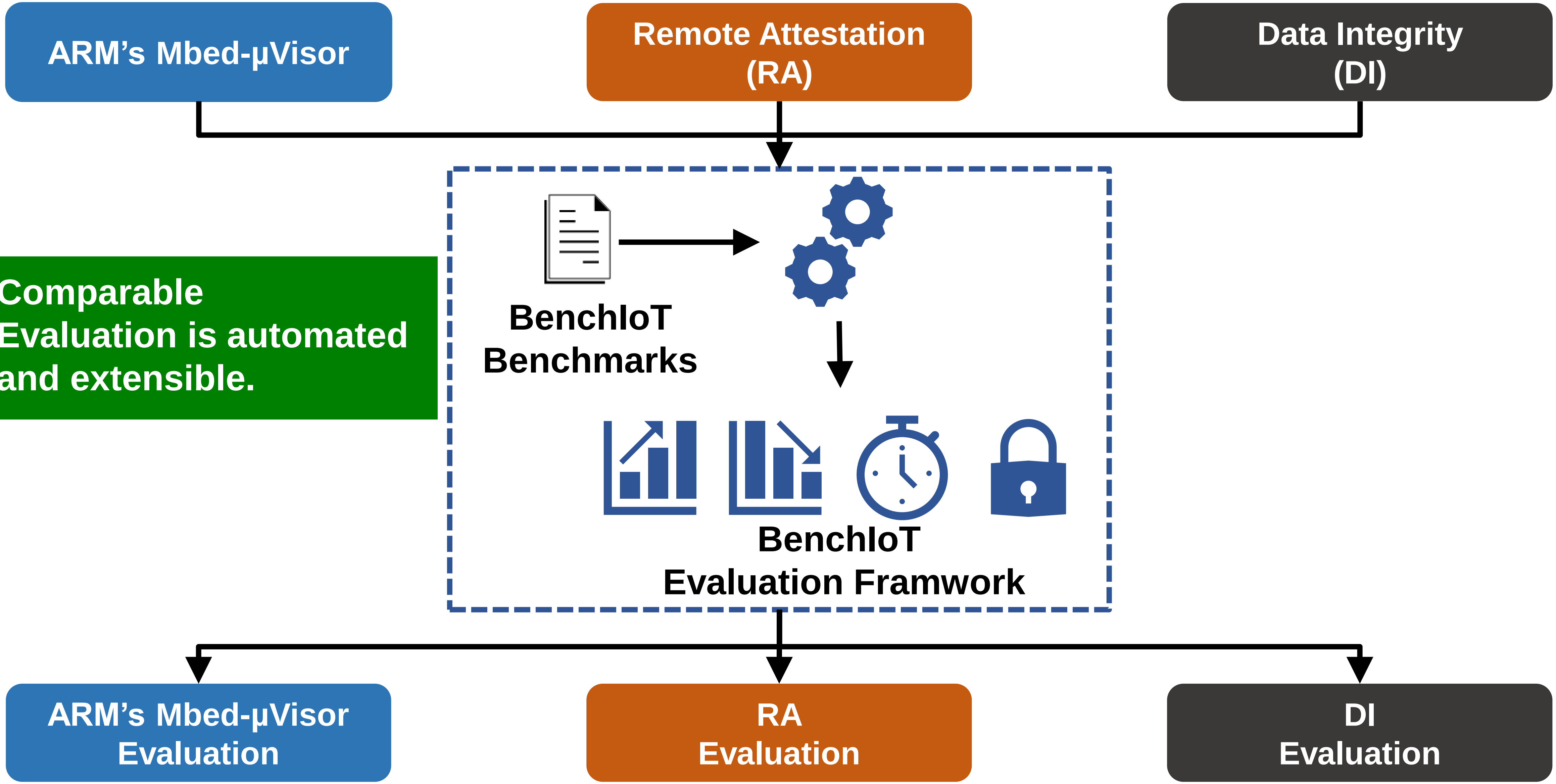
- **The goal is to demonstrate BenchIoT effectiveness in evaluation.**
  - **Non-goal**: To propose a new defense mechanism.

- **ARM’s Mbed-µVisor and Remote Attestation (RA) require an OS.**

- **Data Integrity (DI) is applicable to Bare-Metal (BM) and OS benchmarks.**

BenchIoT Evaluation: Defense Mechanisms

- Comparable
- Evaluation is automated and extensible.



ARM’s Mbed-µVisor

Remote Attestation (RA)

Data Integrity (DI)

BenchIoT Benchmarks

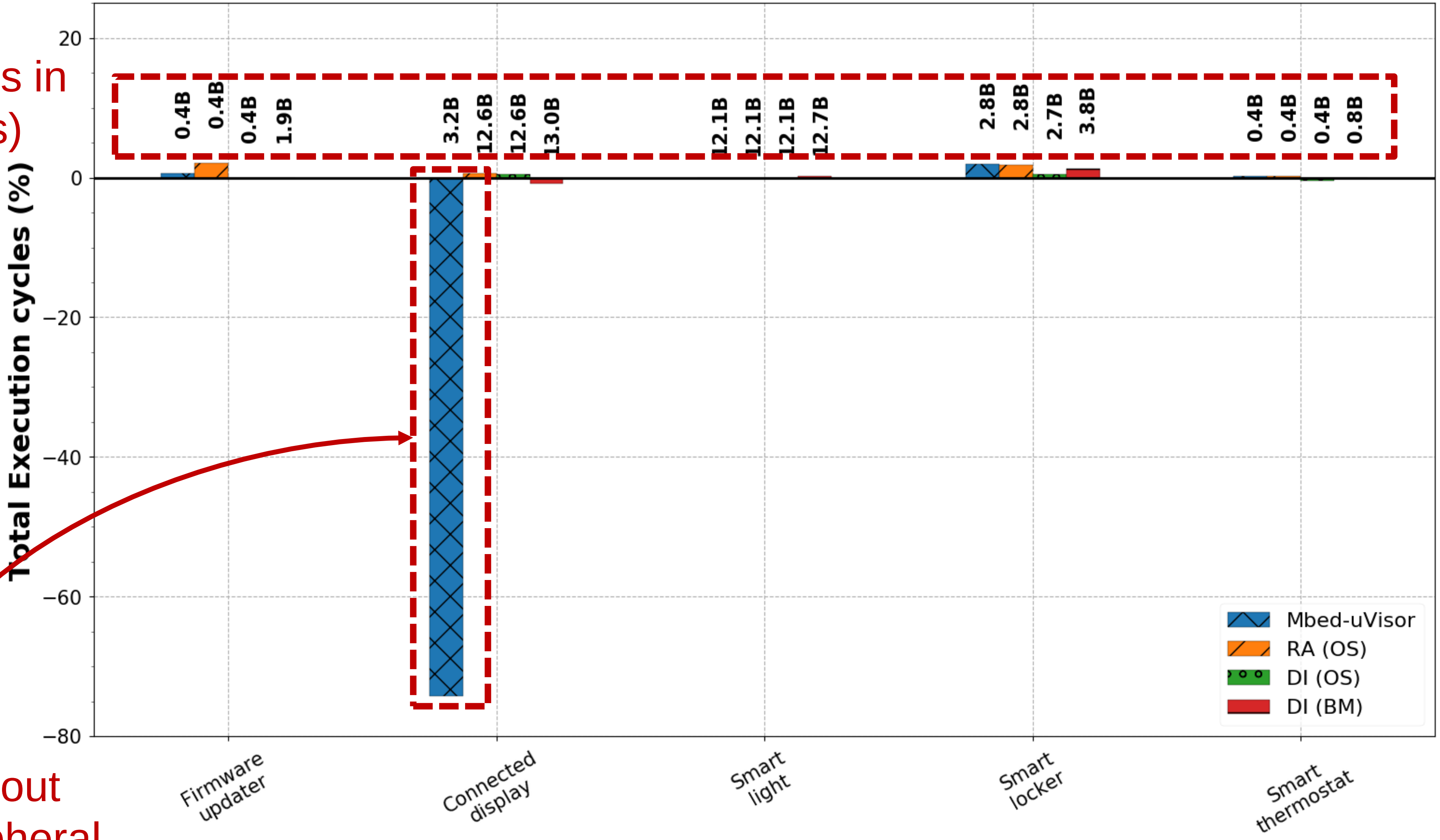
BenchIoT Evaluation Framwork

ARM’s Mbed-µVisor Evaluation

RA Evaluation

DI Evaluation

Performance Results

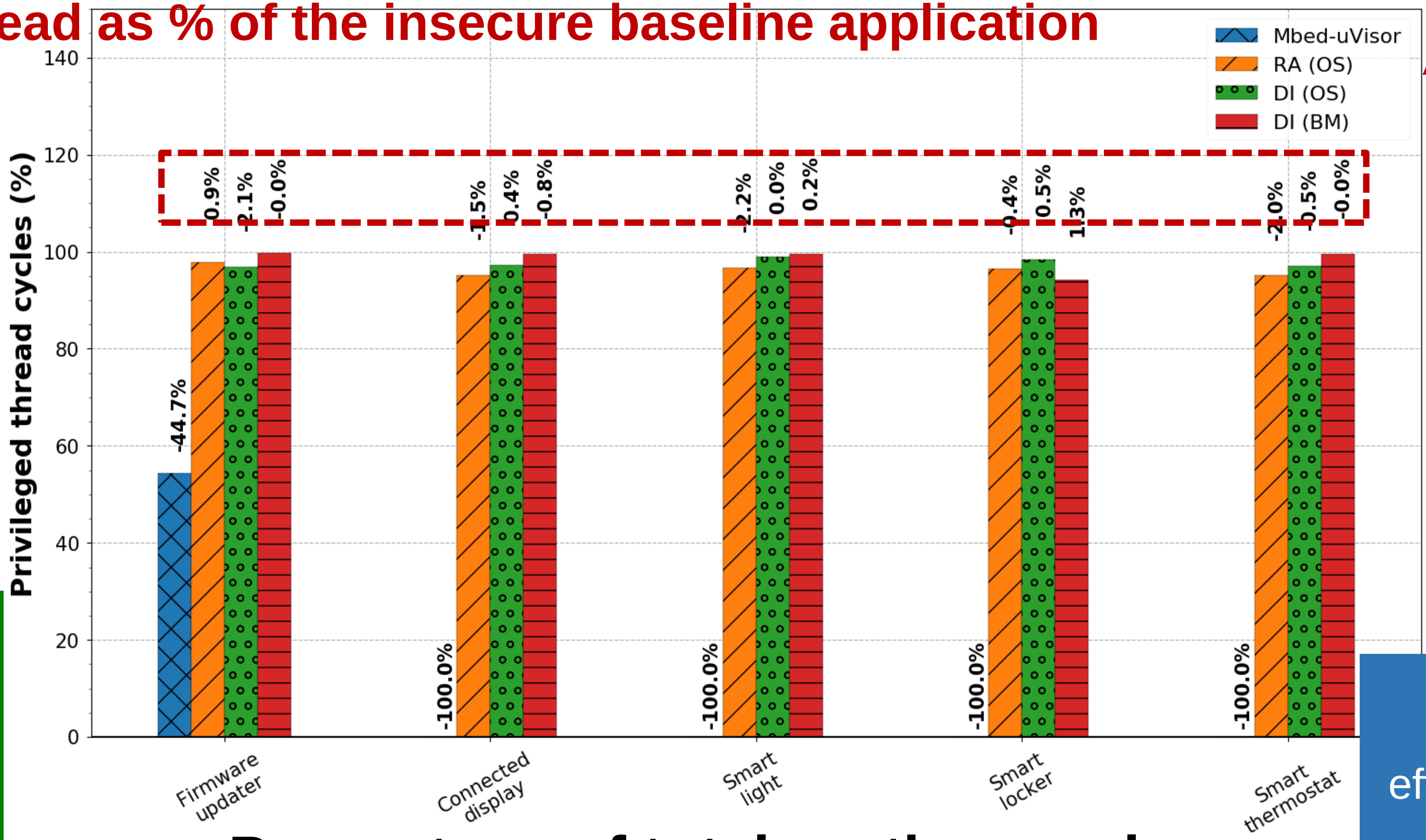


Number of cycles in (Billions/Millions)

Evaluated without the display peripheral

Privileged Execution Minimization Results

- **Overhead as % of the insecure baseline application**



Percentage of total runtime cycles

Almost the entire application runs as privileged for all defenses **Except uVisor**

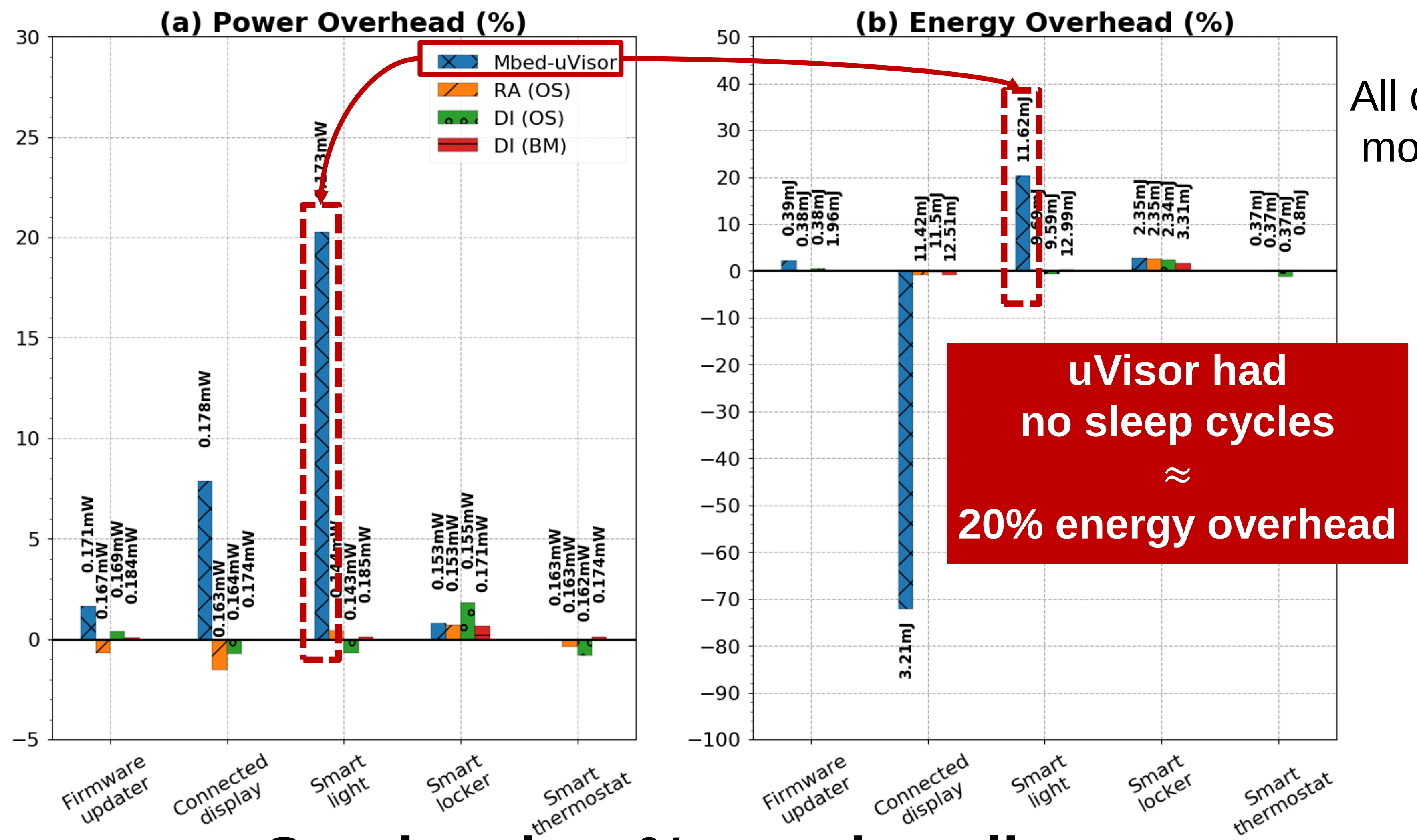
Lower privileged execution → Better Security

uVisor is the most effective defense in reducing privileged execution

Code Injection Evaluation

Defense	Data Execution Prevention (DEP)
Mbed-uVisor	✘ (Heap)
Remote Attestation (OS)	✓
Data Integrity (OS)	✘
Data Integrity (Bare-metal)	✘

Energy Consumption Results

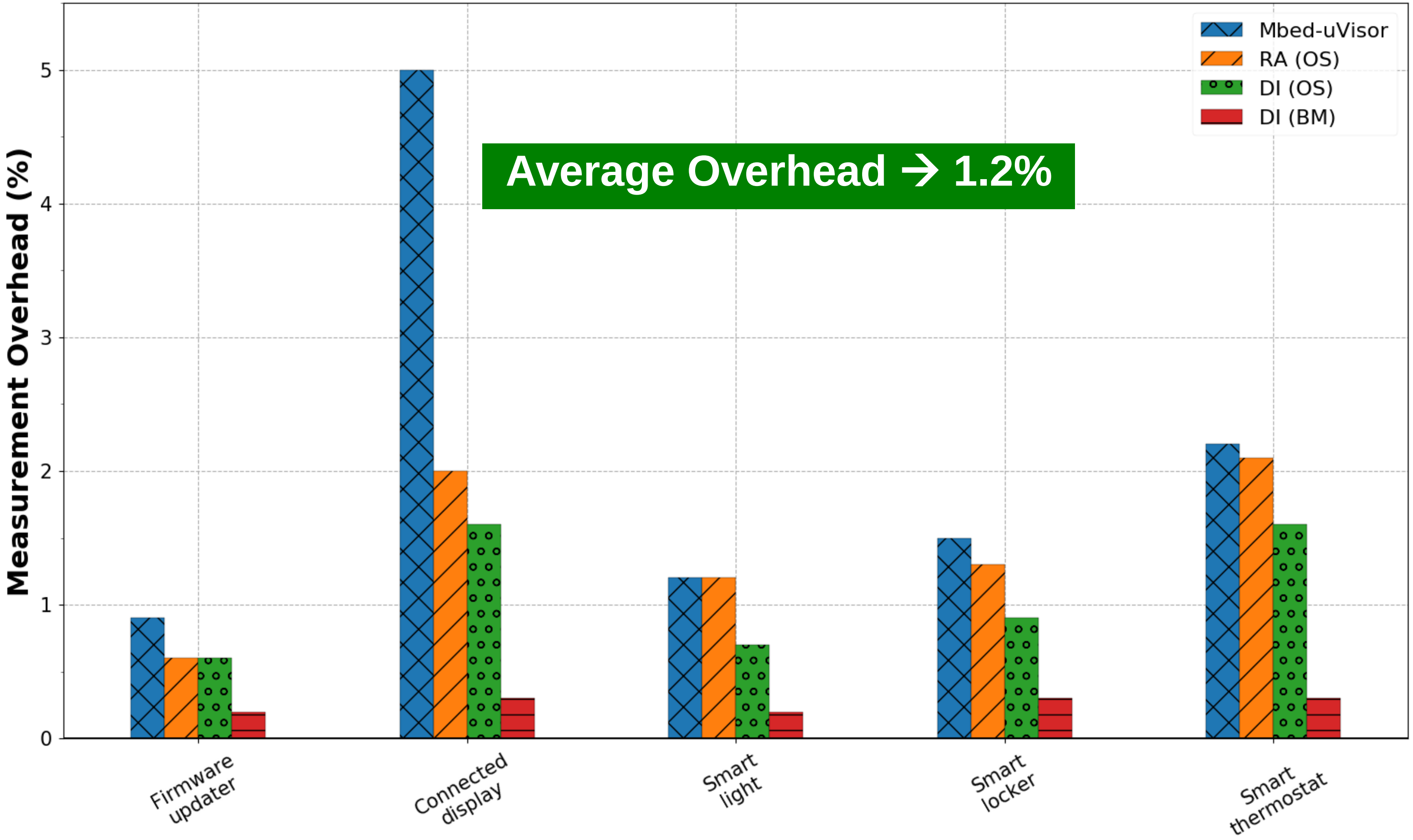


All defenses had modest runtime overhead

uVisor had no sleep cycles ≈ 20% energy overhead

Overhead as % over baseline

Measurement Overhead



Average Overhead → 1.2%

Percentage of total runtime cycles

BenchIoT: Summary

- **Benchmark suite of five realistic IoT applications.**
  - Demonstrates network connectivity, sense, compute, and actuate characteristics.
  - Applies to systems with/without an OS.
- **Evaluation framework:**
  - Covers security, performance, memory usage, and energy consumption.
  - Automated and extensible.
- **Evaluation insights:**
  - Defenses can have similar runtime overhead, but a large difference in energy consumption.
- **Open source:**
  - https://github.com/embedded-sec/BenchIoT