SAND2019-7112C

BenchIoT: A Security Benchmark for The Internet of Things

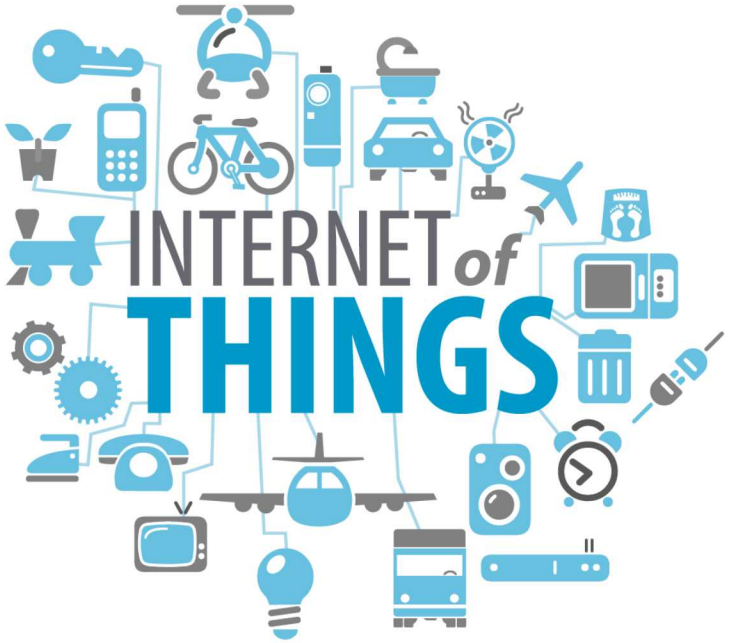
Naif Almakhdhub, Abraham Clements, Mathias Payer, and Saurabh Bagchi

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Internet of Things

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

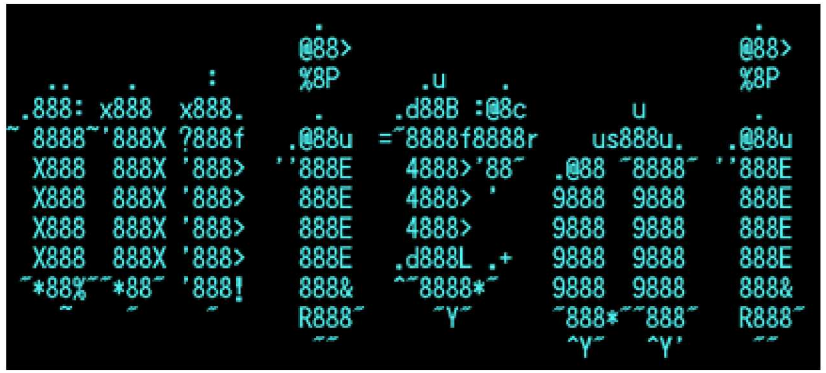
- **Many will be microcontroller based systems (IoT-µCs).**
  - Run directly on the hardware.
  - Can be with/without an OS (bare-metal).
  - Direct access to peripherals and processor.



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

Internet of Things

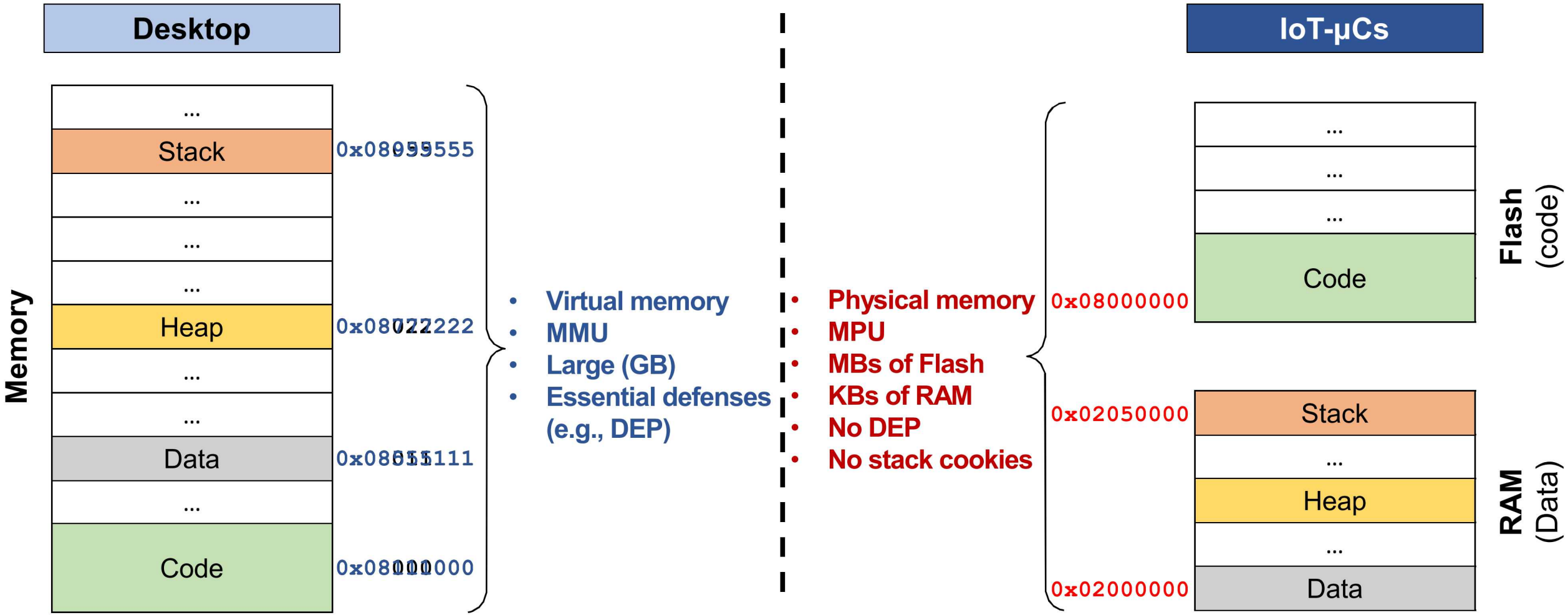
- In 2016, one of the largest DDoS attack to date was caused by IoT devices[1].



- 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

IoT-µCs Challenges



Evaluation in Current IoT Defenses

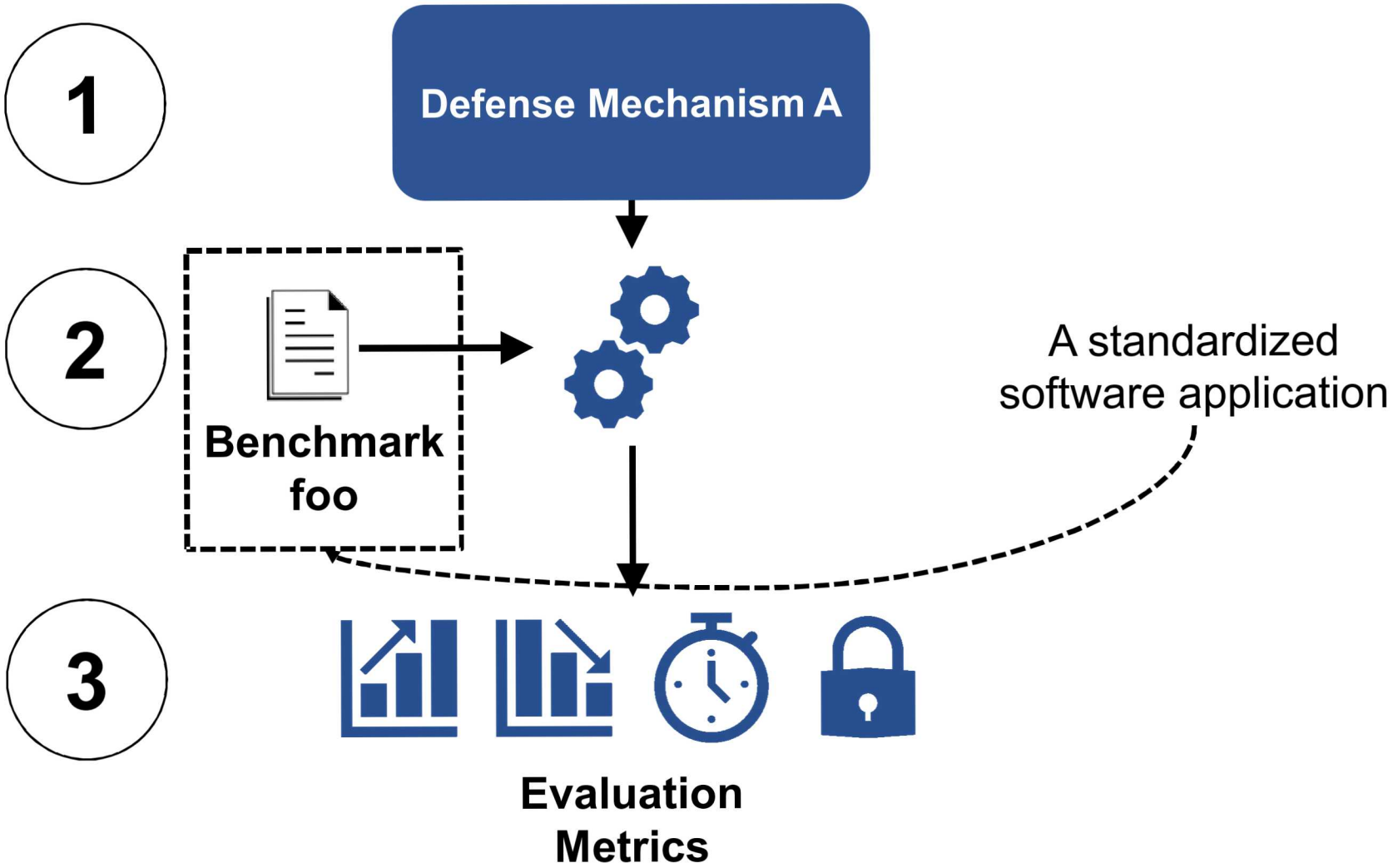
- **Multiple defenses have been proposed.**
  - TyTan[DAC15], TrustLite[EurSys14], C-FLAT [CCS16], nesCheck[AsiaCCS17], SCFP[EuroS&P18], LiteHAX[ICCAD18] CFI CaRE [RAID17], ACES[SEC18], MINION [NDSS18], EPOXY [S&P17]

- **How are they evaluated?**
  - Ad-hoc evaluation.

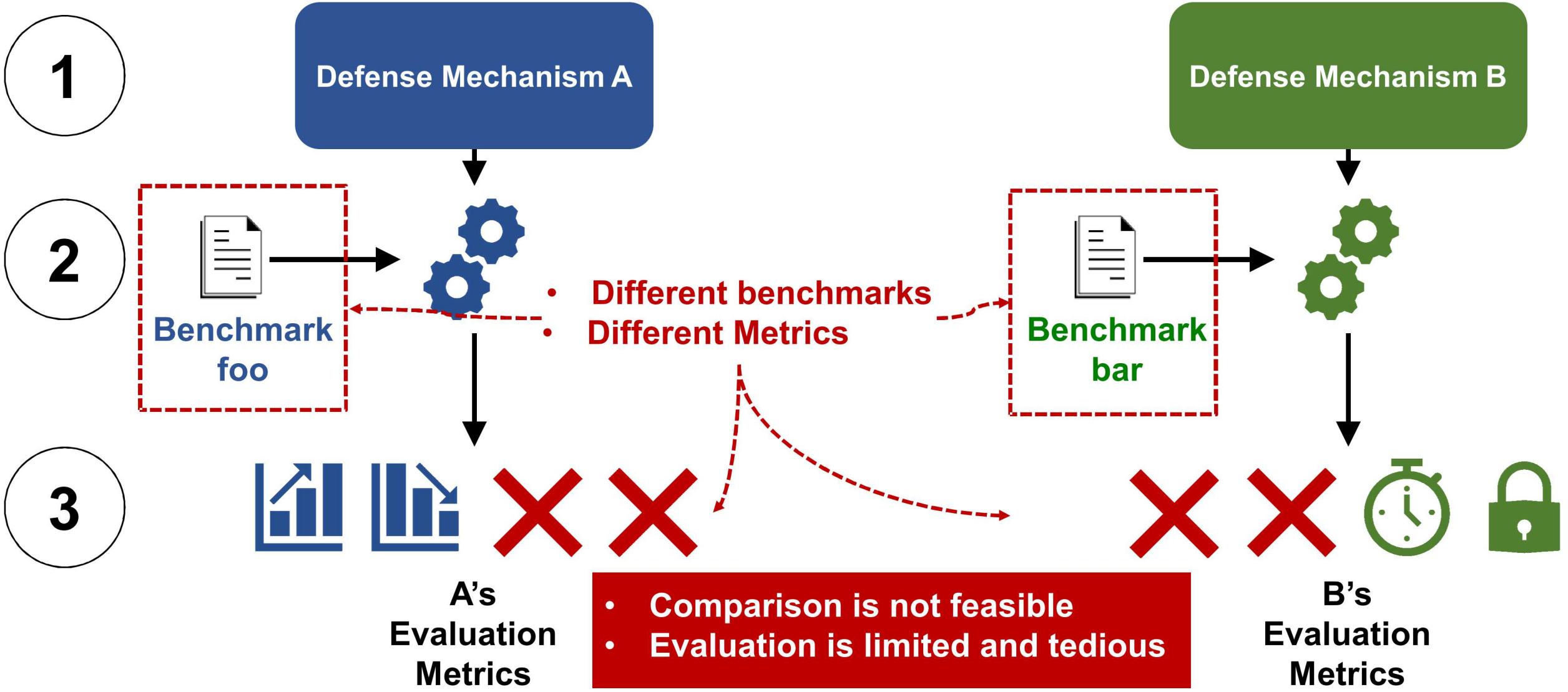
Defense	Evaluation Type	
	Benchmark	Case Study
TyTan		✓
TrustLite		✓
C-FLAT		✓
nesCheck		✓
SCFP	Dhrystone[1]	✓
LiteHAX	CoreMark[2]	✓
CFI CaRE	Dhrystone[1]	✓
ACES		✓
Minion		✓
EPOXY	BEEBS[3]	✓

[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)

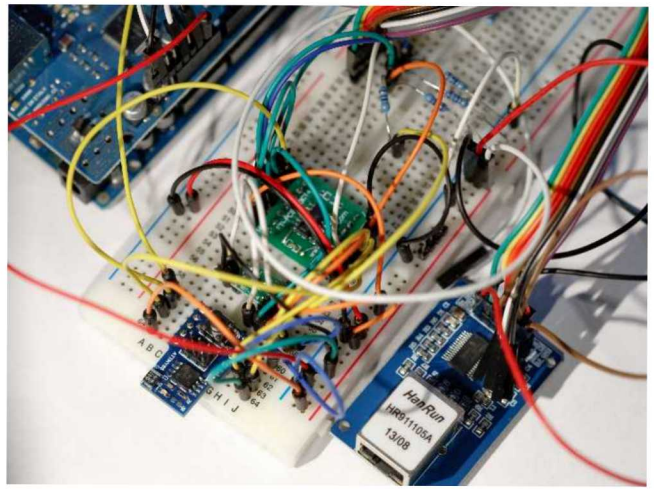


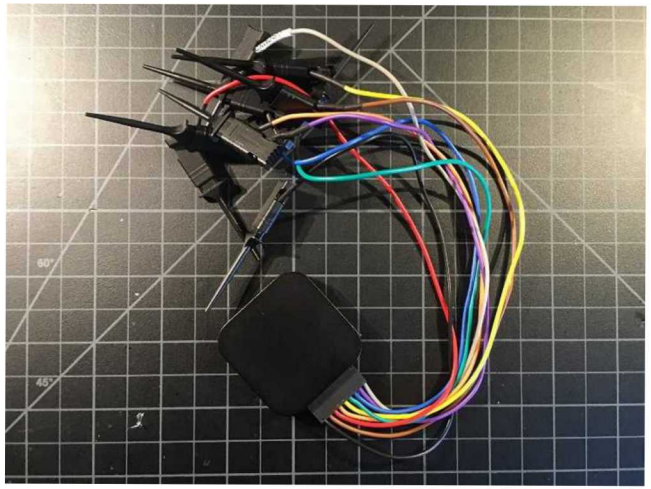
IoT-µCs Evaluation (Reality)

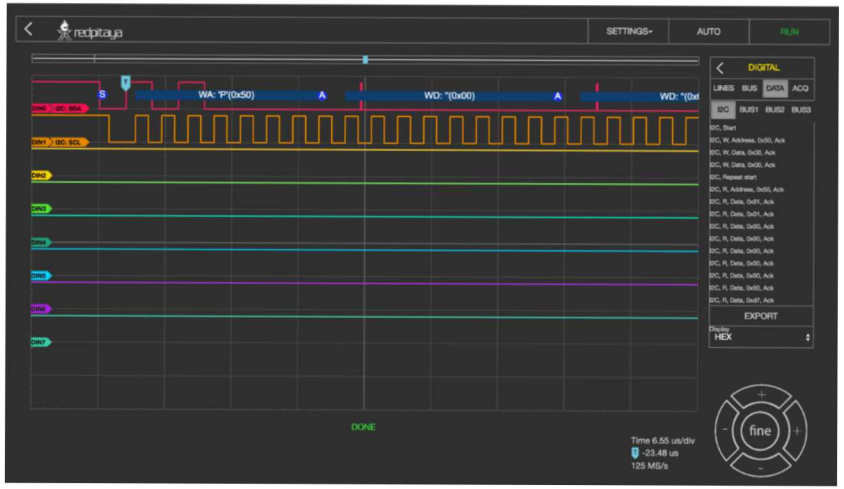


Why not use a benchmark?

- **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.**







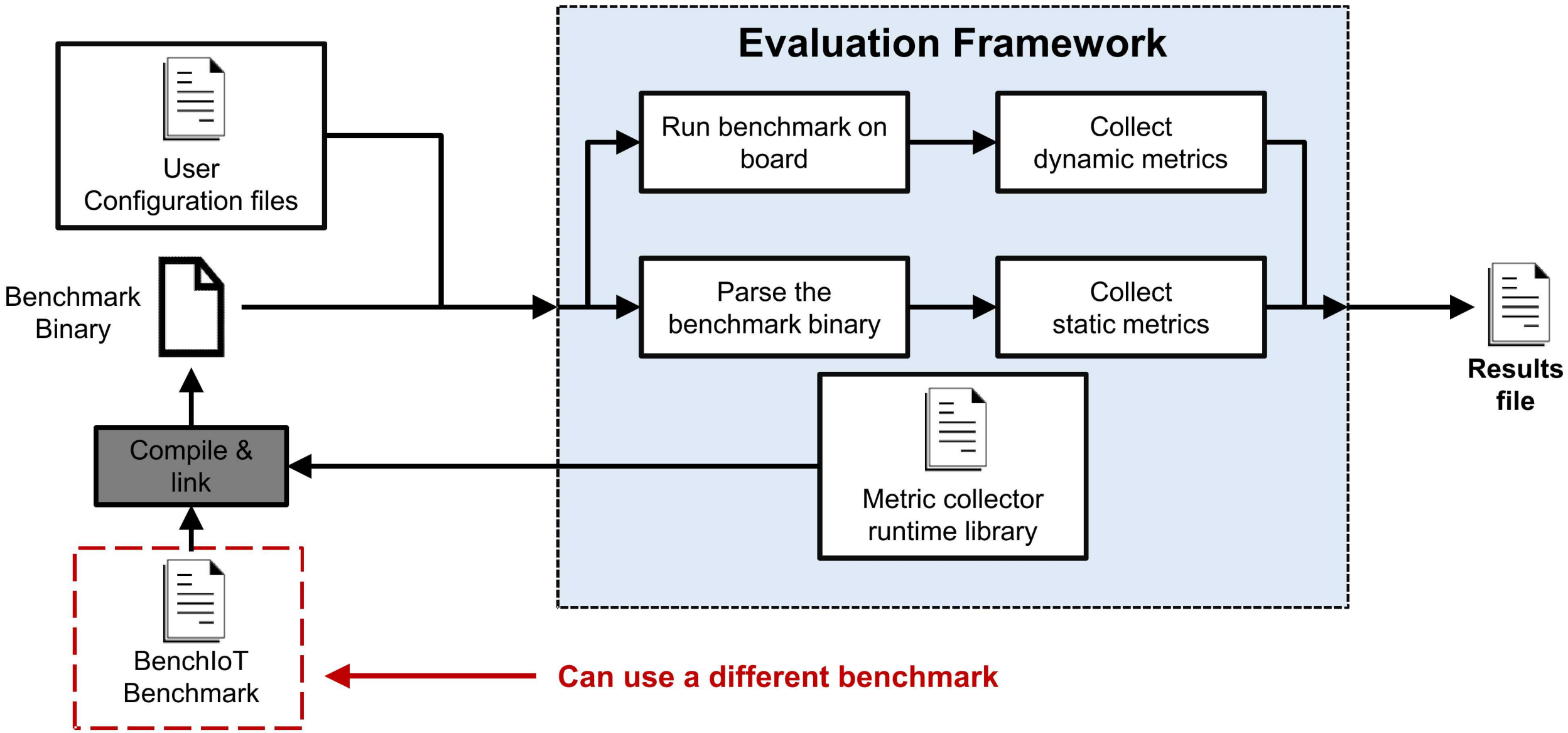
Proposed solution: BenchIoT

- **BenchIoT benchmarks and evaluation framework.**
- **An extensible, portable evaluation framework.**
  - A software based approach.
  - Applicable to other benchmarks.
- **A realistic and portable set of benchmarks.**
  - Works for both with/without an OS.
  - Deterministic execution of external events.
- **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		✓			
Dhrystone		✓			
CoreMark		✓			
IoTMark	✓	✓		Partially (Bluetooth only)	Only I^2C
SecureMark		✓			
BenchIoT	✓	✓	✓	✓	✓

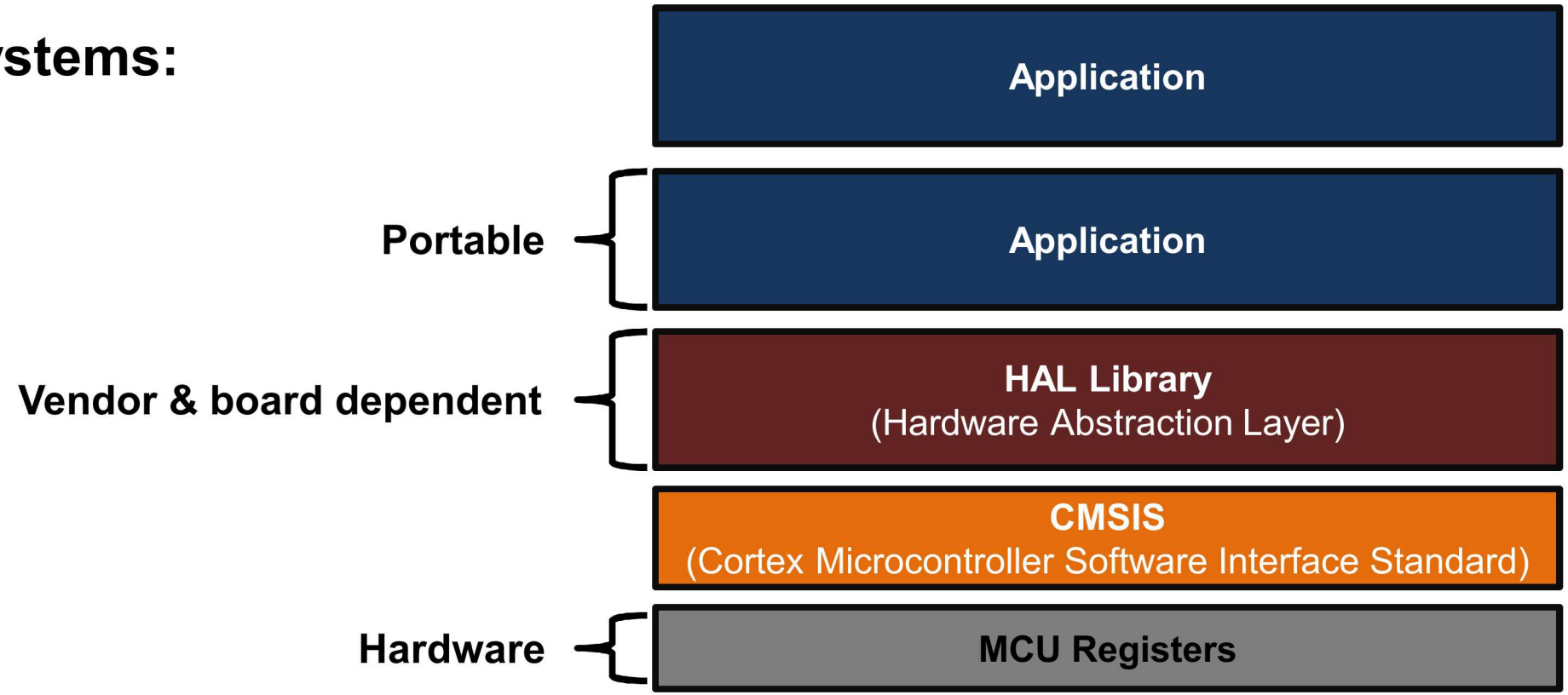
BenchIoT: Overview



BenchIoT design aspects : (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 aspects : (2) Reproducibility

- **Applications are event driven.**
  - Example: User enters a pin.
  - Inconsistent (i.e., timing).

- **Solution: trigger interrupt from software.**
  - Deterministic timing.
  - Allows controlling the benchmarking dataset.

BenchIoT design aspects : (2) Reproducibility

Normal application

```
   /* Pseudocode */
1. void benchmark(void){
2.      do_some_computation();
3.      ...
4.      ...
5.      wait_for_user_input();
6.      read_user_input();
7.      ...
8.
9. }
```

Is this deterministic?

deterministic

BenchIoT

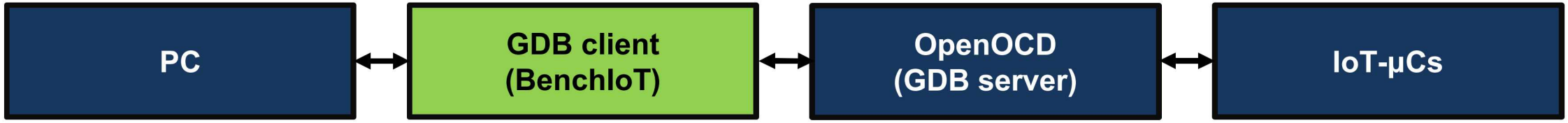
```
   /* Pseudocode */
1. void benchmark(void){
2.      do_some_computation();
3.      ...
4.      ...
5.      trigger_interrupt();
6.      ...
7.      read_user_input();
8.      ...
9.
10.}
```

BenchIoT design aspects : (3) Evaluation

- **Independent form vendor specific hardware.**
  - Software approach.
  - Requirement: Data Watchpoint and Trace unit (DWT).

- **Automated, finer grained metrics.**

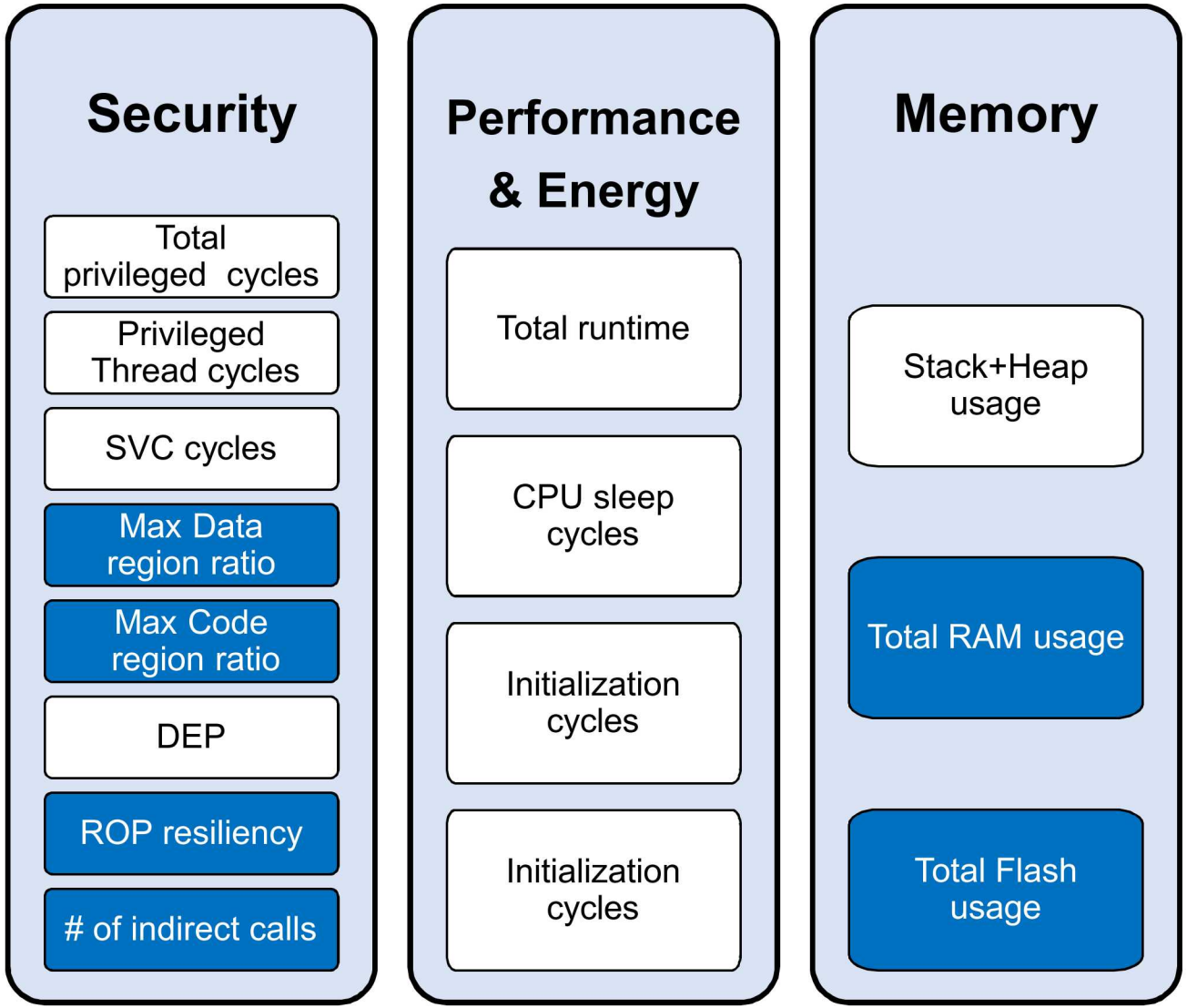
- **Evaluation methods:**
  - Collection of scripts for static metrics.
  - A statically linked runtime library: Metric collector → dynamic.
  - A debugger (i.e., GDB with python scripting) → dynamic.



BenchIoT design aspects : (4) Metrics

: Static metric

: Dynamic metric



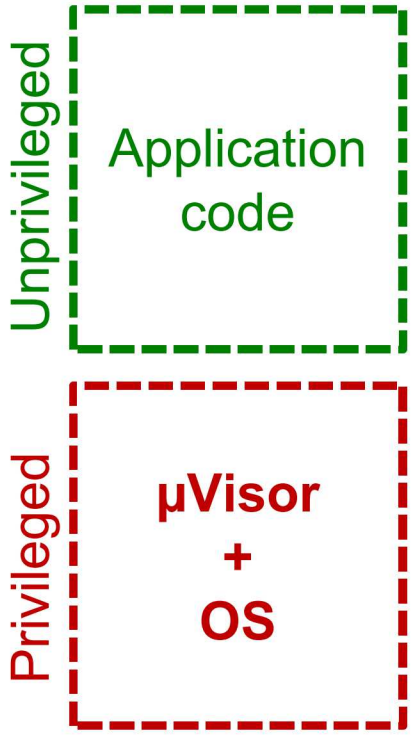
BenchIoT design aspects : (5) Benchmarks

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.**

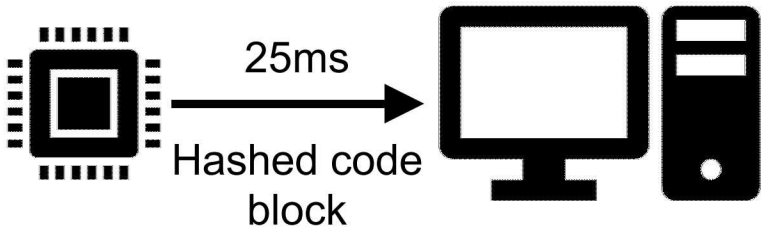
BenchIoT Evaluation: Defense Mechanisms

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.

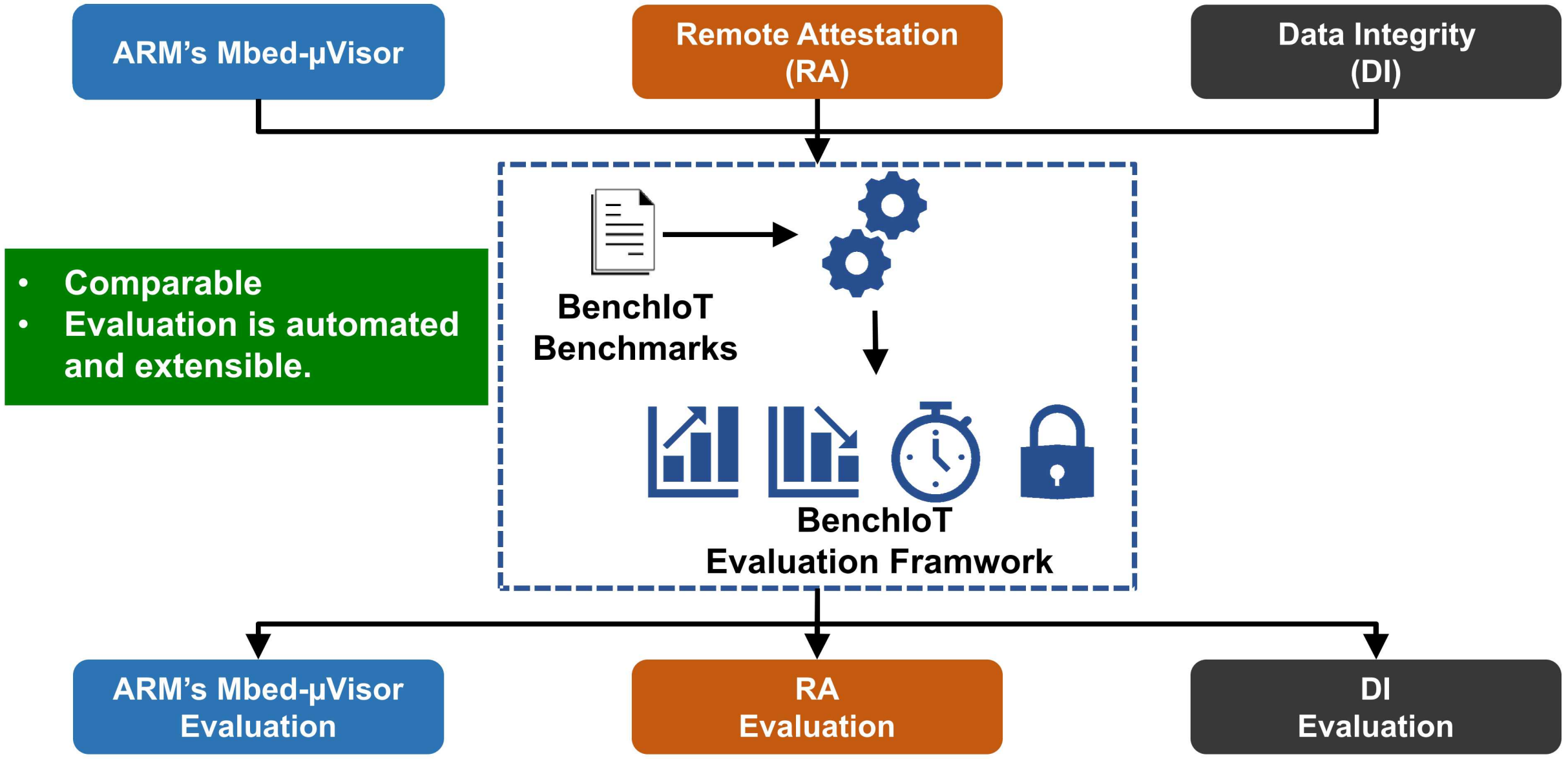
BenchIoT Evaluation: Defense Mechanisms

- **The goal is to demonstrate BenchIoT effectiveness in evaluation.**
  - It **not** the goal to propose a new defense mechanism.

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

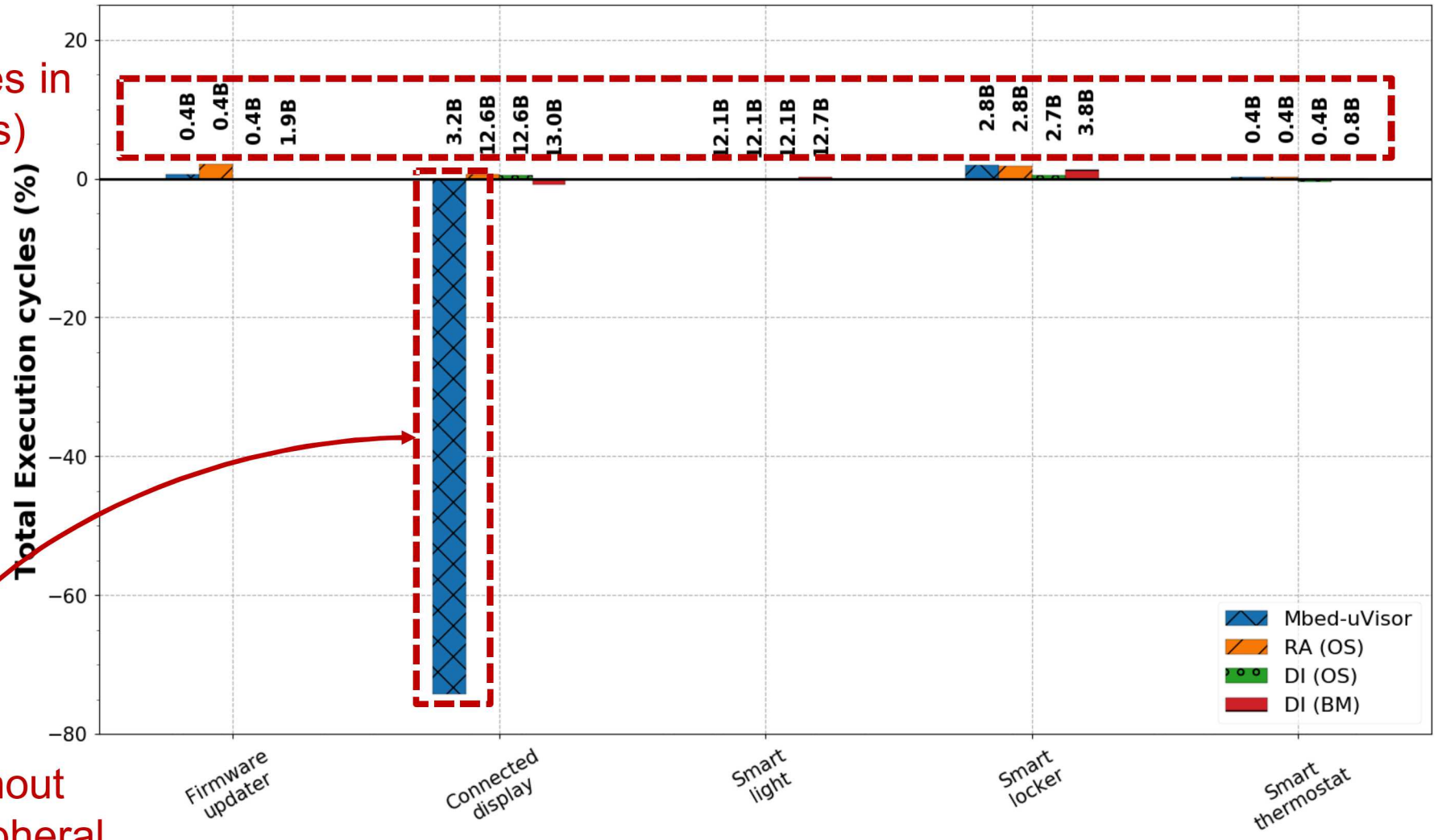
- **Data Integrity (DI) is applicable for Bare-Metal (BM) and OS benchmarks.**

BenchIoT Evaluation: Defense Mechanisms



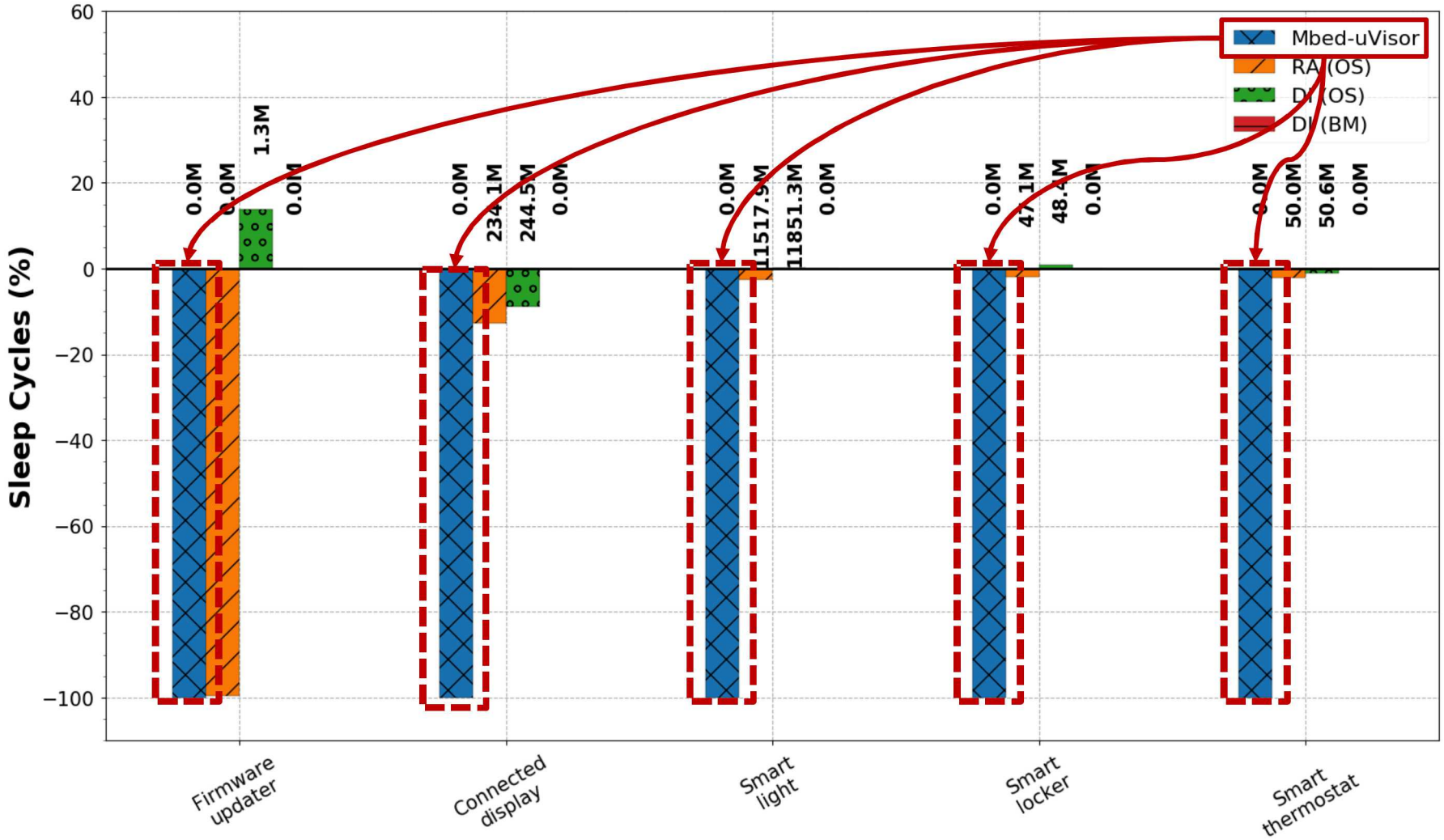
Performance Results

Number of cycles in (Billions/Millions)



Evaluated without the display peripheral

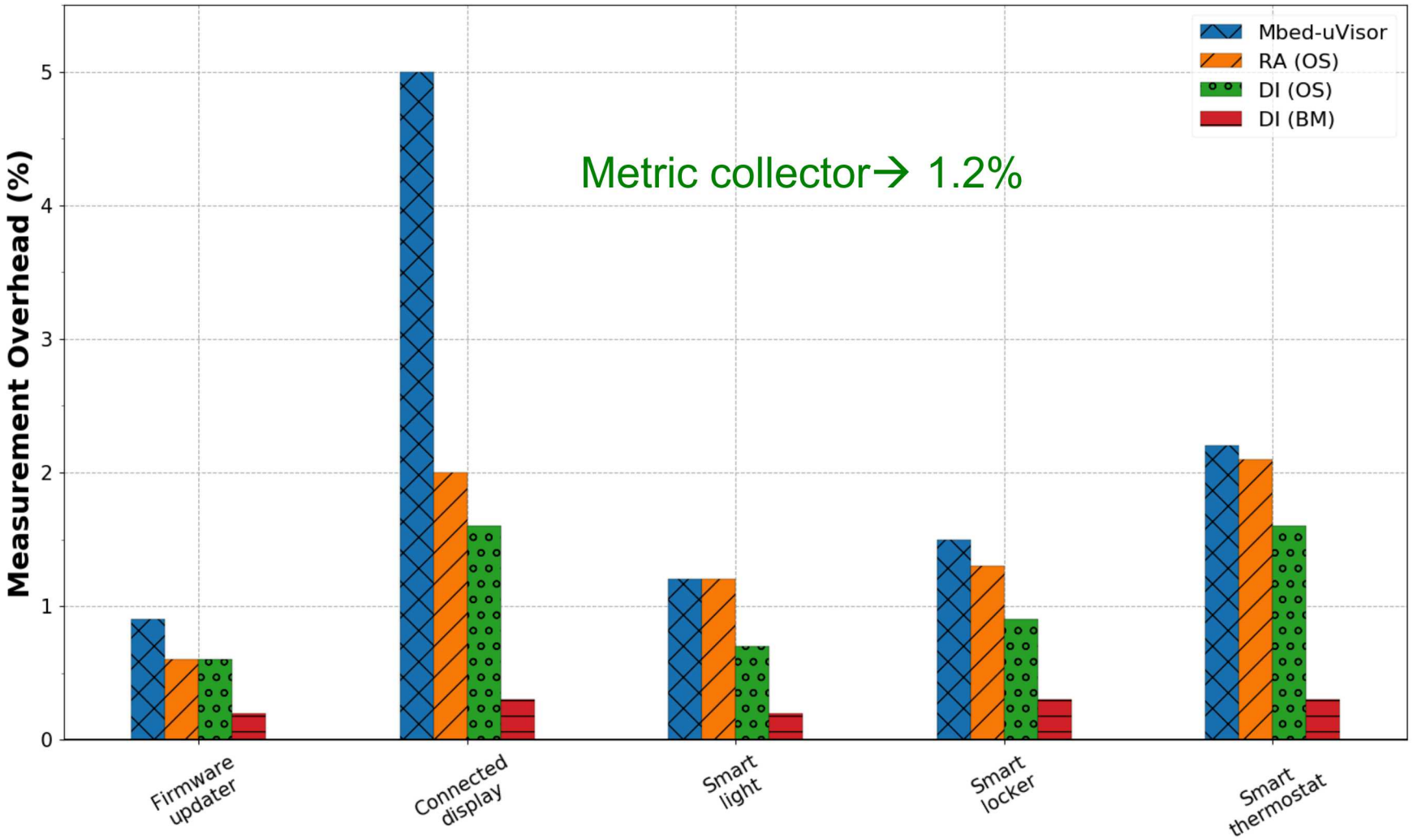
Sleep cycles Results



uVisor disables sleep cycles → adverse effect on energy

Percentage of total runtime cycles

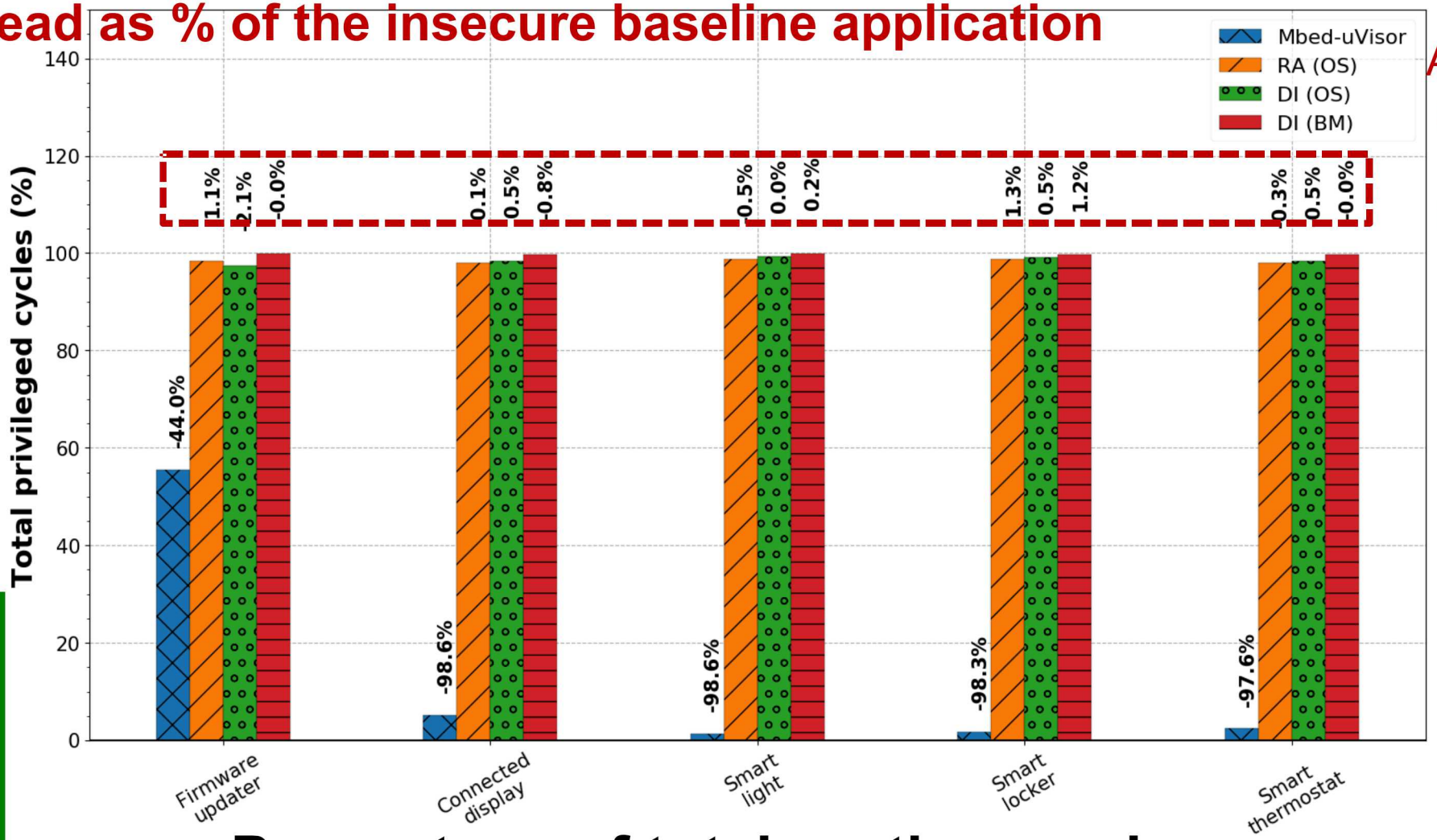
Measurement Overhead



Percentage of total runtime cycles

Privileged Execution Minimization Results

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

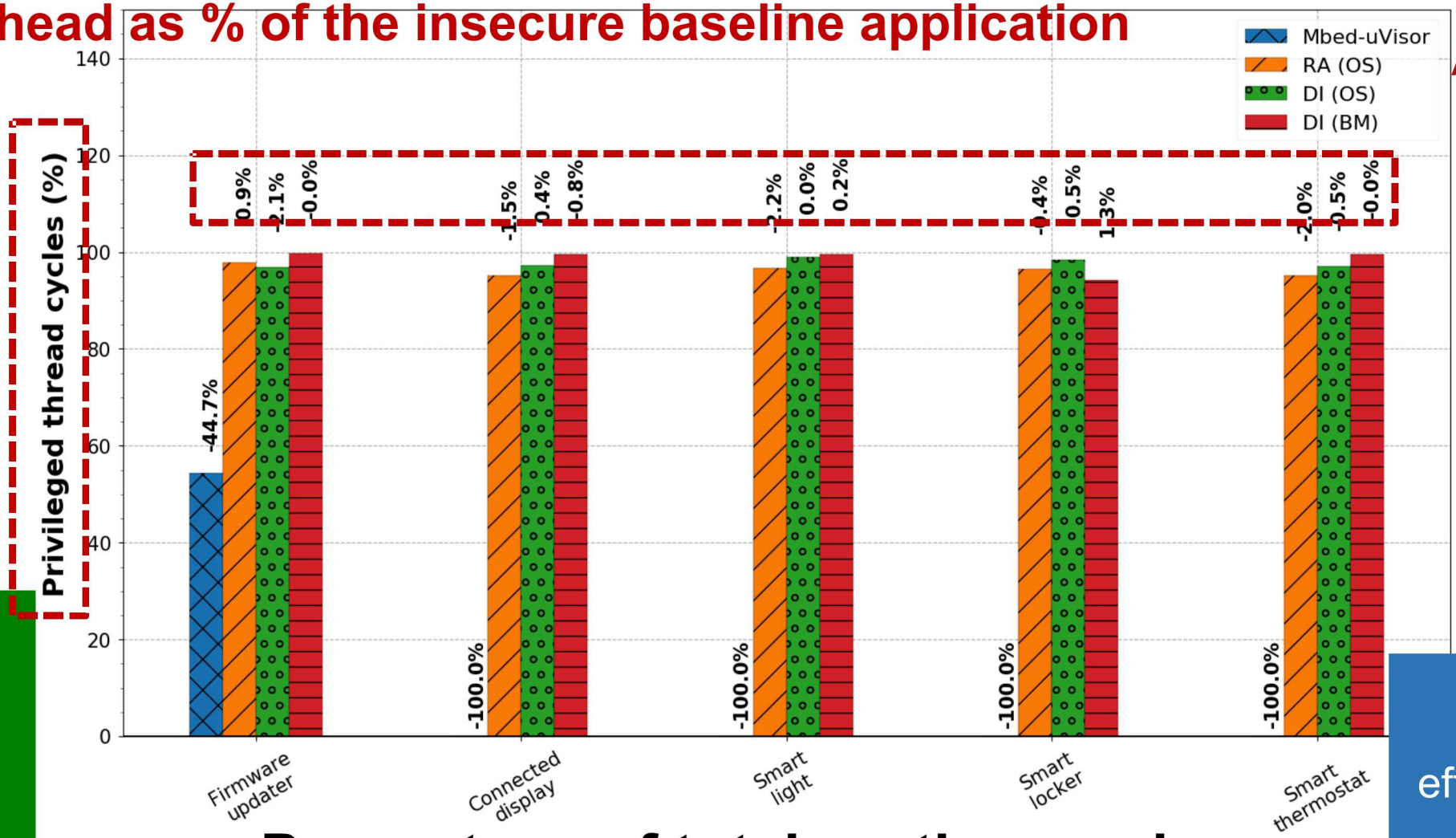


Almost the entire application runs as privileged for all defenses **Except uVisor**

Lower privileged execution → Better Security

Privileged Execution Minimization Results

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



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

Lower privileged execution → Better Security

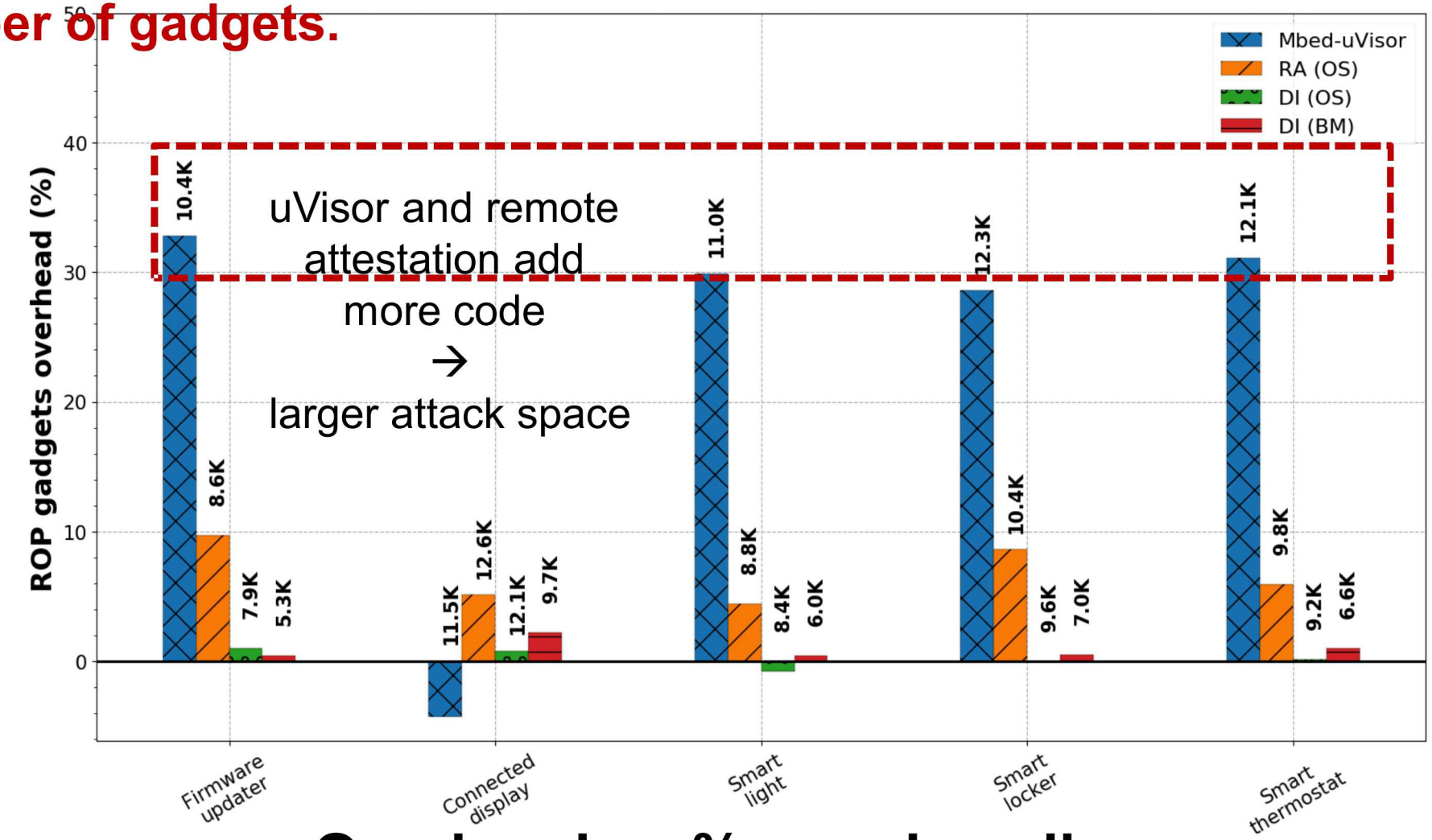
uVisor is the most effective defense in reducing privileged execution

Memory Isolation and Code Injection Results

Defense	Memory Isolation and Code injection Metrics		
	Max code Reg. ratio	Max data Reg. ratio	DEP
Mbed-uVisor	1.0	1.0	✗
Remote Attestation (OS)	0.99	1.0	✓
Data Integrity (OS)	1.0	0.99	✗
Data Integrity (BM)	1.0	0.99	✗

Code Reuse Protection Results

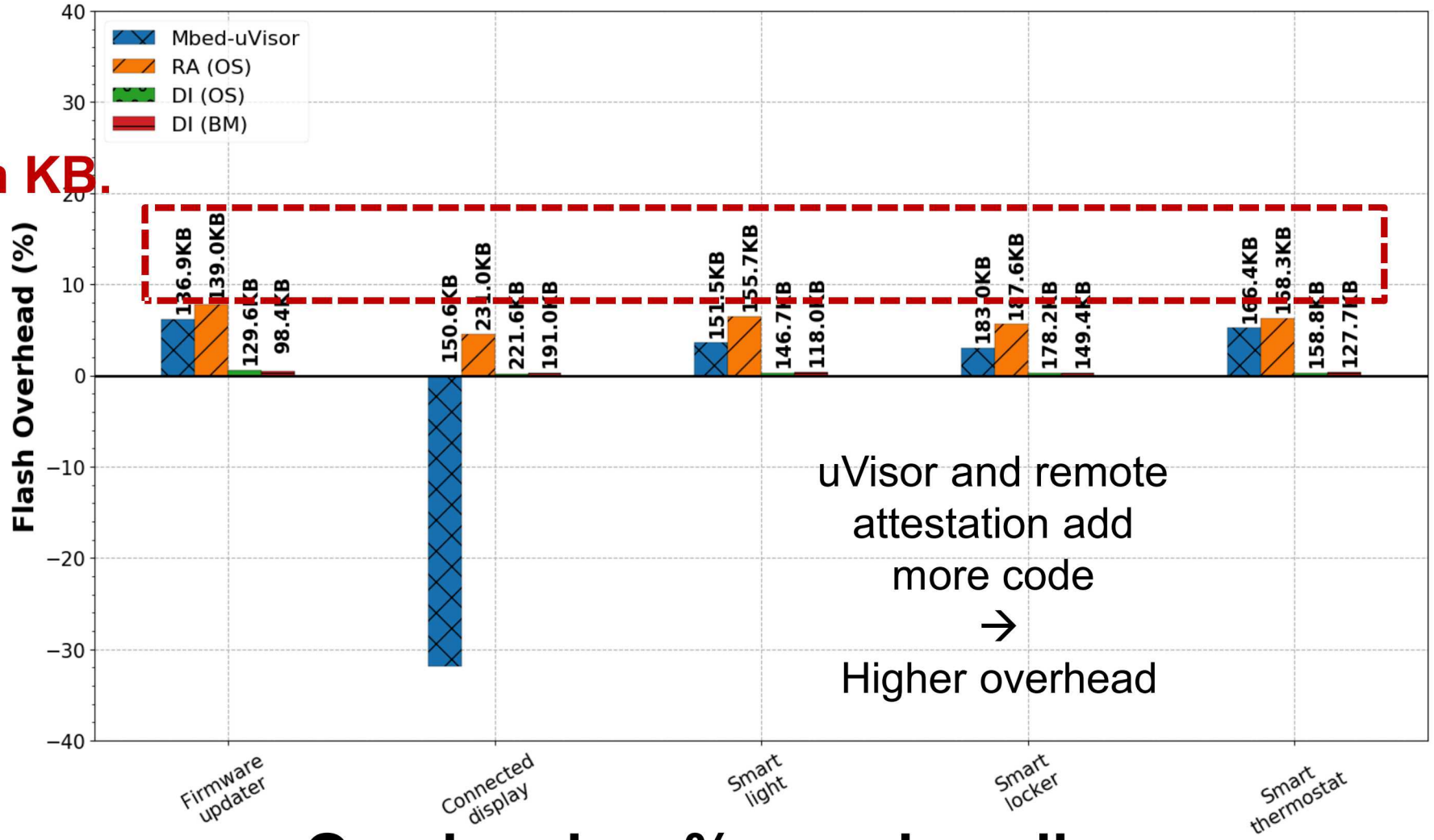
- **Number of gadgets.**



Overhead as % over baseline

Memory Usage Results

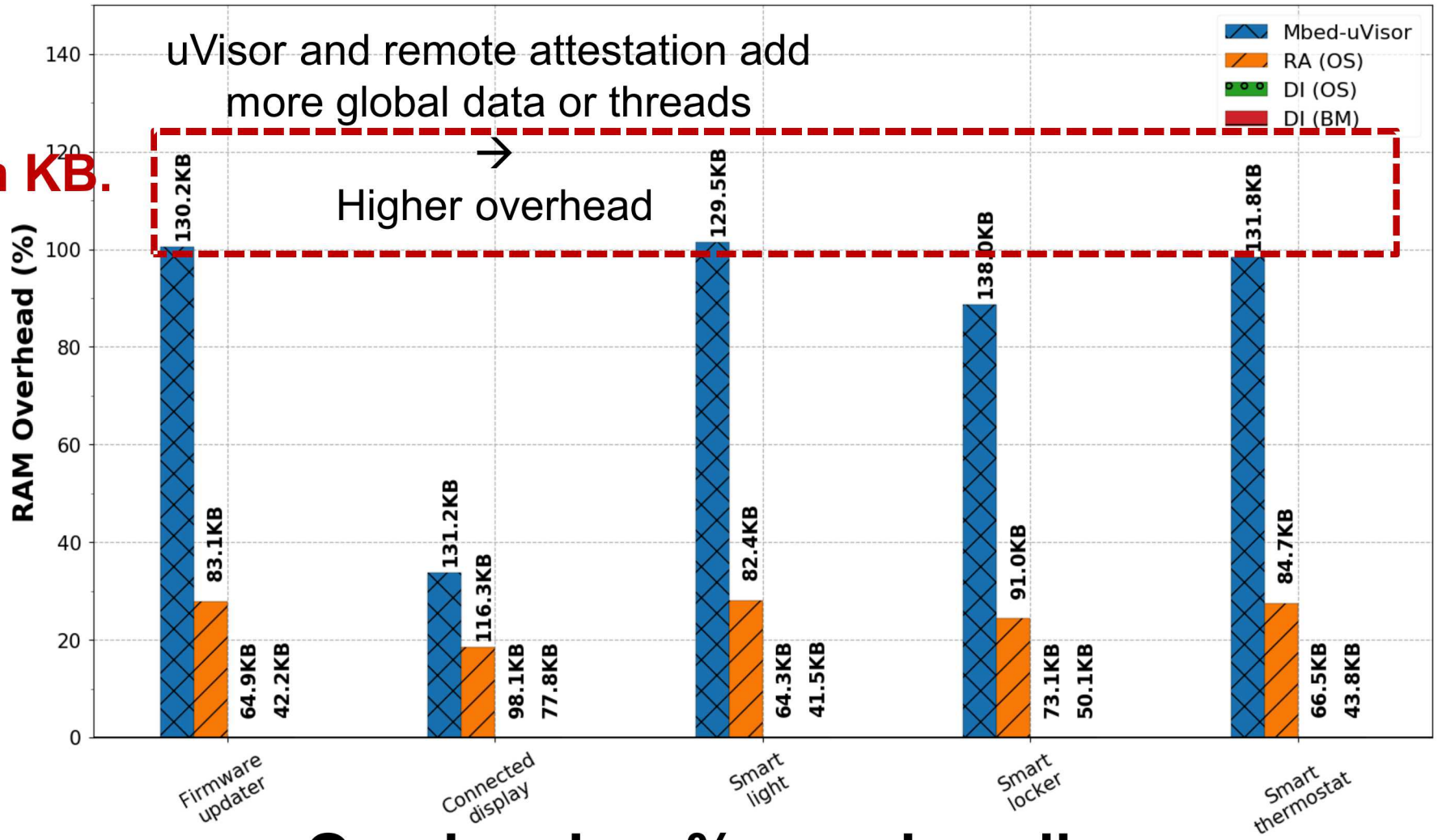
- **Size in KB.**



Overhead as % over baseline

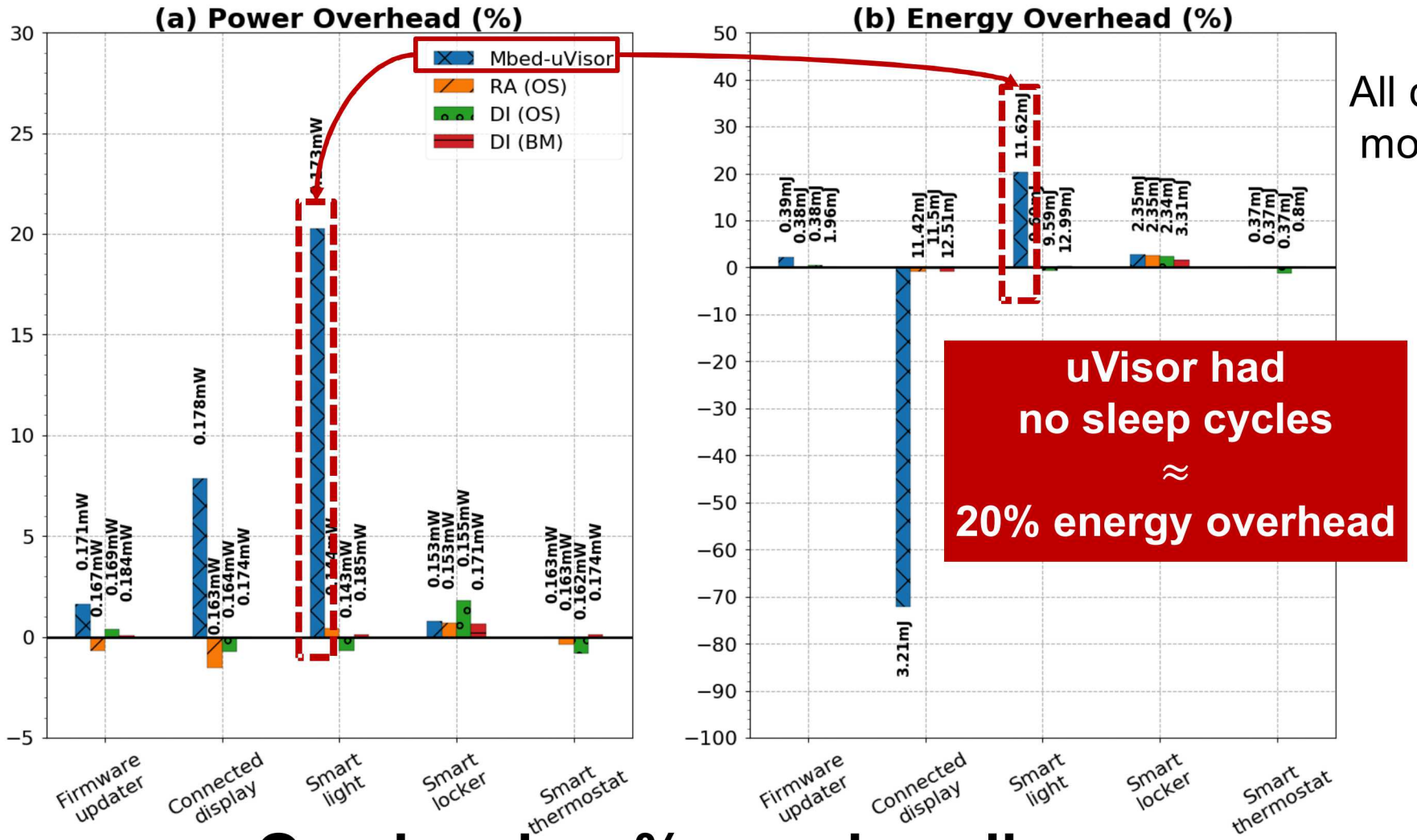
Memory Usage Results

- **Size in KB.**



Overhead as % over baseline

Energy Consumption Results



All defenses had modest runtime overhead

Overhead as % over baseline

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

Thank you!

Backup Slides

Comparison Between BenchIoT and Other Benchmarks

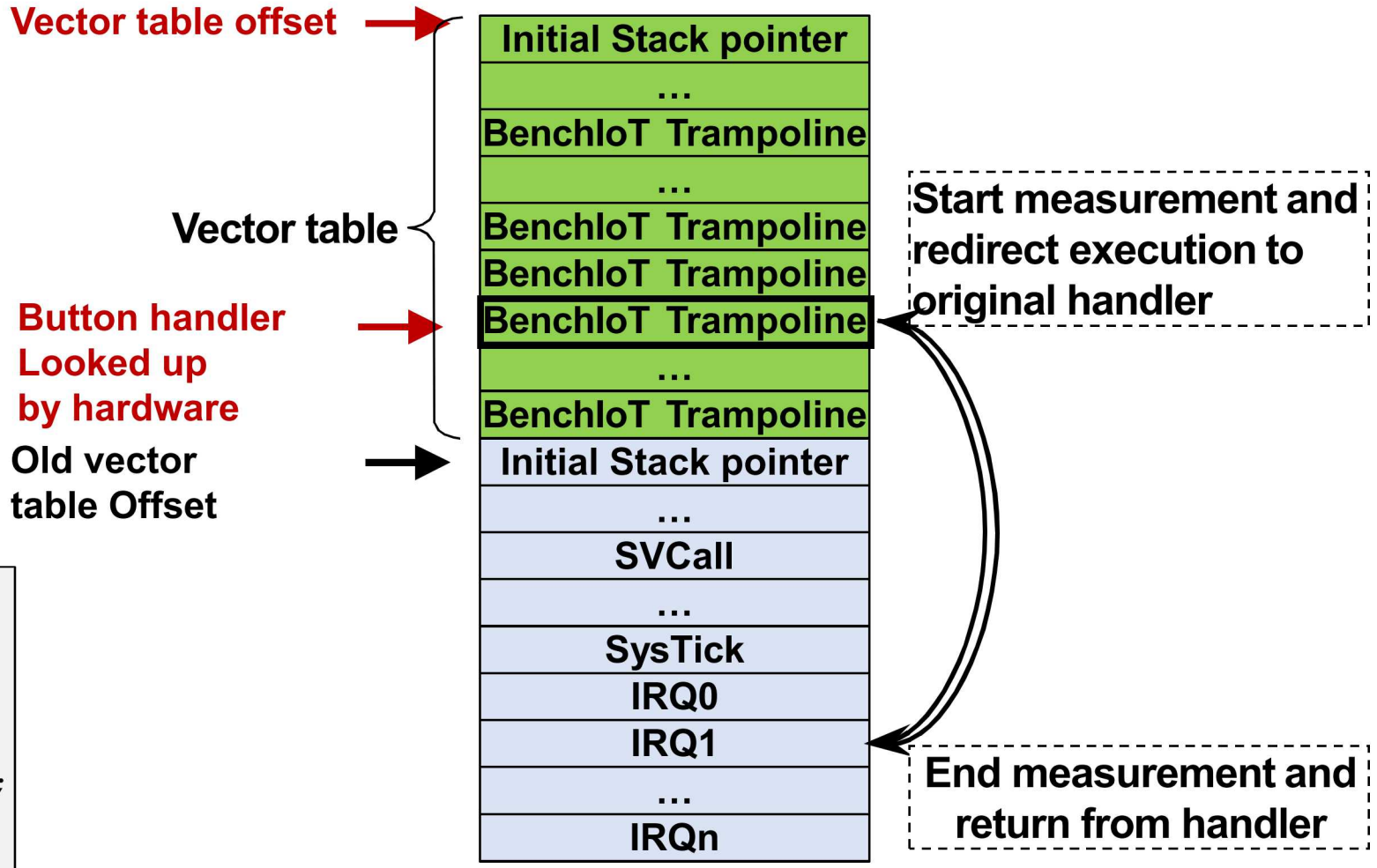
Benchmark	Task Type			Network Connectivity	Peripherals
	Sense	Compute	Actuate		
BEEBS		✓			
Dhrystone		✓			
CoreMark		✓			
IoTMark	✓	✓		Partially (Bluetooth only)	Only I^2C
SecureMark		✓			
BenchIoT	✓	✓	✓	✓	✓

Evaluation in Current IoT Defenses

Defenses	Evaluation Type	
	Benchmark	Case Study
TyTan [8]		✓
TrustLite [9]		✓
C-FLAT [10]		✓
nesCheck [11]		✓
SCFP [12]	Dhrystone [7]	✓
LiteHAX [13]	CoreMark [6]	✓
CFI CaRE [14]	Dhrystone [7]	✓
ACES [15]		✓
Minion [16]		✓
EPOXY [17]	BEEBS [4]	✓

Metric collector measurements

- With BenchIoT




```
   /* Pseudocode */
1. void BenchIoTTrampoline(void){
2.      lookup_exception_number();
3.      handle_stack_and_LR_value();
4.      Start_measurement();
5.      execute_original_exception();
6.      end_measurement();
7.
8.  }
```

Benchmarks Reproducibility

- **Example: reading temperature**

```
/* Pseudocode */
1.  void benchmark_temperature_handler(void){
2.
3.      // If not blocking,
4.      // emulate real execution
5.      read_actual_temperature();
6.      read_temperature_from_dataset();
7.      ...
8.
9.  }
```