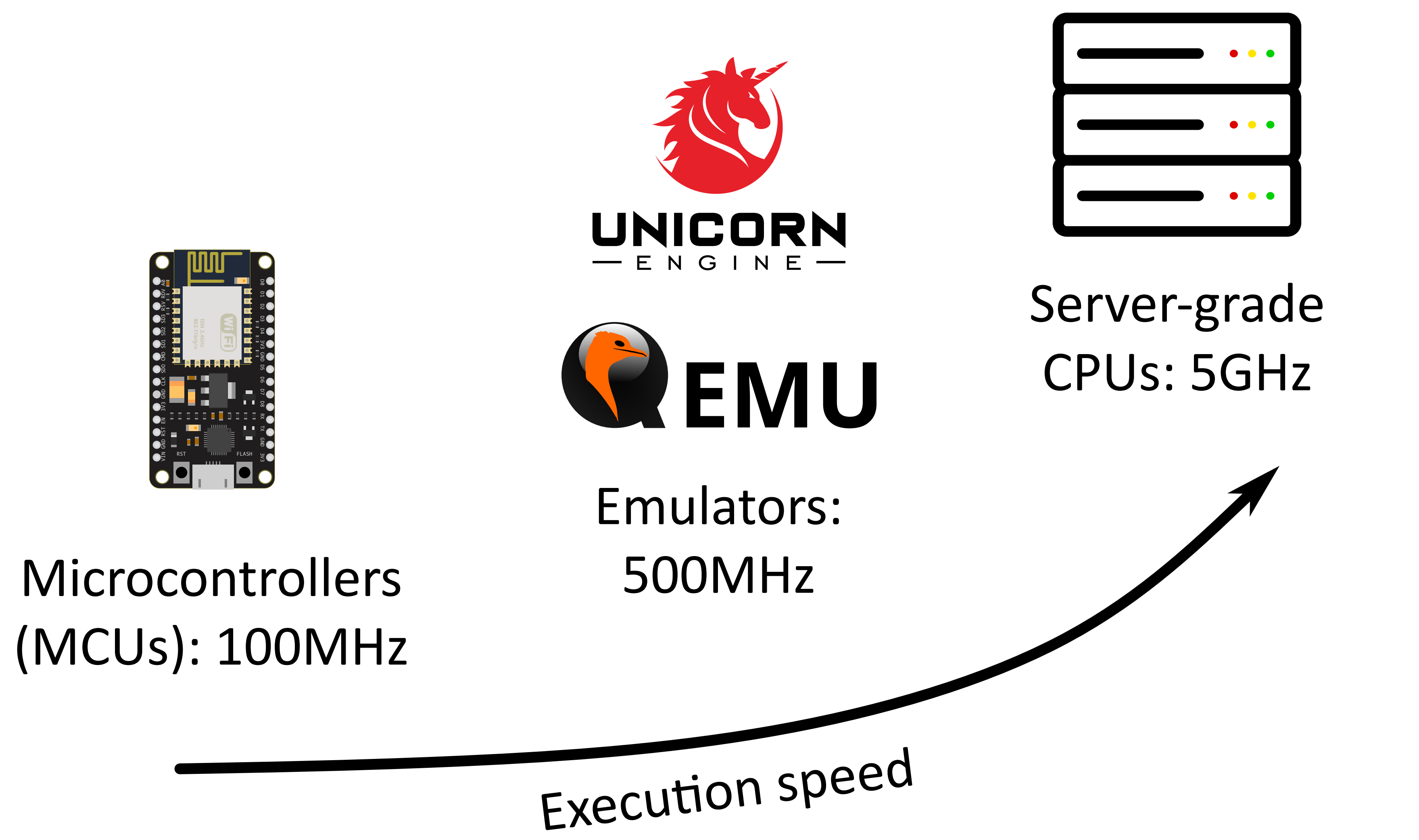
SURGEON: Performant, Flexible, and Accurate Re-Hosting via Transplantation

Best Paper Award

Florian Hofhammer[1], Marcel Busch[1], Qinying Wang[1,2], Manuel Egele[3], Mathias Payer[1]
[1]EPFL, Switzerland, [2]Zhejiang University, China, [3]Boston University, USA

Paper presented at the Workshop on Binary Analysis Research (BAR) 2024

Motivation: Emulation-Based Re-Hosting is Slow



Microcontrollers (MCUs): 100MHz

Emulators: 500MHz

Server-grade CPUs: 5GHz

Execution speed

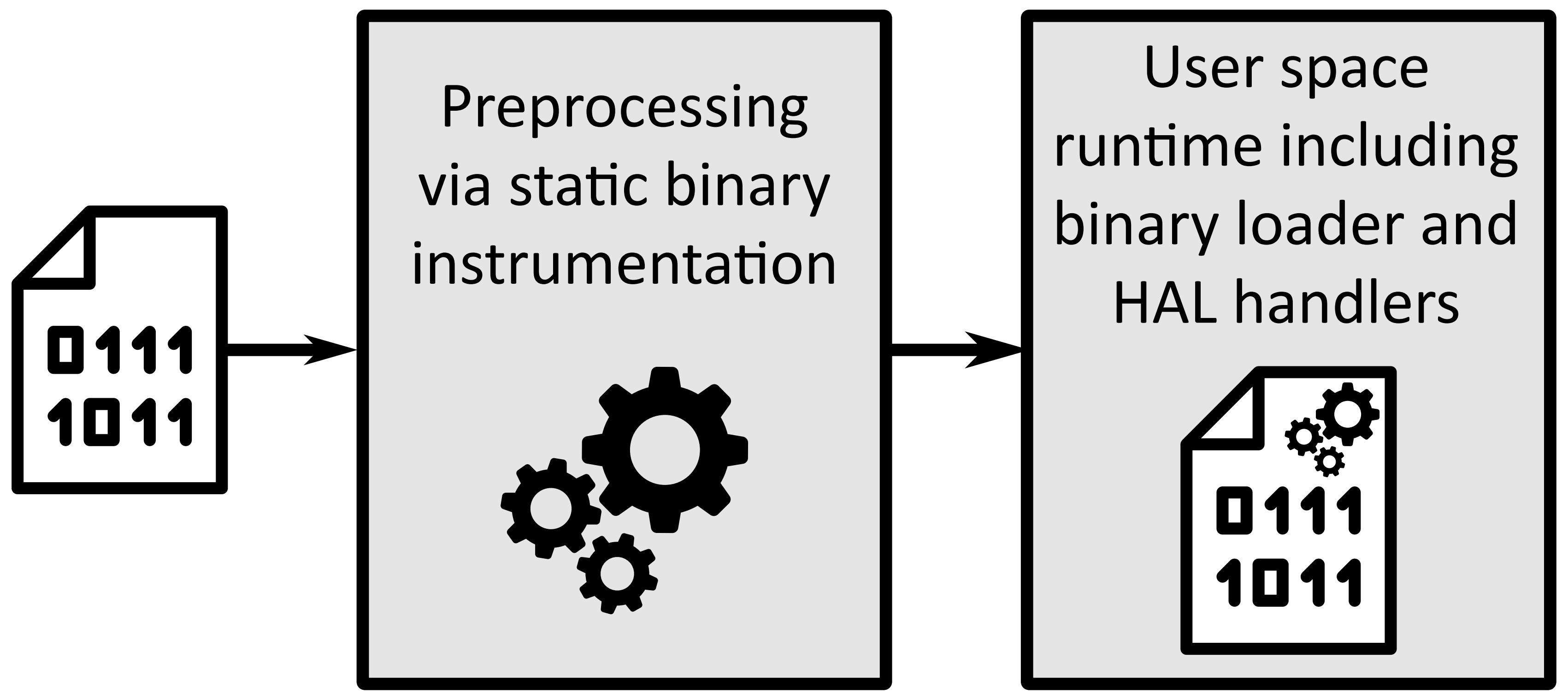
Key Insight: Similar ISA Across Device Classes

Thumb2 on
Arm Cortex-M,
bare-metal MCU

Thumb2 on
Arm Cortex-A,
Linux user space

Key Idea: Cross-ISA *Transplantation*

Adjust a firmware binary and run it on a server-grade CPU sharing a compatible Instruction Set Architecture (ISA) with the MCU at hand



Preprocessing via static binary instrumentation

User space runtime including binary loader and HAL handlers

Preprocessing

Static Binary Instrumentation...

① replaces problematic instructions for native execution
② (optionally) inserts use-case-specific instrumentation code
③ inserts branches to HAL handlers (peripheral emulation)

Examples:

①

```
software_interrupt:
    ~~svc~~  bkpt #1
```

②

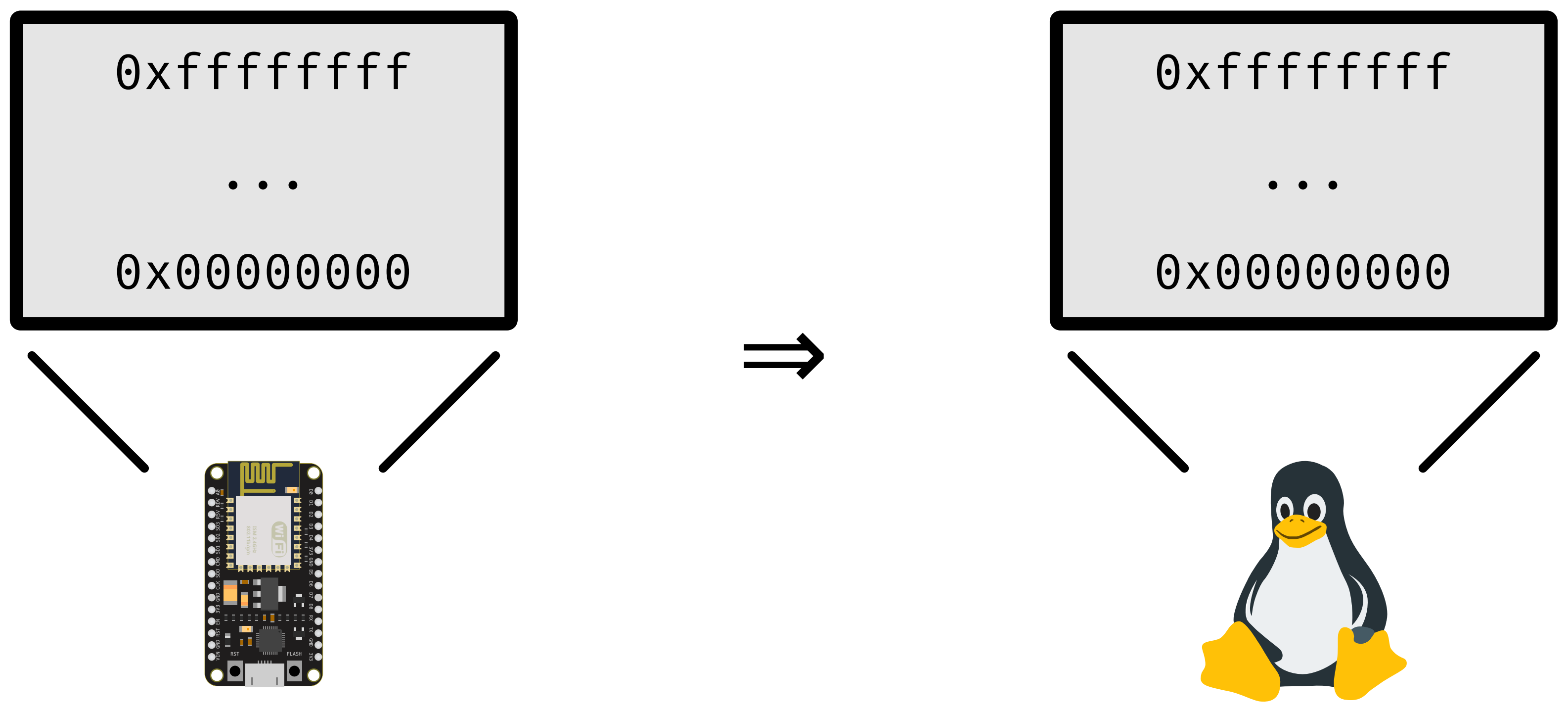
```
func:
    ldr     r0, [pc, #0]
    ~~mov.w   pc, lr~~  b.w trampoline
.word: 0x01234567

trampoline:
    <instrumentation>
    mov.w   pc, lr
```

③

```
HAL_UART_Receive:
  ~~ldrb.w  r3, [r0, #58]~~   movw   ip, #34677   ; 0x8775
  ~~uxtb    r3, r3~~          movt   ip, #61453   ; 0xf00d
  ~~cmp   r3, #32~~
  ~~beq.n   80068e8~~         mov    pc, ip
```

Runtime

Mimic the MCU physical address space in the Linux process's virtual address space



```
0xffffffff
...
0x00000000
```

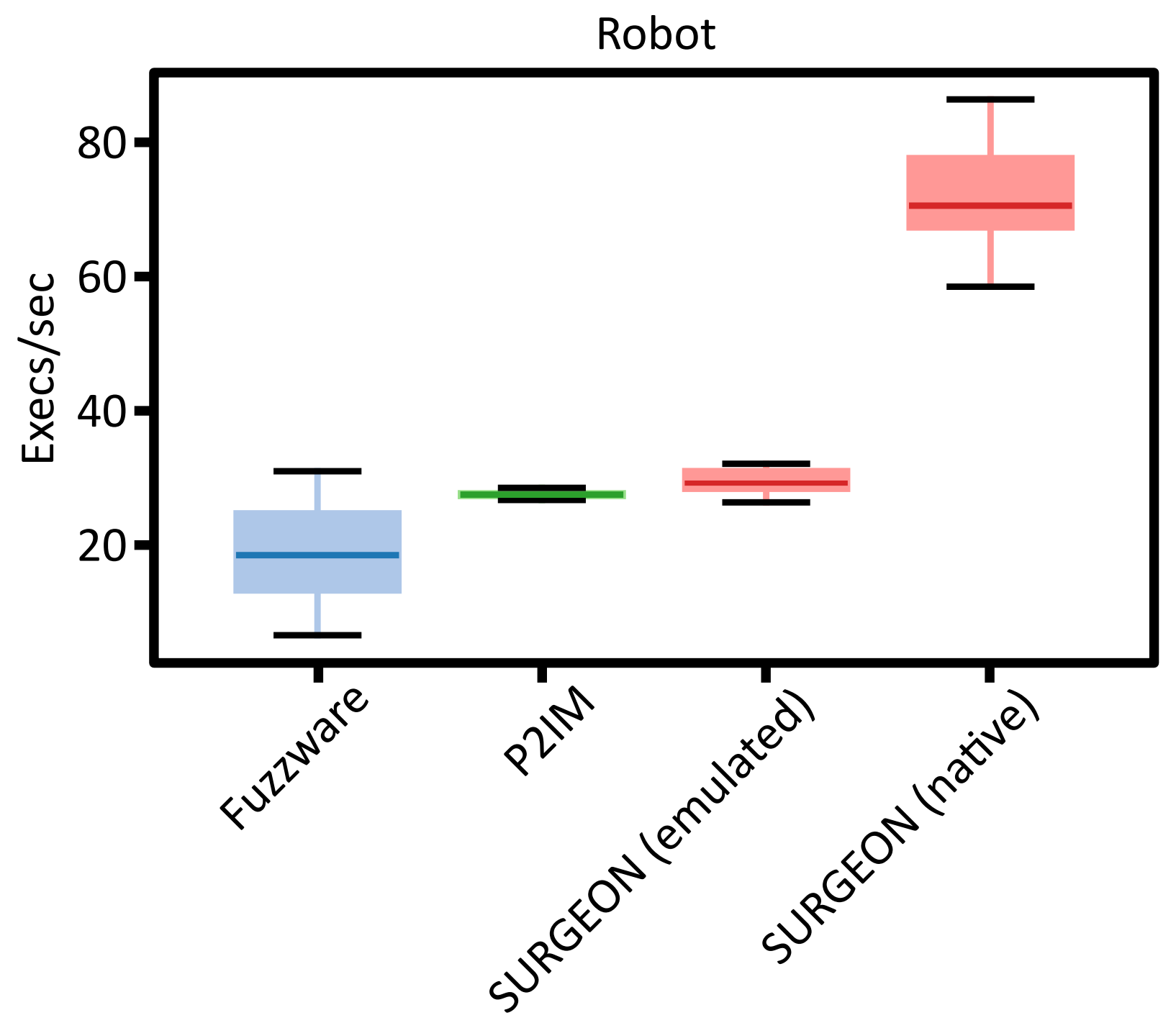
⇒

```
0xffffffff
...
0x00000000
```

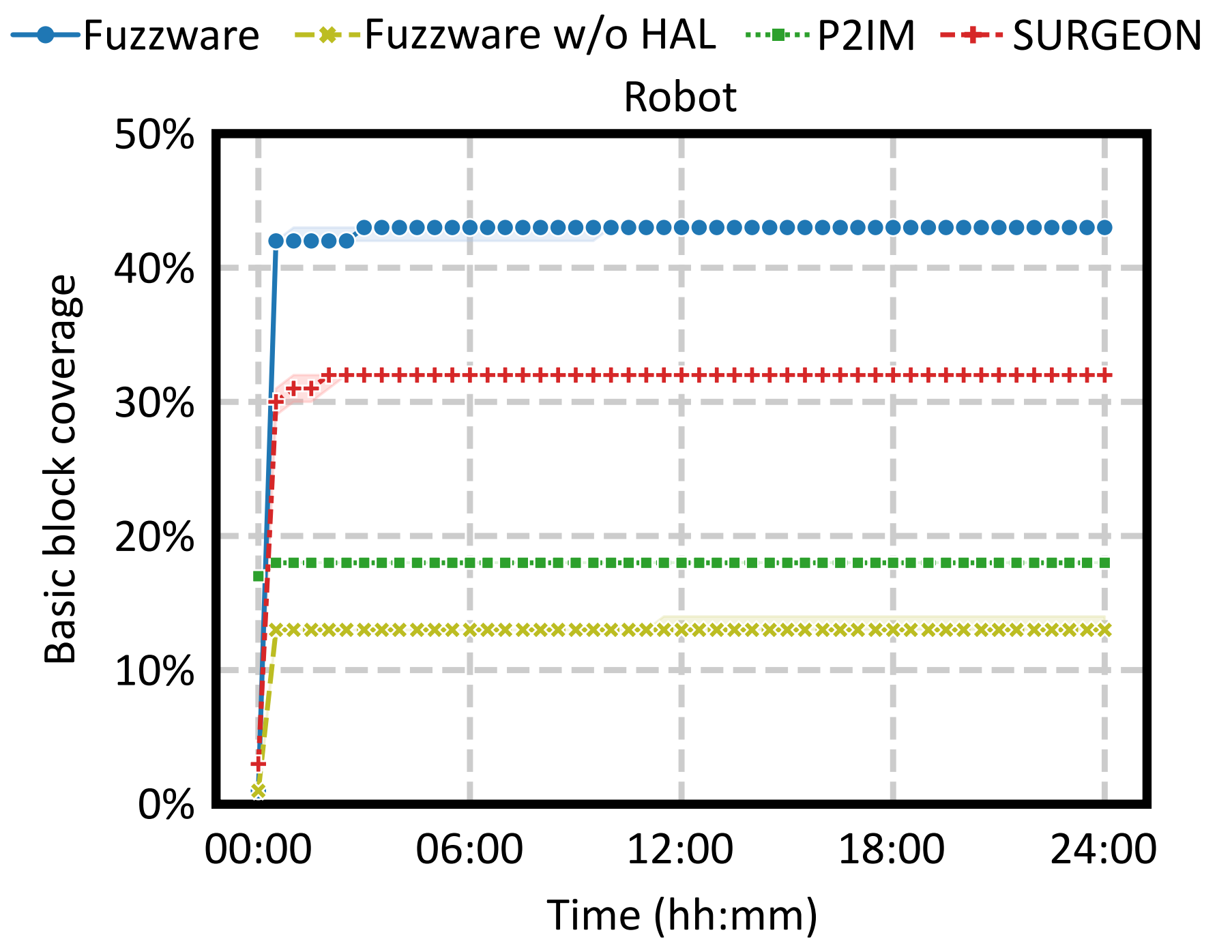
Enable application of arbitrary Linux user space tooling to re-hosted firmware, e.g., off-the-shelf fuzzers or debuggers

Exemplary Use Case: Fuzzing

Applying an off-the-shelf fuzzer (AFL++) yields...



...high throughput thanks to native execution



...high coverage thanks to precise peripheral modeling