ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences

HALucinator: Firmware Re-hosting Through Abstraction Layer Emulation

论文分享

芮志清

zhiqing.rui@gmail.com

中国科学院软件研究所

2020年10月16日

简介

HALucinator: Firmware Re-hosting Through Abstraction Layer Emulation

Abraham A Clements, *Sandia National Laboratories;* Eric Gustafson, *UC Santa Barbara and Sandia National Laboratories;* Tobias Scharnowski, *Ruhr-Universität Bochum;* Paul Grosen, *UC Santa Barbara;* David Fritz, *Sandia National Laboratories;* Christopher Kruegel and Giovanni Vigna, *UC Santa Barbara;* Saurabh Bagchi, *Purdue University;* Mathias Payer, *EPFL*

https://www.usenix.org/conference/usenixsecurity20/presentation/clements

解决的核心问题：固件模拟执行的过程中，由于无法连接实际硬件，导致无法正常执行的问题

URL：https://www.usenix.org/conference/usenixsecurity20/presentation/clements

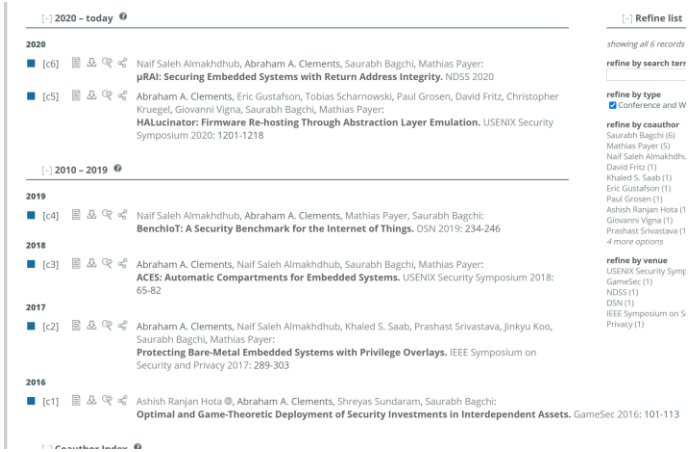
作者

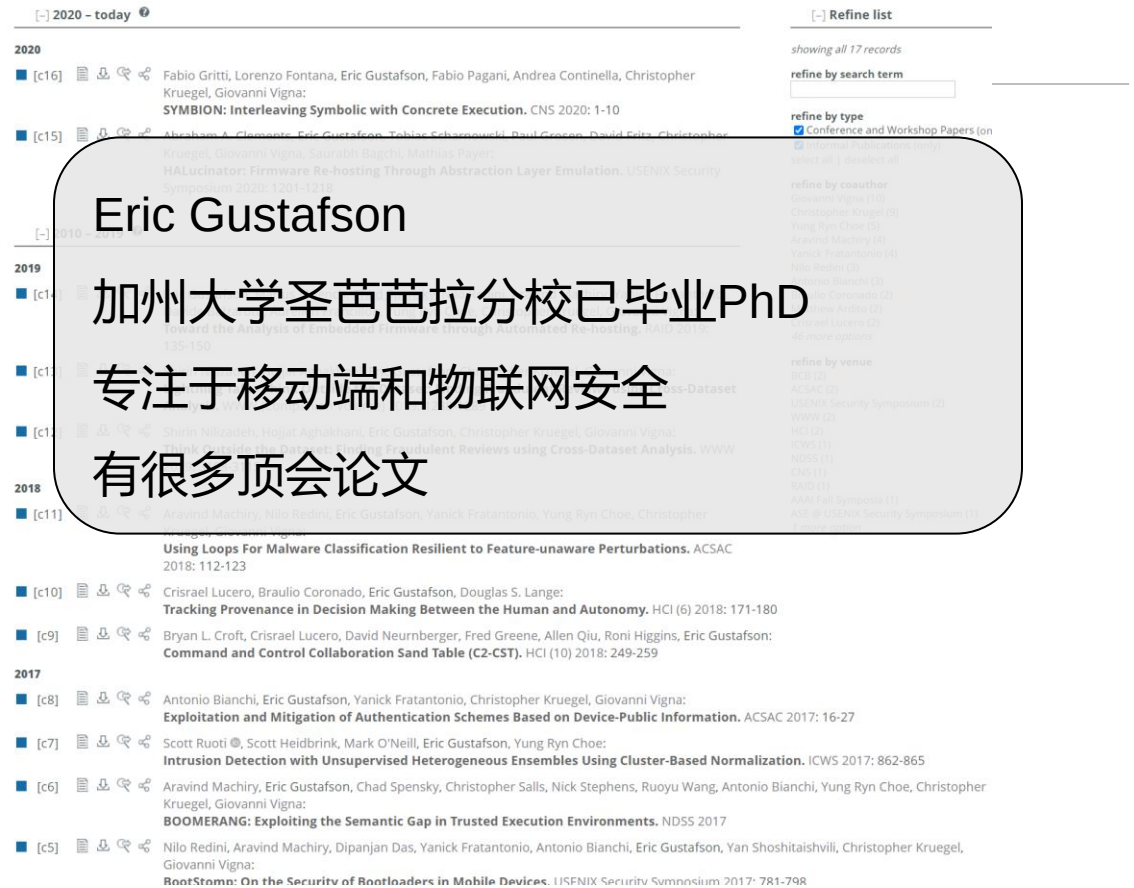
Abraham A. Clements

普渡大学PhD

从2017年开始做物联网安全研究

3篇一作顶会





Eric Gustafson

加州大学圣芭芭拉分校已毕业PhD

专注于移动端和物联网安全

有很多顶会论文

几个合作实验室都大有来头，在安全学术领域很强

背景知识：物联网技术


PHILIPS

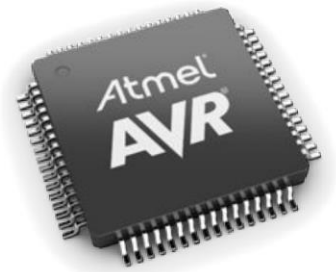

72
ecobee


PUM11
TN623
DI635
DO635
RO663
SP600


1 2 3 4 5
6 7 8 9 0


RIGOL

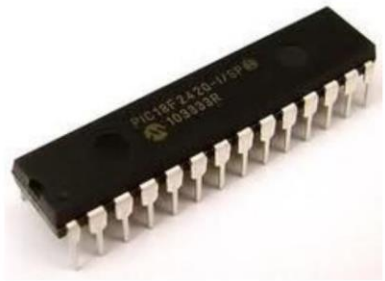
设备芯片


Atmel
AVR


Ambarella
A9-A0-RH
NZV1X
S0101212
A999


Qualcomm
snapdragon

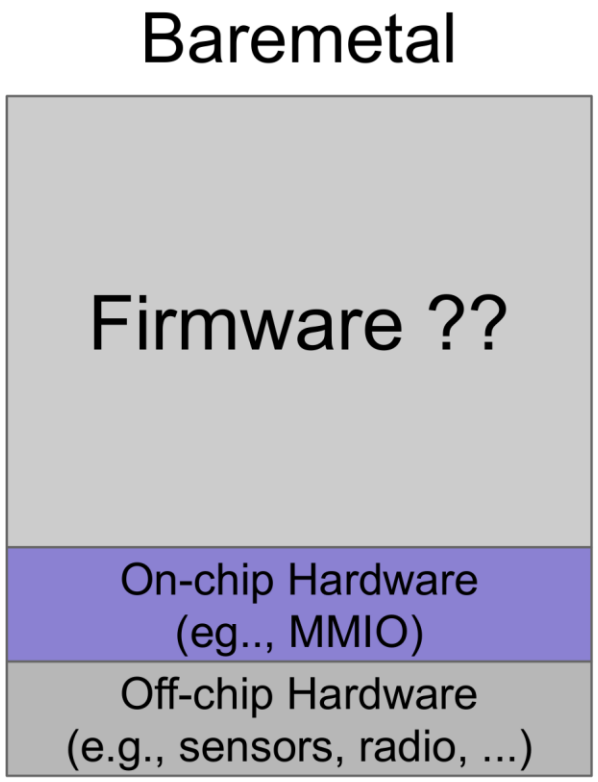

M430F1232
76TG4
DYCRD



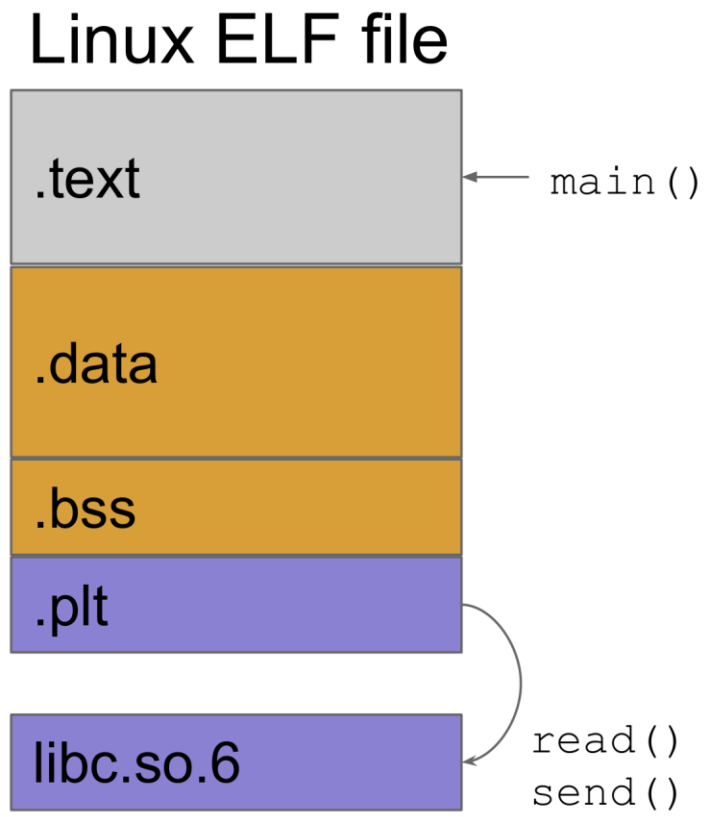

STM32F103
VCT6
ARM


NVIDIA
TEGRA K1
NVIDIA
TEGRA X1

裸金属固件



Raw hardware access

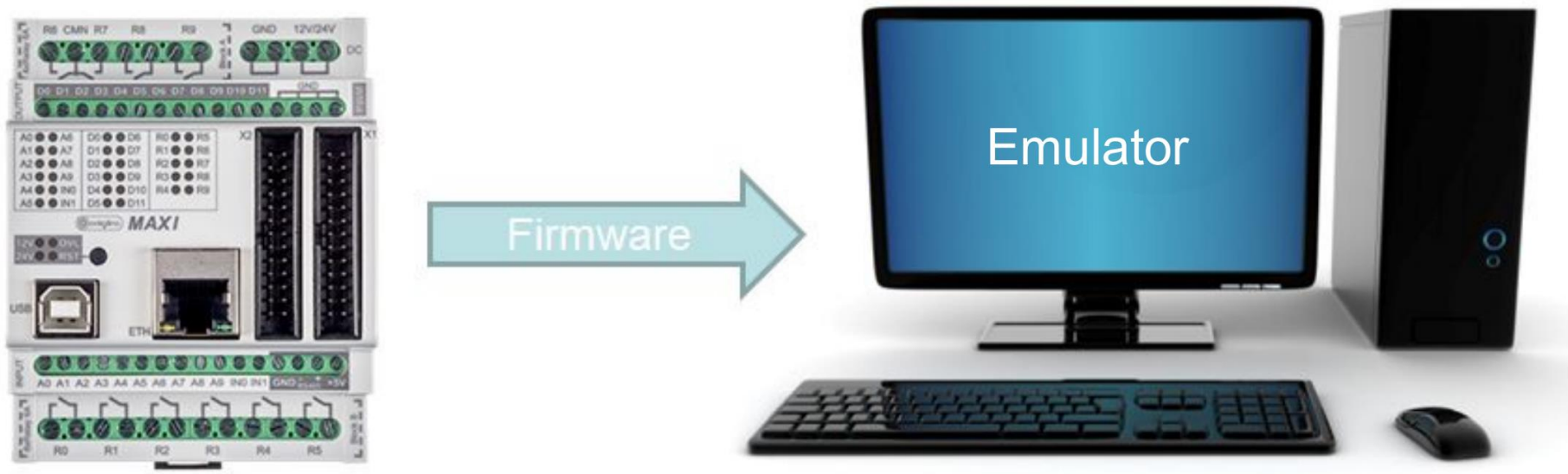


Kernel abstractions used for hardware interactions

硬件是块硬骨头

- 调试接口
  - 被禁用
  - 被限制
- 并行困难
- 其他限制
  - 贵 💎💎💎
  - 易坏💀💀💀

用模拟器模拟固件?



HALucinator的目标:
无需专用硬件即可进行可扩展的固件测试

外围设备阻止模拟

On chip	Off chip
CPU	Ethernet
AES Accelerator	SD-MMC
Hash	GPIO
Coprocessor	Camera
Timers	LCD
Counters	Touch Screen
Flash Controller	Wireless
Clock Config	EEPROM
IAP	Serial
DMA	CAN
	Analog IO
	USB

模拟有多困难?

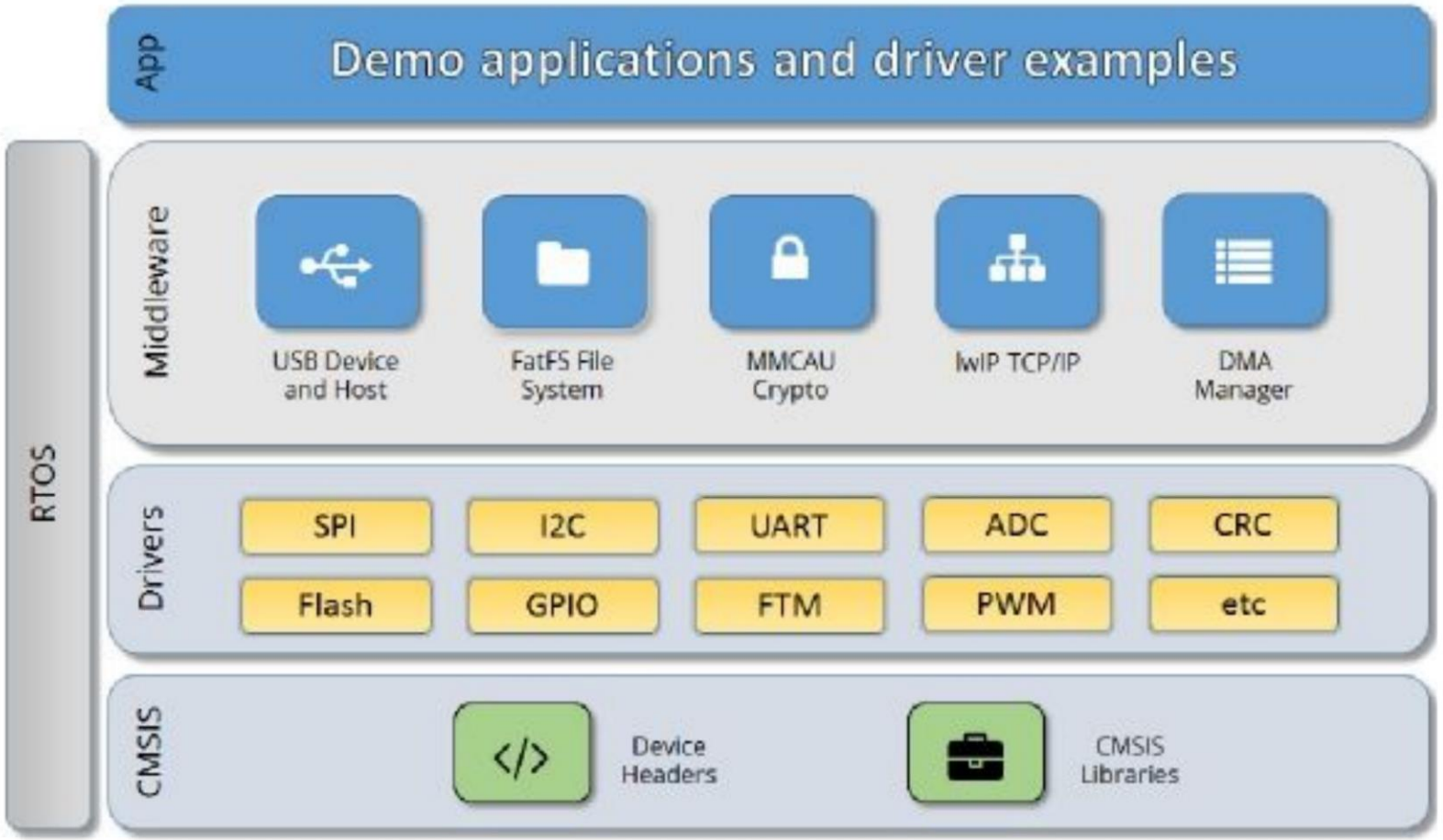
44,520 微控制器

3,502 数据表

26 厂商

如果不支持外设，裸机固件将无法运行!
至少有上万种种外设及其组合!

硬件抽象库


App
Demo applications and driver examples
RTOS
Middleware
USB Device and Host
FatFS File System
MMCAU Crypto
lwIP TCP/IP
DMA Manager
Drivers
SPI
I2C
UART
ADC
CRC
Flash
GPIO
FTM
PWM
etc
CMSIS
Device Headers
CMSIS Libraries

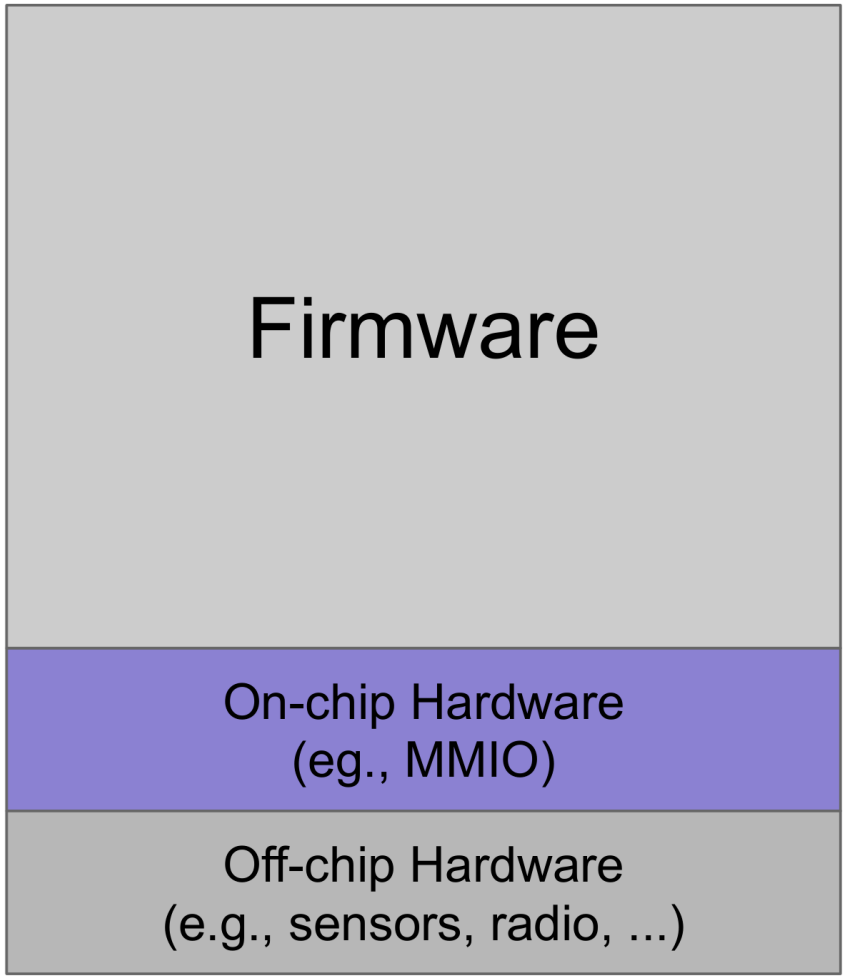

NXP
MCUXpresso

HALs非常普及

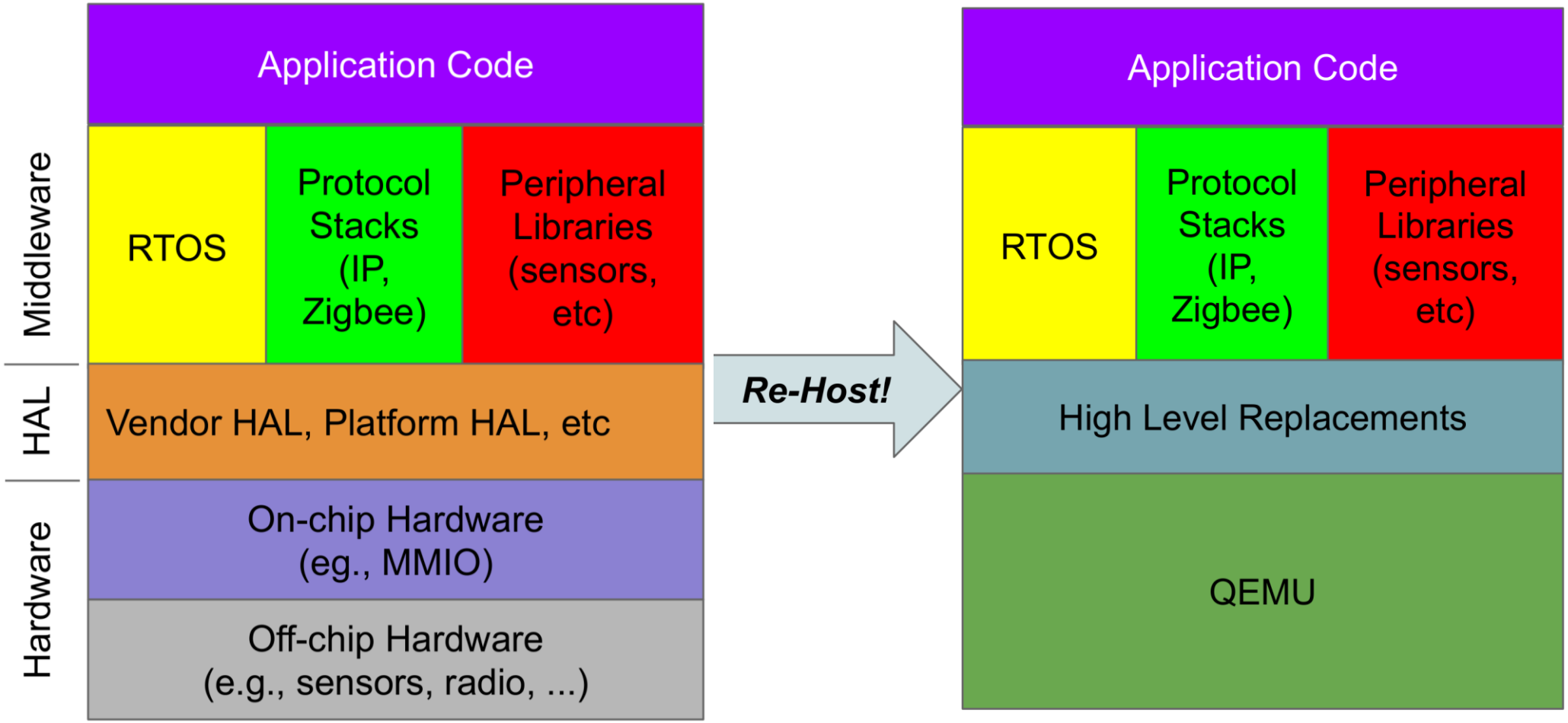
HALucinator

从高层替换HAL和其他库
实现方式：从改造主机托管扩展问题，使其支持数万个设备到数十个HALS

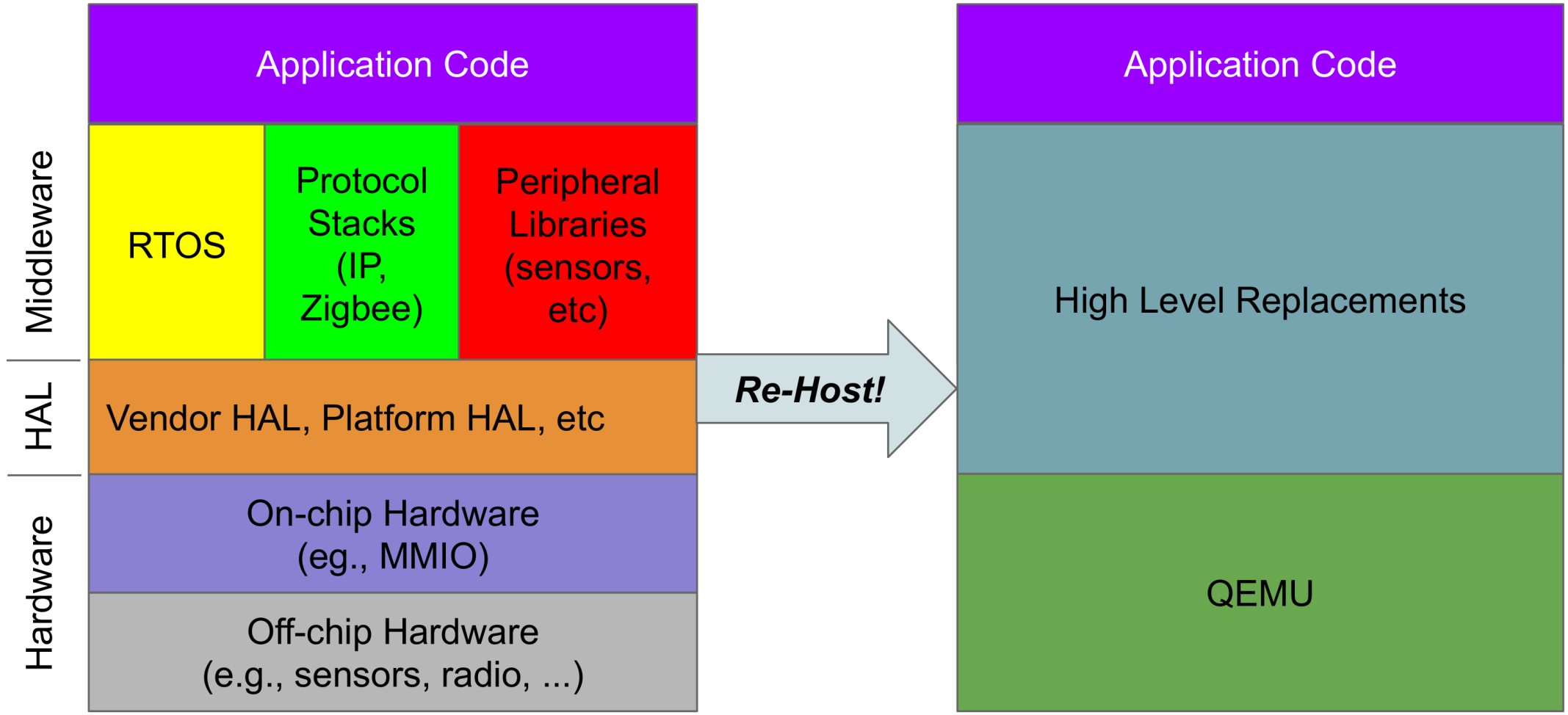
现代的固件架构栈


Firmware
On-chip Hardware
(eg., MMIO)
Off-chip Hardware
(e.g., sensors, radio, ...)

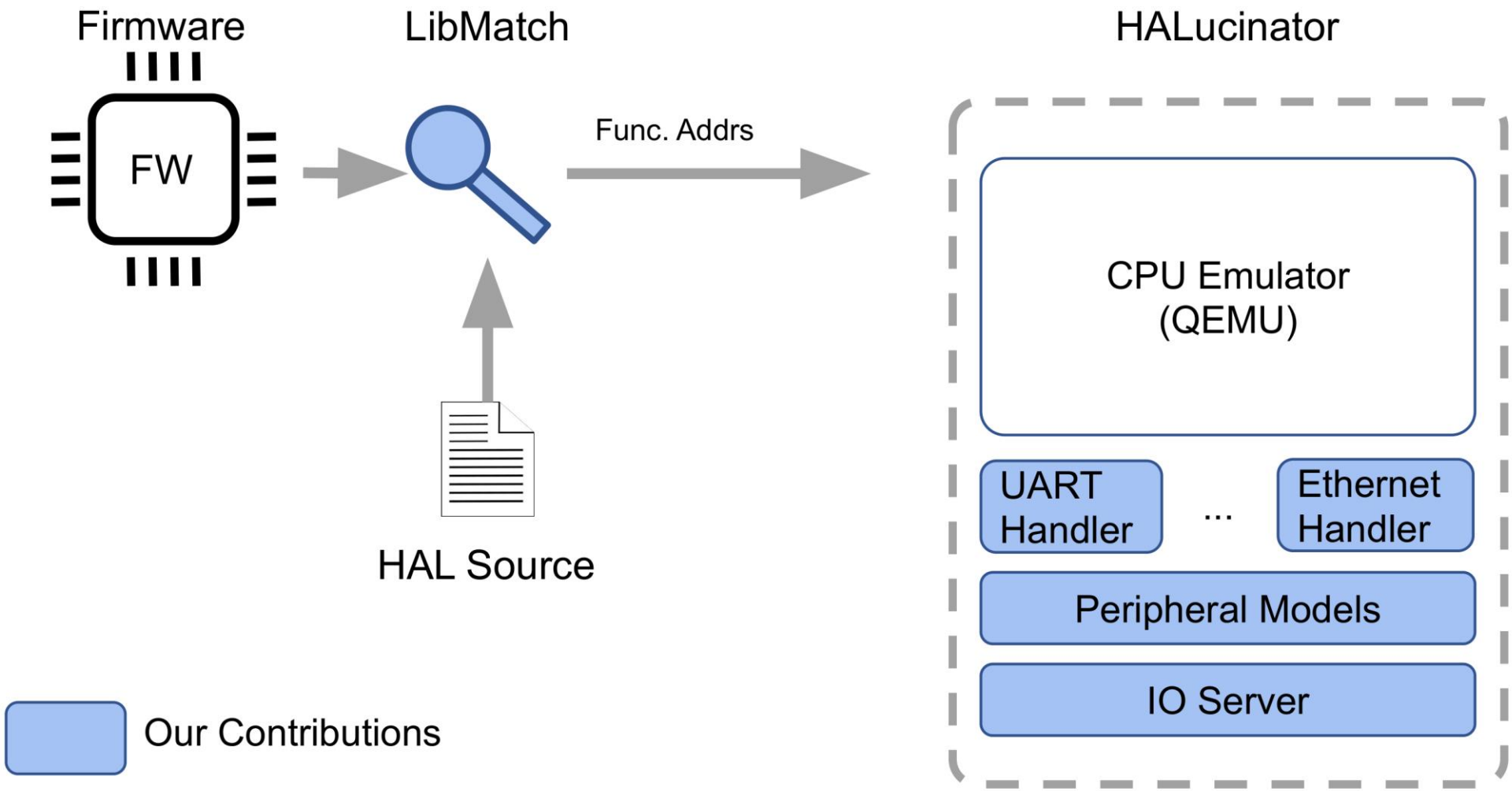
高层级模拟


Application Code
RTOS
Protocol Stacks (IP, Zigbee)
Peripheral Libraries (sensors, etc)
Middleware
HAL
Vendor HAL, Platform HAL, etc
Hardware
On-chip Hardware (eg., MMIO)
Off-chip Hardware (e.g., sensors, radio, ...)
Re-Host!
Application Code
RTOS
Protocol Stacks (IP, Zigbee)
Peripheral Libraries (sensors, etc)
High Level Replacements
QEMU

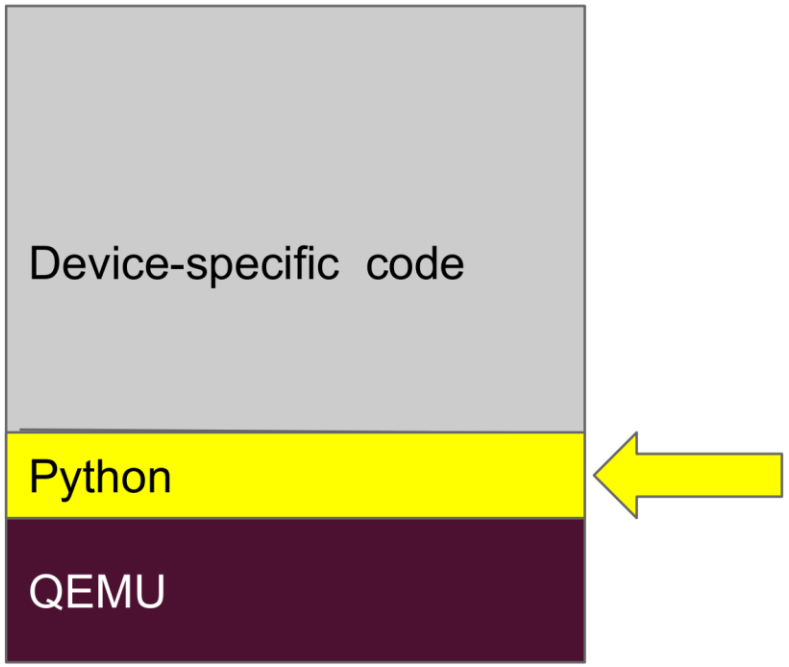
高层级模拟


Application Code
RTOS
Protocol Stacks (IP, Zigbee)
Peripheral Libraries (sensors, etc)
Middleware
HAL
Vendor HAL, Platform HAL, etc
Hardware
On-chip Hardware (eg., MMIO)
Off-chip Hardware (e.g., sensors, radio, ...)
Re-Host!
Application Code
High Level Replacements
QEMU

HALucinator 实现

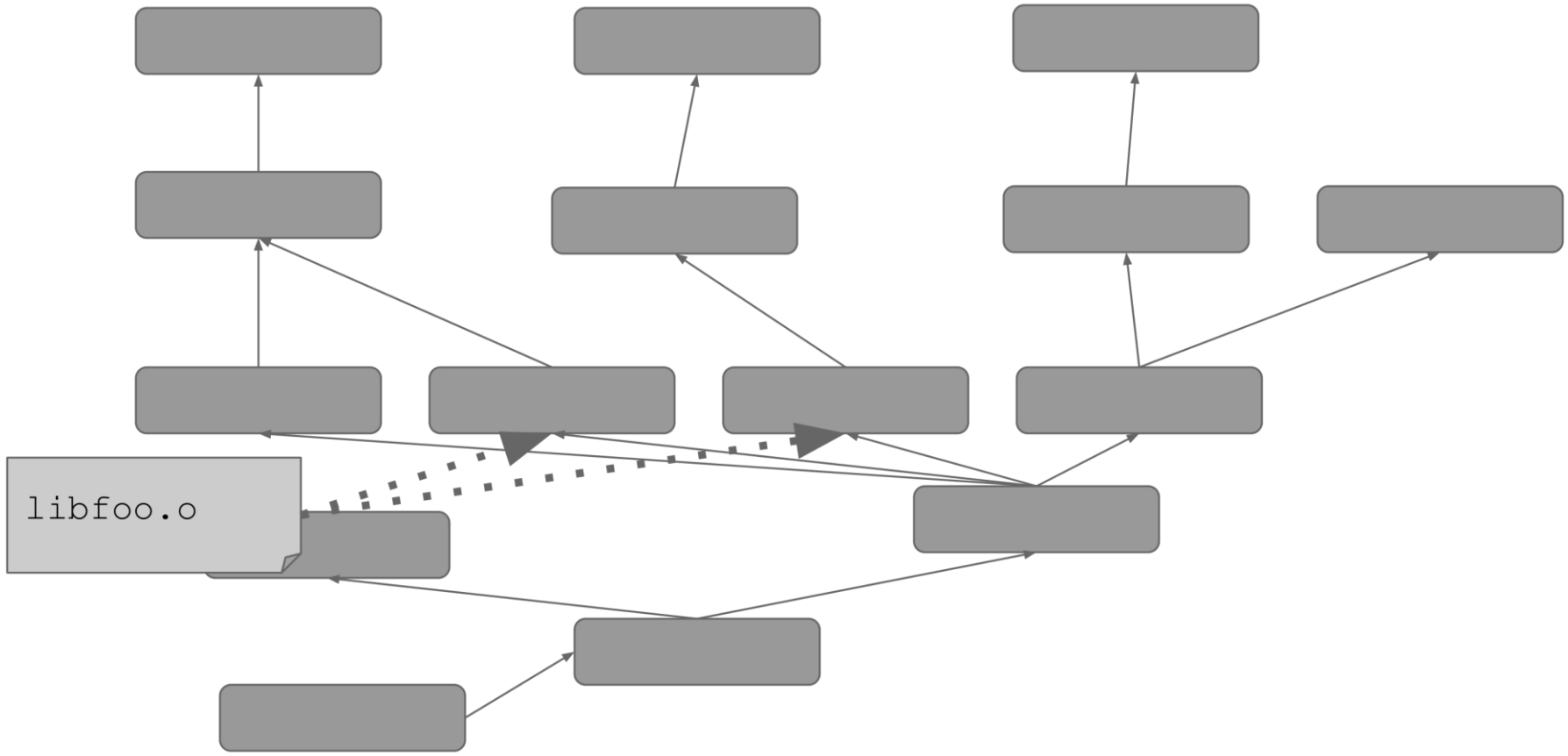


Handler 例子



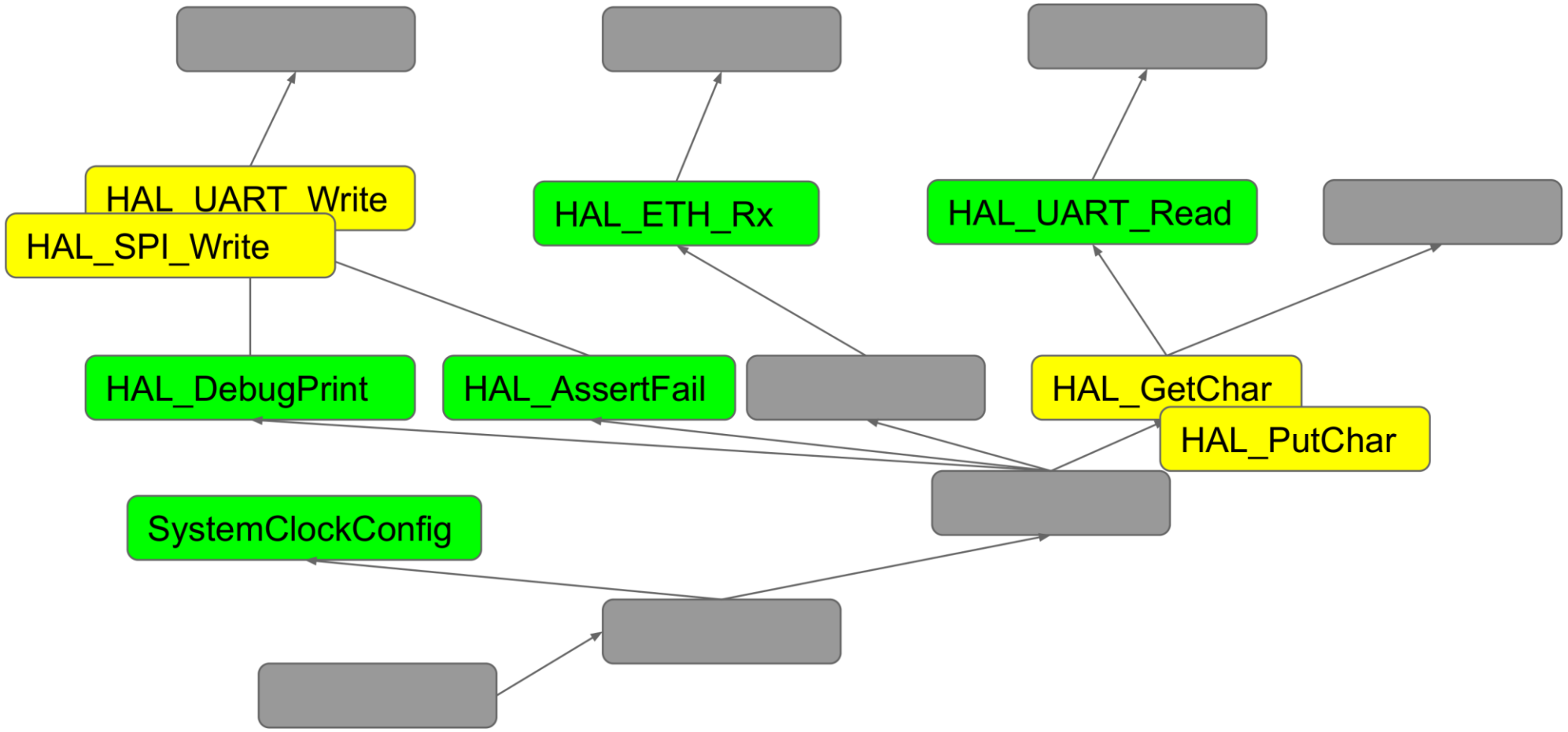

```
def i2c_read_buf(uc):
    # i2c_read_buf(char* buf, int len);
    buf = uc.regs.r1 # arg 0: The buffer
    l = uc.regs.r2   # arg 1: Buffer length
    assert(buf != 0)  # Crash on bad arguments
    assert(len > 0)
    data = I2CModel.rx('i2c', 0, len) # Get the data
                                      # from the virtual bus
    uc.mem[buf] = data     # Store it in the emulator
```

库匹配：匹配库内容

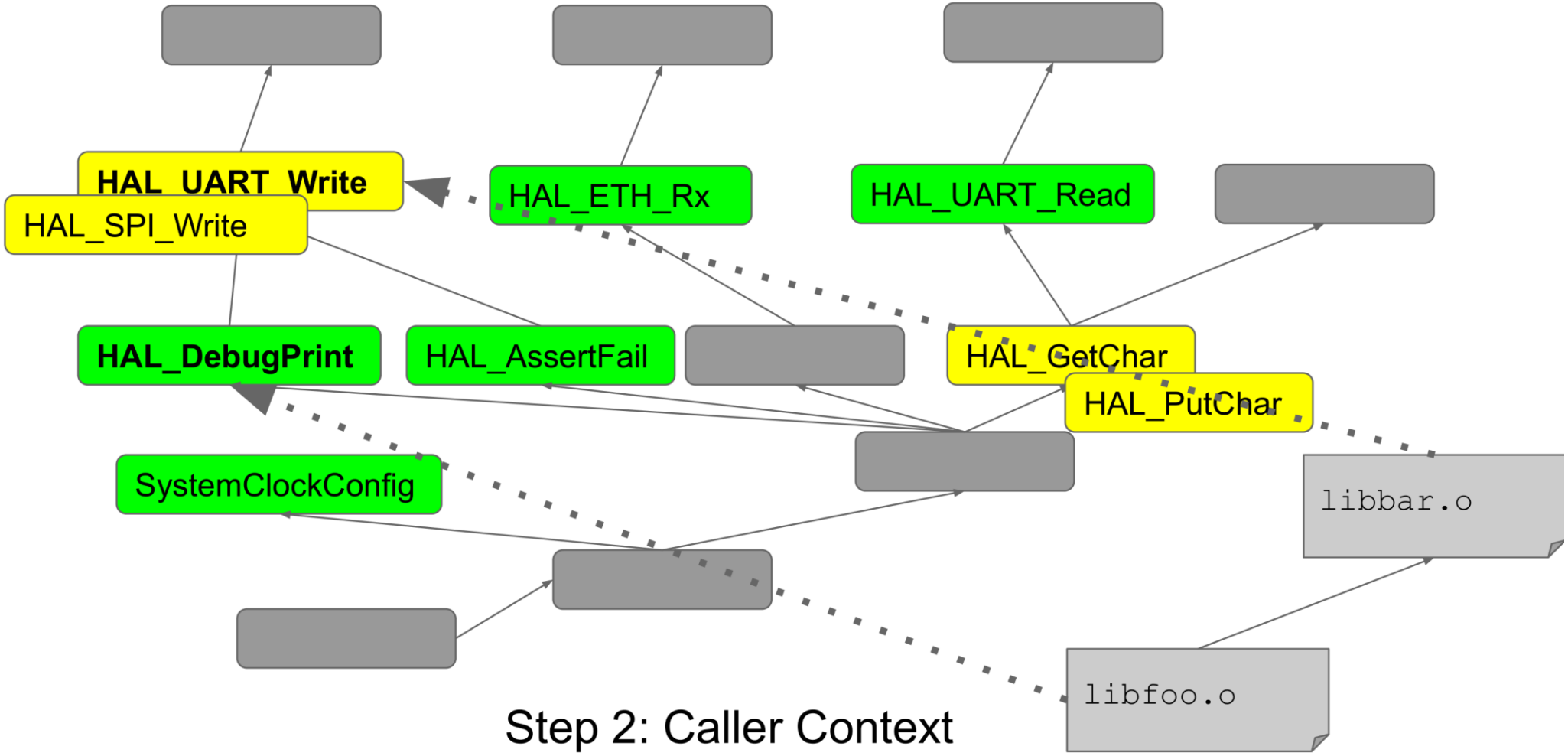


Step 1: Match library content

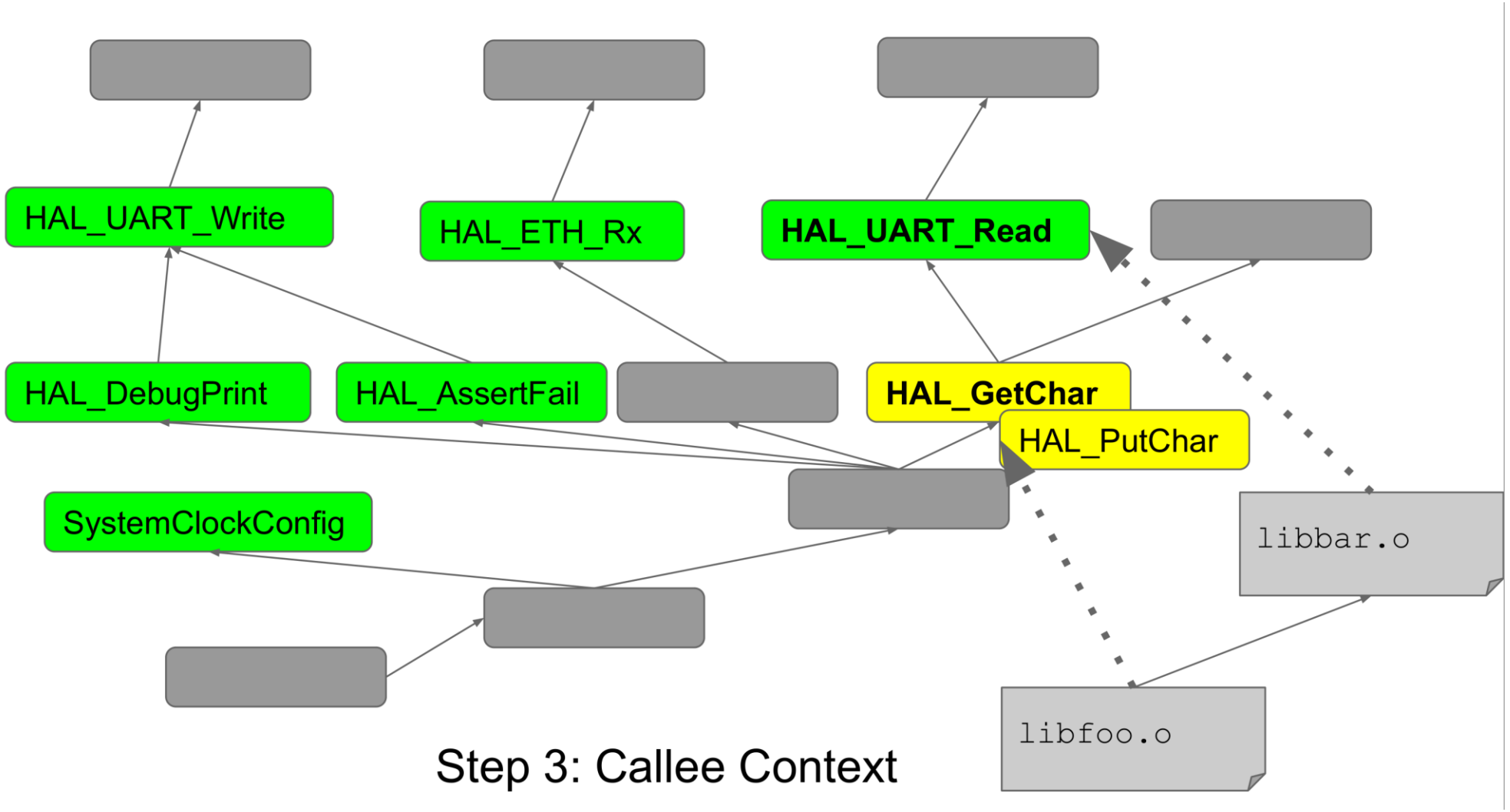
库匹配：匹配库内容


HAL_UART_Write
HAL_SPI_Write
HAL_ETH_Rx
HAL_UART_Read
HAL_DebugPrint
HAL_AssertFail
HAL_GetChar
HAL_PutChar
SystemClockConfig

库匹配：被调用者上下文


HAL_UART_Write
HAL_SPI_Write
HAL_ETH_Rx
HAL_UART_Read
HAL_DebugPrint
HAL_AssertFail
HAL_GetChar
HAL_PutChar
SystemClockConfig
libbar.o
libfoo.o
Step 2: Caller Context

库匹配：被调用者上下文


HAL_UART_Write
HAL_ETH_Rx
HAL_UART_Read
HAL_DebugPrint
HAL_AssertFail
HAL_GetChar
HAL_PutChar
SystemClockConfig
libbar.o
libfoo.o
Step 3: Callee Context

hal-fuzz 模糊测试功能

- 基于AFL-Unicorn
- 输入用尽时程序退出
- 基于块计数的确定性计时器
- 中断事件也基于块计数
- 通过Unicorn自身的错误检测到崩溃
- 检测器以及处理程序断言

16个固件示例

ATMEL ASF
- USART
- FAT32 on SD-Card
- **HTTP Server**
- **6LoWPAN Sender and Receiver**

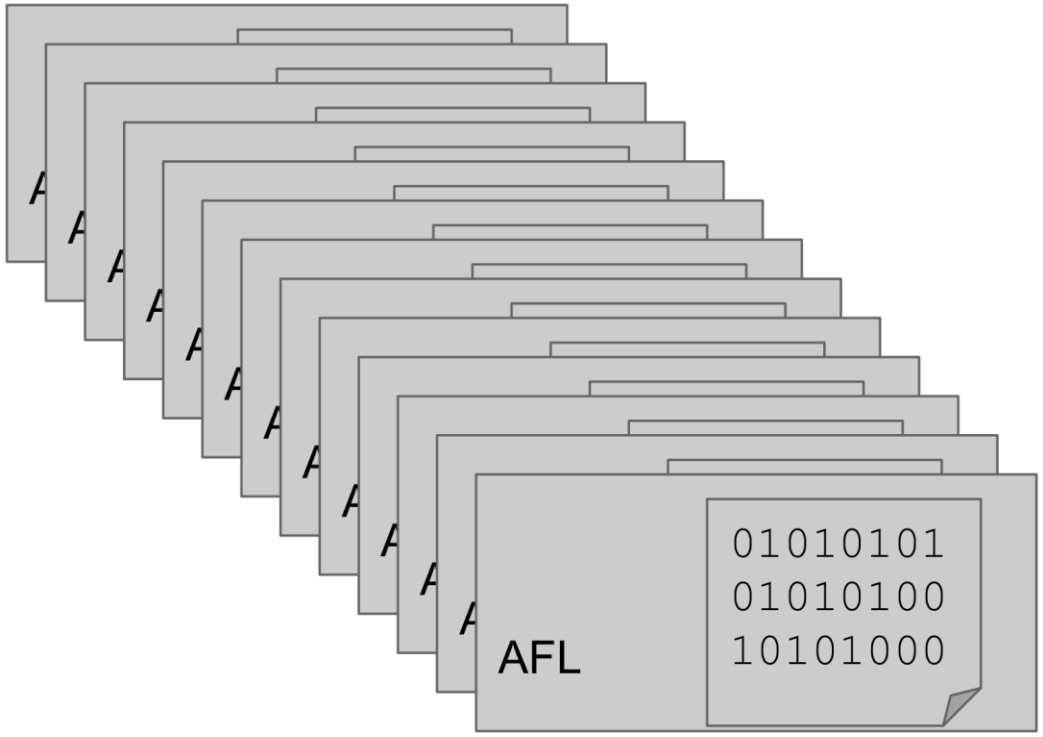
STM32Cube
- UART
- FAT32 on SD-Card
- **UDP-Echo Server and Client**
- **TCP-Echo Server and Client**
- **PLC**

NXP -MCUXpresso
- UART
- **UDP Echo Server**
- **TCP Echo Server**
- **HTTP Server**

库匹配结果

	“Naïve” LibMatch (Bindiff)	LibMatch w/ context
Correct	74.5%	87.4%
Missing	5.0%	3.2%
Collisions	18.8%	8.5%
Incorrect	2.5%	0.9%
External	--	9.96%

Fuzzing

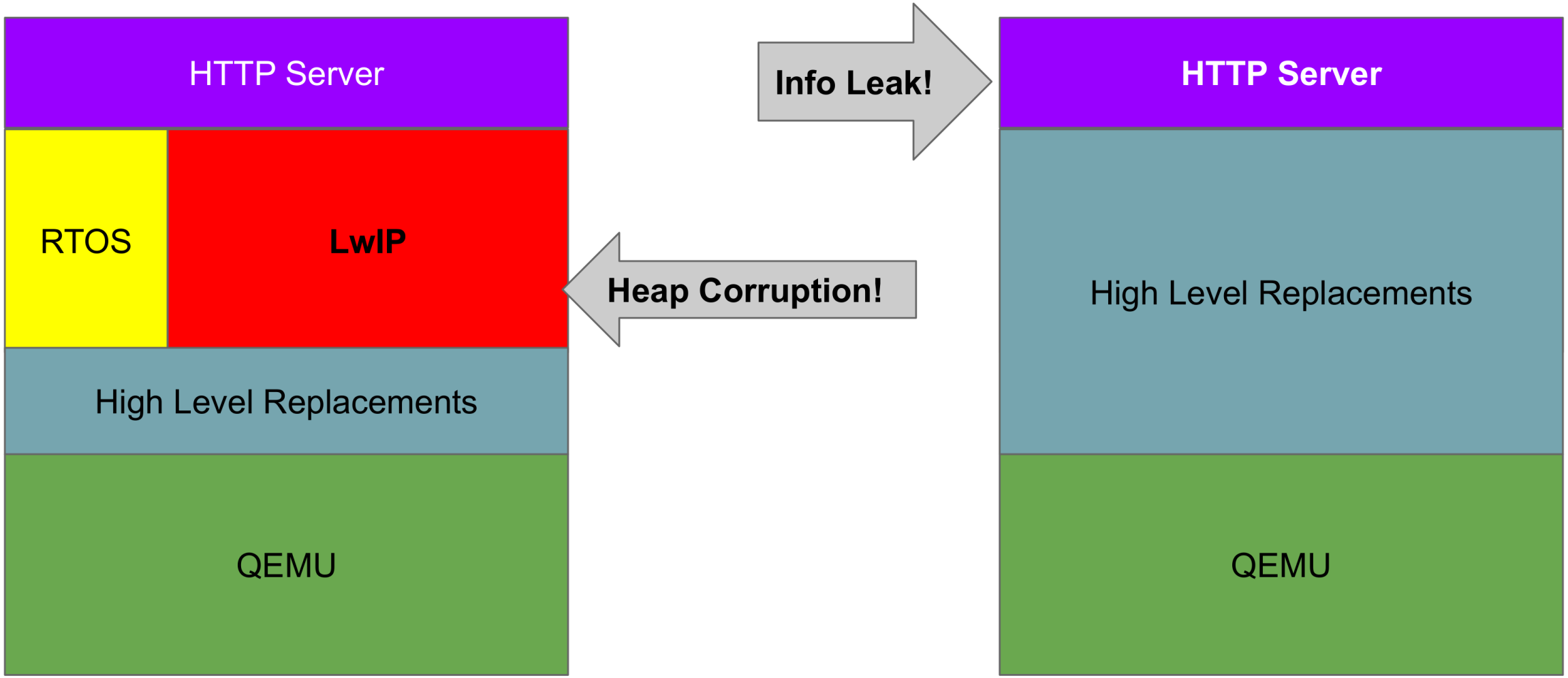


Hundreds of millions of executions with real parallel AFL

New crashes in:

- STM's ST-PLC Kit
- Atmel's HTTP Server example
- Atmel's Contiki 6LowPAN examples

多层Fuzzing


HTTP Server
RTOS
LwIP
High Level Replacements
QEMU
Info Leak!
Heap Corruption!
HTTP Server
High Level Replacements
QEMU

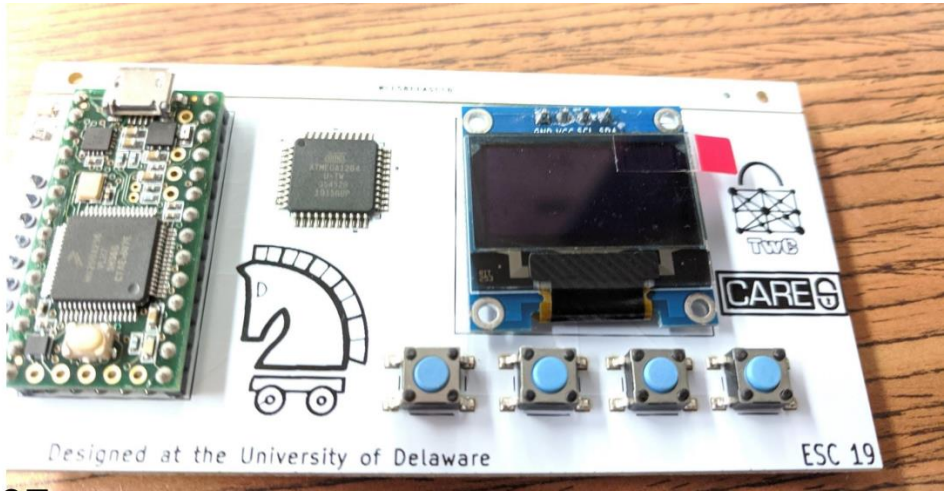
发现的CVE

- CVE-2019-8359：Contiki OS数据包重组功能中的缓冲区溢出导致的远程执行代码漏洞。

- CVE-2019-9183：Contiki OS数据包重组功能中的整数下溢漏洞导致的远程拒绝服务漏洞。

CSAW ESC 2019 results

- ESC是一个比较知名的嵌入式安全比赛。2019年的比赛为无线射频识别(RFID) 的安全性。
- 比赛使用**美国国家安全局(NSA)**开发的逆向工程工具黑客定制的 RFID 读卡器固件。



团队成绩

●模拟了所有挑战集的ARM部分

●解决了18/19的挑战

●使用模糊测试自动解决3个挑战

●获得第一名！

总结

贡献

- 提出**HAL库模拟**技术，可在无需依赖实际硬件的情况下使用QEMU等模拟器仿真二进制固件
- 改进了现有的依赖库匹配技术
- 提出HALucinator仿真系统，可通过抽象处理程序和外围模型库的方式进行交互式仿真和固件Fuzzing
- 通过对16个固件的模拟，证明了方法的实用性。并通过Fuzzing的方式发现了两个0day漏洞

收获

- 这篇论文解决了一篇经典论文Firmadyne在固件模拟上的核心问题
- 可以尝试复现此文章来解决固件漏洞挖掘的问题

代码开源：

- https://github.com/embedded-sec/halucinator
- https://github.com/ucsb-seclab/hal-fuzz

THANKS