Codasip

CHERI Software Ecosystem

Carl Shaw
Senior Safety and Security Manager

November 2024

↘

Agenda

1. Software strategy
2. Language / firmware support
3. Microcontroller software
4. Application core software
5. The way ahead

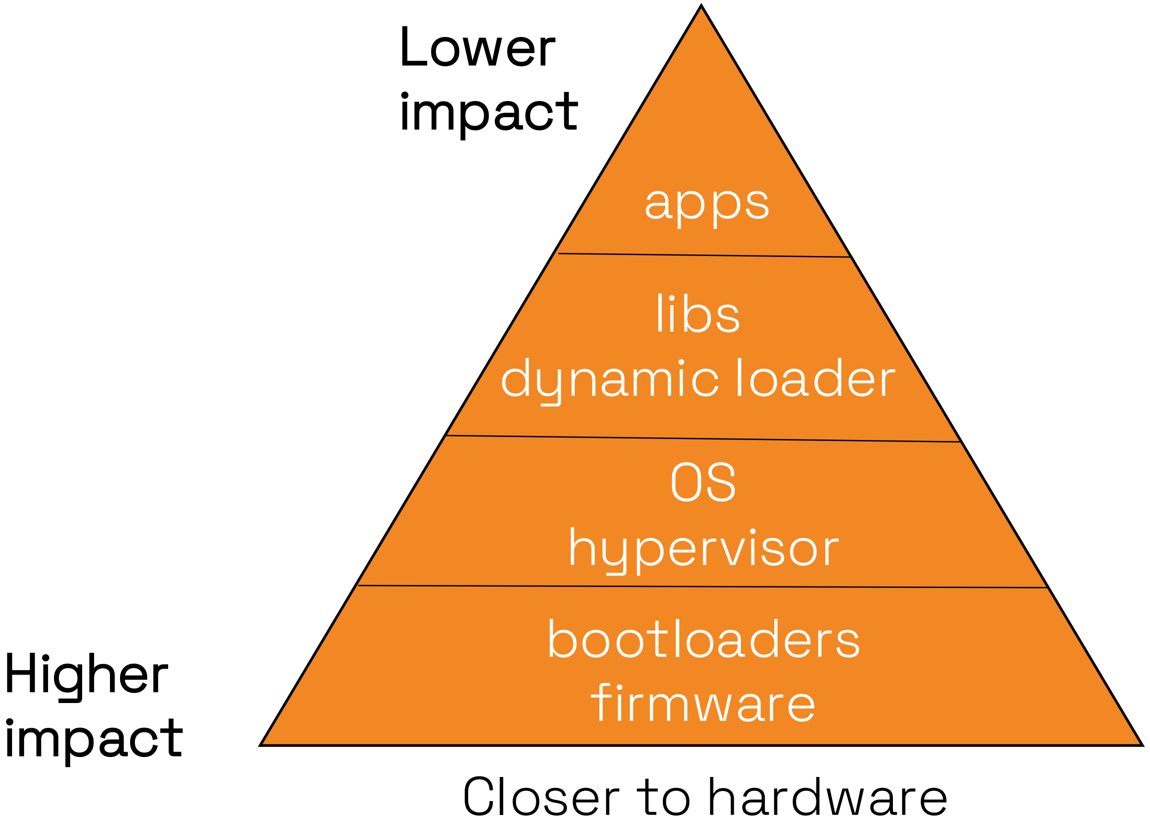
↘

Software strategy

Codasip

→ CHERI software impact

- Any code that manages memory will need to be adapted for CHERI
- The higher the level of abstraction, the less code change required
  - Application code often requires very minimal updates for memory safety
  - KDE required < 0.03% LoC changed
- Exceptions include JIT-based software
  - JIT-enhanced language interpreters



→ Software strategy

To smooth the road to CHERI adoption, we should provide:

- Compilers / toolchains / debuggers
- Firmware – e.g. bootloaders, security monitors
- Operating Systems – Rich OS and RTOS, open source and commercial
- Language runtimes
- Simulators
- Distributions with key infrastructure libraries

→ Current development paths

Three current mainstream development paths:

- Morello : high-end Arm 64-bit research application processor
- CHERI-RISC-V : mid-range RISC-V RV64 64-bit application processor and mid-range RISC-V RV32 32-bit microcontroller based on RISC-V International standardisation work
- CHERIoT : low-end RISC-V RV32e 32-bit microcontroller with own specification (compatible with RISC-V International proposed standard)

(CHERI v9, University of Cambridge research work)

↘

Language support

→ Language support

Language	Morello	CHERI-RISC-V	CHERIoT
C/C++	LLVM15 GCC (early prototype)	LLVM17	LLVM13
Rust	Kent University CyberHive/Embecosm	In discussion	TODO
Java	OpenJDK (Soteria project)	TODO	TODO
Python	Python (University of Cambridge) Micropython (University of Glasgow)	TODO	TODO
Javascript	V8 (Capabilities Ltd.)	TODO	TODO
Ada	Implemented by AdaCore	TODO	TODO

→ C library support

Library	Morello	CHERI-RISC-V	CHERIoT
Newlib	2.4.0	4.4.0	-
Musl	1.2.4	1.2.4	N/A
GLibc	2.39	2.27 (early prototype)	N/A
BSD Libc	YES	(v9, needs ported to new ISA)	N/A

→ Firmware, debugger, emulators

Package	Morello	CHERI-RISC-V	CHERIoT
U-boot	-	2024.10	N/A
OpenSBI	N/A	1.5	-
GDB	11.0.50	14.1	-
QEMU	6.0	6.2	-
MPact-CHERIoT	N/A	N/A	YES
Arm FVP	YES	N/A	N/A

All platforms (Arm Morello, Codasip X730, lowRISC Sonata) have their own platform-specific first stage bootloaders.
RTL emulation (e.g. verilator or QuestSim) is also used for SW development/testing

↘

Microcontroller software

→ Operating Systems

OS	Morello	CHERI-RISC-V	CHERIoT
FreeRTOS	10.4.3	11.1.0	(FreeRTOS compat layer in CHERIoT-RTOS)
Zephyr	-	Planned	-
ThreadX	-	Under evaluation	-
CHERIoT-RTOS	N/A	TODO	YES

↘

Application core software

→ Operating Systems

OS	Morello	CHERI-RISC-V	CHERIoT
CheriBSD	24.05 (FreeBSD 15-CURRENT)	(on v9, needs updated to new ISA)	N/A
seL4	Latest	(on v9, needs updated to new ISA)	N/A
Linux	6.7 (hybrid)	6.10 (purecap)	N/A
VxWorks	YES	-	-

→ CheriBSD

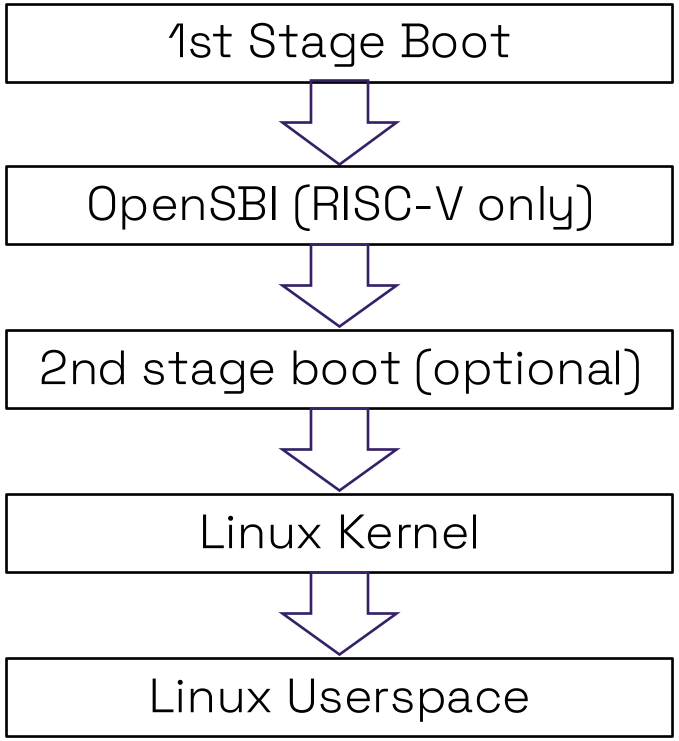
- Most advanced CHERI rich OS
- CHERI-enhanced FreeBSD
- Being ported to new CHERI-RISC-V ISA (supports v9)
- Used as model for Linux development
- Follow at https://www.cheribsd.org/

Features:

- Spatial and temporal safety
  - Temporal for user-level only
- Library compartmentalisation (C18N)
- Hypervisor support
- >10000 memory-safe user-level packages
- In development:
  - Prototype kernel C18N
  - C18N policy framework/tools
  - Colocated process C18N

→ CHERI Linux (kernel)

- Early stage of development
- **Pure capability** v6.10 kernel
- Re-used Morello code where possible
  - Morello Linux currently has hybrid kernel
- First goal is basic spatial memory safety
- Next step is to harden userspace
- Then kernel temporal safety and C18N



Purecap embedded Linux boot flow

↘

The way ahead

→ Software release

https://github.com/CHERI-Alliance

- Architecture-common Morello and CHERI-RISC-V software are being merged on CHERI Alliance Github
- Aligning on purecap Linux kernel (and possibly CheriBSD)
- Align all variants in time
- Defining light-weight governance processes (per open source project)
  - How patches are submitted
  - Issue tracking
  - Patch reviewing processing

Codasip

→

Thank you!

carl.shaw@codasip.com