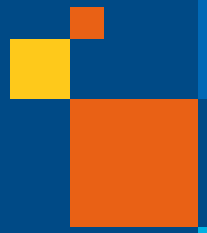
Soft-ISA: kernel built-in emulation engine to extend RISC-V silicon ISA capability

Haicheng Li <haicheng.li@intel.com>

Xiao Wang <xiao.w.wang@intel.com>



intel®

Legal Notices and Disclaimers

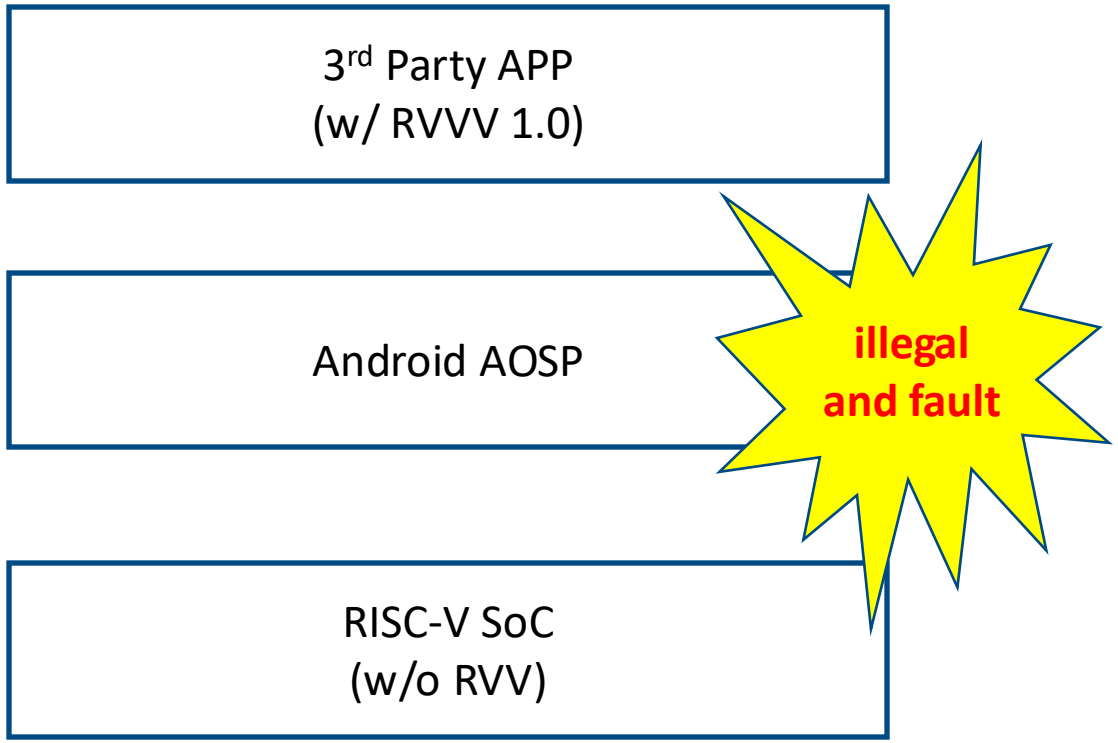
Statements in this document that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.

All product plans and roadmaps are subject to change without notice. Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document. Code names are often used by Intel to identify products, technologies, or services that are in development and usage may change over time. No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

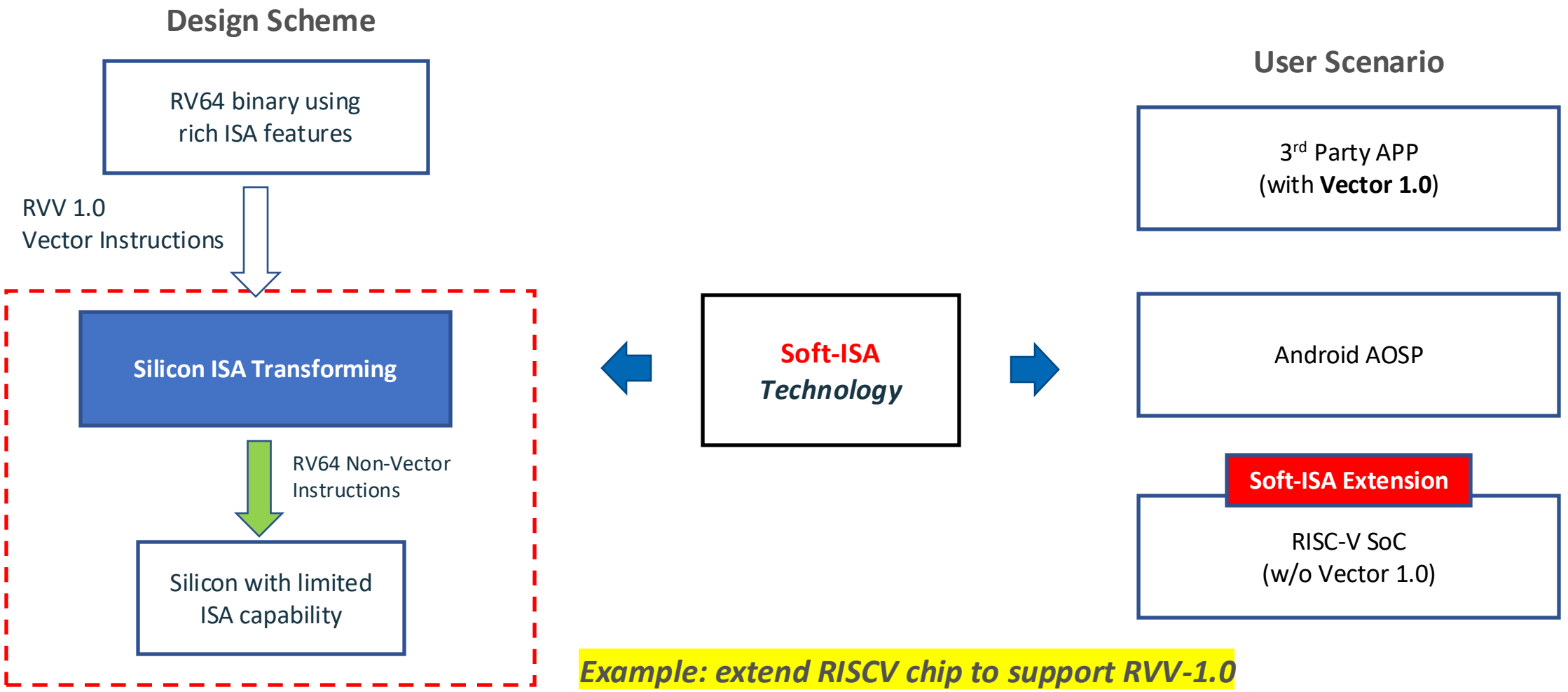
 This document contains information on products and/or processes in development.

Challenge of Compatibility

- **RISCV Advantages**
  - Highly customizable ISA
  - Die size -> PnP efficiency
  - Fast evolving ISA extensions
- **RISCV Pain Points**
  - HW profiles vs. Chip diversity
  - Software compatibility
  - HW readiness vs Fast evolving ISA



Soft-ISA : Software way to extend silicon ISA capability

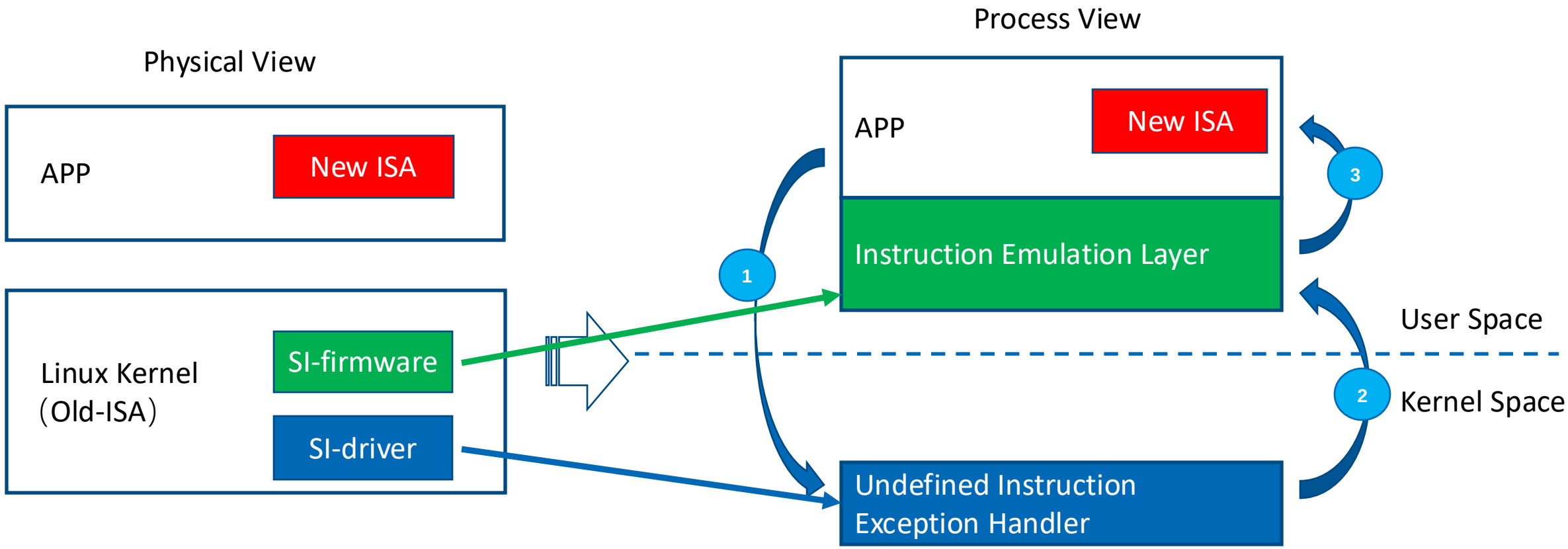


Example: extend RISCV chip to support RVV-1.0

Design Goals

- **APP Agnostic**
  - Maintained by IP/OS vendor
  - Loaded by kernel
  - Mapped into user space
  - Share app’s virtual address space
  - Support signal handling
- **Low Overhead**
  - Adjustable emulation block size
  - Per-thread context (reentrancy)
  - Flexible code jump-in/out
  - Hotspot code identification
  - HW assisted BT

Soft-ISA Conceptual Architecture



1. **APP running new ISA instructions causes illegal instruction exception**
2. **Kernel handles exception and switches context to SI-firmware code mapped in user space**
3. **SI-firmware code emulates a bunch of new ISA instructions (instruction # can be static or dynamic)**

Deployment Model: x86_microcode_firmware-like way

- soft-isa firmware loaded from /usr/lib/firmware/soft-isa/

```
[*] RISC-V CPU instruction emulation firmware loading support
[*]   RISC-V Soft-ISA firmware loading support
[ ]   Late Soft-ISA firmware loading (DANGEROUS) (NEW)
```

- Opt#1: enable kernel support for loading soft-isa firmware via initramfs/initrd

```
General setup  --- >
    [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support (BLK_DEV_INITRD)
```

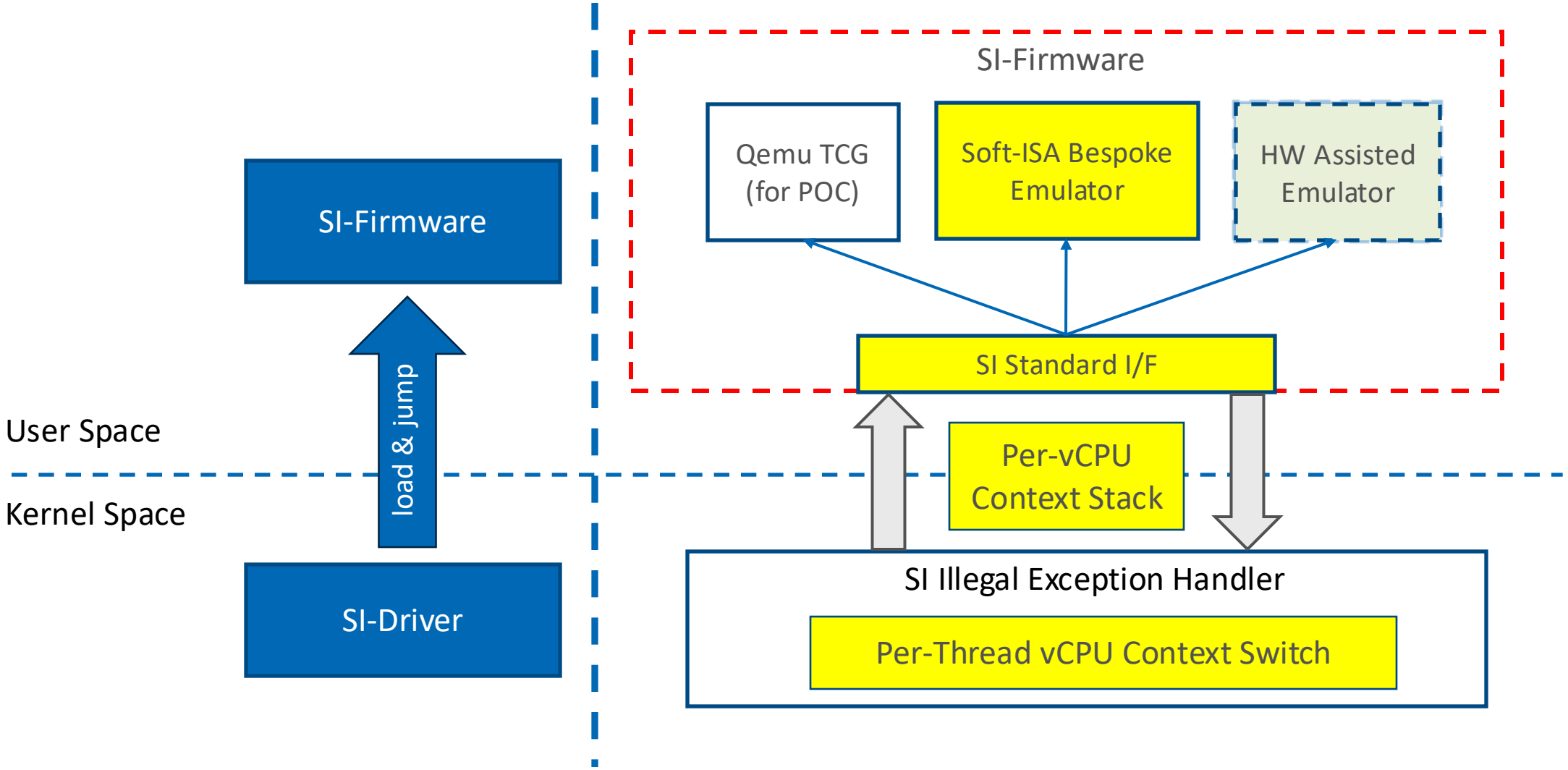
- Opt#2: Soft-isa firmware blobs in-kernel

```
Device Drivers  --->
    Generic Driver Options
        Firmware loader --->
        {*} Firmware loading facility
```

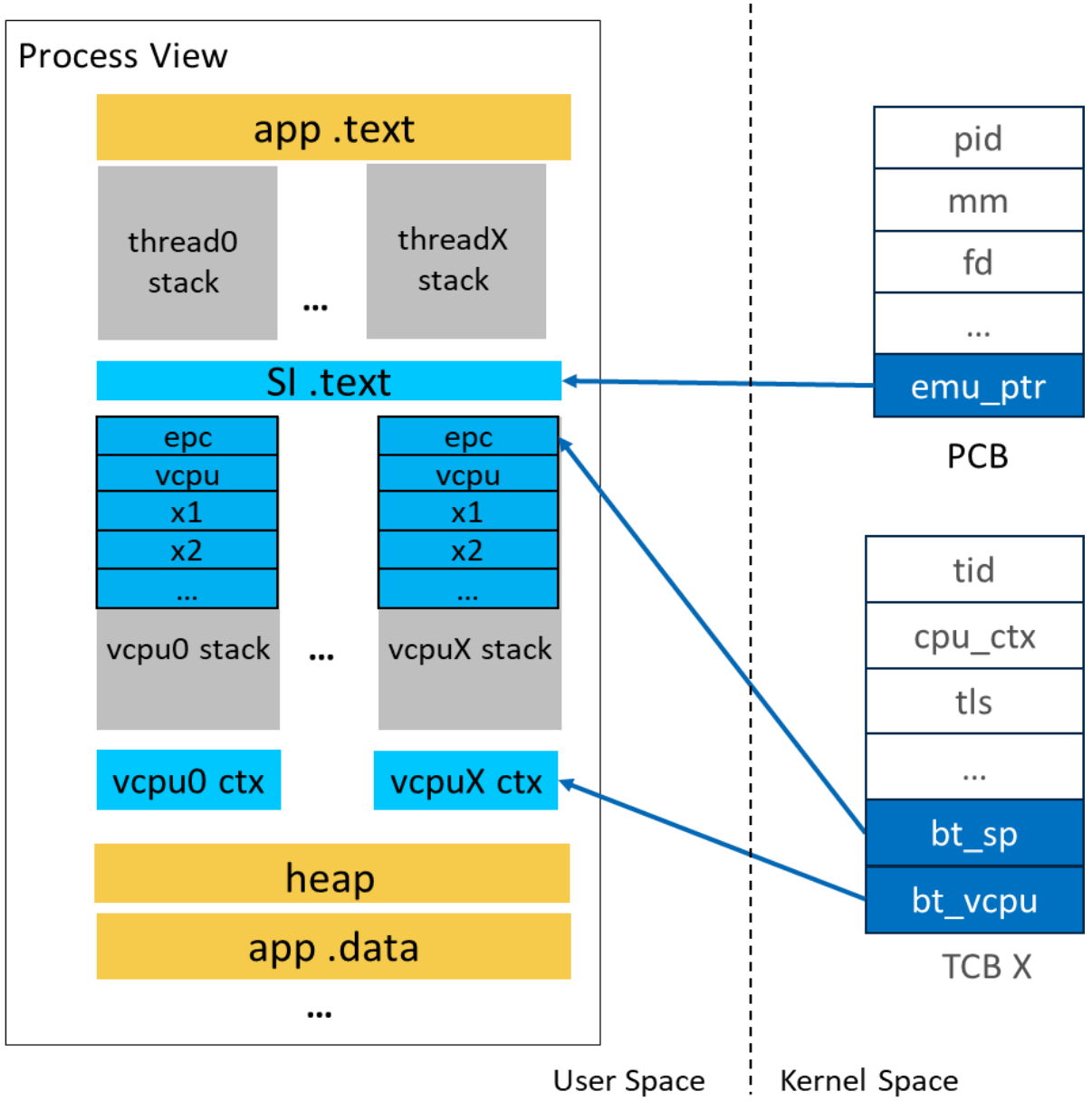
SI-Firmware Upgrade

sudo apt install soft-isa.bin

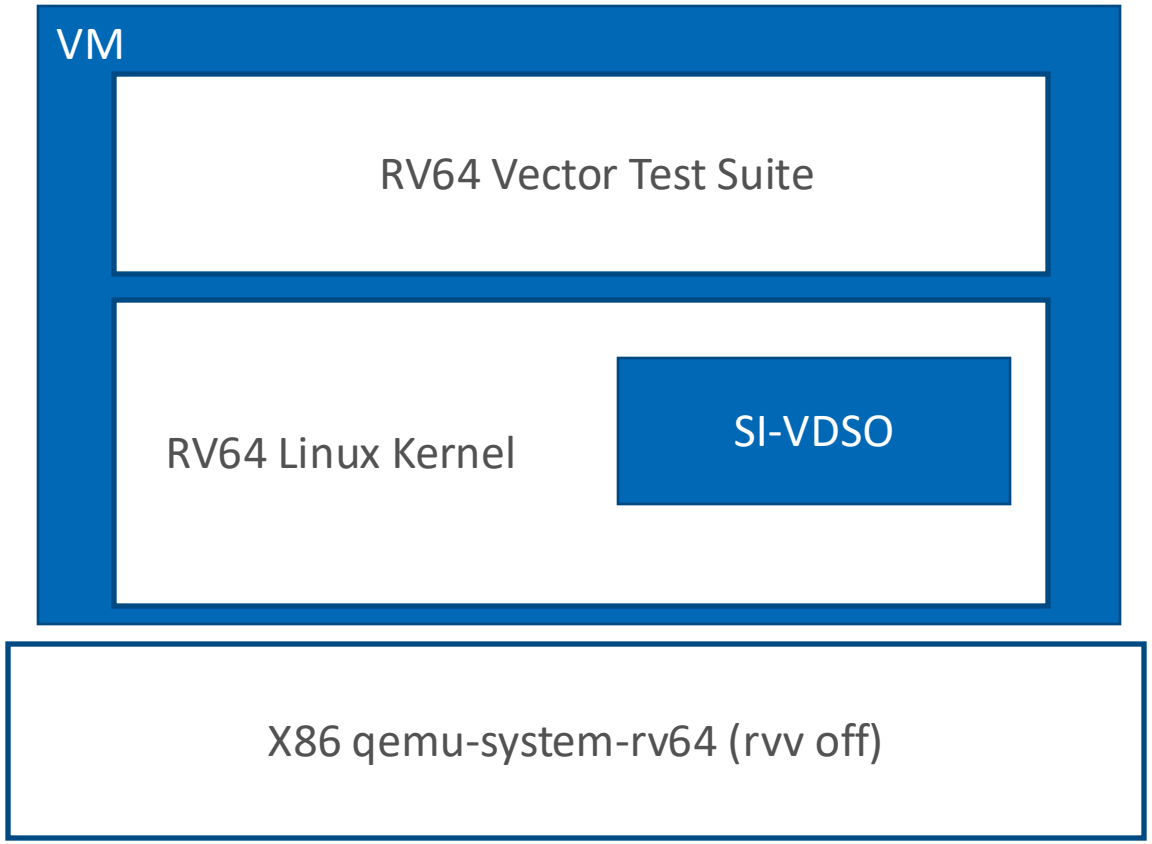
High Level Design



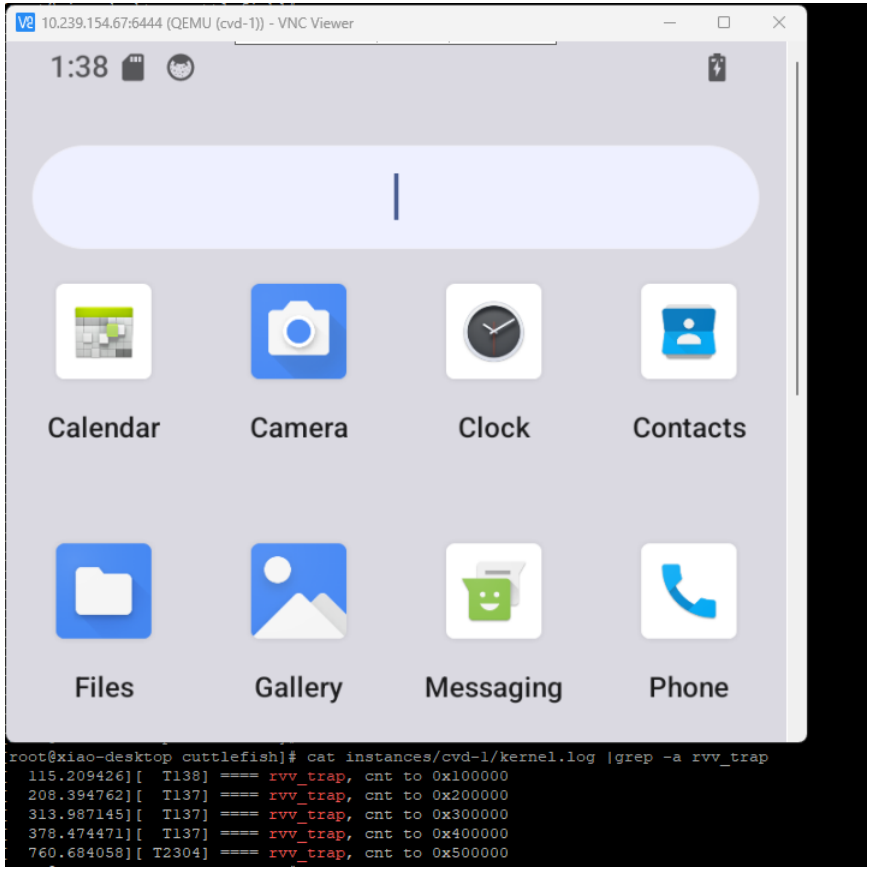
Low Level Design



PoC Status



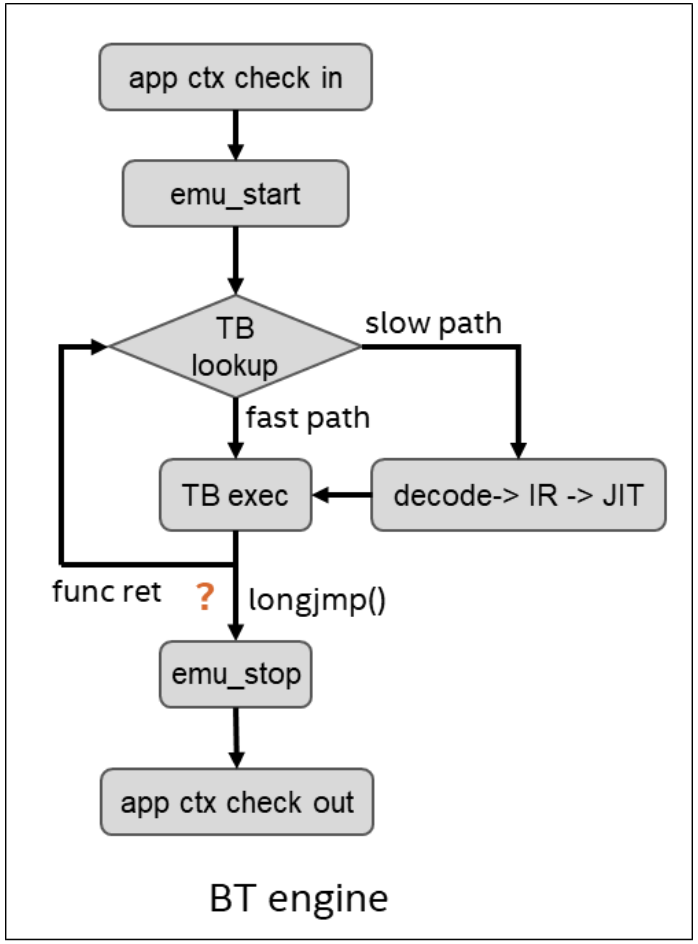
- APP Agnostic
- Multithreading support
- Easy to extend ISA
- Kept synced with RISCV spec

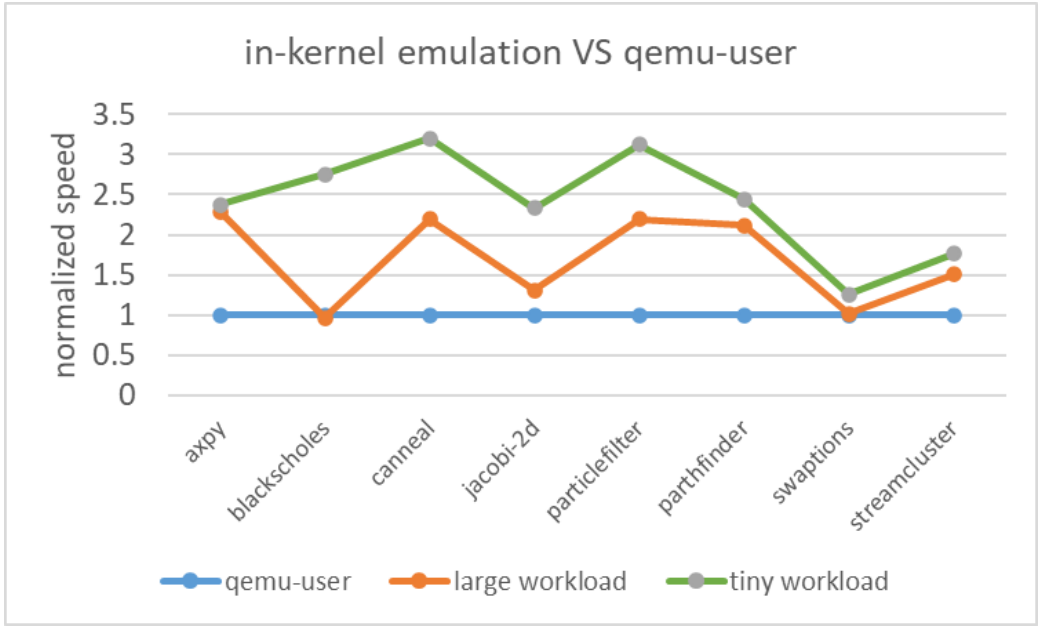


Project Status:

- *Android AOSP bootup with RVV extension emulated by Soft-ISA*
- *Soft-ISA verified also on RISC-V real machine Sophgo-2042 Milk-V*

Soft-ISA Performance



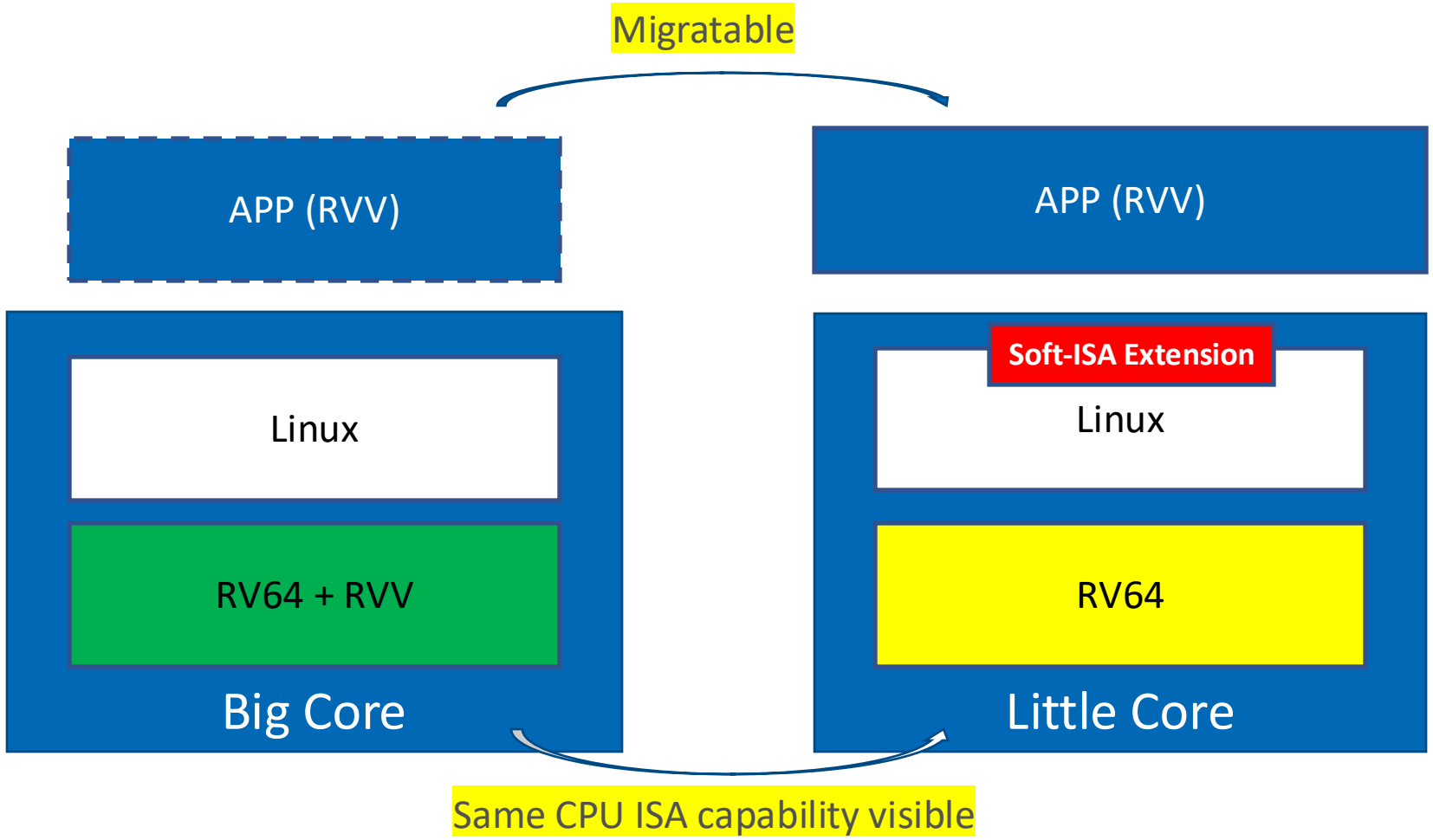


Measured on Sophgo-2042 Milk-V machine

- Generic DBT optimization techniques
- Context switch overhead reductions

Use Case: Improve software compatibility

- Big-Little SoC
- Single Binary
- Run Everywhere



Use Case: Accelerate HW readiness with new ISA support

Silicon

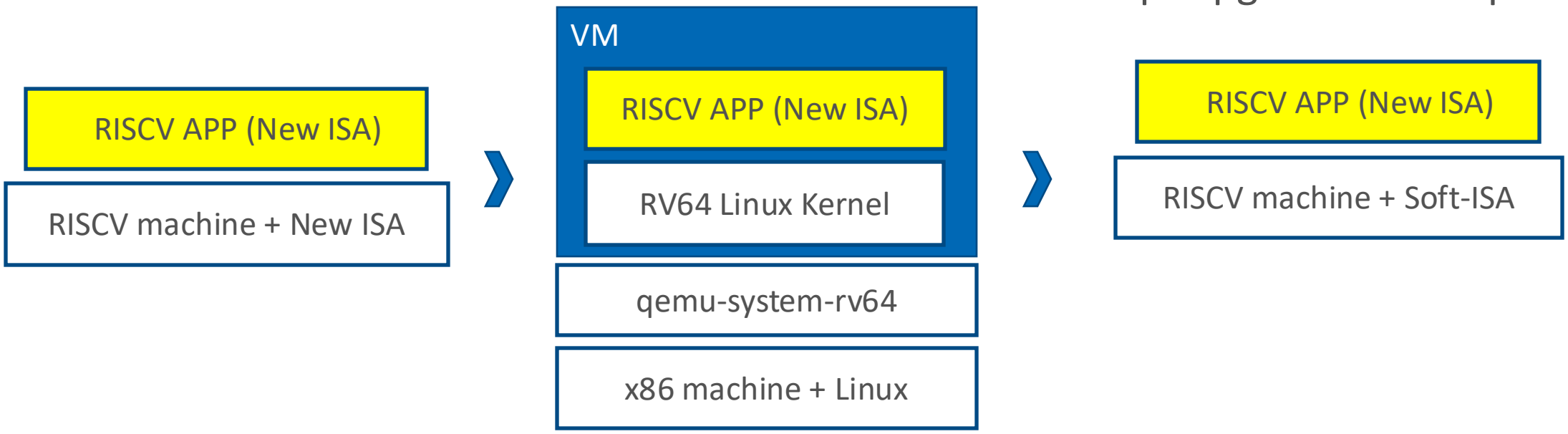
- 2 years late than spec usually
- Best performance

Emulator

- Pre-silicon SW dev
- 3-5 times performance slow-down

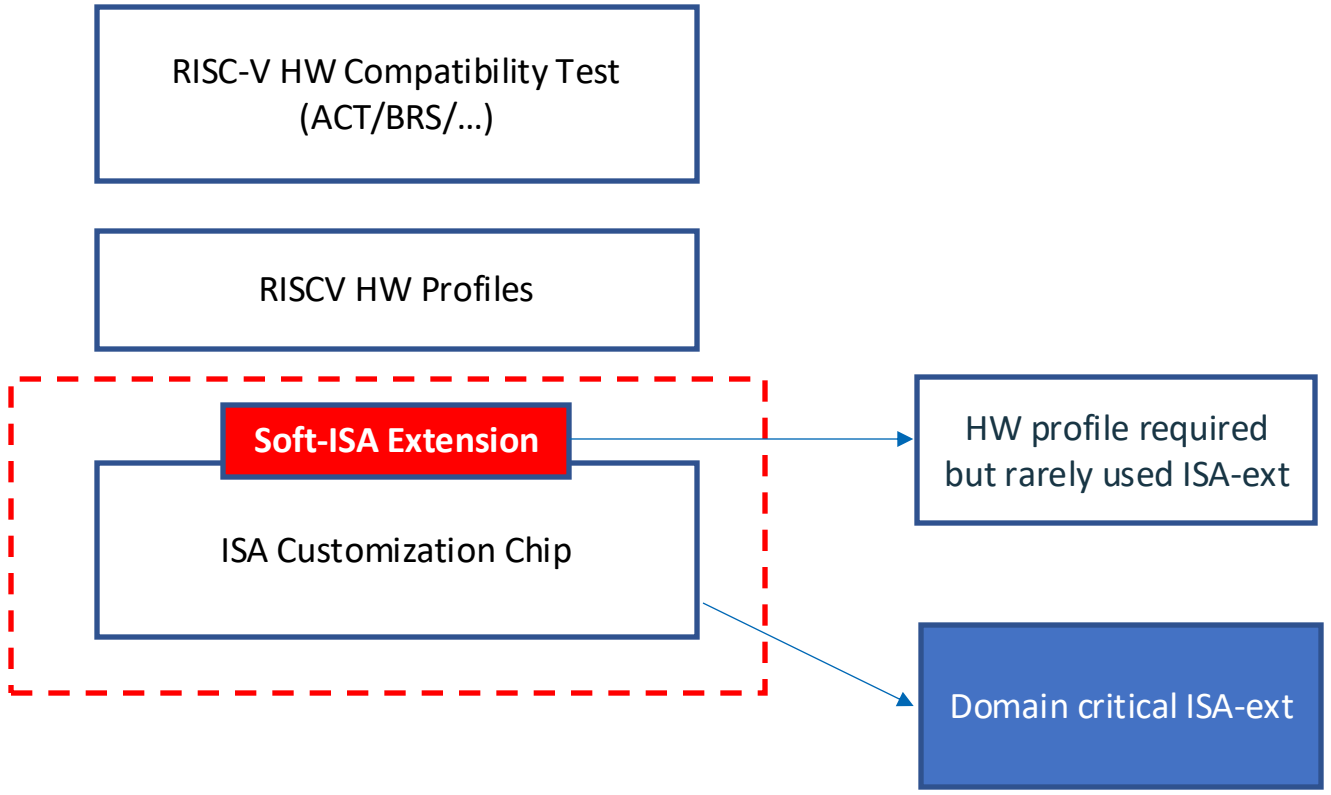
Soft-ISA

- Immediate system upgrade
- Nearly native computing
- Easy to extend ISA
- Kept upgraded with spec



Use Case: Reduce ISA fragmentation & Achieve better PPA

- HW profile compliance via Hybrid ISA support
- SW emulates rarely-used ISA extensions to reduce DIE size
- HW accelerates domain-critical ISA extensions to achieve high Perf/Area efficiency



80% cycles spent on 20% instructions

Thanks!

- https://riseproject.dev

RISE is focused on positive and transparent collaborations with upstream projects to deliver commercial-ready software for various use cases

How: Align on highest priorities & avoid (accidental) duplication of work

Goal: Accelerate open source SW for RISC-V architecture

https://www.intel.com/content/www/us/en/developer/articles/community/rising-to-the-challenge-risc-v-software-readiness.html

Finding more interesting topics from Intel on RISC-V summit China 2024

Topic	When & Where
UXL 软件栈和 RISC-V 的初步探索	August 22 16:45 主会场A
LLVM 工具链 RISC-V 构建实现及其性能优化现状分析与未来展望	August 23 9:40 主会场A
GCC RVV 自动向量化及其应用	August 23 10:00 主会场A
Enhancing RISC-V Security with SBI Secure Service APIs	August 23 10:40 主会场B
Enabling Hardware Sampling Based PGO for RISC-V Platform	August 23 11:40 主会场A
利用 WASM 技术解决多种 ISA 的挑战	August 23 14:20 主会场B
HVP: Hardware Accelerated RISC-V Android Emulator	August 23 14:50 主会场A
Leverage BRS standard to improve RISC-V SW compatibility	August 23 17:30 主会场A
Soft-ISA: kernel built-in emulation engine to extend RISC-V silicon ISA capability	August 23 17:40 主会场A