The Road to RISC-V Server Standardization: UEFI boot, Boot and Runtime Services

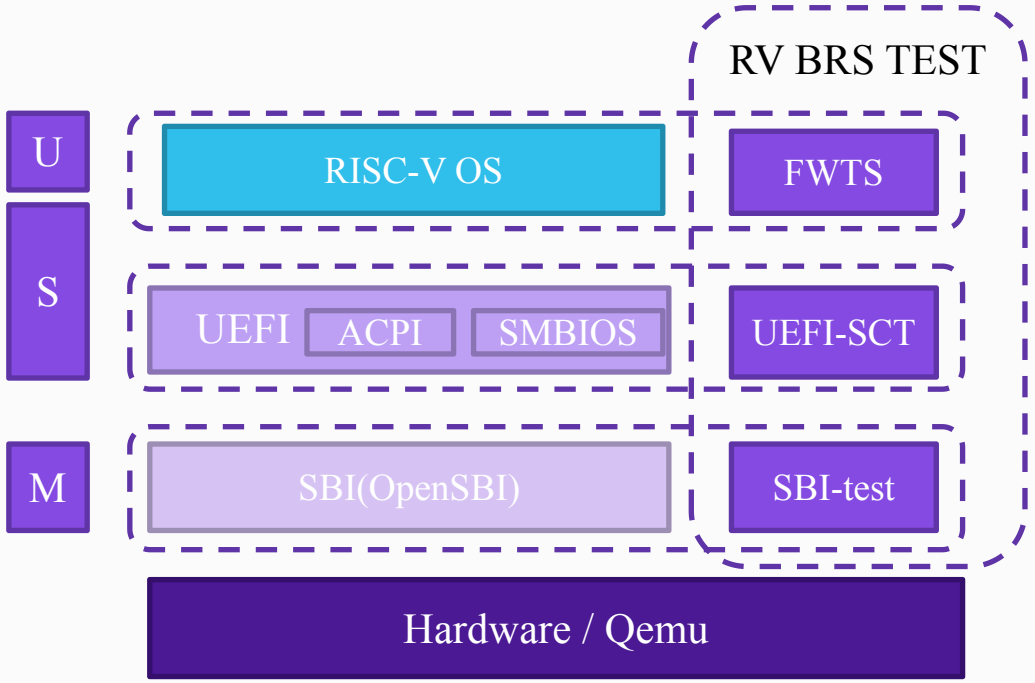
Presenter: Zhen Liu & Evan Chai

Agenda

- UEFI boot & SCT
- Formal Verification of SBI

The Progress of RISC-V Firmware in Community

- ACPI Spec v6.6 was ratified
- BRS Spec v0.9 is in public review
- Server Platform Spec v0.6 is WIP
- RISC-V SBI Spec v3.0 is almost ratified
- Smmtt Spec v0.3.6 is WIP
- Cooperation in the BRS test suite
- Risc Server Cluster and Subsequent Plans



	Profile	UEFI	ACPI（ing）	DT	SBI	SMBIOS（ing）
BRS-L	>= RVA20	>= 2.10	6.6	optional, >= v0.3	>= 2.0	>= 3.7.0 (ing)
BRS-B	>= RVA20	EBBR, >= 2.1.0	optional, >= 6.6	optional, >= v0.3	>= 2.0	optional, >= 3.7.0

Device Tree Specification v0.4

Smbios Specification v3.7.0

ACPI Specification v6.6

riscv-sbi spec v2

UEFI PI Specification v1.8.0

UEFI Specification v2.10

RISC-V boot mode with spec

```
Please choose the step to execute from the following options:
1. Clone Git code
2. Set environment variables
3. Compile tools (BaseTools)
4. Specify platform firmware compilation
5. Compile OpenSBI
6. Compile QEMU RVSP-REF Platform
7. Run QEMU
8. Execute all
Please enter an option (1-8):
```

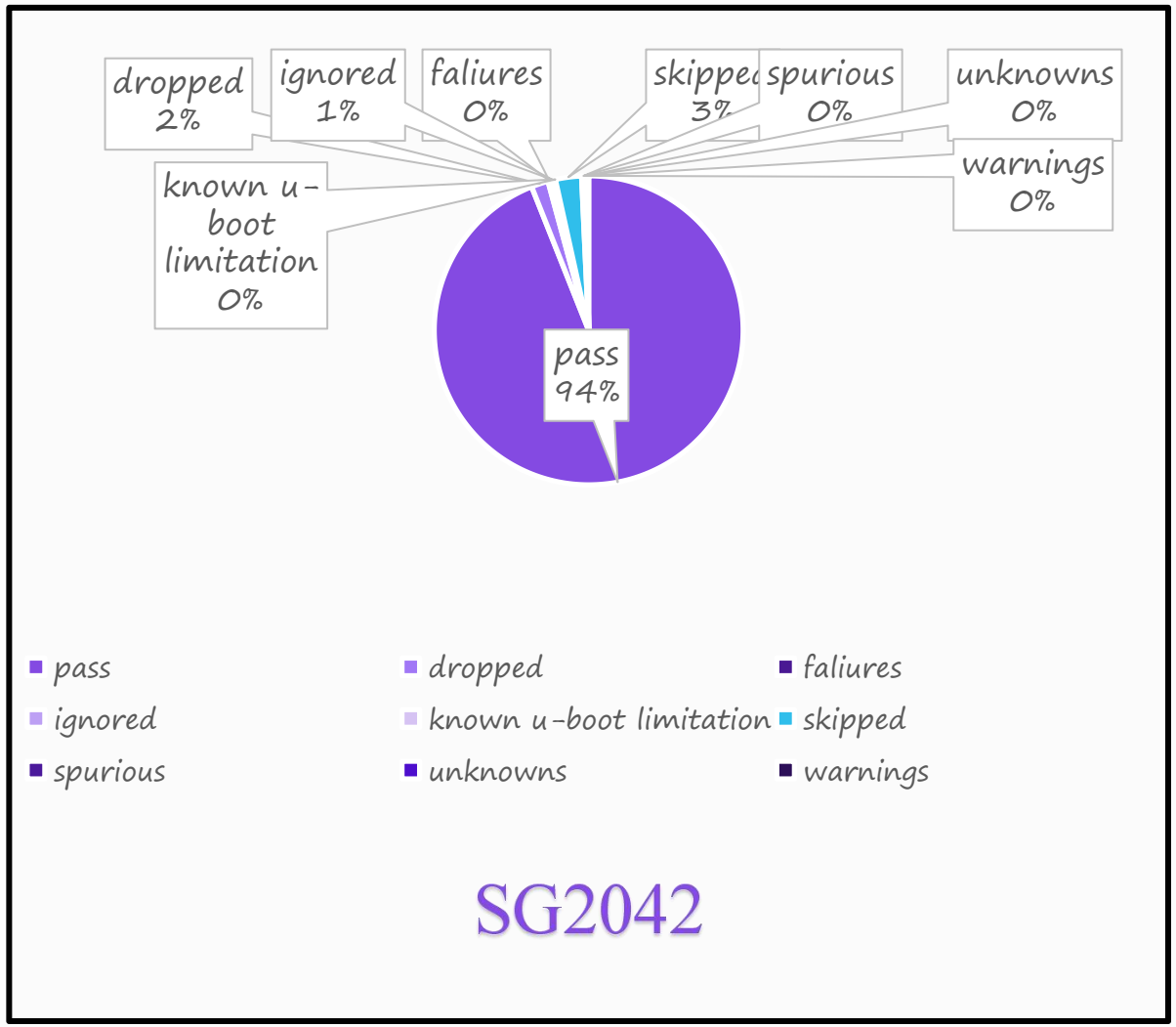
```
===== RISC-V Build Tool v1.0.0 =====
1. Clone All Repositories
2. Build Buildroot
3. Build GRUB
4. Build Linux Kernel
5. Prepare Boot Image
6. Build All
7. Clean Build Directories
8. Exit
============================
Select an option [1-8]:
```

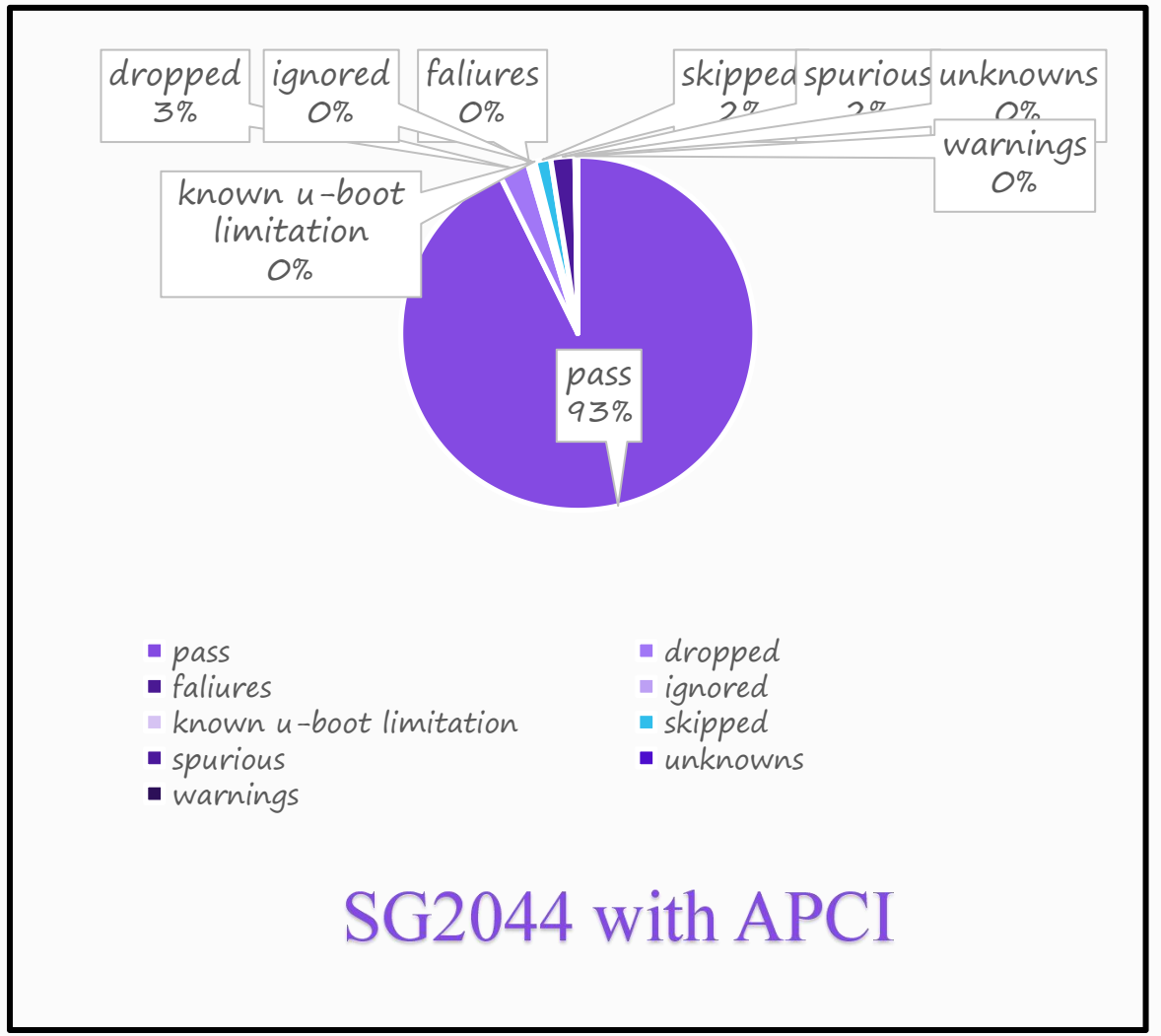
```
Please choose the step to execute from the following options:
1. build SCT test
2. Package the SCT tests into a FAT file system.
3. Run SCT tests on QEMU.
4. Execute all
Please enter an option (1-4):
```

```
1. linux
2. grub
3. edk2
4. edk2-test
5. edk2-test-parser
6. buildroot
7. opensbi
8. sbi-test
9. qemu
10. fwts
C. Compile All Components
0. Back to main menu
Time remaining:  8 seconds | Select option (1-10, C, 0):
```

https://github.com/AII-SDU/rv-sp-test-mod.git

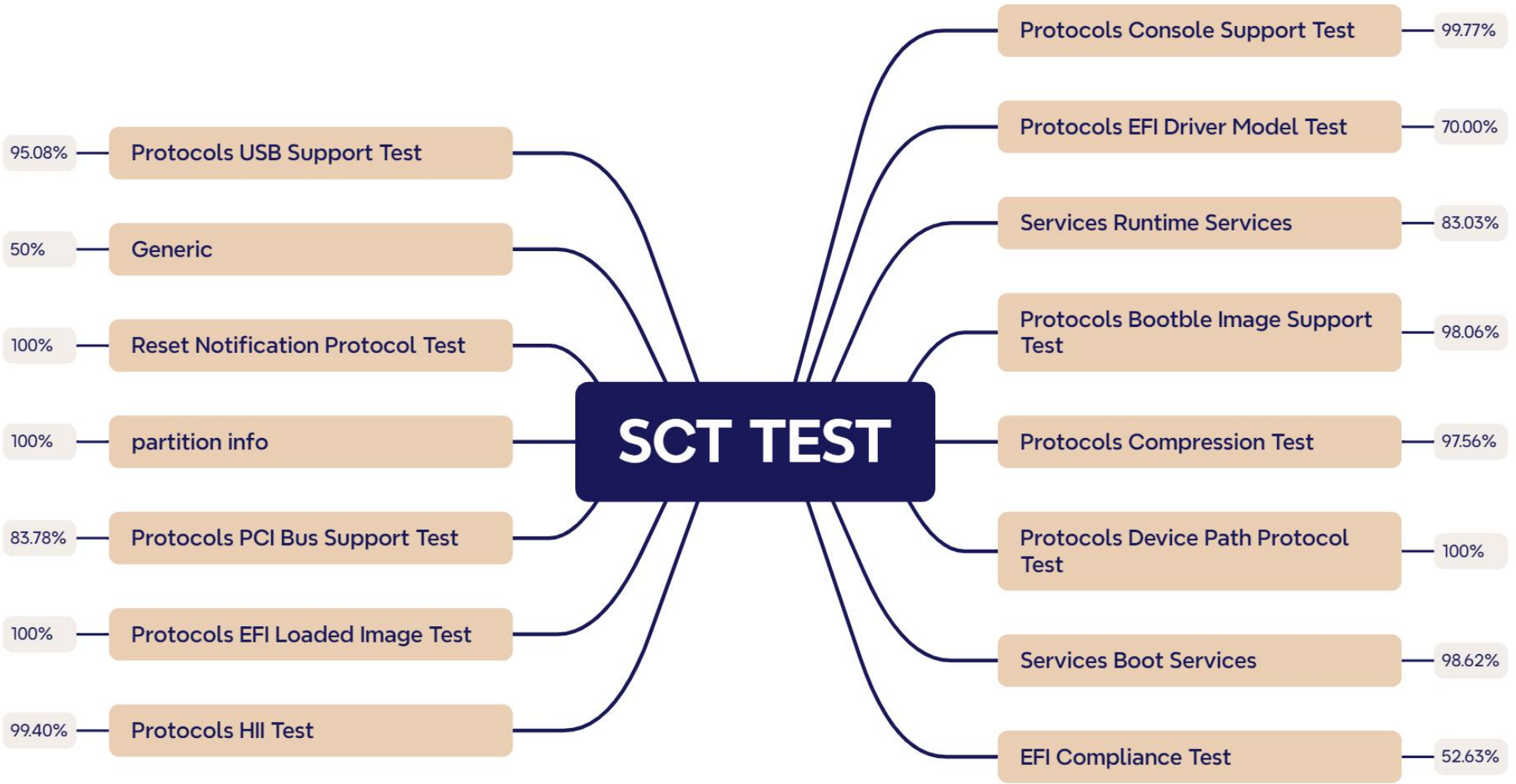
https://github.com/AII-SDU/riscv-brs-tests.git

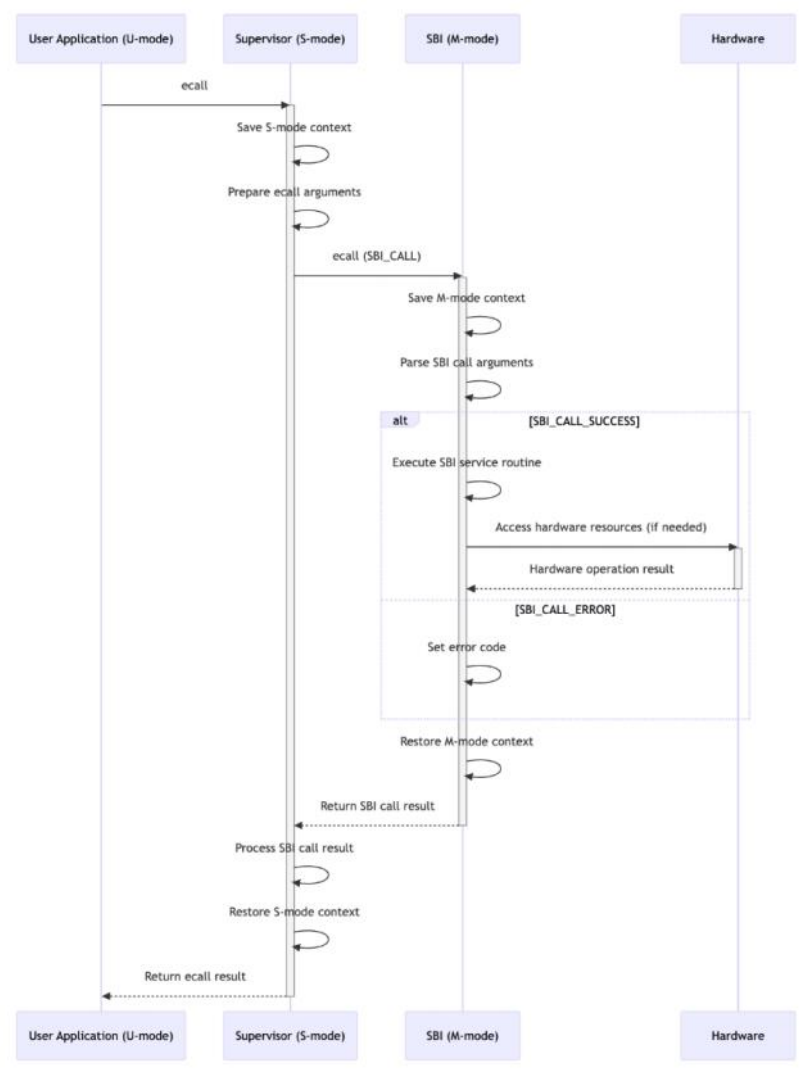




The UEFI Self-Certification Test (UEFI SCT) :

A toolset for platform developers to validate firmware implementation compliance to the UEFI Specification


SCT TEST
95.08%
Protocols USB Support Test
50%
Generic
100%
Reset Notification Protocol Test
100%
partition info
83.78%
Protocols PCI Bus Support Test
100%
Protocols EFI Loaded Image Test
99.40%
Protocols HII Test
Protocols Console Support Test
99.77%
Protocols EFI Driver Model Test
70.00%
Services Runtime Services
83.03%
Protocols Bootble Image Support Test
98.06%
Protocols Compression Test
97.56%
Protocols Device Path Protocol Test
100%
Services Boot Services
98.62%
EFI Compliance Test
52.63%



SBI S/M/U Mode switching

Security. Performance. Proof.

CERTIOS

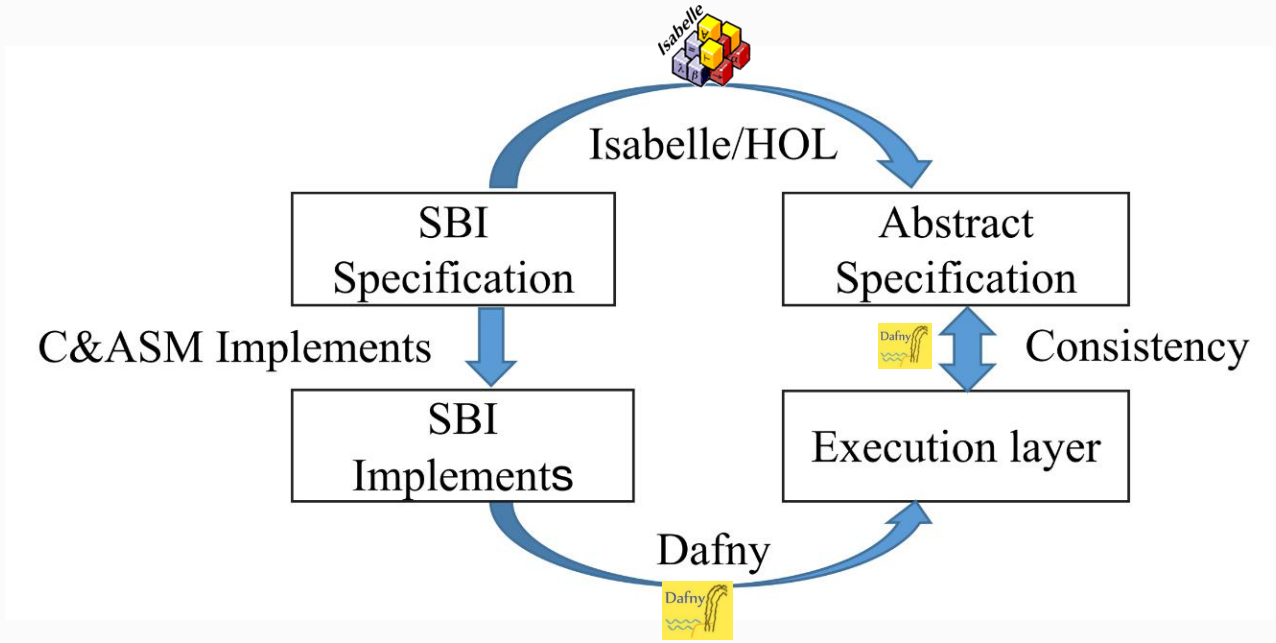
Example: Formal verification of security software

- Mode switching requires security requirements
- The function calls are tight
- There is no method for verification of the RISC-V SBI firmware
- The SBI standard is open source

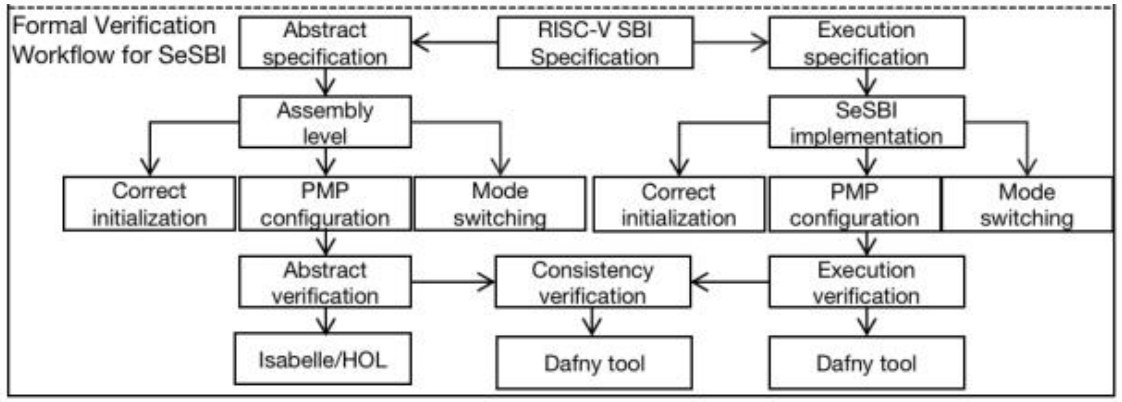
Dafny

Isabelle

Formal verification methods



RISC-V SBI Hierarchical verification framework



RISC-V SBI formal verification flowchart

- Hardware abstract simulation strategy
- The unified privilege level and theoretical interface
- Hierarchical and top-down closed-loop verification

SeSBI:

The first SBI firmware verified for correctness

The specific verification functions involve some key configuration functions such as trap,console,timer,etc.

Thanks