

**\*\*NOTE:** This policy is out-of-date but will not be updated until 2023. It is left here for historical reasons.

# Platforms Policy

Document Version	1.0
Document Issue	Final (In effect)
Issue date	2021-06-14

## Copyright

Copyright © 2021 RISC-V International. All Rights Reserved.

## Document information

Document name	Platforms policy
Version	1.0
Creation date	2021-03-18
Last modified date	2021-06-14
Summary	Rules for platforms' specifications
Author	Dr. Philipp R. Tomsich, Chair of the RISC-V Software Horizontal Committee
Document status	Final (In effect)
Source	<a href="#">link</a>

## Purpose of this document

This document summarizes the policies and procedures adopted for RISC-V Platforms.

A RISC-V Platform specifies a common, reusable execution environment that operating systems and applications can target to improve portability and reuse. A RISC-V Platform provides an interoperability assurance for compatible software (i.e, software that fulfills all requirements of a RISC-V Platform) when run on hardware devices that are also compatible with the same RISC-V Platform.

All RISC-V Platforms are developed and released by the Platforms Horizontal Subcommittee, operating under the auspices of the Software Horizontal Committee.

## Intended audience

This policy document targets the following audiences:

- Contributors, Members, and Leadership of the Platforms Horizontal Subcommittee
- RISC-V technical leadership and program management
- Architects and Integrators of software and hardware solutions for the RISC-V ecosystem

## Table of contents

<b>How to read this document</b>	<b>4</b>
<b>Platforms policy</b>	<b>4</b>
Scope	4
Specification coverage	4
Naming and Versioning	5
Machine-readable identification and experimental versions	5
Claiming Compatibility	6
Structure	6
Deprecation of requirements	7
<b>Platforms release cycle and versioning</b>	<b>8</b>
<b>Platforms Lifecycle</b>	<b>8</b>
Inception	8
Preparatory stage	8
Publication stage	8
Retirement of Platforms	8
<b>Exceptions</b>	<b>9</b>
<b>Glossary</b>	<b>9</b>

## How to read this document

This policy interleaves Rationale and Policy.

For easier readability, the policy elements are **highlighted in bold type**.

## Platforms policy

A solid foundation for software development and interoperability is required to support the growth of the software ecosystem for RISC-V. From a user perspective, binary operating systems distributions (such as the various Linux distributions) should be easily movable between different hardware.

This offers benefits to software developers, who can target an interoperable, common platform with their source code and deliver base functionality quickly (while retaining the ability to provide optimized and value-added functionality after probing their execution environment).

To achieve interoperability, common Platforms are defined that specify:

- features and capabilities that software can expect to be provided by its environment and
- features and capabilities all compliant software must correctly configure and operate.

Both hardware and software are expected to claim compatibility towards a given Platform: such a compatibility claim indicates interoperability for any combination of compatible hardware and software.

## Scope

A Platform is a combination of all the components necessary to construct a working system, and typically consist of some combination of hardware (or an emulation or simulation thereof), firmware, and software and may include a boot loader, system runtime services, a hypervisor, an operating system, and applications. A Platform is an implementation.

A Platform Specification is the description of the semantics of the interfaces between certain specific components of the system. As there can be some ambiguity as to what is considered hardware, firmware, or software (e.g., with certain trap and emulate behavior for certain instructions, or whole system emulation), these terms should be avoided. Instead, the Platform Specification should be described in terms of Execution Environments (as defined in The RISC-V Instruction Set Manual, Volume 1: Unprivileged ISA). An Execution Environment is provided by a lower level implementation and depended on by an upper level implementation. For example:

- an M-mode Execution Environment may be provided by an ASIC and depended on by firmware,
- an S-mode Execution Environment may be provided by an ASIC plus system firmware such as OpenSBI and depended on by an operating system kernel such as Linux, or

- a U-mode Execution Environment may be provided by a complete operating system and depended on by applications such as Firefox.

Each component may be any combination of hardware, firmware, or software.

The focus of the platform standards is software-centric. However, both software and hardware will be affected by the requirements put forth in these specifications: both soft- and hardware will need to claim compatibility against these specifications to ensure interoperability between soft- and hardware products.

**The Platform governs how the software can use the platform, not the implementation details of the underlying hardware platform.** Consequently, a Platform shall avoid performance requirements or whether mechanisms have to be implemented in hardware (i.e. if they can be emulated through trap-and-emulate).

## Specification coverage

**The Platforms released must cover at least the following execution environments:**

- **The “OS-A Platform” specifies a Platform for rich operating systems that target application processors.** It is expected to be augmented with industry-specific Platform extensions for “servers”, “mobile”, “edge computing”, “machine-learning” and “automotive”.
- **The “M Platform” provides a Platform for bare-metal applications and small operating systems on microcontrollers.**

Other execution environments can be added to by the Platforms HSC based on the processes outlined in the section “Platform lifecycle.”

**Each Platform consists of**

- **A base platform**
- **extensions specifications that specify groups of application-specific requirements**

**A base specification and the associated extensions are released and versioned as a single document.**

**Extensions must have broad applicability covering the requirements of entire market segments or industries** (e.g., “mobile,” “automotive,” “server”). Extensions that cover a niche market (or would serve as a marketing tool to individual market participants only) do not justify the resource expenditure for standardization and maintaining requirements and a compatibility test suite.

**An overlap of requirements between extensions is acceptable and is desirable wherever it simplifies the compatibility claim for products.** E.g., from a user perspective, it will be easier to understand if products are compliant “*with the mobile and automotive extensions*”

(both mobile and automotive include virtualization features) rather than “*with the virtualization, mobile and automotive extensions.*”

## Naming and Versioning

The full name of each platform shall be constructed as follows:

- It is prefixed by "RISC-V".
- It is postfixed by its year and a dot-separated revision number (the original issue does not have a revision).

Applying this rule, the following examples result:

- A full release of a platform will be “RISC-V OS-A Platform 2022”.
- The third reissue/revision of the same platform will be "RISC-V OS-A Platform 2022.3".

**Only official Platforms released by RISC-V International can use the “RISC-V” prefix.**

## Machine-readable identification and experimental versions

For machine-identifiable purposes, we use an URI-encoded name, where the scheme is prefixed as 'riscv-platform' for official/standardized platforms and 'x-platform' for experimental versions:

- [riscv-platform://riscv.org/platform/OS-A/2022.3](https://riscv.org/platform/OS-A/2022.3)
- [riscv-platform://riscv.org/platform/OS-A/2022.3/#server-extension](https://riscv.org/platform/OS-A/2022.3/#server-extension)
- [x-platform://semiconductor-company.example.org/embrace-and-extend/007](https://semiconductor-company.example.org/embrace-and-extend/007)

Third parties cannot use the 'riscv-platform' scheme (just like they cannot use 'RISC-V' as part of their platform name).

**Replacing the 'riscv-platform' or 'x-platform' in the URI with 'https' shall result in a valid URL that hosts the specification and ancillary documentation.**

## Claiming Compatibility

Products implementing a RISC-V Platform shall claim compatibility with a Platform and any applicable extensions that the product implements.

The rules for compatibility testing are designed to ensure a surjective compatibility mapping, but not to enforce a bijective compatibility between platforms and software: software targeting the base Platform must also execute in the presence of any extensions to this Platform (but will not support the additional features introduced by the extension), while software targeting a specific extension may require this extension to be present (i.e. it is not required to run on the base Platform).

**A Platform (i.e. hardware/execution environment) product compatibility claim can only be made if a product satisfies the following:**

- all requirements of the respective base Platform; and
- all requirements of each extension the product claims compatibility with.

**No Platform (i.e. hardware/runtime) product shall claim compatibility with an extension if it is not compatible with the respective base specification.**

**A Software product claiming compatibility with a Platform (and extensions) must satisfy:**

- all requirements of the Platform and of all Extensions that it claims compatibility with.

**These two requirements translate to the following compatibility relationship:**

- Any software that works on the base-platform, will also work in the presence of extensions (i.e. extensions are “true” extensions for software-compatibility).
- Any software that requires an extension, may not be compatible in the absence of the extension.

Any compatibility claim must identify the Platforms including their version number.

For the self-certification of compatibility, corresponding Platform Compatibility Tests (PCT) shall be developed and published. Refer to the Platform Compatibility Testing Policy for details.

In order to declare that you are platform compatible (e.g. RISC-V OS-A Platform 2022 compatible) and use the RISC-V Platform Compatible logo, you must pass the compatibility tests (including the profile compatibility tests for the profile included in the platform spec). After passing the PCT, please follow the steps at the following RISC-V website [www.riscv.org/TBD](http://www.riscv.org/TBD).

## Structure

**Platforms consist of:**

- **Requirements (normative), made up of one or more:**
  - **Mandatory subclauses**
  - **Deprecated subclauses** (see below for the meaning of Deprecated)
- **Rationales (informative)**
- **Application notes (informative)**

A requirement may be made up of multiple subclauses that are combined either as “any of” (“or”), or “all of” (“and”). This affects compatible soft- and hardware as follows:

- If the software must support “*A or B*,” then hardware must provide “*A and B*.”
- If the software must support “*A and B*,” then hardware must provide “*A or B*.”

An example of subclauses and of joining subclauses is:

*[Requirement 1]*

*Compatible software for the OS-A Platform must support ALL OF the following:*

- *[Requirement 1, subclause 1: DEPRECATED]*  
*All interrupts in the system are managed by an interrupt controller compatible with the PLIC specification.*
- *[Requirement 1, subclause 2]*  
*All interrupts in the system are managed by an interrupt controller compatible with the AIA specification.*

**Every element shall have the following annotations:**

- **A unique number** (which is not reused, even if requirements are removed in subsequent versions). Having a unique identifier is critical to trace requirements in Platform Compatibility tests, in discussions on Errata, or to reference Rationals and Application notes back to Requirements.
- **Subclauses are numbered hierarchically within each requirement.**
- **Rationales and application notes must reference the corresponding requirement or subclause.**

## Deprecation of requirements

Platforms address both *forward compatibility* and *backward compatibility*:

- *Forward compatibility*: Products compliant with the current version of a platform specification are interoperable with products compliant with future versions of the platform specification.
- *Backward compatibility*: Products compliant with future versions of a platform specification should also support earlier versions.

*Forward compatibility* defines requirements on how the specifications manage required features. Removing a required feature will break *forward compatibility*; hence specifications shall not remove features without prior warning.

**The following deprecation policy applies for requirements:**

- **MANDATORY subclauses have to be retained for at least one full release cycle of the specification.** E.g., a MANDATORY requirement from a 2022 Platform cannot be removed from the 2024 Platform, but can be made DEPRECATED in the 2024 Platform.
- **DEPRECATED subclauses can be dropped from the next full release of the specification.** Note that a DEPRECATED subclause only signals the intent of dropping the requirement, but does not imply a commitment to drop it based on any specific schedule (e.g., delays in the specification of alternate mechanisms may affect the ability to drop a requirement).



**Dropping a requirement from the specification does not require future products to drop the respective feature, as long as the feature is not incompatible with any new requirements.**

## **No non-obvious requirements**

Platforms will frequently reference third-party documents, specifications and standards. This introduces the risk of affecting non-obvious requirements for Platform compatibility, if those external documents do not follow the same documentation conventions or—in turn—use references to other documents.

**The Platforms shall add clarifying language to avoid non-obvious requirements resulting from third-party specifications. If necessary, the list of mandatory requirements introduced through any document reference must be repeated in the Platform specification.**

## **Platforms release cycle and versioning**

**Major versions of platform specifications are published in a bi-annual cadence for even years.** While no major revisions of the platform specifications will be published in odd years, additional extensions can be added in these years and amendments are made to bring the Platforms up to date with new Profiles.

Amendments and new extensions are published as-needed.

## **Platforms Lifecycle**

### **Inception**

**A new platform specification or an extension can be proposed to the Platform HSC by:**

- **The community at large**
- **The Software Horizontal Committee, the TSC, or the CTO**

**Any new Platform must target a market segment where interoperability is desired, and the industry has sufficient demand to ensure that multiple implementations (both hardware and software) are expected.** Platform specifications that are of fringe benefit or would serve only as a marketing tool for implementers of specific solutions are not to be considered.

**Community proposals are advanced through an inquiry process within the Platform HSC to clearly define the scope, use cases, and affected hard- and software products.** Following this inquiry process, the Platform HSC submits the proposal—including a schedule to release—to the Software Horizontal Committee for resource and schedule approval.

## Preparatory stage

If resources and schedules are approved, the Platform HSC drafts a specification document. After completion, it is submitted to the Software HC for review and approval.

## Publication stage

After approval by the Software HC, it is published and enters into immediate effect.

## Retirement of Platform Specifications

Corrections are not issued to update information that has become outdated since publication. The corrections are mentioned in the Front Matter of the corrected version.

**In general, a correction will not be issued for a publication that is older than three years.**

## Exceptions

Implementations (both hardware and software) may decide not to be compatible with any Platform, as long as no misleading compatibility claim is made:

- **Products may not claim compatibility against any of the Platforms for which they do not fulfill all requirements and pass the Platform Compatibility test.**
- **Products may extend on the functionality of the platform's specifications and provide additional functionality, as long as they remain compatible** (i.e., they may not implement incompatible features unless these are disabled by default).

## Revision history

<b>Date</b>	<b>Version</b>	<b>Changes</b>
2021-06-14	1.0-draft-5	Relax/reword requirement for having separate tables clarifying HW and SW requirements.
2021-06-11	1.0-draft-4	Collected changes from review by the joint Platforms HSC and Software HC leadership teams. Relax bijective requirement on extensions being true supersets.
2021-05-13	1.0-draft-3	Major rework, reflecting the following changes <ul style="list-style-type: none"><li>• Platforms Naming</li><li>• Compatibility Testing</li><li>• Explicit requirements for software and hardware</li></ul>
2021-04-29	1.0-draft-2	Partial update with results from leadership discussions
2021-03-21	1.0-draft-1	Draft version for review by the committee leadership.
2021-03-15	1.0	Initial public release.