“CHERI Enabled” program

Program description and guidelines

Version	1.0
Author	Certification WG
Date	25/03/2026

Content

Introduction

About this document

This document is designed to explain what is expected from organisations applying for the **CHERI Enabled** logo for their products and gives guidance to help them quickly go through the process and provide the right information to be successful.

This document will evolve with the certification program and will be improved and complemented as the program covers different type of products. It defines the certification process, integrating requirements, verification, assessment, application process, and lifecycle management.

Please always refer to the current version of this document, which will be the one applicable at any given time. You can find it here: https://cheri-alliance.org/cheri-enabled

The **CHERI Enabled** logo

The **CHERI Enabled** logo can be used along with the marketing material of products which have successfully gone through the certification program. Like other CHERI Alliance marks, it is subject to the organisation's trademark policy: https://cheri-alliance.org/trademark/

About the certification program

The CHERI certification program is designed to improve continuously as:

- new learnings emerge from previous applications
- new standards get adopted
- new test suites get developed

It is therefore expected that certification should be reviewed every **2 years** (the Expiration Period), to ensure that the improvements of the program are reflected in the listed products.

Key warnings

The **CHERI Enabled** logo indicates that the CHERI Alliance believes that, **according to the evidence presented by the applicant**, a product is applying the right security principles from the CHERI technology.

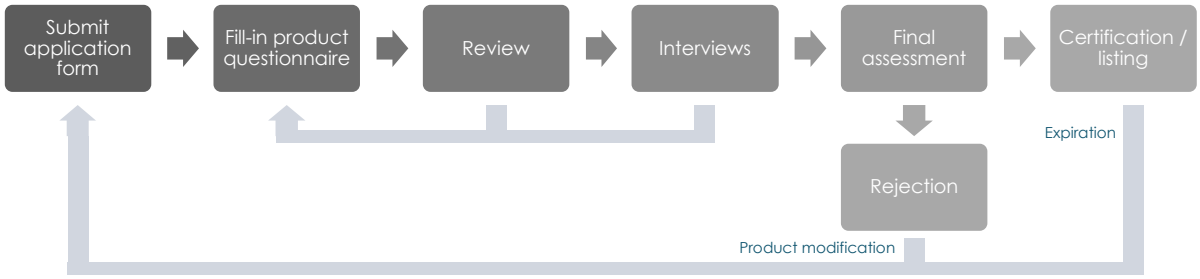
⚠ At this point, the CHERI Alliance does not yet conduct independent testing of the product. Instead, it relies on the applicant's **good faith** responses in the questionnaire and during interviews. If a later review of that product by another party finds that the applicant intentionally provided misleading information, the organisation may be barred from future certifications applications.

⚠ The **CHERI Enabled** logo **does not** guarantee the overall security of a product. Comprehensive security assurance requires a full evaluation of the product, for which CHERI certification materials may serve as useful inputs.

Process

Overview

The process to get a product certified as **CHERI Enabled** is depicted below. The diagram shows the iterative nature of the review and interview processes, that will likely require additional precisions in the product questionnaire.



Products that go through this process for the first time require a full assessment, while the process is simplified for a re-assessment of the same product (new version of the program, product update, …), or a similar product (different configuration, modification, derivative…).

Application form

This form helps identifying the product to be certified and give contact details. This is important as part of the process consists in interactions with the design team.

The main sections of this form are:

- **Organisation** requesting certification – this is open to non-commercial organisations and communities as well
- **Contact person** – note that certification will certainly require experts to be involved in the interview process, but one person should be the main contact to coordinate communication
- **Product** to be certified – identification, type of product, version, description

The form can be found on the CHERI Alliance website: https://cheri-alliance.org/cheri-enabled

It needs to be submitted, along with the payment of the certification fees, to start the whole process.

Certification fees

Application for a product certification is subject to a certification fee, that helps the CHERI Alliance in its mission. This fee is non-refundable.

The fee is reduced for organisations that are members of the CHERI Alliance, and a lower fee is applied in the case of a re-certification of a previously certified product, or a derivative of a certified product.

The applicable details of certification fees at any given time can be obtained from the CHERI Alliance website: https://cheri-alliance.org/cheri-enabled

In the exceptional case of a serious doubt in the assessment, additional costs could be incurred by the applicant (cf the Review and interviews section).

Review and interviews

Responses provided by the applicant are reviewed by a pool of assessors, who are named by the Certification Working Group. They go through the details of all answers and given evidence and interact with the applicant to get clarification or additional details about

points that do not convince them. The final answer to the questionnaire should include these additions.

It is essential to understand that some assessors may be potentially employed by companies competing with the applicant's one. The key principles governing the discussions are as follows:

- Data submitted by the applicant must be **non-confidential** and will be treated as such by the assessors. When certification is granted, the submitted answers will in any case be published publicly
- All parties must comply with the CHERI Alliance **policies**, and in particular:
  - Code of conduct – https://cheri-alliance.org/code-of-conduct/
  - Statement of competition compliance – https://cheri-alliance.org/competition-law-statement/
- Assessors have a duty to review the application only on its technical merits, regardless of any commercial consideration
- The Certification Working Group should control the work of the assessors

Assessors will typically conduct interviews with applicant companies' experts to investigate certain questions further and gain confidence that the applicant has provided accurate and complete information. If an exceptional and substantial doubt is identified that could alter the certification outcome, an independent 3rd party may have to be appointed to review confidential material. The associated costs should be borne by the applicant.

Final assessment

The final assessment is done by the Certification Working Group, using inputs from the assessors. The decision could be:

- **Rejection** – the product is not approved for certification, and a written report is sent to the applicant to explain the reasons for the decision. No public communication is done about this decision, and the applicant only has the possibility to restart a new application after the product has been modified to comply with the issues identified by the assessment
- **Approval** – the product is approved for certification

Listing and marketing

All **CHERI Enabled** products are listed on the CHERI Alliance website – see https://cheri-alliance.org/cheri-enabled. Each product in the list has a separate page where the details of the certification are available:

- **Product description** – precise identification of the product, as entered in the application form
- **Certification details** – date where the certification decision has been taken, version of the certification program
- **Questionnaire answers** – final version of the answers to the questionnaire

Once a product has been approved for certification, its related marketing material can bear the **CHERI Enabled** mark. When these documents are in a digital form (online or offline, e.g. in a PDF document), a link to the CHERI Alliance website must be included, pointing to:

- either the **certification section** of the CHERI Alliance website
- or the **product's page** in the certification section of the CHERI Alliance website

After the Expiration Period, a product needs to go through a light re-certification to maintain its listing, in order to comply with the latest version of this document.

Product questionnaire

The product questionnaire is the key input to the assessment. It is important to provide detailed responses to give confidence to the assessors. Answers to this questionnaire will be published on the certified products' page.

The questionnaire depends on the type of product, and the relevant sub-section should be used for the product to be certified. The **current version** of this specification supports the following products:

- Processor IP
- *Other product types (e.g. chips, tools, software…) will be added in future releases of this document*

Processor IP questionnaire

The processor IP questionnaire consists in the questions detailed in the following subsections. It is applicable to all types of processors (CPU, DSP, GPU…).

Q1 – Configuration space

Describe **the full list** of configuration options and possible values or ranges for each of them, that describe the product being considered.

The configuration space will be discussed with assessors. Outside of this space, a re-certification will be needed.

Configurations points that do not affect security properties (e.g. cache size) are typically accepted as variations of the same certified product. However, more impactful settings (e.g. 32/64-bit instruction set selection) would require re-certification.

Q2 – What CHERI ISA do you implement and what extensions do you have?

Clearly identify the base ISA and CHERI variant, list any standard extensions (e.g. M, A, F, D for RISC-V), and state whether any additional experimental CHERI extensions are implemented. The applicant should demonstrate that they understand the CHERI extension set and are not using an ad-hoc capability scheme.

Recommendations

- State exactly which CHERI ISA **variant** and **version** is implemented (e.g. RV64Y with specific extensions or CHERIoT v1) and provide a **link** to the relevant specification
- List the supported standard **extensions** (e.g. CHERI-RISC-V's Y extension plus MAFD, or Morello with SVE)
- Indicate whether the design is pure-capability, hybrid, or an inhouse capability architecture
- Leave no ambiguity about CHERI compliance

Examples

	Good	Bad
Example answer	Our processor targets the RV64 base instruction set v0.9.7 *[link]* with the standard RISC-V M, A and F extensions. We implement the CHERI-RISC-V (Zcheri) extension to provide capability support. The design supports both hybrid and pure-capability modes, and includes the sentry capability mechanism as defined in the CHERI specification.	We use a RISC-V core with some capability support and additional extensions.
Why is it good / bad	Gives all recommended details	Provides little or no detail, leaving evaluators guessing about the ISA variant or extension set

Q3 – What operations can access memory without an explicit capability operand?

Demonstrate a firm understanding of CHERI's intentional use principle. In CHERI, memory accesses should be authorised by explicit capability registers. Exceptions exist in certain hybrid-mode or privileged cases; good answers should identify these and justify them.

A test suite is available to help applicants provide evidence. This may need to be adapted to the case of the IP being assessed. The source code can be found here: https://github.com/CHERI-Alliance/certification-test-suite

Recommendations

- State explicitly that none of the user-mode instructions can dereference memory without a capability operand, or lists only tightly controlled exceptions
- Explain that legacy load/store instructions in hybrid mode implicitly dereference the Default Data Capability when they cannot name a capability
- Note that any debug or privileged operations that bypass capability checks are disabled in normal execution or restricted to firmware
- Leave no ambiguity about when implicit capabilities are used
- Provide execution logs from the test-suite runs

Examples

	Good	Bad
Example answer	In pure-capability mode, every memory operation uses an explicit capability register. In hybrid mode, legacy RISC-V load and store instructions implicitly dereference the Default Data Capability	Our loads and stores can access memory directly without capabilities if needed.

	because they cannot name a capability. No other operations bypass capability checks.	
Why is it good / bad	Gives all recommended details	Shows ordinary loads / stores can run without capabilities, undermining CHERI's safety guarantees Provides no constraint or justification for exceptions. Leaves evaluators unsure of the system's enforcement of capability discipline.

Q4 – What operations can set tags on arbitrary values?

Demonstrate CHERI's strict provenance rules are applied for capability tags. Tags are set only when a valid capability is stored; they cannot be arbitrarily set by software. Implementers should explain how the tag bit is managed and describe any privileged instructions that might modify tags.

Recommendations

- Show that tags are non-forgeable, non-addressable bits managed by hardware
- Confirm that no general-purpose instruction can set a capability tag; tags should be set to one only when storing a valid capability with appropriate permissions
- Certify that any non-capability store clears the tag bit
- Guarantee that capability tags in registers are cleared whenever an invalid operation is attempted (such as increasing bounds or writing only the address field)
- Attest that if any exception exists (e.g. firmware initialisation), it is strictly restricted to privileged code and never accessible to user programs

Examples

	Good	Bad
Example answer	No user-level instructions can arbitrarily set tag bits. Capability tags are managed by the hardware: any store of data without the C permission clears the tag bit, while storing a valid capability with C permission sets. Tags in registers are cleared automatically if an invalid capability operation is performed. Only firmware can initialise root capabilities at boot time.	User-level stores can set the tag bit by writing to a special register.
Why is it good / bad	Gives all recommended details	States that user-level code can set or clear tags directly.

Q5 – How do you ensure integrity of tags with concurrent overlapping writes?

Show that the design preserves tag-data integrity under concurrency, even in multicore systems. The memory subsystem must maintain tag atomics and how coherence is handled when multiple agents access memory, reassuring evaluators that there is no risk of race conditions or tag corruption.

Recommendations

- Explain the mechanism for tag integrity
- Attest that data and tag bits are moved atomically between registers and memory, so it is impossible to write part of a capability and some data in the same location and end up with a valid capability
- Guarantee that the memory subsystem tracks capability tags per aligned word and clears the tag if any byte of that word is written by an instruction that lacks the capability
- State that caches and buses preserve tag-data atomics and, if multicore or DMA is supported, the coherence protocol propagates tags correctly
- Alternatively, in a single-core design, specifies that there is only one bus master (no DMA) and documents this restriction

Examples

	Good	Bad
Example answer	Our design couples tags and data tightly: tag bits and their corresponding data are moved atomically between registers and memory, preventing partial accesses. The memory subsystem maintains a hidden tag bit for every aligned capability word and clears the tag if any byte is written without C permission. We use the standard cache-coherency protocol to ensure that tag and data updates propagate across cores; there is no DMA support. Thus, concurrent overlapping writes cannot create a forged capability.	We assume overlapping writes won't cause issues because tags are bits in memory.
Why is it good / bad	Gives all recommended details	Provides no explanation for maintaining tag-data atomicity, relying on luck or unspecified behaviours.

Q6 – How do you handle temporal safety?

Explain how the system deals with use-after-free bugs (temporal safety). Identify the mechanism such as revocation, load/store barriers, or tagging and show how it can be integrated with the operating system or hardware.

Hardware alone does not enforce temporal safety; CHERI provides primitives (tag bits and monotonicity) that software can use. Implementers should therefore describe the chosen software/hardware scheme.

Recommendations

- Describe the revocation mechanism used (e.g. CHERIvoke, Cornucopia or Cornucopia Reloaded) where freed memory is quarantined until all references are cleared
- If relevant (e.g. CHERIoT or other embedded variants), mention hardware support such as MMU capability dirty bits and per page load barriers to accelerate revocation
- If the design is an accelerator or encloses tasks, state that memory is zeroed between kernels
- Show how software must use hardware features to implement temporal safety
- Provide evidence of integration (e.g. kernel revoker or memory management unit updates)
- Leave no ambiguity about how temporal safety issues are mitigated

Examples

	Good	Bad
Example answer	CHERI does not directly enforce temporal safety, so we integrate a kernel level revoker. When memory is freed, the revoker quarantines the freed pages and scans only those pages marked "capability dirty" via MMU bits. We use the Cornucopia Reloaded technique, which relies on a per page capability load barrier available in Morello to trap if a stale capability is loaded. This deterministic revocation ensures that use-after-free pointers either continue to reference the old object or trap, preventing them from aliasing new allocation.	The system occasionally suspends all processes to perform a software sweep of memory, removing any dangling references.
Why is it good / bad	Gives all recommended details	Relies on inefficient, imprecise or purely software methods that are not designed for CHERI's guarantees. Leaves evaluators uncertain that temporal safety is enforced.

Q7 – What have you done to evaluate security of the CHERI properties of the system?

Demonstrate that CHERI-specific security properties have been evaluated beyond functional correctness. It is necessary to mention relevant state-of-the-art verification methods such as formal verification, systematic fuzzing, adversarial testing and any third-party reviews. Security evaluation should target CHERI's confidentiality, integrity and temporal safety goals.

A production-grade IP verification process is essential to support CHERI's security guarantees.

Recommendations

- Demonstrate maturity and robustness of the verification process
- Use a formal verification framework (e.g. VeriCHERI) to prove confidentiality and integrity properties at the RTL, checking a set of properties to detect microarchitectural side channels and CHERI specific vulnerabilities
- Show how you conducted extensive fuzzing of load/store and capability manipulation instructions to check for corner cases and implementation bugs
- Commission red team or adversarial evaluations
- Describe any security audits of the software stack (e.g. running CheriBSD or other OSs with CHERI features) running on the product
- Provide evidence that any discovered vulnerabilities were fixed and reverified

Examples

	Good	Bad
Example answer	We used VeriCHERI to formally verify our RTL against global confidentiality and integrity properties, which uncovered and helped fix a Meltdown style timing side channel. We fuzzed all capability manipulation and load/store instructions to ensure correct tag propagation and bounds enforcement. In addition, an external red team performed an adversarial evaluation and confirmed that CHERI mitigated the majority of common memory-safety vulnerabilities.	We ran Linux on our CHERI core and it was fine, so the design is secure.
Why is it good / bad	Gives all recommended details	Mentions only functional testing. Says that running a large OS (e.g. Linux) without crashes proves security. Leaves evaluators unconvinced of robustness.

Q8 – What compilers do you support?

For CHERI to be easy to integrate into products, it is essential to provide good tooling support, making development practical

Identify which compilers and toolchains are supported for CHERI code, demonstrating readiness for software development. It is important to highlight CHERI capable compilers (e.g. LLVM/Clang, GCC) and mention any language-specific or domain-specific compilers.

Recommendations

- Show compatibility with CHERI-enabled compilers toolchains supported by the CHERI Alliance, supporting existing code already ported to CHERI
- Mention any additional compiler support (e.g. rustc with CHERI backend, domain-specific compilers for accelerators)
- Emphasise cross-compilation and pure-capability/hybrid compilation modes
- Give confidence that developers can write and compile CHERI programs

Examples

	Good	Bad
Example answer	We support the CHERI Clang/LLVM and LLD toolchain for C and C++. This toolchain implements pure-capability and hybrid modes for CHERI-MIPS, CHERI-RISC-V and Morello. For Morello we also provide a GCC port. Our SDK includes cross-compilation scripts and supports rustc for capability-aware Rust code.	We provide an assembler; users can write CHERI assembly by hand.
Why is it good / bad	Gives all recommended details	States that only hand-coded assembly is supported. Omits mention of CHERI-capable compilers or toolchains. Suggests developers must build their own toolchain.

Contact

In case of a question about this document, or about the certification process, please contact the CHERI Alliance: https://cheri-alliance.org/contact/