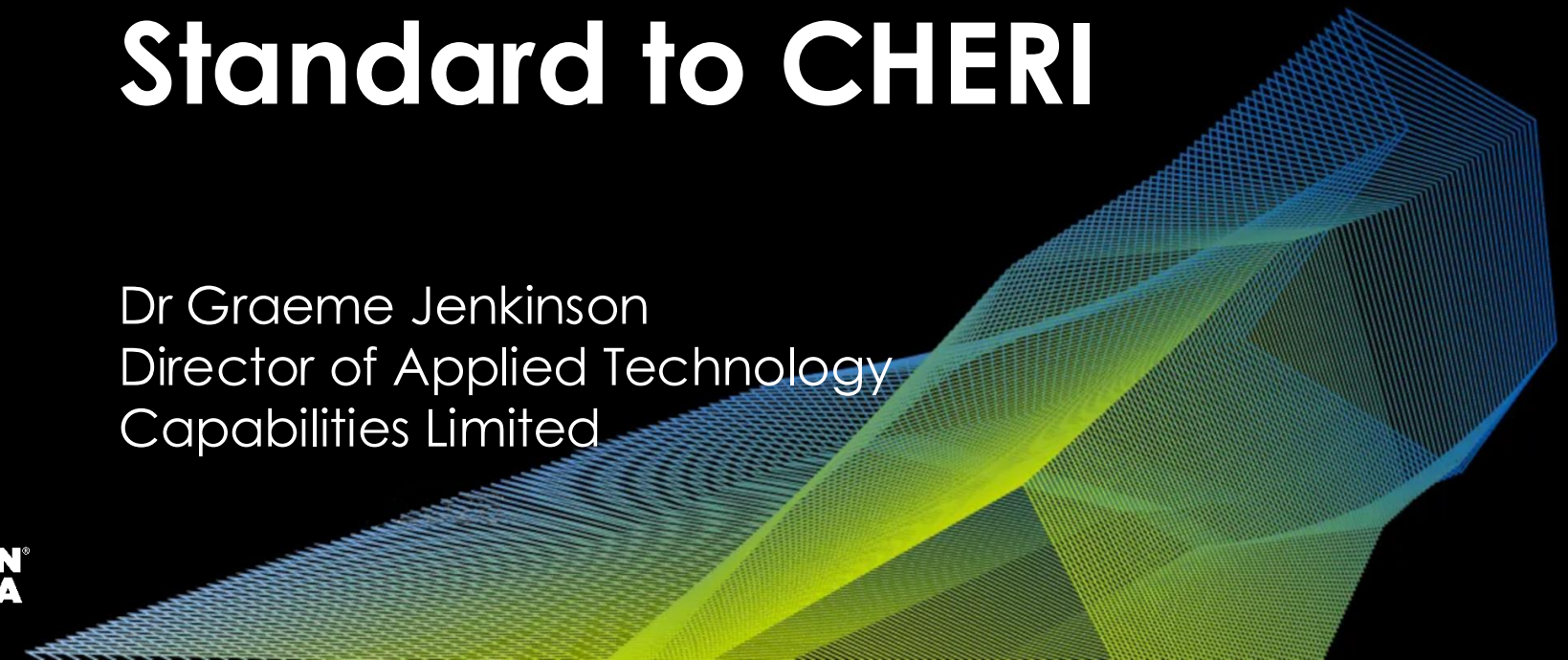
CHERITech'26
CONFERENCE

CHERI

12-13 Nov 2026
Munich, DE

Mapping the ETSI Memory-Safety Technical Standard to CHERI

Dr Graeme Jenkinson
Director of Applied Technology
Capabilities Limited



Located at semi | SEMICON EUROPA

Agenda

- Introduction
- Overview of the ETSI Memory Safety Technical Specification
- Mapping CHERI to the ETSI Technical Specification
- Broader Industry Impact and Future Work

Agenda

- **Introduction:**
  - **Motivations for ETSI Memory Safety Technical Specification**
  - **What is Memory Safety?**
- Overview of the ETSI Memory Safety Technical Specification
- Mapping CHERI to the ETSI Technical Specification
- Broader Industry Impact and Future Work

Motivations for ETSI Memory Safety TS

- A smartphone's general-purpose OS and all of its network-facing applications must be implemented with at least non-deterministic data and control-flow pointer protections within five years, and strong memory safety within 15 years.
- All datacenter TCBs responsible for isolating hosted government systems from each other, and from other customers, must be implemented with strong memory safety within 15 years.
- All networking infrastructure or cyber-physical systems where software interacts with the physical environment shipped after 2034 must be implemented with strong memory safety.

- Requires ability to categorise current memory safety maturity.
- Identify limitations in current memory safety posture.
- Roadmap plan to address those limitation.

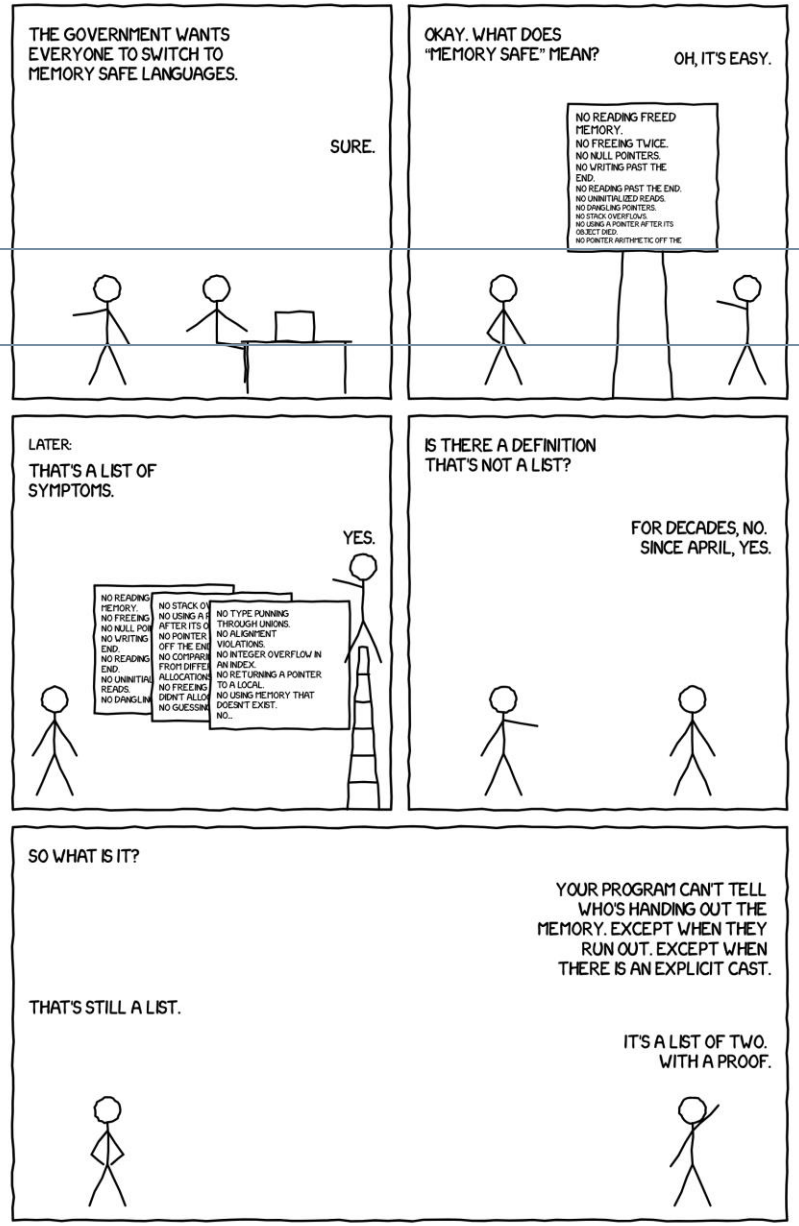
Motivations for ETSI Memory Safety TS

- Provide a definition of memory safety that allows an assessment of:
  - the completeness, the limitations, and confidence of a claim of memory safety.
- But what is exactly is memory safety?
  - And why is it so difficult to define?

What is Memory Safety?

- Define by what is absent

"A common approach when discussing memory safety is to define memory safety as the absence of vulnerabilities that arise from common weaknesses related to improper memory accesses, such as buffer overwrites and buffer over-reads." - TS-104-198



"Understanding memory safety" by Aslan Askarov is licensed under CC BY-NC 3.0.

What is Memory Safety?

Memory safety is a critical concept in computer programming that ensures programs access memory locations securely and correctly. It prevents errors like buffer overflows, use-after-free, and other vulnerabilities that malicious actors can exploit to compromise systems.

SCI Seminconductor

Memory safety, a key concept in software development, addresses how an application handles memory operations, such as reading, writing, allocation, and deallocation. A memory-safe application operates within the bounds of its allocated memory -- it doesn't access or modify memory locations that it's not allowed to access -- and frees memory after use.

Ada Core

Memory safety guarantees that software cannot interact with or modify memory in ways that might result in memory-related errors or vulnerabilities. Memory safety plays an important role in maintaining the security and reliability of software applications.

Codasip

What is Memory Safety?

- These definitions (any many, many others) are often approaching a tautological definition of memory safety.
  - Memory safety is what you get when you access memory safely, but this doesn`t define what accessing memory safely entails.
- As these definitions all refer to memory safety so does that mean these technologies provide the same properties?
  - Clearly, they don't. And the role of the TS is to develop a vocabulary that allows industry (and academia) to describe those differences.

What is Memory Safety?

- Memory safety is about preventing one part of a program interfering with memory it shouldn`t access.
  - More formally, Memory Safety is intrinsically a non-interference property.
- Non-interference is an information flow policy:
  - Rather than constraining a subject`s actions, policy specifies which subjects are allowed to do something with a visible effect to other subjects.

A. Azevedo de Amorim, C. Hriţcu, and B. C. Pierce. “The meaning of memory safety”. In: *Principles of Security and Trust: 7th International Conference, POST 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings 7*. Springer. 2018, pp. 79–105

René Rydhof Hansen, Andreas Stenbæk Larsen, and Aslan Askarov. 2026. The Downgrading Semantics of Memory Safety. Proc. ACM Program. Lang. 10, PLDI, Article 182 (June 2026), 28 pages. https://doi.org/10.1145/3808260

Motivations of ETSI Memory Safety TS

- Do any of these definitions meet the goals of the TS:
  - Requires ability to categorise current memory safety maturity.
  - Identify limitations in current memory safety posture.
  - Roadmap plan to address those limitation.
- Having ruled out the following approaches, how then to proceed?

Agenda

- Introduction
- Overview of the ETSI Memory Safety Technical Specification:
  - Memory safety properties
  - Memory safety assurance levels
- Mapping CHERI to the ETSI Technical Specification
- Broader Industry Impact and Future Work

Memory safety properties

The TS specifies a set of memory-safety properties - that is, logical assertions that a property or condition remains true during execution.

The set of memory-safety properties provided by a given technical solution (referred to as the Memory Safety Execution Environment) specifies its memory-safety semantics. These semantics can then be used, for example, to:

- Assess completeness, highlighting properties that are absent.
- Specify limitations, highlighting conditions in which certain properties do not hold.
- Specify the properties that are mandatory or optional to meet a given level of assurance.

Categories of memory safety properties

- Core memory safety properties:
  - Concurrency memory safety
  - Initialisation memory safety
  - Finalisation memory safety
  - Reference memory safety
  - Spatial memory safety
  - Temporal memory safety

- Properties that build on core-properties
  - Type-safe memory safety
  - Control flow memory safety

Terms and definitions

Challenge: How to specify technology-neutral properties?

- The goal of the TS is to specify memory-safety properties that are independent of any vendor or technology.
- This requires a set of vendor neutral terms and definitions that are used consistently to specify the properties.
- These terms and definitions must be abstract enough that they can be used across the spectrum of memory-safe technologies, but specific and detailed enough that they're meaningful and cannot be claimed vacuously.

Precise spatial memory safety is a property that ensures **memory accesses** via a **memory-object reference** are restricted to those permissible in the programming language's abstract machine.
Any attempt to perform a **memory access** that is not permitted by the programming language's **abstract machine** is an a**ttempted memory-safety violation**.

TS-104-198, 7.2.1

Memory safety assurance

- The claim memory-safety semantics for a given target should identify:
  - the target code, interfaces, and memory operations to which the claim applies;
  - the memory-safe execution environment, programming-language semantics, formal-verification evidence, or software and hardware mechanisms that provide the semantics;
  - the configuration, assumptions, and dependencies on which the claim relies;
  - the coverage and limitations of the claim, including excluded code, foreign-function interfaces, `unsafe` constructs, or other escape mechanisms;
  - the enforcement semantics; and
  - whether the semantics are provided "complete".

Memory safety assurance levels

- An assurance level provides a specific degree of confidence that a system, product, or service meets a specific set of memory-safety semantics.
- A claim of memory safety applies to a give target, encompasses a set of memory safety semantics, and applies to a given assurance level, 0-3:
  - with **0** being no evidence, **I** baseline, **2**, hardened baseline, and **3** enforcement.
- An assurance level is not a guarantee that a target is free from memory-safety violations.
  - Higher levels of assurance require stronger or more comprehensive memory-safety enforcement, stronger supporting evidence, and, where applicable, a broader scope.

Agenda

- Introduction
- Overview of the ETSI Memory Safety Technical Specification
- **Mapping CHERI to the ETSI Technical Specification**
  - **Mapping definitions/terms into CHERI**
  - **TODO properties**
- Broader Industry Impact and Future Work

Mapping terms into CHERI

- For CHERI C/C++ a **memory-allocation reference** maps directly to and architectural or in-memory capability:

```
// Memory allocation:
// A region of memory allocated by the
// memory allocator as the underlying
// storage of a memory object.
char* mem_allocation_ref = malloc(100);
```

- Memory-allocation reference
  - A programming language-level reference, returned explicitly or implicitly by memory allocators. The memory-allocation reference refers to a valid memory allocation for the duration of its memory-allocation lifetime.

Mapping terms into CHERI

- For CHERI C/C++ a **memory-object references** also maps directly to and architectural or in-memory capability:

```
char* mem_allocation_ref = malloc(100);


// Memory object:
// A memory allocation as represented in the
// memory-safety interface, which will at any
// given moment correspond to exactly one
// memory allocation that will be accessed
// when the memory object is accessed. The
// underlying memory allocation's size or
// lifetime may differ from that of the object
ExampleClass* mem_object_ref =
    new (mem_allocation_ref) ExampleClass();
```

- Memory-object-reference
  - A programming language-level reference that refers to a valid memory object for the duration of its memory-object lifetime.

Mapping reference memory safety properties into CHERI

- All memory access such as load/store operations must be performed by a valid capability (that is, where the associated tag bit is set).

- **Reference non-bypassability** is a property whereby all memory accesses are performed via a memory reference (that is a memory-allocation or -object reference).

Mapping spatial memory-safety properties into CHERI

- Any attempt to dereference an invalid capability (that is, where the associated tag bit is unset) results in a CHERI protection fault.
- The enforcement semantics for the CHERI protection fault may vary depending on the specific CHERI platform.

- **Non-empty dereference** is a property whereby:
  - Any attempt to perform a memory access via a memory-object reference that doesn't refer to a memory object is an attempted memory-safety violation.
  - Any attempt to perform a memory access via a memory-allocation reference that doesn't refer to a memory allocation is an attempted memory-safety violation.

Mapping temporal memory-safety properties into CHERI

- With compressed bounds (CHERI Concentrate) the enforced bounds aren't precise so CHERI can't provide this property.
- However, with uncompressed bounds CHERI can meet this property.
- So, it depends, but for practical purposes CHERI doesn't provide this property. Whereas a programming language such as Rust or Ada does provide this property.

- **Precise spatial memory safety** is a property that ensures memory accesses via a memory-object reference are restricted to those permissible in the programming language's abstract machine.

Mapping temporal memory-safety properties into CHERI

- With uncompressed bounds (CHERI Concentrate) CHERI, with support from the TCB (allocator), can provide this property as the bounds can be set to restrict access to an allocation with the allocation sized to match the representable bounds.
- This property is currently evolving to address cases such as Empty Base Optimisation (EBO), where two objects share the same allocation.

- **Non-aliasing spatial memory** safety is a property that ensures memory accesses performed via a memory-allocation reference only access the memory allocation associated with a memory object.

Mapping temporal memory-safety properties into CHERI

- This property is idented as an analogous property to the non-aliasing spatial memory property.
- It permits access outside to the memory object after its lifetime has completed but accesses are are bounded by the lifetime of its associated memory allocation.
- This matches how CHERI manages quarantining of freed memory.
- However, it`s an example of how difficult it can be to be both abstract and precise when specifying properties, and therefore I may need reworking.

- **Non-aliasing temporal memory safety** is a property where memory accesses via the memory-object reference are only permitted during the associated memory allocation's memor y-allocation lifetime.

Agenda

- Introduction
- Overview of the ETSI Memory Safety Technical Specification:
- Mapping CHERI to the ETSI Technical Specification
- **Broader Industry Impact and Future Work**

Broader Industrial Impact

Summary

- **The ETSI TS 104 198 technical specification** represents a major shift in how the technology industry defines, measures, and enforces software security.
- The specification provides a unified framework to compare and combine drastically different mitigation strategies. Organizations can now use the same baseline criteria to evaluate and benchmark software-level protections alongside hardware-enforced architectures.
- TS 104 198 provides a New Tool for Regulators and Compliance to mandate, audit, and gradually phase in secure-by-design requirements across critical infrastructure.

Broader Industrial Impact

Timeline to release of official Technical Standard

- Following the stable draft review, the technical committee (TC CYBER) will finalize the text for **Technical Body (TB) Approval** before publication.
- The ETSI TS 104 198 technical specification is currently targeting late 2026 to early 2027 for its first release.

Future work

Post first release of TS v1

- Addressing emerging properties:
  - Sub-object memory safety
- Assurance of composed systems:
  - Including those with different memory safety execution environments

CHERITech'26 CONFERENCE

12–13 Nov 2026
Munich, DE

Located at semi | SEMICON EUROPA



THANK YOU

Contact contact@cheri-alliance.org

Web www.cheri-alliance.org

CHERI™ and the CHERI logo are protected trademarks. Use of these marks is subject to the CHERI Alliance Trademark Policy – https://cheri-alliance.org/legal-and-policies/trademark/

This work © 2026 by CHERI Alliance is licensed under CC BY-SA 4.0 (Creative Commons Attribution-ShareAlike 4.0 International) – https://creativecommons.org/licenses/by-sa/4.0/