

# Recent updates on CHERI and the CHERI Research Centre (CRC)

Professor Robert N. M. Watson  
Director, CHERI Research Centre  
University of Cambridge

CHERITech'25  
University of Manchester  
14 November 2025



**CHERI**  
CHERI Research Centre

Ask me  
about a  
demo!

(No, there  
*definitely* isn't  
time!)

# About the CHERI Research Centre (CRC)

- Created as of April 2025 at the University of Cambridge
  - Funded by DSIT / EPSRC
  - Further financial support from DARPA, Arm, Google, and Codasip
  - In-kind contributions from Capabilities Limited, CHERI Alliance, and SCI Semi
- Roughly 40 individual contributors, primarily in West Cambridge
  - Led by Professors Robert Watson, Simon Moore, and Peter Sewell
  - ~20 staff members, ~10 affiliated PhD students
  - ~15 regular industrial and academic visitors
- Three high-level mandates:
  - Standards, open source, and infrastructure (e.g., w/RISC-V I)
  - Direct support for industrial CHERI adopters (e.g., μarch advice)
  - Engagement, public policy, evidence, demonstrations (e.g., memory-safety policy)
- If you would like to engage with the CRC, don't hesitate to reach out!



# CHERI Research Centre remit

## Track 1: Standards, open source, and infrastructure

Standardisation and best practices

Open-source software ecosystem

Formal modeling and proof

CHERI Remote Lab and continuous integration (CI)

## Track 2: Direct support for industrial CHERI adopters

Direct architectural and microarchitectural support

Open-source reference designs

Direct software support

## Track 3: Engagement, public policy, evidence, and demonstrations

Public speaking and other promotional activities

Ongoing evidence collection and evaluation

Full-stack demonstrations

Public policy engagement

# A few selected areas of CRC work

# We have a lot going on, so this is just a sample!

- Finalising and validating RISC-V International's RVY spec
- Updating CHERI reference material based on DSbD
- Memory-safety standardisation
- Awards received

More on these  
in a moment

But some other useful things to know about spanning research and transition:

- Ongoing microarchitectural research – using CHERI to improve performance (e.g., capability-aware microarchitectural optimisations), temporal safety, ...
- Rich evaluation of CHERI C/C++ experiences and results
- Adversarial research across a range of CHERI-enabled code bases
- Fine-grained compartmentalisation model and implementation improvements
- CHERI support in language runtimes such as V8
- ...

# Finalising and validating RISC-V International's RVY spec

Alongside our direct contributions to the RVY specification within RISC-V International, we have been working with our industrial partners by:

- Implementing RV64Y in our CHERI-Toooba research microarchitecture
- Driving maturity of application ISA feature variants for compartmentalisation and temporal safety – based on extensive learnings from Morello
- Adapting CheriBSD to RV64Y, including features such as c18n, temporal safety not used in other current OS adaptations
  - Full CheriBSD research software stack now running on three independently implemented RV64Y microarchitectures on FPGA, including Codasip's X730 and CapLtd's CVA6-CHERI
- Working with those partners to mature other RVY adaptations of key Cambridge-originated open-source artifacts such as CHERI LLVM, QEMU
- →← close to having a specification with finalised opcodes .. and ratification!

**Want to be clear that industrial partners such as Codasip, Google, and SCI Semi have played remarkable leadership roles in developing the production ISA!**

# Updating CHERI reference material based on DSbD

CHERI Roadmap for Linux

1. Introduction

2. Goals

2.1. Toolchain

2.2. Emulation and crossbuild platforms

2.3. Kernel support

2.4. Bootloaders and firmware

2.5. User space support

2.6. Linux distributions

2.7. Testing

3. Prior Work on Linux adaptation

4. Technical Reports

5. Selected CHERI research papers

CHERI Roadmap for Linux

2.1. Allison Randal (Capabilities Limited)

2.1. Heston Amavis (CapableCode Limited)

2.1. Brooks Davis (Capabilities Limited)

2.1. Vincenzo Frasino (Arm Limited)

2.1. Ben Lau (Google)

2.1. Carl Richardson (Google)

2.1. Alex Richardson (Google)

2.1. Carl Shaw (Codasip)

2.1. Robert M. Watson (Capabilities Limited and University of Cambridge)

This roadmap was developed by the CHERI Alliance Linux Strategy working group, outlining a path toward unified Linux support for CHERI hardware.

This is a living document; feedback and contributions are welcomed. Please see our GitHub Repository for source code and an issue tracker. There is a rendered version on the web, which is automatically updated when the git repository is committed to.

CHERI C/C++ Programming Guide

1. Introduction

1.1. Definitions

1.2. Version history

2. Background

2.1. CHERI capabilities

2.2. Architectural rules for capability use

3. C/C++ C/C++

3.1. The C/C++ run-time environment

3.2. Inferential, spatial, and temporal safety

3.3. Non-aliasing vs trapping memory safety

4. Impact on the C/C++ programming model

4.1. Capability-related faults

4.2. Pointer provenance validity

4.2.1. Recommended use of C-language types

4.2.2. Capability alignment in memory

4.3. Single-origin provenance

4.3.1. Pointer representation in memory

4.4. Bounds

4.4.1. Bounds from the compiler and linker

4.4.2. Bounds from the heap

This is the CHERI Pure-Capability C/C++ Programming Guide, a short guide to help developers work with pure-capability C/C++ understand the benefits that it brings, any code adaptations they might need to make, and how to interpret new compiler warnings and errors that arise with pure capability code.

This is a living document; feedback and contributions are welcomed. Please see our GitHub Repository for source code and an issue tracker. There is a rendered version on the web, which is automatically updated when the git repository is committed to.

The 2020 published version of the CHERI C/C++ Programmers Guide can be cited as follows:

Robert M. Watson, Alexander Richardson, Brooks Davis, John Baldwin, David Chisnall, Jessica Clarke, Nethandhi Elango, Simon W. Moore, Edward Nispala, Peter Sewell, and Peter G. Neumann. CHERI C/C++ Programming Guide, Technical Report UCAM-CL-TR-947, Computer Laboratory, June 2020.

Or in BibTeX:

- A lot was learned during DSbD
  - Multiple mature industry-developed microarchitectures
  - ~5MLoC of memory-safe C/C++ to >250MLoC!
- Now gathering, distilling, consolidating results via papers + technical reports – and collaborating!
  - New **CHERI Roadmap for Linux** built on lessons from CHERIBSD research OS as well as Arm, Codasip, etc.
  - Updated **CHERI C/C++ Programming Guide** improving advice, specifying C/OS APIs, sub-object bounds etc.
- In general, these are now living documents hosted in mdbook format on [github.io](https://github.io) facilitating easier access and community contributions

# A little more of a deep dive: Memory-safety standardisation

# Goal:

**Achieve universal adoption of strong memory safety, utilizing any suitable technologies, over a 30-year timeline.**

# The bad news about strong memory safety ...

Incentives to adopt architectural, programming-language, and formal methods approaches to strong memory safety at scale appear to be at best mixed:

- Eliminating these vulnerabilities requires **raising industrial best practice**: A significant investment with a multi-decade timeline and developer retraining
- **Vulnerability resistance has little perceived demand** in most consumer and enterprise markets, even if you could quantify those benefits
- The **opportunity cost** for improvements in engineering practice and security are high vs. pursuing instead investing in features customers actually ask for
- **Multi-decade strategies** are hard enough in government, let alone industry

The result is good feelings and (sometimes) token gestures from vendors .. but limited interest in targeting billions of lines of C/C++ Trusted Computing Base (TCB) being continuously deployed even in entirely new products.

# Hyperbolic scepticism slide

While attempting to transition our own work on CHERI, we have frequently encountered the argument that a modest (2%-5%) growth in overheads such as dynamic DRAM access footprint, at data-center scale, in return for universal strong memory safety, would be an unaffordable energy cost for the industry.

Recent news suggests that this is a ...  
... complicated claim ... but ...  
**this is now an issue of incentives, and no longer one of technology.**

## *Hungry for Energy, Amazon, Google and Microsoft Turn to Nuclear Power*

Large technology companies are investing billions of dollars in nuclear energy as an emissions-free source of electricity for artificial intelligence and other businesses

▶ Listen to this article - 7



## Microsoft chooses infamous nuclear site for AI power



FORBES > BUSINESS > ENERGY

## AI Is Pushing The World Toward An Energy Crisis

Ariel Cohen Contributor

Ariel Cohen is a D.C.-based contributor who covers energy and security

Follow

May 23, 2024, 09:00am EDT

## Electricity grids creak as AI causes



GETTY IMAGES

Data centre electricity needs are forecast to double between 2022 and 2030

Chris Baraniuk  
Technology reporter



Data centers used for AI computing will require increased amounts of energy. MICROSOFT AI BLOG

# CACM February 2025: It is time to standardize principles and practices for software memory safety

Co-authored with partners including Arm, Google, Microsoft argued memory-safety standardisation is required to enable:

- Industrial best practice
- Concise acquisition requirements
- Reliable + meaningful procurement
- Product liability legislation and insurance
- Review and audit of systems
- Test and evaluation (T&E)
- Common Criteria Certification Requirements to include lab-certifiable memory safety
- Subsidies, tax incentives, or other mechanisms
- Regulatory interventions for specific classes of products or use cases
- Safe harbor provisions in future software liability regimes

The screenshot shows a news article from the Communications ACM website. The article is titled "It Is Time to Standardize Principles and Practices for Software Memory Safety" and is categorized under "OPINION" and "Security and Privacy". The article discusses the need for standardization in memory-safety research, deployment, and policy. It features a collage of icons related to software development, security, and data analysis. Below the article, there are social sharing options (Share, Download PDF, Print, Join the Discussion, View in the ACM Digital Library), a background note, and a DOI link.

# Memory-safety standardisation

The paper elaborated an agenda that included:

- **Develop an intellectual framework** that allows [formal methods, architectural memory safety, memory-safe languages] to be consistently described, with their benefits and costs documented in common language that can be used in reasoning about potential use cases
- **Develop and document improvements to current industrial practices** able to support the development and composition, of strongly memory-safe systems in a manner acceptable to industry
- **Enable the clear enunciation of technology-neutral memory-safety requirements** facilitated by these technologies, and of improved practices for the purposes of acquisition, compliance, regulation, composition, and so on.

# But this raises a question: What is memory safety?

Memory-safety definitions:  
A CHERI perspective  
WORKING DRAFT - 7 November 2025

Robert N. M. Watson<sup>1</sup> Hesham Almatary<sup>1</sup> John Baldwin<sup>2</sup>  
David Chisnall<sup>1</sup> Brooks Davis<sup>1</sup> Alfredo Mazzinghi<sup>1,3</sup>  
Vadim Sukhomlinov<sup>1</sup> Domagoj Stofa<sup>1</sup> Konrad Witaszczyk<sup>1</sup>

<sup>1</sup> University of Cambridge <sup>2</sup> Capabilities Limited <sup>3</sup> Aranet River Consulting  
§ SCI Semiconductor <sup>§</sup> Google, Inc.

Note: Please contribute to this document only if your organisation has ETSI membership, or you are able to commit to copyright assignment to one of them.

This is a working document created by the CHERI research team, and intended to contribute to discussion about memory-safety definitions associated with the ETSI standardisation process taking place from 2025-2026. This is the first exposure for these definitions, and they will likely require significant change (and more likely a total rewrite) as part of this work.

This document is limited to definitions around memory safety, and is not intended to more broadly address the definition of type safety, a more powerful set of properties that are, in practice, often tested on memory safety.

Although groups in our experiences developing CHERI, and in transitioning CHERI into industrial use, in the evolution of this work that its memory-safety definitions have the potential to span a broad spectrum of technologies and approaches, including:

- Formal verification of memory-safety properties (e.g., seL4's C-language subset and certified properties)
- Memory- and type-safe programming languages (e.g., OCaml, Swift, and safe Rust, as well as, potentially, safe subsets of C and C++)
- Architectural (hardware-enabled) memory safety (e.g., CHERI)

In addition, these definitions should apply with ‘degraded semantics’ to mitigation technologies that partially implement these properties, including:

- Incomplete static validation tools such as various LLVM analysers, Coverity, Fortify, linting tools, and machine-learning-based techniques
- Secrets-based or incomplete software mitigation technologies such as stack protection, memory randomization, and memory safety stacks
- Secrets-based or incomplete hardware mitigation technologies such as pointer authentication, memory tagging, landing pads, and architectural shadow stacks

Necessarily, this document is presented from our own perspective and understanding – particularly, one influenced by experiences in developing and deploying CHERI and its C/C++ memory-safety models. We hope to evolve this document to better incorporate other perspectives, such as from the programming-language and formal-verification communities.

1

This is in fact part of the problem – there are various partial (and often conflicting) definitions spanning various technologies and approaches

To engage with this, we are working to define CHERI-centred abstract definitions of memory safety that we hope will be (fairly) generalisable

Requires us to much better understand a number of ideas that are not necessarily well represented in other higher-level systems; e.g.,

- A clear definition of language-level “referential memory safety” that motivates various CHERI design choices
- What “sub-allocation” means – a key activity for memory allocators, which on CHERI can themselves be memory safe
- What “sub-object” means – a concept possibly specific to C/C++
- Notions of spatial and temporal safety motivated by non-aliasing guarantees, with flexibility on precision and faulting behaviours
- A memory-safety perspective on what “compartmentalisation” is

# Memory-safety standardization in ETSI

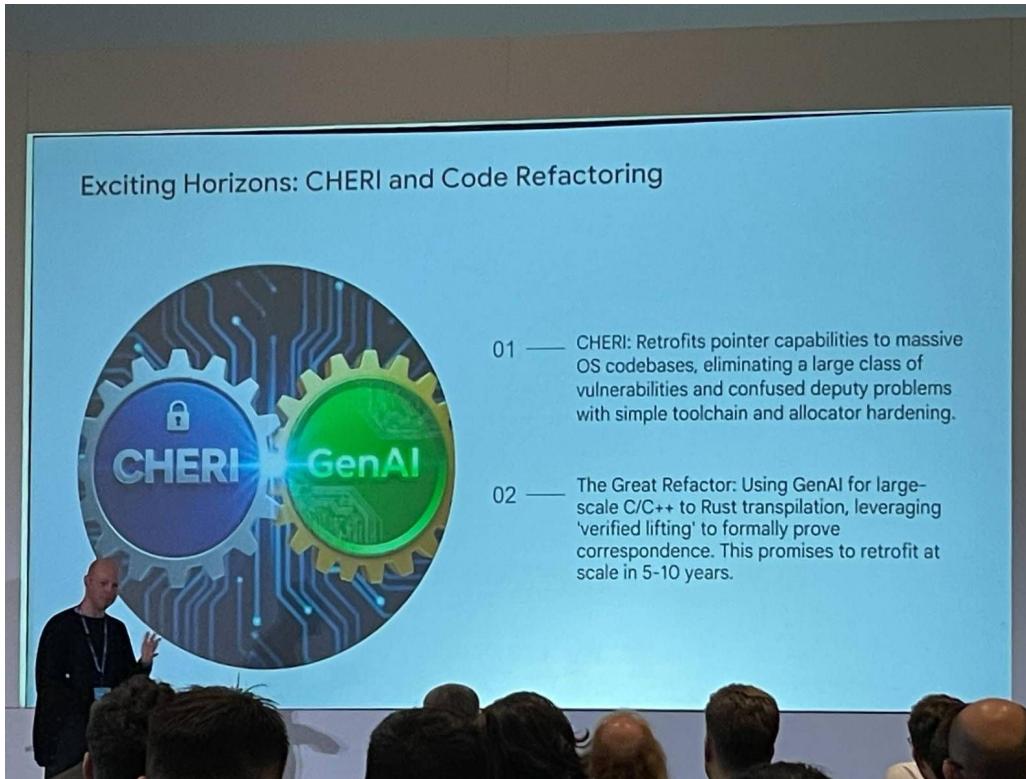
Work Programme					
Details Search   Advanced Search   My Defined Filters   Help					
Details of DTS/CYBER-00165 Work Item					
Work Item Reference	ETSI Doc Number	STF	Responsible Party	Change	Standard Not Ready For Download
DTS/CYBER-00165	TS 104 198		CYBER		
Category	Level	Version	Cover Date	Responsible	Creation Date
Category	Level	Version			
Early draft (2025-11-01)	3.0.2 Final				2025-05-27
Responsible Officer	Technical Officer		Responsible Standard		
Holder Statement	Kris Norstrom		No		
Title					
Cyber-Security (CYBER): Memory safety requirements					
Memory safety					
The scope of this work item is to develop memory safety assurance levels and specific requirements to meet them. In the region, the work will initial the framework for memory safety assurance, including memory safety analysis, memory safety verification, and systematic classification schemes for memory safety technologies. In addition, the work will provide concrete examples that illustrate the extent to which certain technologies fulfill these assurance levels and the corresponding requirements.					
Sponsoring Organization					
Cetaceo Communications, SBS a/s, NCSC, Google Ireland Limited, CIS, ZTIS, Accenture, Palo Alto Networks					
Keywords					
SAFETY, SECURITY					
Official Journal					
2025-11-06 ZWINGMANN A Draft contribution - V1.0.2 contributor for information CYBER/00165/0005 as Early draft					
2025-11-06 ZWINGMANN A A new draft is uploaded - V1.0.2 with comment: Update of document structure;					
2025-09-02 ZWINGMANN A A new draft is uploaded - V1.0.1 with comment: Update of document structure;					
2025-09-02 ZWINGMANN A A new draft is uploaded - V1.0.1 with comment: Update of document structure;					
2025-09-13 Kris Norstrom Work item adopted CYBER, see contribution CYBER/00165/0004					
2025-09-13 Kris Norstrom Work item adopted CYBER, see contribution CYBER/00165/0004					
2025-06-11 ZWINGMANN WI proposed to TB CYBER, see contribution CYBER/00165/0011					
2025-06-12 ZWINGMANN WI proposed to TB CYBER, see contribution CYBER/00165/0012					

- New ETSI working group, part of TC CYBER, to **standardize memory-safety definitions**
  - Met in-person Sophia Antipolis in September 2025 to start work
  - Further online meetings since, with active contributions from CapLtd, Qualcomm, Google building on existing work
  - Target to have significantly complete draft by end March 2026
- Do let us know if you are interested in being involved – whether within or outside ETSI
- This is (hopefully) just the first part of a larger standardisation agenda feeding into industry/sector standards, new engineering practice and processes; e.g.,
  - CRA requirements for operating systems in ESI
  - Industrial sector standards
  - Security certification standards



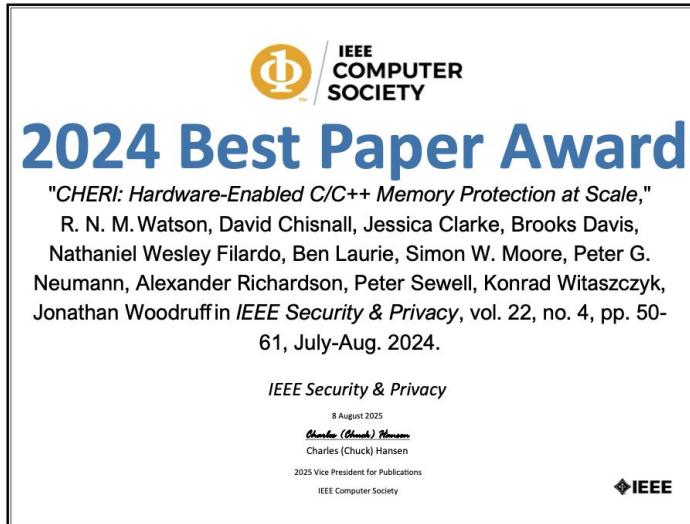
# Wrapping up

# HISC'25 yesterday: Talk from Dave Kleidermacher, VP Engineering, Security and Privacy for Android



GenAI sceptics will feel a real and understandable sense of caution reading this slide, but the key thing is mindshare.

# And picked up two IEEE security and privacy awards



- IEEE S&P **Test of Time Award** for paper on CHERI compartmentalization
- IEEE S&P **2024 Best Paper Award** for CHERI memory protection at scale

# Join us for CHERI Blossoms 2026 in Cambridge!

[WHO WE ARE ▾](#)[MEMBERSHIP ▾](#)[DISCOVER CHERI ▾](#)[NEWS ▾](#)[EVENTS ▾](#)[CONTACT US](#)

[Home](#) » [Events](#) » [CHERI Blossoms Conference 2026](#)

## CHERI Blossoms Conference 2026

**Date:** 26 – 27 March 2026

**Location:** Cambridge, UK

[Register](#)[Call for Papers](#)

And join us for CHERI's 15th birthday party!



# Wrapping up

The CHERI Research Centre aims to support CHERI adoption through a blend of:

- New research (e.g., CHERI for language runtimes, adversarial work),
- Industrial engagement (e.g., reference designs, standards), and
- Engagement (e.g., public policy, demonstrations, hosting events).

We are eager to help understand and resolve challenges to adoption – definitely technical issues e.g., in hardware and software, and technical barriers to adoption, but also non-technical ones.

We welcome collaboration, and would love to work with you to help you figure out how to make CHERI a reality.



**CHERI**  
CHERI Research Centre



But standardising  
security things has  
gone horribly wrong  
in the past!

Yes, definitely, sometimes .. but recent experience with focused specifications on strongly motivating topics is rather more positive!