Introduction to CHERI

Robert N. M. Watson, **Simon W. Moore**, Peter Sewell, Peter G. Neumann, Brooks Davis

Hesham Almatary, Ricardo de Oliveira Almeida, Jonathan Anderson, Alasdair Armstrong, Rosie Baish, Peter Blandford-Baker, John Baldwin, Hadrien Barrel, Thomas Bauereiss, Ruslan Bukin, Brian Campbell, David Chisnall, Jessica Clarke, Nirav Dave, Lawrence Esswood, Nathaniel W. Filardo, Franz Fuchs, Dapeng Gao, Ivan Gomes-Ribeiro, Khilan Gudka, Brett Gutstein, Angus Hammond, Graeme Jenkinson, Alexandre Joannou, Mark Johnston, Robert Kovacsics, Ben Laurie, Jessica Man, A. Theo Markettos, J. Edward Maste, Alfredo Mazzinghi, Alan Mujumdar, Prashanth Mundkur, Steven J. Murdoch, Edward Napierala, George Neville-Neil, Kyndylan Nienhuis, Robert Norton-Wright, Philip Paeps, Lucian Paul-Trifu, Allison Randal, Ivan Ribeiro, Alex Richardson, Michael Roe, Colin Rothwell, Peter Rugg, Hassen Saidi, Thomas Sewell, Stacey Son, Ian Stark, Domagoj Stolfa, Andrew Turner, Munraj Vadera, Konrad Witaszczyk, Jonathan Woodruff, Hongyan Xia, Vadim Zaliva, and Bjoern A. Zeeb

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contracts FA8750-10-C-0237 (“CTSRD”), with additional support from FA8750-11-C-0249 (“MRC2”), HR0011-18-C-0016 (“ECATS”), FA8650-18-C-7809 (“CIFV”), HR001122C0110 (“ETC”), and HR001122S0003 (“MTSS”). The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should notbe interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Approved for public release; distribution is unlimited.

The original CHERI R&D was supported by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237 ("CTSRD"), with additional support from FA8750-11-C-0249 ("MRC2"), HR0011-18-C-0016 ("ECATS"), FA8650-18-C-7809 ("CIFV"), HR001122C0110 ("ETC"), HR001123C0031 ("MTSS"), and FA8750-24-C-B047 ("DEC") as part of the DARPA I2O CRASH, I2O MRC, MTO SSITH, and I2O CPM research programs. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Arm's Morello and significant recent developments in the CHERI software and hardware stacks were supported in part by the Innovate UK project 105694 ("Digital Security by Design (DSbD) Technology Platform Prototype" and Innovate UK project 10027440 ("Developing and Evaluating an Open-Source Desktop for Arm Morello").

We further acknowledge EPSRC REMS (EP/K008528/1), EPSRC CHaOS (EP/V000292/1), ERC ELVER (789108), the Isaac Newton Trust, the UK Higher Education Innovation Fund (HEIF), Thales E-Security, Microsoft Research Cambridge, Arm Limited, Google, Google DeepMind, HP Enterprise, and the Gates Cambridge Trust.

Challenge: Memory Safety

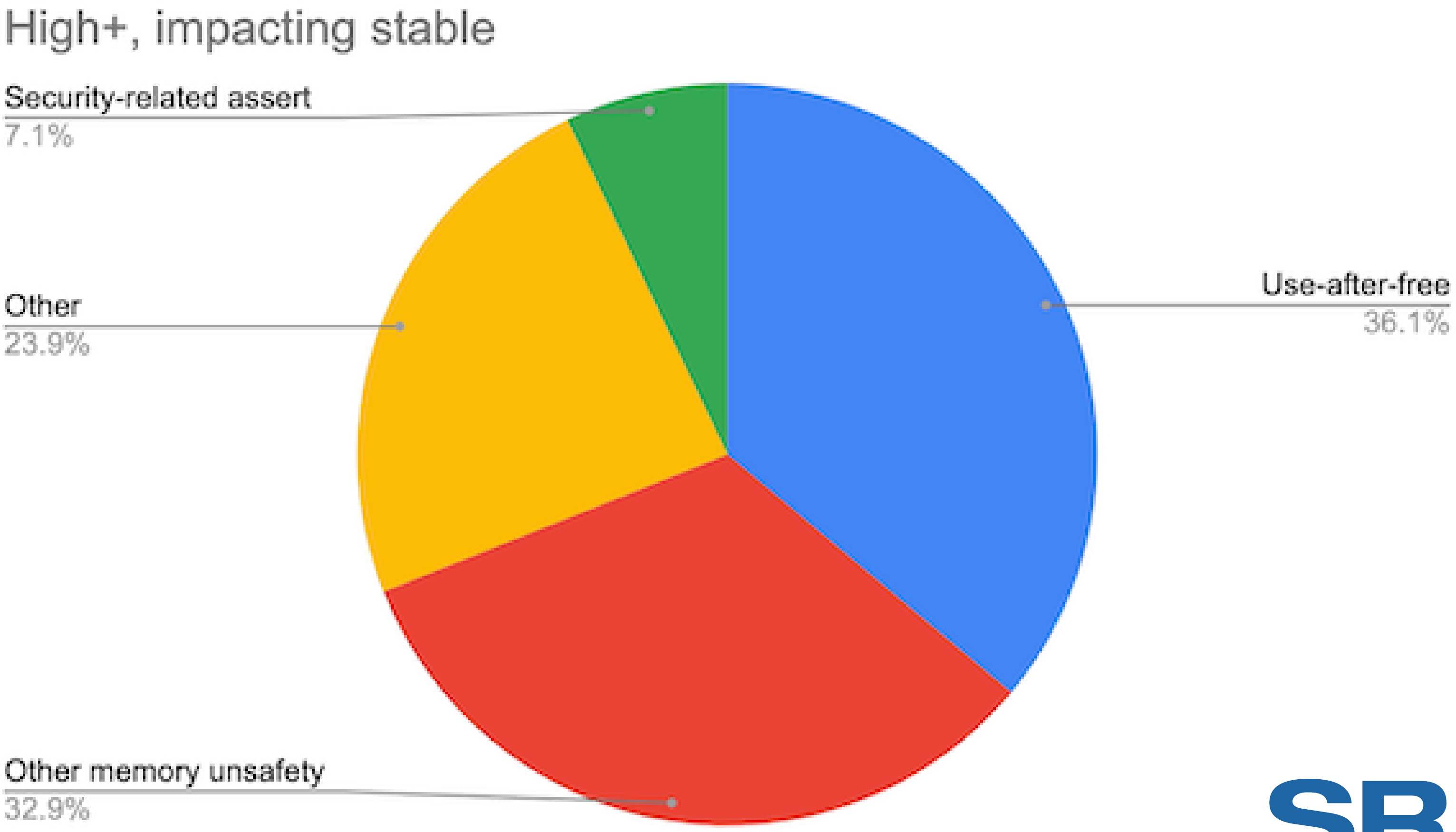
- *"Buffer overflows have not objectively gone down in the last 40 years. The impact of buffer overflows have if anything gone up."*
  Ian Levy, Technical Director at NCSC, 2022

- Matt Miller from MS Response Center @ BlueHat 2019:
  - From 2006 to 2018, year after year, 70% MSFT CVEs are memory safety bugs

Example – Chromium Browser Safety

"70% of our serious security bugs are memory safety problems"

www.chromium.org/Home/chromium-security/memory-safety



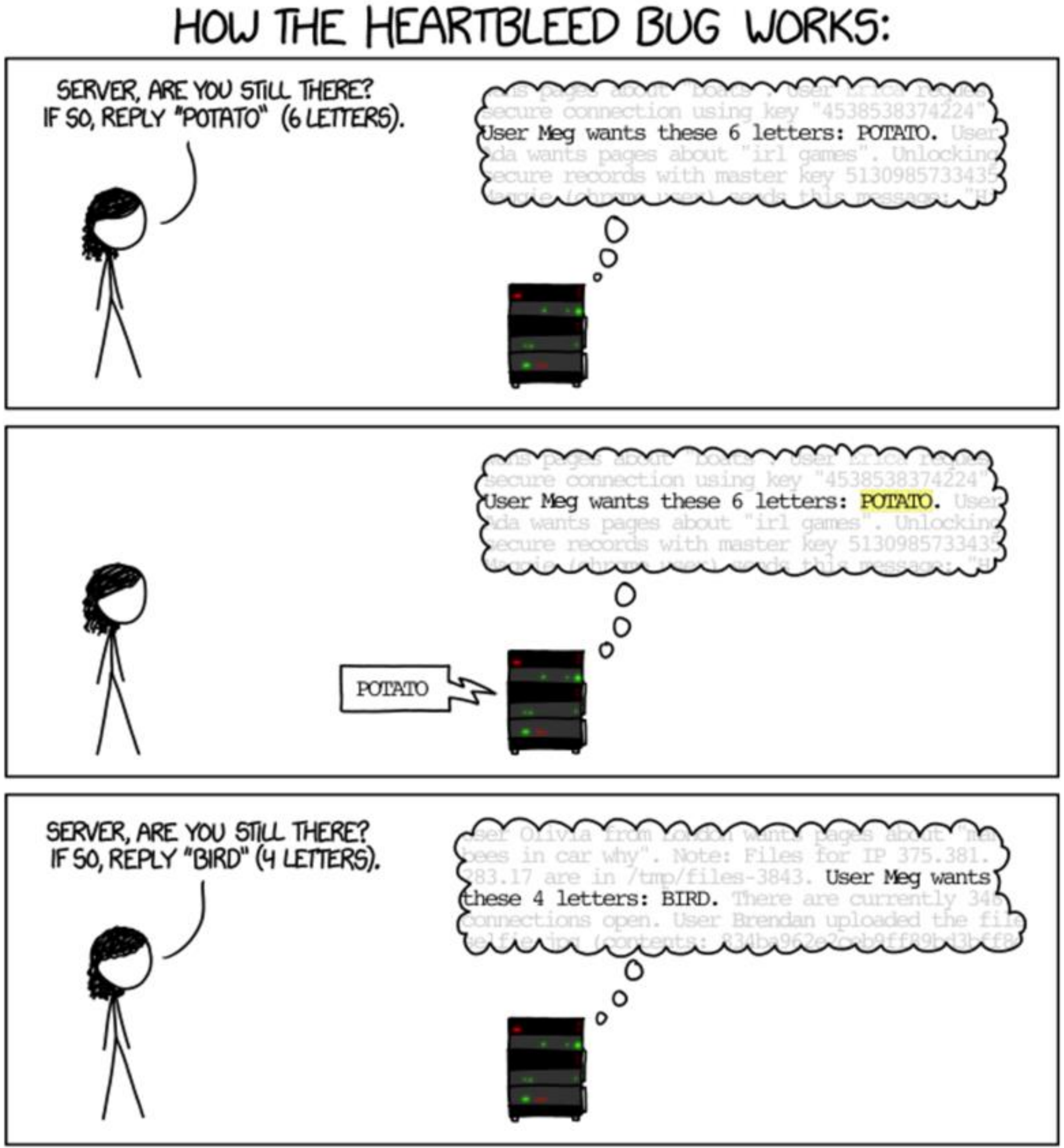
CHERI

SRI

UNIVERSITY OF CAMBRIDGE

Example

HeartBleed

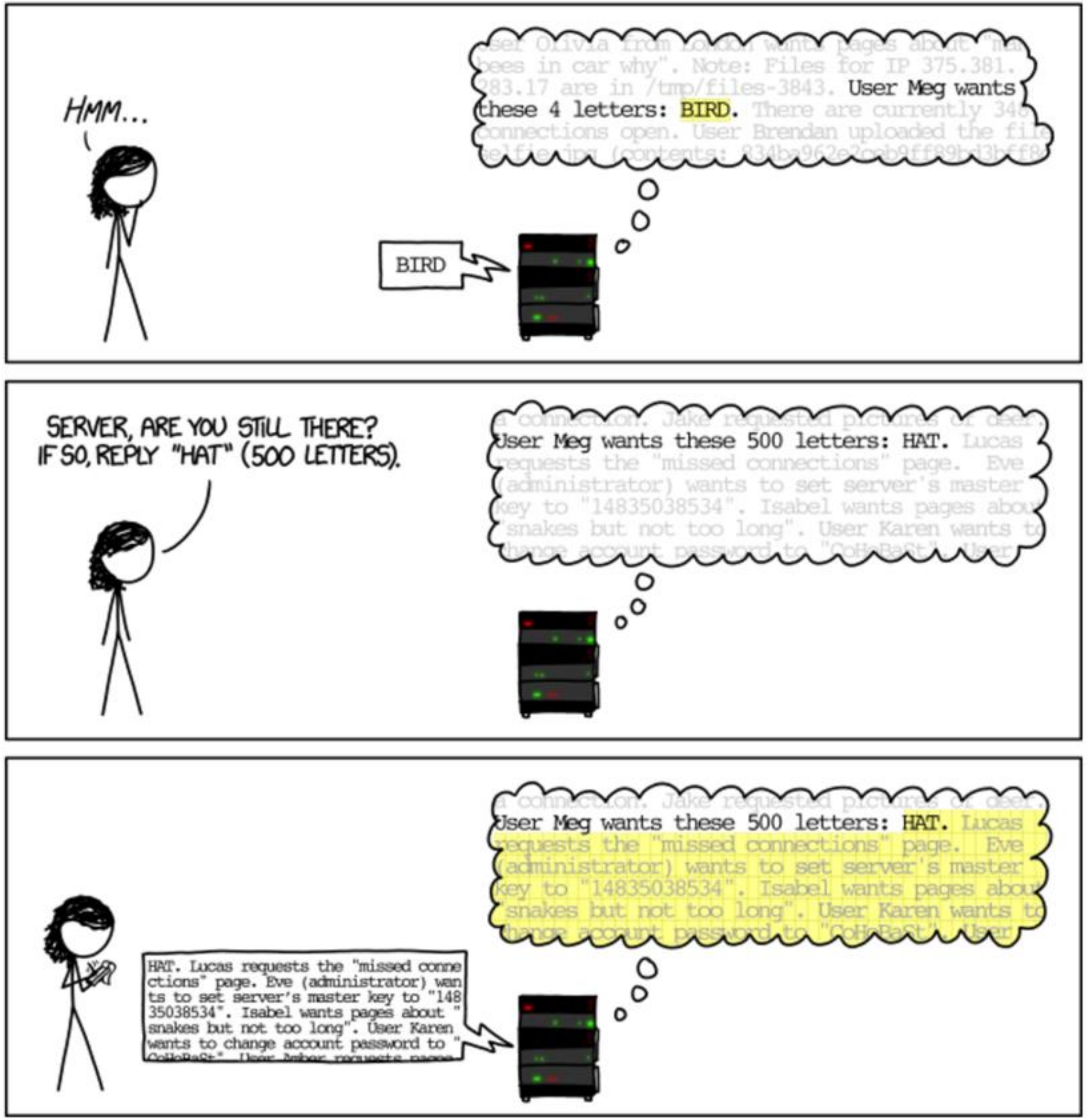


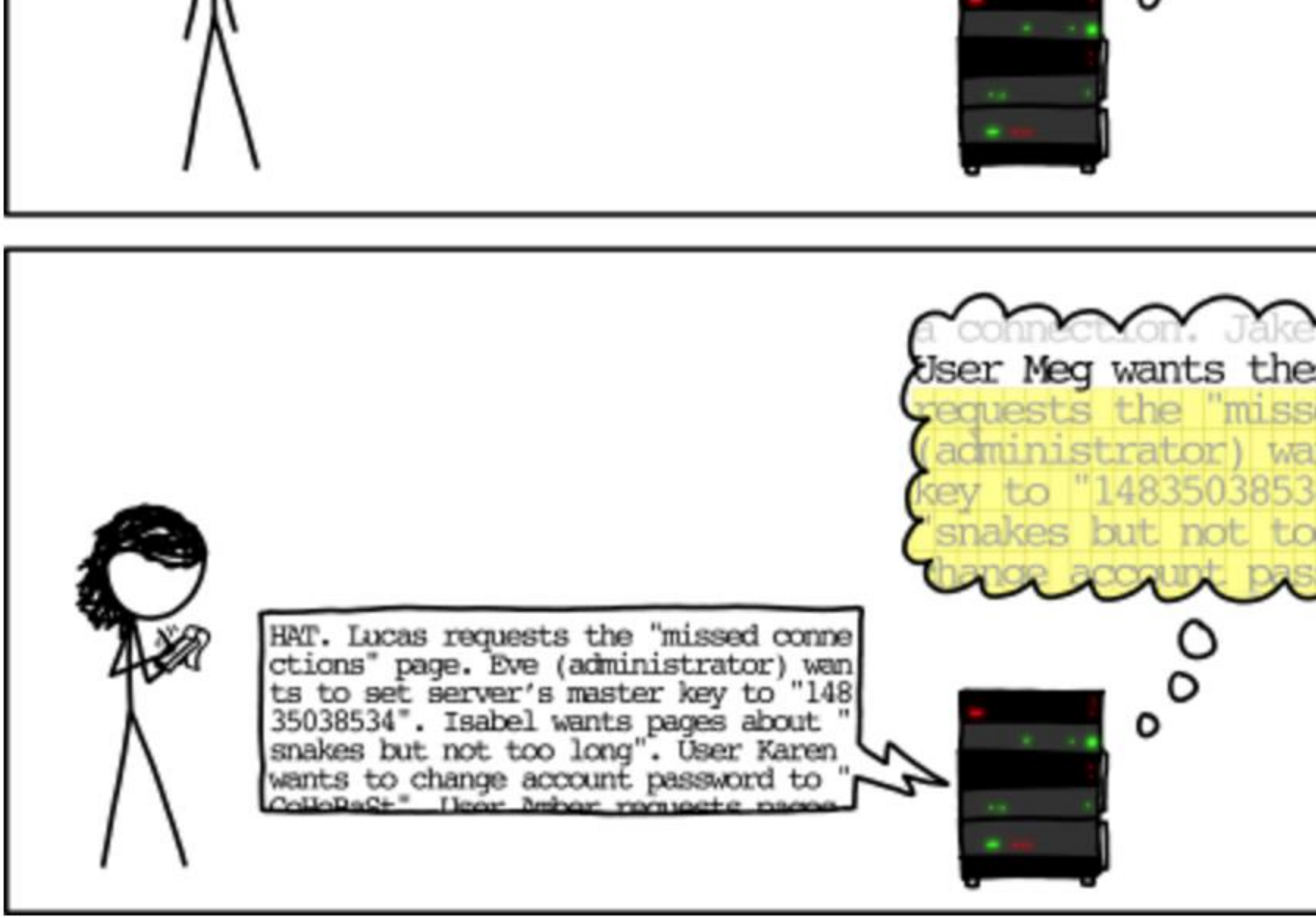
source: http://xkcd.com/1354/

CHERI

UNIVERSITY OF CAMBRIDGE

HeartBleed


HMM...
User Meg wants these 4 letters: BIRD.
BIRD
SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about snakes but not too long". User Karen wants to
User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about snakes but not too long". User Karen wants to
HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "


User Meg wants the
requests the "miss
(administrator) wa
key to "1483503853
snakes but not to
HAT. Lucas requests the "missed conne
ctions" page. Eve (administrator) wan
ts to set server's master key to "148
35038534". Isabel wants pages about "
snakes but not too long". User Karen
wants to change account password to "

CHERI

How can we do better?

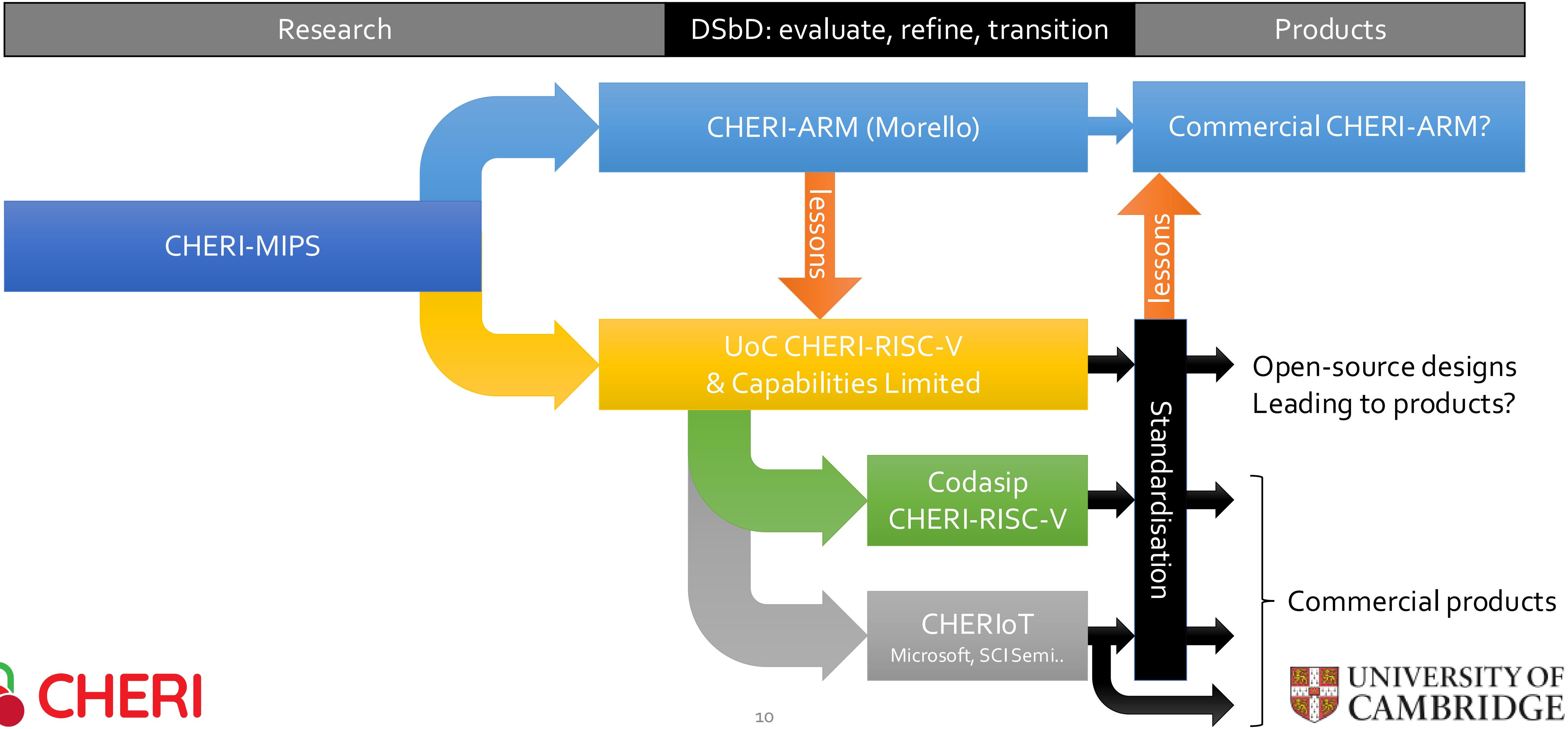
1. Use memory safe languages like Rust?
   - Great for new code
   - Impractical for the vast body of legacy code?

2. Our approach: make C/C++ and other languages memory safe using CHERI capabilities
   - Recompile your code for a CHERI enhanced processor to get memory safety
   - Can also make unsafe Rust much safer!

CHERI – A Modern Capability Architecture

CHERI = Capability Hardware Enhanced RISC Instructions

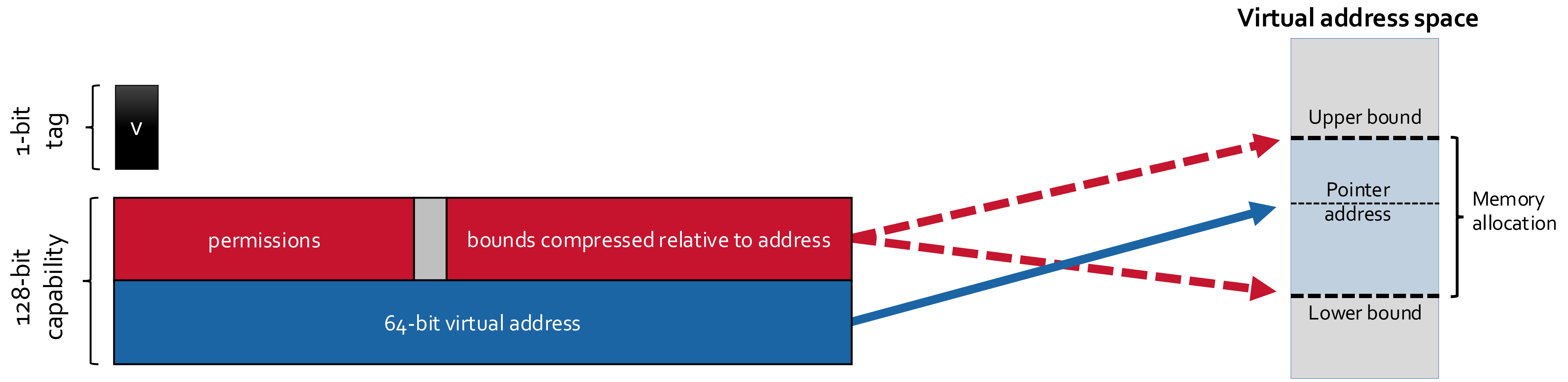
CHERI hardware timeline (abstracted!)



CHERI

UNIVERSITY OF CAMBRIDGE

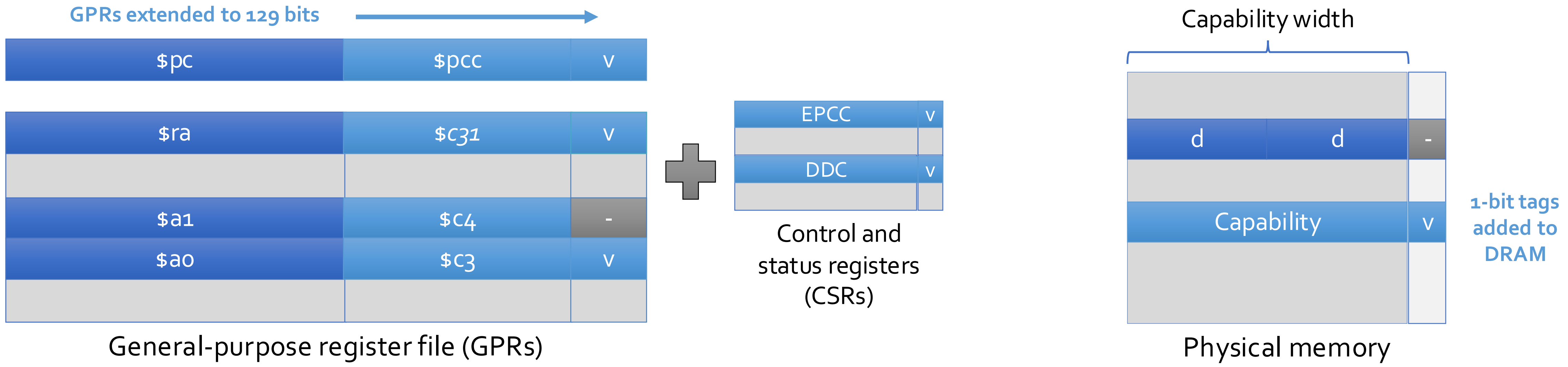
CHERI 128-bit capabilities



- **Capabilities** extend **integer memory addresses**
- **Metadata** (bounds, permissions, …) control how they may be used
- **Guarded manipulation** controls how capabilities may be manipulated; e.g., **provenance validity** and **monotonicity**
- **Tags** protect capability integrity/derivation in registers + memory

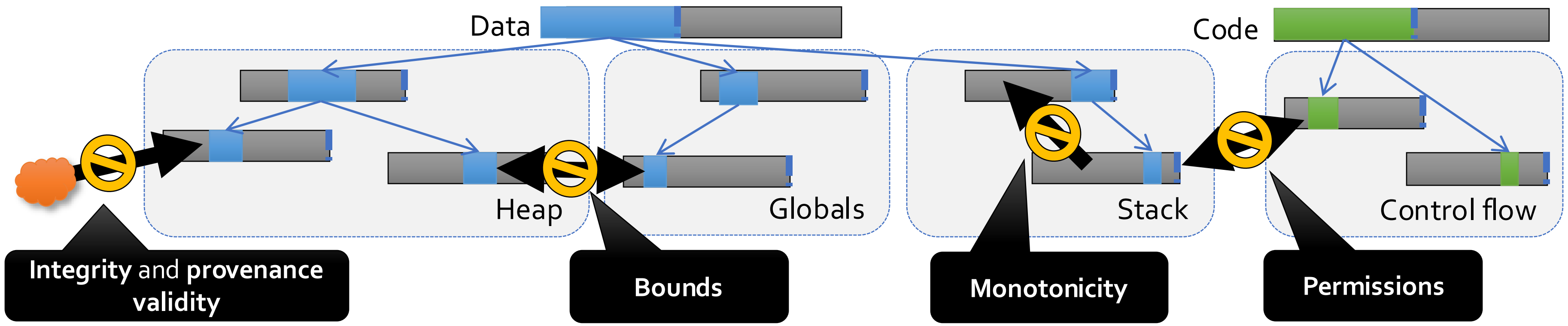
Hardware guarantees correct usage

Capability-extended register file + tagged memory



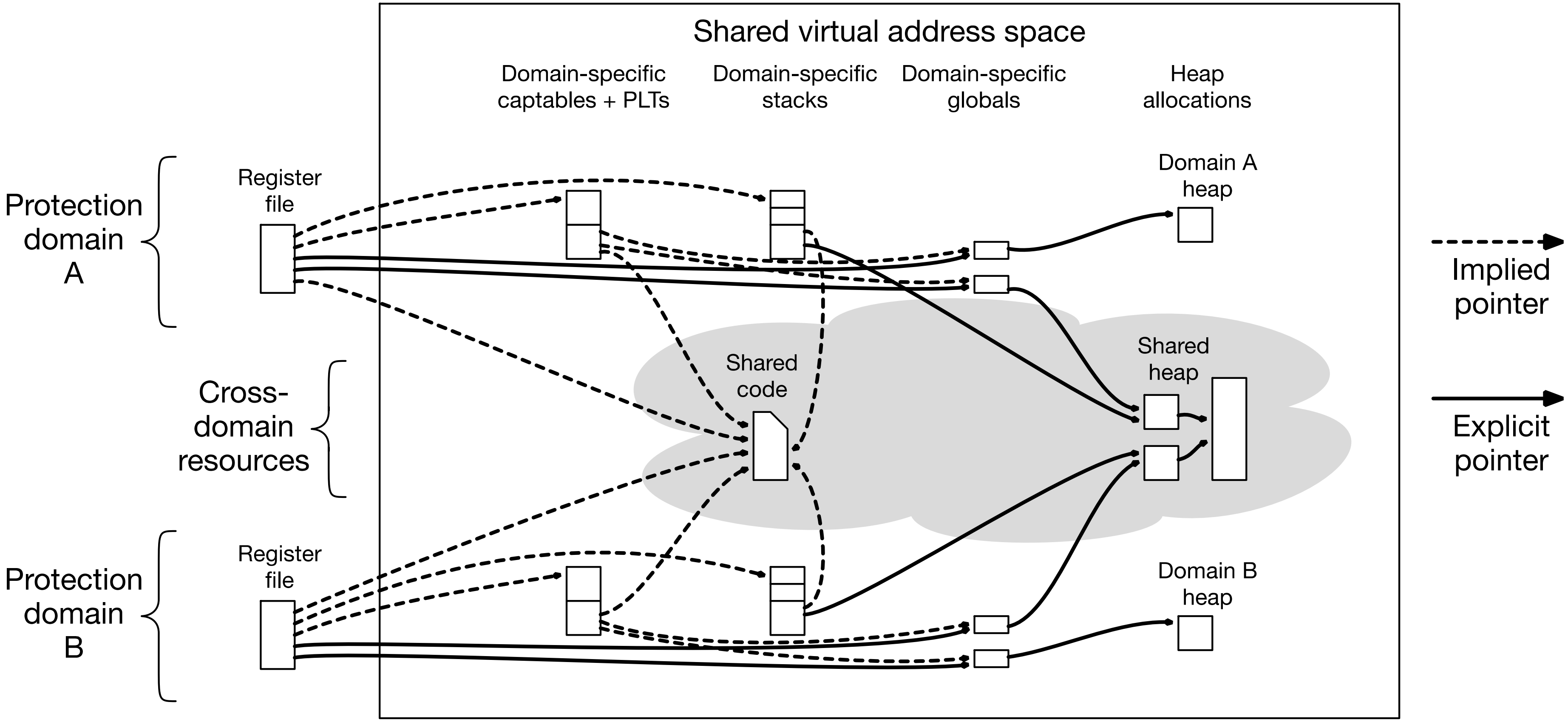
- **64-bit general-purpose registers (GPRs)** extended with **64 bits of metadata** and a **1-bit validity tag**
- **Program counter (PC)** is extended to be the **program-counter capability ($PCC)**
- **Tagged memory** protects capability-sized and -aligned words in DRAM by adding a **1-bit validity tag**
- **New instructions** are used to explicitly load, store, inspect, and manipulate capability values
- **Existing encodings** are reused for capability-relative dereferences when in a suitable mode
- **Default data capability ($DDC)** constrains legacy integer-relative ISA load and store instructions
- **System mechanisms** are extended (e.g., capability-instruction enable control register, new PTE permissions, new exception codes, exception stack pointers + vectors become capabilities, etc.)

CHERI enforces protection semantics for pointers



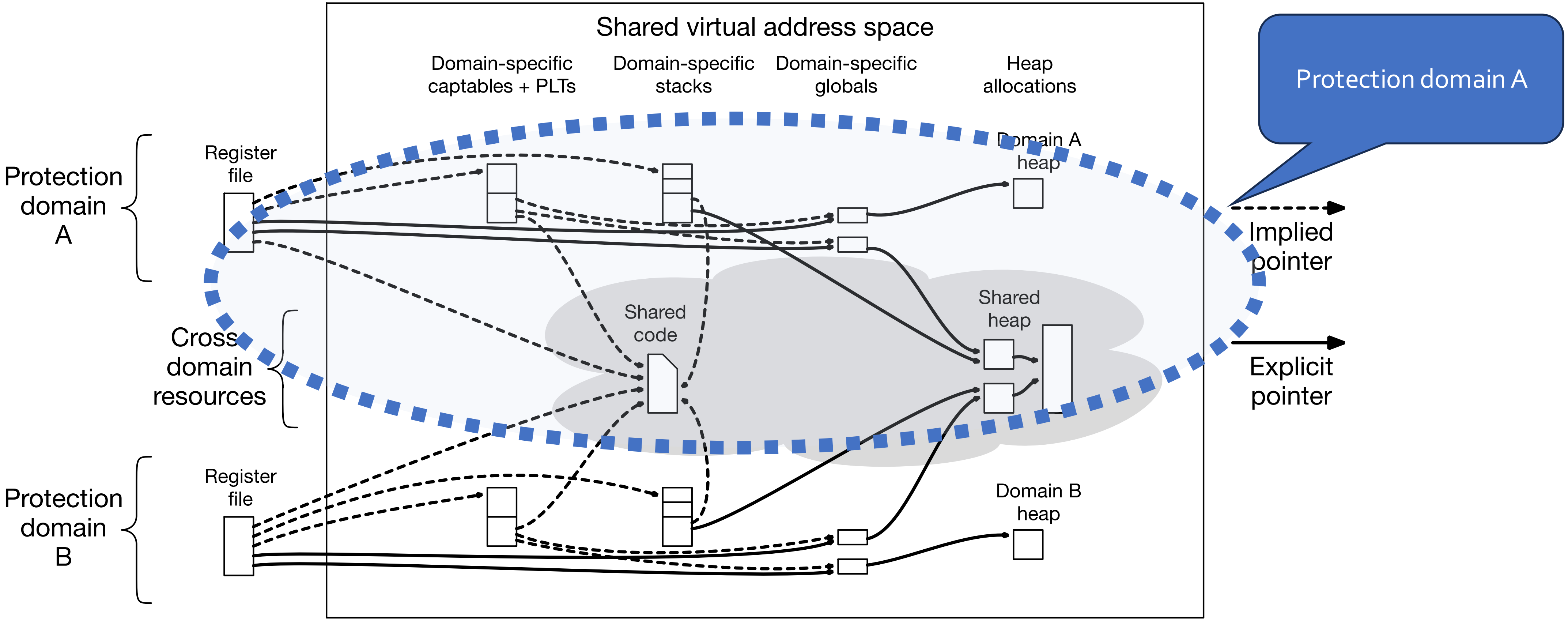
- **Integrity** and **provenance validity** ensure that valid pointers are derived from other valid pointers via valid transformations; **invalid pointers cannot be used**
- **Bounds** prevent pointers from being manipulated to access the wrong object
- **Monotonicity** prevents pointer privilege escalation – e.g., broadening bounds
- **Permissions** limit unintended use of pointers; e.g., W^X for pointers
- These primitives not only allow us to implement **strong spatial and temporal memory protection**, but also higher-level policies such as **scalable software compartmentalisation**

CHERI-based compartmentalization (1/4)



- Building on CHERI memory safety, link multiple instances of code and data within a single address space

CHERI-based compartmentalization (2/4)

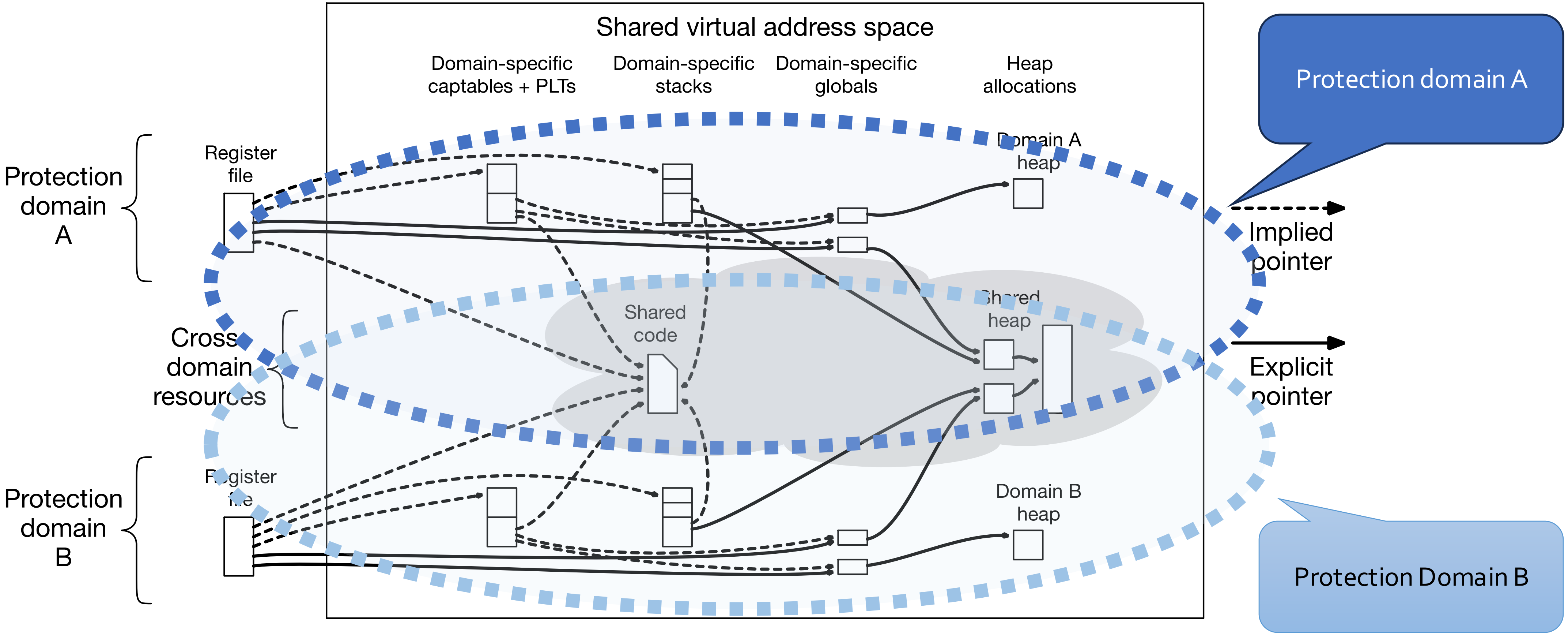


- **Spatial safety**, **provenance validity**, and **monotonicity** enable **compartment isolation**

CHERI

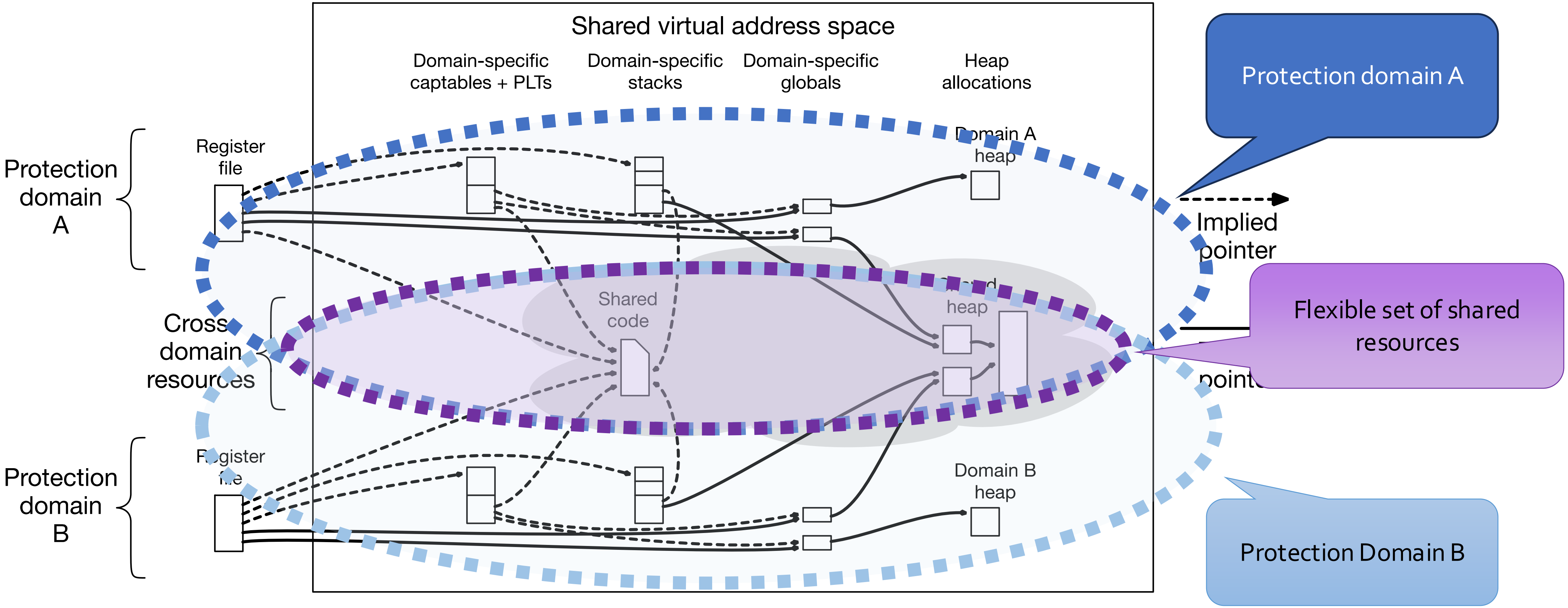
SRI UNIVERSITY OF CAMBRIDGE

CHERI-based compartmentalization (3/4)



- **Domain transition** uses exception-free, non-monotonic domain-transition mechanism based on capability-register jumps

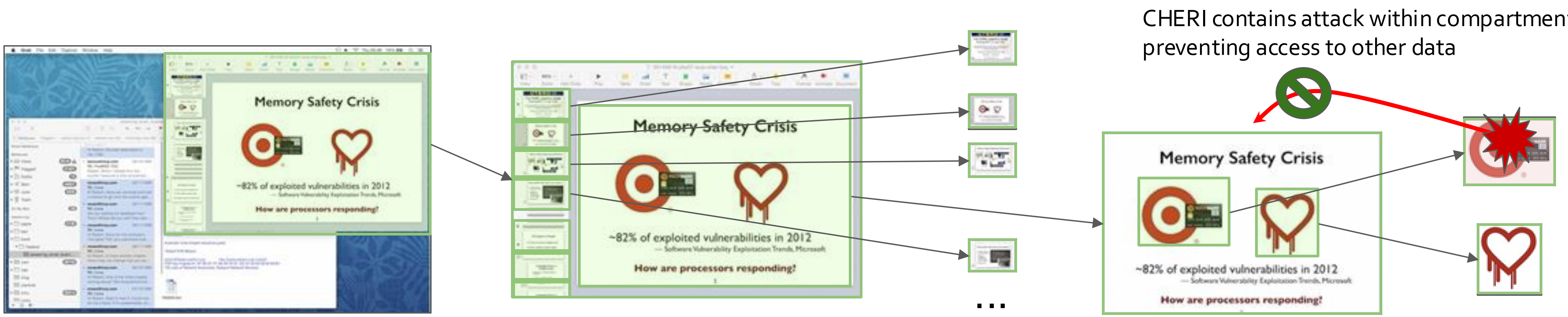
CHERI-based compartmentalization (4/4)



- **Efficient sharing** is possible using capabilities to shared memory, with rights constrained by capability bounds/permissions. Even TLB entries are shared.

CHERI SRI UNIVERSITY OF CAMBRIDGE

Software compartmentalisation at scale



- Current CPUs limit:
  - The number of compartments and rate of their creation/destruction
  - The frequency of switching between them, especially as compartment count grows
  - The nature and performance of memory sharing between compartments
- CHERI is intended to improve each of these
  – **by at least an order of magnitude**

Conclusions

Simon.Moore@cl.cam.ac.uk
Dept. Computer Science & Technology
University of Cambridge

- CHERI provides the hardware with more semantic knowledge of what the programmer intended
  - Allows **deterministic mitigation of memory safety vulnerabilities**
- Efficient **pointer integrity** and **bounds checking**
  - Eliminates buffer overflow/over-read attacks (finally!)
- Provide scalable, efficient compartmentalisation
  - Allows the **principle of least privilege** to be exploited to **mitigate known and unknown attacks**
  - **Large performance improvement over process-based compartmentalisation**
- **Working with industry and the open-source community to deploy the technology**
  - Also working with Government agencies and standardisation bodies
  - Thanks to sponsors: Innovate UK, DARPA, ARM, Codasip, Google, EPSRC, ESRC, HEIF, Isaac Newton Trust, Thales E-Security, HP Labs

https://www.cl.cam.ac.uk/research/security/ctsrd/

How do we sell security rather than performance?

CHERI

SRI UNIVERSITY OF CAMBRIDGE