CTSRD

CRASH-worthy Trustworthy Systems Research and Development

CHERI

CheriOS Microkernel

Hadrien Barral

Robert N. M. Watson, Simon W. Moore, Peter G. Neumann, Jonathan Woodruff, Jonathan Anderson, Ruslan Bukin, David Chisnall, Nirav Dave, Brooks Davis, Lawrence Esswood, Khilan Gudka, Alexandre Joannou, Chris Kitching, Ben Laurie, A. Theo Markettos, Alan Mujumdar, Steven J. Murdoch, Robert Norton, Philip Paeps, Alex Richardson, Michael Roe, Colin Rothwell, Hassen Saidi, Stacey Son, Munraj Vadera, Hongyan Xia, and Bjoern Zeeb

University of Cambridge, SRI International

CHERI Microkernel Workshop – 23 April 2016

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') and FA8750-11-C-0249 ('MRC2'). The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

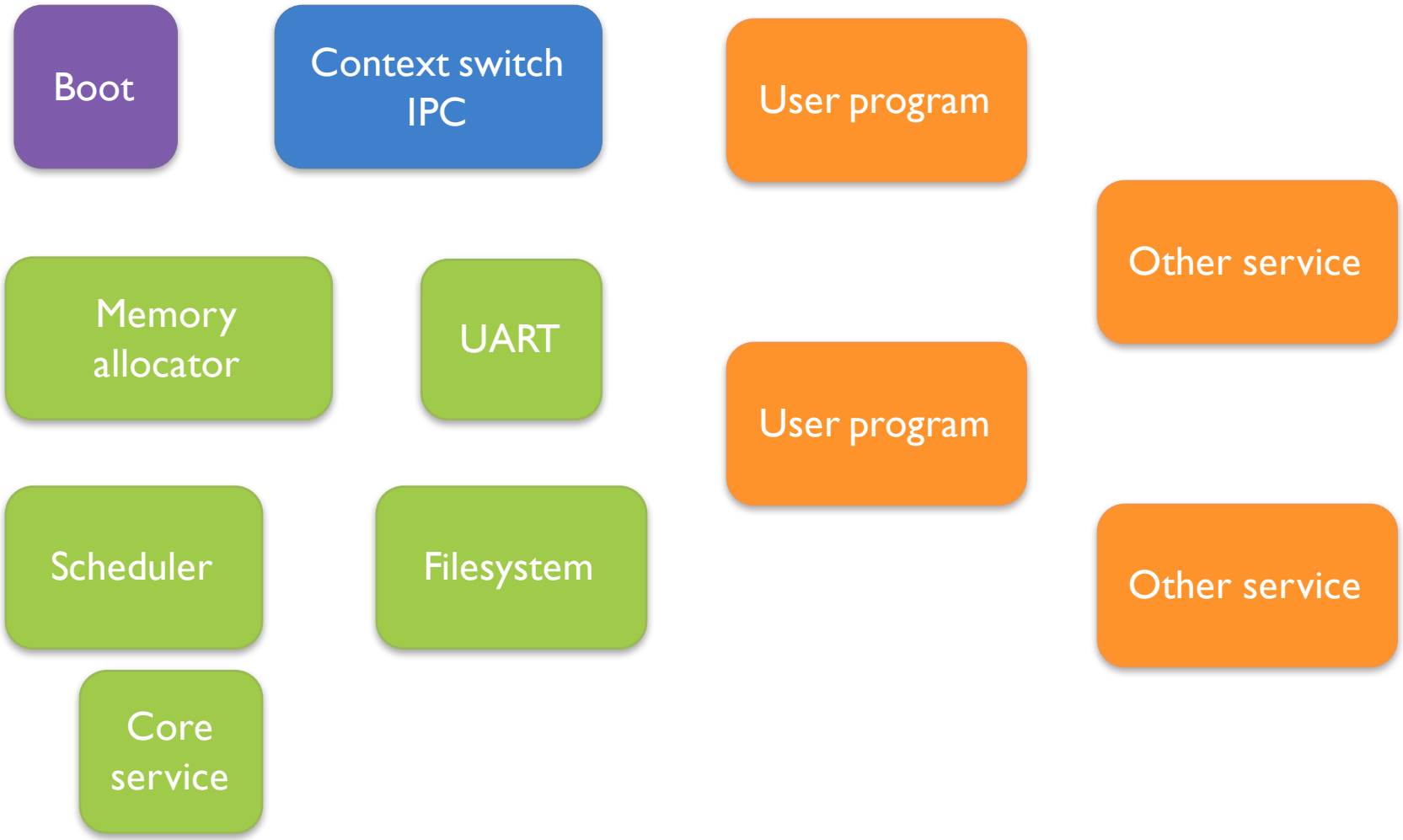
SRI International

University of Cambridge

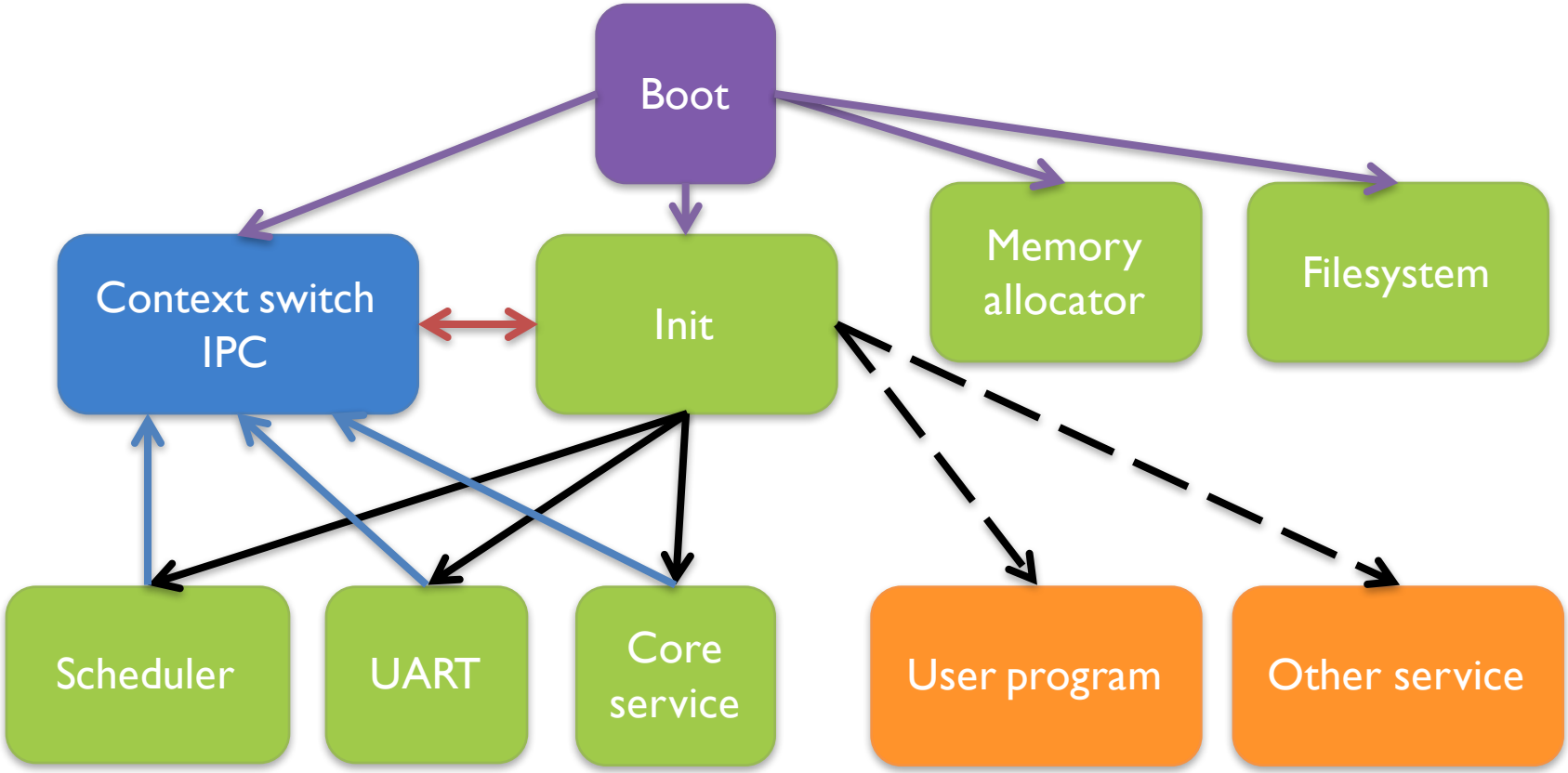
Overview

- Microkernel based on the **CHERI model**
- **Pure-capability** ABI
  - no legacy pointers, no MMU
- Make full use of **compartmentalization**
- Kernel mode**: Context switch**
  - IPC
  - Exceptions

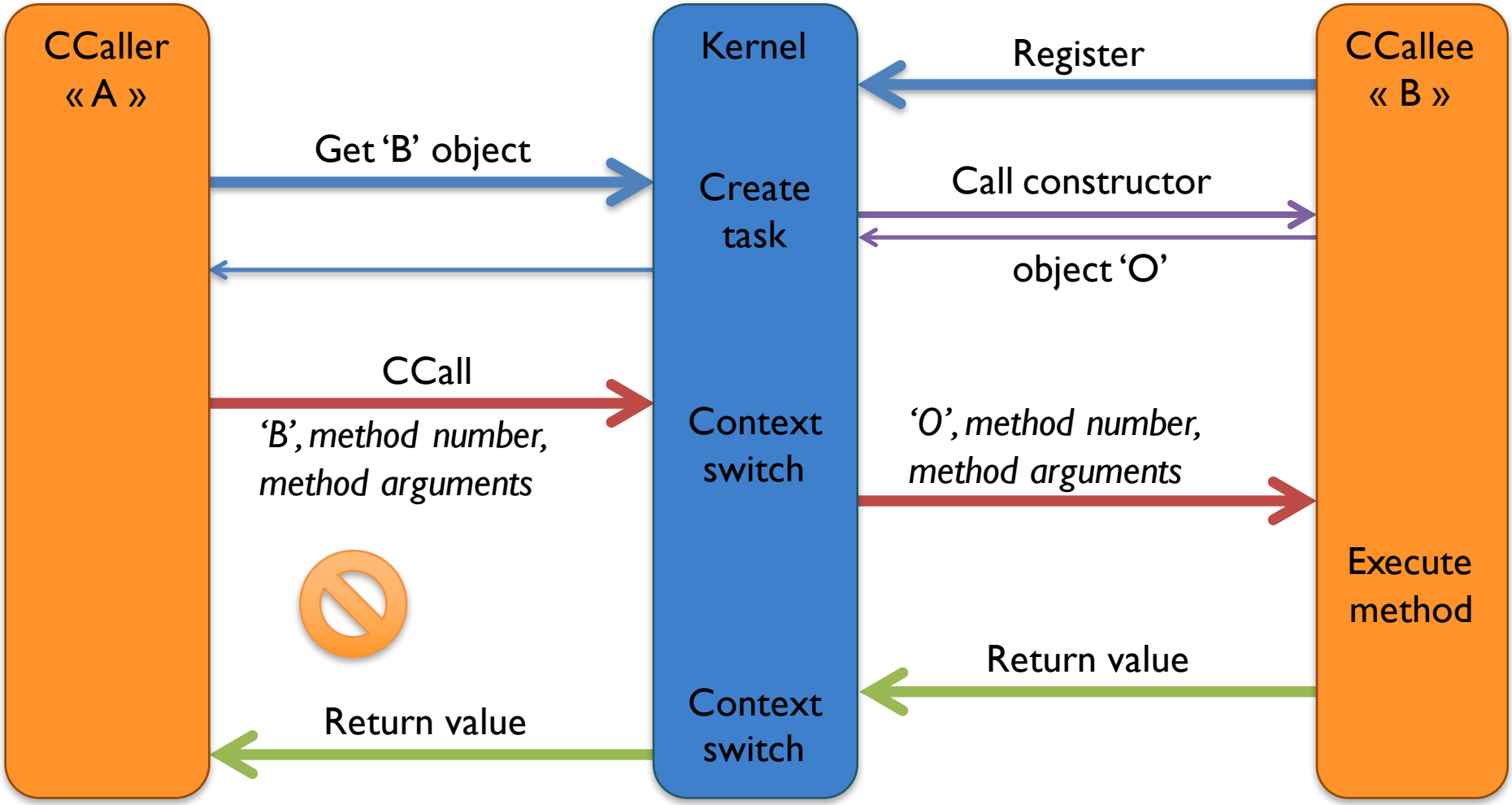
Compartments



Boot process



Inter Process Communication

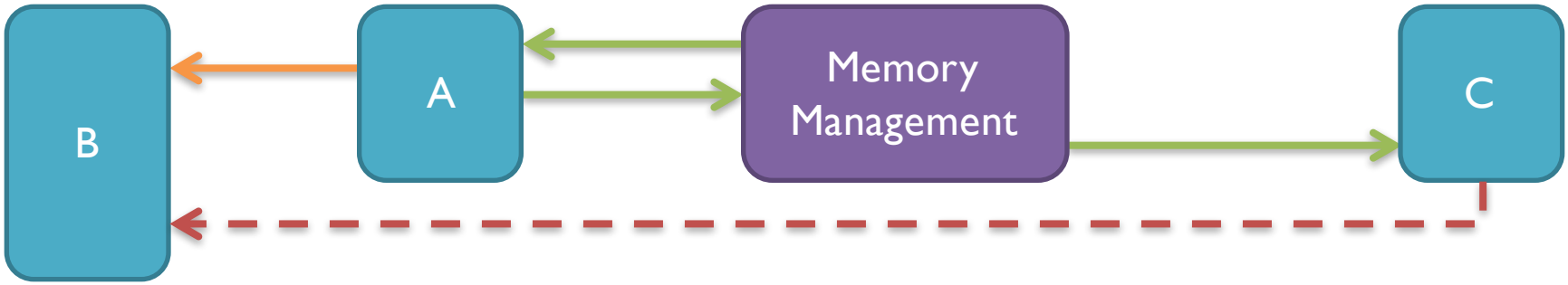


IPC (2)

- Callee might **not return** (bug, untrusted, …)
  ➔Make the call asynchronous
  - **Return to caller**; caller polls the kernel to know if the callee is done
  - Caller gives a **time limit** to the kernel
    - Other kind of limit?
- Communication with the kernel
  - **CCall** or **syscall**?

Memory management

- Revoking a capability is hard



- How can we free memory?
  - Restrict sharing of pointers
  - Garbage-collect
  - MMU

Thank you!

https://github.com/CTSRD-CHERI/cherios