

seL4 on Arm Morello

seL4 Summit 2023
Minneapolis, USA

Dr. Divya Atkins
Managing Director

Dr. Martin Atkins
Technical Director

Mission Critical Applications Limited, Bath United Kingdom




Poster: What we are doing

CHERI: Capability
Hardware Enhanced
RISC Instructions


CHERI protects each
process from misusing
its own memory

Morello is Arm's
implementation of
CHERI for Aarch64


www.mca-ltd.com

seL4 on Morello

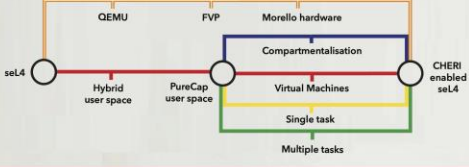
For safety and security critical systems


Arm Morello	seL4 Kernel
<ul style="list-style-type: none">• Arm Neoverse N1• CHERI capabilities• Fine-grained memory protection 	<ul style="list-style-type: none">• Formally proven μKernel• Real-time guarantees• Very fast Inter-Process Communication• Host for Virtual Machines

seL4 on Morello

- CHERI capabilities in user space
- Hybrid and PureCap user tasks
- CHERI enabled virtual machines
- High assurance Separation Kernel

seL4 system and user tasks protected by CHERI



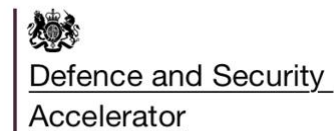


seL4 provides a trusted
computing base for
highly secure systems

seL4 prevents processes
from interfering with each
other's memory and time

History: seL4 on CHERI

- Effort in the USA started earlier
 - DARPA, Trusted Science & Technology
 - *See later slides for their update*
- Arm Research, Austin
 - Summer '22: Sid Agrawal (intern student) et. al.
 - Basic groundwork for CHERI support in seL4 user-space
- Our Work
 - Began Autumn '22, starting from Arm's Research's code
 - UK Govt Initiative: Digital Security by Design (DSbD)
 - Technology Access Programme with Digital Catapult
 - UK Defence And Security Accelerator (DASA)
 - CHERI within Defence and Security Programme
 - Run by DSTL, part of UK MoD



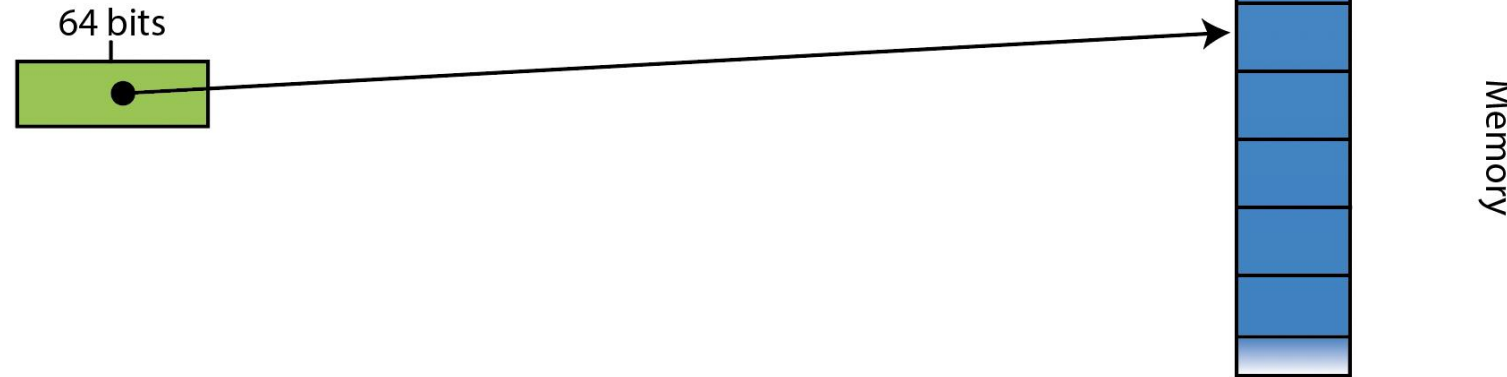
Morello, CHERI Background

- Arm Morello
 - Industrial prototype implementing CHERI
 - Workstation-class Technology demonstrator SoC
 - Developed under UK Govt Initiative (DSbD)
- CHERI replaces pointers by "capabilities"
 - Adds Permissions, Bounds to virtual addresses
 - Disallows illegal uses of pointers
 - Requires code to be re-compiled
 - Provides hardware enforced compartmentalization
- CHERI has two modes of execution and compilation
 1. **Purecap:** All address de-references apply CHERI capability rules
 2. **Hybrid:** Additional instructions for using/manipulating CHERI capabilities



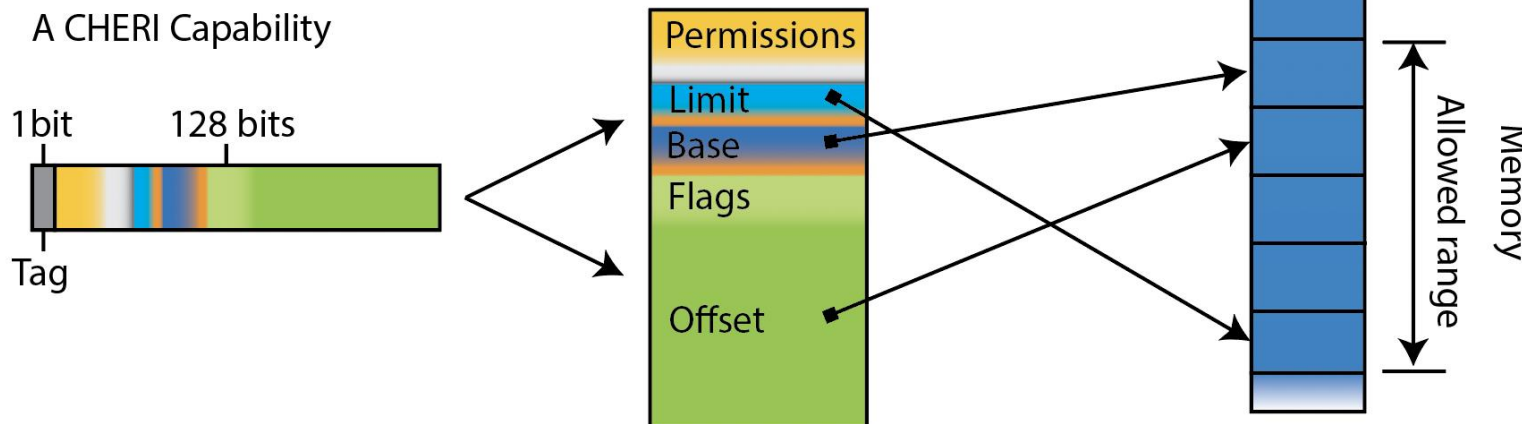
CHERI Capabilities

A "Normal" Pointer



In **Hybrid** mode, **some** pointers are capabilities and others remain unchanged

A CHERI Capability



In **Purecap** mode, **all** pointers are capabilities

Capabilities have a Hardware tag bit on every 128-bit long-word

seL4 micro-kernel

- Emphasis on security
 - Resources accessed through software capabilities.
 - Isolation between user-mode processes
- Small (~10,000 lines of C)
- High-performance IPC
- Kernel is Mathematically proven
 - ~1 million lines of proof
- Can host virtual machines
- Open Source
- *System services are outside the kernel*

- *System Services and Applications could benefit from CHERI...*



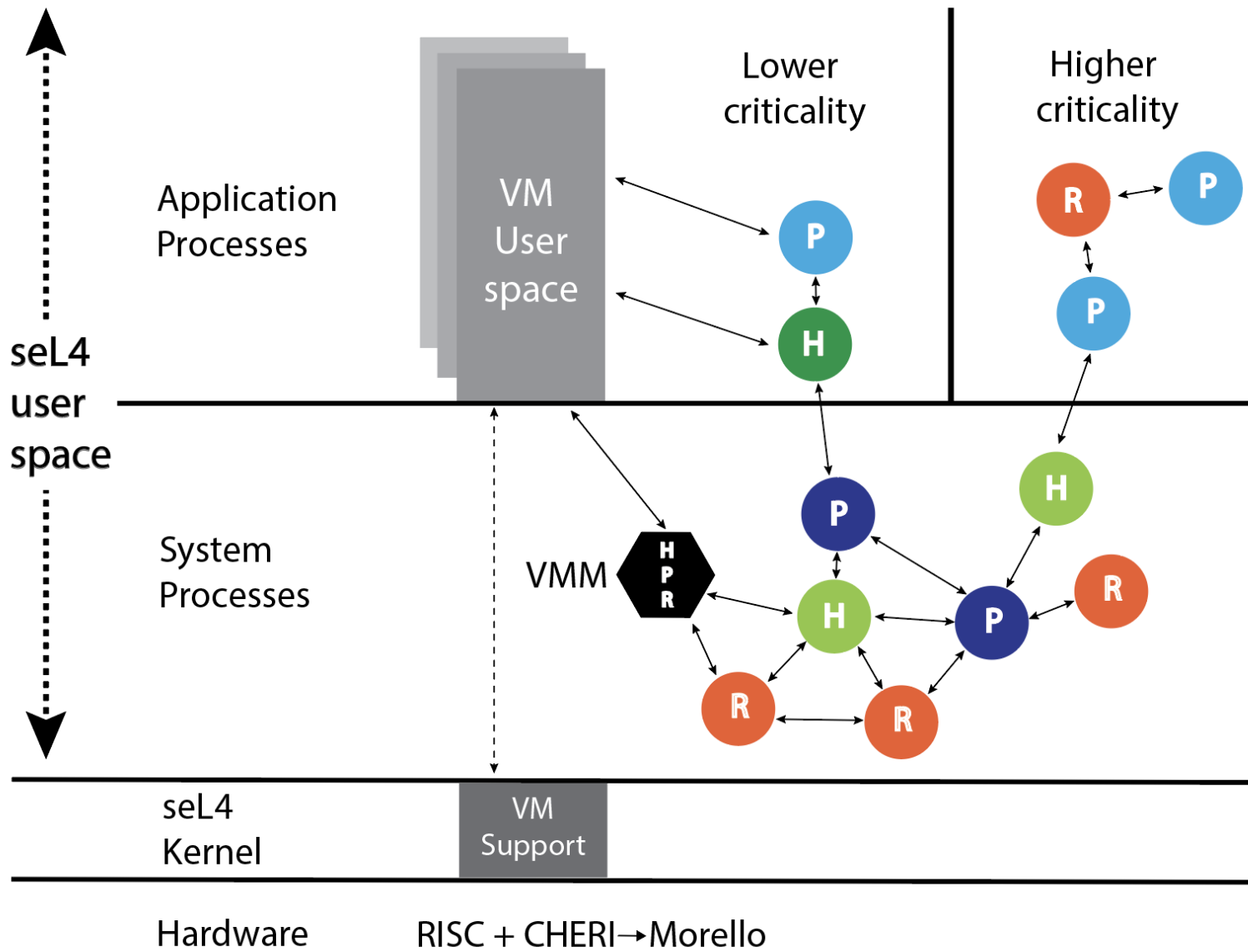
Why CHERI on seL4?

- seL4 provides guaranteed isolation between user tasks
 - Using proof, memory protection
 - with strictly-controlled communication
- CHERI saves a user process from itself
 - A fault or vulnerability can't be used to compromise other parts of the program
- What does seL4 bring to CHERI?
 - A proven Trusted Computing Base (TCB)
- What does CHERI bring to seL4?
 - Safer / more secure user-space processes









Application Architecture Rationale

- **seL4 kernel:** proven
 - Not much to gain from CHERI
- **System Services:** usually not proven, can benefit from CHERI
 - Some can be compiled as Purecap
 - Low-level tasks can be harder to compile as Purecap
 - Use Hybrid (or Rust) for them
- **Applications:** usually not proven, can benefit from CHERI
 - Where feasible, use Purecap (or Rust) (especially for higher criticality)
 - Otherwise accept Hybrid (for lower criticality)
- *Rust provides similar guarantees to CHERI*
 - *Rust: at compile time, CHERI: at runtime*
 - *Not applicable for legacy, or other reasons*

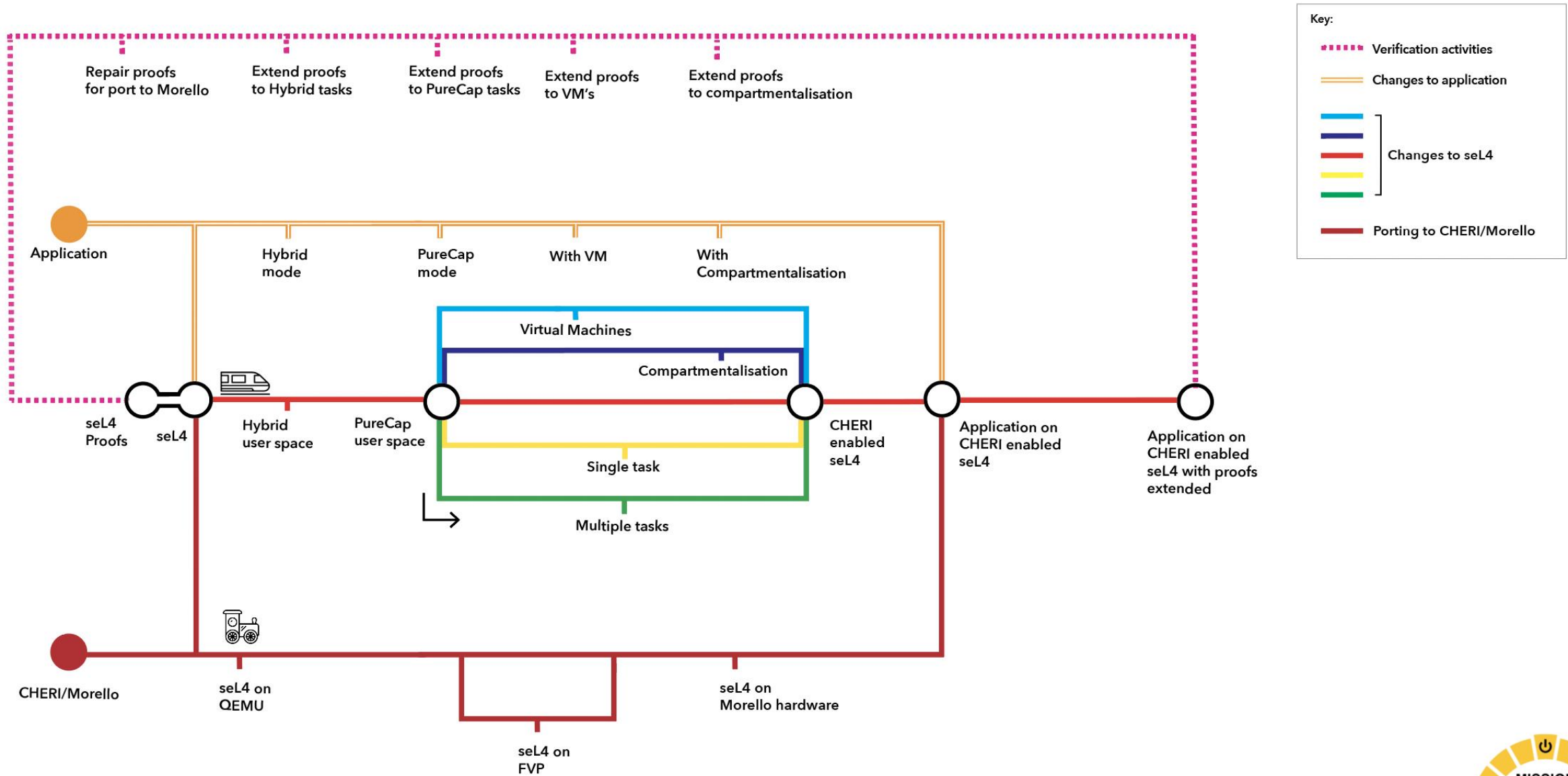
Application Architecture



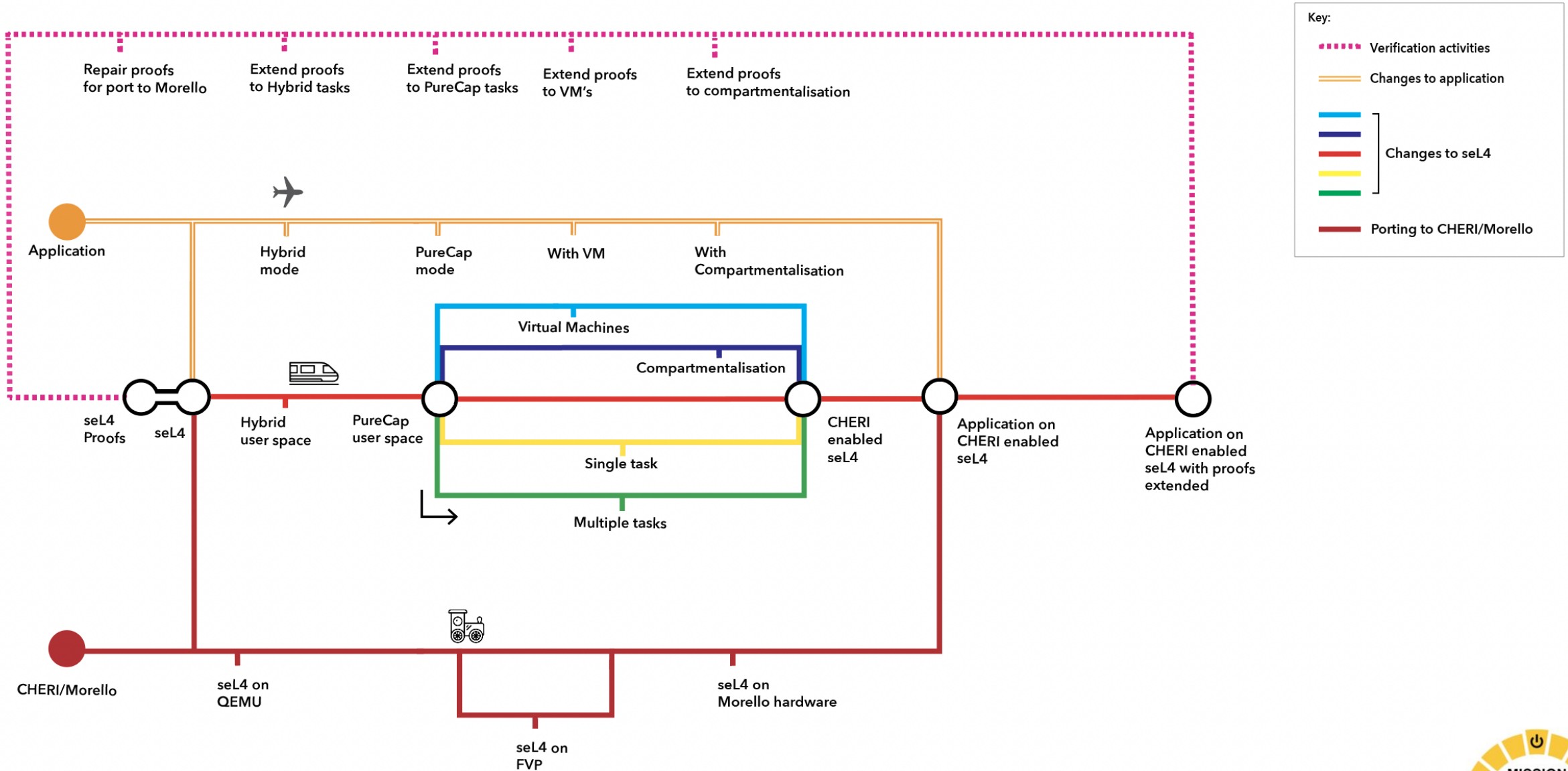
Key:

-  Purecap system process
-  Purecap application process
-  Hybrid system process
-  Hybrid application process
-  Rust
-  VM Virtual machine
-  VMM Virtual machine monitor
-  Hybrid/ Purecap/ Rust

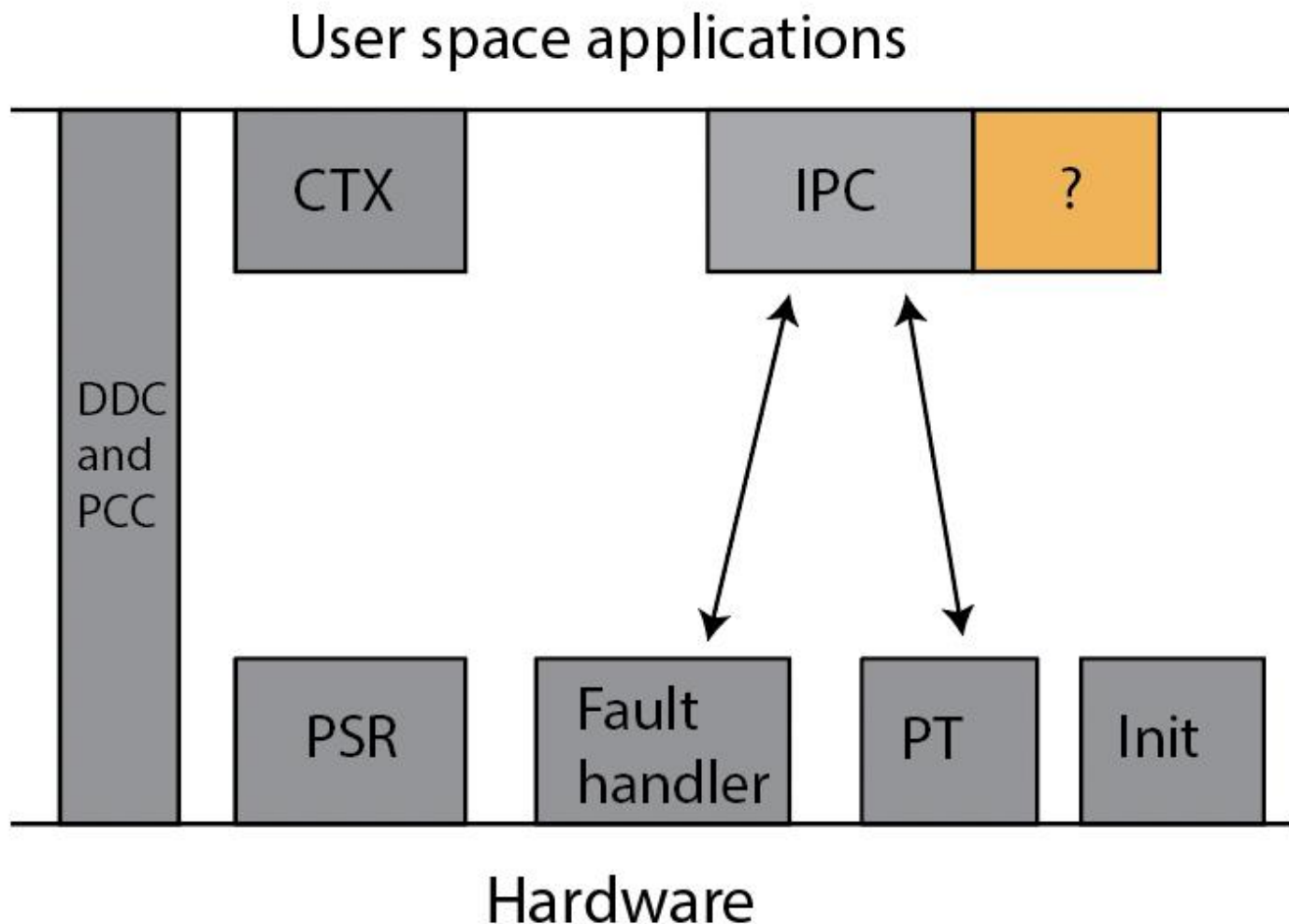
Status at Project Kickoff



Progress So Far



The Nature of the Changes



Key:

CTX - Context switching

PSR - Processor Status Register

DDC - Default Data Capability

PCC - Program Counter Capability

PT - Page Tables

IPC - Inter Process Communication

Init - Initialisation code

Fault handler - New CHERI exceptions

? - Application Binary Interface changes

■ minimum changes for any design

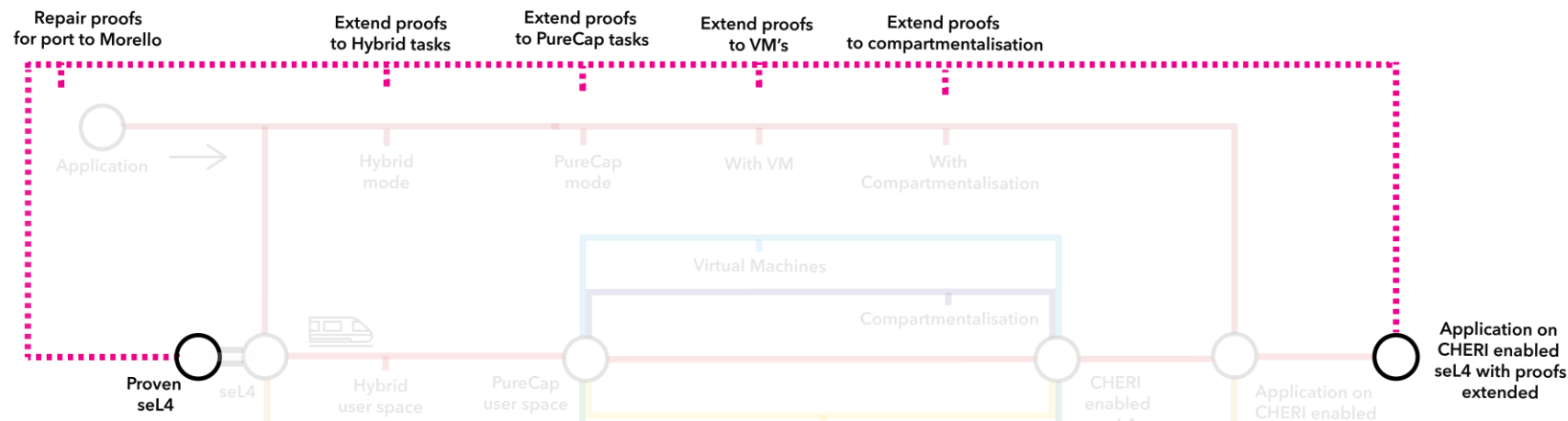
■ optional changes (design choices)

Open Questions

- How are CHERI capabilities managed?
 - User-space management uses IPC
 - CHERI capabilities are virtual addresses
 - How does this interact with shared memory?
- How are CHERI capabilities communicated?
 - Needed for
 - Process creation and control
 - Loading of Purecap processes
 - Fault handling
 - How are they represented in messages?
 - Integration into seL4 permission mechanism?
 - Currently ad hoc, not "principled"
 - Design options are being considered
- Are the answers different for static vs. dynamic systems?

Verification Thoughts

- No active work on this topic, but we are aware of verification impact
- Scoping Impact of Code Changes on Proof Repair
 - Proof roadmap could be influenced by design choices
 - Design decisions could be informed by ease of proof repair
 - Without constraining the R&D effort
- Understanding this will inform route to certification
 - DO-333 discusses formal methods impact on DO-178C
 - Automated tools require qualification as per DO-330



Slides shared by USA DARPA Effort
Performed by Trusted Science and Technology

seL4-based Security and Resilience on ARM Morello

A DARPA Effort performed by Trusted Science and Technology

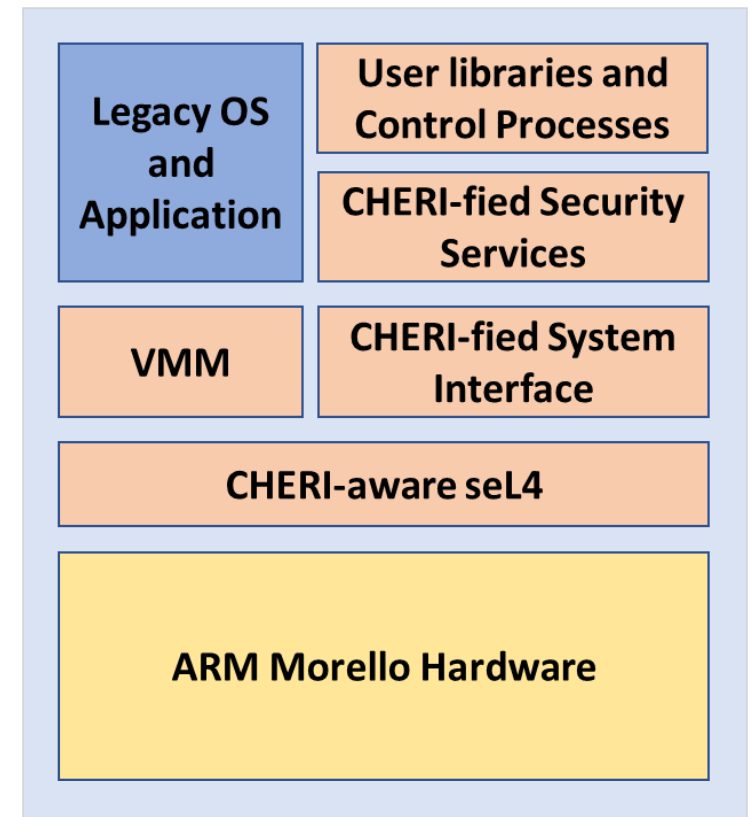
Distribution Statement A – Approved for Public Release, Distribution Unlimited

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Architecture and Strategies

- seL4 should remain **MMU-facing** for coarse-grain memory isolation at the page level
- seL4 can be made **CHERI-aware**, hosting virtual machines and user applications
- CHERI capability enforcement and delegation will be handled by the hardware (low-level)
- High-level capability management will be achieved by **security services**

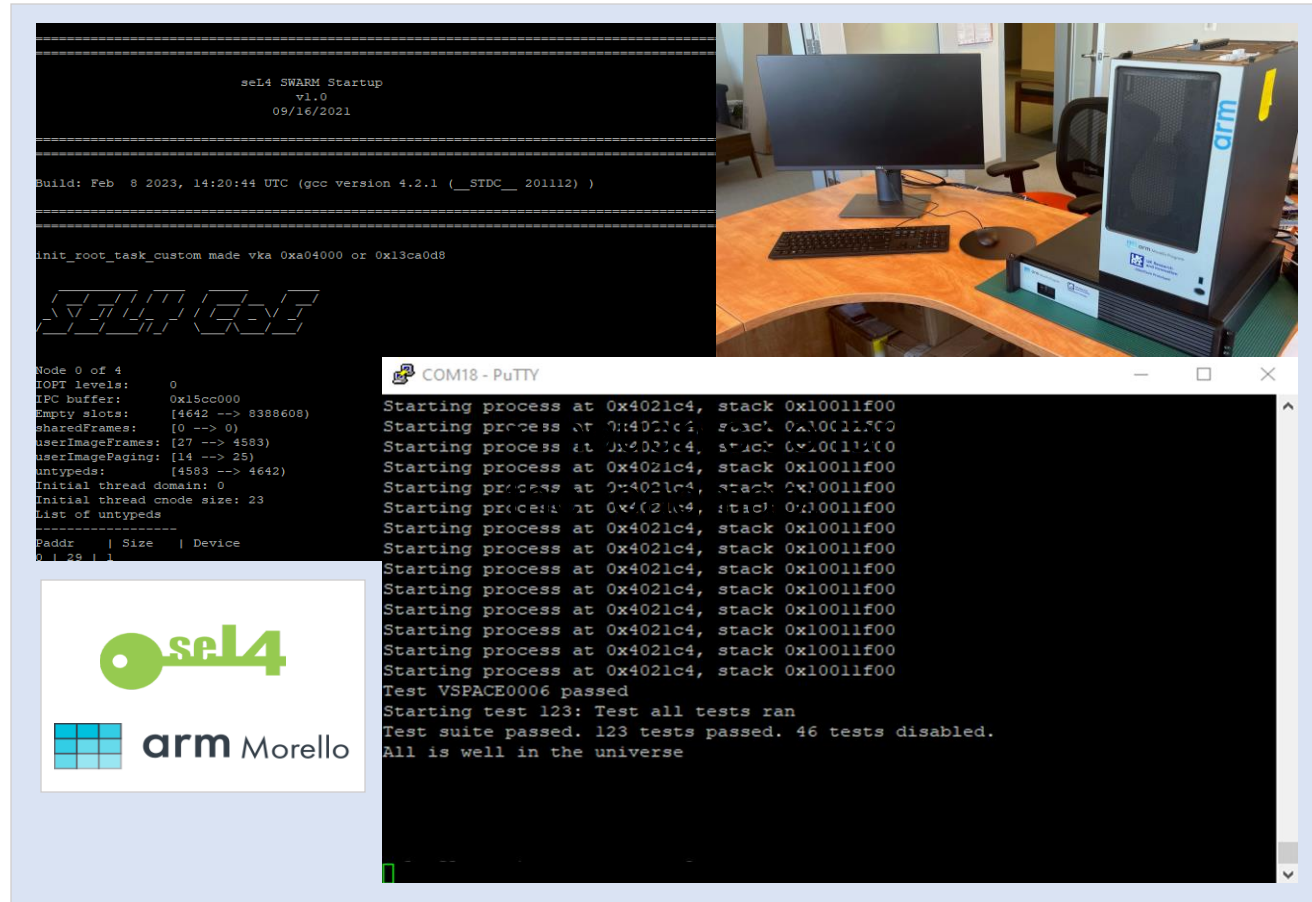


Notional system architecture using Morello and seL4

Progress to Date

- Ported seL4-based security and resilience software stack on ARM Morello platforms
- Designed and added a shim layer to leverage both the CHERI capability model and seL4 capability model
- Designed security services and workflow to monitor information flow and enforce necessary security policies
- Ported seL4-based VMM to support legacy OS and user applications
- “CHERI-fied” necessary security services
- Integrated the end-to-end workflow
- Preparing the relevant demonstration for security and resilience

Some Results



A Morello system with seL4-based test results

Summary

- seL4 and CHERI / Morello are complimentary
 - Both can add to the security of a *system*
- Porting seL4 to Morello is like porting to a new architecture
 - But one **very** similar to Aarch64
- Further work required to make the best of the combination
 - Lots of interesting research questions to work on

Contact

cherisel4@mca-ltd.com

Credits

Sid Agrawal: CHERI, seL4

Nick Spinale: seL4

Paul McKernan: Defence, Certification

Mitali Atkins: Graphics, Design

Acknowledgements

Arm Research

DSbD, DASA, DSTL

DARPA, TrustedST



www.mca-ltd.com