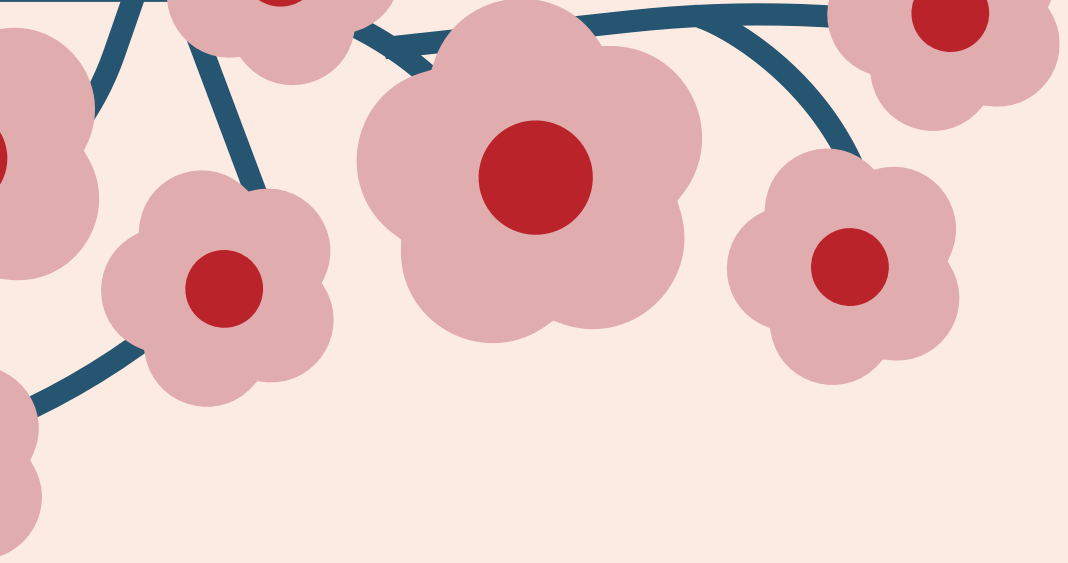
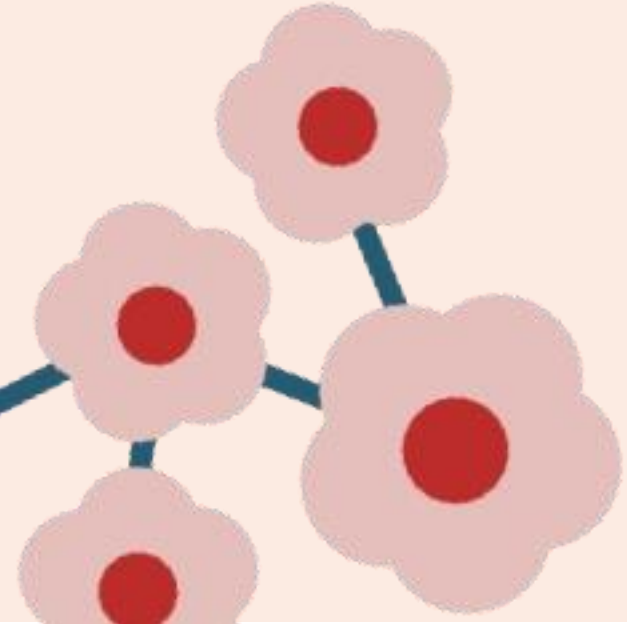
10 March 2026



CHERI

CHERI-enabled RISC-V VP

Capabilities and Applications



Luca Müller
Researcher – German Research Center for AI

dfki ai

Speaker information

- **Researcher** at the German Research Center for Artificial Intelligence
- End of first year towards **doctorate** at University of Bremen
- Interested in **security verification** of hardware and software
- **Team**: Spandan Das, Khushboo Qayyum, Jan Zielasko, Prof. Christoph Lüth, Prof. Rolf Drechsler

Luca Müller

dfki ai

CHERI

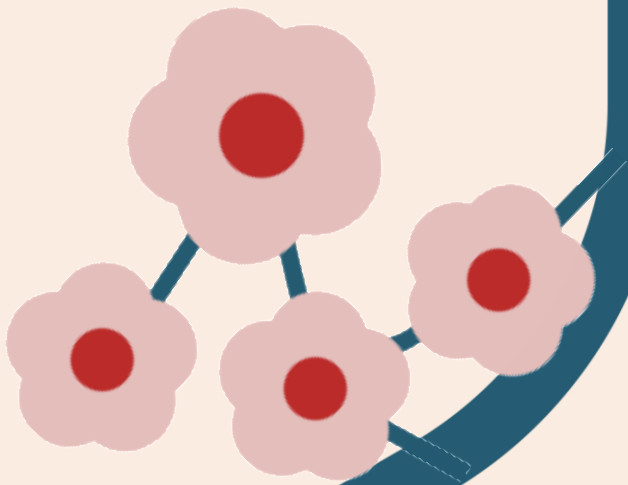
Agenda

- Why Virtual Prototypes?
- **Capabilities**: CHERI-VP Platform
- **Application**: Verification with TestRIG
- **Application**: Detection of security vulnerabilities
- **Application**: In-Memory Computing
- Future directions

Why virtual prototypes?

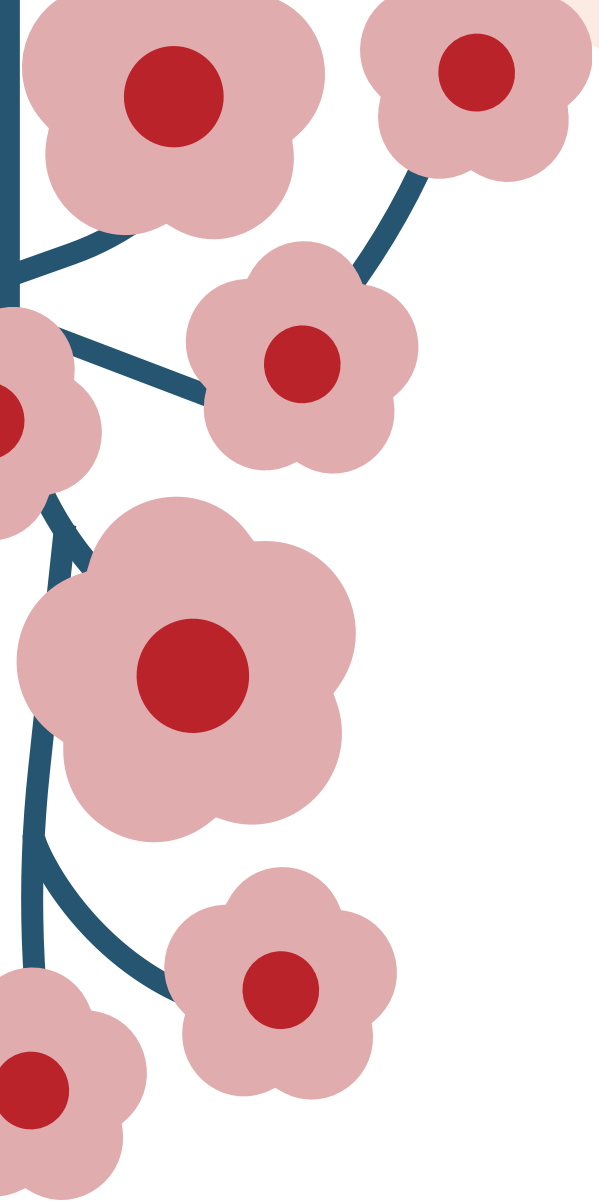
- Waiting for a full RTL implementation **delays software testing** and **hinders hardware-software co-design**
- Virtual prototypes enable **early evaluation** of both hardware and software by providing an executable model
- A **variety of platforms** can be modelled by VPs, ranging from embedded devices to large-scale systems
- Many **evaluations related to CHERI** can be done on VPs at an early stage

Capabilities



CHERI-VP Platform

- Built on **RV32GC** RISC-V VP implemented in **SystemC TLM 2.0**
- CHERI-VP adds support for **32-bit pure-capability** RISC-V binaries
- **Extension** to instruction set simulator, memory model, TLM bus, ...
- C/C++ software is compiled with **custom bootstrap and linker script**



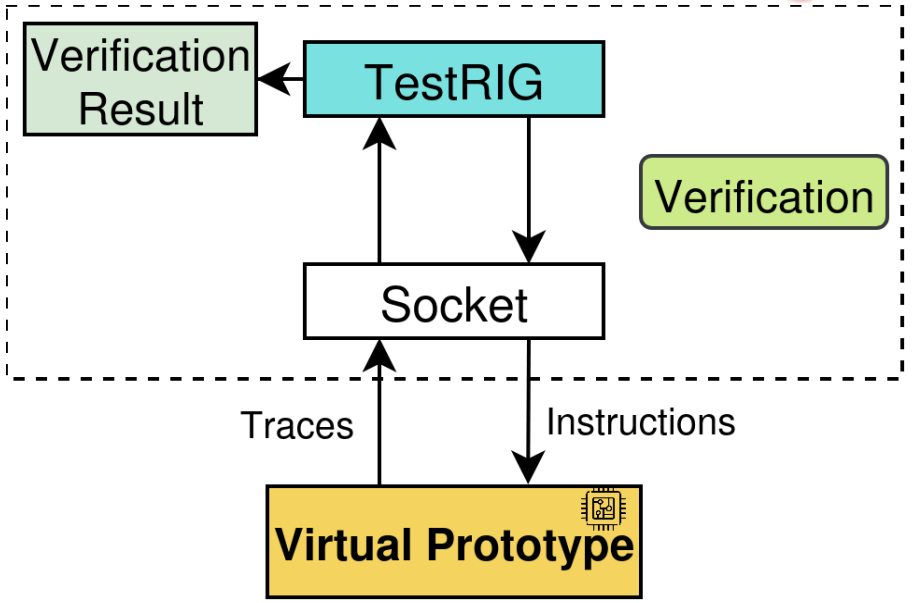
Design challenges: Our takeaways

1. Largest part of implementation effort went towards **new CHERI instructions**
2. Proper handling of **tag memory** posed the biggest challenge
3. CHERI technical report and SAIL model serve as **useful references**

Applications

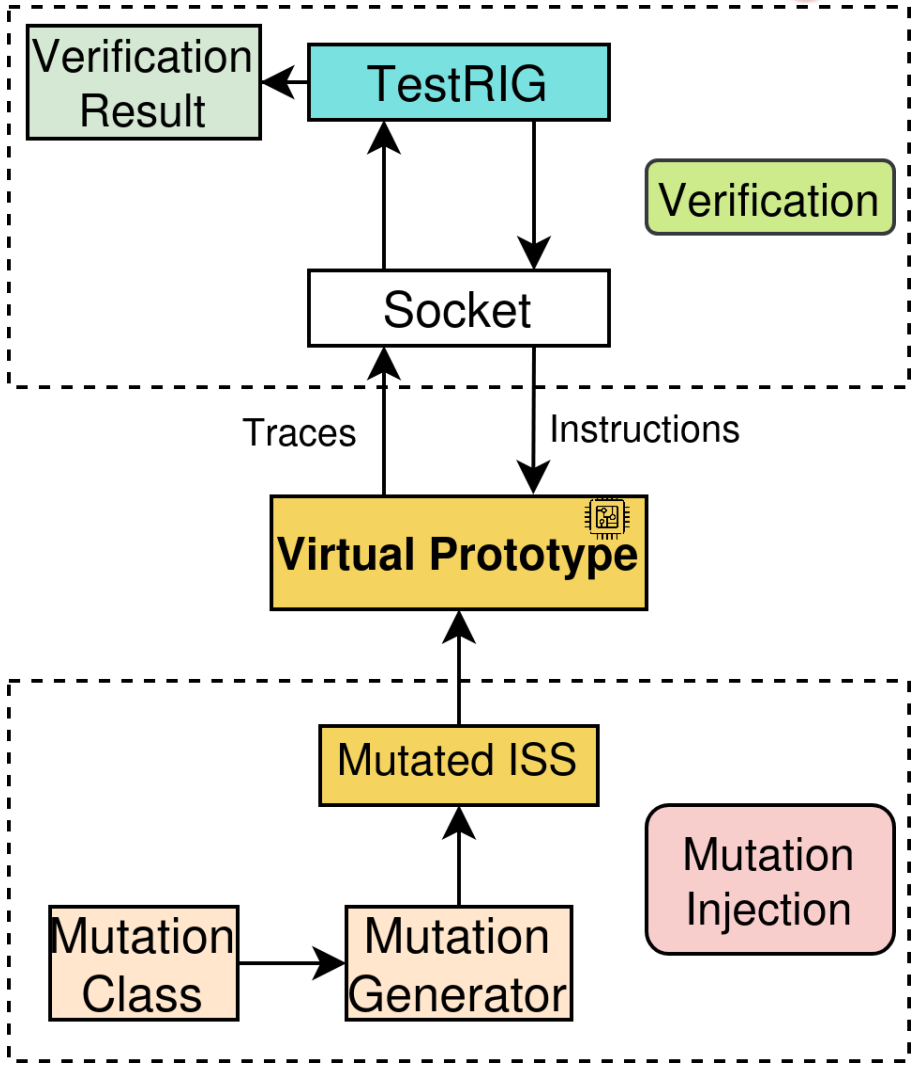
Verification with TestRIG

- To verify our CHERI-VP implementation, we add **support for TestRIG**
- Tests were able to **find issues** related to tag encoding and exception handling



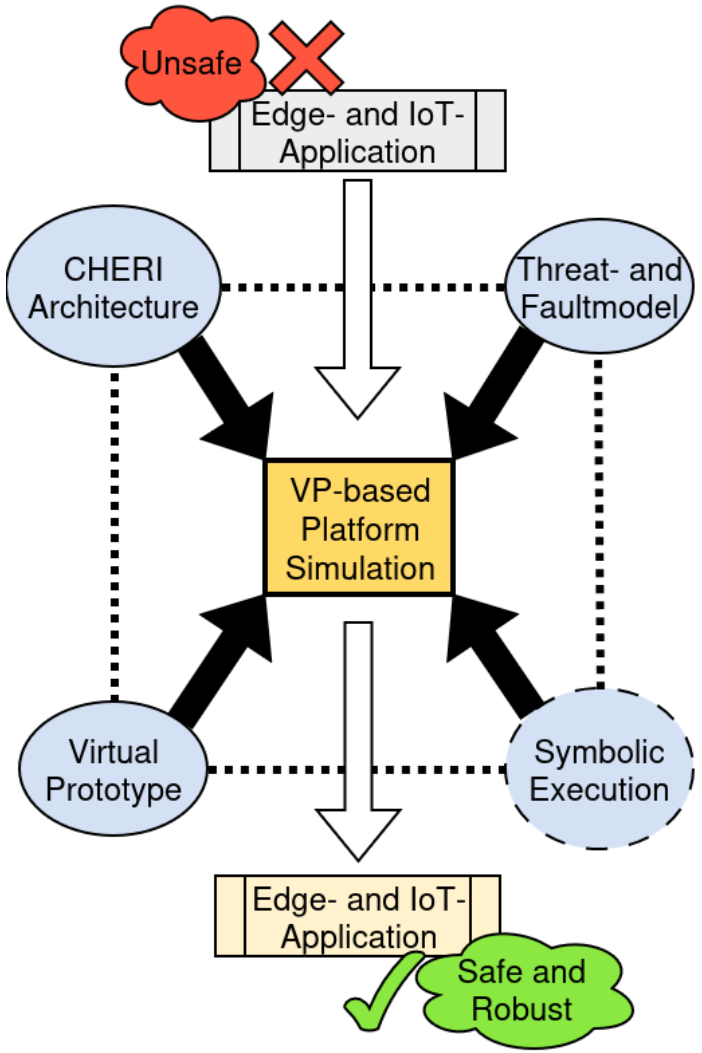
Verification with TestRIG

- To verify our CHERI-VP implementation, we add **support for TestRIG**
- Tests were able to **find issues** related to tag encoding and exception handling
- Ongoing work: Evaluate effectiveness of TestRIG instruction stream for **mutation detection**



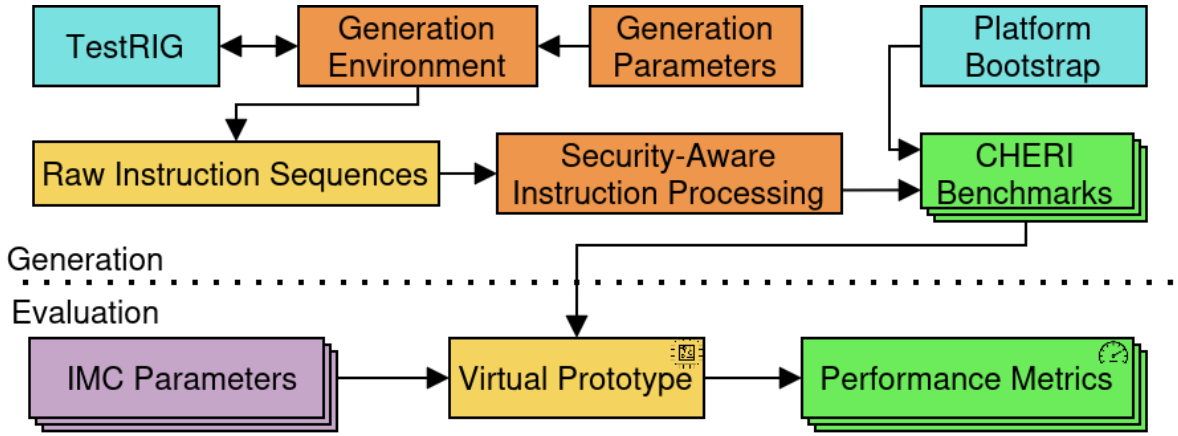
Detecting Security Vulnerabilities

- Capability violation faults can help with **detection of security vulnerabilities** in existing software
- Mapping to **Common Weakness Enumeration** unlocks detection of vulnerabilities which can be mitigated by CHERI
- Early evaluation shows effectiveness for **CWE examples**, showing promise for larger-scale software



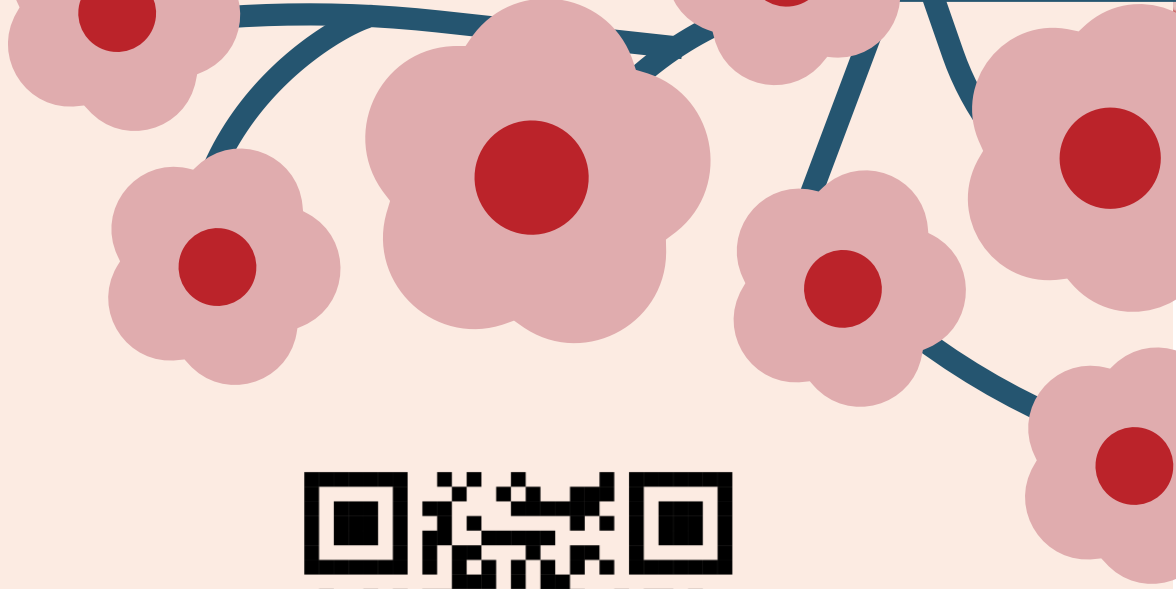
In-Memory Computing

- Explore speedups by using **in-memory computing** for tag memory operations
- TestRIG is used to generate **security-aware instruction sequences**
- **Speedups of 6-11%** were achieved under ideal conditions



Future Directions

- Further **verification** of CHERI-VP implementation
- Integration of **symbolic execution** for systematic exploration of software
- Evaluation of **in-memory computing** under different conditions



CHERI

THANK YOU

Let's connect!

Contact luca.mueller@dfki.de

Web www.dfki.de

dfki ai