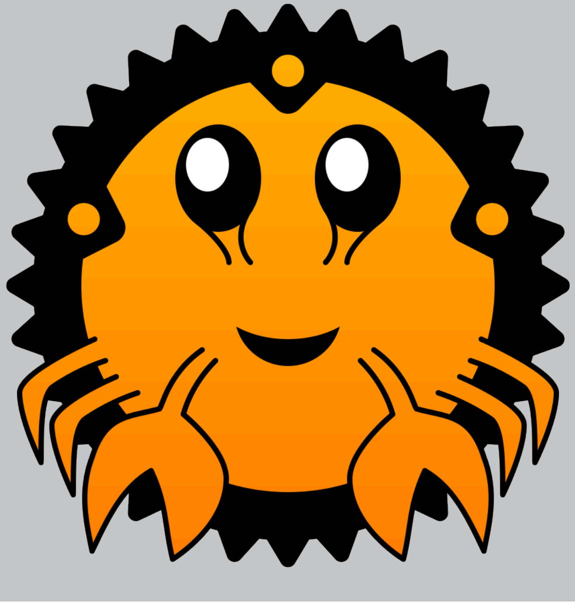


News from the Hermit Crab

From Soundness Foundations to GPU Virtualization

Martin Kröning

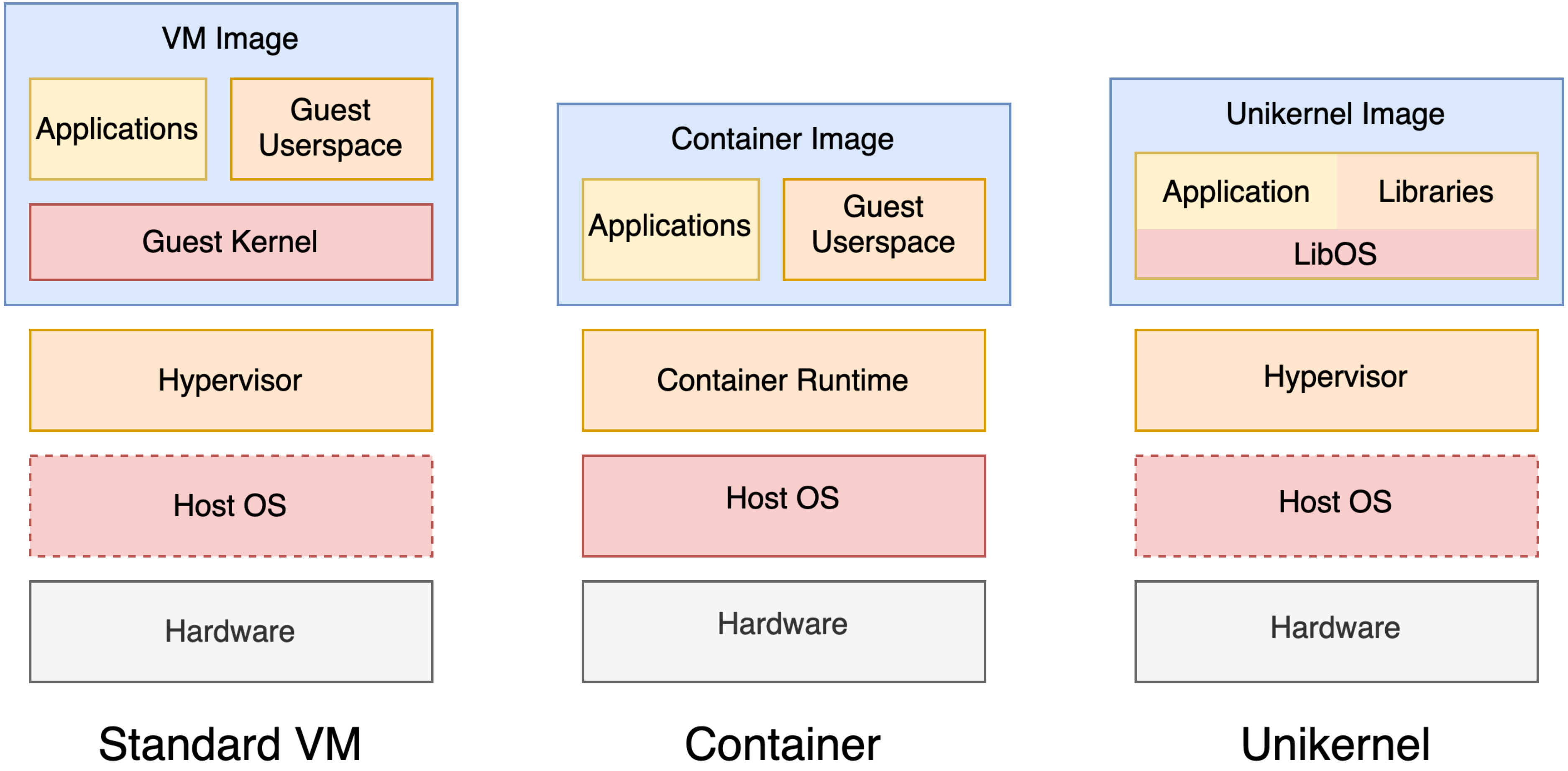
E.ON Energy Research Center | RWTH AACHEN UNIVERSITY

Agenda

1. Introduction to Hermit
2. Interesting Internals
3. GPU Virtualization with Cricket
4. Application & Kernel Profiling

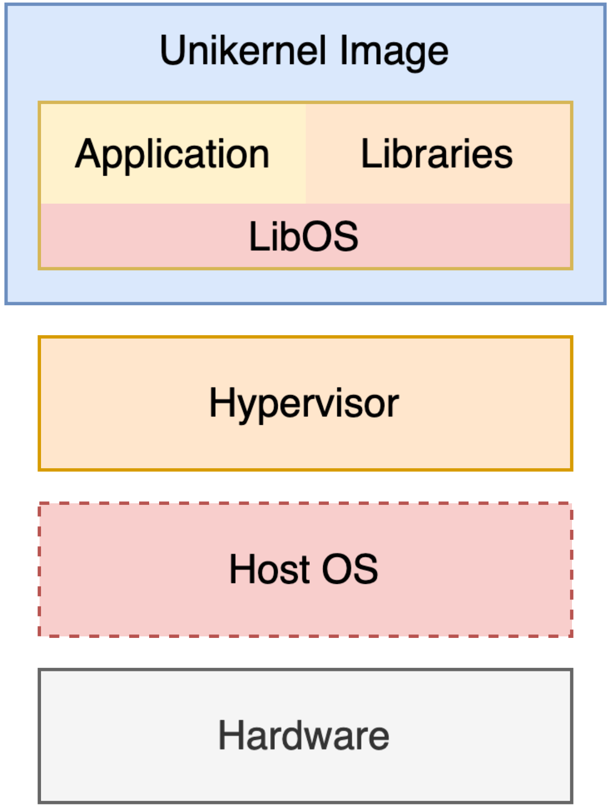
Introduction to Hermit

Unikernels


VM Image
Applications
Guest Userspace
Guest Kernel
Hypervisor
Host OS
Hardware
Standard VM
Container Image
Applications
Guest Userspace
Container Runtime
Host OS
Hardware
Container
Unikernel Image
Application
Libraries
LibOS
Hypervisor
Host OS
Hardware
Unikernel

Unikernels

- Specialized for use cause
  - Tiny images
- One process per image
  - No isolation necessary
- Single address space operating system
  - No address space context switch
- Single privilege level
  - No privilege context switch
- System calls are just function calls



Unikernel

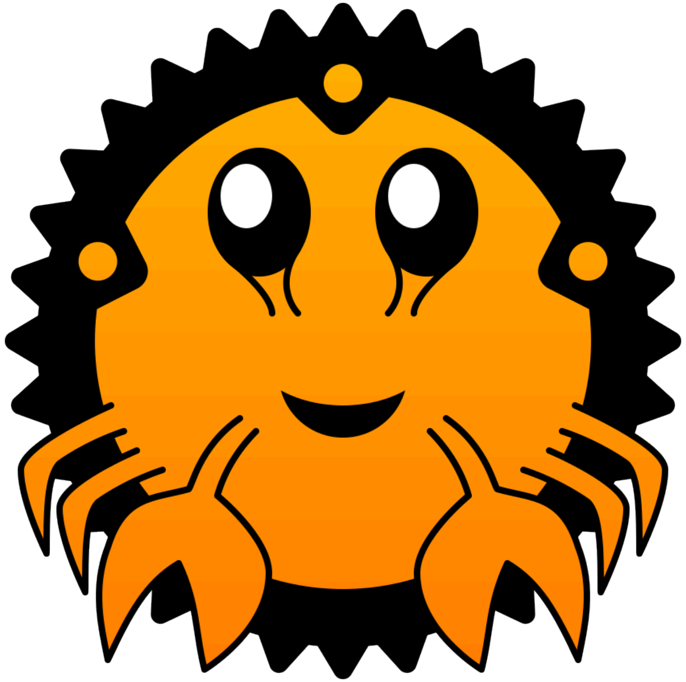
The Hermit Operating System

Overview

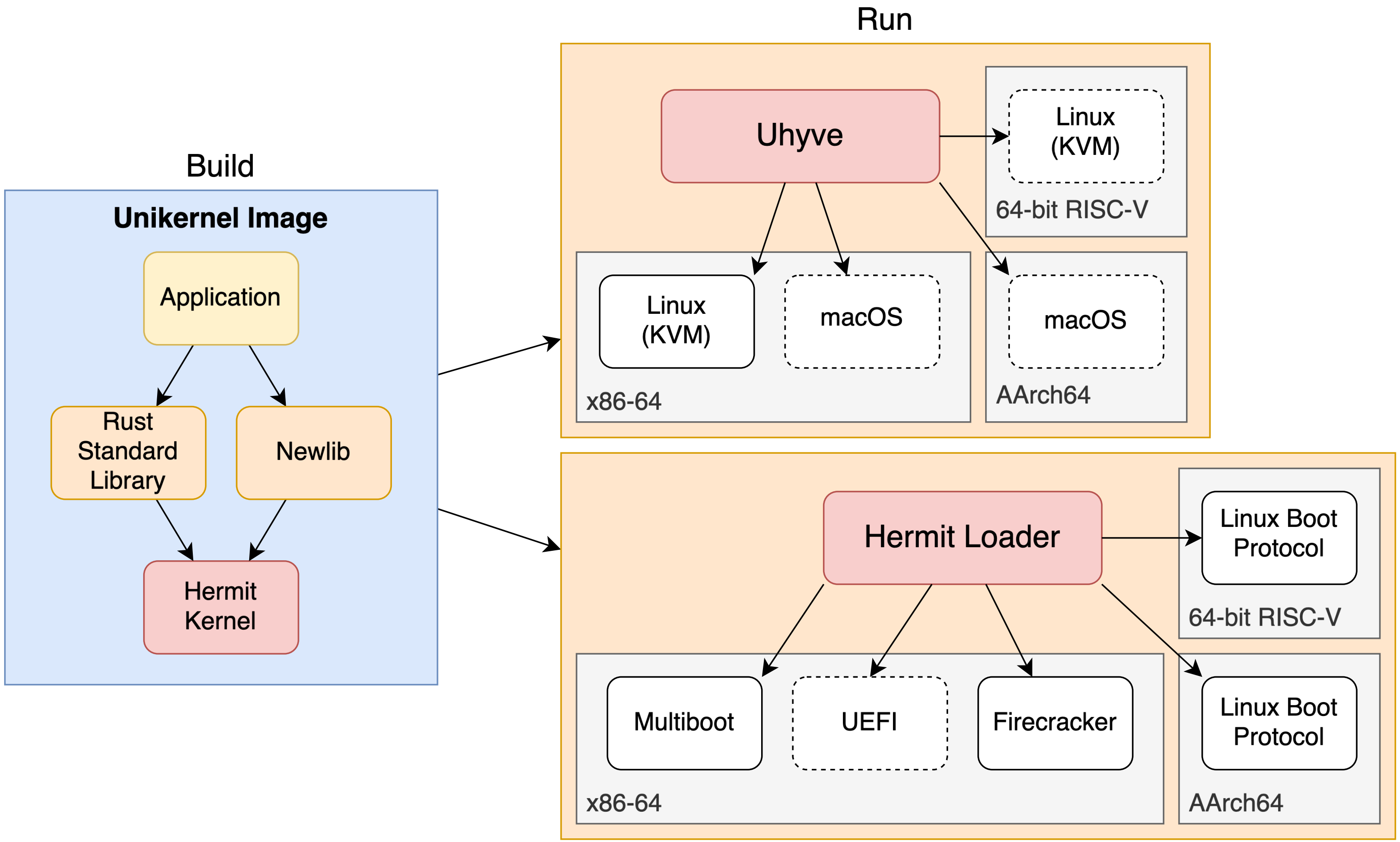
- Written in Rust
- Official tier 3 Rust target for Rust applications
- GCC + Newlib fork for C applications

Features

- Multi-core support
- Easily configurable
- **New**: Compiles on Windows
- **New**: Supports stable Rust through rust-std-hermit

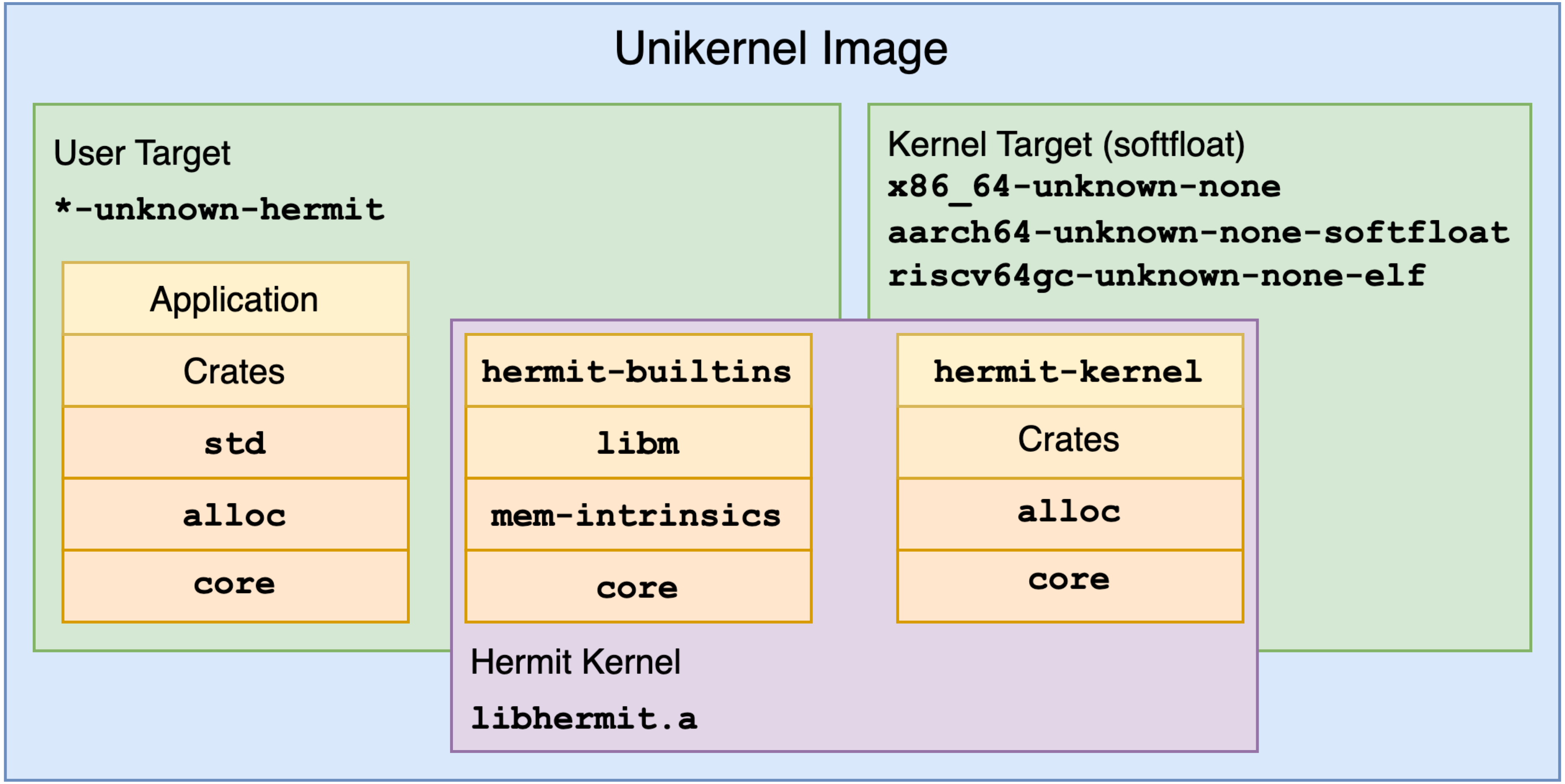


Platform Support



Interesting Internals

Hermit Image

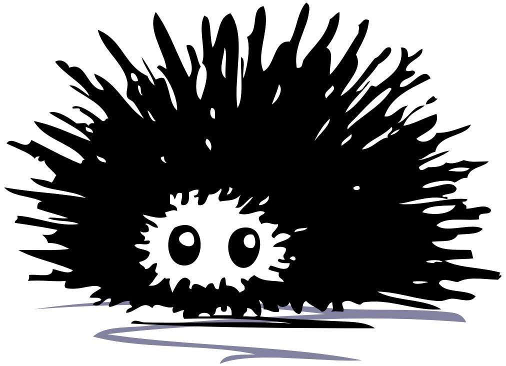


Soundness Foundations

On the Challenge of Sound Code for Operating Systems

DOI: 10.1145/3623759.3624554

Goal: Soundness—Safety must not require context!



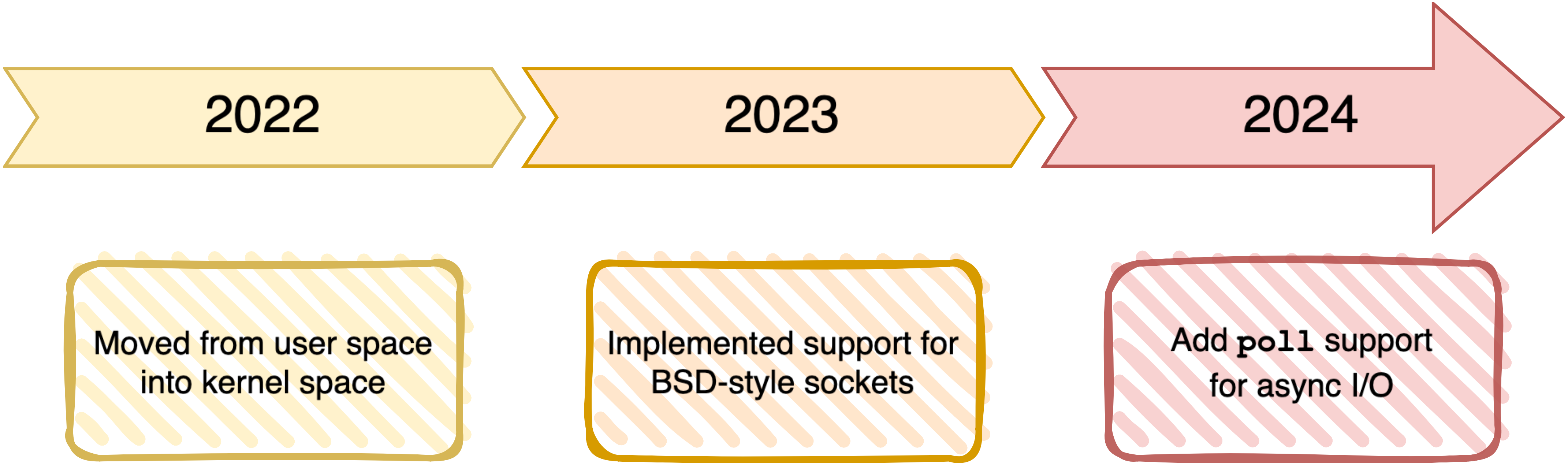
hermit-sync

- `SpinMutex`
- `OnceCell`
- `Lazy`
- `TakeStatic`
- `InterruptMutex`
- `InterruptRefCell`

count-unsafe

- Counts `unsafe` functions, expressions, implementations, etc.

An Evolving Network Stack



2022

2023

2024

Moved from user space into kernel space

Implemented support for BSD-style sockets

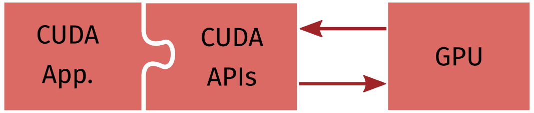
Add `poll` support for async I/O

GPU Virtualization with Cricket

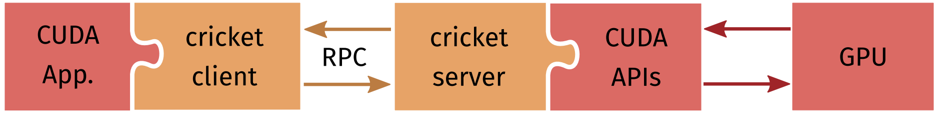
GPU Virtualization with Cricket

github.com/RWTH-ACS/cricket

API Remoting



(a) GPU application without virtualization

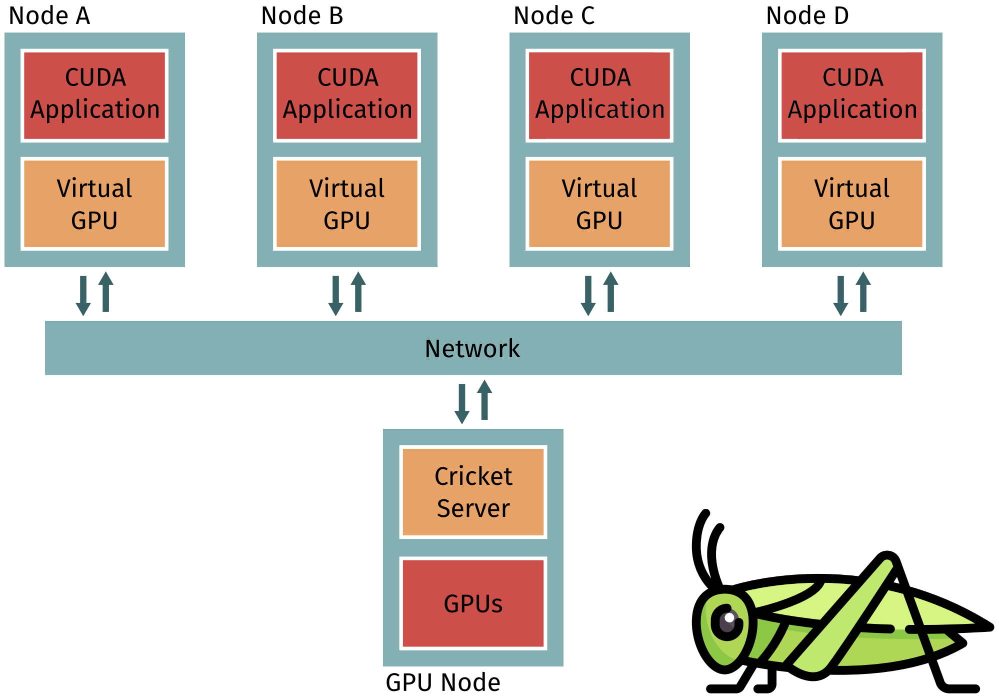


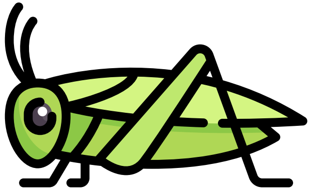
(b) GPU application with virtualization layer

- Separates proprietary device dependent code into separate process
- Allows full control of device interactions
- Low virtualization overhead

Use Cases

Remote execution, scheduling, monitoring





Adapting Cricket for Unikernels

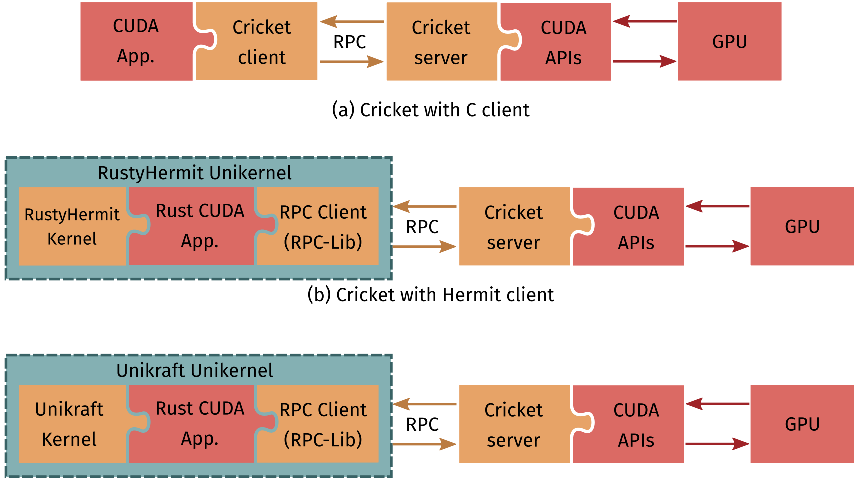
GPU Acceleration in Unikernels Using Cricket GPU Virtualization

DOI: 10.1145/3624062.3624236

API Remoting

- Cricket is based on ONC RPCs
- Reference C impl is old and complex
  - Uses Linux-specific network features
- For unikernels: New Rust impl
- All user code is run inside unikernel
- Only CUDA APIs are run outside

Cricket for Unikernels



(a) Cricket with C client

(b) Cricket with Hermit client

(c) Cricket with Unikraft client

Application & Kernel Profiling

Profiling through Instrumentation with rftrace

github.com/hermit-os/rftrace/tree/next

Rust Source

```rust
fn square(x: i32) -> i32 {
    x * x
}
```

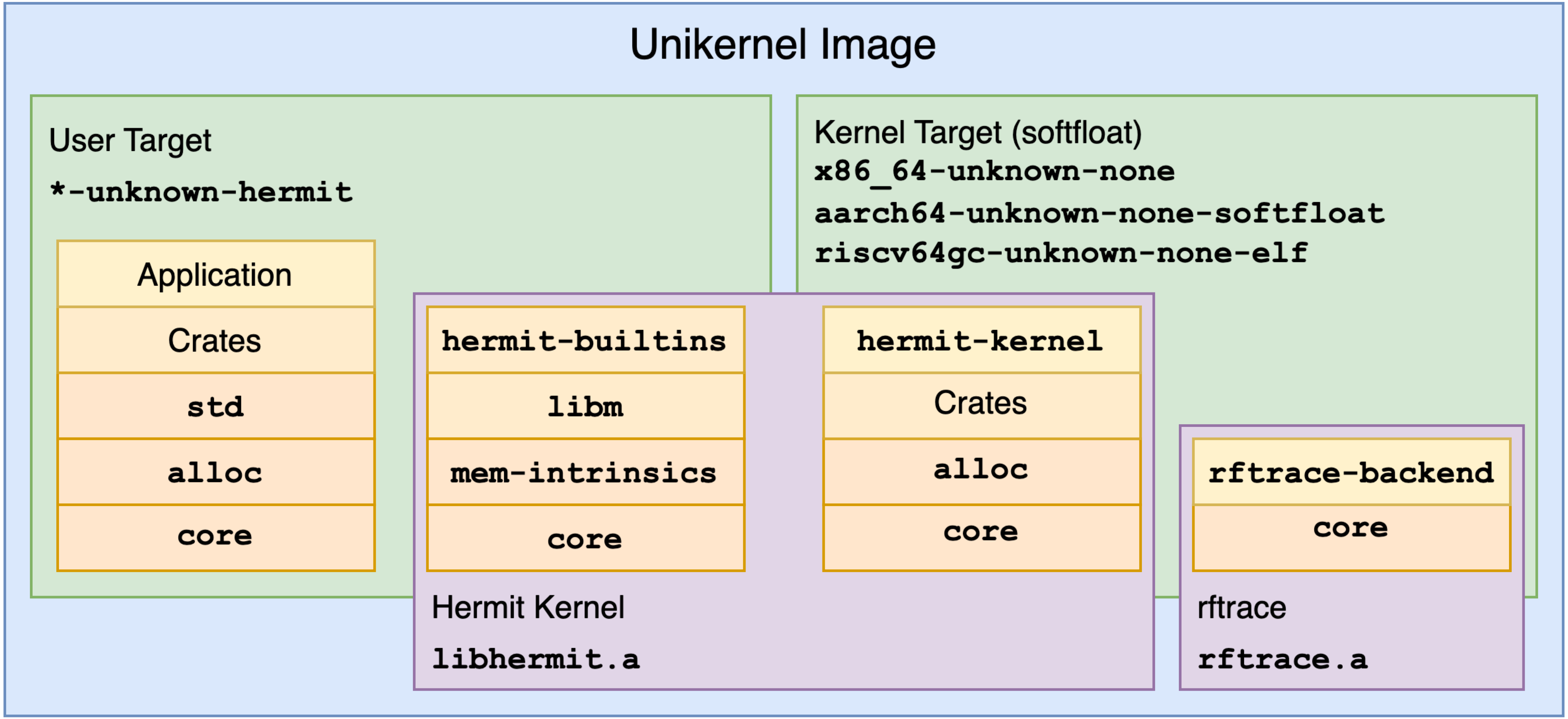
Assembly

```asm
square:
    mov   eax, edi
    imul  eax, edi
    ret
```

Instrumented Assembly

```asm
square:
    push  rbp
    mov   rbp, rsp
    call  mcount
    imul  edi, edi
    mov   eax, edi
    pop   rbp
    ret
```

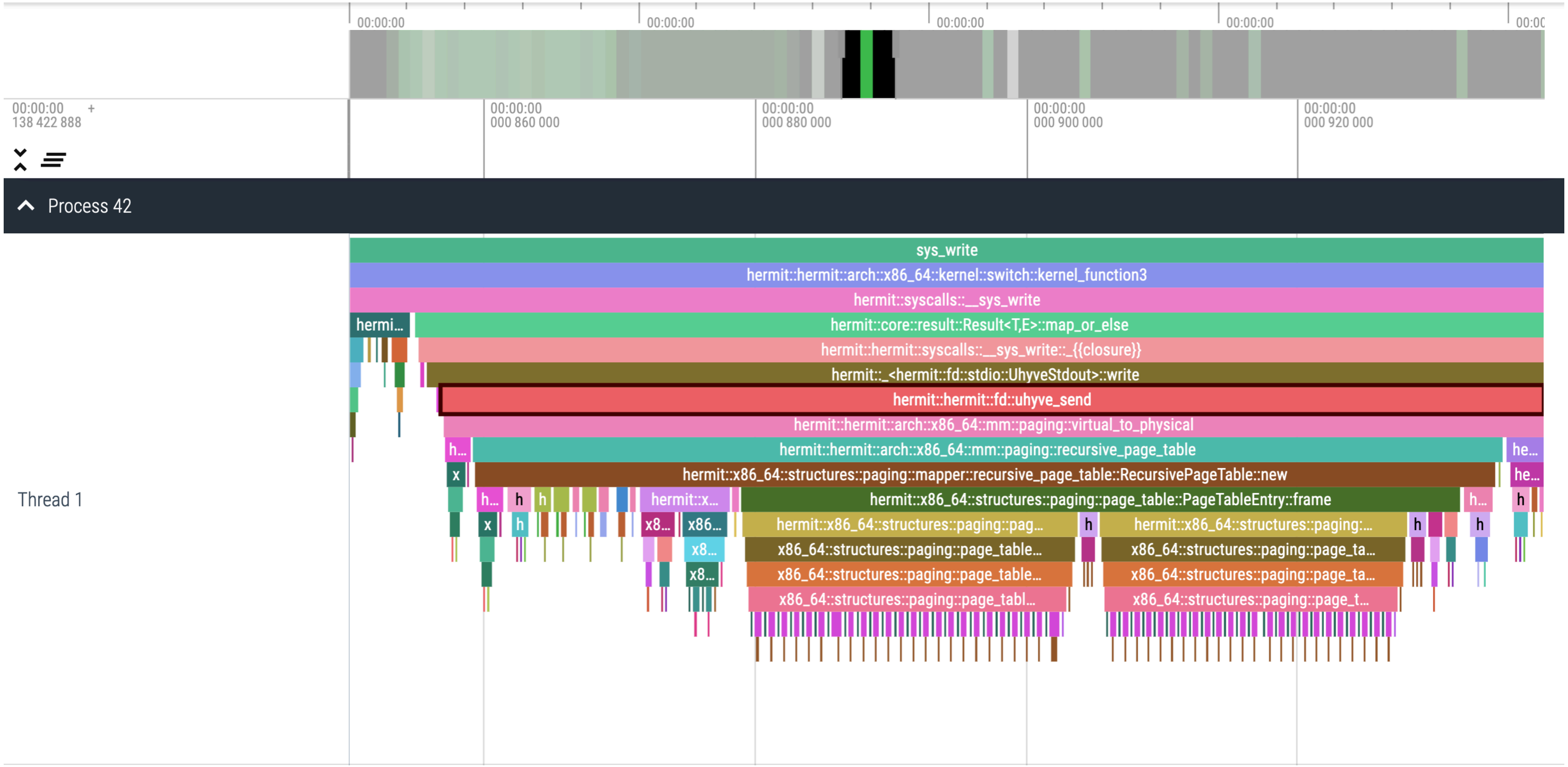
Hermit Image with rftrace



Trace Replay

```
         [     1] | hello_world::test1() {
0.150 us [     1] |   core::fmt::Arguments::new_const();
         [     1] |   sys_write() {
         [     1] |     hermit::hermit::arch::x86_64::kernel::switch::kernel_function3() {
0.136 us [     1] |       hermit::core::ptr::write();
0.130 us [     1] |       hermit::core::ptr::write();
0.134 us [     1] |       hermit::core::ptr::write();
         [     1] |       hermit::hermit::arch::x86_64::kernel::CoreLocal::get() {
0.134 us [     1] |         hermit::x86_64::addr::VirtAddr::zero();
         [     1] |         hermit::x86_64::registers::GsBase::read() {
0.258 us [     1] |           x86_64::registers::model_specific::Msr::read();
         [     1] |           hermit::x86_64::addr::VirtAddr::new() {
```

Trace Visualization


00:00:00
00:00:00
00:00:00
00:00:00
00:0(
00:00:00 +
138 422 888
00:00:00
000 860 000
00:00:00
000 880 000
00:00:00
000 900 000
00:00:00
000 920 000
Process 42
Thread 1
sys_write
hermit::hermit::arch::x86_64::kernel::switch::kernel_function3
hermit::syscalls::__sys_write
hermi…
hermit::core::result::Result<T,E>::map_or_else
hermit::hermit::syscalls::__sys_write::_{{closure}}
hermit::_<hermit::fd::stdio::UhyveStdout>::write
hermit::hermit::fd::uhyve_send
hermit::hermit::arch::x86_64::mm::paging::virtual_to_physical
h…
hermit::hermit::arch::x86_64::mm::paging::recursive_page_table
he…
x
hermit::x86_64::structures::paging::mapper::recursive_page_table::RecursivePageTable::new
he…
h…
h
h
hermit::x…
hermit::x86_64::structures::paging::page_table::PageTableEntry::frame
h…
h
x
h
x8…
x86…
hermit::x86_64::structures::paging::pag…
h
hermit::x86_64::structures::paging:…
h
h
x8…
x86_64::structures::paging::page_table…
x86_64::structures::paging::page_ta…
x8…
x86_64::structures::paging::page_table…
x86_64::structures::paging::page_ta…
x86_64::structures::paging::page_tabl…
x86_64::structures::paging::page_t…

Acknowledgments

SPONSORED BY THE

Federal Ministry
of Education
and Research

Funded by
the European Union

Thank you for your kind attention!

Check us out GitHub: github.com/hermit-os

Come say hi on Zulip: hermit.zulipchat.com

Martin Kröning – martin.kroening@eonerc.rwth-aachen.de

Institute for Automation of Complex Power Systems
E.ON Energy Research Center, RWTH Aachen University
Mathieustraße 10
52074 Aachen

www.acs.eonerc.rwth-aachen.de