General-purpose computing with VirtualBox on Genode/NOVA

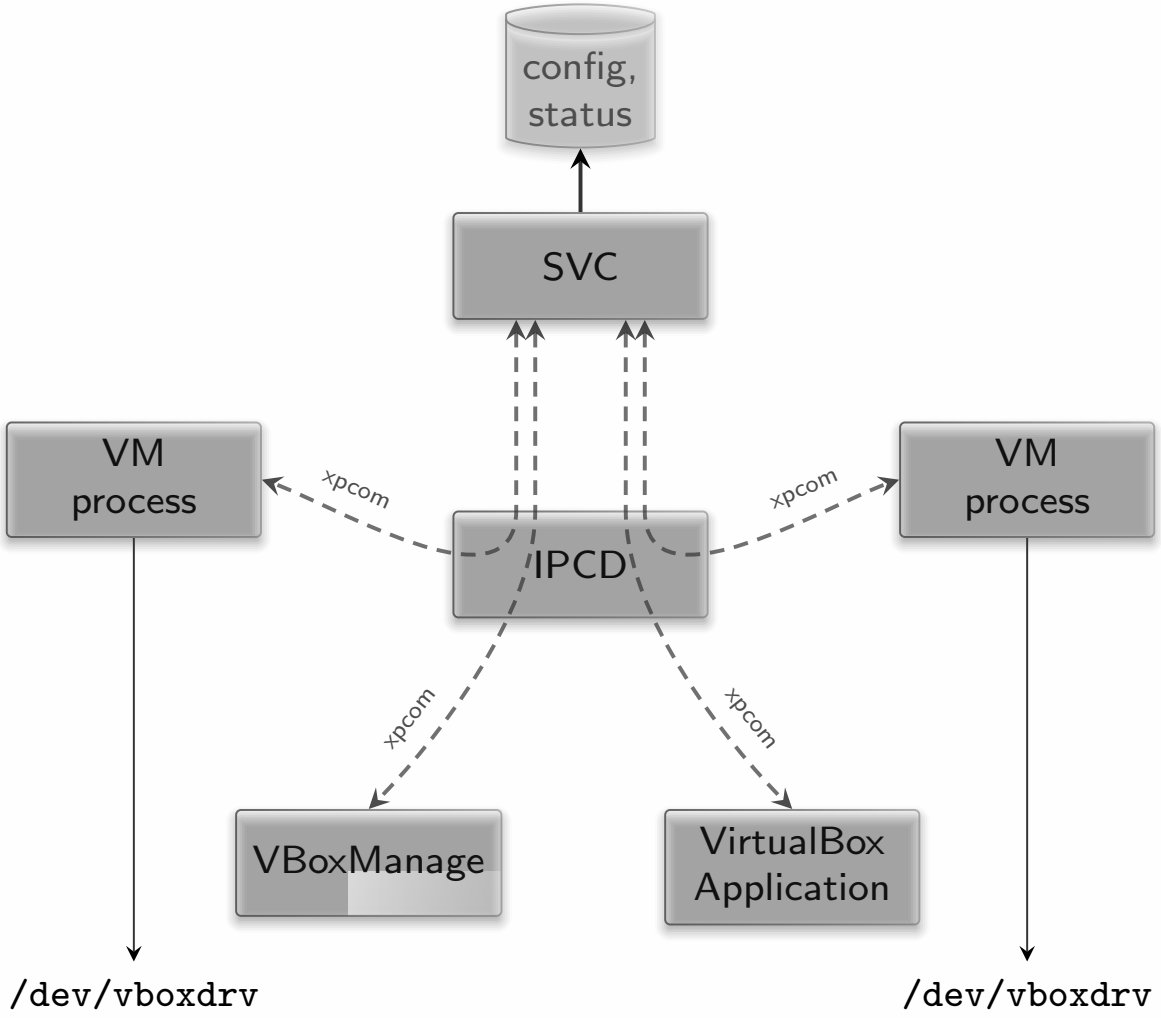
Norman Feske
<norman.feske@genode-labs.com>

Outline

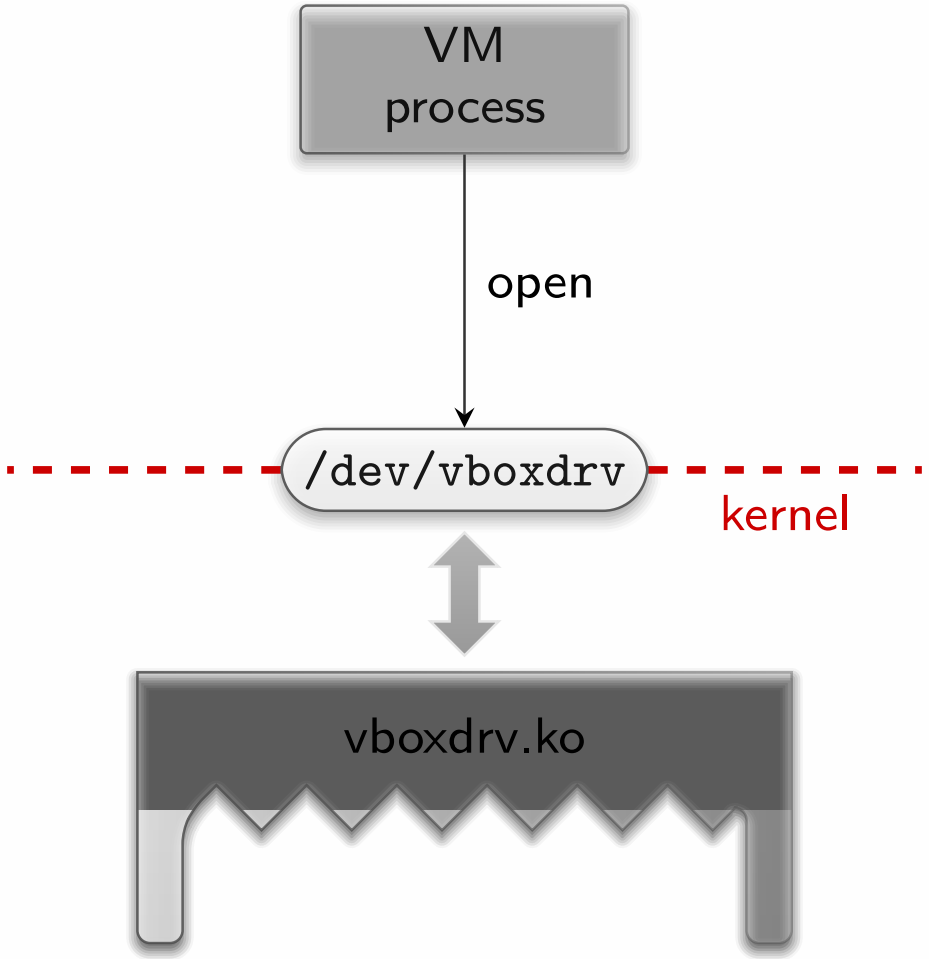
1. VirtualBox
2. NOVA microhypervisor and Genode
3. Transplantation of VirtualBox to NOVA
4. Demo
5. War stories
6. Project Turmvilla
7. The Book “Genode Foundations”

Outline

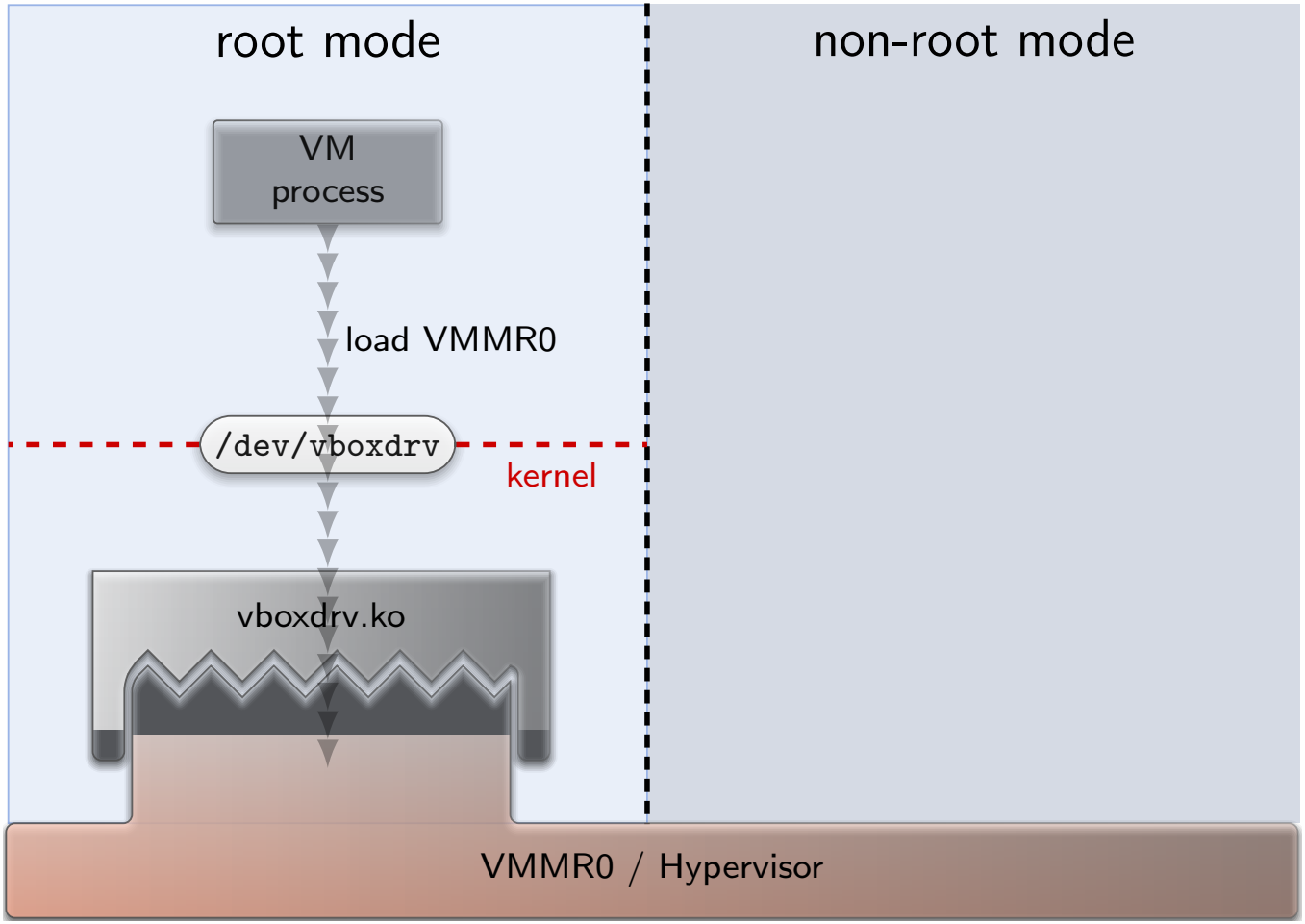
Architecture overview



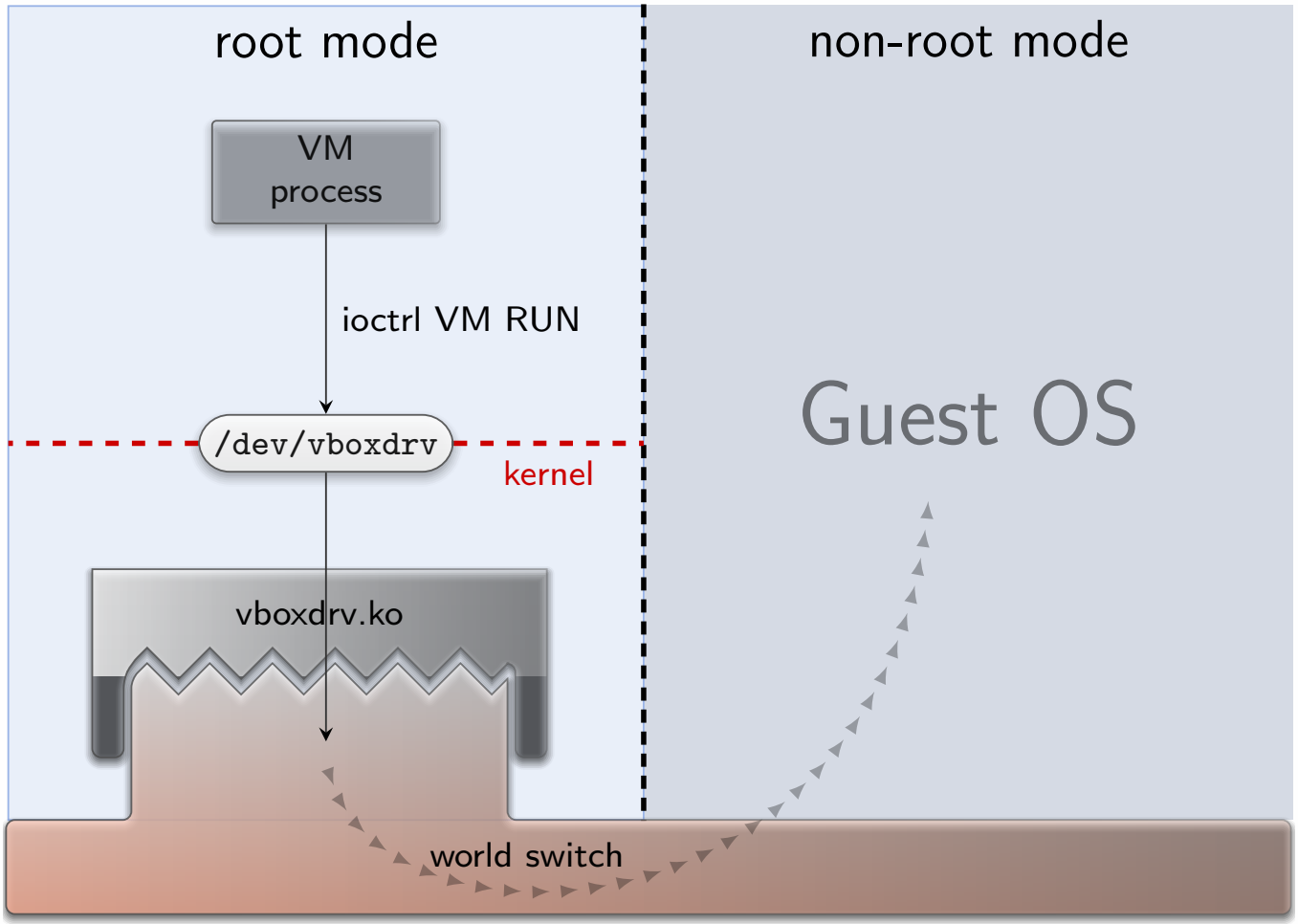
Starting up a VM process



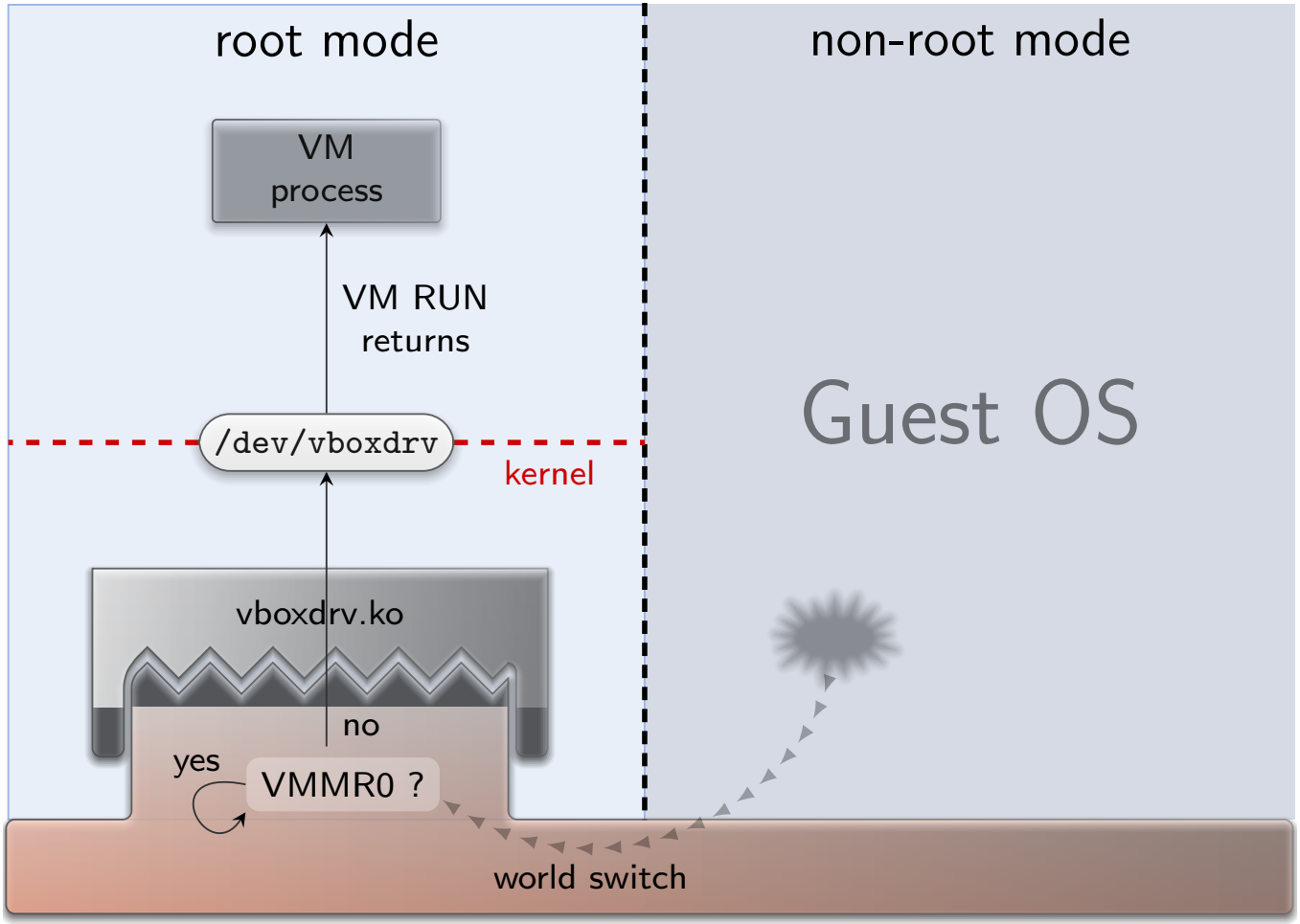
VM process running



Entering the Guest OS

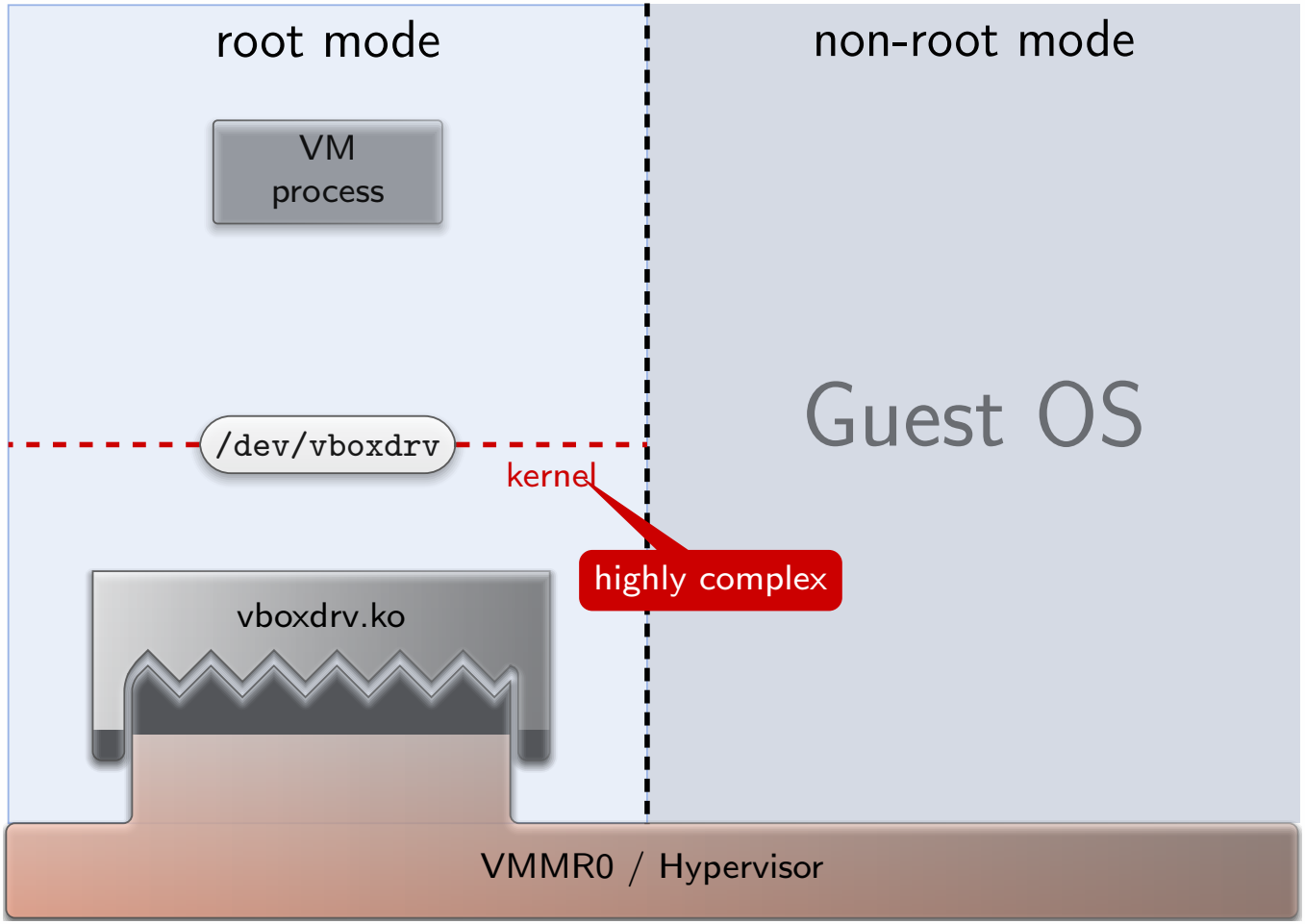


Flow of a virtualization event

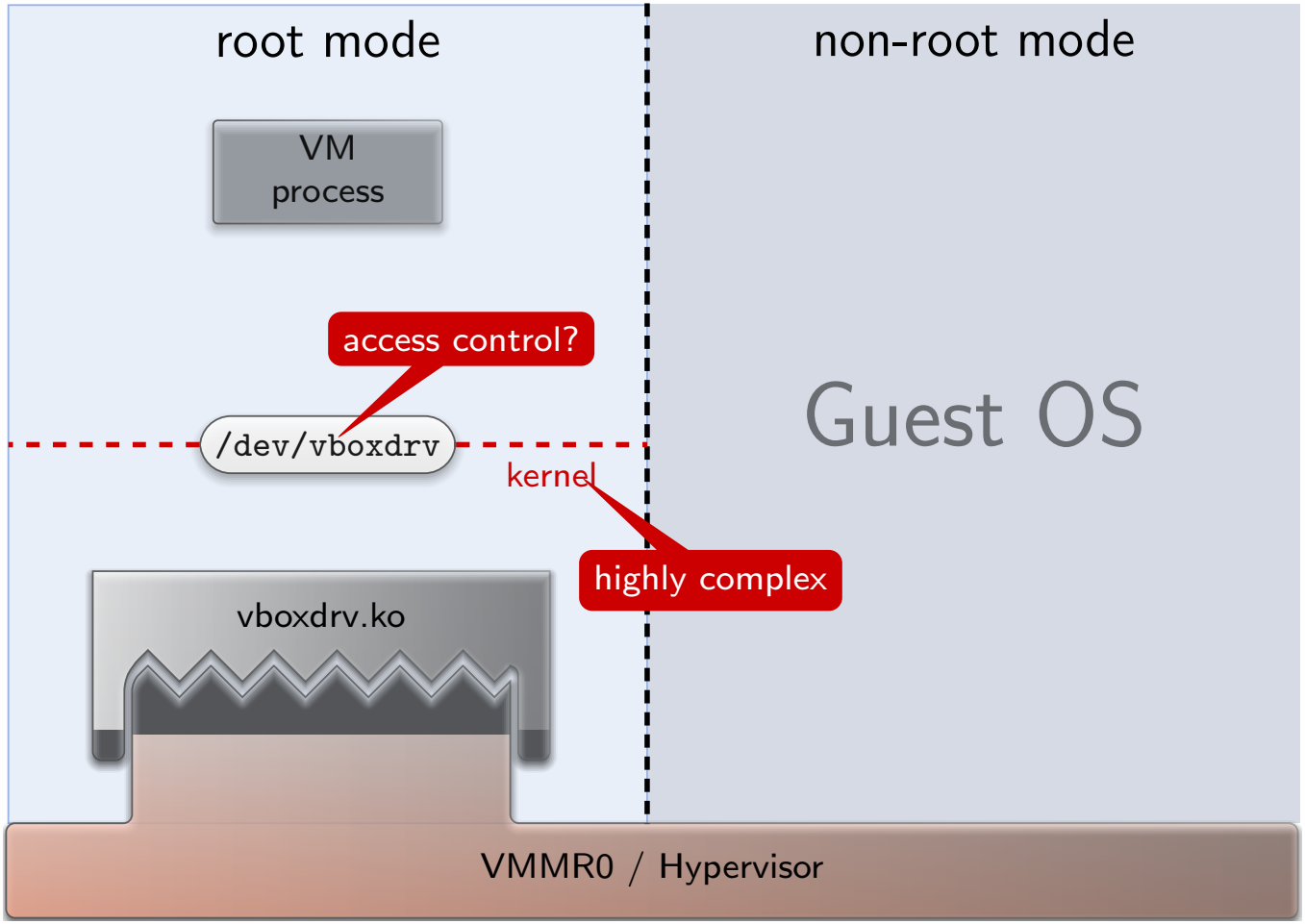


Risks for desktop virtualization

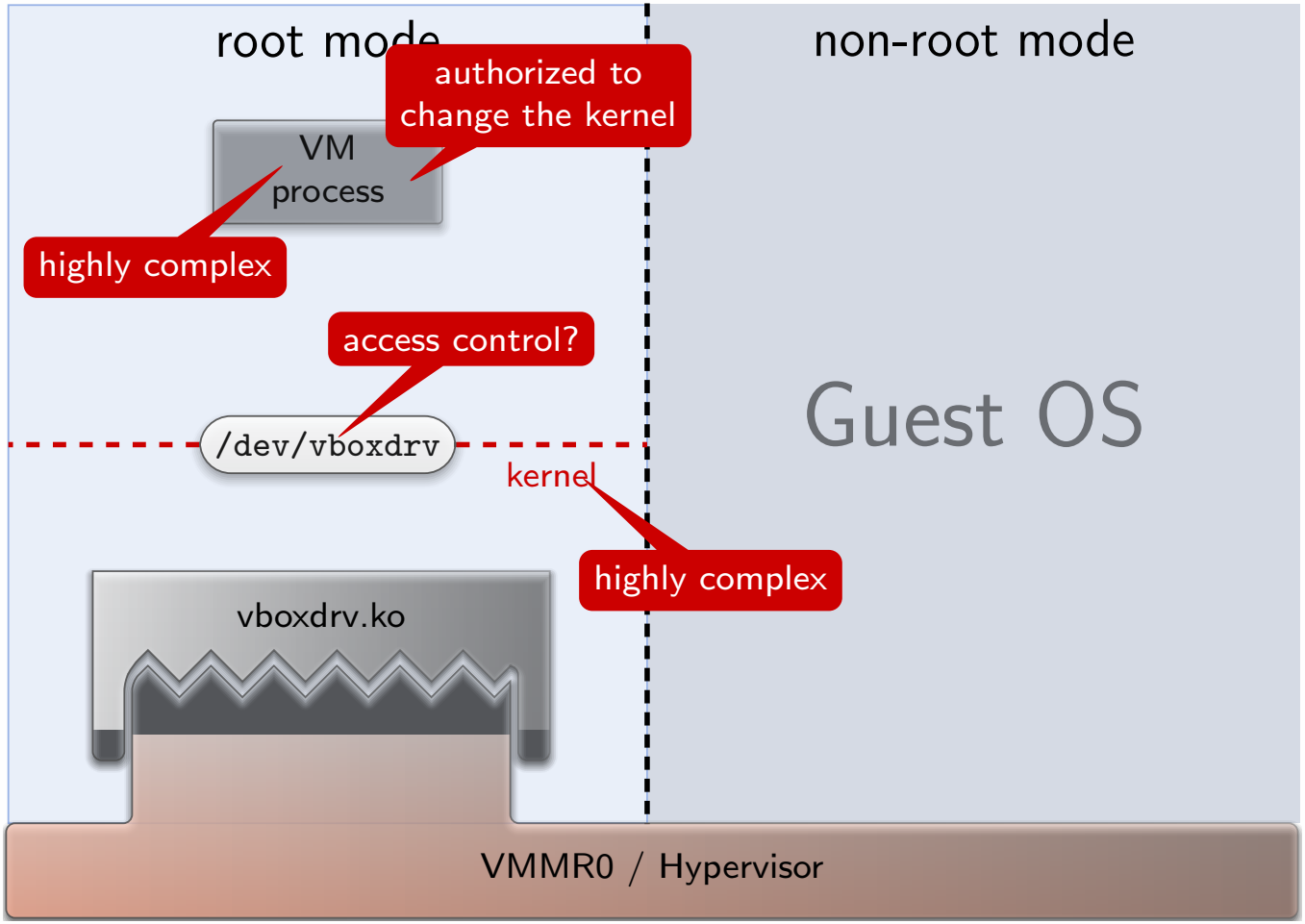
Risks for desktop virtualization



Risks for desktop virtualization

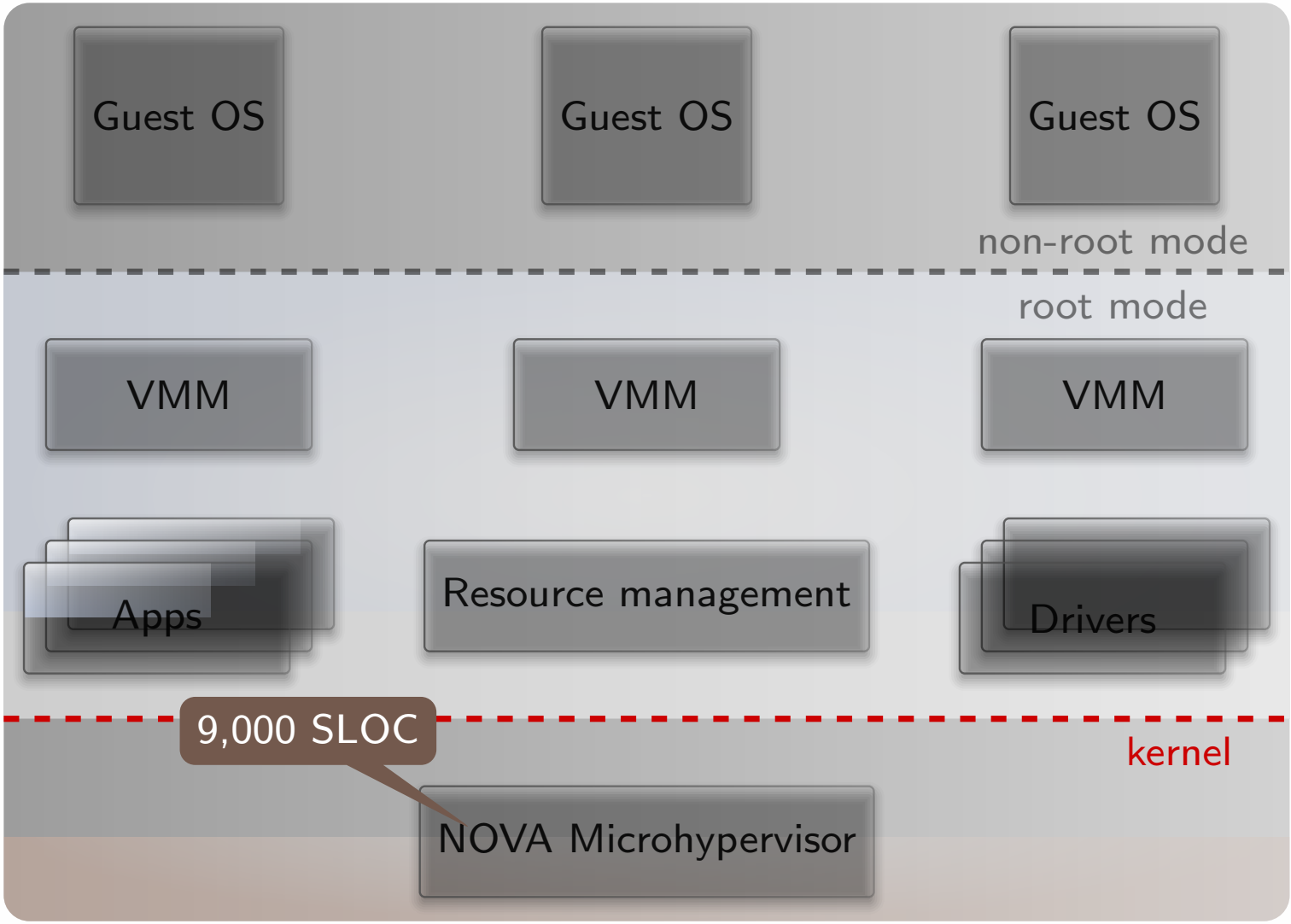


Risks for desktop virtualization

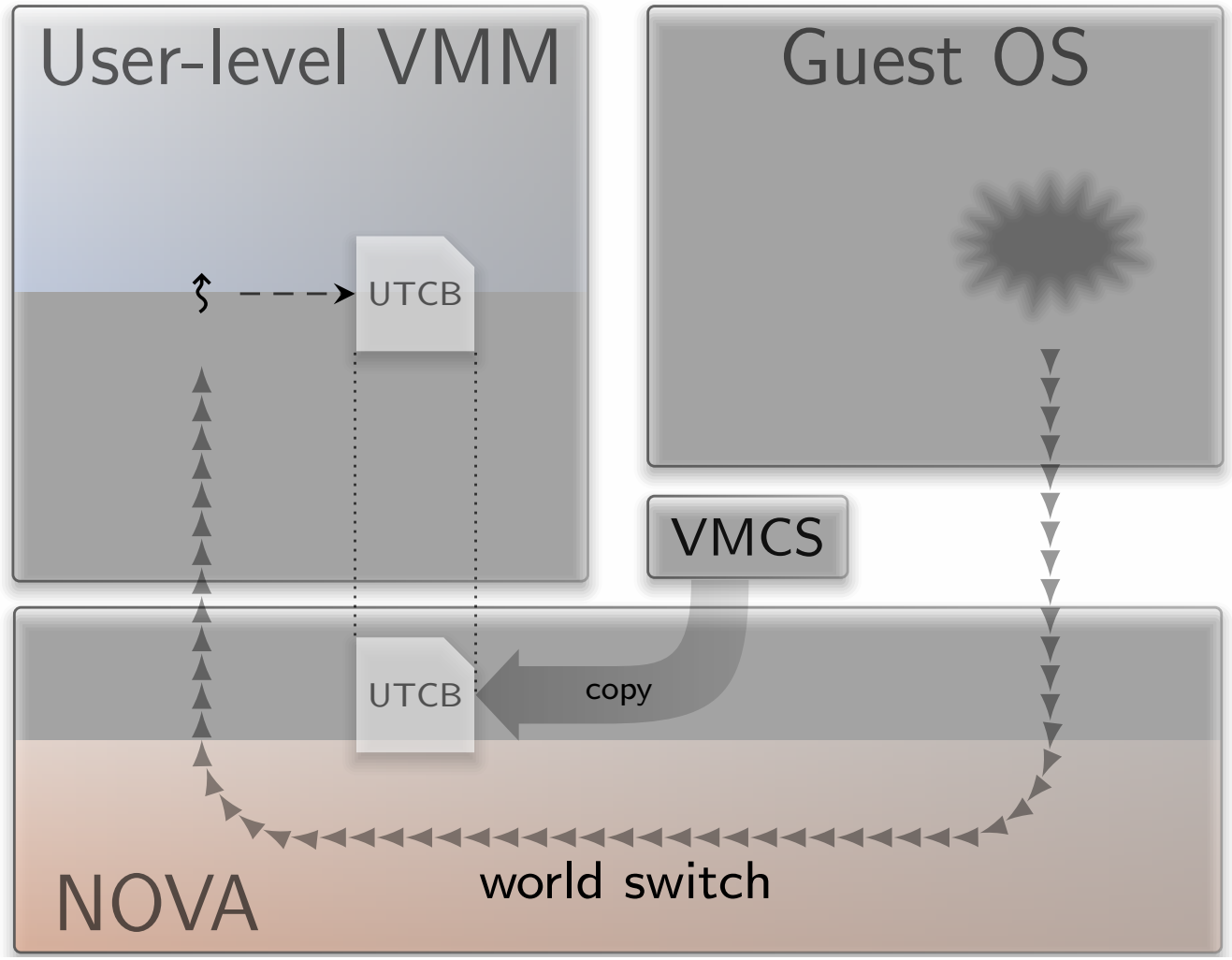


Outline

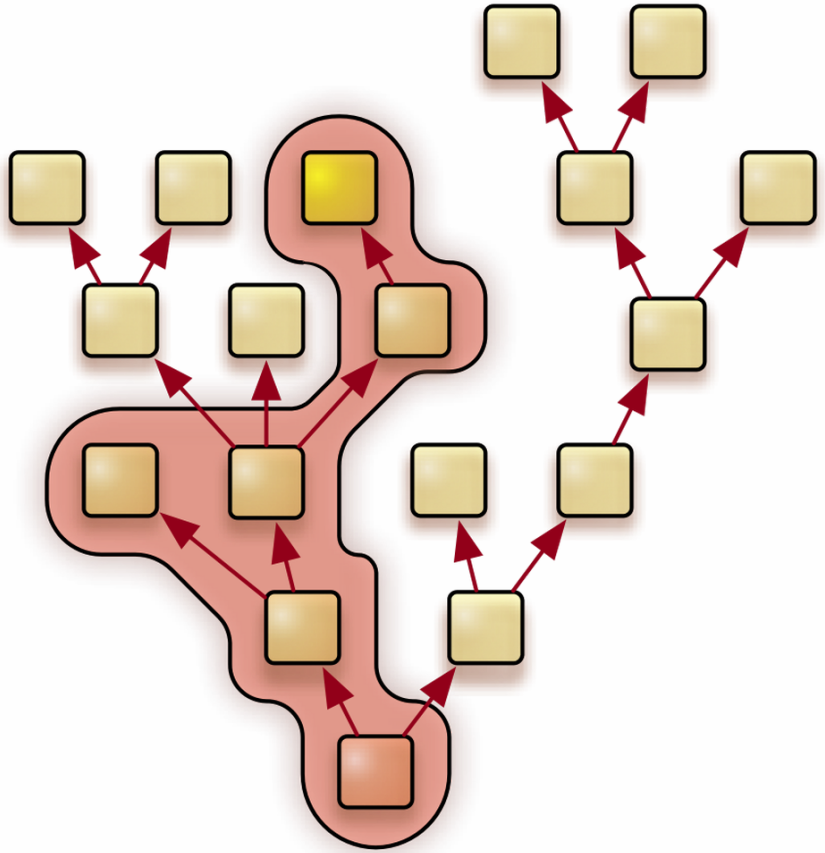
NOVA architecture



Flow of a virtualization event

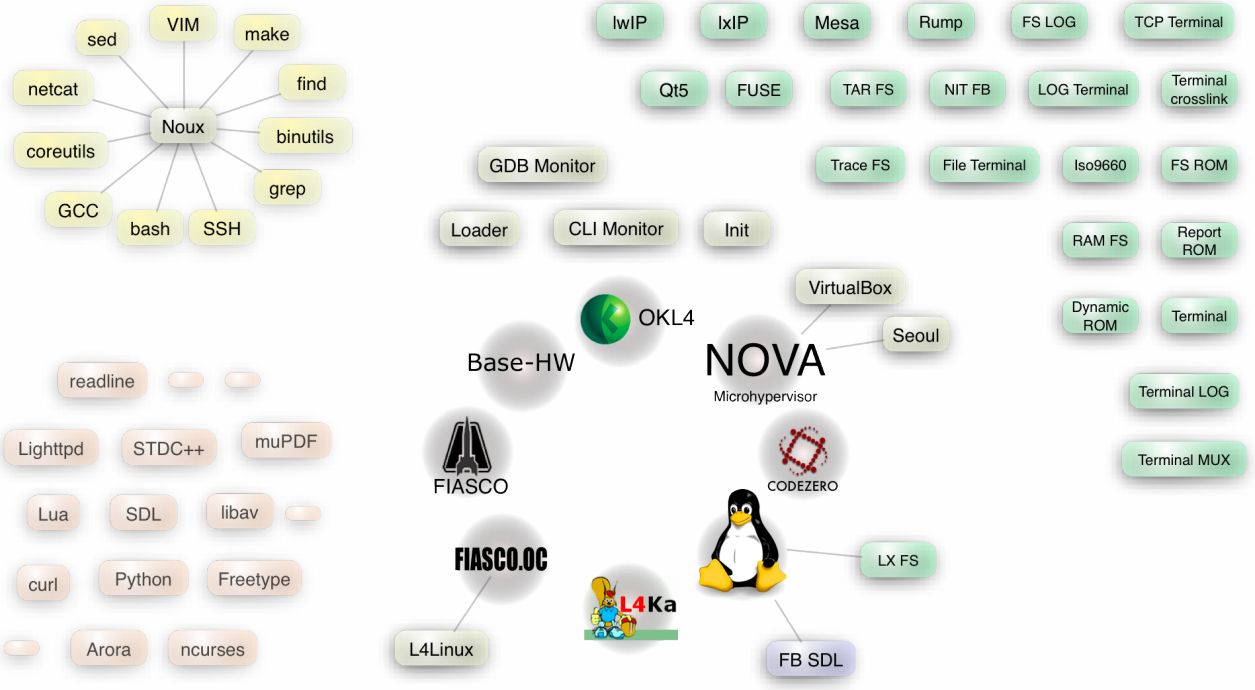


Genode OS architecture

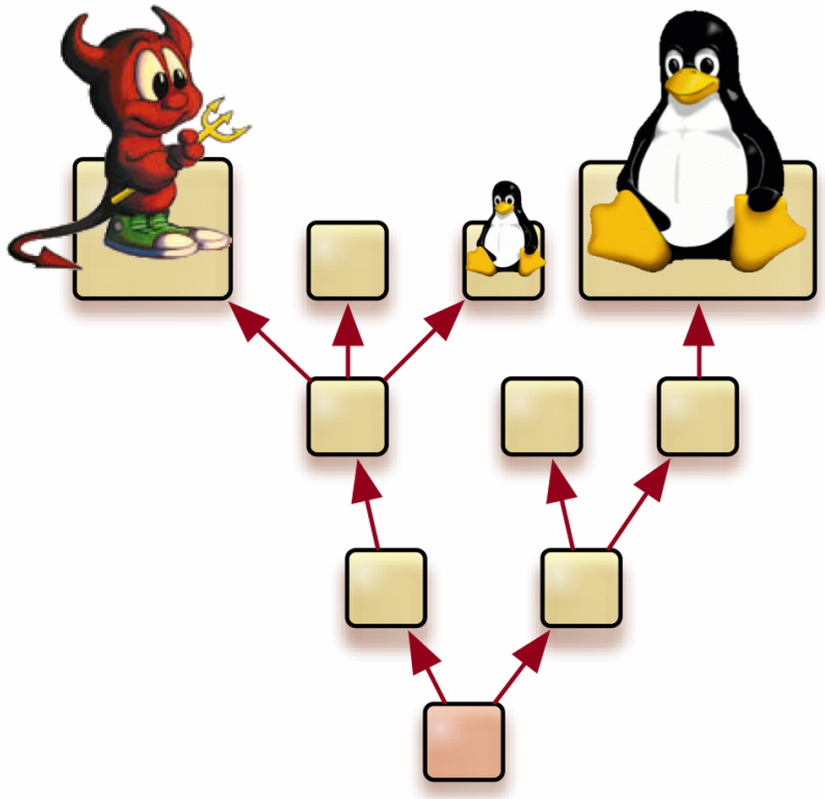


→ Application-specific TCB

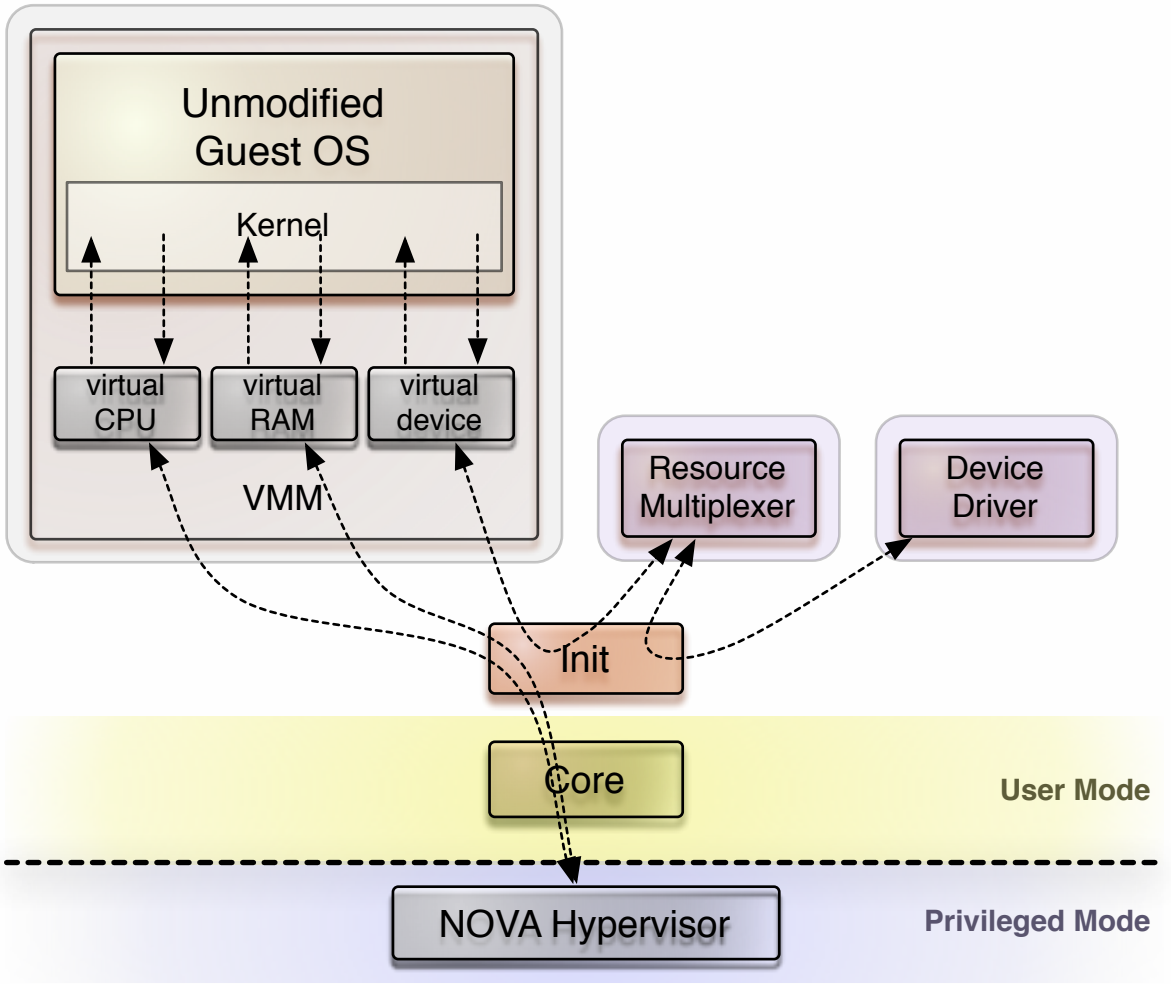
Genode OS framework



Genode combined with virtualization



Seoul VMM on top of Genode/NOVA



Idea

Device models and features of VirtualBox

+

Security of the Genode/NOVA architecture

Outline

Identify the interesting parts

Entire VirtualBox code base

> 4 million lines of code (sloccount)

Narrowed to the interesting parts

> 2 million lines of code

```
src/VBox/VMM
src/VBox/Main
src/VBox/Runtime
src/VBox/Devices
src/VBox/Storage
src/VBox/GuestHost
src/VBox/Disassembler
src/VBox/HostServices
src/recompiler
src/libs/liblzf-3.4
src/libs/liblzf-3.4/cs
src/libs/libxml2-2.6.31
src/libs/zlib-1.2.6
include/VBox
include/iprt
```

Porting the VirtualBox Runtime to Genode

- Facilitate Genode's existing infrastructure
  - 3rd-party software management tools
  - FreeBSD libc
  - Standard C++ library
  - POSIX threads

Porting the VirtualBox Runtime to Genode

- Facilitate Genode's existing infrastructure
  - 3rd-party software management tools
  - FreeBSD libc
  - Standard C++ library
  - POSIX threads

$\rightarrow$ Most parts of the POSIX runtime could be reused

VM process initialization

Enable subsystems one by one

VM process initialization

Enable subsystems one by one

- Guest memory (accessed by recompiler and device models)
  RAM, MMIO

VM process initialization

Enable subsystems one by one

- Guest memory (accessed by recompiler and device models)
  RAM, MMIO
- I/O-port handling

VM process initialization

Enable subsystems one by one

- Guest memory (accessed by recompiler and device models)
  RAM, MMIO
- I/O-port handling
- PGM, HWACCM, TM

VM process initialization

Enable subsystems one by one

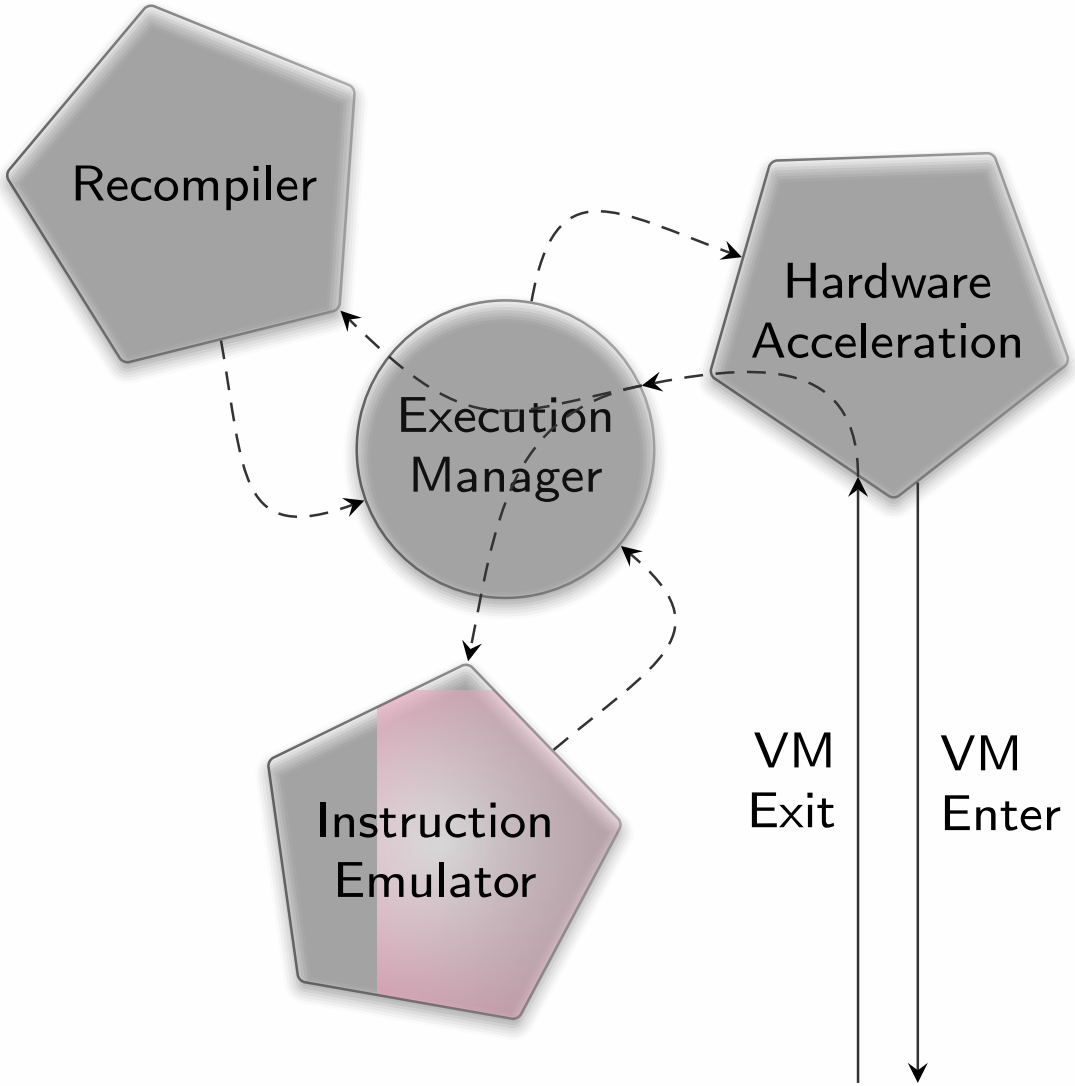
- Guest memory (accessed by recompiler and device models)
  RAM, MMIO
- I/O-port handling
- PGM, HWACCM, TM
- Device models, PDM, BIOS

VM process initialization

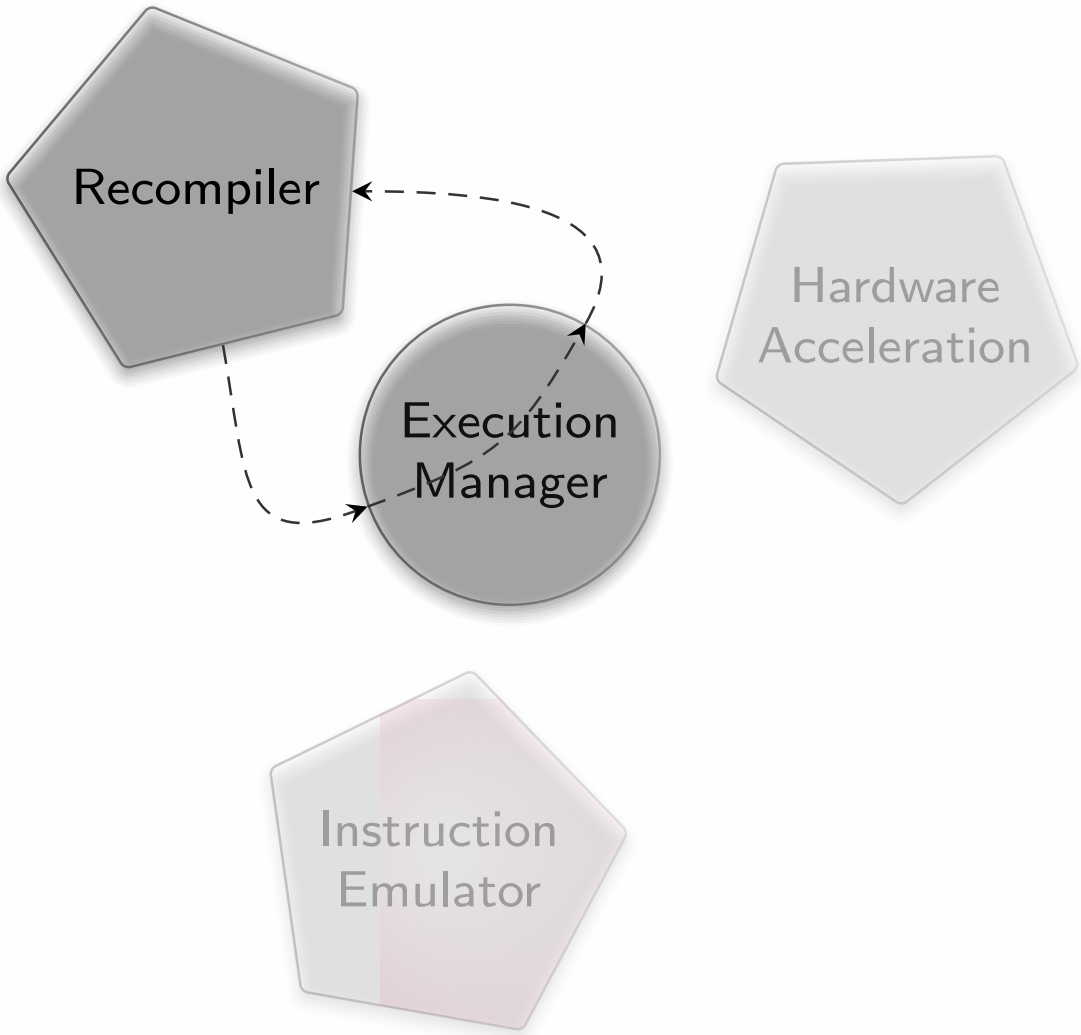
Enable subsystems one by one

- Guest memory (accessed by recompiler and device models)
  RAM, MMIO
- I/O-port handling
- PGM, HWACCM, TM
- Device models, PDM, BIOS
- Host drivers
  - Using the “Basic front end”
  - Reimplement `SDLConsole` interface

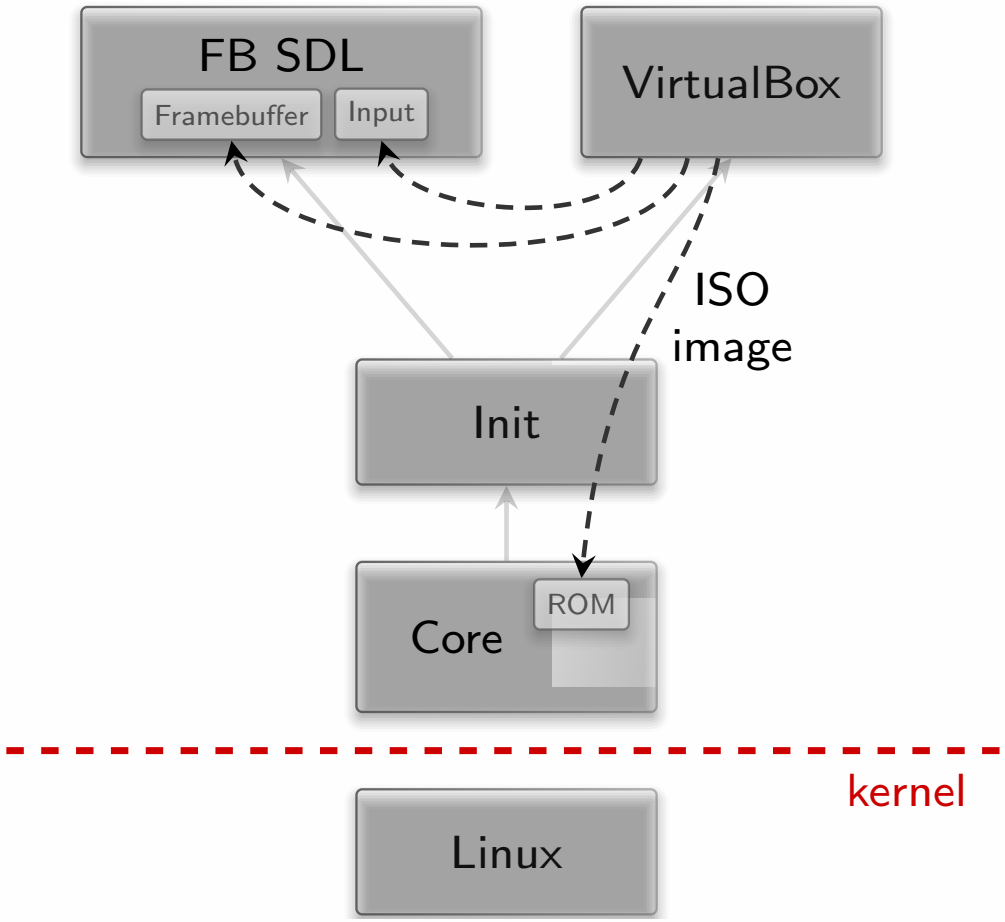
A look inside a VM process



Start with executing the recompiler only



Simple test scenario



Increasing guest complexity

1. Custom-made Genode OS scenarios

Increasing guest complexity

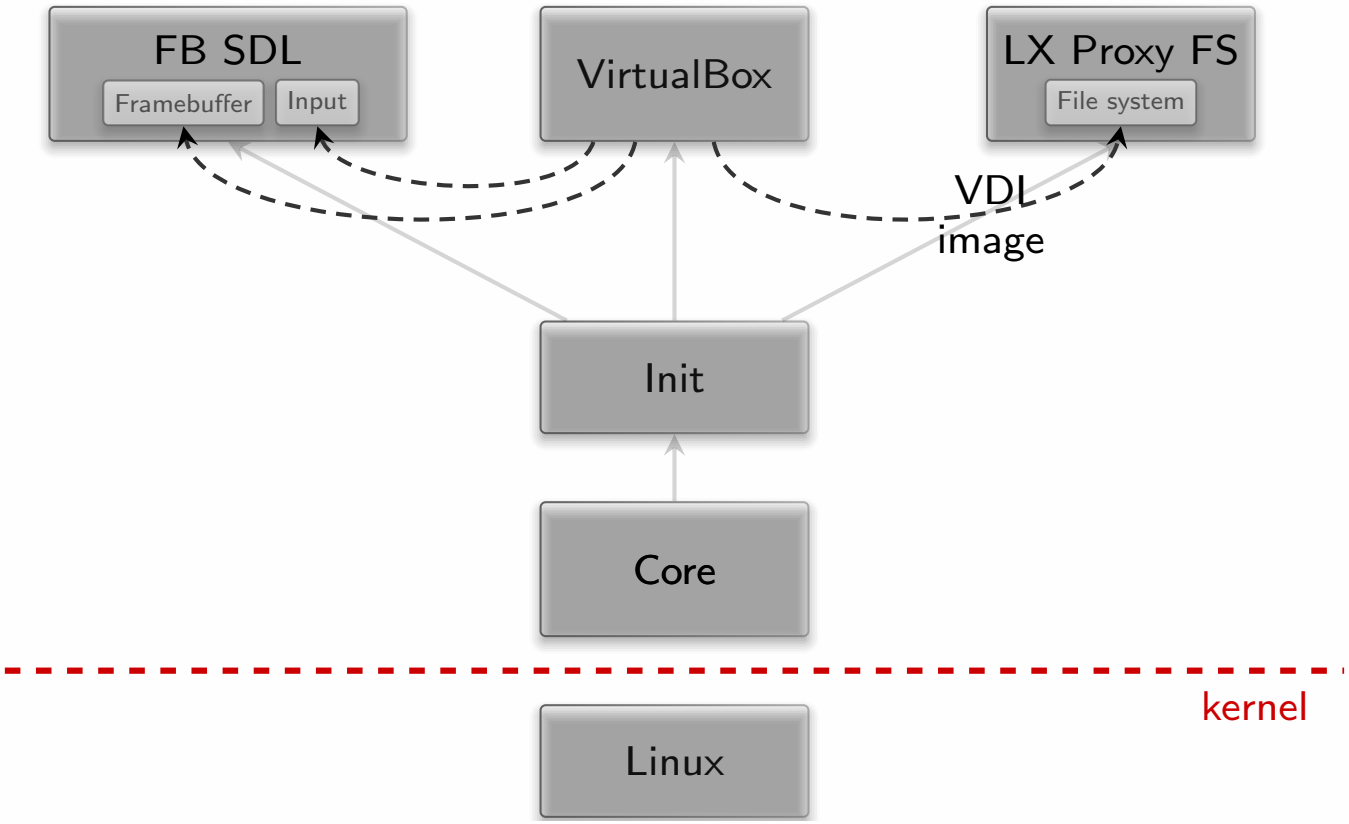
1. Custom-made Genode OS scenarios

2. Small Linux-based images (Tinycore, GRML)

Increasing guest complexity

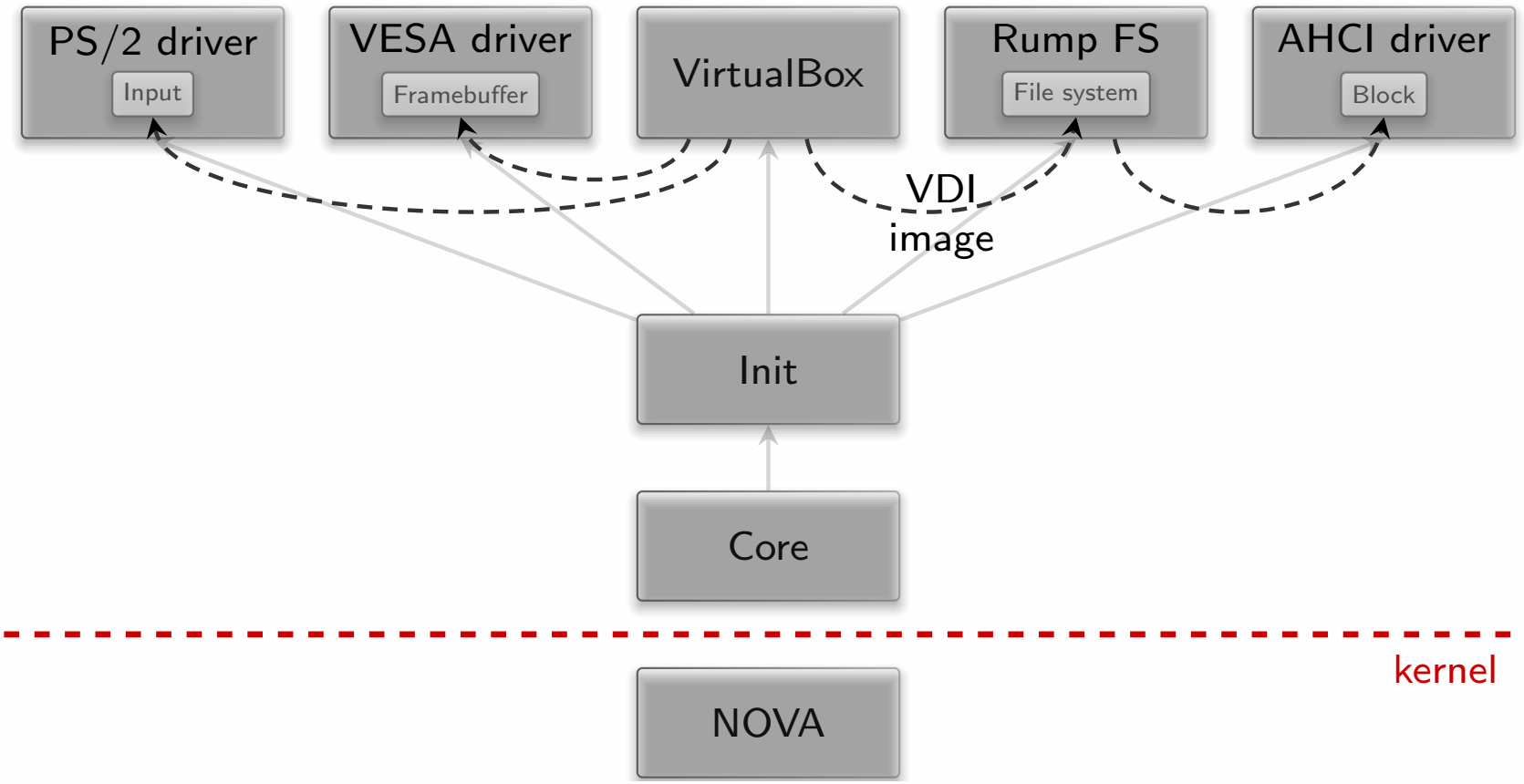
1. Custom-made Genode OS scenarios
2. Small Linux-based images (Tinycore, GRML)
3. Windows XP

Windows XP as a guest

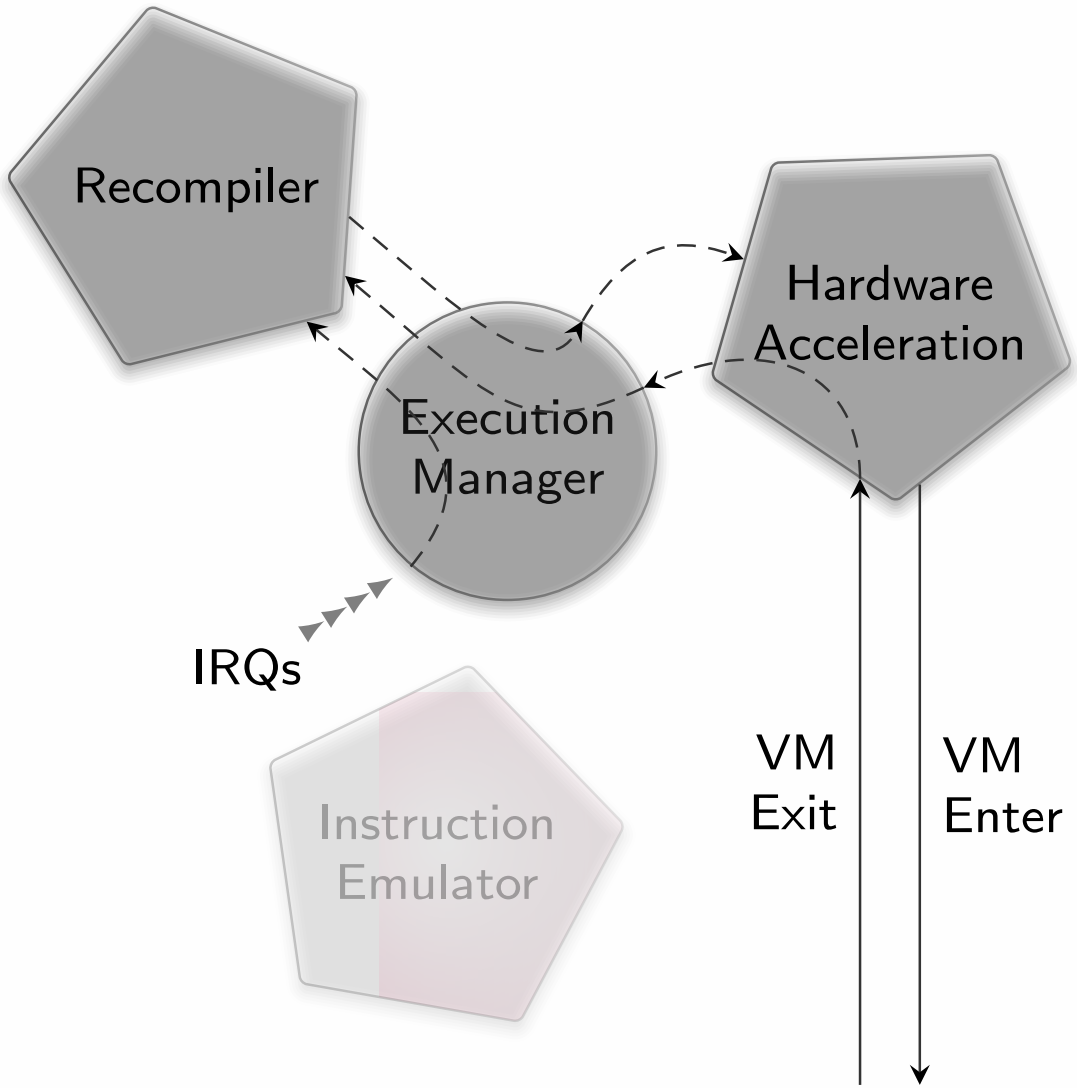


Move scenario to NOVA

Move scenario to NOVA



Entering non-root mode



Entering non-root mode

- VBox VM state $\leftrightarrow$ NOVA UTCB state

Entering non-root mode

- VBox VM state $\leftrightarrow$ NOVA UTCB state
- Virtualization of guest memory
  (EPT faults)

Entering non-root mode

- VBox VM state $\leftrightarrow$ NOVA UTCB state
- Virtualization of guest memory
  (EPT faults)
- Enter VT-x conservatively
  (if protected mode and paging enabled)

Entering non-root mode

- VBox VM state $\leftrightarrow$ NOVA UTCB state
- Virtualization of guest memory
  (EPT faults)
- Enter VT-x conservatively
  (if protected mode and paging enabled)
- Inject IRQs into recompiler

Entering non-root mode

- VBox VM state $\leftrightarrow$ NOVA UTCB state
- Virtualization of guest memory
  (EPT faults)
- Enter VT-x conservatively
  (if protected mode and paging enabled)
- Inject IRQs into recompiler
- Later: IRQ injection via NOVA into VT-X

Adding features

Additional drivers

- Networking

Guest tools

- Shared folders
- Host clock
- Mouse-pointer synchronization

Update to VirtualBox 4.3

- Basic front end no longer supported

- Use of main front end code to NOVA port
  - Custom console implementation
  - Shortcut XPCOM middleware

  $\rightarrow$ Support for using `.vbox` files

Outline

Demo

Windows 7 running in VirtualBox directly on top of NOVA

Adaptation of VirtualBox to Genode/NOVA

Ported code

- 400,000 lines of code (sloccount)

New code

- 6,200 lines (sloccount)
  hm, iommio, ioport, mm, pdm, pgm, sup

Modifications of the original code

- 510 lines added
- 120 lines removed

Current state and outlook

- Usable performance, optimization ongoing

Current state and outlook

- Usable performance, optimization ongoing
- Focused on VT-X, SVM not regularly tested

Current state and outlook

- Usable performance, optimization ongoing
- Focused on VT-X, SVM not regularly tested
- **Reduces TCB complexity to two orders of magnitude**

Current state and outlook

- Usable performance, optimization ongoing
- Focused on VT-X, SVM not regularly tested
- **Reduces TCB complexity to two orders of magnitude**
- Useful for building appliances in high-security computing

Current state and outlook

- Usable performance, optimization ongoing
- Focused on VT-X, SVM not regularly tested
- **Reduces TCB complexity to two orders of magnitude**
- Useful for building appliances in high-security computing
- Stepping stone for using Genode as a general-purpose OS

Outline

War stories

War stories

- Invalid guest state

War stories

- Invalid guest state
- TLB consistency

War stories

- Invalid guest state
- TLB consistency
- Interrupt handling

War stories

- Invalid guest state
- TLB consistency
- Interrupt handling
- Large files in shared folders

Outline

Project Turmvilla

- Use of Genode as our day-to-day OS

Project Turmvilla

- Use of Genode as our day-to-day OS
- VirtualBox as migration path

Project Turmvilla

- Use of Genode as our day-to-day OS
- VirtualBox as migration path
- Reference platform: Lenovo Thinkpad x201

Turmvilla functional requirements

- Wireless networking

Turmvilla functional requirements

- Wireless networking
- Storage (SATA drivers + file system)

Turmvilla functional requirements

- Wireless networking
- Storage (SATA drivers + file system)
- Graphics (driver + GUI stack)

Turmvilla functional requirements

- Wireless networking
- Storage (SATA drivers + file system)
- Graphics (driver + GUI stack)
- User input (PS/2 and USB HID)

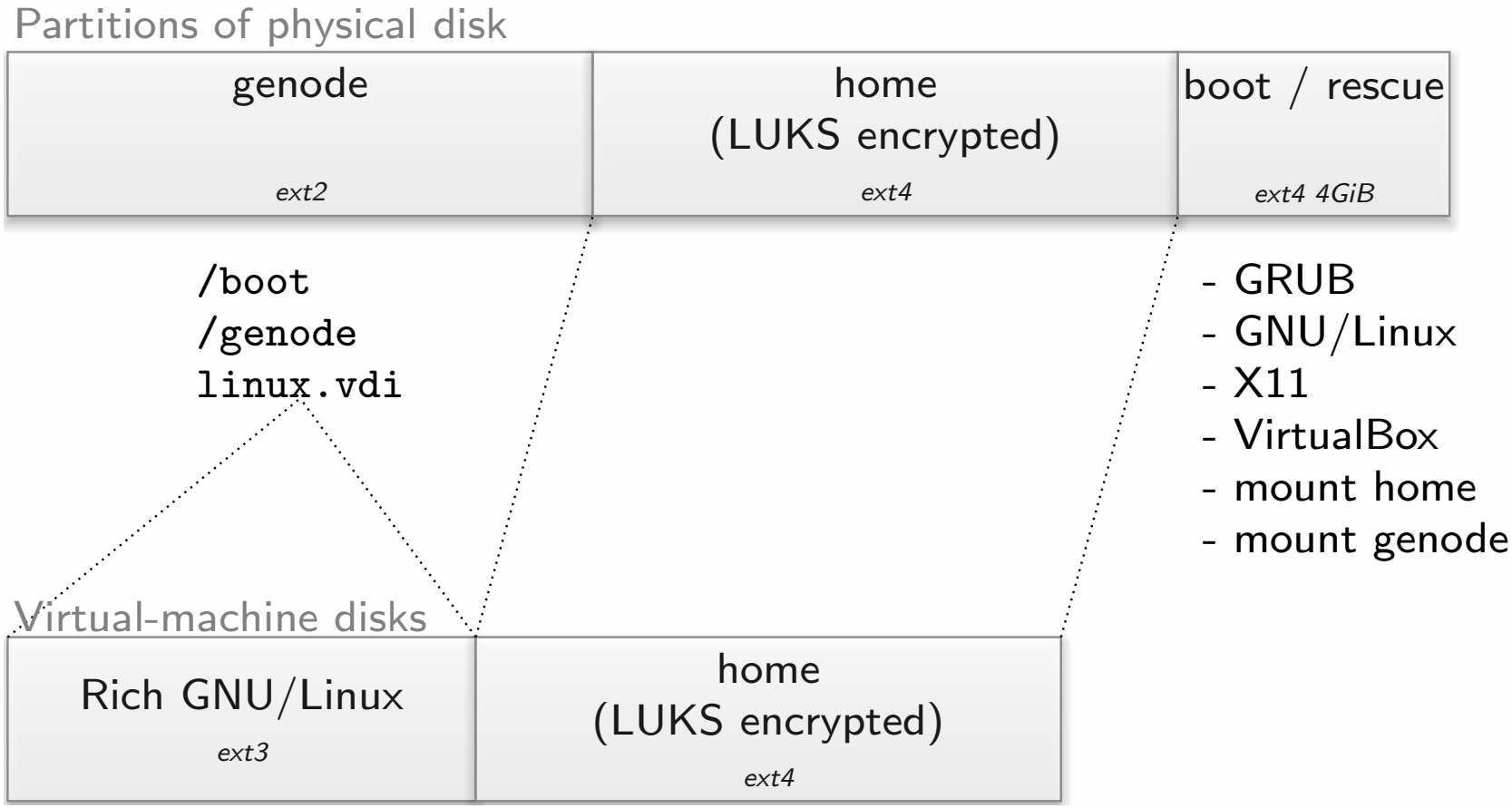
Turmvilla functional requirements

- Wireless networking
- Storage (SATA drivers + file system)
- Graphics (driver + GUI stack)
- User input (PS/2 and USB HID)
- Integration of guest OS and Genode

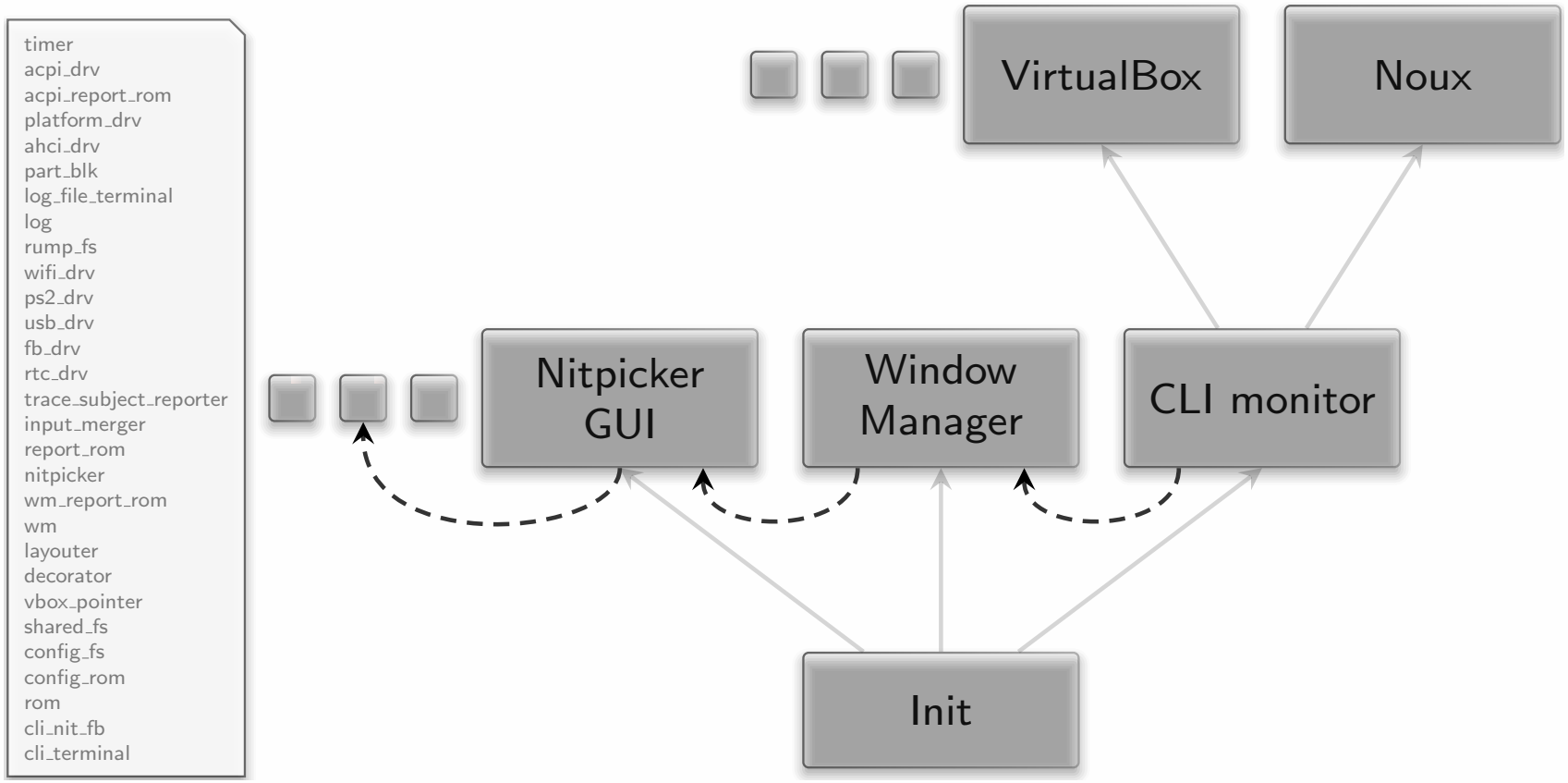
Turmvilla functional requirements

- Wireless networking
- Storage (SATA drivers + file system)
- Graphics (driver + GUI stack)
- User input (PS/2 and USB HID)
- Integration of guest OS and Genode
- A fallback!

Turmvilla dual-boot setup



Turmvilla Genode scenario



Turmvilla state and current focus

Current state:

- My primary OS since the beginning of June
- Team at Genode Labs starts migration

Turmvilla state and current focus

Current state:

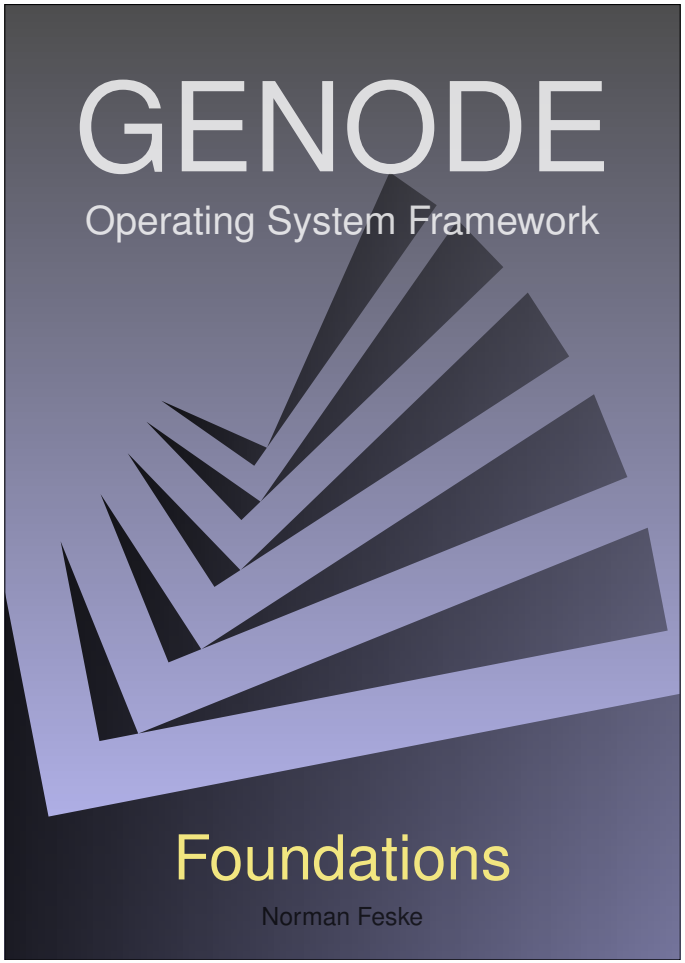
- My primary OS since the beginning of June
- Team at Genode Labs starts migration

Work in progress:

- Tiled and tabbed window manager
- Intel graphics driver
- NOVA kernel-resource management
- Capability-based desktop environment

Outline

The Book “Genode Foundations”



http://genode.org/documentation/genode-foundations-15-05.pdf

Thank you

Genode OS Framework

`http://genode.org`

Genode Labs GmbH

`http://www.genode-labs.com`

Source code at GitHub

`http://github.com/genodelabs/genode`