Genode as Desktop OS

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

Outline

1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

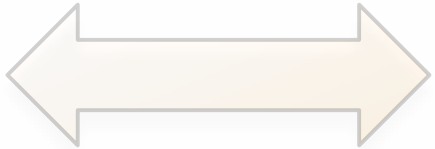
Outline

1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

Universal Truths

Assurance  Scalability

Accountability  Utilization

Security  Ease of use

Problem: Complexity

Today's commodity OSes — Exceedingly complex trusted computing base (TCB)

Problem: Complexity

Today's commodity OSes — Exceedingly complex trusted computing base (TCB)

TCB of an application on Linux:

- Kernel + loaded kernel modules
- Daemons
- X Server + window manager
- Desktop environment
- All running processes of the user

Problem: Complexity

Today's commodity OSes — Exceedingly complex trusted computing base (TCB)

TCB of an application on Linux:

- Kernel + loaded kernel modules
- Daemons
- X Server + window manager
- Desktop environment
- All running processes of the user

→ **User credentials are exposed to millions of lines of code**

Problem: Complexity (II)

Implications:

- High likelihood for bugs (need for frequent security updates)

Problem: Complexity (II)

Implications:

- High likelihood for bugs (need for frequent security updates)
- Huge attack surface for directed attacks

Problem: Complexity (II)

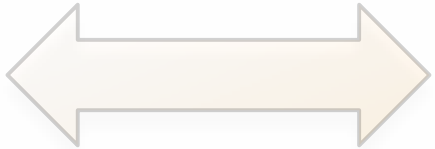
Implications:

- High likelihood for bugs (need for frequent security updates)
- Huge attack surface for directed attacks
- Zero-day exploits

Universal Truths

Assurance  Scalability

Accountability  Utilization

Security  Ease of use

Problem: Resource management

- Pretension of unlimited resources
- Lack of accounting

Problem: Resource management

- Pretension of unlimited resources
- Lack of accounting

  $\rightarrow$ Largely indeterministic behavior

Problem: Resource management

- Pretension of unlimited resources
- Lack of accounting

  $\rightarrow$ Largely indeterministic behavior

  $\rightarrow$ Need for complex heuristics, schedulers

Problem: Resource management

- Pretension of unlimited resources
- Lack of accounting

  → Largely indeterministic behavior

  → Need for complex heuristics, schedulers

```
Jul 24 12:58:30 neo kernel: [72454.482259] cupsd invoked oom-killer: gfp_mask=0x201da, order=0, oom_adj=0, oom_score_adj=0
Jul 24 12:58:30 neo kernel: [72454.482264] cupsd cpuset=/ mems_allowed=0
Jul 24 12:58:30 neo kernel: [72454.482268] Pid: 1416, comm: cupsd Tainted: G        WC  3.0.0-22-generic #36-Ubuntu
Jul 24 12:58:30 neo kernel: [72454.482270] Call Trace:
Jul 24 12:58:30 neo kernel: [72454.482279]  [<ffffffff810b5c8d>] ? cpuset_print_task_mems_allowed+0x9d/0xb0
Jul 24 12:58:30 neo kernel: [72454.482286]  [<ffffffff8110df91>] dump_header+0x91/0xe0
Jul 24 12:58:30 neo kernel: [72454.482289]  [<ffffffff8110e2f5>] oom_kill_process+0x85/0xb0
Jul 24 12:58:30 neo kernel: [72454.482293]  [<ffffffff8110e69a>] out_of_memory+0xfa/0x250
Jul 24 12:58:30 neo kernel: [72454.482298]  [<ffffffff81113f3f>] __alloc_pages_nodemask+0x80f/0x820
Jul 24 12:58:30 neo kernel: [72454.482304]  [<ffffffff8120a1a0>] ? noalloc_get_block_write+0x30/0x30
Jul 24 12:58:30 neo kernel: [72454.482311]  [<ffffffff8114a0a3>] alloc_pages_current+0xa3/0x110
Jul 24 12:58:30 neo kernel: [72454.482314]  [<ffffffff8110ab4f>] __page_cache_alloc+0x8f/0xa0
Jul 24 12:58:30 neo kernel: [72454.482318]  [<ffffffff8110afae>] ? find_get_page+0x1e/0x90
Jul 24 12:58:30 neo kernel: [72454.482321]  [<ffffffff8110cea4>] filemap_fault+0x234/0x3e0
Jul 24 12:58:30 neo kernel: [72454.482326]  [<ffffffff8115f07b>] ? mem_cgroup_update_page_stat+0x2b/0x110
Jul 24 12:58:30 neo kernel: [72454.482330]  [<ffffffff8112caf4>] __do_fault+0x54/0x510
Jul 24 12:58:30 neo kernel: [72454.482334]  [<ffffffff8113021a>] handle_pte_fault+0xfa/0x210
Jul 24 12:58:30 neo kernel: [72454.482337]  [<ffffffff811306e8>] handle_mm_fault+0x1f8/0x350
Jul 24 12:58:30 neo kernel: [72454.482344]  [<ffffffff815f8913>] do_page_fault+0x153/0x530
Jul 24 12:58:30 neo kernel: [72454.482350]  [<ffffffff81011069>] ? read_tsc+0x9/0x20
Jul 24 12:58:30 neo kernel: [72454.482355]  [<ffffffff8108cd2d>] ? ktime_get_ts+0xad/0xe0
Jul 24 12:58:30 neo kernel: [72454.482361]  [<ffffffff8117bb6a>] ? poll_select_set_timeout+0x7a/0x90
Jul 24 12:58:30 neo kernel: [72454.482365]  [<ffffffff815f5615>] page_fault+0x25/0x30
Jul 24 12:58:30 neo kernel: [72454.493363] Out of memory: Kill process 22727 (oom) score 691 or sacrifice child
Jul 24 12:58:30 neo kernel: [72454.493367] Killed process 22727 (oom) total-vm:2702616kB, anon-rss:2701332kB, file-rss:172kB
```

Universal Truths

Assurance  Scalability

Accountability  Utilization

Security  Ease of use

Key technologies

- Microkernels
- Componentization, kernelization
- Capability-based security
- Virtualization

Key technologies

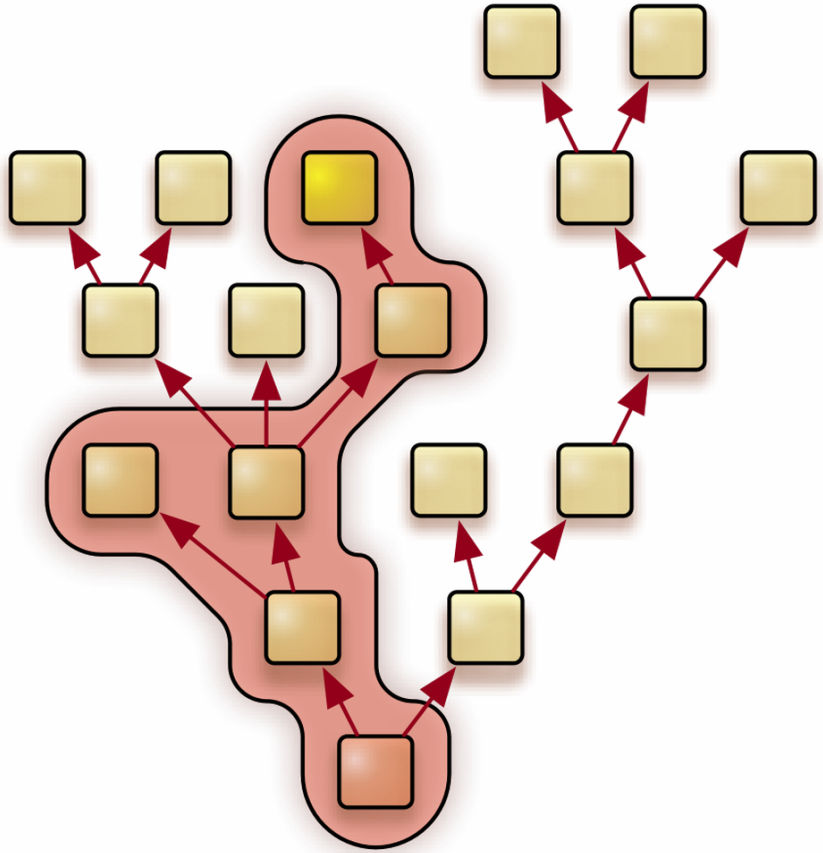
- Microkernels
- Componentization, kernelization
- Capability-based security
- Virtualization

...but how to compose those?

Outline

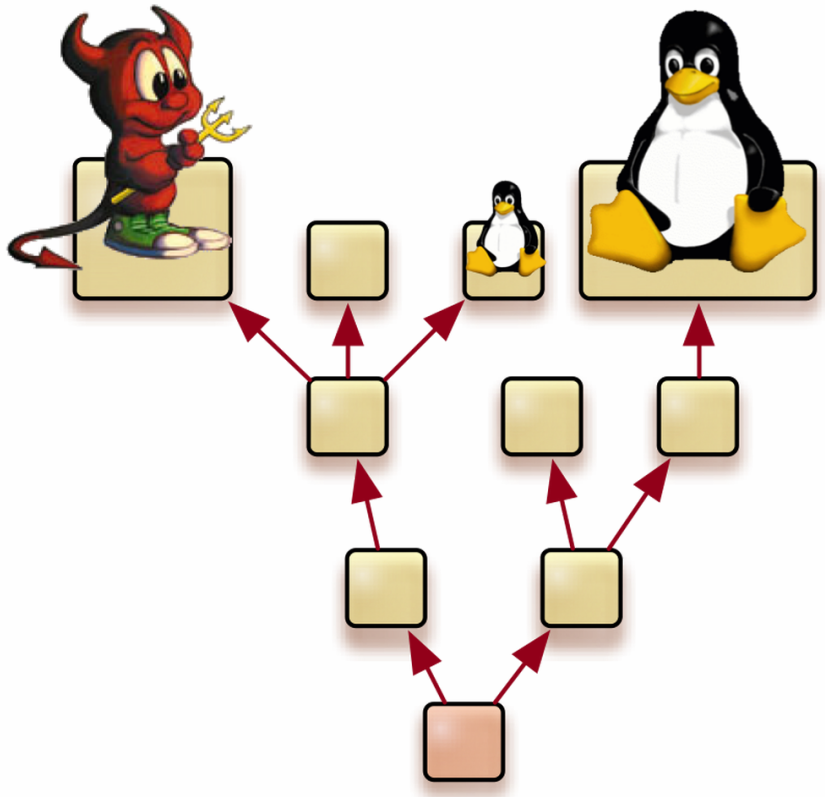
1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

Idea



→ Application-specific TCB

Combined with virtualization



Object capabilities

Delegation of authority between components

Object capabilities

Delegation of authority between components

- Each component lives in a virtual environment

Object capabilities

Delegation of authority between components

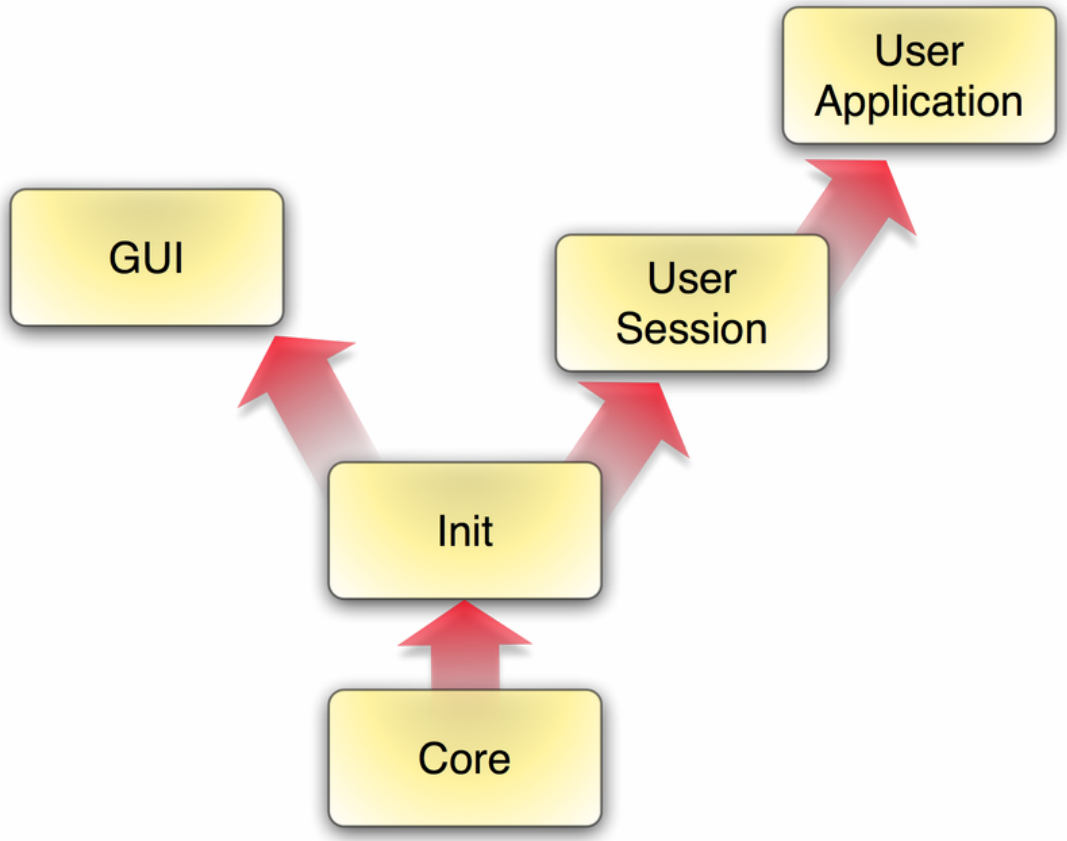
- Each component lives in a virtual environment
- A component that possesses a *capability* can
  - Use it *(invoke)*

Object capabilities

Delegation of authority between components

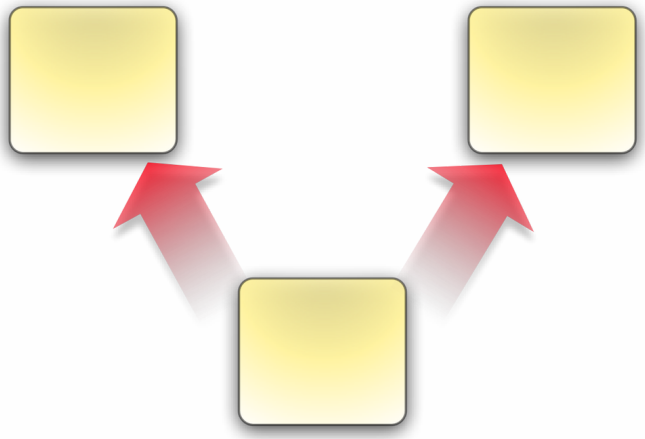
- Each component lives in a virtual environment
- A component that possesses a *capability* can
  - Use it *(invoke)*
  - Delegate it to acquainted components

Recursive system structure



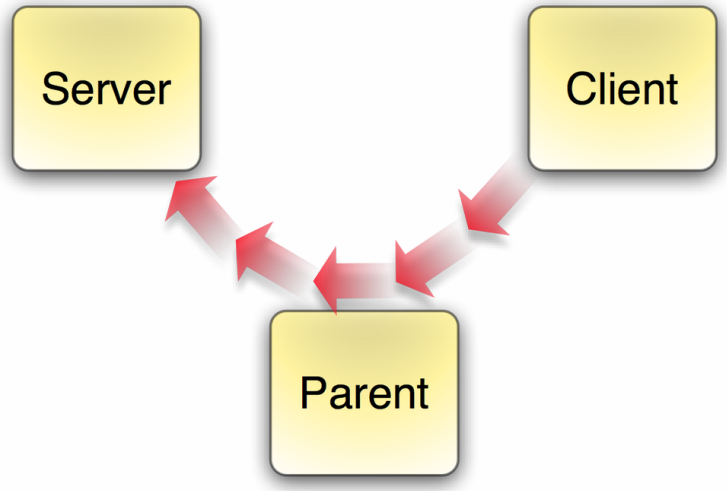
Resource management

Explicit assignment of physical resources to components



Resource management (II)

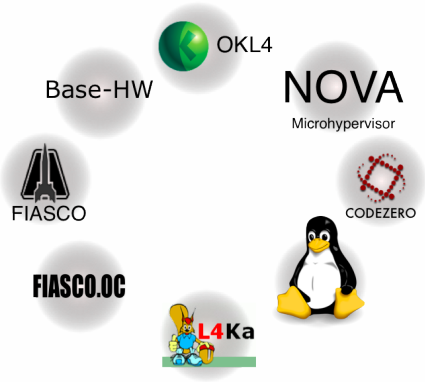
Resources can be attached to sessions



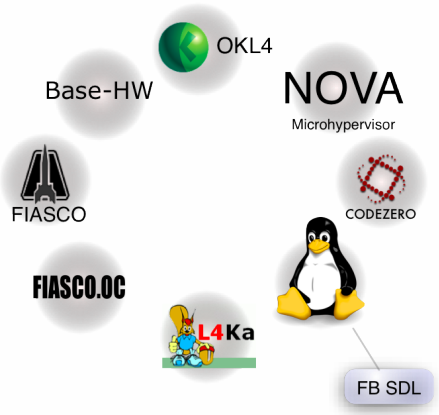
Outline

1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

Components


OKL4
Base-HW
NOVA
Microhypervisor
FIASCO
CODEZERO
FIASCO.OC
L4Ka

Components

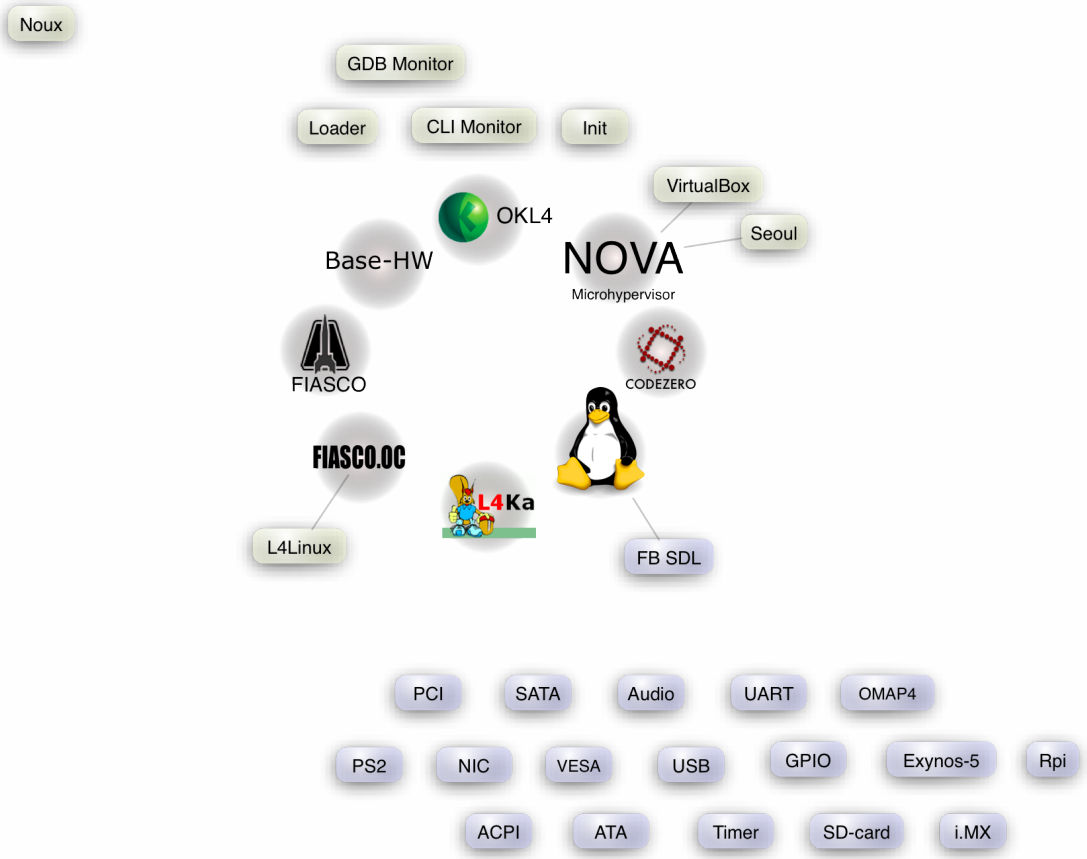


PCI SATA Audio UART OMAP4

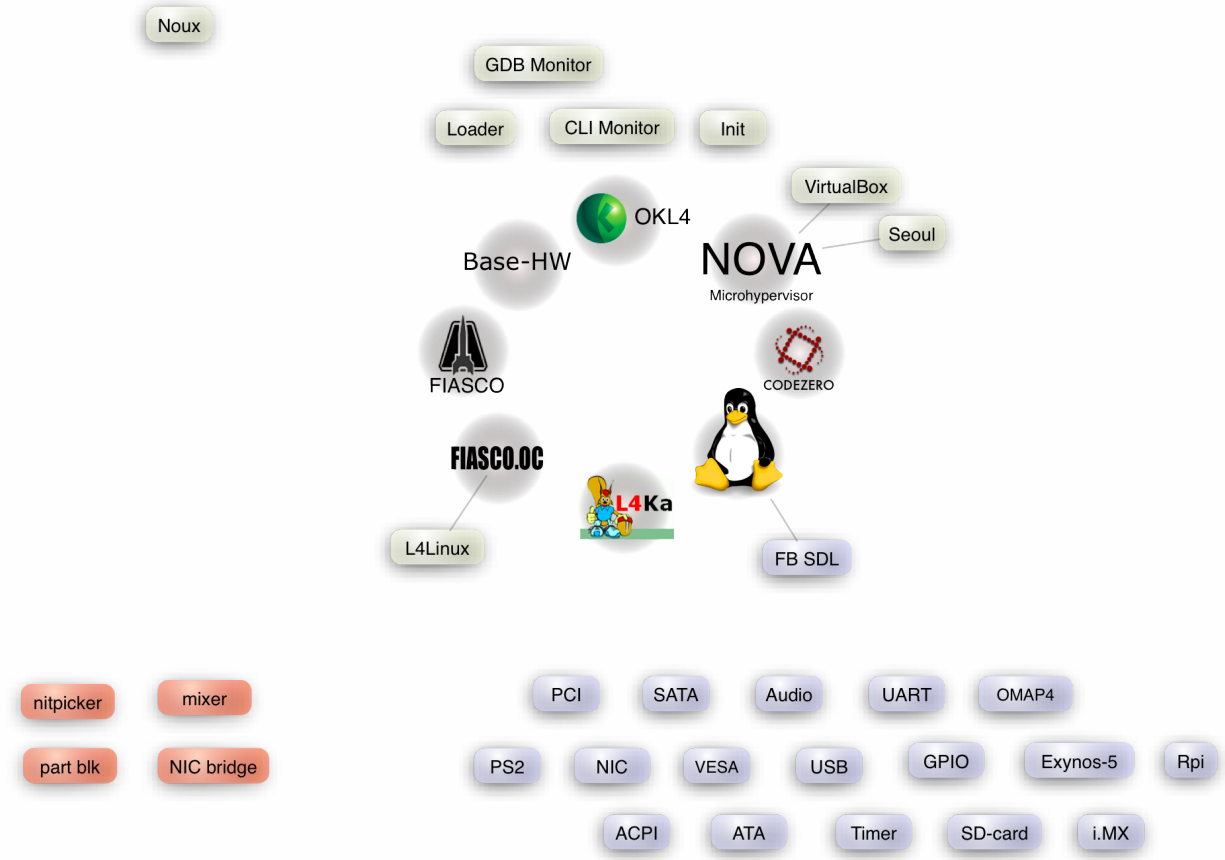
PS2 NIC VESA USB GPIO Exynos-5 Rpi

ACPI ATA Timer SD-card i.MX

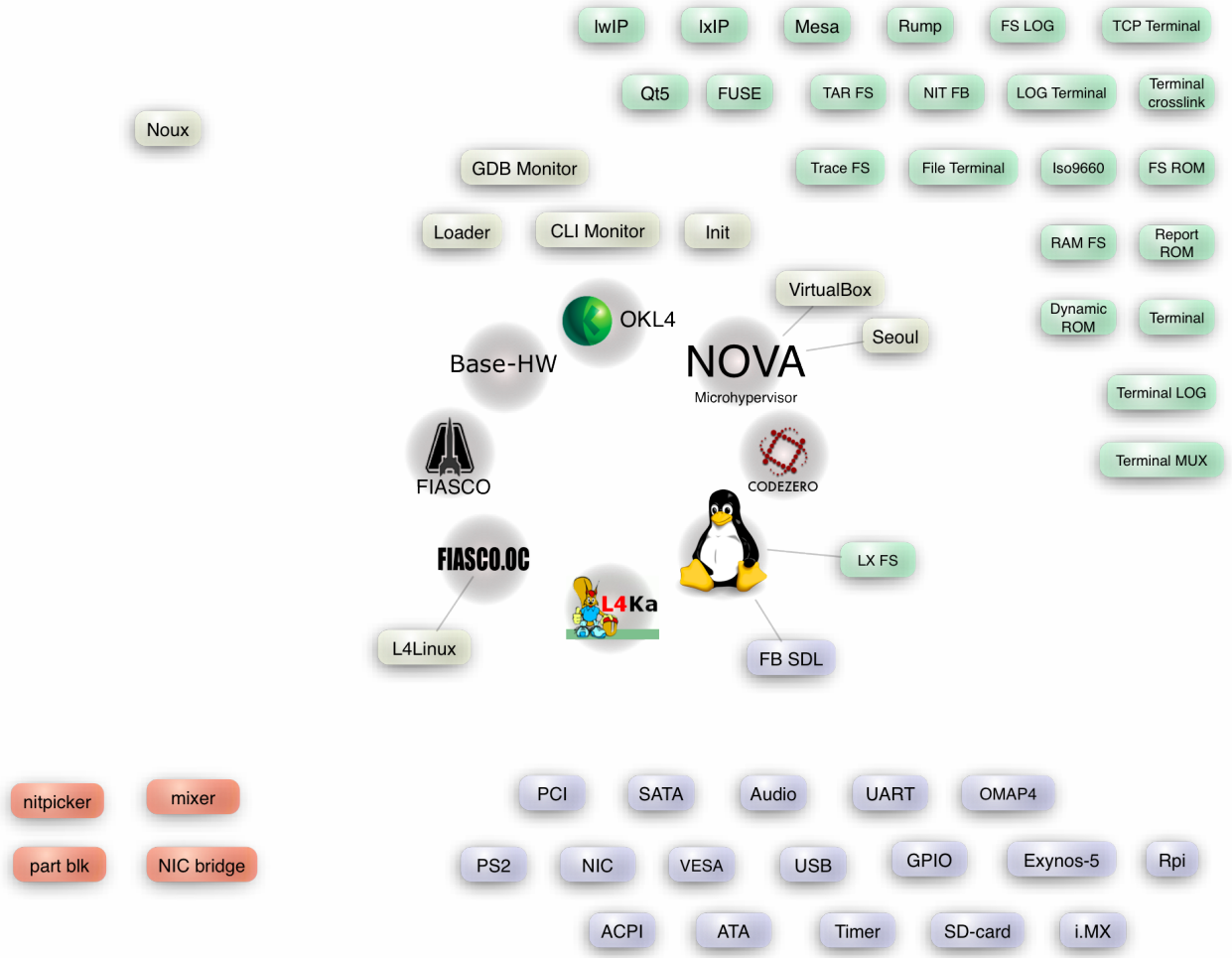
Components


Noux
GDB Monitor
Loader
CLI Monitor
Init
VirtualBox
Seoul
OKL4
Base-HW
NOVA
Microhypervisor
FIASCO
CODEZERO
FIASCO.OC
L4Ka
L4Linux
FB SDL
PCI
SATA
Audio
UART
OMAP4
PS2
NIC
VESA
USB
GPIO
Exynos-5
Rpi
ACPI
ATA
Timer
SD-card
i.MX

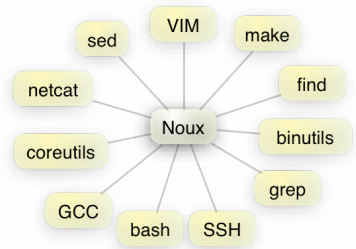
Components

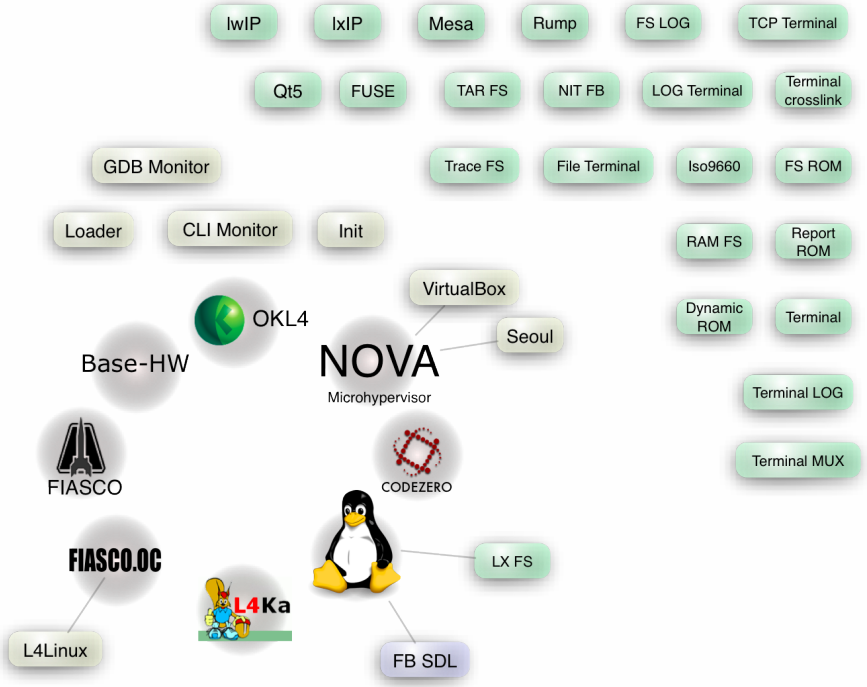

Noux
GDB Monitor
Loader
CLI Monitor
Init
VirtualBox
Seoul
OKL4
Base-HW
NOVA
Microhypervisor
FIASCO
CODEZERO
FIASCO.OC
L4Ka
L4Linux
FB SDL
nitpicker
mixer
part blk
NIC bridge
PCI
SATA
Audio
UART
OMAP4
PS2
NIC
VESA
USB
GPIO
Exynos-5
Rpi
ACPI
ATA
Timer
SD-card
i.MX

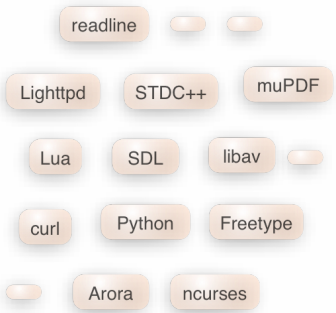
Components


lwIP
lxIP
Mesa
Rump
FS LOG
TCP Terminal
Qt5
FUSE
TAR FS
NIT FB
LOG Terminal
Terminal crosslink
Noux
GDB Monitor
Trace FS
File Terminal
Iso9660
FS ROM
Loader
CLI Monitor
Init
RAM FS
Report ROM
VirtualBox
OKL4
Dynamic ROM
Terminal
Seoul
Base-HW
NOVA
Microhypervisor
Terminal LOG
Terminal MUX
FIASCO
CODEZERO
FIASCO.OC
LX FS
L4Ka
L4Linux
FB SDL
nitpicker
mixer
PCI
SATA
Audio
UART
OMAP4
part blk
NIC bridge
PS2
NIC
VESA
USB
GPIO
Exynos-5
Rpi
ACPI
ATA
Timer
SD-card
i.MX

Components


VIM
sed
make
netcat
find
Noux
binutils
coreutils
grep
GCC
bash
SSH


lwIP
lxIP
Mesa
Rump
FS LOG
TCP Terminal
Qt5
FUSE
TAR FS
NIT FB
LOG Terminal
Terminal crosslink
GDB Monitor
Trace FS
File Terminal
Iso9660
FS ROM
Loader
CLI Monitor
Init
RAM FS
Report ROM
VirtualBox
OKL4
Dynamic ROM
Terminal
Seoul
Base-HW
NOVA
Microhypervisor
Terminal LOG
Terminal MUX
FIASCO
CODEZERO
FIASCO.OC
LX FS
L4Ka
L4Linux
FB SDL


readline
Lighttpd
STDC++
muPDF
Lua
SDL
libav
curl
Python
Freetype
Arora
ncurses

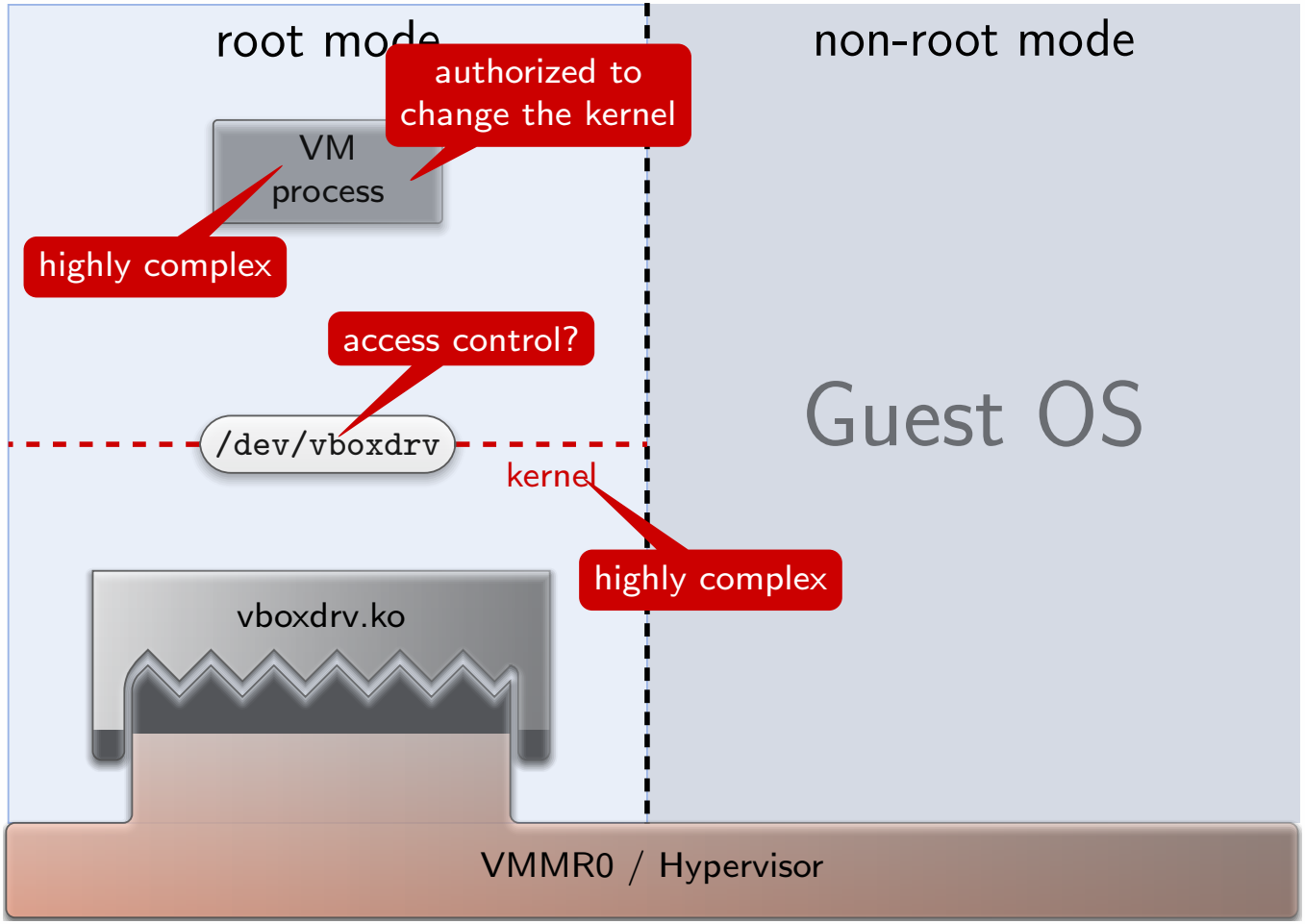

nitpicker
mixer
part blk
NIC bridge


PCI
SATA
Audio
UART
OMAP4
PS2
NIC
VESA
USB
GPIO
Exynos-5
Rpi
ACPI
ATA
Timer
SD-card
i.MX

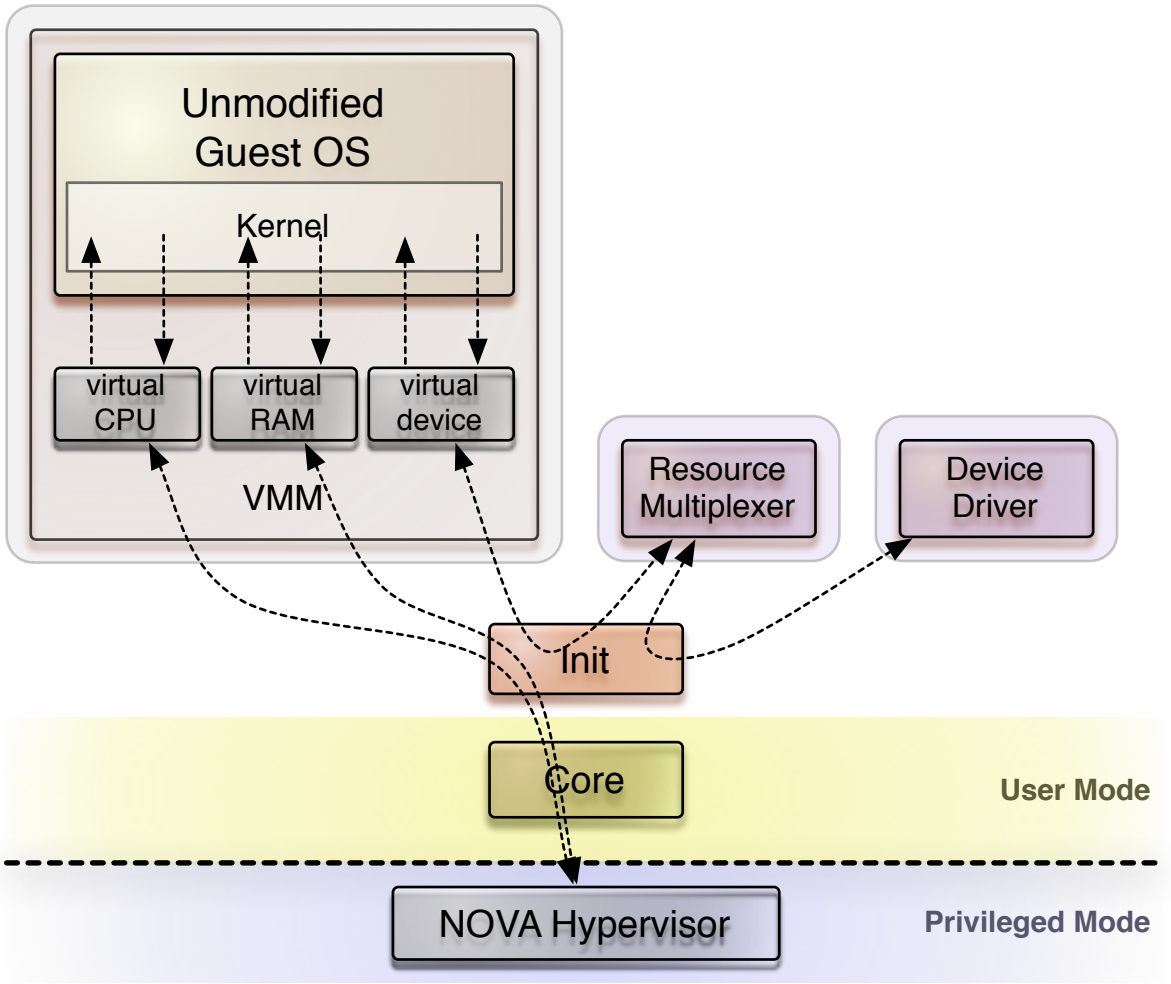
Outline

1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

Faithful virtualization (traditional)

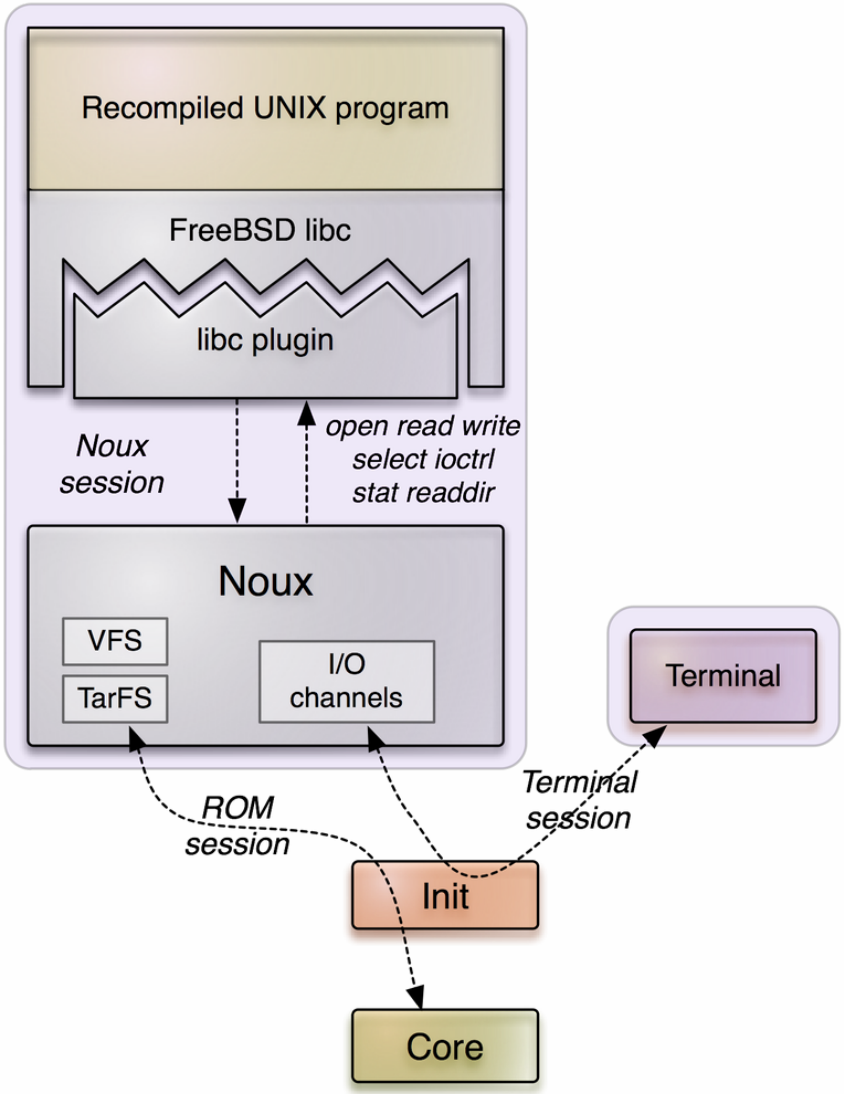


VirtualBox as Genode subsystem

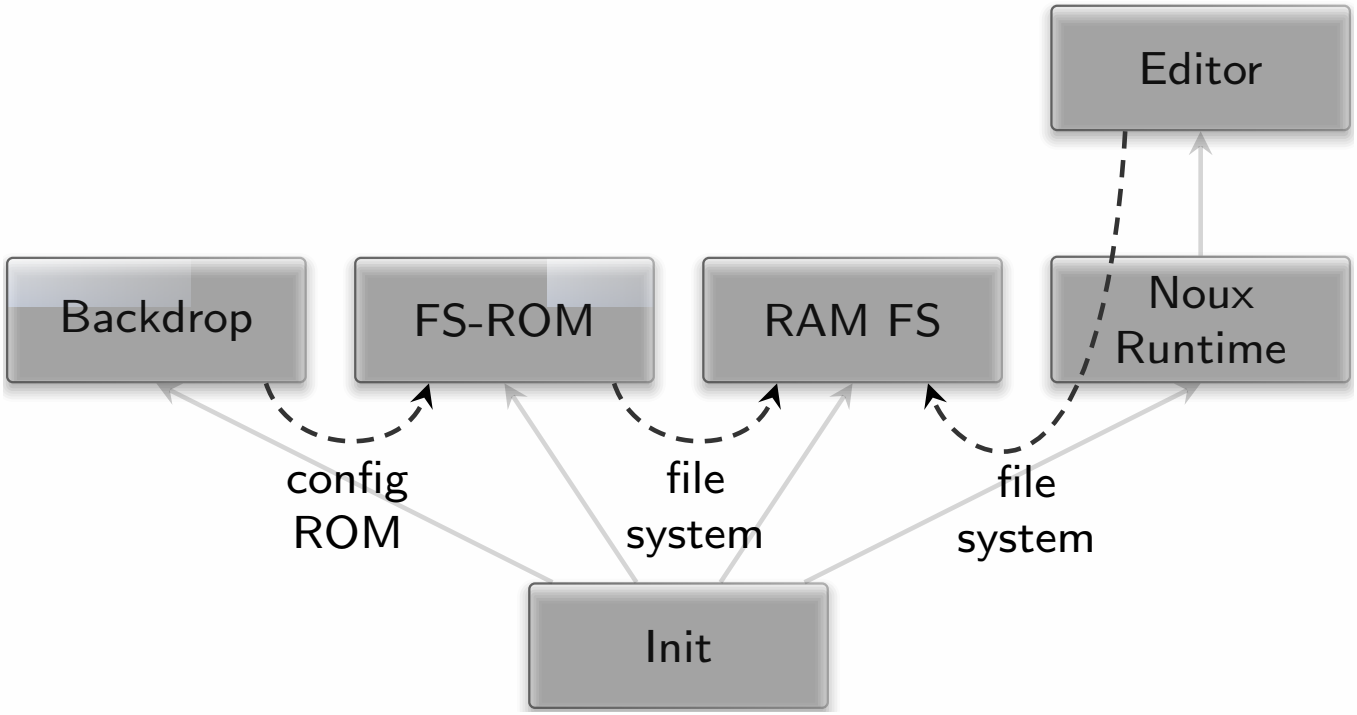


OS-level virtualization

OS-level virtualization

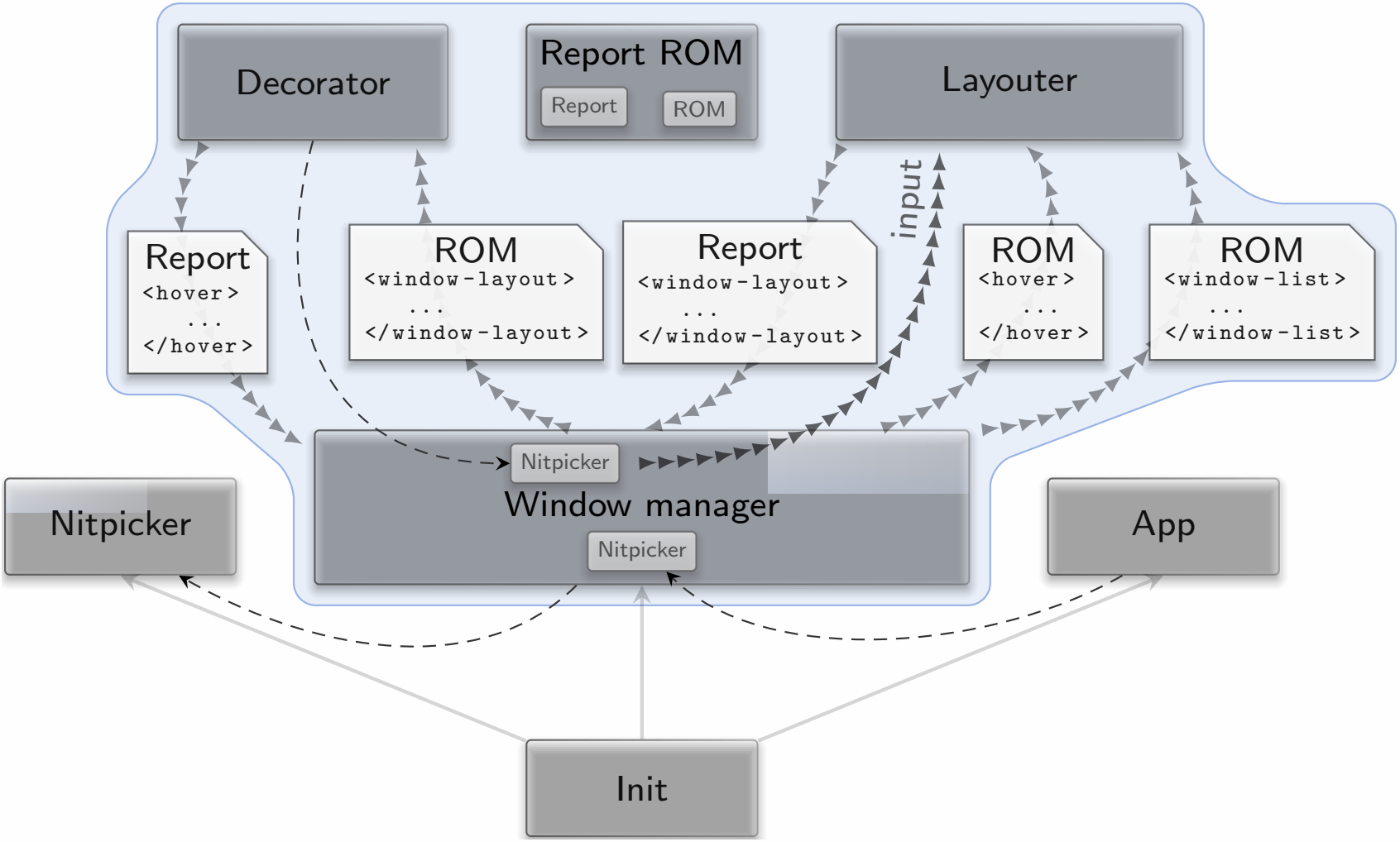


OS-level virtualization (example)

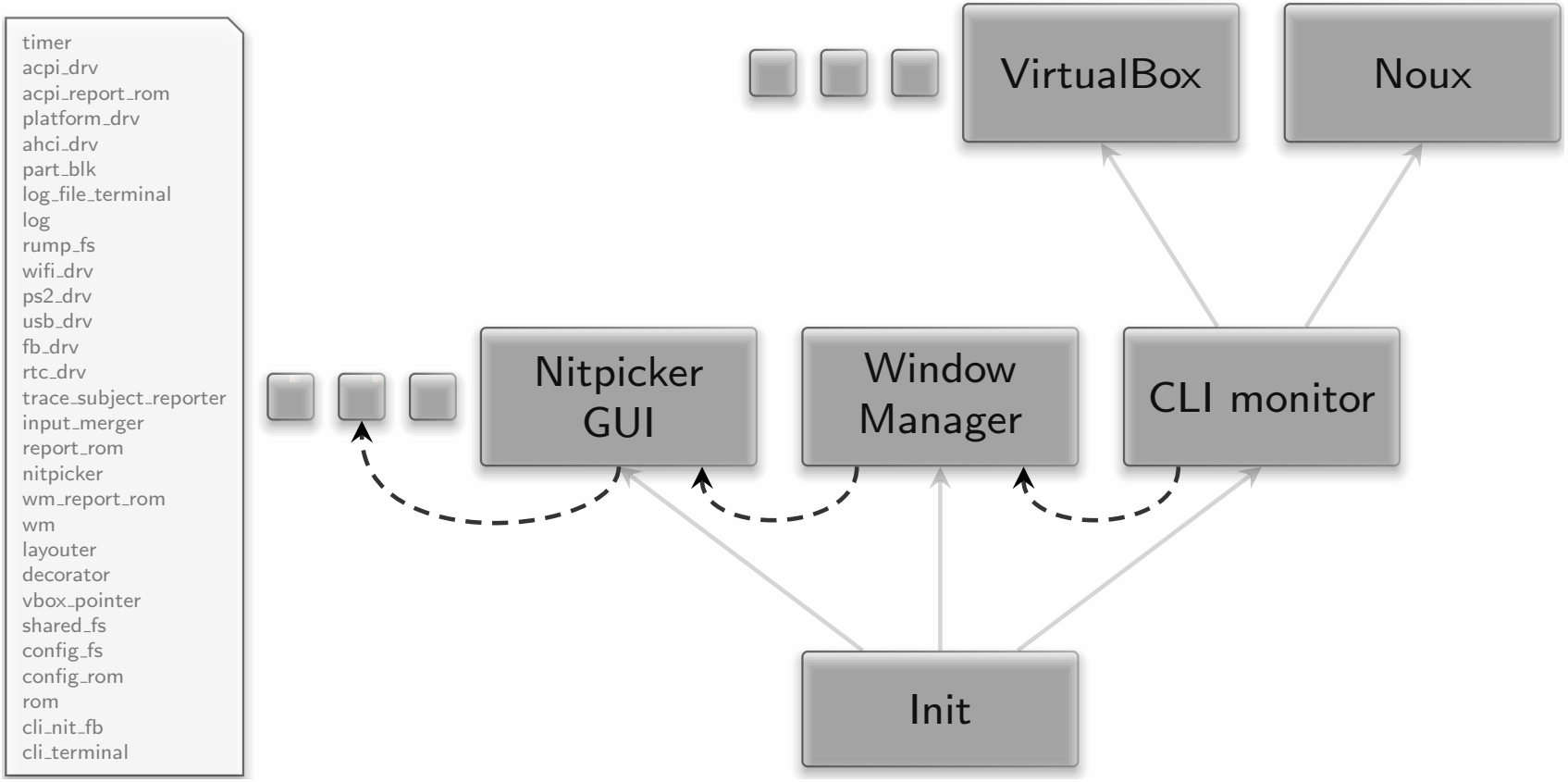


Multi-component applications

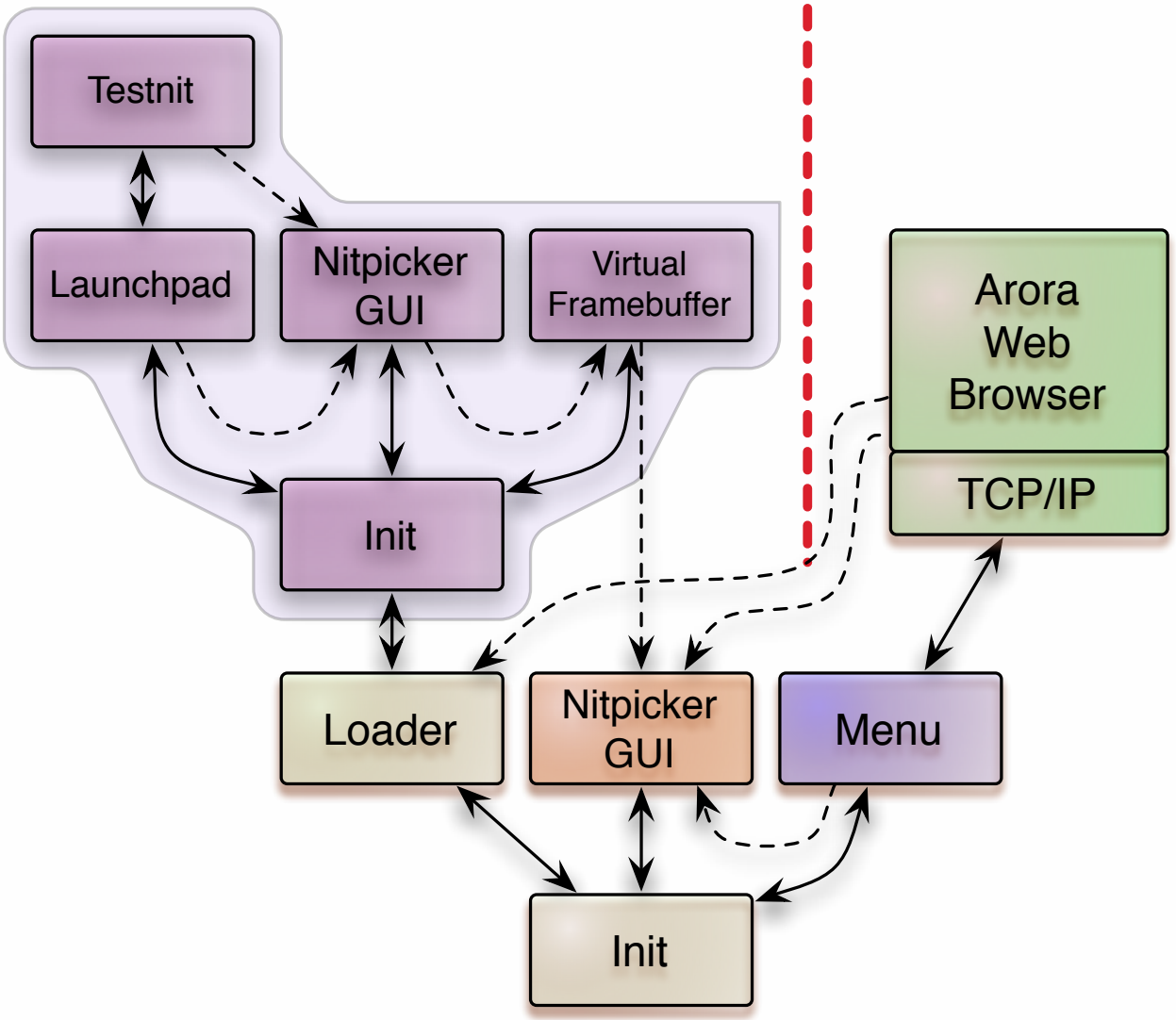
Multi-component applications



“Turmvilla” scenario



Rich applications



Outline

1. Why another operating system?
2. Architectural principles
3. Framework for building operating systems
4. Desktop scenarios
5. Present and future

Disclaimer

- Currently used by only a few enthusiasts
- No package management
- Limited hardware support
- Not yet palatable for uninitiated end users

Ambitions

- Eating our own dog food (tool chain, email, IRC...)

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms
- RISC-V

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms
- RISC-V
- USB Armory

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms
- RISC-V
- USB Armory
- Nix package manager

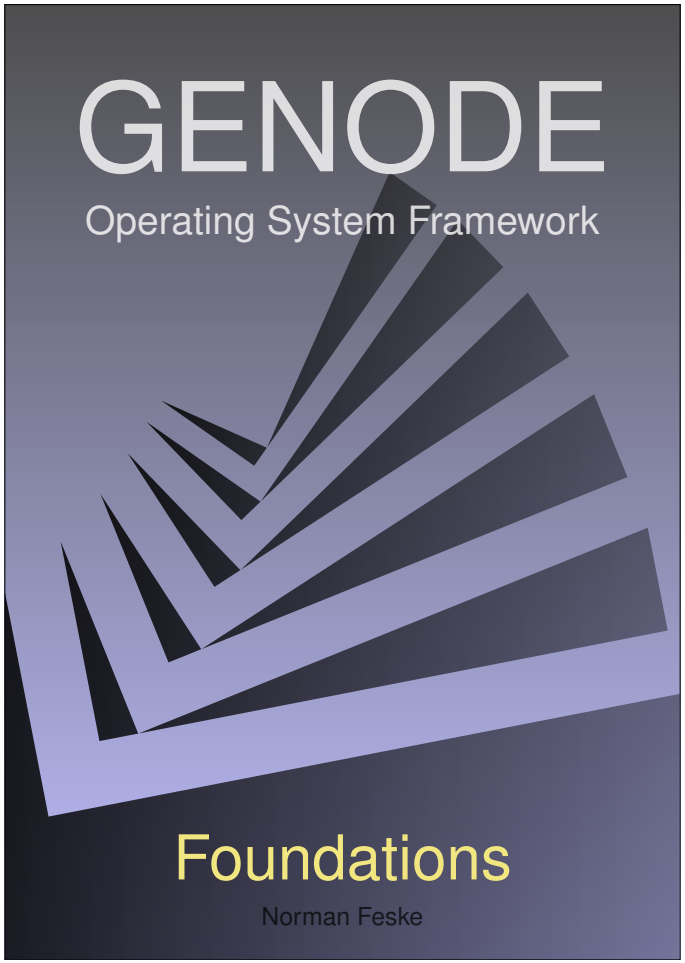
Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms
- RISC-V
- USB Armory
- Nix package manager
- Collaborating with Qubes?

Ambitions

- Eating our own dog food (tool chain, email, IRC...)
- Capability-based desktop environment
- Muen and seL4 as base platforms
- RISC-V
- USB Armory
- Nix package manager
- Collaborating with Qubes?

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