Pluggable Device Drivers for Genode

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

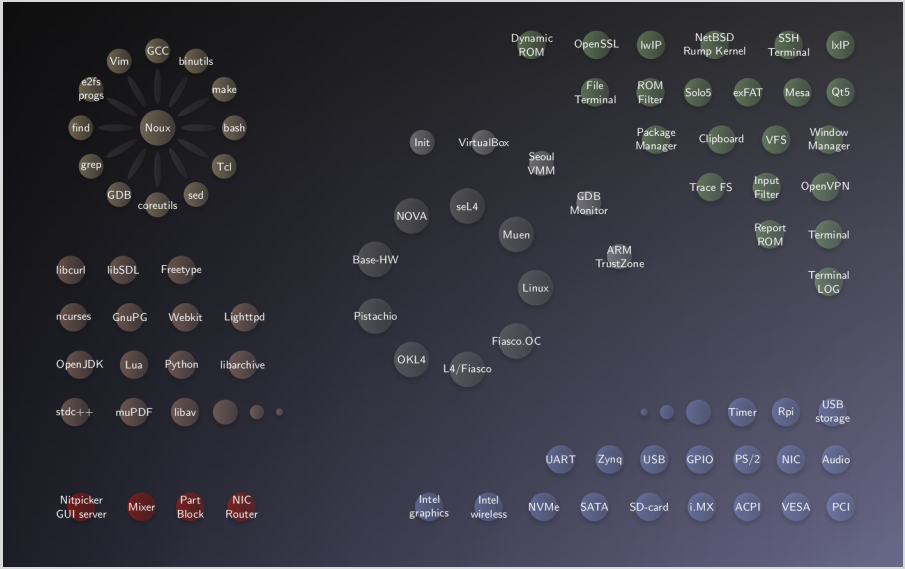
Outline

1. Background
2. Re-stacking the GUI stack
3. Pluggable network drivers
4. Bottom line and Outlook

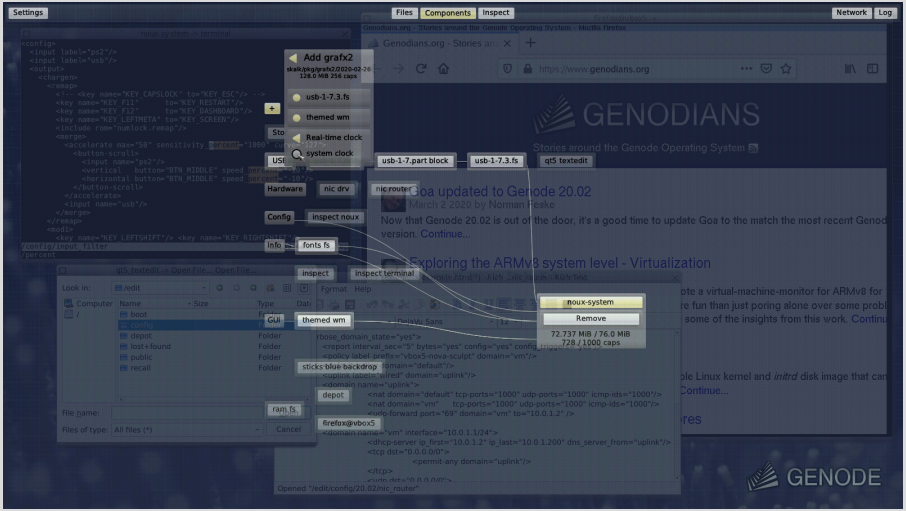
Outline

1. Background
2. Re-stacking the GUI stack
3. Pluggable network drivers
4. Bottom line and Outlook

Genode OS Framework



Fault encapsulation in Sculpt OS



≈60 sandboxes right after boot

Ambitions

Long-running systems

- Partial system updates / downgrades
- Self-healing as promoted by MINIX3

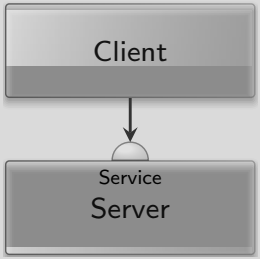
Assurance and fail-safety

- Low-complexity trusted computing base
- Drivers are known to be flaky

Adaptiveness at runtime

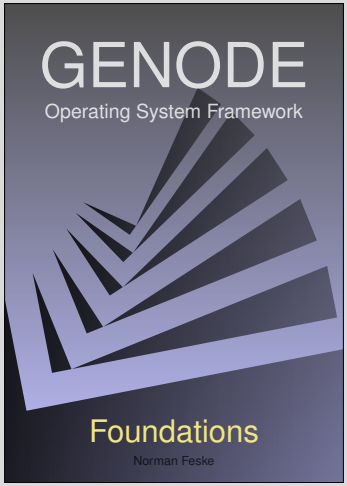
- Changing output / input devices on the fly
- Power-gating of individual devices
  (saving energy of mobile phones)

Genode's strict client-server model



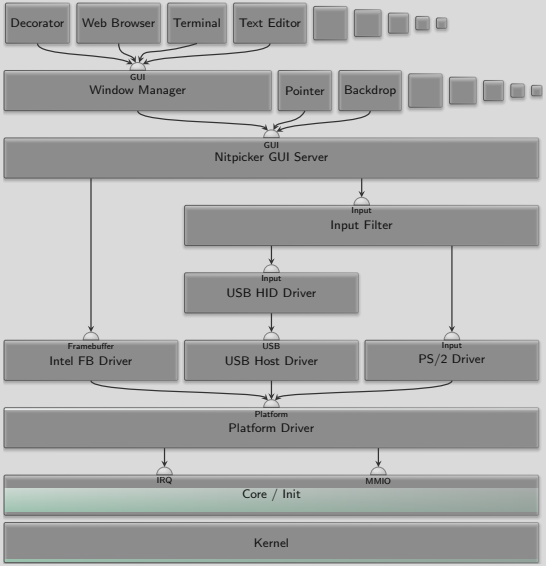
- Clients and servers are sandboxed independently
- Clients lend resources to servers for their services
- Mutual distrust (confidentiality, integrity)
- Liveliness of a client depends on the used servers
- Servers do not depend on clients

The Book “Genode Foundations”

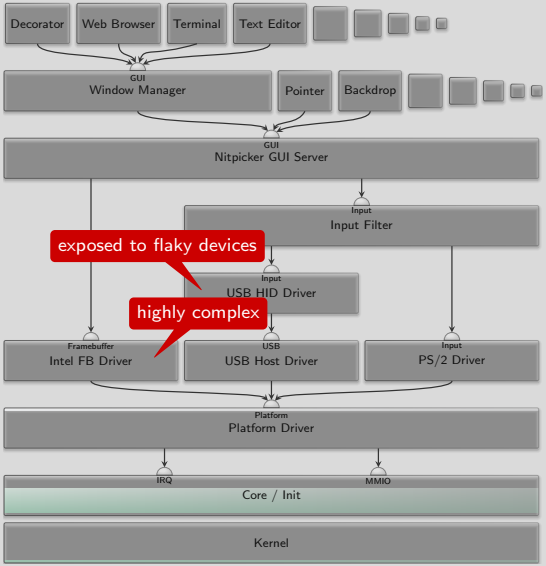


https://genode.org/documentation/genode-foundations-20-05.pdf

Layered architectures



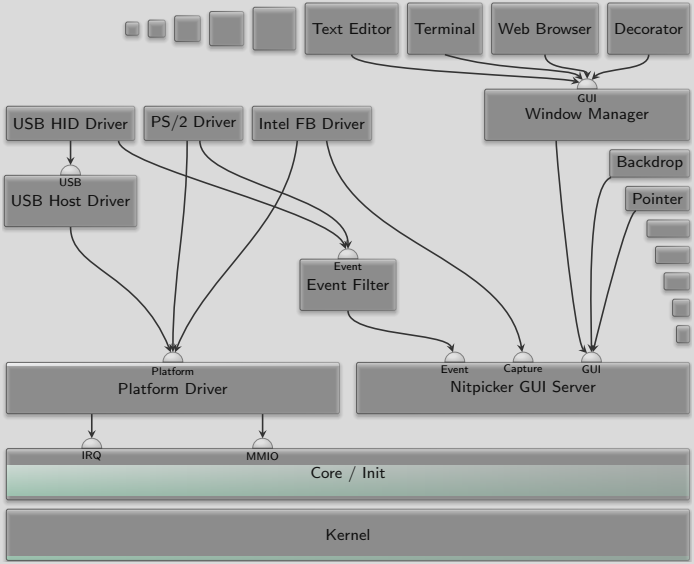
Layered architectures (real world)



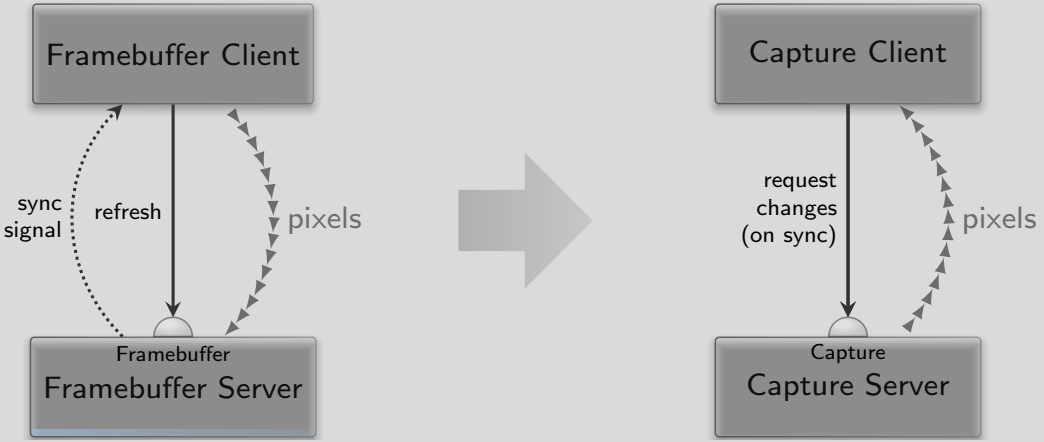
Outline

1. Background
2. Re-stacking the GUI stack
3. Pluggable network drivers
4. Bottom line and Outlook

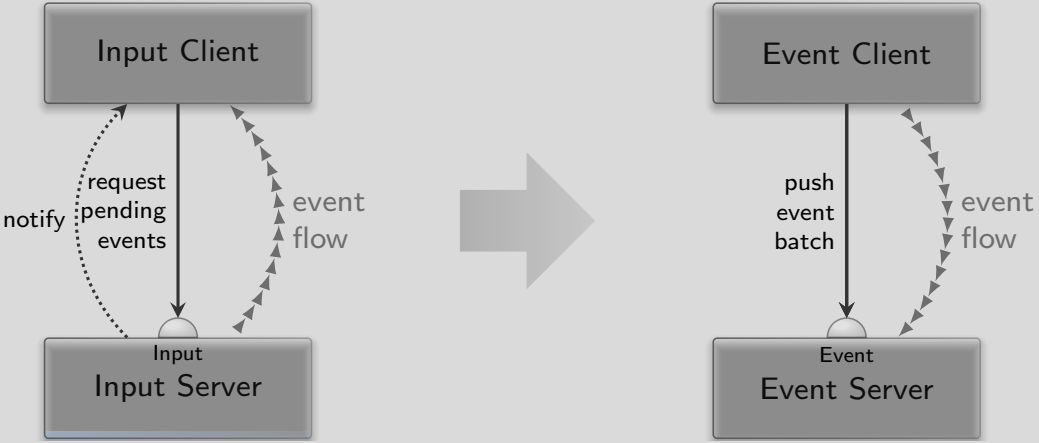
Re-stacking the GUI stack



Introducing the capture service interface



Introducing the event service interface



Challenge

How to get from the old to the new architecture?

- Around 50 existing system scenarios, i. e., Sculpt OS

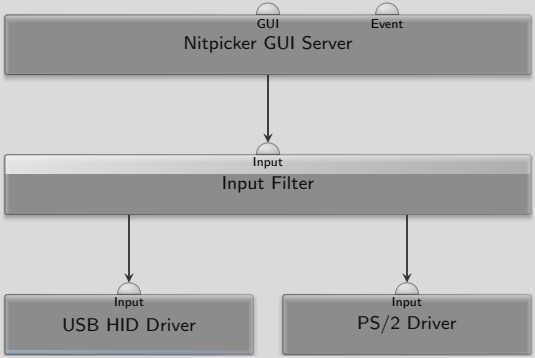
Arsenal of existing device drivers

- Framebuffer drivers
  OMAP4, Exynos5, Intel, i.MX8, i.MX53, Boot, PL11x, Rpi, SDL, VESA
- Input drivers
  PS/2, USB HID, touch, ACPI input

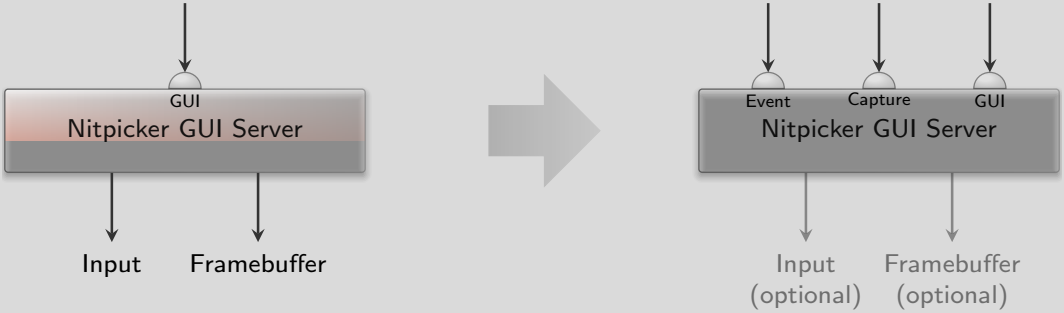
Nitpicker GUI server and companions

- Support new architecture
- Preserve nesting ability
- Reversal of the input filter

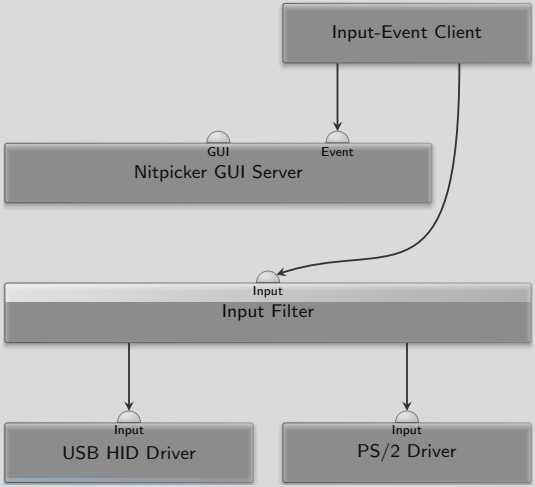
Starting point



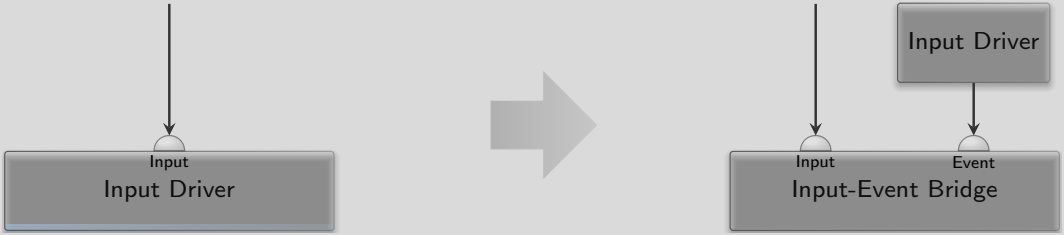
Extending the Nitpicker GUI server



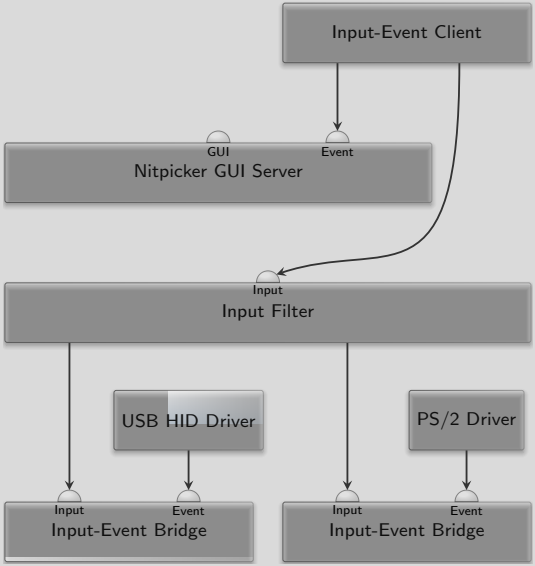
Cutting Nitpicker's dependency to the Input service



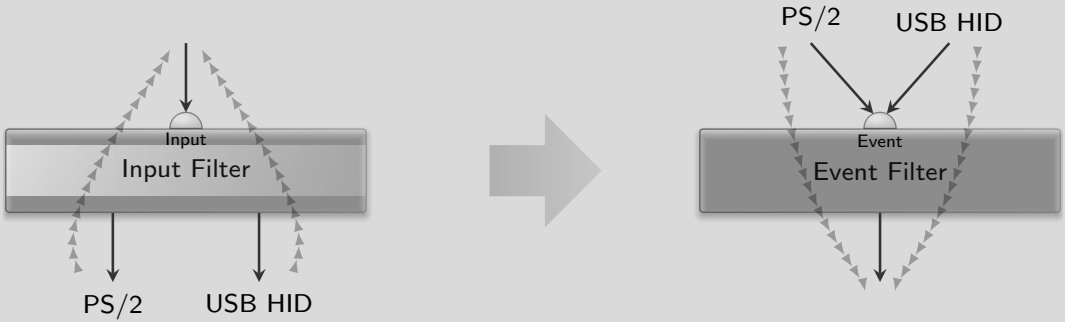
Inverting the drivers piece by piece



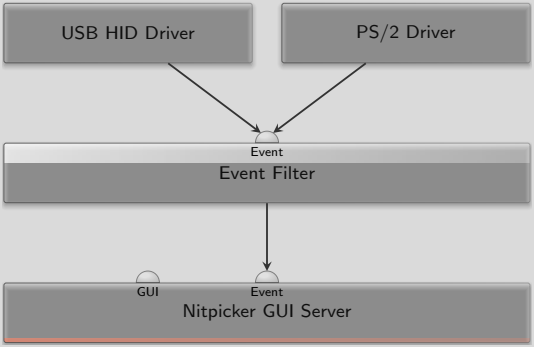
All drivers reversed



Replacing the input filter by new event filter



Complexity collapses



Puzzle pieces coming together

Platform driver

- Arbitration of bus access (access control)
- IOMMU address space per driver
- Power-gating PCI devices

Heartbeat monitoring

- Liveliness of components $\leftrightarrow$ response to I/O

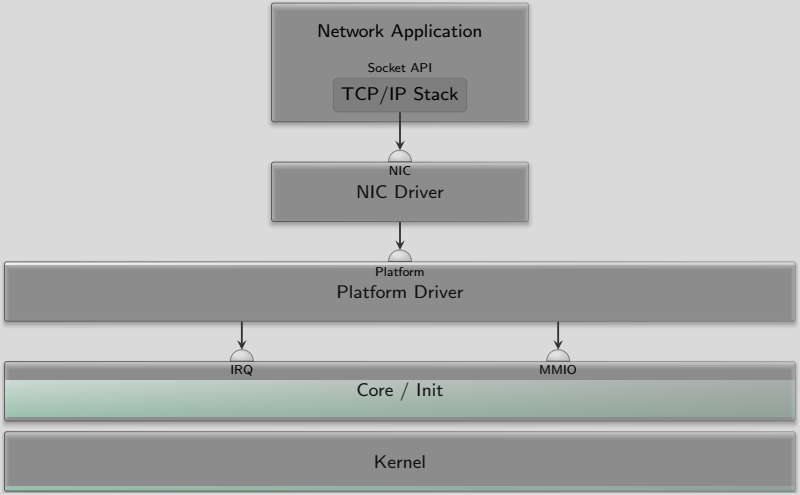
Fault injection mechanism

- Genode's trace service
- Default trace points (RPC call, RPC request, ...)
- Policy code executed by traced subjects (thread-local)
- Division by zero as trace policy code

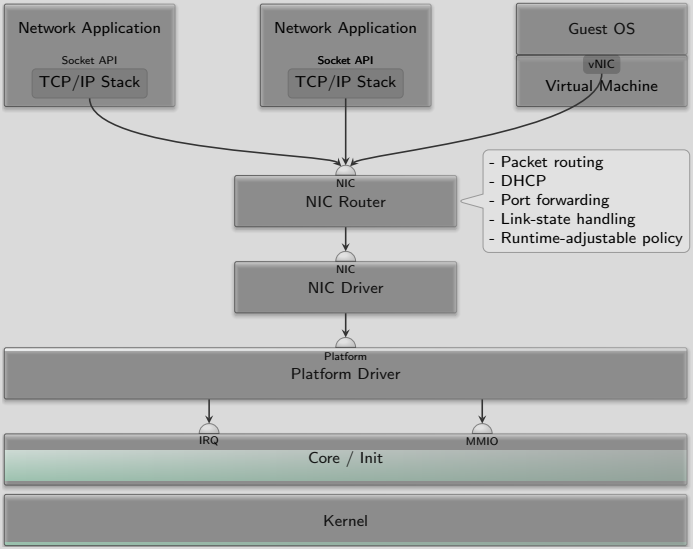
Outline

1. Background
2. Re-stacking the GUI stack
3. Pluggable network drivers
4. Bottom line and Outlook

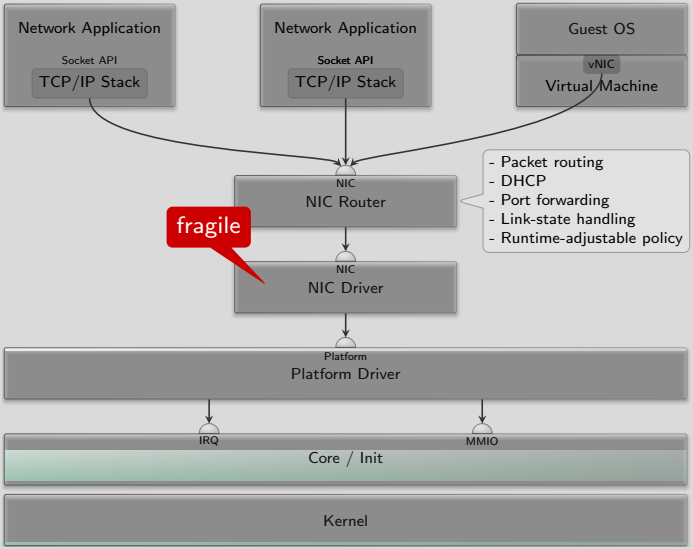
Traditional architecture (single application)



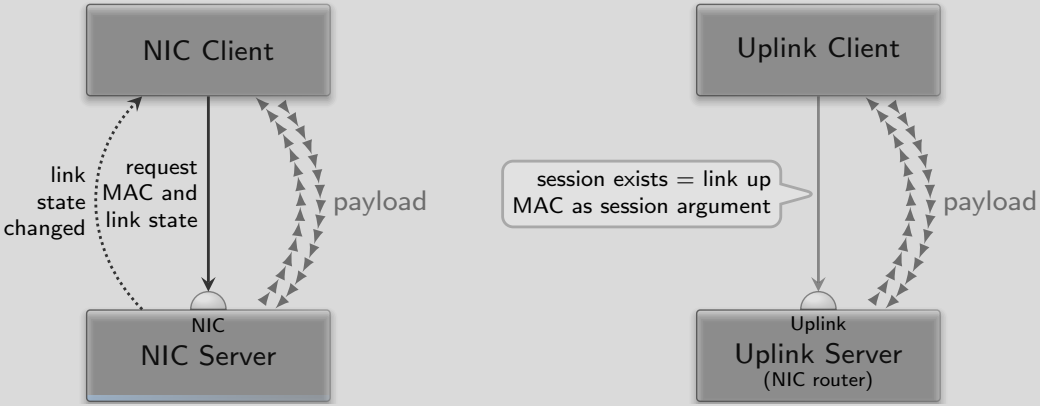
Traditional architecture (practical scenarios)



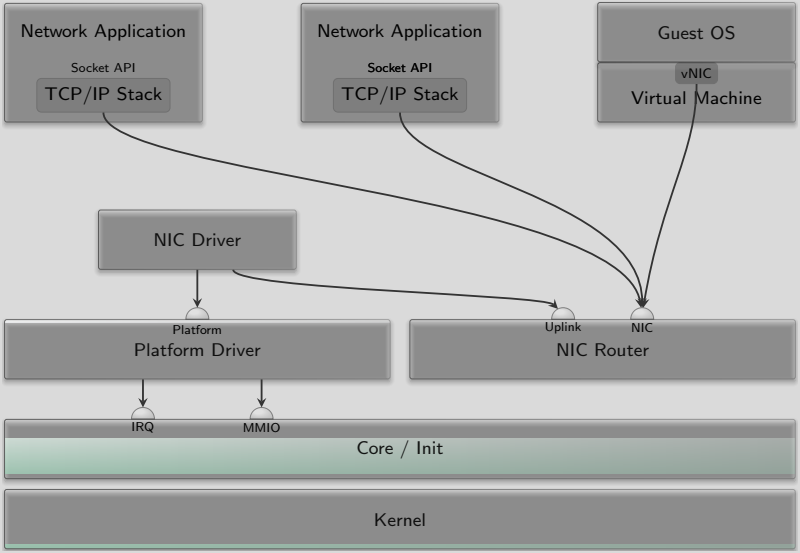
Traditional architecture (real world)



Introducing the Uplink service interface



Disposable NIC drivers



Outline

1. Background
2. Re-stacking the GUI stack
3. Pluggable network drivers
4. Bottom line and Outlook

Limitations

Block-device drivers not covered

- Stateful, consistency, caching
- May be addressed separately...

Bus drivers *e. g., USB host controller*

- Need to be long-living
- May have far reach

Prospects

- Dynamic multi-head scenarios
- Screen capturing
- Virtual on-screen keyboard
- Swapping out drivers (fallbacks)
- Updating and downgrading of drivers during runtime
- Power saving (on-demand drivers, automated power gating)
- Simplify suspend/resume (driver life-cycle management)

Thank you

Genode OS Framework

https://genode.org

Sculpt OS download and manual

https://genode.org/download/sculpt

Genodians.org community blog

https://genodians.org

Genode Labs GmbH

https://www.genode-labs.com

Company newsletter

https://genode-labs.com/newsletter