

# Genode - OS Security By Design



Dr.-Ing. Norman Feske

<[norman.feske@genode-labs.com](mailto:norman.feske@genode-labs.com)>



# Outline

1. Introduction
2. Architectural Principles
3. Showcases
4. Current Topics





# Outline

1. Introduction
2. Architectural Principles
3. Showcases
4. Current Topics





## Universal Truths

**Assurance**  **Scalability**

Accountability  Utilization

Security  Ease of use





## 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)
- 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
  - 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] [] ? cpuset_print_task_mems_allowed+0x9d/0xb0
Jul 24 12:58:30 neo kernel: [72454.482286] [] dump_header+0x91/0xe0
Jul 24 12:58:30 neo kernel: [72454.482289] [] oom_kill_process+0x85/0xb0
Jul 24 12:58:30 neo kernel: [72454.482293] [] out_of_memory+0xfa/0x250
Jul 24 12:58:30 neo kernel: [72454.482298] [] __alloc_pages_nodemask+0x80f/0x820
Jul 24 12:58:30 neo kernel: [72454.482304] [] ? noalloc_get_block_write+0x30/0x30
Jul 24 12:58:30 neo kernel: [72454.482311] [] alloc_pages_current+0xa3/0x110
Jul 24 12:58:30 neo kernel: [72454.482314] [] __page_cache_alloc+0x8f/0xa0
Jul 24 12:58:30 neo kernel: [72454.482318] [] ? find_get_page+0x1e/0x90
Jul 24 12:58:30 neo kernel: [72454.482321] [] filemap_fault+0x234/0x3e0
Jul 24 12:58:30 neo kernel: [72454.482326] [] ? mem_cgroup_update_page_stat+0x2b/0x110
Jul 24 12:58:30 neo kernel: [72454.482330] [] __do_fault+0x54/0x510
Jul 24 12:58:30 neo kernel: [72454.482334] [] handle_pte_fault+0xfa/0x210
Jul 24 12:58:30 neo kernel: [72454.482337] [] handle_mm_fault+0x1f8/0x350
Jul 24 12:58:30 neo kernel: [72454.482344] [] do_page_fault+0x153/0x530
Jul 24 12:58:30 neo kernel: [72454.482350] [] ? read_tsc+0x9/0x20
Jul 24 12:58:30 neo kernel: [72454.482355] [] ? ktime_get_ts+0xad/0xe0
Jul 24 12:58:30 neo kernel: [72454.482361] [] ? poll_select_set_timeout+0x7a/0x90
Jul 24 12:58:30 neo kernel: [72454.482365] [] 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

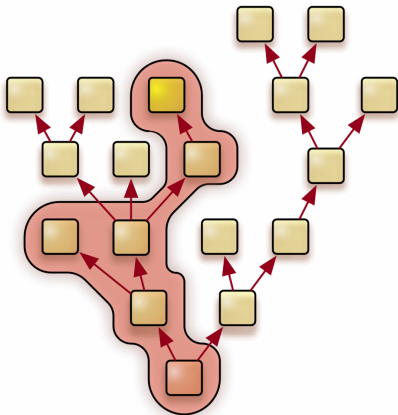
- Microkernels
- Componentization, kernelization
- Capability-based security
- Virtualization

*...but how to compose those?*





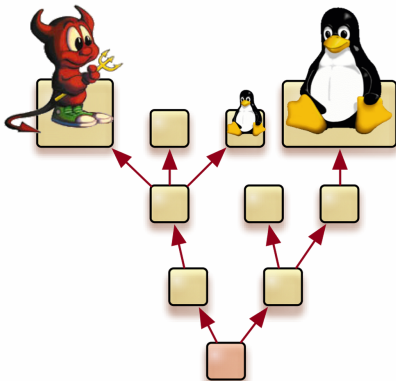
## Genode architecture



→ Application-specific TCB



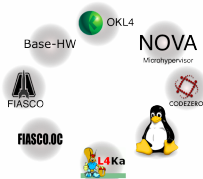
## Combined with virtualization





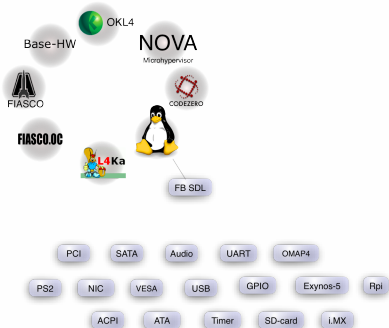


# Components



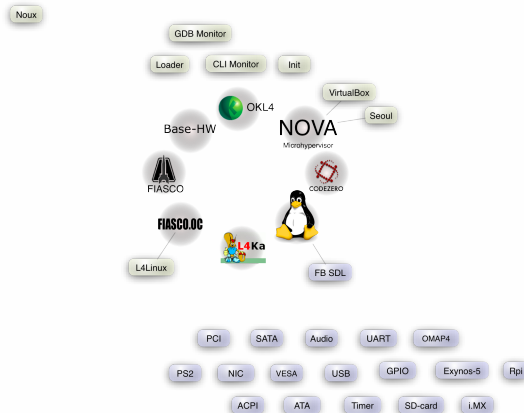


# Components



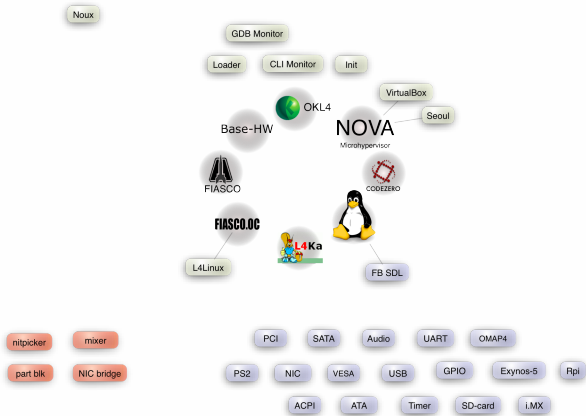


# Components



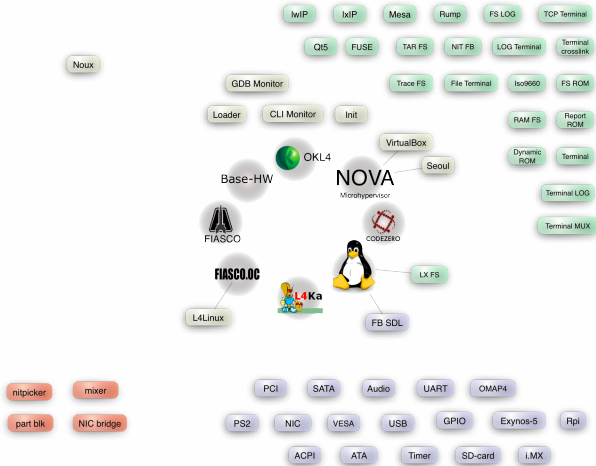


# Components



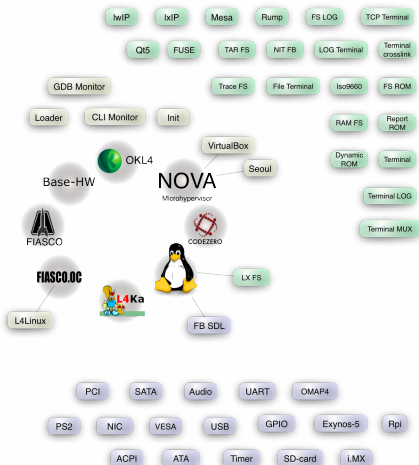
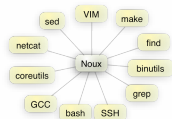


# Components





# Components





# Outline

1. Introduction
2. Architectural Principles
3. Showcases
4. Current Topics





## Object capabilities

### Delegation of authority between components

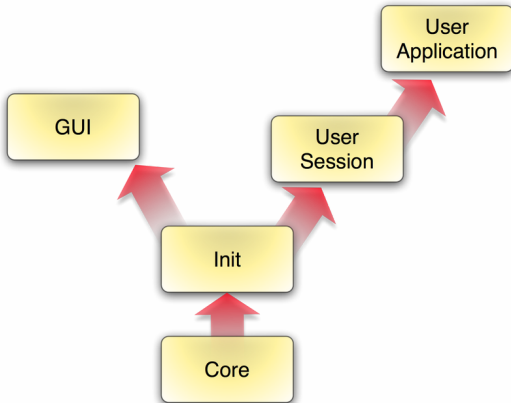
- Each component lives in a virtual environment
- A component that possesses a *capability* can
  - ▶ Use it (*invoke*)
  - ▶ Delegate it to acquainted components





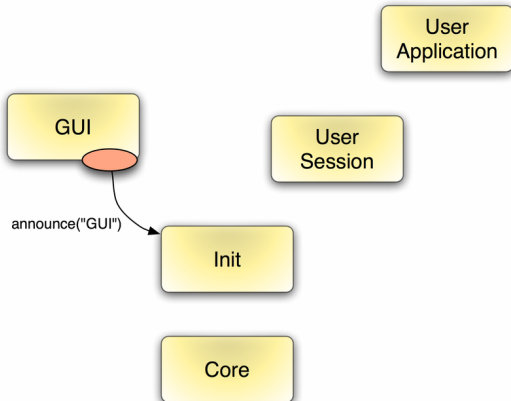


# Recursive system structure



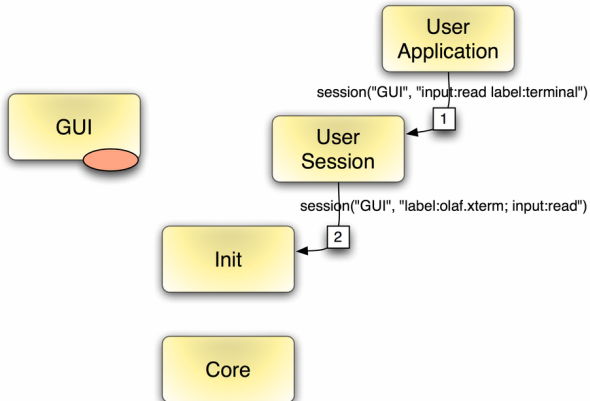


# Service announcement



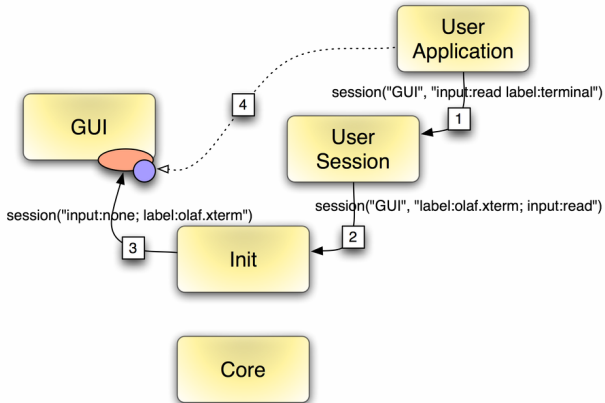


# Session creation





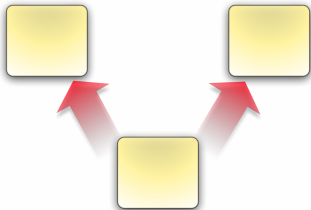
# Session creation





## Resource management

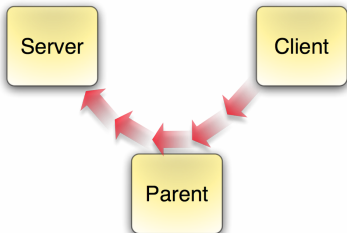
Explicit assignment of physical resources to components





## Resource management (II)

Resources can be attached to sessions





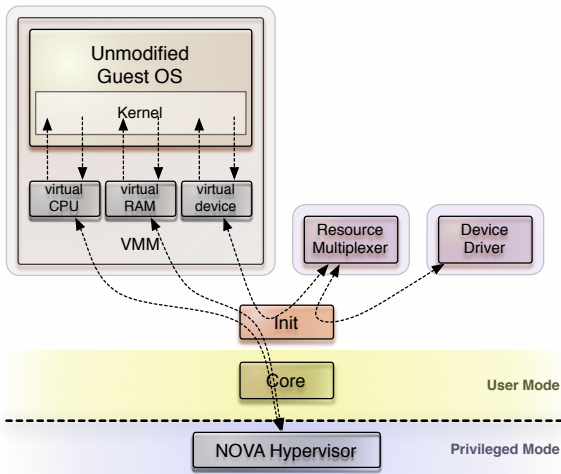
# Outline

1. Introduction
2. Architectural Principles
- 3. Showcases**
4. Current Topics





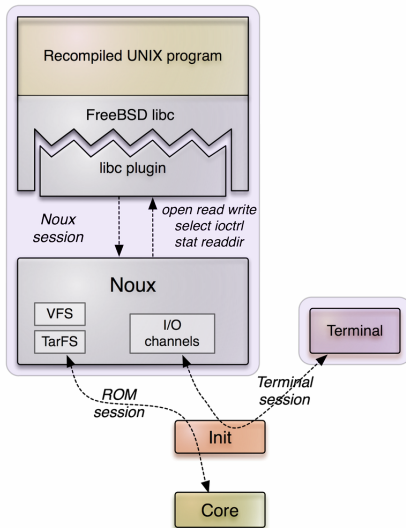
# Faithful Virtualization





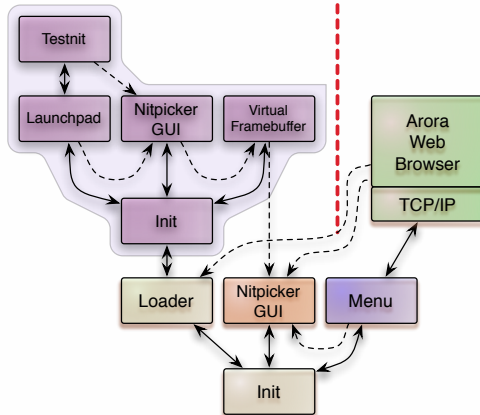


# OS-level Virtualization





# Rich applications





# Outline

1. Introduction
2. Architectural Principles
3. Showcases
4. Current Topics





## Current Topics

- Eating our own dog food
  - ▶ Noux (GCC, VIM, bash, coreutils...)
  - ▶ Wireless networking
- Capability-based user interface
- seL4 kernel as base platform
- ARM Virtualization
- Package management





# 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>

