

The agony of choice - the diversity of microkernels in Genode



Stefan Kalkowski

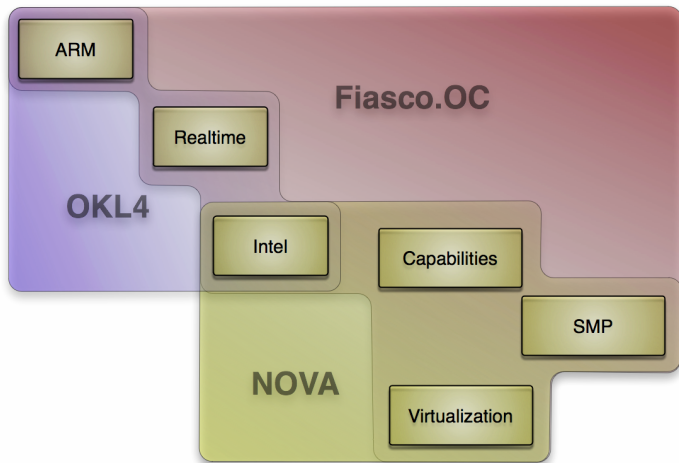


Outline

1. Advantages of diversity
 - Motivation
 - Code quality
 - Handling the multiteity
2. Porting Genode to a kernel
 - What is needed
 - Course of action
 - Conclusion

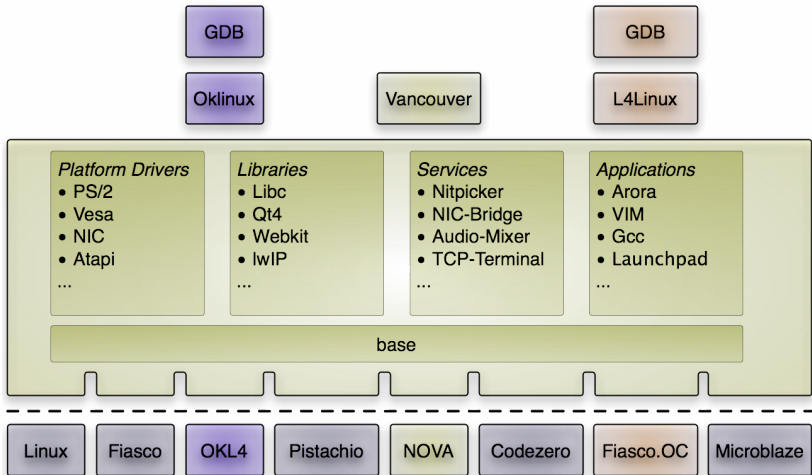


Multi-Dimensional Feature Space



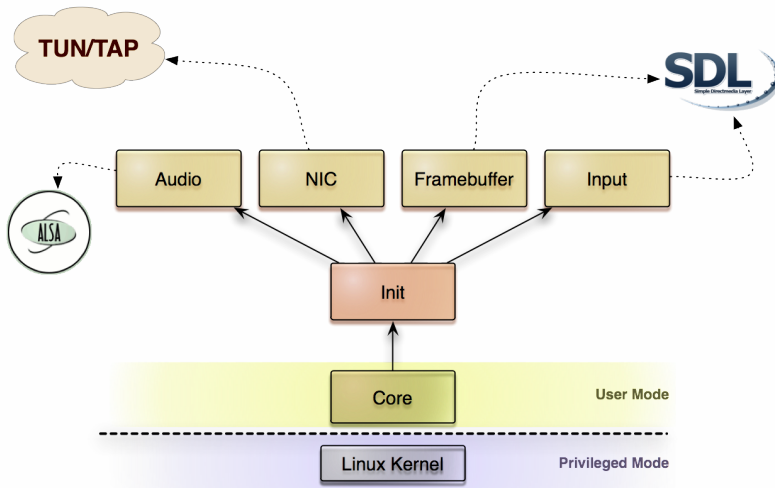


Get All-Inclusive



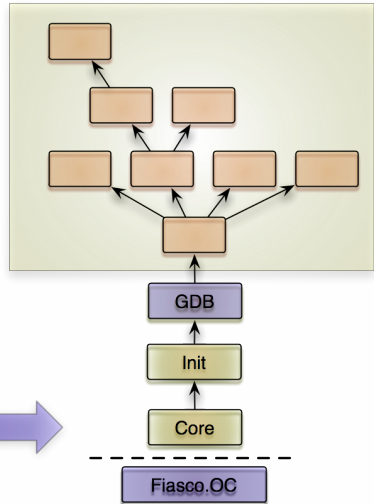
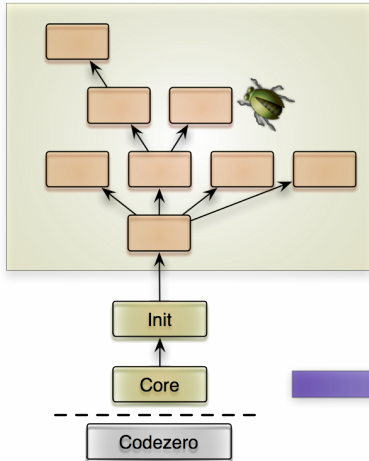


Convenient Developing Under Linux



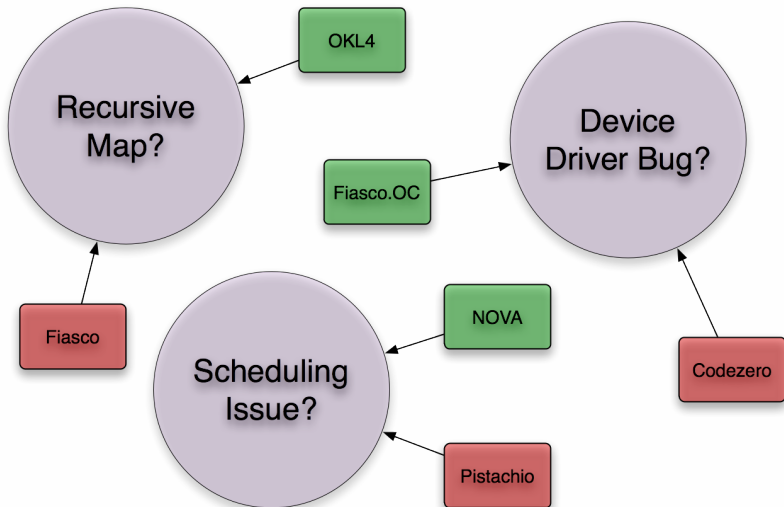


Develop For One, Debug With Another





Problem Localization





Outline

1. Advantages of diversity

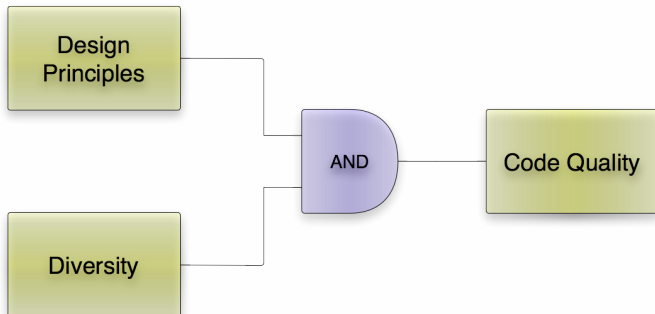
- Motivation
- Code quality
- Handling the multitude

2. Porting Genode to a kernel

- What is needed
- Course of action
- Conclusion



Enhanced Code Quality





Essential Principles

- Low complexity



Essential Principles

- Low complexity
- Strive for a narrow API



Essential Principles

- Low complexity
- Strive for a narrow API
- Unify wherever possible, avoid code duplication

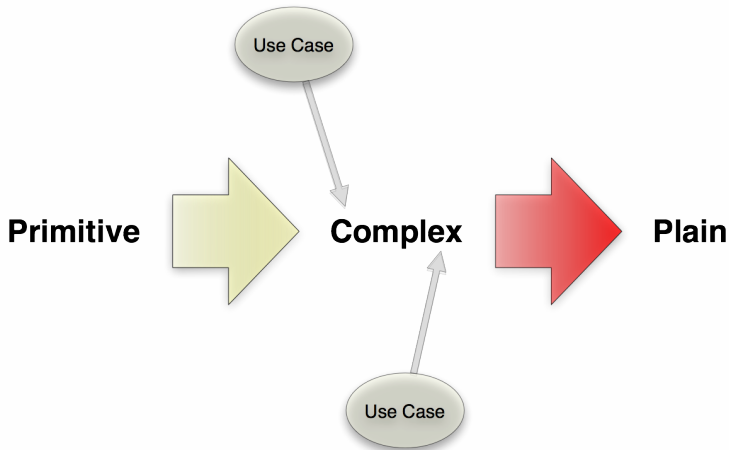


Essential Principles

- Low complexity
- Strive for a narrow API
- Unify wherever possible, avoid code duplication
- No premature optimization

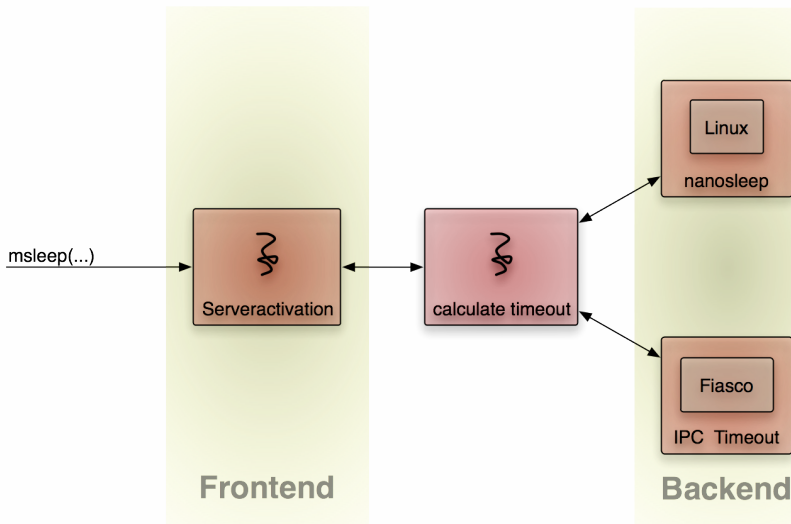


Component's Lifecycle



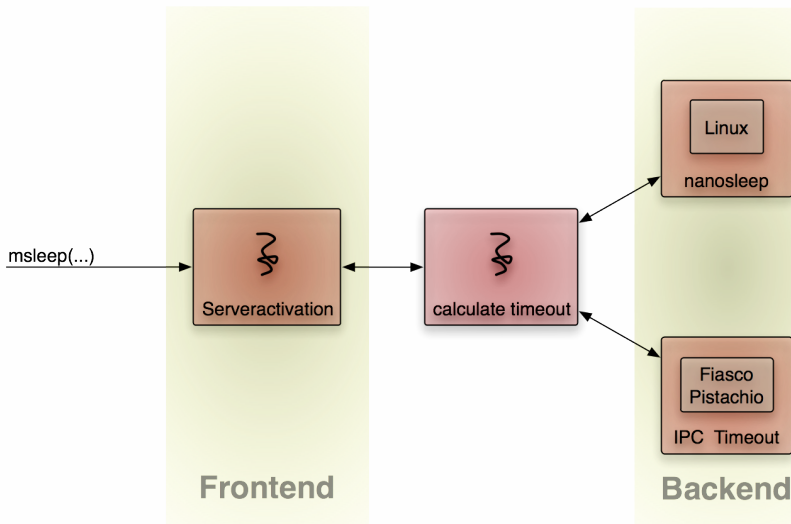


Timer - An Unfinished Example



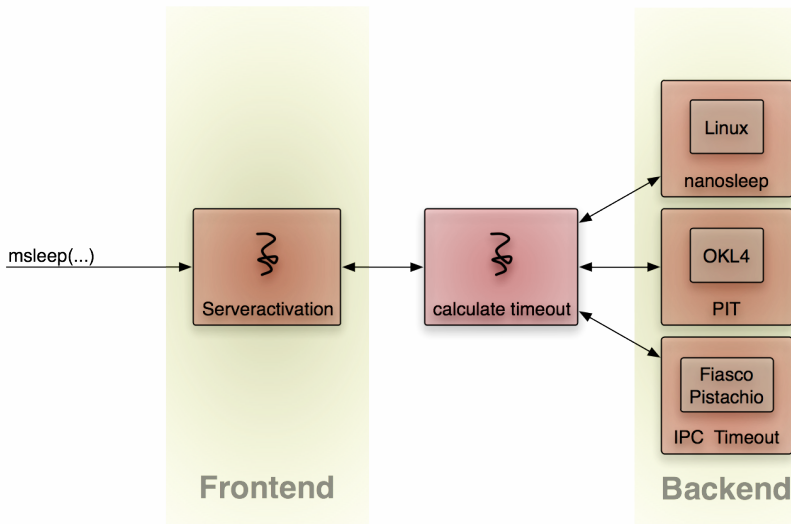


Timer - Introducing Pistachio



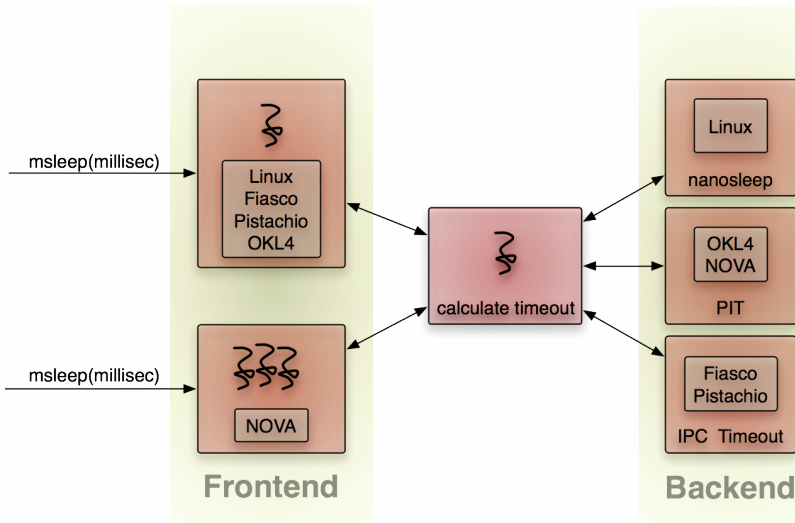


Timer - Introducing OKL4



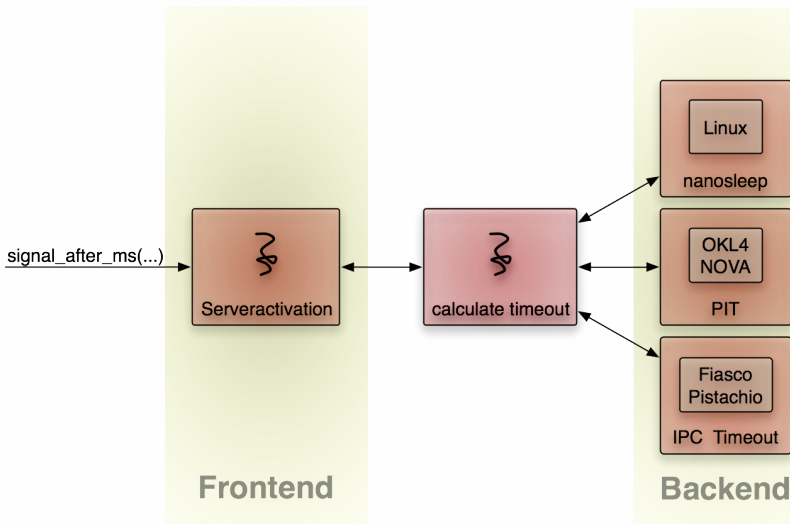


Timer - Introducing NOVA





Timer - Change API





Outline

1. Advantages of diversity

- Motivation
- Code quality
- **Handling the multevity**

2. Porting Genode to a kernel

- What is needed
- Course of action
- Conclusion



The Dark Side Of The Force

- Plethora of tools needed



The Dark Side Of The Force

- Plethora of tools needed
- Knowledge of build-systems required



The Dark Side Of The Force

- Plethora of tools needed
- Knowledge of build-systems required
- How to boot the system



The Dark Side Of The Force

- Plethora of tools needed
- Knowledge of build-systems required
- How to boot the system
- Unmaintained software



The Dark Side Of The Force

- Plethora of tools needed
- Knowledge of build-systems required
- How to boot the system
- Unmaintained software

Solution: unify toolchain + convenience tools



Demo

Short demo ...



Run-Scripts

```
#  
# Example run-script  
#  
build {  
    core init drivers/timer ...  
}  
  
create_boot_directory  
install_config { ... }  
  
set boot_modules {  
    core init timer ...  
}  
lappend_if [have_spec linux] boot_modules fb_sdl  
build_boot_image $boot_modules  
  
append_qemu_args " -m 256 "  
run_genode_until forever
```

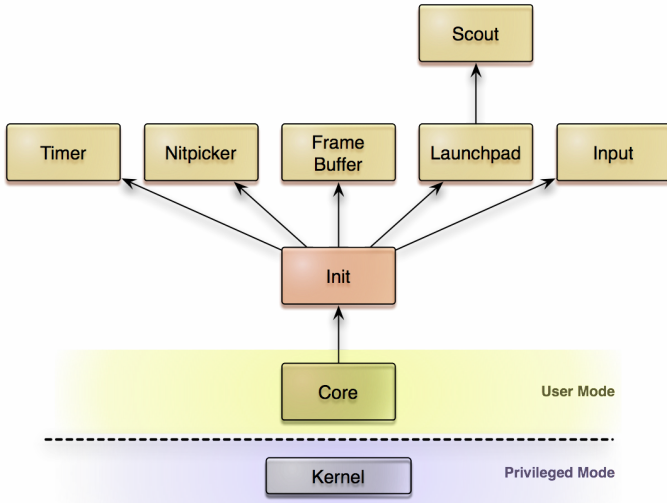


Outline

1. Advantages of diversity
 - Motivation
 - Code quality
 - Handling the multiteity
2. Porting Genode to a kernel
 - What is needed
 - Course of action
 - Conclusion

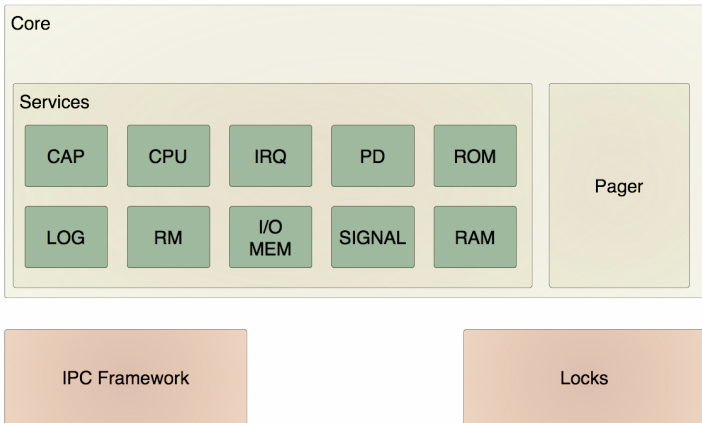


Goal



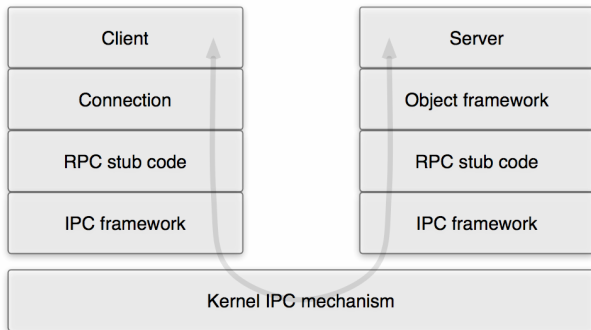


Kernel-Specific Parts



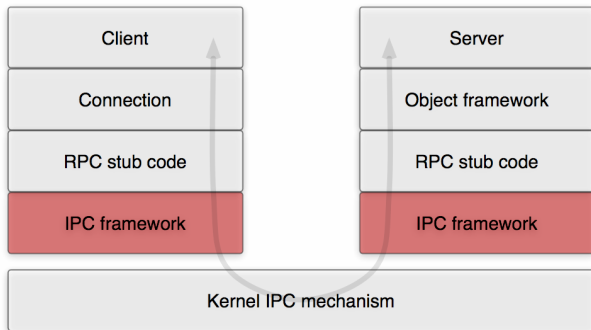


IPC Framework



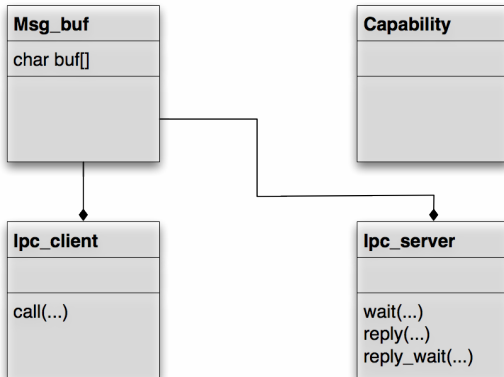


IPC Framework



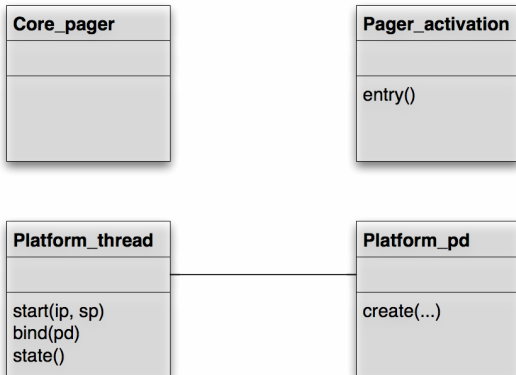


IPC Framework





Threads, Protection Domains, Pager





Lock

```
/**
 * base-<kernel>/src/base/lock_helper.h
 */

void thread_yield();

bool thread_check_stopped_and_restart(Native_thread_id id);

Native_thread_id thread_get_my_native_id();

Native_thread_id thread_invalid_id();

bool thread_id_valid(Native_thread_id id);

void thread_switch_to(Native_thread_id id);

void thread_stop_myself();
```



Platform Information

- Parse kernel + bootloader info
- Platform specific compile-time knowledge



Platform Information

- Parse kernel + bootloader info
- Platform specific compile-time knowledge
- Sizing allocators and databases for
 - ▶ RAM
 - ▶ ROM modules
 - ▶ IRQ numbers
 - ▶ I/O memory (and ports)



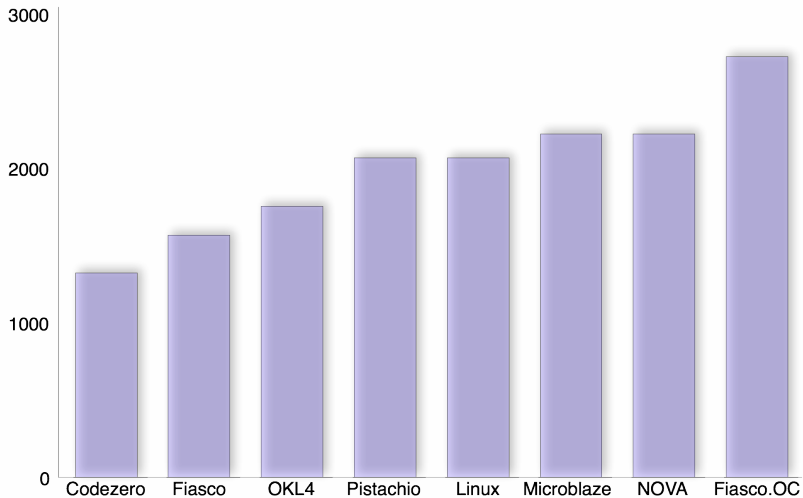
Things Left

- Interrupts
- Timer
- Signals





Effort: Kernel-Specific LOC





What Benefit Do I Have?



What Benefit Do I Have?

```
VIM - VI Improved
~
~
~
~      version 7.3
~      by Bram Moolenaar et al.
~
~  Vim is open source and freely distributable
~
~
~      Help poor children in Uganda!
~      type :help iccf<Enter> for information
~
~
~      type :q<Enter> to exit
~      type :help<Enter> or <F1> for on-line help
~      type :help version7<Enter> for version info
~
~
~
~ [No Name]
```

Genode Web-Browser Demo (1/4) - Arora

File Edit View History Bookmarks Window Tools Help

Genode Web-Bro...

Genode Web-Browser Demo (1/4)

Welcome to the web-browser demo, showcasing the Arora web browser with a stack running natively on a microkernel. Beware, it is a tech demo and may be. But we hope you will recognise the potential that lies in the combination of exi Genode's concepts.

The demo consists of three parts. The first two parts do not rely on a network c comes from the Live-CD. So if your network connection does not work, you are the first two experiments. The third demo, however, requires an internet conn

Why bringing Arora to Genode?

There were two reasons for porting Arora to the Genode Framework. First, beca browsers, including Arora, are extremely complex, porting such a huge softwa operating system is a great challenge. Arora has become one of our most adva stressing the base system, the dynamic linker, the TCP/IP stack, and device dri among the available web browsers because we already had Qt4 running on Ger seemed an evolutionary step, which actually turned out to be the case.

But in addition to the technical challenge, we quickly recognised the potential t porting work. Inspired by recent developments of sandboxing techniques for br introduced by Google Chrome, we pursued a generalization of these ideas. Who recursive structure of Genode and its capability-based security are able to vast of browser plugins while, at the same time, making the plugins more flexible.

[Continue: Run a complete Genode subsystem as browser plugin](#)

Finished loading