The agony of choice - the diversity of microkernels in Genode

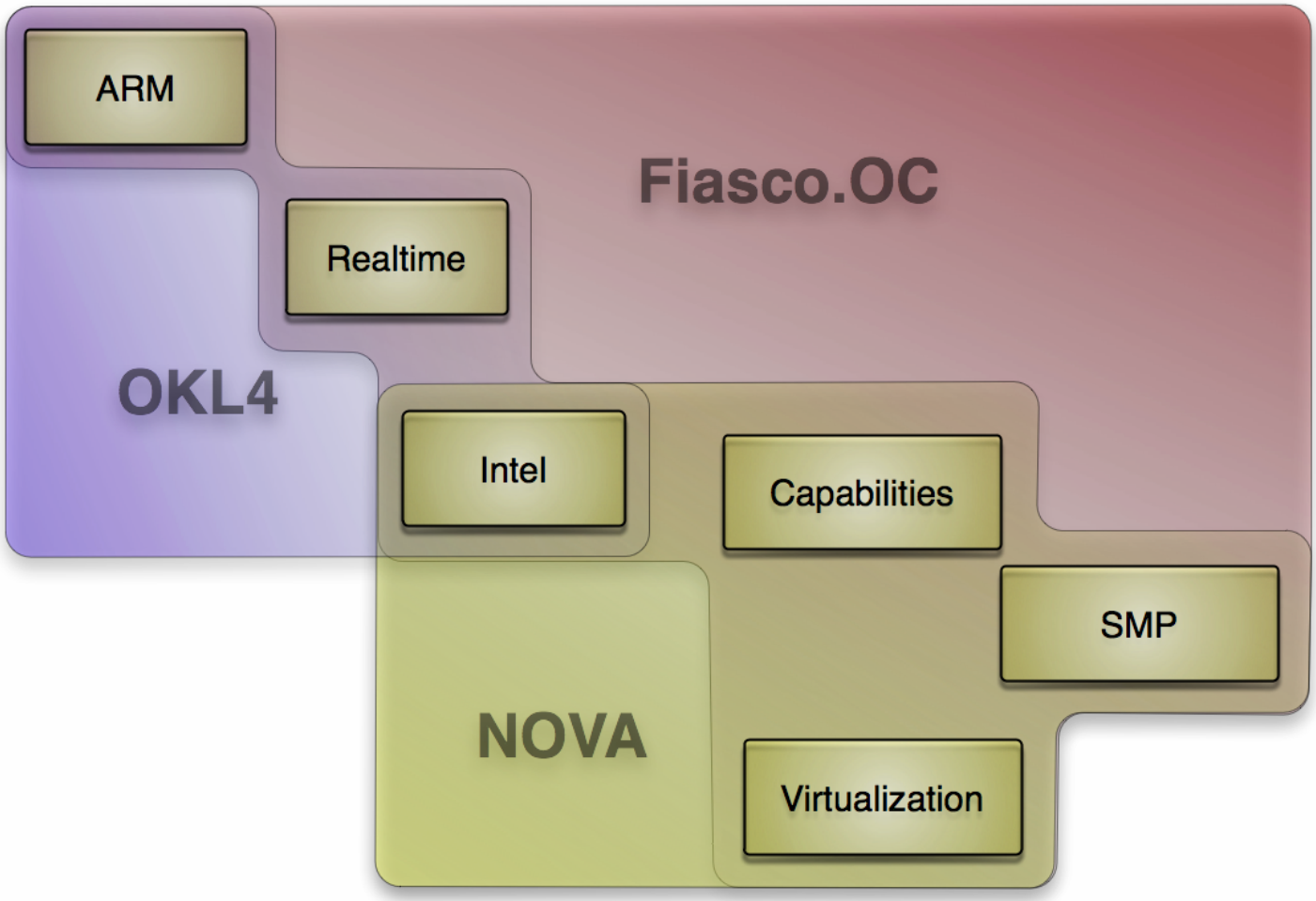
Stefan Kalkowski

Outline

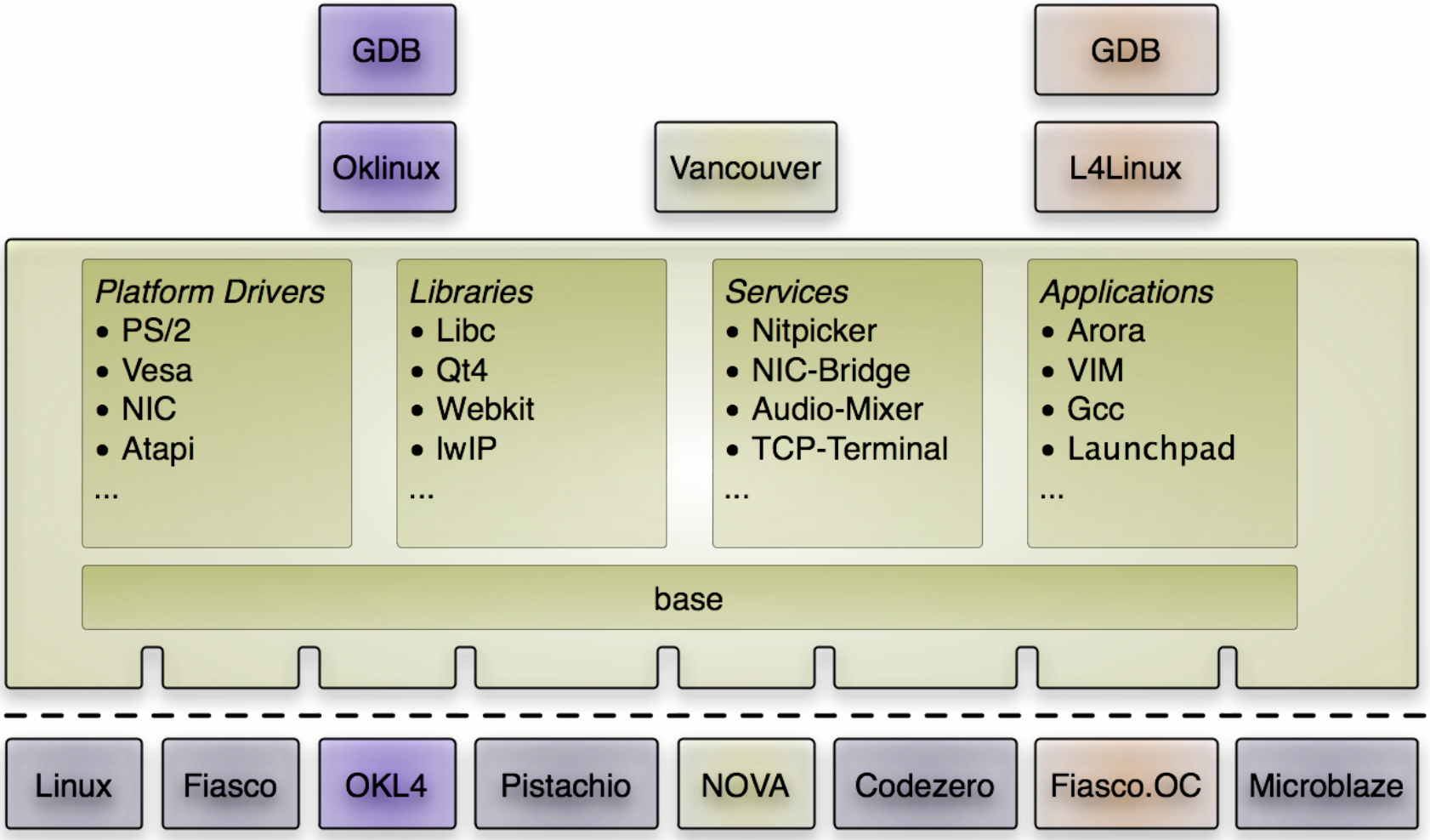
1. Advantages of diversity
   - Motivation
   - Code quality
   - Handling the multeity

2. Porting Genode to a kernel
   - What is needed
   - Course of action
   - Conclusion

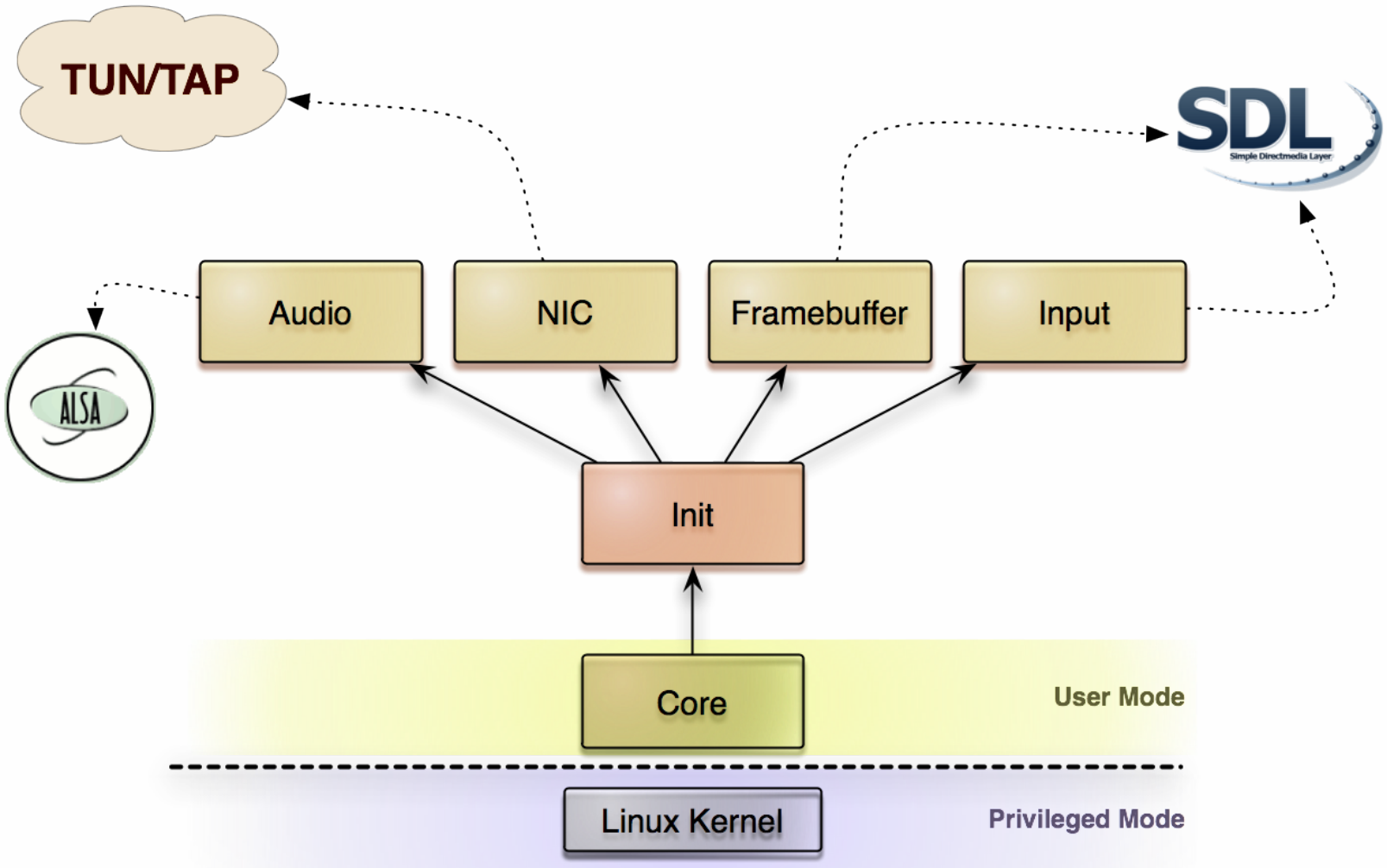
Multi-Dimensional Feature Space



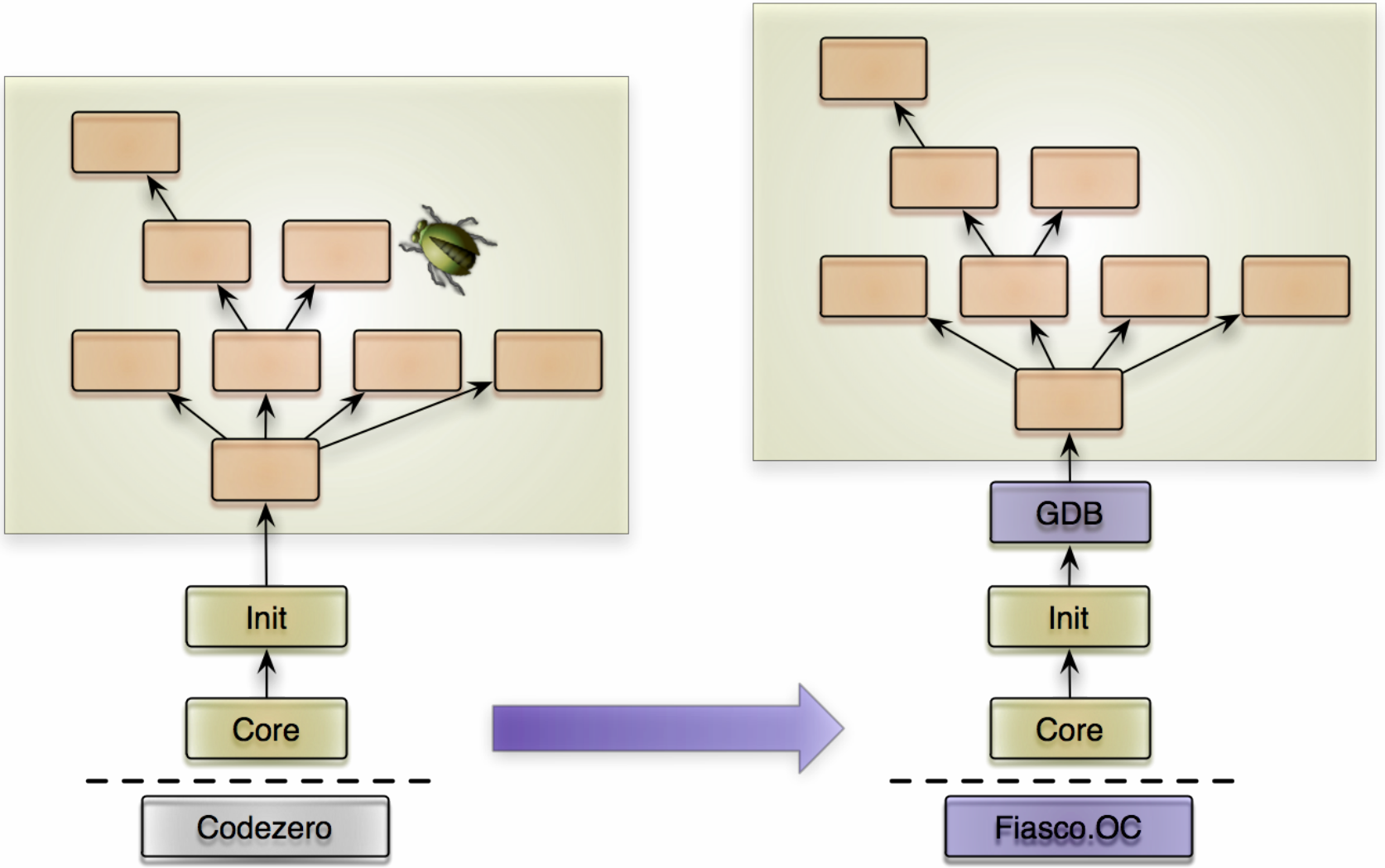
Get All-Inclusive



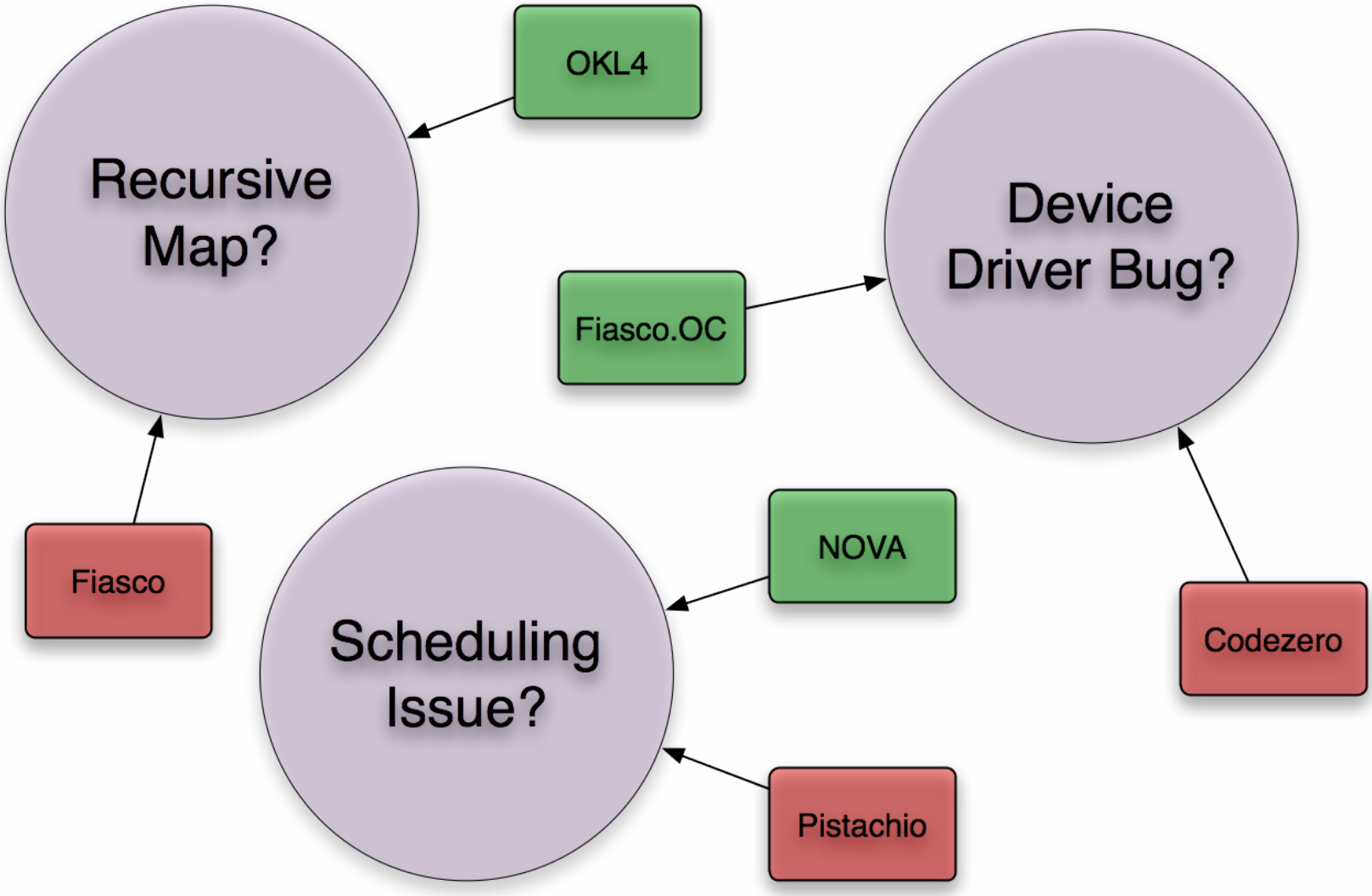
Convenient Developing Under Linux



Develop For One, Debug With Another



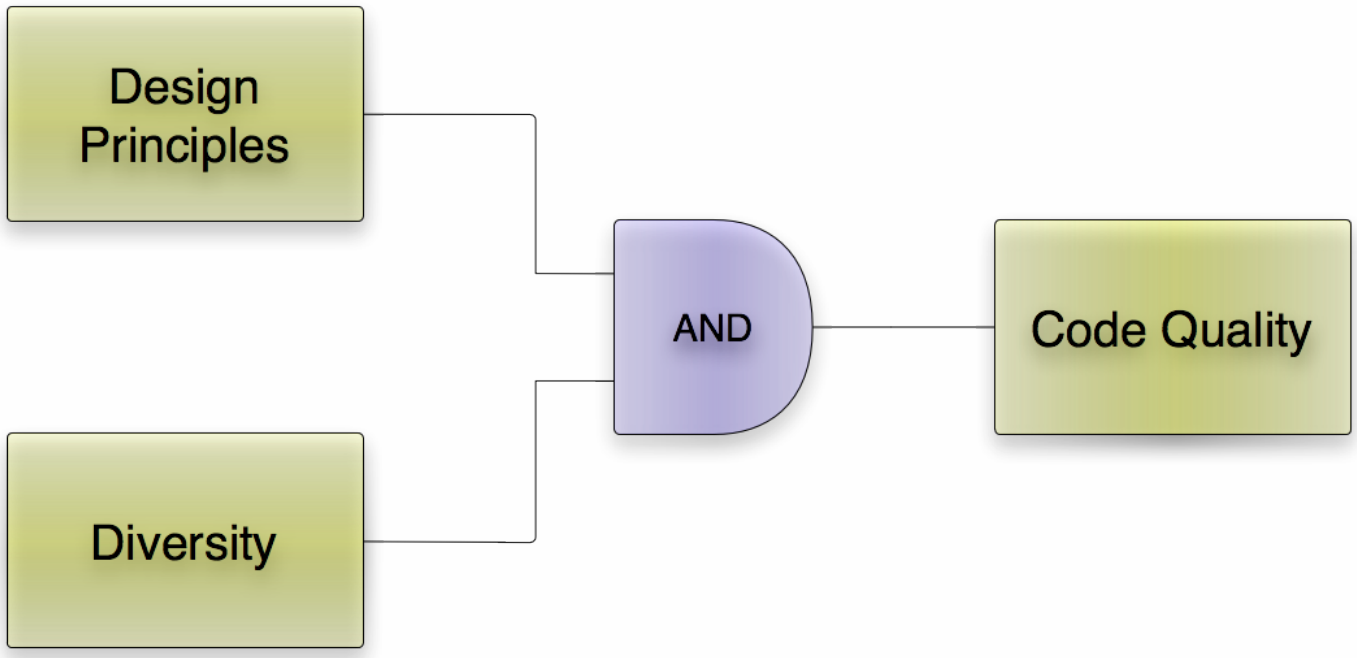
Problem Localization



Outline

1. Advantages of diversity
   - Motivation
   - Code quality
   - Handling the multeity
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

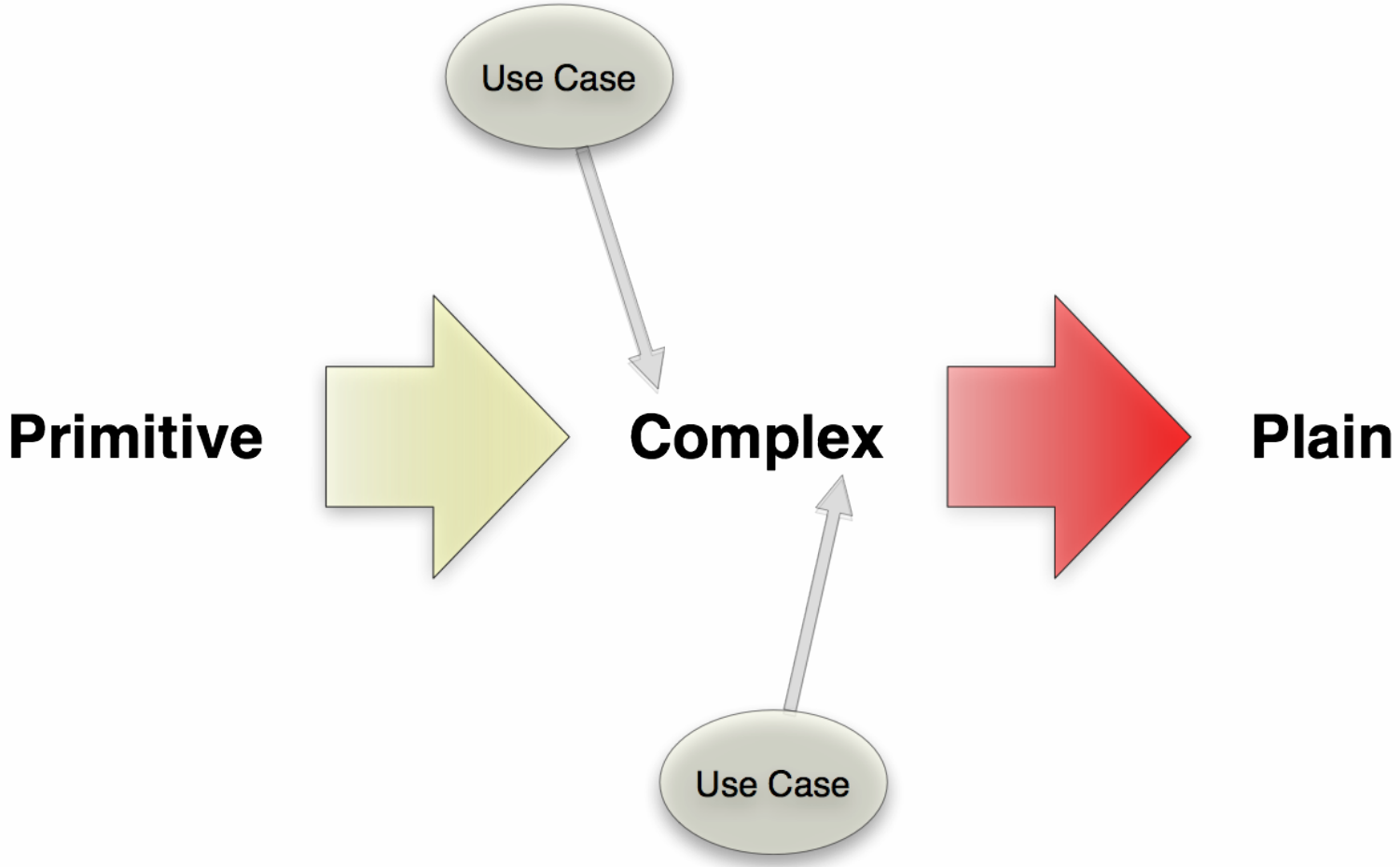
Essential Principles

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

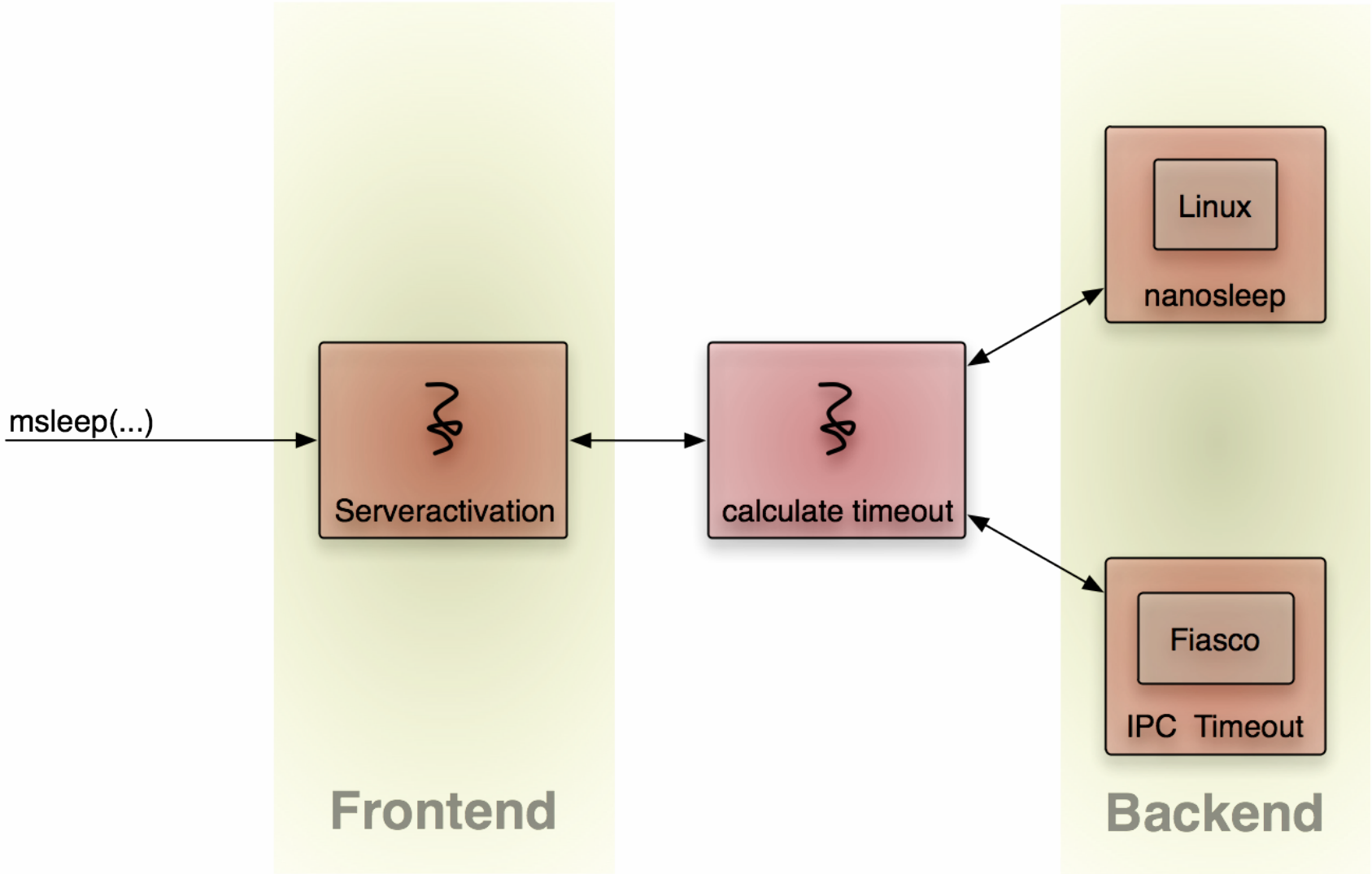
Essential Principles

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

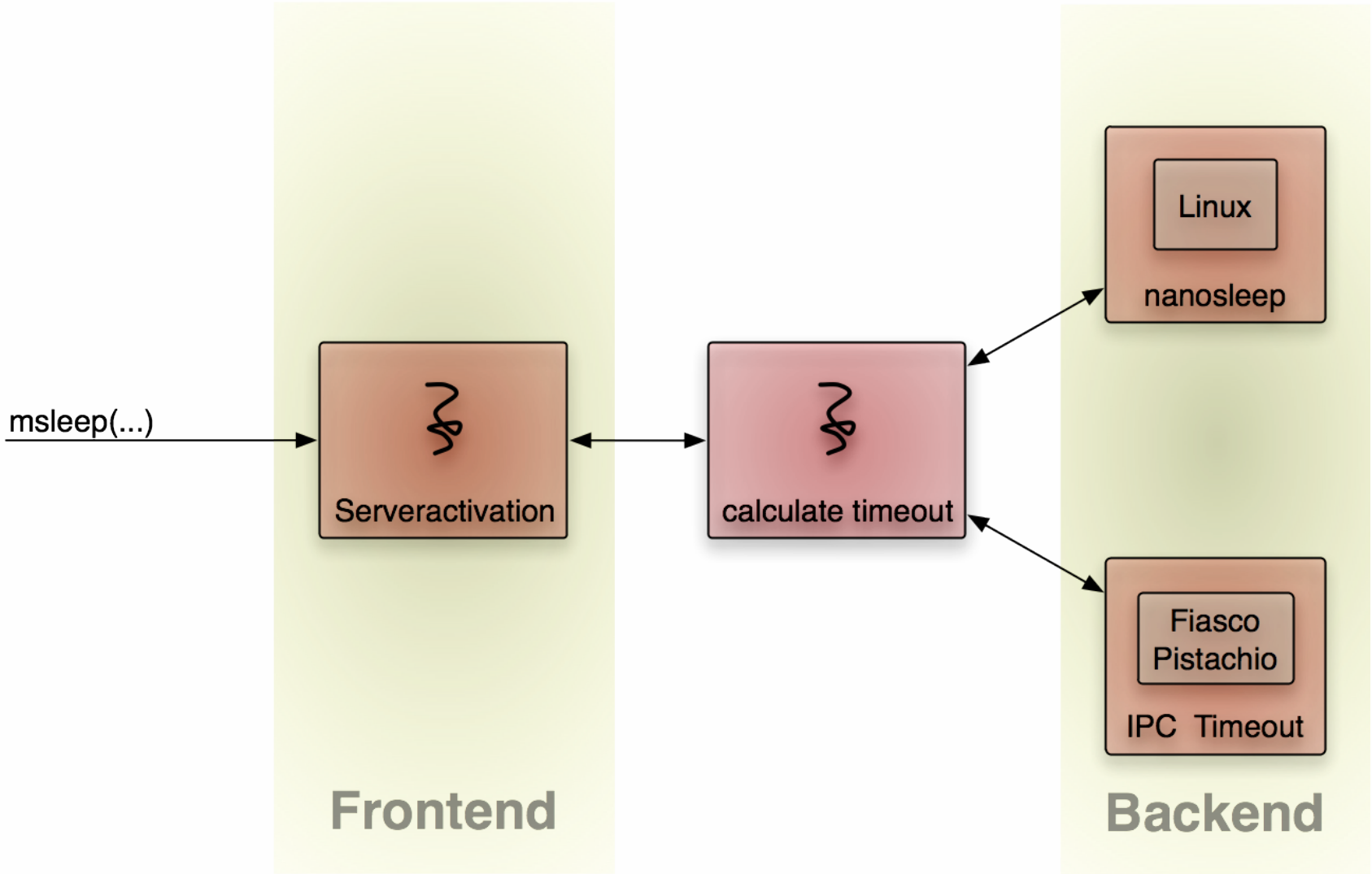
Component's Lifecycle



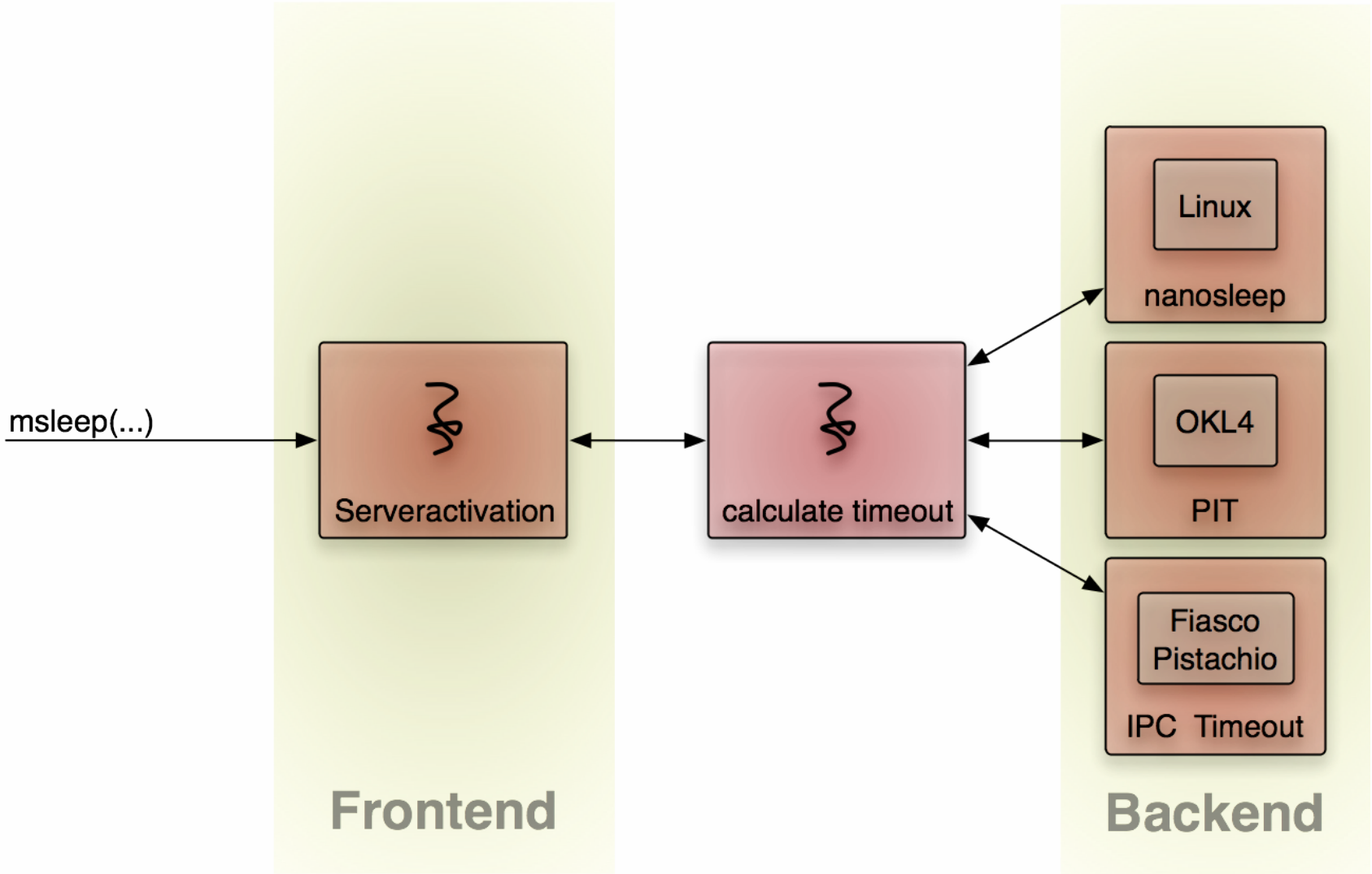
Timer - An Unfinished Example



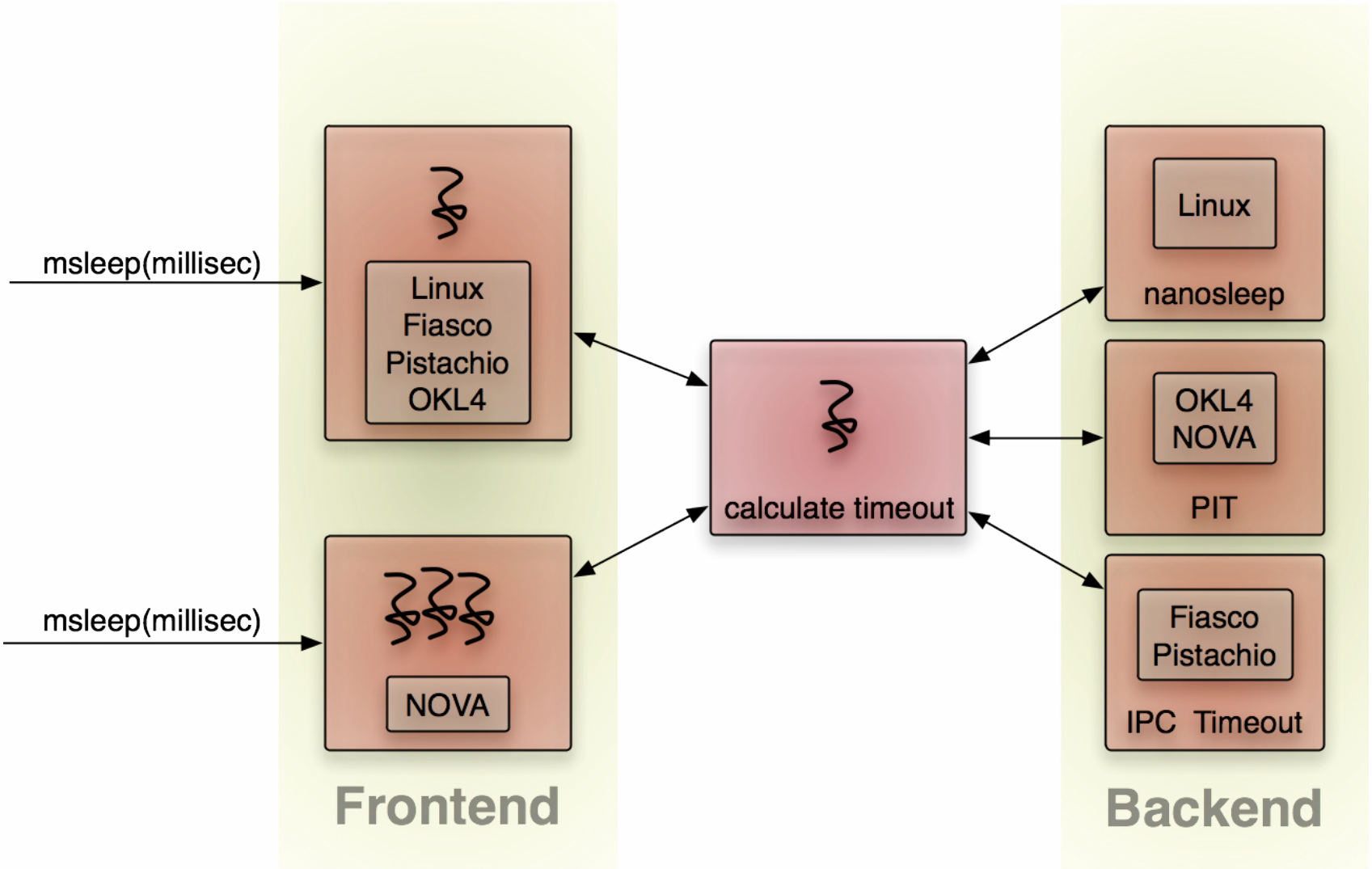
Timer - Introducing Pistachio



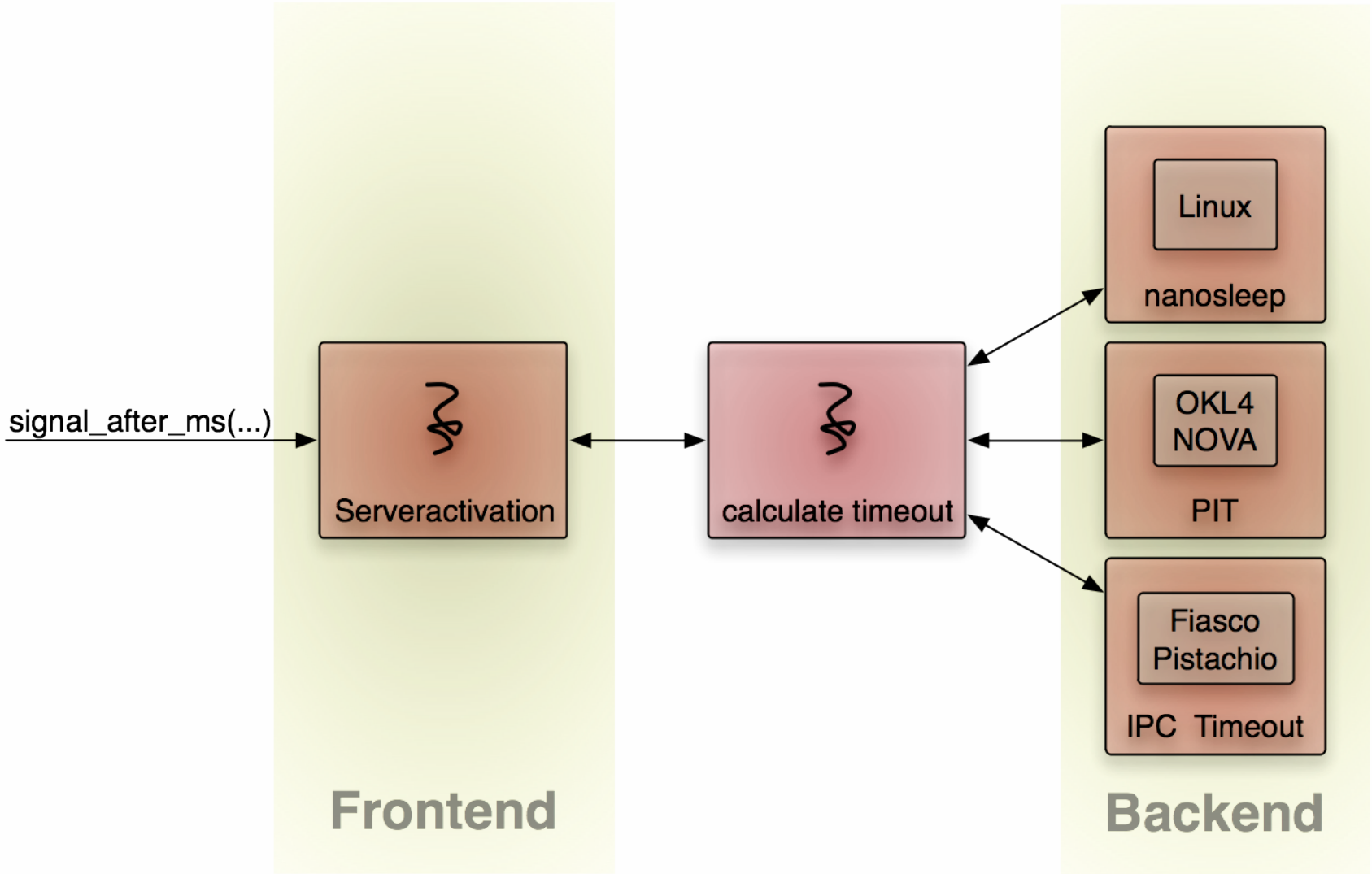
Timer - Introducing OKL4



Timer - Introducing NOVA



Timer - Change API



Outline

1. Advantages of diversity
   - Motivation
   - Code quality
   - Handling the multeity

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

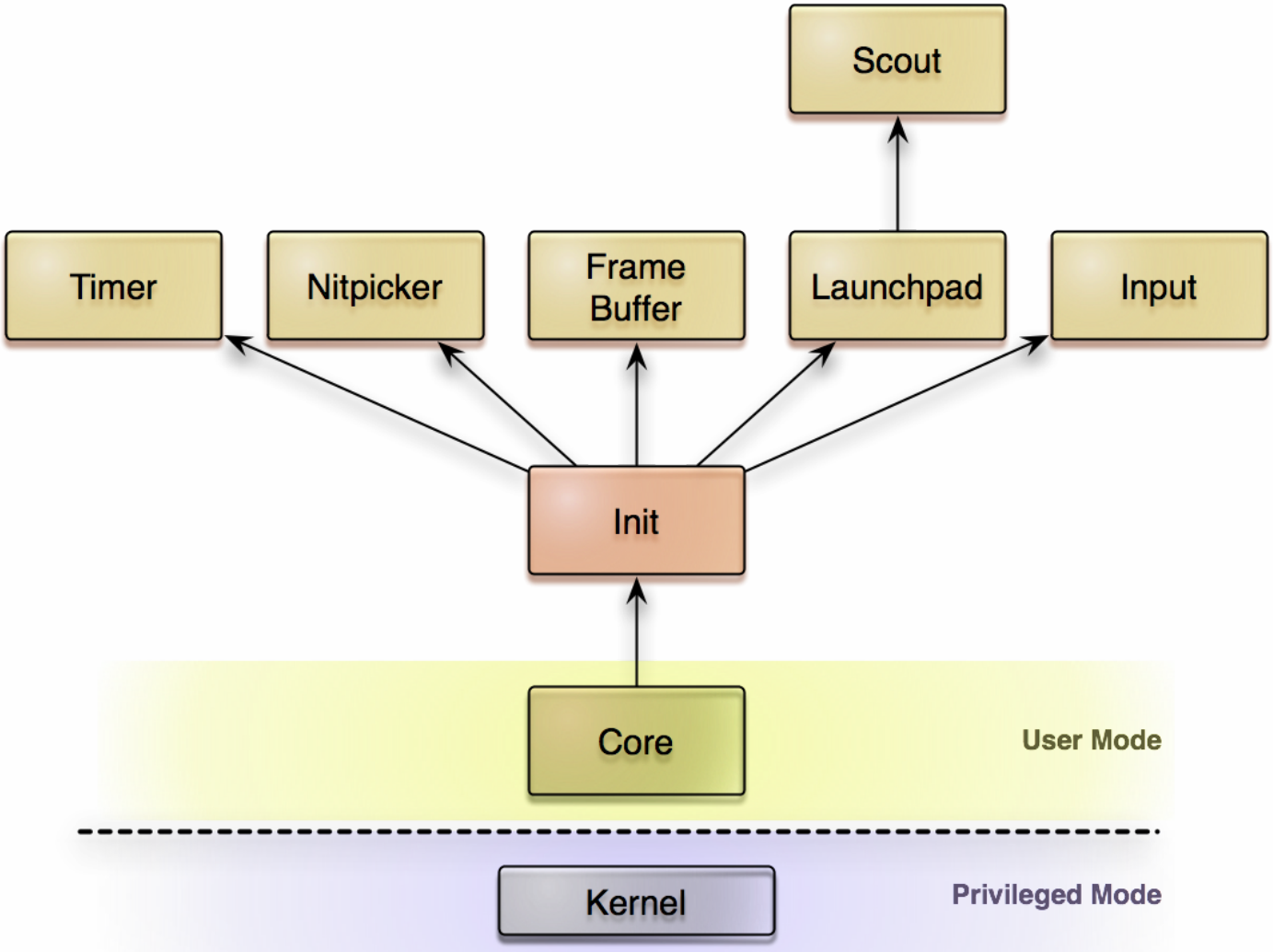
```
#
# Example run-script
#

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

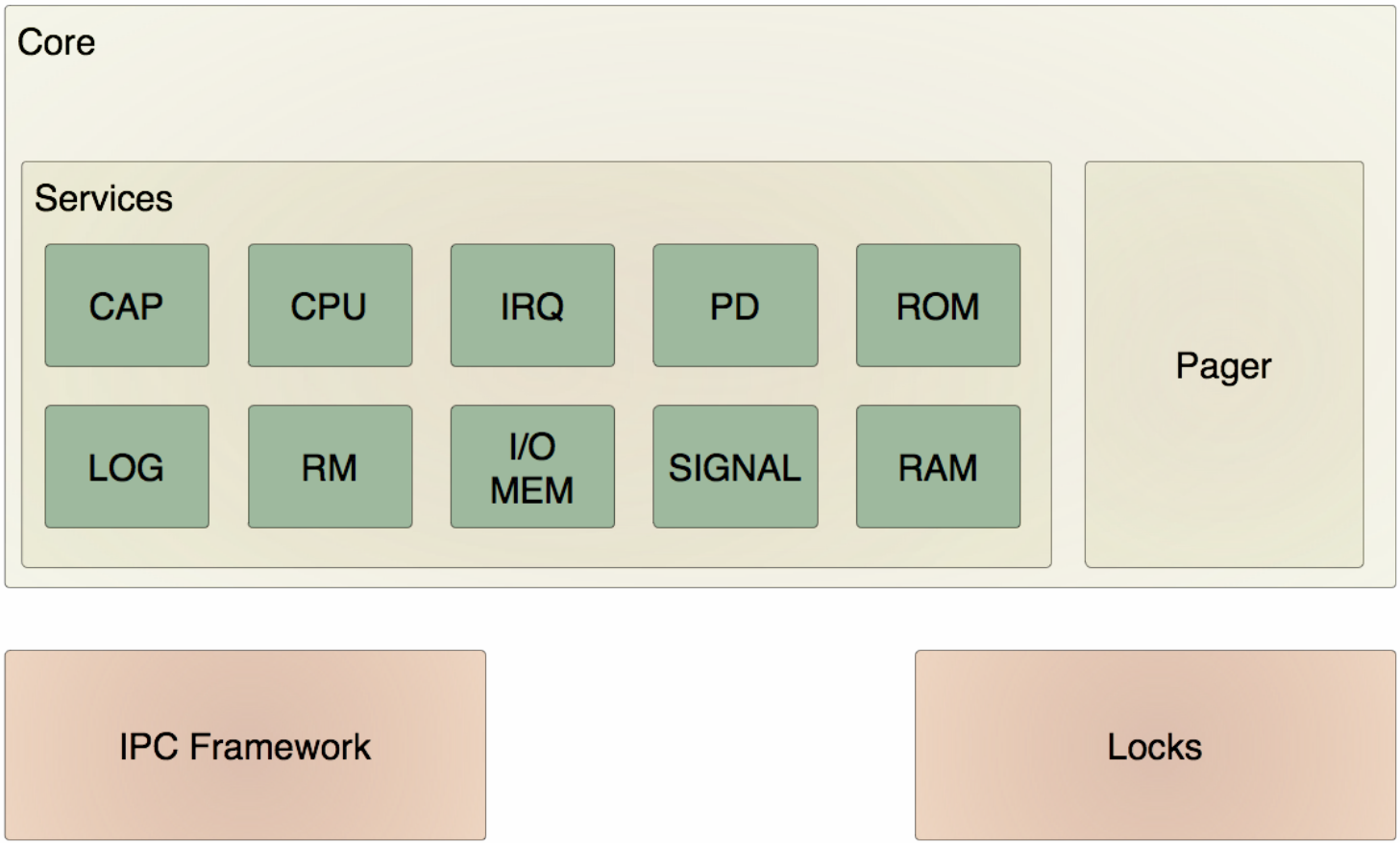
1. Advantages of diversity
   - Motivation
   - Code quality
   - Handling the multeity

2. Porting Genode to a kernel
   - What is needed
   - Course of action
   - Conclusion

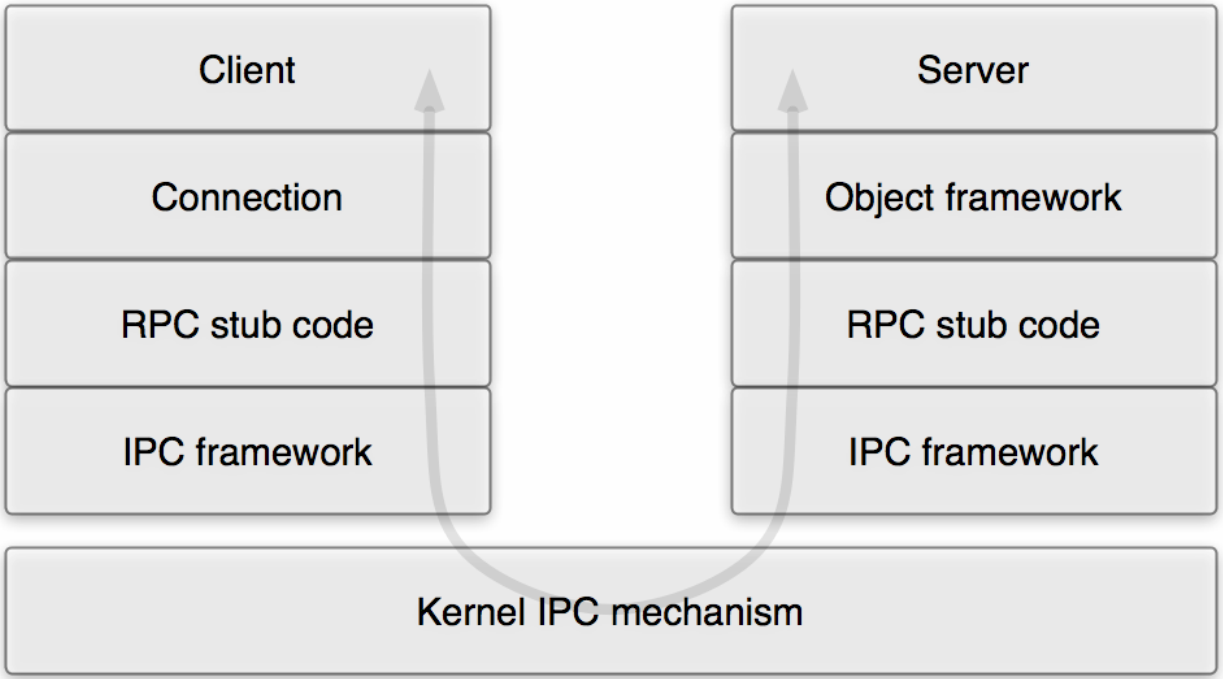
Goal


Scout
Timer
Nitpicker
Frame Buffer
Launchpad
Input
Init
Core
User Mode
Kernel
Privileged Mode

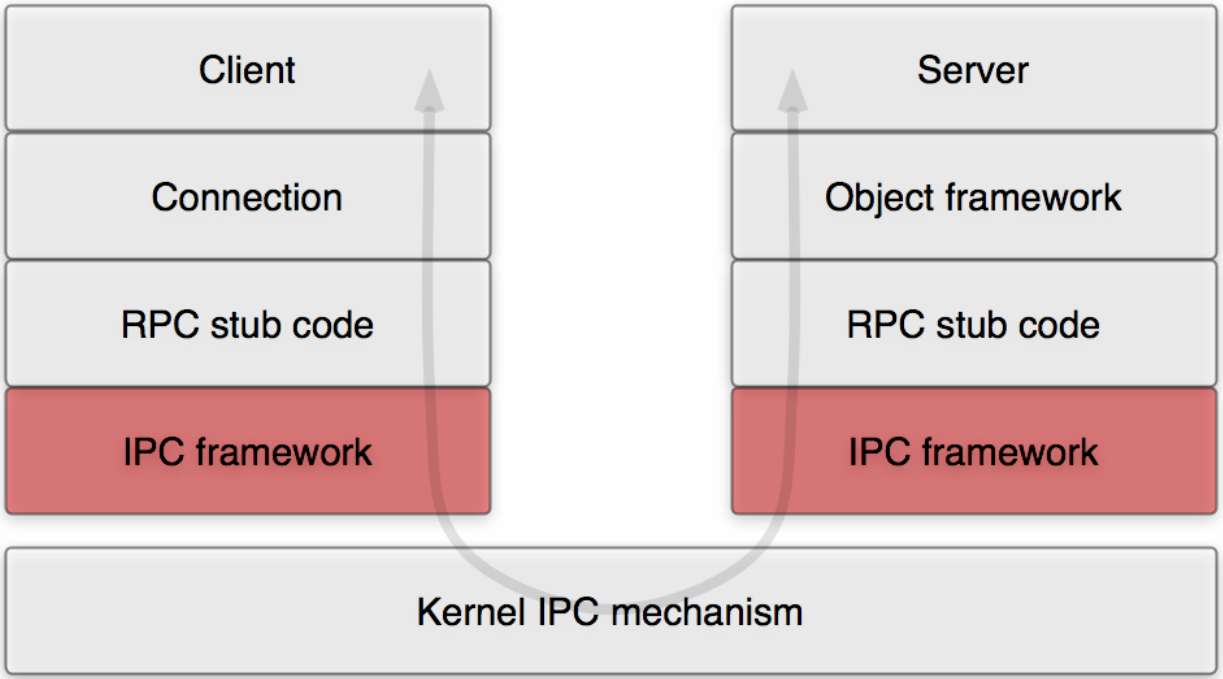
Kernel-Specific Parts



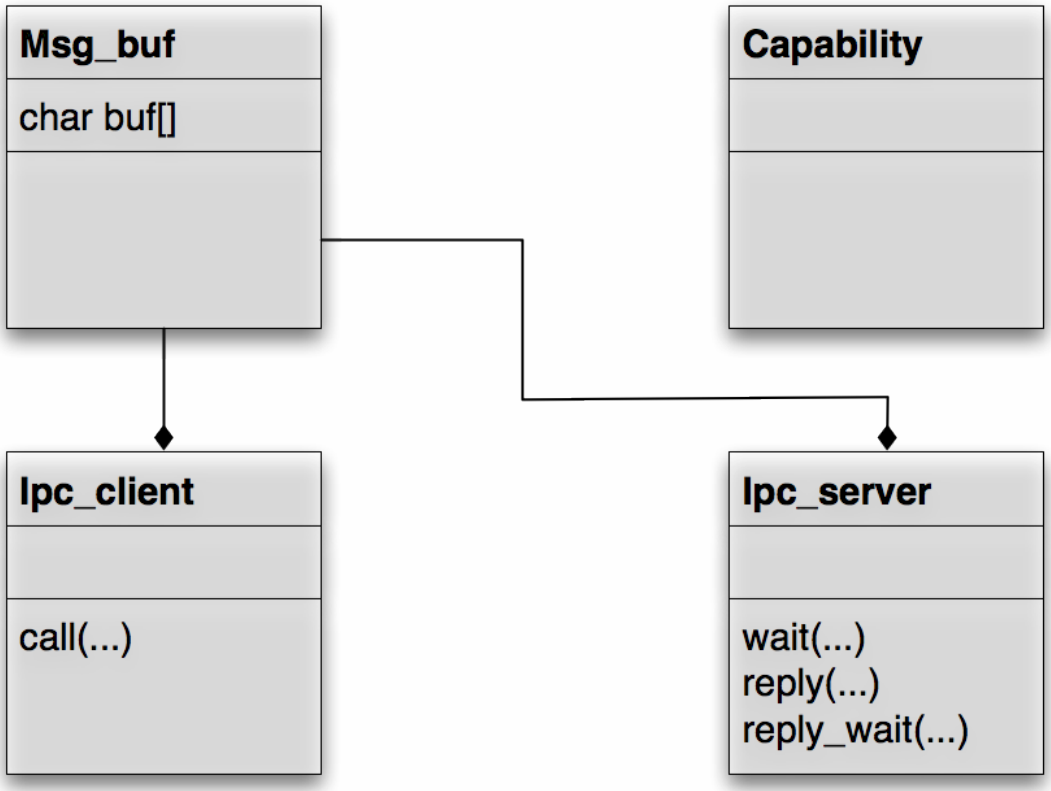
IPC Framework



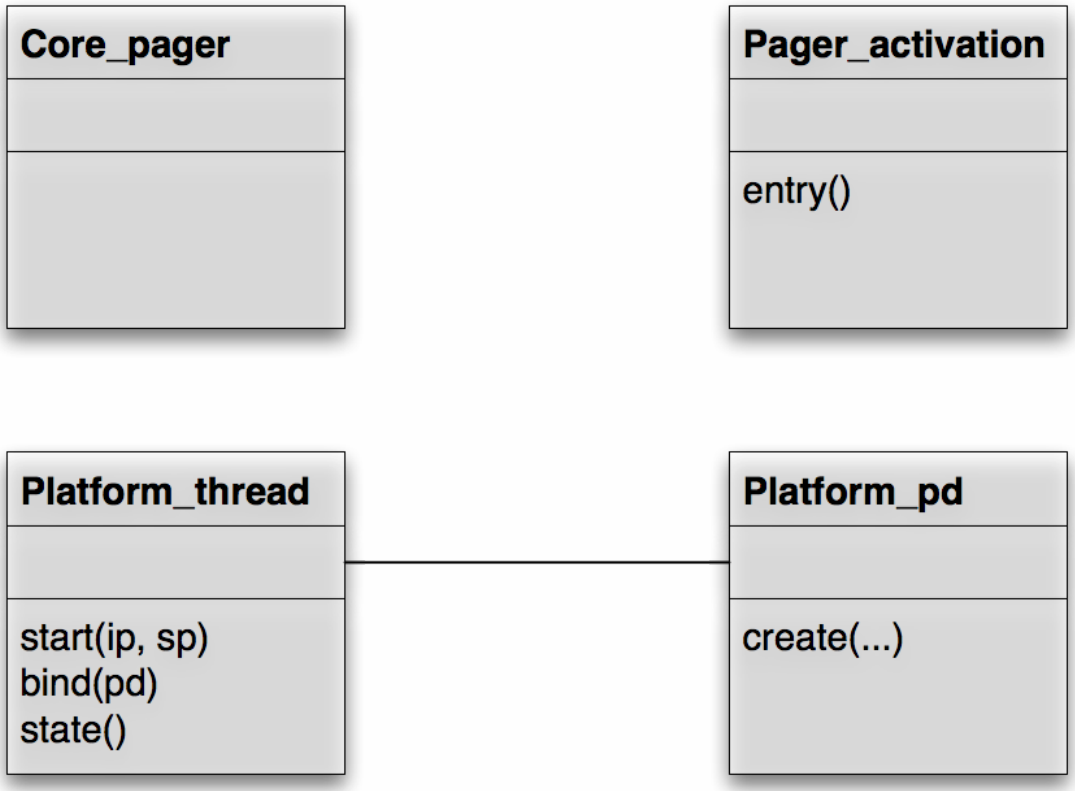
IPC Framework



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

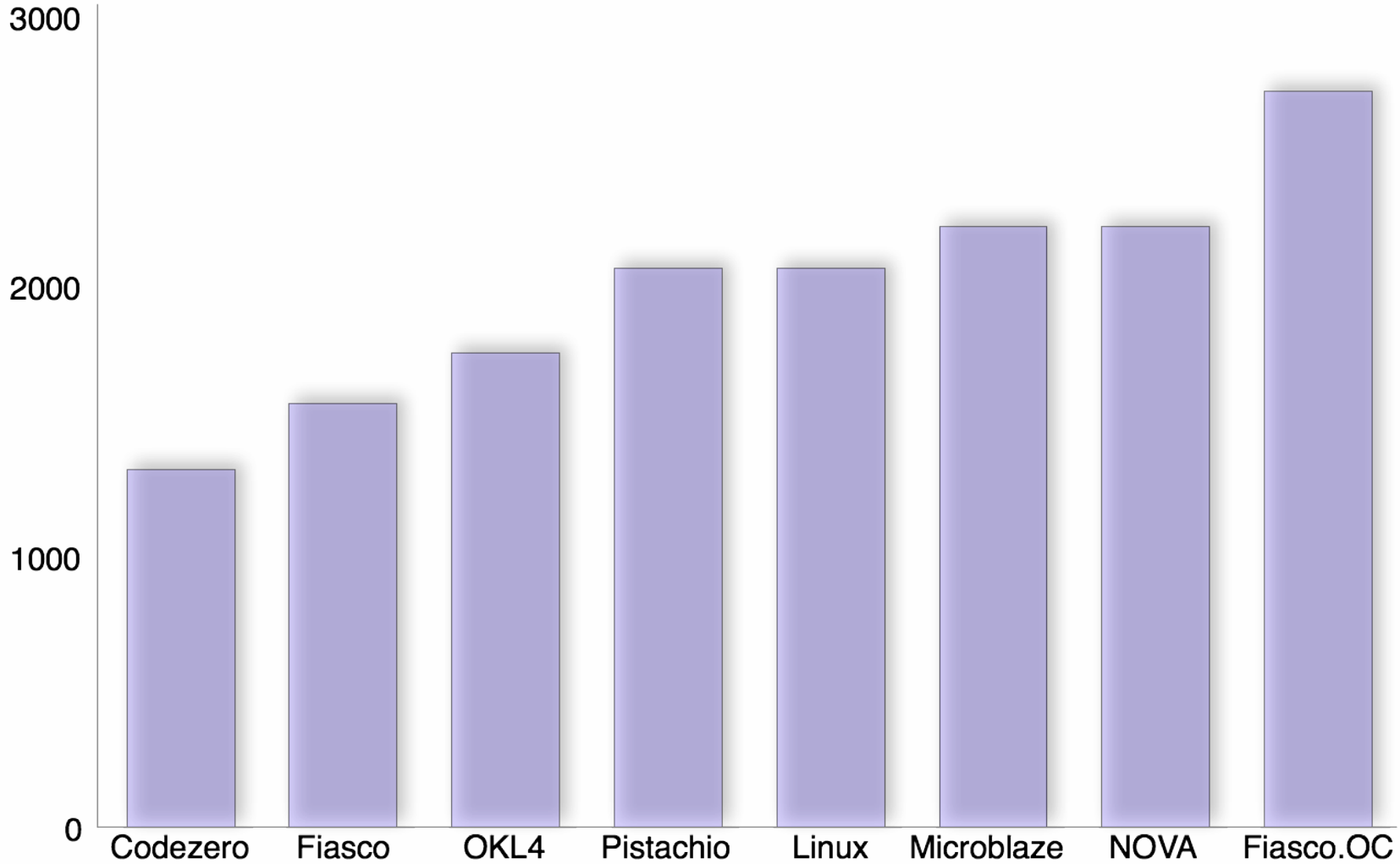
Platform Information

- 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?