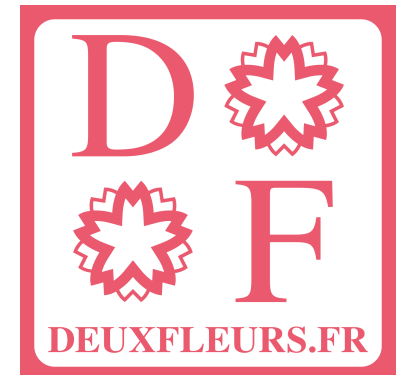


# A microkernel-based orchestrator for distributed Internet services?

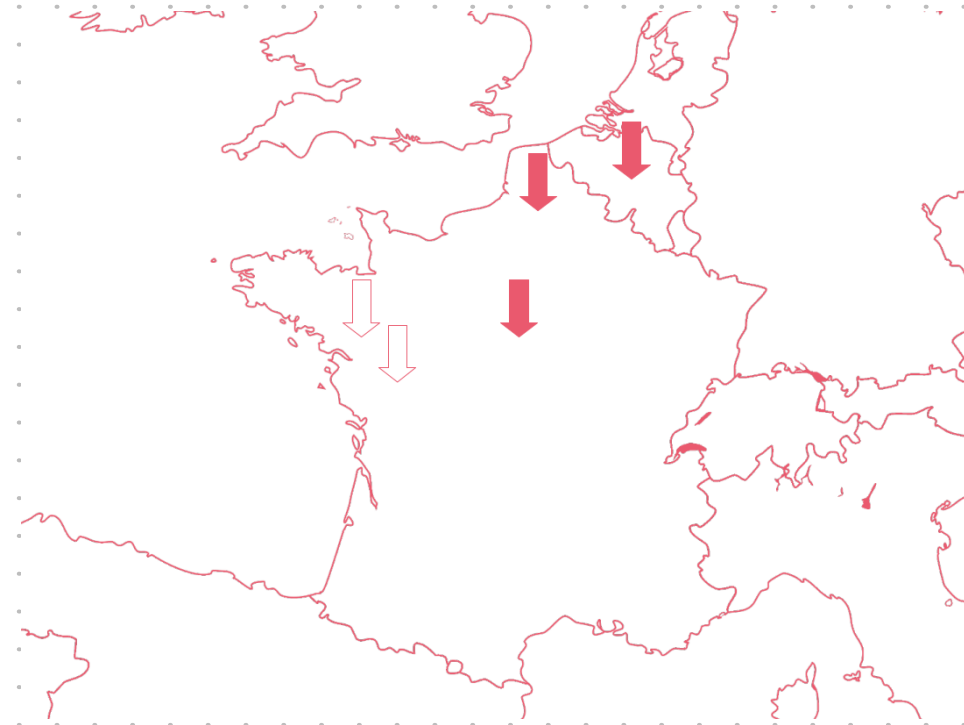
Alex Auvolat <[alex@adnab.me](mailto:alex@adnab.me)>  
Deuxfleurs <https://deuxfleurs.fr>  
FOSDEM 2024



# Distributed system self-hosting

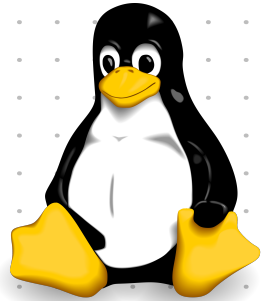


**Low-cost, low-power, second-hand hardware  
running the Deuxfleurs infrastructure**

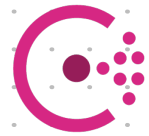


**Servers in several geographical locations  
for redundancy**

# Our current stack



OS



HashiCorp

Consul



Nomad



WIREGUARD



docker

Platform



Garage



PostgreSQL

STOLON

Storage services



Jitsi Meet



CryptPad

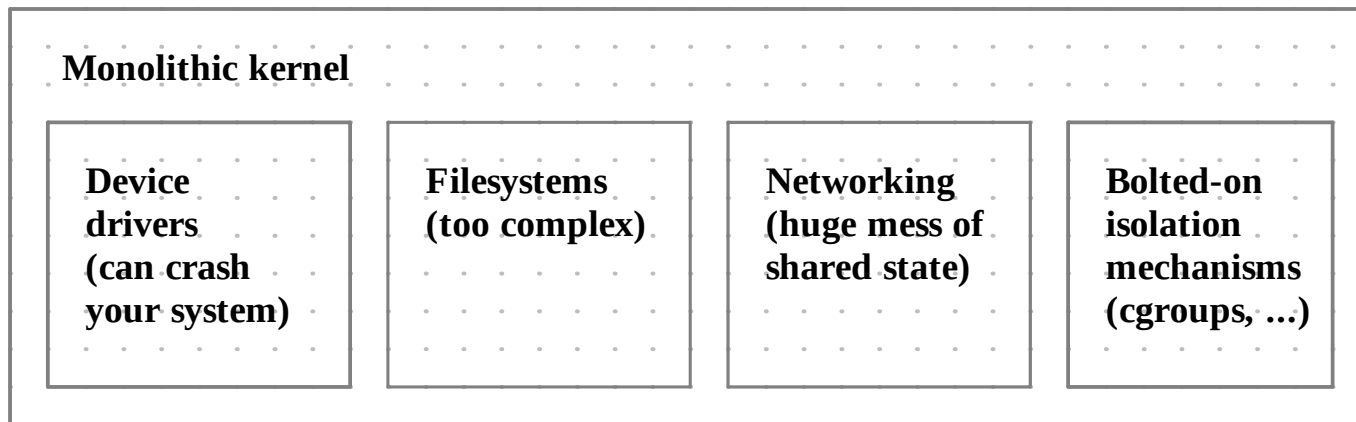
etc...



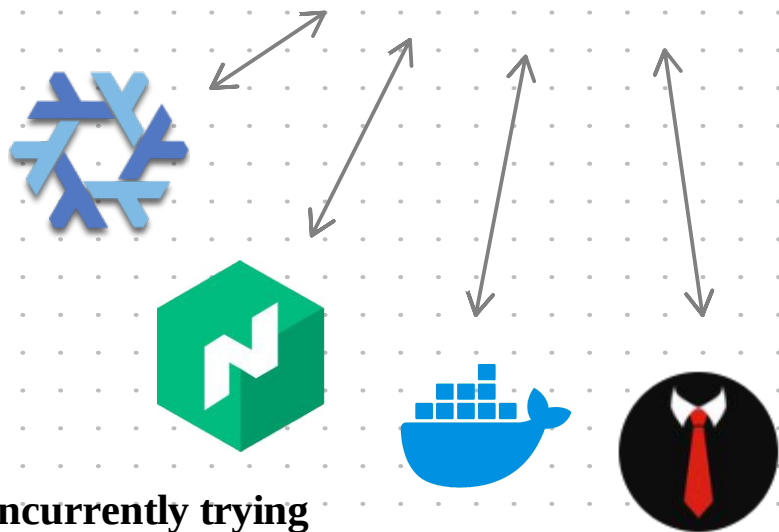
Applications

Distributed operating system?

# Deep dive into this horrible mess

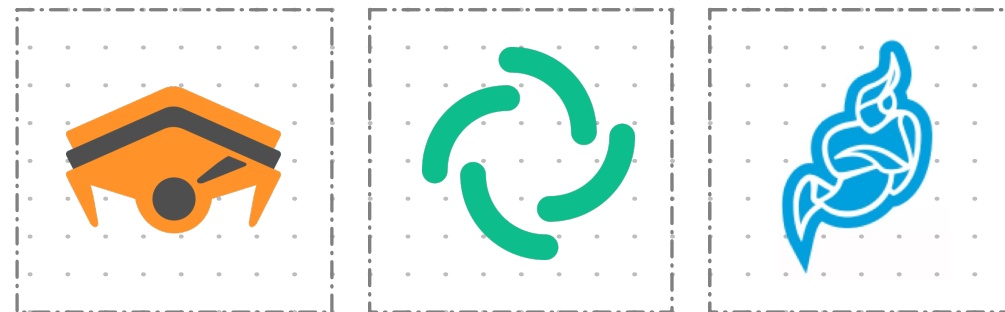


← **All running with highest privilege level on your CPU**



**All concurrently trying to mutate some shared state**

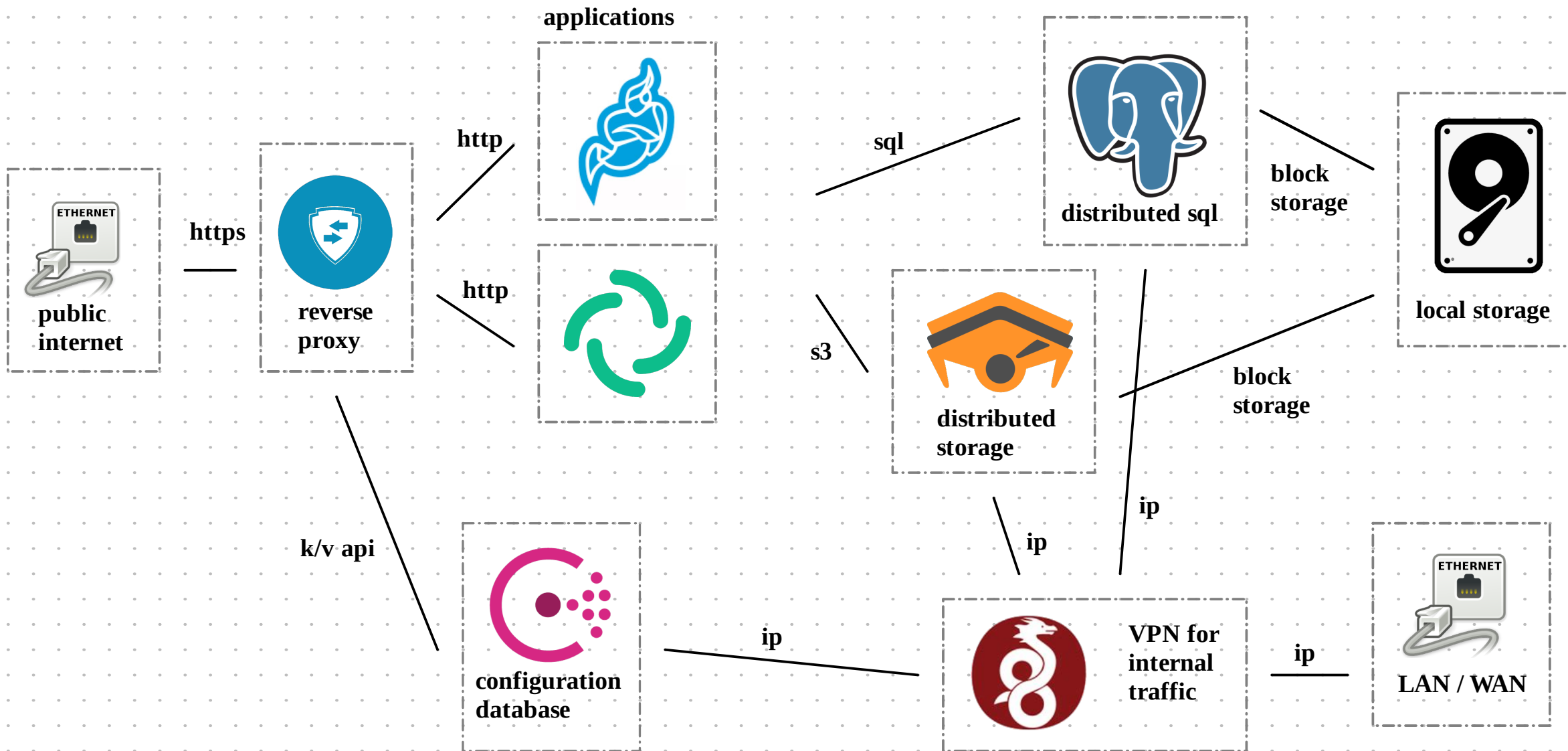
**IP networking (??)**



**Containers (isolation that doesn't really work)**

**Stuff is slow, easily broken, and hard to work on**

# Distributed systems are boxes connected by arrows



# Microkernels would work

**boxes = processes (incl. device drivers)**

**arrows = IPC (of various kinds)**

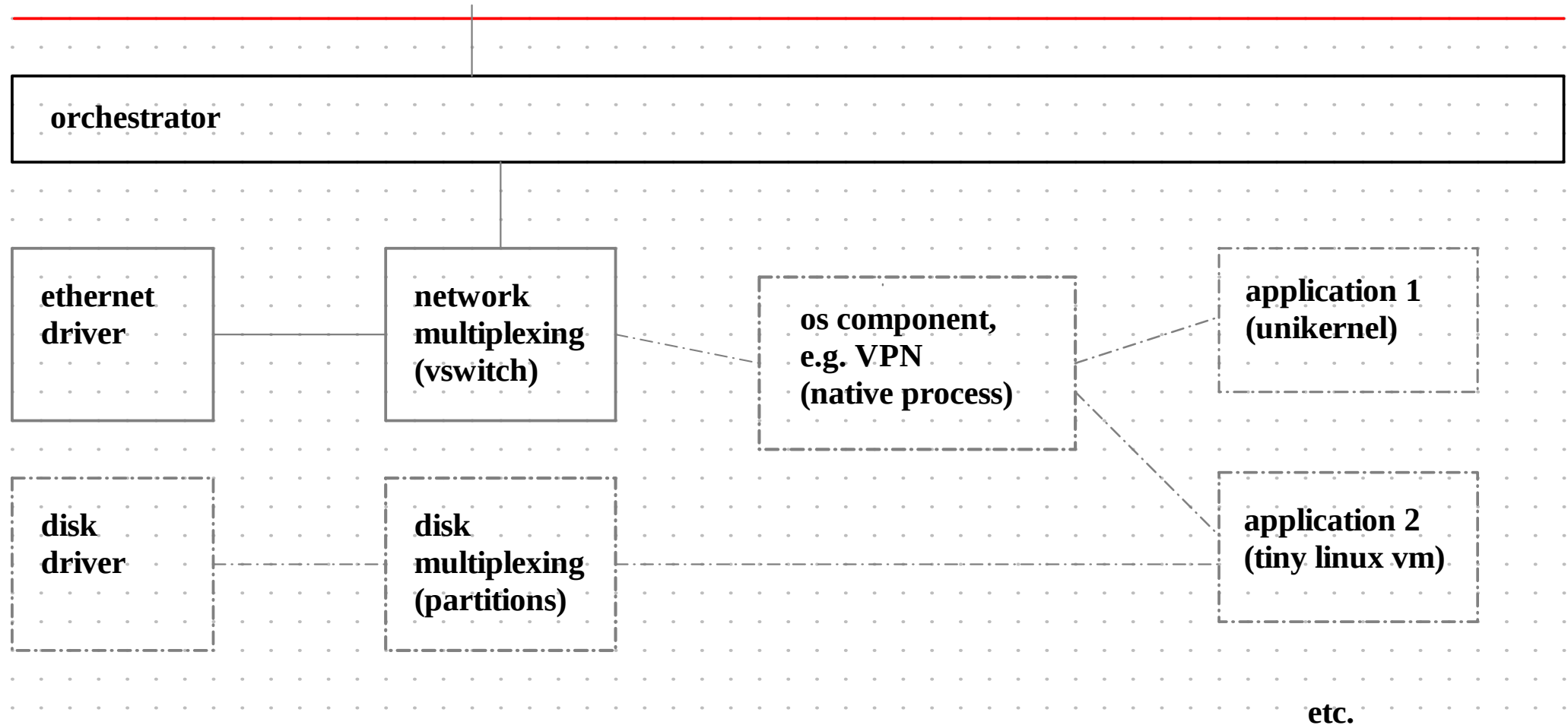
**what microkernels do:**

- **manage processes**
  - > **address space isolation = memory management**
  - > **cpu time sharing = scheduling**
  - > **controlled hardware access**
  - = **multiplexing of fundamental CPU resources**
- **various IPC mechanisms**
  - > **message passing**
  - > **shared memory + semaphores**

**looks like a match !**

# What this could look like

## Microkernel



# What we need

- a way to describe boxes and their connections dynamically  
= an orchestrator
- some standard box and arrow types

## box types:

hardware drivers  
resource multiplexers  
orchestration & mgmt components  
native components  
VM apps (unikernels, tiny linux vms)

## arrow types:

management APIs  
observability (e.g. logging)  
block storage  
networking

- tooling and ecosystem



# Goals and non-goals

## **Main focuses:**

- **declarative configuration**
- **remote management and observability**
- **clustering support & dynamic reconfiguration**
- **I/O performance (async I/O all the way)**
- **simplicity & minimalism**

## **Non-goals:**

- **POSIX API compatibility**
- **desktop operating system**

# Leveraging the existing

## microkernels

L4 / seL4

NOVA

Fuschia

...

## I/O models

VirtIO

io\_uring

9P

...

## frameworks & OSes

Genode

Redox OS

GNU Hurd

...

Where to start from?