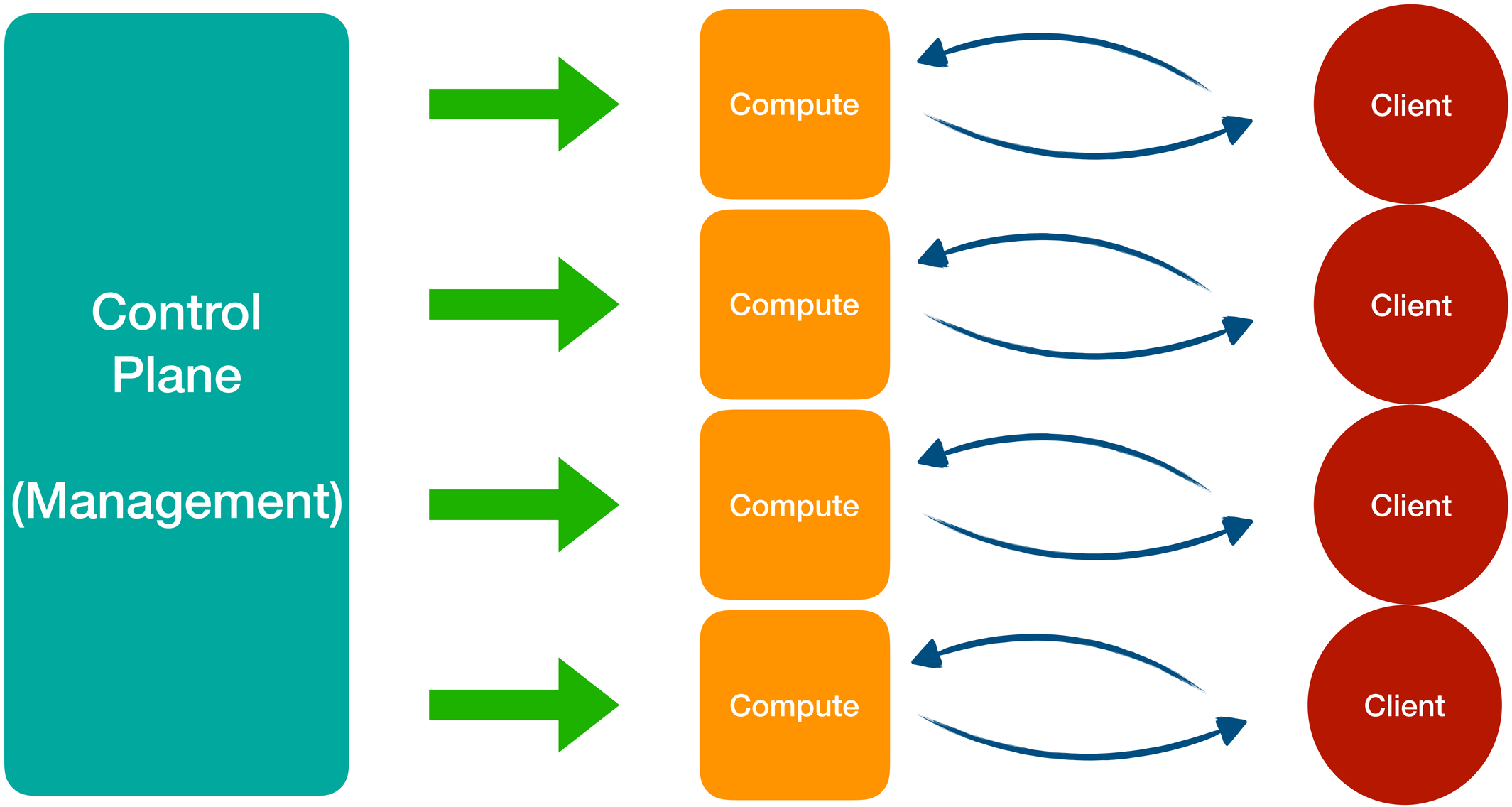
Why don’t our computers have
a separate control plane?


Control
Plane
(Management)
Compute
Compute
Compute
Compute
Client
Client
Client
Client

Why have we granted computers the power to modify themselves?

Unikernels aren’t dead

They’re just not containers.

Per Buer

CEO IncludeOS
(Varnish, Linpro)

What is a Unikernel and how does it behave?

Where can this be applied?

Experiences.

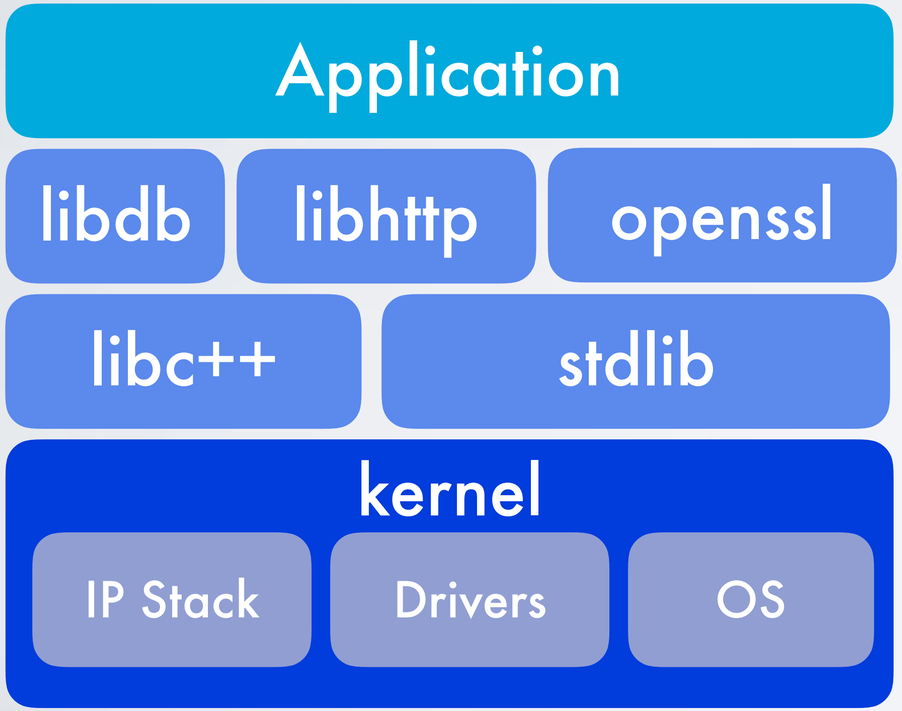
Unikernel primer

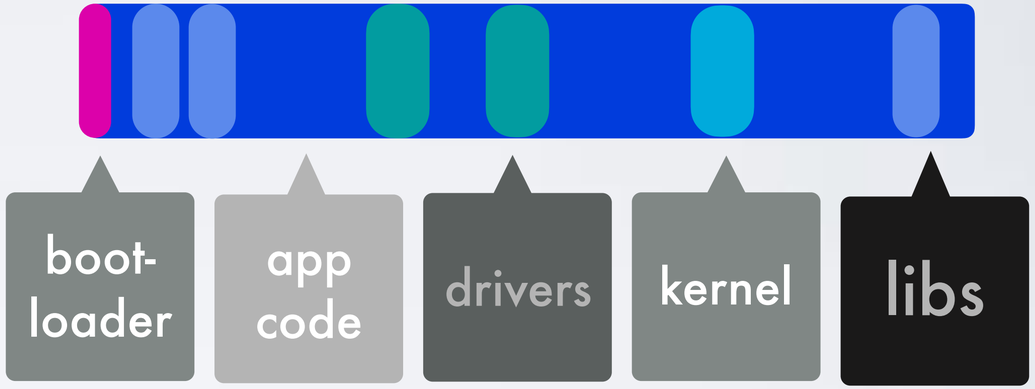
Make the OS a library

Link the OS
into application

Add bootloader, hw init ++

Traditional vs unikernel application





includeOS

Underlying system
-
Bare metal or Virtual Machine

Hello world.

Unikernel Characteristics

Fundamental Immutability

Predictable behavior

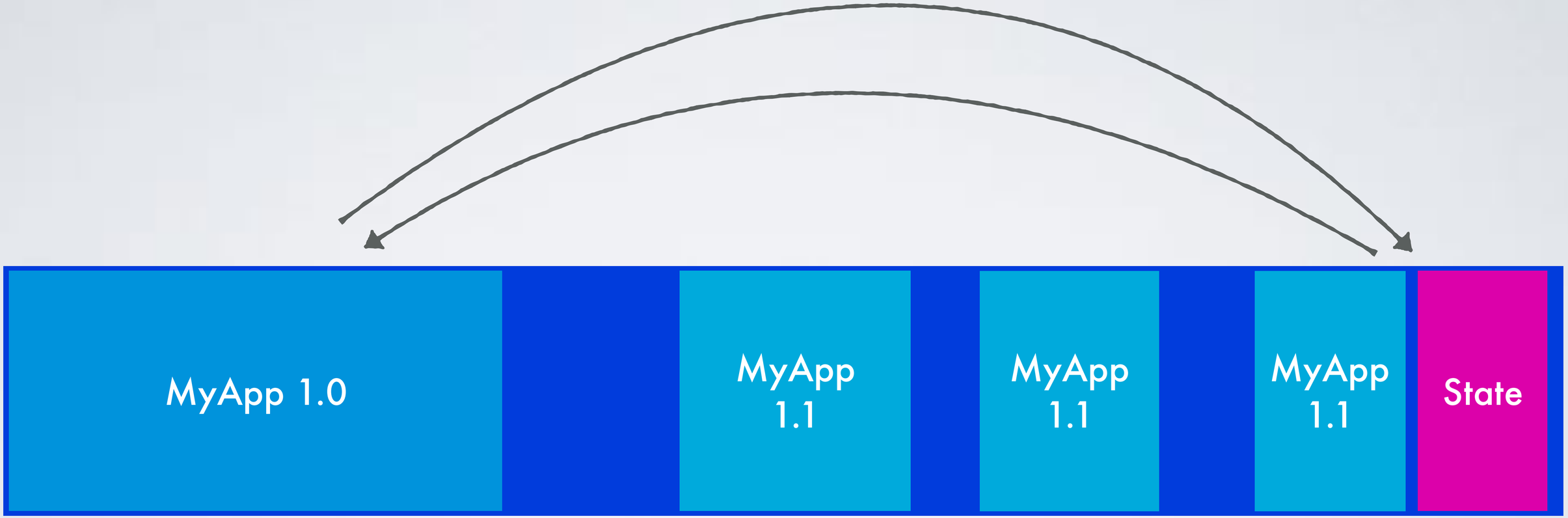
Self contained
&
simple

Limited compatibility
&
limited runtime

IncludeOS specifics and experiences

C++

Multicore



Memory

includeOS

Configuration

Data

Program code
(another vendor)

Local adaptations

Operating system environment
(vendor supplied)

includeOS

NaCl

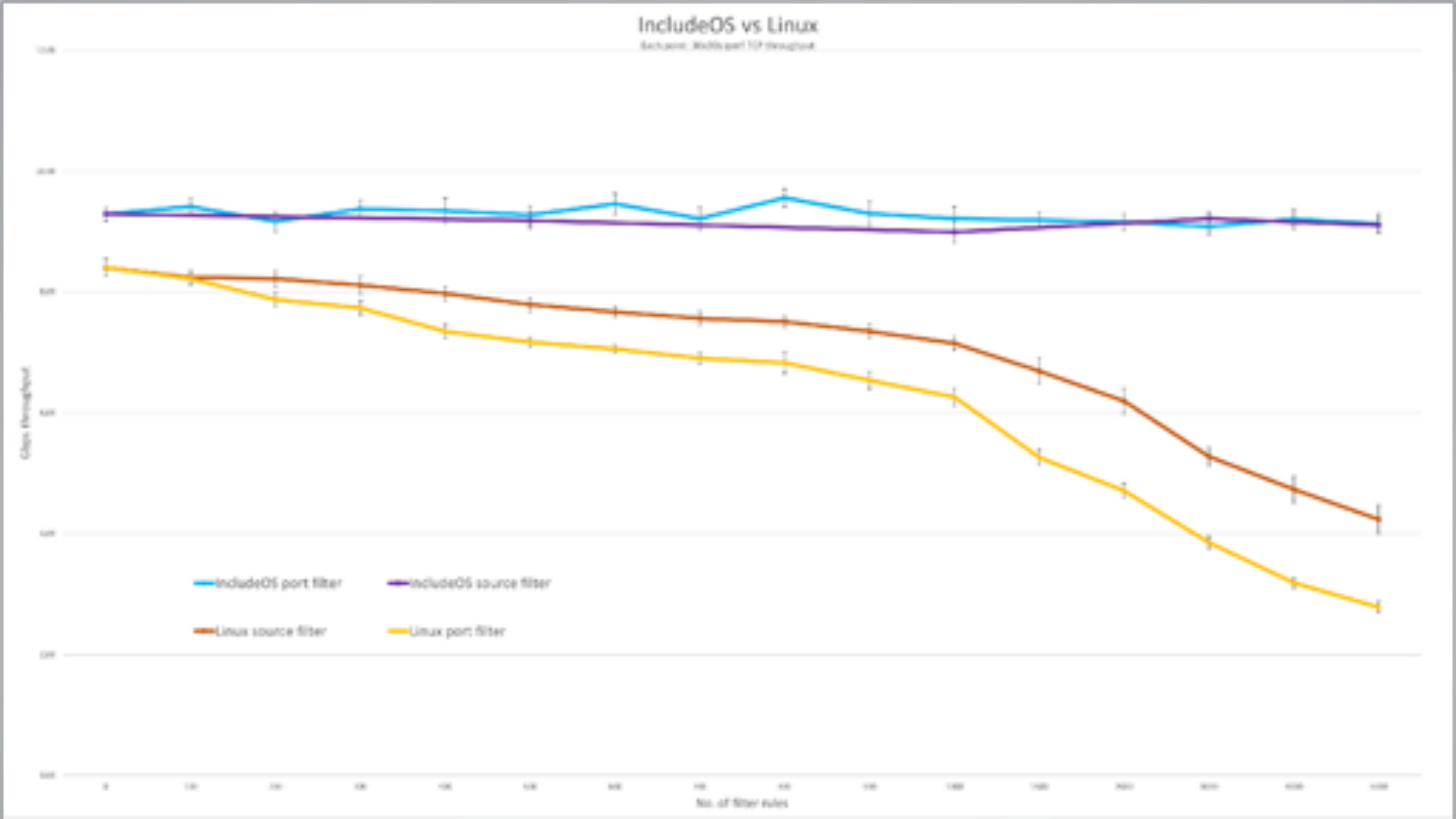
An alternative to configuration files.

```
bastion_host: 10.55.1.1
allowed_services: [80,443]
allowed_hosts: [192.168.0.10 - 192.168.0.200]

Filter::IP firewallchain {
    if (ct.state == established) {
        syslog(INFO, "Packet from est. conn. (src: ", ip.saddr, ")")
        accept
    }

    if (ip.saddr == bastion_host) {
        accept
    }
    Filter::TCP {  // typecast
            if (ip.daddr in allowed_hosts and tcp.dport in allowed_services) {
                    syslog(INFO, "New conn (from ", ip.saddr, ")")
                accept
            }
    } // end TCP

    syslog(WARNING, "Dropping packet from saddr ", ip.saddr," to ", ip.daddr)
    drop
}
```


IncludeOS vs Linux
Gbps throughput
IncludeOS port filter
IncludeOS source filter
Linux source filter
Linux port filter
No. of filter rules

Performant.

Simple.

We’re effectively run-time less.

When are Unikernels relevant?

Secure.

Contained

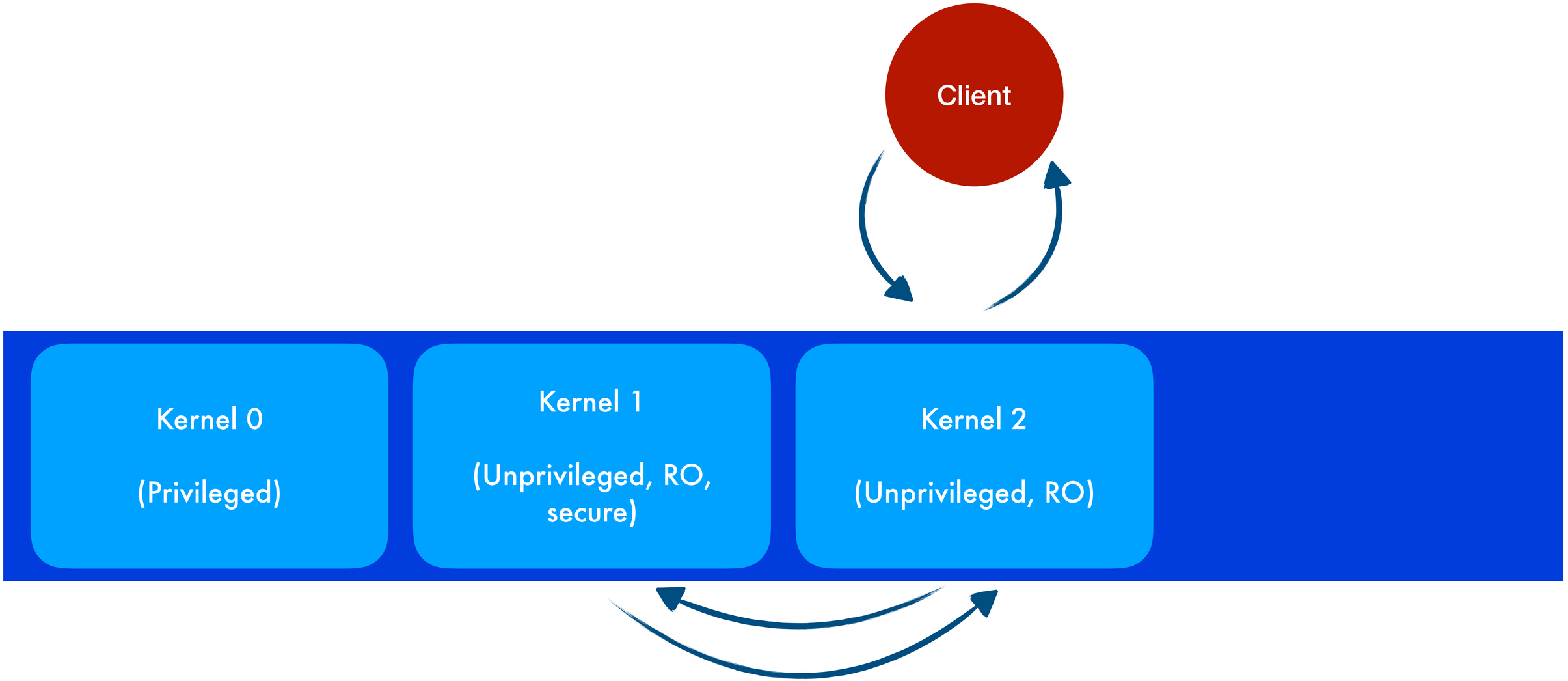
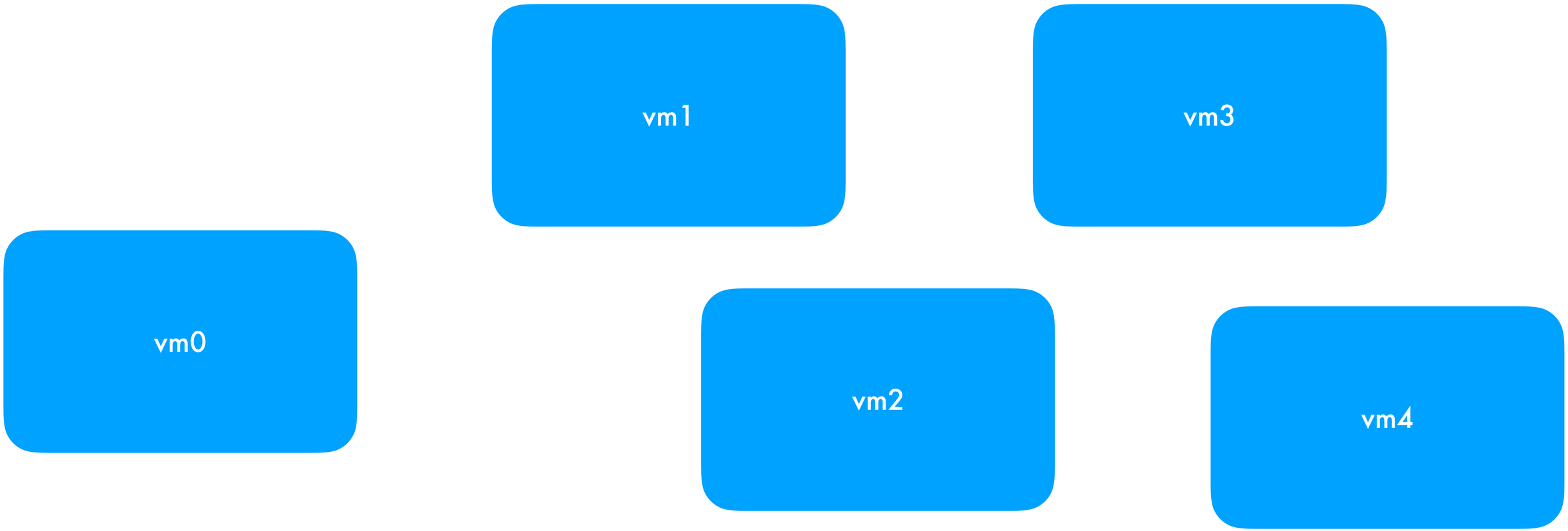
Predictable

Performant

No kernel/user boundary

Blue sky stuff

Shadowkernels



Distributed systems