

RUMP KERNELS and {why,how} we got here

*New Directions in Operating Systems
November 2014, London*

Antti Kantee, Fixup Software Ltd.
pooka@rumpkernel.org
@anttikantee

Motivations

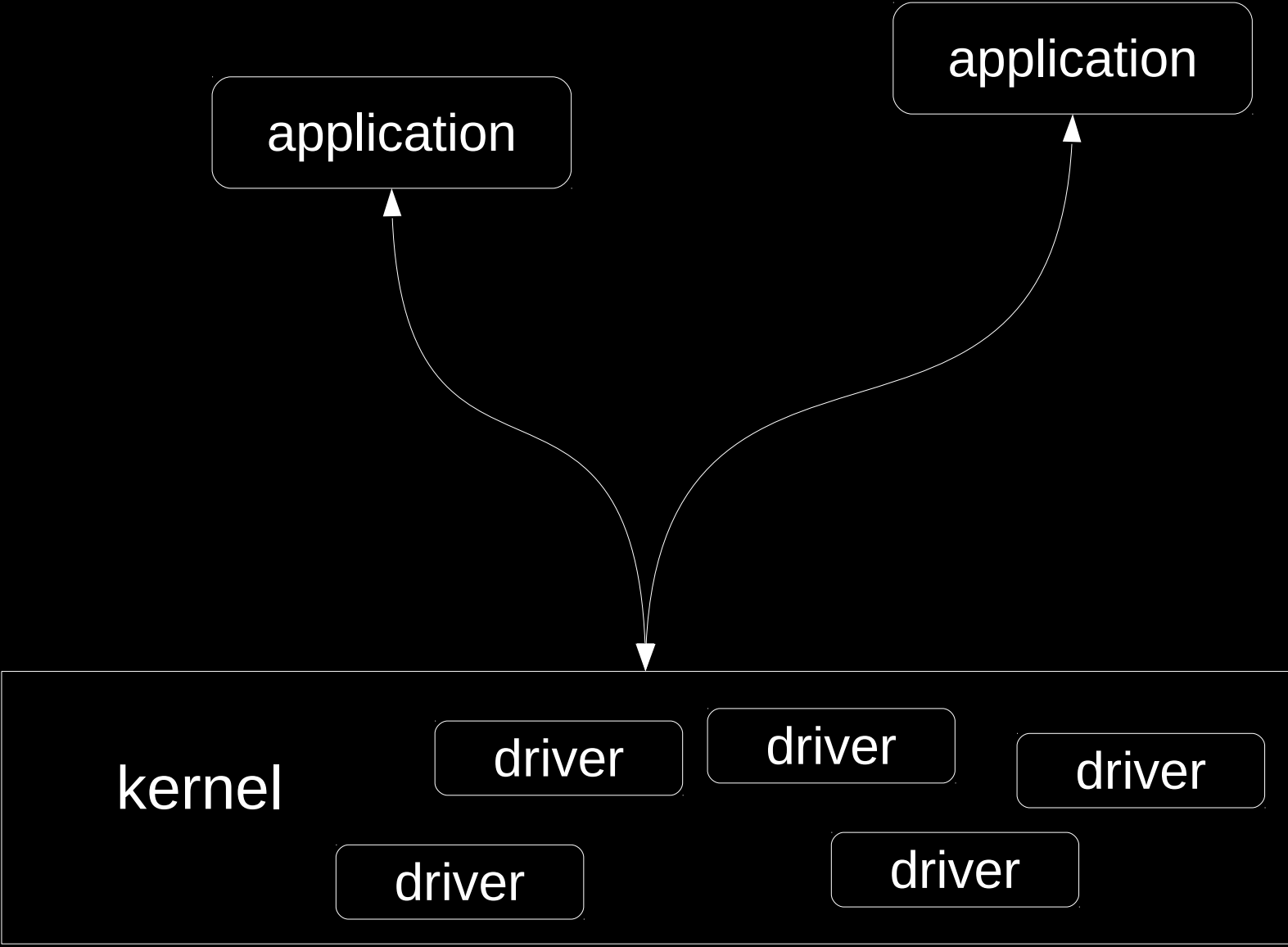
- want to run an application, not an OS
- want a better operating system
- “operating system gets in the way”

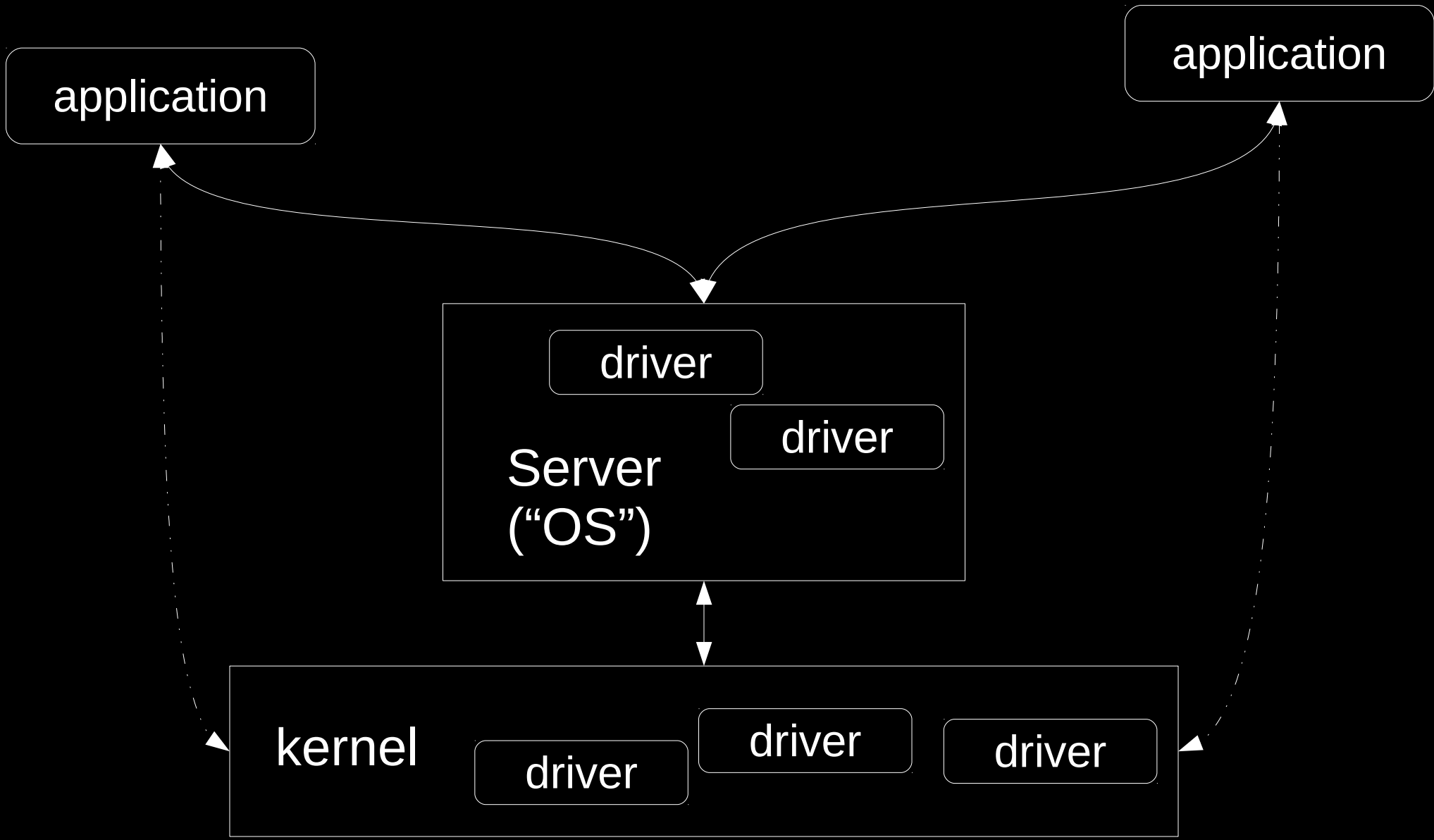
FIRST HALF

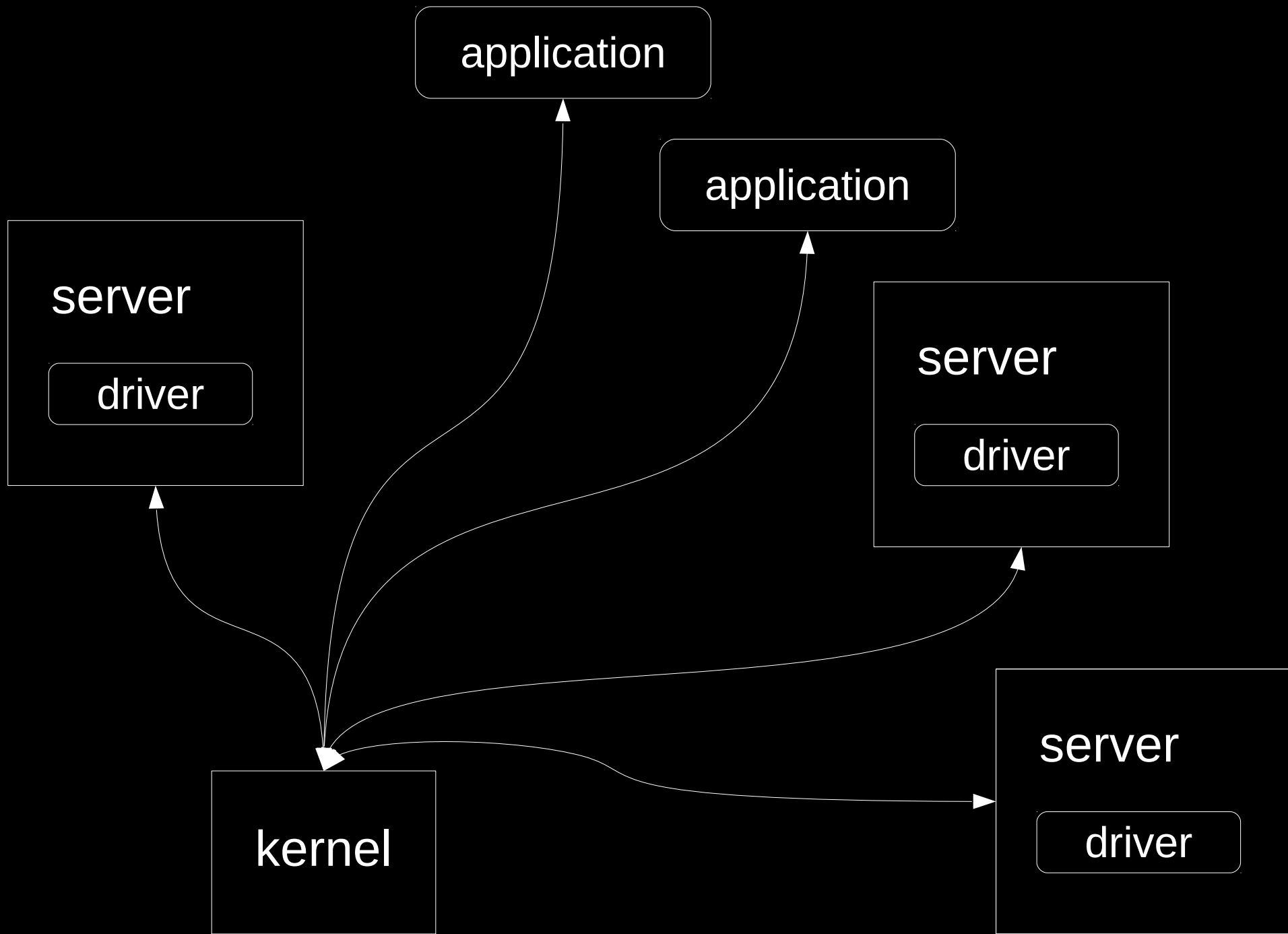
what is an operating system

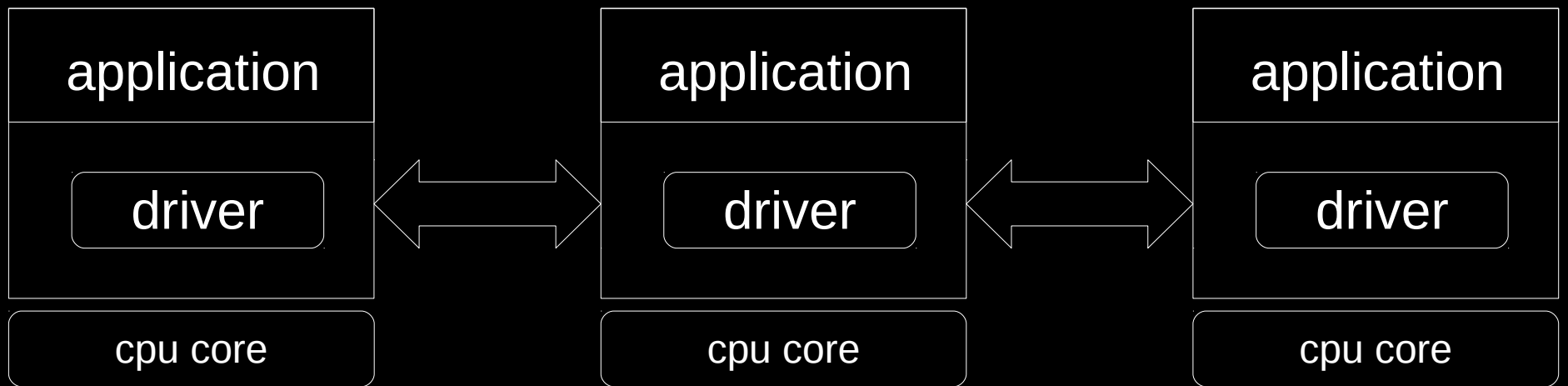
Summary of OS's

- drivers
 - for enabling applications to run
 - $n \cdot 10^6$ LoC
- optional goop defining relation between drivers and applications
 - for protection, resource sharing, ...
 - $10^3 - 10^5$ LoC









application

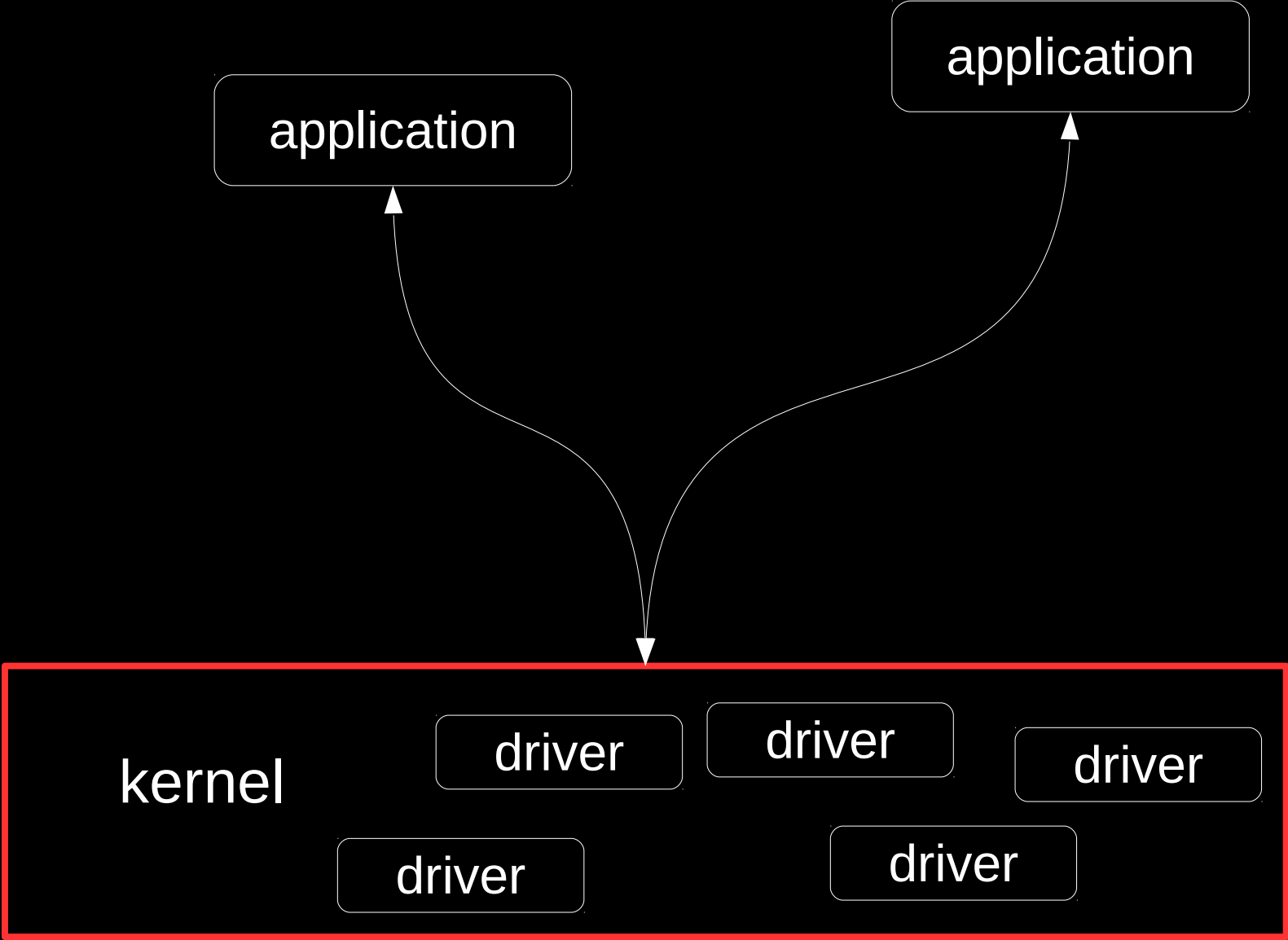
driver

driver

driver

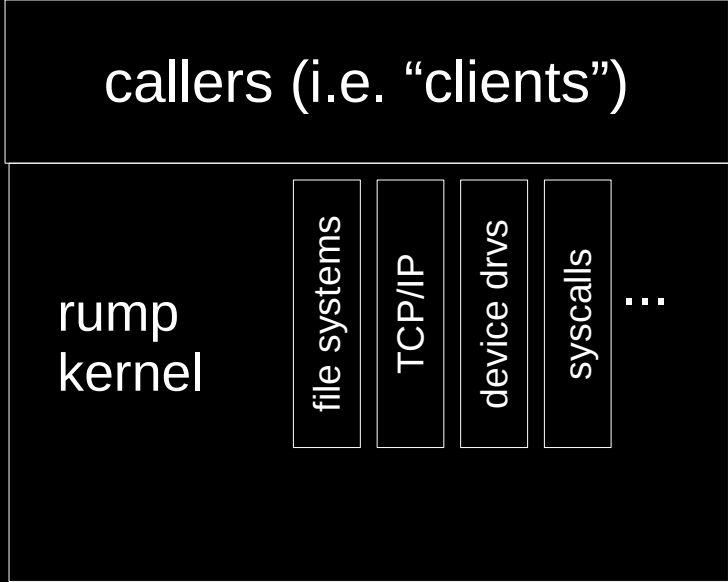
driver

kernel



SECOND HALF

what is a rump kernel



hypercall interface

hypercall implementation

platform

rump (n):

small or inferior remnant or offshoot; especially: a group (as a parliament) carrying on in the name of the original body after the departure or expulsion of a large number of its members

rump kernel (n):

small or inferior remnant or offshoot; specifically: a monolithic OS kernel carrying on in the name of the original body after the departure or expulsion of a large number of its subsystems

A rump kernel does not provide threads, a scheduler, exec, or virtual memory, nor does it require privileged mode (or emulation of it) or interrupts

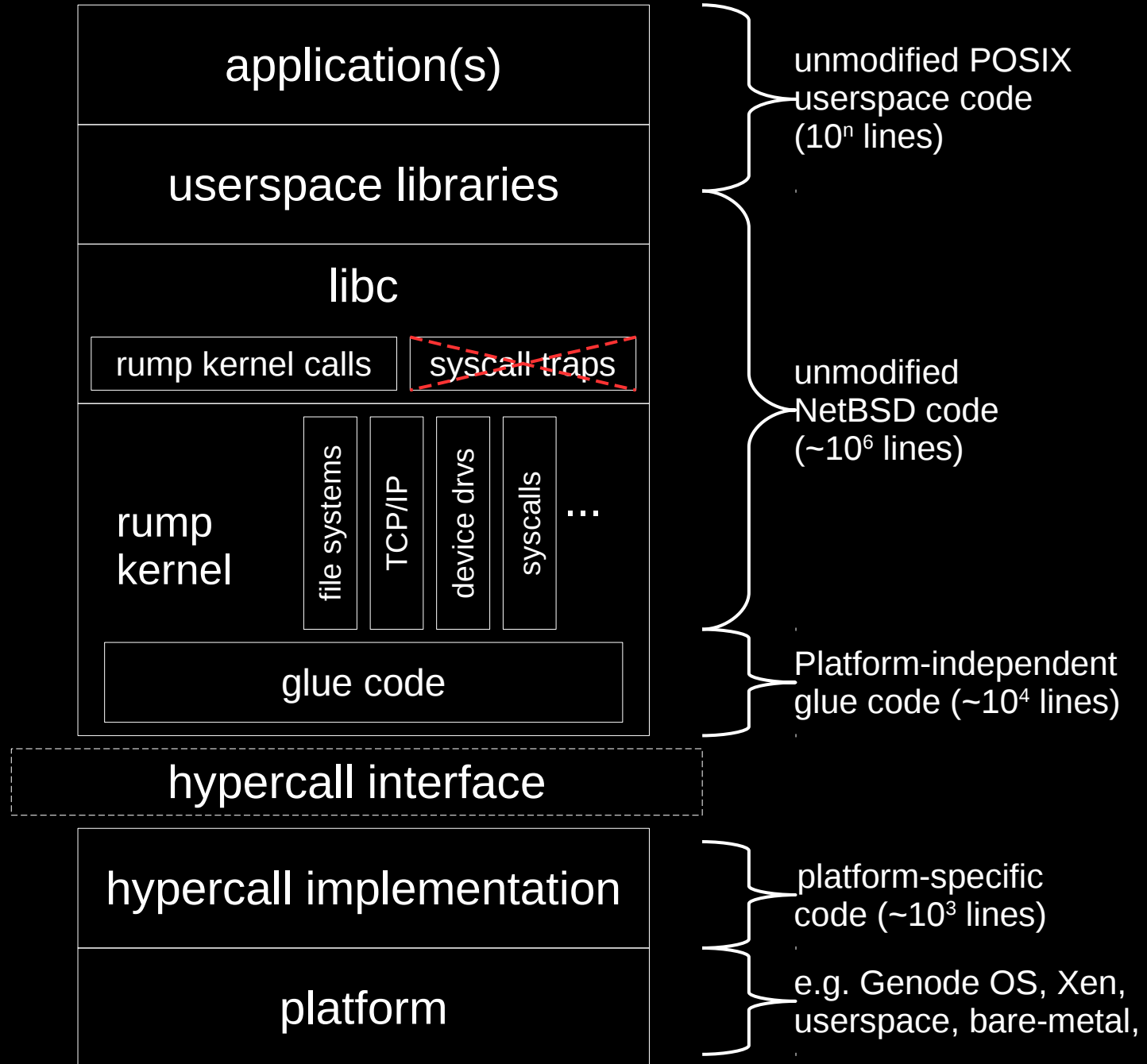
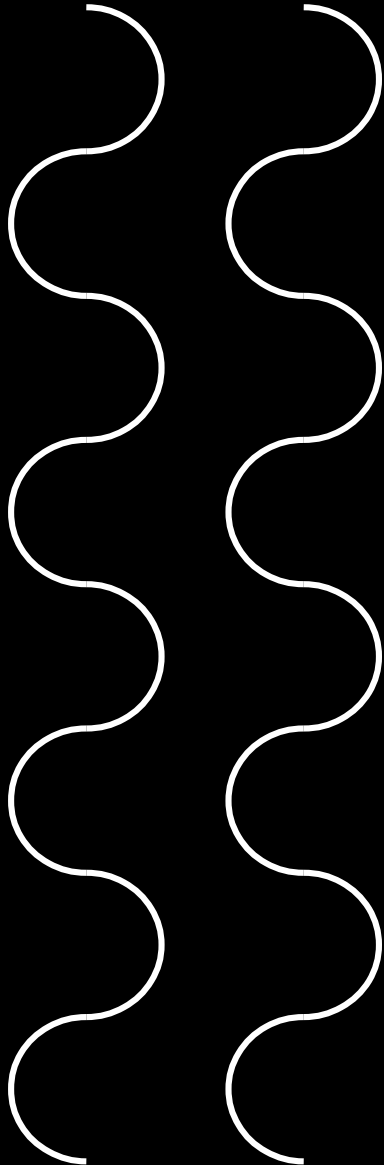
- > runs anywhere
- > integrates into other systems

Wait, that doesn't explain where the
drivers come from

< anykernel (NetBSD)

AN EXAMPLE!

same thread
throughout entire stack

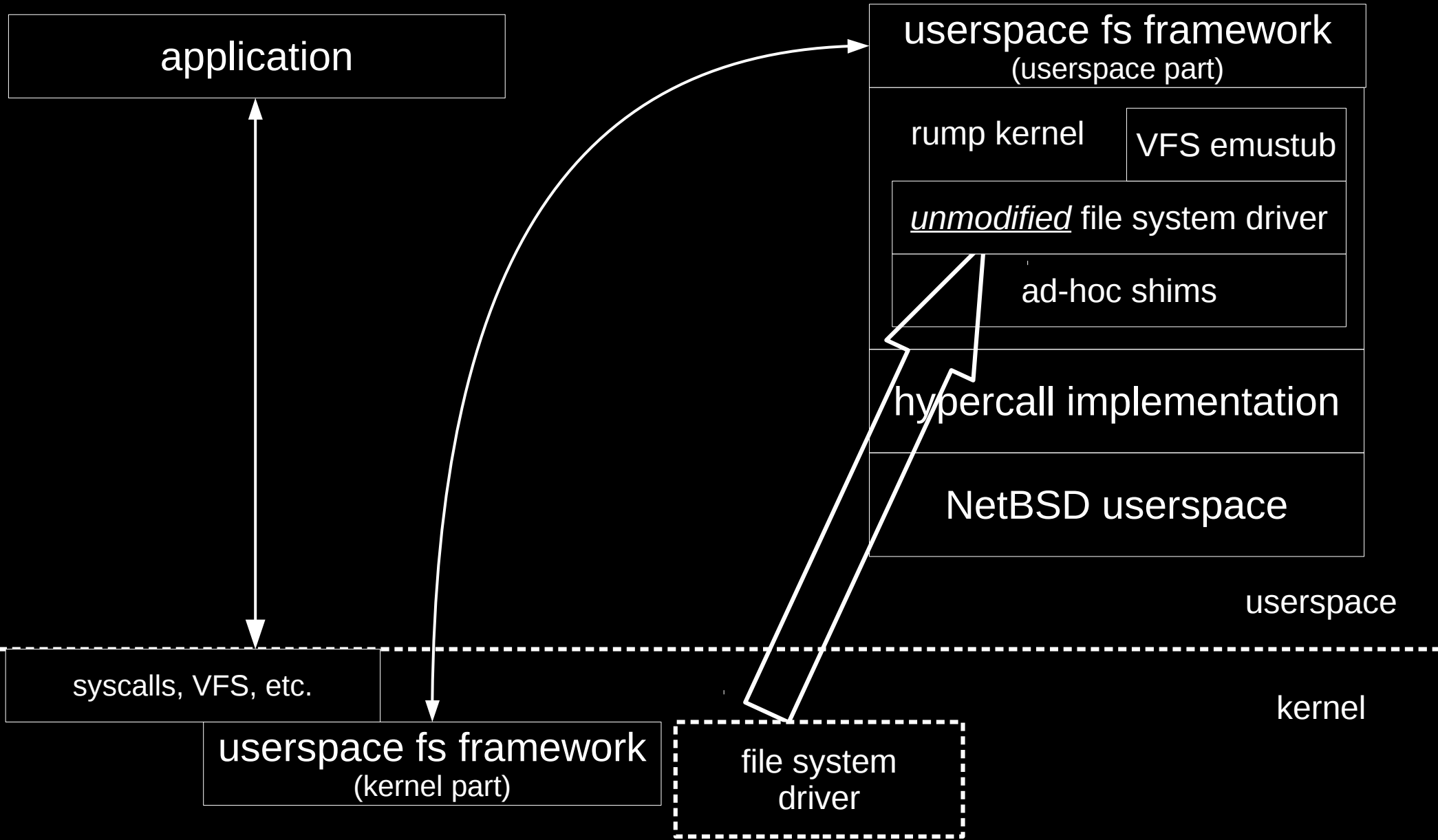


THIRD HALF

(with operating systems, expect the unexpected)

how rump kernels happened

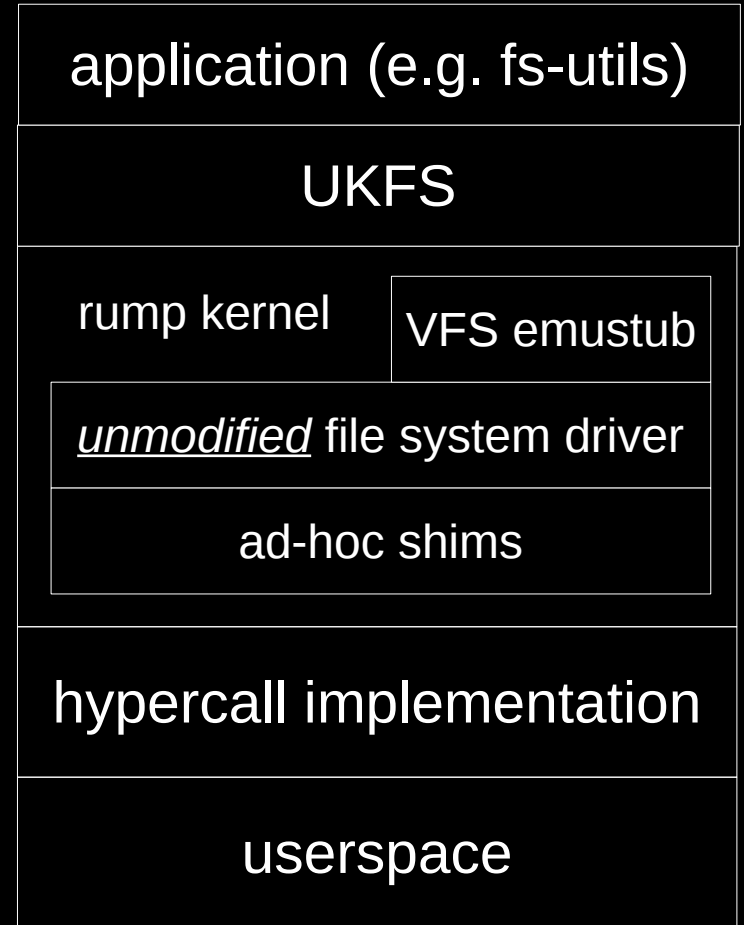
Step 1: RUMP (2007)



Step 2: UKFS (2007)

Q: how hard can implementing
a few syscalls be?

A: very

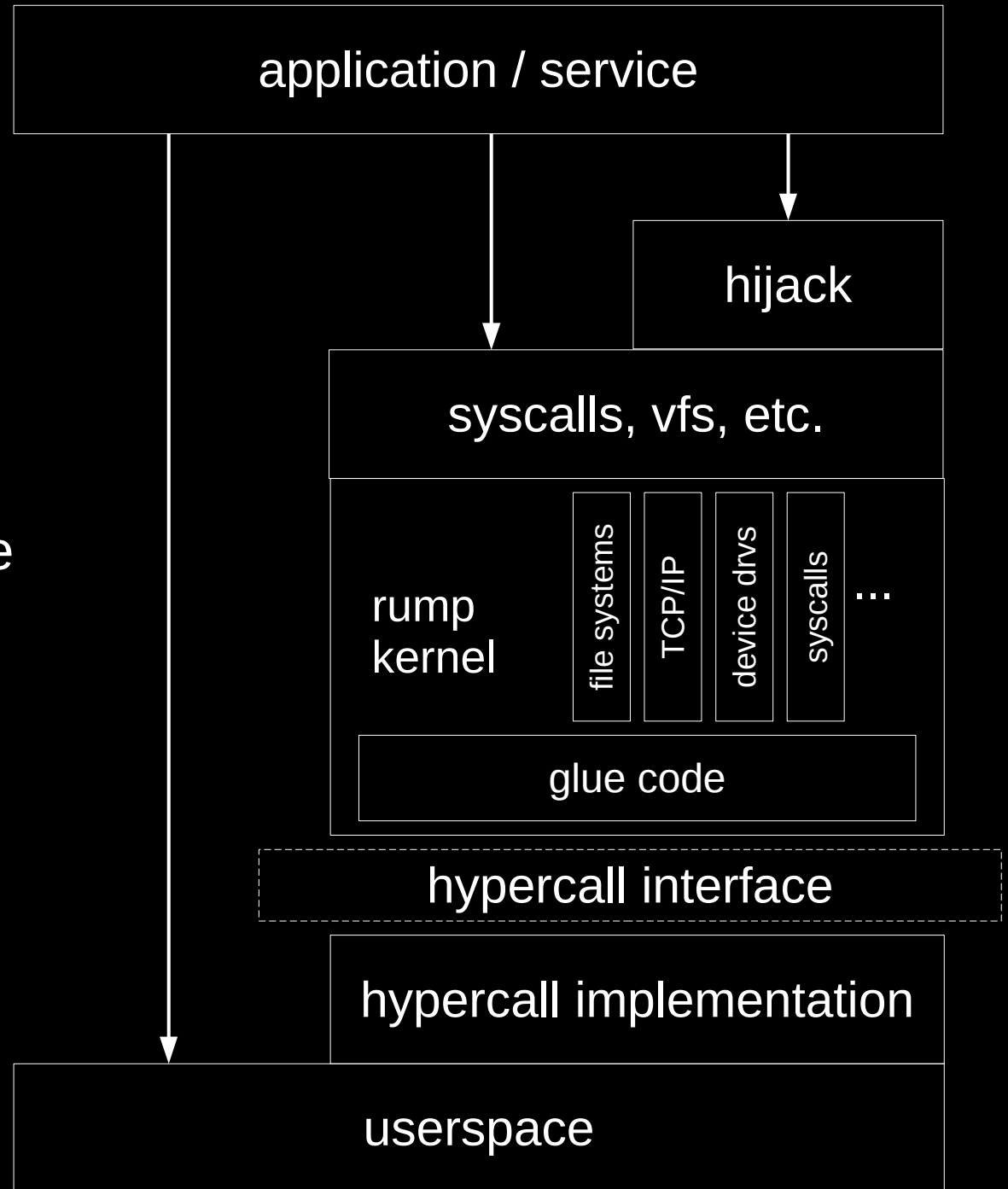


userspace

kernel

Step 3: a lot (2008 - 2011)

- support for all driver subsystems
- isolation from the host
- stable hypercall interface
- anykernel completed
- production quality
- rump kernels used for testing NetBSD
- no libc for rump kernels, applications ran partially on the host



Step 3.5: visions (not an actual step)

ca. turn of the year 2011/2012:

“An anykernel architecture can be seen as a gateway from current all-purpose operating systems to more specialized operating systems running on ASICs. The anykernel enables the device manufacturer to provide a compact hypervisor and select only the critical drivers from the original OS for their purposes. The unique advantage is that drivers which have been used and proven in general purpose systems, e.g. the TCP/IP stack, may be included without modification as standalone drivers in embedded products.”

Step 4: portability to POSIX
2007-2012, 2012-
buildrump.sh (2012-)

4.4STEP: beyond POSIX (201[234])

Type help if you feel like you need some help

Cmd:
Arg1:
Data:

Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2006, 2007, 2008, 2009, 2010, 2011, 2012
The NetBSD Foundation, Inc. All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
The Regents of the University of California.

NetBSD 6.99.14 (RUMP_ROAST) #0: Sat Nov 3 18:28
pooka@T61:/home/pooka/src/nbsd/src/sys/rump
total memory = unlimited (host limit)
timecounter: Timecounters tick every 10.000 msec
cpu0 at thinair0: rump virtual cpu
root file system type: rumpfs
/devffs: hostpath /test.ffs (1408 KB)

Inspector Console Debugger Style Editor Profiler Network

Search scripts (Ctrl+P)

Sources Call Stack

- ffs_mountfs rump.js00.txt:352420
- ffs_mount rump.js00.txt:350069
- VFS_MOUNT rump.js00.txt:213360
- _mount_domount rump.js00.txt:203592
- _do_sys_mount rump.js00.txt:221413
- _sys__mount50 rump.js00.txt:221198
- _sy_call1297 rump.js00.txt:112835
- _rsys_syscall rump.js00.txt:102443
- _rump__sy...pl_mount50 rump.js00.txt:110579
- _domount rump.js00.txt:1117
- _doinit rump.js00.txt:983
- Module.preRun index00.html:24
- run rump.js00.txt:413359
- removeRunDependency rump.js00.txt:936
- dataFile.onload rump.js00.txt:413562

```

352411     var $z=error;
352412     STACKTOP = __stackBase_;
352413     return $z;
352414     default: assert(0, "bad label: " + __label__);
352415 }
352416 }
352417 _ffs_modcmd["X"]=1;
352418
352419 function ffs mountfs($devvp, $mp, $l) {
352420     var __stackBase__ = STACKTOP; STACKTOP += 236; assert(STACKTOP % 4 == 0, "Stack
352421     var __label__;
352422     __label__ = 2;
352423     while(1) switch(__label__) {
352424     case 2:
352425         var $1;
352426         var $2;
352427         var $3;
352428         var $4;
352429         var $mp;
352430         var $bp=__stackBase__;
352431         var $fs;
352432         var $dev=(__stackBase__)+(4);
352433         var $dkw=(__stackBase__)+(12);
352434         var $space;
352435         var $blockloc=(__stackBase__)+(220);
352436         var $fsblockloc=(__stackBase__)+(228);
352437         var $blks;
352438         var $fstype;
352439         var $error;
352440         var $i;
352441         var $bsize;
352442         var $ronly;
352443         var $bset;
352444         var $needswap;
352445         var $lp;
352446         var $cred;
352447         var $bssize;
352448         var $allocsbsize;
352449         var $fsbsize;
352450         $2=$devvp;
352451         $3=$mp;
352452         $4=$l;
352453         $bset=0;
352454         $needswap=0;
352455         $bssize=8192;
352456         var $5=$2;
352457         var $6=((($5+152)|0);
352458         var $7=$6;
352459         var $8=HEAP32[(((($7)>>2)];
352460         var $9=((($8+12)|0);
352461         var $st$36$0=((($9)|0);
352462         var $10$0=HEAP32[(((($st$36$0)>>2)];
352463

```

```
[ 57.775993] initiating rump kernel bootstrap
[ 57.776462] Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 200
4, 2005,
[ 57.776494]     2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013
[ 57.776525]     The NetBSD Foundation, Inc. All rights reserved.
[ 57.776554] Copyright (c) 1982, 1986, 1989, 1991, 1993
[ 57.776583]     The Regents of the University of California. All rights rese
rved.
```

```
[ 57.776635] NetBSD 6.99.17 (RUMP-ROAST) #0: Mon Apr 22 22:28:13 UTC 2013
[ 57.776661] pooka@voidus:/home/pooka/buildrump.sh/obj/lib/librump
[ 57.776689] total memory = unlimited (host limit)
[ 57.776923] timecounter: Timecounters tick every 10.000 msec
[ 57.776960] timecounter: Timecounter "rumpclk" frequency 100 Hz quality 0
[ 57.784151] cpu0 at thinair0: rump virtual cpu
[ 57.784321] cpu1 at thinair0: rump virtual cpu
[ 57.784621] cpu2 at thinair0: rump virtual cpu
[ 57.784996] cpu3 at thinair0: rump virtual cpu
[ 57.923691] rump kernel bootstrap complete, scheduling demo
[ 57.925928] IPv6: ADDRCONF(NETDEV_CHANGE): tun0: link becomes ready
[ 57.925997] br0: port 2(tun0) entered forwarding state
[ 57.926090] br0: port 2(tun0) entered forwarding state
[ 58.939953] dhcp: virt0: adding IP address 10.0.2.17/24
[ 58.939966] dhcp: virt0: adding route to 10.0.2.0/24
[ 58.939978] dhcp: virt0: adding default route via 10.0.2.2
[ 58.939989] lease time: 86400 seconds
[ 59.133318] wrote http request, rv 16
[ 59.311235] read 3072 bytes
[ 59.311235] that was an educational experience. we learned:
[ 59.311235] >
```

```
Questions about VGER's services: &lt;mailto:postmaster@vger.kernel.org&gt;<br>
Postmaster contact addresses as an <A HREF="real-vger-postmasters.gif">image</
a> - do prefer above given one<br>
FAQ answers: <A HREF="http://www.tux.org/lkml/">http://www.tux.org/lkml/</A><
br>
<!--
<A HREF="mailto:honey+1366672895@vger.kernel.org">mail to Honey</A>
mailto: hunaja+1366672895@vger.kernel.org
-->
Bo and others want <A HREF="bo.html">email</A> to teach filters about spam.
</P>
</BODY>
</HTML>
# _
```

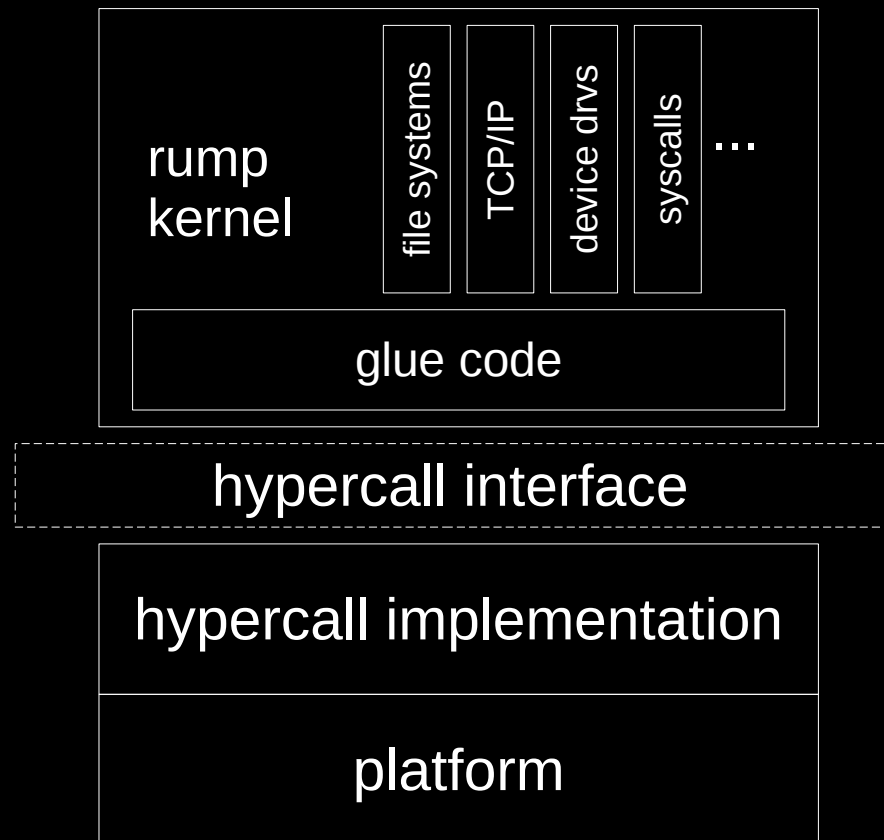


```
pci0 dev 30 function 0 not configured
vendor 0x8086 product 0x2811 (ISA bridge, revision 0x03) at pci0 dev 31 function
0 not configured
vendor 0x8086 product 0x2850 (IDE mass storage, interface 0x8a, revision 0x03) a
pci0 dev 31 function 1 not configured
vendor 0x8086 product 0x2829 (SATA mass storage, AHCI 1.0, revision 0x03) at pci
0 dev 31 function 2 not configured
vendor 0x8086 product 0x283e (SMBus serial bus, revision 0x03) at pci0 dev 31 fu
nction 3 not configured
dhcp: wm0: adding IP address 192.168.2.111/24
dhcp: wm0: adding route to 192.168.2.0/24
dhcp: wm0: adding default route via 192.168.2.1
lease time: infinite
got response:
HTTP/1.1 200 OK
Date: Wed, 13 Aug 2014 17:46:52 GMT
Server: Apache/2.4.10 (Unix)
Last-Modified: Sun, 20 Jul 2014 11:30:00 GMT
ETag: "4d99-4fe9e4f9f1c46"
Accept-Ranges: bytes
Content-Length: 19865
Connection: close
Content-Type: text/html; charset=ISO-8859-1
[omitting rest ...]
```

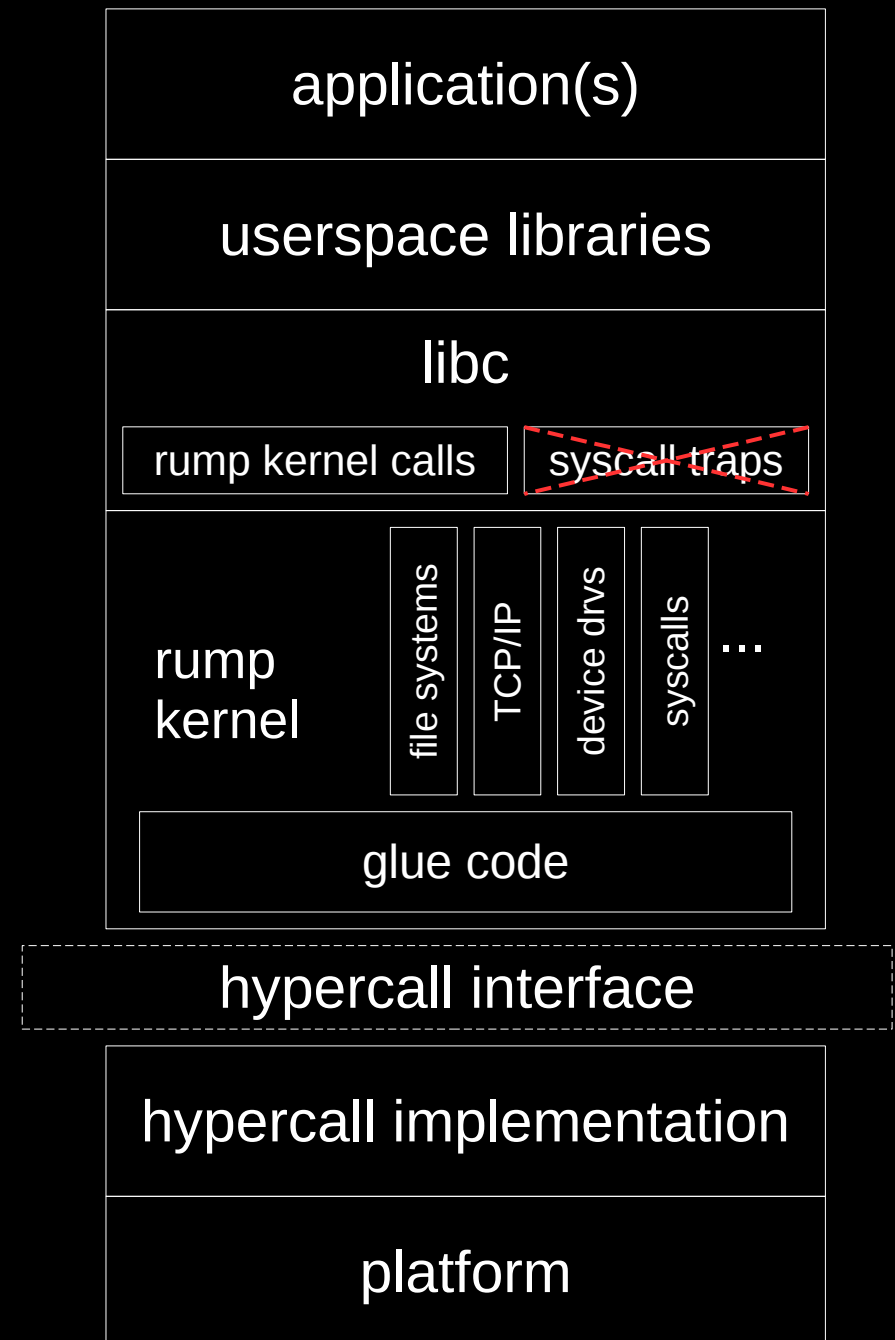
```
7 not configured
vendor 0x8086 product 0x8c26 (USB serial bus, EHCI, revision 0x04) at pci0
9 function 0 not configured
vendor 0x8086 product 0x8c4f (ISA bridge, revision 0x04) at pci0 dev 31 func
0 not configured
vendor 0x8086 product 0x8c03 (SATA mass storage, AHCI 1.0, revision 0x04) at
0 dev 31 function 2 not configured
vendor 0x8086 product 0x8c22 (SMBus serial bus, revision 0x04) at pci0 dev 3
nction 3 not configured
dhcp: wm0: adding IP address 192.168.2.101/24
dhcp: wm0: adding route to 192.168.2.0/24
dhcp: wm0: adding default route via 192.168.2.1
lease time: infinite
got response:
HTTP/1.1 200 OK
Date: Wed, 13 Aug 2014 17:47:33 GMT
Server: Apache/2.4.10 (Unix)
Last-Modified: Sun, 20 Jul 2014 11:30:00 GMT
ETag: "4d99-4fe9e4f9f1c46"
Accept-Ranges: bytes
Content-Length: 19865
Connection: close
Content-Type: text/html; charset=ISO-8859-1
[omitting rest ...]
```

FUJITSU

Step 5.1: rump run (2013, 2014)



Step 5.2: rump run (2013, 2014)



FINAL HALF

conclusions & other tidbits

All le gory technical details:

<http://book.rumpkernel.org/>

2nd edition is work in progress

Will be available as free pdf, hopefully printed too

Community

- <http://rumpkernel.org/>
- <http://repo.rumpkernel.org/>
 - BSD-licensed source code
- <http://wiki.rumpkernel.org/>
- rumpkernel-users@lists.sourceforge.net
- #rumpkernel on irc.freenode.net
- @rumpkernel

The actual conclusions

*You can make an omelette without
breaking the kitchen!*