lowRISC Summer of Code

TCP/IP Offload to Minion Cores using Rump Kernels

Sebastian Wicki

Mentored by: Justin Cormack, Antti Kantee
Organizer: Alex Bradbury, lowRISC

We want an open computing eco-system:

comprehensible, flexible and **reusable**

Operating System

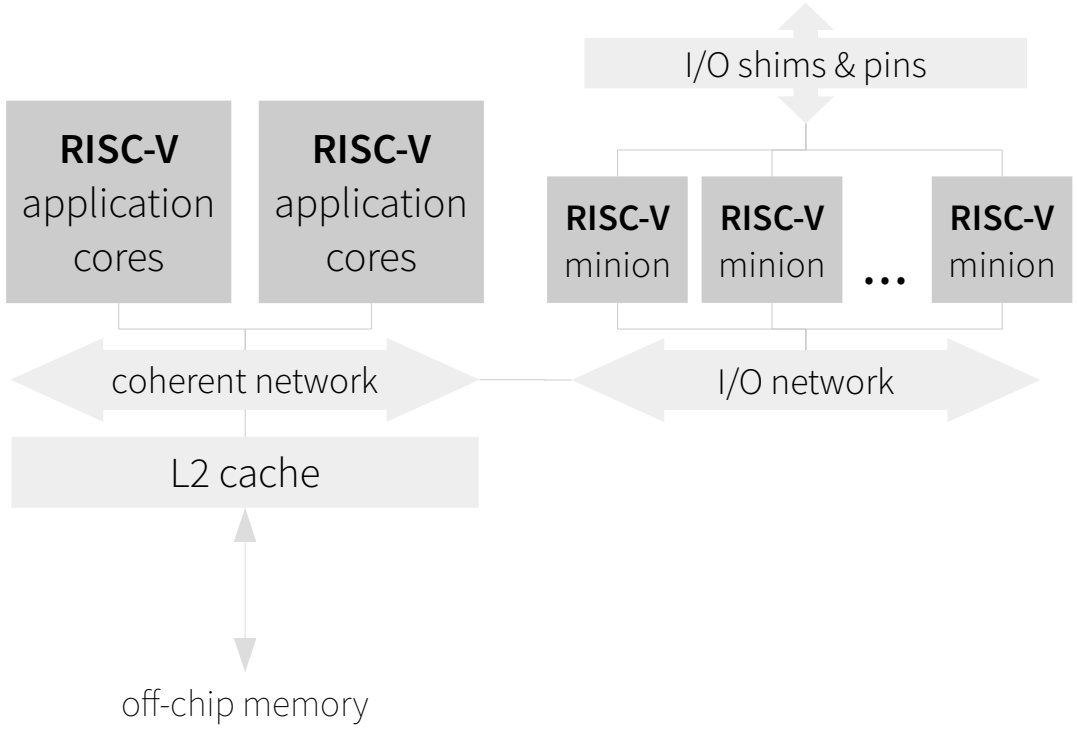
Processor

Peripherals

hardware platform

lowRISC minion cores

- small, dedicated RISC-V cores
- specialized for I/O processing
- protocols in software
  - e.g. SPI, I2C, SDIO



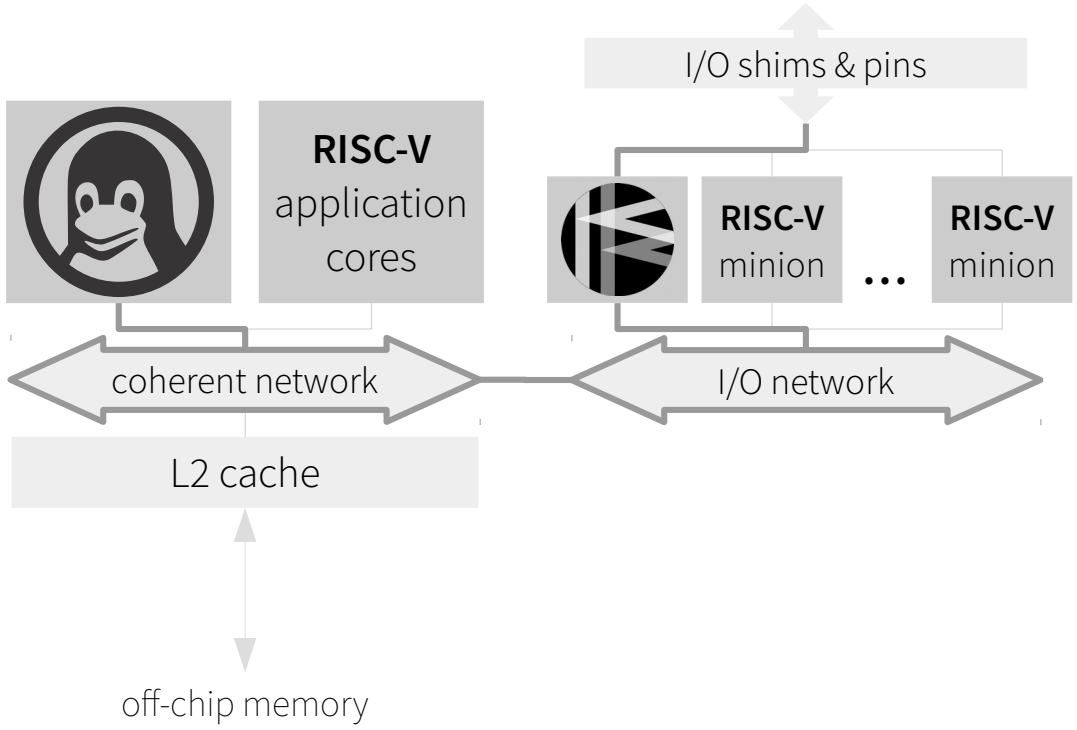
software platform

rump kernels

- driver container
  - file systems, network, audio
- run anywhere
  - userspace, hypervisors, bare-metal
  - integrate into own system
- based on NetBSD

project proposal

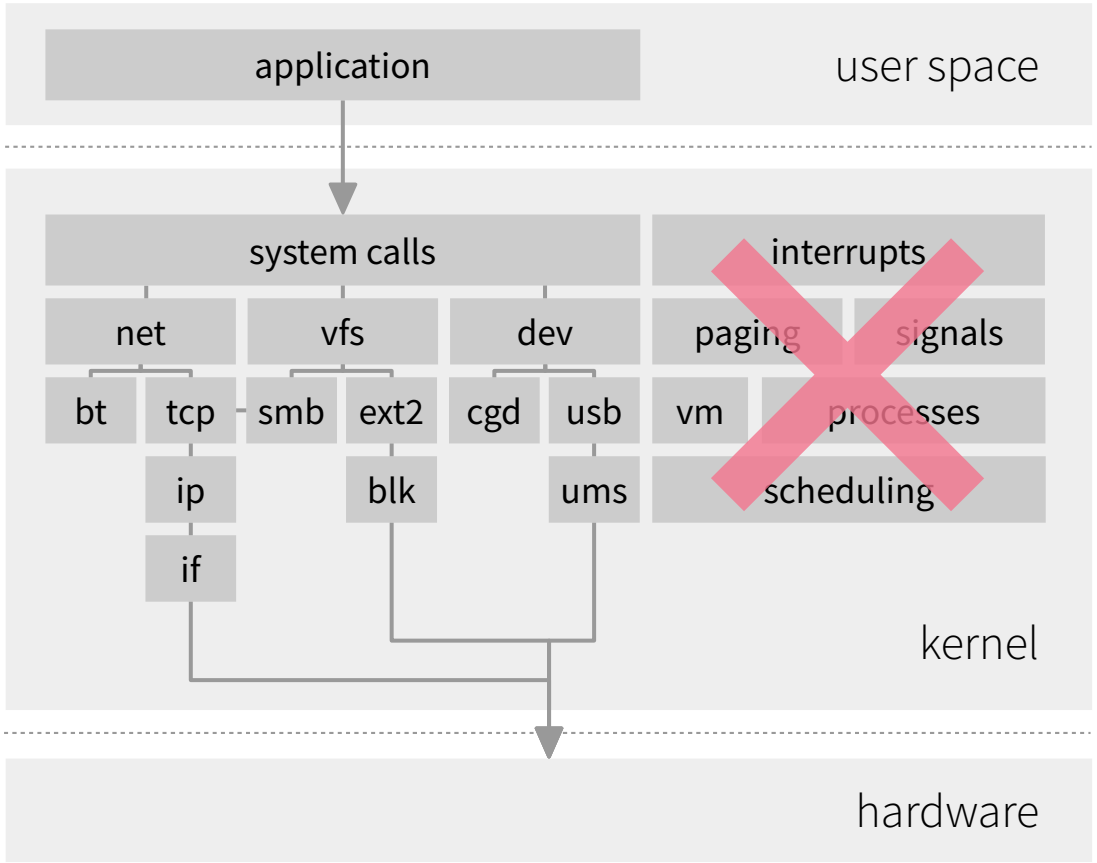
- rump kernels on minion cores
- process I/O using rump drivers
  - e.g. TCP/IP
- traditional OS on application cores



rump kernels

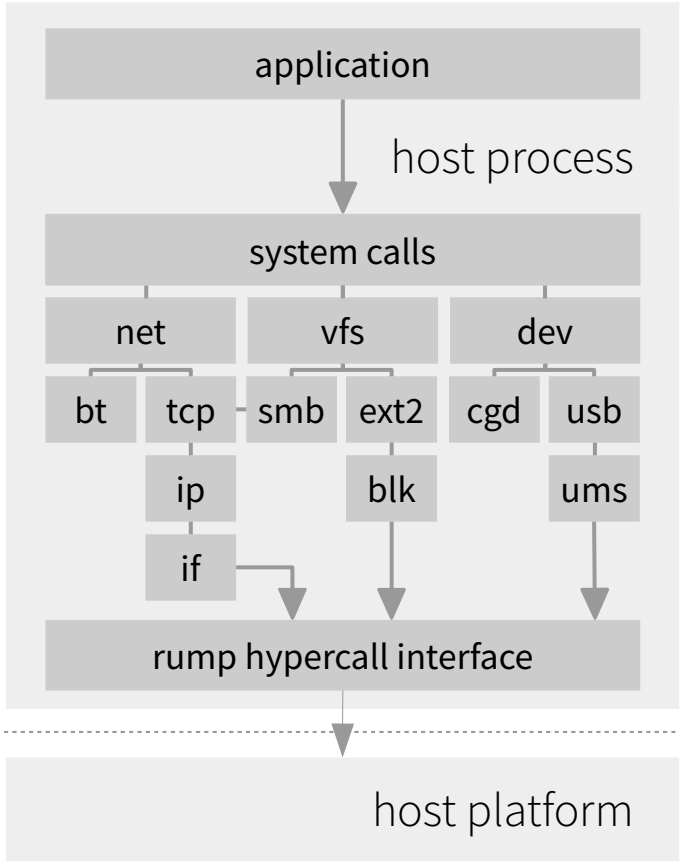
rump kernels: not an operating system

- no processes or threading
- no virtual memory management
- no privilege levels
- no interrupt handling

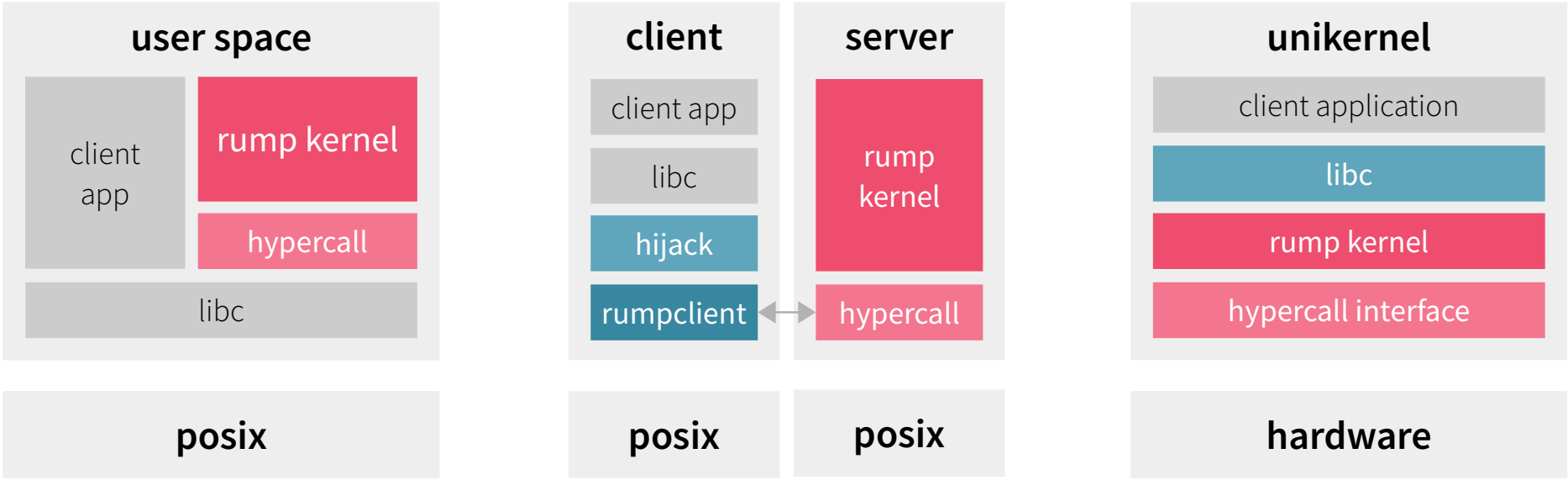


rump kernels: not an operating system

- hypercall interface
  - thread scheduling
  - memory allocation
  - console output
  - I/O hypercalls



rump kernels: run anywhere

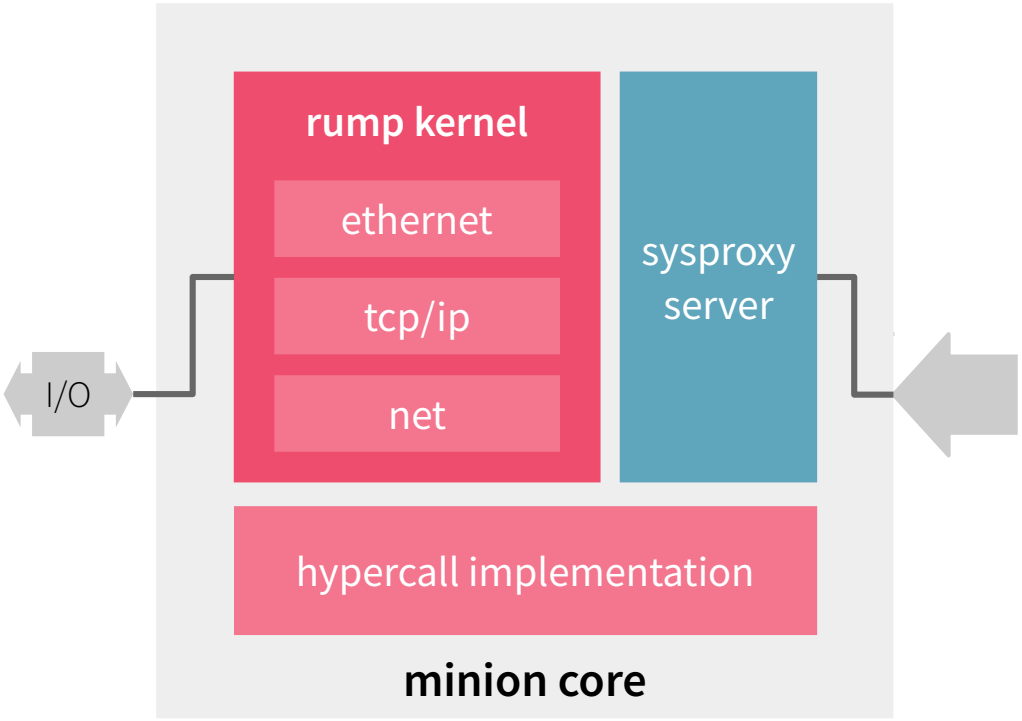


- client code is aware of rump kernel

- unmodified client code
- intercept libc calls
- forward over network

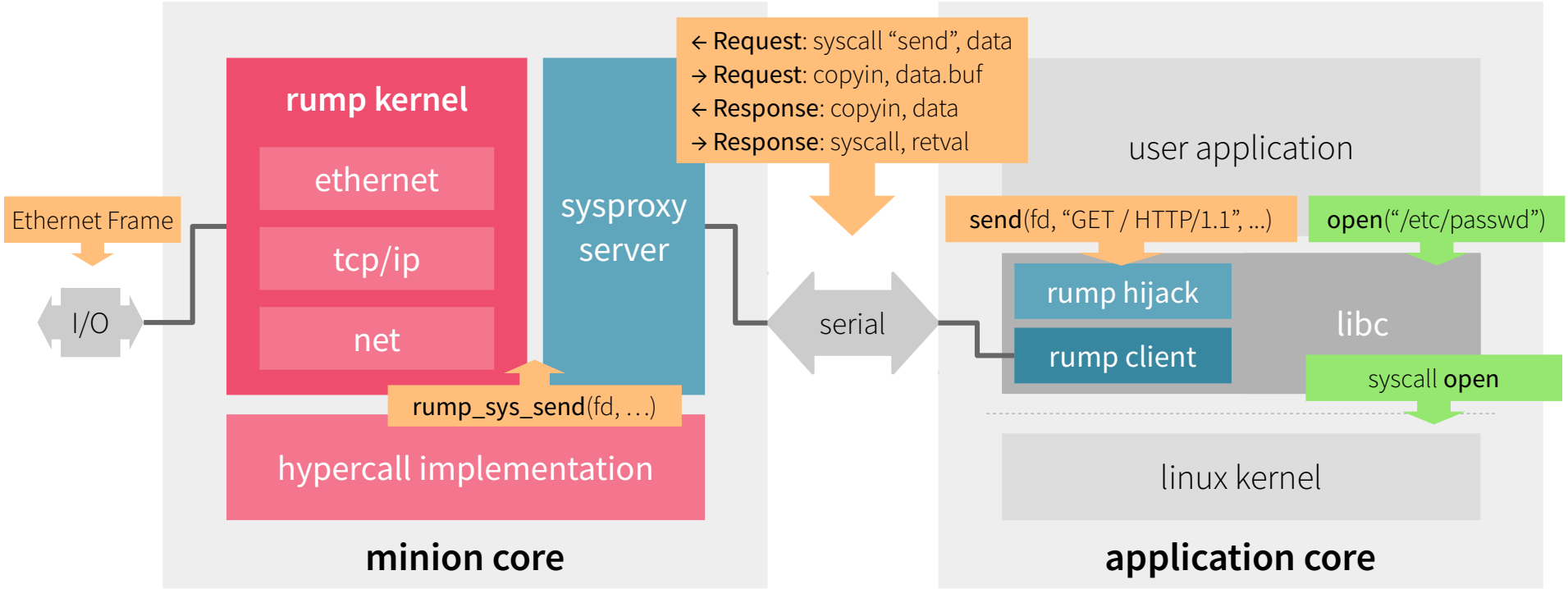
- unmodified client code
- rump kernel aware libc

rump kernels on minion cores



- hypercalls for bare-metal RISC-V
  - based on the rumprun unikernel
- drivers running on rump
  - custom ethernet driver
  - TCP/IP stack from NetBSD
- syscall proxy server
  - rewritten to run on bare-metal
  - uses serial line instead of network

rump kernels on minion cores



Limitations

- currently only runs in Spike
  - wrote virtual network card
  - simulates minion core
  - host offloads into Spike
- system requirements
- everything over serial protocol
  - use DMA engine for `copy{in,out}`
- code needs some cleanup

Conclusions

- rump kernels are flexible
  - not limited to TCP/IP offload
  - no need to use syscall proxy
  - run your own apps on minion cores
- code is reusable

http://rumpkernel.org

@rumpkernel

#rumpkernel irc.freenode.net

Contact: @gandro23