Rptun framewrok & services between multi-CPUs (IPC)

Guiding Li

China, Xiaomi Inc, VELA

THE APACHE SOFTWARE FOUNDATION

Introduction

- Guiding Li
- From Xiaomi inc, China, Beijing
- Focus on Nuttx kernel : IPC, scheduler, SMP, arm, MM ...
- Working on VELA (Nuttx kernel)

Xiaomi Vela

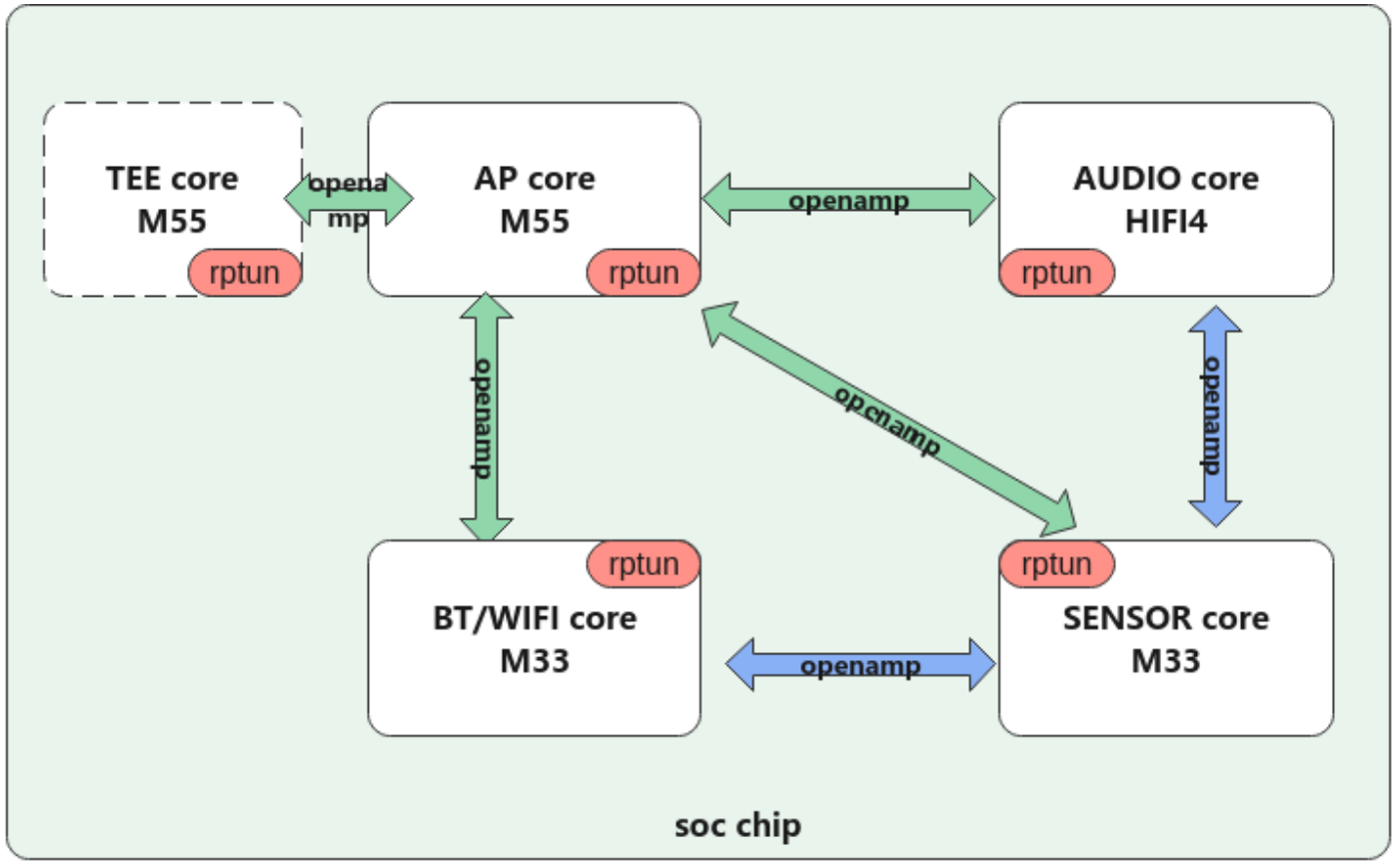
What is rptun framework

Introduction:

- Rptun means remote proc tunnel
- A tunnel for Multi-cpus
- Based on openamp

Design purpose:

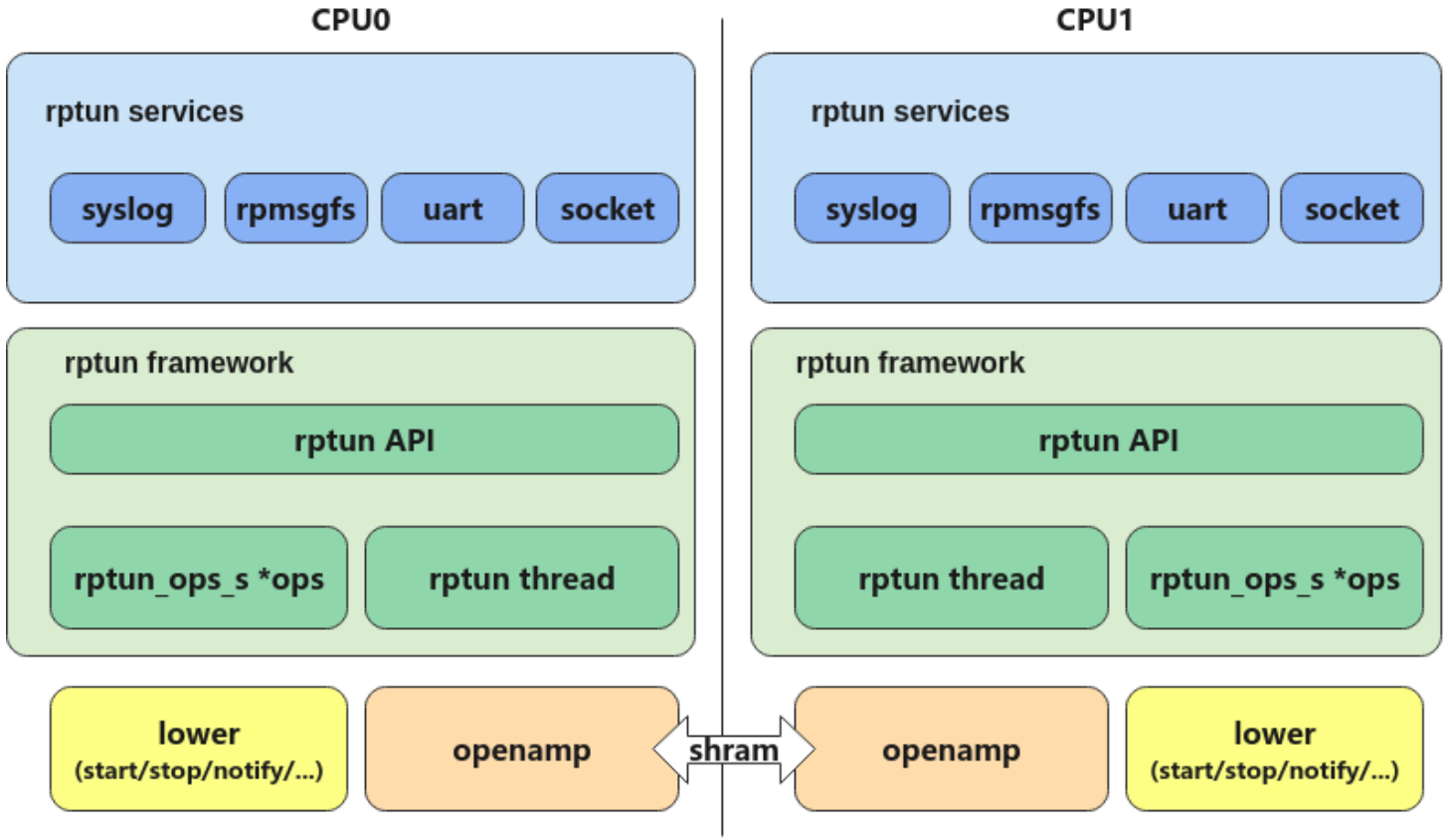
- A better way for resource shareable
- A standard communication method for multi-cpus, (connected with Linux)
- A efficient transmission method
- Easy to use



What does rptun framework do

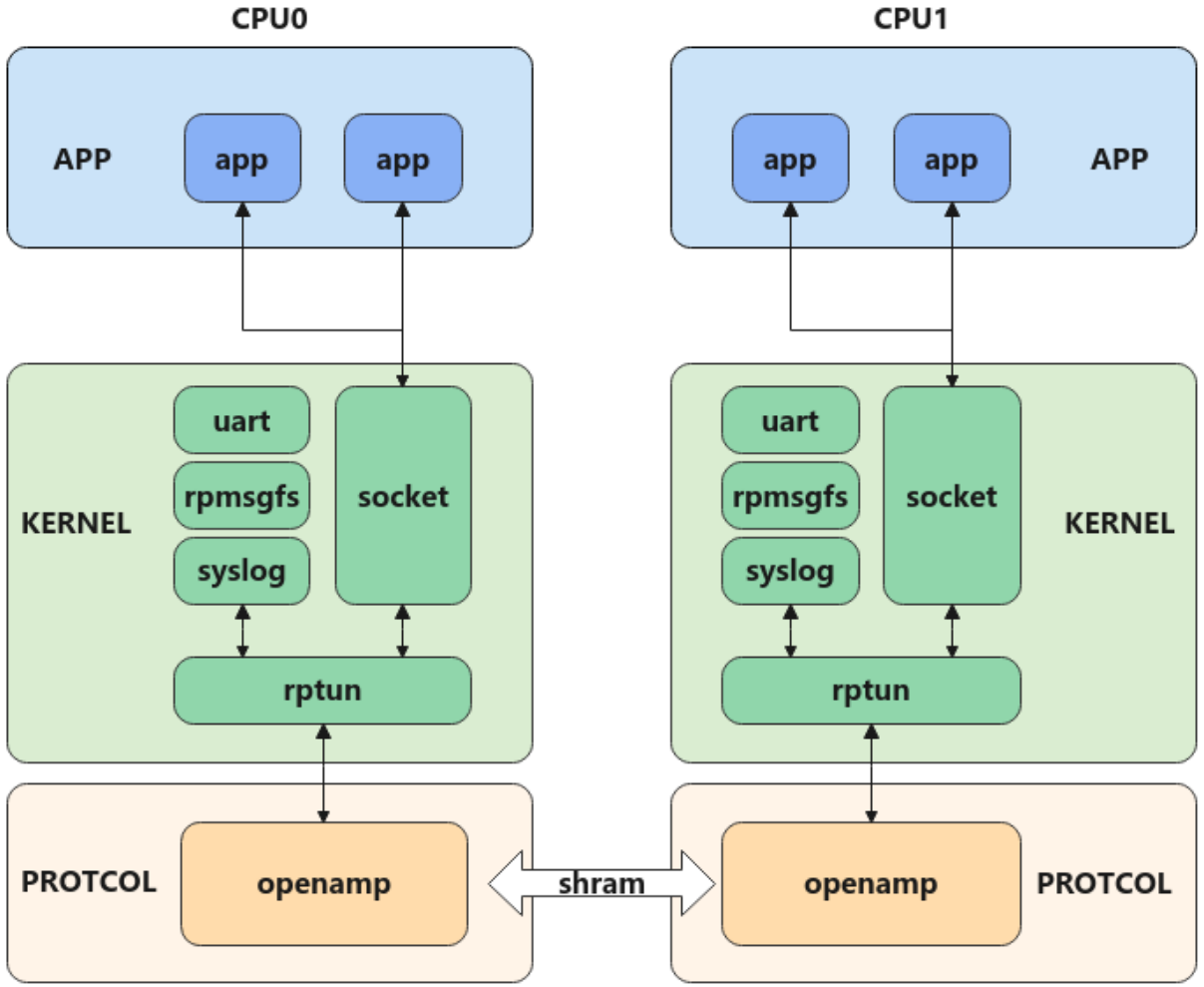
Features:

- **Provide common lower API for drivers**
- **Provide common API for rptun services**
- **Init & setup openamp**
- **Handle remote cpu incoming msg**
- **Handle ns_bind special msg missing**



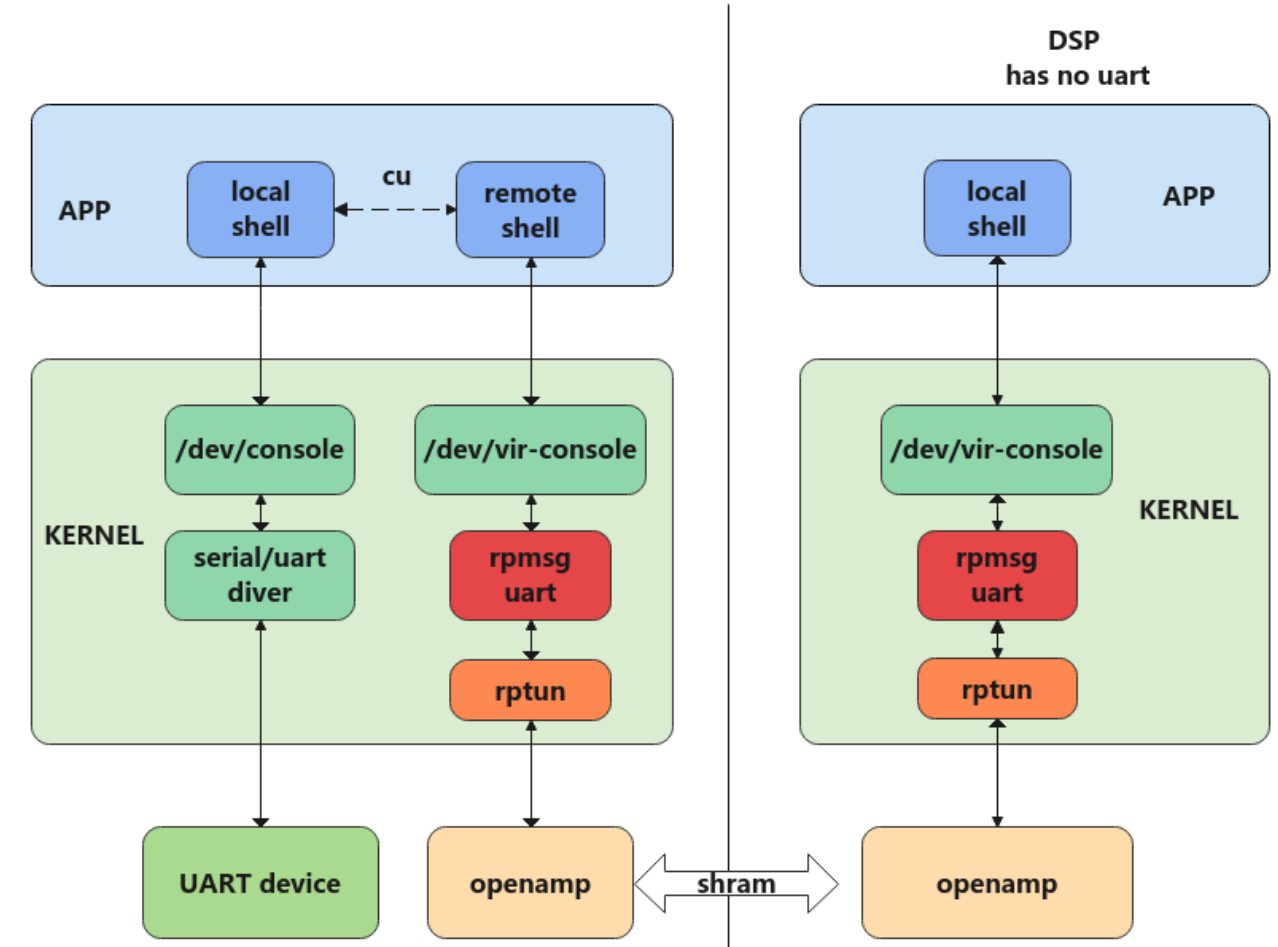
Rptun framewrok & services

- Rptun services base on rptun framework
- BOTH on kernel space & user space



Rptun services - rpmsg-UART

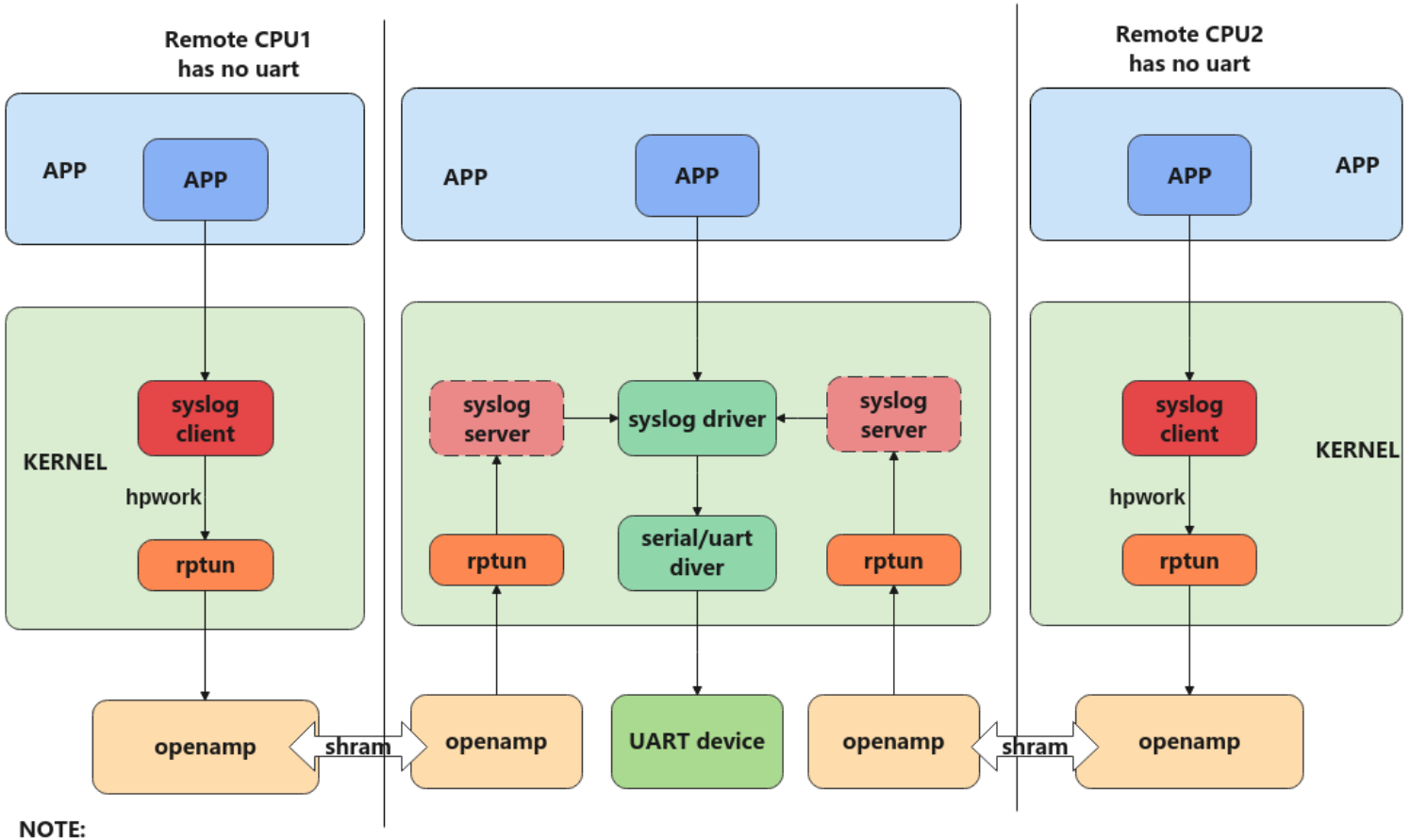
- Design for access remote shell
- A telnet/ssh liked protocol
- Send cmd to remote, receive remote returns
- Use system cmd ‘cu’ as remote terminal



NOTE:
1. With uart rpmsg you can easily execute the remote shell, like your local shell.
2. This is likely SSH, but its for remote cpu

Rptun services - rpmsg-syslog

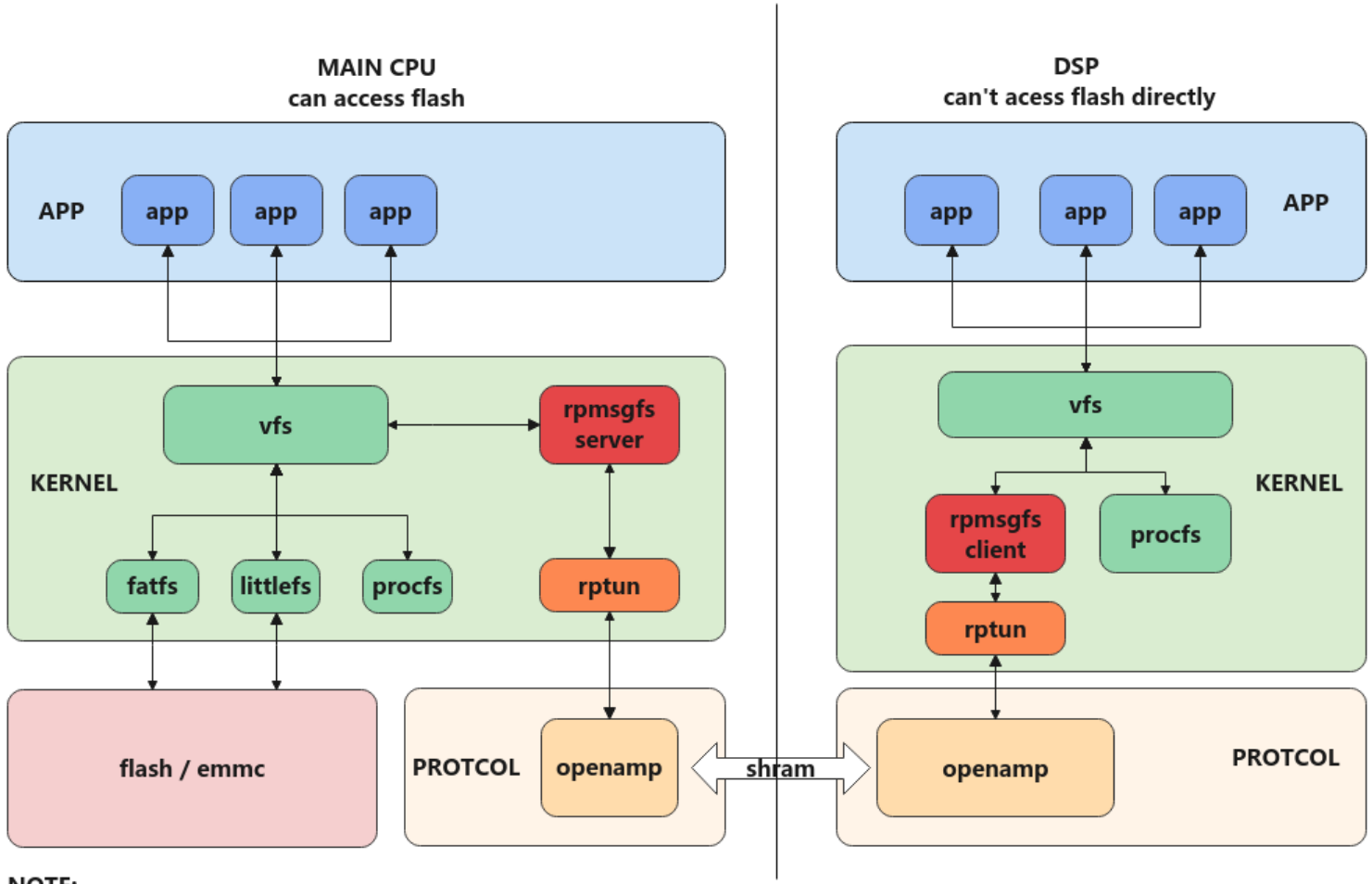
- Design for access remote syslog
- Client syslog buffer buffered logs
- Watermark trigger or \n trigger hpwork for transfering log
- Syslog align with \n in server side In case of cross with other CPU



NOTE:
1. With syslog rpmsg you can dirctly call syslog from remote CPU
2. The crash log from remote CPU can buffer in client syslog rpmsg buffer, can continue logout when system reboot.

Rptun services - rpmsg-fs

- Design for access remote filesystem
- A NFS/SMB liked protocol
- Good experience for DSP
- Mount cmd support
- Folders operation support
- High speed up to 10MB/s



NOTE:
1. With rpmsgfs you can easily access all the remote filesystem, like your own filesystem.
2. Not only file operation, but also folder operation

Rptun services - rpmsg-fs

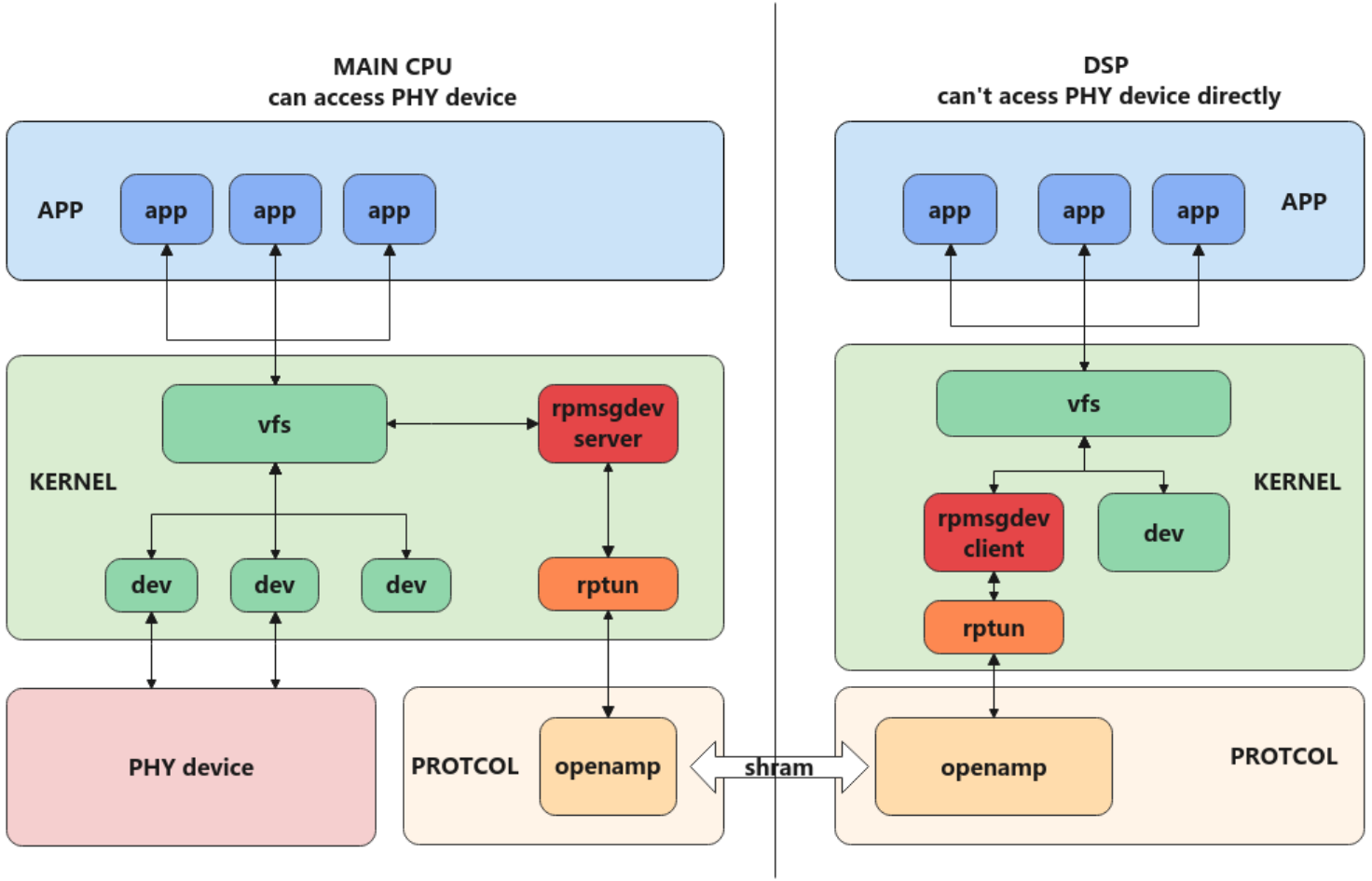
- Data from L61, Xiaomi watch S1 pro
- Clock ap 200M
- Clock cp 100M
- Upper data from tmpfs on ap
- Lower data from rpmsgfs on cp

tmpfs (ap)	Read(KB/s)	Write(KB/s)
dd if=/dev/zero of=/tmp/test1 bs=512 count=1024	74898	58254
dd if=/dev/zero of=/tmp/test2 bs=1024 count=1024	95325	69905
dd if=/dev/zero of=/tmp/test3 bs=2048 count=1024	110376	87381
dd if=/dev/zero of=/tmp/test4 bs=4096 count=1024	116508	97541
dd if=/dev/zero of=/tmp/test5 bs=8192 count=1024	113359	98689

rpmsgfs (cp -> ap)	Read(KB/s)	Write(KB/s)
dd if=/dev/zero of=/tmp/test12 bs=512 count=1024	2803	2008
dd if=/dev/zero of=/tmp/test22 bs=1024 count=1024	4211	3256
dd if=/dev/zero of=/tmp/test32 bs=2048 count=1024	6026	6078
dd if=/dev/zero of=/tmp/test42 bs=4096 count=1024	9915	10407
dd if=/dev/zero of=/tmp/test52 bs=8192 count=1024	15060	14074

Rptun services - rpmsg-dev

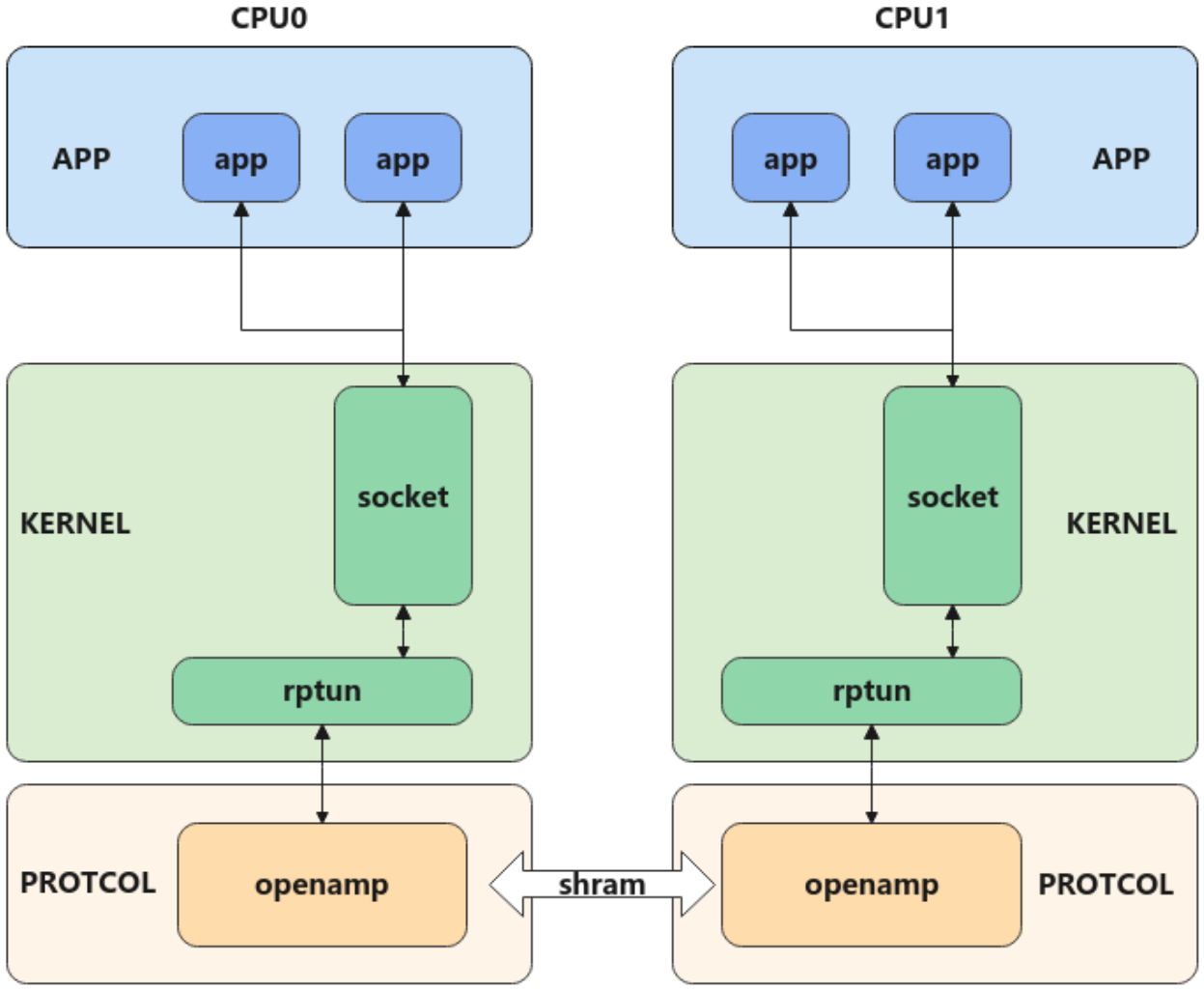
- Design for access remote device
- Foucus on single device file
- Poll operation support
- Planing/doing:
  rpmsg-mtd-dev
  rpmsg-block-dev



NOTE:
1. With rpmsgdev you can easily access all the remote device, like your own device.
2. Poll operation support

Rptun services - rpmsg-socket

- Design for IPC usage from APP layer
- A local socket liked protocol
- Full socket API supported
- Easy for APP to use
  (socket bind listen connect accept)
- BOTH support stream & dgram mode
- BOTH support block & unblock mode
- Efficient flow control
- High speed 10MB/s



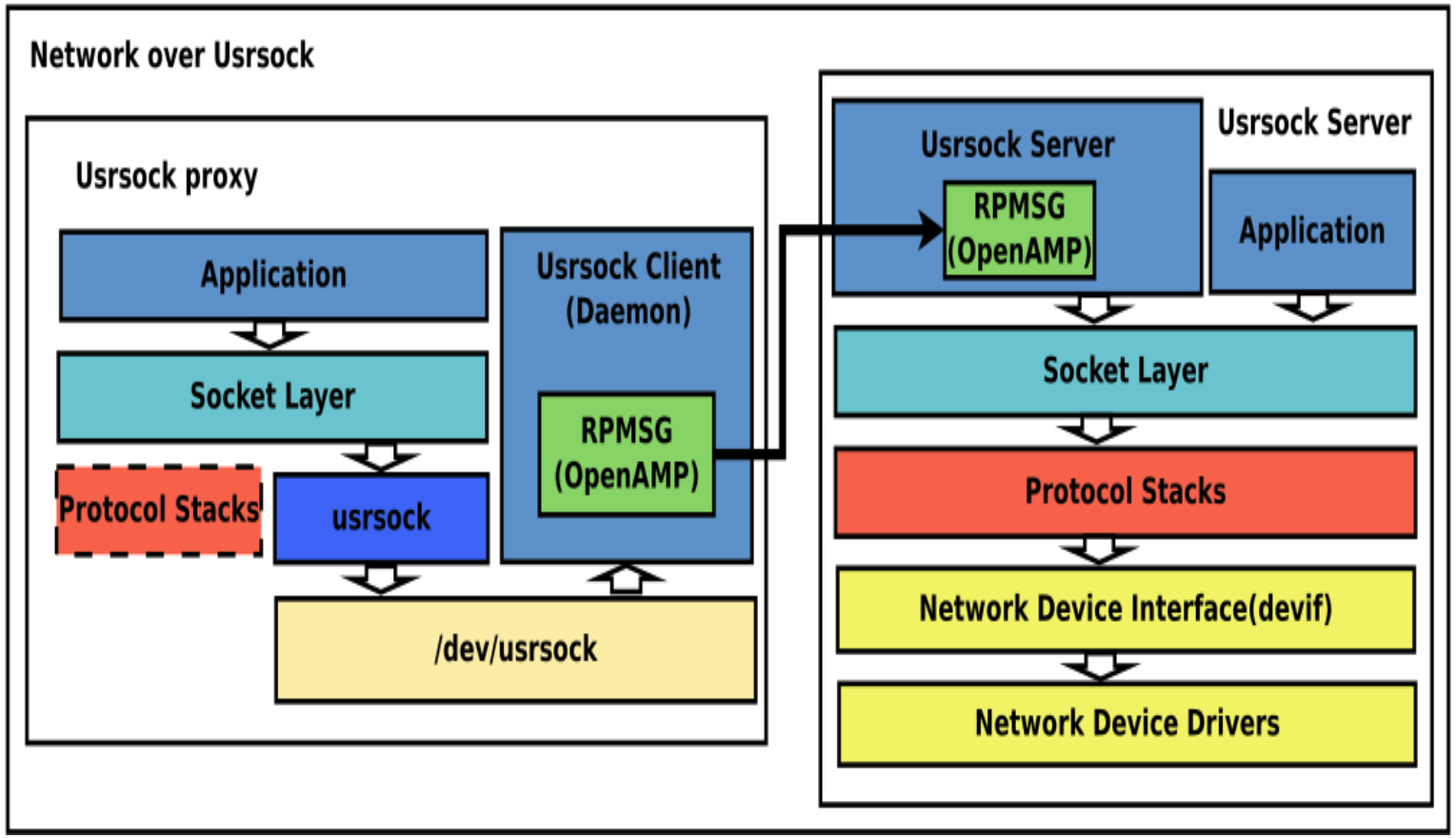
Rptun services - rpmsg-socket

- Data from L61, Xiaomi watch S1 pro
- Clock ap & audio 200M
- Clock cp & sensor 100M
- Measured by iperf2

sock rpmsg		
server	client	speed (Mb/s)
ap	cp	75.5
cp	ap	67.9
ap	audio	83.6
audio	ap	82.6
ap	sensor	69.4
sensor	ap	19.3
ap	tee	9.26
tee	ap	8.98
cp	audio	70.3
audio	cp	63.8
cp	sensor	11.8
sensor	cp	17.6

Rptun services - rpmsg-usrsock

- Design for access remote network
- Full feature supported
- APP unknowless
- Speed:
  Server with STACK : 14Mb/s
  Client with usrsock : 11Mb/s
- Speed 20% off,
  we are working on it.



Rptun services - others

- rpmsg gpio --- access remote cpu GPIO
- rpmsg rtc --- RTC sync for all cpus
- rpmsg regulator --- access remote cpu regulator
- rpmsg uorb --- a mulit-core message publish & subscribe method
- rpmsg clock --- distributed clock tree support

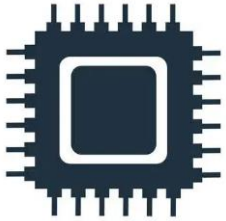
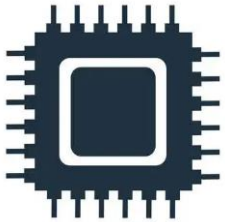
....

Rptun APPs

- rexec --- can execute remote shell cmd
- kvdb --- distributed key value database
- rptun ping --- A tool for measuring IPC latency

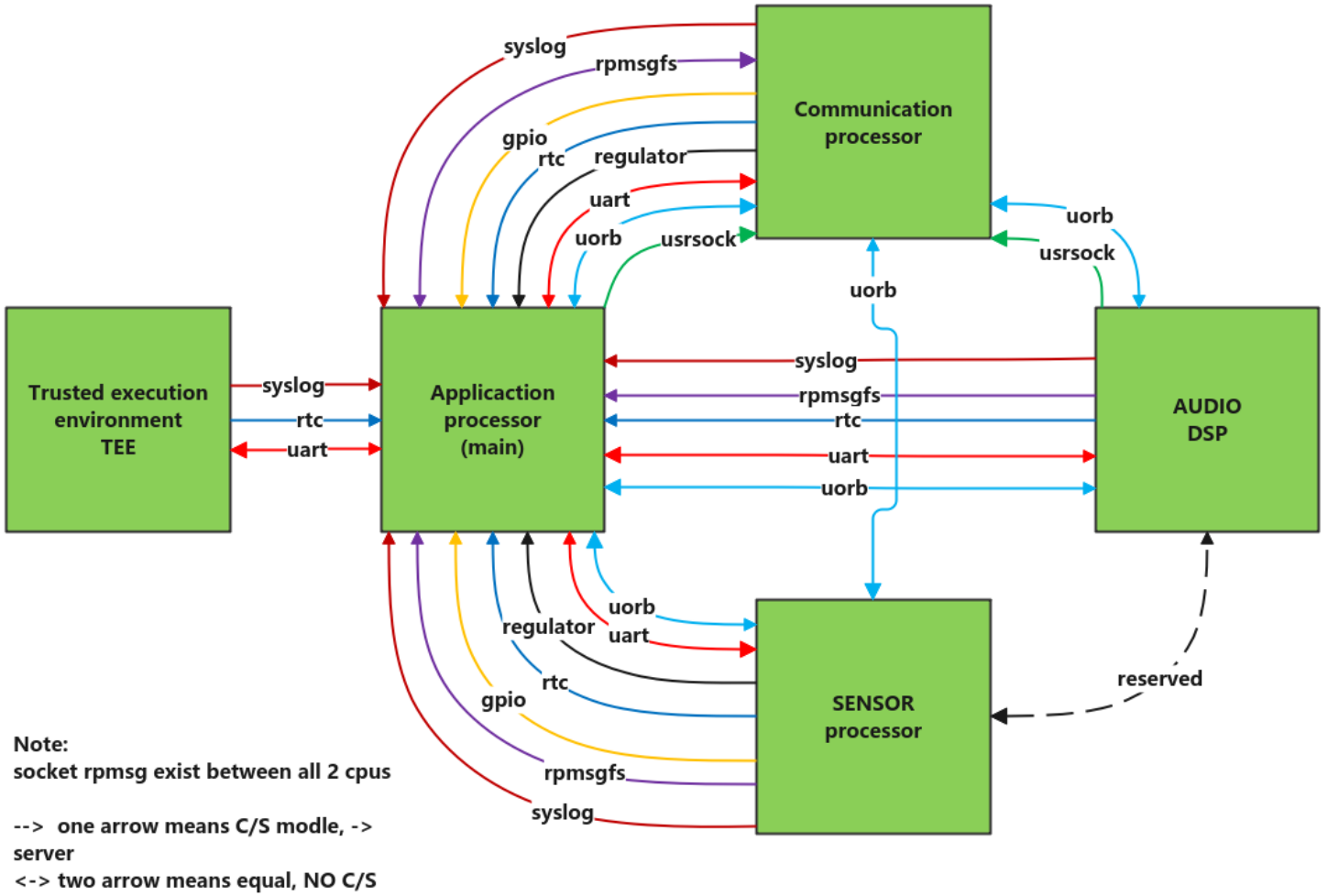
....

Rptun services on real device

2018	2019	2021	2021	2022
Xiaomi Surge S1	Unicorn1	Unicorn2	L05C	L61
				
Xiaomi 5C 2 cores IPC Linux DSP	NBIOT SOC 3 cores IPC AP CP SP	BT headset SOC 2 cores IPC AP DSP	Sound box 2 cores IPC AP DSP	Watch S1 Pro 5 cores IPC AP CP DSP SENSOR TEE

Rptun services on real device

- Rptun services on Xiaomi VELA L61 (Xiaomi watch S1 pro, 2022.8.11)
- 5 cores communication
- Reticular formation
- Efficient cooperation on Mulit CPUs



How to add a rptun service

Function define:

```
rpmsg_register_callback(FAR void *priv_,
                        rpmsg_dev_cb_t device_created,
                        rpmsg_dev_cb_t device_destroy,
                        rpmsg_match_cb_t ns_match,
                        rpmsg_bind_cb_t ns_bind)
```

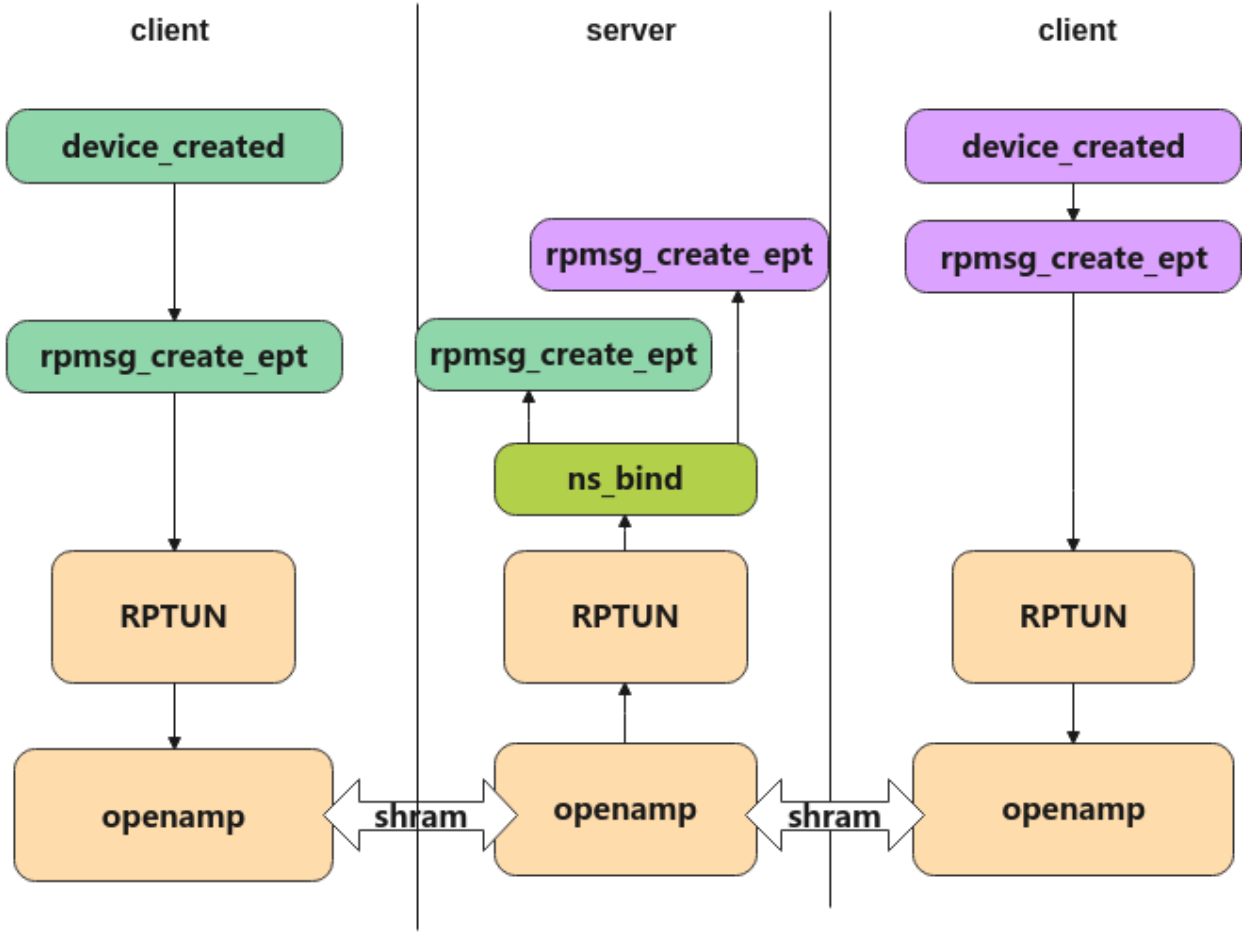
Server called:

```
rpmsg_register_callback(NULL,
                        NULL,
                        NULL,
                        syslog_rpmsg_ns_match,
                        syslog_rpmsg_ns_bind);
```

Client called:

```
rpmsg_register_callback(&g_syslog_rpmsg,
                        syslog_rpmsg_device_created,
                        syslog_rpmsg_device_destroy,
                        NULL,
                        NULL);
```

Several-to-one:
syslog rpmgfs uorb clock ...



How to add a rptun service

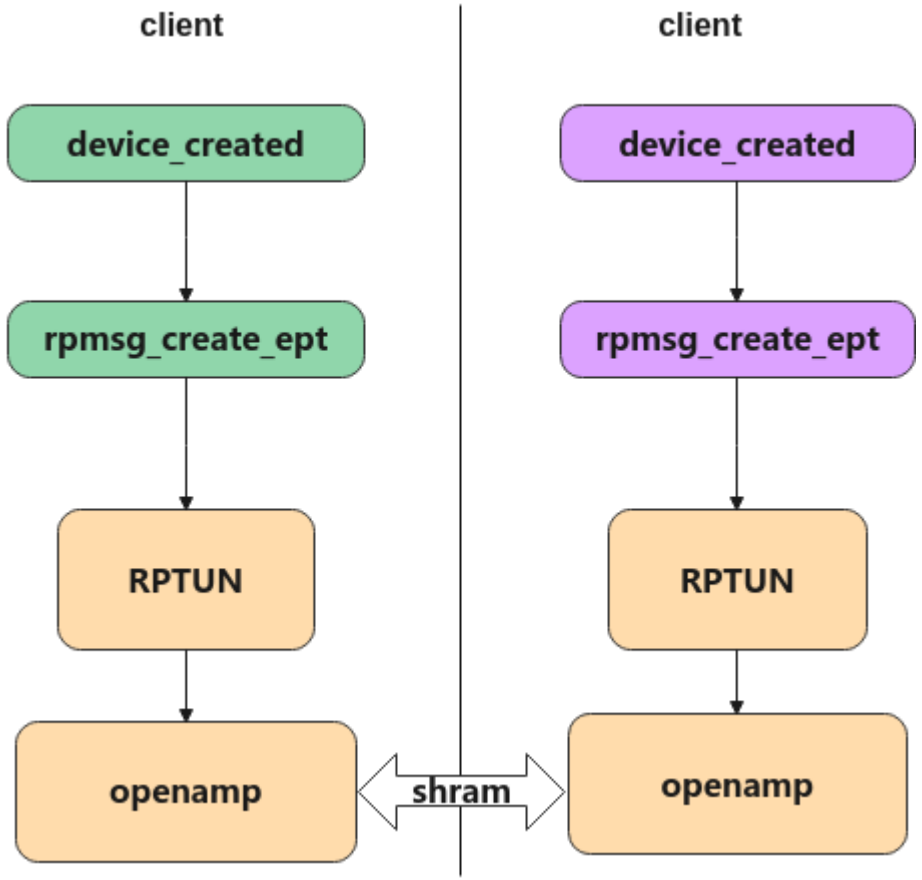
Function define:

```
rpmsg_register_callback(FAR void *priv_,
                        rpmsg_dev_cb_t device_created,
                        rpmsg_dev_cb_t device_destroy,
                        rpmsg_match_cb_t ns_match,
                        rpmsg_bind_cb_t ns_bind)
```

Both called:

```
rpmsg_register_callback(dev,
                        uart_rpmsg_device_created,
                        uart_rpmsg_device_destroy,
                        NULL,
                        NULL);
```

One-to-one:
uart ...

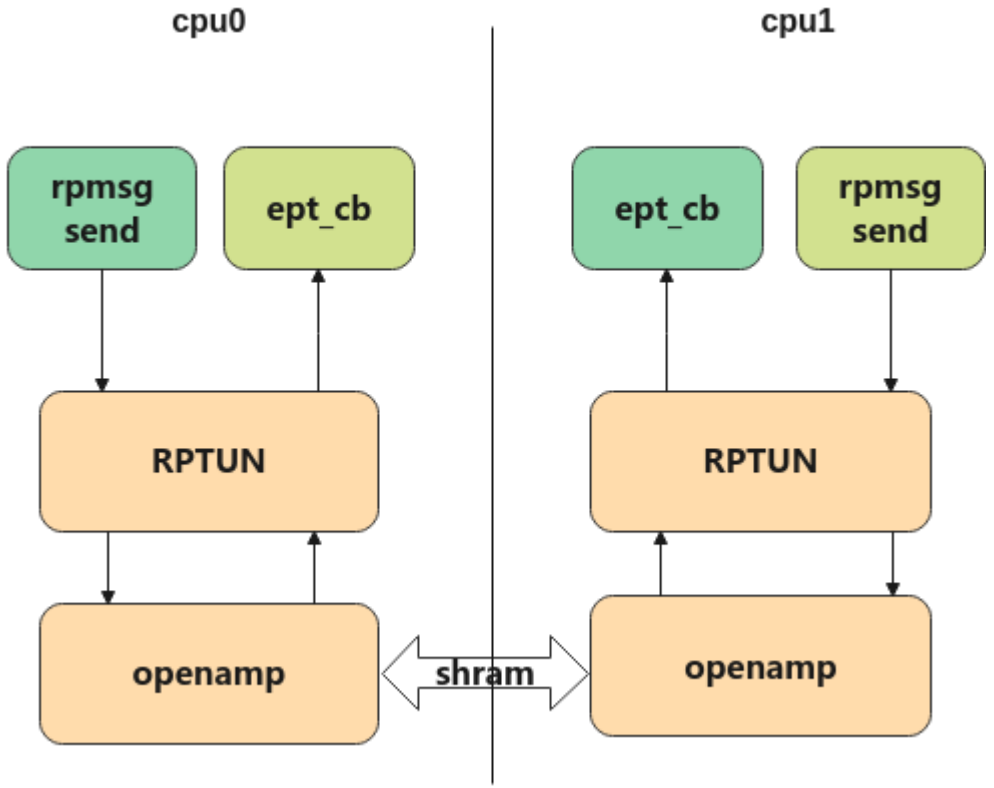


How to send/recv msg

rpmsg_send(struct rpmsg_endpoint *ept,
const void *data, int len)

rpmsg_create_ept(a, ..., rpmsg_ept_cb cb, ...)

rpmsgfs_ept_cb(struct rpmsg_endpoint *ept,
void *data, size_t len,
uint32_t src, void *priv)



Thanks

liguiding1@xiaomi.com