中国移动 China Mobile

Linux LIO 与 TCMU 用户空间透传

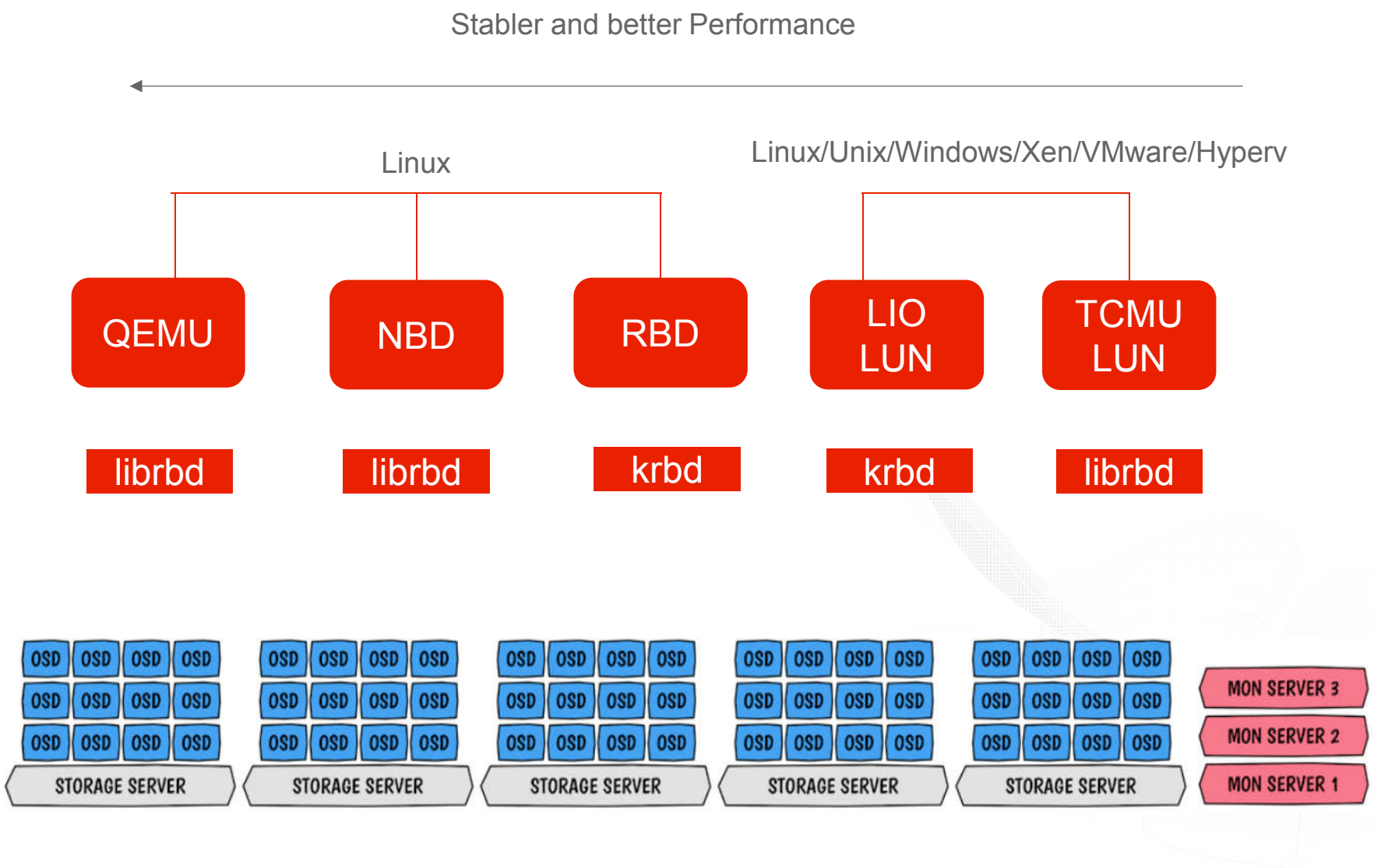
李秀波

01 **Overview of Ceph RBD iSCSI**

02 LIO, TCMU and Passthrough

03 The status of the tcmu-runner

Overview of Ceph Block Interface



Ceph RBD iSCSI Options

	LIO	TCMU	SCST	TGT
Developers	SUSE	Redhat, China Mobile, IBM	N/A	None
Mainline Kernel	No*	Yes	No	None
SDS Backend	Ceph	Ceph, GlusterFS qcow ...	N/A	Sheepdog, Ceph, GlusterFS
Advanced Features	MP/CHAP/iSER	MP/CHAP/ALUA/iSER/VAAI/ODX	N/A	None
GA ready	Yes	End of 2017	N/A	No

* LIO is in the mainline while Suse's improvements on krbd are not yet.

Why China Mobile need iSCSI solution

- Legancy applications on IP/FC-SAN, VMware (70%+), Hyperv, Xen...
- Advanced features:
  - Multipath & load balance (Active/Active)
  - VMware VAAI (vStorage APIs for Array Integration)
  - Windows ODX (Offloaded Data Transfer)

Our contributions:

1.TCMU Ceph engine

2.Ceph side VAAI native support

3.Industrial logger & configuration framework

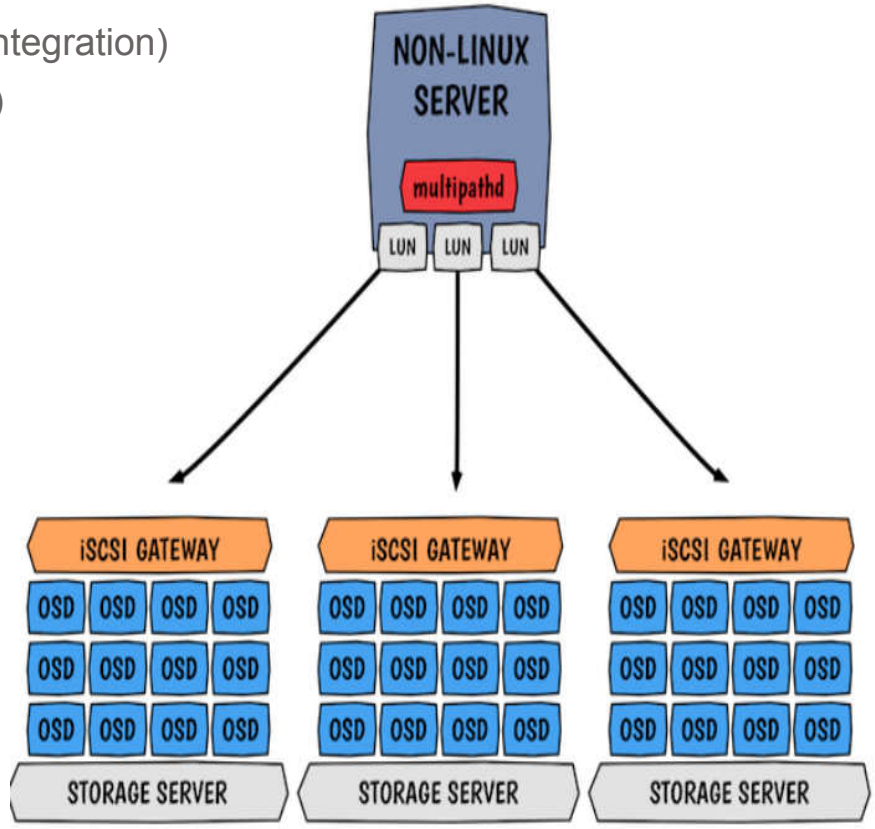
4.VMware VAAI feature

5.Windows ODX feature

6.Multipath enhancement

7.Kernel module ring buffer scalability

8.Tons of bug fixes and code refactor…



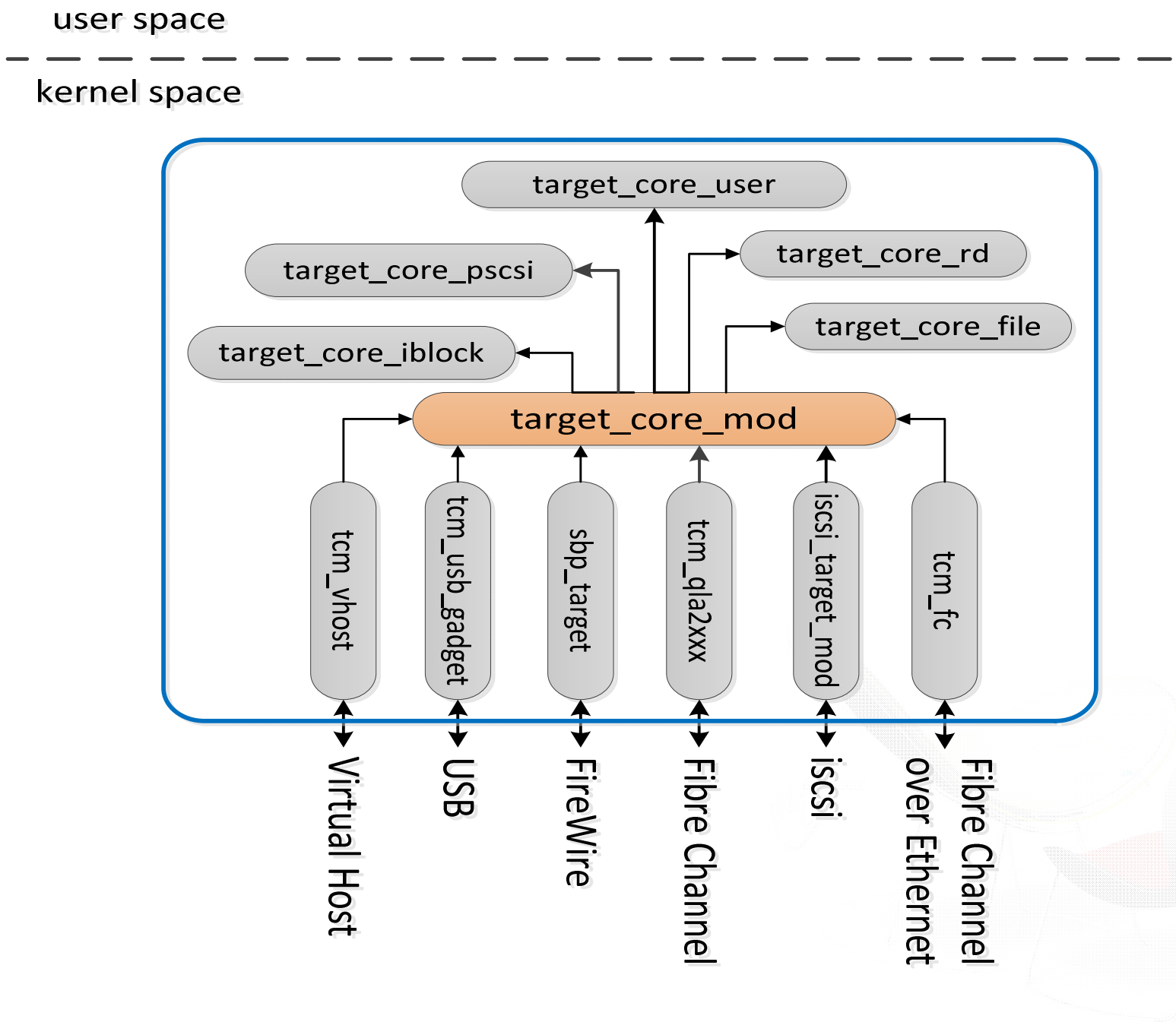
01 Overview of Ceph RBD iSCSI

02 **LIO, TCMU and Passthrough**

03 The status of the tcmu-runner

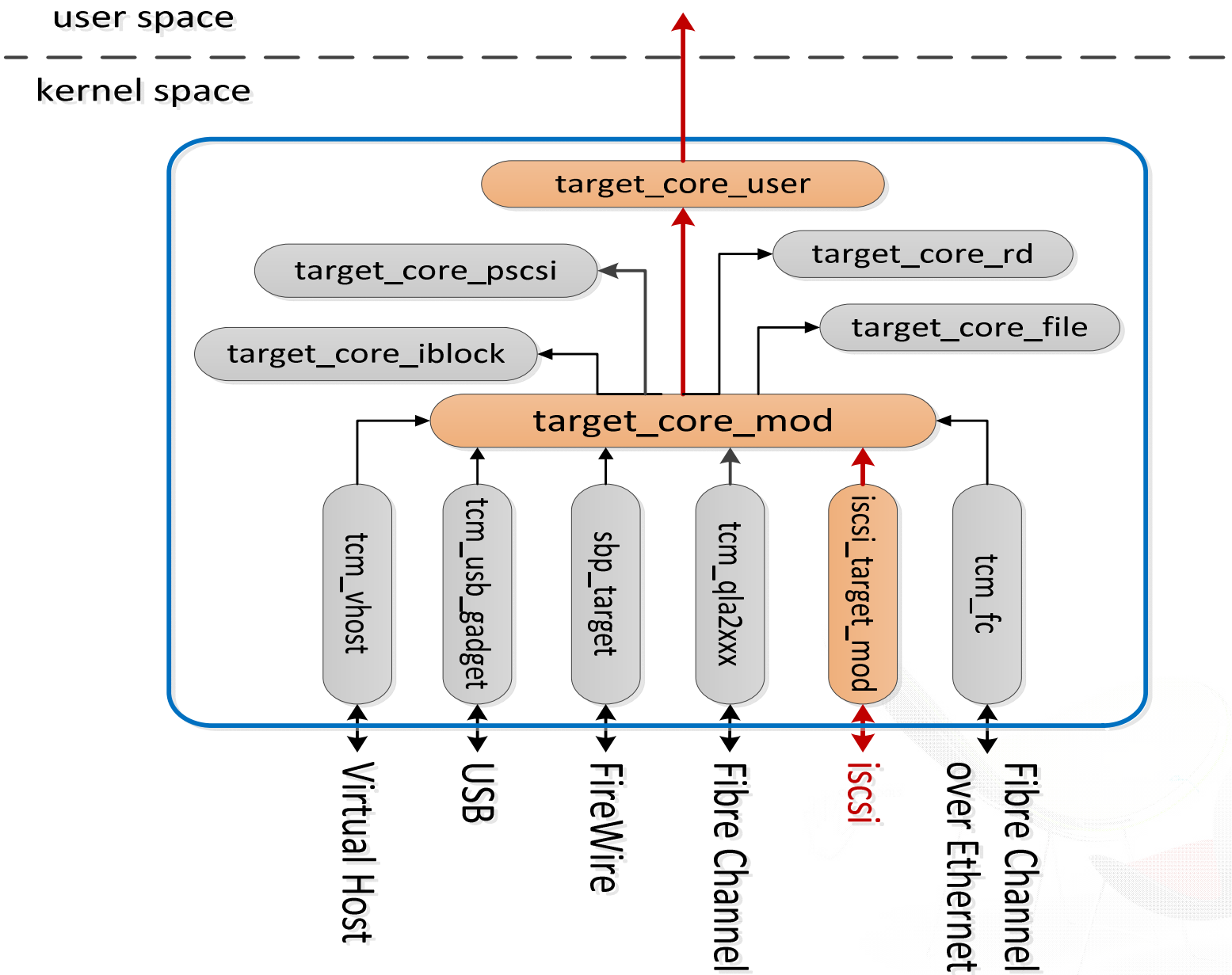
Linux IO(LIO) ?

LIO(**L**inux **IO**) is an implementation of SCSI target.



TCM, TCMU ?

TCM is another name for LIO, TCMU is the TCM in Userspace.

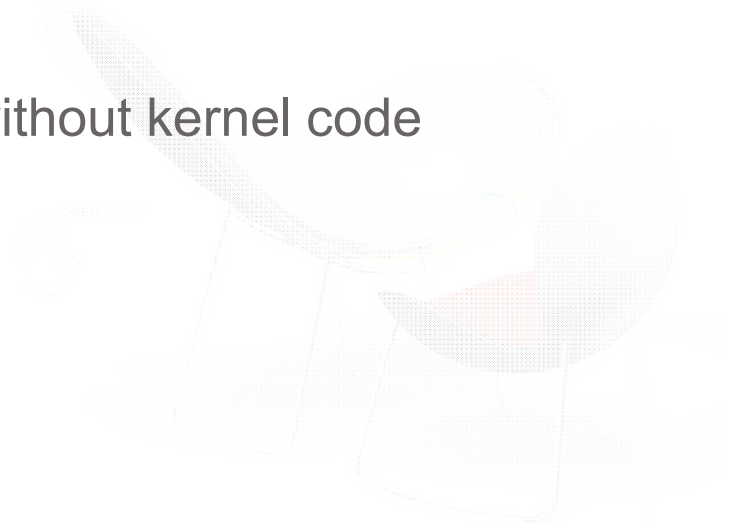

user space
kernel space
target_core_user
target_core_pscsi
target_core_rd
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
Virtual Host
USB
FireWire
Fibre Channel
iscsi
Fibre Channel over Ethernet

Why Passthrough to Userspace?

Qemu is in userspace and is capable of accessing storage locally and remotely using various protocol drivers.

Librbd in userspace is the latest and more active than krbd.ko

Enables wider variety of backstores without kernel code

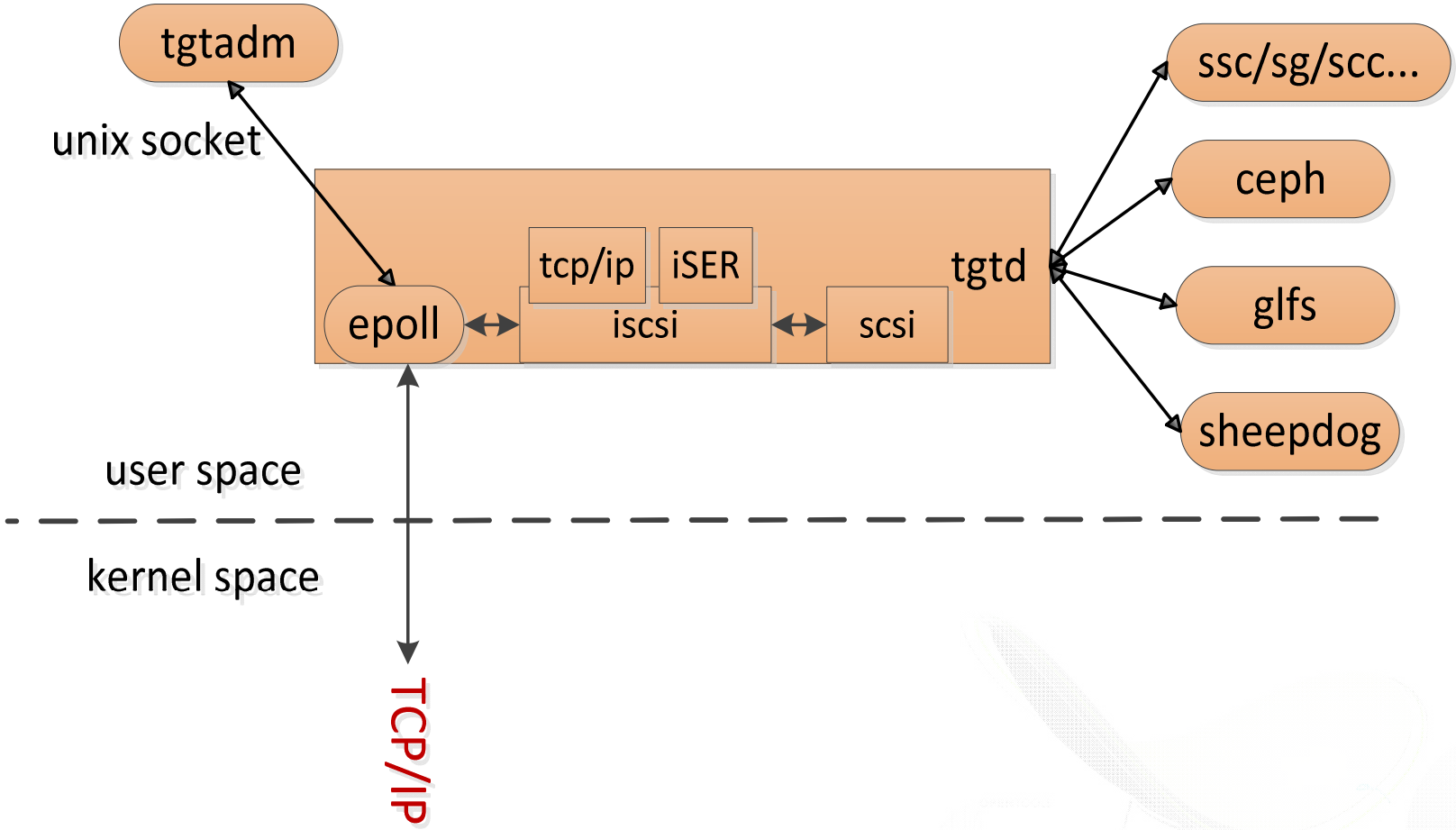


Passthrough what ?

Only SCSI commands with their Datas, not Iscsi commands.

Why not Iscsi directly ?

That will like what the stgt does, and then we couldn't take advantage of the LIO's mess features.


tgtadm
unix socket
tcp/ip
iSER
tgtd
epoll
iscsi
scsi
ssc/sg/scc...
ceph
glfs
sheepdog
user space
kernel space
TCP/IP

Why not STGT ?

Time has proved that STGT is too weak to satisfy modern storage requirements.

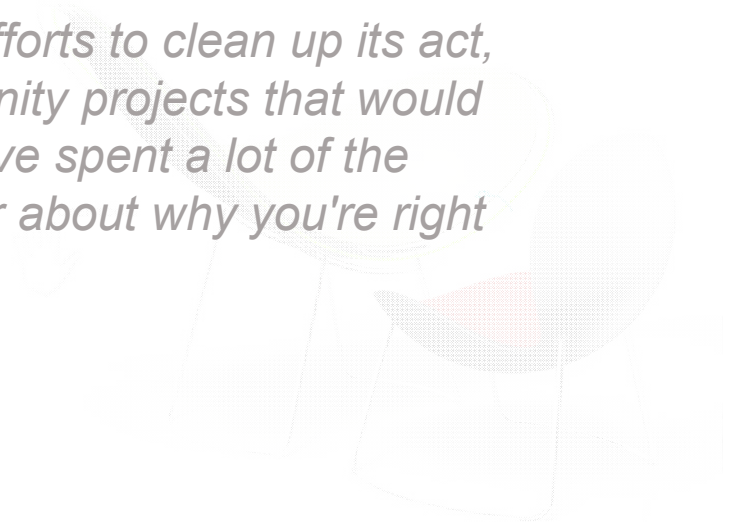
Now it is obsolete and has been removed from the mainline kernel.

Then why not SCST ?

Thought SCST is far more mature as a general purpose target, it was abandoned by James, the SCSI maintainer.

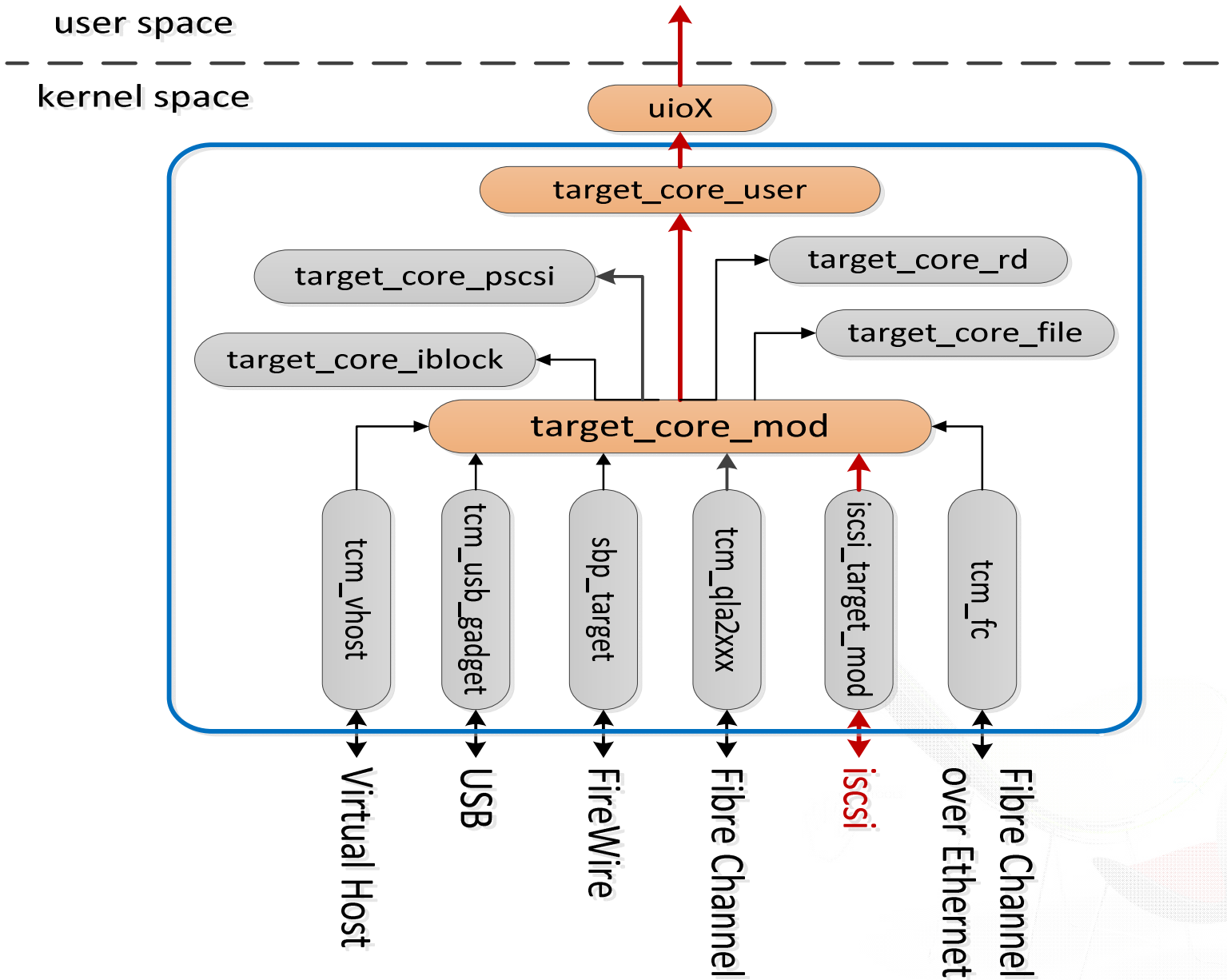
“This isn't a democracy ... it's about choosing the most community oriented code base so that it's easily maintainable and easy to add feature requests and improvements as and when they come along. ”

“In the past six months, LIO has made genuine efforts to clean up its act, streamline its code and support the other community projects that would need to go above and around it. You seem to have spent a lot of the intervening time arguing with the sysfs maintainer about why you're right and he's wrong.”



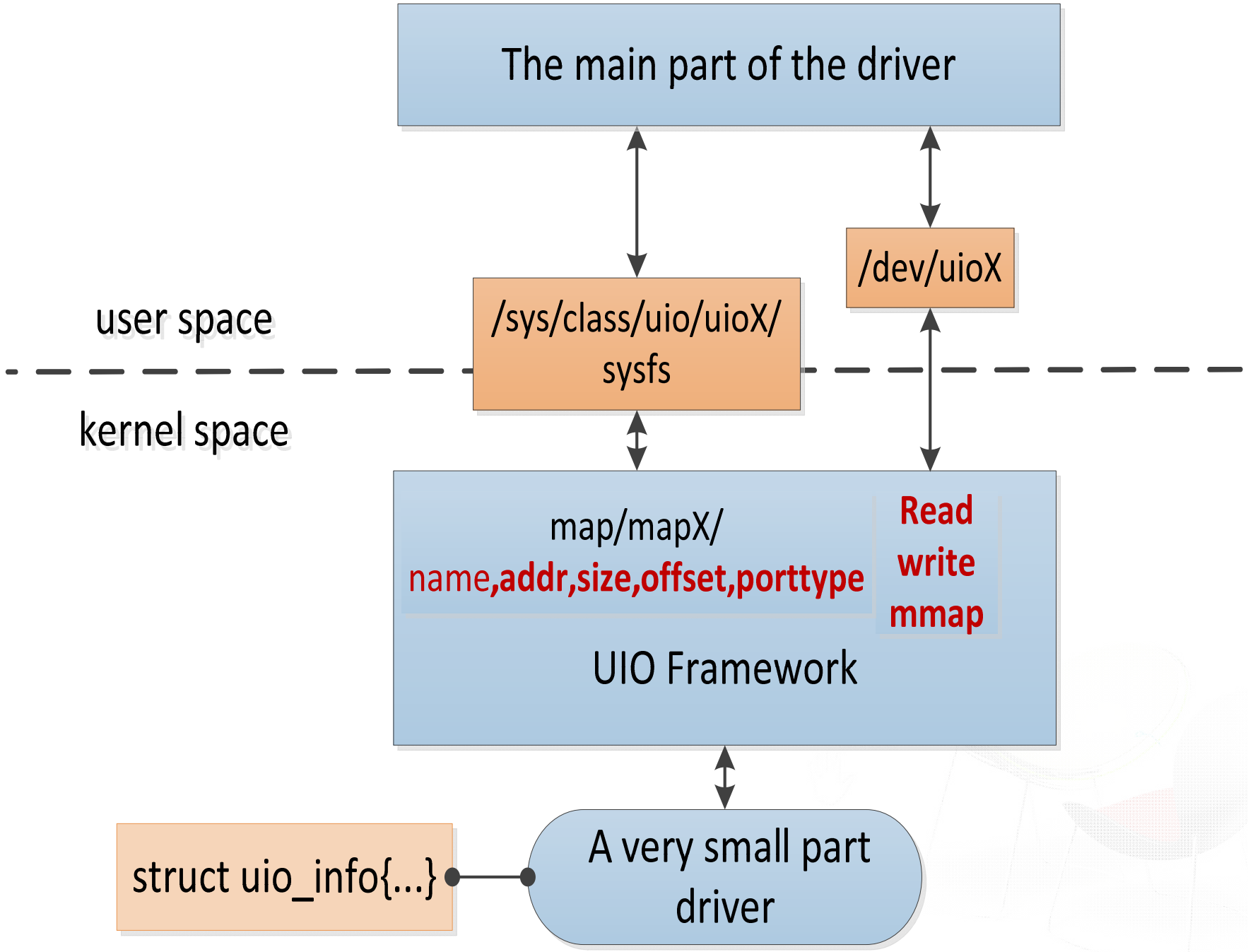
How to Passthrough to Userspace?

TCMU utilizes the traditional UIO subsystem, which is designed to allow device driver development in userspace


user space
kernel space
uioX
target_core_user
target_core_pscsi
target_core_rd
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
Virtual Host
USB
FireWire
Fibre Channel
iscsi
Fibre Channel over Ethernet

What's UIO ?

The Userspace I/O, which allows to implement the device driver in userspace.


The main part of the driver
/dev/uioX
/sys/class/uio/uioX/
sysfs
user space
kernel space
map/mapX/
name,addr,size,offset,porttype
Read
write
mmap
UIO Framework
struct uio_info{...}
A very small part
driver

struct uio_info {}

```
 81 /**
 82  * struct uio_info - UIO device capabilities
 83  * @uio_dev:            the UIO device this info belongs to
 84  * @name:               device name
 85  * @version:            device driver version
 86  * @mem:                list of mappable memory regions, size==0 for end of list
 87  * @port:               list of port regions, size==0 for end of list
 88  * @irq:                interrupt number or UIO_IRQ_CUSTOM
 89  * @irq_flags:          flags for request_irq()
 90  * @priv:               optional private data
 91  * @handler:            the device's irq handler
 92  * @mmap:               mmap operation for this uio device
 93  * @open:               open operation for this uio device
 94  * @release:            release operation for this uio device
 95  * @irqcontrol:         disable/enable irqs when 0/1 is written to /dev/uioX
 96  */
 97 struct uio_info {
 98         struct uio_device       *uio_dev;
 99         const char              *name;
100         const char              *version;
101         struct uio_mem          mem[MAX_UIO_MAPS];
102         struct uio_port         port[MAX_UIO_PORT_REGIONS];
103         long                    irq;
104         unsigned long           irq_flags;
105         void                    *priv;
106         irqreturn_t (*handler)(int irq, struct uio_info *dev_info);
107         int (*mmap)(struct uio_info *info, struct vm_area_struct *vma);
108         int (*open)(struct uio_info *info, struct inode *inode);
109         int (*release)(struct uio_info *info, struct inode *inode);
110         int (*irqcontrol)(struct uio_info *info, s32 irq_on);
111 };
```

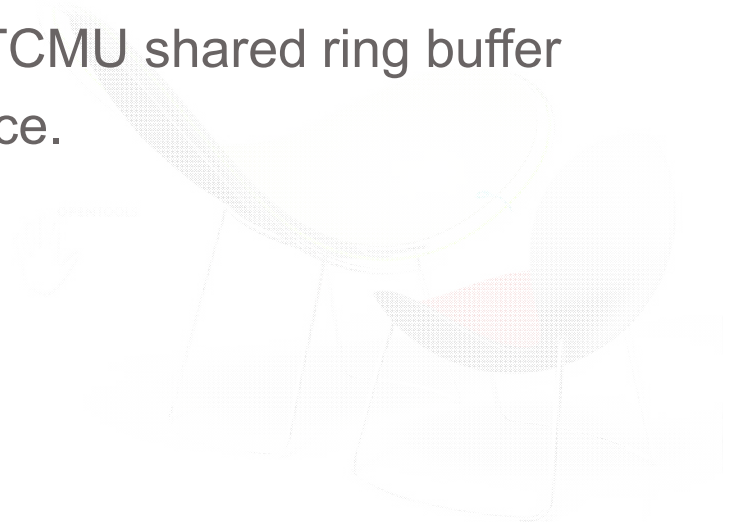
mmap(/dev/uioX) what?

addr = /sys/class/uio/uio0/maps/map0/addr

size = /sys/class/uio/uio0/maps/map0/size

off = /sys/class/uio/uio0/maps/map0/offset

Here mmap(addr, size, ..., off) will map the TCMU shared ring buffer from kernel to userspace.



TCMU Ring Buffer ?

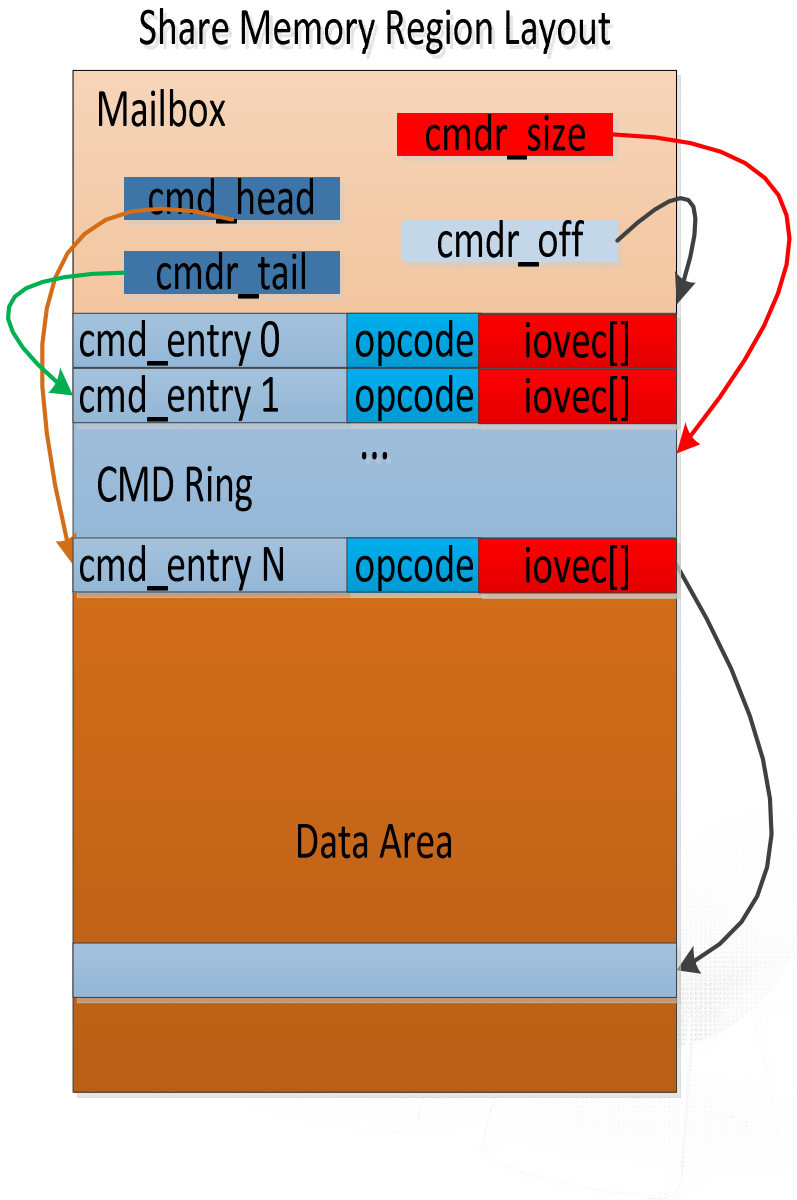
mailbox

cmd area(bitmap)

data area(bitmap --> dynamic)

Old Ring Buffer ?

```
68 #define DATA_BLOCK_BITS 256
69 #define DATA_BLOCK_SIZE 4096
70
71 #define CMDR_SIZE (16 * 4096)
72 #define DATA_SIZE (DATA_BLOCK_BITS * DATA_BLOCK_SIZE)
73
74 #define TCMU_RING_SIZE (CMDR_SIZE + DATA_SIZE)
```



The **total size** of the ring buffer is fixed to:

(256+16) * 4096 = 1M + 64K

sizeof(Mailbox + CMD Ring) = 64K

sizeof(Data Area) = 1M

New Ring Buffer ?

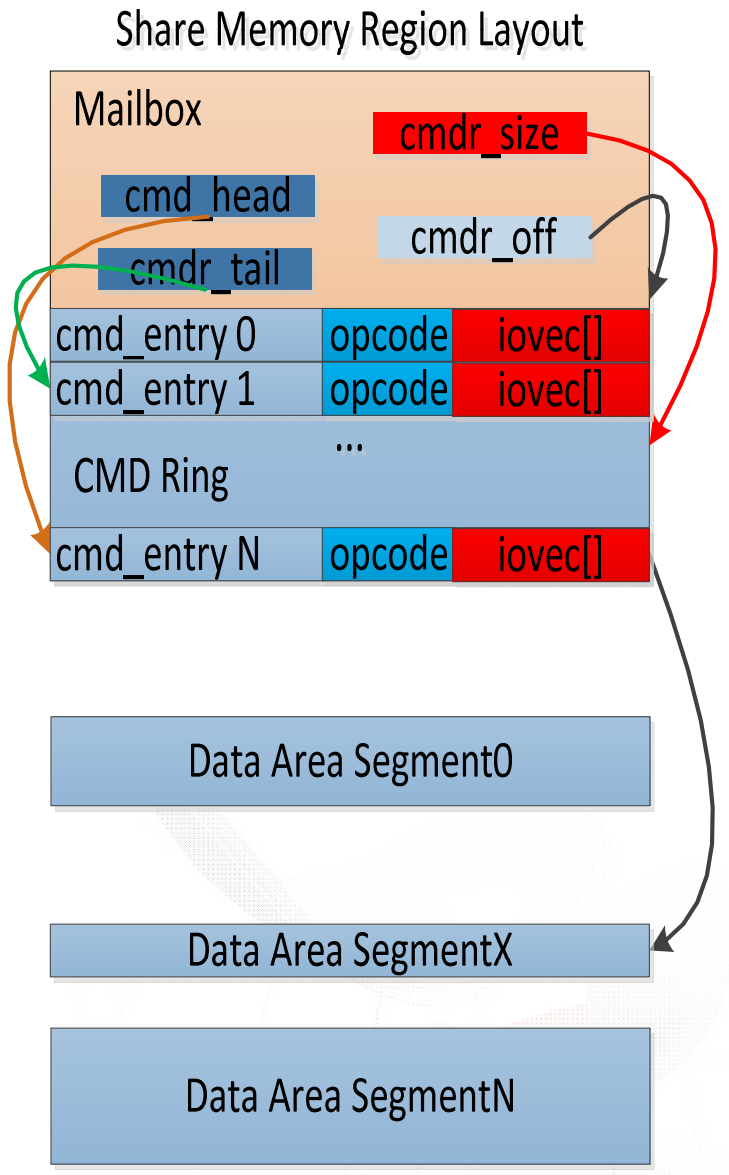
```
72 /* For cmd area, the size is fixed 8MB */
73 #define CMDR_SIZE (8 * 1024 * 1024)
74
75 /*
76  * For data area, the block size is PAGE_SIZE and
77  * the total size is 256K * PAGE_SIZE.
78  */
79 #define DATA_BLOCK_SIZE PAGE_SIZE
80 #define DATA_BLOCK_BITS (256 * 1024)
81 #define DATA_SIZE (DATA_BLOCK_BITS * DATA_BLOCK_SIZE)
82 #define DATA_BLOCK_INIT_BITS 128
83
84 /* The total size of the ring is 8M + 256K * PAGE_SIZE */
85 #define TCMU_RING_SIZE (CMDR_SIZE + DATA_SIZE)
86
87 /* Default maximum of the global data blocks(512K * PAGE_SIZE) */
88 #define TCMU_GLOBAL_MAX_BLOCKS (512 * 1024)
```

The **total size** of the ring buffer is variable from to:

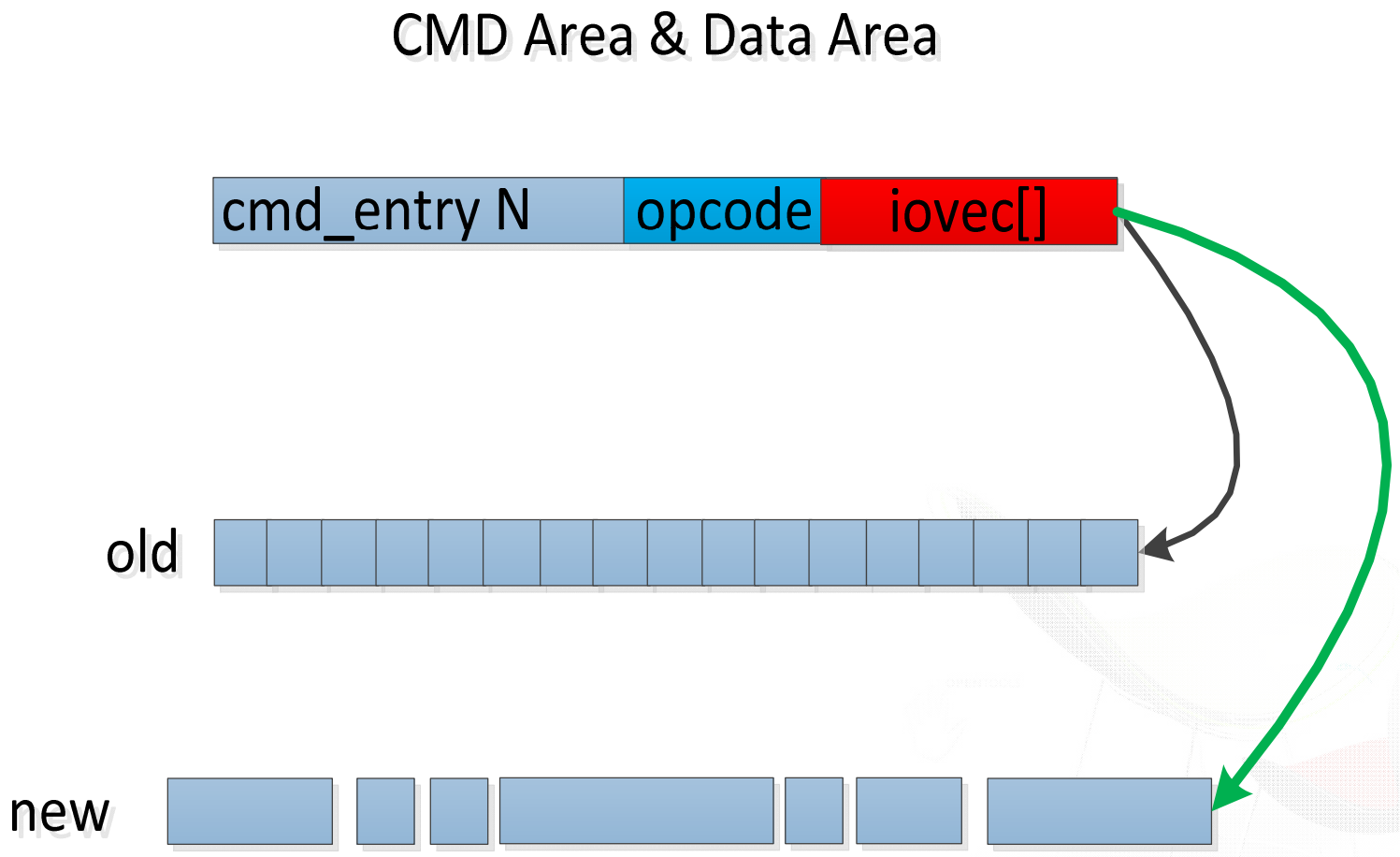
8M ~ 8M + 256K * PAGE_SIZE

sizeof(Mailbox + CMD Ring) = 8M

*sizeof(Data Area) = 256K * PAGE_SIZE*



CMD & Data Area Improvement ?



CMD Area TODO?

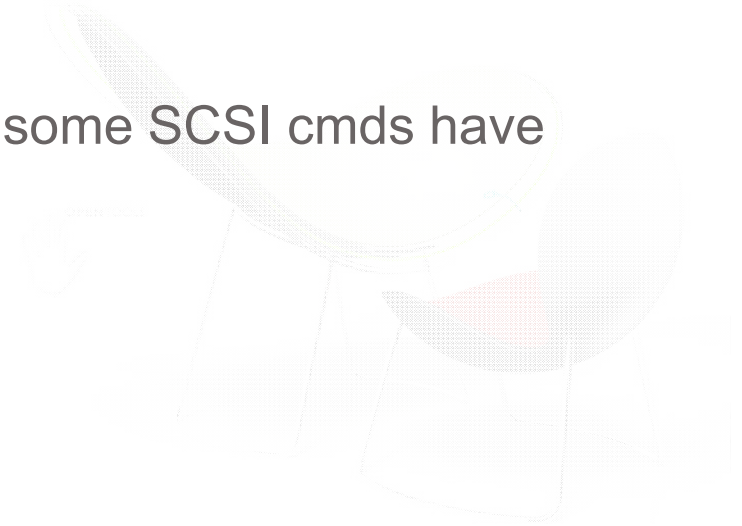
The CMD Area size is FIXED to 8MB for now, and needs one way to support dynamic grow/shrink like the DATA Area does.

IRQ emulate: read/write(/dev/uioX) ?

1, read() will be blocked until there has new SCSI cmds come, then the consumer will continue to read cmds from ring buffer.

2, each ucmd->done will update the results of the cmd to ring buffer.

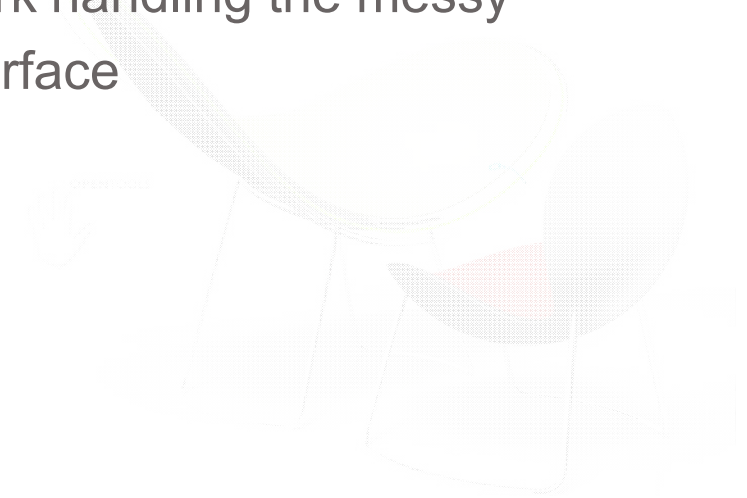
3,write() will only used to tell the TCMU that some SCSI cmds have been handled done(success or fail)

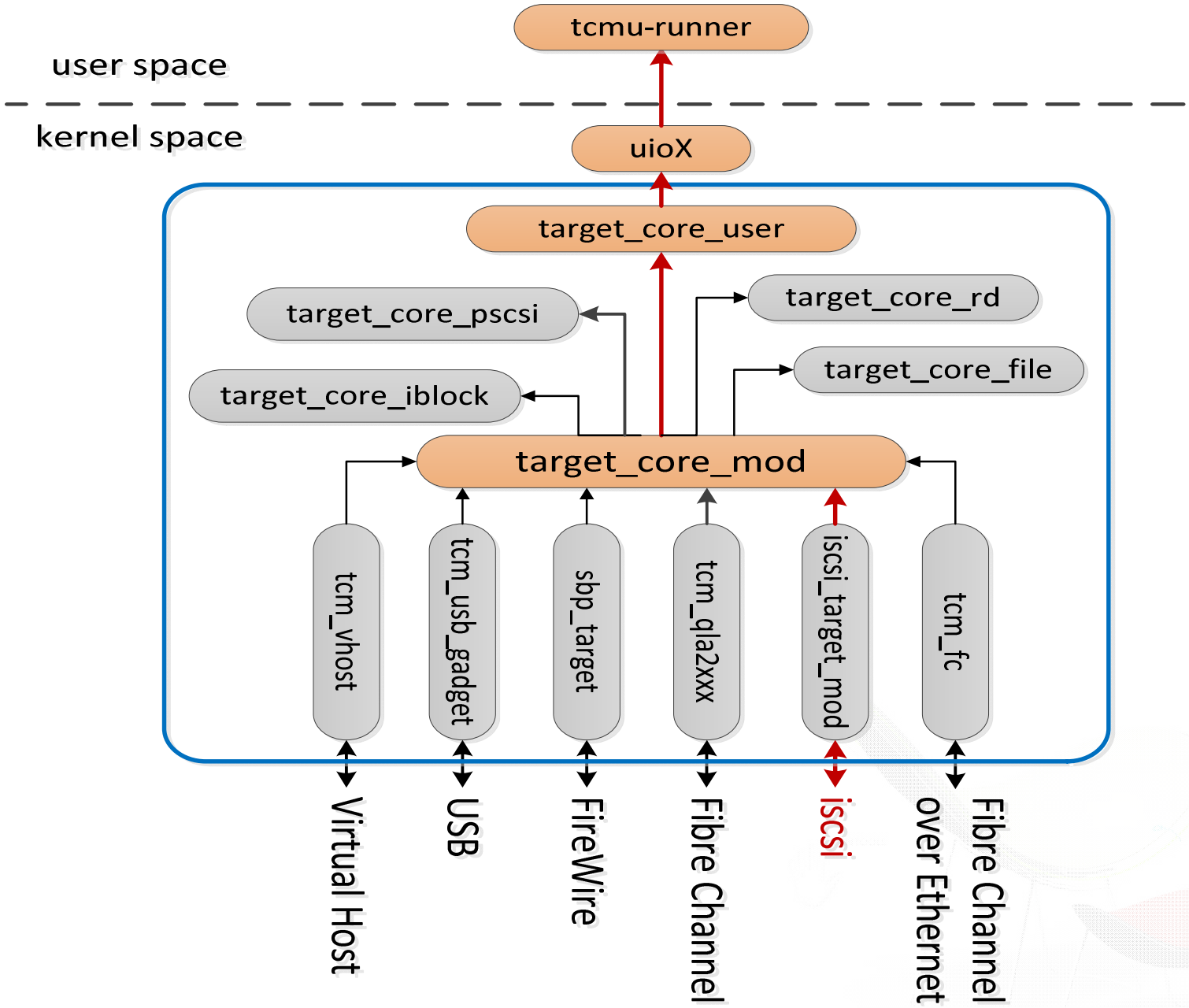


To Whom?

Tcmu-runner actually is another small SCSI target in userspace, very similar to TCM in kernel space.

Tcmu-runner utilizes the TCMU framework handling the messy details of the TCMU interface



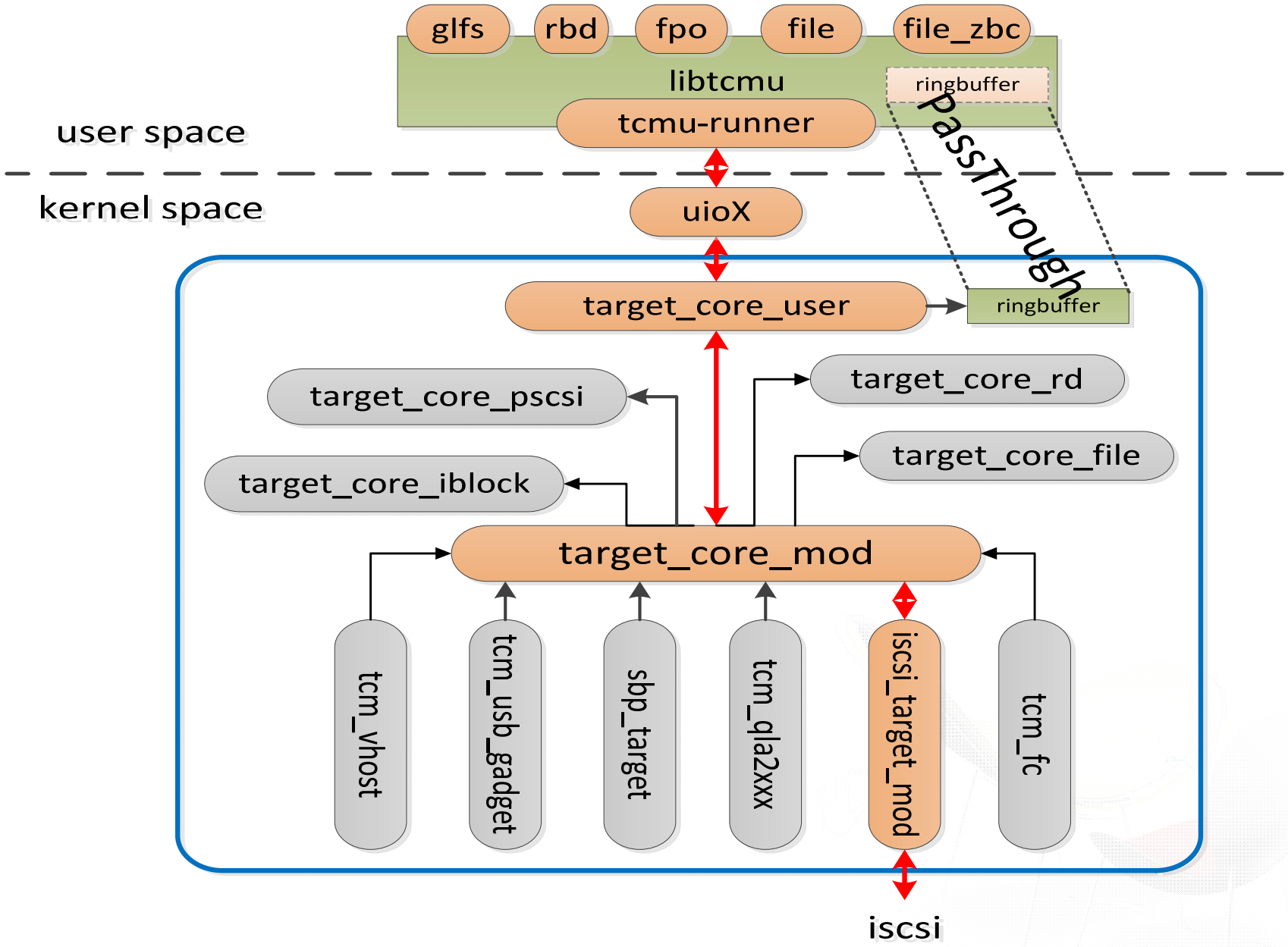

tcmu-runner
user space
kernel space
uioX
target_core_user
target_core_pscsi
target_core_rd
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
Virtual Host
USB
FireWire
Fibre Channel
iscsi
Fibre Channel over Ethernet

What tcmu-runner will do ?

Reads SCSI commands from mmaped TCMU Ring Buffer

Handles SCSI commands to specified handlers, such as rbd/glfs...

Update the results to the TCMU Ring Buffer


glfs
rbd
fpo
file
file_zbc
libtcmu
ringbuffer
tcmu-runner
user space
kernel space
uioX
PassThrough
target_core_user
ringbuffer
target_core_pscsi
target_core_rd
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
iscsi

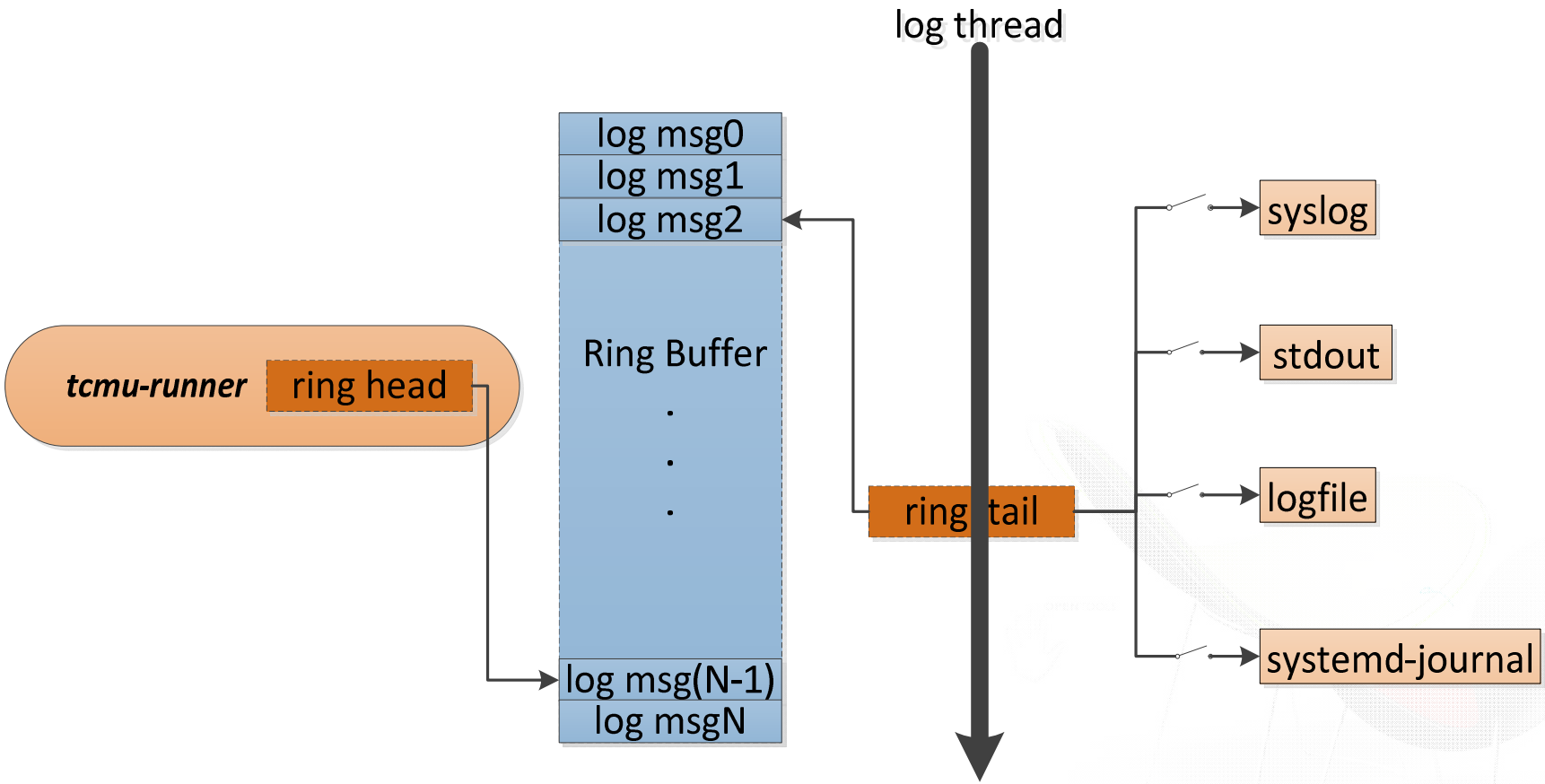
01 Overview of Ceph RBD iSCSI

02 LIO, TCMU and Passthrough

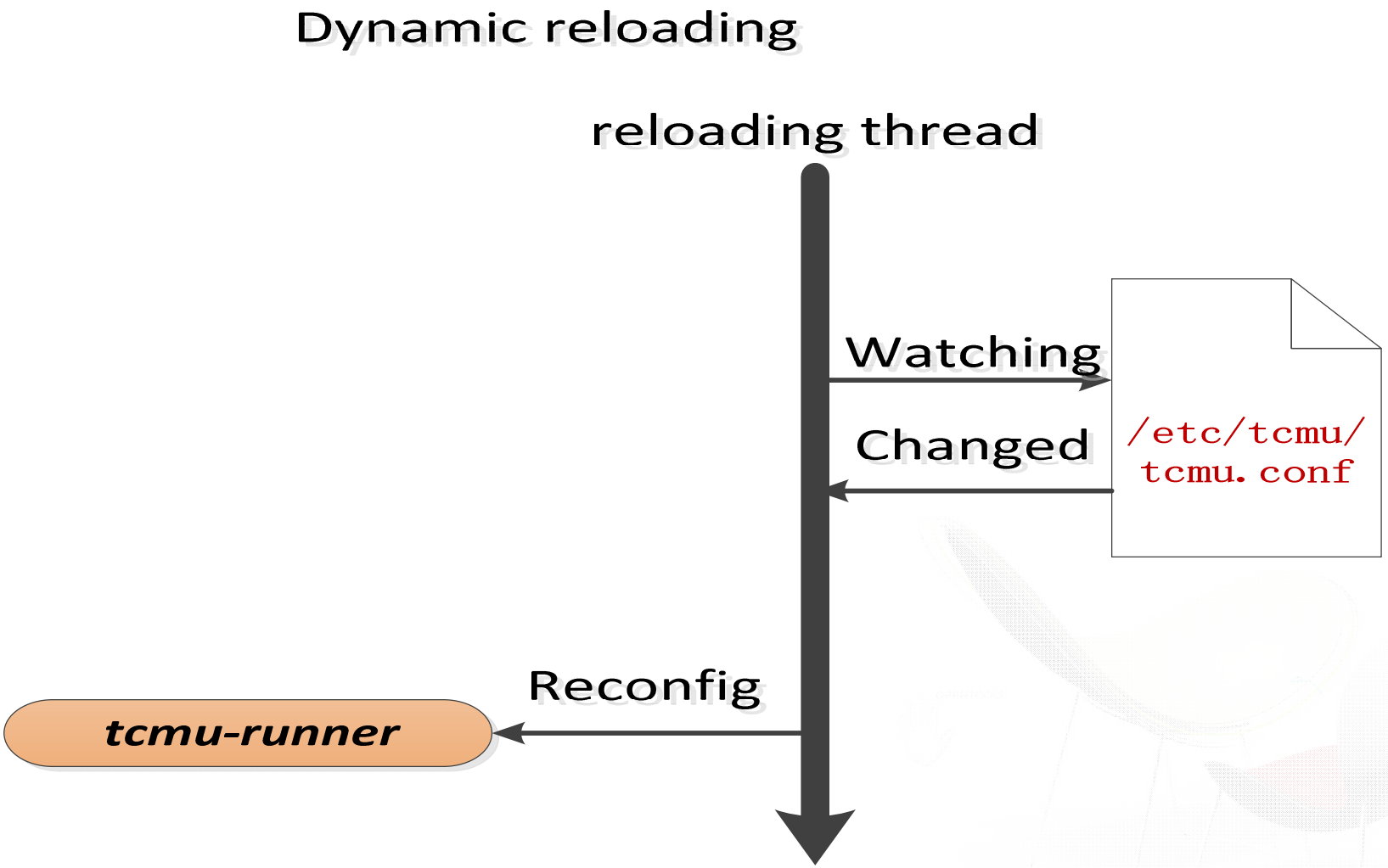
03 **The status of the tcmu-runner**

Logger system

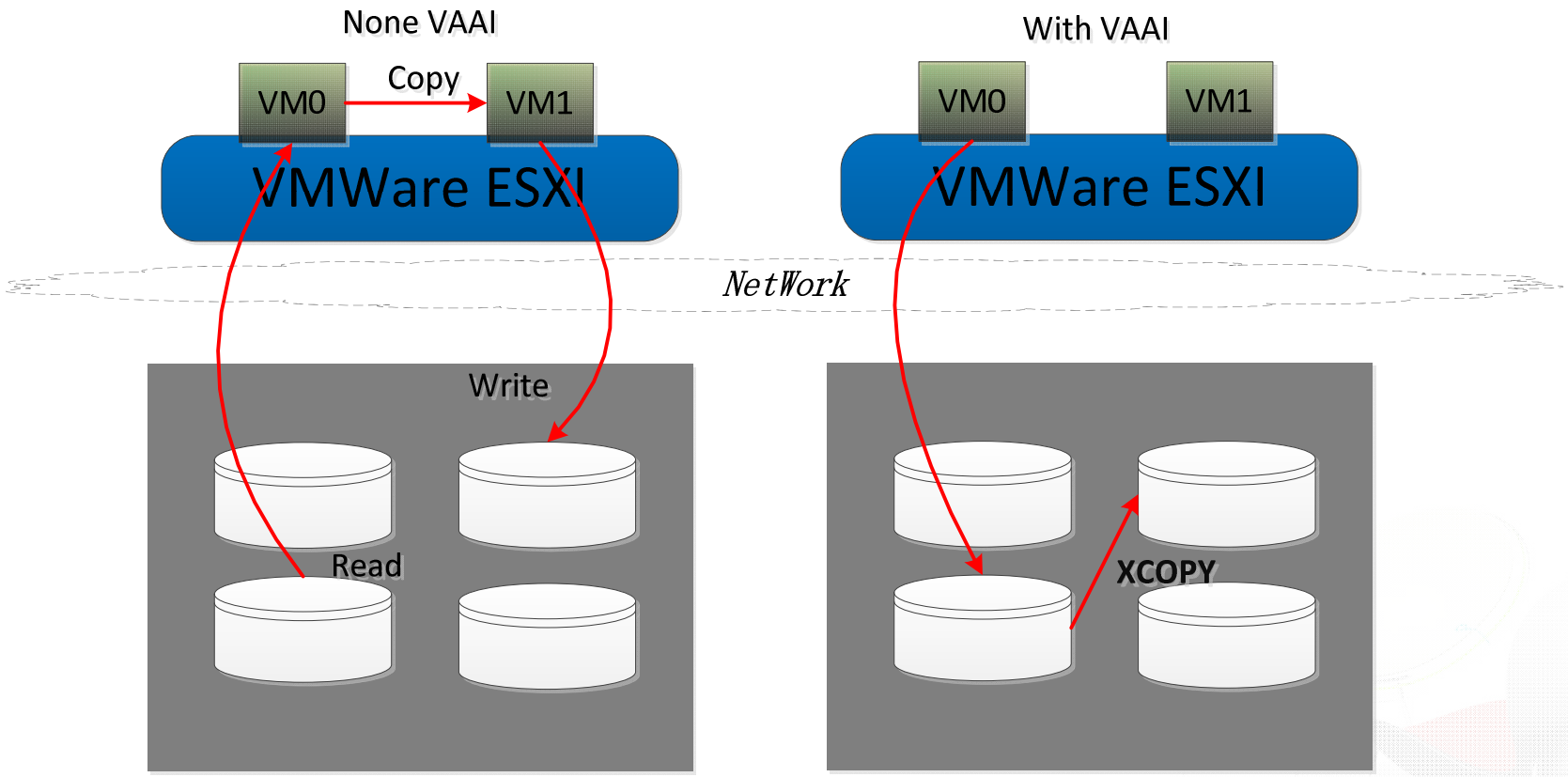
Non-block logger system



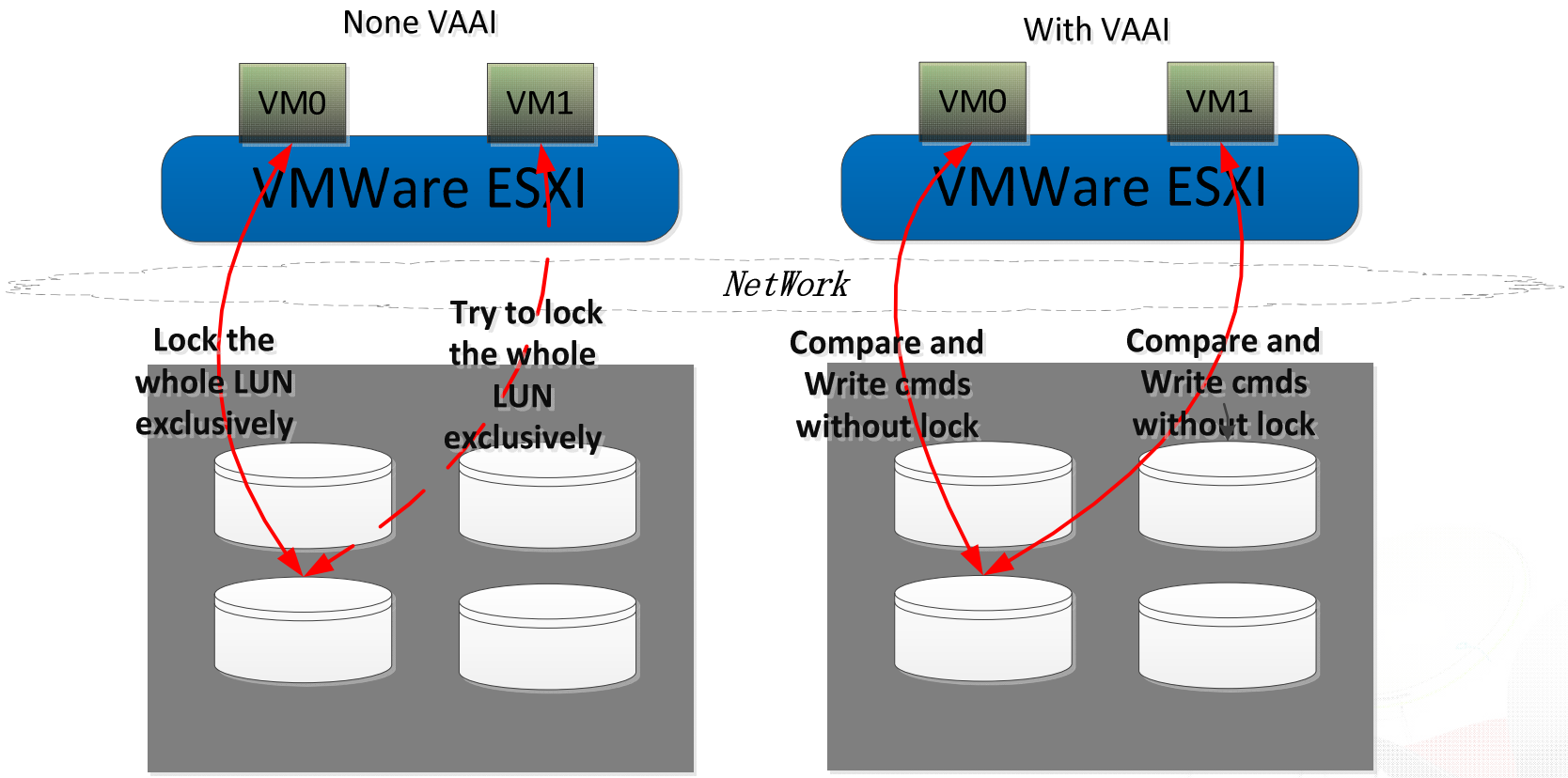
Dynamic config system



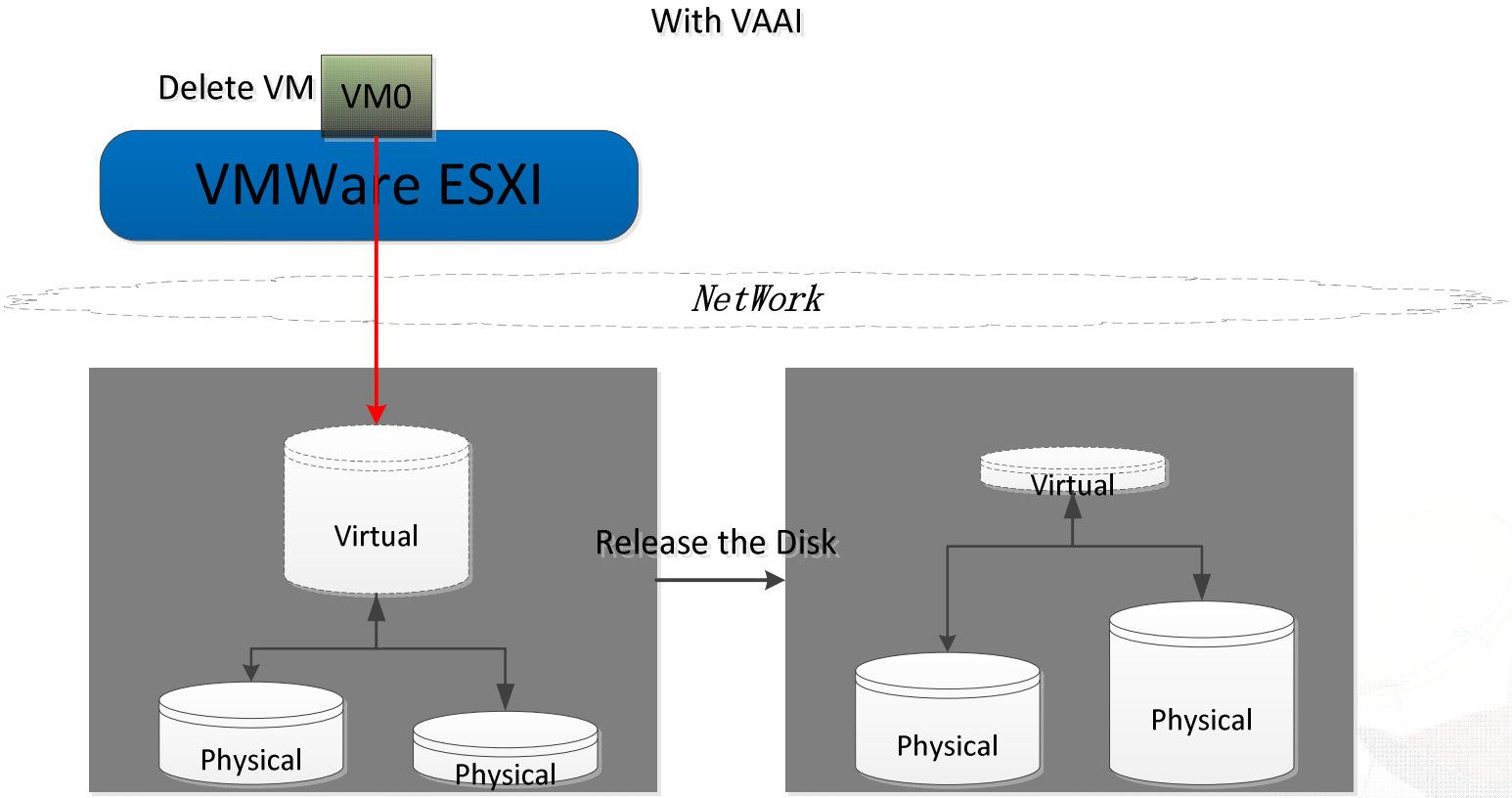
VMWare VAAI XCOPY primitive support



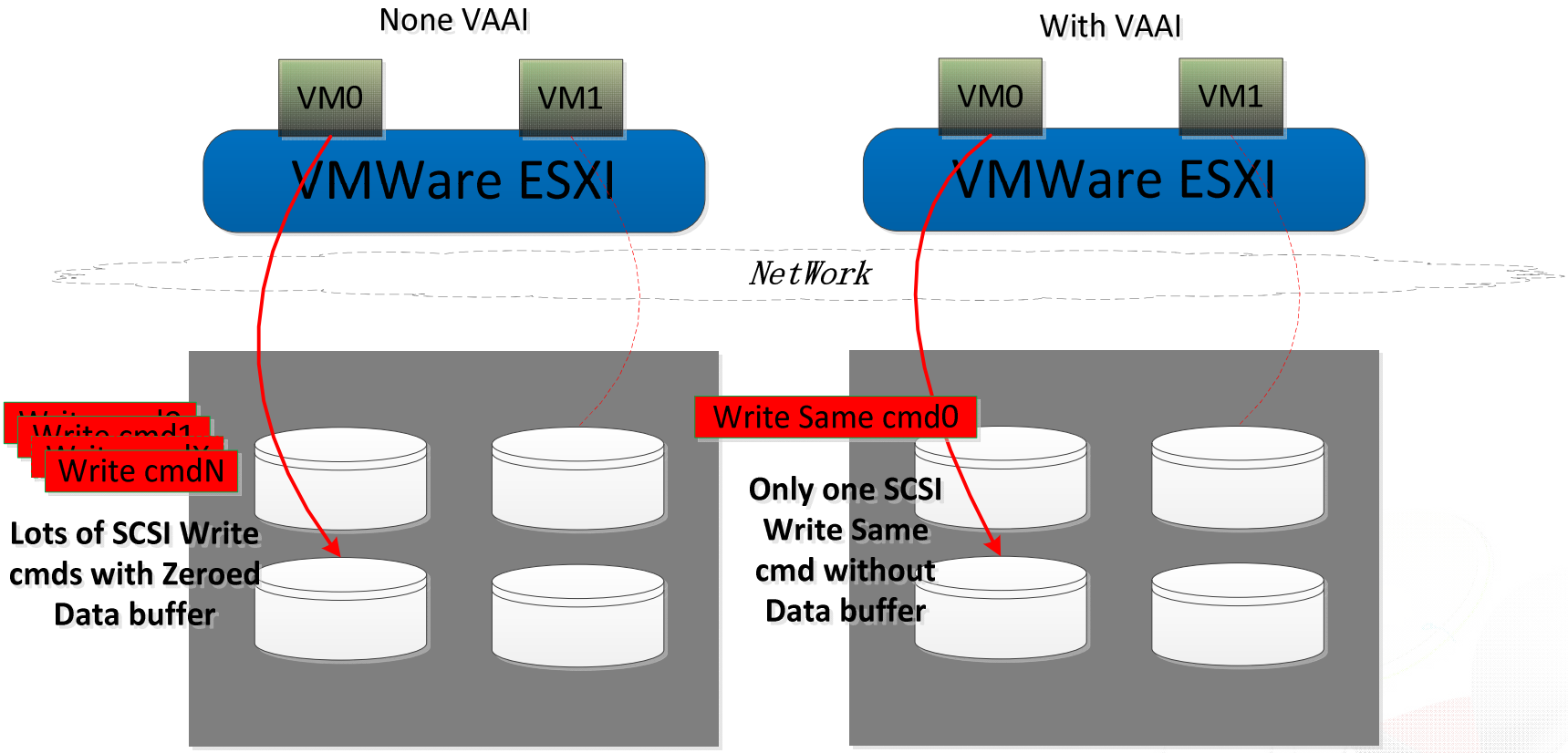
VMWare VAAI ATS primitive support



VMWare VAAI UNMAP primitive support



VMWare VAAI ZERO primitive support



Tcmu-runner daemon dynamic upgrade

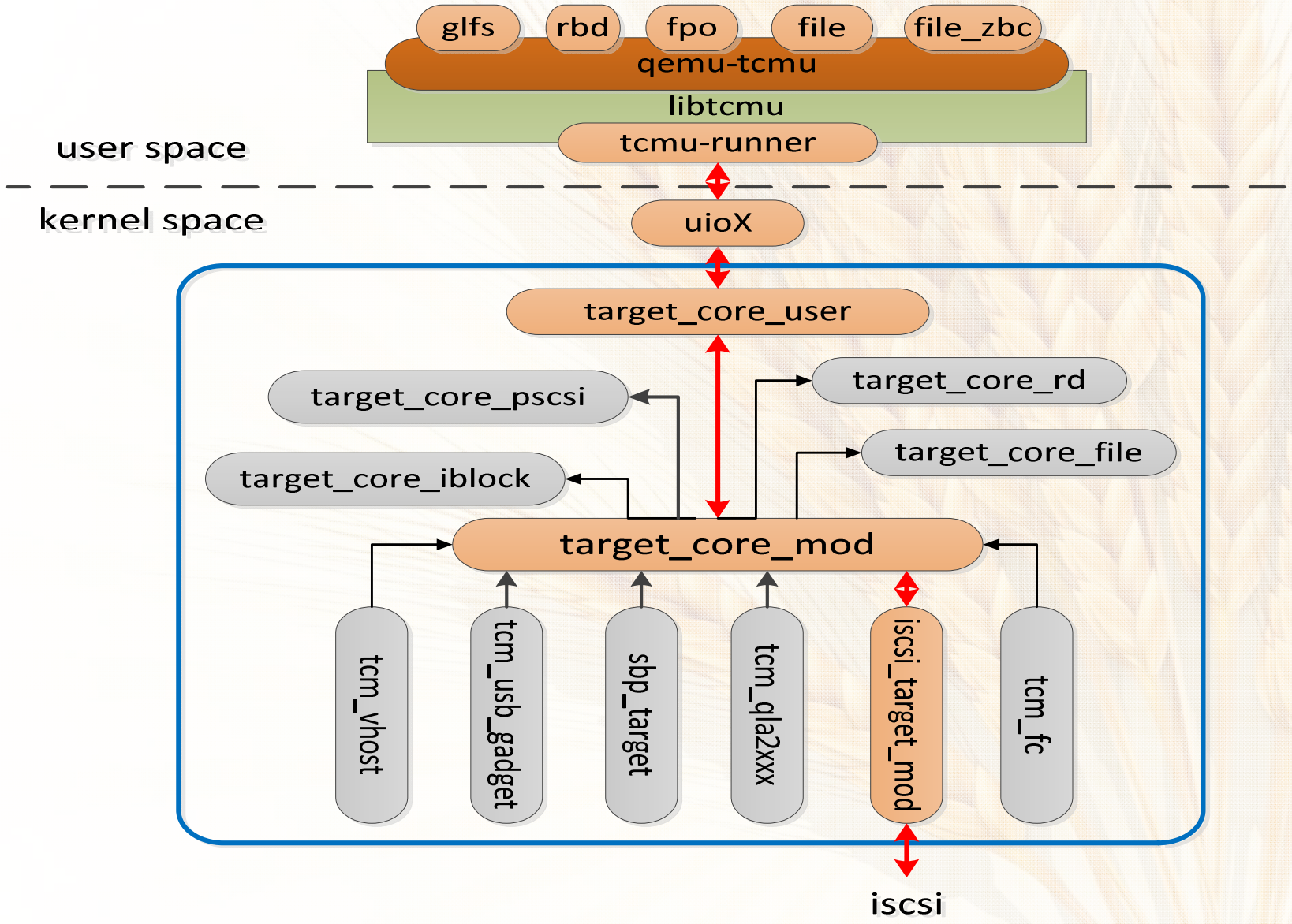
Restart the tcmu-runner.service without interrupting the service

Failover and Failback & ALUA support

For now only **Implicit** transition support

qemu-tcmu handler ?

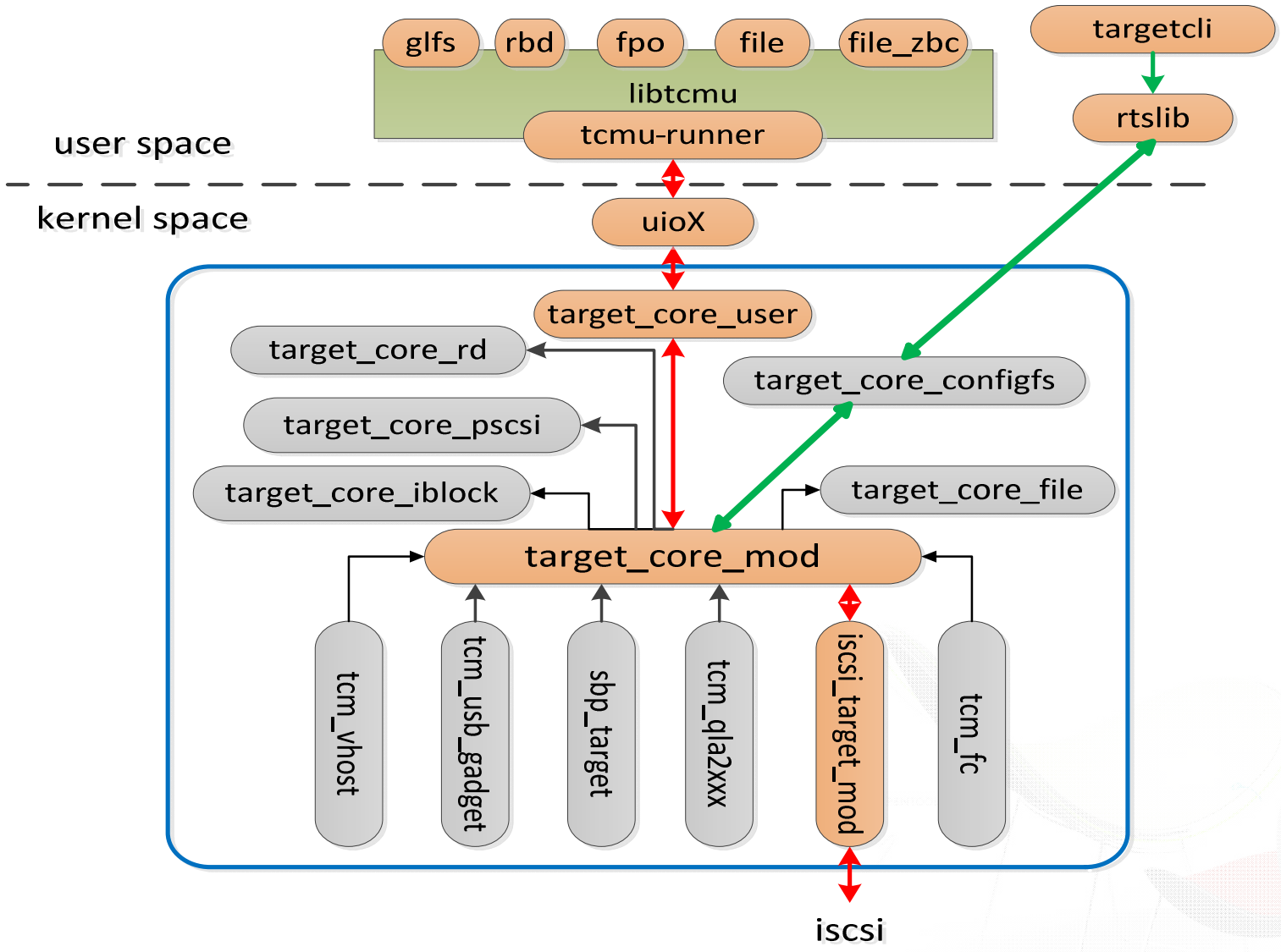
The third version patch set will be done soon by @Yaowei Bai.


glfs
rbd
fpo
file
file_zbc
qemu-tcmu
libtcmu
tcmu-runner
user space
kernel space
uioX
target_core_user
target_core_pscsi
target_core_rd
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
iscsi

Target config tools

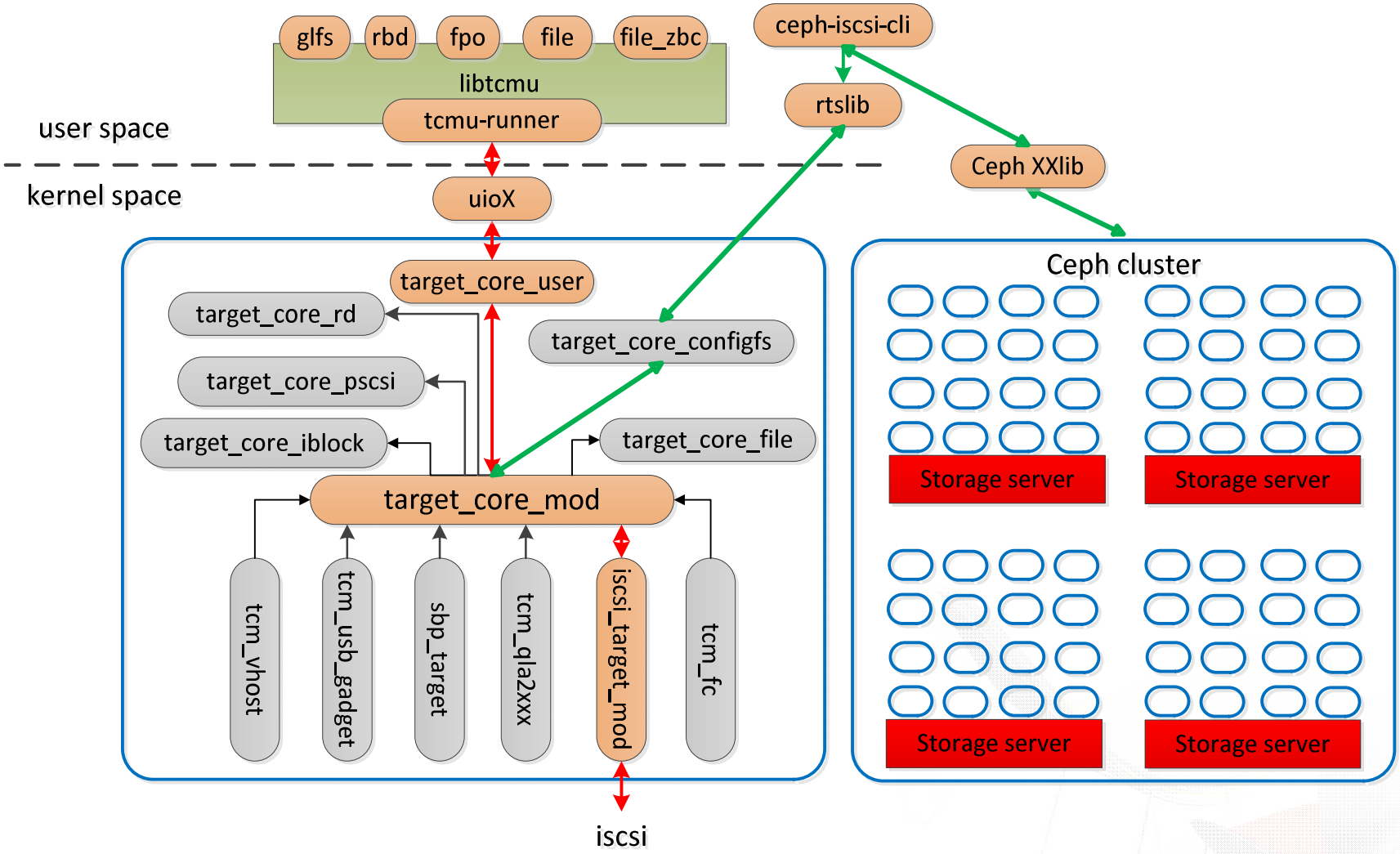
targetcli

ceph iscsi gateway tools


glfs
rbd
fpo
file
file_zbc
libtcmu
tcmu-runner
targetcli
rtslib
user space
kernel space
uioX
target_core_user
target_core_rd
target_core_configfs
target_core_pscsi
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
iscsi

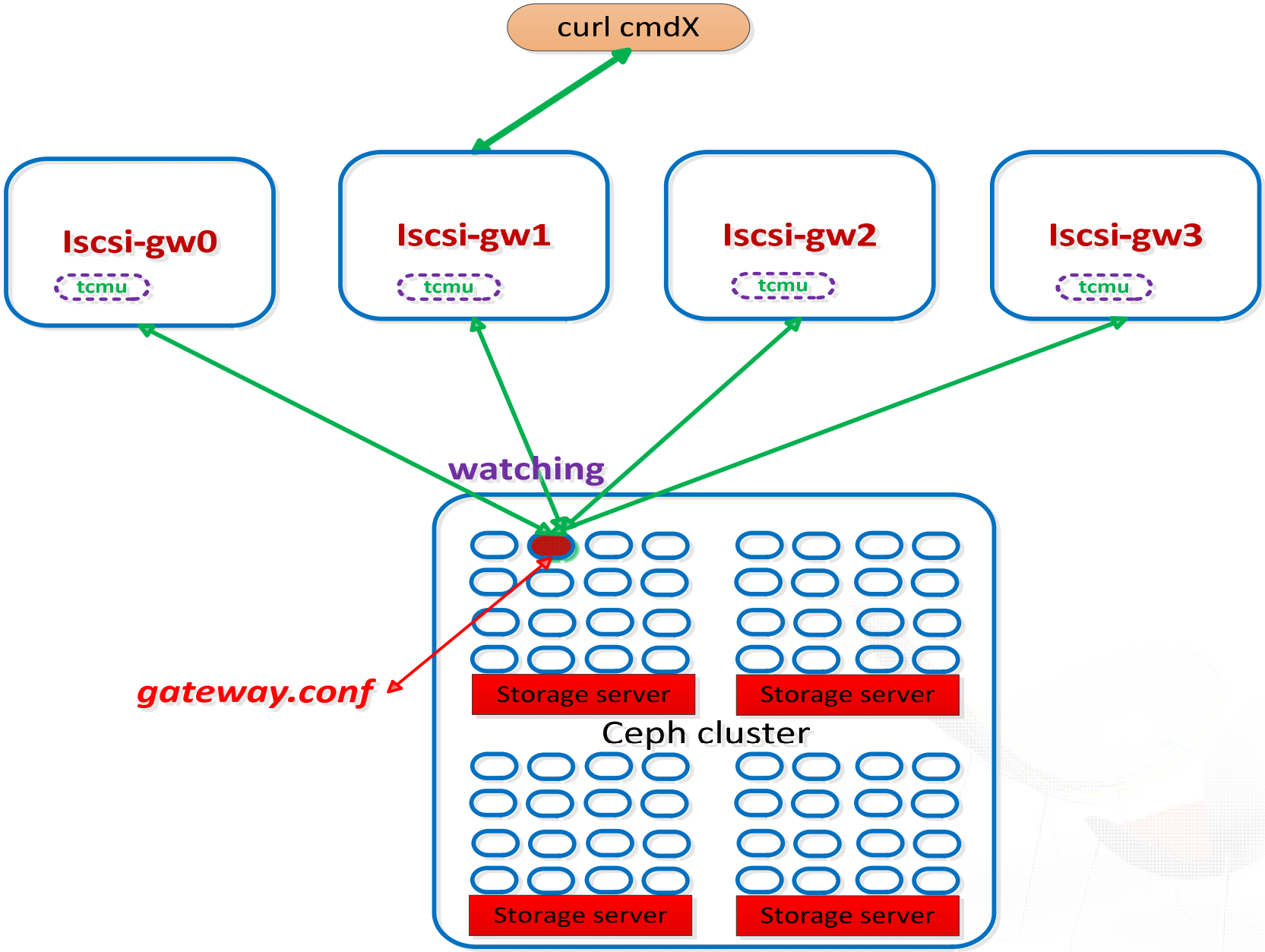
targetcli

```
[root@wzy-3 data]# targetcli ls
cmd: ls
o- / ..... [...]
  o- backstores ..... [...]
  | o- block ..... [Storage Objects: 0]
  | o- fileio ..... [Storage Objects: 0]
  | o- pscsi ..... [Storage Objects: 0]
  | o- ramdisk ..... [Storage Objects: 0]
  | o- user:glfs ..... [Storage Objects: 0]
  | o- user:qcow ..... [Storage Objects: 0]
  | o- user:rbd ..... [Storage Objects: 1]
  |   o- block0 ..... [rbd/block0 (10.0GiB) activated]
  |     o- alua ..... [ALUA Groups: 2]
  |       o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
  |       o- tpg0 ..... [ALUA state: Active/optimized]
  o- iscsi ..... [Targets: 1]
  | o- iqn.2017-03.org.ceph:10.142.40.222:0 ..... [TPGs: 1]
  |   o- tpg1 ..... [no-gen-acls, no-auth]
  |     o- acls ..... [ACLs: 2]
  |     | o- iqn.2017-03.org.ceph:10.142.40.220 ..... [Mapped LUNs: 1]
  |     | | o- mapped_lun0 ..... [lun0 user/block0 (rw)]
  |     | o- iqn.2017-03.org.ceph:10.142.40.221 ..... [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ..... [lun0 user/block0 (rw)]
  |     o- luns ..... [LUNs: 1]
  |     | o- lun0 ..... [user/block0 (tpg0)]
  |     o- portals ..... [Portals: 1]
  |       o- 0.0.0.0:3260 ..... [OK]
  o- loopback ..... [Targets: 0]
  o- vhost ..... [Targets: 0]
  o- xen_pvscsi ..... [Targets: 0]
```


glfs
rbd
fpo
file
file_zbc
libtcmu
tcmu-runner
ceph-iscsi-cli
rtslib
Ceph XXlib
user space
kernel space
uioX
target_core_user
target_core_rd
target_core_configfs
target_core_pscsi
target_core_iblock
target_core_file
target_core_mod
tcm_vhost
tcm_usb_gadget
sbp_target
tcm_qla2xxx
iscsi_target_mod
tcm_fc
iscsi
Ceph cluster
Storage server
Storage server
Storage server
Storage server

ceph iscsi gateway supports

```
/> ls
o- / ..................................................................... [...]
  o- clusters ..................................................... [Clusters: 1]
  | o- ceph ....................................................... [HEALTH_OK]
  |   o- pools ..................................................... [Pools: 3]
  |   | o- ec ..................................... [(2+1), Commit: 0b/40G (0%), Used: 0b]
  |   | o- iscsi .................................. [(x3), Commit: 0b/20G (0%), Used: 18b]
  |   | o- rbd .................................... [(x3), Commit: 8G/20G (40%), Used: 5K]
  |   o- topology ............................................... [OSDs: 3,MONs: 3]
  o- disks ....................................................... [8G, Disks: 5]
  | o- rbd.disk_1 ................................................ [disk_1 (1G)]
  | o- rbd.disk_2 ................................................ [disk_2 (2G)]
  | o- rbd.disk_3 ................................................ [disk_3 (2G)]
  | o- rbd.disk_4 ................................................ [disk_4 (1G)]
  | o- rbd.disk_5 ................................................ [disk_5 (2G)]
  o- iscsi-target ................................................. [Targets: 1]
    o- iqn.2003-01.com.redhat.iscsi-gw:ceph-gw ..................... [Gateways: 2]
      o- gateways ....................................... [Up: 2/2, Portals: 2]
      | o- rh7-gw1 ..................................... [192.168.122.69 (UP)]
      | o- rh7-gw2 ..................................... [192.168.122.104 (UP)]
      o- hosts ..................................................... [Hosts: 2]
        o- iqn.1994-05.com.redhat:myhost1 ....................... [Auth: None, Disks: 1(1G)]
        | o- lun 0 ..................................... [rbd.disk_1(1G), Owner: rh7-gw2]
        o- iqn.1994-05.com.redhat:rh7-client .......... [LOGGED-IN, Auth: CHAP, Disks: 1(2G)]
          o- lun 0 ..................................... [rbd.disk_5(2G), Owner: rh7-gw2]
```


curl cmdX
Iscsi-gw0
tcmu
Iscsi-gw1
tcmu
Iscsi-gw2
tcmu
Iscsi-gw3
tcmu
watching
gateway.conf
Storage server
Storage server
Ceph cluster
Storage server
Storage server


intel
IBM
IT大咖说
知识分享平台
中国移动
China Mobile
THANKS