

Communication via Internal Shared Memory (ISM) – Time to open up

DIBS - Direct Internal Buffer Sharing -
Proposal for a device class

Netdev 0x19
Zagreb, 2025

Alexandra Winter
wintera@linux.ibm.com

TRADEMARKS

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries

IBM® and the IBM logo

z/Architecture®, IBM Z®

z/VM®, z/OS®, z/TPF®, z/VSE®

A current list trademarks of IBM is available at

<https://www.ibm.com/legal/copyright-trademark>

The following are trademarks or registered trademarks of other companies or persons

Linux® is a registered trademark of Linus Torvalds in the United States and/or other countries

PCIe® is a registered trademark of the PCI-SIG in the United States and/or other countries

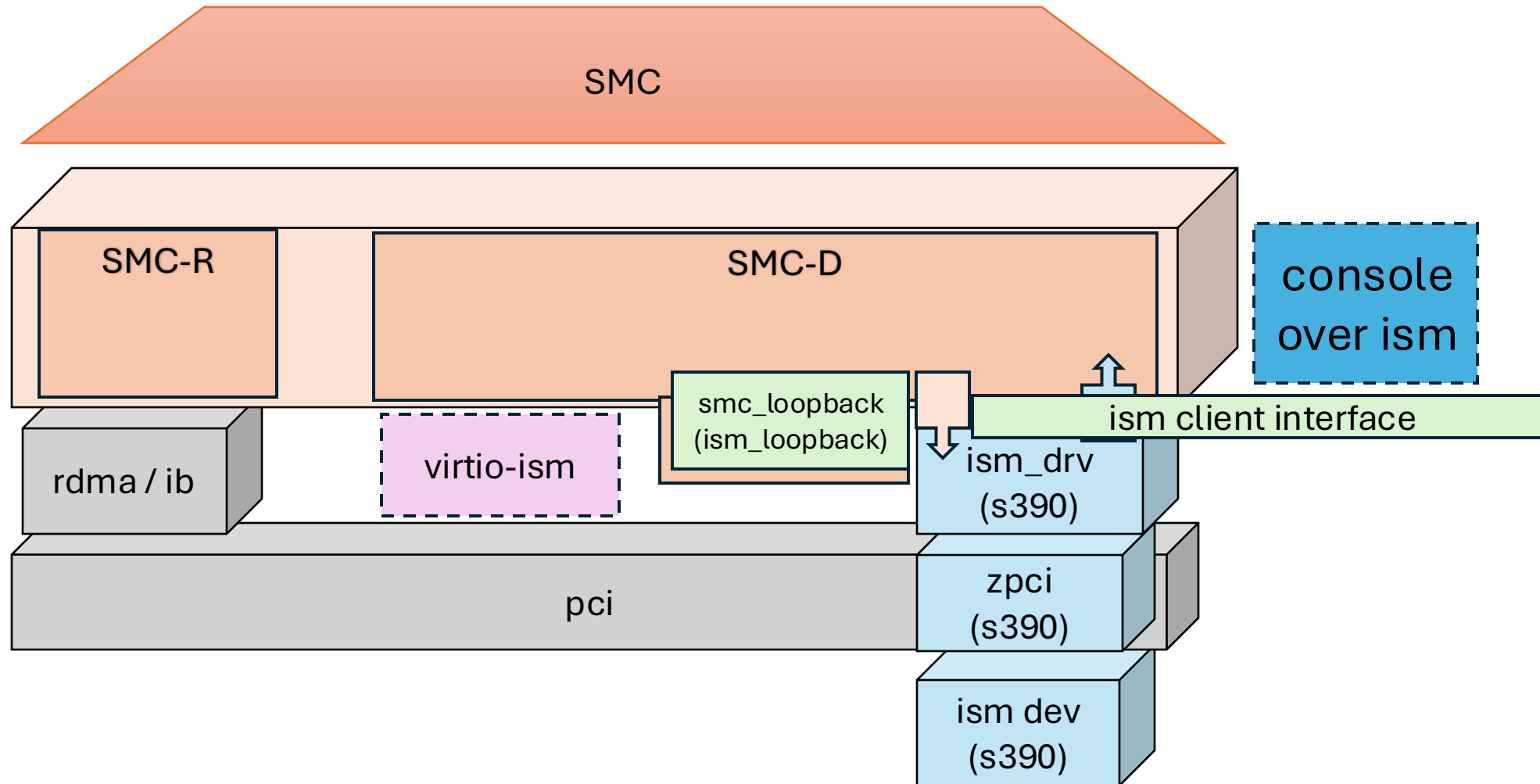
Other company, product, or service names may be trademarks or services marks of others.

AGENDA

- I. Motivation
- II. General concept of a dibs device
- III. Details
- IV. Next steps

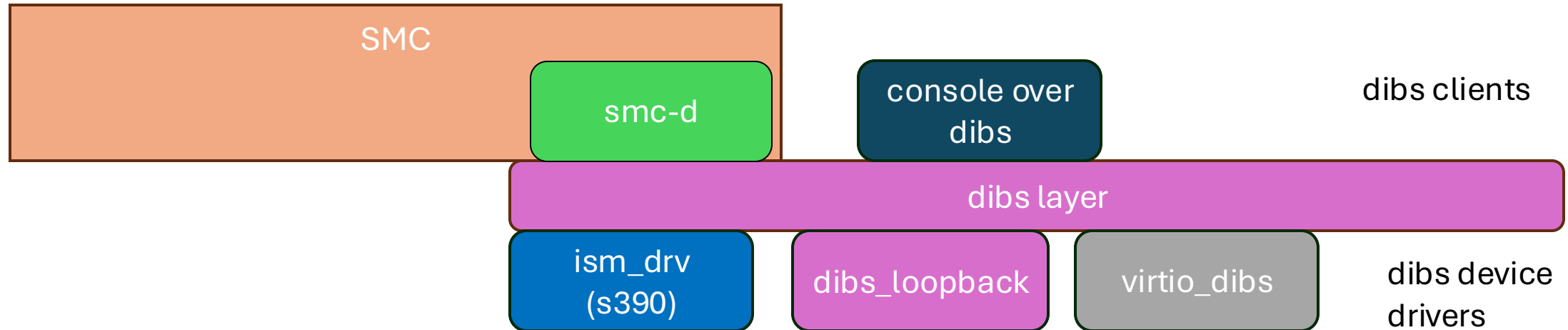
I. MOTIVATION

Status today (historically grown)



Goal: DIBS layer – Direct Internal Buffer Sharing

- Note: dibs is a working title, most suitable name is still under discussion



ism: Firmware fabric between s390 logical partitions

dibs_loopback: for testing of dibs clients or between containers

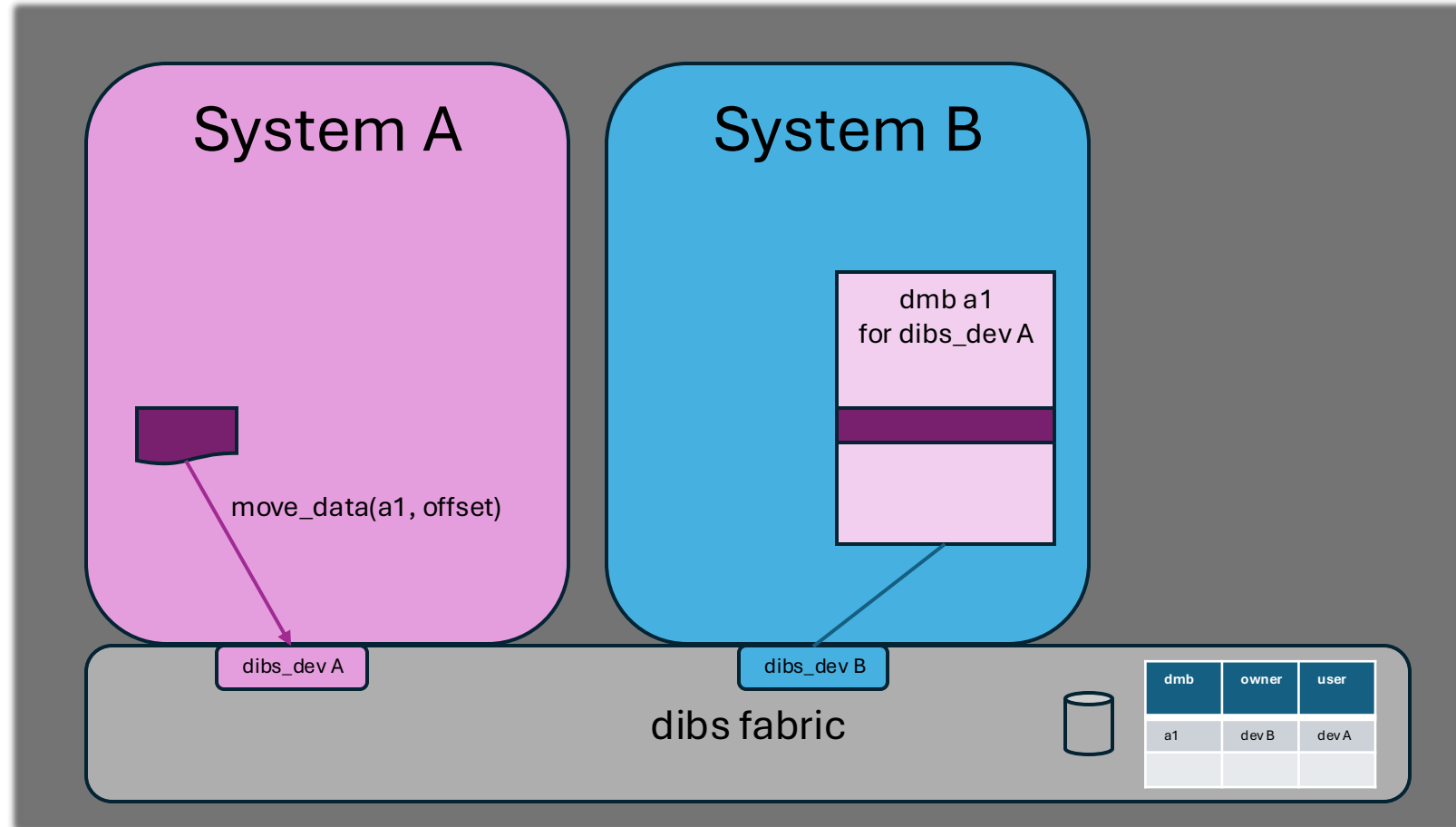
virtio_dibs: Fabric between KVM guests in the same KVM host system

Motivation of dibs layer

- NOT: invent a brandnew device type from scratch.
- BUT: Streamline existing devices and clients and ongoing work
 - define and describe internal APIs
 - remove duplicate code
 - remove dependencies
- If we do a good job, it could be a foundation for even more usecases or expansions.

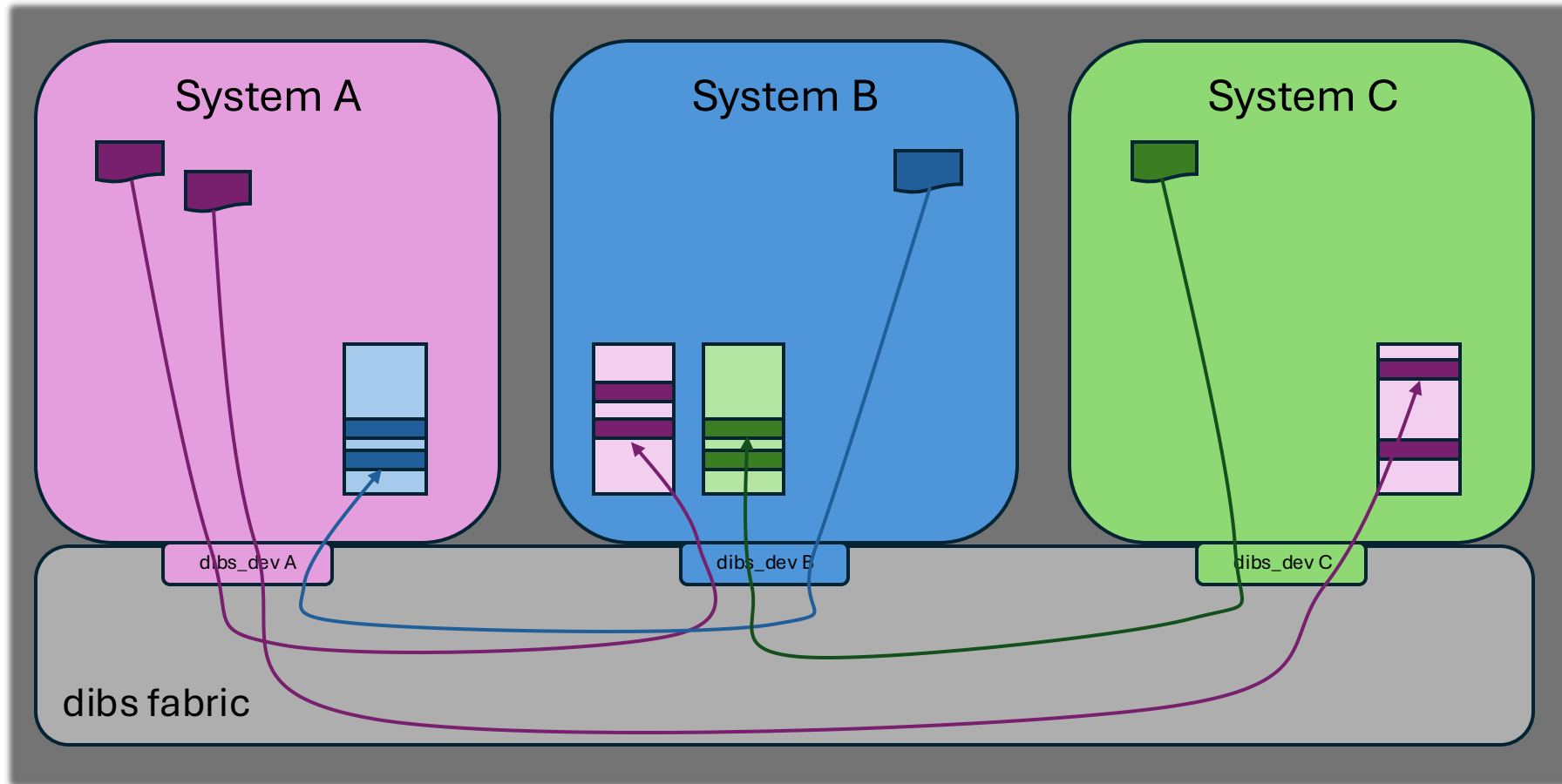
II. GENERAL CONCEPT

dibs fabric, dibs devices, dmb buffers



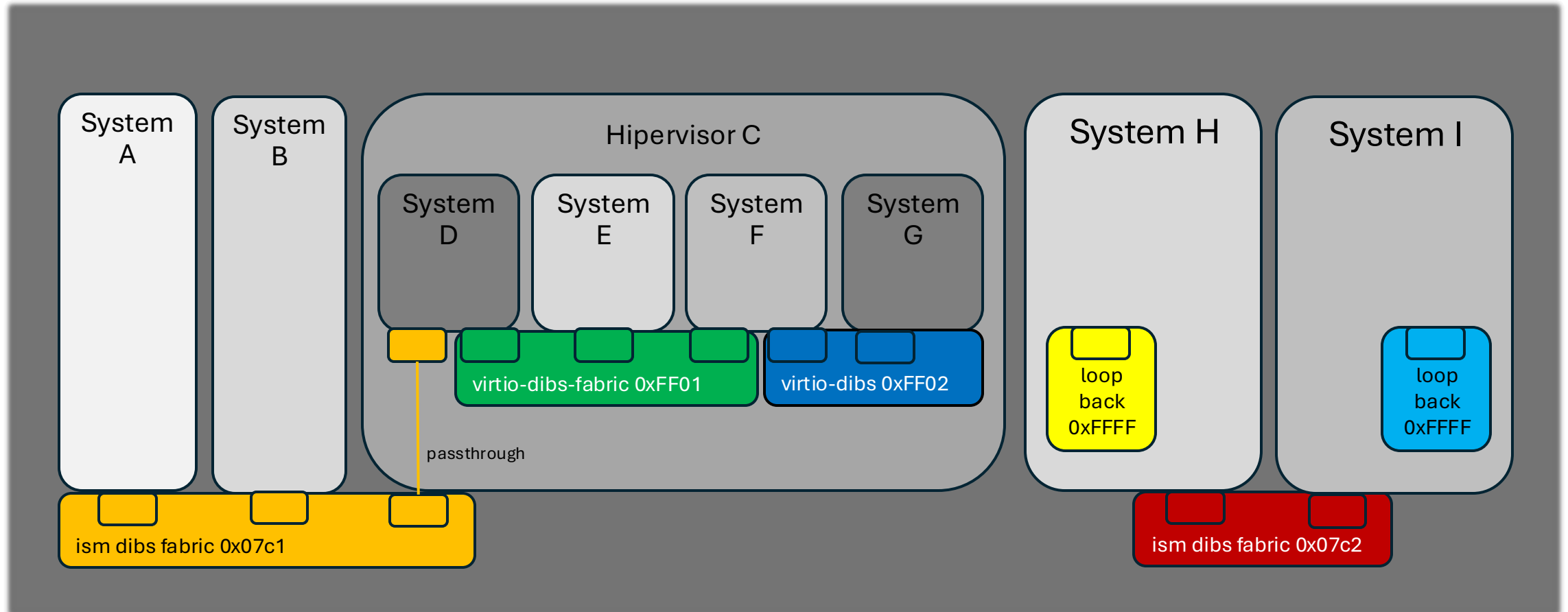
- dibs – direct internal buffer sharing
- dmb – direct memory buffer

many-to-many, but access controlled



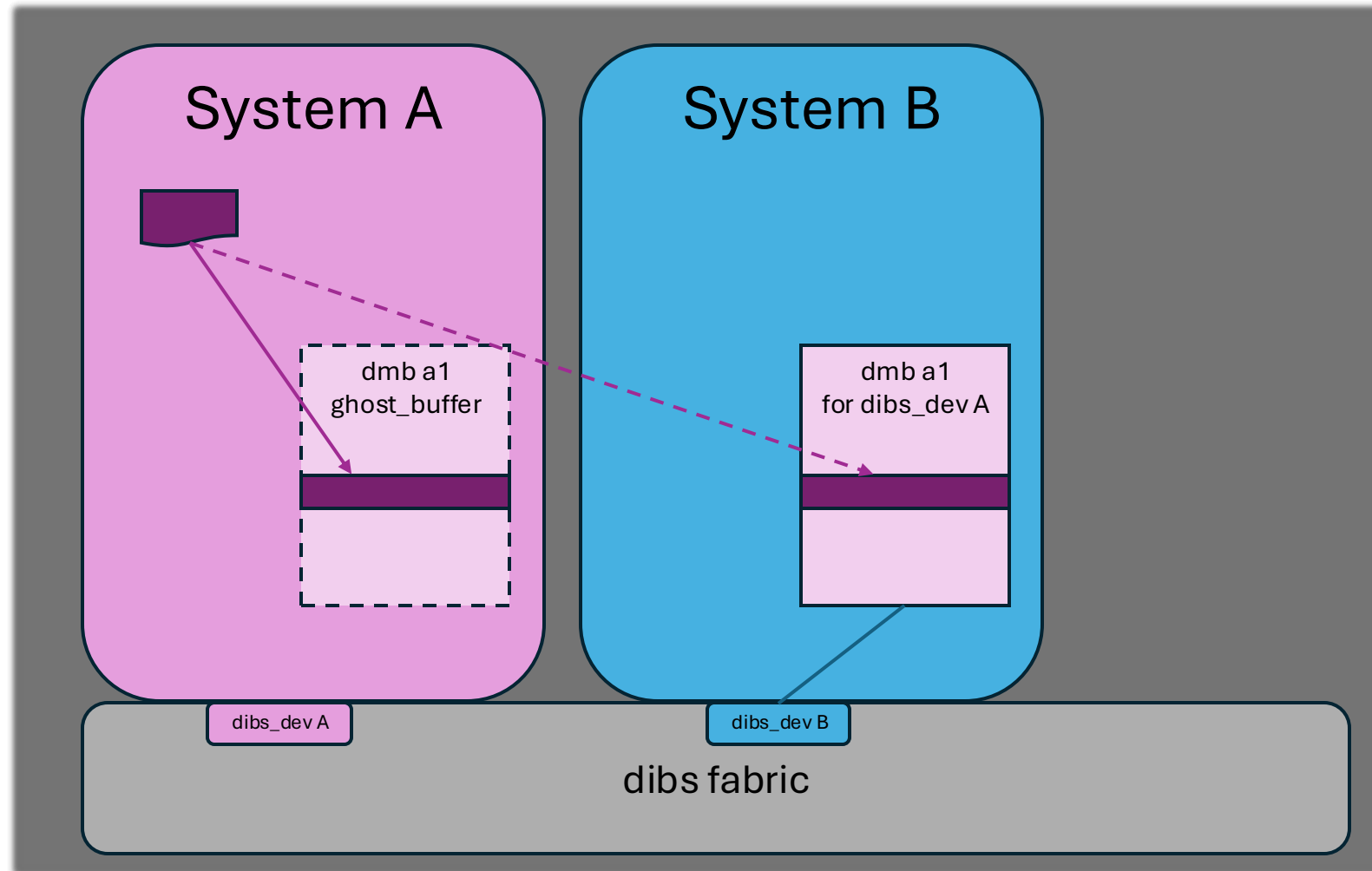
- Owning system decides which remote device is authorized to access a dmb; dibs fabric enforces this decision
- Owner provides memory for dmbs

multiple and different fabrics



- Only dibs devices of same fabric can communicate.
- All dibs loopback devices have fabric id 0xFFFF, but it cannot communicate with each other.

Mapped dmbs (optional)



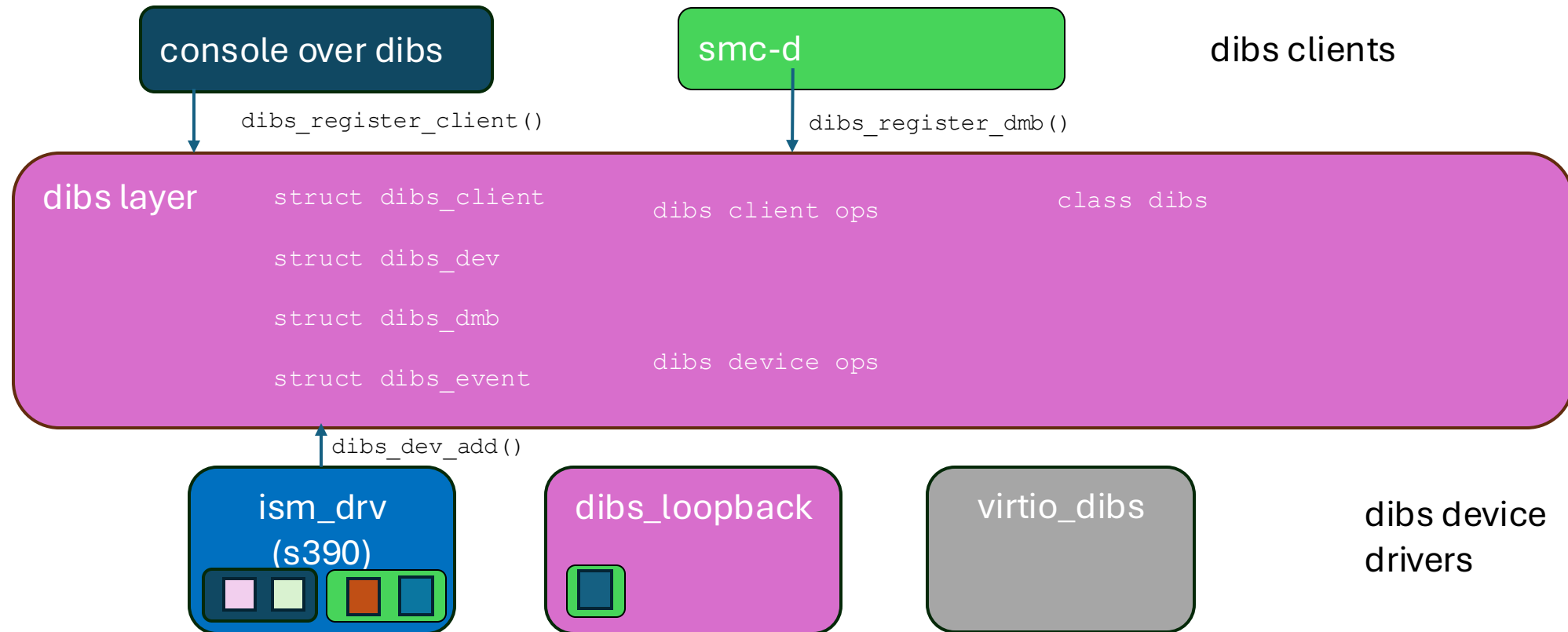
DIBS – Direct Internal Buffer Sharing

- Based on devices and fabric
- DMB: Direct Memory Buffers, dedicated to one external device
- Write directly at any offset into a remote dmb
- Access control by fabric

III. DETAILS

Disclaimer: This is work in progress, the final implementation may look different.

dibs layer



- Every client can access all dibs devices
- dmbs are owned by a client, but allocated by device drivers

class structure

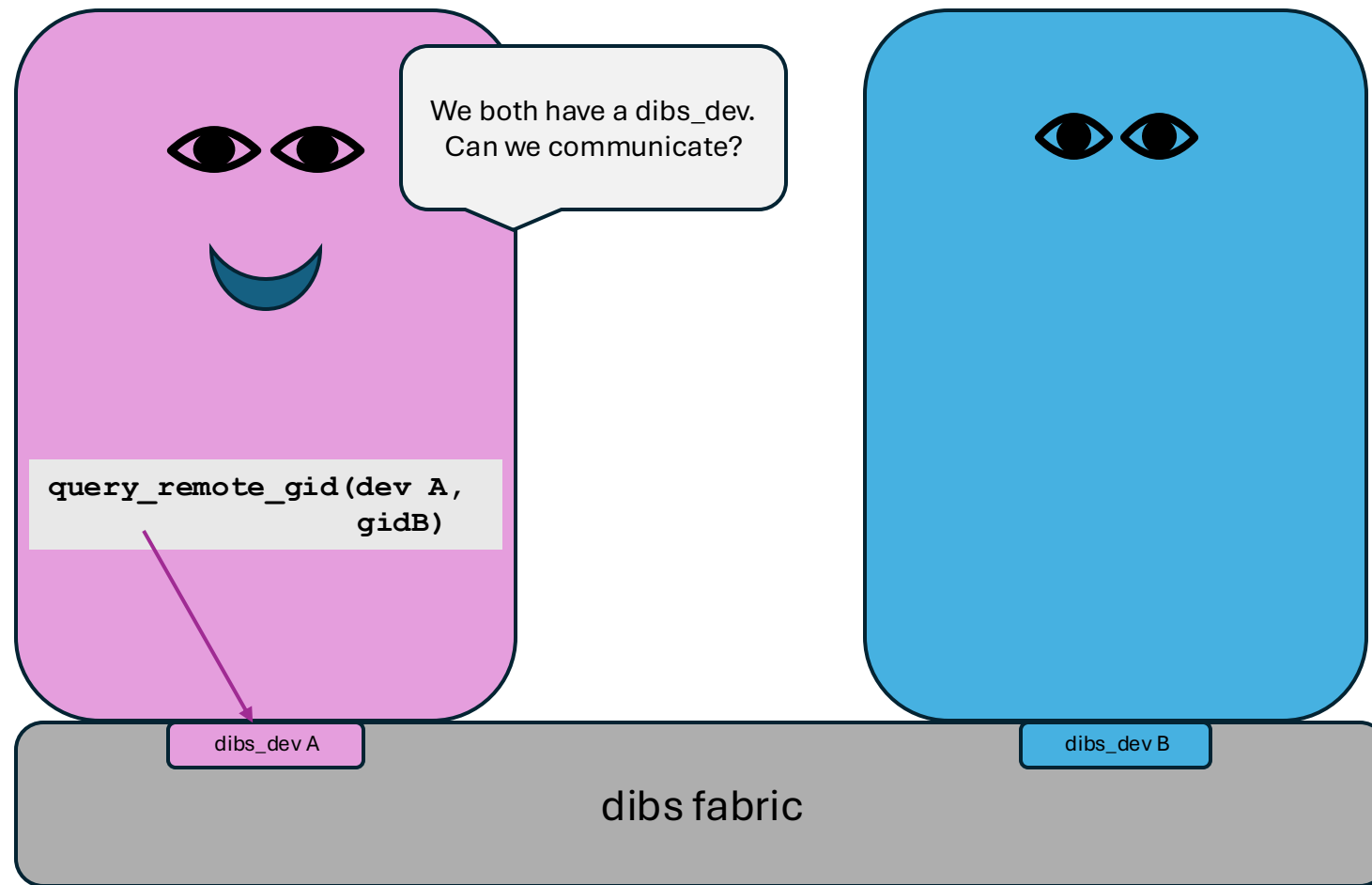
```
> ls /sys/class/dibs/
ism0@  lo@
>
> ls -l /sys/class/dibs/*/
/sys/class/dibs/ism0/:
total 0
lrwxrwxrwx. 1 root root    0 Mar  4 08:52 device -> ../../../../0000:00:00.0/
-r--r--r--. 1 root root 4.0K Mar  4 08:52 fabric_id
-r--r--r--. 1 root root 4.0K Mar  4 08:52 gid
lrwxrwxrwx. 1 root root    0 Mar  4 08:52 subsystem -> ../../../../class/dibs/
-rw-r--r--. 1 root root 4.0K Mar  4 08:52 uevent

/sys/class/dibs/lo/:
total 0
-r--r--r--. 1 root root 4.0K Mar  4 08:52 fabric_id
-r--r--r--. 1 root root 4.0K Mar  4 08:52 gid
lrwxrwxrwx. 1 root root    0 Mar  4 08:52 subsystem -> ../../../../class/dibs/
-rw-r--r--. 1 root root 4.0K Mar  4 08:52 uevent
>
> grep "" /sys/class/dibs/*/gid
/sys/class/dibs/ism0/gid:4a00fab-66a8-8561-0000-000000000000
/sys/class/dibs/lo/gid:eb2d8caf-16e2-4a15-96c9-6a9f3c85dc9e
>
> grep "" /sys/class/dibs/*/fabric_id
/sys/class/dibs/ism0/fabric_id:0x07c1
/sys/class/dibs/lo/fabric_id:0xffff
```

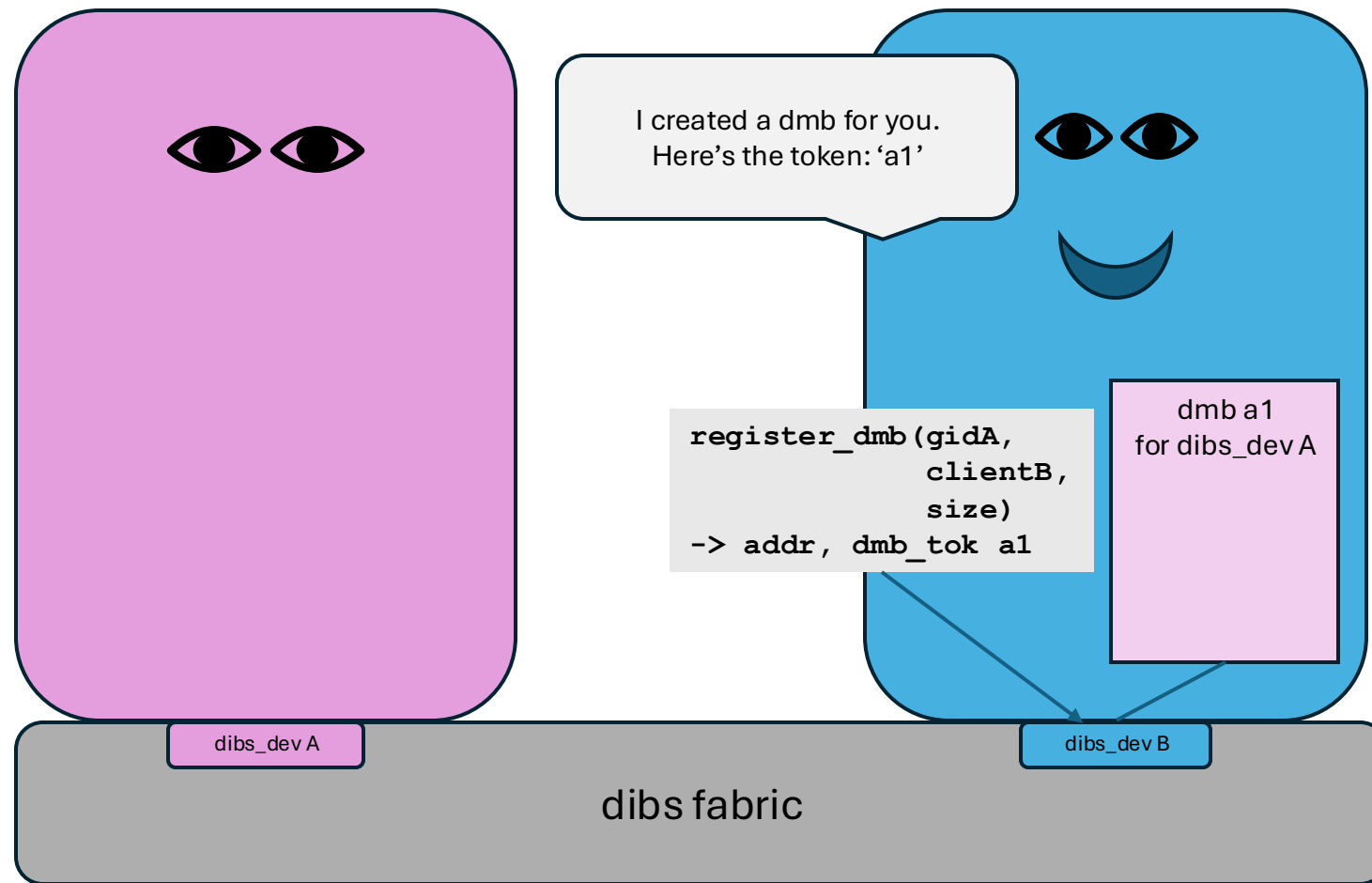
Tokens

- dibs fabric id:
 - 16 Bit
 - unique within the same hardware
(with the exception of the dibs loopback fabric: 1 device – 1 fabric; fabric id 0xFFFF)
- dibs device global identifier (gid):
 - 128 Bit uuid_t format
 - guaranteed to be unique within the same fabric
and statistically likely to be globally unique
- dmb token:
 - 64 Bit
 - unique within the same fabric
- Tokens are handed out by fabric
- Token exchange between clients is not part of dibs layer

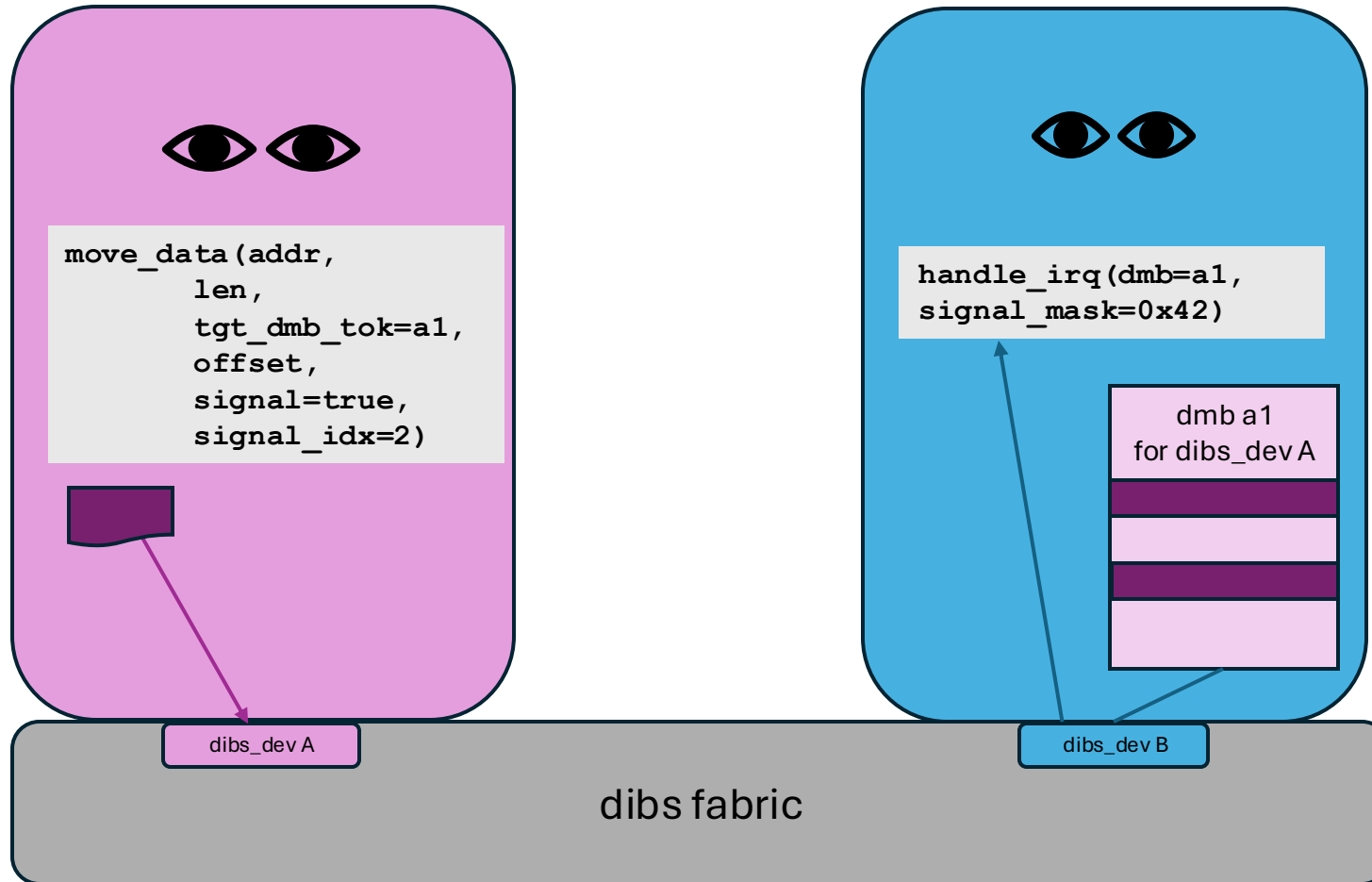
dibs_dev_ops: query_remote_gid



dibs_dev_ops: register_dmb

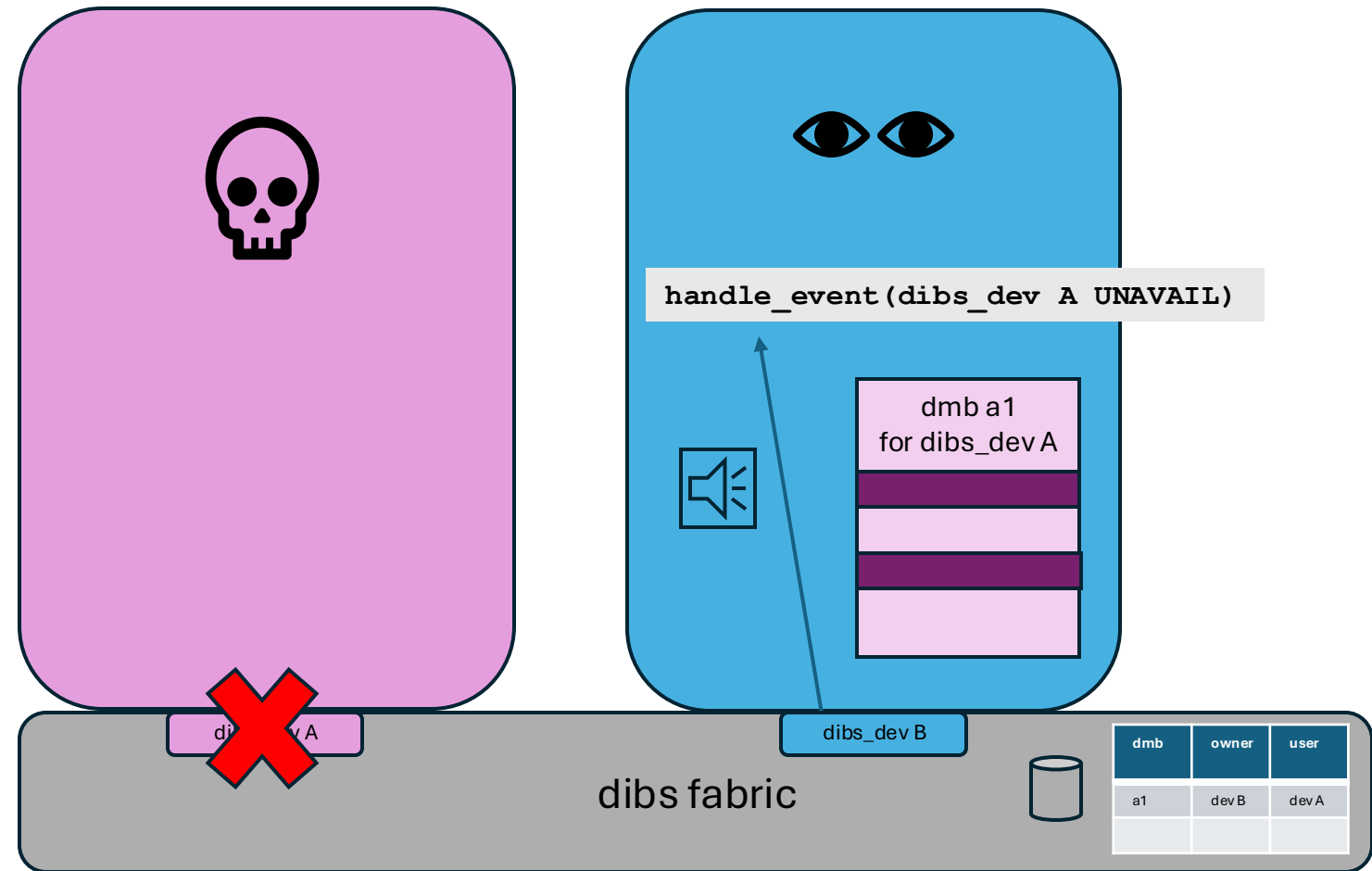


dibs_dev_ops: move_data



- Meaning of signal indices and signal mask is defined by clients.
- Target device can coalesce signal irqs.

dibs events (optional)



Optional feature:

- fabric can generate events
- fabric can provide for client-generated events

IV. Next steps

- Patchset as RFC to netdev mailinglist
 - v1 in Jan/2025:
[RFC net-next 0/7] Provide an ism layer
(thanks for all the comments!)
 - expect v2 within the next weeks
- Follow-ups
 - more simplifications
 - more common helper functions

Future possibilities

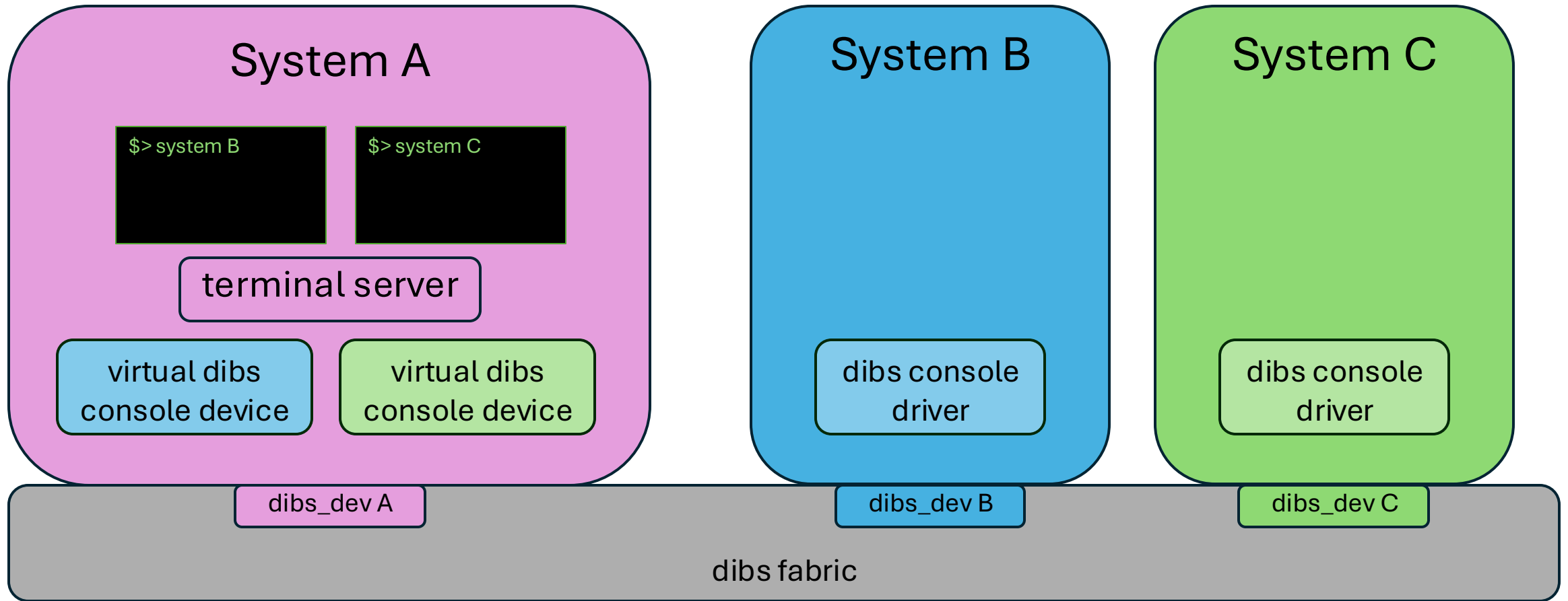
- virtio-dibs devices:

virtio-ism:

[3] <https://groups.oasis-open.org/communities/community-home/digestviewer/viewthread?GroupId=3973&MessageKey=c060ecf9-ea1a-49a2-9827-c92f0e6447b2&CommunityKey=2f26be99-3aa1-48f6-93a5-018dce262226&hlmlt=VT>

- console client over dibs:
see next slide

Future possibility: console over dibs



- network independent
- available during early boot / install

But first let's improve the foundation

