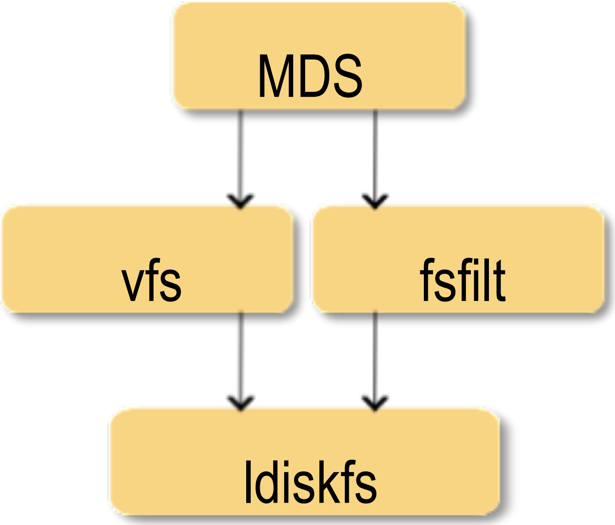
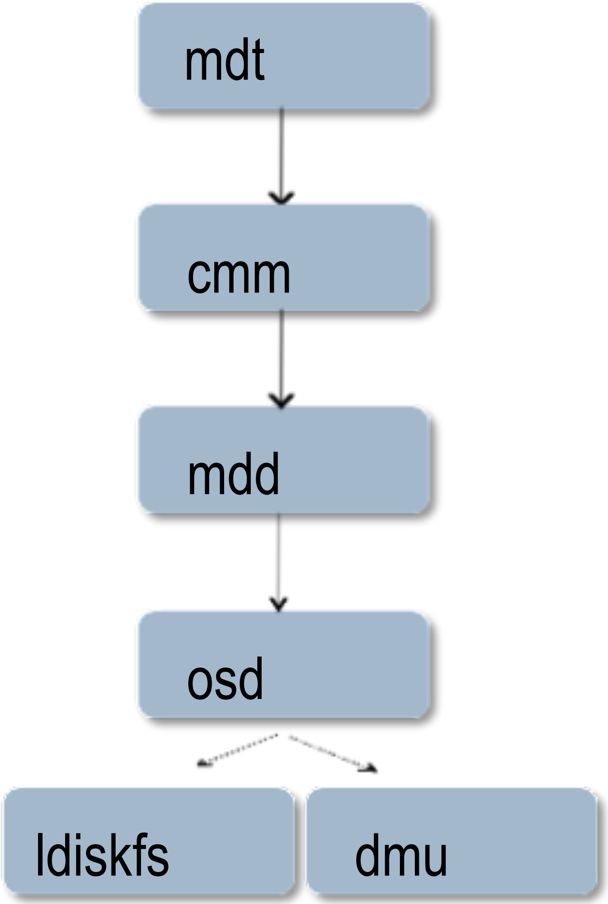
MDT Stack

Nikita Danilov
Staff Engineer
Lustre Group

Old (1.4-1.6) and new (HEAD) MDT





- fids
- md stack rewrite
- Integration of new features
- Don't lose performance

Aside: fids

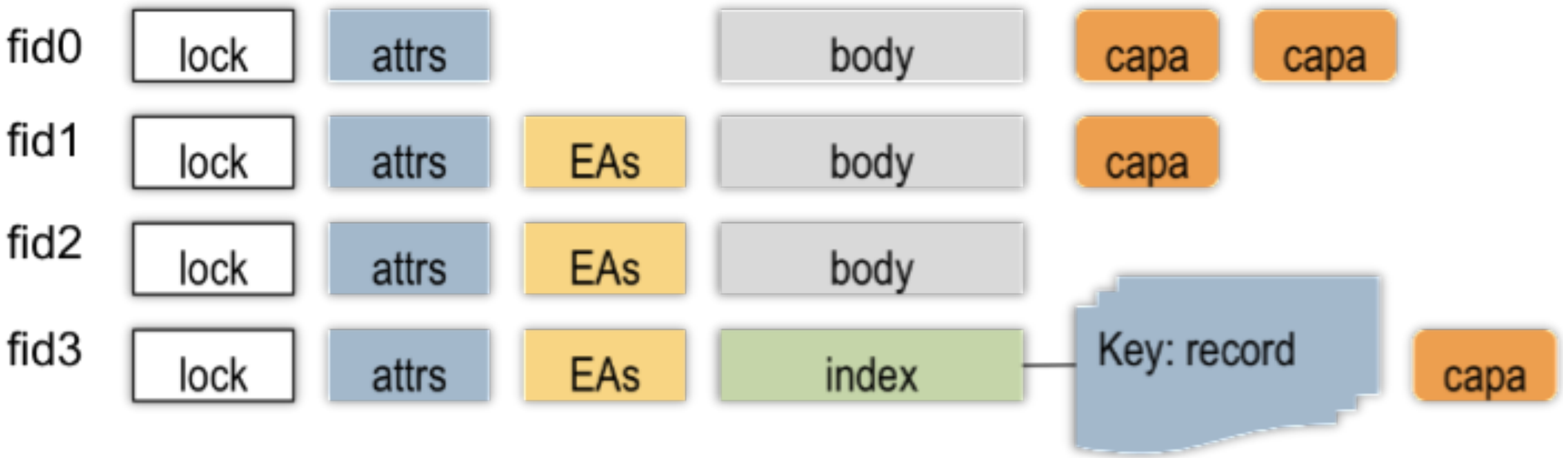
- 1.6: "storage cookie" (MDS ino)
    - Fid: not bound to the specific MDS
    - Fid: is never reused.
    - Fid: **can be generated by client**.

Problems:

- how clients generate fids locally without colliding with each other?
- how to locate a server where fid lives?
- how a server locates an object?

Answer: seq; fld; object index.

osd



fid0 lock attrs body capa capa

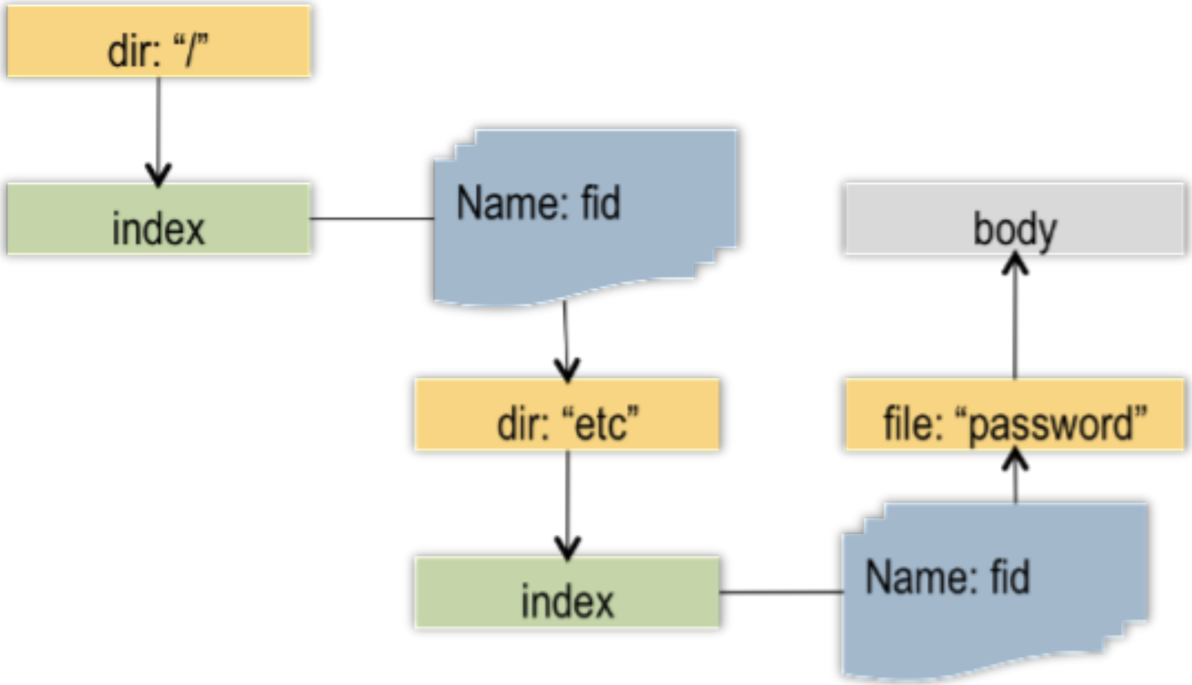
fid1 lock attrs EAs body capa

fid2 lock attrs EAs body

fid3 lock attrs EAs index Key: record capa

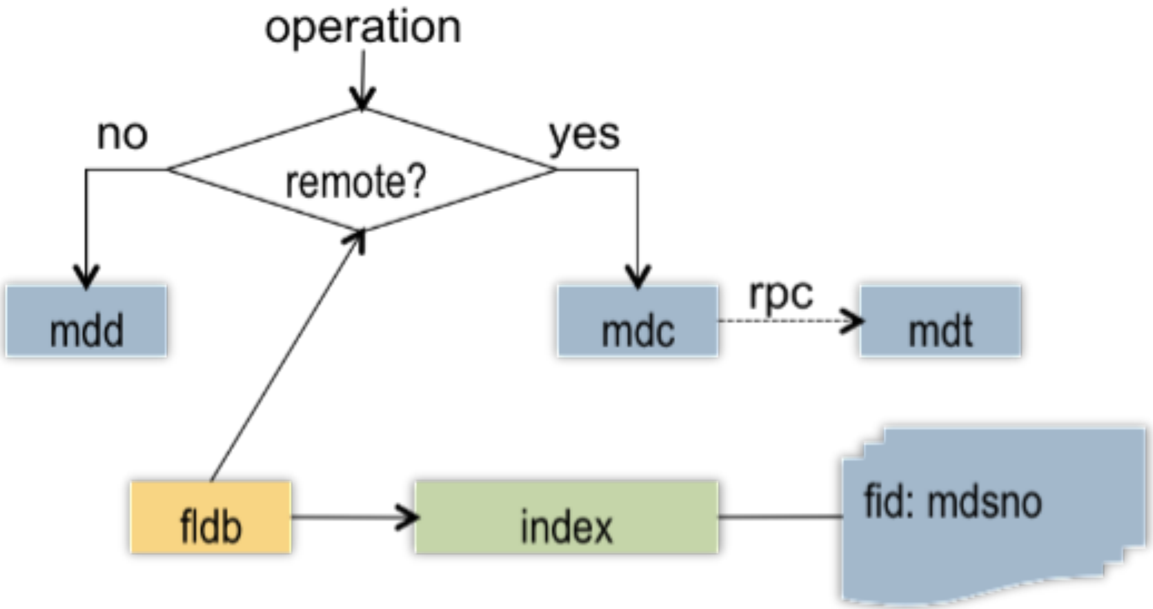
- fid-addressable objects
- Body (read, write, truncate)
- Fixed set of attributes
- Extended attributes
- indices
- Object lock, transactions, capabilities

mdd



- Scalable directories (lookup, (un)link, readdir)
- Permission checks
- acls
- Orphan handling

cmm



- Remote operations
- Partial operations
- Split directories
- fldb data-base

mdt

- Request packing and unpacking
- Replies
- Recovery
- dlm
- intents
- ptlrpc services, threads
- Reintegration

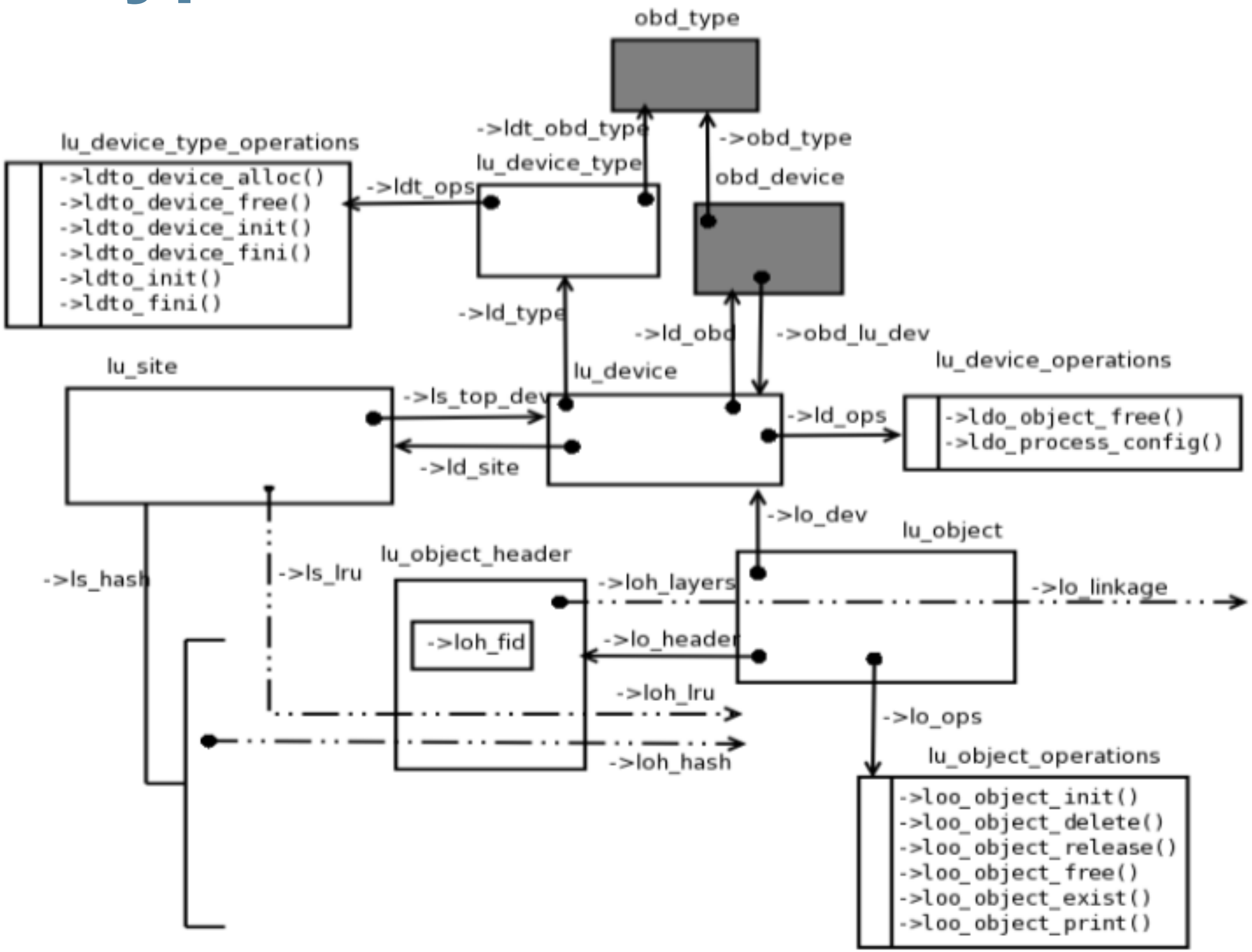
An example: file creation

- client: CREATE(pfid, "foo", cfid) rpc
- mdt:
  - create objects for pfid and cfid
  - take dlm locks
- cmm
  - execute remote object creation (mdc...mdt)
  - insert name locally
- mdd
  - call osd to insert a record (cfid) with a key (“foo”) into index, associated with pfid
- osd
  - insert (key, record) into an index (iam, zap)

Layering

- Layer interfaces:
  - lu: all devices: creation, configuration, destruction
  - dt: osd: fid-addressable objects, indices, transactions
  - md: mdt, cmm, mdd: meta-data operations
- Layer data-types:
  - device: mdt_device, osd_device
  - object: cmr_object, cml_object, osd_object

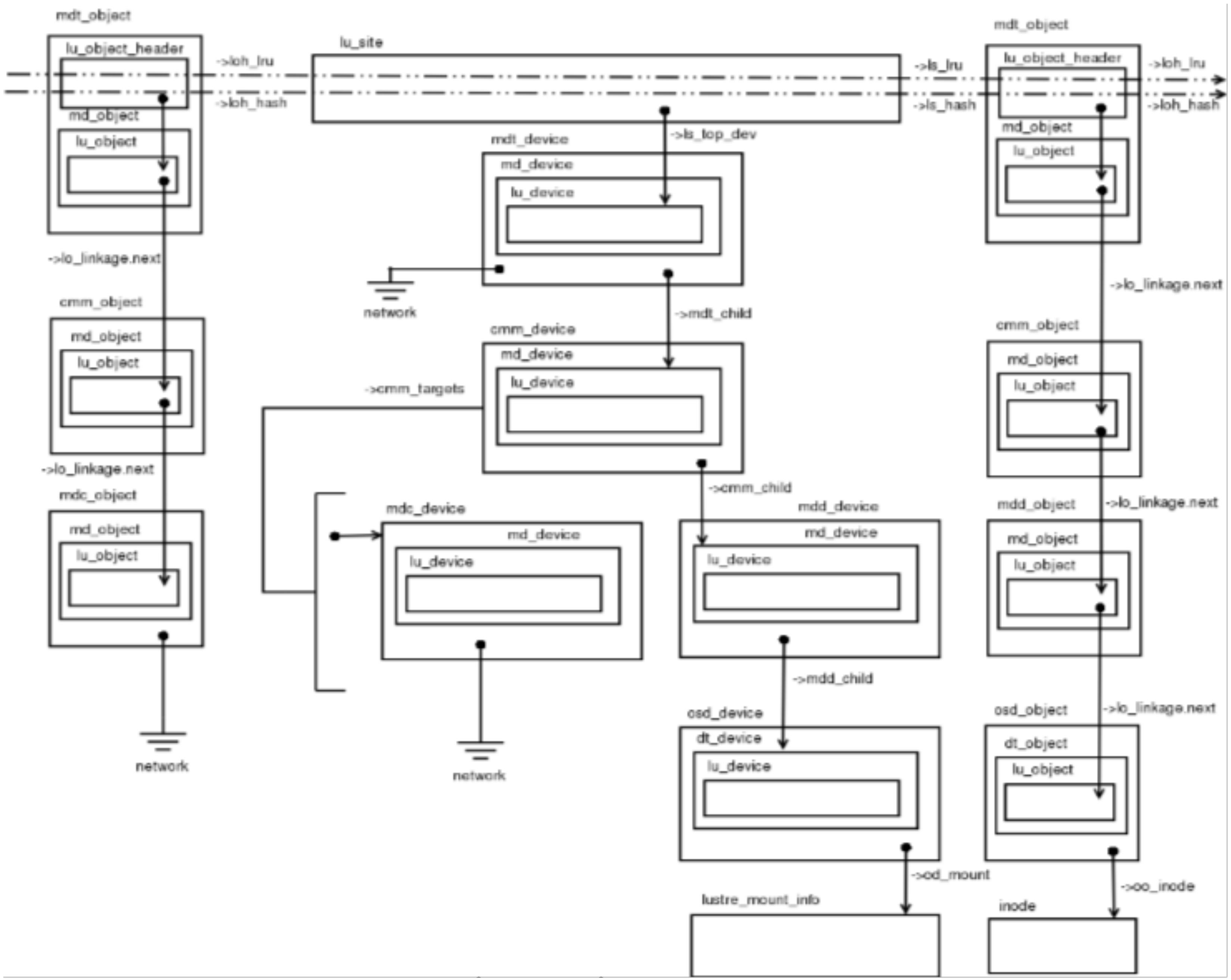
data-types


obd_type
->ldt_obd_type
->obd_type
lu_device_type_operations
->ldto_device_alloc()
->ldto_device_free()
->ldto_device_init()
->ldto_device_fini()
->ldto_init()
->ldto_fini()
->ldt_ops
lu_device_type
obd_device
->ld_type
->ld_obd
->obd_lu_dev
lu_device_operations
lu_site
->ls_top_dev
lu_device
->ld_ops
->ldo_object_free()
->ldo_process_config()
->ld_site
->lo_dev
lu_object
->ls_hash
->ls_lru
lu_object_header
->loh_layers
->lo_linkage
->loh_fid
->lo_header
->loh_lru
->lo_ops
->loh_hash
lu_object_operations
->loo_object_init()
->loo_object_delete()
->loo_object_release()
->loo_object_free()
->loo_object_exist()
->loo_object_print()

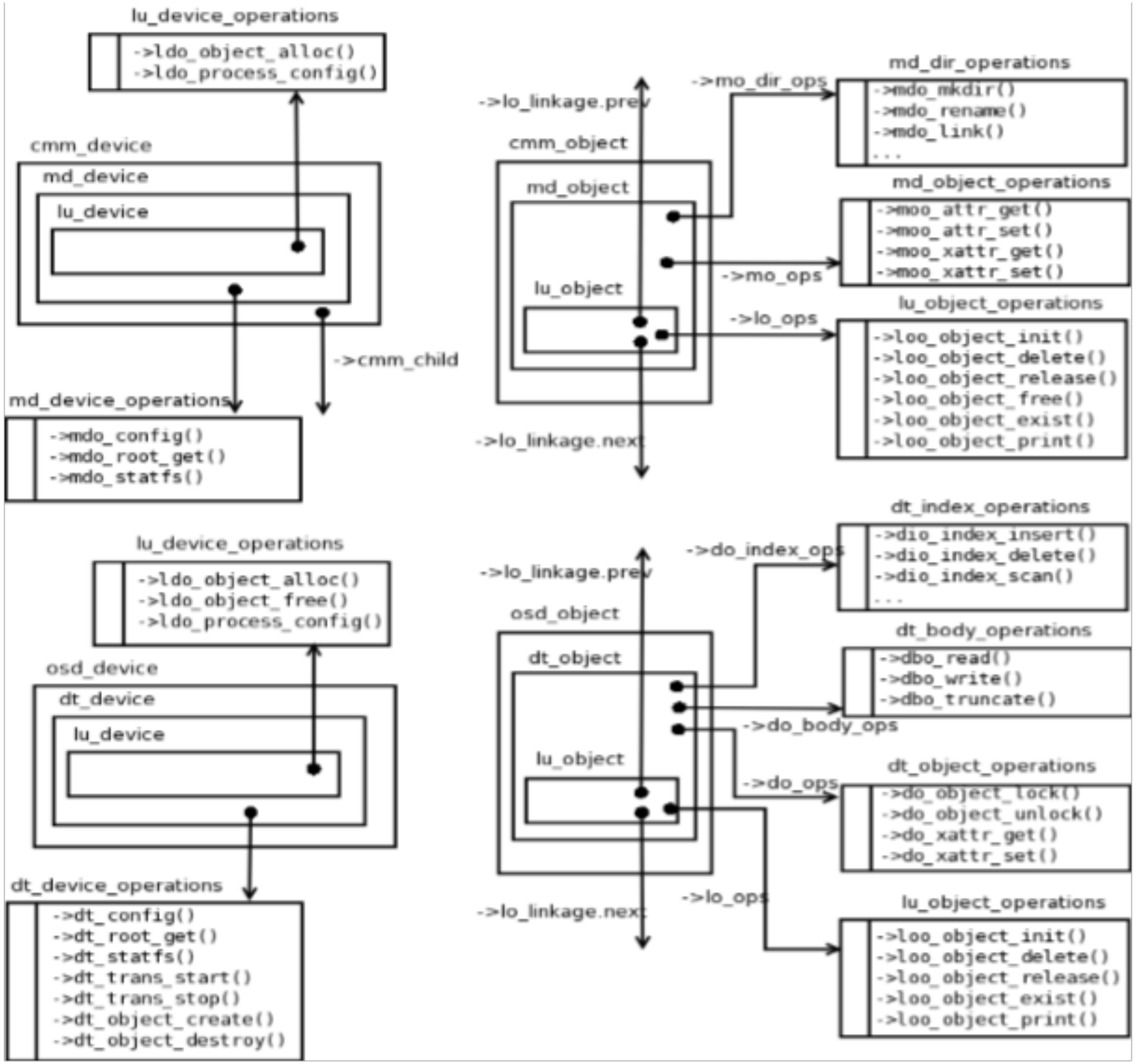
Layered objects

- Unique fid
- State private for a layer
- Operation vectors at every layer
- Generic code
- Caching
- Indexing (hashing)
- LRU, purge

Object Stack



Operation Vectors



lu_device_operations
- ->ldo_object_alloc()
- ->ldo_process_config()

cmm_device
md_device
lu_device

->cmm_child

md_device_operations
- ->mdo_config()
- ->mdo_root_get()
- ->mdo_statfs()

->lo_linkage.prev
cmm_object
md_object
lu_object
->lo_linkage.next

->mo_dir_ops
md_dir_operations
- ->mdo_mkdir()
- ->mdo_rename()
- ->mdo_link()
- ...

->mo_ops
md_object_operations
- ->moo_attr_get()
- ->moo_attr_set()
- ->moo_xattr_get()
- ->moo_xattr_set()

->lo_ops
lu_object_operations
- ->loo_object_init()
- ->loo_object_delete()
- ->loo_object_release()
- ->loo_object_free()
- ->loo_object_exist()
- ->loo_object_print()

lu_device_operations
- ->ldo_object_alloc()
- ->ldo_object_free()
- ->ldo_process_config()

osd_device
dt_device
lu_device

dt_device_operations
- ->dt_config()
- ->dt_root_get()
- ->dt_statfs()
- ->dt_trans_start()
- ->dt_trans_stop()
- ->dt_object_create()
- ->dt_object_destroy()

->lo_linkage.prev
osd_object
dt_object
lu_object
->lo_linkage.next

->do_index_ops
dt_index_operations
- ->dio_index_insert()
- ->dio_index_delete()
- ->dio_index_scan()
- ...

->do_body_ops
dt_body_operations
- ->dbo_read()
- ->dbo_write()
- ->dbo_truncate()

->do_ops
dt_object_operations
- ->do_object_lock()
- ->do_object_unlock()
- ->do_xattr_get()
- ->do_xattr_set()

->lo_ops
lu_object_operations
- ->loo_object_init()
- ->loo_object_delete()
- ->loo_object_release()
- ->loo_object_exist()
- ->loo_object_print()

Nikita.Danilov@sun.com