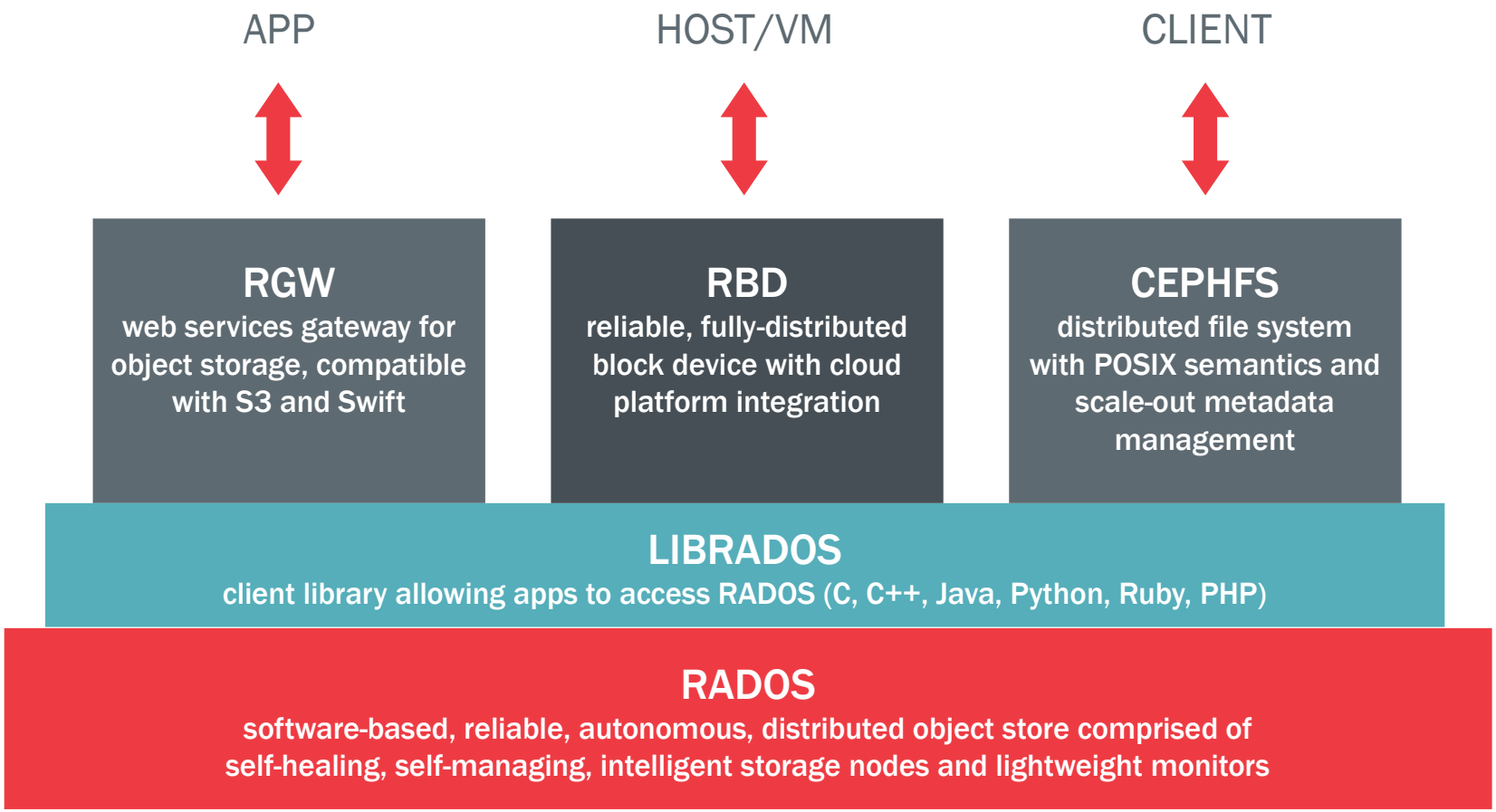
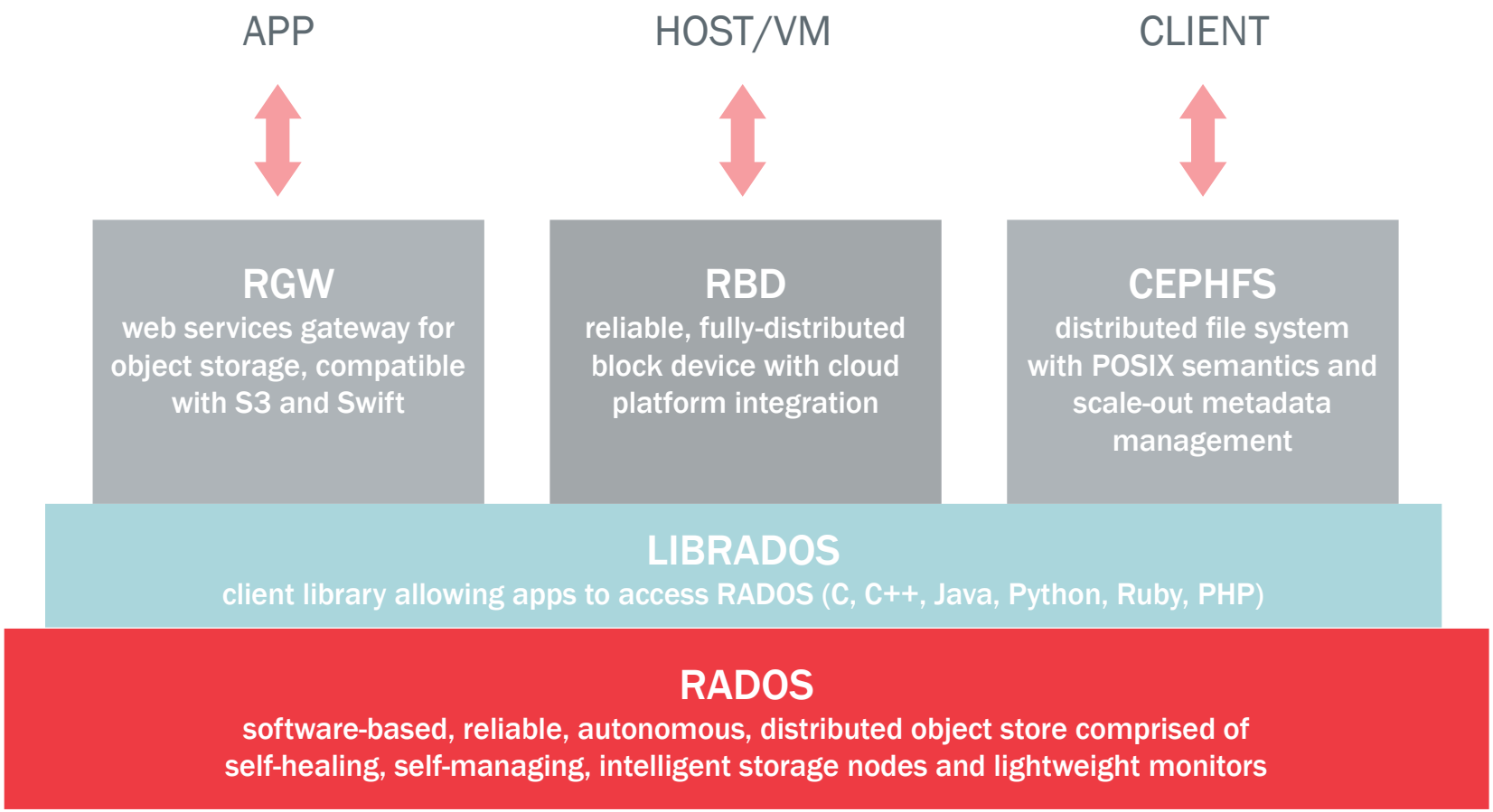
Ceph Internals & Data Processing Capabilities

Joao Eduardo Luis
Senior Software Engineer
jluis@suse.com / joao@suse.de

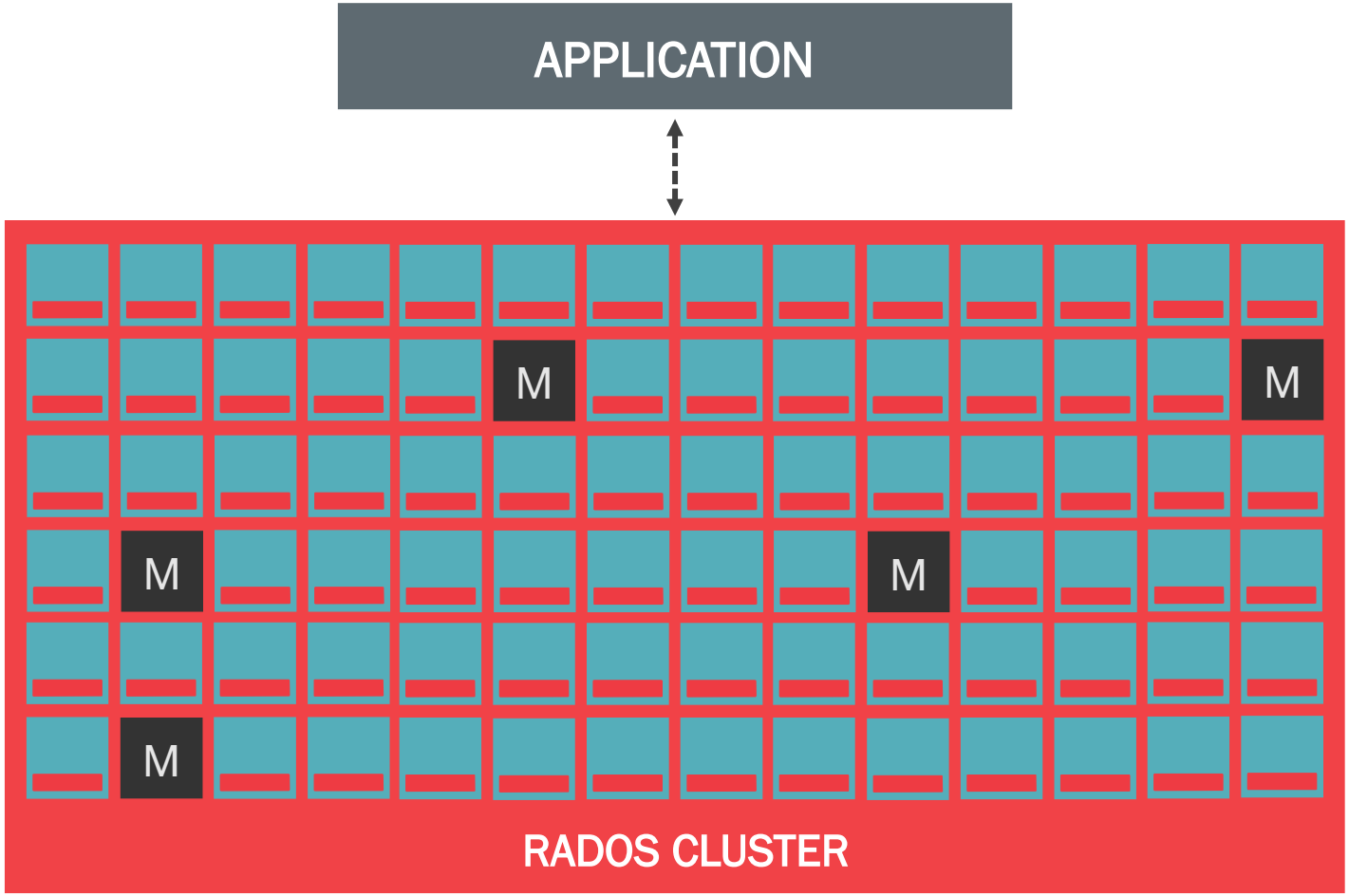
OVERALL ARCHITECTURE



OVERALL ARCHITECTURE



RADOS CLUSTER



RADOS COMPONENTS

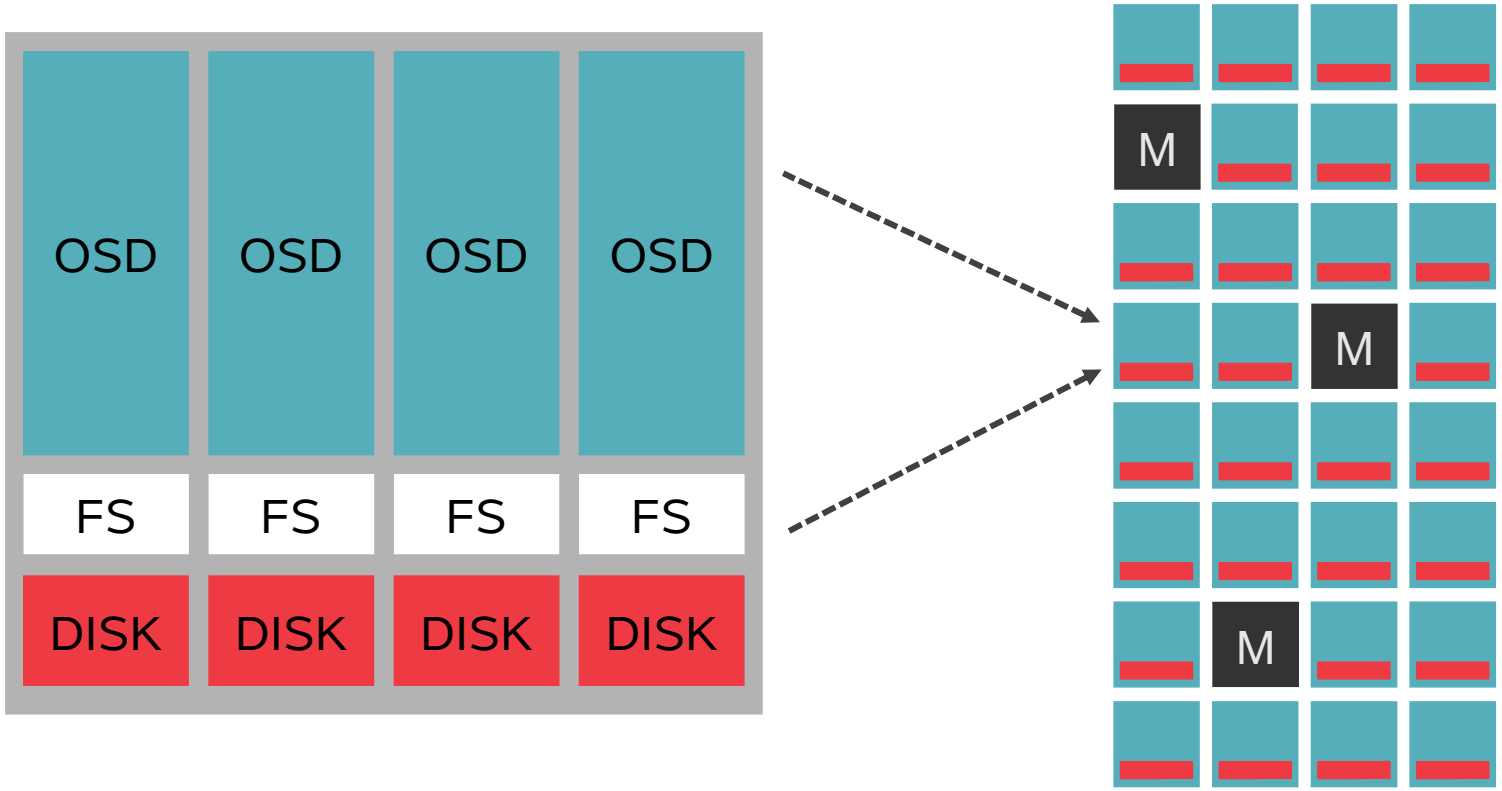
OSDs

- Smart storage
- Resilient, Distributed, Self-healing, etc
- 100's to thousands

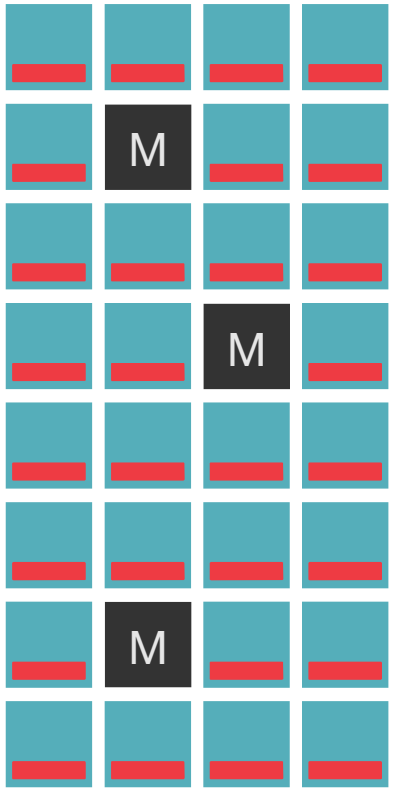
Monitors

- Keep track of cluster state
- Always consistent, or otherwise...
- 3, 5, 7, ...

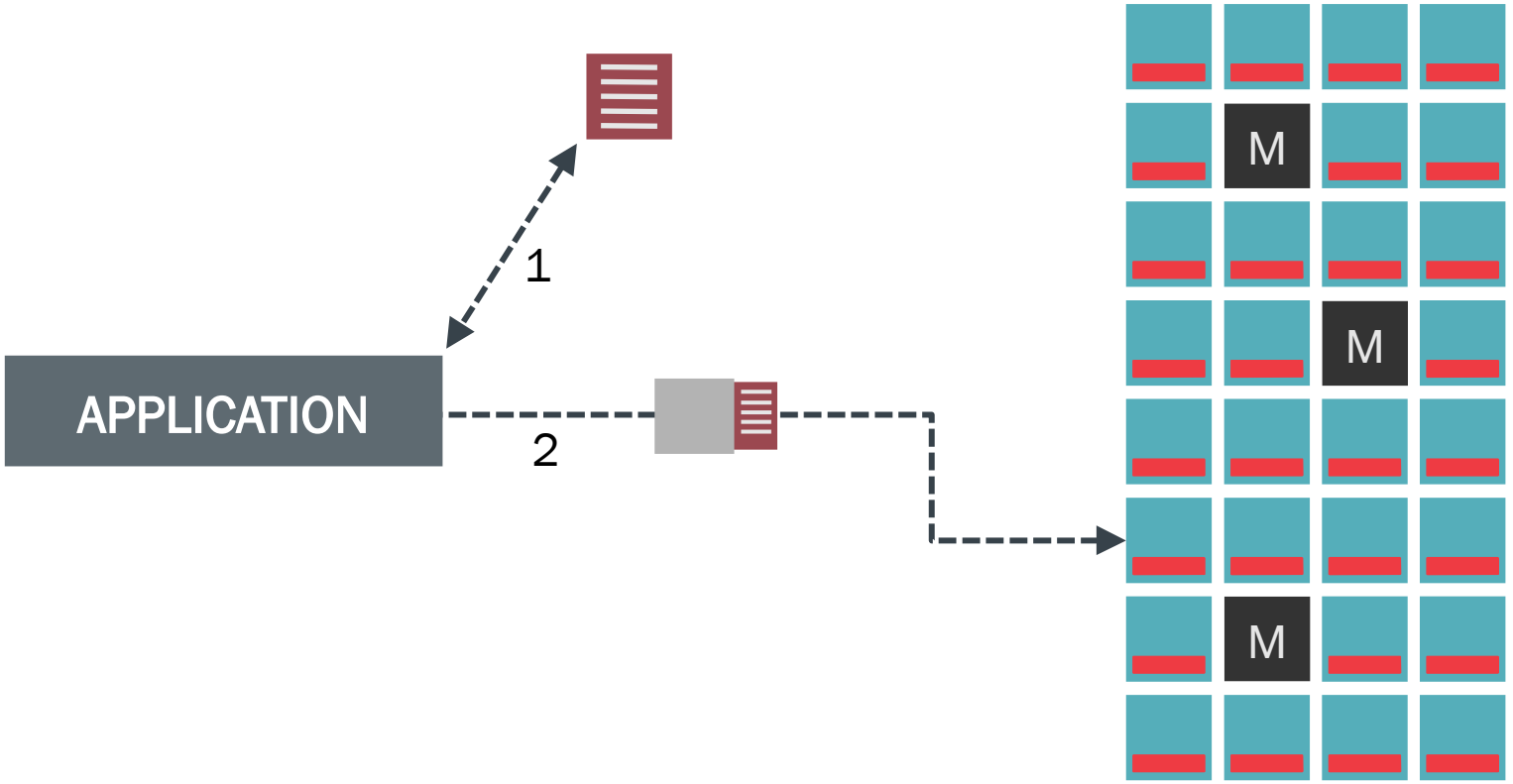
OBJECT STORAGE DAEMONS



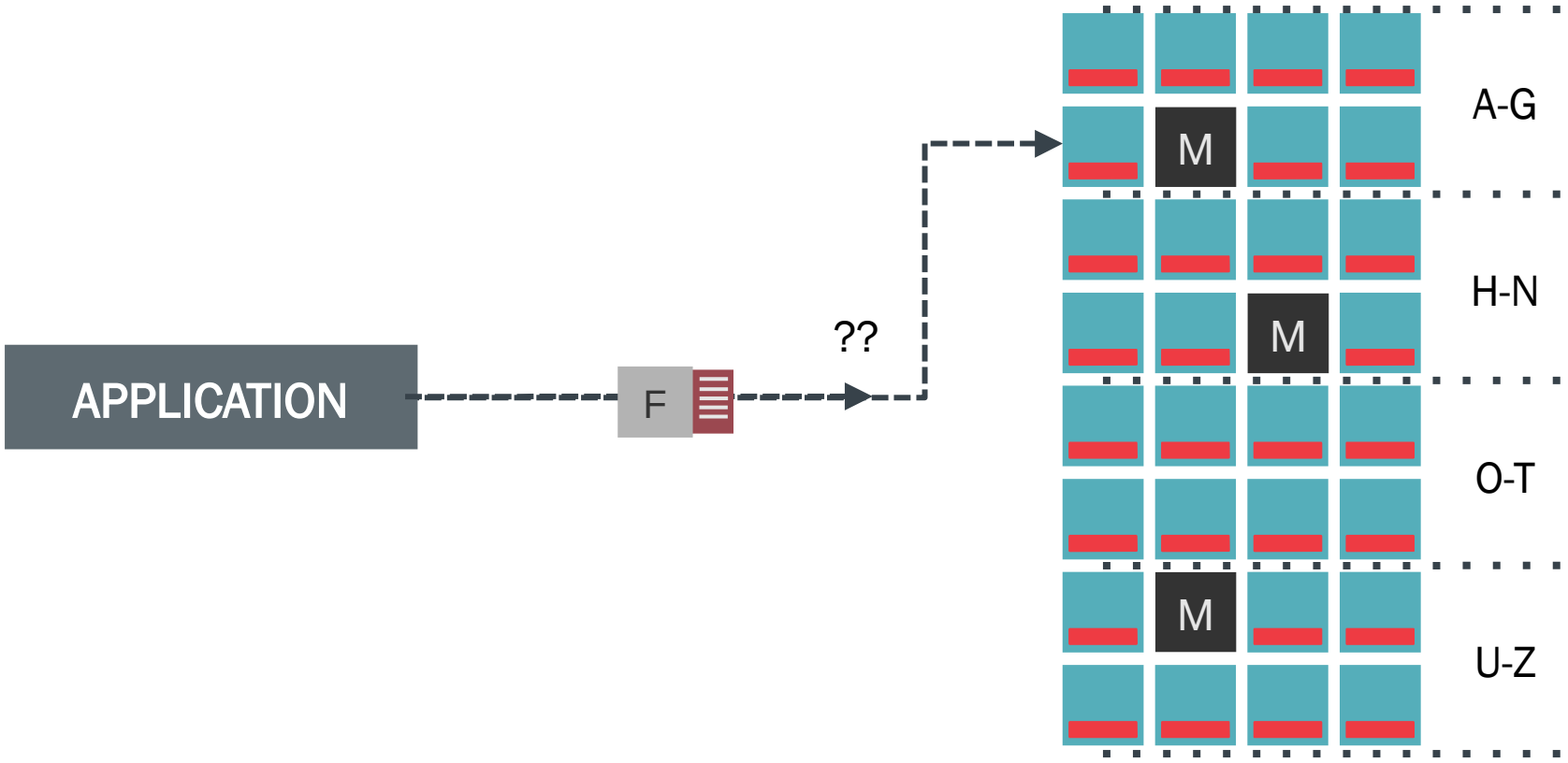
WHERE IS MY OBJECT?



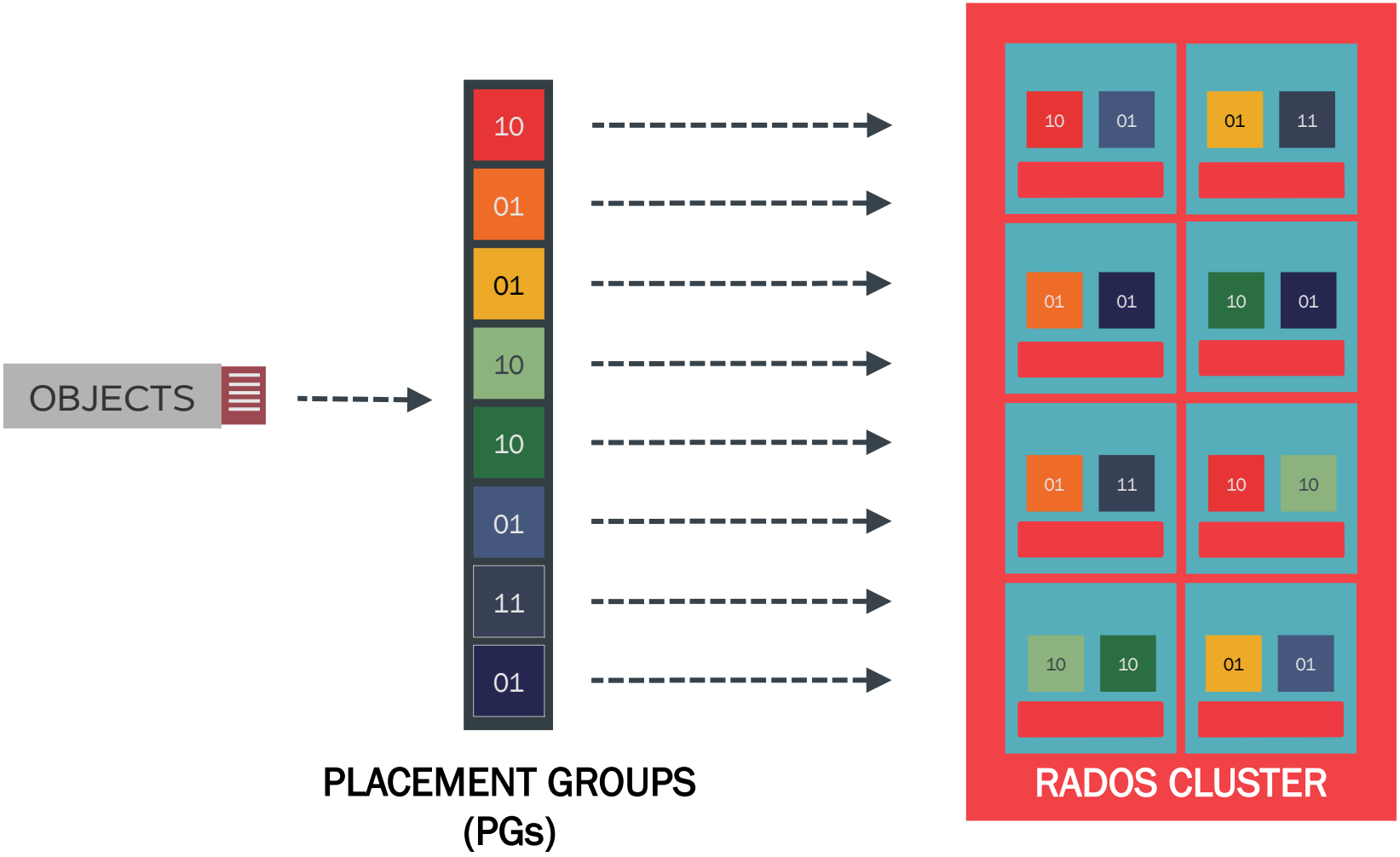
METADATA SERVER?



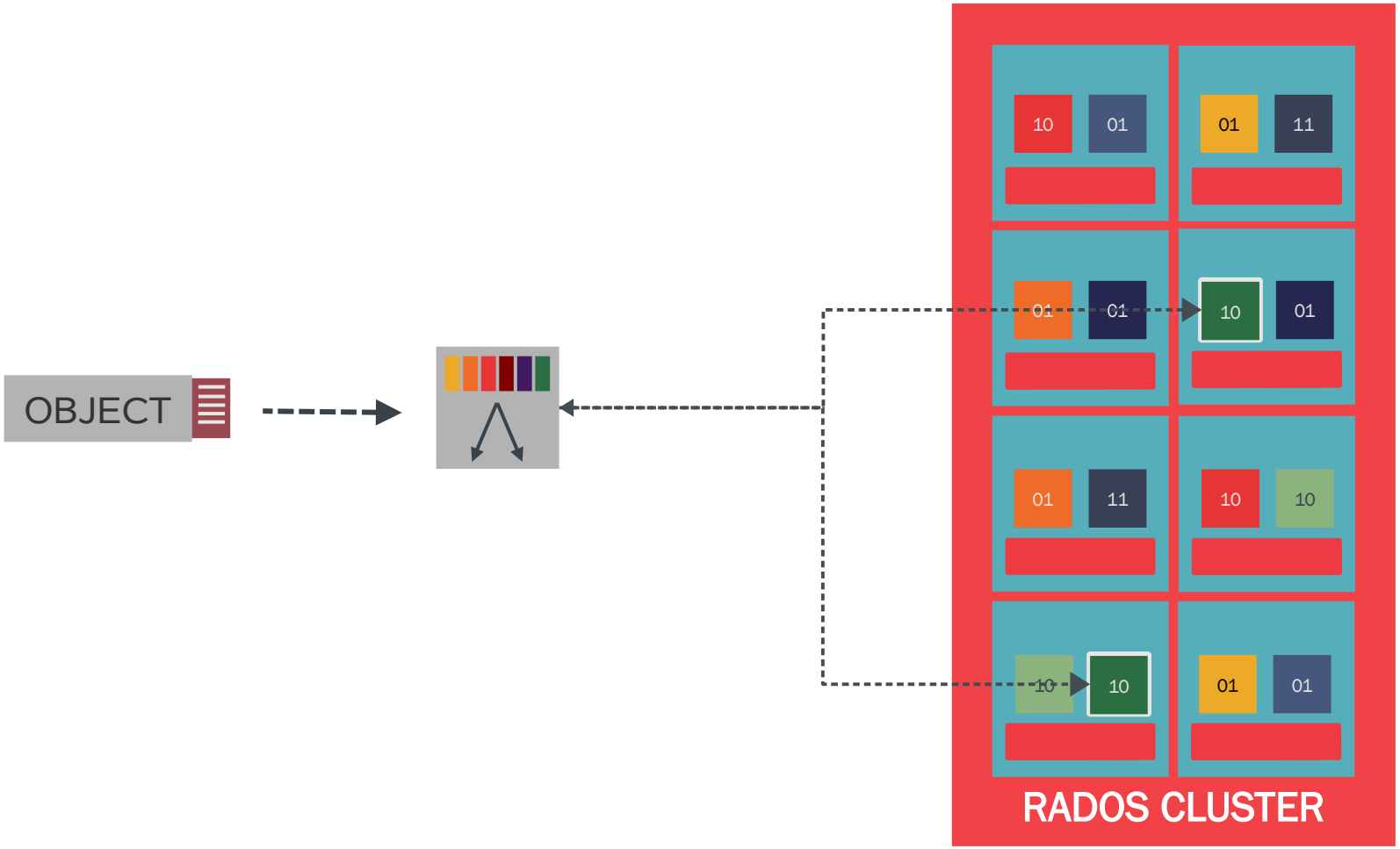
CALCULATED PLACEMENT?



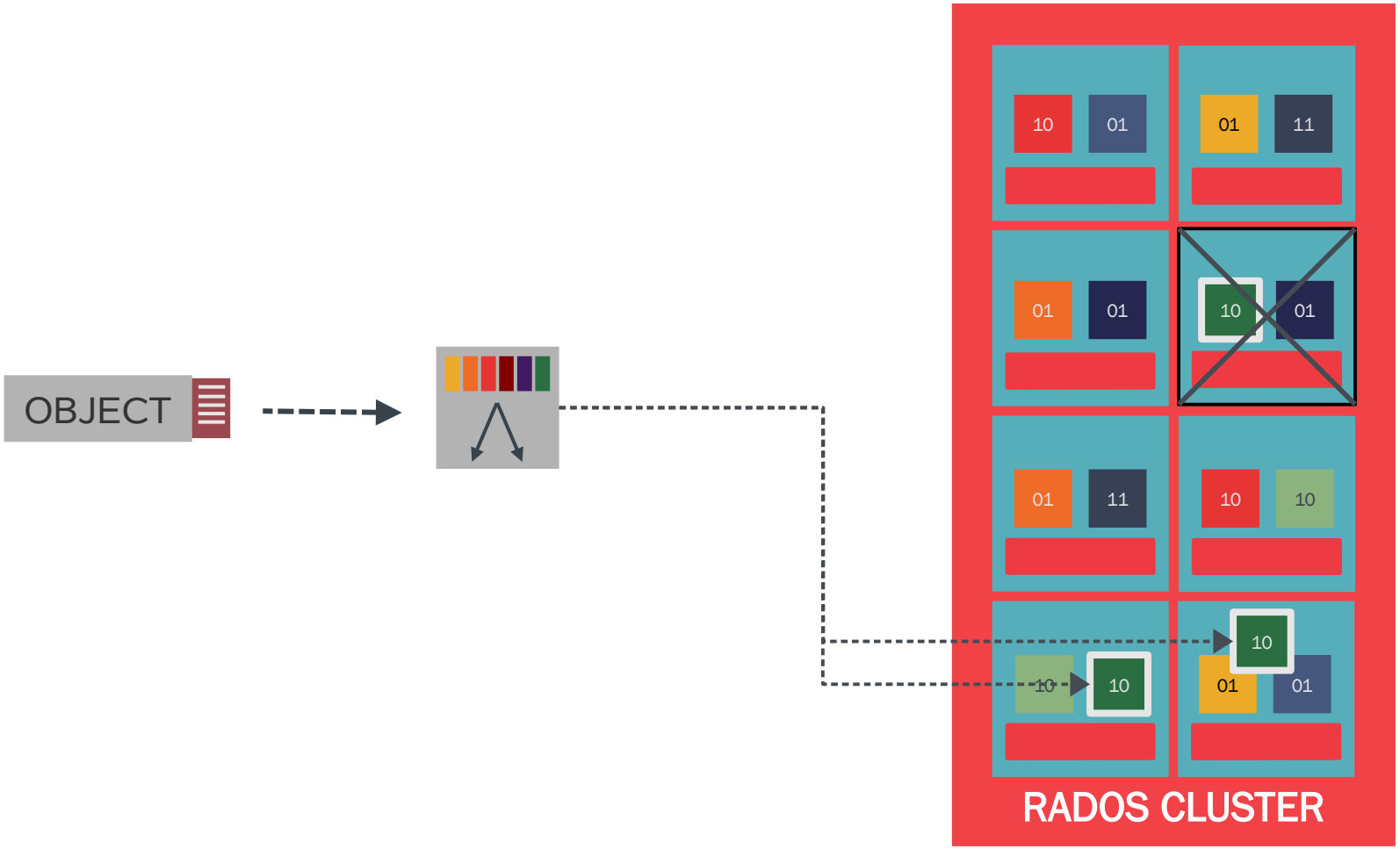
CRUSH


OBJECTS
10
01
01
10
10
01
11
01
PLACEMENT GROUPS
(PGs)
10 01
01 11
01 01
10 01
01 11
10 10
10 10
01 01
RADOS CLUSTER

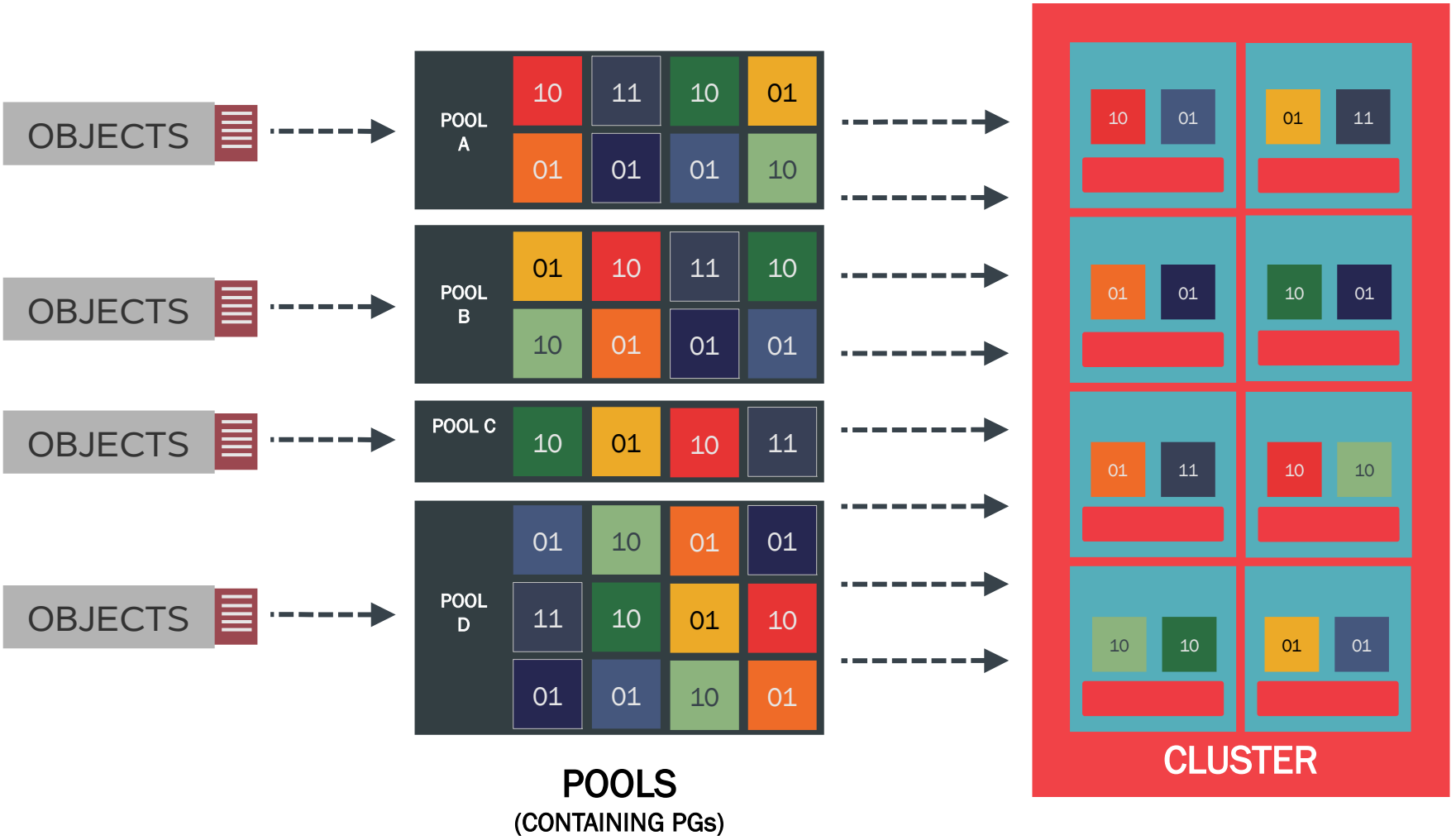
CRUSH


OBJECT
10
01
01
11
01
01
10
01
01
11
10
10
10
10
01
01
RADOS CLUSTER

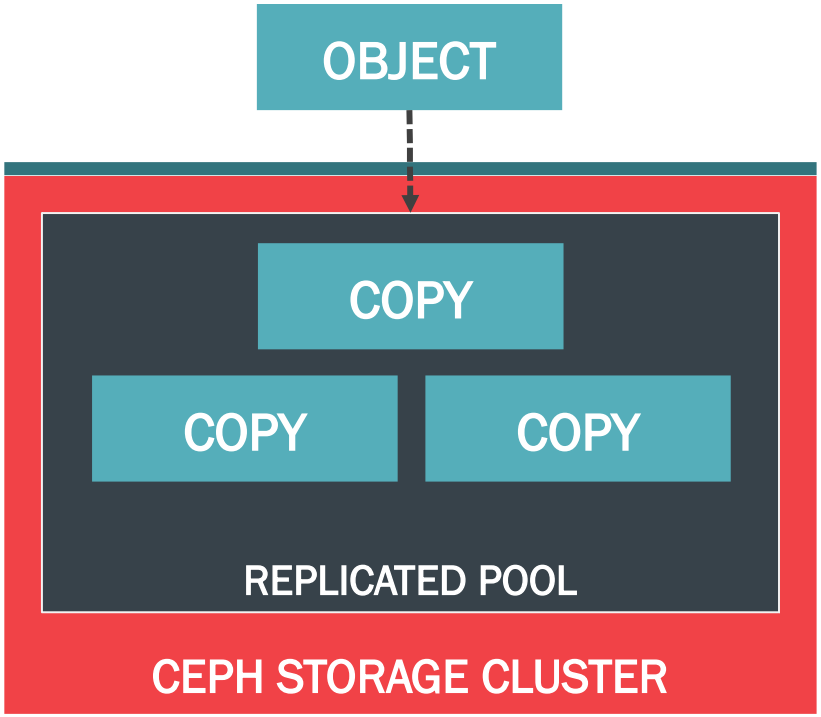
CRUSH – Failure?



DATA ORGANIZED INTO POOLS

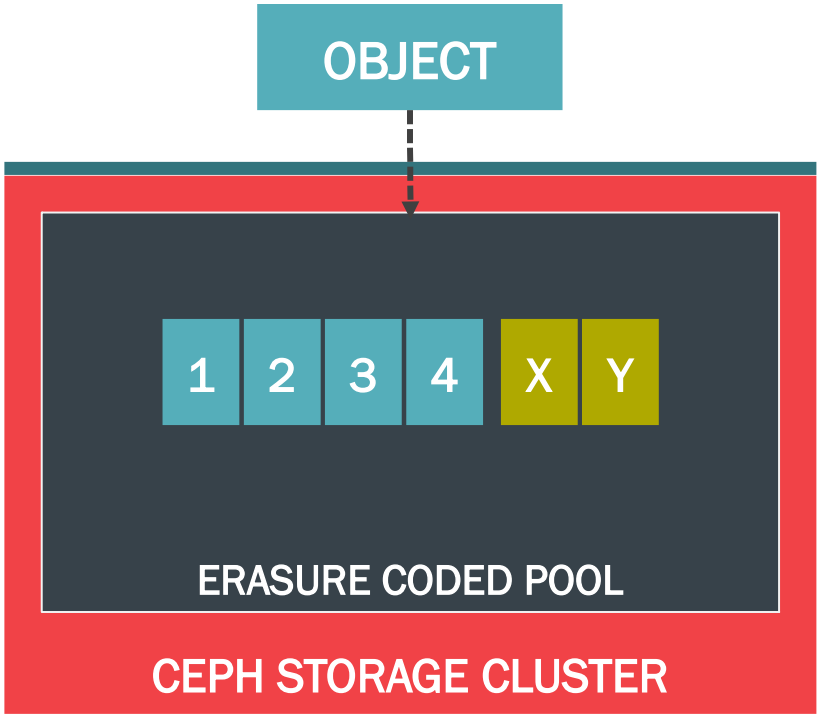


POOLS



Full copies of stored objects

- Very high durability
- 3x (200% overhead)
- Quicker recovery

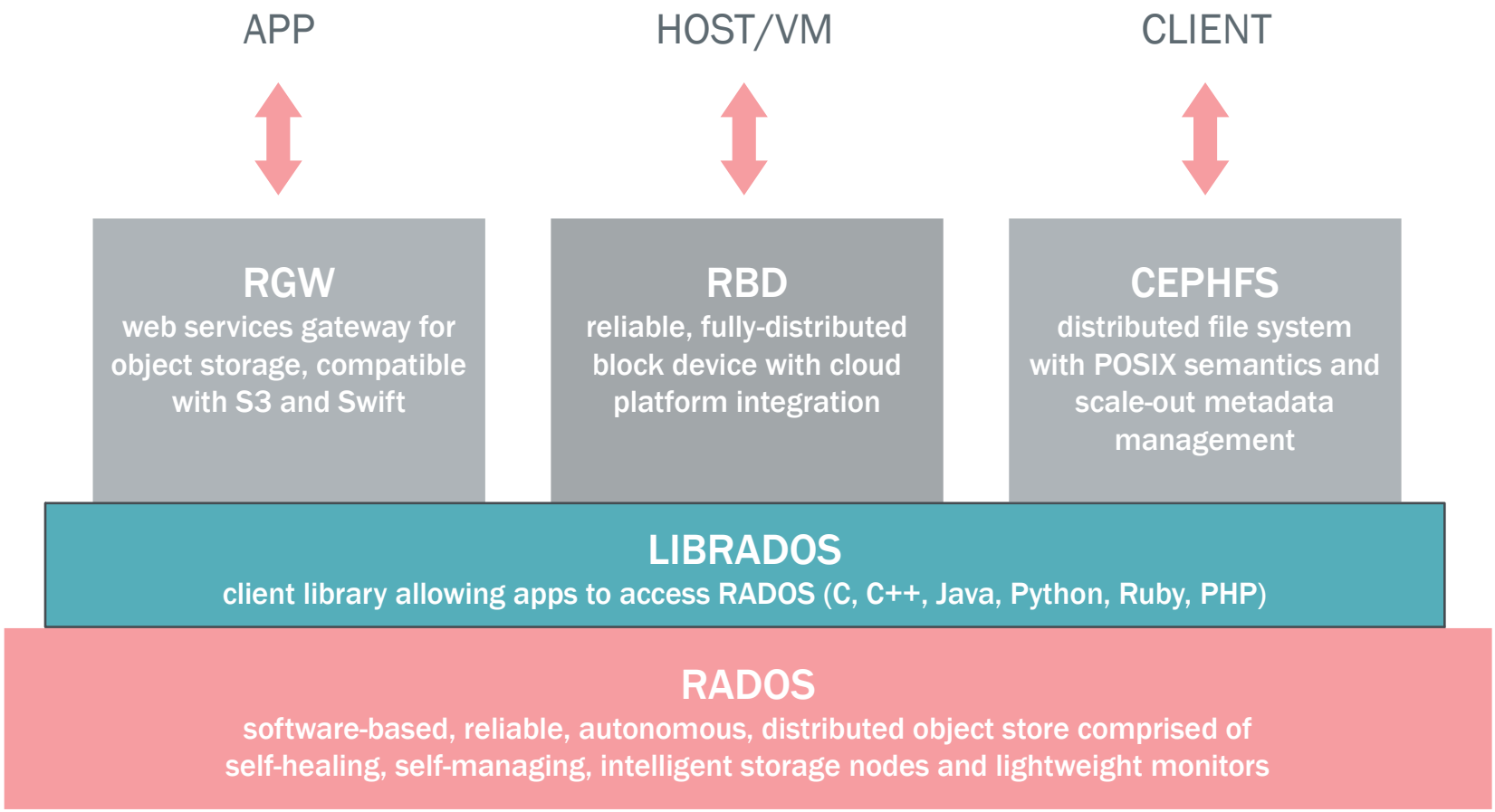


One copy plus parity

- Cost-effective durability
- 1.5x (50% overhead)
- Expensive recovery

LIBRADOS

OVERALL ARCHITECTURE



EXAMPLE `HELLO WORLD!`

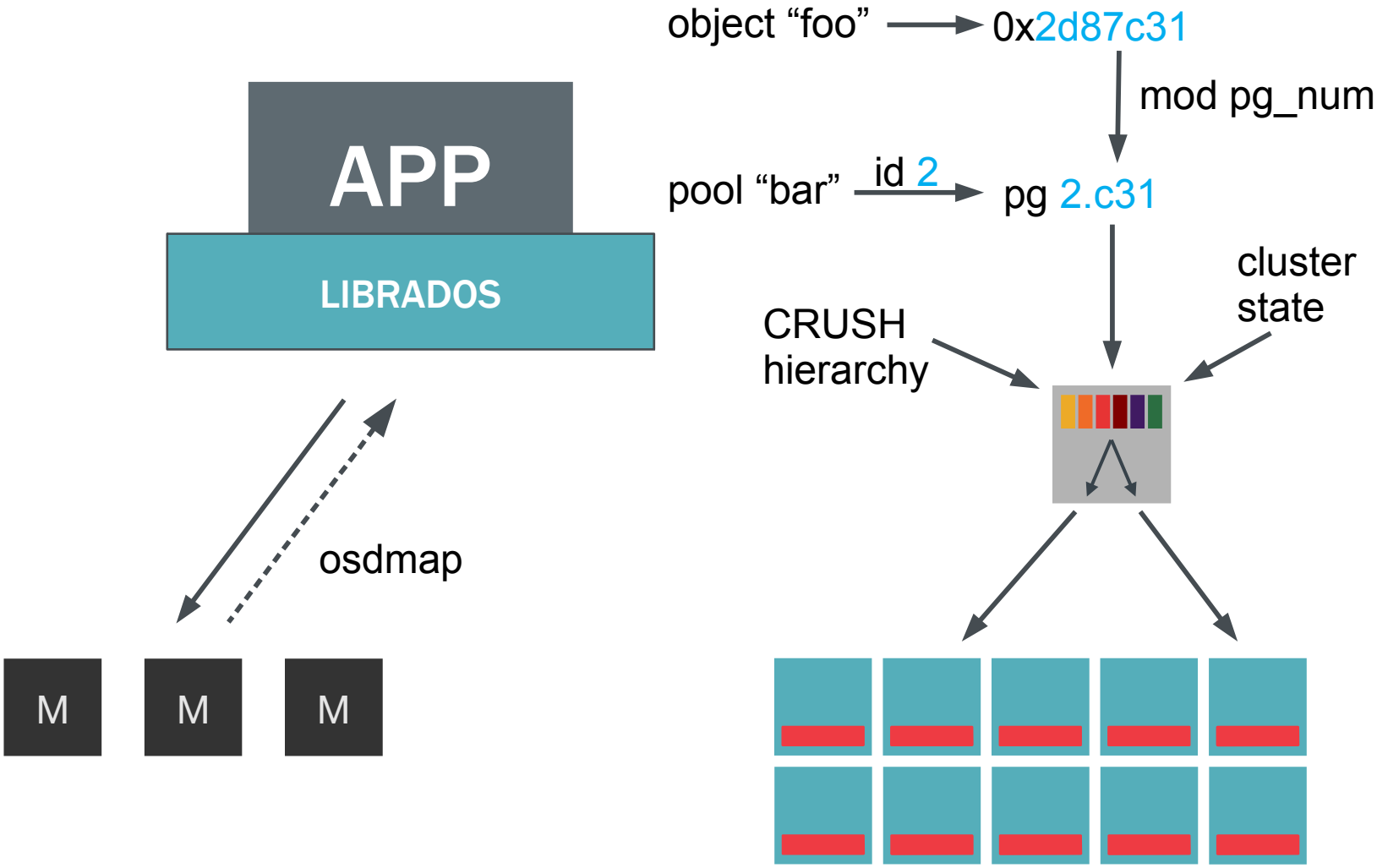
```
#include <rados/librados.hpp>

librados::Rados rados;
rados.init("admin");
rados.connect();

rados.pool_create("hello_pool");

librados::IoCtx ctx;
rados.ioctx_create("hello_pool", ctx);

bufferlist data;
data.append("hello world!");
ctx.write_full("hello_object", data);

bufferlist attr;
attr.append("1");
ctx.setxattr("hello_object", "version", attr);

rados.shutdown();
```

Connect to cluster

Create pool

Atomic (re)write

LIBRADOS


APP
LIBRADOS
M
M
M
osdmap
object “foo”
0x2d87c31
mod pg_num
pool “bar”
id 2
pg 2.c31
cluster state
CRUSH hierarchy

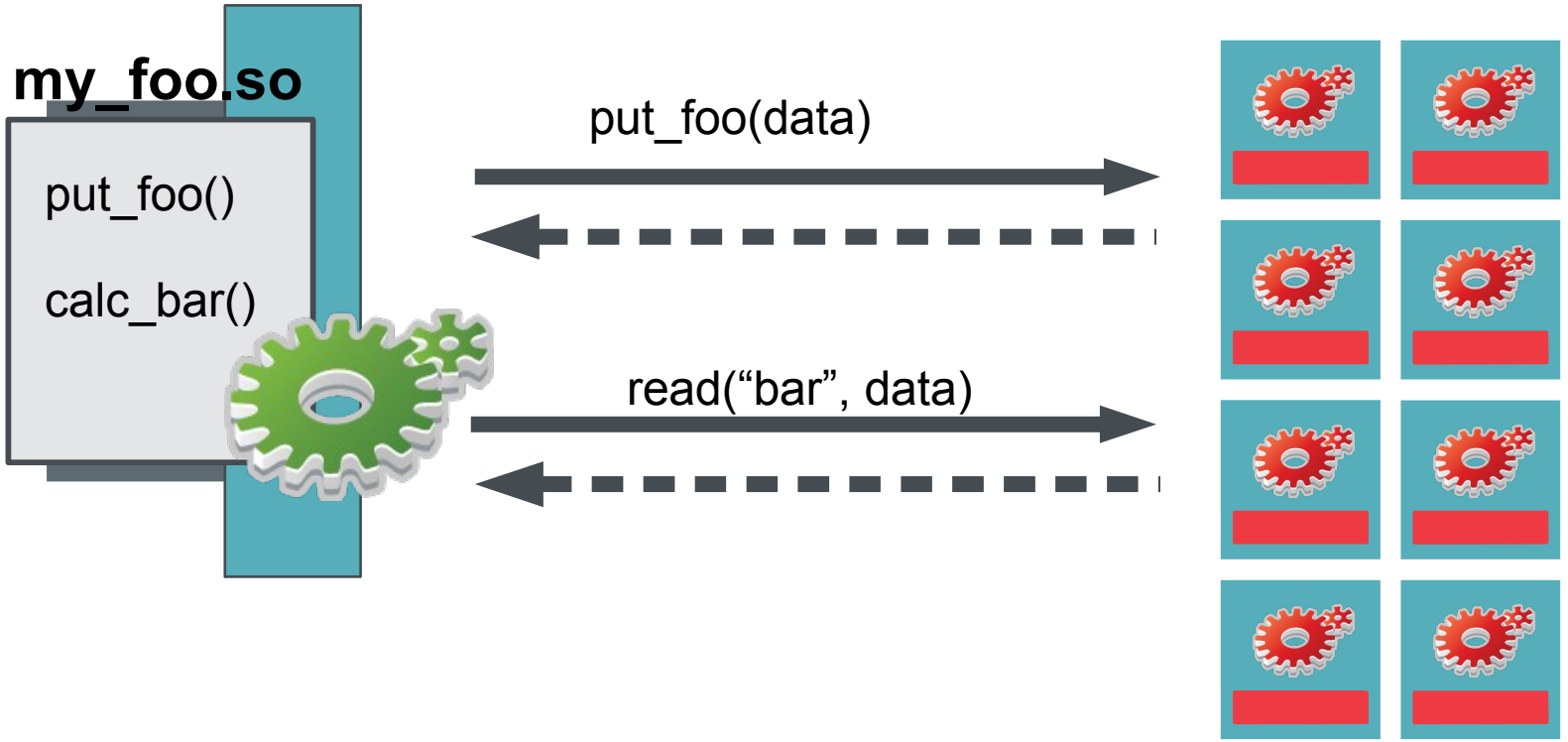
COMPOUND OBJECT OPERATIONS

```
#include <rados/librados.hpp>

librados::Rados rados;
rados.init("admin");
rados.connect();

rados.pool_create("hello_pool");

librados::IoCtx ctx;
rados.ioctx_create("hello_pool", ctx);

ObjectWriteOperation op;

bufferlist data;
data.append("hello world!");
op.write_full(data);

bufferlist attr;
attr.append("1");
op.setxattr("version", attr);

ctx.operate("hello_object", &op);

rados.shutdown();
```

Connect to cluster

Create pool

RADOS OBJECT CLASSES



EXAMPLE

Server-side class (hello_hash_class)

```
int compute_md5(cls_method_context_t hctx,
                bufferlist *in, bufferlist *out)
{
 size_t size;
 int ret = cls_cxx_stat(hctx, &size, NULL);
 if (ret < 0)
  return ret;

 bufferlist data;
 ret = cls_cxx_read(hctx, 0, size, data);
 if (ret < 0)
  return ret;

 byte digest[AES::BLOCKSIZE];
 MD5().CalculateDigest(digest,
        (byte*)data.c_str(),
        data.length());

 out->append(digest, sizeof(digest));

 return 0;
}
```

librados client

```
bufferlist input, output;

ioctx.exec("hello_object", "hello_hash_class",
           "compute_md5", input, output);
```

Compound operations

```
ObjectReadOperation op;

uint64_t  size;
time_t    m_time;
op.stat(&size, &m_time, NULL);

bufferlist in, out;
op.exec("hello_hash_class", "compute_md5",
        in, &out);

int r = op.operate("hello_object", &op);
```

REAL APPLICATIONS

· Cooperative Locking

· Simple Object Reference Counting

· Image manipulation

· RADOS Block Device (RBD) & Gateway (RGW)

sources in src/cls/*

DYNAMIC OBJECT CLASSES IN LUA

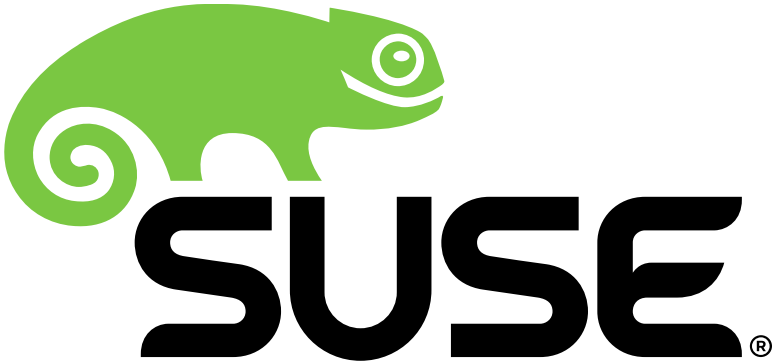
- Noah Watkins (UCSC / Red Hat)
  - http://ceph.com/rados/dynamic-object-interfaces-with-lua/

```
local script = [[
function say_hello(input, output)
  output:append("Hello, ")
  if #input == 0 then
    output:append("world")
  else
    output:append(input:str())
  end
  output:append("!")
end
cls.register(say_hello)
]]
```

```
local ret, outdata = clslua.exec(ioctx,
                                 "oid", script,
                                 "say_hello", "")
print(outdata)


local ret, outdata = clslua.exec(ioctx,
                                 "oid", script,
                                 "say_hello", "John")
print(outdata)
```

ceph-devel@vger.kernel.org
ceph-users@lists.ceph.com
#ceph / #ceph-devel @ OFTC
www.ceph.com

Thank you.


SUSE®
We adapt. You succeed.