Lustre

Paul Bienkowski
2bienkow@informatik.uni-hamburg.de

Proseminar "Ein-/Ausgabe - Stand der Wissenschaft"

2013-06-10

Outline

What is Lustre

parallel, scaling, for clusters, based within linux kernel...

The Project

Goals

2007: performance >features >stability
“itś a science project”
2010: stability >performance >features
used in high-performance production environments

History

- started as a research project in 1999 by Peter Braam
- Braam founs **Cluster File Systems**
- 1.0 released in 2003
- **Sun Microsystems** aquires Cluster File Systems in 2007
- **Oracle Corporation** aquires Sun Mircrosystems in 2010
- Oracle ceases Lustre development, many new Organizations continue development, including **Xyratec**, **Whamcloud**, and more
- in 2012, **Intel** aquires Whamcloud
- in 2013, Xyratec purchases the original Lustre trademark from Oracle

Who is involved?

Oracle *no development*, only pre-1.8 support

Intel funding, preparing for *exascale computing*

Cray funding, development (Titan Supercomputer)

Xyratex hardware bundling

OpenSFS (Open Scalable File Systems) "keeping Lustre open"

EOFS (EUROPEAN Open File Systems) (community collaboration)

FOSS Community many joined one of the above to help development (e.g. Braam works for Xyratex now)

DDN, Dell, NetApp, Terascala, Xyratex storage hardware bundled with Lustre

Supercomputers

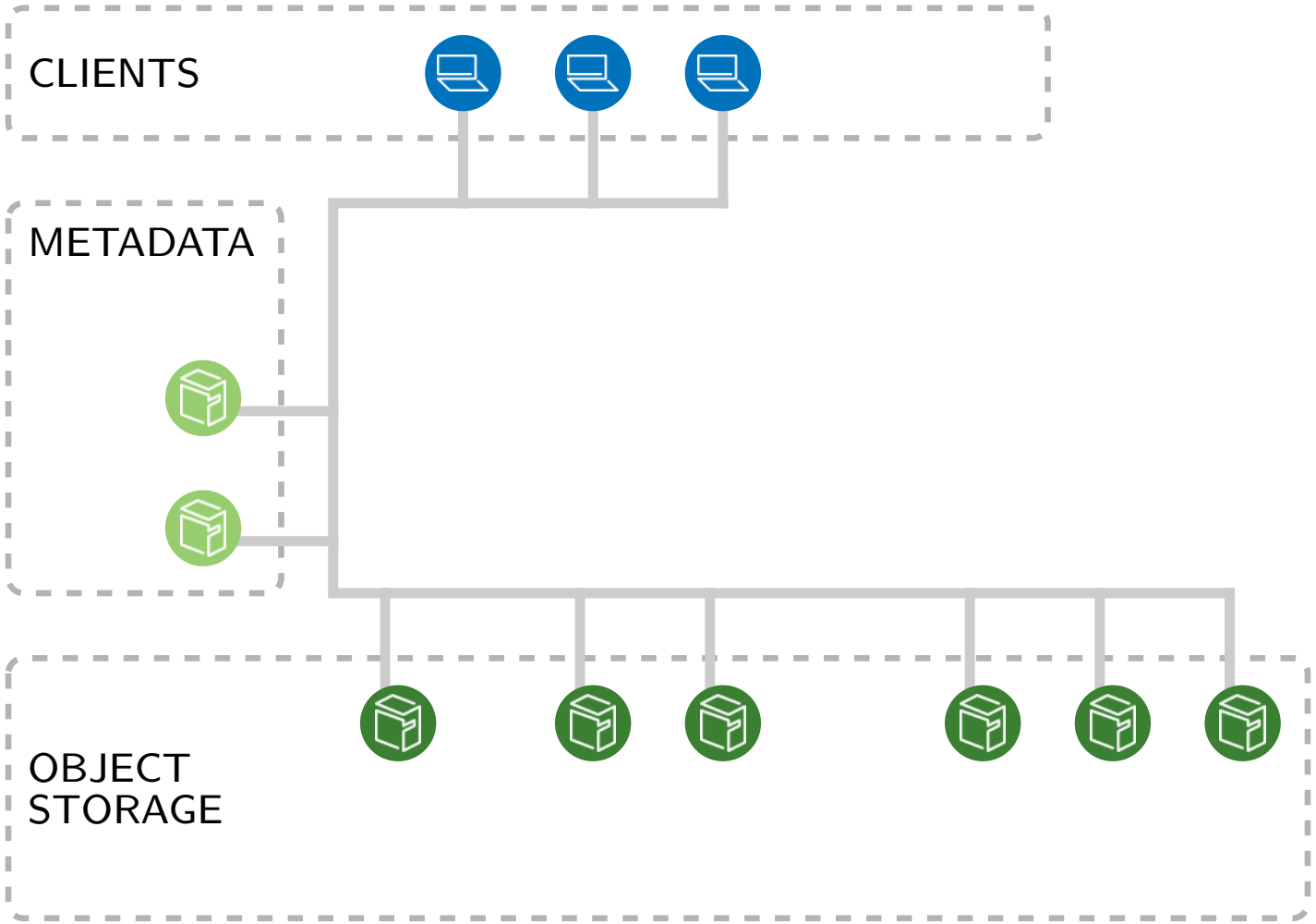
Titan & Co. use it!

Lustre Architecture

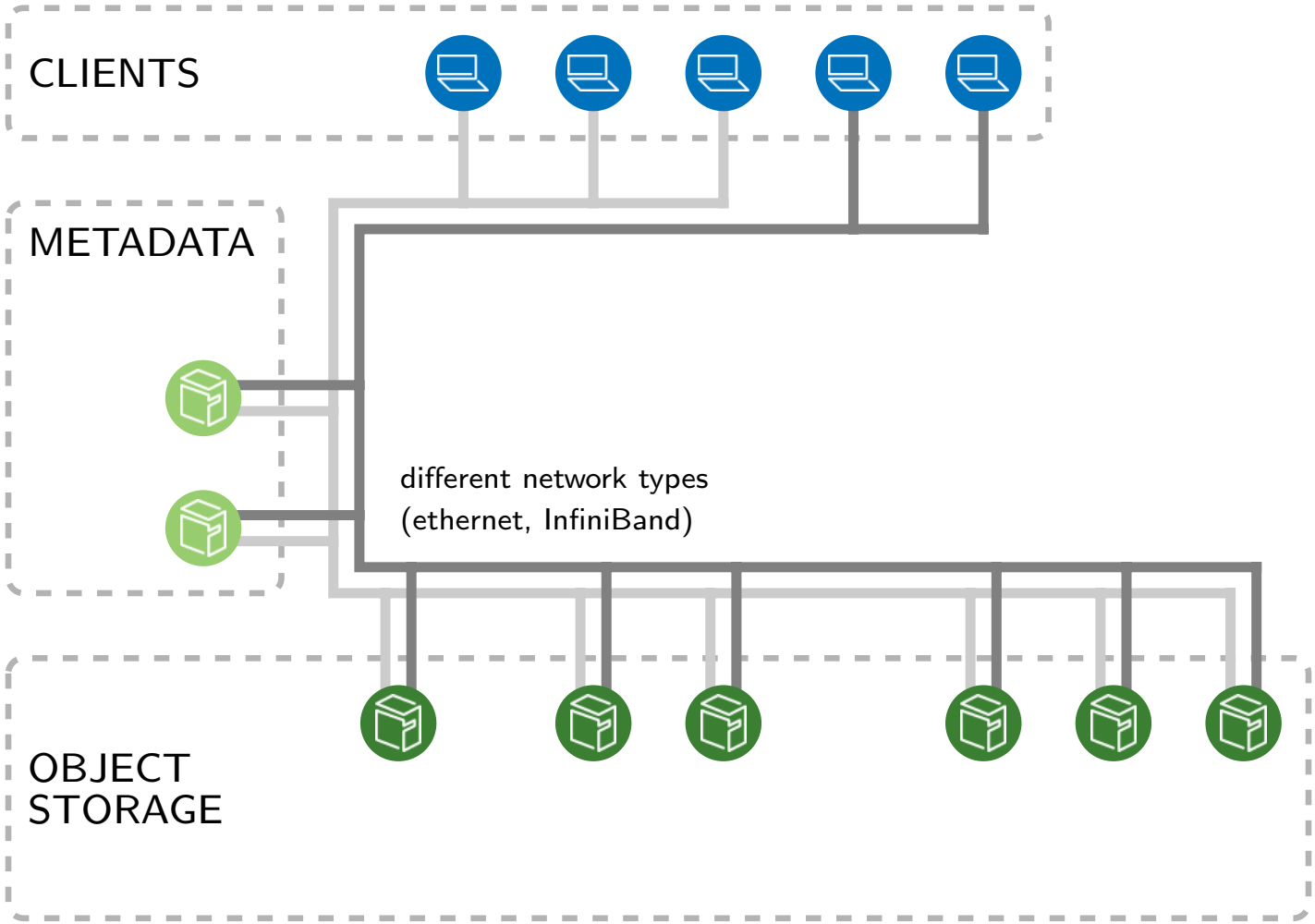
Network Structure

Network Structure

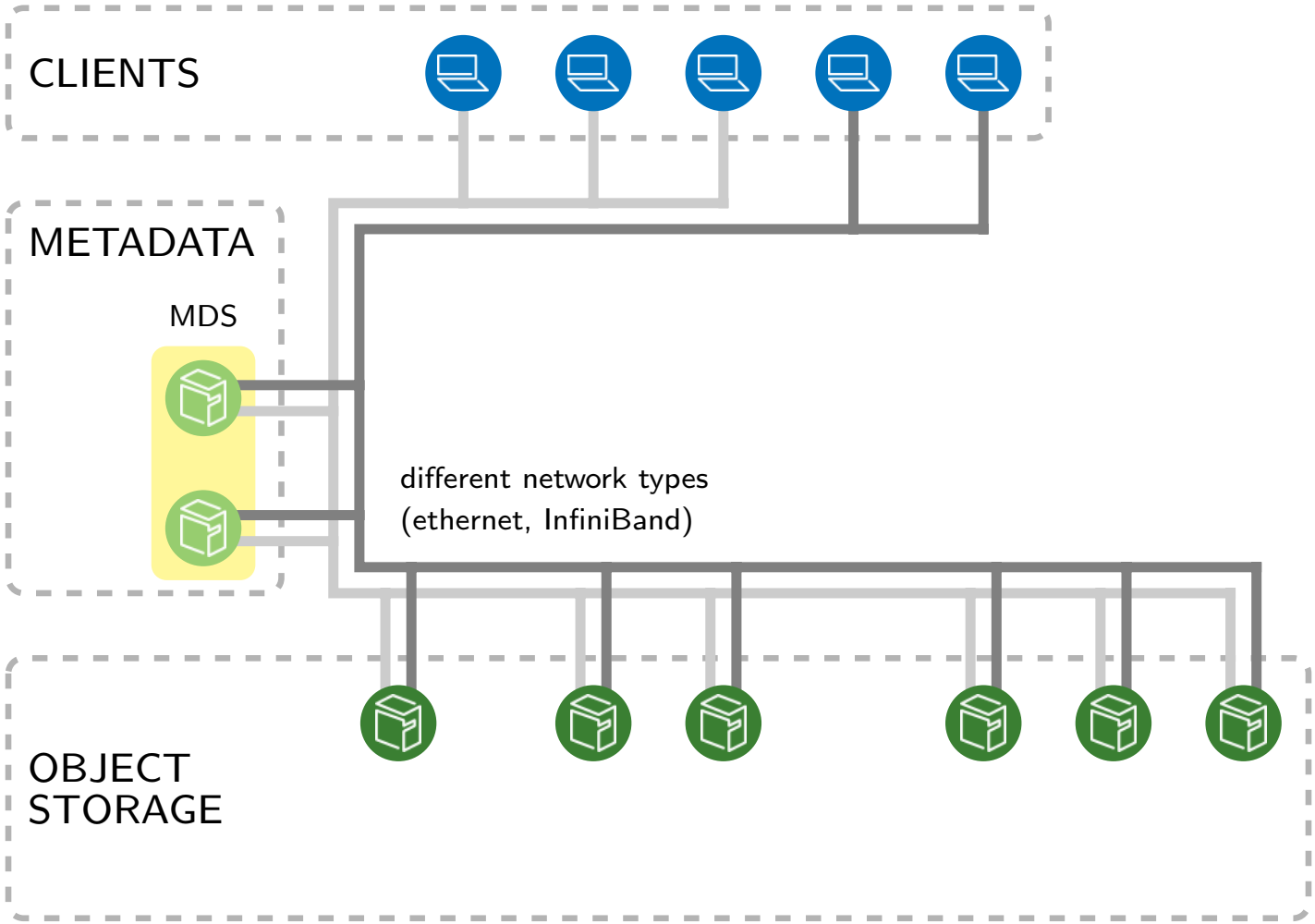
Network Structure



Network Structure



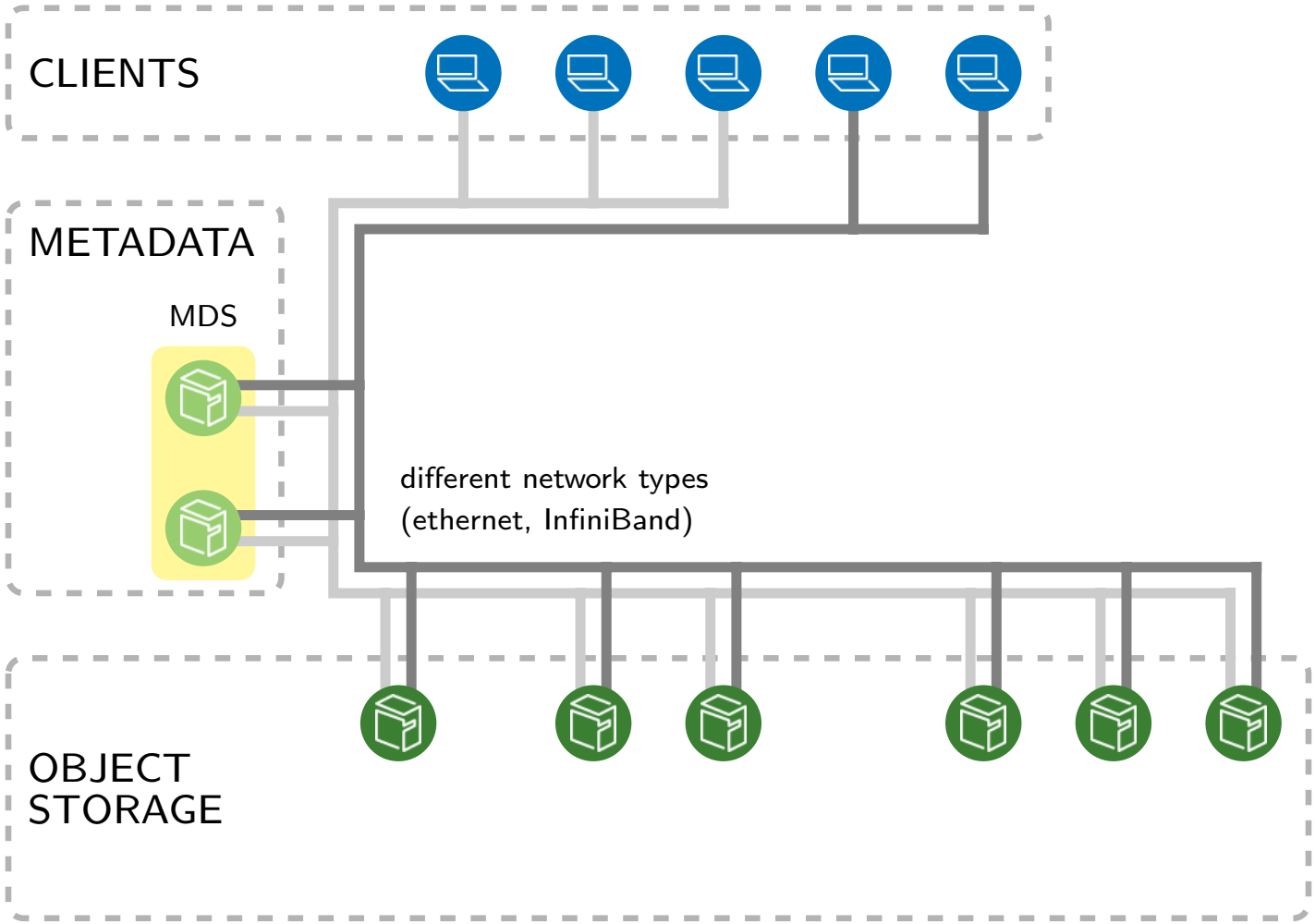
Network Structure



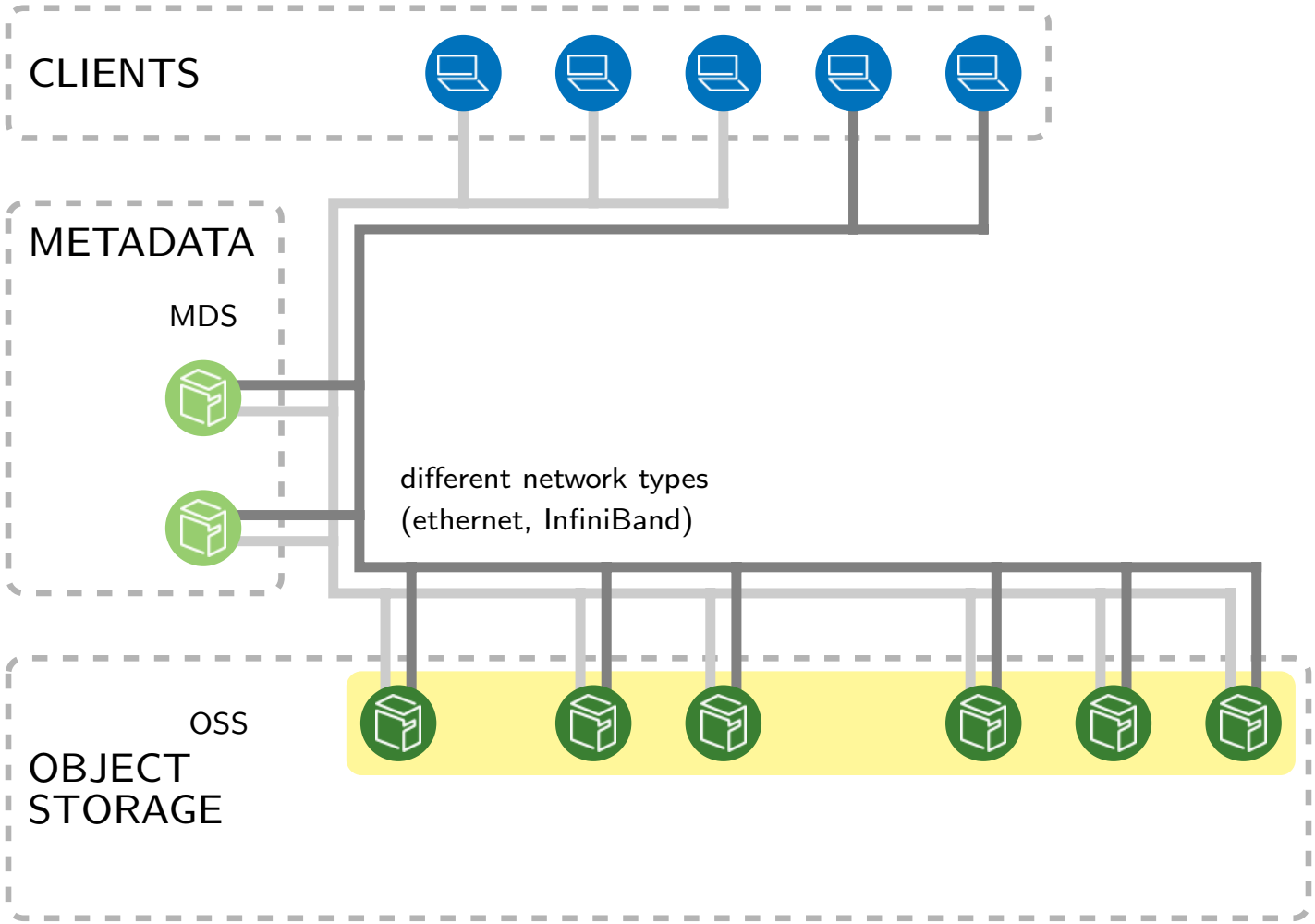
Metadata Server (**MDS**)

- store file information (metadata)
- accessed by clients to access files
- *manage* data storage
- at least one required
- up to ∼100 possible (failovers)

Network Structure



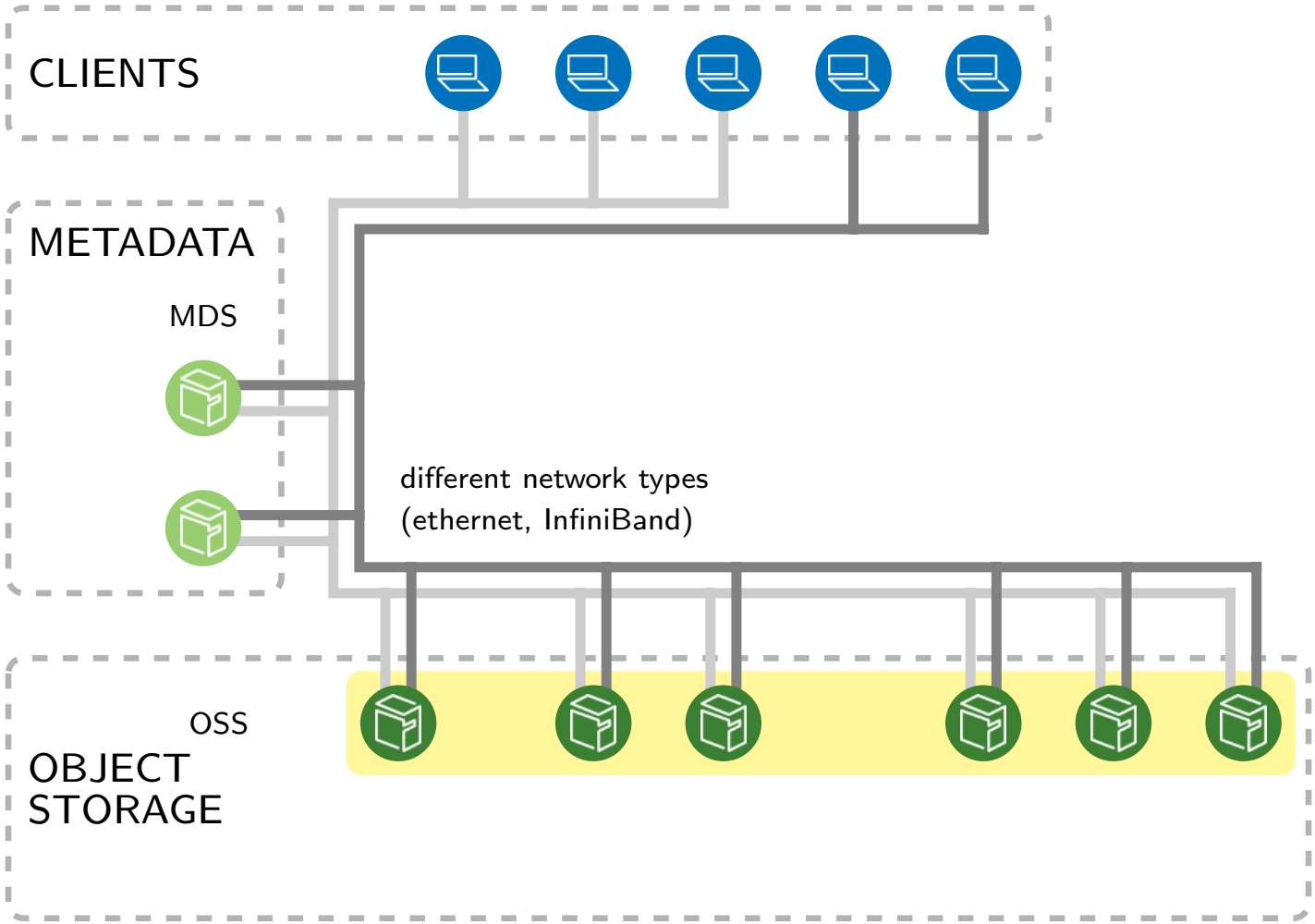
Network Structure



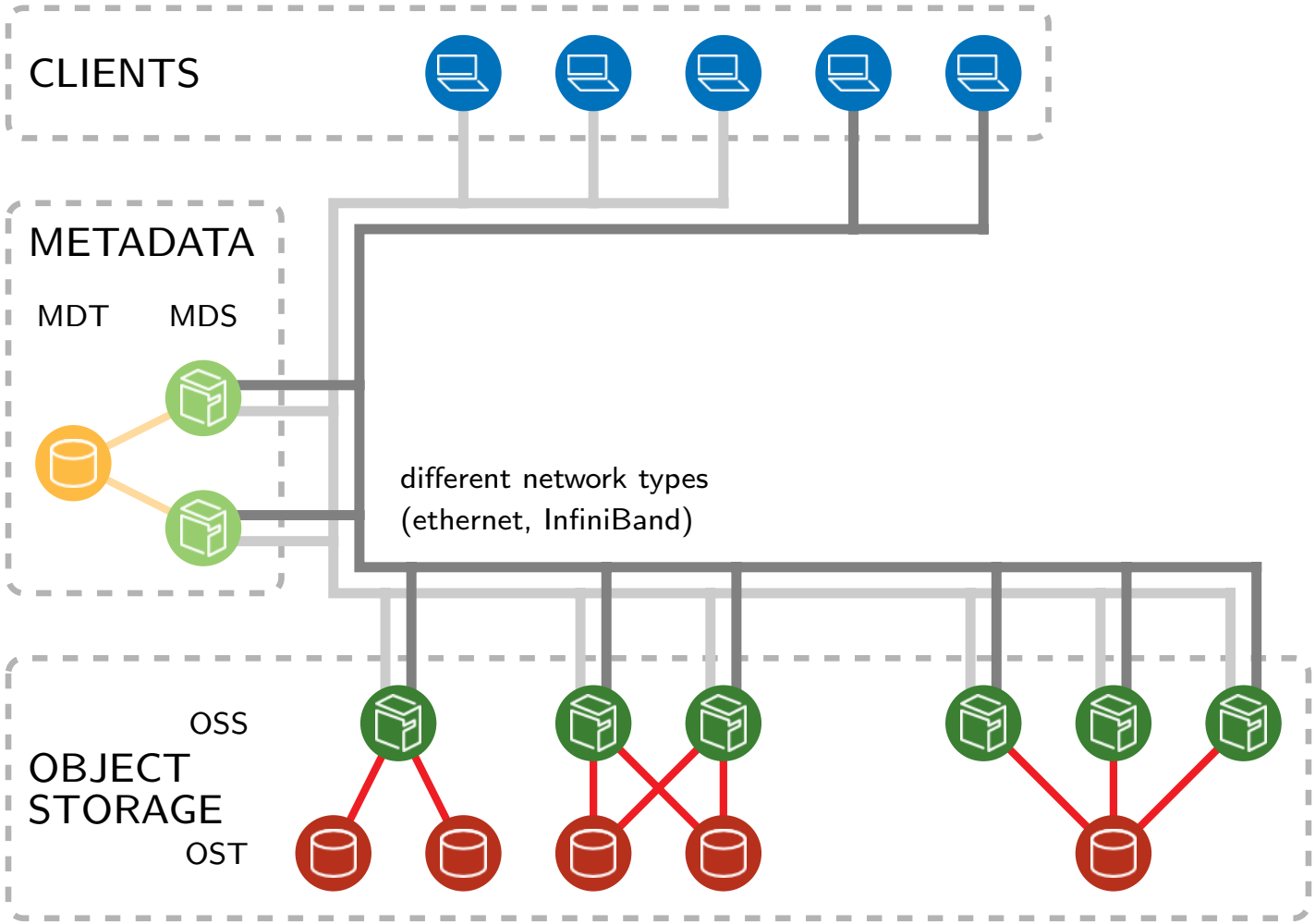
Object Storage Server (**OSS**)

- store file content (objects)
- accessed by clients directly
- at least one required
- $>$ 10000 OSS are used in large scale computers
- multiple targets per server
- multiple servers per target

Network Structure



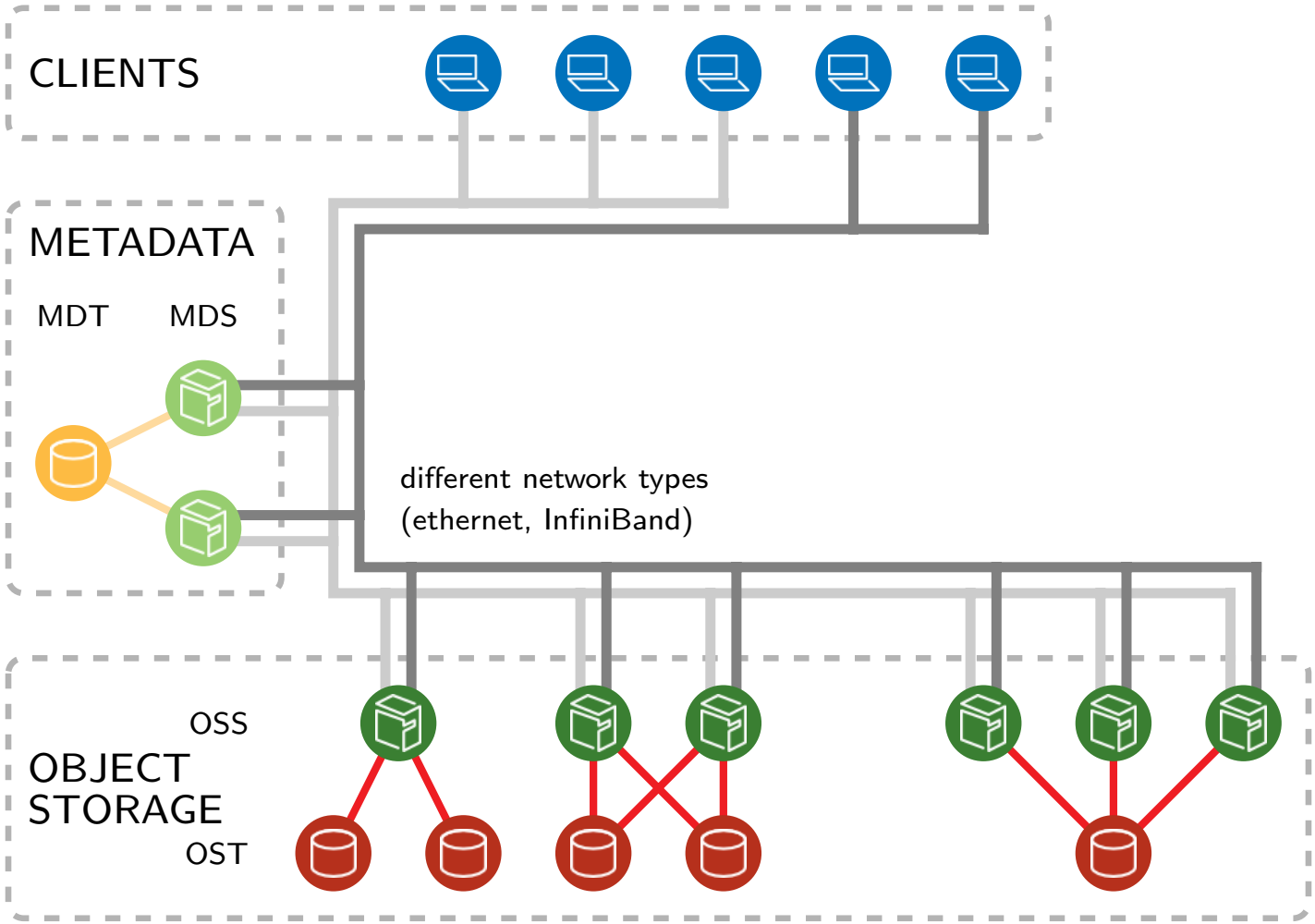
Network Structure



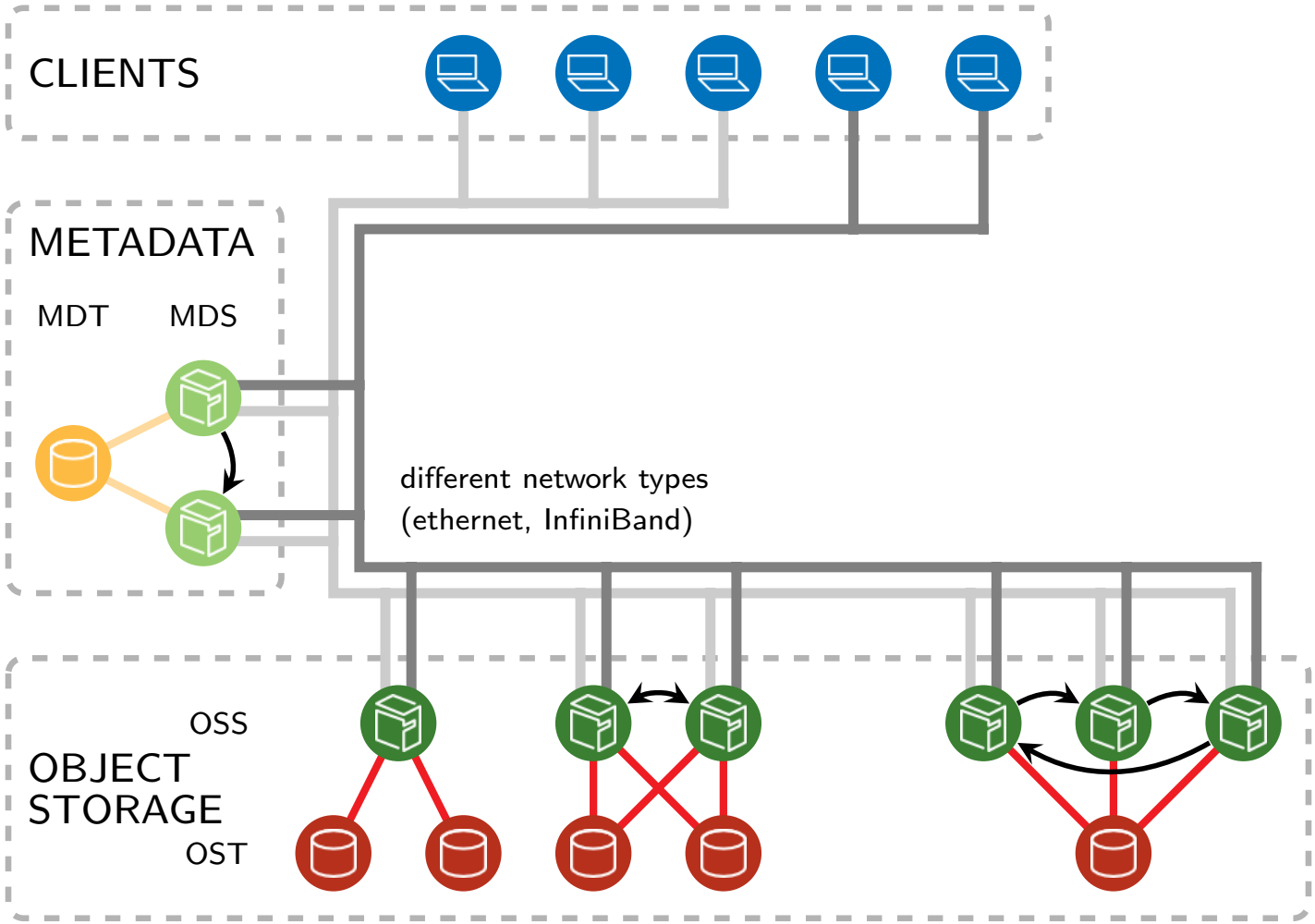
Targets

- two types
  - object storage target (OST)
  - metadata target (MDT)
- can be any block device
  - normal hard disk / flash drive / SSD
  - advanced storage arrays
- will be formatted for lustre

Network Structure



Network Structure



Failover

- if one server failes, another one takes over
- backup server needs access to targets
- enabled on-line software upgrades (one-by-one)

Network Capabilities

How is data transfered?
Protocol stack (TCP, ...)
Different network types (ethernet, infiniband, ...)

Excursion: INodes

... because MDS do something similar (metadata records)
(graph)
compare this on next slide

Metadata

how metadata is stored in the MDS
what metadata is stored?
how metadata is fetched from the MDS

Striping

[repeat] what is striping (RAID 0)
why do they use it in lustre – speed advantage

Software Architecture

what software is running where?

Interversion Compatibility

Sun "guarantees" [citation needed] compatibbility between minor versions
$\rightarrow$ on-line upgrade-ability using failover systems

ldiskfs - Customized ext3

why we need a customized filesystem to work ON TOP of

Kernel patching (serverside)

just tell them the kernel needs to be patched (2.6.*) and what that means

Patchfree Client

How can clients access the data? (lustre-fs, liblustre, NFS)
(kernel-independent)
even NFS, that works everywhere!

Limitations

Server

very platform dependent

needs compatible kernel

Client

all linux kernel >2.6 supported

NFS for Windows, MacOS

even FUSE support on the way

Performance

Throughput Examples

Yes, the speeds add up!
There are systems with 5000 OSS.
Up to 160 OSS / file.
16 OST/OSS is quite normal.
1 TiB/OST

Scalability

Just multiply. Works as long as your network supports it (e.g. InfiniBand >WiFi ...)

???

Maybe I can find some more data samples – where is the interesting stuff?

Conclusion

...

References

...