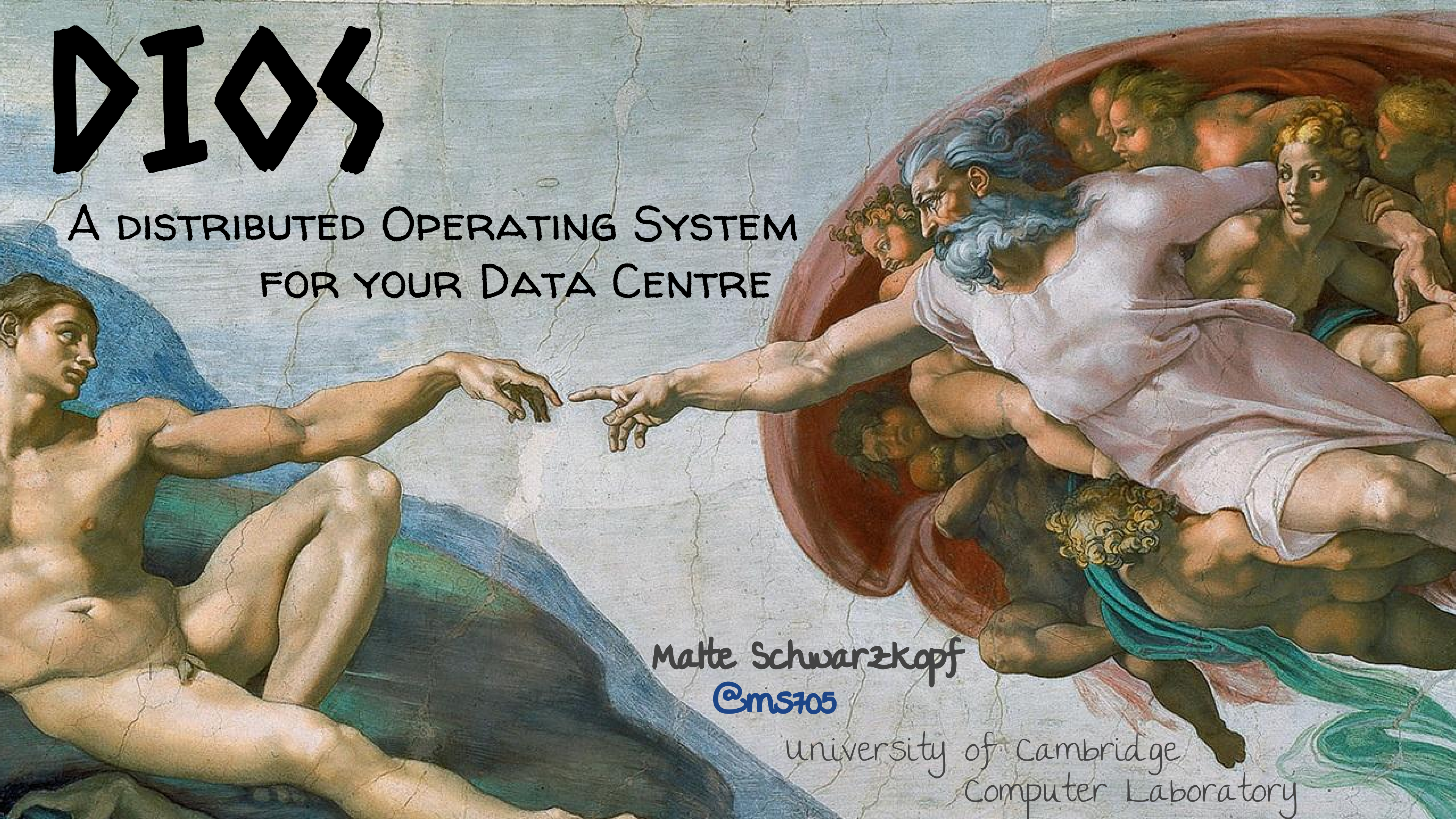
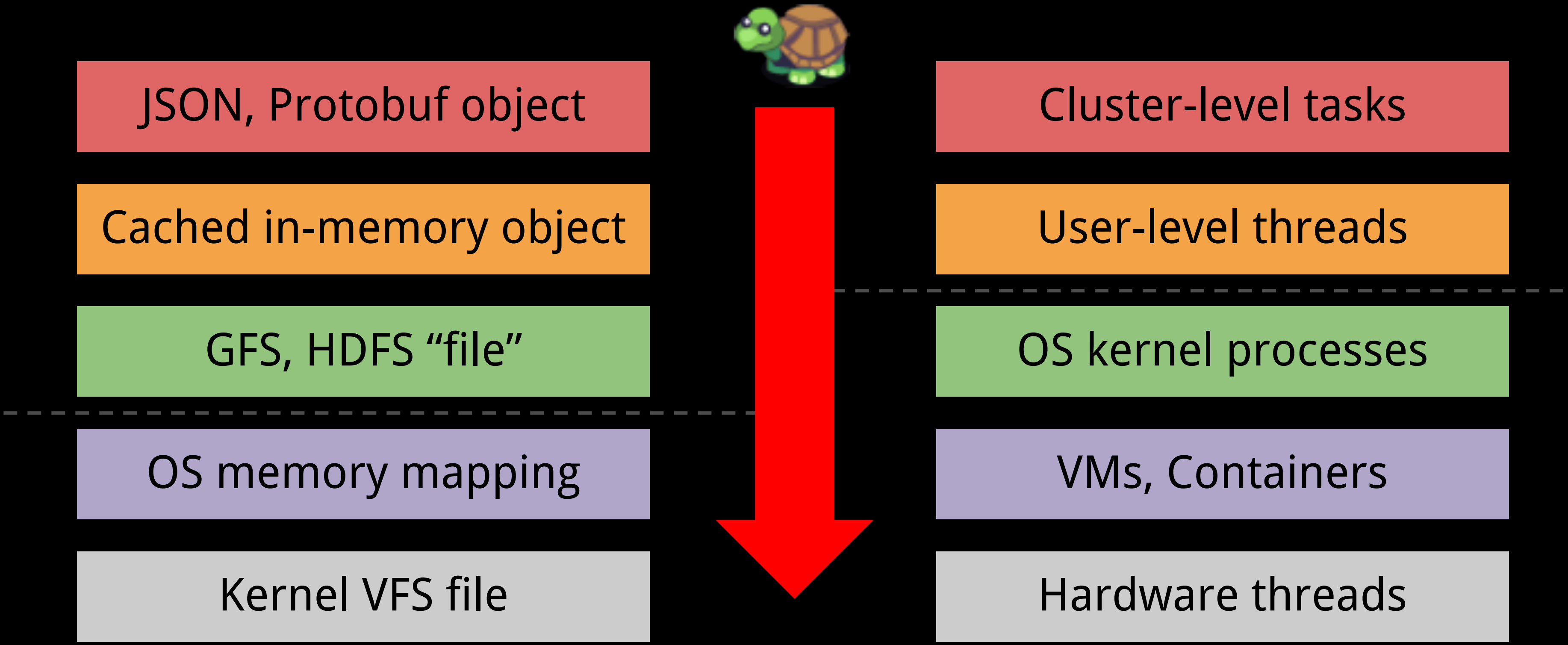


DIOS

A distributed Operating System for your Data Centre

Malte Schwarzkopf
@ms705
University of Cambridge
Computer Laboratory

Abstraction turtles all the way down...



JSON, Protobuf object	Cluster-level tasks
Cached in-memory object	User-level threads
GFS, HDFS “file”	OS kernel processes
OS memory mapping	VMs, Containers
Kernel VFS file	Hardware threads

Good for…

abstraction

portability

BAD for...

scalability

data-flow tracking

co-scheduling

security

locality

optimisations

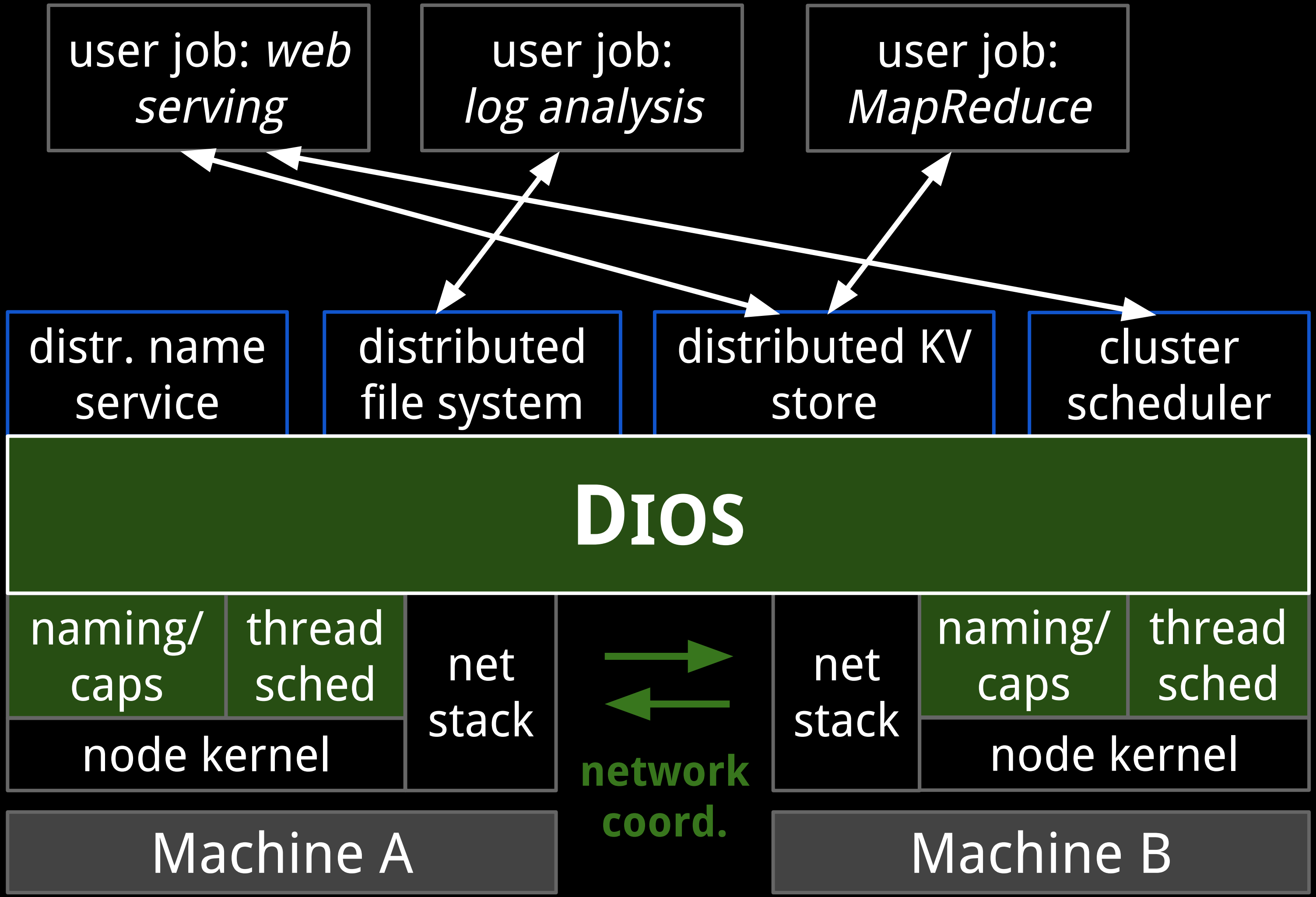
The plan:
vertically integrate abstractions

Distributed **application**

Distributed **infrastructure**

Distributed **operating system**

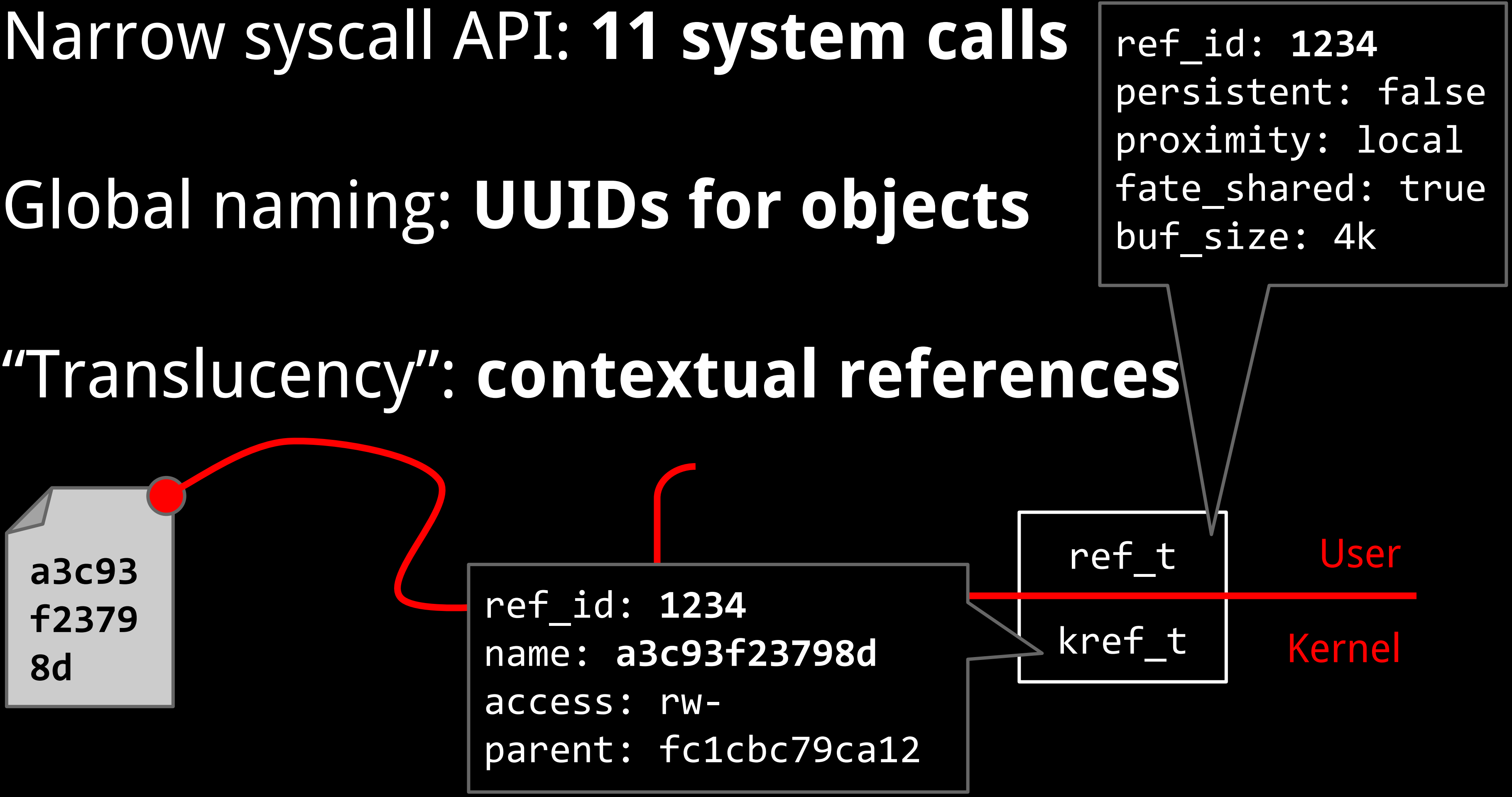
All use: **one** distributed object abstraction


user job: *web serving*
user job: *log analysis*
user job: *MapReduce*
distr. name service
distributed file system
distributed KV store
cluster scheduler
DIOS
naming/ caps
thread sched
net stack
node kernel
network coord.
net stack
naming/ caps
thread sched
node kernel
Machine A
Machine B

Narrow syscall API: **11 system calls**

Global naming: **UUIDs for objects**

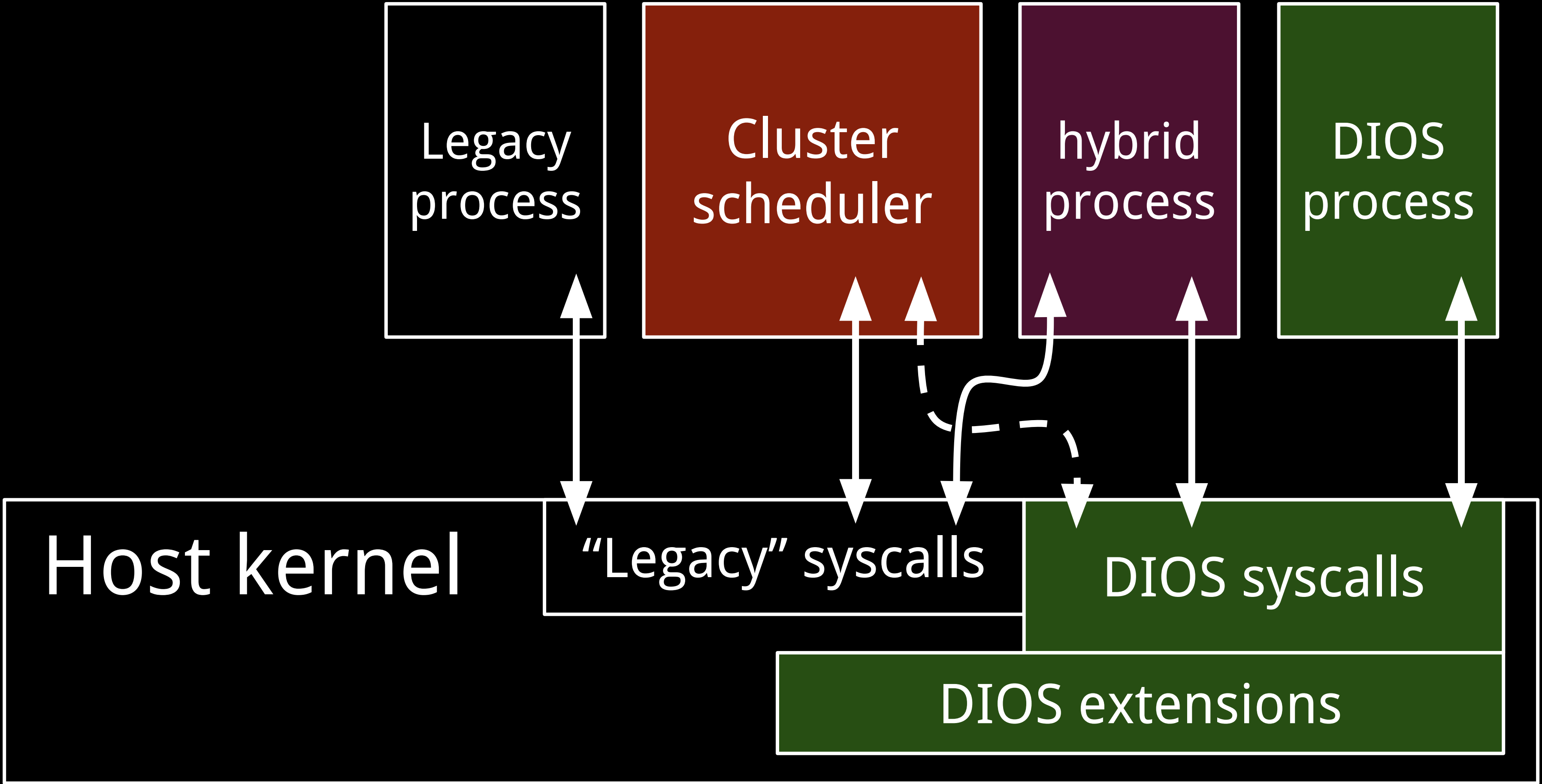
“Translucency”: **contextual references**

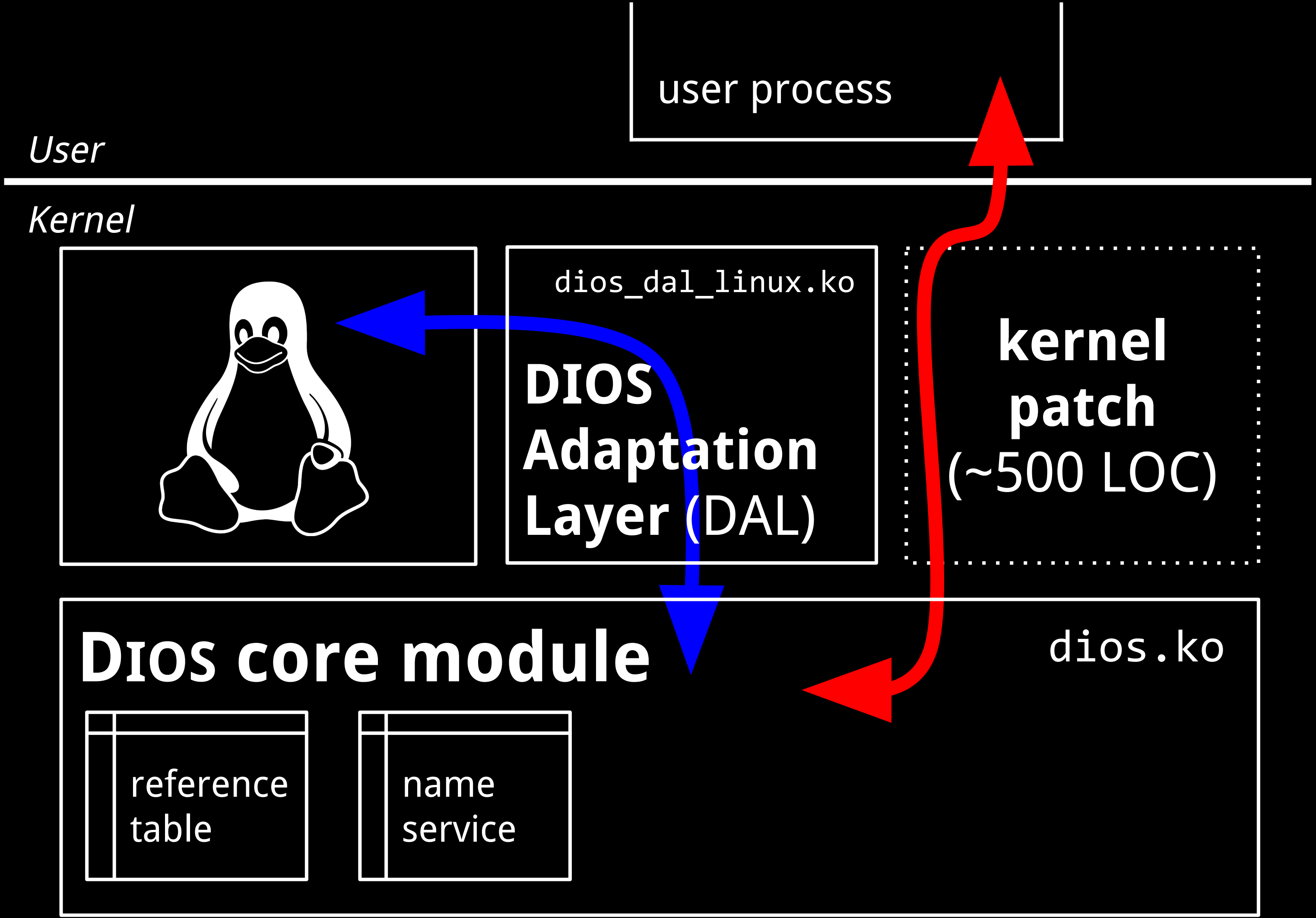


ref_id: **1234**
persistent: false
proximity: local
fate_shared: true
buf_size: 4k

a3c93
f2379
8d

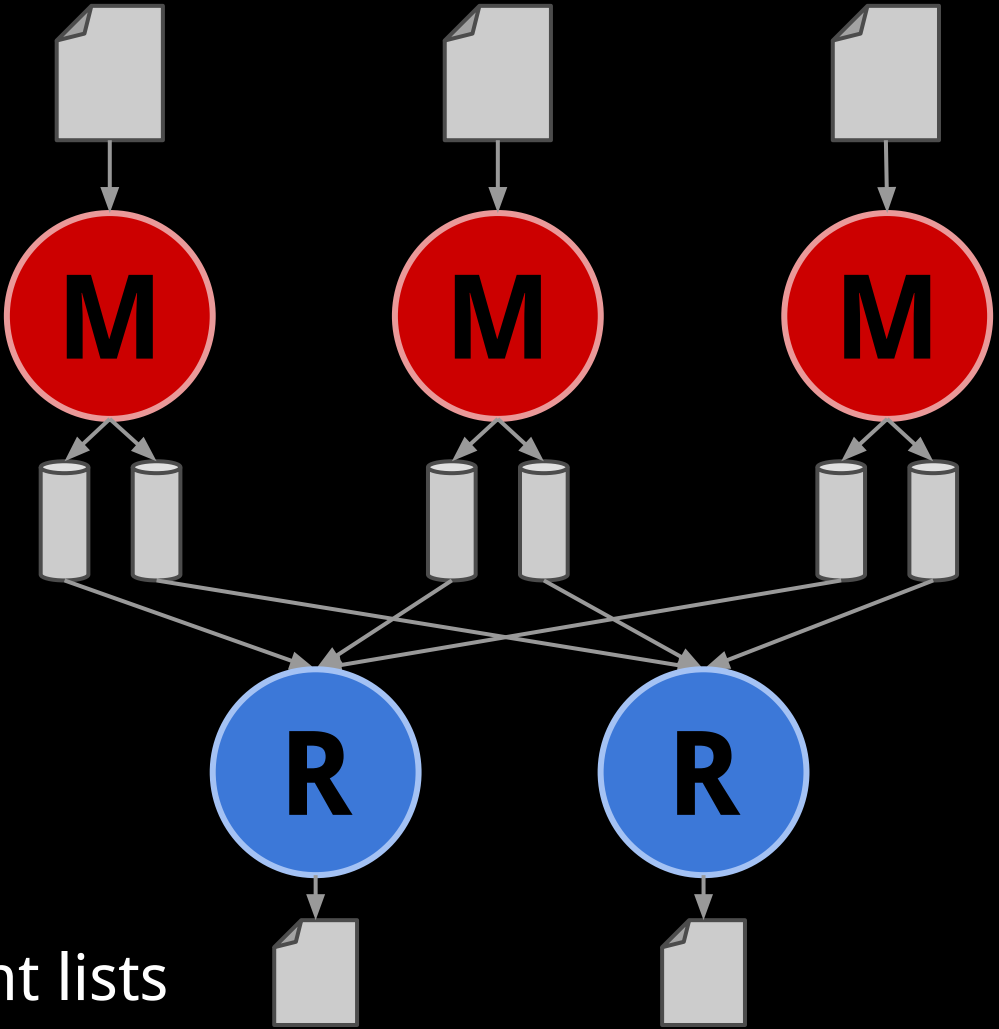
ref_id: **1234**
name: **a3c93f23798d**
access: rw-
parent: fc1cbc79ca12


Legacy process
Cluster scheduler
hybrid process
DIOS process
Host kernel
“Legacy” syscalls
DIOS syscalls
DIOS extensions


user process
User
Kernel
dios_dal_linux.ko
DIOS Adaptation Layer (DAL)
kernel patch (~500 LOC)
DIOS core module
dios.ko
reference table
name service

Demo time!

(this is where the kernel crashes...)



input

map

```
<"cat", 1>
<"dog", 1>
<"cat", 1>
<"fish", 1>
```

reduce

```
<"cat", 2>
<"dog", 1>
<"fish", 1>
```

word count lists

Status: `alpha` (at best!)

Work in progress:

- High-level language support (working on Rust runtime)
- `libd` C standard library
- MapReduce, web server, key-value store …

UNIVERSITY OF CAMBRIDGE

Malte Schwarzkopf

@ms705

in collaboration with

Matthew Grosvenor
Ionel Gog
Andrew Scull
Matthew Huxtable
Gustaf Helgesson
Steven Hand

***DIOS** is a **Cambridge Systems at Scale** project:*

http://www.cl.cam.ac.uk/netos/camsas/

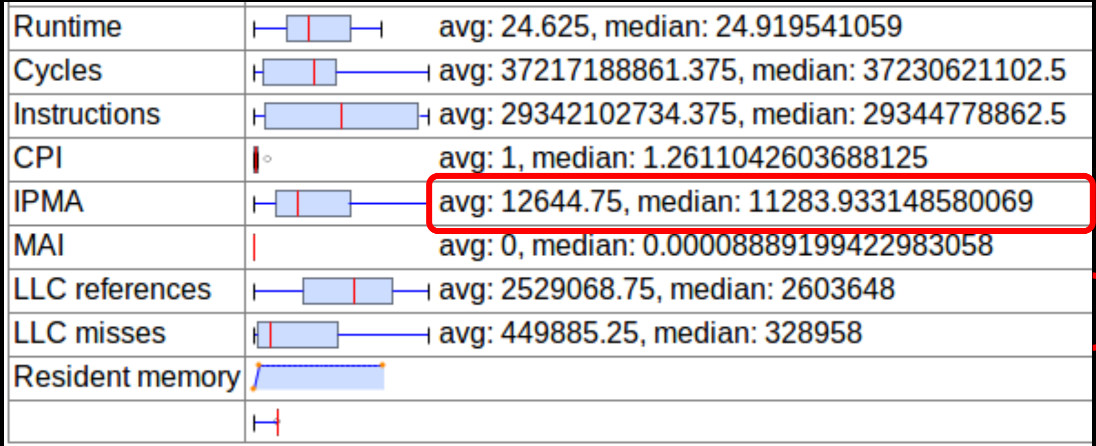
Gratuitous Docker slide :)

- DIOS is **Docker-compliant**!
  - isolate containers by restricting name resolution
  - but DIOS objects can also be shared by containers
  - Firmament scheduler can manage containers

- Benefits of DIOS + Docker
  - data-flow tracking + IFC **across** containers
  - can allow legacy syscalls within containers, but only DIOS syscalls on the host (“hypervisor mode”)

Task monitoring

Pi approximation (CPU-bound)

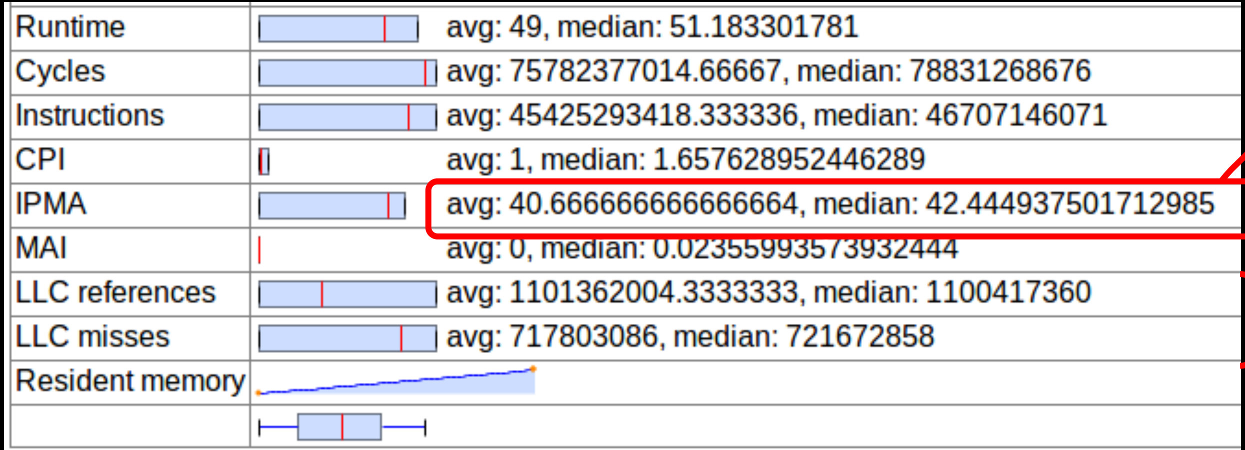


Runtime	avg: 24.625, median: 24.919541059
Cycles	avg: 37217188861.375, median: 37230621102.5
Instructions	avg: 29342102734.375, median: 29344778862.5
CPI	avg: 1, median: 1.2611042603688125
IPMA	avg: 12644.75, median: 11283.933148580069
MAI	avg: 0, median: 0.00008889199422983058
LLC references	avg: 2529068.75, median: 2603648
LLC misses	avg: 449885.25, median: 328958
Resident memory	

~12,000 instr. per mem access

12.6% miss

Matrix multiplication (memory-bound)



Runtime	avg: 49, median: 51.183301781
Cycles	avg: 75782377014.66667, median: 78831268676
Instructions	avg: 45425293418.333336, median: 46707146071
CPI	avg: 1, median: 1.657628952446289
IPMA	avg: 40.666666666666664, median: 42.444937501712985
MAI	avg: 0, median: 0.02355993573932444
LLC references	avg: 1101362004.3333333, median: 1100417360
LLC misses	avg: 717803086, median: 721672858
Resident memory	

~40 instr. per mem access

65.6% miss

Concept slides

Bullet points follow!

Why?

- Vertical integration of abstractions
  - enables optimisations, e.g. co-scheduling, locality
- Security, auditing, IFC
  - restrict and monitor data-flow
  - no way to bypass
- Because we can :)

How?

- Narrow syscall API: 11 syscalls
    - co-exist with POSIX, or replace
- Distributed object abstraction
    - object ~= “blob of bytes, stream of bytes or task”
- Security: distributed capabilities
    - Names: resolvable identifiers
    - References: FD-like handles with context info

Status?

- Prototype: Linux kernel extension
  - Tiny kernel patch (~500 LoC)
  - Two kernel modules
    - Adaptation layer: GPL
    - DIOS core: BSD
- HLL: Rust runtime port in progress

Demo!

- Simple streaming MapReduce
  - WordCount