Parallel computing, data and storage

“Grids, Clouds and distributed filesystems”

Mário David - LIP Lisbon

david@lip.pt

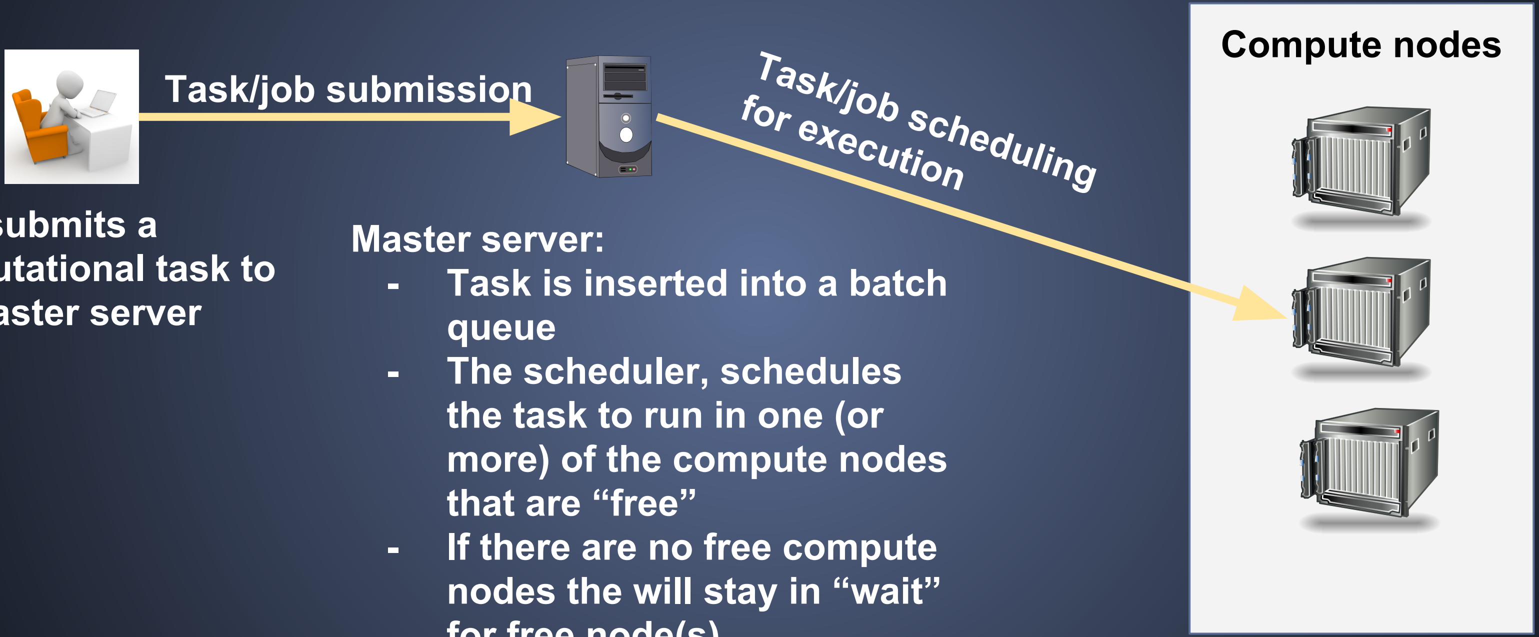
Overview

- ➢ Batch clusters
- ➢ Grid computing
- ➢ Cloud computing
  - ○ Infrastructure as a Service (IaaS)
  - ○ Platform as a Service (PaaS)
  - ○ Software as a Service (SaaS)
- ➢ Distributed data and storage
  - ○ Objects, Blocks and Filesystems (POSIX)
  - ○ Parallel filesystems (Lustre filesystem case)
  - ○ Object storage (Ceph case)

Batch clusters: Introduction

- Set of compute nodes connected through a LAN
- Execute computational tasks
- Orchestrated by a master server:
  - Scheduler
  - Batch (queue) system
- Compute nodes in general: homogeneous hardware and operating system (OS):
  - Different hardware and OS can be grouped into different partitions (batch queues)
- Input and Output data for the computational tasks are served through a shared/distributed filesystem

Batch clusters: usage (simplified view)

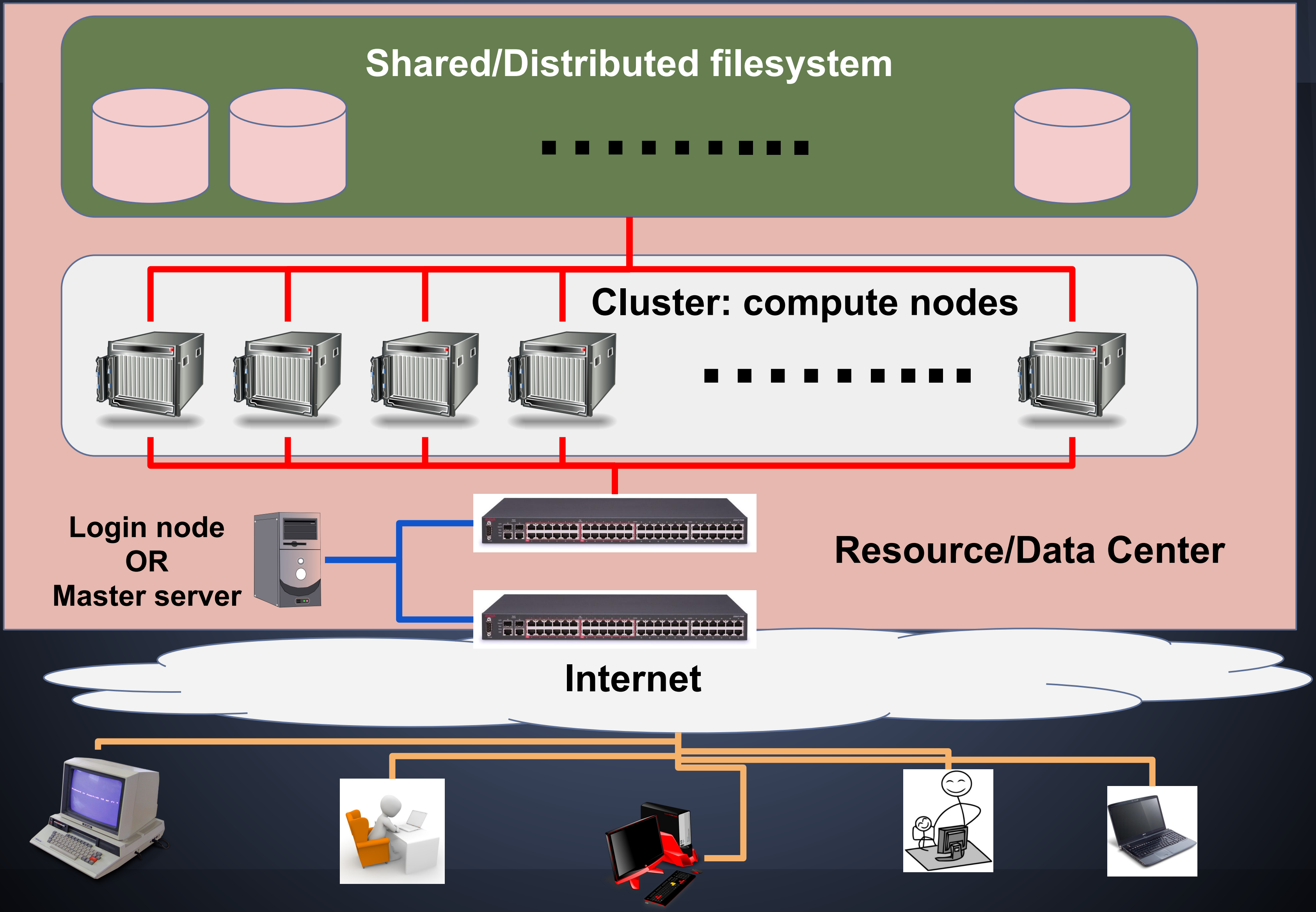


User submits a computational task to the master server

Master server:

- Task is inserted into a batch queue
- The scheduler, schedules the task to run in one (or more) of the compute nodes that are “free”
- If there are no free compute nodes the will stay in “wait” for free node(s)

Batch clusters: Typical architecture



Shared/Distributed filesystem

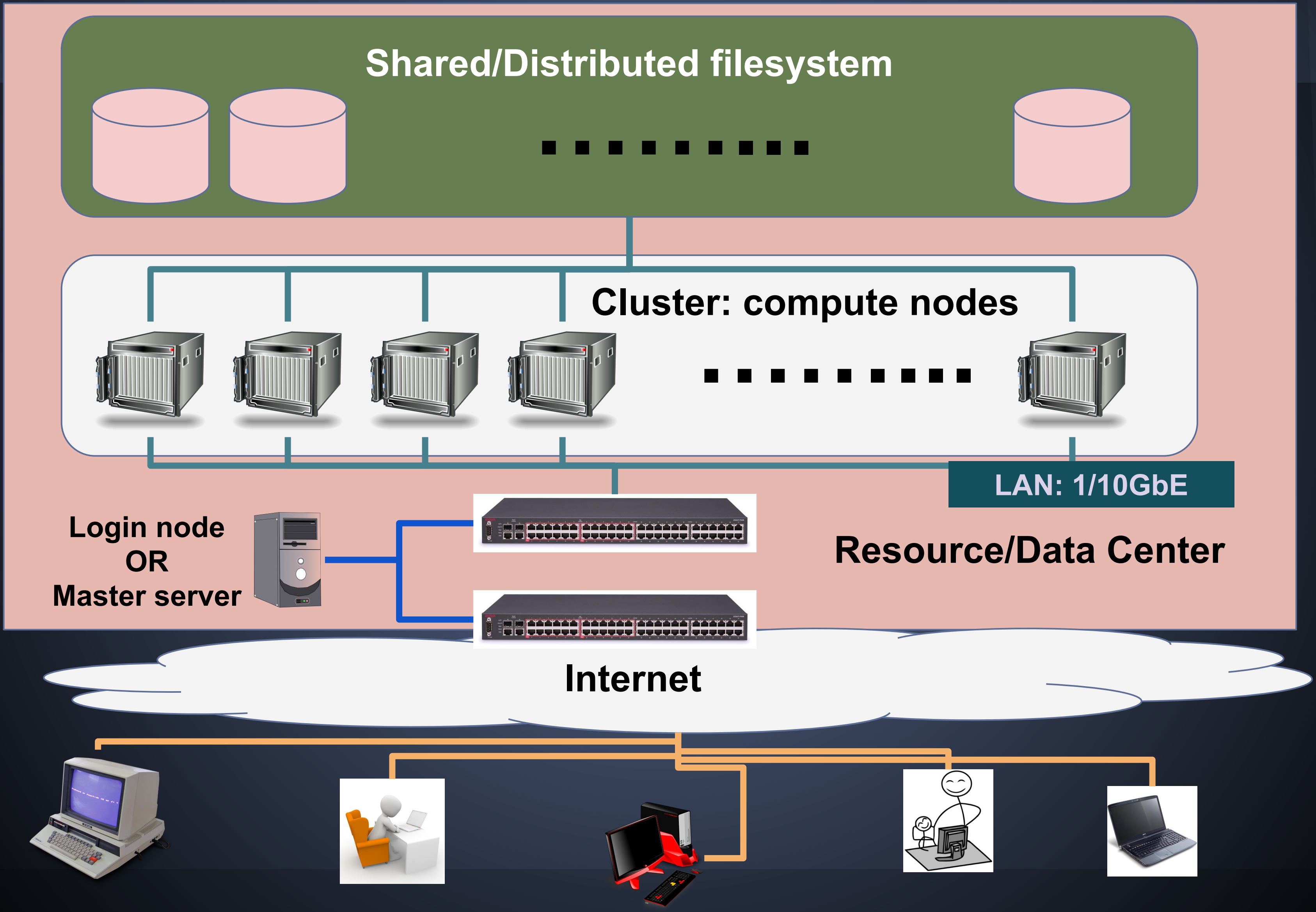
Cluster: compute nodes

Login node
OR
Master server

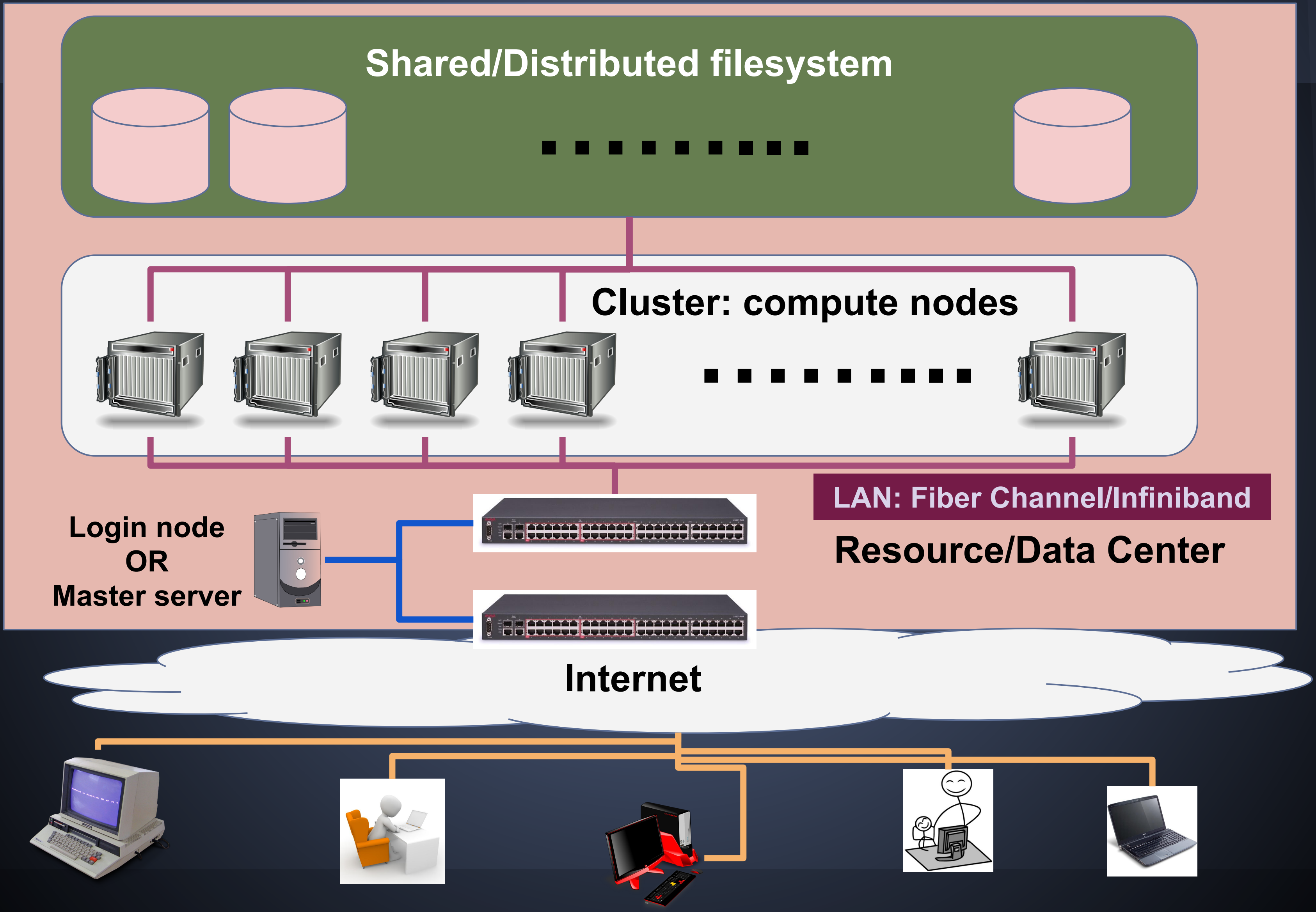
Resource/Data Center

Internet

Batch clusters: High Throughput Computing



Batch clusters: High Performance Computing



Batch clusters: Types - HTC

➢ High Throughput Computing (HTC):

- ○ Computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks.
- ○ Adequate both for data intensive (I/O bound) and compute intensive (CPU bound) applications
- ○ Adequate for **serial** applications.
- ○ **Embarrassingly** parallel applications.

For example, processing/analysis of independent events ↦

IF:

you have 1000 events, and 100 CPUs

THEN:

distributing the processing of 10 events/CPU would yield a gain of 100 over a serial processing of all events in a single CPU

Batch clusters: Types - HPC

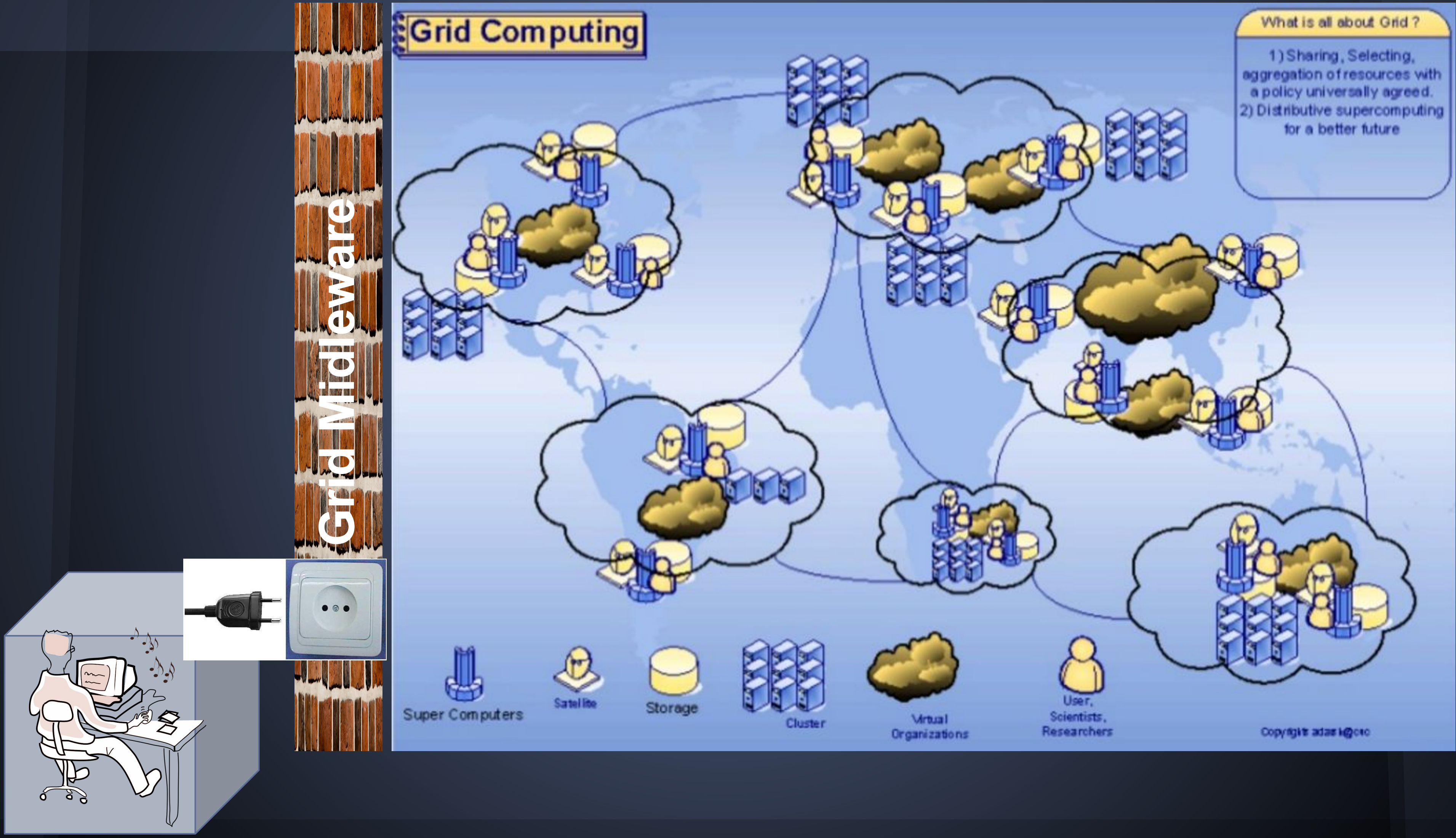
- High Performance Computing (HPC):
  - Focus on tightly coupled parallel jobs and fast job execution.
  - Main difference in HW with respect to HTC, LAN is "**low latency**" such as Infiniband.
  - Adequate for compute intensive applications (CPU bound)
  - Adequate for parallel applications:
- Very common making use of **parallel programing**, such as using some implementation of MPI (Message Passing Interface) standard.
- Processes/Tasks need to communicate (send/receive messages) from other Processes/Tasks.

1/10 GbE over TCP/IP	**Latency ~ 10-100µs**
Infiniband 10 - 100 Gb/s	**Latency ≲ 1µs**

Grid computing: Introduction

- Federation of clusters that are **geographically** distributed:
  - Each cluster is independent from the others: increase in heterogeneity with respect to a single cluster.
  - It has different administrative domains and policies
  - BUT, the users/researchers want a single way of “interaction” with all resources/clusters of the Grid:
    - Common APIs, CLIs
    - Common/single Authentication and Authorization system

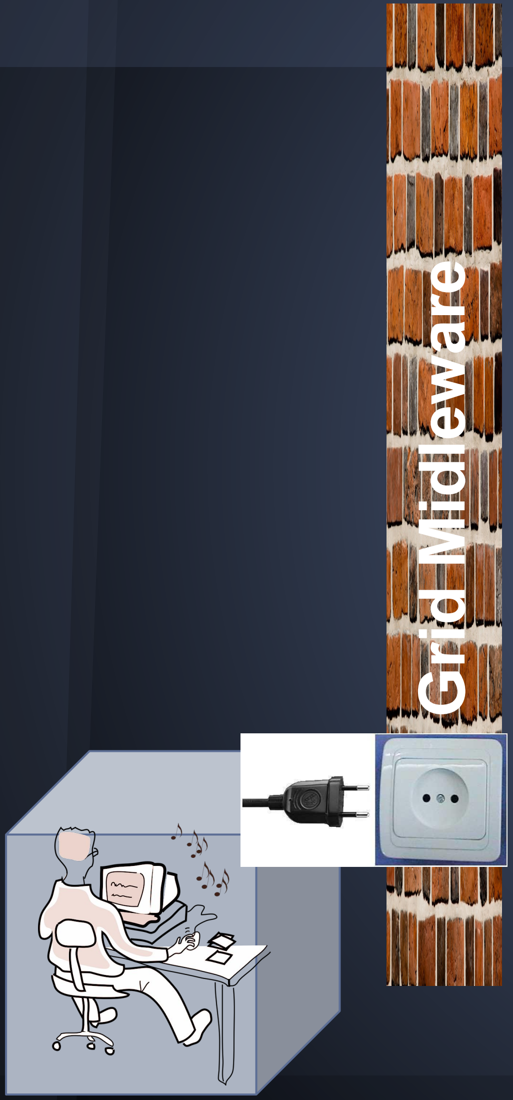
Grid computing: Architecture



What is all about Grid ?

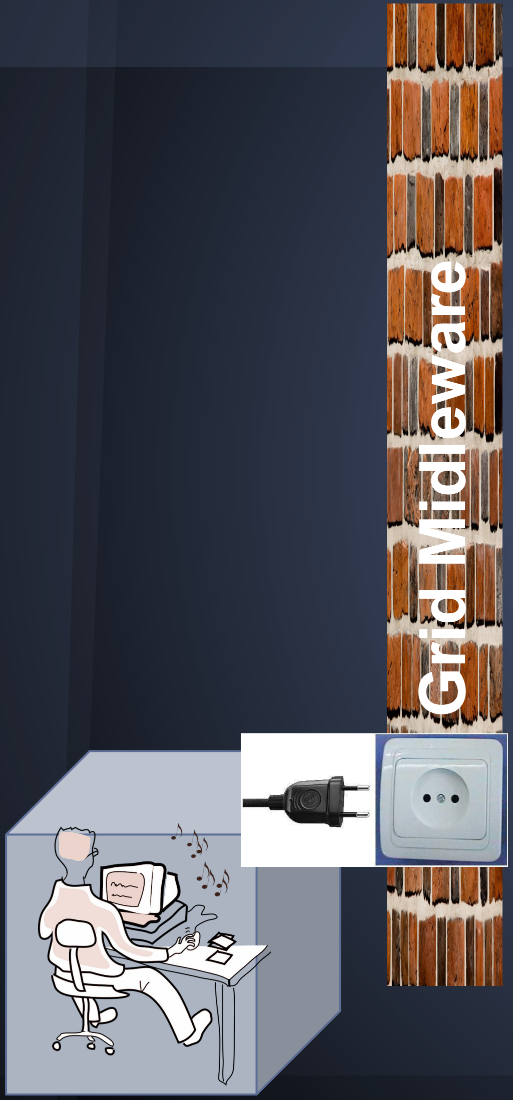
1) Sharing, Selecting, aggregation of resources with a policy universally agreed.
2) Distributive supercomputing for a better future

Grid computing: Grid middleware I



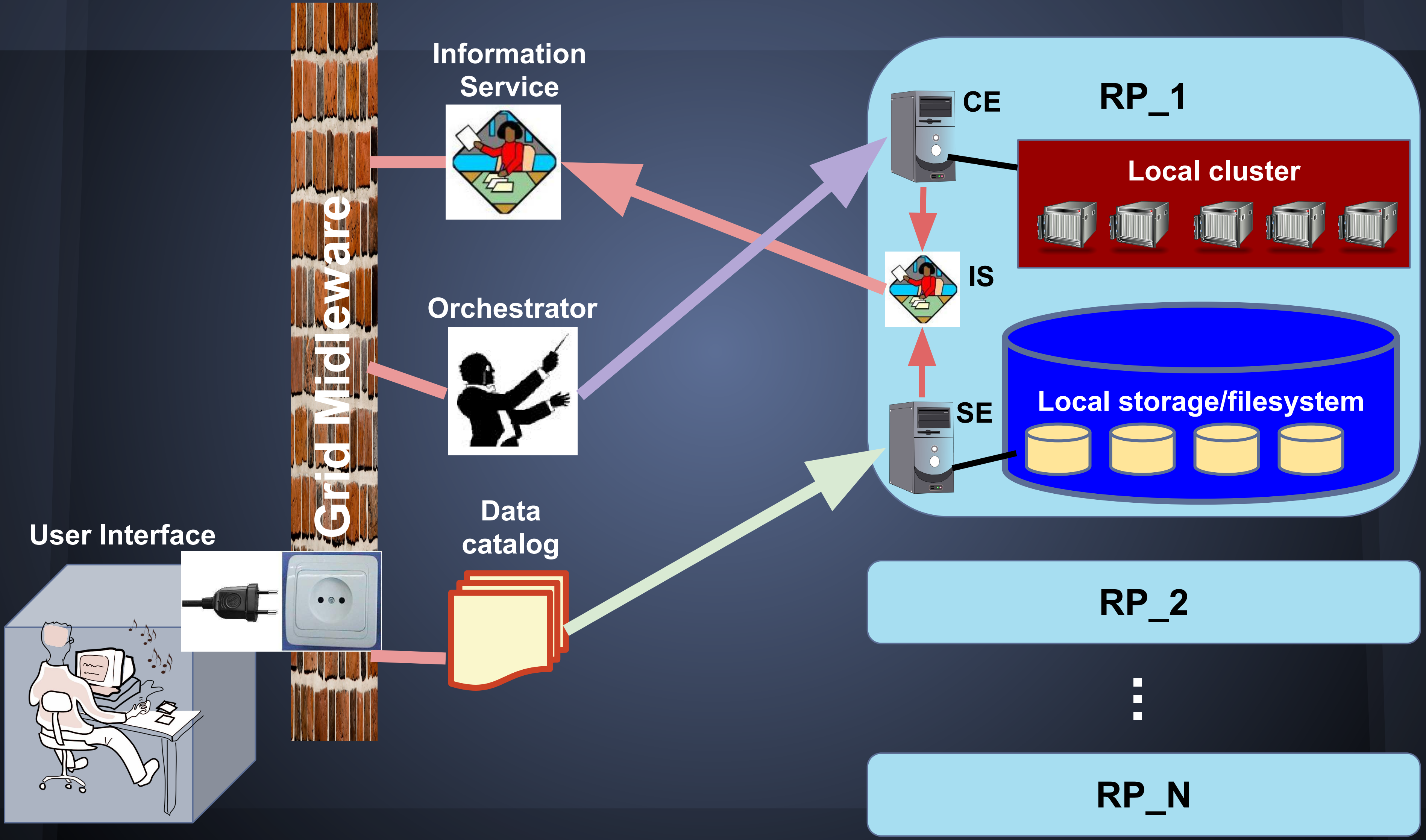
- **Common Authentication mechanism**:
  - X.509 certificates
  - Certification Authorities
- **Common Authorization mechanism**:
  - Users grouped by Virtual Organizations (VOs)
  - Resource providers authorize VOs to access and use their resources (computing and storage)
- **Compute Element (CE)**:
  - Frontend service exposing the local computing cluster to users through a common API/CLI
- **Storage Element (SE)**:
  - Frontend service exposing the local storage system to users through a common API/CLI

Grid computing: Grid middleware II



- **Information services (IS)**:
  - Gather and publish information about the resources
- **Global data catalogs**:
  - Global view of data/files that are spread through several Storage Elements
  - Provide information about the physical location of the files
- **Orchestrator service/Resource Broker**:
  - Schedules compute tasks to Compute Elements based on the Information service and authorization policies (supported VOs)
- **File Transfer Service**:
  - Management of data movement/transfer between resource providers

Grid computing: Grid middleware III



Cloud Computing I

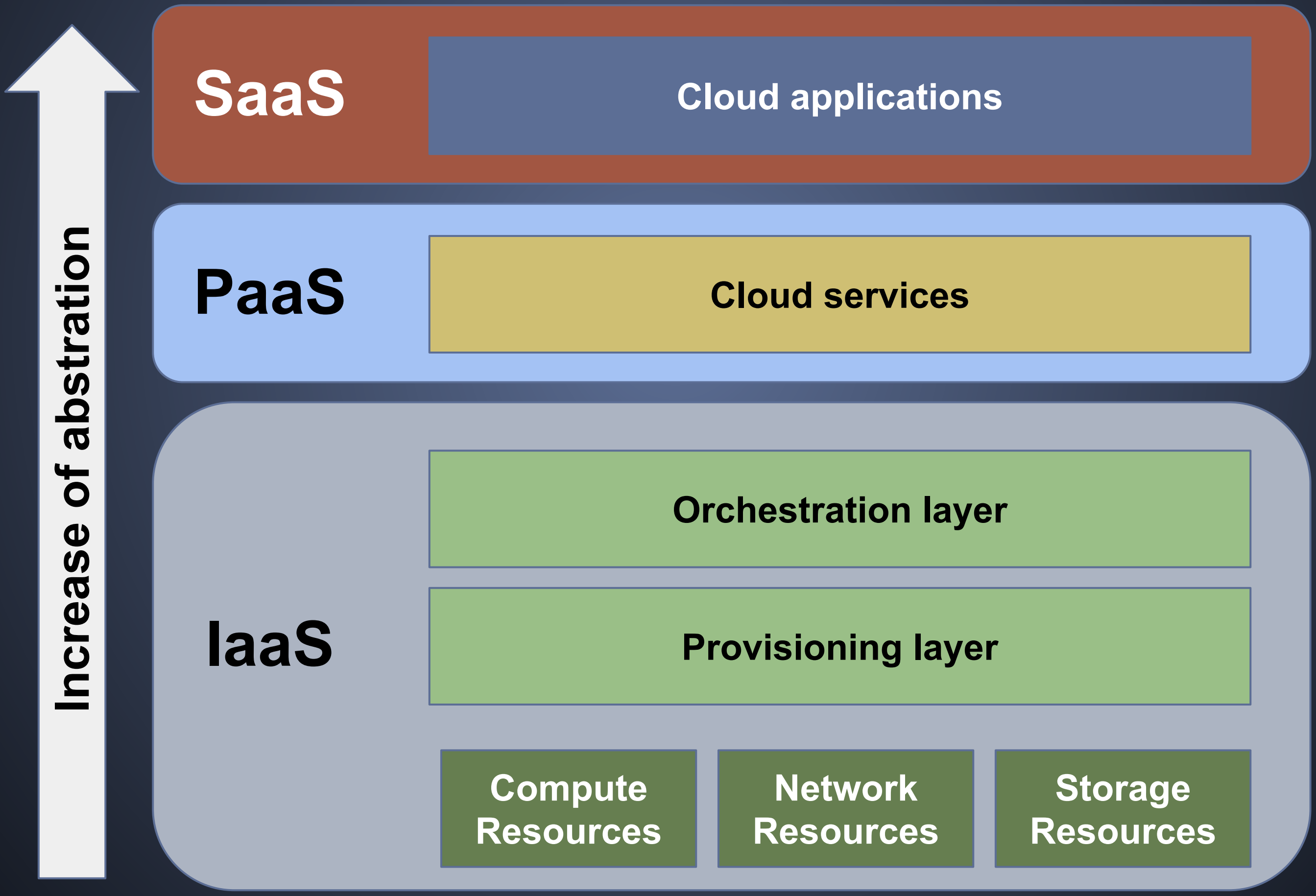
“(...) a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

- ➢ **On-demand Self Service**
  - ○ The users are able to provision and manage their own computing environment according to their needs, without further intervention from the provider.
- ➢ **Elastic Provisioning and Scalability**
  - ○ The cloud model tries to deliver easily and rapidly the resources to the users, in a short-deadline basis.
  - ○ Users are able to scale in and out their infrastructure so as to satisfy the real demand, not only by increasing their capacity, but also by shrinking it whenever it is not needed.

Cloud Computing II

- **Metered Usage and Billing**
  - Resources accounted by their usage, rather than following a subscription mode.
- **Multi-tenancy and Dynamic Resource Pooling**
  - Ability for a software or provider to deliver a service to several parties simultaneously.
  - Services owned by several users are being co-located in the same resources.
  - Tenants resources are isolated from each other.
  - Each tenant manages creates and manages it’s own compute, storage and local network.
  - Important for organizations supporting multiple projects/groups, and service providers supporting multiple users.

Cloud Computing: X as a Service models



Cloud Computing: Classification I

- **Infrastructure as a Service (IaaS):**
  - Lowest level of abstraction
  - Considered as the foundation of the cloud model.
  - Offers its infrastructure resources: computing, networking and storage.
  - Users can deploy its own OS, software, network configuration, etc.
  - Abstracts the underlying fabric (physical resources) into a uniform resource layer:
    - **Virtualization** or encapsulation the raw resources.
    - Users get **transparent access** to this layer as if they were using the bare metal resources.
    - Able to deploy any infrastructure on top of it without the extra burden of directly managing the different physical resources.

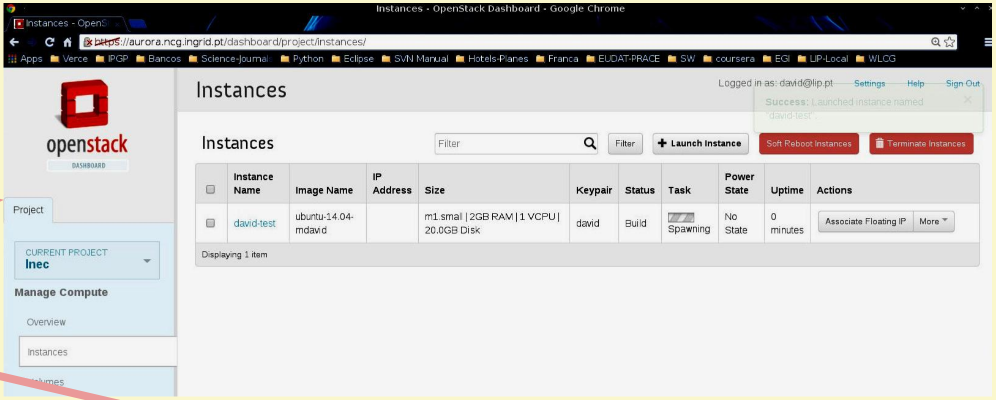
IaaS: Openstack CFM

Instantiate a VM machine

Web - GUI/Dashboard

CLI

Openstack (Cloud Management Framework)



Instances

Logged in as: david@lip.pt

Success: Launched instance named "david-test".

Instances

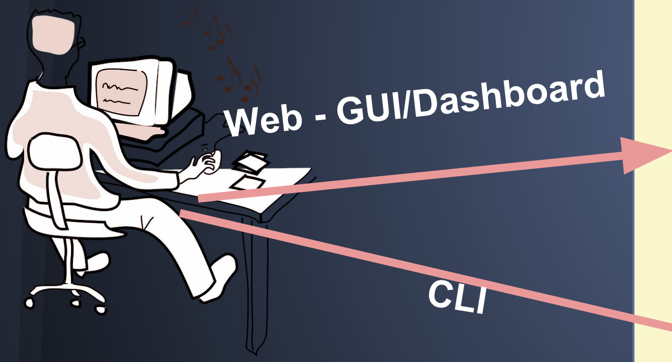
☐	Instance Name	Image Name	IP Address	Size	Keypair	Status	Task	Power State	Uptime	Actions
☐	david-test	ubuntu-14.04-mdavid		m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	david	Build	Spawning	No State	0 minutes	Associate Floating IP More

Displaying 1 item

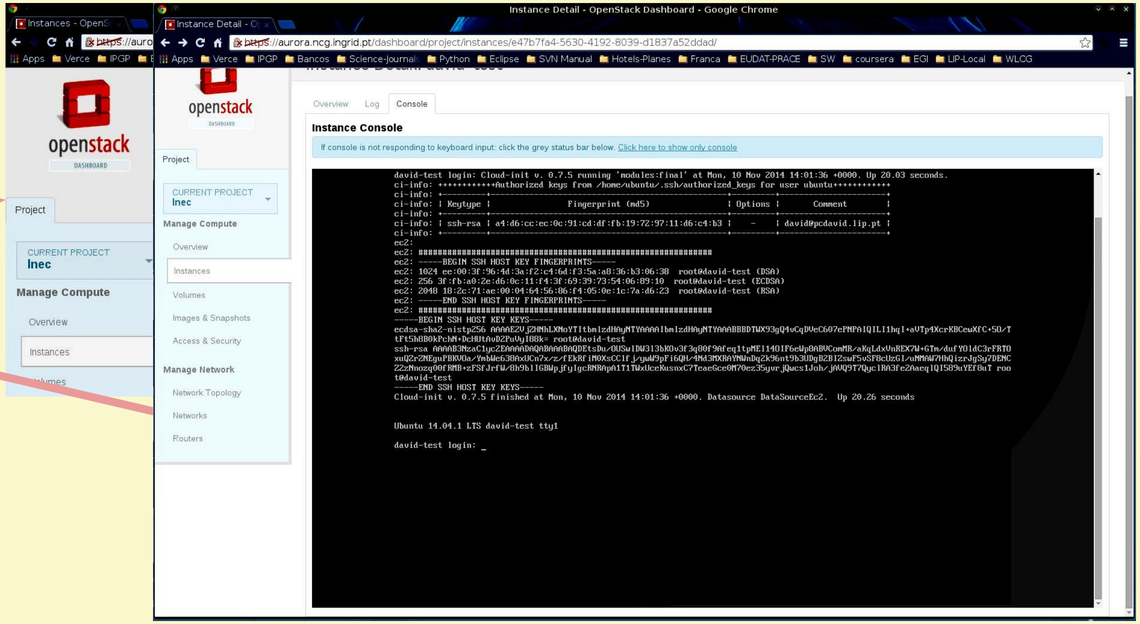
```
$ nova boot ...
```

IaaS: Openstack CFM

Instantiate a VM machine



Openstack (Cloud Management Framework)



Instance Console

If console is not responding to keyboard input: click the grey status bar below. Click here to show only console

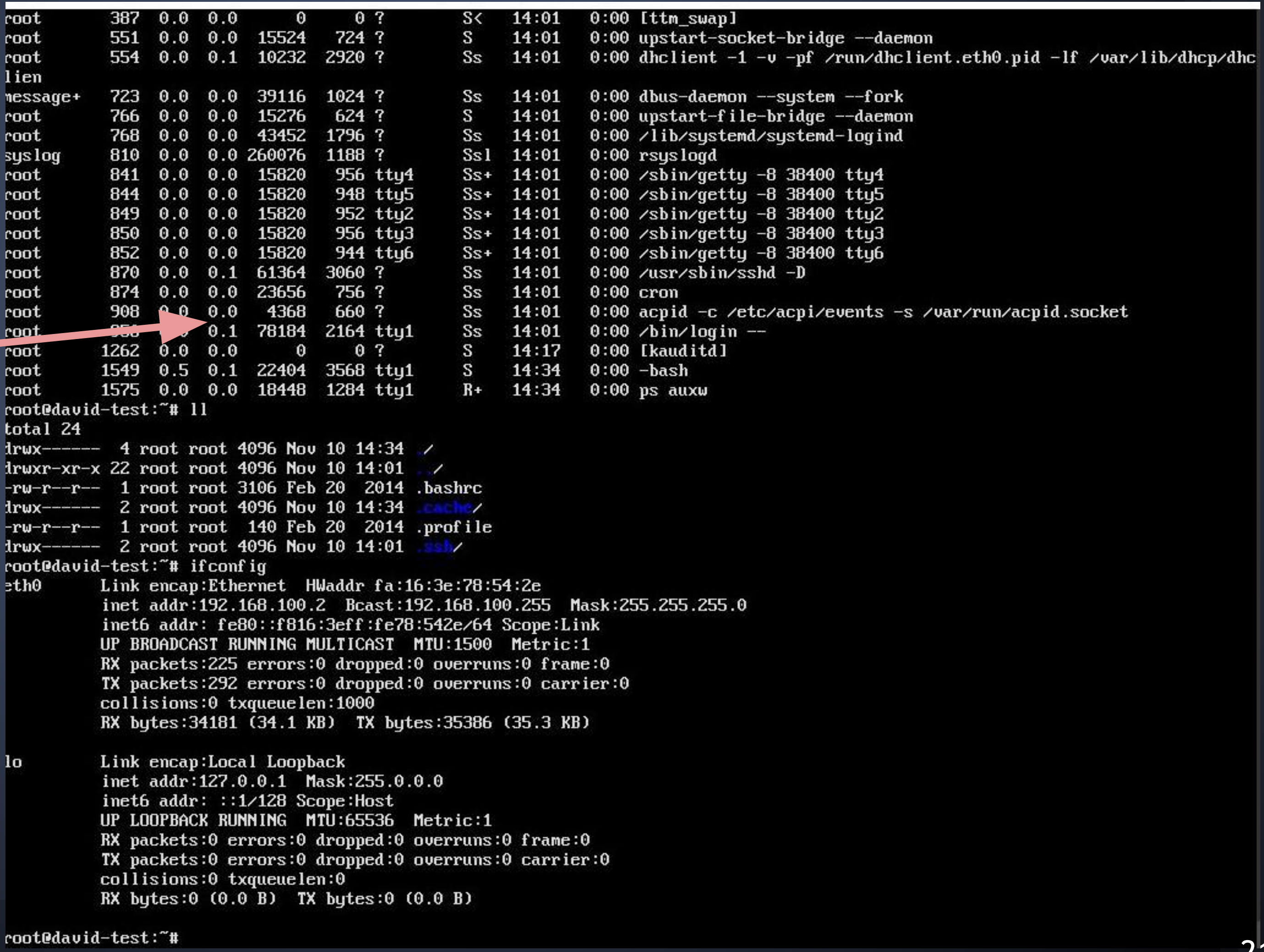
```
david-test login: Cloud-init v. 0.7.5 running 'modules:final' at Mon, 10 Nov 2014 14:01:36 +0000. Up 20.03 seconds.
ci-info: ++++++++++++Authorized keys from /home/ubuntu/.ssh/authorized_keys for user ubuntu++++++++++++
ci-info: +---------+-------------------------------------------------+---------+-----------------------+
ci-info: ¦ Keytype ¦                Fingerprint (md5)                ¦ Options ¦        Comment        ¦
ci-info: +---------+-------------------------------------------------+---------+-----------------------+
ci-info: ¦ ssh-rsa ¦ a4:d6:cc:ec:0c:91:cd:df:fb:19:72:97:11:d6:c4:b3 ¦    -    ¦ david@pcdavid.lip.pt ¦
ci-info: +---------+-------------------------------------------------+---------+-----------------------+
ec2:
ec2: #############################################################
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 1024 ee:00:3f:96:4d:3a:f2:c4:6d:f3:5a:a8:36:b3:06:38  root@david-test (DSA)
ec2: 256 3f:fb:a0:2e:d6:0c:11:f4:3f:69:39:73:54:06:89:10  root@david-test (ECDSA)
ec2: 2048 18:2c:71:ae:00:04:64:56:86:f4:05:0e:1c:7a:d6:23  root@david-test (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #############################################################
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDTWX93gQ4vCqDVeC607ePNPAIQILI1hqI+aVTp4XcrKBCewXfC+5O/T
tFt5h8B0kPchN+DcHUtAvD2PuVyI88k= root@david-test
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDEtsDu/OUSwlDW3l3bKOv3f3q80f9Afeq1tpMEl14OIF6eWp8ABVComMR/aKqLdxVnREX7W+GTm/dufYOldC3rFRTO
xuQZrZNEguPBKVOa/YmbWe638AxUCn7x/z/fEkRfiM0XsCClfj/ywW9pFi6QH/4Nd3MXRAYNWnDq2k96nt9b3UDgBZBIZswF5vSF8cUzGI/uNMAW7HhQizrJgSy7DENC
Z2zNnozq00fRMB+zFSfJrfW/8h9blIGBWpjfyIgcRNRApA1T1TWxUceKusnxC7TeaeGce0M70ez35yvrjQwcs1Joh/jAVQ9T7QyclRA3feZAaeqlQI589uYEf8uT roo
t@david-test
-----END SSH HOST KEY KEYS-----
Cloud-init v. 0.7.5 finished at Mon, 10 Nov 2014 14:01:36 +0000. Datasource DataSourceEc2.  Up 20.26 seconds


Ubuntu 14.04.1 LTS david-test tty1

david-test login: _
```

IaaS: Openstack CFM

Login into the VM



```
root         387  0.0  0.0      0     0 ?        S<   14:01   0:00 [ttm_swap]
root         551  0.0  0.0  15524   724 ?        S    14:01   0:00 upstart-socket-bridge --daemon
root         554  0.0  0.1  10232  2920 ?        Ss   14:01   0:00 dhclient -1 -v -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp/dhc
lien
message+     723  0.0  0.0  39116  1024 ?        Ss   14:01   0:00 dbus-daemon --system --fork
root         766  0.0  0.0  15276   624 ?        S    14:01   0:00 upstart-file-bridge --daemon
root         768  0.0  0.0  43452  1796 ?        Ss   14:01   0:00 /lib/systemd/systemd-logind
syslog       810  0.0  0.0 260076  1188 ?        Ssl  14:01   0:00 rsyslogd
root         841  0.0  0.0  15820   956 tty4     Ss+  14:01   0:00 /sbin/getty -8 38400 tty4
root         844  0.0  0.0  15820   948 tty5     Ss+  14:01   0:00 /sbin/getty -8 38400 tty5
root         849  0.0  0.0  15820   952 tty2     Ss+  14:01   0:00 /sbin/getty -8 38400 tty2
root         850  0.0  0.0  15820   956 tty3     Ss+  14:01   0:00 /sbin/getty -8 38400 tty3
root         852  0.0  0.0  15820   944 tty6     Ss+  14:01   0:00 /sbin/getty -8 38400 tty6
root         870  0.0  0.1  61364  3060 ?        Ss   14:01   0:00 /usr/sbin/sshd -D
root         874  0.0  0.0  23656   756 ?        Ss   14:01   0:00 cron
root         908  0.0  0.0   4368   660 ?        Ss   14:01   0:00 acpid -c /etc/acpi/events -s /var/run/acpid.socket
root         950  0.0  0.1  78184  2164 tty1     Ss   14:01   0:00 /bin/login --
root        1262  0.0  0.0      0     0 ?        S    14:17   0:00 [kauditd]
root        1549  0.5  0.1  22404  3568 tty1     S    14:34   0:00 -bash
root        1575  0.0  0.0  18448  1284 tty1     R+   14:34   0:00 ps auxw
root@david-test:~# ll
total 24
drwx------  4 root root 4096 Nov 10 14:34 ./
drwxr-xr-x 22 root root 4096 Nov 10 14:01 ../
-rw-r--r--  1 root root 3106 Feb 20  2014 .bashrc
drwx------  2 root root 4096 Nov 10 14:34 .cache/
-rw-r--r--  1 root root  140 Feb 20  2014 .profile
drwx------  2 root root 4096 Nov 10 14:01 .ssh/
root@david-test:~# ifconfig
eth0      Link encap:Ethernet  HWaddr fa:16:3e:78:54:2e
          inet addr:192.168.100.2  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe78:542e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:225 errors:0 dropped:0 overruns:0 frame:0
          TX packets:292 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34181 (34.1 KB)  TX bytes:35386 (35.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@david-test:~#
```

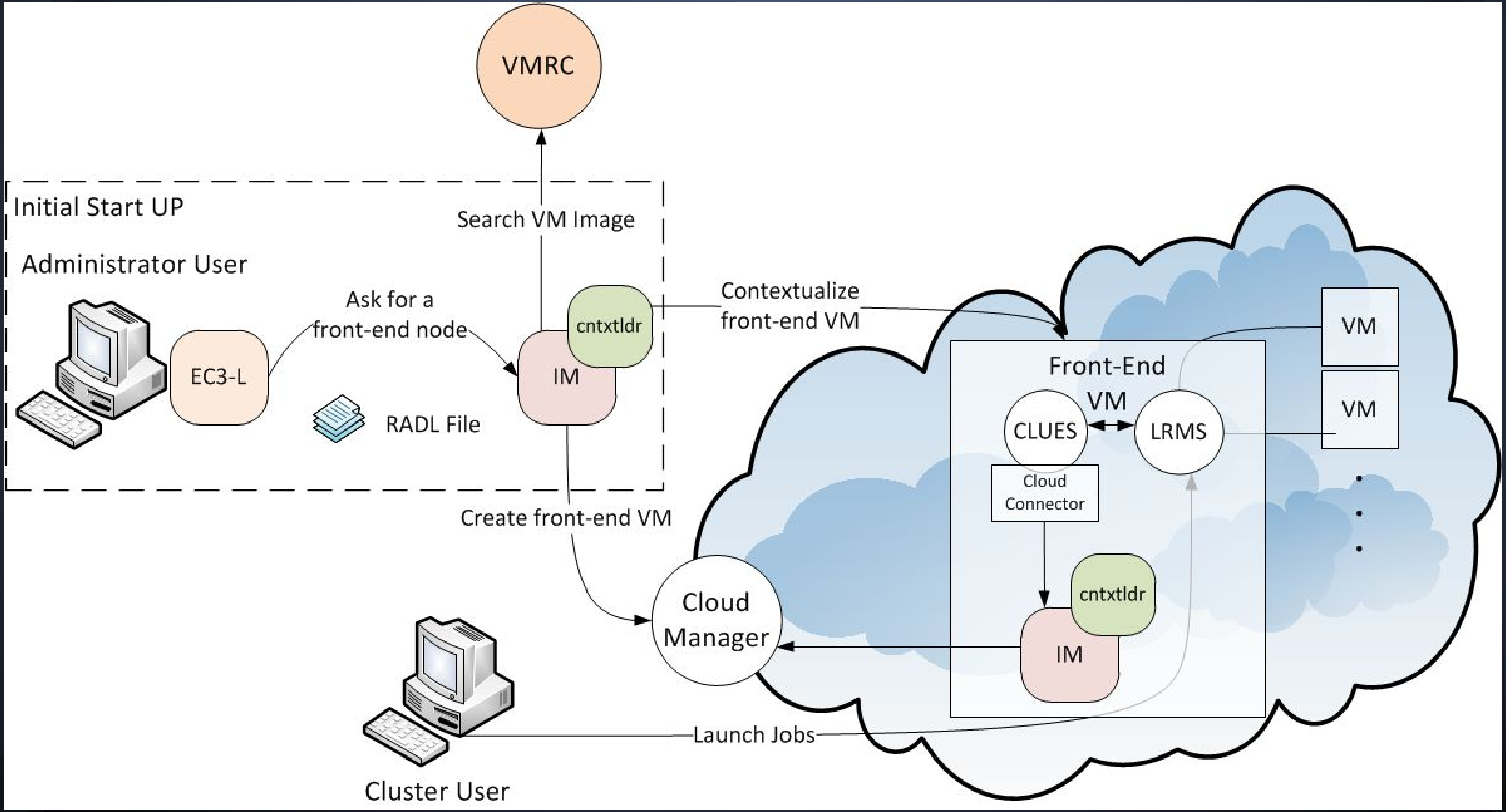
Cloud Computing: Classification II

- **Platform as a Service (PaaS):**
  - Second step in the abstraction level
  - Resources coming from an IaaS are **composed** so that they can be consumed by the users without requiring the management or the knowledge of the underlying infrastructure.
  - Offer an environment where a user **can deploy and manage its applications** using the libraries, software, tools, APIs, etc. supported by the provider
  - Makes possible to deliver **complex applications and services** involving different components to end-users:
    - No need of direct managing of the machine configurations and deployment
    - Allows to define the requirements of those applications, so that the platform layer is **able to orchestrate the resources**.

PaaS: Infrastructure Manager (IM)

From Univ. Valencia: http://www.grycap.upv.es/im/index.php

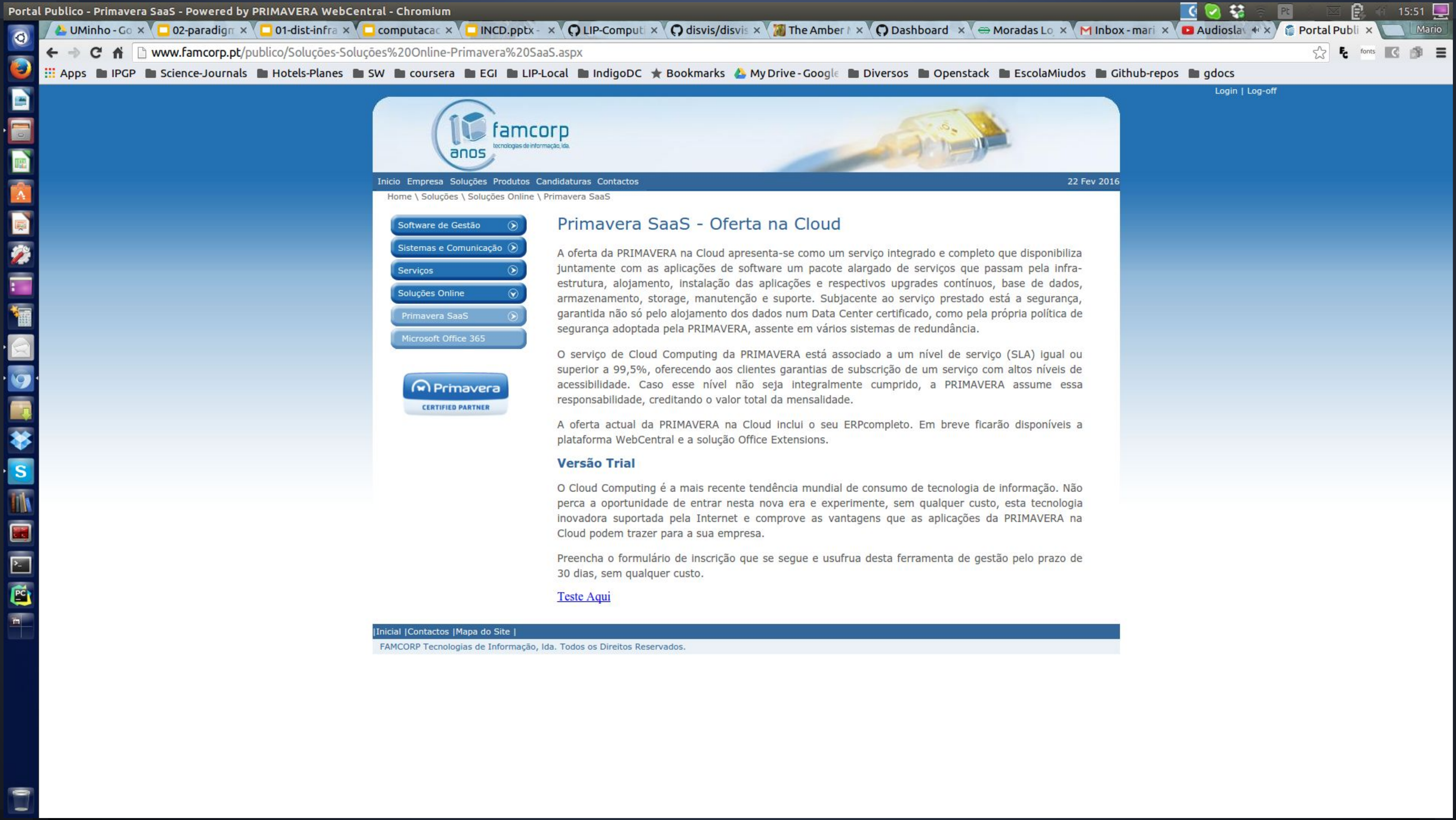
EC3 (Elastic Cloud Computing Cluster)



Cloud Computing: Classification III

- **Software as a Service (SaaS):**
  - The highest level of abstraction.
  - Comprises the applications that are running on top of a cloud infrastructure.
  - Access to SaaS applications are normally addressed using ad-hoc thin clients executed inside web browsers or applications that are executed on tablets or smartphones, directly addressing the end user.
  - Change in paradigm: FROM buy software TO buy service:
    - Delegate the software management (to service provider) and focus on the use of the service/software
    - One example: the Primavera ERP http://www.famcorp.pt/publico/Solu%C3%A7%C3%B5es-Solu%C3%A7%C3%B5es%20Online-Primavera%20SaaS.aspx

Cloud Computing: Classification III



Primavera SaaS - Oferta na Cloud

A oferta da PRIMAVERA na Cloud apresenta-se como um serviço integrado e completo que disponibiliza juntamente com as aplicações de software um pacote alargado de serviços que passam pela infra-estrutura, alojamento, instalação das aplicações e respectivos upgrades contínuos, base de dados, armazenamento, storage, manutenção e suporte. Subjacente ao serviço prestado está a segurança, garantida não só pelo alojamento dos dados num Data Center certificado, como pela própria política de segurança adoptada pela PRIMAVERA, assente em vários sistemas de redundância.

O serviço de Cloud Computing da PRIMAVERA está associado a um nível de serviço (SLA) igual ou superior a 99,5%, oferecendo aos clientes garantias de subscrição de um serviço com altos níveis de acessibilidade. Caso esse nível não seja integralmente cumprido, a PRIMAVERA assume essa responsabilidade, creditando o valor total da mensalidade.

A oferta actual da PRIMAVERA na Cloud inclui o seu ERPcompleto. Em breve ficarão disponíveis a plataforma WebCentral e a solução Office Extensions.

Versão Trial

O Cloud Computing é a mais recente tendência mundial de consumo de tecnologia de informação. Não perca a oportunidade de entrar nesta nova era e experimente, sem qualquer custo, esta tecnologia inovadora suportada pela Internet e comprove as vantagens que as aplicações da PRIMAVERA na Cloud podem trazer para a sua empresa.

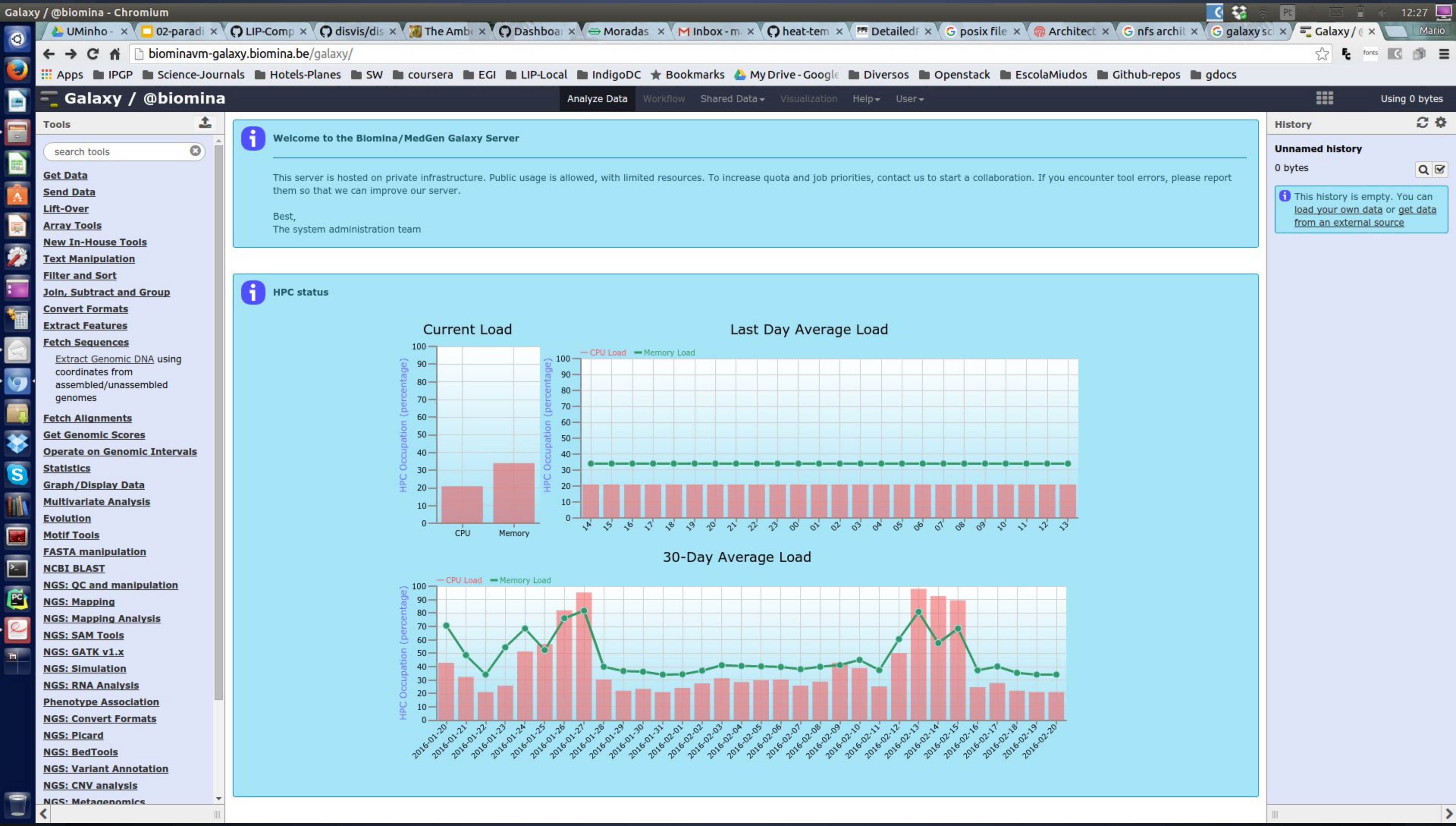
Preencha o formulário de inscrição que se segue e usufrua desta ferramenta de gestão pelo prazo de 30 dias, sem qualquer custo.

Teste Aqui

|Inicial |Contactos |Mapa do Site |

FAMCORP Tecnologias de Informação, lda. Todos os Direitos Reservados.

SaaS: Galaxy science portal



Galaxy / @biomina - Chromium
12:27
biominavm-galaxy.biomina.be/galaxy/
Apps IPGP Science-Journals Hotels-Planes SW coursera EGI LIP-Local IndigoDC Bookmarks My Drive - Google Diversos Openstack EscolaMiudos Github-repos gdocs
Galaxy / @biomina
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 0 bytes
Tools
search tools
History

Get Data
Send Data
Lift-Over
Array Tools
New In-House Tools
Text Manipulation
Filter and Sort
Join, Subtract and Group
Convert Formats
Extract Features
Fetch Sequences
Extract Genomic DNA using coordinates from assembled/unassembled genomes
Fetch Alignments
Get Genomic Scores
Operate on Genomic Intervals
Statistics
Graph/Display Data
Multivariate Analysis
Evolution
Motif Tools
FASTA manipulation
NCBI BLAST
NGS: QC and manipulation
NGS: Mapping
NGS: Mapping Analysis
NGS: SAM Tools
NGS: GATK v1.x
NGS: Simulation
NGS: RNA Analysis
Phenotype Association
NGS: Convert Formats
NGS: Picard
NGS: BedTools
NGS: Variant Annotation
NGS: CNV analysis
NGS: Metagenomics

Welcome to the Biomina/MedGen Galaxy Server

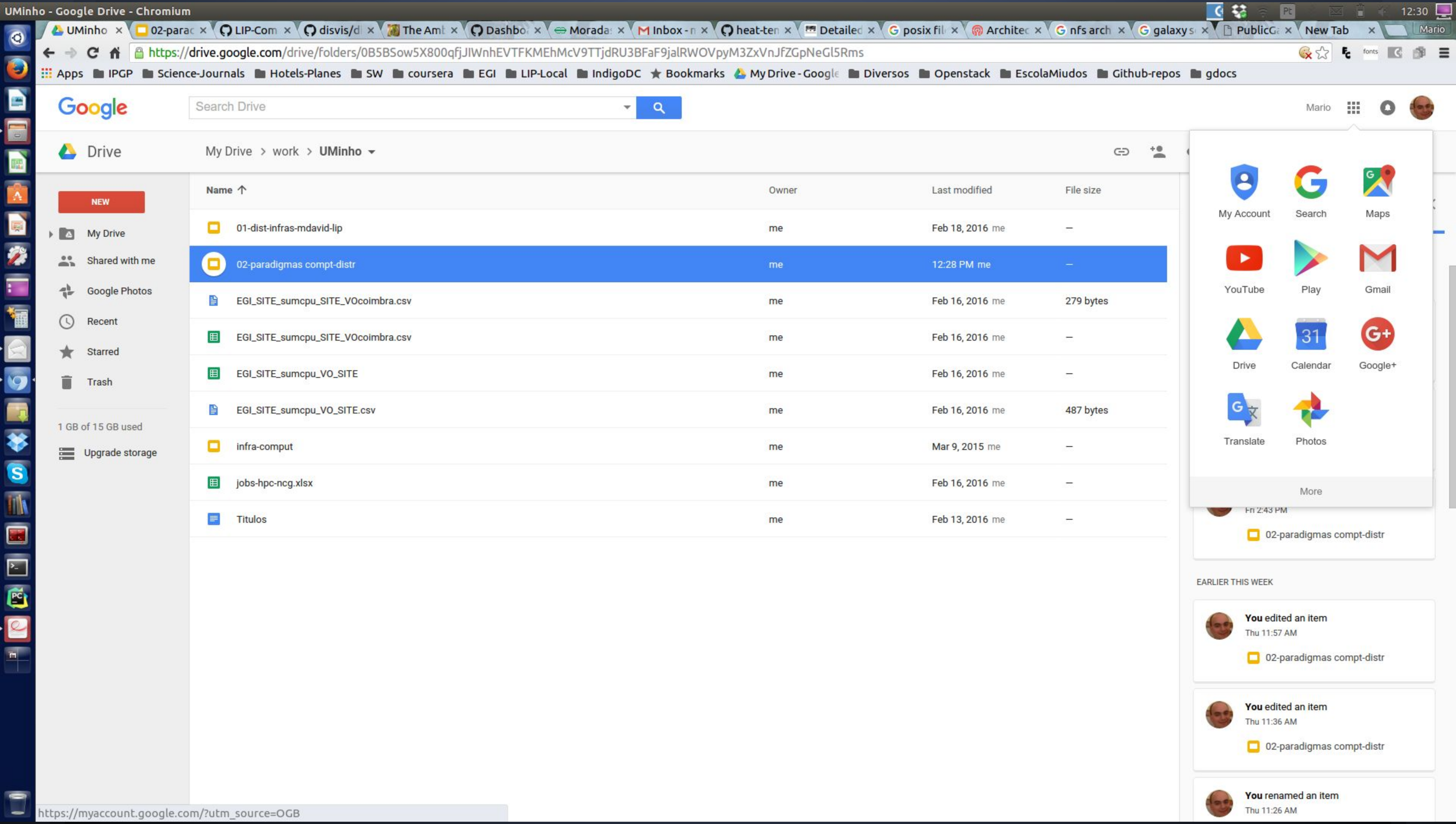
This server is hosted on private infrastructure. Public usage is allowed, with limited resources. To increase quota and job priorities, contact us to start a collaboration. If you encounter tool errors, please report them so that we can improve our server.

Best,
The system administration team

HPC status

Unnamed history
0 bytes
This history is empty. You can load your own data or get data from an external source

SaaS: Public provider



https://drive.google.com/drive/folders/0B5BSow5X800qfjJlWnhEVTFKMEhMcV9TTjdRU3BFaF9jalRWOVpyM3ZxVnJfZGpNeGl5Rms

My Drive > work > UMinho

Name ↑	Owner	Last modified	File size
01-dist-infras-mdavid-lip	me	Feb 18, 2016 me	–
02-paradigmas compt-distr	me	12:28 PM me	–
EGI_SITE_sumcpu_SITE_VOcoimbra.csv	me	Feb 16, 2016 me	279 bytes
EGI_SITE_sumcpu_SITE_VOcoimbra.csv	me	Feb 16, 2016 me	–
EGI_SITE_sumcpu_VO_SITE	me	Feb 16, 2016 me	–
EGI_SITE_sumcpu_VO_SITE.csv	me	Feb 16, 2016 me	487 bytes
infra-comput	me	Mar 9, 2015 me	–
jobs-hpc-ncg.xlsx	me	Feb 16, 2016 me	–
Titulos	me	Feb 13, 2016 me	–

Fri 2:43 PM

02-paradigmas compt-distr

EARLIER THIS WEEK

You edited an item
Thu 11:57 AM
02-paradigmas compt-distr

You edited an item
Thu 11:36 AM
02-paradigmas compt-distr

You renamed an item
Thu 11:26 AM

https://myaccount.google.com/?utm_source=OGB

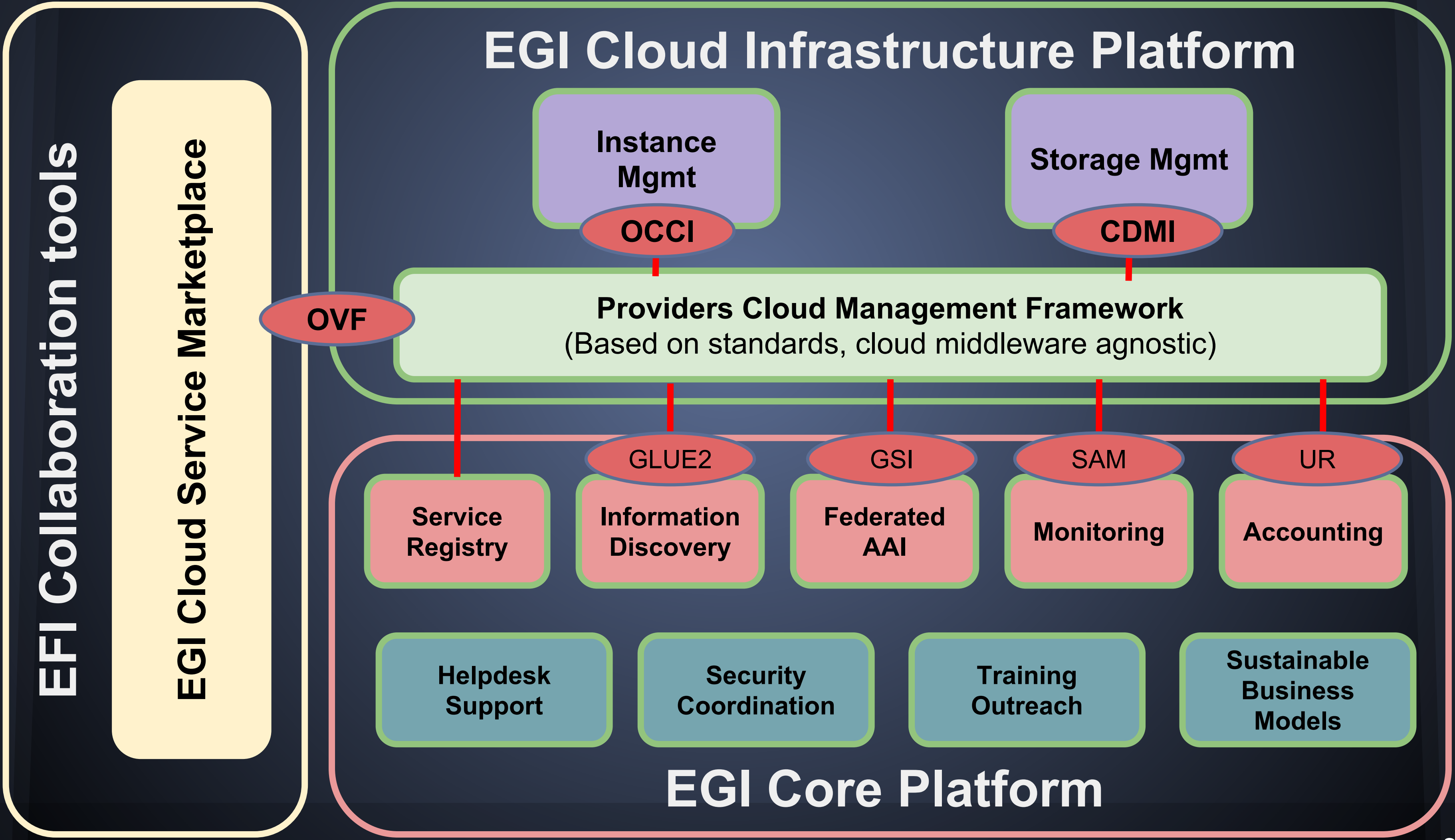
Grids/Clusters versus Clouds

Grids/Clusters	Clouds
Fixed environments: OS, applications	Flexible environments: users choose OS, applications, through virtualization (1)
Amount of resources are fixed apriori	Amount of resources are elastic: can increase or decrease according to users needs
Applications/tasks are executed during some fixed amount of time	Can run applications/tasks during a fixed amount of time, but can also run long term services such as web and scientific portals, databases, etc.
Applications are scheduled to batch queues	On demand "almost" real time provisioning of resources
Grid is a federation of clusters (resource providers)	Federation of clouds is still quite difficult and a strong hot topic. One such case is the EGI FedCloud infrastructure (2)

(1) It's also possible to deploy flexible environment in bare metal (physical) nodes

(2) EGI - European Grid Infrastructure: Provides a Grid <u>and</u> a Federated Cloud infrastructure at European level

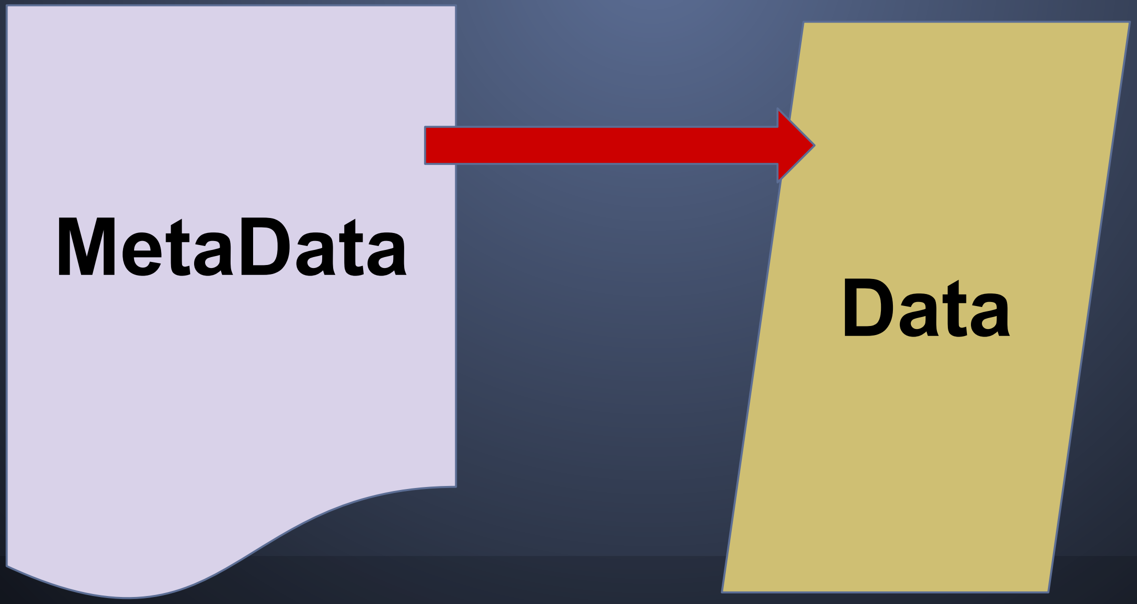
EGI Federated cloud: Architecture



Distributed Data and Storage

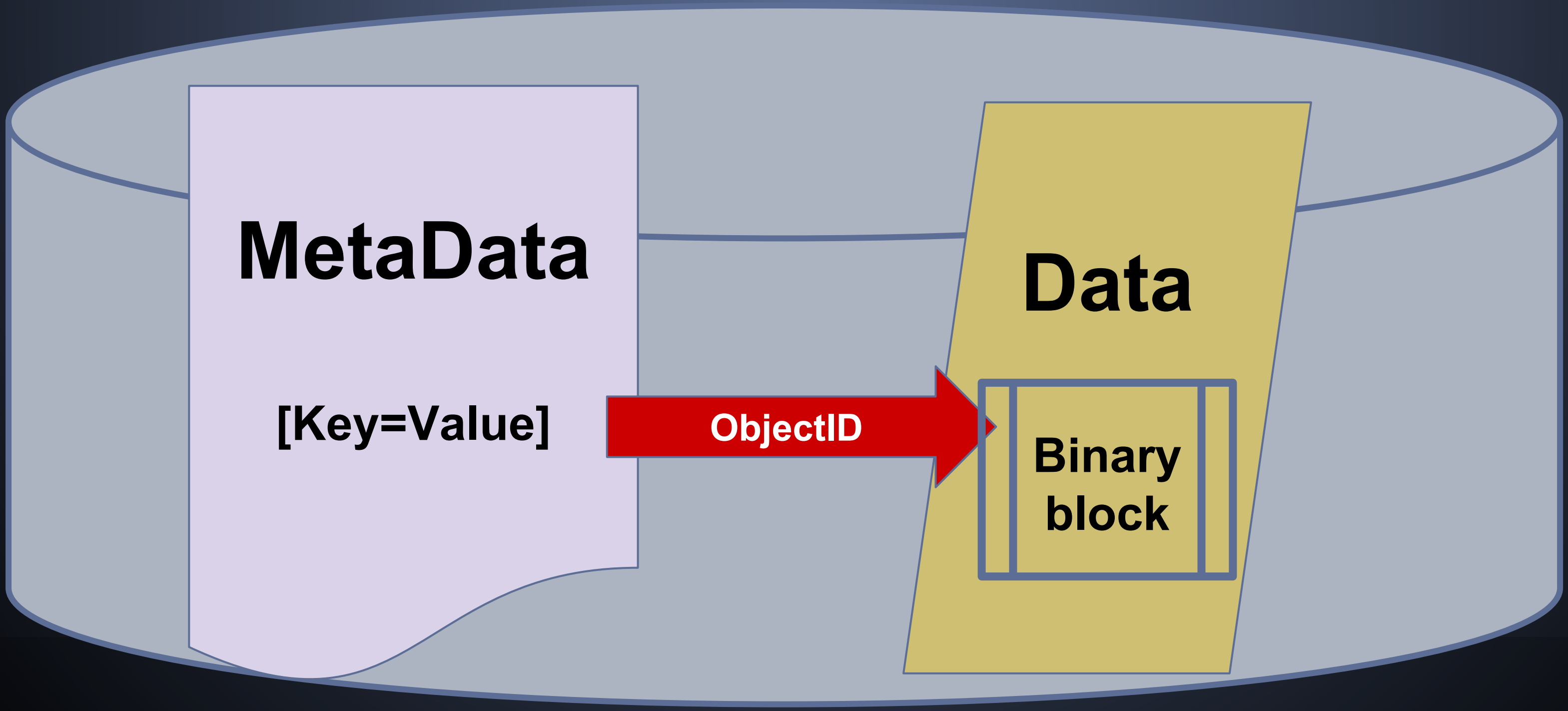
Distributed data and storage

- Data + Metadata
  - Object storage
  - Block storage
  - File storage and filesystems (POSIX)



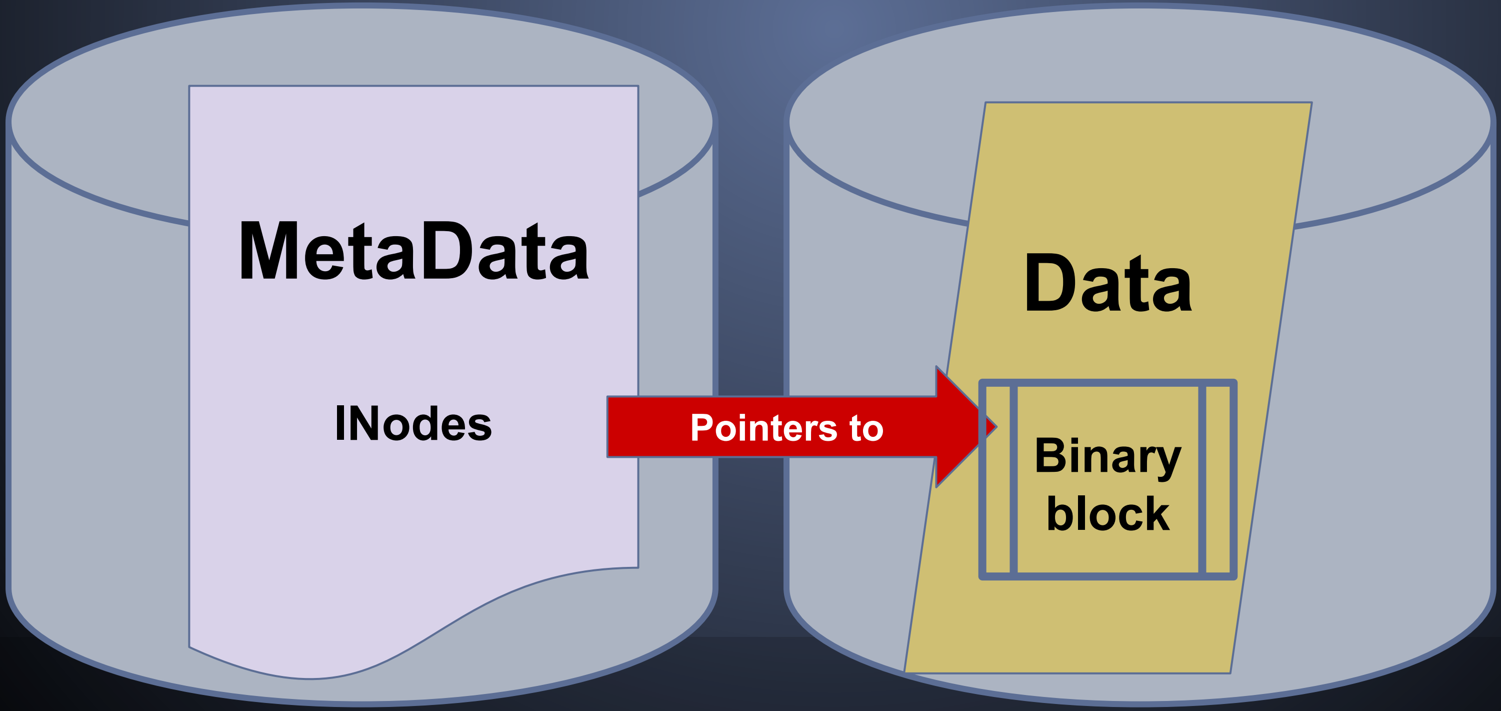
Distributed data and storage

- Data + Metadata
- **Object storage**
- Block storage
- File storage and filesystems (POSIX)



Distributed data and storage

- Data + Metadata
- Object storage
- Block storage
- **File storage and filesystems (POSIX)**



Storage: POSIX vs Objects

POSIX filesystem

- **Directories**
  - Only metadata
  - List of filenames and corresponding inode number, etc.
- **Files**
  - Metadata (inodes)
  - Data

- Hierarchical - Tree structure

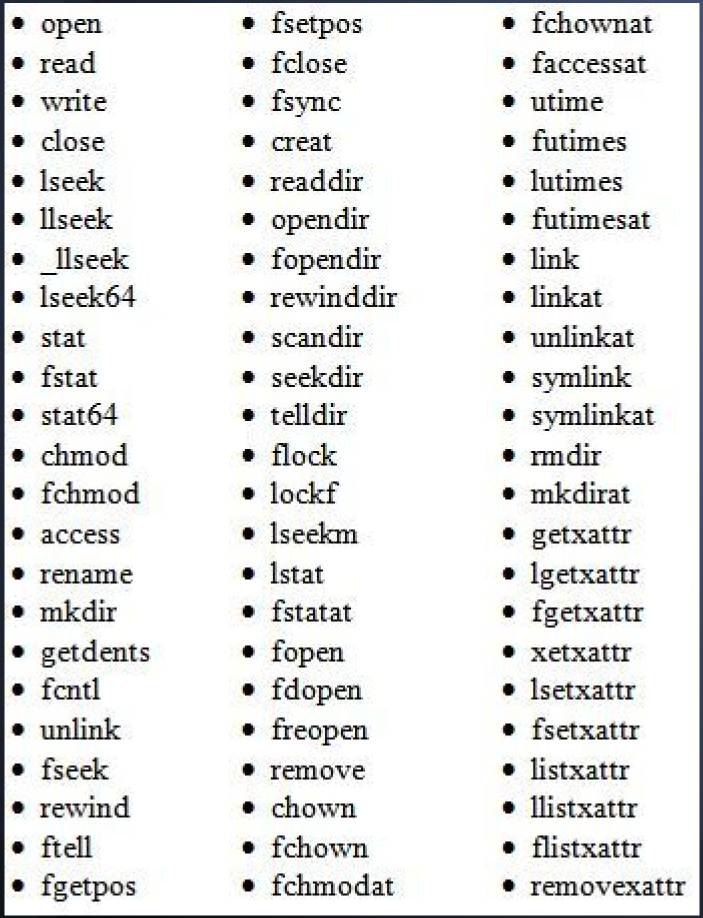
Object storage

- **Containers**
  - Only metadata
  - Information about the objects contained in _this_ container
- **Objects**
  - Metadata AND data AND ObjID

- Flat structure
  - Horizontal scalability

Storage: POSIX vs Objects

POSIX (partial list)



- open
- read
- write
- close
- lseek
- llseek
- _llseek
- lseek64
- stat
- fstat
- stat64
- chmod
- fchmod
- access
- rename
- mkdir
- getdents
- fcntl
- unlink
- fseek
- rewind
- ftell
- fgetpos
- fsetpos
- fclose
- fsync
- creat
- readdir
- opendir
- fopendir
- rewinddir
- scandir
- seekdir
- telldir
- flock
- lockf
- lseekm
- lstat
- fstatat
- fopen
- fdopen
- freopen
- remove
- chown
- fchown
- fchmodat
- fchownat
- faccessat
- utime
- futimes
- lutimes
- futimesat
- link
- linkat
- unlinkat
- symlink
- symlinkat
- rmdir
- mkdirat
- getxattr
- lgetxattr
- fgetxattr
- xetxattr
- lsetxattr
- fsetxattr
- listxattr
- llistxattr
- flistxattr
- removexattr

Objects (RESTful API)

PUT ("write"): PUT the object into the storage

GET ("read"): GET the object from the storage

DELETE: delete the object which is the file

POST: create, update, delete metadata

HEAD: returns an object's metadata

Object Storage

- Object storage is a storage architecture that manages data as objects
- Each object typically includes
  - the data itself
  - a variable amount of metadata
  - a globally unique identifier: Object ID.
- Access through RESTful API
  - Example: Ceph object storage offers access through S3 and SWIFT APIs

Block Storage

A Device:

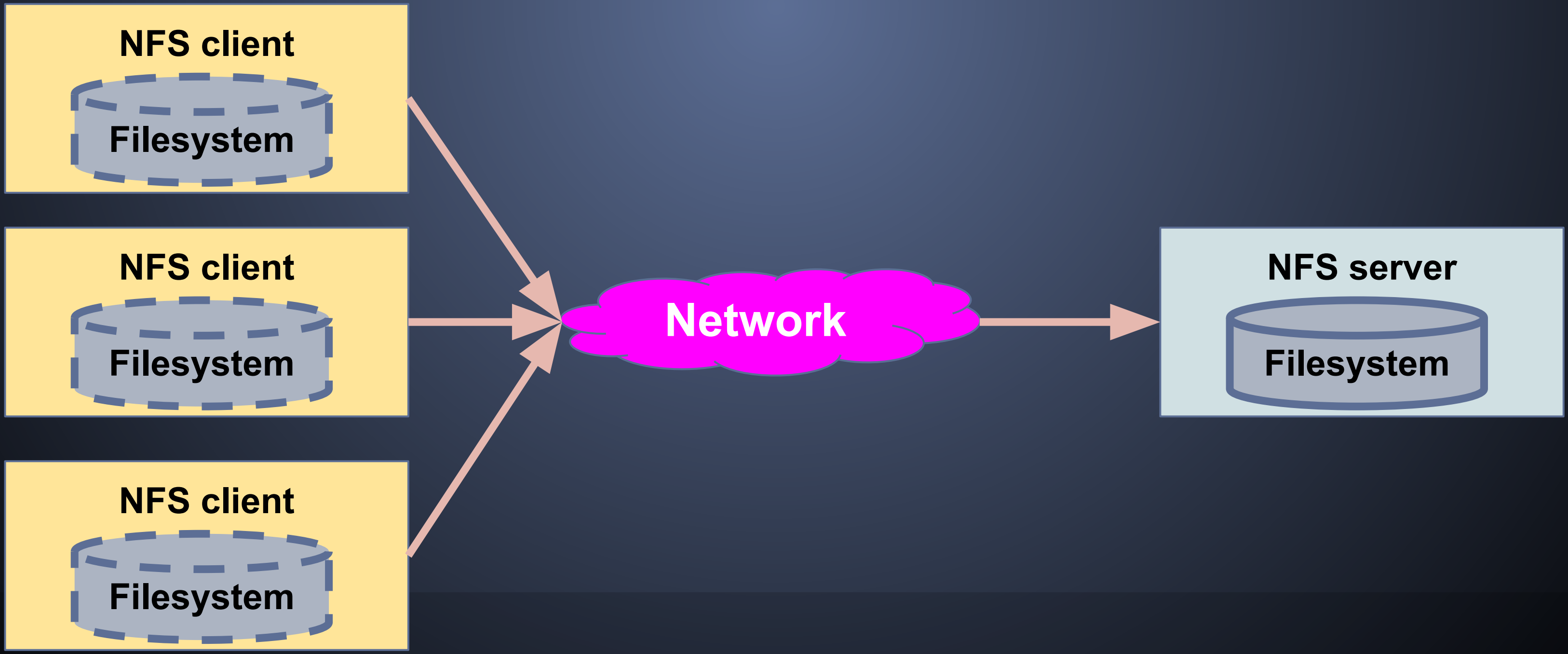
- ➢ Harddisk (and/or disk partition)
- ➢ CD
- ➢ DVD
- ➢ Disk array
- ➢ …

On the cloud:

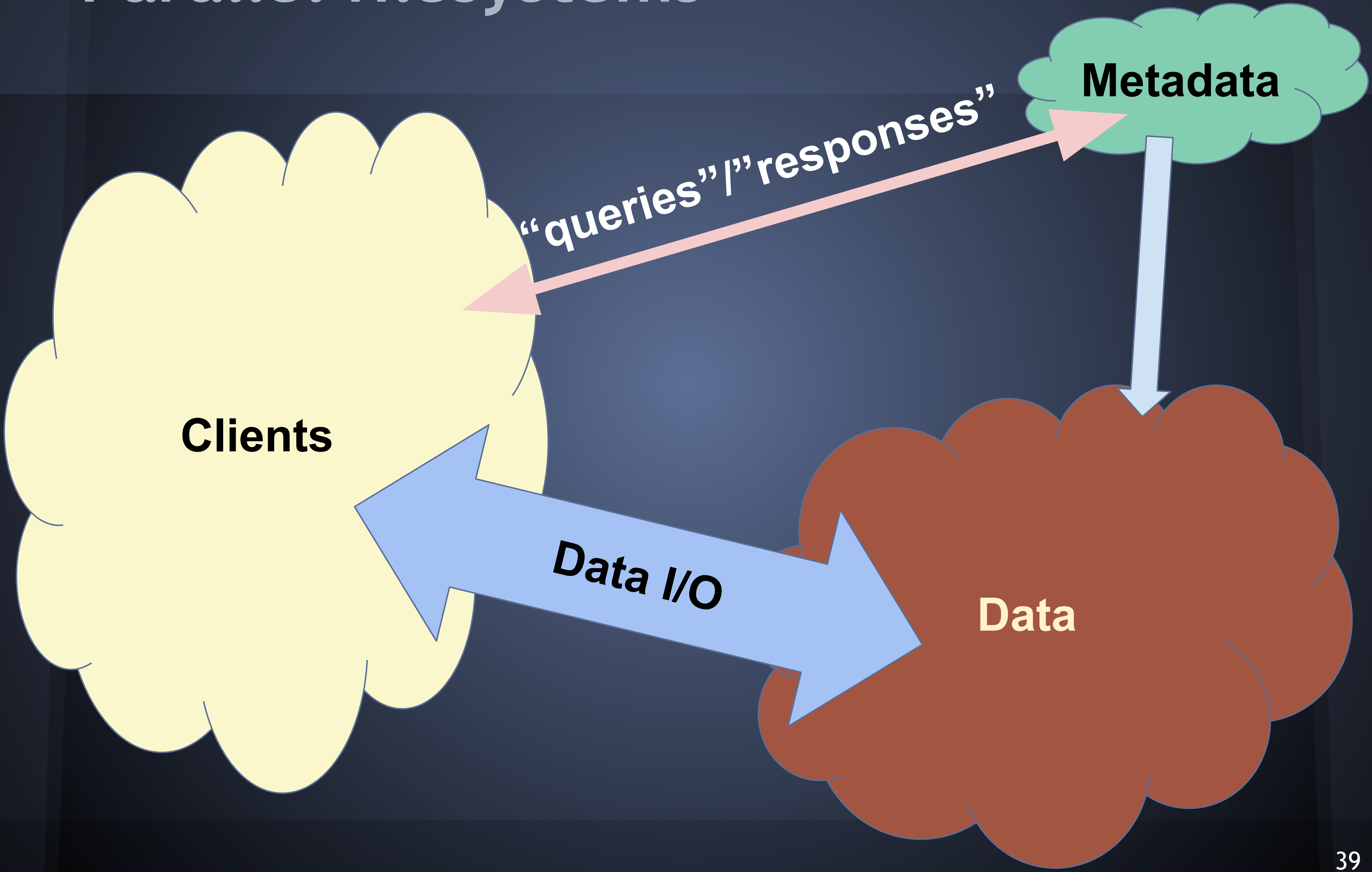
- ➢ On demand request for a disk volume
  - ○ Attach to a VM instance, as local storage to increase the storage capacity available to the instance.
  - ○ Can be formated with whatever filesystem the user wants.

Network File System: NFS

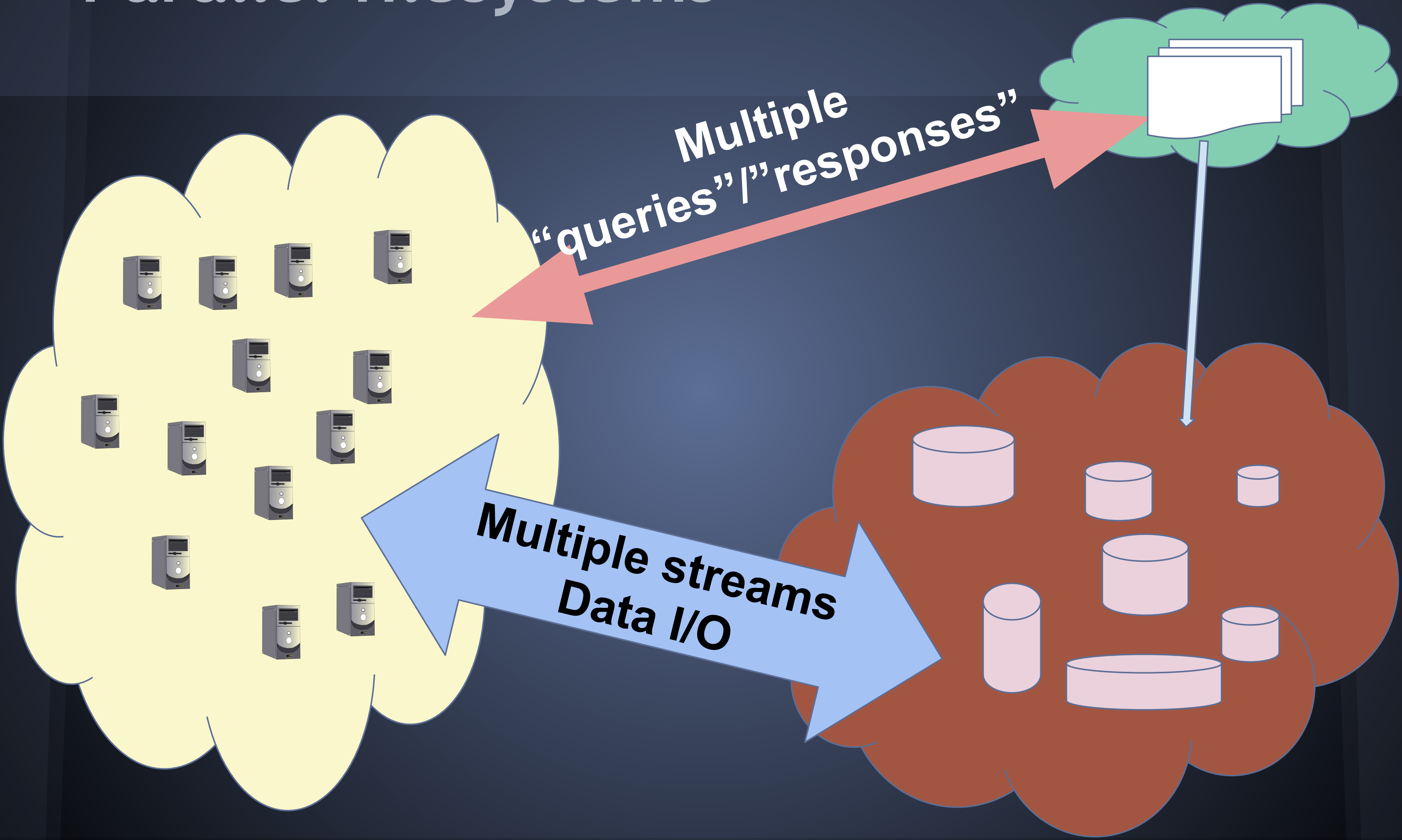
- Distributed filesystem:
  - Protocol originally developed by Sun Microsystems in 1984
  - Client/Server architecture.
  - Based on RPCs (Remote Procedure Calls)
  - Reads and writes on the client are **mapped** to read and writes on the server.
  - Only the server accesses the filesystem, managing all calls from the multiple clients.



Parallel filesystems



Parallel filesystems

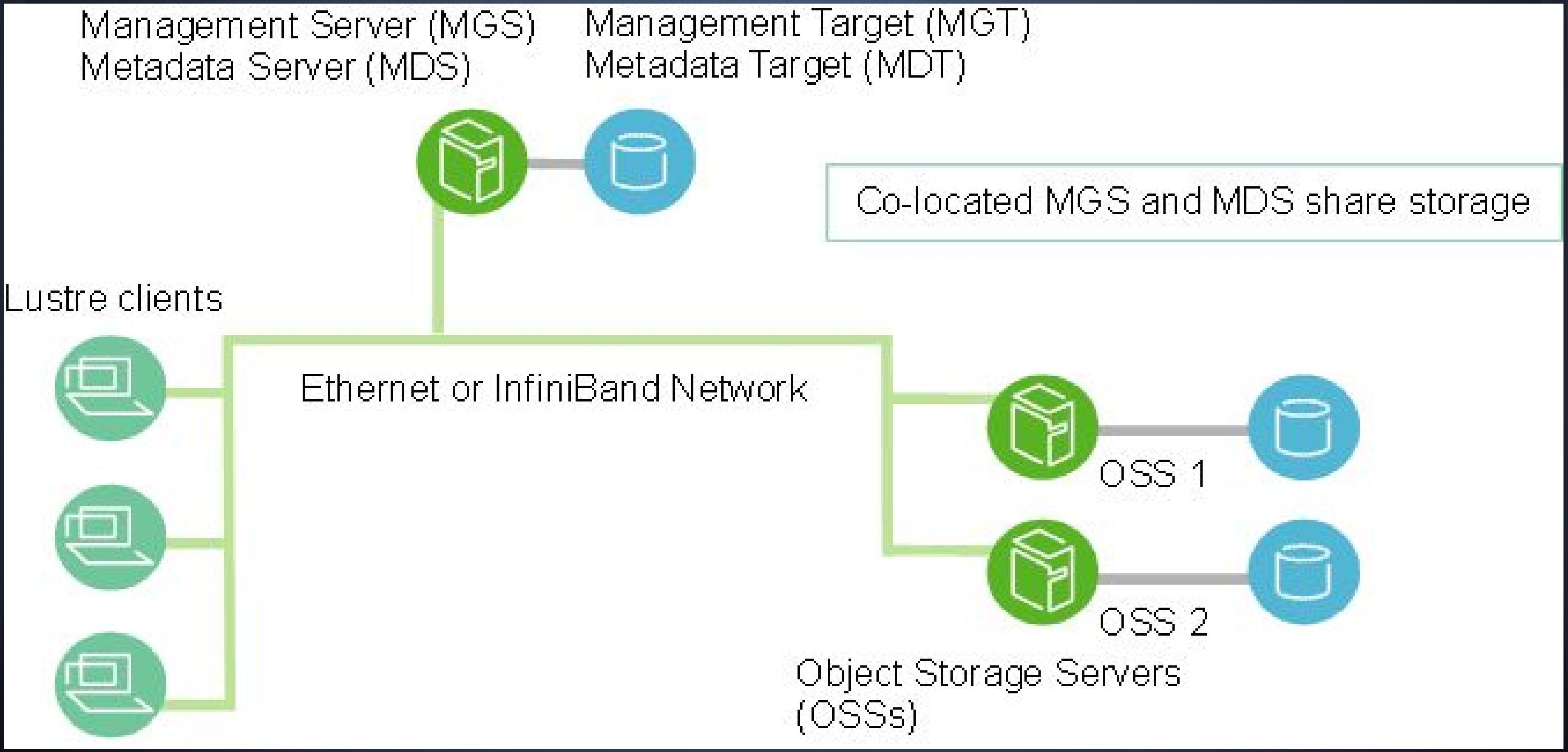


Multiple "queries"/"responses"

Multiple streams
Data I/O

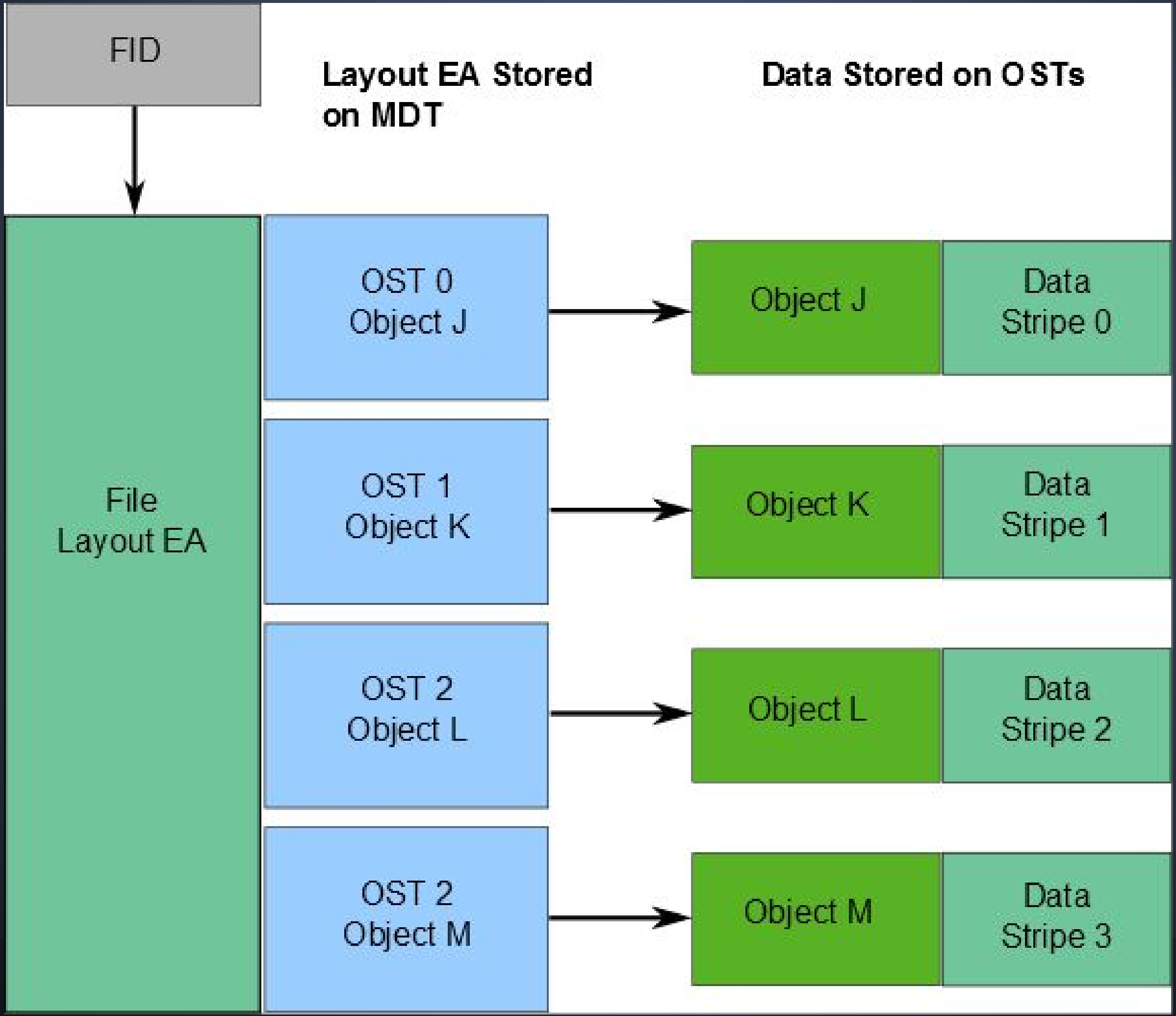
Lustre filesystem: Architecture

Lustre file system components



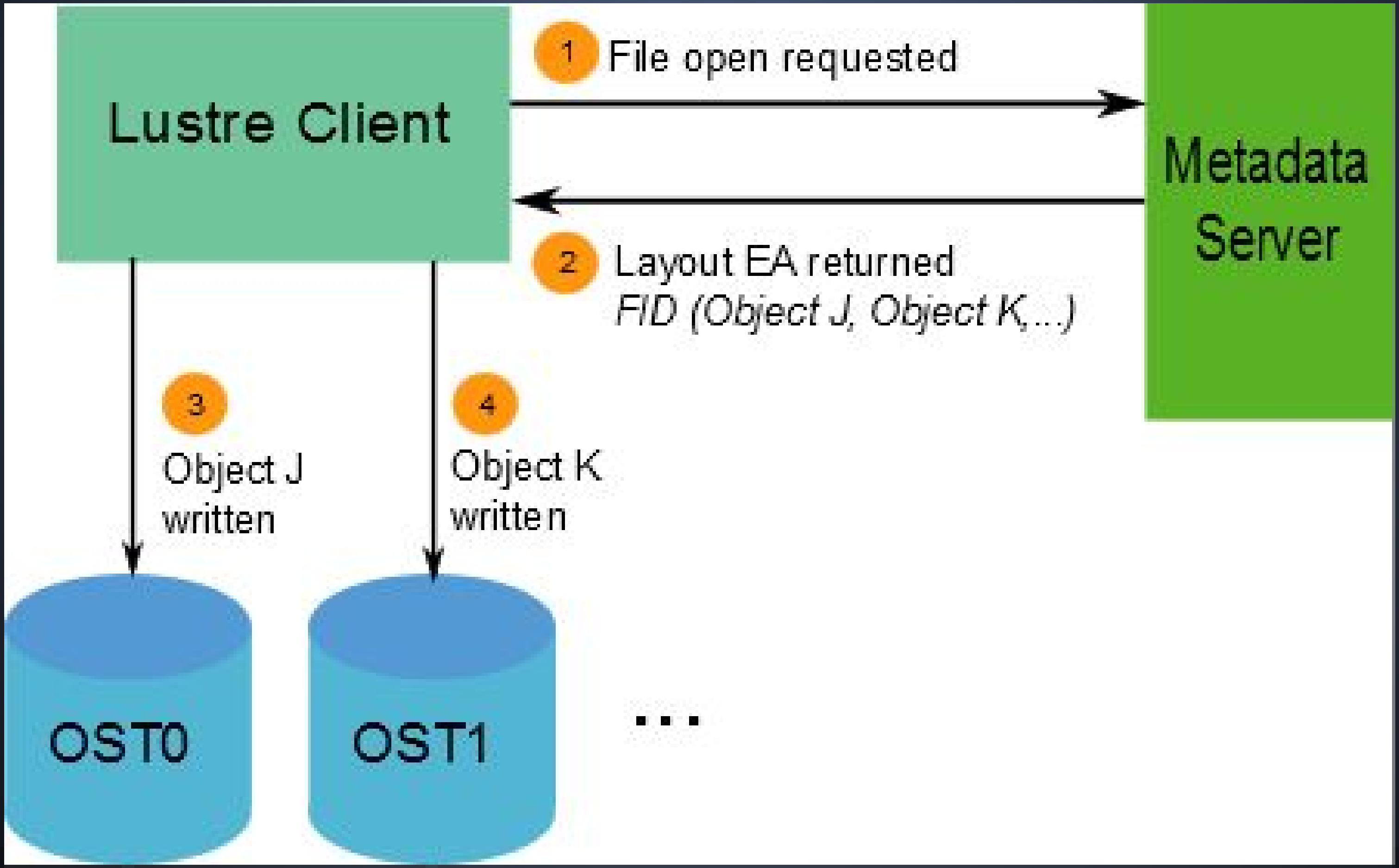
Lustre filesystem I

Layout Extended Attribute (EA) on MDT pointing to file data on OSTs



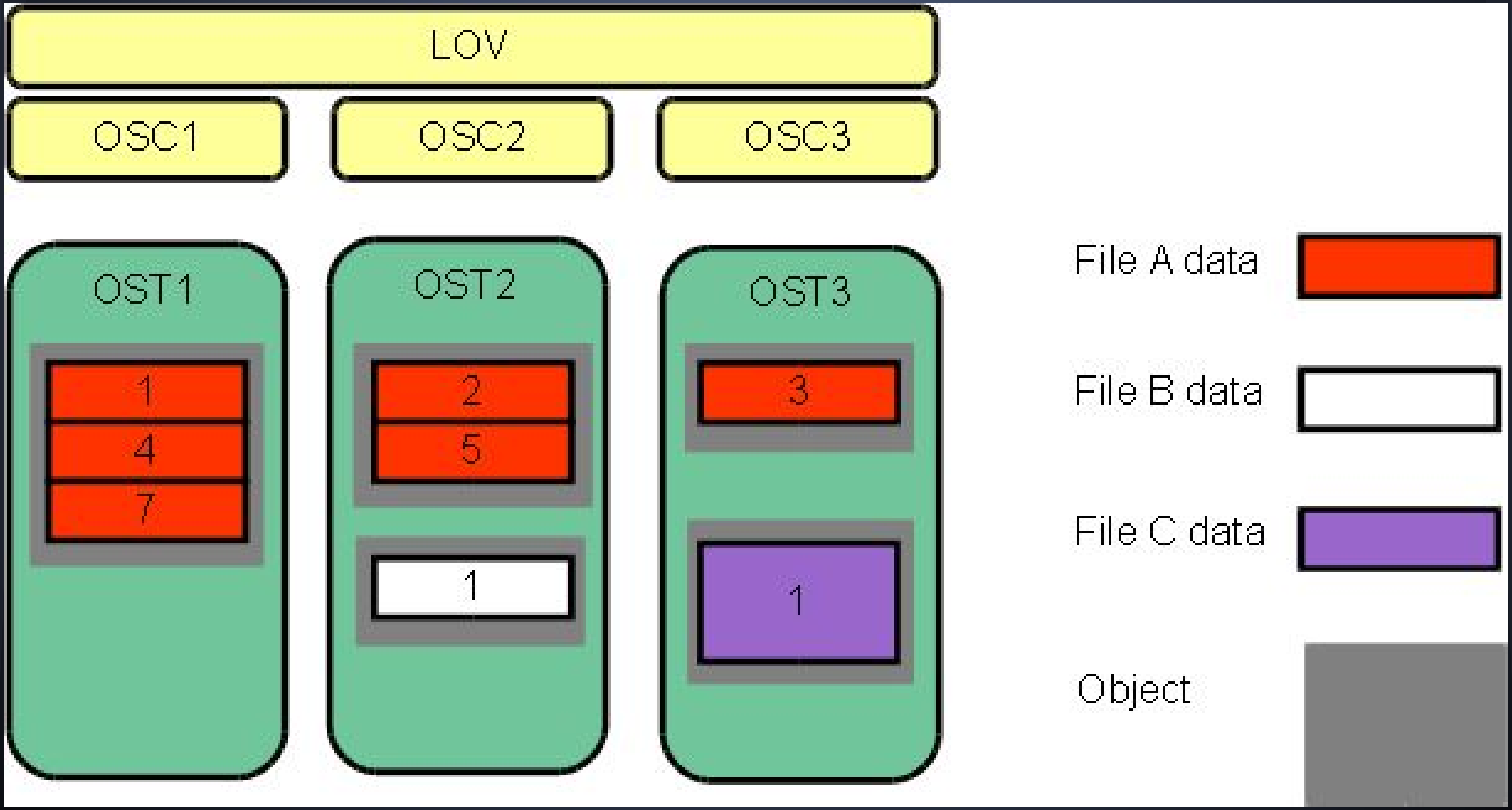
Lustre filesystem II

Lustre client requesting file data



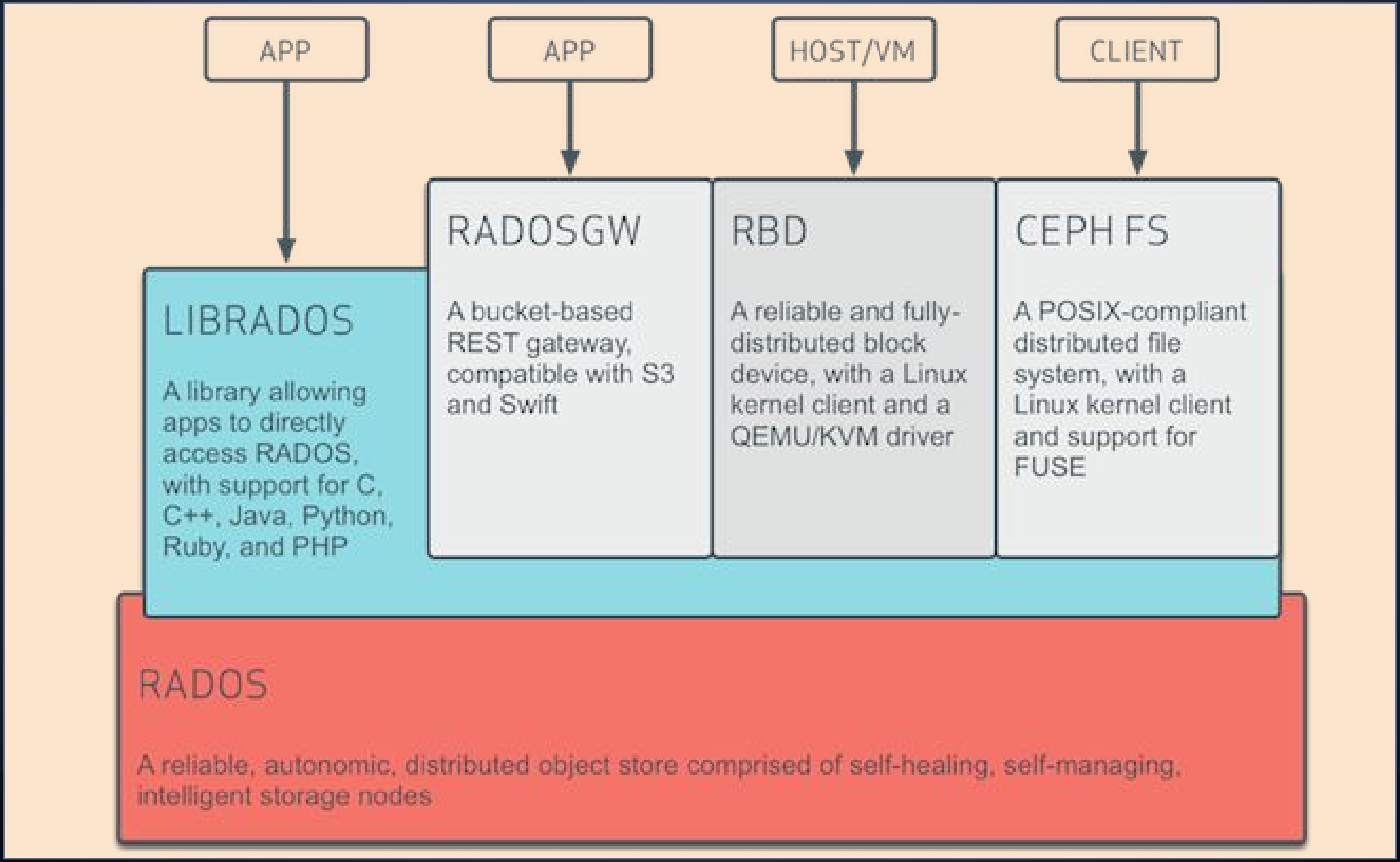
Lustre filesystem III

File striping on a Lustre file system



Ceph: Architecture

Ceph uniquely delivers object, block, and file storage in one unified system



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

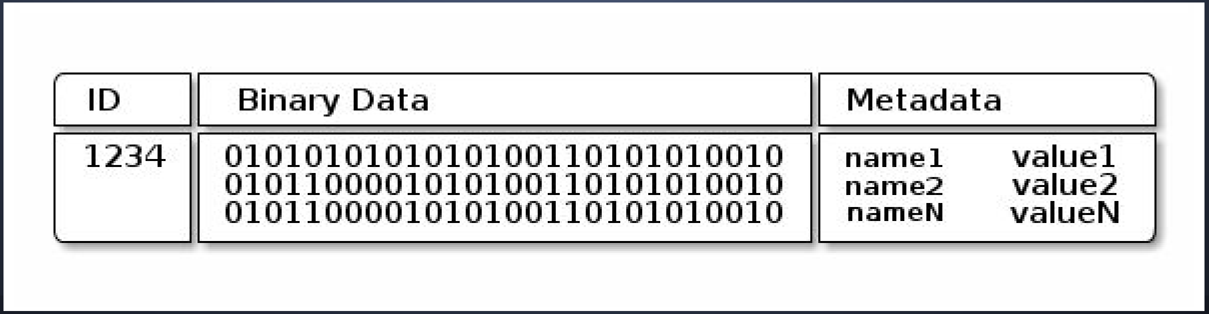
A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Ceph: Components

- **OSDs:**
  - Stores data, handles data replication, recovery, etc.
  - Provides some monitoring information to Ceph Monitors
- **Monitors:**
  - Maintains maps of the cluster state, including the monitor map, the OSD map, etc.
  - Maintains a history (called an “epoch”) of each state change in the Ceph Monitors, Ceph OSD Daemons, etc.
- **MDSs:**
  - Stores metadata on behalf of the Ceph Filesystem (POSIX).
  - Ceph Block Devices and Ceph Object Storage do not use MDS.
  - Make it feasible for POSIX file system users to execute basic commands like ls, find, etc. without placing an enormous burden on the Ceph Storage Cluster.

Ceph: Object storage I

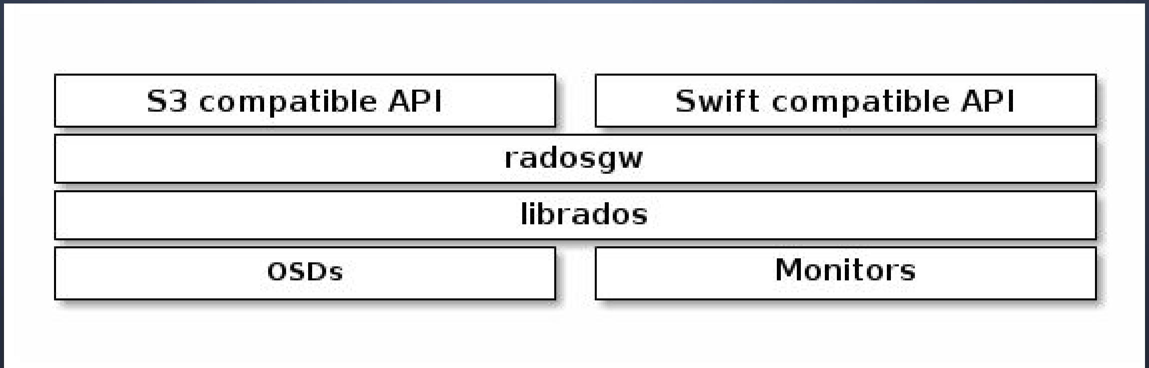
- Ceph Object Gateway - **radosgw**:
  - Object storage interface built on top of librados to provide applications with a RESTful gateway to Ceph Storage Clusters.
- Ceph Object Storage supports two interfaces:
  - **S3-compatible**: Provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API.
  - **Swift-compatible**: Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.



ID	Binary Data	Metadata	
1234	0101010101010100110101010010 0101100001010100110101010010 0101100001010100110101010010	name1 name2 nameN	value1 value2 valueN

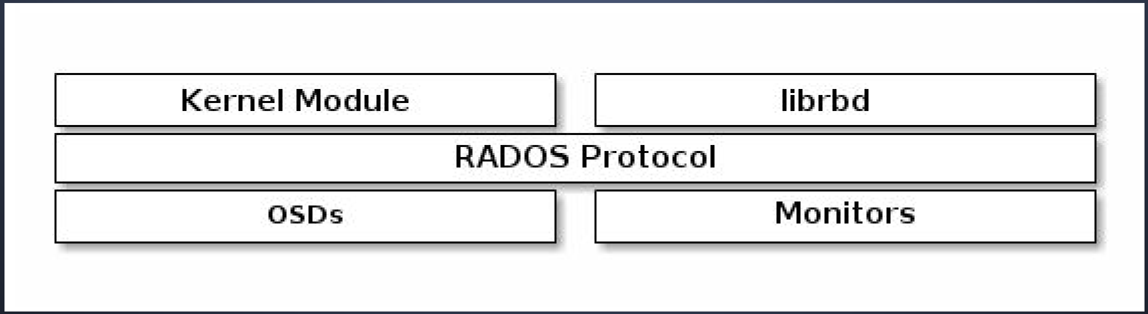
Ceph: Object storage II

- Ceph Object Gateway - **radosgw**:
  - Has its own user management
  - It can store data in the same Ceph Storage Cluster used to store data from Ceph Filesystem clients or Ceph Block Device clients.
  - The S3 and Swift APIs share a common namespace, so you may write data with one API and retrieve it with the other.



Ceph: Block storage

- Ceph block devices are:
  - Thin-provisioned, resizable and store data striped over multiple OSDs in a Ceph cluster.
  - Leverage RADOS capabilities such as snapshotting, replication and consistency.
- Ceph’s RADOS Block Devices (RBD) interact with OSDs using kernel modules or the librbd library.



Kernel Module

librbd

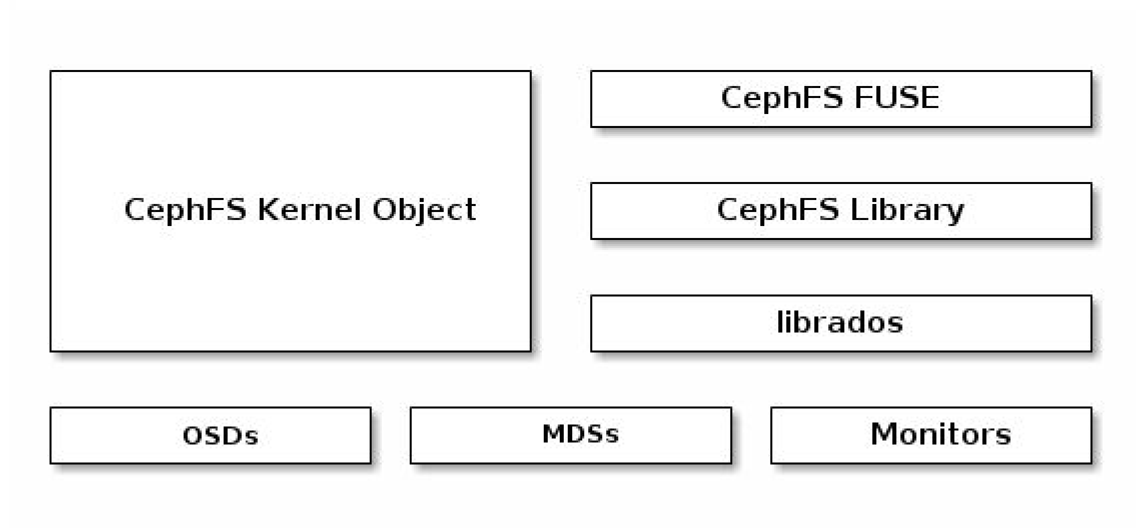
RADOS Protocol

OSDs

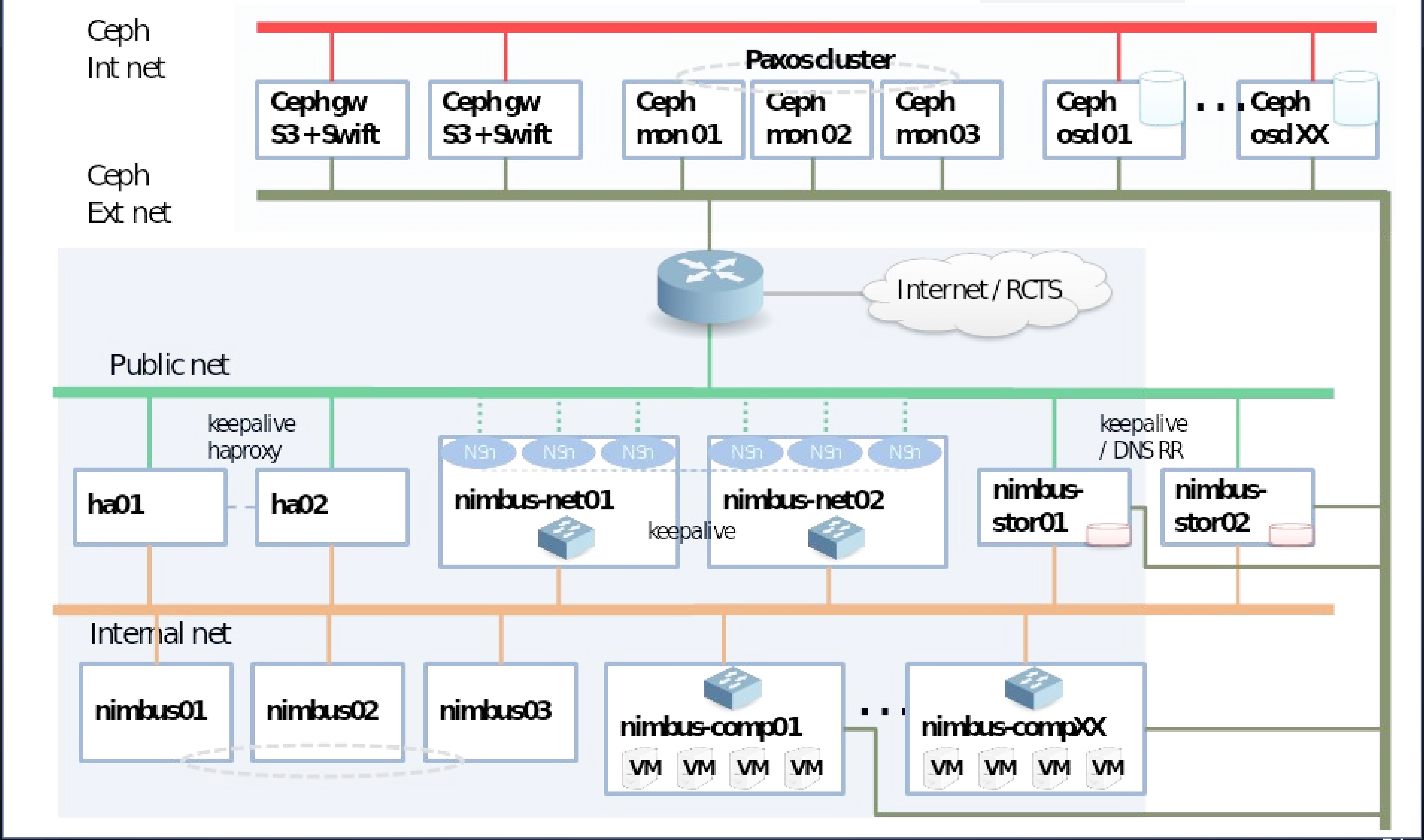
Monitors

Ceph: Filesystem

- The Ceph Filesystem (Ceph FS):
  - POSIX-compliant filesystem that uses a Ceph Storage Cluster to store its data.
  - Uses the same Ceph Storage Cluster system as Ceph Block Devices, Ceph Object Storage with its S3 and Swift APIs, or native bindings (librados).



Putting it all together
The IaaS Openstack infrastructure



Thanks!!

Questions??