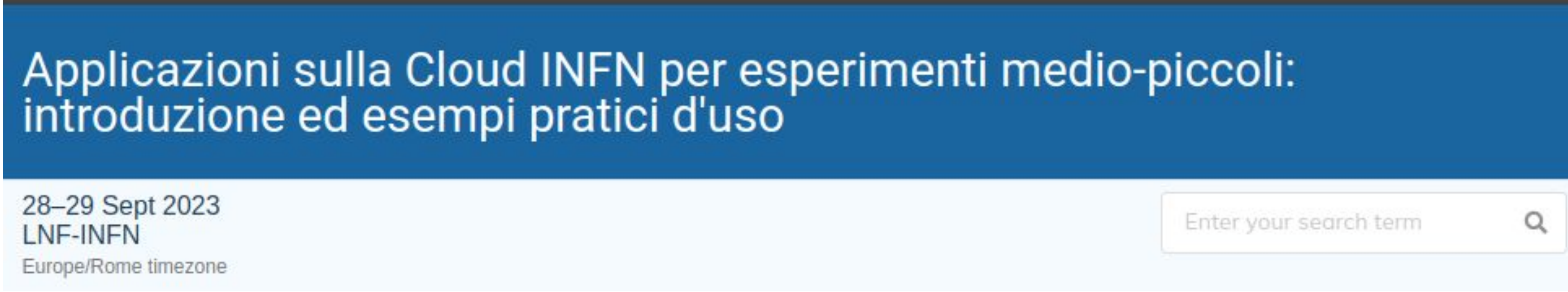


Applicazioni sulla Cloud INFN per esperimenti medio-piccoli: introduzione ed esempi pratici d'uso

28–29 Sept 2023
LNF-INFN
Europe/Rome timezone

INFN Cloud, applicazioni e prospettive

Daniele Spiga, INFN-Perugia on behalf of **[see next slide]**

spiga@pg.infn.it

LNF - 28 - 29 October 2023
INFN Perugia

Agenda

- ➢ Quick Introduction to the context
- ➢ What is there and few “advanced” examples
- ➢ Few technical challenges worth to know… and how INFN

NOTES:

- INFN-Cloud has been rebranded as DataCloud. We don’t discuss this here, for us is “just a name” . So please today consider them as synonyms
  - Not 100% correct tough
- All the material presented today is the results of the work of many people. Thus I’m presenting on their behalf, surely on behalf of the INFN-Cloud TEAM

The context: INFN-Cloud

An **internal effort** at the INFN level in order to manage a (large) fraction of the INFN resources, in order to decouple user needs from the availability of local and dedicated hardware: this applies both to data and compute

Aims at providing solutions for a wide rage of user/community needs:

- Computing **Resources optimization**
- **Reuse** of solutions
- Support R&D: **design your computing model**
- A platform for **training**
- … And of course cover the increasing needs of the community doing AI research (which needs accelerators, large systems, fast access to training data)

Few highlights

The vision

Allow researchers to exploit "free" and open services to manage workflows, build pipelines, data processing and analysis and, of course, to share/to reuse technical solutions

- Allow researchers to focus on science

Technical drivers:

- to enable users **to create and provision infrastructure deployments, automatically and repeatedly, with almost zero effort**.
- To Implement the ***Infrastructure as Code* paradigm** based on declarative approach: **allows to describe "What" instead of "How"**
  - Let the underlying system to deal with technicalities
- To promote (and support) **container-based solutions**
- To grant data sharing among users/infrastructures

…and from user perspective: few pillars

end users should handle just few pillars

- What the user should/might see out of all of the underlying system?

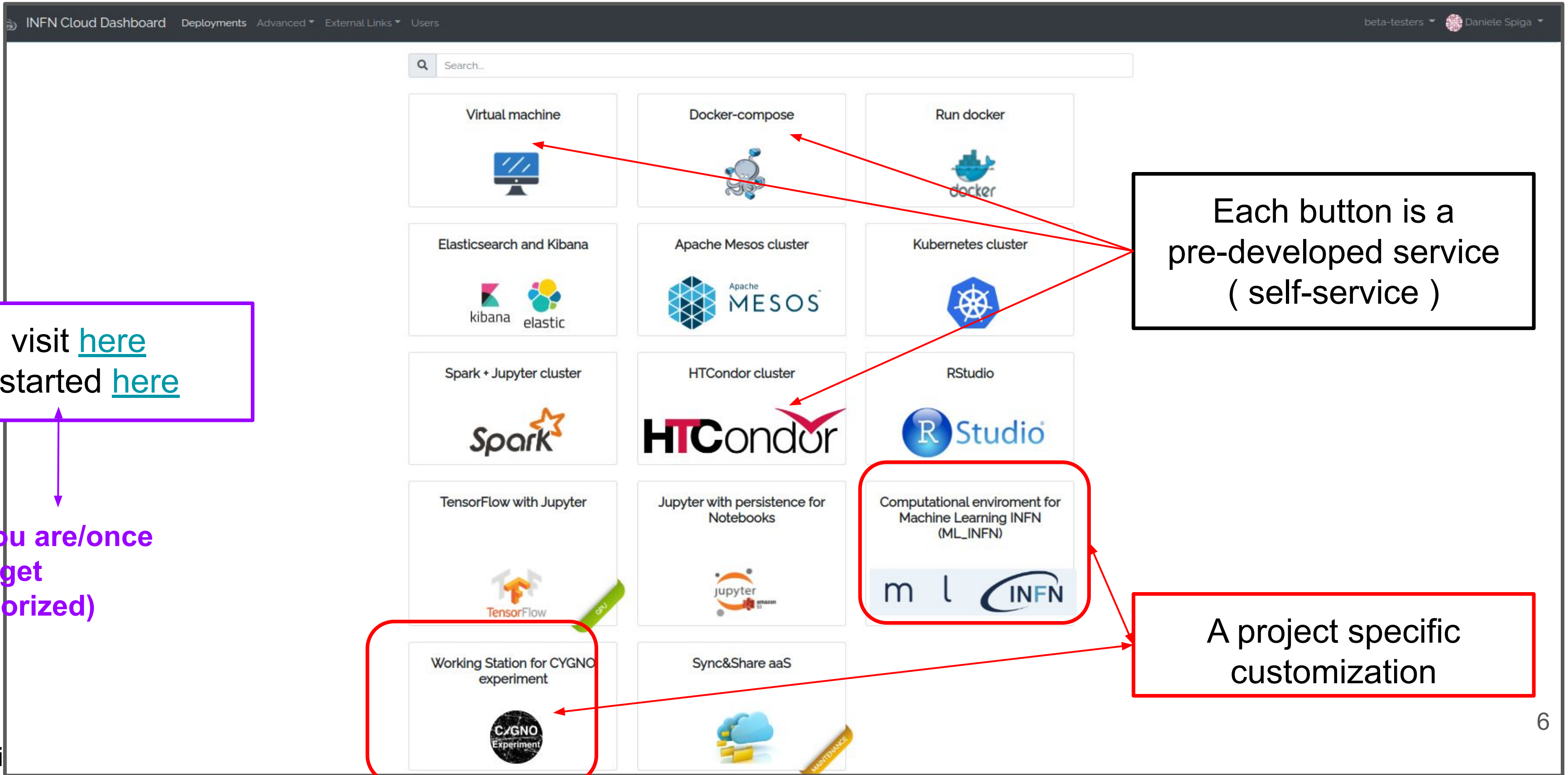
Software management: a central role is played by container. A standard unit of software suitable to create **user tailored environment**, (share and port everywhere).

- Users create containers, the system distribute them via global file systems…

Infrastructure management: in principle user might chose to know “nothing” about infrastructure (SaaS model and above).

- If a researcher need/swants to customize its infrastructure, the system (the Cloud) should offer handles… **through templates** [see later]

It already exist! Try it out yourself



Each button is a pre-developed service (self-service)

A project specific customization

You can visit here
Getting started here

(if you are/once you get authorized)

The service implementation strategy in a nutshell

The employed strategy is based on the **Infrastructure as Code paradigm**.
Users describe "**What**" is needed rather than "**How**" a specific service or functionality should be implemented.
The adopted technologies enable a Lego-like approach: services can be composed and modules reused to create the desired infrastructure.

OASIS TOSCA

TOSCA is used to model the topology of the whole application stack

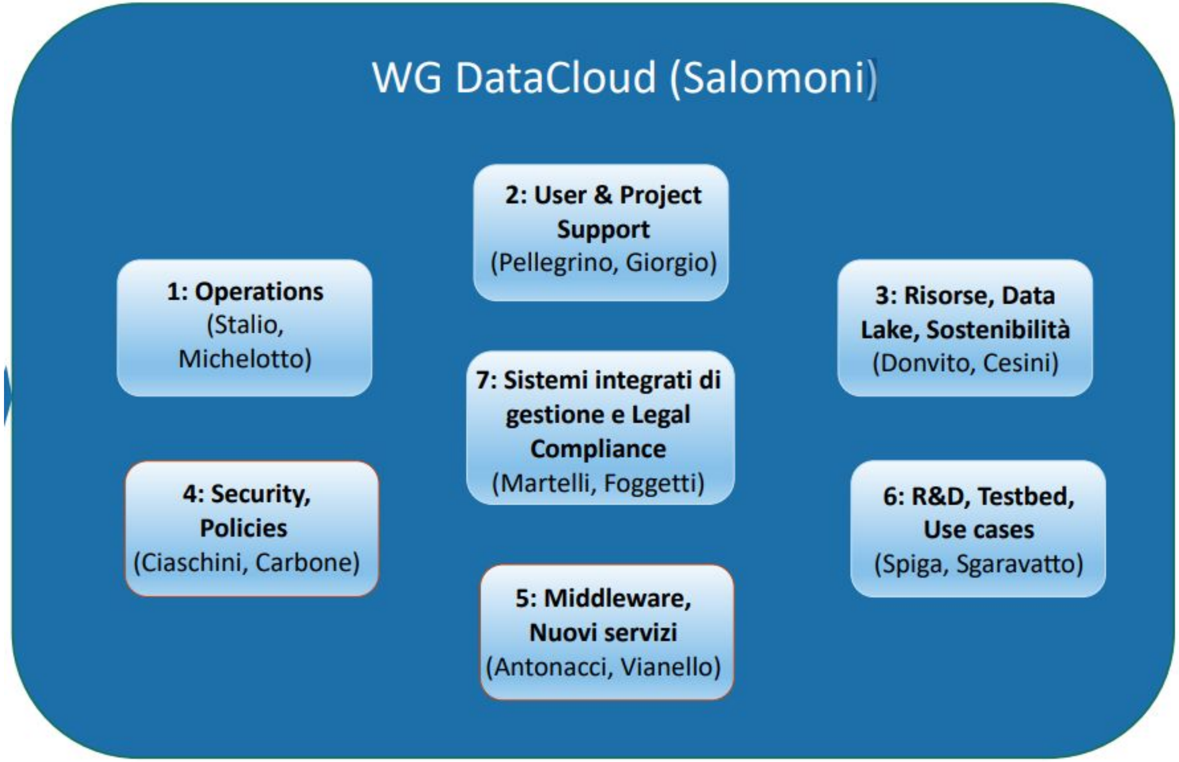
ANSIBLE

Ansible is used to automate the configuration of the virtual environments

docker

Docker is used to encapsulate the high-level application software and runtime

Quick NOTE: Not only technical aspects



WG DataCloud (Salomoni)

2: User & Project Support (Pellegrino, Giorgio)

1: Operations (Stalio, Michelotto)

3: Risorse, Data Lake, Sostenibilità (Donvito, Cesini)

7: Sistemi integrati di gestione e Legal Compliance (Martelli, Foggetti)

4: Security, Policies (Ciaschini, Carbone)

6: R&D, Testbed, Use cases (Spiga, Sgaravatto)

5: Middleware, Nuovi servizi (Antonacci, Vianello)

The INFN project that is responsible for all that is DataCloud.

- 7 Work Packages with clearly defined roles and responsibilities.

What’s coming

An **internal effort** at the INFN level in order to manag fraction of the INFN resources, in order to decoupl from the availability of local and dedicated hard plies both to data and compute

The INFN-Cloud is going to evolve as the Middleware of the National Center ICSC

INFN CLOUD

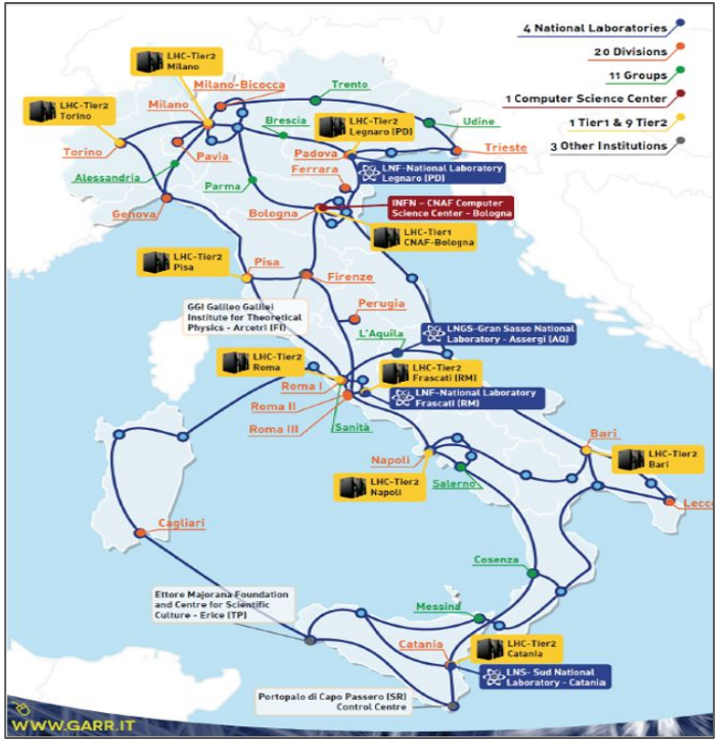
Aims at providing solutions for a wide rage of user/community needs:

- Computing **Resources optimization**
- **Reuse** of solutions
- Support R&D: **design your computing model**
- A platform for **training**
- … And of course cover the increasing needs of the community doing AI research (which needs accelerators, large systems, fast access to training data)

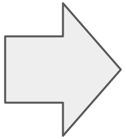
Few highlights

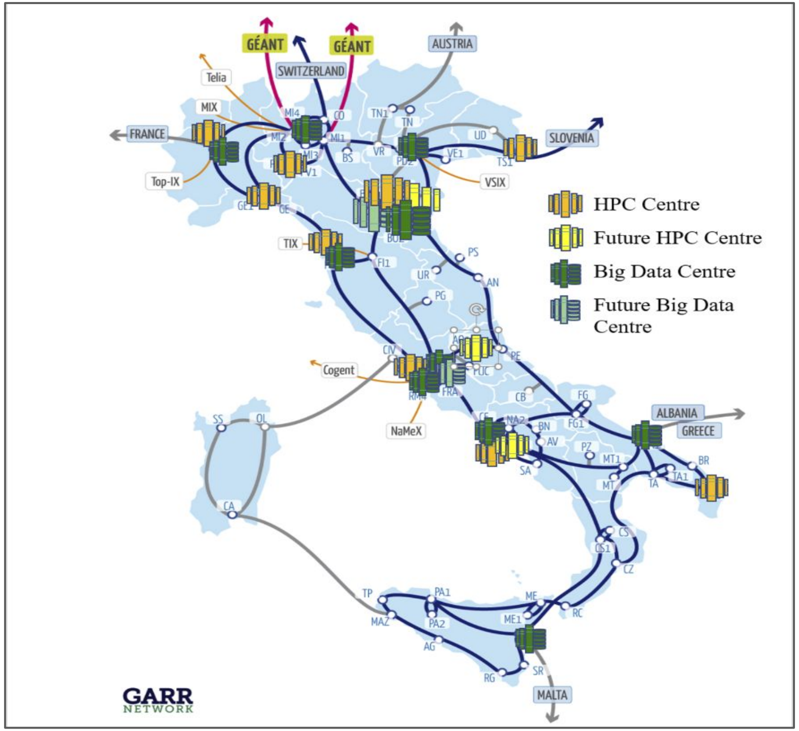
ICSC (and TeraBIT)...

INFN Computing today



ICSC Tomorrow





▥ HPC Centre

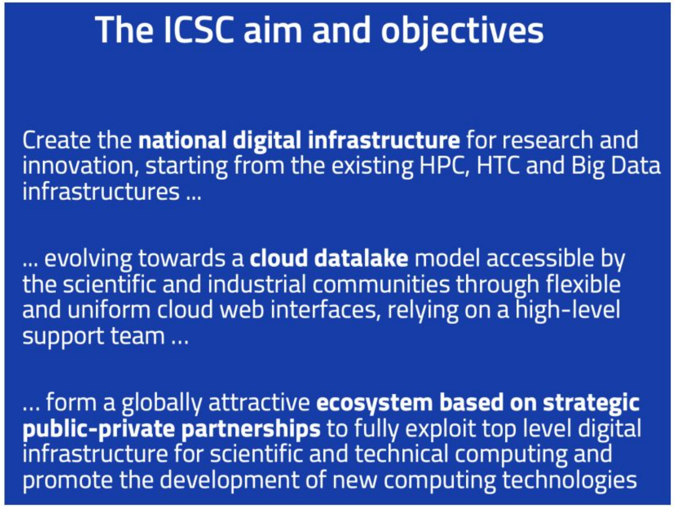
▥ Future HPC Centre

▥ Big Data Centre

▥ Future Big Data Centre

One of the Challenge: the continuum

One of the main enhancements we foreseen is the integration of all the resources (irrespective of the interfaces used to expose them) to hide technicalities and facilitate the exploitation

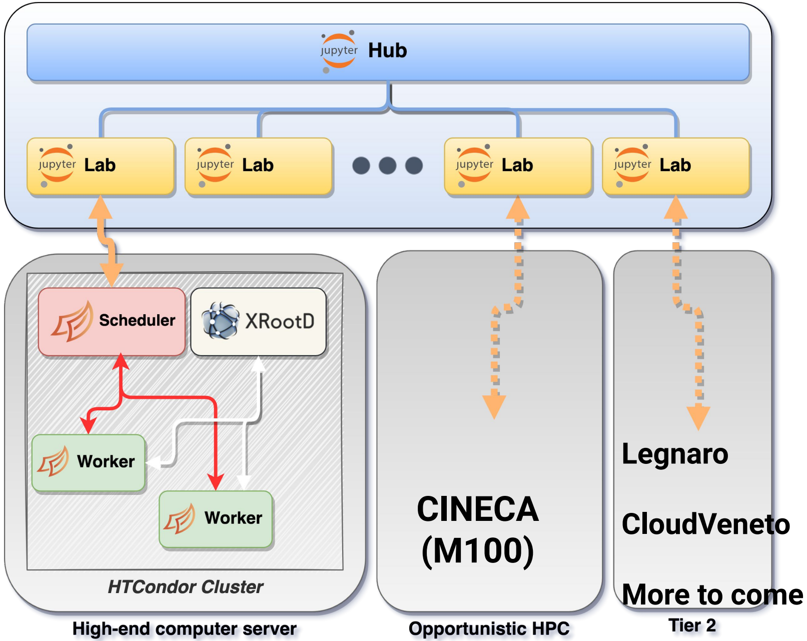


The ICSC aim and objectives

Create the **national digital infrastructure** for research and innovation, starting from the existing HPC, HTC and Big Data infrastructures …

… evolving towards a **cloud datalake** model accessible by the scientific and industrial communities through flexible and uniform cloud web interfaces, relying on a high-level support team …

… form a globally attractive **ecosystem based on strategic public-private partnerships** to fully exploit top level digital infrastructure for scientific and technical computing and promote the development of new computing technologies



And now: the user perspectives

INFN-Cloud is offering a always evolving (use-case driven) portfolio of services. This is the idea of “self service”... however **there are additional handles** offered to the users

- Access INFN-Cloud **Managed services**
  - Few examples: **Nootebook as a Serivce, Storage (S3), Data Management, Software Management**, **offloading**…
    - Note few of them are production ready, other in test oder in dev
- To **extend/customize existing services**
- To **develop brand new services**

  and/or cloud applications

In the following we will see few examples here

What about if a service you need is not already there ?

You can implement it yourself and make it available to other communities / users.

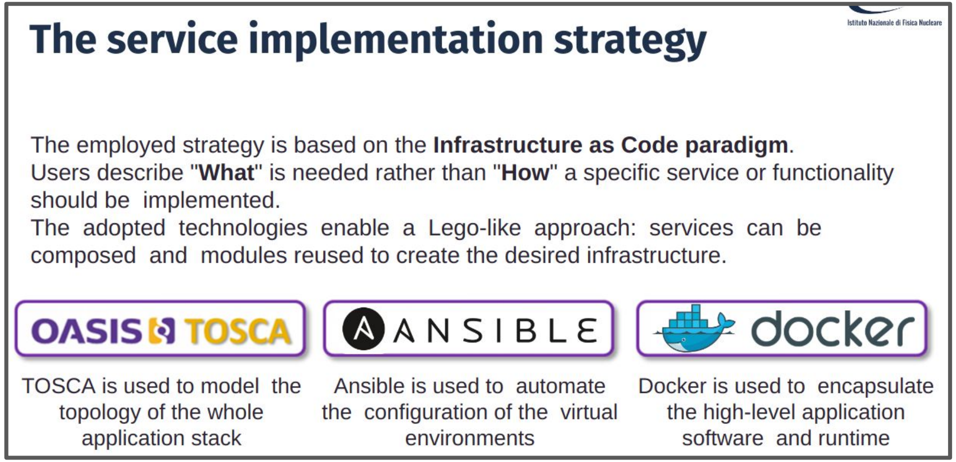
This is the idea behind the service portfolio..

- Simple example: CYGNO wasn’t there by magic …

How ?

INFN-Cloud offers support / playground to implement your own system

- WP2/WP5/WP6 are supposed to be where to look for “support”



The service implementation strategy

The employed strategy is based on the **Infrastructure as Code paradigm**. Users describe "**What**" is needed rather than "**How**" a specific service or functionality should be implemented. The adopted technologies enable a Lego-like approach: services can be composed and modules reused to create the desired infrastructure.

TOSCA is used to model the topology of the whole application stack

Ansible is used to automate the configuration of the virtual environments

Docker is used to encapsulate the high-level application software and runtime

So if you’d like to raise the bar: what could be done

Some Examples…

Specialized hardware: 1

- “pioneered” by INFN-Cloud & ML_INFN joint venture (GPU access)

Workflow management: interactive processing 2

- Exploiting parallelism, Implement Interactive analysis workflow

Automation: Building your pipeline 3

- Exploiting cloud-native services to build a pipeline

Data Management: inject/eject your data from Datalake 4

1

ML_INFN

“Machine Learning at INFN” (ML_INFN) project, is a INFN CSN5 funded project

Three assets:

- ➔ Organize training/hackathons events addressing ML related theoretical aspects and use case specific implementation
- ➔ Create a knowledge base to collect/promote/share already existing studies in our domain (and in literature)
- ➔ Build and promote the adoption of performant and tailored technological platforms because a effective platform for Machine Learning prototyping and developing might represent a technical challenge

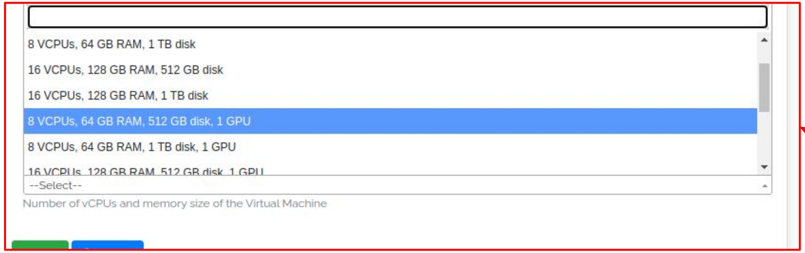
The rationale: researchers need **interactive environment**, the **access to hardware accelerators** (GPUs) and possibly a non-limiting **access to data** (i.e. training data). Handles to **create user tailored environment** is a key and finally groups need to **collaborate and share resources, data and code**.

1

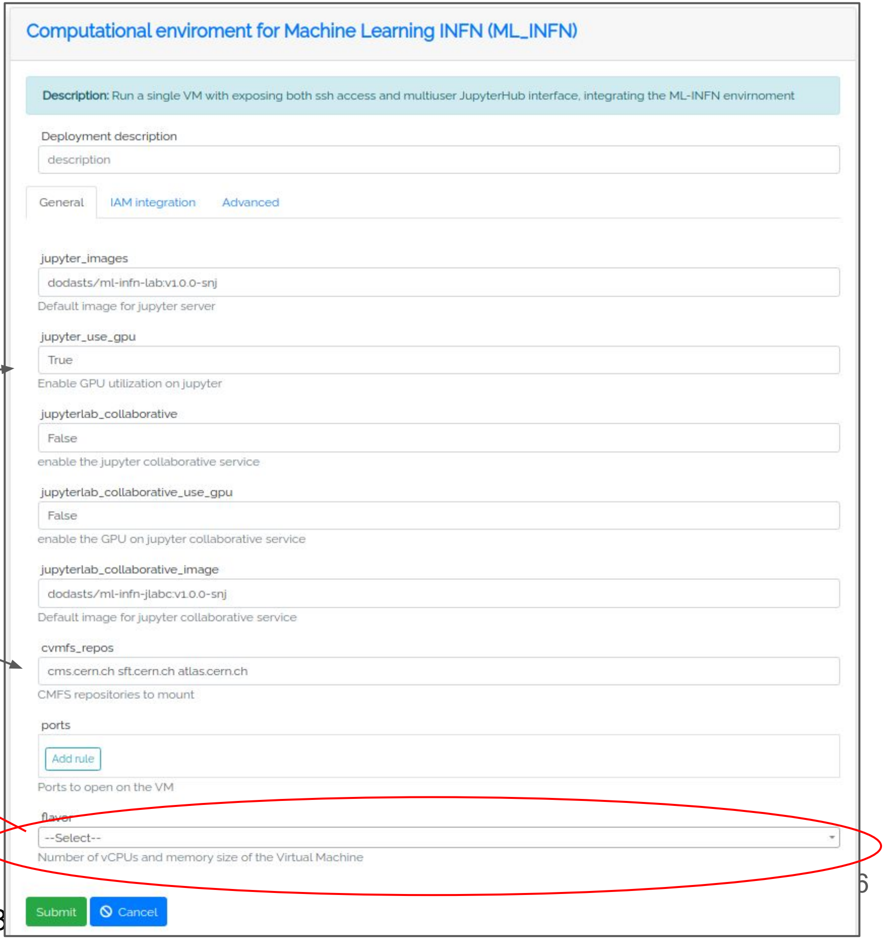
The ML_INFN customized implementation

Simple high-level configuration template to create your personal environment

- Either for single user and multi users (group activities)
- Ask for CVMFS areas, GPUs, ...



8 VCPUs, 64 GB RAM, 1 TB disk
16 VCPUs, 128 GB RAM, 512 GB disk
16 VCPUs, 128 GB RAM, 1 TB disk
8 VCPUs, 64 GB RAM, 512 GB disk, 1 GPU
8 VCPUs, 64 GB RAM, 1 TB disk, 1 GPU
16 VCPUs, 128 GB RAM, 512 GB disk, 1 GPU
--Select--
Number of vCPUs and memory size of the Virtual Machine



Computational enviroment for Machine Learning INFN (ML_INFN)

Description: Run a single VM with exposing both ssh access and multiuser JupyterHub interface, integrating the ML-INFN envirnoment

Deployment description: description

jupyter_images: dodasts/ml-infn-lab:v1.0.0-snj
Default image for jupyter server

jupyter_use_gpu: True
Enable GPU utilization on jupyter

jupyterlab_collaborative: False
enable the jupyter collaborative service

jupyterlab_collaborative_use_gpu: False
enable the GPU on jupyter collaborative service

jupyterlab_collaborative_image: dodasts/ml-infn-jlabc:v1.0.0-snj
Default image for jupyter collaborative service

cvmfs_repos: cms.cern.ch sft.cern.ch atlas.cern.ch
CMFS repositories to mount

ports

Ports to open on the VM

flavor: --Select--
Number of vCPUs and memory size of the Virtual Machine

1

ML_INFN evolution…

ML_INFN project proposed a new round of funding. **From the technological perspective the vision is to move to a more flexible provisioning model**

- **From a VM based approach to the k8s based one**

Still, the user experience will be the same (well improved…) while there will be room for more functionalities and surely there will be handles to optimize the Hardware usage.

P.I Lucio Anderlini INFN-FI Anderlini lucio.anderlini<AT>fi.infn.it

Managing workload: Transparent offloading

Distributing workload to parallelize data processing can be a complex task.

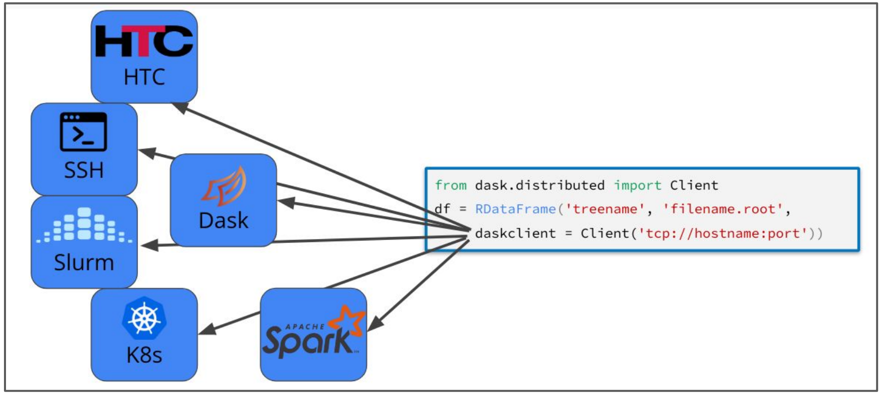
A dream would be to be able to access huge amount of computing capacity quickly and easily

- to process (huge amount of) data → **Interactive** or **Quasi interactive**
- reduce the time to insight: going interactive over huge amount of data

Simplifying: to being able to scale up to a full workstation and transparently scale out to a cloud

Several communities are exploiting the use high level frameworks capable of leverage distributed computing engine

- I.e. RDataFrame is getting traction @HEP

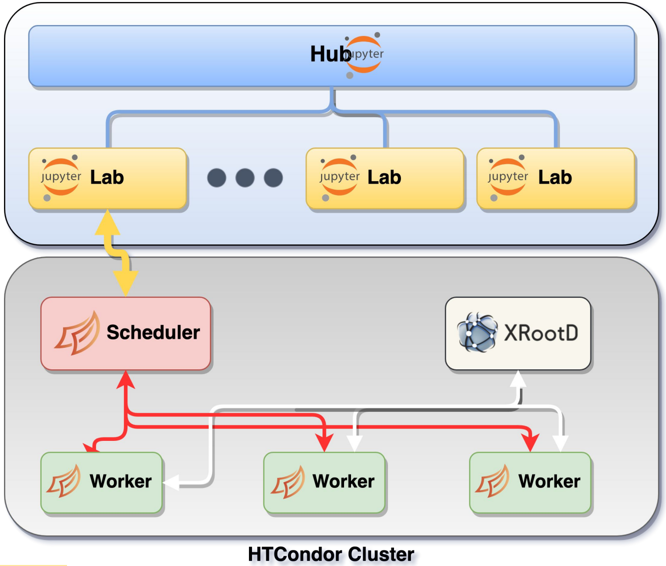


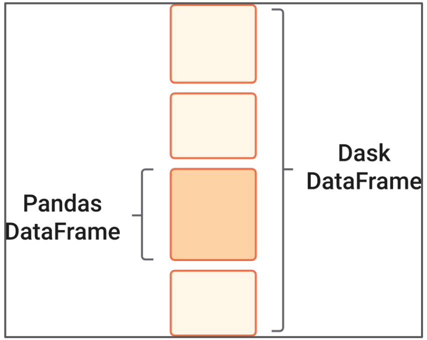
```
from dask.distributed import Client
df = RDataFrame('treename', 'filename.root',
daskclient = Client('tcp://hostname:port'))
```

High rate analysis (HEP Example)

2

- **JupyterHub** (JHub) and **JupyterLab** (JLab) to manage the **user-facing part of the infrastructure**
  - **Comple abstraction**
- **DASK** to introduce the **scaling over a (highly) distributed system**
  - **Huge amount of resources, quickly and easily**
- **[XRootD is a bit HEP specific.. See it as a way to access any data anywhere]**





Dask is not HEP. It is a library that allow to scale the existing **Python and PyData ecosystem**.

- Looks and feels like the pandas API, but for parallel and distributed workflows.

R&D on interactive data analysis More details here

2

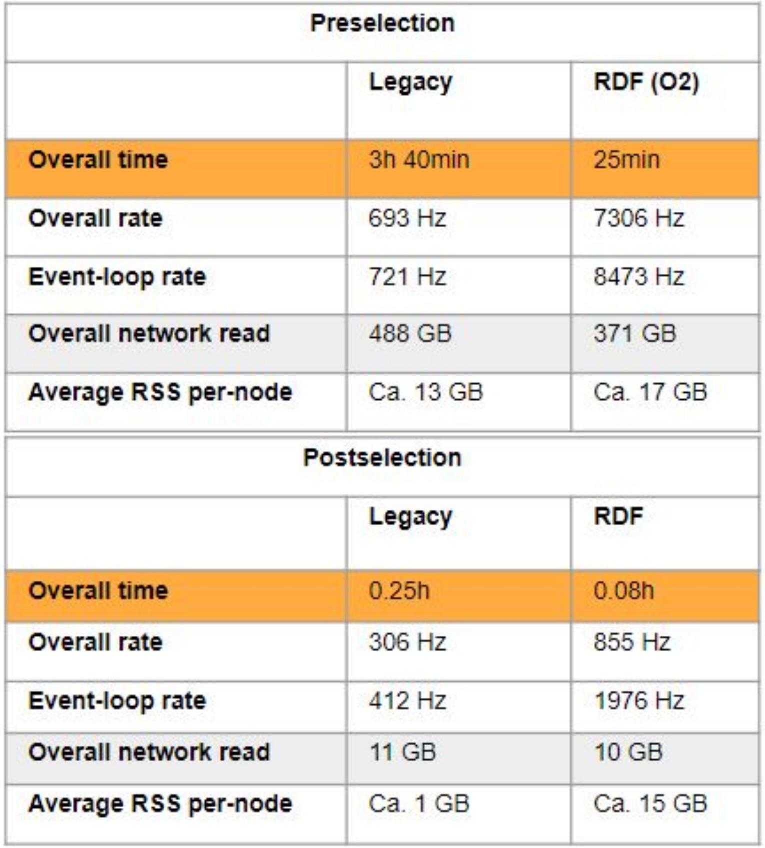
Early results

This is CMS Specific but still a good example

What about if you’re memory limited during your data (pre-)processing…

Measured two different workflow distribution approaches

- Using VBS SSWW with a light lepton and an hadronic tau in final state
  - ported from legacy approach (nanoAOD-tools/plain PyROOT-based) to RDataFrame.
- Data processed about 2TB (Data + Monte Carlo)
- The comparison tests are performed
  - on the same nodes of the cluster
  - very same HTCondor infrastructure
  - A fair benchmark.



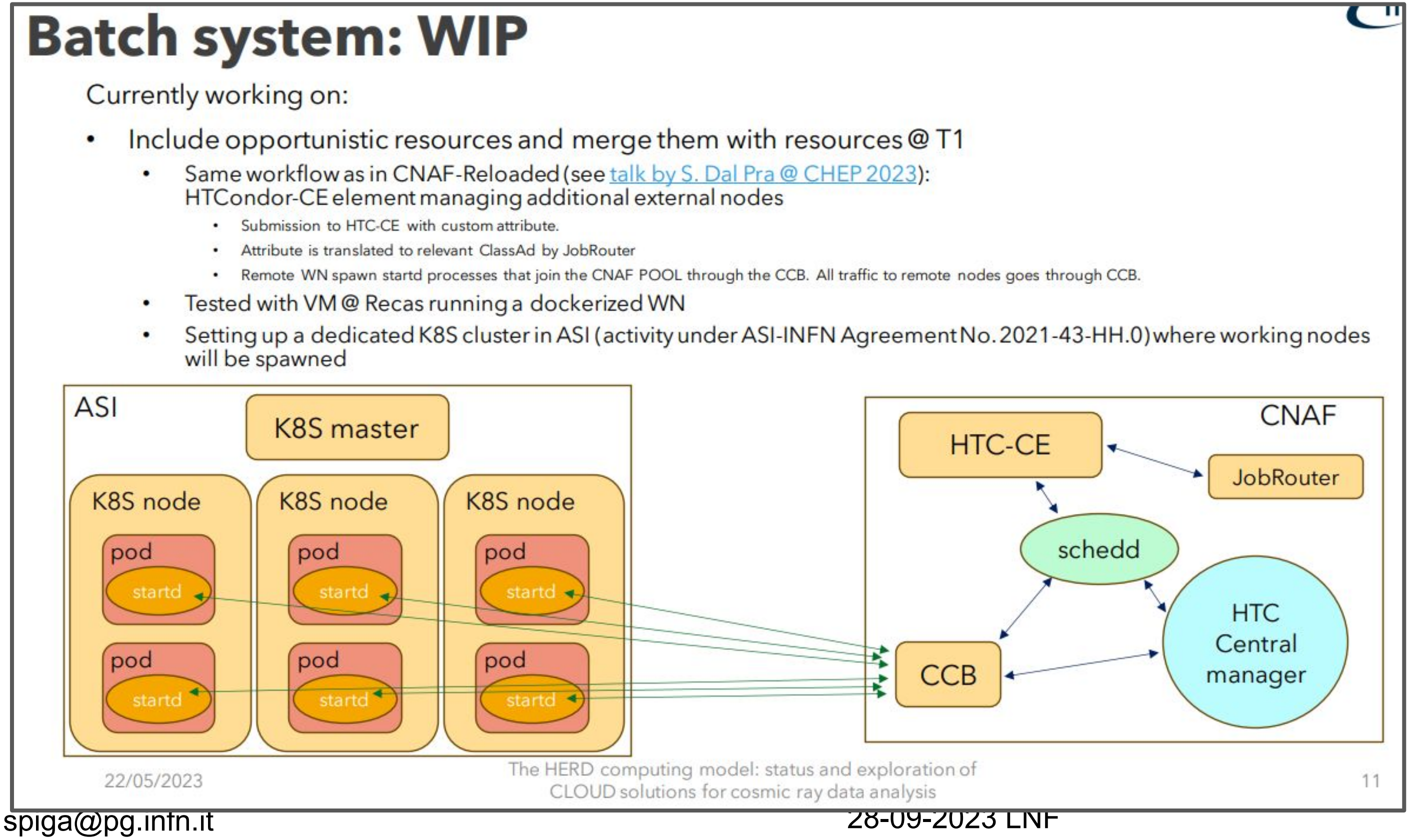
Preselection

	Legacy	RDF (O2)
Overall time	3h 40min	25min
Overall rate	693 Hz	7306 Hz
Event-loop rate	721 Hz	8473 Hz
Overall network read	488 GB	371 GB
Average RSS per-node	Ca. 13 GB	Ca. 17 GB

Postselection

	Legacy	RDF
Overall time	0.25h	0.08h
Overall rate	306 Hz	855 Hz
Event-loop rate	412 Hz	1976 Hz
Overall network read	11 GB	10 GB
Average RSS per-node	Ca. 1 GB	Ca. 15 GB

Another nice R&D…



Batch system: WIP

Currently working on:

- Include opportunistic resources and merge them with resources @ T1
  - Same workflow as in CNAF-Reloaded (see talk by S. Dal Pra @ CHEP 2023): HTCondor-CE element managing additional external nodes
    - Submission to HTC-CE with custom attribute.
    - Attribute is translated to relevant ClassAd by JobRouter
    - Remote WN spawn startd processes that join the CNAF POOL through the CCB. All traffic to remote nodes goes through CCB.
  - Tested with VM @ Recas running a dockerized WN
  - Setting up a dedicated K8S cluster in ASI (activity under ASI-INFN Agreement No. 2021-43-HH.0) where working nodes will be spawned

22/05/2023

The HERD computing model: status and exploration of CLOUD solutions for cosmic ray data analysis

11

N.Mori et al (see here)

So far the “easy” part… now few challenges

Let’s see now where the (we see just two, tha main one) challenges are once you need to deal with a distributed environment

Software distribution

- **Objectives:**
  - Make your runtime environment available anytime anywhere in the geo-distributed infrastructure
  - Learn about and use software portability techniques

Data access

- **Objectives:**
  - Transfer data from Place A - to place B
  - Define what data access is about (input and output)
  - Possible patterns to access data in a distributed environment

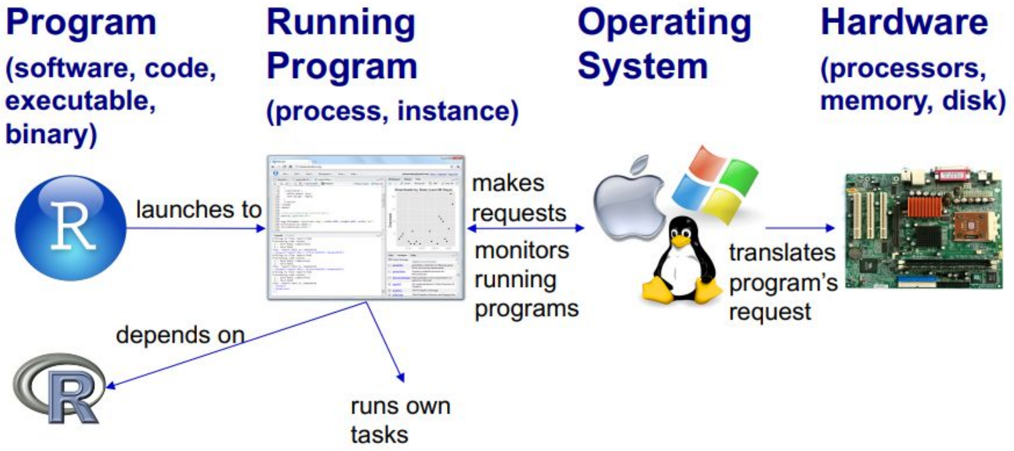
The problem

The principle

- Software is a **set of files**.
- These files have **instructions for the computer to execute**
- Moreover: Software **depends on the operating system, and other** installed **programs** (dependecies)

distributed environment implication

- Software must be able to run on **target operating system** (usually Linux). -
- Know **what else your software depends on**



Program
(software, code, executable, binary)
Running Program
(process, instance)
Operating System
Hardware
(processors, memory, disk)
launches to
makes requests
monitors running programs
translates program's request
depends on
runs own tasks

Isolate the specific software files needed for a programme and bring them along

Why it is a problem

When you are on your pc (in principle) You have full control.

- You know what you already have
- All the software you need is already installed.
- You know where everything is (mostly).
- You can add new programs when and where you want.

If you start using someone else pc (in principle) you’ve not full control.

- What’s already there?
- Is R installed? Or Python
- What about the packages you need?
- Do you know where anything is?
- Are you allowed to change whatever you want?

The solution: Software Portability

Take your software with you Install it anywhere and Run anywhere

Run “anywhere” by:

- bringing along the (Linux-compatible) software files you need…
- to a location you can access/control…
- telling the command line where that location is…
- and using it to run your code.

Easiest case:

- Job written in a single compiled language (Link Statically)

Harder case:

- jobs with a combination of languages, libraries, scripts

Containers (recap)

Containers are a tool for **capturing an entire job "environment"** (software, libraries, operating system) into an "image" that can be used again.

Why use containers instead of compiling/installing code

- **Permissions**: Software that can't be moved, do files or libraries have to be at a specific path?
- **Complex installations**: software that has a lot of dependencies or components.
- **Sharing with others**: one container can be used by a whole group that's doing the same thing.
- **Running on different systems**: The same container can run on Linux, Mac and Windows
- **Reproducibility**: save a copy of your environment.

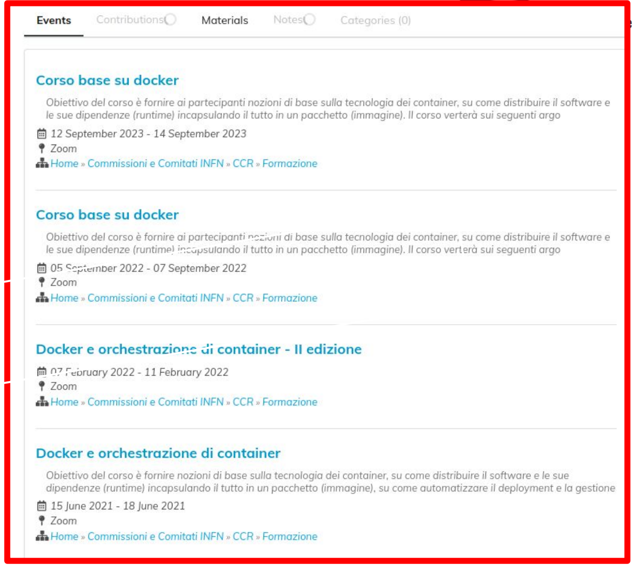
Several trainings organized by M.Antonacci

Container (cont)

To use a container as your software portability tool, need to either:

- **Find a pre-existing container with what you need.**
    - Dockerhub is your friend
- **Build your own container**

How to..



Corso base su docker

Obiettivo del corso è fornire ai partecipanti nozioni di base sulla tecnologia dei container, su come distribuire il software e le sue dipendenze (runtime) incapsulando il tutto in un pacchetto (immagine). Il corso verterà sui seguenti argo

12 September 2023 - 14 September 2023
Zoom
Home » Commissioni e Comitati INFN » CCR » Formazione

Corso base su docker

Obiettivo del corso è fornire ai partecipanti nozioni di base sulla tecnologia dei container, su come distribuire il software e le sue dipendenze (runtime) incapsulando il tutto in un pacchetto (immagine). Il corso verterà sui seguenti argo

05 September 2022 - 07 September 2022
Zoom
Home » Commissioni e Comitati INFN » CCR » Formazione

Docker e orchestrazione di container - II edizione

07 February 2022 - 11 February 2022
Zoom
Home » Commissioni e Comitati INFN » CCR » Formazione

Docker e orchestrazione di container

Obiettivo del corso è fornire nozioni di base sulla tecnologia dei container, su come distribuire il software e le sue dipendenze (runtime) incapsulando il tutto in un pacchetto (immagine), su come automatizzare il deployment e la gestione

15 June 2021 - 18 June 2021
Zoom
Home » Commissioni e Comitati INFN » CCR » Formazione

Two common container systems:

docker

widely used container system for HPC

APPTAINER

Apptainer

Another path… pre-existing software

Like if you could execute your programme in a distributed environment, **working in the assumption all the needed software is pre-installed everywhere** (in the same location/PATH)

Ideally:

- you distribute software as simple as you install it in your laptop
- Find it, everywhere, same path, same library no dependency issue
- Keep everything always up to date and in synch

Access the same software as if it is in the local working station

This is possible: CernVM-FS

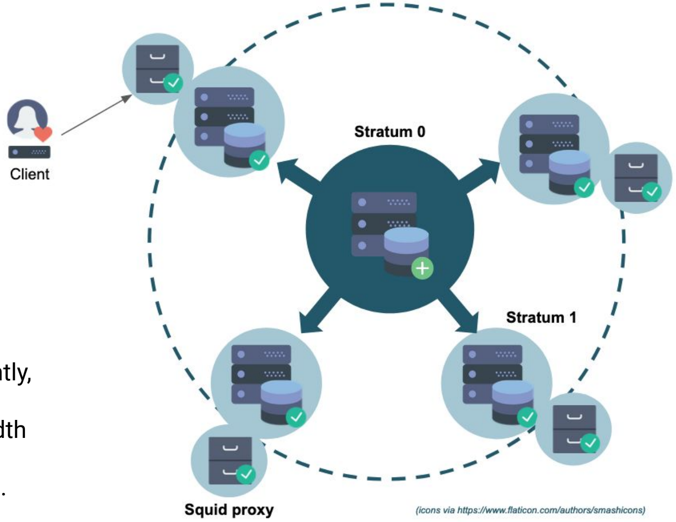
The CernVM File System (CernVM-FS) **provides a scalable and reliable software distribution service**, which is implemented as a **read-only POSIX filesystem in user space** (a FUSE module).

- Files and directories are hosted on standard web servers and mounted in the universal namespace **/cvmfs.**

How is made (from 10km far)

- the central *Stratum 0* server which hosts the filesystem;
- the *Stratum 1* replica servers, and the associated *proxies*;
- the *client* accessing the filesystem provided via CernVM-FS.

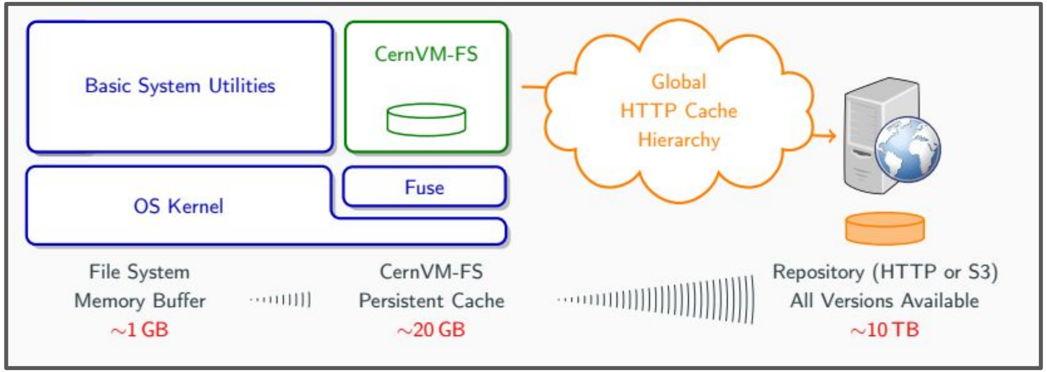
A Squid proxy caches content that has been accessed recently, and helps to reduce bandwidth and improve response times.



(icons via https://www.flaticon.com/authors/smashicons)

How it works from a user perspective

Populate and propagate new and updated content

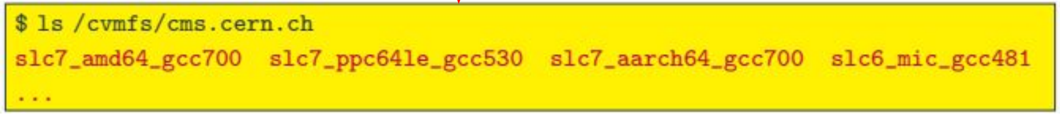


A few “software librarians” can publish into **/cvmfs**

All content in /cvmfs is cryptographically signed

Transactional writes as in git commit/push

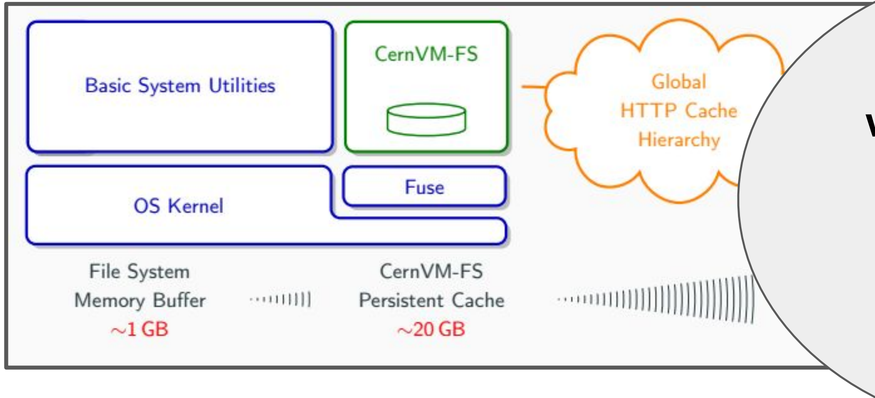
CernVM-FS provides uniform, consistent, and versioned **POSIX file system access to /cvmfs**



```
$ ls /cvmfs/cms.cern.ch
slc7_amd64_gcc700  slc7_ppc64le_gcc530  slc7_aarch64_gcc700  slc6_mic_gcc481
...
```

How it works from a user perspective

Populate and propagate new and updated content



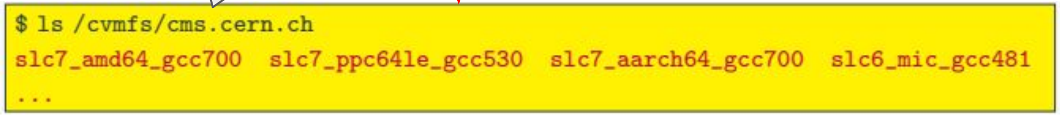
Basic System Utilities
CernVM-FS
Global HTTP Cache Hierarchy
OS Kernel
Fuse
File System Memory Buffer
~1 GB
CernVM-FS Persistent Cache
~20 GB

This is true and working on grids, clouds, supercomputers and end user laptops

A few “software librarians” can
ublish into **/cvmfs**

ntent in /cvmfs is
graphically signed

sactional writes as in git
mmit/push



```
$ ls /cvmfs/cms.cern.ch
slc7_amd64_gcc700   slc7_ppc64le_gcc530   slc7_aarch64_gcc700   slc6_mic_gcc481
...
```

CernVM-FS provides uniform, consistent, and versioned **POSIX file system access to /cvmfs**

Recap

CernVM-FS is a ***read-only*** filesystem for those who access it : This is you

Who administer the **stratum 0** is able to add or change the contents.

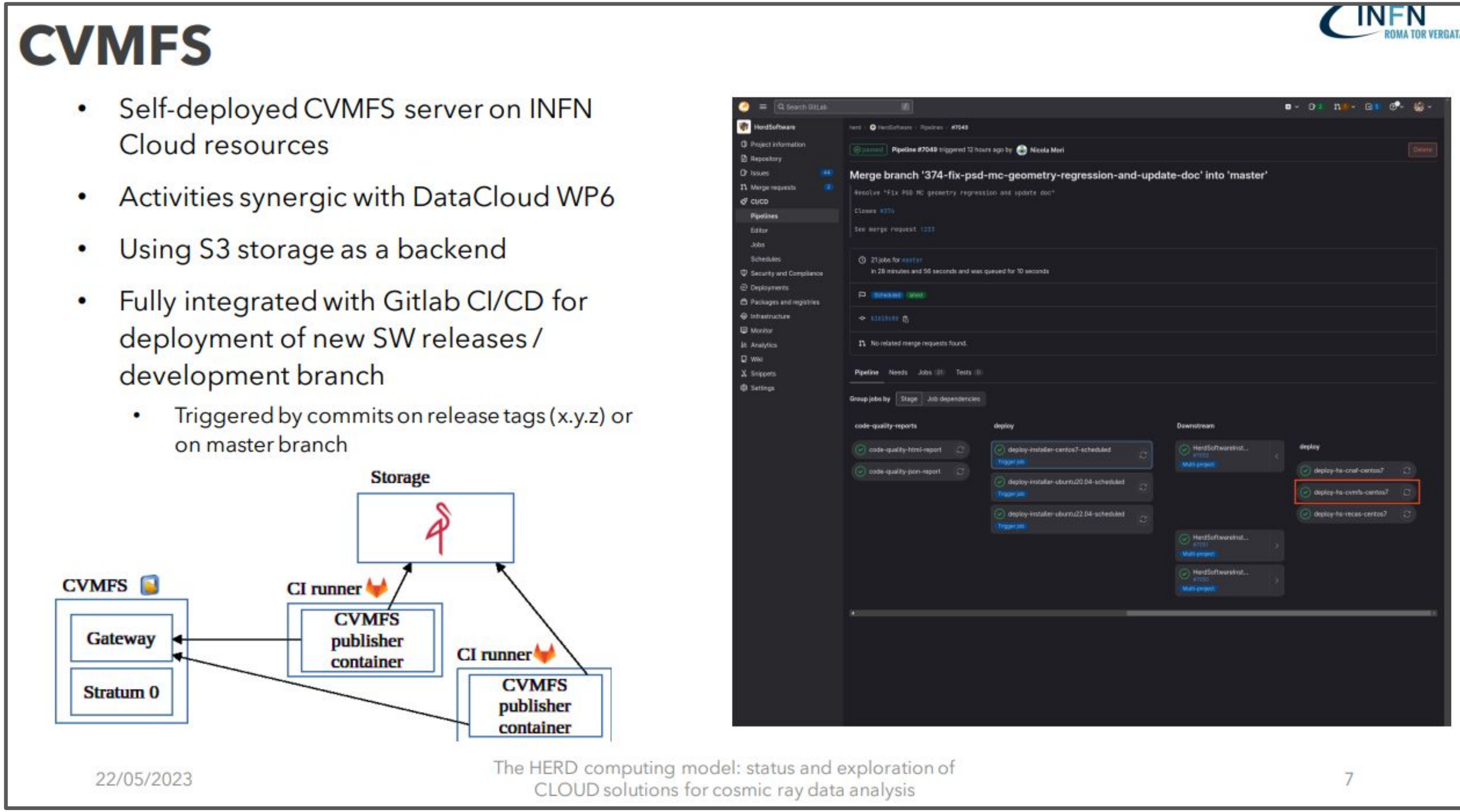
- Someone should allow you (or your friend) to write on a stratum 0

What is the most important message to remember so far:

- You know it exist and what it does
- You ask for support if you think you need it for your research
    - To who? **A possibility is INFN-Cloud**

Few references: https://cernvm.cern.ch/fs/ ; https://github.com/cvmfs/cvmfs

An Example from HERD experiment



CVMFS

- Self-deployed CVMFS server on INFN Cloud resources
- Activities synergic with DataCloud WP6
- Using S3 storage as a backend
- Fully integrated with Gitlab CI/CD for deployment of new SW releases / development branch
    - Triggered by commits on release tags (x.y.z) or on master branch

Pipeline #7049 triggered 12 hours ago by Nicola Mori

Merge branch '374-fix-psd-mc-geometry-regression-and-update-doc' into 'master'

```
Resolve "Fix PSD MC geometry regression and update doc"

Closes #374

See merge request !233
```

21 jobs for master
in 28 minutes and 56 seconds and was queued for 10 seconds

No related merge requests found.

22/05/2023

The HERD computing model: status and exploration of CLOUD solutions for cosmic ray data analysis

7

Container and CVMFS

Can works together, ideally use containers for isolation and orchestration, and CVMFS for distribution

- Container are fine for distribution but no necessarily ideal

If **/cvmfs** is available on the host

- Bind mount from host to container as an external volume (Davide told you about this)

Container Images can be on /cvmfs, particularly suitable for **large scale distribution** but not only... I think that this become a ideal configuration

- Example1: **unpacked at CERN** is a service that unpacks Docker images and makes them available via a dedicated CVMFS area. Images will be automatically synchronised from the image registry to the CVMFS area within a few minutes whenever you create a new version of the image.
- Example2: **singularity.opensciencegrid.org**
  - /cvmfs/singularity.opensciencegrid.org/centos/pyt hon-34-centos7:latest

We (at INFN) are working for you :)

INFN-Cloud portfolio of solution will be extended with two main services

1. Automate the CVMFS stratum 0 setup and configuration:

- Translation: user doesn't need to know how to setup such a system. Push a button, the cloud system provides a personal server. In turn you become librarian of yourself/your group
  - Use the client everyware

2. Enable everybody to be librarians hiding completely the interaction with CMVFS stratum 0

- Dropbox like approach

WORK IN PROGRES

Wrap-up

Create/find software package. Account for all dependencies, files, and requirements and then options:

- download pre-compiled code
- compile your own on the node
- create/find a container
- use CVMFS

In addition, in the real life script to set up the environment on a remote host often is needed,.

- I.e. you might need to play with software paths etc.

Data handling

"**Input**" includes any files needed for the job to run

- executable
- **data to process**
- (and software)

"**Output**" includes **any files produced** that you need to come back

- Output results
- Log, error

Input data handling: possible choices

Data can be **shipped to the execution host** (a node of the distributed environment) together with the executable

- **Via input sandbox**
- This is ok for **testing purposes and more in general when you deal with (very) small input data**, otherwise you will hit:
  - Hardware transfer limits
  - Hardware storage limits (spool)
  - Network bottleneck

Data can be **pre-placed** on the execution host

- This can be done manually by end users (scp, rsync). Ineffective and time consuming, error prone
- So what? The answer is: **Data Management**

A path forward: Download or stream data

You are aware of cloud storage

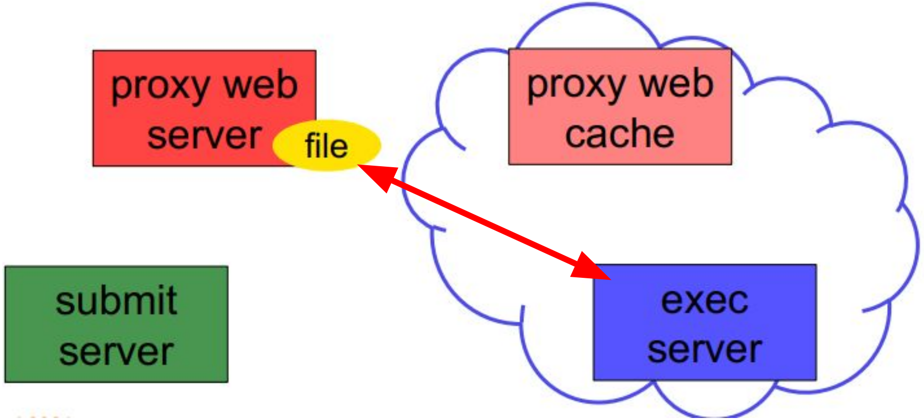
- Other protocols are also possible as well as other technologies

Data (files) **can be placed onto a Cloud Storage** and programme (execution servers) get data from there

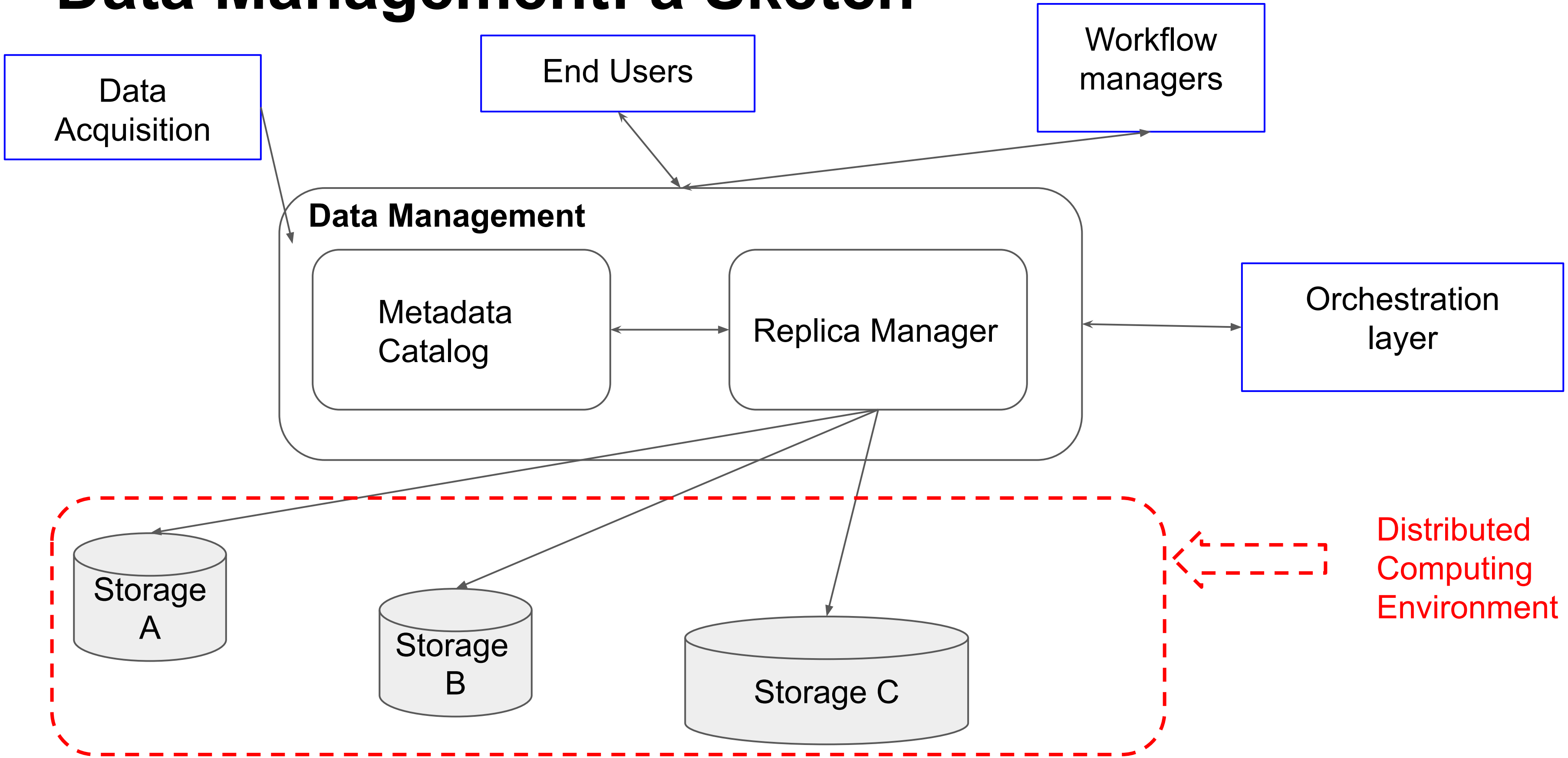
- Simple example the presigned url

Once data are in a storage:

- **Lazy download** to the exec host
- **Remote data read** (stream)



Data Management: a Sketch



Strategia

Rucio+FTS è stato scelto come **punto di partenza**, per quello che conosciamo in **esperimenti LHC (de-facto standard)** ed ESCAPE ci ha insegnato essere **applicabile ad altre comunità.**

Il focus e quindi sull'**integrazione di soluzioni che già conosciamo** facendo in modo da soddisfare le nostre esigenze specifiche.

Particle Physics / Nuclear Physics
Astrophysics / Astroparticle Physics
FAIR
HiLumi HL-LHC PROJECT
SKAO
cta cherenkov telescope array
LSST
ESCAPE
European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures
VIRGO
KM3NeT
ESO
JIVE
EST european solar telescope
5

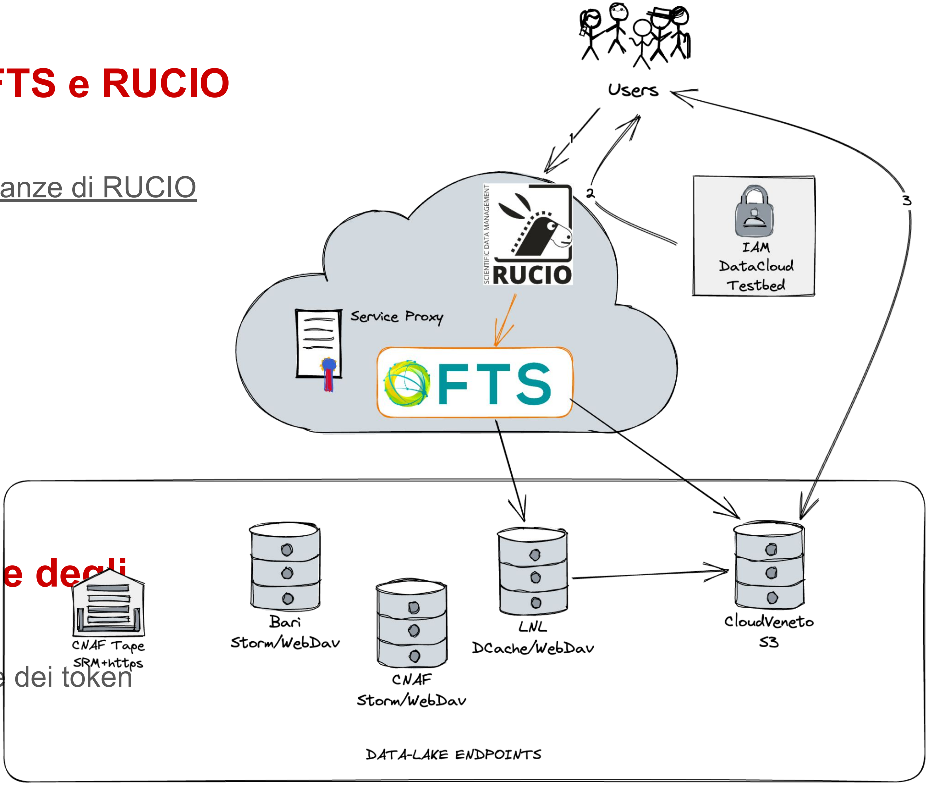
E fondamentale quindi avere un testbed che permetta di confrontarsi con gli utenti e verificare che queste esigenze siano soddisfatte.

In sintesi:

Obiettivo	Avviare una sperimentazione al fine di implementare un sistema di data lake a livello di infrastruttura nazionale
Target	In particolare i piccoli esperimenti (WLCG è già autonomo) L'infrastruttura nazionale distribuita
Strumenti identificati	Possiamo partire dagli strumenti a noi noti (quelli di WLCG in particolare) e quindi a servizi quali RUCIO e FTS

Dove siamo

- **Instanziato su risorse cloud i server IAM, FTS e RUCIO dedicati al testbed DataCloud**
  - FTS può essere mantenuto “centralmente” e servire istanze di RUCIO multiple
  - IAM per AuthZ fine gestita centralmente, vedi dopo
- **Federato 5 siti con storage eterogenei:**
  - Uno storage con protocollo S3 su ceph @CloudVeneto
  - Tre storage con protocollo WebDav
    - Due basati su STORM (CNAF, Bari)
    - Uno su dCache (LNL)
  - Un endpoint tape @CNAF
- **Automatizzato la registrazione e la gestione degli utenti via IAM**
  - AuthZ gestita centralmente, i siti autorizzano sulla base dei token rilasciati



WP6 (M.Sgaravatto)