



# Introduction to Cloud Computing, VRE Data Analysis Tools (JupyterLab, CCP Method Executions)

**IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System**  
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-  
Mission 4 “Education and Research” - Component 2: “From research to business” - Investment  
3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”



# What is Cloud Computing



*“Cloud computing is 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.”*

- USA National Institute of Standards and Technology
- Resources: networks, servers, storage, application and services
- Minimal management effort/interaction

## **...our definition of Cloud Computing:**

*“Cloud computing allows users to run their scientific and collaborative applications in a web-based environment without the need to install software locally or worry about managing the underlying infrastructure. This approach facilitates data sharing, distributed processing, and collaboration among research groups.”*

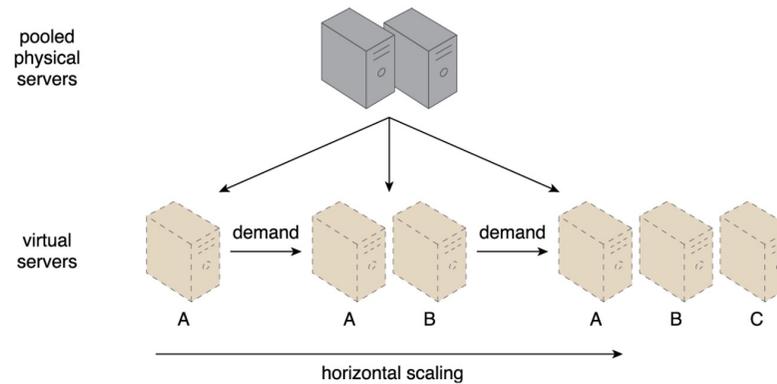
# The Benefits of Cloud Computing

- ④ Reduction or outright elimination of up-front IT investments, namely hardware and software purchases and ownership costs.
- ④ On-demand access to pay-as-you-go computing resources on a short-term basis (such as processors by the hour), and the ability to release these computing resources when they are no longer needed.
- ④ The perception of having unlimited computing resources that are available on demand, thereby reducing the need to prepare for provisioning.
- ④ The ability to add or remove IT resources at a fine-grained level, such as modifying available storage disk space by single gigabyte increments.
- ④ Abstraction of the infrastructure so applications are not locked into devices or locations and can be easily moved if needed.

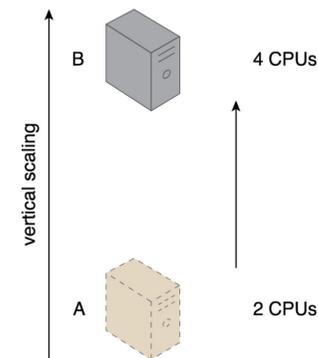
# IT-Resources Scaling

🌐 Scaling represents the ability of the IT resource to handle increased or decreased usage demands.

🌐 Horizontal Scaling: scaling out and scaling in

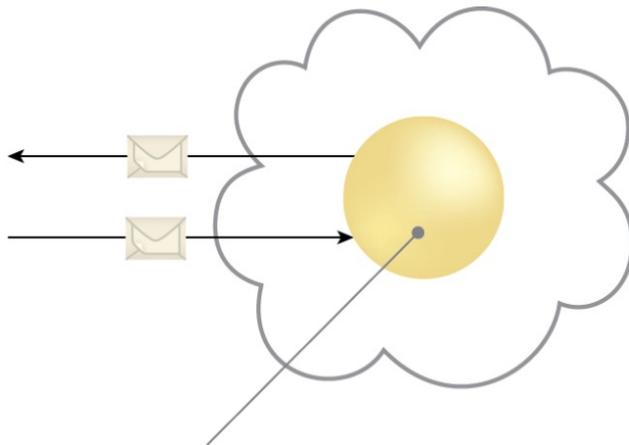


🌐 Vertical Scaling: scaling up and scaling down

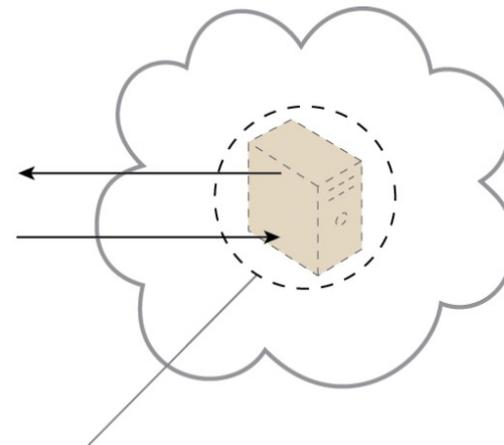


# What is a Cloud Service

- 🌐 A cloud service is an IT resource that is made remotely accessible via a cloud.
- 🌐 It can be as simple as a Web-based software program with a technical interface invoked via the use of a messaging protocol
- 🌐 Or as a remote access point for administrative tools or larger environments and other IT resources.



remotely accessed Web service  
acting as a cloud service



remotely accessed virtual server  
acting as a cloud service

# Delivery Models for Cloud

🌐 A Cloud delivery model is a specific, pre-packaged combination of IT resources offered by a cloud provider

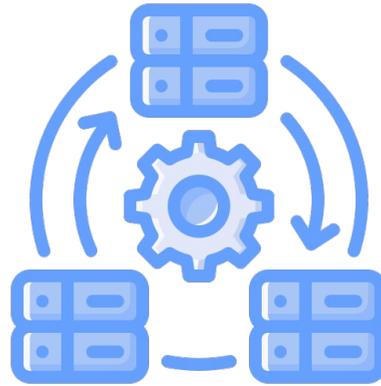
🌐 Three are the most common delivery models:

- ✓ Infrastructure-as-a-Service (IaaS)
- ✓ Platform-as-a-Service (PaaS)
- ✓ Software-as-a-Service (SaaS)

🌐 Function-as-a-Service (FaaS) as evolution of the IaaS and SaaS

- ✓ Allows consumer to deploy “functions” that are executed in (Docker) containers on the cloud provider infrastructure
- ✓ Very useful for the provider: fine-grain balancing of the resources
- ✓ Very useful for the consumer: easy to develop little functions that can be grouped and interact directly with the cloud services

## D4Science - Analytics Frameworks



## ANALYTICS Computing FFramework

# Analytics Frameworks

JupyterLab

Rstudio 4+

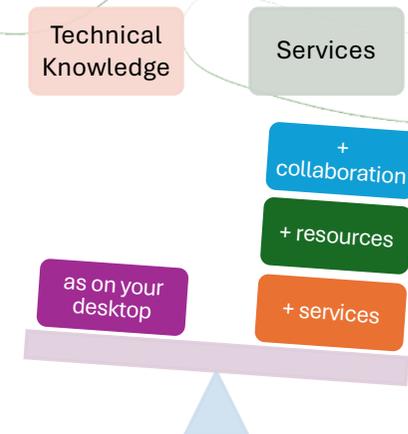
Analytics Engines

- Data Miner
- Cloud Computing Platform, CCP (the new engine)

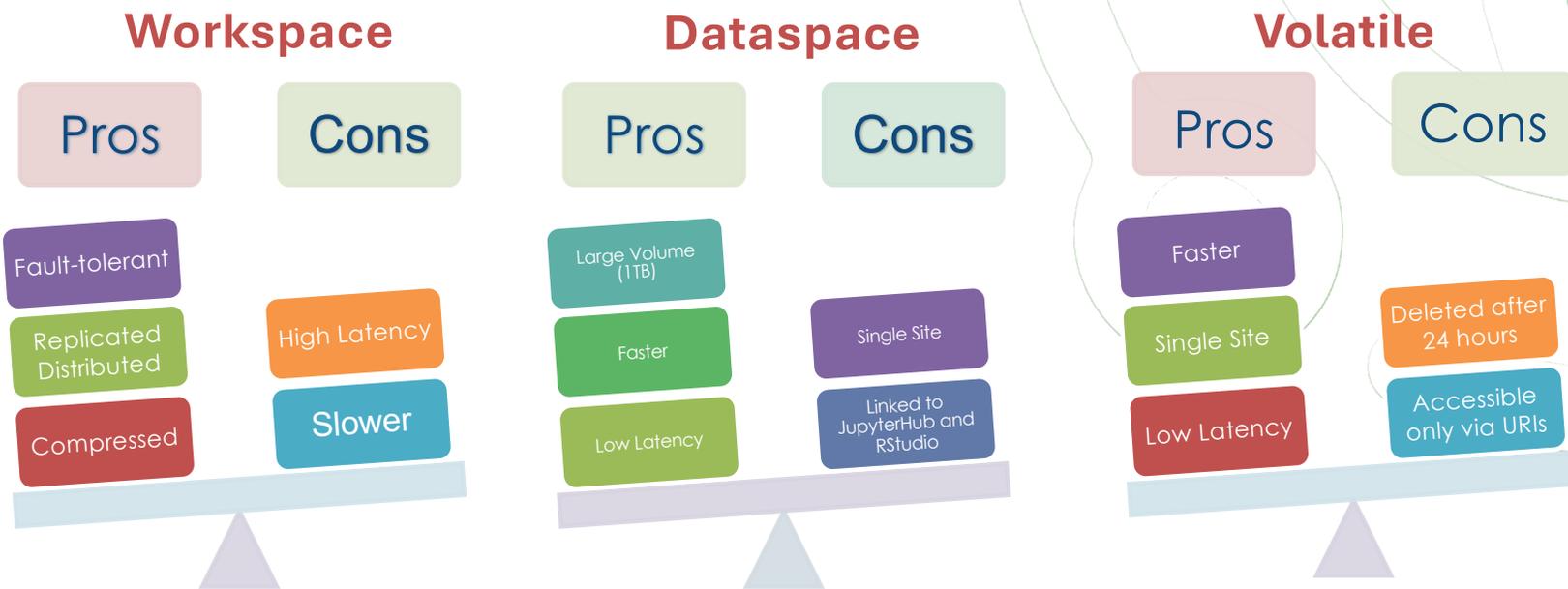
# One place for executing analysis and processes

🌐 interactive notebooks via JupyterHub and community-specific applications delivered as a Docker container extend the Analytics framework

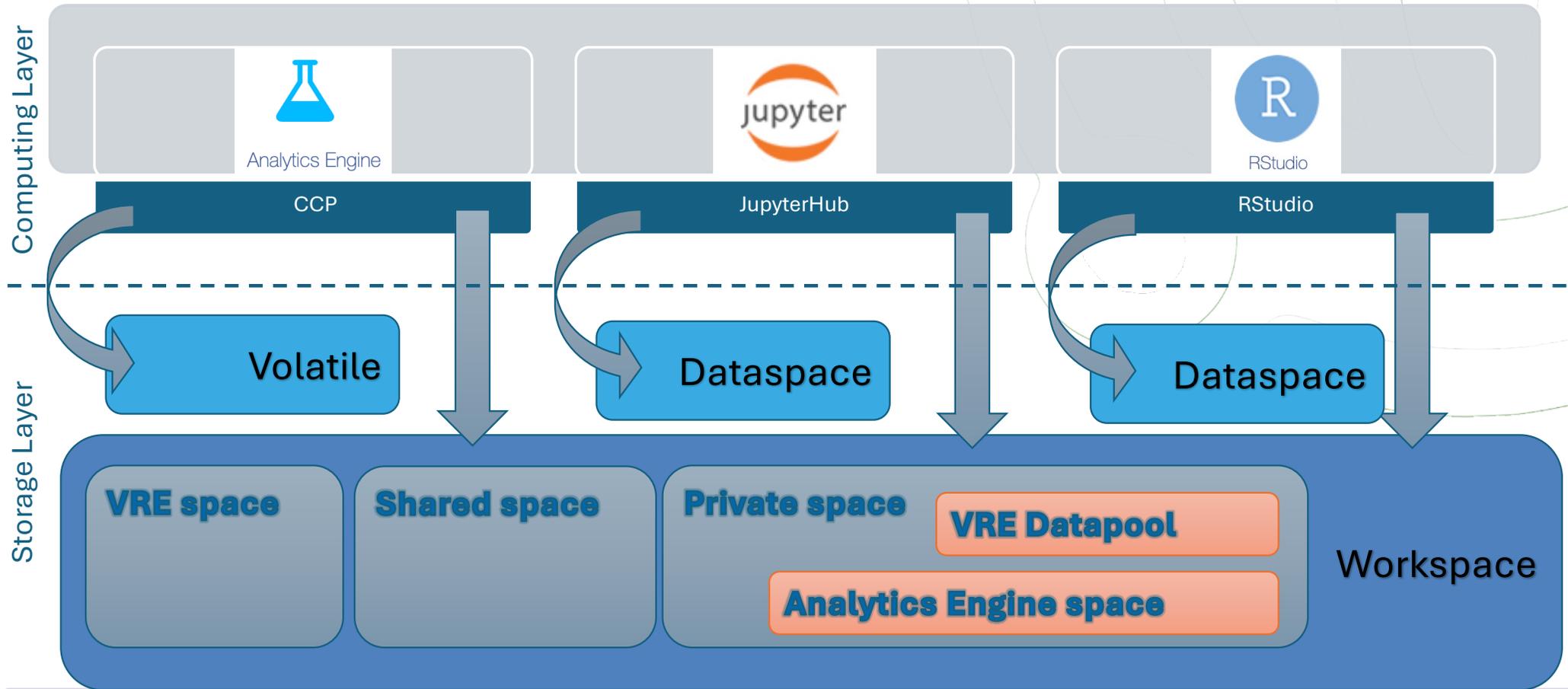
 <p>JupyterHub</p> <p>JupyterHub enables the exploitation of computational environments and resources without burdening users with installation and maintenance tasks. This JupyterHub environment is (i) preconfigured with libraries and packages to ease the execution of common data analytics tasks, and (ii) provides access to the Workspace enabling sharing of resources with other members much easier.</p>	 <p>RStudio</p> <p>RStudio provides an integrated development environment for R. It includes a console and a syntax-highlighting editor and it enables code execution. Tools for plotting are also included. This RStudio environment is (i) preconfigured with libraries and packages to ease the execution of common data analytics tasks, and (ii) provides seamless access to the Workspace enabling sharing of resources with other members much easier.</p>
  <p>Analytics Engine</p> <p>Analytics Engine (DataMiner) permits the execution of an array of analytics methods by transparently relying on distributed computing infrastructure. Executions can run either on multi-core machines or on different computational platforms, such as D4Science and other different private and commercial Cloud providers. New software can be integrated by using the dedicated Software Importer (SAI).</p>	  <p>Analytics Engine</p> <p>Analytics Engine (DataMiner) permits the execution of an array of analytics methods by transparently relying on distributed computing infrastructure. Executions can run either on multi-core machines or on different computational platforms, such as D4Science and other different private and commercial Cloud providers. New software can be integrated by using the dedicated Software Importer (SAI).</p>



# Analytics Framework – Storage Layer



# Data Analytics Options and Opportunities



# One Workspace

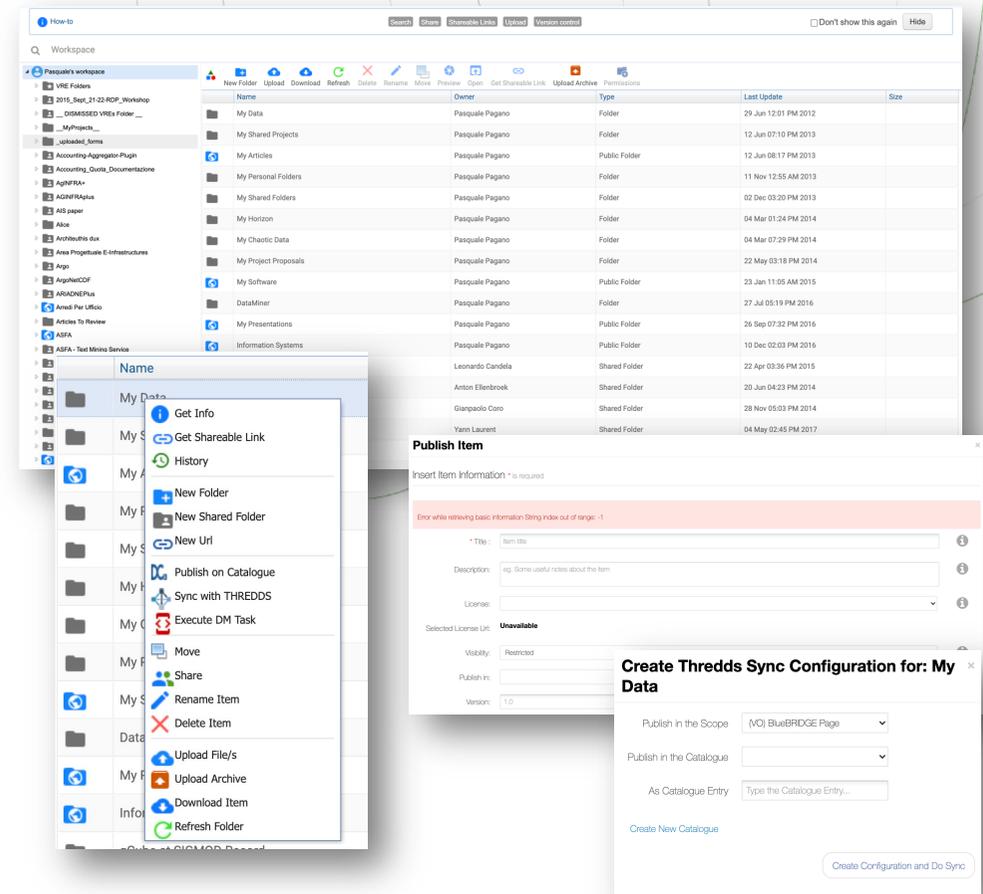


Just one working area across all different services

- VLab
- Rstudio
- JupyterHub notebooks
- Analytics Engine

Just one working area where to store data, script, notebooks, computational methods

- Private
- Shared with your colleagues
- Shared with all Vlab members



# One Workspace

Resembles a **typical file system**

with **files organised in folders,**

yet it supports

**files, datasets, software methods, workflows,  
maps, ...**



The screenshot displays the ITINERIS workspace interface. At the top, there is a navigation bar with a 'How-to' link, search, share, shareable links, upload, and version control buttons, along with a 'Don't show this again' checkbox and a 'Hide' button. Below this is a search bar containing the word 'Workspace'. The main area is divided into two panes. The left pane shows a tree view of folders under 'Pasquale's workspace', including 'VRE Folders', '2015\_Sept\_21-22-RDP\_Workshop', and many others. The right pane shows a table of files and folders with columns for Name, Owner, Type, Last Update, and Size. The table lists various folders like 'My Data', 'My Shared Projects', 'My Articles', etc., with their respective owners and last update dates. At the bottom of the interface, it shows '70525 items, 194.96 GB' and '82 items' in the current view.

Name	Owner	Type	Last Update	Size
My Data	Pasquale Pagano	Folder	29 Jun 12:01 PM 2012	
My Shared Projects	Pasquale Pagano	Folder	12 Jun 07:10 PM 2013	
My Articles	Pasquale Pagano	Public Folder	12 Jun 08:17 PM 2013	
My Personal Folders	Pasquale Pagano	Folder	11 Nov 12:55 AM 2013	
My Shared Folders	Pasquale Pagano	Folder	02 Dec 03:20 PM 2013	
My Horizon	Pasquale Pagano	Folder	04 Mar 01:24 PM 2014	
My Chaotic Data	Pasquale Pagano	Folder	04 Mar 07:29 PM 2014	
My Project Proposals	Pasquale Pagano	Folder	22 May 03:18 PM 2014	
My Software	Pasquale Pagano	Public Folder	23 Jan 11:05 AM 2015	
DataMiner	Pasquale Pagano	Folder	27 Jul 05:19 PM 2016	
My Presentations	Pasquale Pagano	Public Folder	26 Sep 07:32 PM 2016	
Information Systems	Pasquale Pagano	Public Folder	10 Dec 02:03 PM 2016	
gCube at SIGMOD Record	Leonardo Candela	Shared Folder	22 Apr 03:36 PM 2015	
IMB5	Anton Ellenbroek	Shared Folder	20 Jun 04:23 PM 2014	
Architeuthis dux	Gianpaolo Coro	Shared Folder	28 Nov 05:03 PM 2014	
2015_Sept_21-22-RDP_Workshop	Yann Laurent	Shared Folder	04 May 02:45 PM 2017	
Area Progettuale E-Infrastructures	Franco Zoppi	Shared Folder	13 Nov 05:02 PM 2015	

# Workspace Folder Types

	<b>Private Folders</b> <ul style="list-style-type: none"><li>• Only you can access the files you upload</li></ul>
	<b>Shared Folders</b> <ul style="list-style-type: none"><li>• You decide who can access the files (Selective Sharing)</li></ul>
	<b>VRE Folders</b> <ul style="list-style-type: none"><li>• All VRE members can access the files (Controlled Sharing)</li></ul>
	<b>Public Folders</b> <ul style="list-style-type: none"><li>• Anyone with the link can access the files</li></ul>

# D4Science – Analytics Framework



- 🌐 Offers a User-friendly *data analytics platform* with on-demand and scalable *computing resources* on the Cloud;
- 🌐 Allows to *import* and *execute* methods, to *collaborate* by *sharing* methods, method executions, method inputs and results with colleagues;
- 🌐 Fosters better collaboration, *enabling teams to work together* from anywhere with an internet connection and access the same data and applications;



# How does the D4Science Analytics Framework support Open Science?

## What is Open Science

*An approach to the scientific process based on cooperative work and ways of disseminating knowledge, improving accessibility to and re-usability of research outputs by using digital technologies and collaborative tools.*

SOURCE: COMMISSION  
RECOMMENDATION (EU) 2018-790 of  
2018

Uses digital technologies and collaborative tools

Enables the *repeatability* and *reproducibility* of method executions

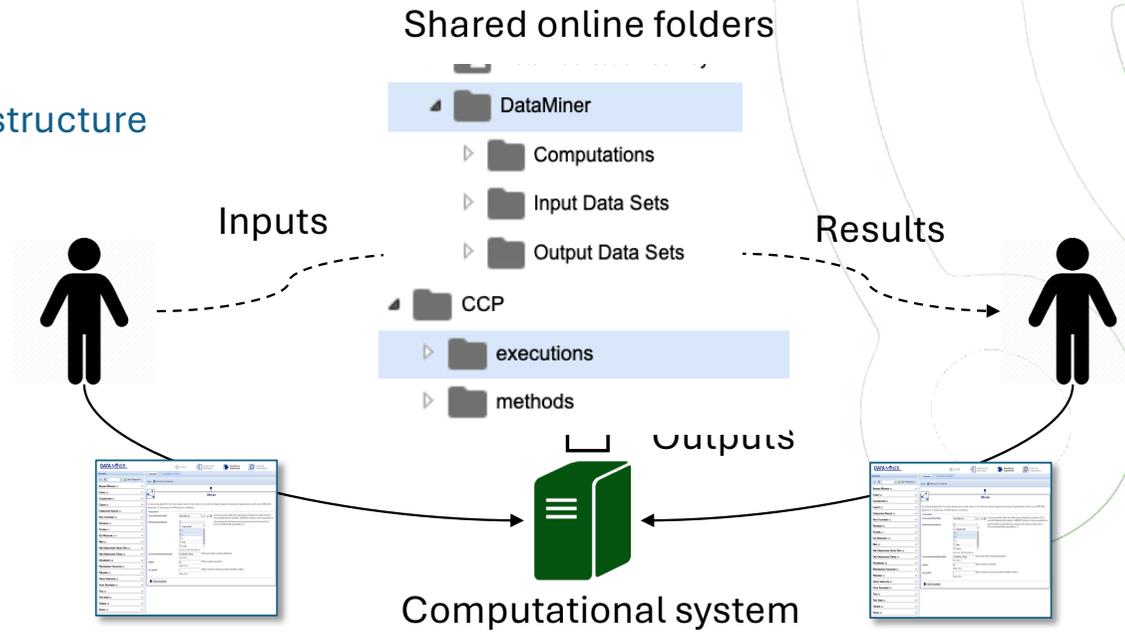
Enables *re-usability* of methods and their outputs

Improves transparency and ways of disseminating knowledge

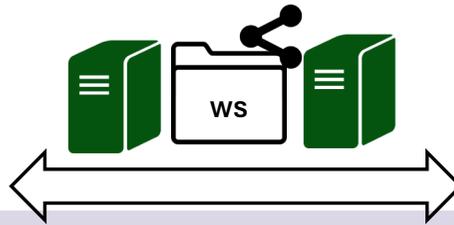
# D4Science - Collaborative experiments



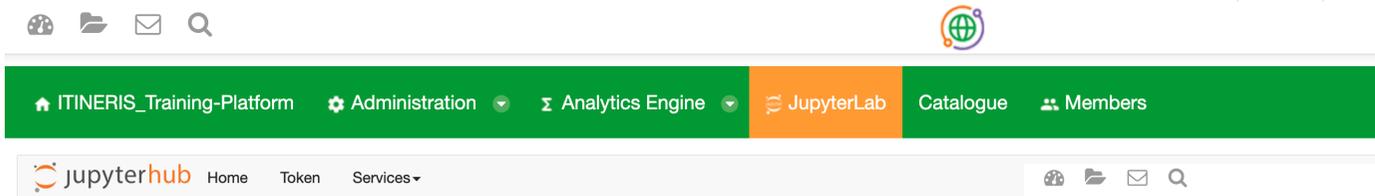
In the e-Infrastructure



Through third party software (Via OGC Standards WPS or OGC API)



# JupyterLab platform benefits



## Server Opt

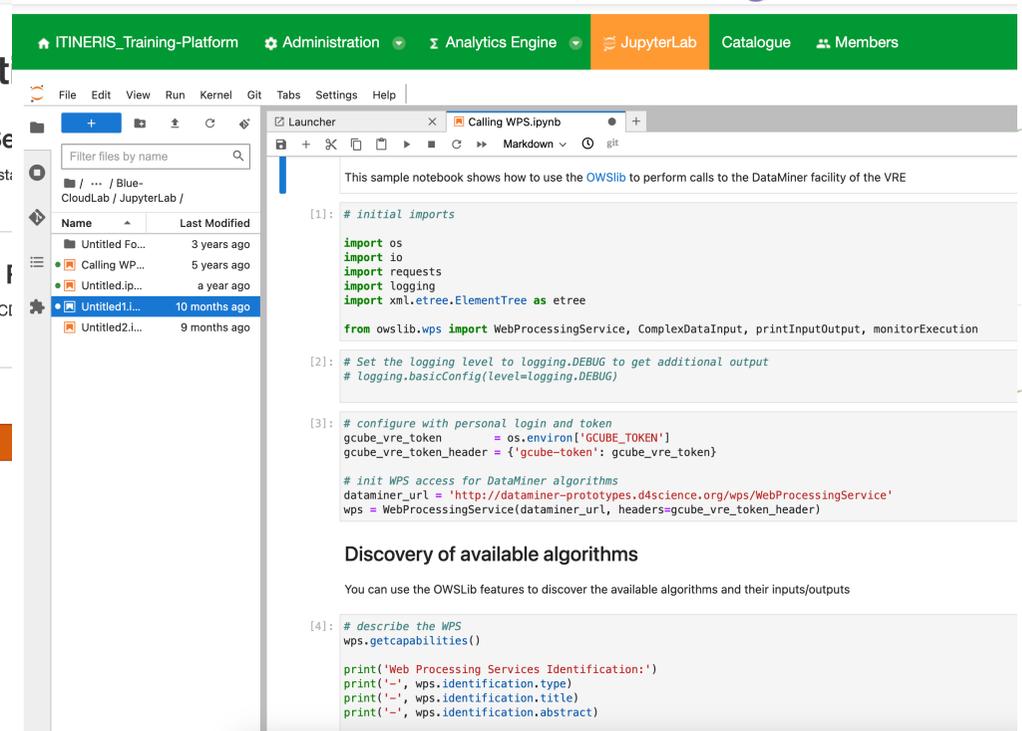
### ITINERIS Essential Variables VRE Se

This server includes Python, R and a number of libraries preinst and more ...

### VRE Default Server - 4 Cores / 16G I

Python 3.10.11, preinstalled modules pandas, gdal, scipy, netCI driven research workflows.

Start



# JupyterLab platform benefits



🌐 The Server Options can be customised differently for each Vlab / WB

- Server image and required libraries
- No. of CPU Cores
- RAM Memory avail.

The screenshot shows the JupyterLab interface with a navigation bar at the top containing the JupyterLab logo, "Home", "Token", and "Services -". Below the navigation bar is a section titled "Server Options" which lists six different server configurations, each with a radio button for selection and a brief description of the included kernels and libraries.

### Server Options

- Default Large - 4 Cores / 32G RAM**  
The Default notebook servers include Python, R, Julia, Octave and Java kernels and a number of community libraries preinstalled for Python and Julia.
- Default Medium - 4 Cores / 16G RAM**  
The Default notebook servers include Python, R, Julia, Octave and Java kernels and a number of community libraries preinstalled for Python and Julia.
- Julia 1.7.1 Medium - 4 Cores / 16G RAM**  
Julia 1.7.1 version for testing
- Julia 1.7.1 Small - 4 Cores / 8G RAM**  
Julia 1.7.1 version for testing
- R Env Large (New version) - 8 Cores / 32G RAM**  
The R Env new notebook servers include Python, R and Julia kernels and a number of community libraries preinstalled for R
- R Env Medium - 4 Cores / 16G RAM**  
The R Env notebook servers include Python, R and Julia kernels and a number of community libraries preinstalled for R

## Rstudio platform benefits

# RStudio

- RStudio server **no longer shared**, dedicated instance spawned on demand
- Each VLab can define its RStudio servers configuration, i.e. select packages

# Rstudio platform benefits



The Server Options can be customised differently for each Vlab / WB

- Server image and required libraries
- No. of CPU Cores
- RAM Memory available.

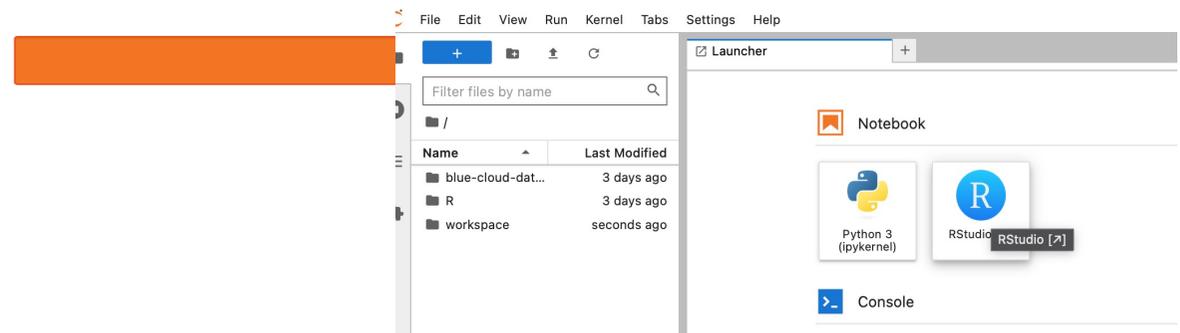
## Server Options

**RStudio v4 Large - 8 Cores / 32G RAM**

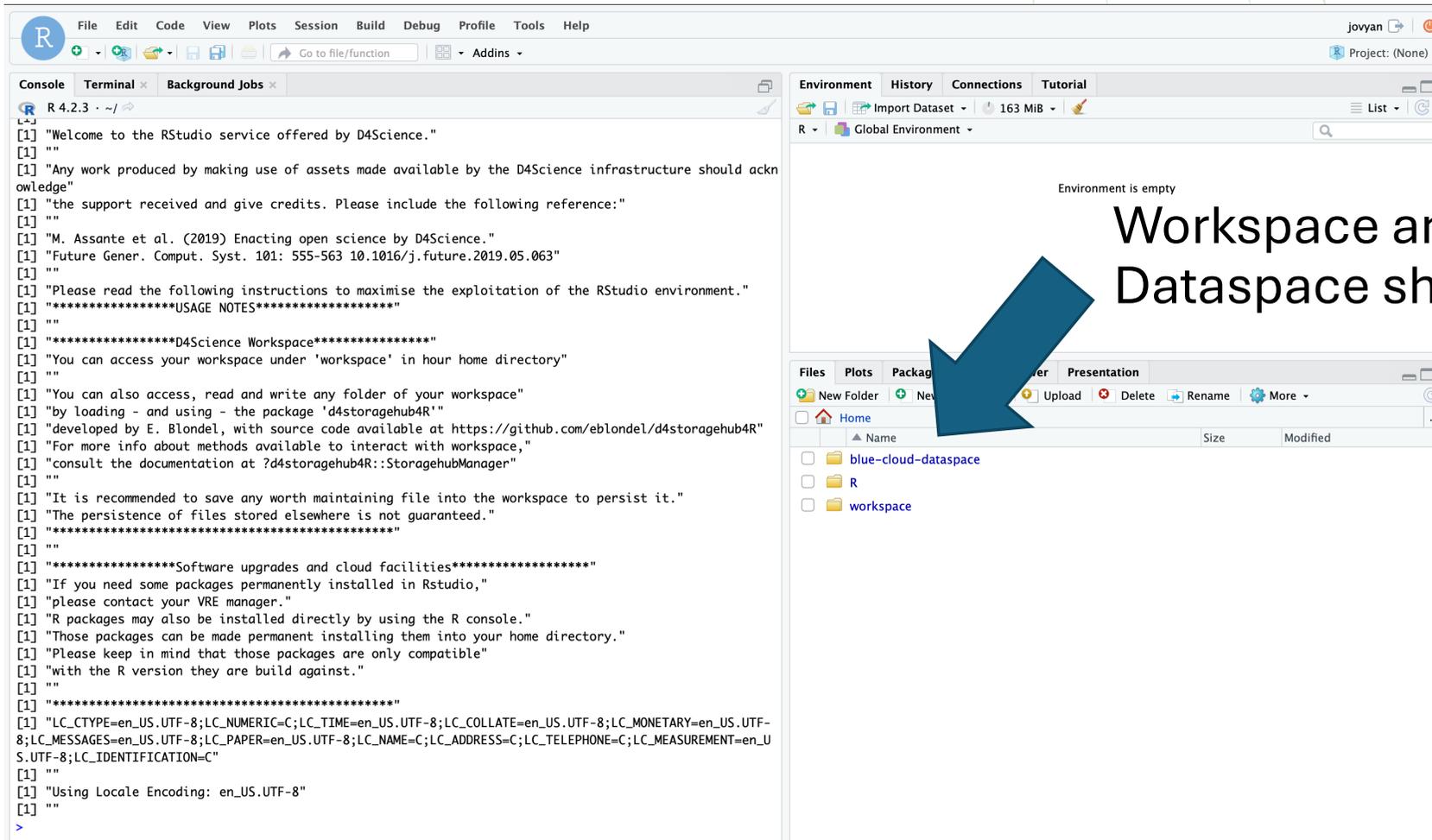
This RStudio server includes community libraries preinstalled for R

**RStudio v4 Standard - 4 Cores / 8G RAM**

This RStudio server includes community libraries preinstalled for R



# Rstudio example



The screenshot shows the RStudio interface. The console on the left displays the following output:

```
R 4.2.3 · ~/ ·  
[1] "Welcome to the RStudio service offered by D4Science."  
[1] ""  
[1] "Any work produced by making use of assets made available by the D4Science infrastructure should acknowledge the support received and give credits. Please include the following reference:"  
[1] ""  
[1] "M. Assante et al. (2019) Enacting open science by D4Science."  
[1] "Future Gener. Comput. Syst. 101: 555-563 10.1016/j.future.2019.05.063"  
[1] ""  
[1] "Please read the following instructions to maximise the exploitation of the RStudio environment."  
[1] "*****USAGE NOTES*****"  
[1] ""  
[1] "*****D4Science Workspace*****"  
[1] "You can access your workspace under 'workspace' in your home directory"  
[1] ""  
[1] "You can also access, read and write any folder of your workspace"  
[1] "by loading - and using - the package 'd4storagehub4R'"  
[1] "developed by E. Blondel, with source code available at https://github.com/eblondel/d4storagehub4R"  
[1] "For more info about methods available to interact with workspace,"  
[1] "consult the documentation at ?d4storagehub4R::StoragehubManager"  
[1] ""  
[1] "It is recommended to save any worth maintaining file into the workspace to persist it."  
[1] "The persistence of files stored elsewhere is not guaranteed."  
[1] "*****"  
[1] ""  
[1] "*****Software upgrades and cloud facilities*****"  
[1] "If you need some packages permanently installed in Rstudio,"  
[1] "please contact your VRE manager."  
[1] "R packages may also be installed directly by using the R console."  
[1] "Those packages can be made permanent installing them into your home directory."  
[1] "Please keep in mind that those packages are only compatible"  
[1] "with the R version they are build against."  
[1] ""  
[1] "*****"  
[1] "LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=en_US.UTF-8;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=en_US.UTF-8;LC_IDENTIFICATION=C"  
[1] ""  
[1] "Using Locale Encoding: en_US.UTF-8"  
[1] ""  
>
```

The file explorer on the right shows the following folders:

- blue-cloud-dataspace
- R
- workspace

A blue arrow points from the text "Workspace and Dataspace shared folders" to the "workspace" folder in the file explorer.

Workspace and Dataspace shared folders

## Analytics Engines

- Data Miner
- Cloud Computing Platform, CCP

# Cloud Computing Platform Vision

3 years of work but decade long experience (D4Science)

- Presentations, interviews, minutes and write-ups

Stems from a use case based analysis ...

- 10 abstract use cases
- A sound logical and architectural model

Developed in agility with early prototypes

Embraces heterogeneity

- supporting as many as possible different libraries, languages, tools, interfaces, services
- keep being manageable

# CCP Main Features

Container based – isolated runtime

Multiple execution infras on the same Vlab, selectable by users

Per-method Web UI and REST API access  
via Standard

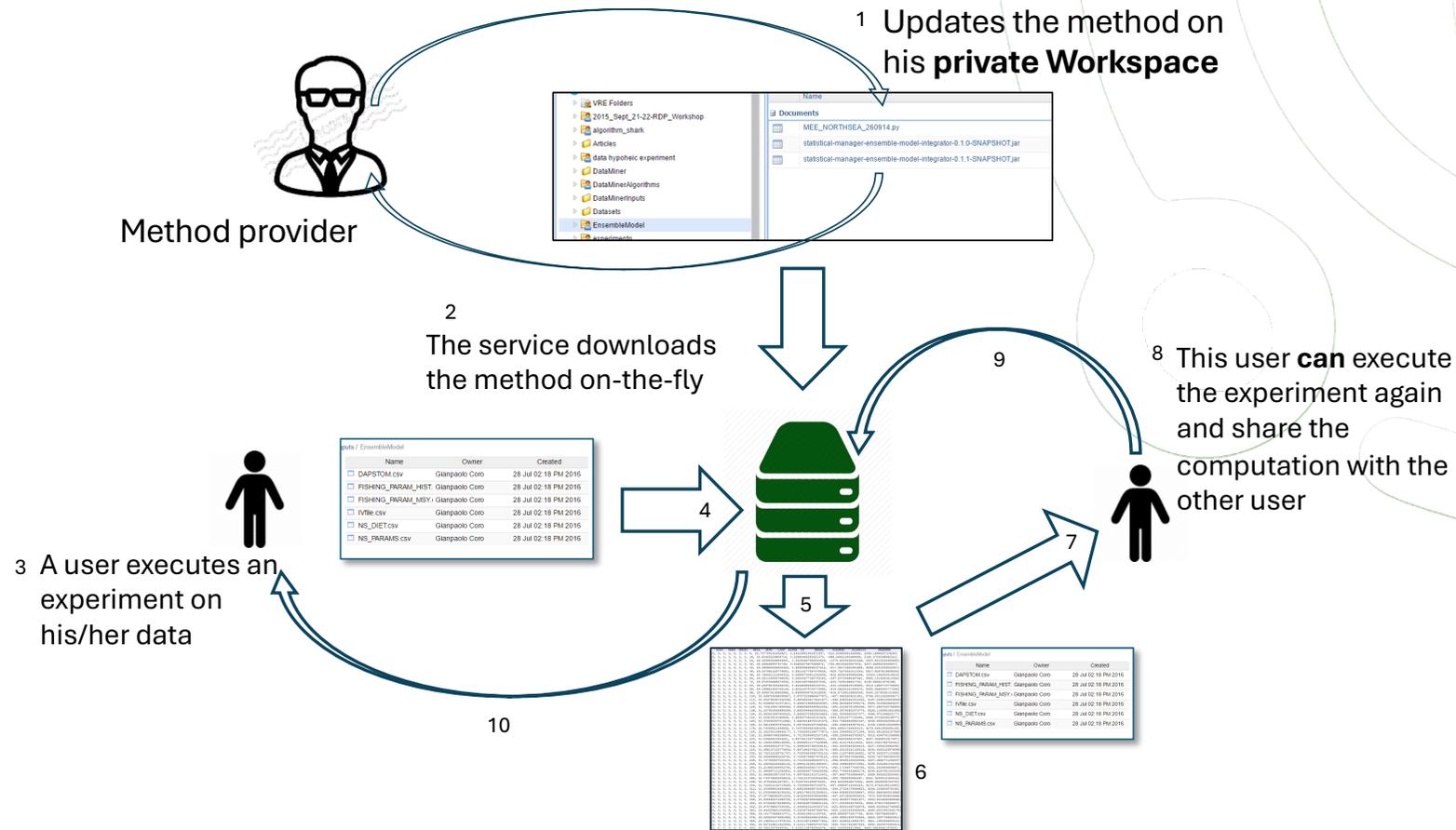
Method import support via a dedicated tool supporting any prog. language

“*out-of-the-box*” methods **as-a-Service** and automatic code generation for  
Python, R, Julia, IPy notebooks

Provenance management



# CCP Security and privacy



# A note on CCP provenance management

- 🌐 In our context is ***the lineage and processing history of a research data product, and the record of the processes that led to it;***
- 🌐 Provenance plays an important role in Data Science;
  - Experiments are often developed iteratively involving multiple executions with different versions of data sources, accessing multiple applications and infrastructure services;
- 🌐 CCP adopts the ***the PROV Ontology (PROV-O)*** expresses the PROV Data Model using the OWL2 Web Ontology Language (OWL2).

*“Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.”*

**W3C Recommendation** <https://www.w3.org/TR/prov-dm/>

```

1 <prov:document
2   xmlns:prov="http://www.w3.org/ns/prov#"
3   xmlns:d4s="http://d4science.org/#"
4   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6 <prov:activity id="activity-f07872aa-b783-4b4e-8512-95545aec7a94">
7   <prov:startTime>2024-03-26T07:53:17.376Z</prov:startTime>
8   <prov:endTime>2024-03-26T07:56:03.386Z</prov:endTime>
9   <prov:type xsi:type="xsd:QName">d4s:computation</prov:type>
10  <prov:softwareAgent prov:id="d4s:sobigdata.d4science.org"/>
11  <prov:person prov:id="d4s:andrea.rossi"/>
12  <prov:entity prov:id="d4s:operator_name">
13    <prov:value xsi:type="xsd:string">SimpleImageClassifier</prov:value>
14  </prov:entity>
15  <prov:entity prov:id="d4s:operator_version">
16    <prov:value xsi:type="xsd:string">1.0.1</prov:value>
17  </prov:entity>
18  <prov:entity prov:id="d4s:operator_description">
19    <prov:value xsi:type="xsd:string">A simple image classifier with parametrizable url to input picture compatib
20  </prov:entity>
21  <prov:entity prov:id="d4s:VRE">

```

# CCP Methods, editor, and execution

Method:  
Script, Algorithm  
Software

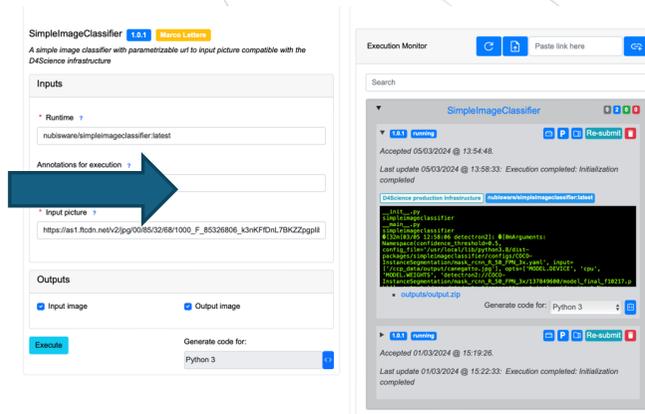


Importing phase



**METHOD**  
**EDITOR**

Resulting Web UI & Web API



**METHOD**  
**EXECUTIONS**

# CCP Methods, editor, and execution in the workspace

## Web UI / Web API

SimpleImageClassifier **1.0.1** **Marco Lettore**

A simple image classifier with parametrizable url to input picture compatible with the D4Science infrastructure

**Inputs**

**Runtime**

nubisware/simpleimageclassifier:latest

**Annotations for execution**

Annotations for execution

**Input picture**

https://as1.ftcdn.net/v2/jpg/00/85/32/68/1000\_F\_85326806\_K3nKFIDnL7BKZZpgpIlt

**Outputs**

Input image  Output image

**Execute**

Generate code for: Python 3

Execution Monitor

Search

SimpleImageClassifier **1.0.1** **running**

Accepted 05/03/2024 @ 13:54:48.

Last update 05/03/2024 @ 13:58:33: Execution completed: Initialization completed

```
D4Science production Infrastructure nubisware/simpleimageclassifier:latest
__init__.py
simpleimageclassifier
__main__.py
simpleimageclassifier
[32m[03/05 12:58:48 detectron2] @Arguments:
Namespace(confidence_threshold=0.5,
config_file='/usr/local/lib/python3.8/dist-packages/simpleimageclassifier/config.py',
instanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml', input=
['/ccp_data/output/008532681000_F_85326806_K3nKFIDnL7BKZZpgpIlt'], output='MODEL_OUTPUT',
'MODEL_WEIGHTS', 'detectron2://COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x/137849600/model_f18217.p
```

outputs/output.zip

Generate code for: Python 3

**1.0.1** **running**

Accepted 01/03/2024 @ 15:19:26.

Last update 01/03/2024 @ 15:22:33: Execution completed: Initialization completed

Workspace > CCP > executions > SimpleImageClassifier\_1.0.1\_2024-03-26T07-56\_f07872aa-b783-4b4e-8512-95545

Andrea's workspace

- VRE Folders
- \_shared attachments
- \_uploaded\_forms
- CCP
  - executions
    - SimpleImageClassifier\_1.0.1\_2024-03-26T07-56\_f07872aa-b783-4b4e-8512-95545
  - DataMiner
  - DCF-RDB-RECOFI-andrea.rossi
  - My Methods
  - Notebooks
  - OpenVROpenCourse

Name	Owner	Type
ccp-entrypoint.sh	Andrea Rossi	application/x-sh
ccpenv	Andrea Rossi	text/plain
method.yaml	Andrea Rossi	text/x-yaml
output	Andrea Rossi	Folder
stderr.0flq2hacvtxu.txt	Andrea Rossi	text/plain
stdout.0flq2hacvtxu.txt	Andrea Rossi	text/plain

## METHOD EXECUTIONS



# CCP Methods, editor, and execution programmatically



```
1 import os
2 import torch
3 #torch.manual_seed(5)#3,5
4 import random
5 #random.seed(0)
6 import numpy as np
7 #np.random.seed(0)
8
9 '''os.environ["OMP_NUM_THREADS"] = "4" # export OMP_NUM_THREADS=1
10 os.environ["OPENBLAS_NUM_THREADS"] = "4" # export OPENBLAS_NUM_THREADS=1
11 os.environ["MKL_NUM_THREADS"] = "4" # export MKL_NUM_THREADS=1
12 os.environ["VECLIB_MAXIMUM_THREADS"] = "4" # export VECLIB_MAXIMUM_THREADS=1
13 os.environ["NUMEXPR_NUM_THREADS"] = "4" # export NUMEXPR_NUM_THREADS=1'''
14
15 from src.evaluation.evaluator_manager import EvaluatorManager
16 from src.evaluation.evaluator_manager_do import EvaluatorManager as PairedEvaluatorManager
17
18 from src.utils.context import Context
19 import sys
20
21 if __name__ == "__main__":
22     print(f"Generating context for: {sys.argv[1]}")
23     context = Context.get_context(sys.argv[1])
24     context.run_number = int(sys.argv[2]) if len(sys.argv) == 3 else -1
25
26
27 '''if torch.backends.mps.is_available():
28     context.logger.info(f"MPS support founded switch to torch.set_default_dtype(torch.
29     context.logger.info(f"Clean the cache if torch.float64 where used before")
30     torch.set_default_dtype(torch.float32)'''
31
32 context.logger.info(f"Executing: {context.config_file} Run: {context.run_number}")
33 context.logger.info(
34     "Creating the evaluation manager.....")
```

SimpleImageClassifier 1.0.1 Marco Lettore

A simple image classifier with parametrizable url to input picture compatible with the D4Science infrastructure

Runtime ?  
nubisware/simpleimageclassifier:latest

Annotations for execution ?  
Annotations for execution

Inputs  
https://as1.ticm.net/vzjpp00683z08n1000\_f\_63226806\_KMKFDNL7Bx...ipili

Outputs  
 Input image  Output image

Execute Generate code for: Python 3

Execution Monitor

Search

SimpleImageClassifier 0 2 0 0

1.0.1 running

Accepted 05/03/2024 @ 13:54:48.

Last update 05/03/2024 @ 13:58:33: Execution completed: Initialization completed

D4Science production infrastructure nubisware/simpleimageclassifier:latest

```
!init .py
SimpleImageClassifier
__main__.py
SimpleImageClassifier
@1326(02/05 12:58:00 detection): @([onArguments:
NoneSpace1 confidence: threshold: 0.8, 2)
config_file="/usr/local/lib/python3.8/dist-
packages/simpleimageclassifier/configs/CCCP-
InstanceSegmentation/mask_rcnn_r_50_fpn_3x.yaml", inputs:
{ 'face_data/output/competition.jpg', opts={'MODEL_DEVICE': 'cpu',
'MODEL_WEIGHTS': 'detection2//000-
InstanceSegmentation/mask_rcnn_r_50_fpn_3x/137849600/model_final_110217.p
```

outputs/output.zip Generate code for: Python 3

1.0.1 running

Accepted 01/03/2024 @ 15:19:26.

Last update 01/03/2024 @ 15:22:33: Execution completed: Initialization completed

## METHOD EXECUTIONS

# Method Execution

The screenshot displays the ITINERIS Method Execution interface, which is divided into several panels:

- Methods List:** A sidebar on the left lists various methods such as "Image Classifier", "Species distro", and "Archaeological Text Description". A callout points to this list: "Select or search method from list of available ones".
- Method Configuration:** The central panel shows the configuration for a method named "First Test". It includes sections for "Runtime" (set to "bash"), "My env variable" (set to "defaultvalue"), "Workspace link" (a URL to a workspace), "Boolean" (checked), "A number" (set to "4"), and "A multiple choice" (radio buttons for "a", "b", and "c"). A callout says: "Specify values for inputs".
- Execution Status:** Below the configuration, there are "Options" (e.g., "Automatically archive outputs to workspace"), "Outputs" (checkboxes for "Standard output" and "Standard error"), and an "Execute" button. A callout says: "Request execution interactively or programmatically".
- Execution Output:** The right panel shows the execution status as "Accepted 3/25/2024 @ 10:15:57 AM" and a terminal window with a large block of text output. A callout says: "Monitor execution status in realtime".
- Execution Controls:** At the bottom of the right panel, there are buttons for "Bolernet With Custom Model", "Bolernet", and "Python38". A callout says: "Generate code for programmatic re-execution and direct links to use in publications".
- Share Outputs:** A callout at the top right says: "Share outputs to WS, archive, restore, resubmit execution. Generate Prov-O files".

## Method execution: few details



- D4Science infrastructure is based on Docker containers.
- Any publicly available Docker registry
  - Including an experimental D4Science specific registry
- Environmental variables are passed by default to your Method
  - ease authentication when interacting with D4Science services (**ccpiamurl**, **ccpclientid**, **ccpcontext**, **ccprefreshtoken**)
  - **ccptaskname** and **ccptaskid**
- Special folder **/ccp\_data** which is zipped and uploaded to the execution repository
  - The size is limited so DON'T upload huge files.
  - Use workspace or other storage facilities for large files.
  - Some meta outputs are kept in the outputs for debugging purposes
- Please consider good housekeeping :-)
  - Delete useless executions. They are considered volatile.
  - Archive to your workspace executions or execution outputs that have a value.
  - Backup your Methods to the workspace



Let's play with CCP now!

THANKS!

**IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System**  
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-  
Mission 4 "Education and Research" - Component 2: "From research to business" - Investment  
3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"

