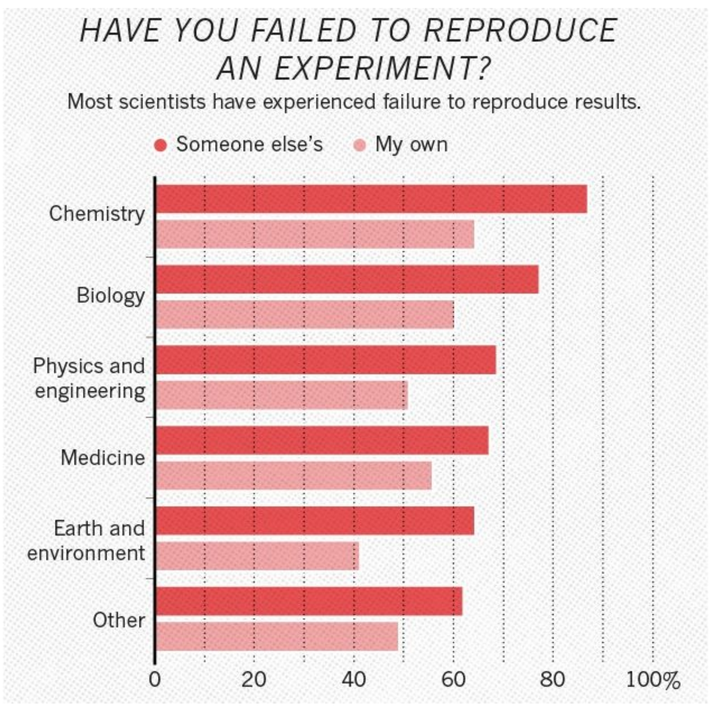
Support for HTCondor high-throughput computing workflows in the **REANA** reusable analysis platform

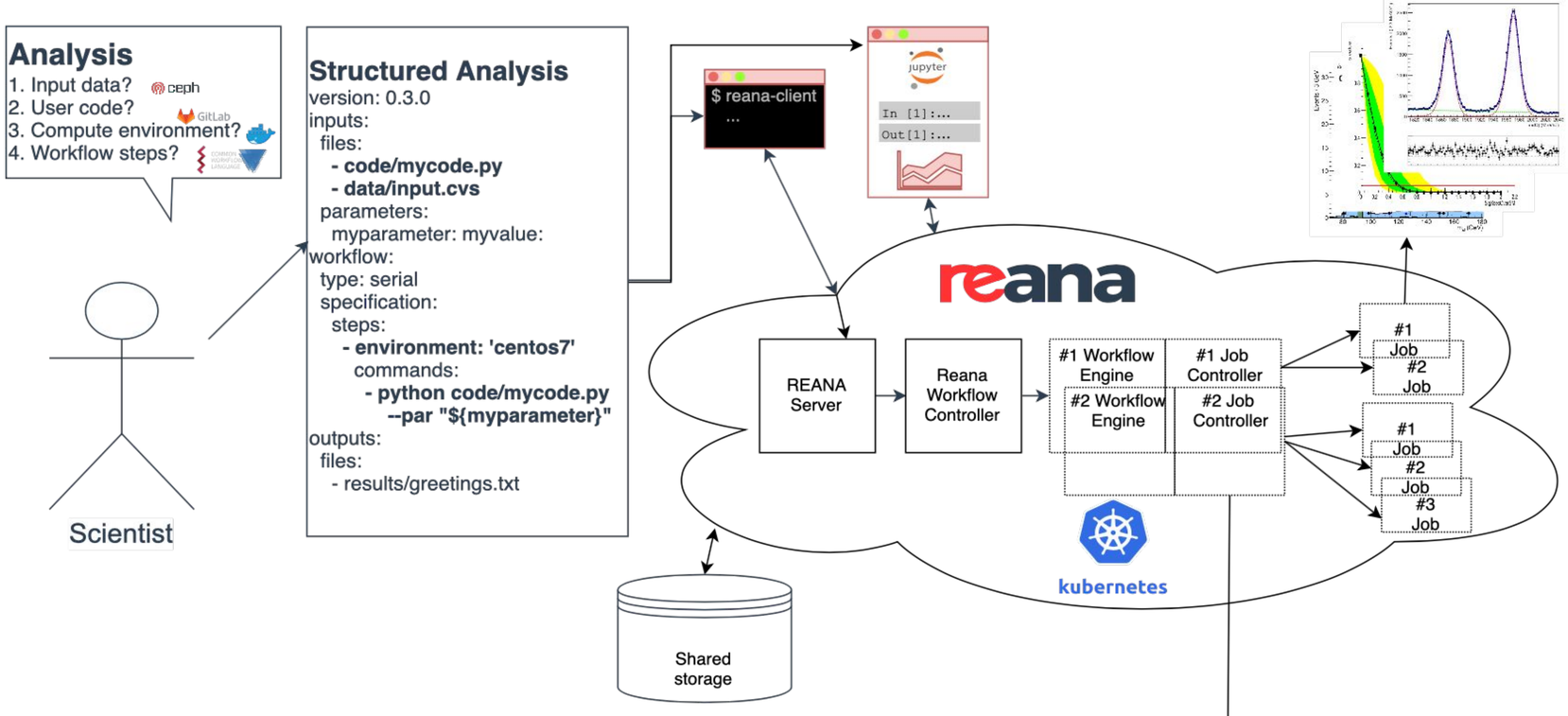
Rokas Mačiulaitis	Paul Brenner	Scott Hampton	Michael D. Hildreth	Kenyi Paolo Hurtado Anampa	Irena Johnson	Cody Kankel	Jan Okraska	Diego Rodriguez	Tibor Šimko
CERN	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	CERN	CERN	CERN
Switzerland	United States	United States	United States	United States	United States	United States	Switzerland	Switzerland	Switzerland
rokas.maciulaitis@cern.ch	paul.r.brenner@nd.edu	shampton@nd.edu	mhildret@nd.edu	khurtado@nd.edu	ijohnso1@nd.edu	ckankel@nd.edu	jan.okraska@cern.ch	diego.rodriguez@cern.ch	tibor.simko@cern.ch

Reproducible research

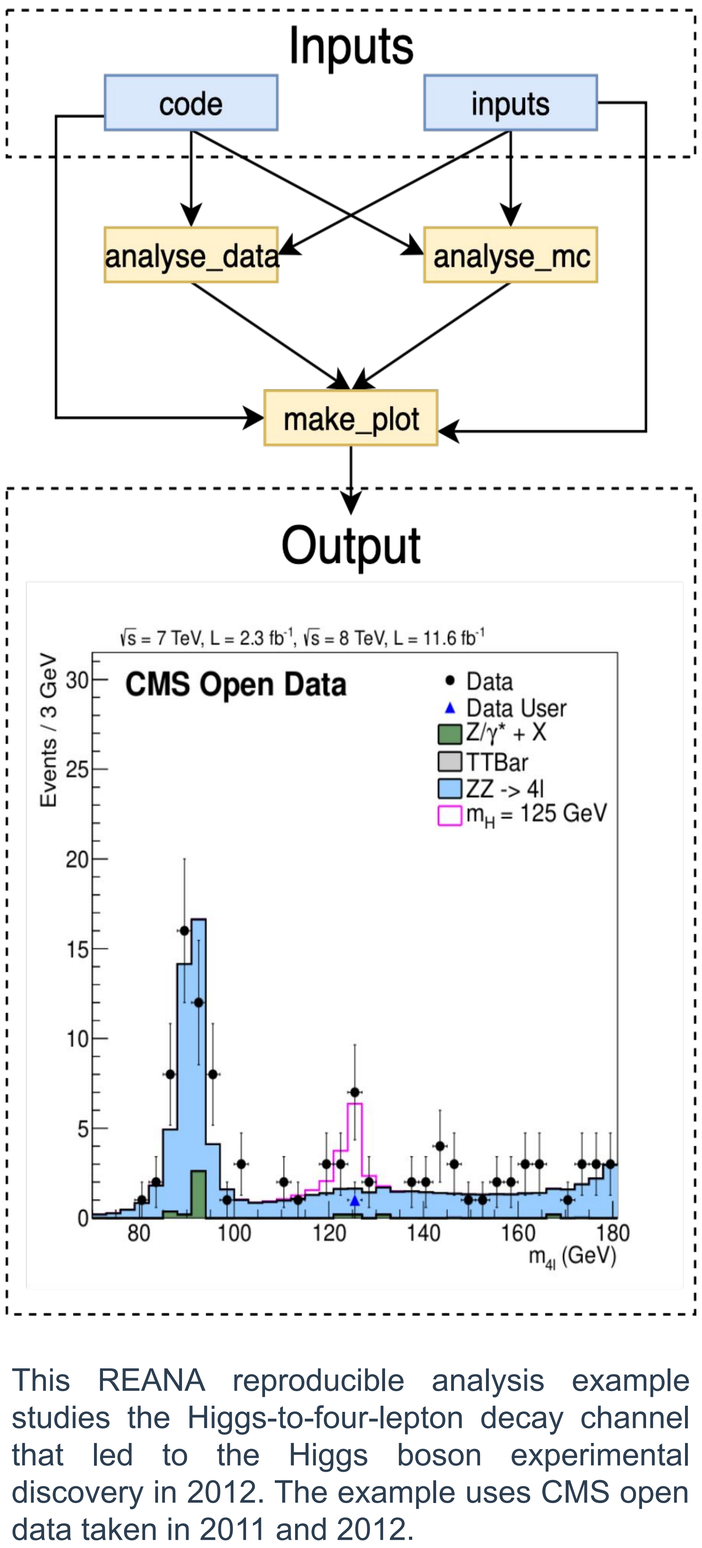


https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970

REANA approach



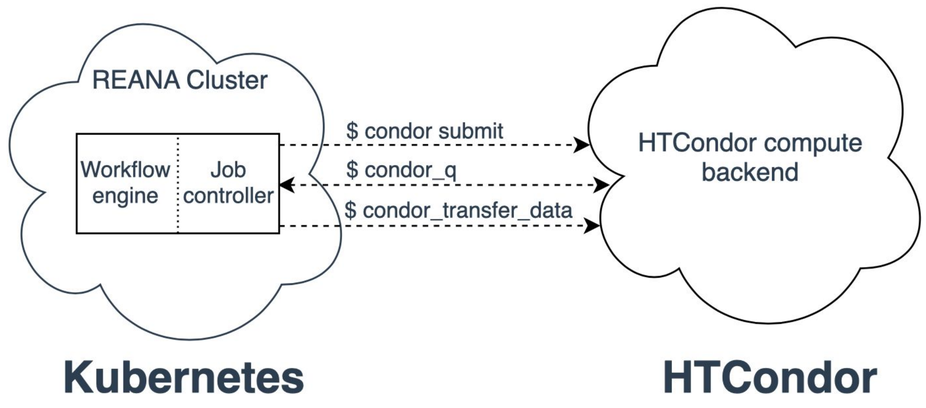
Example workflow



This REANA reproducible analysis example studies the Higgs-to-four-lepton decay channel that led to the Higgs boson experimental discovery in 2012. The example uses CMS open data taken in 2011 and 2012.

https://github.com/reanahub/reana-demo-cms-h4l

Extending REANA with HTCondor

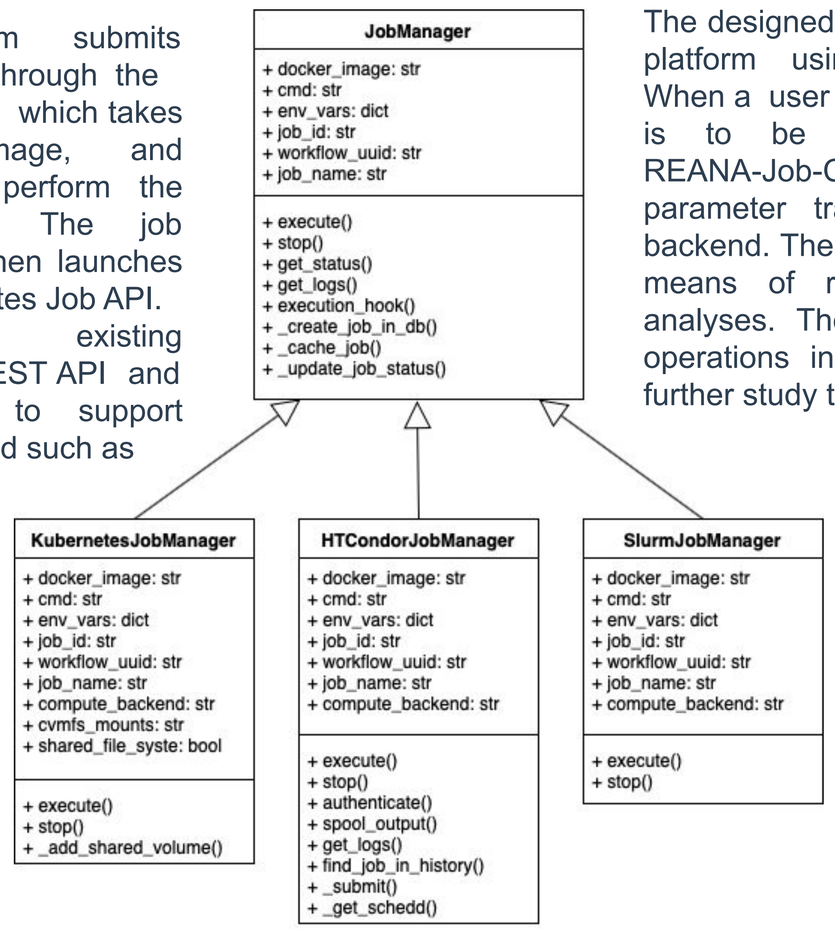


The typical interaction between the REANA-Job-Controller component and the remote HTCondor computing backend involves three steps:

1. Job submission + transfering input files.
2. Job monitoring
3. Transferring back output files.

Abstracting job submission

The REANA platform submits analysis workflow steps through the Job-Controller component which takes inputs, container image, and commands to run to perform the workflow step tasks. The job controller component then launches the job using the Kubernetes Job API. We have taken existing REANA-Job-Controller REST API and extended the design to support arbitrary compute backend such as HTCondor for high-throughput computing or Slurm for high-performance computing. The abstraction regards job submission and execution, job status monitoring, and the input/output data transfer amongst supported backends.



Validation

The designed solution was prototyped in the REANA platform using the CERN HTCondor cluster. When a user specifies that a certain workflow job is to be run on the HTCondor backend, the REANA-Job-Controller container takes care of job parameter translation for the targeted compute backend. The developed prototype was tested by means of running several particle physics model analyses. The configurable level of "map-reduce" operations in the DAG workflow graph allows to further study the scalability of the solution.

We have furthermore integrated REANA with the Virtual Clusters for Community Computation (VC3) environment. We have developed Ansible templates allowing individual users to deploy the personal REANA system in a VC3 environment. The deployment benefits from integration with a range of high-throughput computing backend that the VC3 environment natively supports. Our design was thus validated by means of two independent deployment scenarios.

www.reana.io

GitHub @reanahub

Gitter gitter.im/reanahub/reana

Docs reana.readthedocs.io

Email info@reanahub.io

DockerHub reanahub

Twitter @reanahub