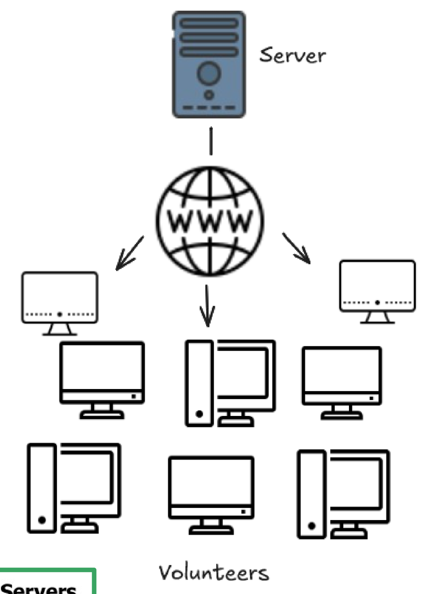
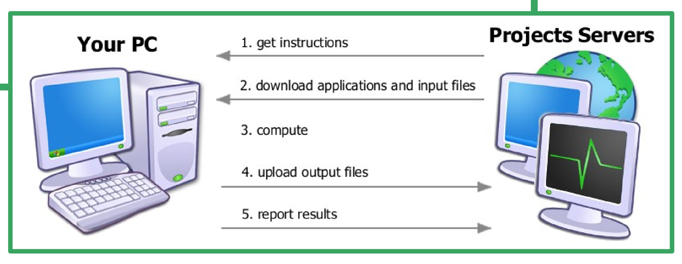
The evolution of the CMS@Home project

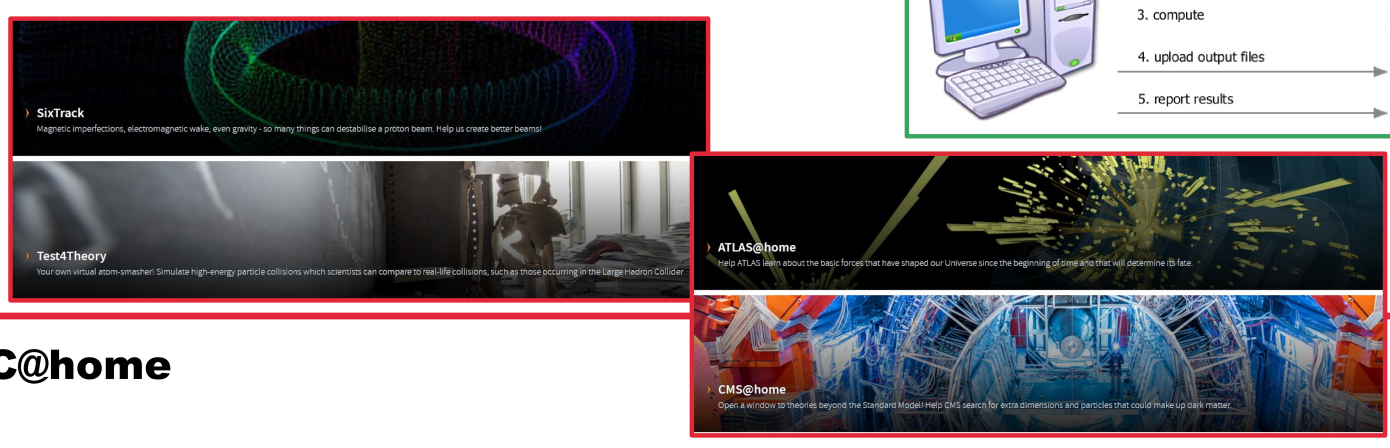
Federica Fanzago (INFN – Padova), Laurence Field (CERN), Ivan D.Reid (Brunel University London),
Marco Mascheroni (Univ. of California San Diego), Alan Malta Rodrigues (Univ. of Notre Dame),
Daniele Spiga (INFN-Perugia), Christoph Wissing (Deutsches Elektronen-Synchrotron)

Volunteer computer

Volunteer computing is a well-established form of **citizen science** where users download scientific software onto their own devices and run simulations or analyses, in a way that does not interfere with their computer's normal operation. **People donate their unused resources to a research-oriented project**, sometimes in exchange for credit points. Exploiting voluntary computing resources as **additional capacity for experiments** at the LHC gave rise to individual initiatives such as the CMS@Home program, which were subsequently embraced into the wider LHC@Home project.





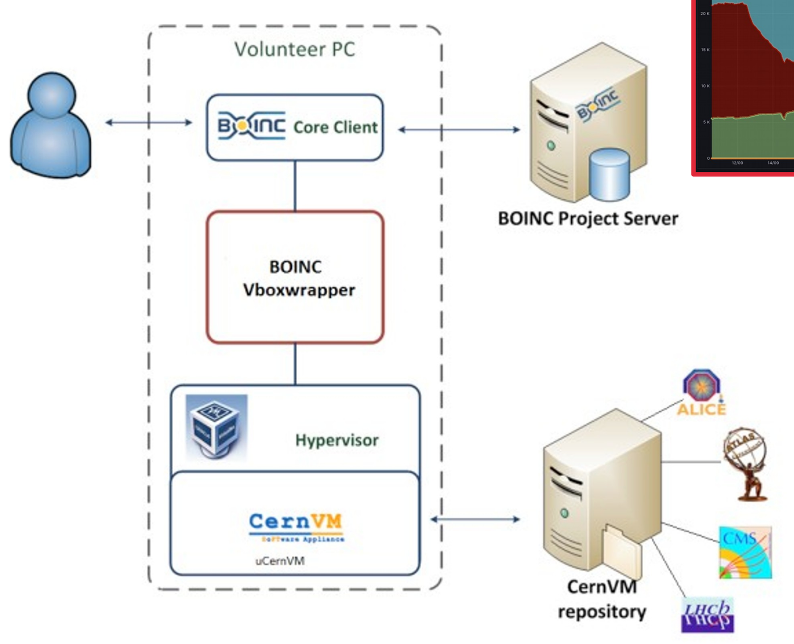


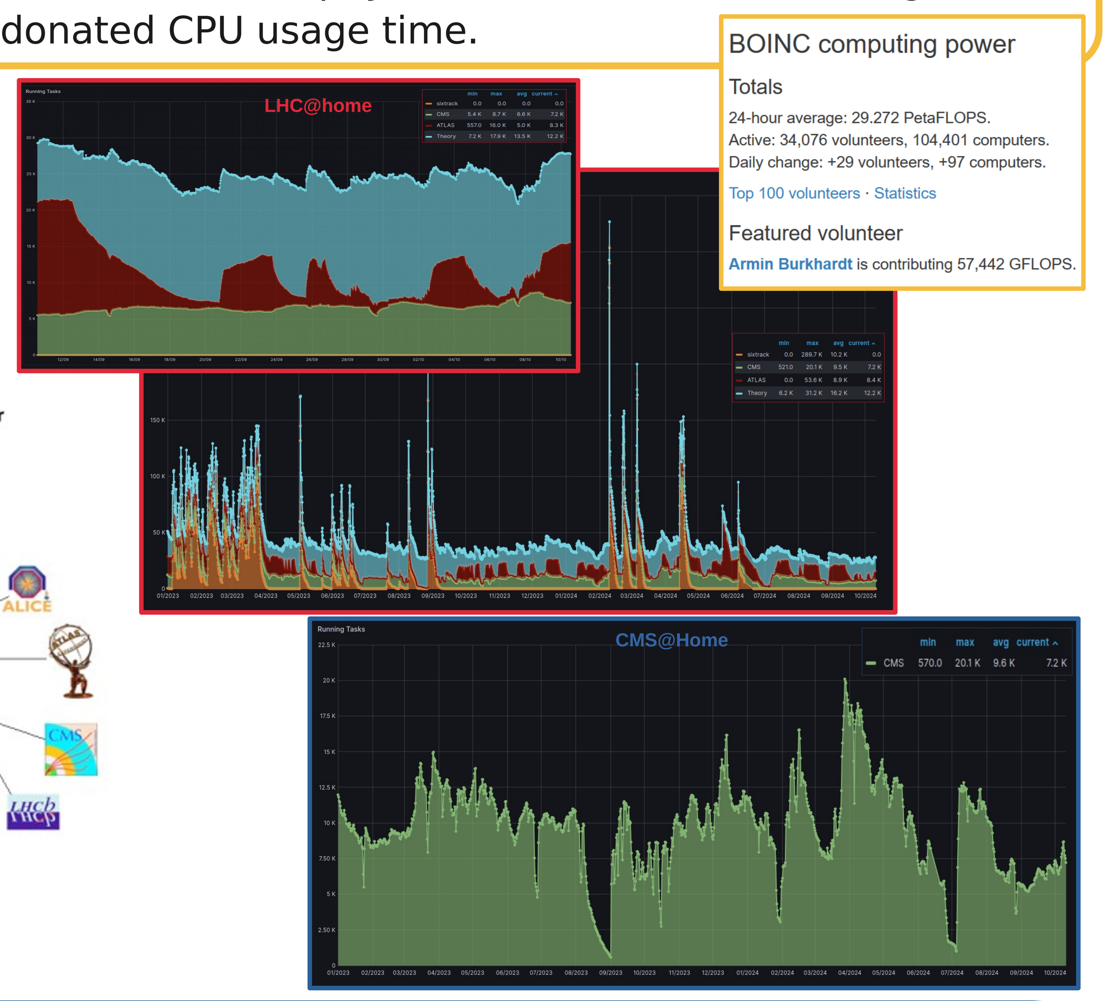
LHC@home

Launched as a BOINC project in 2004 and initially focused on the accelerator physics numerical simulation code SixTrack, since 2011 this has expanded to a wider range of LHC applications, thanks to **the adoption of virtualization** which allows software to run on **heterogeneous resources**, without the need to recompile code for the different systems.
In this way, code developed in Linux systems can also run inside Virtual Machines on MacOS and Windows as well, widening the pool of potential volunteers. The Linux VMs use CernVM as the base for application images, and OracleVirtual Box as the local hypervisor. The VMs are managed by the BOINC client using the Vboxwrapper software module, supported by the BOINC development team.
The aim to LHC@home is to increase the computational power of experiments for simulations and upgrade studies by exploiting idle volunteer resources, involving lay people in real science.
Volunteer resources are free but their integration into an experiment's framework needs some effort. The same BOINC server is used for all the @Home projects.

The BOINC system

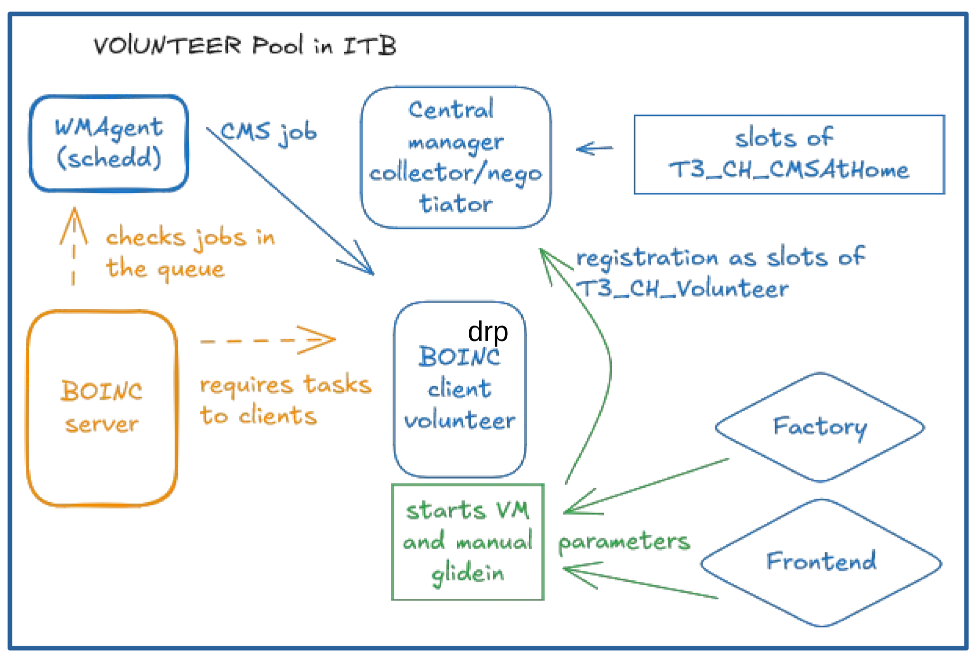
The Berkeley Open Infrastructure for Network Computing (BOINC) is a software that **enables computers to put unused CPU/GPU cycles to work for projects needing more computational power.**
Developed to manage the **SETI@home** project in 1999 to use volunteers' computing capacity to analyze signals from the Arecibo Observatory's radio telescope, it was then also adopted for other scientific projects.
BOINC is a **client/server platform for distributed internet-connected computing** designed to support 'high throughput computing', in which there are large numbers of independent compute-intensive jobs. **The client runs on volunteer computers** and the server, dedicated to a specific project, manages scheduling, tasks and users. **The server provides project applications** to be executed on volunteer hosts. **Computing requirements are defined by the project itself**. Volunteers configure the client, through the BOINC manager, deciding when and how to make available their resources. To entice people to participate in the project a "credit system" was implemented in BOINC. There is no payment involved, but a recognition in credits based on the donated CPU usage time.





CMS@Home

A research project that uses volunteer computing to run simulations of the CMS experiment, CMS@Home is one of the applications hosted by the LHC@home project.
Born as a proof of concept project in 2015, with a starting point of R&D prototypes and projects such as "jobs in the Vacuum" and SETI@Home, demonstrating the feasibility of using volunteer resources to run CMS applications for data simulations/reconstruction via CRAB, a tool of the Workload Management group mainly dedicated to user analysis.
Over the past years **it has evolved by interfacing with the official tool for the central production of datasets for CMS,** the WMAgent, demonstrating the ability to follow and adapt to changes of the CMS computing infrastructure.
The CMS workload management system is based on GlideinWMS and HTCondor, which manage all the computational resources as a single pool, the CMS global pool. The current method to run jobs on a trusted remote resource (grid or cloud node) is to use the "pilot" approach (glideins), job that checks if a remote node is correctly configured to execute real jobs (payload). After the suitability is confirmed, the pilot allows the connection of the node to the HTCondor pool obtaining jobs from the central queue tasks of CMS.
In the classical pilot-based pool the submission of pilots are triggered by the CMS Frontend service from the Factory, upon detecting workloads in a submit node (schedd).
But **volunteer resources are not trusted** by the experiment, so they can not be added to the pool in the same way.



Production WMAgents can flock jobs to the volunteer pool

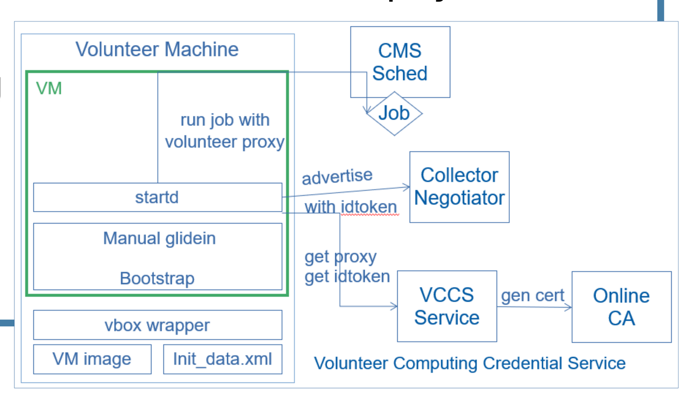
CMS@Home challenges

Volunteer resources are volunteered!

- **untrusted resources**: anyone can subscribe to the project and donate resources → needs a dedicated HTCondor pool and manual glideins for "vacuum-like" resources provisioning.
- **limited bandwidth** (up and down): connection can be via a home network → running MonteCarlo workflows producing small outputs (~50 MB)
- **no volunteer storage**→ no external input data, only MC task-chain workflow.
- **outputs have to be "quarantined"** in a dedicated storage (DataBridge) before being injected into the production system.
- **resources availability isn't guaranteed**: users may disconnect at will → MaxHibernate time required in HTCondor (7200 s) and low priority workflow
- **limited CPU and RAM** → legacy single and multicore workflows (currently max 4 cores, 2GB RAM)

CMS@Home setup

When a **data production request** is injected into the system, the **WMagent prepares jobs and queues them in its HTCondor schedd. The BOINC server periodically checks the HTCondor queue and creates BOINC tasks ready for BOINC clients** to process.
Clients connected to the server **start a VM using the dedicated CMS application image** provided by the server.
The bootstrap script of the VM launches the manual glide wrapper script that downloads, from the factory and the frontend, details about the required CMS environment and the parameters for the setup of HTCondor in the machine, and starts the pilot. If checks and setup are successful, the pilot starts the HTCondor daemon connecting the resource to the volunteer pool as **slots of the T3_CH_Volunteer site** and payloads can arrive and be executed.
If the pilot stays running 30 minutes without receiving a real job, it terminates, triggering the switch off of the VM.

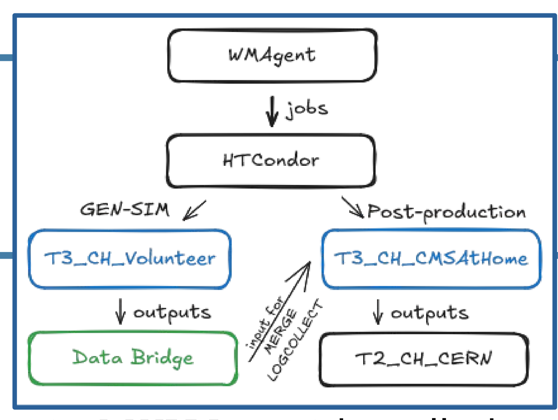


Authentication/authorization

Users register to the project with their BOINC username and password. A **Volunteer Computing Credential Service** (VCCS) was developed to generate a short-lived, 7-days, **X.509 proxy from the BOINC credentials**. This proxy is made available during the VM's bootstrap allowing jobs to run on volunteer hosts and to copy results and logs onto dedicated storage, the DataBridge.
Connection of volunteer resources to the HTCondor pool also requires authorization. In the past this was done through the proxy, but currently it is based on an "idtoken" which is produced centrally, renewed periodically, and also passed to the VM during the bootstrap.

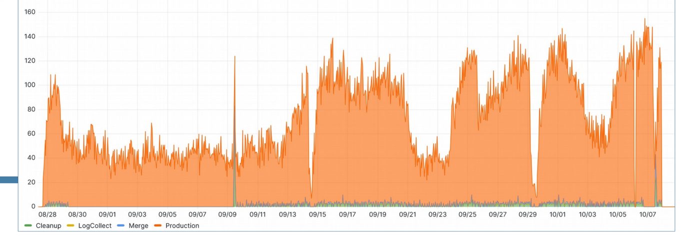
Storage

The **DataBridge is a S3 Ceph storage system located at CERN**, where **volunteers can only write** their output files, deleting/overwriting is forbidden to them. **Postproduction steps** (merge, cleanup and LogCollect) **run** on a dedicated site, **T3_CH_CMSAtHome**, and having an associated production proxy, they can remove files from DataBridge. The URL redirector is provided by DynaFed. The storage provides an **automatic cleanup** policy to remove files older than 60 days.



Workflow

- Using a **dedicated WMAgent** installed at CERN to inject jobs to the HTCondor pool, without special setup
- Managing **single- or multi-core jobs**, but the volunteers must select their VM preferences to match.
- Over the last 90 days, the number of running cores has varied between 700-2200. Since April, 220 GB of results have been collected, representing 2.4 million filtered events from a total of 17.1 billion (efficiency ~0.00018).
- Currently (October) we are creating **4-core jobs**; they average around 10 hours running time.



Google font Atkinson Hyperlegible and icons from https://icons8.it

https://lhcathome.web.cern.ch/projects/cms