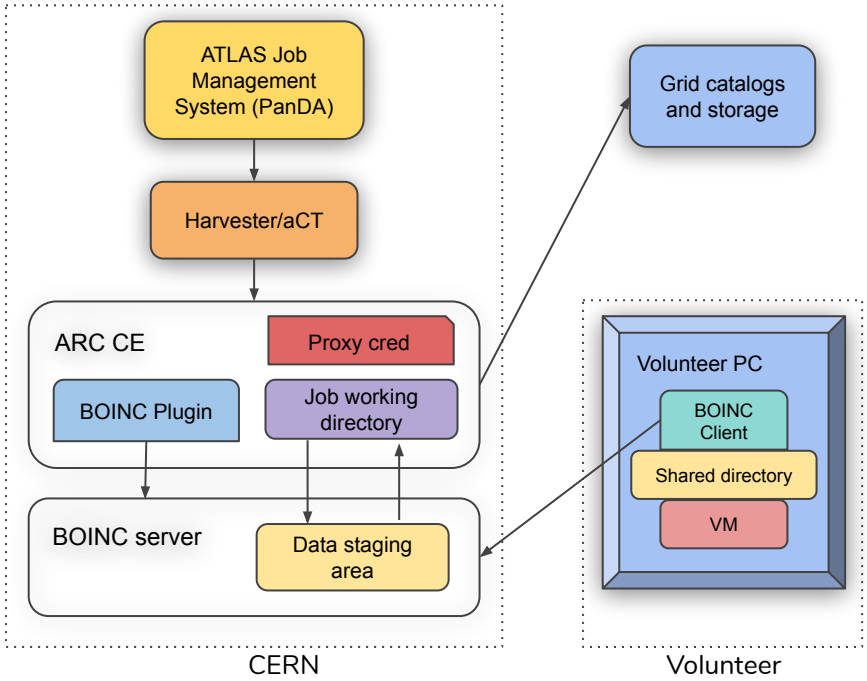
Adapting ATLAS@Home to trusted and semi-trusted resources

David Cameron, University of Oslo
Vincent Garonne, University of Oslo
Paul Millar, DESY
Shaojun Sun, University of Wisconsin Madison
Wenjing Wu, University of Michigan
On behalf of the ATLAS collaboration

ATLAS@Home: Volunteer computing for ATLAS

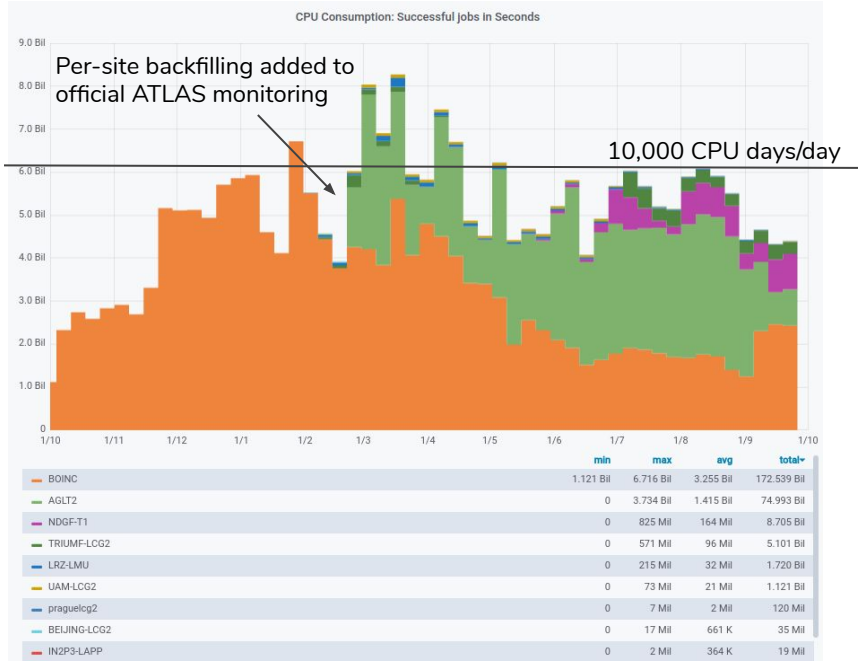
- “Free” but untrusted computing resources
- Architecture was designed to give a clear separation between volunteers and the rest of the ATLAS grid
  - Isolated payload
  - Data staging sandbox
  - Result validation



Expansion beyond traditional volunteers

- ATLAS@Home is now also used as a backfilling platform on Grid sites
  - Native Linux version running in Singularity container instead of VirtualBox
  - Independent from batch system and grid jobs
  - Not affected by grid services downtime
  - By design BOINC does backfilling
- Up to 50% of work performed by Grid backfilling
- Can be installed on a worker node in one line:

```
curl http://atlasathome.cern.ch/Atlas-test/scripts/install_atlasathome.sh
| bash -s -- --uid <boinc_authenticator> --ProDir /home/boinc
```

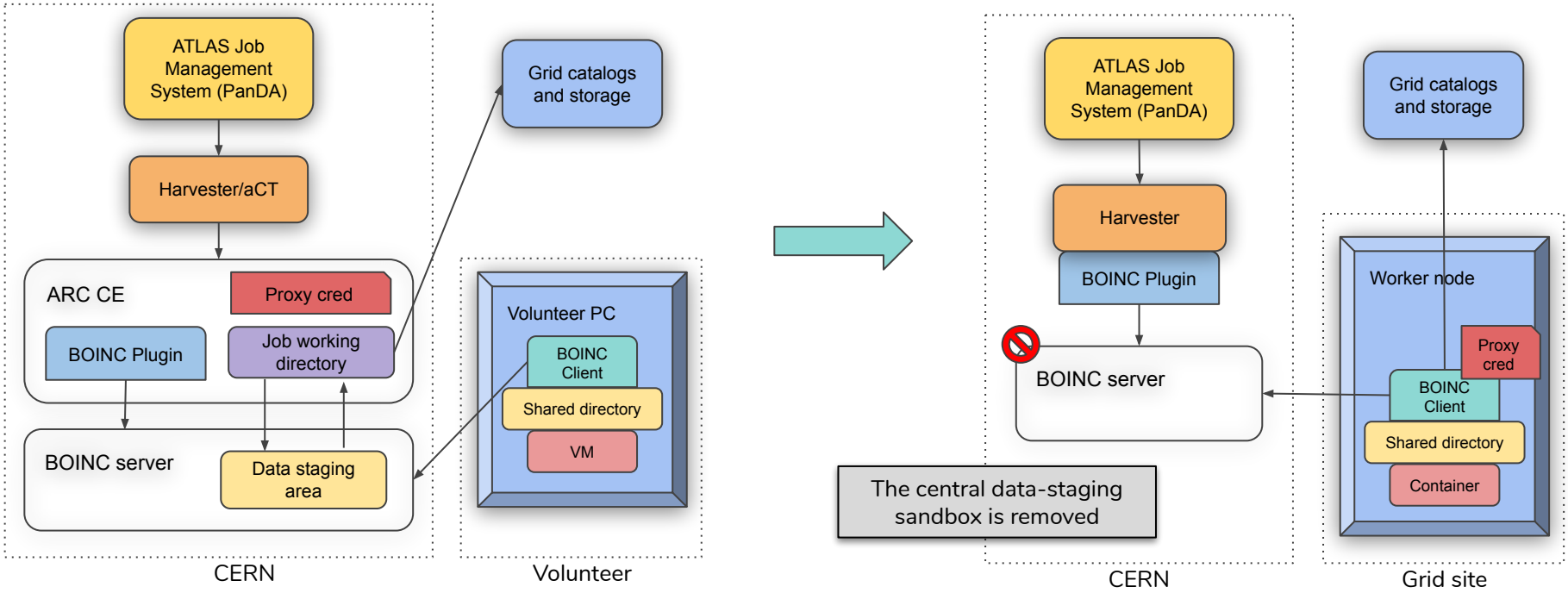


CPU consumption of ATLAS@Home jobs from Oct ‘18 to Oct ‘19. From March ‘19 jobs running on grid worker nodes are accounted under the corresponding site.

Backfilling trusted resources

- Backfilling resources (Grid worker nodes) are trustable
- In addition scaling up could cause a data transfer bottleneck in the sandbox
  - Current tasks are MC simulation
  - 200 events per job, 5-30 mins per event (depending on physics and CPU power)
  - 4000 - 20000 jobs per day
  - 300MB input, 200MB output
  - Up to 10TB traffic per day (100MB/s)
  - Probably not feasible for scaling up one order of magnitude

Backfilling trusted resources



How it works

- A BOINC plugin was written for Harvester
  - Uses BOINC RPC (python wrapper around XML) to communicate with BOINC server
  - Submits proxy and simple wrapper which downloads ATLAS pilot wrapper then runs it
  - Queries status of jobs
  - Copy logs (pilot stdout) of finished jobs to harvester web area
- Why not just run a “pilot launcher script”, vacuum model etc?
  - It would require:
    - A scheduling system to start processes on each worker node
    - A way to control how they run alongside the real jobs from the batch system
    - A secure means of getting the X.509 credential
    - A way for central operations to monitor these processes
  - i.e. implementing something that looks a lot like BOINC

Backfilling status

- A test set up exists with a private BOINC server and a PanDA queue BOINC_BACKFILL running event service jobs
- Future plans:
  - Setting up on a real grid site
  - Dynamically adapting to the site running the job (data transfer to/from local storage)
  - Possibly using other workflows than simulation (eg event generation)

Finished job: https://bigpanda.cern.ch/job?pandaid=4500811827

PanDA Event Service job details for job **4500811827**

PandaID	Owner WG	TaskID	Mode	Cores	Status	Substate	Created
4500811827	mnegrini / AP_HIGG	19195544		1	finished	es_wait	2019-10-03 06:20:07

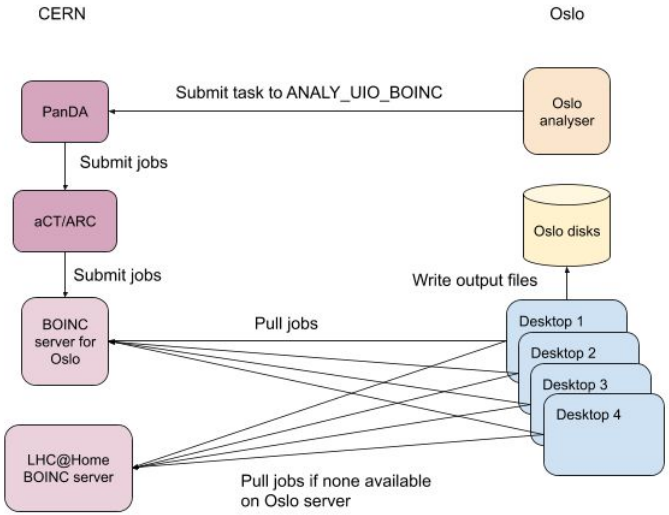
Job type: simul **Job transform:** Sim_tf.py **Dispatched event states:** merged(224)

Datasets: In: mc16_13TeV:mc16_13TeV.345337.PowhegPythia8EvtGen_NNPDF3_AZNLO_ZH125J_MINLO_llWWlvlv.merge.EVNT.e5810_e5984_tid19195541_00
Out: mc16_13TeV.345337.PowhegPythia8EvtGen_NNPDF3_AZNLO_ZH125J_MINLO_llWWlvlv.simul.HITS.e5810_e5984_s3126_tid19195544_00

Dispatched event states	merged(224)
TaskBuffer return code	115
TaskBuffer message	finished since an associated ES or merge job PandaID=4501641145 finished

ATLAS@Work

- "*ATLAS@Work: Boosting research by running ATLAS@Home on HEP desktops and other opportunistic computers*"
- A project at the University of Oslo to use idle office desktops to boost local user analysis
- Users submit regular PanDA tasks but specify running on a special queue
- Desktops are connected to a private BOINC server which distributes tasks
- Outputs are written directly to local Oslo grid disks



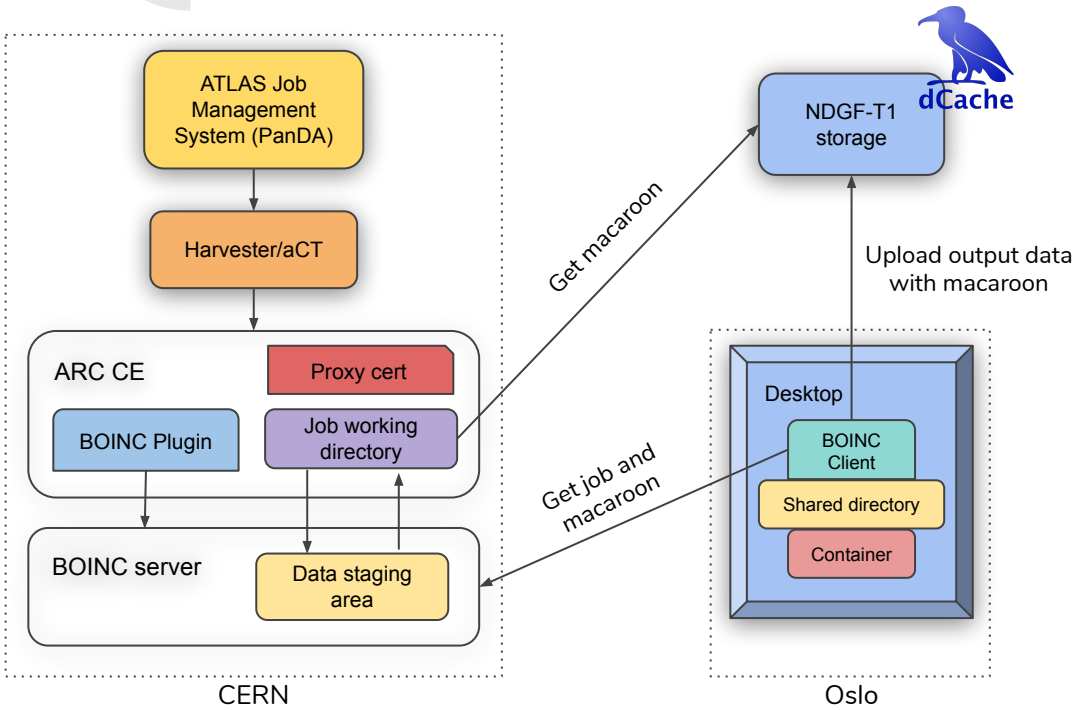
ATLAS@Work: Data management using Macaroons

- Student desktops are a “semi-trusted” resource
  - Allowed to interact with grid storage to read input data and write output data
  - But not allowed production grid proxies with full powers
- Macaroons are a simple way of providing time- and resource-limited access to a storage resource through “caveats”
  - “With this macaroon you can upload to http://storage.org/data/file1 for the next 24 hours”
  - A macaroon can only be used for a specific operation on a specific file, making them more secure than grid proxies
- ATLAS@Work macaroons are requested like this:

```
auth="-E $userProxy --cacert $userProxy  --capath /etc/grid-security/certificates"
caveats_boinc={\"caveats\":[\"activity:UPLOAD\"],\"validity\":\"PT24H\"}
curl $auth -sH 'Content-Type:application/macaroon-request' -X POST -d '$caveats_boinc' $output_file
```

- “Give me a 24h token to upload to this filename”

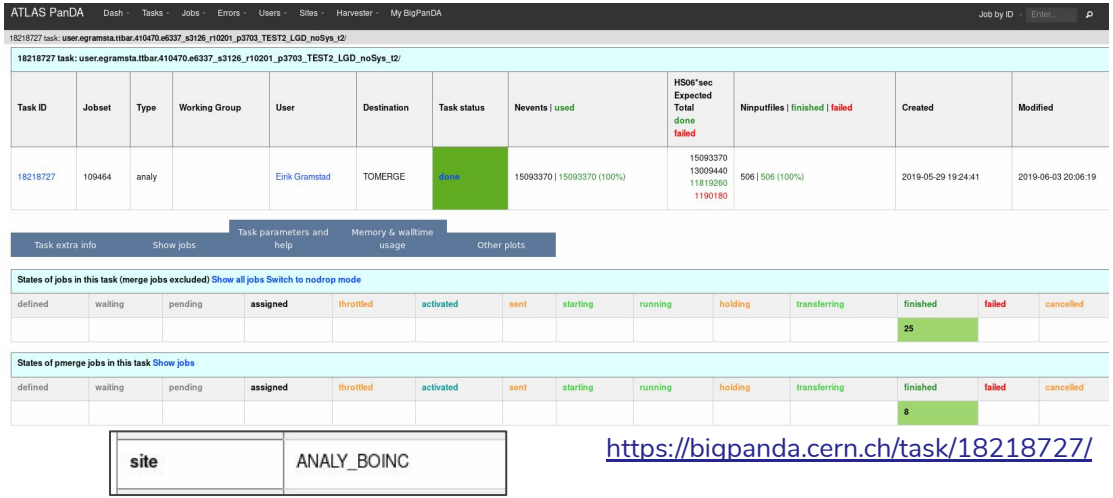
Macaroon Implementation in ATLAS@Work



- Macaroons are enabled on NDGF-T1 dCache WebDAV door
- ARC CE uses the proxy to request macaroons for the job output files
- It passes the macaroons to BOINC along with the job description
- The BOINC client downloads the job and macaroons
- At the end of the job it uploads the output files using the macaroons

ATLAS@Work status

- A queue in PanDA ANALY_BOINC sends jobs to the private BOINC server for Oslo users
- Some desktop machines in Oslo are connected to this server
- Real analysis tasks have been completed successfully
- Future plans:
  - Increase the pool of resources
  - Try the macaroon concept with real volunteers



https://bigpanda.cern.ch/task/18218727/

Conclusions

- The expansion of ATLAS@Home beyond its traditional “@home” base has driven some new developments and optimisations
- Running in a “Grid pilot-like” way on fully trusted resources like worker nodes relieves potential data management bottlenecks
- Macaroons provide a useful mechanism to allow restricted access to Grid storage, and could potentially be used by the real @home volunteers

Acknowledgements

- Thanks to
  - CERN IT for hosting the BOINC infrastructure and help integrating to LHC@Home
  - All our volunteers not just for resources provided over the years but for support in helping out others with problems
  - Grid site admins willing to risk running ATLAS@Home alongside their Grid jobs
  - Analysers at University of Oslo testing out ATLAS@Work

- Join us! http://lhcathome.web.cern.ch/projects/atlas