ATLAS Open Data Tutorial - EXPLORE Session

CERN, Wednesday 26 November 2025. 14:00-17:30

EXPLORE: Open Data Analysis Without Barriers

Speaker: Baida Achkar | baida.achkar@phys.uni-goettingen.de

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

ATLAS EXPERIMENT

PUNCH 4NFDI

Agenda

14:00–14:30 ATLAS Global Resources, EXPLORE: Infrastructure, Access & FAIR, Discussion / Questions
14:30–15:30 Hands-On: $t\bar{t}$ Analysis on EXPLORE
15:30–16:00 ☕ Break
16:00–16:30 Inspect jobs logs, outputs, summarize results and download to local laptops,
16:30–17:00 Optional: Physlite Workflow
17:00–17:30 Reflection & Wrap-Up

 Part I - Global Resources for Open Data Analysis

Global ATLAS Analysis Resources - US

U.S. ATLAS Analysis Facilities - Bridging Local & Global Grid Computing

- Sites include Brookhaven (BNL), University of Chicago, and SLAC (California).
- Shared Tier-3 computing environments for ATLAS collaborators
- Support both batch & Jupyter-based workflows for interactive notebook analysis and large-scale Grid production.

🔒 Require federated login (CERN / Institutional).

🔗 Learn More: https://cds.cern.ch/record/2872151
https://usatlas.readthedocs.io/projects/af-docs/en/latest/

Coffea-Casa @UNL: Cloud-Native Analysis Facility

- Hosted at the University of Nebraska-Lincoln, developed in collaboration with Fermilab.
- A cloud-native facility designed for Open Data and high-energy physics analysis.
- Runs Python-based Coffea workflows through Jupyter notebooks.
- Uses Kubernetes and Dask clusters for scalable, parallel computation.
- Acts as a bridge between research computing & open data accessibility.

🔗 Learn More: https://coffea-casa.readthedocs.io
https://arxiv.org/abs/2203.10161

Global ATLAS Analysis Resources - EU

ESCAPE: Connecting European Science and Open Data

- Connects data & compute infrastructures across Europe
- Integrates HPC, HTC, & cloud resources under the European Open Science Cloud (EOSC).
- Provides the ESAP (Science Analysis Platform), a unified hub for data discovery, workflow management, & notebooks execution.
- Promotes FAIR Data Principles: *Findable*, *Accessible*, *Interoperable*, *Reusable*.
- Enables educational and research collaboration across Europe.

🔗 Learn More: https://projectescape.eu , https://arxiv.org/abs/2302.11453

EXPLORE: Open Data Analysis Without Barriers

- Open access: no CERN/institutional login (SSH registration).
- Reproducible: containerized jobs (Apptainer + CVMFS).
- Educational focus: teachers, students, independent researchers.
- Sustainable: migrating to NHR EMMY in PUNCH4NFDI-2.0.

🔗 Learn More: https://zenodo.org/records/16736008. https://indico.desy.de/category/897/

Who Can Use EXPLORE?

EXPLORE is designed for anyone curious about particle physics - not only CERN insiders!

Audience	Purpose
High-school & university students	Learn real LHC analysis workflows & explore physics careers
Teachers & educators	Integrate real research data into STEM education
Researchers without LHC access	Analyze ATLAS Open Data without credentials or grid access
General public / citizen scientists	Experience real HEP computing & reproducible workflows

No CERN account needed - open, inclusive, and education-focused.

Part II - EXPLORE: “From Federated Science Platforms to Open Data for Everyone”

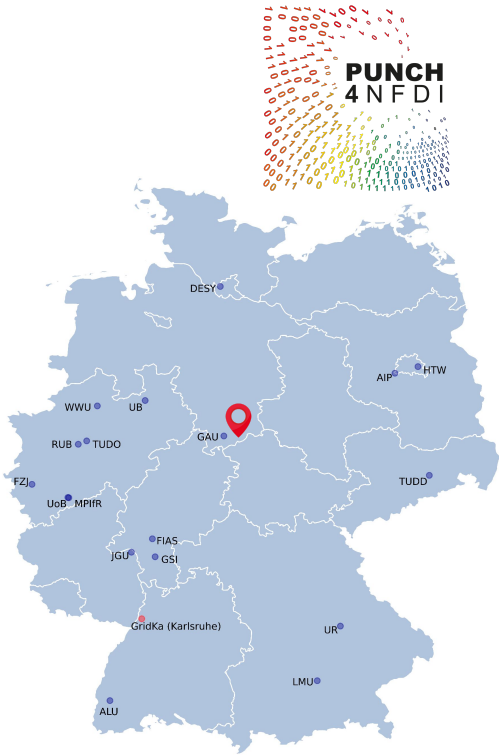
🏛️ **Federated Platforms** → 🚀 **EXPLORE** → 🌍 **Open Users**

PUNCH4NFDI Consortium at a Glance

- PUNCH4NFDI, the NFDI (National Research Data Infrastructure) consortium of Particle, Astro-particle, Hadron & Nuclear physics and Astronomy.
- About 9000 scientists in Germany, from universities, the Max Planck Society, the Leibniz Association & the Helmholtz Association.
- PUNCH4NFDI aims to setup a federated and "FAIR" Science Data Platform PUNCH-SDP.

➔ *offers infrastructures & interfaces for the access to & use of data and computing resources & tools of the PUNCH communities & beyond,*

➔ *helps to master the challenges of big data, open data & "FAIR" workflows.*

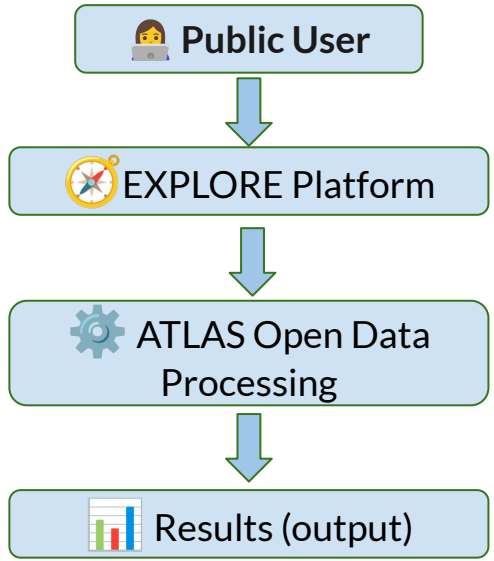


Borrowed from PUNCH4NFDI TA2 (Data Management) official talks

EXPLORE @GoeGrid → Open Data without Barriers

Purpose: EXPLORE is a gateway to practical LHC Open Data analysis.

- Developed at Uni-Göttingen / II Institut für Physik within PUNCH4NFDI initiative, promoting cross-disciplinary FAIR data practices in physics!
- Removes major technical + access barriers:
  - Ready-to-use container environments (ROOT + libraries).
  - No CERN credentials required.
  - Remote data access pre-configured.
- Empowers **educators**, **students** & **independent researchers** to analyze real ATLAS Open Data securely and at scale



Why Use EXPLORE?

- **True HEP compute environment:** scalable batch system, containers, CVMFS
- **Free & open access:** no institutional login or Grid certificate required
- **Reproducible workflows:** containerized execution identical to ATLAS computing
- **Bridges theory & practice:** run real analyses with production-grade tools
- **Empowers education & outreach:** hands-on access to Open Data for everyone

EXPLORE provides real analysis power - not just toy examples!

EXPLORE: Infrastructure & Workflows

EXPLORE repurposes a fraction of **GoeGrid (a Tier-2 WLCG ATLAS site)** Göttingen's computing resources, as Open Analysis Resources, enabling LHC Open Data analysis **without** requiring CERN or institutional affiliation.

Core Components

- **HTCondor Overlay Batch System:** merges distributed compute nodes into one job pool.
- **Dynamic Resource Management (COBalD / TARDIS):** auto-scales compute power
- **Containerization** (Apptainer + CERN CVMFS):

  ⟶ full reproducibility.
- **Monitoring**: Prometheus + Grafana dashboards for real-time metrics.
- **Access to EXPLORE Pool:** open SSH login via punchlogin.goegrid.gwdg.de

EXPLORE: Registration Workflow & User Access

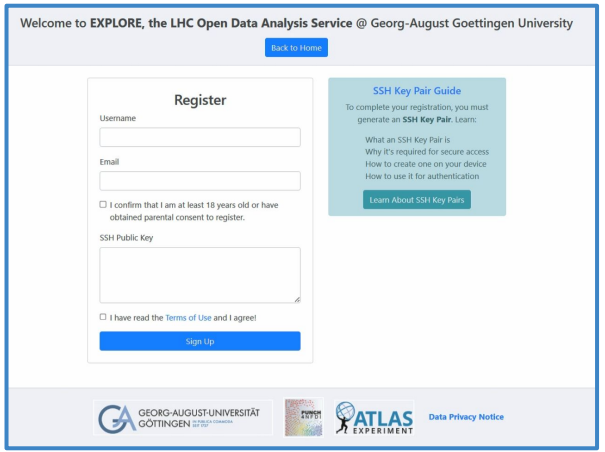
🔍 Fill out form (with valid Email)

🔑 Submit SSH key pair

✅ Receive confirmation & access instructions

💻 Log in via Submission Node:

- ssh -i .ssh/id_rsa <username>@punchlogin.goegrid.gwdg.de



👉 Register Now: https://punchlogin.goegrid.gwdg.de

⚙️ Part III – Hands-on: $t\bar{t}$ Analysis on EXPLORE

EXPLORE: top-antitop Analysis in the l+jets channel

Top–antitop ($t\bar{t}$) production is a key process at LHC:

- Test the Standard Model (QCD at HE)
- Search for new physics beyond the SM
- Background for Higgs and BSM studies

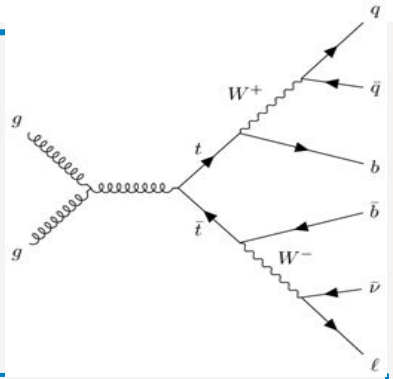
Educational value:

- Realistic yet manageable complexity
- Rich variety of physics objects: leptons, jets, MET
- Ideal for teaching event selection, reconstruction, and statistical analysis

l+jets Channel:

Final State:

- 1 high-p□ lepton (e/μ)
- ≥4 jets (2 b-jets, 2 from W → qq′)
- Missing transverse energy (ν from W → ℓν)



Advantages:

- Large statistics, clean event signature
- Balance between complexity and yield

Datasets:

- **ATLAS Open Data 2025 Beta Release** (>100M events across samples)
- Includes MC simulations for validation and comparison

EXPLORE: Accessing to EXPLORE & Exploring Environment

Not allowed on EXPLORE login node

❌ Compiling or testing code.

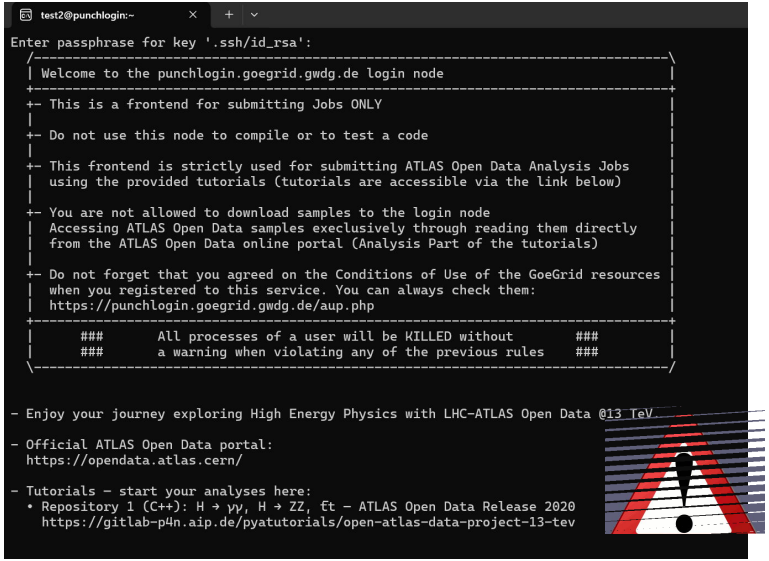
❌ Installing new software/packages.

❌ Downloading ATLAS samples to the login node.

❌ Opening a web browser / GUI apps.

Access tutorials & docs: Open links to tutorials repositories on your local laptop's browser.

https://gitlab-p4n.aip.de/pyatutorials/atlas-outreach-cpp-framework-13tev-release-25.git



*Open tutorials links in your **local** browser- no web browser on the login node.*

EXPLORE: Accessing to EXPLORE & Exploring Environment

Quick checks after login (optional)

lscpu | head -5 # confirm CPU info

Where am I? whoami; hostname; pwd

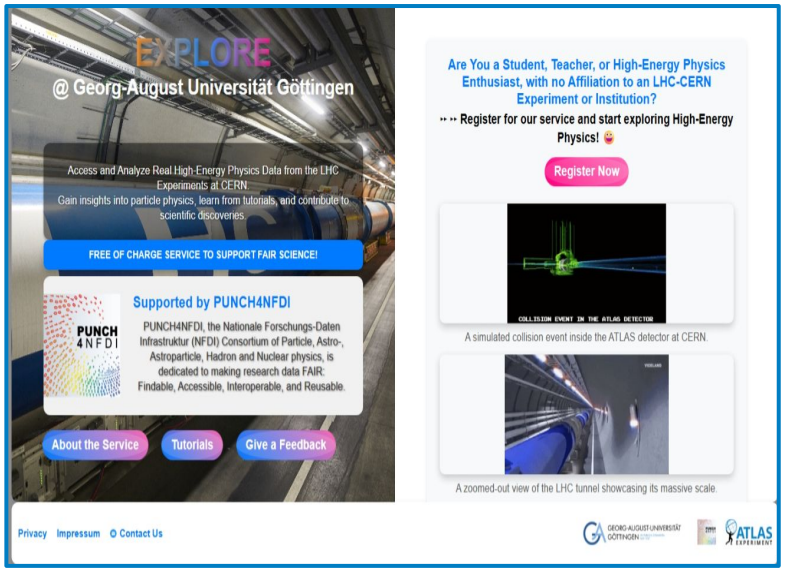
Shell & basics: echo $SHELL; echo $0

Account & data retention

Accounts last 3 months → download your job results before expiry.

Get results off the system →

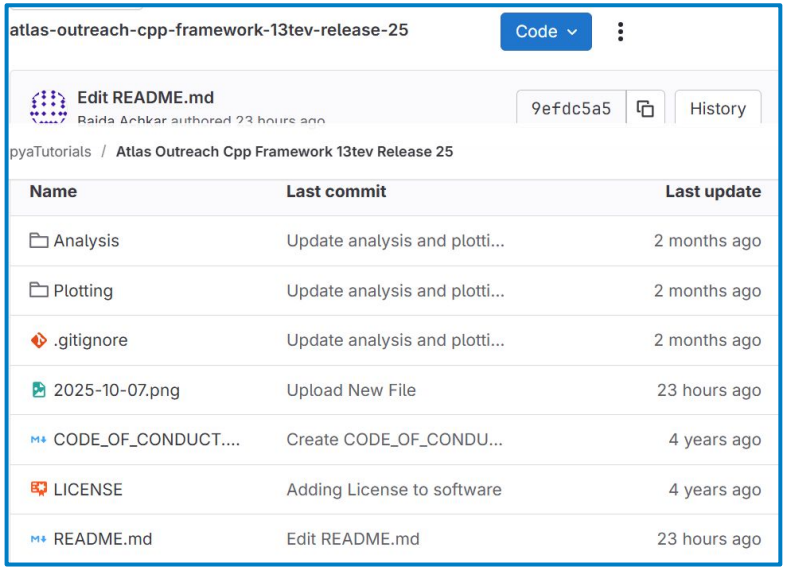
scp -r <username>@punchlogin.goegrid.gwdg.de:/path/to/results ./local_folder

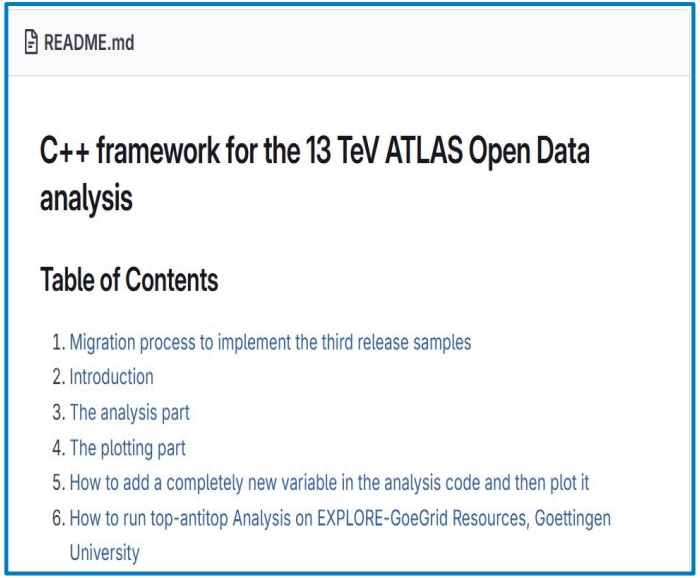


EXPLORE: TTbarAnalysis Tutorial Repository

- Using **your web browser on your laptop** go to the tutorial GitLab repository: [pyaTutorials / Atlas Outreach Cpp Framework 13tev Release 25 · GitLab]
- Go to the section: *How to run top-antitop Analysis on EXPLORE-GoeGrid Resources, Goettingen University and follow the steps:*
- **Step 1: Clone Framework**
  - git clone https://gitlab-p4n.aip.de/pyatutorials/atlas-outreach-cpp-framework-13tev-release-25.git
  - *cd ~/atlas-outreach-cpp-framework-13tev-release-25/Analysis/TTbarAnalysis*
  - *mkdir logs # Collect information about job execution*
- **Step 2: Submit Job to the HTCondor Batch System**
  - *condor_submit C4P_TTbarAnalysis.jdl*
  - *condor_q # Monitor status*
  - *tail -f logs/*.out | tail -f logs/*.err*

TTbarAnalysis Tutorial Repository @GitLab





https://gitlab-p4n.aip.de/pyatutorials/atlas-outreach-cpp-framework-13tev-release-25.git

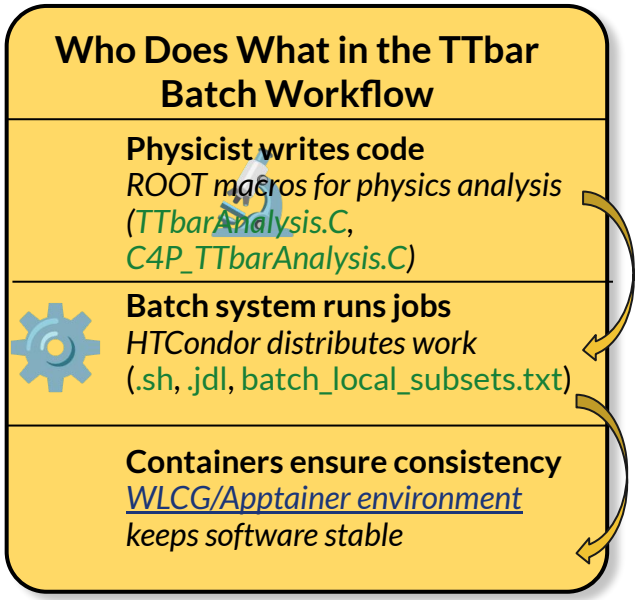
Analyzing with EXPLORE Resources

/Analysis

Component	Naming	Purpose / Function
4 Sub-folders	***NNAnalysis (TTbar, HWW, TTbarDilep, ZTauTau)***	Organizes analysis files for each physics process.
Main Code	***NNAnalysis.C***	Executes the analysis logic & produces histograms.
Main Header	***NNAnalysis.h***	Provides access to input data and declares variables.
Histogram Header	***NNAnalysisHistograms.h***	Ensures consistent histogram naming and structure across analyses.
Control Code (Batch Mode)	***C4P_TTbarAnalysis.C***	Manages input samples for batch analysis. Paths adjusted per analysis.
Executable Script	***C4P_TTbarAnalysis.sh***	Sources environment & runs the analysis code in the batch system.
Job Description File (.jdl)	***C4P_TTbarAnalysis.jdl***	Specifies resource requirements, log files, I/O transfer, & singularity container for safe execution.
Sample List(s)	***Samples_list.txt***	**Ensure that listed samples exist & are available on the ATLAS Open Data portal.**

How the TTbarAnalysis Runs in Batch Mode

Layer	What it does	Example file
Physics analysis	Reads events, applies selection, fills histograms	TTbarAnalysis.C
Analysis driver	Prepares one dataset, builds a ROOT chain, runs physics code	C4P_TTbarAnalysis.C
Job control	Starts ROOT for each dataset inside a container	C4P_TTbarAnalysis.sh + C4P_TTbarAnalysis.jdl



TTbarAnalysis: Job Executable

🖥️ *Worker node loads environment*
→ runs ROOT
→ writes to Output_TTbarAnalysis.

⬆️ *Condor sends outputs back to submit host.*

```
#! /bin/bash
set -e
#------------------------------------------------------------
# Mild XRootD client hardening (respected by ROOT's xrd client)
#------------------------------------------------------------
export XRD_CONNECTIONWINDOW=30
export XRD_REQUESTTIMEOUT=60
export XRD_REDIRECTLIMIT=2
export XRD_STREAMTIMEOUT=120
# ------------------------------
# Environment & Timing Setup
# ------------------------------
start_time=$(date +%s)
start_time_readable=$(date)
sample=$1
# Setup Output Directory
# ------------------------------
mkdir -p Output_TTbarAnalysis
# Load ROOT (ATLAS LCG Environment)
# ------------------------------
source /cvmfs/sft.cern.ch/lcg/views/LCG_105/x86_64-el9-gcc11-opt/setup.sh
# Execute Analysis
# ------------------------------
root -b -q -x "C4P_TTbarAnalysis.C+(\"$sample\")"
# Timing and Completion Report
# ------------------------------
end_time=$(date +%s)
end_time_readable=$(date)
duration=$(( (end_time - start_time + 30) / 60 ))  # Rounded to nearest minute
```

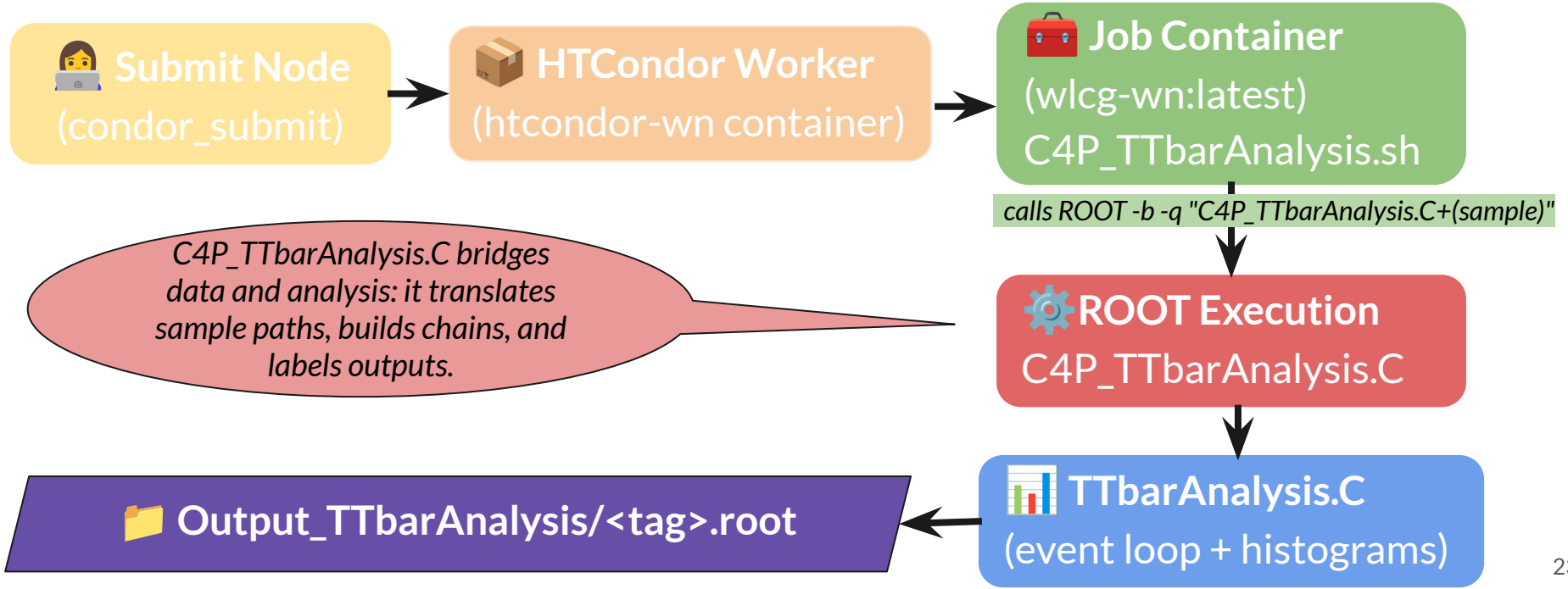
Job Description File (.jdl)

🗒️ *JDL → defines 'what' and 'how' your job runs.*
📦 *Condor moves scripts → executes → returns results.*

Typical resources:
1 CPU, ~1–2 GB RAM, few GB disk

```
executable = C4P_TTbarAnalysis.sh
# where to store log files
output = logs/$(cluster).$(process).out
error = logs/$(cluster).$(process).err
log = logs/cluster.log
arguments = $(sample)
#Transfer files to the batch node & later on back to the submit host, if your job was successful
ShouldTransferFiles = YES
WhenToTransferOutput = ON_SUCCESS
transfer_input_files = TTbarAnalysis.C, C4P_TTbarAnalysis.C, TTbarAnalysis.h, TTbarAnalysisHistograms.h, Samples_list.txt
transfer_output_files = Output_TTbarAnalysis
# Resources (right-sized for single-thread ROOT)
request_cpus           = 1
request_memory         = 1500
request_disk           = 2000
+MaxRuntime            = 3600
# In which container your job should be executed.
+SINGULARITY_JOB_CONTAINER = "wlcg-wn:latest"
# —- See: Compute4PUNCH / Container Stacks · GitLab
# Participants specify which batch file to use via:
#   condor_submit -a "BATCHFILE=batch_01.txt" C4P_TTbarAnalysis.jdl
queue sample from $(BATCHFILE)
```

Execution Chain in Action



Why Containers Matter?

Concept	Explanation
🧱 Reproducibility	Same ROOT, same compiler, same LCG stack everywhere.
🔒 Security	Jobs isolated - no interference between users.
🔁 Portability	Can run the same analysis at CERN, university, or cloud site.
⚡ Performance	Local data mounted directly → fast I/O from /atlas-data.

Local Data Mode 🟢 **/atlas-data → NFS bind**
Remote Mode 🌐 **https:// → root:// (converted automatically)**

The driver macro detects which mode to use — physicists don't need to change their code.

Plotting with EXPLORE Resources

/Plotting

Component	Naming	Purpose / Function
Main Code	*Plotting.cxx*	Controls plotting logic & automatically determines what to do for each analysis.
Main Header	*Plotting.h*	Provides definitions & declarations required for the plotting process.
Helper (Histo Control)	*list_histos / HistoList_ANALYSISNAME.txt*	Specifies which histograms are selected for plotting in each analysis.
Helper Directory (Input Files)	*inputfiles / Files_ANALYSISNAME.txt*	Ensures consistency & correctness of input samples for each analysis.
Executable Script (.sh)	*C4P_PlotTTbar.sh*	Sources environment, creates 'histograms/NNAnalysis' folders, & calls the main code for the selected analysis.
Job Description File (.jdl)	*C4P_PlotTTbar.jdl*	Sets job requirements, manages log files, I/O transfers, & runs inside a Singularity container for safety & consistency.

TTbarAnalysis: Plotting Part and Downloading Results

- **Step 3: Plotting Results**
  - *cd ~/atlas-outreach-cpp-framework-13tev-release-25/Plotting*
  - *mkdir logs*
  - *condor_submit C4P_PlotTTbar.jdl*
  - Output: histograms/TTbarAnalysis
  - Download with SCP to your Laptop:
  - *scp -r -i ~/.ssh/id_rsa <username>@punchlogin.goegrid.gwdg.de:~/atlas-outreach-cpp-framework-13tev-release-25/Plotting/histograms ./*
  - Versioning tip: mv Output_TTbarAnalysis Output_TTbarAnalysis_$(date +%Y%m%d_%H%M)

- **Step 4:** Visualizing Histograms on your Laptop
- **Step 5:** Optional: Use the parse_job_logs.py python script to get analysis jobs summary: per-sample durations, yields, anomalies if you would like. Do not forget to change the job id in the script to match your job's id.

Understanding HTCondor log files, Troubleshootings

- **Per job files:**
  - logs/<cluster>.<proc>.out — stdout (your prints + summary)
  - logs/<cluster>.<proc>.err — stderr (errors/tracebacks)
  - logs/<cluster>.<proc>.log — Condor event log
- **Key event codes:** 000 Submit, 001 Execute, 040 Transfer-out, 005 Terminated, 006 Image size
- **History:** condor_history -limit 20; condor_history <ClusterId> -long | less
- *No output? Check .err/.log for missing input or path typos*
- *Job in Idle? Queue busy or JDL mismatch - wait or resubmit*
- *SSH/SCP issues? Verify key (-i), username, remote paths*
- *Avoid overwriting: rename output dirs before re-runs*

Reflections: Physics, workflows, operations

- Which selection step impacts yield most?
- Compare data vs MC shapes - where do they agree/disagree?
- If you tighten lepton pT or b-tag criteria, how do histograms change?
- How did parallel submission affect turnaround time?
- Which Condor events appeared most (000, 001, 040, 005, 006)?
- What did you find in .out vs .err vs .log for a long job?
- If a job fails, which log do you check first and why?
- How to rerun only failed samples? How to version outputs?

🧩 Part IV - Optional Exercise: ATLAS PHYSLITE Analysis Tutorial

ATLAS Physlite Workflow - Repository Setup & Submission

Goal: Learn how to run an ATLAS Open Data PhysLite analysis in both ROOT (C++) and Python workflows using batch submission on an HTCondor pool.

Input Sample:

root://eospublic.cern.ch//eos/opendata/atlas/rucio/mc20_13TeV/DAOD_PHYSLITE.37620644._000012.pool.root.1

Outputs:

outputs/electron_pt_<Cluster>_<Proc>.png
outputs/analysis_output_<Cluster>_<Proc>.root
logs/root_execution_<Cluster>_<Proc>.log

git clone <physlite_repo> # physlite-on-GitLab

cd physlite-for-atlas-tutorial-2025

mkdir logs

condor_submit submit_remote.jdl

condor_submit submit_remote_python.jdl

condor_q; tail -f logs/.out*

Results & Performance Tasks

Produced Artifacts

Each Condor job automatically returns:

- **ROOT file** → analysis_output.root (histograms)
- **PNG plots** → outputs/*.png (electron pT, ΔR, cutflow, etc.)
- **Summary text** → outputs/analysis_summary.txt
- **Log file** → logs/root_execution.log

Quantity	Value	*Comment / Interpretation*
Total events processed	_____	*Check matches dataset size*
Electrons before selection	_____	*Number of reconstructed e^- in all events*
Electrons after selection	_____	*Should be fewer after applying cuts*
Loop CPU time (s)	_____	*Compute performance metric*
Real (time wall-clock, s)	_____	*Indicates I/O overhead*
Electron pT peak (GeV)	_____	*Approximate from histogram*
Average ΔR(jet,e)	_____	*Check jet–electron isolation*
Number of plots	_____	*Ensure full set produced*

Physlite - GitLab Repository Structure

physlite-for-atlas-tutorial-2025/

```
├── physlite_full_analysis.C          # Main ROOT analysis macro
├── physlite_full_analysis.py         # Equivalent Python analysis
├── run_remote.sh                     # Executable for ROOT batch jobs
├── run_remote_python.sh              # Executable for Python jobs
├── submit_remote.jdl                 # HTCondor job description (ROOT)
└── submit_remote_python.jdl          # HTCondor job description (Python)
```

Takeaways & Reflections

- You now know how to:
  ✅ Run C++ and Python PhysLite analyses in batch mode
  ✅ Use HTCondor to manage parameter sweeps
  ✅ Keep a clean output structure (outputs/ & logs/)
  ✅ Handle OpenData files via both HTTPS and XRootD
- These are the same principles used in real ATLAS distributed analysis workflows.

1. What is the difference between HTTPS and XRootD file access?
2. Why separate outputs and logs directories?
3. *Applied Exercise - Batch Parameter Sweep*

Edit your submit_remote.jdl file to launch four parallel jobs that vary the electron pT threshold. Insert the following lines:

```
arguments = "<URL> outputs analysis_output.root $(ELPT)"

queue ELPT from (25 30 35 40)
```

What does each variable value represent, and how will Condor handle these jobs?

🌍 Part V - From Open Data to Open Science

EXPLORE - Key Takeaways

✔ **Removes barriers** to ATLAS Open Data access and use.

✔ **Open to everyone:** no CERN or institutional login required.

✔ **Ready-to-run environment:** containers + distributed computing.

✔ **Simplifies analysis:** complex setups become batch workflows.

✔ **Scales up:** handles compute-intensive studies easily.

✔ **Bridges theory and practice:** from Open Data to real analysis.

✔ **Empowers learning:** Open Data for students, teachers, and researchers.

✔ **Hands-on impact:** you ran real analyses, used reproducible workflows, and built professional-level skills.

Acknowledgments & Links

- **EXPLORE Registration:** https://punchlogin.goegrid.gwdg.de/
- **PUNCH4NFDI Website:** https://www.punch4nfdi.de/
- **ATLAS Open Data Portal:** https://opendata.atlas.cern/
- **Tutorial GitLab Repository:** https://gitlab-p4n.aip.de/pyatutorials

This work was [in part] supported by DFG fund „NFDI 39/1“ for the PUNCH4NFDI consortium.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation

PUNCH 4NFDI

Time for Questions

THANK YOU

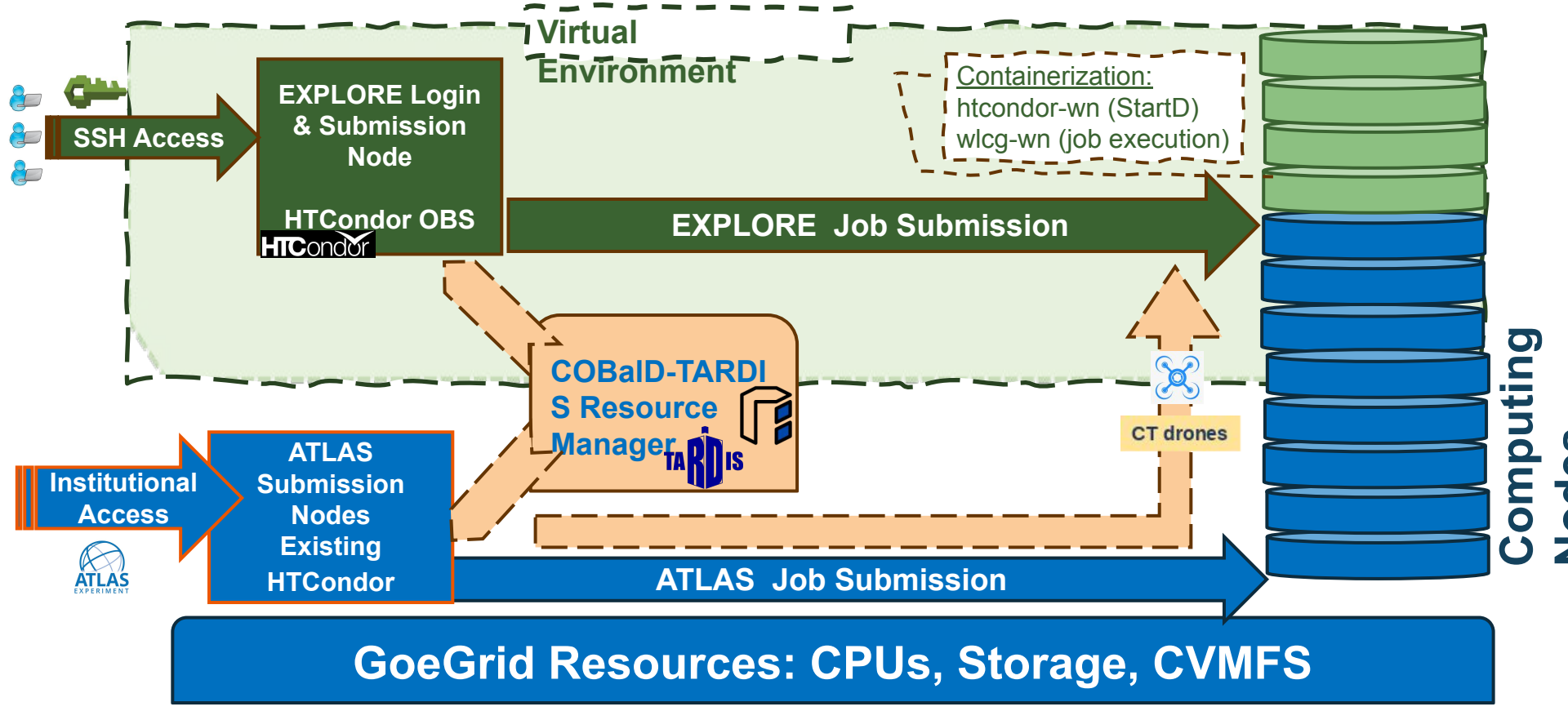
Acknowledgements

This work was [in part] supported by DFG fund „NFDI 39/1“
for the PUNCH4NFDI consortium.

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

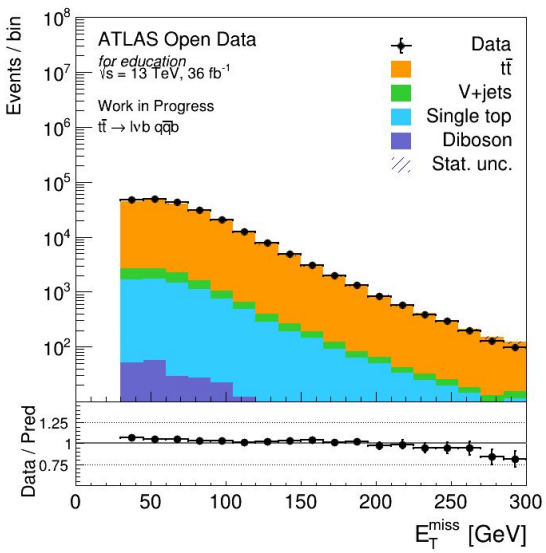
Backup

EXPLORE setup within GoeGrid cluster-Göttingen

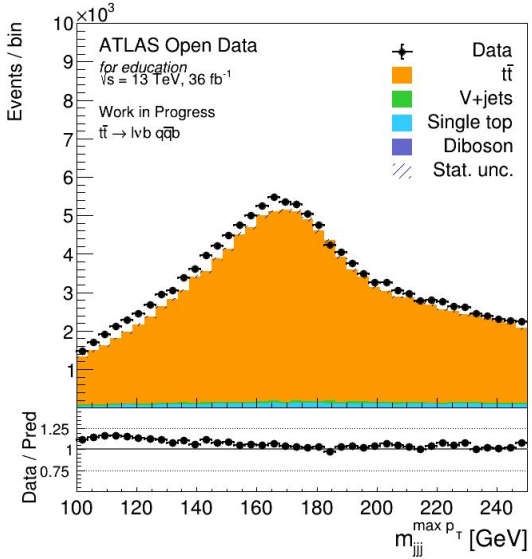


Top-Antitop Analysis Plots

Missing transverse energy of the leading b-jet



Invariant mass of the three leading jets



Transverse momentum