



# 2024 Plan of Work

Stacks  $\equiv$  {SPI, Key4hep}

André Sailer

CERN-EP-SFT

SFT Group Meeting  
January 15, 2024

# Table of Contents



- 1 Stacks: Team Members
- 2 SPI: Software Process and Infrastructure
  - Deliverables
  - Activities in 2023
  - Plans for 2024
- 3 Key4hep
  - Developments in 2023
  - Plans for 2024



# Section 1:



## 1 Stacks: Team Members

# Team Members



		2023	2024	
A Sailer	STAF	100%	100%	S,K,I <sup>a</sup>
I Goulas	STAF	100%	100%	S
D Konstantinov	PJAS	50%	25%	S
O Morud	TECH	66%	/	S
T Ehmann (+ N.N.?)	TECH	33%	66% (+33%)	S
S M Muzaffar	STAF	10%	10%	S
B Hegner	STAF	70%	70%	K
G Stewart	STAF	15%	15%	K
J Carceller <sup>b</sup>	FELL	92%	100%	K
S Sasikumar <sup>b</sup>	FELL	83%	100%	K
M Fila <sup>b</sup>	FELL	25%	100%	K
L Reichenbach <sup>c</sup>	DOCT	83%	100%	K
L Valentini <sup>d</sup>	TECH	92%	25%	I

<sup>a</sup> S(PI), K(ey4hep)/Framework, I(LCDirac)

<sup>b</sup> EP R&D funding      <sup>c</sup> Gentner funding

<sup>d</sup> EURIZON funding until 01/24, then Future Collider

## Section 2:



### 2 SPI: Software Process and Infrastructure

- Deliverables
- Activities in 2023
- Plans for 2024

Providing compilers and consistent software stacks: 750+ packages

- For a large set of architectures, operating systems, compilers
  - ▶  $\text{len}((x86, ARM) \wedge (EL7, 8, 9, mac11, \dots) \wedge (gcc11, \dots, clang16, \dots) \wedge (opt, dbg) \wedge (dev3, \dots)) \approx 50$
- ROOT, Geant4, MC Generators, ML packages, ...
- Every night (except Sundays) to CVMFS
- LCG releases and experiment specific stacks (“layers”) are provided on CVMFS and as RPMs
- Librarian and Integrators Meeting (“LIM”) every 2 weeks to discuss and decide on the content of nightlies and release dates
- Information/Documentation: <https://spi.web.cern.ch/>, <https://lcginfo.cern.ch>,  
<https://lcgdocs.web.cern.ch>

# Stakeholders and their Consumables



- Known Customers: ATLAS, LHCb, SWAN, CERN-BE (“NXCALS”), CERN-IT, WLCG, NA61, NA62,
- There are also users that we do not know that pick this up from CVMFS (in and outside of CERN)
- Partial stacks provided for development builds for specific projects
  - Geant4, VecGeom, AdePT

	ATLAS	LHCb	SWAN	BE	SME
Default Nightlies (CVMFS)	✓				✓
Special Nightlies (CVMFS)		✓	✓	✓	
Releases (CVMFS)	✓		✓		✓
Releases (RPMs)	✓	✓			

# Plan of Work Items 2023



## ■ Plan of Work 2023 presentation

- The current infrastructure and build, test, deployment system needs maintenance and consolidation
  - Jenkins 🚩
  - Move to Alma9 🚩
  - Fully integrate aarch64 Openstack VMs 🚩
- Build System
  - Clean recipes 🚩
  - Investigate further optimisations of re-use of binaries to minimise redundant compilation 🚫
  - Prototype support for micro-architecture levels builds (x86\_64\_{v2,v3,v4}) 🚫
- Move to using CernVM-FS Gateway 🚩
- Investigate use of S3 for caching binary artifacts 🚫
- Provision of baseline-system-definition containers 🚩
- Finalize inclusion of roottest in regular tests 🚫
- Workshop with stakeholders and users to get feedback 🚫
- A firm step forward towards Spack adoption 🚩



# Main Deliveries in 2023



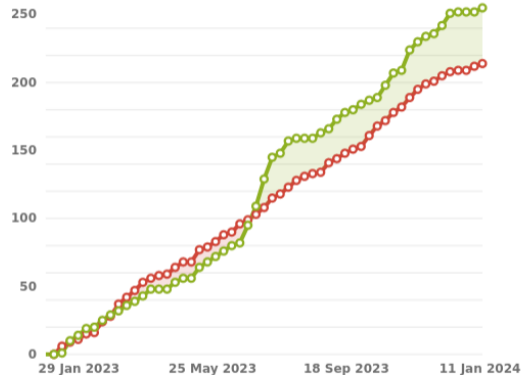
## Deliveries:

- LCG\_103: February 2023: ROOT 6.28/00 , 688 packages in x86\_64-centos-gcc11-opt
- LCG\_104[a,b]: August 2023: ROOT 6.28/04, 760 packages in x86\_64-centos-gcc11-opt
- LCG\_105: December 2023: ROOT 6.30/02, 791 packages in x86\_64-centos-gcc11-opt
- Started to provide stacks on AlmaLinux9, and 8 (dropping builds on CentOS8/9)
- Integrated aarch64 VMs to build aarch64 stacks on different OS (CentOS7 and AlmaLinux9)
- Integrated gcc 13 and clang 16 and c++20
- Layers for ATLAS and LHCb
- Core-docker containers for stack users

# Issues: Jira, ServiceNow, Email



- Requests for new packages or updates
- Support for users when things go wrong
- New layers
- Improvements in our own systems



**Issues in the last 377 days** (grouped weekly) [View in Issue Navigator](#)

- Created issues (214)
- Resolved issues (255)

# “Backend” Improvements in 2023



- Build machines migrated to AlmaLinux8
- Cleanup of outdated nightlies, views, and packages from `/cvmfs/sft-nightlies`
  - ▶ Including fixed garbage collection job
- [lcginfo.cern.ch](https://lcginfo.cern.ch) bug fixes and improvements
- Jenkins updates



Have not yet build based on spack that is deployed every night to CVMFS

- S3 (CERN openstack) buildcache integrated: push and pull packages to re-use and not re-build
  - ▶ Clean-up strategy to be defined
- Spack continues to be successfully used for Key4hep stacks, benefiting from synergy

# Plans for 2024

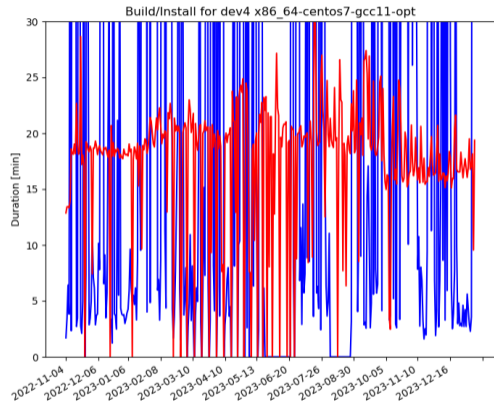


- Continue to provide compilers and consistent software stacks for our stakeholders
  - Integrate new packages and updates, provide user support
- Maintain build infrastructure: Build machines, Puppet recipes, storage
  - CentOS7 EoL in June 2024; migration of some workflows, e.g., rpmbuild
- JIRA migration
- Adapt CVMFS-gateways for parallel publications
- Decide whether or with what to replace LCGCMake

# CVMFS Publication Speed



- Publishing 50 nightly stacks sequentially takes a very long time. Some nightlies only deployed after the work day is over
  - ▶ Though something made things faster in October 2023 it seems
- On going developments to move to the CVMFS gateways to parallelize publication
  - ▶ But: need to change directory hierarchy
    - ★ From “package/version/platform”
    - ★ To “platform/package/version”
  - ▶ to allow packages for different architectures to be published in parallel
    - ★ Publishing package by package is too slow
    - ★ Changing this is a bit cumbersome



Build times and Publication times for one stack from day to day. Build times mostly depend on the number of changed packages in the stack.

# CVMFS Publication Speed



- Publishing 50 nightly stacks sequentially takes a very long time. Some nightlies only deployed after the work day is over
  - ▶ Though something made things faster in October 2023 it seems
- On going developments to move to the CVMFS gateways to parallelize publication
  - ▶ But: need to change directory hierarchy
    - ★ From “package/version/platform”
    - ★ To “platform/package/version”
  - ▶ to allow packages for different architectures to be published in parallel
    - ★ *Publishing package by package is too slow*
    - ★ Changing this is a bit cumbersome

Installation Time Comparison: Full Platform vs. Package-by-Package on CVMFS



Installation time of a specific stack using CVMFS gateways. The full stack, or all its packages at a time. No view creation. Installation for all packages, not the usual incremental installation.

# LCGCMake Replacement: Spack(?)



- LCGCMake requires a lot of effort to maintain package recipes, update packages, ensure consistency in package dependencies
- *Spack* has a larger contributor base
  - Package recipes, updated versions, consistency checks done by its community
  - Need to balance against danger of things moving to fast and breaking too often
  - Easier to change deployment locations (cf. above)
  - Support for different (Micro-)architectures
- **Agreed with LHCb to provide (partial) LCGStack by middle of February and then re-evaluate feasibility**
- Keep providing RPMs to ATLAS regardless of system used for building



# Other Objectives



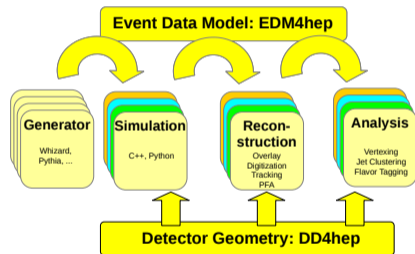
- Jenkins
  - ▶ Move to Java 18
  - ▶ Clean-up obsolete jobs: LCG\_88b is no longer needed
- Inclusion of roottest in regular tests

# Section 3:



- 3 Key4hep
  - Developments in 2023
  - Plans for 2024

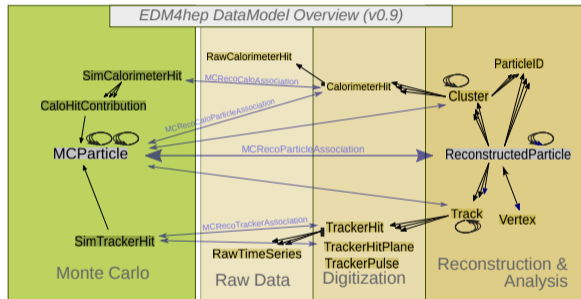
- Turnkey software for future accelerators
- Share components to reduce maintenance and development cost and allow everyone to benefit from its improvements
- Complete data processing framework, from generation to data analysis
- Supported by EP R&D WP7 Phase 1 → Framework developments in Phase II



## Plan of Work 2023 presentation

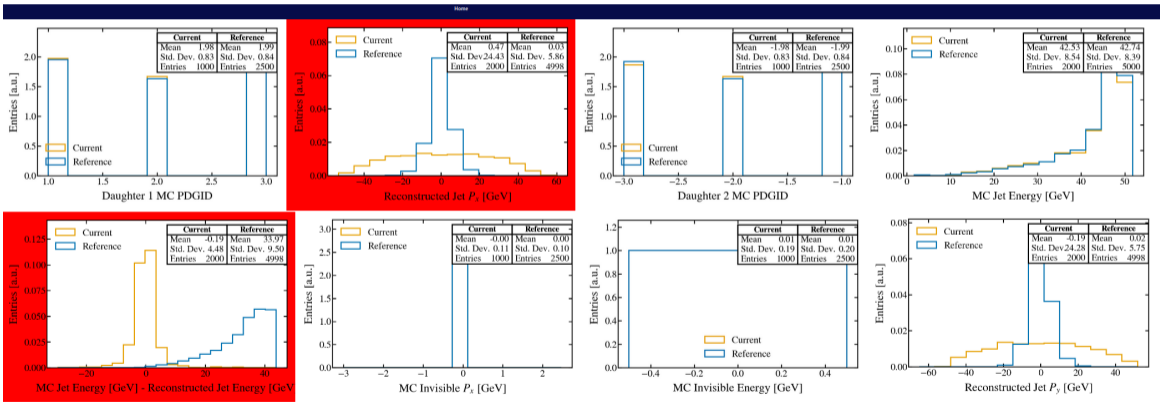
- Implementation of a continuous validation framework for Key4hep 🚧
- Continue with integration of ACTS and CLUE (+PandoraPFA) reconstruction packages 🚧
- Developments for multi-threading with Gaudi 🚧
- Further consolidate and synergize simulation approaches, also with respect to Gaussino 🚧
  - ▶ FCC decided to use *ddsim* in near term
- Improve documentation in structure and content 🚧

- Schema evolution: it is possible to modify our definitions in EDM4hep and still be able to read old data
  - ▶ Renaming, adding a new member, removing a member is now possible
  - ▶ Leverages schema evolution support from ROOT
- New RNTuple backend to write RNTuples (experimental new format for ROOT files)
- Python bindings for EDM4hep
- Few minor changes in the model itself → very stable



- The *Frame* (from podio) is a data container where collections can be stored → Key for multi-threading support
- Automatic *Frame* reading and writing was introduced to the Key4hep Gaudi algorithms
- Changes were transparent; users did not need to modify their code
  - Except when reading files “manually”
- Advanced investigations into “functional” algorithms

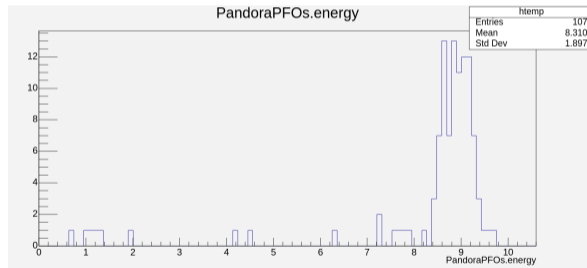
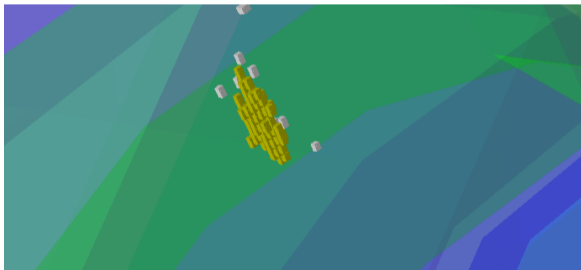
- Daily check to ensure simulation and reconstruction chains not negatively affected
- <https://key4hep-validation.web.cern.ch/>



# Integration of External Reco Tools



- k4ACTSTracking developments restarted
- k4CLUE creating clusters (E. Brondolin)
- PandoraPFA adapted to LAr calorimeter (below)







- Key4hep software stack build with Spack, provided on CVMFS for CentOS7, Ubuntu22, Alma9
  - ▶ Nightlies on [/cvmfs/sw-nightlies.hsf.org](https://cvmfs/sw-nightlies.hsf.org)
  - ▶ Releases on [/cvmfs/sw.hsf.org](https://cvmfs/sw.hsf.org), latest release 2023-11-23

# Plans for 2024



## Key4hep

- Bring the validation system into production
- ACTS track fitting
- DDGaudiPandora interface
- EDM4hep native Overlay background processing
- Validate detector models for use in physics studies

## Heterogeneous Frameworks

- Extract representative workflows of Gaudi-based applications. Construction of data flow and control flow graphs complemented with timing and memory size information. (started 2023)
- Prepare a single-node mock-up application of a realistic workload
- Evaluate computation offloading mechanisms and prototype a single-node Gaudi scheduler for heterogeneous resources

Thank you for your attention!