




HerdSoftware: an overview

Nicola Mori

on behalf of the Herd SW development team

Summary

-  Introduction
-  Requirements
-  Architecture
-  Workflow
-  Infrastructure
-  Resources



Introduction

- Software framework for the Herd collaboration
 - Shared development
 - Common “language” for better interaction (e.g. code circulation)
- Tasks:
 - Reconstruction
 - Analysis
 - Simulation



Introduction

- Boundary conditions:
 - Brand new experiment, no SW legacy
 - Still in design phase, no firm detector structure
 - Lifetime ~ 15-20 years
 - Potentially ~ hundred(s) users
 - Limited development manpower (w.r.t. e.g. collider experiments)



Requirements

- Modern programming languages and techniques for durability, correctness and maintainability
 - C++14 for core libraries
 - CMake for build, git (GitLab) for code repository
 - Right emphasis on documentation and testing



Requirements

- Flexibility to cope with the current experiment phase
 - Many possible detector layouts
 - Development/analysis activities still to be organized



Requirements

- Configurability for dealing with different computations and environments
 - Countless reconstruction, analysis and simulation scenarios
 - Suitable for working nodes, analysis workstations etc.



Requirements

- Future-oriented architecture
 - Design effort to create something that could be the “final” version of the Herd official software right from the start
 - Full exploitation of early development efforts



Requirements

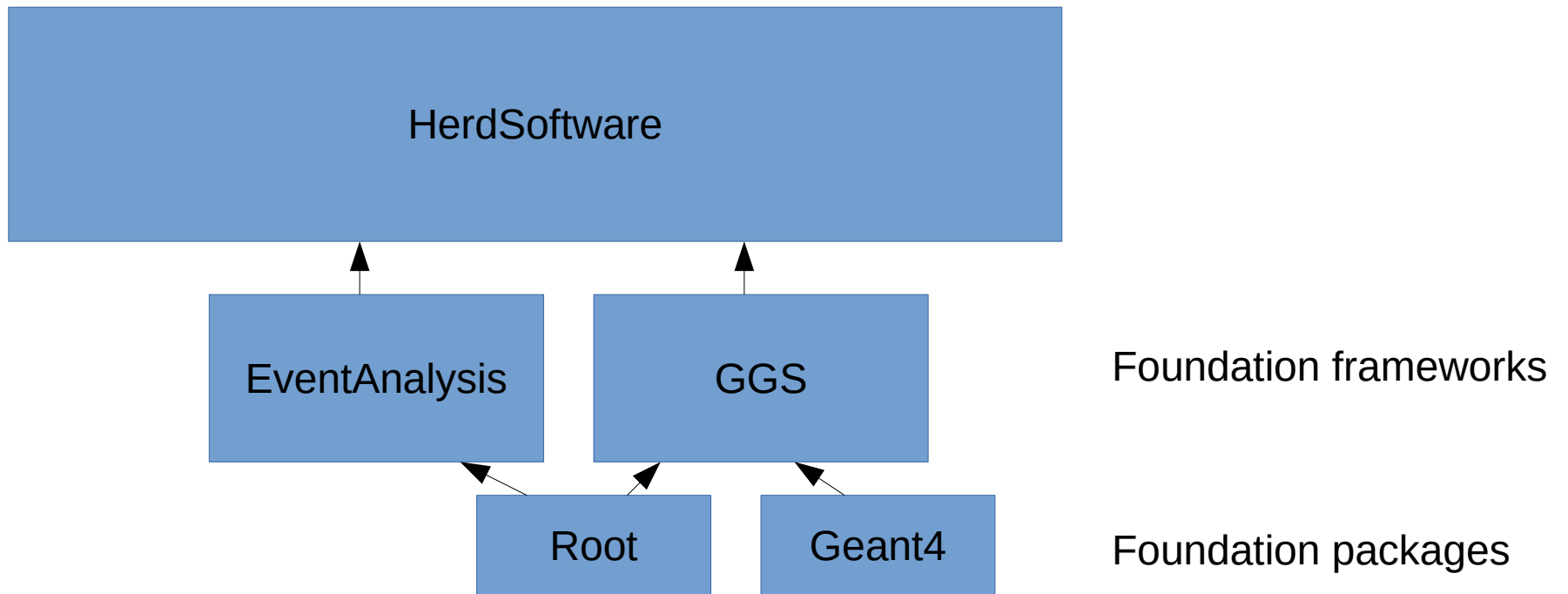
- Manageability
 - Must be handled by few people and used by many more
 - Easy code sharing
 - Shared maintenance
- Familiarity
 - Use common, well-known tools to boost productivity
 - Root, Geant4



Architecture

- Flexibility+manageability+future:
 - Base the software on generic, small and extensible external “foundation” frameworks
 - Profit from existing generic code for common HEP tasks (e.g. simulate and analyze event data)
 - Focus on Herd-specific developments within these foundation frameworks
- Simulation: GGS
- Analysis and reconstruction: EventAnalysis

Architecture



is a dependence for 



Architecture

- HerdSoftware is just made of libraries containing Herd-specific implementations of the components of the two generic foundation frameworks
 - i.e. C++ classes inheriting from the base classes of the foundation frameworks
- Herd components are “plugged” at runtime into the generic simulation/analysis workflows (executables) of GGS/EventAnalysis



Architecture

- GGS: Generic Geant4 Simulation
 - Ease the development and deploy of MC simulations for any kind of detector
 - Generic implementation of Geant4 user classes (hits, user actions)
 - Only geometry has to be mandatorily defined by the user
 - Extensible with custom implementations via plugin libraries in external projects (like HerdSoftware)
 - Persistence on Root files
 - Adapts to any detector geometry



Architecture

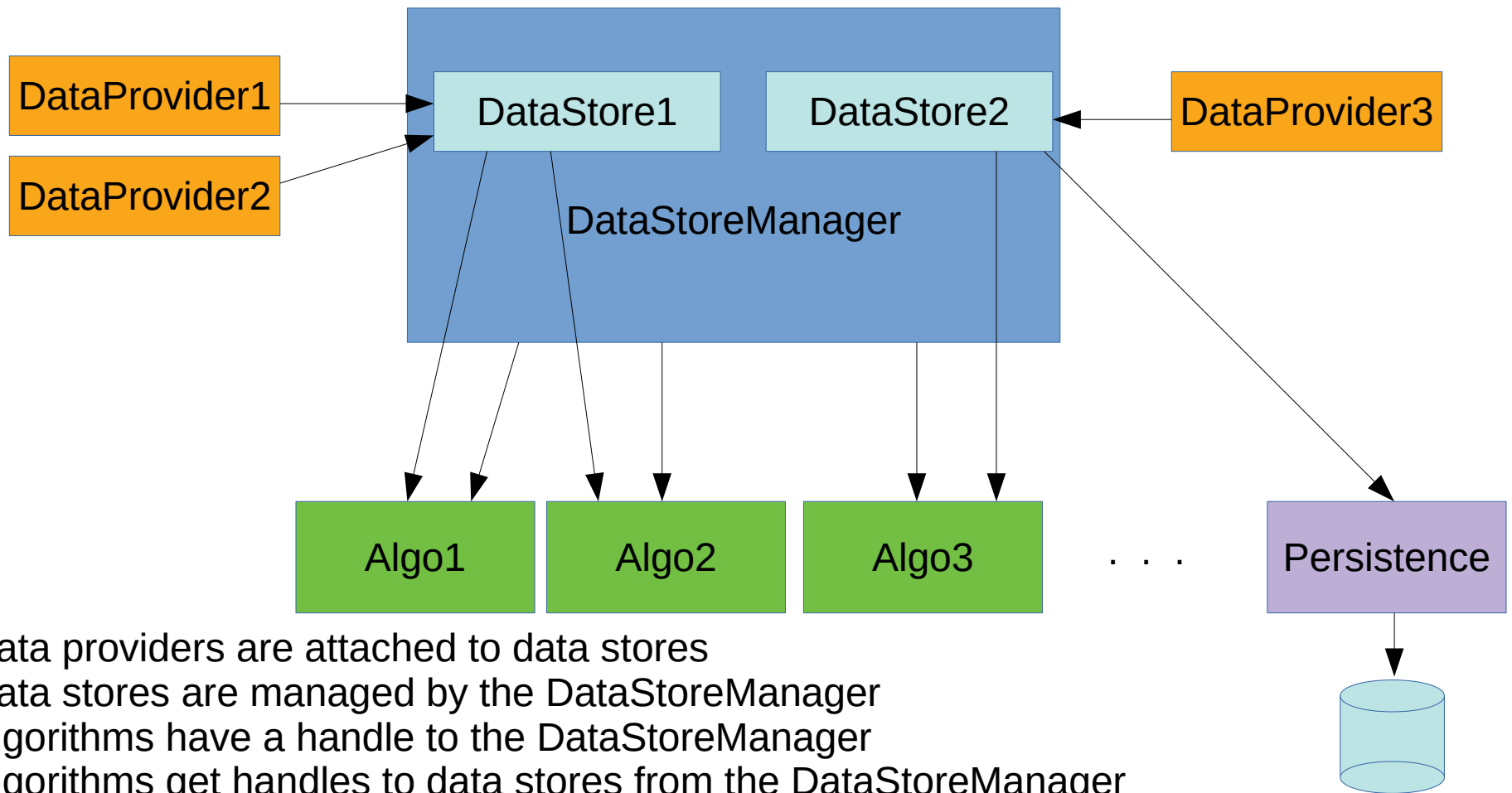
- GGS in HerdSoftware:
 - Herd-specific components developed as GGS plugin libraries:
 - “Parametric” detector geometry (modified baseline + tunable parameters)
 - Isotropic particle generator from a spherical surface
 - Easy to add more geometries/generators
 - Standard GGS components for hits, output, readout etc.



Architecture

- EventAnalysis:
 - Event-based data analysis framework
 - Implement the common HEP analysis workflow in a generic way
 - Initialization, event loop, save results etc.
 - Specific components (algorithms, data access etc.) implemented as plugins in external projects (like HerdSoftware)
 - Well-defined (analysis entity) ↔ (code component) correspondence and analysis flow
 - Useful facilities (e.g. runtime configuration, automatic I/O with Root)
 - Very lightweight w.r.t. similar packages (e.g. GAUDI)

Architecture



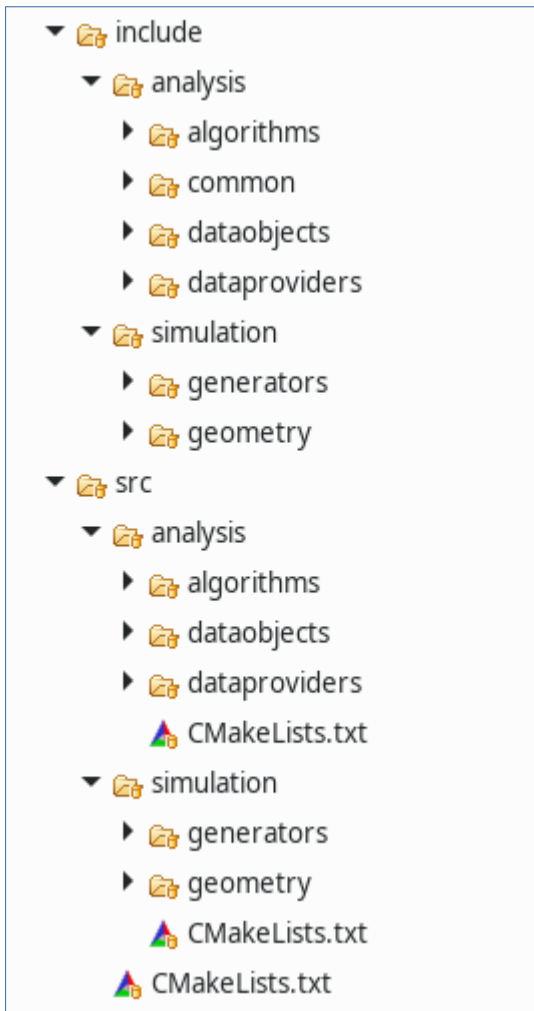
- Data providers are attached to data stores
- Data stores are managed by the DataStoreManager
- Algorithms have a handle to the DataStoreManager
- Algorithms get handles to data stores from the DataStoreManager
- Algorithms fetch (push) data objects from (to) data stores
- Data stores provide available objects, eventually asking the attached data providers
- Persistence store booked objects in file, db etc.



Architecture

- EventAnalysis in HerdSoftware:
 - Herd-specific components developed as EventAnalysis plugin libraries:
 - Algorithms
 - Data providers
 - Persistence
 - EventAnalysis standard components for job configuration and execution

Architecture



- Neat code structure
- Easy addition of new components
- Modular
 - Build only what you want
 - e.g. no simulation for a data analysis workstation



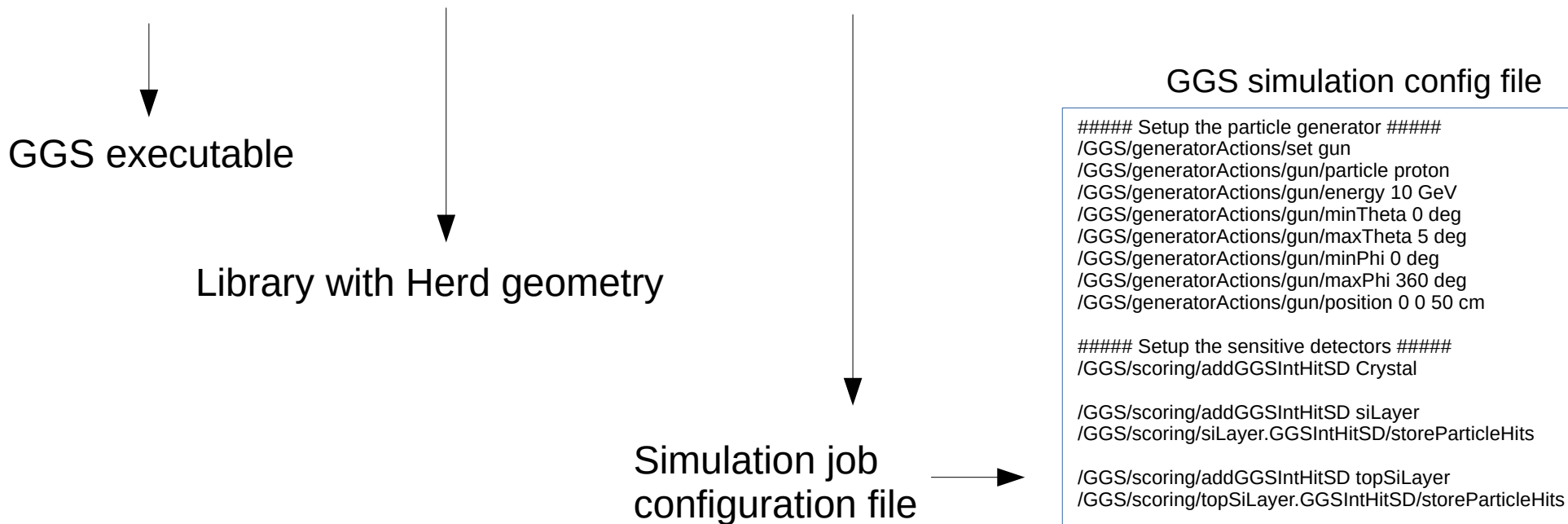
Workflow

- Develop Herd components in HerdSoftware
 - Or in a personal external project
- Build HerdSoftware plugin libraries
- Launch executables provided by foundation frameworks providing them a job configuration file and the plugin libraries with Herd components

Workflow

- **Simulation:**

```
$ GGSPenny -g libHerdParamGeo.so -d config.mac -ro output.root
```



Output file



GGS simulation config file

```
##### Setup the particle generator #####
/GGS/generatorActions/set gun
/GGS/generatorActions/gun/particle proton
/GGS/generatorActions/gun/energy 10 GeV
/GGS/generatorActions/gun/minTheta 0 deg
/GGS/generatorActions/gun/maxTheta 5 deg
/GGS/generatorActions/gun/minPhi 0 deg
/GGS/generatorActions/gun/maxPhi 360 deg
/GGS/generatorActions/gun/position 0 0 50 cm

##### Setup the sensitive detectors #####
/GGS/scoring/addGGSIHitSD Crystal

/GGS/scoring/addGGSIHitSD siLayer
/GGS/scoring/siLayer.GGSIHitSD/storeParticleHits

/GGS/scoring/addGGSIHitSD topSiLayer
/GGS/scoring/topSiLayer.GGSIHitSD/storeParticleHits

/GGS/scoring/addGGSIHitSD top_psd
/GGS/scoring/top_psd.GGSIHitSD/storeParticleHits

/GGS/scoring/addGGSIHitSD side_psd
/GGS/scoring/top_psd.GGSIHitSD/storeParticleHits

##### Setup the output on ROOT file #####
/GGS/userActions/addGGSHitsAction
/GGS/userActions/addGGSMCTruthAction

##### Shoot 100 particles #####
/run/beamOn 100
```

Workflow

- Analysis/reconstruction:

```
$ EventAnalysis -c config.txt
```

↓
EventAnalysis
executable

↓
Analysis job
configuration file →

EventAnalysis config file

```
Plugin HerdDataProviders
Plugin HerdAlgorithms
Plugin RootPersistence
Plugin HerdDataObjectsDict

DataProvider GGSDDataProvider ggsDataProvider simulated.root
AttachToStore evStore event
AttachToStore globStore global

Persistence RootPersistenceService rootPersistence output.root
Book caloHits event evStore
Book stkHits event evStore
Book stkGeoParams global globStore
Book psdHits event evStore
Book psdGeoParams global globStore

EventLoop
Algo StkGeometricDigitizerAlgo stkGeomDig
Set pitch 0.0150
Algo PsdGeometricDigitizerAlgo psdGeomDig
Set topAggrFactorX 4
Set topAggrFactorY 4
Set latAggrFactorXY 4
Set latAggrFactorZ 5
```

Workflow

Plugin libraries to be loaded

Setup of input/output

Analysis definition

EventAnalysis config file

```
Plugin HerdDataProviders
Plugin HerdAlgorithms
Plugin RootPersistence
Plugin HerdDataObjectsDict

DataProvider GGSDDataProvider ggsDataProvider simulated.root
  AttachToStore evStore event
  AttachToStore globStore global

Persistence RootPersistenceService rootPersistence output.root
  Book caloHits event evStore
  Book stkHits event evStore
  Book stkGeoParams global globStore
  Book psdHits event evStore
  Book psdGeoParams global globStore

EventLoop
  Algo StkGeometricDigitizerAlgo stkGeomDig
  Set pitch 0.0150
  Algo PsdGeometricDigitizerAlgo psdGeomDig
  Set topAggrFactorX 4
  Set topAggrFactorY 4
  Set latAggrFactorXY 4
  Set latAggrFactorZ 5
```



Infrastructure

- GitLab instance hosted by ReCaS computing center in Bari (Italy)
 - Git repository
 - Issue tracker
 - Continuous integration
 - Wiki with documentation
- Restricted access
 - Roles (guest, developer, maintainer etc.)

Infrastructure

The screenshot shows the main page of the HerdSoftware project on the ReCAS platform. The page includes a navigation sidebar on the left with options like Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (8), Merge Requests (0), CI/CD, Operations, Registry, Wiki, Snippets, and Settings. The main content area displays the project name 'HerdSoftware' with a lock icon and 'Project ID: 9'. It shows statistics for 189 commits, 1 branch, 0 tags, and 1.7 MB of files. A pipeline status bar indicates 'passed' and 'coverage 57.00%'. Below this is a commit history section showing a recent merge of branch 'version' into 'master' by Nicola Mori. At the bottom, there are buttons to add README, CHANGELOG, CONTRIBUTING, and a Kubernetes cluster, along with a CI/CD configuration button. A table lists the project's files and their last commit details.

Name	Last commit	Last update
doc	doc: do not generate doxygen documentation in ...	3 weeks ago
examples	examples: small fix for README file.	2 weeks ago
include	Merge branch 'example' into 'master'.	1 day ago
src	Merge branch 'version' into 'master'.	1 day ago
unit-tests	Merge branch 'simulation' into 'master'.	1 day ago
.clang-format	Add code formatting style for clang-format.	2 months ago
.gitlab-ci.yml	CI: fix docker image name for deploy-doc job.	3 weeks ago

Infrastructure

The screenshot shows the ReCAS Issues page for the HerdSoftware project. The left sidebar contains navigation options: Project, Repository, Issues (8), List, Board, Labels, Milestones, Merge Requests (0), CI / CD, Operations, Registry, Wiki, Snippets, and Settings. The main content area displays a list of 8 open issues. Each issue entry includes a title, ID, author, labels, and the time it was last updated.

Issue ID	Title	Author	Labels	Updated
#30	Document how to create and run external projects	Nicola Mori	Documentation, To Do	updated 30 minutes ago
#29	Add documentation for simulation (0 of 3 tasks completed)	Nicola Mori	Documentation, To Do	updated 2 hours ago
#27	Add installation	Nicola Mori	Deploy, To Do	updated 30 minutes ago
#26	Clean the horrible ArrayForwarder mess	Nicola Mori	Data objects, To Do	updated 2 weeks ago
#25	Use CooArray, SidesArray etc. in data objects	Nicola Mori	Data objects, To Do	updated 3 weeks ago
#22	Write a shower clustering algorithm for the calorimeter	Valerio Formato	Algorithms, To Do	updated 4 weeks ago
#17	HoughTrackDisplayAlgo doesn't draw tracks on the XY view.	Valerio Formato	Algorithms, To Do	updated 1 month ago
#16	Write tests for GGSDDataProvider	Nicola Mori	Tests, To Do	updated 1 month ago

Issues: a list of features to implement and problems to fix

Infrastructure

The screenshot displays the ReCAS CI/CD interface for the 'HerdSoftware' project. The top navigation bar includes 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. A search bar is present on the right. The left sidebar shows navigation options: Project, Repository, Issues (8), Merge Requests (0), CI/CD, Pipelines, Jobs, Schedules, Charts, Operations, Registry, Wiki, Snippets, and Settings. The main content area shows a list of pipeline runs with columns for Status, Pipeline, Commit, and Stages. The status for all runs is 'passed'. The pipeline list includes details such as pipeline number, commit hash, commit message, stage status, duration, and time since completion.

Status	Pipeline	Commit	Stages	Duration	Time Ago
passed	#143 by latest	master -> bfd46f59 Merge branch 'version' i...	✓ ✓ ✓	00:04:50	1 day ago
passed	#141 by	master -> a5e39ca4 cmake: make EventAnaly...	✓ ✓ ✓	00:04:54	2 days ago
passed	#140 by	version -> 489f75e7 Add a few doxygen com...	✓ ✓	00:08:53	5 days ago
passed	#137 by	example -> eadb1b07 examples: small fix for R...	✓ ✓	00:03:25	1 week ago
passed	#136 by	simulation -> 5ad197f9 Change code structure.	✓ ✓	00:04:58	1 week ago
passed	#135 by	simulation -> d784cfc1 montecarlo: add paramet...	✓ ✓	00:09:33	1 week ago
passed	#134 by	master -> 7b4d30cc common: fix generation ...	✓ ✓ ✓	00:09:29	1 week ago
passed	#133 by	example -> d8bfda62 examples: add the analyz...	✓ ✓	00:07:54	2 weeks ago

Continuous integration: automatically build code and test it on each code commit

Infrastructure

The screenshot shows the ReCAS Wiki page for 'Algorithms'. The page is part of the 'HerdSoftware' project. The main content area is titled 'Algorithms' and was last edited by Nicola Mori 3 weeks ago. It contains a paragraph explaining that HerdSoftware algorithms process data objects and are defined by a sequence of algorithms. Below this, there are sections for 'Library: libHerdAlgorithms', 'Clustering algorithms', and 'Digitization algorithms'. The 'Clustering algorithms' section includes a link to 'StkClusteringAlgo', and the 'Digitization algorithms' section includes a link to 'StkGeometricDigitizerAlgo'. The right sidebar contains a 'Clone repository' button and a list of navigation links including 'Home', 'Examples', 'User's-manual', and 'Algorithms'. The left sidebar shows the project navigation menu with 'Wiki' selected.

ReCAS.com Projects Groups Activity Milestones Snippets

Search or jump to...

HerdSoftware

Project
Repository
Issues 8
Merge Requests 0
CI / CD
Operations
Registry
Wiki
Snippets
Settings

herd > HerdSoftware > Wiki > Algorithms

Algorithms

Last edited by Nicola Mori 3 weeks ago

New page Page history Edit

The HerdSoftware algorithms contain the code which processes the data objects; a sequence of algorithms defines an analysis. Algorithms can be combined and eventually appear multiple times in an analysis, depending on their implementation. Algorithms need data objects as input; since some data objects can be produced by algorithms, the algorithm sequence must account for these producer-consumer relationships. Detailed lists of the data objects consumed and produced by each algorithm can be obtained from the [doxygen documentation](#).

Library: libHerdAlgorithms

Clustering algorithms

These algorithms deal with the clustering of hits, i.e. creating cluster objects enclosing sets of neighbouring hits.

- [StkClusteringAlgo](#)
This algorithm creates clusters of Si strips hits from the STK. The clustering criterion is based on the signal-to-noise ratio: a seed strip and all the neighbouring hits above a given S/N value are grouped in a cluster.

Digitization algorithms

These algorithms convolve Monte Carlo data with the detector's response in order to reproduce the instrument response corresponding to the simulated data.

- [StkGeometricDigitizerAlgo](#)
This algorithm digitizes the STK detailed hits (i.e. hits of single particles on Si wafers) into hits on Si strips, applying a purely geometric criterion (i.e. it does not account for electronic noise, conversion to ADC, pedestals, charge drift in silicon etc.). Given a microstrip pitch value, it groups all the particle hits hitting the same strip and produces a hit for the strip.

Clone repository

Home

Examples

- Ex01: digitize MC
- Ex02: analyze MC

Overview

User's-manual

Algorithms

- Data model
- Data providers
- Develop new elements
- Download, configure and build
- Introduction
- Table of contents
- Usage

More Pages

<< Collapse sidebar

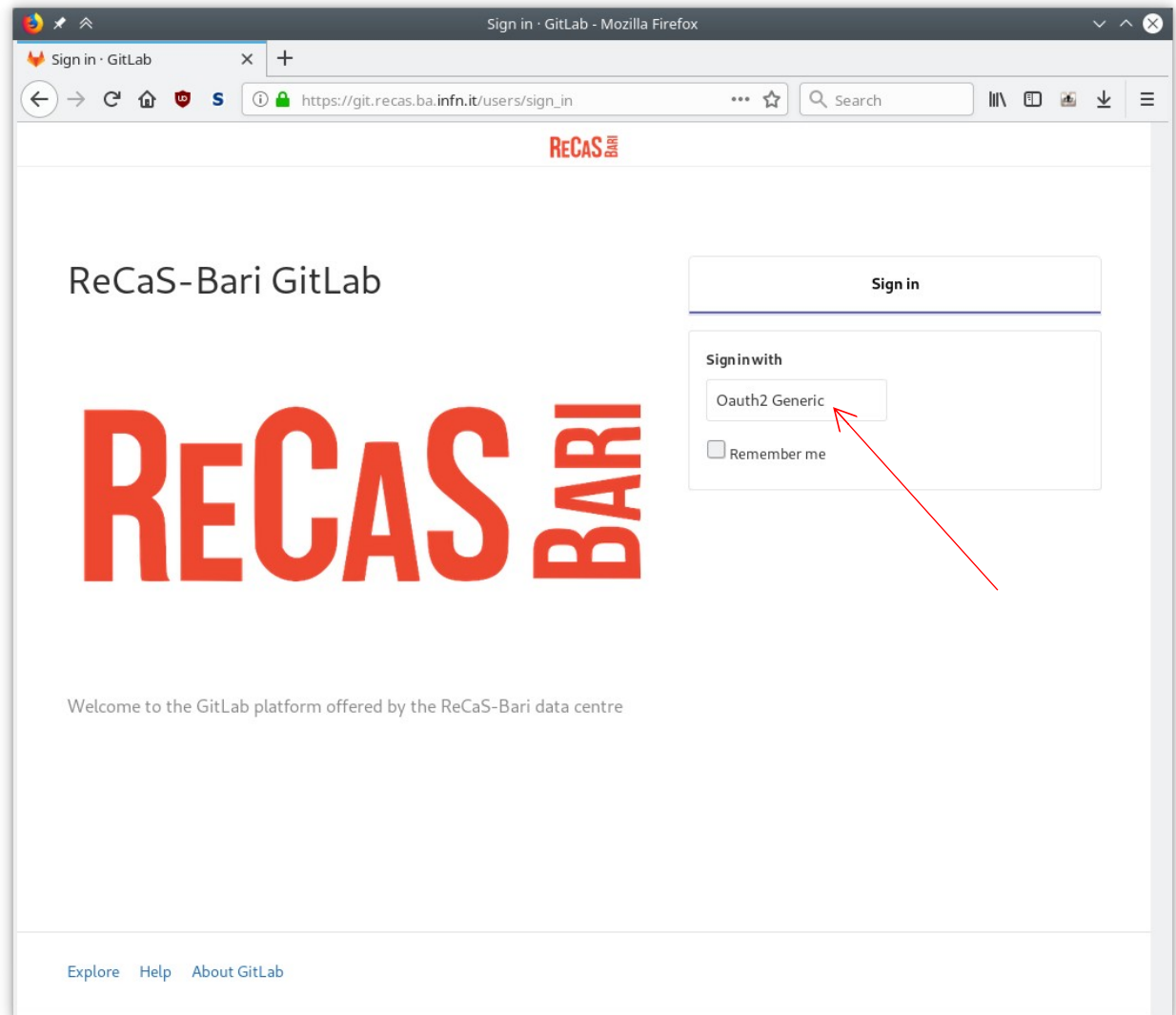


Infrastructure

- How to get access:
 - 1) Ask for registration on <https://git.recas.ba.infn.it/>

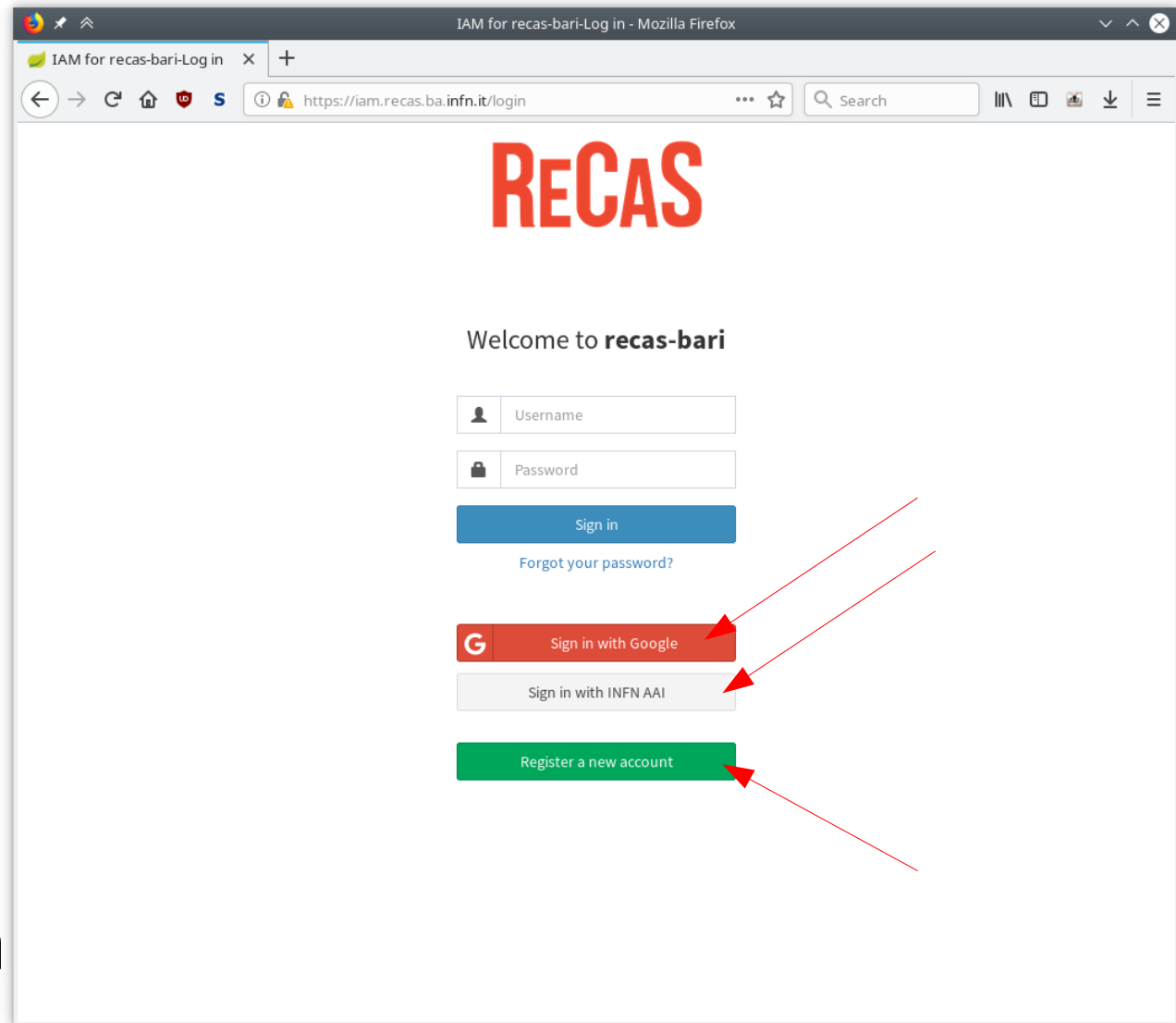
Infrastructure

- Welcome page



Infrastructure

- If you have an INFN or Google account you can use it to automatically retrieve the personal information needed for registration
- Otherwise click on the green button



Infrastructure

- Fill the registration form
- In the “Notes” field type:

Access to the Herd repository (contact person: Fabio Gargano, INFN Bari)

IAM for recas-bari-Register - Mozilla Firefox

IAM for recas-bari-Register x +

https://iam.recas.ba.infn.it/register

RECAS

Register at **recas-bari**

This is the recas-bari registration page.

If you want to register using an external identity provider, like Google or your institution identity provider, head back to the [login page](#) and click the "Sign in with" button of your preferred identity provider.

To proceed with the registration please fill in your personal information below.

Given name

Family name

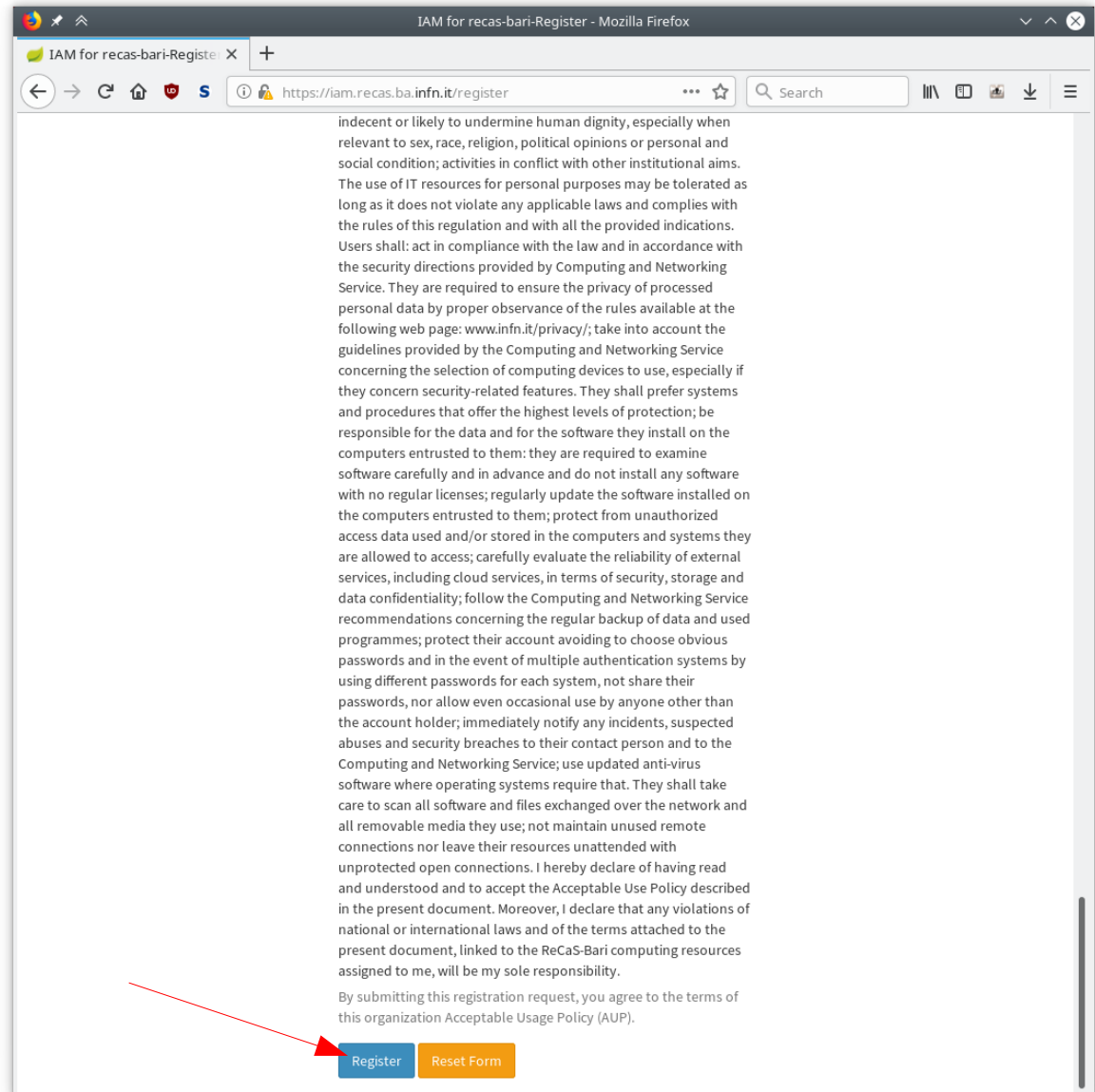
Email

Username

Notes

Infrastructure

- Click on the “Register” button at the end of the page



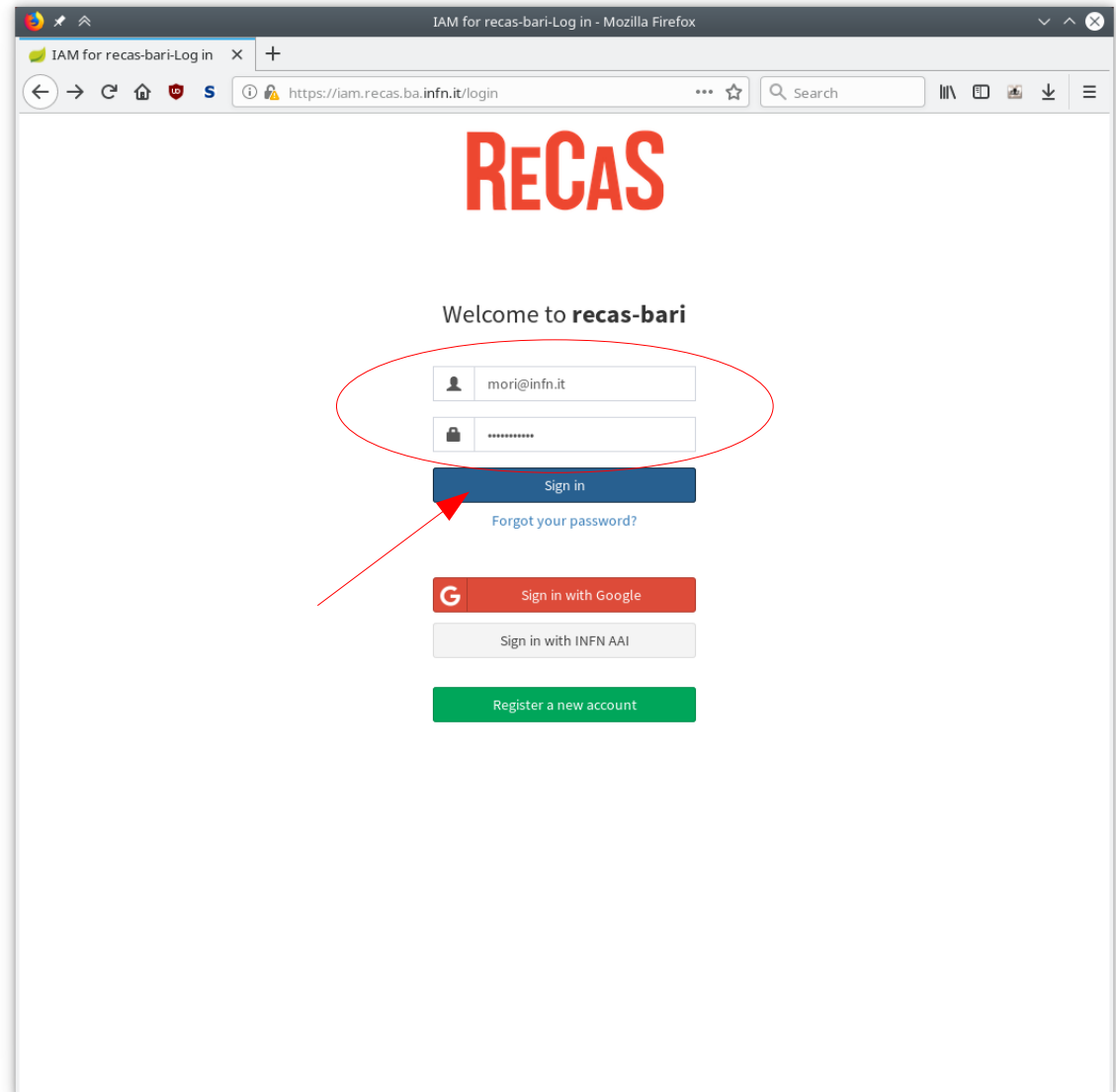


Infrastructure

- How to get access:
 - 1) Ask for registration on <https://git.recas.ba.infn.it/>
 - 2) Login for the first time

Infrastructure

- Wait for credentials (check the email)
- Login using the provided username and password





Infrastructure

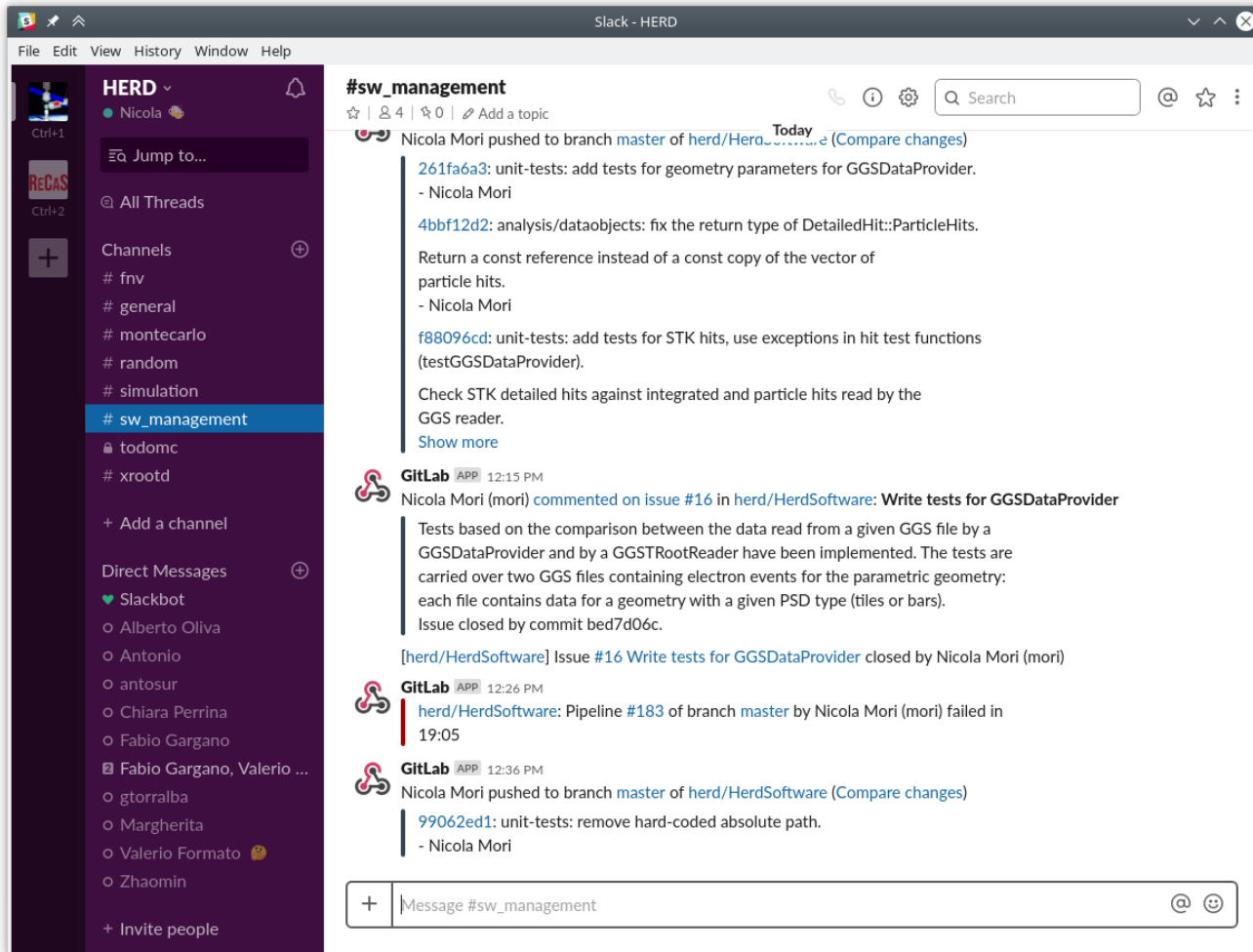
- How to get access:
 - 1) Ask for registration on <https://git.recas.ba.infn.it/>
 - 2) Login for the first time
 - 3) Ask for access to Herd repository to fabio.gargano@ba.infn.it
and wait for reply
 - 4) Login again



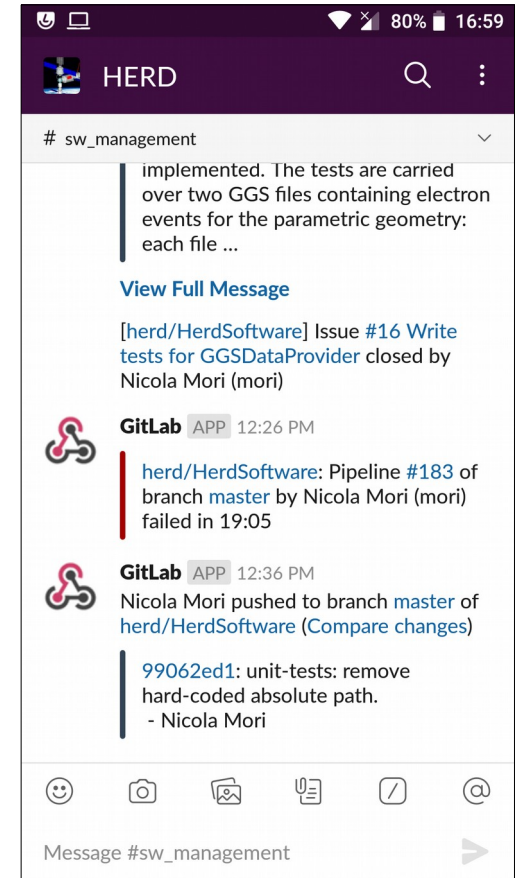
Infrastructure

- Slack channel
 - Instant messaging (like WhatsApp)
 - Groups for different topics (simulation, sw development etc.)
 - Apps (e.g. automatic notifications from GitLab)
 - Clients for desktop and smartphone

Infrastructure



Linux desktop client



Android mobile app



Infrastructure

- How to get access:
 - 1) Click [here](#) and follow the instructions
 - 2) Alternatively, ask for access to:
fabio.gargano@ba.infn.it
or to:
mori@fi.infn.it
then wait for the invitation email, click on the invitation link and follow the instructions



Resources

- HerdSoftware:
 - GitLab page
<https://git.recas.ba.infn.it/herd/HerdSoftware>
- Foundation frameworks:
 - GGS
<https://baltig.infn.it/mori/GGSSoftware> (code)
<https://wizard.fi.infn.it/ggs/> (documentation)
 - EventAnalysis
<https://baltig.infn.it/mori/EventAnalysis> (code)
<https://wizard.fi.infn.it/eventanalysis/> (documentation)