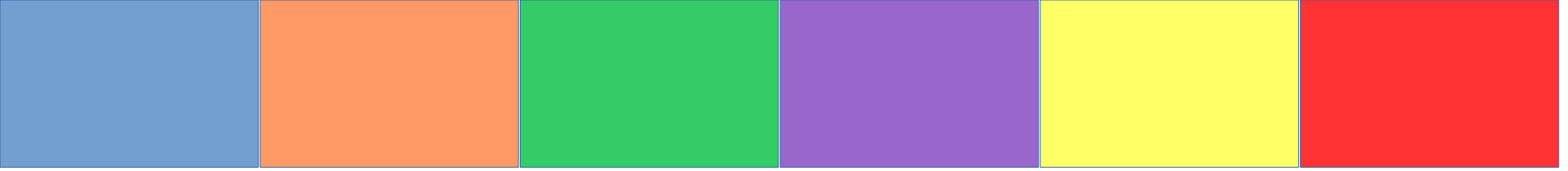


HerdSoftware: an overview

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

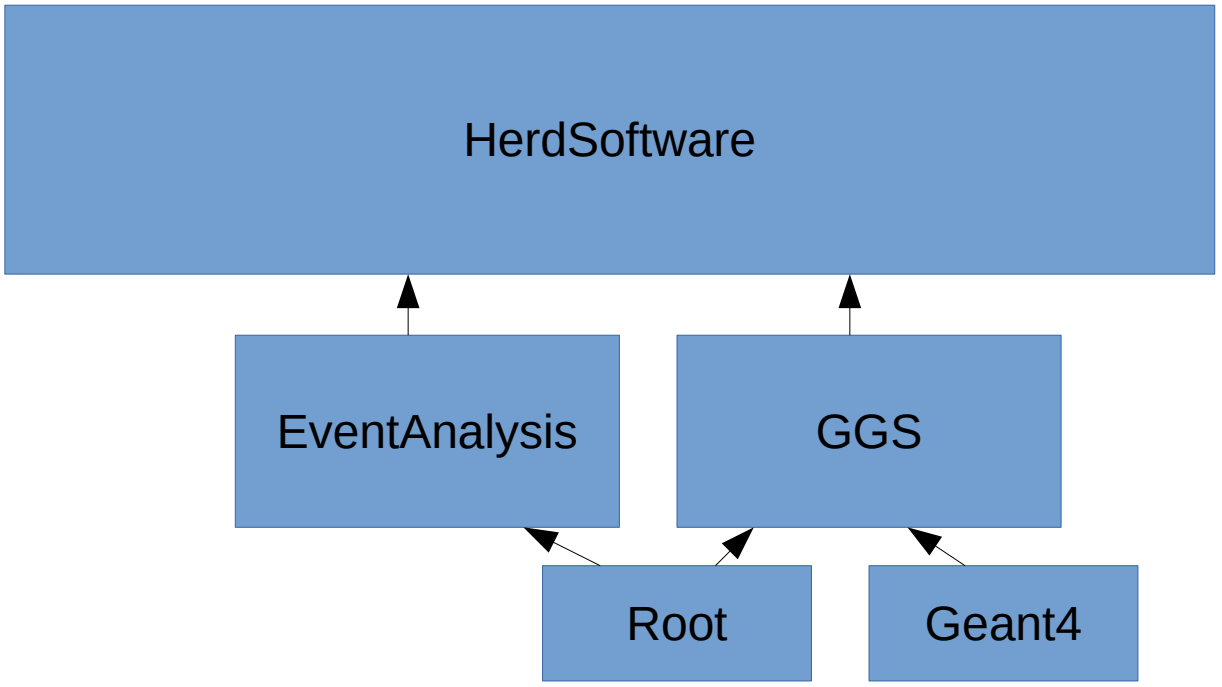
Requirements

- 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



Foundation frameworks

Foundation packages

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

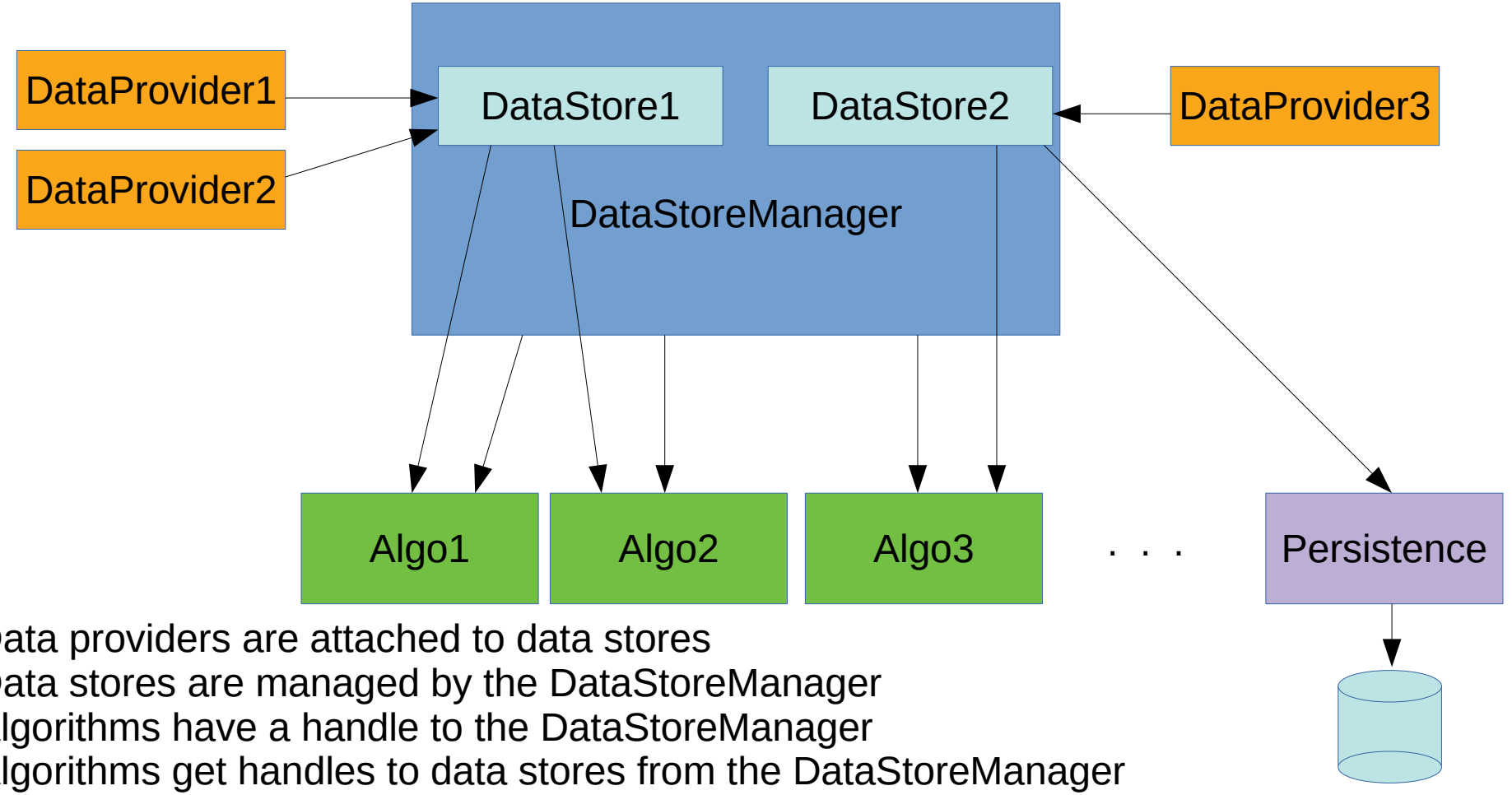
Architecture

- 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

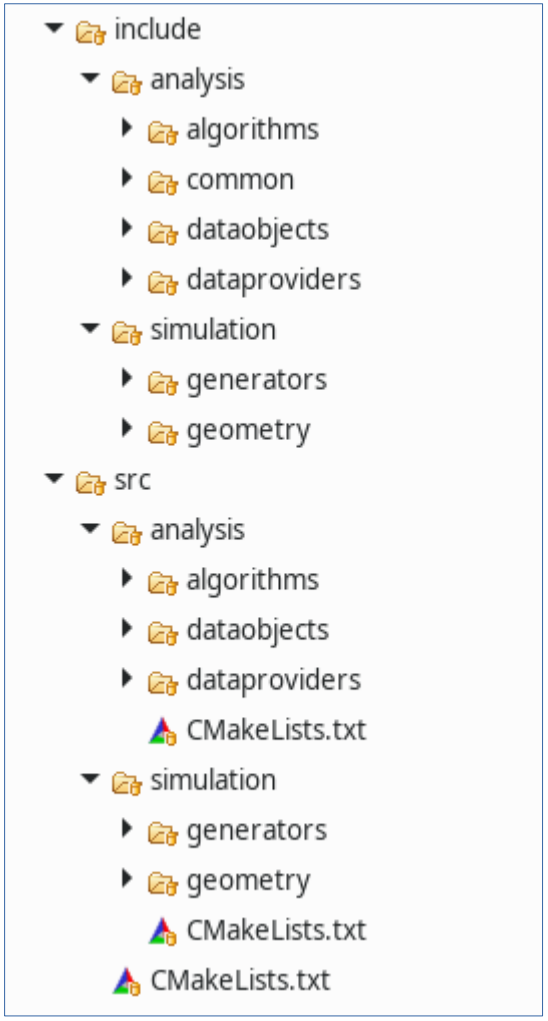


- Data providers are attached to data stores
- Data stores are managed by the DataStoreManager
- Algorithms have a handle to the DataStoreManager
- Algorithms get handles to data stores from the DataStoreManager
- Algorithms fetch (push) data objects from (to) data stores
- Data stores provide available 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:

Output file

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

GGS executable

Library with Herd geometry

Simulation job configuration 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/addGGSIntHitSD Crystal

/GGS/scoring/addGGSIntHitSD siLayer
/GGS/scoring/siLayer.GGSIntHitSD/storeParticleHits

/GGS/scoring/addGGSIntHitSD topSiLayer
/GGS/scoring/topSiLayer.GGSIntHitSD/storeParticleHits

/GGS/scoring/addGGSIntHitSD top_psd
/GGS/scoring/top_psd.GGSIntHitSD/storeParticleHits

/GGS/scoring/addGGSIntHitSD side_psd
/GGS/scoring/top_psd.GGSIntHitSD/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 GGSDataProvider 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

EventAnalysis config file

Plugin libraries to be loaded

Setup of input/output

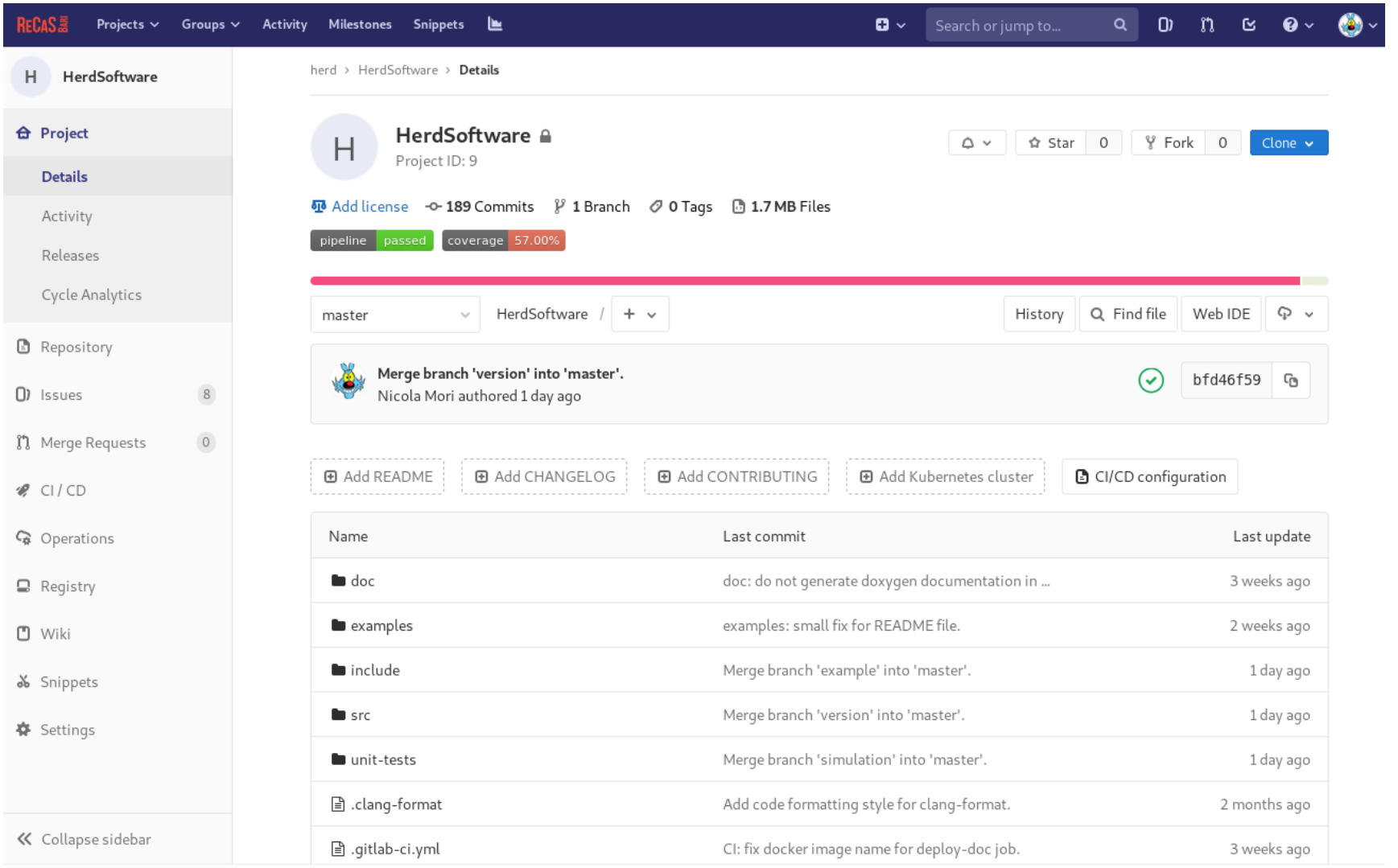
Analysis definition

```
Plugin HerdDataProviders
Plugin HerdAlgorithms
Plugin RootPersistence
Plugin HerdDataObjectsDict

DataProvider GGSDataProvider 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

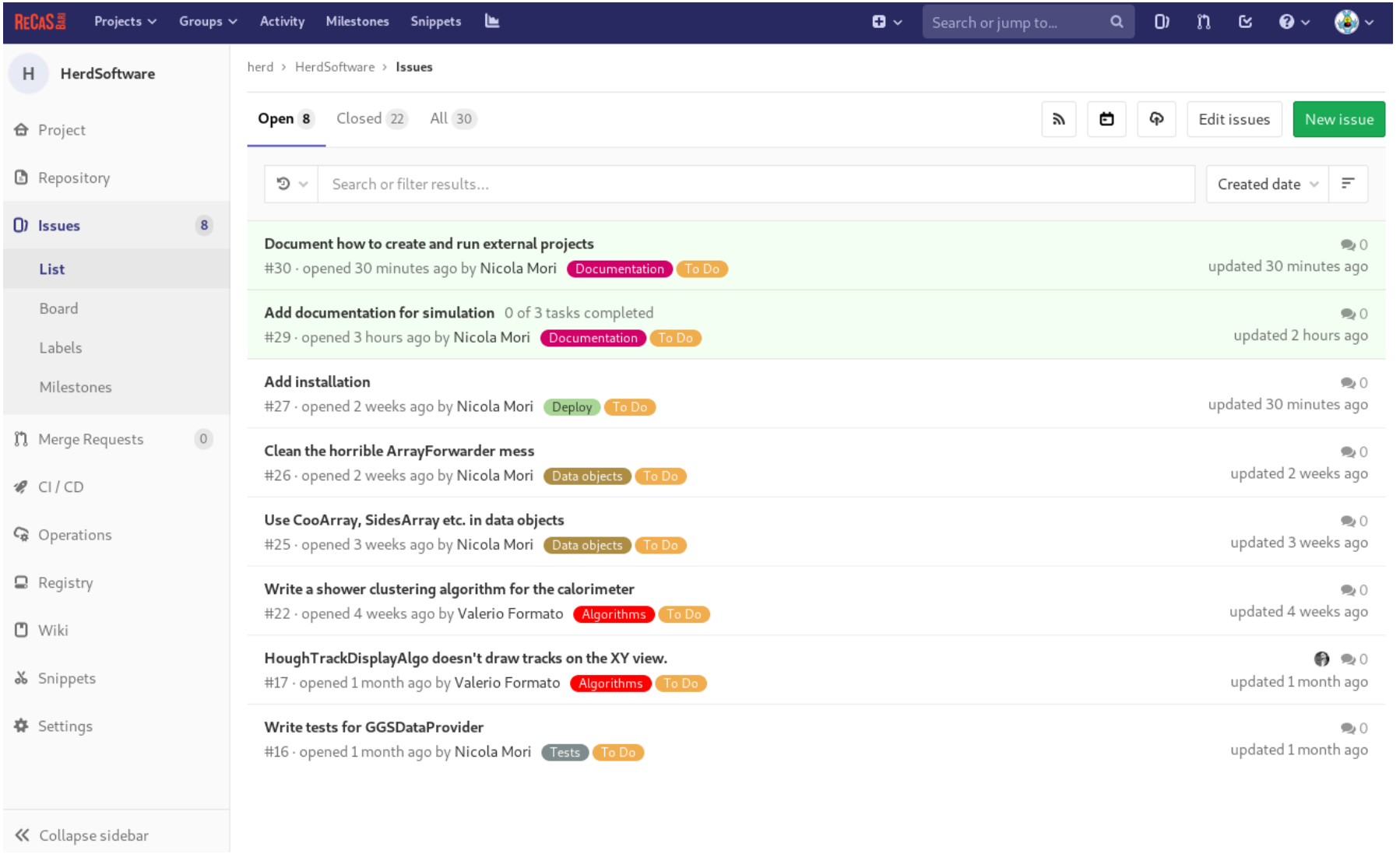
- GitLab instance hosted by ReCaS computing center in Bari (Italy)
  - Git repository
  - Issue tracker
  - Continuous integration
  - Wiki with documentation
- Restricted access
  - Roles (guest, developer, maintainer etc.)

Infrastructure



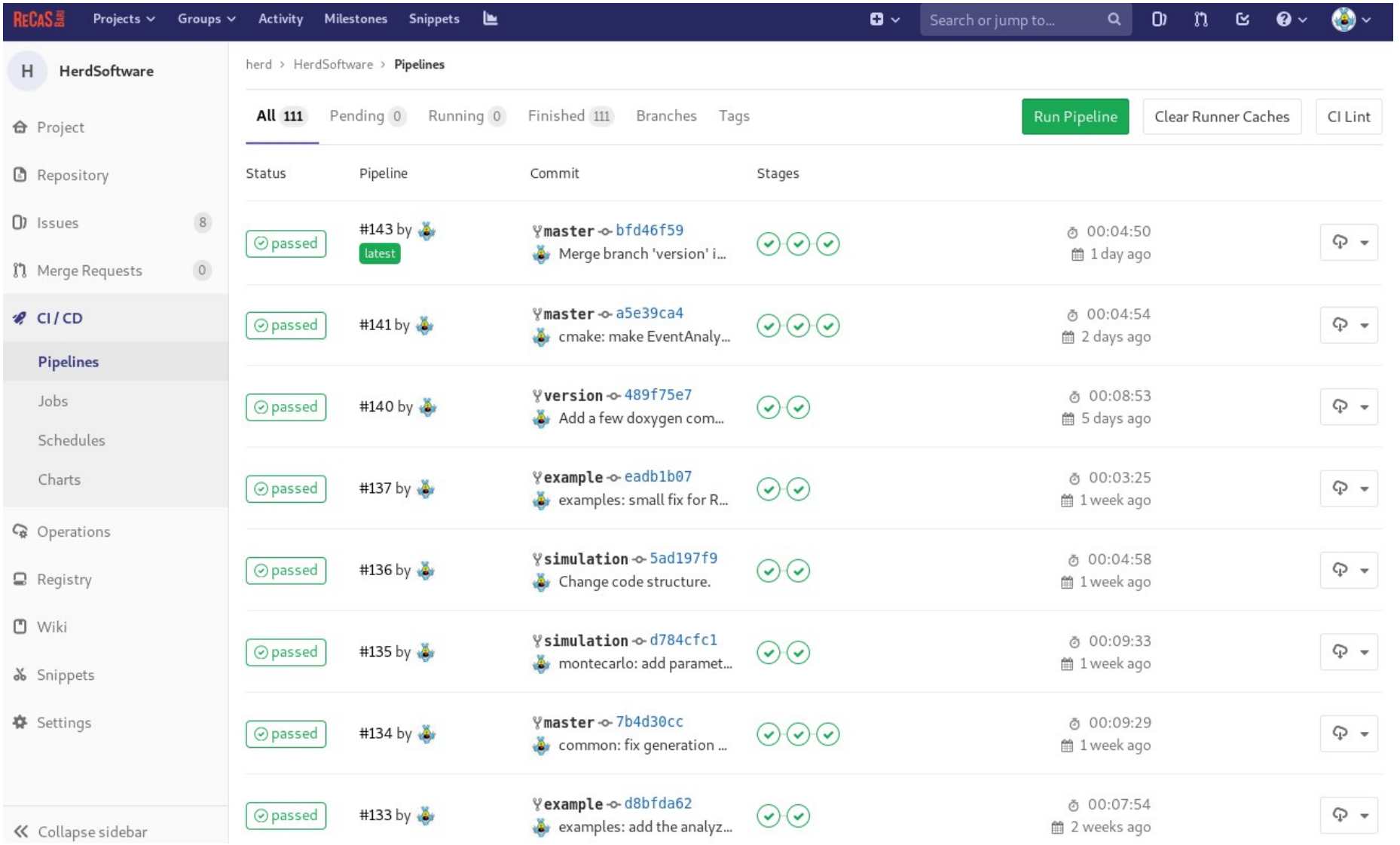
Main page

Infrastructure



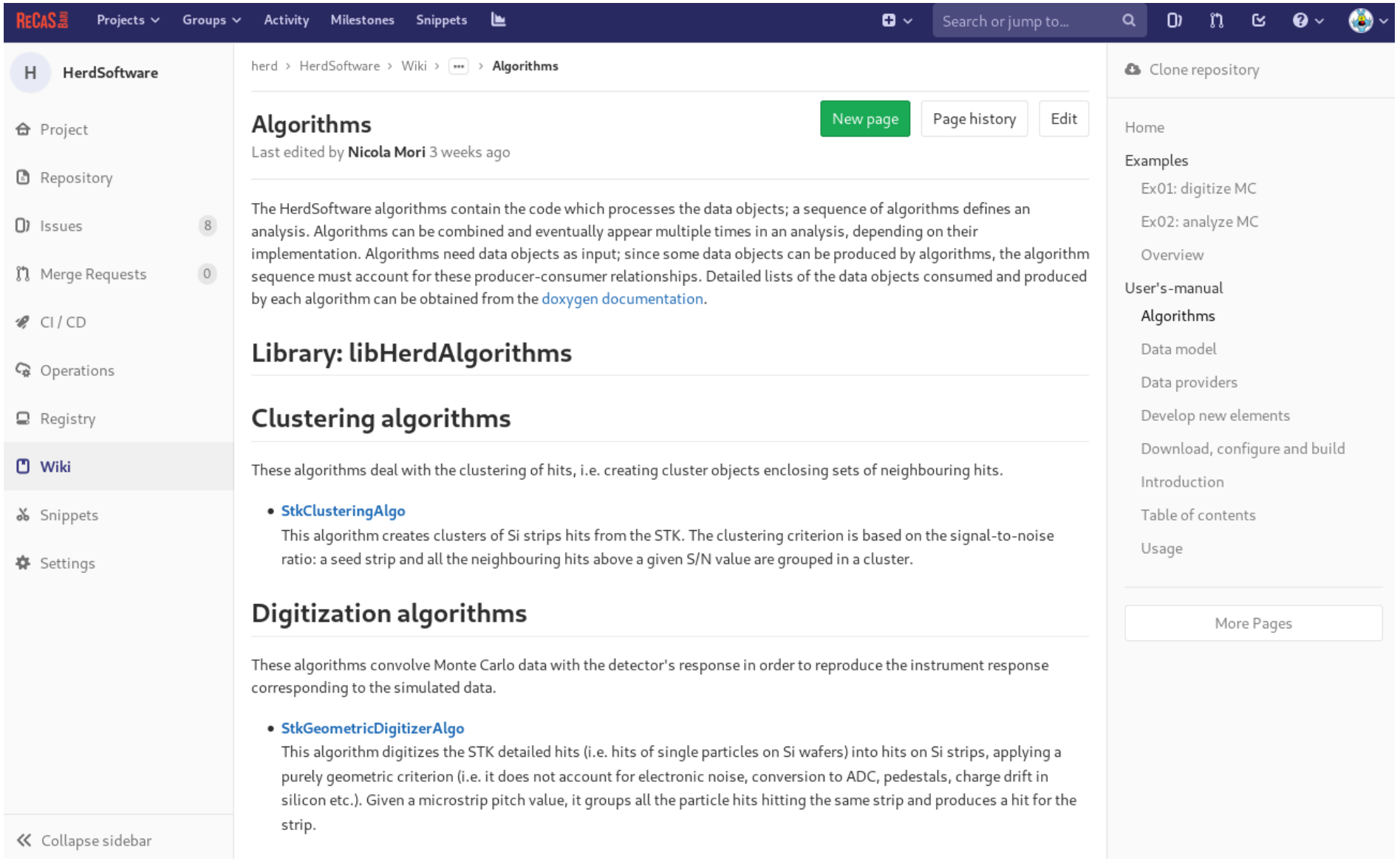
Issues: a list of features to implement and problems to fix

Infrastructure



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

Infrastructure



Wiki: documentation and guides

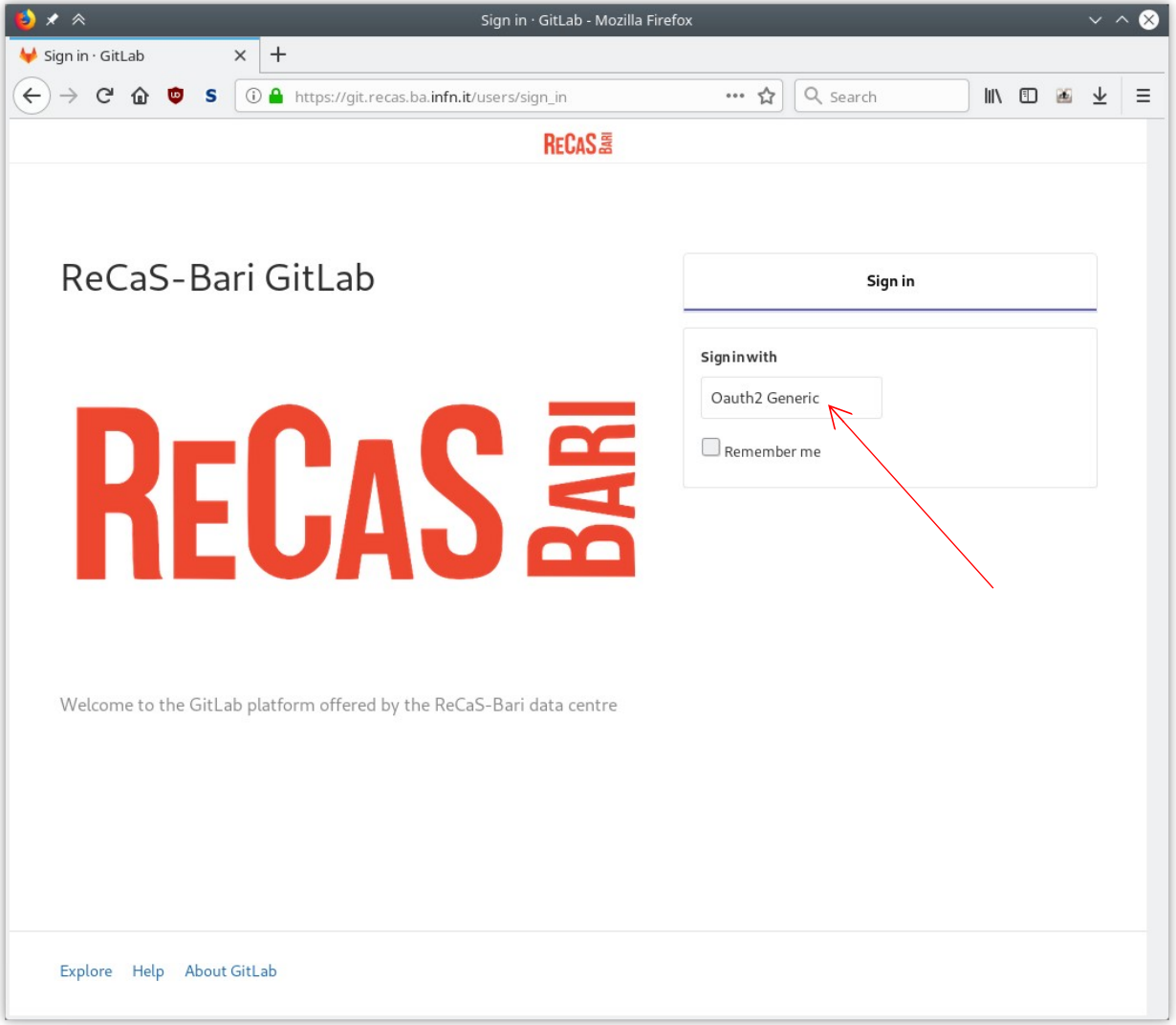
Infrastructure

- How to get access:

  1) Ask for registration on https://git.recas.ba.infn.it/

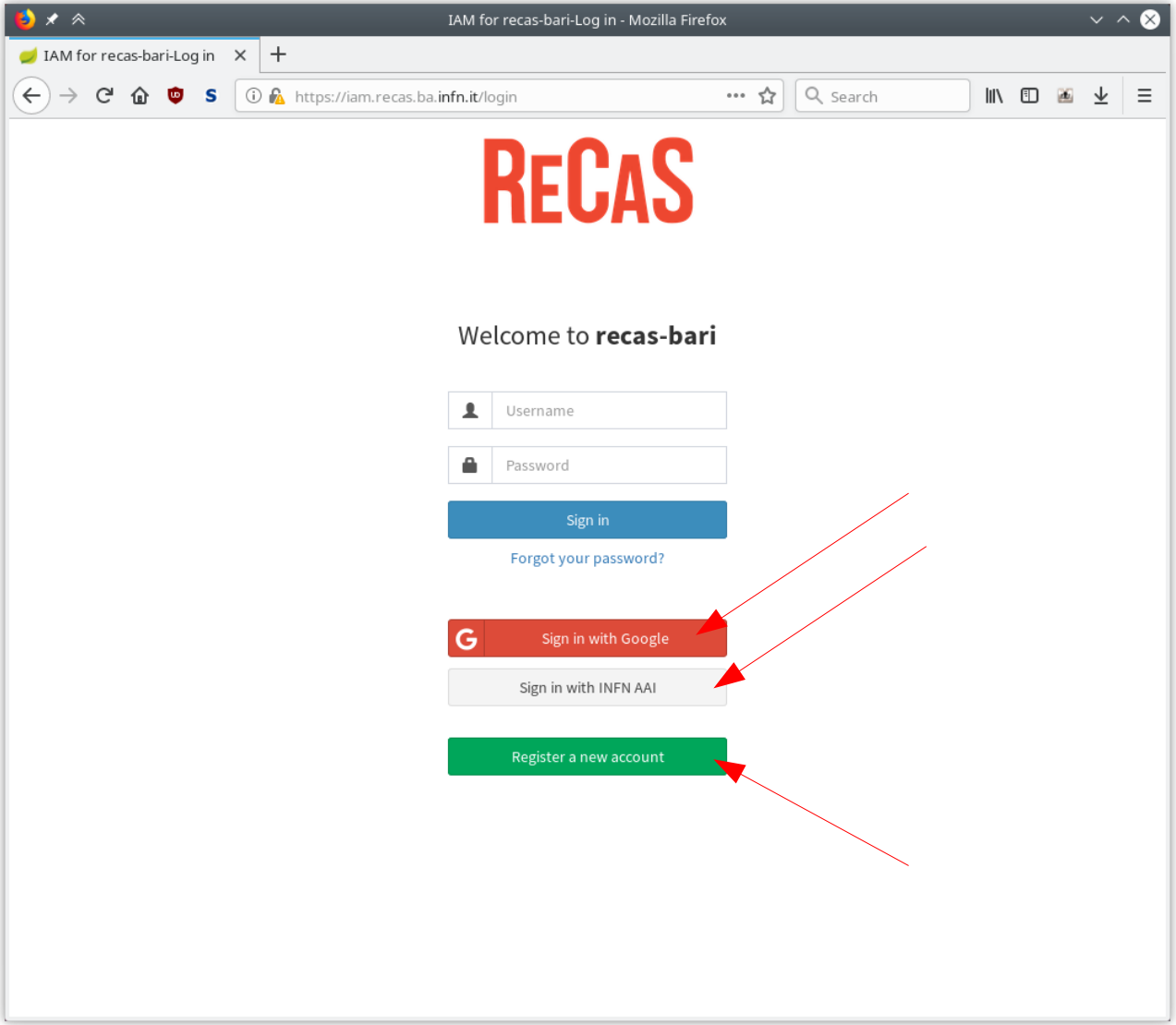
Infrastructure

- Welcome page



Infrastructure

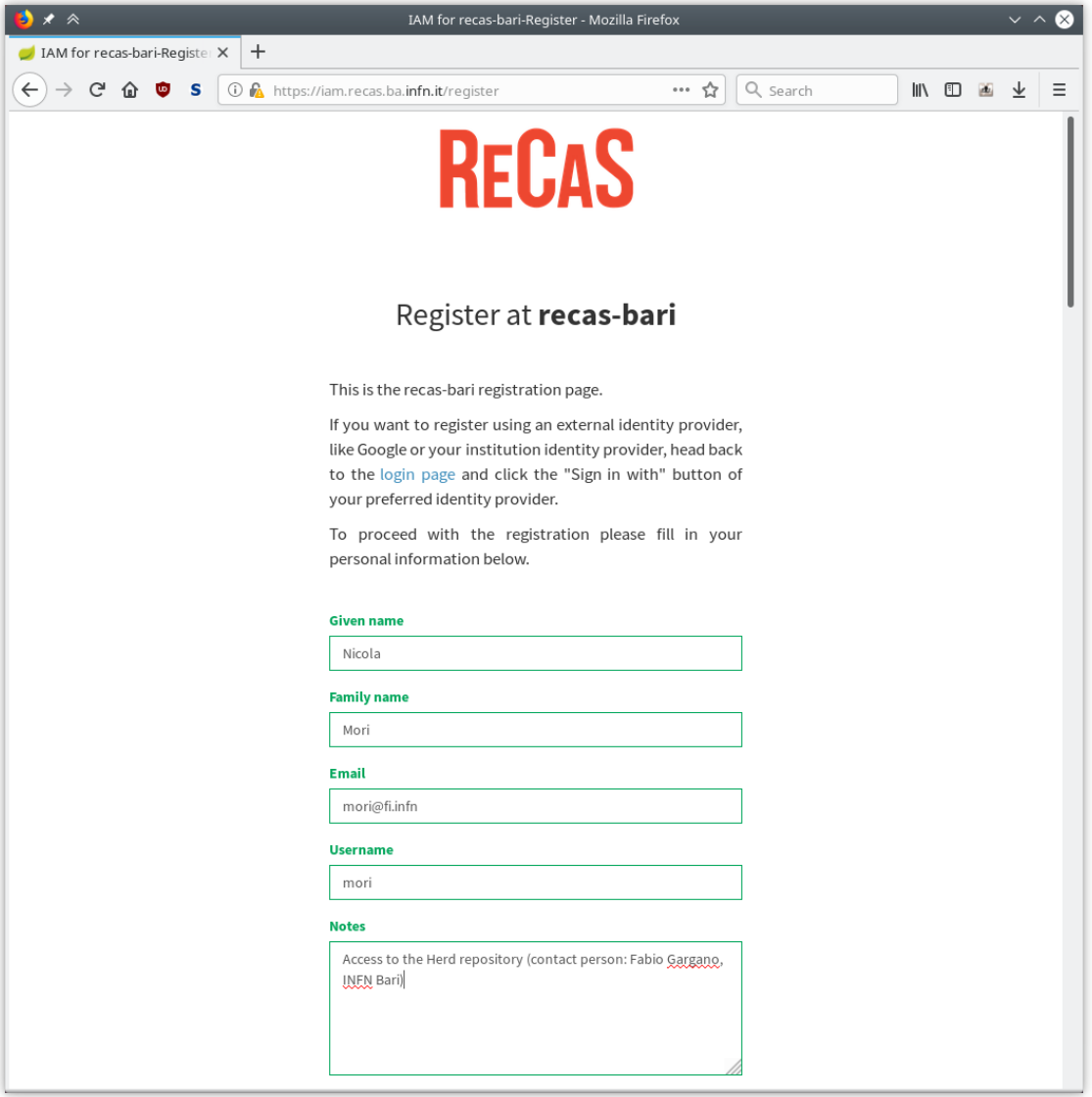
- If you have an INFN or Google account you can use it to automatically retrieve the personal information needed for registration
- Otherwise click on the green button



Infrastructure

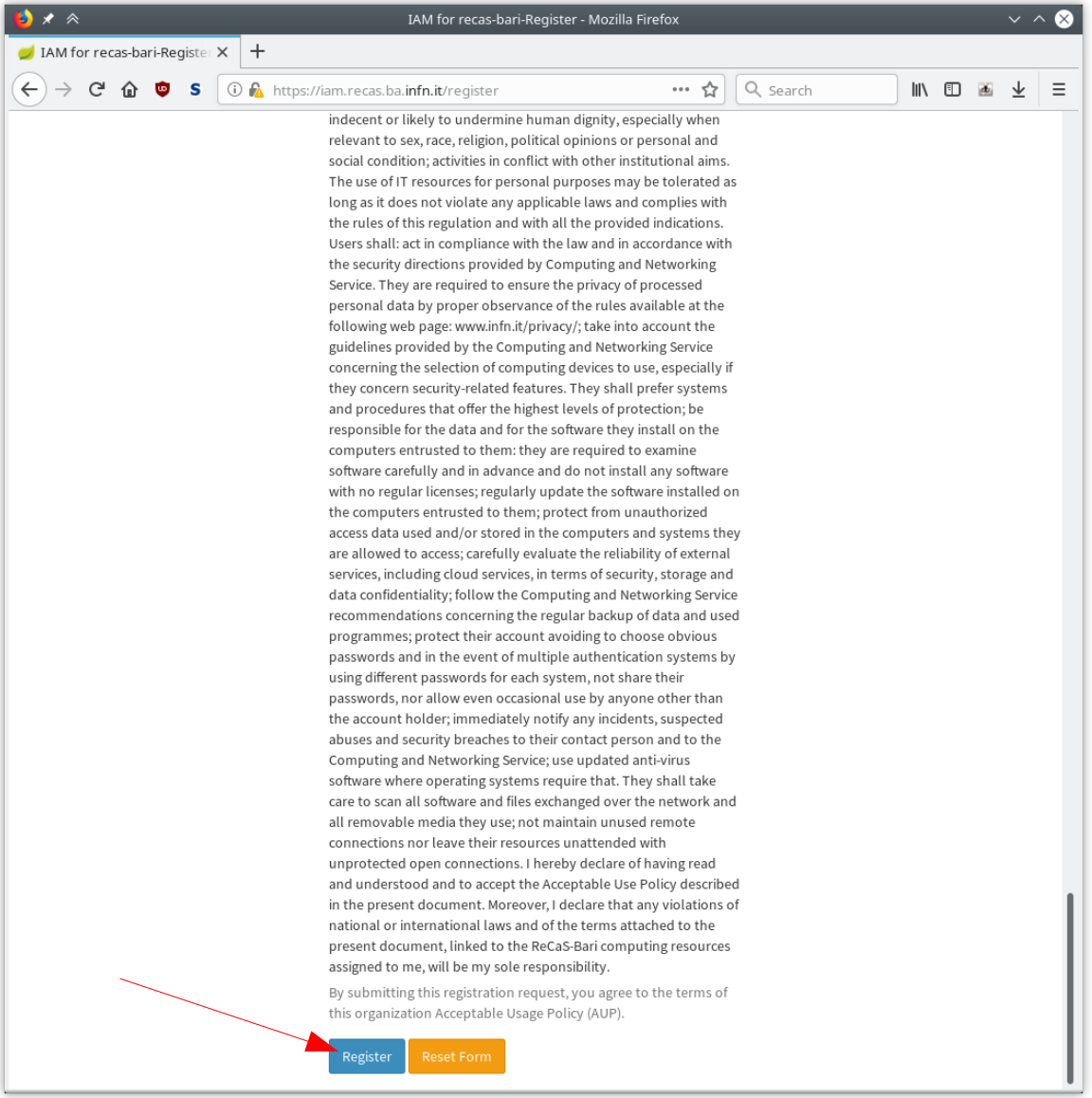
- Fill the registration form
- In the “Notes” field type:

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



Infrastructure

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



Infrastructure

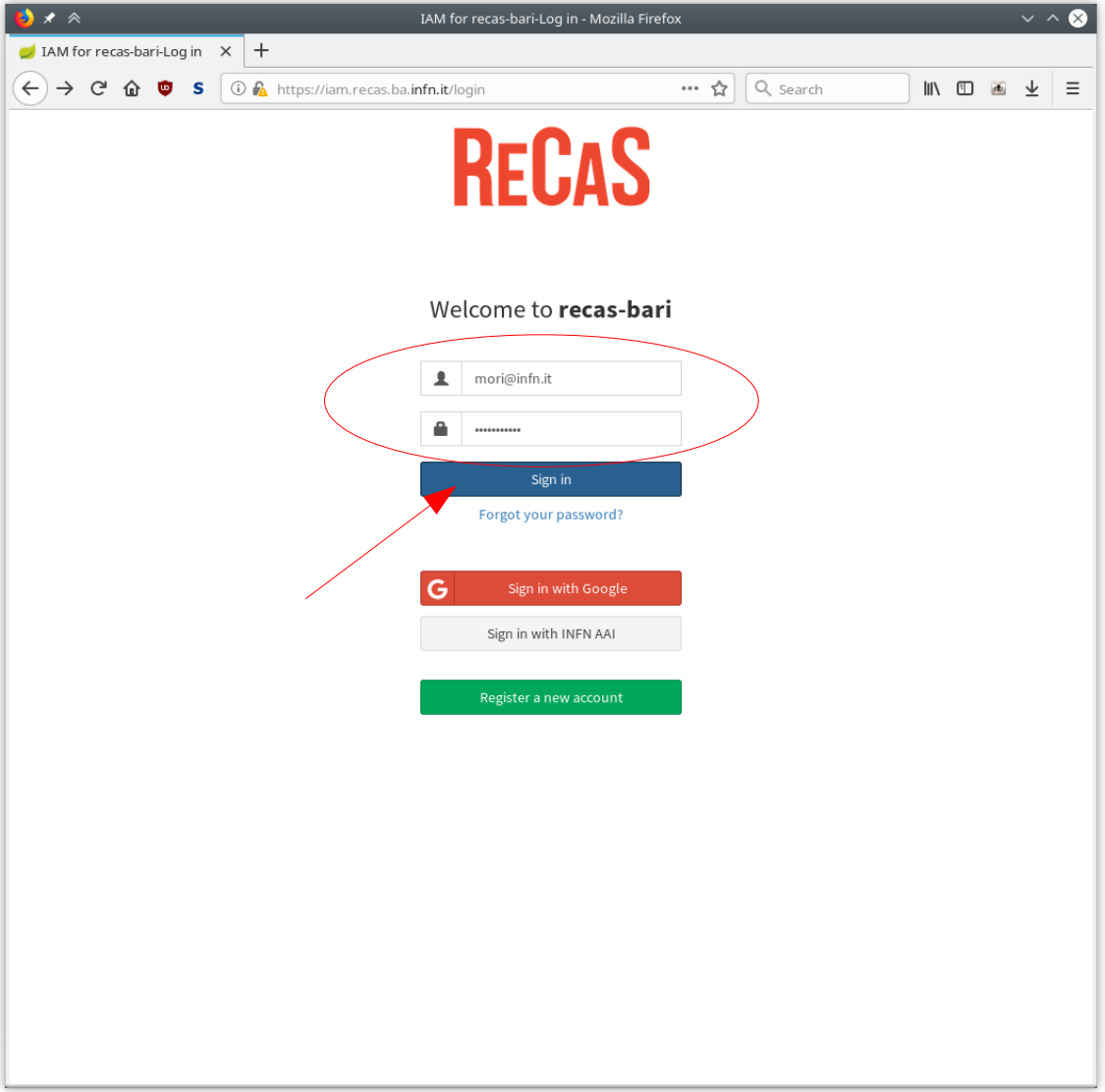
- How to get access:

  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:

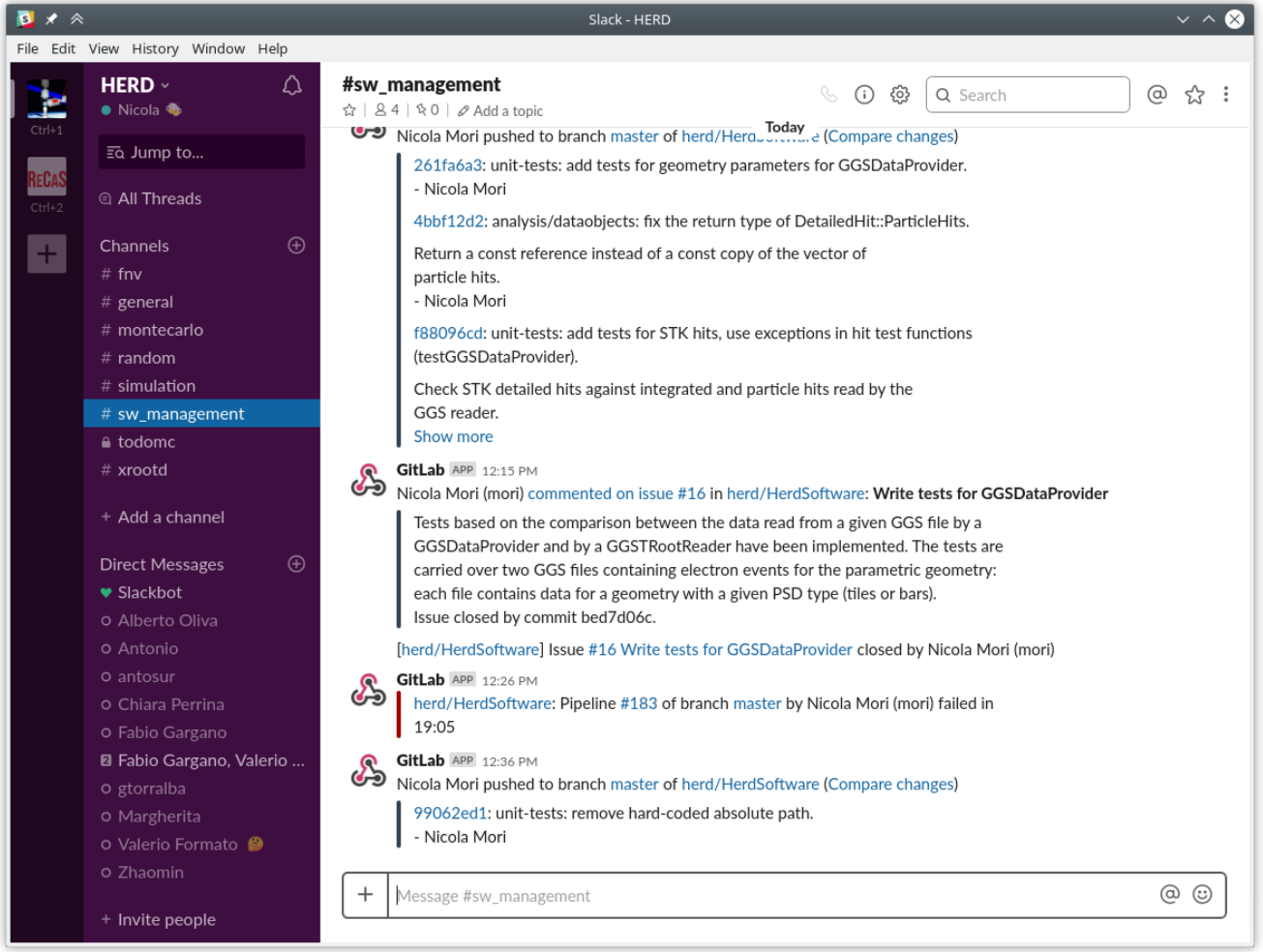
  1) Ask for registration on https://git.recas.ba.infn.it/
  2) Login for the first time
  3) Ask for access to Herd repository to fabio.gargano@ba.infn.it

     and wait for reply
  4) Login again

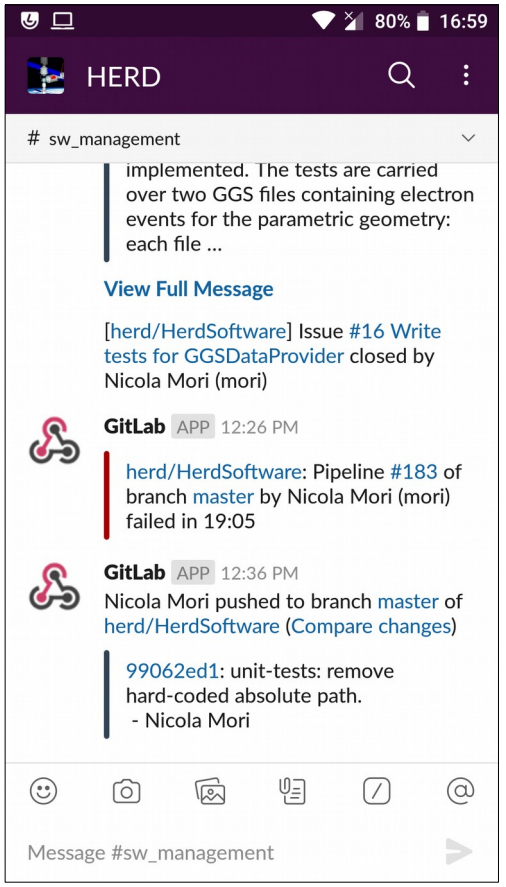
Infrastructure

- 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)