gAn

gAn tutorial

Germano & Ruggero

AEgIS Analysis Framework

CERN - 21/09/2016

Premise

- the AEgIS raw data are written by the DAQ in a ROOT file

- for “analysis” we mean accessing the raw data and producing numbers and distributions

- at the moment each users doing analyses is using his own program (mainly root macros)

gAn objective

- to have a common Framework for all the AEgIS analyses

gAn is working but it is still in beta version

gAn basics

- the gAn framework it is based on ROOT and has been developed in C++

- the gAn reads the raw data and produces numbers and distributions

- it is intended to be used by everybody (even with little or no knowledge of C++)

- it can be expanded and enriched by users with some knowledge of C++ (*developers*) and the additions can be easily included in the framework for all the other users

Tutorial summary

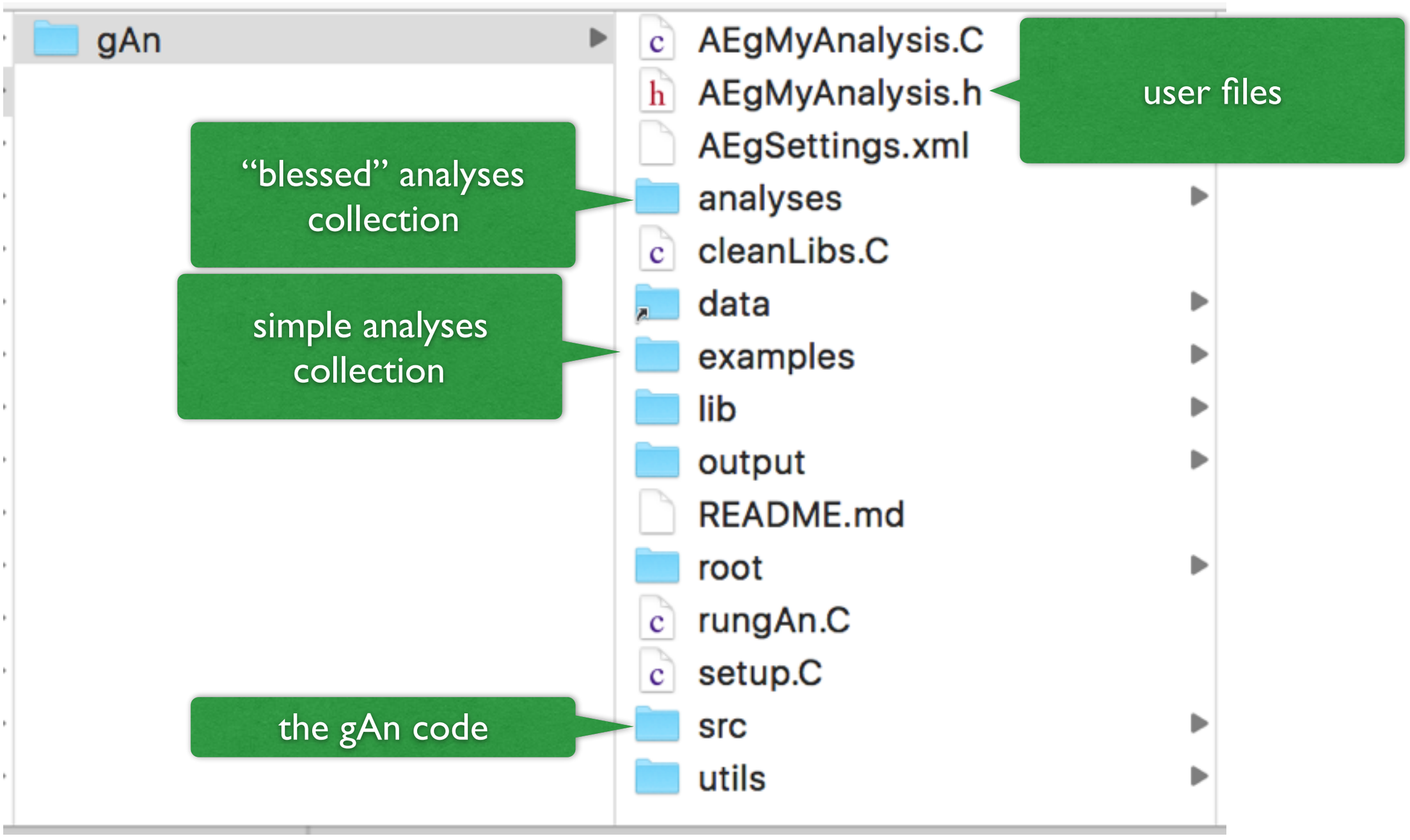
- framework general description

- code management
  -> GitLab repository (git)
  -> Access levels

- general use
  -> documentation
  -> how to get the gAn and how to get the data
  -> how to run it (live tutorial)
  -> the AEgSettings.xml (live tutorial)
  -> how to run an available analysis (live tutorial)
  -> how to write your own analysis (live tutorial)

- Q&A

the “directory” structure



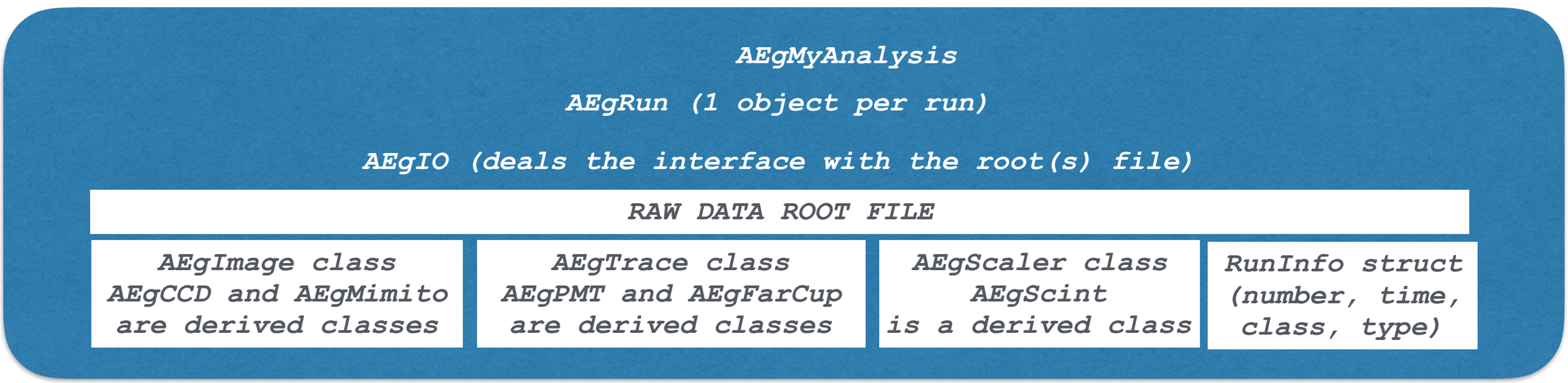
framework general description

the User Code is handled by the AEgMyAnalysis class (inherited from AEgAnalysisBase)

AEgMyAnalysis.C *and* ***AEgMyAnalysis.h*** *files are not part of the repository - they are copied from* ***utils/AEgTemplate.C and AEgTemplate.h*** *at the setup stage*
In this way every user can develop his own analysis, without changing the code in the repository.
Users can then exchange ***AEgTemplate.C and AEgTemplate.h*** *so that every other user can test his analysis*

the user run gAn via the rungAn macro [`.x rungAn.C(run)` or `.x rungAn("user_file_list.txt")`]

actually ***rungAn.C*** *loads the class libraries and calls* ***execgAn.C*** *that in its turn load the AEgMyAnalysis class and launch the analysis [this workaround is needed by ROOT6]*



framework general description

the "main" program

```
void execgAn(Int_t run_start, Int_t run_end) {
     // Initialize settings system
     AEgSettingsManager::Instance().LoadConfigFile("AEgSettings.xml");

     // Create user code
    AEgMyAnalysis *User = new AEgMyAnalysis();

    // Open output file
    User->CreateOutputFile(run_start, run_end);

     // Cycle on the list of runs
    for (int run=run_start; run<=run_end; ++run) {
     // Prepare settings for this run number
          AEgSettingsManager::Instance().SetRunNumber(run);

    // Open the run
        AEgRun *Run = new AEgRun(run);

        // Check if it is valid
        if(!Run->GetIsRun())
          continue;

    // Process the analysis
    bool Ret = User->AnalyzeRun(Run);

    // If run was marked valid, save the data
    if (Ret)
          User->PushAnalyzedData(Run);

    // Check if in batch mode
    if (gBatchMode) {
          // Delete run
          delete Run;
    }
}
```

it reads the user's settings

it creates the user's "analysis"

it creates the "AEgRun" object (raw data decoding etc.)

the "analysis" is performed

it save the data

go to the next "run"

the AEgSettings.xml file

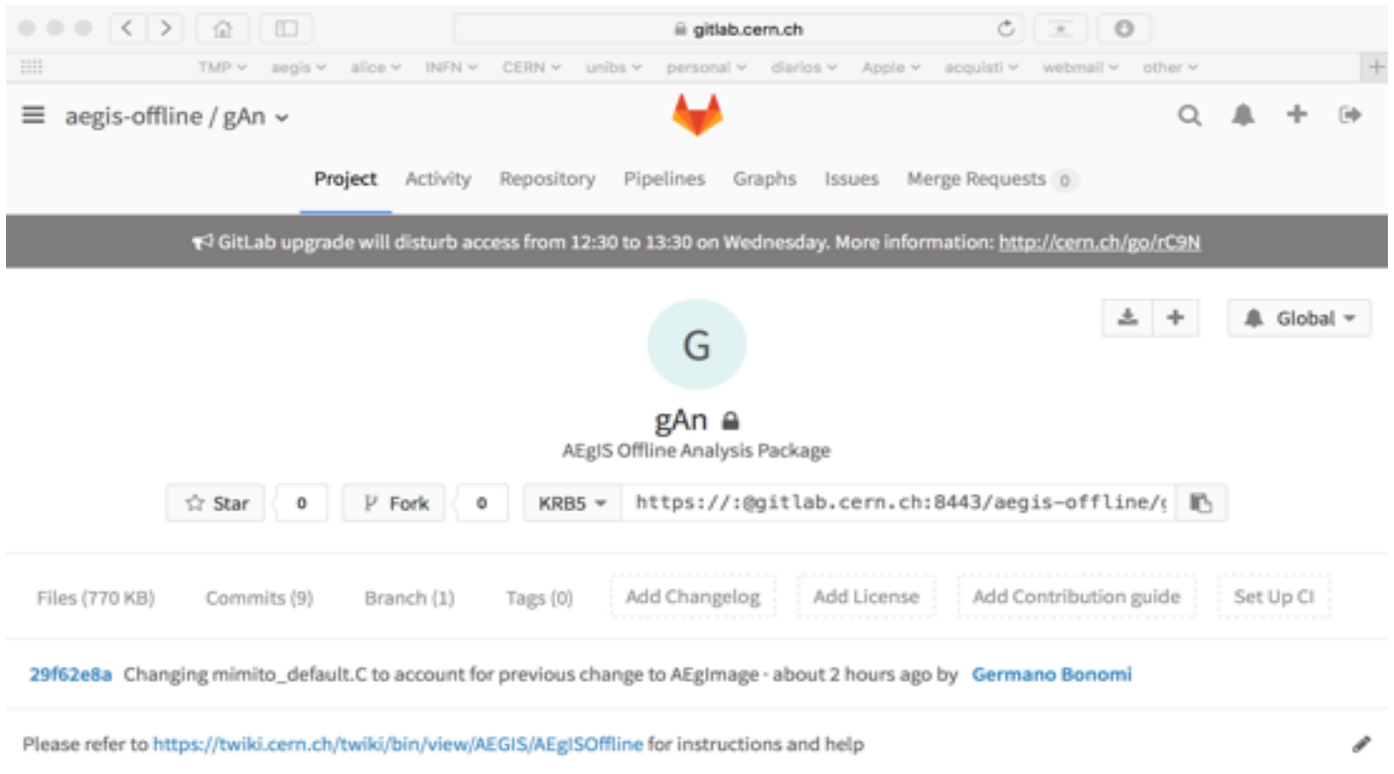
```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<!-- General settings of gAn -->
<general>
    <verbosity  v="1" />
    <batch_mode v="0" />
</general>


<!-- Paths where to access the files -->
<path>
    <enviro_data v="enviro/" />
    <run_data    v="data/" />
    <out_data    v="output/" />

<!--
    <run_data v="data/run_2012/" from="0" to="9999" />
    <run_data v="data/run_2013/" from="0" to="9999" />
    <run_data v="data/run_2014/" from="0" to="9999" />
    <run_data v="data/run_2015/" from="0" to="9999" />
    <run_data v="data/run_2016/" from="0" to="9999" />
-->
</path>


<!-- Output configuration -->
<output>
    <file       v="gAnOut" />
    <mode       v="recreate" />
    <tree_name  v="out" />
    <tree_desc  v="Analyzed gAn data" />
</output>


<!-- Analysis base configuration -->
<analysis_base>
    <enable_mimito  v="1" />
    <enable_scint   v="1" />
    <enable_mcp     v="1" />
    <enable_hdmcp   v="1" />
    <enable_pmt     v="0" />
    <enable_farcup  v="1" />
</analysis_base>
```

framework general description

gAn-web

There is a "side-project" called gAn-web that has the goal to develop a web interface to gAn. It would be a "server-side" program that would run "default" analyses and produce "standard" plots. For such outputs there would be no need to installation and configuration, just typing a run number and push a button

code management



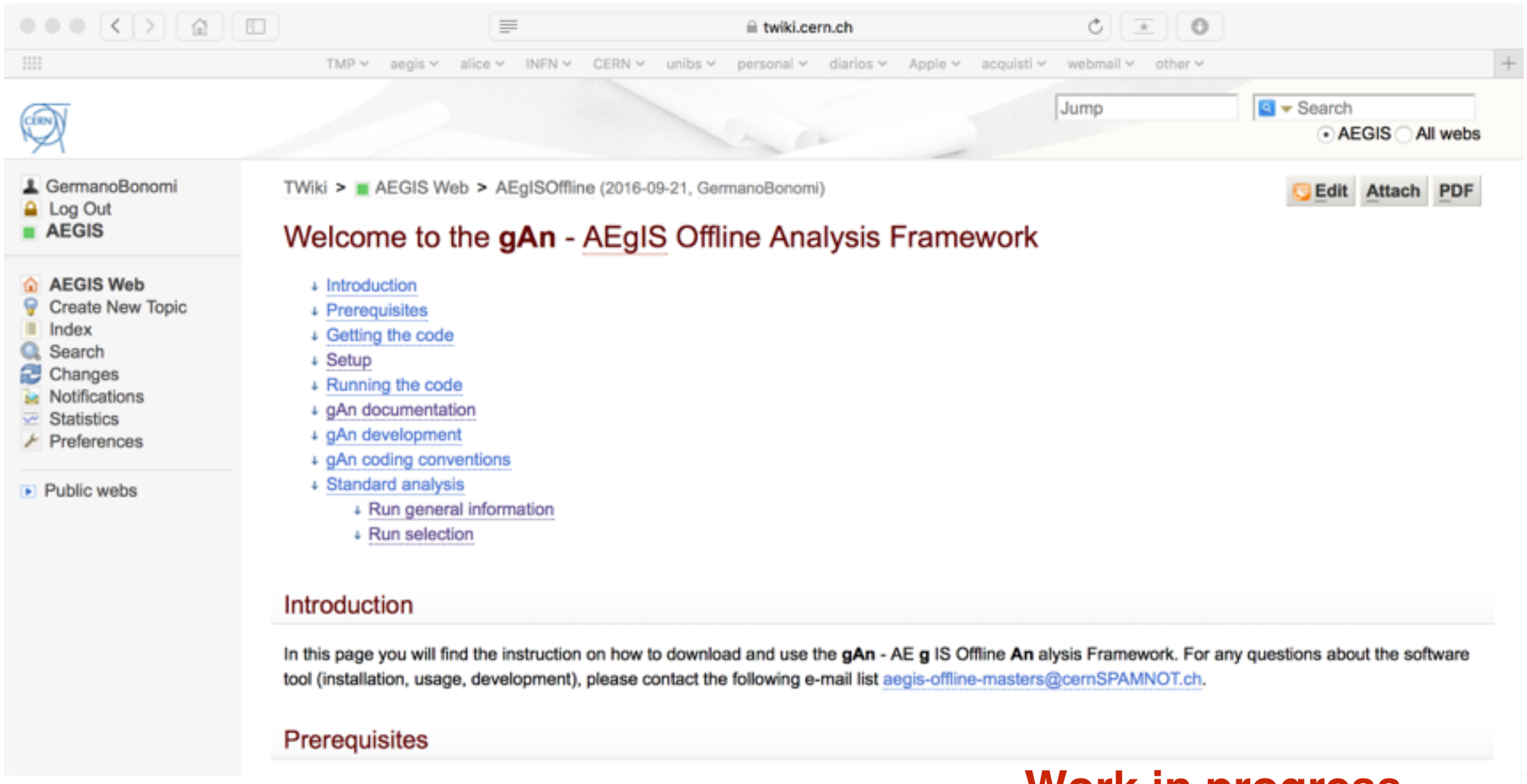
The **gAn** is hosted in the CERN GitLab [aegis-offline GitLab group]

- The project **is** accessible via 3 levels of permissions
  - **users** -> able to install and use the package (writing his own analysis)
  - **developers** -> users that can create new branches and develop code [but not able to commit and push to protected brances]
  - **masters** -> managers of the repository

some **git** knowledge is necessary only for developers but not for users

documentation

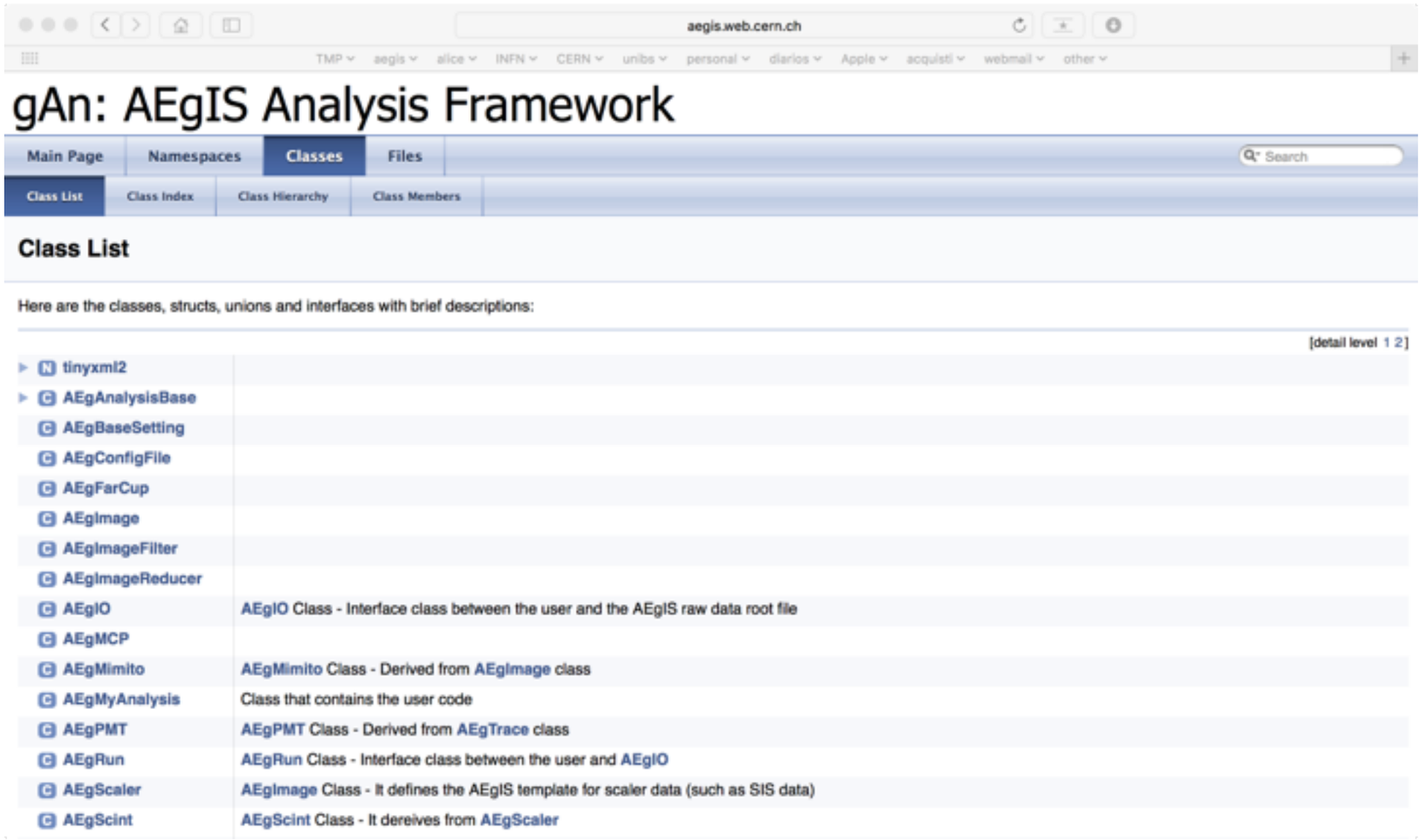
The **gAn** manual page is available from our TWiki page:
https://twiki.cern.ch/twiki/bin/view/AEGIS/AEgISOffline



Work in progress …

documentation

The **gAn** class description is available here:
http://aegis.web.cern.ch/aegis/aegis-offline/gAn/html/index.html



Work in progress …

first steps

```
git clone https://gitlab.cern.ch/aegis-offline/gAn.git
cd gAn
root setup.C
```

how to get the raw data [*the data need to be copied in the "data" sub-folder*]

The **raw data** are available from the web:
https://aegisgateway.cern.ch:8443/daq/root_files/?C=M;O=D

The **enviro data** are available from the web:
https://aegisgateway.cern.ch:8443/daq/root_files/environmental_monitor/

In alternative they can be copied via *scp* or *sftp* from one of the linux machines in the control room (i.e. *aegislx01*, *aegislx04*, *aegiscl01*) from the /data directory

… now you are ready to run the gAn …