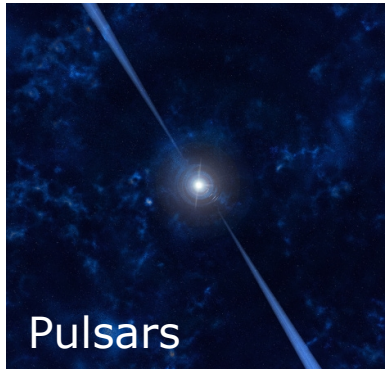
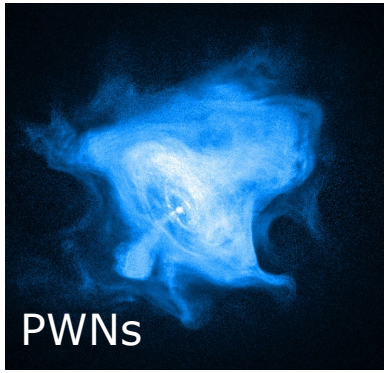
GammaLib

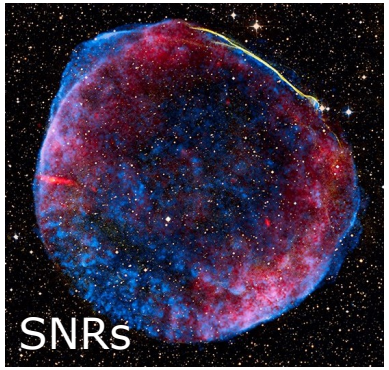
A versatile framework for the analysis of astronomical gamma-ray data

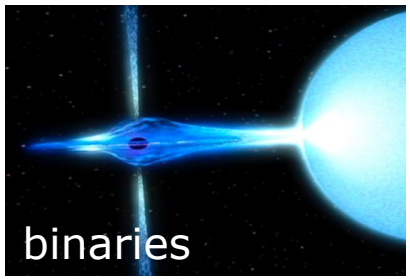
Jürgen Knödlseder, on behalf of the GammaLib development team

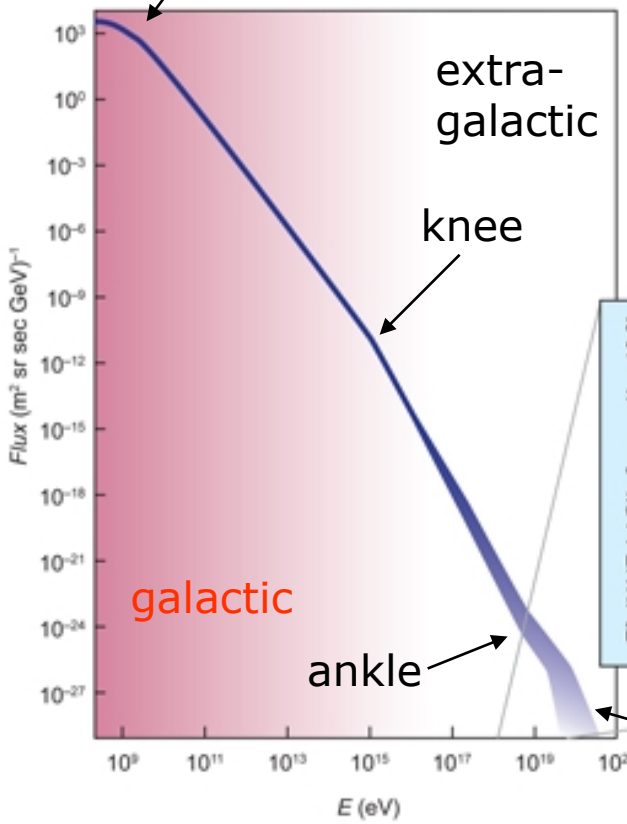
Understanding the non-thermal Universe

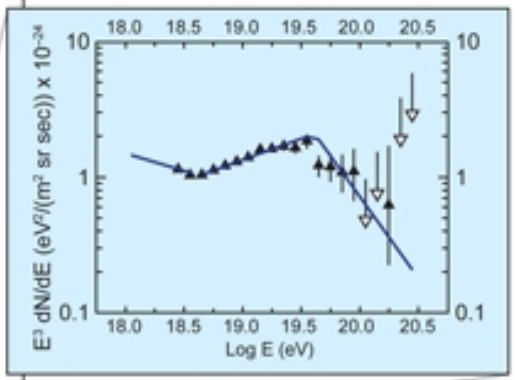


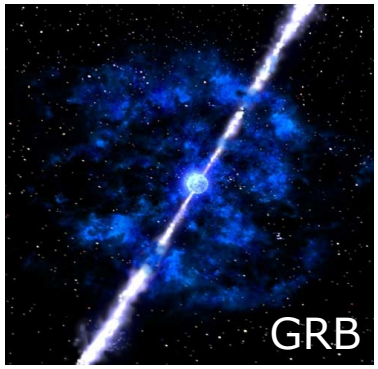


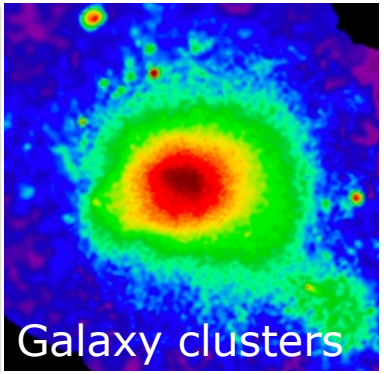




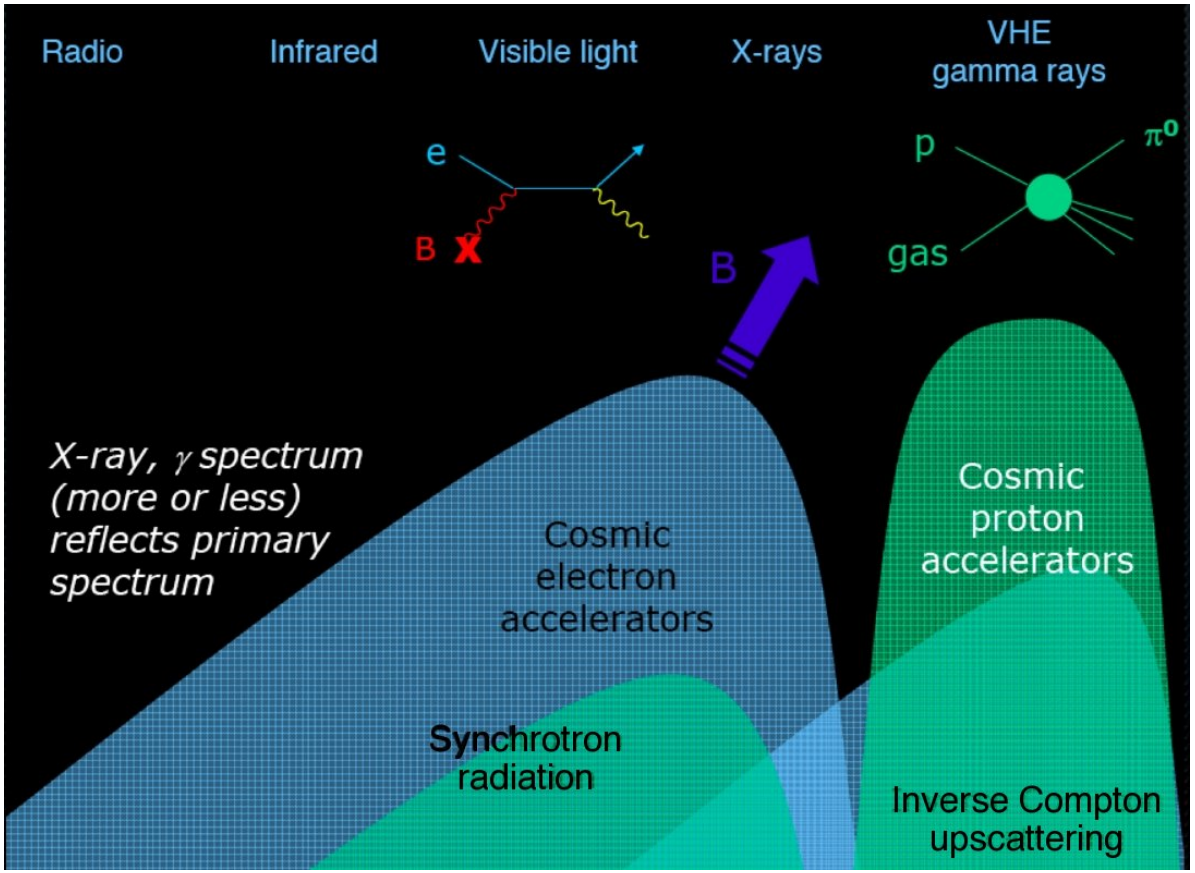






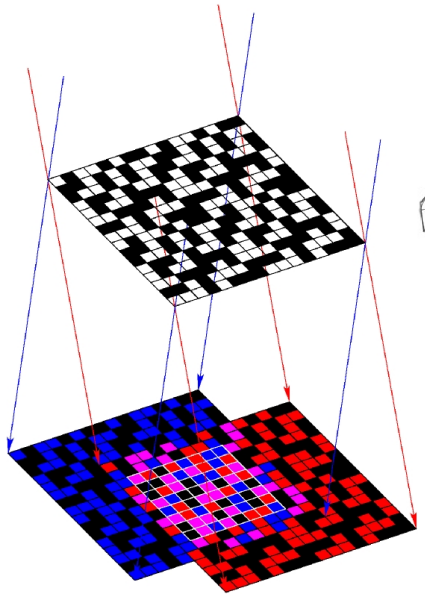


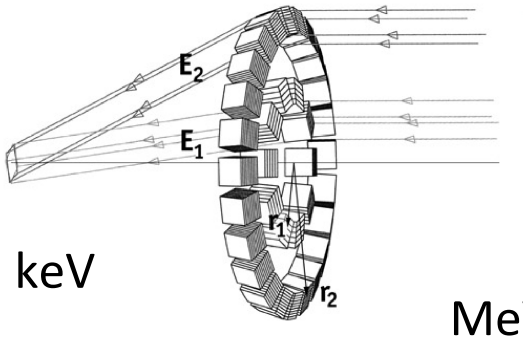
An inherent multi-wavelength endeavor



Non-thermal particles emit over the **entire** electromagnetic spectrum

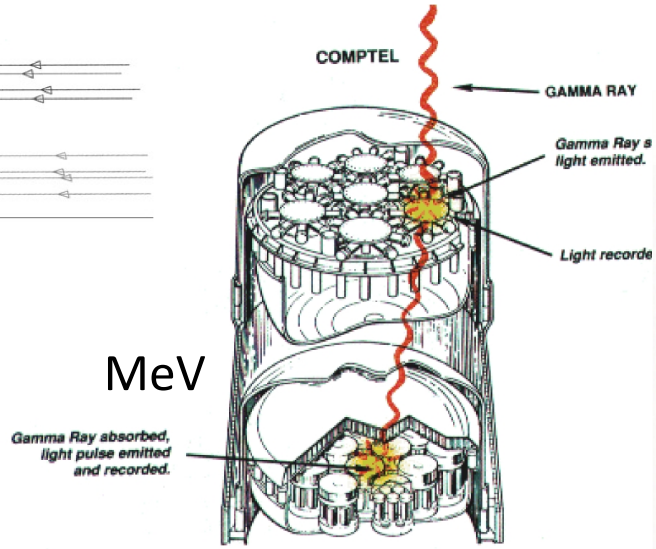
Observing gamma rays

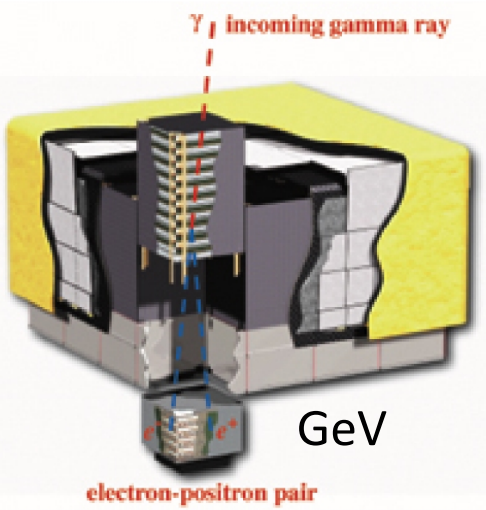




keV

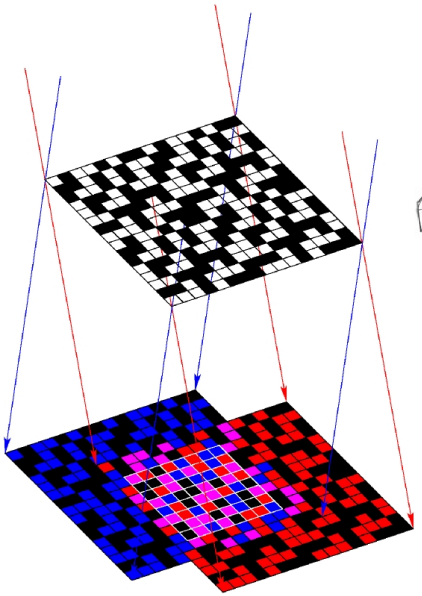
MeV

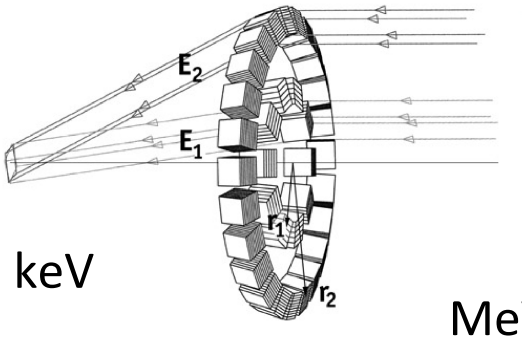




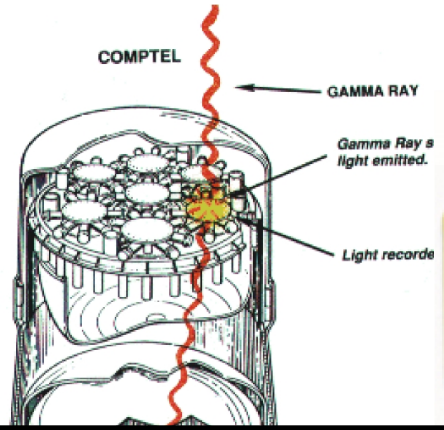
TeV

Observing gamma rays

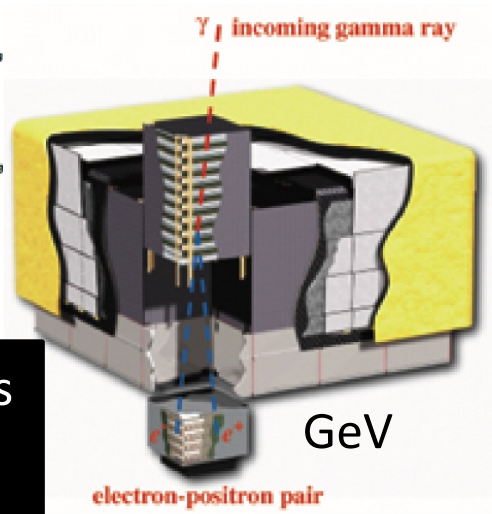




keV



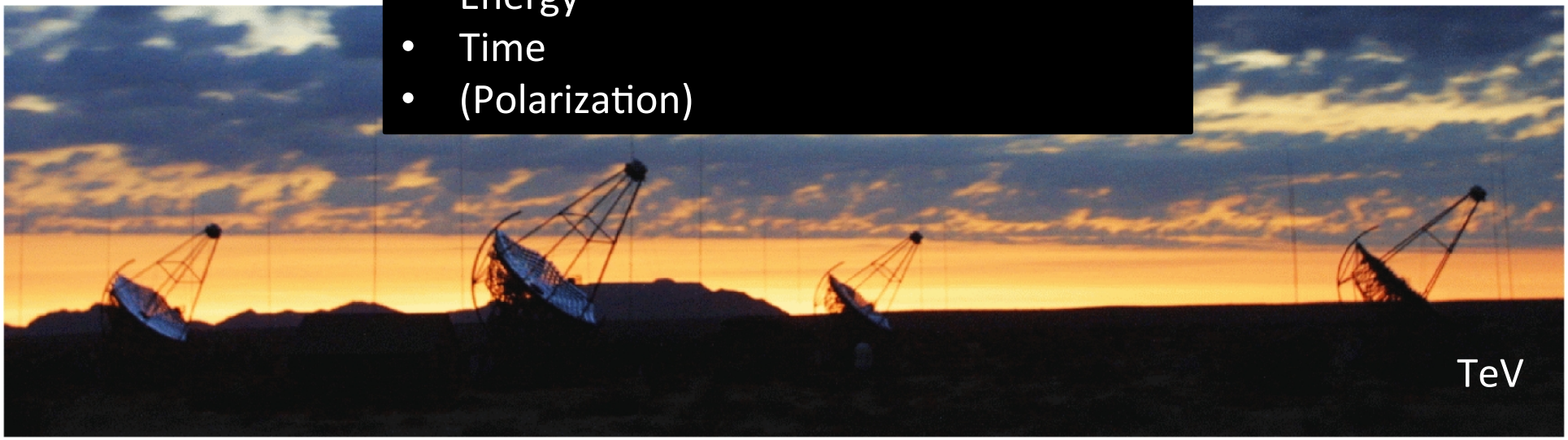
MeV



GeV

All instruments measure individual photons

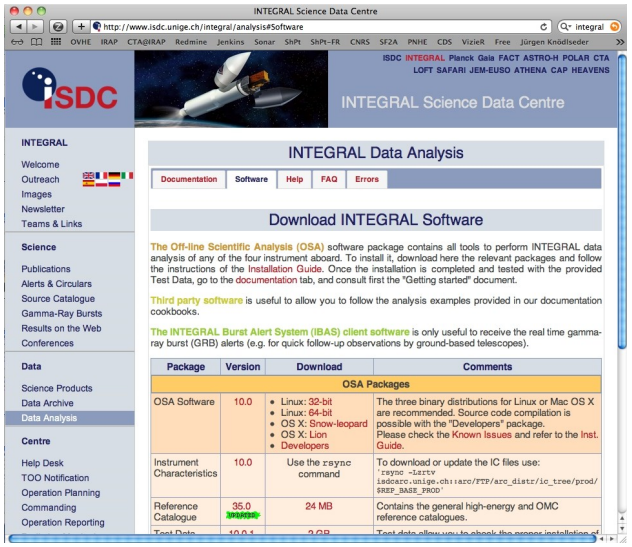
- Localization
- Energy
- Time
- (Polarization)



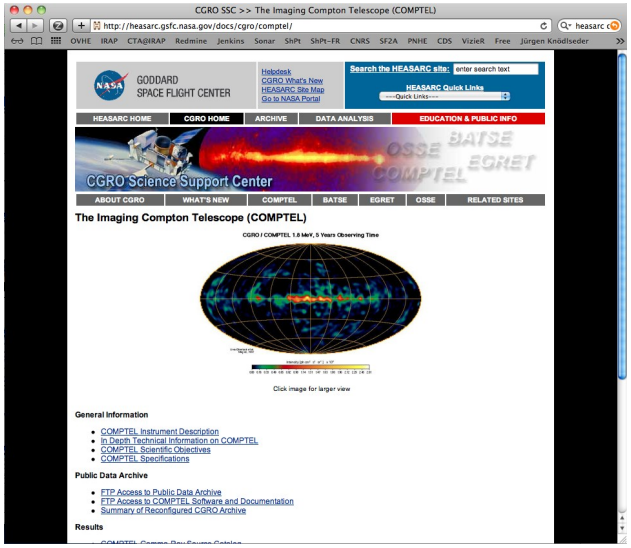
TeV

Analyzing gamma-ray data today

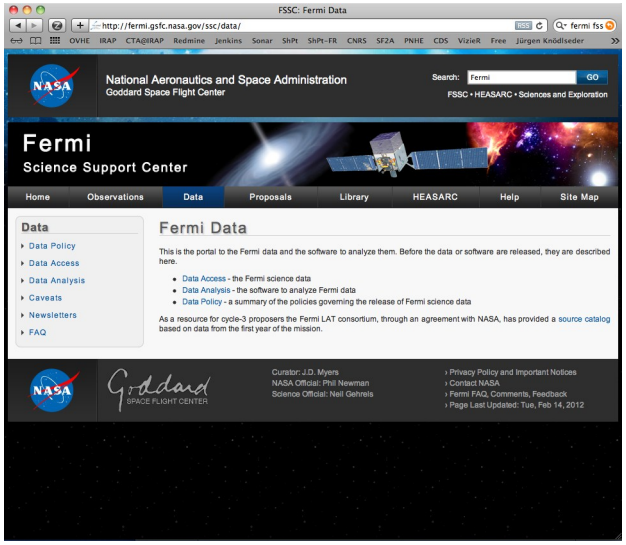
keV



MeV



GeV



INTEGRAL science analysis software (OSA) and data provided by ISDC (Geneva).

COMPTEL data provided by HEASARC/GSFC (US).

No public science analysis software available. **Data are unexploitable …**

Fermi science analysis software (Science Tools) and data provided by Fermi Science Support Center (HEASARC, US).

TeV: join a collaboration **(no public data and software so far …)**

Analyzing gamma-ray data tomorrow

Data (still missing)
and VO tools

Interoperability (still missing)

Software

What is GammaLib?

A self-contained, instrument independent, open source, multi-platform C++ library that implements all code required for high-level science analysis of astronomical gamma-ray data.

Self-contained: does not depend on external libraries (except for cfitsio for FITS file access)

Instrument independent: potentially supports any high-energy astronomy instrument; enables simultaneous multi-instrument analysis

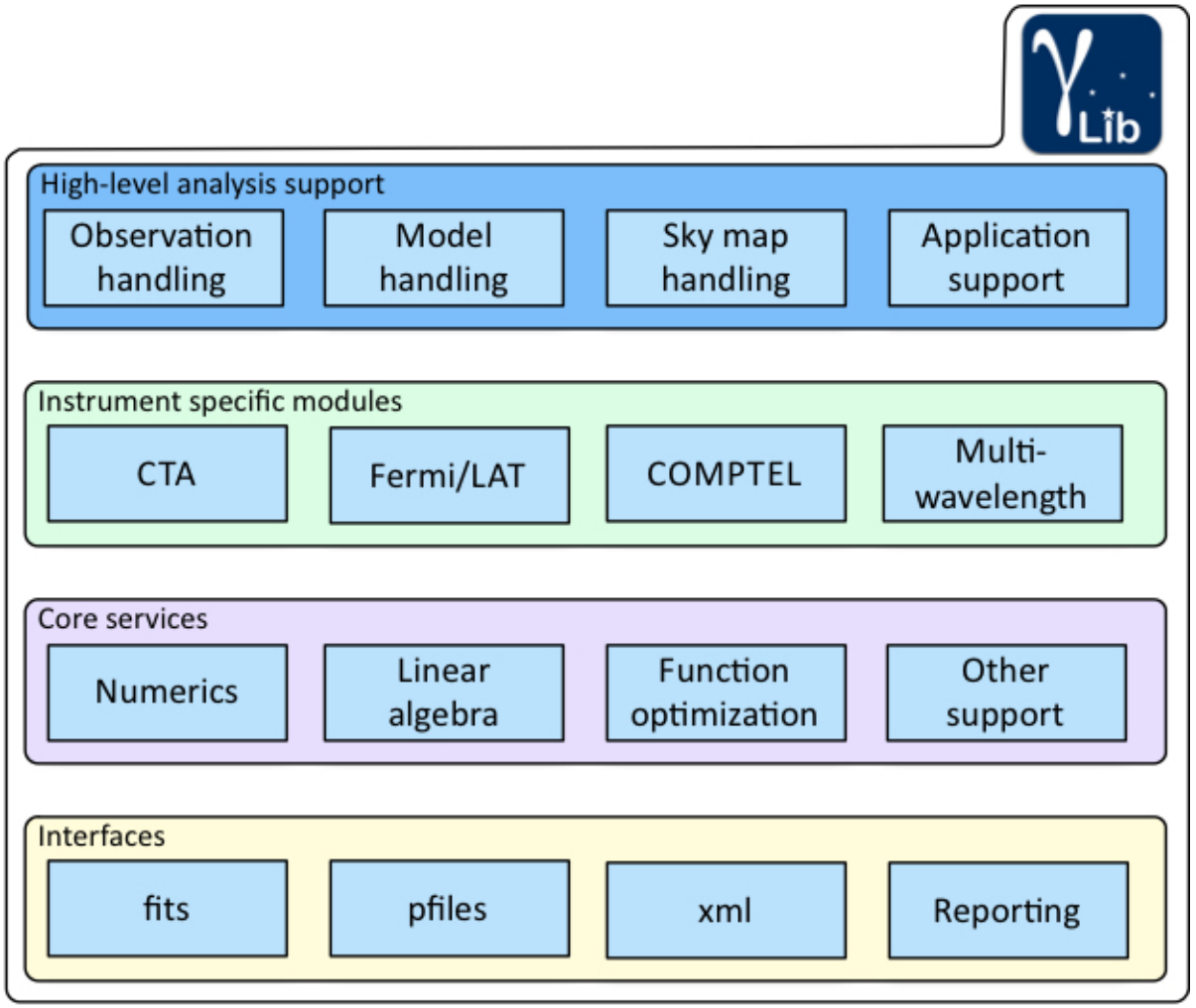
Open source: source code from `http://sourceforge.net/projects/gammalib/`

Multi-platform: compiles on any POSIX compliant Unix platform (Linux, Mac OS X, OpenSolaris, FreeBSD)

C++ library: object oriented framework, uses class abstraction for instrument independence

Python module: fully binds in Python (swig)

How is it organized?



Abstract

Extendable

Native

Compliant

How to install it?

1. Get it

2. Compile it (and check it)

```
> ./configure
> make
> make check
  ====================
  All 18 tests passed
  ====================
```

3. Install it

```
> [sudo] make install
```

4. Configure it

```
> export GAMMALIB=/usr/local/gamma
> source $GAMMALIB/bin/gammalib-init.sh
```

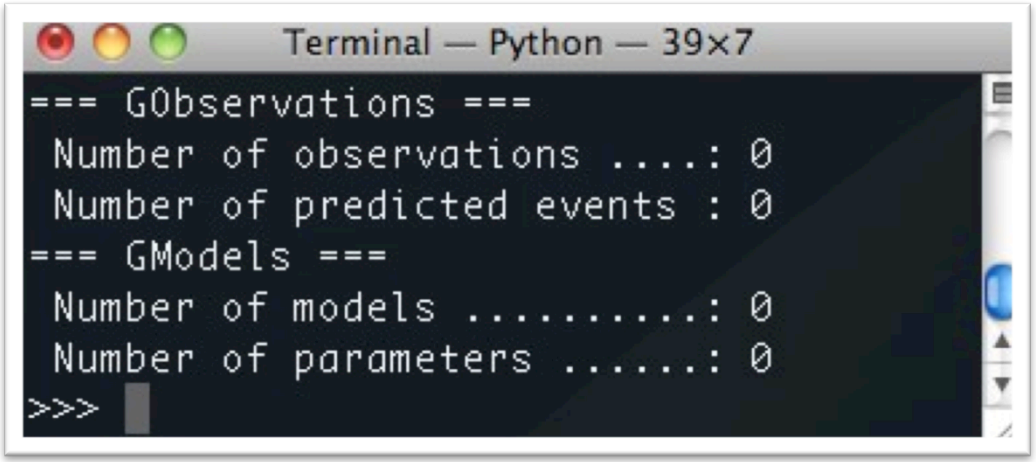
How to use it?

Build a C++ application:

```
#include <iostream>
#include "GammaLib.hpp"
int main(void) {
    GObservations obs;
    std::cout << obs << std::endl;
    return 0;
}
```

Use python:

```
$ python
>>> from gammalib import *
>>> obs = GObservations()
>>> print obs
```



A simple Use Case

Joint spectral analysis of the Crab nebula using data from COMPTEL (MeV), Fermi-LAT (GeV) and H.E.S.S. (TeV) *(credits: Marie-Hélène Grondin)*

1. Specify observation data (locally on disk)

```
<observation_list title="observation library">
  <observation name="Crab" id="vp0001_0" instrument="COM">
    <parameter name="DRE" file="/project-data/comptel/phase01/vp0001_0/m50439_dre.fits"/>
    <parameter name="DRB" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRG" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRX" file="/project-data/comptel/phase01/vp0001_0/m32171_drx.fits"/>
    <parameter name="IAQ" file="com/u47569_iaq.fits"/>
  </observation>
  <observation name="Crab" id="00001" instrument="LAT">
    <parameter name="CountsMap" file="/project-data/cta/data/fermi/crab/srcmap.fits"/>
    <parameter name="ExposureMap" file="/project-data/cta/data/fermi/crab/binned_expmap.fits"/>
    <parameter name="LiveTimeCube" file="/project-data/cta/data/fermi/crab/ltcube.fits"/>
    <parameter name="IRF" value="P7SOURCE_V6"/>
  </observation>
  <observation name="Crab" id="00023523" instrument="HESS">
    <parameter name="EventList" file="/project-data/cta/data/cta-1dc/data/hess/CTA1DC-HESS-run_(
    <parameter name="ARF" file="/project-data/cta/data/cta-1dc/data/hess/CTA1DC-HESS-run023523_s
    <parameter name="RMF" file=""/>
    <parameter name="PSF" file="/project-data/cta/data/cta-1dc/data/hess/CTA1DC-HESS-run0002352
  </observation>
</observation_list>
```

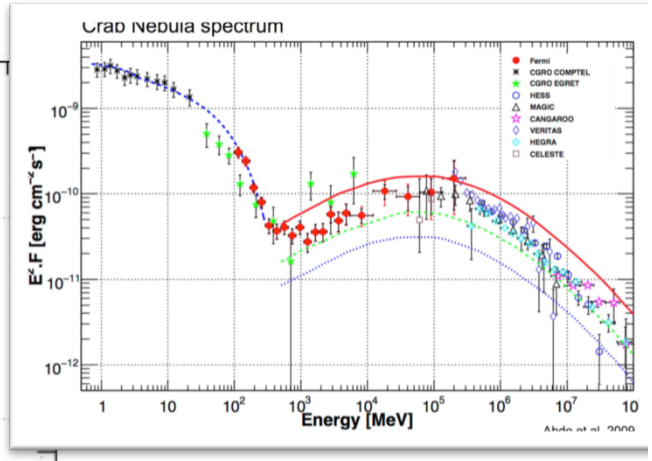
A simple Use Case

2. Specify a spatial-spectral-temporal model

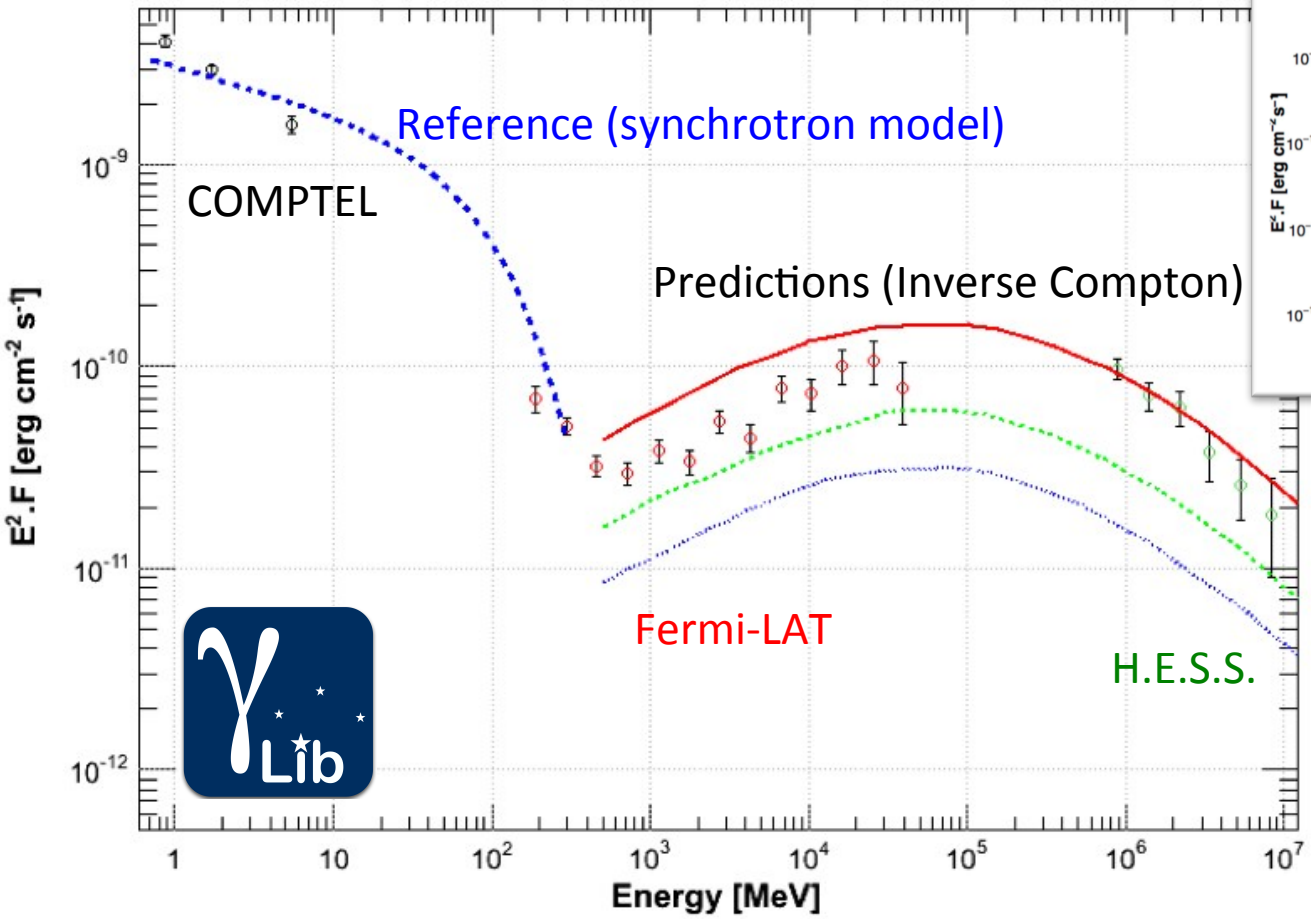
```
<source_library title="source library">
  <source type="PointSource" name="Crab">
    <spectrum type="NodeFunction">
      <node>
        <parameter scale="1.0" name="Energy" min="0.866" max="0.866" value="0.866" free="0"/>
        <parameter scale="0.00378740451299" name="Intensity" min="1e-5" max="1e5" value="1.0" f
      </node>
      ...
      <node>
        <parameter scale="1e6" name="Energy" min="20.0" max="20.0" value="20.0" free="0"/>
        <parameter scale="1.30650238384e-20" name="Intensity" min="1e-5" max="1e5" value="1.0"
      </node>
    </spectrum>
    <spatialModel type="SkyDirFunction">
      <parameter scale="1" name="RA" min="-360" max="360" value="83.6331" free="0"/>
      <parameter scale="1" name="DEC" min="-90" max="90" value="22.0145" free="0"/>
    </spatialModel>
  </source>
  <source type="DiffuseSource" name="Extragal_diffuse" instrument="LAT">
    <spectrum type="FileFunction" file="/project-data/cta/data/fermi/diffuse/isotrop_2year_P76_
      <parameter scale="1.0" name="Normalization" min="0.0" max="1000.0" value="1.0" free="1"/>
    </spectrum>
    <spatialModel type="ConstantValue">
      <parameter scale="1" name="Value" min="0" max="10" value="1" free="0"/>
    </spatialModel>
  </source>
  ...
</source_library>
```

A simple Use Case

Independent analyses

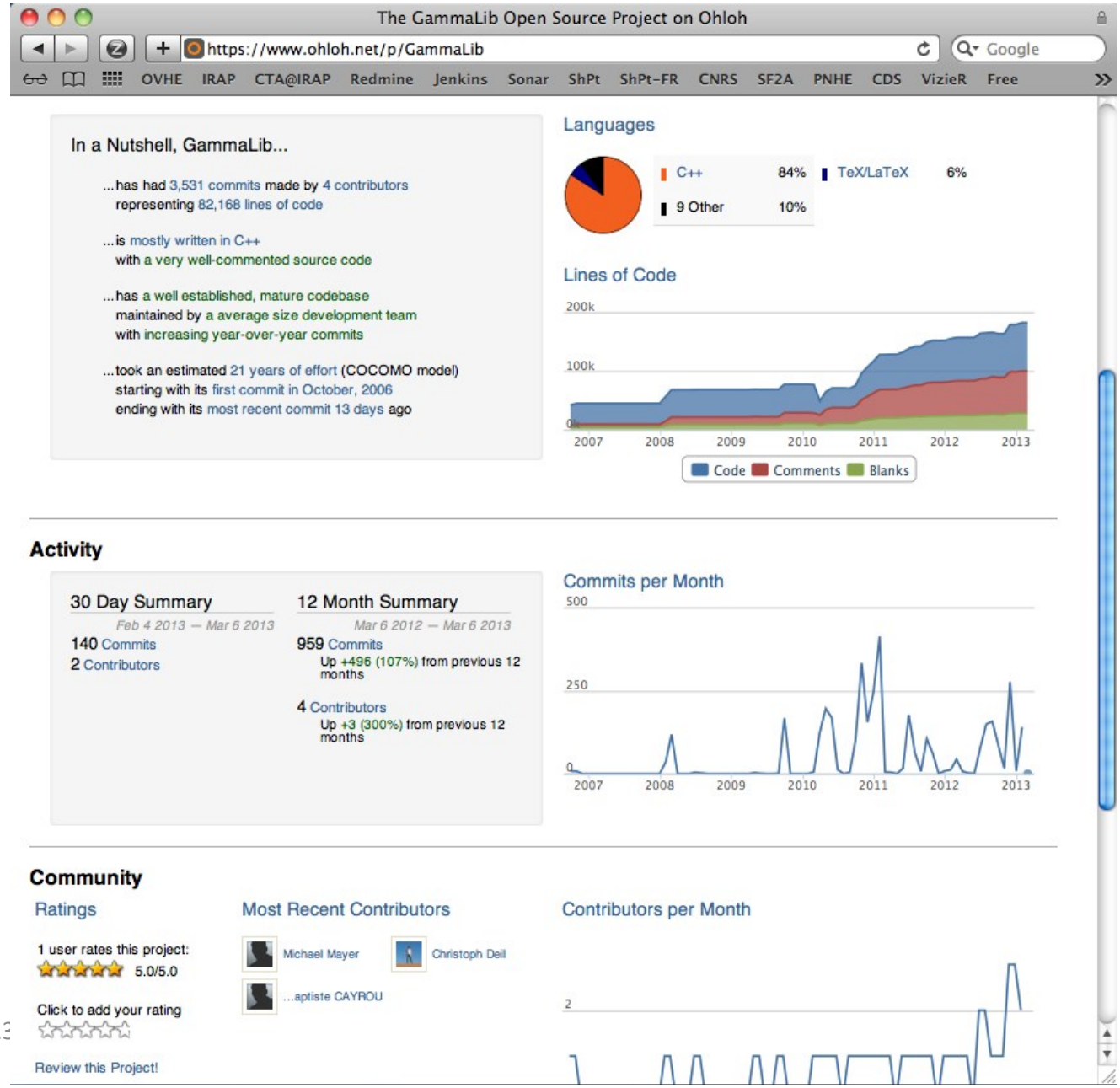


Abdo et al. (2009)



Credits: Marie-Hélène Grondin

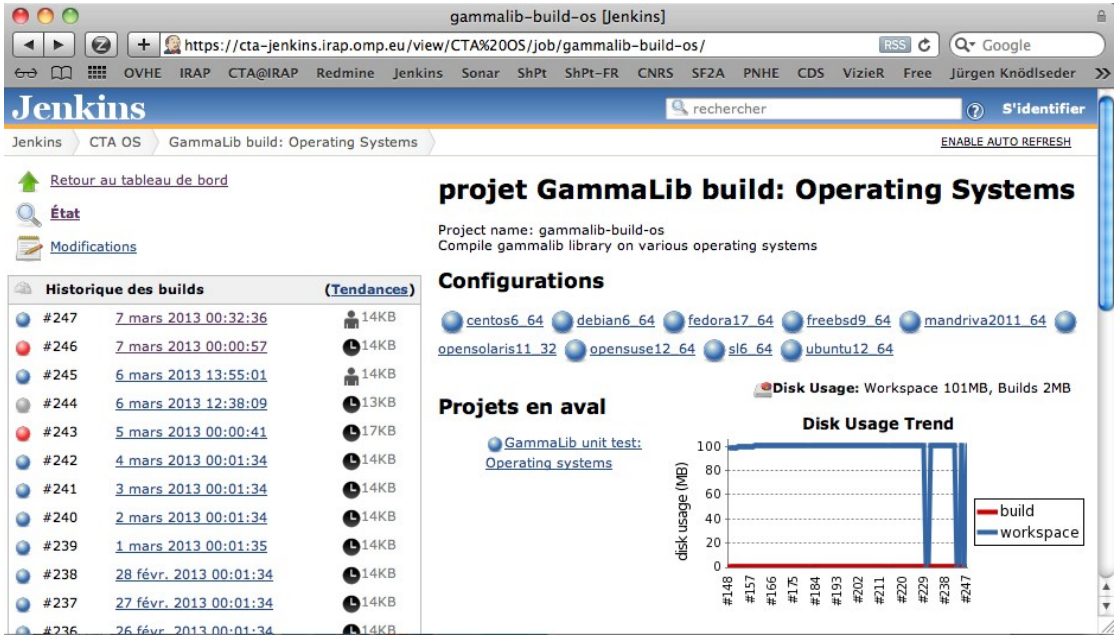
A collaborative development

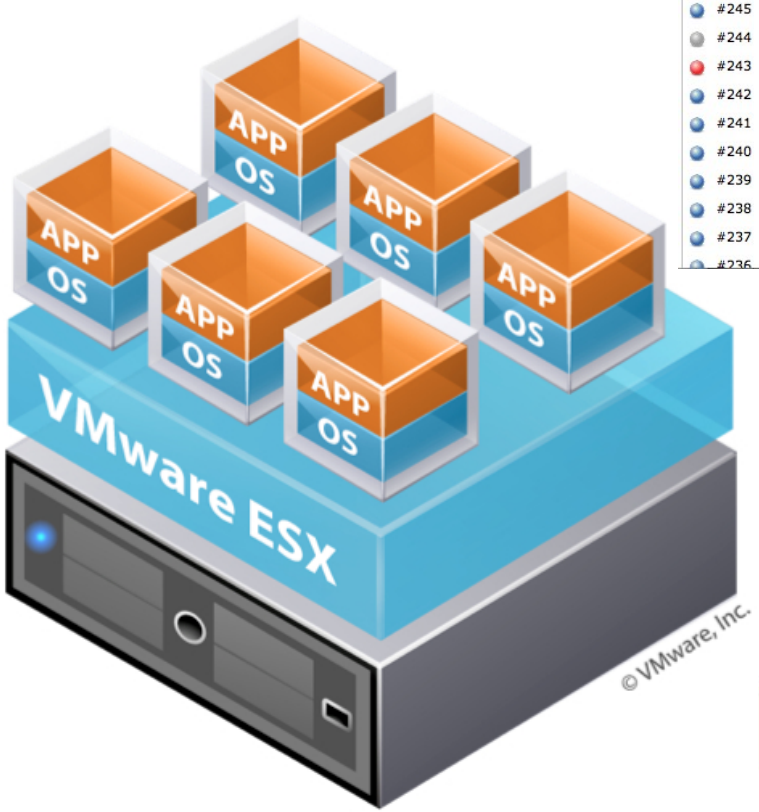


Code quality (continuous integration)

Jenkins

Continuous integration
(build, check, install, analysis)



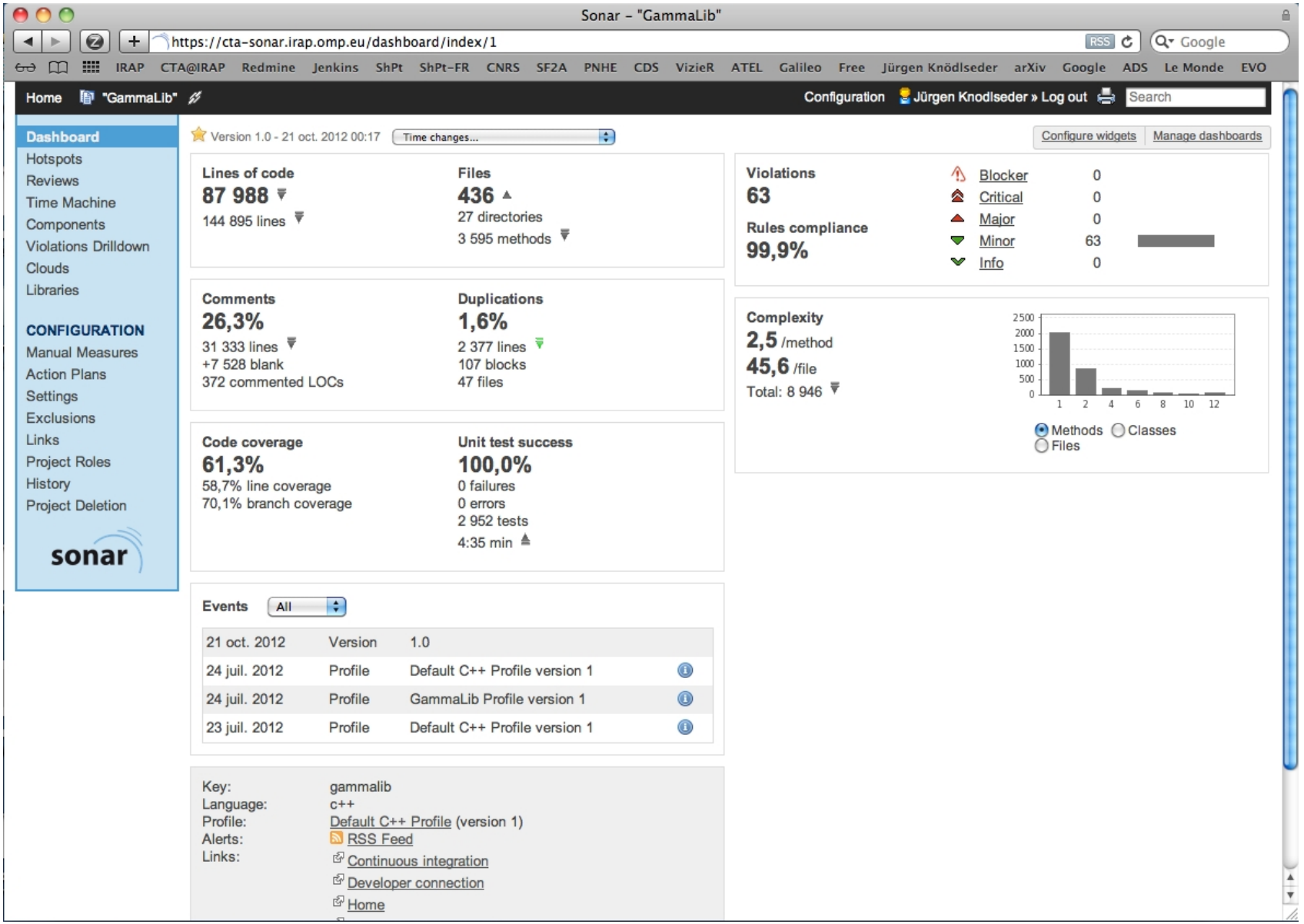


Multi-platform, multi-compiler, 32/64 Bit,
Python version, swig version, …

Virtual box with relevant OS

vmware®

Code analysis



Want to know more?