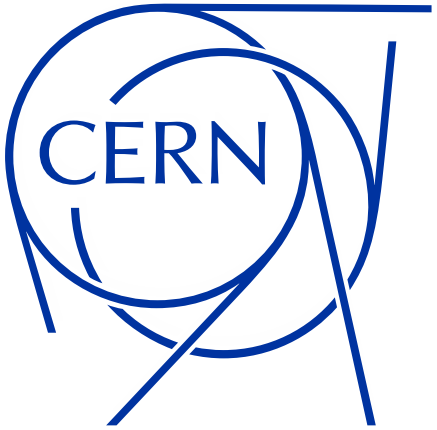


LGC open-source:
a strategy to share adjustment software and algorithmic development

G. Kautzmann, J. Gutekunst, F. Klumb, P. Elson

10/10/2024

LGC Overview

What does it do? From where does it come from? How is it used?

How is it integrated? What does QA look like?

LGC?

“Logiciel Général de Compensation” → “General Adjustment Software”

What does LGC do?

Process the position of objects and their associated precisions using geometrical observations (distances, angles, etc..) through a weighted least square statistical analysis

For what?

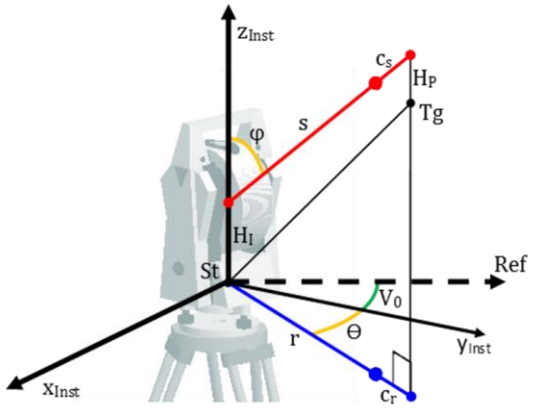
Giving surveyors a reliable way to process and analyse their measurements with a non-black box software

LGC Principle

Observation model $\boldsymbol{observation} = f_{parametric}(\boldsymbol{unknows})$

Constraint model $f_{constraint}(\boldsymbol{unknows}) = 0$

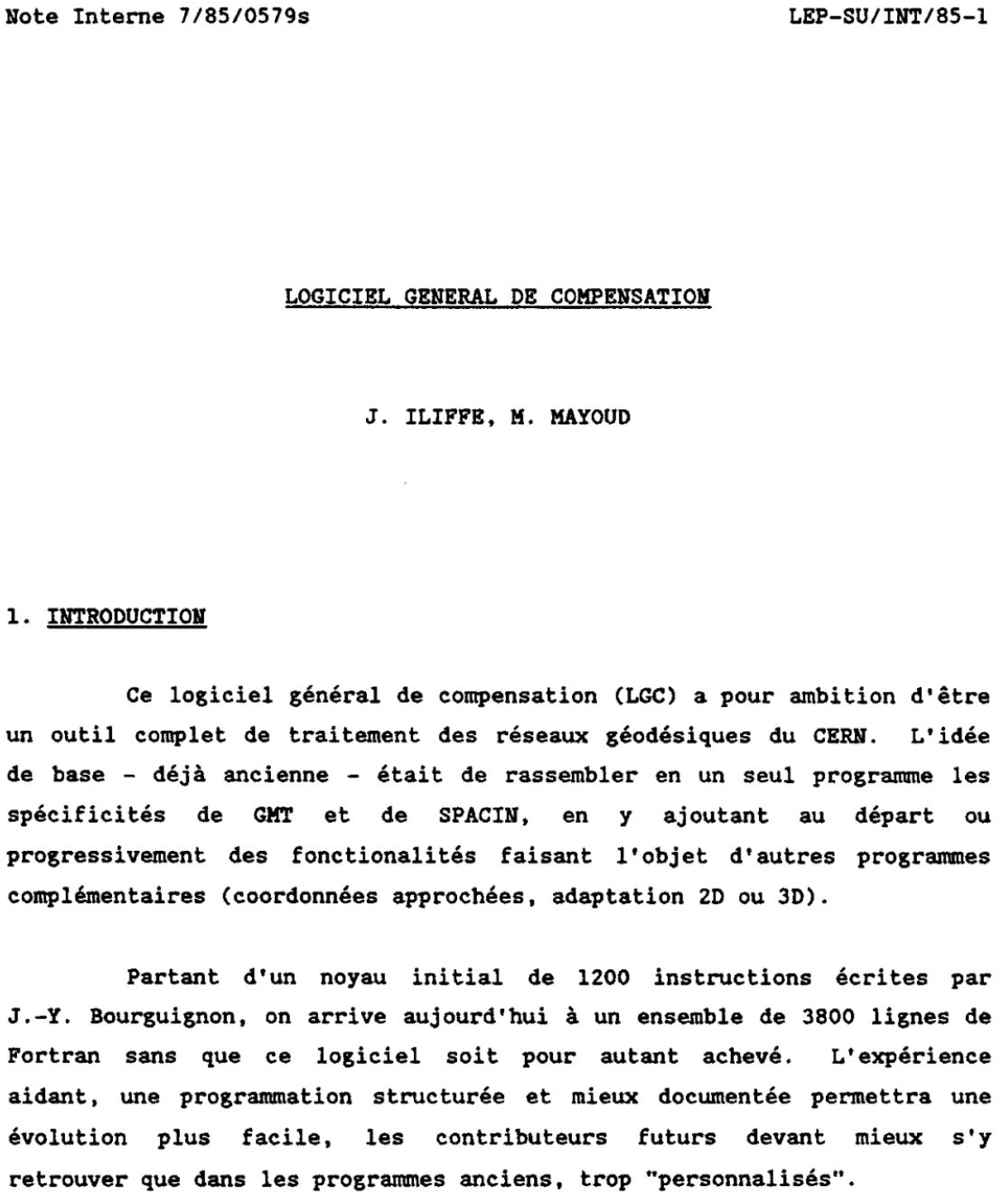
Stochastic model $\boldsymbol{\sigma_{obs}^2} = \mathrm{f}(\mathbf{observation}, \boldsymbol{\sigma_{a-priori}})$



Weighted Least Square model $\sum wv^2 = w_1 v_1^2 + w_2 v_2^2 + \cdots + w_n v_n^2 = minimum$

LGC: A bit of history

First written trace dating back to 1985



Note Interne 7/85/0579s LEP-SU/INT/85-1

LOGICIEL GENERAL DE COMPENSATION

J. ILIFFE, M. MAYOUD

1. INTRODUCTION

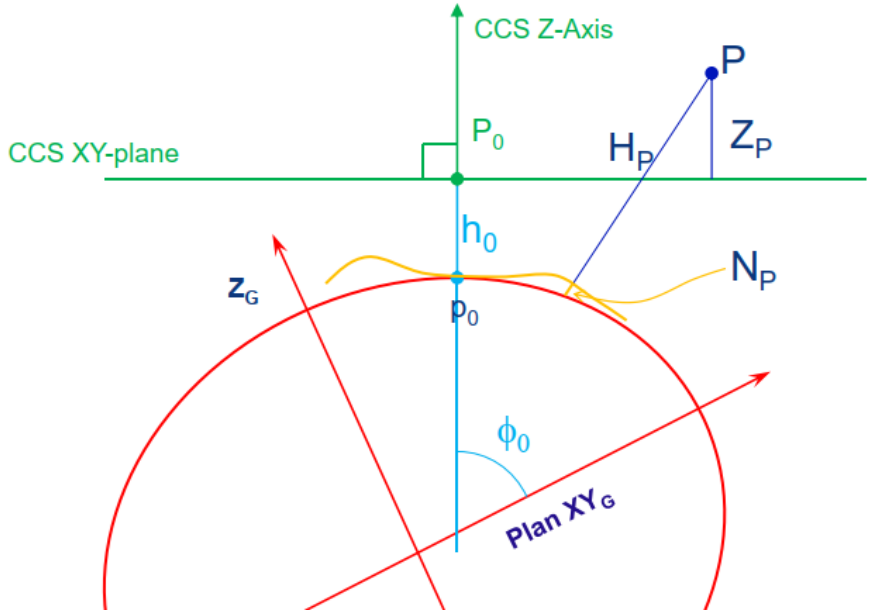
Ce logiciel général de compensation (LGC) a pour ambition d'être un outil complet de traitement des réseaux géodésiques du CERN. L'idée de base - déjà ancienne - était de rassembler en un seul programme les spécificités de GMT et de SPACIN, en y ajoutant au départ ou progressivement des fonctionalités faisant l'objet d'autres programmes complémentaires (coordonnées approchées, adaptation 2D ou 3D).

Partant d'un noyau initial de 1200 instructions écrites par J.-Y. Bourguignon, on arrive aujourd'hui à un ensemble de 3800 lignes de Fortran sans que ce logiciel soit pour autant achevé. L'expérience aidant, une programmation structurée et mieux documentée permettra une évolution plus facile, les contributeurs futurs devant mieux s'y retrouver que dans les programmes anciens, trop "personnalisés".

Ambitions

- Be the unified tool to compute geodetical networks at CERN
- 3D compensation
- Considering the local geoid model

LGC: A bit of history



First Fortran Version developed in the 80's

LGC v1 - C++ version in the 2000's

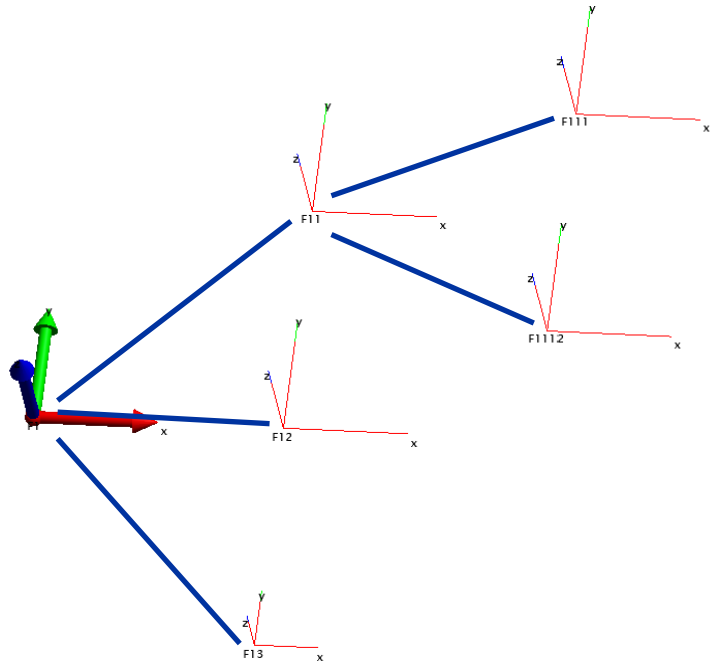
New observation and stochastic models

New library (**SurveyLib**) shareable with other projects

LGC v2 - in 2016:

Important refactoring

Introduction of the FRAME concept

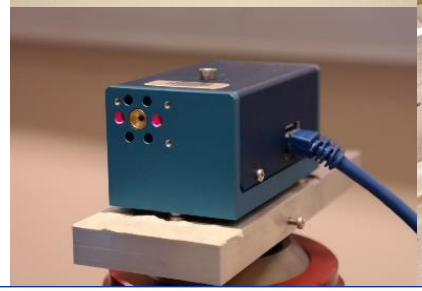


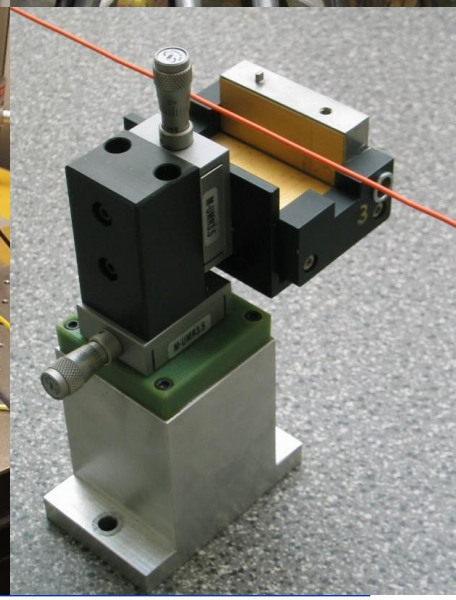
Actual version: v.2.7.0 (May 2024)

Observation models

21 observation models implemented today

- Total Station / Laser Tracker
- Levels
- Distance measurement
- Offset to wire
- BCAM
- Basic Coordinate Measurement Machine
- Inclinometers
- Gyroscope
- HLS
- WPS
- …





Example

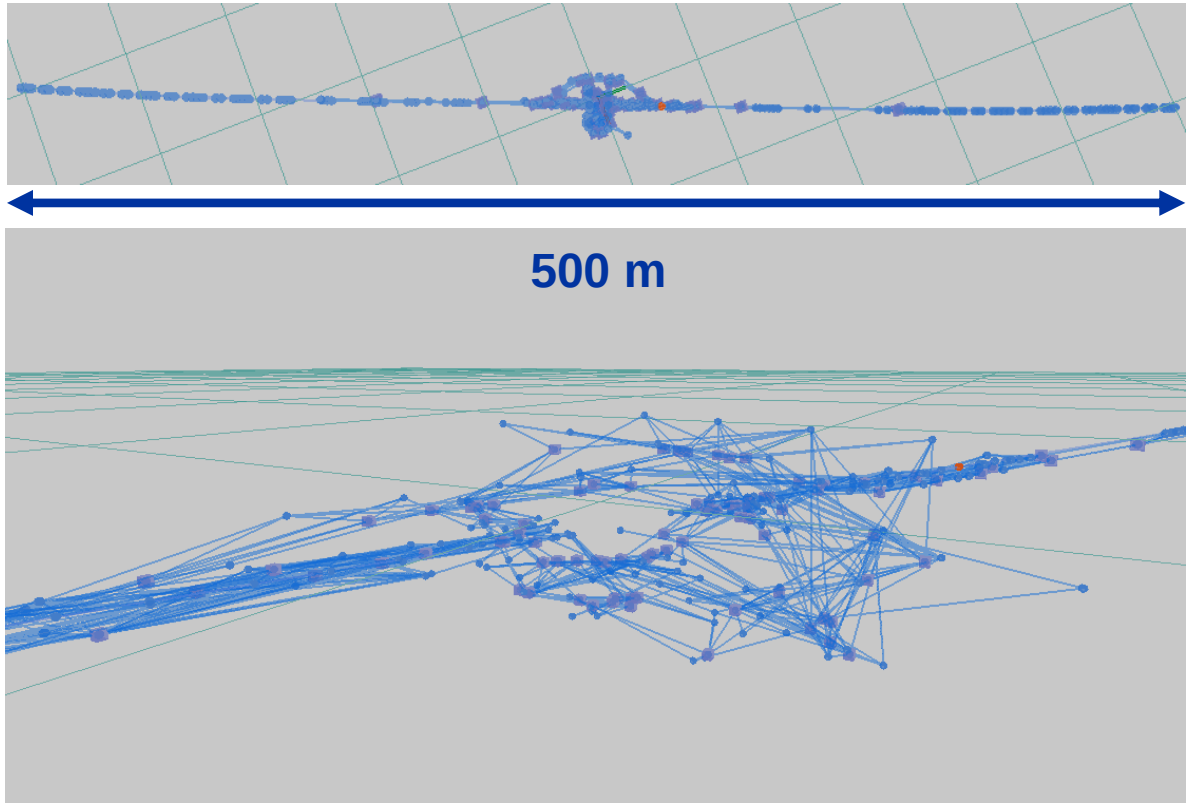
2019 IP 8, mixing Total Station, Offset to wire and Gyro (3870 obs, 1208 unkn.)

Input file Extract

```
*TITR
IP 8 Computation EYETS 2019
*RS2K
*FAUT      .01      .10
*INSTR
*PDOR
*CALA
*POIN
*VXY
*TSTN    LHC.GGPSO.1L804.    AT402.392052     IHFIX    IH   1.583900
*V0
*ANGL
LHC.MBXWH.1L8.S                              306.123102
LHC.MBXWH.1L8.E                              289.322504
LHC.POTCE.T07C02.                            374.364281
LHCB.PLATE.00B00+0P.                         316.544347
LHCB.PLATE.00A00+0P.                         315.236698
LHC.NID.1L8-01.                              320.370438
LHC.NID.1L8-02.                              345.599364
LHC.NID.1L8-03.                              106.588766
LHC.NID.1L8-04.                              136.529148
LHC.NID.1L8-07.                              111.439343
LHC.NID.1L8-09.                              117.122844
```

...

Associated visualization



Interacting with LGC today

SurveyPad Interface

(v1.1.0 official release in May 2024)

Input file

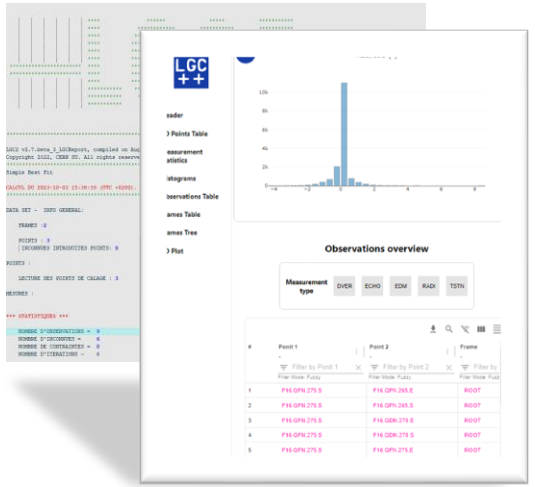
```
*TITR
Simple Best Fit
*OLOC
*CHABA
*INSTR
*CALA
PT1      1000.321      1999.111      3000.32
PT2      1000.243      1999.576      3001.661
PT3       999.292      2000.005      3000.707
*FRAME BF 0 0 0 0 0 0 1 TX TY TZ RX RY RZ
*OBSXYZ
PT1 1 0 0 0.001  0.001 0.001
PT2 1 1 1 0.001  0.001 0.001
PT3 0 0 1 0.001  0.001 0.001
*ENDFRAME
*END
```

API

Command line mode

***lgc.exe** –i inputPath*

Result files



Fileless mode

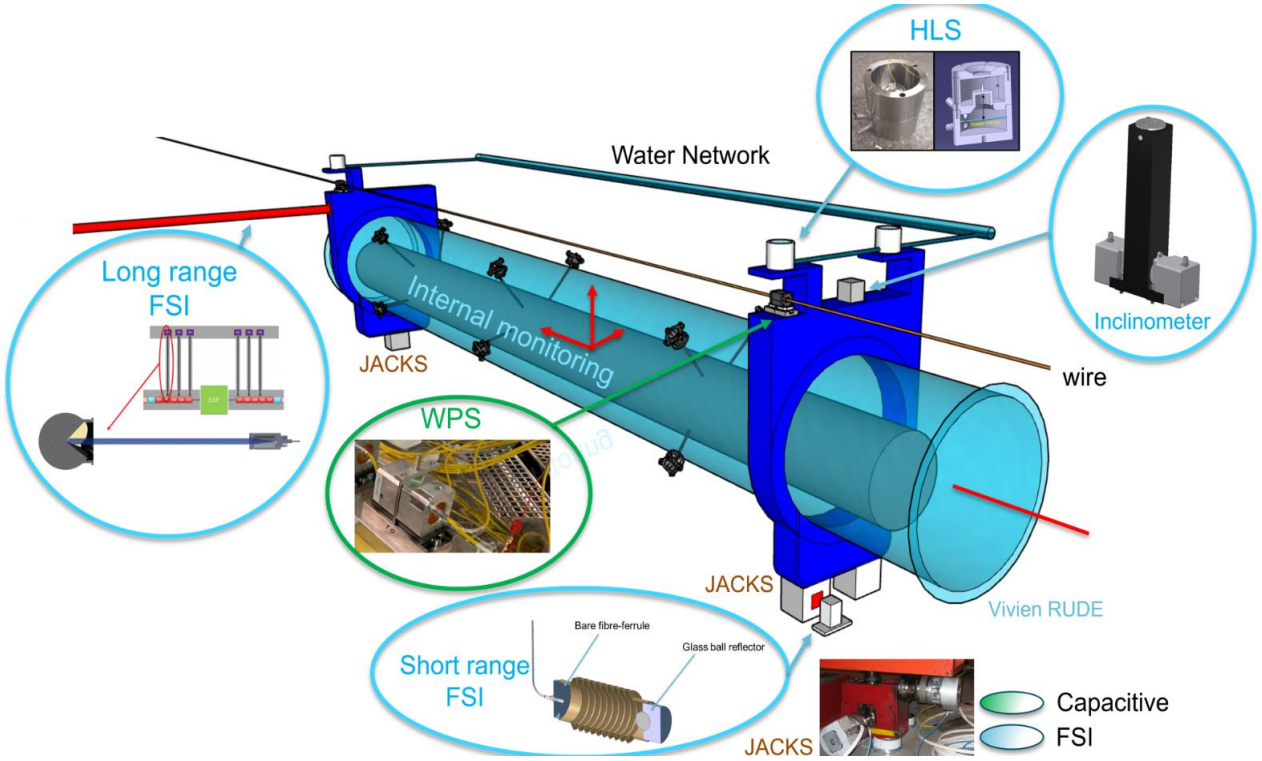
```
myLGC.updateMeas(id, new_measurement);
status = myLGC.adjust();
myLGC.getPointEstimatePrec(pointName, "ROOT");
myLGC.getSigma0();
```

Other APP
(ex.: Monitoring)

LGC usage

Used extensively by CERN Surveyors

- **Main adjustment software**
- Integrated in **SurveyPad** Interface and within the in-house field acquisition Software **TSUNAMI** for most of the computations
- Creation of input file and reading result via **GEODE** (web interface to CERN Survey Database)
- Computation core for **Monitoring Systems**
- Used for **R&D** projects (HL-LHC, Survey Train, FCC)



Courtesy: BE-GM-HPA

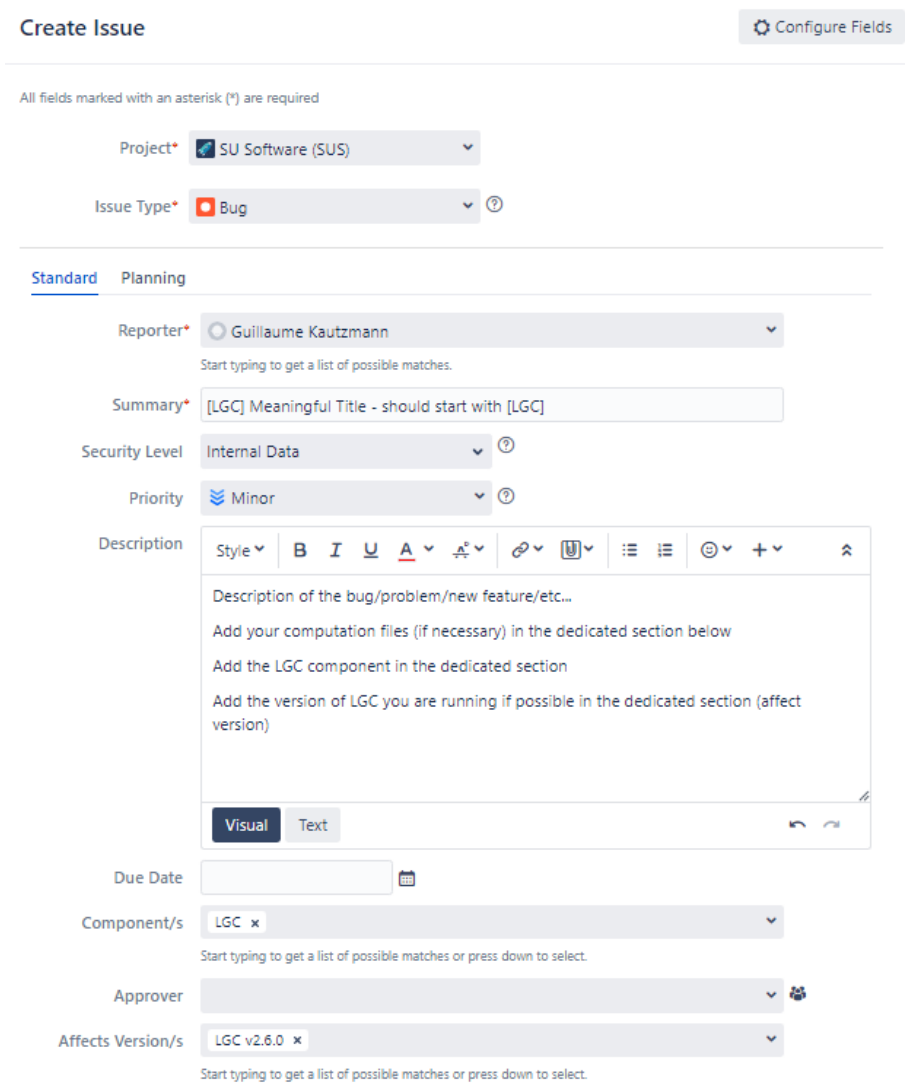
QA: Ticketing

JIRA in use for ticketing and planning

Widely adopted by our user base

Regular meeting to define the priorities

Commercial product, not open to CERN externals



QA: Gitlab.cern.ch

Systematic Code review for every feature

CI/CD: automated tests, generation of executables/installation packages

Runners on CERN infrastructure

Releases ~2 times a year / Beta version more frequent

Not open to CERN external

In summary

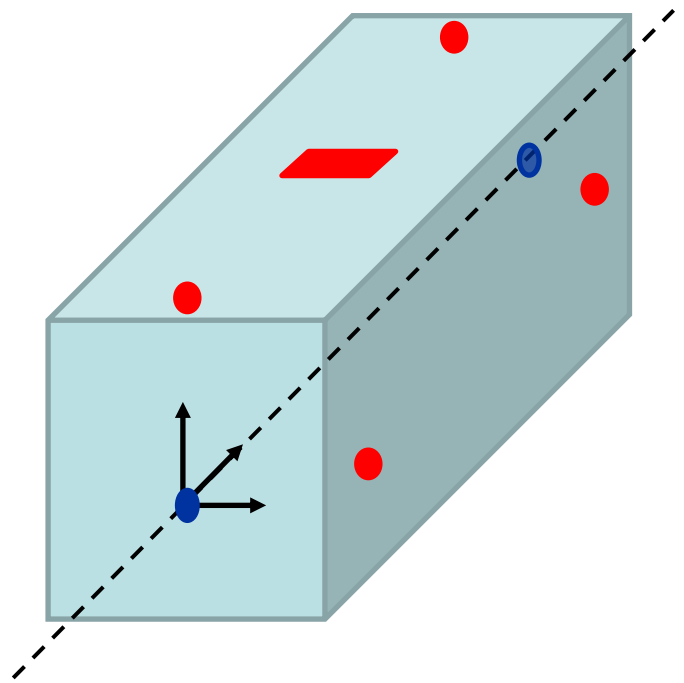
There are a lot of functionalities available

LGC covers simple to complex networks

Used every day by CERN Surveyors

40 years of testing and development

QA workflow well in place



An element as a rigid floating body with the FRAME keyword

→ allow beam point alignment with their associated statistics

LGC Open-Source

Why? Why now?

Strategy and status?

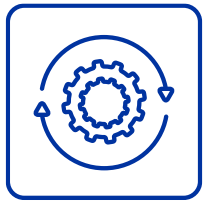
Expectations and challenges foreseen?

What is open-source ?

“Open source is source code that is made freely available for possible modification and redistribution.

Products include permission to use the source code, design documents, or content of the product.

The open-source model is a decentralized software development model that encourages open collaboration.



A main principle of open-source software development is peer production, with products such as source code, blueprints, and documentation freely available to the public.

The open-source movement in software began as a response to the limitations of proprietary code.”

Source: Wikipedia

Main motivations

Allow other particle physics institutes and academic partners to use and contribute to it.

Potentially have a bigger LGC community to get contributions, code review, bug reporting, improvement ideas, etc. → more use cases and mutual aids!

Develop an open ecosystem of post- and pre-processing tools (ex.: Python scripts) shareable within the community.

Overall increase LGC quality and transparency

Why Now?

Some exchanges in the past with CERN Knowledge Transfer services

European Surveying Accelerator Seminars
wish to use and contribute to a common adjustment software based on transparent/documented algorithms

Establishment of the CERN Open-Source Program Office (OSPO) in 2024


@ESRF 2023


@CERN 2024

CERN Open-Source Program Office

“CERN Open-Source Policy” adopted end 2022

Both for Open Software and Hardware

Internally

- Consult, advise, train on Open-Source best practices, tools, licenses, etc…
- Advise on open-sourcing CERN software, gateware and hardware
- Identify/track FOSS dependencies for critical services
- Advise CERN management on Open-Source matters

Externally

- Facilitate partnerships with external entities
- Showcase CERN contributions to Open Source
- Promote CERN as an Open-Source lab

Source: J. Serrano, indico.cern.ch/event/132756

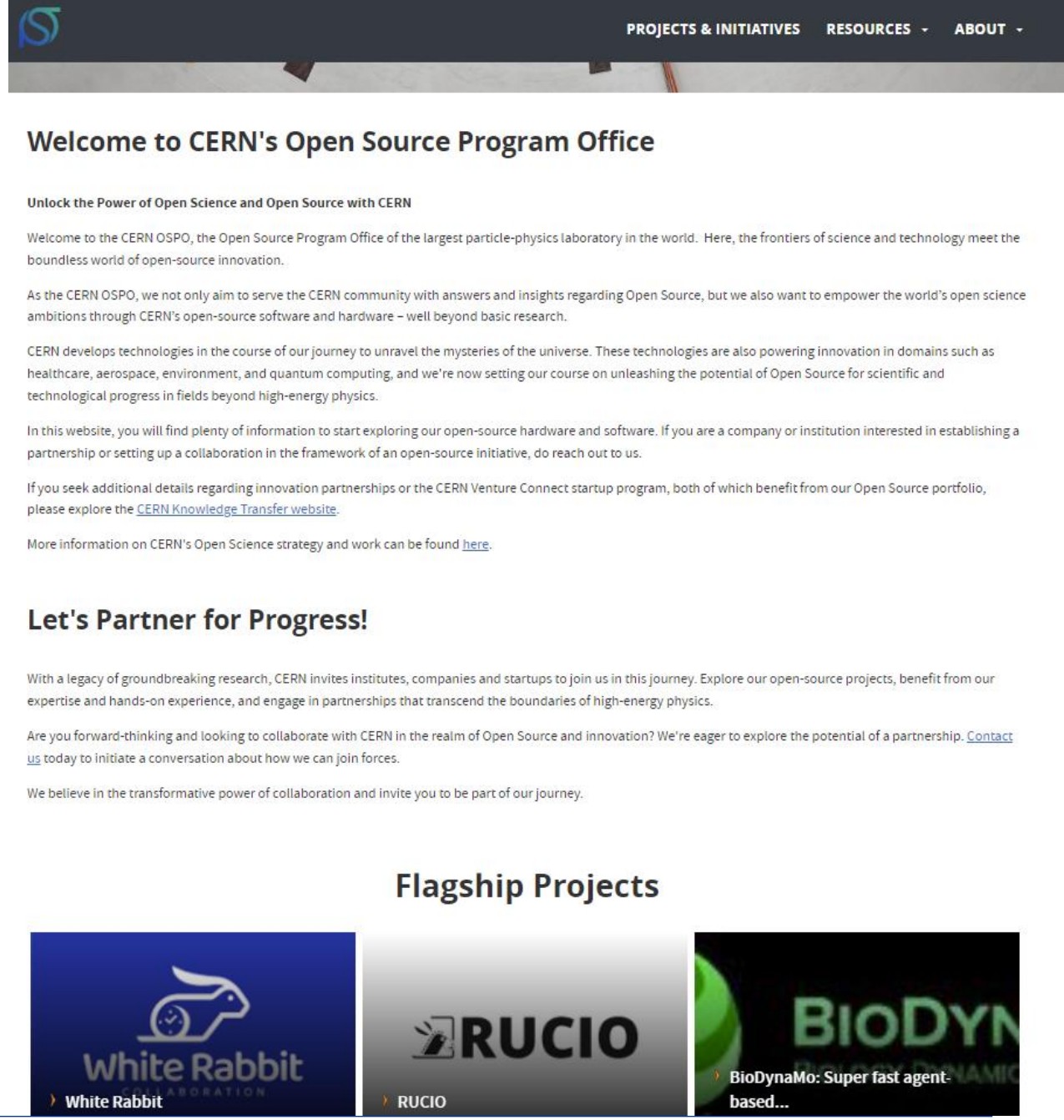
CERN OSPO

More information on:

https://opensource.web.cern.ch/

Additional actions

- Maintain a catalogue of open-source software and hardware published by CERN
- Organise courses in collaboration with Learning & Development Group
- Liaise with Purchasing Group in matters regarding the procurement of open-source products
- Define metrics for Open-Source adoption at CERN and publish a yearly report



Source: J. Serrano, indico.cern.ch/event/132756

Strategy (1/3) : Preparation and Planning

Audit by the OSPO of LGC and SurveyLib codebases

Check that all components, including third party libraries are compliant with the public distribution

Implementation of OSPO recommendation and guidelines

Review of the documentation

Promotional effort and setting up communication channels

Strategy (2/3) : Launch and promotion

Launch will introduce

- LGC and SurveyLib as Open-Source
- GUI (SurveyPad) will be distributed as binaries

Outreach efforts are planned

- Training sessions for beginners
- Presentations to academic sector
- Contacts with some potential industrial partners

Strategy (3/3) : Growth and Sustainment

Iterative Development phase

- Updates considering community feedback
- Keeping the Software relevant and effective

Communication efforts

- Moderate discussion and manage contributions
- Maintain a constructive community environment
- Regular roadmap updates

Short term expectations

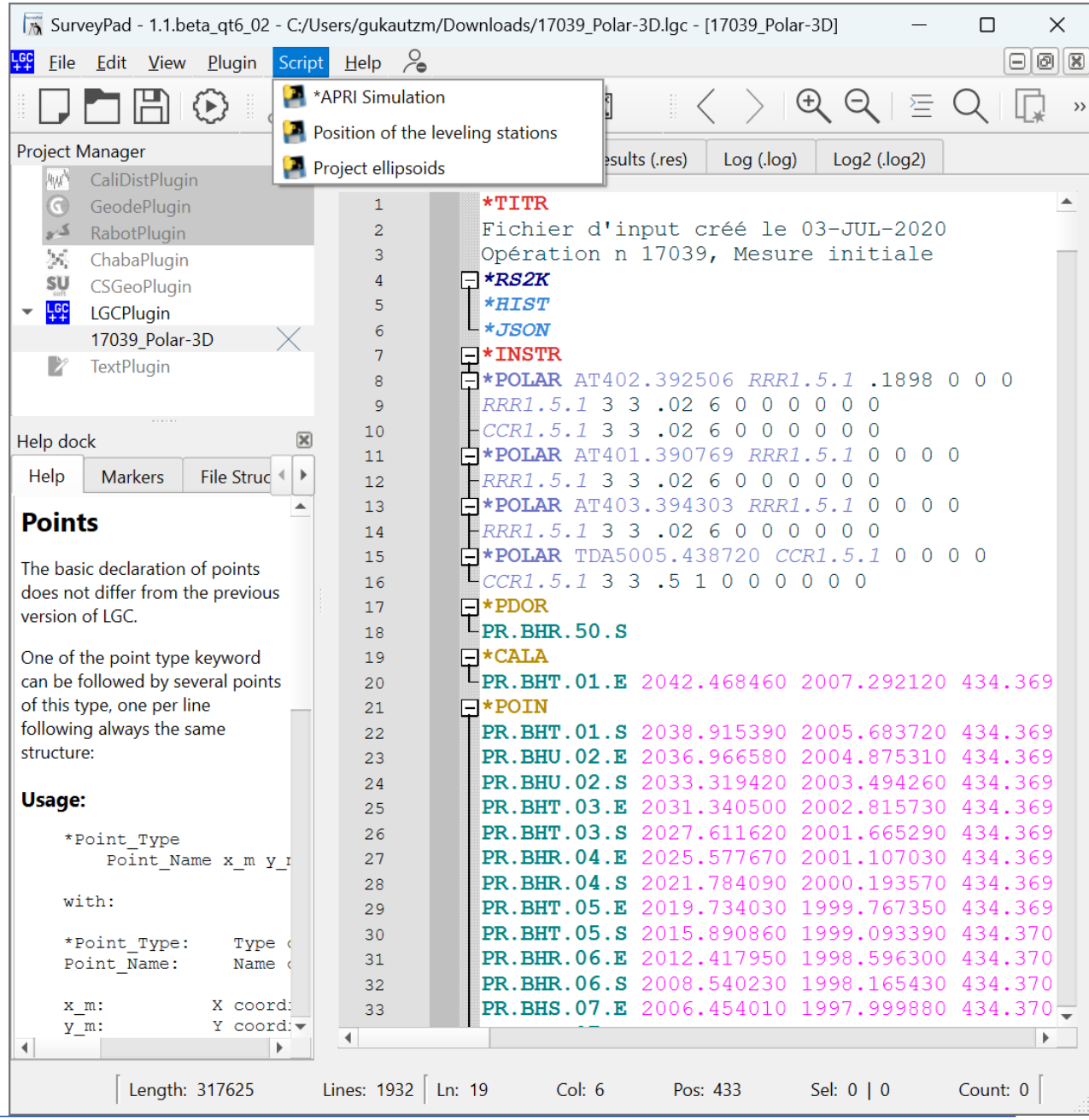
Contribution to

- C++ codebase: minor
- LGC ecosystem: tools to pre- or post-process data
- New ideas, process

SurveyPad has a scripting option that allows anyone to write and share scripts

Example of potential contribution

Translate popular measurement format into LGC one (SA, STAR*NET, etc…)



HTML Report example

Traditional Text result file generated by LGC

No interactivity at all (static display)

Touching the C++ code base

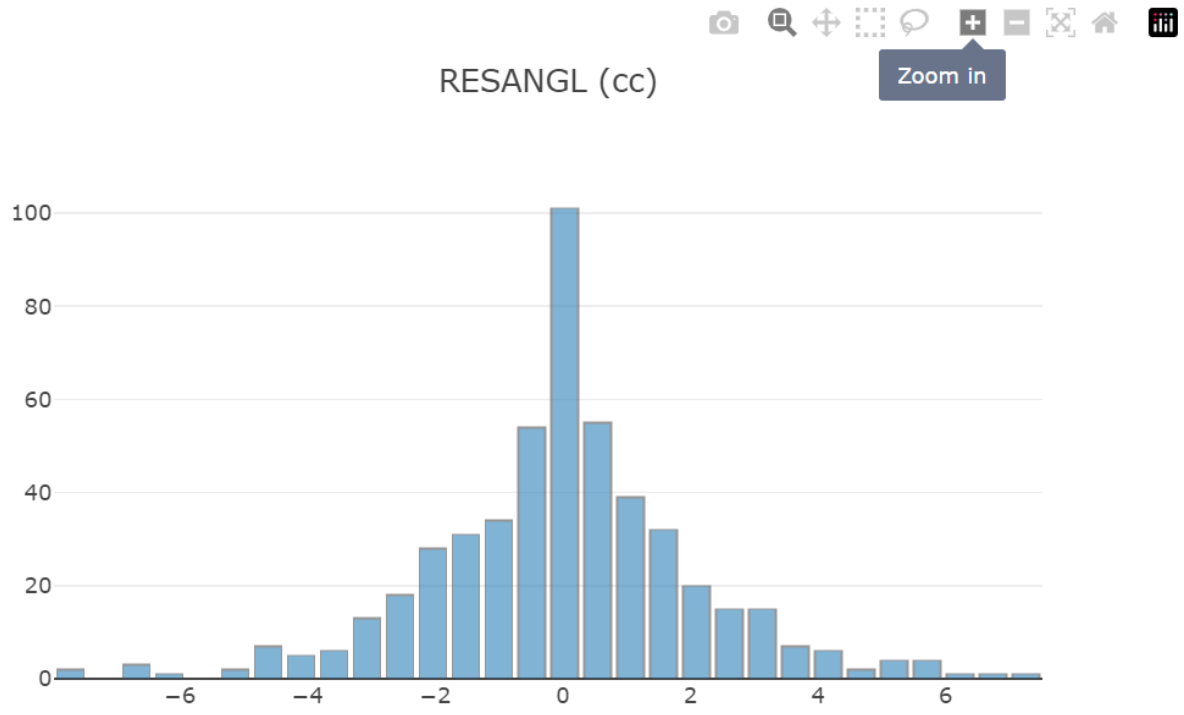
```
REPARTITION DES ANGL   (CC)

-9.00
 +
-8.00
 +    **
-7.00
 +    ****
-6.00
 +    **
-5.00
 +    *********
-4.00
 +    *********o*********o
-3.00
 +    *********o*********o*********o**
-2.00
 +    *********o*********o*********o*********o*********o********
-1.00
 +    *********o*********o*********o*********o*********o*********o*********o*********+... 123
0.00
 +    *********o*********o*********o*********o*********o*********o*********o*********+... 126
1.00
 +    *********o*********o*********o*********o*********o*********o******
2.00
 +    *********o*********o********
3.00
 +    *********o*********o*
4.00
 +    ******
5.00
 +    *******
6.00
 +    **
```

HTML report based on the JSON output

Interactive plots, tables, etc.…

No need for LGC expert to develop it



Challenges

Core development team is small and has other responsibilities at CERN

please be patient :)

Selecting the appropriate license

GNU General Public license is recommended CERN-wide (but under review by the OSPO)

Governance scheme will be initially under strict CERN Control

with plans to evolve

Implement OSPO recommendations

OSPO should have finished them in the upcoming months

Challenges

Effort in Planification considering external contributions

QA workflow to adapt ← Current limiting Factor

Migration from gitlab.cern.ch to an open DevOps platform

Managing access, roles, permission

CI/CD runners to adapt

JIRA ticketing system not adapted

…

Perspectives

LGC can be used for a large variety of operations

Full LGC Open-Source will still require some implementation

Planned for 2025

The challenges will be overcome gradually

Perspectives

LGC can be used for a large variety of operations

Full LGC Open-Source will still require some implementation

Planned for 2025

The challenges will be overcome gradually

When could you use it?

Perspectives

LGC can be used for a large variety of operations

Full LGC Open-Source will still require some implementation

Planned for 2025

The challenges will be overcome gradually

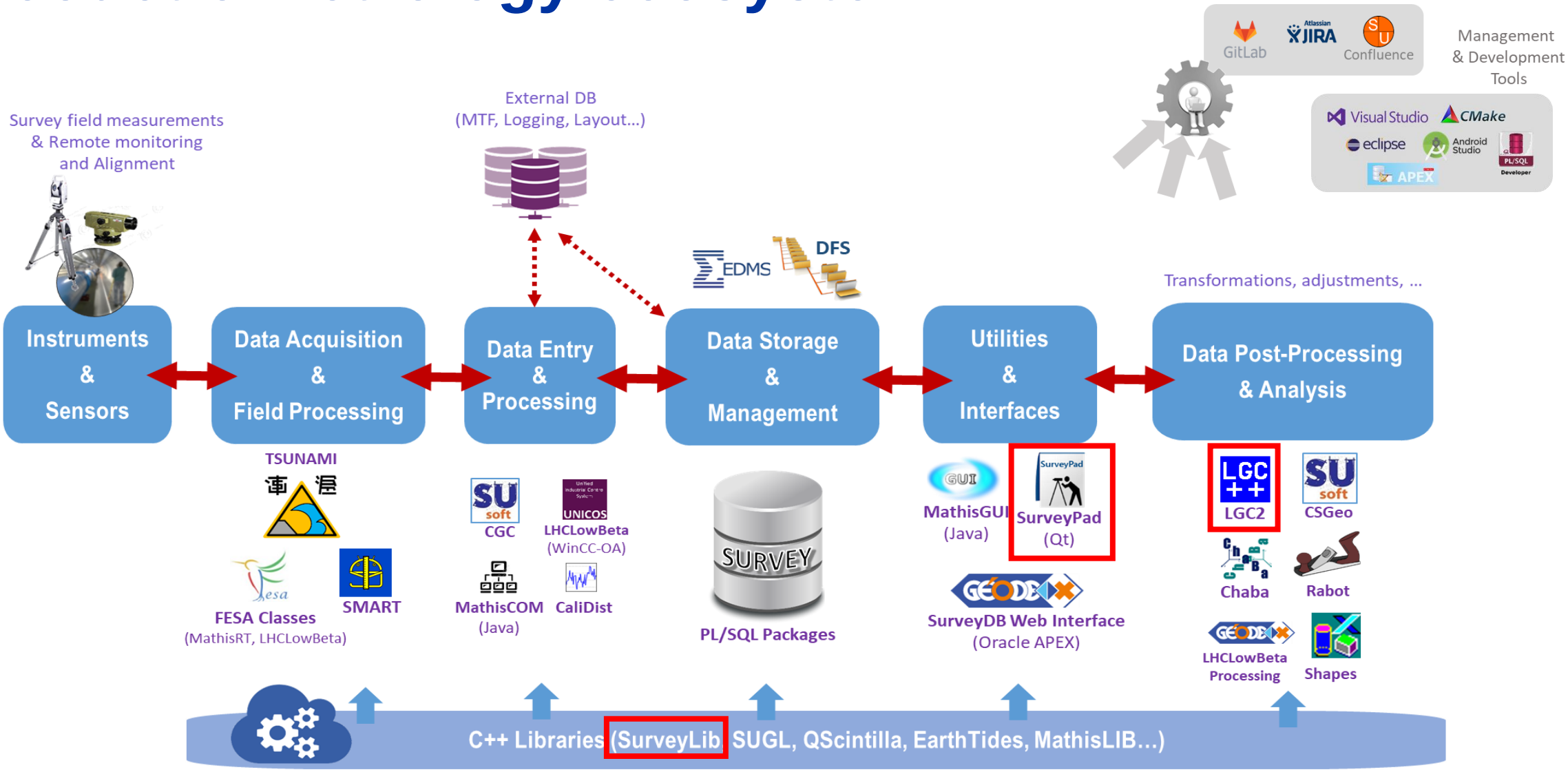
When could you use it?

Recent news from OSPO

LGC, SurveyLib and SurveyPad can already be distributed as binaries

home.cern

Geodetic Metrology ecosystem



LGC : Main design and releases

LGC v2 Main Software and test protocols:

~31’000 lines of C++ code

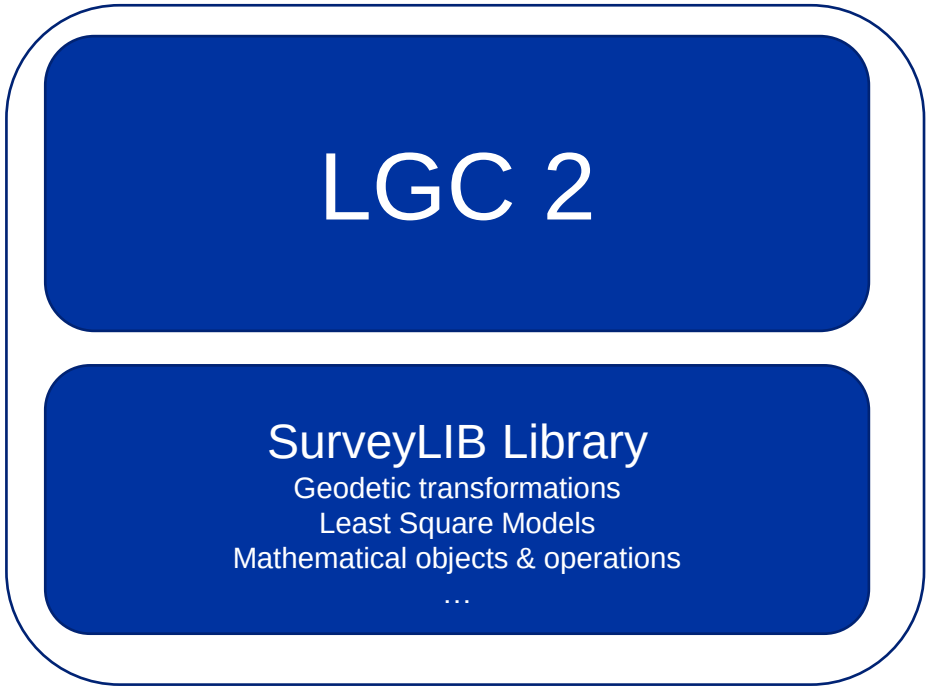
SurveyLib Library

~48’000 lines of C++ code

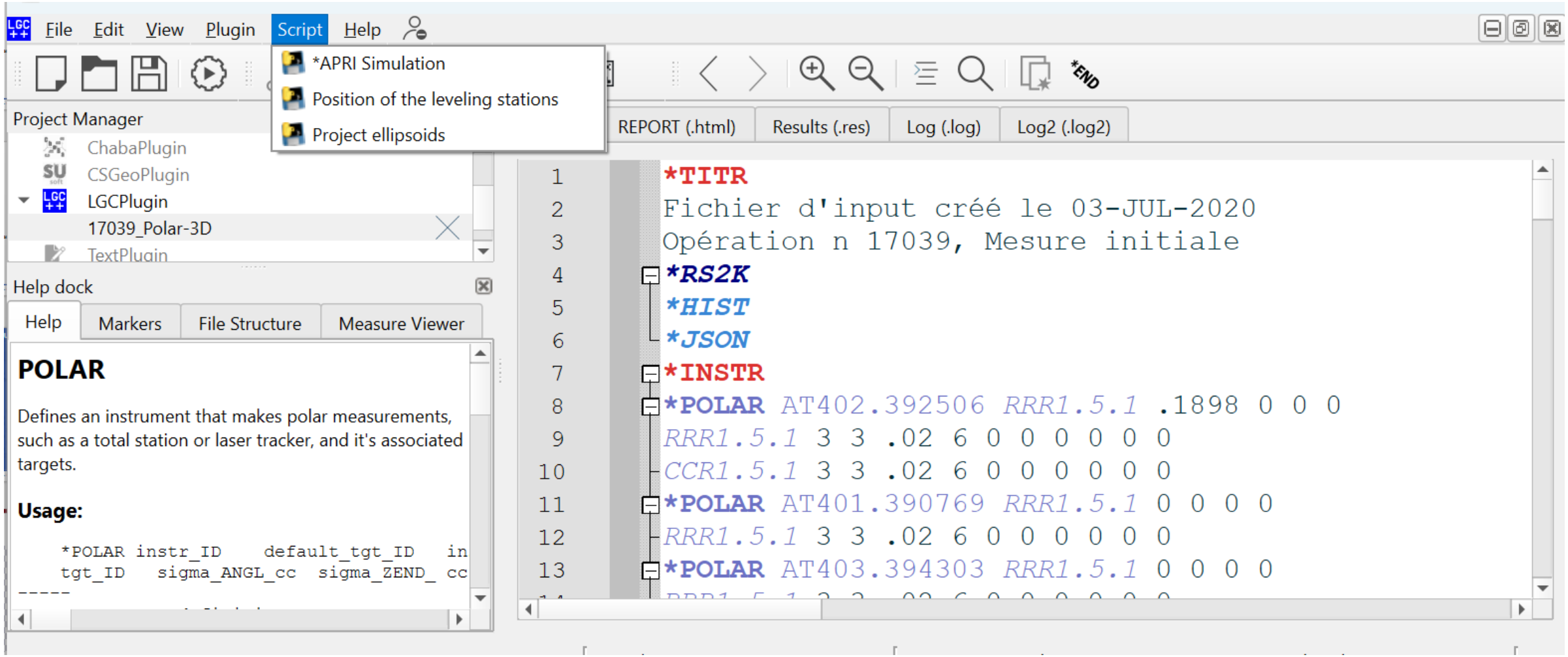
Two major releases per year

Spring and Autumn

Last release V2.7.0 (May 2024)



SurveyPad Interface



Improvements

LGC API/DLL first version in 2023

On-going Python Wrapper dev

Result files available in serialized JSON form since LGC 2.6.0 (May 2023)

```
class Moni
{
public:
        DECLSPEC Moni(std::string inputFilePath);
        DECLSPEC ~Moni();
        //DECLSPEC void Moni::writeJsonFile(TLGCData const *const dat, const std::string &outputFileLocation);
        // write the results from the current estimation in a json file with timestamp

#if USE_SERIALIZER
        DECLSPEC void writeResultFile();
#endif
        // write lgc input file with current measurements with timestamp
        DECLSPEC void writeLGCInputFile();
        // updating observations
        DECLSPEC void setActivationStatus(std::string id, bool status);
        // set activtaionStatus
        DECLSPEC void updateMeas(std::string id, Eigen::VectorXd measurementVector);
        // changing observation Sigma
        DECLSPEC void setObsSigma(std::string id, Eigen::VectorXd sigma);
        // changing the value of a fixed frame parameter
        DECLSPEC void setFixedFrameParameter(std::string frameName, int idx, double val);
        DECLSPEC void setFixedPointParameter(std::string pointName, int idx, double val);
        // "freezing" of free parameters
        DECLSPEC void freezeFrameParameter(std::string frameName, int idx, double val);
        // "unfreezing" of free parameters
        DECLSPEC void unfreezeFrameParameter(std::string frameName, int idx);
        // "freezing" of free parameters
        DECLSPEC void freezePointParameter(std::string pointName, int idx, double val);
        // "unfreezing" of free parameters
        DECLSPEC void unfreezePointParameter(std::string pointName, int idx);

        // triggering the adjustment calculation
        DECLSPEC bool adjust();
        // for checking the estimation status
        DECLSPEC bool getStatus();
        // get estimate of point
        DECLSPEC Eigen::VectorXd getPointEstimate(std::string);
        // get estimate of point in subframe
        DECLSPEC Eigen::VectorXd getPointEstimate(std::string, std::string);
        // get estimate of frame
        DECLSPEC Eigen::VectorXd getFrameEstimate(std::string);
        DECLSPEC Eigen::VectorXd getFrameEstimatePrec(std::string);
        // get diagonal elements of covariances of the estimated parameters
        DECLSPEC Eigen::VectorXd getPointEstimatePrec(std::string);
```

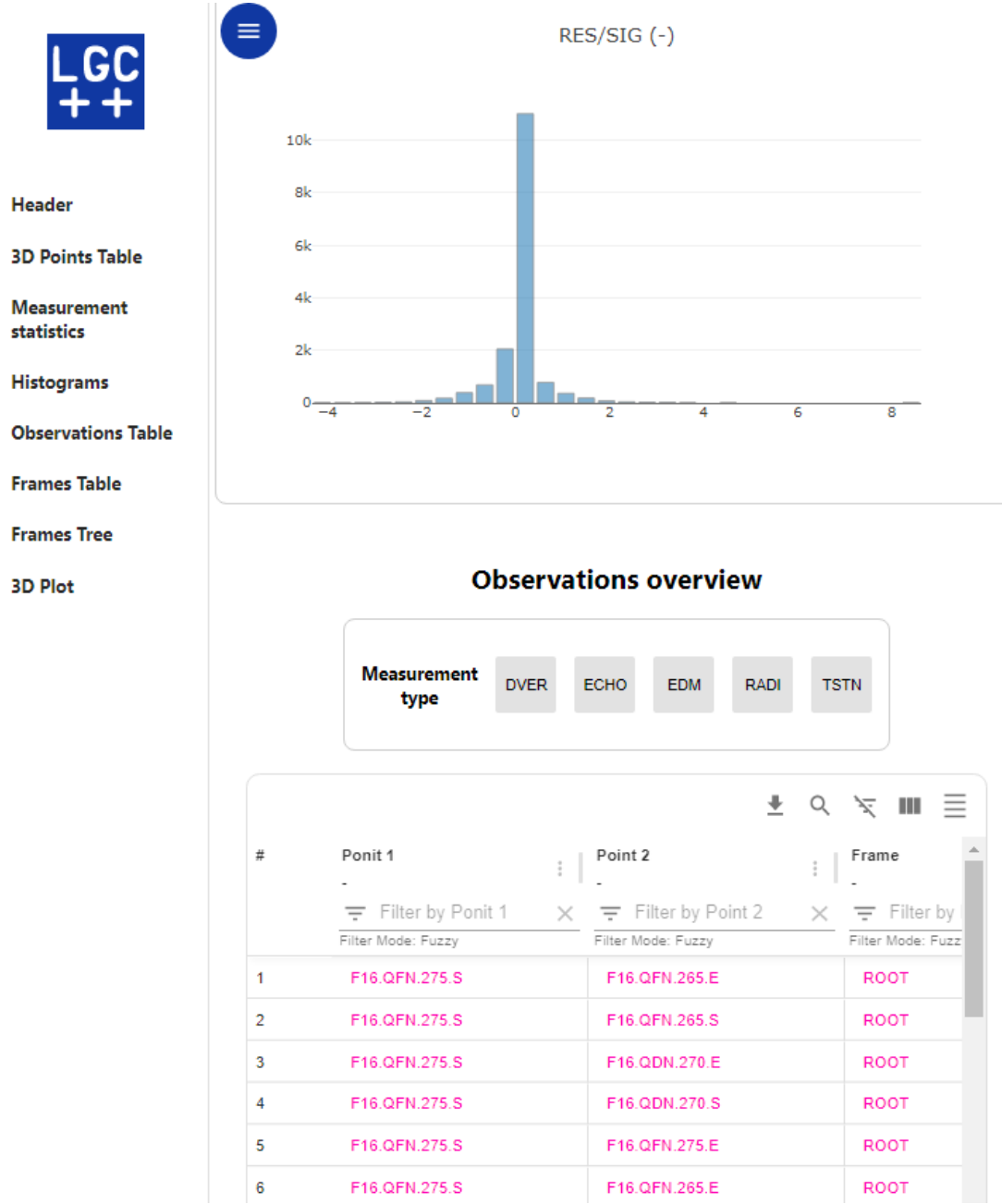
Improvements

LGC API/DLL first version in 2023

On-going Python Wrapper dev

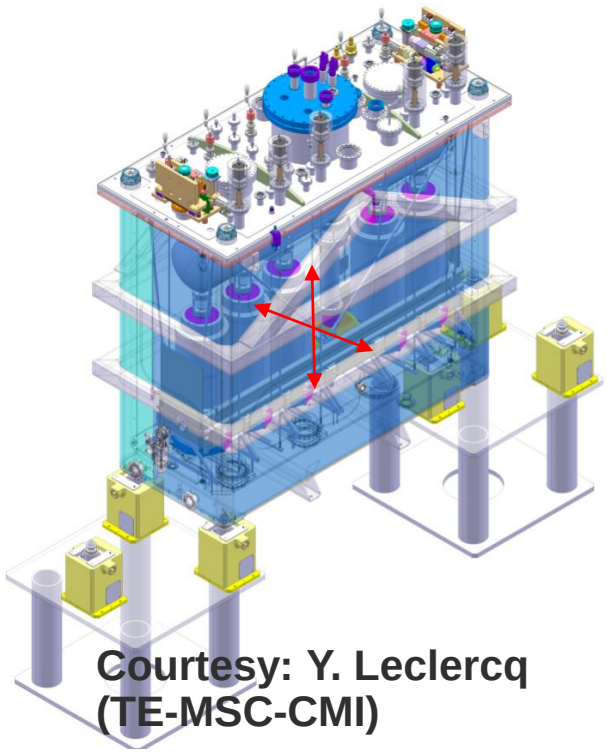
Result files available in serialized JSON form since LGC 2.6.0 (May 2023)

- Better integration in other processes
  ex: “real-time” computation for FRAS monitoring
- Easier development of postprocessing tools (no need to touch the LGC core)
  Allows external partners to exploit data (students, etc..)
  ex: HTML report for the result file

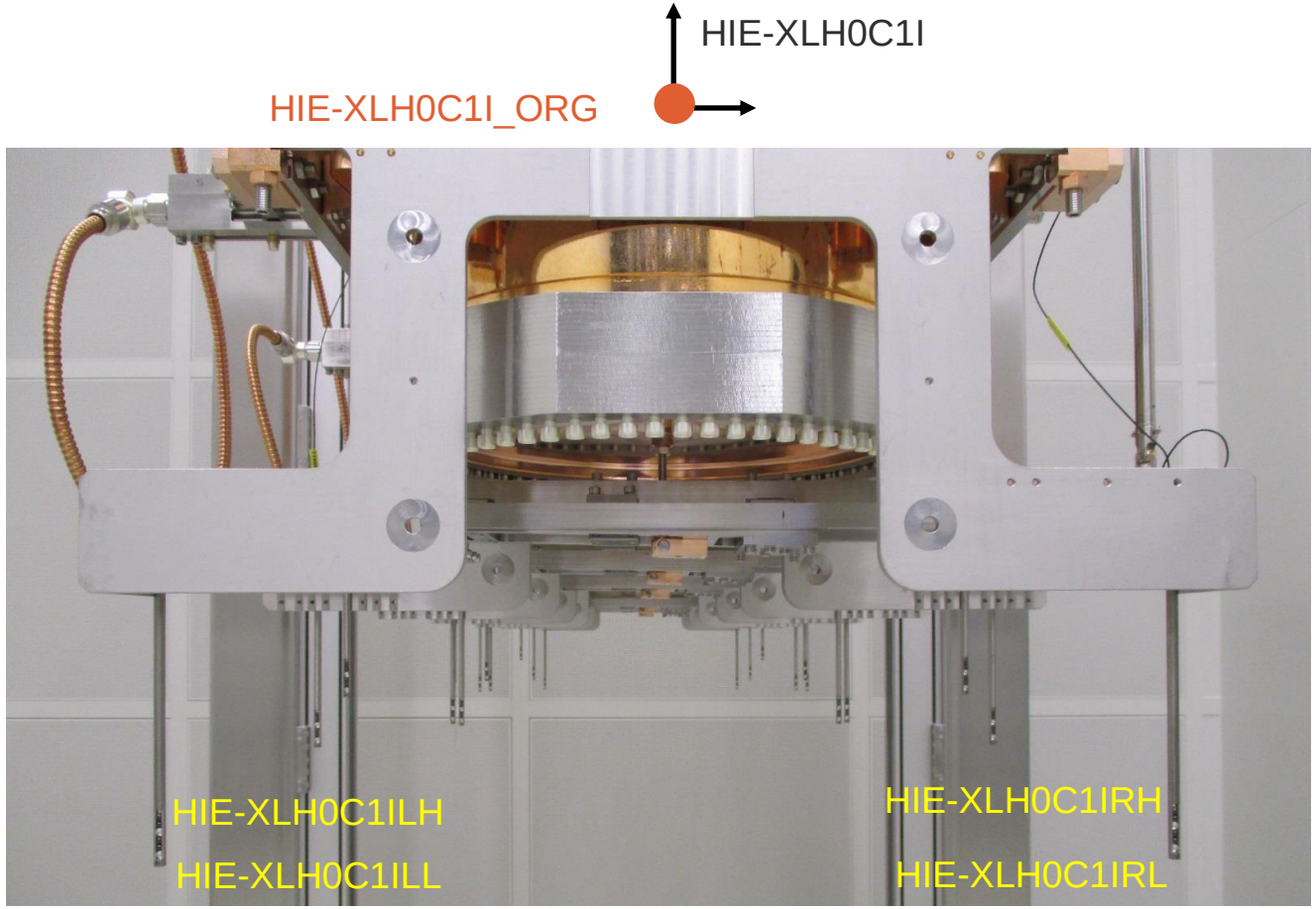


Monitoring

- Link sensors to elements or common support
- Move object in bulk abstracted by points / frames



Courtesy: Y. Leclercq (TE-MSC-CMI)




```
*FRAME   HIE-XLH0       0.001444      17.259302      -0.000526    399.9958041      0.0095356   399.947185  1
*FRAME HIE-XLH0C1I -0.000839 -1.064527 -0.001362671 399.5184 0.028299 0.1129 1 TX TY TZ RY
*CALA
HIE-XLH0C1I_ORG       0.000000       0.000000       0.000000
HIE-XLH0C1ILH -0.2301135 0.050087 -0.4061526
HIE-XLH0C1ILL -0.2301045 0.050062 -0.4181359
HIE-XLH0C1IRH 0.2305565 0.04981 -0.4061656
HIE-XLH0C1IRL 0.2305645 0.049816 -0.4181489
*ENDFRAME
```

Monitoring

- Link sensors to elements or common support
- Move object in bulk abstracted by points / frame
- Versatile (no hard coded parameters) that can accommodate a lot of configuration

FRAS and the HIE ISOLDE Cryomodule monitoring project use the same version of LGC.

Some figures from the HIE ISOLDE Cryomodule monitoring project:

- ~90 frames necessary in total for 4 cryomodules
- Stages installation that required no update of LGC
- Objects to monitor are under cryogenic conditions
- Some instruments linked together by frames
- Allowed to make a self calibration of the system in the least square process

What LGC can bring?

- LGC lifecycle management policy: Long-term support with stable release maintained for CERN usage
- Mixing different types of observation in one computation block
- Not a black box, everything is exposed and documented
- Interoperability: easy integration to processes and interfaces thanks to data serialization and LGC API
- Can accommodate simple and complex networks with advanced features (FRAME concept, monitoring system, etc…)
- Mutual aids between institutes: speaking the same language + the more users and use cases, the better
- Extensive Tests and Training possible by the CERN Developers / Users