VISUALISATION OF TILECAL BY WEBGL API

SHARMAZANASHVILI Alexander
Georgian Technical University

PATARIDZE Lasha
KHELASHVILI Levan
UDZILAURI Nikoloz
KOBAKHIDZE Shota
Georgian Technical University

About the Project

- This project is inside the collaboration agreement between ATLAS and Georgian Technical University, Tbilisi, Georgia – AA366/10-add5-2019, Working Package #04

- We are working together with ATLAS TileCal team – contact person Alexander SOLODKOV

- All manpower involved in this project is funded by the Georgian Technical University

WP4: **"Development of Interactive Detector Display Software Application for Visualisation and Maintenance of Detector Subsystems"**

Interactive Detector Display (IDD) is web-based internet application for interactive visualization of structure and detailed content of detector subsystems and physical events carrying out on that.

Application is hardware/software platform independent and requires no installation for running. Generic version of application with basic functionalities and generic geometries is already developed and available from here: http://tracer.web.cern.ch

On the second phase of development customization of generic version on Tile Calorimeter requirements will be done. Short todo list includes visualization of – cells; energy deposits in cells for reconstructed ATLAS events with various options of filtering; Tile Calorimeter components; services in particular regions.

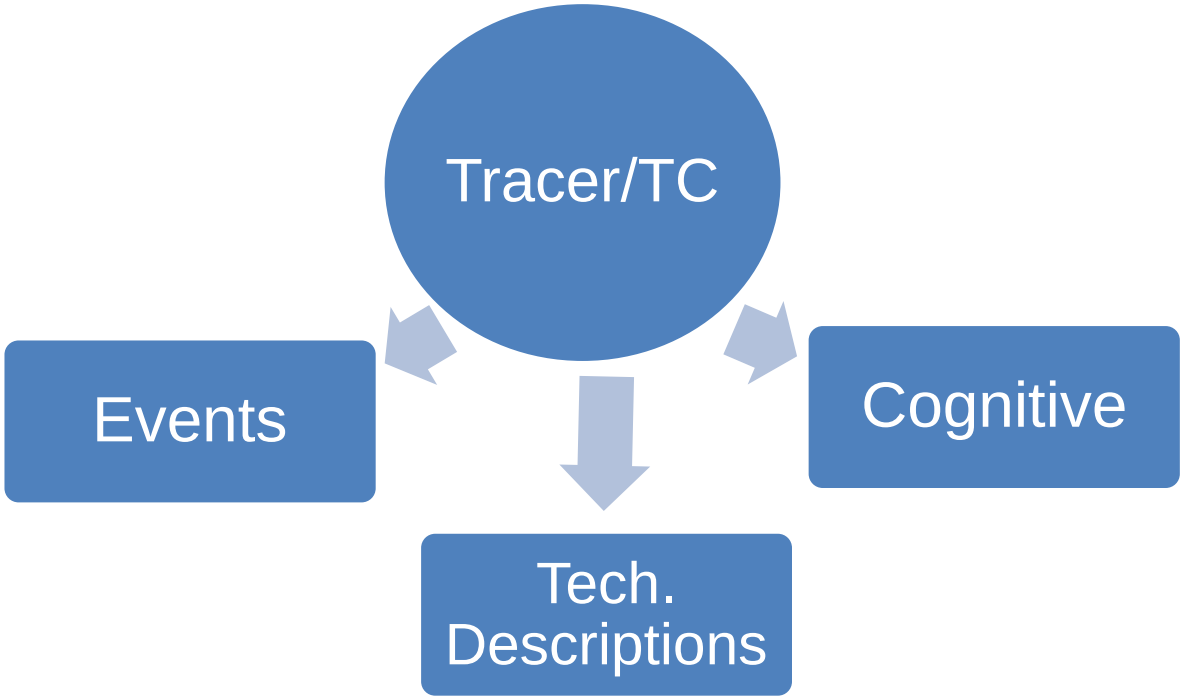
Deliverables: Javascript applications and functions

Contact person from ATLAS: **Alexander SOLODKOV**

Manpower from GTU: 2FTE/Javascript programmer

About Tracer/TC

- We call Api Tracer/TC and it will be an Interactive Detector Display software application especially developed for Tile Calorimeter
- Main functionality will cover 3 directions:



Concept

- Application will be platform independent and workable on all types of desktop PC's, notebooks and tablets/phones
- It will be compatible with Windows, Mac-OS, Linux and Android
- Will requires no Installation just Click-And-Go and will be accessible in Browser
- All these benefits are coming from WebGL/Three.js we are going to use as an development platform

- However as gaming engine WebGL has its limitations, especially in performance and visualization of complex geometries
- So our goal is to find agreement between system requirements and limitations of platform

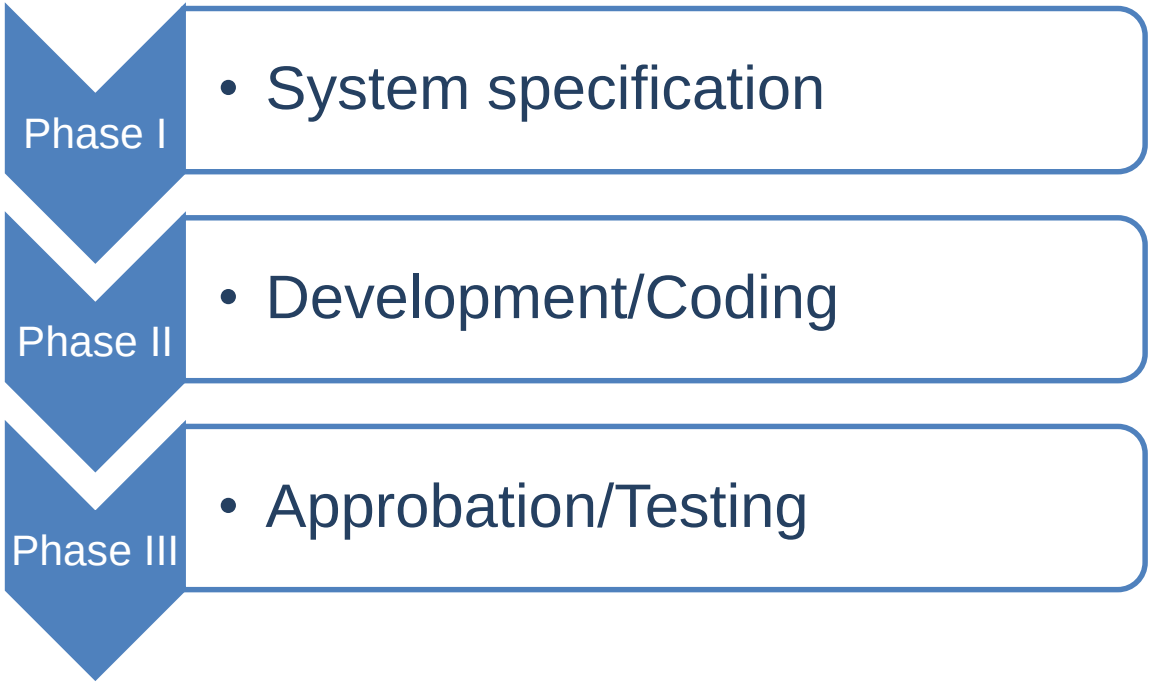
Concept

- WebGL *advantages* we are looking for:
  - Well developed visualization scenes
  - Realistic and high performed rendering
  - Open and fast-growing platform
  - Built-in libraries for browsers

- WebGL *disadvantages* for us:
  - Poor performance for complex geometries as we have in ATLAS detector
  - Low performance for exported non-native scenes
  - Poor geometry cuts for non-native scenes
- Unique ways should be found for the solutions

Development Life Cycle

- We foresee 3 general phases of development



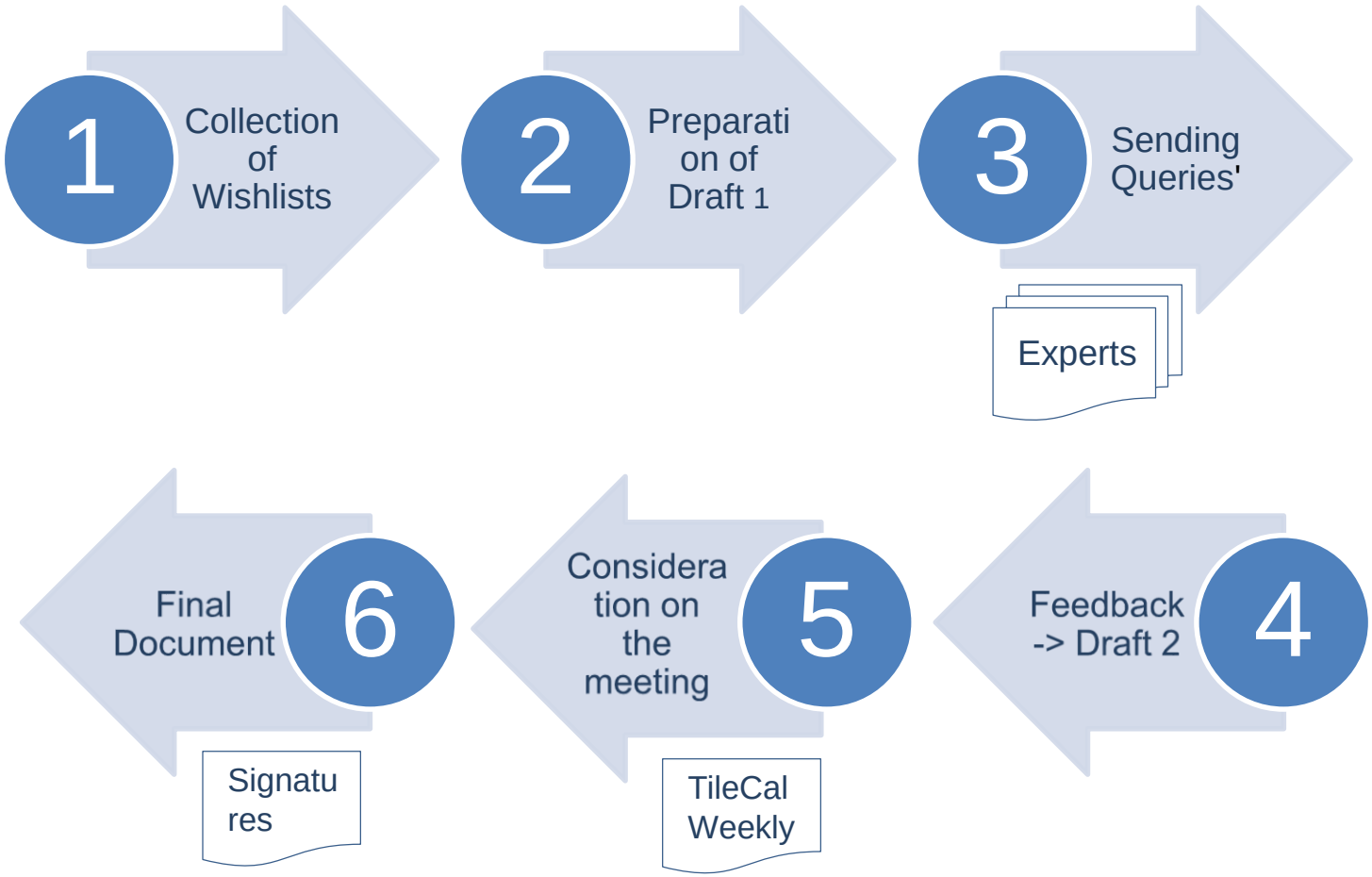
I. System Specification

- Currently we are doing paper work (everybody hate it but this is most important part for success)

- Thanks to TileCal experts – Oleg, Sanya, Pawel, Irakli, Henric, Yosef to contribute in this process
- We have very successful visit here at CERN of Lasha PATARIDZE, member of our group, thanks to TileCal collaboration to make it possible
- We have collected 9 packages with wish lists and become quite good experts of TileCal ☺

I. System Specification

- Steps to be done in this direction:



I. System Specification

- For the moment we are close to generate Draft 1
- We have squeezed 9 wish lists from Oleg, Sanya, Pawel, Irakli, Henric, Yosef, etc. and cast system specifications
- We separate 4 categories:
  1. General requirements
  2. Interactive visualization of events
  3. Cognitive requirements
  4. Technical Description specifications
- For events we have 3 subgroups:
  1. Cell visualization requirements
  2. Tracks visualization requirements
  3. Jets visualization requirements

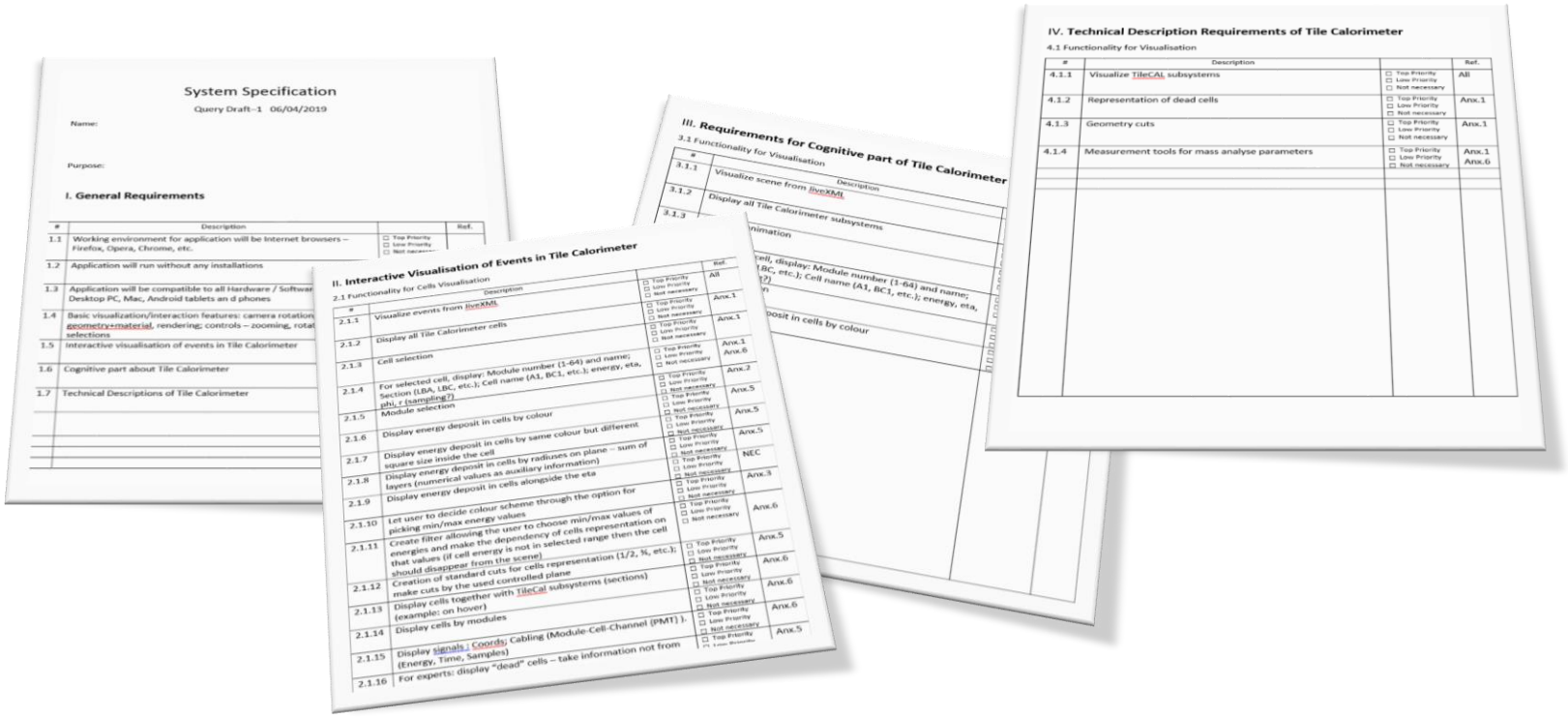
II. Interactive Visualisation of Events in Tile Calorimeter

2.1 Functionality for Cells Visualisation

#	Description		Ref.
2.1.1	Visualize events from JiveXML	☐ Top Priority ☐ Low Priority ☐ Not necessary	All
2.1.2	Display all Tile Calorimeter cells	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.1
2.1.3	Cell selection	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.1
2.1.4	For selected cell, display: Module number (1-64) and name; Section (LBA, LBC, etc.); Cell name (A1, BC1, etc.); energy, eta, phi, r (sampling?)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.1 Anx.6
2.1.5	Module selection	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.2
2.1.6	Display energy deposit in cells by colour	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.5
2.1.7	Display energy deposit in cells by same colour but different square size inside the cell	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.5
2.1.8	Display energy deposit in cells by radiuses on plane – sum of layers (numerical values as auxiliary information)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.5
2.1.9	Display energy deposit in cells alongside the eta	☐ Top Priority ☐ Low Priority ☐ Not necessary	NEC
2.1.10	Let user to decide colour scheme through the option for picking min/max energy values	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.3
2.1.11	Create filter allowing the user to choose min/max values of energies and make the dependency of cells representation on that values (if cell energy is not in selected range then the cell should disappear from the scene)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.6
2.1.12	Creation of standard cuts for cells representation (1/2, ¼, etc.); make cuts by the used controlled plane	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.5
2.1.13	Display cells together with TileCal subsystems (sections) (example: on hover)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.6
2.1.14	Display cells by modules	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.6
2.1.15	Display signals : Coords; Cabling (Module-Cell-Channel {PMT}). (Energy, Time, Samples)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.6
2.1.16	For experts: display "dead" cells – take information not from JiveXML but from other sources (?)	☐ Top Priority ☐ Low Priority ☐ Not necessary	Anx.5
2.1.17	Display what : Energy, Time or Energy+Time	☐ Top Priority ☐ Low Priority	Anx.5

I. System Specification

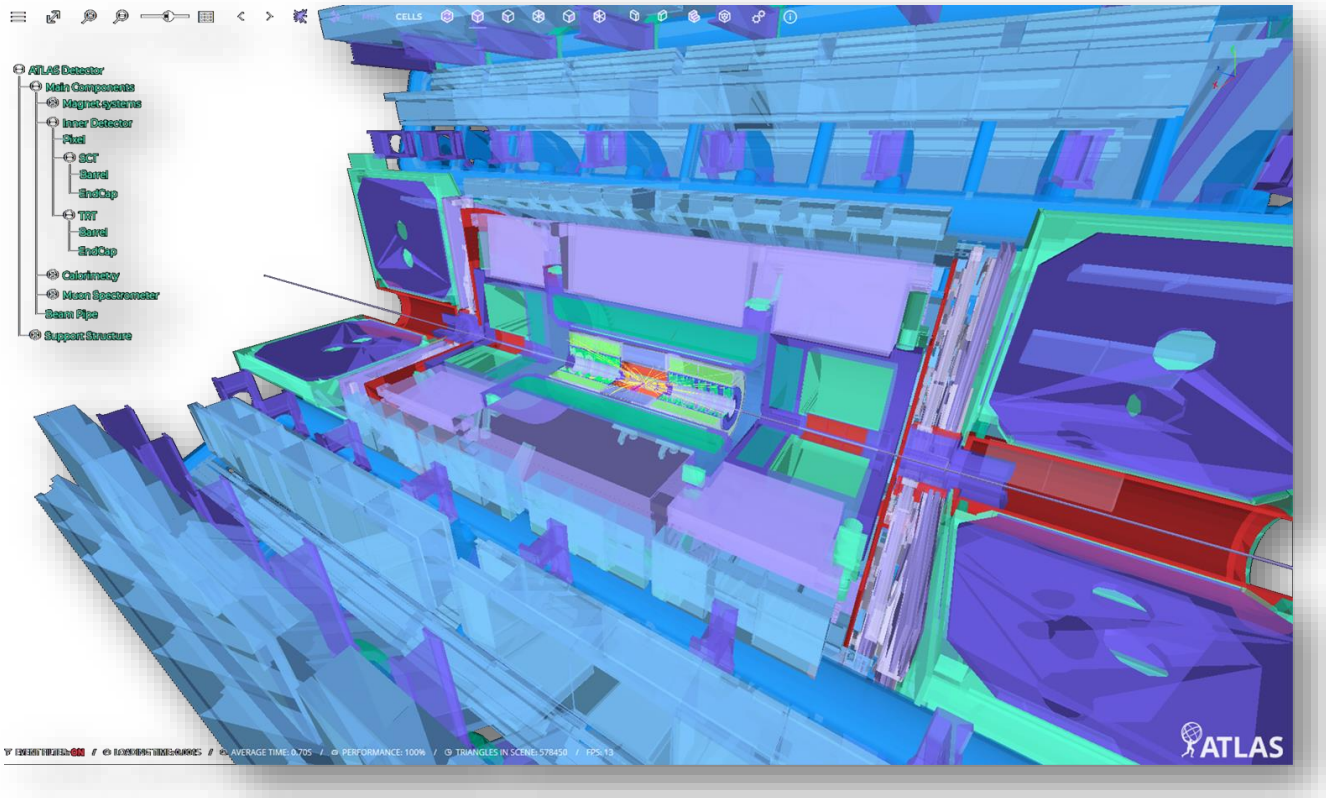
- For the moment we have number of requirements as follow: General requirements (7), Events visualization (34), cognitive requirements (6) and technical specifications (4)



II. Development/Coding

- We bit pushing development steps as well
- System core engine on the base of WebGL almost ready

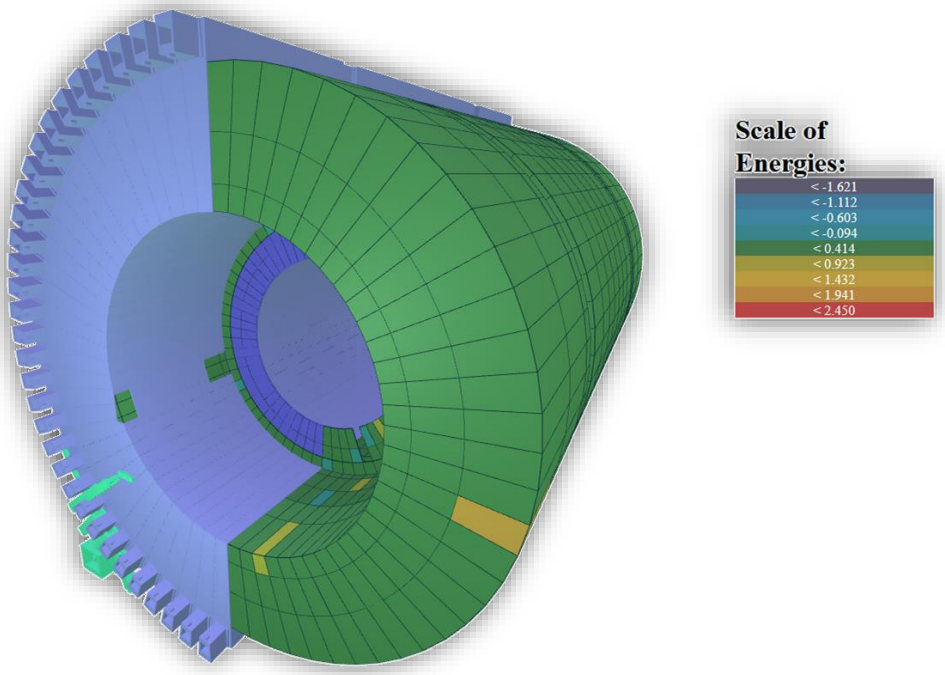
http://cadcamge.ch/at/r3.2
* This is development folder with bugs



II. Development/Coding

- We did generic system prototype

http://cadcamge.ch/at/TileCal
* This is development folder with bugs



- We are revising all our algorithms for Track/Jet visualization from JiveXML

Upcoming Steps:

1. Finish Draft-1 and sending queries to experts (next week)
2. Collecting Feedbacks and produce Draft-2 (end of June, 2019)
3. Preparation of final document of system specification (July, 2019)
4. Preparation of business plan for development (July, 2019)
5. Starting of Development/Coding (September, 2019)

Thanks for attention,

Comments are welcome

Lasha.sharmazanashvili@cern.ch