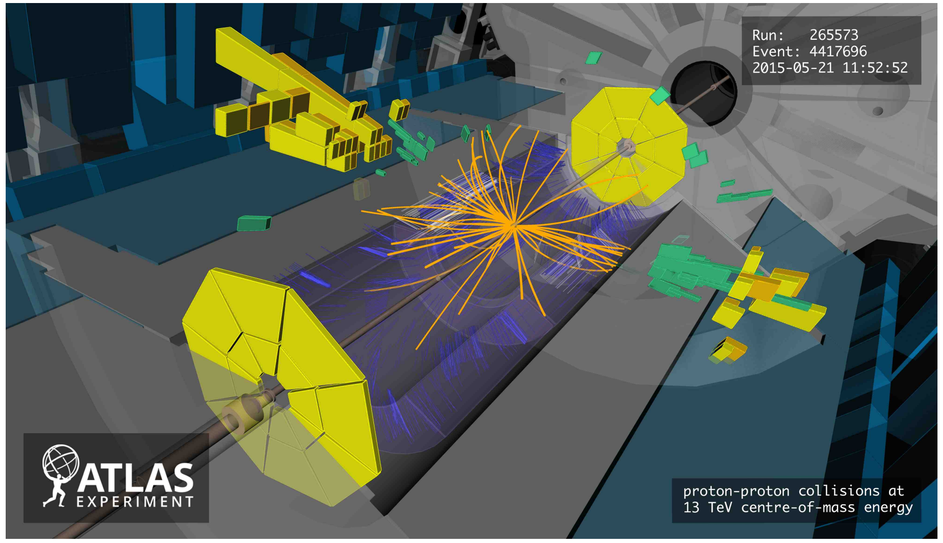
3D event display in ATLAS: VP1

Riccardo Maria BIANCHI (Pittsburgh)
HSF Visualization Workshop - 28 Mar 2017

“VP1” (ATLAS)



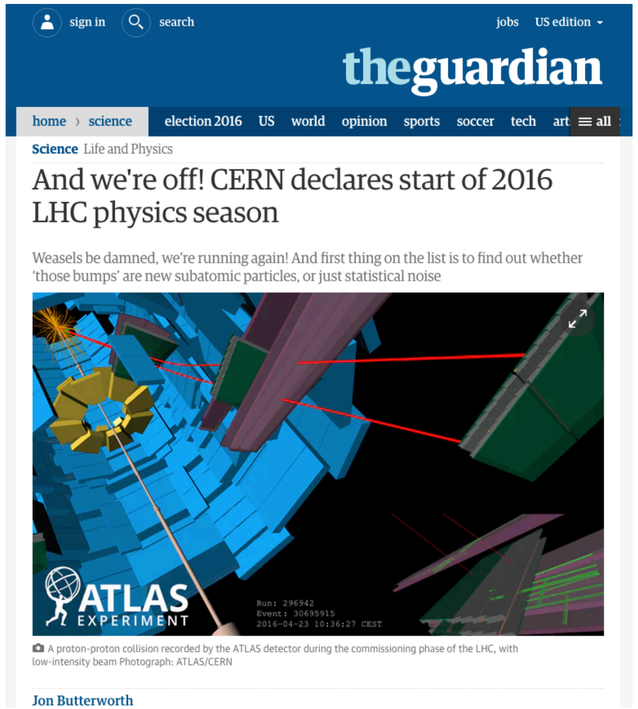
VP1 is a **3D event display** for ATLAS

VP1 is **part of the experiment’s framework**

it is a **general-purpose** tool used in ATLAS for:

- physics analysis,
- detector development,
- reconstruction and simulation checking and debugging,
- outreach, press releases, ...

https://atlas-vp1.web.cern.ch/

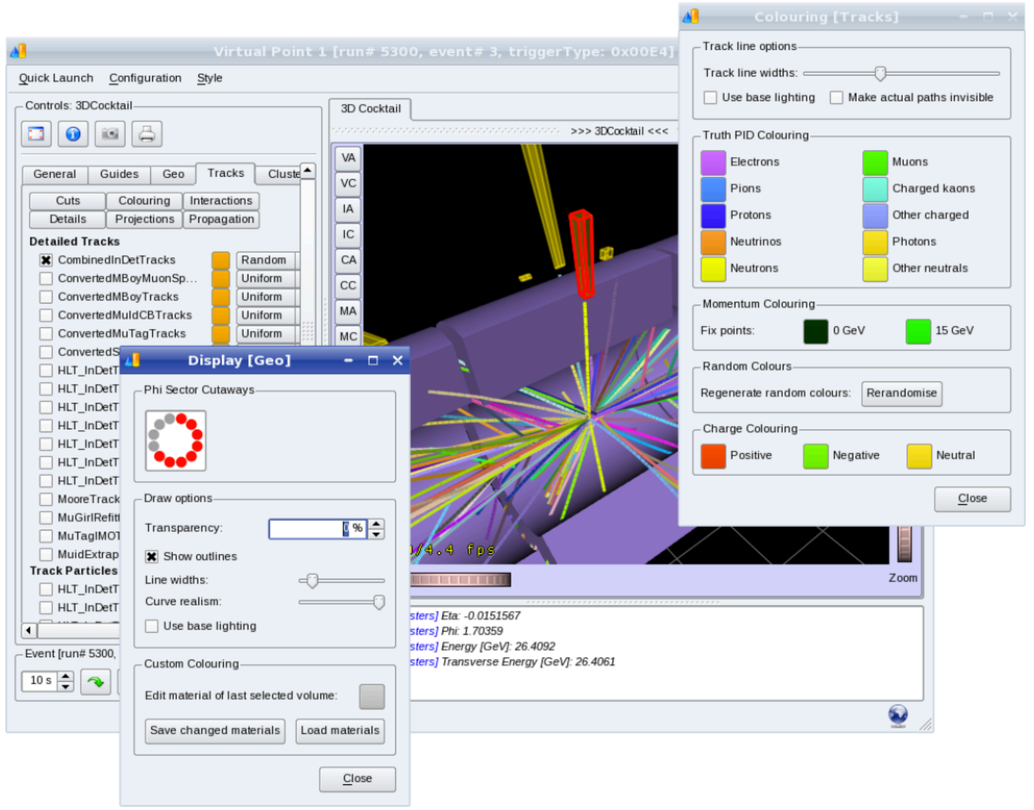


https://atlas-vp1.web.cern.ch/

Technology

Technologies:

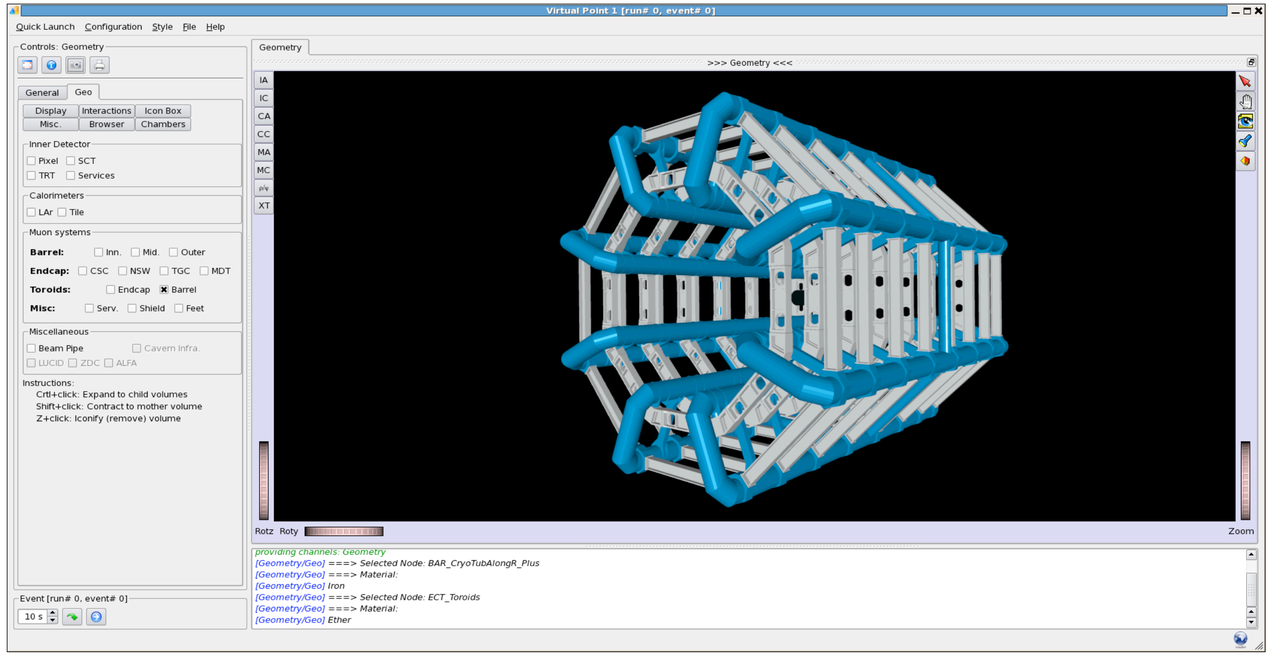
- **C++**
- graphics engine:
  Coin3D (OpenGL) + **SoQt**
- GUI:
  Qt



https://atlas-vp1.web.cern.ch/

Geometry

- VP1 takes the ATLAS geometry from the ATLAS online geometry DB
- it shows the actual ATLAS geometry

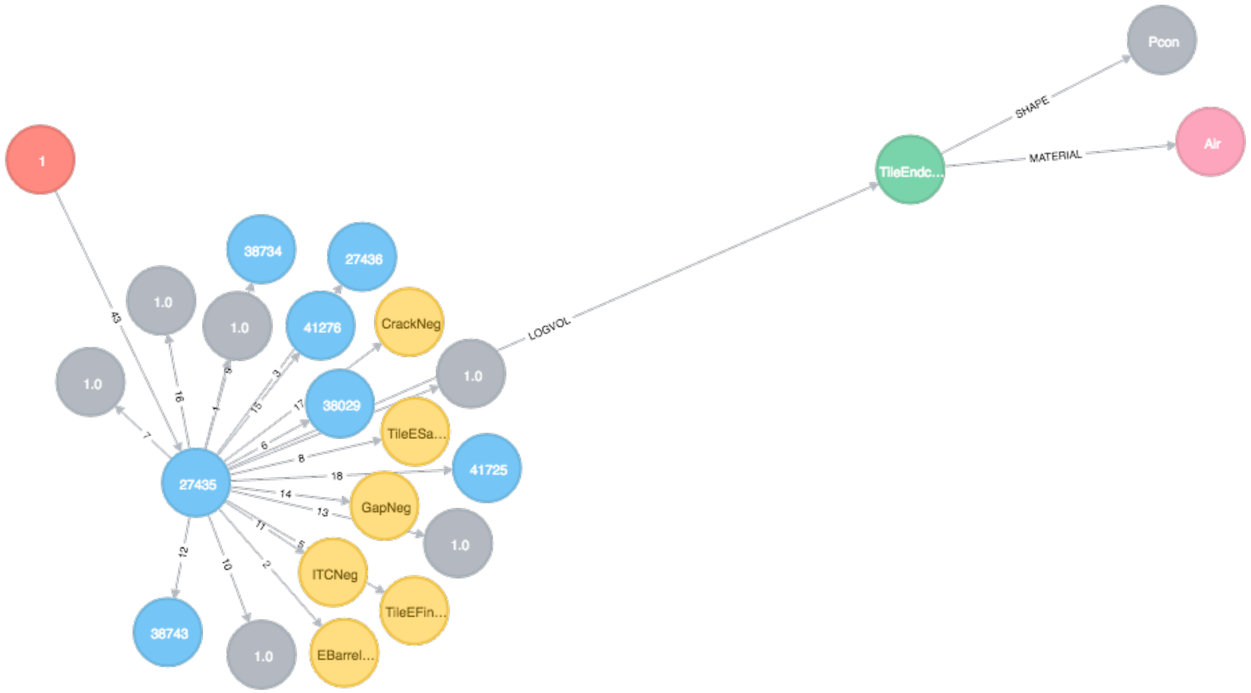


https://atlas-vp1.web.cern.ch/

Geometry

- VP1 takes the ATLAS geometry from the ATLAS online geometry DB
- it shows the actual ATLAS geometry

- **recently**, a new mechanism to **persistify the Geometry** in SQLIte files has been developed *(Joe, Ric)* and presented at **CHEP 2016**
- Now, we are developing *(Ilija, Ric)* a mechanism to **serve the geometry** both from a **REST API** and a **Neo4j server**. It will be submitted to ACAT 2017



https://atlas-vp1.web.cern.ch/

VP1 Pros

Pros:

Architecture:

- Being integrated in the experiment framework, VP1 can **access all ATLAS data**

Graphics libraries:

- Coin3D is "***scene-graph***" based: easy to match tree-like GeoModel objects to Coin primitives

https://atlas-vp1.web.cern.ch/

VP1 Cons

Cons:

Architecture:

- VP1 needs the whole experiment framework to work. It's **not cross-platform**
- Very **slow** when **run remotely** through SSH because of 3D data being sent through the network.
- It is **widely used by experts**, but **less by physics analysis end users**, because most of them they do not know how to run the ATLAS framework and they only run the analysis on ROOT ntuples on their laptops

Graphics libraries:

- Coin3D and SoQt are **not supported and maintained anymore**. Also, they are very old and sometimes they show issues with modern compilers
- Coin3D has **known issues** with transparency and it does not exploit modern graphics API

https://atlas-vp1.web.cern.ch/

What we would like

Technologies:

- An event display tool should be **easy to install** and **use** by (analysis) end-users
- Ideally, it should be the **same tool** for detector/experiment data (through the framework) and analysis objects (official *ntuples*)
- It should be built on top of **modern** and, most of all, **maintained libraries**
- The libraries should be easily integrated/interfaced in/to existing code
- If "game engines", they should provide tools to **use external libraries** as well: for example, currently ROOT is needed to open the data files

Event data picking:

- An **easy way to get events**, in **different formats**, ready to be read in the visualization tool (an "event service" is under development, in ATLAS).
- **Event picking** should be easy for users… ***à la* Google Maps!** *(as Joe would say!)*