Going standalone and platform-independent, an example from recent work on the ATLAS Detector Description and interactive data visualization

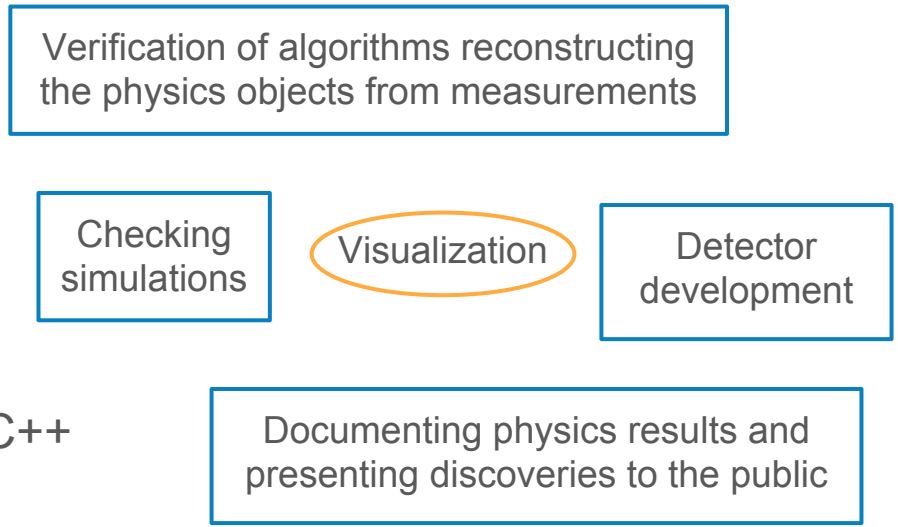
Sebastian Andreas Merkt (Pittsburgh),
Riccardo Maria Bianchi (Pittsburgh), Joseph Boudreau (Pittsburgh), Paul Gessinger (Mainz),
Edward Moyse (Massachusetts), Andreas Salzburger (CERN), Vakho Tsulaia (LBNL)

on behalf of the ATLAS Collaboration

CHEP 2018, Sofia, 9-13 July 2018

Describing and visualizing the experiment's geometry

In ATLAS we need accurate, detailed, and interactive **visualization** of the detector geometry for a number of tasks:



The detector is described by shapes and classes from **GeoModel**, a C++ library. The geometry is built on-the-fly from the C++ code upon request.

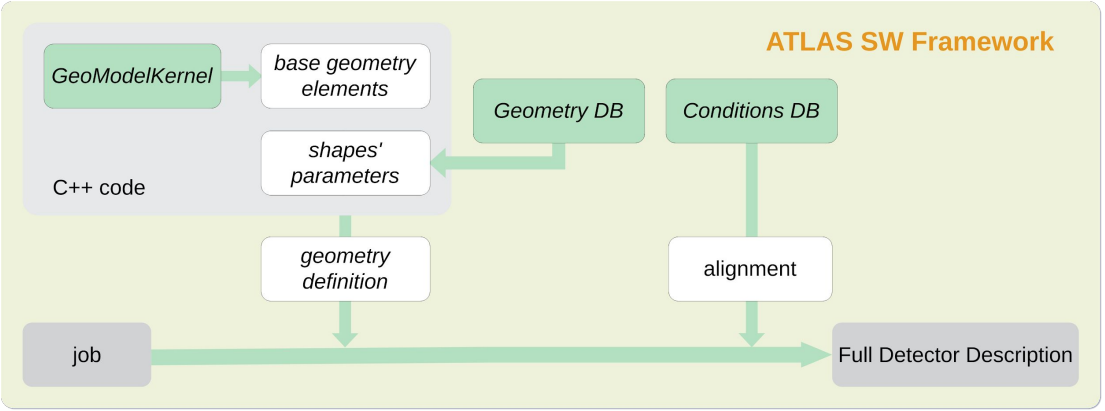
For visualizing the geometry, **VP1** [1,2] is a used, a general-purpose 3D visualization tool. Its integration in the experiment's software framework makes it possible to access all possible data from the experiment, but also limits its usage.

Current implementation of the ATLAS Detector Description

GeoModel [3] and Geometry DB [4] have served ATLAS well for ~15 years.

They have many **advantages**:

- Optimization techniques to describe complicated geometries
- Minimal memory consumption
- Mechanisms for applying alignments on top of regular geometry



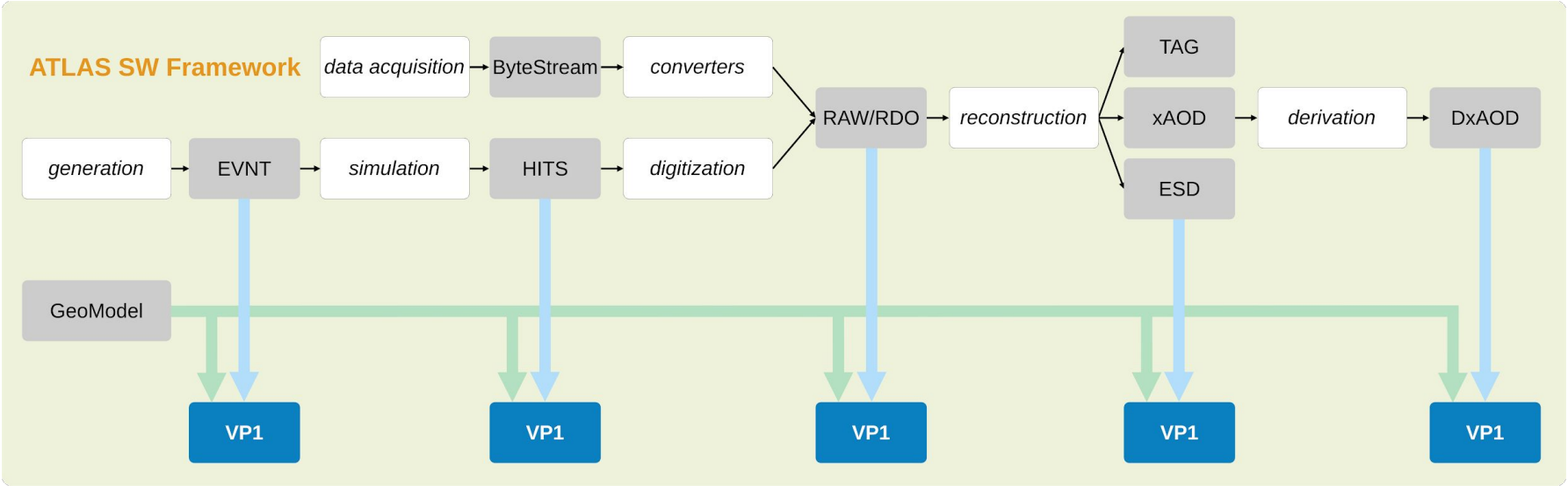
However, the current implementation has a few **drawbacks**:

- The need for the full framework makes the development of standalone applications impossible and currently restricts them to SLC6.
- Geometry is built on-the-fly only, making it impossible to have a persistent copy to store and share the geometry with other applications

(See CHEP talk on GeoModel+ConditionsDB: https://indico.cern.ch/event/587955/contributions/2936850/)

Accessing and visualizing experimental data

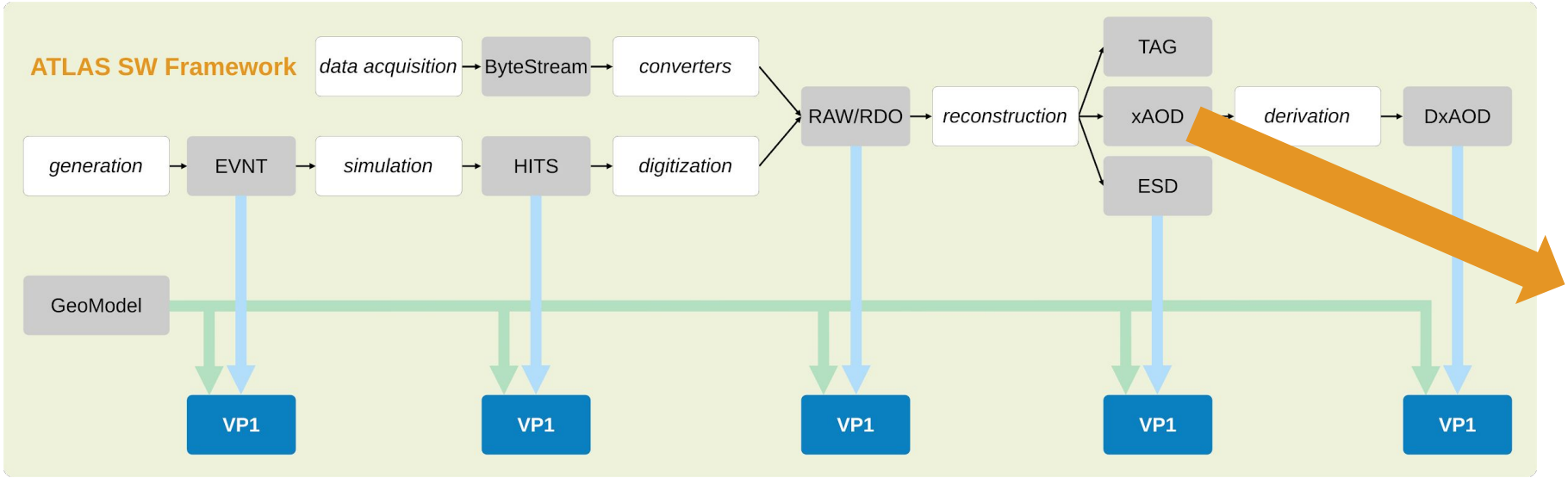
VP1 can access all experimental data, at every step of the data chain. It also reads the GeoModel-based detector description to render the ATLAS geometry.



Accessing and visualizing experimental data

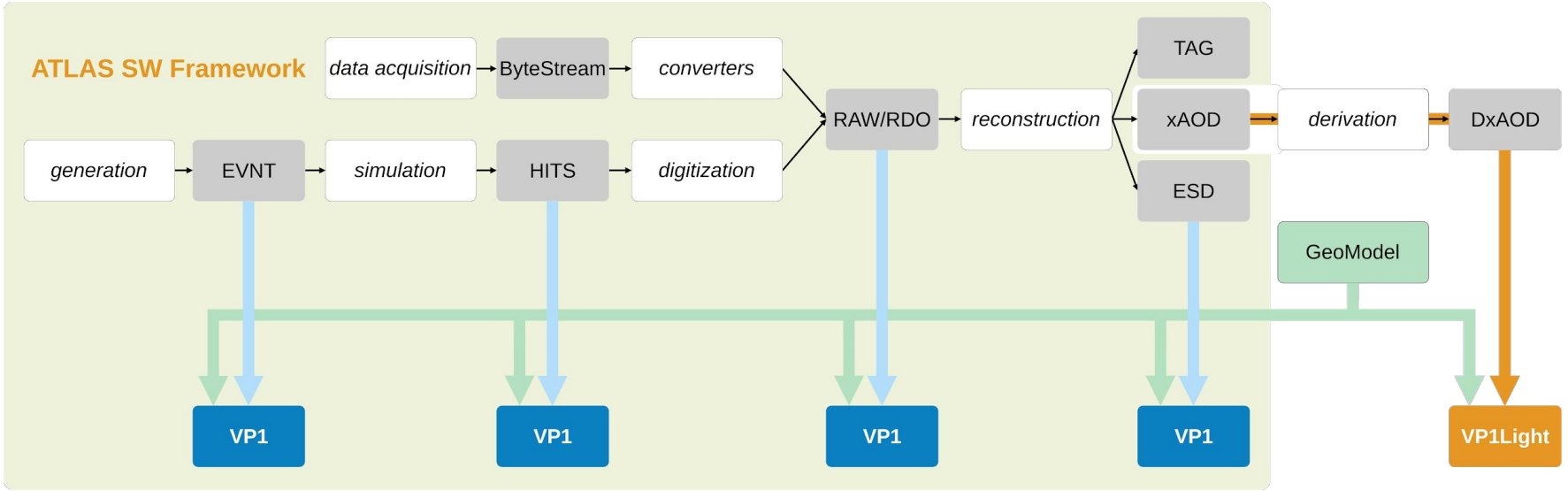
xAOD files store filtered and calibrated data, “good for physics”. In principle, they can be accessed without running Athena, but we still need the full ATLAS framework if we want to open them in VP1.

Another motivation for our **going standalone**



Going standalone: From VP1 to VP1Light

Standalone GeoModel and VP1Light can access the xAOD files without the need for the experiment's software framework.



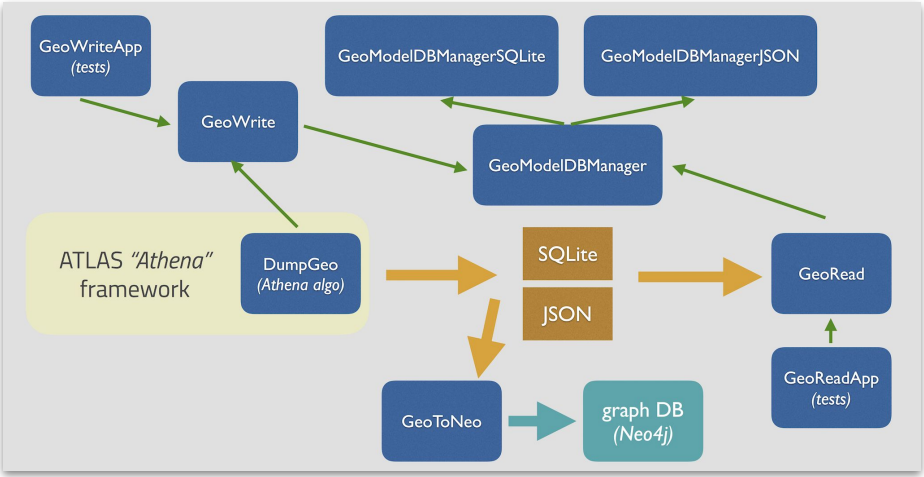
Standalone GeoModel

- GeoModelKernel and the other core packages have been moved outside of Athena. **GeoModel is now a standalone package.**
- Removed most of the dependencies and it now **depends only on Eigen.** (dropped CLHEP)
- **Alignment** is used for the accurate ATLAS geometry, needed for Simulation and Reconstruction. But it is not needed for analysis end-users or for detector development. At the moment alignment data is not available outside the software framework.
- A **persistent copy of the Detector Description** is needed when building the ATLAS geometry outside of the framework (no access to the Geometry DB).

GeoModel persistification - SQLite & JSON

We are now able to persistify the GeoModel description and the parameters stored in the Geometry DB (“geometry tag”) in a single file, in two formats:

- **SQLite**, compact, for minimal storage size, easy to send (see also [5])
- **JSON**, human-readable, easy to explore and combine



Geometry SQLite data format

The **SQLite** data model has been designed for **compactness**:

- Only the GeoModel objects constructor's parameters are stored.
- Objects built upon other objects refer to their IDs.
- Parent-child relationships between objects are stored in an auxiliary table.
- Shared objects are stored once.

Full ATLAS geometry in SQLite is only **~50 Mb** on disk.

- However, the SQLite format is not suitable for exploring and debugging.
- It is also difficult to split or merge among multiple files.

```
$ sqlite3 geometry_simplestToyDetectorFactory.db

SQLite version 3.13.0 2016-05-18 10:57:30
Enter ".help" for usage hints.
sqlite> .headers ON
sqlite> .tables
AlignableTransforms  LogVols              SerialDenominators
ChildrenPositions    Materials            SerialTransformers
FullPhysVols         NameTags             Shapes
Functions            PhysVols             Transforms
GeoNodesTypes        RootVolume           dbversion


sqlite> select * from GeoNodesTypes;
id|nodeType|tableName
1|GeoPhysVol|PhysVols
2|GeoFullPhysVol|FullPhysVols
3|GeoLogVol|LogVols
4|GeoMaterial|Materials
5|GeoShape|Shapes
6|GeoSerialDenominator|SerialDenominators
7|Function|Functions
8|GeoSerialTransformer|SerialTransformers
9|GeoTransform|Transforms
10|GeoAlignableTransform|AlignableTransforms
11|GeoNameTag|NameTags


sqlite> select * from RootVolume;
1|1|1

sqlite> select * from PhysVols;
id|logvol|parent
1|1|NULL
2|2|1
3|3|2
4|4|3
```

List of *GeoPhysVol* nodes

List of *GeoShape* nodes

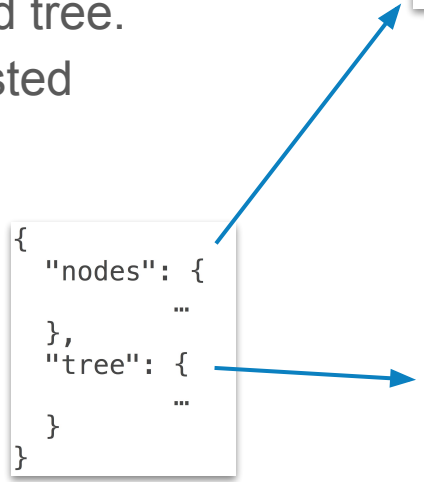
```
sqlite> select * from LogVols;
id|name|shape|material
1|WorldLog|1|1
2|ToyLog|2|2
3|Passive|3|3
4|InnerPassive|4|2

sqlite> select * from Shapes;
id|type|parameters
1|Box|XHalfLength=12000;YHalfLength=12000;ZHalfLength=12000
2|Box|XHalfLength=8000;YHalfLength=8000;ZHalfLength=10000
3|Box|XHalfLength=50;YHalfLength=300;ZHalfLength=300
4|Box|XHalfLength=40;YHalfLength=250;ZHalfLength=250

sqlite> select * from ChildrenPositions;
id|parentId|parentTable|position|childTable|childId
1|NULL|NULL|1|1|1
2|NULL|NULL|2|11|1
3|1|1|1|1|2
4|2|1|1|1|3
5|3|1|1|1|4
```

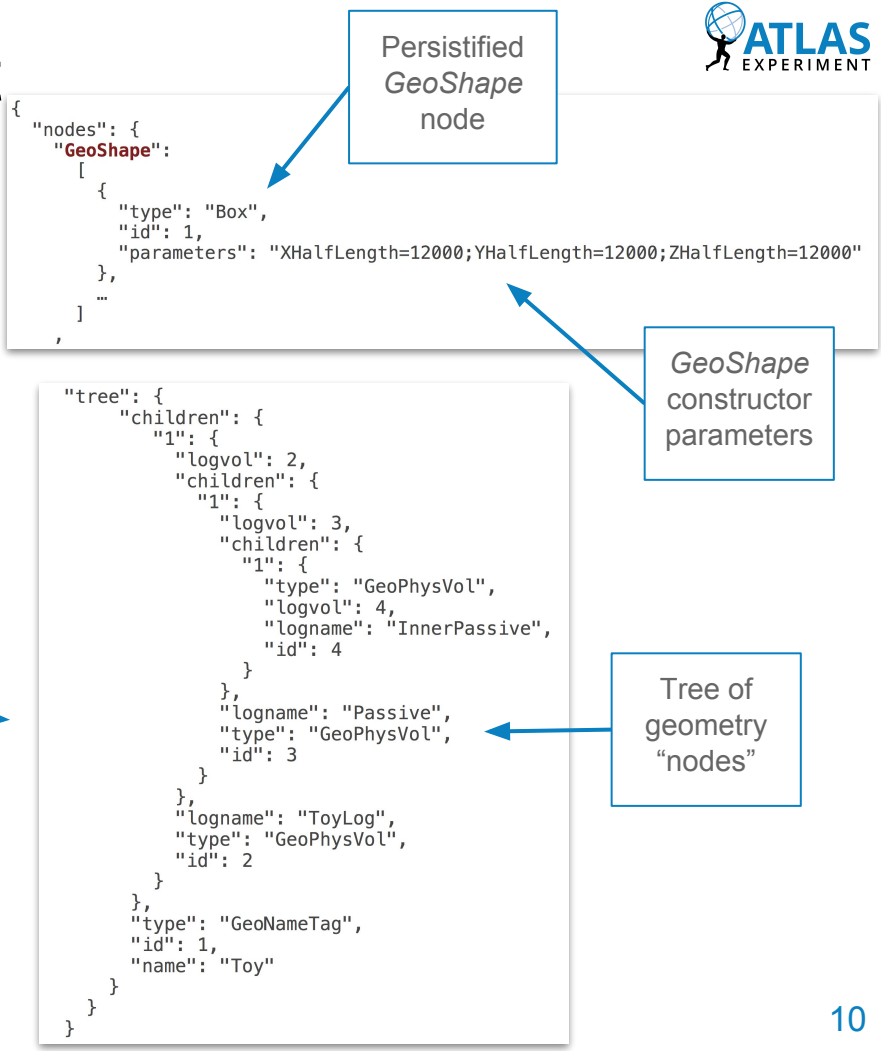
GeoModel tree structure stored in SQLite ("id"s only)

Geometry JSON data format

The **JSON** file is organized into two parts:

- A part **listing all the nodes** which are part of the persistified tree. **Shared** nodes are listed only **once**.
- A part storing the geometry “**tree**”.



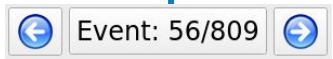


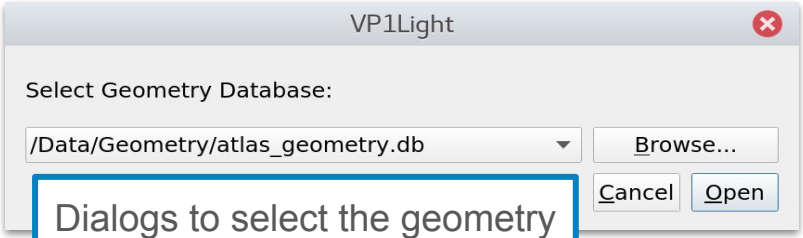
VP1Light - A standalone VP1

- Despite the advantage in terms of data access, the integration of VP1 in the experiment's framework puts limits when visualizing the geometry and developing or modernizing its code.
- We started a long procedure of porting base visualization packages outside of Athena: GUI, Utils, base data access, base functionalities
- VP1Light offers a lightweight experience to users who want to access the physics data stored in xAOD files and the geometry.
- Cross-platform, VP1Light can be run on Ubuntu Linux and macOS
- Much easier to modernize VP1Light than VP1: Running a lightweight framework simplifies improvement of visualization techniques/engines/libs

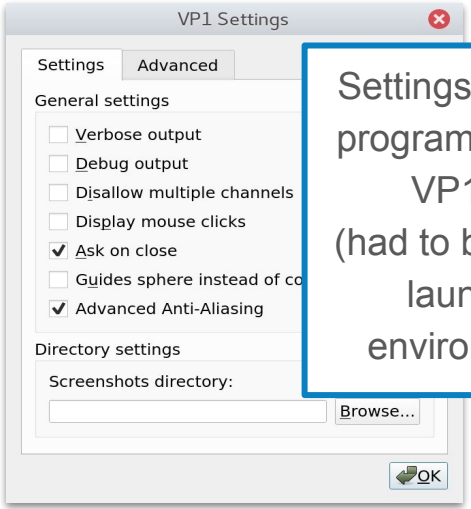
VP1Light: Improved user experience

Updated event browsing: Directly accessing xAOD files allows to not only go forward in an event file but also to go backwards and select specific events



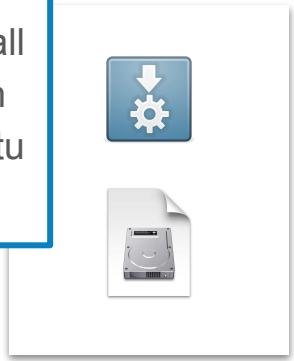


Dialogs to select the geometry database and xAOD file



Settings dialogs to change program options inside the VP1Light main UI (had to be activated before launching VP1 via environment variables)

Distribute application bundle including VP1Light binaries and all its dependencies. Users can run VP1Light out of the box on Ubuntu and macOS.



Conclusions

GeoModel

- GeoModel is now standalone and only dependent on Eigen
- Can be used as experiment-agnostic geometry library for detector description
- Standalone Geo2G4 export tool will offer Geant4 export (to be implemented)

VP1Light

- VP1Light is a lightweight, standalone event display for physics users
- Displays persistified geometry and physics events through xAOD files
- Will be available as an application bundle for Ubuntu and macOS

References

[1] **VP1@CHEP 2009**: Kittelmann T, Tsulaia V, Boudreau J, Moyse E, 2010 "*The Virtual Point 1 event display for the ATLAS experiment",* Journal of Physics: Conf. Ser. 219 032012, doi:10.1088/1742-6596/219/3/032012

[2] **VP1@CHEP 2016**: Bianchi R M, Boudreau J, Konstantinidis N, Martyniuk A C, Moyse E, Thomas J, Waugh B M, Yallup D P, *"Event visualization in ATLAS"* IOP Conf. Series: Journal of Physics: Conf. Series 898 (2017) 072014, doi :10.1088/1742-6596/898/7/072014

[3] **GeoModel@CHEP 2004**: Boudreau J and Tsulaia V 2004 "*The GeoModel Toolkit for Detector Description"*, CHEP '04, Book of Abstracts, https://indico.cern.ch/event/0/contributions/1294152/

[4] **GeometryDB@CHEP 2006**: Boudreau J, Tsulaia V, Hawkings R, Valassi A, Schaffer A *"Software Solutions for Variable ATLAS Detector Description", https://indico.cern.ch/event/408139/timetable/?view=standard#67-software-solutions-for-a-va*

[5] **GeoModel@CHEP 2016**: Bianchi R M, Boudreau J, Vukotic I, 2017 *J. Phys.: Conf. Ser.* **898** 072015, https://doi.org/10.1088/1742-6596/898/7/072015