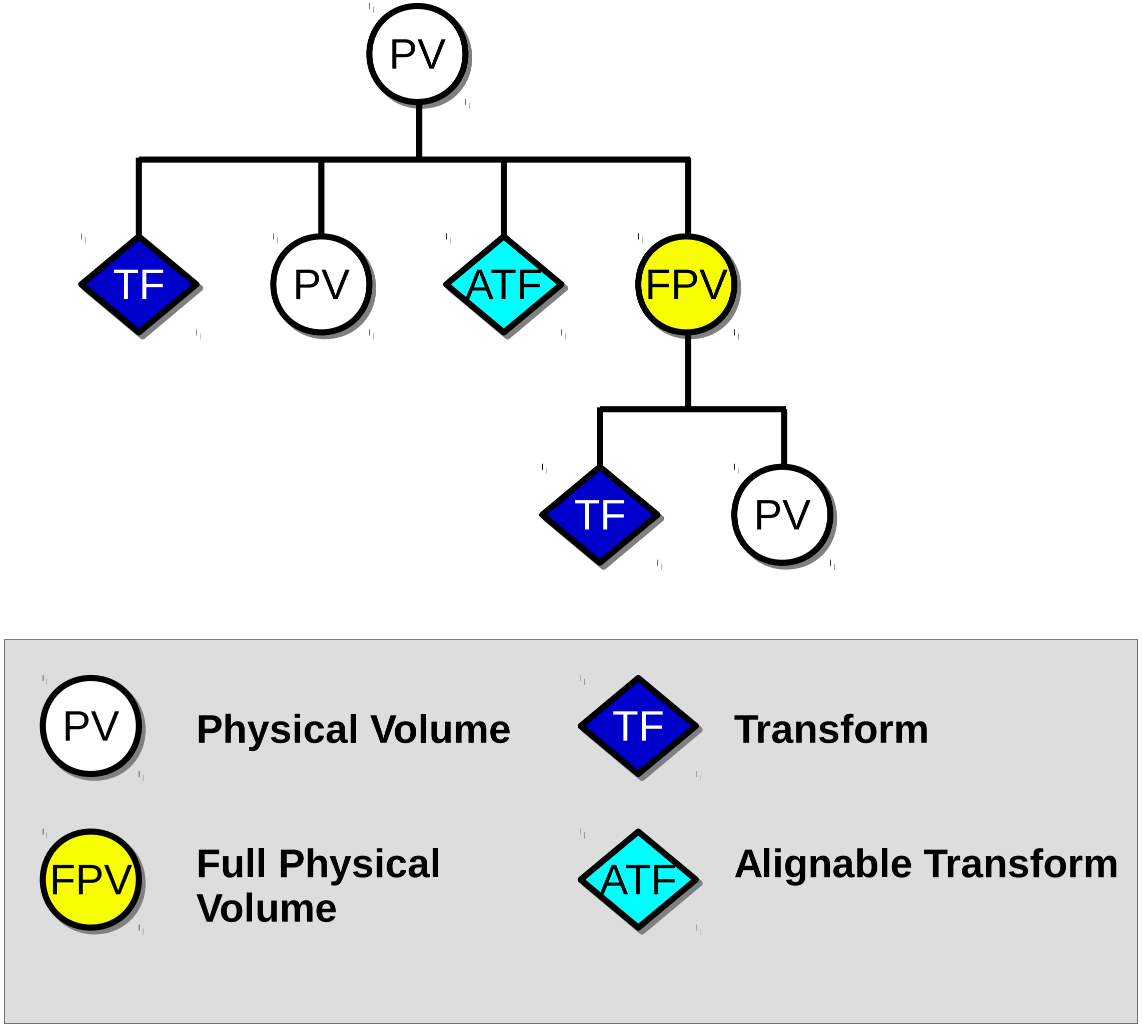
Alignments in AthenaMT

Proposing changes wrt the current system
Prototype implementation
Plans for the future

Vakho Tsulaia (LBNL)

ATLAS Software TIM, Glasgow
June 6, 2016

Static GeoModel tree



- **Physical Volumes.** Basic building blocks of the tree

- **Full Physical Volume**
  - **Computes and caches its absolute transform when this method gets called**

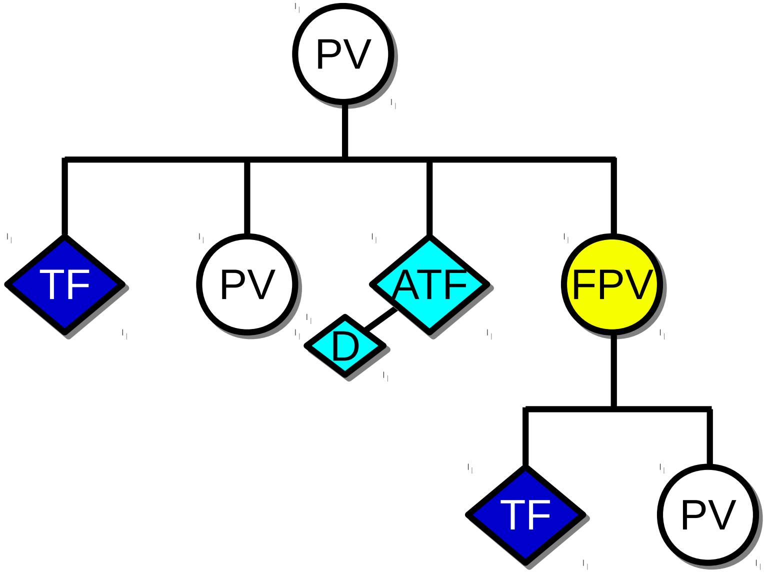
```
GeoVFullPhysVol::
    getAbsoluteTransform()
```

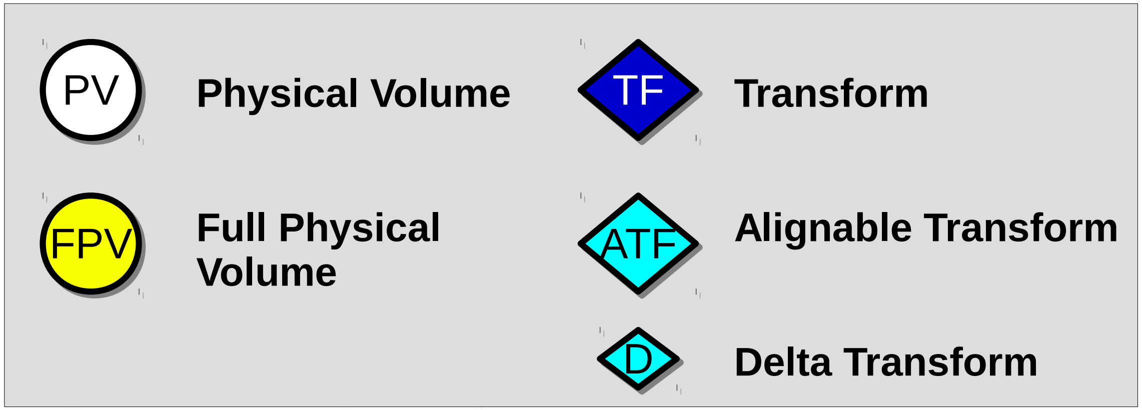
- **Transform.** Cannot be altered after construction

- **Alignable Transform**
  - **Can be altered multiple times during the job by calling**

```
GeoAlignableTransform::setDelta
   (const HepGeom::Transform3D&)
```

Applying alignments




PV — Physical Volume

TF — Transform

FPV — Full Physical Volume

ATF — Alignable Transform

D — Delta Transform

- **Alignment corrections** are stored in the **Conditions Database** in the form of **Delta Transforms**

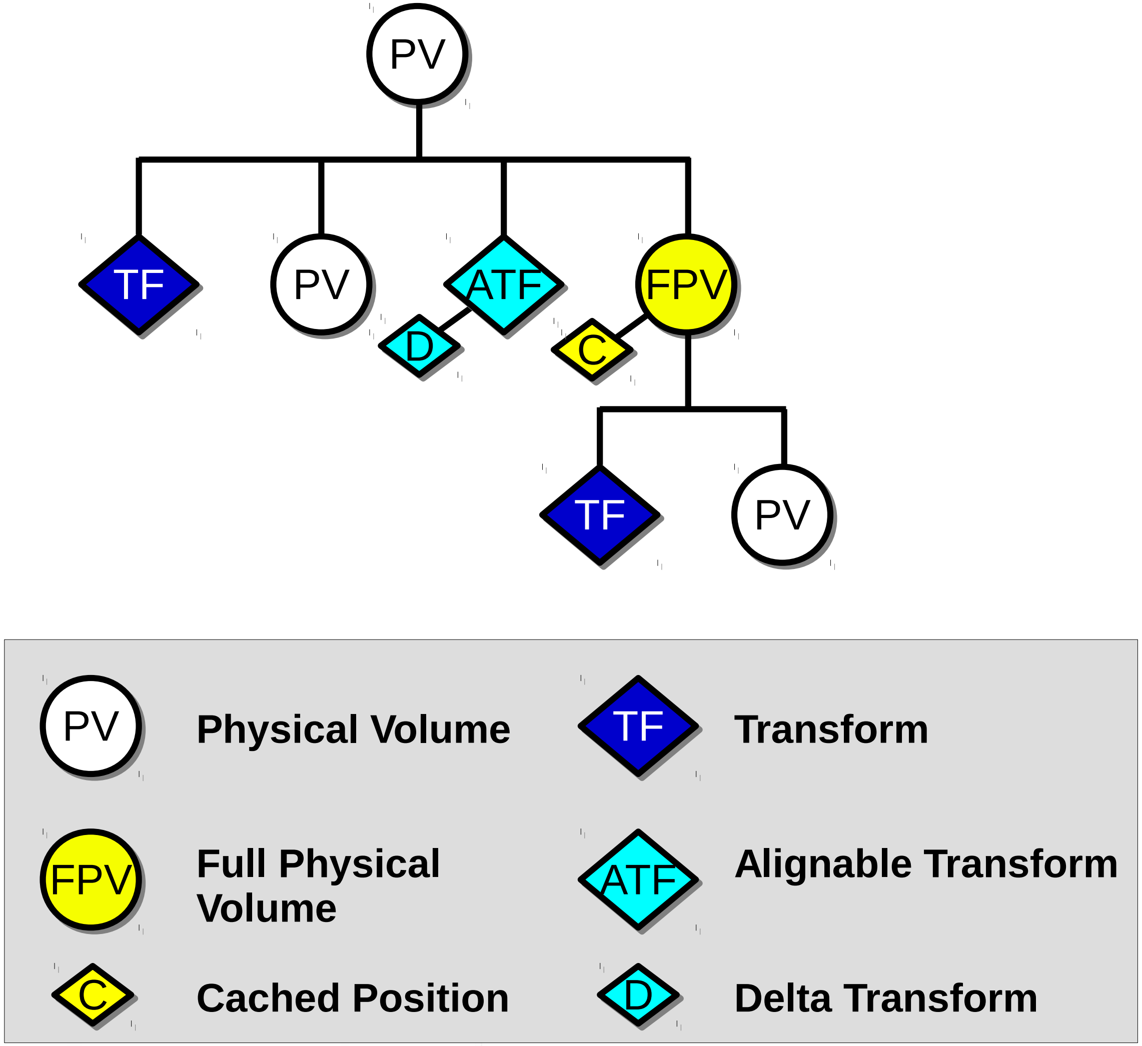
- **Each subsystem** manages its alignment data **independently from other subsystem**

- **Alignment data** is read in **callbacks** and is applied to GeoModel by using

```
GeoAlignableTransform::setDelta
    (const HepGeom::Transform3D&)
```

- **Delta is kept internally by Alignable Transform**

Caching global positions

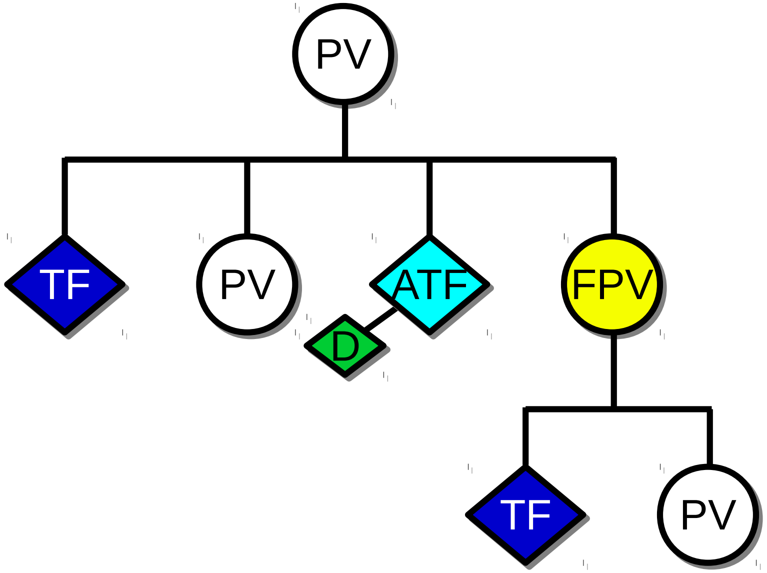


- When some client asks **Full Physical Volume** for its **Global Position …**

- … the **Full Physical Volume** computes the global position and **caches it**
  - This can happen at any time (not bound to any callbacks)

- **Cached position is kept internally by the Full Physical Volume**

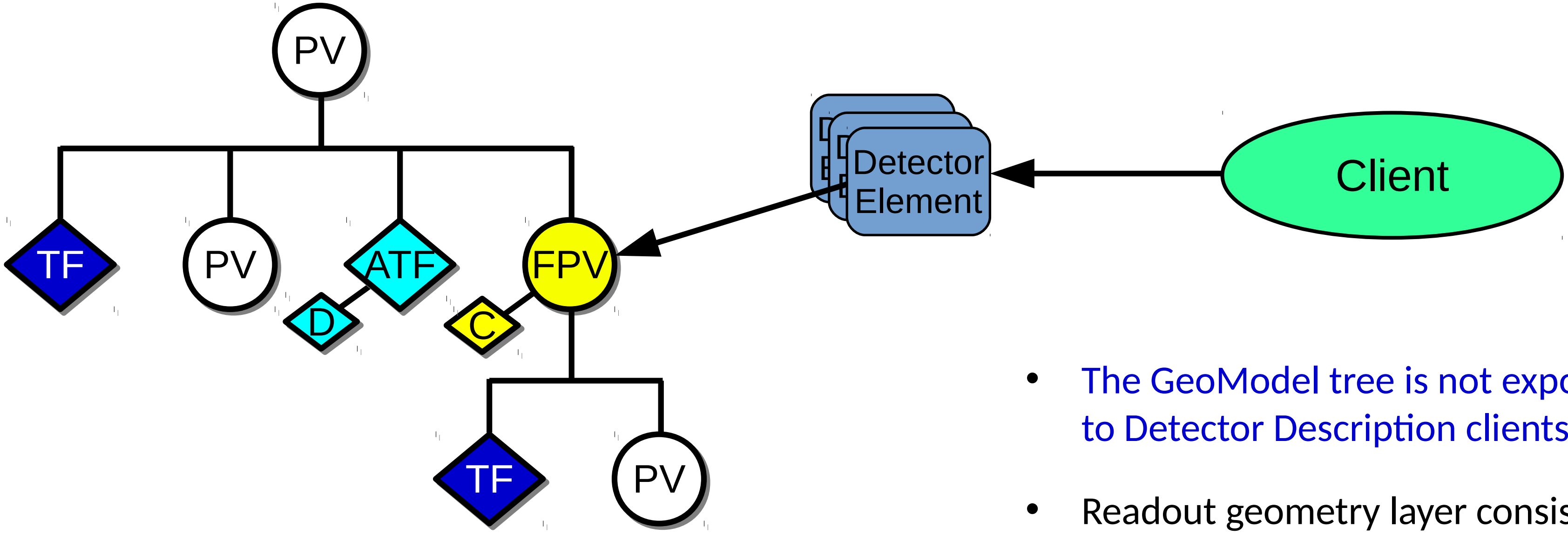
Updating alignments



PV	Physical Volume	TF	Transform
FPV	Full Physical Volume	ATF	Alignable Transform
C	Cached Position	D	Delta Transform

- When **alignments change** during the job …
- … callbacks **overwrite previous deltas** with new deltas and …
- … Full Physical Volume **position caches are cleared**

Readout geometry



PV

TF

PV

ATF

D

FPV

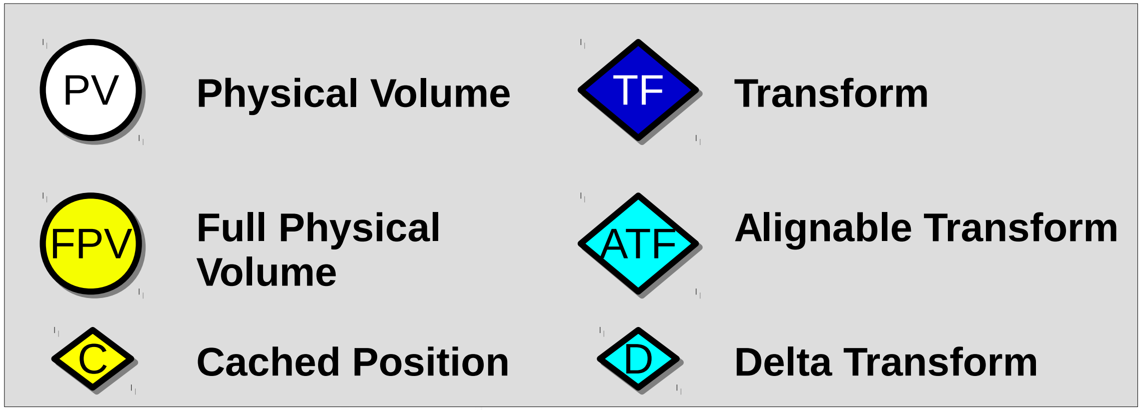
C

TF

PV

Detector Element

Client



PV	**Physical Volume**	TF	**Transform**
FPV	**Full Physical Volume**	ATF	**Alignable Transform**
C	**Cached Position**	D	**Delta Transform**

- The GeoModel tree is not exposed to Detector Description clients
- Readout geometry layer consists of substem specific **Detector Elements**
- Each Detector Element has a **pointer to Full Physical Volume**

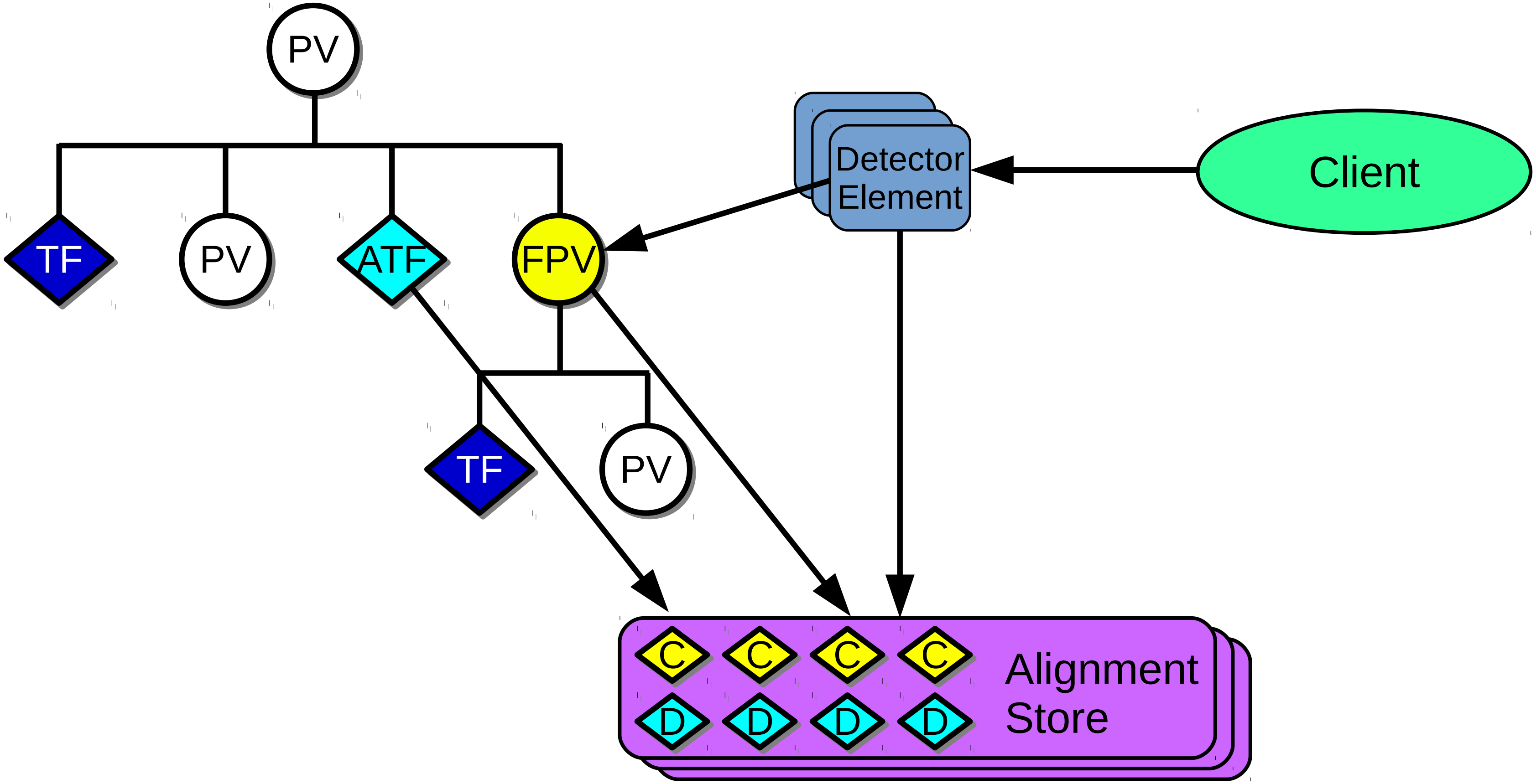
Alignments in MT

- The mechanism described so far **works well in serial** Athena, but it is **not going to work in MT** environment if we want to support multiple alignments in flight

- For AthenaMT we need to **decouple the static (read-only) part of GeoModel from the part that is sensitive to alignment changes:**
  - **Deltas of Alignable Transforms**
  - **Position caches of Full Physical Volumes**

- The proposal is to introduce a special container for the alignment-sensitive information: the **Alignment Store**

Alignment Store



Alignment Store (contd.)

- **The Alignment Store** is a regular **Conditions Object**, so it should be handled as any other Conditions Object in AthenaMT
  - Created by a **Conditions Algorithm** (replacement of current callback function)
  - Stored into **Conditions Container** using **Write Conditions Handle**

- By making **Detector Elements aware of the Alignment Store** we can hopefully **make the transition transparent to Detector Description clients**

Progress so far

- The decoupling of read-only GeoModel from the alignment-sensitive part was **tested in a prototype**, which was developed in early January
  - The changes affected 10 out of 70+ classes in **GeoModelKernel**

- By that time the new Conditions Access infrastructure was not yet implemented, so **the prototype did not use Conditions Handles, Conditions Containers etc.**

- On the client side, the mechanism was successfully tested in **TRT GeoModel**
  - The testing was done only in serial mode
  - Proof of principle …

Next steps

- Migrate the existing prototype – core GeoModel and TRT_GeoModel code – to the Conditions Access infrastructure in AthenaMT

- Test it in serial Athena and AthenaMT

- Discuss the implementation details with the developers of Detector Description clients
    - A presentation in RIG meeting?

- Proceed with putting the code into release, migration of all subsystems etc.