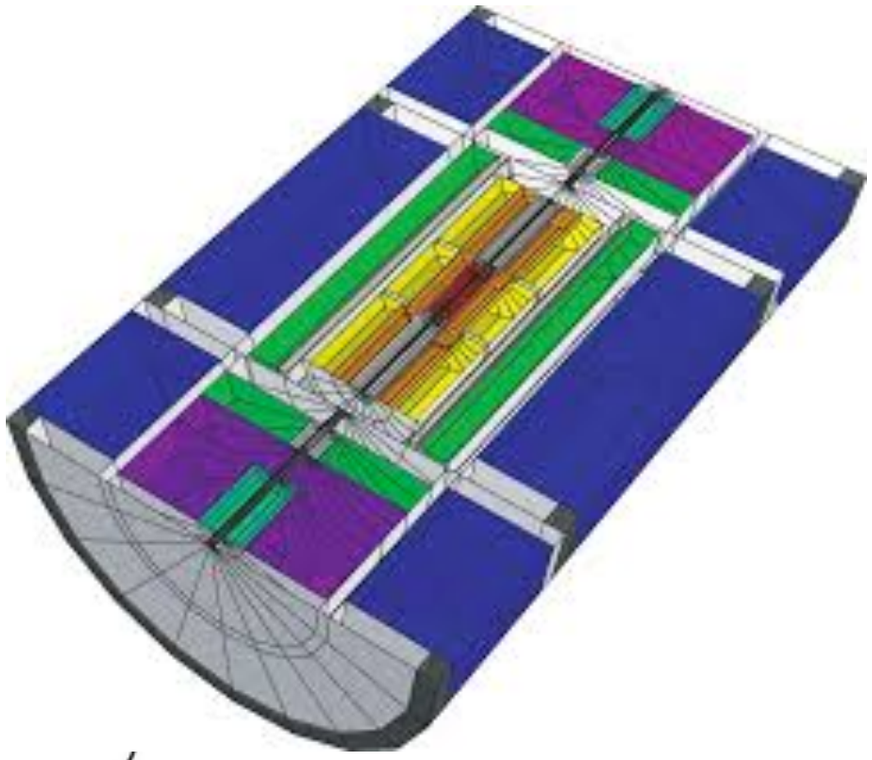
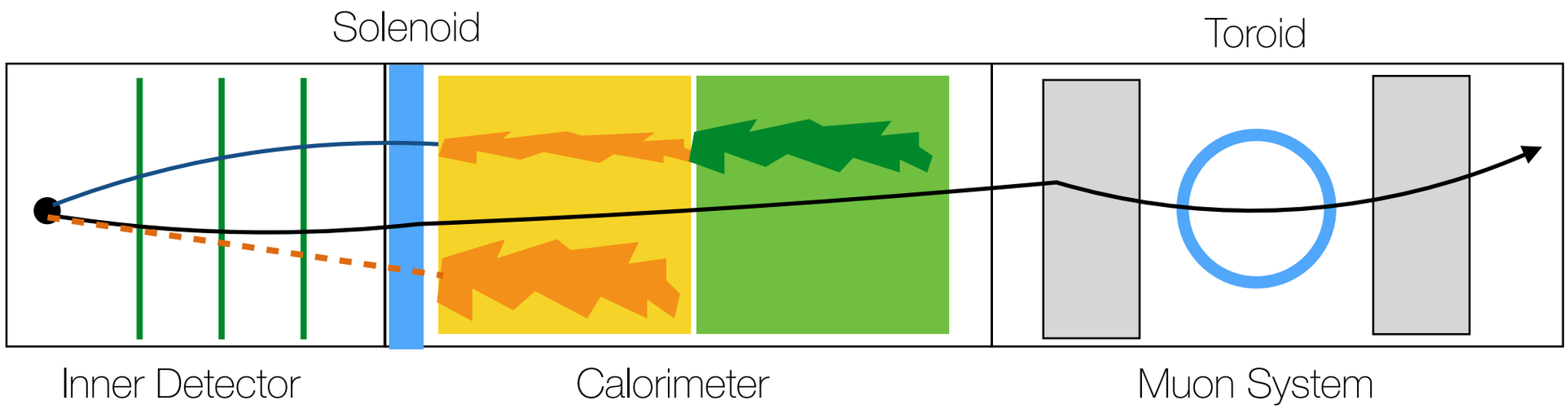


ATLAS Tracking Software: history, status & prospects

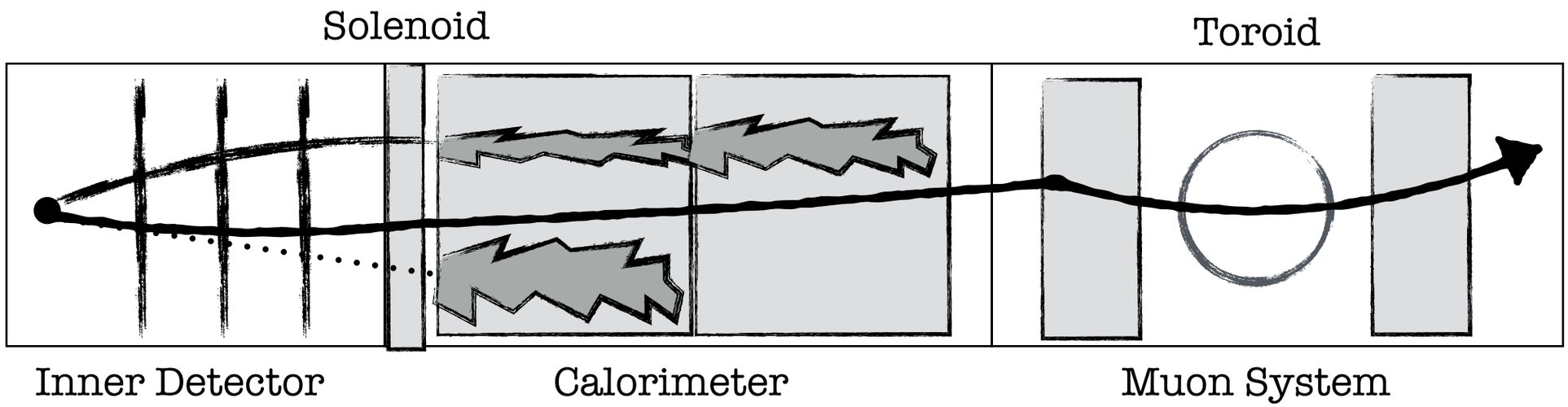
A. Salzburger (CERN) on behalf of the ATLAS Tracking SW

The Reality



- A simplified “Tracker” view of ATLAS
  - two precision tracking systems having
    very different magnetic field setups
    very different detecting technologies
    very different dimensions
  - some lump of material in between

The **FORTRAN** times



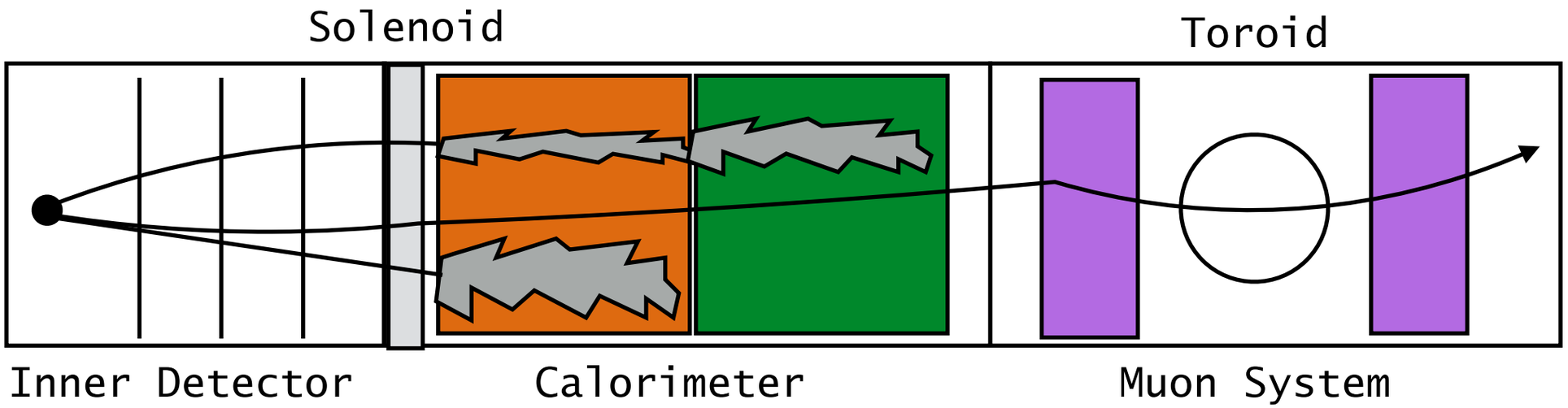
2 monolithic track reconstruction programs in FORTRAN:

- helical track model
- internal event data model
- from pattern recognition to track fitting within one program

something nasty you have to deal with

1 main track reconstruction programs in FORTRAN with a chi2 combination with the Inner Detector reconstruction, no combined fit

The **C** times / Early ATHENA days



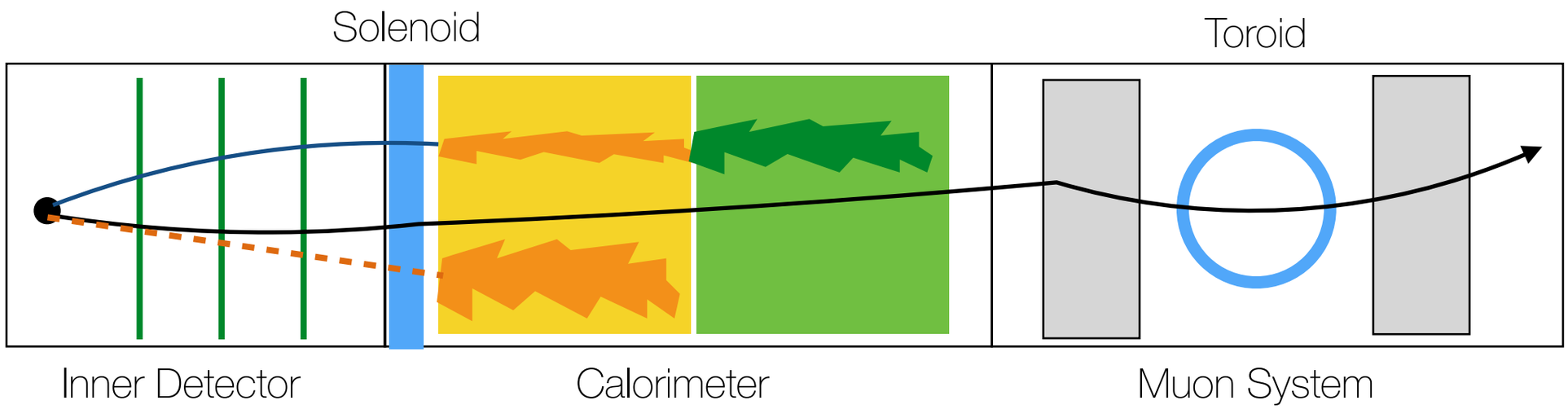
2 monolithic track reconstruction programs in FORTRAN with semicolons at the end
- helical track model
- internal event data model
- from pattern recognition to track fitting within one program

something nasty you have to deal with, 2 independent models to do so

2 main track reconstruction programs, one in FORTRAN, on C, with either a chi2 combination with the Inner Detector reconstruction, or a combined track fit

The RTF report / ATLAS Common Tracking

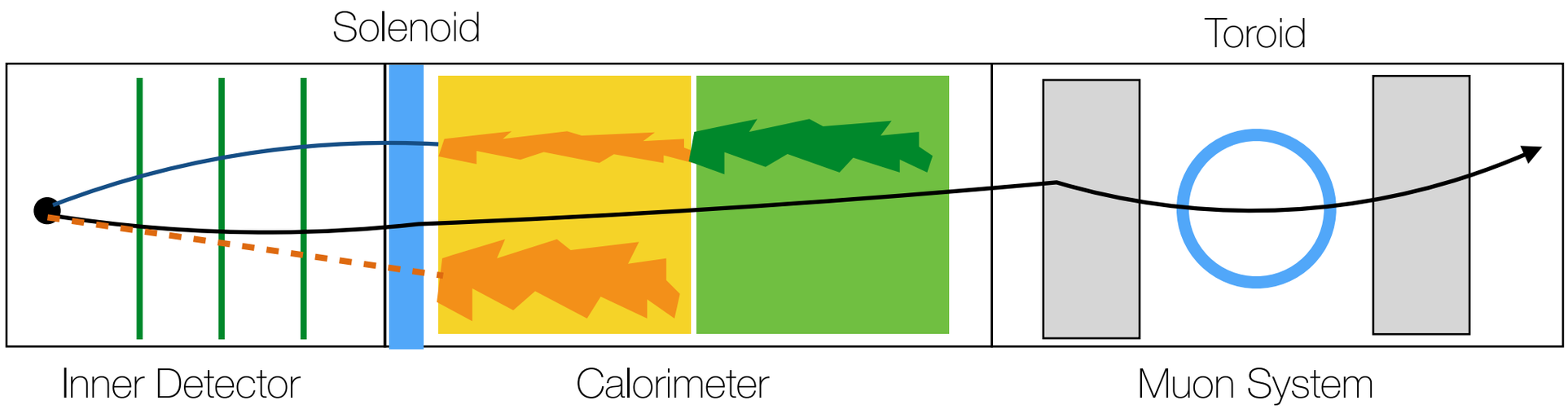
ATL-SOFT-2003-010



- ATLAS Reconstruction Task Force (RTF) review & recommendations
  - final report issued in 2003, recommendations to restructure ATLAS software
- A simplified "Tracker" view of ATLAS
  - two precision tracking systems having

very different magnetic field setups	→	field-agnostic parameterisation
very different detecting technologies	→	technology-agnostic high level tracking
very different dimensions	→	re-calibration on demand
- some lump of material in between	→	integration of calorimeter into tracking

Current structure - from reality to repository



Tracking

Geometry (TrkDetDescr)	Event Data Model (TrkEvent)	Extrapolation (TrkExtrapolation) A	Fitting (TrkFitter) A	Calibration, general (TrkTools) A	Alignment (TrkAlignment) A

InnerDetector

EDM (InDetRecEvent)	Geometry (InDet TrackingGeometry) A	Tools (InDetRecTools) A

Calorimeter

EDM	Geometry (Calo TrackingGeometry) A

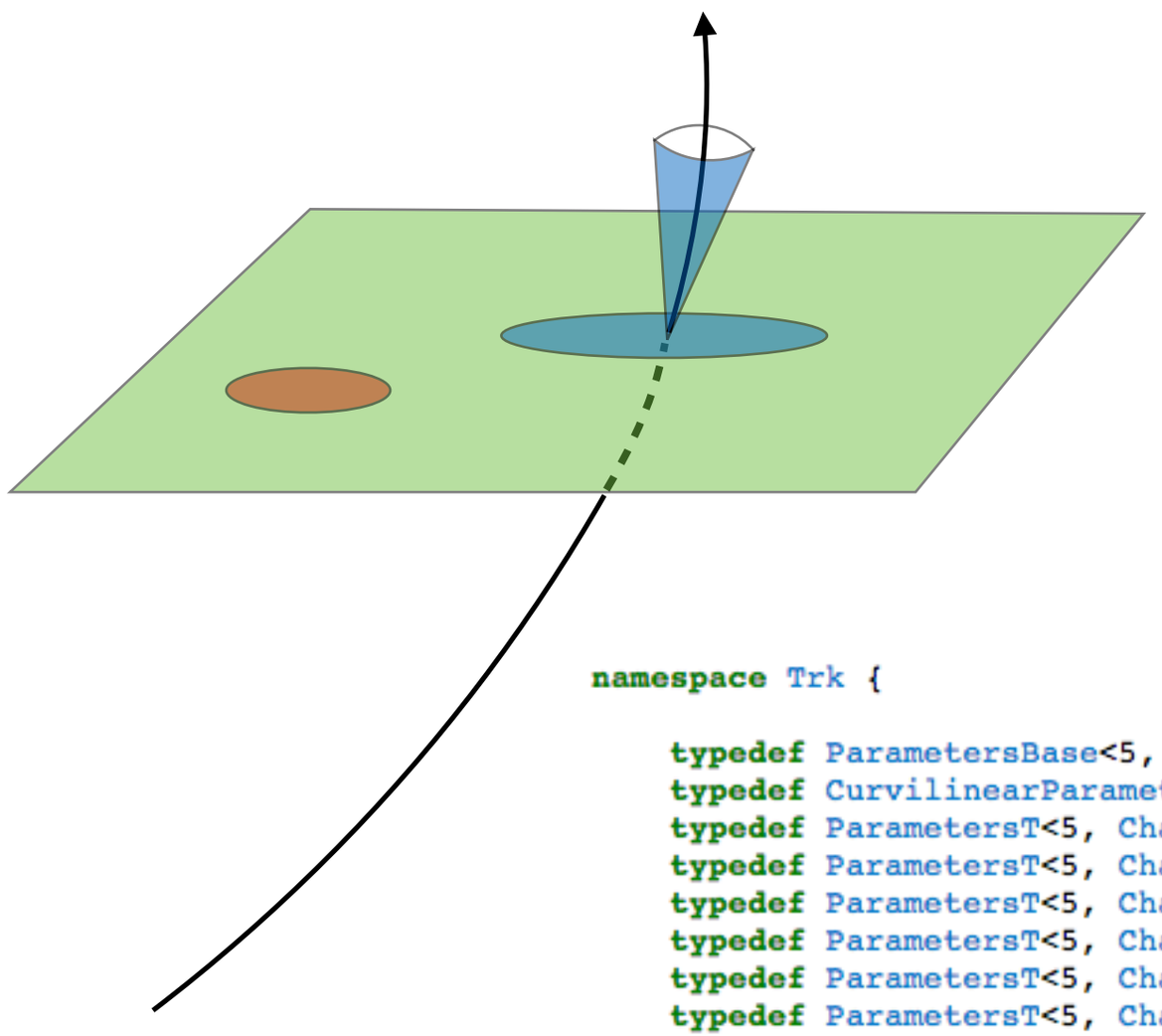
MuonSpectrometer

EDM (MuonRecEvent)	Geometry (Muon TrackingGeometry) A	Tools (MuonRecTools) A

A … embedded in Gaudi/Athena structure with AlgTools/Algorithms/Services

ATLAS Tracking Geometry - in a sketch (1)

`Trk::Surface` class acts as a representation of a detector element
(proxy mechanism allows to bind it to basically all geometry models)

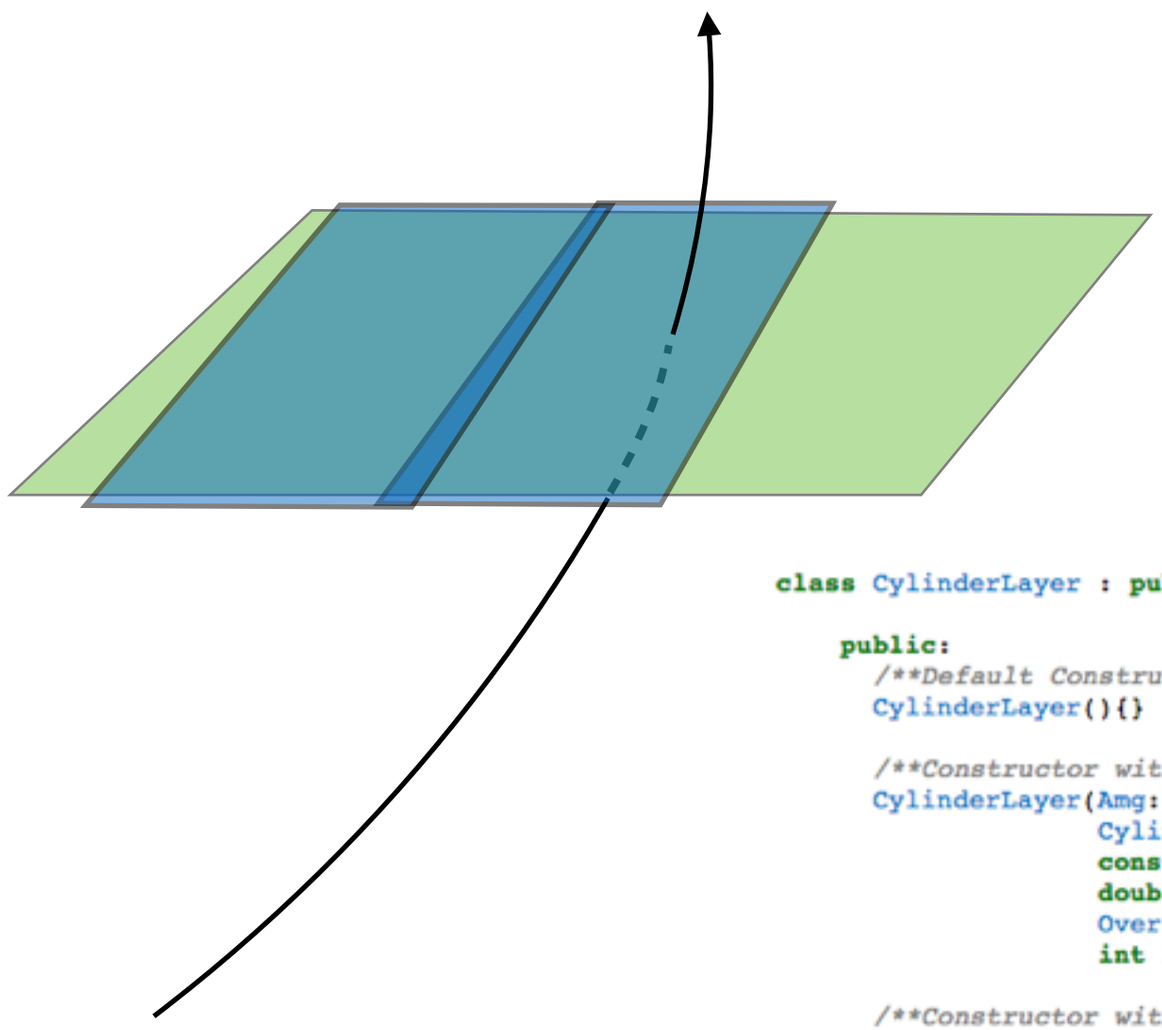


for measurements
and track parameterisation

```
namespace Trk {

    typedef ParametersBase<5, Charged>                          TrackParameters;
    typedef CurvilinearParametersT<5, Charged, PlaneSurface> CurvilinearParameters;
    typedef ParametersT<5, Charged, ConeSurface>                AtaCone;
    typedef ParametersT<5, Charged, CylinderSurface>            AtaCylinder;
    typedef ParametersT<5, Charged, DiscSurface>                AtaDisc;
    typedef ParametersT<5, Charged, PerigeeSurface>             Perigee;
    typedef ParametersT<5, Charged, PlaneSurface>               AtaPlane;
    typedef ParametersT<5, Charged, StraightLineSurface>        AtaStraightLine;

}
```

ATLAS Tracking Geometry - in a sketch (2)

`Trk::Surface` class acts as a base for **`Trk::Layer`**

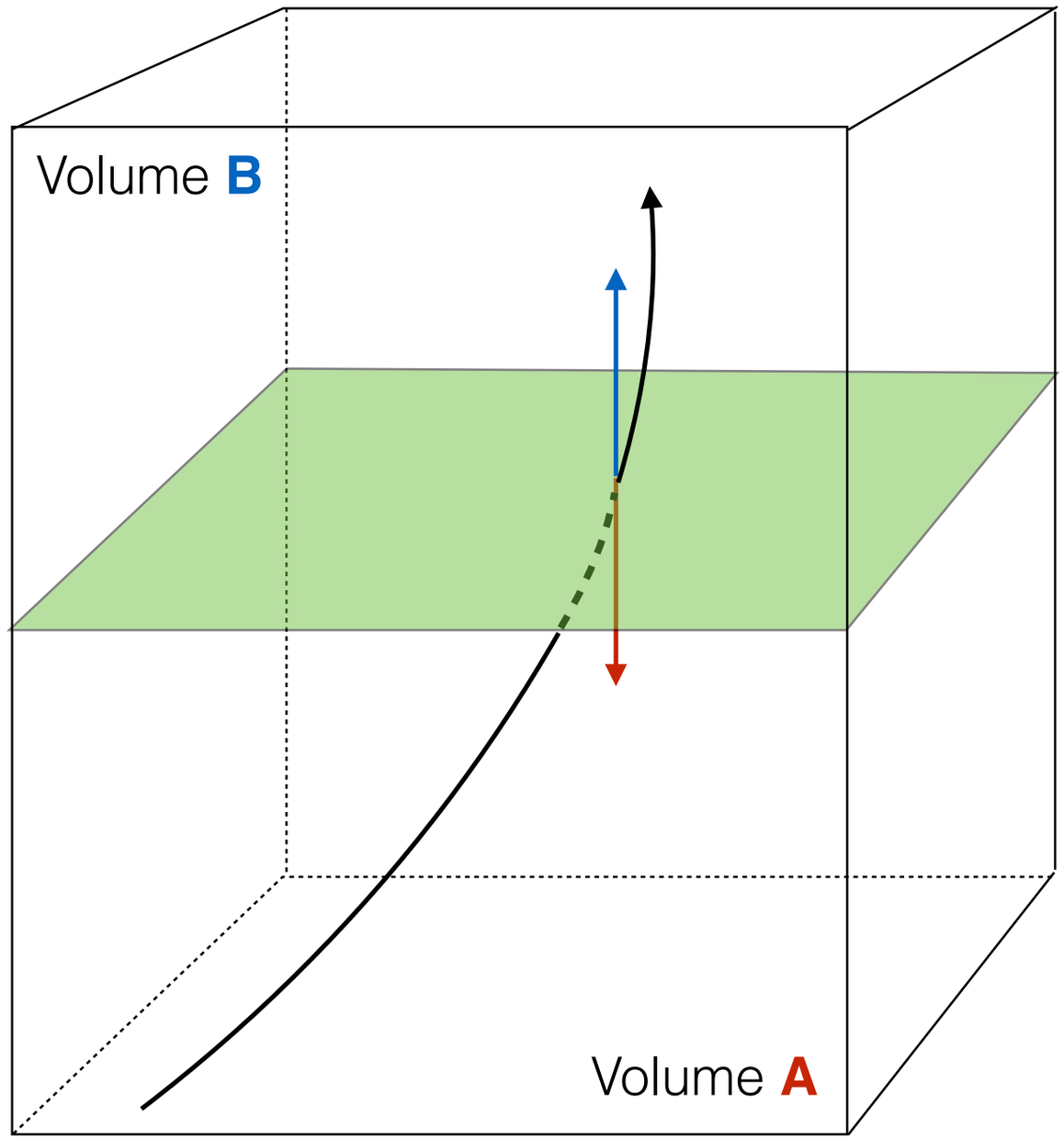


for grouping objects, e.g
detector elements on layers

```
class CylinderLayer : public CylinderSurface, public Layer {

    public:
      /**Default Constructor*/
      CylinderLayer(){}

      /**Constructor with CylinderSurface components and  MaterialProperties */
      CylinderLayer(Amg::Transform3D* transform,
                    CylinderBounds* cbounds,
                    const LayerMaterialProperties& laymatprop,
                    double thickness = 0.,
                    OverlapDescriptor* od = 0,
                    int laytyp=int(Trk::active));

      /**Constructor with CylinderSurface and  MaterialProperties */
      CylinderLayer(CylinderSurface* cyl,
                    const LayerMaterialProperties& laymatprop,
                    double thickness = 0.,
                    OverlapDescriptor* od = 0,
                    int laytyp=int(Trk::active));
```

ATLAS Tracking Geometry - in a sketch (3)

`Trk::Surface` class acts as a (shared) boundaries for **`Trk::Volumes`**

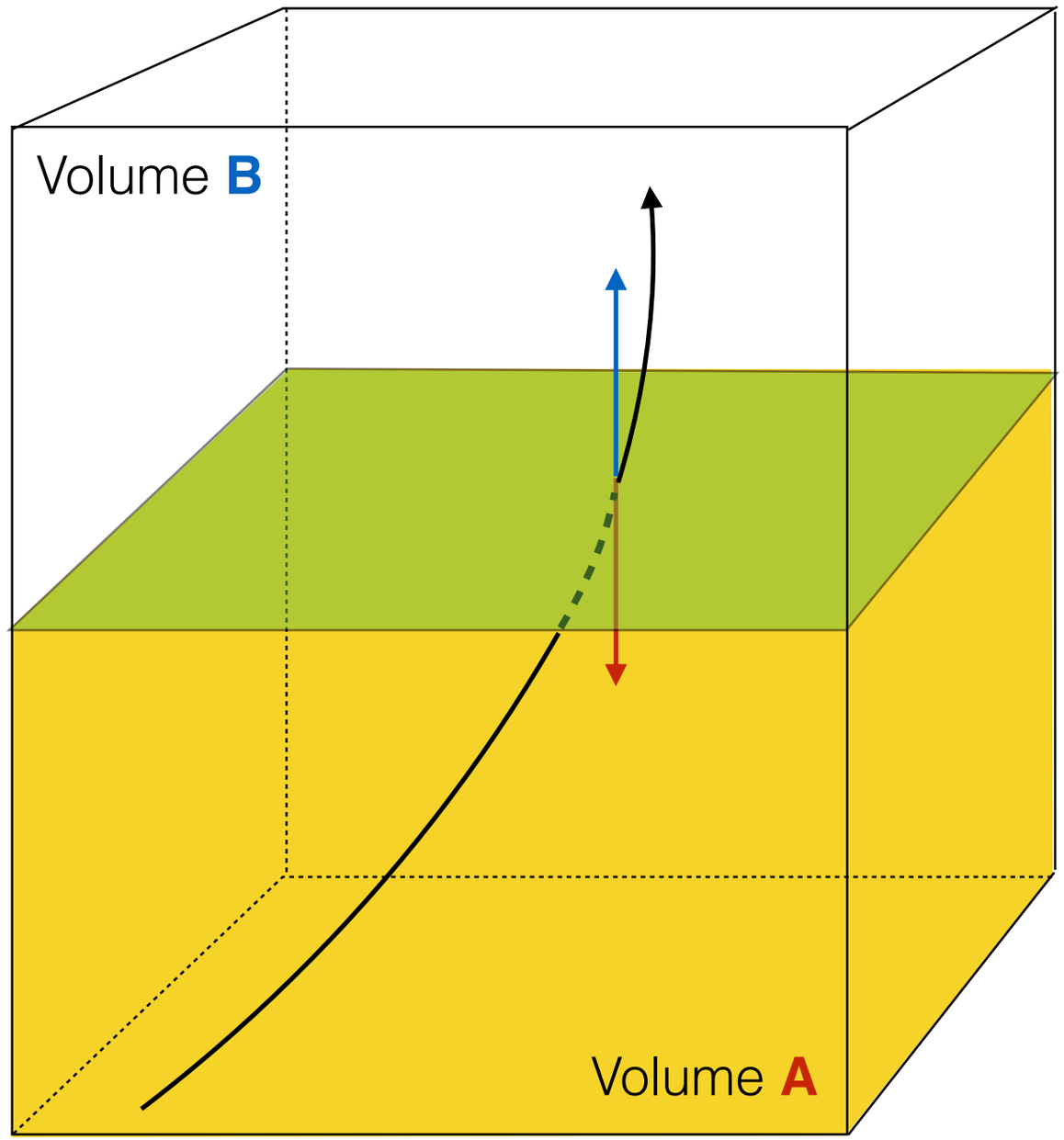


full connective geometry,
i.e. every boundary surface is
attached to the next volume(s)

navigation through geometry
comes as an intrinsic feature
of the extrapolation process.

ATLAS Tracking Geometry - in a sketch (4)

`Trk::Volumes` exist also in a **dense** flavour (e.g. for calorimeter description)

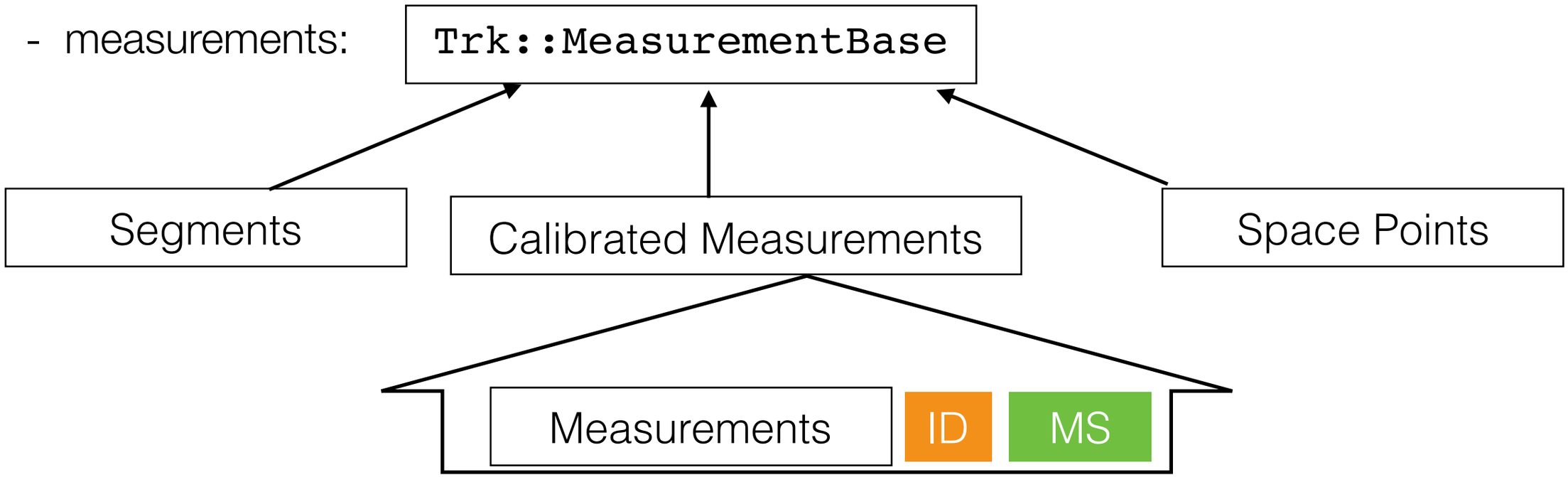


ATLAS has developed a special propagation engine for propagation through dense material with an instantaneous integration of material interactions

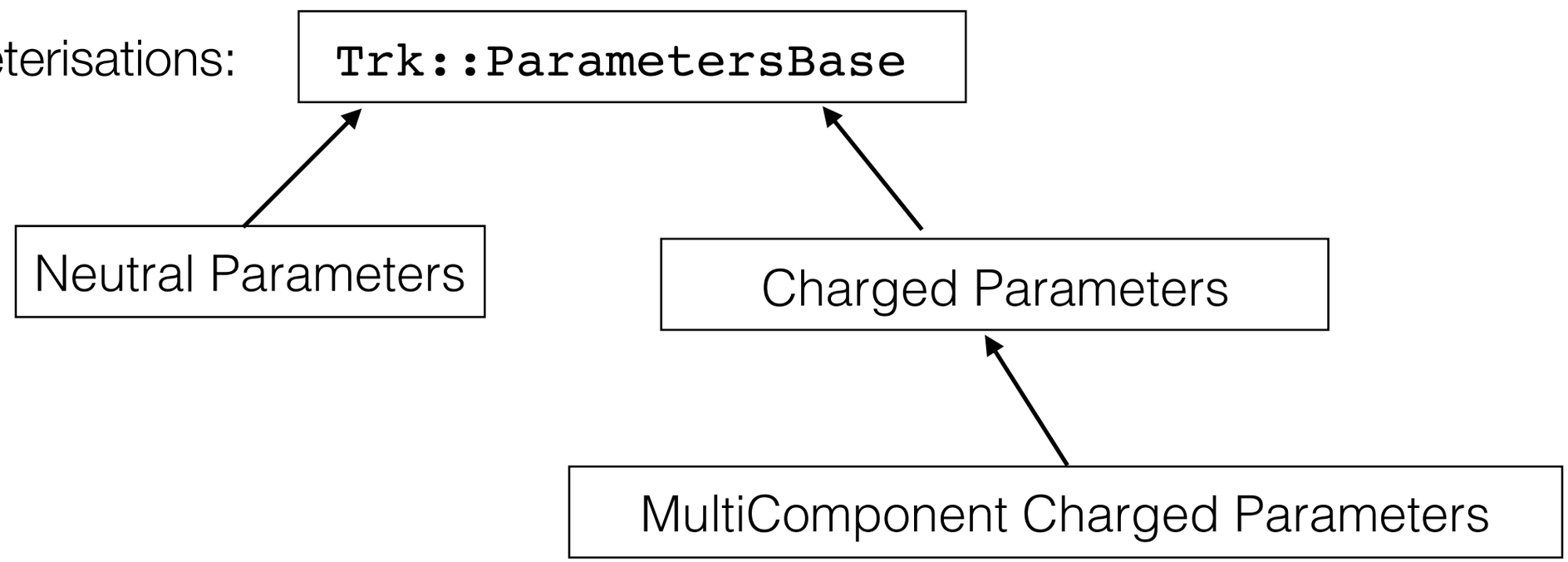
STEP_Propagator
ATL-SOFT-PUB-2008-003

ATLAS Tracking SW EDM - 2 columns

- (Transient) Tracking EDM is built upon two base classes describing
  - measurements:

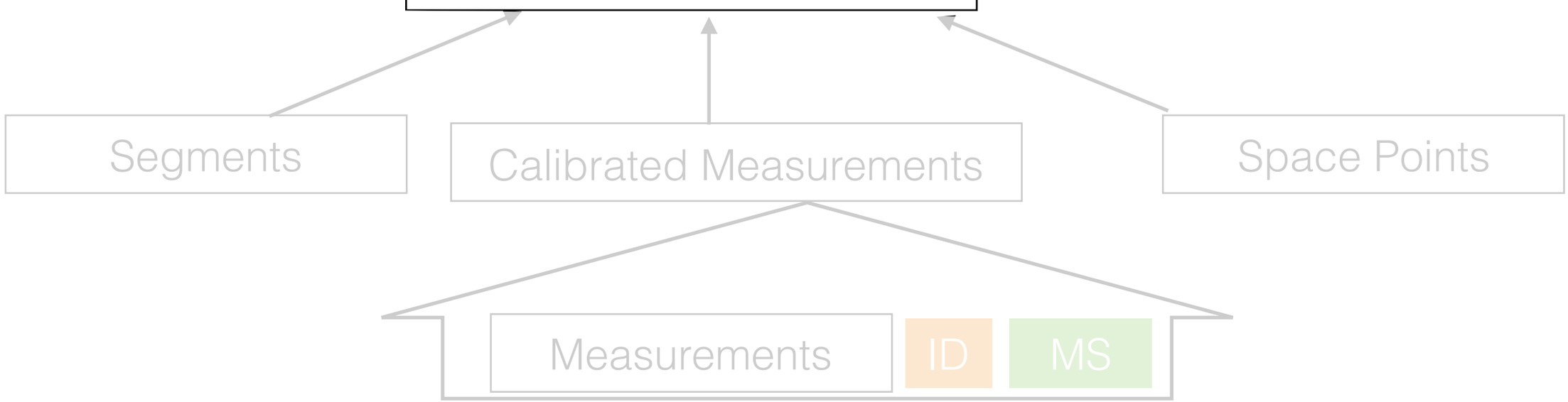


  - parameterisations:

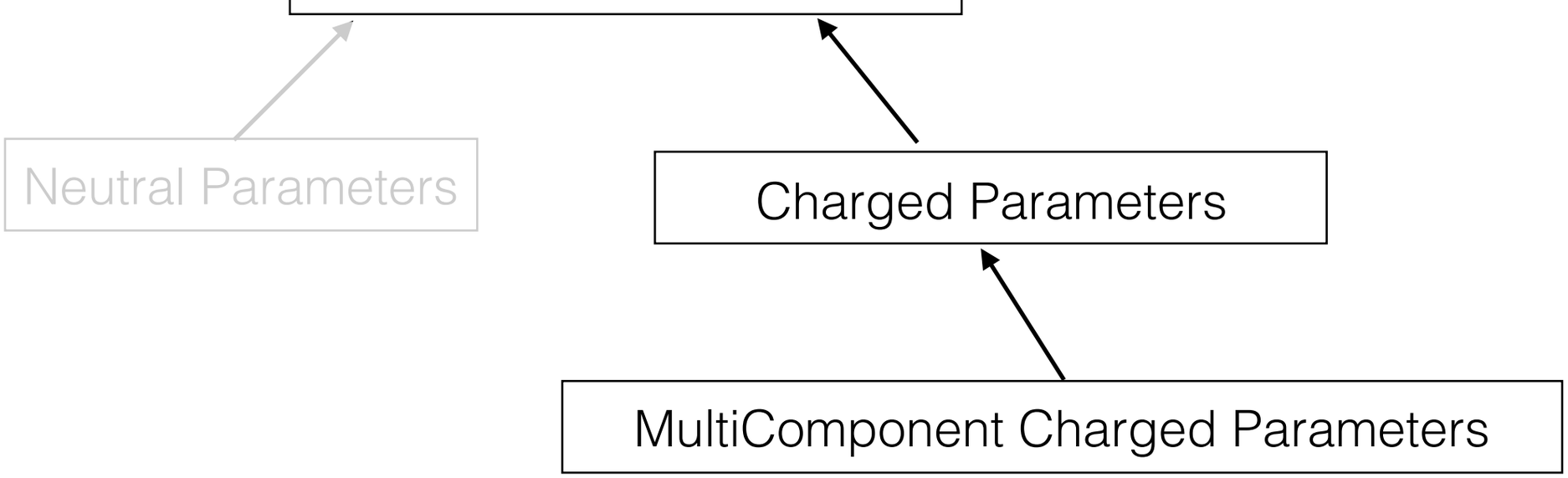


ATLAS Tracking SW EDM - Fitter view

- That's what the TrackFitter sees
  - measurements: `Trk::MeasurementBase`

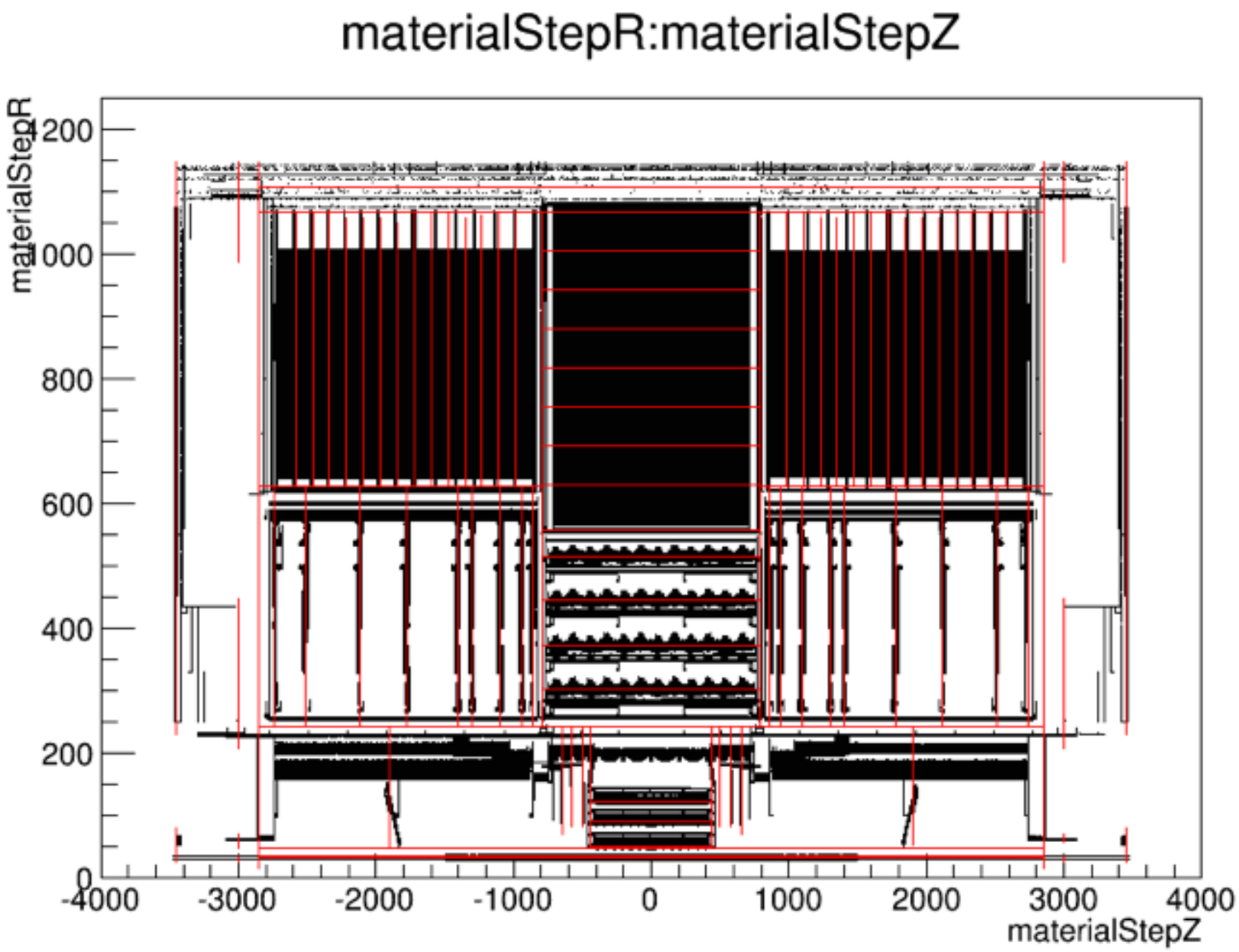


  - parameterisations: `Trk::ParametersBase`



TrackingGeometry - simplification of nodes

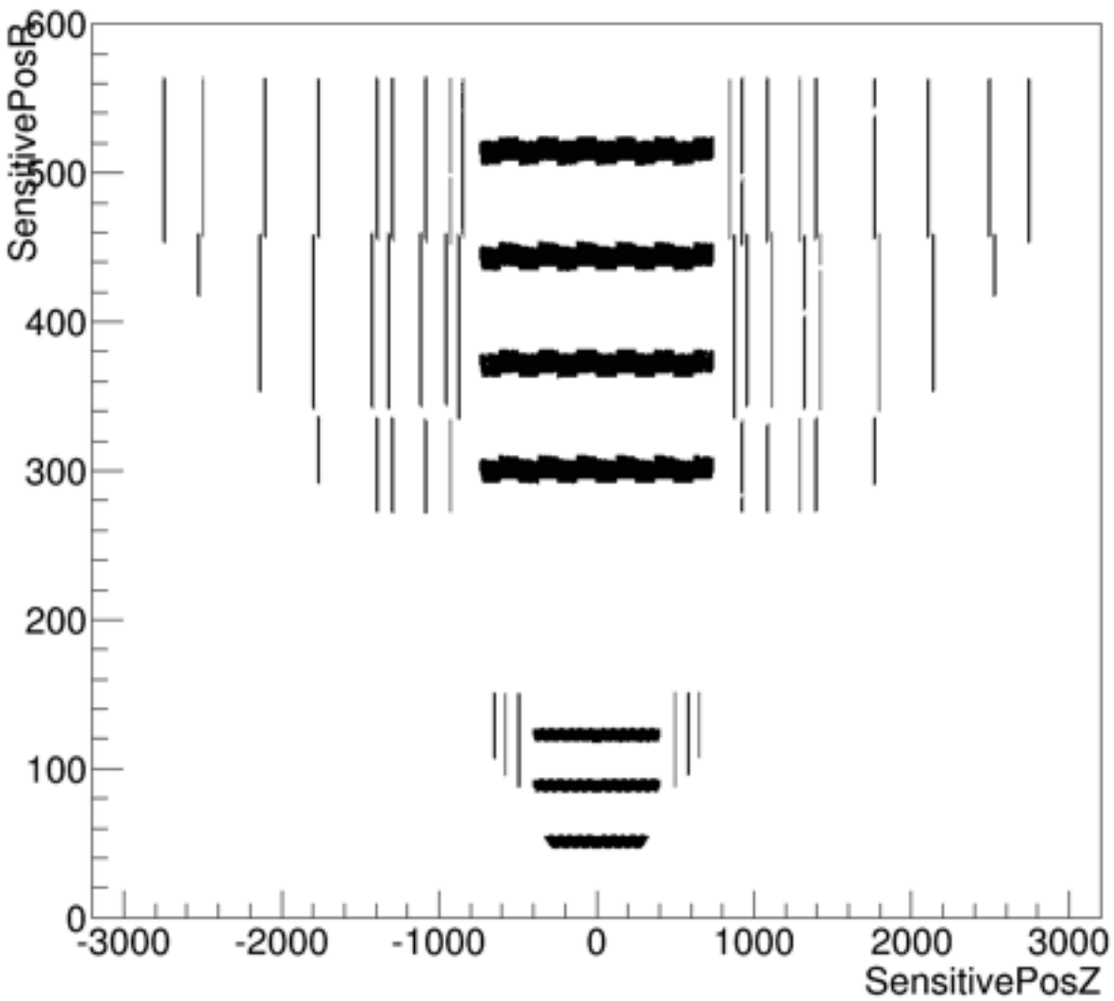
- Layers are described as cylinders and disks
  - sensitive material in CURRENT tracking geometry only on those
  - even though the modules are perfectly known to the TrackingGeometry

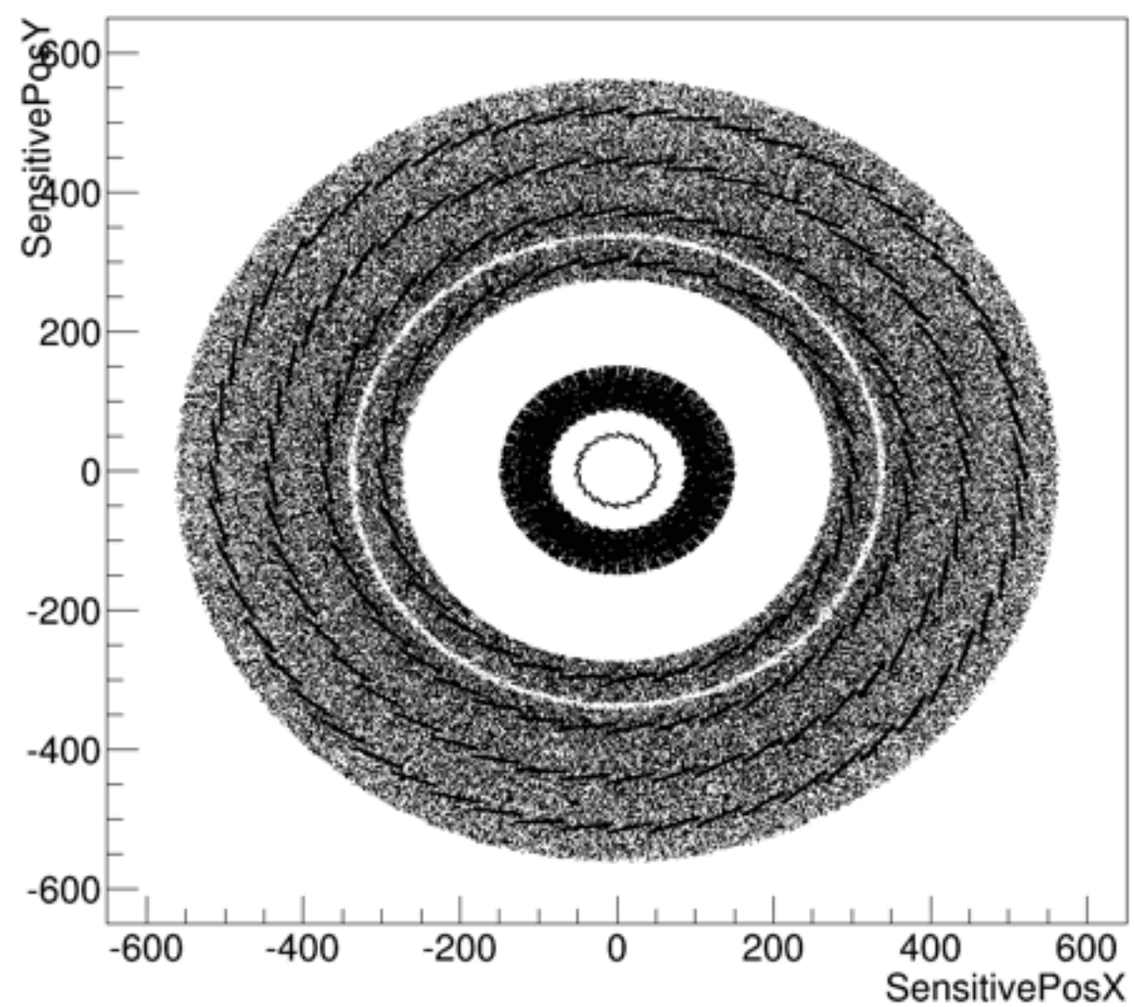


Intermezzo 1 - Fast Track Simulation (Fatras)

ATL-SOFT-PUB-2008-001

- Embedded navigation with Extrapolation engine is a fast track simulation
  - simply changing stochastic material effects integration into MC based one
  - using the reconstruction geometry as a simulation geometry is a common concept for fast simulation



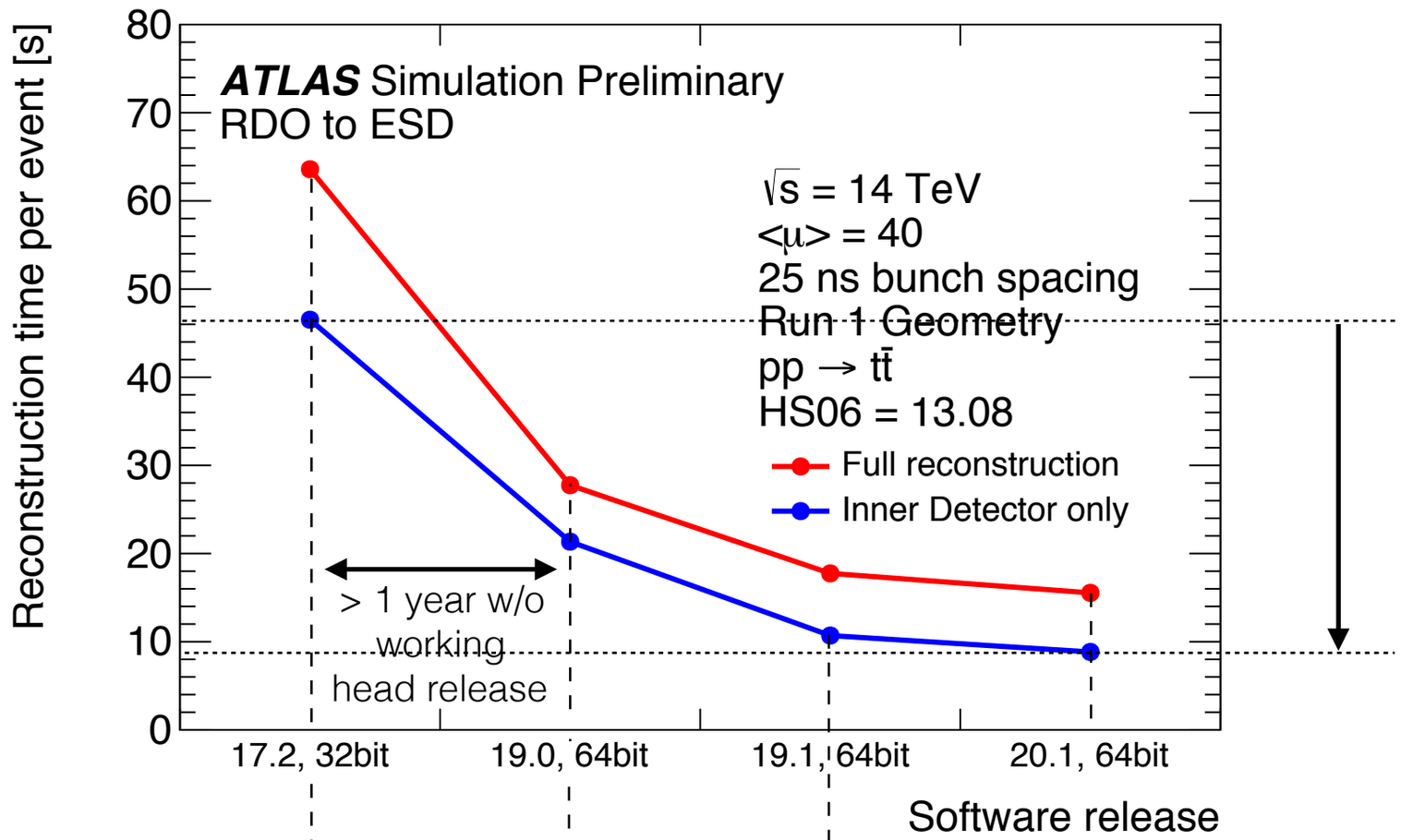


Intermezzo 2 - the LS1 campaign

- LHC reconstruction software is arguably the best-tested reconstruction software in HEP
  - Run-1 data and yearly MC campaigns are O(10^9) events each
  - gives a lot of possibilities to test topological cases
  - takes a lot of time
- ATLAS ran a big software campaign to clean up reconstruction software for Run-2
  - infrastructure changes (move from CLHEP to Eigen math library)
  - simplification of the Event Data Model (both transient & analysis side)
  - code cleanup and optimisations
  - updates to pattern recognition strategies

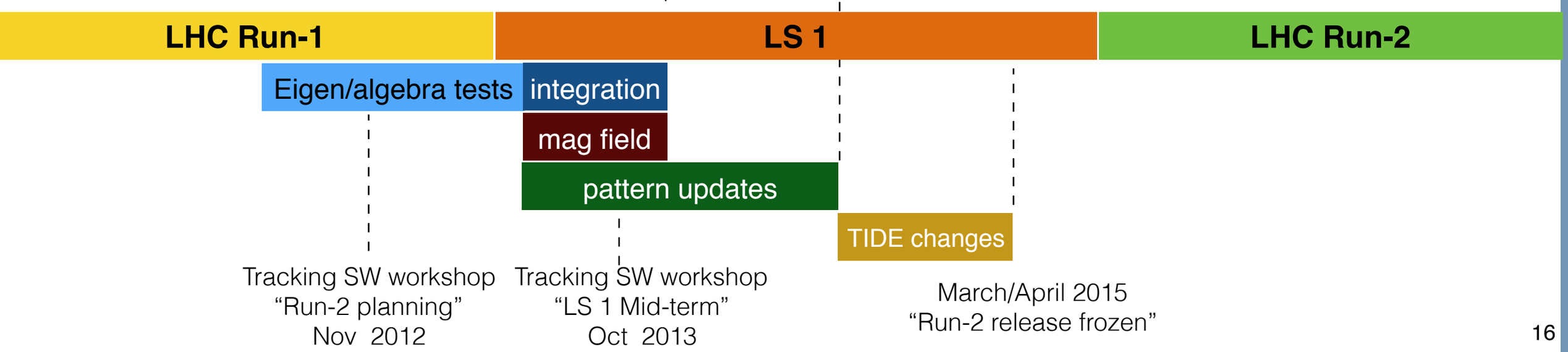
Outcome of the LS1 campaign

- CPU comparison during LS-1 without loss of physics performance

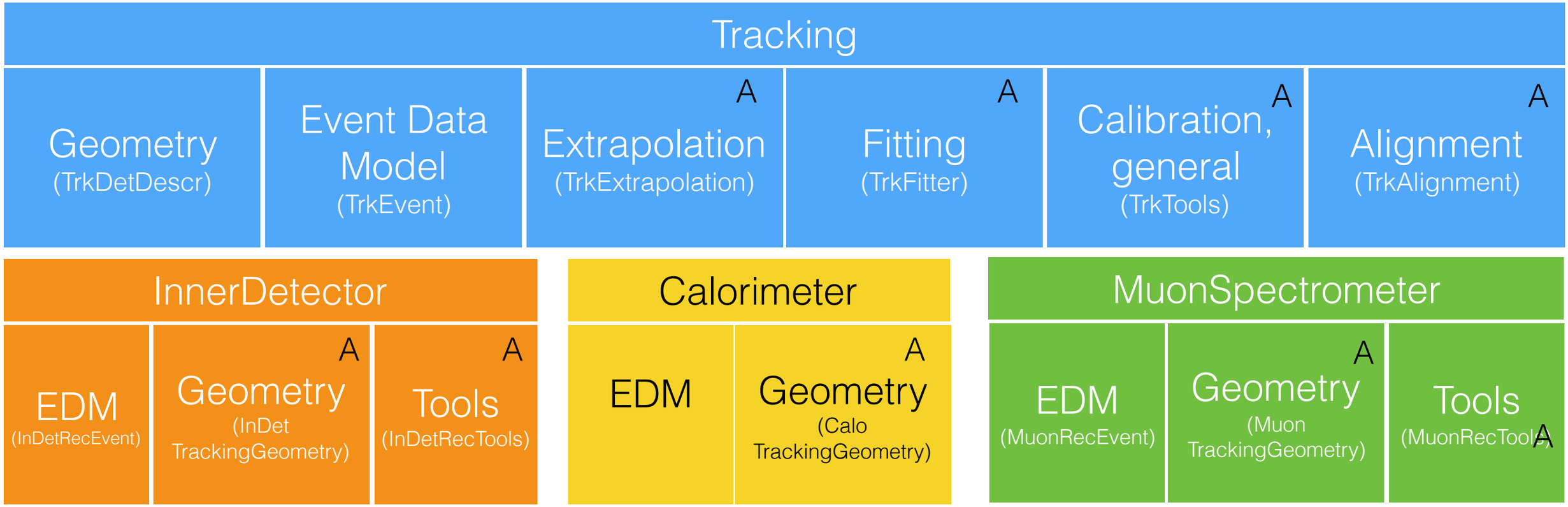


> 5 CPU gain, no loss of physics performance

A. Salzburger, CHEP2015



Extending the ATLAS Tracking SW structure



- Within Phase-2 upgrade we developed a fast detector prototyping
  - extended ATLAS tracking EDM/geometry with generic XML based builders
  - could easily run fast track simulation and refitting without actually building ATLAS (2014/15 in parallel a test study within FCC software context & DD4Hep binding)

Decoupling the ATLAS Tracking SW ?

Tracking

Geometry (TrkDetDescr)	Event Data Model (TrkEvent)	A Extrapolation (TrkExtrapolation)	A Fitting (TrkFitter)	A Calibration, general (TrkTools)	A Alignment (TrkAlignment)

no ATLAS dependency

ATLAS specifications & dependencies

InnerDetector

EDM (InDetRecEvent)	A Geometry (InDet TrackingGeometry)	A Tools (InDetRecTools)

Calorimeter

EDM	A Geometry (Calo TrackingGeometry)

MuonSpectrometer

EDM (MuonRecEvent)	A Geometry (Muon TrackingGeometry)	A Tools (MuonRecTools)

Introducing - The ATS project

Tracking					
Geometry (TrkDetDescr)	Event Data Model (TrkEvent)	G/A Extrapolation (TrkExtrapolation)	G/A Fitting (TrkFitter)	G/A Calibration, general (TrkTools)	G/A Alignment (TrkAlignment)

a-tracking-sw

Attempt to encapsulate the ATLAS Tracking software from ATLAS

currently restricted access

★ 0 | ⑂ 1 | SSH | HTTPS | ssh://git@gitlab.cern.ch:7999/ | GLOBAL

65 COMMITS | 5 BRANCHES | 0 TAGS | 13.21 MB | ADD CHANGELOG | ADD LICENSE | ADD CONTRIBUTION GUIDE

A Tracking Software (ATS) Project

This library is based on the track reconstruction software developed by the ATLAS Collaboration.

The main philosophy is to provide high-level track reconstruction modules that can be specified for detector technologies by simple extension.

- Event Data Model (EDM)
- Geometry
- track reconstruction tools

The library is attempted to build against Gaudi and Gaudi-Athena, while additional external dependencies are kept at a minimum.

double build philosophy

ATS - Decoupling from ATHENA

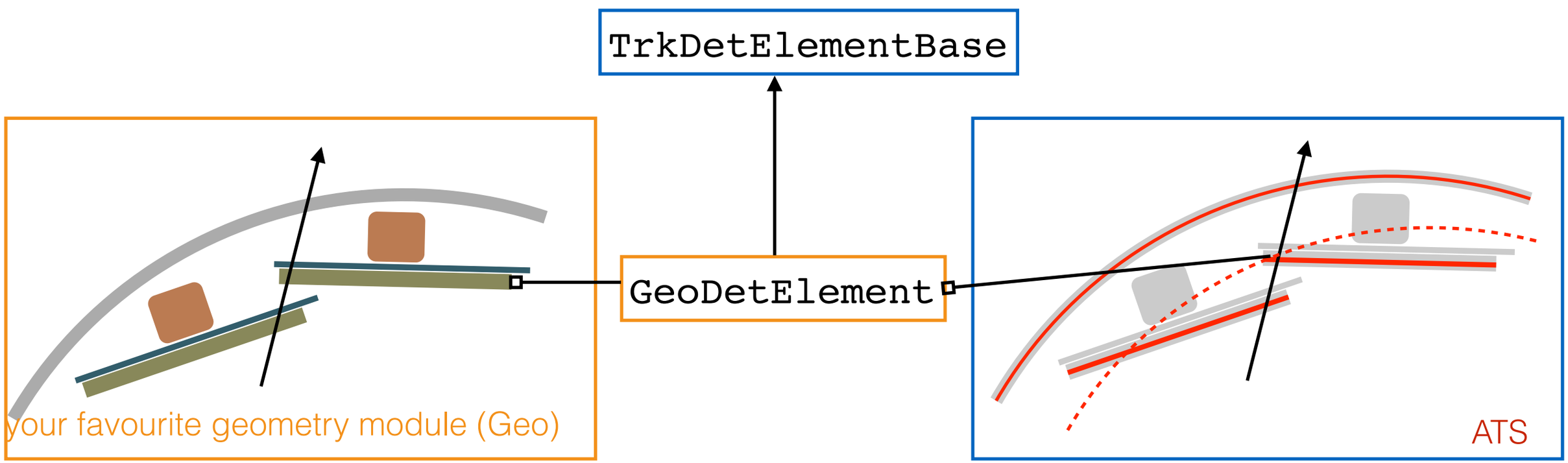
- Athena is based on Gaudi, but it's not Gaudi itself
  - first step was to decouple from Athena and AthenaBaseComponents
  - allows building blocks
    `AlgToolBase`
    `AlgorithmBase`
    `ServiceBase`

```
////////////////////////////////////////////////////////////////////////
// AlgToolBase.h, ATS project
////////////////////////////////////////////////////////////////////////

#ifndef BASECOMPSINTERFACES_ALGTOOLBASE_H
#define BASECOMPSINTERFACES_ALGTOOLBASE_H 1

// ATLAS build
#ifndef ATS_GAUDI_BUILD

// Athena version
#include "AthenaBaseComps/AthAlgTool.h"
#include "AtsBaseComponents/MsgMacros.h"

namespace Ats {
    typedef AthAlgTool AlgToolBase;
}

// GAUDI build
#else

#include "GaudiKernel/AlgTool.h"
#include "AtsBaseComponents/MsgBase.h"

namespace Ats {

    /** @class AlgToolBase
        simply extend the AlgTool class with a MsgBase */
    class AlgToolBase : public ::AlgTool, public MsgBase {
        public:
            /** Constructor */
            AlgToolBase( const std::string& type, const std::string& name, const IInterface* parent) :
                ::AlgTool(type, name, parent),
                MsgBase(msgSvc(), name)
            {}

            /** Virtual Destructor */
            virtual ~AlgToolBase() {}
    }
}

#endif // ATLAS-GAUDI-build
```

- Other dependencies
  - Eigen (part of LCG)
  - Identifier

ATS - Decoupling from ATLAS GeoModel

- ATLAS uses GeoModel library for detector description
  - not supported/used by other experiments
  - there are many other geometry models on the market, e.g. FCC used DD4Hep/TGeo geometry
  - single connection from Tracking software to geometry library via a `TrkDetElementBase`



alignment updates,
conditions data

ATS - Why ?

- LHC detector software has really been stress-tested
  - and I think we learned a lot, and we start working on Upgrade/FCC
  - however, our concepts are sometime > 30 years old !

APPLICATION OF KALMAN FILTERING TO TRACK AND VERTEX FITTING

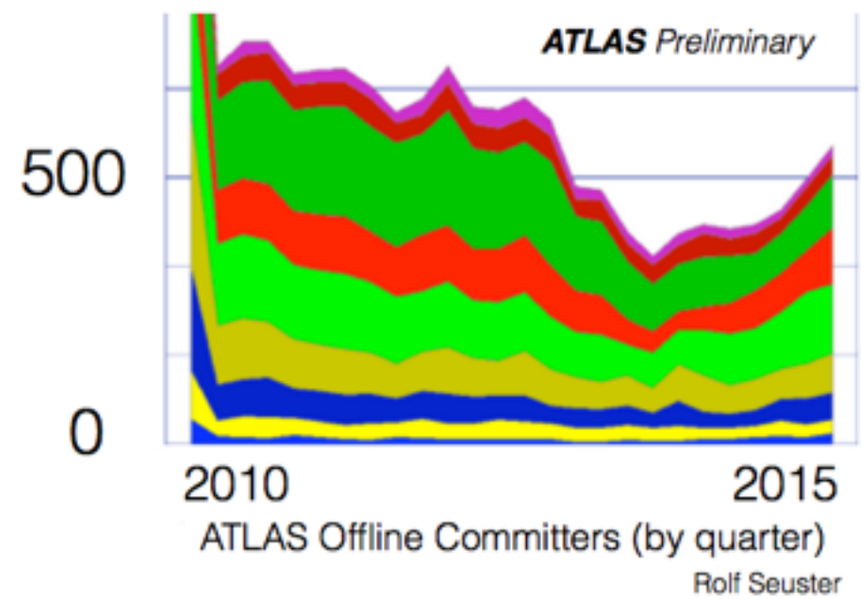
R. FRÜHWIRTH

Institut für Hochenergiephysik der Österreichischen Akademie der Wissenschaften, Vienna, Austria

Received 30 June 1987

- More importantly even

Algorithmic Code Evolution



- Highest investment in algorithmic code — O(100M$) for LHC experiments
  - Vast majority of offline packages

Graeme Stewart, Evolution of HEP SW, CHEP2015

ATS - Status, plans and timescales

- Prototype for geometry building decoupled from ATHENA
  - builds against Gaudi (Cmake) and Athena (cmt) : done for core classes
  - next step: dummy detector building in Gaudi/ATHENA and compare: ongoing
  - geometry building from DD4Hep in FCC context : started
- Integration of Extrapolation engine and basic EDM
  - proof of extrapolation concept and little fast track sim: started
- Integration of Truth tracking module: not started
- Integration of KalmanFitter: started

demonstrator for FCC Rome workshop April 2016 planned
including Geometry - Fast Simulation - Truth Tracking - Track Fitting

Some concluding remarks

- Is it time to think about reconstruction toolkits ?
  - simulation (Geant) has done this 30 years ago during LEP

    it was simply not possible for experiments to write their full simulation programs (needed a toolkit)

- (Track) reconstruction will be one of the most challenging aspects
  - already for Phase-2 upgrade
  - even more for FCC(-hh)

- New concepts, players and R&D is needed
  - and these are also coming in !

    e.g. Machine learning approaches for HEP:
    IM LHC Maching Learning WG Meeting, tomorrow

15:55 - 16:15	Machine Learning Challenges in HEP: Upcoming Tracking Challenge *20'* Speaker: Andreas Salzburger (CERN)	

Some further reading

ATLAS RTF report **ATL-SOFT-2003-010**

ATLAS Tracking Geometry **ATL-SOFT-PUB-2007-004**

ATLAS Tracking Event Data Model **ATL-SOFT-PUB-2006-004**, **ATL-SOFT-PUB-2007-003**

Common Tools **ATL-SOFT-PUB-2007-007**

Extrapolation & STEP propagator **ATL-SOFT-PUB-2007-005** **ATL-SOFT-PUB-2008-003**

Fast Track Simulation **ATL-SOFT-PUB-2008-001**

Backup slides

ATLAS Tracking Geometry - summary

- ATLAS Tracking Geometry is a geometry model designed around a few basic concepts:
  - core component **`Trk::Surface`** class (all nodes extend, contain surfaces)
  - embedded navigation through interlinking and volume glueing

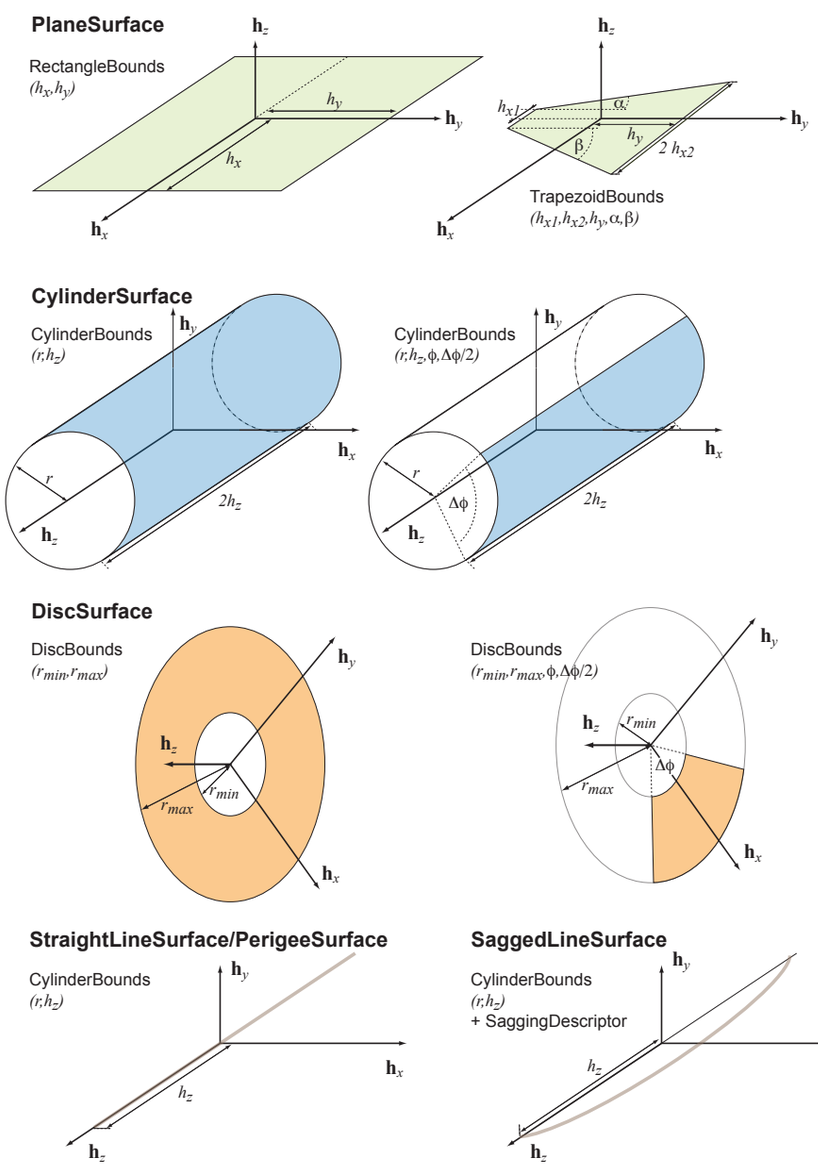
- **`Trk::Surface`** class builds the core of the geometry and the Tracking Event Data Model (EDM)
  - track parameterisations are done on with respect to a surface
  - extrapolation, track fitting, measurement parameterisations

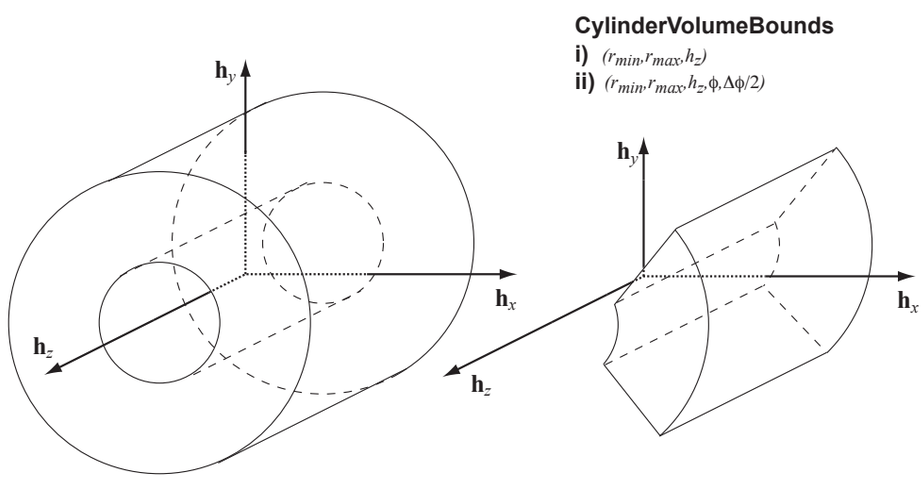
- The Tracking Geometry comes with an Extrapolation tool
  - uses the embedded navigation to move through the geometry
  - uses optimised propagation & material effects integration (point-like, dense)

The Tracking Geometry

- TrackingGeometry is a simplified geometry for reconstruction purpose
  - it was one of the first modules written for new Tracking

- It's core components are is the Trk::Surface class
  - Surface
  - Layer extend Surface
  - Volume is a collection of Surfaces

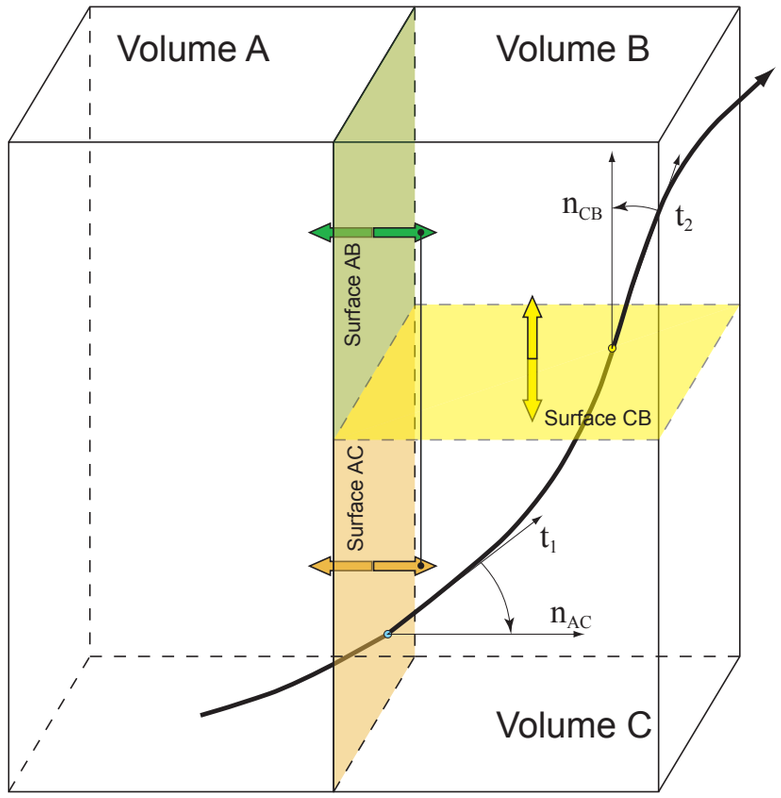




The Tracking Geometry - navigation

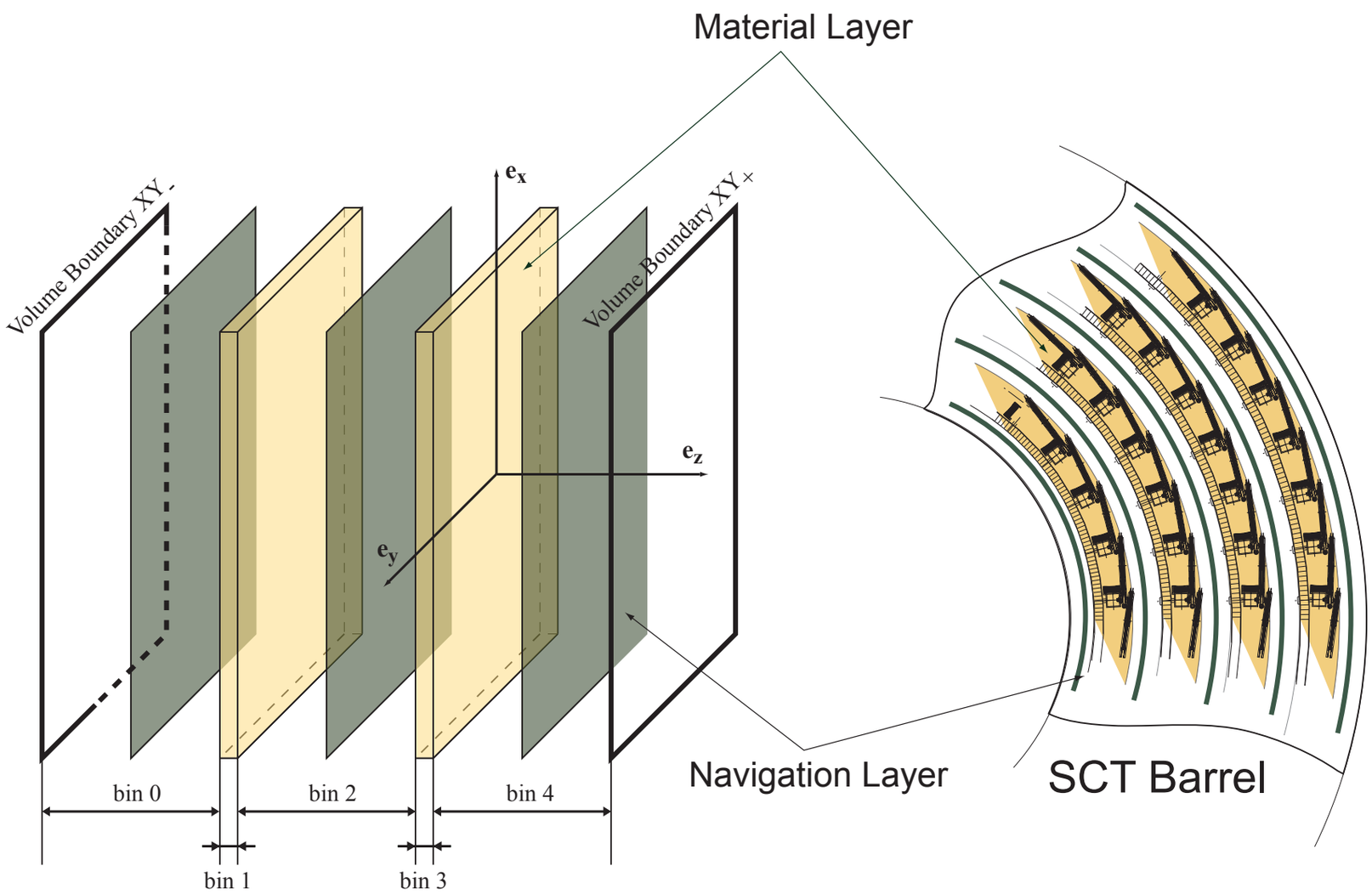
- TrackingGeometry has two jobs
  - provide the detector surfaces and navigation through the detector
  - provide the material

- Part 1 - Navigation
  - navigation is provided by a “glueing” mechanism
  - this is the key (and complicated part of the geometry building)



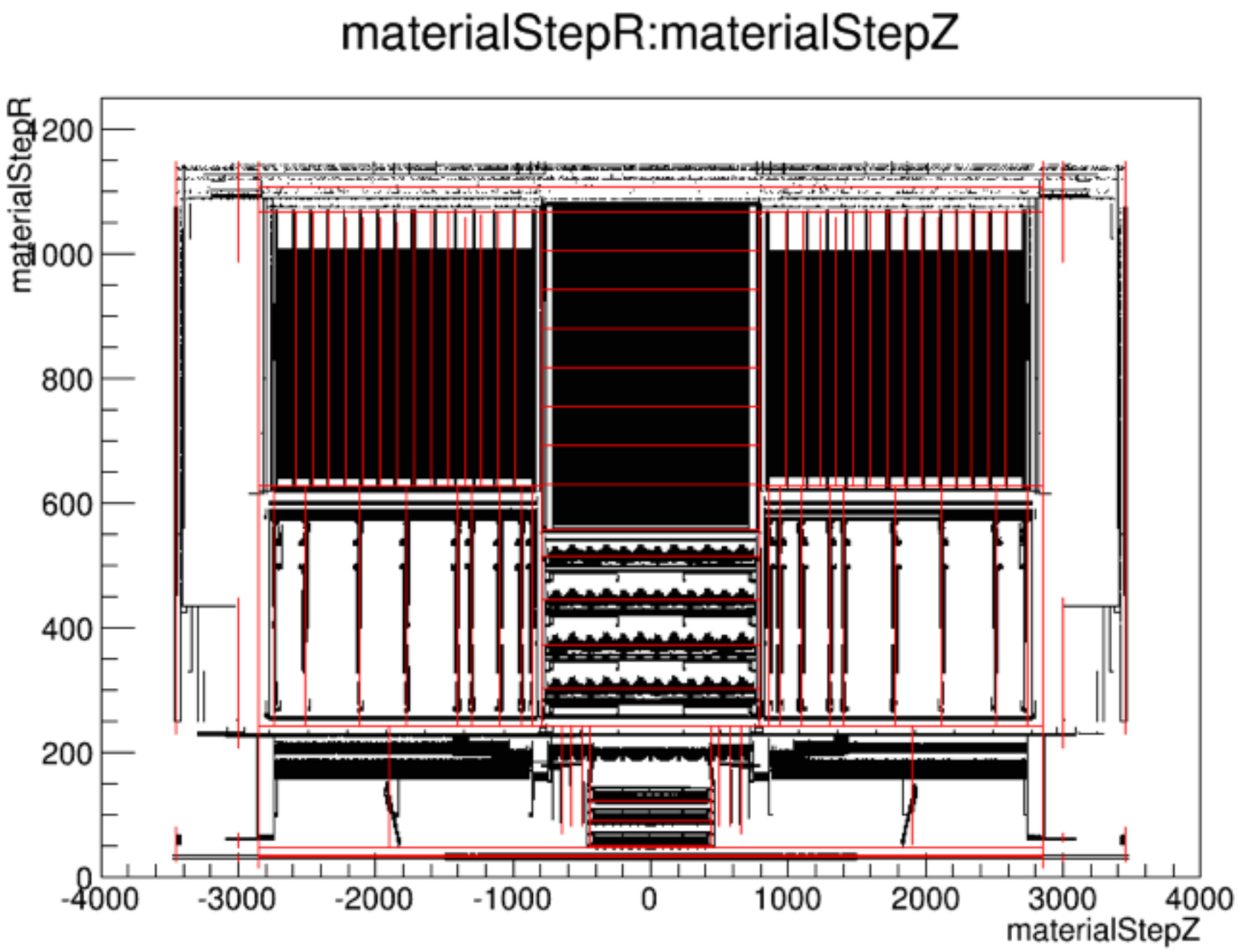
TrackingGeometry - navigation modes

- The TrackingGeometry knows several navigation modes
  - Static : that's the ID, everything is set up at building (let's talk about this)
  - Dense : these are dense volumes mostly for the call
  - Detached/Unordered: this is for the muon System



TrackingGeometry - simplification of nodes

- Layers are described as cylinders and disks
  - sensitive material in CURRENT tracking geometry only on those
  - even though the modules are perfectly known to the TrackingGeometry



ATLAS Tracking SW EDM - TrackParameters

- TrackParameters have been entirely reworked during LS1
  - among with much of the ATLAS **`Tracking`** code (95% of code line savings)

Package	C++	C/C++ Header	C++	C/C++ Header
TrkParameterBase	63	561	11	214
TrkParameters	1715	602	0	52
TrkNeutralParameters	1425	663	0	48
ExtendedTrkParameterBase	0	295	0	0
ExtendedTrkParameters	1412	514	0	0
ExtendedTrkNeutralParameters	1416	514	0	0
Total	**6031**	**3149**	**11**	**266**

ATS binding geometry link - TrkDetElementBase

- Connects any favourite geometry module to ATS

```
/** Identifier */
virtual Identifier identify() const = 0;

/**Identifier hash*/
virtual IdentifierHash identifyHash() const = 0;

/**Return local to global transform*/
virtual const Amg::Transform3D & transform() const = 0;

/**Return local to global transform associated with this identifier*/
virtual const Amg::Transform3D & transform(const Identifier& id) const = 0;

/**Return surface associated with this detector element*/
virtual const Surface & surface () const = 0;

/**Return surface associated with this identifier, which should come from the
PrepRawData object
(i.e. Surface s = PRD.detElement().surface( PRD.identify() ).
This is only really relevant for the TRT (where there are several
surfaces per detector element). For other detector elements it will just
return surface() ... the Identifier is ignored.
*/
virtual const Surface & surface (const Identifier& id) const = 0;

/** Returns the full list of all detection surfaces associated to this detector element */
virtual const std::vector<const Surface*>& surfaces() const = 0;
```