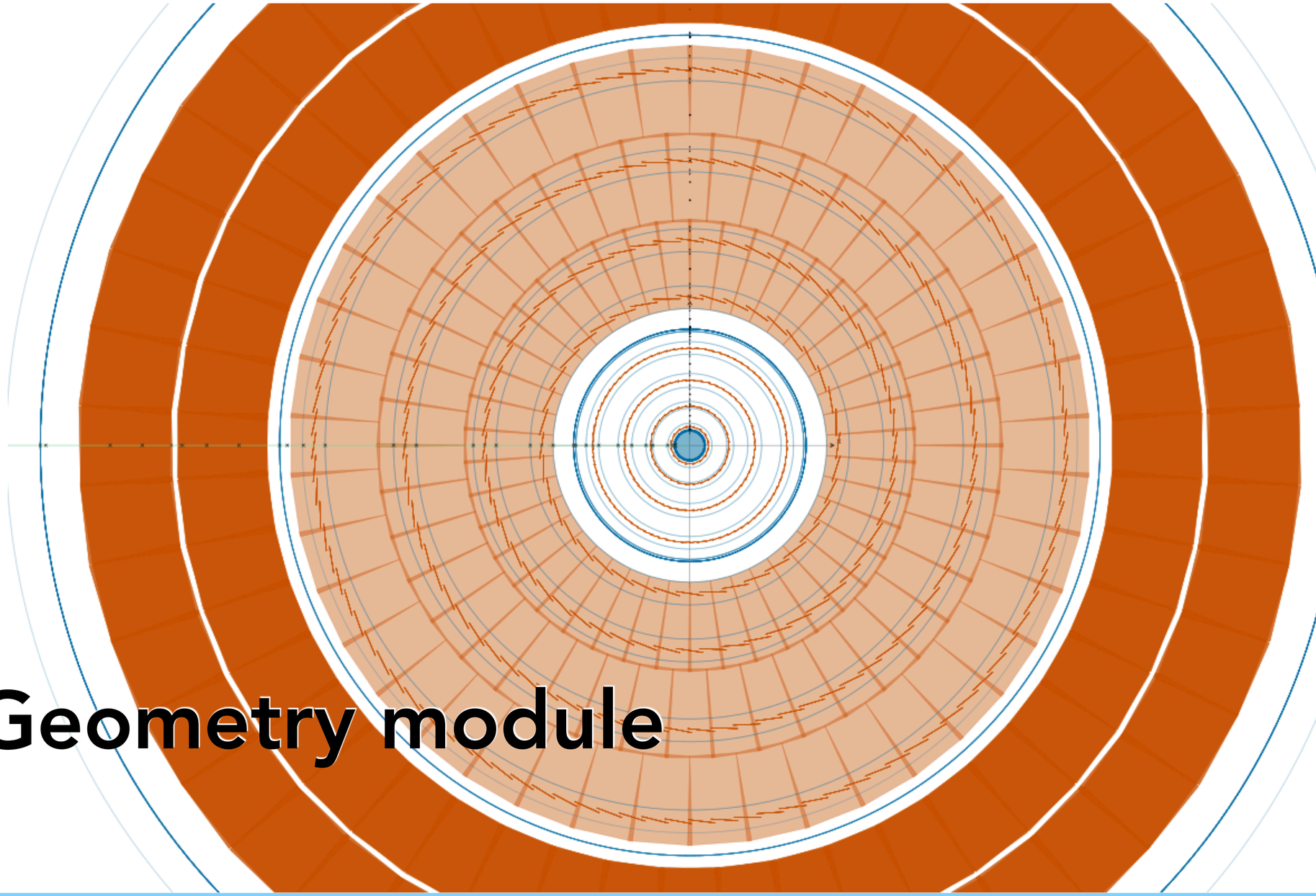
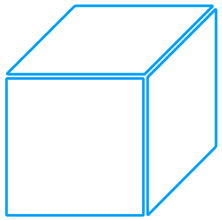
ACTS Workshop, IJCLab, Orsay, Nov 8, 2023



Geometry module

A. Salzburger (CERN) for the ACTS project

Selected Core modules



Geometry/Detector*
(Surface based geometry)

Plugins to DD4hep, TGeo, etc.

Event Data Model
target track reconstruction

backend separation
with different I/O models

Propagation
parameter + covariance
transport through
magnetic field

Seeding
Seed finding with
Triplet seeder,
OrthogonalSeedFinder

Track Fitting
parameter estimation
with Kalman Filter,
GSF, GX2F**

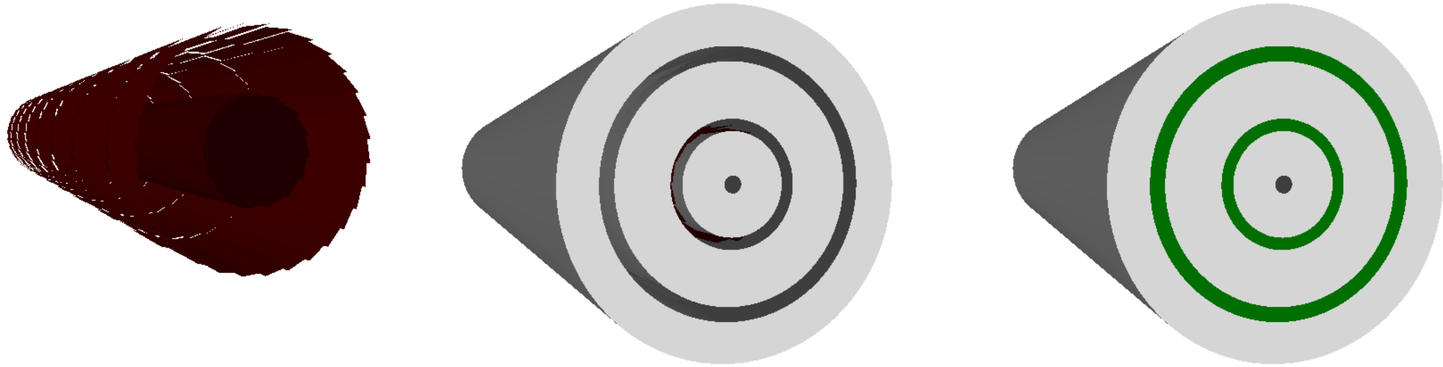
Combinatorial track finding
Combinatorial Kalman Filter
for track finding

Vertex finding + fitting
Iterative, multi variant
primary vertex finders
and fitters

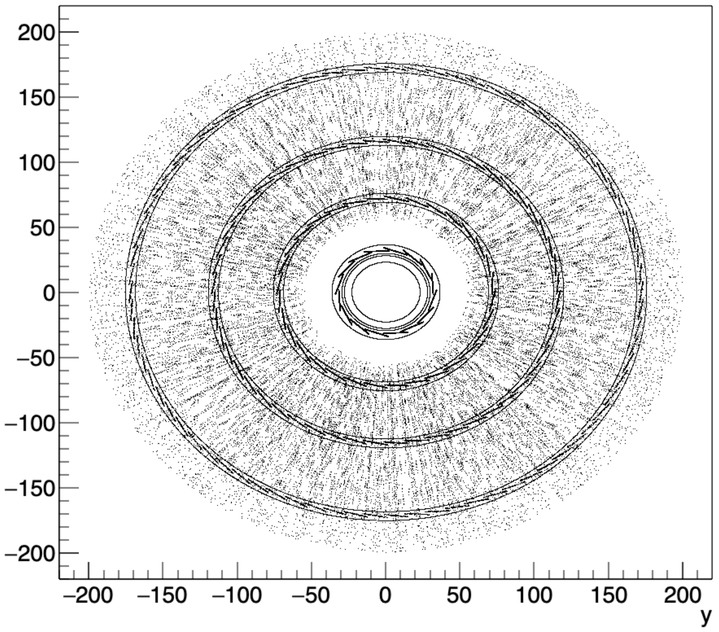
Detector alignment
KF based alignment
functionality

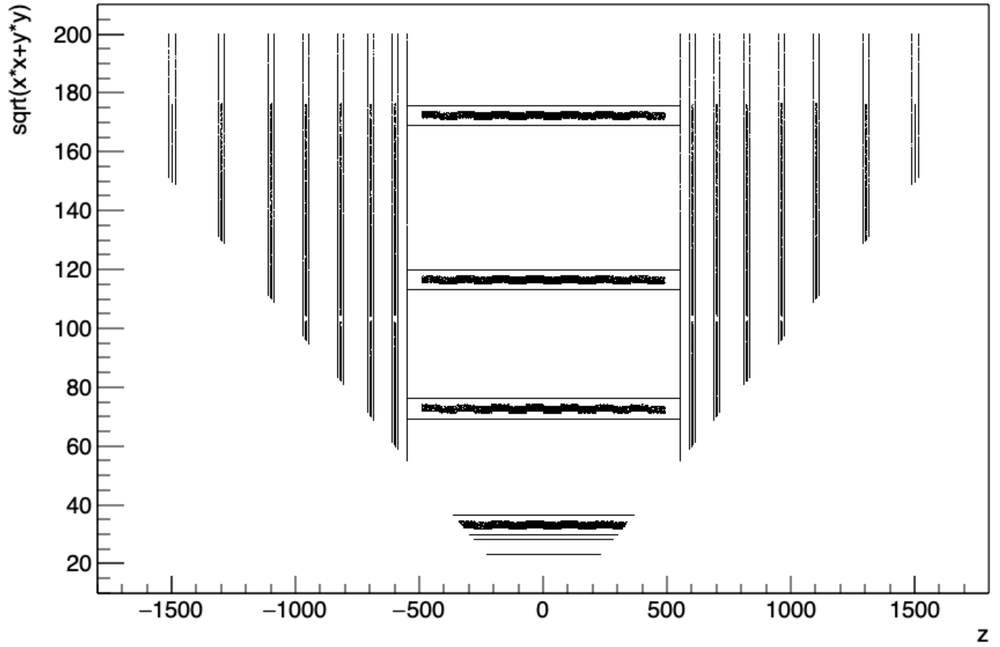
***new geometry model to be deployed in 2023, **GX2F prototype to come 2024**

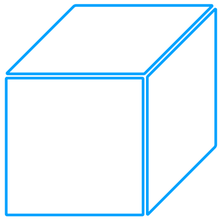
Geometry + Navigation



Surface based geometry model, optimised for reconstruction,
can be **connected to standard HEP geometry modellers**





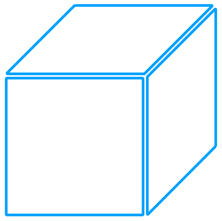


Geometry + Interfacing

connected to standard HEP geometry modellers

Detailed geometry model,
e.g. DD4hep, TGeo, GeoModel, etc.

ACTS geometry model
with built-in navigation

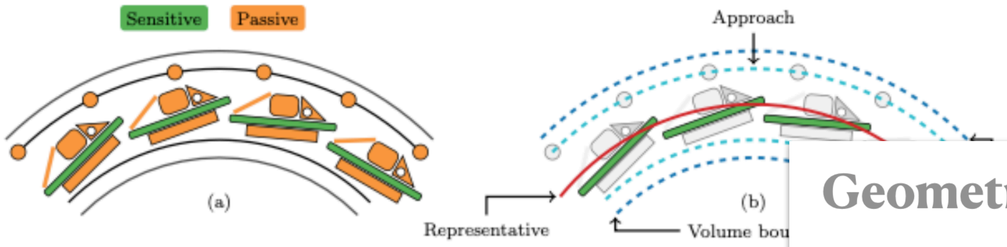


New `Acts::Detector` model

detray geometry was developed w/o layers

- **this worked,** developed a new, alternative geometry model for Acts/Core as well

Geometry | Surface - Layer - Volumes



Layers with sub structure have a volume-like feel already

```
///  Return the abstract volume that represents the layer
///
/// @return the representing volume of the layer
const AbstractVolume* representingVolume() const;

/// return the LayerType
LayerType layerType() const;
```

can be r

To repr

naviga

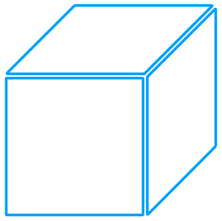
Nov. 2022

Geometry | Surface - Layer - Volumes

Consequence: `Navigator` code is complicated & brittle

	Navigator
LOC	760
Comments	452

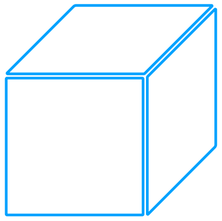
Suggestion: Remove Layer concept & only glue volumes with internal structure



New `Acts::Detector` model - glossary

This geometry is currently still under the `Experimental` namespace

`Acts::Surface`	`Acts::Surface`	Surface objects are unchanged, allows client code to be untouched
`Acts::Layer`		Layer objects do not exist anymore, they are represented by volumes
`Acts::TrackingVolume`	`Acts::Experimental::DetectorVolume`	Double serving of volumes as containers or navigation volumes omitted
`Acts::BoundarySurfaceT<Acts::TrackingVolume>`	`Acts::Experimental::Portal`	Portal objects are not templated anymore, they are holder classes of surfaces and volume switches
`Acts::TrackingGeometry`	`Acts::Experimental::Detector`	Portal objects the top level entry point that will guide into the root volumes



Acts::Detector features (1)

Navigation paradigm shift

- **DetectorVolume** carries a navigation delegate that provides candidate surfaces (i.e. sensitive, passive, portal surfaces) to the propagator
- Allows to switch navigation strategy in different parts of the detector
- Used for ATLAS Muon System geometry templating (see Tomohiros' talk tomorrow)

Detector building const correctness

- Dedicated care for const correctness in the detector building (TrackingGeometry building is a bit bit of a mess in this respect)
- Strategy: non-const objects until the **final Detector class** is built, only c**onst access** afterwards

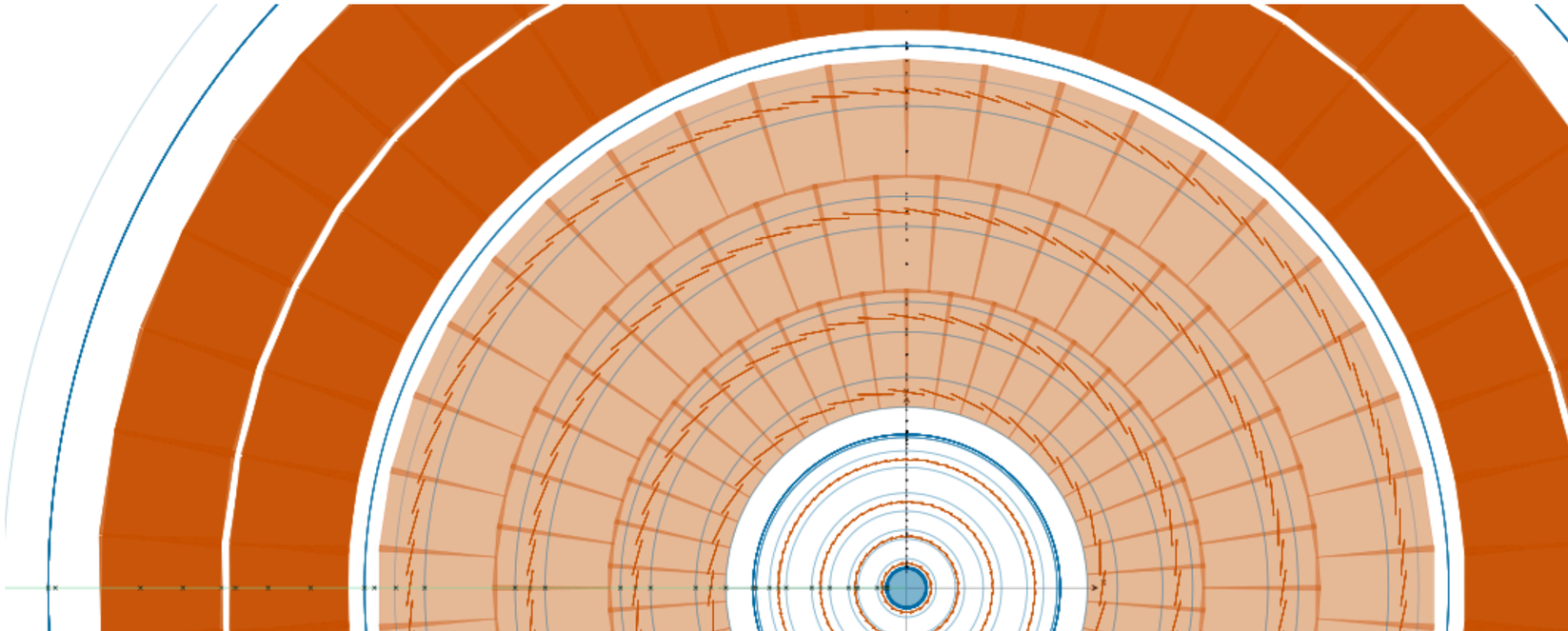
`Acts::Detector` features (2)

Json writing and reading

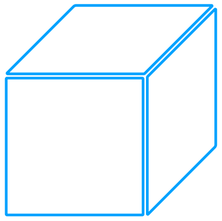
- **Full detector** can be written to Json file and back, including navigation structure
- Possible through indexed based navigation structures

Json export to detray

- Dedicated `toJsonDetray(...)` function to write the detector in detray format



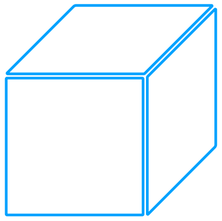
ODD exported to detray json, read back in displayed with actsvg



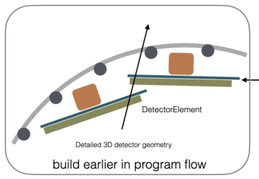
Acts::Detector features (2)

- **Detector** folder is relatively well unit tested, even the building code

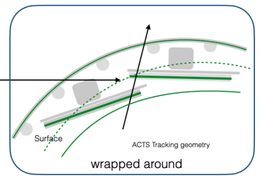
interface	7	7	0	0	100.00%
Blueprint.hpp	22	16	5	1	72.73%
CylindricalContainerBuilder.hpp	2	2	0	0	100.00%
DetectorBuilder.hpp	2	2	0	0	100.00%
DetectorVolume.hpp	35	26	9	0	74.29%
DetectorVolumeBuilder.hpp	2	2	0	0	100.00%
GeometryIdGenerator.hpp	4	4	0	0	100.00%
KdtSurfacesProvider.hpp	46	28	17	1	60.87%
LayerStructureBuilder.hpp	5	5	0	0	100.00%
MultiWireStructureBuilder.hpp	2	2	0	0	100.00%
Portal.hpp	1	1	0	0	100.00%
ProtoBinning.hpp	24	19	5	0	79.17%
ProtoDetector.hpp	5	5	0	0	100.00%
VolumeStructureBuilder.hpp	1	1	0	0	100.00%



Acts::Detector building process



build earlier in program flow



wrapped around

Detailed geometry model,
e.g. DD4hep, TGeo, GeoModel, etc.

ACTS geometry model
with built-in navigation

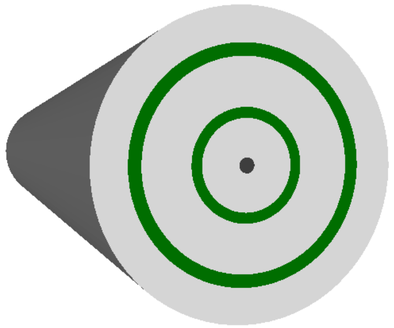
Translation of objects from geometry model,
e.g. DD4hep

from one source, but not necessarily

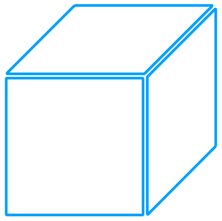
Logic of how to build/group
e.g. DD4hep

Detector Blueprint

Instruction set how to build the detector



Blueprint to be written to .json

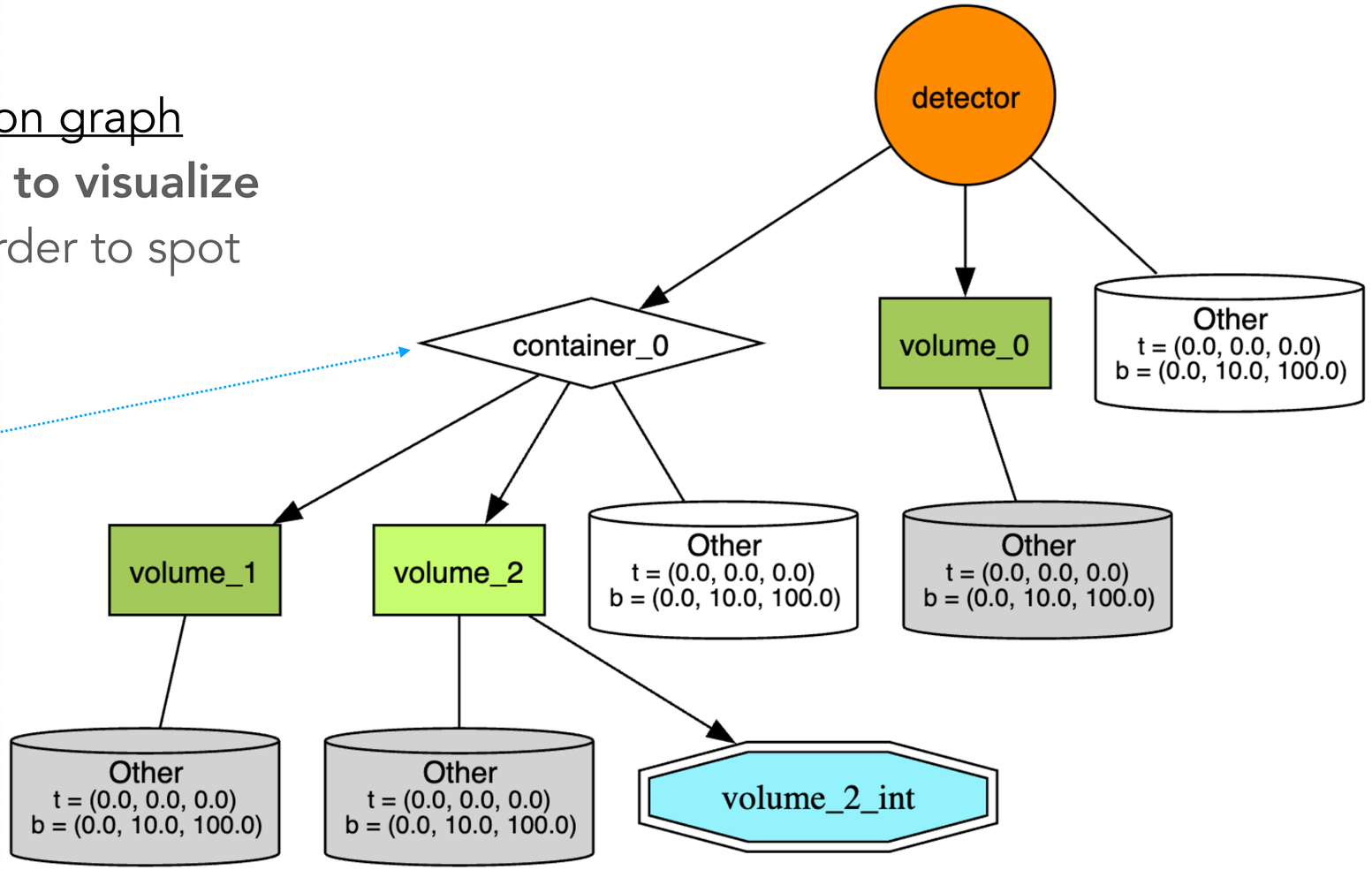


Blueprint visualisation

Blueprint is an instruction graph

- **Added functionality to visualize** before building, in order to spot problems

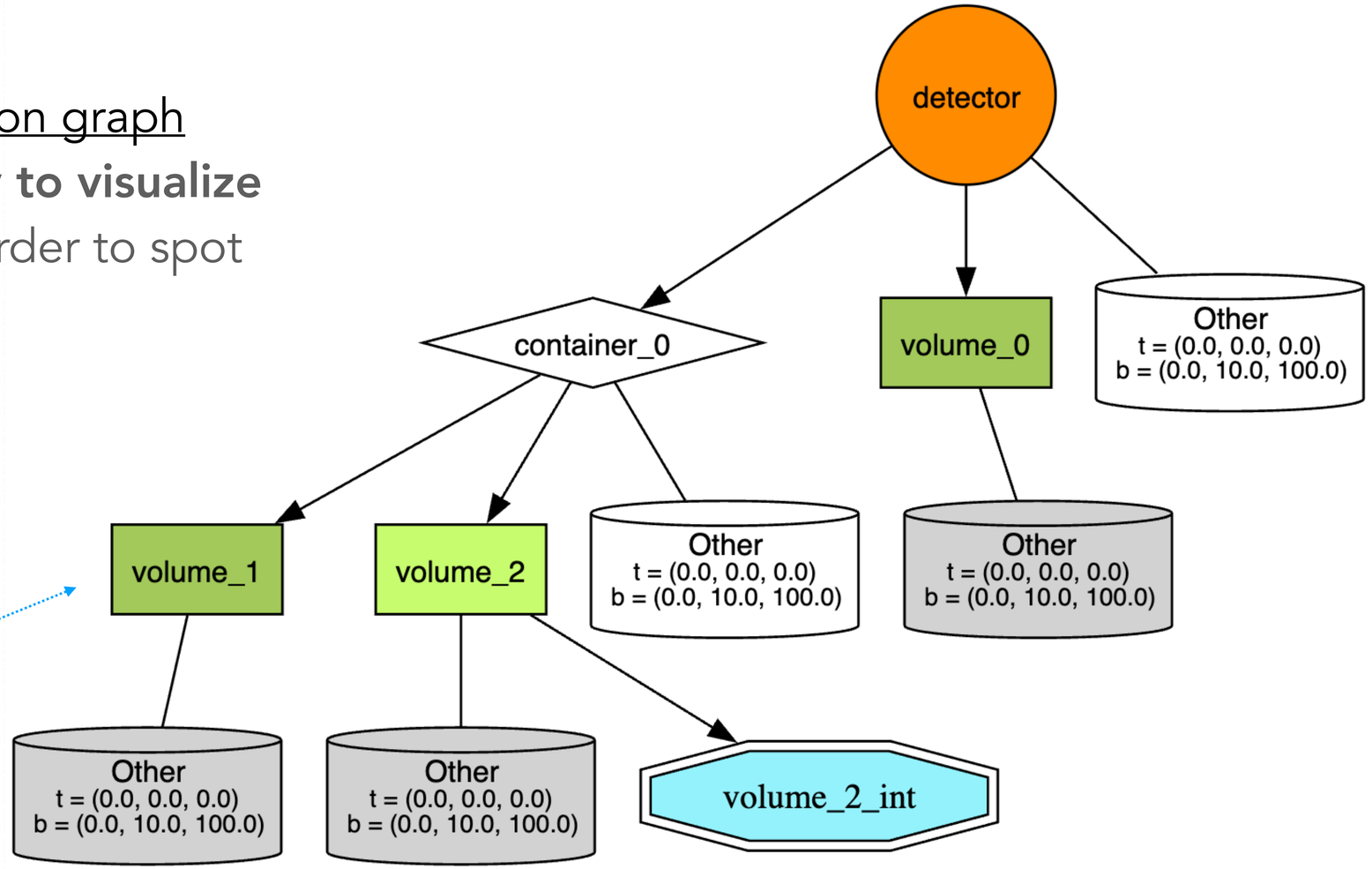
non-coloured nodes are virtual containers



Blueprint visualisation

Blueprint is an instruction graph

- **Added functionality to visualize** before building, in order to spot problems

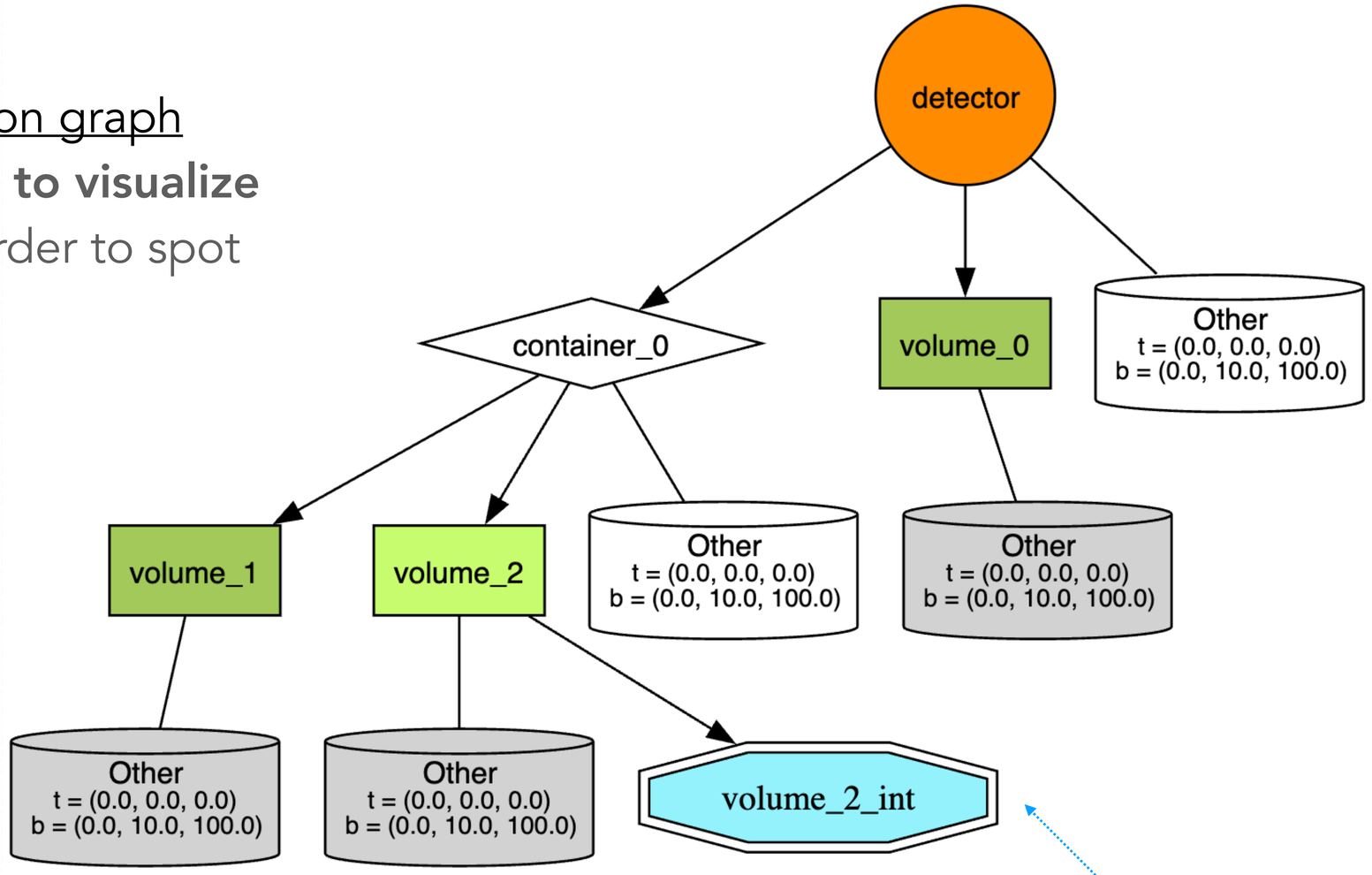


gap volumes can be automatically Created

Blueprint visualisation

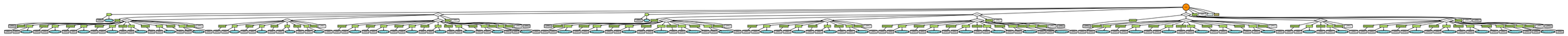
Blueprint is an instruction graph

- **Added functionality to visualize** before building, in order to spot problems

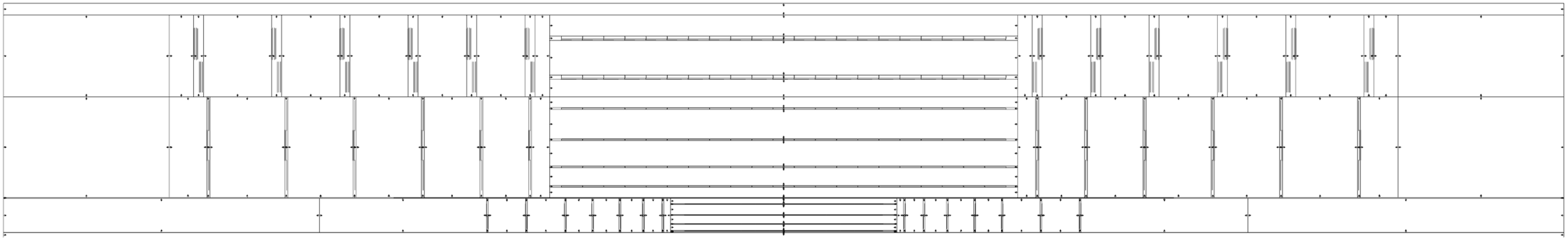


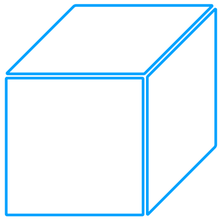
Visualisation

ODD building blueprint from DD4hep:



Resulting ODD detector





Building the ODD

Extensively use the variant parameters plugin from DD4hep, purely in XML

A container definition:

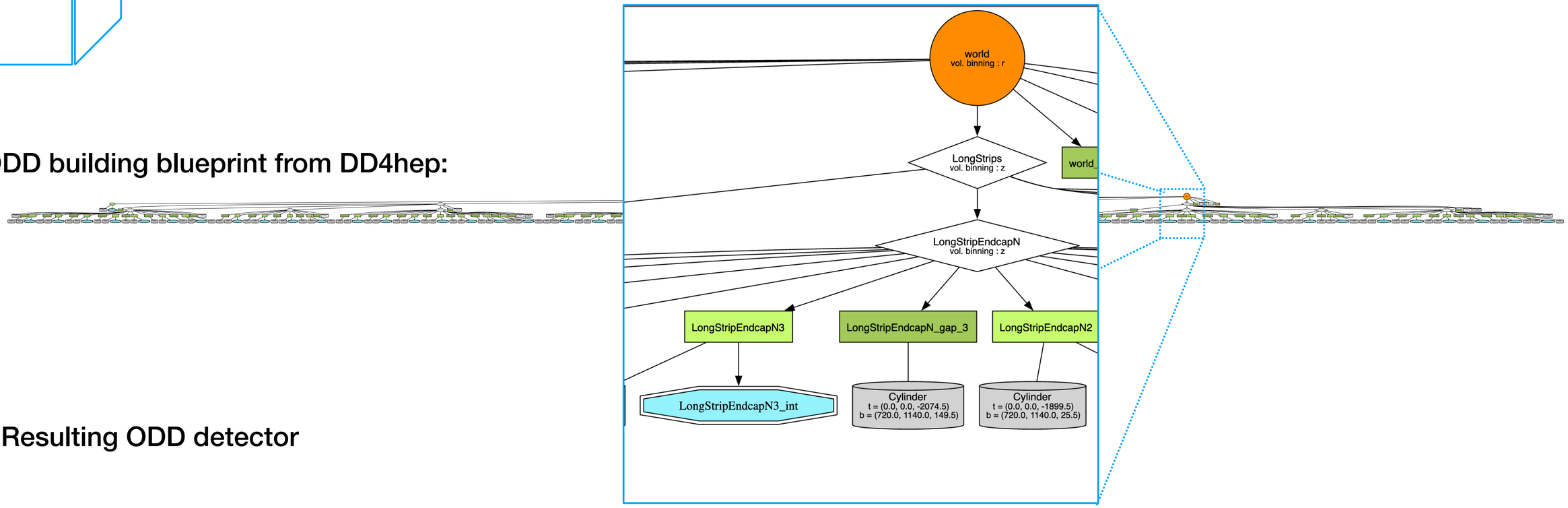
```
    <plugin name="DD4hep_ParametersPlugin">
      <argument value="PixelBarrel"/>
      <argument value="layer_pattern=PixelLayer\d"/>
      <argument value="acts_container: bool = true"/>
      <argument value="acts_container_type: int = 3"/>
      <argument value="acts_container_bvalues_n: int = 3"/>
      <argument value="acts_container_bvalues_0: double = pix_env_rmin"/>
      <argument value="acts_container_bvalues_1: double = pix_env_rmax"/>
      <argument value="acts_container_bvalues_2: double = pix_b_dz"/>
      <argument value="acts_container_binning: str = r"/>
    </plugin>
```

A layer volume:

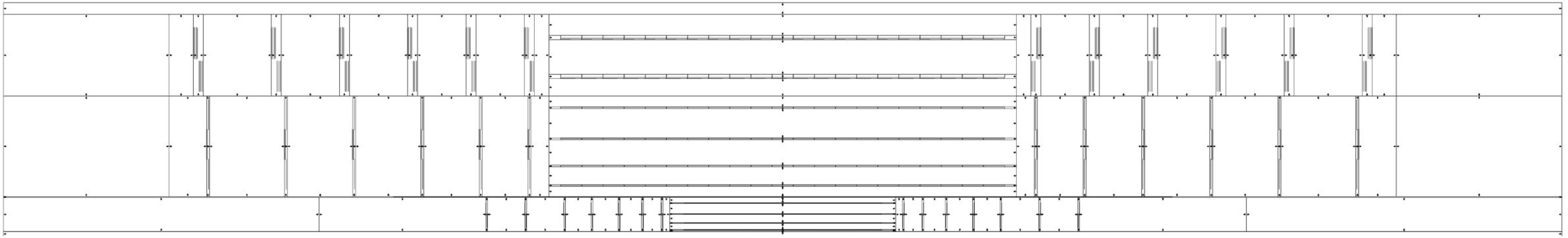
```
    <plugin name="DD4hep_ParametersPlugin">
      <argument value="/world/OpenDataDetector/Pixels/PixelBarrel/PixelLayer1"/>
      <argument value="acts_volume: bool = true"/>
      <argument value="acts_volume_type: int = 3"/>
      <argument value="acts_volume_internals: bool = true"/>
      <argument value="acts_volume_internals_type: str = layer"/>
      <argument value="acts_volume_internals_measure: str = z,r,r"/>
      <argument value="acts_volume_internals_clearance: double = pix_b_clear"/>
      <argument value="acts_surface_binning_dim: int = pix_b_sf_b"/>
      <argument value="acts_surface_binning_z_n: int = pix_b_sf_b_z"/>
      <argument value="acts_surface_binning_z_exp: int = pix_b_sf_e_z"/>
      <argument value="acts_surface_binning_phi_n: int = pix_b1_sf_b_phi"/>
      <argument value="acts_surface_binning_phi_exp: int = pix_b_sf_e_phi"/>
    </plugin>
```

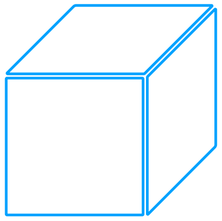
Visualisation

ODD building blueprint from DD4hep:



Resulting ODD detector



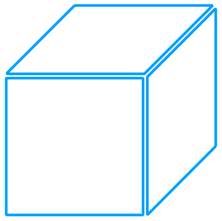


New detector strategy - current code organisation

Surfaces	Geometry	Detector	Navigation
Surfaces Surface Bounds	Volume Bounds GeometryContext GeometryIdentifier Extent	Portals DetectorVolume Detector Detector building	Navigation deletages
	Layers BoundarySurfaceT<> TrackingVolume TrackingGeometry TrackingGeometry building		

TrackingGeometry schema

Detector schema



New detector strategy - **suggested** code organisation

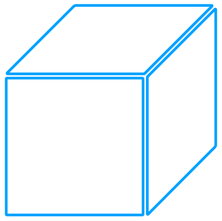
TrackingGeometry	Volumes	Surfaces	Geometry	Detector	Navigation
Layers BoundarySurfaceT<> TrackingVolume TrackingGeometry TrackingGeometry building	Volume Bounds	Surfaces Surface Bounds	GeometryContext GeometryIdentifier Extent	Portals DetectorVolume Detector Detector building	Navigation deletages

TrackingGeometry schema

Detector schema

Benefit

- This would allow to build TrackingGeometry/Detector model with a ACTS_DETECTOR_MODEL=On flag
- See also discussion afterwards with contribution from Andreas Stefl



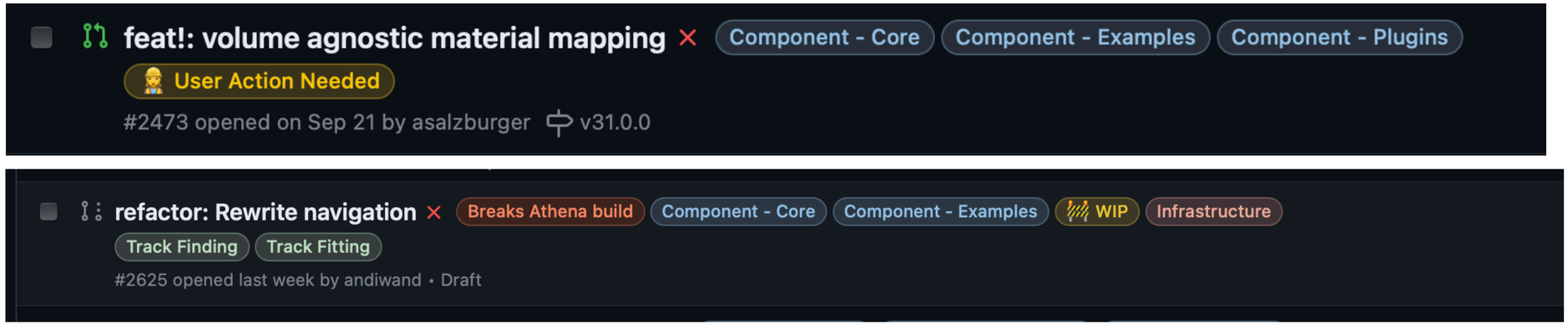
Acts::Detector status & plans

Infrastructure

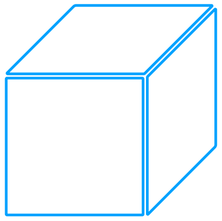
- Full building infrastructure **for cylindrical detectors** in place (and merged)
- **DD4hep** translation **for cylindrical detectors** available (and merged)

ODD as test template:

- branch with updated OpenDataDetector (with variant description) open, plan to make a MR to ODD in O(week)
- Next steps:



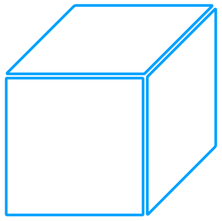
+ include infrastructure for cubic Detectors, branch exists, should be there in O(1 month)



`transform()`: copy vs. const reference

Detailed geometry model,
e.g. DD4hep, TGeo, GeoModel, etc.

ACTS geometry model
with built-in navigation



`transform()`: copy vs. const reference

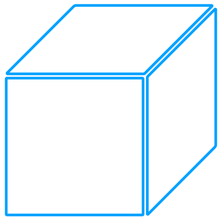
```
/// @class DetectorElementBase
///
/// This is the default base class for all tracking detector elements
/// with read-out relevant information. It provides the minimal interface
/// for the Acts proxy mechanism for surfaces, i.e. surfaces in the
/// Tracking geometry representing actual detection devices
///
class DetectorElementBase {
 public:
  DetectorElementBase() = default;
  virtual ~DetectorElementBase() = default;

  /// Return the transform for the Element proxy mechanism
  ///
  /// @param gctx The current geometry context object, e.g. alignment
  virtual const Transform3& transform(const GeometryContext& gctx) const = 0;
```

Alignment:
aligned position
can not be generated
on the fly

Wire chambers:
surfaces need to
be precomputed

Preformance
No copy/allocation
needed



`transform()`: copy vs. const reference

PR #2527

This PR tests how much performance loss we would experience when changing the return object of `Surface` objects from

```
const Transform& transform(const GeometryContext& gctx) const;
```

to

```
Transform transform(const GeometryContext& gctx) const;
```

This change would allow us to create surfaces on the fly, (or at least their transforms) for e.g. Drift straws, etc.

Initial tests - running the propagation test only, which is relatively highly effected by this - shows.

Propagation test

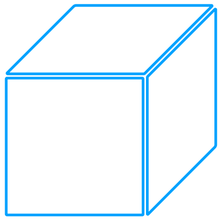
without this change

```
09:57:30    Sequencer      INFO      Average time per event: 19.184849 ms/event
```

with this change

```
09:56:55    Sequencer      INFO      Average time per event: 20.279401 ms/event
```

It indicates a 5% penalty in this workflow.



`transform()`: copy vs. const reference

PR #2527

Truth Tracking

```
14:07:33    Sequencer      INFO      Average time per event: 44.465655 ms/event
```

vs.

```
14:07:42    Sequencer      INFO      Average time per event: 45.216276 ms/event
```

So, closer to 2 % effect there, already interesting, I will do a full chain run as well.

- Next steps:
  Test with ODD full chain example and then we should decide on it in a future developers meeting.