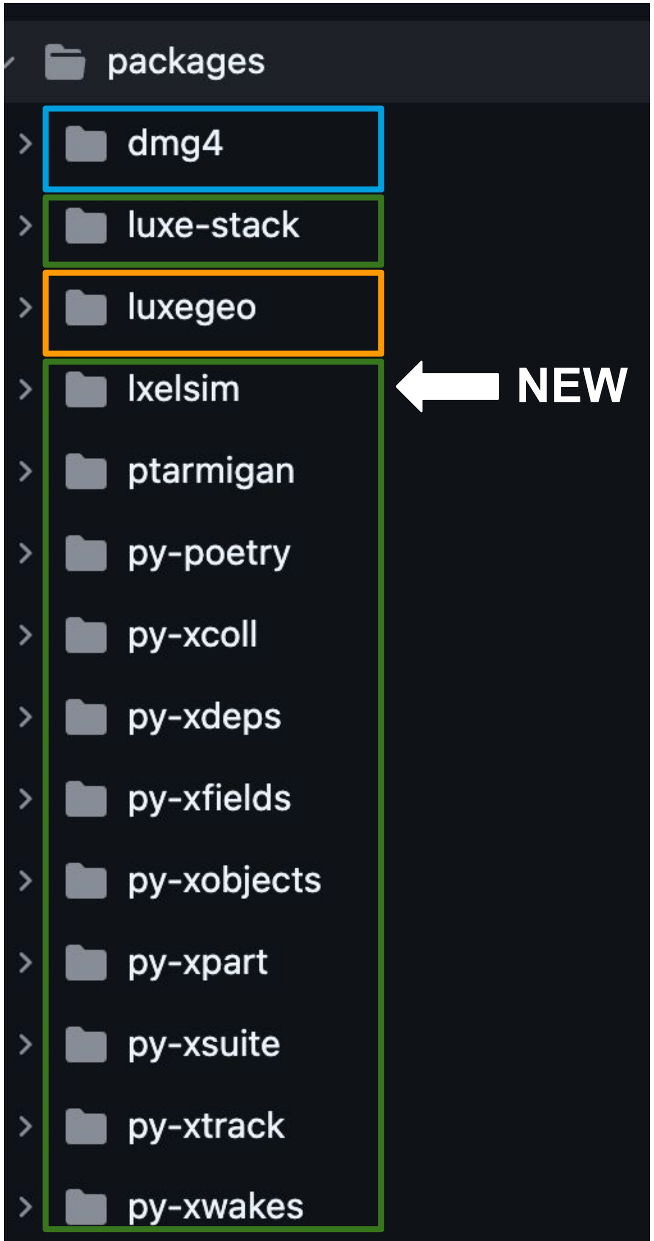
DD4hep-based LUXE model and simulation

A rough preview of what is to come, and plans

Federico Meloni (DESY),
including work from several others!

LUXE SAS meeting
26/01/2026

Recap of the software release status



packages
- dmg4
- luxe-stack
- luxegeo
- lxelsim
- ptarmigan
- py-poetry
- py-xcoll
- py-xdeps
- py-xfields
- py-xobjects
- py-xpart
- py-xsuite
- py-xtrack
- py-xwakes

🟩	Done and working
🟧	Prototype included
🟦	In the queue

Focusing next on collecting/porting over the prototype **dd4hep geometries** that were setup for tracker and ECALs

TODAY

What else do we want or need to include?

Including a package is as easy as writing an additional spack recipe

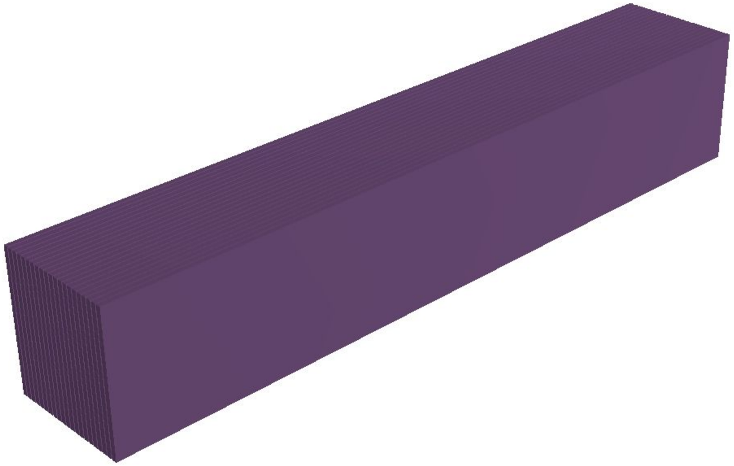
- Get in touch!
- Make a PR directly

DD4hep LUXE ECAL-P

Initial import of ECAL-P DD4hep model by Shan in PR 6

Standalone (floating) detector based on geometry used in TB2025

- Known to be not fully consistent with standalone Geant4 model
- 20 layers / 20 X_0
  - Pure tungsten absorber,
  - Compact silicon sandwich (CSIS): carbon fibre support, regular glue, fanout film, conductive glue, Si wafer, conductive glue, High-voltage film
  - Air gap
  - Per-slice module placement to implement the dead areas (6 CSIS)

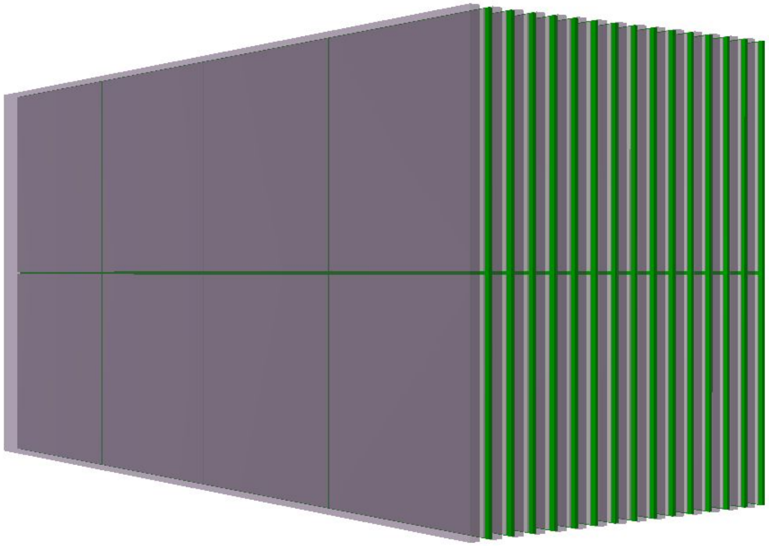


DD4hep LUXE ECAL-E

Initial import of ECAL-E DD4hep model by Shan in PR 6

Standalone (floating) detector based on geometry used ECAL-NPOD study in 2024-25

- Pixelated readout added at module level
- 15 layers /18 X_0
  - Pure tungsten absorber,
  - Carbon fibre support
  - Active sensor unit (ASU): High-voltage film, Conductive glue, Si wafer, Conductive glue, PCB board, On board ASIC
  - Air gap
- This variant: 2 ASUs and 8 wafers



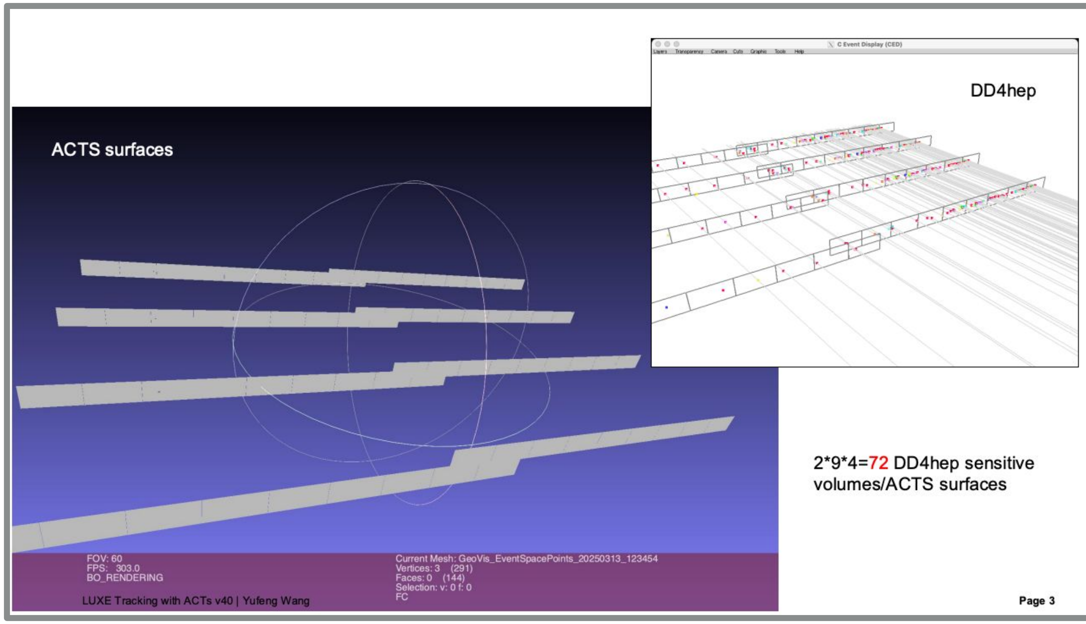
DD4hep LUXE tracker

Initial import from Yee and Michal’s studies

Standalone (floating) detector based on TDR geometry

Very simple modelling

- Layers made exclusively of sensitive silicon



```
FOV: 60
FPS:   303.0
BO_RENDERING
```

```
Current Mesh: GeoVis_EventSpacePoints_20250313_123454
Vertices: 3    (291)
Faces: 0    (144)
Selection: v: 0 f: 0
FC
```

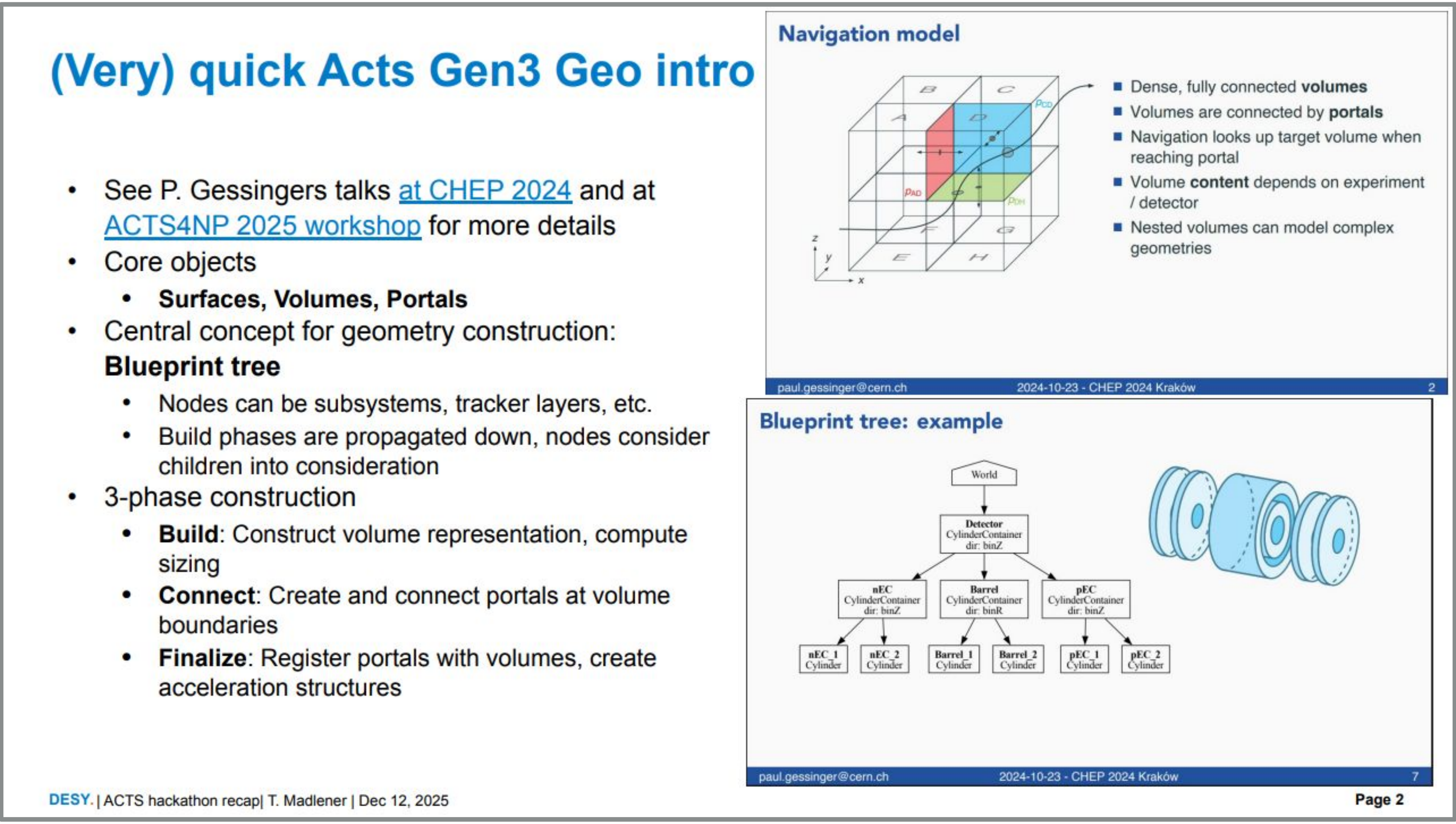
LUXE Tracking with ACTs v40 | Yufeng Wang

2*9*4=72 DD4hep sensitive volumes/ACTS surfaces

Page 3

Render from Y. Wang

Making the tracker geo usable in Acts 1/3



(Very) quick Acts Gen3 Geo intro

- See P. Gessingers talks at CHEP 2024 and at ACTS4NP 2025 workshop for more details
- Core objects
  - **Surfaces, Volumes, Portals**
- Central concept for geometry construction: **Blueprint tree**
  - Nodes can be subsystems, tracker layers, etc.
  - Build phases are propagated down, nodes consider children into consideration
- 3-phase construction
  - **Build**: Construct volume representation, compute sizing
  - **Connect**: Create and connect portals at volume boundaries
  - **Finalize**: Register portals with volumes, create acceleration structures

Navigation model

- Dense, fully connected **volumes**
- Volumes are connected by **portals**
- Navigation looks up target volume when reaching portal
- Volume **content** depends on experiment / detector
- Nested volumes can model complex geometries

paul.gessinger@cern.ch 2024-10-23 - CHEP 2024 Kraków 2

Blueprint tree: example

World

Detector
CylinderContainer
dir: binZ

nEC
CylinderContainer
dir: binZ

Barrel
CylinderContainer
dir: binR

pEC
CylinderContainer
dir: binZ

nEC_1
Cylinder

nEC_2
Cylinder

Barrel_1
Cylinder

Barrel_2
Cylinder

pEC_1
Cylinder

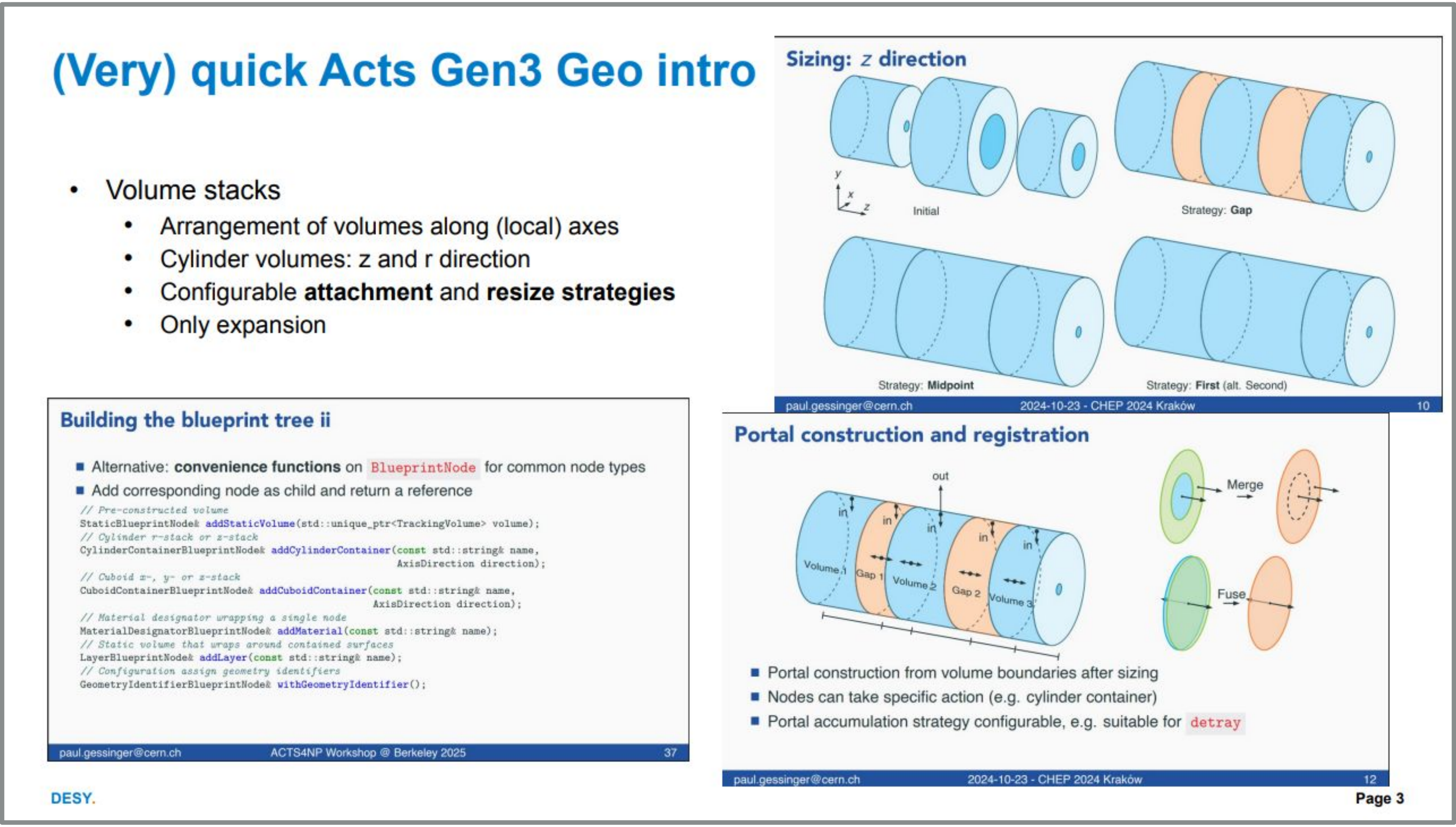
pEC_2
Cylinder

paul.gessinger@cern.ch 2024-10-23 - CHEP 2024 Kraków 7

DESY. | ACTS hackathon recap| T. Madlener | Dec 12, 2025

Page 2

Making the tracker geo usable in Acts 2/3



(Very) quick Acts Gen3 Geo intro

- Volume stacks
  - Arrangement of volumes along (local) axes
  - Cylinder volumes: z and r direction
  - Configurable **attachment** and **resize strategies**
  - Only expansion

Sizing: z direction

paul.gessinger@cern.ch 2024-10-23 - CHEP 2024 Kraków 10

Building the blueprint tree ii

- Alternative: **convenience functions** on `BlueprintNode` for common node types
- Add corresponding node as child and return a reference

```
// Pre-constructed volume
StaticBlueprintNode& addStaticVolume(std::unique_ptr<TrackingVolume> volume);
// Cylinder r-stack or z-stack
CylinderContainerBlueprintNode& addCylinderContainer(const std::string& name,
                                                     AxisDirection direction);
// Cuboid x-, y- or z-stack
CuboidContainerBlueprintNode& addCuboidContainer(const std::string& name,
                                                 AxisDirection direction);
// Material designator wrapping a single node
MaterialDesignatorBlueprintNode& addMaterial(const std::string& name);
// Static volume that wraps around contained surfaces
LayerBlueprintNode& addLayer(const std::string& name);
// Configuration assign geometry identifiers
GeometryIdentifierBlueprintNode& withGeometryIdentifier();
```

paul.gessinger@cern.ch ACTS4NP Workshop @ Berkeley 2025 37

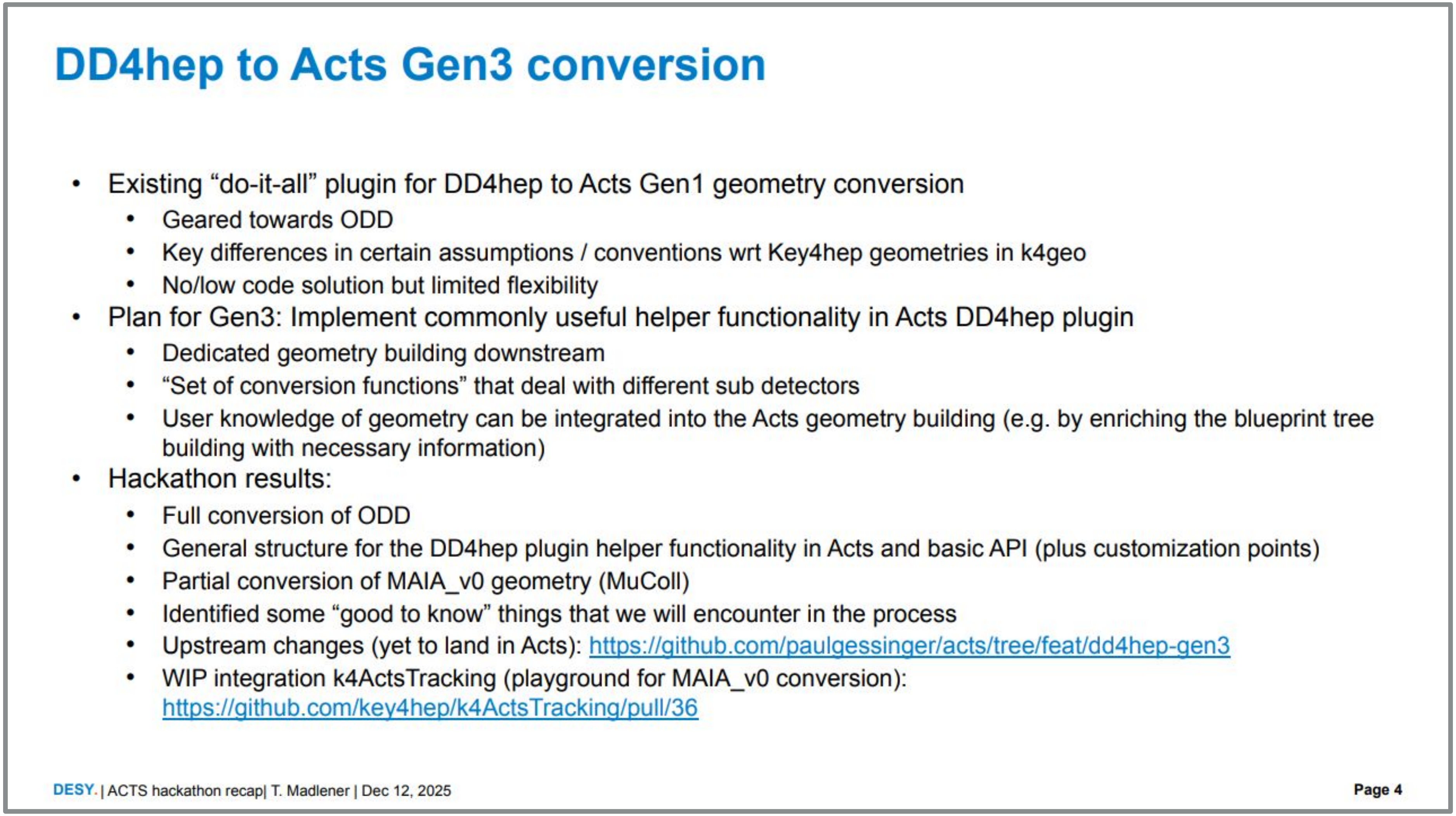
Portal construction and registration

- Portal construction from volume boundaries after sizing
- Nodes can take specific action (e.g. cylinder container)
- Portal accumulation strategy configurable, e.g. suitable for `detray`

paul.gessinger@cern.ch 2024-10-23 - CHEP 2024 Kraków 12

DESY. Page 3

Making the tracker geo usable in Acts 3/3



DD4hep to Acts Gen3 conversion

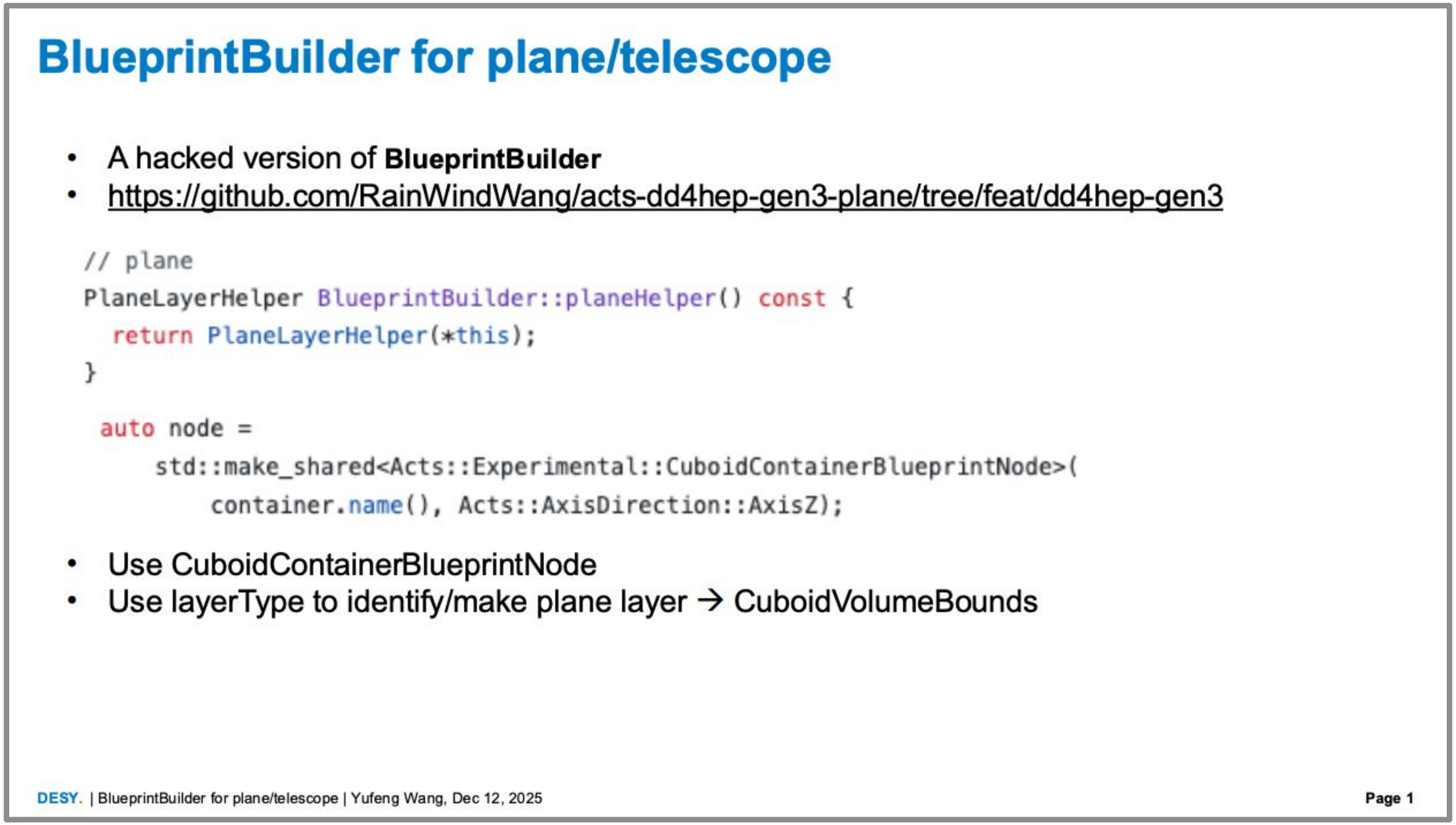
- Existing “do-it-all” plugin for DD4hep to Acts Gen1 geometry conversion
  - Geared towards ODD
  - Key differences in certain assumptions / conventions wrt Key4hep geometries in k4geo
  - No/low code solution but limited flexibility
- Plan for Gen3: Implement commonly useful helper functionality in Acts DD4hep plugin
  - Dedicated geometry building downstream
  - “Set of conversion functions” that deal with different sub detectors
  - User knowledge of geometry can be integrated into the Acts geometry building (e.g. by enriching the blueprint tree building with necessary information)
- Hackathon results:
  - Full conversion of ODD
  - General structure for the DD4hep plugin helper functionality in Acts and basic API (plus customization points)
  - Partial conversion of MAIA_v0 geometry (MuColl)
  - Identified some “good to know” things that we will encounter in the process
  - Upstream changes (yet to land in Acts): https://github.com/paulgessinger/acts/tree/feat/dd4hep-gen3
  - WIP integration k4ActsTracking (playground for MAIA_v0 conversion): https://github.com/key4hep/k4ActsTracking/pull/36

DESY. | ACTS hackathon recap| T. Madlener | Dec 12, 2025 Page 4

Marlin-to-gaudi tracking with Acts

Targeting ACTS > v44.1.0 and gaudi

- Existing implementation in MarlinACTSTracking and ACTS v13
- Focus on building geometry with ACTS Gen3 geo, rest of algorithms almost directly portable



BlueprintBuilder for plane/telescope

- A hacked version of **BlueprintBuilder**
- https://github.com/RainWindWang/acts-dd4hep-gen3-plane/tree/feat/dd4hep-gen3

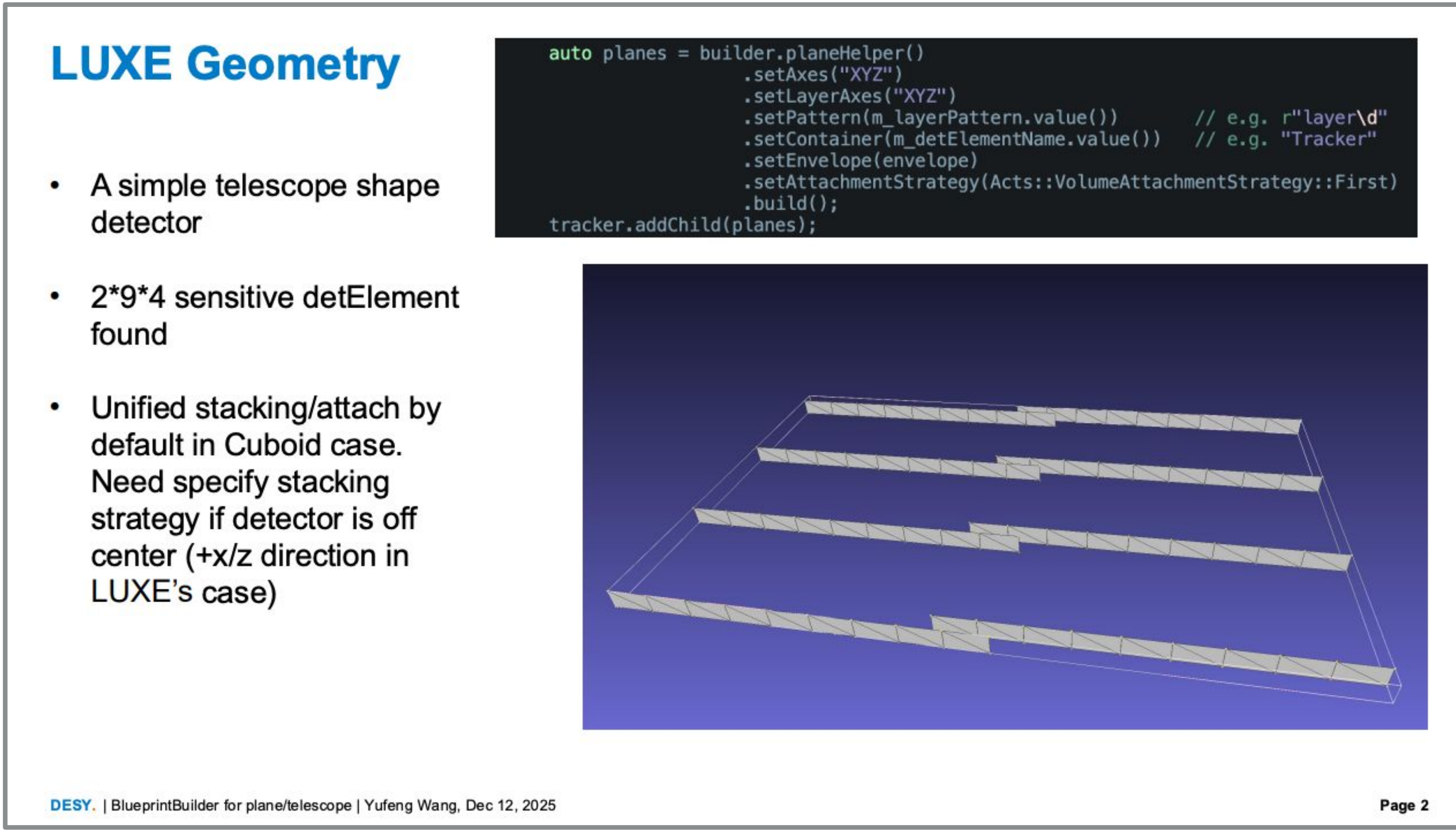
```
// plane
PlaneLayerHelper BlueprintBuilder::planeHelper() const {
  return PlaneLayerHelper(*this);
}

 auto node =
     std::make_shared<Acts::Experimental::CuboidContainerBlueprintNode>(
         container.name(), Acts::AxisDirection::AxisZ);
```

- Use CuboidContainerBlueprintNode
- Use layerType to identify/make plane layer → CuboidVolumeBounds

DESY. | BlueprintBuilder for plane/telescope | Yufeng Wang, Dec 12, 2025 Page 1

LUXE in ACTS Gen3 geometry



LUXE Geometry

- A simple telescope shape detector
- 2*9*4 sensitive detElement found
- Unified stacking/attach by default in Cuboid case. Need specify stacking strategy if detector is off center (+x/z direction in LUXE's case)

```
auto planes = builder.planeHelper()
                  .setAxes("XYZ")
                  .setLayerAxes("XYZ")
                  .setPattern(m_layerPattern.value())       // e.g. r"layer\d"
                  .setContainer(m_detElementName.value())   // e.g. "Tracker"
                  .setEnvelope(envelope)
                  .setAttachmentStrategy(Acts::VolumeAttachmentStrategy::First)
                  .build();
tracker.addChild(planes);
```

DESY. | BlueprintBuilder for plane/telescope | Yufeng Wang, Dec 12, 2025

Page 2

Goals for collaboration meeting

- Place detector models in their respective positions
- Add dipole field after IP
- Simulate particle gun events, validate propagation through detector volume, look at basic simhit-related distributions

Bonus (in no particular order) and next steps

- Run tracker digitisation and reconstruct tracks with ACTS
- Run Pandora PFO reconstruction to combine tracker and calorimeter
- Port PTARMIGAN hdf5-to-LCIO conversion scripts into conversion to edm4hep and run simulation on actual particles
- Validate simulated detector hits against lxelsim

Thank you!

How do I use the software releases

On your laptop / standalone setup:

```
> docker pull ghcr.io/luxesoftware/luxe-sw:latest (you only need to do this once!)
> docker run -it -v /optional/local/folder:/target/in/image --rm
ghcr.io/luxesoftware/luxe-sw:latest
> setup_luxe
```

On a cvmfs-enabled machine:

```
> apptainer run /cvmfs/unpacked.cern.ch/ghcr.io/luxesoftware/luxe-sw:latest
> setup_luxe
```

The last command will give you access to the whole suite of packages, you can then just proceed with your work

Problems? Email me - I’ll help you getting this started