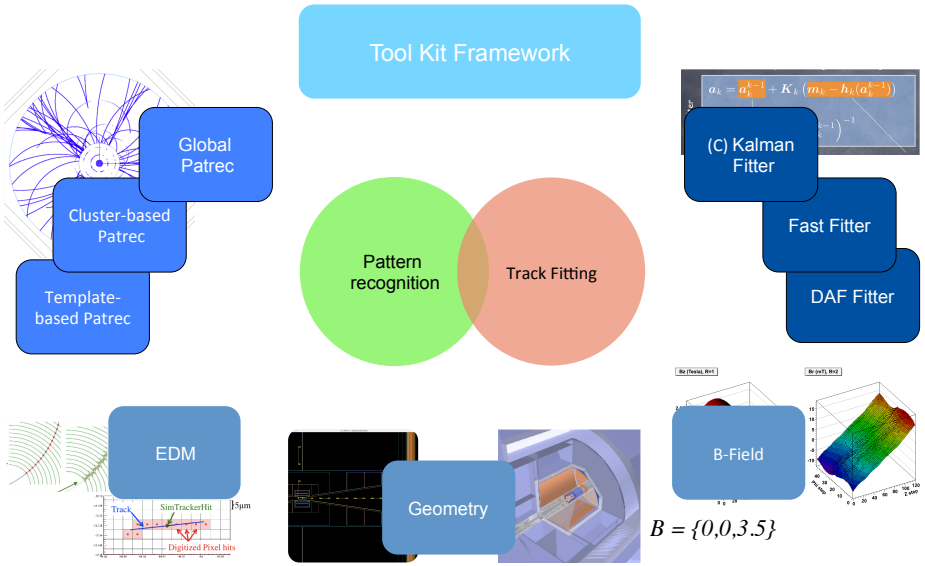
ILD Tracking
Status and Plans



Ch. Rosemann

DESY

Towards the current status

Development in the past two to three years driven by DBD requirements

- Realistic and accurate detector models are a must
- Inclusion of beam-induced background

Status quo pre-DBD

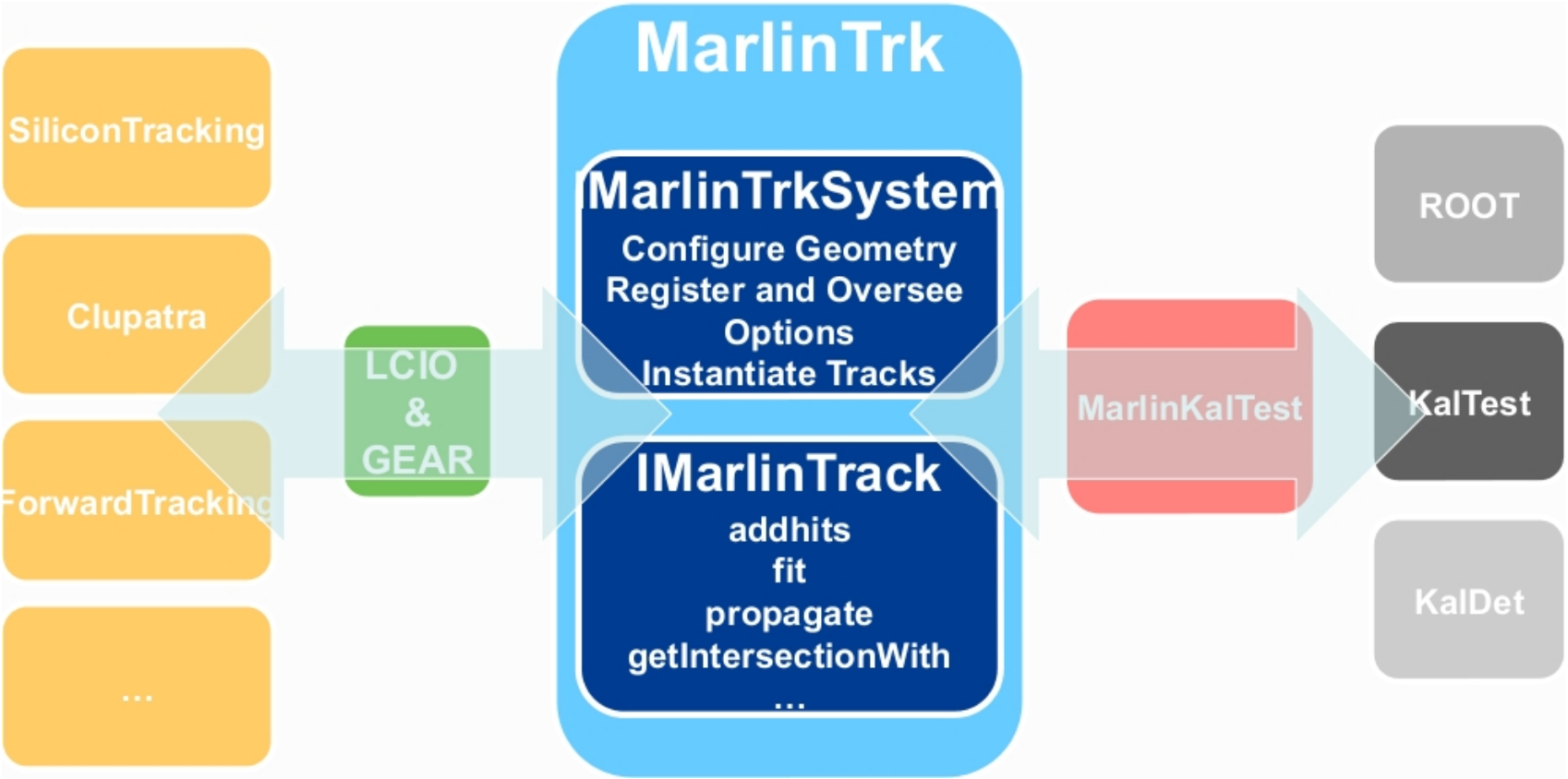
- Model for simulation & reconstruction:
  mixture of simplified and realistic descriptions
- For track reconstruction a patchwork of code was used
- Lots of F77, lots of common blocks

Objective

- Replace F77 tracking code
- Make software maintainable (and extensible)
- Optimize Performance

At the core: EDM prototype MarlinTrk

- IMarlinTrk: Interface class to track fitting/parameter manipulation
- IMarlinTrkSystem: Infrastructure/interface to geometry & persistency



The new three working horses

New C++ Kalman Filter

KalTest/KalDet by K. Fujii et al.

- Common usage with hardware collaboration
- Implemented in MarlinTrk as MarlinKalTest (S. Aplin)

New C++ pattern recognition

Clupatra by F. Gaede

- Now also in use for test beam data (O. Volynets)
- Uses MarlinTrk interface in implementation

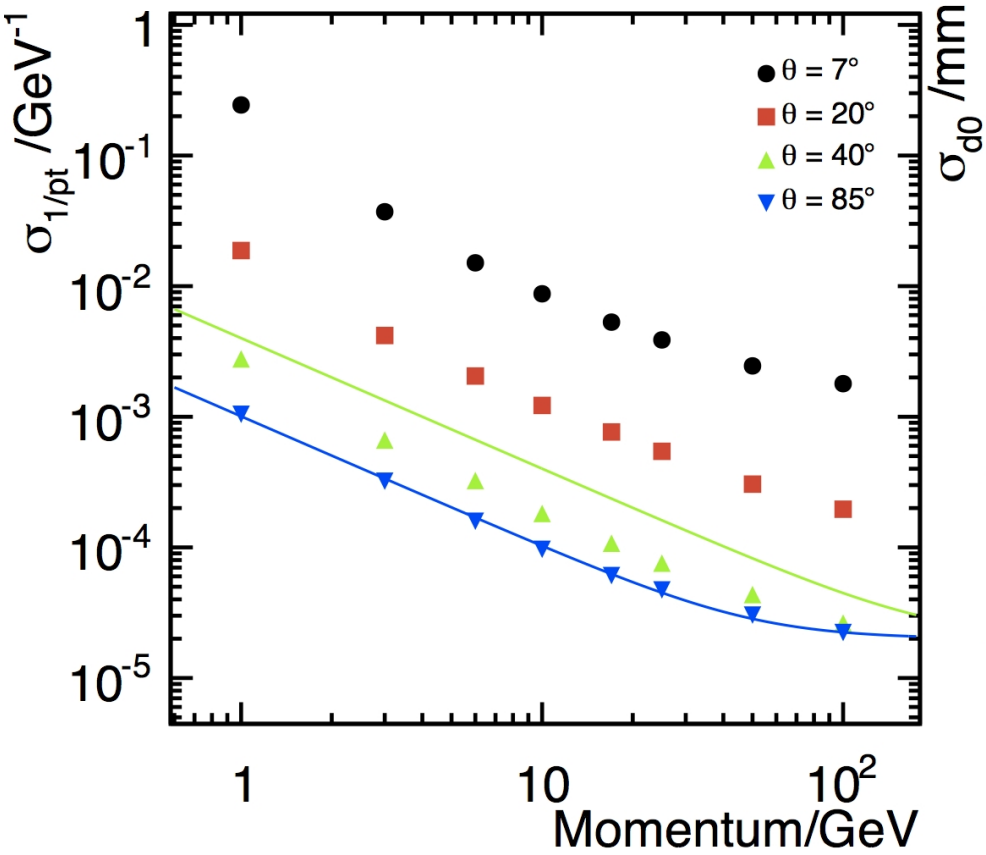
New C++ Silicon Tracking

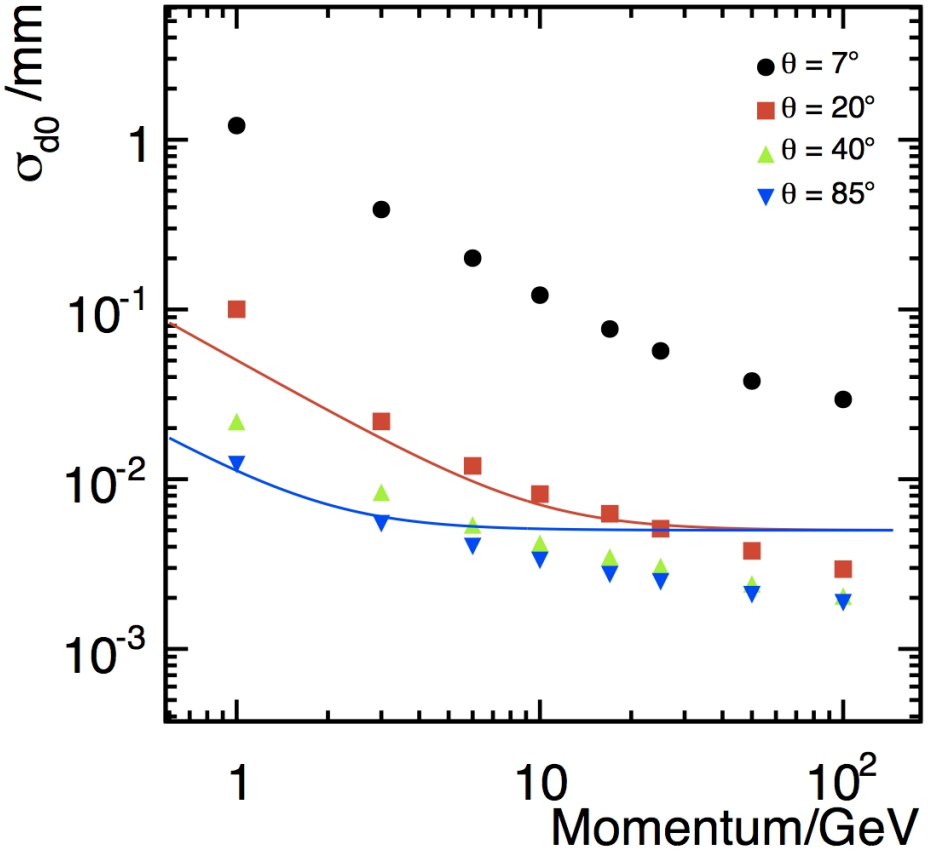
- Forward tracking by R. Glattauer
  Interfaced to MarlinTrk in FwdTracking
- SiliconTracking_MarlinTrk, see dedicated talk by Yorgos Voutsinas

Performance

Huge validation

DBD production: 10^7 events reconstructed





Where to go from here?

(Very) Limited man power available

- Move towards common software tools
- Obvious benefits: avoid duplication of work, sharing of workload
- DBD left us with a lot to build on

Window of opportunity

- Move into direction of common and generic tracking package
- Specific application (+testbed, +example, etc.): ILD tracking

Funding and collaboration

EU Framework Programme: AIDA – WP2: common software tools

Current development objectives

Generic tracking package

- MarlinTrk is the first proof-of-concept prototype
- Interfaced to GEAR and KalTest

Geometry interfacing

- GEAR doesn't fulfill requirements
- dd4hep is successor

Tracking in a nutshell

1. Algorithm and its steering
2. Geometry relations: track parameter propagation and material effects

We have (1) but (2) is neither complete nor compatible

Geometry related needs in a nutshell

To perform a track fit on a set of hits one needs the

1. amount of passive material (e.g. in X0) passed by a trajectory between two hits
2. exact point which is going to be hit on an active readout surface while following a trajectory
3. normal vector(s) of the measurement surface at the crossing point, a.k.a the local measurement directions

- Not everything can be done by geometry package
- But very tight connection to geometry
- Need good and well-defined interface!
- Two challenges:
  1. Solve the problem(s): design interface and algorithms
  2. Efficiency: accuracy vs. time consumption

Three parts toward a generic tracking package

Major part: generic geometry toolkit

- DD4Hep: one single and unique description for simulation and reconstruction geometry
- Interface to GEANT exists
- Implementation in TGeo exists
- *Tracking geometry* not yet defined

Extend event/tracking data model

- Prototype exists
- Look at existing tool kits and extend if needed

Connection of tracking algorithms

- Look for larger overlap with geometry toolkit
- Chosen project already exists: GENFIT

Currently in progress

Exchange the algorithmic side for easier connection with geometry

- Started working with GENFIT: generic tracking tool, used for Belle-2
- Very large overlap with goals of tracking WP
- Geometry is based on/built with TGeo
- Clear connection to dd4hep

GENFIT in 5 s

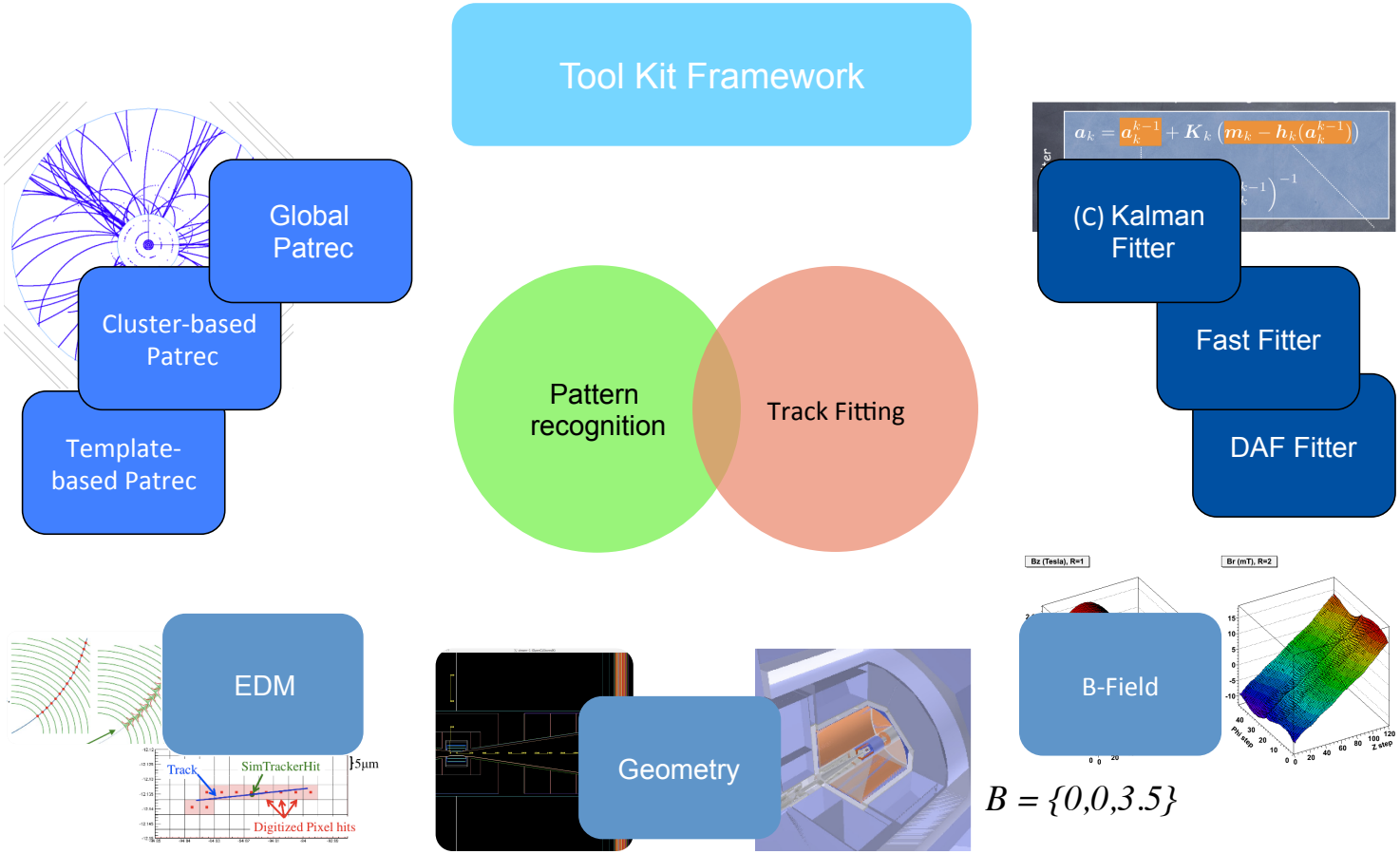
- Algorithmic content: validated Kalman, DAF, General Broken Lines
- Track propagation with different tools, including GEANE
- Runs with full simulation geometry in micro stepping mode
- Major revision (GENFIT2) in progress
- First working track fit in GENFIT2 since last week

The goal: a common tracking toolkit

The plans in detail

- Take ILD tracking to next level:
  easier maintenance, easier inclusion of new features
- Arrive at a working tracking toolkit for all LC studies
- AIDA WP2 deliverable:
  Detector independent tracking toolkit
- Fundamental design choice:
  Create independent packages

The goal: a common tracking toolkit



(pic by Steve Aplin)