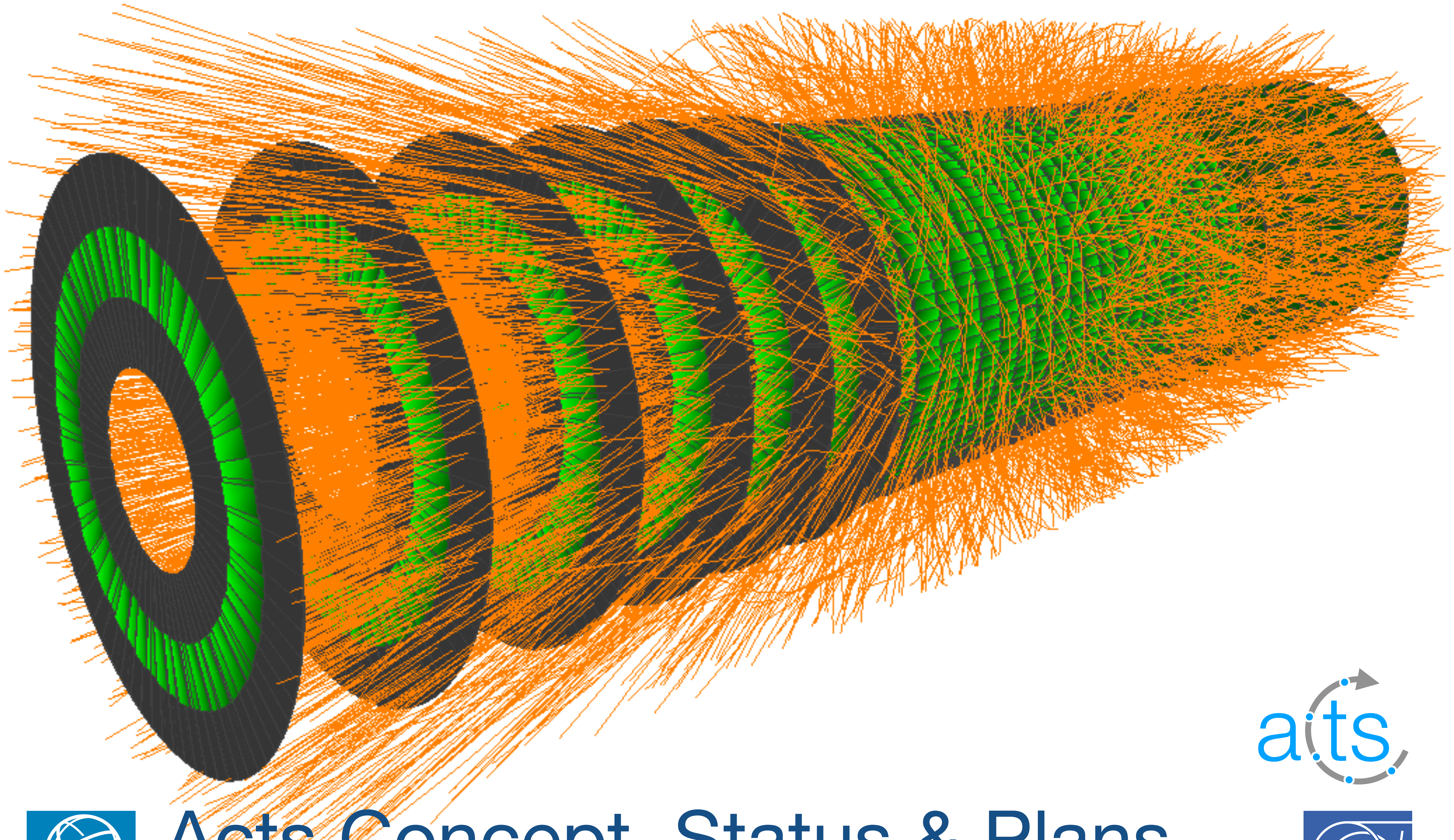
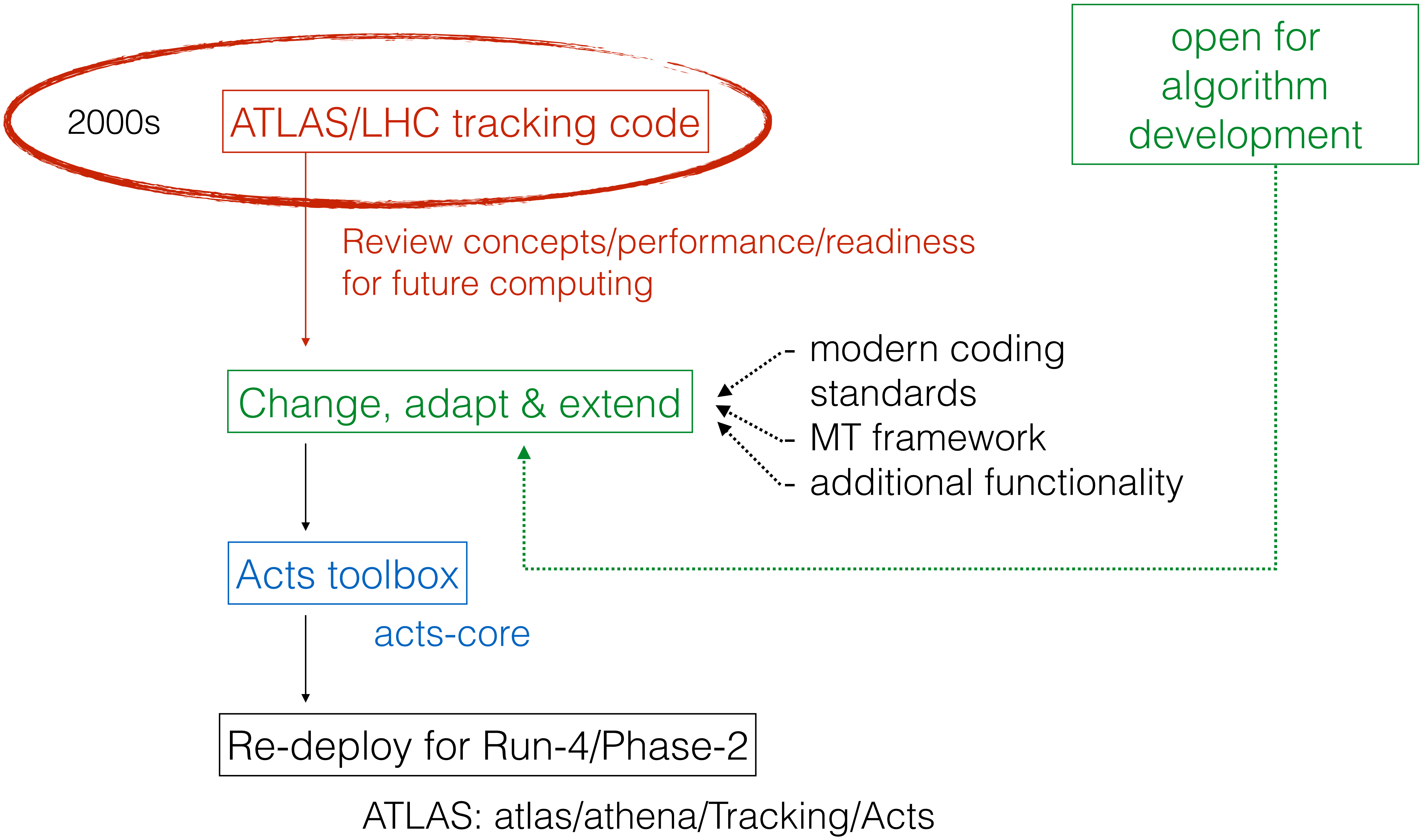


Acts Concept, Status & Plans

https://cern.ch/acts

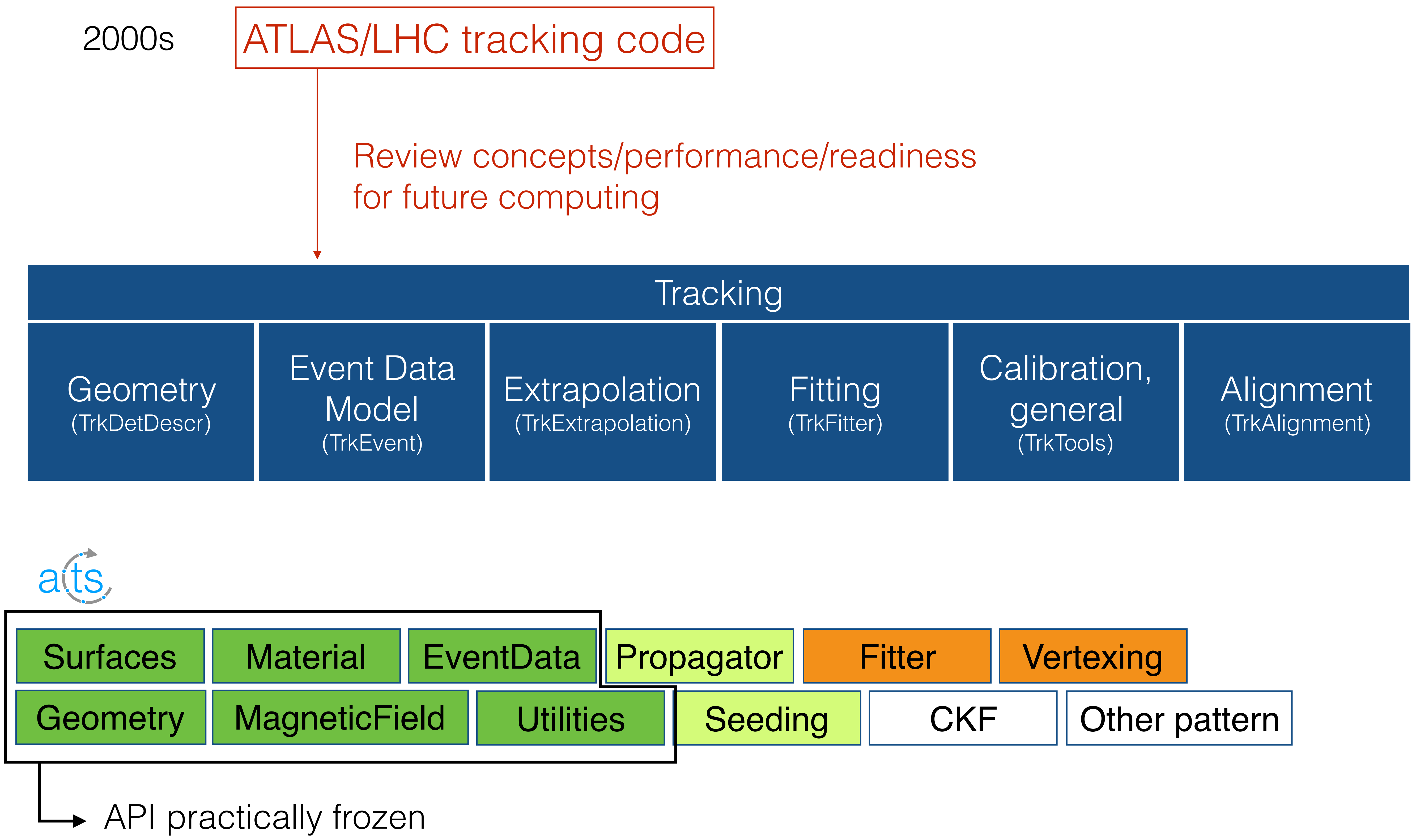
A. Salzburger (CERN)

Mission statement

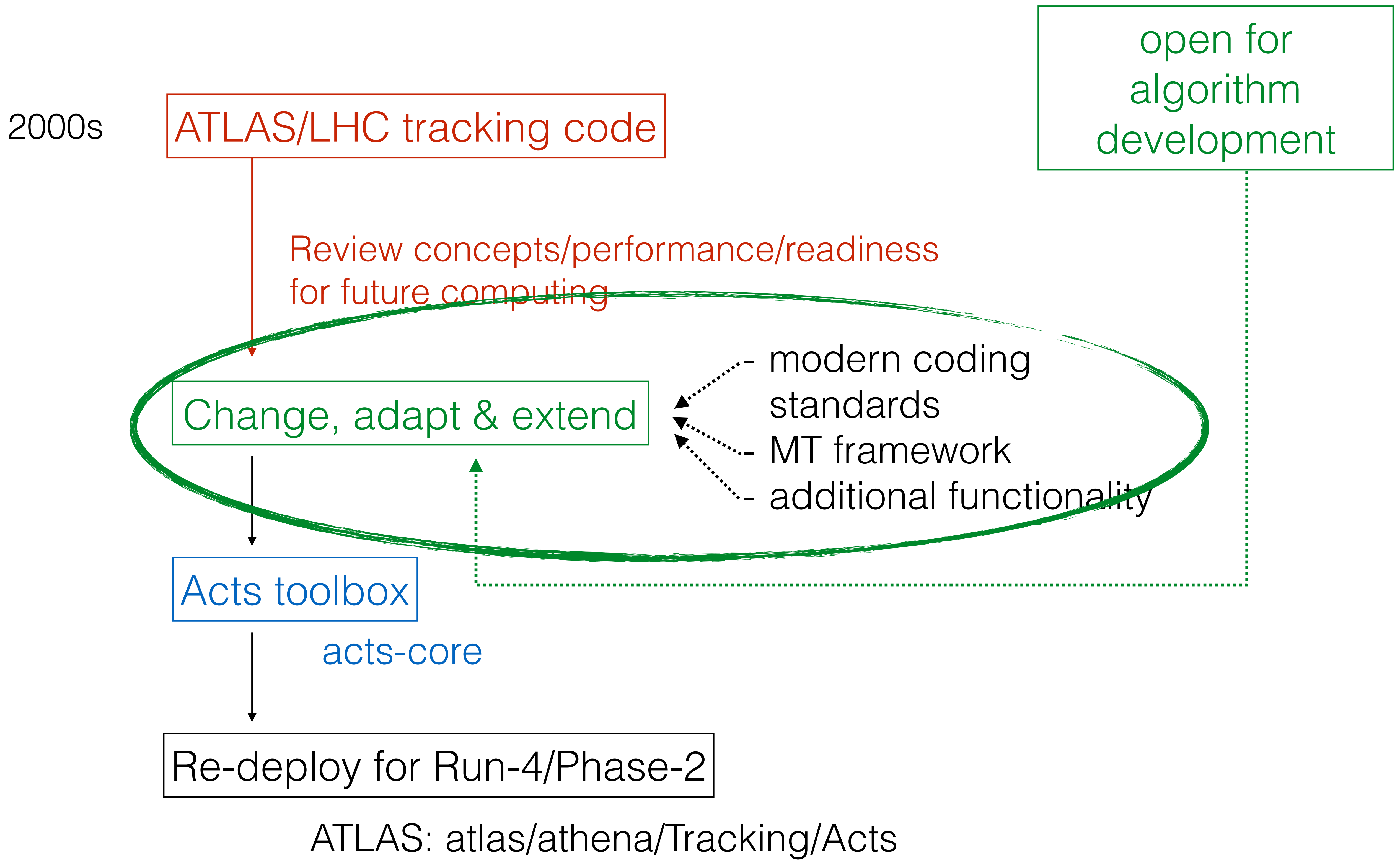


2000s
ATLAS/LHC tracking code
Review concepts/performance/readiness for future computing
Change, adapt & extend
- modern coding standards
- MT framework
- additional functionality
open for algorithm development
Acts toolbox
acts-core
Re-deploy for Run-4/Phase-2
ATLAS: atlas/athena/Tracking/Acts

Status Tracking code porting



Mission statement


2000s
ATLAS/LHC tracking code
Review concepts/performance/readiness for future computing
open for algorithm development
Change, adapt & extend
- modern coding standards
- MT framework
- additional functionality
Acts toolbox
acts-core
Re-deploy for Run-4/Phase-2
ATLAS: atlas/athena/Tracking/Acts

Feature completing

E.g., included time-component into full tracking stack

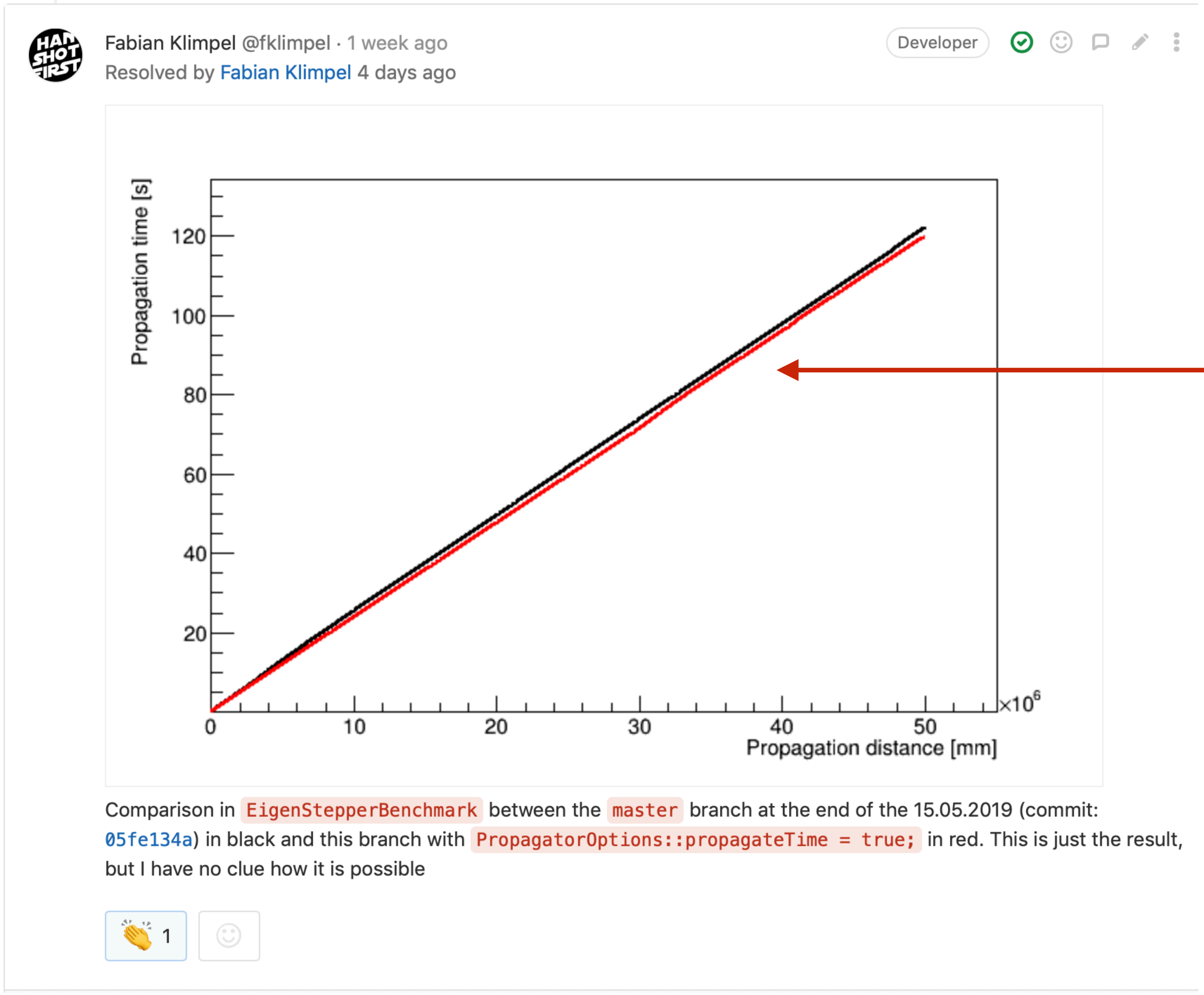
- Most of LHC code it is (whenever needed) post-fitted

Internal representation expanded from **7x7** description to **8x8**

- full time covariance transport developed (and numerically tested)
- positive impact on execution speed

$$\mathbf{q} = \boxed{(l_1, l_2,}\boxed{\phi, \theta, q/p}\,, t)$$

Tracking
Acts



Fabian Klimpel @fklimpel · 1 week ago
Resolved by Fabian Klimpel 4 days ago

Comparison in `EigenStepperBenchmark` between the `master` branch at the end of the 15.05.2019 (commit: `05fe134a`) in black and this branch with `PropagatorOptions::propagateTime = true;` in red. This is just the result, but I have no clue how it is possible

👏 1

- could be better vectorisation, not confirmed yet

Getting fully MT ready *the extra mile*

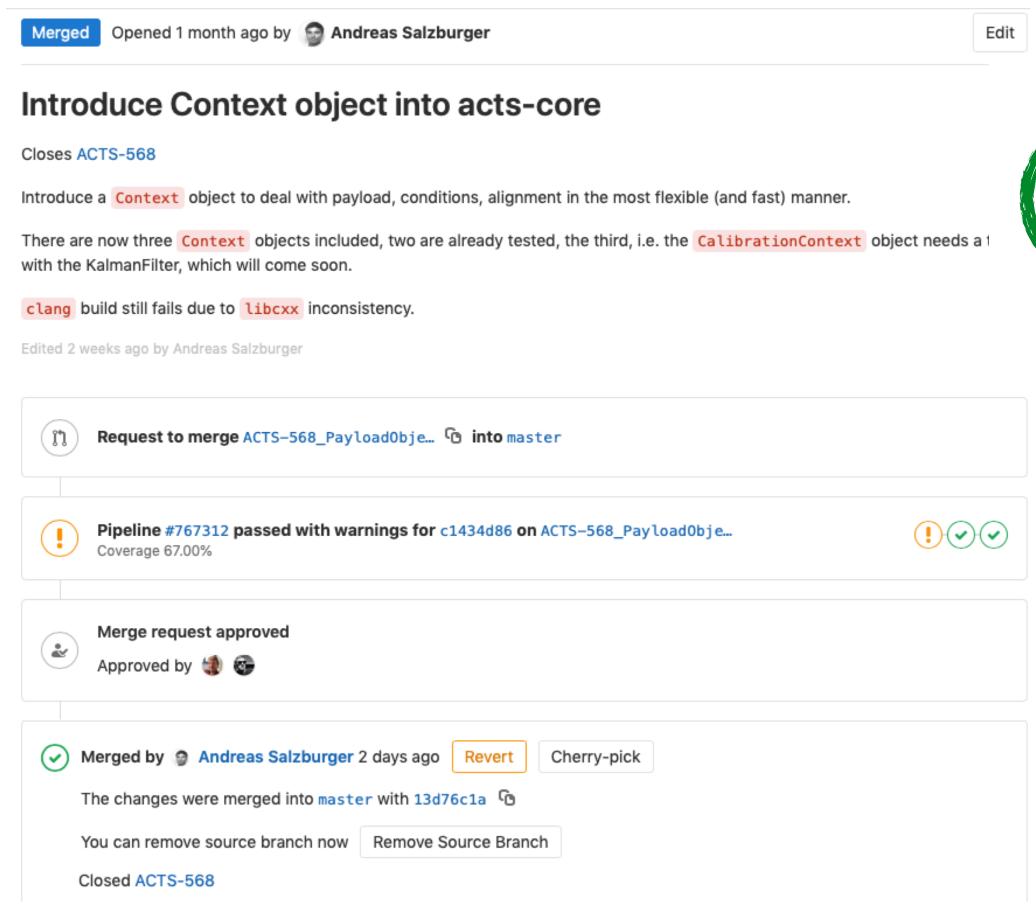
Based on ATLAS needs developed a full contextual tracking setup

- Lumiblock based alignment for ATLAS due to IBL bowing

  Not the most standard workflow, however, needs a clean solution

  *In a real **MULTI**-threaded environment, several alignment or conditions might have to be in memory*

  (e.g. high selective trigger stream)



Merged Opened 1 month ago by Andreas Salzburger Edit

Introduce Context object into acts-core

Closes ACTS-568

Introduce a `Context` object to deal with payload, conditions, alignment in the most flexible (and fast) manner.

There are now three `Context` objects included, two are already tested, the third, i.e. the `CalibrationContext` object needs a t
with the KalmanFilter, which will come soon.

`clang` build still fails due to `libcxx` inconsistency.

Edited 2 weeks ago by Andreas Salzburger

Request to merge `ACTS-568_PayloadObje…` **into** `master`

Pipeline `#767312` **passed with warnings for** `c1434d86` **on** `ACTS-568_PayloadObje…`
Coverage 67.00%

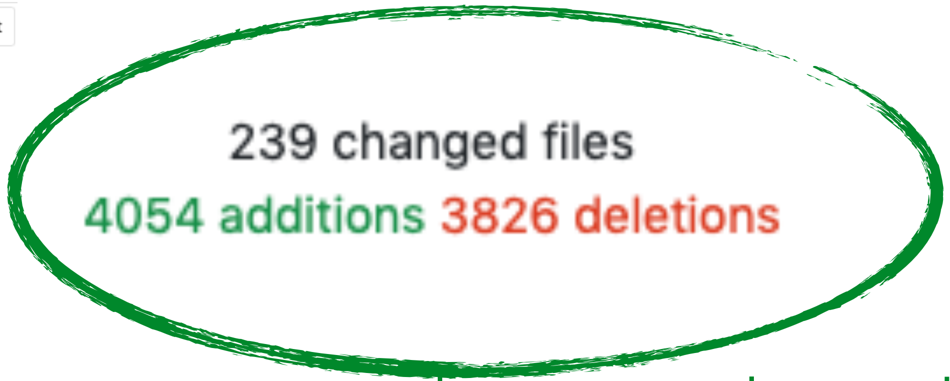
Merge request approved
Approved by

Merged by Andreas Salzburger 2 days ago Revert Cherry-pick

The changes were merged into `master` with `13d76c1a`

You can remove source branch now Remove Source Branch

Closed ACTS-568

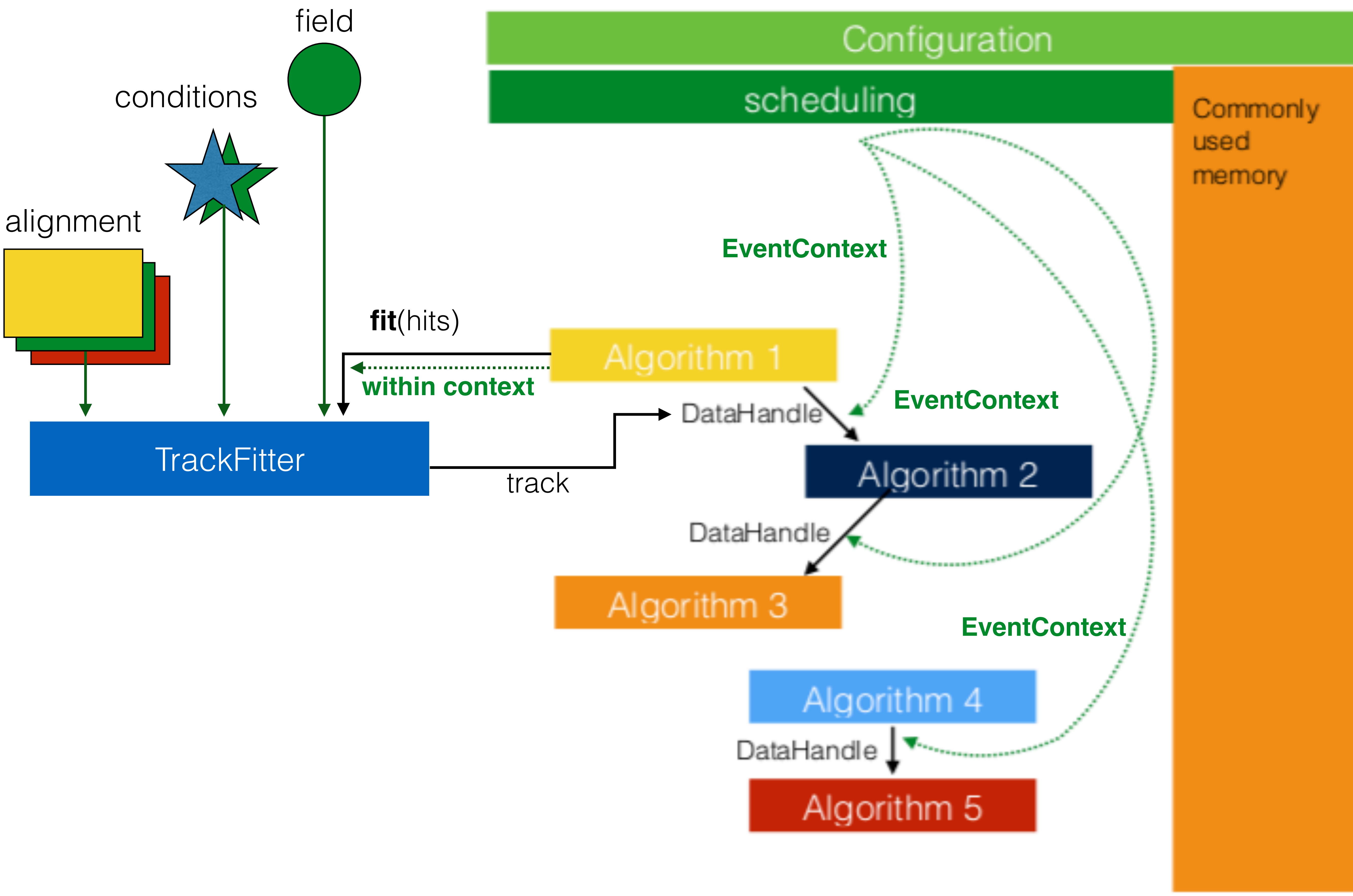


239 changed files

4054 additions 3826 deletions

large scale update
for MT capability

Contextual Tracking The "clean" solution



ACTS with Context

Introduced context objects in `acts-core` & testes in `acts-framework`

- nomen est omen

```
/// Aggregated information to run one algorithm over one event.
struct AlgorithmContext
{
  size_t                      algorithmNumber;   ///< Unique algorithm identifier
  size_t                      eventNumber;      ///< Unique event identifier
  WhiteBoard&                 eventStore;       ///< Per-event data store
  Acts::GeometryContext       geoContext;       ///< Per-event geometry context
  Acts::MagneticFieldContext  magFieldContext;  ///< Per-event magnetic Field context
  Acts::CalibrationContext    calibContext;     ///< Per-event calibration context
};
```

While they are untouched in `acts-core` and simply defined as

```
#pragma once

/// Set the identifier PLUGIN
#ifdef ACTS_CORE_GEOMETRYCONTEXT_PLUGIN
#include ACTS_CORE_GEOMETRYCONTEXT_PLUGIN
#else
#include <any>

namespace Acts {

using GeometryContext        = std::any;
using DefaultGeometryContext = GeometryContext;

}  // namespace Acts

#endif
```

$\longleftarrow$ can even be overloaded

Parallelism testbed

Test with different alignment every single event

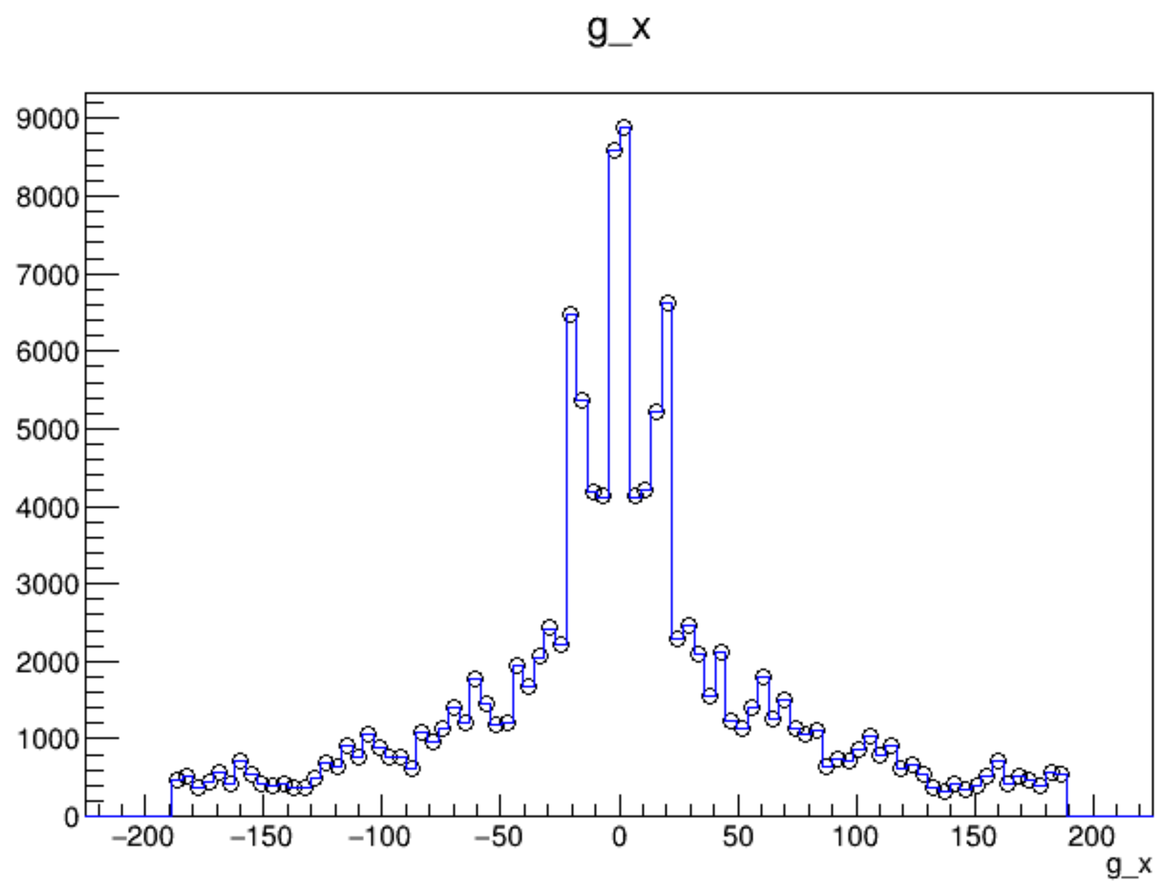
```
salzburg$ export ACTSFW_NUM_THREADS=1
salzburg$ ./ACTFWAlignablePropagationExample -n10 --prop-ntests 1000 --bf-values 0 0 2 --output-root 1
12:49:10    Sequencer      INFO      Added context decorator GeometryRotationDecorator
12:49:10    Sequencer      INFO      Added service RandomNumbersSvc
12:49:10    Sequencer      INFO      Appended algorithm PropagationAlgorithm
12:49:11    Sequencer      INFO      Added writer RootPropagationStepsWriter
12:49:11    Sequencer      INFO      Starting event loop for
12:49:11    Sequencer      INFO        1 services
12:49:11    Sequencer      INFO        0 readers
12:49:11    Sequencer      INFO        1 writers
12:49:11    Sequencer      INFO        1 algorithms
12:49:11    Sequencer      INFO      Run the event loop
12:49:11    Sequencer      INFO      start event 0
12:49:12    Sequencer      INFO      event 0 done
12:49:12    Sequencer      INFO      start event 1
12:49:13    Sequencer      INFO      event 1 done
12:49:13    Sequencer      INFO      start event 2
12:49:14    Sequencer      INFO      event 2 done
12:49:14    Sequencer      INFO      start event 3
12:49:15    Sequencer      INFO      event 3 done
12:49:15    Sequencer      INFO      start event 4
12:49:16    Sequencer      INFO      event 4 done
12:49:16    Sequencer      INFO      start event 5
12:49:17    Sequencer      INFO      event 5 done
12:49:17    Sequencer      INFO      start event 6
12:49:19    Sequencer      INFO      event 6 done
12:49:19    Sequencer      INFO      start event 7
12:49:19    Sequencer      INFO      event 7 done
12:49:19    Sequencer      INFO      start event 8
12:49:20    Sequencer      INFO      event 8 done
12:49:20    Sequencer      INFO      start event 9
12:49:22    Sequencer      INFO      event 9 done
12:49:22    Sequencer      INFO      Running end-of-run hooks of writers and services
```

```
salzburg$ export ACTSFW_NUM_THREADS=4
12:51:19    Sequencer      INFO      start event 0
12:51:19    Sequencer      INFO      start event 5
12:51:19    Sequencer      INFO      start event 8
12:51:19    Sequencer      INFO      start event 7
12:51:20    Sequencer      INFO      event 7 done
12:51:20    Sequencer      INFO      start event 2
12:51:21    Sequencer      INFO      event 8 done
12:51:21    Sequencer      INFO      start event 9
12:51:21    Sequencer      INFO      event 5 done
12:51:21    Sequencer      INFO      start event 6
12:51:21    Sequencer      INFO      event 0 done
12:51:21    Sequencer      INFO      start event 1
12:51:22    Sequencer      INFO      event 2 done
12:51:22    Sequencer      INFO      start event 3
12:51:23    Sequencer      INFO      event 9 done
12:51:23    Sequencer      INFO      start event 4
12:51:23    Sequencer      INFO      event 6 done
12:51:23    Sequencer      INFO      event 1 done
12:51:23    Sequencer      INFO      event 3 done
12:51:24    Sequencer      INFO      event 4 done
```

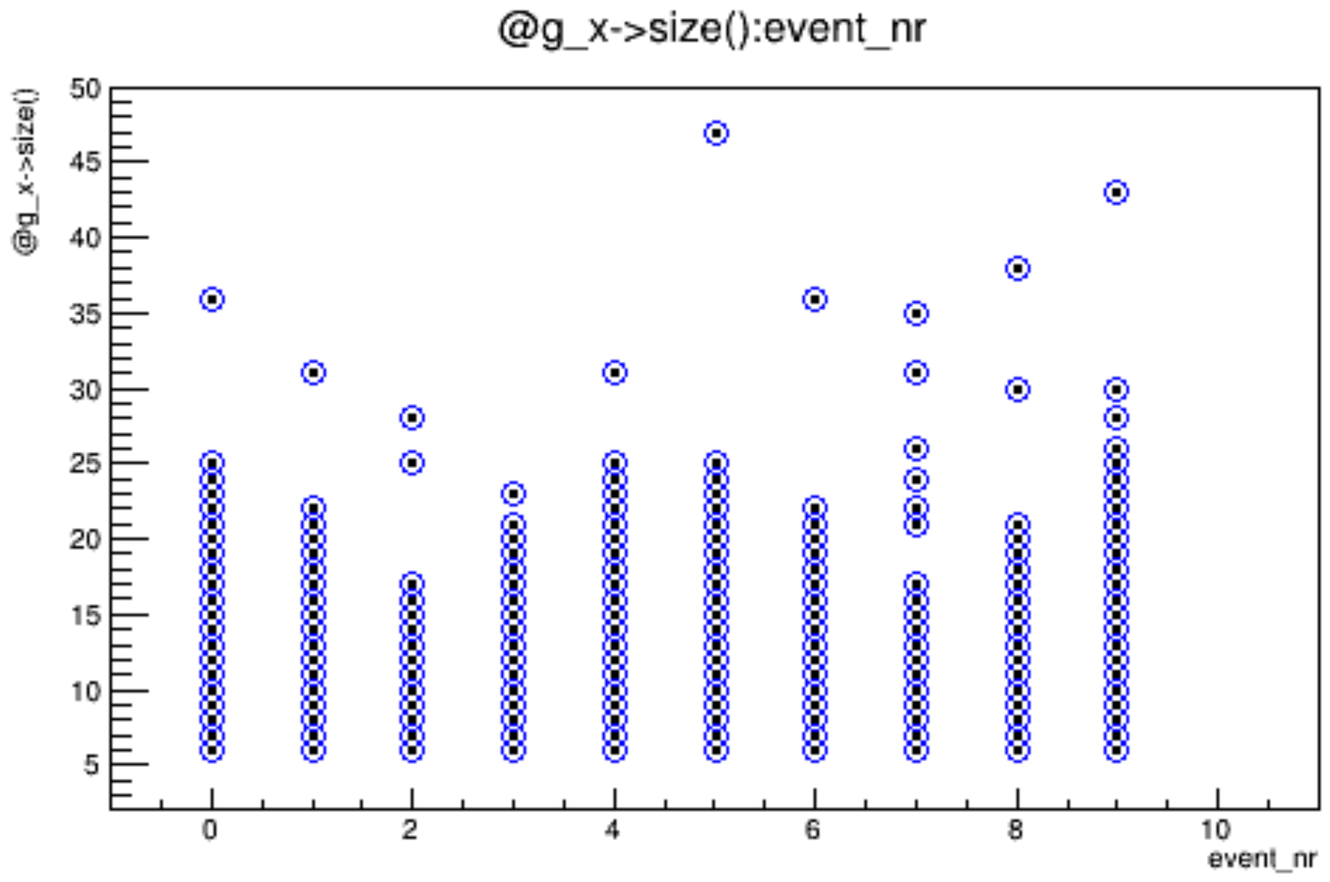
12 seconds $\longrightarrow$ 5 seconds

GeometryContext Comparing the two

Total comparison:

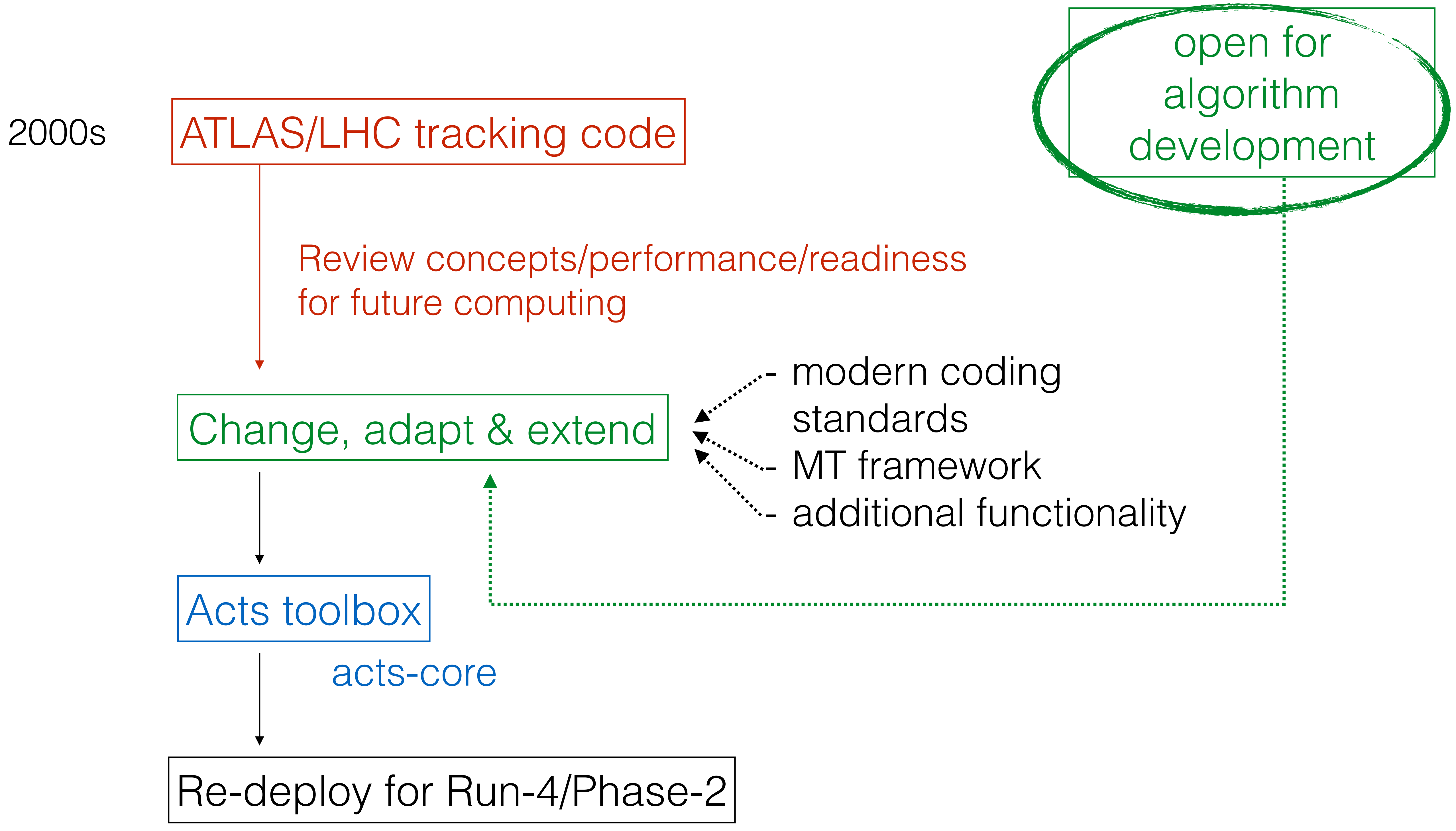


Per event comparison:



```
salzburg$ export ACTSFW_NUM_THREADS=1
salzburg$ export ACTSFW_NUM_THREADS=4
```

Mission statement



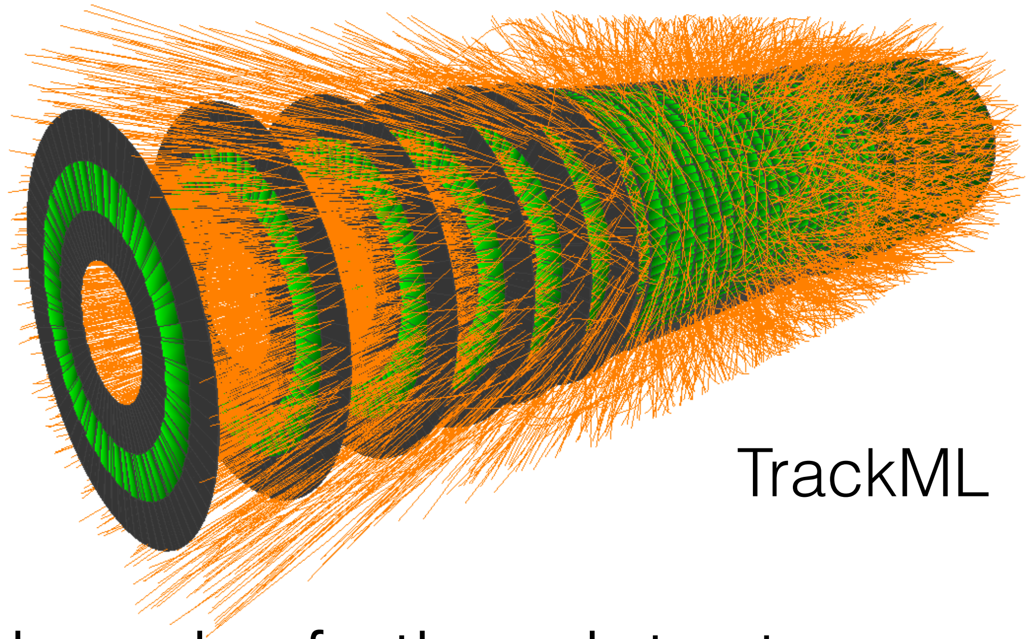
2000s
ATLAS/LHC tracking code
Review concepts/performance/readiness for future computing
Change, adapt & extend
- modern coding standards
- MT framework
- additional functionality
open for algorithm development
Acts toolbox
acts-core
Re-deploy for Run-4/Phase-2

ATLAS: atlas/athena/Tracking/Acts

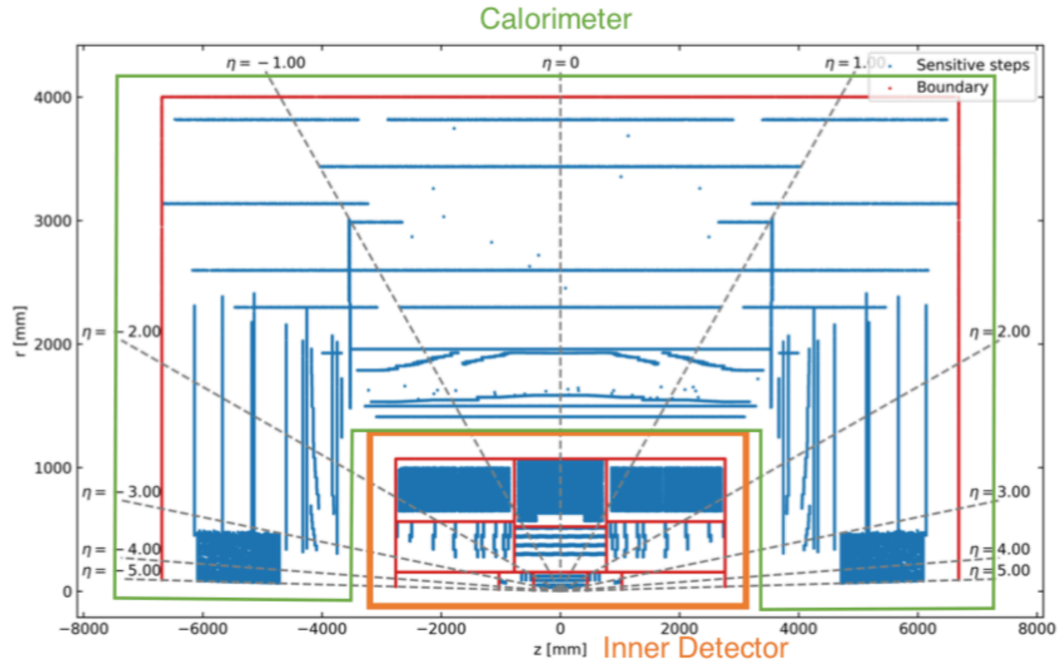
TrackML Aftermath

Started to port first TrackML algorithms into `acts-framework`

- Idea is to create a testbed for algorithm development and templating
- provide several detectors to test on

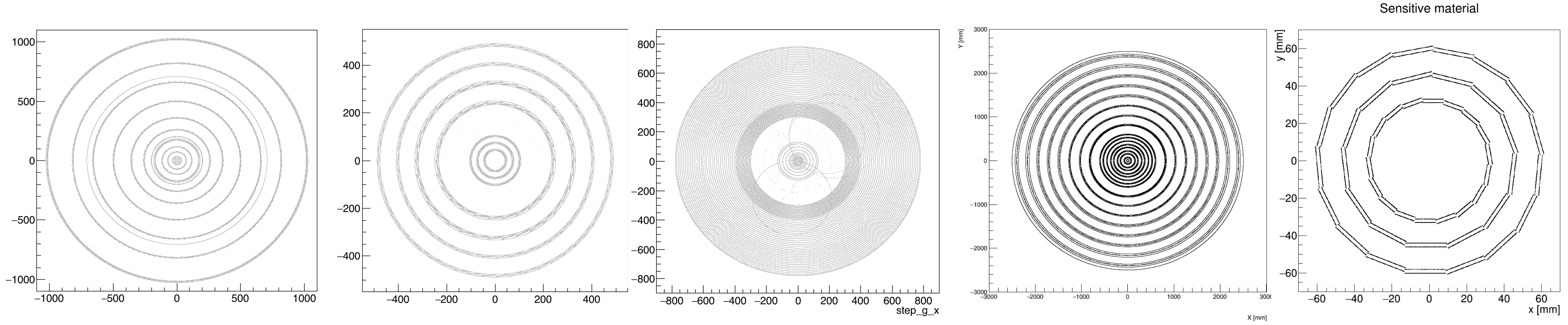


TrackML



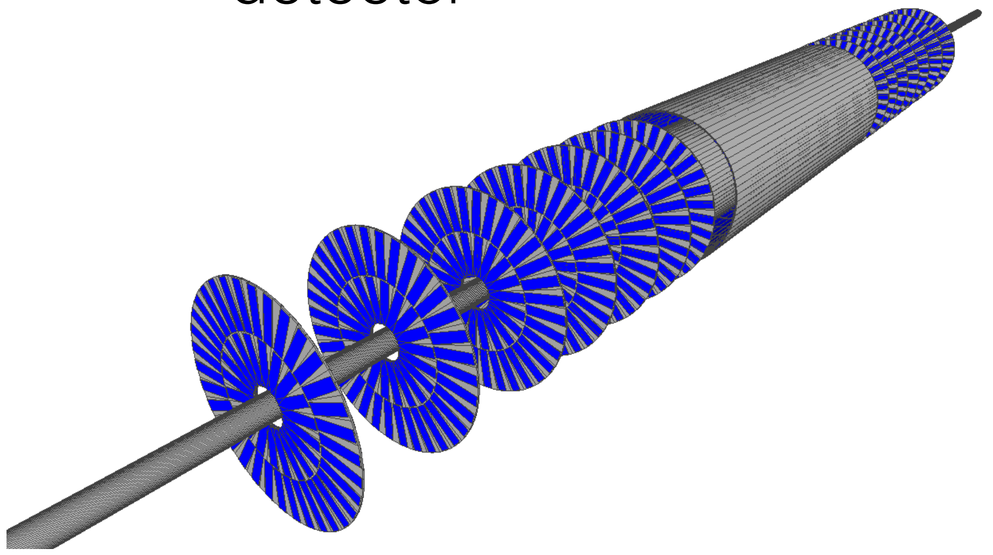
ATLAS

A bunch of other detectors:

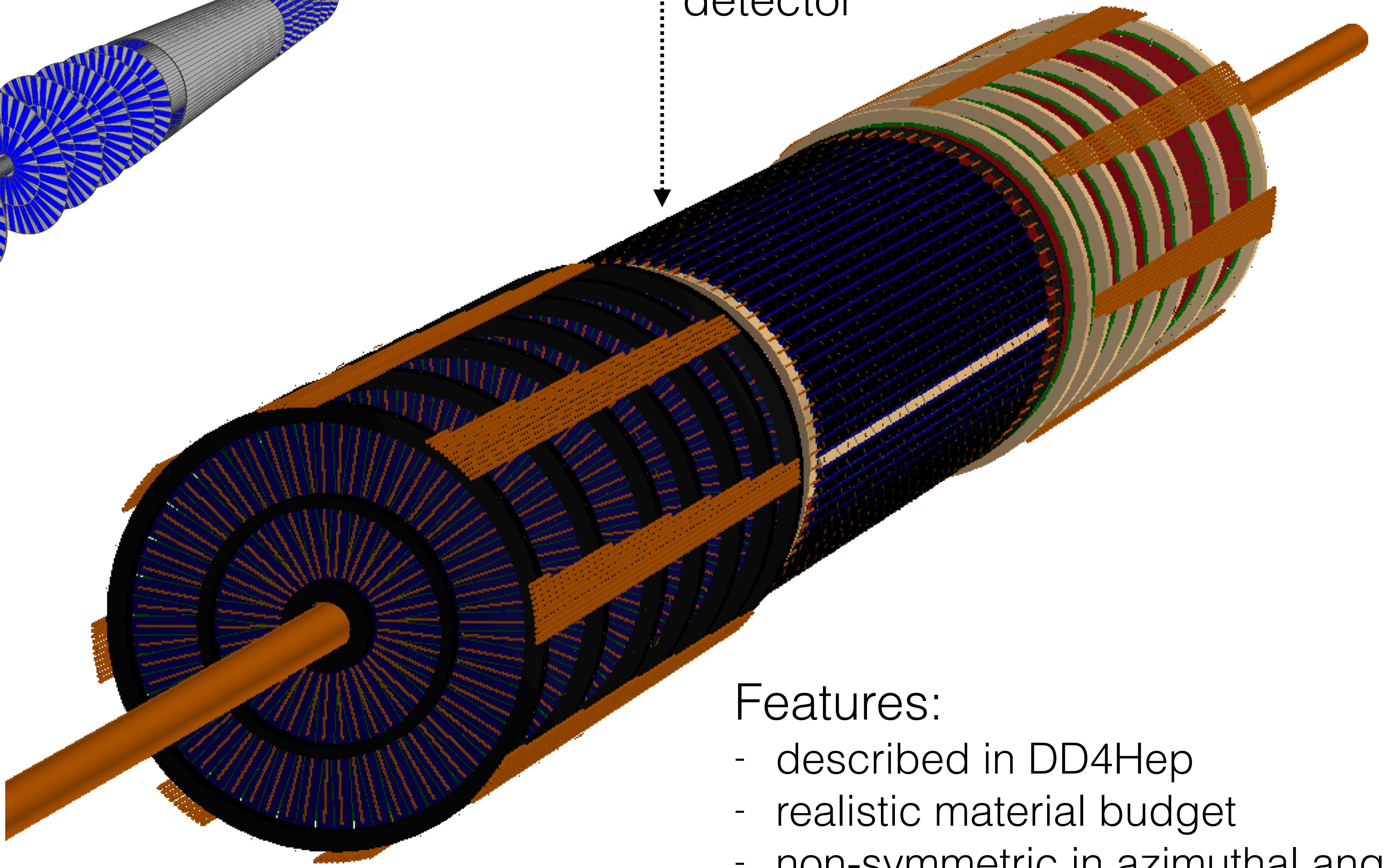


OpenData Detector

TrackML Pixel detector



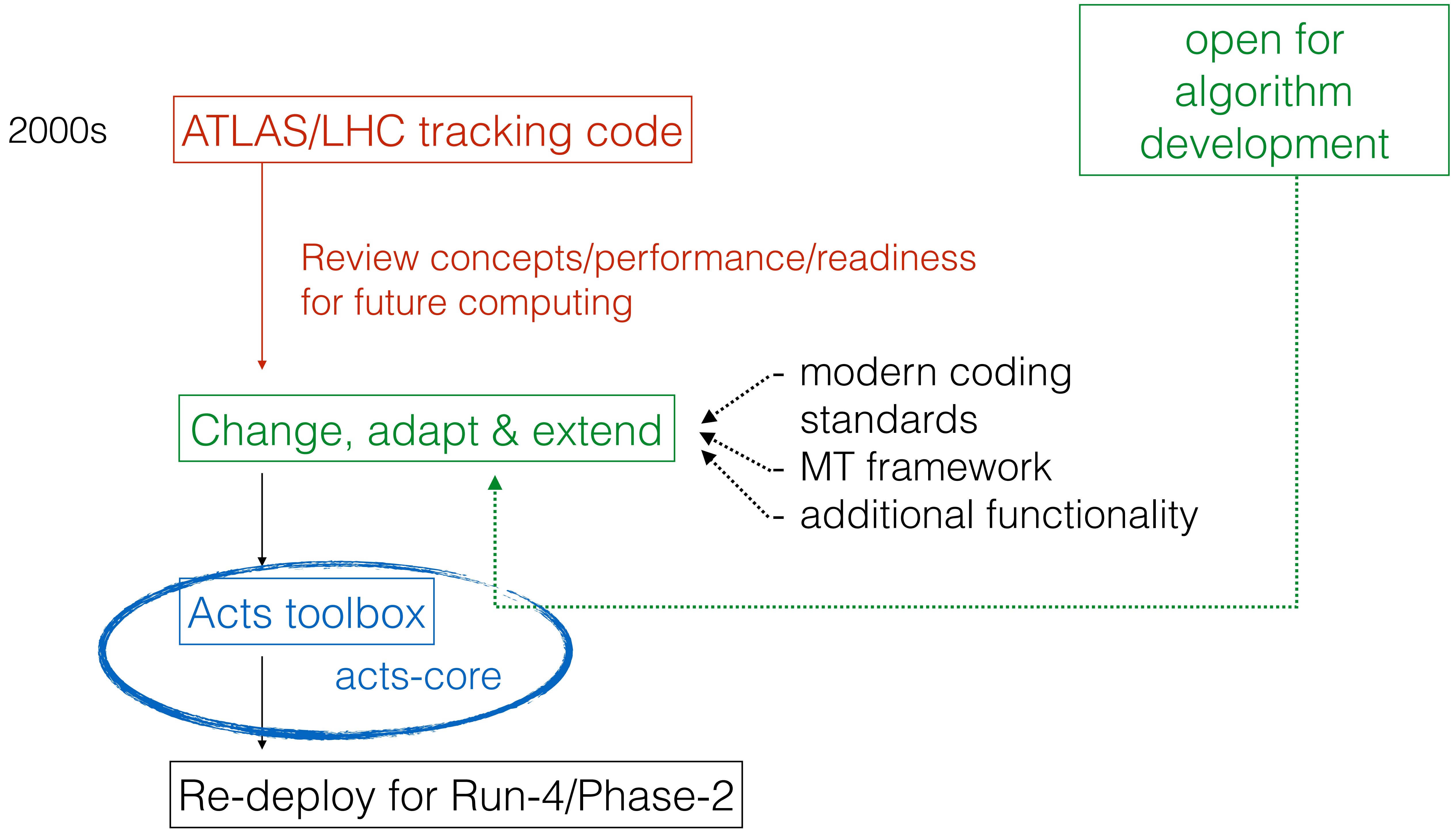
OpenData Pixel detector



Features:

- described in DD4Hep
- realistic material budget
- non-symmetric in azimuthal angle
- full (G4) and fast (ACTS) simulation
- misalignment possibility

Mission statement

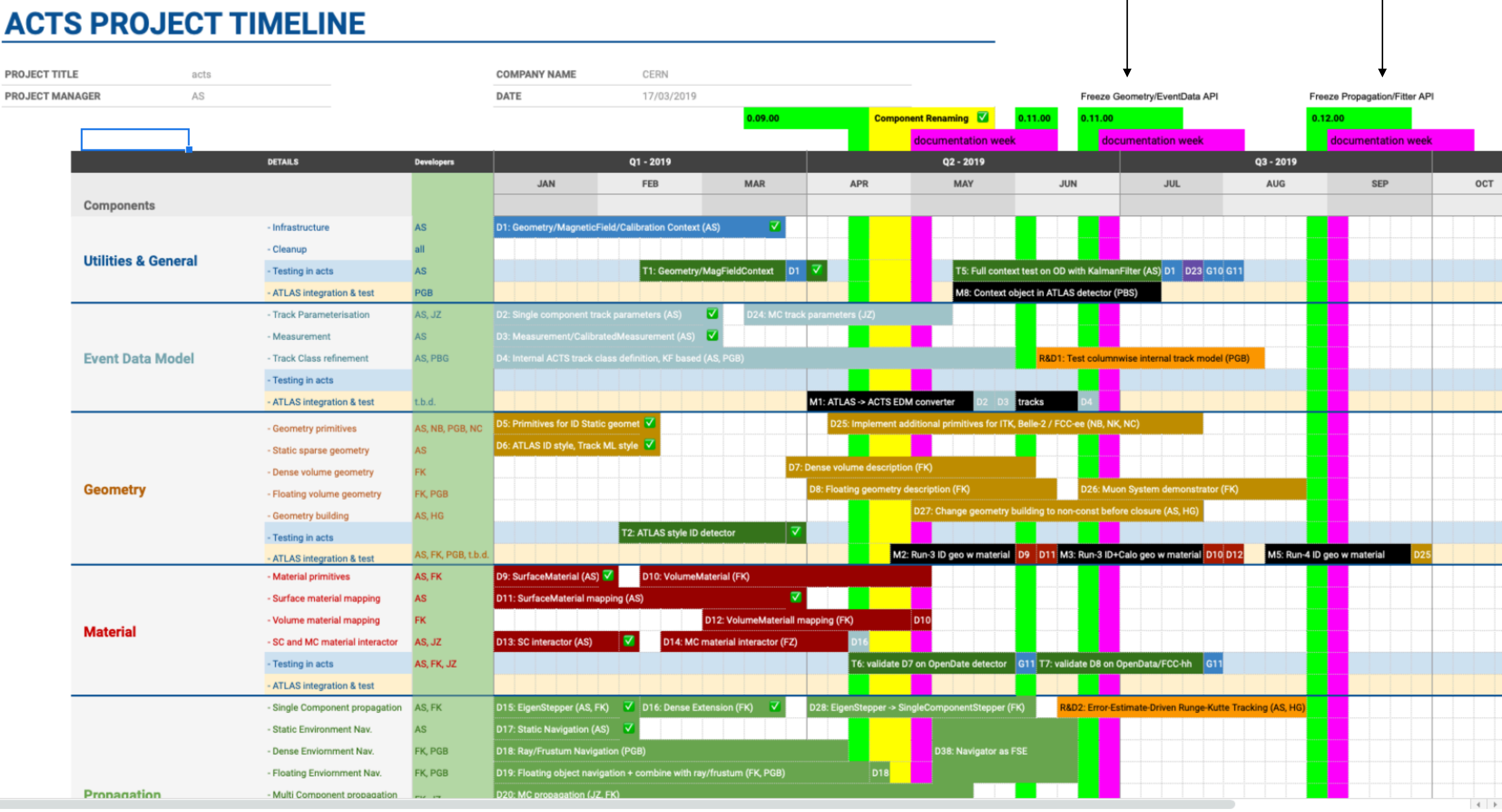

2000s
ATLAS/LHC tracking code
Review concepts/performance/readiness for future computing
Change, adapt & extend
- modern coding standards
- MT framework
- additional functionality
open for algorithm development
Acts toolbox
acts-core
Re-deploy for Run-4/Phase-2

ATLAS: atlas/athena/Tracking/Acts

acts-core repository

freeze geometry & EDM API

freeze propagation API



ACTS PROJECT TIMELINE

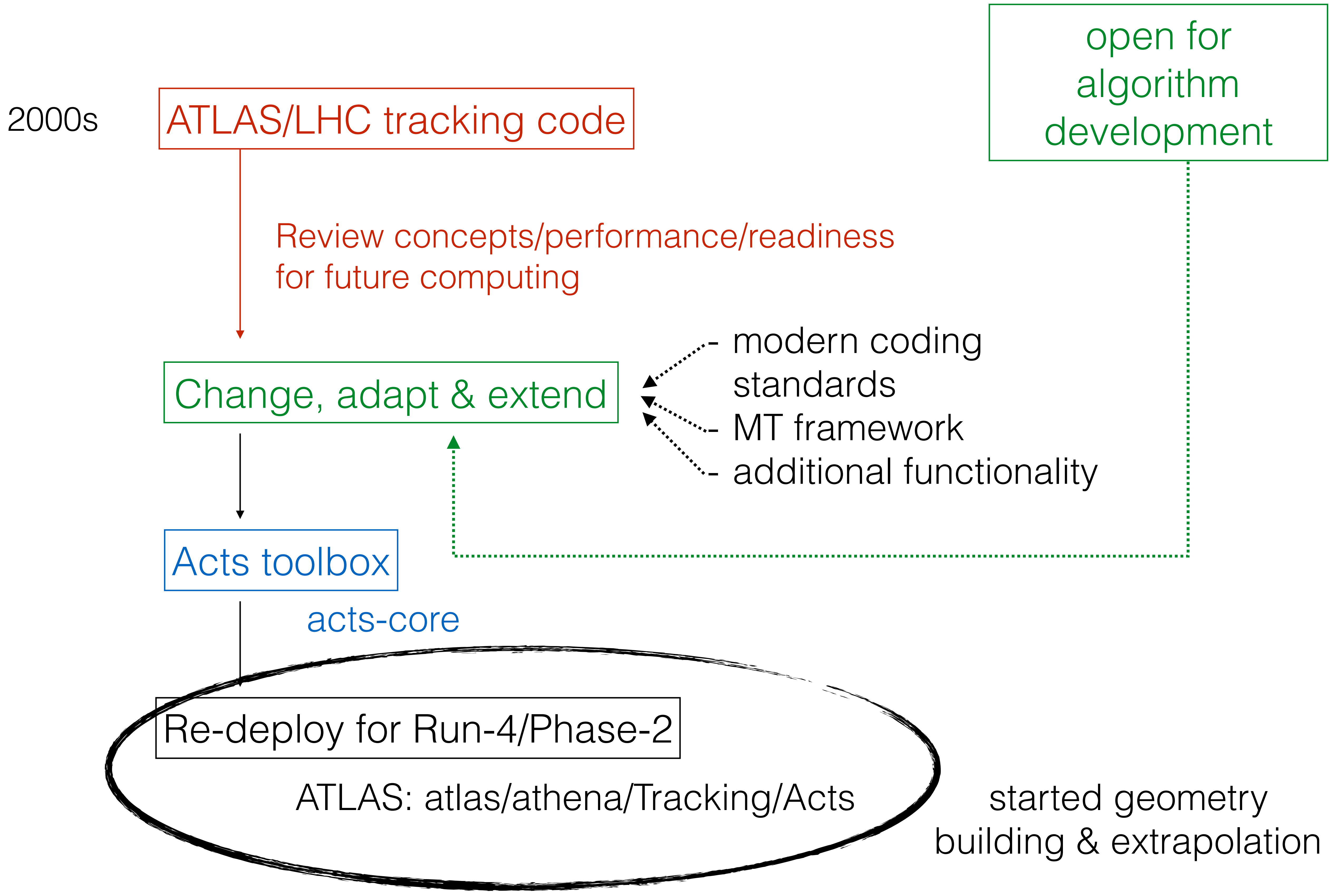
PROJECT TITLE	acts	COMPANY NAME	CERN
PROJECT MANAGER	AS	DATE	17/03/2019

Milestones: 0.09.00 · Component Renaming ✓ · documentation week · 0.11.00 · Freeze Geometry/EventData API · 0.11.00 · documentation week · Freeze Propagation/Fitter API · 0.12.00 · documentation week

Q1 - 2019: JAN, FEB, MAR · Q2 - 2019: APR, MAY, JUN · Q3 - 2019: JUL, AUG, SEP · OCT

Components	DETAILS	Developers	Tasks
Utilities & General	- Infrastructure	AS	D1: Geometry/MagneticField/Calibration Context (AS) ✓
	- Cleanup	all	
	- Testing in acts	AS	T1: Geometry/MagFieldContext; D1 ✓; T5: Full context test on OD with KalmanFilter (AS); D1; D23; G10; G11
	- ATLAS integration & test	PGB	M8: Context object in ATLAS detector (PBS)
Event Data Model	- Track Parameterisation	AS, JZ	D2: Single component track parameters (AS) ✓; D24: MC track parameters (JZ)
	- Measurement	AS	D3: Measurement/CalibratedMeasurement (AS) ✓
	- Track Class refinement	AS, PBG	D4: Internal ACTS track class definition, KF based (AS, PGB); R&D1: Test columnwise internal track model (PGB)
	- Testing in acts		
	- ATLAS integration & test	t.b.d.	M1: ATLAS -> ACTS EDM converter; D2; D3; tracks; D4
Geometry	- Geometry primitives	AS, NB, PGB, NC	D5: Primitives for ID Static geomet ✓; D25: Implement additional primitives for ITK, Belle-2 / FCC-ee (NB, NK, NC)
	- Static sparse geometry	AS	D6: ATLAS ID style, Track ML style ✓
	- Dense volume geometry	FK	D7: Dense volume description (FK)
	- Floating volume geometry	FK, PGB	D8: Floating geometry description (FK); D26: Muon System demonstrator (FK)
	- Geometry building	AS, HG	D27: Change geometry building to non-const before closure (AS, HG)
	- Testing in acts		T2: ATLAS style ID detector ✓
	- ATLAS integration & test	AS, FK, PGB, t.b.d.	M2: Run-3 ID geo w material; D9; D11; M3: Run-3 ID+Calo geo w material; D10; D12; M5: Run-4 ID geo w material; D25
Material	- Material primitives	AS, FK	D9: SurfaceMaterial (AS) ✓; D10: VolumeMaterial (FK)
	- Surface material mapping	AS	D11: SurfaceMaterial mapping (AS) ✓
	- Volume material mapping	FK	D12: VolumeMateriall mapping (FK); D10
	- SC and MC material interactor	AS, JZ	D13: SC interactor (AS) ✓; D14: MC material interactor (FZ); D16
	- Testing in acts	AS, FK, JZ	T6: validate D7 on OpenDate detector; G11; T7: validate D8 on OpenData/FCC-hh; G11
	- ATLAS integration & test		
Propagation	- Single Component propagation	AS, FK	D15: EigenStepper (AS, FK) ✓; D16: Dense Extension (FK) ✓; D28: EigenStepper -> SingleComponentStepper (FK); R&D2: Error-Estimate-Driven Runge-Kutte Tracking (AS, HG)
	- Static Environment Nav.	AS	D17: Static Navigation (AS) ✓
	- Dense Enviornment Nav.	FK, PGB	D18: Ray/Frustum Navigation (PGB); D38: Navigator as FSE
	- Floating Enviornment Nav.	FK, PGB	D19: Floating object navigation + combine with ray/frustum (FK, PGB); D18
	- Multi Component propagation		D20: MC propagation (JZ, FK)

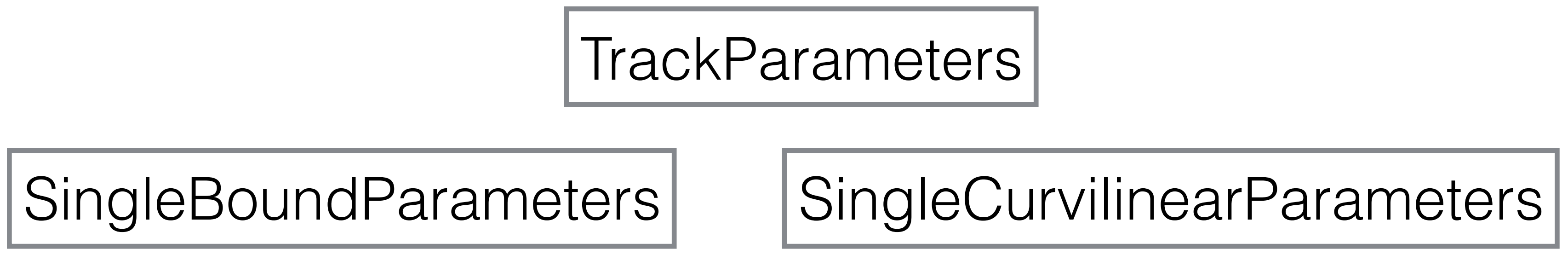
Mission statement



2000s
ATLAS/LHC tracking code
Review concepts/performance/readiness for future computing
Change, adapt & extend
- modern coding standards
- MT framework
- additional functionality
open for algorithm development
Acts toolbox
acts-core
Re-deploy for Run-4/Phase-2
ATLAS: atlas/athena/Tracking/Acts
started geometry building & extrapolation

Backup

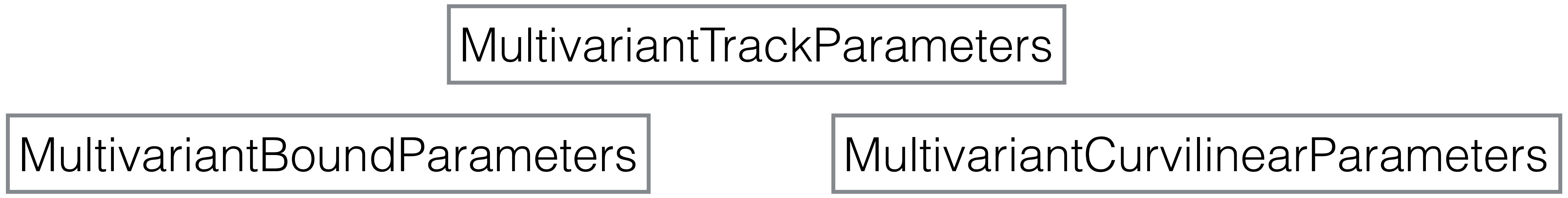
Event Data Model Track Parameters



Extension for Multi Component representation

- avoid copying of Extrapolator (as done in ATLAS) and Fitter infrastructure for multi-variant fitters (MultiTrackFitter, GSF)

  act as single track parameters in navigation, but will be propagated as multiple components in between

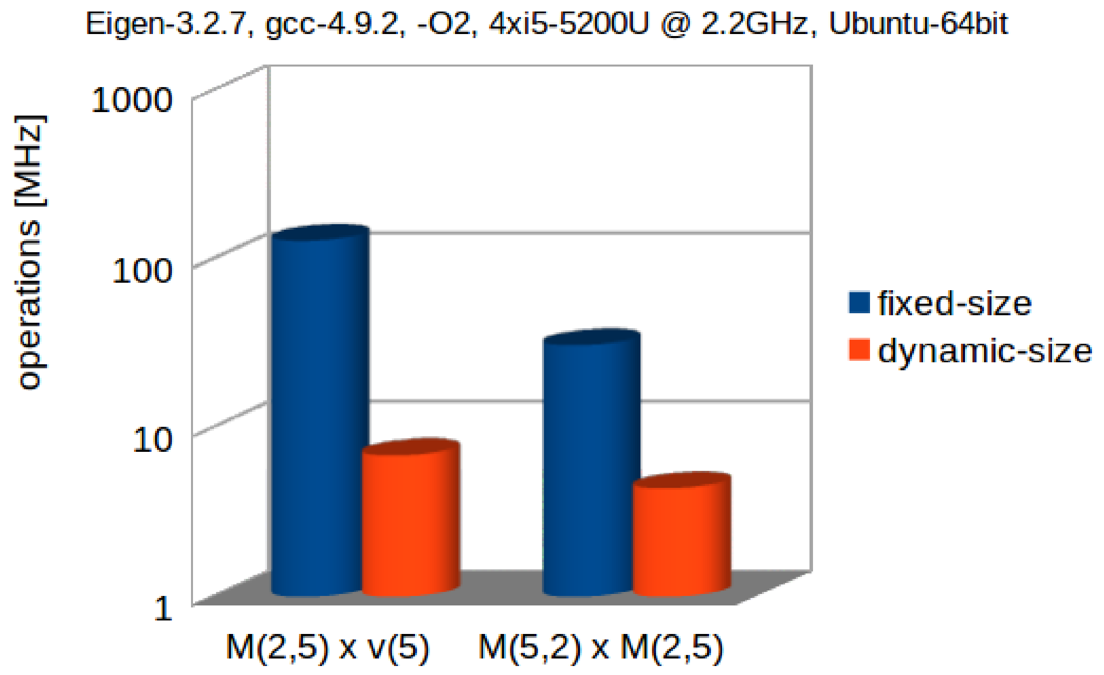

MultivariantTrackParameters
MultivariantBoundParameters
MultivariantCurvilinearParameters

Event Data Model Measurements

Fixed size matrix operations are evidently faster

- Acts EDM uses fixed-size
- needs container for heterogenous measurements:

  *e.g. PixelCluster (*2D*), StripCluster (*1D*), Segment (*4D*), how to combine them in a track class or containers ?*



currently using **`std::variant<>`**

Investigating a more xAOD type storage in the background (MR open for testing)

Status Binding to detector software & framework

Acts designed to have minimal overhead when being integrated in detector software

Algebra library is Eigen but dependencies are minimal

- may change to a template implementation (if beneficial)

No dependency on Identifier

- Detector calibration is resolved in detector geometry

Screen logging can be replaced by ld pre-loading

- needs a simple struct on the detector framework side that provides a logger() method.
- tested with different loggers:
  - Acts logger in acts-framework
  - Gaudi logger within FCCSW

Status Binding to framework configuration

ACTS tools have a nested configuration struct:

```
namespace Acts {
  /// doxygen documentation
  class WorkHorse {
    /// @struct Config for To
    struct Config {
       float coatColor; ///< configure the coat color
       float maxPath;    ///< set the max path this horse can run
    };
  };
}
```

These structs are then configured by the detector framework, e.g. through Gaudi/Athena

```
/// feed from Framework into ACTS configuration
declareProperty(“CoatColor”, m_cfg.coatColor);
declareProperty(“MaxPath”,   m_cfg.maxPath);
```

tested with Gaudi for FCCSW & AthenaMT

Configuration Strategy

Nested configuration struct by convention

```
namespace Acts {
  /// doxygen documentation
  class SomeComponent {
    /// @struct Config for this Component
    struct Config {
       bool run_faster = false; ///< configuration flag
    };
  /// Constructor with config object
   SomeComponent(Config& cfg);
 };
}
```

Inside the framework Wrapper

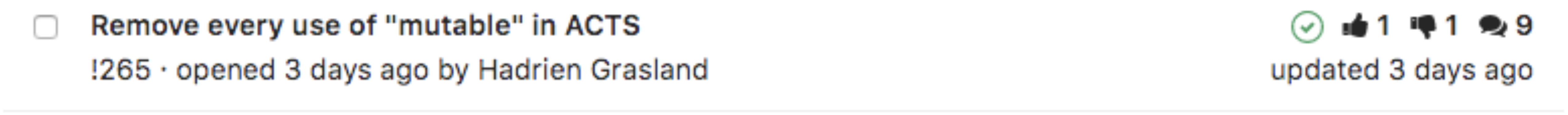
```
#include “ACTS/Package/SomeComponent.hpp”

…
  /// create the config sruct
  Acts::SomeComponent::Config scConfig;

  /// bind to your framework configuration
  declareProperty(“RunFastVersion”, scConfig.run_faster);
   Acts::SomeComponent sc(scConfig);
```

Concurrency Strategy

const-correctness



☐ **Remove every use of "mutable" in ACTS**
!265 · opened 3 days ago by Hadrien Grasland

✓ 👍 1 👎 1 💬 9
updated 3 days ago

statelessness engines

- cache visitor pattern for calls that need to run concurrently

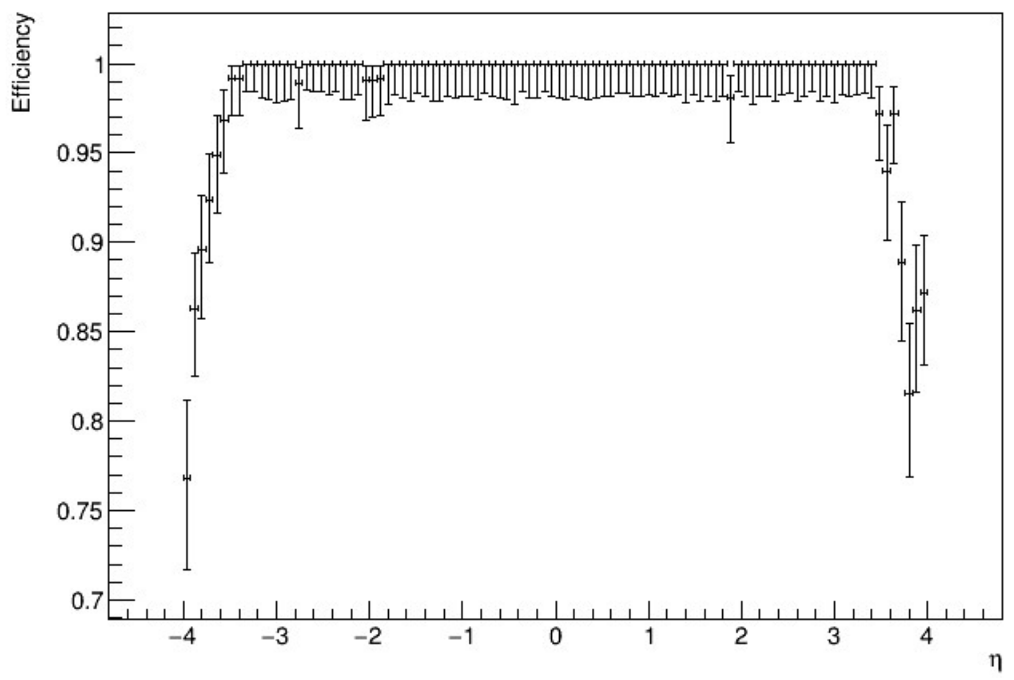
```
namespace Acts {
  /// doxygen documentation
  class WorkHorse {
    /// @struct Cache for the WorkHorse
    struct State {
       float accumulatedPath = 0.; ///< the passed path so far
    };
    /// method to make the horse run
    /// @param hState - cache tracker for this horse
    /// @param coords - place where the horse should run to
    /// @return a result, horse may drop dead if max path is reached
    const RunResult run(State& hState, const Vector3D& coords) const;
 };
}
```

Fitter Development & Validation

KalmanFitter prototype in validation currently

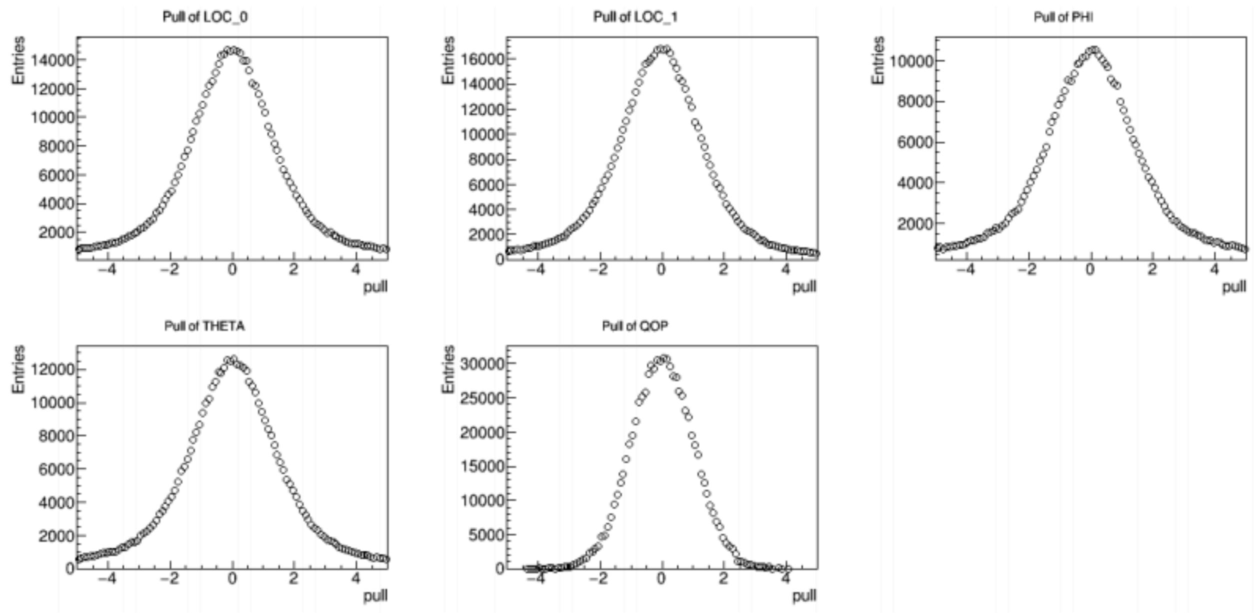
- multi-step validation program
    - *maths*
    - *material effects*
    - *transport*
    - *hole search*
- should exercise the full contextual chain including calibration

Fraction of smoothed track



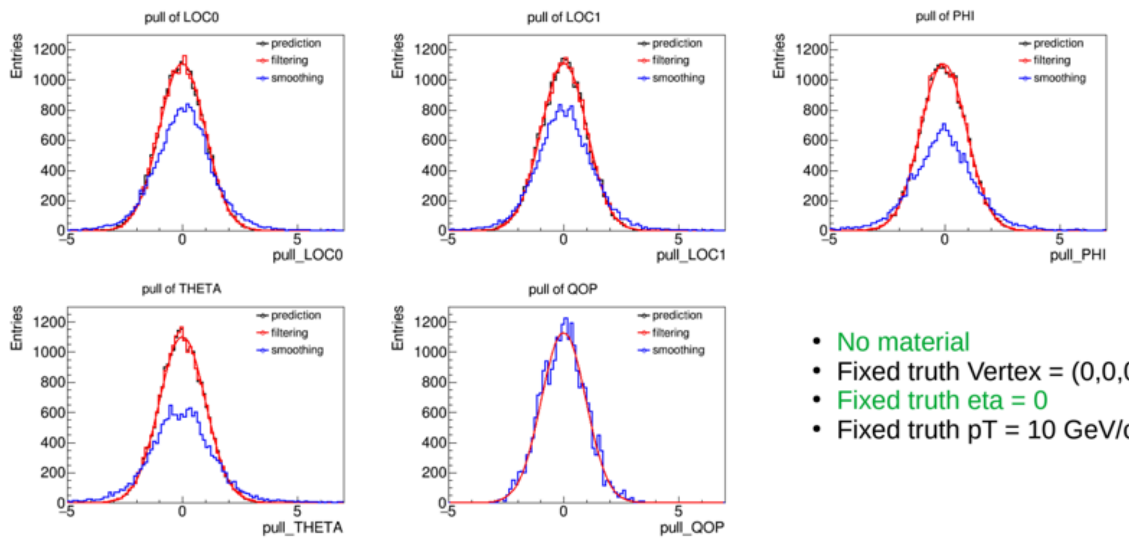
Pull distribution

50000 single muon samples at pT = 10 GeV/c with DD4hepGeometry



Pull of track parameter

Pulls have Gaussian(0,1) distributions when track is almost perpendicular to plane surface



- No material
- Fixed truth Vertex = (0,0,0
- Fixed truth eta = 0
- Fixed truth pT = 10 GeV/c