

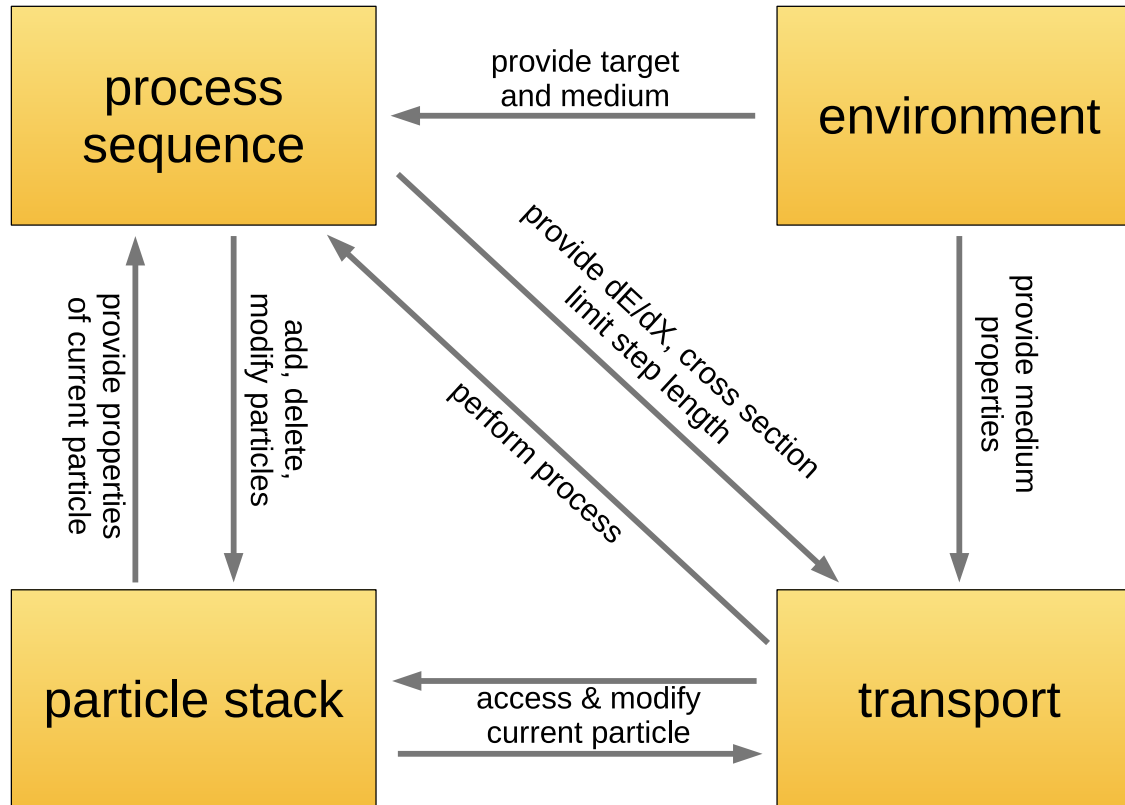
# Air shower simulations with CORSIKA 8

Maximilian Reininghaus



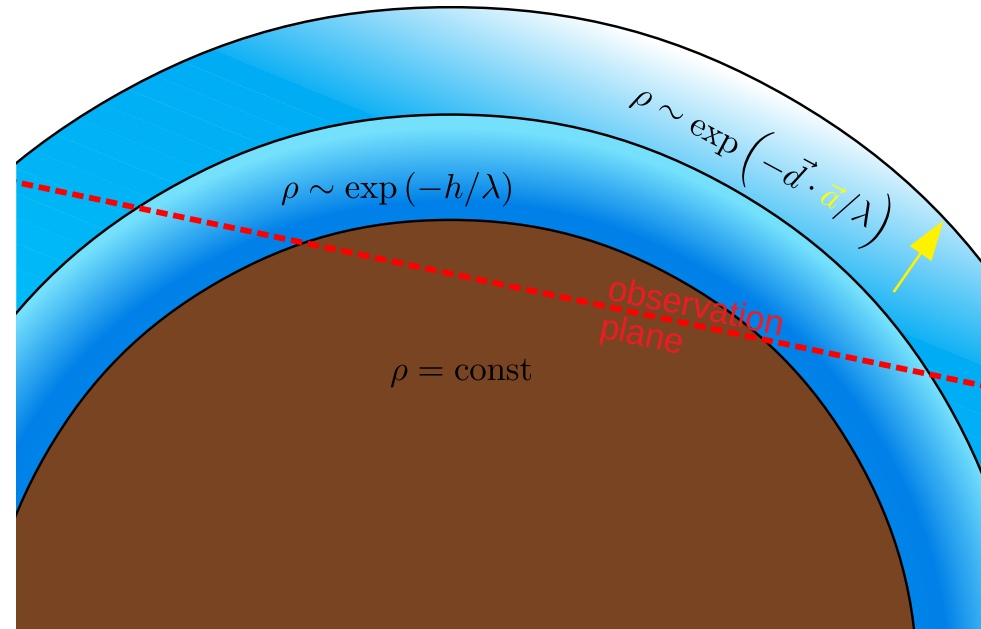
# Part I: Current status of the project

# Building blocks of C8



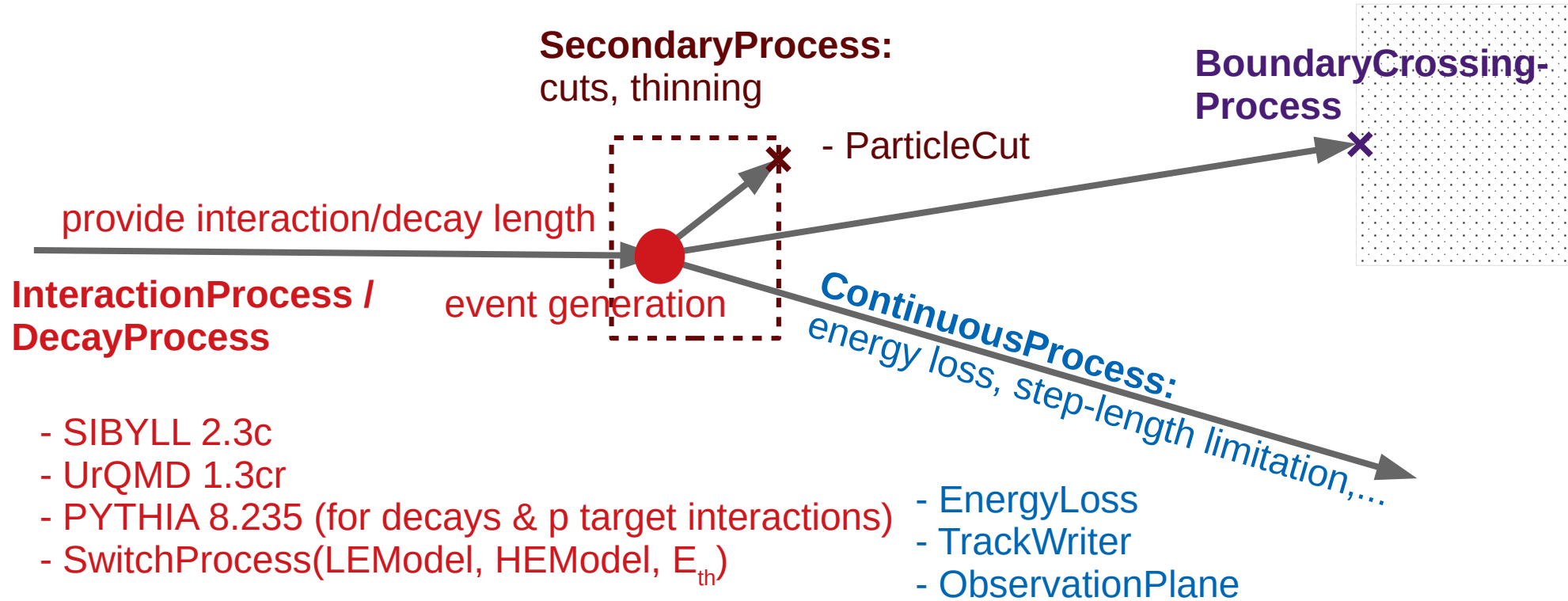
# Worldbuilding

- world entirely composable by user
- different materials/density models in different regions of space
- currently available:
  - homogeneous
  - exponential flat & curved (“sliding planar approximation”)
- easily extensible by own models



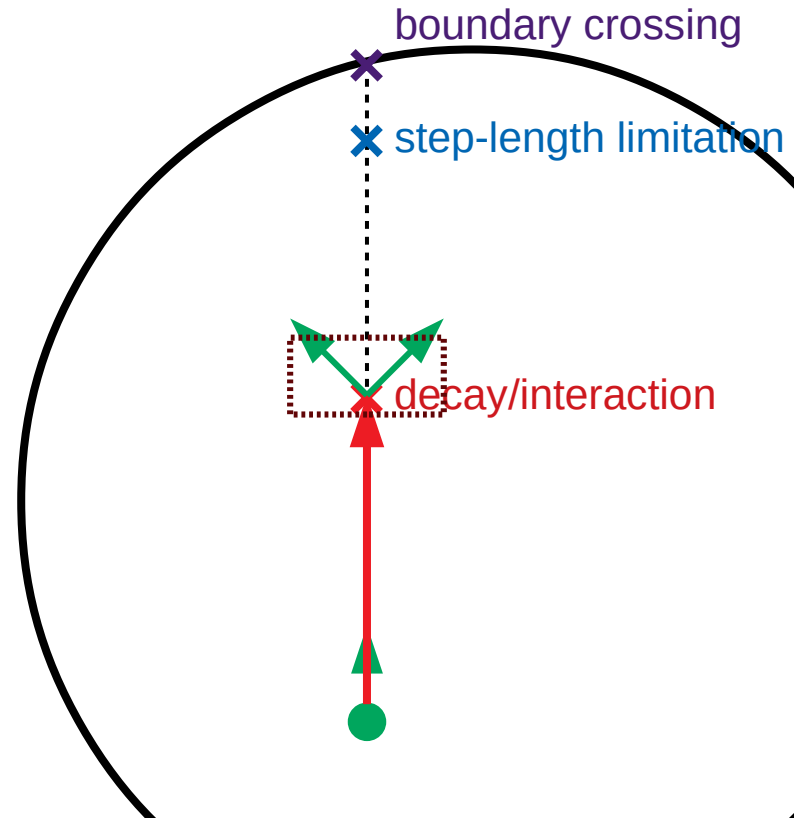


# Process classes



# Cascade step

- determine step-length
- apply continuous processes
- perform interaction/decay
- apply secondary processes



## Part II: First results & comparison

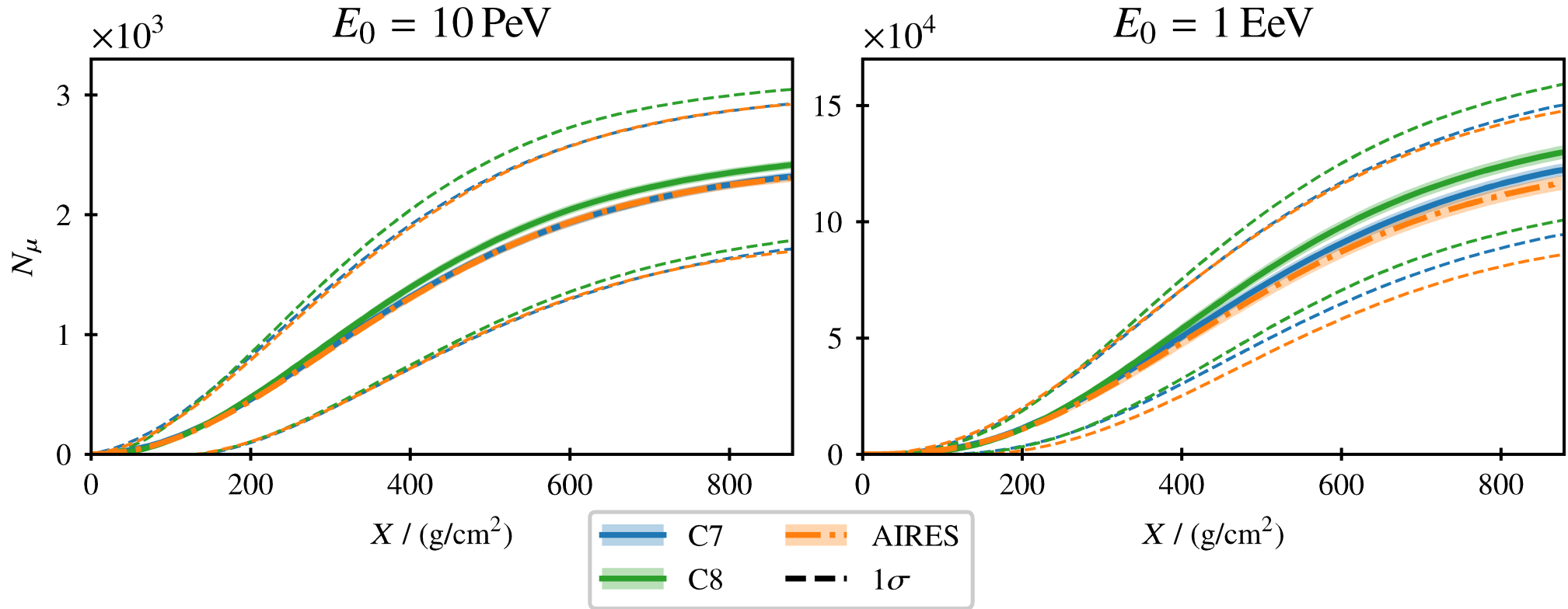
see D. Melo, MR, F. Riehn, R. Ulrich, PoS ICRC2019, 399

# Setup

- mostly equivalent setups for C8, C7.64, AIRES 19.04
- flat exponential atmosphere 878 g/cm<sup>2</sup>
- propagation of hadrons & muons
- vertical proton showers @ 10 PeV & 1 EeV

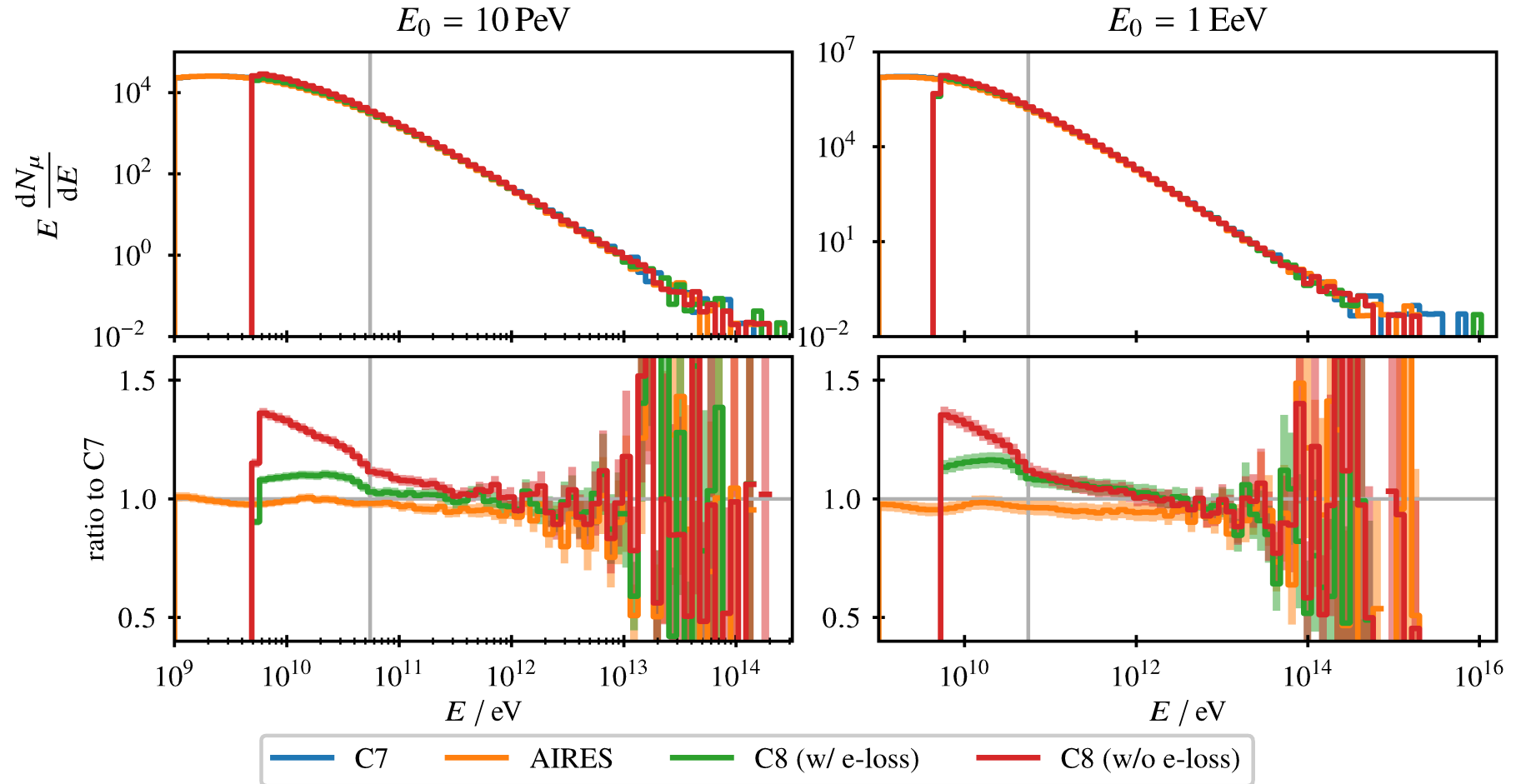
	HE model	LE model
CORSIKA 8	SIBYLL 2.3c	UrQMD 1.3
CORSIKA 7		
AIRES 18.10		Hillas Splitting Algorithm
	transition at <b>55 GeV</b>	

# Muon longitudinal profile



$E_\mu \geq 55 \text{ GeV}$

# Muon energy spectrum

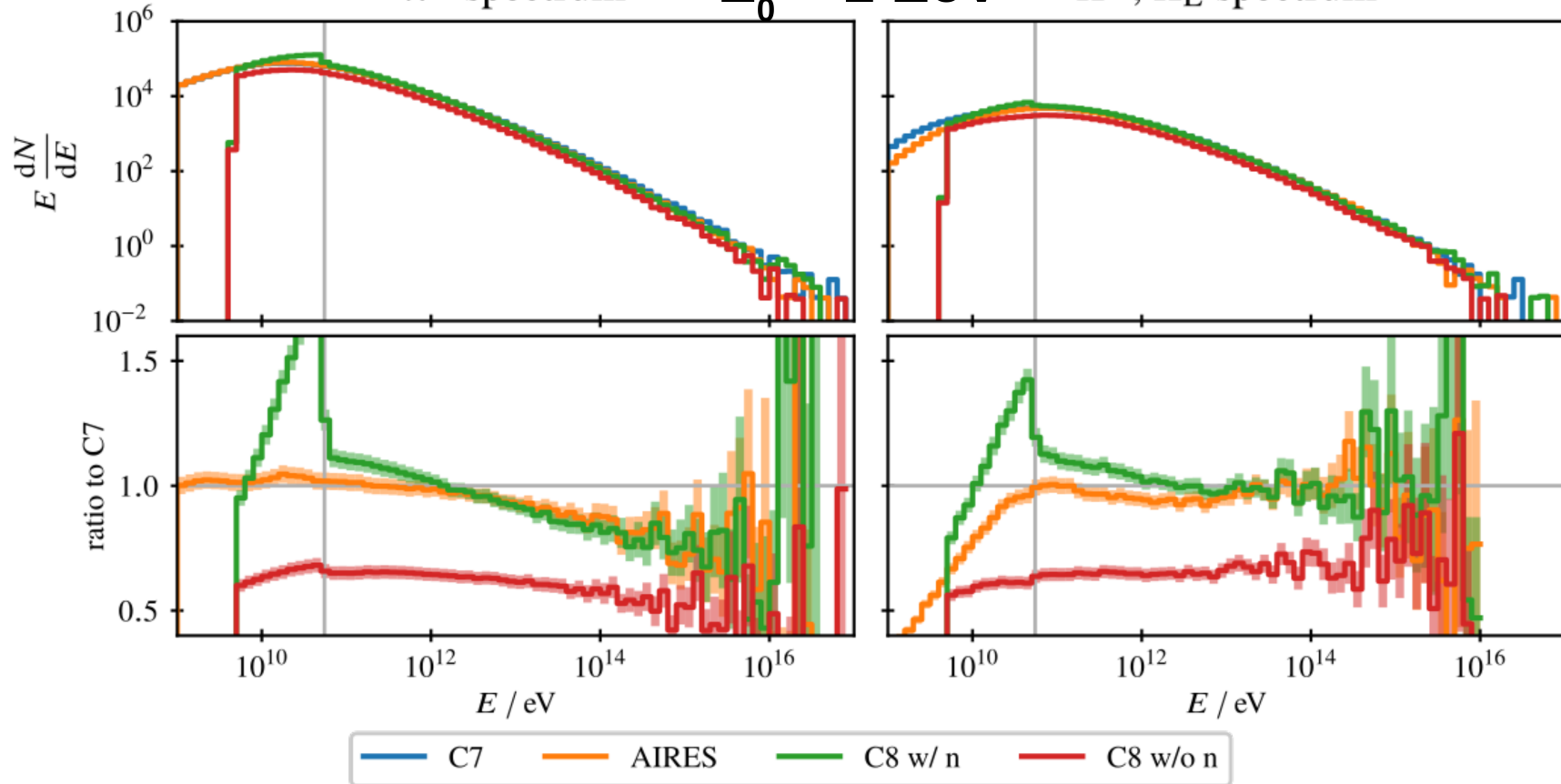


# Energy spectra at 7 km

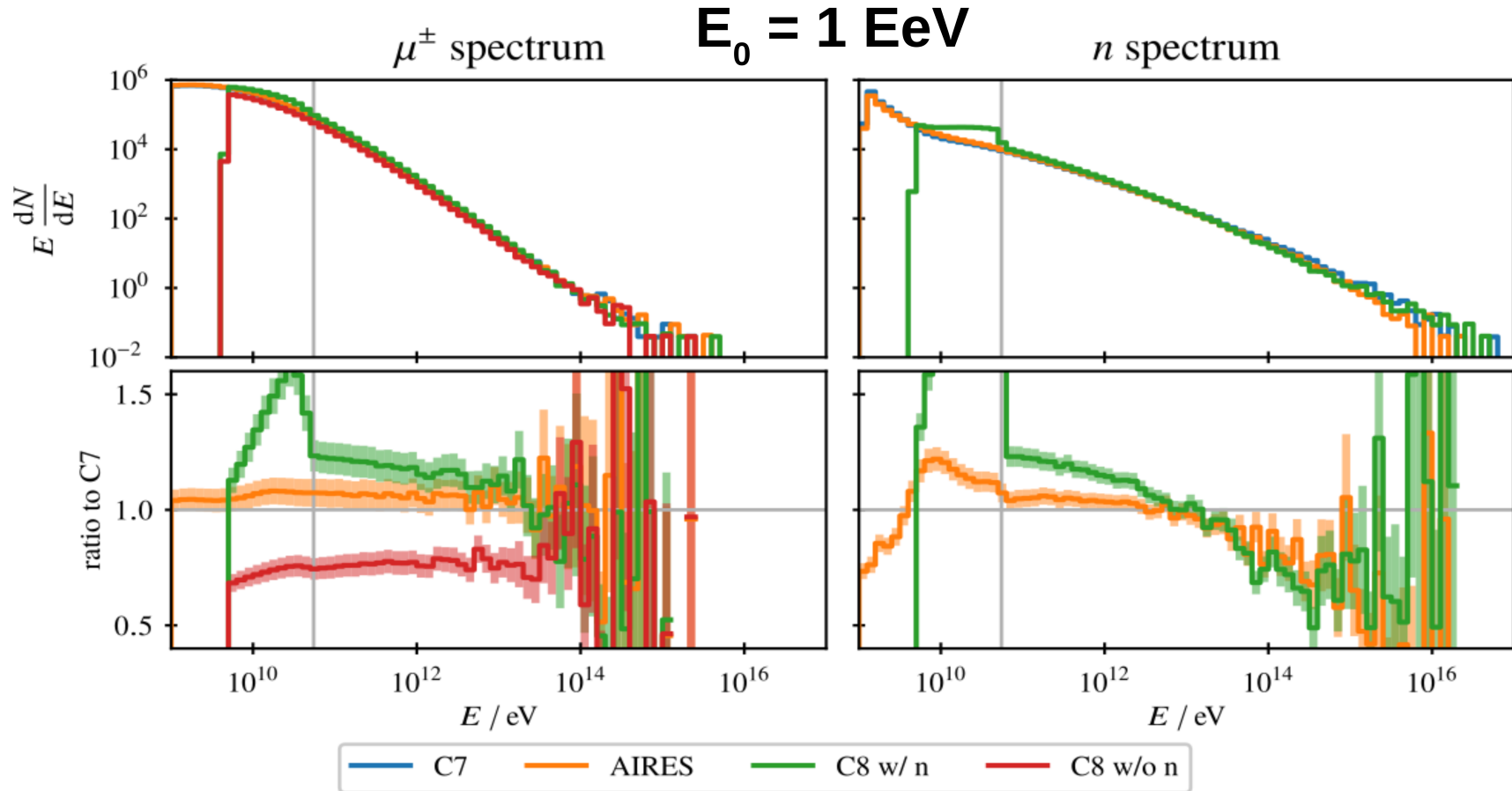
$\pi^\pm$  spectrum

$E_0 = 1 \text{ EeV}$

$K^\pm, K_L$  spectrum

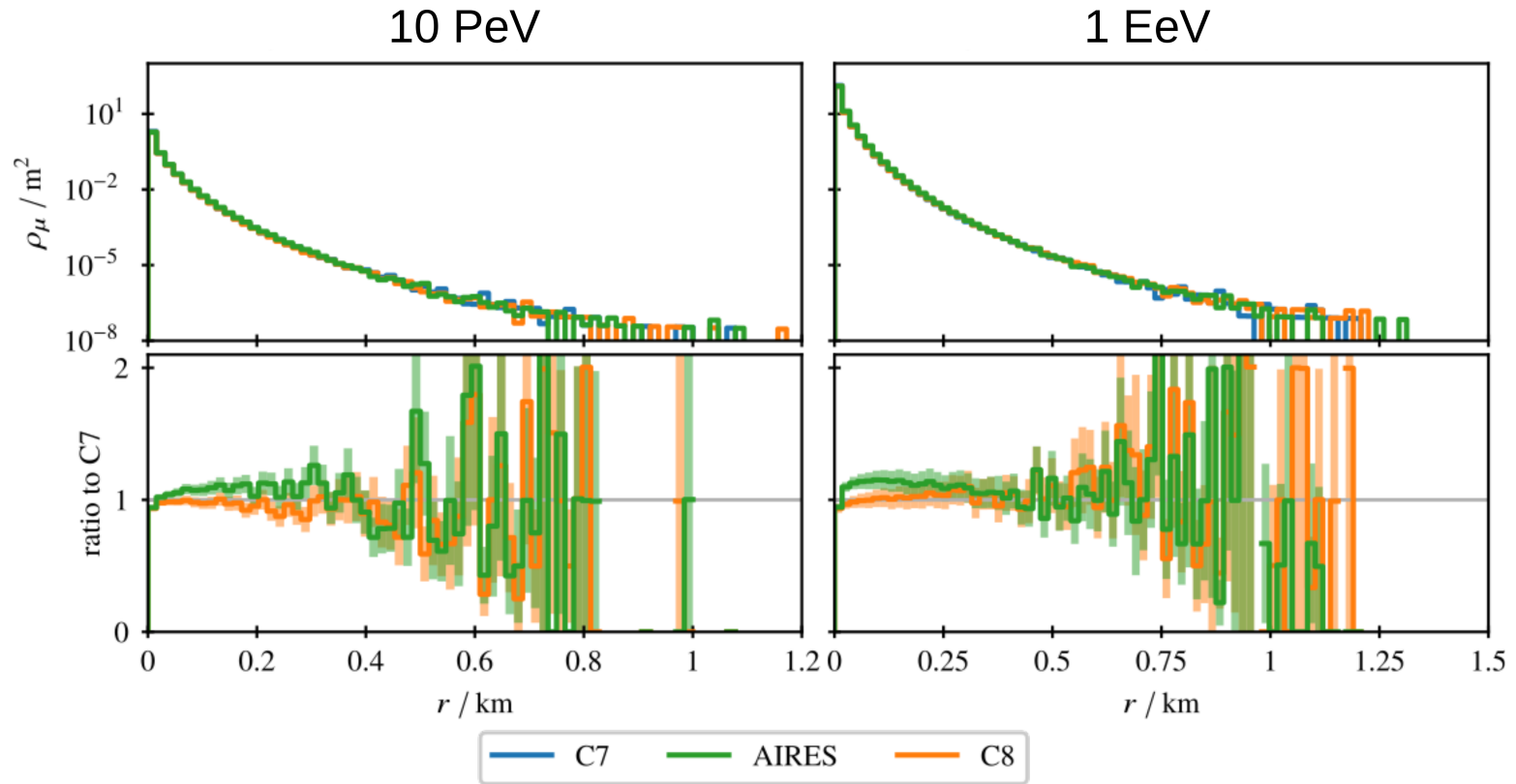


# Energy spectra at 7 km





# Muon LDF



$E_\mu \geq 55 \text{ GeV}$

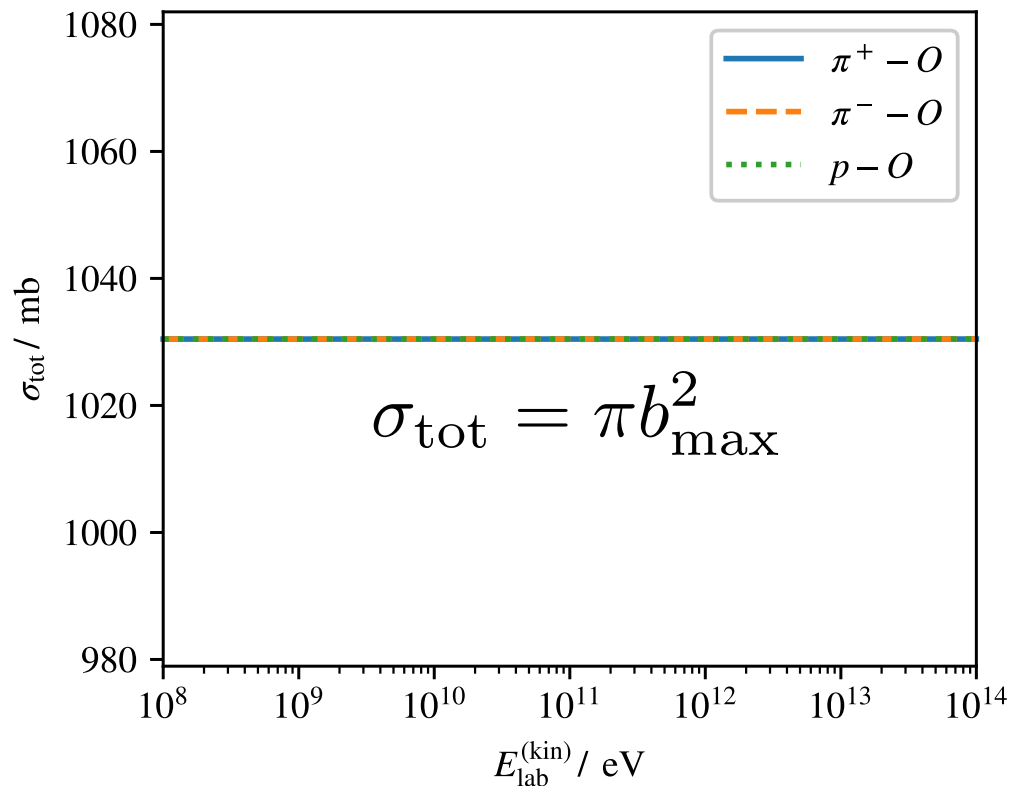
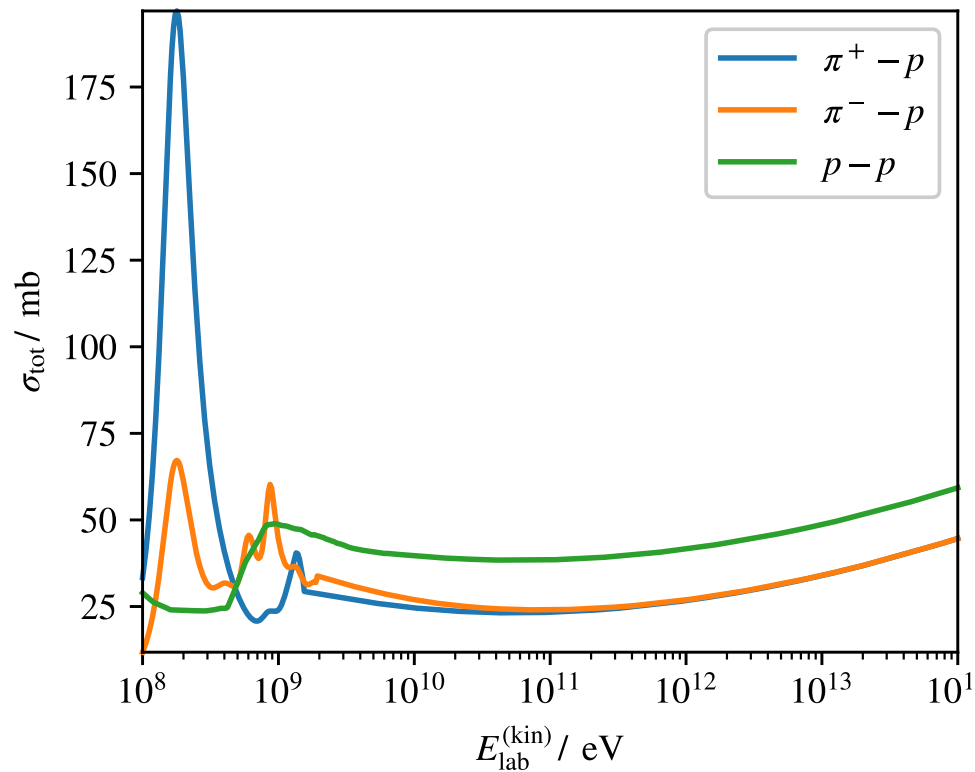
# Summary

- CORSIKA 8 is alive and kicking
- hadrons and muons in air already possible
- first results are comparable to other codes
- discrepancies will likely shed light on muon production

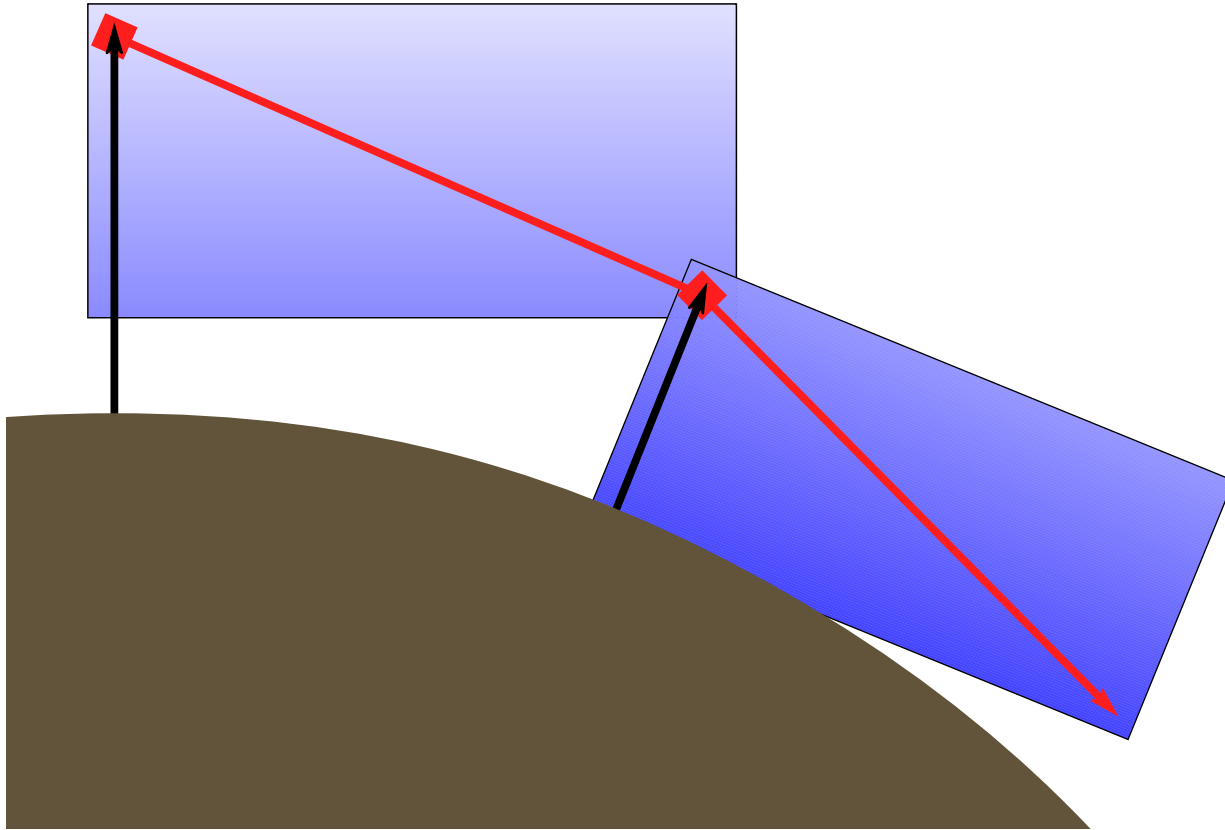


Backup

# UrQMD cross-sections

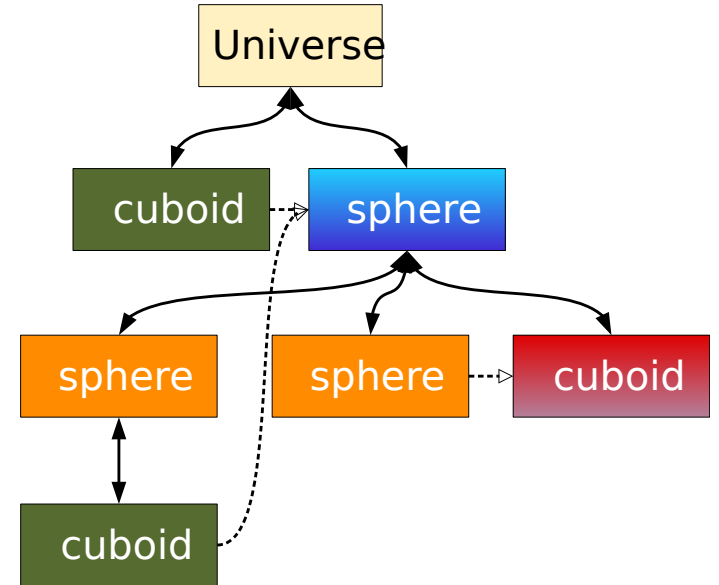
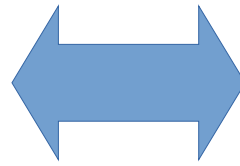
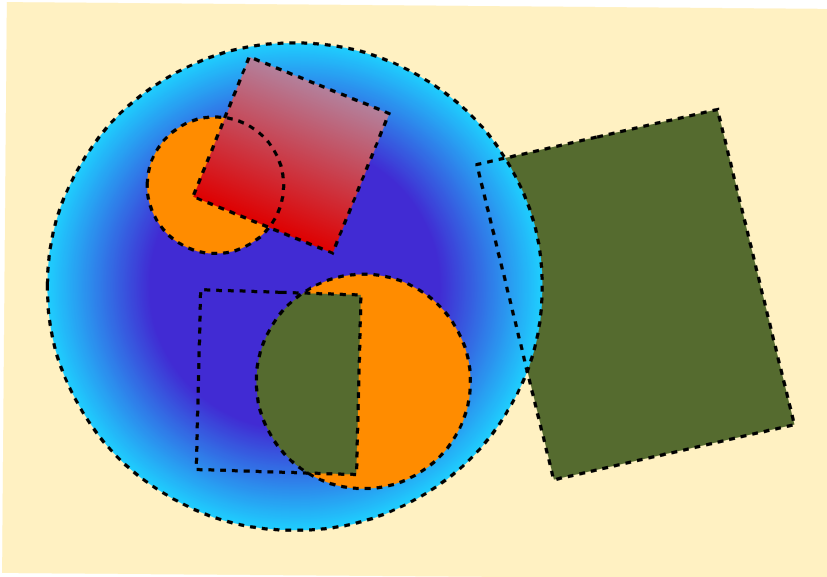


# Sliding planar approximation



# Worldbuilding

- geometric primitives furnished with material properties
- assembled in a tree representing containment



# under the hood

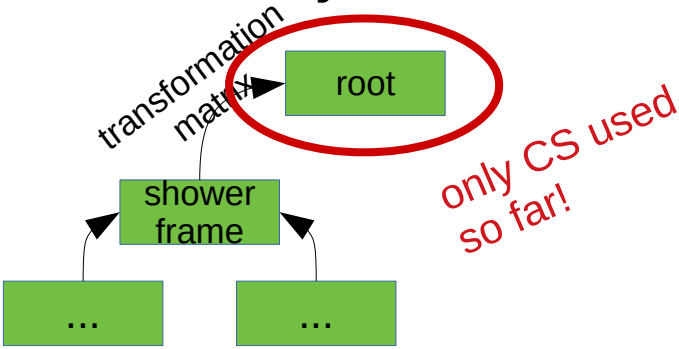
## statically typed units

→ compile-time dimensional analysis

```
MassDensityType rho = 4_g /  
    cube(cm);  
  
auto length = 5.4_m;  
auto X = length * rho  
// → GrammageType
```

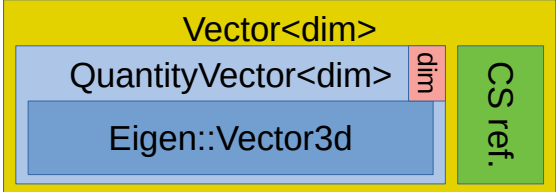
conversion between  
SI & natural units provided!

## coordinate systems



Definition of new CS by:  
- rotation and/or translation  
- reference CS

## Vectors & points



automatic transformations into  
common CS when necessary:  
`Vector<length> v(cs1,...),  
 w(cs2,...);`  
`auto u = v + w;`

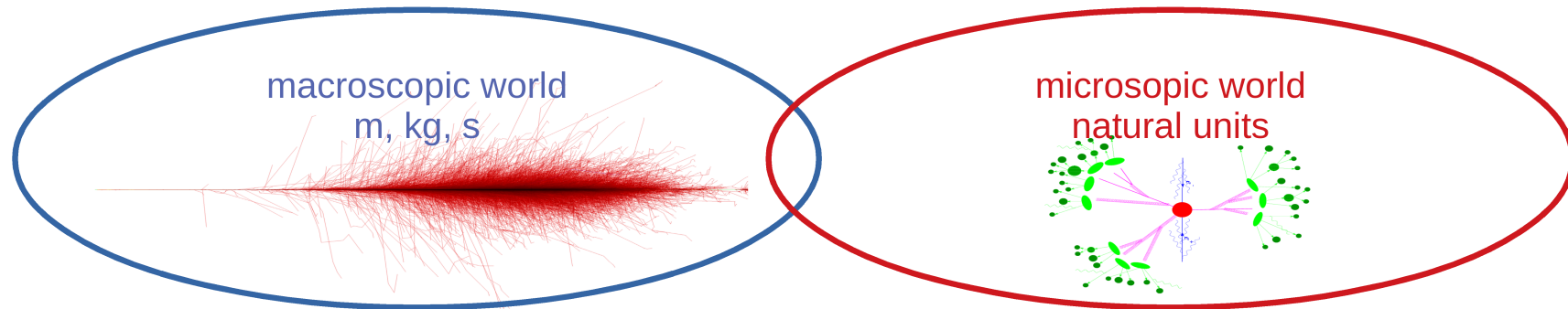
# Units

- Compile-time dimensional analysis based on *PhysUnits C++11*
- Example:
 

```

      MassDensityType rho = 4_g / cube(cm);
      auto length = 5.4_m;
      auto X = length * rho // GrammageType
      
```

 fully integrated into geometry framework: `Vector<speed_u>, ...`
- conversion between SI & natural units provided!





# Profiling

