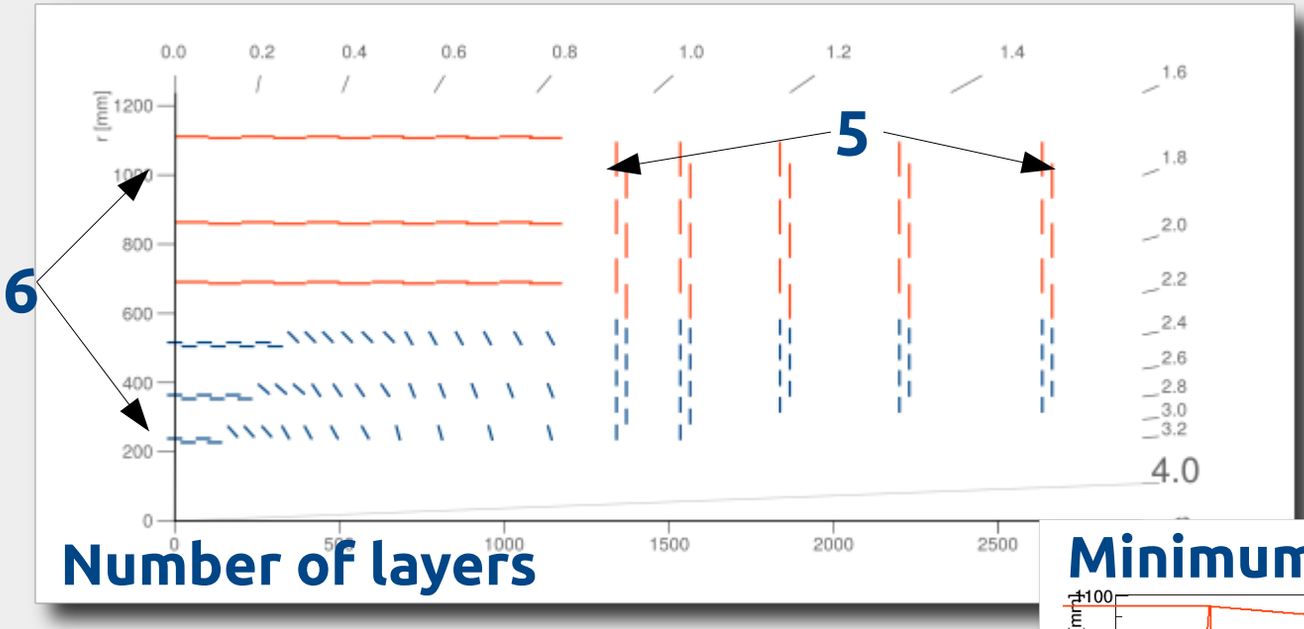
tkLayout
a tracker design and optimization tool

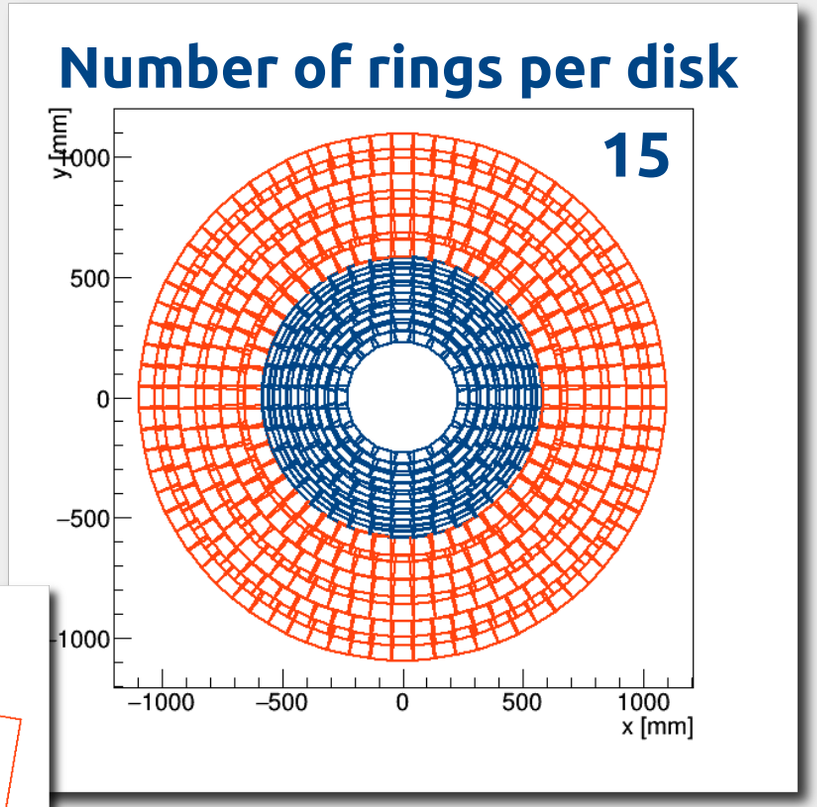
Stefano Mersi
Software R&D Second Lightning Talks Session
2018-02-26

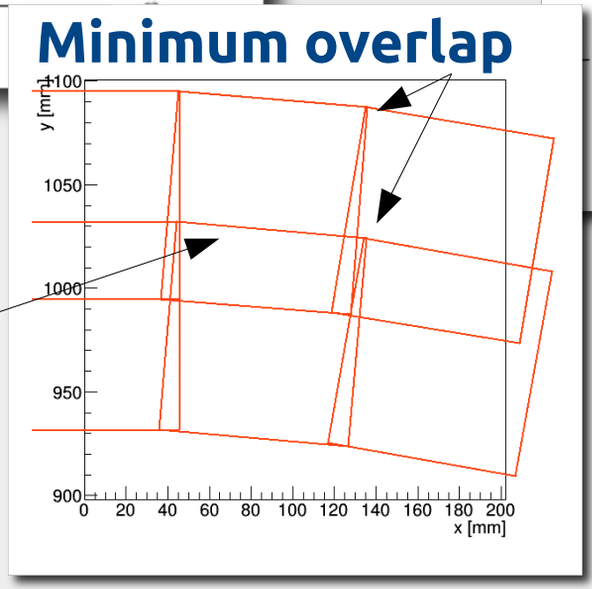
What
Why
How
What next?

tkLayout layout building

Based on a simple set of parameters

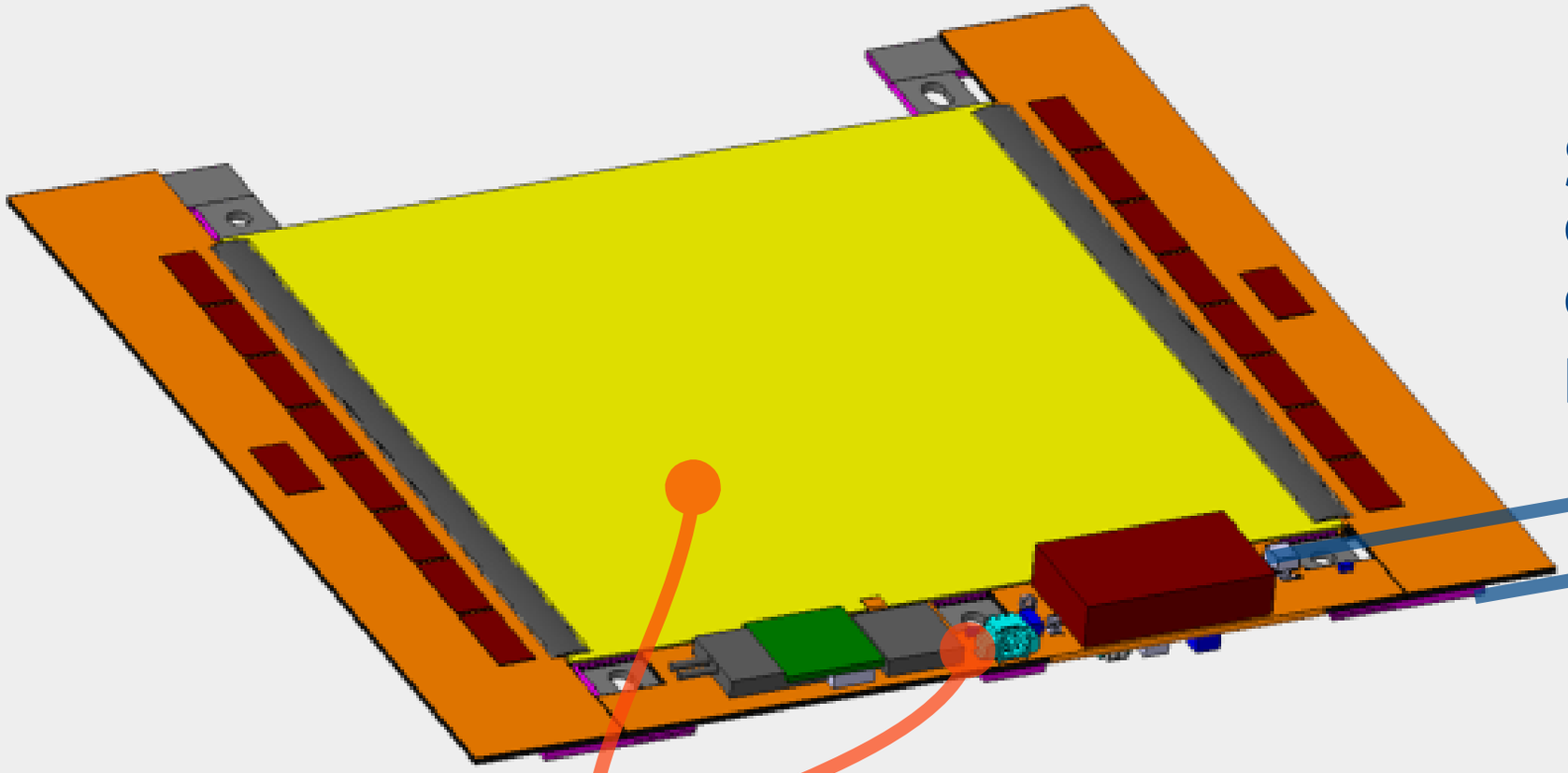






Sensor geometry
(e.g. square ~10×10 cm²)

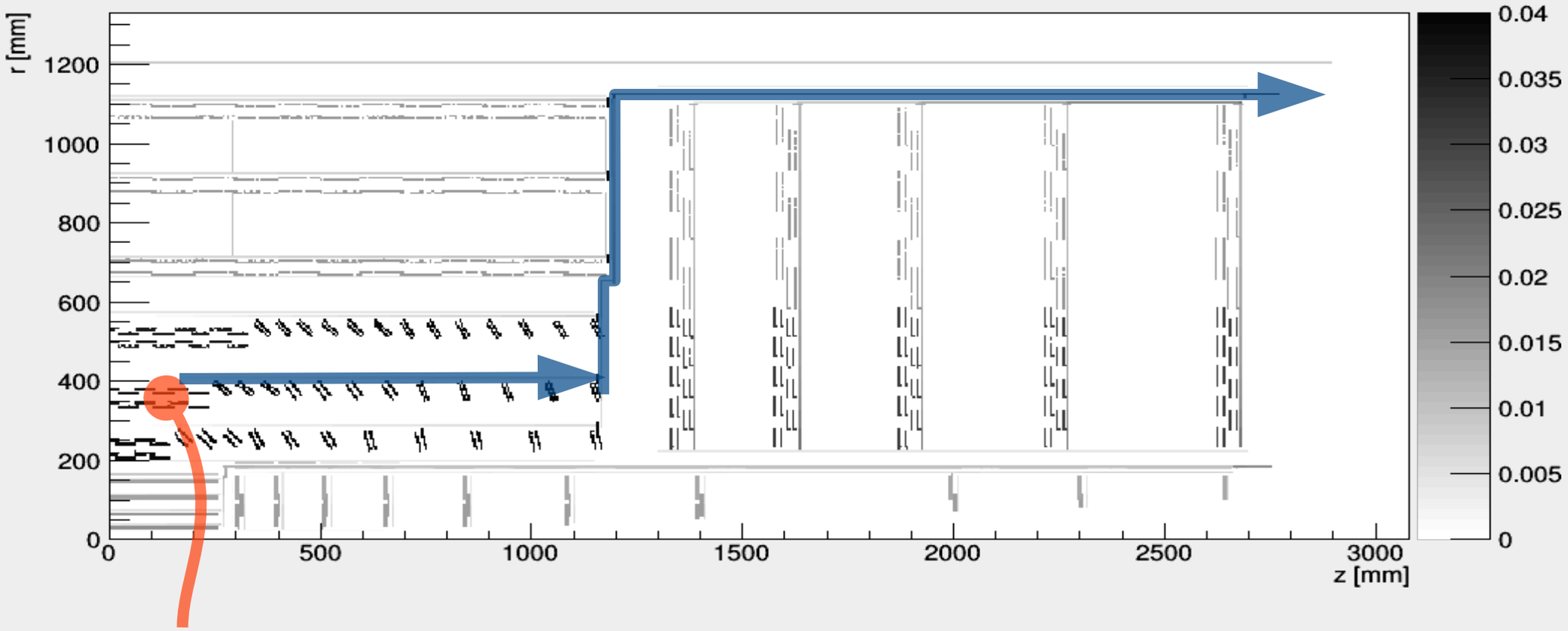
tkLayout material estimation



services
cooling pipes
optical fibres
power cables

on the module
sensor
hybrids & ASICs
cooling contacts

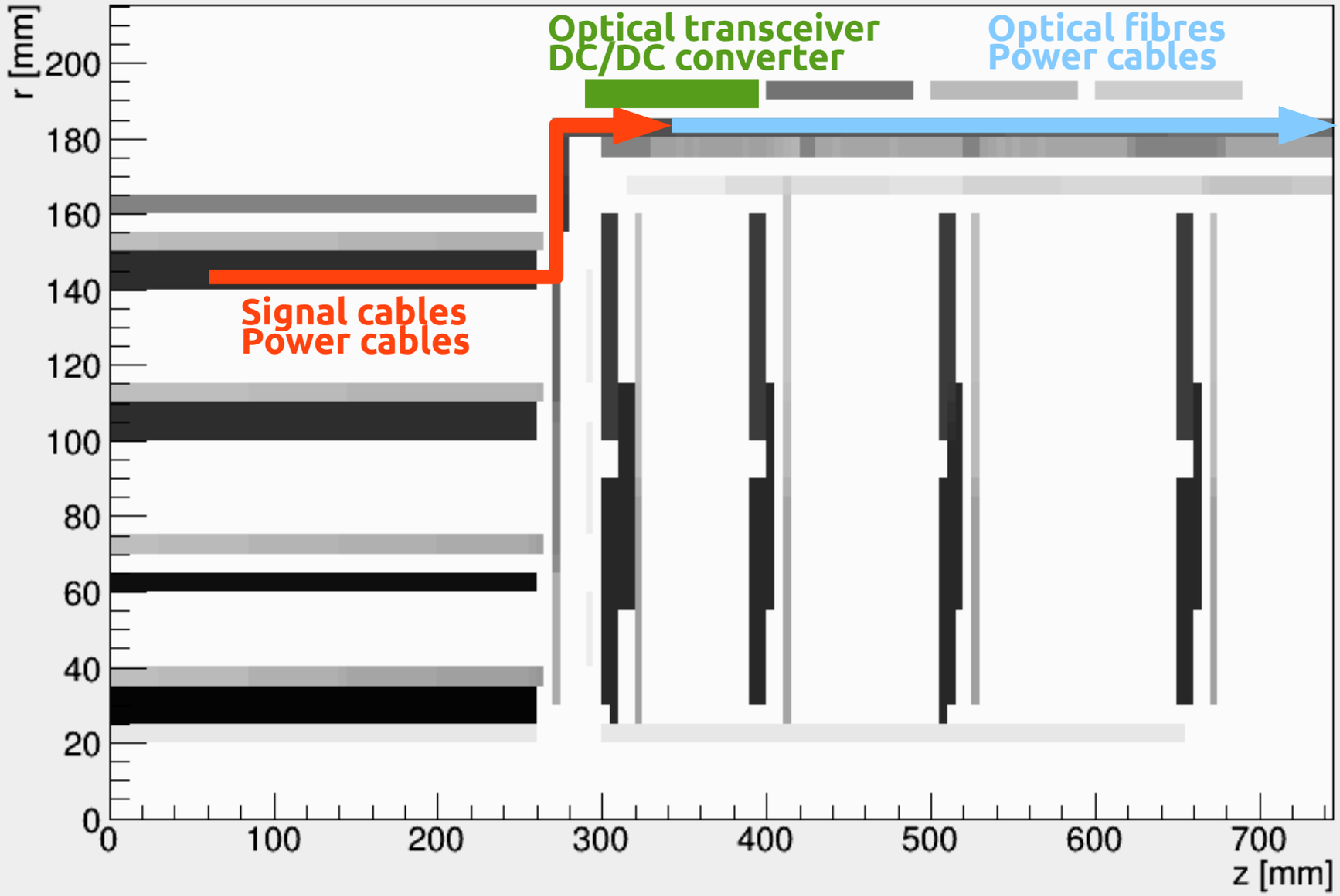
tkLayout material estimation



Material on active elements + Material for services automatically routed

tkLayout material estimation

Example: pixel detector v1.2

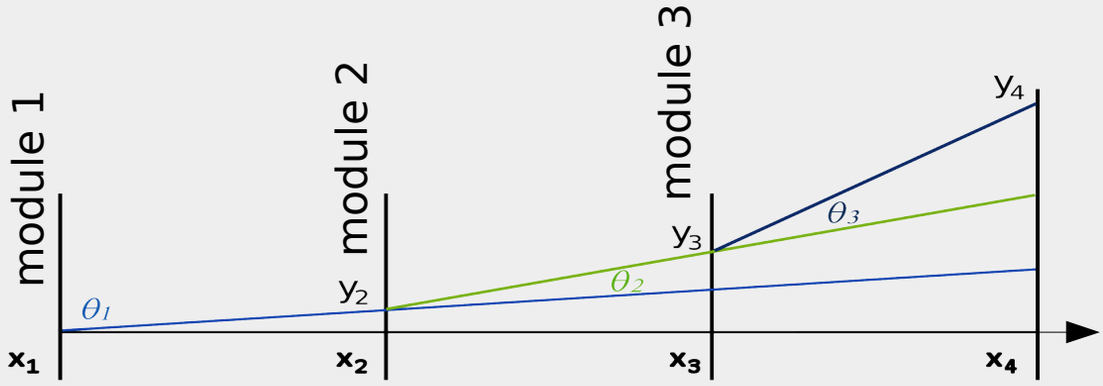


tkLayout μ resolution estimation

- A priori error estimation
  - No Monte Carlo
  - No fit actually done
- Error propagation to estimate resolution of track parameters
  - **Intrinsic resolution** of the measurement point
  - **Multiple scattering** treated as a (correlated) measurement error

$$\sigma_n^2 = \frac{p^2}{12} \quad \text{(or parametric)}$$

$$\sigma_{n,m} = \langle y_n y_m \rangle = \sum_{i=1}^{n-1} (x_m - x_i)(x_n - x_i) \langle \theta_i^2 \rangle$$



[1] V. Karimäki – CMS Note 1997/064 [NIM A410 (1998) 284] NIM A305 (1991) 187
[2] G. Hall – Calculating parameters for the Pixel and Tracker upgrade performance studies (Tracker Week) http://bit.ly/eXvi8L
[3] G. Bianchi – tkLayout: a design tool for innovative silicon tracking detectors JINST Volume 9, March 2014 http://iopscience.iop.org/article/10.1088/1748-0221/9/03/C03054/meta

Workflow

Simple cfg files

Few seconds to
few minutes

HTML pages
on disk

Why analytic approach

Full Monte-Carlo simulation

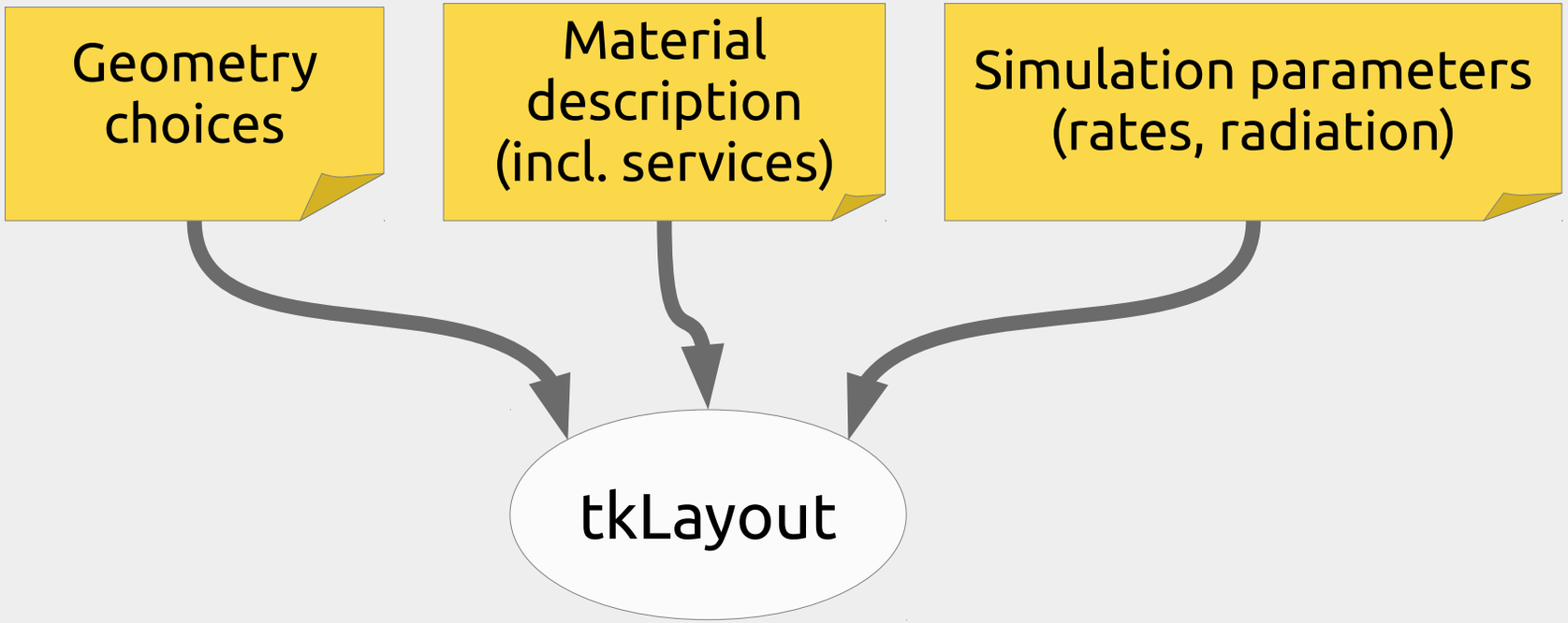
- Setup is time consuming
- Running is resource intensive
- Interpretation requires effort
- Depends on well-tuned algorithms
- Full picture of physics reach

Analytic approach

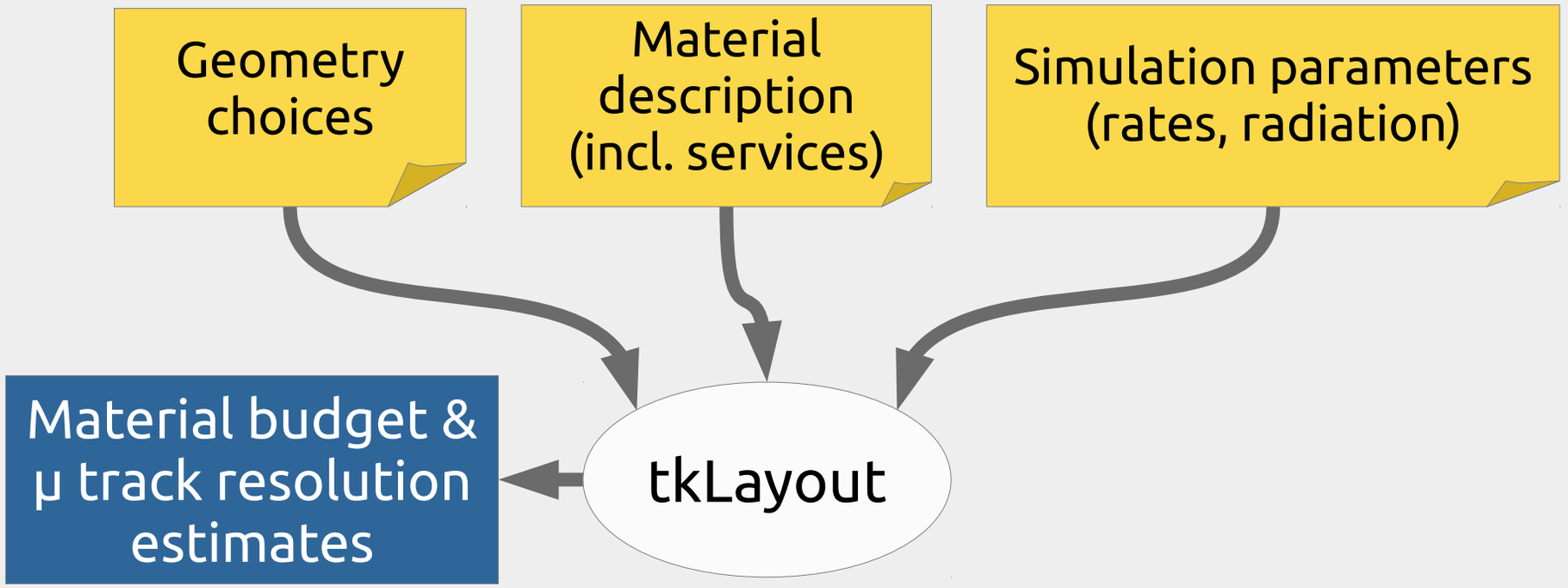
- Setup is fast
- Running is resource lean
- Interpretation straightforward
- Does not depend on specific algorithms
- Some basic performance parameters

Fast feedback cycle!

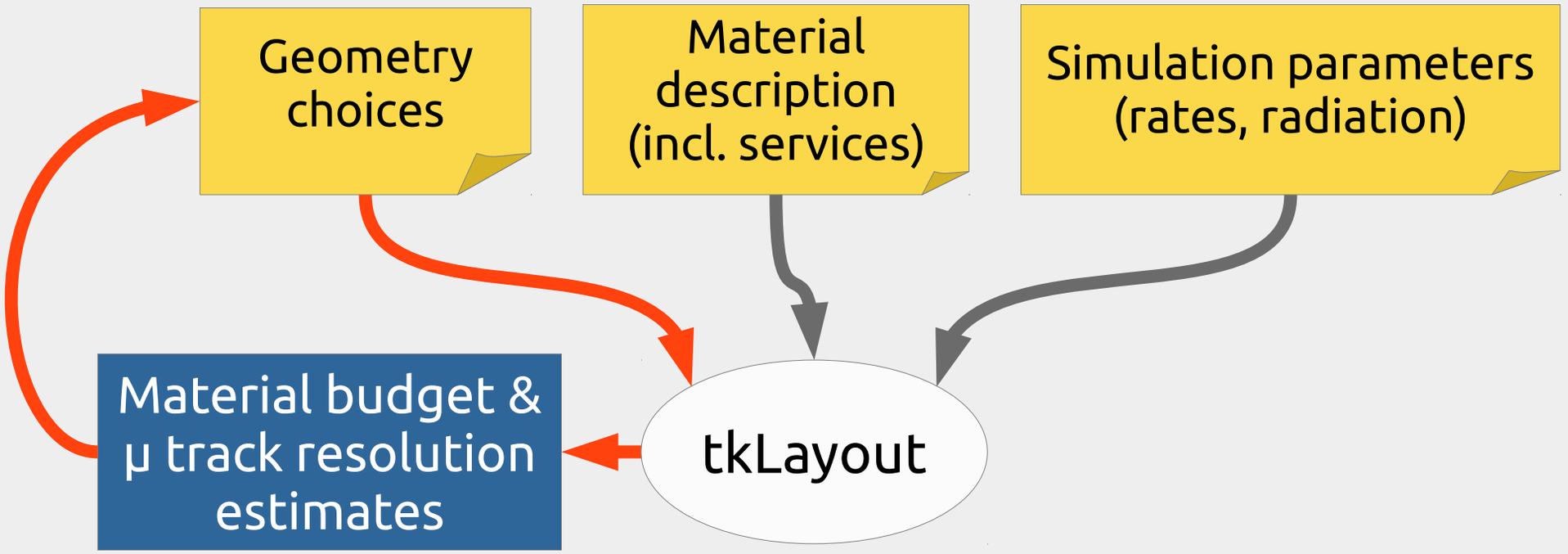
Design optimization work-flow



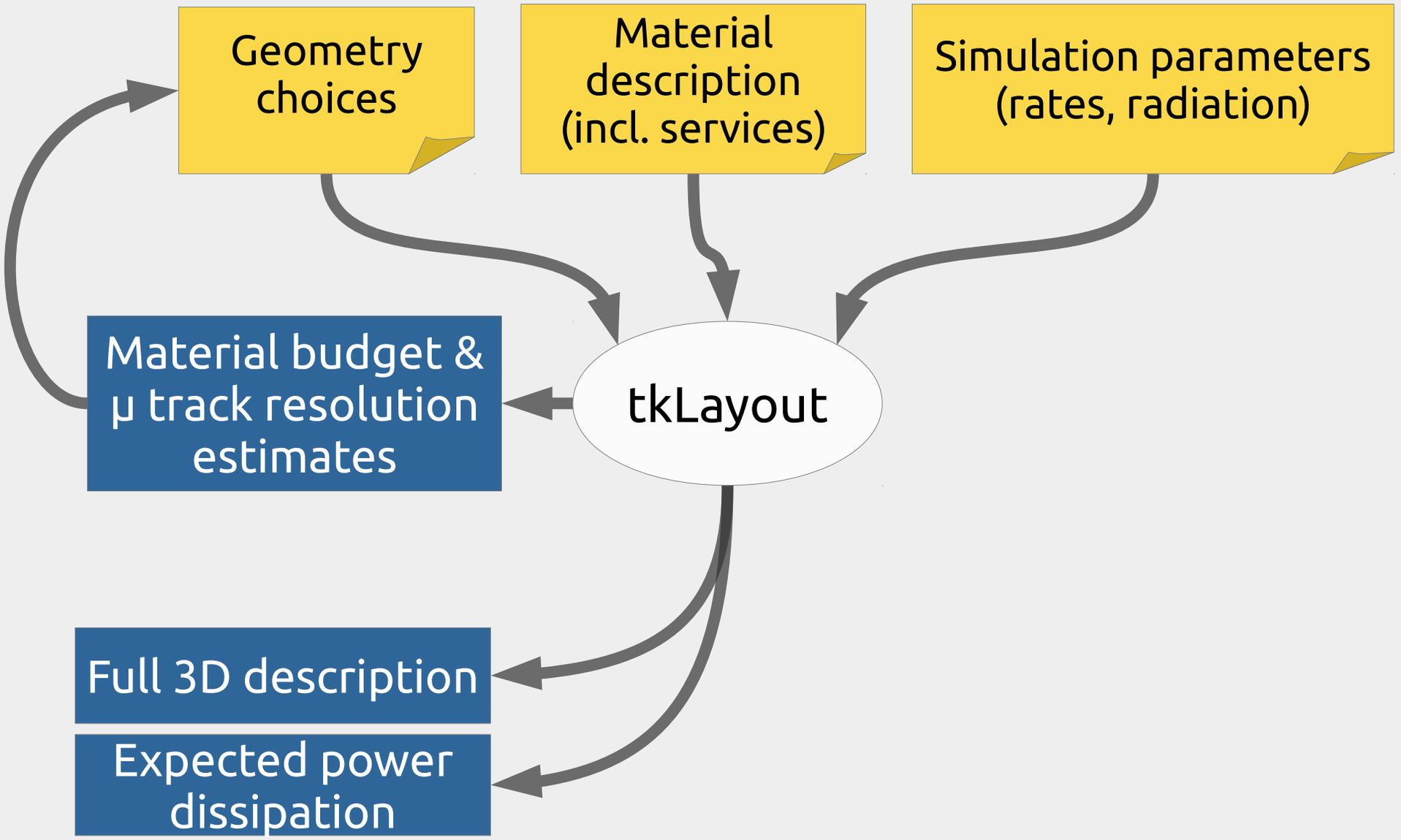
Design optimization work-flow



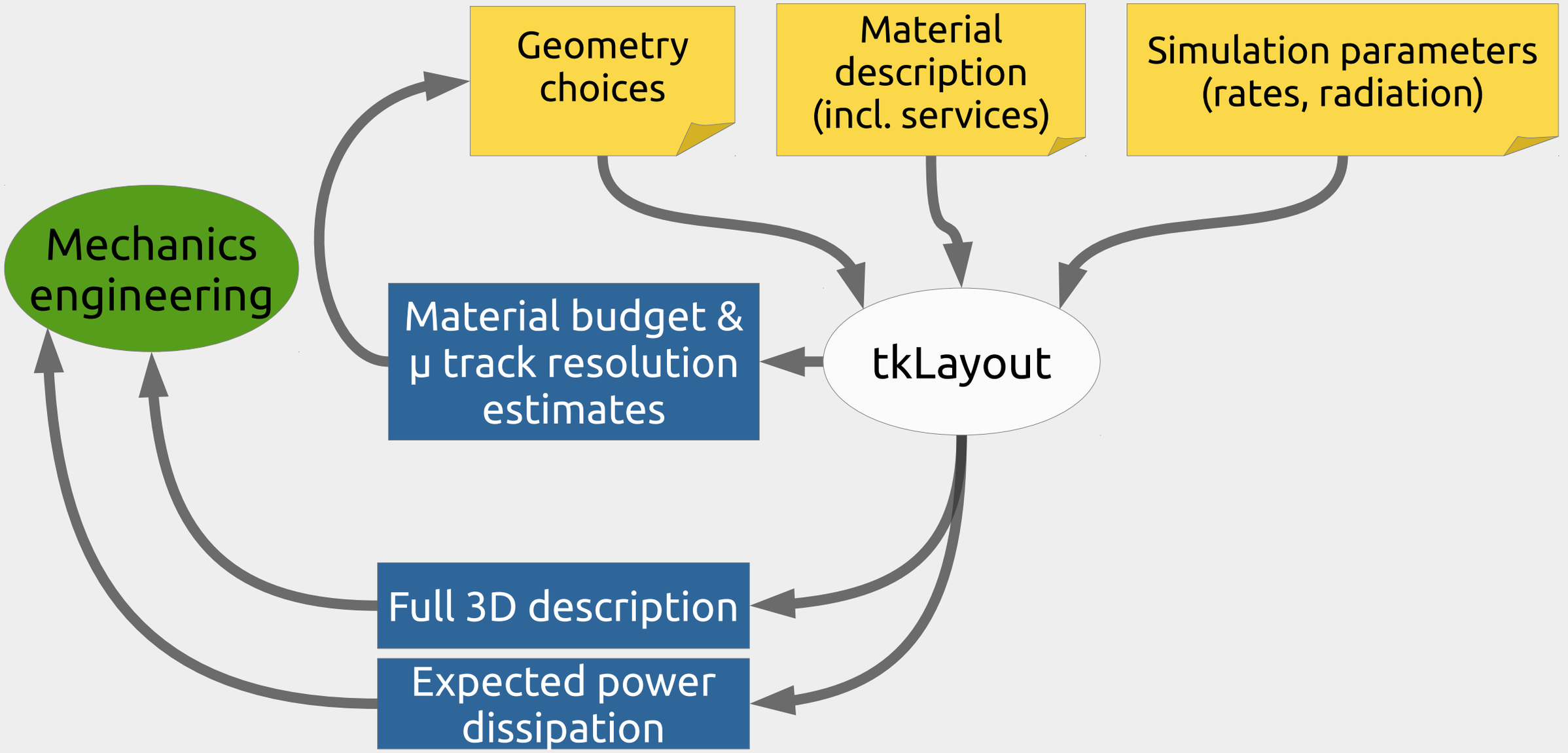
Design optimization work-flow



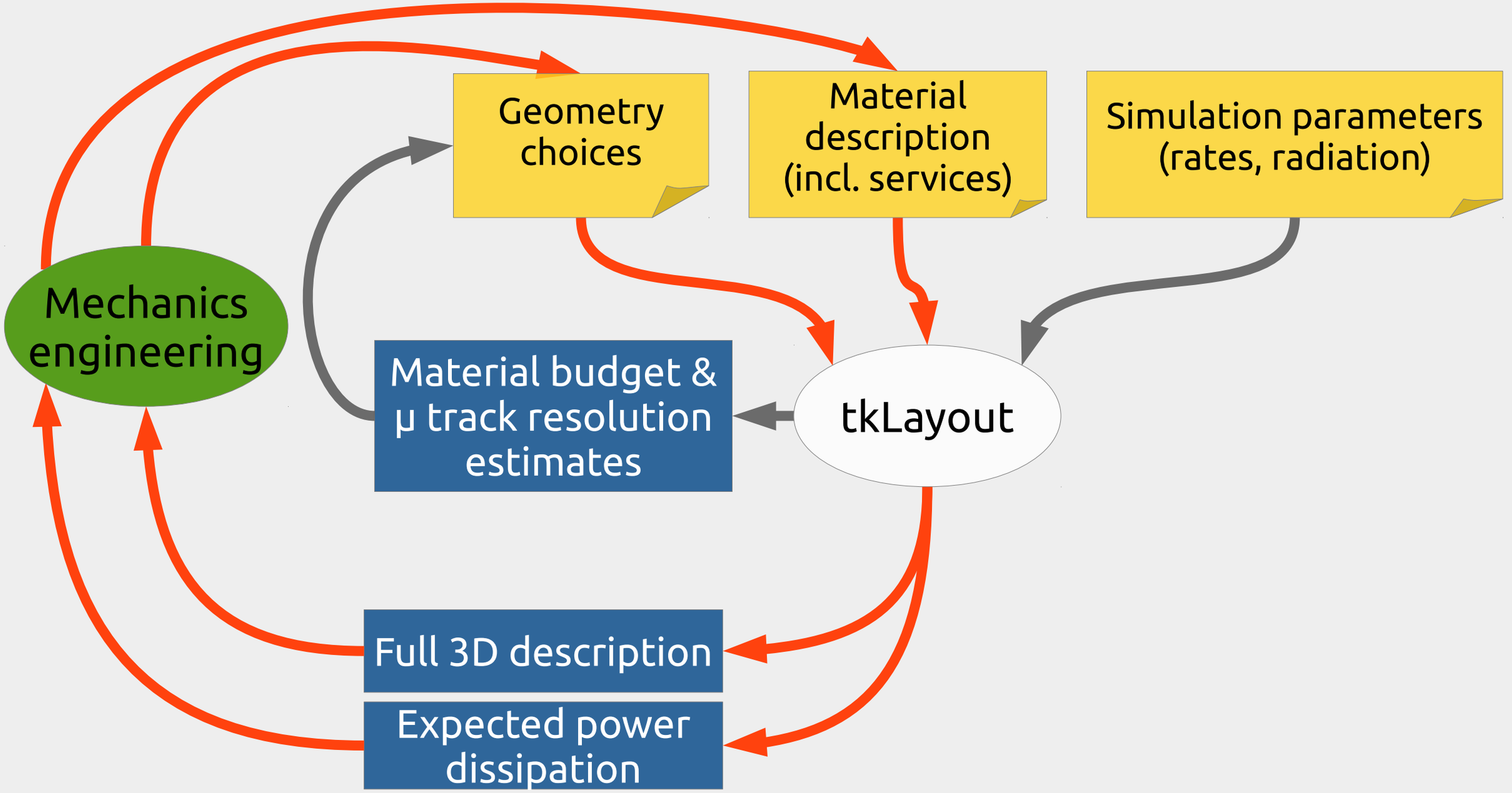
Design optimization work-flow



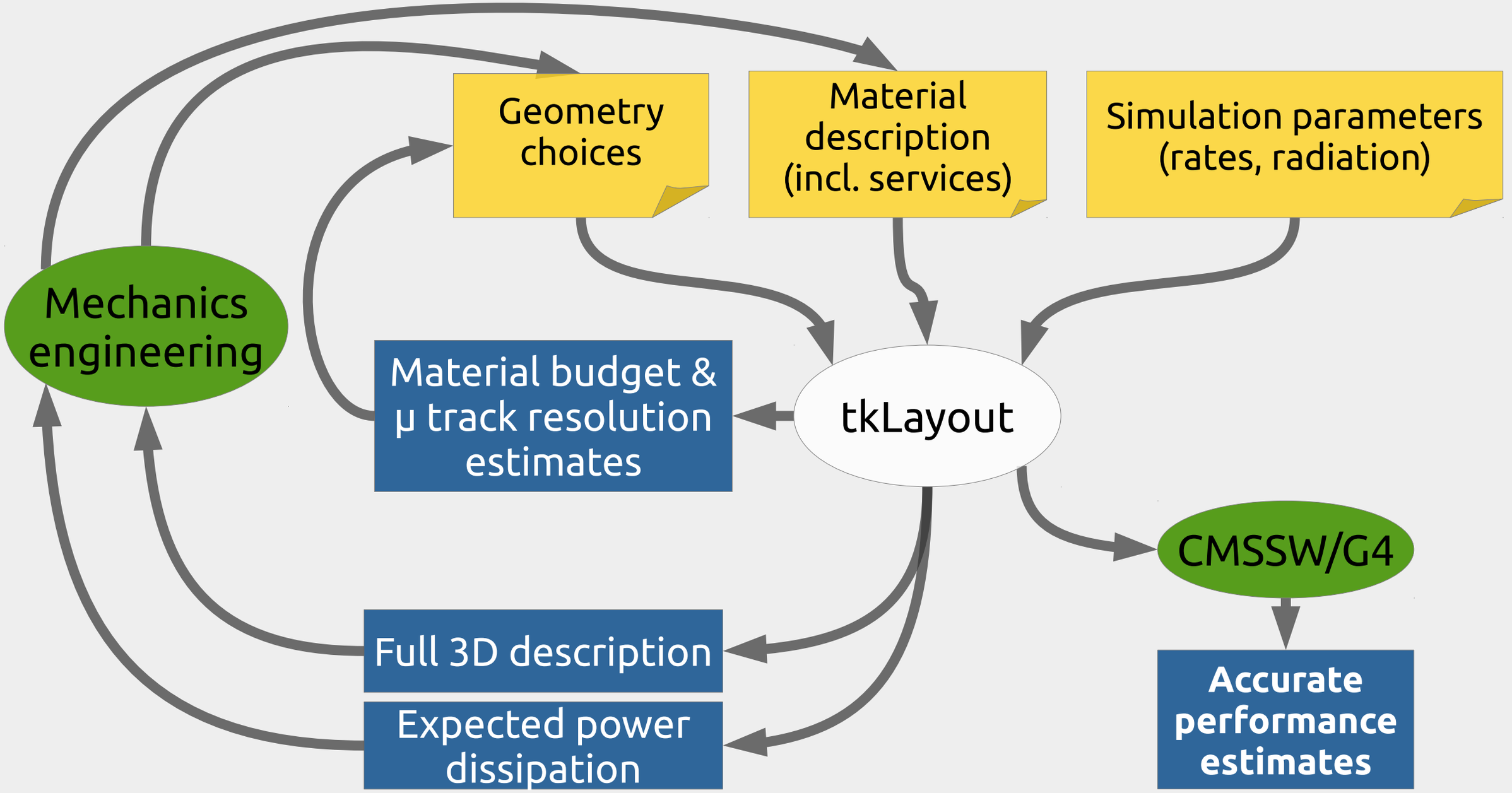
Design optimization work-flow



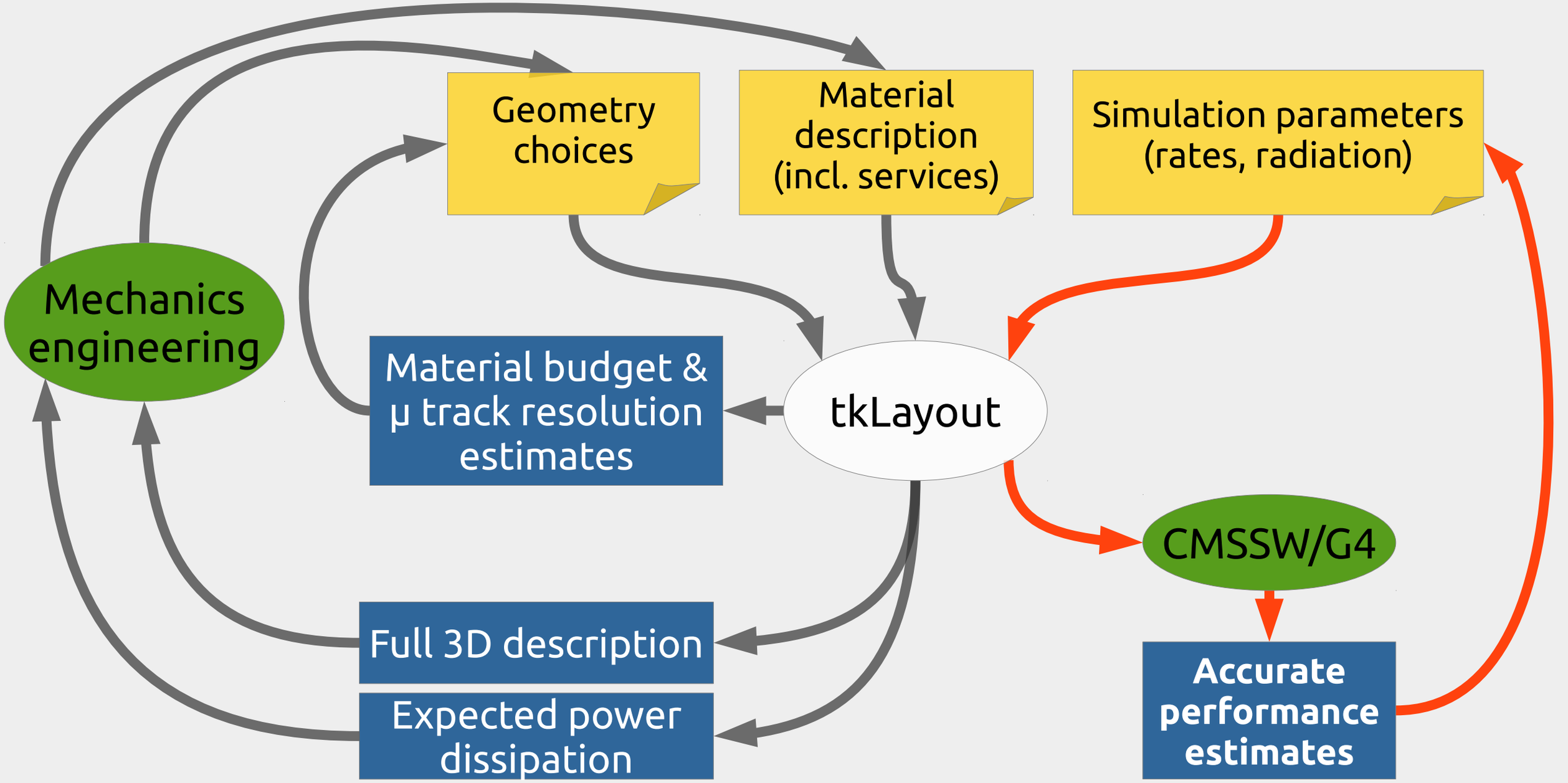
Design optimization work-flow



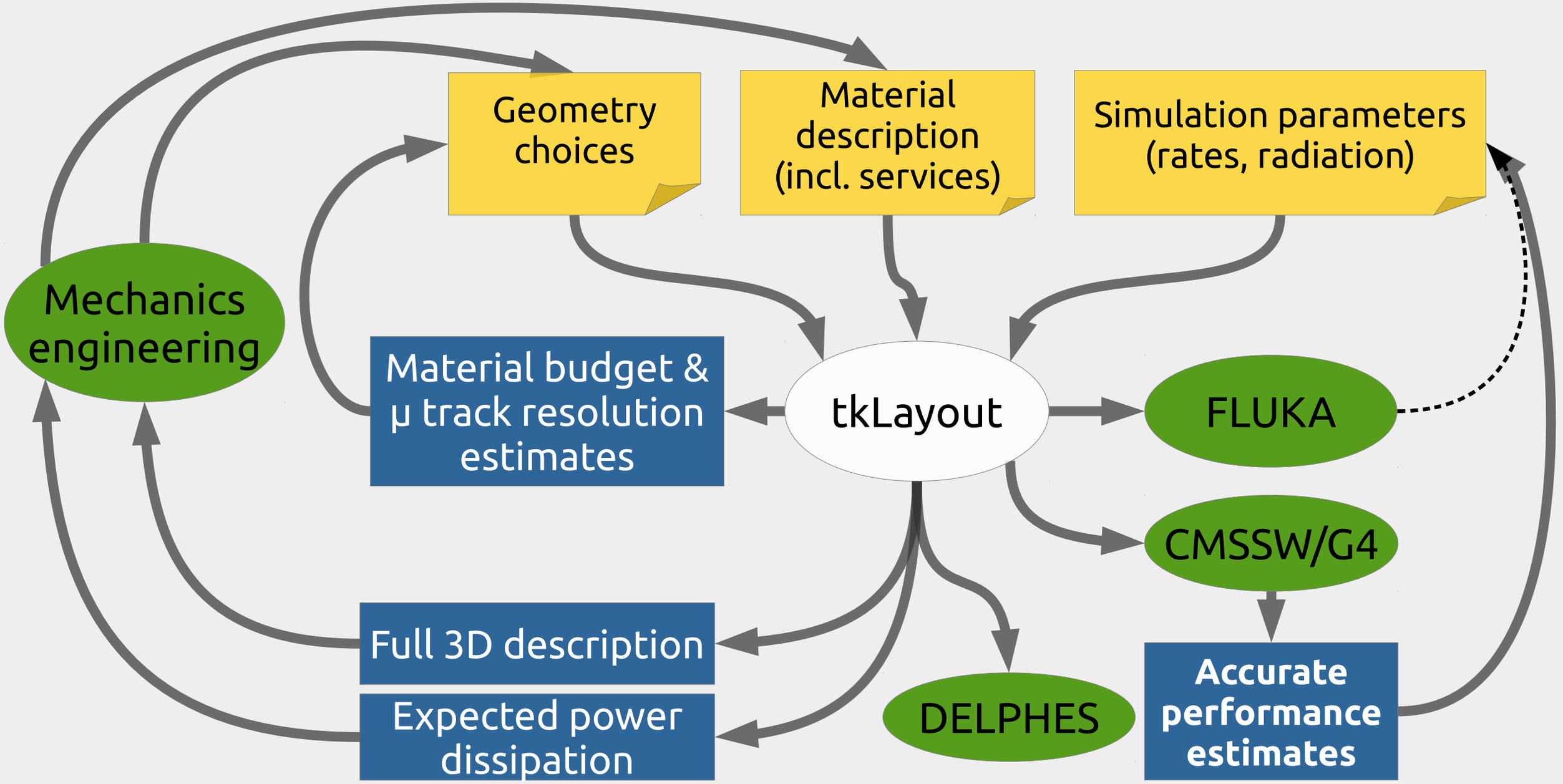
Design optimization work-flow



Design optimization work-flow



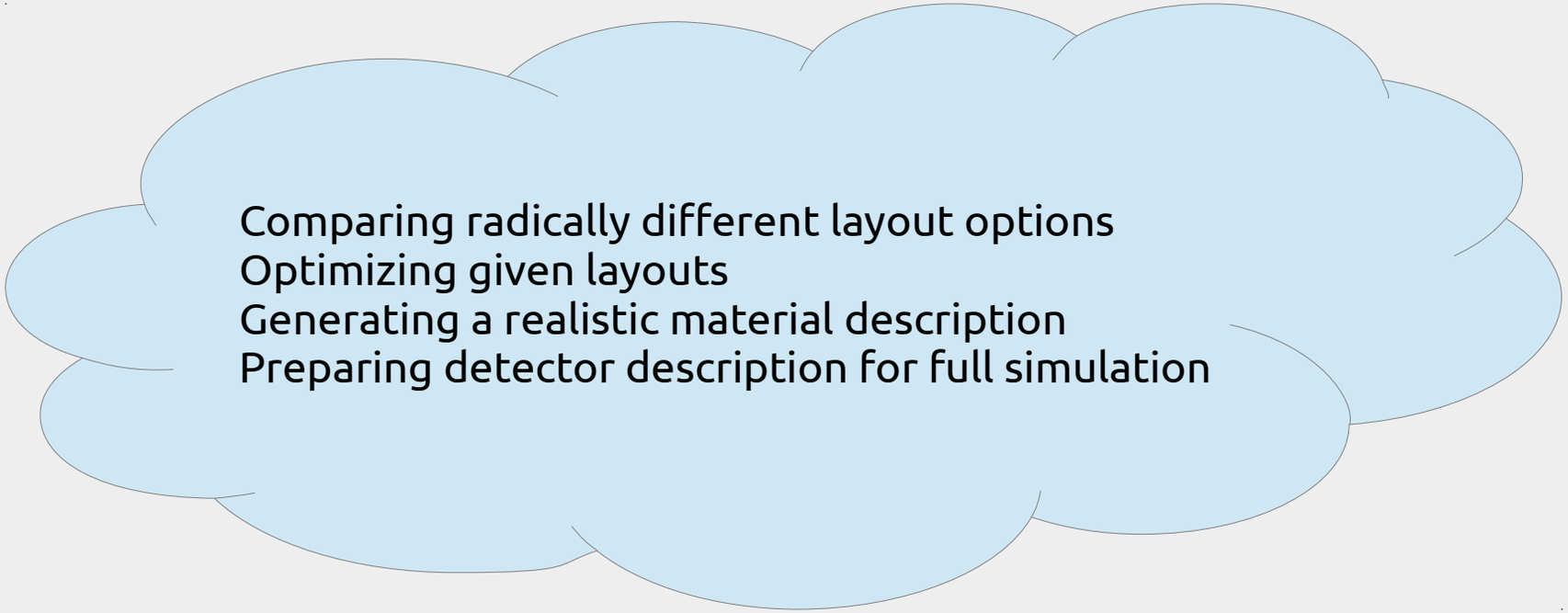
Design optimization work-flow



Results

Key tool for the design of two large detectors
(different level of development stage):

- CMS Tracker for HL-LHC
- Tracker for FCC-hh proposal

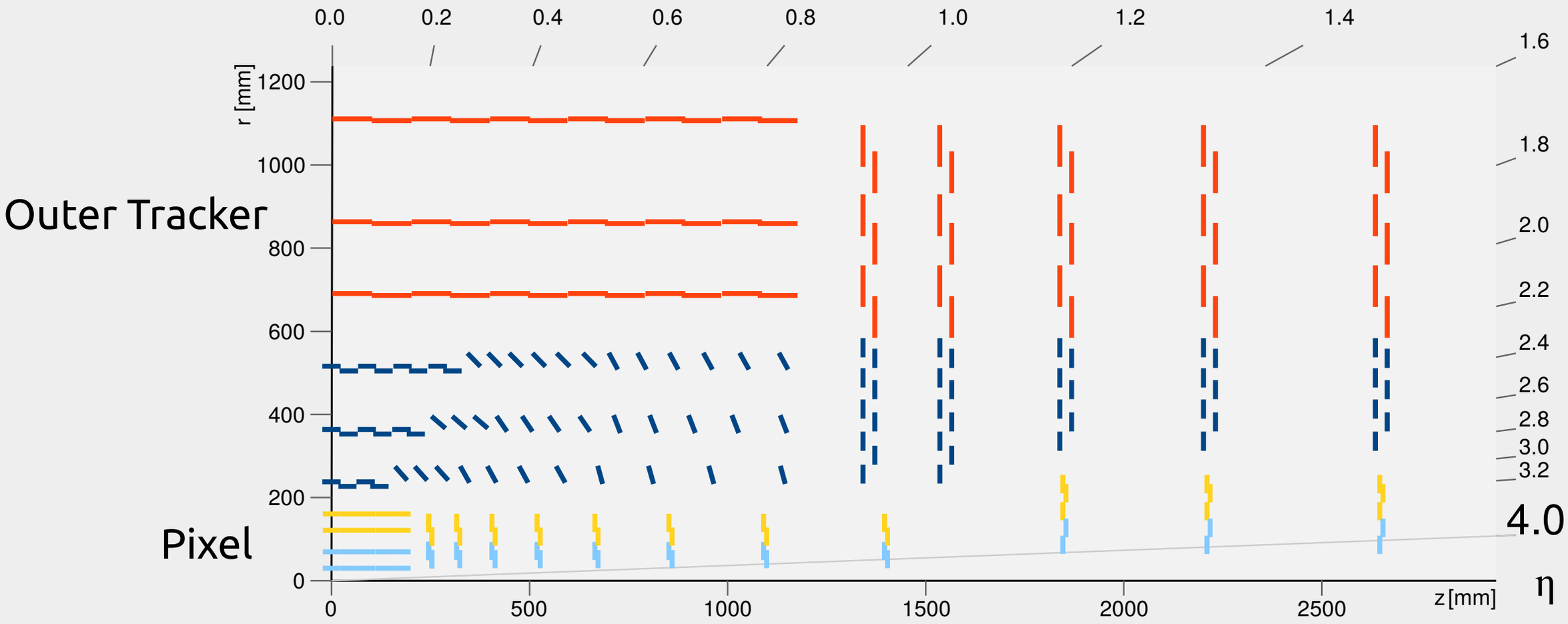


Software development

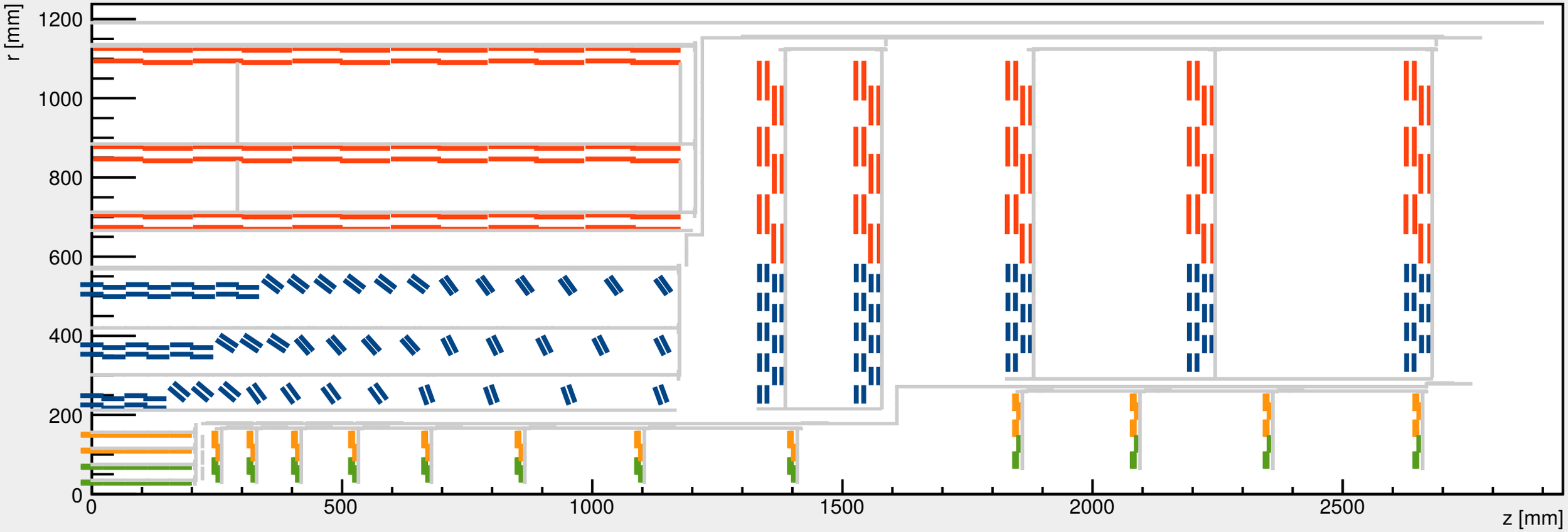
- **Master branch**
  - Development for CMS
  - Generic “principle”, but many detector-specific parts of the code
  - Lots of features large development effort
- **Lite branch**
  - Port for generic development → FCC-hh
  - Quite generic in implementation
  - Good code structure & quality
  - Not all functionalities plugged in
- **Huge amount of integrated work & potential: it should be saved for the future. Proposed deliverables:**
  - **Create a completely detector-agnostic port**
  - **Improve the code structure → more maintainable / extendible**
  - **Add documentation**
  - (possibly) new running environment? Web-based?
  - (possibly) extend to different tracking detectors (e.g. smaller, but more common: telescopes)
  - (possibly) study & implement new figures of merit

Back-up:
just some random output :-)

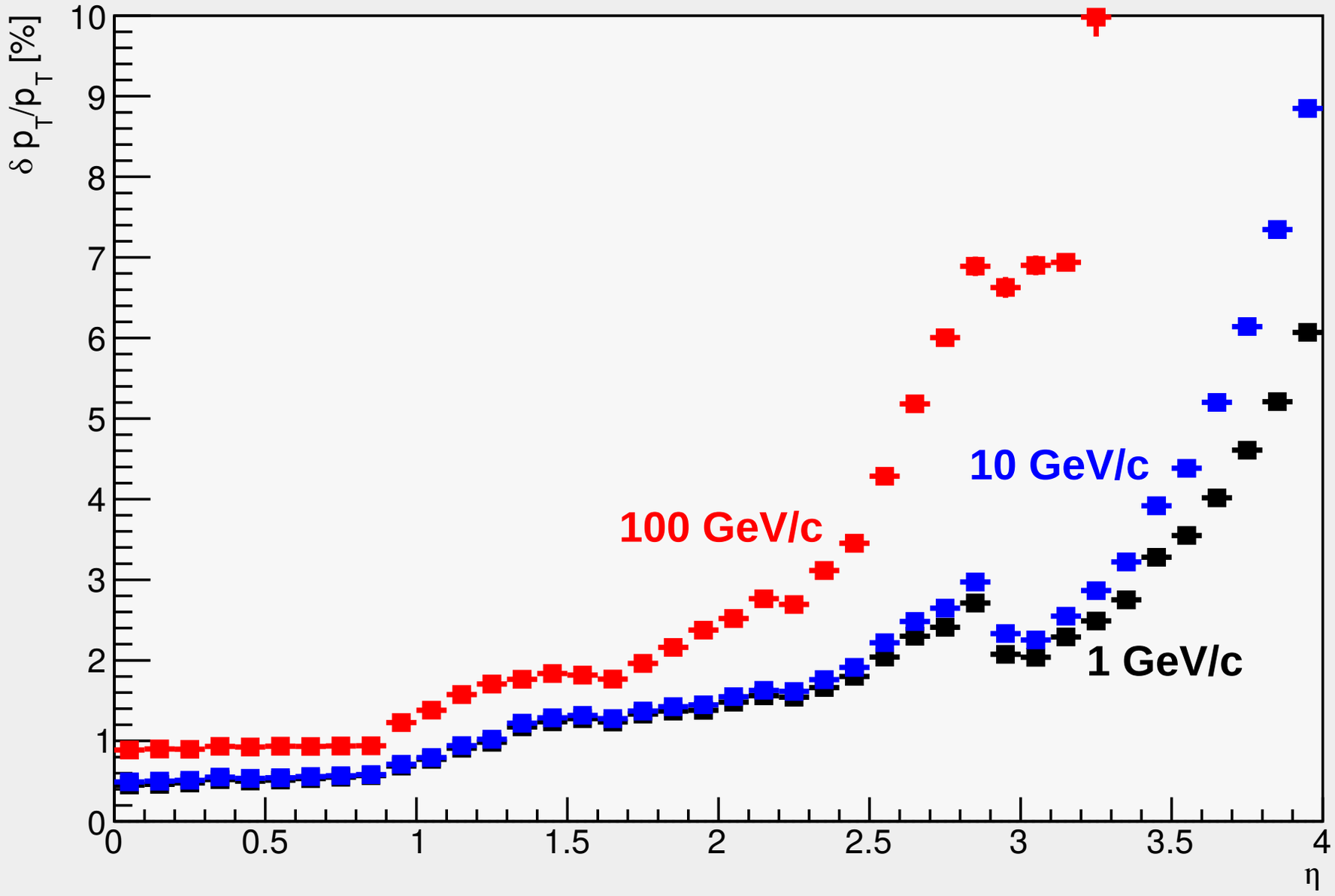
CMS HL-LHC layout



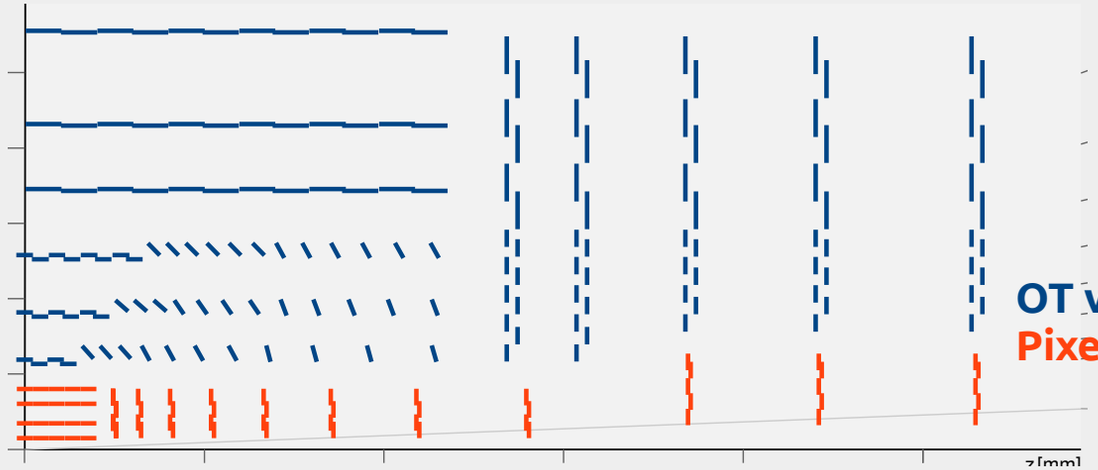
Service distribution

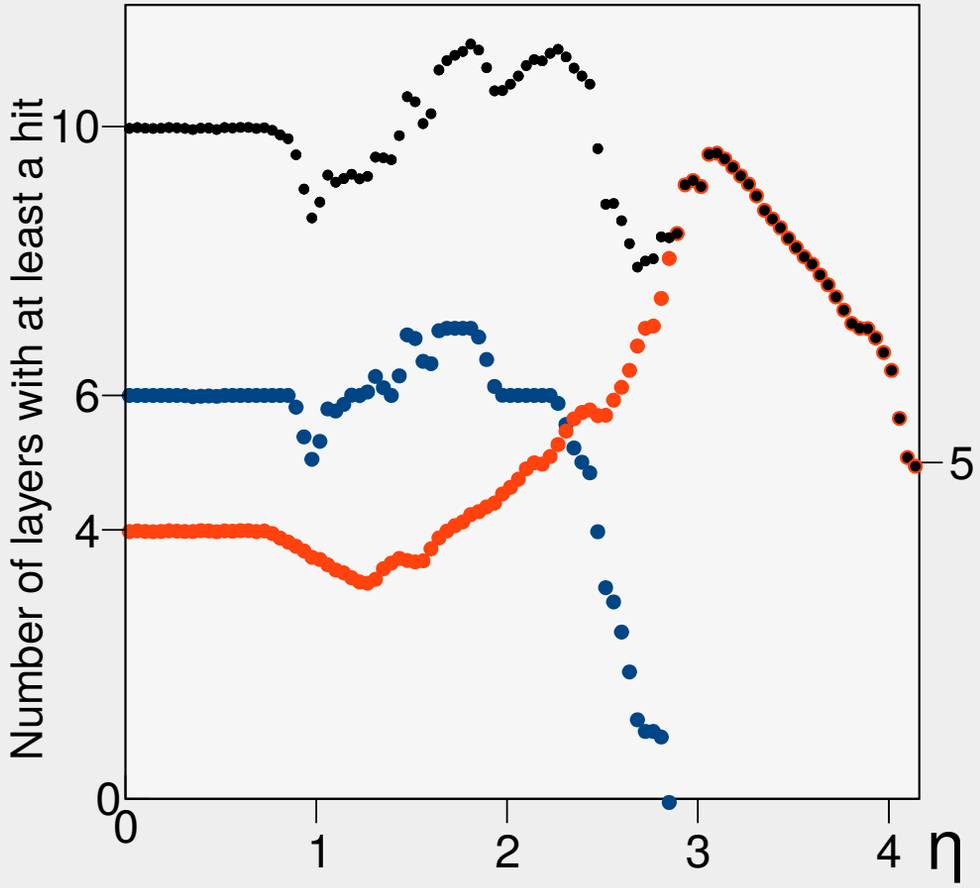


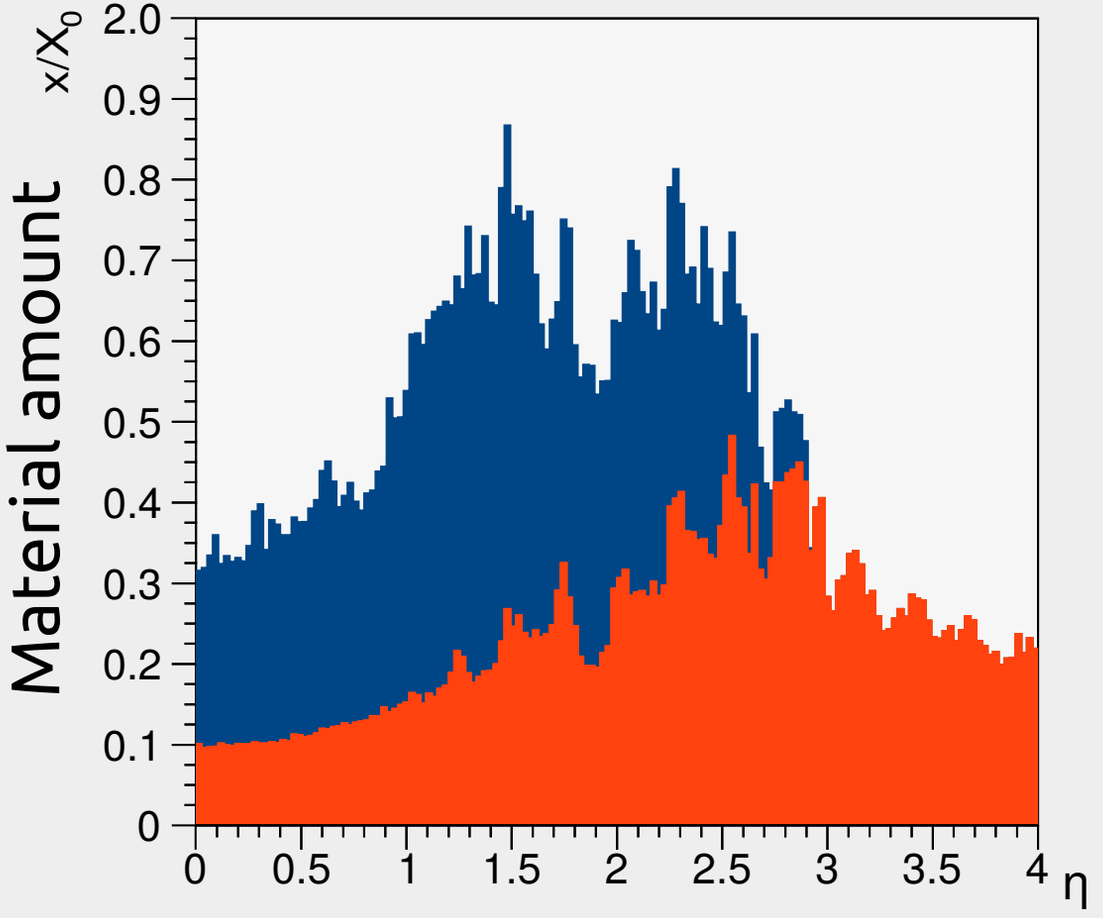
pT resolution



Material budget & layer coverage







Layer hermeticity

Hermetic coverage limited for **barrel-only geometry**