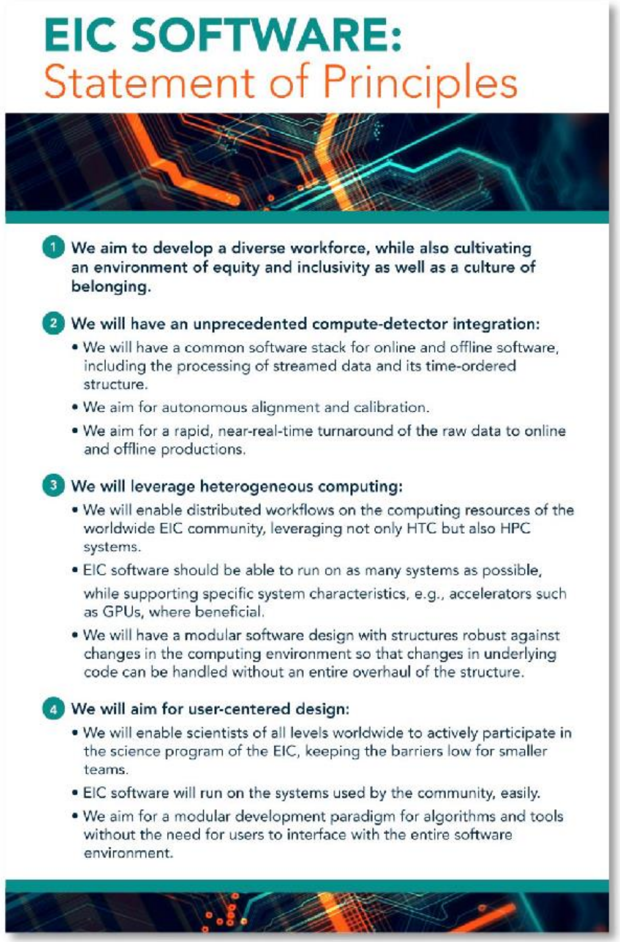
Introduction to ePIC Software:
Getting Started with Detector and Physics Studies

Tutorials will follow: https://github.com/hszumila/cfns-ss

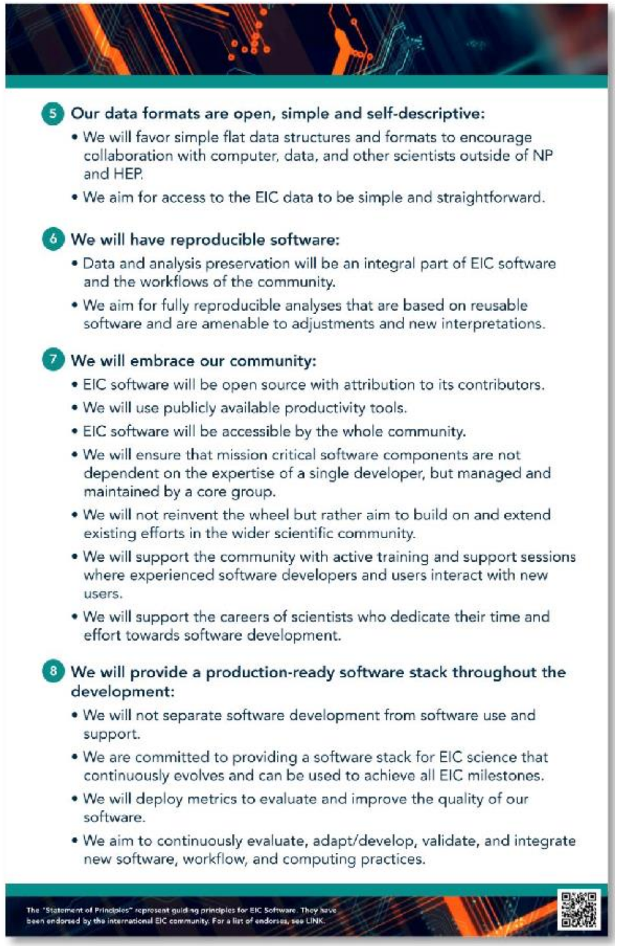
Holly Szumila-Vance
Florida International University
CFNS EIC Summer School
June 2026

“The right tools should be easy to find and easy to adopt”



EIC SOFTWARE: Statement of Principles

1. **We aim to develop a diverse workforce, while also cultivating an environment of equity and inclusivity as well as a culture of belonging.**
2. **We will have an unprecedented compute-detector integration:**
   - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
   - We aim for autonomous alignment and calibration.
   - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.
3. **We will leverage heterogeneous computing:**
   - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
   - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
   - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.
4. **We will aim for user-centered design:**
   - We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
   - EIC software will run on the systems used by the community, easily.
   - We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.



5. **Our data formats are open, simple and self-descriptive:**
   - We will favor simple flat data structures and formats to encourage collaboration with computer, data, and other scientists outside of NP and HEP.
   - We aim for access to the EIC data to be simple and straightforward.
6. **We will have reproducible software:**
   - Data and analysis preservation will be an integral part of EIC software and the workflows of the community.
   - We aim for fully reproducible analyses that are based on reusable software and are amenable to adjustments and new interpretations.
7. **We will embrace our community:**
   - EIC software will be open source with attribution to its contributors.
   - We will use publicly available productivity tools.
   - EIC software will be accessible by the whole community.
   - We will ensure that mission critical software components are not dependent on the expertise of a single developer, but managed and maintained by a core group.
   - We will not reinvent the wheel but rather aim to build on and extend existing efforts in the wider scientific community.
   - We will support the community with active training and support sessions where experienced software developers and users interact with new users.
   - We will support the careers of scientists who dedicate their time and effort towards software development.
8. **We will provide a production-ready software stack throughout the development:**
   - We will not separate software development from software use and support.
   - We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
   - We will deploy metrics to evaluate and improve the quality of our software.
   - We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

The "Statement of Principles" represent guiding principles for EIC Software. They have been endorsed by the international EIC community. For a list of endorses, see LINK.

EIC Software is:

1. Diverse
2. Integrative
3. Heterogeneous
4. User-centered
5. Accessible
6. Reproducible
7. Collaborative
8. Agile

https://eic.github.io/activities/principles.html

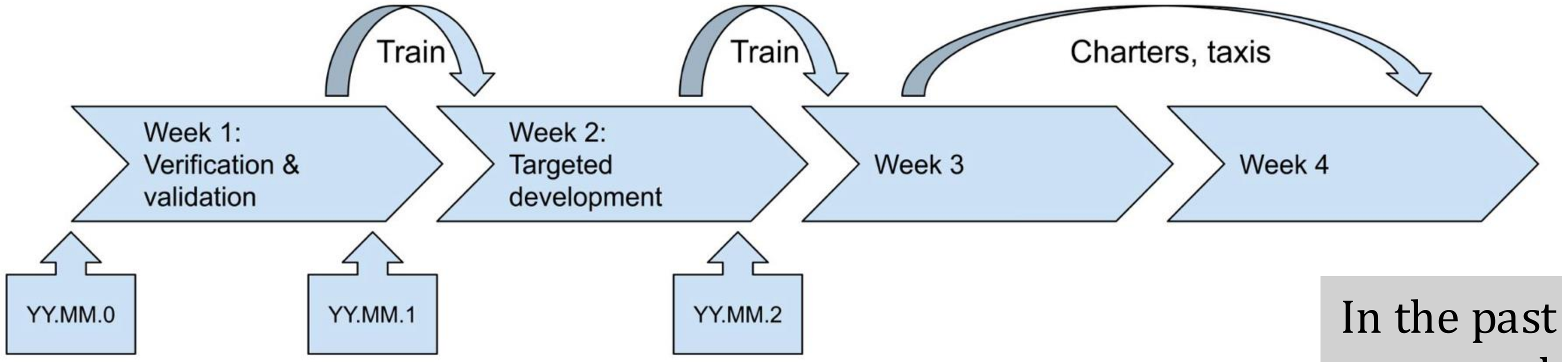
ePIC strategy: Production-ready software

Objectives:

1. Achieve continuous deployment of the software used for detector and physics simulations

2. Ensure regular updates of simulation productions for detector and physics studies, and for geometry and algorithm development

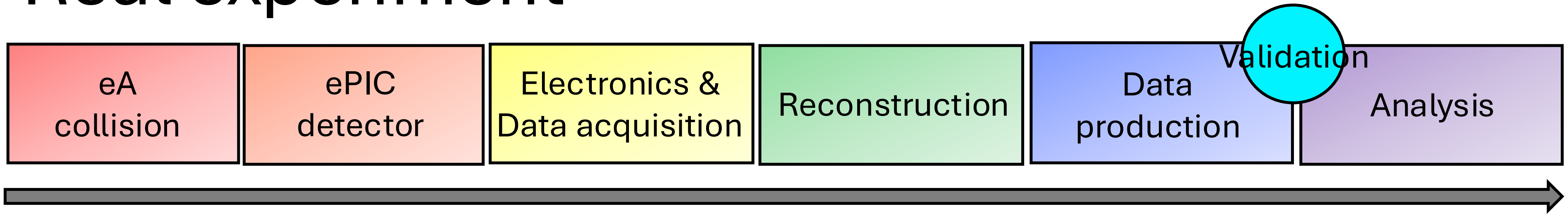
3. Implement timely validation and quality control for simulation productions on datasets that require substantial time and resources



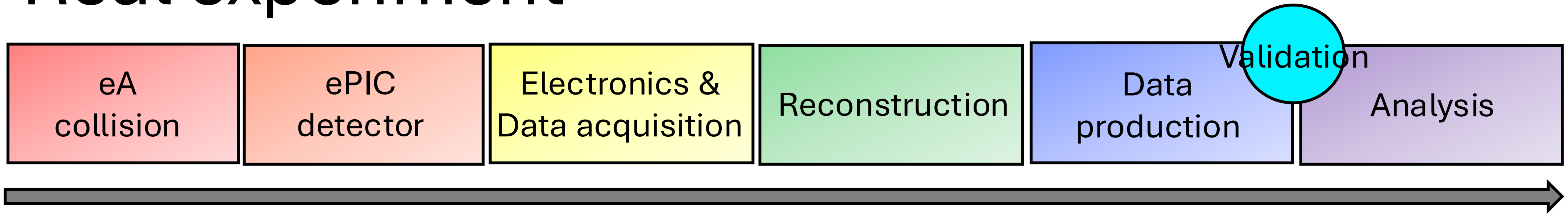
Tagged software releases

In the past year, monthly simulation campaigns consumed approximately 15 million core hours on the OSF, generating over 500 TB of simulation data.

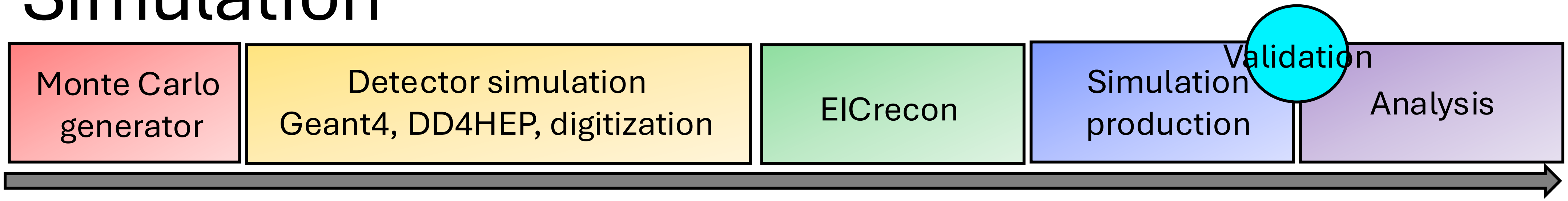
Real experiment



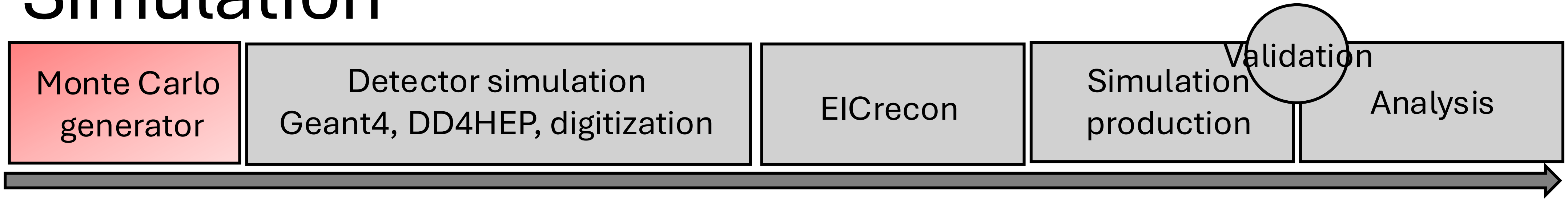
Real experiment



Simulation

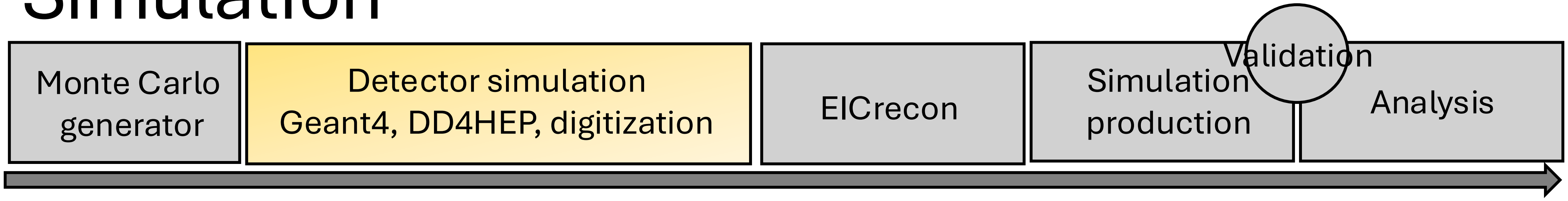


Simulation



Input events from MC event generators or particle guns, with optional physics background merging

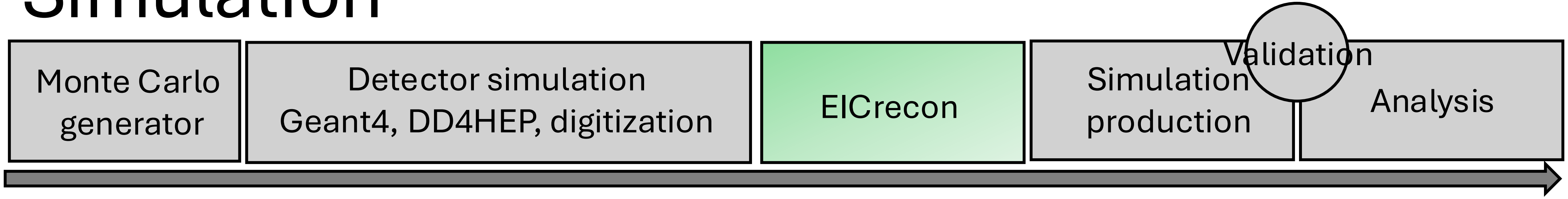
Simulation



GEANT4 simulations driven by DD4hep, output data in the EIC Data Model (EDM4hep + EDM4eic, described in Podio)

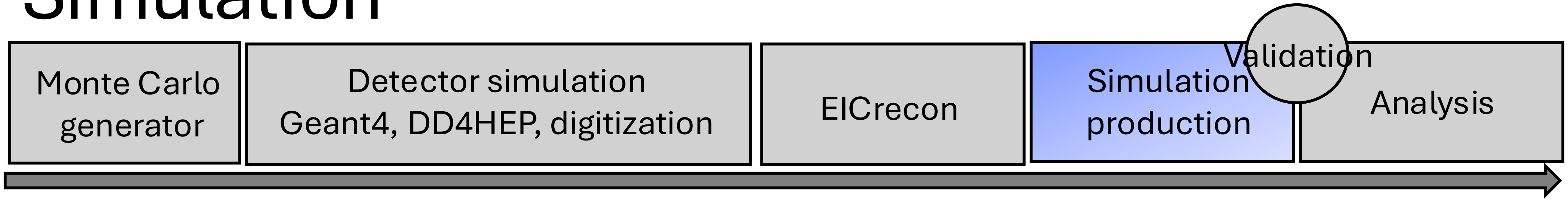
Algorithms to transform the GEANT4 hits to mimic real detector readout, including background stacking, “pileup”, DAQ frames

Simulation



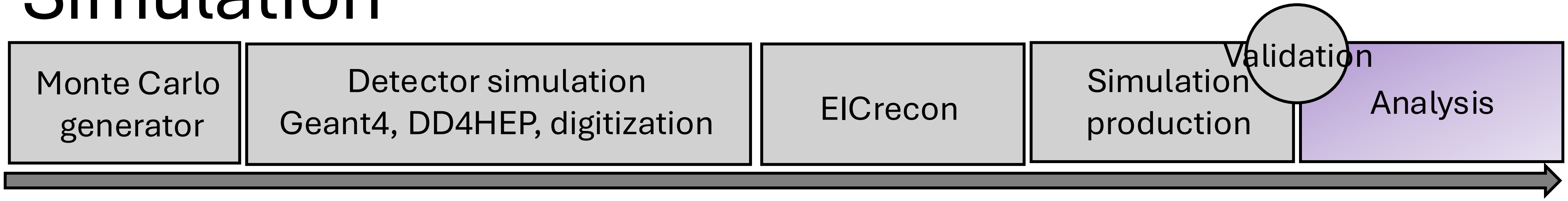
Realistic reconstruction algorithms starting from raw detector output (digitization to look like real data)

Simulation



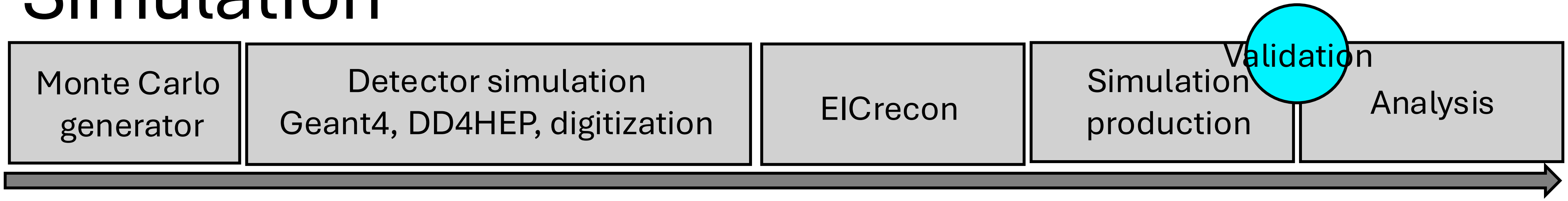
Processing of the data using calibration information, corrections, etc

Simulation



User analyses in plain C++/ROOT or Python/uproot, facilitated by using a flat data model

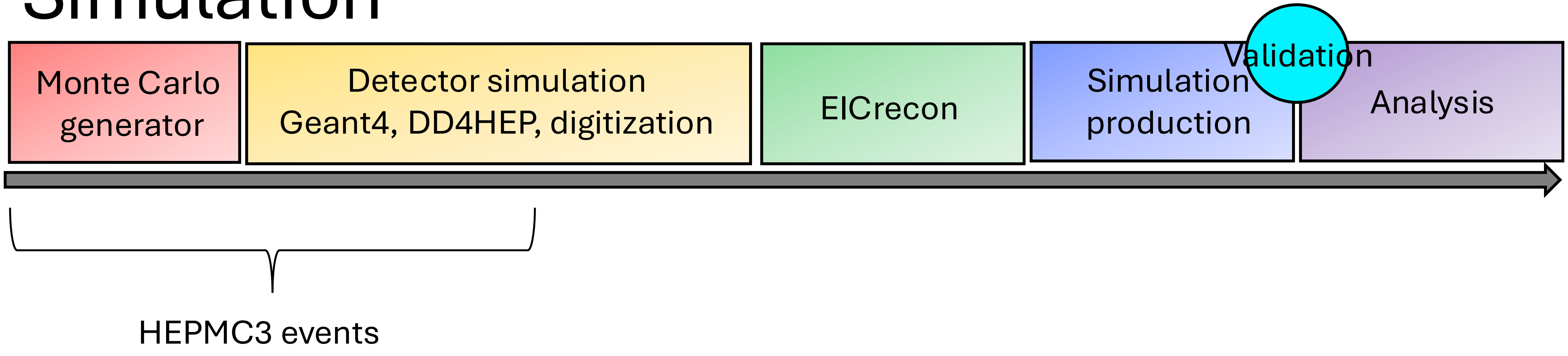
Simulation



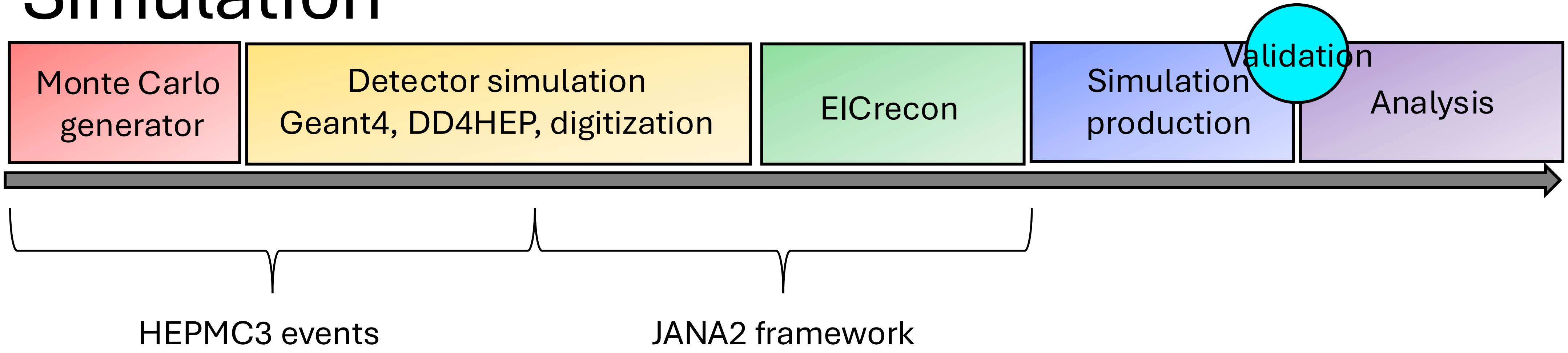
Benchmarks on both detectors and physics

https://eic.github.io/tutorial-developing-benchmarks/

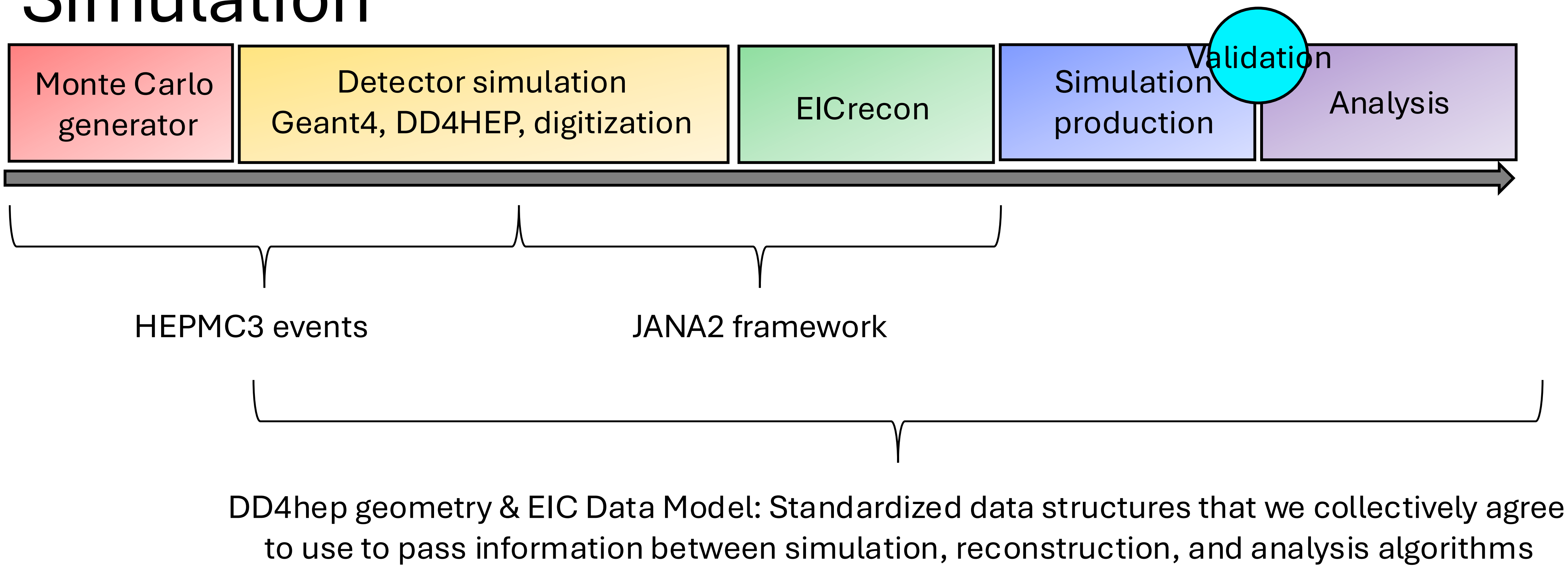
Simulation


Monte Carlo generator
Detector simulation
Geant4, DD4HEP, digitization
EICrecon
Simulation production
Validation
Analysis
HEPMC3 events

Simulation

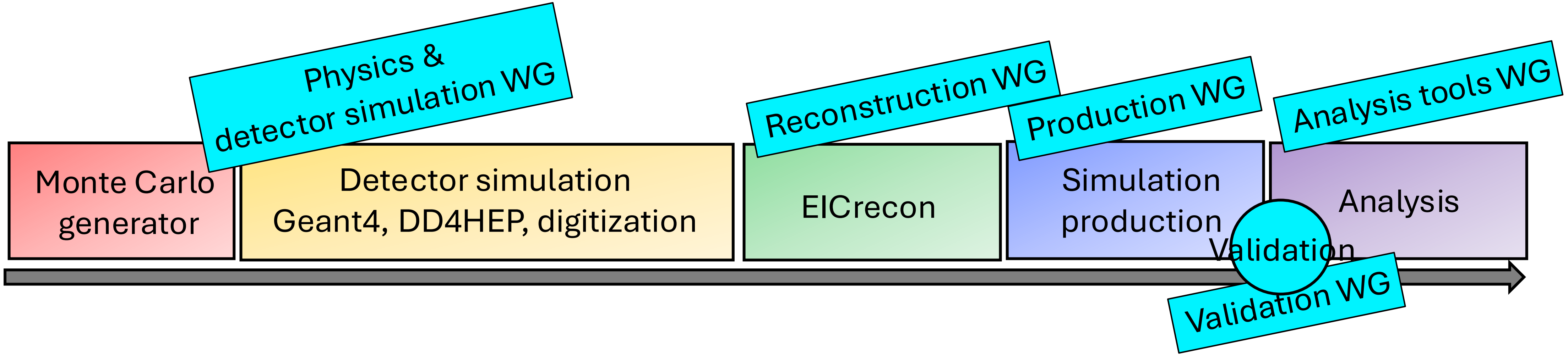

Monte Carlo generator
Detector simulation
Geant4, DD4HEP, digitization
EICrecon
Simulation production
Validation
Analysis
HEPMC3 events
JANA2 framework

Simulation



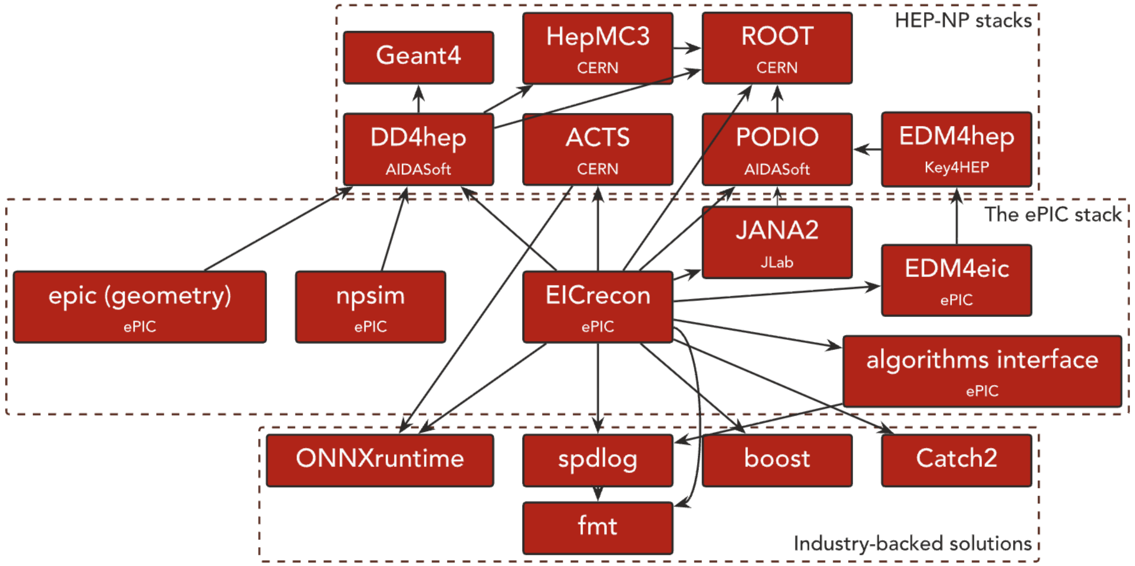
Monte Carlo generator
Detector simulation
Geant4, DD4HEP, digitization
EICrecon
Simulation production
Validation
Analysis
HEPMC3 events
JANA2 framework
DD4hep geometry & EIC Data Model: Standardized data structures that we collectively agree to use to pass information between simulation, reconstruction, and analysis algorithms

Interfacing with the Working Groups



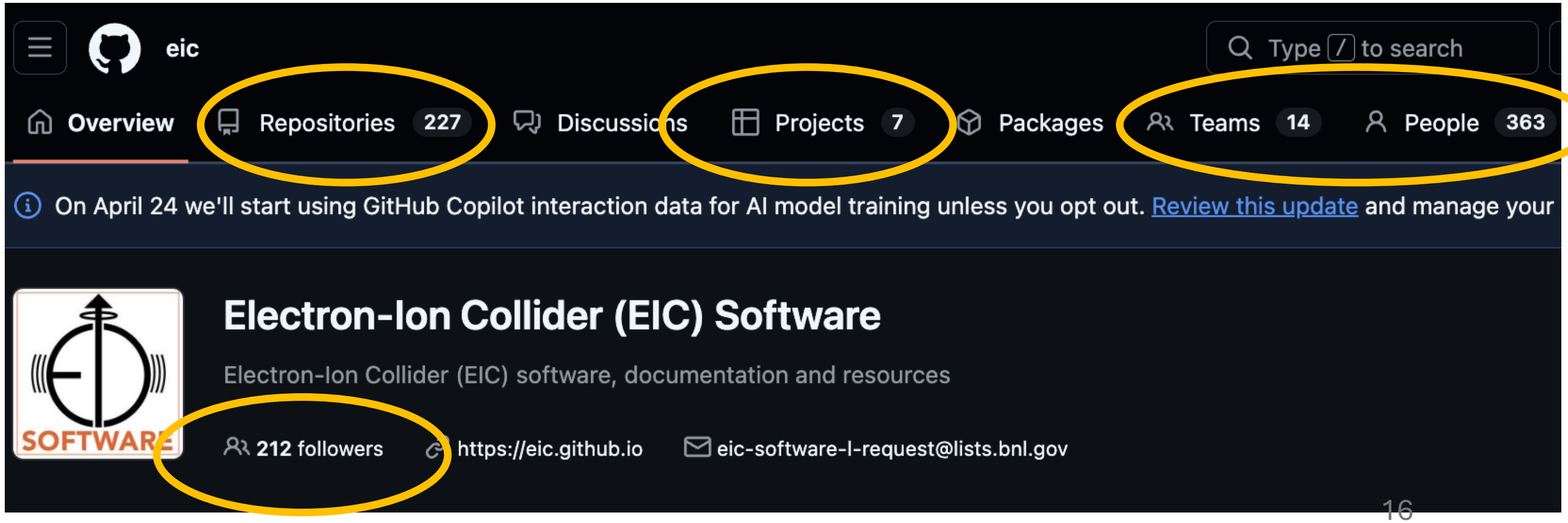
Lost? Overcoming challenges….

ePIC Software Stack



Ask questions!

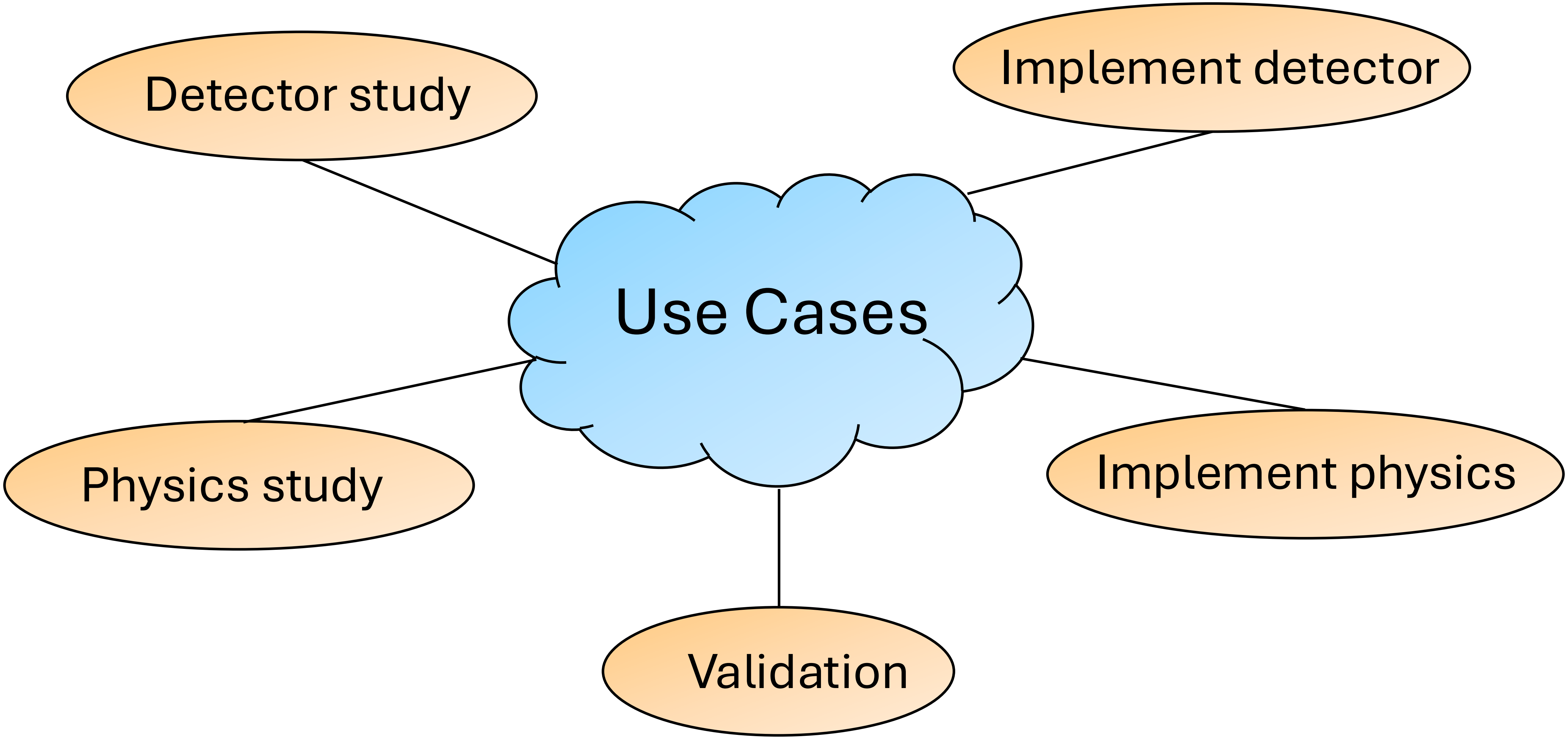
Contribute!

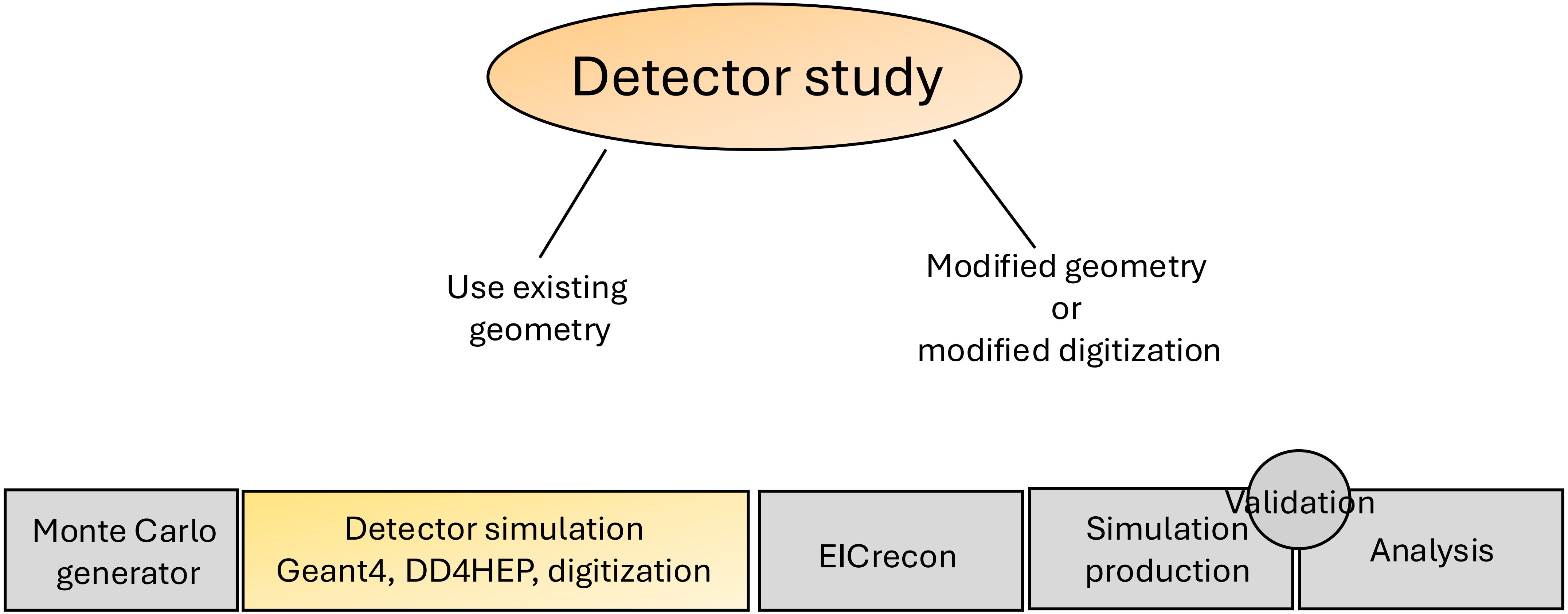


What do I want to study?

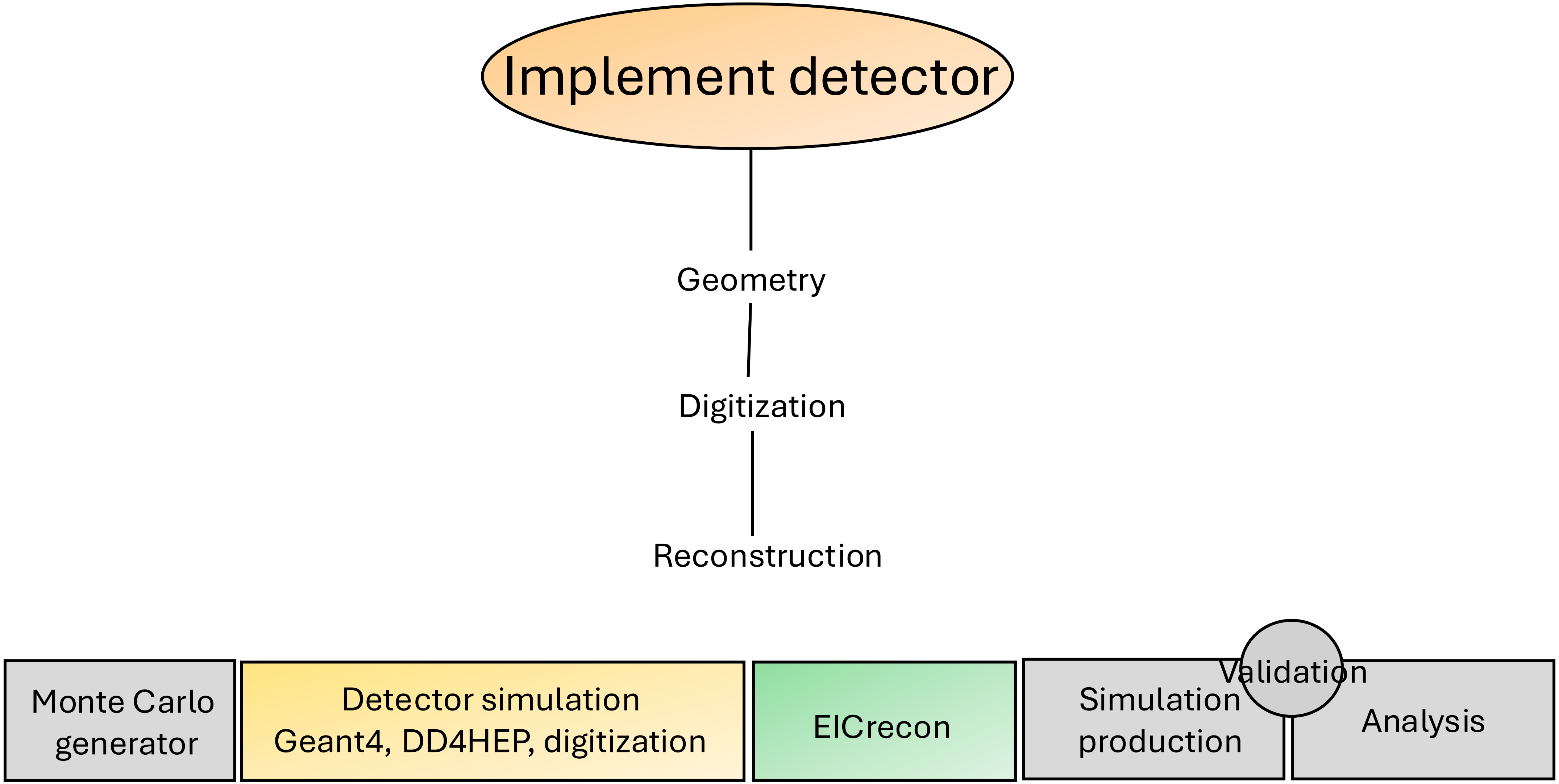
Detector:
- Resolution
- Efficiency
- Digitization
- Optimization
- Geometry
- …

Physics:
- Kinematics
- Phase space
- Acceptance
- Algorithms
- …

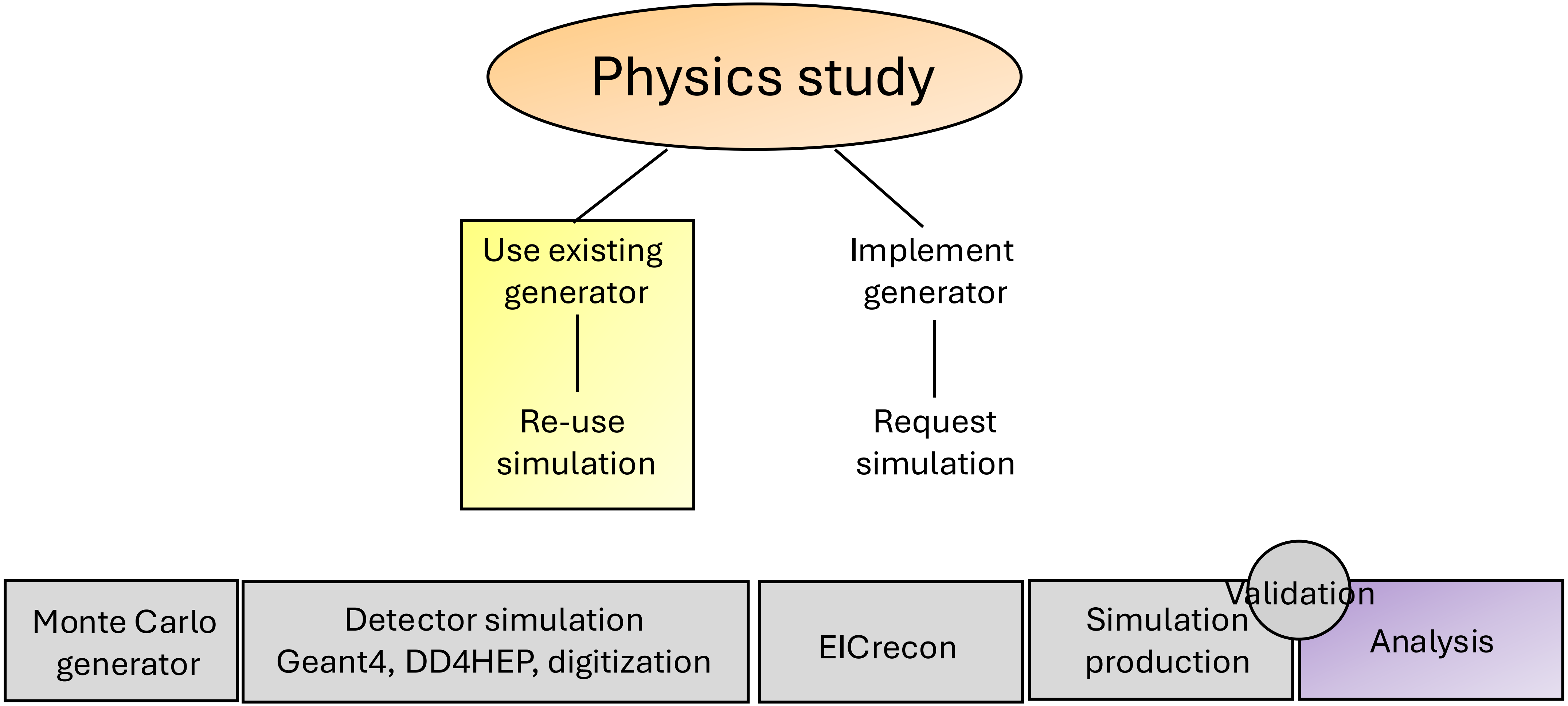

Use Cases
Detector study
Implement detector
Physics study
Implement physics
Validation



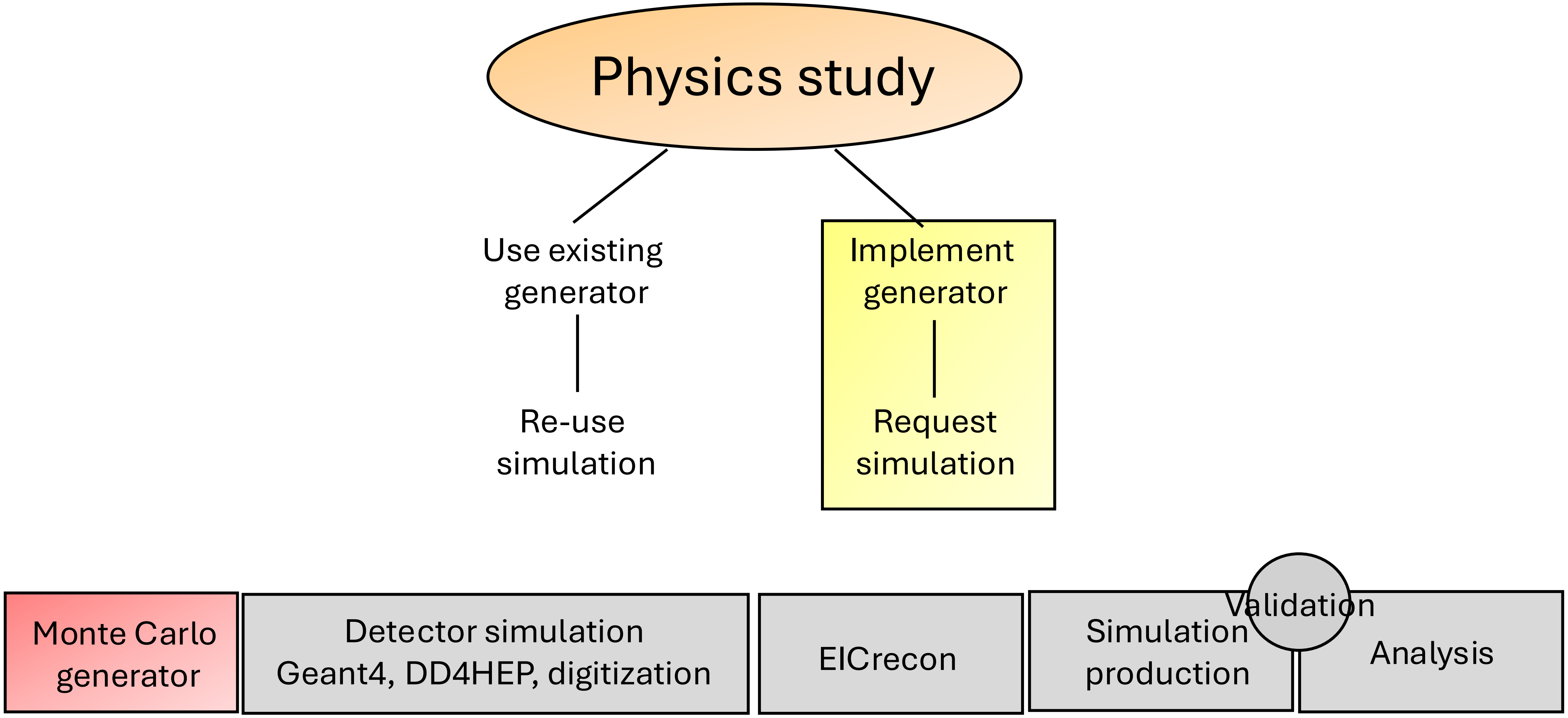
Previous tutorial: https://eic.github.io/tutorial-modifying-geometry-digitization-etc
https://eic.github.io/tutorial-geometry-development-using-dd4hep/



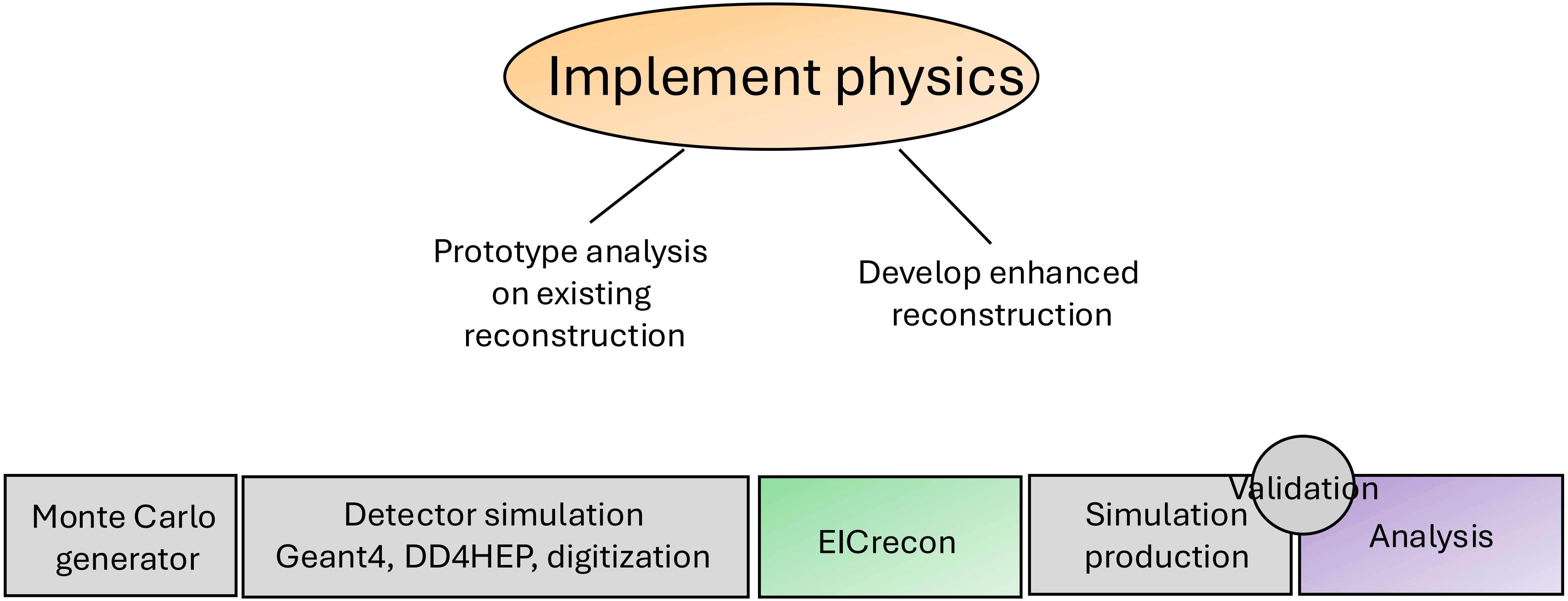
Previous tutorial: https://eic.github.io/tutorial-simulations-using-ddsim-and-geant4/
https://eic.github.io/tutorial-modifying-geometry-digitization-etc
https://eic.github.io/tutorial-geometry-development-using-dd4hep/



Previous tutorial: https://eic.github.io/tutorial-analysis
https://github.com/eic/python-analysis-bootcamp



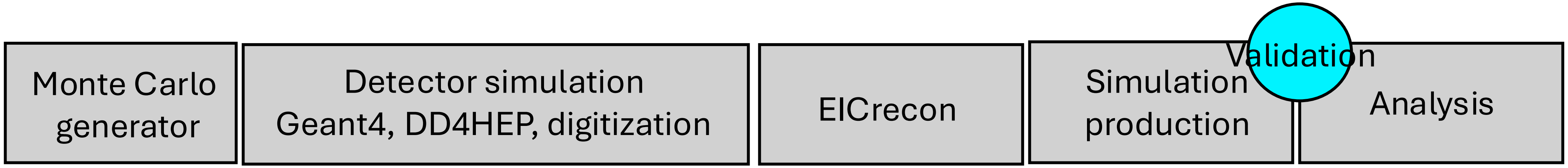
Previous tutorial: https://eic.github.io/tutorial-simulations-using-ddsim-and-geant4/



Previous tutorial: https://github.com/eic/python-analysis-bootcamp
https://eic.github.io/tutorial-analysis
https://eic.github.io/tutorial-reconstruction-algorithms

Validation

Compare the output of the physics and detector studies



Previous tutorial: https://eic.github.io/tutorial-developing-benchmarks

Now, we begin!

We will be used GitHub codespaces, so you will need a GitHub account.

Start here: https://github.com/hszumila/cfns-ss

Lots of help and resources at eic.github.io