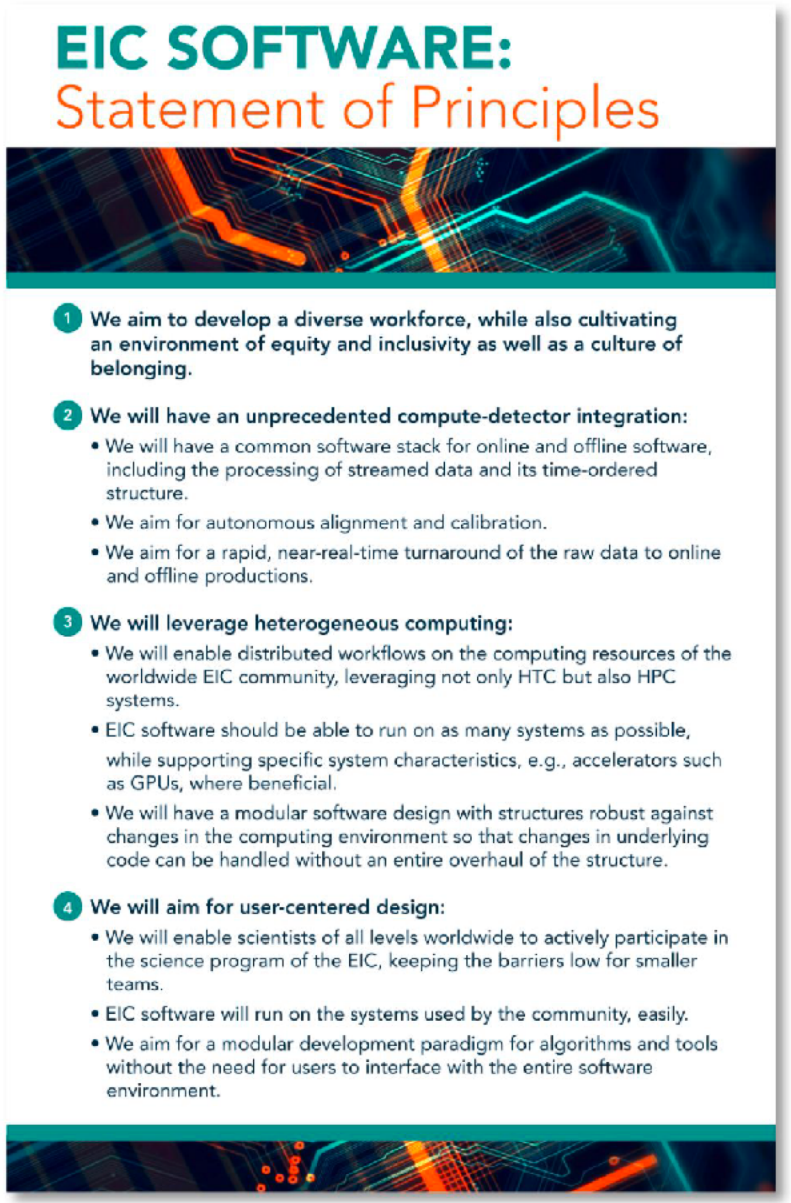
Introduction to ePIC Software: Getting Started with Detector and Physics Studies

Holly Szumila-Vance
ePIC User Learning
ePIC Software & Computing Workshop
CERN, April 2024



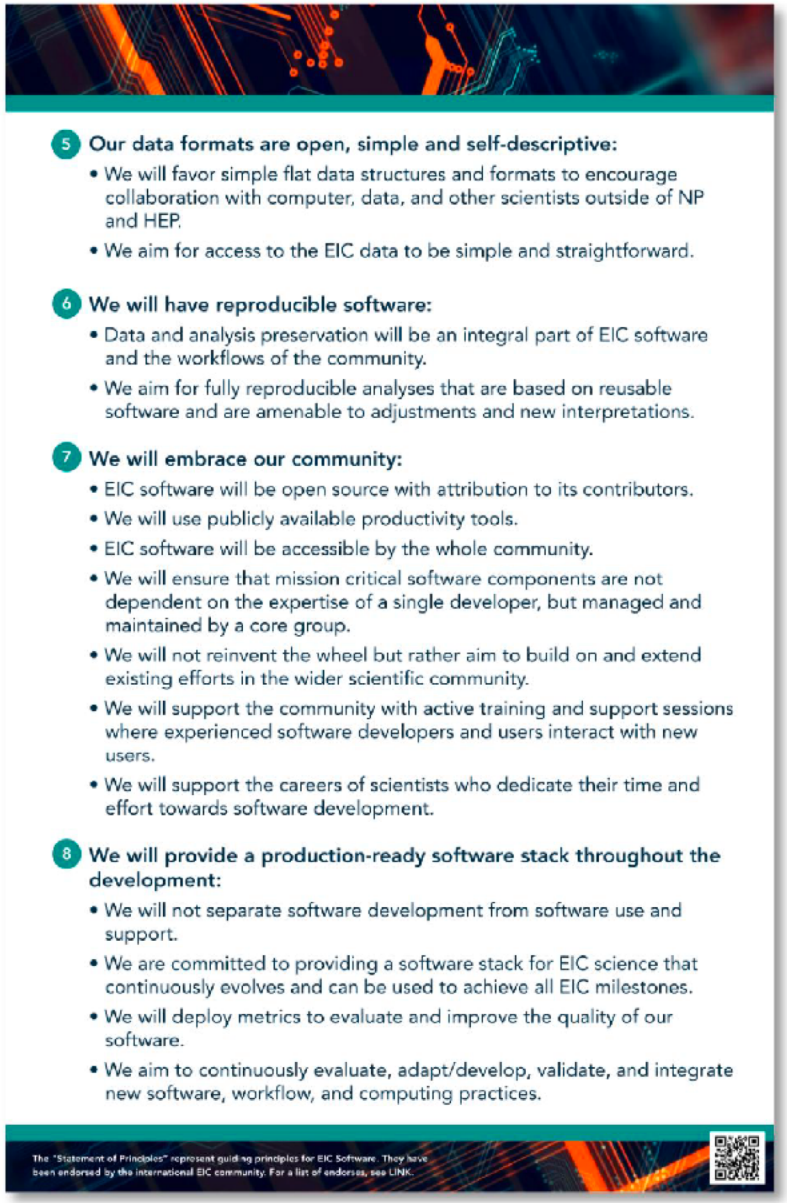
EIC SOFTWARE:
Statement of Principles

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 endorsers, 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

General structure …

https://wiki.bnl.gov/EPIC/index.php?title=Collaboration

SPOKESPERSON'S OFFICE
J. Lajoie (ORNL), Spokesperson
S. Dalla Torre (INFN), Deputy Spokesperson

TECHNICAL COORDINATION OFFICE
Silvia Dalla Torre (INFN, interim)
Prakhar Garg (Yale), Deputy
Oskar Hartbrich (ORNL), Deputy
Matt Posik (Temple), Deputy

ELECTRONICS, READOUT AND DAQ
Fernando Barbosa (JLab)
Jeff Landgraf (BNL)
Jin Huang (BNL)

PID
Thomas Ullrich (BNL)

TRACKING
Ernst Sichtermann (LBNL)

FAR FORWARD/ FAR BACKWARD
Simon Gardner (Glasgow)
Nathaly Santiesteban (UNH)

CALORIMETRY
Oleg Tsai (UCLA)

SOFTWARE AND COMPUTING COORDINATOR
Markus Diefenthaler (JLab), Coordinator
Wouter Deconinck (Manitoba), Deputy
Sylvester Joosten (ANL), Deputy
Torre Wenaus (BNL), Deputy

PRODUCTION
Sakib Rahman (Manitoba)
Thomas Britton (JLab)

PHYSICS AND DETECTOR SIMULATION
Kolja Kauder (BNL)
Chao Peng (ANL)

USER LEARNING
Holly Szumila-Vance (JLab)
Kolja Kauder (BNL)

RECONSTRUCTION
Shujie Li (LBNL)
Derek Anderson (ISU)

VALIDATION
Tori Jeske (JLab)
Dmitry Kalinkin (Kentucky)

ANALYSIS TOOLS
Kong Tu (BNL)
TBD

STREAMING COMPUTING MODEL
Marco Battaglieri (INFN Genova)
Jin Huang (BNL)

ANALYSIS COORDINATORS
Salvatore Fazio (Unical)
Rosi Reed (Lehigh)

INCLUSIVE PHYSICS
Tyler Kutz (MIT)
Claire Gwenlan (Oxford)

SEMI-INCLUSIVE PHYSICS
Charlotte Van Hulse (Alcala)
Stefan Diehl (UConn)

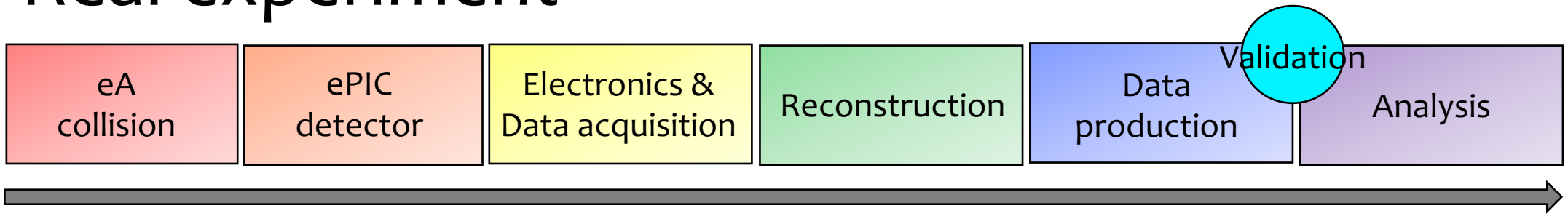
EXCLUSIVE, DIFFRACTION AND TAGGING
Raphael Dupre (Orsay)
Rachel Montgomery (Glasgow)

JETS AND HEAVY FLAVOR
Olga Evdokimov (UIC)
Brian Page (BNL)

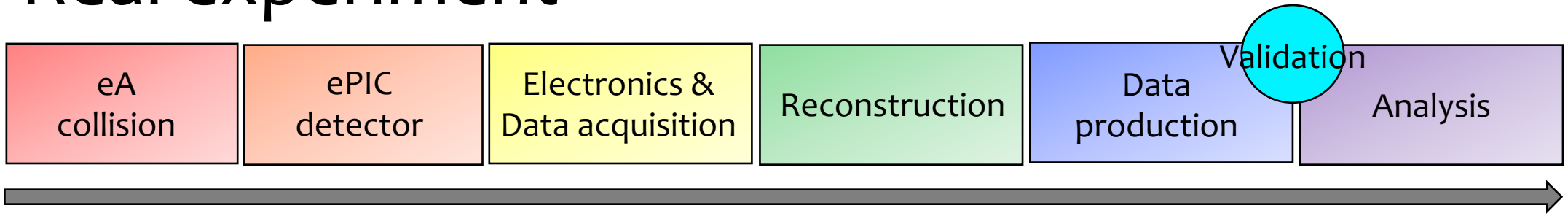
BSM AND PRECISION EW
Ciprian Gal (JLab)
Michael Nycz (Virginia)

ePIC Working Groups

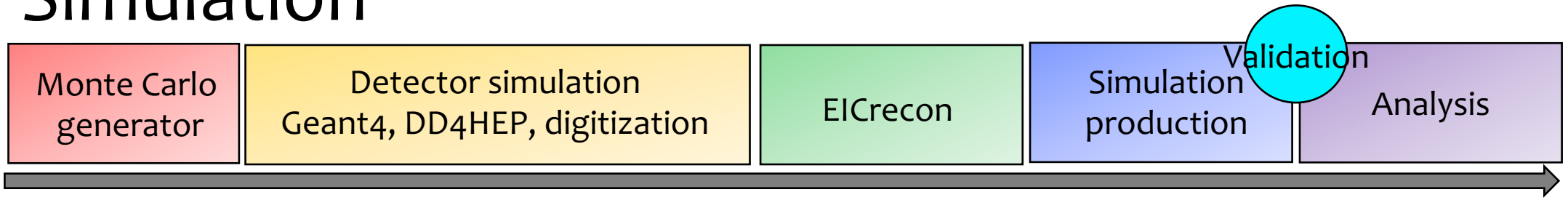
Real experiment



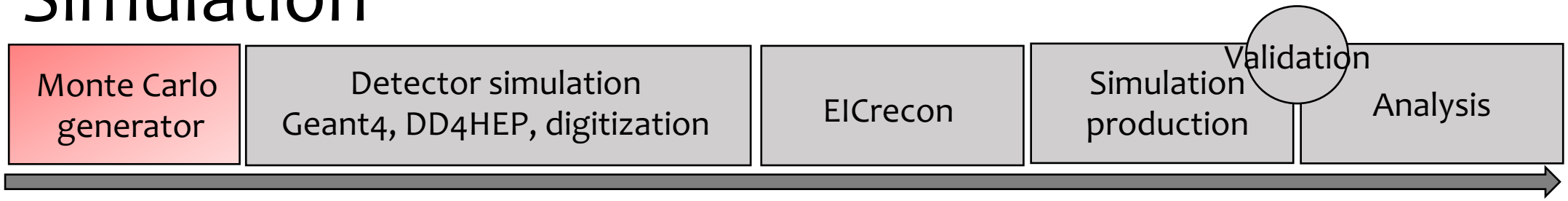
Real experiment



Simulation

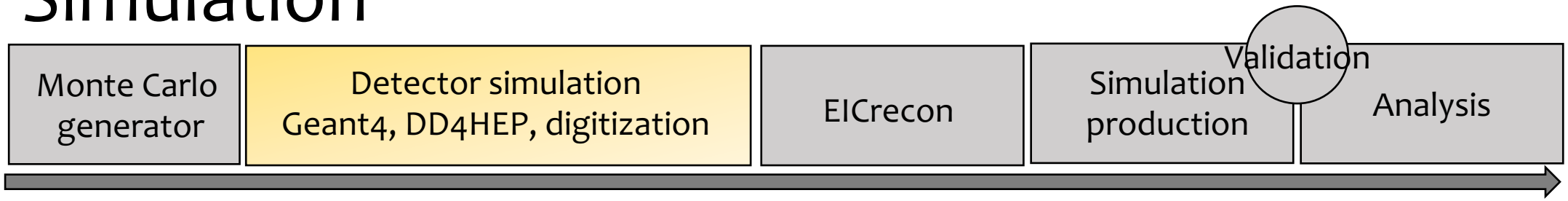


Simulation



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

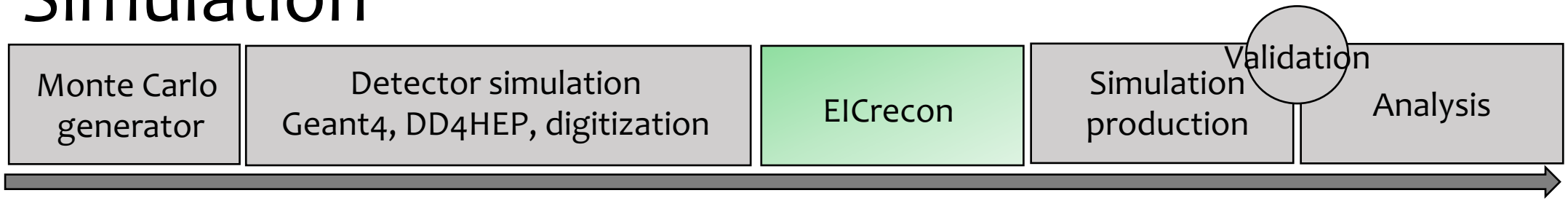
Simulation



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

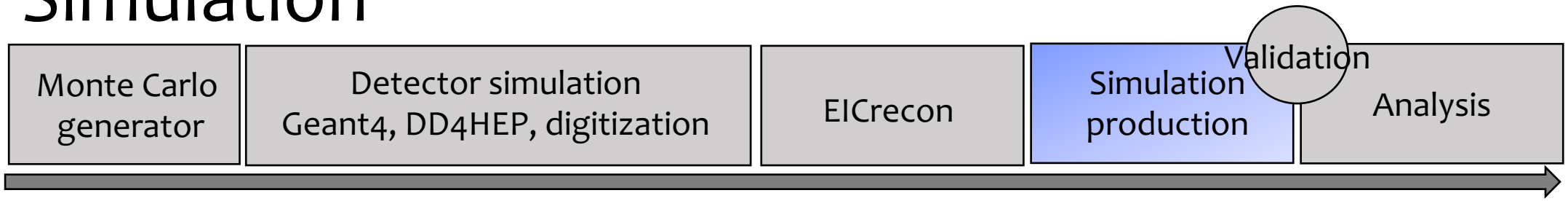
Algorithms to transform the GEANT4 hits to mimic real detector
readout, including background stacking, “pileup”, DAQ frames

Simulation



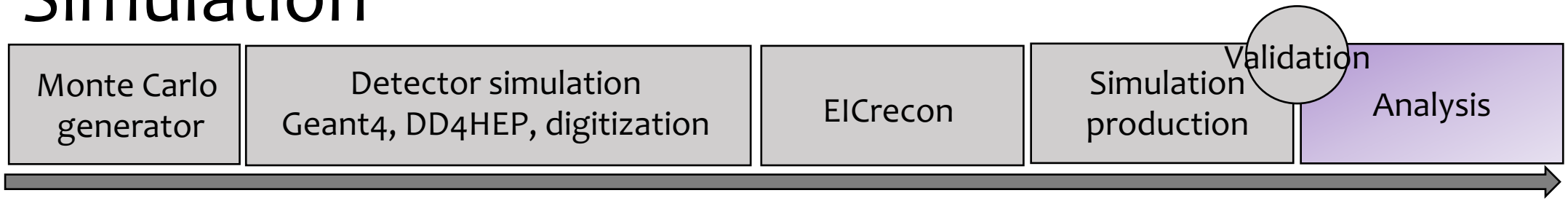
Realistic reconstruction algorithms starting from raw detector output (digitization to look like real data)

Simulation



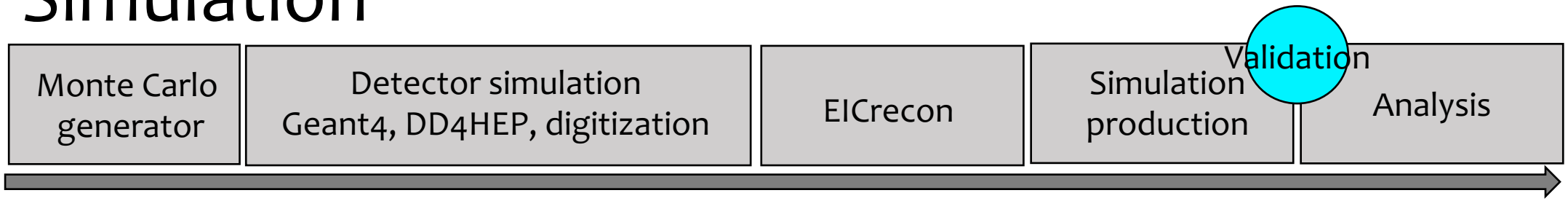
Processing of the data using calibration information, corrections, etc

Simulation



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

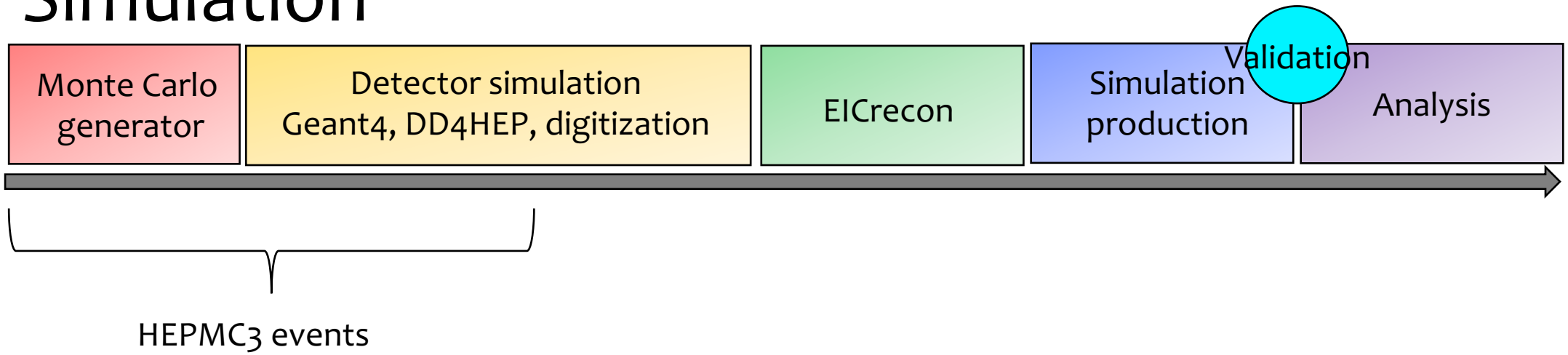
Simulation



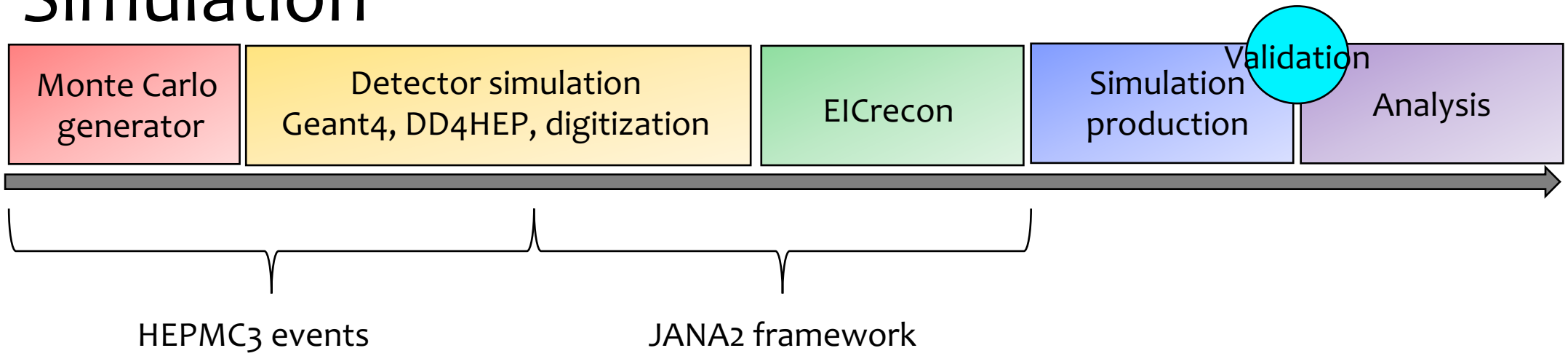
Benchmarks on both detectors and physics

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

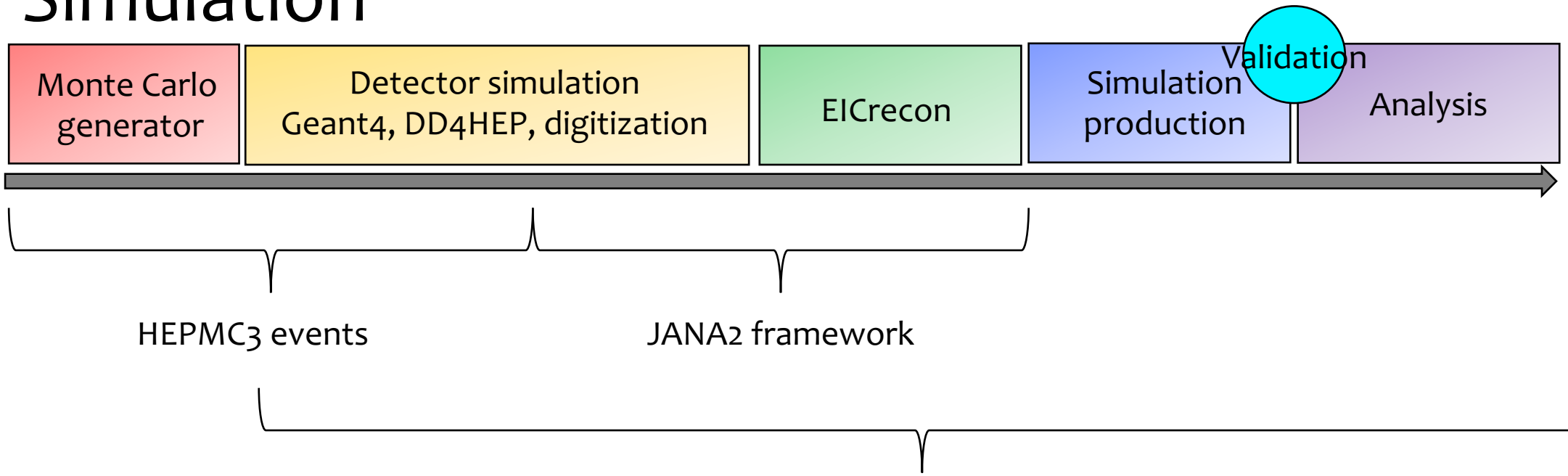
Simulation


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

Simulation

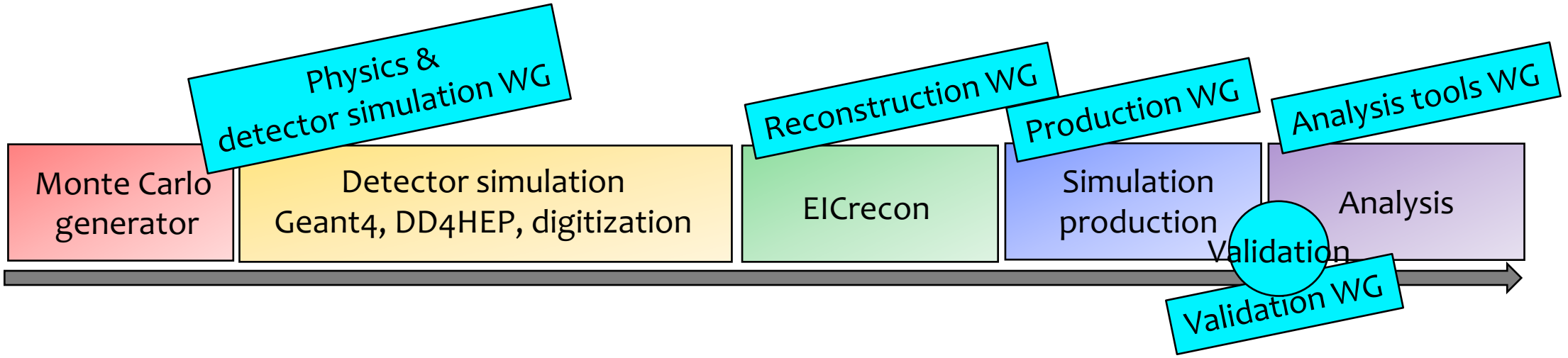

Monte Carlo generator
Detector simulation
Geant4, DD4HEP, digitization
EICrecon
Simulation production
Validation
Analysis
HEPMC3 events
JANA2 framework

Simulation



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



Simulation Campaign Strategy

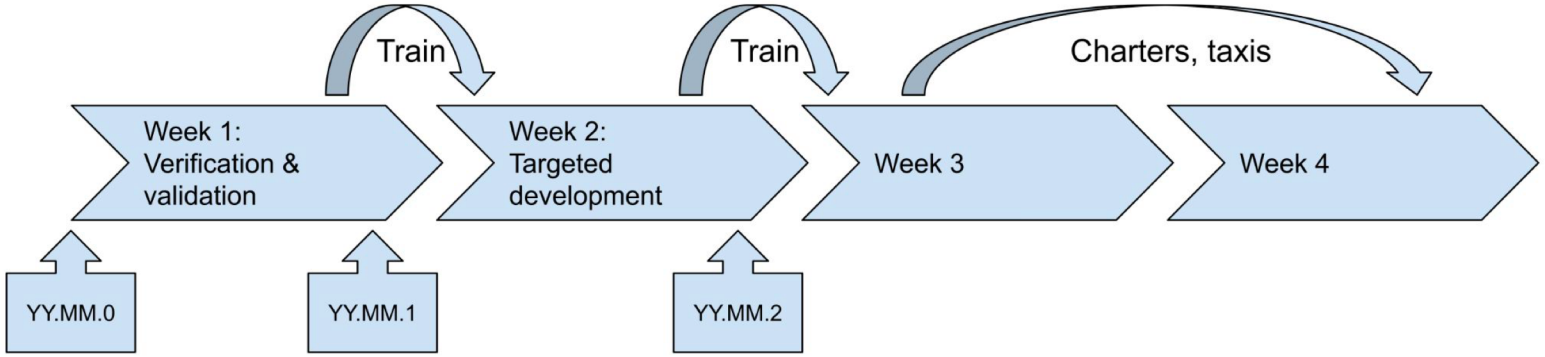
Production-ready!

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

Train: Major central campaign at a fixed (monthly) schedule
Charters: Special interest runs for working groups
Taxis: Bespoke runs for individual users

Latest Completed Production Campaigns (24.02.0, 24.02.1, 24.03.1)

For live campaign updates, follow the firehose mattermost channel.

To see what was run in past campaigns, review our campaign history pages for reconstructed output files and full geant4 simulation output files.

Update is provided at the end of the campaign via email/web. To learn how to access files on xrootd, review our FAQ page.

To learn about our simulation and analysis framework, review the tutorials on the ePIC Collaboration Landing Page.

Example:
Listing and viewing Reconstructed Outputs for a particular dataset

xrdfs root://dtn-eic.jlab.org ls
/work/eic2/EPIC/RECO/24.03.1/epic_craterlake/DIS/NC/18x275/minQ2=10

where the different segments indicate server address, base address, detector config, physics process and beam properties respectively.

The corresponding event generator files will be at
/work/eic2/EPIC/EVGEN/DIS/NC/18x275/minQ2=10
and geant4 simulation output (for small subset) will be at
/work/eic2/EPIC/FULL/24.03.1/epic_craterlake/DIS/NC/18x275/minQ2=10

Copy files locally using xrdcp or open them in root directly

Current Campaign (24.04.0)

We have organized a group of datasets (Physics Processes and Backgrounds) that will be run every campaign (moving gradually towards version control for everything). Besides regular production campaign datasets, we will accommodate charter requests from Physics Working Groups depending on availability of resources after approval by Analysis and Software coordinators (AC/SCs).

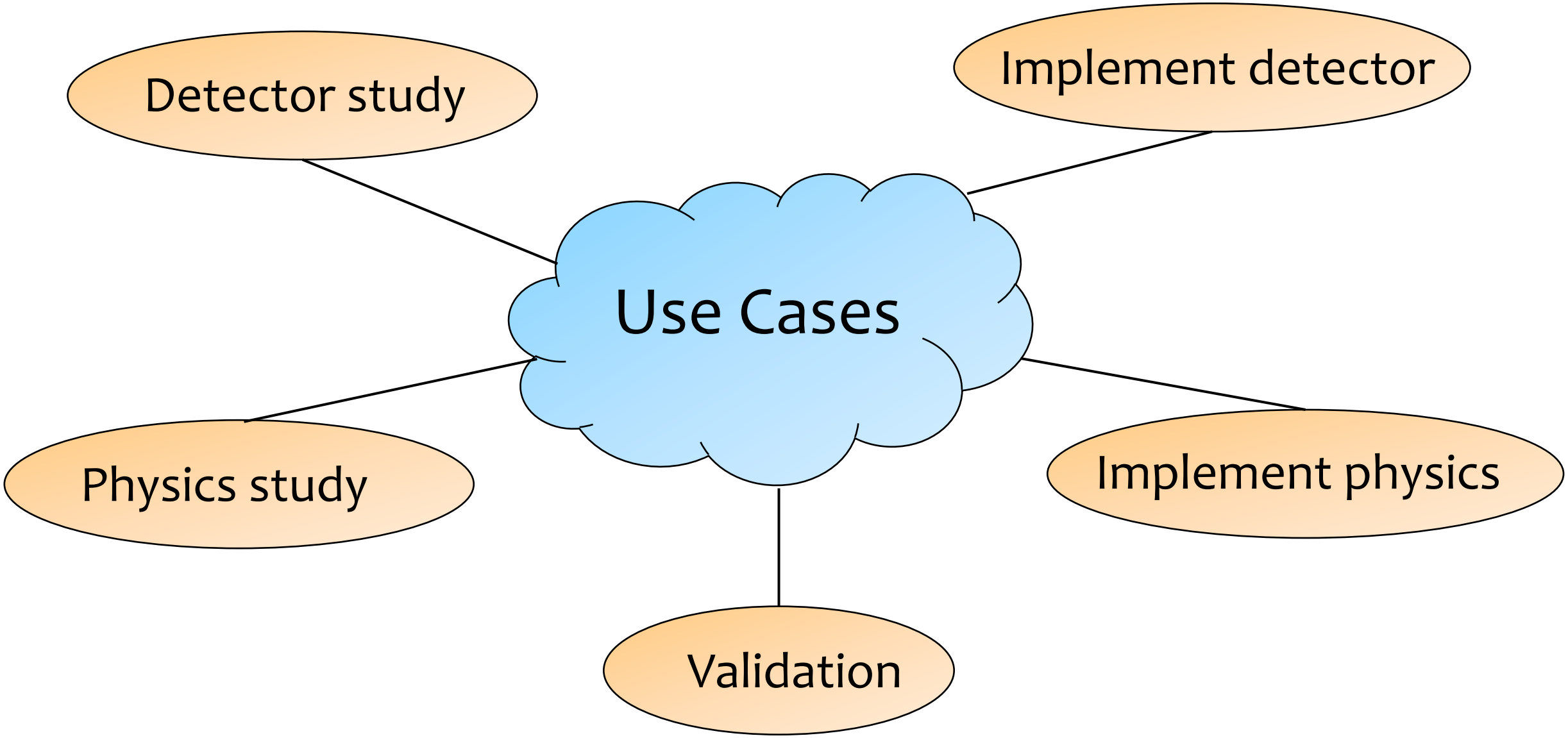
What do I want to study?

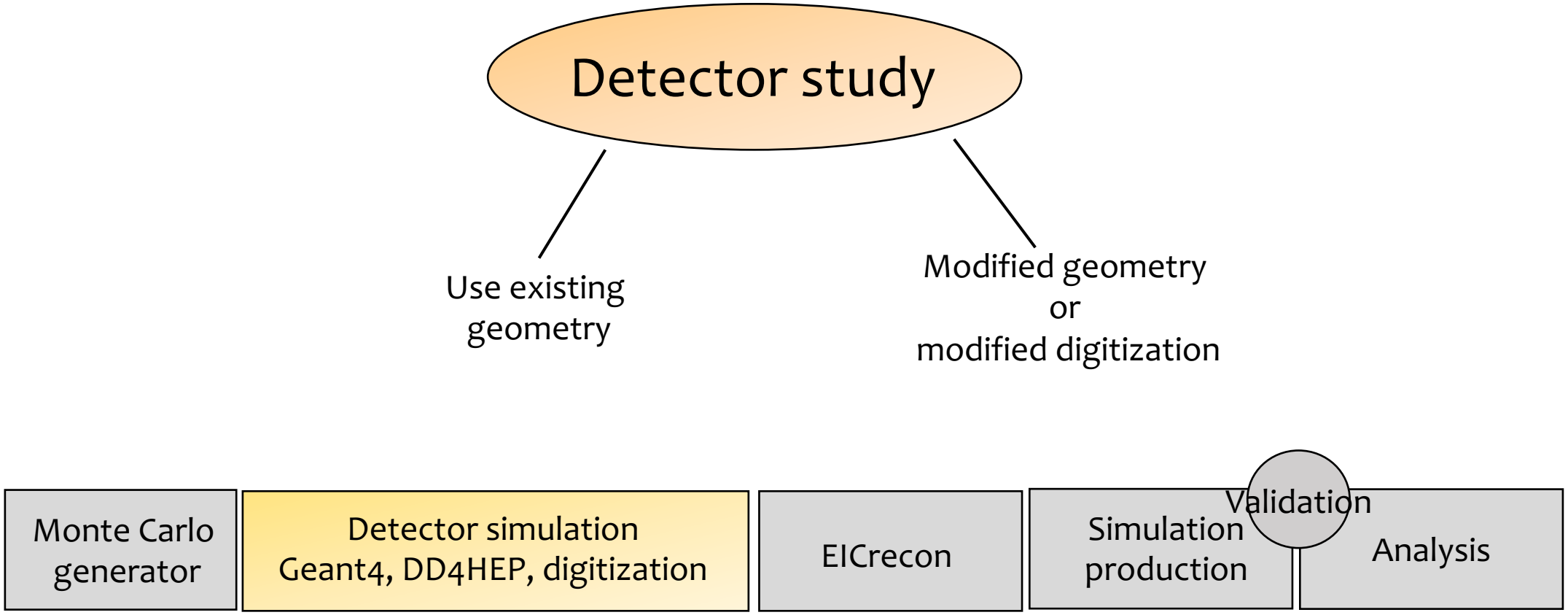
Detector:

- Resolution
- Efficiency
- Digitization
- Optimization
- Geometry
- …

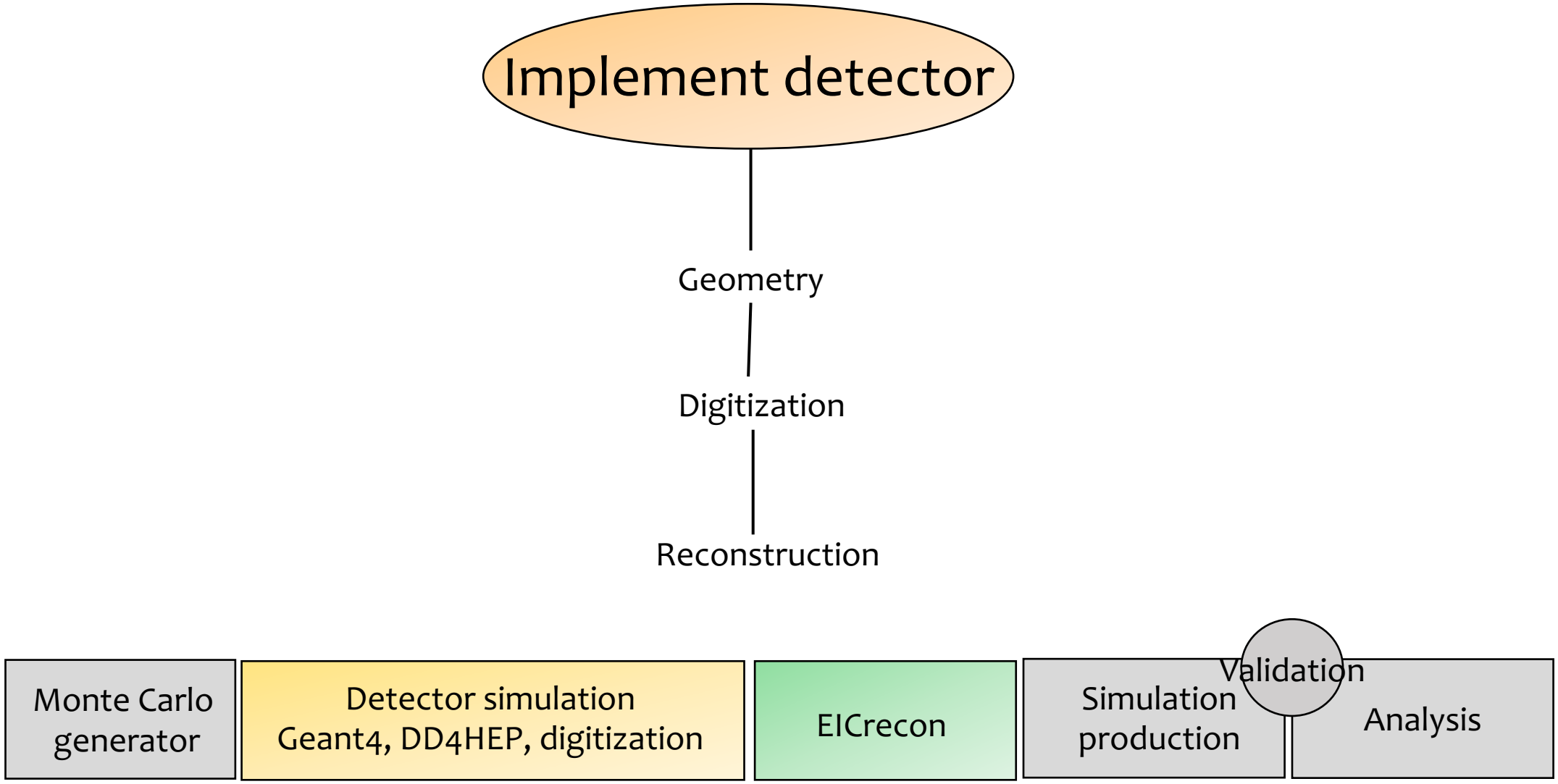
Physics:

- Kinematics
- Phase space
- Acceptance
- Algorithms
- …

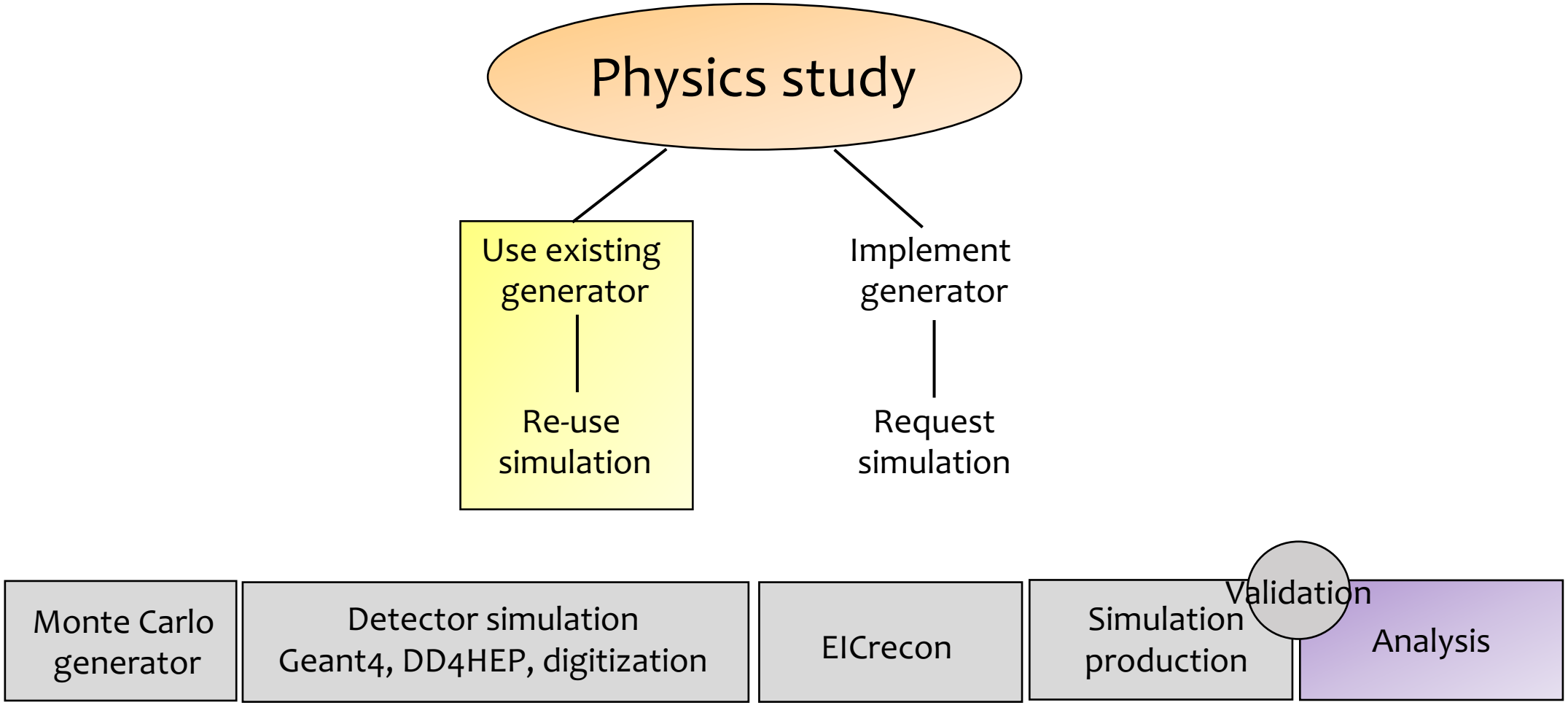

Detector study
Implement detector
Use Cases
Physics study
Implement physics
Validation



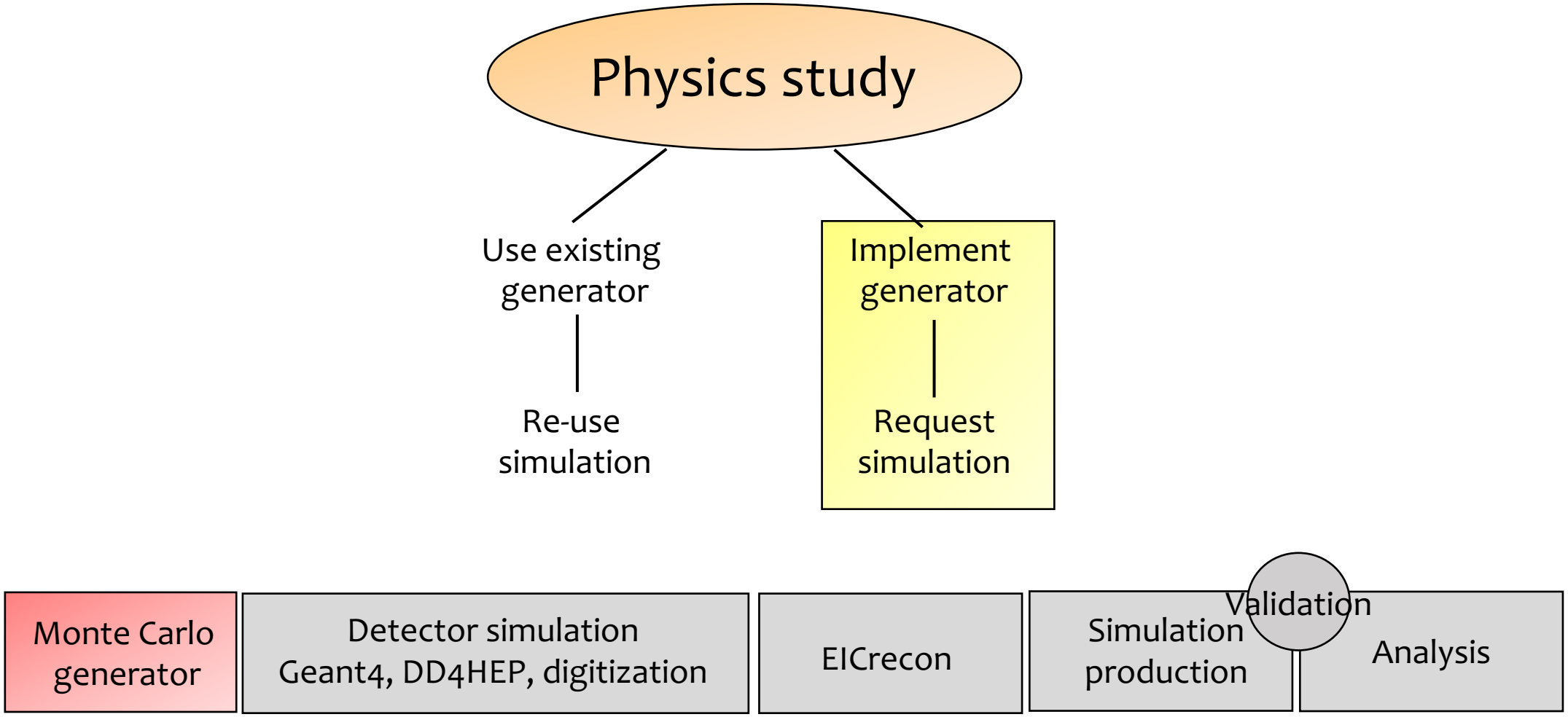
Previous tutorial: https://eic.github.io/tutorial-modifying-geometry-digitization-etc
https://eic.github.io/tutorial-geometry-development-using-dd4hep/



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



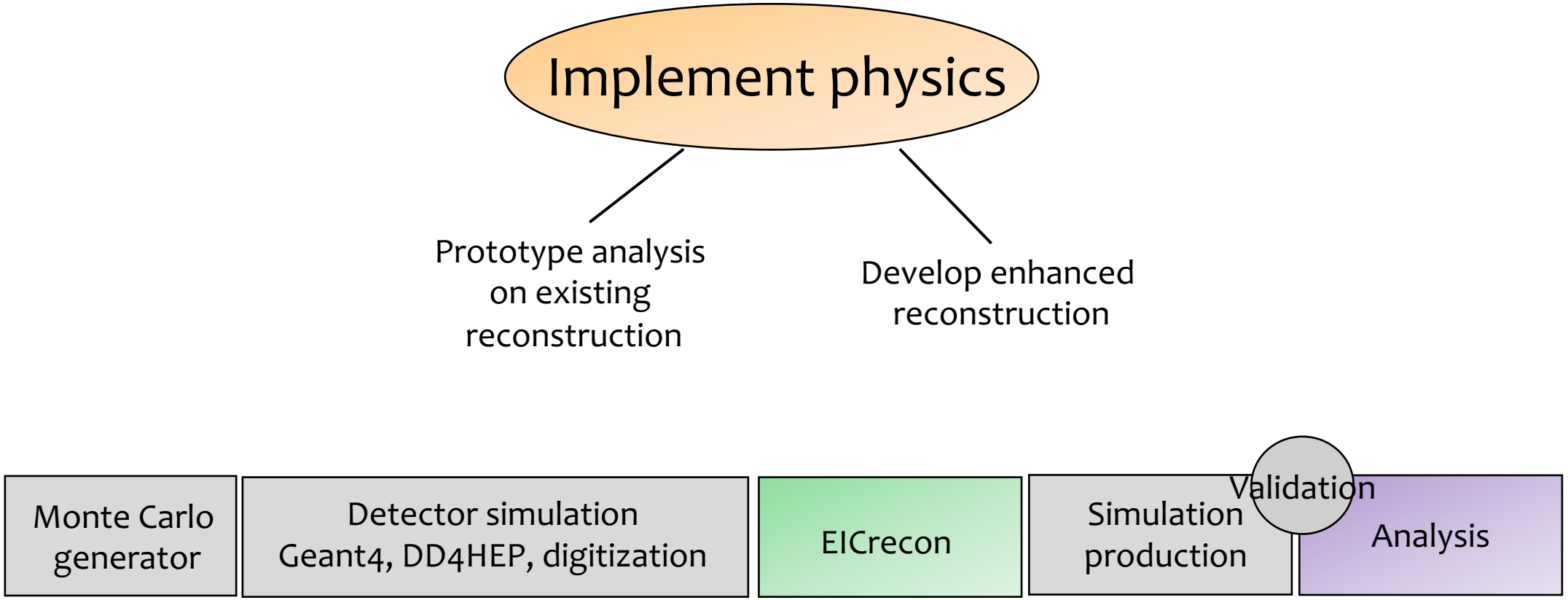
Previous tutorial: https://eic.github.io/tutorial-simulations-using-ddsim-and-geant4/

Reminder: Criteria for MCEGs to be Included in Production

1) Must not duplicate effort. Need to have reference generator for each process.
2) Must be in hepmc3.tree.root format.
3) Must be version-tracked in a publicly accessible repository: Source code, steering files, run cards, etc. Follow the input preprocessing guidelines.

File Nomenclature and Organization

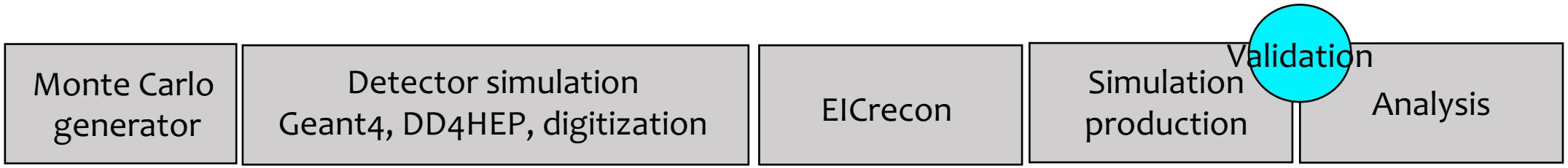
Organization of files	Example
<physics processes>/<generator repository release tag>/<electron momentum>x<proton momentum>/q2_<minimum q2>to<maximum q2>/<generator repository release tag>_<physics processes>_<electron momentum>x<proton momentum>_q2_<minimum q2>to<maximum q2>_run<index>.hepmc3.tree.root	DIS/NC/pythia6.428-1.0/10x100/q2_10to100/pythia6.428-1.0_DIS-NC_10x100_q2_10to100_run001.hepmc3.tree.root



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

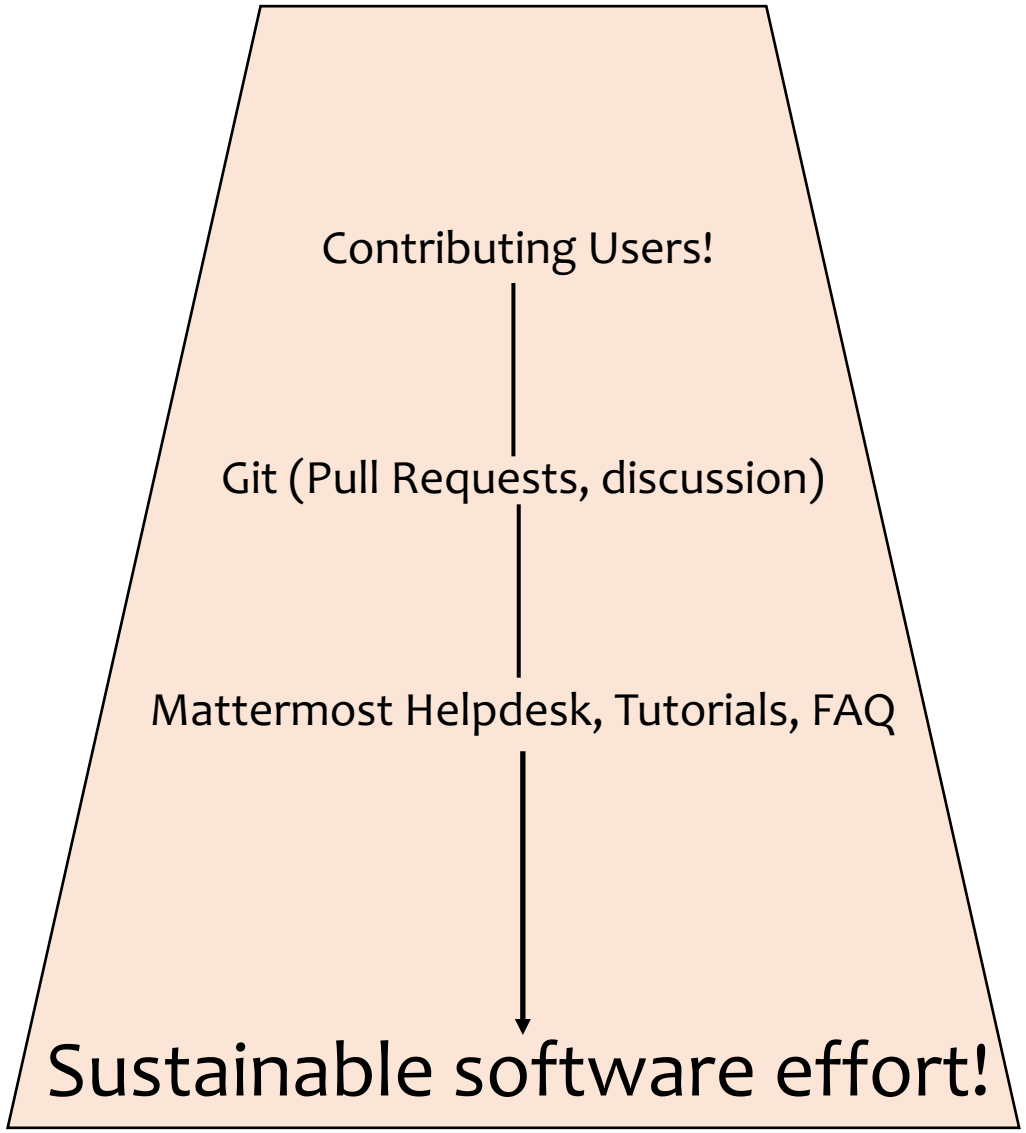
Validation

Compare the output of the physics and detector studies



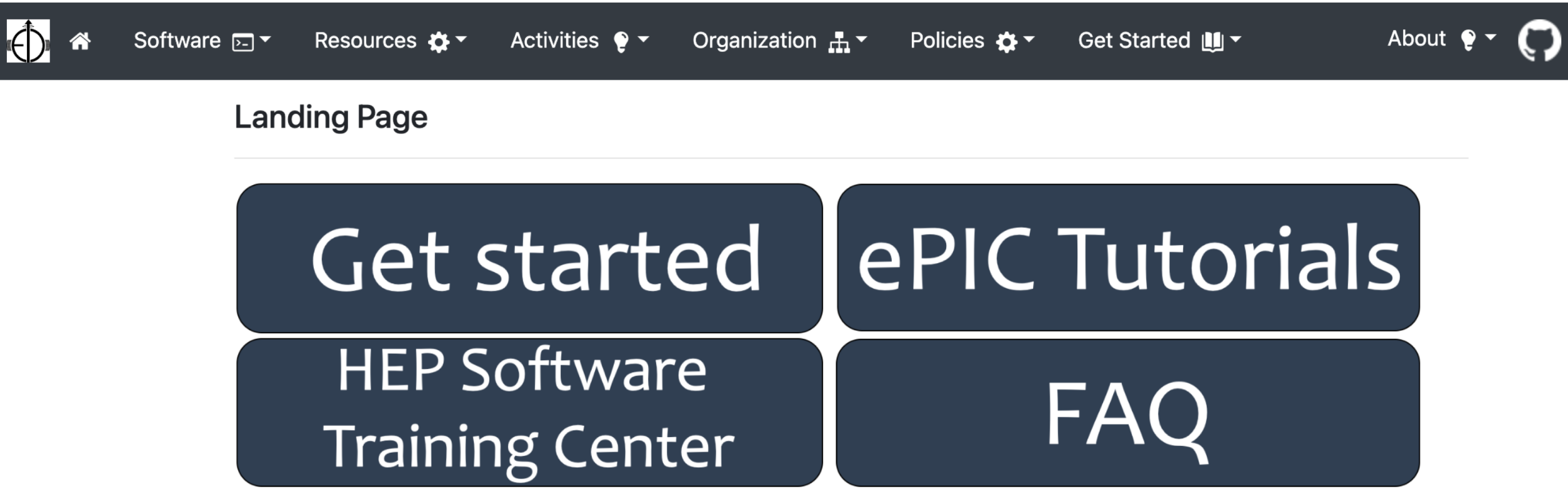
Previous tutorial: https://eic.github.io/tutorial-developing-benchmarks

Requirements to make a collaborative environment



Previous tutorial: https://eic.github.io/tutorial-setting-up-environment

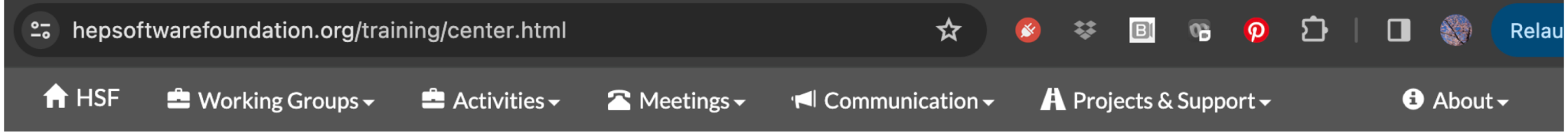
https://eic.github.io/documentation/landingpage.html



Get Started

Welcome to the **Get Started** section. This page will guide you through the steps to get setup for contributing and working in our software framework.

1. Join GitHub: https://github.com/eic
   - GitHub serves as a central platform for version control, code review, issue tracking, and documentation. We maintain the EIC organization on GitHub for collaborative development of all software related to the EIC.
   - **Read Access:** Contact ✉ **eic-software-l-owner@lists.bnl.gov** from your institutional email address. Include in your email your GitHub username and confirmation of whether you or your sponsor/advisor is a member of the EICUG or ePIC.
   - **Write Access:** For access to specific repositories, you can request to join various GitHub teams. For example, join 'EPIC Devs' for software development within the ePIC collaboration.
2. Join Mattermost: https://chat.epic-eic.org/
   - We use Mattermost for our main communication channel.
   - You can join by emailing any group member to be added.
3. Sign up for our mailing lists:
   - Collaboration mailing list (subscribe here): ✉ **eic-projdet-collab-l@lists.bnl.gov**
   - Software mailing list (subscribe here): ✉ **eic-projdet-compsw-l@lists.bnl.gov**
4. Join a project! Checkout the ePIC wiki to get involved:
   - Physics Working Group
   - Detector Subsystems
   - Software Working Group
5. Refer back to the landing page to checkout HEP Software and ePIC tutorials



HSF Software Training Center

See below for more information about the modules listed here.

Basics

The UNIX Shell

A guide through the basics of the file systems and the shell.

Start learning now!

Contribute!

Version controlling with git

Track code changes, undo mistakes, collaborate. This module is a must.

Start learning now!

Contribute!

Programming with python

Get started with an incredibly popular programming language.

Start learning now!

Contribute!

SSH

Introduction to the Secure Shell (SSH)

Status: Early development

Start learning now!

Contribute!

Machine learning

Get behind the buzzword and teach machines to work for you intelligently!

Start learning now!

Watch the videos!

Contribute!

Matplotlib for HEP

Make science prettier with beautiful plots!

Status: Beta testing

Start learning now!

Contribute!

ePIC Tutorials

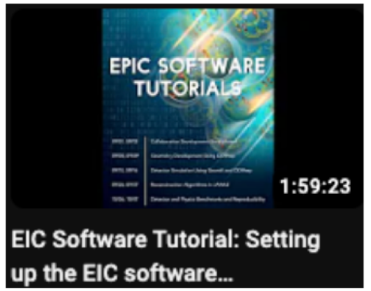
Submit feedback here!

Submit feedback on the Apr. 2024 tutorials here

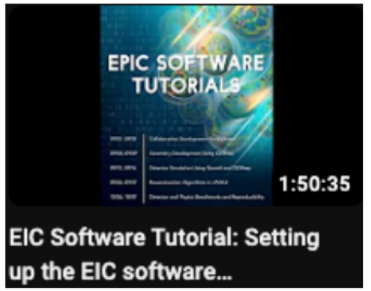
Future tutorials & schedule: Check back soon!

Setting up an environment

Setup	Download files required for the lesson
1. Introduction	What is the EIC environment?
2. The EIC Software Environment	What is the EIC environment?
3. Using GitHub with EIC Software	How do we use GitHub within the EIC community?



EIC Software Tutorial: Setting up the EIC software...



EIC Software Tutorial: Setting up the EIC software...

Analysis and working with the simulation output

Setup	Download files required for the lesson
1. Introduction	How do I locate and access the simulation output?
2. The Reconstruction Output Tree	How is the reconstruction output tree populated?
3. Analyzing the Reconstruction Output	How does one utilize the reconstruction output trees to do an analysis?

Simulating detectors and their readout

Great place to get involved!

FAQ

Welcome to the **Frequently Asked Questions** page! Please use the Helpdesk on Mattermost for urgent correspondence.

Submit new questions and comments by email: ✉

Q: How can we browse the simulation output from a specific campaign and locate certain output files?

A: Visit the epicprod website to view the list of avaiable campaigns. Pick a campaign that you want to view in detail. For example: 23.12.0. Here, you will find the output directories listed in a nested tree structure. The directory nomenclature usually follows the pattern:

`<base directory>/<campaign tag>/<detector config>/<physics processes>/<generator release tag if available>/<electron momentum>x<proton momentum>/<q2 range>/`

The preferred method to list the files in a directory is to use the xrdfs interface within eic-shell container. For example:

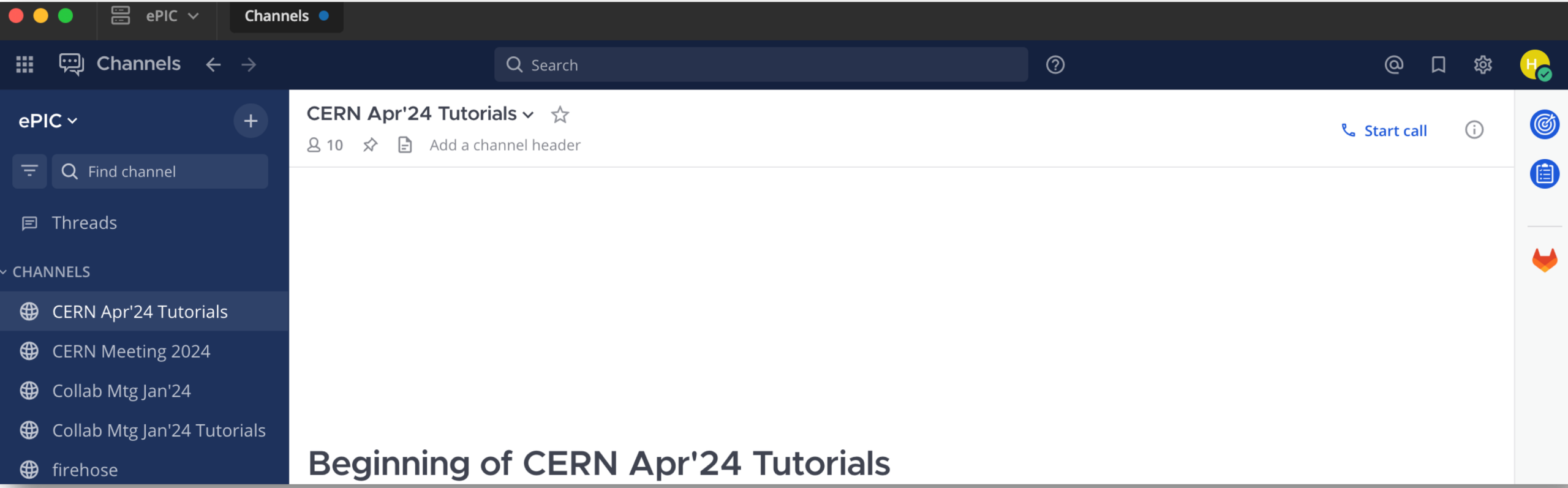
```
xrdfs root://dtn-eic.jlab.org
ls /work/eic2/EPIC/RECO/23.09.1/epic_craterlake/DIS/NC/18x275/minQ2=1000
```

See more details here.

Q: What input datasets are used in a particular production campaign and where can I find them?

A: The directory structure of the input datasets mimic the directory structure of the output files from <physics processes>

Join the Mattermost channel for this tutorial



Questions or issues before we begin?