# The NTuple Wizard

## An NTuple production service for accessing LHCb Open Data

Chris Burr[1], Dillon Fitzgerald[2], Adam Morris[1,3], **Ryunosuke O'Neil[4]**,
Donijor Tropmann[1] *on behalf of the LHCb Collaboration*
[1]CERN [2]University of Michigan [3]University of Bonn [4]University of Edinburgh

## Introduction

**Making the large datasets collected at the LHC accessible to the public is a considerable challenge given the volume and complexity of data.**

- We aim to enable meaningful access to the data by the broadest physics community possible.
- The **LHCb NTuple Wizard** enables third-party users to request derived data samples in the same format used in LHCb physics analysis (**NTuples**).
- Issues of computer security and access control are addressed within the design, while still offering datasets suitable for scientific research through the CERN Open Data Portal (`https://opendata.cern.ch`).

## LHCb Open Data Policy[1]

- Open the same datasets that are used by LHCb internally.
- Release 50% of an LHCb Dataset after 5 years, increasing to 100% after 10 years.
- Provide tools to perform calibrations & corrections - but no further support.
- Require proper acknowledgement in papers that use LHCb Open Data.
- LHCb Collaborators cannot sign papers that use LHCb Open Data.

## NTupling @ LHCb

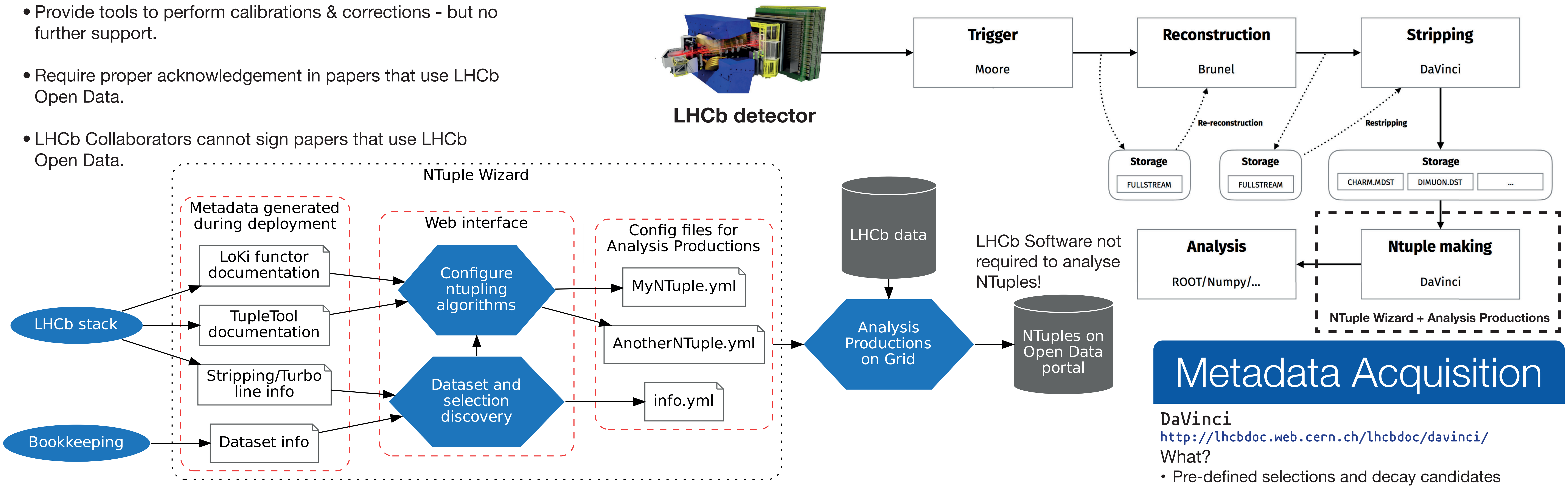
**NTuples are an ordered set of particle or decay candidates cataloging measured quantities chosen by the user.**

- **NTuples** are created by the LHCb `DaVinci` Analysis application using LHCb datasets.
- `DecayTreeTuple` selects pre-built decay candidates and writes a ROOT `TTree` with information on each particle in the decay, plus the event itself.
- NTuples in LHCb are stored in ROOT files and have a rectangular structure, with one entry/row per reconstructed decay candidate. This is in contrast to "jagged" structure NTuples encountered in other experiments.
- `TupleTools` configure what information is written (C++ classes configured via a Python interface)
- In LHCb, the **Analysis Productions** framework is used to submit distributed computing jobs to the Grid, which run the `DaVinci` application and produce NTuples (see right).

## Analysis Productions

**Configuration, testing, and automation of distributed computing workflows used to produce NTuples for LHCb analyses.**

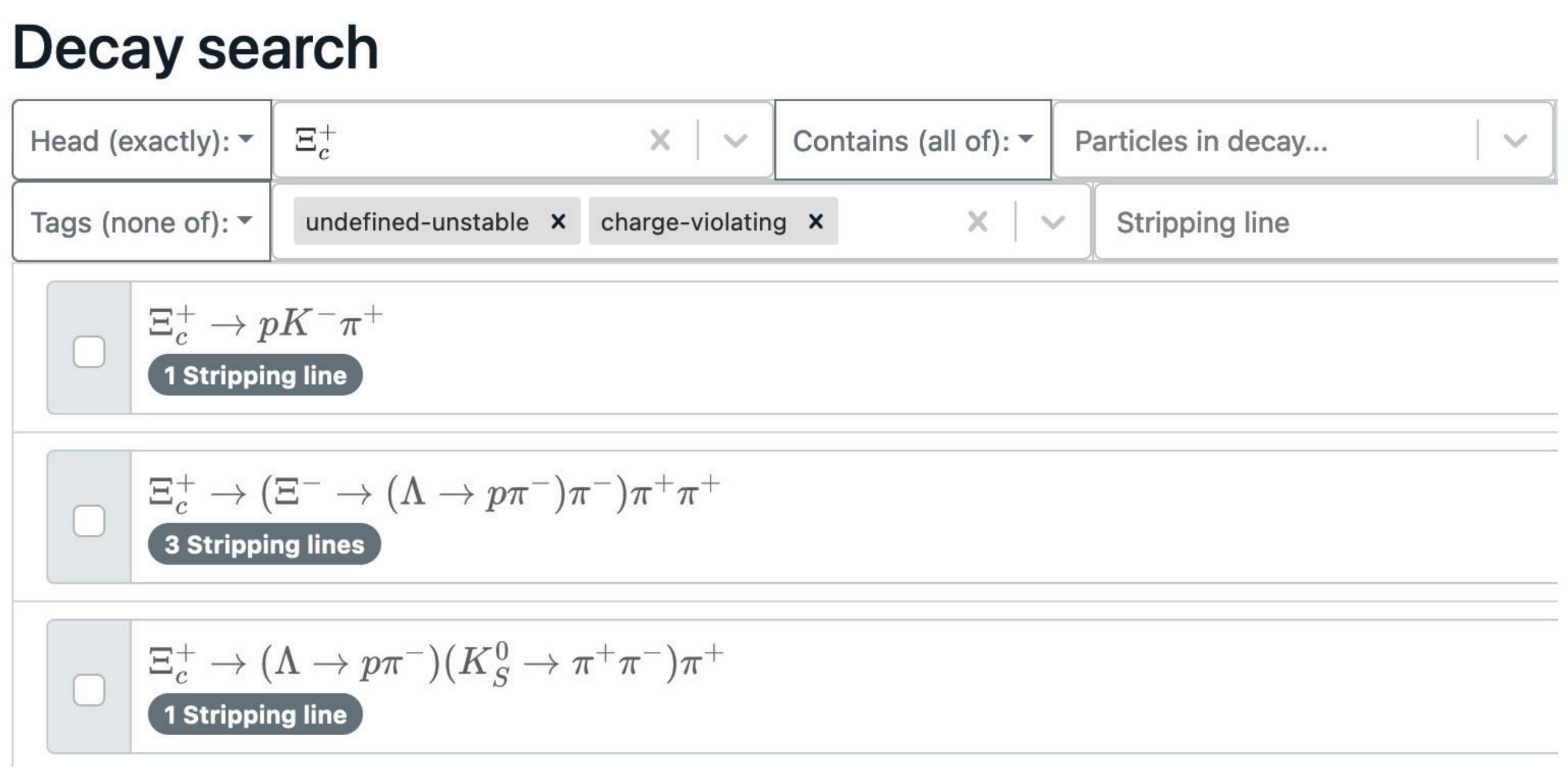
- Created to simplify or automate the workflows involved in producing NTuples, due to an anticipated increase in data volumes expected for Run 3.
- Users write YAML to configure their production, and Python-based configuration files for the analysis application(s).
- Automated continuous integration testing and local testing of proposed productions are part of the workflow.
- Job submission, monitoring, followed by storage, and preservation of analysis metadata are considered in the design, ensuring LHCb NTuples are preserved and reproducible.
- Analysis Productions are the default workflow for producing NTuples for analysis from 2022 onwards.



## The NTuple Wizard

**A web app for configuring an NTuple production with minimal knowledge of LHCb software.**

1. Choose from any of the pre-defined decays and selections present in the data.



### Decay search

□ $\Xi_c^+ \to pK^-\pi^+$ — 1 Stripping line

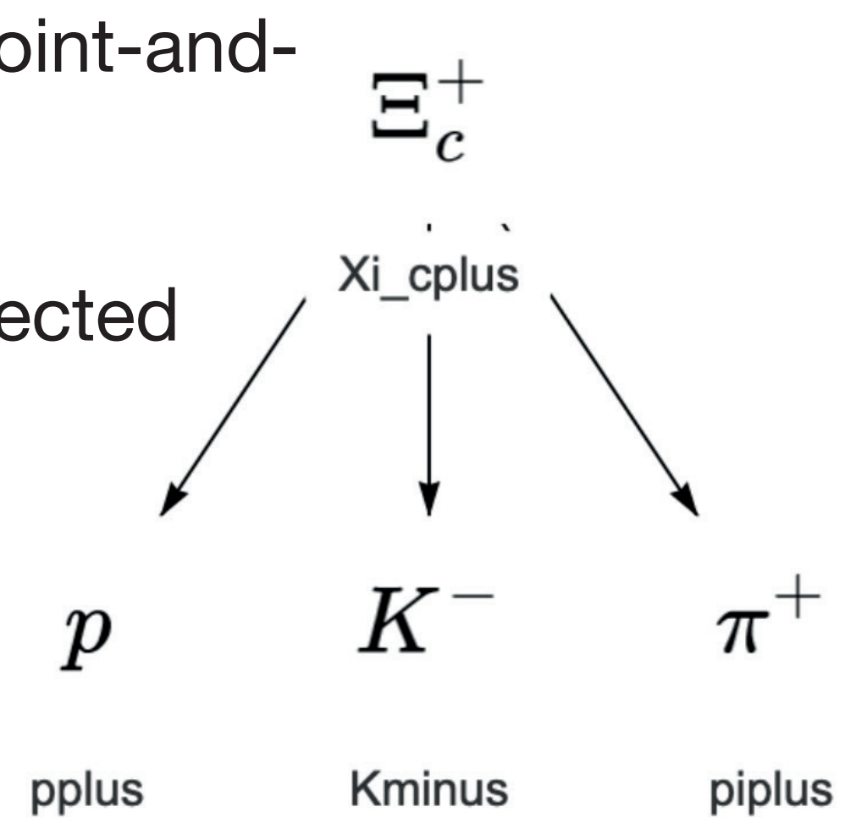
□ $\Xi_c^+ \to (\Xi^- \to (\Lambda \to p\pi^-)\pi^-)\pi^+\pi^+$ — 3 Stripping lines

□ $\Xi_c^+ \to (\Lambda \to p\pi^-)(K_S^0 \to \pi^+\pi^-)\pi^+$ — 1 Stripping line

2. Select from datasets containing the chosen decay candidates.

3. Configure `DecayTreeTuple` with point-and-click interface.

4. Decays are visually rendered as directed acyclic graphs, and each node is configurable.



`TupleTool` documentation is always shown in the correct context.

### Security and permissions

- LHCb applications are traditionally configured with Python scripts. Potential security concern accepting & running code from the public!
- NTuple Wizard output is pure data structures (YAML) containing the configuration, to be interpreted by internal parsers.
- Information from the LHCb software stack and bookkeeping service is scraped at deployment time and served statically (see **Metadata Acquisition**).
- The NTuple Wizard design avoids security issues while providing a smooth experience and access to LHCb Data.

## Metadata Acquisition

`DaVinci`
`http://lhcbdoc.web.cern.ch/lhcbdoc/davinci/`
What?

- Pre-defined selections and decay candidates
- Algorithm configuration interfaces

How?
Run `Gaudi` Python scripts in an LHCb container.

`Doxygen`
What?

- LHCb Software Stack
- Documentation of `TupleTools` and selection functors.

How?

- Non-trivial URL discovery with custom scripts.
- Parse HTML pages with `BeautifulSoup4` (`https://pypi.org/project/beautifulsoup4`)

`scikit-hep/particle`
`https://github.com/scikit-hep/particle`
What?

- Physical properties and particle categories
- Particle names in HTML and LaTeX.

How?
Custom Python code.

`LHCb Bookkeeping`
What?
Paths to available LHCb datasets.
How?
Query `LHCbDIRAC` [2] (the distributed computing workflow management system managing LHCb resources).

### References and Links

[1] "*CERN Open Data Policy for the LHC Experiments*," CERN, Geneva, Tech. Rep., Nov. 2020. [Online]. Available: https://cds.cern.ch/record/2745133

[2] F. Stagni et al., "*LHCbDirac: distributed computing in LHCb*," Journal of Physics: Conference Series, vol. 396, no. 3, p. 032104, Dec. 2012, doi: 10.1088/1742-6596/396/3/032104.