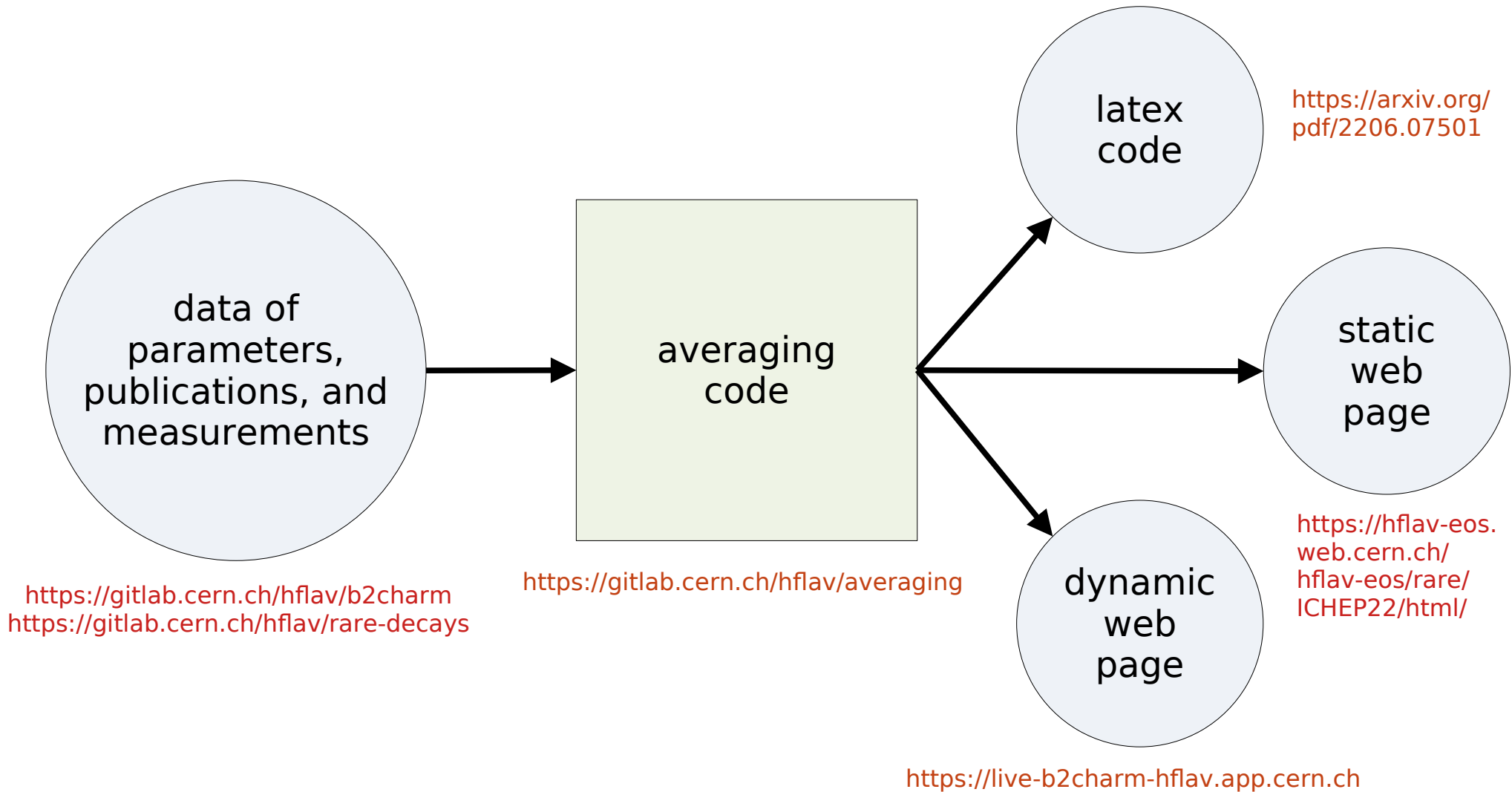


# **Tutorial on Averaging Code**

10.10.2022

# Overview

---



# Design Principles

---

- **Make the addition of new results as easy as possible**
  - Take values as given in the papers  
(no manual update for changed input parameters)
  - json format
  - Import available data from inspire
  - Generate latex code from parameter name (→ requires convention)
  - Warn about potential issues
  - Easy installation (on not too old systems)
- **Let the framework do what it can do**
  - Automated averaging of a large number of parameters
  - Propagation of updates of input values
  - Obtaining of PDG values
  - Generation of output in multiple formats
  - Full deploy chain to live web page

# Design Principles

---

- **Maximize the usefulness for our customers**
  - Easy and intuitive search for averages
  - Quote averages for the same parameters as PDG: (derived) BRs, not ratios or products of BRs
  - Comparison with PDG averages
  - Visualization of input measurements
  - Links to all input measurements
  - Information about correlations among parameters
  - Custom summary plots
- **Welcome contributions to the averaging code**
  - Open source averaging code
  - Python
  - Documentation (and this tutorial)

# Averaging Method

---

- Construct negative log likelihood (NLL) of parameters  $p_i$  and measurements  $x_j \pm \sigma_j$  and minimize it
- Terms for measurements, external parameters (e.g. D BRs), and correlations
- Central value with symmetric uncertainty:  $NLL = \frac{(p - x)^2}{2\sigma^2}$
- Central value with asymmetric uncertainty:  $NLL = \frac{(p - x)^2}{2V}$   
with  $V = \sigma_{\pm}^2$  outside  $\pm\sigma_{\pm}$ , linear inside  $\pm\sigma_{\pm}$
- Functions of parameters can be used, e.g. ratio of Brs
- $x \rightarrow x + n * \sigma_{\text{syst}}$  with normal distributed nuisance parameter  $n$  for measurement with correlated systematic uncertainty  $\sigma_{\text{syst}}$
- Arbitrary NLL can be given as python function (not tested yet)

# Let's Do Some Averaging...

---

Documentation: <https://gitlab.cern.ch/hflav/averaging/-/blob/main/README.md>

- Installation in virtual environment
- Execution of averaging code

Example:  $B^0 \rightarrow J/\psi \eta$

- Define parameter (see naming convention!)
- Add limit from BaBar
- Run averaging and look at the result
- Add measurement from Belle
- Add ratio measurement from LHCb and external parameter
- Generate latex code
- Add an input parameter
- Add a correlation between measurements

# Advanced Topics

---

- `hflav_rename`
- Adding a new particle
- Deployment
- ...

# Further Ideas for Improvements

---

- Protected preview web page for merge requests
- Documentation/script for deployment
- Upgrade to django 4
- API for averages/likelihoods
- Open source data?
- Consistent change of particle names to new convention?
- ...