A RIVETing journey:
Analysis Preservation and Generator Tuning in High Energy Physics
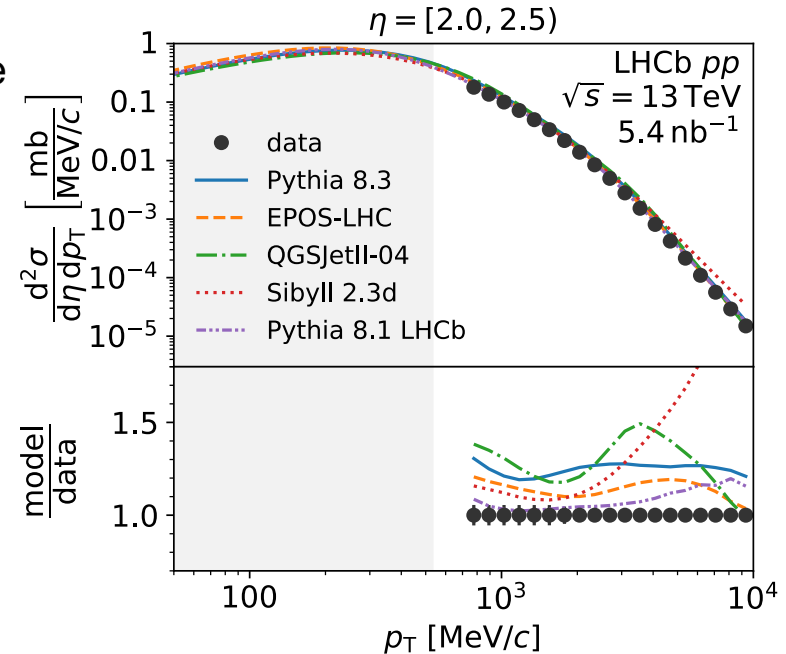
Johannes Albrecht, Hans Dembinski, **Lars Kolk**

Workshop on the tuning of hadronic interaction models
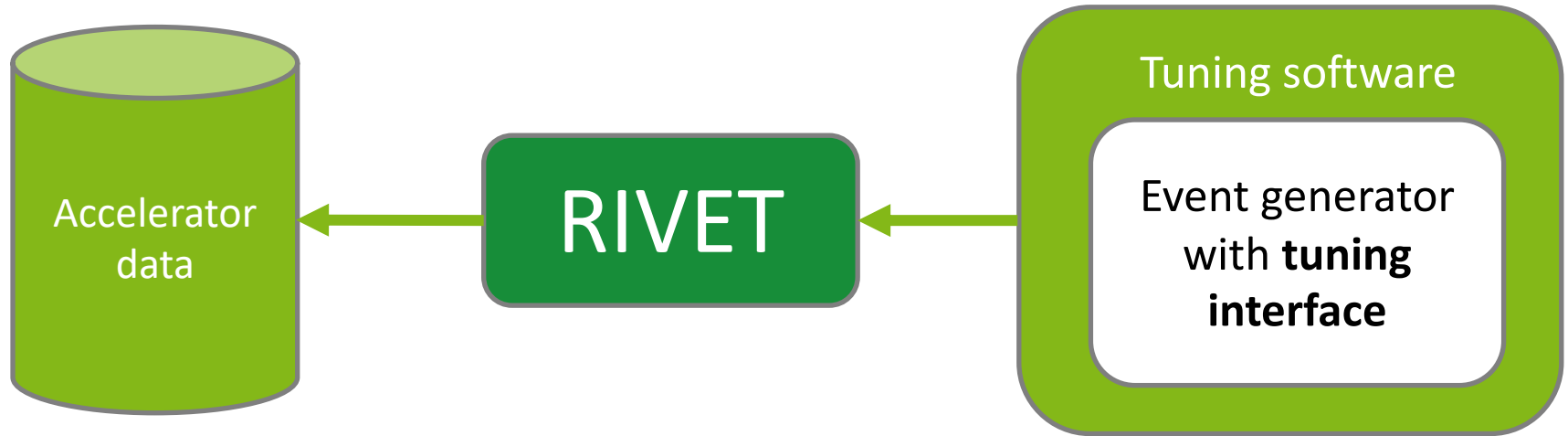
Wuppertal, 24.01.2024

Motivation

- Goal: Make paper results more accessible/usable
  - Make results publicly accessible
  - Provide results in machine readable format
  - Provide interface for **generator tuning**

- Archiving published analyses for public access
  - Promotes transparency and reproducibility
  - Increases impact of the analysis/paper
  - Improves utilisation of analysis for **generator tuning**

- Analysis preservation
  1. Data: HEPdata
  2. Analysis Code: Rivet

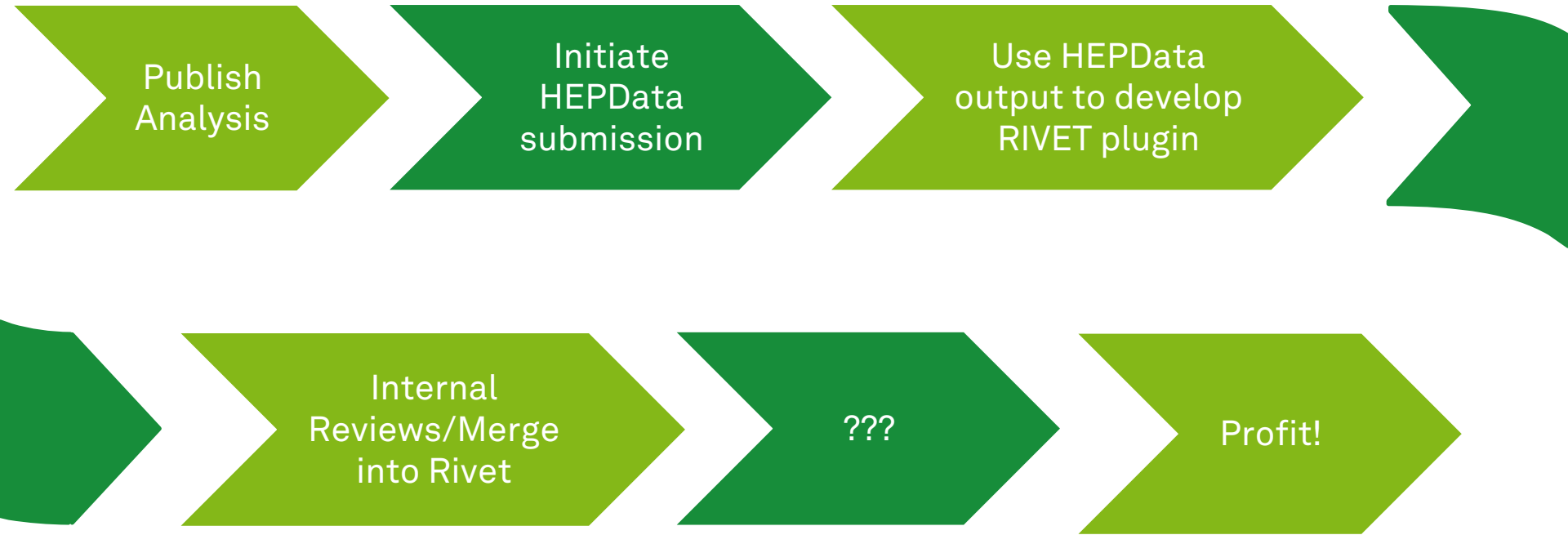$\eta = [2.0, 2.5)$

LHCb pp
$\sqrt{s} = 13\,\text{TeV}$
$5.4\,\text{nb}^{-1}$

- data
- Pythia 8.3
- EPOS-LHC
- QGSJetII-04
- SibyII 2.3d
- Pythia 8.1 LHCb

$\frac{d^2\sigma}{d\eta\,dp_{\text{T}}} \left[\frac{\text{mb}}{\text{MeV}/c} \right]$

$\frac{\text{model}}{\text{data}}$

p_{T} [MeV/c]

LHCb Collaboration, *JHEP* 01 (2022) 166

Lars Kolk

2

HEP Generator tuning in a nutshell

Accelerator data → RIVET ← Tuning software / Event generator with **tuning interface**

Modified from H. Dembinski

Analysis Preservation Roadmap

Publish Analysis → Initiate HEPData submission → Use HEPData output to develop RIVET plugin →

Internal Reviews/Merge into Rivet → ??? → Profit!

Lars Kolk

4

HEPDATA

Lars Kolk

HEPData

- HEPData: Open-access repository used to preserve and share experimental data
  - Maintained by CERN
  - Huge catalogue of preserved analyses
  - Ensures long-term preservation of experimental results

- HEPData Entries
  - Can be searched via many categories
  - Come with abstract and proper citation
  - Store data in tables

Lars Kolk

Example HEPData entry

Example HEPData entry

HEPData-Submissions

- Each submission consists of …
- One submission.yaml
  - Links to every table in the submission
  - Provides metadata for the HEPData entry
- One .yaml file for every table in the entry
  - Contain data points, errors
- Refer to documentation or previously uploaded entries :)

- Contact your coordinator to initiate a submission

- Use hepdata-validator or HEPData sandbox to validate your files

Lars Kolk

9

HEPData-Submissions

- Each submission consists of ...
- One submission.yaml
  - Links to every table in the submission
  - Provides metadata for the HEPData entry
- One .yaml file for every table in the entry
  - Contain data points, errors
- Refer to documentation or previously uploaded entries :)

- Contact your coordinator to initiate a submission

- Use hepdata-validator or HEPData sandbox to validate your files

Lars Kolk

10

RIVET

Lars Kolk

Rivet

- **Rivet:** Most widespread way by which analysis code from the LHC and other high-energy collider experiments is preserved.
  - Used for generator tuning
  - Provides set of experimental analyses
  - Rivet wishlist, installation

- Analysis name: <Experiment>_<publication_year>_I<InspireID>
  - E.g.: LHCB_2021_I1889335

- Each Rivet-analysis consists of one ...
  - .cc file containing the analysis code
  - .yoda file containing the data points (from HEPData)
  - .info file containing metadata
  - .plot file setting plotting options

Lars Kolk

12
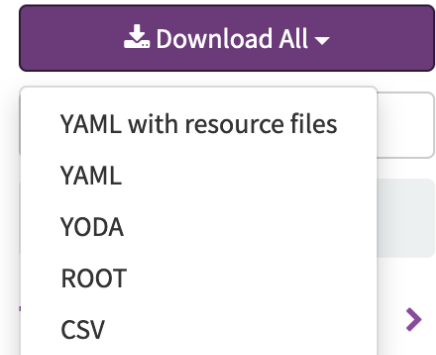
Rivet Analysis Code

- An analysis is usually written in **one** .cc file containing
  - A no-argument constructor
  - Three analysis event loop methods:
    1. init: book histograms, initialise counters, etc.
    2. analyze: per event: select particles, apply cuts, construct observables, fill histograms etc.
    3. finalize: normalize/scale/divide histograms, tuples, etc.
  - A minimal hook into the plugin system

- Projections: observable calculator objects that are called by the analysis' apply(event) method.
  - Refer to projection declared in init method

- Compiled via rivet-build <NAME>.cc
  - Generates a RivetAnalysis.so file

Lars Kolk

13

Rivet Analysis Code

-

-

-

Rivet Analysis Code

-

```
18    /// Book histograms and initialise projections before the run
19    void init()
20    {
21
22        // Register projection for primary particles
23        declare(ALICE::PrimaryParticles(Cuts::etaIn(ETAMIN, ETAMAX) && Cuts::abscharge > 0), "APRIM");
24
25        {Histo1DPtr tmp; _h_ppInel_neg.add(2.0, 2.5, book(tmp, 1, 1, 1));}
26        {Histo1DPtr tmp; _h_ppInel_neg.add(2.5, 3.0, book(tmp, 1, 1, 2));}
27        {Histo1DPtr tmp; _h_ppInel_neg.add(3.0, 3.5, book(tmp, 1, 1, 3));}
28        {Histo1DPtr tmp; _h_ppInel_neg.add(3.5, 4.0, book(tmp, 1, 1, 4));}
29        {Histo1DPtr tmp; _h_ppInel_neg.add(4.0, 4.5, book(tmp, 1, 1, 5));}
30        {Histo1DPtr tmp; _h_ppInel_neg.add(4.5, 4.8, book(tmp, 1, 1, 6));}
31
32        {Histo1DPtr tmp; _h_ppInel_pos.add(2.0, 2.5, book(tmp, 2, 1, 1));}
33        {Histo1DPtr tmp; _h_ppInel_pos.add(2.5, 3.0, book(tmp, 2, 1, 2));}
34        {Histo1DPtr tmp; _h_ppInel_pos.add(3.0, 3.5, book(tmp, 2, 1, 3));}
35        {Histo1DPtr tmp; _h_ppInel_pos.add(3.5, 4.0, book(tmp, 2, 1, 4));}
36        {Histo1DPtr tmp; _h_ppInel_pos.add(4.0, 4.5, book(tmp, 2, 1, 5));}
37        {Histo1DPtr tmp; _h_ppInel_pos.add(4.5, 4.8, book(tmp, 2, 1, 6));}
38    }
39
```

-

-

Rivet Analysis Code

-

```
38        }
39
40      void analyze(const Event &event)
41      {
42
43        const Particles cfs = apply<ALICE::PrimaryParticles>(event, "APRIM").particles();
44
45        for (const Particle& myp : cfs)
46        {
47          if (myp.charge() < 0)
48          {
49            _h_ppInel_neg.fill(myp.pseudorapidity(), myp.momentum().pT());
50          }
51          else
52          {
53            _h_ppInel_pos.fill(myp.pseudorapidity(), myp.momentum().pT());
54          }
55        }
56      }
57
```

-

-

  - Generates a RivetAnalysis.so file

Lars Kolk

Rivet Analysis Code

```
58      /// Normalise histograms etc., after the run
59      void finalize()
60      {
61        const double scale_factor = crossSection() / millibarn / sumOfWeights();
62        std::vector<double> binWidths = {0.5, 0.5, 0.5, 0.5, 0.5, 0.3};
63        for (size_t i = 0; i < binWidths.size(); i++)
64        {
65          _h_ppInel_neg.histos()[i]->scaleW(scale_factor / binWidths[i]);
66          _h_ppInel_pos.histos()[i]->scaleW(scale_factor / binWidths[i]);
67        }
68      }
69
70      /// @name Histogram
71      BinnedHistogram _h_ppInel_neg;
72      BinnedHistogram _h_ppInel_pos;
73
74      /// Cut constants
75      const double ETAMIN = 2.0, ETAMAX = 4.8;
76    };
77
78    RIVET_DECLARE_PLUGIN(LHCB_2021_I1889335);
79  }
```

Running and submitting a Rivet-plugin

- You can run your compiled code via
  rivet -a <Name> <inputfile>
  - inputfile: **.hepmc** file
  - Style: **.plot** file
  - Rivet provides example files to test on
  - Alternative: Let your favourite generator generate some events for you!
  - Produces **Rivet.yoda** file

- Plotting: rivet-mkhtml --errs Rivet.yoda

- Outputs look sane → Internal Review /MR in RIVET

Negative charge particle multiplicity $4.0 \leq \eta < 4.5$
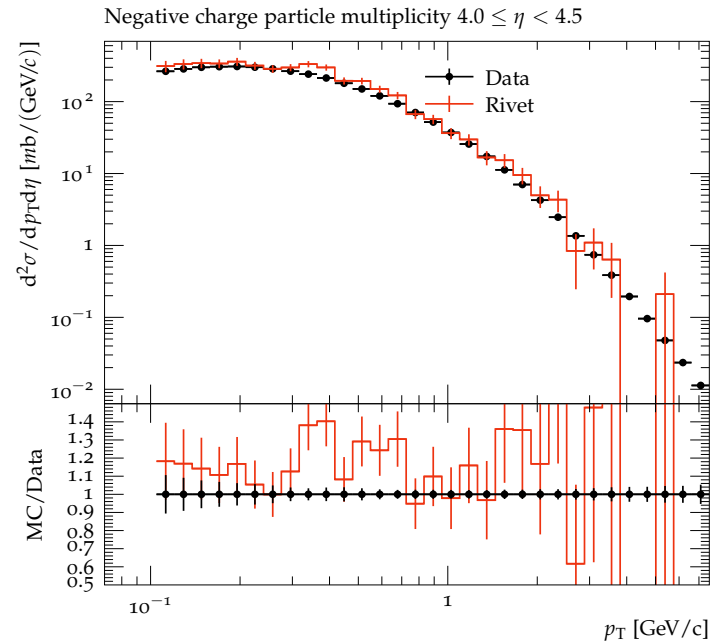
10^3 MB events - LHC-13-Minbias.hepmc.gz

Lars Kolk

# Running and submitting a Rivet-plugin

- You can run your compiled code via
  rivet -a <Name> <inputfile>
  - inputfile: **.hepmc** file
  - Style: **.plot** file
  - Rivet provides <u>example files</u> to test on
  - Alternative: Let your favourite generator generate some events for you!
  - Produces **Rivet.yoda** file

- Plotting: rivet-mkhtml --errs Rivet.yoda
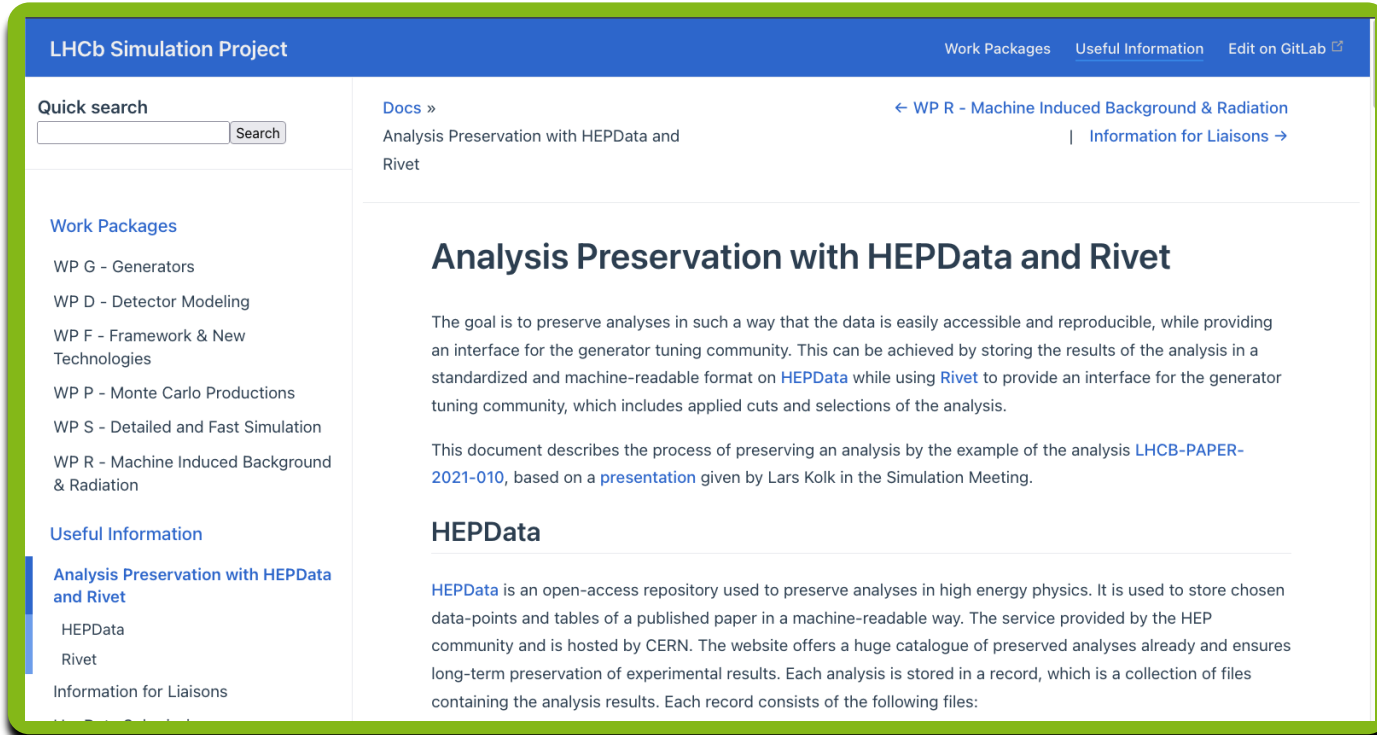
- Outputs look sane → Internal Review /MR in <u>RIVET</u>

Negative charge particle multiplicity $4.0 \leq \eta < 4.5$
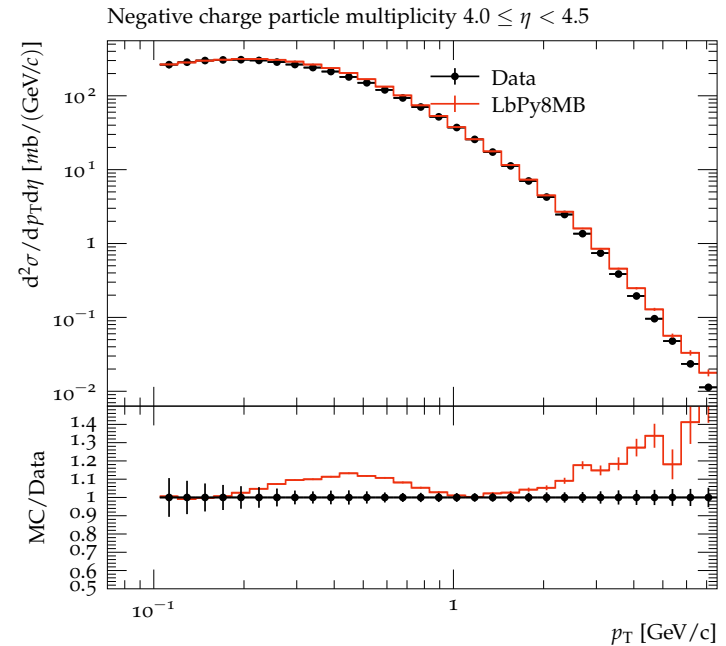
10^6 MB events - Gauss v56r4
provided by A. Grecu

Lars Kolk

Additional Remarks

Lars Kolk

Analysis Preservation Guide

- Written Analysis Preservation Guide, including all important steps

Lars Kolk

Connection to SFB1491

- F3 & F4 Deliverables: "Selected LHCb analyses will have been adopted for the use with Rivet"
- Provide plugins for generator tuning in (astro)-particle physics

- Several rivet plugins written by Chloé, e.g.
  - NA61SHINE_2019_I1754136
  - HARP_2008_I778842
  - HARPCDP_2010_I863735
  - LHCB_2011_I891233
- Pion/Kaon/Proton projectile against carbon target in forward rapidity regions

Lars Kolk

Summary

- Archiving published analyses for public access
    - Promotes transparency and reproducibility
    - Increases impact of analyses
- Analyses can be archived …
    - Data: HEPData
    - Code: Rivet

- Preservation procedure shown by example LHCB_2021_I1889335
- Written guide to help with analysis preservation
- Rivet plugins are important deliverables for the SFB

- Rivet plugins can be used by **anybody's** generators based off **your** results!

Lars Kolk