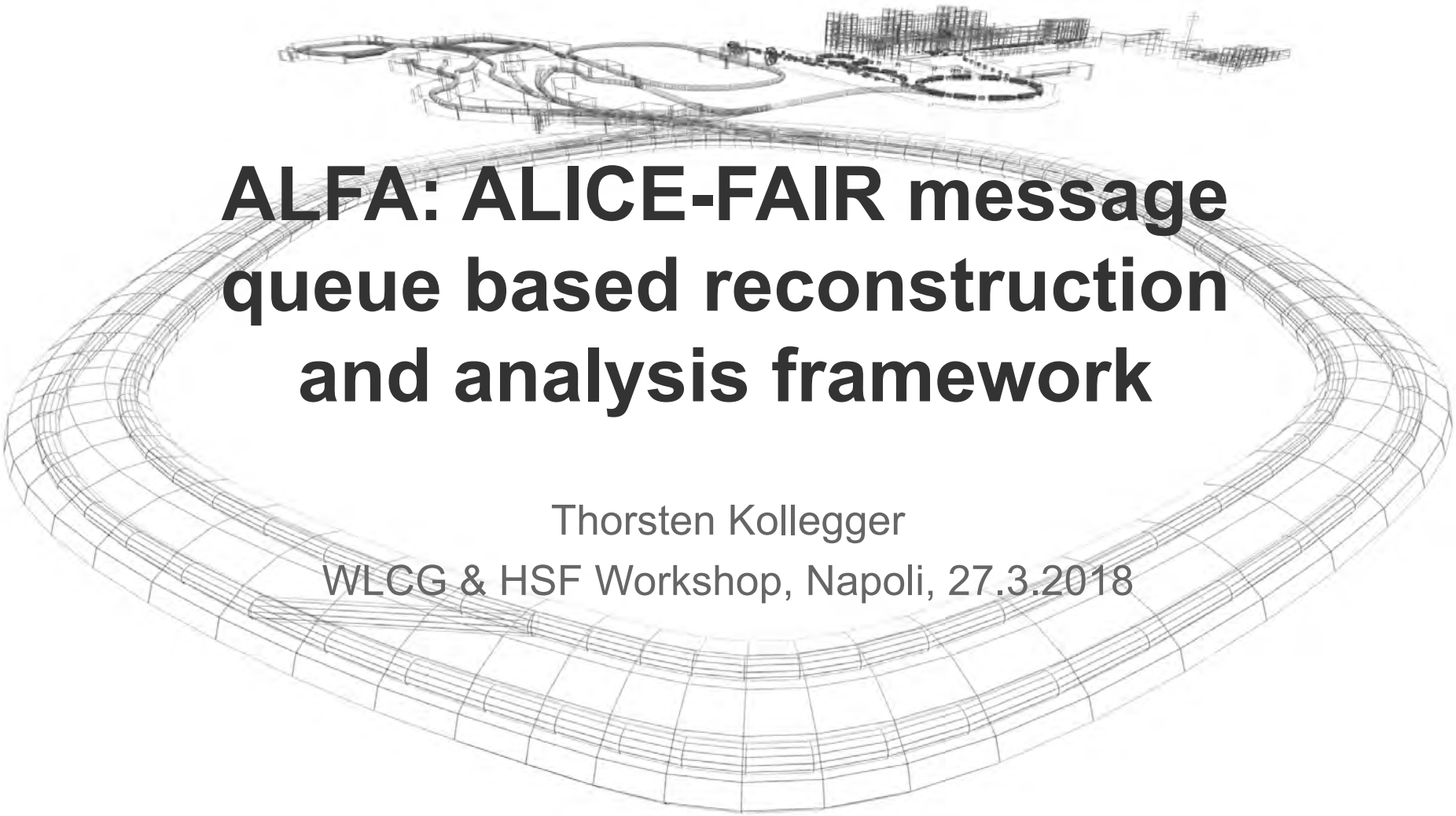
GSI Helmholtzzentrum für Schwerionenforschung GmbH



ALFA: ALICE-FAIR message queue based reconstruction and analysis framework

Thorsten Kollegger

WLCG & HSF Workshop, Napoli, 27.3.2018

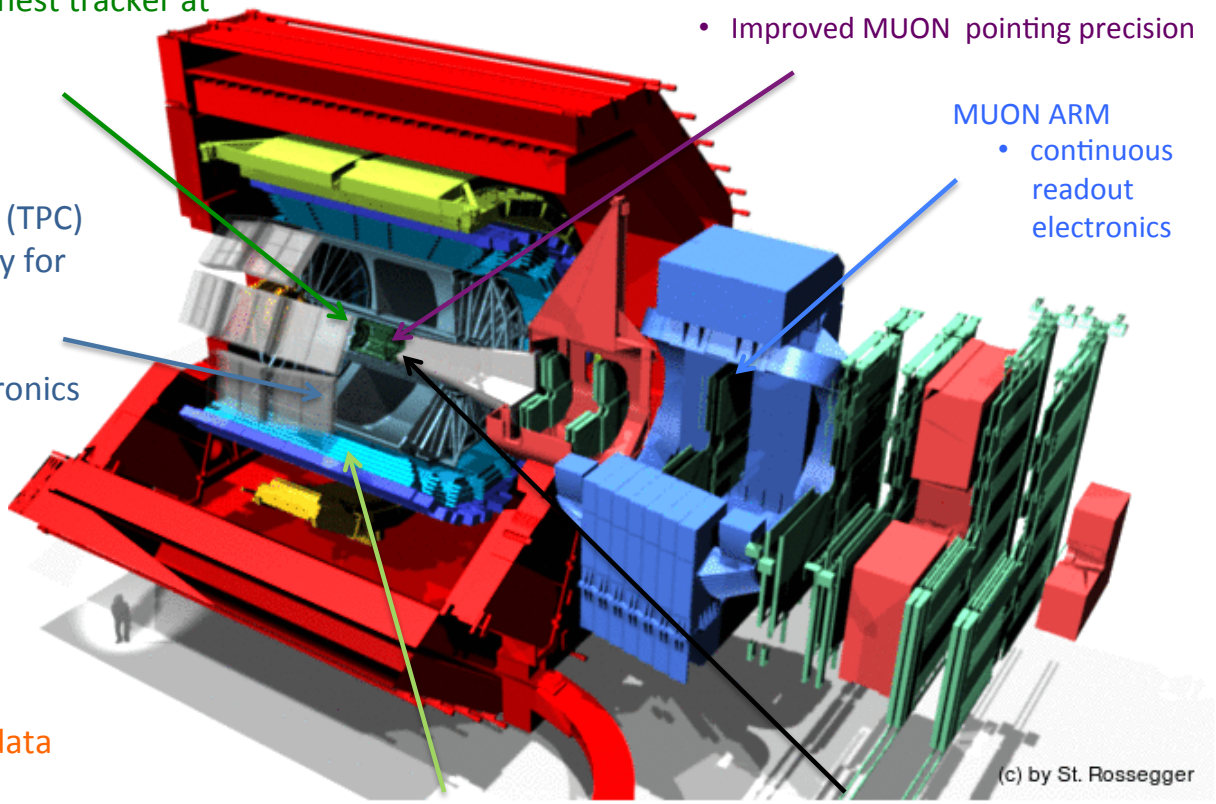
ALICE LS2 Upgrade

New Inner Tracking System (ITS)
- improved pointing precision
- less material -> thinnest tracker at the LHC

Muon Forward Tracker (MFT)
- new Si tracker
- Improved MUON pointing precision

MUON ARM
- continuous readout electronics

Time Projection Chamber (TPC)
- new GEM technology for readout chambers
- continuous readout
- faster readout electronics



Data Acquisition (DAQ)/
High Level Trigger (HLT)
- new architecture
- on line tracking & data compression
- 50kHz Pbb event rate

TOF, TRD
- Faster readout

New Trigger Detectors (FIT)

O2 System from the Letter of Intent

Design Guidelines

- Handle >3 TByte/s detector input
- Produce (timely) physics result

} Online Reconstruction to reduce data volume
Output of System AODs

Minimize “risk” for physics results

- Allow for reconstruction with improved calibration, e.g. store clusters associated to tracks instead of tracks
- Minimize dependence on initial calibration accuracy

Keep cost “reasonable”

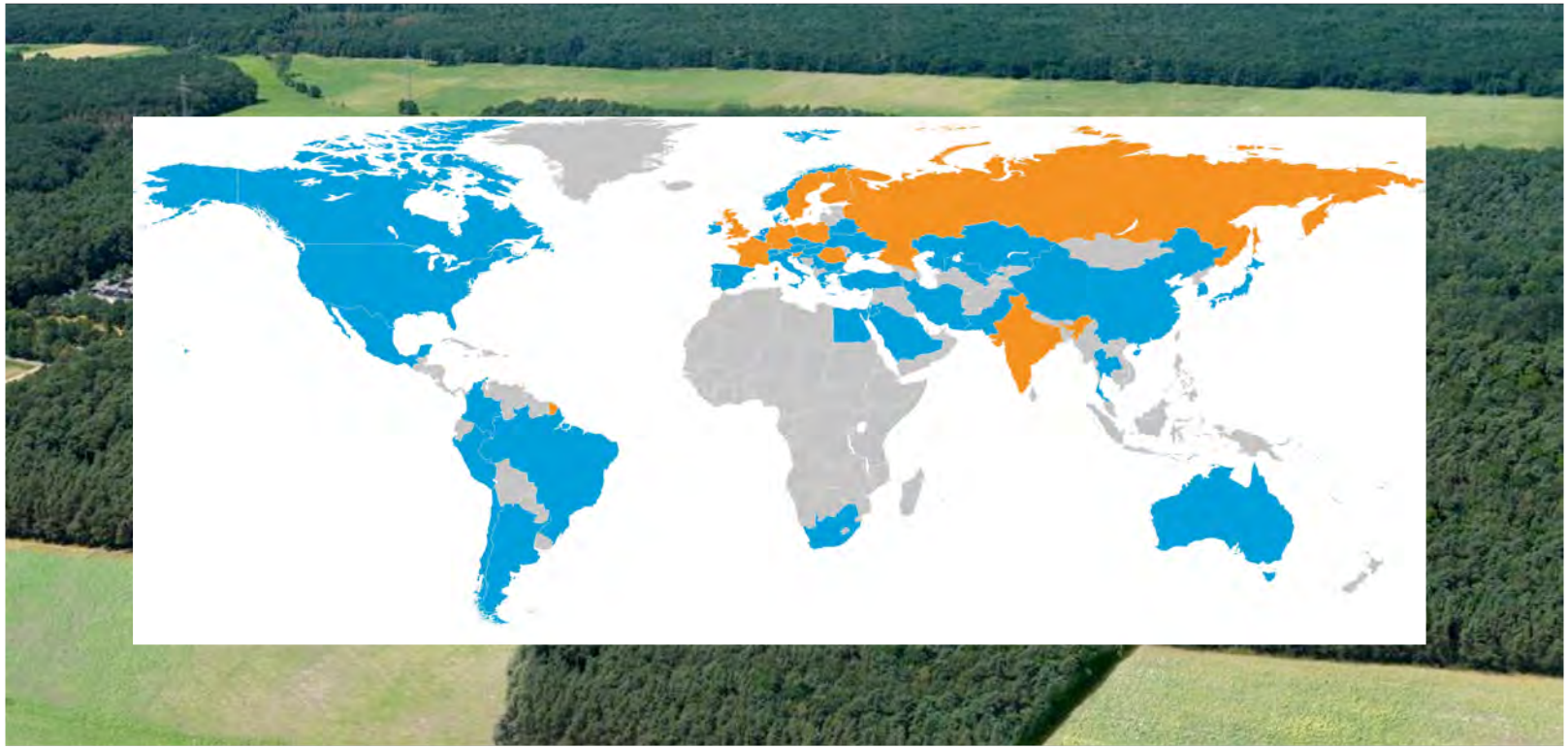
- Limit storage system bandwidth to ~80 GB/s peak and 20 GByte/s average
- Optimal usage of compute nodes

FAIR Project

New International Accelerator Facility in Darmstadt, Germany

- Budget: 1.3 Billion Euros (2005)
- Upgraded accelerators from GSI as injectors
- 9 Member States + worldwide Partners

FAIR Project

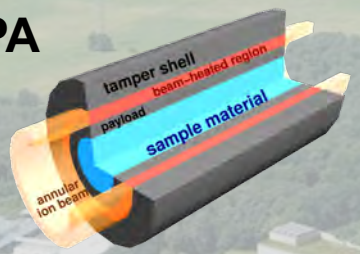


New International Accelerator Facility in Darmstadt, Germany

- Budget: 1.3 Billion Euros (2005)
- Upgraded accelerators from GSI as injectors
- 9 Member States + worldwide Partners

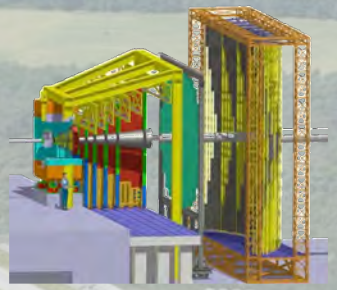
FAIR Scientific Pillars

APPA



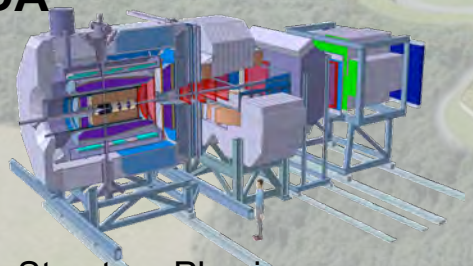
Atomic, Plasma Physics and Applications

CBM



Compressed Baryonic Matter

PANDA

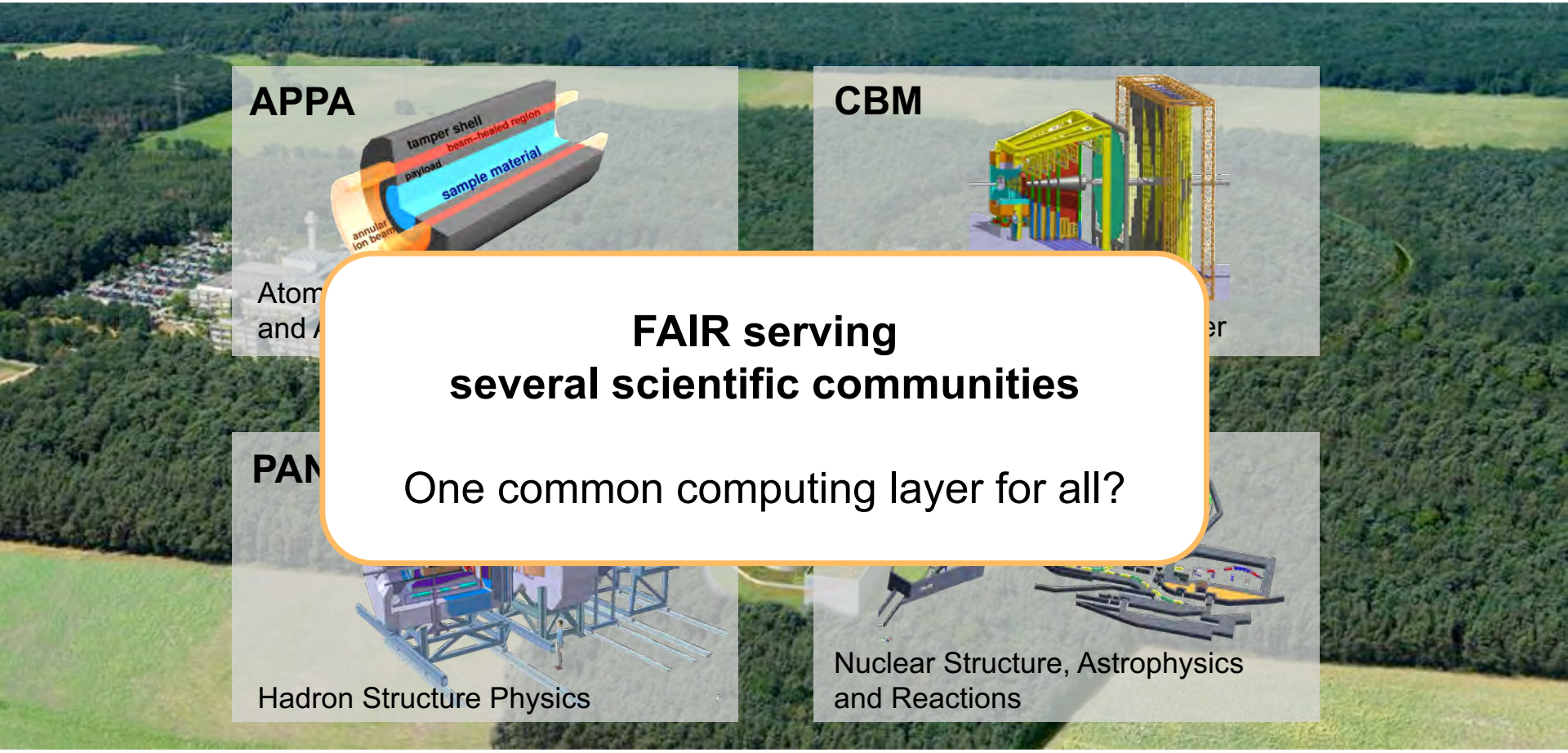


Hadron Structure Physics

NUSTAR

Nuclear Structure, Astrophysics and Reactions

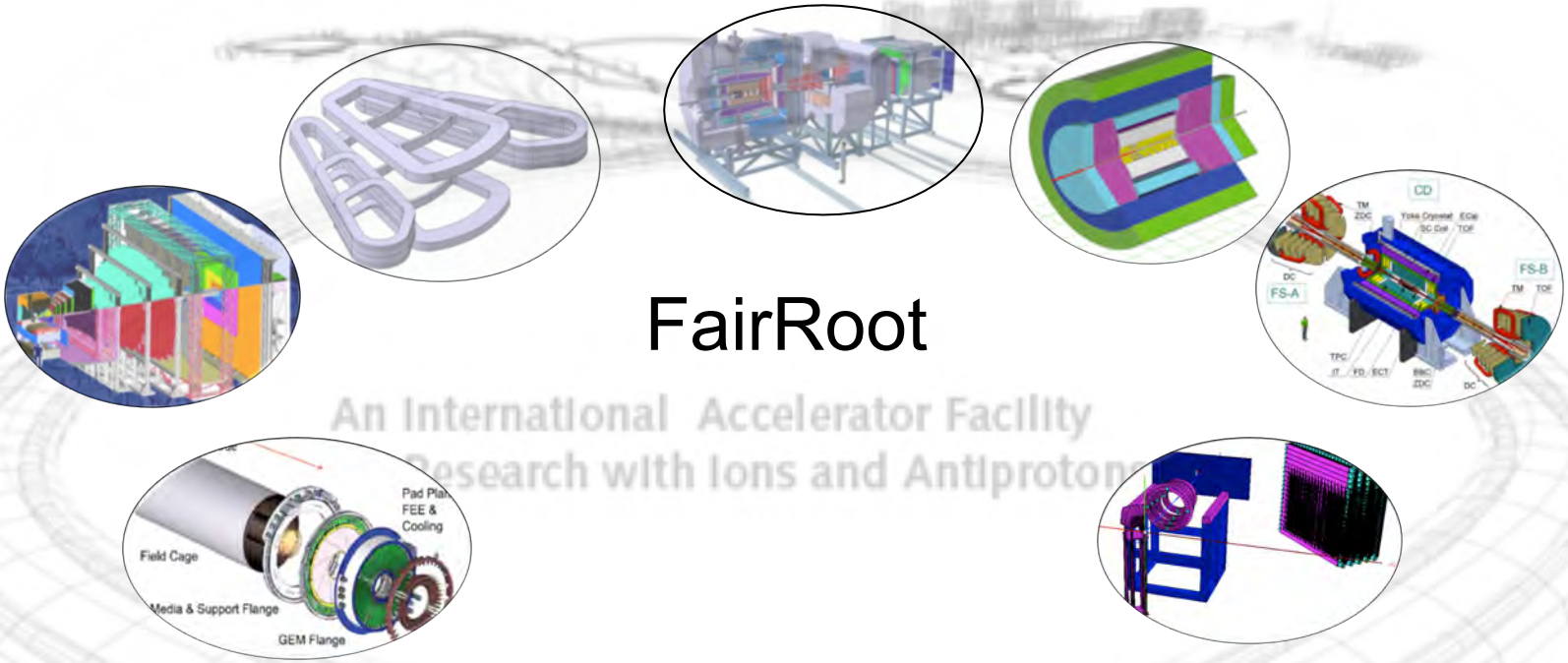
FAIR Scientific Pillars



FAIR serving several scientific communities

One common computing layer for all?

FairRoot



Common simulation and reconstruction software framework for the FAIR experiments (and beyond)

https://github.com/FairRootGroup/FairRoot

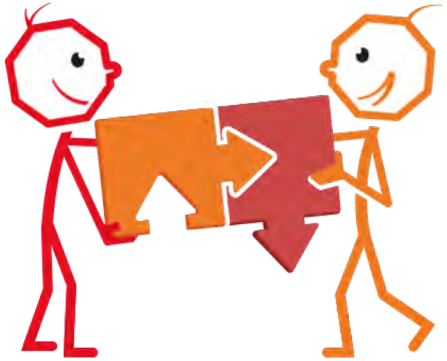
ALICE and FAIR

Two projects – same requirements

Massive data volume reduction (1 TByte/s input)

- Data reduction by (partial) online reconstruction
- Online reconstruction and event selection

Much tighter coupling between online and offline reconstruction software



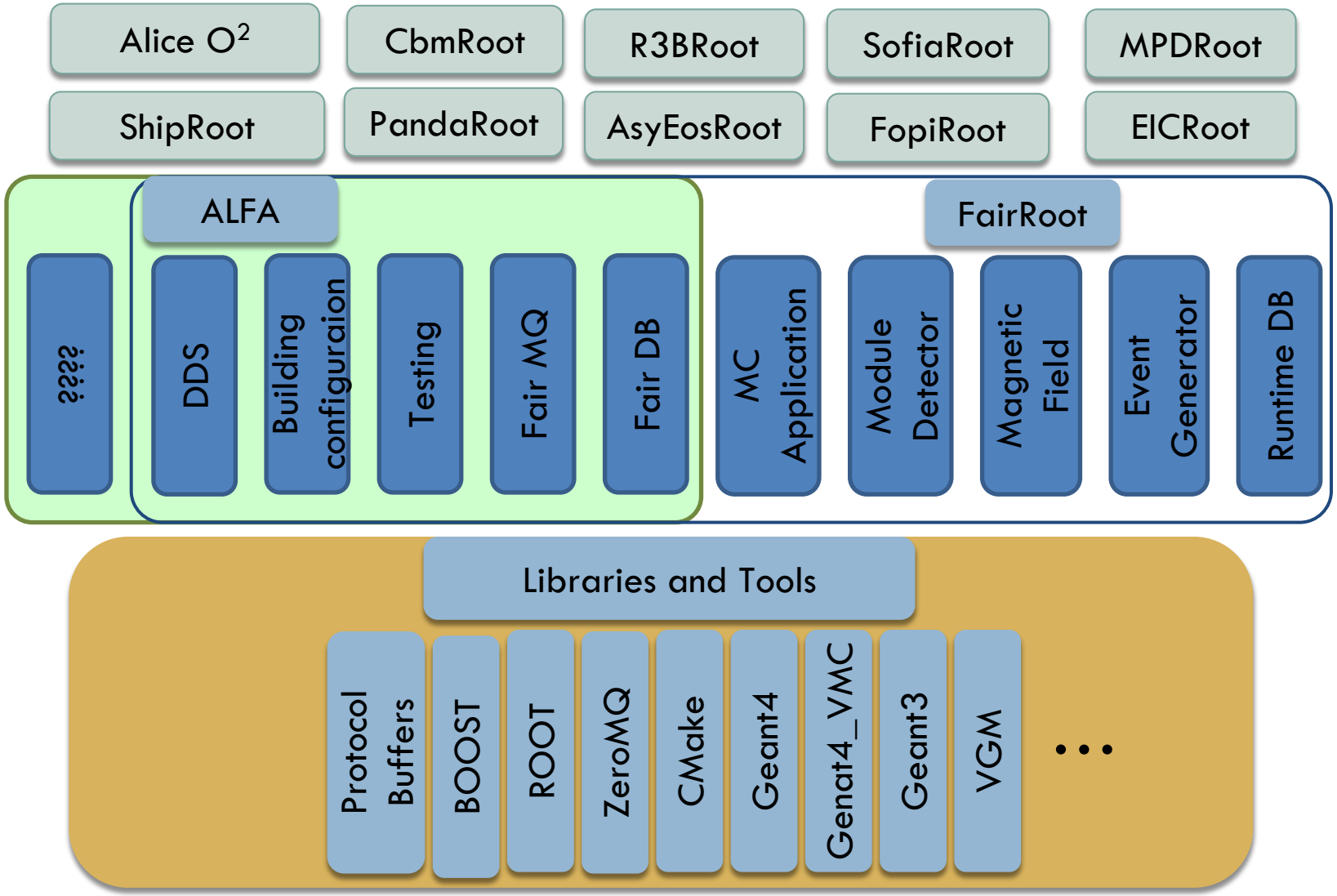
Lets work together

Common ALICE-FAIR Software Framework

A data-flow based model
(Message Queues based multi-processing)

- Transport layer (FairMQ, based on: ZeroMQ, nanomsg, shared memory, RDMA)
- Configuration tools
- Management and monitoring tools
- Unified access to configuration parameters and databases.

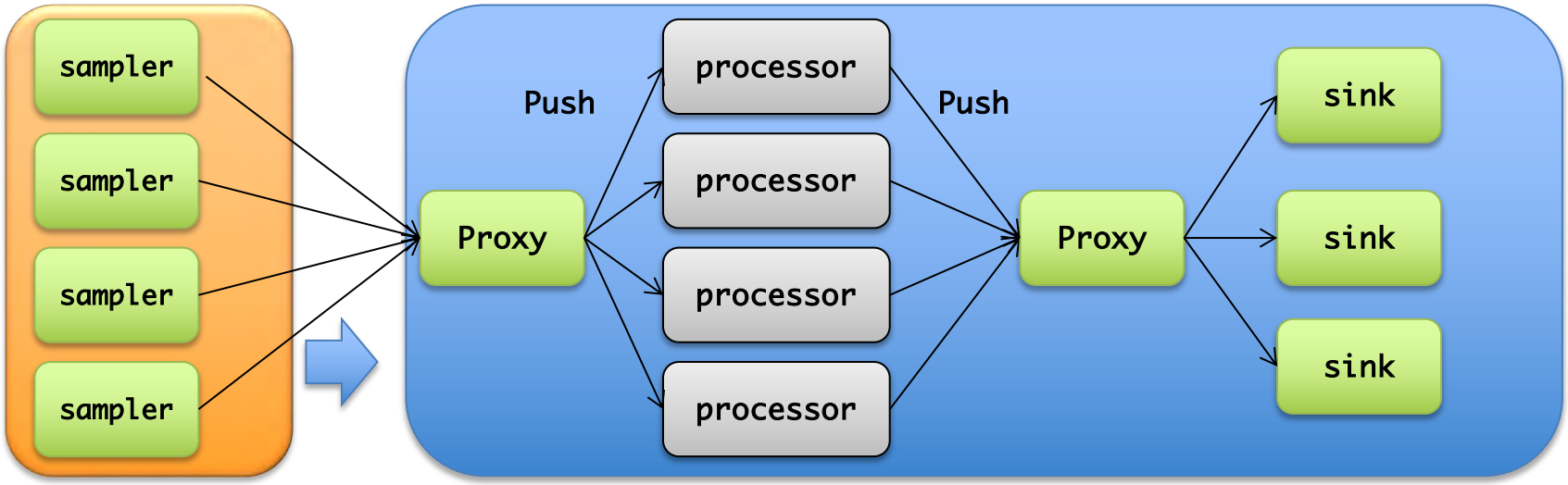
FairRoot & ALFA



FairMQ

The Data Processing Component of ALFA

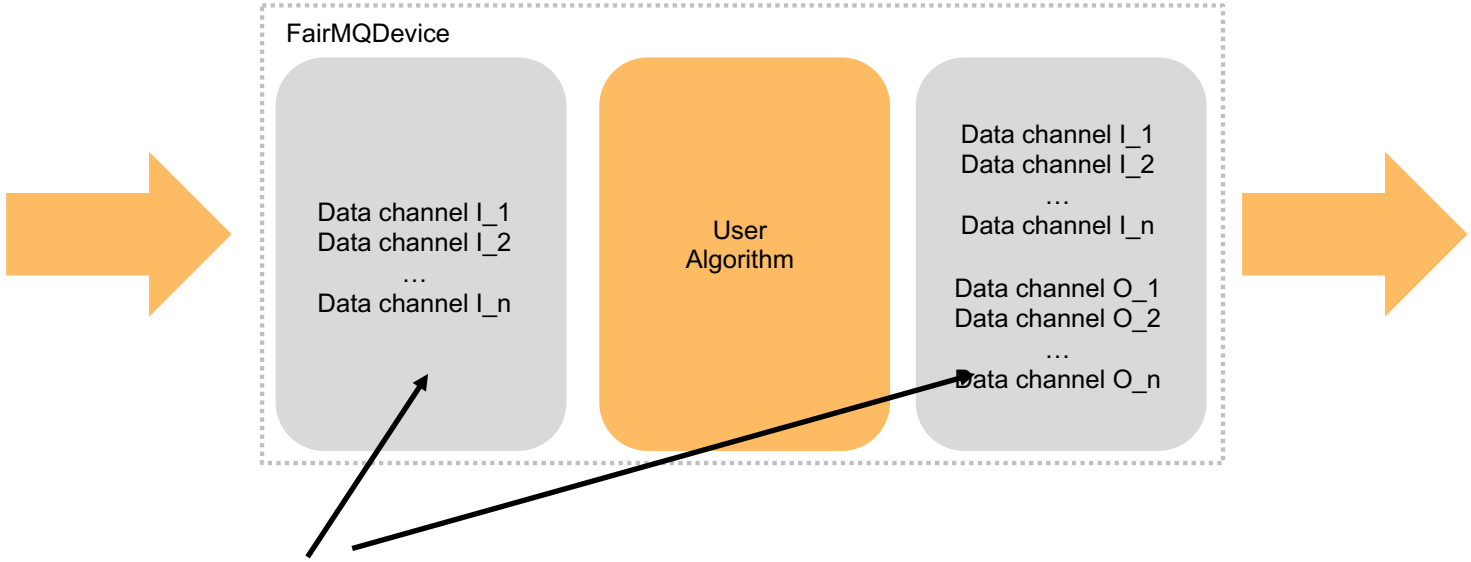
- Multi-process concept (specialized **devices**)
- Data-flow model: Message queues for data exchange,
  technology agnostic



Design Goals

- Scalability, Maintainability, Reliability
  - efficient use of multi-core architectures
- Reusable with common data processing components
  - Reduce cost of new developments, agile development

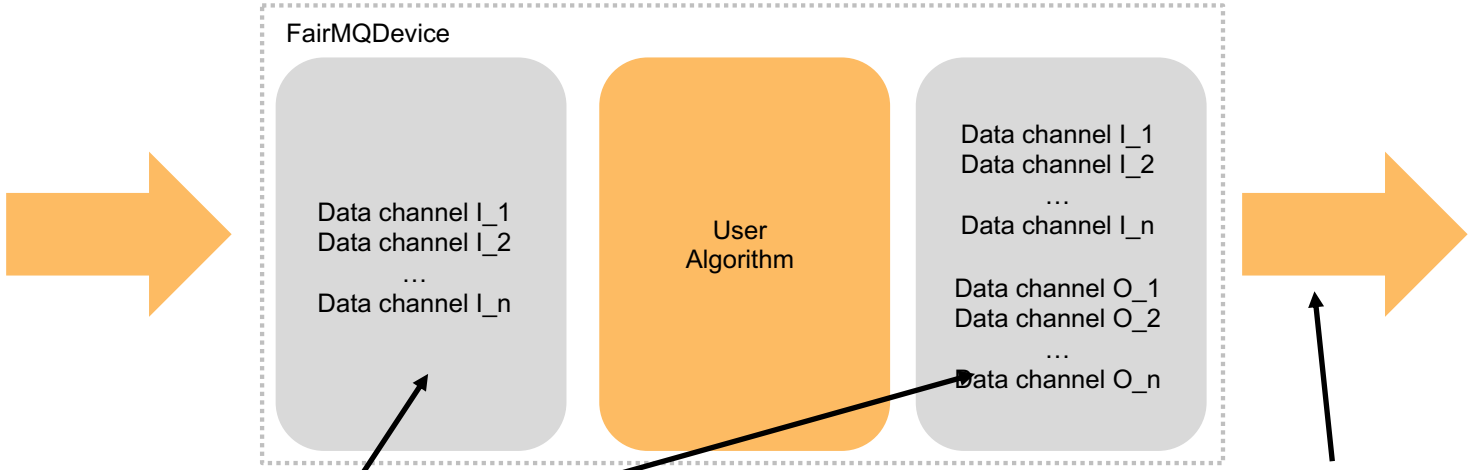
FairMQ Devices



Input and Output designed as Message Queues

- Device takes/passes ownership of data, no central data management unit

FairMQ Devices



Input and Output designed as Message Queues

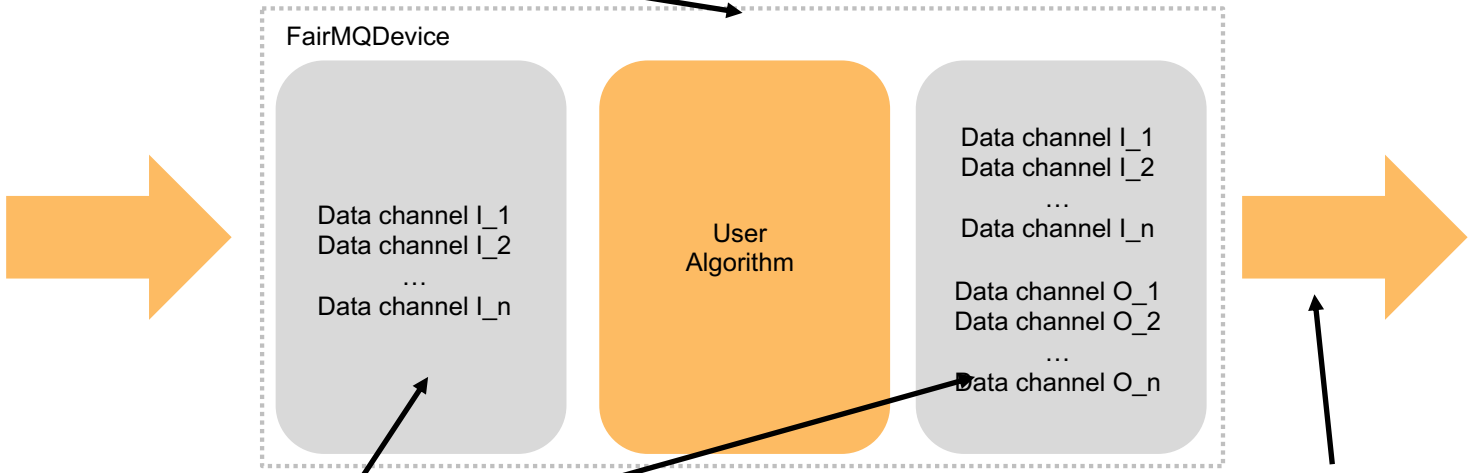
- Device takes/passes ownership of data, no central data management unit

Several message transports and serialization protocols supported

- 0MQ, nanomsg, shared memory, RDMA
- ROOT, protobuf, boost, …

FairMQ Devices

User Algorithm executed when necessary input data from all channels available



Input and Output designed as Message Queues

- Device takes/passes ownership of data, no central data management unit

Several message transports and serialization protocols supported

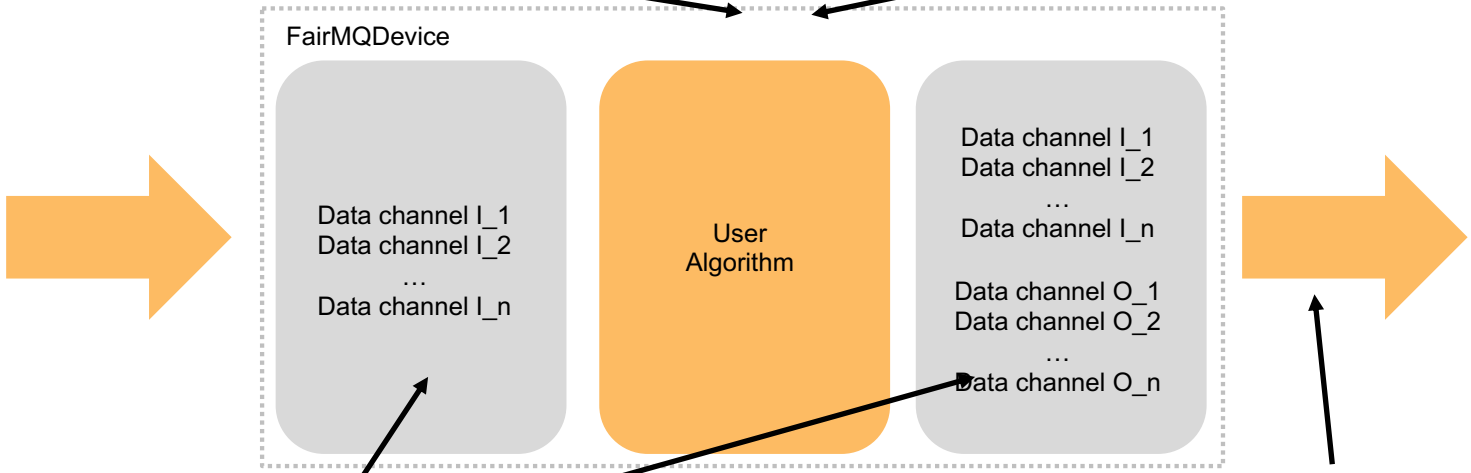
- 0MQ, nanomsg, shared memory, RDMA
- ROOT, protobuf, boost, …

FairMQ Devices

User Algorithm executed when necessary input data from all channels available

Each ”Device" is a separate process, which:

- Can be multithreaded, SIMDized, …
- Runs on different hardware (CPU, GPU, …)
- Be written in an any supported language



Input and Output designed as Message Queues

- Device takes/passes ownership of data, no central data management unit

Several message transports and serialization protocols supported

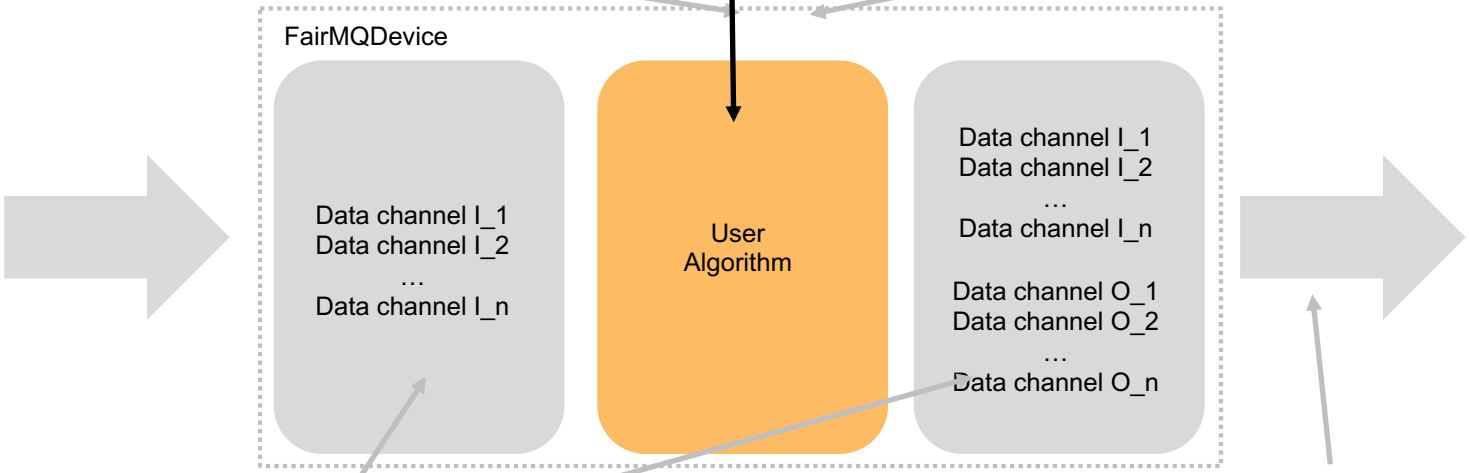
- 0MQ, nanomsg, shared memory, RDMA
- ROOT, protobuf, boost, …

FairMQ Devices

User Algorithm execute
necessary input data
from all channels availa

separate process, which:
eaded, SIMDized, …
nt hardware (CPU, GPU, …)
- Be written in an any supported language

All the framework user sees:
Callback to his algorithm



Input and Output designed
as Message Queues
- Device takes/passes ownership of data, no central data management unit

Several message transports
and serialization protocols
supported
- 0MQ, nanomsg, shared memory, RDMA
- ROOT, protobuf, boost, …

FairMQ

Looking at the IT landscape: shift towards

- Microservices
  - Unbundled, decentralized modules
  - Organized around specific capability
- Containers
- Algorithm Economy

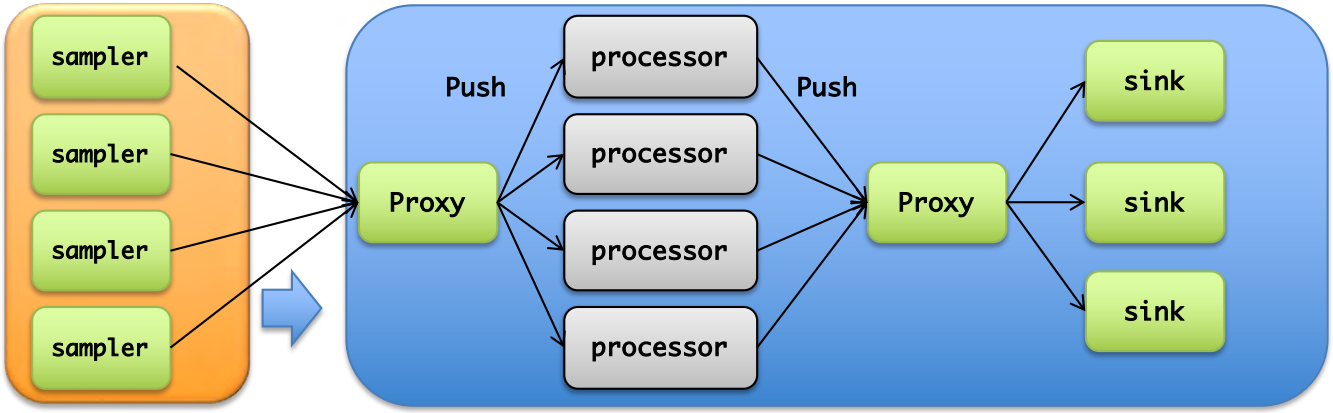
These are at the heart of the
„cloud/app“ business model/economy

- driven by scalability and reliability demands
- based on multi-process and message exchange
- development cost advantage

FairMQ uses many of these technologies under the hood;
replacing custom code (e.g. ALICE HLT framework)

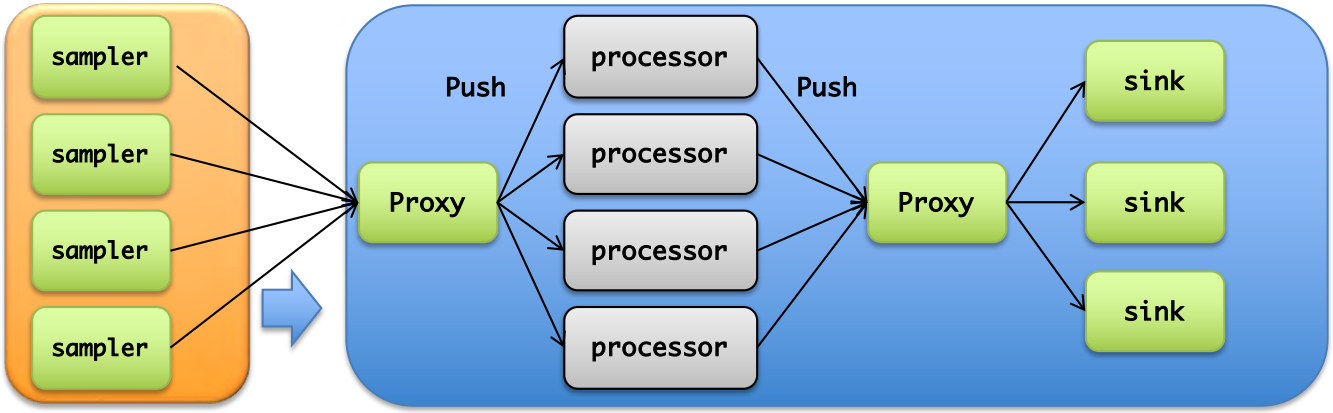
DDS

Complex processing topologies
can be build by connecting devices, scalability by
running multiple instances



DDS

Complex processing topologies
can be build by connecting devices, scalability by
running multiple instances

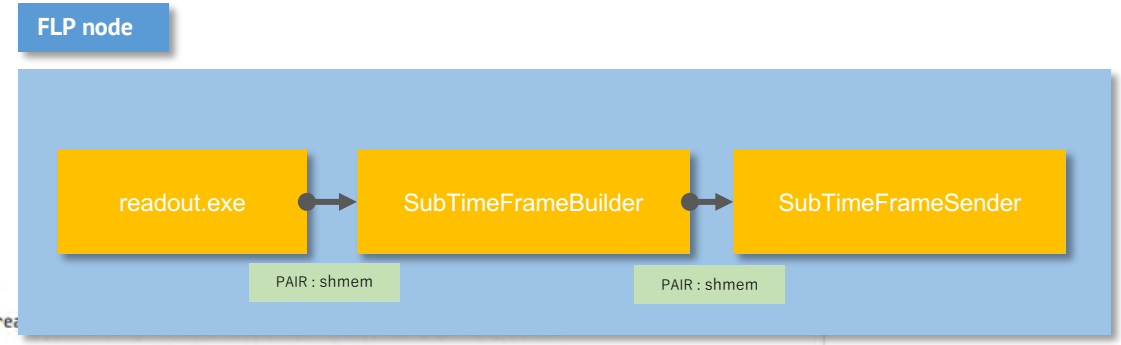


Topology needs to be described,
processes deployed and controlled:

DDS (Dynamic Deployment System),
another module in the ALFA Framework

DDS – Processing Topology

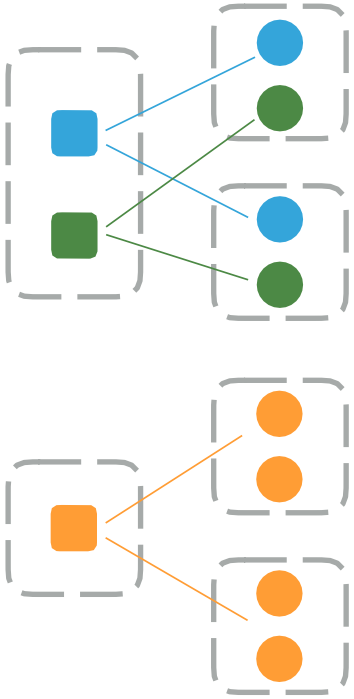
Processing Topology (i.e. data flow) described by XML files, composition of devices, collections of devices, …



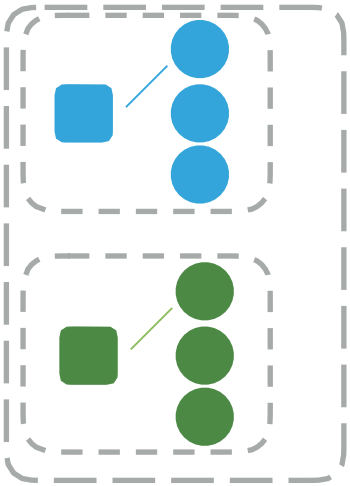

```
16    <decltask id="Readout"> <!-- Readout -->
17        <exe reachable="true">$INFOLOGGER_ROOT/bin/rea
18    </decltask>

20    <decltask id="SubTimeFrameBuilder"> <!-- SubTimeFrameBuilder to run with Readout -->
21        <exe reachable="true">$O2_ROOT/bin/SubTimeFrameBuilderDevice --id stf_builder_%taskIndex% --session ${session} --transport shmem --
22        <properties>
23            <id access="read">builder-stf-channel</id>
24        </properties>
25    </decltask>

42    <decltask id="SubTimeFrameSender">
43        <exe reachable="true">$O2_ROOT/bin/SubTimeFrameSenderDevice --id stf_sender_%taskIndex% --session ${session} --transport shmem --sh
44        <properties>
45            <id access="write">builder-stf-channel</id>
46            <id access="write">sender-stf-channel</id>
47        </properties>
48    </decltask>
```

DDS

Classic deployment
Central DDS commanders (can share the same host), remote agents.



Batch deployment
Central DDS commanders on the same host with their agents. DDS runs in a batch mode. No UI connection is foreseen.



Same tool/topology description for online and offline/Grid

Examples

- PANDA MVD readout (Tobias Stockmanns)
- Porting of ALICE HLT Run1/2 GPU tracker (David Rohr, Ruben Shahoyan)

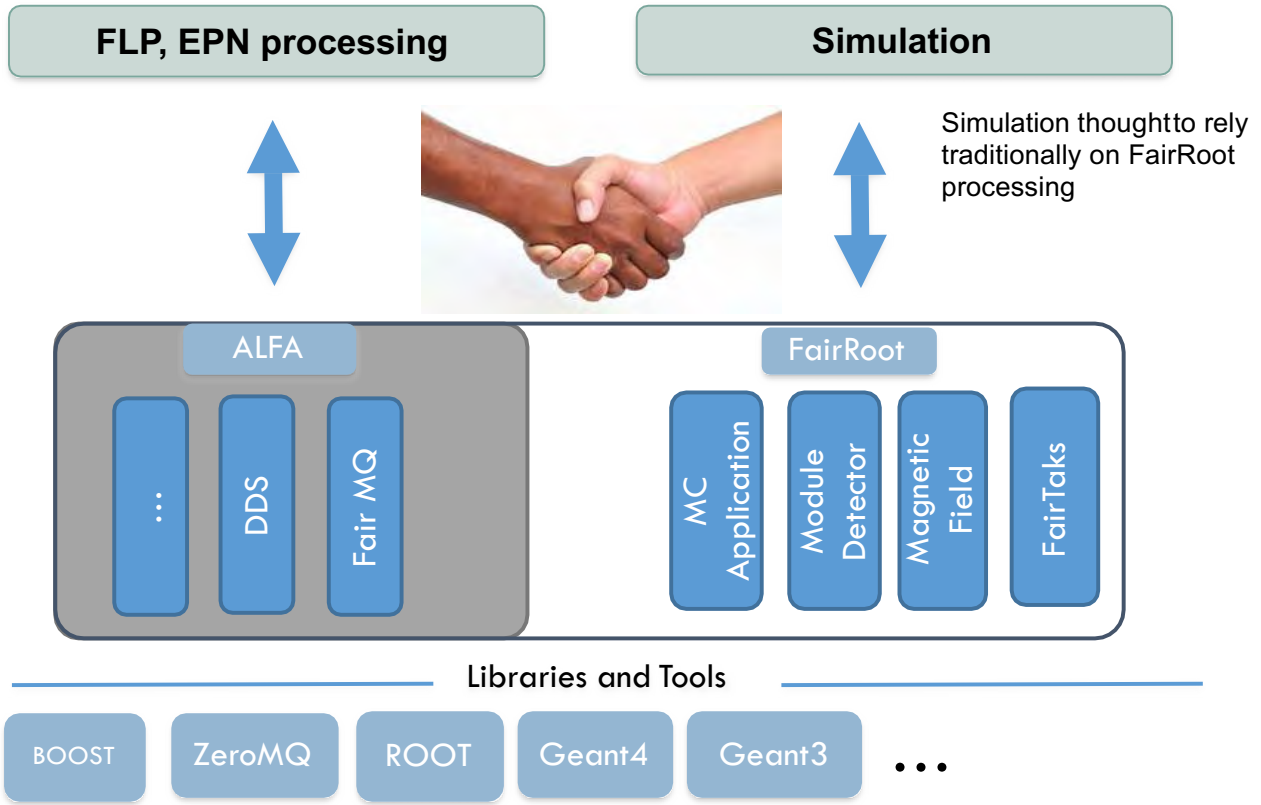
Much more interesting ideas in the pipeline, wait for CHEP2019

- CBM/MiniCBM FLES (Florian Uhlig)
- ALICE O2 Data Processing Layer, ALICE Data Model on top of ALFA (Giulio Eulisse)
- ALICE O2 parallelized simulation (Sandro Wenzel)

ALICE Parallel Simulation

A Large Ion Collider Experiment

A new handshake between FairRoot and ALFA

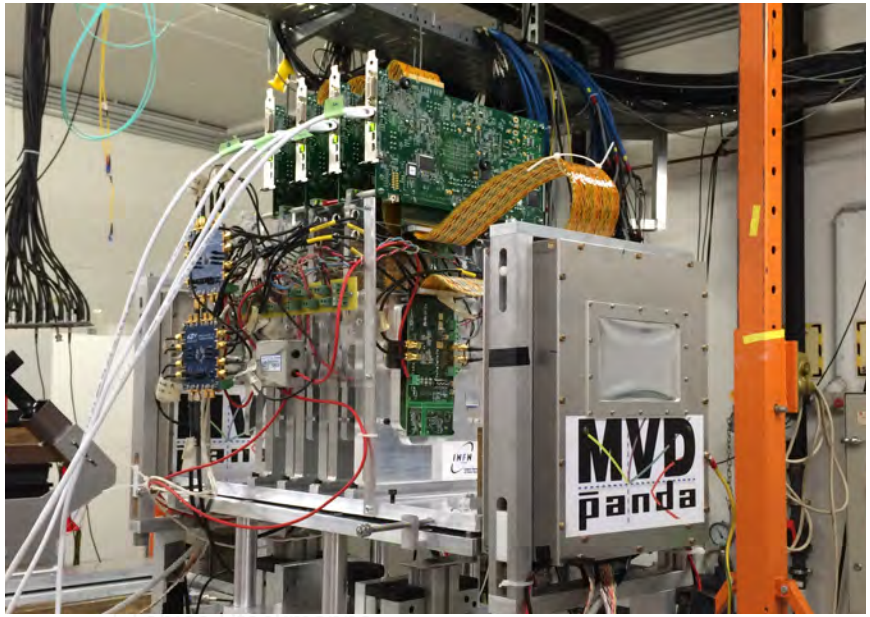


Sandro Wenzel

Adapted from M. Turany @ CHEP15

ALICE OFFLINE WEEK | March 15th, 2018 | Sandro Wenzel

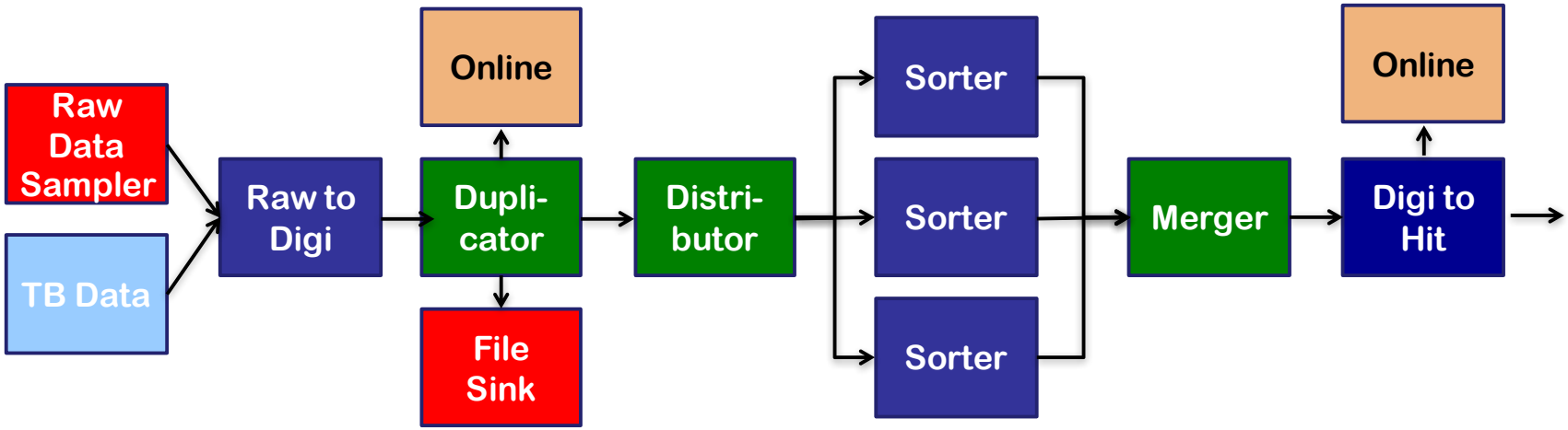
Test of FairMQ with Real Data

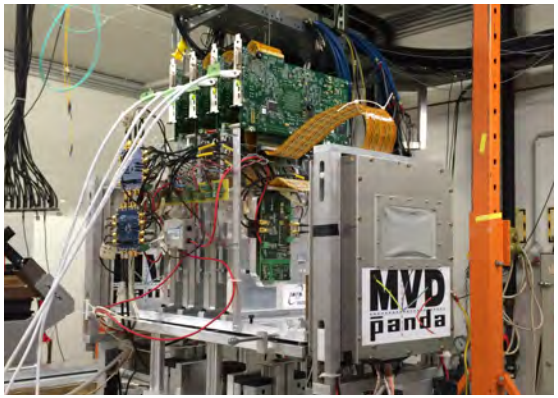


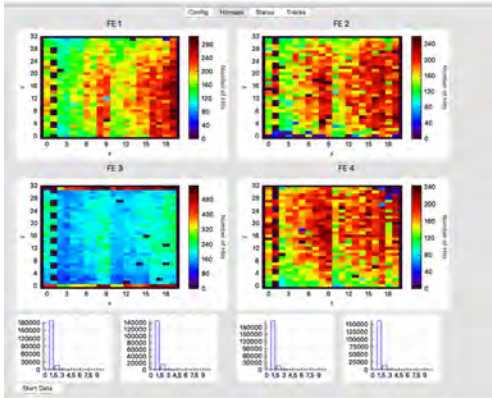
- Parallel readout of 4 pixel detectors
- Readout done by 4 FPGA boards sending their data to two PCs
- On the PC bitstream converted into raw data
- Raw data send via FairMQ to a FileSink

Tobias Stockmanns
https://indico.cern.ch/event/505613/contributions/2227258/

FairMQ

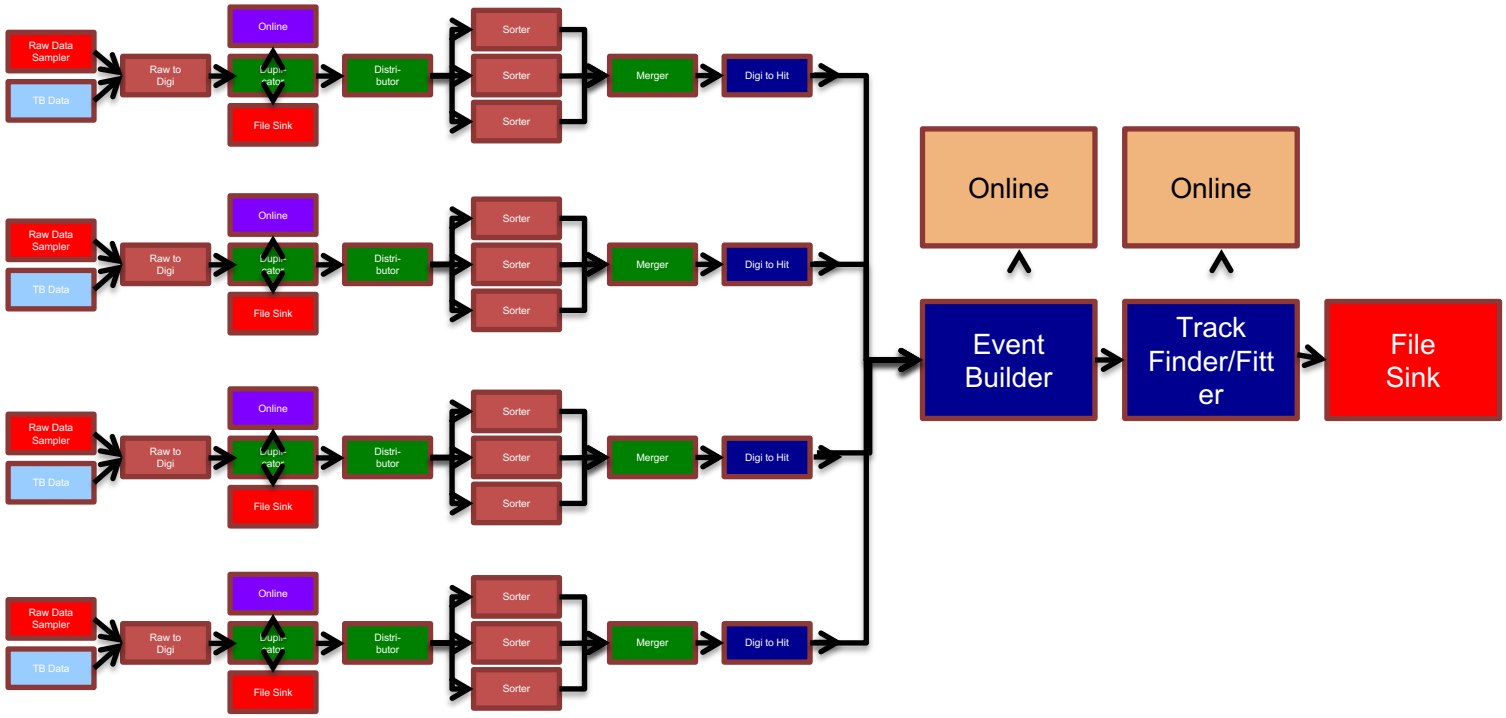






Tobias Stockmanns
https://indico.cern.ch/event/505613/contributions/2227258/

Multiple Front-Ends



Tobias Stockmanns
https://indico.cern.ch/event/505613/contributions/2227258/

ALICE Run3 Tracking

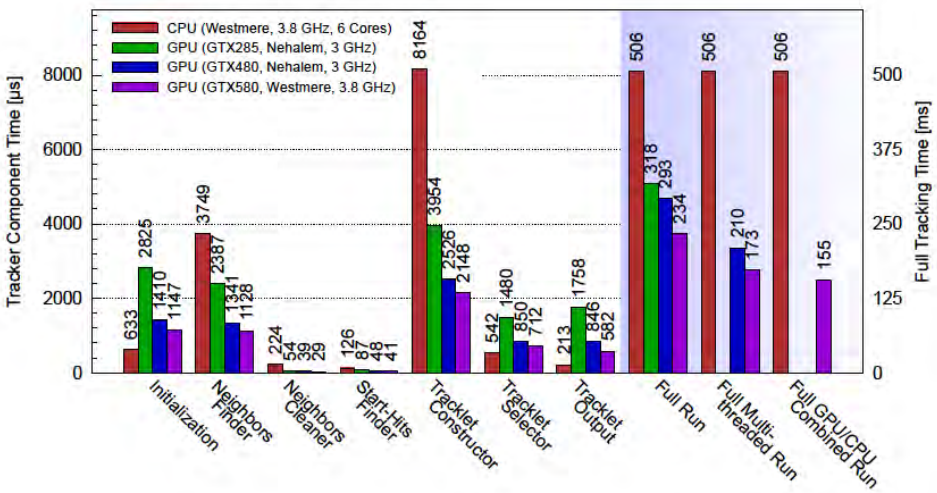
GPU based CA tracking used in Run 1 + 2 in ALICE HLT

A Large Ion Collider Experiment

Tracking Performance

HLT uses GPUs for TPC tracking

- Unique at LHC, other experiments following now with R&D
- Factor 4 in total tracking time: factor 3 less nodes in system



David Rohr

HLT | Offline Week | 12.06.2013 | Thorsten Kollegger 16

ALICE Run3 Tracking

GPU based CA tracking used in Run 1 + 2 in ALICE HLT

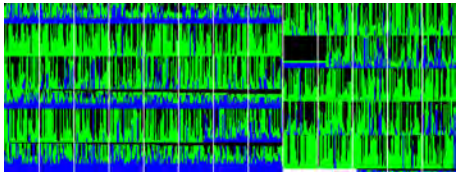
ALICE HLT GPU Experience

Experience quite promising, will continue/expand in Run 2

- Allowed to reduce system size by factor 3
- Stable operation even with consumer hardware

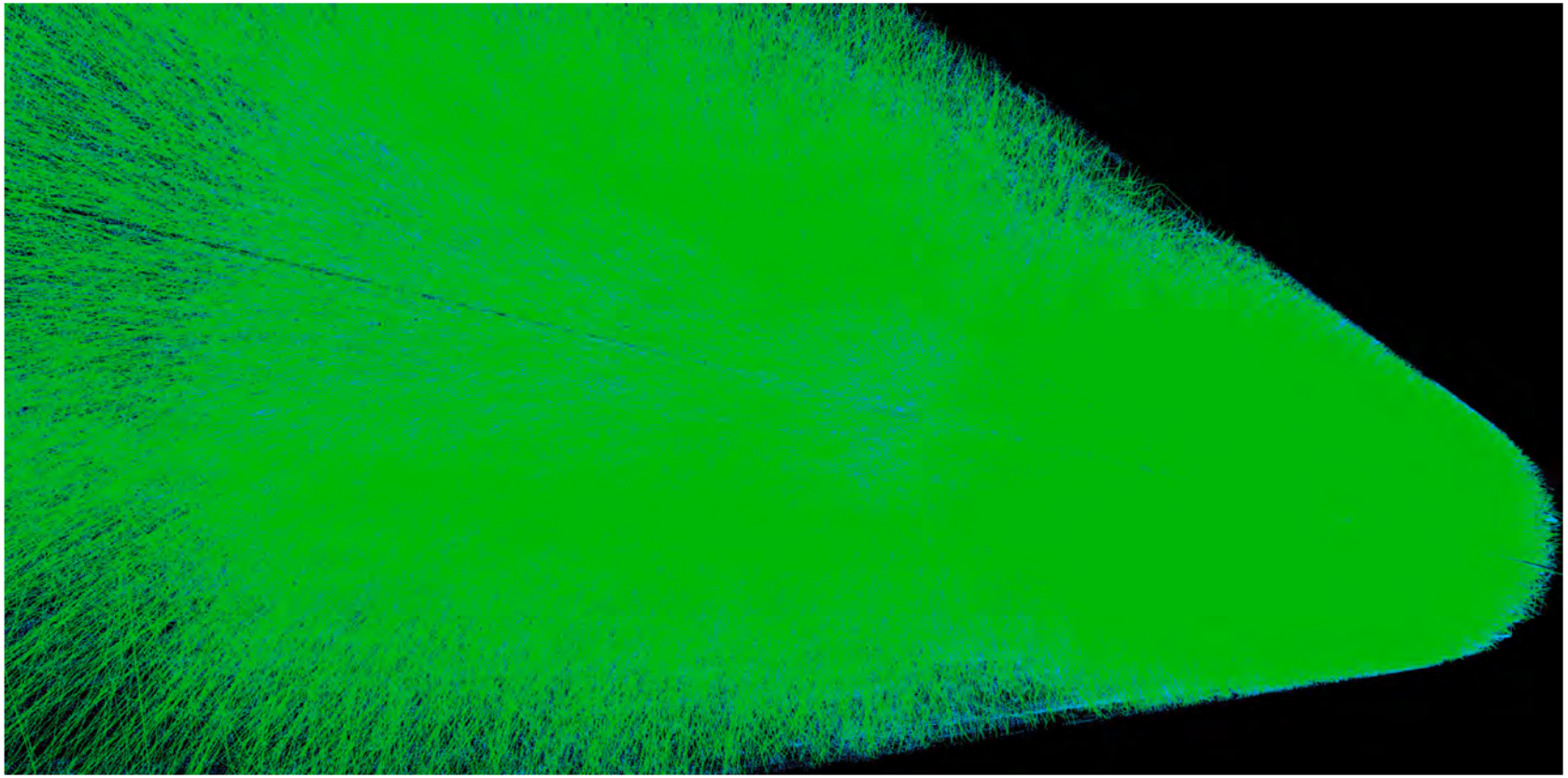
Comes with some cost

- Initial porting to CUDA, change to SP: 1.5 PhD students/1 year
- Every new GPU generation requires re-tuning (even same chip)
- Need to support two versions (CPU for simulation, GPU)
- Full loading of GPU requires quite some effort: currently at 67%



David Rohr

ALICE Run 3 TimeFrame



A 20 GigaByte continuous data frame
several hundred overlapping collisions with several thousands tracks each

ALICE and FAIR

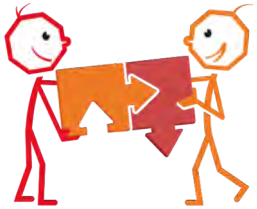
Two projects – same requirements

Massive data volume reduction (1 TByte/s input)

- Data reduction by (partial) online reconstruction
- Online reconstruction and event selection

Much tighter coupling between online and offline reconstruction software – **ALFA**

Also used by others (NICA, …)



Lets work together