

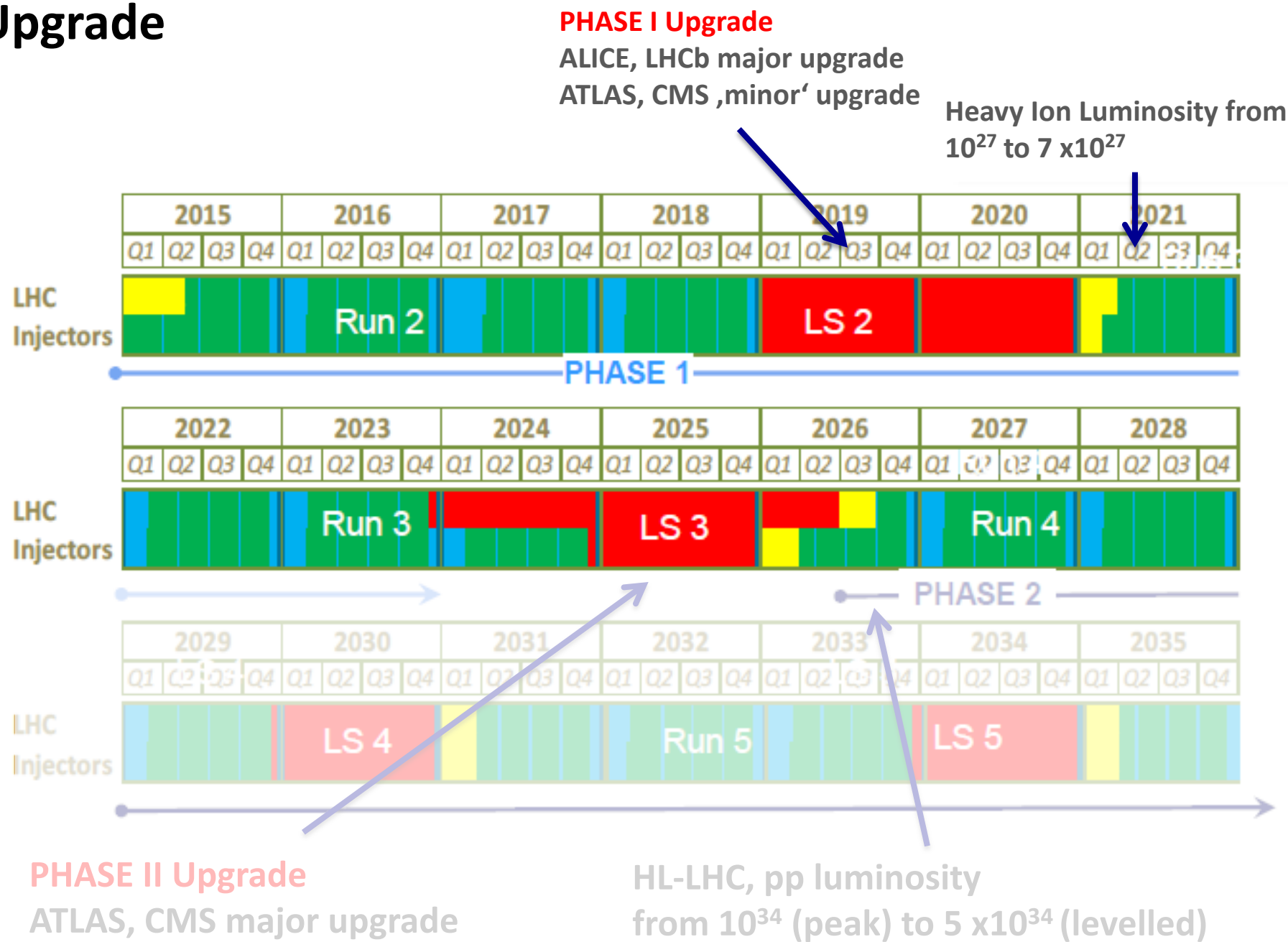
Analysis infrastructures and framework for massive parallel analysis

Mohammad Al-Turany
GSI Darmstadt

Disclaimer

I am not trying to give an unbiased talk or general overview of all existing infrastructures but more my experience and our plans in FAIR and ALICE to face the challenges in computing in the next few years!

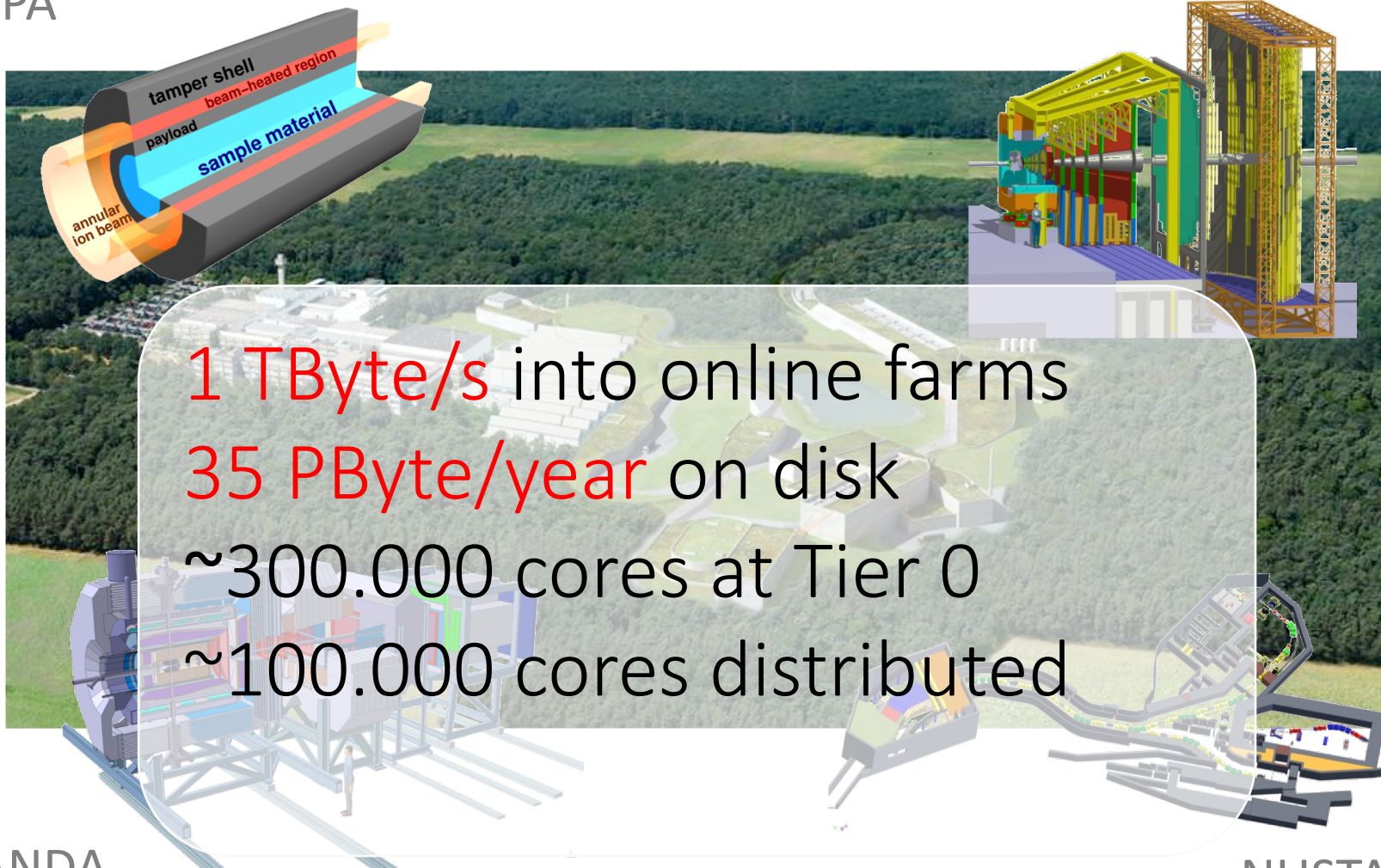
LHC Upgrade



Computing at FAIR

APPA

CBM



1 TByte/s into online farms

35 PByte/year on disk

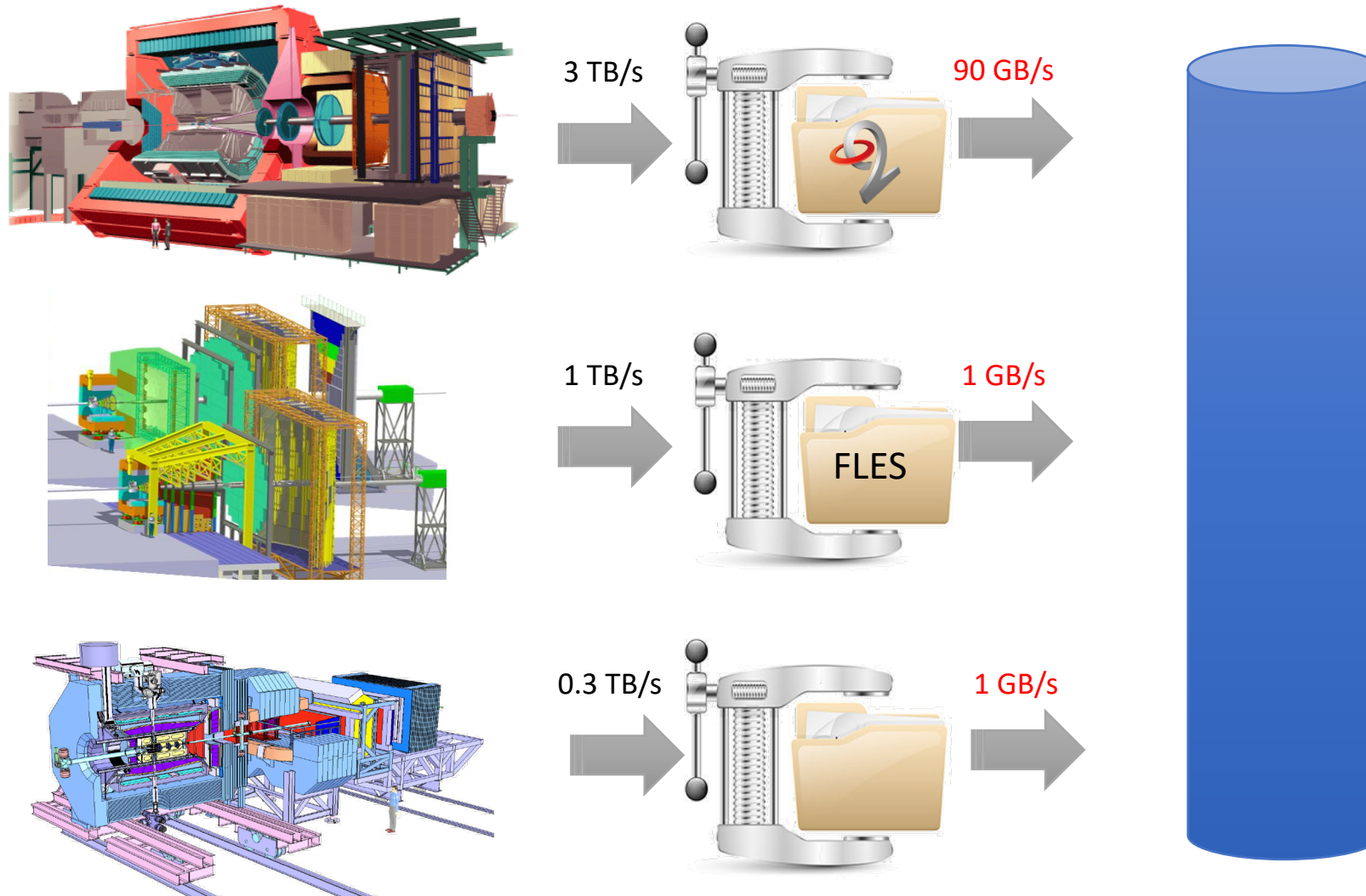
~300.000 cores at Tier 0

~100.000 cores distributed

PANDA

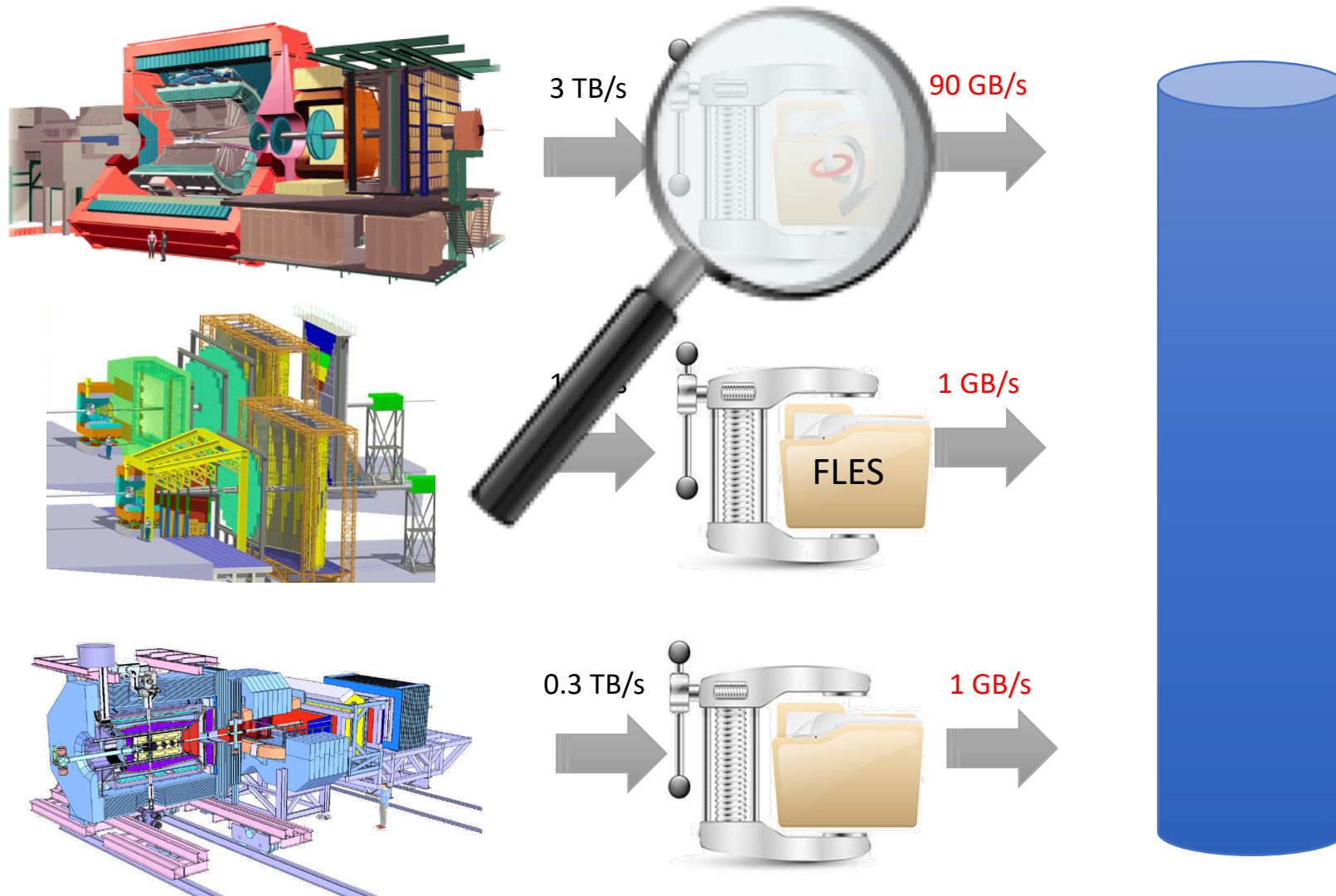
NUSTAR

Different experiment similar requirements!



Data reduction by (partial) online reconstruction

Different experiment similar requirements!



Alice O^2

+ 463 FPGAs

- Detector readout and fast cluster finder

+ 100'000 CPU cores

- To compress 1.1 TB/s data stream by overall factor 14

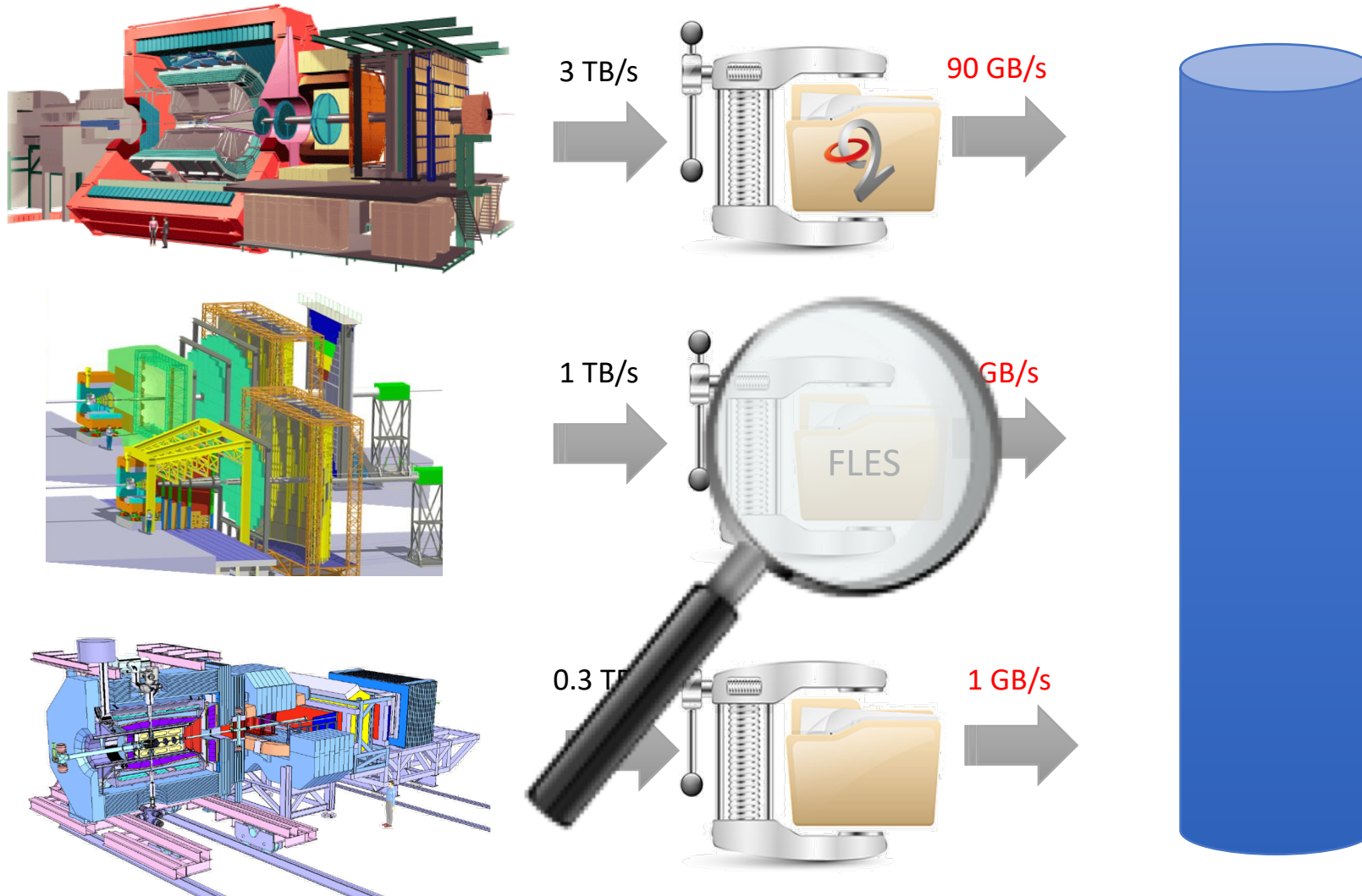
+ 3000 GPUs

- To speed up the reconstruction
- 3 CPU¹⁾ + 1 GPU²⁾ = 28 CPUs

+ 60 PB of disk

- To buy us an extra time and allow more precise calibration

Different experiment similar requirements!



CBM FLES

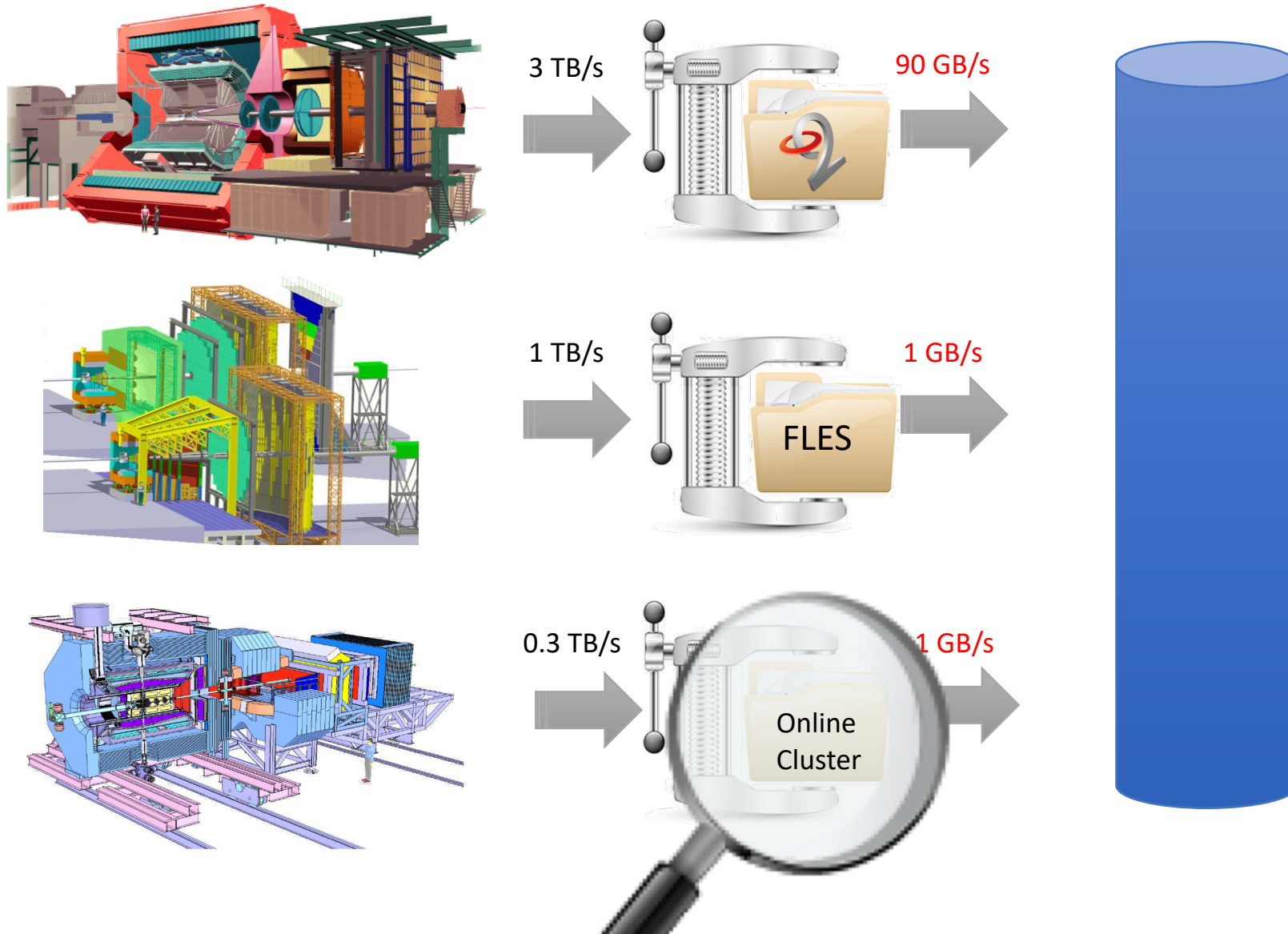
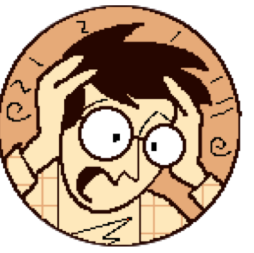
+ 60'000 CPU cores

- To perform online a full event reconstruction on the 1 TB/s input data stream

+ ? GPUs

- To speed up the reconstruction

Different experiment similar requirements!



Panda online

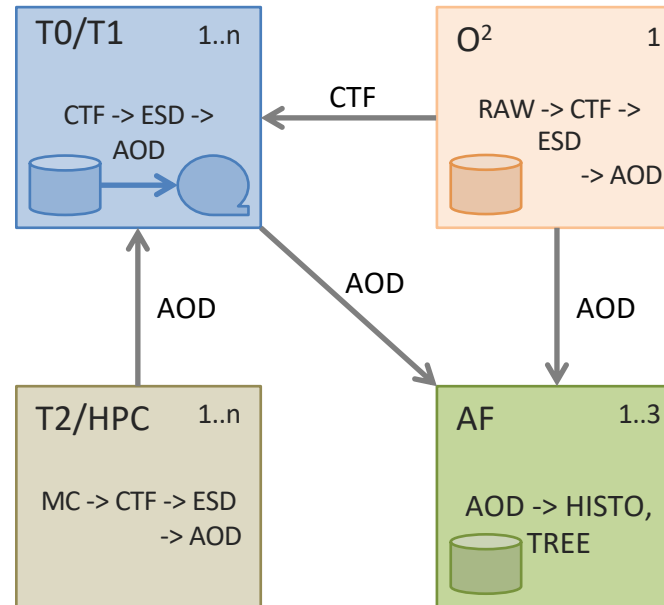
+ 66'000 CPU cores

- To perform online a full event reconstruction on the 300 GB/s input data stream

+ ? GPUs

- To speed up the reconstruction

ALICE Run 3 Computing Model

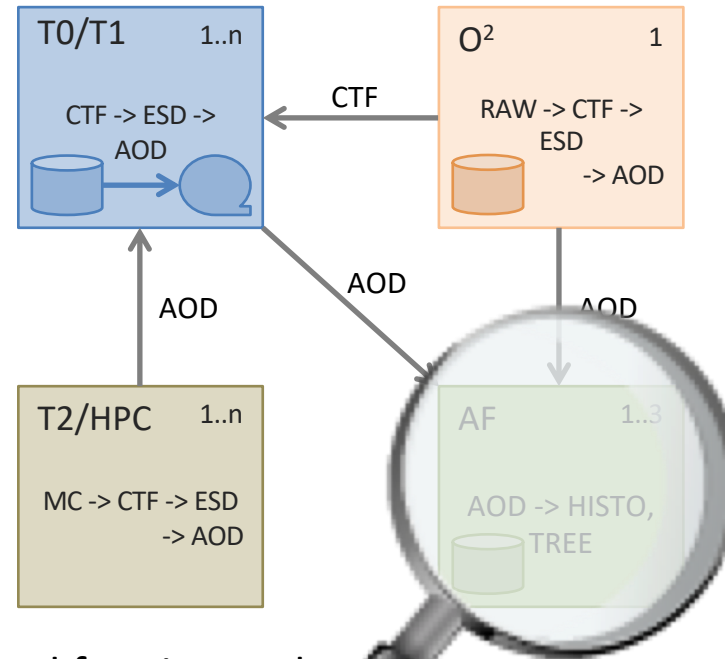


Grid Tiers mostly specialized for given role

- O2 facility (2/3 of reconstruction and calibration), T1s (1/3 of reconstruction and calibration, archiving to tape), T2s (simulation)
- All AODs will be collected on the specialized Analysis Facilities (AF) capable of processing ~5 PB of data within ½ day timescale

The goal is to minimize data movement and optimize processing efficiency

ALICE Run 3 Computing Model

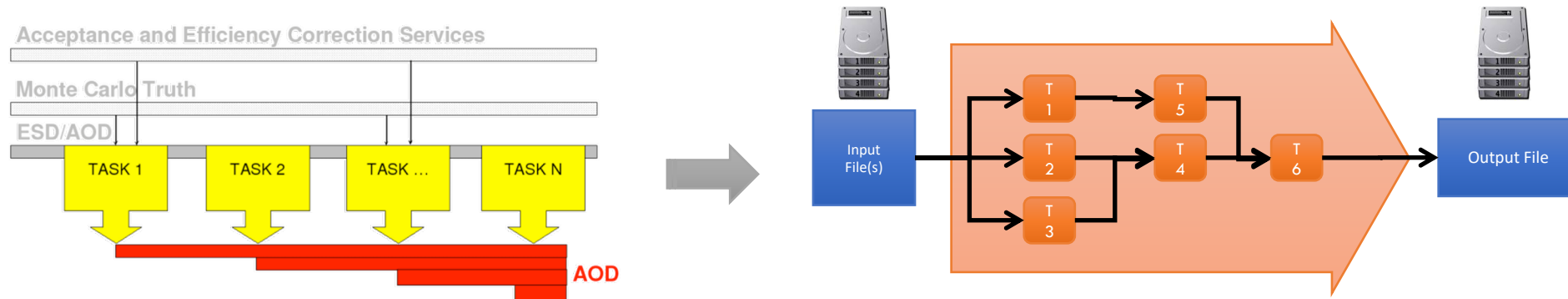


Grid Tiers mostly specialized for given role

- O2 facility (2/3 of reconstruction and calibration), T1s (1/3 of reconstruction and calibration, archiving to tape), T2s (simulation)
- All AODs will be collected on specialized Analysis Facilities (AF) capable of processing ~5 PB of data within ½ day timescale

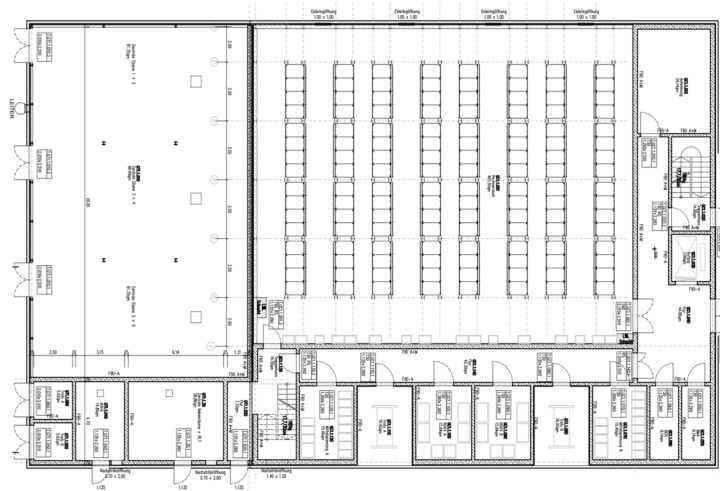
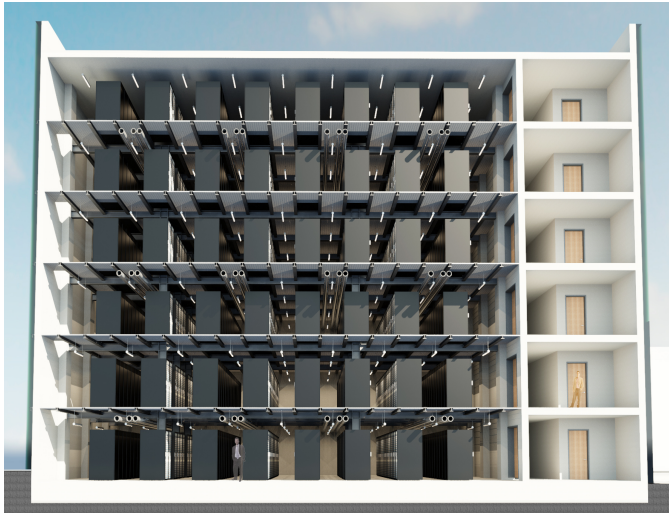
The goal is to minimize data movement and optimize processing efficiency

ALICE Analysis Facilities



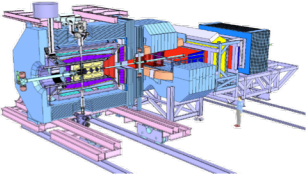
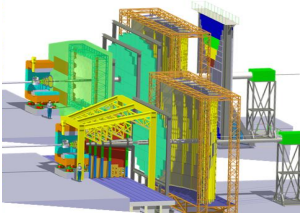
- Motivation
 - Analysis remains I/O bound in spite of attempts to make it more efficient by using the train approach
- Solution
 - Collect AODs on a dedicated sites that are optimized for fast processing of a large local datasets
 - Run organized analysis on local data like we do today on the Grid
 - Requires 20-30'000 cores and 5-10 PB of disk on very performant file system
 - Such sites can be elected between the existing T1s (or even T2s) but ideally this would be a facility with an architecture optimized for such workflow

The FAIR Data Center (Green Cube)



- **Space for 768 19" racks (2,2m)**
- **4 MW cooling (baseline)**
- **Max cooling power 12 MW**
- **Can be used for any commercial IT**
- **PUE <1.07**
- **In operation since Feb. 2016**

Dynamically allocated
resources for exclusive
usage and limited time



Analysis Facilities

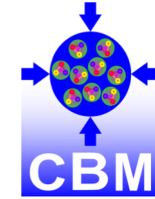


Computing at FAIR: The resources in the Green Cube will be shared between the different FAIR/GSI Partners



No sperate hardware for the online clusters of CBM and PANDA

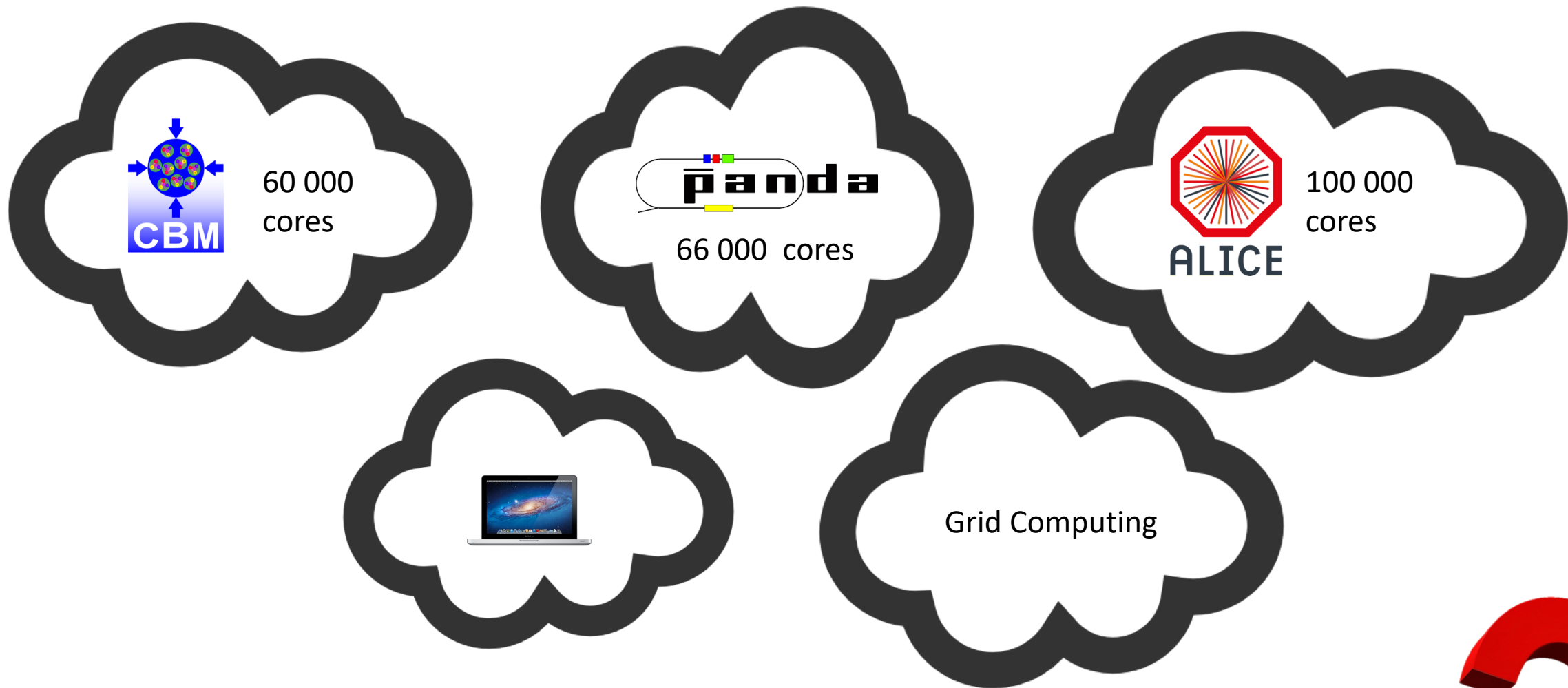
Generic batch farm for
GSI/FAIR Users



• • • • •

Different and large infrastructures but what about software? Tools and Frameworks?





You have to support all these cases



AliceO2
<http://alice-o2.web.cern.ch/>

CbmRoot
<https://fair-center.eu/for-users/experiments/cbm.html>

PandaRoot
<https://panda.gsi.de/>

R3BRoot
<https://www.gsi.de/r3b>

FairShip
<http://ship.web.cern.ch/ship/>

SofiaRoot

AsyEosRoot

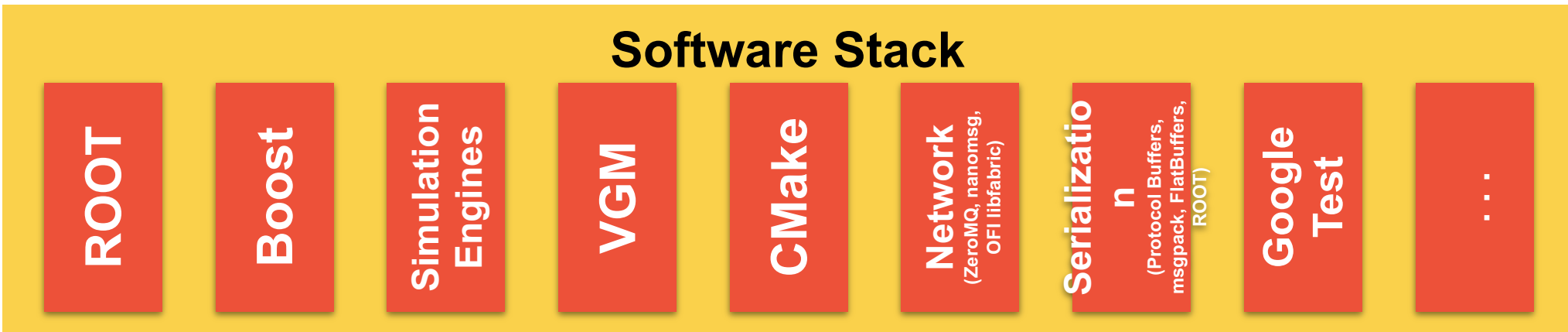
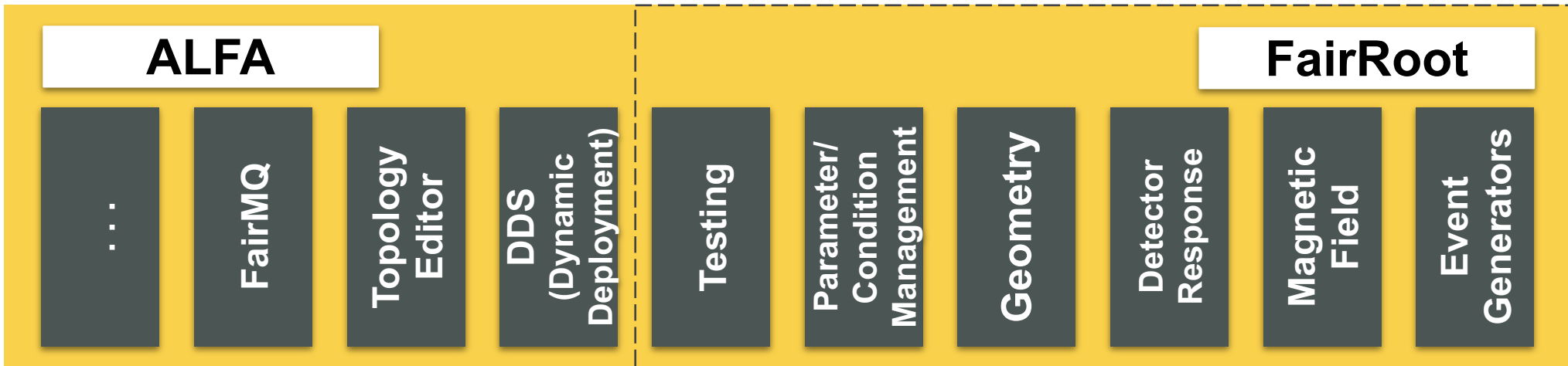
MPDRoot
<http://mpd.jinr.ru>

ExpertRoot
<http://er.jinr.ru/>

EnsarRoot
<http://igfae.usc.es/satnurse/ensarroot.html>

ATTPCRootv2
<https://github.com/ATTPC/ATTPCROOTv2>

BNMRoot
<http://mpd.jinr.ru>



Design

- Looking at the IT landscape: Clear 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

FairRoot/ALFA

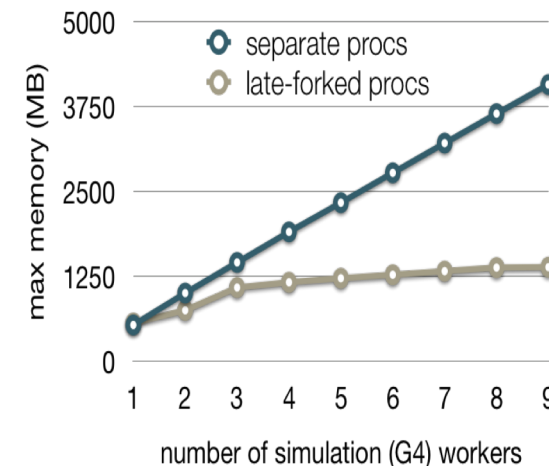
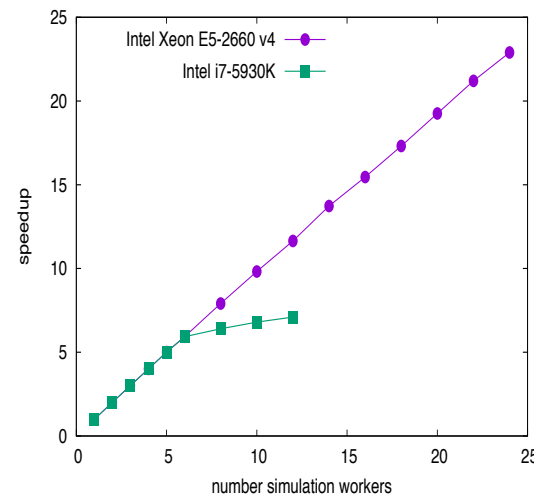
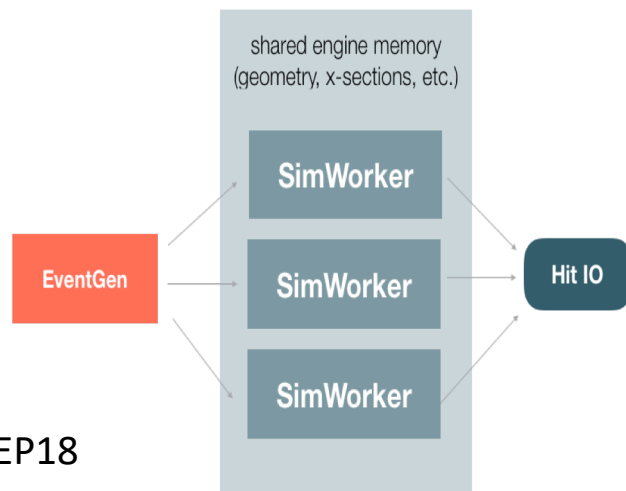
- Core Team at the GSI
 - 4 FTE IT
 - 4 shared FTE ($4 * \{ \frac{1}{2} \text{ FTE for each experiment} + \frac{1}{2} \text{ FTE} \}$)
- Close collaboration with external experiments (ALICE, NICA, SHiP)
- Open source with LGPL and distributed on GitHub:
 - Many contributions from people out side HEP community

Simulation



Parallel high-performance simulation framework

- Development of a scalable and asynchronous parallel simulation system based on independent actors and FairMQ messaging
- Supports parallelization of simulation for any VMC engine
- Supports sub-event parallelism
 - Make simulation jobs more fine-granular for improved scheduling and resource utilization
- Demonstrated strong scaling speedup (24 core server) for workers collaborating on few large Pb-Pb event
- Small memory footprint due to particular "late-forking" technique (demonstrated with Geant4)
- In result, reduce wall-time to treat a Pb-Pb events from O(h) to few minutes and consequently gain access to opportunistic resources



Another new tool with focus on analysis



law
luigi analysis workflow



Distributed make-like Analyses on the Grid
based on Spotify's Pipelining Package **Luigi**

Marcel Rieger

law Summary



**RWTHAACHEN
UNIVERSITY**



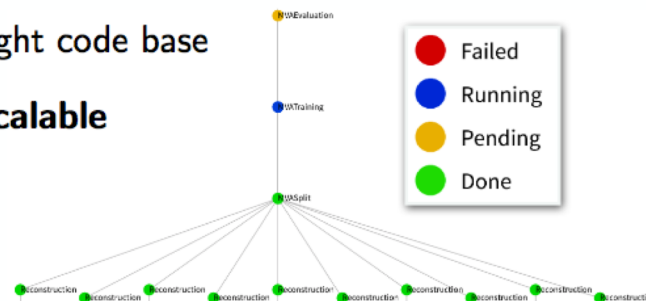
Motivation

- HEP analyses **will** increase in scale and complexity
 - ▷ Workflow management **essential** for success of future measurements
 - ▷ We **need a toolbox** providing an analysis **design** pattern, **not another framework**



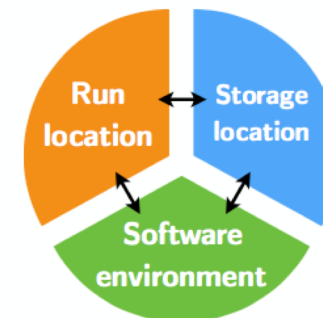
- Powerful Python package from **Spotify** for building **reproducible workflows**
 - Active open-source community
 - Lightweight code base
 - Highly **scalable**
- Extends **luigi** to HEP infrastructures
 - **Goal:** "Develop analyses fully independent of **execution**, **storage** and **environment**."
 - ▷ **Job submission** to HTCondor, WLCG, ...
 - ▷ **File access** on dCache, EOS, XRootD, ...
 - ▷ **Sandboxing** via Docker, Singularity, ...
 - **Successfully used** in multiple analyses

github.com/riga/law



Achievements

1. **Entire analysis** executable make-like with a single command
2. Decoupling of **run locations**, **storage locations** & **software environments**
 - ▷ **No limitation** to particular resources, **effortless adaption** to new ones
3. **Analysis preservation** out-of-the-box



Take home message for software development:

- Stable core team for the software development at the GSI that is funded partially by the experiment was/is crucial for such a large project (15 experiments in different institutes)
- Working with Hochschule Darmstadt (KOSI Program) we could win computing scientist and hire them later to support the experiment, you cannot rely only on physics student and post docs in such environment.
- Open source policy and license helped also in improving the software

Backup

How to deploy ALFA on a laptop, few PCs or a cluster?













DDS: Dynamic Deployment System

Users describe desired tasks and their dependencies using topology (graph) files

Users are provided with a WEB GUI to create topology (Can be created manually as well).

<http://dds.gsi.de/>

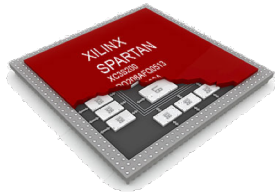


- *law* - *luigi* analysis workflow
 - Repository  github.com/riga/law
 - Paper  [1706.00955](#) (CHEP16 proceedings)
 - Documentation  law.readthedocs.io (in preparation)
 - Minimal example  github.com/riga/law/tree/master/examples/loremipsum
 - HTCondor example  github.com/riga/law/tree/master/examples/htcondor_at_cern
 - Contact  [Marcel Rieger](#)
- *luigi* - Powerful Python pipelining package (by Spotify)
 - Repository  github.com/spotify/luigi
 - Documentation  luigi.readthedocs.io
 - "Hello world!"  github.com/spotify/luigi/blob/master/examples/hello_world.py
- Technologies
 - GFAL2  dmc.web.cern.ch/projects/gfal-2/home
 - Docker  docker.com
 - Singularity  singularity.lbl.gov

Message format ?



The framework does not impose any format on messages.



It supports different serialization standards

- BOOST C++ serialization
- Google's protocol buffers
- ROOT
- Flatbuffers
- MessagePack
- User defined



O2 Facility

