Report from Streaming Readout X (May 17–19)

Markus Diefenthaler (Jefferson Lab)

Streaming Readout I – X

Workshop series on streaming readout R&D for the EIC and other NP experiments

Organized by EIC Streaming Readout Consortium (PIs: Marco Battaglieri (INFN Genoa), Jan Bernauer (SBU))

2017	**SR I** (online)	
2018	**SR II** (MIT)	**SR III** (CNU)
2019	**SR IV** (Italy)	**SR V** (BNL)
2020	**SR VI** (JLab)	**SR VII** (BNL)
2021	**SR VIII** (MIT)	**SR IX** (ORNL)
2022	**SR X** (JLab)	

Streaming Readout X

May 17–19, 2022

Streaming Readout X (Jefferson Lab)

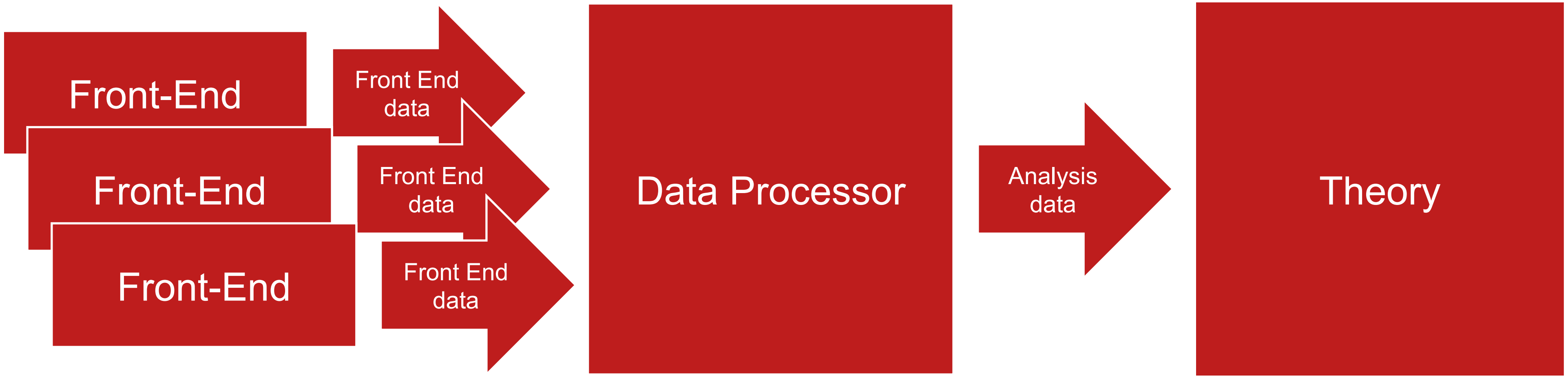
Hybrid workshop, hosted by Jefferson Lab

74 participants

- **Following up on**:
  - Review of the detector collaboration proposals for the EIC (ATHENA, CORE, ECCE).
  - Ongoing collaboration formation to support the realization of the EIC project detector.
- **Focus**:
  - Review the progress on streaming readout electronics, computing, and software.
  - Discuss the future priorities for the EIC Streaming Readout Consortium.
  - Mini town hall meeting on streaming readout technologies of the NP community.

Overall Product

Integration of DAQ, analysis and theory to optimize physics reach



Compute-Detector Integration

- Research model with seamless data processing from DAQ to data analysis:
  - Not about building the best detector,
  - But the best detector that fully supports streaming readout and fast ML for alignment, calibration, and reconstruction in near real time.
  - For rapid turnaround of data for the physics analysis and to start the work on publications.

Compute-Detector Integration for the EIC

Definition of Streaming Readout

- Data is digitized at a fixed rate with thresholds and zero suppression applied locally.
- Data is read out in continuous parallel streams that are encoded with information about when and where the data was taken.
- Event building, filtering, monitoring, and other processing is deferred until the data is at rest in tiered storage.

Advantages of Streaming Readout

- **Simplification of readout**: No custom trigger hardware and firmware.
- Trigger-less readout ideal for the **general-purpose detectors** of the EIC.
- **Opportunity to streamline workflows**: Merging of online and offline computing with combined software stack.
- Take advantage of other emerging technologies:
  - **AI**: Intelligent decisions in all aspects of data processing from detector readout and control to analysis.
  - **Heterogenous computing**.

Streaming Readout in “EIC Software: Statement of Principles”

EIC Software: Statement of Principles

- **Guiding principles to frame the discussion about requirements for EIC Software and resulting approaches and solutions**.

2) **We will have an unprecedented compute-detector integration:**
   - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
   - We aim for autonomous alignment and calibration.
   - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.
3) **We will leverage heterogeneous computing:**
   - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
   - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
   - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.

Streaming Readout X: Program

Streaming Readout Status

Streaming DAQ

Streaming Readout Electronics

Streaming Readout Data

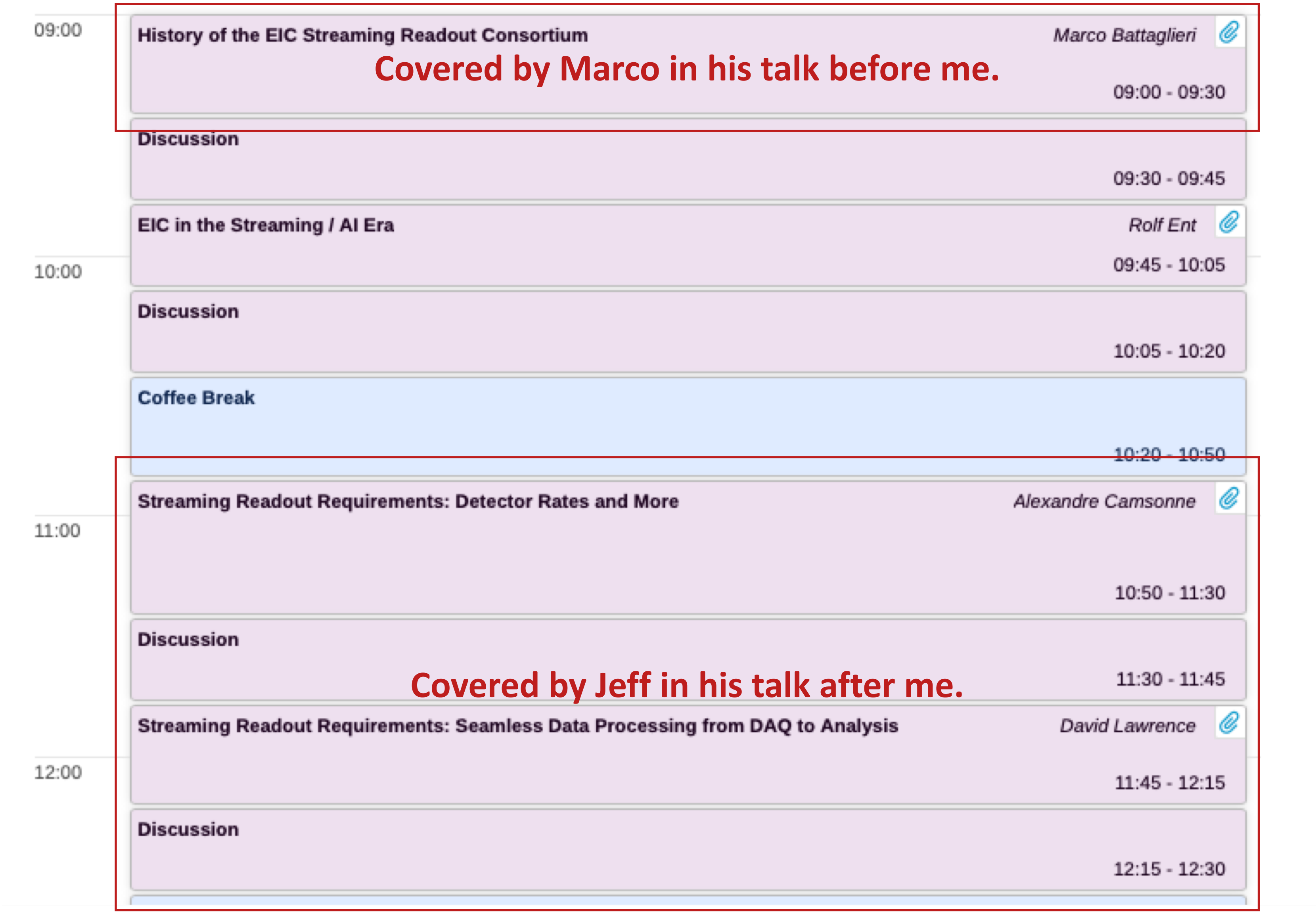
Streaming Readout in NHEP

Streaming Readout Community

Streaming Readout X: Program

- **Streaming Readout Status**
- **Streaming DAQ**
- **Streaming Readout Data**
- **Streaming Readout Electronics**
- **Streaming Readout in NHEP**
- **Streaming Readout Community**

Streaming Readout Status



09:00

History of the EIC Streaming Readout Consortium — *Marco Battaglieri* — 09:00 - 09:30

Covered by Marco in his talk before me.

Discussion — 09:30 - 09:45

EIC in the Streaming / AI Era — *Rolf Ent* — 09:45 - 10:05

10:00

Discussion — 10:05 - 10:20

Coffee Break — 10:20 - 10:50

11:00

Streaming Readout Requirements: Detector Rates and More — *Alexandre Camsonne* — 10:50 - 11:30

Discussion — 11:30 - 11:45

Covered by Jeff in his talk after me.

12:00

Streaming Readout Requirements: Seamless Data Processing from DAQ to Analysis — *David Lawrence* — 11:45 - 12:15

Discussion — 12:15 - 12:30

Summary – EIC in the Streaming/AI Era

Slide by Rolf Ent

- Community efforts towards streaming at the EIC started in earnest in ~2018
  - Now, a mere four years later, streaming readout is the default for the envisioned EIC detector
  - The advances in microelectronics and commercial data handling hardware are our friends ☺
  - Many efforts are ongoing withing the EIC community
  - And we are only busy ~4 years with a decade to go before EIC detector operations start

- Community efforts towards AI at the EIC started in earnest in ~2020
  - The AI4NP workshop and white paper, AI4NP winter school (369 registered participants), AI4EIC workshop series (1st workshop with 243 registered participants) all were a huge success
  - AI is our friend and a perfect fit for the nuclear science we do
  - AI is being integrated in all aspects of the EIC detector (design, calibration, simulation, reconstruction, analysis)
  - Here also amazing momentum has been gathered

- To take full benefit of streaming and AI implies use of heterogeneous computing
  - AI requires to integrate the power of GPUs in our workflow
  - Our colleagues in Lattice QCD have illustrated the power of combining CPUs, GPUs and modern software
  - The increase in network bandwidth is our friend ☺ - just imagine 1.6 Tbps by then…
  - Similar, the developments in statistical methods and data science are our friends ☺

Summary – EIC in the Streaming/AI Era

Slide by Rolf Ent

Combining Streaming, AI, heterogeneous computing and modern software in our physics detector, data handling and analysis (from calibration to high-level physics analysis) is **a no-brainer**.

We “just” have to make it work.

From the workshop discussion: “Making it work” requires that we understand the requirements and biases of this approach.

Heterogenous architectures, AI/ML, and the other technologies are rapidly evolving…
On the timescale of EIC data taking the landscape is likely to be completely different.
We must ensure an agile framework that can evolve rapidly over time!

Streaming Readout X: Program

Streaming Readout Status

Streaming DAQ

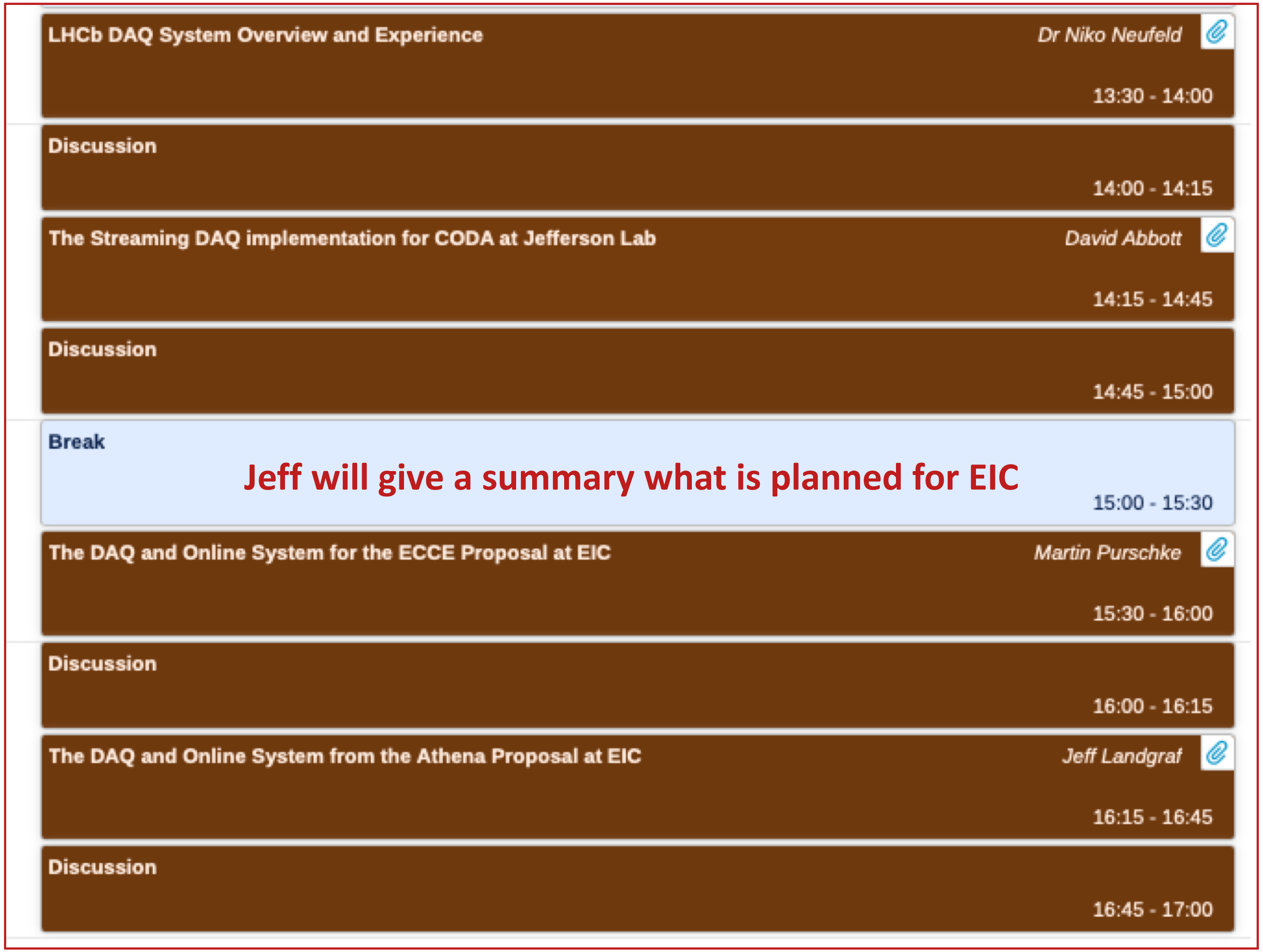
Streaming Readout Data

Streaming Readout Electronics

Streaming Readout in NHEP

Streaming Readout Community

Streaming DAQ



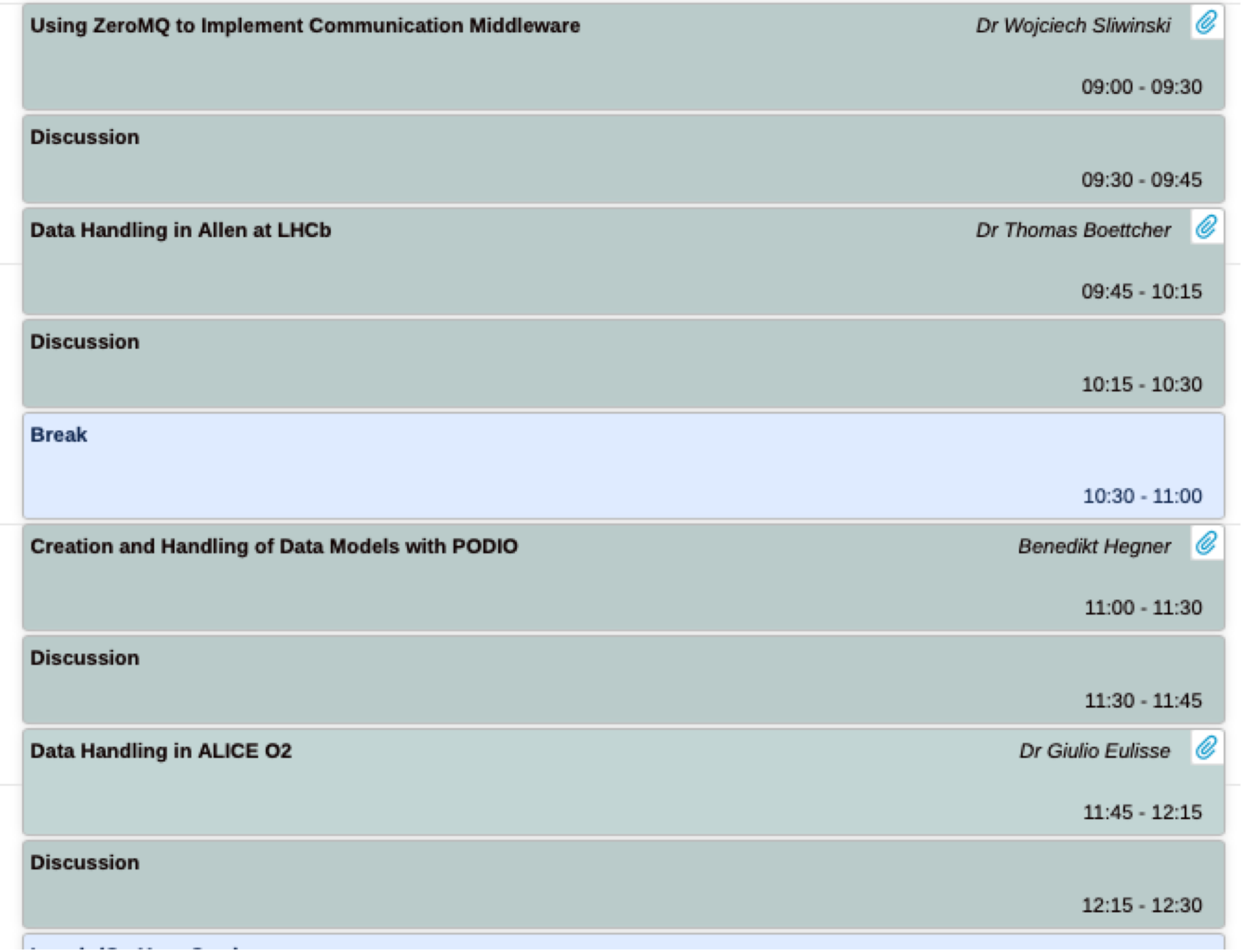
LHCb DAQ System Overview and Experience	*Dr Niko Neufeld*	13:30 - 14:00
Discussion		14:00 - 14:15
The Streaming DAQ implementation for CODA at Jefferson Lab	*David Abbott*	14:15 - 14:45
Discussion		14:45 - 15:00
Break		15:00 - 15:30
The DAQ and Online System for the ECCE Proposal at EIC	*Martin Purschke*	15:30 - 16:00
Discussion		16:00 - 16:15
The DAQ and Online System from the Athena Proposal at EIC	*Jeff Landgraf*	16:15 - 16:45
Discussion		16:45 - 17:00

Jeff will give a summary what is planned for EIC

Streaming Readout X: Program

- Streaming Readout Status
- Streaming DAQ
- **Streaming Readout Data**
- Streaming Readout Electronics
- Streaming Readout in NHEP
- Streaming Readout Community

Streaming Readout Data



Using ZeroMQ to Implement Communication Middleware *Dr Wojciech Sliwinski*
09:00 - 09:30

Discussion
09:30 - 09:45

Data Handling in Allen at LHCb *Dr Thomas Boettcher*
09:45 - 10:15

Discussion
10:15 - 10:30

Break
10:30 - 11:00

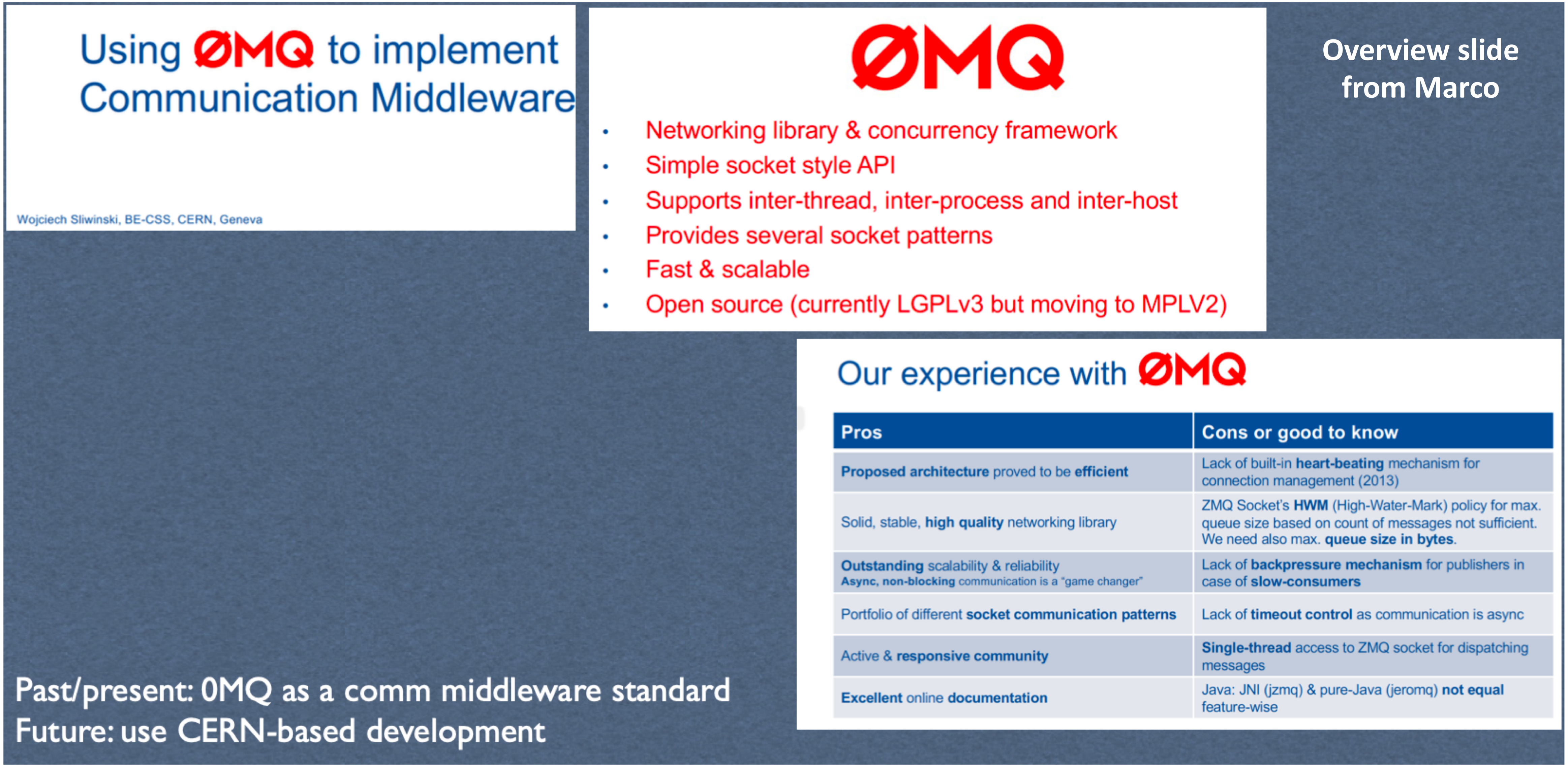
Creation and Handling of Data Models with PODIO *Benedikt Hegner*
11:00 - 11:30

Discussion
11:30 - 11:45

Data Handling in ALICE O2 *Dr Giulio Eulisse*
11:45 - 12:15

Discussion
12:15 - 12:30

Using 0MQ to implement Communication Middleware (Wojciech Sliwinski, CERN)



Using ØMQ to implement Communication Middleware

Wojciech Sliwinski, BE-CSS, CERN, Geneva

ØMQ

- Networking library & concurrency framework
- Simple socket style API
- Supports inter-thread, inter-process and inter-host
- Provides several socket patterns
- Fast & scalable
- Open source (currently LGPLv3 but moving to MPLV2)

Overview slide from Marco

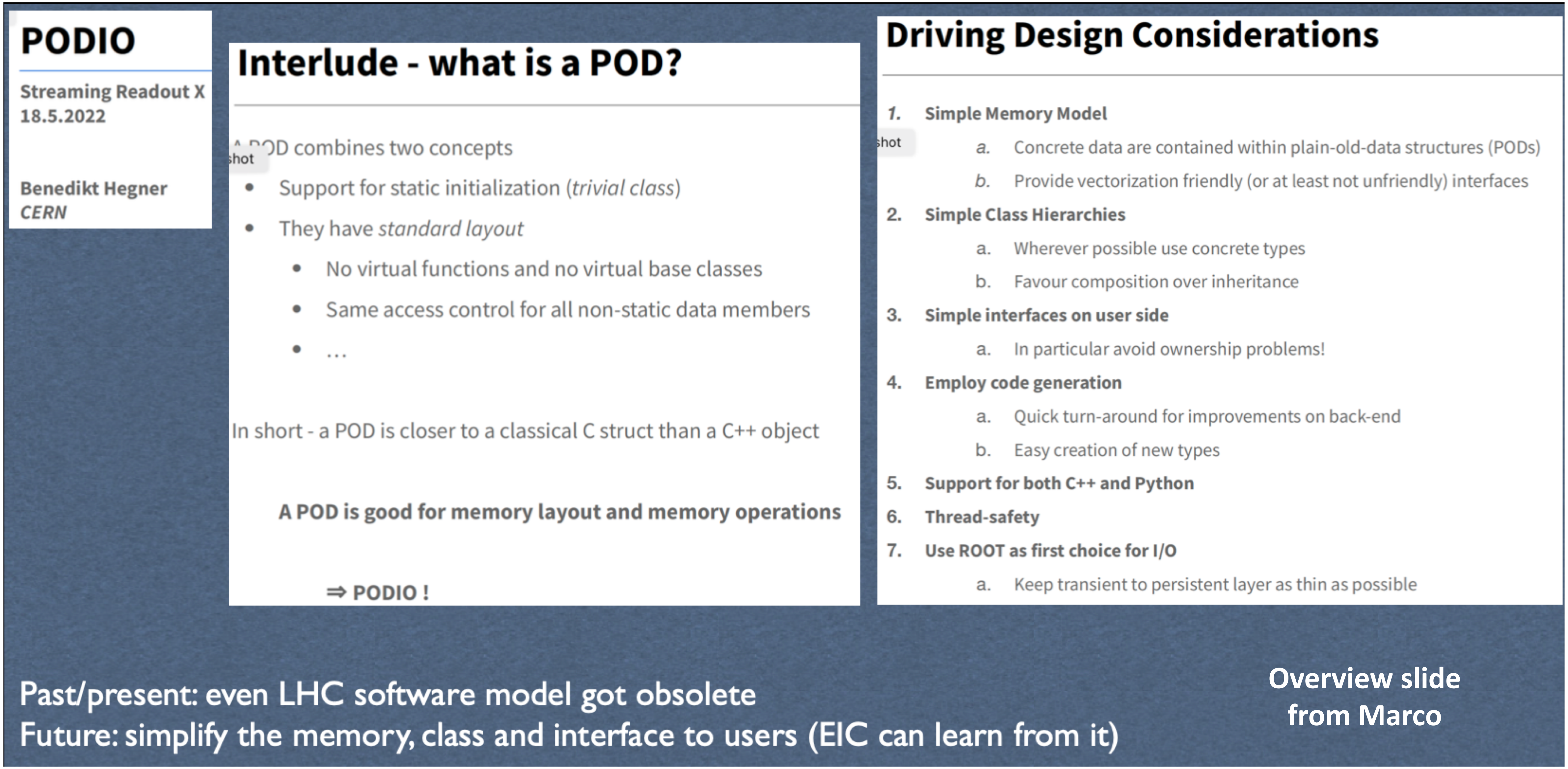
Our experience with ØMQ

Pros	Cons or good to know
Proposed architecture proved to be **efficient**	Lack of built-in **heart-beating** mechanism for connection management (2013)
Solid, stable, **high quality** networking library	ZMQ Socket's **HWM** (High-Water-Mark) policy for max. queue size based on count of messages not sufficient. We need also max. **queue size in bytes**.
Outstanding scalability & reliability **Async, non-blocking** communication is a "game changer"	Lack of **backpressure mechanism** for publishers in case of **slow-consumers**
Portfolio of different **socket communication patterns**	Lack of **timeout control** as communication is async
Active & **responsive community**	**Single-thread** access to ZMQ socket for dispatching messages
Excellent online **documentation**	Java: JNI (jzmq) & pure-Java (jeromq) **not equal** feature-wise

Past/present: 0MQ as a comm middleware standard

Future: use CERN-based development

PODIO (Benedikt Hegner, CERN)



PODIO

Streaming Readout X
18.5.2022

Benedikt Hegner
CERN

Interlude - what is a POD?

A POD combines two concepts

- Support for static initialization (*trivial class*)
- They have *standard layout*
  - No virtual functions and no virtual base classes
  - Same access control for all non-static data members
  - …

In short - a POD is closer to a classical C struct than a C++ object

A POD is good for memory layout and memory operations

⇒ PODIO !

Driving Design Considerations

1. **Simple Memory Model**
   a. Concrete data are contained within plain-old-data structures (PODs)
   b. Provide vectorization friendly (or at least not unfriendly) interfaces
2. **Simple Class Hierarchies**
   a. Wherever possible use concrete types
   b. Favour composition over inheritance
3. **Simple interfaces on user side**
   a. In particular avoid ownership problems!
4. **Employ code generation**
   a. Quick turn-around for improvements on back-end
   b. Easy creation of new types
5. **Support for both C++ and Python**
6. **Thread-safety**
7. **Use ROOT as first choice for I/O**
   a. Keep transient to persistent layer as thin as possible

Past/present: even LHC software model got obsolete
Future: simplify the memory, class and interface to users (EIC can learn from it)

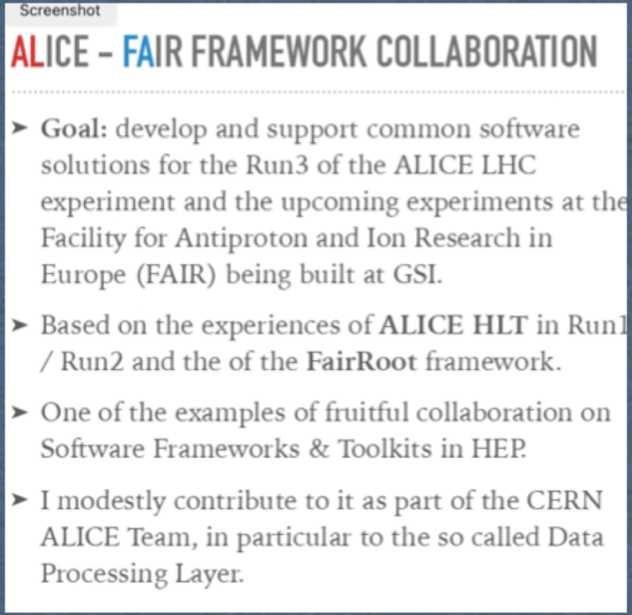
Overview slide
from Marco

ALICE Software Framework for Run 3 (Giulio Eulisse, CERN)

Screenshot

ALICE SOFTWARE FRAMEWORK FOR RUN 3

Giulio Eulisse - CERN

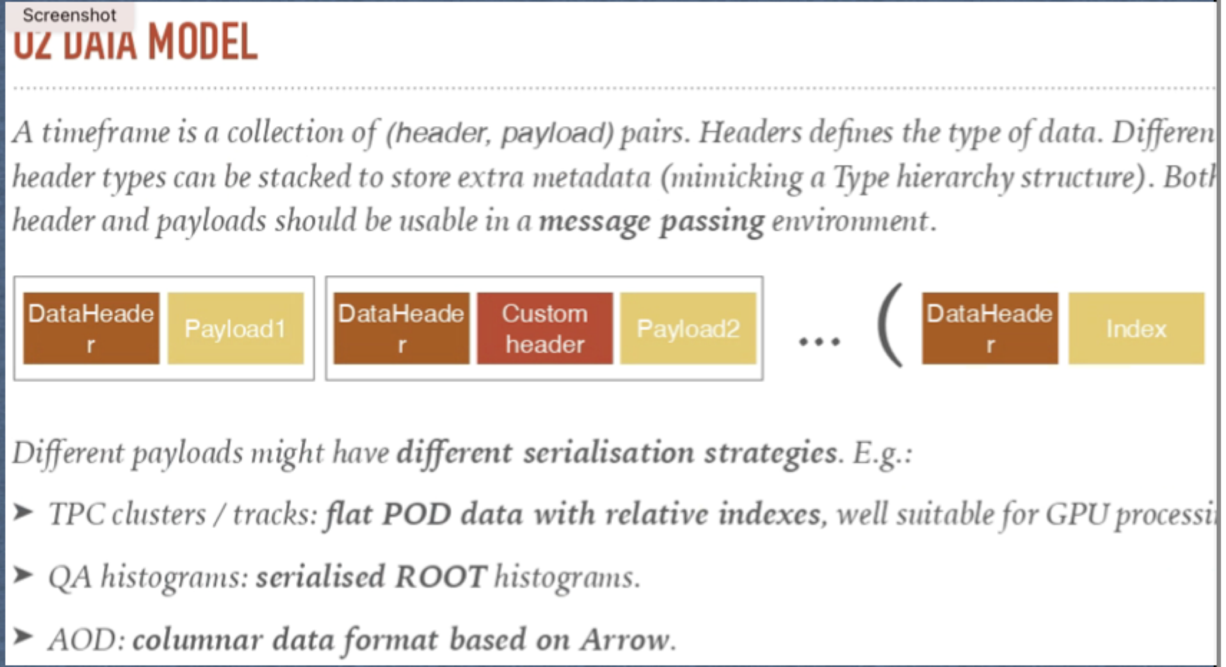


Screenshot

ALICE – FAIR FRAMEWORK COLLABORATION

- **Goal:** develop and support common software solutions for the Run3 of the ALICE LHC experiment and the upcoming experiments at the Facility for Antiproton and Ion Research in Europe (FAIR) being built at GSI.
- Based on the experiences of **ALICE HLT** in Run1 / Run2 and the of the **FairRoot** framework.
- One of the examples of fruitful collaboration on Software Frameworks & Toolkits in HEP.
- I modestly contribute to it as part of the CERN ALICE Team, in particular to the so called Data Processing Layer.

Overview slide from Marco



Screenshot

O2 DATA MODEL

A timeframe is a collection of (header, payload) pairs. Headers defines the type of data. Differen header types can be stacked to store extra metadata (mimicking a Type hierarchy structure). Both header and payloads should be usable in a ***message passing*** *environment.*

DataHeader | Payload1 | DataHeader | Custom header | Payload2 | … (| DataHeader | Index

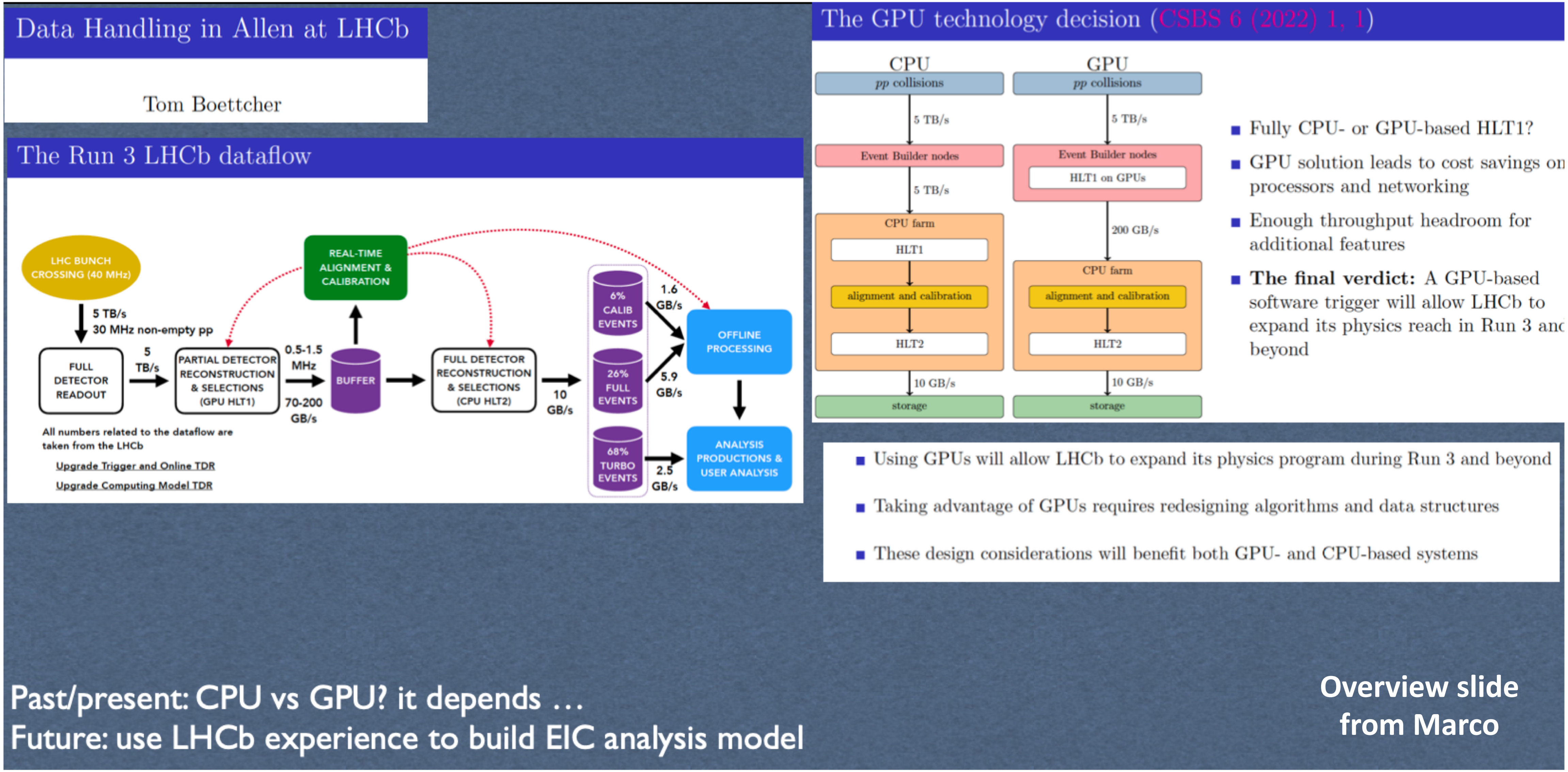
Different payloads might have ***different serialisation strategies****. E.g.:*

- *TPC clusters / tracks:* ***flat POD data with relative indexes****, well suitable for GPU processi*
- *QA histograms:* ***serialised ROOT*** *histograms.*
- *AOD:* ***columnar data format based on Arrow****.*

Past/present: another LHC data analysis model to learn from

Future: EIC SRO design will benefit by an early off(on)-line sw model definition

Data Handling in Allen at LHCb (Tom Boettcher)



Data Handling in Allen at LHCb

Tom Boettcher

The Run 3 LHCb dataflow

LHC BUNCH CROSSING (40 MHz)
5 TB/s
30 MHz non-empty pp
FULL DETECTOR READOUT
5 TB/s
PARTIAL DETECTOR RECONSTRUCTION & SELECTIONS (GPU HLT1)
0.5-1.5 MHz
70-200 GB/s
REAL-TIME ALIGNMENT & CALIBRATION
BUFFER
FULL DETECTOR RECONSTRUCTION & SELECTIONS (CPU HLT2)
10 GB/s
6% CALIB EVENTS
1.6 GB/s
26% FULL EVENTS
5.9 GB/s
68% TURBO EVENTS
2.5 GB/s
OFFLINE PROCESSING
ANALYSIS PRODUCTIONS & USER ANALYSIS

All numbers related to the dataflow are taken from the LHCb

Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

The GPU technology decision (CSBS 6 (2022) 1, 1)

- Fully CPU- or GPU-based HLT1?
- GPU solution leads to cost savings on processors and networking
- Enough throughput headroom for additional features
- **The final verdict:** A GPU-based software trigger will allow LHCb to expand its physics reach in Run 3 anc beyond

- Using GPUs will allow LHCb to expand its physics program during Run 3 and beyond
- Taking advantage of GPUs requires redesigning algorithms and data structures
- These design considerations will benefit both GPU- and CPU-based systems

Past/present: CPU vs GPU? it depends …
Future: use LHCb experience to build EIC analysis model

Overview slide from Marco

Streaming Readout X: Program

Streaming Readout Status

Streaming DAQ

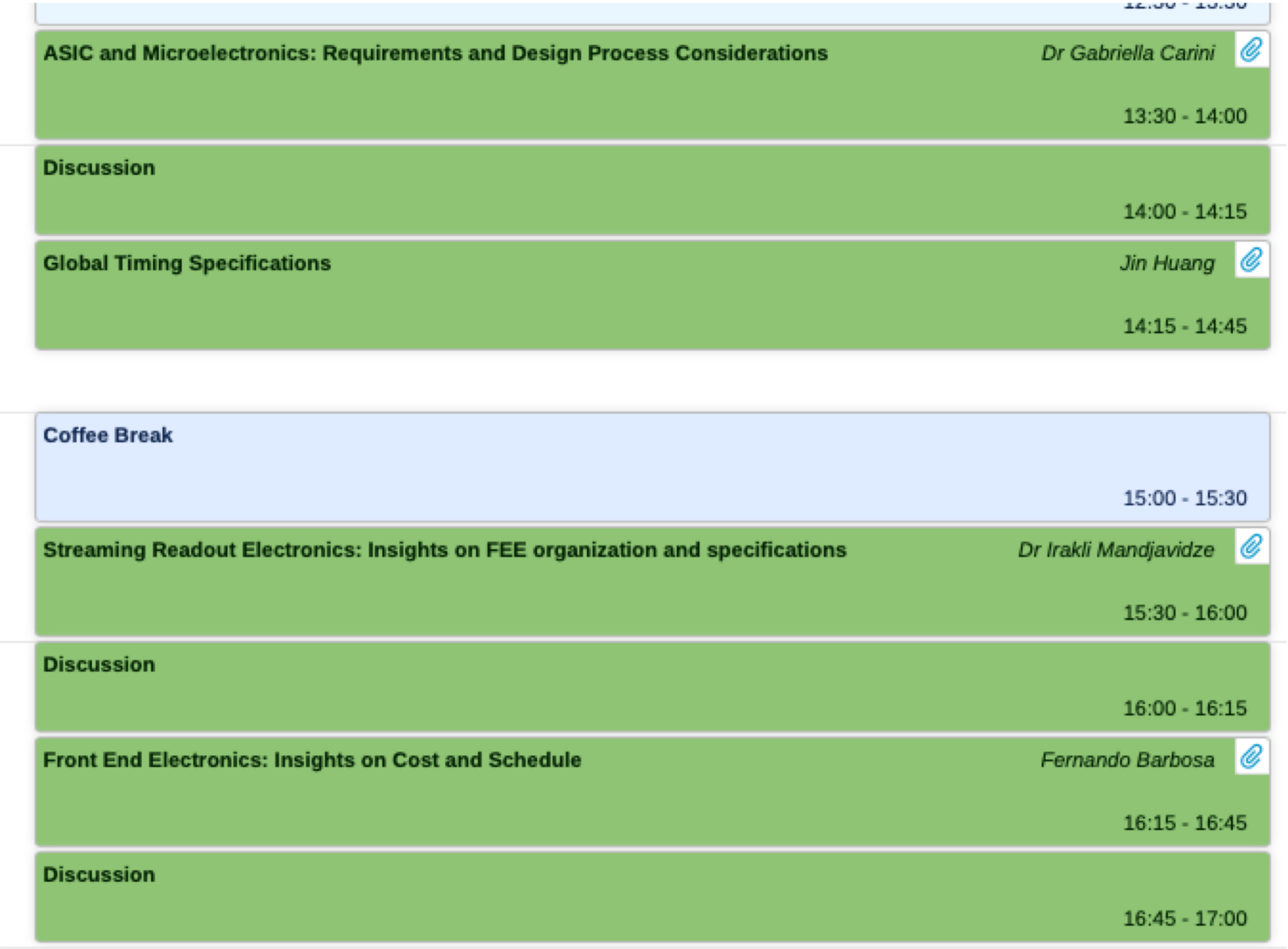
Streaming Readout Data

Streaming Readout Electronics

Streaming Readout in NHEP

Streaming Readout Community

Streaming Readout Electronics



12:30 - 13:30

ASIC and Microelectronics: Requirements and Design Process Considerations *Dr Gabriella Carini*
13:30 - 14:00

Discussion
14:00 - 14:15

Global Timing Specifications *Jin Huang*
14:15 - 14:45

Coffee Break
15:00 - 15:30

Streaming Readout Electronics: Insights on FEE organization and specifications *Dr Irakli Mandjavidze*
15:30 - 16:00

Discussion
16:00 - 16:15

Front End Electronics: Insights on Cost and Schedule *Fernando Barbosa*
16:15 - 16:45

Discussion
16:45 - 17:00

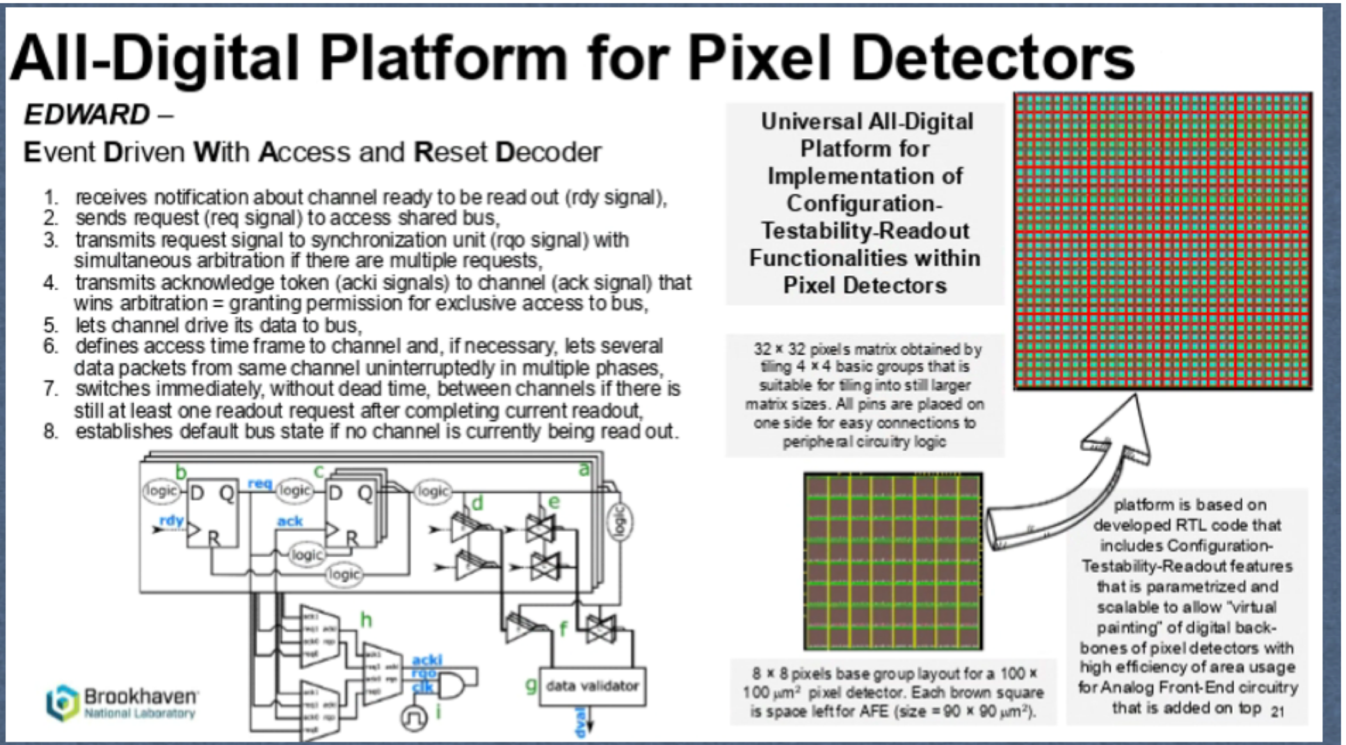
ASIC and Microelectronics: Requirements and Design Process Considerations (Gabriella Carini, BNL)

ASIC and Microelectronics: Requirements and Design Process Considerations

Gabriella Carini
carini@bnl.gov

Overview slide from Marco

Past/present: significant experience in ASICs design at BNL
Future: define the best ASIC for EIC SRO



All-Digital Platform for Pixel Detectors

EDWARD –
Event **D**riven **W**ith **A**ccess and **R**eset **D**ecoder

1. receives notification about channel ready to be read out (rdy signal),
2. sends request (req signal) to access shared bus,
3. transmits request signal to synchronization unit (rqo signal) with simultaneous arbitration if there are multiple requests,
4. transmits acknowledge token (acki signals) to channel (ack signal) that wins arbitration = granting permission for exclusive access to bus,
5. lets channel drive its data to bus,
6. defines access time frame to channel and, if necessary, lets several data packets from same channel uninterruptedly in multiple phases,
7. switches immediately, without dead time, between channels if there is still at least one readout request after completing current readout,
8. establishes default bus state if no channel is currently being read out.

Universal All-Digital Platform for Implementation of Configuration-Testability-Readout Functionalities within Pixel Detectors

32 × 32 pixels matrix obtained by tiling 4 × 4 basic groups that is suitable for tiling into still larger matrix sizes. All pins are placed on one side for easy connections to peripheral circuitry logic

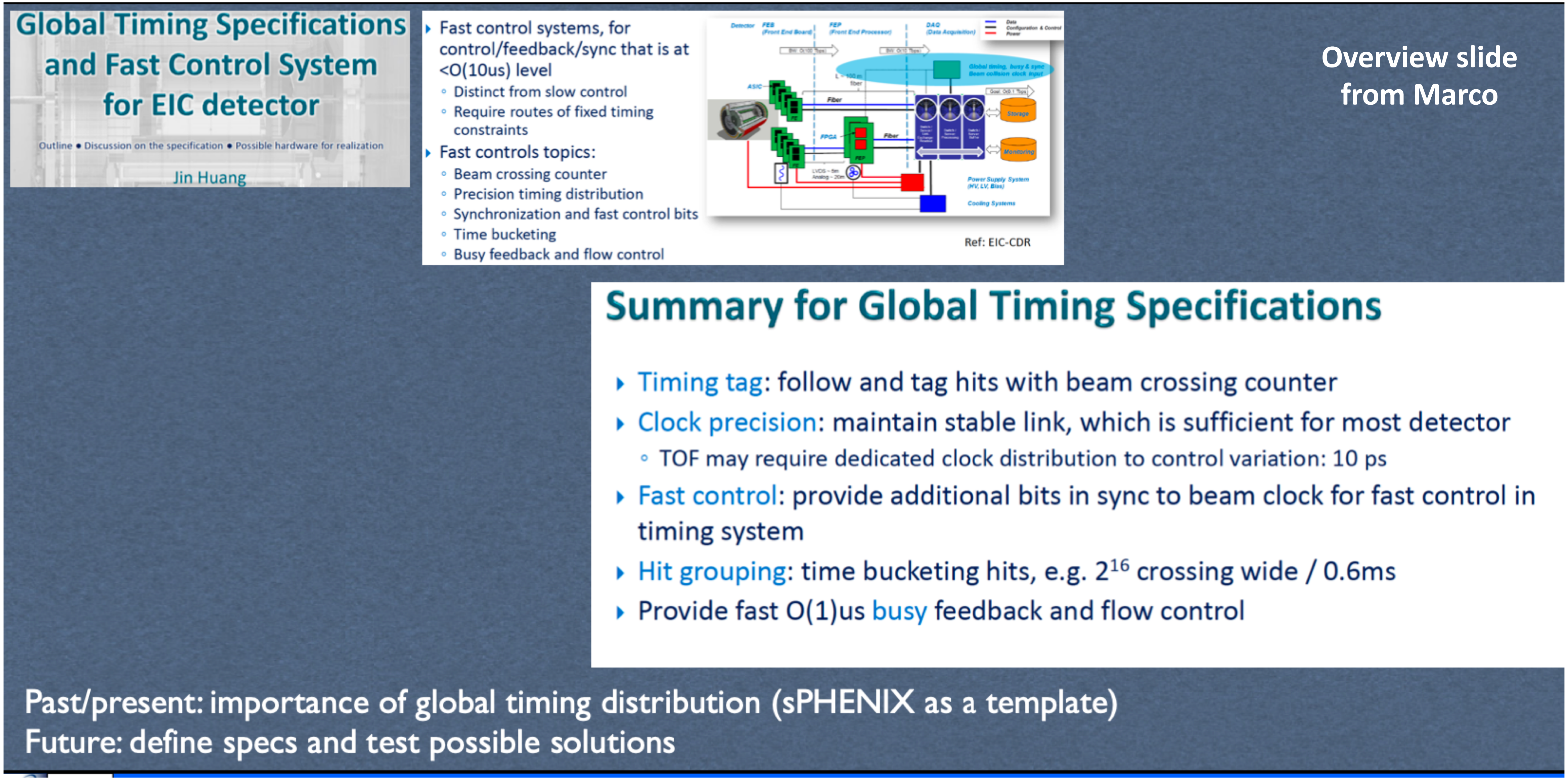
8 × 8 pixels base group layout for a 100 × 100 μm² pixel detector. Each brown square is space left for AFE (size = 90 × 90 μm²).

platform is based on developed RTL code that includes Configuration-Testability-Readout features that is parametrized and scalable to allow "virtual painting" of digital back-bones of pixel detectors with high efficiency of area usage for Analog Front-End circuitry that is added on top 21

Summary

- ASIC and microelectronics continue to evolve to address the challenges of new experiments
- Design methodologies and process technologies are critical drivers
- Competing tendencies: system on a chip vs chiplet approach
- Streaming comes with challenges and opportunities
  - New AI/ML methodologies implemented at the edge to improve data quality/rates
- Concepts and approaches presented in this talk have broad application

Global Timing Specifications and Fast Control System for EIC Detector (Jin Huang, BNL)



Global Timing Specifications and Fast Control System for EIC detector

Outline ● Discussion on the specification ● Possible hardware for realization

Jin Huang

- Fast control systems, for control/feedback/sync that is at <O(10us) level
  - Distinct from slow control
  - Require routes of fixed timing constraints
- Fast controls topics:
  - Beam crossing counter
  - Precision timing distribution
  - Synchronization and fast control bits
  - Time bucketing
  - Busy feedback and flow control

Ref: EIC-CDR

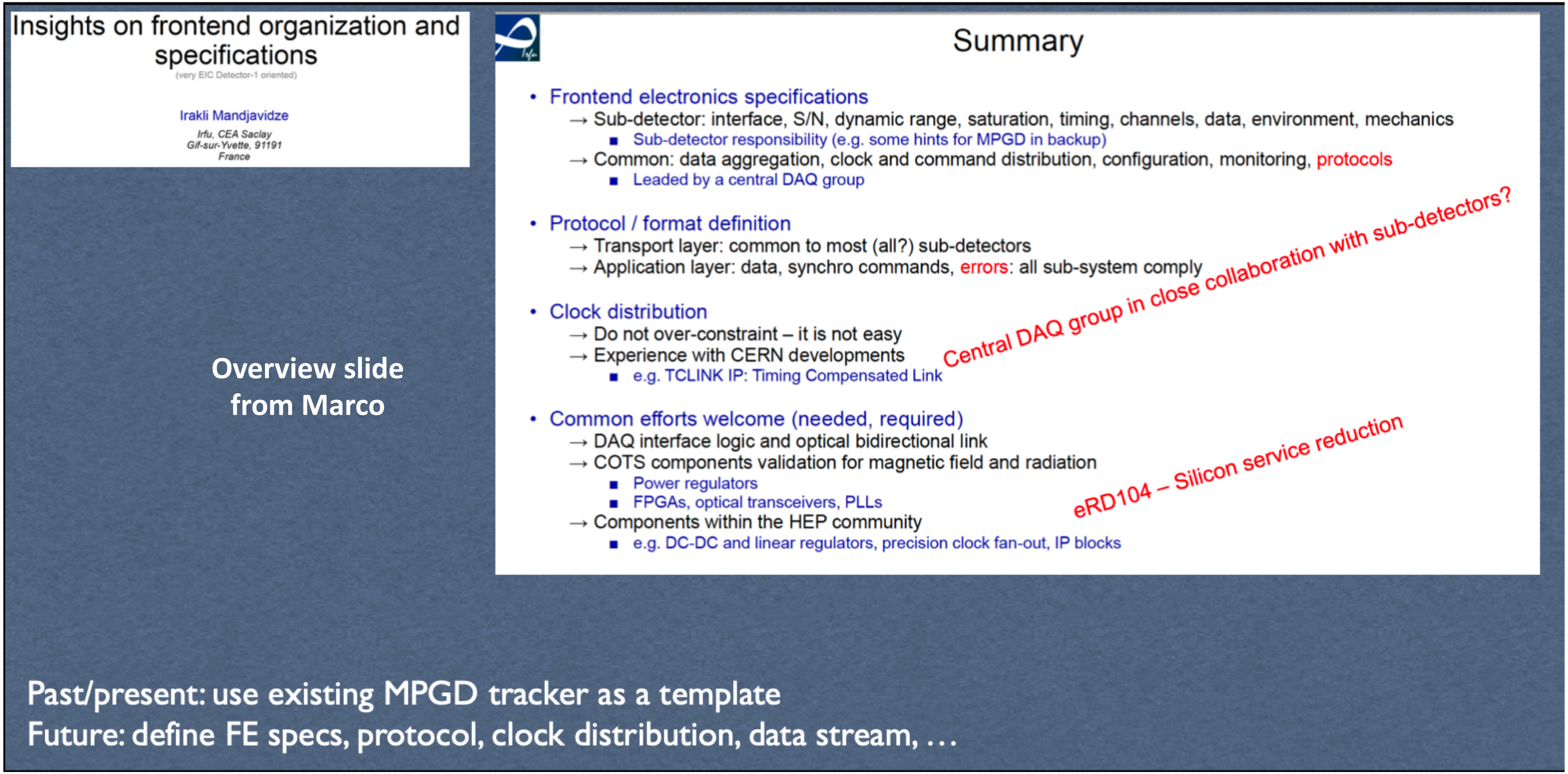
Overview slide from Marco

Summary for Global Timing Specifications

- Timing tag: follow and tag hits with beam crossing counter
- Clock precision: maintain stable link, which is sufficient for most detector
  - TOF may require dedicated clock distribution to control variation: 10 ps
- Fast control: provide additional bits in sync to beam clock for fast control in timing system
- Hit grouping: time bucketing hits, e.g. 2^{16} crossing wide / 0.6ms
- Provide fast O(1)us busy feedback and flow control

Past/present: importance of global timing distribution (sPHENIX as a template)
Future: define specs and test possible solutions

Insights on Frontend Organization and Specifications (Irakli Mandjavidze, CEA Saclay)



Insights on frontend organization and specifications

(very EIC Detector-1 oriented)

Irakli Mandjavidze

Irfu, CEA Saclay
Gif-sur-Yvette, 91191
France

Summary

- Frontend electronics specifications
  - → Sub-detector: interface, S/N, dynamic range, saturation, timing, channels, data, environment, mechanics
    - ▪ Sub-detector responsibility (e.g. some hints for MPGD in backup)
  - → Common: data aggregation, clock and command distribution, configuration, monitoring, protocols
    - ▪ Leaded by a central DAQ group
- Protocol / format definition
  - → Transport layer: common to most (all?) sub-detectors
  - → Application layer: data, synchro commands, errors: all sub-system comply
- Clock distribution
  - → Do not over-constraint – it is not easy
  - → Experience with CERN developments
    - ▪ e.g. TCLINK IP: Timing Compensated Link
- Common efforts welcome (needed, required)
  - → DAQ interface logic and optical bidirectional link
  - → COTS components validation for magnetic field and radiation
    - ▪ Power regulators
    - ▪ FPGAs, optical transceivers, PLLs
  - → Components within the HEP community
    - ▪ e.g. DC-DC and linear regulators, precision clock fan-out, IP blocks

Central DAQ group in close collaboration with sub-detectors?

eRD104 – Silicon service reduction

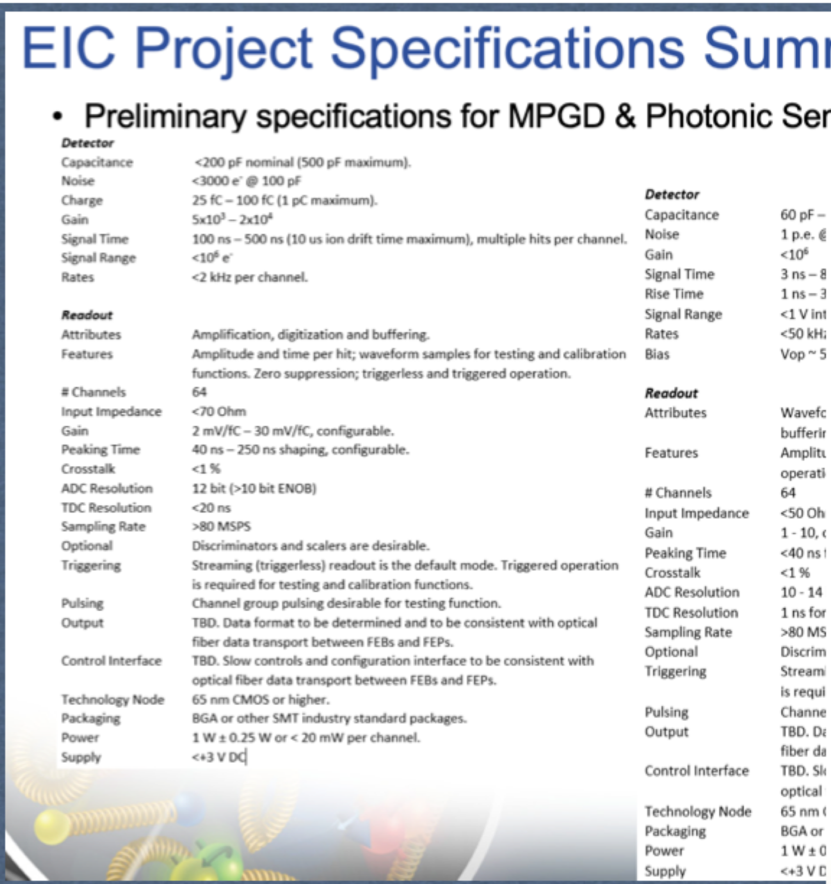
Overview slide from Marco

Past/present: use existing MPGD tracker as a template
Future: define FE specs, protocol, clock distribution, data stream, …

Front End Electronics: Insights on Cost and Schedule (Fernando Barbossa, Jefferson Lab)

Streaming Readout X Front End Electronics: Insights on Cost and Schedule

Fernando Barbosa, JLab



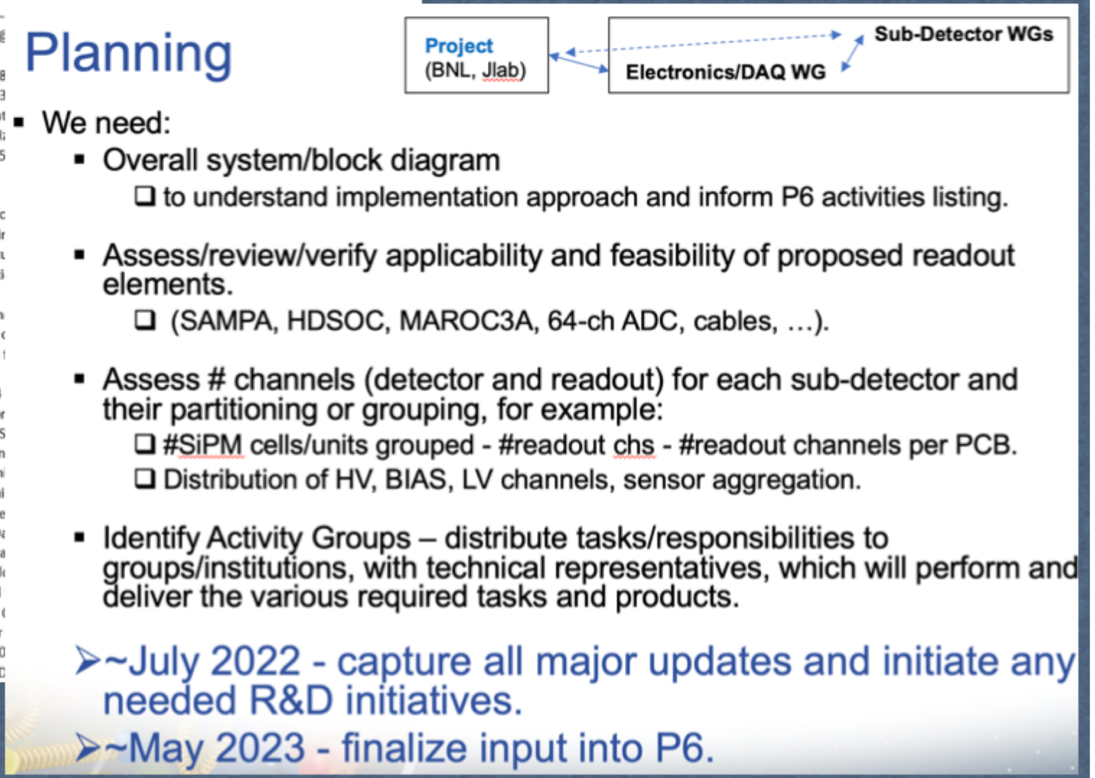
EIC Project Specifications Summary

- Preliminary specifications for MPGD & Photonic Sensors (ASICs)

Detector	
Capacitance	<200 pF nominal (500 pF maximum).
Noise	<3000 e⁻ @ 100 pF
Charge	25 fC – 100 fC (1 pC maximum).
Gain	$5x10^3 – 2x10^4$
Signal Time	100 ns – 500 ns (10 us ion drift time maximum), multiple hits per channel.
Signal Range	$<10^6$ e⁻
Rates	<2 kHz per channel.
Readout	
Attributes	Amplification, digitization and buffering.
Features	Amplitude and time per hit; waveform samples for testing and calibration functions. Zero suppression; triggerless and triggered operation.
# Channels	64
Input Impedance	<70 Ohm
Gain	2 mV/fC – 30 mV/fC, configurable.
Peaking Time	40 ns – 250 ns shaping, configurable.
Crosstalk	<1 %
ADC Resolution	12 bit (>10 bit ENOB)
TDC Resolution	<20 ns
Sampling Rate	>80 MSPS
Optional	Discriminators and scalers are desirable.
Triggering	Streaming (triggerless) readout is the default mode. Triggered operation is required for testing and calibration functions.
Pulsing	Channel group pulsing desirable for testing function.
Output	TBD. Data format to be determined and to be consistent with optical fiber data transport between FEBs and FEPs.
Control Interface	TBD. Slow controls and configuration interface to be consistent with optical fiber data transport between FEBs and FEPs.
Technology Node	65 nm CMOS or higher.
Packaging	BGA or other SMT industry standard packages.
Power	1 W ± 0.25 W or < 20 mW per channel.
Supply	<+3 V DC

Detector	
Capacitance	60 pF –
Noise	1 p.e. @
Gain	$<10^6$
Signal Time	3 ns – 8
Rise Time	1 ns – 3
Signal Range	<1 V int
Rates	<50 kH:
Bias	Vop ~ 5
Readout	
Attributes	Wavefc bufferir
Features	Amplitu operati
# Channels	64
Input Impedance	<50 Oh
Gain	1 - 10, (
Peaking Time	<40 ns !
Crosstalk	<1 %
ADC Resolution	10 - 14
TDC Resolution	1 ns for
Sampling Rate	>80 MS
Optional	Discrim
Triggering	Streami is requi
Pulsing	Channe
Output	TBD. Da fiber da
Control Interface	TBD. Sl optical
Technology Node	65 nm (
Packaging	BGA or
Power	1 W ± 0
Supply	<+3 V D

Overview slide from Marco



Planning

Project (BNL, Jlab)

Electronics/DAQ WG

Sub-Detector WGs

- We need:
  - Overall system/block diagram
    - ❑ to understand implementation approach and inform P6 activities listing.
  - Assess/review/verify applicability and feasibility of proposed readout elements.
    - ❑ (SAMPA, HDSOC, MAROC3A, 64-ch ADC, cables, …).
  - Assess # channels (detector and readout) for each sub-detector and their partitioning or grouping, for example:
    - ❑ #SiPM cells/units grouped - #readout chs - #readout channels per PCB.
    - ❑ Distribution of HV, BIAS, LV channels, sensor aggregation.
  - Identify Activity Groups – distribute tasks/responsibilities to groups/institutions, with technical representatives, which will perform and deliver the various required tasks and products.

➢~July 2022 - capture all major updates and initiate any needed R&D initiatives.

➢~May 2023 - finalize input into P6.

Past/present: sPHENIX HCal (130k channels) as test a test bench…

Future: needs a firm plan (specs/solutions) to be ready for CD 2/3 in spring/fall 2023

Streaming Readout X: Program

Streaming Readout Status

Streaming DAQ

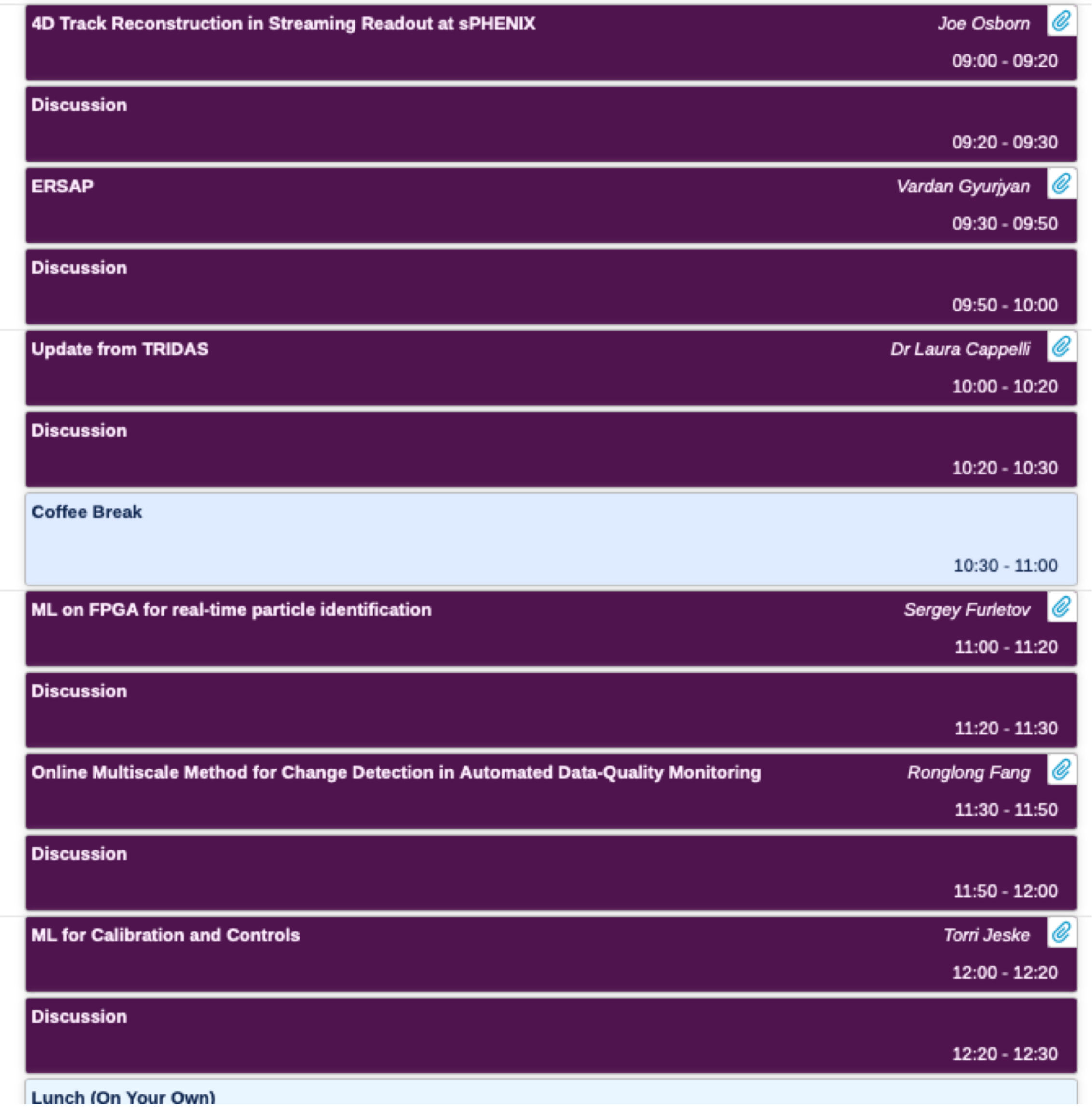
Streaming Readout Data

Streaming Readout Electronics

Streaming Readout in NHEP

Streaming Readout Community

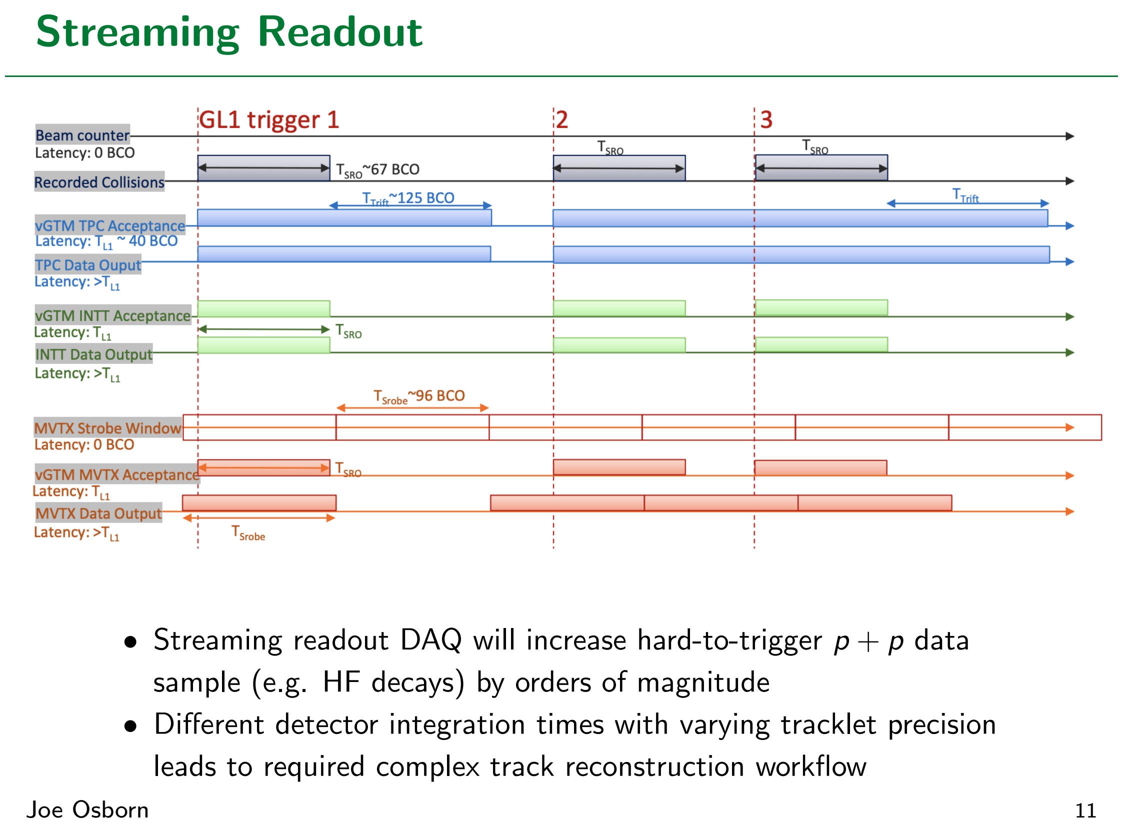
Streaming Readout in NHEP



4D Track Reconstruction in Streaming Readout at sPHENIX	*Joe Osborn*	09:00 - 09:20
Discussion		09:20 - 09:30
ERSAP	*Vardan Gyurjyan*	09:30 - 09:50
Discussion		09:50 - 10:00
Update from TRIDAS	*Dr Laura Cappelli*	10:00 - 10:20
Discussion		10:20 - 10:30
Coffee Break		10:30 - 11:00
ML on FPGA for real-time particle identification	*Sergey Furletov*	11:00 - 11:20
Discussion		11:20 - 11:30
Online Multiscale Method for Change Detection in Automated Data-Quality Monitoring	*Ronglong Fang*	11:30 - 11:50
Discussion		11:50 - 12:00
ML for Calibration and Controls	*Torri Jeske*	12:00 - 12:20
Discussion		12:20 - 12:30
Lunch (On Your Own)		

4D Track Reconstruction at sPHENIX (Joe Osborn, ORNL and BNL)

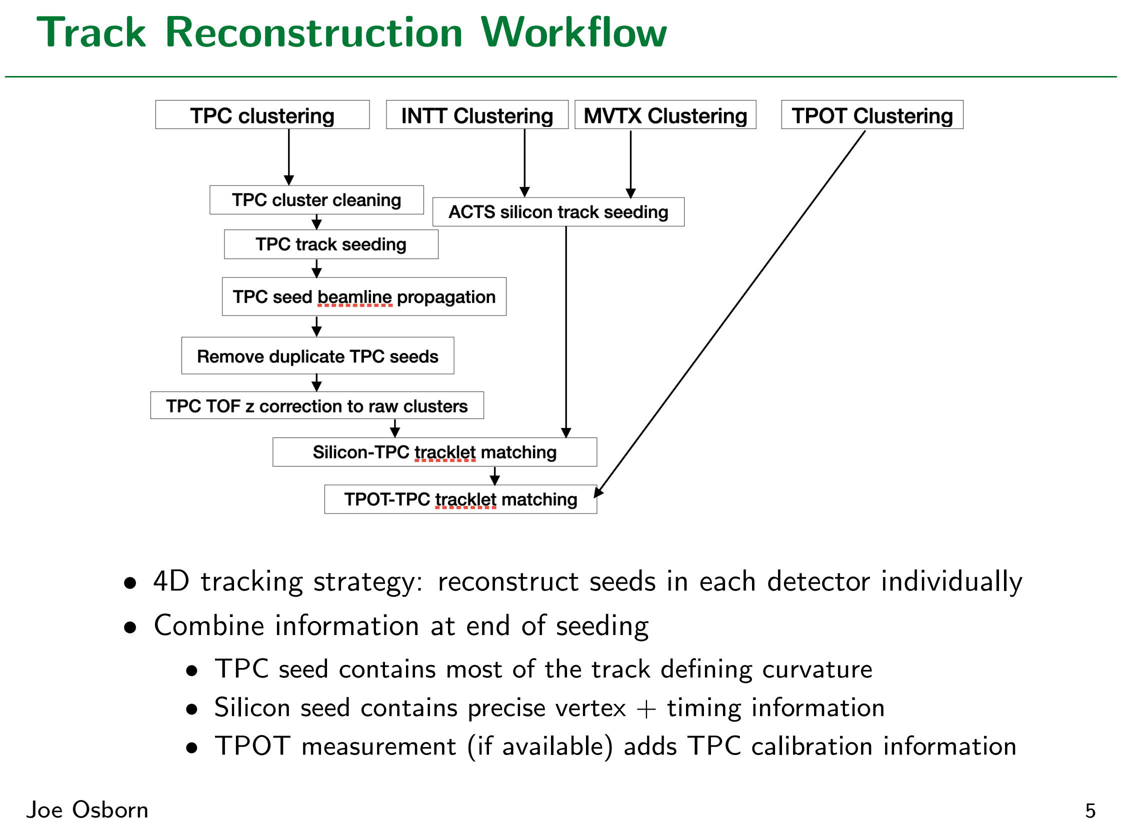
Requirement: reconstruct tracks produced up to 7μs after the trigger



Streaming Readout

- Streaming readout DAQ will increase hard-to-trigger $p+p$ data sample (e.g. HF decays) by orders of magnitude
- Different detector integration times with varying tracklet precision leads to required complex track reconstruction workflow

Joe Osborn 11



Track Reconstruction Workflow

TPC clustering
INTT Clustering
MVTX Clustering
TPOT Clustering
TPC cluster cleaning
ACTS silicon track seeding
TPC track seeding
TPC seed beamline propagation
Remove duplicate TPC seeds
TPC TOF z correction to raw clusters
Silicon-TPC tracklet matching
TPOT-TPC tracklet matching

- 4D tracking strategy: reconstruct seeds in each detector individually
- Combine information at end of seeding
  - TPC seed contains most of the track defining curvature
  - Silicon seed contains precise vertex + timing information
  - TPOT measurement (if available) adds TPC calibration information

Joe Osborn 5

Environment for Real-time Streaming, Acquisition and Processing

Vardan Gyurjyan (Jefferson Lab)

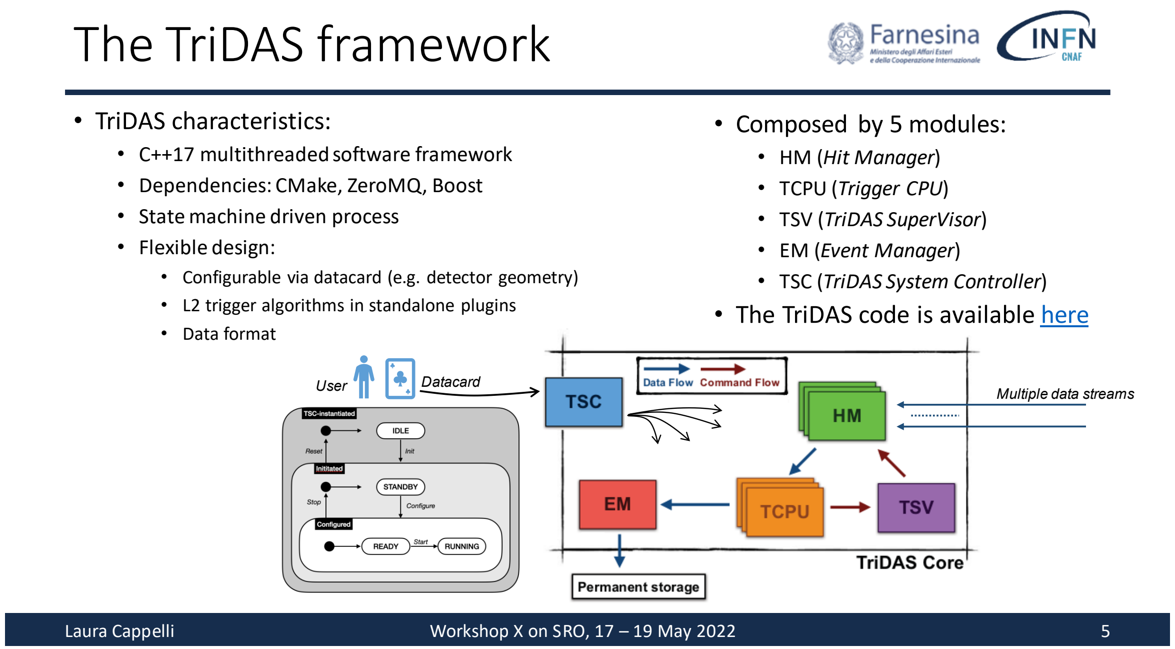
- ERSAP is a software LEGO system
  - Encourages application design based on software artifacts (LEGO bricks)
    - Easier to understand and develop
    - Reduced develop-deploy-debug cycle
    - Easy to migrate to data
    - Scales independently
    - Independent optimizations
- Improves fault isolation
- Easy to embrace hardware as well as software heterogeneity.
- Eliminates long term commitment to a single technology stack.

Agile framework that makes easy software evolution over time!

- ERSAP is a reactive actor/micro-service based data-stream processing framework. https://wiki.jlab.org/epsciwiki/index.php/ERSAP
- Combines decade-long experience: CODA, AFECS and CLARA
  - ERSAP Java binding, betta release: https://github.com/JeffersonLab/ersap-java.git
  - ERSAP C++ binding development in progress: https://github.com/JeffersonLab/ersap-cpp.git
  - ERSAP Python binding in the design stage
  - Plans to design ERSAP Julia binding
- Many ERSAP engine development projects are in progress
  - CODA engines: https://github.com/JeffersonLab/ersap-coda.git
  - JANA2 based engines: https://github.com/JeffersonLab/ersap-jana.git
  - TriDAS engines: https://github.com/JeffersonLab/ersap-tridas.git
  - CLAS12 AI reconstruction engines https://github.com/JeffersonLab/ersap-vtp.git
  - INDRA ASTRA project ML engines
- Collaborative effort between JLAB Physics and CST divisions.

TriDAS (Laura Cappelli, INFN-CNAF)

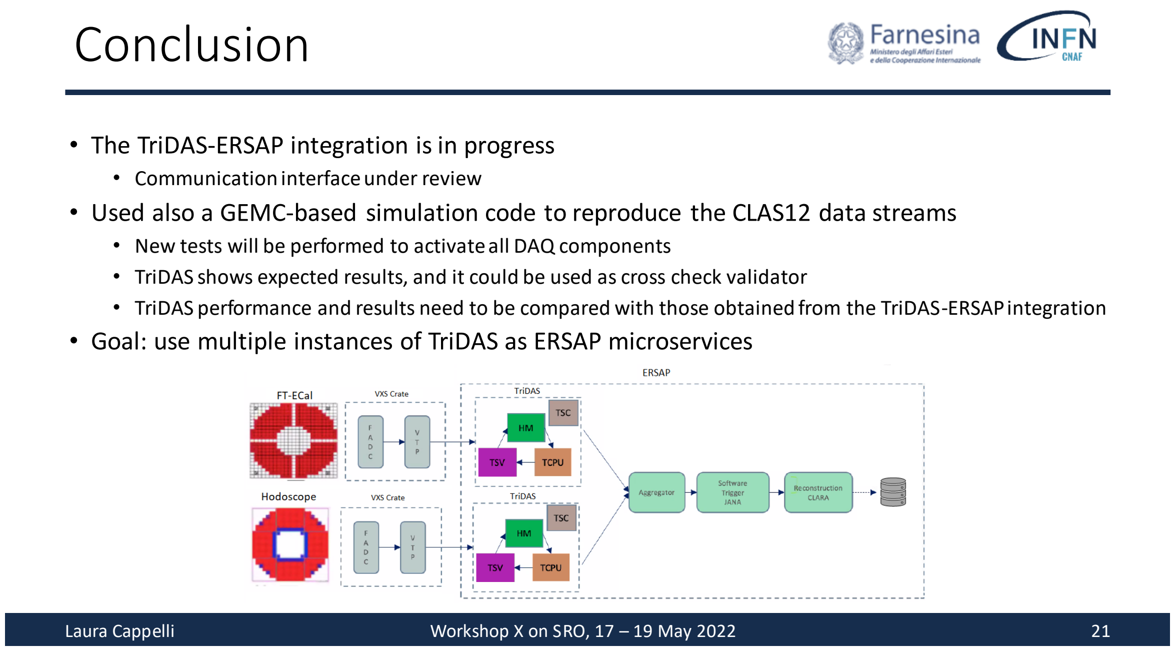
Streaming DAQ from astroparticle physics community (NEMO, KM3NeT-ITA)



The TriDAS framework

- TriDAS characteristics:
  - C++17 multithreaded software framework
  - Dependencies: CMake, ZeroMQ, Boost
  - State machine driven process
  - Flexible design:
    - Configurable via datacard (e.g. detector geometry)
    - L2 trigger algorithms in standalone plugins
    - Data format
- Composed by 5 modules:
  - HM (*Hit Manager*)
  - TCPU (*Trigger CPU*)
  - TSV (*TriDAS SuperVisor*)
  - EM (*Event Manager*)
  - TSC (*TriDAS System Controller*)
- The TriDAS code is available here

Laura Cappelli Workshop X on SRO, 17 – 19 May 2022 5



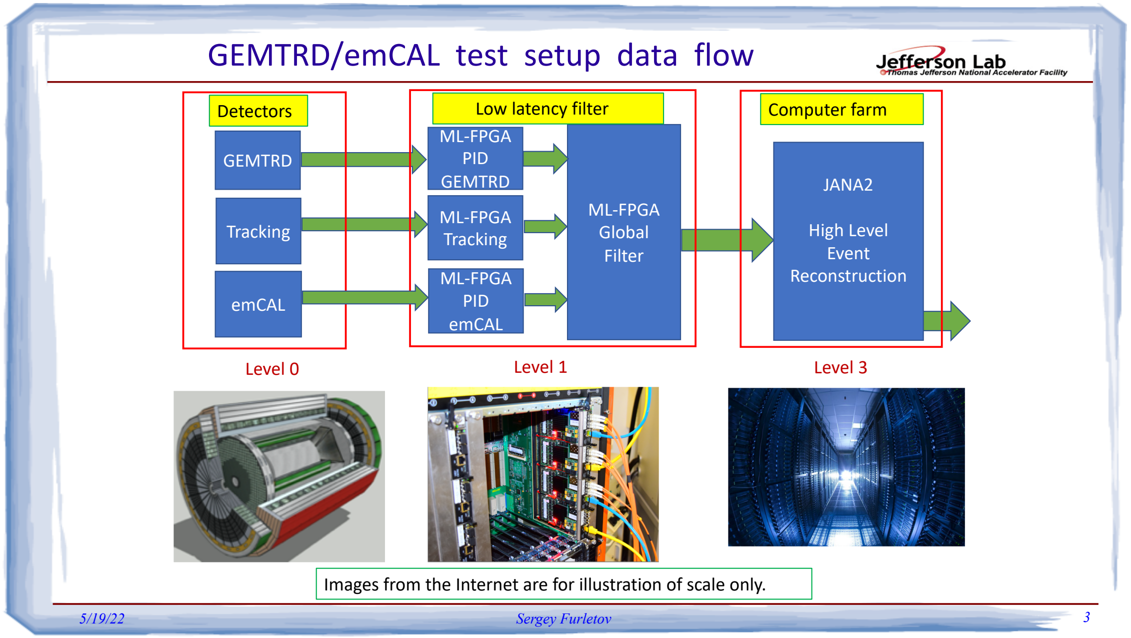
Conclusion

- The TriDAS-ERSAP integration is in progress
  - Communication interface under review
- Used also a GEMC-based simulation code to reproduce the CLAS12 data streams
  - New tests will be performed to activate all DAQ components
  - TriDAS shows expected results, and it could be used as cross check validator
  - TriDAS performance and results need to be compared with those obtained from the TriDAS-ERSAP integration
- Goal: use multiple instances of TriDAS as ERSAP microservices

Laura Cappelli Workshop X on SRO, 17 – 19 May 2022 21

ML on FPGA for real-time particle identification

Sergey Furletov (Jefferson Lab)

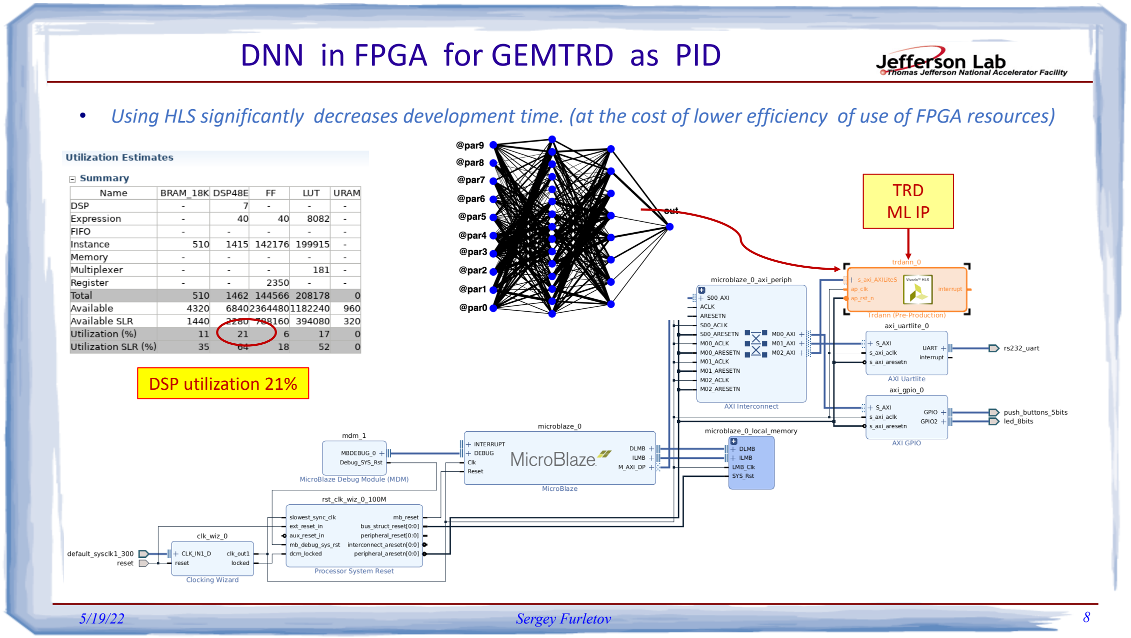


GEMTRD/emCAL test setup data flow

Jefferson Lab
Thomas Jefferson National Accelerator Facility

Images from the Internet are for illustration of scale only.

5/19/22 Sergey Furletov 3



DNN in FPGA for GEMTRD as PID

Jefferson Lab
Thomas Jefferson National Accelerator Facility

- *Using HLS significantly decreases development time. (at the cost of lower efficiency of use of FPGA resources)*

Utilization Estimates

⊟ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	7	-	-	-
Expression	-	40	40	8082	-
FIFO	-	-	-	-	-
Instance	510	1415	142176	199915	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	181	-
Register	-	-	2350	-	-
Total	510	1462	144566	208178	0
Available	4320	6840	2364480	1182240	960
Available SLR	1440	2280	788160	394080	320
Utilization (%)	11	21	6	17	0
Utilization SLR (%)	35	64	18	52	0

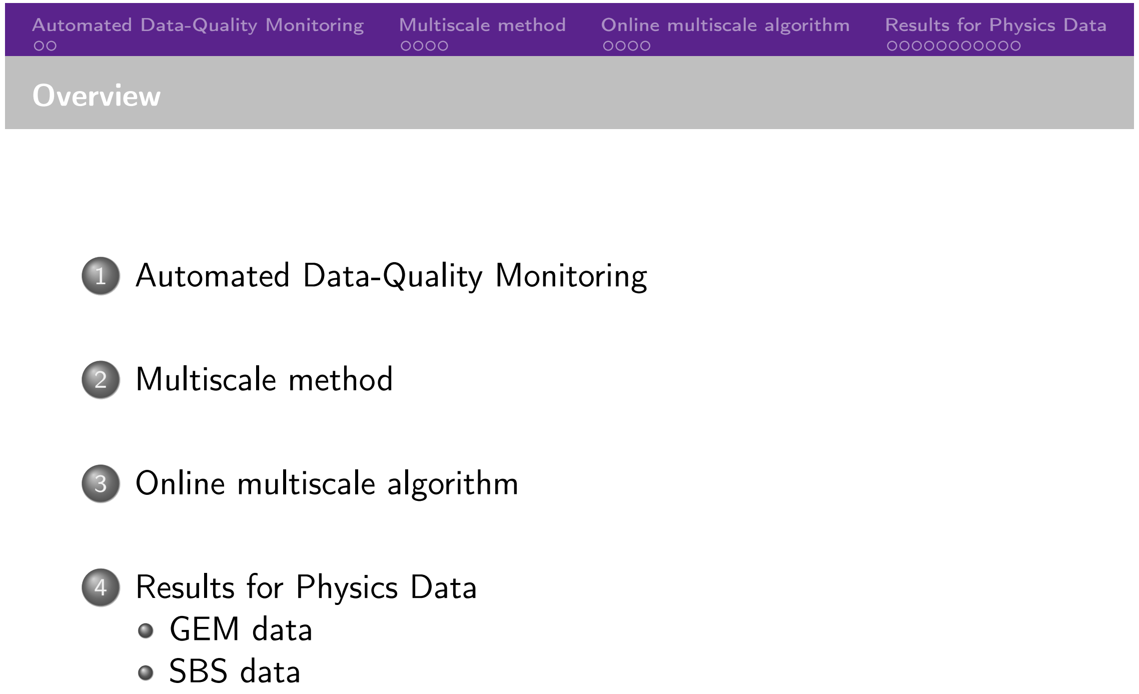
DSP utilization 21%

TRD ML IP

5/19/22 Sergey Furletov 8

Online Multiscale Method for Change Detection in Automated Data-Quality Monitoring

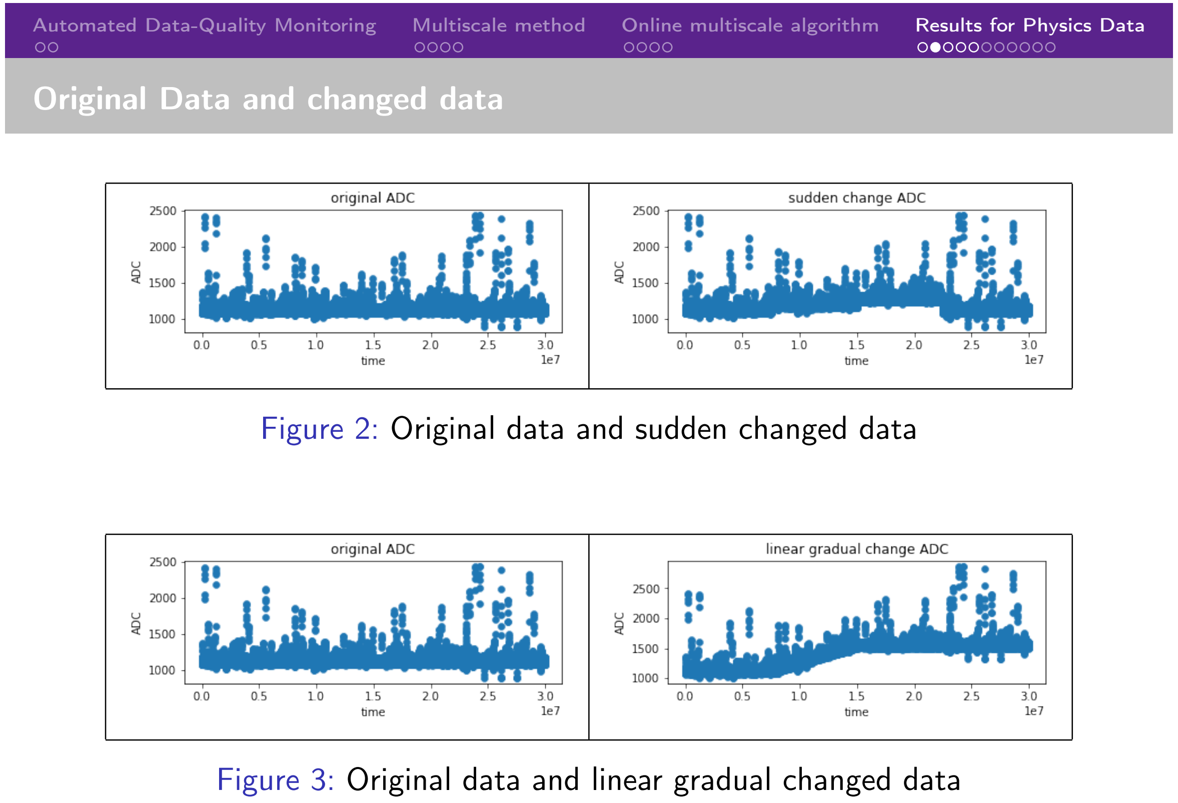
Ronglong Fang (ODU)



Automated Data-Quality Monitoring ○○ Multiscale method ○○○○ Online multiscale algorithm ○○○○ Results for Physics Data ○○○○○○○○○○○

Overview

1. Automated Data-Quality Monitoring
2. Multiscale method
3. Online multiscale algorithm
4. Results for Physics Data
   - GEM data
   - SBS data



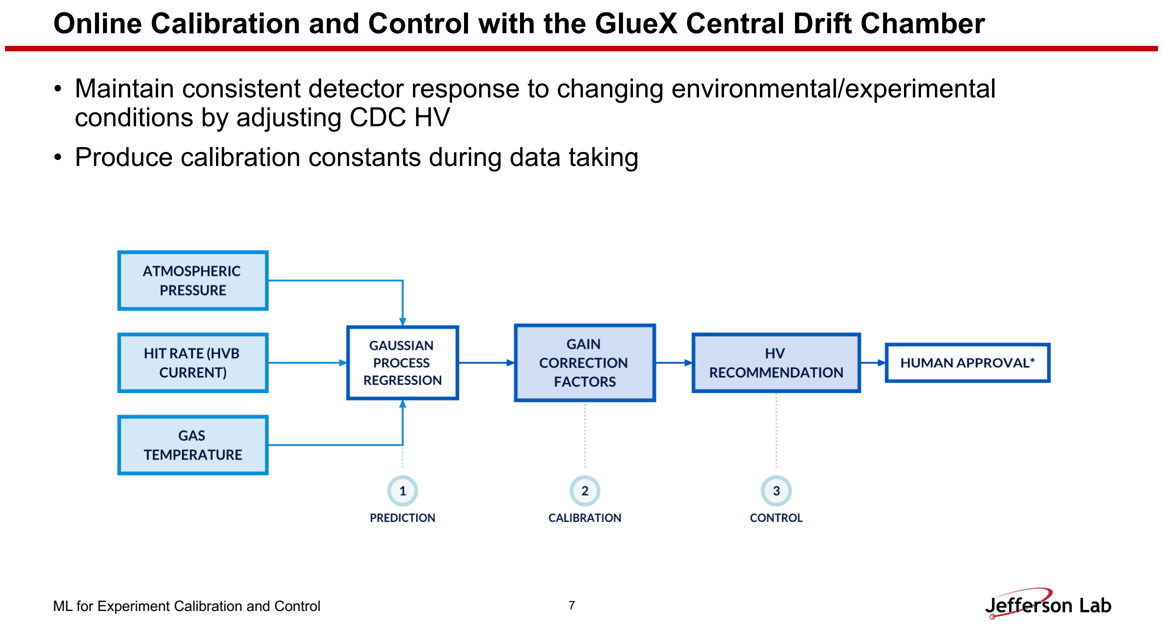
Automated Data-Quality Monitoring ○○ Multiscale method ○○○○ Online multiscale algorithm ○○○○ Results for Physics Data ○●○○○○○○○○○

Original Data and changed data

Figure 2: Original data and sudden changed data

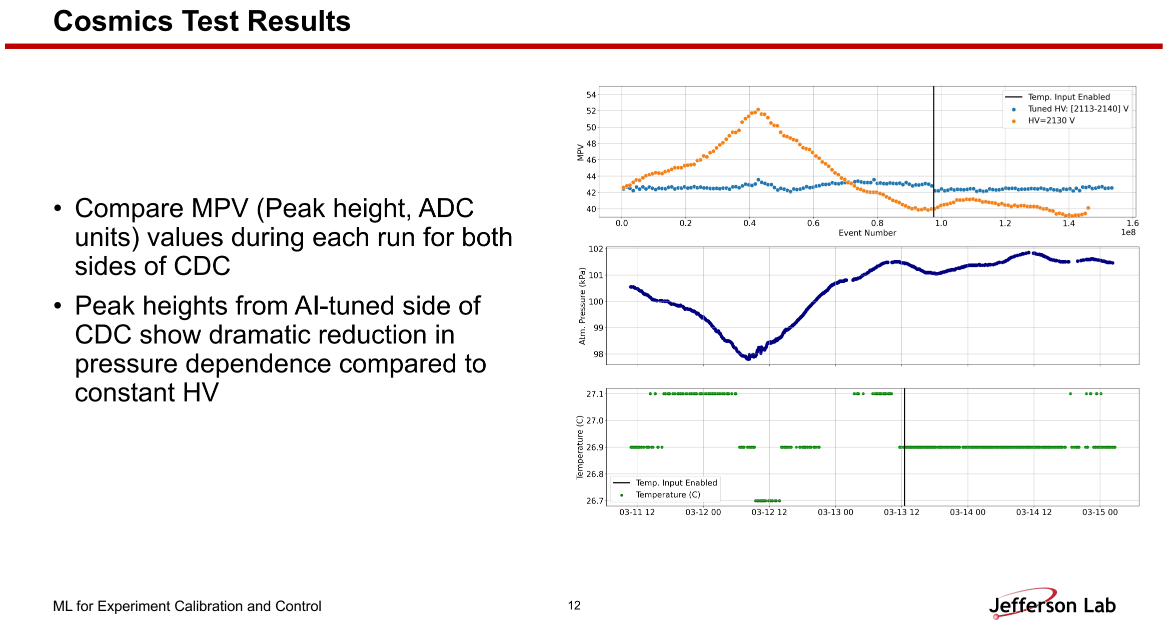
Figure 3: Original data and linear gradual changed data

ML for Experiment Calibration and Control (Torri Jeske, Jefferson Lab)



Online Calibration and Control with the GlueX Central Drift Chamber

- Maintain consistent detector response to changing environmental/experimental conditions by adjusting CDC HV
- Produce calibration constants during data taking



Cosmics Test Results

- Compare MPV (Peak height, ADC units) values during each run for both sides of CDC
- Peak heights from AI-tuned side of CDC show dramatic reduction in pressure dependence compared to constant HV

Streaming Readout X: Program

Streaming Readout Status

Streaming DAQ

Streaming Readout Data

Streaming Readout Electronics

Streaming Readout in NHEP

Streaming Readout Community

Group Discussion

- **Do we agree on a coordinated effort on a streaming readout system in the NP community?**
  - Can we agree on a baseline / common system?
  - Will we have common services for the front end? E.g., power distribution scheme.
- **Timeline for design and development of a streaming readout system for the EIC**
  - Is there any need for R&D?
  - Can we build a simple test setup? How will we scale it up? Can we use it for test beam?
  - How is the stream aggregation done?
  - How are we building events? Do we need to build events online?
  - Generalize electrical - optical interface
  - What protocols are used for the DAQ?
  - Hardware and software (data handling, communication; calibrations, reconstruction, analysis)
  - We need good simulations of the entire data stream (emphasis on digitization)
  - Interface of streaming readout and experimental control, including slow control, and also accelerator control
  - Is there a requirement for analysis-ready data from the beamline?
  - What computing resources are required directly at the experiments?
  - How will we handle firmware and software updates?

Group Discussion

- **What are the biases in the design and implementation and how to prevent them?**
- **We need data quality monitoring for each layer of the read out and data processing, including feedback for accelerator control**.
- **What are the computing resources needed for the streaming readout?**
  - What are the available and affordable resources?
- **How can we manage background and noise reliably?**
- **For each detector component**: How will we handle calibrations? Do we need a triggered system for calibrations? What are the requirements for calibrations? What would be the required turnaround time for calibrations?
- **FPGA**:
  - Early aggregation: Is there a need for data processing before the frontend?
  - Do we want FPGAs at the frontend? What are the limitations and challenges?
- **How do we coordinate the purchase of front-end electronics?**
- **How do we coordinate the purchase of other components, e.g., GPUs?**
- **We have to define the clock distribution. How will it be done?**
  - Timing system needs to allow for simultaneous test of the detector components.
- **What are the boundaries between DAQ, online and offline data processing? Will this be fully integrated?**

Streaming Readout X

Markus Diefenthaler

mdiefent@jlab.org

- Workshop on streaming readout R&D for the EIC and other NP experiments
- Workshop report in preparation
- Next workshop foreseen for November and December 2022

Backup

Benefits of Heterogeneous Hardware to Nuclear Physics

- GFlop/Watt is significantly lower for GPUs than CPUs
  - *e.g. https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/*
- Price per flop is lower for GPUs than CPUs
  - *n.b. can be hard though to keep GPU fully busy*
- Large HPC/HTC systems have significant compute capability tied up in heterogeneous hardware *(including cloud services)*
- Higher memory bandwidth *(good for streaming)*
  - See LHCb Allen project: https://arxiv.org/abs/1912.09161
- Well-suited for AI/ML models
  - *not all models are efficient on GPUs, but some (e.g. CNNs) are extremely efficient*
  - *tools like HLS4ML making FPGAs more accessible (https://fastmachinelearning.org/hls4ml/)*
- Faster simulation via GANs

Slide courtesy David Lawrence (JLab)