



PixESL at MAPS 2026

Francesco E. Brambilla

francesco.enrico.brambilla@cern.ch

SLAC, Stanford, USA

27-4 Jul-Aug 2026

WP5 IC Tech
EP R&D

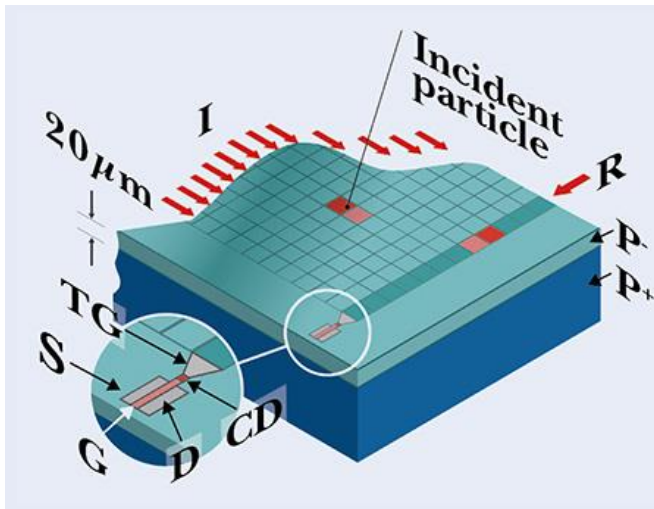
Introduction

1. Overview
2. ESL design
3. Electronic design flow
4. PixESL in simulation flow

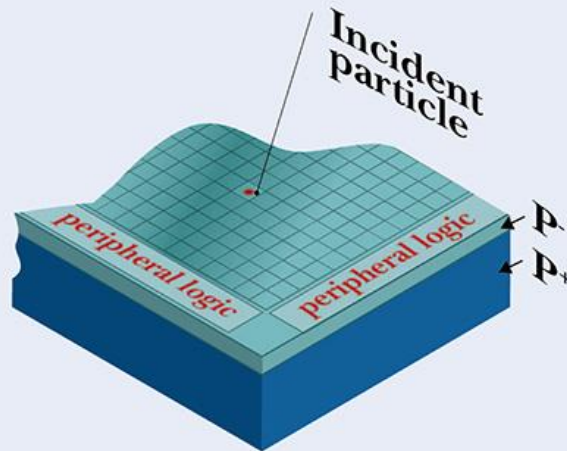
Overview – Pixel detectors

- Pixel detectors measure the position and charge of the particle interactions
- Various types:
 - Charge-coupled devices: charge is collected on pixels and then shifted to the periphery.
 - Monolithic Active Pixel Sensors: per-pixel amplification and measurement.
 - Hybrid detectors: sensing substrate bonded to readout chip.

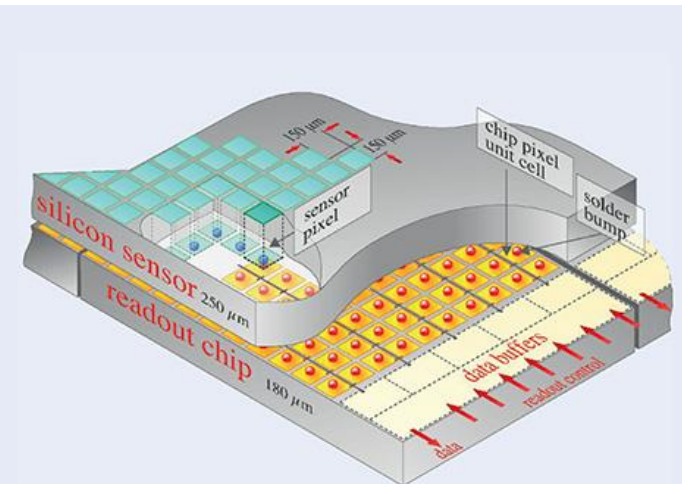
CCD



MAPS



Hybrid



Overview – Detector Specifications & Prototyping

- **The latest generations of pixel detectors push towards higher performance in:**
 - Pixel pitch and count, sustainable particle rate, charge and timestamping resolution...
- **The performance targets are defined by the physics applications:**
 - How can we draw a set of specifications matching that performance?
 - How do we validate those specifications against the application?
- **Electronic System-Level (ESL) modeling can guide us to write an abstract model of the detector to prototype and study its performance in the early design phase.**

Overview - PixESL



Project

- **PixESL: Pixel detector virtual Electronic System-Level** prototyping framework
- GitLab repositories: [PixESL core](#) & [PixESL4school](#)



Context

- [EP R&D Programme](#) on Technologies for Future Experiments: [IC Technology Work-Package \(WP5\)](#)
- Used to prototype MAPS and hybrids detectors at CERN.



Publications

- [2024] Paper: [Virtual Prototyping Framework for Pixel Detector Electronics in High Energy Physics](#)
- [2024] Paper: [SystemC framework for architecture modelling of electronic systems in future particle detectors](#)

Electronic System Level – ESL design

ESL is a methodology for **modelling** and **simulating complex electronic systems**.

Key points:

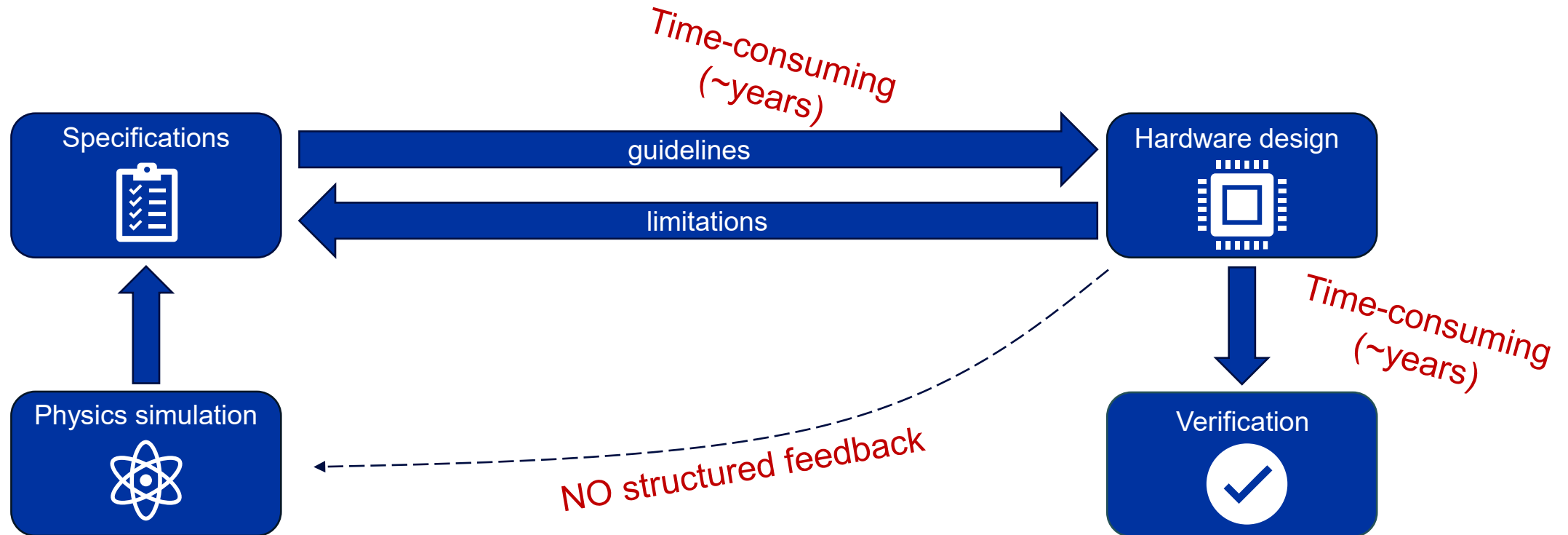
- **Abstraction**
- **System-level perspective**
- **Simulation and analysis**
- **Design space exploration**
- **Validation**



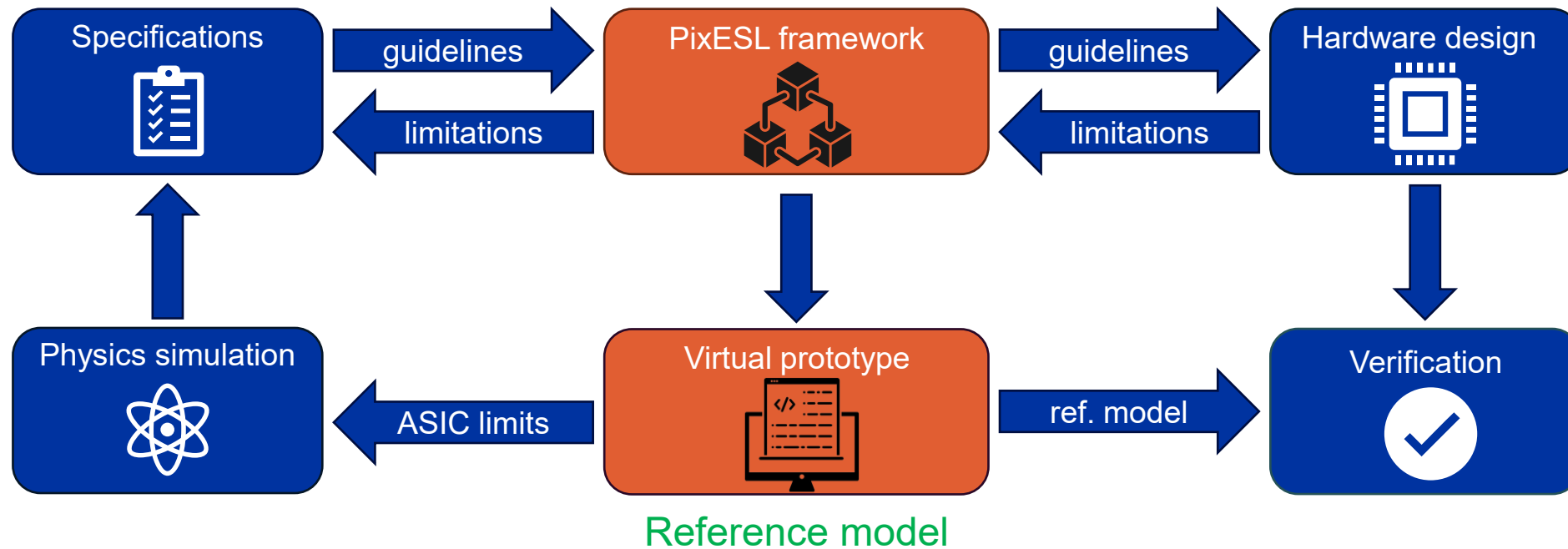
PixESL

Pixel detectors virtual **E**lectronic **S**ystem-**L**evel
prototyping framework

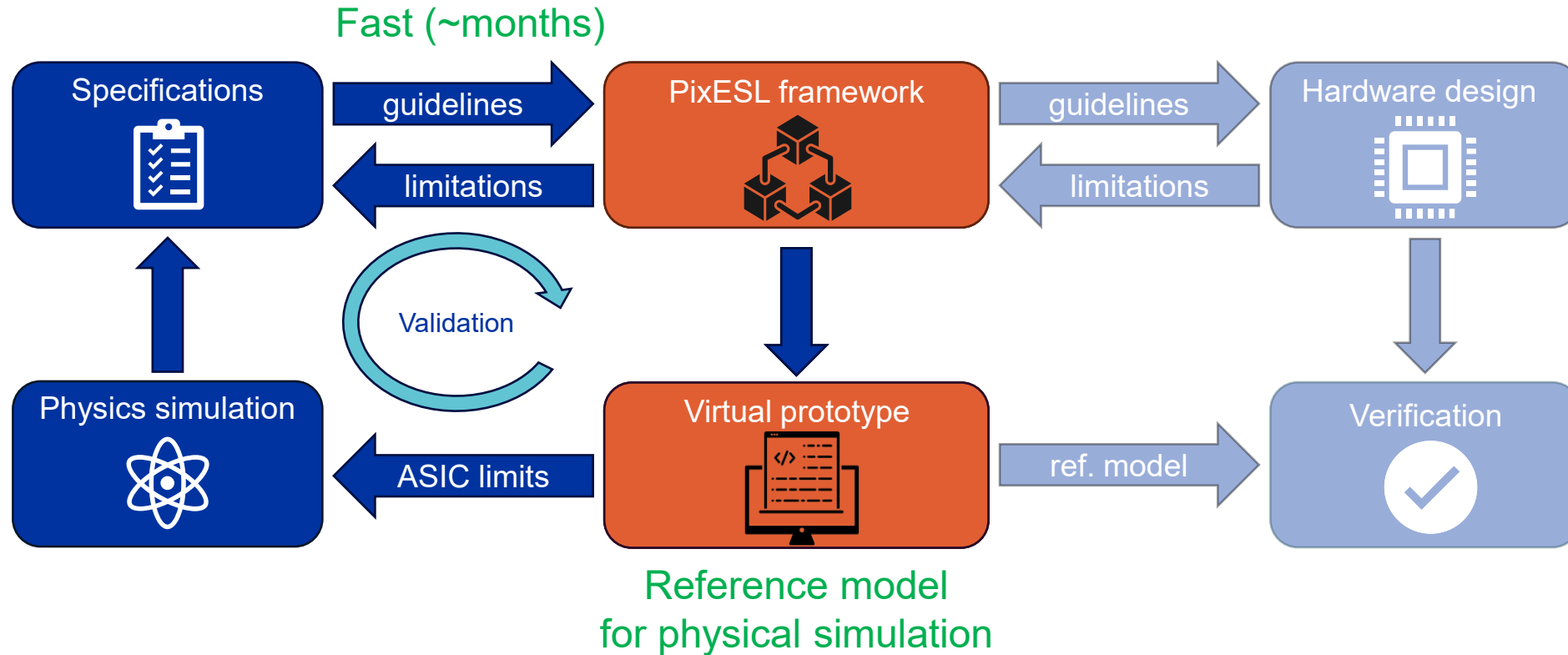
Electronics design flow



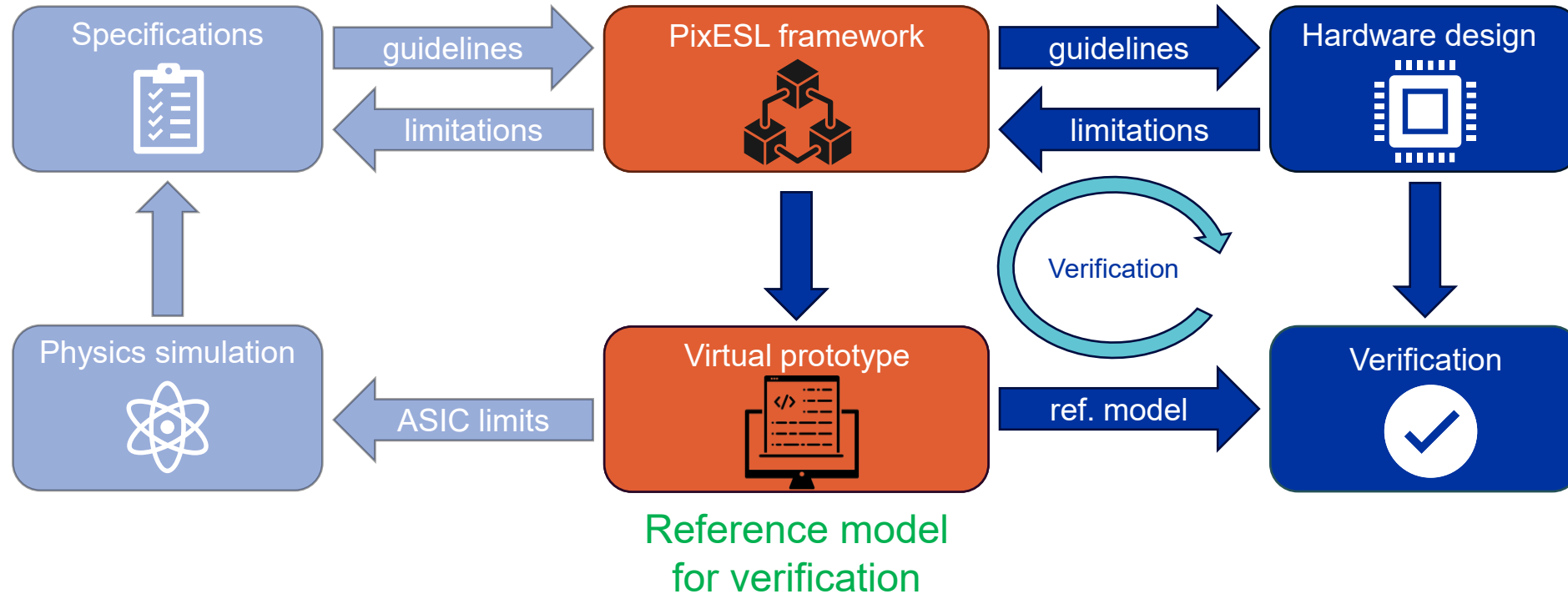
Electronics design flow with ESL prototype



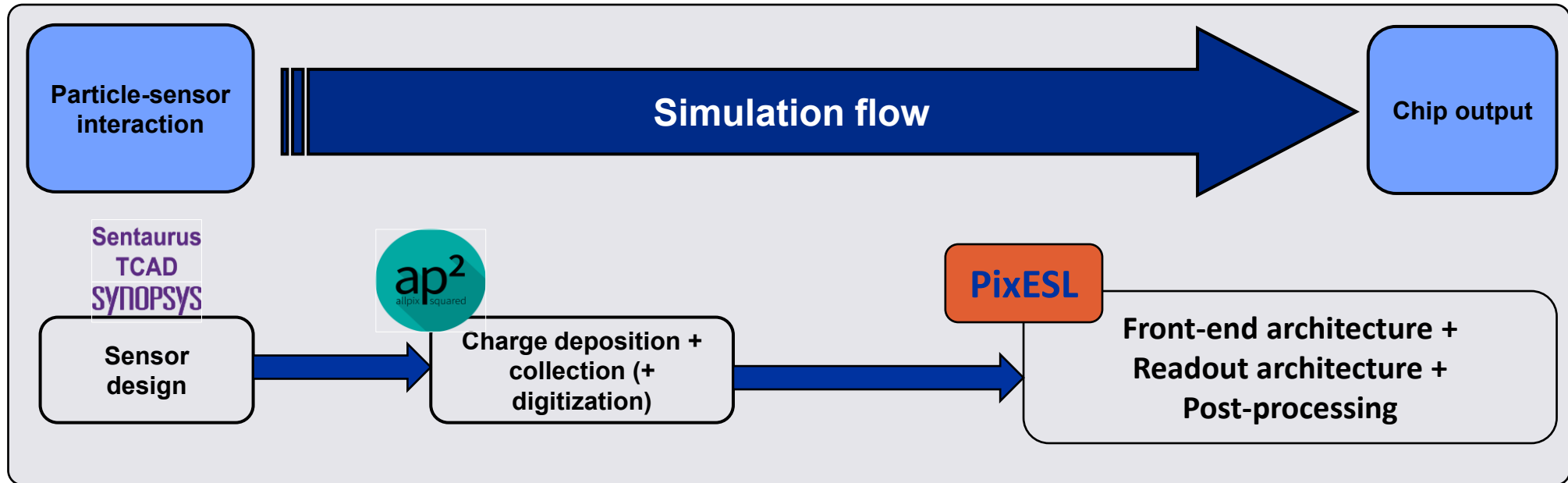
Electronics design flow with ESL prototype - Validation



Electronics design flow with ESL prototype - Verification



PixESL in simulation flow



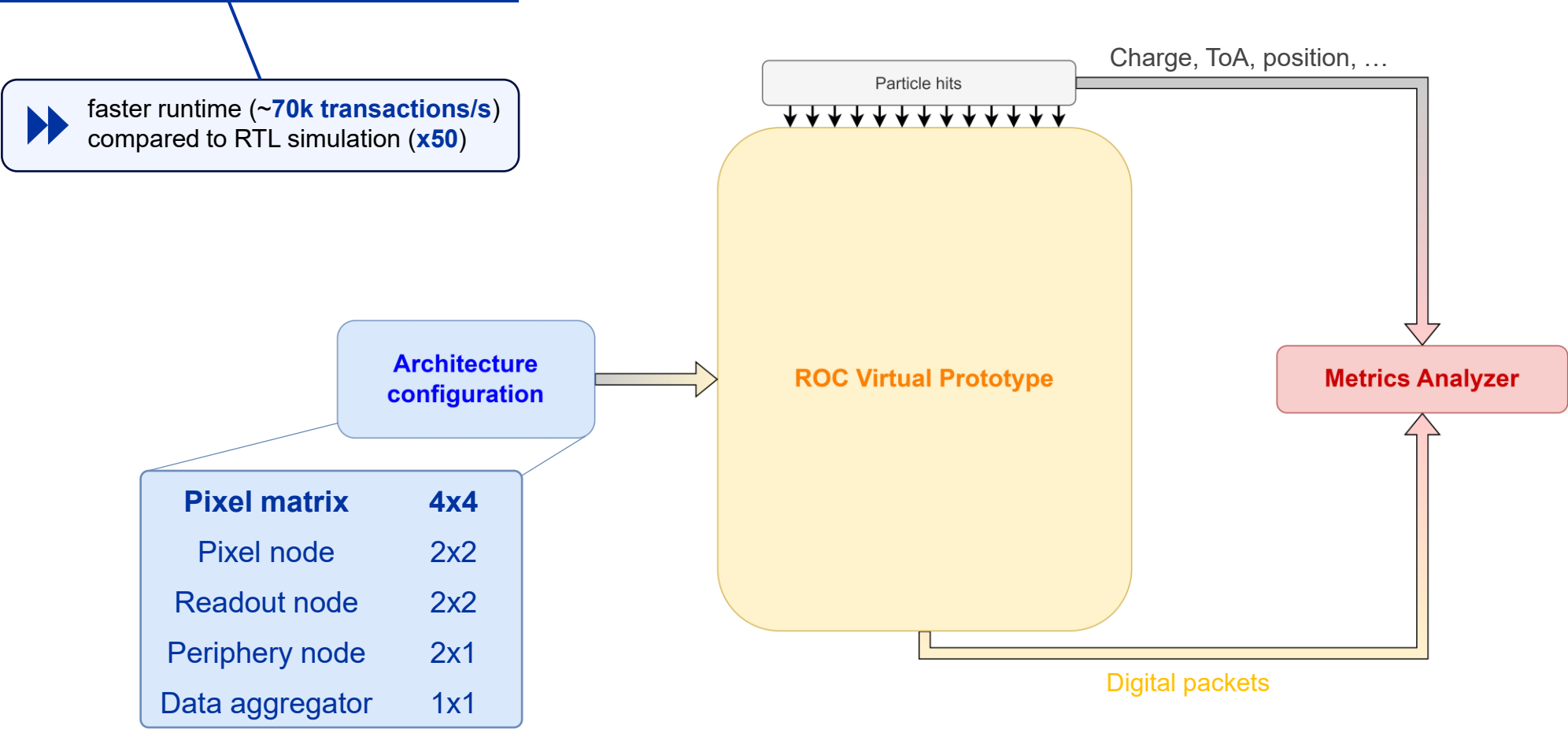
- **TCAD** focuses on simulating the detailed structure and fabrication of semiconductor sensor devices
- **Allpix²** focuses on simulating the physical processes in silicon detectors
- **PixESL** focuses on modelling the behaviour of pixel detector circuits

PixESL framework

1. Virtual Prototyping
2. Modeling Methodology
3. Front-End model
4. Readout model
5. PixESL in a nutshell

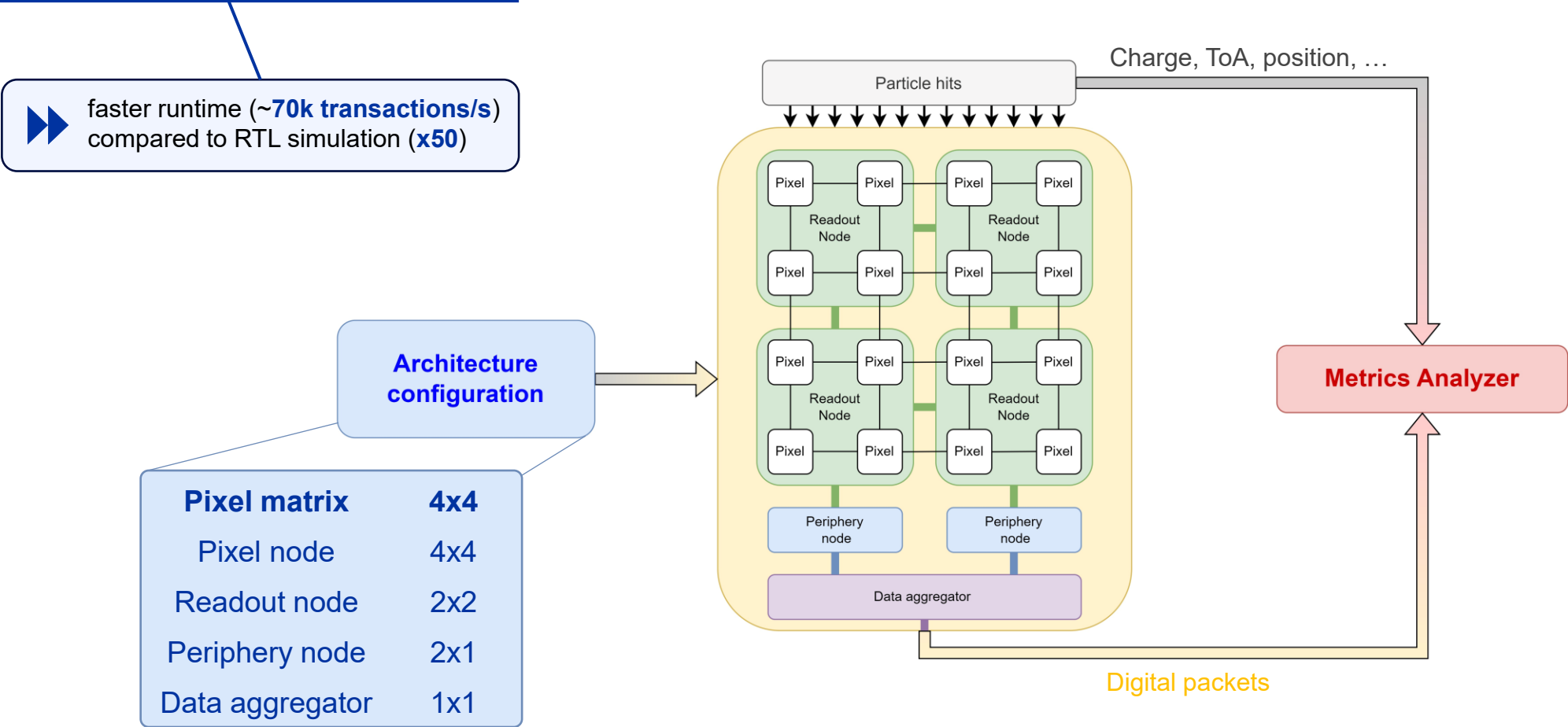
PixESL – Virtual Prototyping

PixESL applies the **Electronic System Level** concepts to **pixel detectors** design flow for a **cycle-accurate** simulation from **analog front-end** to **chip/system readout**.



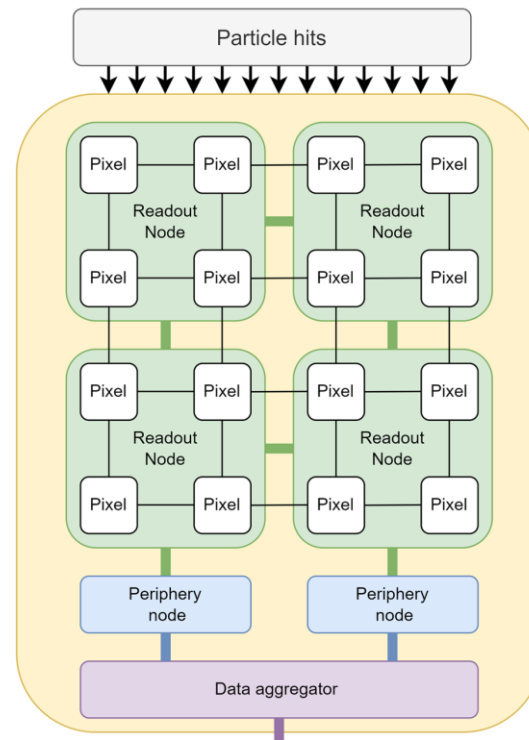
PixESL – Virtual Prototyping

PixESL applies the **Electronic System Level** concepts to **pixel detectors** design flow for a **cycle-accurate** simulation from **analog front-end** to **chip/system readout**.



PixESL – Virtual Prototyping

PixESL applies the **Electronic System Level** concepts to **pixel detectors** design flow for a **cycle-accurate** simulation from **analog front-end** to **chip/system readout**.



Detector Simulation e.g. AllPix2, Geant4,..., provides pixel hit input data with charge and time of injection

ROC Virtual Prototype in PixESL provides a C++-based high-level simulation platform to study the entire electronics system chain at an early stage of development.

PixESL – Modeling Methodology

Pixel [C++/SystemC]

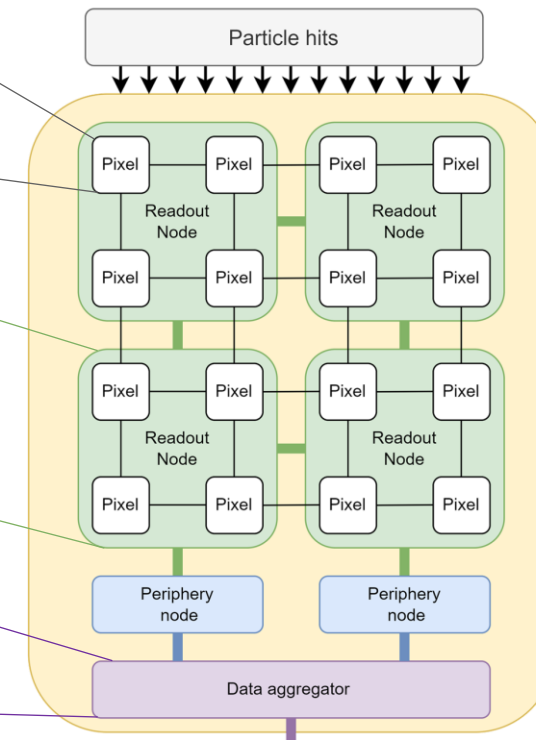
- Analog Front-End
- Digital Front-End

Readout & Periphery [SystemC TLM 2.0]

- Readout network
- Output link

Data processing [SystemC TLM 2.0]

- Ex: Event aggregator



Detector Simulation e.g. AllPix2, Geant4,..., provides pixel hit input data with charge and time of injection

ROC Virtual Prototype in PixESL provides a C++-based high-level simulation platform to study the entire electronics system chain at an early stage of development.

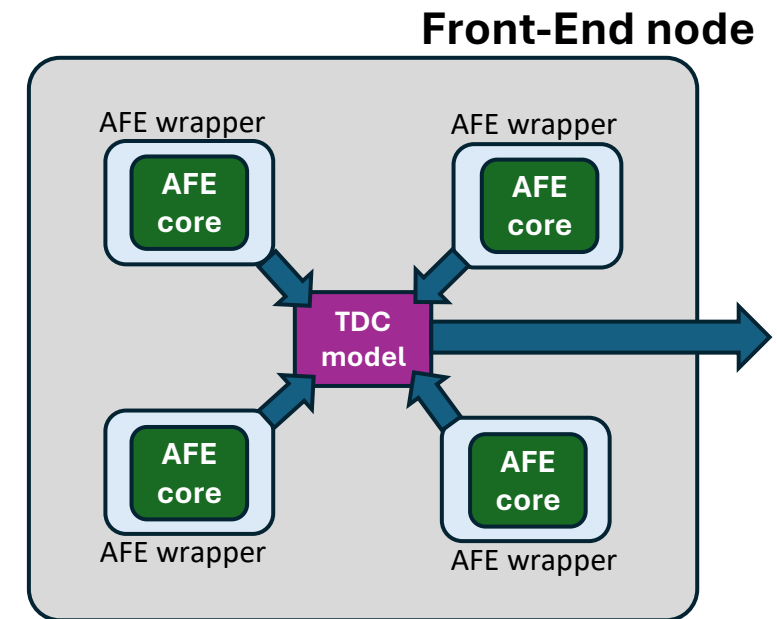
Readout Chip output data with performance metrics

PixESL – Front-End model

Modular design approach: each module can be easily changed to model different functionality.

Modeled with **C++** and **SystemC** specialized classes:

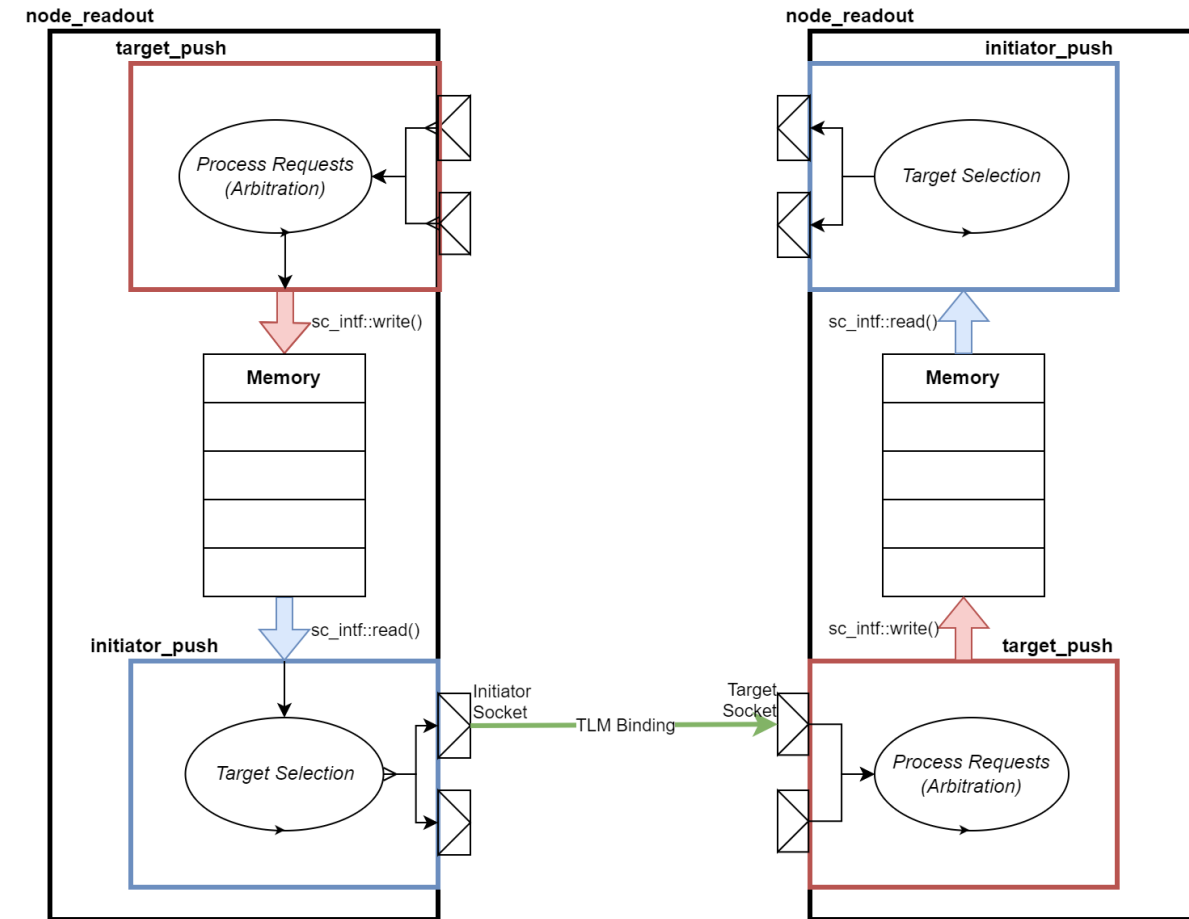
- **Analog Front-End core [C++]**
analytical model of the analog circuits
- **AFE Wrapper [C++ → SystemC]**
translates the AFE outputs into SystemC events
- **TDC model [SystemC]**
formats the pixel packet and routes it outside the node



PixESL – Readout model

Modeled with **SystemC** Transaction Level Modeling 2.0 (**TLM 2.0**):

- Each node contains:
 - **Initiator**: Selects a target and transmits packets to other nodes
 - **Target**: arbitrates incoming packets
 - **Memory**: buffers packets between init. and target
- **Packet transfer protocol** implemented with **TLM 2.0 library**
- Nodes can be **connected arbitrarily** and their **functionality customized**



PixESL in a nutshell

Open-source

- The model is based on **C++** and  [\[link\]](#)
- Performance analysis is based on **Python**

User-friendly

- **Reconfigurable architectures** and **network**
- **User** and **developer** roles are separated

Reusable

- **Generalized** layers and **standardized** packet transport
- A **library** of useful layers and components
- **Validation** of the specifications and **verification** of the hardware design

Applications at CERN

1. LHCb VeLo Upgrade II (hybrid data-driven detector)
2. ALMIRA Alice3 Upgrade (Monolithic tracker)

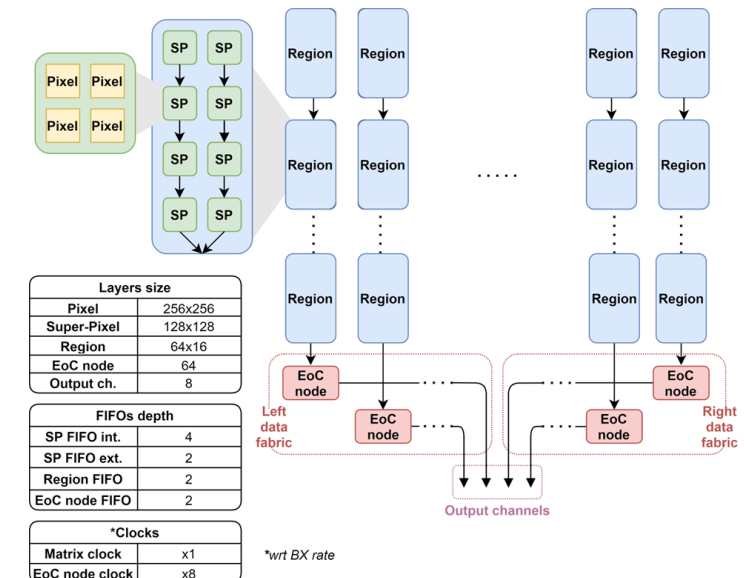
PixESL for LHCb VELO Upgrade II

The Readout Chip for the LHCb tracker upgrade II is the prototype of **Velopix2**:

- **Data-driven** readout architecture
- Pixel hit rate **3.5 GHz/cm²**
- **28 nm** CMOS technology
- Design derived from Velopix and Timepix4
- **Time resolution 25 ps**, 50 μm pixel pitch

PixESL model:

- The **pixel** emulates the **Analog Front-End model**
- **Super-Pixel**, **region** and **EoC node** are digital readout stages
- The simulation is based on **physical events**
- The **geometry** of the architecture is **user-configurable**

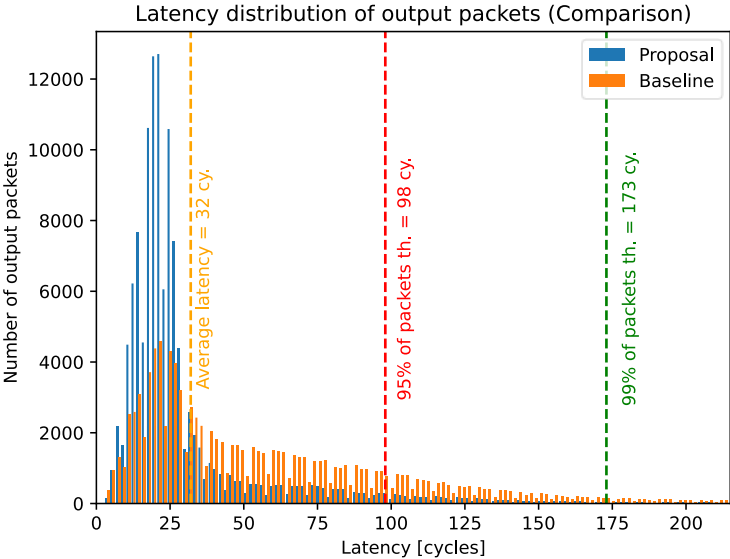
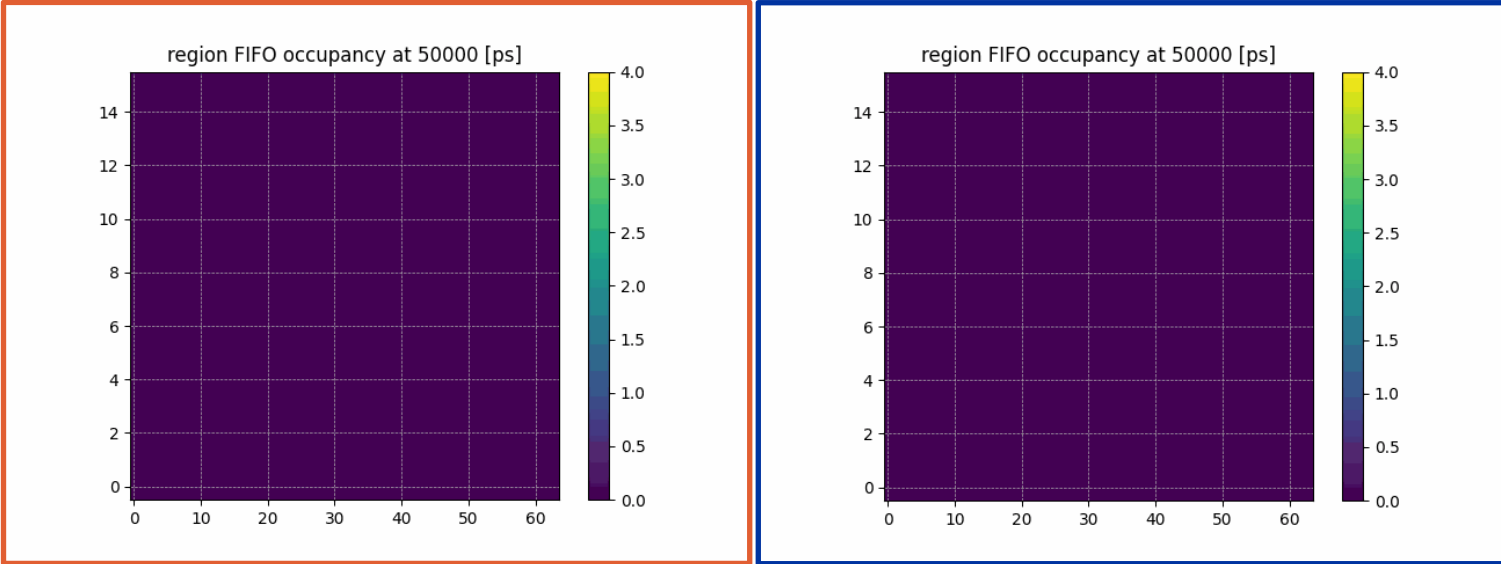


PixESL for LHCb VELO Upgrade II - Results

	Velopix	Proposed
Pixel	256x256	256x256
SP	128x128	128x128
Regions	64x8	128x8 *
EoC	64 (8 ch.)	128 (16 ch.)

Arbitration	Round-robin	Back-pressure
Readout eff.	90 %	99 %
Avg. latency	~100 cy.	~30 cy.

*By doubling the transactions per cycle

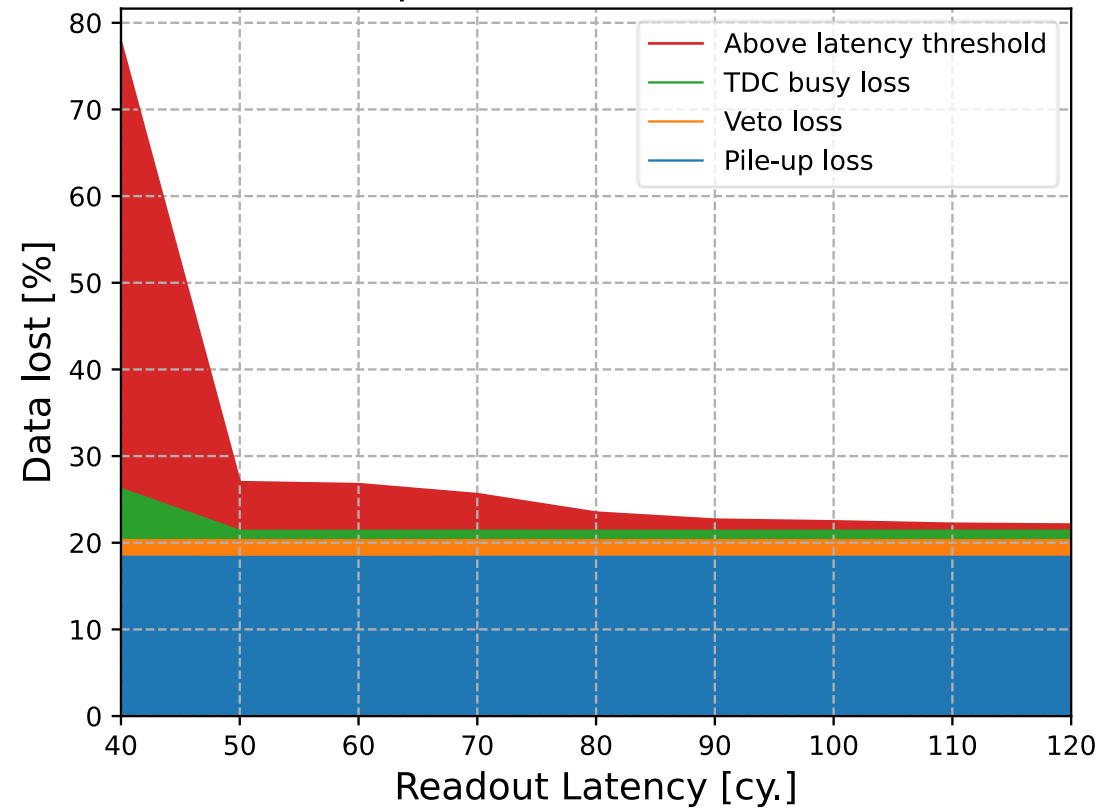


PixESL for LHCb VELO Upgrade II - Results

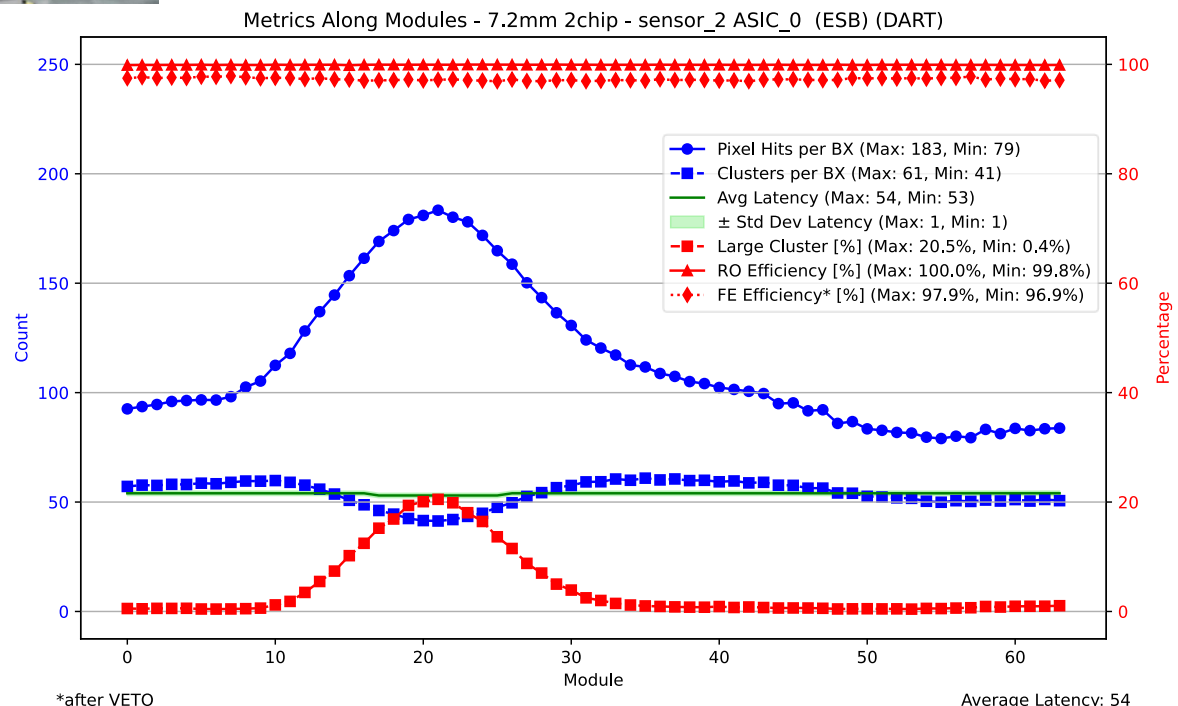
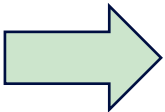
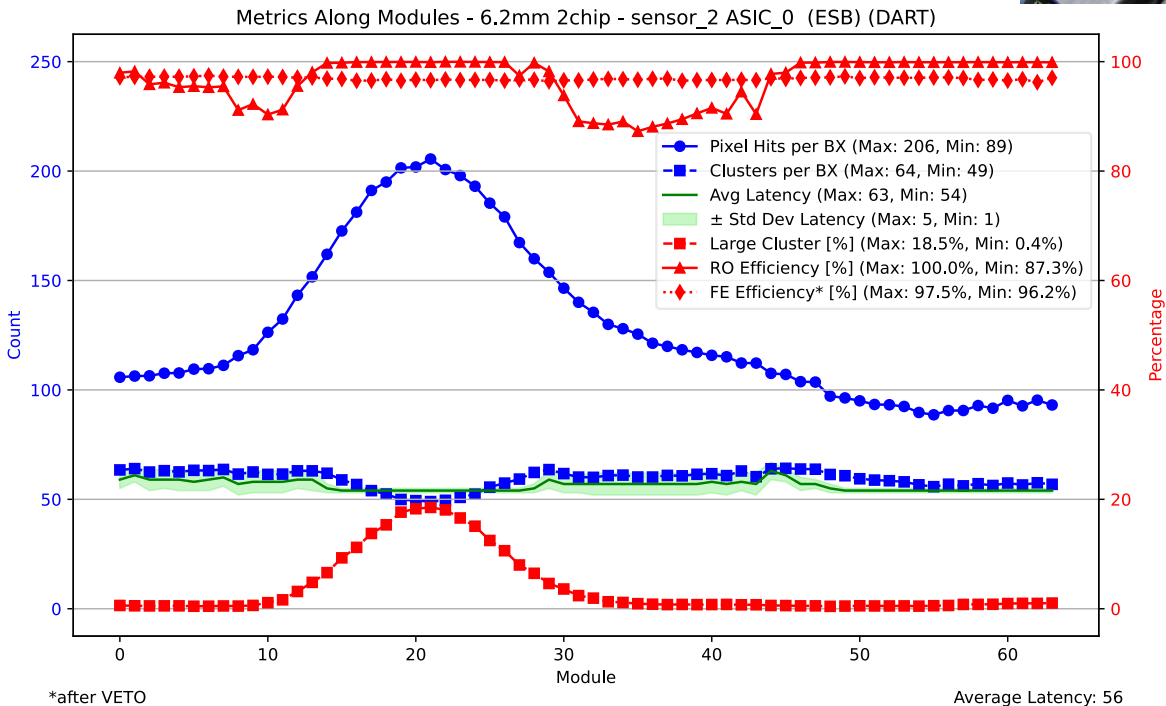
PixESL allows to perform design space exploration to:

- analyze **bottlenecks**
- separate **bugs** from **system inefficiencies**

Parametric Sweep : Data Losses vs Readout Latency



PixESL for LHCb VELO Upgrade II - Results



PixESL for ALICE3 Tracker Upgrade

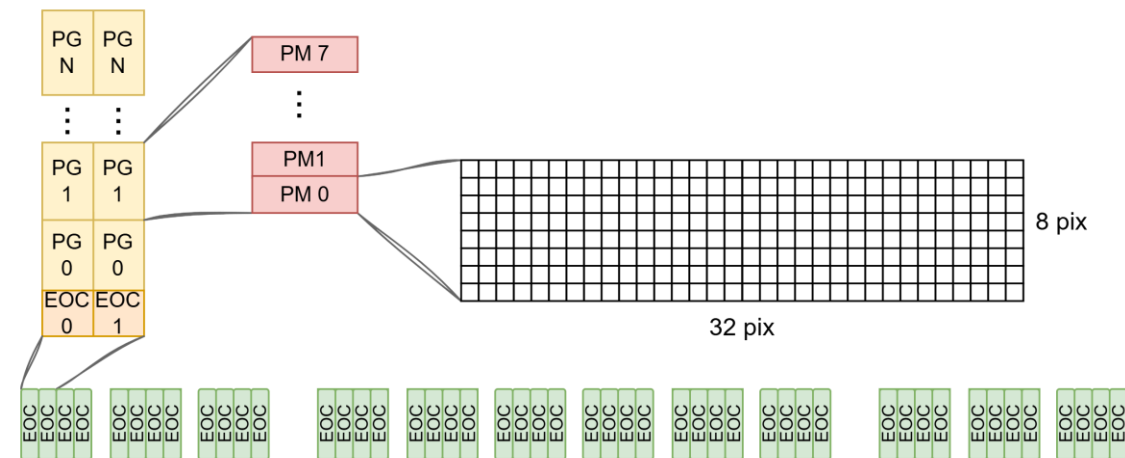
ALMIRA project is developing the next generation of vertex and tracking detectors for ALICE:

- **Monolithic stitched** sensor for wafer-sized detector chips.
- Pixel hit rate **100 MHz/cm²** and **70mW/cm²** power target.
- Pixel pitch of 10 μm with 100ns time-of-arrival resolution.
- **TPSCo 65nm** CMOS technology.
- First readout stage is **asynchronous non-zero-suppressed** in an 8x8 pixel area.
- Synchronous data and control busses for global data transmission.

Thanks to the
ALMIRA project
for these slides

PixESL model:

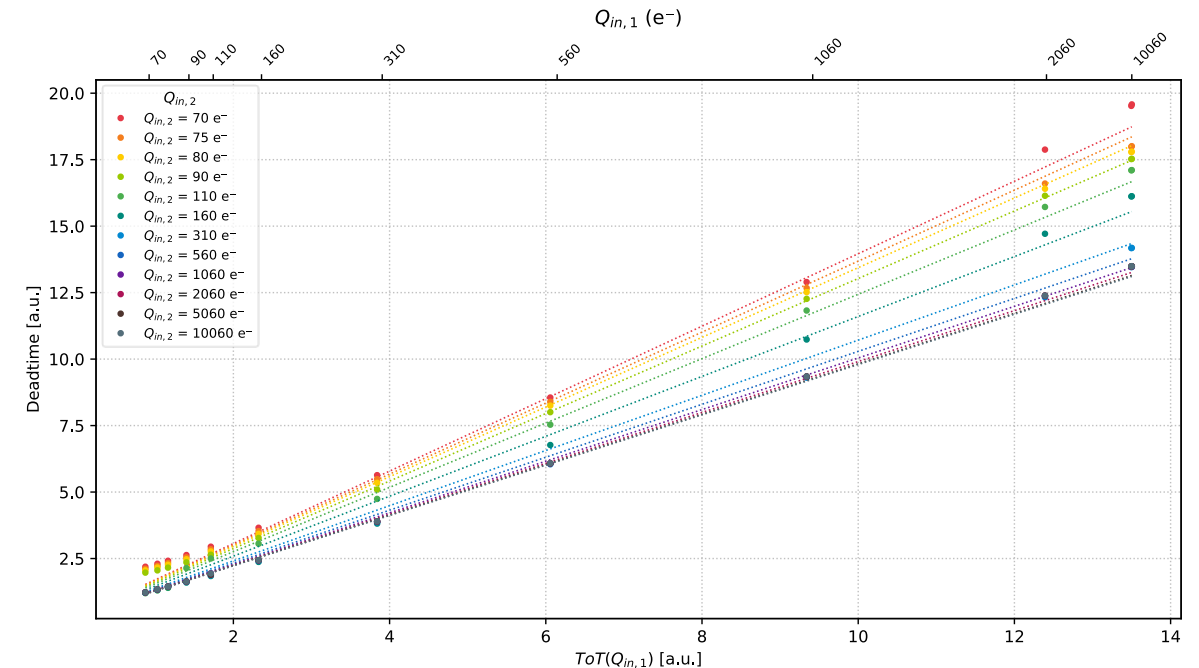
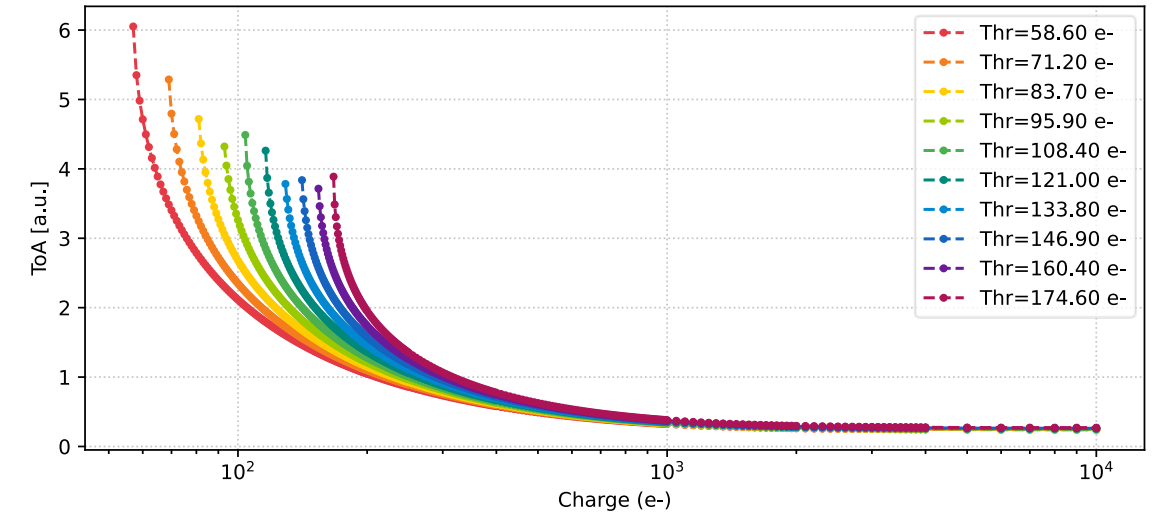
- Analog FE model derived from analog simulation:
 - LUTs map the input charge(s) to ToT and ToA.
- **Asynch. model** of the Pixel Macro (PM)
- **Pixel Group** and **EoC node** are synch. readout stages
- The simulation is based on **physical events**



ALICE3 Tracker Upgrade Analog Front-end Studies

The analog FE is accurately modeled through automated scripts that map the AFE circuitry behaviour in the analog Spectre simulation to a LUT-based PixESL Analog FE. It models:

- **ToT** and **ToA** as a function of **charge** and set **threshold**
- **Deadtime** between two hits as function of their charges
- Random **Threshold Mismatch**
- **Charge Noise**



ALICE3 Tracker Upgrade - Grouping Efficiency in the Pixel Macro

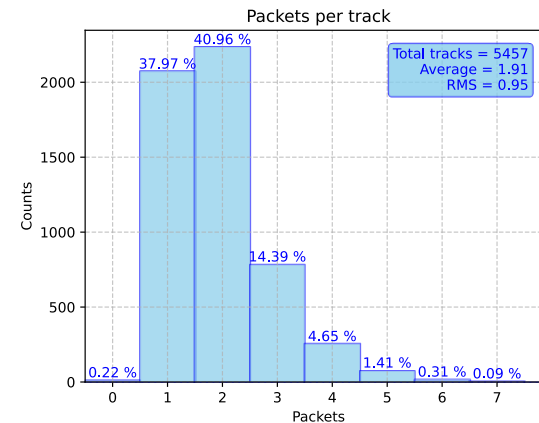
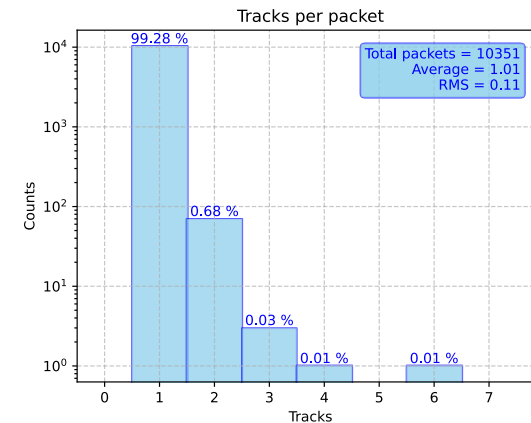
Each pixel macro collects hits/tracks for a configurable integration time.

The **longer** the integration, the more tracks show up in a single packet, thus **reducing bandwidth**.

For an excessively long integration time, multiple tracks per packet can occur in the same location, thus increasing the chance of losing hits due to **pileup**.

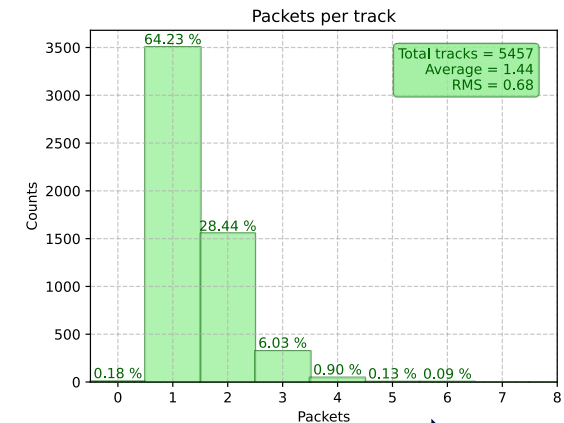
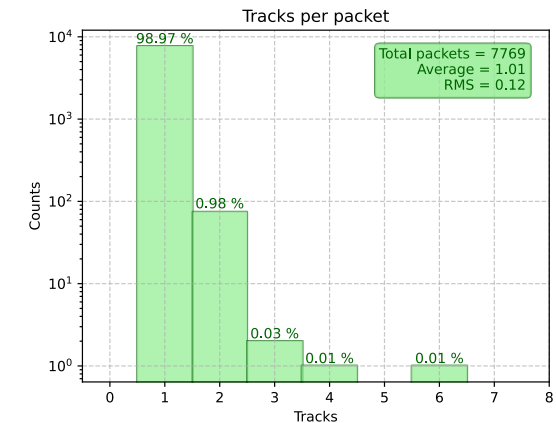
Integration time: 175 ns

10351 packets



Integration time: 315 ns

7769 packets



More pileup

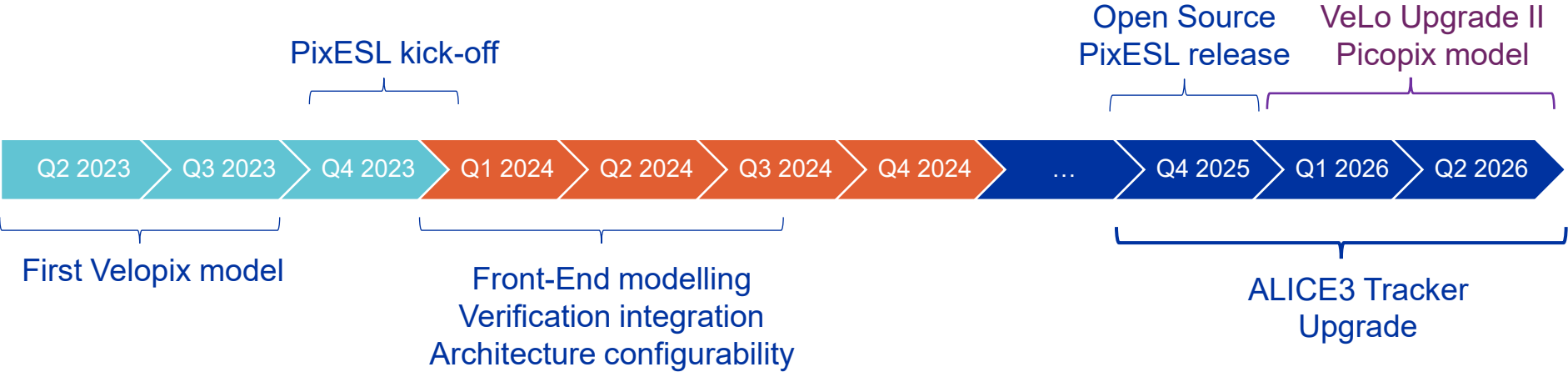
More packets

Conclusions

- **PixESL** is a valid tool for **high-level modelling** and **architectural analysis**
- The project is part of the **ECFA DRD7 collaboration**



funded by 



**Contacts**

PixESL Maintainers:
PixESL-maintainers@cern.ch

**PixESL Project**

Repo: <https://gitlab.cern.ch/iod/pixesl-group/core>
Docs: <https://cern.ch/pixesl>

Accessible upon request

Thank you!



home.cern