# Virtual prototyping of pixel detectors with PixESL framework in High Energy Physics

WP5 IC Tech
EP R&D

***Jashandeep Dhaliwal***[1], *Francesco Enrico Brambilla*[1,2], *Davide Ceresa*[1], *Stefano Esposito*[1], *Kostas Kloukinas*[1]

*[1]CERN, 1211 Geneva 23, Switzerland*
*[2]KU Leuven, 3000 Leuven, Belgium*

*Contact info:*
***jashandeep.dhaliwal@cern.ch***

## Abstract

*PixESL pioneers a virtual prototyping framework for future particle detectors in high-energy physics. Developed at CERN under the EP R&D Work-Package 5, this framework enables high-level abstraction, simulating the full detector chain from particle interaction to data packet readout. It facilitates early optimization of chip and system architecture, which is critical for meeting experiment specifications. PixESL models crucial components such as analog front-end, digital circuitry, and data readout networks, empowering designers to analyze interactions and optimize performance. Leveraging SystemC, PixESL offers rapid simulation runtime and above-RTL abstraction, presenting a pivotal tool for advancing particle detector design and verification.*

## Introduction

The **high cost of prototyping** at advanced technology nodes, as well as the **complexity of future detectors**, necessitate the use of a system design technique widely used in industry: **design space exploration** through **high-level architecture studies** to establish precise and optimal requirements. This work presents **PixESL**: a **programmable SystemC framework** for simulating the **readout chain** from the front-end chips to the detector back-end.

## Methodology

- The design language is **SystemC**: a C++ library for system-level design
- The structure enables **quick architectural exploration** thanks to **modularity** and **code-reusability**
- The model is **event-based** with an **approximately-timed** coding style: system components operating synchronously on a shared time base triggered by events
- Communication is based on an **always-push configuration**
- Packet transfers are based on Transaction Level Modelling (**TLM2.0**) sockets

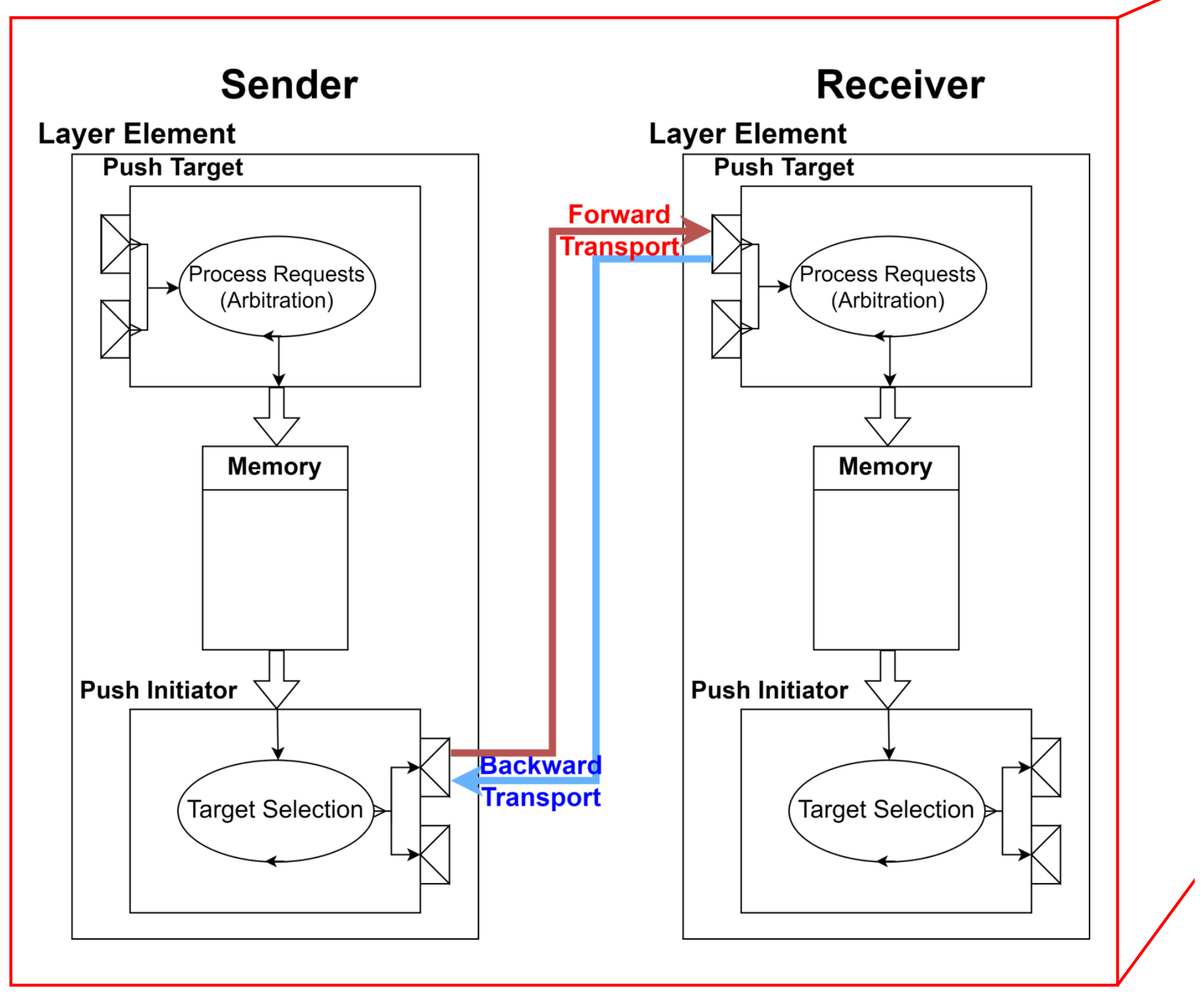
## Framework description

The **stimuli** as input data can come from:

- **External file**: external data deriving from physics-level detector simulations
- **Event generator**: internally generated events for parametrized studies

The **Pixel Front-End (FE) model** is described in 3 classes:

- **C++ analog FE**: defines the behaviour of the analog circuits: analytically evaluates the rising and falling edges of the output discriminator
- **SystemC Wrapper**: converts the instant C++ events into timed SystemC signals
- **SystemC Pixel**: instantiates the SystemC Wrapper and uses the SystemC signals to perform filtering, clustering, and to generate digital data packets for the readout network



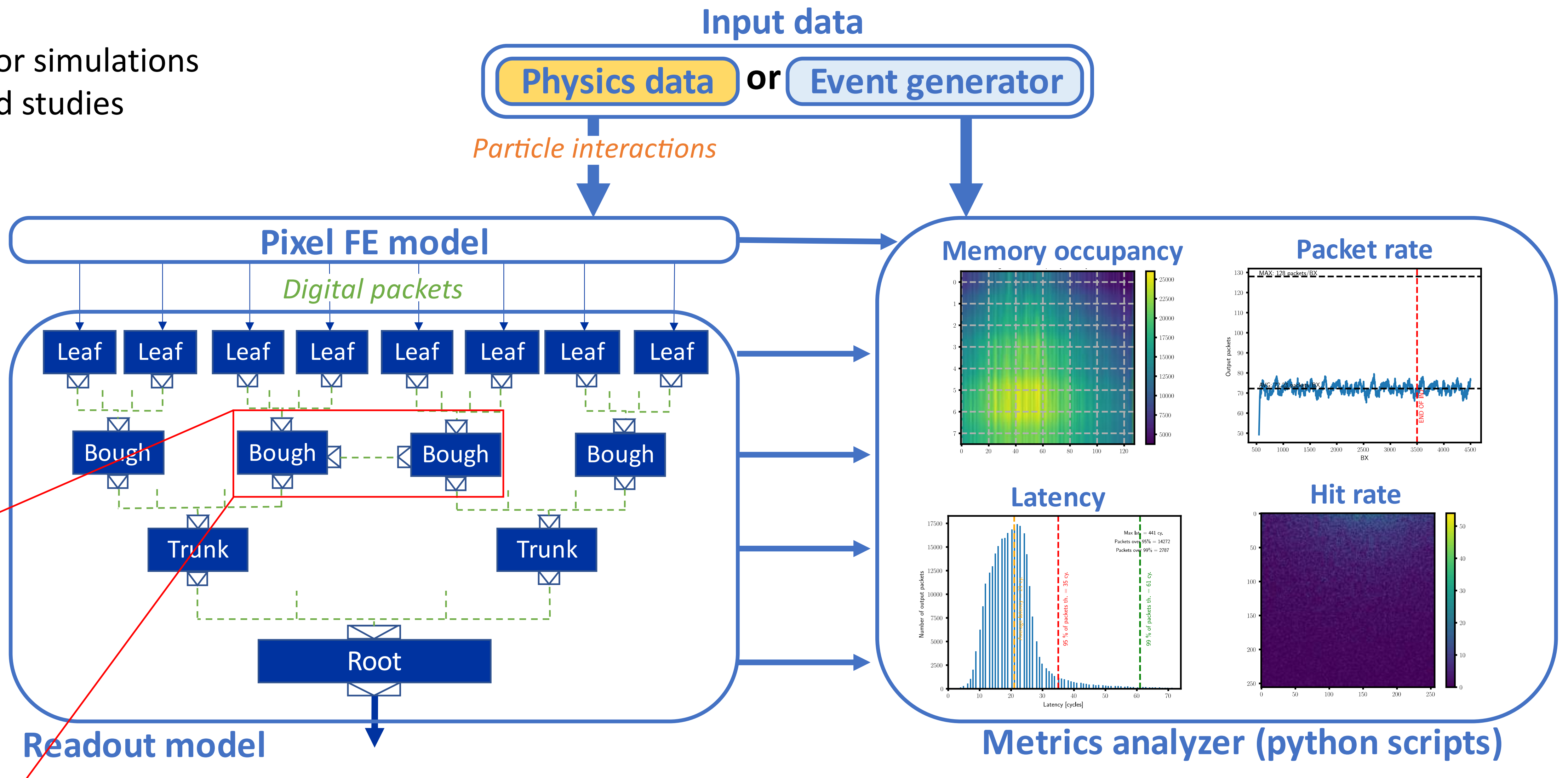


*Figure 1: Scheme of the framework and detail of TLM communication.*

The **SystemC model**, which **simulates the readout dataflow**, is based on two main components:

- **Layer**: contains the processing and readout modules: it stores data, communicates with other modules, performs arbitration and data routing
- **Network**: instantiates the connections between same-layer modules (intra-layer) and different-layer modules (inter-layer)

The **Pixel FE and readout model** provide a **SystemC reference model** to be integrated within the **SystemVerilog UVM environment**.

The **metrics analyzer** collects information across the model to compute the **readout efficiency**, **latency**, and **average queue occupancy**.

## Results

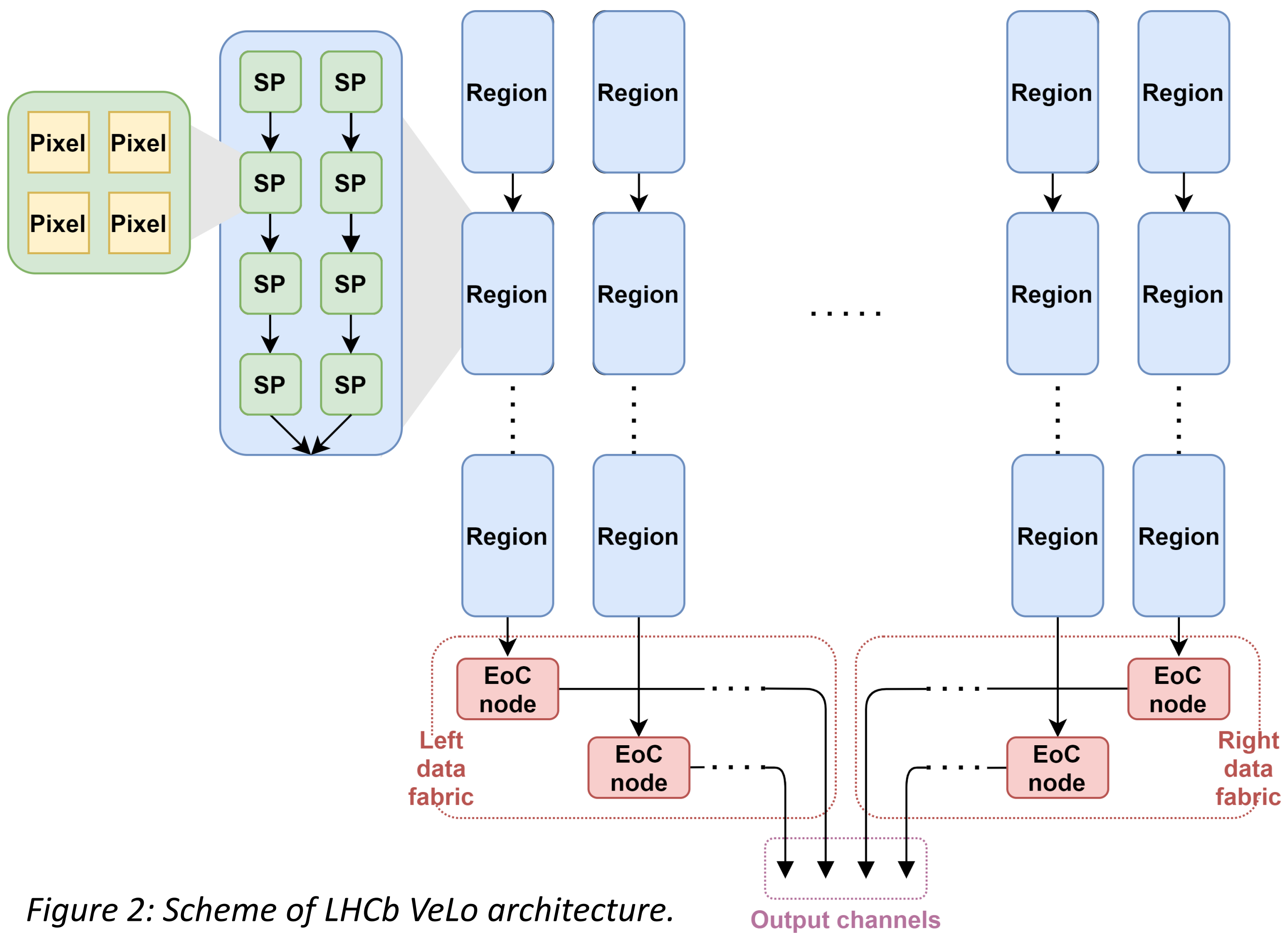


*Figure 2: Scheme of LHCb VeLo architecture.*

The first case study chosen as a proof-of-concept of the framework is the **LHCb VeLo upgrade II** (figure 2), a **data-driven readout architecture** based on four layers: *pixels*, *super-pixels (SP)*, *regions*, and *end-of-column (EoC) nodes*.
Table 1 shows the architectures and the results of a **parametric study** that started from **VeloPix** to reach a **new proposal** to **increase the performance** to counteract the extremely high input rate of future upgrades.
The **VeloPix** architecture shows **several lost packets** because of the **drastic congestion of data packets** at the region level.

The **proposed architecture** addresses the issue by doubling *region* columns, *EoC nodes*, and *output channels* to increase maximum throughput to **128 packets/cycle** while halving the number of regions per column to **mitigate hardware overhead**.
Furthermore, the **pixel FE model** shows a **99% matching** with a **50x faster simulation time** compared to RTL.

| | | VeloPix | Proposed |
|---|---|---|---|
| Layers size | Pixel | 256x256 | |
| | Super-Pixel | 128×128 | |
| | Region | 64×16 | 128×8 |
| | EoC node | 64 | 128 |
| | Output ch. | 8 | 16 |
| FIFOs depth | SP FIFO int. | 4 | |
| | SP FIFO ext. | 2 | |
| | Region FIFO | 4 | |
| | EoC node FIFO | 2 | |
| Clock | Matrix clock | ×1 | |
| | EoC node clock | ×8 | |
| Results | Readout eff. | 86% | 100% |
| | Avg. latency | 114 cy. | 17 cy. |

*The clock is referred to the Bunch-Crossing rate

*Table 1: Configurations and results of VeloPix and proposed architectures.*

## Conclusions

The **PixESL** framework proposes a **rapid** and **efficient prototyping approach** for system-level development, **50 times faster** compared to RTL simulations, using well-established open-source languages like **SystemC** and **Python**.

For instance, a readout network with approximately **20 thousand nodes** requires around **5 seconds** of **build time** and processes up to **70 thousand transaction/s**.

In addition, the framework ships with a **UVC interface** which **allows co-simulation** of the model within a **UVM environment** as a **golden reference**, capable of distinguishing design implementation bugs from architecture inefficiencies.

The **release of the framework** is currently foreseen at the **end of 2024**.

Presented at the **CERN EP R&D Day**
by **Jashandeep Dhaliwal**
**May 22, 2024**