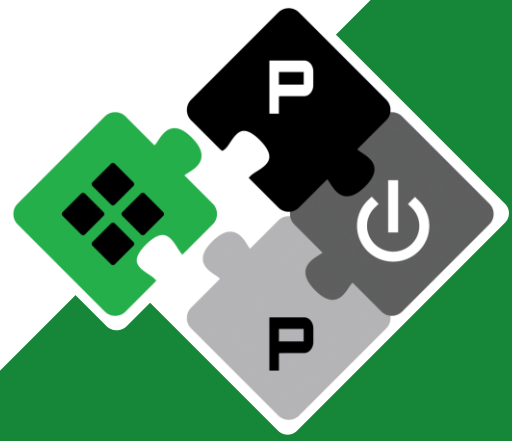
ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

CachePool: Many-core cluster of customizable, lightweight scalar-vector PEs for irregular L2 data-plane workloads

Integrated Systems Laboratory (ETH Zürich)

Zexin Fu, Diyou Shen zexifu, dishen@iis.ee.ethz.ch

Alessandro Vanelli-Coralli avanelli@iis.ee.ethz.ch
Luca Benini lbenini@iis.ee.ethz.ch



@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

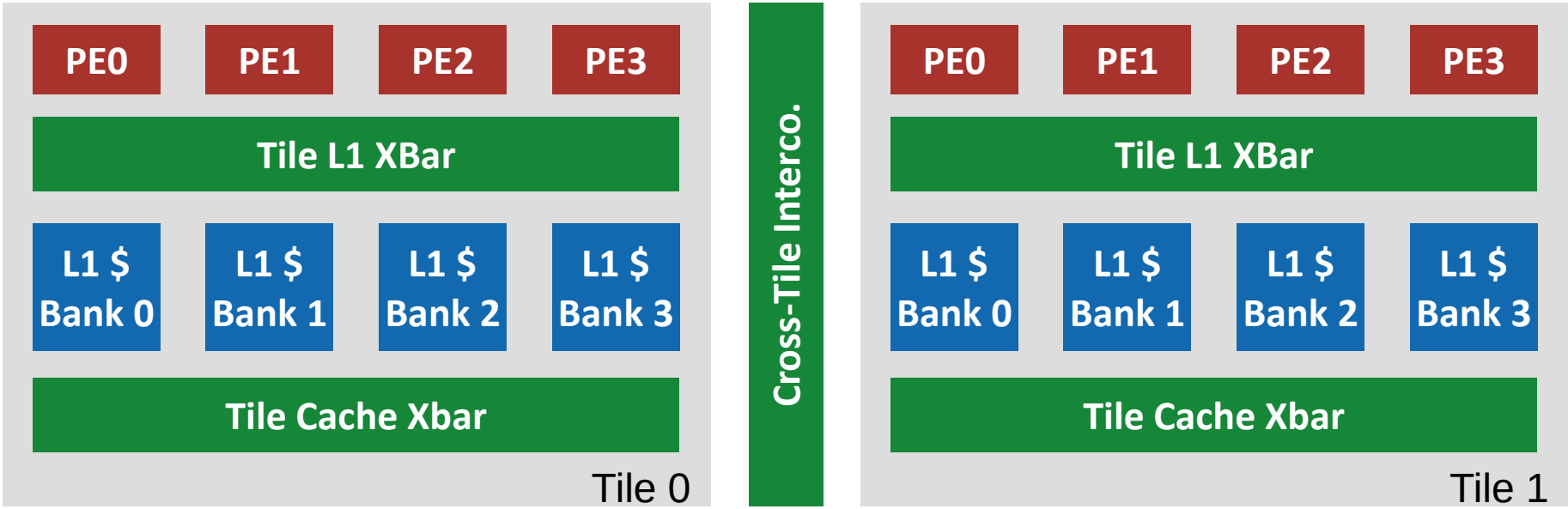
PULP Platform
Open Source Hardware, the way it should be!

Status Update: Hardware

- **Finish the implementation of a single Tile**
  - Four 64b Snitch-Spatz4 PEs
  - Four 256b-wide cache banks
  - Programable selection in XBar
    - Can apply different mux offset bits from a CSR
    - Prepare for the partition functionality

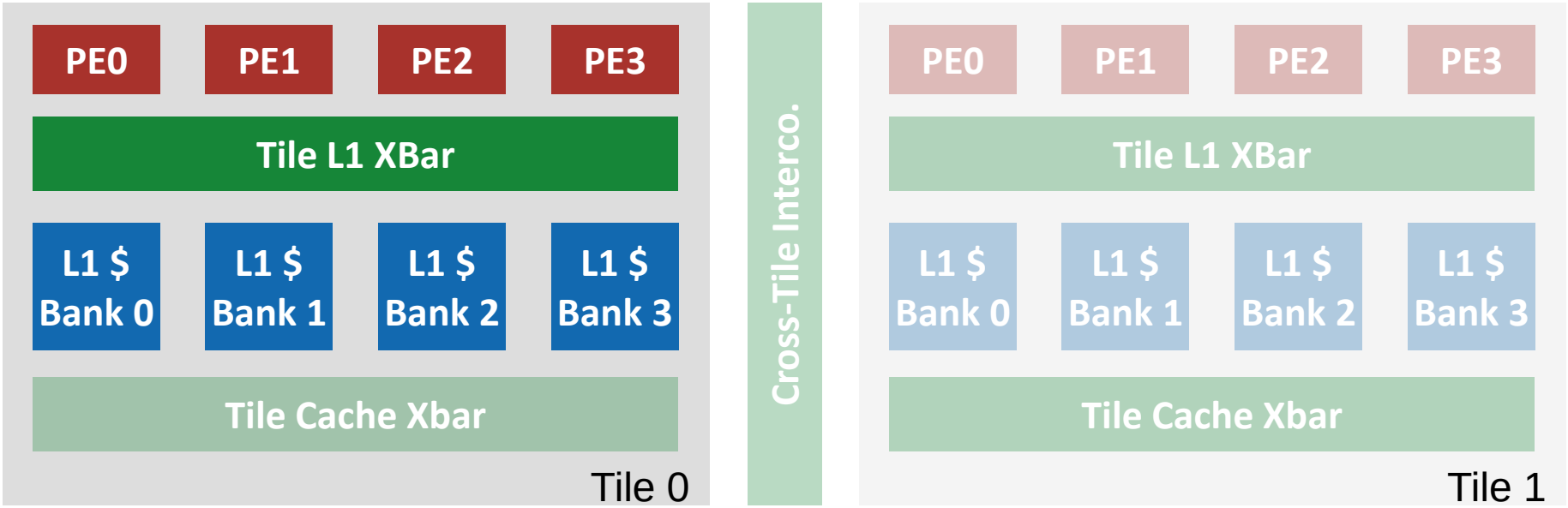
Status Update: Hardware

- **Finish the implementation of a single Tile**
  - Four 64b Snitch-Spatz4 PEs
  - Four 256b-wide cache banks
  - Programable selection in XBar



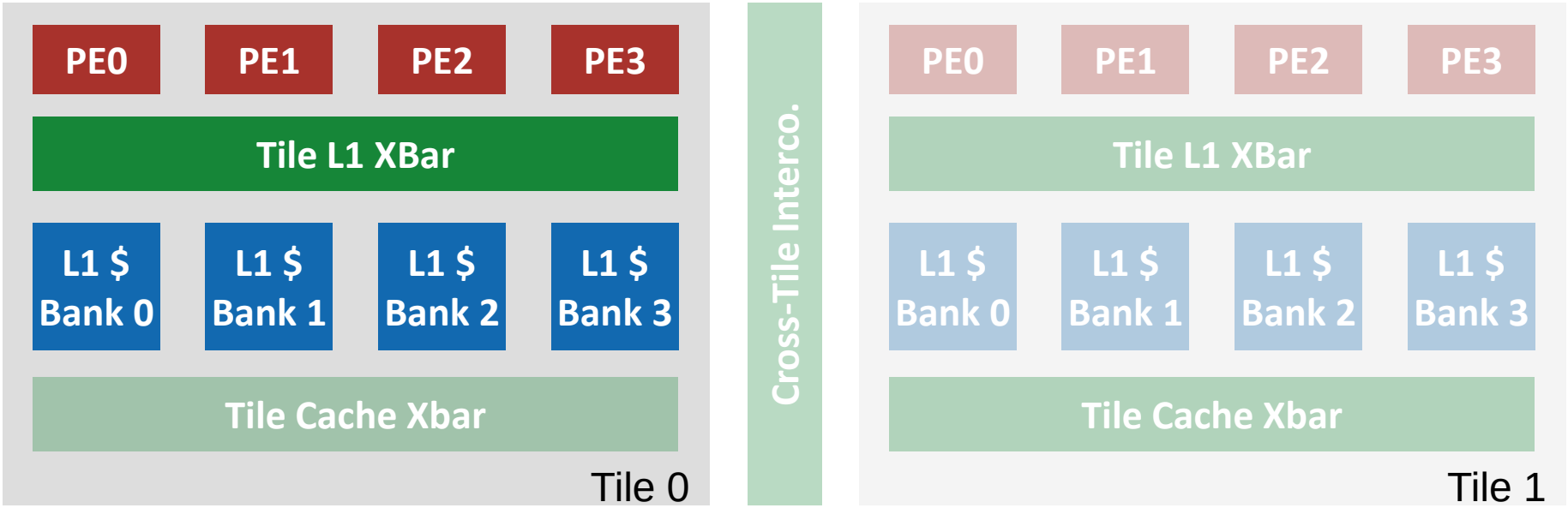
Status Update: Hardware

- **Finish the implementation of a single Tile**
  - Four 64b Snitch-Spatz4 PEs
  - Four 256b-wide cache banks
  - Programable selection in XBar



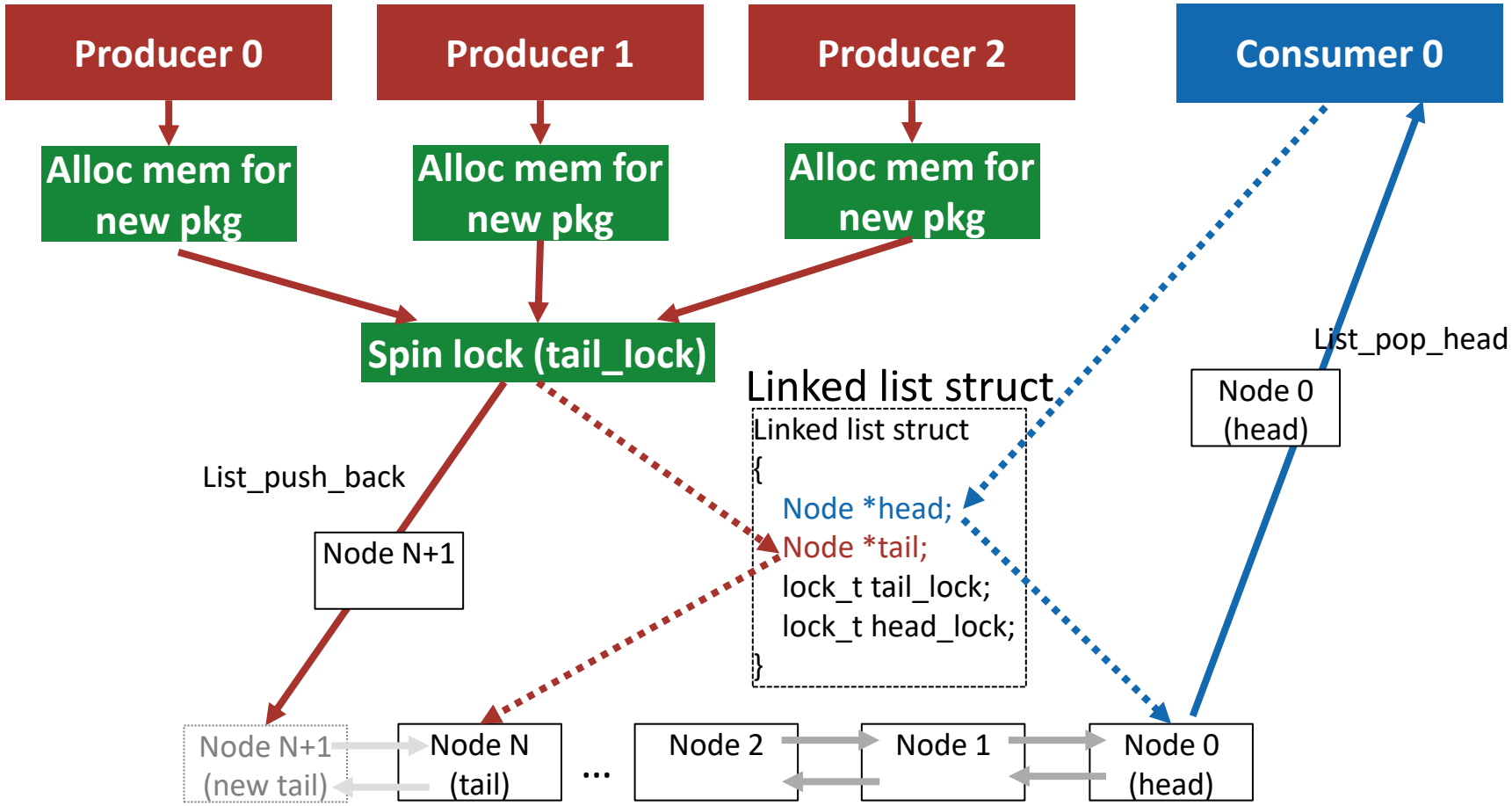
Status Update: Hardware

- **Next Steps**
  - Clean up the code and push to GitHub
  - Currently use a single channel fake main memory in testbench => dramsys
  - **Atomic support (high priority)**



Status Update: Software Plan

1. Multiple producers, single consumer double linked-list kernel

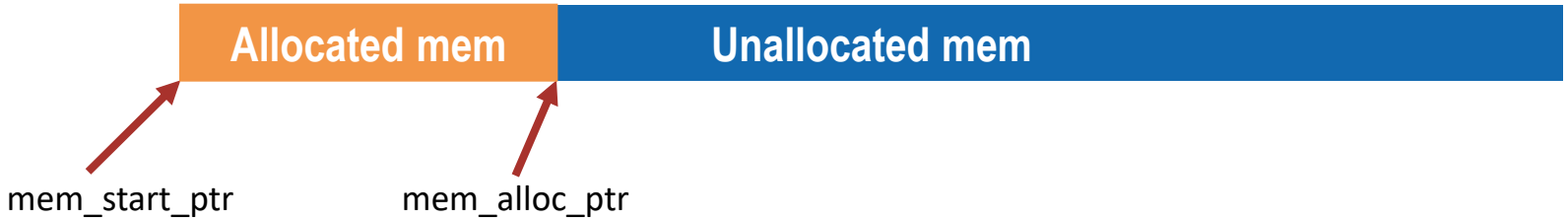


Status Update: Software Plan

2. Memory management runtime

- Assuming a fixed memory block size (a page) for each allocation and deallocation

1) Initial allocation: allocate new page from the sequential memory space



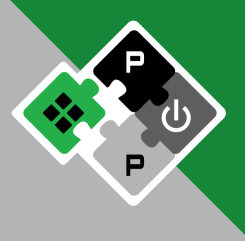
2) Free stage: push ptr of the deallocated page into a free mem linked list

3) Common allocation: pop the head node from the mem linked list, get the ptr to the new page

Status Update: Software

- **Progress of work**

  1. Multiple producers, single consumer double linked-list kernel
     - Multiple producers, single consumer read/write shared linked list
     - TODO:
       - The simulation of the data movement of the consumer
       - Try to use multiple cores for data movement
       - Full atomic operation support from HW
  2. Memory management runtime
     - Dynamic memory allocation, deallocation, and reclamation at fine granularity (per package)
     - TODO:
       - Variable allocation granularity



Thank you!