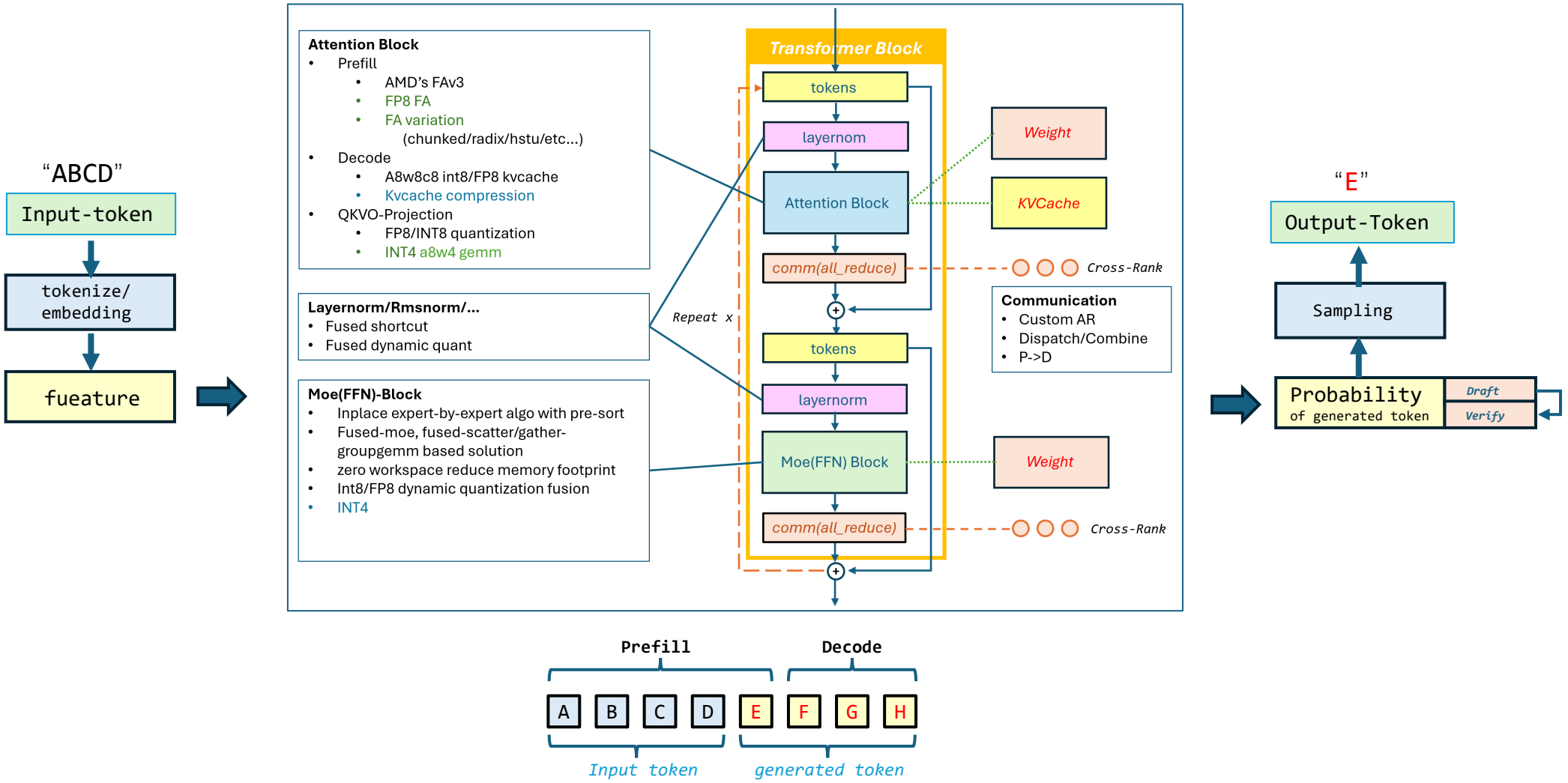
AITER / MoRI

Introduction

carlus.huang@amd.com

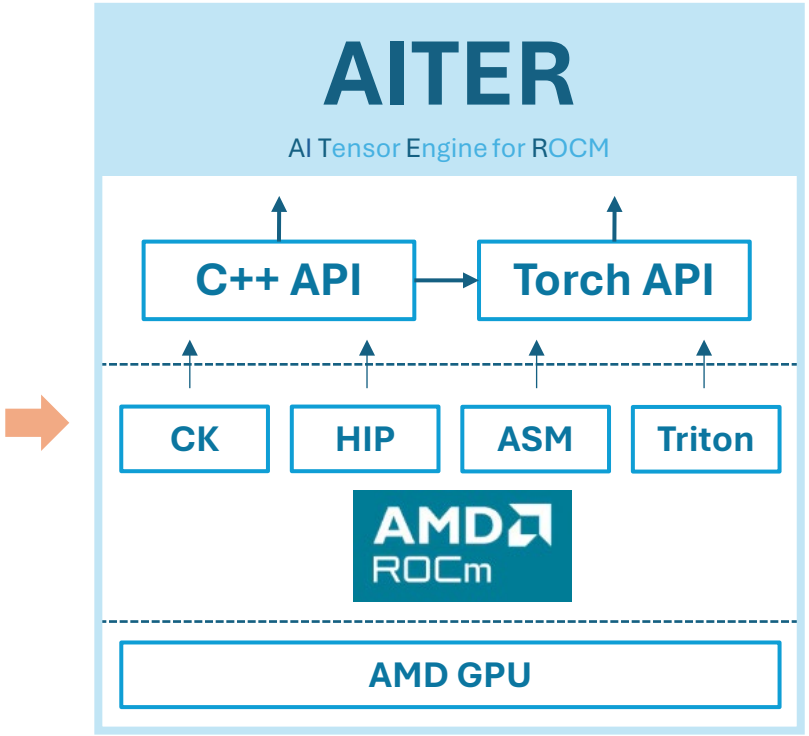


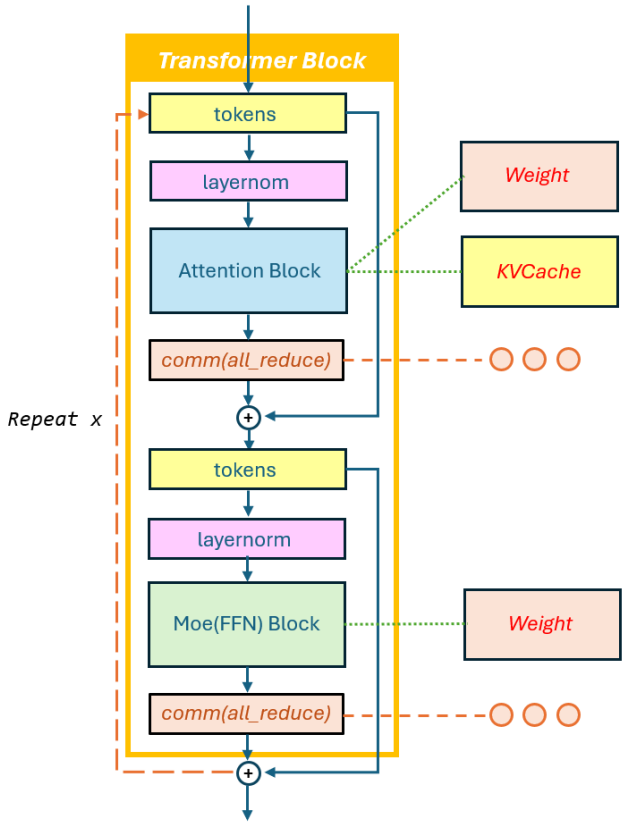
=> View LLM as Building Block

AITER overview

GITHUB : ROCm/aiter

Prefill Attention	F/B	FAv3 FWD FP16/BF16
		FAv3 BWD FP16/BF16
		MLA FP16/BF16
		FA FP8 FWD+BWD (Block-Scale)
		Chunked-prefill
Decode Attention	F	Paged Attention FP16/BF16
		Paged Attention FP8 per-tensor quant
		Paged Attention FP8/INT8 with KV per-token quant
		KVCache update & Rotary
		Batched Decoding
		MLA Decoding
Fused-Moe	F	Moe-Sorting kernel and tiling solution
		FP16/BF16 per-token Fused-Moe
		FP8/INT8 per-token Fused-moe
		FP8 per-tensor Fused-Moe
		FP8/INT4 per-tensor Fused-Moe
		Fused-FFN
Low Precision Gemm	F	FP8 per-token/channel Gemm
		FP8 Block Scale Gemm
		INT4 weight only gemm
Distributed Gemm	F/B	(need evaluation)
Normalization and fusion	F	layernorm+quant/shortcut
		rmsnorm+quant/shortcut
Custom Comm.	F/B	AR/AG... fused with noamalizatoin
		AR/AG... quantized
		Opitmized hipgraph support
Conv2d/2d	F/B	fp16/bf16 fwd/bwd/wrw
		fusion with bias/activation, etc..
And more fused ops...		





AITER scope:

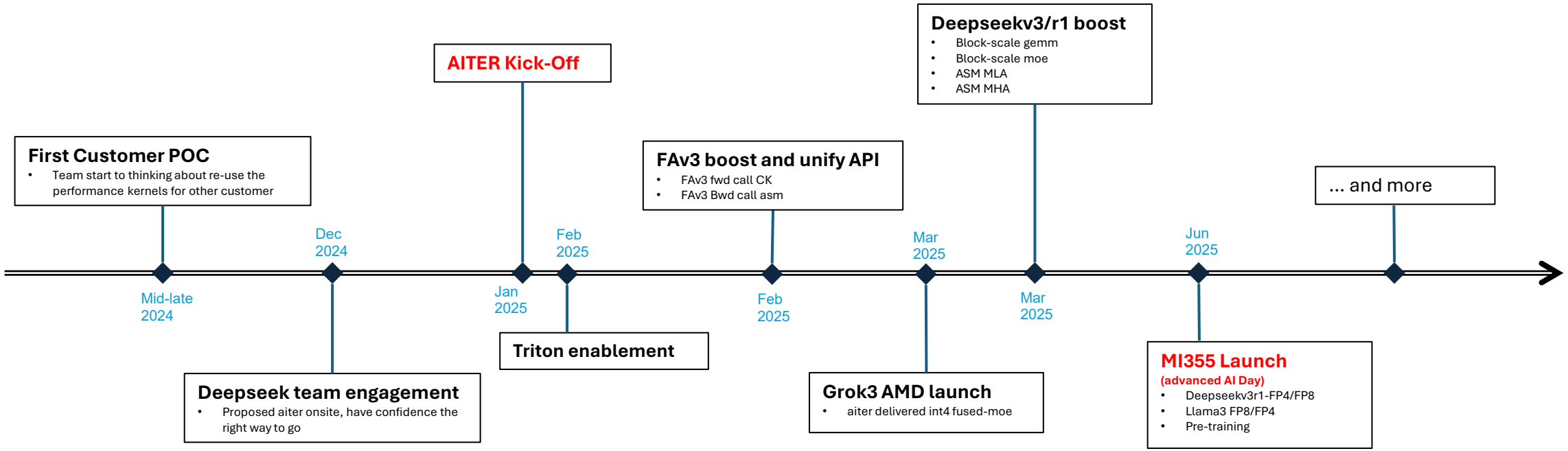
- A lightweight and customizable/tunable/deployable AI tensor engine.
- A collection of inference AI operators
- A collection of training AI operators
- Fused dynamic quantization kernels and offline utilities.
- LLM block-level solutions to plug into customer's framework. (fused-moe, quant-KVCache, etc...)
- C++/Pytorch API and package
- heuristic/user-driver-tuning for self-hosted kernels written in CK/ASM/HIP/Triton

AITER is not:

- a LLM service engine or LLM workload balance scheduler
- a distributed inference or training framework
- a graph compiler
- a thin wrapper on top of existing rocm libraries.

AITER timeline

- Hosting reusable optimized kernels
- Establishing standardized integration frameworks
- Enabling rapid deployment across clients



SGL

vLLM

Alibaba

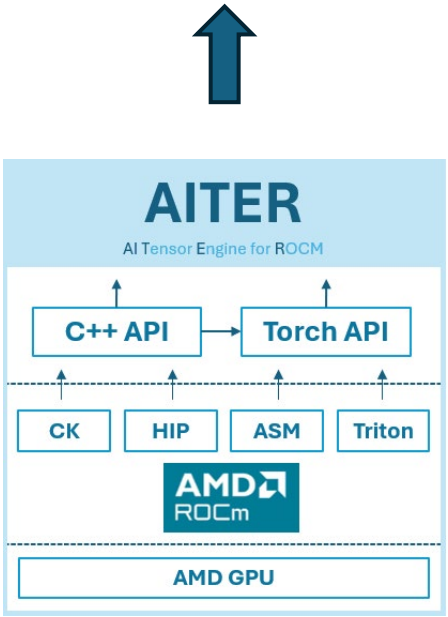
Tencent 腾讯

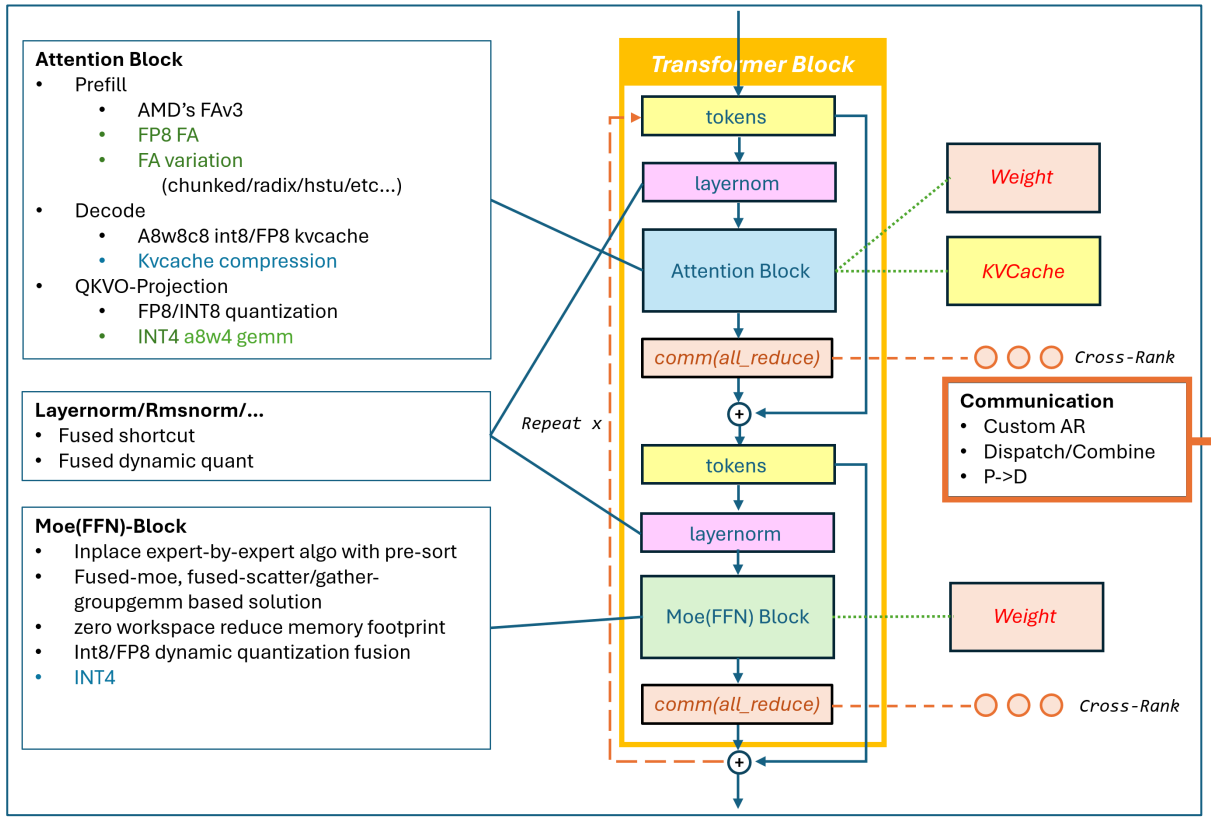
ByteDance

Grok

Meta

deepseek





MoRI

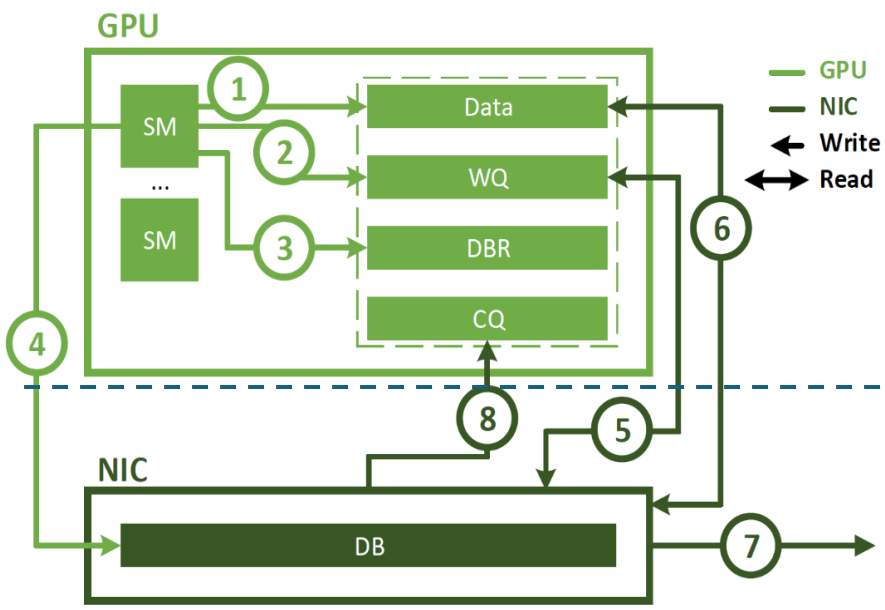
(Modular RDMA Interface)

LLM Native Communication Engine

GITHUB : ROCm/mori

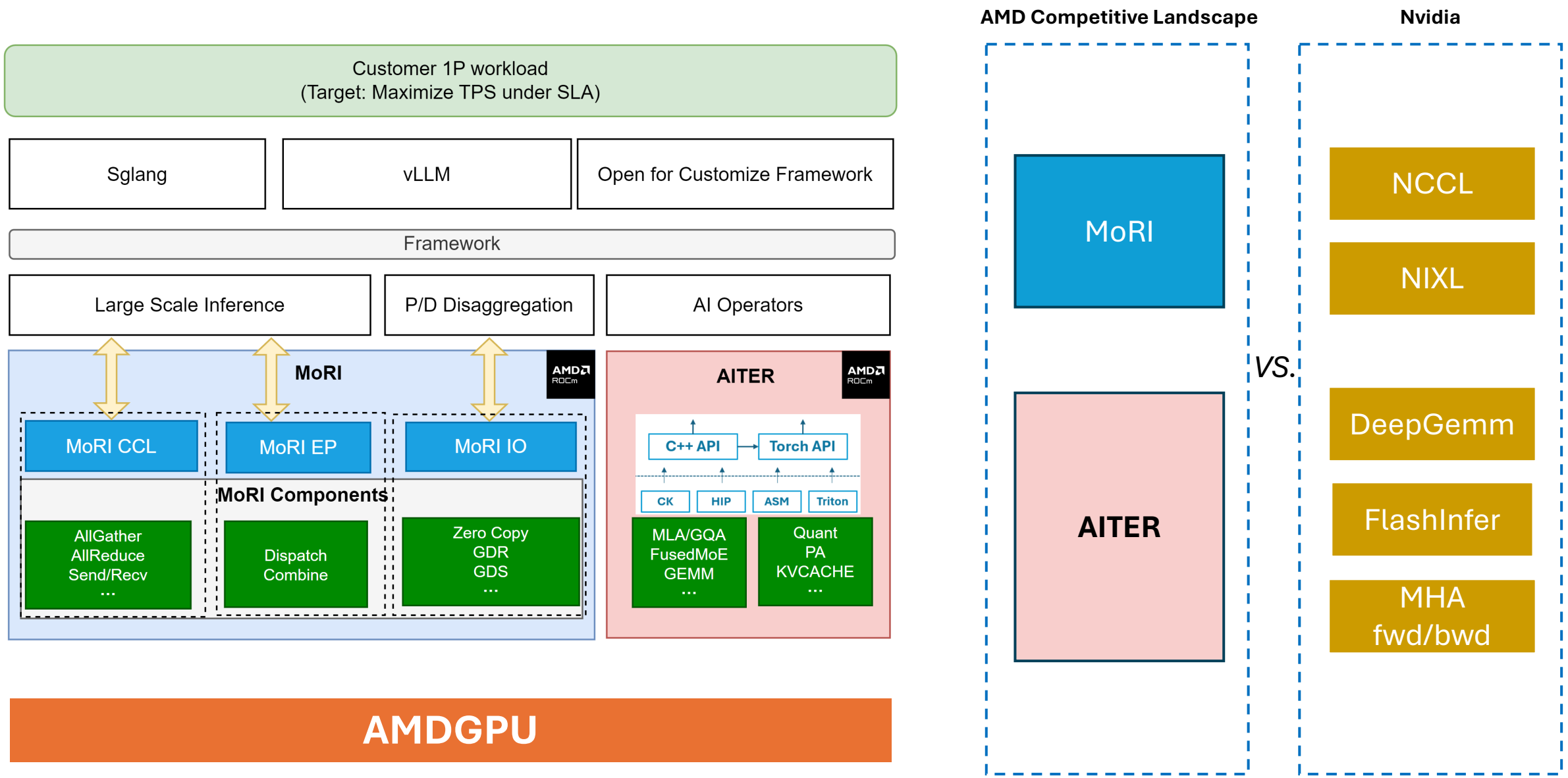
MoRI is all you need to implement high-performance communication on GPGPUs.

MoRI = DeepEP + shmem + CCL + ...



MoRI

application	Custom Kernels (e.g DeepEp / pplx-kernel)	SHMEM API
	CCL	Fused Kernels
framework	Transport Context Management	Buffer Management
	Topology Detection	...
core	thread-level / wave-level / block-level	
	data transfer	synchronization / signal & atomic / reduction & quantization
transport	IBGDA	GPU P2P / NVMe
Hardware	Broadcom	Mellanox / Pensando


Customer 1P workload
(Target: Maximize TPS under SLA)
Sglang
vLLM
Open for Customize Framework
Framework
Large Scale Inference
P/D Disaggregation
AI Operators
MoRI
MoRI CCL
MoRI EP
MoRI IO
MoRI Components
AllGather
AllReduce
Send/Recv
...
Dispatch
Combine
Zero Copy
GDR
GDS
...
AITER
C++ API
Torch API
CK
HIP
ASM
Triton
MLA/GQA
FusedMoE
GEMM
...
Quant
PA
KVCACHE
...
AMDGPU
AMD Competitive Landscape
MoRI
AITER
VS.
Nvidia
NCCL
NIXL
DeepGemm
FlashInfer
MHA
fwd/bwd

MORI-IO

PoC Done

- Customer-X on-board for P/D disaggregation
- Batch transfer(PoC ready)
- Connection pool
- Topo detection

Production Ready

- Fault tolerant
- Perf optimization

Customer-X Deployment

MORI-EP

PoC Done

- 43GB/s BW on 4-node cluster w/ CX7
- Customer-Y Training communication optimization WIP
- Customer-X Large-EP on-board

- Thor2 IBGDA functional ready
  - IBGDA basic op(write/amo)
  - Shmem function(put/Imm)
  - Dispatch/combine kernel based on Thor2 IBGDA

Production Ready

- Thor2 perf optimization ready

- Training support

Customer-X/Y Deployment

MORI-CCL

Development Kick-off

- Ultra-low latency CCL kernels
- Zero-CU CCL kernels
- Computation & Communication overlap
- Low precision CCL (experimental)

SGLANG+MORI Integration Evaluating the Tech-Path now!

AMD's LLM Optimization

Attention Block
- Prefill
  - AMD's FAv3
  - FP8 FA
  - FA variation (chunked/radix/hstu/etc...)
- Decode
  - A8w8c8 int8/FP8 kvcache
  - Kvcache compression
- QKVO-Projection
  - FP8/INT8 quantization
  - INT4 a8w4 gemm

Layernorm/Rmsnorm/...
- Fused shortcut
- Fused dynamic quant

Moe(FFN)-Block
- Inplace expert-by-expert algo with pre-sort
- Fused-moe, fused-scatter/gather-groupgemm based solution
- zero workspace reduce memory footprint
- Int8/FP8 dynamic quantization fusion
- INT4