ZKPU

Make ZK ASIC Decentralized, Scalable and Open-Source

AGENDA

- Background
- Market
- Problem
- Solution
- Founders
- Proof of Concept

BACKGROUND- ZKP DEVELOPMENT

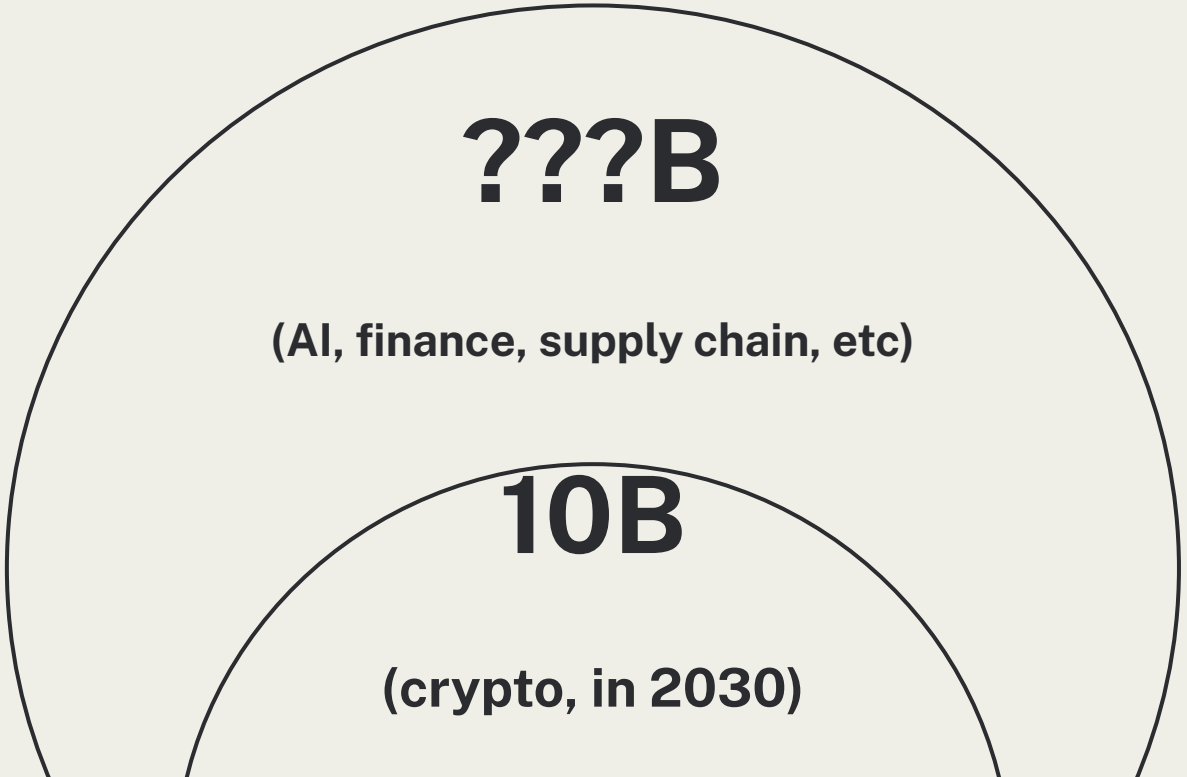
Timeline:

- Original concept of ZKP in 1980s,
- zk-SNARK(zero-knowledge **Succinct Non-Interactive** Argument of Knowledge): 2011
- SNARK became practical due to new algorithm/protocol invented: 2016 (Groth)
- Crypto boom, new software protocols are invented: PLONK, STARK, PLONKY2, HALO2, NOVA, etc.

Focus:

- Proof generation and verification.
- Fast generation, short proof, post quantum, integration, decentralization.
- **Mostly driven by crypto teams now.**

MARKET



Crypto

Ethereum L2 in 2030

- ~90 billion zk proofs
- 83,000 transactions per second

Next Big Thing: BTC ZK Rollup

AI

ZKML is being actively researched now.

Others

Finance, Supply Chain, etc.

PROBLEM – ZKP Workloads Overwhelm CPUs

Demanding Workloads:

- **MSMs**: Require massive parallelism, slow on CPUs.
- **NTTs**: Data shuffling and bandwidth limits cause slowdowns.

CPU Challenges:

- General-purpose CPUs are too slow and lack the resources required for efficient ZKP proof generation.

Solution:

- Dedicated **Co-Processors** are essential to handle these workloads efficiently while **maintaining decentralization**.

PROBLEM - Early-Stage Solution

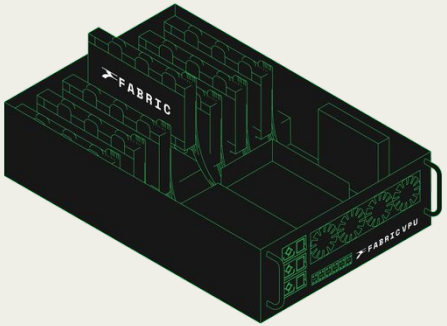
GPU

FPGA

- **Costly**: High acquisition cost and TCO(Total Cost of Ownership) -> hard to scale efficiently.
- **Latency(GPU)**: Not optimized for Crypto Algorithm -> Slow proof generation and resource-inefficiency
- **Complexity(FPGA)**: Requires hardware programming -> complicated, centralized and unscalable

FPGA <<< ASIC!!!

PROBLEM – In-development ASIC



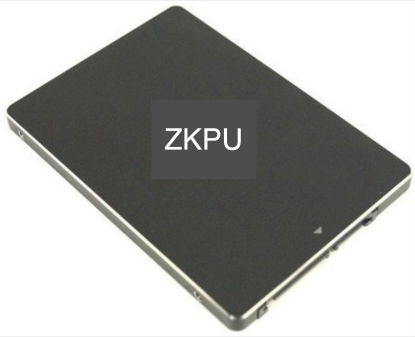
Fabric

Cysic

Features: Proprietary Server Box

- **Costly**: High acquisition cost and TCO (Total Cost of Ownership) → Limits efficient scaling.
- **Centralized**: Complex deployment and expenses → Reliance on centralized operators & Censorship.
- **Risky**: Rigid design + evolving ZK protocols → Inflexible ASICs face high risks.

SOLUTION – ZKPU in Edge

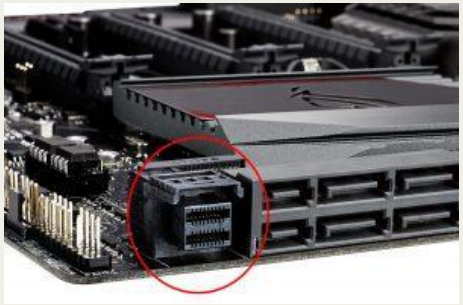


ZKPU in **U.2** Form Factor

OR

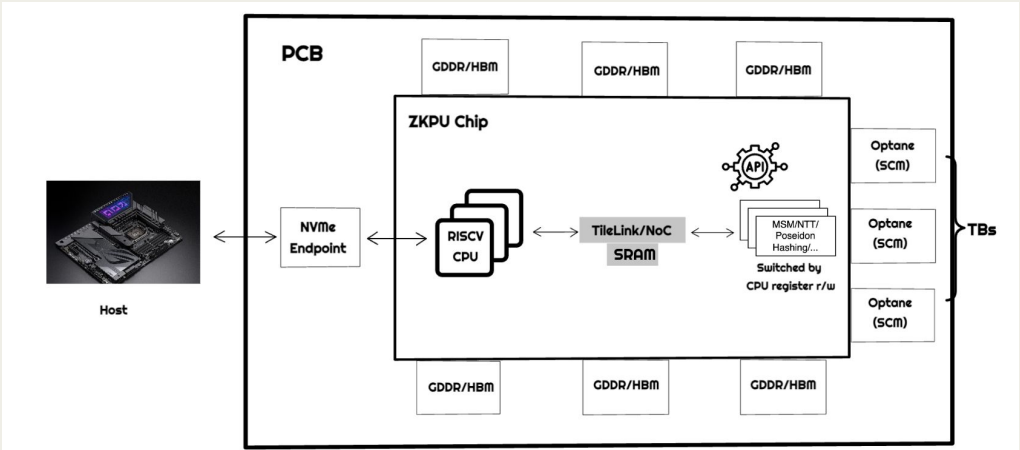
ZKPU with **U.2 to USB** adapter

Plug-and-Play

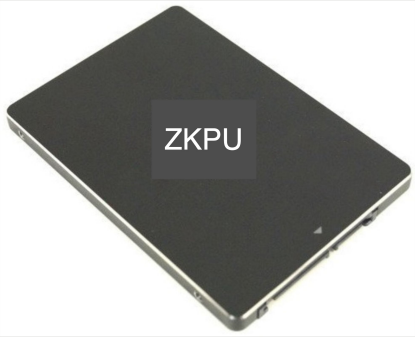


Any Motherboard with Any OS(even Raspberry Pi)

- **Feature: RISC-V SoC**(System on Chip) with support of **NVMe** (Non-Volatile Memory express)



SOLUTION – ZKPU in Edge

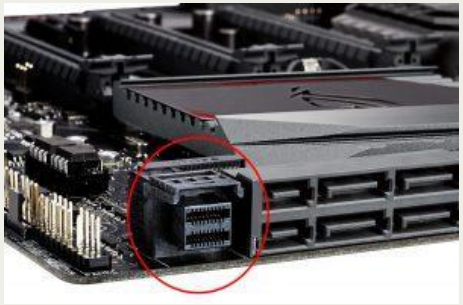


ZKPU in **U.2** Form Factor

OR

ZKPU with **U.2 to USB** adapter

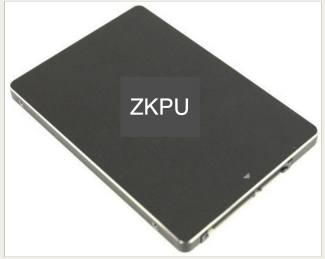
Plug-and-Play



Any Motherboard with Any OS(even Raspberry Pi)

- Feature: RISC-V SoC(System on Chip) with support of NVMe (Non-Volatile Memory express)
- Affordable: $200~300 BoM(1/10 of GPU); easily scalable
- Low TCO(Total Cost of Ownership): Low deployment cost and low power consumption(25 watt, 1/20 of GPU)
- Decentralized: Compatible with any OS supporting NVMe
- Configurability: on-chip RISC-V CPUs & eFPGA -> flexible Modular Arithmetic modules
- Low-latency: Optimized algorithms and data communication for high performance
- **On-Chip Flash Storage:** ZKVM On-Chip, No PCIe Loading/Unloading, Off-chain Data Interaction

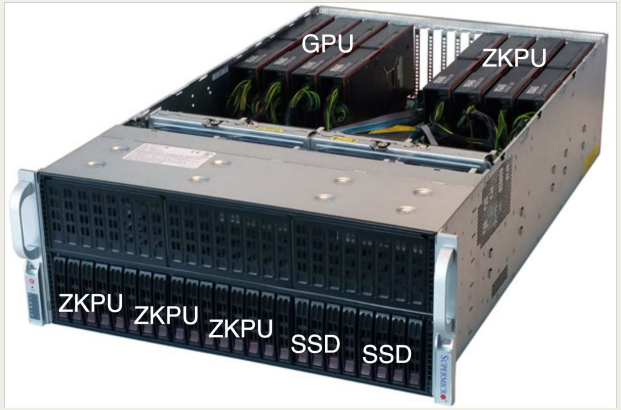
SOLUTION – ZKPU in Datacenter



ZKPU in **U.2** Form Factor

ZKPU in **AIC** Form Factor

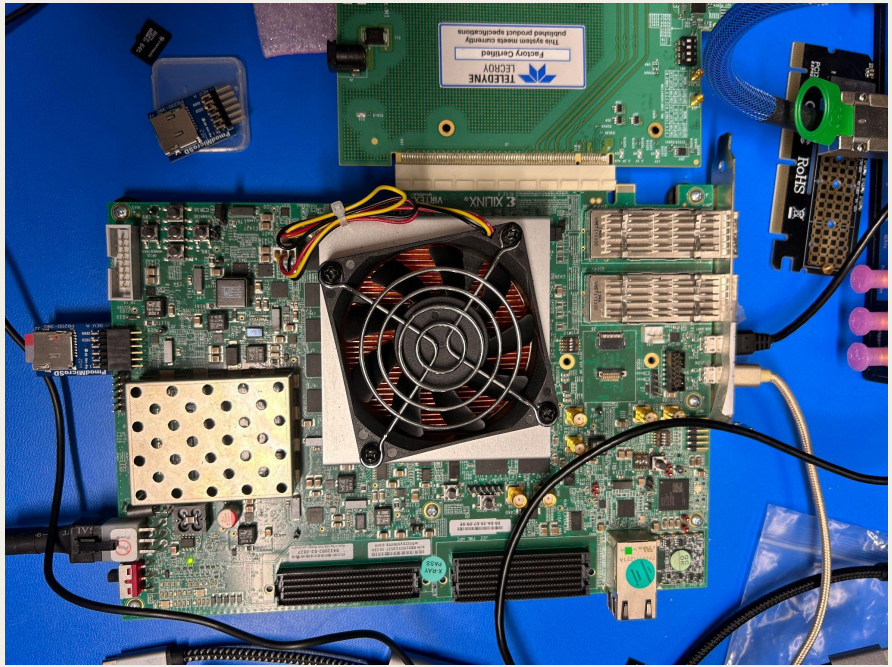
Plug-and-Play



ZKPU in a Datacenter Setup

- Data Center Ready: ZKPU + NVMe + NVMe-oF = Custom ASIC + DIZK = Proof-as-a-Service (PaaS).
- Integration: NVMe + Nvidia GPUDirect = P2P connect with GPU & SSDs
- Open-Source: Built on UC Berkeley's **Chipyard** project, fostering innovation through an open-source design.

Proof of Concept



Demo: https://docs.open-proof.com/demo/nvme-msm
FPGA: XIlinx VCU118
CPU: RISC-V
System: Chipyard based
Clock: 100Mhz
Data Transport Protocol: standard NVMe
Tools: Python script calling nvme-cli

N	Our_MSM_100 MHz (s)	Our_MSM_250MH z (s) (expected)	CycloneMSM (AWS F1 FPGA 250MHz) (s)	gnark-crypto (AWS F1 4x 2.3GHz) (s)
20	1.396	0.546	0.559	3.192

Addition Resources

- **Videos:**
  - **1-minute overview video**: https://www.youtube.com/watch?v=VUTBqpwyIAQ
  - **6-minute video** touching on more details: https://www.youtube.com/watch?v=qnRF5LgDOes
  - **33-minute deep dive video** on technical details: https://www.youtube.com/watch?v=xDHMmRuL32w
- **Contact us:**
  - Twitter: https://x.com/OpenProof_ZKP
  - Telegram ID: dingsen_2
  - Email: dingsen.shi@open-proof.com
- **OpenProof Website:** https://docs.open-proof.com/
- **NVMe-MSM FPGA Demo:** https://docs.open-proof.com/demo/nvme-msm