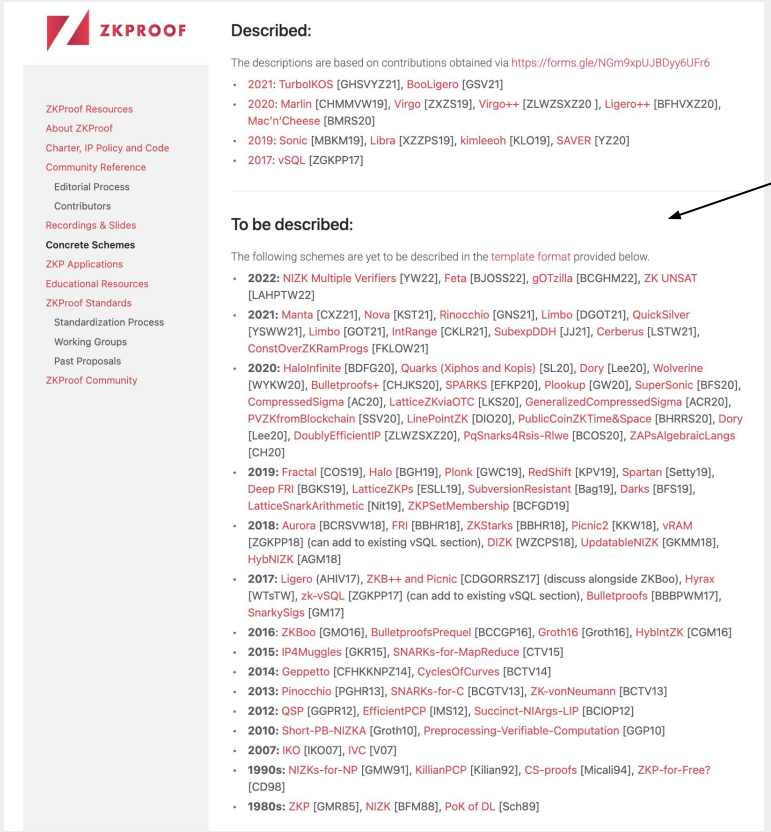
# ADVENTURES FROM FHE TO ZKP: DESIGNING + DEPLOYING THE VPU

COED Day | June 26th | Stanford

# CRYPTOGRAPHIC ALGORITHMS CHANGE FAST



ZKPROOF

ZKProof Resources
About ZKProof
Charter, IP Policy and Code
Community Reference
- Editorial Process
- Contributors

Recordings & Slides
**Concrete Schemes**
ZKP Applications
Educational Resources
ZKProof Standards
- Standardization Process
- Working Groups
- Past Proposals

ZKProof Community

## Described:

The descriptions are based on contributions obtained via https://forms.gle/NGm9xpUJBDyy6UFr6

- 2021: TurboIKOS [GHSVYZ21], BooLigero [GSV21]
- 2020: Marlin [CHMMVW19], Virgo [ZXZS19], Virgo++ [ZLWZSXZ20 ], Ligero++ [BFHVXZ20], Mac'n'Cheese [BMRS20]
- 2019: Sonic [MBKM19], Libra [XZZPS19], kimleeoh [KLO19], SAVER [YZ20]
- 2017: vSQL [ZGKPP17]

## To be described:

The following schemes are yet to be described in the template format provided below.

- **2022:** NIZK Multiple Verifiers [YW22], Feta [BJOSS22], gOTzilla [BCGHM22], ZK UNSAT [LAHPTW22]
- **2021:** Manta [CXZ21], Nova [KST21], Rinocchio [GNS21], Limbo [DGOT21], QuickSilver [YSWW21], Limbo [GOT21], IntRange [CKLR21], SubexpDDH [JJ21], Cerberus [LSTW21], ConstOverZKRamProgs [FKLOW21]
- **2020:** HaloInfinite [BDFG20], Quarks (Xiphos and Kopis) [SL20], Dory [Lee20], Wolverine [WYKW20], Bulletproofs+ [CHJKS20], SPARKS [EFKP20], Plookup [GW20], SuperSonic [BFS20], CompressedSigma [AC20], LatticeZKviaOTC [LKS20], GeneralizedCompressedSigma [ACR20], PVZKfromBlockchain [SSV20], LinePointZK [DIO20], PublicCoinZKTime&Space [BHRRS20], Dory [Lee20], DoublyEfficientIP [ZLWZSXZ20], PqSnarks4Rsis-Rlwe [BCOS20], ZAPsAlgebraicLangs [CH20]
- **2019:** Fractal [COS19], Halo [BGH19], Plonk [GWC19], RedShift [KPV19], Spartan [Setty19], Deep FRI [BGKS19], LatticeZKPs [ESLL19], SubversionResistant [Bag19], Darks [BFS19], LatticeSnarkArithmetic [Nit19], ZKPSetMembership [BCFGD19]
- **2018:** Aurora [BCRSVW18], FRI [BBHR18], ZKStarks [BBHR18], Picnic2 [KKW18], vRAM [ZGKPP18] (can add to existing vSQL section), DIZK [WZCPS18], UpdatableNIZK [GKMM18], HybNIZK [AGM18]
- **2017:** Ligero (AHIV17), ZKB++ and Picnic [CDGORRSZ17] (discuss alongside ZKBoo), Hyrax [WTsTW], zk-vSQL [ZGKPP17] (can add to existing vSQL section), Bulletproofs [BBBPWM17], SnarkySigs [GM17]
- **2016:** ZKBoo [GMO16], BulletproofsPrequel [BCCGP16], Groth16 [Groth16], HybIntZK [CGM16]
- **2015:** IP4Muggles [GKR15], SNARKs-for-MapReduce [CTV15]
- **2014:** Geppetto [CFHKKNPZ14], CyclesOfCurves [BCTV14]
- **2013:** Pinocchio [PGHR13], SNARKs-for-C [BCGTV13], ZK-vonNeumann [BCTV13]
- **2012:** QSP [GGPR12], EfficientPCP [IMS12], Succinct-NIArgs-LIP [BCIOP12]
- **2010:** Short-PB-NIZKA [Groth10], Preprocessing-Verifiable-Computation [GGP10]
- **2007:** IKO [IKO07], IVC [V07]
- **1990s:** NIZKs-for-NP [GMW91], KillianPCP [Kilian92], CS-proofs [Micali94], ZKP-for-Free? [CD98]
- **1980s:** ZKP [GMR85], NIZK [BFM88], PoK of DL [Sch89]

**SAME FOR FHE**

**NOT INCLUDING MANY INDUSTRY SYSTEMS**

### 2025

- Relaxed Functional Bootstrapping: A New Perspective on BGV and BFV Bootstrapping by **Zeyu Liu** on **Feb 20th, 2025**
- General Functional Bootstrapping using CKKS by **Yuriy Polyakov** on **Jan 16th, 2025**

### 2024

- …pments in SNARKs and Their Connection to FHE by **Dan Boneh** on **Dec 5th, 2024**
- …ristic for Mapping Boolean Circuits to Functional Bootstrapping by **Sergiu Carpov** on **Nov 21st, 2024**
- Privacy-Preserving Graph ML with FHE for Collaborative Anti-Money Laundering by **Fabrianne Effendi** on **Nov 14th, 2024**
- Faster NTRU-based Bootstrapping in less than 4 ms by **Zhihao Li** on **Oct 24th, 2024**
- Designing a General-Purpose 8-bit (T)FHE Processor Abstraction by **Daphné Trama** on **Nov 17th, 2024**
- Private and Secure Fuzzy Name Matching by **Harsh Kasyap and Ugur Atmaca** on **Oct 10th, 2024**
- Concrete ML - Machine Learning on Encrypted Data by **Andrei Stoian** on **Sept 26th, 2024**
- New Secret Keys for Enhanced Performance in (T)FHE by **Loris Bergerat** on **Sept 12th, 2024**
- Practical q-IND-CPA-D-Secure Approximate Homomorphic Encryption by **Lea Nürnberger** on **Jul 18th, 2024**
- FHE Beyond IND-CCA1 Security by **Jérôme Nguyen** on **Jul 11th, 2024**
- Greco: Fast Zero-Knowledge Proofs for Valid FHE RLWE Ciphertexts Formation by **Enrico Bottazzi** on **Jun 27th, 2024**
- FHE: Past, Present and Future by **Craig Gentry** on **Jun 13th, 2024**
- Functional bootstrapping for FV style cryptosystems by **Seonhong Min** on **May 30th, 2024**
- Fregata: Faster Homomorphic Evaluation of AES via TFHE by **Benqiang Wei** on **May 9th, 2024**
- On the Concrete Security of Approximate FHE Schemes with Noise-Flooding Countermeasures by **Hunter Kippen** on **May 2nd, 2024**
- Convolution-friendly Image Compression in FHE by **Sergi Rovira** and **Axel Mertens** on **Apr 26th, 2024**
- Homomorphic Logic Gates and Integrated Circuits Designs and Applications by **Song Bian** on **Apr 11th, 2024**
- Fast Blind Rotation for Bootstrapping FHEs by **Dai Yiran** on **Mar 7th, 2024**
- Simpler and Faster BFV Bootstrapping for Arbitrary Plaintext Modulus from CKKS by **Jinyeong Seo** on **Feb 29th, 2024**
- A New Perspective on Key Switching for BGV-like Schemes by **Johannes Mono** on **Feb 22nd, 2024**
- Towards Practical Transciphering for FHE with Setup Independent of the Plaintext Space by **Jeongeun Park** on **Feb 8th, 2024**
- Designs for practical SHE schemes based on Ring-LWR by **Erin Hales** on **Jan 25th, 2024**
- High-precision RNS-CKKS on small word-size architectures by **Duhyeong Kim** on **Jan 11th, 2024**
- Efficient Pruning for Machine Learning under Homomorphic Encryption by **Subhankar Pal** on **Jan 4th, 2024**

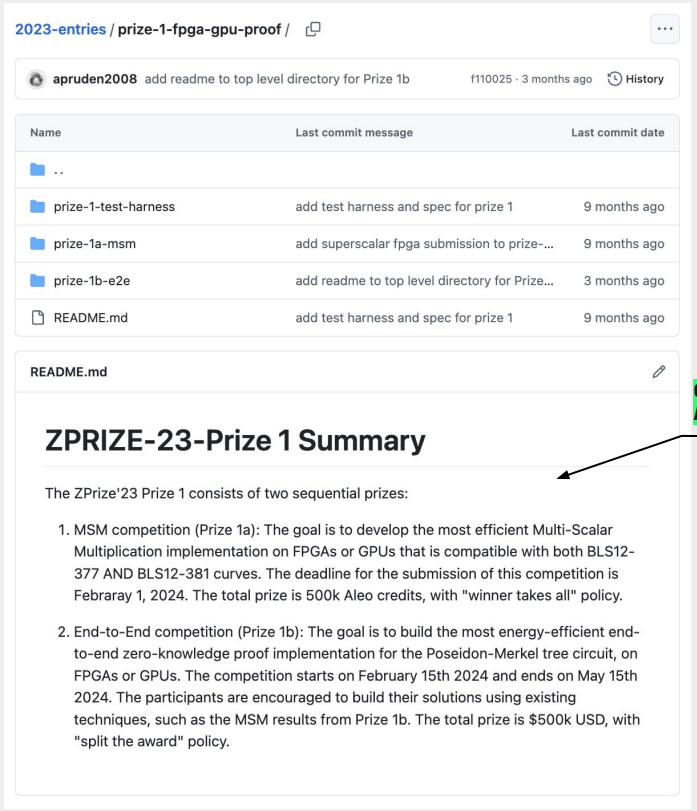
### 2023

- Lattigo v5: Deep Dive by **Jean-Philippe Bossuat** on **Dec, 12th, 2023**
- Crypto Dark Matter on the Torus: Oblivious PRFs from shallow PRFs and FHE by **Alex Davidson** on **Nov, 23rd, 2023**
- FHE ring packing - affordable and convenient by **Jaehyung Kim** on **Nov 2nd, 2023**
- HEIR: A foundation for FHE compilers by **Jeremy Kun** on **Oct 19th, 2023**
- Homomorphic Polynomial Evaluation using Galois structure and application to BFV bootstrapping by **Simon Pohmann** on **Oct 5th, 2023**

# THE ACCELERATION PARADIGM

ACADEMIC ACCELERATION

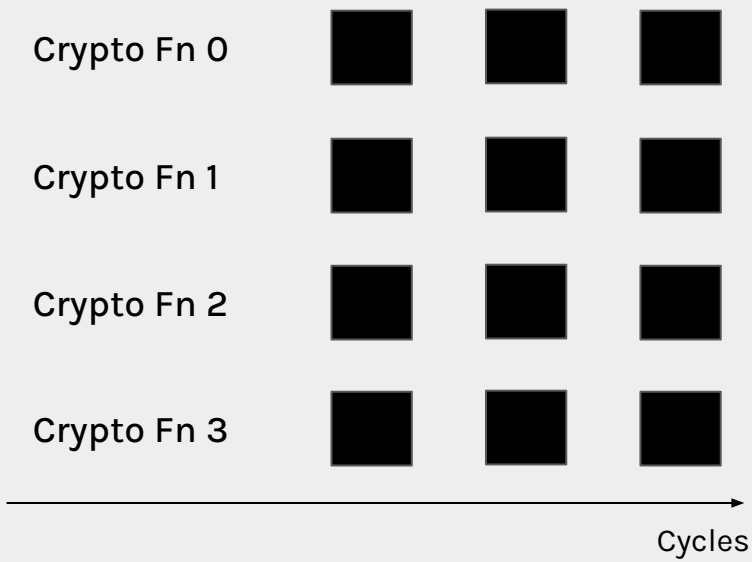

COMPETITIVE ACCELERATION



# ACCELERATION VIA GPUs or ASICs?

**Not built for workload (e.g. GPU)**
(x% **area** used y% of the **time**)

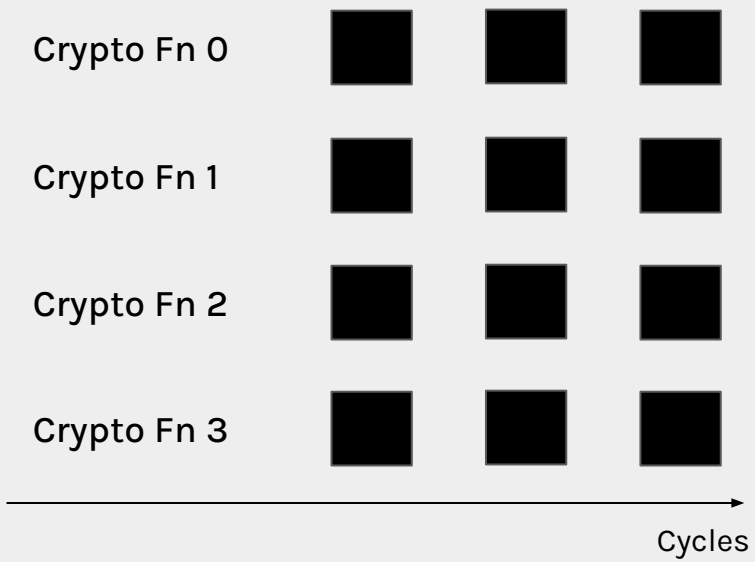

**Built specifically for workload (FF/GP)**
(**GOAL**: 100% **area** used 100% of the **time**)



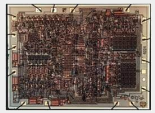
# ACCELERATION VIA GPUs or ASICs?

**Not built for workload (e.g. GPU)**
(x% **area** used y% of the **time**)


**Built specifically for workload (FF/GP)**
(**GOAL**: 100% **area** used 100% of the **time**)



# EVERY GENERATION OF COMPUTING IS DEFINED BY A BREAKTHROUGH PROCESSOR



**1970s**

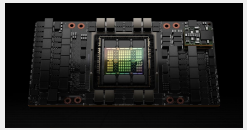
## CPU

Personal Computing


**1990s**

## GPU

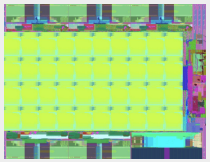
3d Graphics



**2010s**

## GPU/TPU

Artificial Intelligence

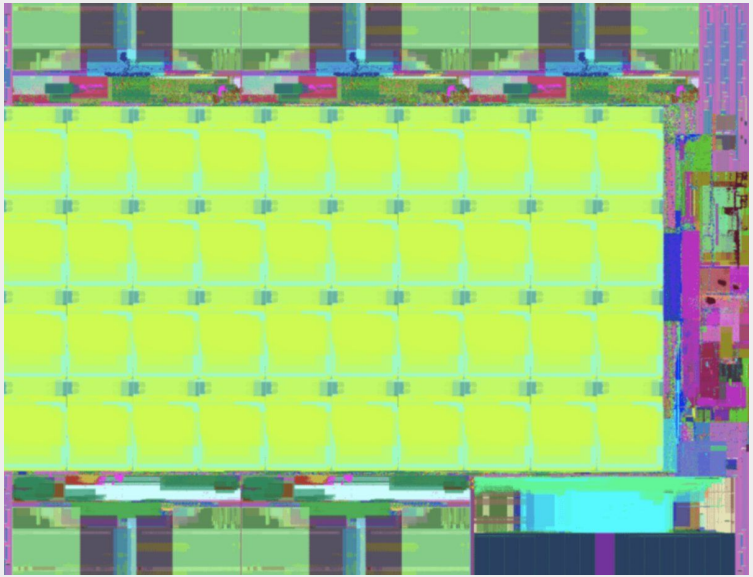


**2025**

## VPU

Next-Gen Cryptography
(FHE, ZKP, 🤖)

# OVER THE PAST TWO YEARS, WE DESIGNED JUST THAT

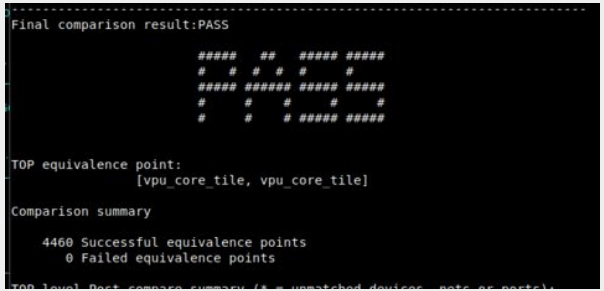


→ **First programmable cryptography chip in the world!**

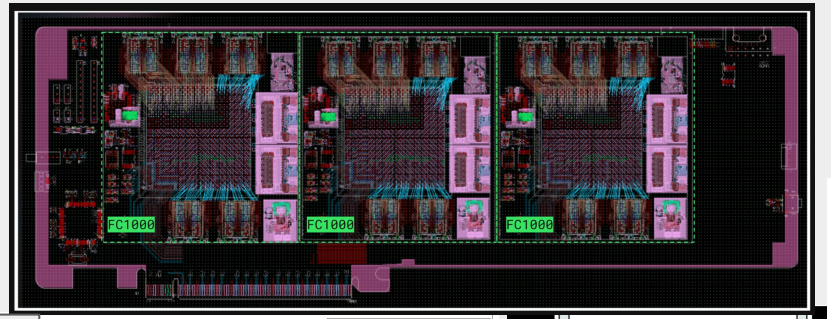
→ TSMC 12nm (12FFC)

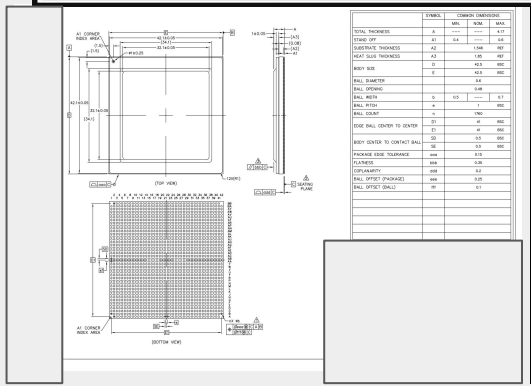
→ 1536-bit SIMD vector lanes for cryptography

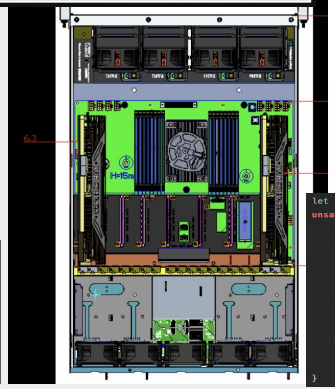
→ Custom instructions for number theory



# IT TAPED OUT…



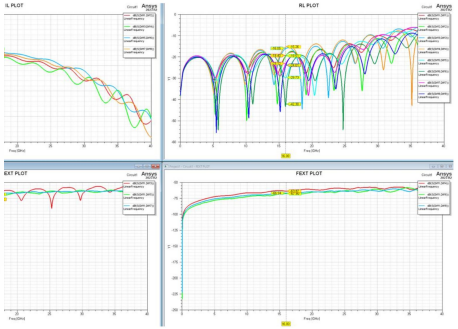


[SECURITY C] Fabric / Turing / [redacted] / MT-[redacted]-001-11 Mask Database Check Result

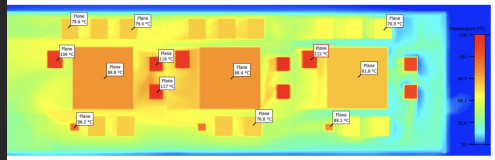
Summarize this email

TSMC Integrated Tape-Out Service <[redacted]@tsmc.com>

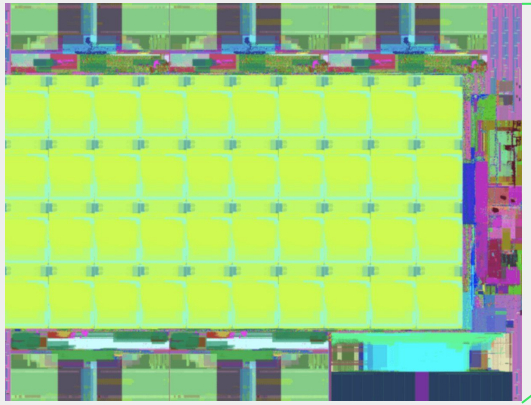
[redacted]

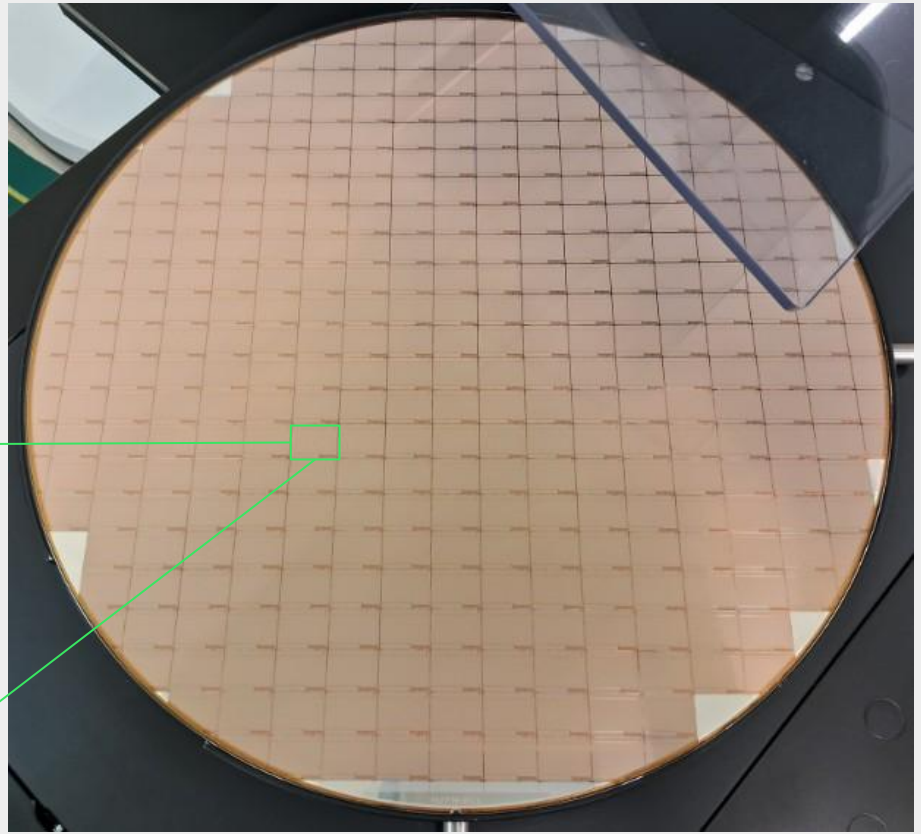



```
let retval = offloaded_commit(value_as_offloadable.clone(), rate_bits.clone());
unsafe {
    let babybear_commitment: *const p3_symmetric::Hash<BabyBear, BabyBear, 8
    > = &retval.0;
    let c = babybear_commitment as *const Commitment;
    let concrete_prover_data: *const FieldMerkleTree<
        BabyBear,
        BabyBear,
        DenseMatrix<BabyBear>,
        8,
    > = &retval.1;
    let prover_data = concrete_prover_data as *const ProverData;
    ((*c).clone(), (*prover_data).clone())
}
```



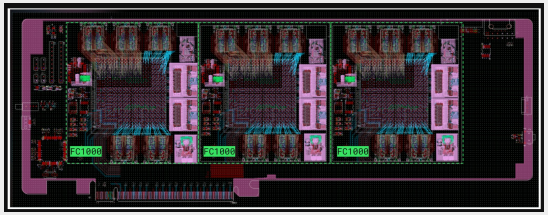
# … AND CAME BACK!





# GEN1 for ZK: WHO WOULD WIN?

## Fabric VPU



12nm technology
1 billion transistors
1 GHz
Dozens of ppl team

## Nvidia RTX 5090

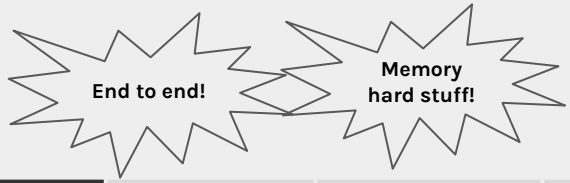

3nm technology
76 billion transistors
2.25 GHz
10000s of ppl team

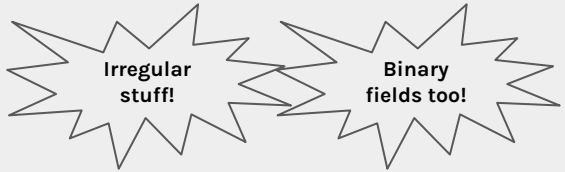
# PRELIMINARY BENCHMARKS VS RTX 4090

| Algorithm | End-to-end RISC Zero | NTT | Merkle Tree | EvalCheck | Tower Field Sum Checks |
|---|---|---|---|---|---|
| Performance per $ | 6.4x | 8.7x | 3.0x | 11.1x | 10x |

**Measured by TCO, $/hr**
**Note:** MSRP might change as demand grows

# BY THE WAY, THESE ARE NOT EASY




| Algorithm | End-to-end RISC Zero | NTT | Merkle Tree | EvalCheck | Tower Field Sum Checks |
|---|---|---|---|---|---|
| Performance per $ | 6.4x | 8.7x | 3.0x | 11.1x | 10x |

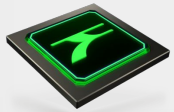
**Measured by TCO, $/hr**
**Note:** MSRP might change as demand grows

# A FULL-STACK SOLUTION FOR OPERATORS

## FABRIC

### HARDWARE [2025]

CHIP
FC1000



SYSTEM
VPU 8060 / FERMAT

### SOFTWARE [2025]

- GRAPH COMPILER
- KERNELS
- LIBRARIES

### IaaS → SaaS [2025+]

- VPU CLOUD
- DATA CENTER OPERATIONS THRU PARTNERS

polygon

RISC ZERO

## NVIDIA

### HARDWARE [1995]

CHIP

SYSTEM

### SOFTWARE [2007]

- ALGORITHMS
- LIBRARIES

## CoreWeave

### IaaS / PaaS / SaaS [2017]

- SPECIALIZED CLOUD
- DATA CENTER OPERATIONS
- PLATFORM OFFERINGS

# INTRODUCING THE FABRIC FERMAT SERVER

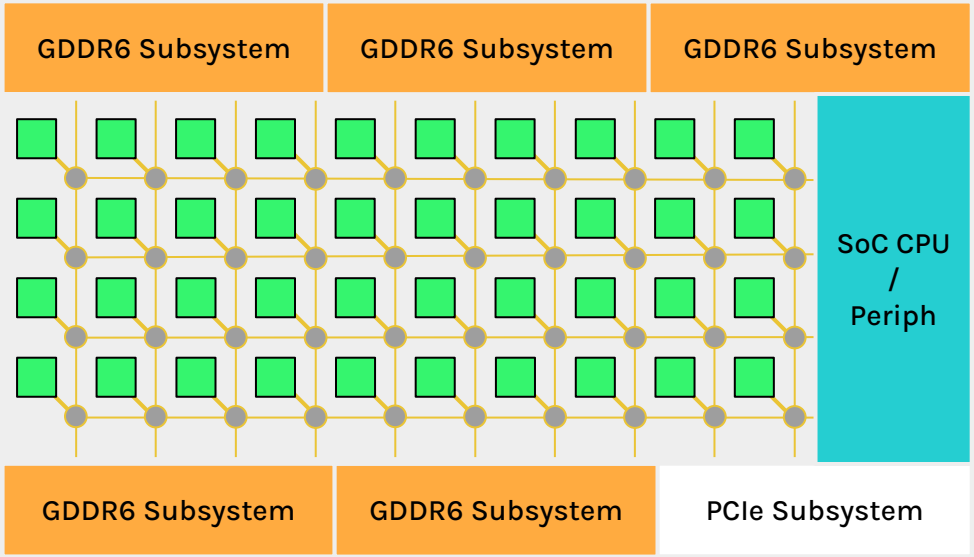
*Powered by the Verifiable Processing Unit*

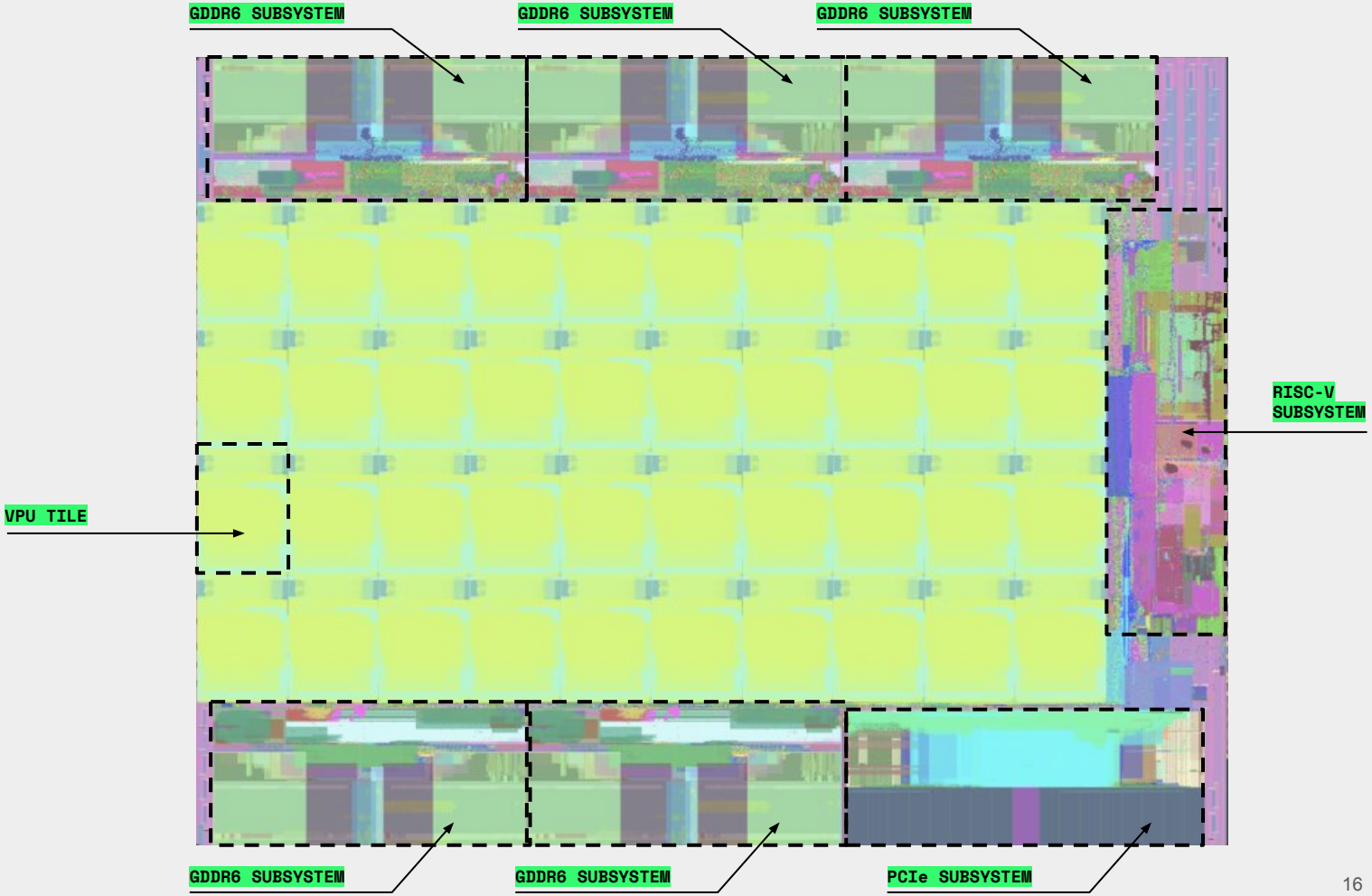
# THE ARCHITECTURE UNDER THE HOOD

**40 VPU Tiles** with **Big-Integer, Finite Field ALUs** Working in **Parallel**

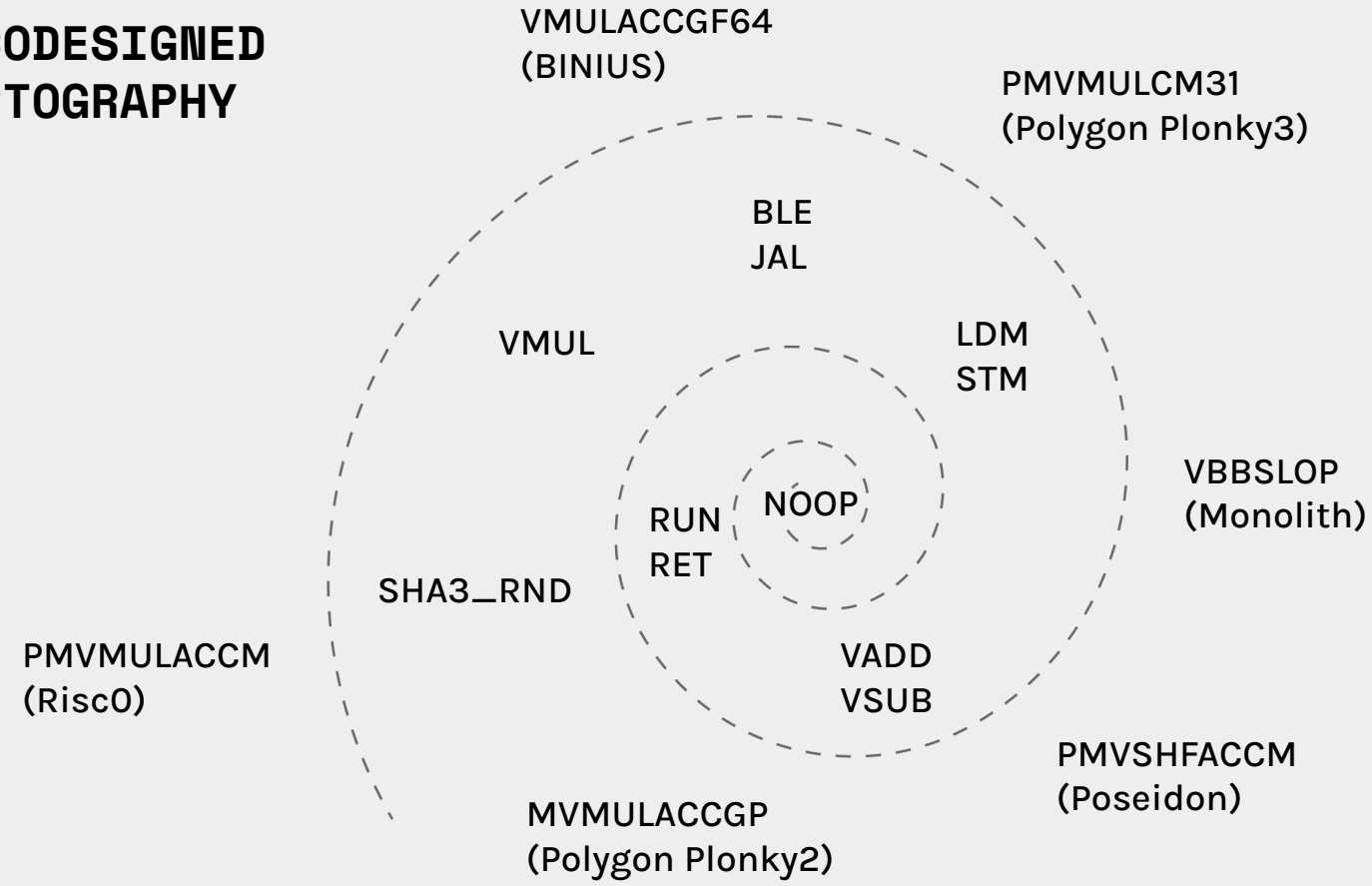
**RISC-V SoC CPU complex** on the Chip

Ultra-Fast **Network-on-Chip** to access Tiles and **GDDR6 Memory**

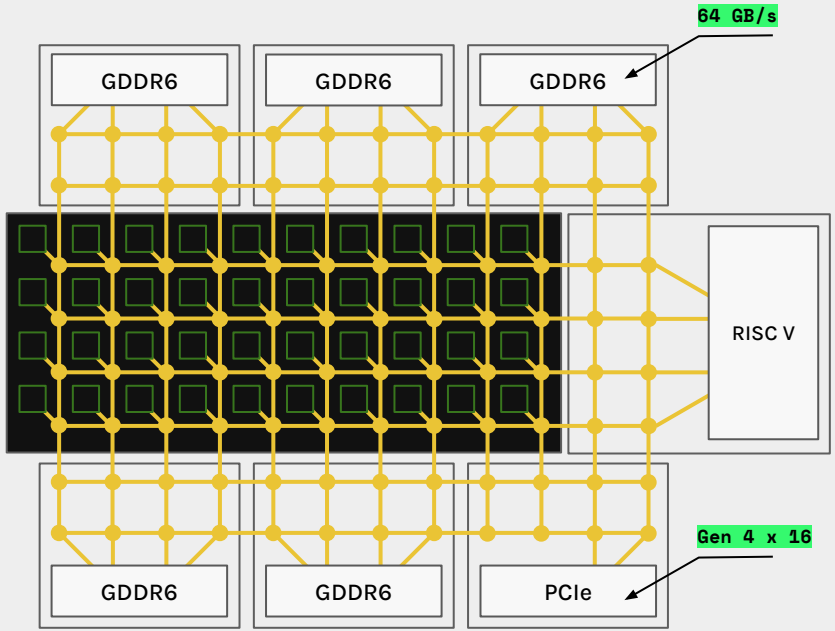



GDDR6 SUBSYSTEM
GDDR6 SUBSYSTEM
GDDR6 SUBSYSTEM
RISC-V SUBSYSTEM
VPU TILE
GDDR6 SUBSYSTEM
GDDR6 SUBSYSTEM
PCIe SUBSYSTEM

# AN ISA CODESIGNED FOR CRYPTOGRAPHY



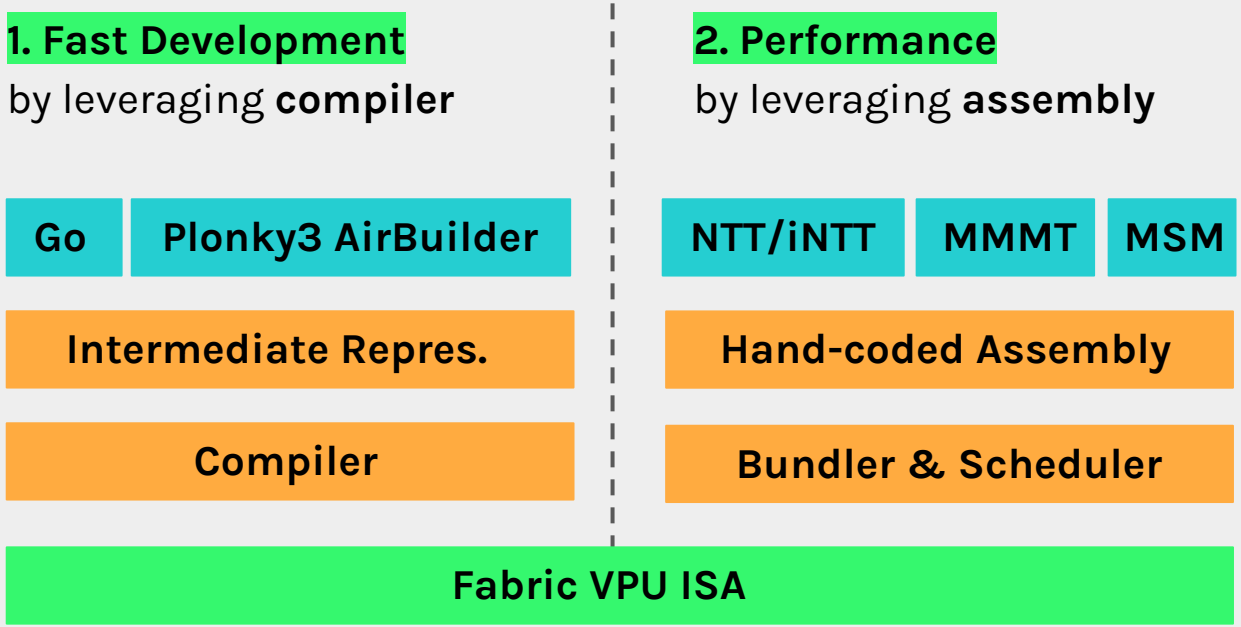
# A NETWORK CODESIGNED WITHOUT PCIE BOTTLENECK



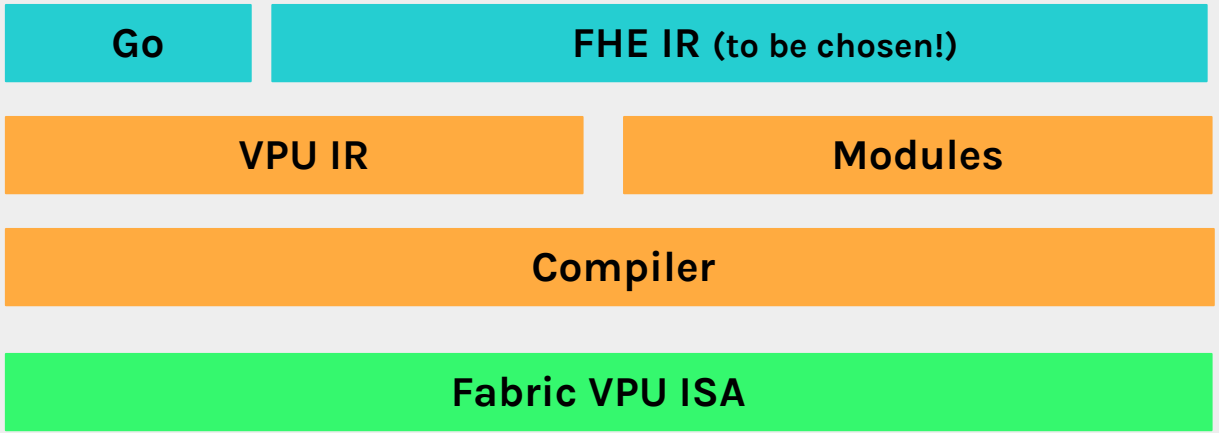

Each **Tile** can introduce and extract **32GB/s into the NoC**
**GDDR** subsystems introduce and extract **128GB/s into the NoC**
**PCIe** subsystems introduce and extract **64GB/s into the NoC**
**Random Number Generator** can deliver **32GB/s into the NoC**

11 TB/s Network-on-Chip (NOC)
Data movement managed by distributed programmable DMA

# SOFTWARE DESIGNED WITH TWO PURPOSES



# WE LOOK FORWARD TO SUPPORTING FHE!

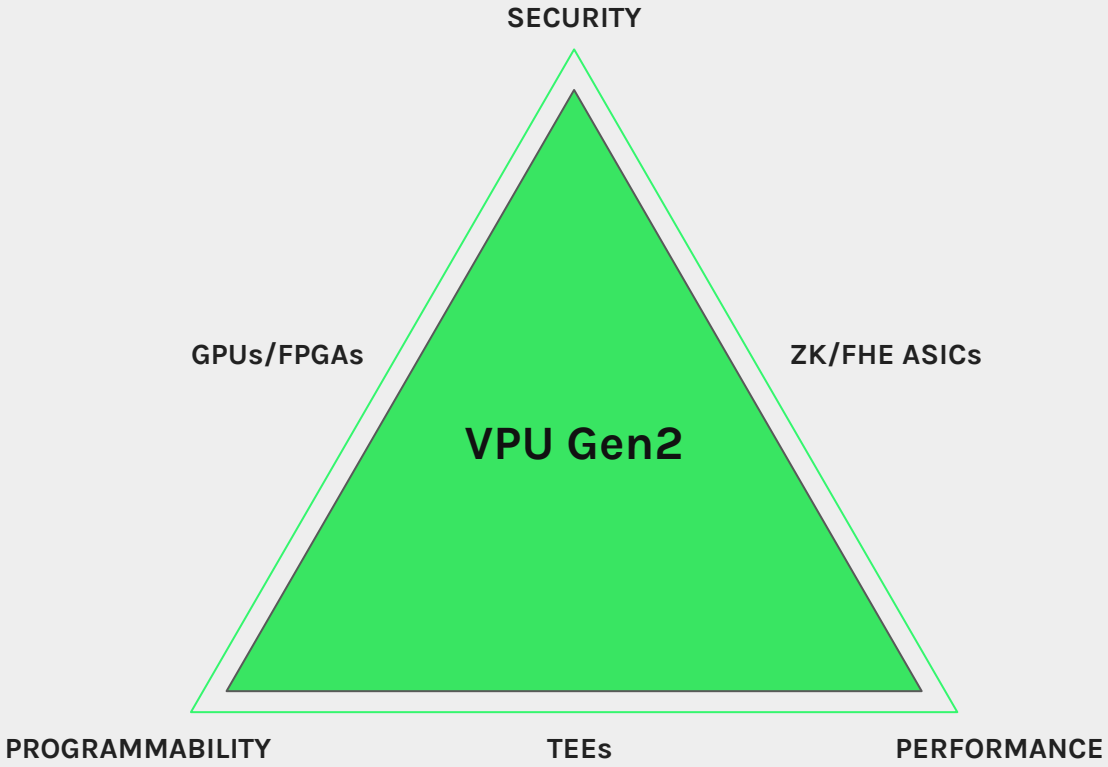


# HOW CAN WE FLEXIBLY IMPLEMENT ZKPS?

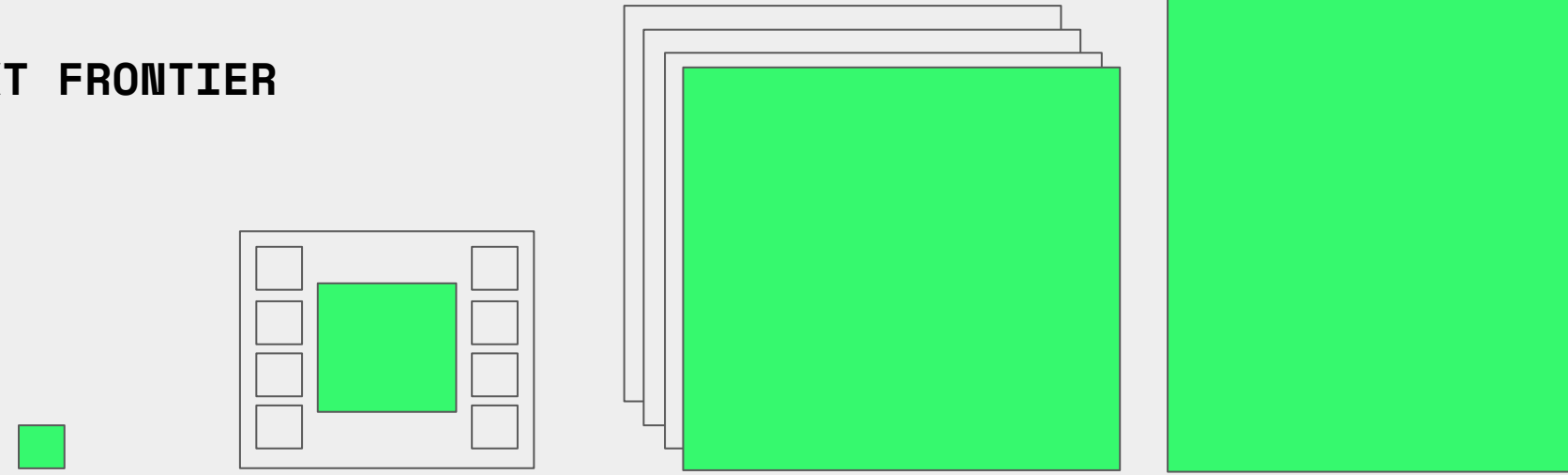

Commitments are fixed blocks
Other blocks are circuit-dependent
Novel provers use new blocks

handle with a library
handle with a compiler
handle with a high-level language

# FC2000: FHE + ZK + TEEs



# THE NEXT FRONTIER



**2025**

## FC1000

12nm, 1st gen

**2027**

## FC2000

4nm, 2nd gen

**2029**

## FC3000

2nm, 3rd gen

**2031**

## FC4000

1.4A, 4th gen

# LET'S WORK TOGETHER TO BUILD THIS FUTURE!

**Real time** zero-knowledge proofs
**Real time** fully homomorphic encryption
**Real time** witness encryption
**Real time** functional encryption
**Real time** indistinguishability obfuscation

Programmable, General Purpose, Scalable.