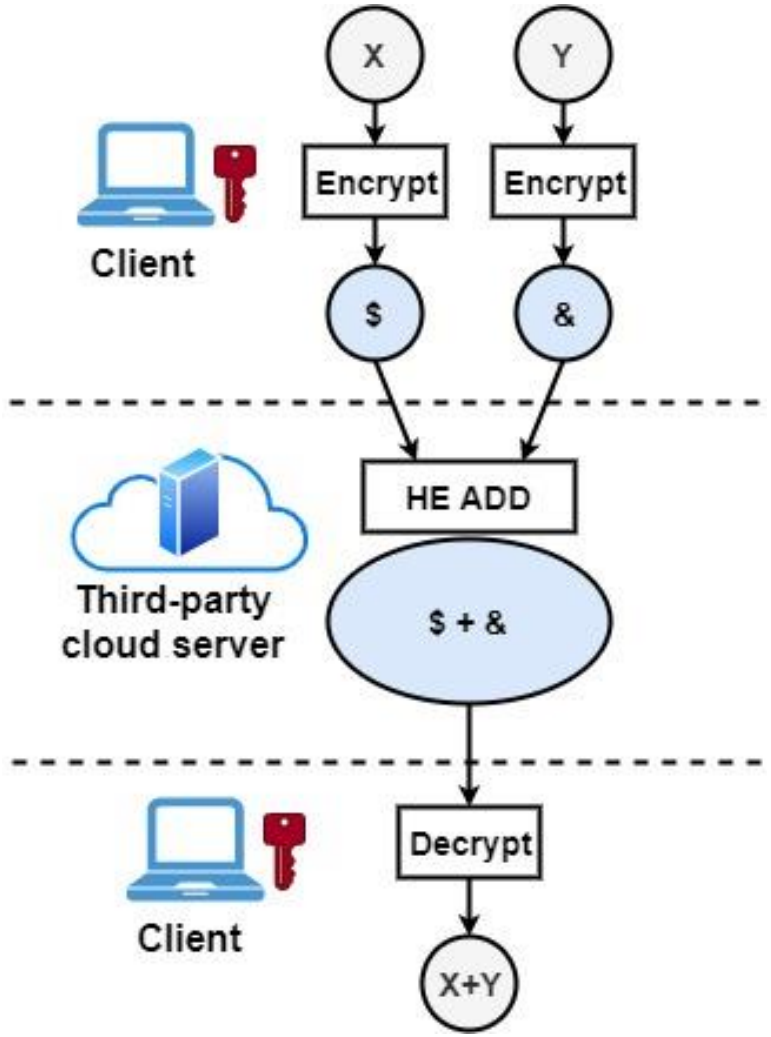
Affordable and Practical FPGA-based Fully Homomorphic Encryption

Rashmi Agrawal
CTO, CipherSonic Labs
rashmi@ciphersoniclabs.io

Presented at NIST WPEC 2024 on September 25

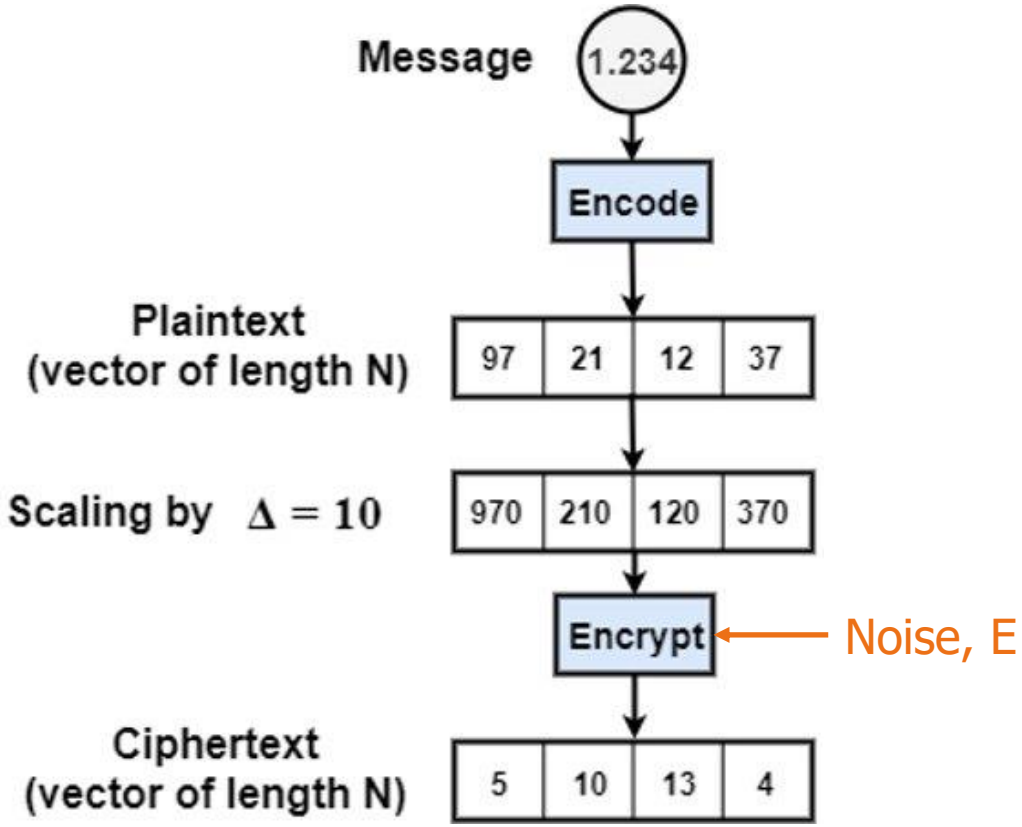
Fully Homomorphic Encryption (FHE)

- Allows computations on encrypted data
  - Data is encrypted end-to-end
- Allows secure outsourcing of computations on third-party cloud servers
  - Server has no access to the data in plaintext form
- Allows protection of secret key
  - Server has no access to the key



Various FHE Schemes

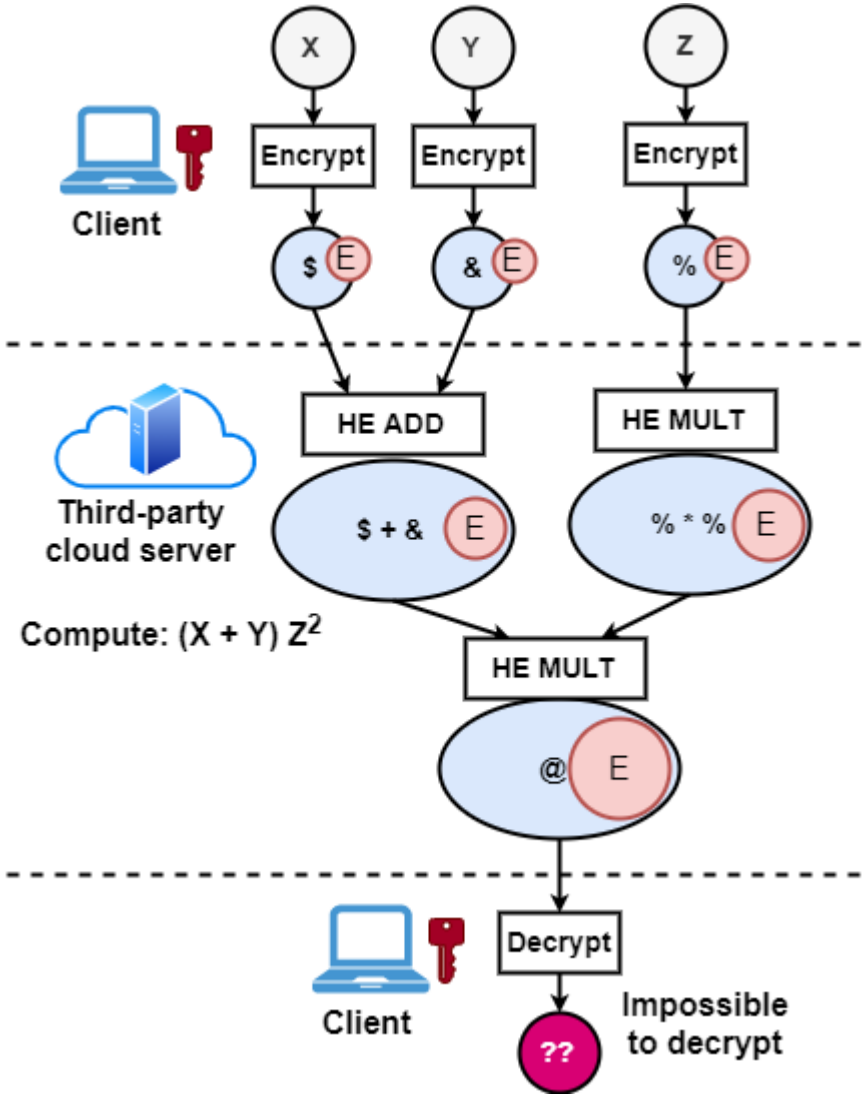
- Binary arithmetic schemes
  - Allow operations on bits
  - Operations like comparisons are fast
  - TFHE, FHEW schemes
- Exact arithmetic schemes
  - Allow operations on integers
  - Operations like add and multiply are fast
  - BGV, BFV schemes
- Approximate arithmetic schemes
  - Allow operations on real numbers
    - Enabling real-time privacy-preserving applications
  - CKKS scheme



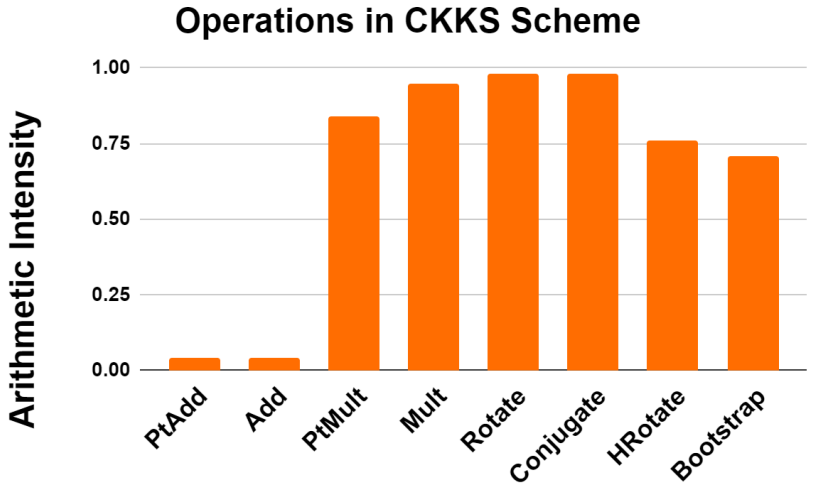
The Challenges

- Noise growth
  - Add doubles the noise
    - $E + E = 2E$
  - Mul squares the noise
    - $E * E = E^2$

- To tackle noise growth
  - Large parameters
    - $N = 2^{17}$ and $\log Q = 2240$
  - Bootstrapping
    - De-noises the ciphertext



The Challenges



- Large parameters
  - $N = 2^{17}$ and $\log Q = 2240$
  - Ciphertext size = 73.4 MB
  - 40 MB last level cache in commercial systems like CPU/GPU
  - Issues:
    - Frequent main memory access → Increased main memory bandwidth requirement
    - Underutilization of compute resources
- Bootstrapping

Compute platform	Bootstrapping runtime	Speed up
CPU [1]	8.7 minutes	-
GPU [2]	1.5 seconds	350x
FPGA [3]	15.6 ms	100x
ASIC [4]	0.39 ms	40x

  - Up to 95% of run time in an application spent in bootstrapping

The Current CKKS Acceleration Efforts

- CPU / GPU / FPGA / Hardware
  - FPGAs have emerged as one of the viable hardware acceleration platforms

2020

FPGA
HEAX, Riazi et al.

2021

CPU
Non-sparse keys, Bossuat et al.

GPU
Over 100x, Jung et al.

ASIC
F1, Samardzic et al.

2022

ASIC
1) BTS, Kim et al.
2) CraterLake, Samardzic et al.
3) ARK, Kim et al.

2023

GPU
GME, Shivdikar et al.

FPGA
1) FAB, Agrawal et al.
2) Poseidon, Yang et al.

ASIC
SHARP, Kim et al.

2024

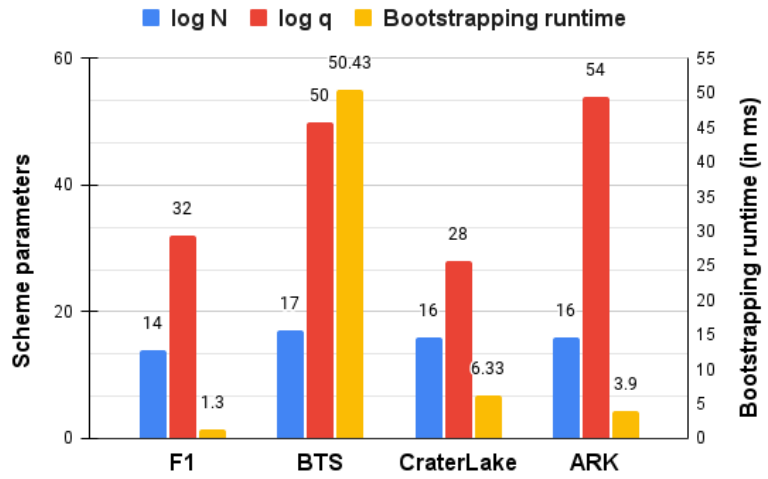
GPU
Cheddar, Kim et al.

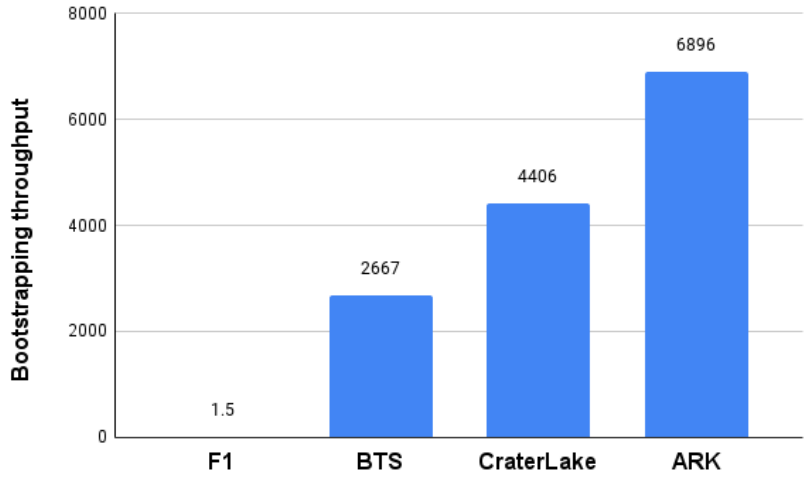
FPGA
HEAP, Agrawal et al.

Hardware Acceleration Efforts

$$\text{throughput} = \frac{n \cdot \log Q_1 \cdot \text{bp}}{\text{brt}}$$

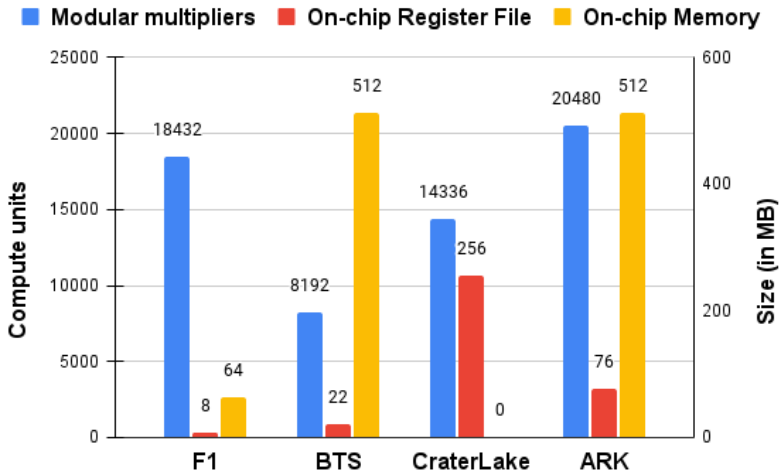
Practical parameters and runtime

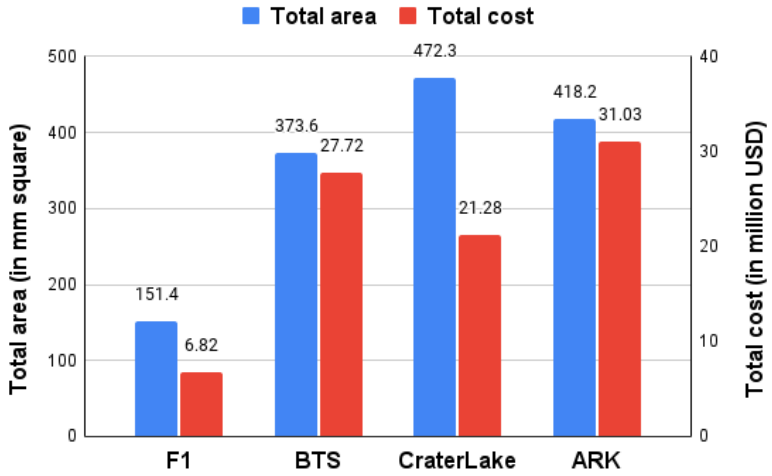




Improved throughput

Thousands of compute units and large on-chip memory





Large chip area and cost

CIPHERSONIC LABS

Memory-aware Design (MAD) Optimizations

- Reduce memory accesses during KeySwitch operation

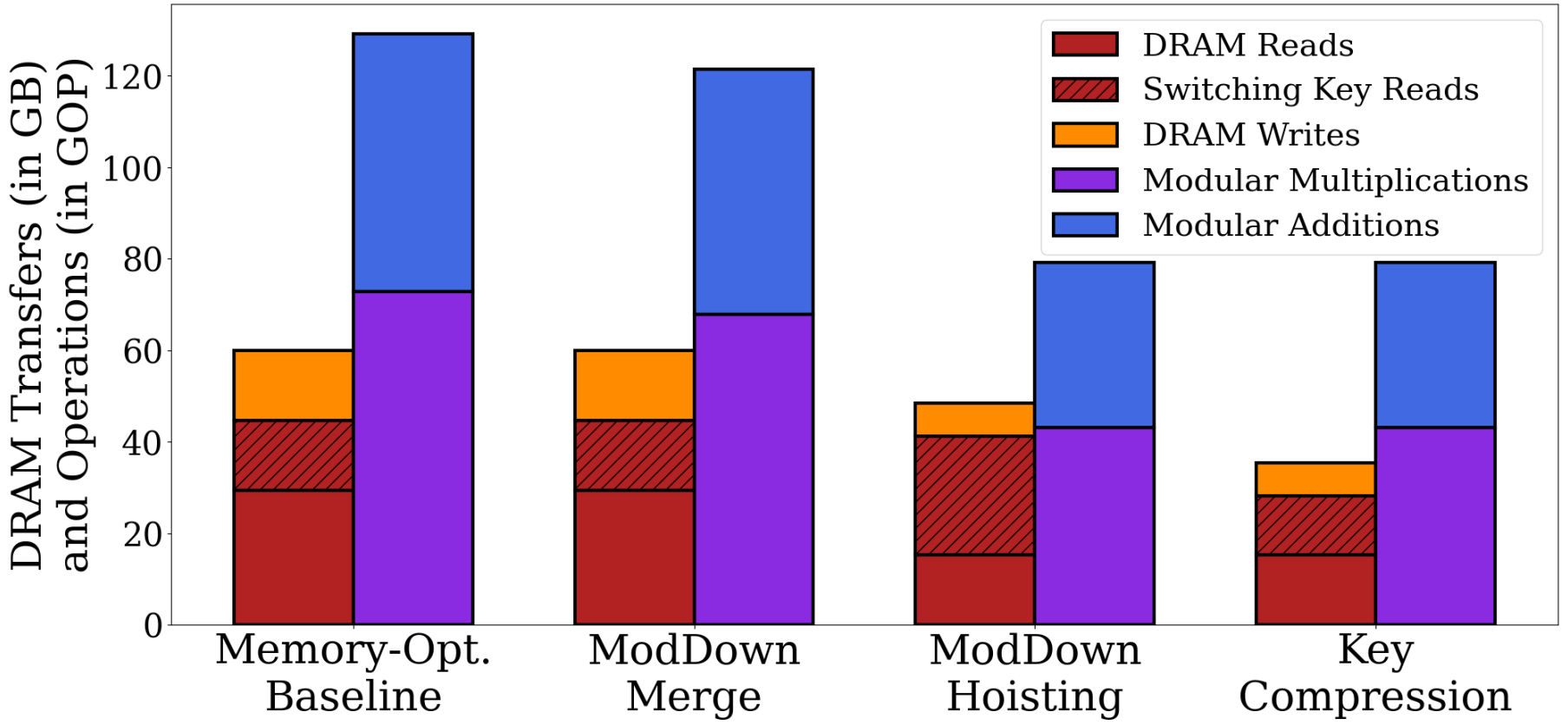
- Caching optimizations
  - Caching O(1) limbs
    - 1 MB on-chip memory
    - Compute as much as possible on a single limb → operation fusing
  - Caching O(β) limbs
    - Beta sized on-chip memory
  - Caching O(α) limbs
    - Alpha sized on-chip memory
  - Reordering limb computations
    - Builds on top of alpha limb caching optimization

- Reduce number of operations during bootstrapping

- Algorithmic optimizations
  - ModDown merge
    - Combining ModDown and Rescale in multiplication operation
  - ModDown hoisting
    - Back-to-back rotations without ModDown
  - Key Compression
    - Generate pseudorandom polynomials (half of the key switch keys) with a PRNG

Improvements

- Memory-aware design optimizations help reduce the DRAM transfers and number of operations



The Need

- Affordable
  - ASIC solutions are architecturally aggressive → large resources
  - Large chips → Expensive → long and painful design efforts
- Practical
  - CPU / GPU solutions provide flexibility but are inefficient
- Balanced hardware design
  - Compute vs. memory bandwidth
- Easy to deploy
  - Readily available inexpensive hardware platform in the cloud environment
    - commercial off-the shelf hardware

FAB

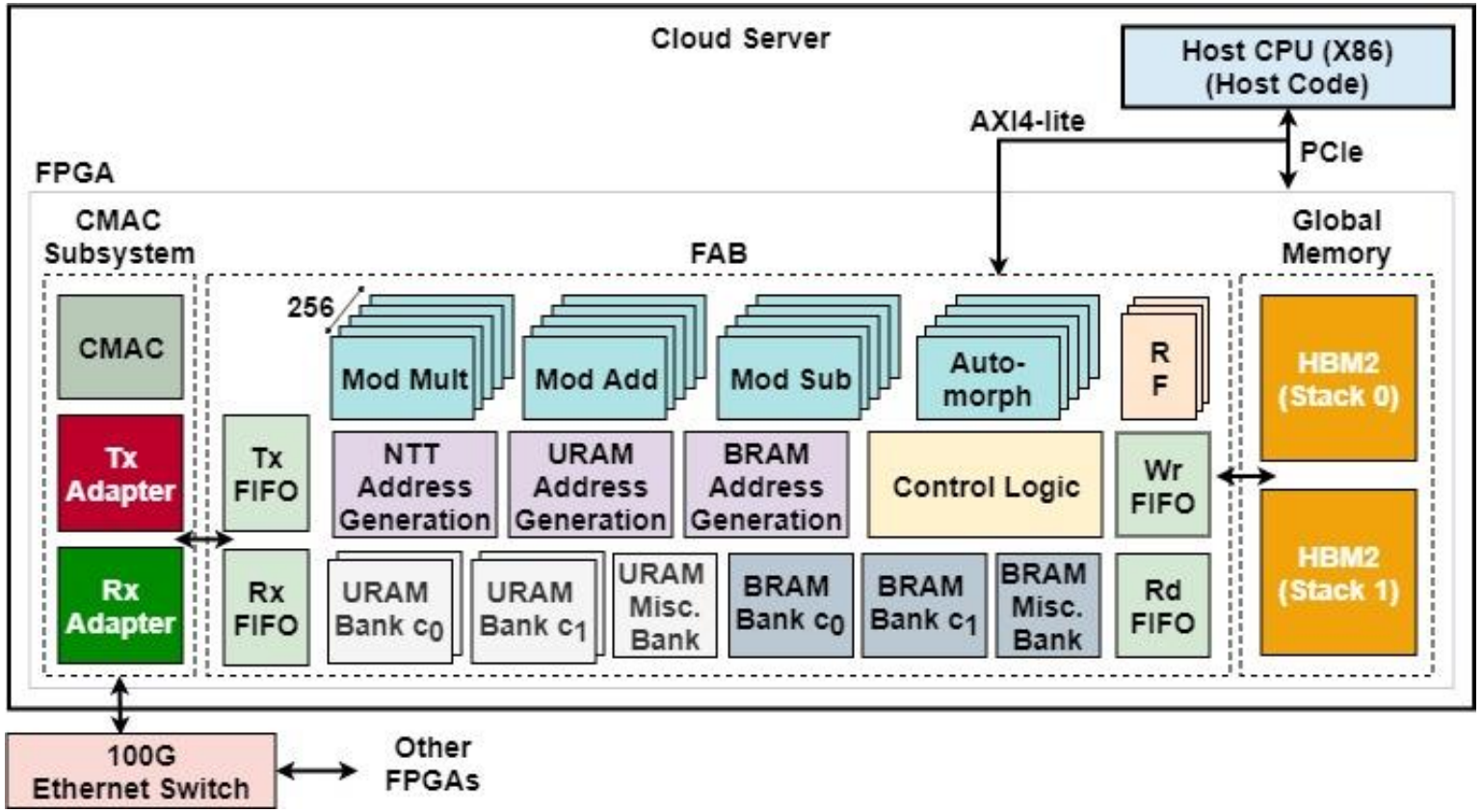
- A novel FGPA-based accelerator
  - Supports practical parameter set → $N = 2^{16}$ and $\log Q = 1728$
  - Performs first-ever fully-packed bootstrapping implementation on FPGA

- Highly resource efficient
  - Leverages 256 functional units with highly FPGA-tailored modular arithmetic units
  - Exploits maximal pipelining and parallelism to meet computational demands

- Balanced design with high compute throughput
  - Manages limited 43 MB on-chip memory through datapath modification, smart operation scheduling, on-chip memory management techniques

FAB

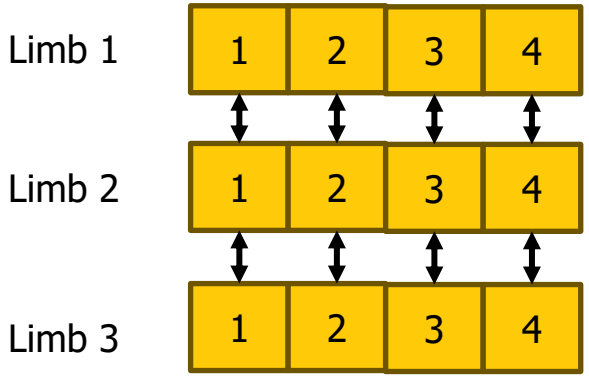
- Overall architecture contains four components
  - Host CPU
  - RTL code
  - HBM stacks
  - CMAC subsystem
- Mapped on Alveo U280 FPGA accelerator card
- Operating frequency
  - 300 MHz

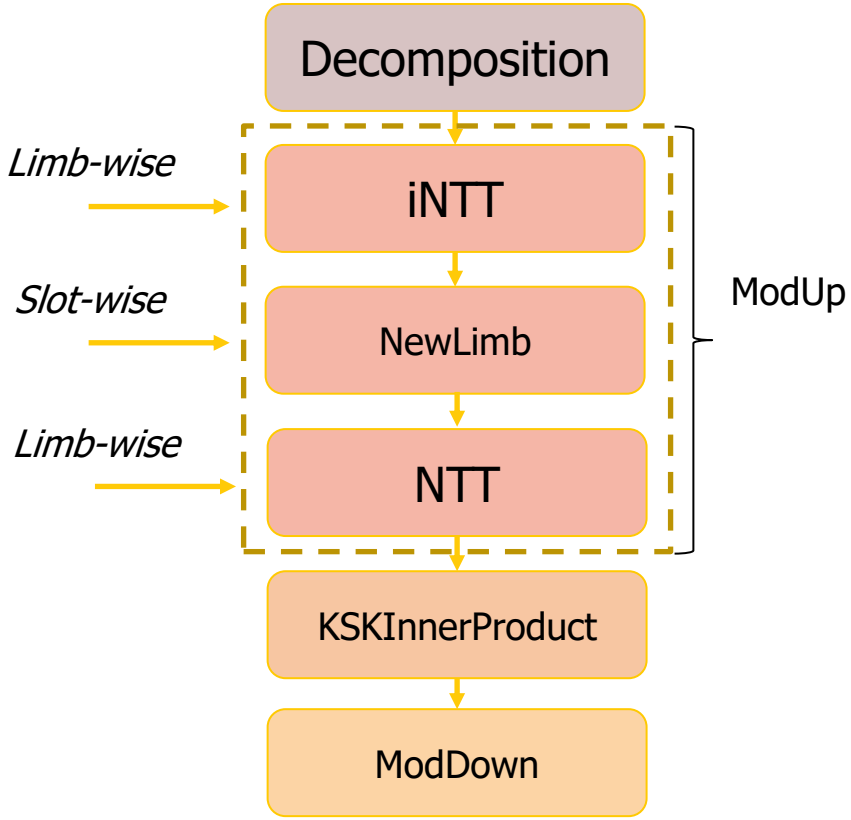


FAB

- Performance
  - Bootstrapping
    - 213x faster than CPU [1]
    - 1.5x faster than GPU [2]
  - Logistic regression model training
    - 456x faster than CPU [1]
    - 9.5x faster than GPU [2]
- Performance limited by the performance of bootstrapping
  - Too many KeySwitch operations in bootstrapping
  - Parallelization is a challenge with inter-limb data dependency

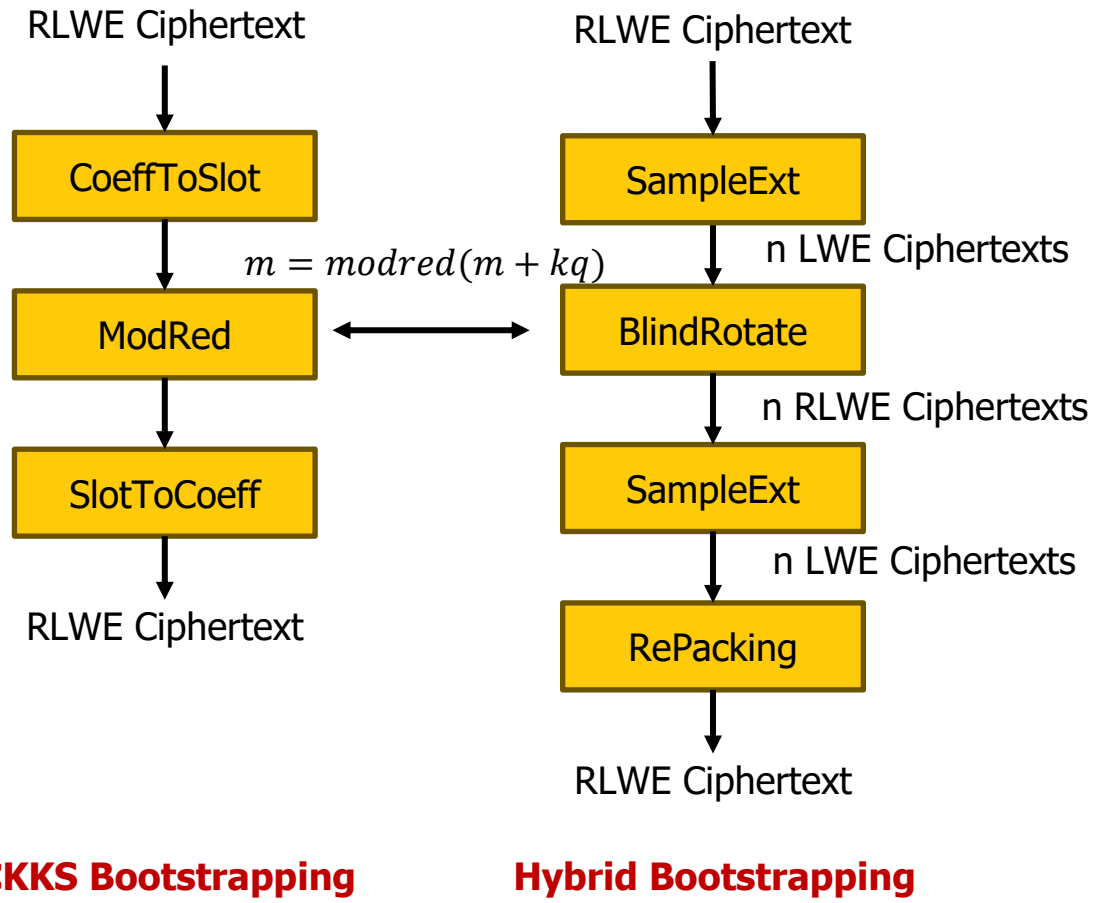
Slot-wise Interaction between the limbs makes them hard to parallelize





Sequence of sub-operations in KeySwitch

HEAP

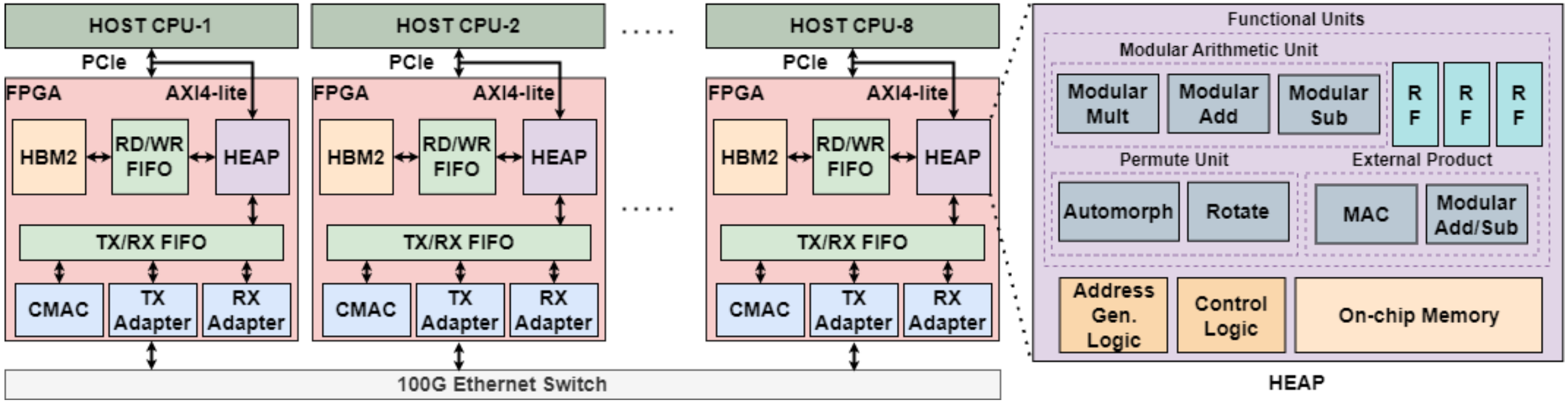


CKKS Bootstrapping **Hybrid Bootstrapping**

- Parallelizable bootstrapping using scheme switching
  - LWE ciphertexts can be processed in parallel using TFHE
  - Bootstrapping using smaller parameters $N = 2^{13}$
- Main advantage:
  - Amount of data read from main memory is 18x lower

HEAP

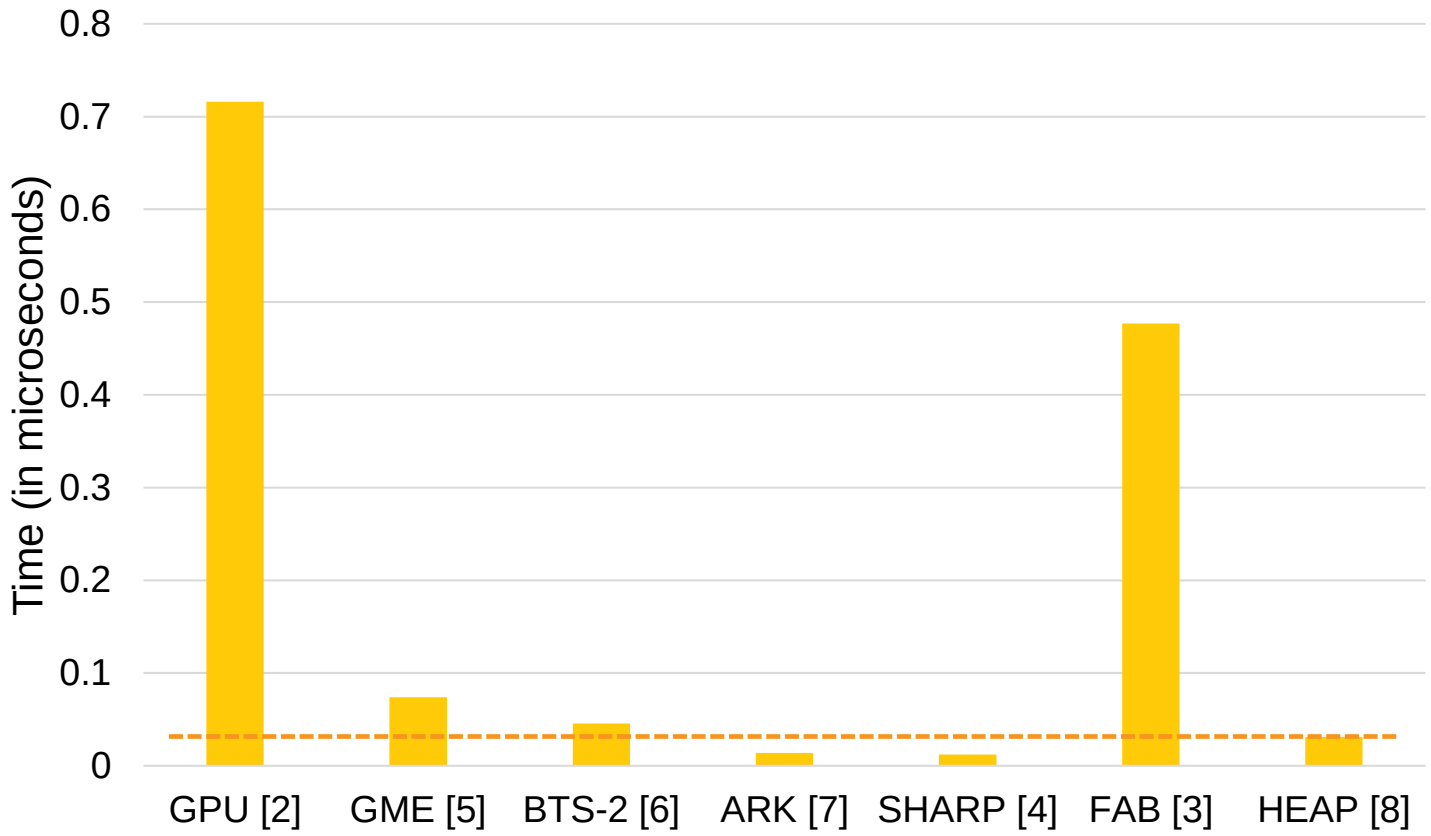
- Scalable bootstrapping accelerator → Multi-FPGA system
  - Uses low-latency tightly coupled functional units with fine-grained pipelining
  - Implements highly optimized NTT and BlindRotate datapath for highly parallel execution



HEAP

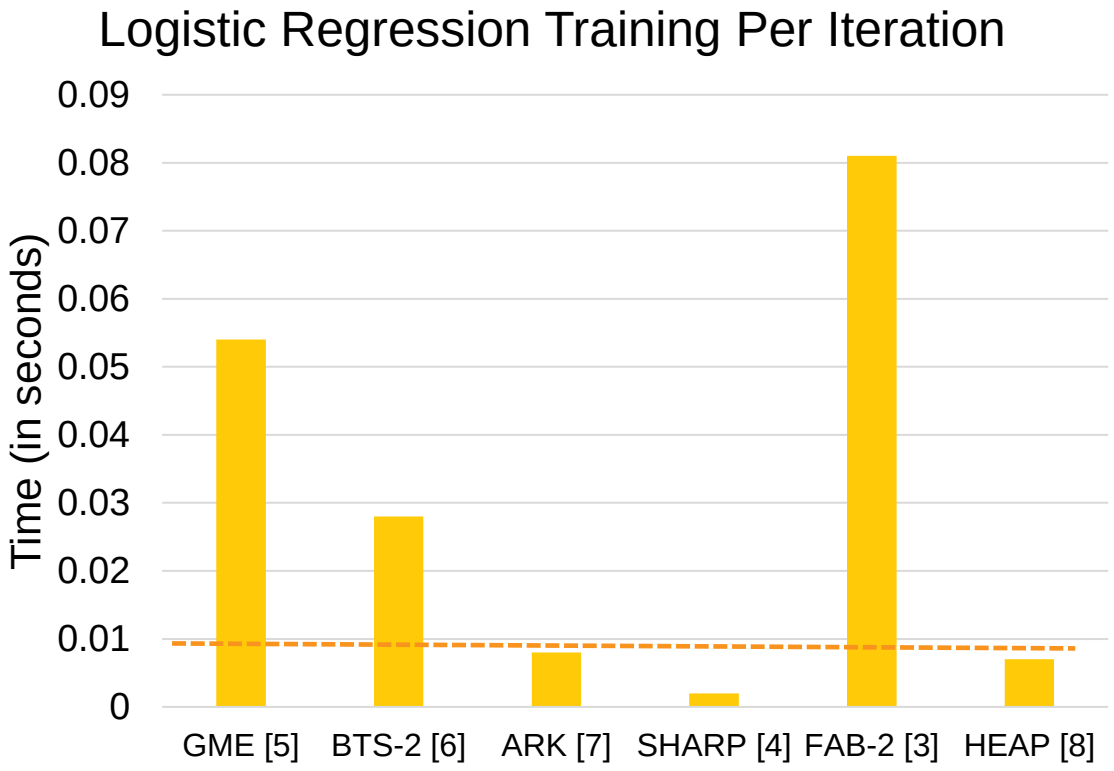
$$T_{\mathbf{Mult},a/\mathsf{slot}} := \frac{T_{\mathsf{BS}} + \sum_{i=1}^{\ell} T_{\mathbf{Mult}}(i)}{\ell \cdot n}$$

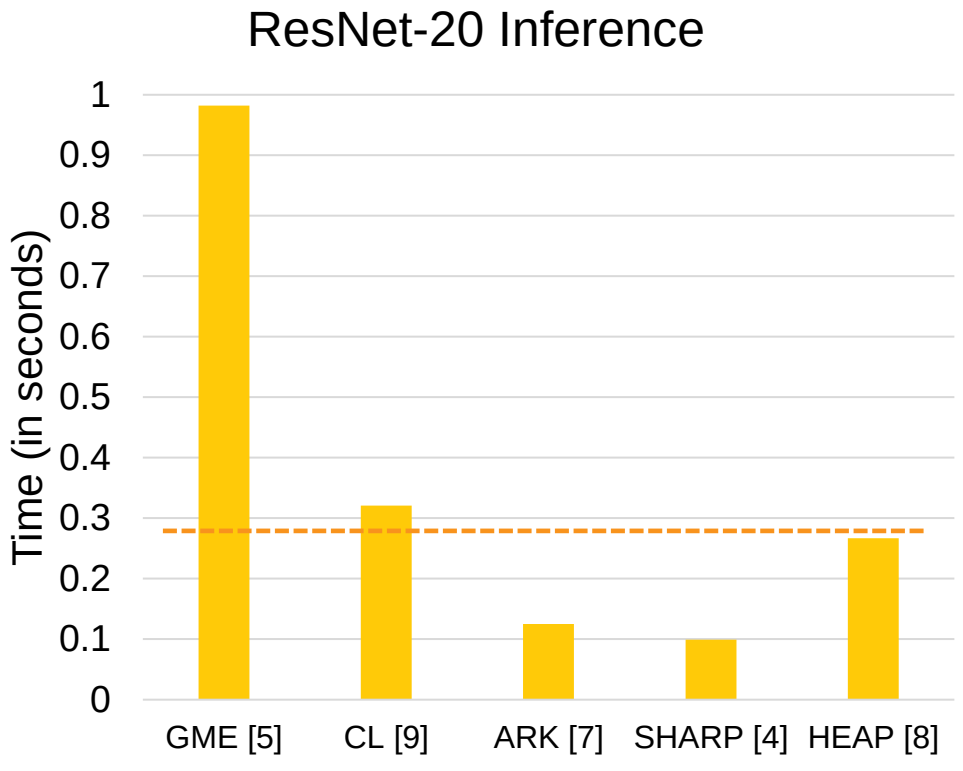
- Bootstrapping performance
  - In terms of cycle count, HEAP is ~4x faster than ARK and has same performance as SHARP



HEAP

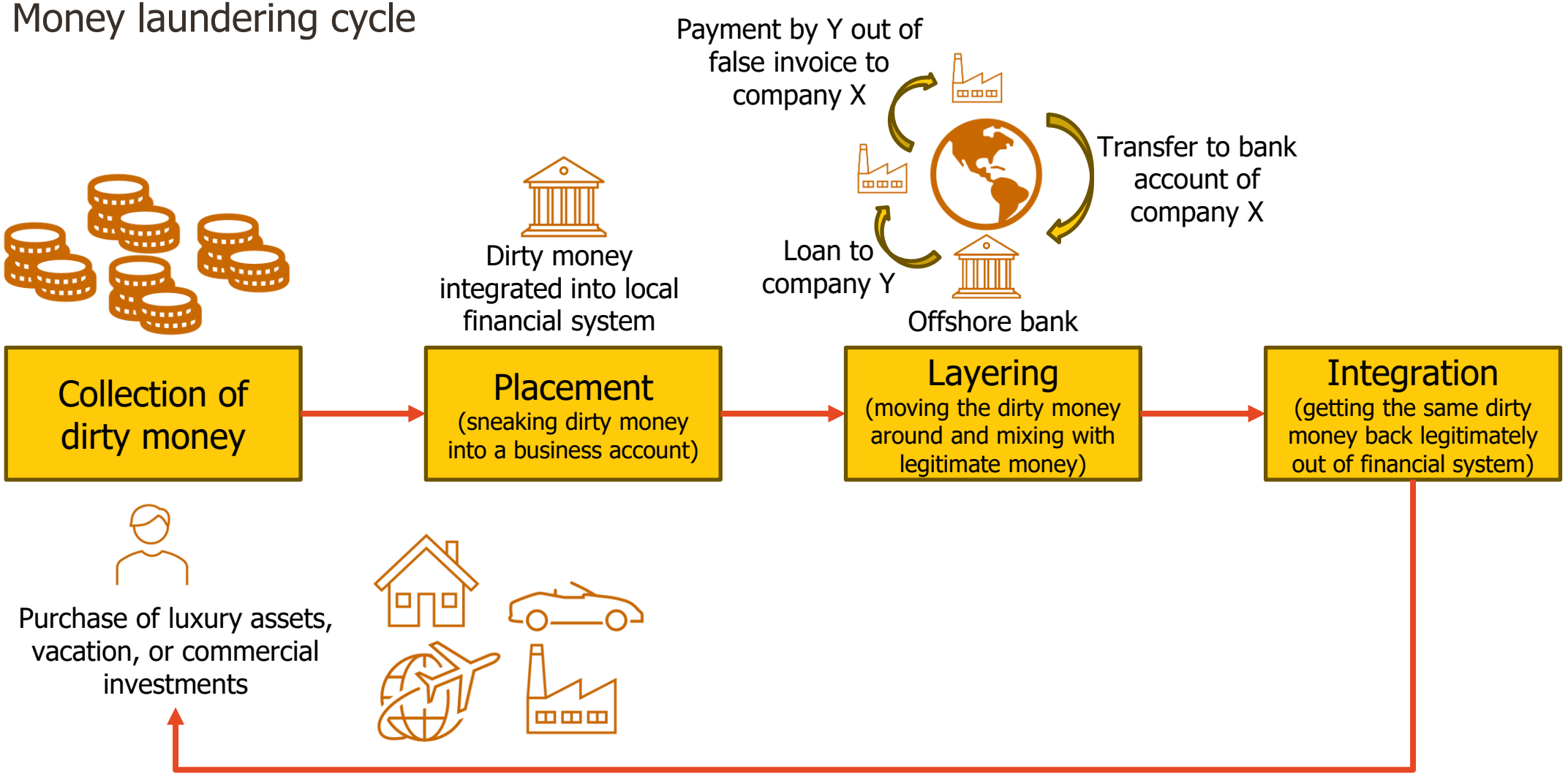
- Application performance
  - In terms of cycle count, HEAP is 1.56x and 1.23x faster than ARK and SHARP, respectively





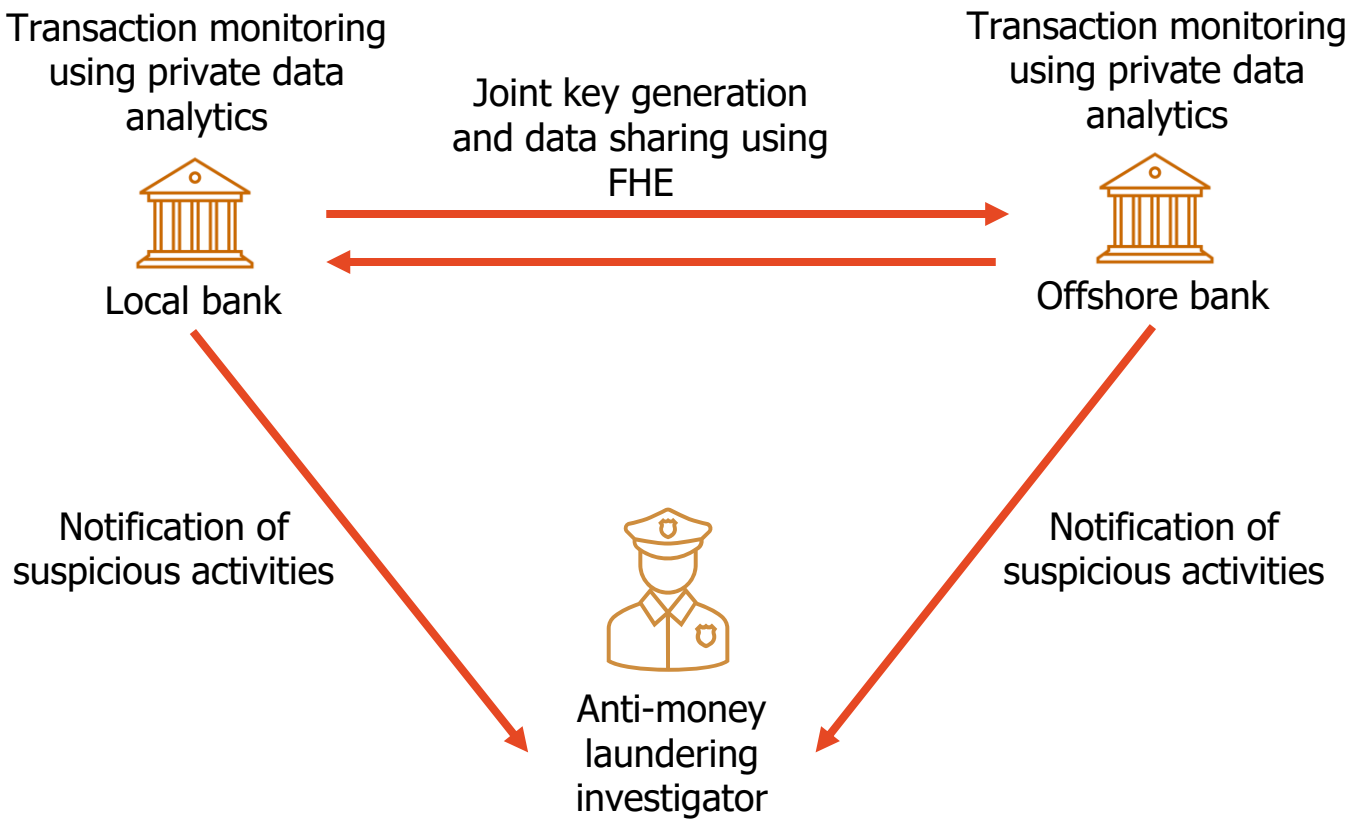
Use Case 1

- Money laundering cycle



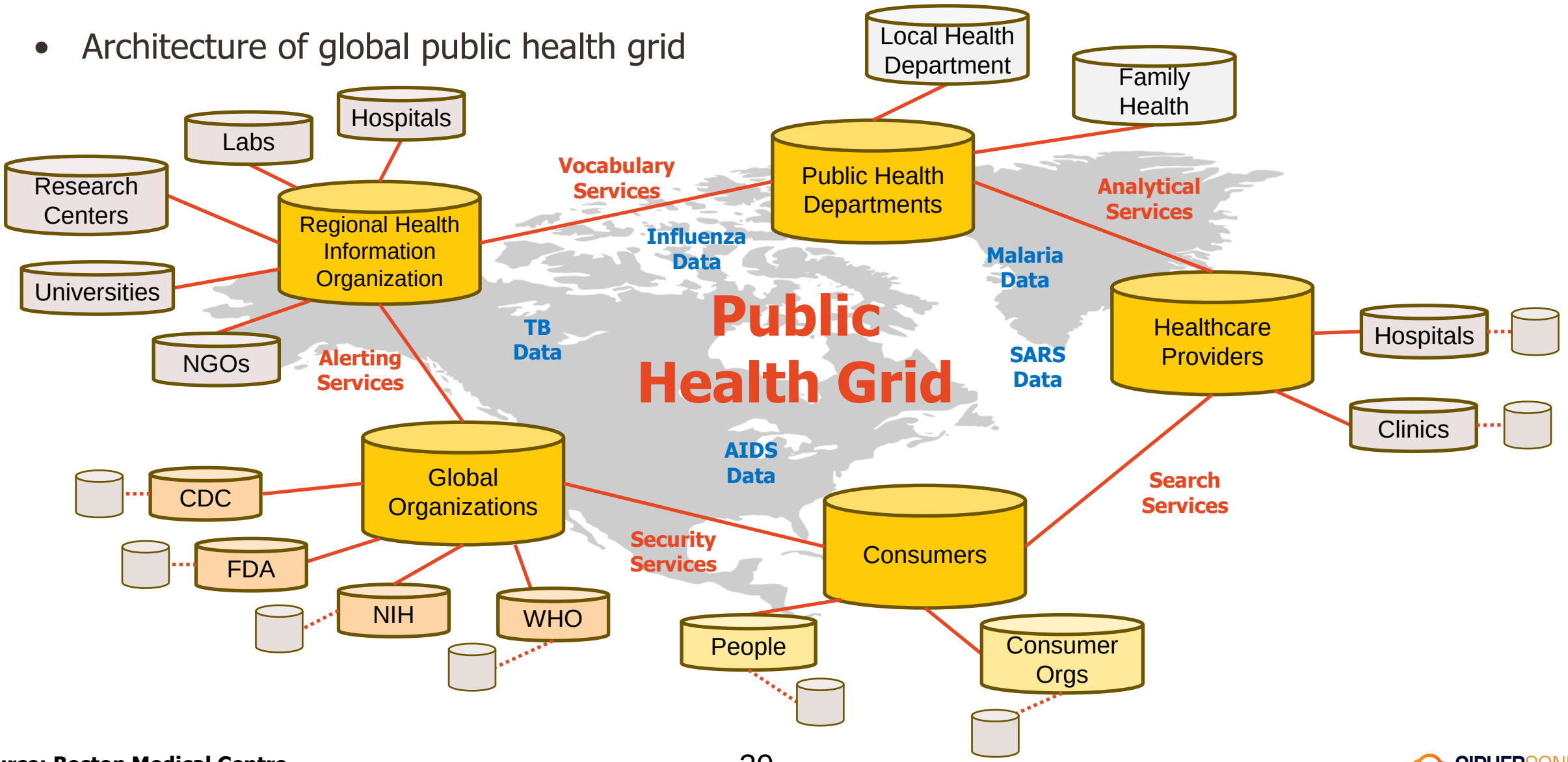
Use Case 1

- Anti-money laundering using FHE



Use Case 2

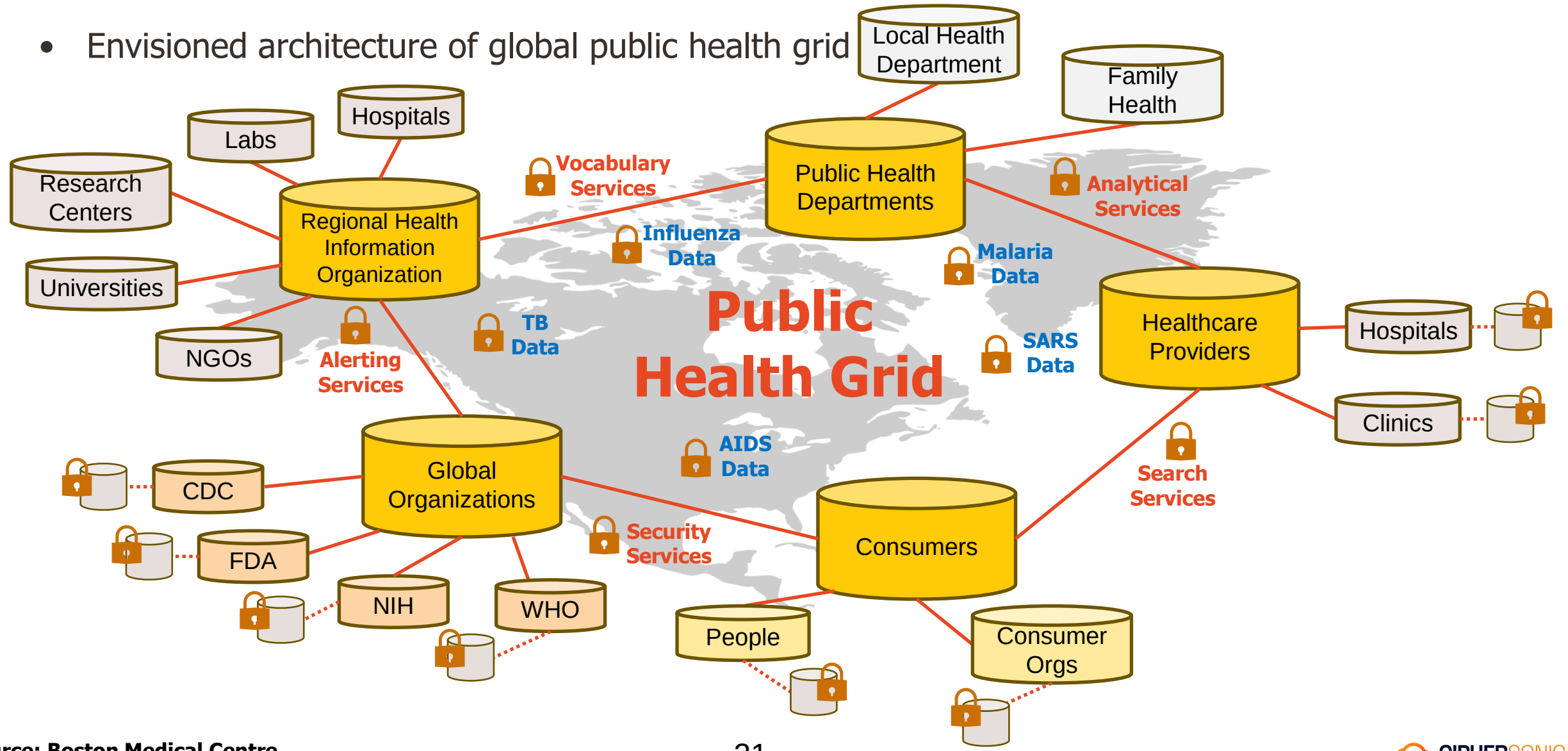
- Architecture of global public health grid



Source: Boston Medical Centre

Use Case 2

- Envisioned architecture of global public health grid



Source: Boston Medical Centre

Summary

- FHE is a plausible way forward for real-world privacy preserving applications
  - Helps prevent against data breaches
  - Helps meet various data regulations and compliance

- FPGAs provide a sweet spot for FHE acceleration
  - Practical performance at a fraction of ASIC cost
  - Can be deployed on existing cloud infrastructure using commercial off-the-shelf hardware

- For widespread FHE adoption,
  - Standardization of FHE algorithms
  - Proven use cases and success stories
  - Awareness is essential

References

[1] Cheon et al., Bootstrapping for Approximate Homomorphic Encryption, ICTACT, 2018

[2] Jung et al., Over 100x Faster Bootstrapping in Fully Homomorphic Encryption, CHES, 2021

[3] Agrawal et al., FAB: An FPGA-based Bootstrappable Fully Homomorphic Encryption Accelerator, HPCA, 2023

[4] Kim et al., SHARP: A Short-word Hierarchical Accelerator for Robust and Practical Fully Homomorphic Encryption , ISCA, 2023

[5] Shivdikar et al., GME: GPU-based Microarchitectural Extensions to Accelerate Homomorphic Encryption, MICRO, 2023

[6] Kim et al., BTS: An Accelerator for Bootstrappable Fully Homomorphic Encryption, ISCA, 2022

[7] Kim et al., ARK: Fully Homomorphic Encryption Accelerator with Runtime Data Generation and Inter-Operation Key Reuse, MICRO, 2022

[8] Agrawal et al., HEAP: A Fully Homomorphic Encryption Accelerator with Parallelized Bootstrapping, ISCA, 2024

[9] Samardzic et al., CraterLake: A Hardware Accelerator for Efficient Unbounded Computation on Encrypted Data, ISCA, 2022