

DECENTRALIZED FHE COMPUTER AND IT'S APPLICATIONS

>

Gurgen Arakelov

Presented at NIST WPEC 2024 on September 25

>AGENDA

—

> 1:

INTRO TO FHE

> 2:

CHALLENGES IN FHE FOR DEVELOPERS

> 3:

FHE COMPONENT BASED APPROACH

> 4:

FHE COMPUTER: OVERVIEW

> 5:

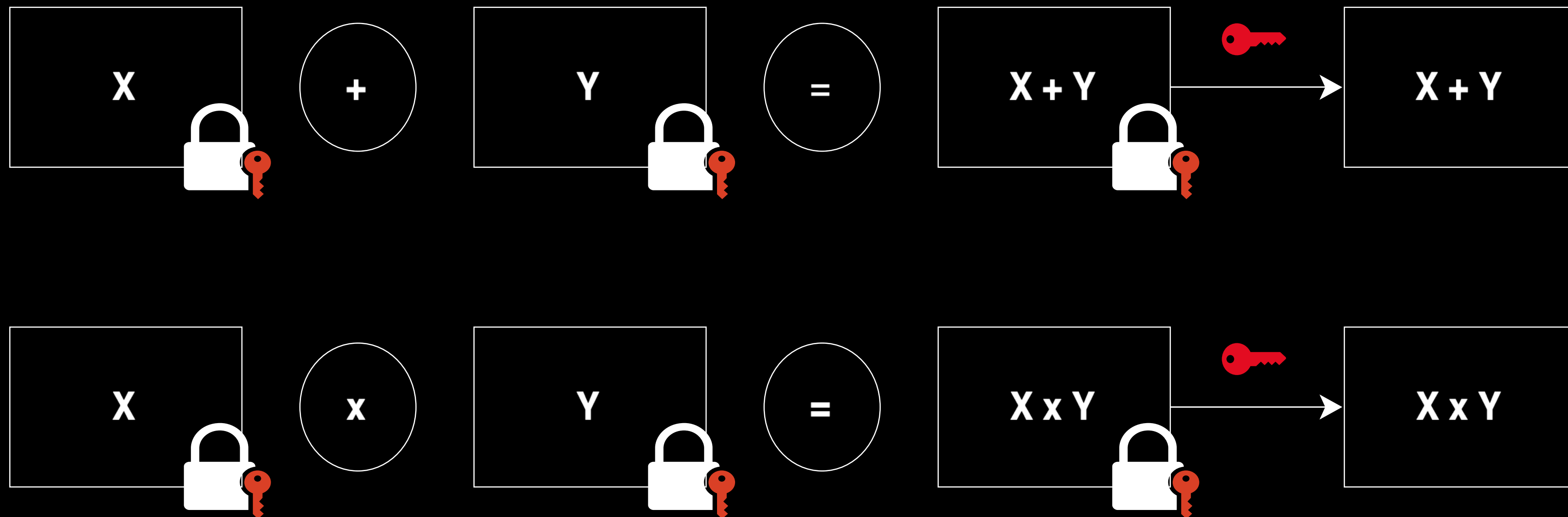
USECASE: VECTOR DATABASE

FHE

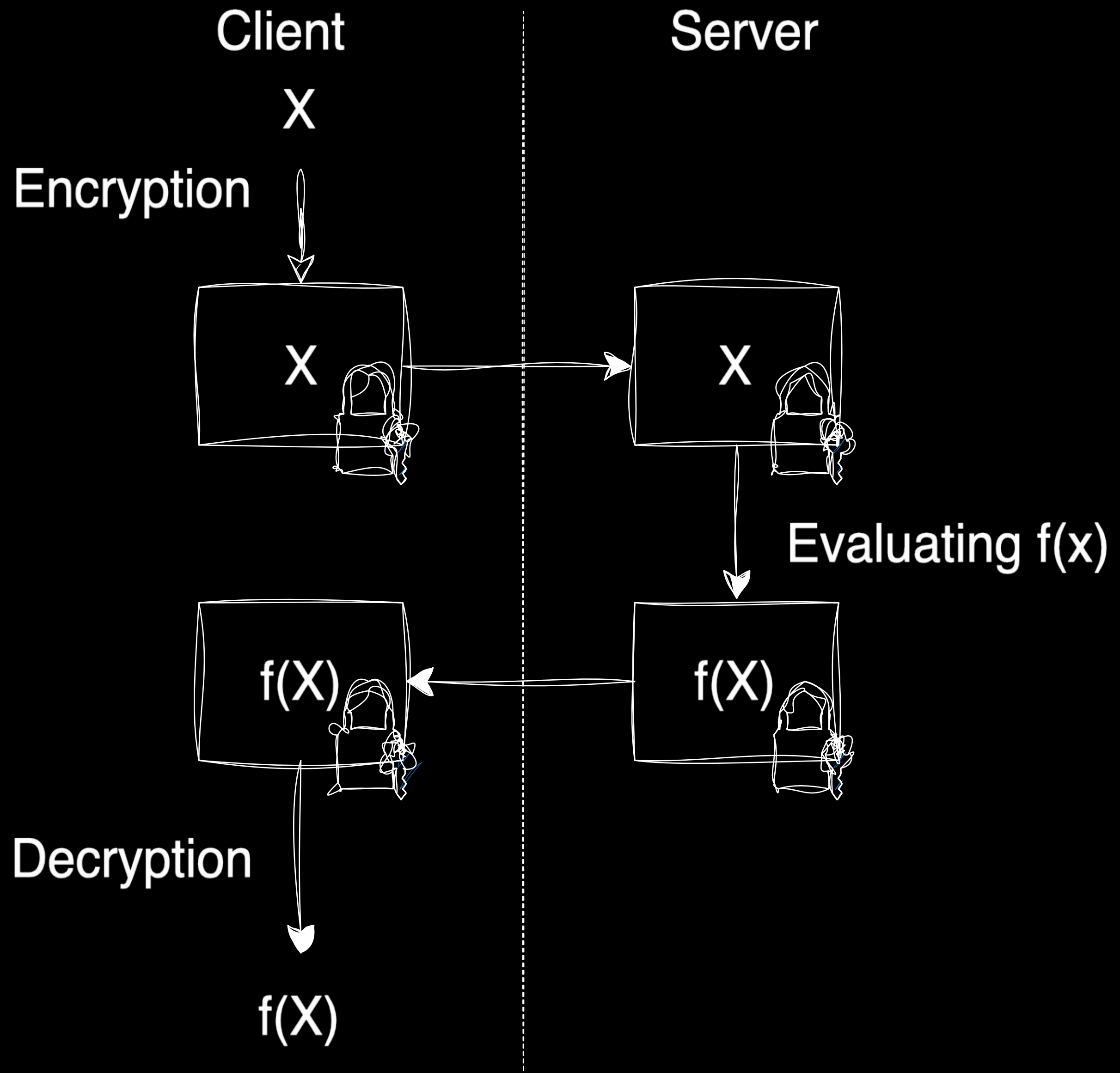
>Fully Homomorphic Encryption (FHE)

>How it works

—



>FHE
>How it works



>FHE Landscape

>Schemes

Schemes	Data types	Advantages	Limitations
1. BGV/BFV	Modular arithmetic operations over finite fields	Great for vectors of relatively small integers and finite fields	Slow Bootstrapping
2. FHEW/TFHE	Boolean circuits	Fast bootstrapping / arbitrary depth of computation	Scalability
3. CKKS	Approximate computations over vectors of real and complex numbers	The best choice for many ML applications	Slow Bootstrapping / Approximate result
...

>FHE Landscape

>Libraries

Library	Schemes	Languages	License
OpenFHE	BFV, BGV, CKKS, DM/FHEW, CGGI/TFHE, LMKCDEY	C/C++, Python, Rust	BSD 2-Clause
Lattigo	BFV, BGV, CKKS, LMKCDEY	Go	Apache-2.0
tfhe-rs	CGGI/TFHE	rust	BSD 3-Clause Clear License
SEAL	BFV, BGV, CKKS	C++/Python	MIT License
...

Challenges for Developers

>FHE

>Issues for developers.

—

1. Limited basis of operations

We can perform a limited number of operations on encrypted data and the result remains encrypted.

>FHE

>Issues for developers.

—

1. Limited basis of operations

We can perform a limited number of operations on encrypted data and the result remains encrypted.

2. Performance

FHE is still a performance-intensive technology.

>FHE

>Issues for developers.

1. Limited basis of operations

We can perform a limited number of operations on encrypted data and the result remains encrypted.

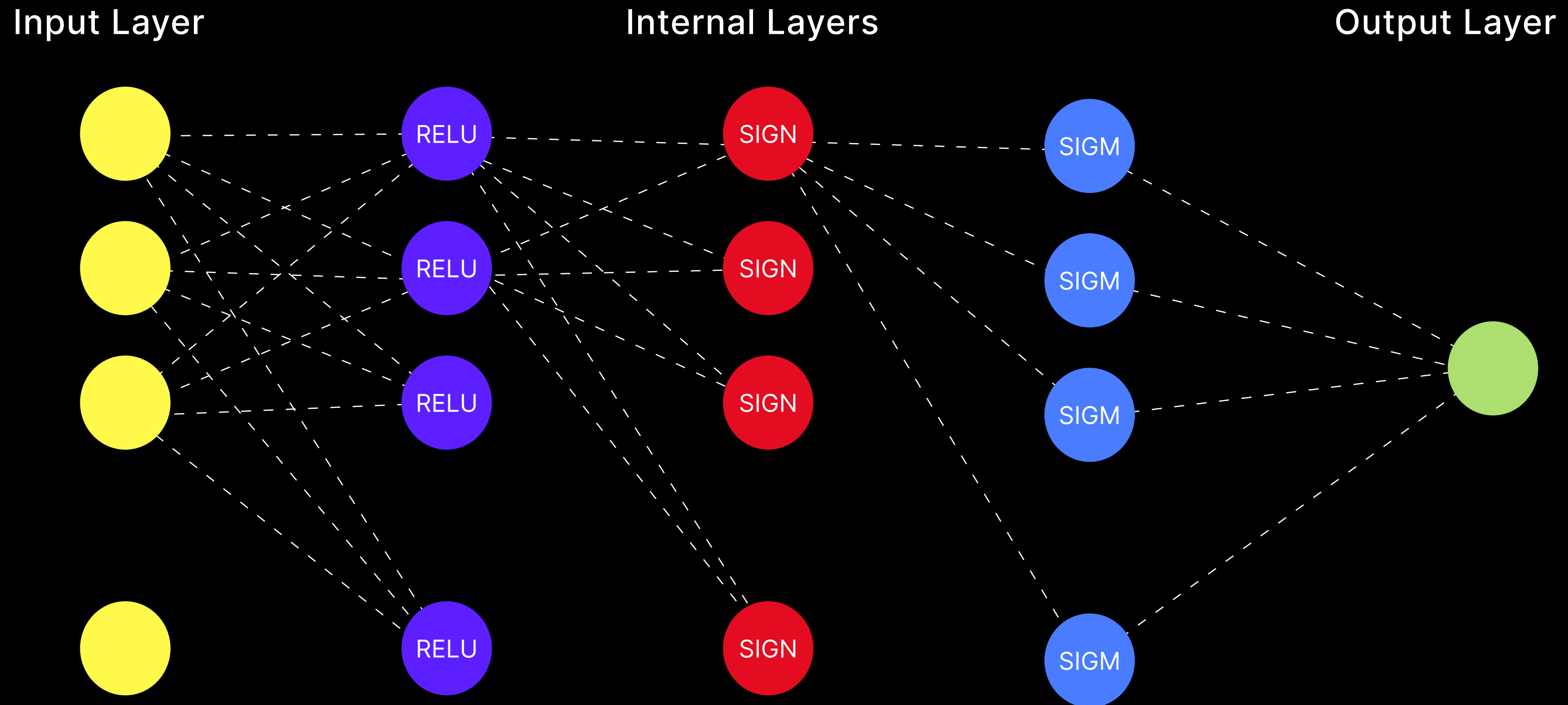
2. Performance

FHE is still a performance-intensive technology.

3. Lack of Developers ecosystem

Existing tools and libraries either require expertise in FHE or have limited functionality.

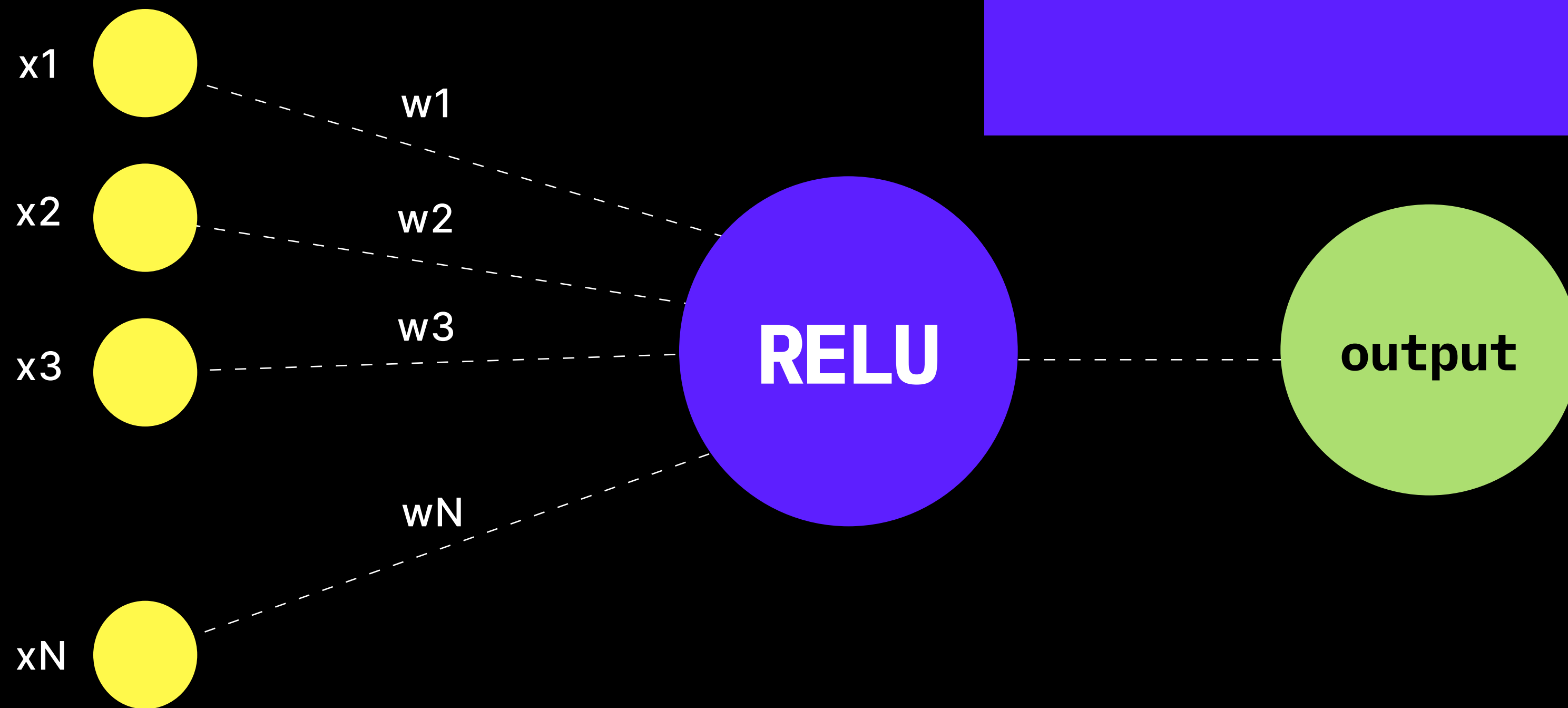
> Privacy Preserving Neural Network



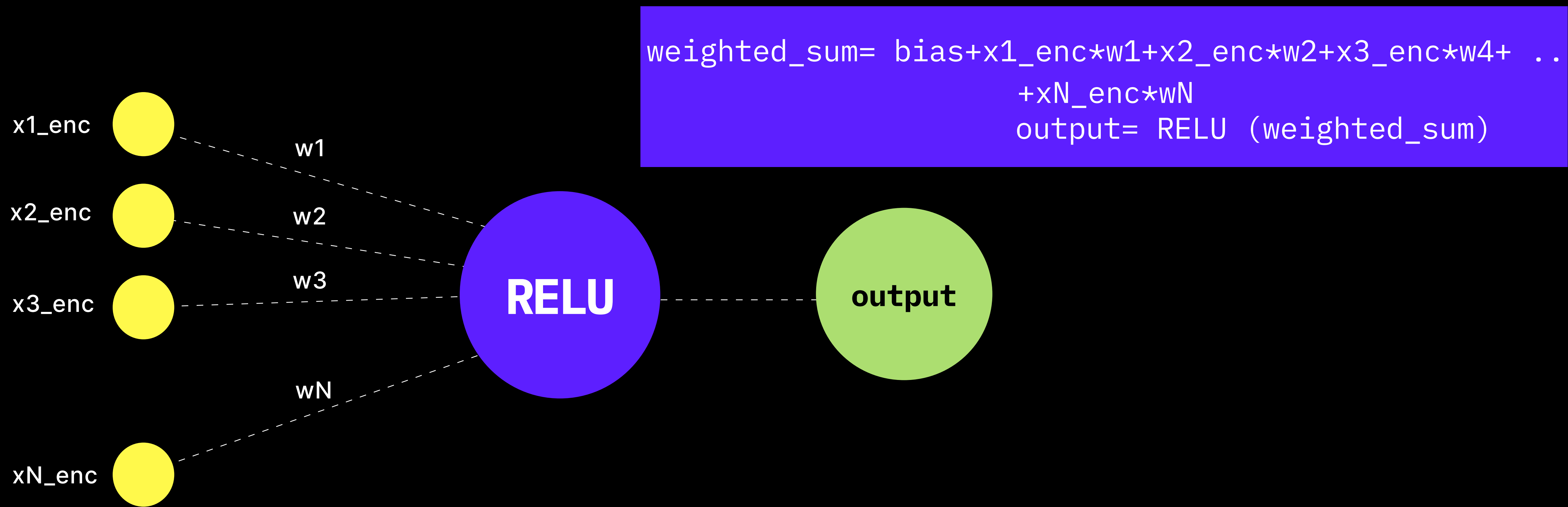
> Privacy Preserving Neural Network

$$\text{weighted_sum} = \text{bias} + x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \dots + x_N * w_N$$

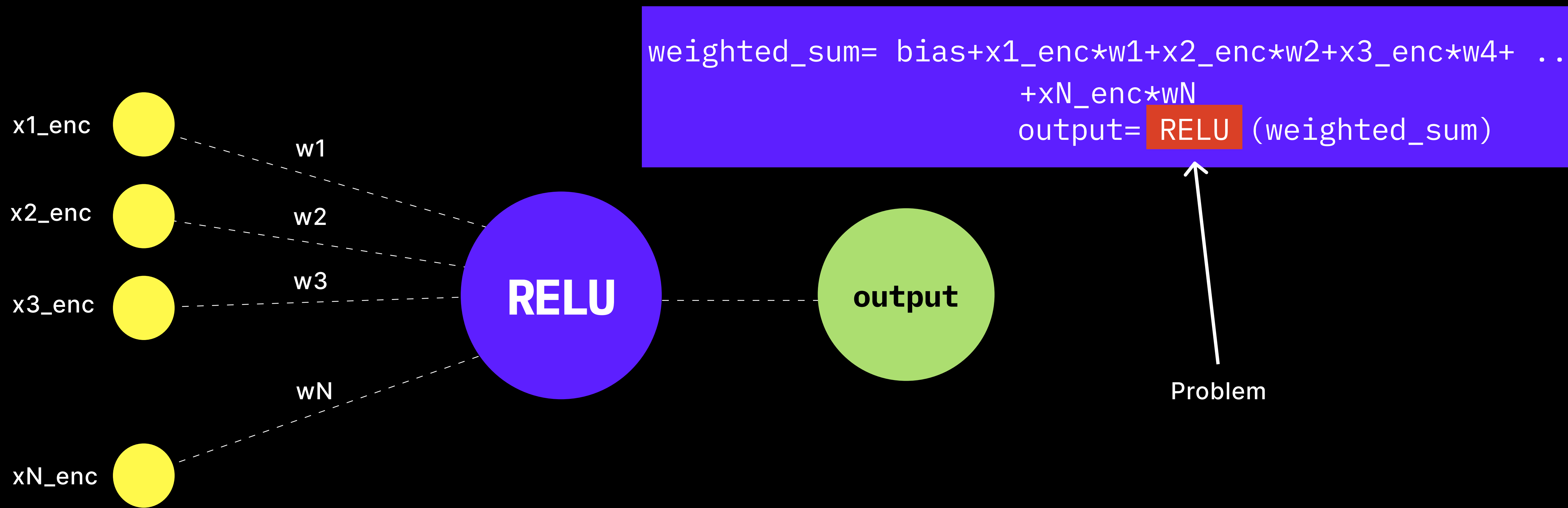
$$\text{output} = \text{RELU}(\text{weighted_sum})$$



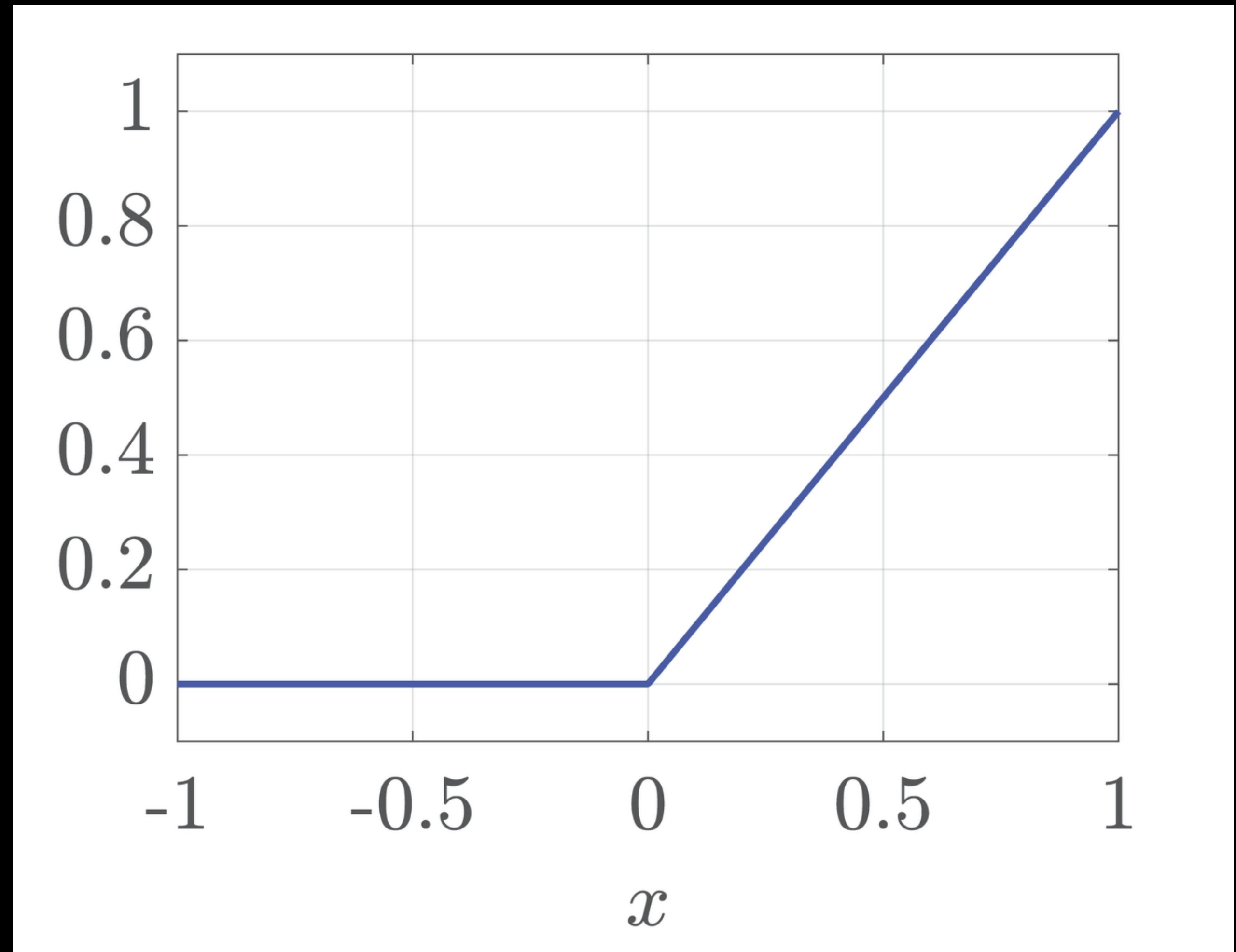
> Privacy Preserving Neural Network



> Privacy Preserving Neural Network

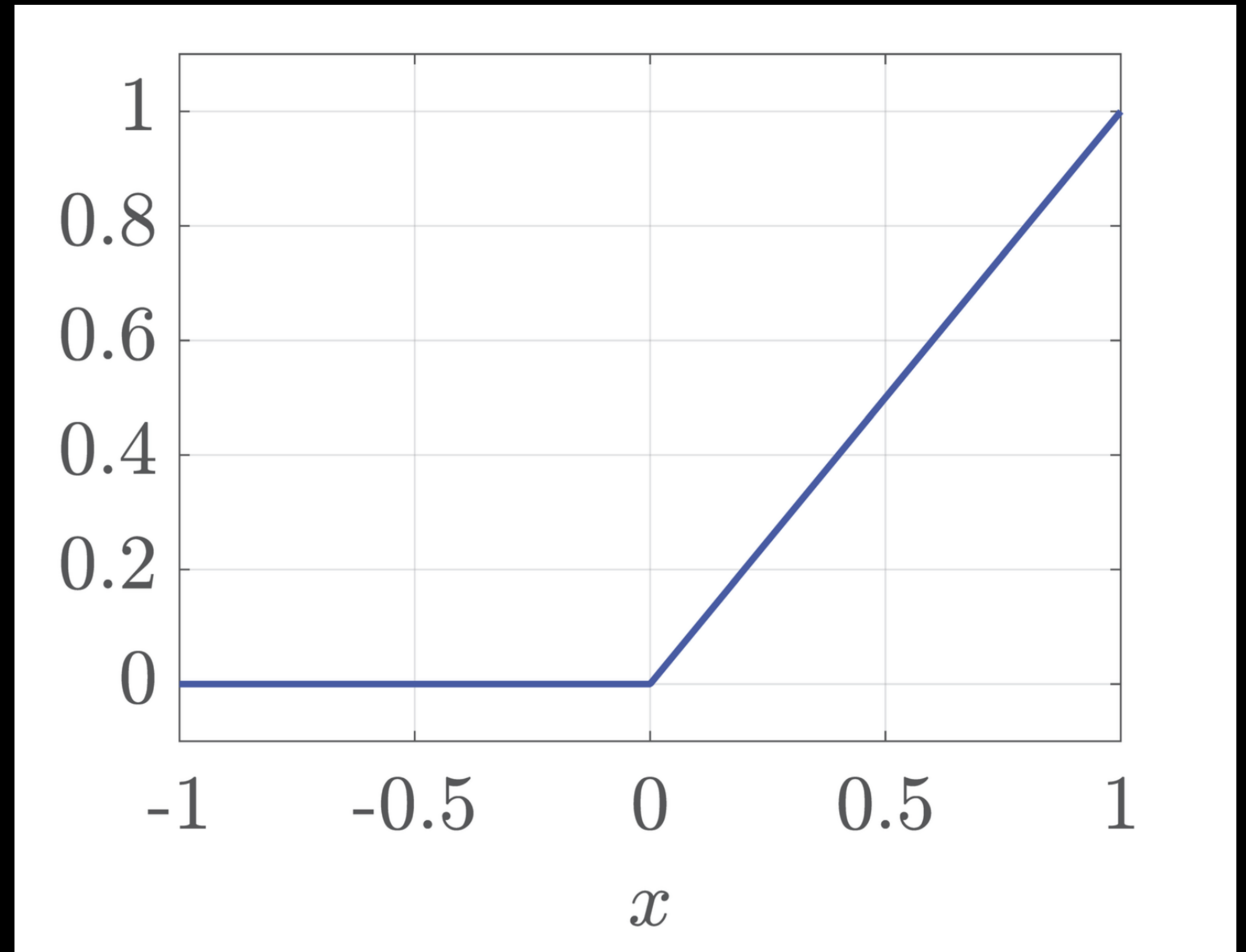
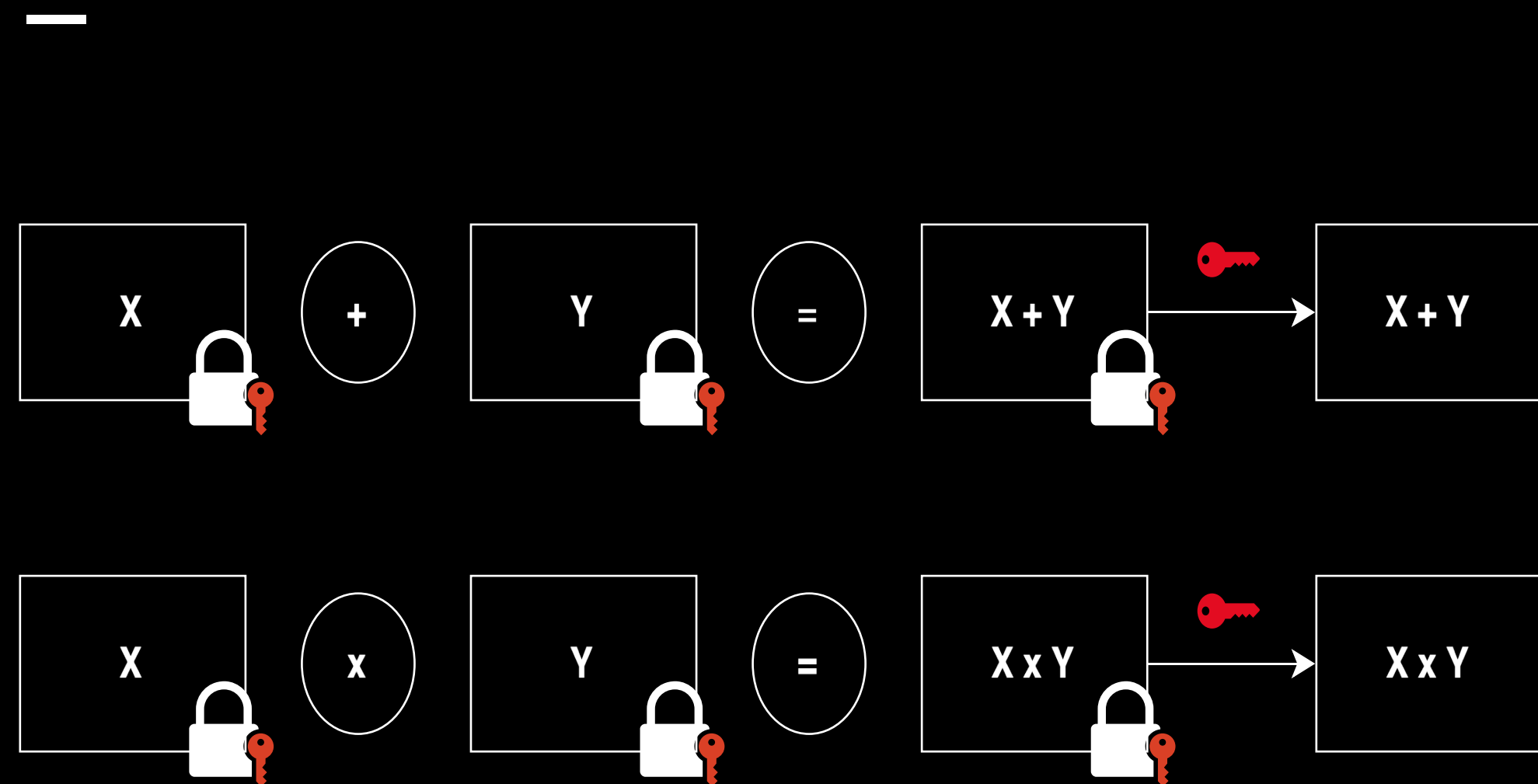


>Privacy Preserving Neural Network



> Privacy Preserving Neural Network

>How it works



>Privacy Preserving Neural Network
>Basic opportunities

—

1. Polynomial
Approximation

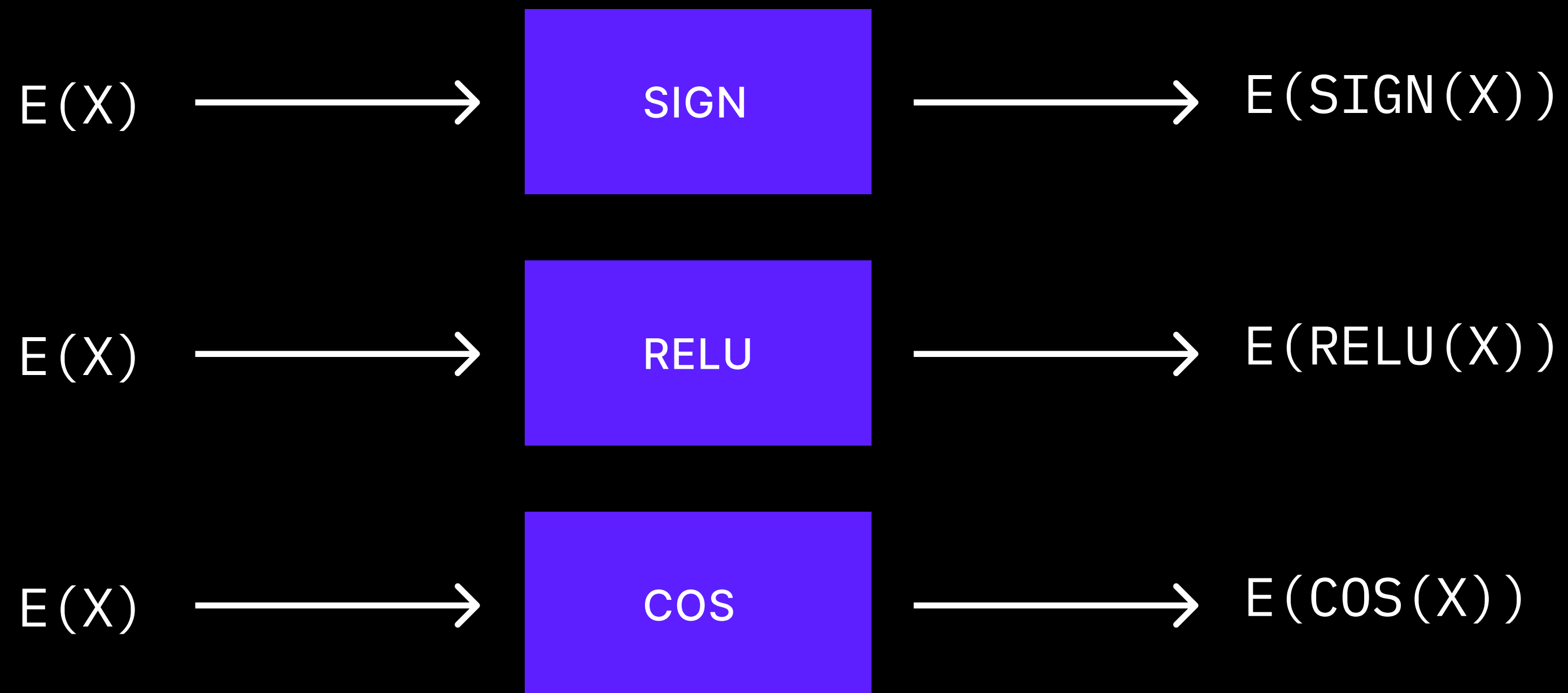
2. Play with
packing

3. Functional
Bootstrapping (PBS)

Component Based FHE Apps

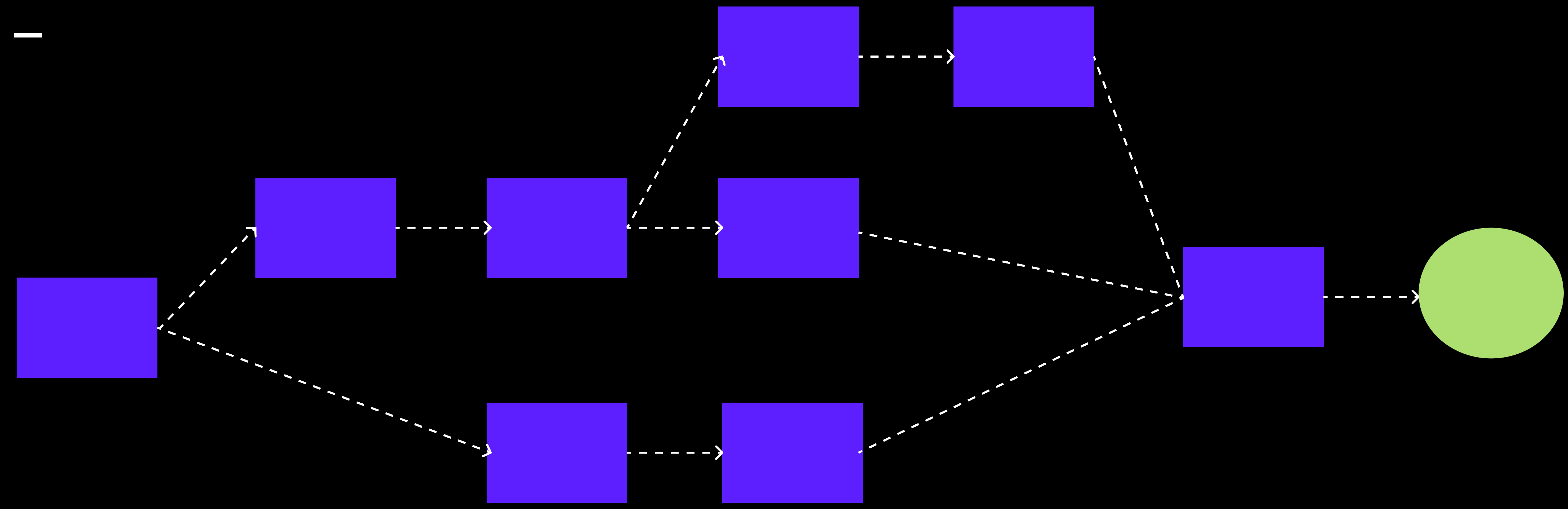
>FHE Components

>Examples



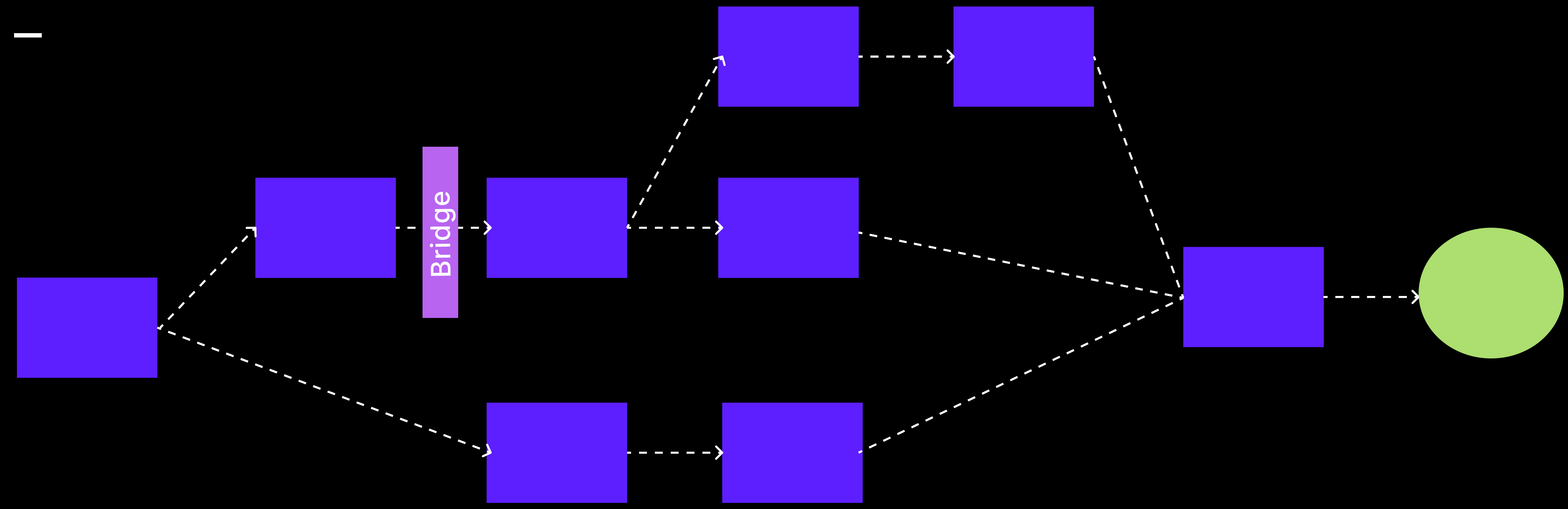
>FHE Components

>Examples



>FHE Components

>Examples



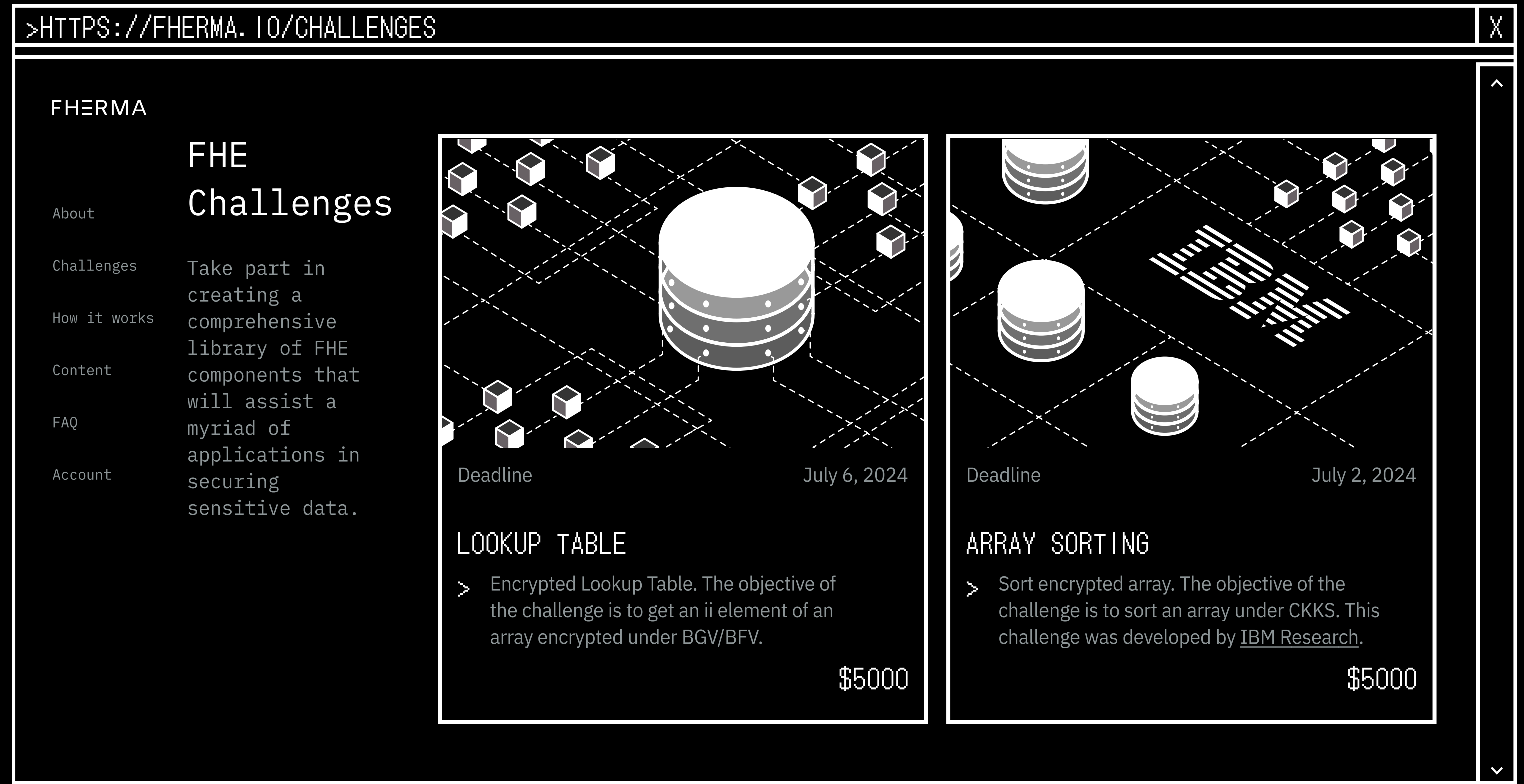
>fherma.io: >FHERMA FHE Challenges

FHERMA is a joint project by Fair Math and OpenFHE teams.

> Fully automated and transparent

> Black Box & White Box Challenges

> Library agnostic



The screenshot shows the website interface for FHERMA FHE Challenges. It includes a navigation menu on the left with links for About, Challenges, How it works, Content, FAQ, and Account. The main content area features a title 'FHE Challenges' and a description: 'Take part in creating a comprehensive library of FHE components that will assist a myriad of applications in securing sensitive data.' Below this, two challenge cards are displayed. The first card is for 'LOOKUP TABLE' with a deadline of July 6, 2024, a \$5000 prize, and a description: 'Encrypted Lookup Table. The objective of the challenge is to get an i element of an array encrypted under BGV/BFV.' The second card is for 'ARRAY SORTING' with a deadline of July 2, 2024, a \$5000 prize, and a description: 'Sort encrypted array. The objective of the challenge is to sort an array under CKKS. This challenge was developed by IBM Research.'

>POLY CIRCUIT

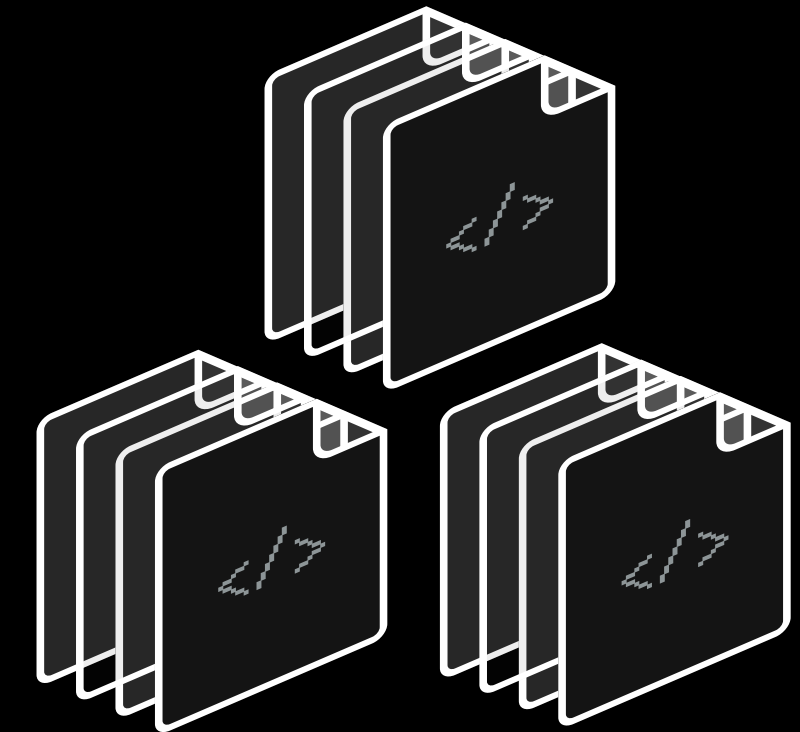
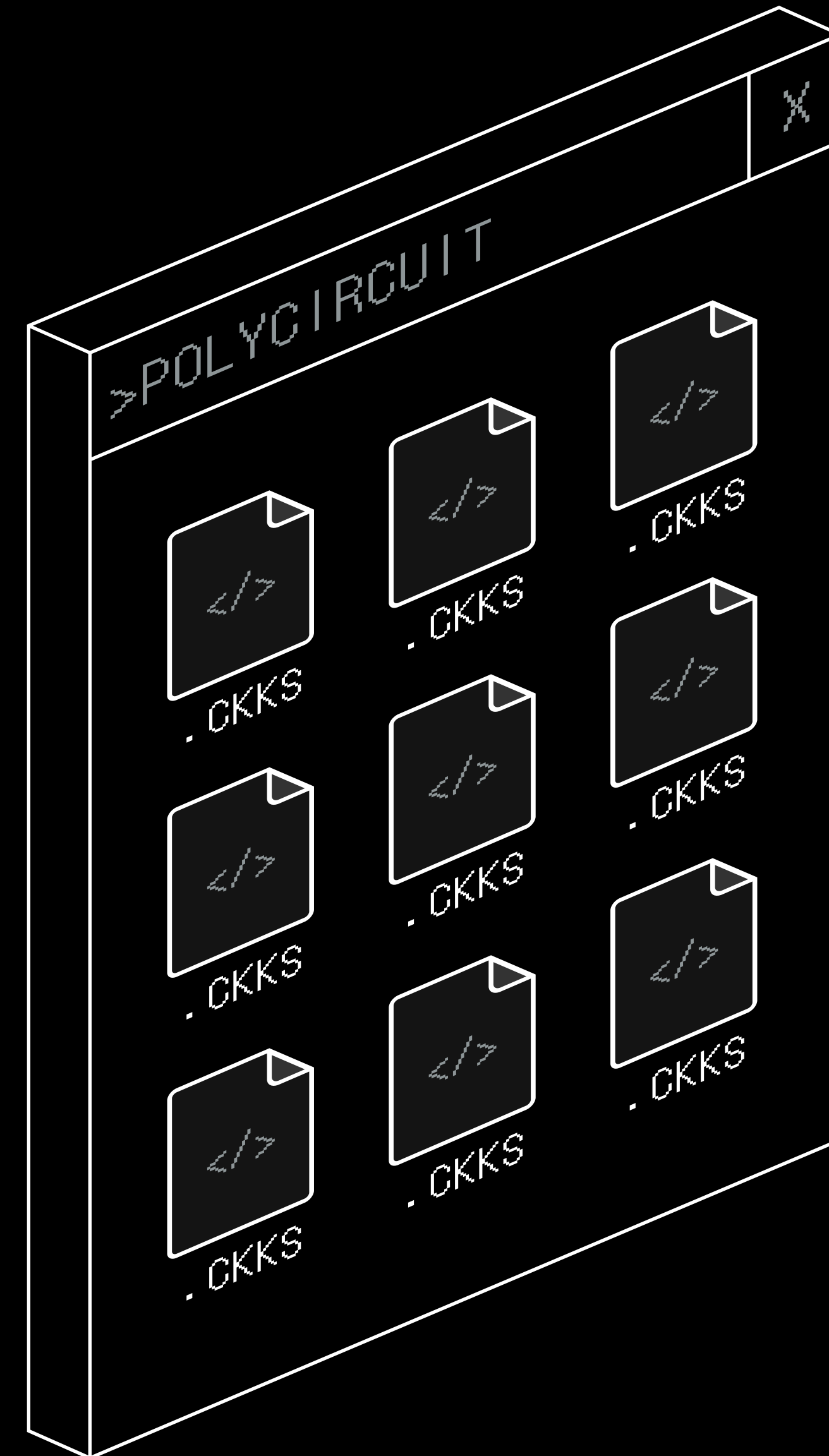
>FHE Components

> Polycircuit is an open-source platform for storing and distributing FHE Components

> Support and Community Engagement

> Wide range of validated and optimised functionality

> Storage and Distribution platform for generalized FHE components



> FHERMA Whitepaper

FHERMA: Building the Open-Source FHE Components Library for Practical Use

Gurgen Arakelov¹, Nikita Kaskov¹, Daria Pinykh¹ and Yuriy Polyakov²

¹ Fair Math, {gurgen,nikita,daria}@fairmath.xyz

² Duality Technologies, ypolyakov@dualitytech.com

Abstract. Fully Homomorphic Encryption (FHE) is a powerful Privacy-Enhancing Technology (PET) that enables computations on encrypted data without having access to the secret key. While FHE holds immense potential for enhancing data privacy and security, creating its practical applications is associated with many difficulties. A significant barrier is the absence of easy-to-use, standardized components that developers can utilize as foundational building blocks. Addressing this gap requires constructing a comprehensive library of FHE components, a complex endeavor due to multiple inherent problems. We propose a competition-based approach for building such a library. More concretely, we present FHERMA, a new challenge platform that introduces black-box and white-box challenges, and fully automated evaluation of submitted FHE solutions. The initial challenges on the FHERMA platform are motivated by practical problems in machine learning and blockchain. The winning solutions get integrated into an open-source library of FHE components, which is available to all members of the PETs community under the Apache 2.0 license.

Keywords: fully homomorphic encryption, privacy-enhancing technologies, challenges, cryptography, privacy

SHIFT
BFV

L1
CKKS

L2
TFHE

RELU
CKKS

MAX
TFHE

ROTATE
CKKS

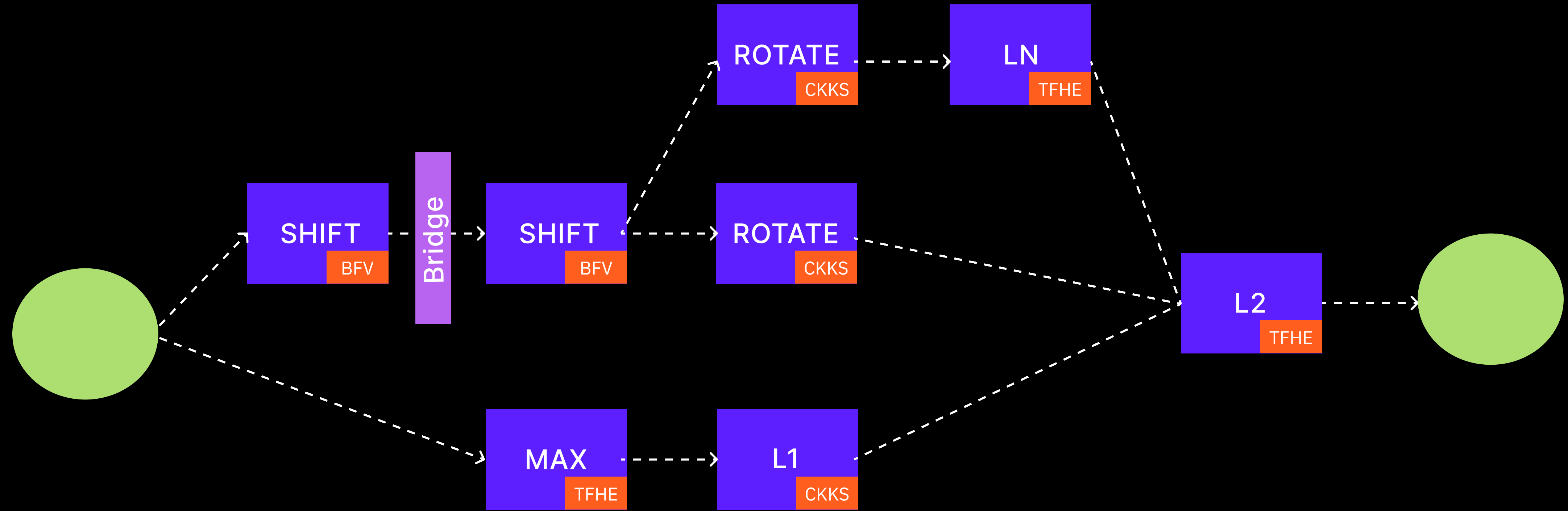
LN
TFHE

SIGNUM
BGV

ROTATE
CKKS

SORT
BGV

MIN
CKKS

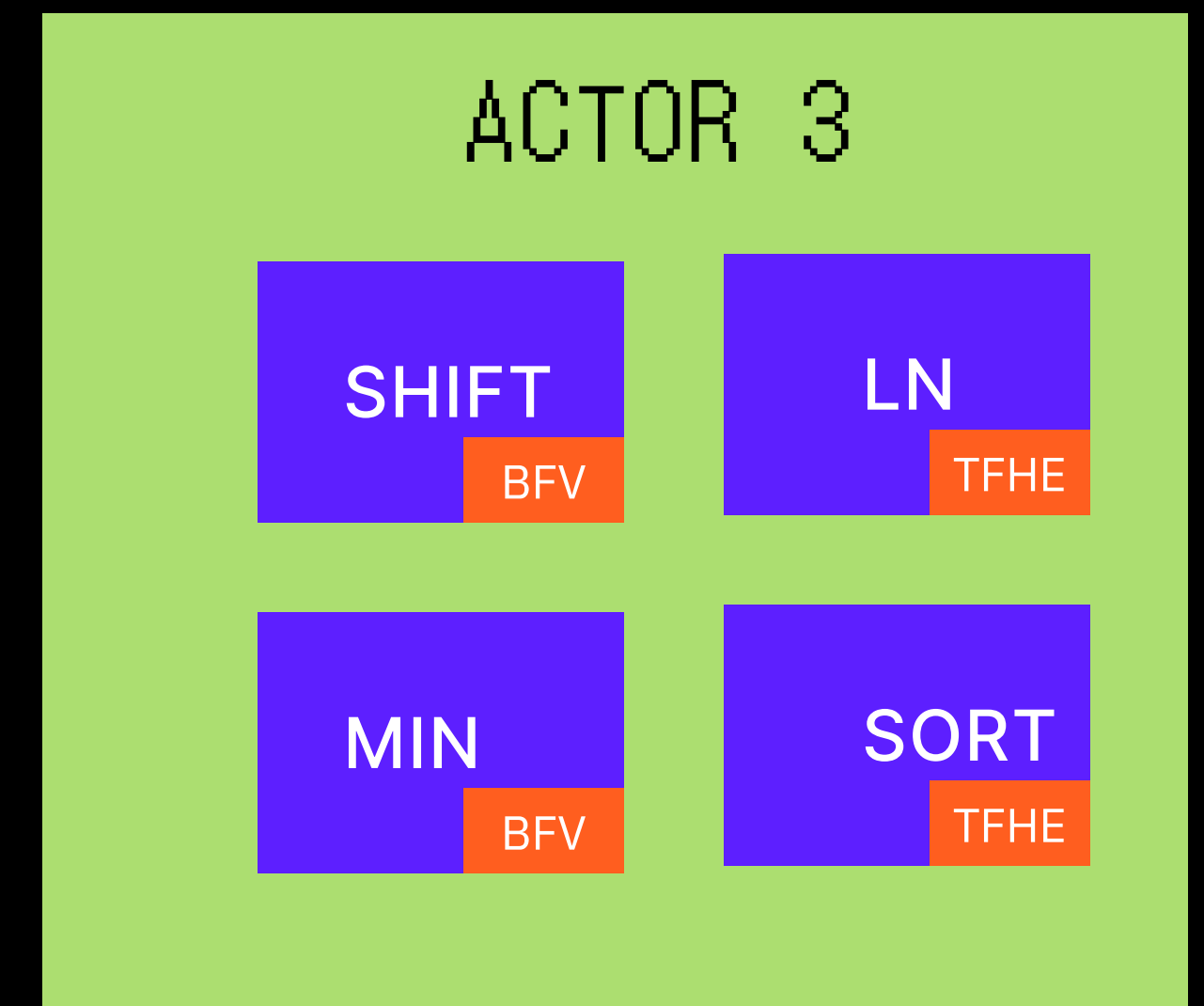
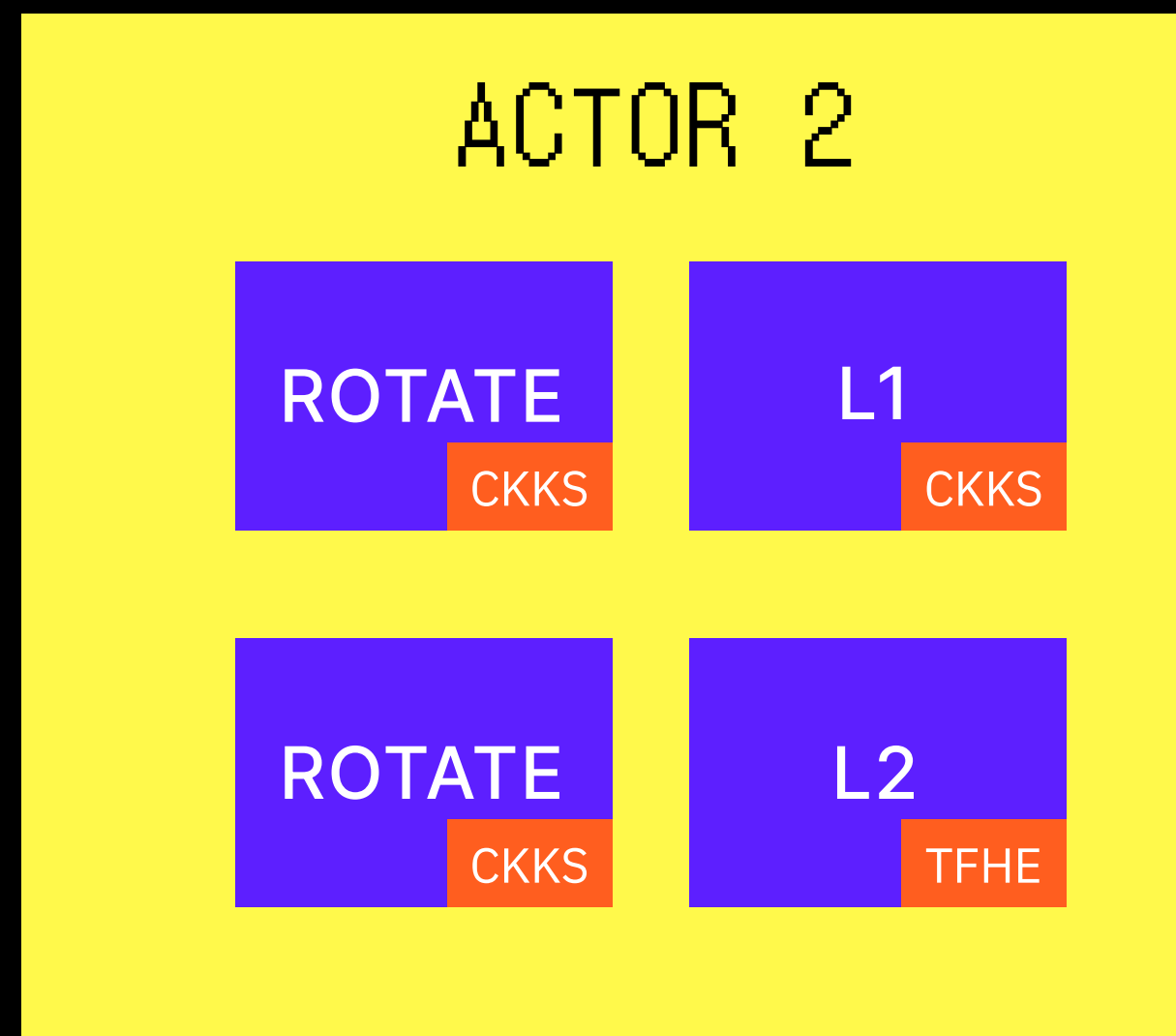
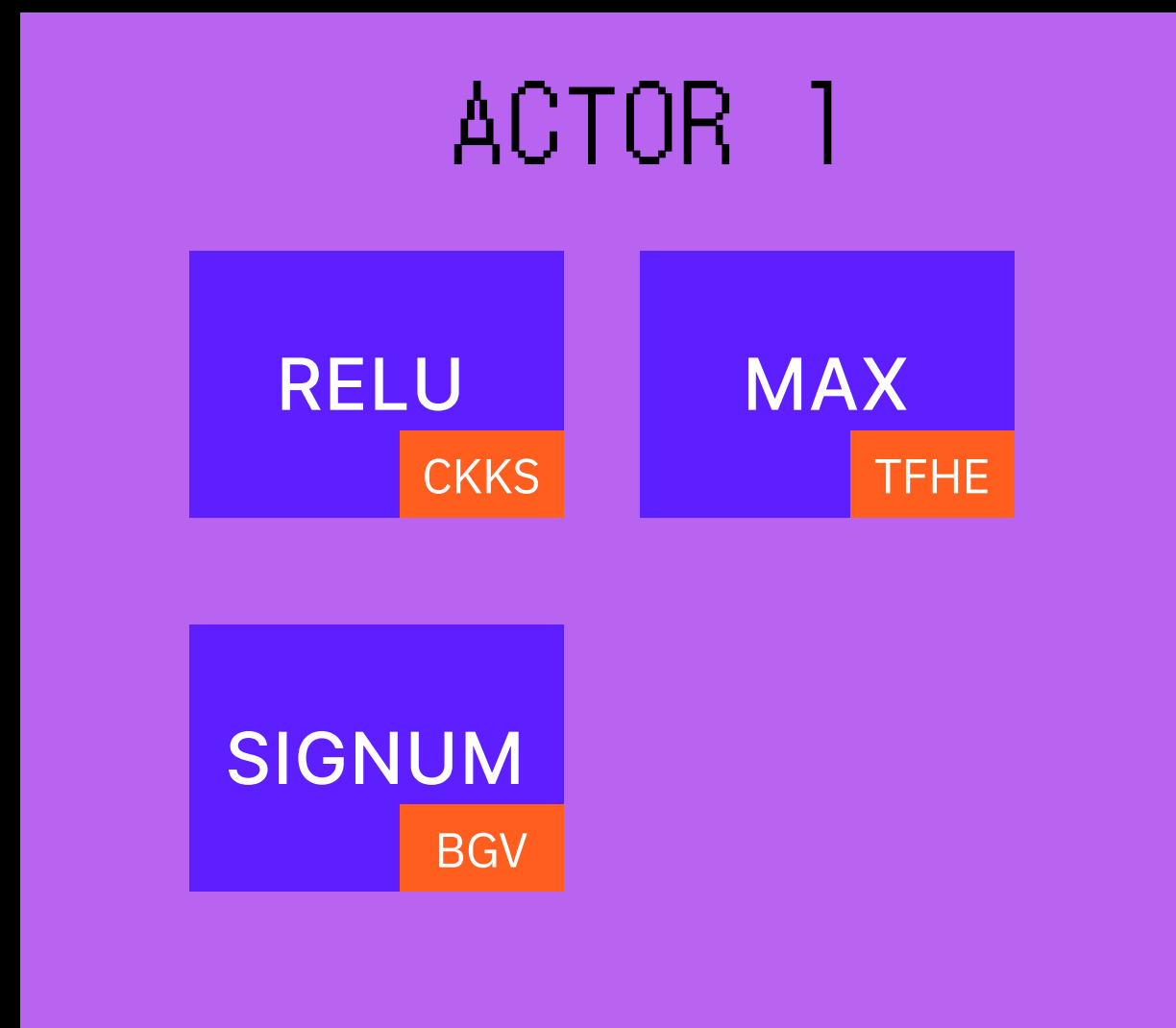


HOW TO DEPLOY AND RUN?

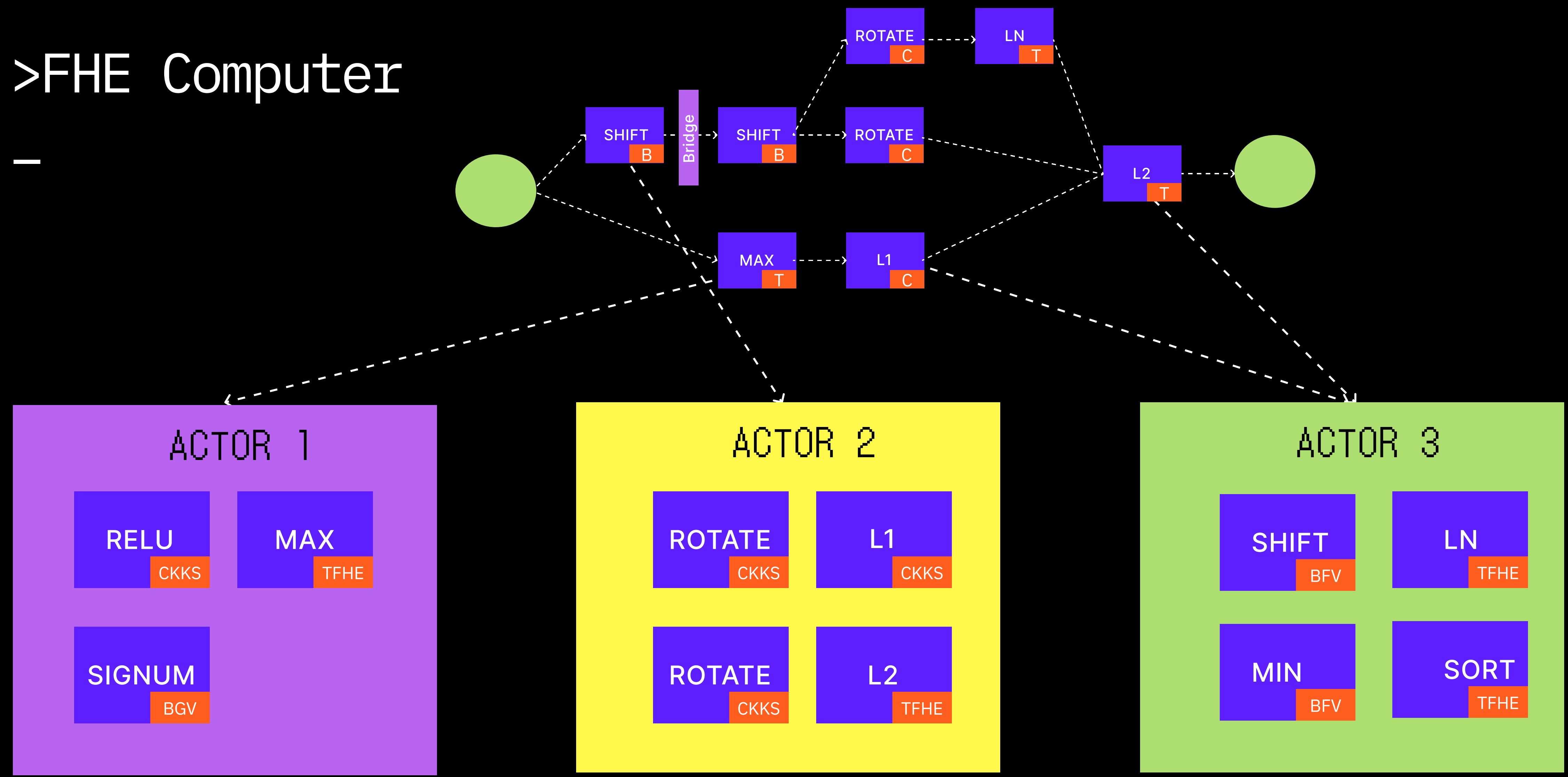
FHE Computer

>FHE Computer

—

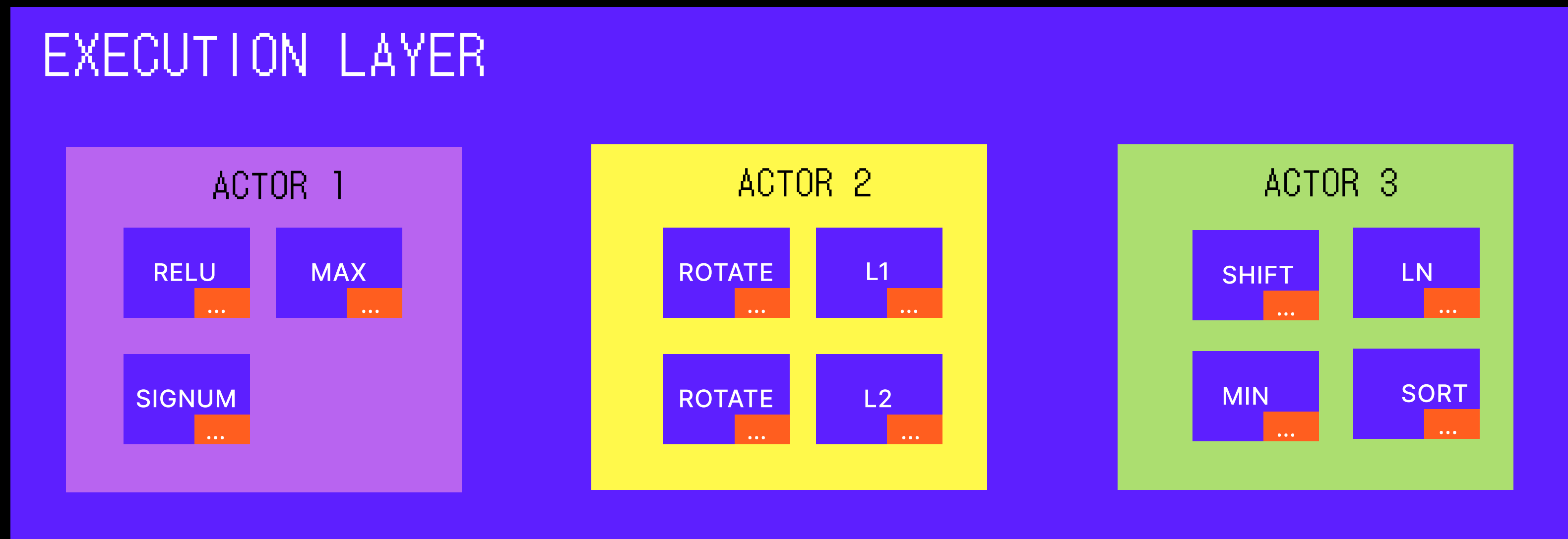


>FHE Computer



>FHE Computer

—

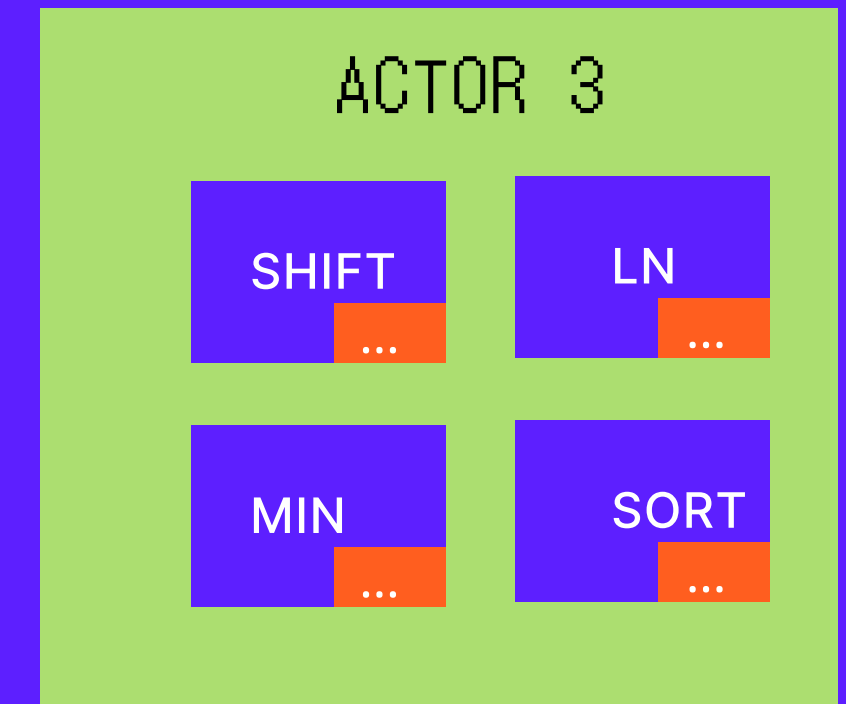
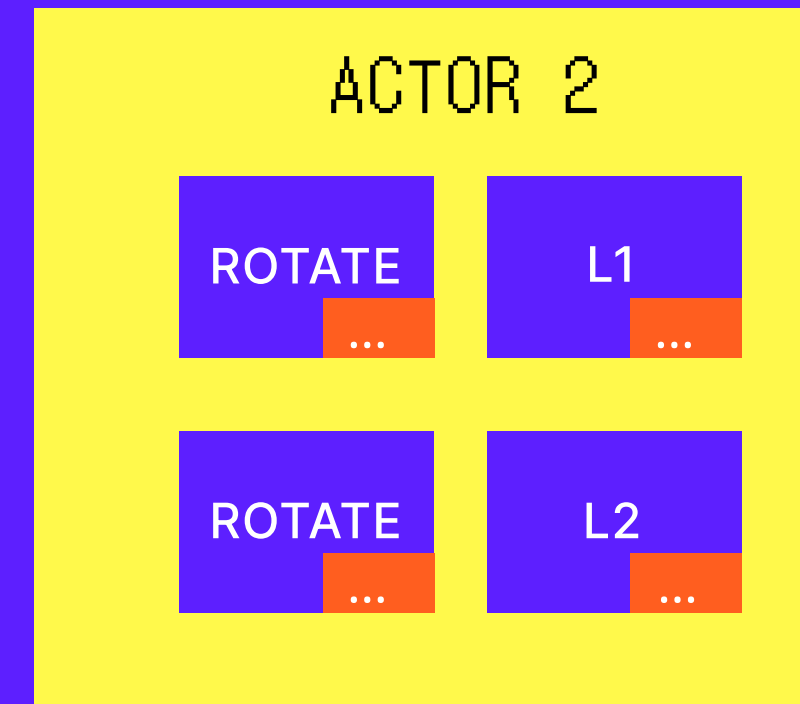
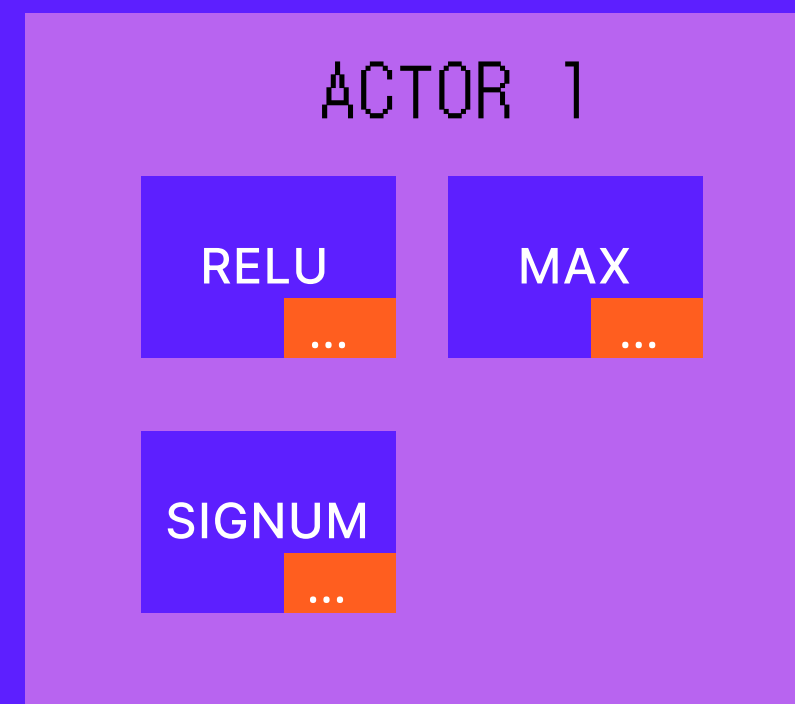


>FHE Computer

—

APPLICATION LAYER: FHE APPS

EXECUTION LAYER



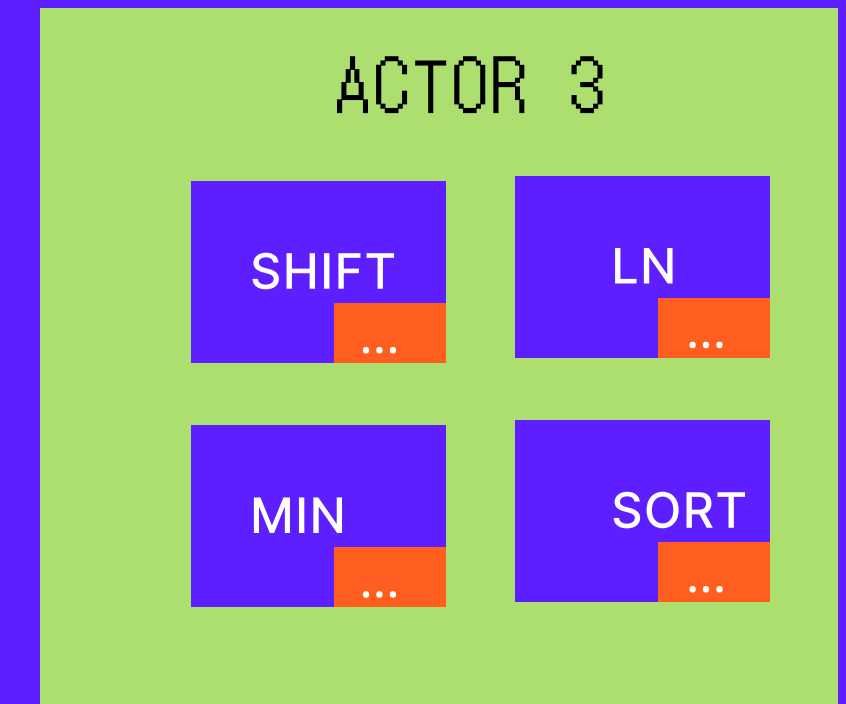
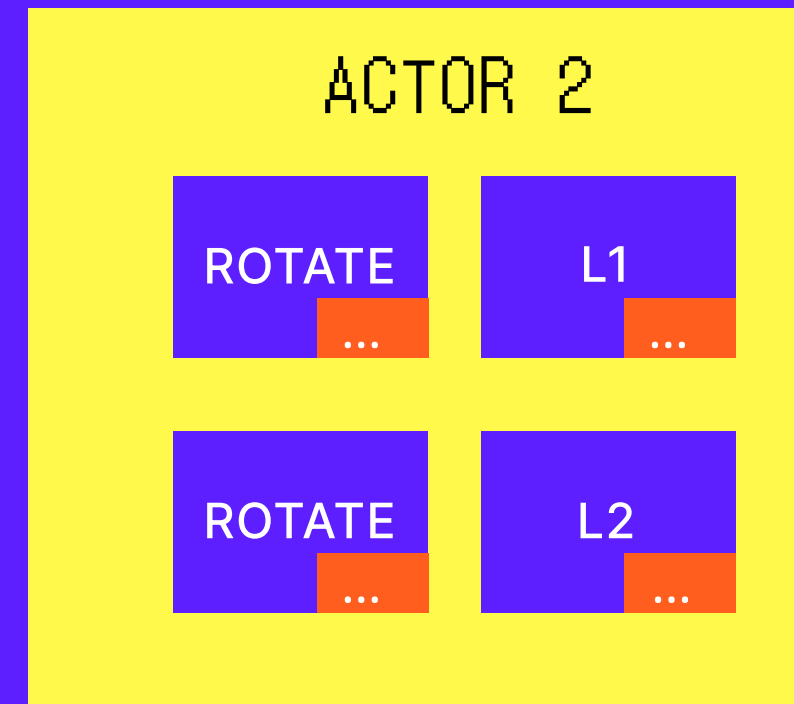
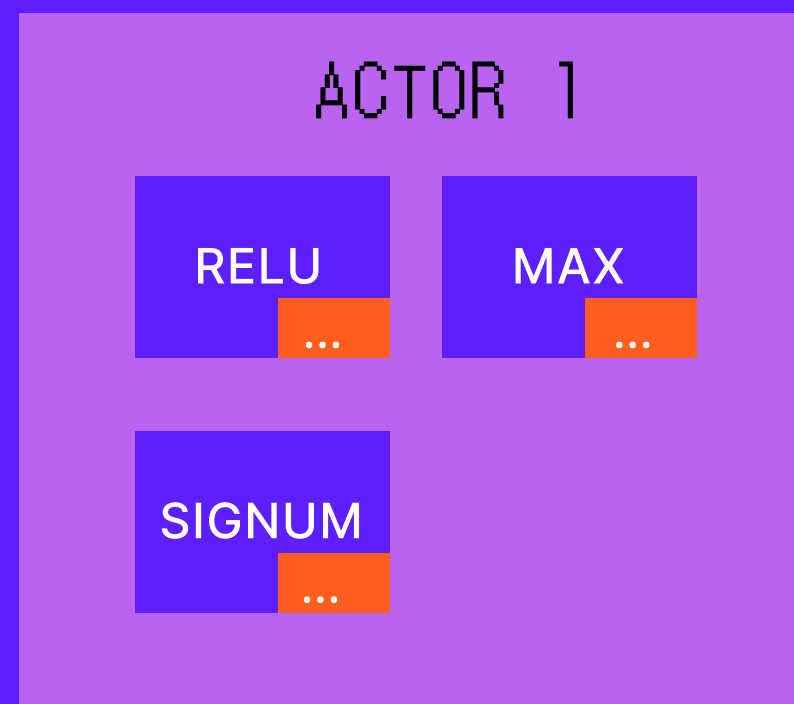
>FHE Computer

—

APPLICATION LAYER: FHE APPS

ORCHESTRATION LAYER

EXECUTION LAYER



>FHE Computer

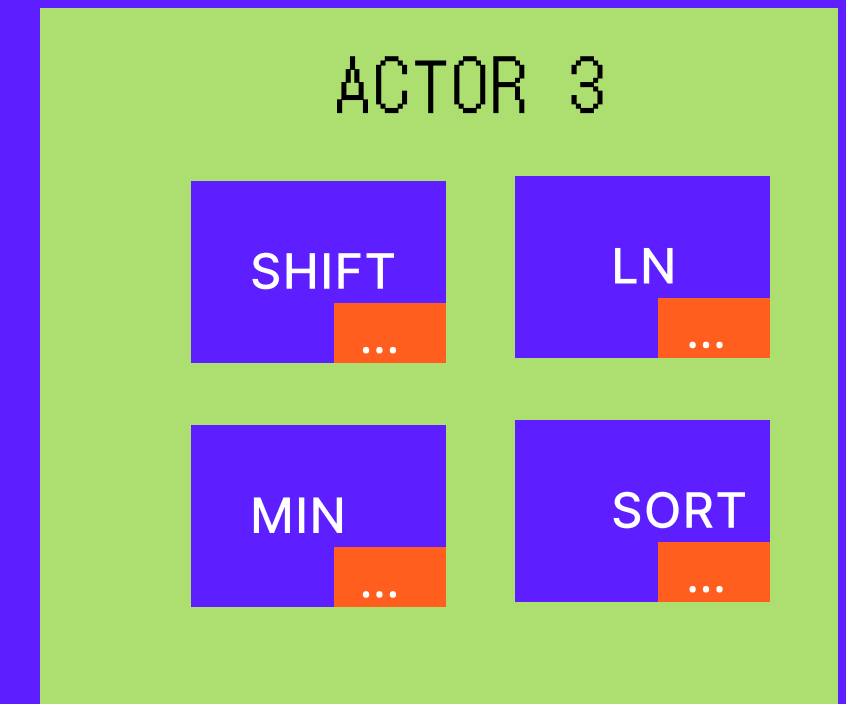
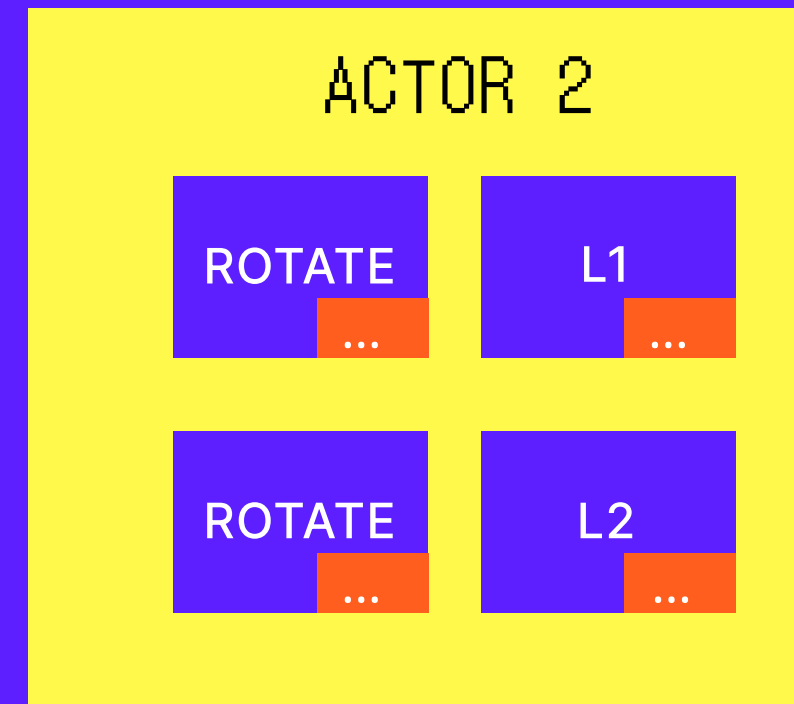
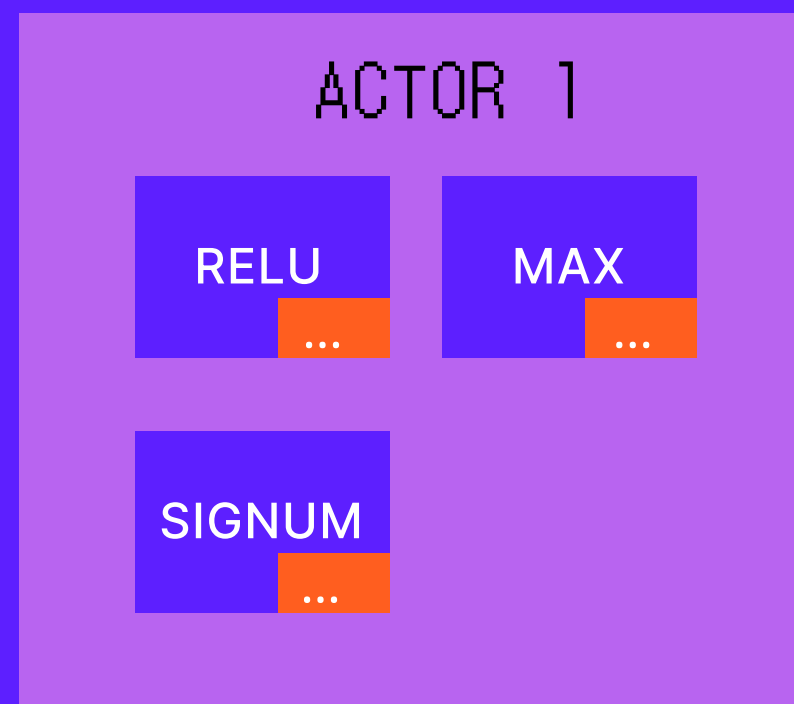
—

APPLICATION LAYER: FHE APPS

ORCHESTRATION LAYER

VERIFICATION LAYER

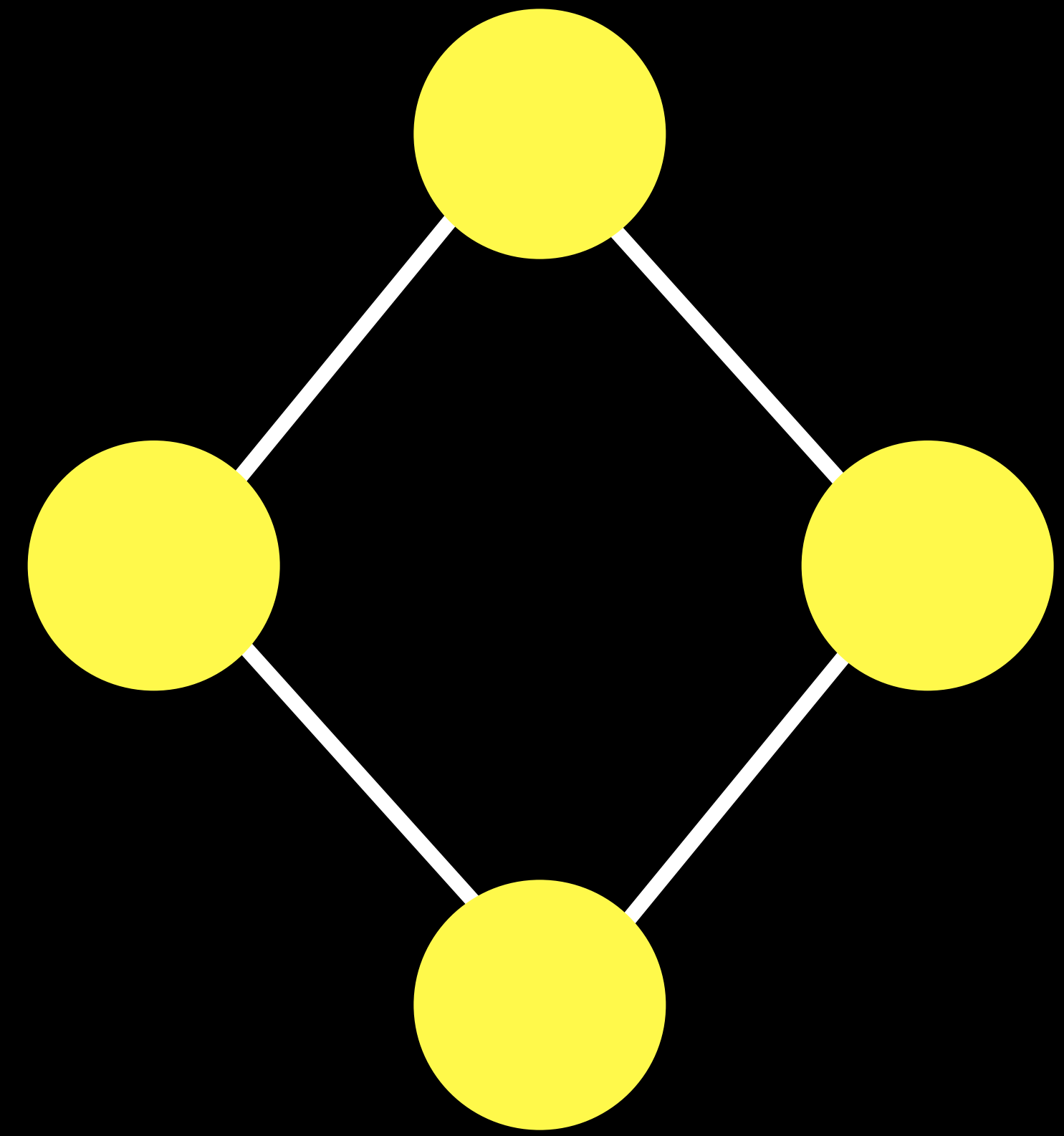
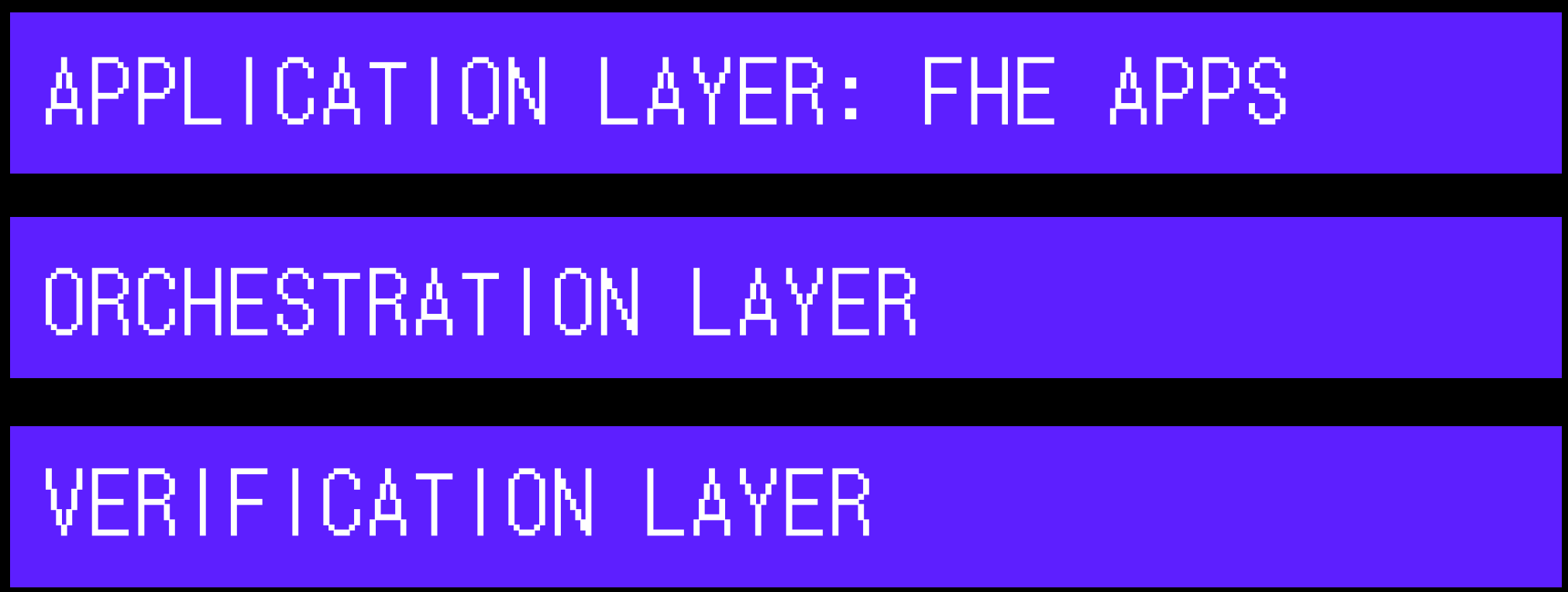
EXECUTION LAYER

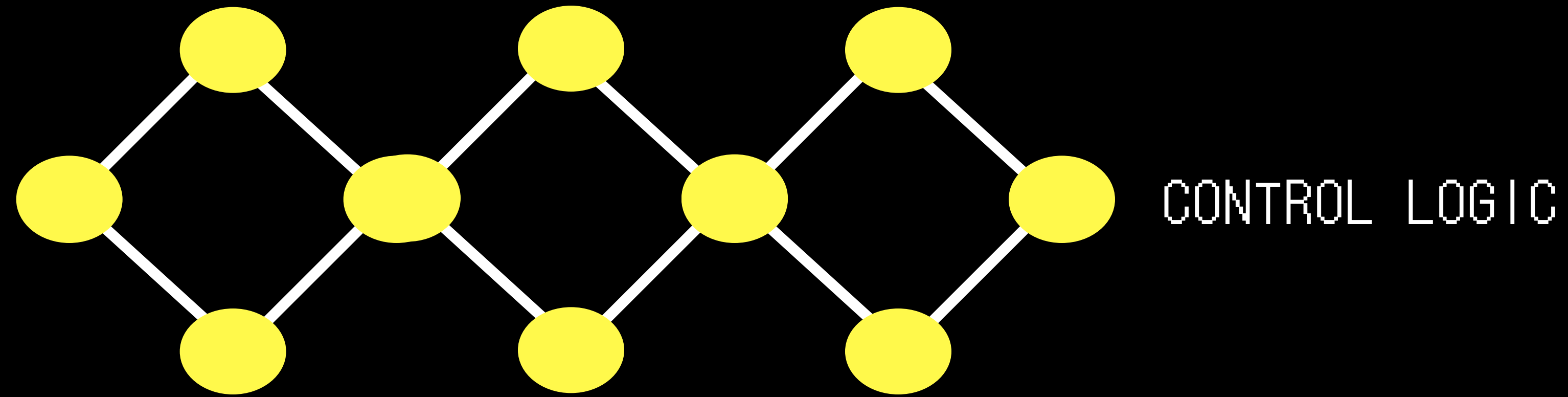


>FHE Computer

|

BLOCKCHAIN





ACTOR 1

RELU C...	MAX T...
SIGNUM B...	

ACTOR 1 is a purple rectangular block containing three blue boxes with orange accents. The first row contains 'RELU' (with 'C...' below) and 'MAX' (with 'T...' below). The second row contains 'SIGNUM' (with 'B...' below) and an empty space.

ACTOR 2

ROTATE C...	L1 C...
ROTATE C...	L2 T...

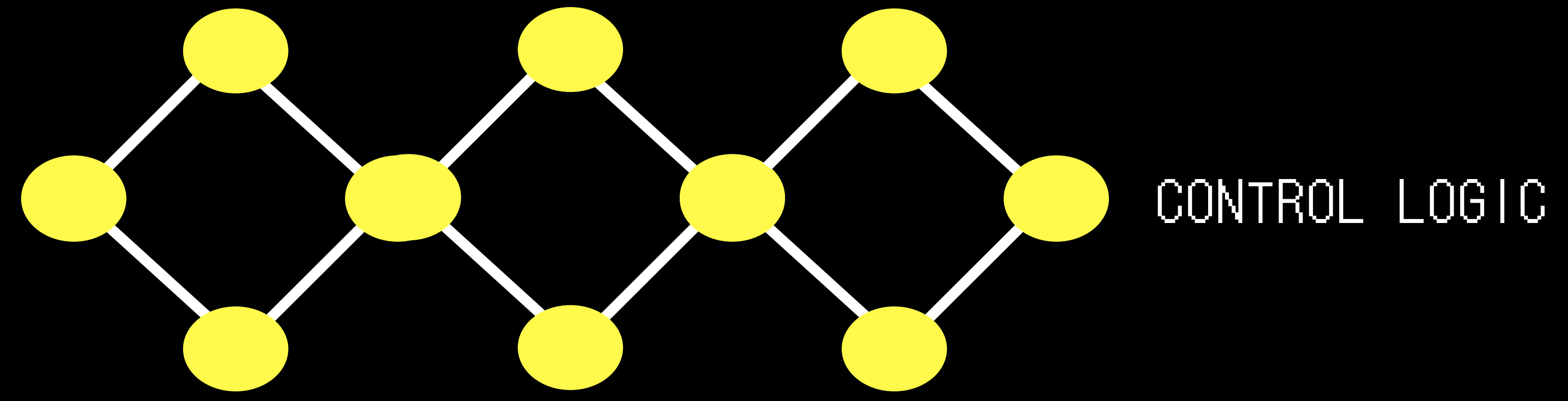
ACTOR 2 is a yellow rectangular block containing four blue boxes with orange accents. The first row contains 'ROTATE' (with 'C...' below) and 'L1' (with 'C...' below). The second row contains 'ROTATE' (with 'C...' below) and 'L2' (with 'T...' below).

ACTOR 3

SHIFT ...	LN T...
MIN ...	SORT T...

ACTOR 3 is a green rectangular block containing four blue boxes with orange accents. The first row contains 'SHIFT' (with '...' below) and 'LN' (with 'T...' below). The second row contains 'MIN' (with '...' below) and 'SORT' (with 'T...' below).

EXECUTION LAYER



CONTROL LOGIC

ACTOR 1

RELU C...	MAX T...
SIGNUM B...	

ACTOR 2

ROTATE C...	L1 C...
ROTATE C...	L2 T...

ACTOR 3

SHIFT ...	LN T...
MIN ...	SORT T...

EXECUTION LAYER

Data Layer

>FHE Computer

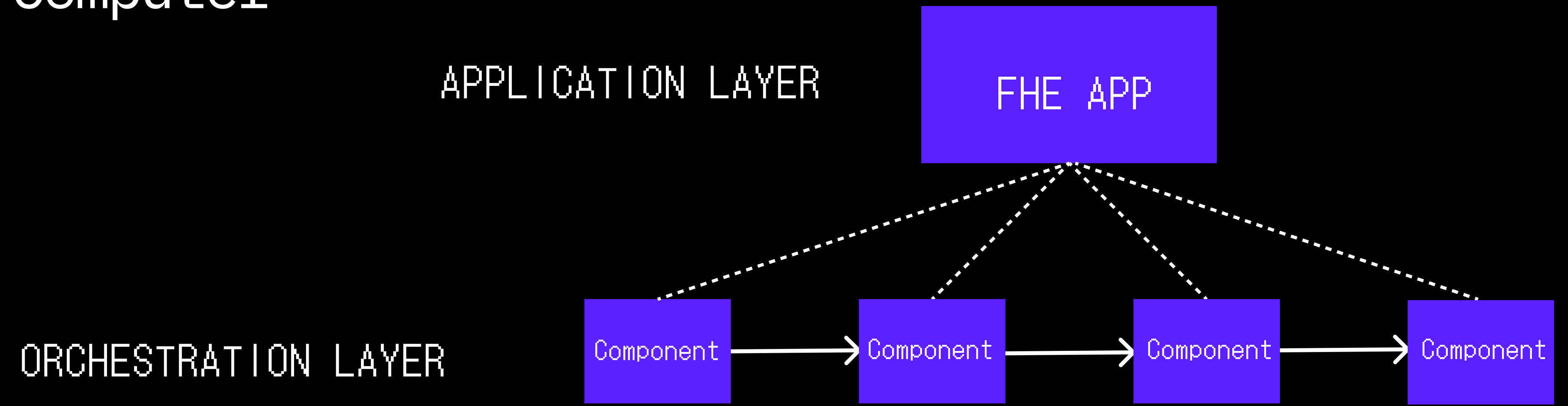
—

APPLICATION LAYER



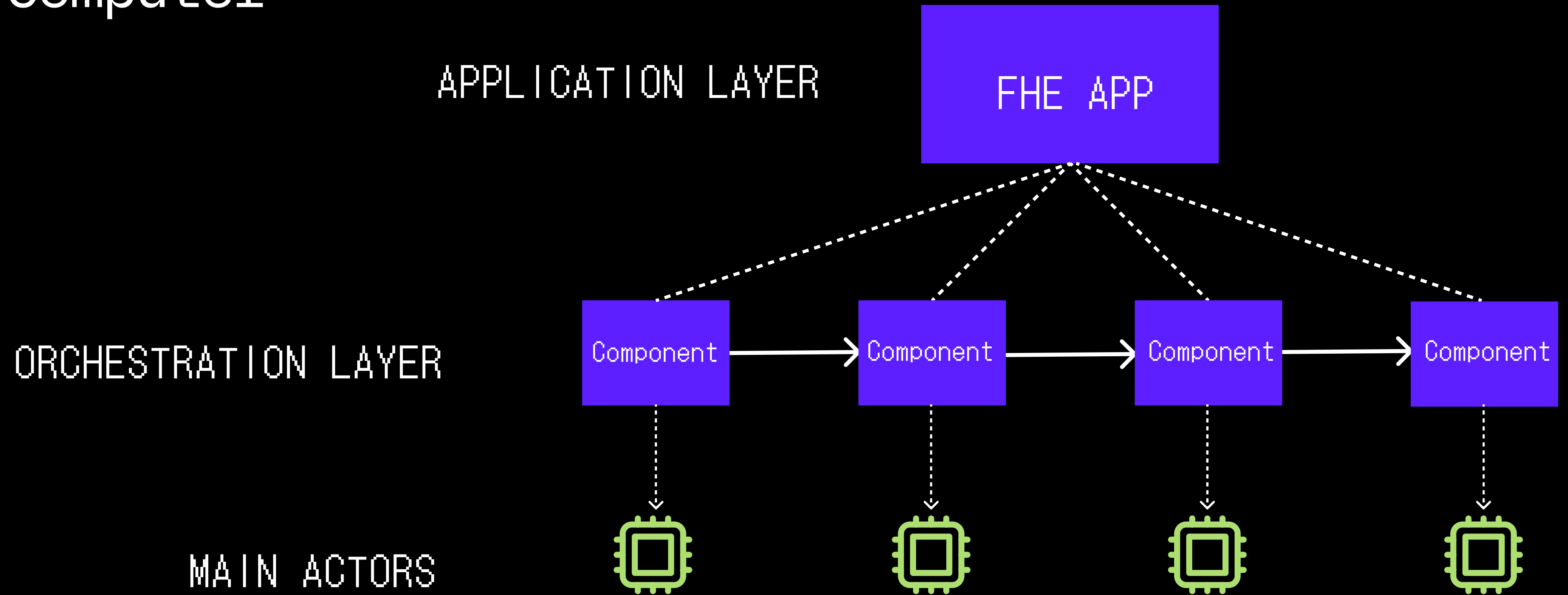
>FHE Computer

—



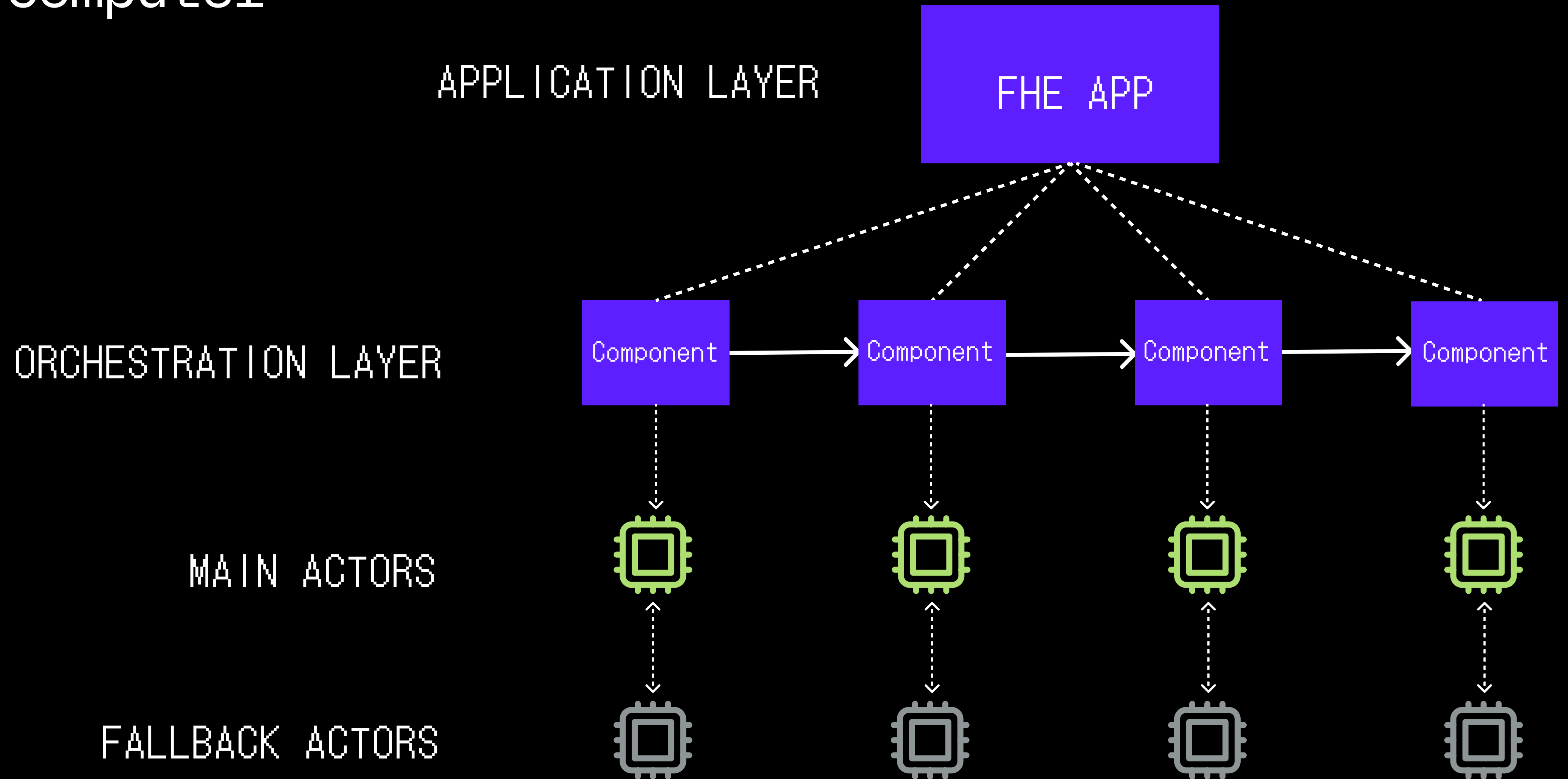
>FHE Computer

—



>FHE Computer

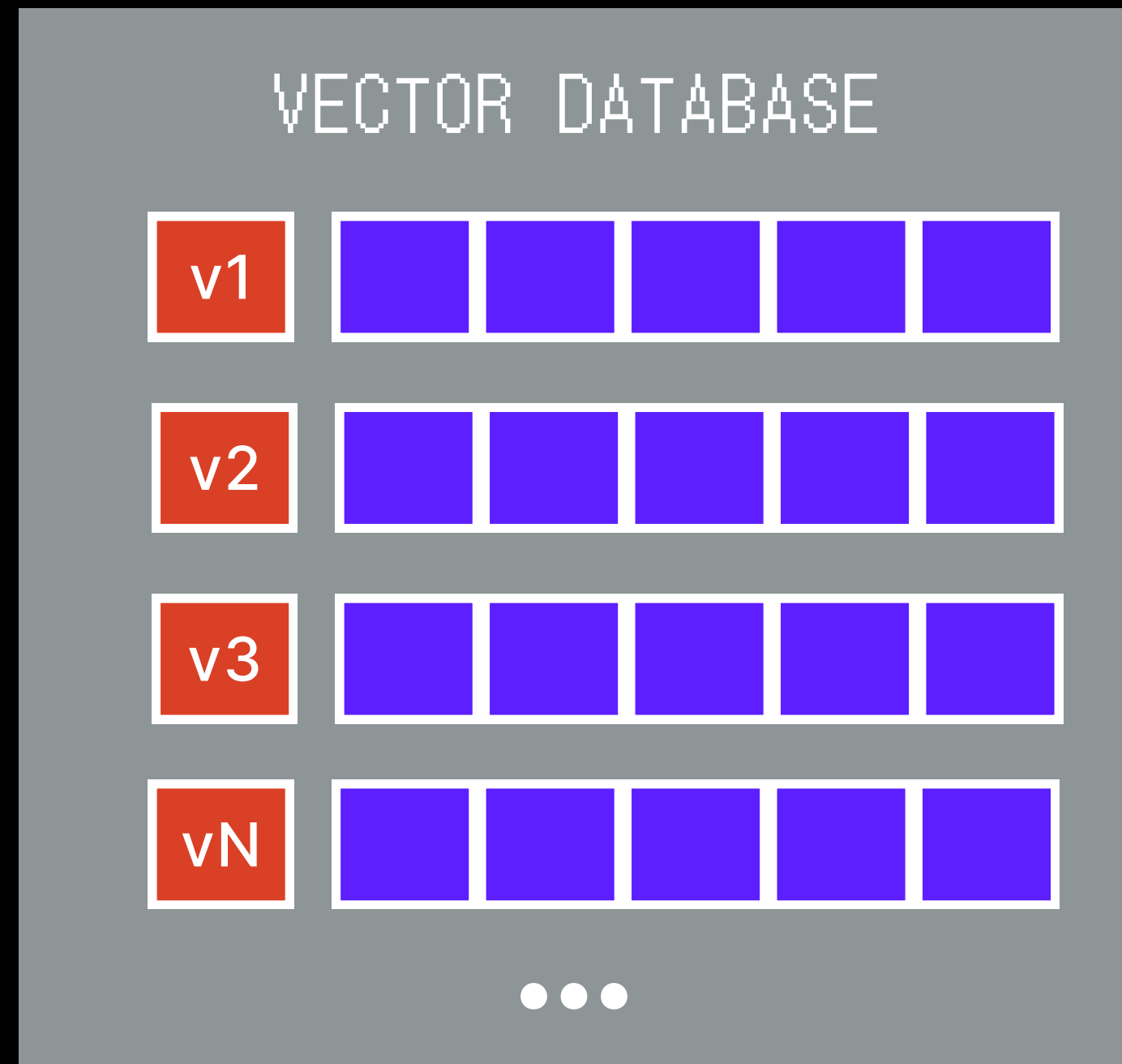
—



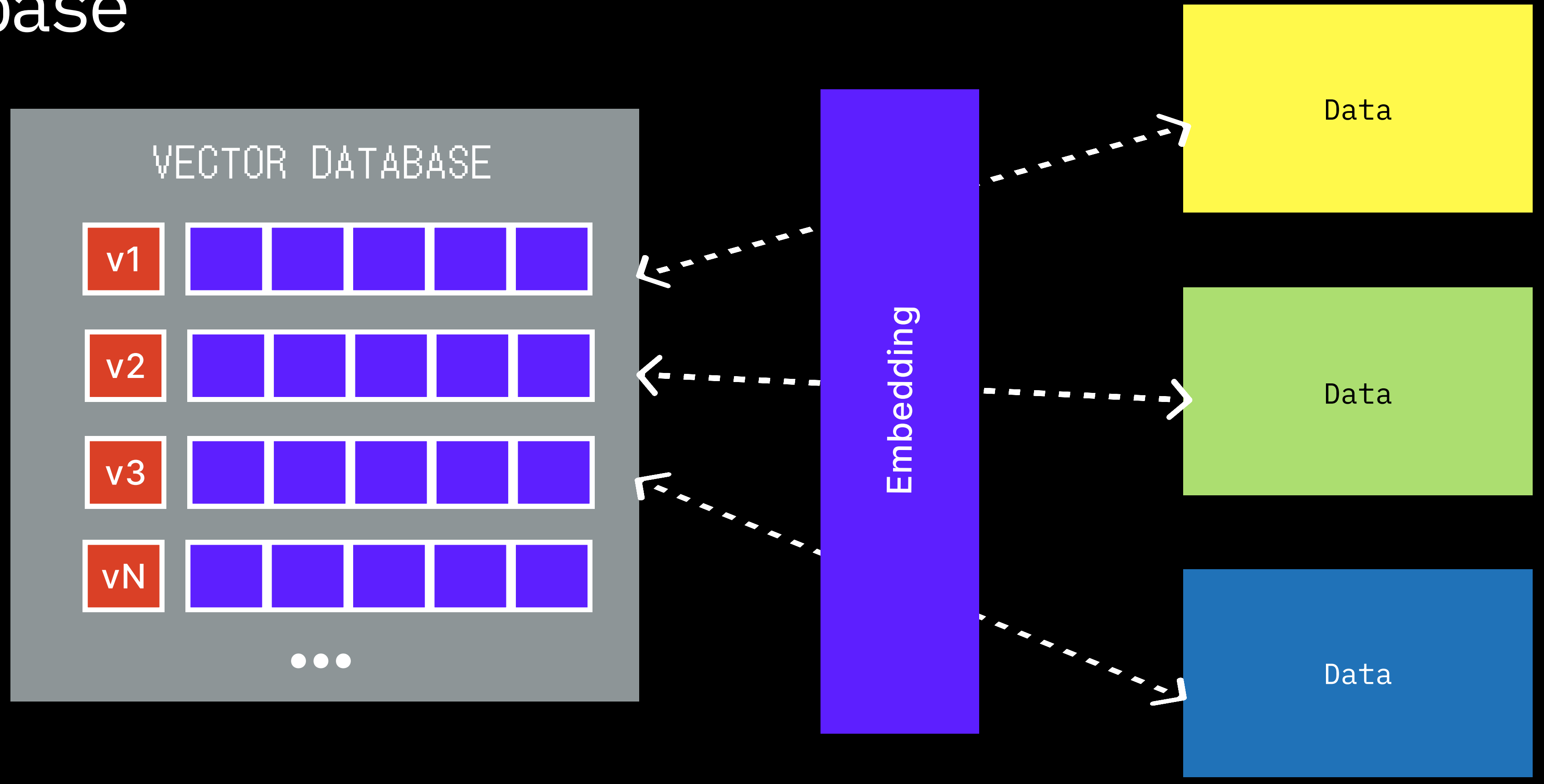
Usecases

>	VAR. 03
<h1 data-bbox="599 844 1865 1256">FHE Vector Database</h1> <p data-bbox="599 1313 2598 1407">> Ensures secure storage of sensitive data while allowing users confidential access through private requests.</p>	

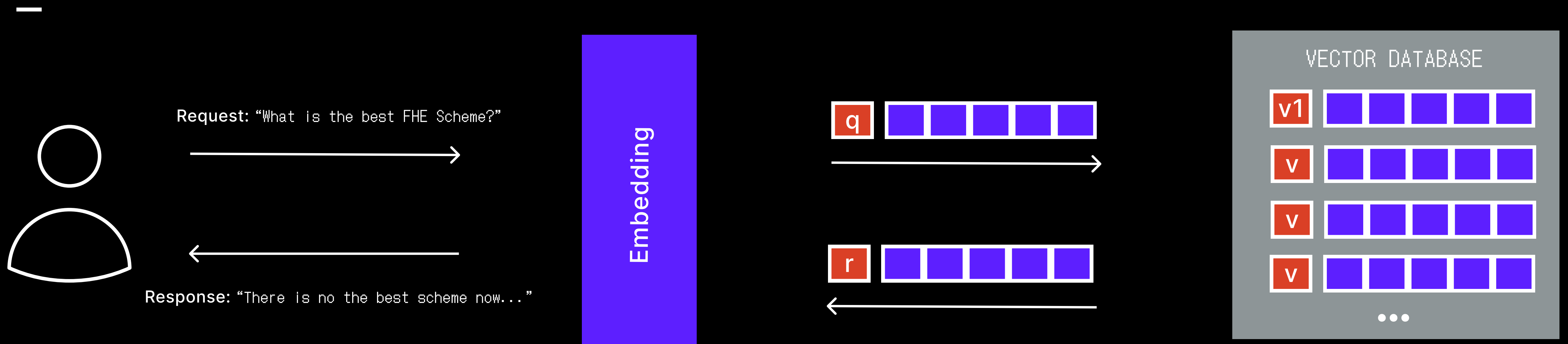
>FHE Computer:
>Vector Database



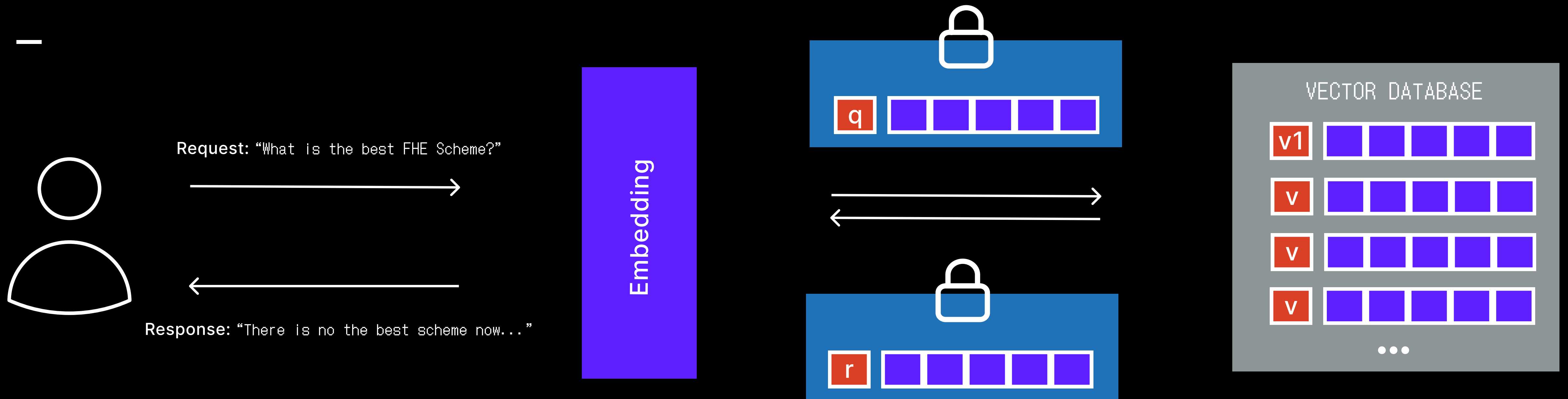
>FHE Computer:
>Vector Database



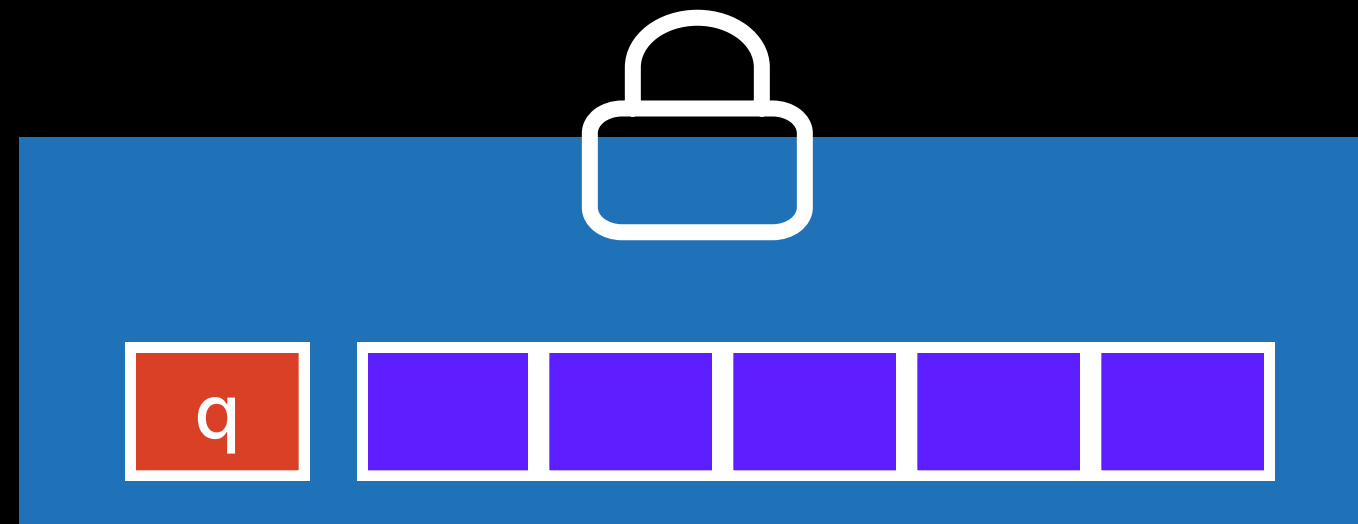
>FHE Computer: >Vector Database



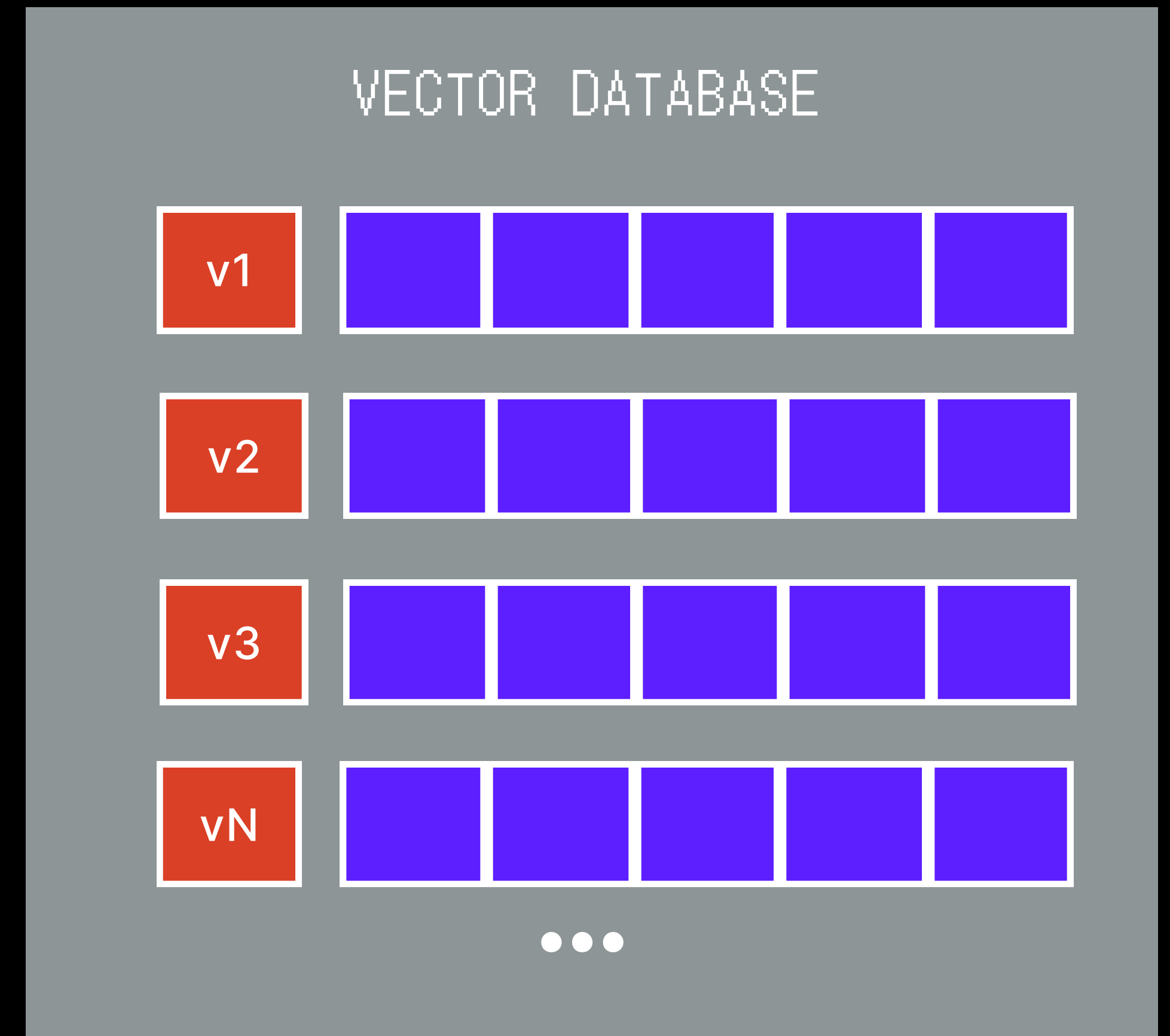
>FHE Computer: >Vector Database



>FHE Computer:
>Vector Database



Request to DB is equal to:
"Find the nearest vector based on L2 norm"



THANK YOU!

- `gurgen@fairmath.xyz`
- `@arake1ov_g`
- `fairmath.xyz`

