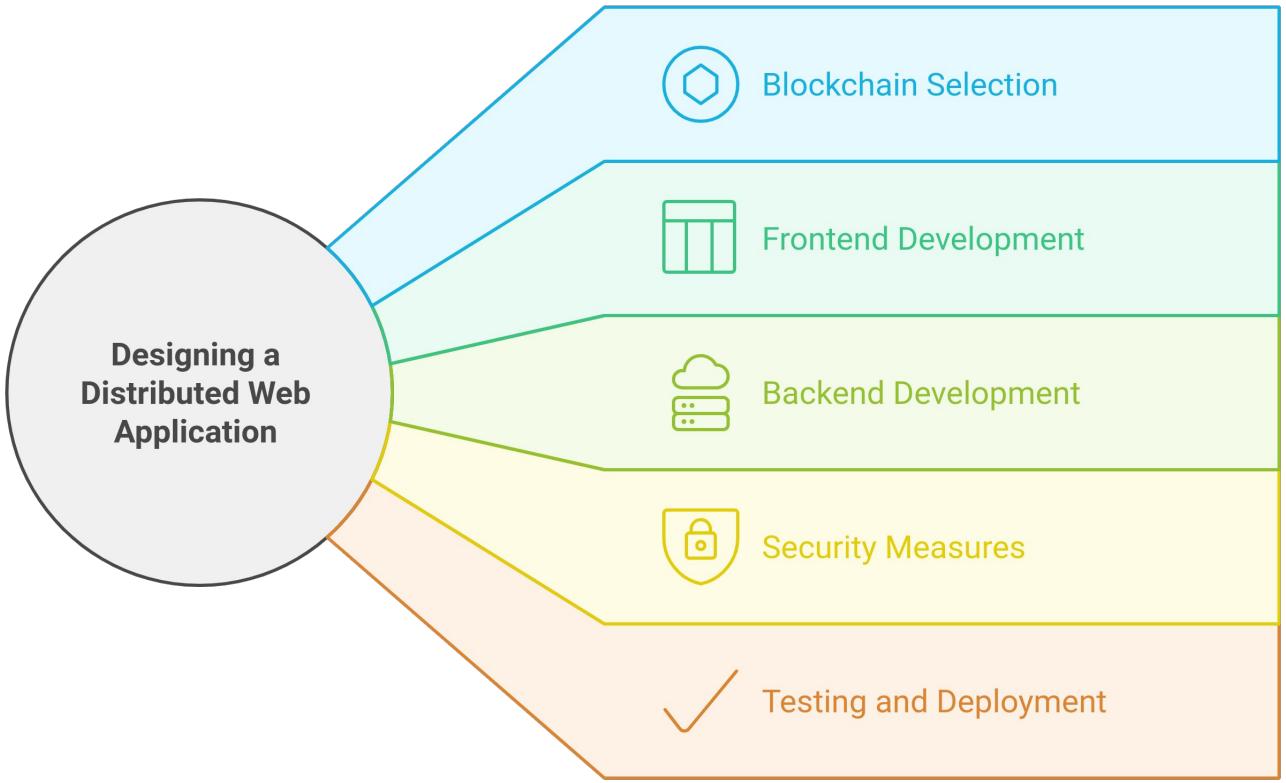
Enhancing Election Integrity

A Comprehensive Overview of a Secure Voting Distributed Application

The Idea

“Create a distributed application designed to manage a university system based on Ethereum”



The Users of the Application

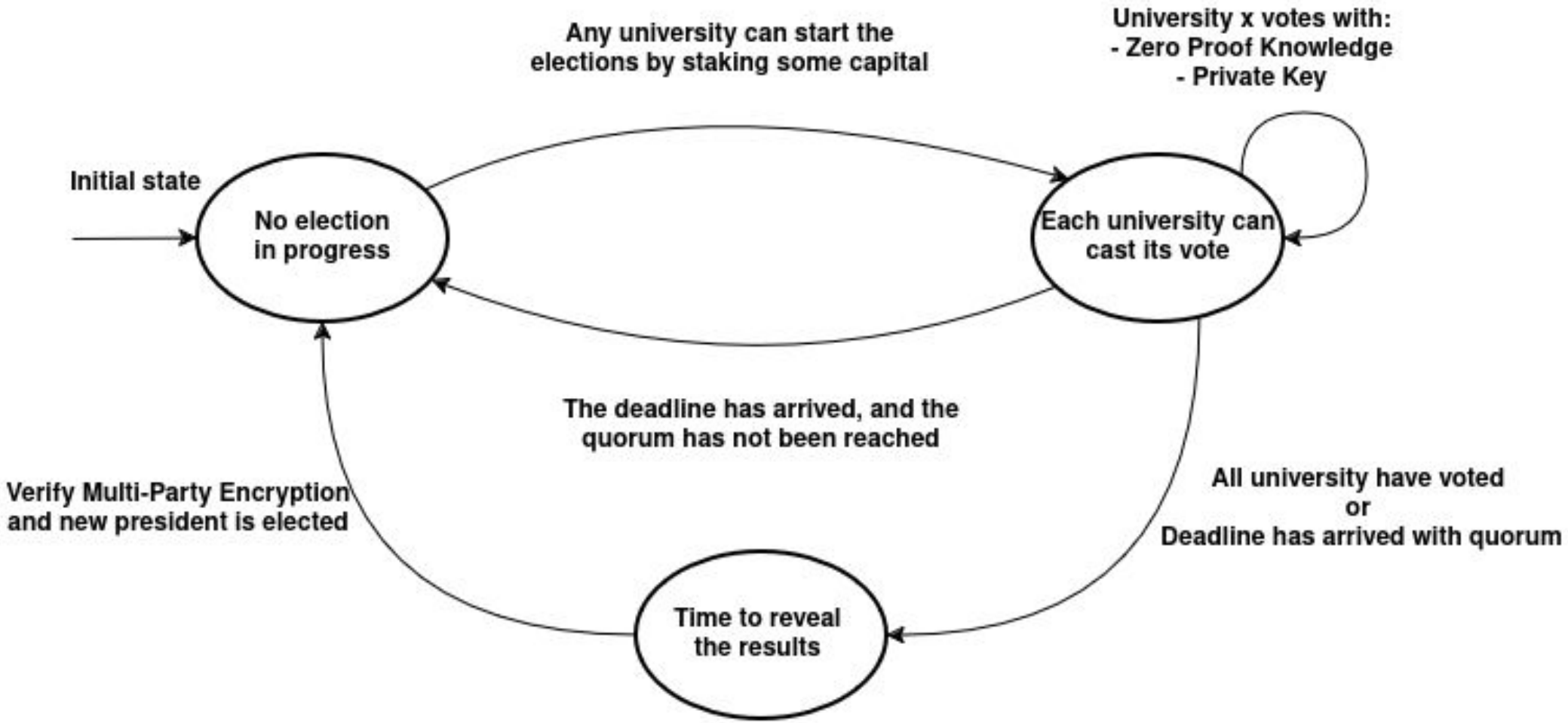
Universities:

- Initiate a presidential election.
- Vote during the election.
- Monitor the number of registered professors.

Professors:

- Register with a university by paying a registration fee.
- Withdraw from a university.

The Protocol



Can we trust the President ?

In our protocol, the president is never involved.

Can we trust all universities?

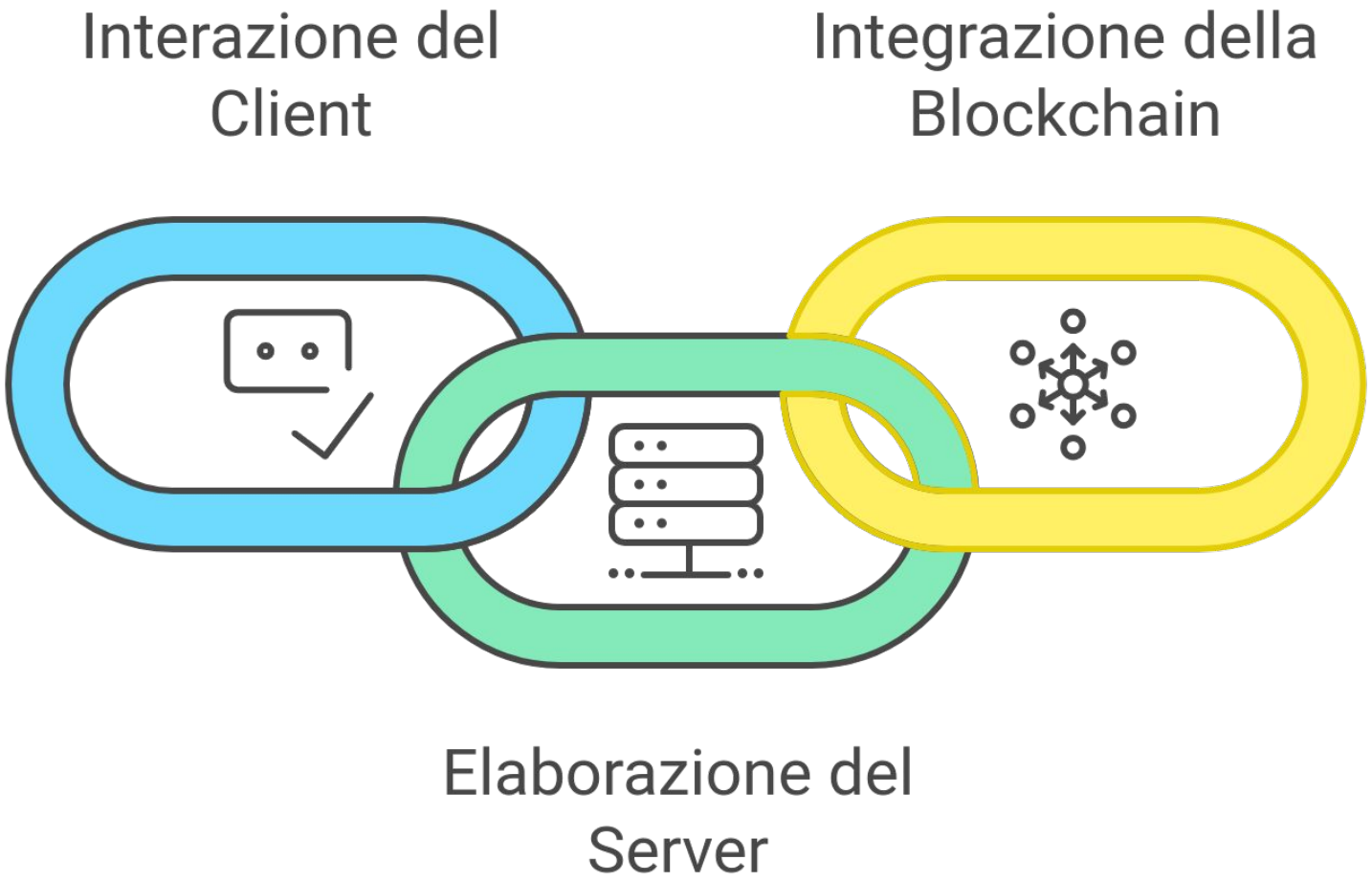
No university, regardless of its level of corruption, will have enough people to manipulate the voting process; because no university can have more than a certain number of professors in the association.

Resilience of the Blockchain to Attacks

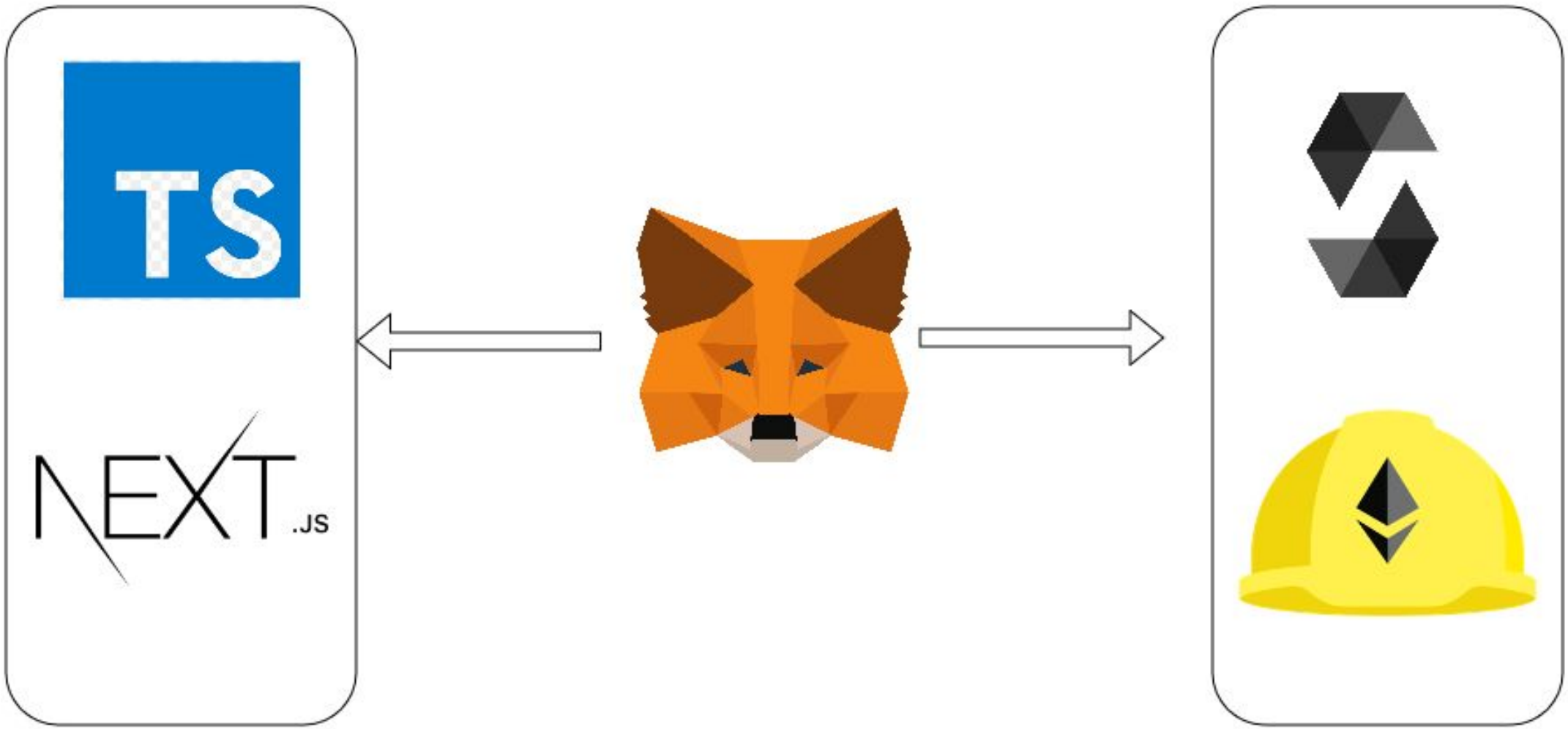
Blockchain is utilized to ensure that the university's vote is securely signed by the owner, immutable, and encrypted using homomorphic encryption.

The election results are safeguarded against attacks, as they are encrypted and cannot be altered due to being stored on the blockchain.

Structure: Client-Server-Blockchain



Tech Stack

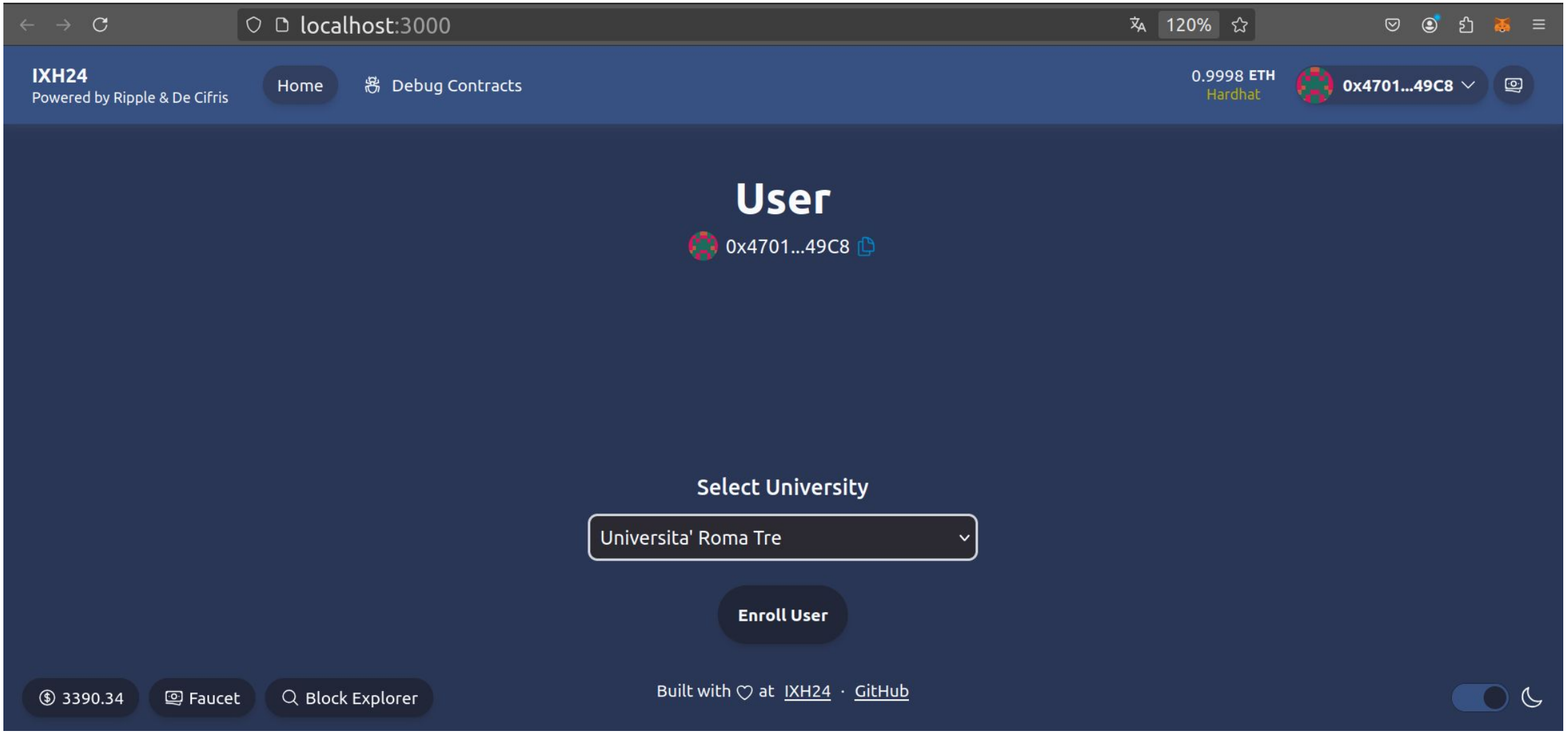


Implementation of Homomorphic Cryptographic

packages > nextjs > crypto > TS fhe.ts > ...

```ts
import * as paillier from 'paillier-bigint';

async function getKeyPair() {
    const { publicKey, privateKey } = await paillier.generateRandomKeys(3072);
    return { publicKey, privateKey };
}

async function encrypt(publicKey: paillier.PublicKey, value: bigint) {
    return publicKey.encrypt(value);
}

async function decrypt(privateKey: paillier.PrivateKey, value: bigint) {
    return privateKey.decrypt(value);
}

function sum_encrypted(publicKey: paillier.PublicKey, ...values: bigint[]) {
    return publicKey.addition(...values);
}

const { publicKey, privateKey } = await getKeyPair();

export { encrypt, decrypt, sum_encrypted, publicKey, privateKey };
```

Demo


localhost:3000
120%
IXH24
Powered by Ripple & De Cifris
Home
Debug Contracts
0.9998 ETH
Hardhat
0x4701...49C8
User
0x4701...49C8
Select University
Universita' Roma Tre
Enroll User
3390.34
Faucet
Block Explorer
Built with at IXH24 · GitHub