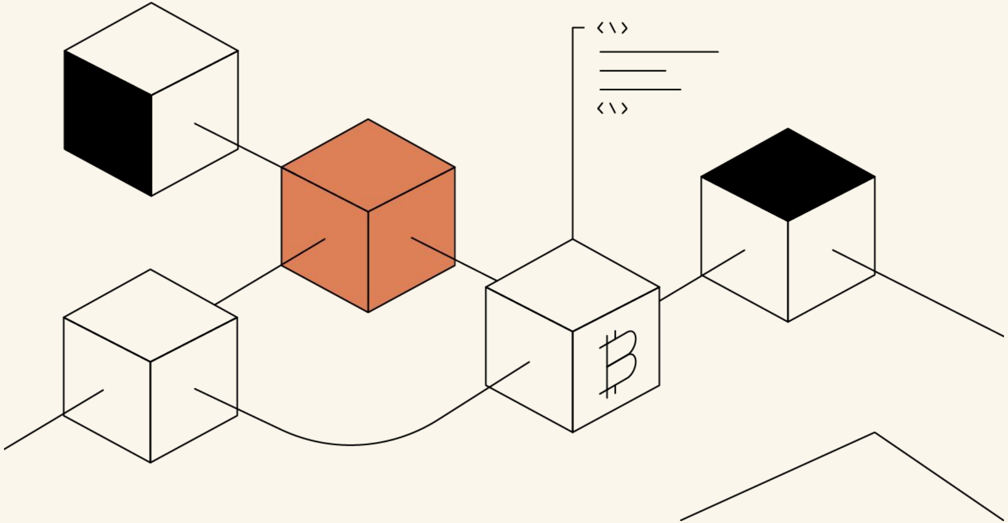
Enabling Fully Confidential Transactions with Zero Knowledge

Tal Derei

Agenda



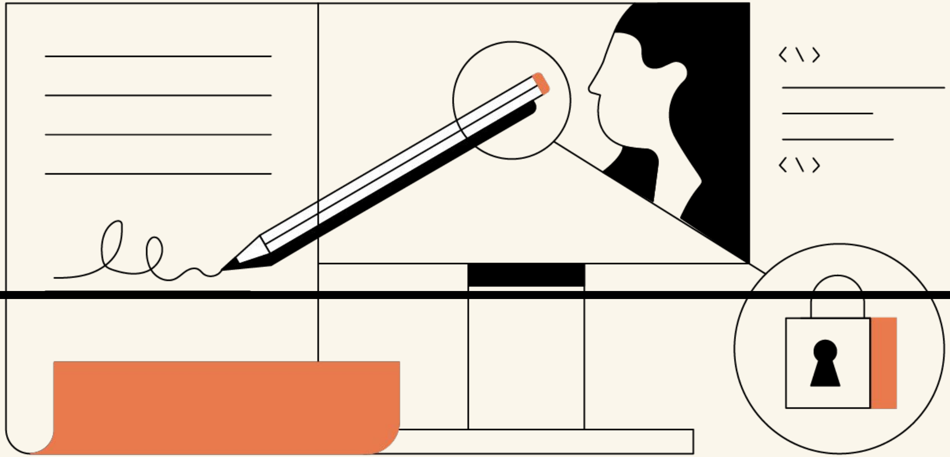
1. **Privacy on Public Blockchain**
2. **Aztec Protocol**
3. **Aztec's Applications (zk.money and Connect Bridge)**

Terms

- **L1** = Layer-1 (Main Chain)
- **L2** = Layer-2 (ZK-Rollups)
- **ZK** = Zero-Knowledge
- **zk-SNARKs** = Succinct Non-Interactive Argument of Knowledge *Proofs*

Agenda

> **“Obfuscation of a money trail isn't necessarily money laundering!”**



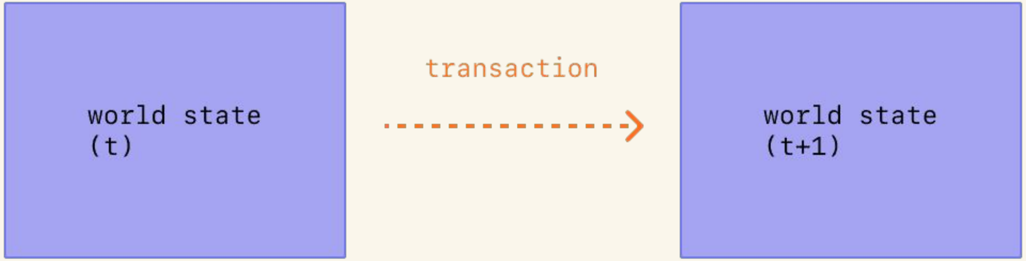
Privacy On Public Chain

Public Transaction Model

→ **1.** An account will initiate a transaction to update the state of the Ethereum network

→ **2.** Transactions are cryptographically signed instructions from **accounts** (i.e. accounts initiate transactions)

→ **3.** Miner executes the transaction, initiating a **state change** of the EVM that's broadcasted to the entire network (e.g. *Gossip Protocol* + *Consensus Mechanism)*



Public Transaction Model

Korth sends Palmieri → 10 ETH

- **Korth ‘s account is debited: -10.0042 ETH**
- **Palmieri’s account is credited: 10.0042 ETH**
- **Base fee (*EIP-1559*) will be burned: -0.00399 ETH**
- **Miner keeps the tip: +0.000210 ETH**

Executing Transactions

Transaction Object’s Payload:

```
{
    from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
    to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
    gasLimit: "21000",
    maxFeePerGas: "300",
    maxPriorityFeePerGas: "10",
    nonce: "0",
    value: "10000000000"
}
```

Signing Transactions

→ Transaction object needs to be signed using the sender private key

JSON-RPC Call:

```
    {
            "id": 2,
            "jsonrpc": "2.0",
            "method": "account_signTransaction",
            "params": [
                {
                    "from": "0x1923f626bb8dc025849e00f99c25fe2b2f7fb0db",
                    "gas": "0x55555",
                    "maxFeePerGas": "0x1234",
                     "maxPriorityFeePerGas": "0x1234",
                    "input": "0xabcd",
                    "nonce": "0x0",
                    "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
                    "value": "0x1234"
                }
            ]
}
```

Transaction Response

JSON-RPC Response:

```
{
        "jsonrpc": "2.0",
        "id": 2,
        "result":
                {
                "raw":"0xf88380018203339407a565b7ed7d7a678680a4c162885bedbb695fe080a44401a6e400000000",
                "tx": {
                        "nonce": "0x0",
                        "maxFeePerGas": "0x1234",
                        "maxPriorityFeePerGas": "0x1234",
                        "gas": "0x55555",
                        "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
                        "value": "0x1234",
                        "input": "0xabcd",
                        "v": "0x26",
                        "r": "0x223a7c9bcf5531c99be5ea7082183816eb20cfe0bbc322e97cc5c7f71ab8b20e",
                        "s": "0x2aadee6b34b45bb15bc42d9c09de4a6754e7000908da72d48cc7704971491663",
                        "hash": "0xeba2df809e7a612a0a0d444ccfa5c839624bdc00dd29e3340d46df3870f8a30e"
                        }
                }
 }
```

Potential Problem?

Sender and Recipient addresses are <u>PUBLIC</u>!

Pseudonymous, but still…

<u>Unanswered Questions:</u>

Q. Want to receive annual salary in cryptocurrency, but don’t want to reveal how much you make?

Q. Want to receive interest/dividend payments (i.e. more complex payment types)?

Q. Want to Interact with DeFi protocols privately (i.e. to take out a loan)?

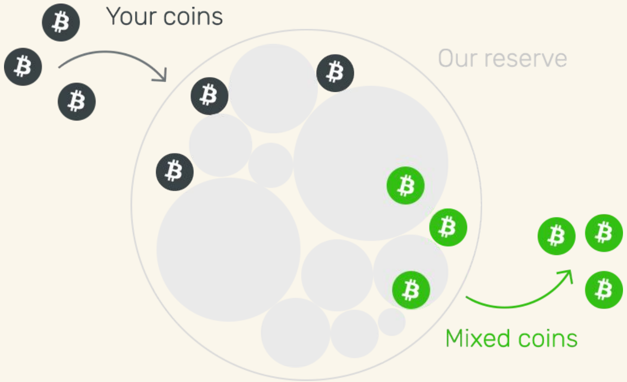
Potential Solution?

CryptoMixer

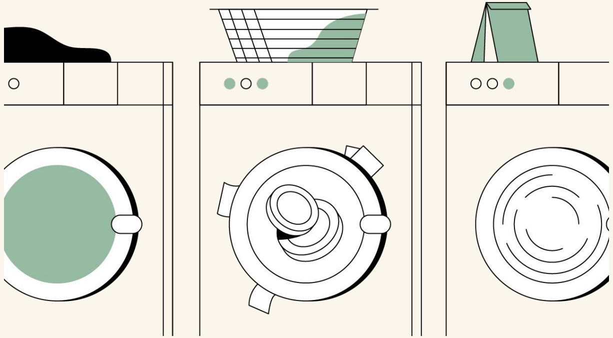
<u>On-Chain Mixers</u>

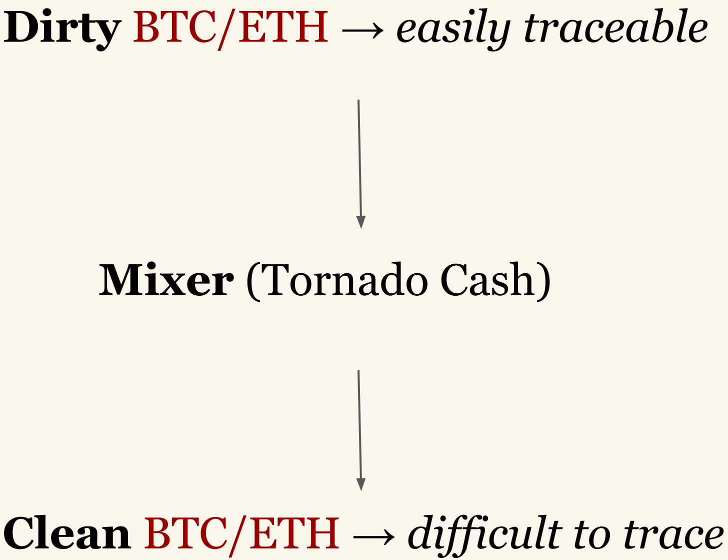
On-Chain Mixers

- **On-Chain Mixers** = Provide services that mix and shuffle cryptocurrency
  - For a small fee, mixers allow users to obscure the exact <u>chain of custody</u> of funds and <u>secure their privacy</u>



On-Chain Mixers





Dirty BTC/ETH → *easily traceable*

Mixer (Tornado Cash)

Clean BTC/ETH → *difficult to trace*

Washing Money 101

1. **Placement**
2. **Layering**
3. **Integration**

Tornado Cash

Tornado Cash: Ethereum-based mixer developed by Zcash

→ Improves the privacy of transactions by breaking the on-chain link between a source and a destination address

→ Transactions are kept anonymous using **zk-SNARK proofs**

tornado

Tornado Cash

Q. How Does Tornado Cash Work?

1. Deployed smart contract that accepts ETH deposits

2. User deposits ETH into smart contract + generates secret + send a hash (**called a commitment**) along with the deposit amount

3. Smart contract adds the funds to its list of deposits

4. User has to provide the corresponding secret that matches the unspent deposit from the Tornado Cash deposit list upon withdrawal

Chain-Hopping

Chain-hopping: mixing funds across multiple accounts and exchanges

e.g. $4.5 billion Bitfinex hack in 2016

BITFINEX

Are On-Chain Mixers Sufficient?

NO!

→ <u>High</u> Gas Fees

- ○ Deposit + Withdrawal (⅓ deposit fee) + relayer tx fees
- ○ L2 solutions that are 25x cheaper than on-chain mixers

→ <u>Low</u> Privacy Set

Unanswered Questions

Q. More efficient way to receive annual salary in cryptocurrency, but don’t want to reveal how much you make?

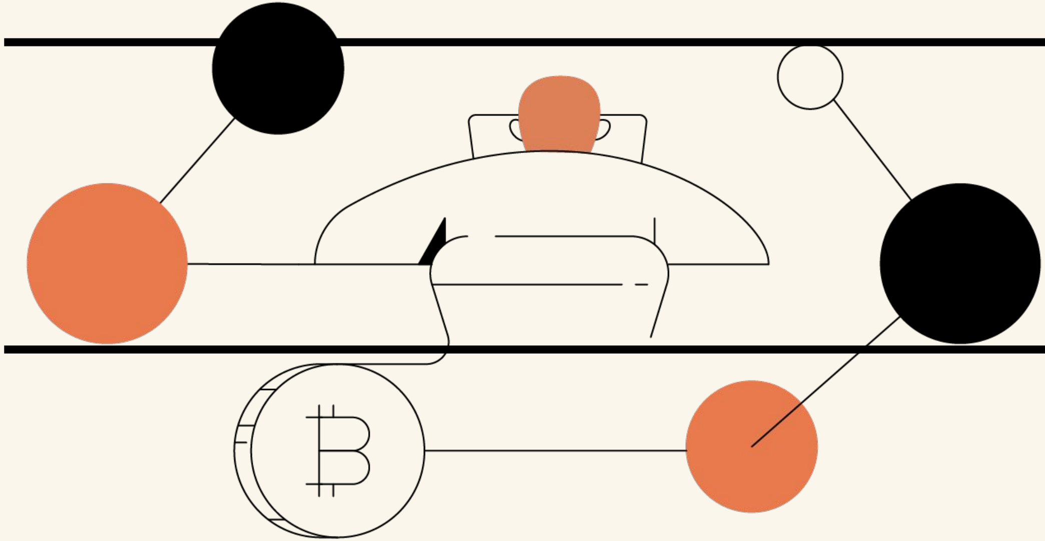
Q. Want to receive interest/dividend payments (i.e. more complex payment types)?

Q. Want to Interact with DeFi protocols privately (i.e. to take out a loan)?

New Solution

Enable developers to build Privacy-focused DApps w/ “shielded transactions” built on L2 zk-rollups

Aztec Protocol to the rescue!



Aztec Protocol

Problems

Building Zero-knowledge systems on Ethereum is **flawed...**

→ Slow proof construction

→ Expensive on-chain verification gas costs

→ Lack of interoperability between zk assets

aztec

Aztec Protocol

<u>Aztec Protocol</u>: open source zero-knowledge protocol for building **privacy** on blockchains

→ PLONK

- Recursive zk-SNARK Proof System
- ZK prover behind their zk-zk-rollup

→ NOIR

- zkSNARK programming language for programmable private smart contracts
- Alternative to achieving zkEVM functionality

aztec

Aztec Protocol

Components:

→ Standardized API

→ L2 ZK-ZK-Rollup (**ZK.Money**)

→ Bridge (**Aztec Connect**)

aztec

Standardized API

Standardized API

Aztec is focusing on a **generic solution** for confidential transactions and confidential cross-asset settlements using…

.

.

.

Zero-Knowledge Proofs

aztec

Building Blocks

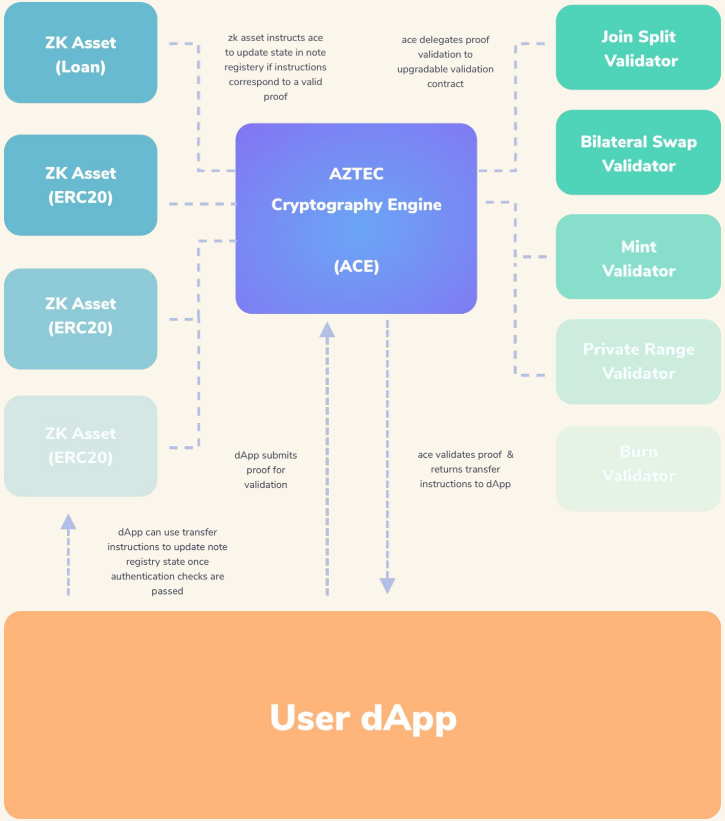
Building Blocks

- Family of **Zero Knowledge Proofs,** sharing a common reference string
  - Modular proofs for different business logic (i.e. *interest rate proof*)
  - Efficient range proofs (allows the **prover** to prove to a **verifier**, that a number is within a specific range)

- Cryptography Engine, ACE (**ERC-1723**) - shared suite of zk validator smart contracts for zk proofs
  - ACE accept zk proofs and spits out transfer instructions

- A confidential token standard (**ERC-1724**)
  - Confidential asset with common interface

System Design



ZK Asset (Loan)

ZK Asset (ERC20)

ZK Asset (ERC20)

ZK Asset (ERC20)

zk asset instructs ace to update state in note registery if instructions correspond to a valid proof

ace delegates proof validation to upgradable validation contract

AZTEC Cryptography Engine

(ACE)

Join Split Validator

Bilateral Swap Validator

Mint Validator

Private Range Validator

Burn Validator

dApp submits proof for validation

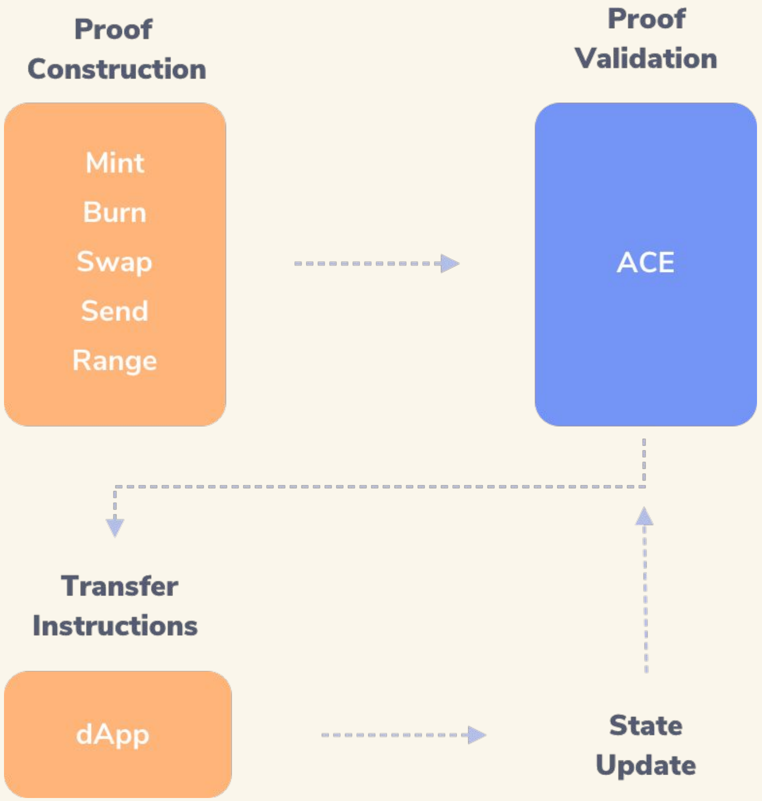
ace validates proof & returns transfer instructions to dApp

dApp can use transfer instructions to update note registry state once authentication checks are passed

User dApp

aztec

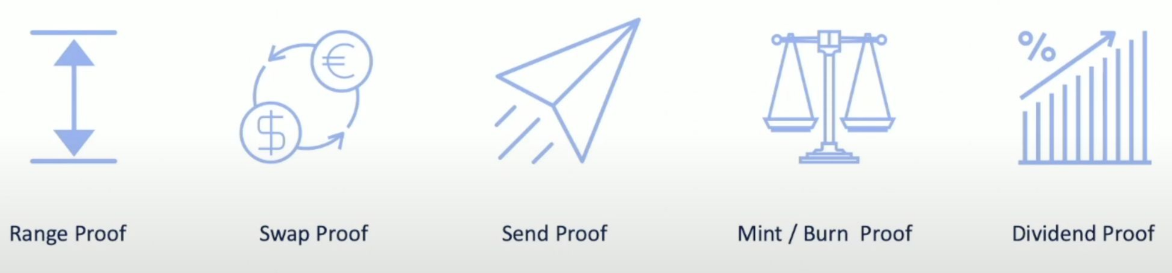
System Design



aztec

Family of ZK Proofs

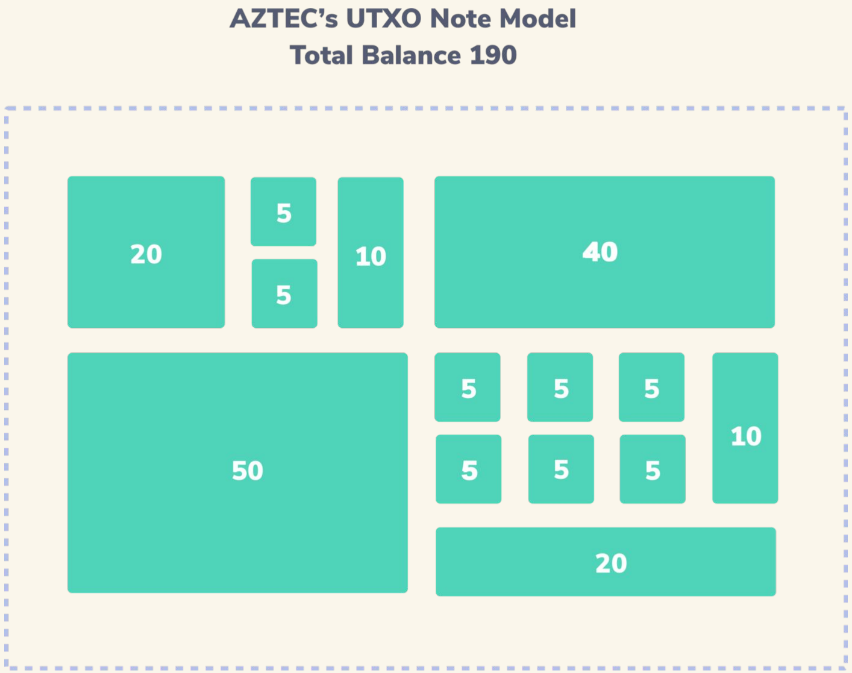
"AZTEC proofs are the **building blocks** for privacy in Ethereum and allow **discrete chunks of logic** to be executed confidentially on-chain."



aztec

Aztec Note

AZTEC follows a **UTXO** model similar to that of Bitcoin. The core of any AZTEC transaction is a **Note**



aztec

Confidential Transfers are *Balancing Relationships…*

ERC-1723 note

(encrypted representation of value)

Makeup of a note:

On-Chain Data

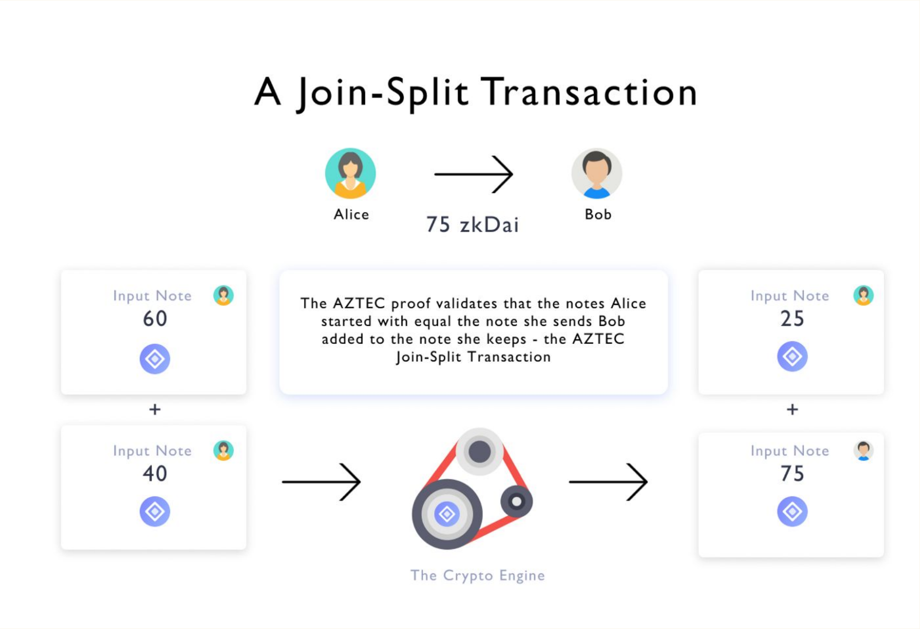
→ 1. ETH address of **owner**
→ 2. Aztec note **public key**
→ 3. Aztec **metadata**

Private Data

→ 1. Note **value**
→ 2. Note **viewing key**
→ 3. Note **spending key**

aztec

Confidential Transfers are *Balancing Relationships…*



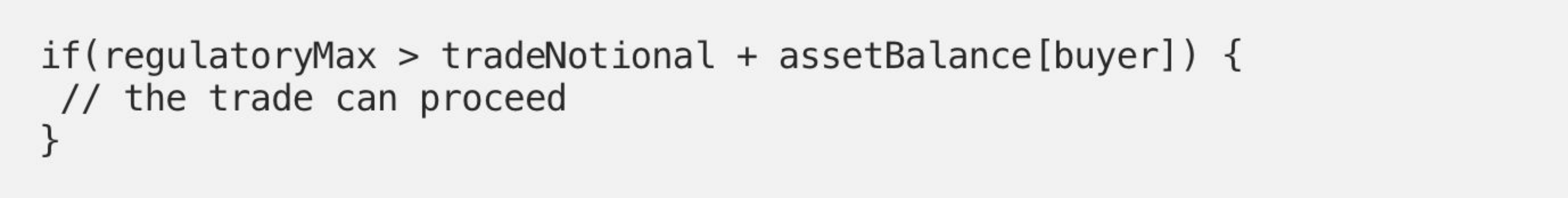
A Join-Split Transaction

The AZTEC proof validates that the notes Alice started with equal the note she sends Bob added to the note she keeps - the AZTEC Join-Split Transaction

"**Join Split** proof allows a set of input notes to be joined or split into a set of output notes."

Example: Range Proof

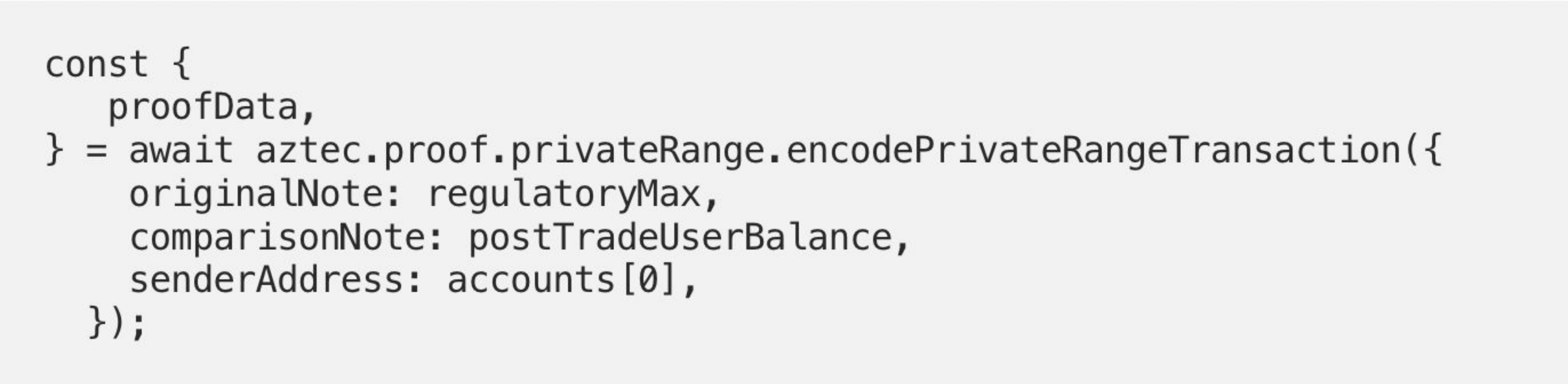
Want to prove: a traders post trade asset balance is less than a regulatory maximum



```
if(regulatoryMax > tradeNotional + assetBalance[buyer]) {
 // the trade can proceed
}
```

Example: Range Proof

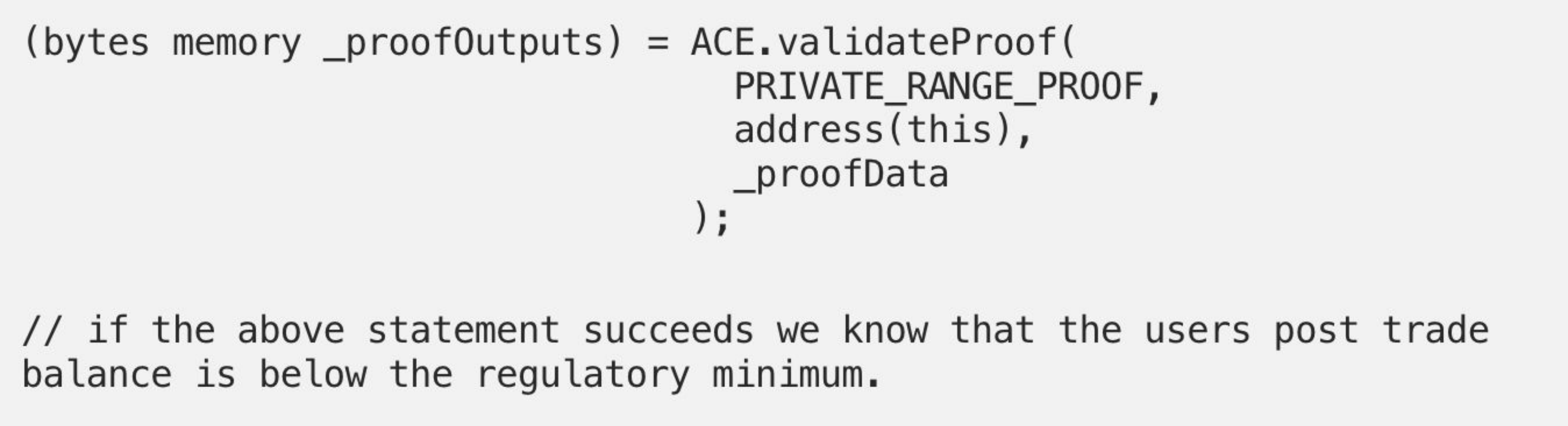
Aztec Dapp performs the same check using **AZTEC proofs**



```
const {
   proofData,
} = await aztec.proof.privateRange.encodePrivateRangeTransaction({
    originalNote: regulatoryMax,
    comparisonNote: postTradeUserBalance,
    senderAddress: accounts[0],
  });
```

Example: Range Proof

Once the proof is constructed, it can be relayed to ACE for validation.

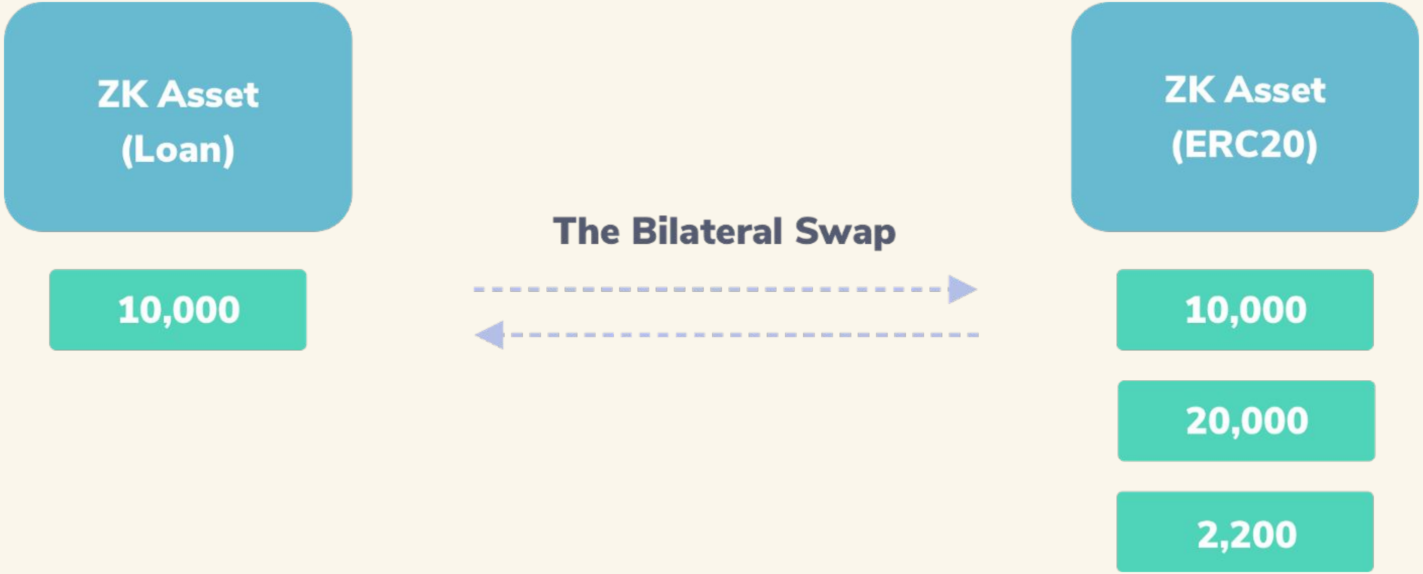


```
(bytes memory _proofOutputs) = ACE.validateProof(
                                 PRIVATE_RANGE_PROOF,
                                 address(this),
                                 _proofData
                               );


// if the above statement succeeds we know that the users post trade
balance is below the regulatory minimum.
```

Example: Bilateral Swap

Exchange between two zk assets



ZK Asset (Loan)

10,000

The Bilateral Swap

ZK Asset (ERC20)

10,000

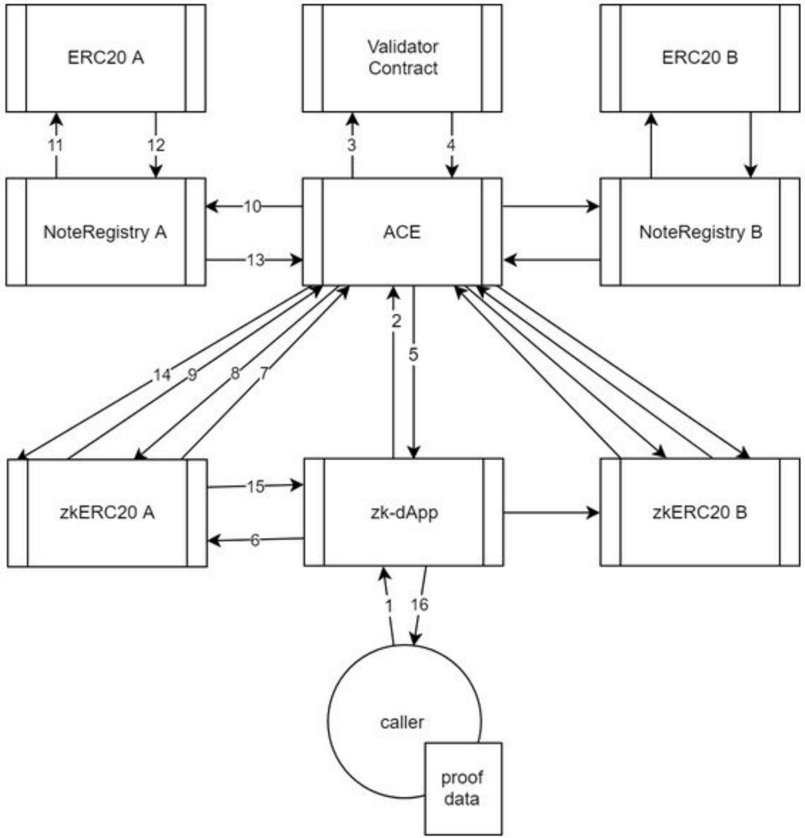
20,000

2,200

aztec

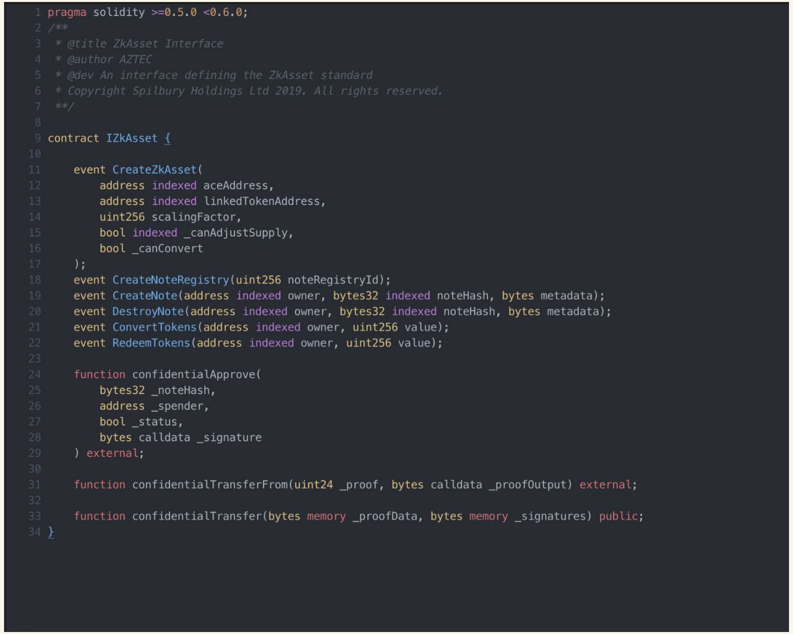
Example: Bilateral Swap

1. User broadcasts the swap proof to a zk-dapp
2. ACE validates proof and returns transfer instructions
3. Zk-dapp broadcasts transfer instructions to zkERC20
4. zkERC20 queries ACE to check instruction validity
5. zkERC20 instructs ACE to update note registry
6. Zk notes created/destroyed, tokens transferred



Creating Confidential Assets

Aztec created a standard interface for interacting with confidential assets that conform to a UTXO based models



```
pragma solidity >=0.5.0 <0.6.0;
/**
 * @title ZkAsset Interface
 * @author AZTEC
 * @dev An interface defining the ZkAsset standard
 * Copyright Spilbury Holdings Ltd 2019. All rights reserved.
 **/

contract IZkAsset {

    event CreateZkAsset(
        address indexed aceAddress,
        address indexed linkedTokenAddress,
        uint256 scalingFactor,
        bool indexed _canAdjustSupply,
        bool _canConvert
    );
    event CreateNoteRegistry(uint256 noteRegistryId);
    event CreateNote(address indexed owner, bytes32 indexed noteHash, bytes metadata);
    event DestroyNote(address indexed owner, bytes32 indexed noteHash, bytes metadata);
    event ConvertTokens(address indexed owner, uint256 value);
    event RedeemTokens(address indexed owner, uint256 value);

    function confidentialApprove(
        bytes32 _noteHash,
        address _spender,
        bool _status,
        bytes calldata _signature
    ) external;

    function confidentialTransferFrom(uint24 _proof, bytes calldata _proofOutput) external;

    function confidentialTransfer(bytes memory _proofData, bytes memory _signatures) public;
}
```

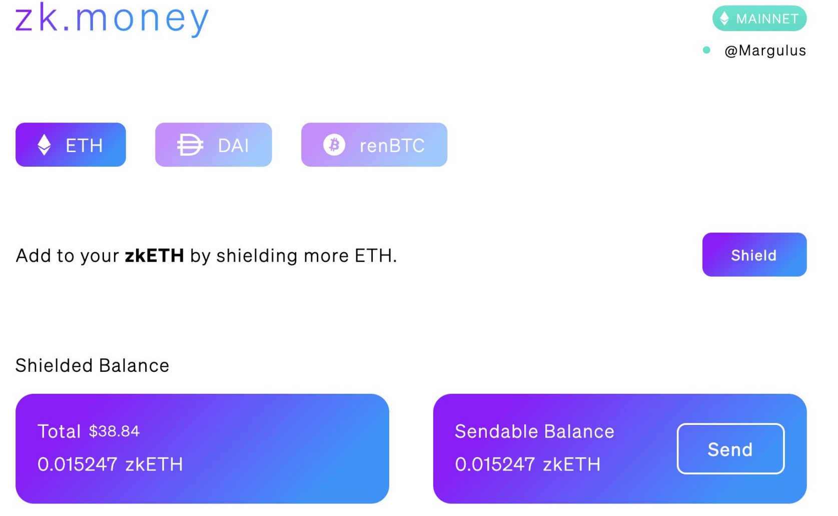
ZK-ZK-Rollup (Zk.Money)

aztec

Zk.money

zk.money ~ private vemno

→ **Private Layer 2 ZK-Rollup** based on range proofs that guarantees private assets and payments/transactions on Ethereum:



zk.money

MAINNET

@Margulus

ETH DAI renBTC

Add to your **zkETH** by shielding more ETH.

Shield

Shielded Balance

Total $38.84
0.015247 zkETH

Sendable Balance
0.015247 zkETH

Send

aztec

Zk.money Construction

Two-circuit construction:

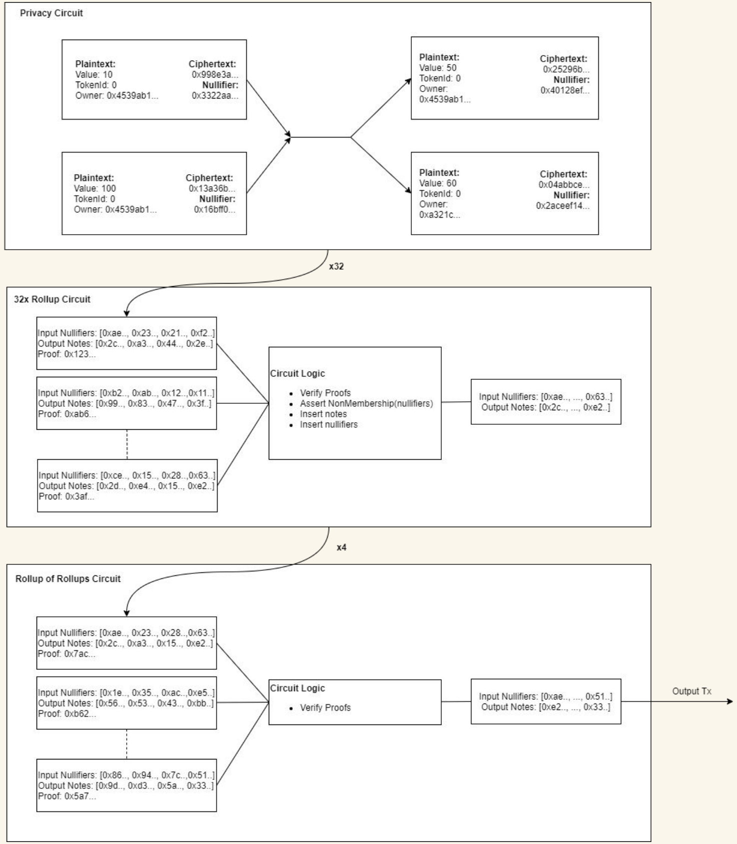
1. **Privacy Circuit** = proves the correctness of a single private transaction (*client-side hardware*)
2. **Rollup circuit** = validates the correctness of a batch of privacy proofs (*rollup provider*)

Uses Recursion: Verifying single **privacy SNARK** inside of a batched **rollup SNARK**

aztec

Zk.money Construction

Zk-Rollup Circuit



Privacy Circuit

Plaintext:
Value: 10
TokenId: 0
Owner: 0x4539ab1...
Ciphertext:
0x998e3a...
Nullifier:
0x3322aa...

Plaintext:
Value: 100
TokenId: 0
Owner: 0x4539ab1...
Ciphertext:
0x13a36b...
Nullifier:
0x16bff0...

Plaintext:
Value: 50
TokenId: 0
Owner:
0x4539ab1...
Ciphertext:
0x25296b...
Nullifier:
0x40128ef...

Plaintext:
Value: 60
TokenId: 0
Owner:
0xa321c...
Ciphertext:
0x04abbce...
Nullifier:
0x2aceef14...

x32

32x Rollup Circuit

Input Nullifiers: [0xae.., 0x23.., 0x21.., 0xf2..]
Output Notes: [0x2c.., 0xa3.., 0x44.., 0x2e..]
Proof: 0x123...

Input Nullifiers: [0xb2.., 0xab.., 0x12..,0x11..]
Output Notes: [0x99.., 0x83.., 0x47.., 0x3f..]
Proof: 0xab6...

Input Nullifiers: [0xce.., 0x15.., 0x28..,0x63..]
Output Notes: [0x2d.., 0xe4.., 0x15.., 0xe2..]
Proof: 0x3af...

Circuit Logic

- Verify Proofs
- Assert NonMembership(nullifiers)
- Insert notes
- Insert nullifiers

Input Nullifiers: [0xae.., ..., 0x63..]
Output Notes: [0x2c.., ..., 0xe2..]

x4

Rollup of Rollups Circuit

Input Nullifiers: [0xae.., 0x23.., 0x28..,0x63..]
Output Notes: [0x2c.., 0xa3.., 0x15.., 0xe2..]
Proof: 0x7ac...

Input Nullifiers: [0x1e.., 0x35.., 0xac..,0xe5..]
Output Notes: [0x56.., 0x53.., 0x43.., 0xbb..]
Proof: 0xb62...

Input Nullifiers: [0x86.., 0x94.., 0x7c..,0x51..]
Output Notes: [0x9d.., 0xd3.., 0x5a.., 0x33..]
Proof: 0x5a7...

Circuit Logic

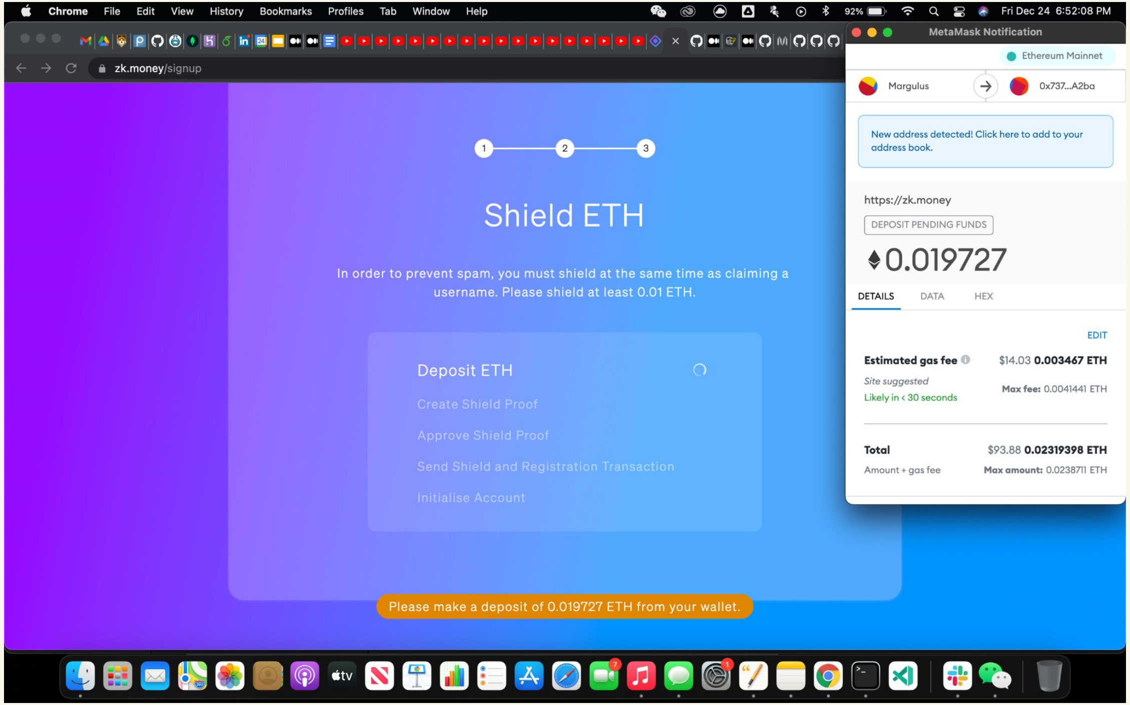
- Verify Proofs

Input Nullifiers: [0xae.., ..., 0x51..]
Output Notes: [0xe2.., ..., 0x33..]

Output Tx

aztec

Zk.money Transaction



Shield ETH

In order to prevent spam, you must shield at the same time as claiming a username. Please shield at least 0.01 ETH.

Deposit ETH

Create Shield Proof

Approve Shield Proof

Send Shield and Registration Transaction

Initialise Account

Please make a deposit of 0.019727 ETH from your wallet.

Ethereum Mainnet

Margulus → 0x737...A2ba

New address detected! Click here to add to your address book.

https://zk.money

DEPOSIT PENDING FUNDS

0.019727

Estimated gas fee $14.03 0.003467 ETH

Site suggested

Likely in < 30 seconds

Max fee: 0.0041441 ETH

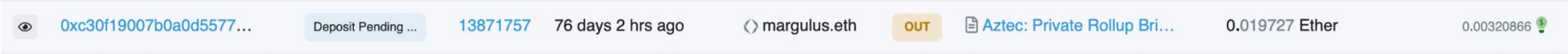
Total $93.88 0.02319398 ETH

Amount + gas fee

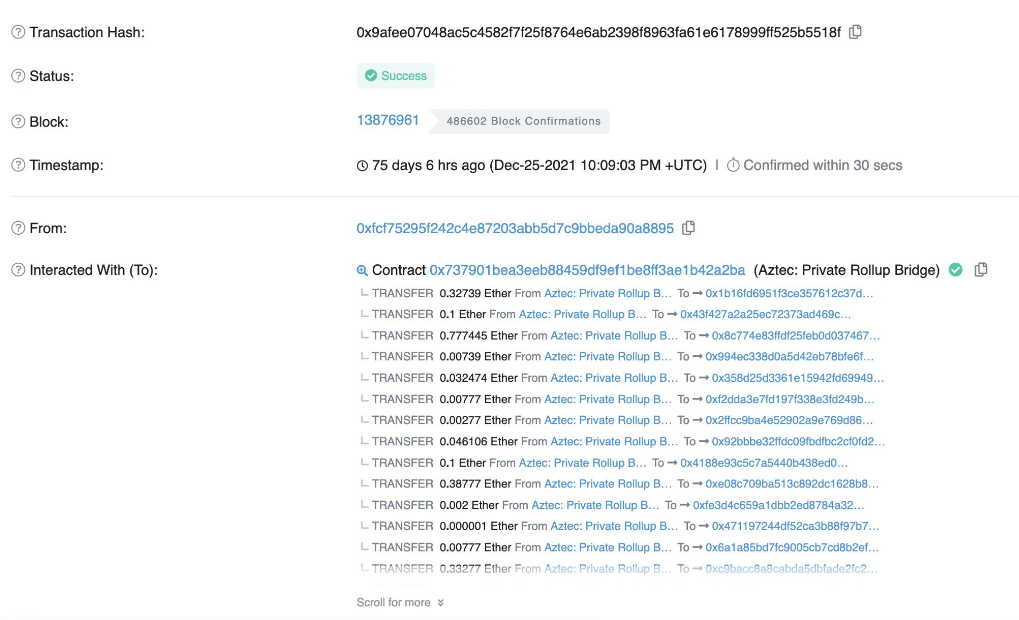
Max amount: 0.0238711 ETH

“Shielded” deposits

Zk.money Transaction



0xc30f19007b0a0d5577…	Deposit Pending ...	13871757	76 days 2 hrs ago	margulus.eth	OUT	Aztec: Private Rollup Bri…	0.019727 Ether	0.00320866



Transaction Hash: 0x9afee07048ac5c4582f7f25f8764e6ab2398f8963fa61e6178999ff525b5518f

Status: Success

Block: 13876961 486602 Block Confirmations

Timestamp: 75 days 6 hrs ago (Dec-25-2021 10:09:03 PM +UTC) | Confirmed within 30 secs

From: 0xfcf75295f242c4e87203abb5d7c9bbeda90a8895

Interacted With (To): Contract 0x737901bea3eeb88459df9ef1be8ff3ae1b42a2ba (Aztec: Private Rollup Bridge)

- TRANSFER 0.32739 Ether From Aztec: Private Rollup B… To → 0x1b16fd6951f3ce357612c37d…
- TRANSFER 0.1 Ether From Aztec: Private Rollup B… To → 0x43f427a2a25ec72373ad469c…
- TRANSFER 0.777445 Ether From Aztec: Private Rollup B… To → 0x8c774e83ffdf25feb0d037467…
- TRANSFER 0.00739 Ether From Aztec: Private Rollup B… To → 0x994ec338d0a5d42eb78bfe6f…
- TRANSFER 0.032474 Ether From Aztec: Private Rollup B… To → 0x358d25d3361e15942fd69949…
- TRANSFER 0.00777 Ether From Aztec: Private Rollup B… To → 0xf2dda3e7fd197f338e3fd249b…
- TRANSFER 0.00277 Ether From Aztec: Private Rollup B… To → 0x2ffcc9ba4e52902a9e769d86…
- TRANSFER 0.046106 Ether From Aztec: Private Rollup B… To → 0x92bbbe32ffdc09fbdfbc2cf0fd2…
- TRANSFER 0.1 Ether From Aztec: Private Rollup B… To → 0x4188e93c5c7a5440b438ed0…
- TRANSFER 0.38777 Ether From Aztec: Private Rollup B… To → 0xe08c709ba513c892dc1628b8…
- TRANSFER 0.002 Ether From Aztec: Private Rollup B… To → 0xfe3d4c659a1dbb2ed8784a32…
- TRANSFER 0.000001 Ether From Aztec: Private Rollup B… To → 0x471197244df52ca3b88f97b7…
- TRANSFER 0.00777 Ether From Aztec: Private Rollup B… To → 0x6a1a85bd7fc9005cb7cd8b2ef…
- TRANSFER 0.33277 Ether From Aztec: Private Rollup B… To → 0xc9bacc8a8cabda5dbfade2fc2…

Aztec Connect Bridge

Aztec Connect Bridge

"It's not practical to make **private versions of DeFi protocols**, instead it's more practical to **privately interact with DeFi protocols** using Layer-2!"

aztec

Aztec Connect Bridge

Aztec Connect: the first private bridge allowing anyone to interact with DeFi contracts on Layer 1, connecting the Aztec L2 to Mainnet.

- **Bridge Contract** = simple 50-100 line interface allowing Aztec's PLONK zkRollup to interact with a given Layer 1 smart contract w/ 100x cost savings

References

https://ethereum.org/en/developers/docs/transactions/

https://tornado.cash/

https://aztec.network/

https://www.youtube.com/watch?v=NyBwdcIMT0M&ab_channel=Bankless

https://www.youtube.com/watch?v=srnkQZxkGOo&ab_channel=ZeroKnowledge

https://www.youtube.com/watch?v=dljPSrwgJZ8&ab_channel=ZeroKnowledge

Thank you!

SCALABLE SYSTEMS
& SOFTWARE
RESEARCH GROUP

https://sss.cse.lehigh.edu/