

ZK SHANGHAI
零知识证明工作坊

WORKSHOP!

应用ZK结构 2

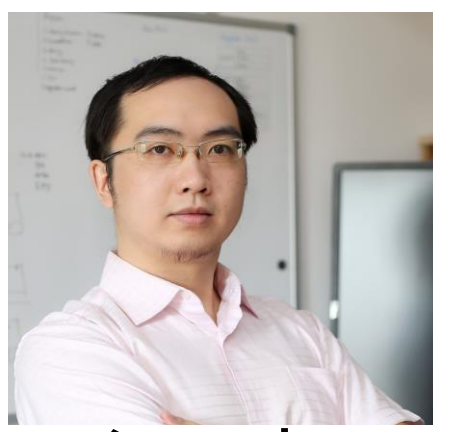
现代零知识密码学

Hosted by **SutuLabs** & **Kepler42B-ZK Planet**

课程资源: zkshanghai.xyz



个人介绍



梁爽

区块链 架构师

上海交大 计算机博士生
(休学创业中)

微信: icerdesign
微博: @wizicer
Github: @wizicer
Twitter: @icerdesign
LinkedIn: www.linkedin.com/in/icerdesign

- 1999年**
 - 正式开始学习写程序
- 2009年**
 - 在新媒传信（飞信）做高性能服务器程序架构及开发
- 2012年**
 - 在Honeywell工业控制部门做PLC、RTU上位机组态软件架构及开发
- 2017年**
 - 接触区块链，并开始创业开发区块链数据库
- 2020年**
 - 入学上海交大攻读博士学位，研究零知识证明数据库
- 2022年**
 - 获Chia全球开发大赛第一名，并开始Pawket钱包的开发
- 2023年**
 - 获得零知识链Mina的项目资助

今日课程内容

- Dark Forest
- ZK数据市场
- ZKML
- ZK应用

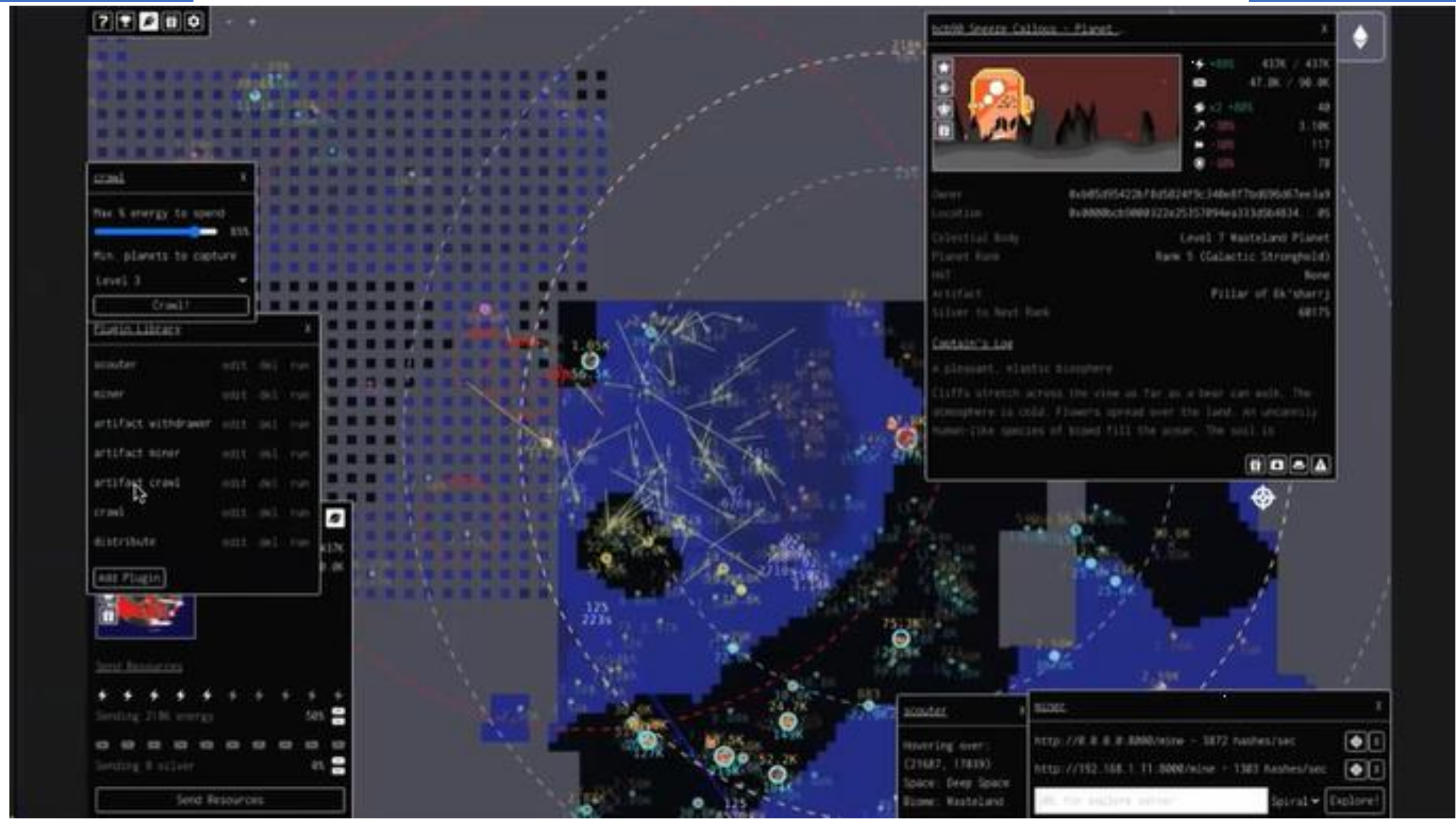
今日课程将回答以下问题

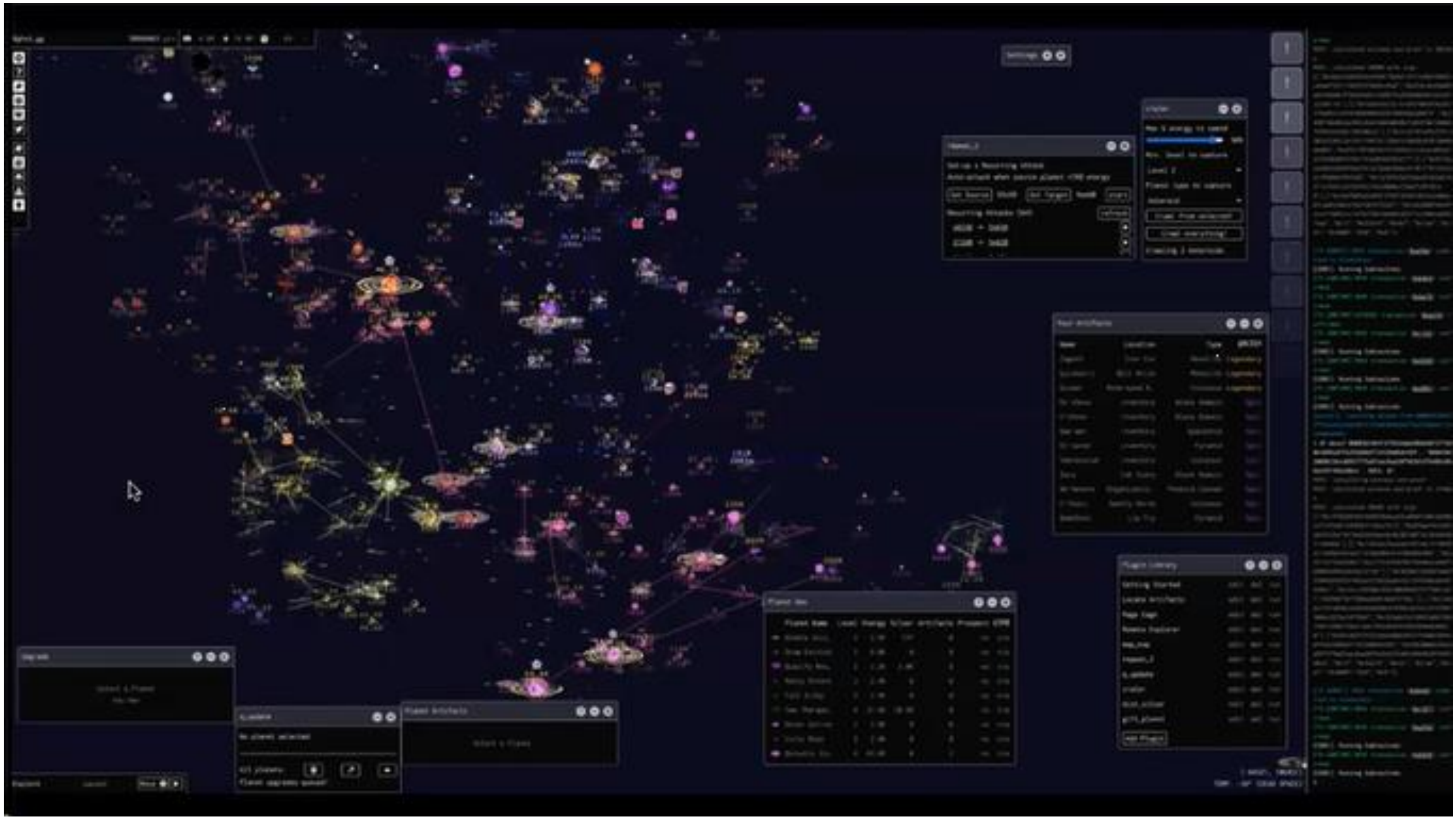
- 非对称信息游戏是如何实现的?
- 如何在交易前验证数据的有效性?
- 零知识证明如何与机器学习结合?

Dark Forest

Dark Forest是一个使用zkSNARKs构建在以太坊上的去中心化MMORTS游戏。







完全信息 vs 不完全信息



完全信息游戏



○		
×	×	○
○		×

信息非对称



信息非对称



zkSNARKs

- 我正在从一副牌中抽一张牌，并将其加入到我的手牌中。
- 我不会向你展示我的牌，但我可以证明我是从一副经过适当洗牌的牌中随机抽取的。

ZKP: Fog of War

- 我正在将我的骑士从秘密位置A移动到秘密位置B。
- 我不会告诉你A和B在哪里，但我可以证明它们是被正确移动的。

Dark Forest 游戏构造

(2019 – v0.3)

游戏构造

- 每个玩家都在一个大的二维网格上。
- 对于位置 (x,y) , $\text{hash}(x,y)$ 是该位置的公共地址。
 - 这些坐标本身是该位置的私有地址。
- 当 $\text{hash}(x,y) < \text{DIFFICULTY_THRESHOLD}$ 时
 - 位置 (x,y) 上有适合居住的星球。
 - 所有其他空间都是空的。
- 由玩家控制的单位存在于玩家拥有的星球上。

游戏构造：状态

- 公共状态
 - 拥有哪些公共地址，谁拥有它们以及它们的人口数量
- 私有状态
 - 玩家行星的私有地址 (x, y)
 - 通过计算获得信息

玩家动作：初始化

function initializePlayer(uint planetId, uint claimedDist, Proof zkProof)

在坐标上使用planetId初始化玩家。

还检查这些坐标是否在距离原点的某个声明距离内。

玩家动作：初始化

function initializePlayer(uint planetId, uint claimedDist, Proof zkProof)

zkProof: 我知道某坐标 (x, y) , 使得

- $\text{hash}(x, y) = \text{planetId}$
- $x^2 + y^2 < \text{claimedDist}^2$

玩家动作：初始化

```
template Main() {
    signal private input x;
    signal private input y;

    signal input r;

    signal output pub;

    /* check abs(x), abs(y), abs(r) < 2^32 */
    component rp = MultiRangeProof(2, 40, 2 ** 32);
    rp.in[0] <== x;
    rp.in[1] <== y;

    /* check x^2 + y^2 < r^2 */
    component comp = LessThan(32);
    signal xSq;
    signal ySq;
    signal rSq;
    xSq <== x * x;
    ySq <== y * y;
    rSq <== r * r;
    comp.in[0] <== xSq + ySq
    comp.in[1] <== rSq
    comp.out == 1;

    /* check MiMCSponge(x,y) = pub */
    /*
       220 = 2 * ceil(log_5 p), as specified by mimc paper, where
       p = 21888242871839275222246405745257275088548364400416034343698204186575808495617
    */
    component mimc = MiMCSponge(2, 220, 1);

    mimc.ins[0] <== x;
    mimc.ins[1] <== y;
    mimc.k <== 0;

    pub <== mimc.outs[0];
}
```

范围证明

Range Proof

```
// input: n field elements, whose abs are claimed to be less than max_abs_value
// output: none
template MultiRangeProof(n, bits, max_abs_value) {
    signal input in[n];
    component rangeProofs[n];

    for (var i = 0; i < n; i++) {
        rangeProofs[i] = RangeProof(bits, max_abs_value);
        rangeProofs[i].in <== in[i];
    }
}
```

```
// NB: RangeProof is inclusive.
// input: field element, whose abs is claimed to be less than max_abs_value
// output: none
// we also want something like  $4 * (\text{abs}(\text{in}) + \text{max\_abs\_value}) < 2 ** \text{bits}$ 
// and  $\text{bits} \ll 256$ 
template RangeProof(bits, max_abs_value) {
    signal input in;

    component lowerBound = LessThan(bits);
    component upperBound = LessThan(bits);

    lowerBound.in[0] <== max_abs_value + in;
    lowerBound.in[1] <== 0;
    lowerBound.out == 0

    upperBound.in[0] <== 2 * max_abs_value;
    upperBound.in[1] <== max_abs_value + in;
    upperBound.out == 0
}
```


玩家动作：移动

function move(uint fromPlanetId, uint toPlanetId, uint worldRadius, uint maxDist)

从 fromPlanetId 移动兵力到 toPlanetId:

- 检查这两个星球是否“在边界内”。
- 支付一些费用，具体取决于两个星球之间的最大距离。

玩家动作：移动

function move(uint fromPlanetId, uint toPlanetId, uint worldRadius, uint maxDist)

zkProof: 我知道某坐标 $(x1, y1)$ 和 $(x2, y2)$ 使得:

- $\text{hash}(x1, y1) = \text{fromPlanetId}$
- $\text{hash}(x2, y2) = \text{toPlanetId}$
- $x2^2 + y2^2 < \text{worldRadius}^2$
- $(x1-x2)^2 + (y1-y2)^2 < \text{distMax}^2$

玩家动作：移动

```
template Main() {  
    signal private input x1;  
    signal private input y1;  
    signal private input x2;  
    signal private input y2;  
  
    signal input r;  
    signal input distMax;  
  
    signal output pub1;  
    signal output pub2;
```

```
/* check abs(x1), abs(y1), abs(x2), abs(y2) < 2 ** 32 */  
component rp = MultiRangeProof(4, 40, 2 ** 32);  
rp.in[0] <== x1;  
rp.in[1] <== y1;  
rp.in[2] <== x2;  
rp.in[3] <== y2;  
  
/* check x2^2 + y2^2 < r^2 */  
  
component comp2 = LessThan(32);  
signal x2Sq;  
signal y2Sq;  
signal rSq;  
x2Sq <== x2 * x2;  
y2Sq <== y2 * y2;  
rSq <== r * r;  
comp2.in[0] <== x2Sq + y2Sq  
comp2.in[1] <== rSq  
comp2.out == 1;
```

玩家动作：移动

```
/* check (x1-x2)^2 + (y1-y2)^2 <= distMax^2 */

signal diffX;
diffX <== x1 - x2;
signal diffY;
diffY <== y1 - y2;

component ltDist = LessThan(32);
signal firstDistSquare;
signal secondDistSquare
firstDistSquare <== diffX * diffX;
secondDistSquare <== diffY * diffY;
ltDist.in[0] <== firstDistSquare + secondDistSquare;
ltDist.in[1] <== distMax * distMax + 1;
ltDist.out == 1;
```

```
/* check MiMCSponge(x1,y1) = pub1, MiMCSponge(x2,y2) = pub2 */
/*
    220 = 2 * ceil(log_5 p), as specified by mimc paper, where
    p = 21888242871839275222246405745257275088548364400416034343698204186575808495617
*/
component mimc1 = MiMCSponge(2, 220, 1);
component mimc2 = MiMCSponge(2, 220, 1);

mimc1.ins[0] <== x1;
mimc1.ins[1] <== y1;
mimc1.k <== 0;
mimc2.ins[0] <== x2;
mimc2.ins[1] <== y2;
mimc2.k <== 0;

pub1 <== mimc1.outs[0];
pub2 <== mimc2.outs[0];
```

无需许可的互操作性

- **Dark Forest**是一个以太坊智能合约，任何人（玩家、机器人或智能合约）都可以编程与之交互。

CLIENT-SIDE PLUGINS

v0.8.0 **Towards Center** **Utility**

when you choose one planet .), you can make it towards center.

v0.8.0 **Scoring Planets** **Utility**

Planet	Score
Planet 1	100
Planet 2	200
Planet 3	300
Planet 4	400
Planet 5	500
Planet 6	600
Planet 7	700
Planet 8	800
Planet 9	900
Planet 10	1000

See the top scoring planets within your vision

v0.8.0 **Tiny Leaderboard** **Utility**

11:54:50

10	Player123	11005451
11	Player234	10001456
12	Player345	10011040
13	Player456	10100172
14	Player567	10011178
15	Player678	10011425
16	Player789	10011425
17	Player890	10011425
18	Player901	10011425
19	Player012	10011425
20	Player123	10011425

Shows a tiny leaderboard with timer

v0.8.0 **Remote Snarker** **Productivity**

Speed up shark computation using servers.

v0.8.0 **Heatmap** **Visual**

Highlight territories around you!

gakonst / dark-forest

THIRD-PARTY CLIENT IMPLEMENTATIONS

Code Issues Pull requests

master 3 branches 0 tags

Go to file Add file Code

gakonst Merge pull request #113 from mattsse/misc 117 comments 12 days ago

- github/workflows ci: do not double run tests 14 days ago
- abis fix: update abis 17 days ago
- contracts feat: add smart contract account scaffolding 2 months ago
- crates rustfmt 12 days ago
- scripts feat: add abis and codegen for bindings 2 months ago
- test-vectors fix: off by one error in threshold calc + more tests cases 2 months ago
- .gitignore feat: add abis and codegen for bindings 2 months ago
- Cargo.lock chore: update deps 14 days ago
- Cargo.toml simplify members 12 days ago
- README.md docs: add readme for cli 14 days ago

README.md

dark-forest.rs

Terminal UI implementation and types for the Dark Forest game

Tests failing

About

[WIP] Rust implementation of the Dark Forest game client

rust cryptography ethereum

dark-forest

Readme

Releases

No releases published

Packages

No packages published

Contributors


- gakonst Georgios Konstantopoulos
- kobigurk Kobi Durkan
- mattsse Matthias Seitz

Languages

Rust 91.7% Solidity 7.0%

REMOTE MINERS

Step 2: Connect to your Remote Explorer server



theonlyjohnny-darkforest

Once you see the green "Deployed" badge, you're ready to connect! Click the Copy icon next to the Visit button.

In your game UI, find the Remote Explorer Server URL input box. Paste the URL, you just copied, and add /mine to the end of it.

Example: <https://theonlyjohnny-darkforest-rs.zet.app/mine>



DF EXPLORER

Powered by ZK Workshop

A Dark Forest Cuda Explorer

Contributor: @yungandeng

❤️ @darkforest_eth ❤️

Build Waterdrop  MarrowDAC

 **Long Rock Labs** @LongRockLabs - Aug 20

Today we released a new explorer for @darkforest_eth . This miner can use either the CPU or GPU to uncover the Dark Forest. Comes with a nice performance gain over the standard miner.
longrocklabs.com/articles/the-e...

AUTOMATIONS AND BOT



nick.eth @nicksdjohnson · Aug 16

Weekend project: A [@darkforest_eth](#) AI.

So far it attacks planets, distributes silver, upgrades planets, and prospects and finds artefacts.

Still on the todo list is distributing energy. Figuring out how to allocate and use artefacts will be tricky though.

```

Rose Sleepy (L4R0)      Sending 27000 silver to Oatmeal Breathe (L5R0)
Six Hiss (L4R0)         Sending 14250 silver to Oatmeal Breathe (L5R0)
Fog Resell (L4R0)       Sending 25471 silver to Onerous Jaded (L5R2)
Jail Amuck (L4R0)       5% to prospect
Changeable Ludicrous   Capturing Redo Convince (L5R0) with 28125
(L4R0)
Neasily Depend (L4R0)   12% to attack Redo Convince (L5R0)
Daffy Sneeze (L4R0)    10% energy to send silver to Fold Rapid (L3R0)
Otter Button (L4R0)    30% energy to send silver to Fold Rapid (L3R0)
Harbor Country (L4R1)   89% to attack Redo Convince (L5R0)
Bridge Convince (L4R0)  34% to attack Redo Convince (L5R0)
Lose Jail (L4R0)        Capturing Rhythic Snap (L4R0) with 34771
Childhood Partake (L4R2) 91% to attack Rhythic Snap (L4R0)

```

9 12 85 Tip

NEW CORE GAMEPLAY FEATURES

Project Sophon

Twitter GitHub

Broadcast Market

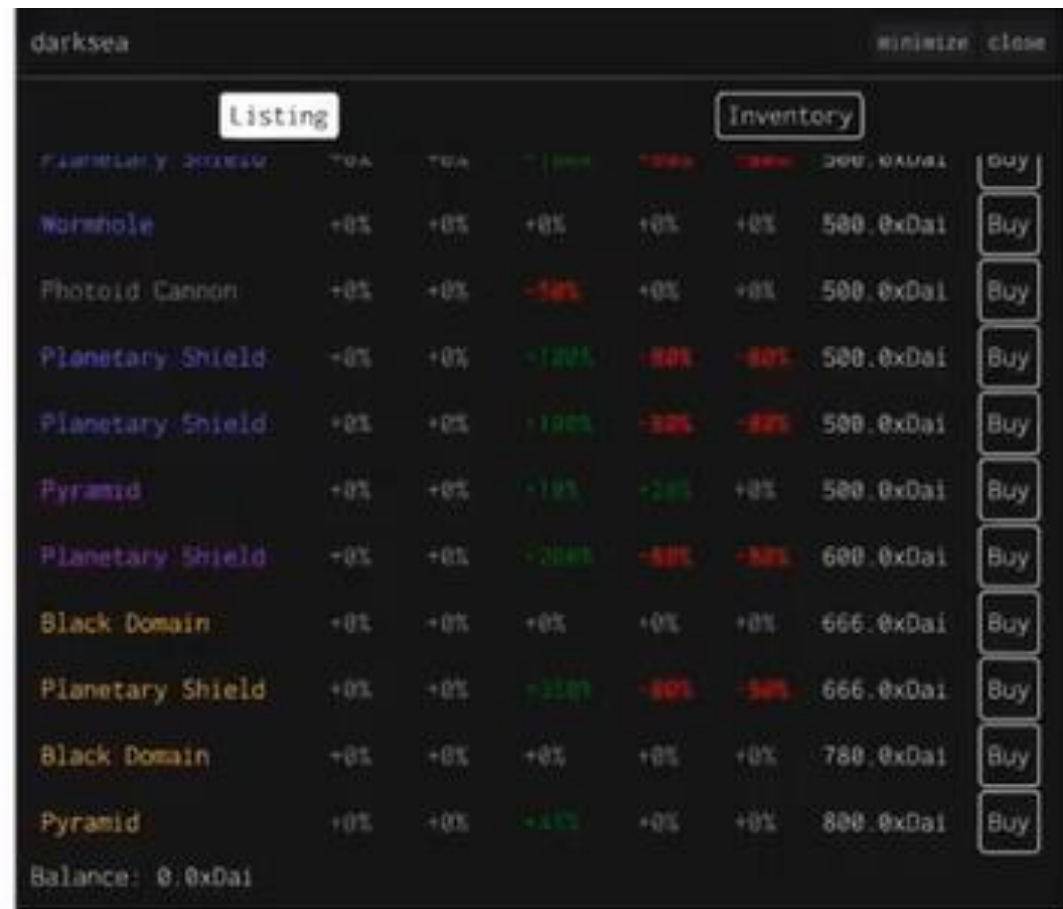
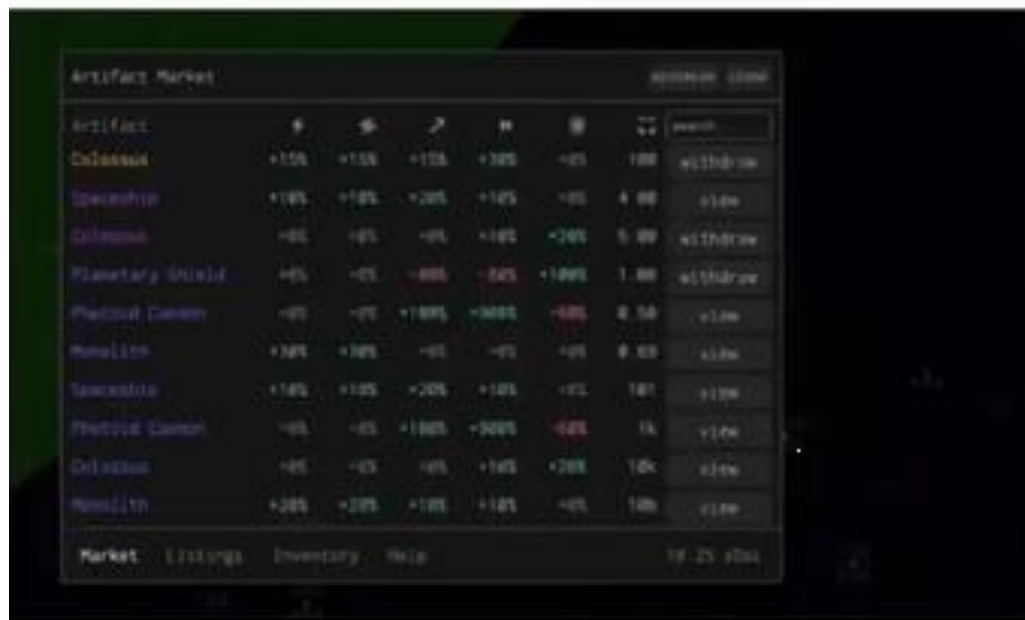
Forget refilling across the xDai Bridge. Play to Earn with Sophon.

Each player only gets one Planet Broadcast per day—don't waste it, you could be getting paid to play!

Just create a new plugin containing:

```
1 export { default } from "https://play2earn.projectsophon
```

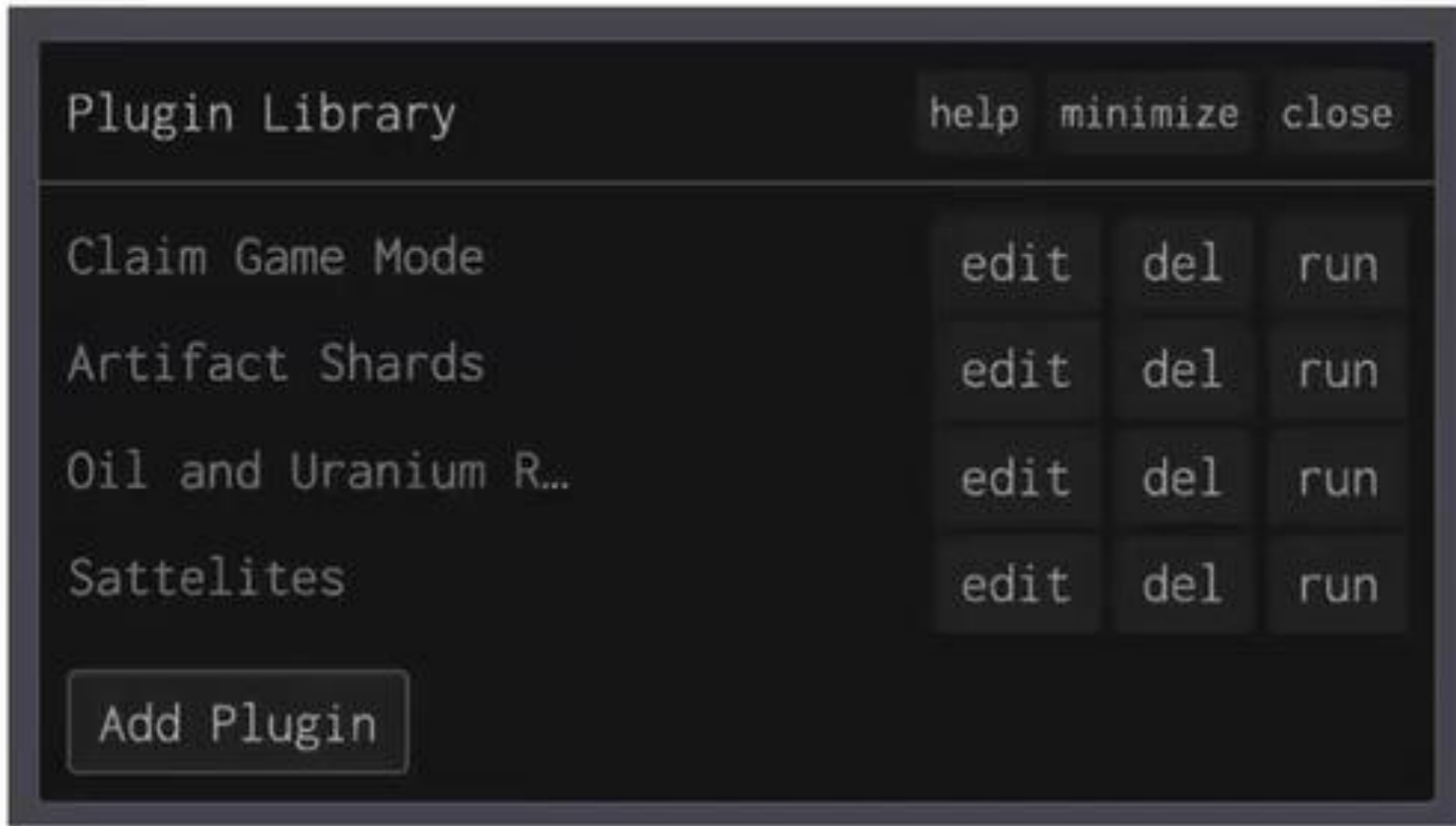




The Astral Colossus - Inventory

The Astral Colossus minimize close

0. The Astral Colossus (Rank 34: 76528735)	76528735
1. @TheVelorum (Rank 189: 585491)	17700133
2. @davidryan59 (Rank 100: 5275298)	13898245
3. @scotato (Rank 228: 268000)	7979795
4. @tofu4956 (Rank 91: 8120164)	3747725
5. @jojazzas (Rank 117: 2958264)	3477702
6. @orden_gg (Rank 1: 777777777)	3461902
7. @MJ659600 (Rank 70: 30316438)	3347518
8. @xJunshen (Rank 68: 34673145)	3165856
9. 0xcF0cc... (Rank 221: 281000)	3081000
10. @CryptoPriest6 (Rank 61: 48465292)	2822434
11. 0x2616B... (Rank 10: 154472935)	2734149
12. @thelegendoftin1 (Rank 184: 592176)	2073140
13. @vjotav (Rank 72: 26777836)	1650000
14. ... (Rank ...: ...)	...
Contribute	7979795
Leaderboard	
Help	



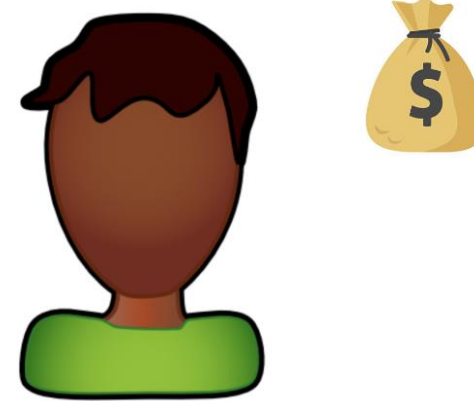
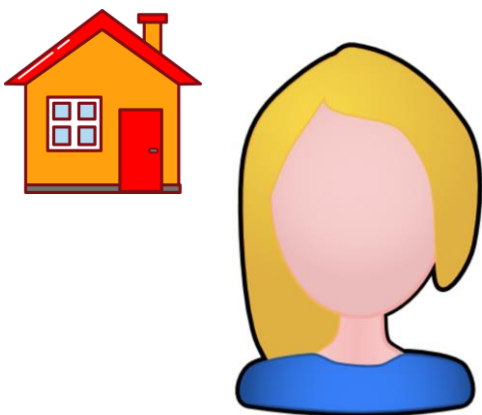
为什么需要：无许可的互操作性？

- 无许可的互操作性意味着玩家可以构建游戏体验，而不仅仅是核心开发人员。
- 这意味着其他游戏可以连接到这个游戏，共享资产、身份系统、机制等等。
- 不需要“支持团队”或“业务发展团队”。
- 市场、玩家公司、游戏内嵌套游戏、新资源和机制、自动化、备用客户端、可互操作的资产和统计数据、共享声誉系统等都可以实现。

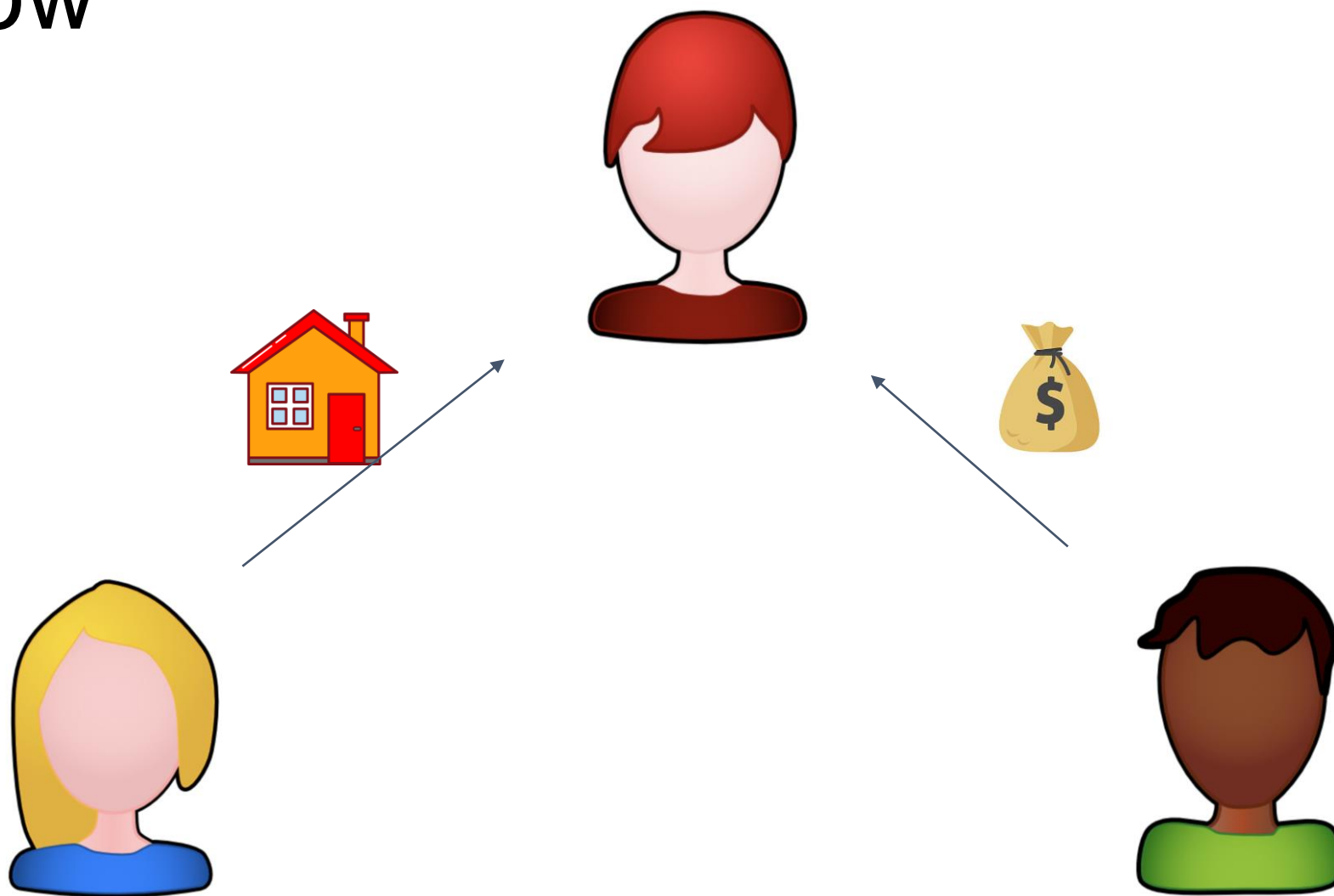
零知识数据市场

ZK Data Marketplace

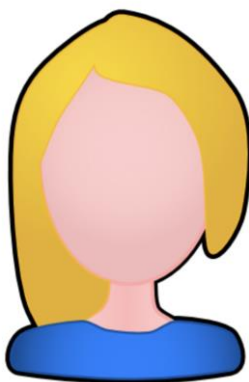
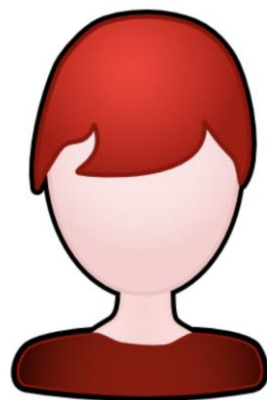
Escrow



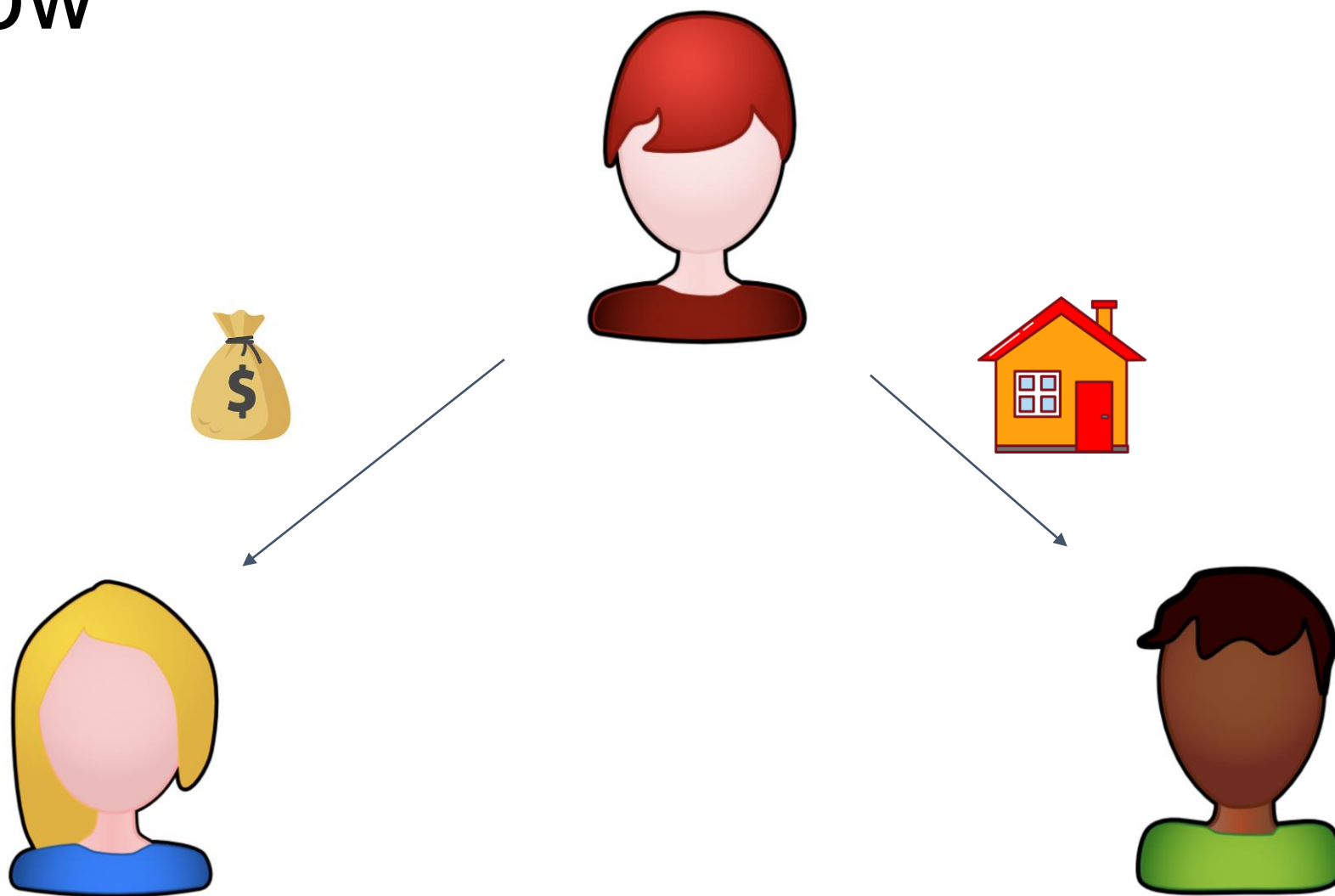
Escrow




Escrow



Escrow

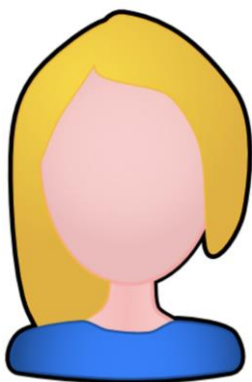


Escrow

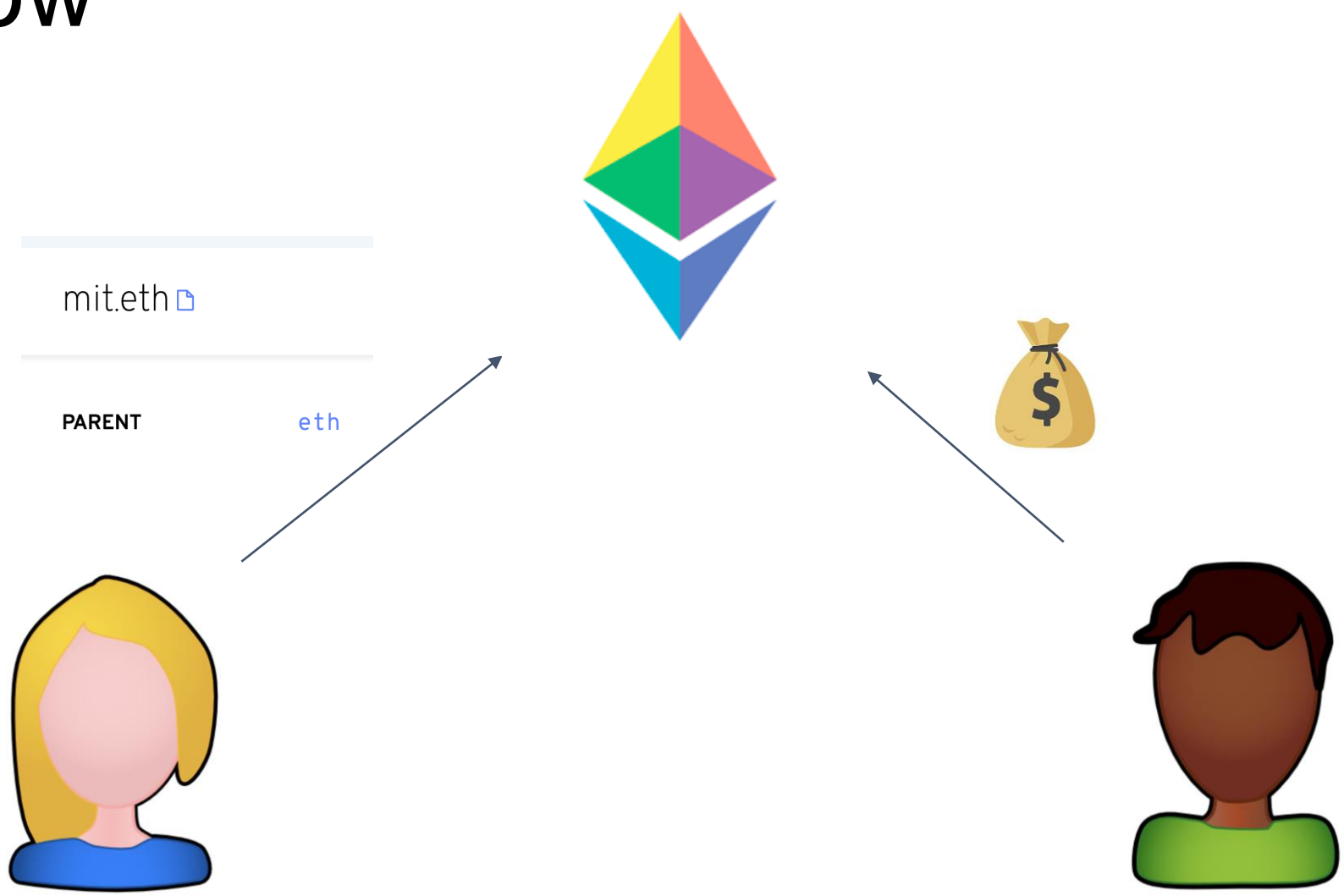
mit.eth 

PARENT

eth



Escrow

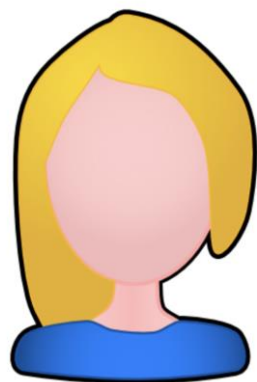


Escrow

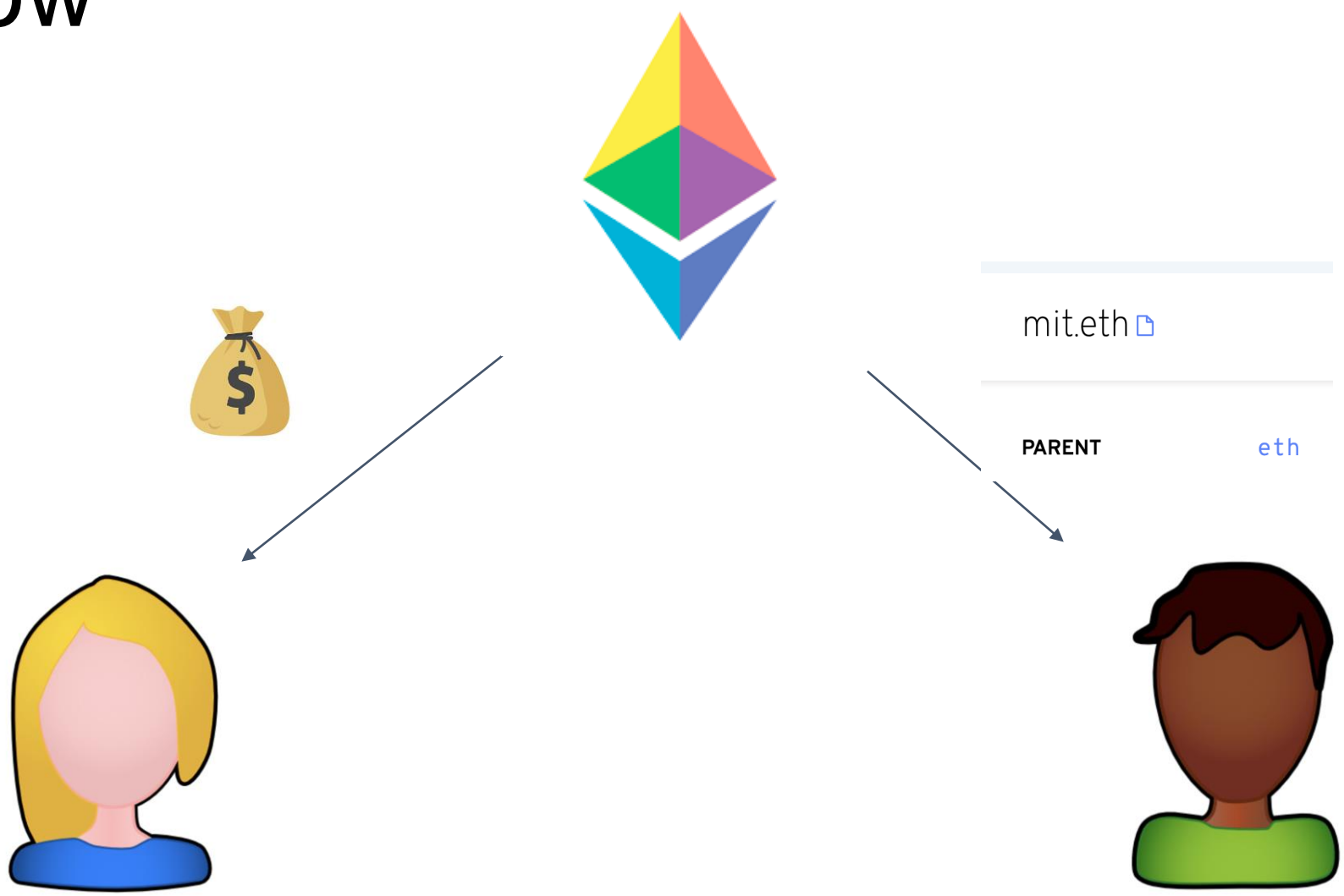
mit.eth 

PARENT

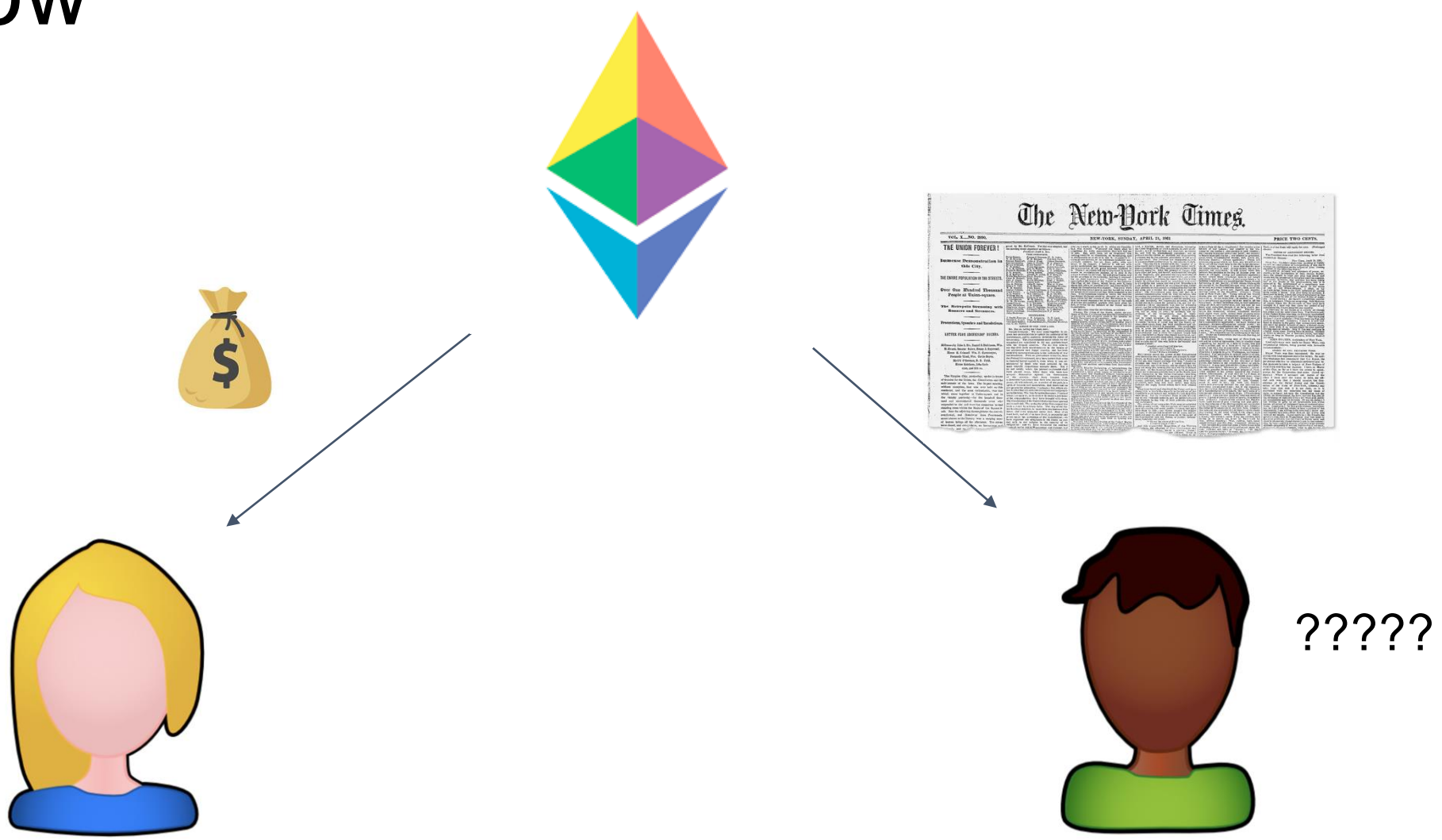
eth



Escrow



Escrow



链上交易市场

On-Chain Marketplace

- 简单示例：Bob想从Alice那里购买0x98b3f001的原像（即哈希的原文）。

链上交易市场

On-Chain Marketplace

Escrow合约检查买方和卖方是否已经满足条件：

- Bob已经将\$\$锁定在了Escrow合约中。
- Alice已经发布了买方想要的的数据。

问题：

- 合约能够检查卖方的条件的唯一方法是读取卖方的数据！

解决方案

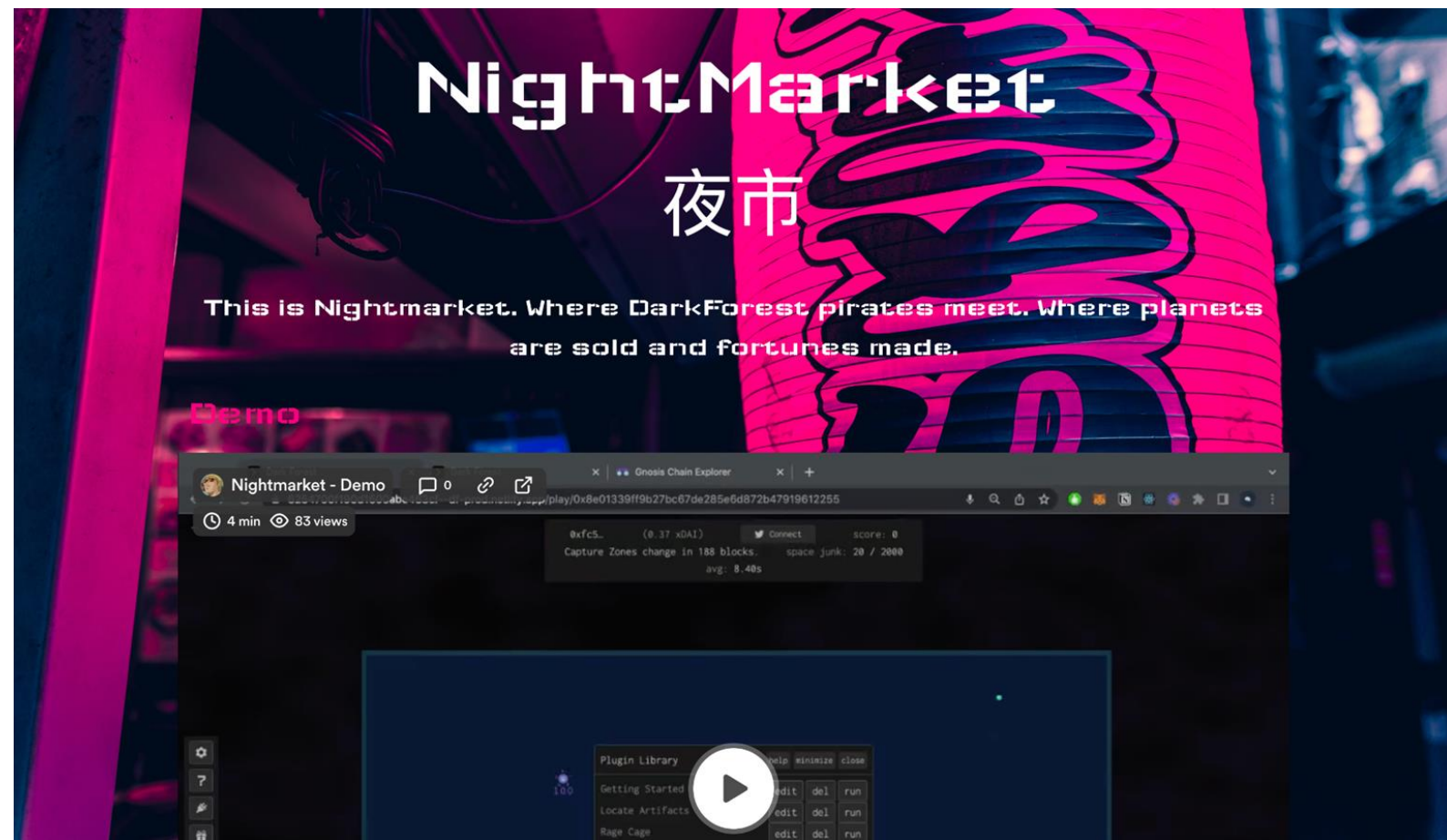
- Alice使用买方的公钥加密数据并加以发布。
- 同时，Alice还需要发布zkSNARK证据，证明该密文是使用Bob的公钥正确加密的数据。
- 只有当zkSNARK证据被验证后，智能合约才会向Alice释放资金。

<https://github.com/nulven/EthDataMarketplace> - Nick Ulven
(2021)

解决方案

- 公共输入：
 - 买方公钥 pk
 - 密文 c
 - 承诺 h
- 私密输入：
 - 隐私数据 s
- 证明：
 - $\text{Hash}(s) = h$
 - $\text{Enc}(s, pk) = c$

Nightmarket



Nightmarket

<https://blog.zkga.me/nightmarket> - 0xSage, xyz_pierre (2022)

<https://nightmart.xyz/>

Nightmarket

Key constraints are as follows:



Seller

Constraint	Publicly Verifiable Value
<code>hash (PLANET_X/Y, PLANETHASH_KEY)</code>	A valid planet hash
<code>perlin (PLANET_X/Y, BIOMEBASE_KEY)</code>	The correct biomebase
<code>poseidon_encode_check (CIPHERTEXT, PLANET_ADDR, KEY)</code>	Valid ciphertext using KEY
<code>hash (KEY)</code>	KEY won't change later
<code>SELLER_ADDR * SELLER_ADDR</code>	Watermark proof to Seller

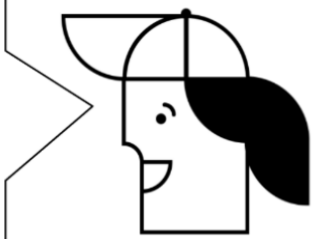
• 详细来说，需要确保：

- `hash(PLANET_X, PLANET_Y, PLANETHASH_KEY)`：卖家证明他们知道一个星球坐标，买家可以在链上验证。
- `poseidon_encode_check(CIPHERTEXT, PLANET_X, PLANET_Y, KEY)`：卖家承诺已经正确地使用对称密钥在星球坐标上加密了 CIPHERTEXT。卖家会将密文发布到链上以便以后解密。
- `hash(KEY)`：卖家单独承诺用于上述步骤的秘密 KEY。值得注意的是，实际上出售的物品不是原始的星球坐标，而是用于对坐标进行对称加密的秘密 KEY。拥有该 KEY 意味着任何人都可以随后解密密文并检索原始的星球坐标。

Nightmarket

- 然后，多个买家可以在一个订单簿上挑选下单，具体如右图：
- 买家先确认订单的有效性
- 买家向Escrow合约存入保证金。
- 买家通过离线ECDH构建 SHARED_KEY密钥。
- 并将SHARED_KEY密钥的哈希值发布到链上。

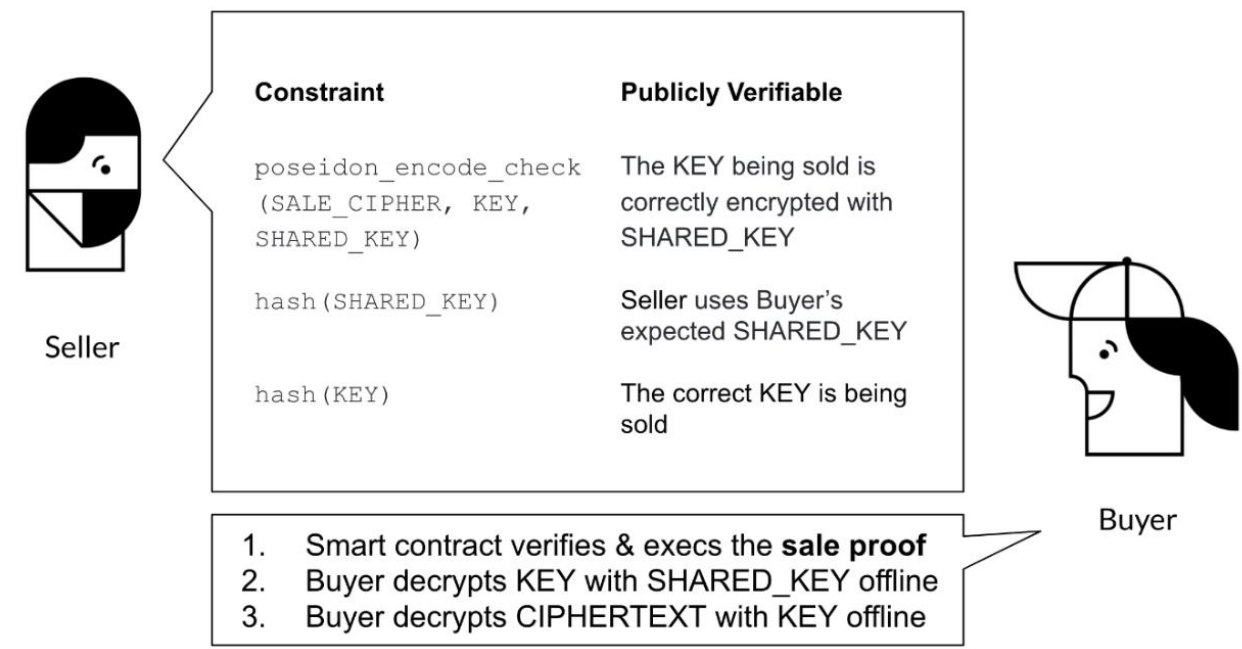
1. (offline) I verify that
 - a. Seller's LIST proof is valid
 - b. Publicly committed values are valid
2. I order listing(s) by depositing my money in the escrow smart contract
3. I preemptively declare how the final sale should be encrypted
 - a. `ecdh(MY PRIVKEY, SELLER PUBKEY)`
 - b. `hash(SHARED_KEY) -> publish`



Buyer

Nightmarket

- 卖家通过销售电路执行买家的购买请求，证明如右图：
- 卖家离线执行ECDH生成与买家相同的SHARED_KEY。
- 卖家用SHARED_KEY对原始KEY进行加密，并将其加密为SALE_CIPHER广播到链上。
- 买家使用SHARED_KEY解密此SALE_CIPHER，以获取KEY。
- 买家使用KEY解密CIPHERTEXT，并检索原始的星球坐标。



还有什么可以卖的？

- 一个比特币、以太坊、SSH、DKIM私钥
- 一个智能合约漏洞（或更普遍地说是程序漏洞）
- 一张鸟的图片
- ...

ZKML

神经网络是一个函数。

把它放进 zkSNARK 中!

ZKML

zkonduit / ezkl Public

Watch 8 Fork 6 Star 135

Code Issues 4 Pull requests 2 Discussions Actions Projects Security Insights

main 8 branches 0 tags

Go to file Add file Code

alexander-camuto chore: update dependencies (#101)	8caa4bf 12 hours ago	170 commits
.github/workflows	chore: update dependencies (#101)	12 hours ago
benches	chore: error bubbling (#93)	2 weeks ago
examples	chore: error bubbling (#93)	2 weeks ago
src	chore: update dependencies (#101)	12 hours ago
tests	chore: update dependencies (#101)	12 hours ago
.gitignore	add: leaky_relu non-linearity (#72)	last month
.gitmodules	refactor: move onnx examples to external repo (#69)	2 months ago
Cargo.lock	chore: update dependencies (#101)	12 hours ago
Cargo.toml	chore: update dependencies (#101)	12 hours ago
LICENSE	Create LICENSE	5 months ago
README.md	fix: kzg verification with EVM (#83)	last month
data.sh	add(mnist data) (#53)	3 months ago

README.md

EZKL

Rust passing

About

No description, website, or topics provided.

- Readme
- Apache-2.0 license
- 135 stars
- 8 watching
- 6 forks

Releases

No releases published

Packages

No packages published

Contributors 5

Languages

- Rust 99.9%
- Shell 0.1%

ZKML

<https://github.com/zkonduit/ezkl> - Jason Morton (2022)

<https://arxiv.org/pdf/2210.08674.pdf> - Daniel Kang (2022)

示例应用程序1：可验证计算

未来，假设使用LLM来审判一个案件（或提供专家建议等）。

- 谁来运行LLM？
- 他们是否正确运行？
- 如果LLM的模型参数是私有的呢？



示例应用程序1：可验证计算

在任何案例开始之前：OpenAI 承诺模型 (model commit) = C

然后，在任何案例的推断中，证明...

示例应用程序1：可验证计算

- 公共输入：
 - 输入 x
 - 声明的输出 y
 - 模型承诺 c
- 隐私输入：
 - 模型 M
- 证明：
 - $M(x) = y$
 - $\text{commit}(M) = c$

示例应用2：零知识生物识别（ZK Biometrics）

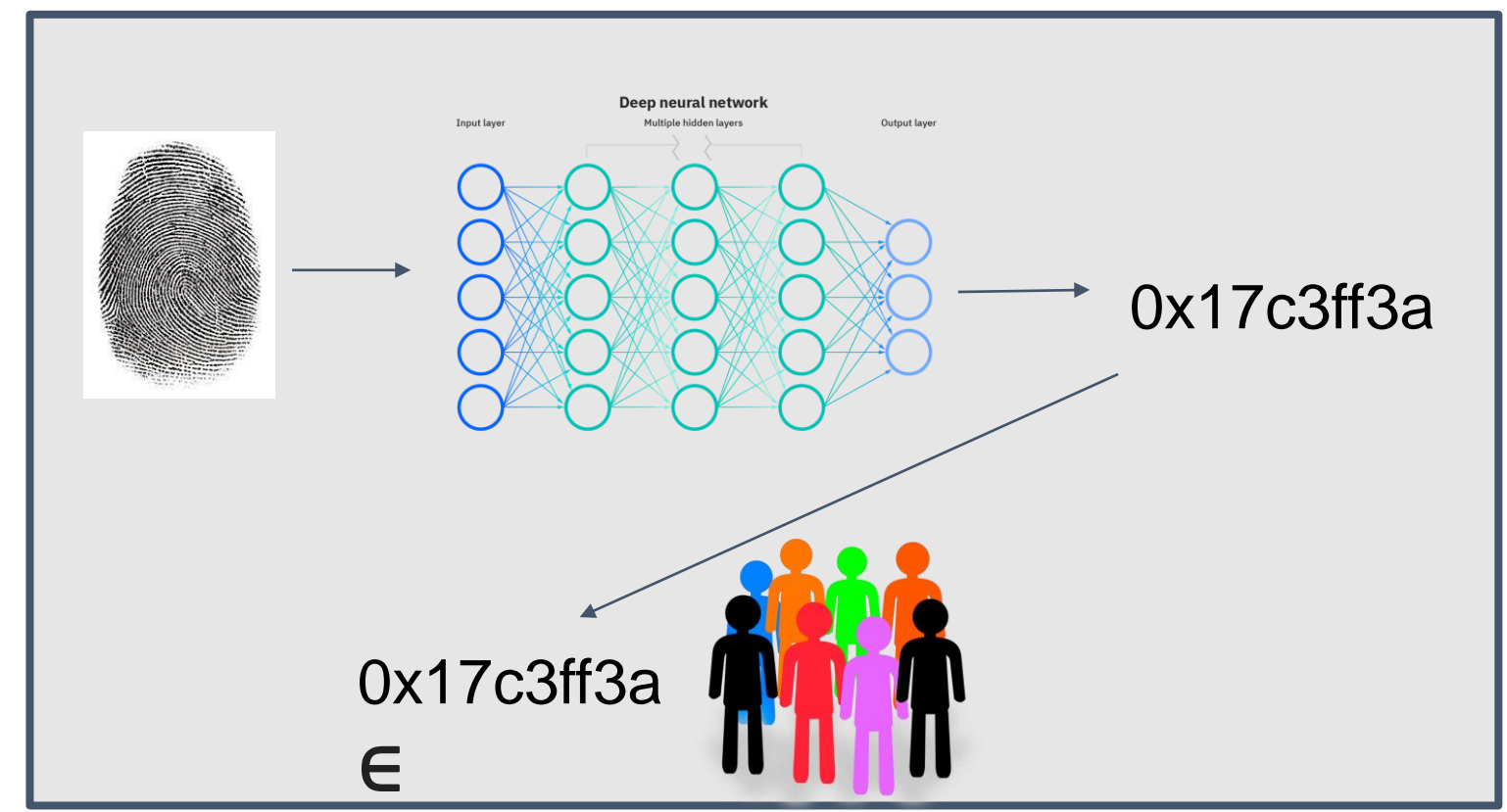
问题：生物识别认证只有在大型机构存储（或可以访问）我们的生物识别数据的情况下才可能实现吗？



Transportation Security
Administration

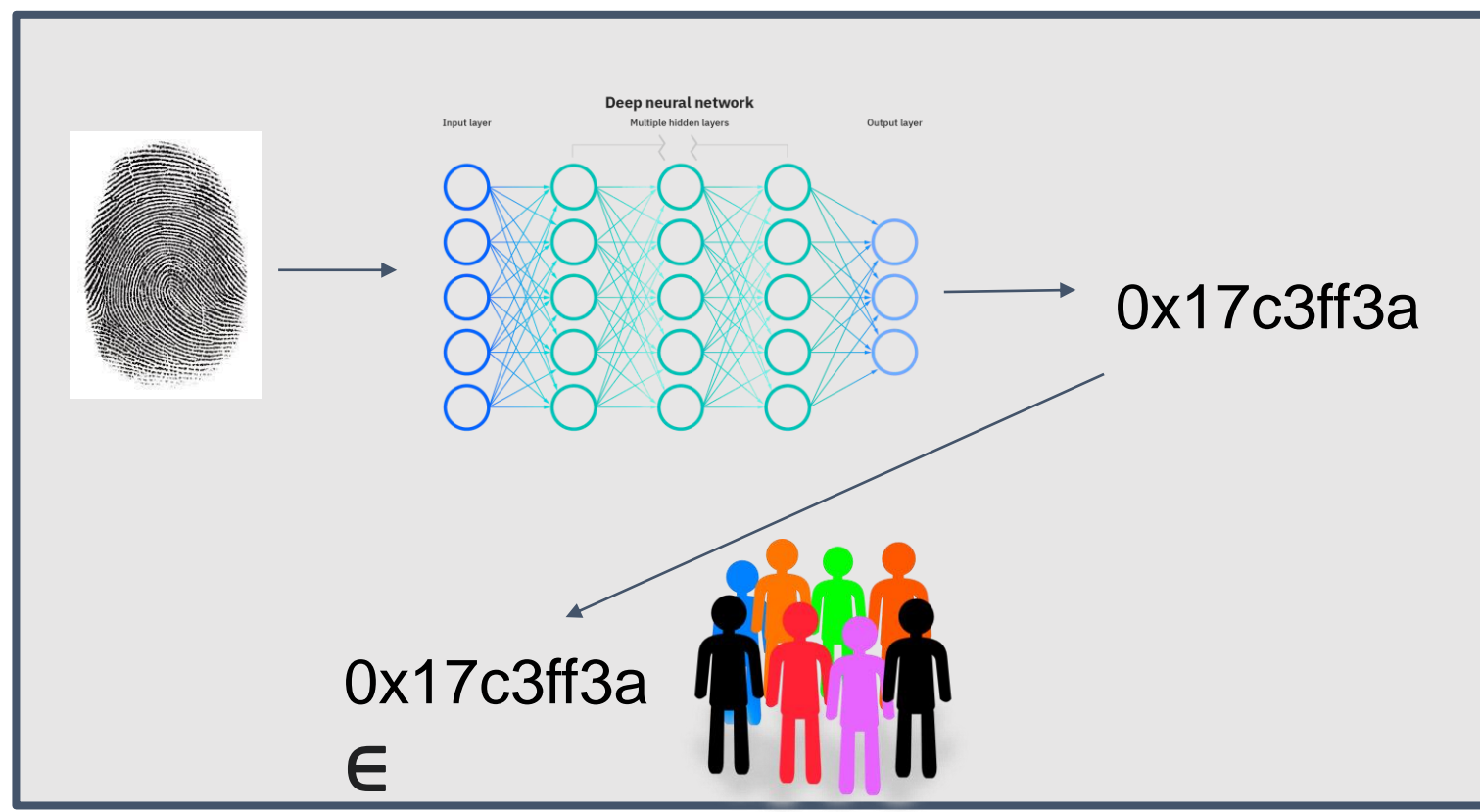
示例应用2：零知识生物识别（ZK Biometrics）

如果已知生物识别哈希的数据库，则可以在不透露我们的生物识别数据或具体ID的情况下证明授权。

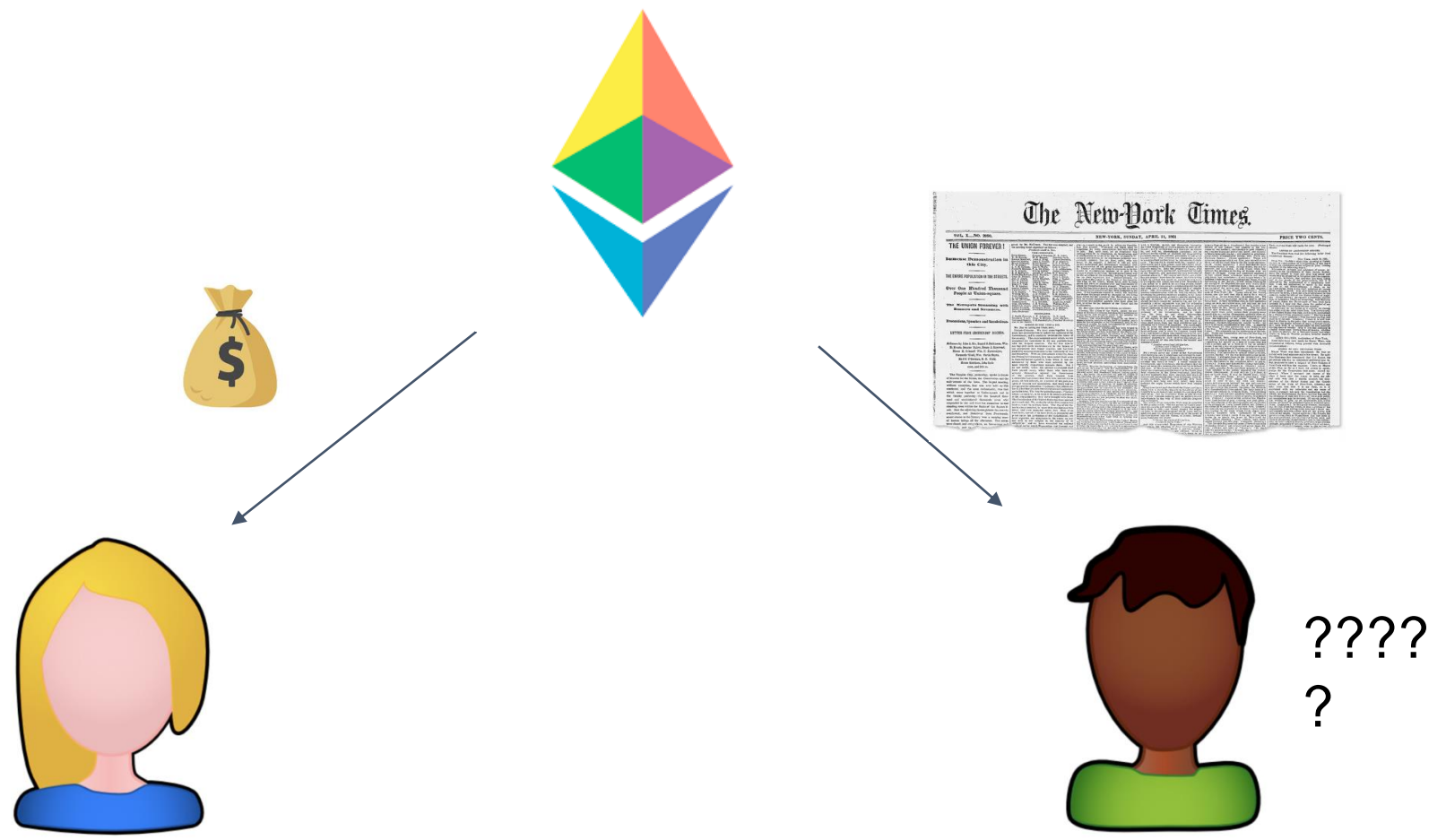


示例应用2：零知识生物识别（ZK Biometrics）

问：如果无法追踪
每个地址对应的人，
世界币如何向全球
每个人发放1个世
界币？



应用3：去中心化Kaggle，模型或数据交易



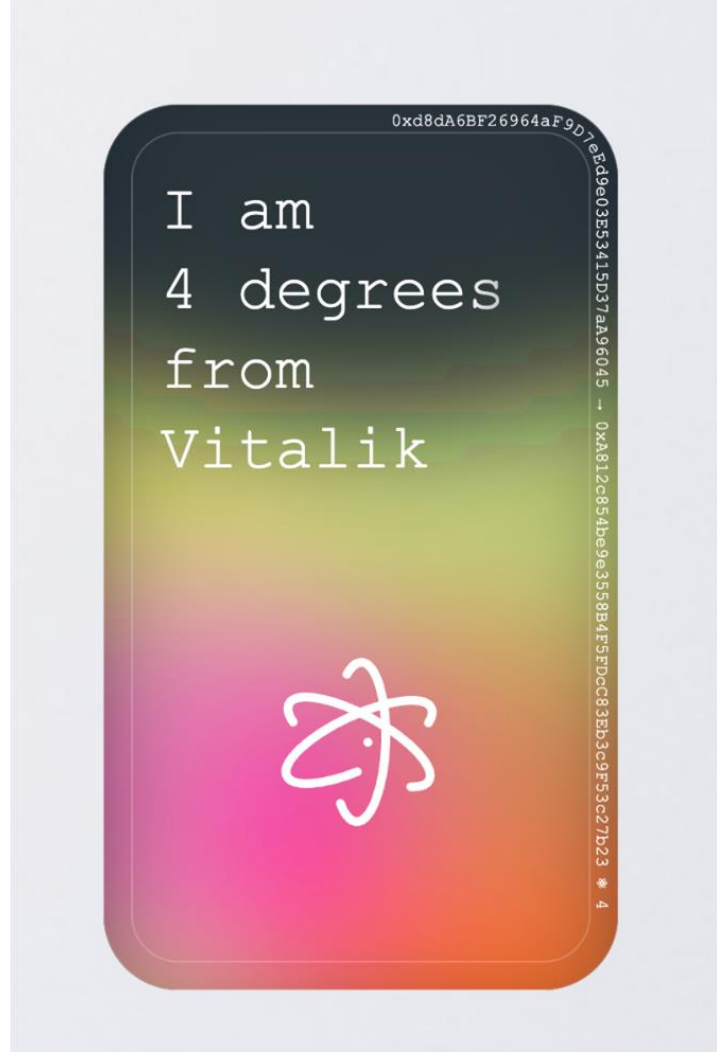
递归零知识证明

zkSNARKs允许您生成任何程序执行的简洁证明。

zkSNARK验证本身也是一个程序!

Application 1: ETHdos

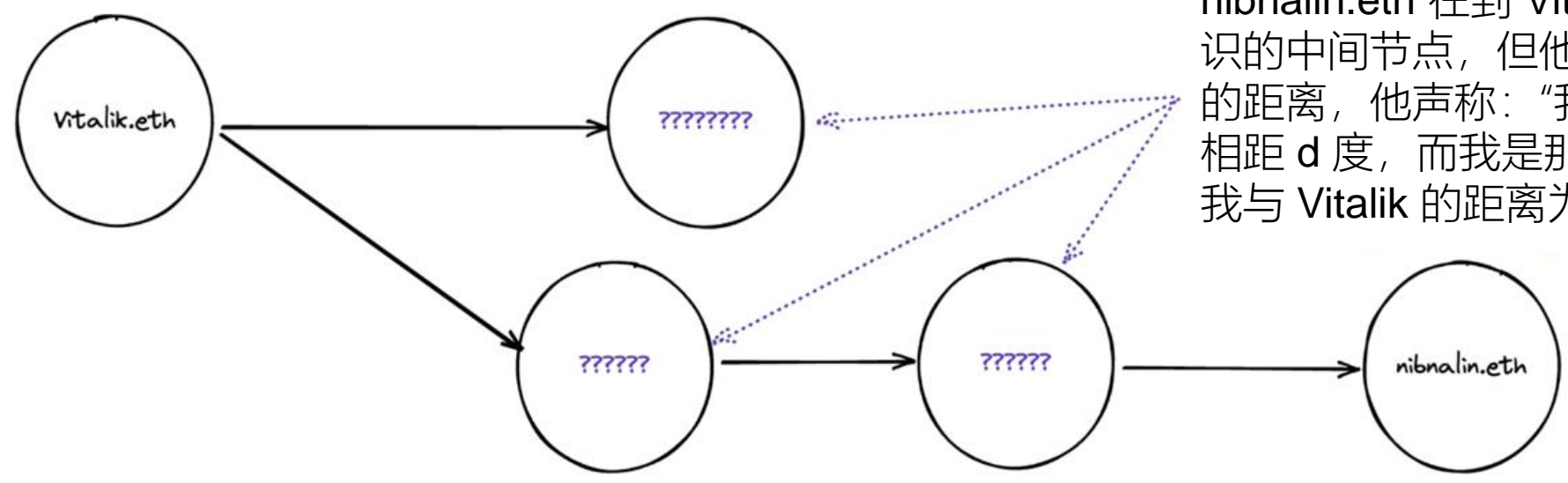
- ETHdos是一项独特的社会实验，利用递归SNARKs的独特可组合性质。
- 就像Erdos数字和Bacon数字一样，ETHdos数字衡量您与Vitalik之间的分离程度。您的数字越低，您离Vitalik越近！
- 要获得ETHdos数字，请向您的朋友请求添加！零知识证明将隐藏您与Vitalik之间的中间路径，不仅对他人，而且对您自己也是如此！



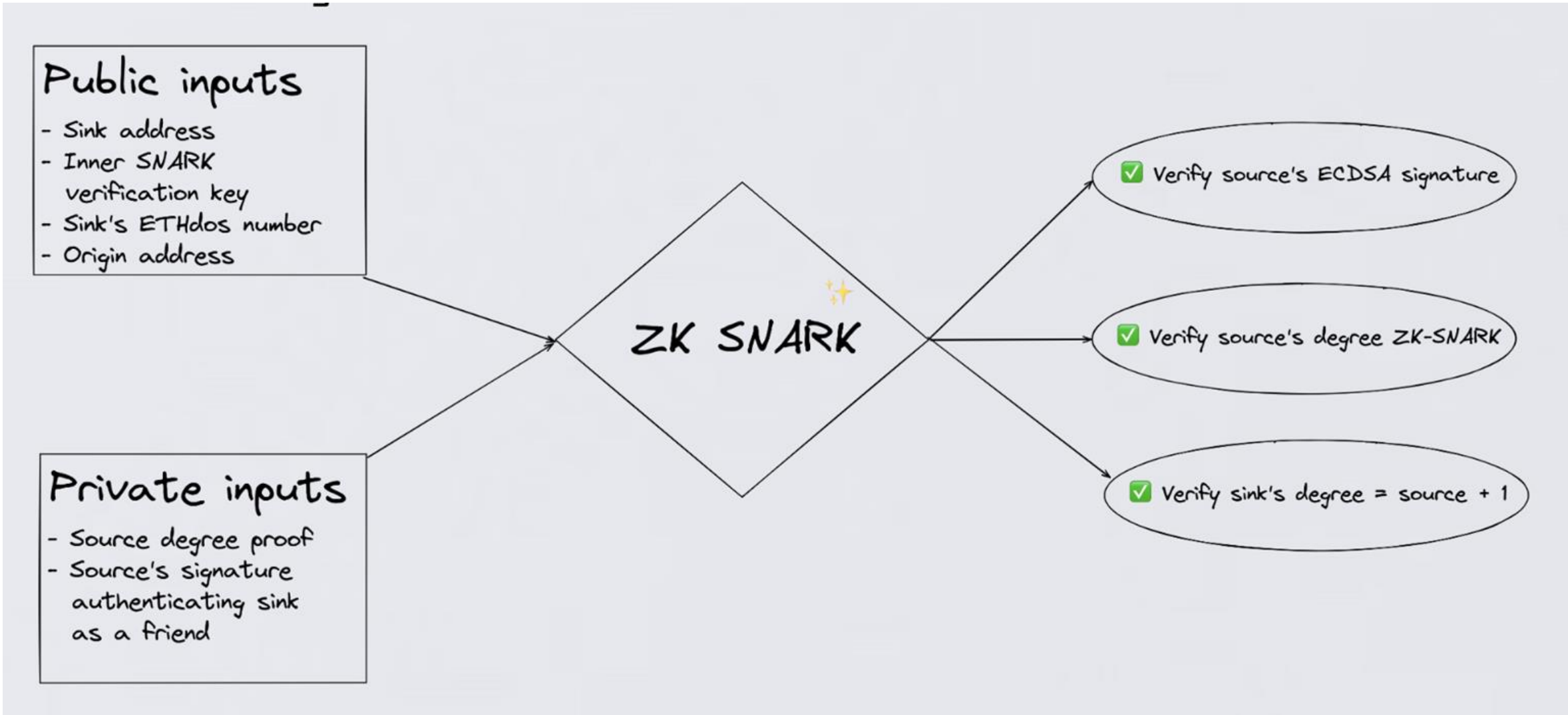
Application 1: ETHdos

From nibnalin.eth's perspective

nibnalin.eth 在到 Vitalik 的路径中没有认识的中间节点，但他知道自己与 Vitalik 的距离，他声称：“我知道有人与 Vitalik 相距 d 度，而我是那个人的朋友。因此，我与 Vitalik 的距离为 3 度。”



Application 1: ETHdos



ETHdos

<https://ethdos.xyz/blog>

Nalin, Adhyyan, Vivek, Sampriti