

# Decentralized Finance

## Decentralized Identities

Instructor: Dan Boneh, Arthur Gervais, **Andrew Miller**, Christine Parlour, Dawn Song



# Outline

---

## ***1. Linking External Accounts to DeFi***

- Attestation model, Decentralized Identifiers
- Anonymous credentials from Zero Knowledge Proofs

## ***2. Identity Authorities and Real Names***

- Real names and regulations
- Using CanDID to bootstrap credentials from legacy authorities

## ***3. Future of Defi: relying less on authority***

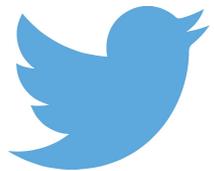
- Webs-of-Trust
- Proofs of Personhood



# 1a: Linking external Accounts

---

# Motivating Example: Airdrop to twitter users



@AliceToGo



Alice

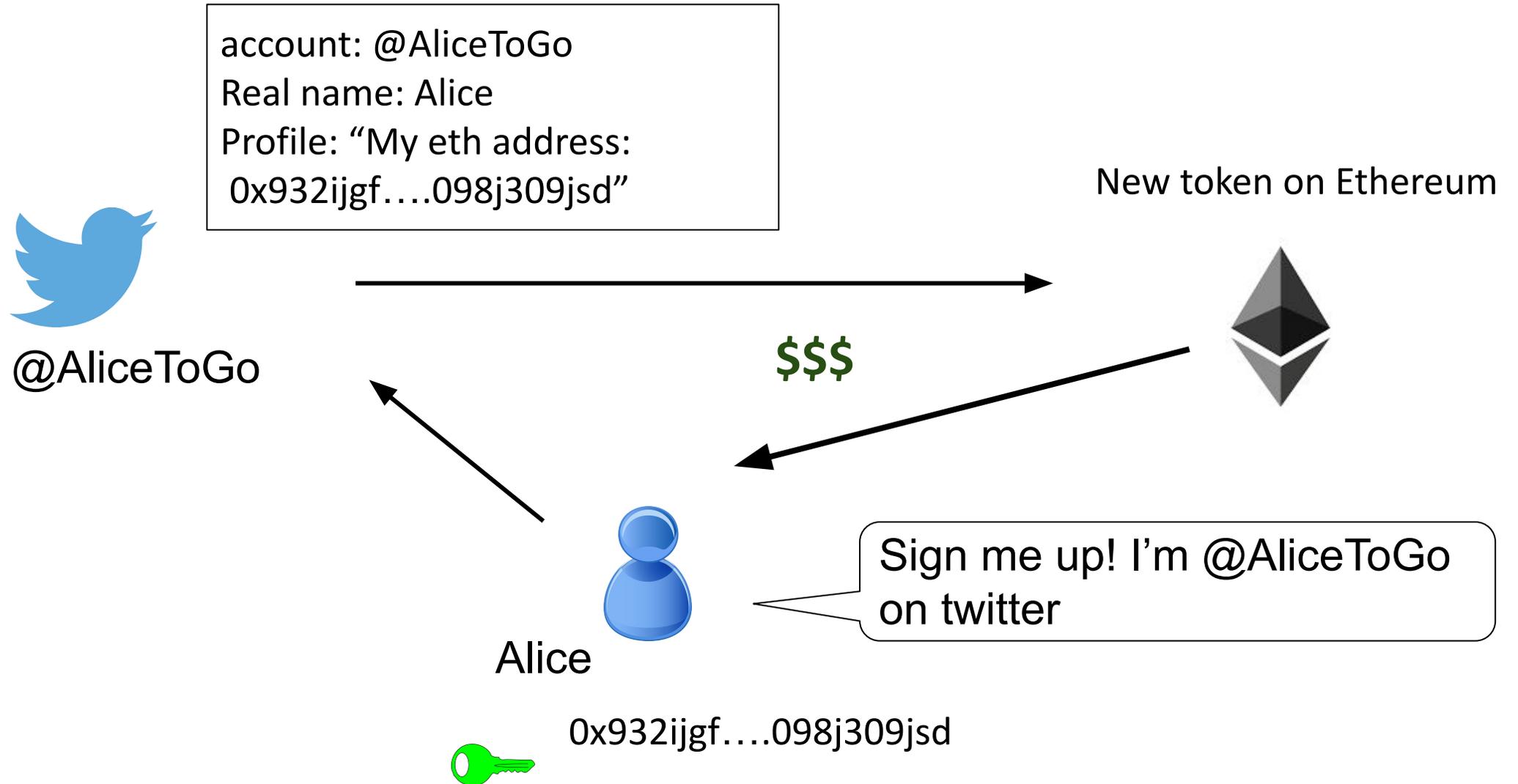
\$\$\$

New token on Ethereum



Sign me up! I'm @AliceToGo on twitter

# Motivating Example: Airdrop to twitter users



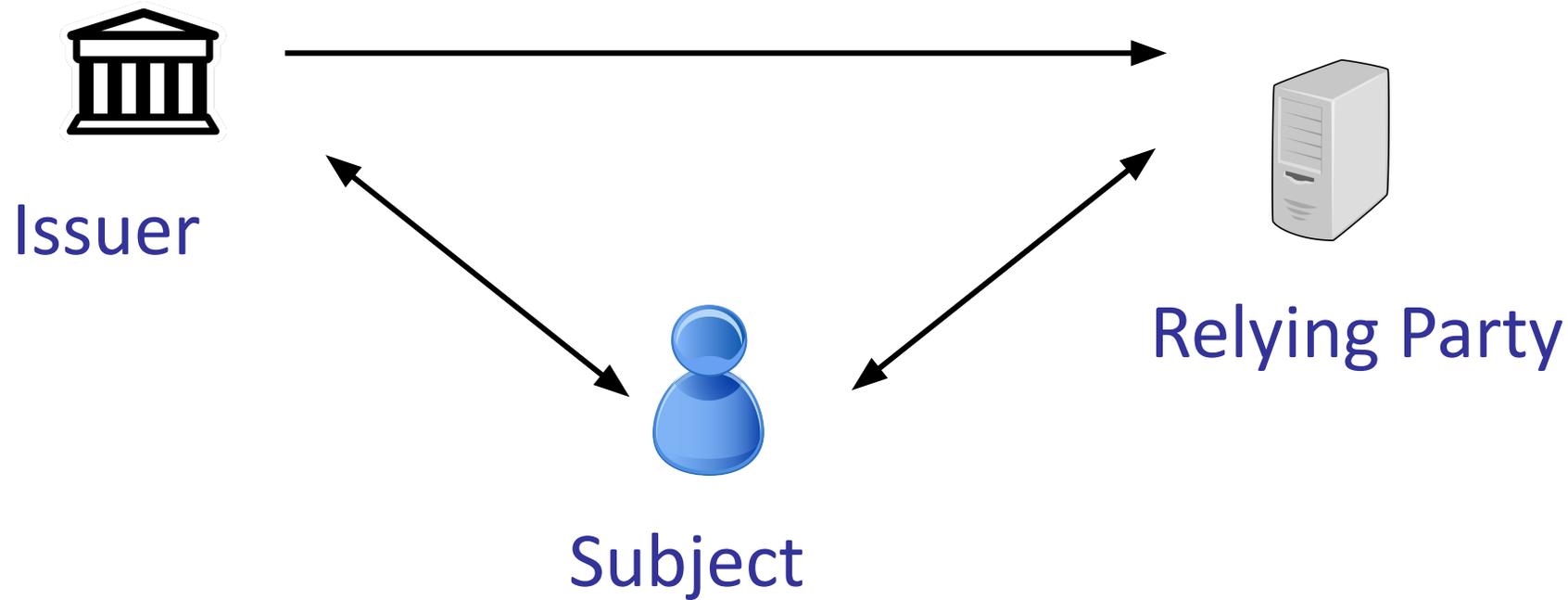
# Why would DeFi want to link external accounts?

---

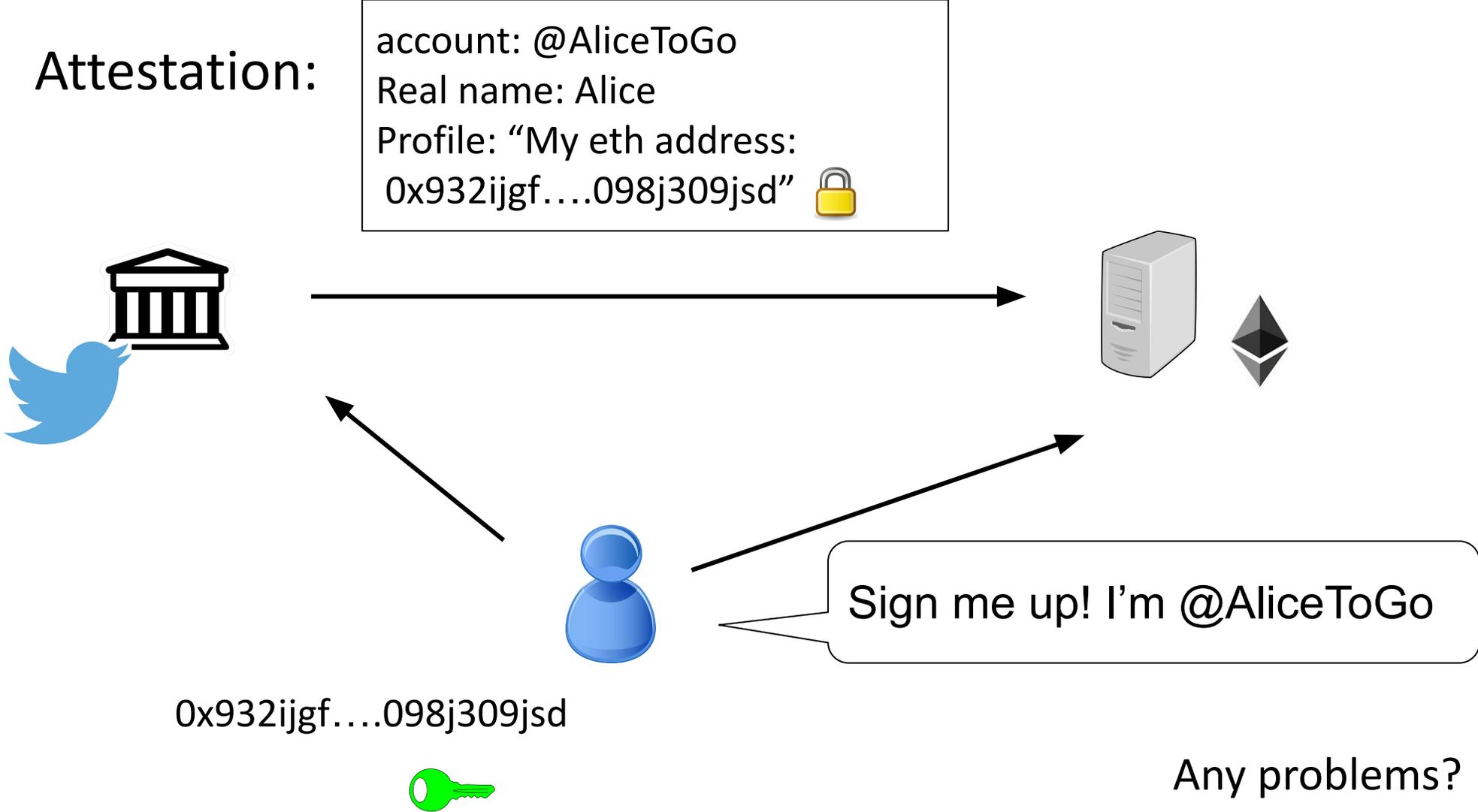
- Airdrops, tips, giveaways to social media accounts
- Reputation as collateral
- Simplifying user registration & login with Single-Sign-On
- Avoid spammers, botnets, farms, and sybil attacks
- Provide an alternate way to recover a lost account
- Ensure “one person one vote” for fairness in governance

# Simple Attestation Model

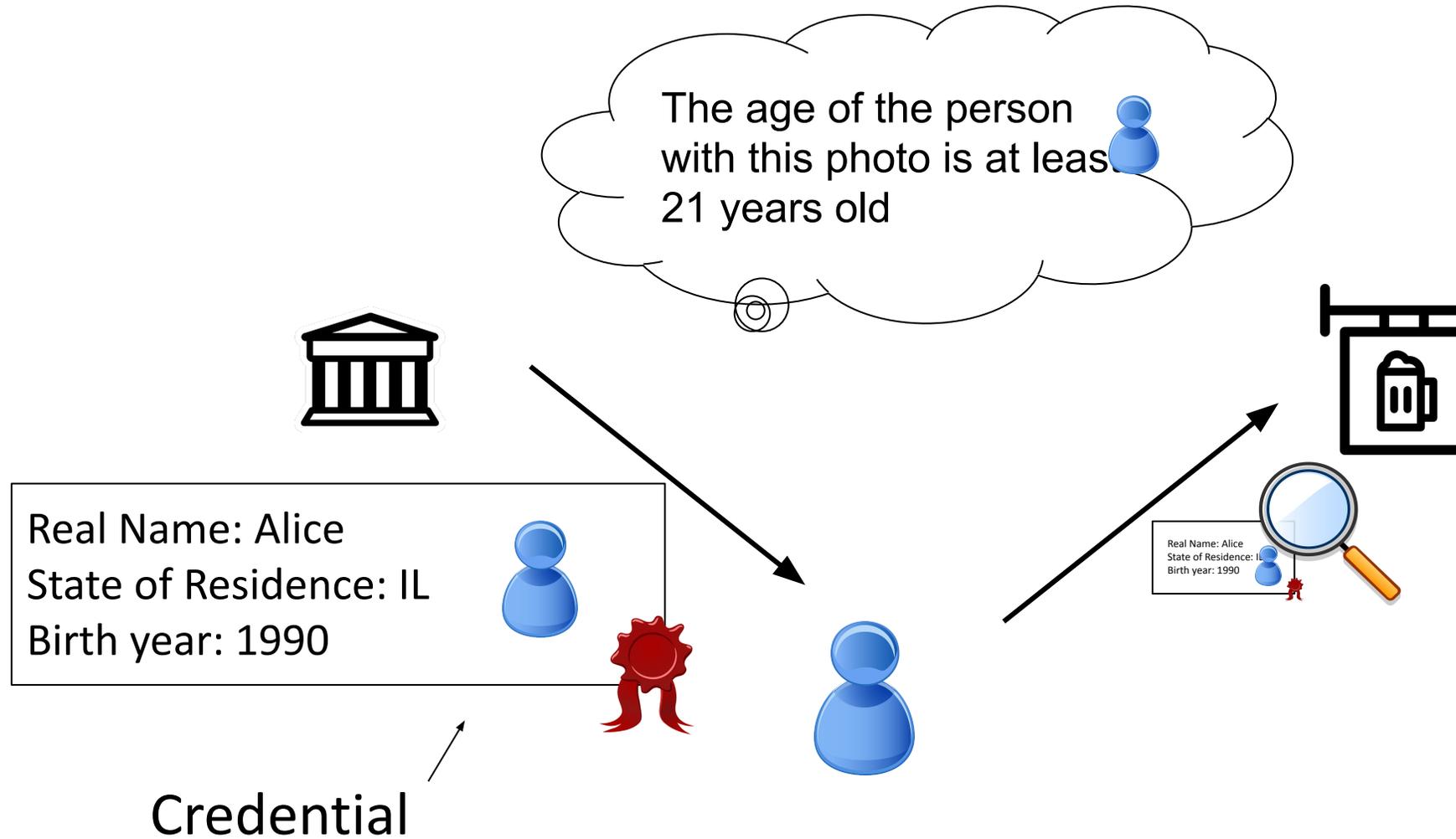
Attestation: “something about the subject”



# Twitter Airdrops in the Attestation model



# Entering a Bar in the Attestation model



# Summary: questions to ask in Attestation model

---

- ***Privacy***. What does the issuer learn about the Subject's interaction with Relying party?
- ***Availability***. Can the Issuer prevent the interaction?
- ***Revocation***. Does the Issuer have the ability to revoke the attestation? Does the Subject?
- ***Meaning***. What does the attestation say? Does the Issuer guarantee it's accurate?



# 1b: From OAuth to Anonymous Credentials



### Log In ✕

User

Password

[I forgot my password](#)

 with Facebook

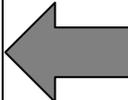
 with Google

 with GitHub

 with Twitter

 with VK

 Log In    [Create New Account](#)





## Authorize discourse.pro



**discourse.pro** by [discourse-forum](#)  
wants to access your **amiller** account



**Personal user data**  
Email addresses (read-only)



Cancel

Authorize discourse-forum

Authorizing will redirect to  
<https://discourse.pro>



Not owned or  
operated by GitHub



Created 7 years ago



Fewer than 10  
GitHub users

[Learn more about OAuth](#)

# Logins with OAuth

Attestation:

Account: @amiller  
Email address: amiller@email.com



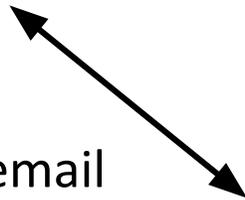
GitHub

Requesting: Read Email address



discourse.pro

Approve: Share email  
with discourse.pro?



Requests login



# Credentials based on digital signatures

Attestation:

Account: @amiller  
Email address: amiller@email.com  
Enrollment dates: 2021-2022



Defi University



Signed enrollment  
certificate



studentdiscount.com



Presents  
certificate

# Improvement: Fine-grained disclosure

Attestation:

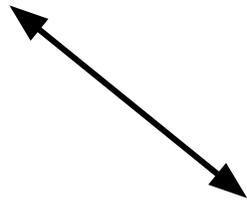
Account: @amiller Email address: amiller@email.com	
Account: @amiller Enrollment dates: 2021-2022	
Account: @amiller Date of birth: 05/05/2005	



studentdiscount.com



Defi University



agediscount.com

Signed enrollment certificate consisting of 3 different attestations

# Improvement: Fine-grained disclosure

Attestation:

Account: @amiller Email address: amiller@email.com	
Account: @amiller Enrollment dates: 2021-2022	



Defi University



Signed enrollment certificate consisting of 3 different attestations



studentdiscount.com



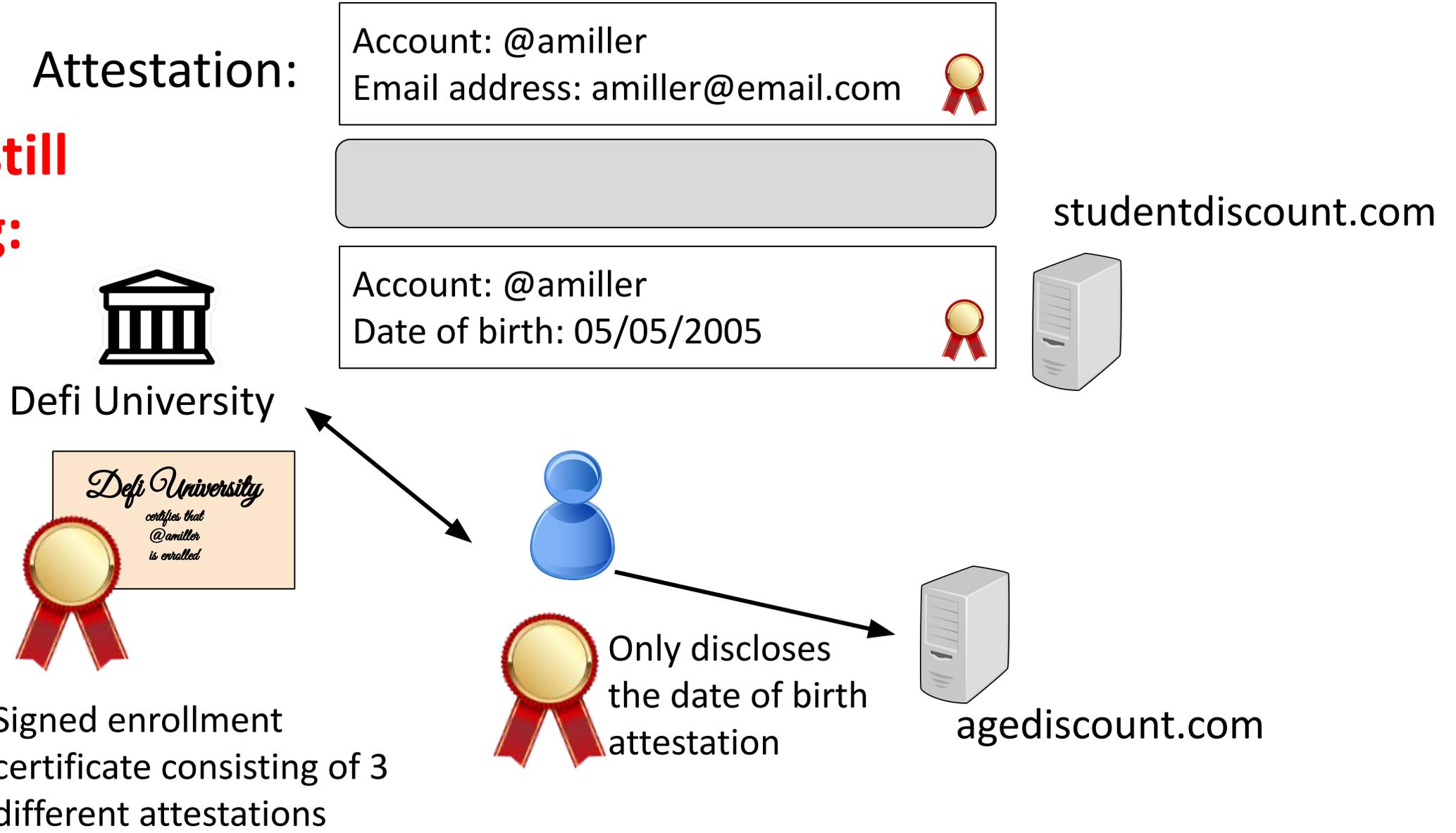
Only discloses the enrollment attestation



agediscount.com

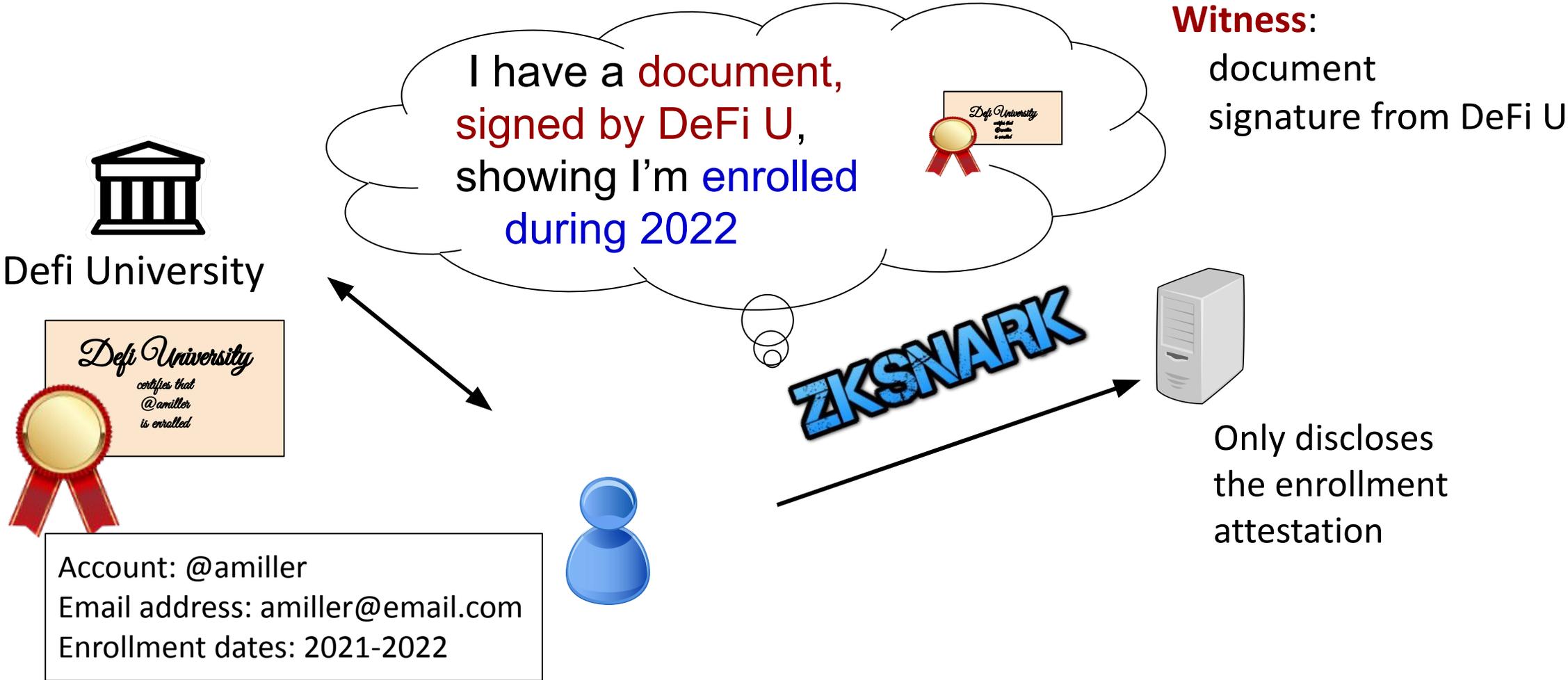
# Improvement: Fine-grained disclosure

**Problem still remaining:  
Privacy!**

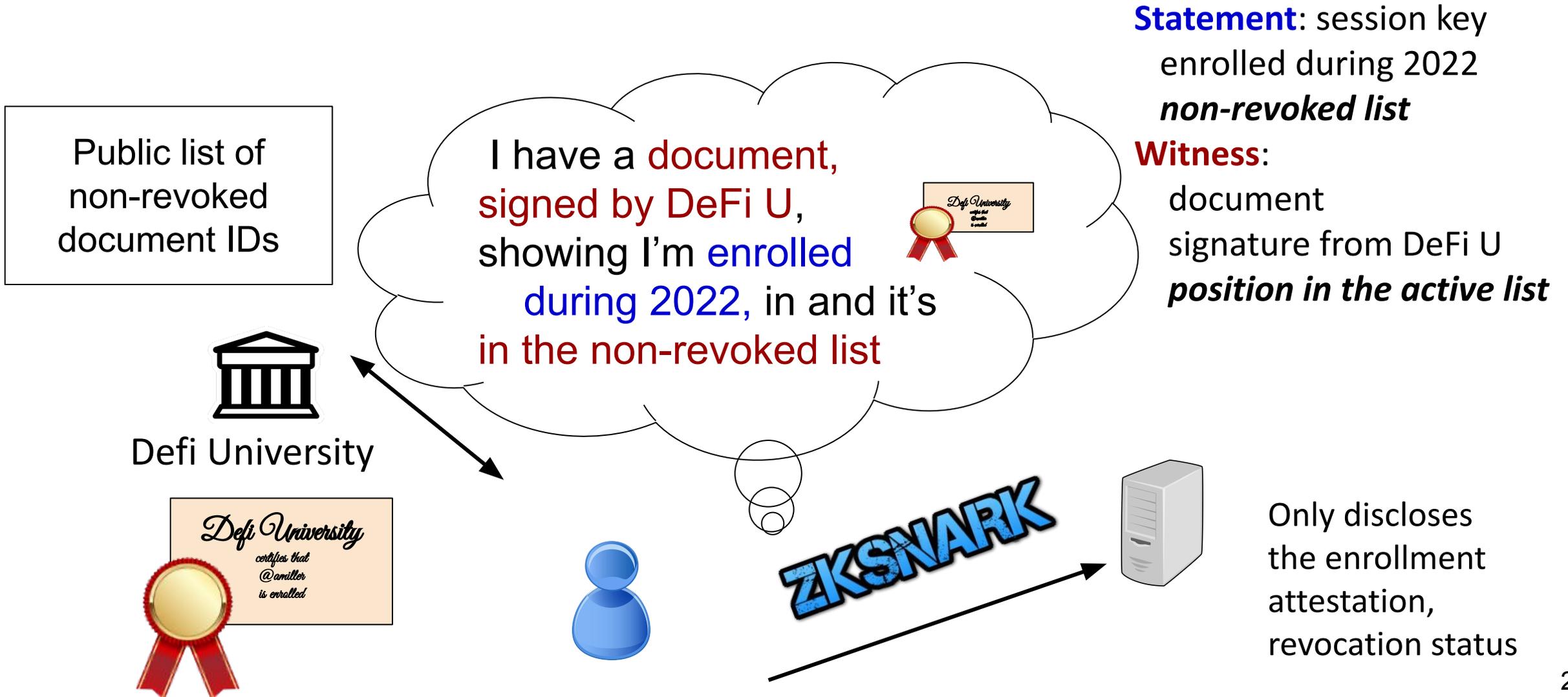


# Anonymous Credentials

Use a ZK SNARK! (like from Lecture 10)



# Anonymous Credentials w/ Revocation



# Summary so far

---

Lots of uses for linking external accounts into the DeFi system

It's simple to do, if we don't consider privacy at all.

Digital signatures and zkSNARKs can enable anonymous credentials that require minimal privacy and reliance on the issuer.



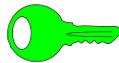
# 1c: Decentralized Identifiers

---

# Identifiers and Short Names

Does this string identify a unique person?

- Alice



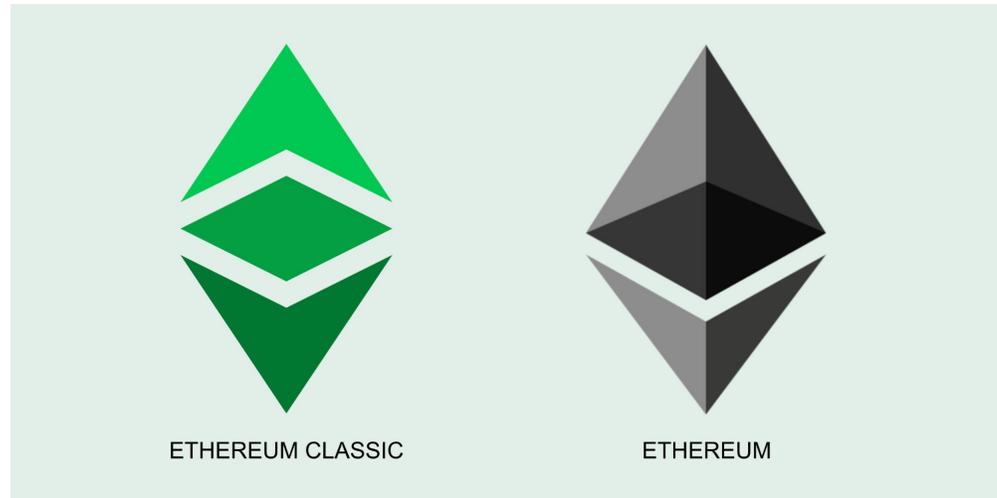
What about

- 0xj980w43g90j0934g09j43g09jw (Alice's public key)

# Identifiers and Short Names

Surely at least this is global...

- 0xj980w43g90j0934g09j43g09jw (Alice's public key)



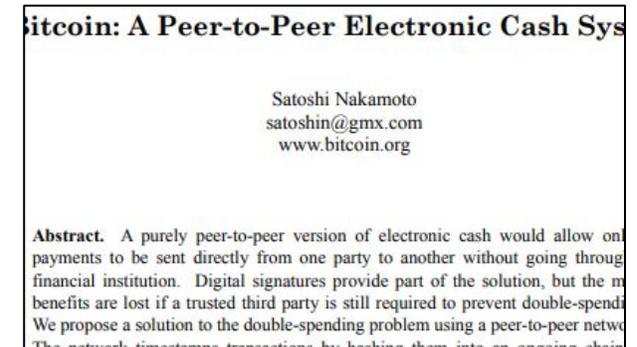
- eth:address:0xj980w43g90j0934g09j43g09jw

- etc:address:0xj980w43g90j0934g09j43g09jw

# Identifiers and Short Names

Does this string identify a unique document?

- bitcoin.pdf
- <https://bitcoin.org/bitcoin.pdf>
- <https://bitco.in/bitcoin.pdf>
- <ipfs://QmTzD4g5FFgn...XQCnYyfffxN/bitcoin.pdf>
- <https://ipfs.io/ipfs/QmTzD4g5FFgn...XQCnYyfffxN/bitcoin.pdf>



---

**“There are only two hard things in Computer Science: cache invalidation and naming things.” - Phil Karlton**

<https://www.karlton.org/2017/12/naming-things-hard/>

**“A name is good if it survives into its future context, but of course you don’t know what other names it will coexist with.” - Simon Hui**

# Smart contracts provide rich and flexible access control by default

---

```
address Alice = 0x230923907230984230994823;
```

- a public key?
- a multi-user policy
- some other program?

*Really, it doesn't matter to your defi smart contract*

```
function letMeIn() {  
    require(msg.sender == Alice);  
    // carry on  
}
```

# Some imaginative language features for the future

---

```
address Alice = 0x230923907230984230994823;
```

```
address Alice = twitter:@AliceToGo;
```

```
function letMeIn() {  
    require(msg.sender == Alice);  
    // carry on  
}
```

# To summarize... a few insights on Identifiers

---

Expect to manage between Implicit and Explicit Context

`alice` vs `context:alice`

Including cryptography in the identifier makes it longer, but adds support for access control:

- *hash* for uniquely defining a static document
- *public key* for identifying an owner who can update it
- a *program* defining some other policy

Try to pick names that will survive into their future context



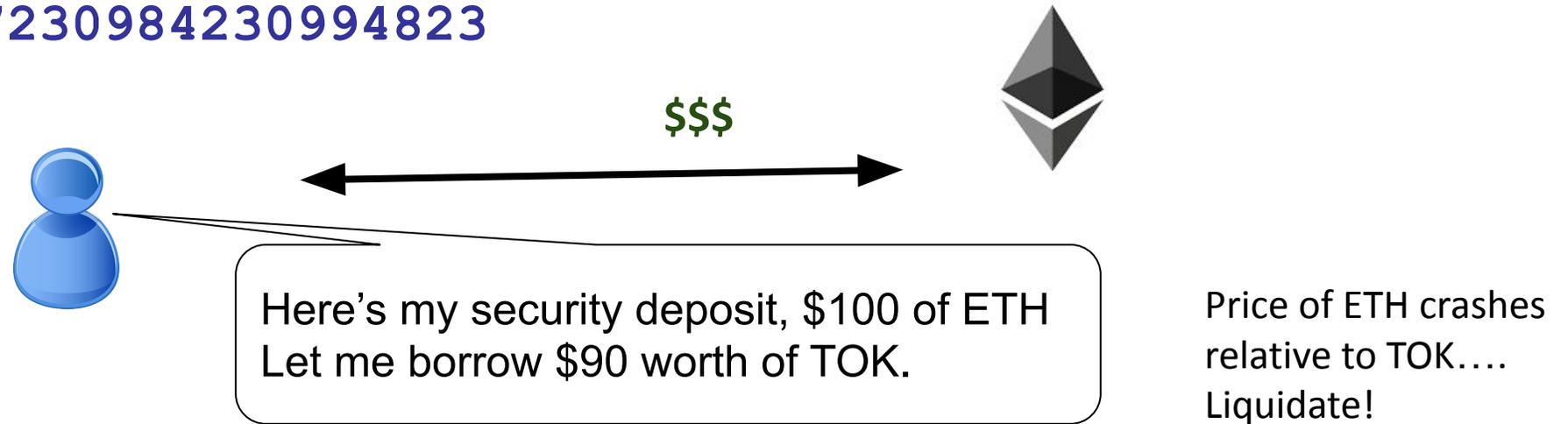
# 2: Authorities and Identity

---

# Real Names vs Accounts

## Lending and Borrowing in DeFi:

0x230923907230984230994823



If Alice becomes insolvent and owes more than she can pay, she can simply abandon the position  
The mechanism will liquidate the collateral, as best it can.  
Her other accounts are unaffected.

# Real Names vs Accounts

Borrowing in the world of real names:

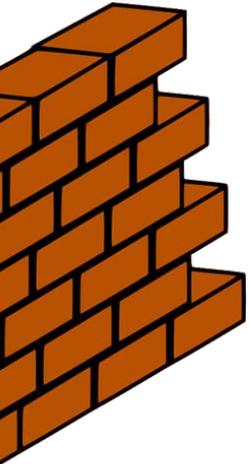
Alice

010111001010101



I'd like to borrow \$100.

\$\$\$



# Some obvious ways Real Names differ from accounts

---



VS



- Can't abandon them to escape debts, indictment, etc.
- Each person only gets one, can't create a second
- Not allowed to transfer or sell it to someone else

Real name identities are required for:

- exchange services
- even cryptocurrency ATMs
- ... many others



# Regulations create specific Identity requirements

---

## ***Example 1: Office of Foreign Asset Control (OFAC) Sanctions List***

- The US OFAC provides a sanctions list: parties (countries, groups of individuals) US organizations are not allowed to transact with.
- *“using the blocking of assets and trade restrictions to accomplish foreign policy and national security goals.”*

## ***Example 2: Bank Secrecy Act (BSA) aka Travel Rule***

- All “money services businesses” must comply with travel rule.
- Regulations regarding collecting and storing (for govt. audits) user data.
- Crypto exchanges and wallet providers are now deemed “money services businesses”.

***Exposes Tension between privacy and accountability...***

# Some more nuanced features of real name ID

---

- **Ability to show up in person to reissue**
- **Example: Personally Identifying Information (PII) as a Liability**  
Service providers that suffer data breach affecting PII of customers or employees, often have obligations to report it
- **Example: Right to be Forgotten**  
Service providers that receive consent to store PII, must also respect requests to remove them
- **Example: Bankruptcy protections**

*Alternatives to real name IDs  
may aim to satisfy some of these*



# 2b: Bootstrapping Legacy Credentials with CanDID

---

# The Bootstrapping Problem

---

Anonymous credentials assume there exist issuers willing to sign statements. Several problems with this:

- **Chicken and egg problem:**

- Actually convincing issuers to sign credentials: incentive if identity ecosystem exists.
- No identity ecosystem without issuers.

- **Limited APIs:**

- Even if issuers are convinced, the capabilities would be limited.
- Issuers have incentive to hoard data, taking away user control of it.

- **Privacy loss:**

- Issuers learn the credential APIs users are accessing.

# CanDID: Can-do Decentralized Identity

---

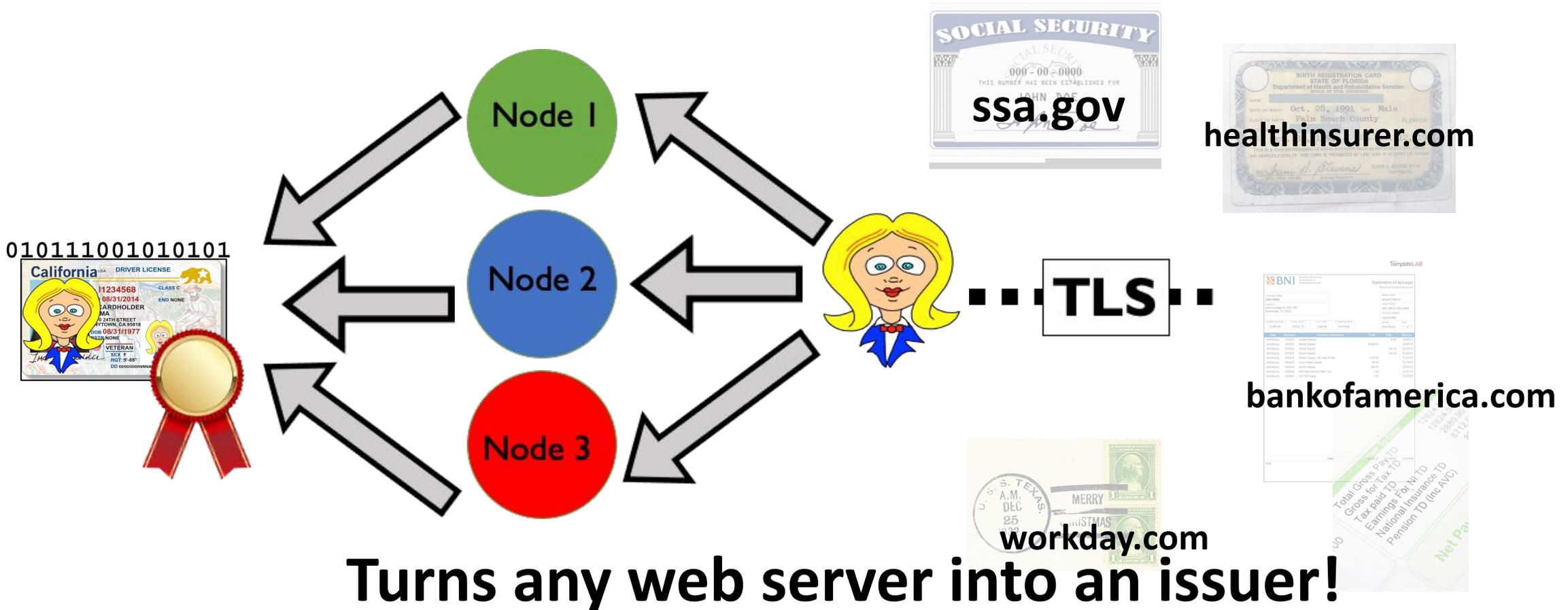
CanDID is meant to provide an identity system which provides the following capabilities:

- Bootstrapping identity, using legacy data [flexibly!].
- Key management / ability to re-issue
- Sybil-resistance (one-person-one-ID)
- Support for regulation compliance

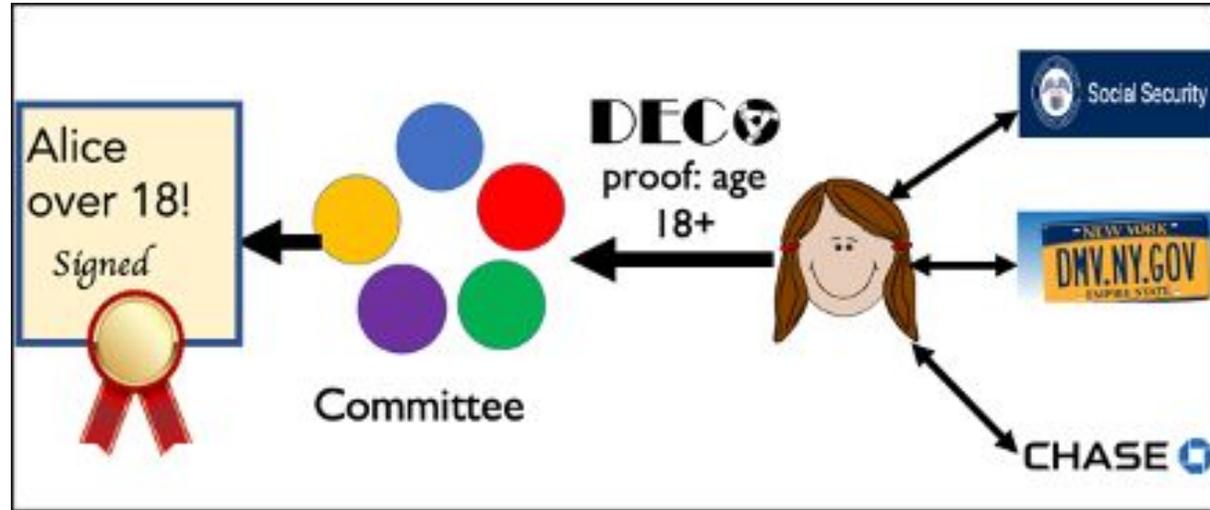


# Bootstrapping Identity: For DIDs?

- **Solution:** Oracles from Lecture 8.
- Can be used for porting authenticated data from legacy sources.

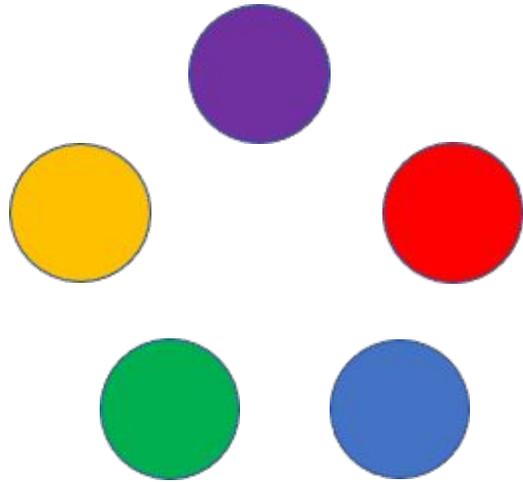


# Oracle-based Credential Issuance Properties



- **Privacy:** Committee learns only attested data
- **Legacy compatibility:** No web server modification
- **Unlimited attestation types:** Any web data can feed attestation

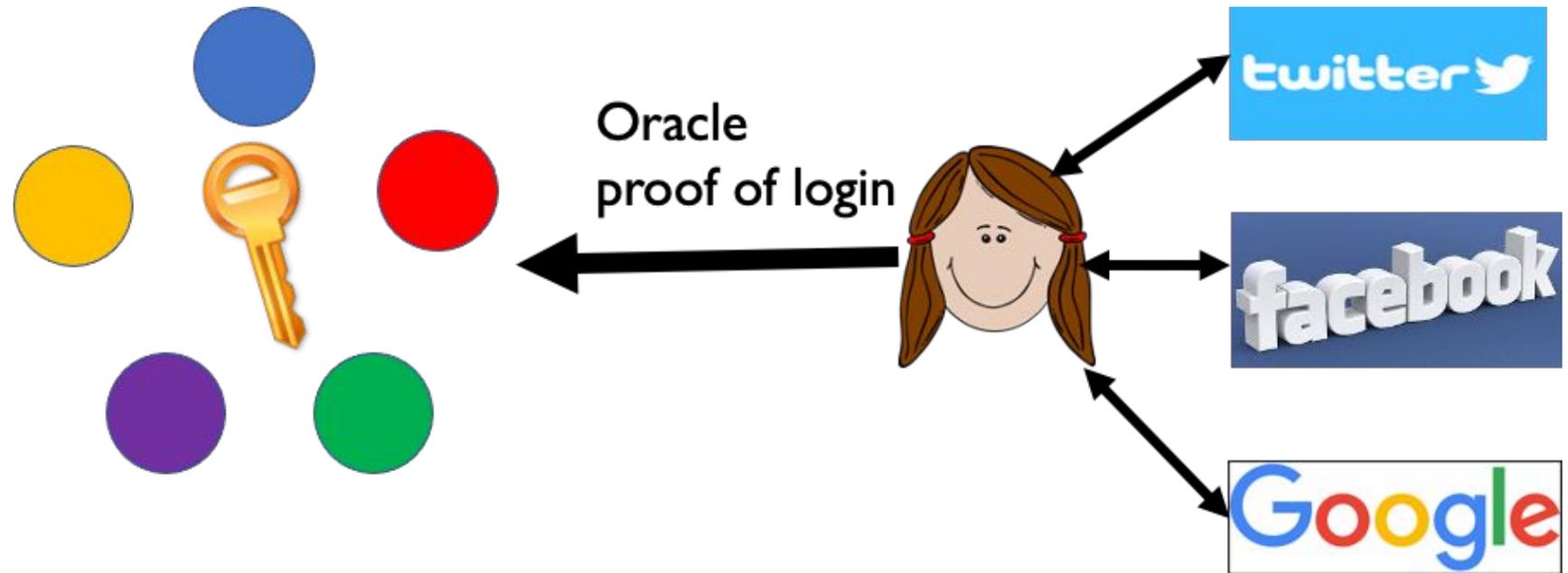
# Tool for Key-Management: Secret-Sharing



Committee of nodes

- Each node in the committee holds a “share of a secret”.
- No node learns secret by itself.
- Some threshold ***t-out-of-n*** nodes needed to reconstruct the secret.
- Eg: to store the binary random number 1011 secret shared by 3 nodes:
  - Share 1: Picked at random = 0001
  - Share 2: Picked at random = 1010
  - Share 3: Picked to allow reconstruction = 0000 = 1011 XOR 0001 XOR 1010
  - 3-out-of-3 secret sharing

# Oracle-based Key Recovery



- Secret-share the key with a committee of nodes.
- Set a policy for the accounts you need to show you can log in to for recovering the key.

# Summary: Bootstrapping legacy credentials

---

Desired properties of anonymous credentials can still be compatible with the use of identity authorities.

This can work even without requiring legacy providers to support this.

Key technique: oracles, using technology like zkSNARKs or trusted hardware



# 3: Future of Identities in DeFi

---

# Principle of Least Authority

From information security:

never log in as “Admin” when “guest user” will do

When something goes wrong,  
limited damage

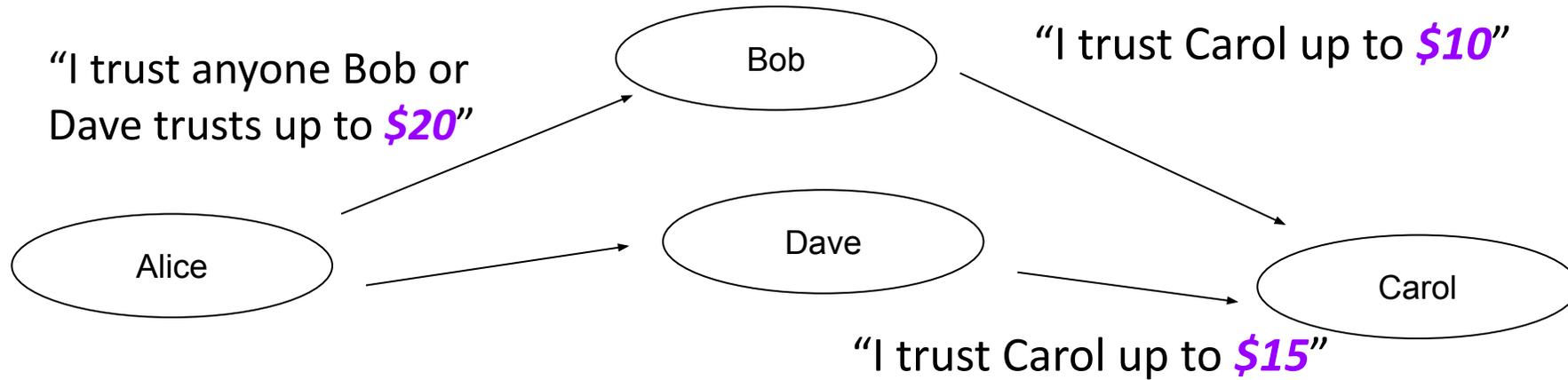
Trust premium.... less reliable  
alternatives are often cheaper

Avoid collecting PII... it's a liability



Idea 1: Can we build a reputation system without relying on an authoritative credit score?

# Webs of Trust



**Nodes:** People, Accounts, entities

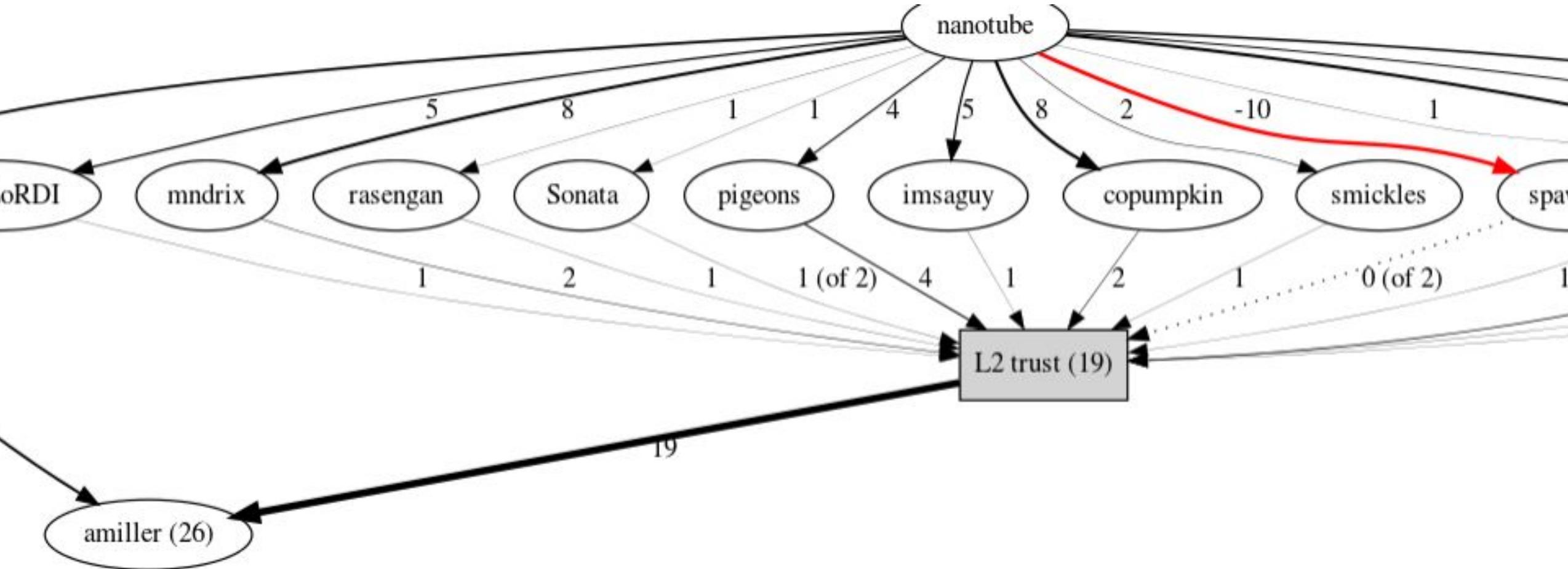
**Edges:** an expression of “trust,” a credit line, or a record of prior interaction

**Paths / Flows:** indicate a “friend of a friend,” transitive trust

Should we recommend an **\$18** trade between Alice and Carol

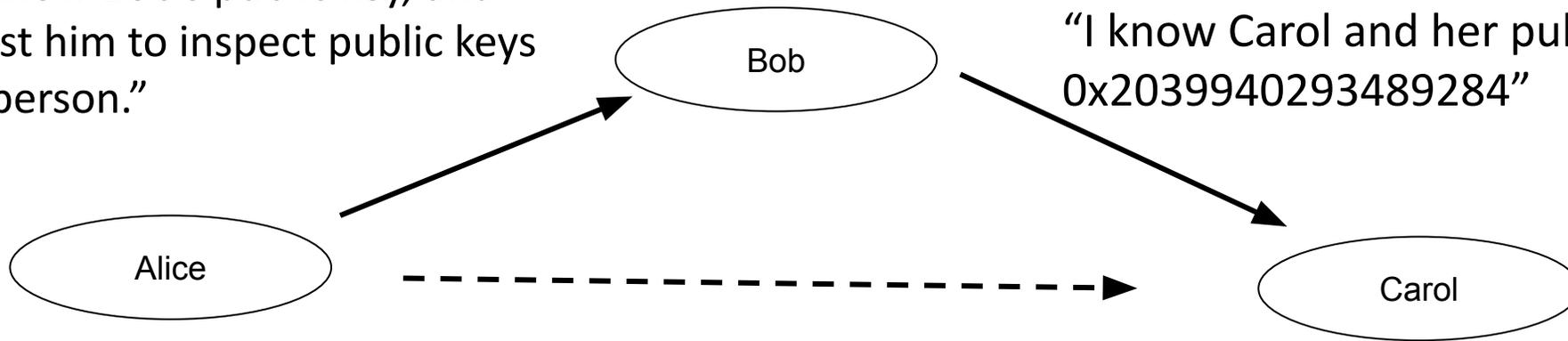
# Bitcoin-OTC: 12,000 users with PGP keys

“Web-of-Trust” reputation tracking tool



# Webs of Trust for Identity

“I know Bob’s public key, and I trust him to inspect public keys in person.”



“I know Carol and her public key is 0x2039940293489284”

<u>Buddy List</u>	<u>Pubkey</u>
Carol	0x20390...

imported Carol’s public key  
based on transitive trust

Idea 2:

If all we want is ***one-person-one-account***,  
but no other identifying info is needed,

can we do this in a ***privacy-preserving*** and  
***least-authority*** way?

# **ATTENTION CUSTOMERS**

**Due to high demand, quantities of  
these select products are limited to  
2 PER CUSTOMER.**

*Thank You*

**for your understanding and cooperation.**

# Quadratic Funding & Why it needs identity

---

Donors contribute to public goods project they want to see funded.  
A large matching organization is willing to match donations,  
but still wants to harness the wisdom of the crowds

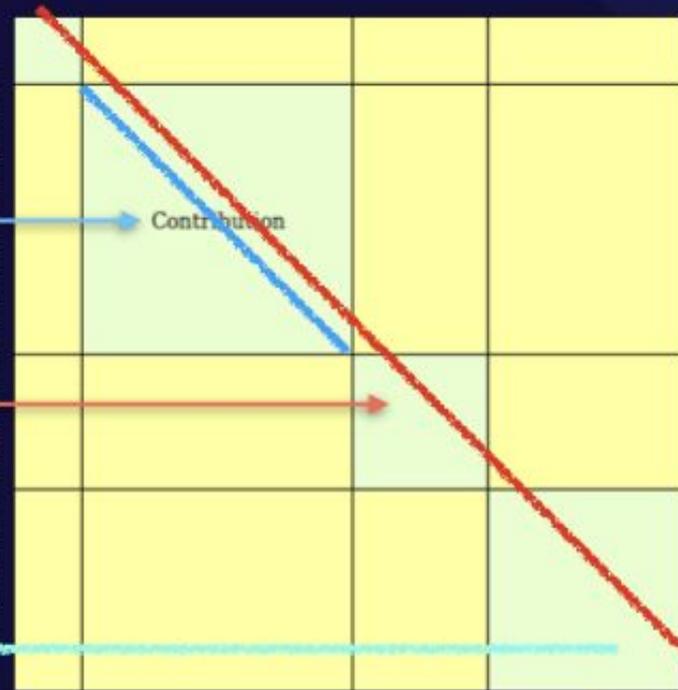
***Tension:*** Want to encourage small contributors, while still allowing large contributors to contribute.

***Idea:*** scale the amount of “matching” by the *square-root* of each contribution

square root  
of a single contribution

the sum of  
all square roots

the largest square:  
total fund given



<https://vitalik.ca/general/2019/12/07/quadratic.html>



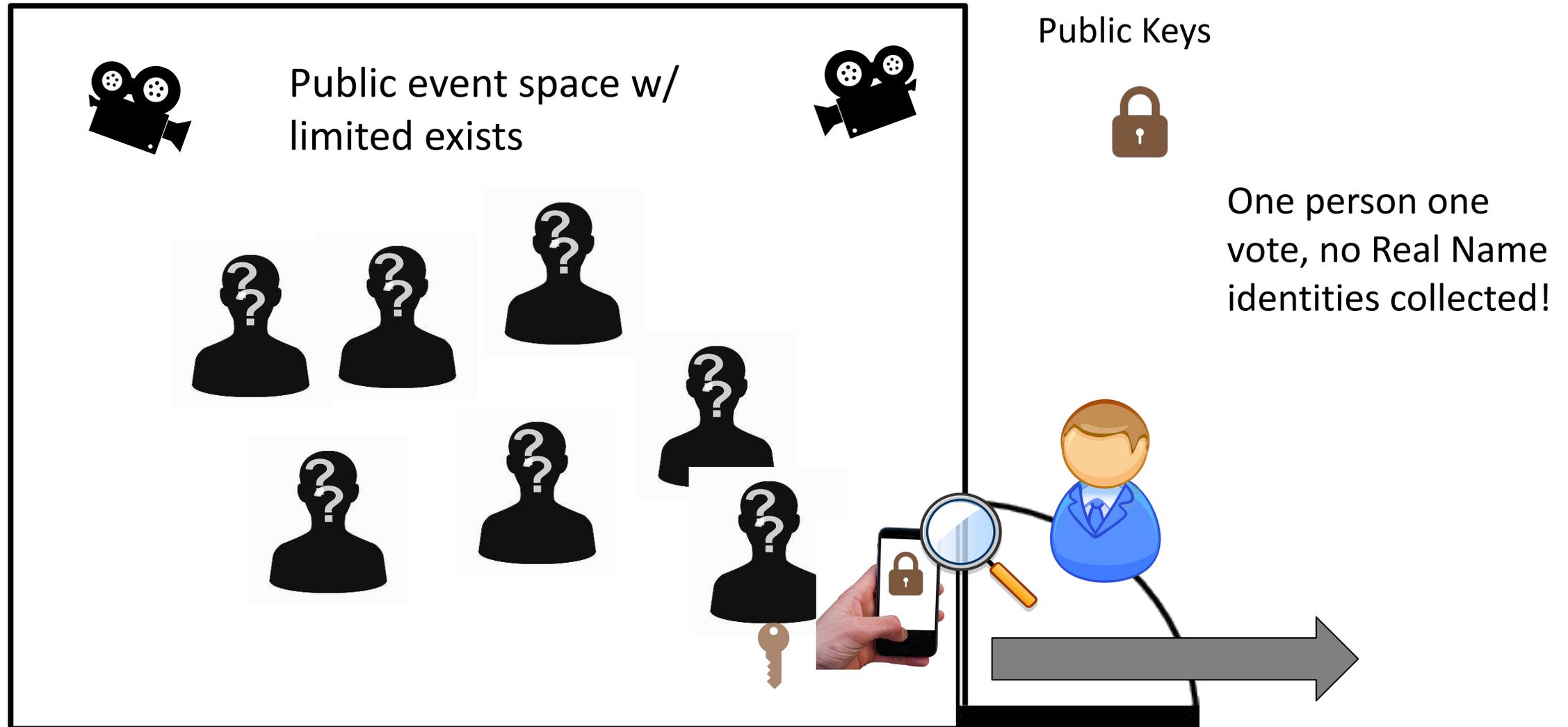
What can go wrong  
if a large  
contributor can  
create dozens of  
identities?

Quadratic Funding takes the square root of each community contribution, adds them up, and takes the square of its sum. After that, the grant agency (Gitcoin) pays for the difference between the “QF” result with the matching fund from large institutional donors like the Ethereum Foundation and other prominent DeFi projects.

As a result, this algorithm disproportionately awards matching funds to grants that have a lot of small donors over grants that have few large donors, in effect giving more credence to the number of people supporting a project rather than the number of dollars supporting it.

How can we achieve one-person-one-account  
without requiring real name identity?

# Proofs of Personhood - Pseudonym party



# Closely related: Proofs of Attendance



# Summary: Decentralized Identity in DeFi

---

- DeFi developers have a variety of technical tools for dealing with identity management.... digital signatures, zkSNARKs, oracles.  
These can minimize the reliance on issuers for privacy, availability
- We can expect needing to be flexible with naming schemes due to forks, competing projects, and rapidly evolving nature.
- The Future of decentralized identity may require creative ways to use low-authority identifiers rather than traditional ones