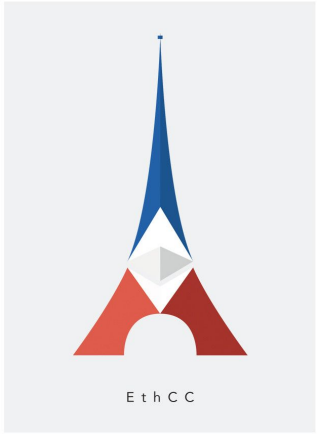


OKIMS

iden3

iden3.io
@identhree

Scalable distributed identity system on ZKP

Jordi Baylina, Technical Lead
David Suarez, Project Lead

OKIMS

About iden3 project

- Beyond circom and snarkjs
- Protocols, data structures and modules
- Open Source with reference implementation

iden3

OKIMS

iden3 current objectives

- **Self-sovereign** Ethereum-based identities for all at **no cost**
- **Scalability** by off-chain model which minimizes on-chain transactions
- **Privacy by design** with zero knowledge and non-reusable proofs
- Complete solution with user **wallet**, libraries and **key management**
- Focus on **community standardization** and foundation for use cases

iden3

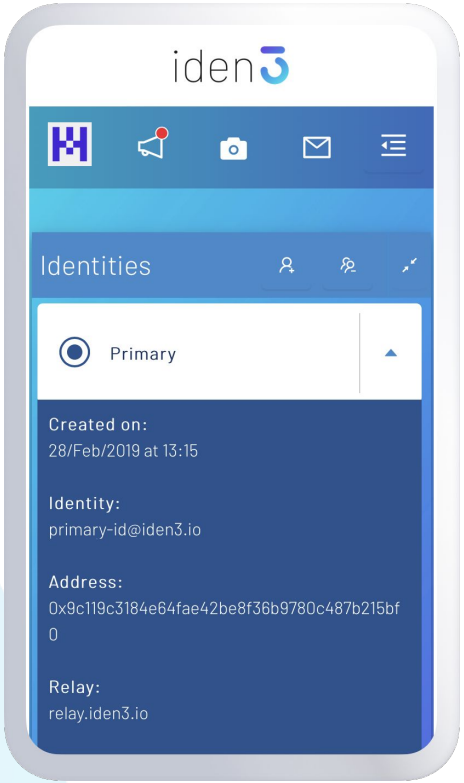
OKIMS

Identities wallet

- A new implementation of the web-based wallet:

  wallet.iden3.io

- Recovery by export/import of the complete wallet

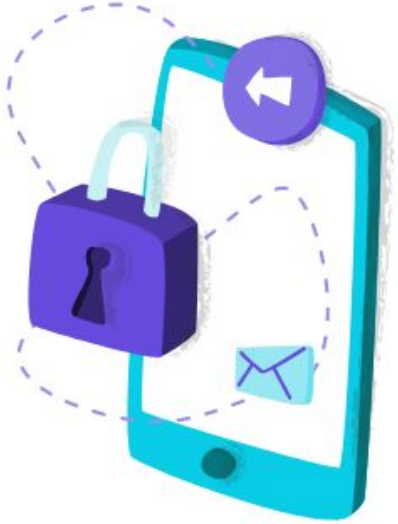


iden3

OKIMS

Design principles

- Design for Trust
- Design for Understanding
- Design for Data Ownership and Control
- Design for Usefulness



iden3

OKIMS

Design process

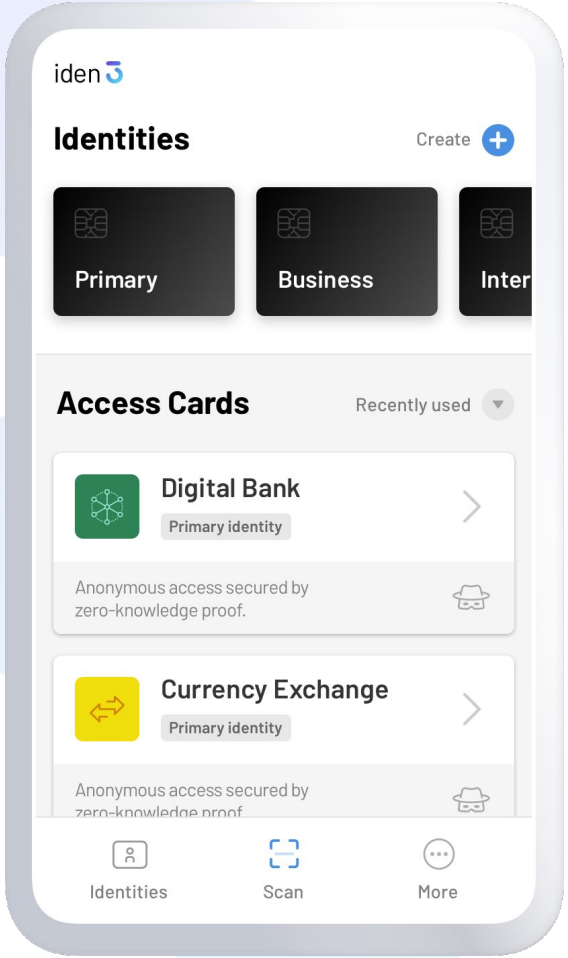
Research → Flows mapping → Usability testing → Release

- **User Research:** Understand user needs, security and privacy issues.
- **User Flows Mapping**: Prototype user interface for real-life scenarios.
- **Usability Testing and Iteration**: Validate design concepts and test the end-to-end service.
- **iden3 Identity Wallet Release**: Ongoing performance monitoring and improvement lifecycle.

iden3

OKIMS

Future wallet



iden3

OKIMS

Single sign-on for Apps

Complete identity provider scenario for classical centralized apps (Alpha)

All libraries and process of installation at:

https://github.com/iden3/centralized-login-demo

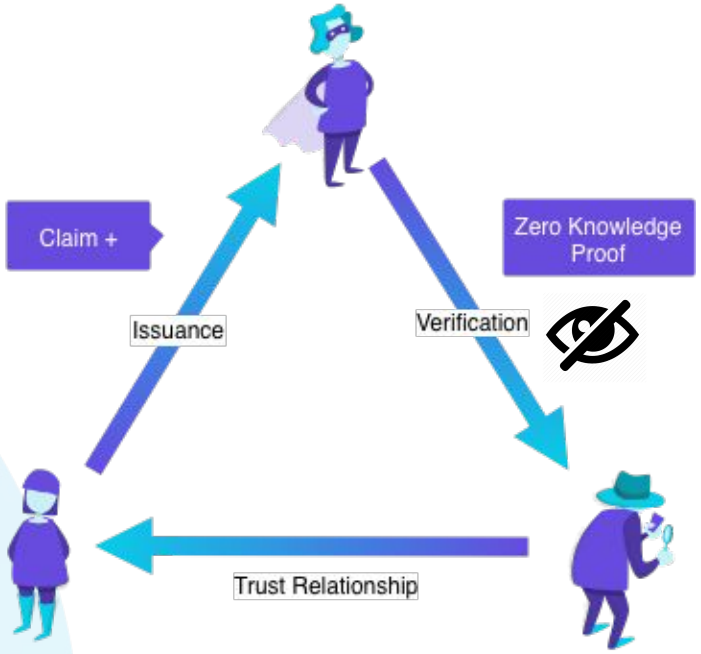
iden3

OKIMS

Privacy:
Zero-knowledge proofs

You can prove something **without revealing** unnecessary information:

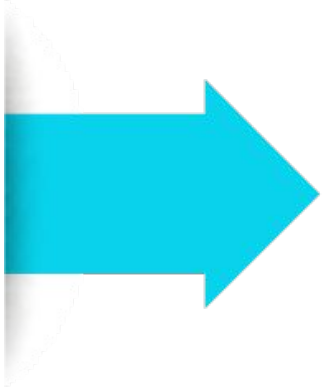
- Prove you paid your taxes
- Entrance to a nightclub proving that you are 18+
- Anonymous voting
- Participate in an ICO anonymously but with the warranty that a 3rd party KYC'd you



iden3

Identity genesis

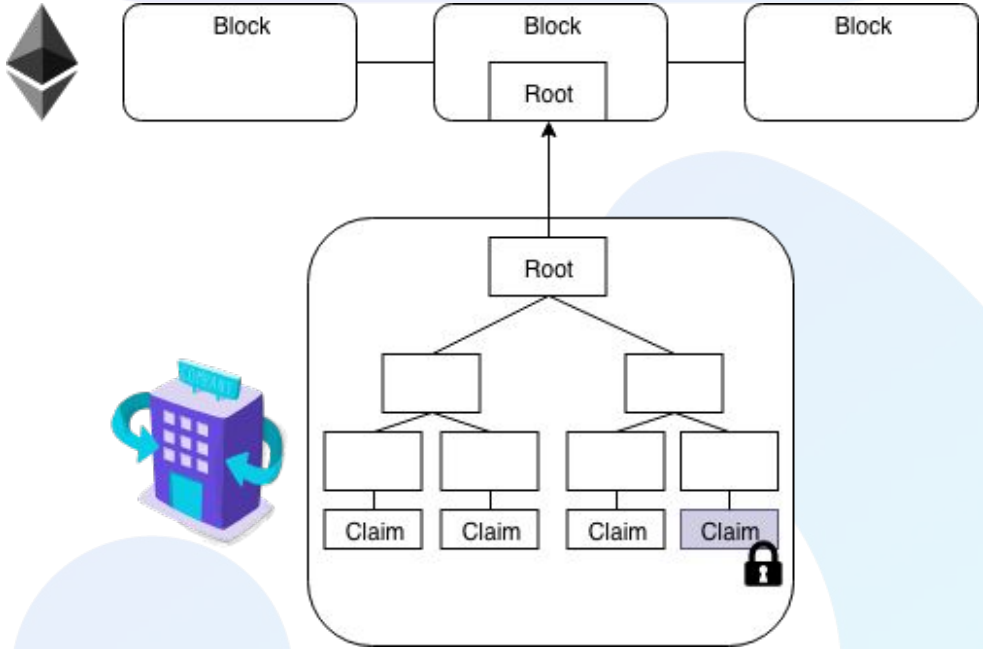
- Identity protocol
- Location of the official claim merkle root for this identity
- Recovery Mechanism
- Revocation key/s
- Initial operational keys
- Other metadata...



iden3

OKIMS

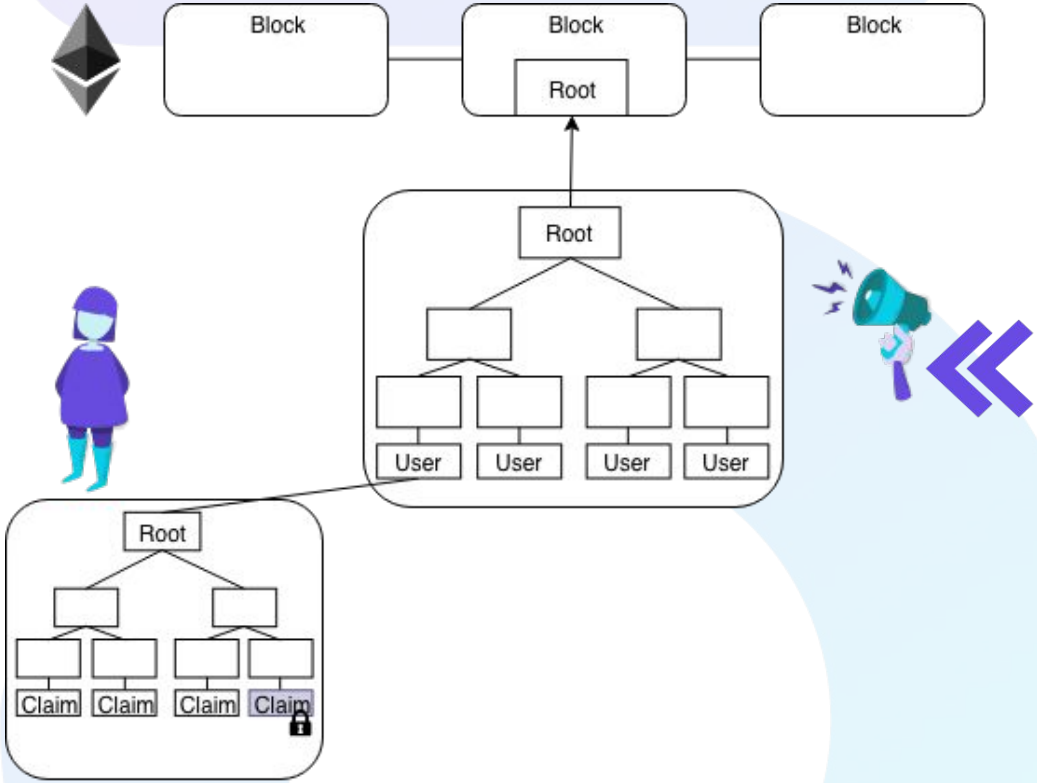
Direct Claims



- Claims are stored in a Merkle tree and the merkle root stored on chain.
- The history is **kept on chain**.
- A prolific claim generator can add/modify **millions of claims** on a single transaction.

iden3

Indirect claims



- An identity **signs** the claim tree root and **sends** it off chain to the relayer.
- With relayers, millions of users can create millions of claims on mainnet **without** spending any **gas**.

iden3

OKIMS

Identity Discovery Protocol

searcher

p2p network

Identity relayer

iden3

OKIMS

Identity based services

- Email
- Messaging
- Home page (Like linkedin)
- Public claims
- Social applications
- ...

iden3

OKIMS

Basic operations of an identity

Sign

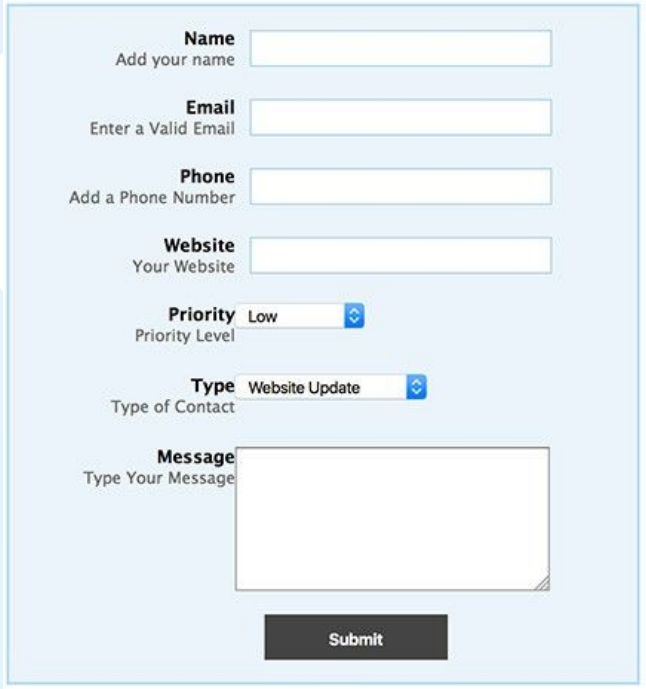
Make a Claim

iden3

OKIMS

Form request / Filling protocol

- Data (Example: Name, address, email, phone, picture, etc).
- Signature
- Cryptographic proofs



iden3

OKIMS

Identity naming (aliases)

- Binding names to identities with iden3 name resolver module (i.e.):

  jordi@iden3.io -> 0xC23a677...

- Notifications module to receive at wallet on protocol or workflow steps

iden3

OKIMS

Circom - Circomlib - SnarkJS

Circom: DSL Language to generate ZK circuits

Circomlib: Standard components for Circom language

- BabyJub EDDSA
- Pedersen/MiMC Hashes
- Sparse Merkle trees
- ...

Snarkjs: Independent zkSnarks implementation in javascript

- Browser ready

iden3

OKIMS

Anonymous Logins

- Any employee can login to a platform without revealing which employee you are.

- Only a user is allowed per employee. (Nullifier)

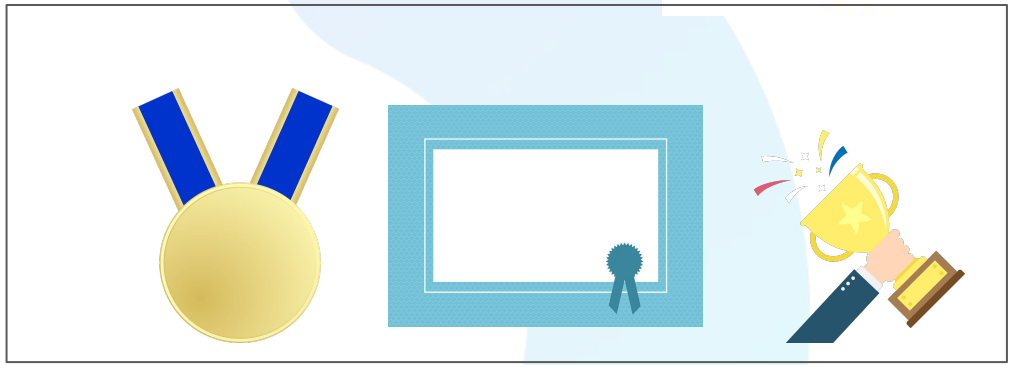
iden3

OKIMS

Reputation proof

- An identity can proof a given reputation calculated with a given algorithm, but does not need to reveal the sources.

Reputation score= 25

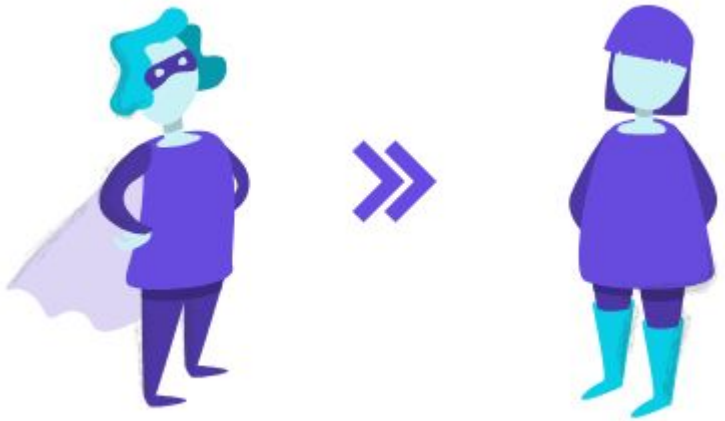


iden3

OKIMS

Cross identity proof

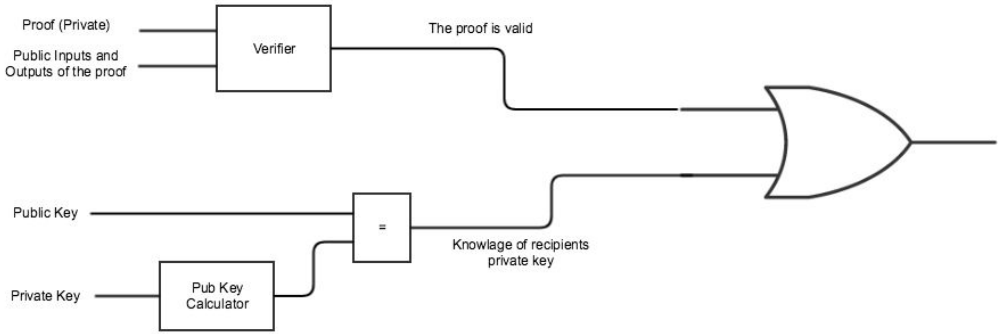
- You can prove that you hold other identities with some claims (reputation).



iden3

OKIMS

Privacy:
Non-reusable proofs



- A received proof is **not valid** to send to a **third** identity.
- To prove A, you create a new proof A' that is valid either if A is valid OR you know the private key of the recipient.

iden3

OKIMS

Thank you

- zkSNARKs: optimization work in progress
- New protocols for a complete solution
- Contributing to standardization of self-sovereign identity
- Working to enable universal login
- Wallet redesign for user adoption

@jbaylina

@davidsrz

iden3

Zero Knowledge Technologies State of the Art

OKIMS

Scheme	Runtime		Size		PQ?	Universal?	Untrusted setup?	Assumptions
	Prover	Verifier	CRS	Proof				
Hyrax	$d(hc + c\log c) + w$	$\ell + d(h + \log(hc))$	$\sqrt{w}$	$d\log(hc) + \sqrt{w}$	○	●	●	DL
ZK vSQL	$n\log(c)$	$\ell + d\,\mathrm{polylog}(n)$	$\log(n)$	$d\log(c)$	○	●	◐	q-type, KOE
Ligero	$n\log(n)$	$c\log(c) + h\log(h)$	0	$\sqrt{n}$	◐	●	●	CRHF
Bootle et al. [22]	n	n	0	$\sqrt{n}$	◐	●	●	CRHF
Baum et. al. [4]	$n\log(n)$	n	$\sqrt{n}$	$\sqrt{n\log(n)}$	◐	●	●	SIS
STARKs	$n\,\mathrm{polylog}(n)$	$\mathrm{polylog}(n)$	0	$\log^2(n)$	◐	●	●	CRHF
Aurora	$n\log(n)$	n	0	n	◐	●	●	CRHF
Bulletproofs	$n\log(n)$	$n\log(n)$	n	$\log(n)$	○	●	●	DL
SNARKs	$n\log(n)$	ℓ	n	1	○	○	○	q-type, KOE
Groth et al. [44]	$n\log(n)$	ℓ	n^2	1	○	●	◐	q-type, KOE
Sonic	$n\log(n)$	ℓ	n	1	○	●	◐	AGM

Table 1: Asymptotic efficiency comparison of zero-knowledge proofs for arithmetic circuits. Here n is the number gates, d is the depth of the circuit, h is the width of the subcircuits, c is the number of copies of the subcircuits, ℓ is the size of the instance, and w is the size of the witness. An empty circle denotes that the scheme does not have this property and a full circle denotes that the scheme does have this property. A half circle for post-quantum security denotes that it is feasibly post-quantum secure but that there is no proof. A half circle for untrusted setup denotes that the scheme is updatable. DL stands for discrete log, CRHF stands for collision-resistant hash functions, KOE stands for knowledge-of-exponent, and AGM stands for algebraic group model.

Source: [Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updateable Structured Reference Strings]

iden3