

# Shielder.js

**Aleph Zero**

Developed by Degens Alliance



# Table of Contents

0

## Introduction

What is Aleph Zero  
**Shielder** ?

1

## The Problem

What makes using  
**Shielder** difficult ?

2

## Solution

**Shielder.js** as an answer  
to the UX issues

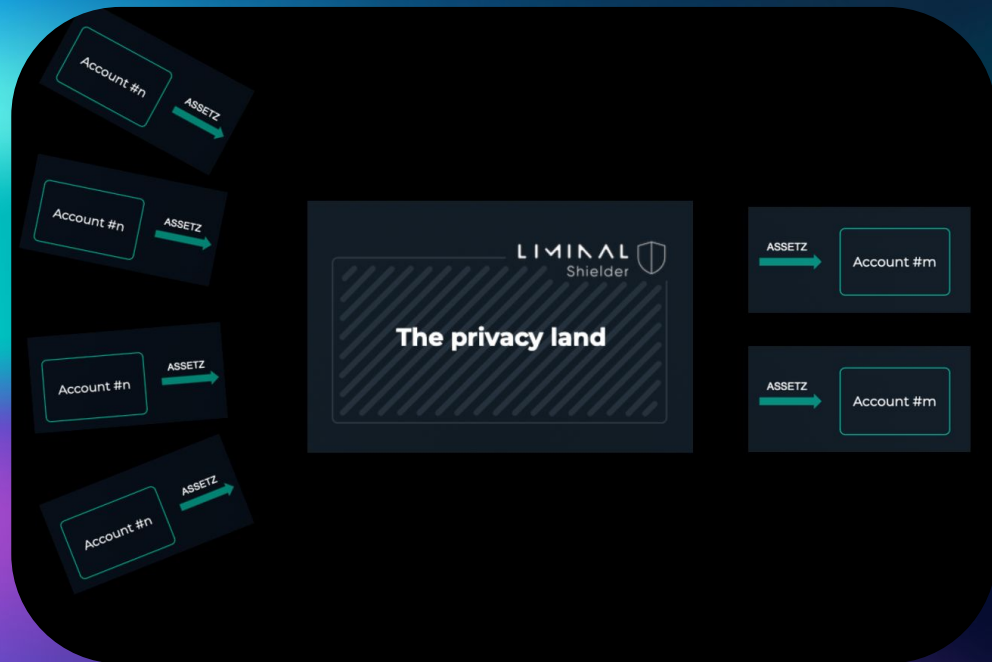
3

## How it works?

A brief intro to  
**Shielder.js** SDK



# Aleph Zero Shielder



# Liminal Shielder Private Transfers

Consist of zk-SNARKs proofs generator, Shielder Smart Contract, and custom Aleph Zero Node Extension with verifier



# The Problem

## UX

# Shielder CLI

```
shielder > ① README.md > ② # Shielder > ③ ## Usage > ④ ### Depositing assets

24
25 After a few minutes and several screens of logs, you should have everything installed. Note that as a last step,
26 the script will ask you to provide a password to encrypt the shielder state. If you are just experimenting on
27 your local machine, you can just press 'Enter' to use an empty password.
28
29 maciejzelazczyk, last month | 2 authors (piotrMocz and others)
30 ## Usage
31
32 First of all, it's worth noting that we already have pretty much everything set up for us:
33 * two PSP22 token contract instances (you can learn their addresses by inspecting the 'deploy/accounts.json'
34 file: they will come in handy!)
35 * a Shielder contract instance (its address is also a part of 'deploy/accounts.json')
36 * two accounts: Damian and Hans, with addresses '5D34dL5prEuaGNQtPP23yN5Y6BnkfXunXXXz6fo7ZJbLwRRH' and
37 '5GBNewRhZc2jXu7D55rBimKYDk8PGk8itRYFTpF8R3JLK65o', respectively (inspect the 'deploy/deploy.sh' if the ones
38 pasted here don't work for you).
39 * there's also a third account, Alice, which has admin rights. That said, for the purpose of this tutorial we'll
40 stick to using the regular accounts.
41
42 piotrMocz, 2 months ago | 1 author (piotrMocz)
43 ## Showing assets

44 We can view what we've deposited inside the Shielder contract (however, note that this gives you virtually no
45 information about who actually put what there). In order to do that, we will need to use the 'shielder-cli'. The
46 easiest way to obtain it is to get it from the 'cli/target/release/' folder, like so:
47 ```bash
48 cd cli
49 ./target/release/shielder-cli show-assets
50 ```
51
52 The output is not exactly fascinating right now, because we haven't deposited any assets yet. It will contain a
53 line like:
54 ...
```

```
PROBLEMS 32 OUTPUT DEBUG CONSOLE TERMINAL

• pawel@Pawels-MacBook-Air cli % cargo run show-assets
  Finished dev [unoptimized + debuginfo] target(s) in 1.47s
  Running `target/debug/shielder-cli show-assets`
  > Password (for encrypting local state): *****
  2023-05-21T20:49:16.684836Z INFO File with state was found. Reading the state from "/Users/pawel/.shielder-state".
  2023-05-21T20:49:16.878917Z INFO assets=[]
  pawel@Pawels-MacBook-Air cli %
```

# Current **Shielder** Issues

- Interaction only from the Command Line Interface
- Out-of-the-box it supports only 3 hard-coded accounts
- Other Accounts can be linked by passing seed phrase
- Necessity to download and compile large application locally
- Requires some degree of technical knowledge and tools, to compile and run Shielder Application
- Takes even up to a few hours to compile all of the components (on Macs\*)

2

## Shielder.js as a solution



# Overview of Shielder.js



## WASM

Moves zk-SNARK proofs generation to WASM module



## Easy to Use

Can be used entirely in the web app, no need to download, compile, nor run CLI



## Polkadot.js Compatible

Any account from Polkadot.js Wallet can be used

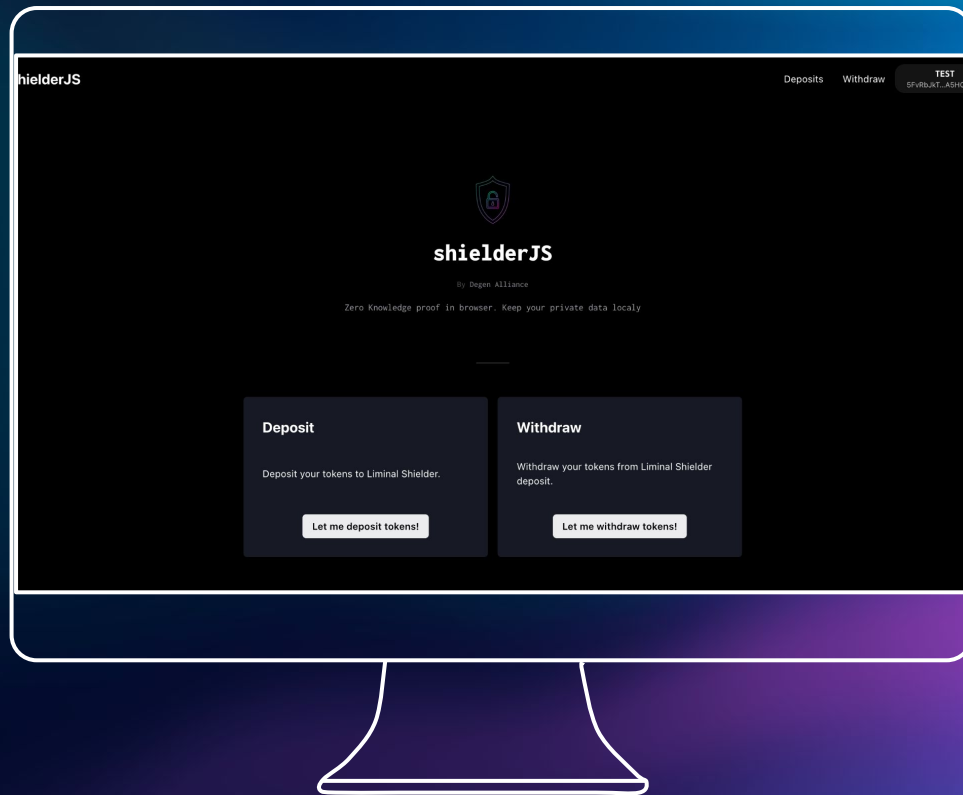


## SDK

NPM Bundle available for building custom applications with Shielder.js

# Shielder.js in action

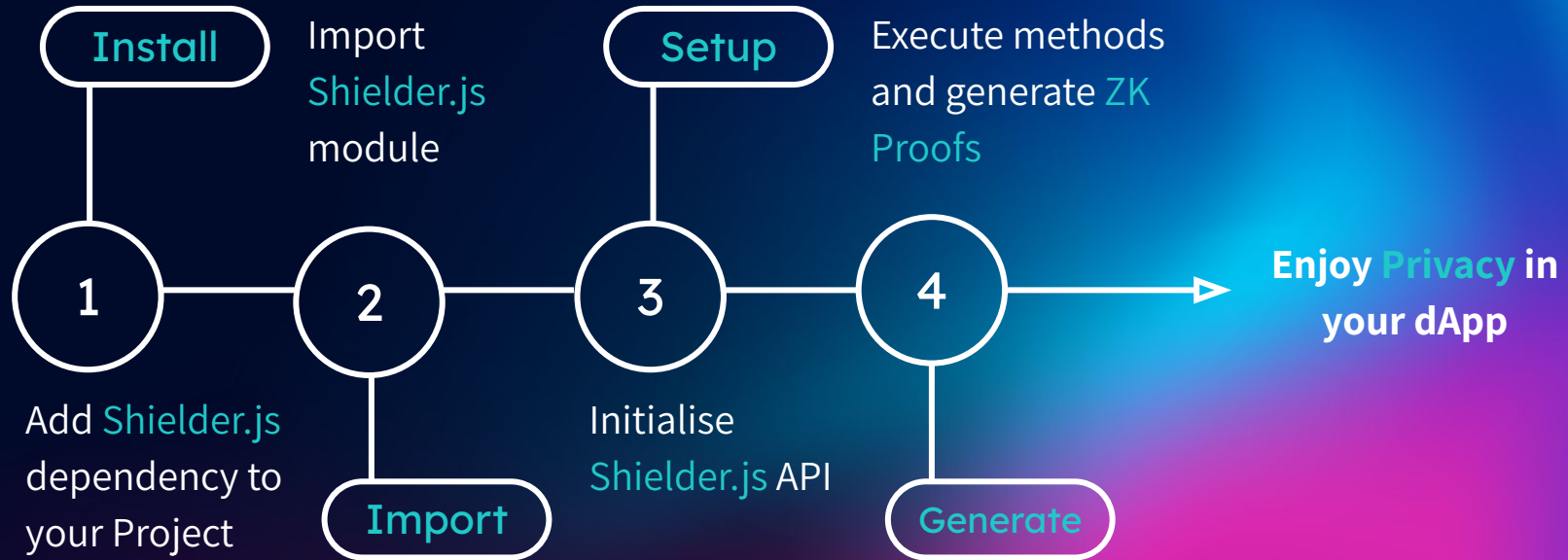
Shielder.js can be used to build dapps that leverage Liminal Shielder, without a need to get out of the Web Browser





**How it  
Works?**

# Shielder.js SDK



# The Team



**Przemek  
Paczosi**

Experienced front-end developer, quality engineer, and lead, with a strong background of software development process, and blockchain technologies.



**Jan  
Kuczma**

An Alumni of the University of Sussex with Bachelor's at Computer Sciences with focus on AI and Computer Architectures.



**Paweł  
Kowalewski**

Software Engineer, with experience in several blockchain technologies including projects based on Lightning Network.



**Krzysiek  
Kuczma**

Software engineer with 5+ year hands-on experience, specializing in Front-end, with some knowledge in nodejs based Backend technologies. Knowledgeable about cloud.

# Thank You



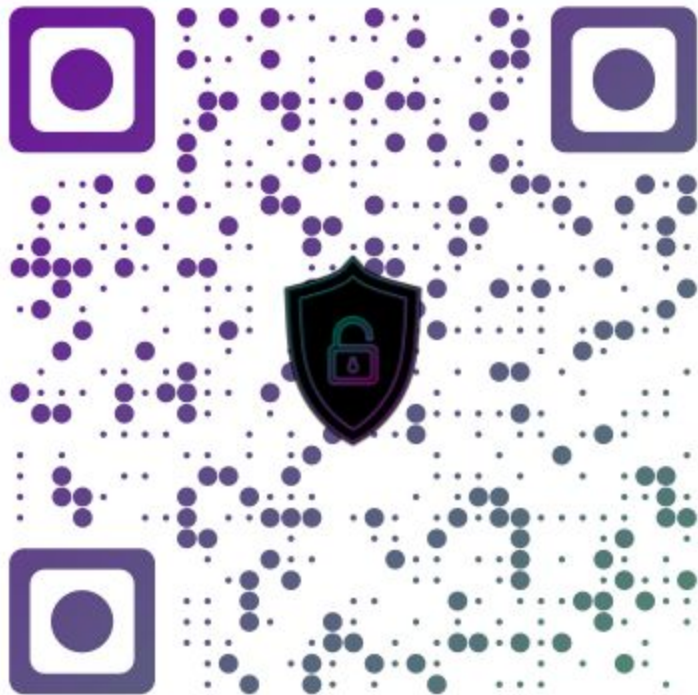
Contact :

@CirstosKrzyS

@kodziak

@JanTheJug

@Paweldev



GitHub Repos