ZK Email: Zuthailand Workshop

Aayush Gupta,
Sora Suegami
with Aditya Bisht, Dimitri Dumonet, Saul Garcia, John Guilding, Wataru Shinohara, Shreyas Londhe, Shubham Gupta, Prakhar Singh, Eesha Irfan

Roadmap

- Basics
- New Registry and SDK
- Account Recovery
- Devtooling

ZK Email: High Level Primitive

Emails are **signed** according to DKIM protocol

```
rsa_sign(sha256(from:<>, to:<>, subject:<>,
<body hash>,...), RSA private key of domain)
```

ZK Email proves the full signature within a proof, along with parsing and selective disclosure!

ZK Email proofs are:

private

Keep arbitrary data private

provenant

Verify the data from the web2 service's mail server directly

portable

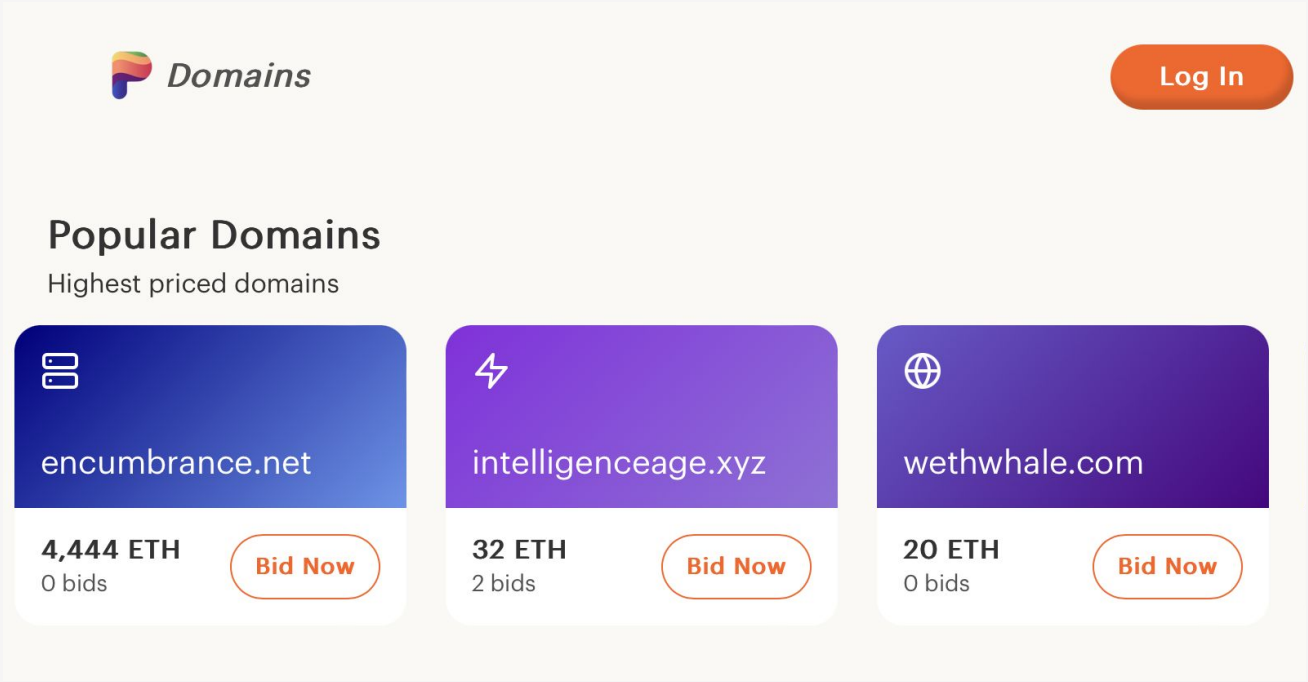
Move proofs onto chains

Whistleblowing

Decentralized Domain Marketplace

zkp2p.xyz

Prove you own a Namecheap transfer email, and build marketplace atop it



Roadmap

Basics

New Registry and SDK

Account Recovery

Devtooling

zk email proof infrastructure

Registry

Reuse proofs that other people have already defined

SDK

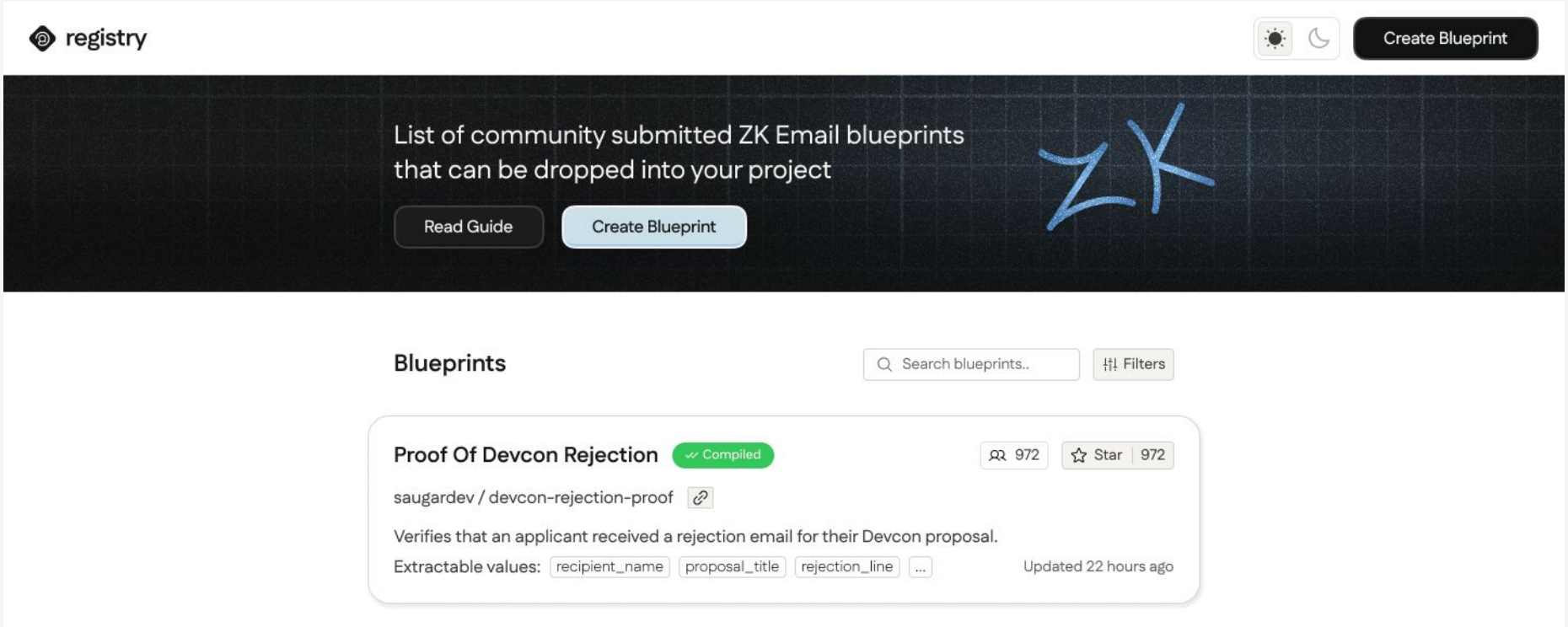
Easily use proofs from any proof system in your app

Apps

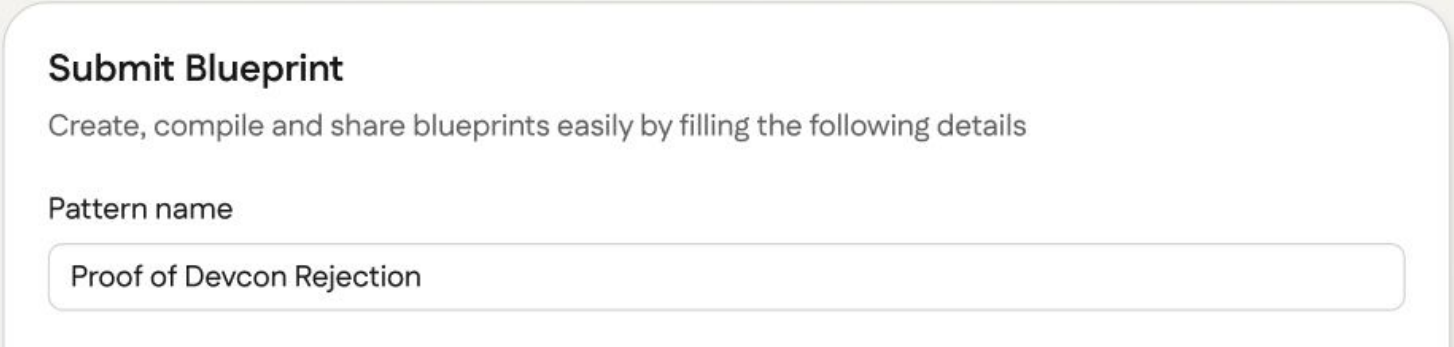
What are people building with this?

Don’t make app devs choose

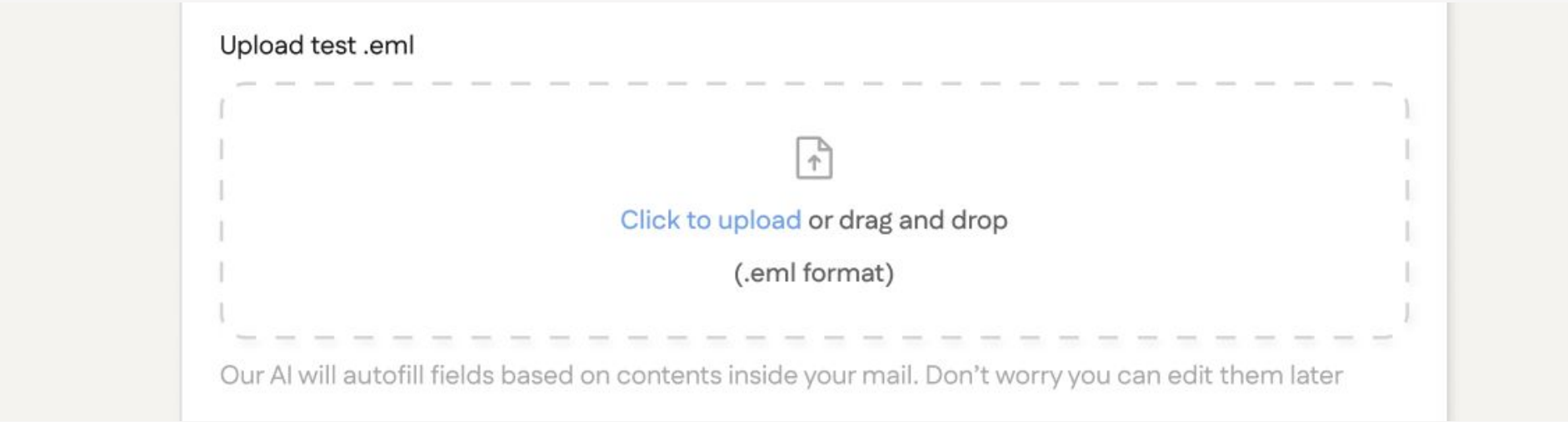
Registry


registry
Create Blueprint
List of community submitted ZK Email blueprints
that can be dropped into your project
Read Guide
Create Blueprint
ZK
Blueprints
Search blueprints..
Filters
Proof Of Devcon Rejection
Compiled
972
Star
972
saugardev / devcon-rejection-proof
Verifies that an applicant received a rejection email for their Devcon proposal.
Extractable values:
recipient_name
proposal_title
rejection_line
...
Updated 22 hours ago

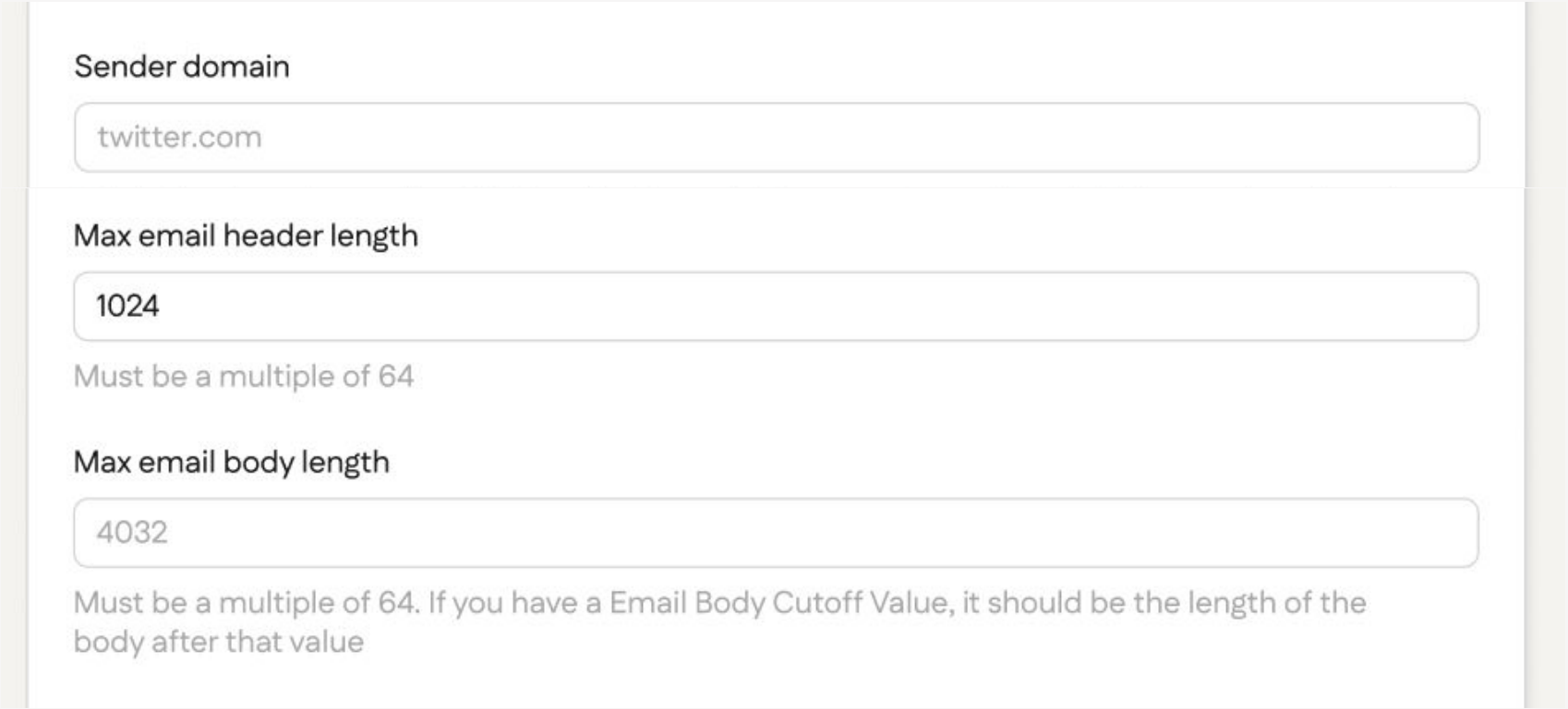
Registry: Create New Patterns in 5 Minutes



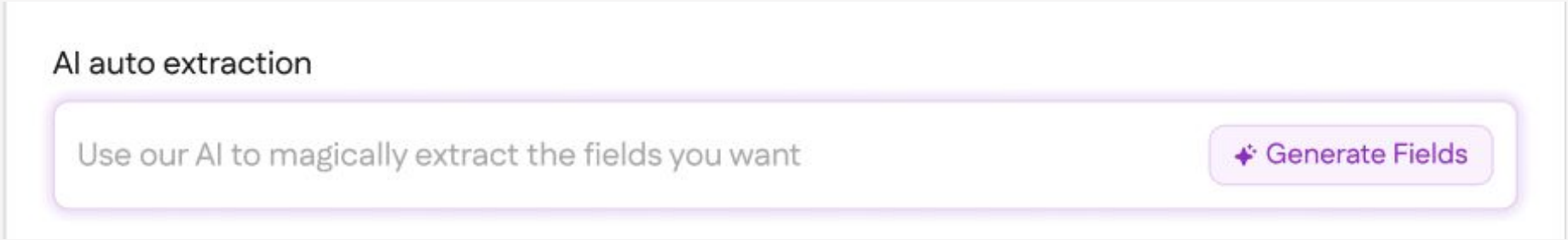
Registry: Add example email to parse



Registry: Auto-extract From Sample Email



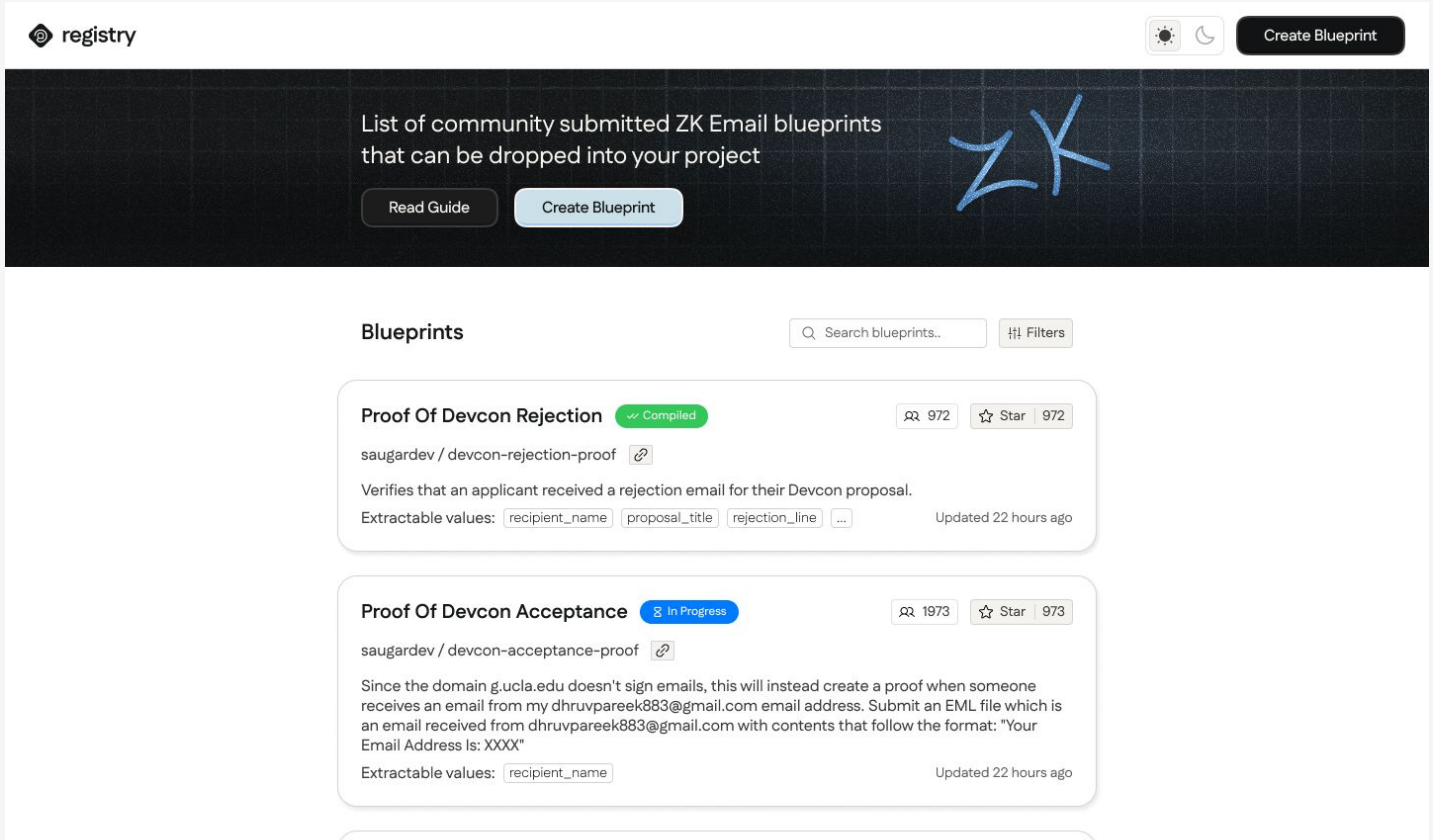
Registry: AI to Automatically Define Regexes



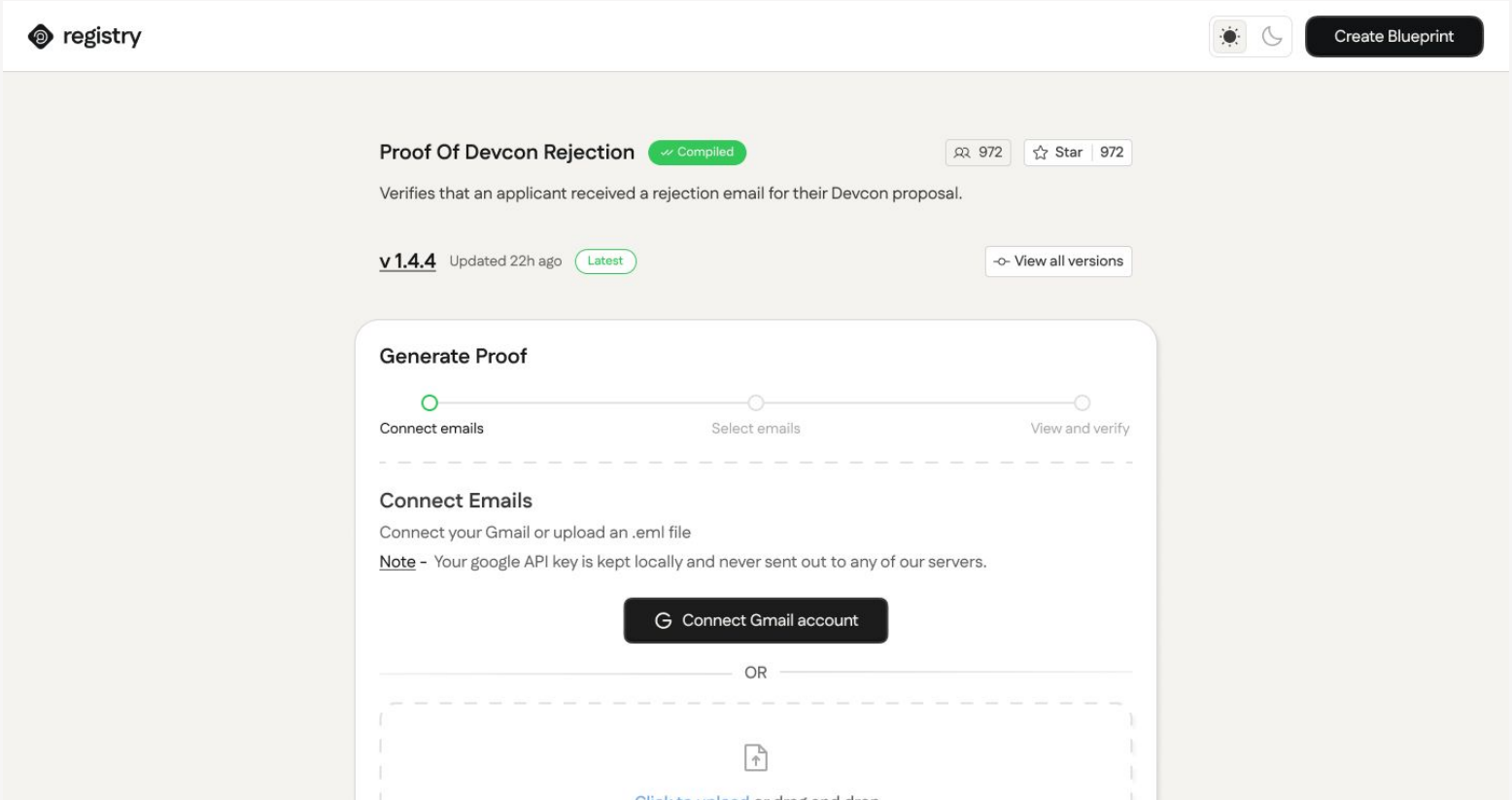
Registry: Final Config


```
    {
      "name": "proposal_title",
      "parts": [
        {
          "is_public": false,
          "regex_def": "your proposal \""
        },
        {
          "is_public": true,
          "regex_def": "[^\"]*"
        }
      ],
      "location": "body",
      "maxLength": 192
    },
    {
      "name": "rejection_line",
      "parts": [
        {
          "is_public": false,
          "regex_def": "we regret to inform you that "
        },
        {
          "is_public": true,
          "regex_def": "we were unable to accept this submission
        }
      ],
      "location": "body",
      "maxLength": 64
    }
  ],
  "version": "v2",
  "dkimSelector": "kh3h37a2jhlhfkth4f6smd3zlfz4sffz",
  "senderDomain": "devcon.org",
  "enableMasking": false,
  "externalInputs": [],
  "emailBodyMaxLength": 5120,
  "ignoreBodyHashCheck": false,
  "shaPrecomputeSelector": "padding: 20px 28px 0 28px;"
```

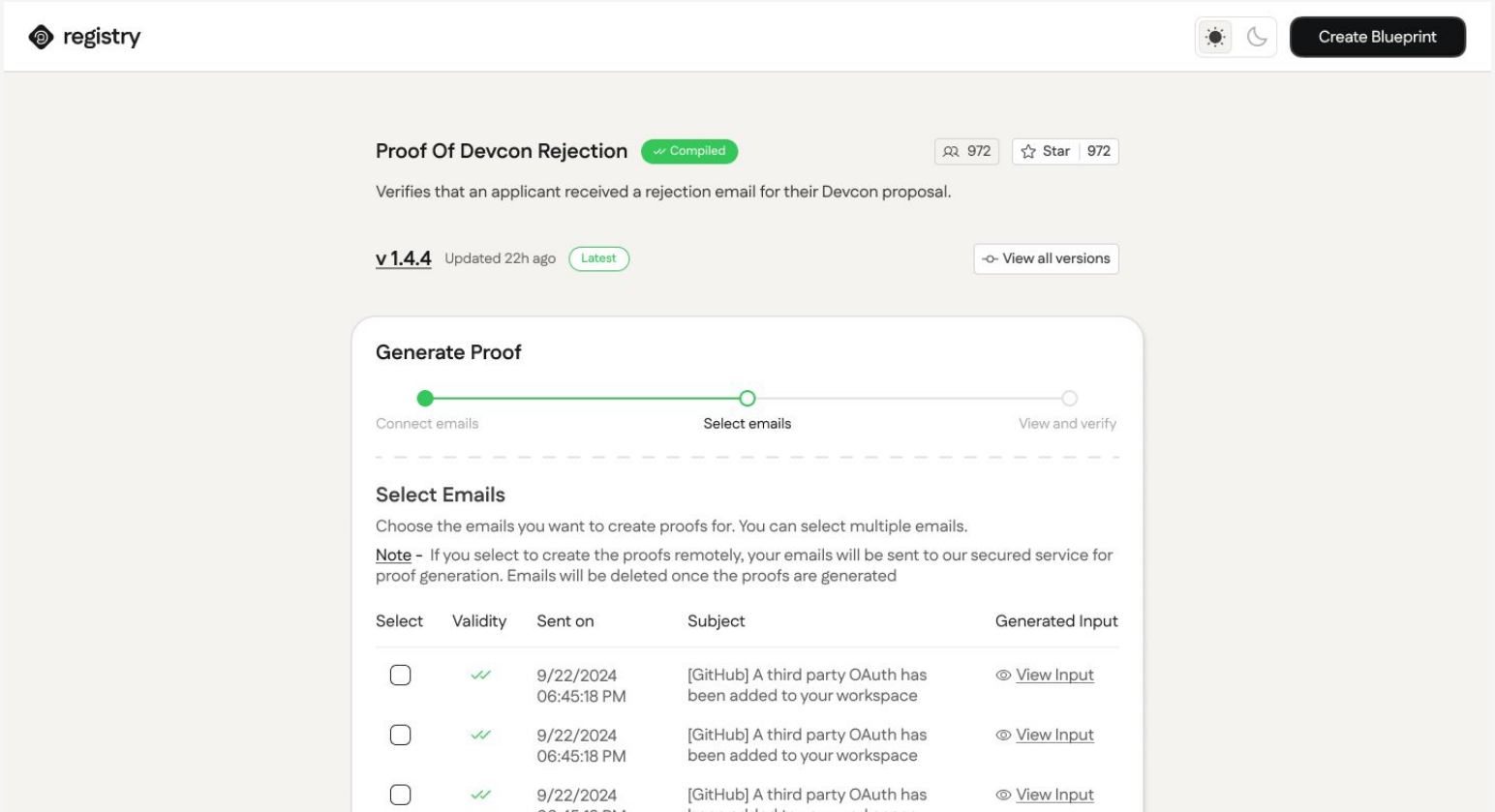
Registry: Easily View All Existing Proofs



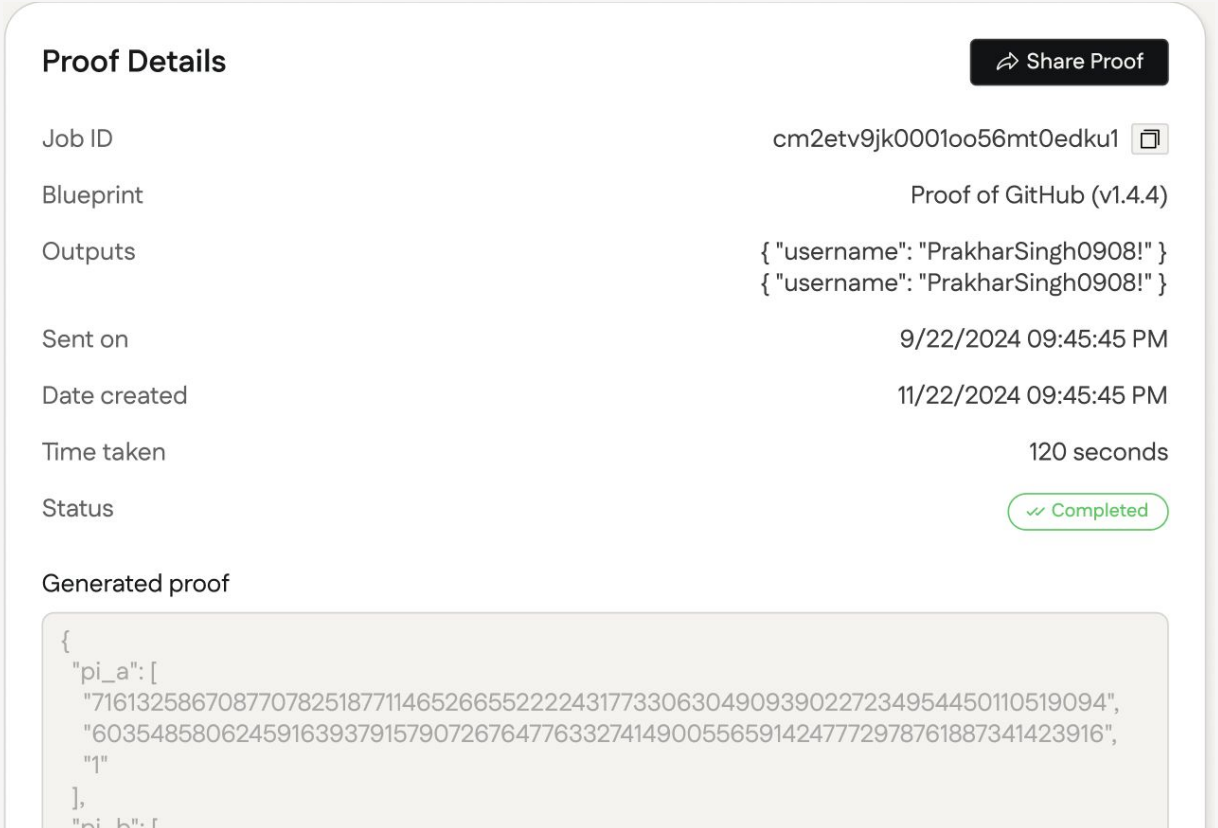
Registry = “Vercel” of ZK Email: Autodeploy interface



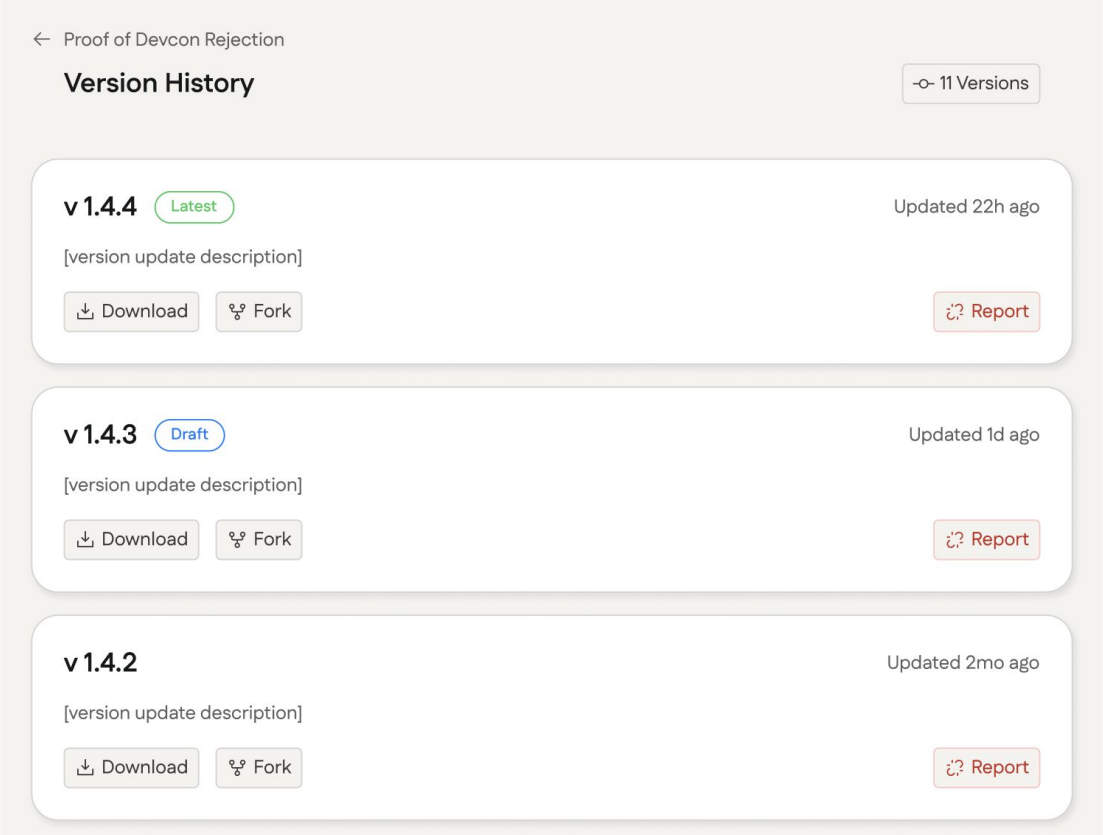
Registry: Automatically Filter Emails Client-side



Registry: Share finished proof



Registry = “Github” of ZK Email: Version History, Forks



registry

Login

List of community submitted ZK Email blueprints that can be dropped into your project

Read Guide

Blueprints

Search blueprints..

Filter and Sort

Proof of Devcon Acceptance

Compiled

saugardev/DevconAcceptance

Prove you got a Devcon acceptance email.

Extractable values: subject_congrats

Updated 1 days ago

Proof of Stripe payment

Compiled

saugardev/ProofStripePayment

Proof of Stripe payment

Extractable values: merchant order_id

Updated 1 days ago

registry.zk.email

SDK: Decides the proof system for you

```
import { createProver, parseEmail } from 'zk-email-sdk';

const blueprint = await sdk.getBlueprintById("saugardev/devcon-rejection");
const prover = blueprint.createProver({isLocal: false});
```

Proof of Pad Thai

receipt food grab thailand pad thai

johnguilding/proof-of-pad-thai

Last modified: 17/10/2024

Prove you're an enthusiast for Thailand's national dish by submitting a proof of Grab receipt which contains reference to a Pad Thai order

Proof of Devcon Rejection

devcon rejection talk

saugardev/devcon-rejection-proof

Last modified: 18/10/2024

Verifies that an applicant received a rejection email for their Devcon proposal. Extractable values: recipient_name, proposal_title, rejection_line

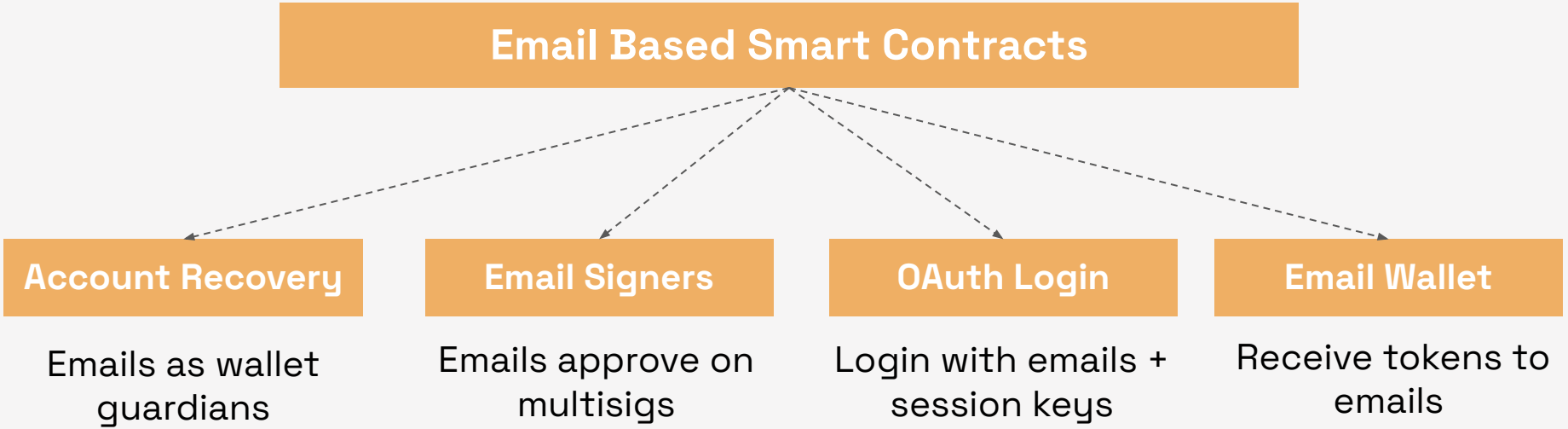
Try it out | Download Example Project | Download .zkey | View Parameters

Roadmap

- Proofs of Received Emails
- New Registry and SDK
- **Account Recovery**
- Devtooling

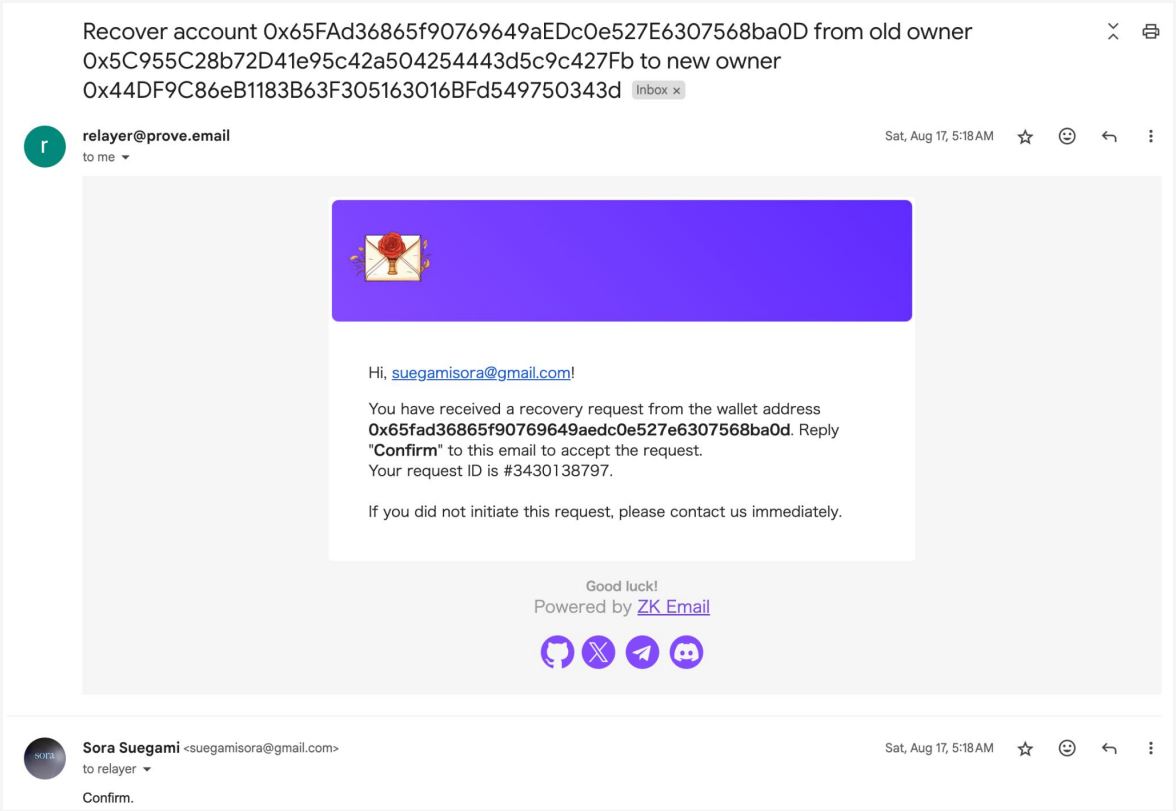
Email Triggered Transactions

ZK Email improves security & UX of smart accounts

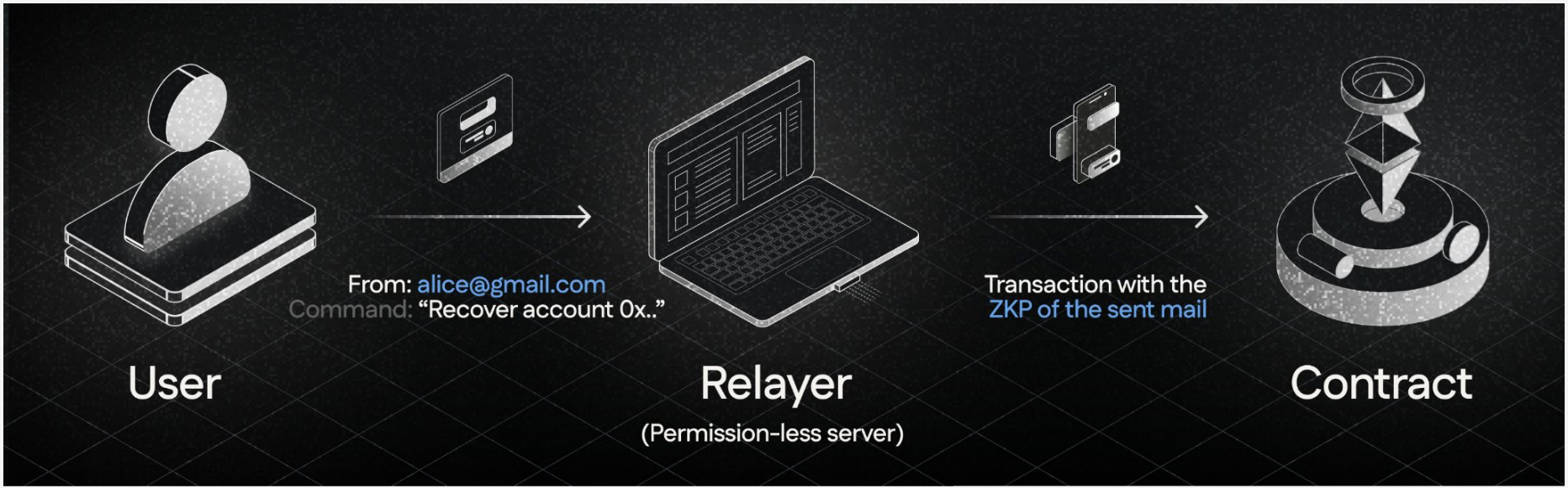


Send Commands to Smart Contracts through Emails

Users authorize transactions by simply replying to emails.

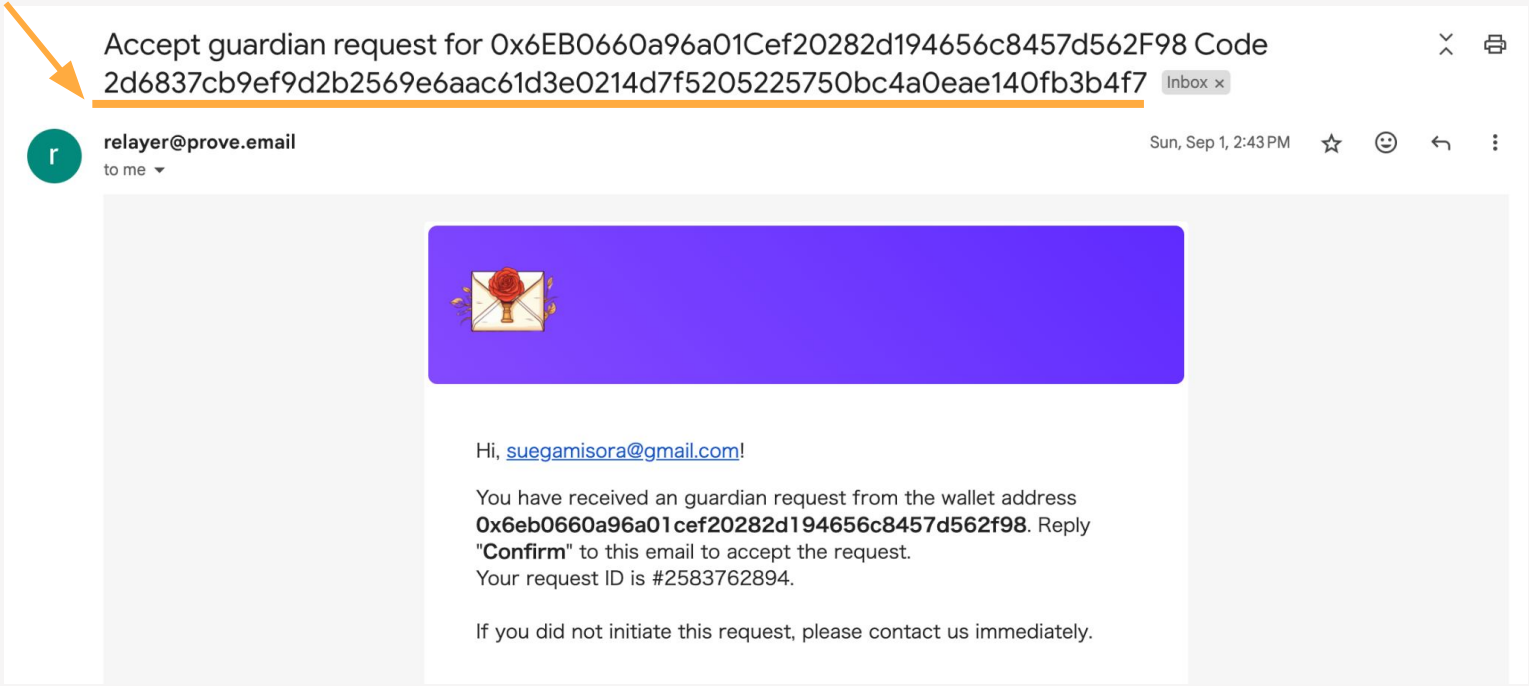


Send Commands to Smart Contracts through Emails



Send Commands to Smart Contracts through Emails

Account Code



Account Code Check
-> Privacy & Decentralization

- **hash(email address, code)** ⇒eth address: **email privacy on-chain**

- prove **availability** to user in email ⇒**access cannot be withheld**

- **relayer decentralization:** fault tolerant and censorship resistant

Email Replies or JWTs

zkemail

Hi,

You have received a guardian acceptance request ...

Reply "**Confirm**" to this email to accept the request. Your request ID is d116b3ba-dd09-45b2-9051-d212e581308b.

If you did not initiate this request, please contact us immediately.

Cheers,
The ZK Email Team

Powered by ZK Email

JWT-Wallet

Welcome to JWT-Wallet! Follow these steps to get started:

1. **Enter a command** in the input field below (e.g., "Send 0.12 ETH to 0x1234...")
2. The **Google Sign-In button** will become active once you've entered a command
3. **Click the Google Sign-In button** to authenticate and generate a JWT
4. **Check the console** for the decoded JWT information

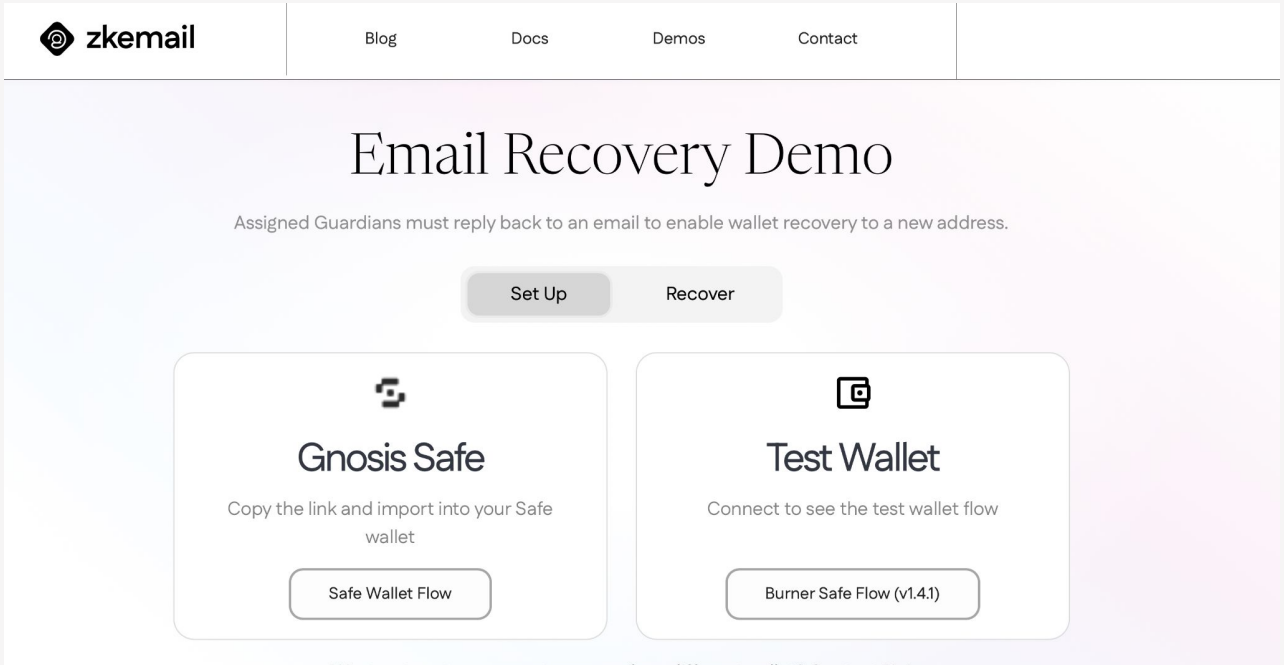
Enter your command here

Sign in with Google

JWT Generation > Proof Generation > Proof Complete > Submit to Contract

Email account recovery

Use email guardians to recover your account



Email Recovery: Social Recovery using Emails

Just like a bank account or paypal, anyone with an email address can help recover your account.

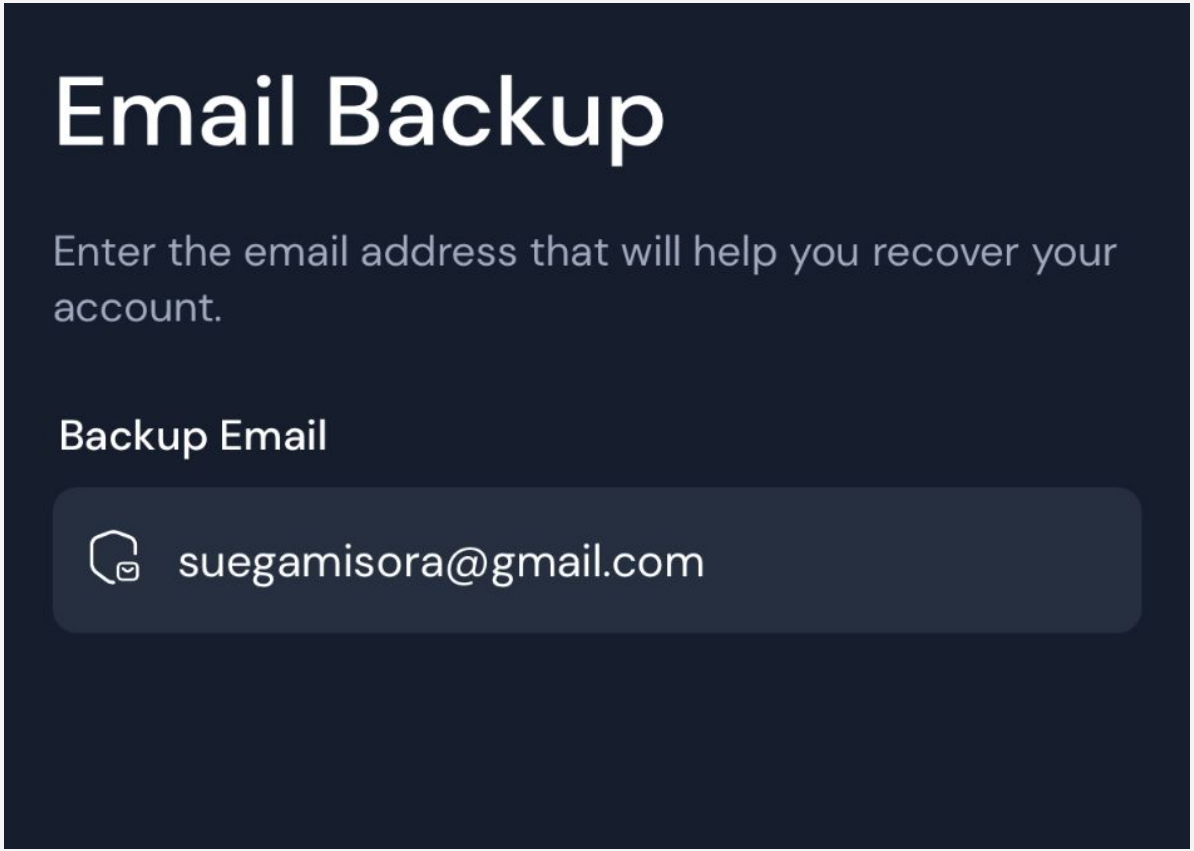
Email Recovery with Passkeys

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

How it works

1. Configure recovery settings



Clave

How it works

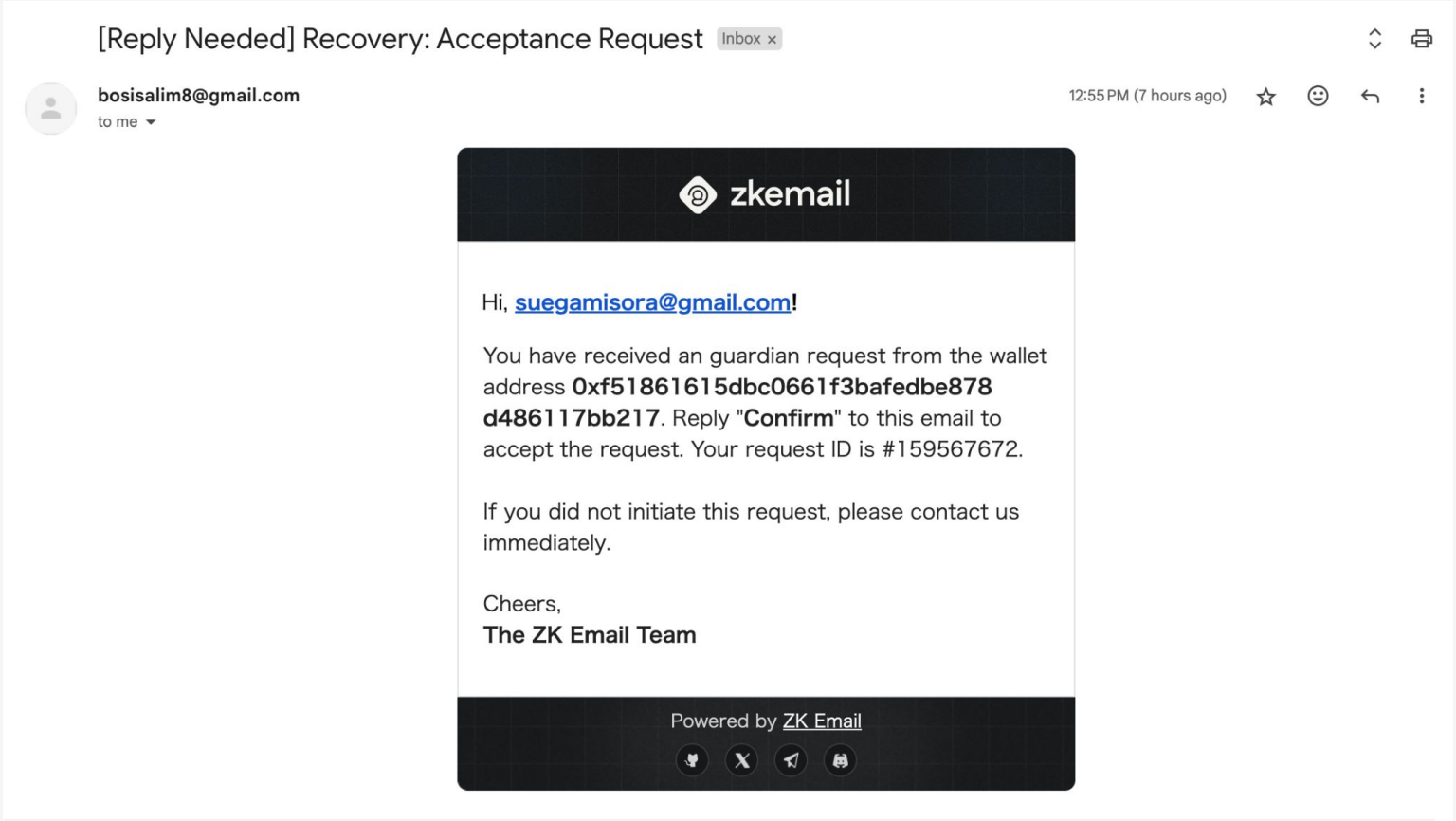
1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

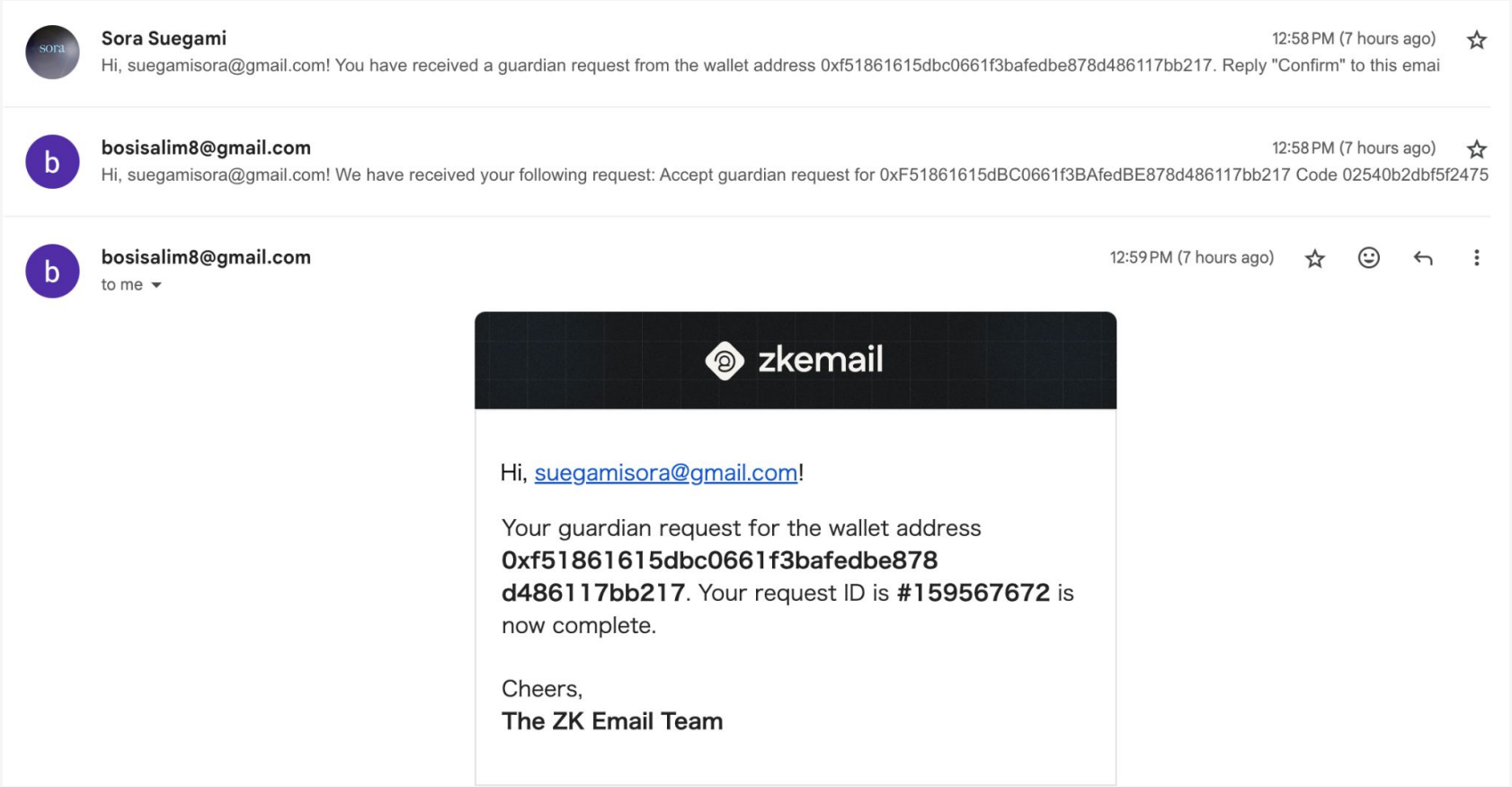
2. Accept guardian

Guardians explicitly accept being a guardian.
Prevents broken recovery setups

2. Accept Guardian



2. Accept Guardian



How it works

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

2. Accept guardian

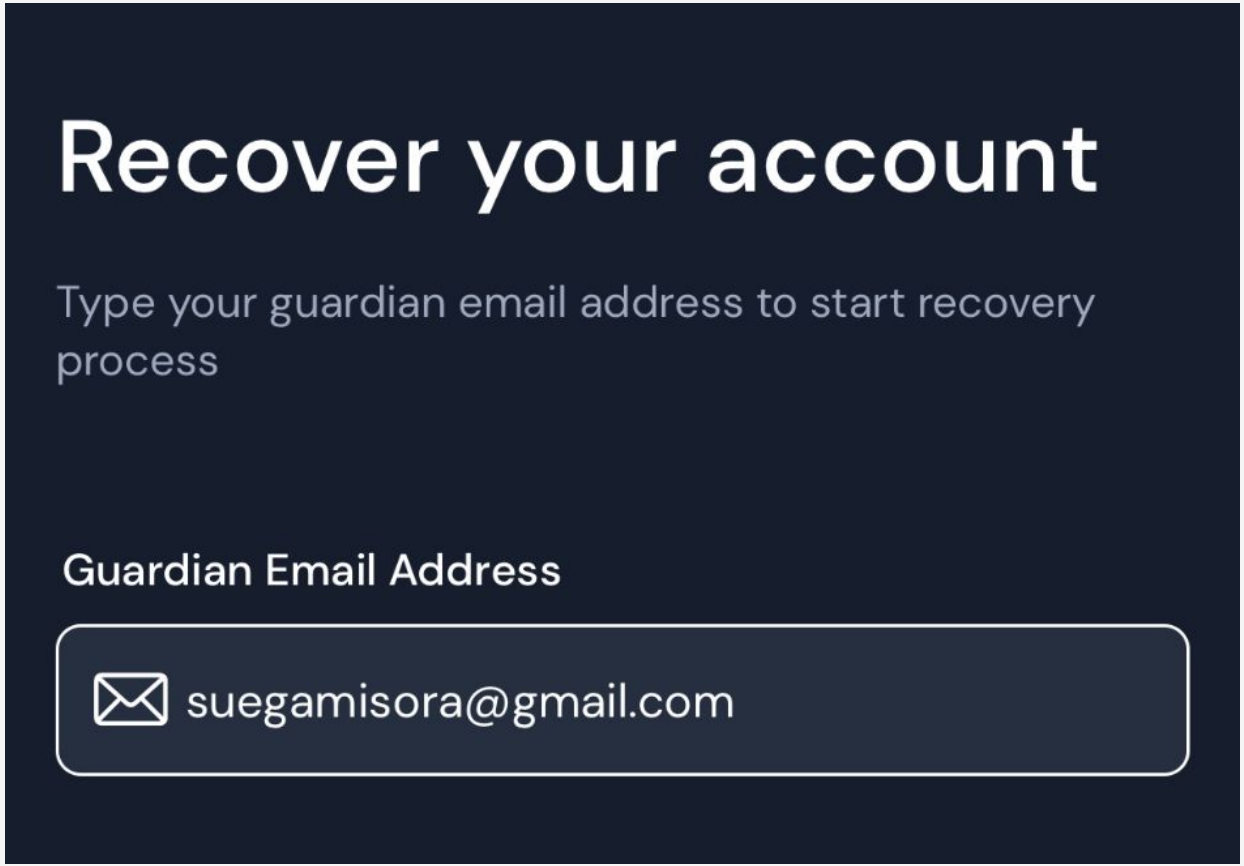
Guardians explicitly accept being a guardian.
Prevents broken recovery setups

2.5. You lose your key

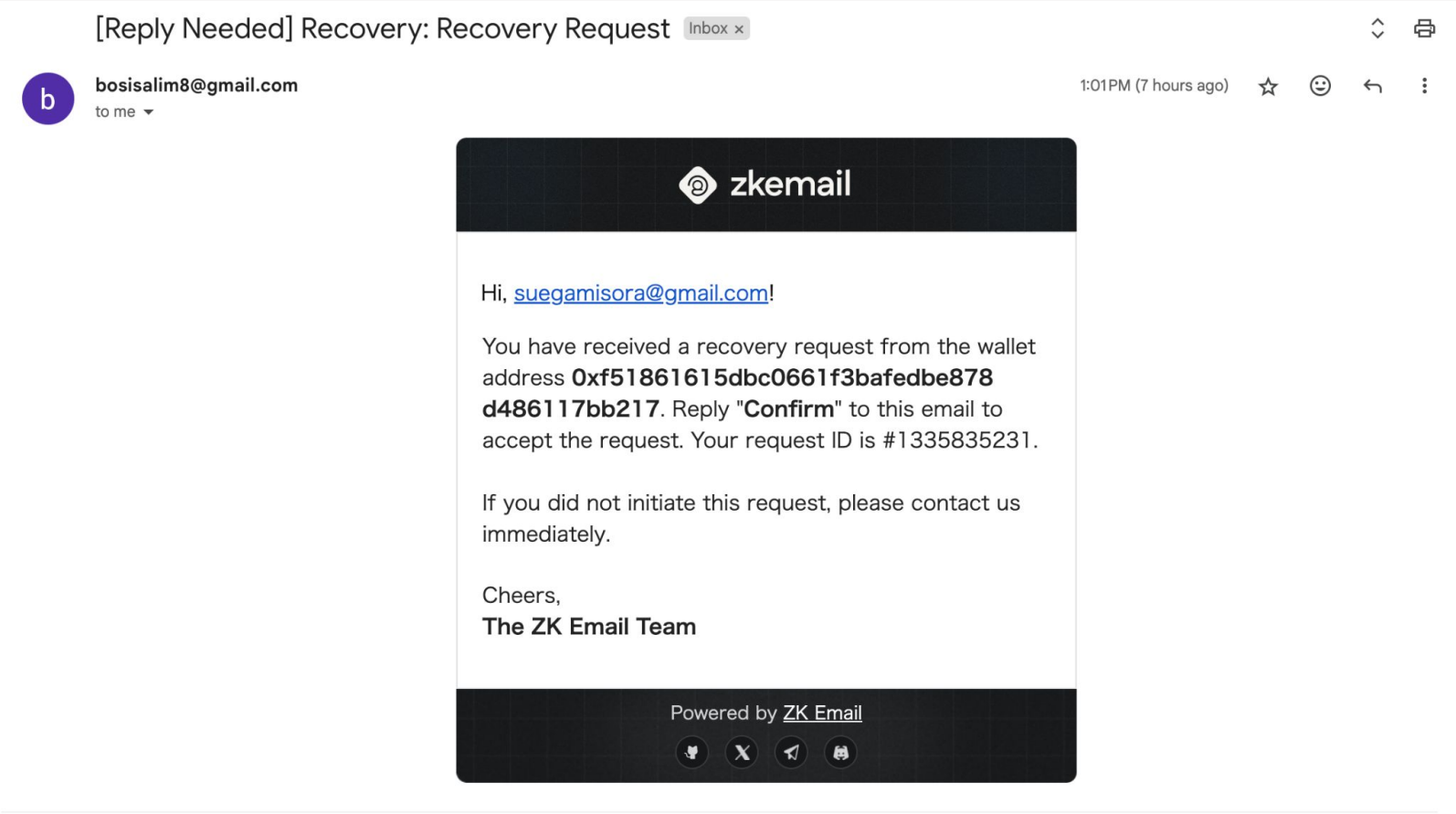
3. Process recovery

Guardians approve a recovery request. Can complete recovery once threshold is met

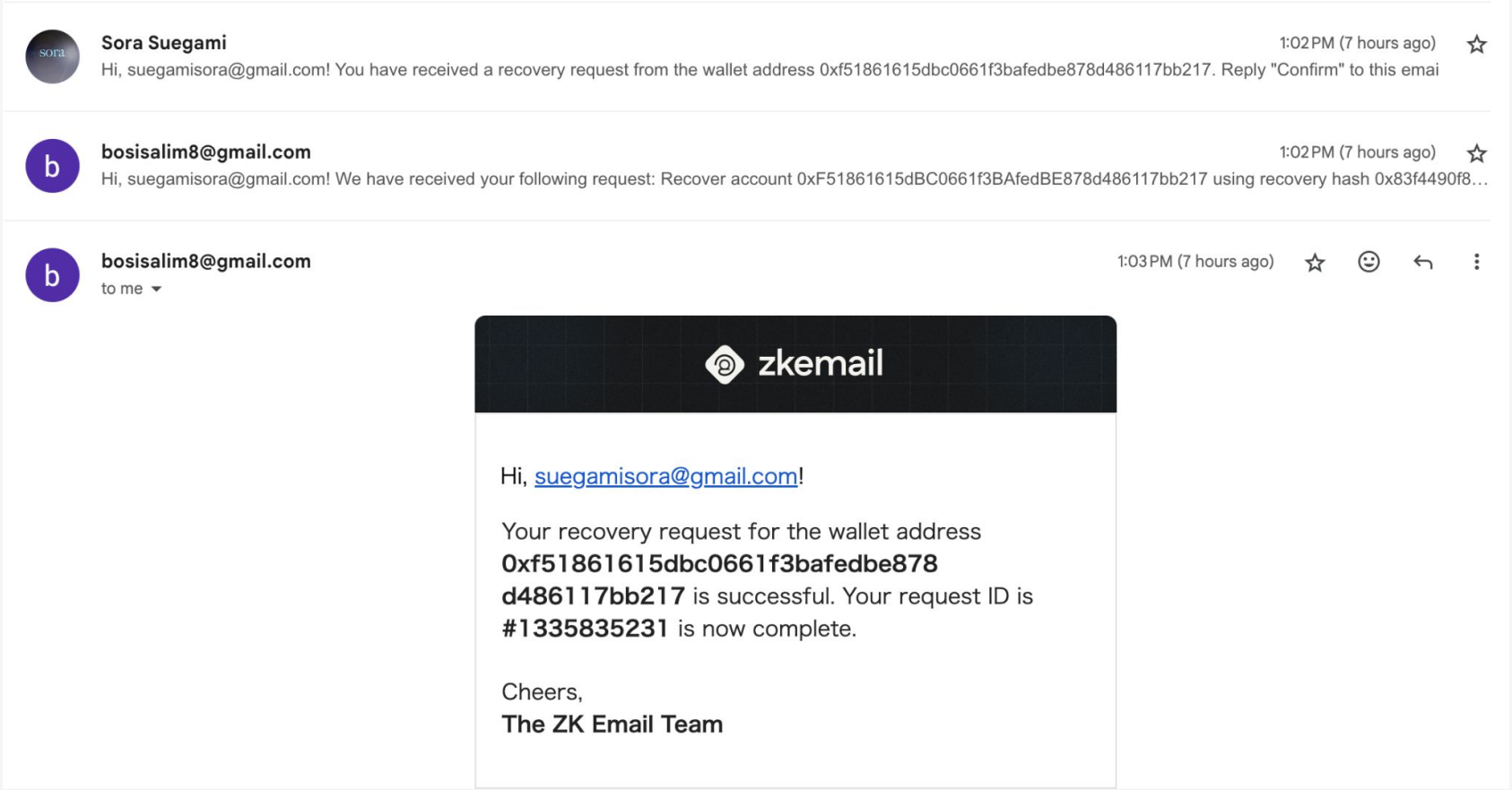
3. Process Recovery



3. Process Recovery



3. Process Recovery



How it works

1. Configure recovery settings

Configure necessary recovery settings such as guardians, threshold, recovery timelock etc

2. Accept guardian

Guardians explicitly accept being a guardian. Prevents broken recovery setups

2.5. You lose your key

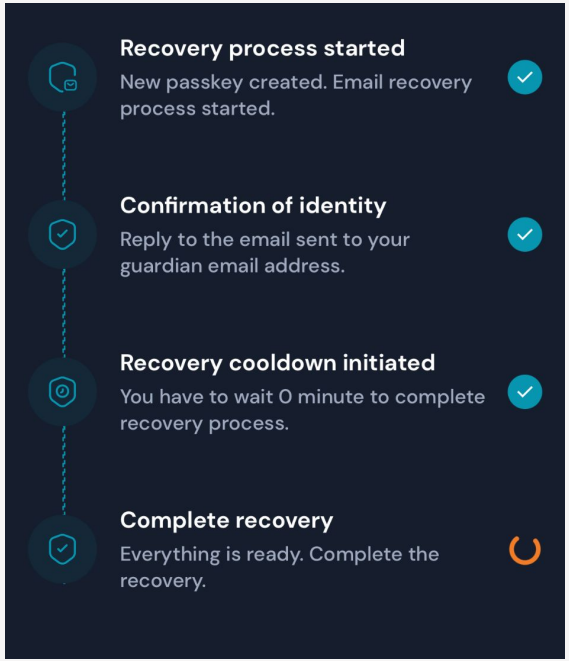
3. Process recovery

Guardians approve a recovery request. Can complete recovery once threshold is met

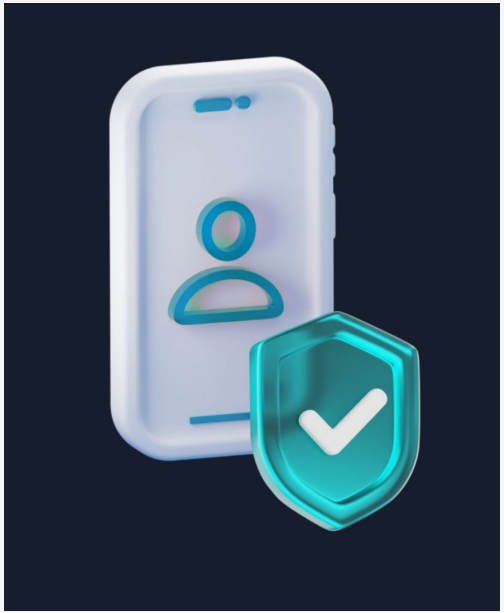
4. Complete recovery

Separate step to allow for recovery delays. Protects against malicious recovery attempts

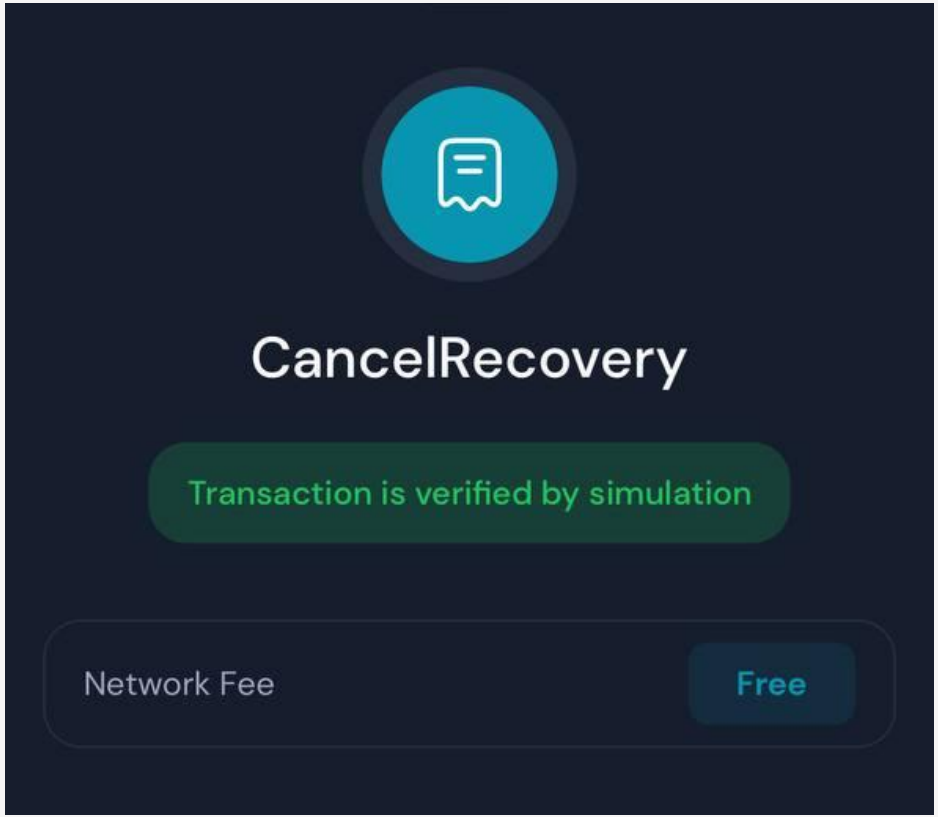
4. Complete Recovery



N minutes later...



Cancel recovery if the guardian is hacked



Roadmap

Basics

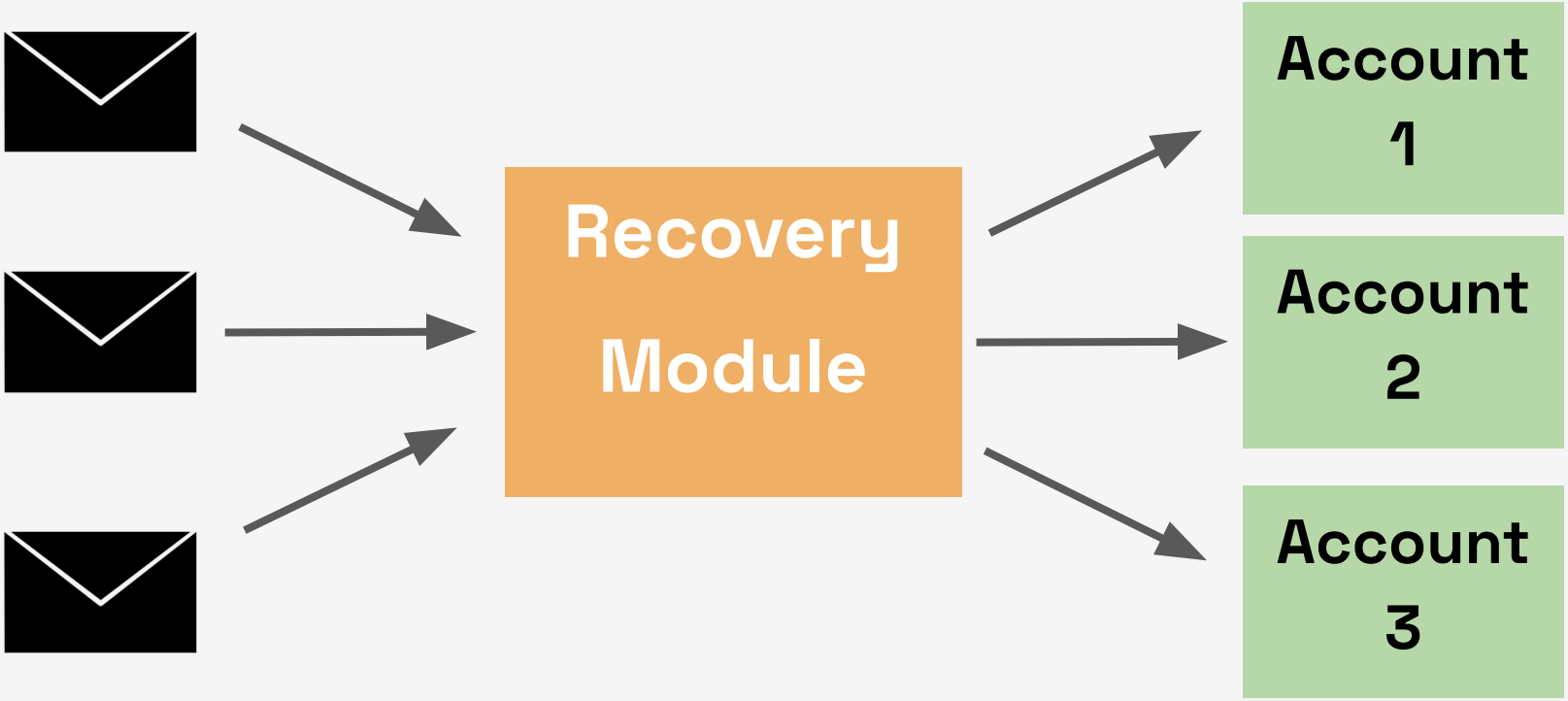
New Registry and SDK

Account Recovery

Devtooling

Email Recovery as ERC7579 Module

Recover existing Safes or any smart account via ERC7579 standard



Email Recovery as ERC7579 Module

Recover existing Safes or any smart account via ERC7579 standard

1. Configure recovery settings	Install the module on each account.
2. Accept guardian	Call `${relayerApiUrl}/acceptanceRequest`
3. Process recovery	Call `${relayerApiUrl}/recoveryRequest`
4. Complete recovery	Call `${relayerApiUrl}/completeRequest`

Add account recovery to any smart account

```
const userOpHash = await smartAccountClient.installModule({
  type: "executor",
  address: universalEmailRecoveryModuleAddress,
  context: moduleData,
});

const receipt = await pimlicoClient.waitForUserOperationReceipt({
  hash: userOpHash,
});
```

docs.zk.email

Email Tx Builder: Generic Solidity Proofs + Relayer

```solidity
function recoverySubjectTemplates() public pure returns
(string[][] memory) {
        string[][] memory templates = new string[][](1);
        templates[0] = new string[](11);
        templates[0][0] = "Recover";
        templates[0][1] = "account";
        templates[0][2] = "{ethAddr}";
        templates[0][3] = "to";
        templates[0][4] = "new";
        templates[0][5] = "owner";
        templates[0][6] = "{ethAddr}";
        templates[0][7] = "using";
        templates[0][8] = "recovery";
        templates[0][9] = "module";
        templates[0][10] = "{ethAddr}";
        return templates;
    }
```

- Example API call for `/submit`

2

```
{
  "contractAddress": "0xec3a1eAD94BDc7527Aa807B97Ff7E3A2cBCbC75f",
  "dkimContractAddress": "0x0a3921e8Bb2d682f92a39cB006b77AEc8939d1B9",
  "accountCode": "0x22a2d51a892f866cf3c6cc4e138ba87a8a5059a1d80dea5b8ee8232034a105
  "codeExistsInEmail": true,
  "functionAbi": abi
  "commandTemplate": "Emit string {string}",
  "commandParams": ["testing"],
  "templateId": "0x25d6c3eada7b2926c822bbfebfc3173123afb205cf093a8cae6622a56712f8a
  "remainingArgs": [
    { "Address": "0x9401296121FC9B78F84fc856B1F8dC88f4415B2e" },
    { "Uint": "0x0" }
  ],
  "emailAddress": "bisht.s.aditya@gmail.com",
  "subject": "Testing",
  "body": "Testing",
  "chain": "baseSepolia"
}
```

Email Transaction Builder

docs.zk.email/ > Email Tx Auth > Quickstart

Built with Email Transaction Builder

Account Recovery

Emails as wallet guardians

Email Signers

Emails approve on multisigs

OAuth Login

Login with emails + session keys

Email Wallet

Receive tokens to emails

Roadmap

- Basics
- New Registry and SDK
- Email Triggered Transactions
- Devtooling

Booths & Events

Museum Exhibit at Cursive

Nov 12-15

Cursive Connections Museum, Main Venue (SKNCC)

Interpretive exhibit representing ZK Email!

ZK Email Sessions at PSE Impact Booth

Nov 12, 14:00-17:00 and Nov 13, 11:00-14:00

Main Venue (SKNCC)

Come meet the team, solve challenges, and try out our new tech with the folks who built it!

ZK Email Impact Booth

Nov 14-15 all day

Main Venue (SKNCC)

Come meet the team, solve challenges, and try out our new tech with the folks who built it!

Discussion at AA Community Hub

Nov 15, 14:15-15:00

Main Venue (SKNCC)

Demo and discussion on ZK Email for Account Abstraction.

Talks & Panels

Mainstage Talk on ZK Email

Nov 13, 10:30 AM

Main Venue (SKNCC), Stage 3

All the ZK Email updates — community progress, fresh devtools, fast proofs, and new mainnet launches!

SDKs & DevTooling Workshop

Nov 14, 13:30-15:00

Main Venue (SKNCC), Classroom E

Build your own email proofs, email triggered transactions, and/or account recovery — Get a headstart on your hackathon projects!

Verifiable Data: No BS Debate

Nov 14, 4:15 PM - 4:45 PM

VLayer Panel, JW Marriott Hotel Bangkok

A debate on verifiable data.

ProgCrypto Lightning Talk

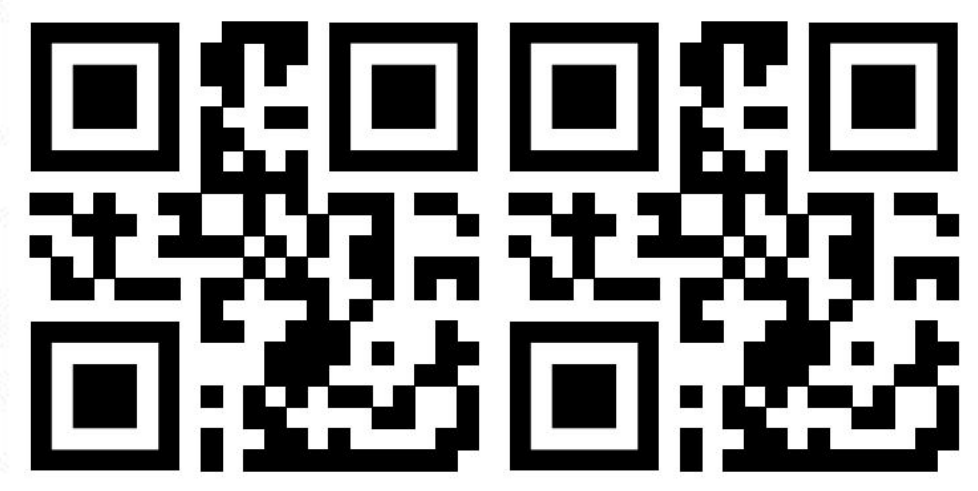
Nov 15, 12:30 PM - 2:00 PM

Progcrypto community led session, Main Venue (SKNCC), Breakout 2

A 7-9 minute lightning talk with ZK Email updates for ProgCrypto!

zkemail

DC7 SEA



t.me/zkemail

https://zk.email

docs.zk.email/

Slides

zkemail