



Concordium.com

Concordium @ techFiesta

VOTES



Hackathon
Kickoff Session

Doris Benda



Concordium.com

Concordium Blockchain



Facts and Stats (Mainnet)

2-4 s
Transaction
Finality



Low transaction fee
(€0.0001 Euro - simple transfer);
Pegged to the Euro

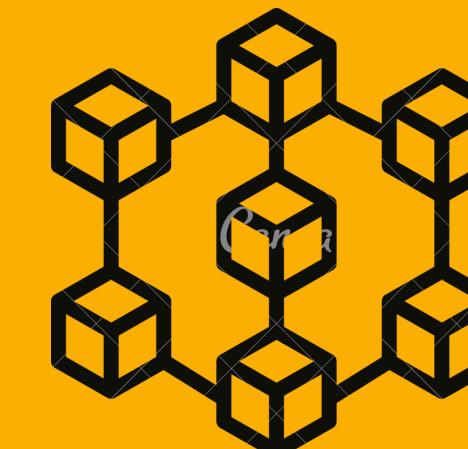


EuroE
StableCoin

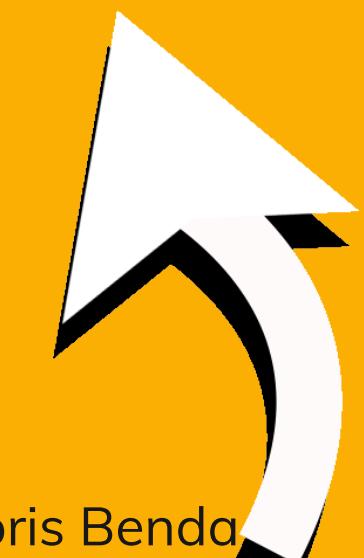


9,521 Smart Contracts;
99,199 Accounts

ConcordiumBFT



5-10 TPS;
87,502,892 Total
Transaction Processed





Ecosystem



[https://www.concordium.com/
ecosystem-partnerships](https://www.concordium.com/ecosystem-partnerships)



Blockchain + Identity =



“Concordium is a science-based, Proof-of-stake, Public, Permissionless Blockchain with Identity built into the core protocol layer ...”

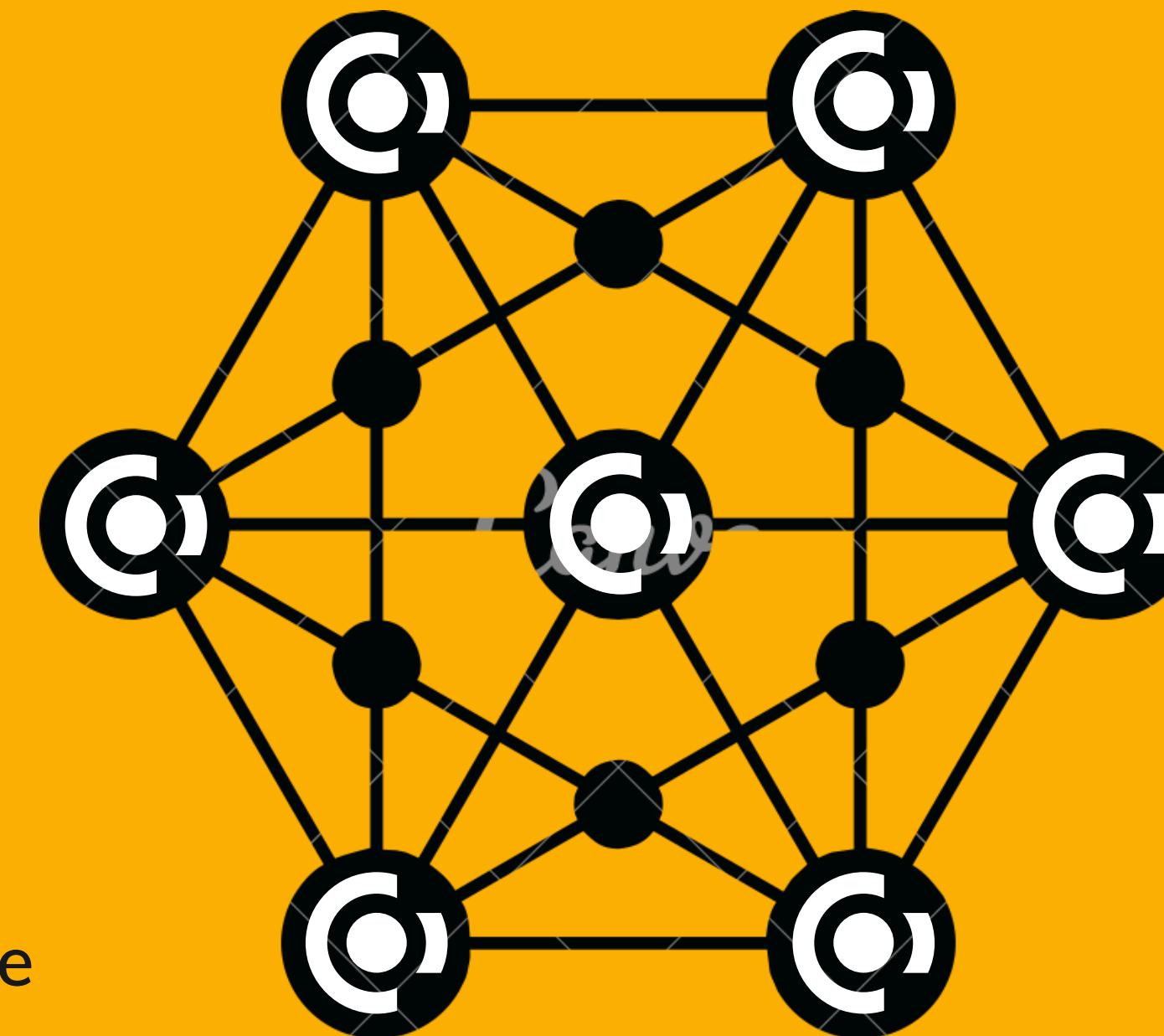


Blockchain + Identity =

Identity: Requires ID check
to open Accounts

ZK-Proofs: Verify information
about real-world identities

Public: Anyone can run a node



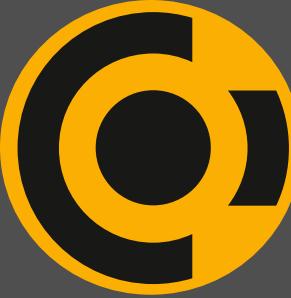
Rust: Smart
Contract Language

Science-based: Built
by leading scientists

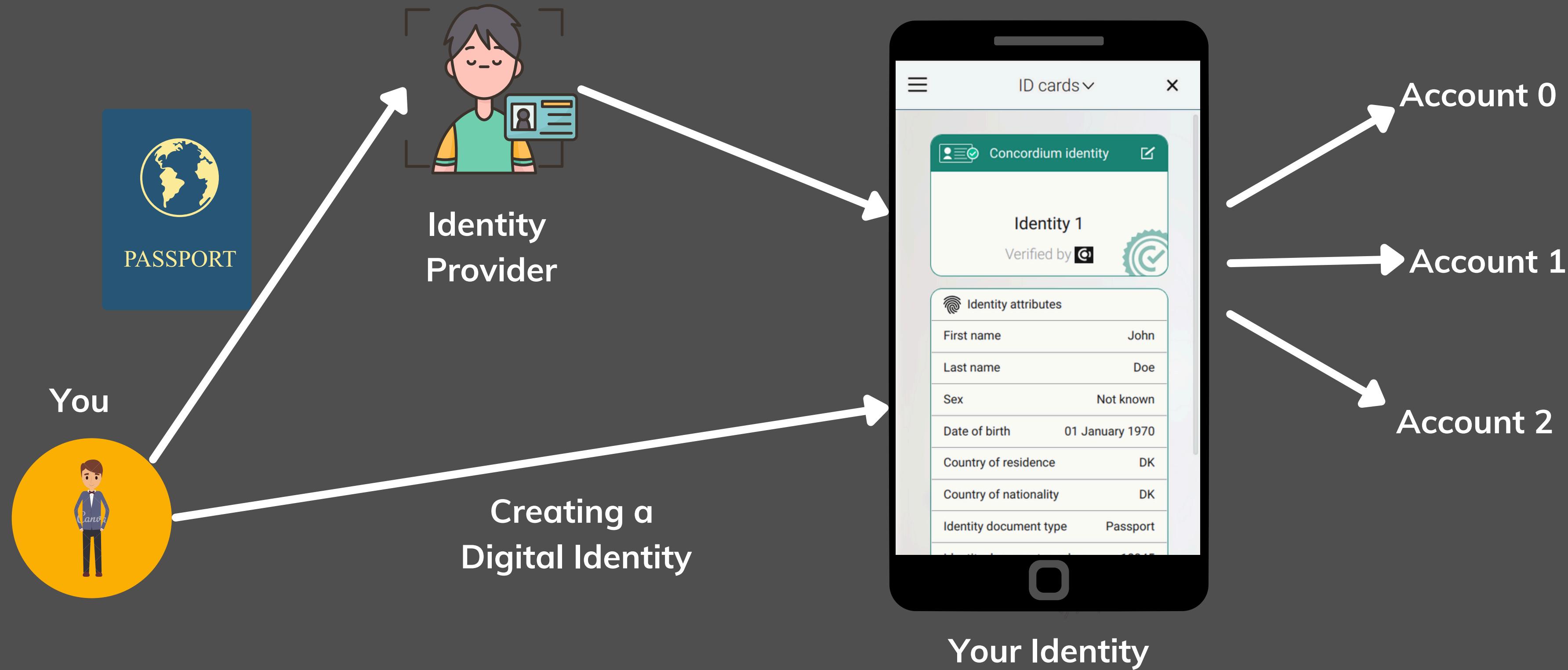
Proof-of-stake: Eco-friendly.
Weighted lottery tokenomics



Account Creation on Concordium



Concordium's ID-layer and Account Creation





Identity Disclosure Process



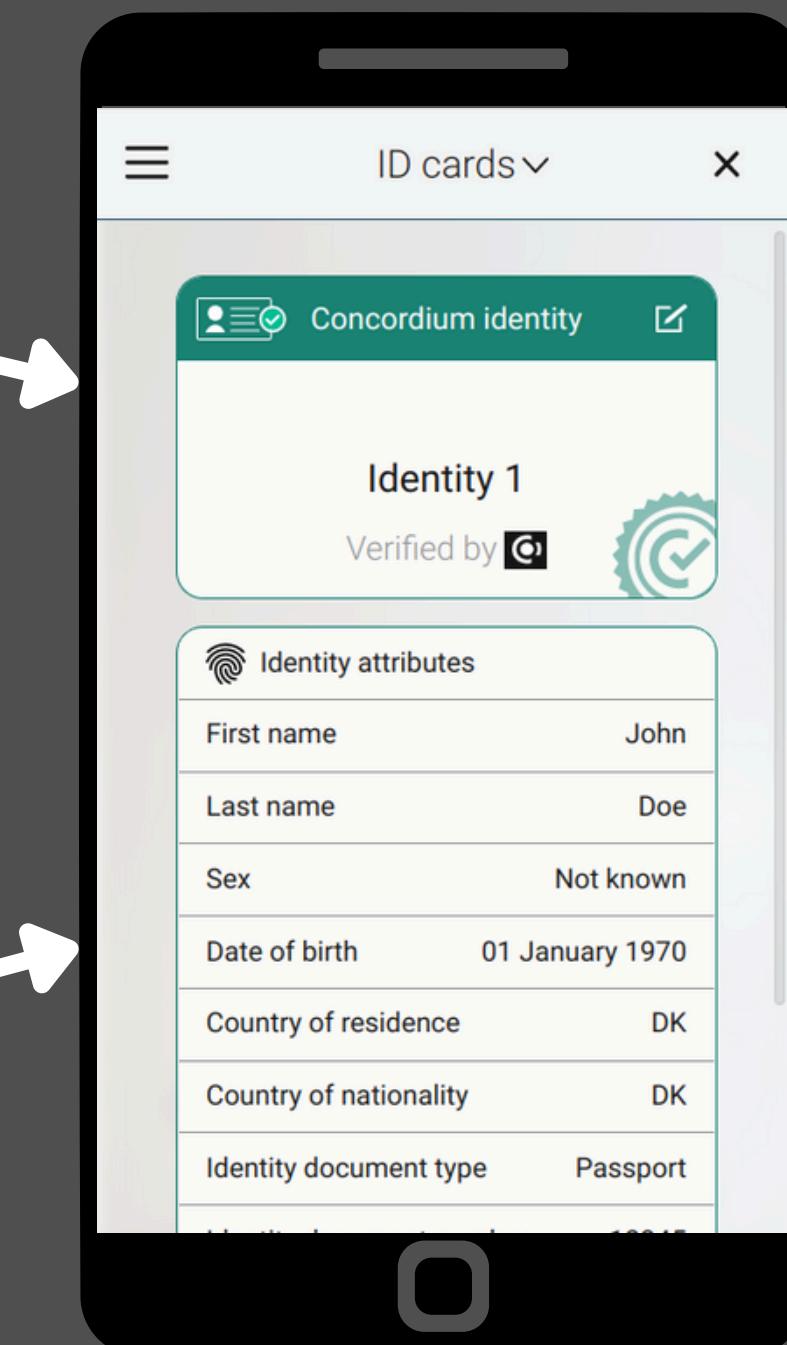
Identity Disclosure
Authorities



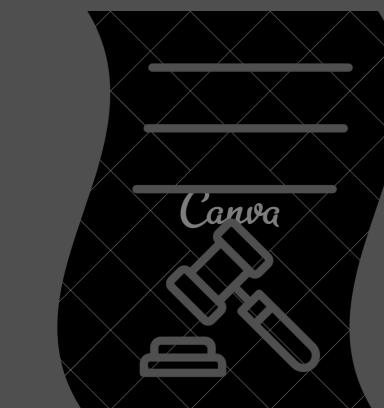
Identity Provider



Law
Enforcement



Your Identity



You
Legal
Warrant



Installing Prerequisites

- Rust programming language
 - <https://www.rust-lang.org/tools/install>
- Cargo Concordium (Tool to compile your smart contract)
 - <https://developer.concordium.software/en/mainnet/net/installation/downloads-testnet.html>
- Concordium Browser Wallet (or Mobile Wallet)
 - <https://developer.concordium.software/en/mainnet/net/installation/downloads.html#bw>



Concordium.com

Concordium Browser Wallet

- The browser wallet hosts the **private keys** corresponding to the accounts of the user and a link that points to a server that hosts a **Concordium node**.
- Tips:
 - Create an identity and then create an account.
 - Use the top identity provider on testnet:
 - (allows you to create any identity on testnet; no ID required)
 - Request funds from the testnet faucet inside the wallet.

The screenshot shows the 'ID cards' section of the wallet. It displays three identity provider options: 'Concordium testnet IP' (highlighted with a red box), 'Digital Trust Solutions TestNet', and 'Notabene (Staging)'. Below this, there's a summary of account details: Public balance total (C0.00), Public amount at disposal (C0.00), Stake / delegation amount (C0.00). A QR code is also present.



Let's download the testnet browser wallet:
<https://developer.concordium.software/en/mainnet/net/installation/downloads.html#bw>

The screenshot shows the 'Accounts' section of the wallet. It lists a single account: '3ed2...6dGc' (Identity 1), which is connected. The account has a public balance of C1,987.270061. Below the account summary, there's a transaction history table with columns for date, type, time, and fee. The history includes an update, a contract initiation, another contract initiation, and a transfer of C2,000.00.



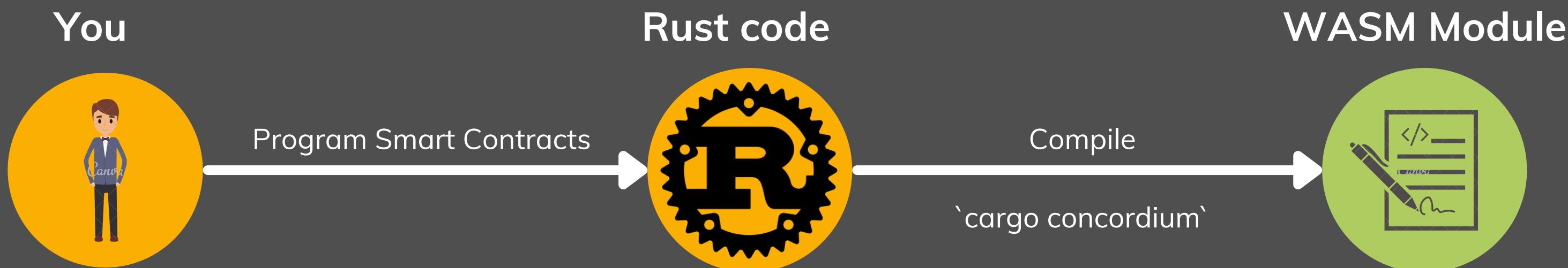
Rust Smart Contracts





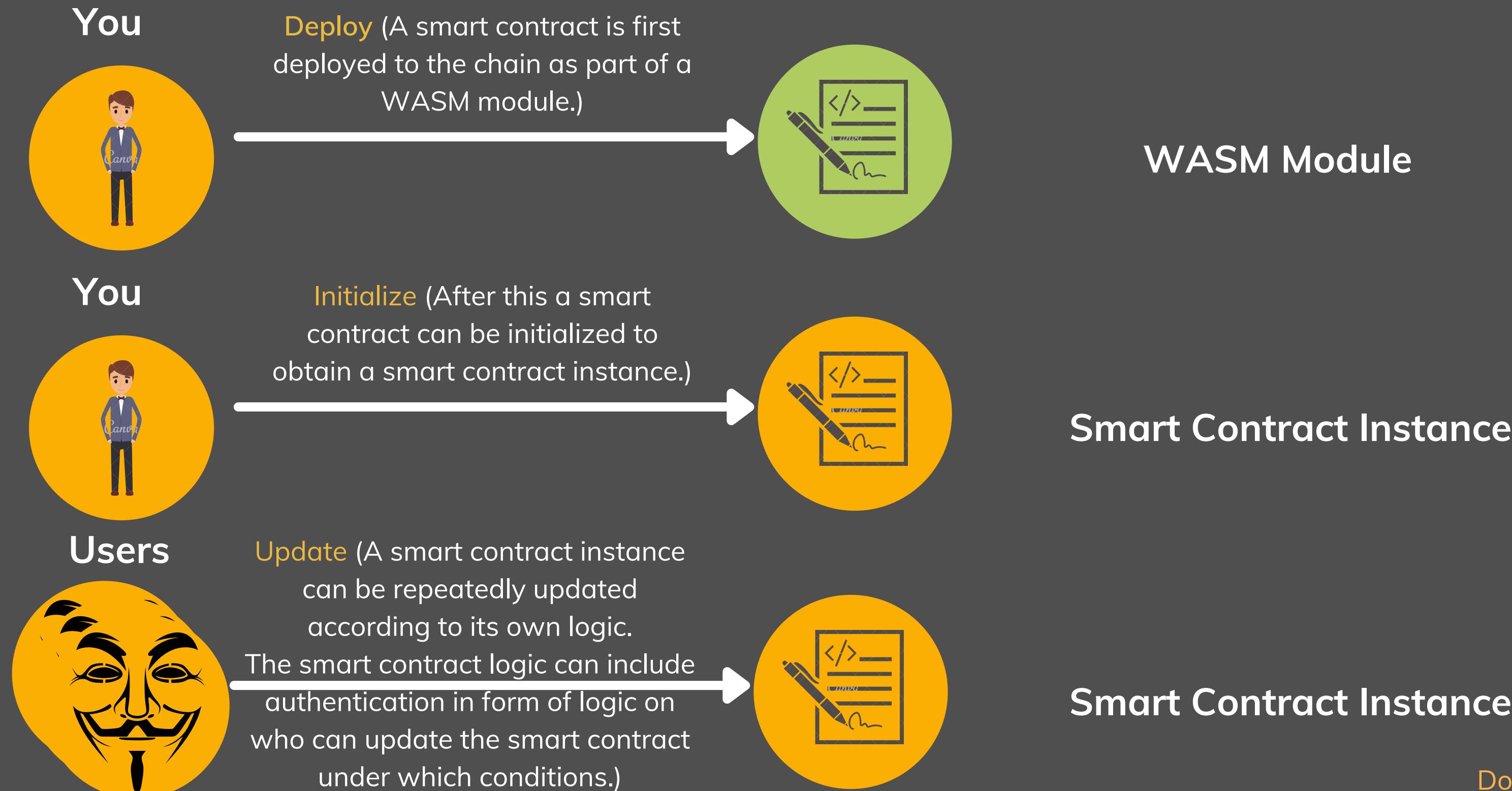
Introduction To Smart Contracts

Smart contracts are deployed as WASM modules on the Concordium chain. Rust currently has the best support to write and compile your smart contract into a WASM module that then can be deployed to the Concordium chain.





Life Cycle Of A Smart Contract On Concordium



Developer Tools

- Dapp Template
- Smart-Contract Tool
- ZK-Proof Explorer (ZK = zero knowledge)





Dapp Template

- Smart Contract Folder:
 - `cargo concordium build --out ./concordium-out/module.wasm.v1 --schema-embed` (to compile your smart contract)
 - Tips:
 - Use `--schema-embed` flag
 - Will simplify interacting with the smart contract once deployed on-chain.
- Front End Folder:
 - `yarn` (to install dependency)
 - `yarn dev` (to run the front-end in developer mode)
- `ccd-js-gen` Tool:
 - `yarn generate` (execute in the front-end folder)
 - Smart Contract Client Generator
 - Generates an Interface from the smart contract to be used at the front-end.



Let's run our first dApp:
<https://github.com/Concordium/concordium-dapp-starter>





Smart Contract Tool

- Works on mainnet and testnet
- Deploy a smart contract module
- Initialize a smart contract instance
- Read from any smart contract
- Write to any smart contract
- Tips:
 - Contract addresses on Concordium have an index and a subindex: <9350,0>.
 - Compile your smart contract with `--embedded schema` flag.
 - No embedded schema: e.g. <9329, 0>
 - Embedded schema: e.g. <9345,0>
 - 3 Types of error codes (<https://developer.concordium.software/en/mainnet/smart-contracts/tutorials/piggy-bank/deploying.html#concordium-std-crate-errors>):
 - -1, -2, -3, ... (smart contract logic errors)
 - -2147483636, ... (concordium-std errors: e.g. not-payable-error)
 - https://docs.rs/concordium-std/latest/concordium_std/#signalling-errors
 - -42000001, ... (cis2 token errors, e.g. insufficient funds), ...
 - <https://github.com/Concordium/concordium-rust-smart-contracts/blob/main/concordium-cis2/src/lib.rs#L1084>

Let's deploy our first smart contract:
<https://sctools.mainnet.concordium.software/>



The screenshot shows a deployed module with the ID 2e5c46. It displays basic information like age (Jun 6, 2024, 2:54 PM) and creator (3uNL2S). Below this, there are tabs for 'Linked contracts (1)', 'Linking events (0)', and 'Rejected events (0)'. A red arrow points to the 'Schema' tab, which is currently selected. Under 'Contract Address', it shows <9329, 0>. To the right, there is a timestamp of 3:00 PM and a date of Jun 6, 2024.

The screenshot shows a deployed module with the ID 6ba123. It displays basic information like age (Jun 10, 2024, 8:11 AM) and creator (3yhH9M). Below this, there are tabs for 'Linked contracts (1)', 'Linking events (0)', 'Rejected events (0)', and 'Schema'. A red arrow points to the 'Schema' tab, which is currently selected. Under 'Contract Address', it shows <9345, 0>. To the right, there is a timestamp of 8:15 AM and a date of Jun 10, 2024.



ZK Proof Explorer

- Front end creates a request.
- Request is sent to the wallet.
- User accepts or rejects the request. If accepted, the wallet generates a ZK proof.
- Backend verifies the proof.



1. Send Request

2. Return ZK Proof

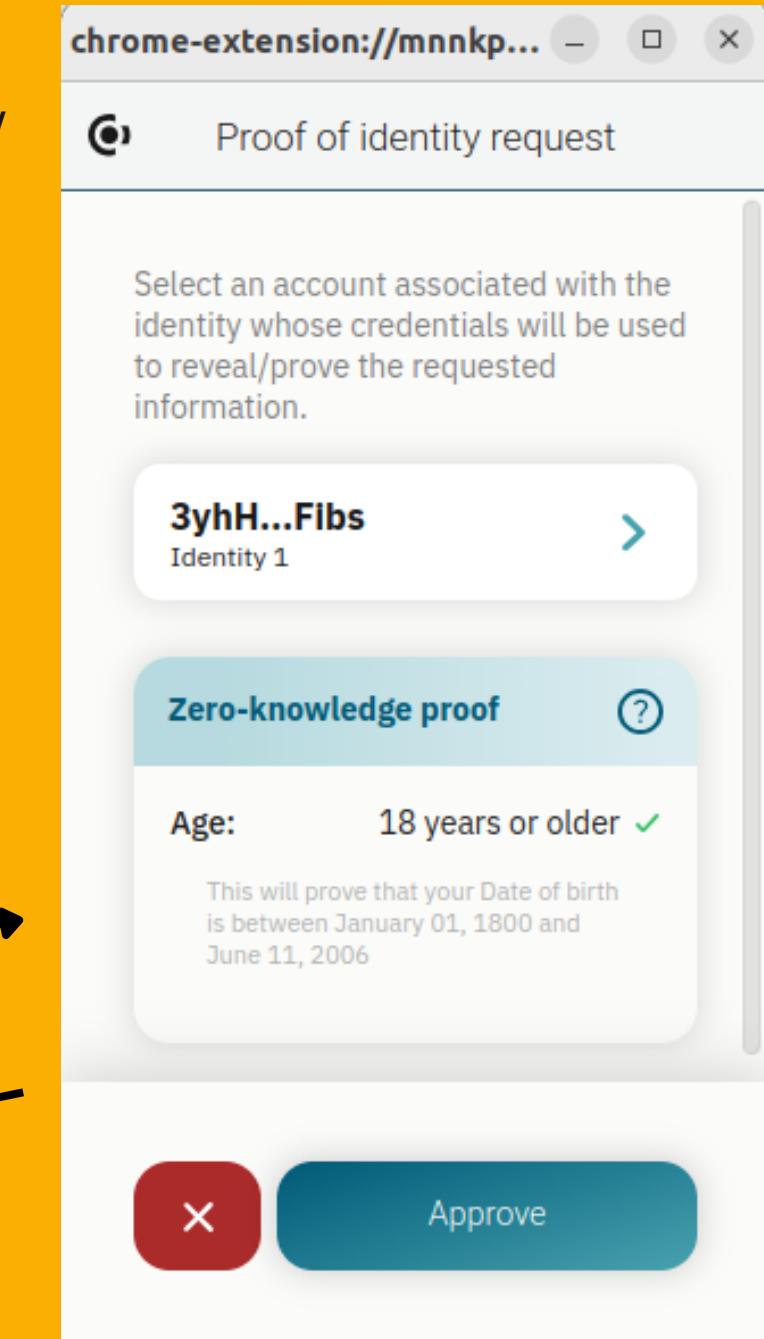
3. Send ZK Proof

Front End

5. Proof valid/notValid

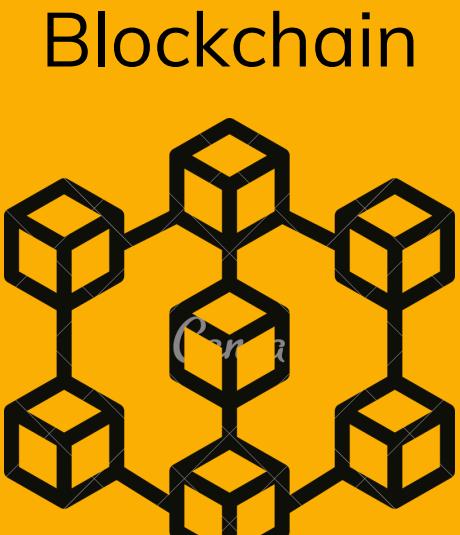


Let's explore ZK Proofs:
<https://web3id-proof-explorer.testnet.concordium.com/>

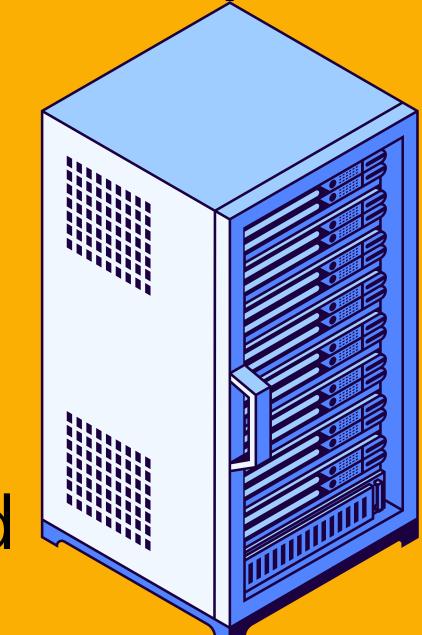


Wallet

Back End



4. LookUp
Cryptographic
Commitment





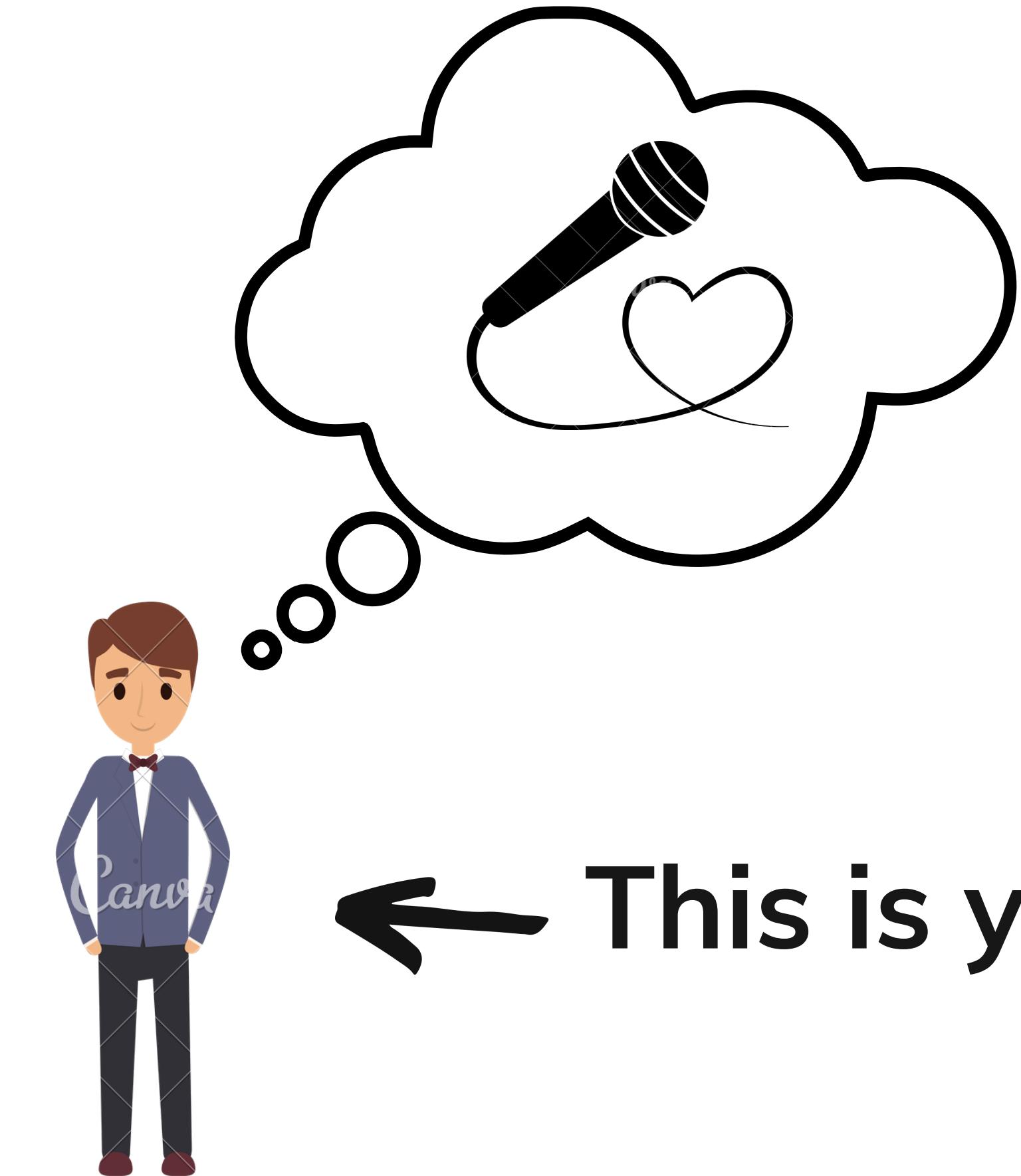
A Very Realistic Problem



Concordium.com



This is you..



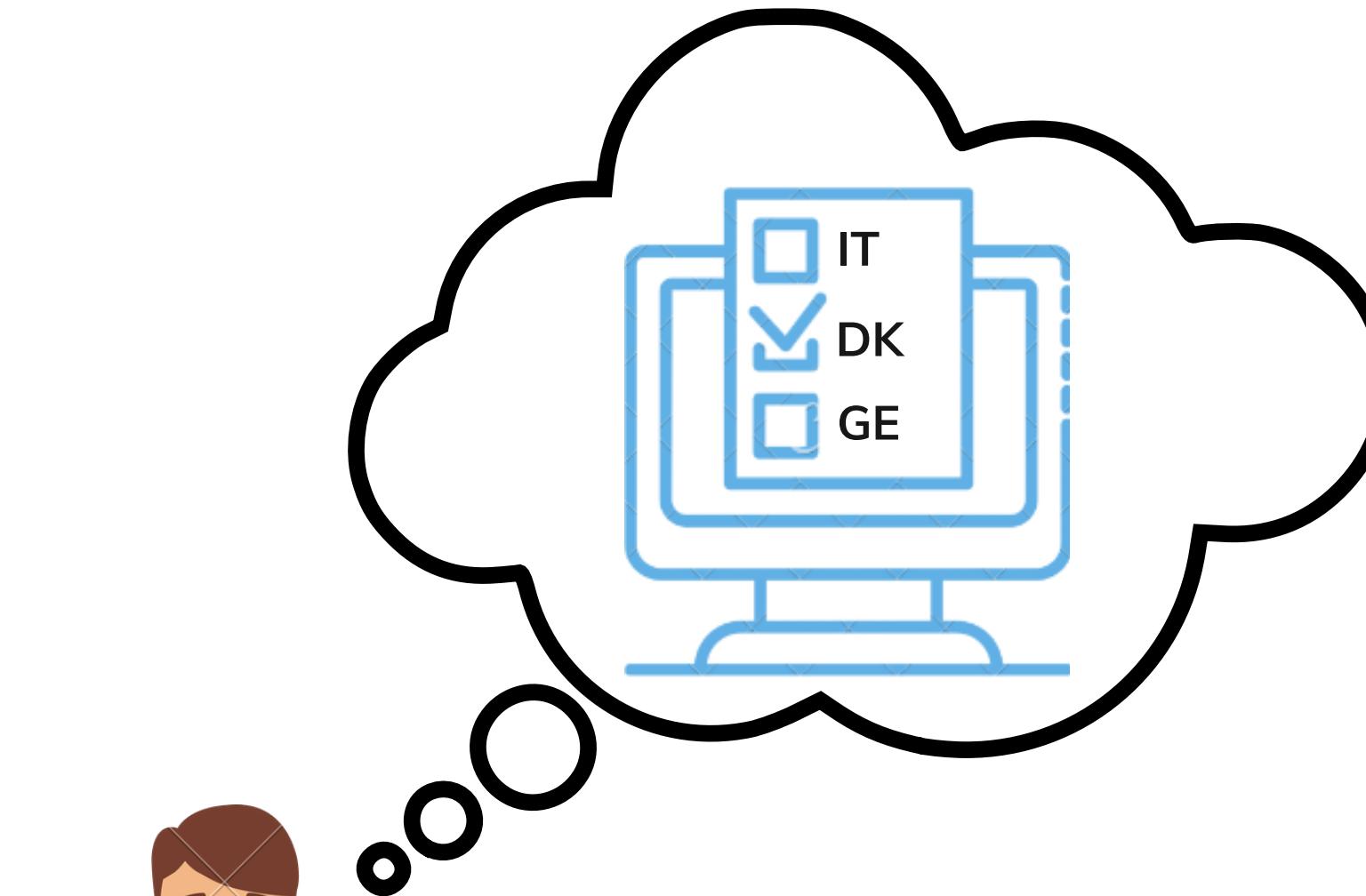
← This is you..



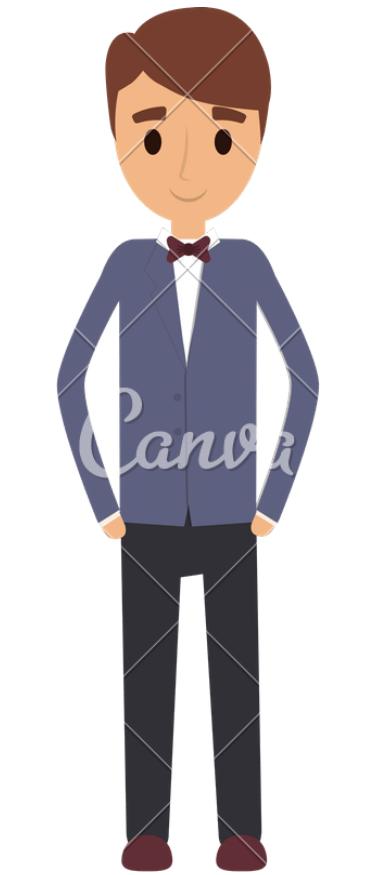
Concordium.com



← This is you..



This is you..

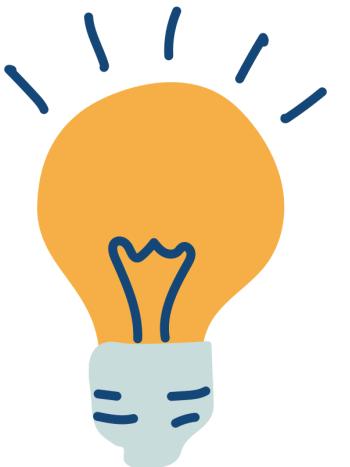


**Mister X should
not be able to
vote for his own
country.**

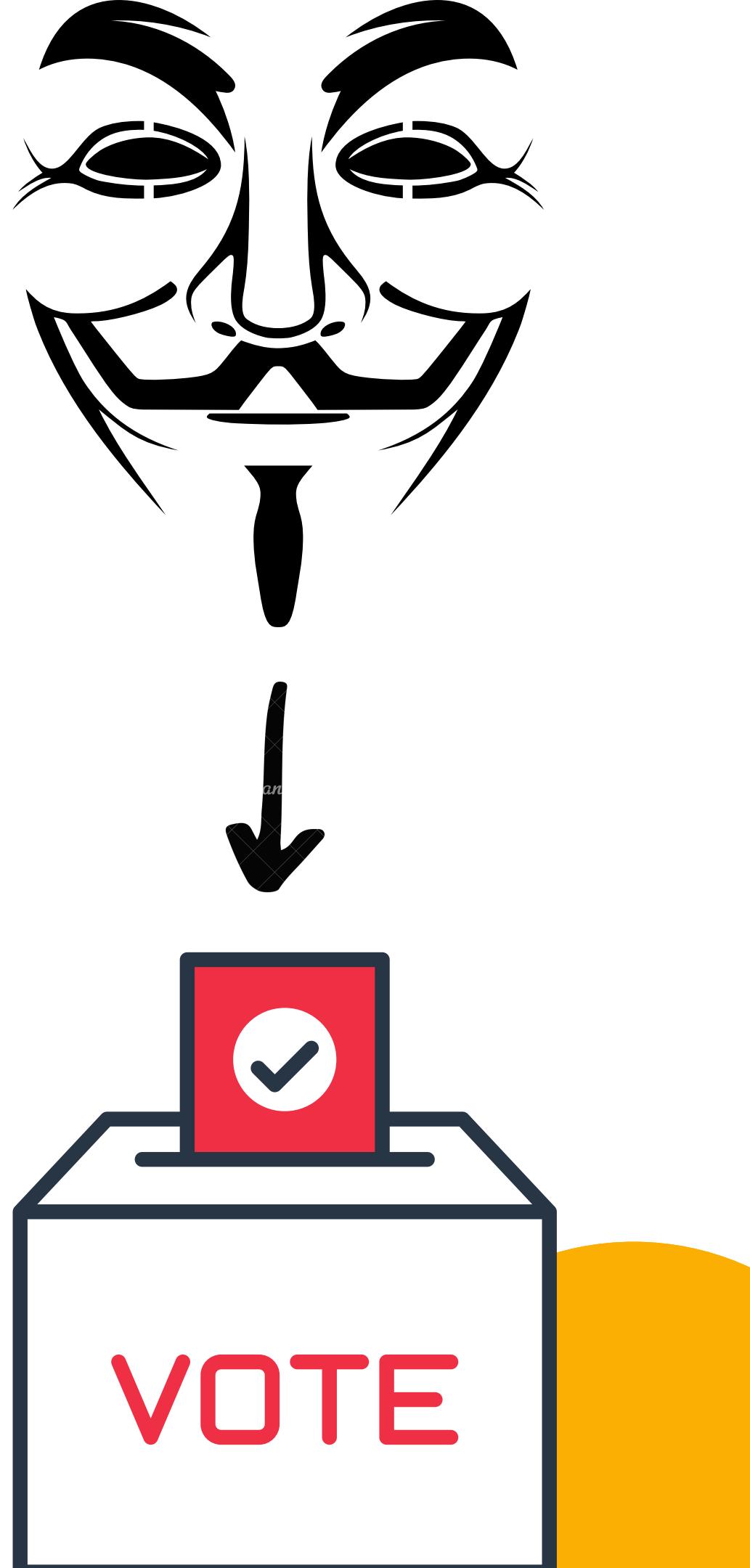




Concordium.com

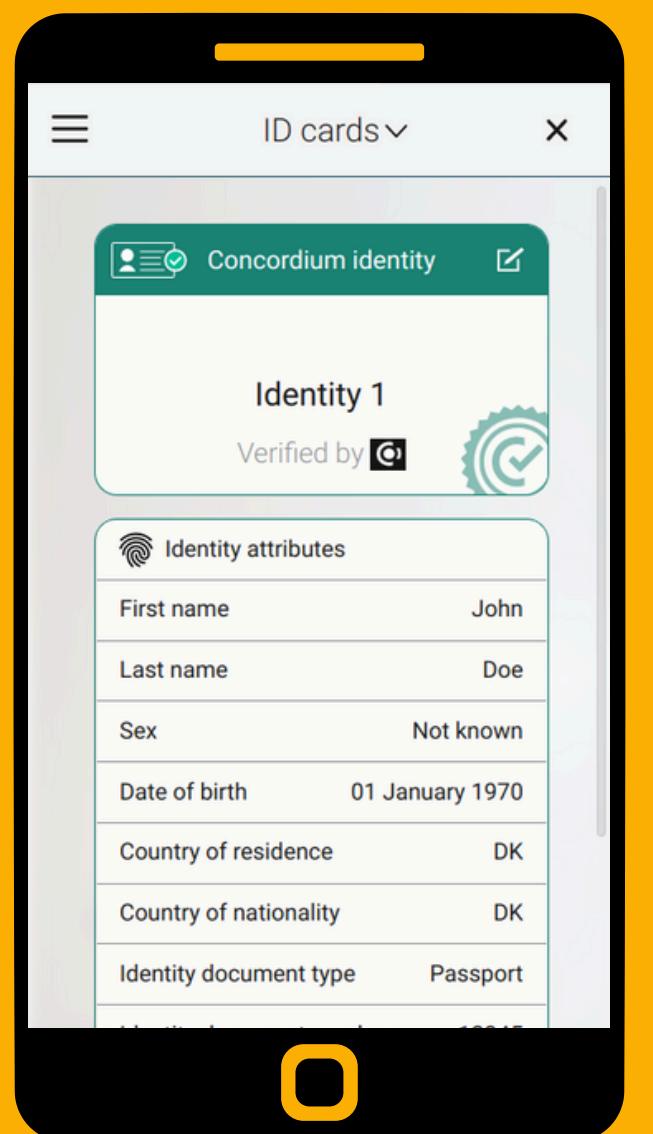


Concordium and its identity layer

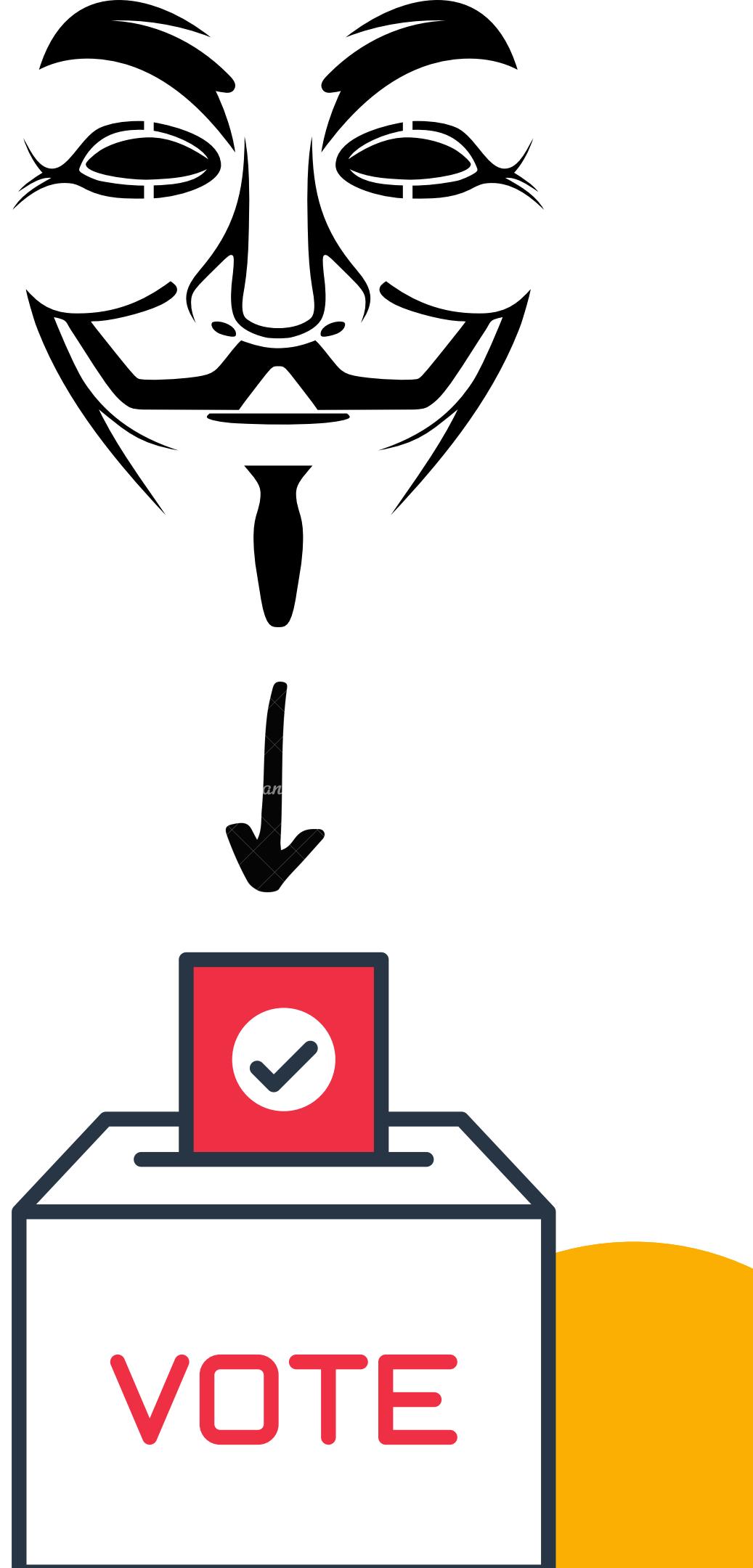




Concordium.com



Concordium and its identity layer





Concordium.com

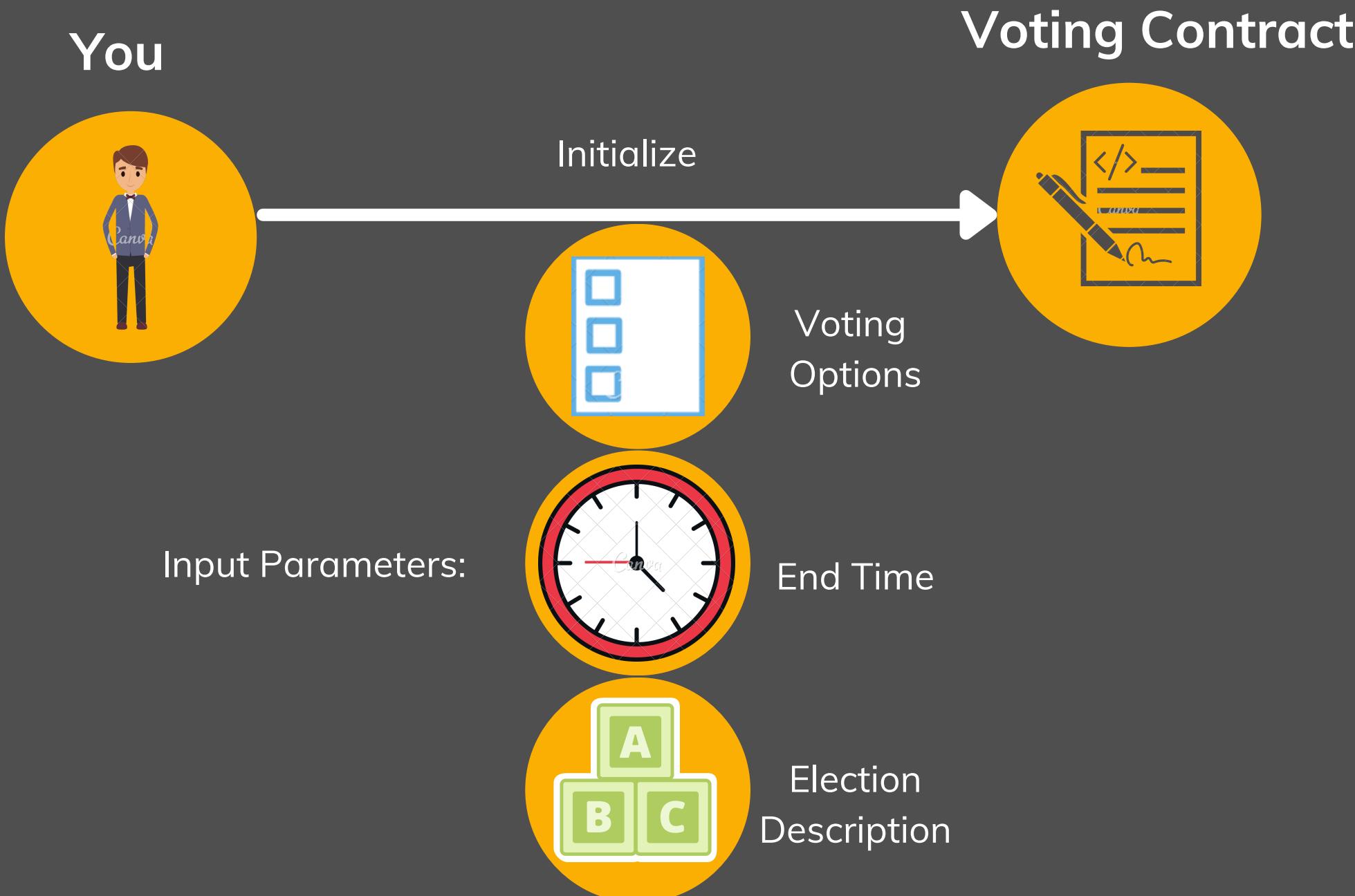
Identity Based Voting™

You





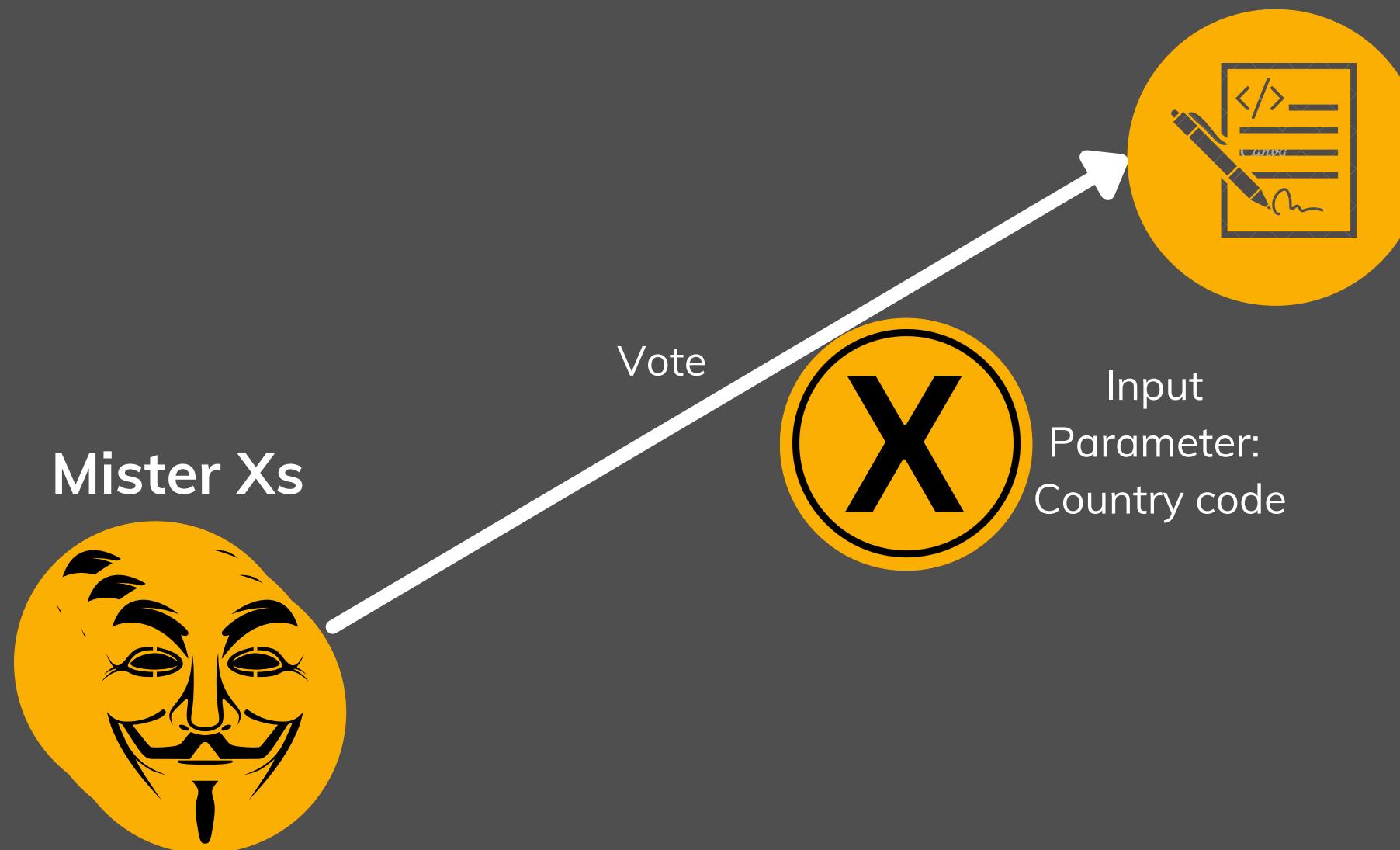
Identity Based Voting™





Identity Based Voting™

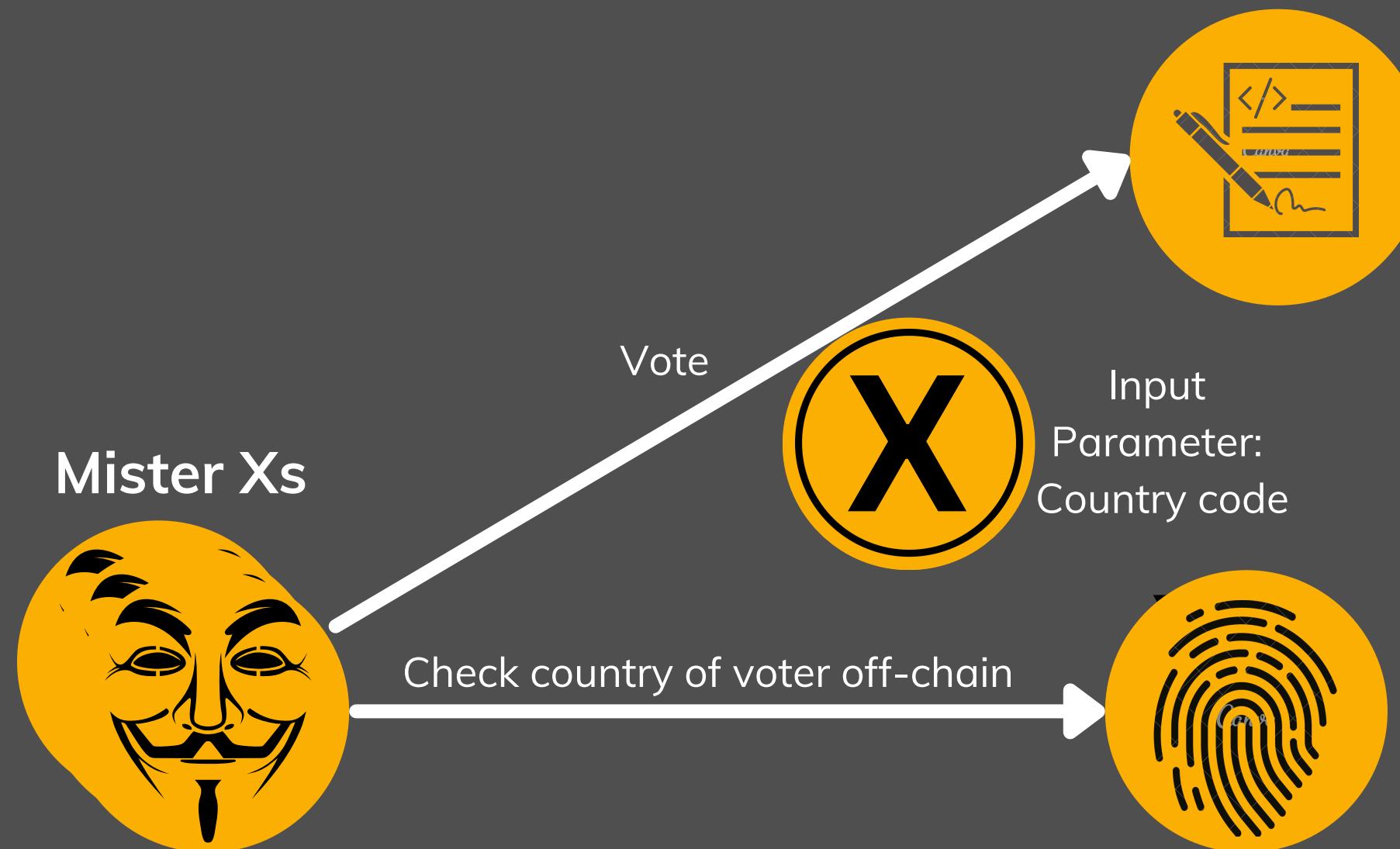
Voting Contract





Identity Based Voting™

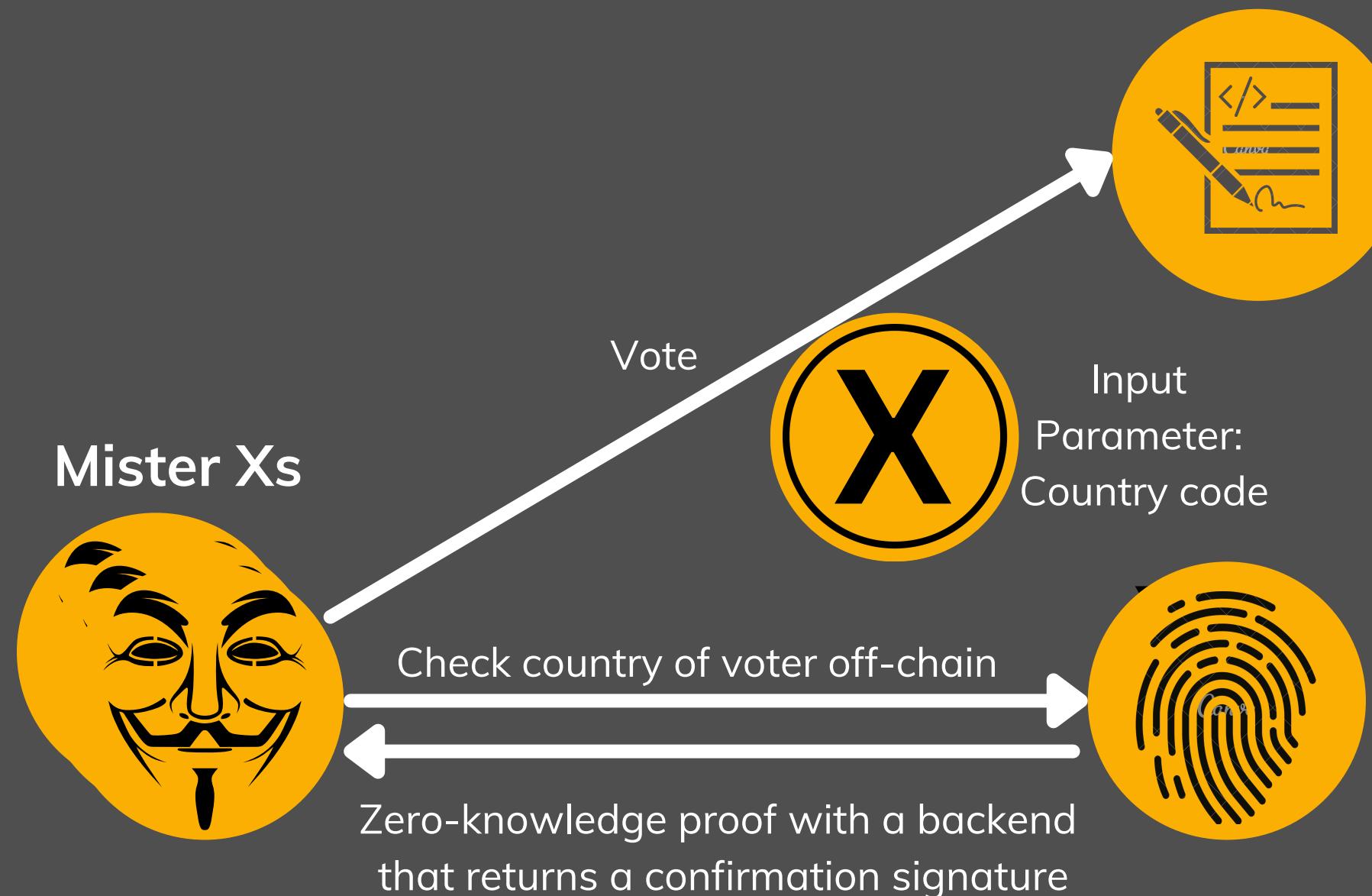
Voting Contract





Identity Based Voting™

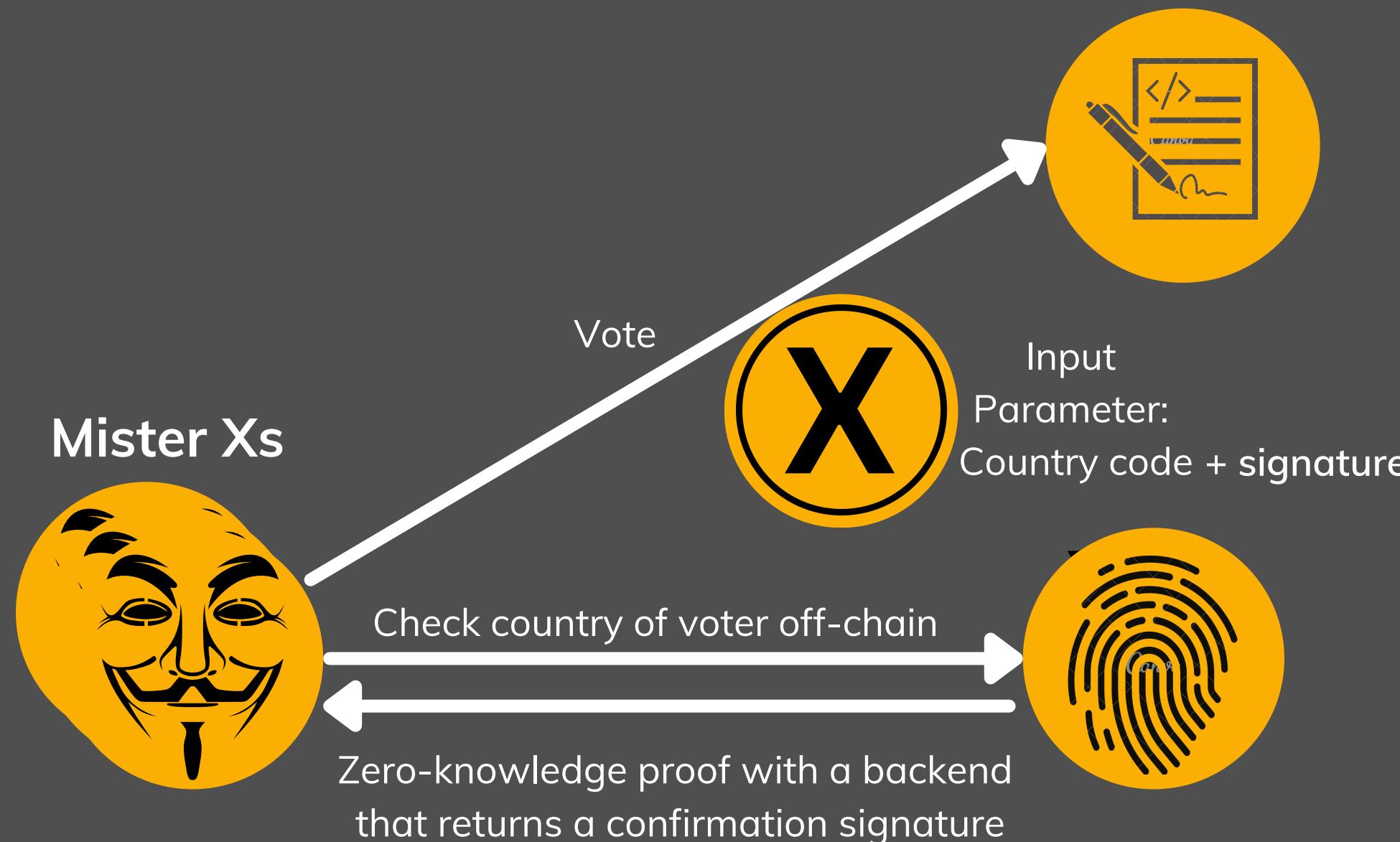
Voting Contract





Identity Based Voting™

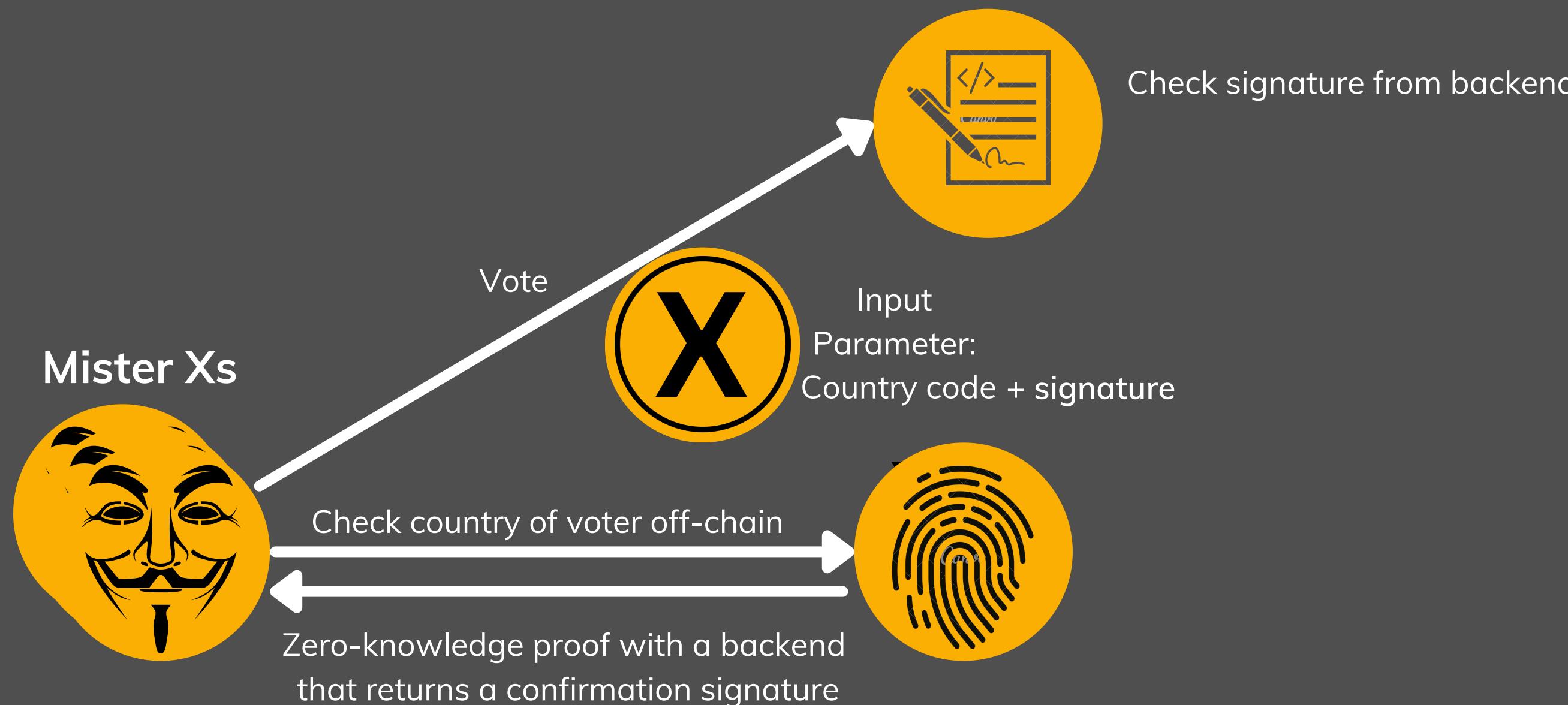
Voting Contract





Identity Based Voting™

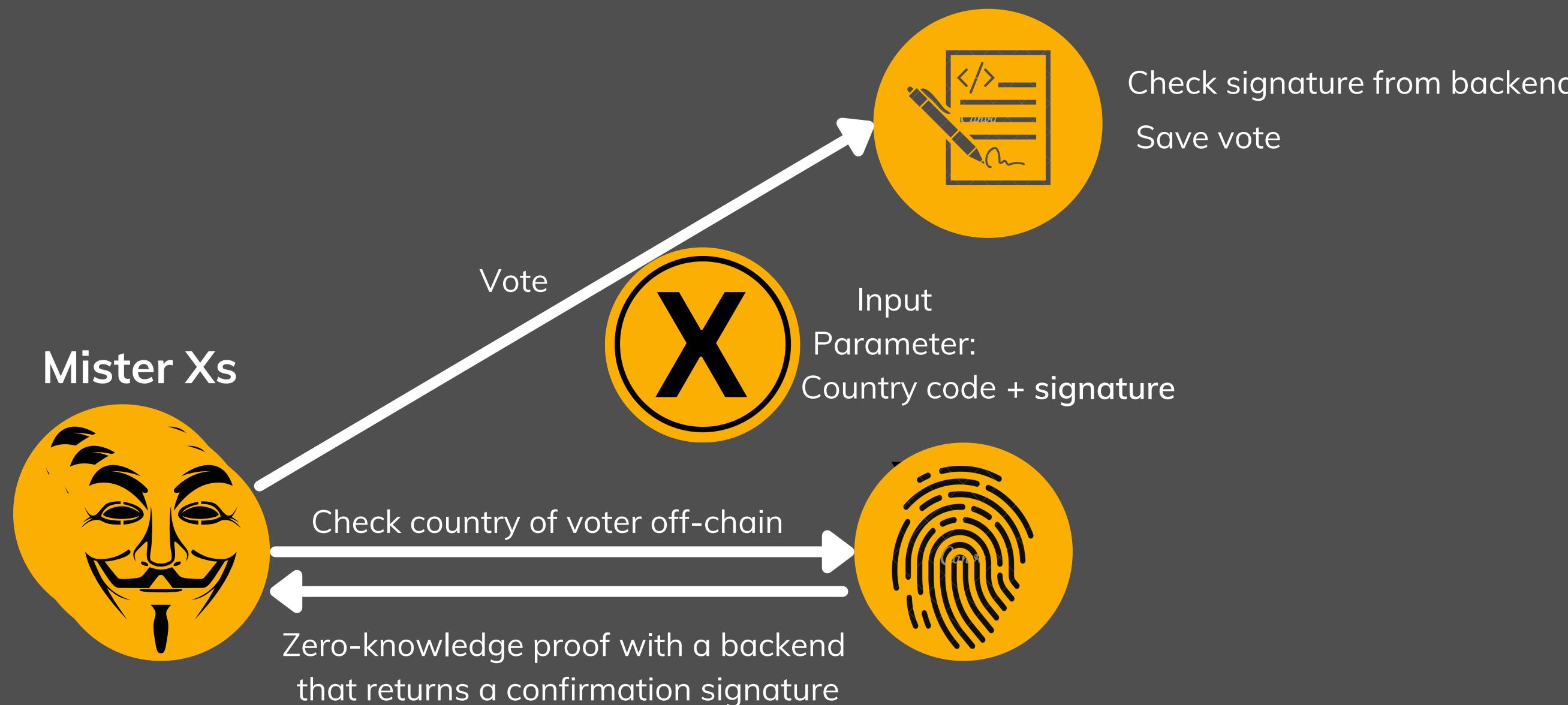
Voting Contract





Identity Based Voting™

Voting Contract





Workshop Material

<https://github.com/Concordium/workshops-and-hackathons>



Ideation session



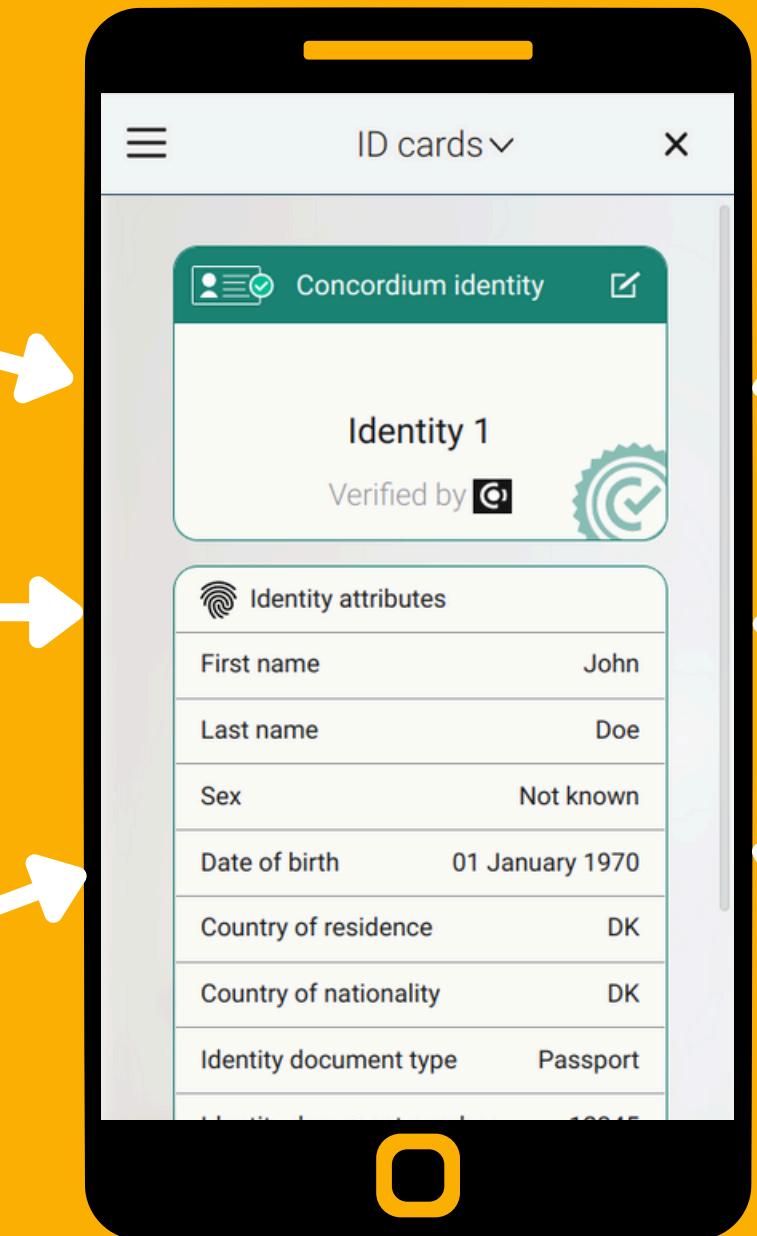
Frequent
Flyer Card



Employment Relation



Education
Certificates



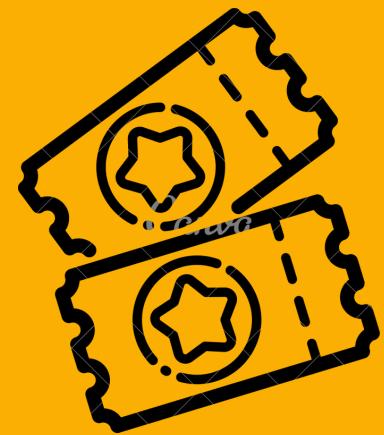
Are you older
than 18?



Who are you? Really?



Are you a
participant of
this event?





Additional Resources

- **Developer Documentation:**
 - <https://developer.concordium.software/en/mainnet/net/guides/learn-about-concordium.html>
- **Tutorials:**
 - <https://developer.concordium.software/en/mainnet/smart-contracts/tutorials/index.html>
- **Concordium Academy:**
 - <https://academy.concordium.software/>
- **Dapp Starter Template:**
 - <https://github.com/Concordium/concordium-dapp-starter>
- **Dapp Examples:**
 - <https://github.com/Concordium/concordium-dapp-examples>
 - <https://github.com/Concordium/concordium-browser-wallet/tree/main/examples>
- **Smart Contract Examples:**
 - <https://github.com/Concordium/concordium-rust-smart-contracts/tree/main/examples>
- **Deploy, Init, Read and Write Tool:**
 - <https://sctools.mainnet.concordium.software/>
- **Block Explorers:**
 - <https://ccdscan.io>
 - <https://ccdexplorer.io>



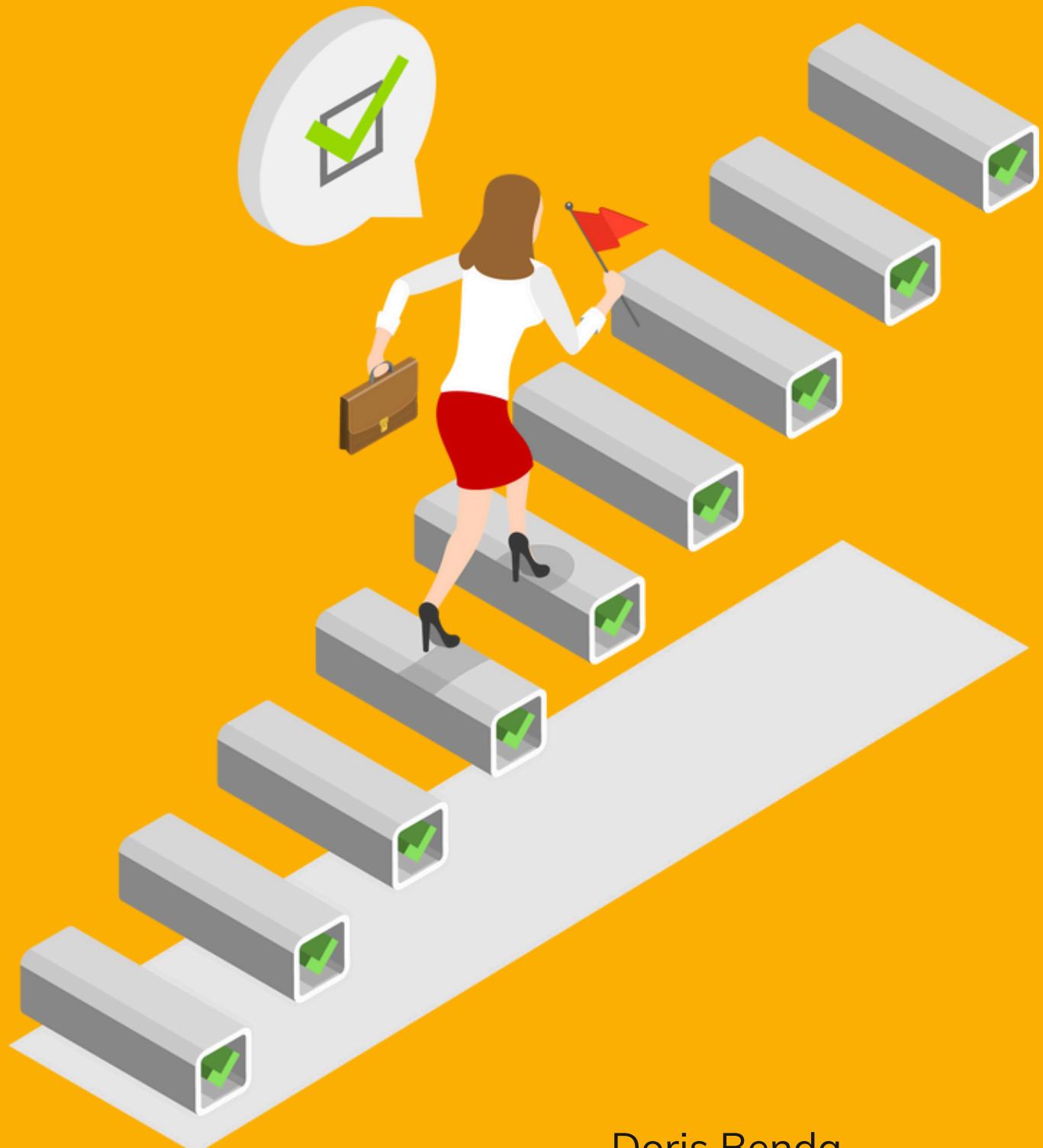
Additional Resources

- Public Testnet Nodes:
 - `node.testnet.concordium.com:20000`
 - e.g. `concordium-client consensus status --grpc-port 20000 --grpc-ip node.testnet.concordium.com`
 - `grpc.testnet.concordium.com:20000` (TLS enabled)
 - e.g. `concordium-client consensus status --grpc-port 20000 --grpc-ip grpc.testnet.concordium.com --secure`
- General software support:
 - <https://support.concordium.software/>
- Node-SDK:
 - <https://github.com/Concordium/concordium-node-sdk-js>
- Hosted front-ends:
 - <https://sponsored.testnet.concordium.com/>
 - <https://trackntrace.testnet.concordium.com/>
 - <https://wccd.testnet.concordium.com/>
 - <https://web3id-proof-explorer.testnet.concordium.com/>
 - <https://gallery.testnet.concordium.com/>
 - <https://piggybank.testnet.concordium.com/>
 - <http://signmessage.testnet.concordium.com/>



After The Hackathon - What's next?

- Apply for a Concordium Grant with your Project:
 - <https://www.concordium.com/grant-program>
- Jump on our Announced Ecosystem Grant Projects:
 - The PRs that gets reviewed and merged into the code base will receive the payout.
 - <https://github.com/bogacyigitbasi/Concordium-Bounty-Program/blob/main/README.md>
- Continue Tech Support:
 - <https://support.concordium.software/>





Questions?



**Good Luck And
Happy Hacking**

**Reach Out To Us
If You Have
Questions**