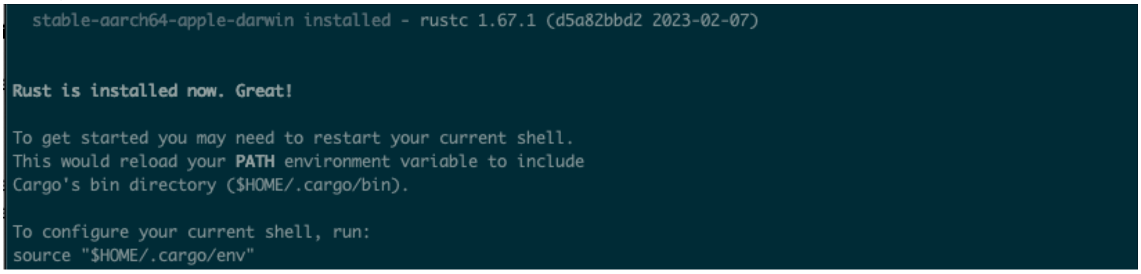
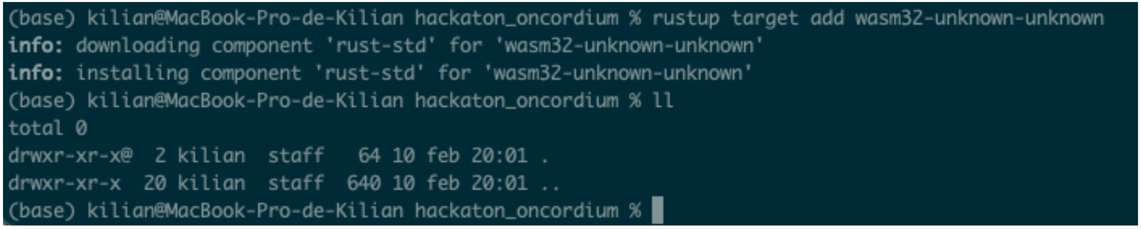
First, I have installed Rust.



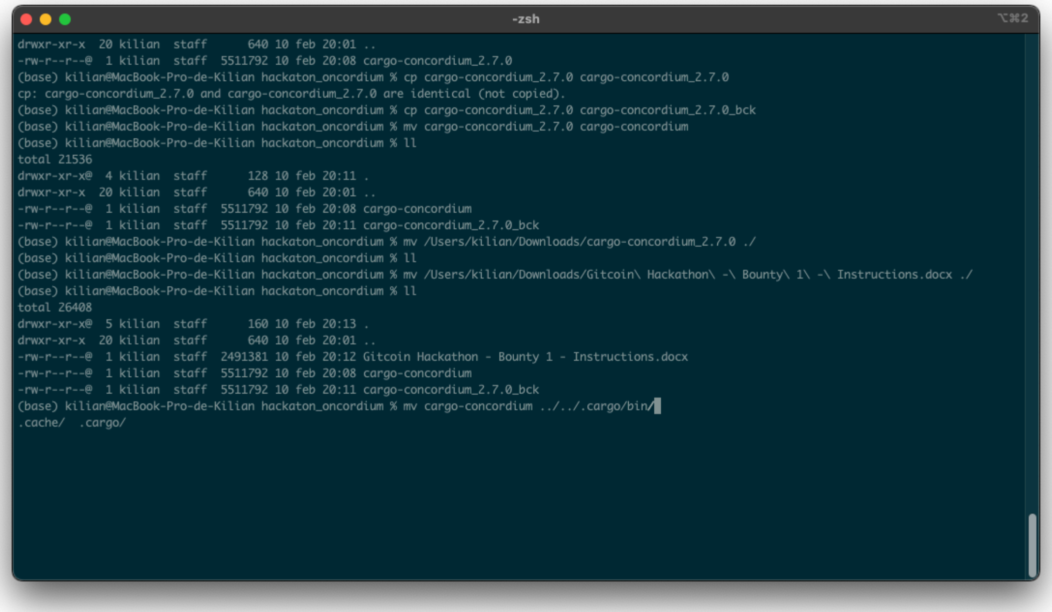
```
  stable-aarch64-apple-darwin installed - rustc 1.67.1 (d5a82bbd2 2023-02-07)


Rust is installed now. Great!

To get started you may need to restart your current shell.
This would reload your PATH environment variable to include
Cargo's bin directory ($HOME/.cargo/bin).

To configure your current shell, run:
source "$HOME/.cargo/env"
```

Secondly, I installed wasm32



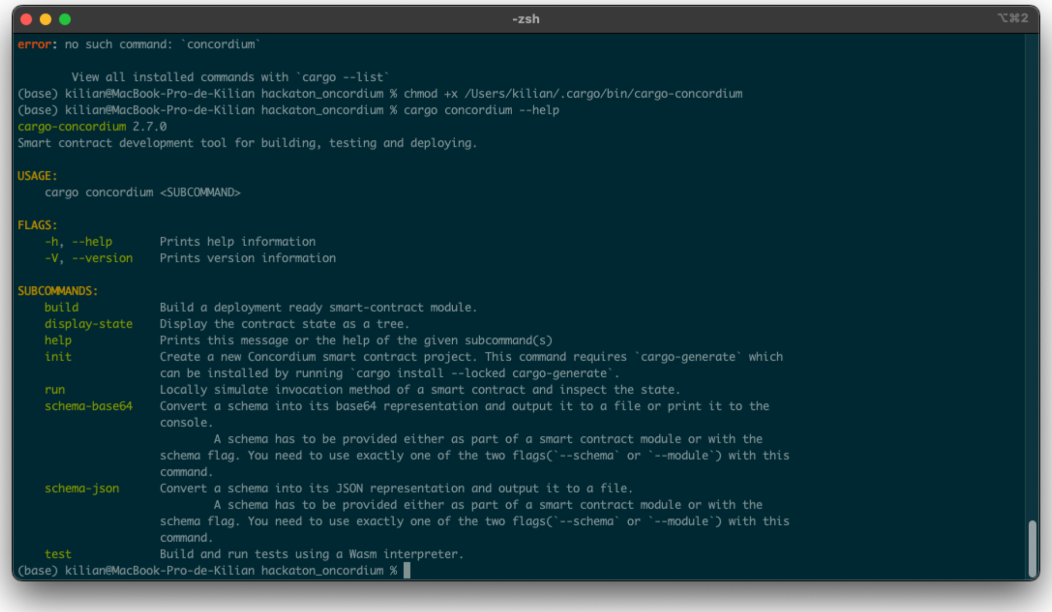
```
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % rustup target add wasm32-unknown-unknown
info: downloading component 'rust-std' for 'wasm32-unknown-unknown'
info: installing component 'rust-std' for 'wasm32-unknown-unknown'
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % ll
total 0
drwxr-xr-x@  2 kilian  staff   64 10 feb 20:01 .
drwxr-xr-x  20 kilian  staff  640 10 feb 20:01 ..
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium %
```

Download cargo-concordium



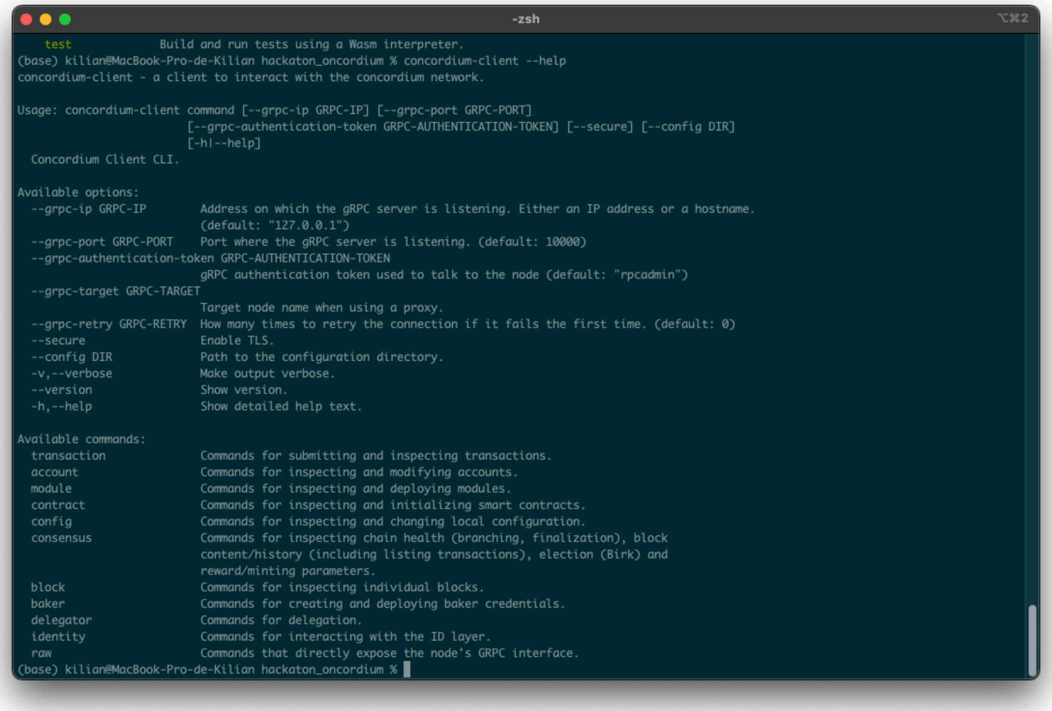
```
drwxr-xr-x  20 kilian  staff      640 10 feb 20:01 ..
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:08 cargo-concordium_2.7.0
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % cp cargo-concordium_2.7.0 cargo-concordium_2.7.0
cp: cargo-concordium_2.7.0 and cargo-concordium_2.7.0 are identical (not copied).
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % cp cargo-concordium_2.7.0 cargo-concordium_2.7.0_bck
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % mv cargo-concordium_2.7.0 cargo-concordium
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % ll
total 21536
drwxr-xr-x@  4 kilian  staff      128 10 feb 20:11 .
drwxr-xr-x  20 kilian  staff      640 10 feb 20:01 ..
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:08 cargo-concordium
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:11 cargo-concordium_2.7.0_bck
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % mv /Users/kilian/Downloads/cargo-concordium_2.7.0 ./
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % ll
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % mv /Users/kilian/Downloads/Gitcoin\ Hackathon\ -\ Bounty\ 1\ -\ Instructions.docx ./
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % ll
total 26408
drwxr-xr-x@  5 kilian  staff      160 10 feb 20:13 .
drwxr-xr-x  20 kilian  staff      640 10 feb 20:01 ..
-rw-r--r--@  1 kilian  staff  2491381 10 feb 20:12 Gitcoin Hackathon - Bounty 1 - Instructions.docx
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:08 cargo-concordium
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:11 cargo-concordium_2.7.0_bck
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % mv cargo-concordium ../../.cargo/bin/
.cache/  .cargo/
```

It works!



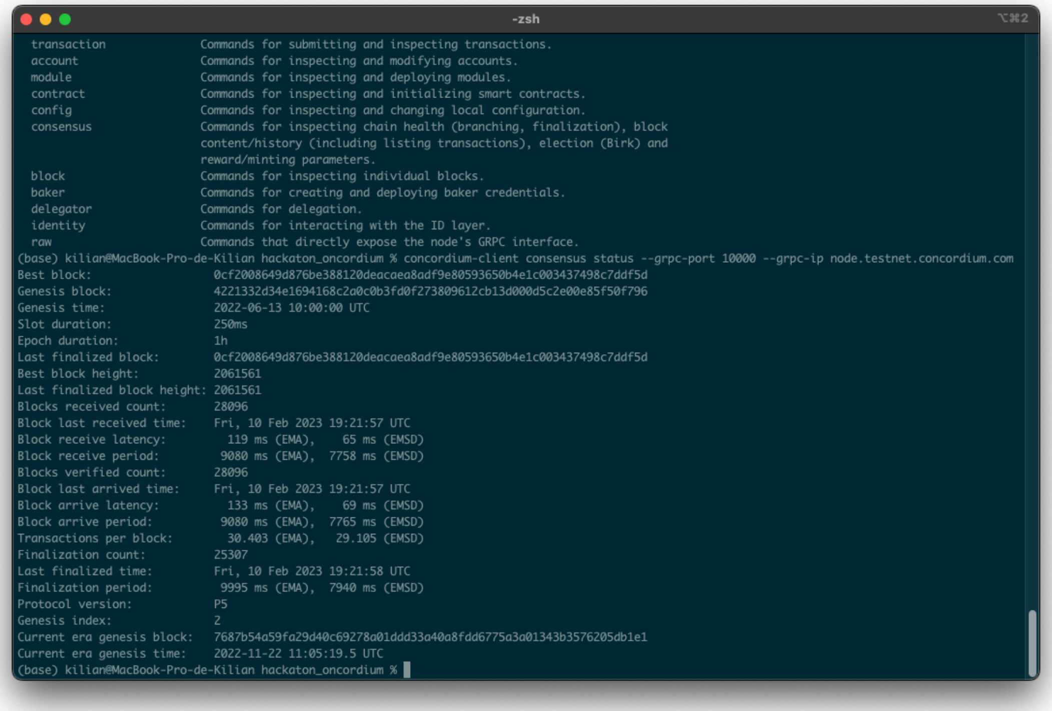
```
error: no such command: `concordium`

        View all installed commands with `cargo --list`
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % chmod +x /Users/kilian/.cargo/bin/cargo-concordium
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % cargo concordium --help
cargo-concordium 2.7.0
Smart contract development tool for building, testing and deploying.

USAGE:
    cargo concordium <SUBCOMMAND>

FLAGS:
    -h, --help       Prints help information
    -V, --version    Prints version information

SUBCOMMANDS:
    build            Build a deployment ready smart-contract module.
    display-state    Display the contract state as a tree.
    help             Prints this message or the help of the given subcommand(s)
    init             Create a new Concordium smart contract project. This command requires `cargo-generate` which
                     can be installed by running `cargo install --locked cargo-generate`.
    run              Locally simulate invocation method of a smart contract and inspect the state.
    schema-base64    Convert a schema into its base64 representation and output it to a file or print it to the
                     console.
                             A schema has to be provided either as part of a smart contract module or with the
                     schema flag. You need to use exactly one of the two flags(`--schema` or `--module`) with this
                     command.
    schema-json      Convert a schema into its JSON representation and output it to a file.
                             A schema has to be provided either as part of a smart contract module or with the
                     schema flag. You need to use exactly one of the two flags(`--schema` or `--module`) with this
                     command.
    test             Build and run tests using a Wasm interpreter.
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % 
```

Also concordium-client.



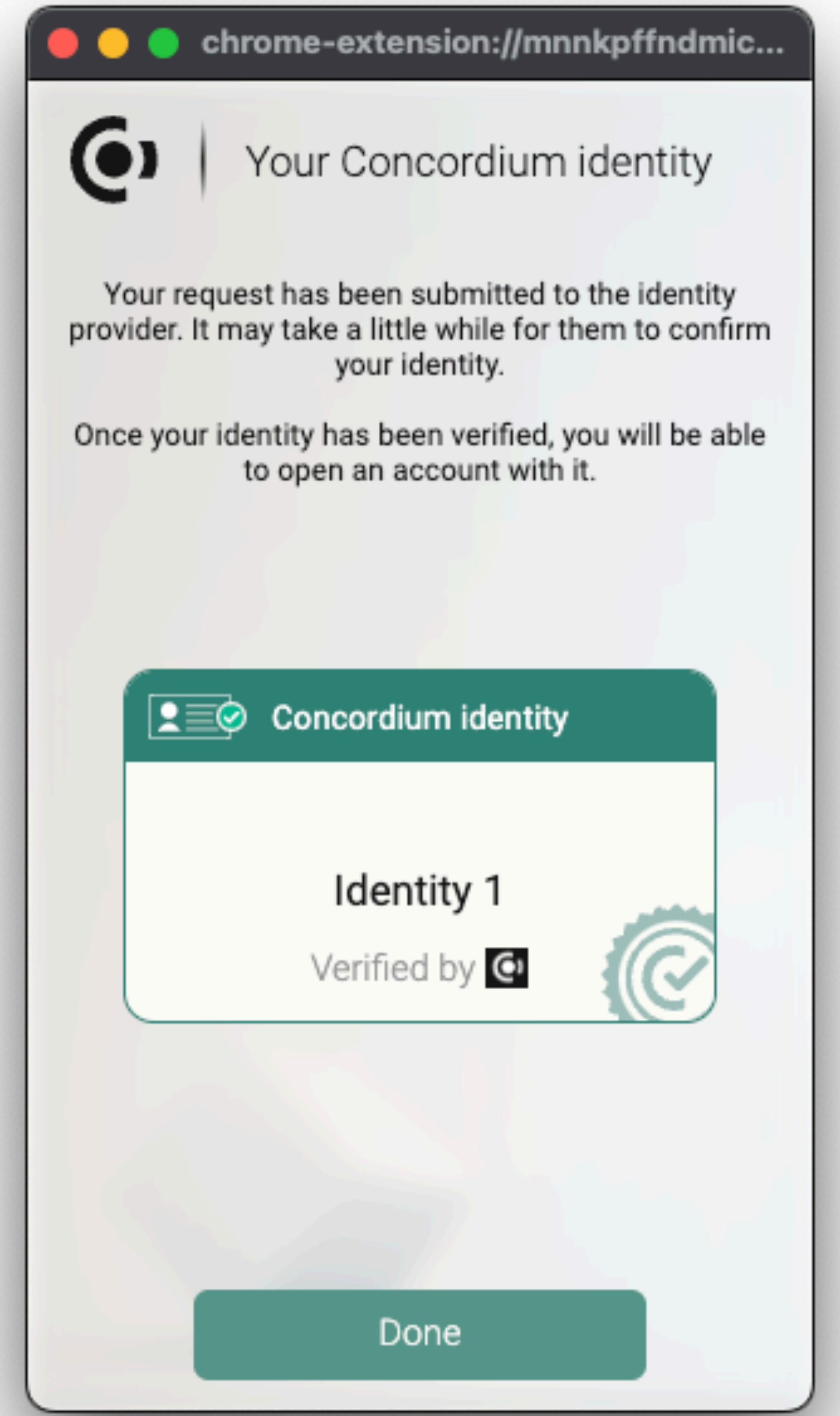
```
    test             Build and run tests using a Wasm interpreter.
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % concordium-client --help
concordium-client - a client to interact with the concordium network.

Usage: concordium-client command [--grpc-ip GRPC-IP] [--grpc-port GRPC-PORT]
                         [--grpc-authentication-token GRPC-AUTHENTICATION-TOKEN] [--secure] [--config DIR]
                         [-h|--help]
  Concordium Client CLI.

Available options:
  --grpc-ip GRPC-IP        Address on which the gRPC server is listening. Either an IP address or a hostname.
                           (default: "127.0.0.1")
  --grpc-port GRPC-PORT    Port where the gRPC server is listening. (default: 10000)
  --grpc-authentication-token GRPC-AUTHENTICATION-TOKEN
                           gRPC authentication token used to talk to the node (default: "rpcadmin")
  --grpc-target GRPC-TARGET
                           Target node name when using a proxy.
  --grpc-retry GRPC-RETRY  How many times to retry the connection if it fails the first time. (default: 0)
  --secure                 Enable TLS.
  --config DIR             Path to the configuration directory.
  -v,--verbose             Make output verbose.
  --version                Show version.
  -h,--help                Show detailed help text.

Available commands:
  transaction              Commands for submitting and inspecting transactions.
  account                  Commands for inspecting and modifying accounts.
  module                   Commands for inspecting and deploying modules.
  contract                 Commands for inspecting and initializing smart contracts.
  config                   Commands for inspecting and changing local configuration.
  consensus                Commands for inspecting chain health (branching, finalization), block
                           content/history (including listing transactions), election (Birk) and
                           reward/minting parameters.
  block                    Commands for inspecting individual blocks.
  baker                    Commands for creating and deploying baker credentials.
  delegator                Commands for delegation.
  identity                 Commands for interacting with the ID layer.
  raw                      Commands that directly expose the node's GRPC interface.
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % 
```

Connected to test network



```
  transaction              Commands for submitting and inspecting transactions.
  account                  Commands for inspecting and modifying accounts.
  module                   Commands for inspecting and deploying modules.
  contract                 Commands for inspecting and initializing smart contracts.
  config                   Commands for inspecting and changing local configuration.
  consensus                Commands for inspecting chain health (branching, finalization), block
                           content/history (including listing transactions), election (Birk) and
                           reward/minting parameters.
  block                    Commands for inspecting individual blocks.
  baker                    Commands for creating and deploying baker credentials.
  delegator                Commands for delegation.
  identity                 Commands for interacting with the ID layer.
  raw                      Commands that directly expose the node's GRPC interface.
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % concordium-client consensus status --grpc-port 10000 --grpc-ip node.testnet.concordium.com
Best block:                 0cf2008649d876be388120deacaea8adf9e80593650b4e1c003437498c7ddf5d
Genesis block:              4221332d34e1694168c2a0c0b3fd0f273809612cb13d000d5c2e00e85f50f796
Genesis time:               2022-06-13 10:00:00 UTC
Slot duration:              250ms
Epoch duration:             1h
Last finalized block:       0cf2008649d876be388120deacaea8adf9e80593650b4e1c003437498c7ddf5d
Best block height:          2061561
Last finalized block height: 2061561
Blocks received count:      28096
Block last received time:   Fri, 10 Feb 2023 19:21:57 UTC
Block receive latency:        119 ms (EMA),    65 ms (EMSD)
Block receive period:        9080 ms (EMA),  7758 ms (EMSD)
Blocks verified count:      28096
Block last arrived time:    Fri, 10 Feb 2023 19:21:57 UTC
Block arrive latency:         133 ms (EMA),    69 ms (EMSD)
Block arrive period:         9080 ms (EMA),  7765 ms (EMSD)
Transactions per block:       30.403 (EMA),   29.105 (EMSD)
Finalization count:         25307
Last finalized time:        Fri, 10 Feb 2023 19:21:58 UTC
Finalization period:         9995 ms (EMA),  7940 ms (EMSD)
Protocol version:           P5
Genesis index:              2
Current era genesis block:  7687b54a59fa29d40c69278a01ddd33a40a8fdd6775a3a01343b3576205db1e1
Current era genesis time:   2022-11-22 11:05:19.5 UTC
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium %
```

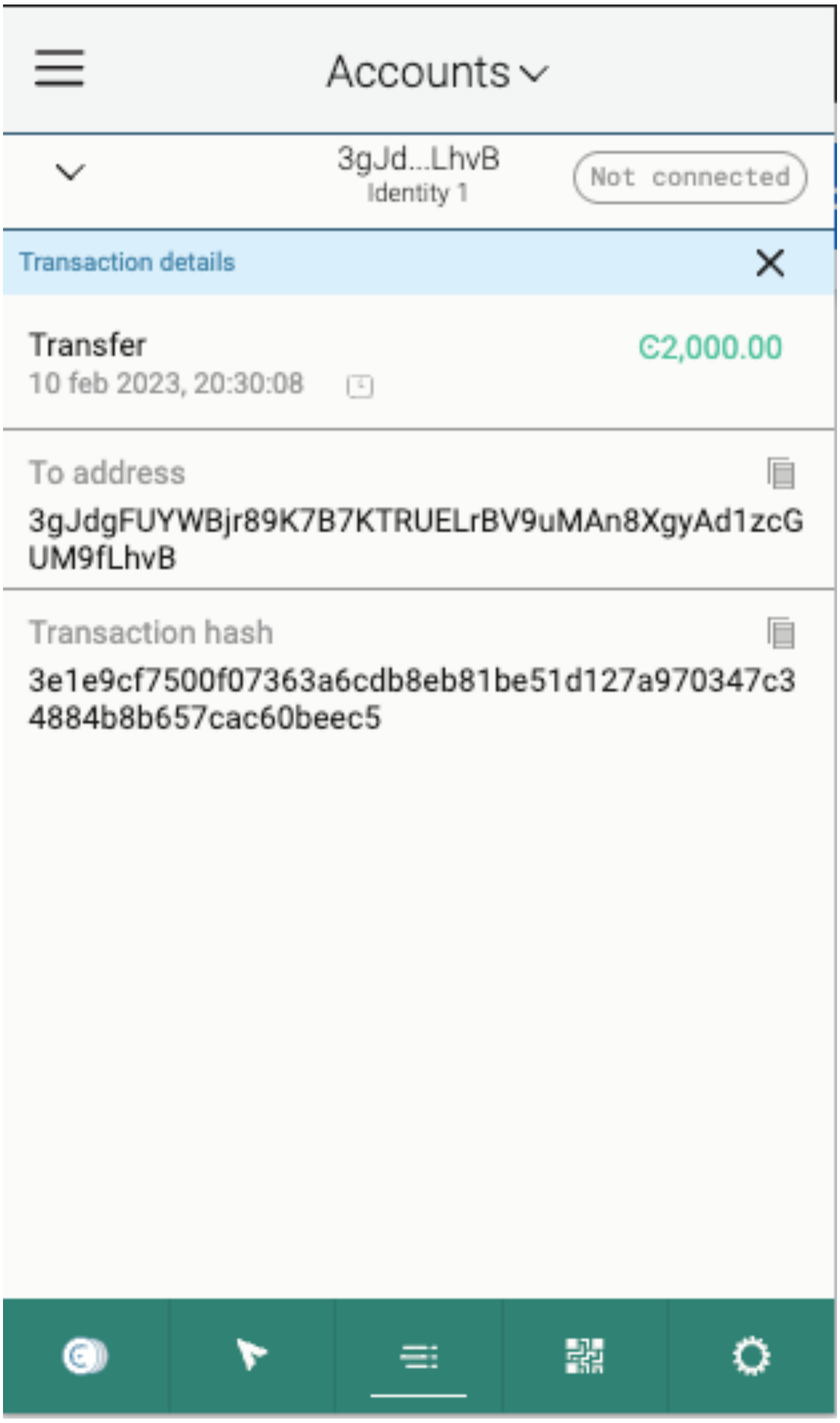
And finally, the screenshots of the wallet set up for testnet and mainnet



Your Concordium identity

Your request has been submitted to the identity provider. It may take a little while for them to confirm your identity.

Once your identity has been verified, you will be able to open an account with it.



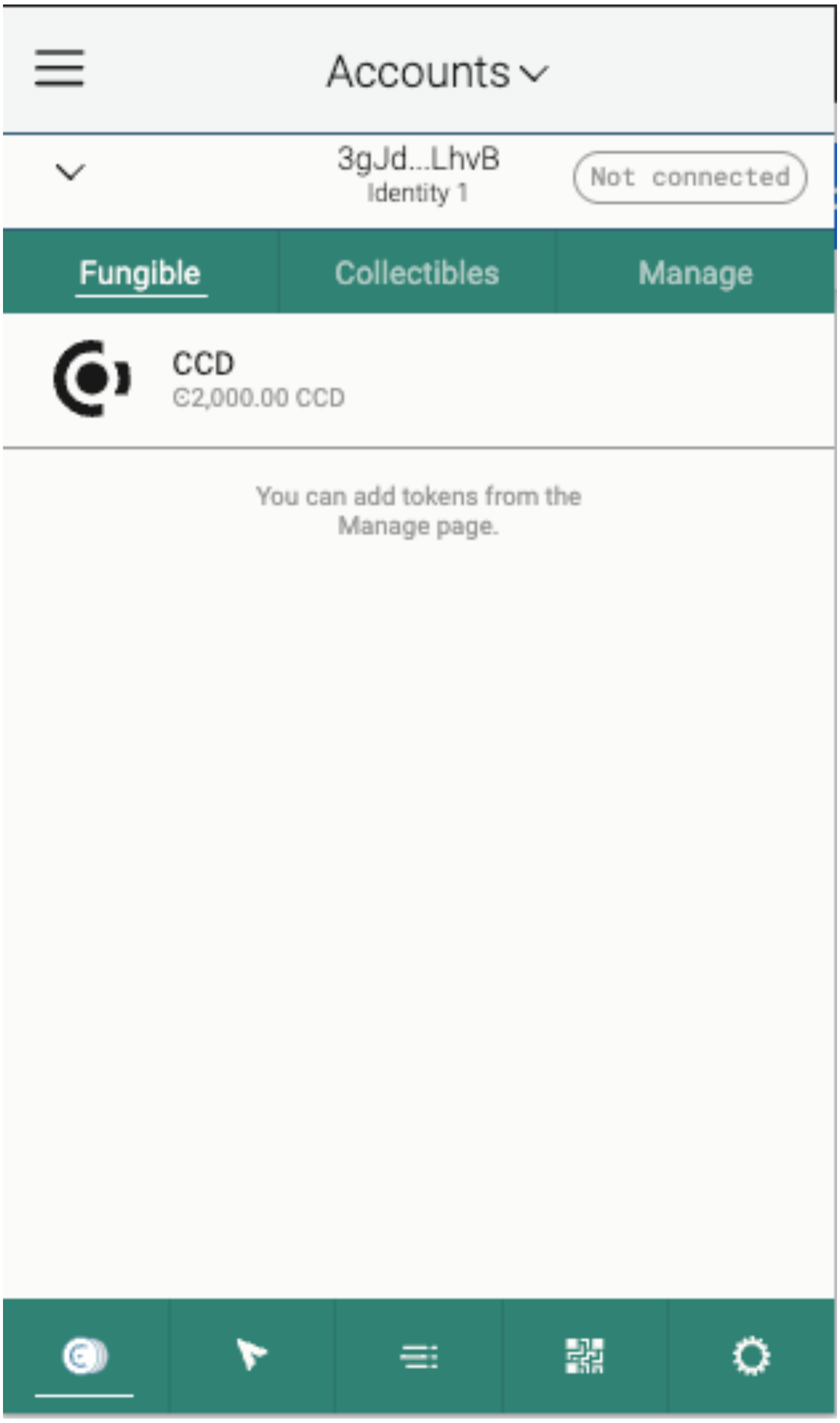
Transaction details

Transfer
10 feb 2023, 20:30:08

Ͼ2,000.00

To address
3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB

Transaction hash
3e1e9cf7500f07363a6cdb8eb81be51d127a970347c34884b8b657cac60beec5



Accounts

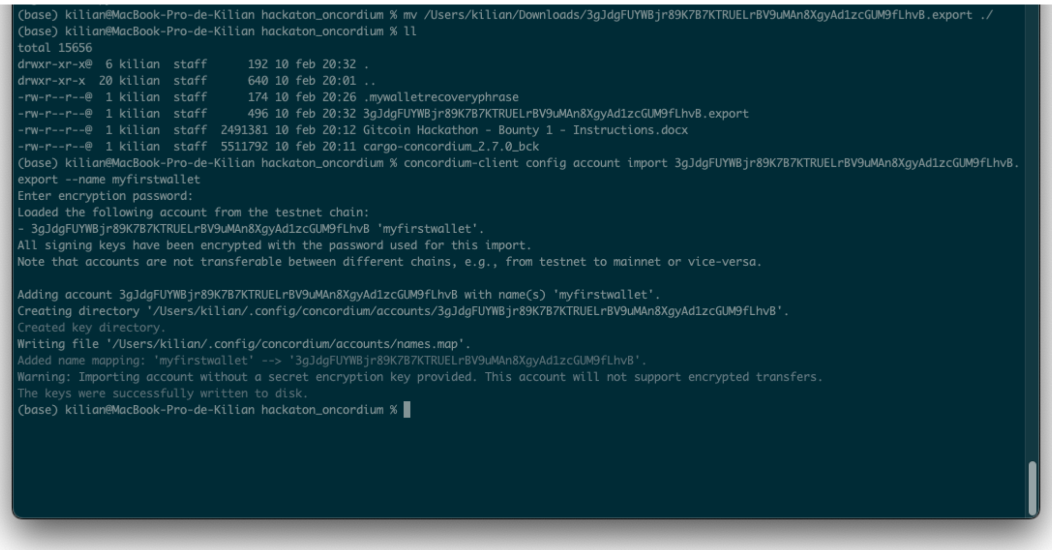
3gJd...LhvB
Identity 1

Not connected

Fungible
Collectibles
Manage

CCD
Ͼ2,000.00 CCD

You can add tokens from the Manage page.



```
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % mv /Users/kilian/Downloads/3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB.export ./
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % ll
total 15656
drwxr-xr-x@  6 kilian  staff      192 10 feb 20:32 .
drwxr-xr-x  20 kilian  staff      640 10 feb 20:01 ..
-rw-r--r--@  1 kilian  staff      174 10 feb 20:26 .mywalletrecoveryphrase
-rw-r--r--@  1 kilian  staff      496 10 feb 20:32 3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB.export
-rw-r--r--@  1 kilian  staff  2491381 10 feb 20:12 Gitcoin Hackathon - Bounty 1 - Instructions.docx
-rw-r--r--@  1 kilian  staff  5511792 10 feb 20:11 cargo-concordium_2.7.0_bck
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium % concordium-client config account import 3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB.
export --name myfirstwallet
Enter encryption password:
Loaded the following account from the testnet chain:
- 3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB 'myfirstwallet'.
All signing keys have been encrypted with the password used for this import.
Note that accounts are not transferable between different chains, e.g., from testnet to mainnet or vice-versa.

Adding account 3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB with name(s) 'myfirstwallet'.
Creating directory '/Users/kilian/.config/concordium/accounts/3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB'.
Created key directory.
Writing file '/Users/kilian/.config/concordium/accounts/names.map'.
Added name mapping: 'myfirstwallet' --> '3gJdgFUYWBjr89K7B7KTRUELrBV9uMAn8XgyAd1zcGUM9fLhvB'.
Warning: Importing account without a secret encryption key provided. This account will not support encrypted transfers.
The keys were successfully written to disk.
(base) kilian@MacBook-Pro-de-Kilian hackaton_oncordium %
```