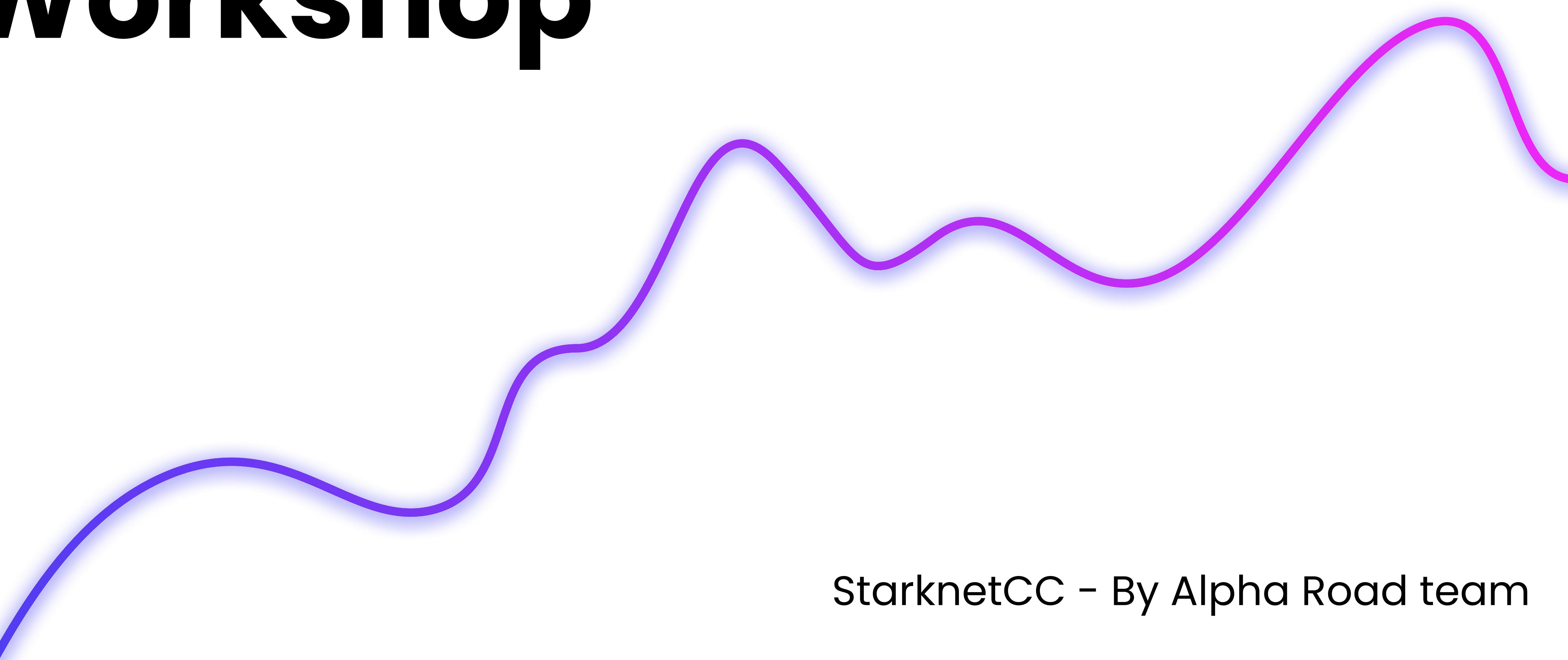
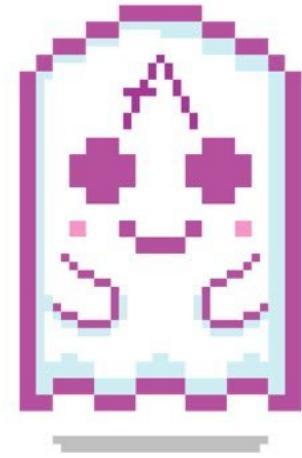


Full Starknet dApp Workshop



StarknetCC - By Alpha Road team

Who we are



Lenny Aebischer

Co-founder & CTO @ Alpha Road

Twitter: 0xlenny_eth



starknet-ecosystem.com



Alpha Road

Florian Bellotti

Senior Software Engineer @ Alpha Road

Twitter: @FlorianBellotti

The workshop (1)

What's the purpose of this workshop

Use case

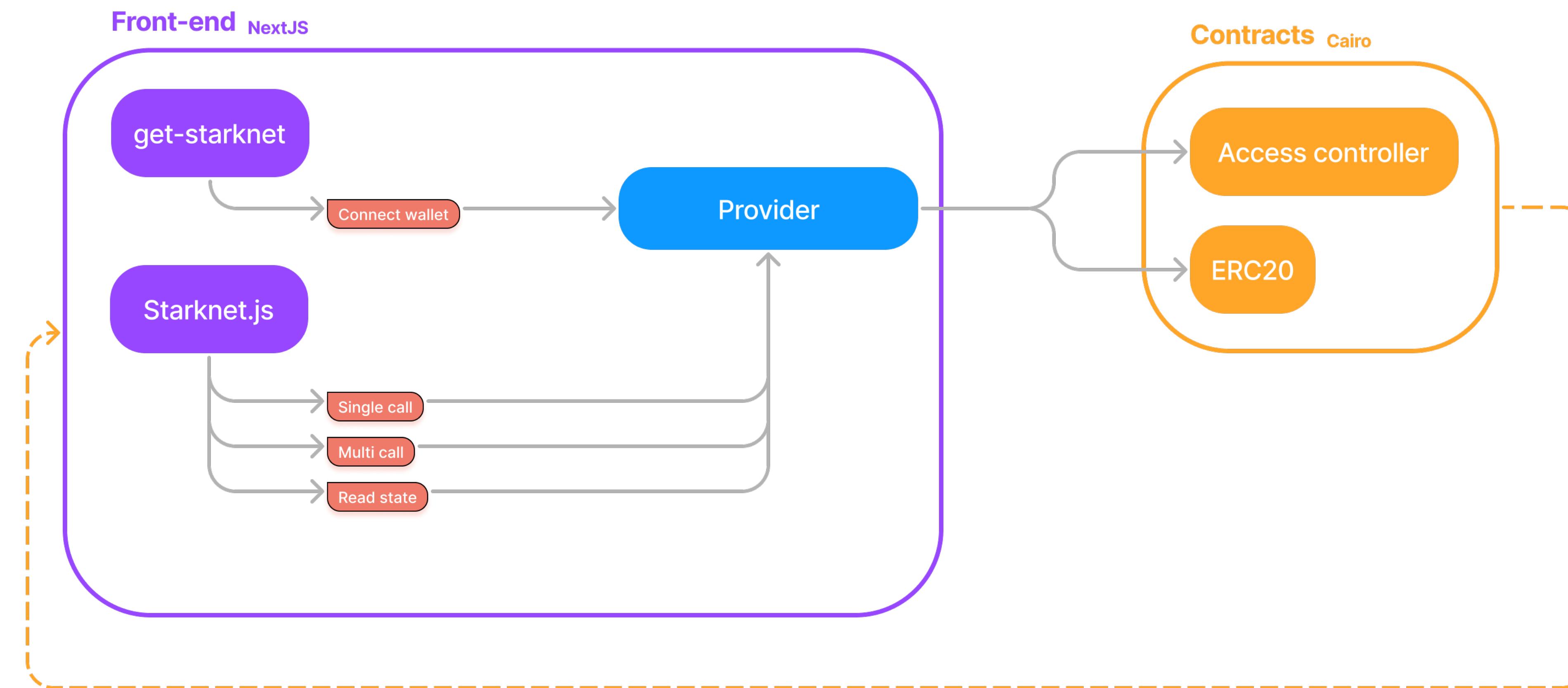
- Contract interactions
 - Start with existing contracts
 - Deploy & connect your own
- Access controller
 - freeSlotsCount
 - isAllowed
 - register
- Minting tokens (multicalls)

Global overview of a full dApp in Starknet

- 1h30 of workshop

The workshop (2)

What you'll work on during this workshop



The technologies

What you need

NextJS

- Built on top of React
- Fast static website that behaves like dynamic
- **SEO**

get-starknet

- Allow dApps to seamlessly connect to any wallet on StarkNet
- Open source / Maintained by community

Starknet.js

- JavaScript library for StarkNet

Cairo

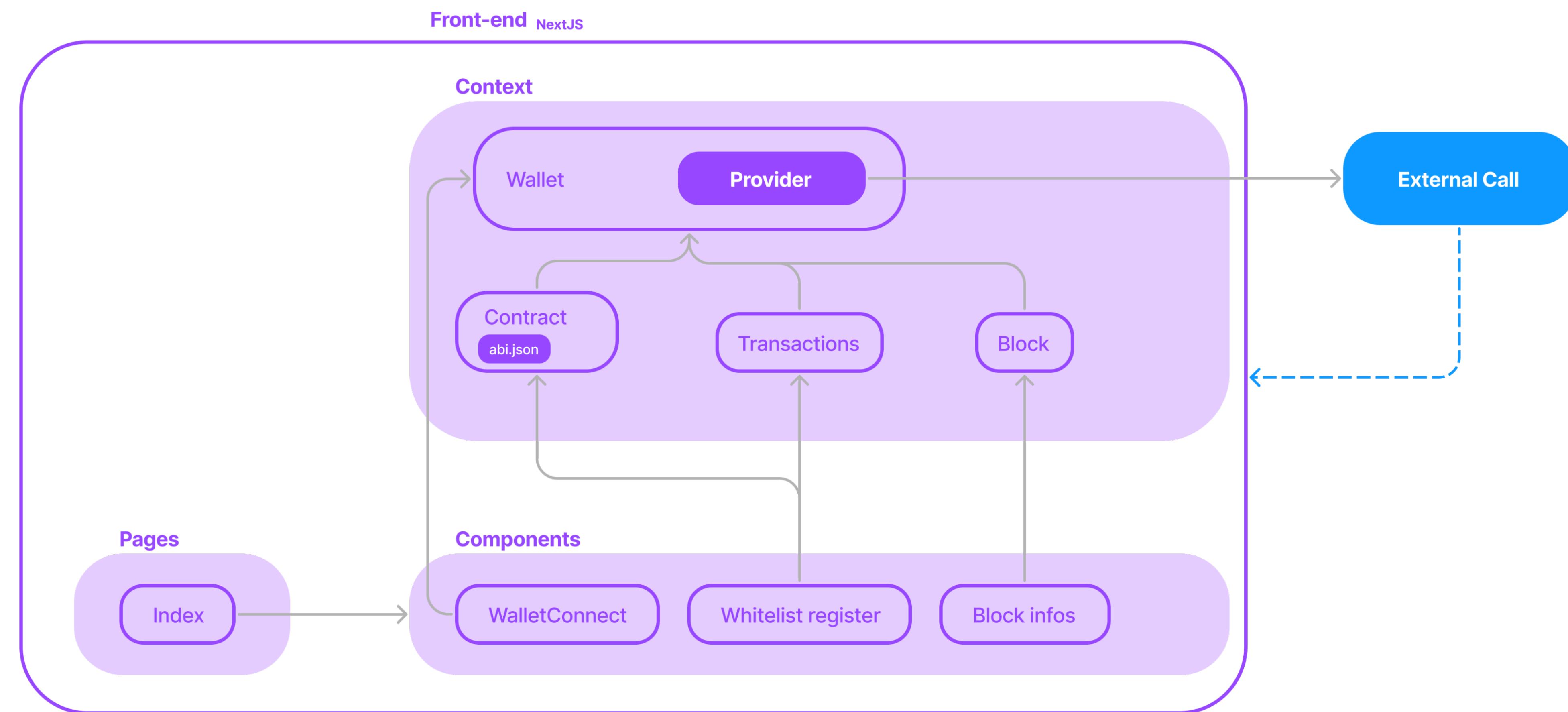
- Programming language for writing provable programs
- StarkNet contracts
- Use of OpenZeppelin standards & coding rules

Tooling

- NPM or Yarn
- Nile
- Cairo-lang

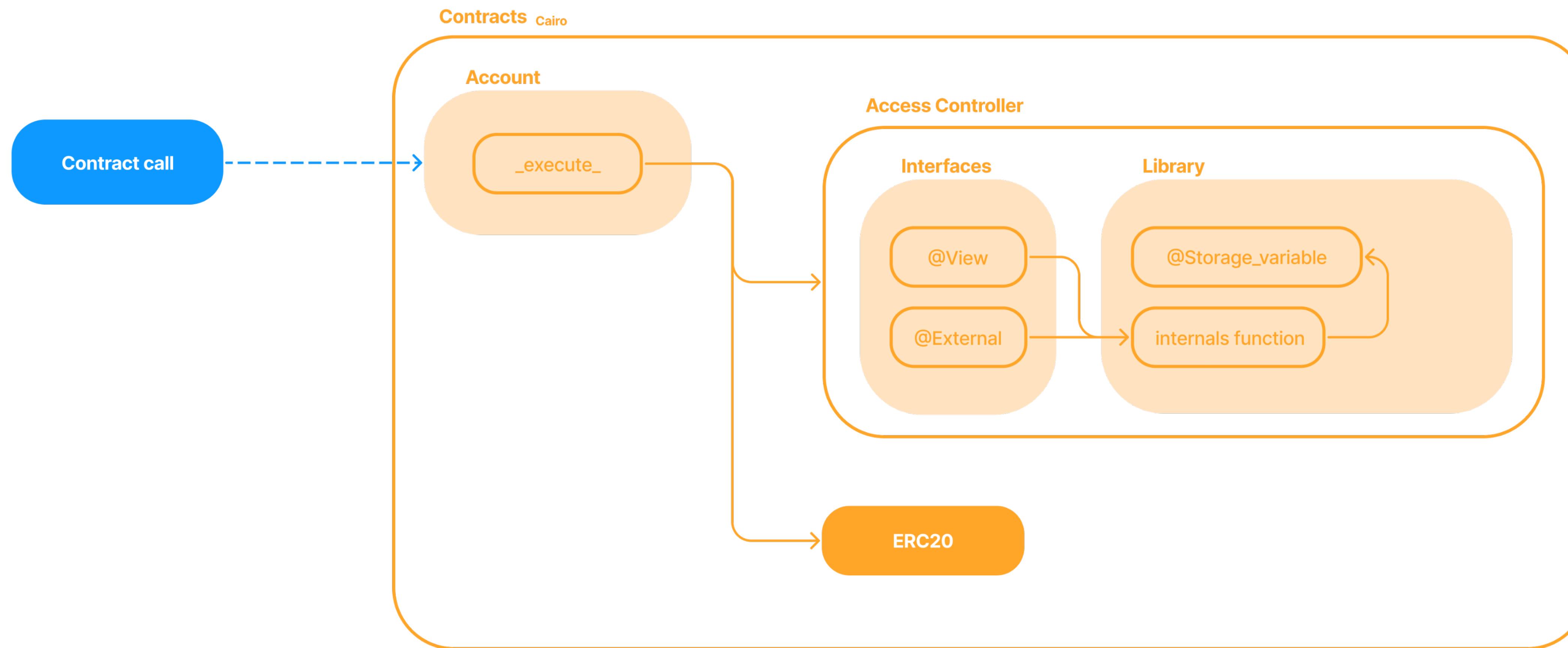
The architecture (1)

Front-end architecture implementation



The architecture (2)

Contracts architecture implementation



The goal

What you have todo

2

- Fetch & display current block infos

1

- Checkout project
- Tooling
- Readme on git

3

- Handling get-starknet lib
- Connect wallet

4

- Connect an existing access controller
- Read free slots / Register status
- Your first transaction-> *Register to whitelist*

5

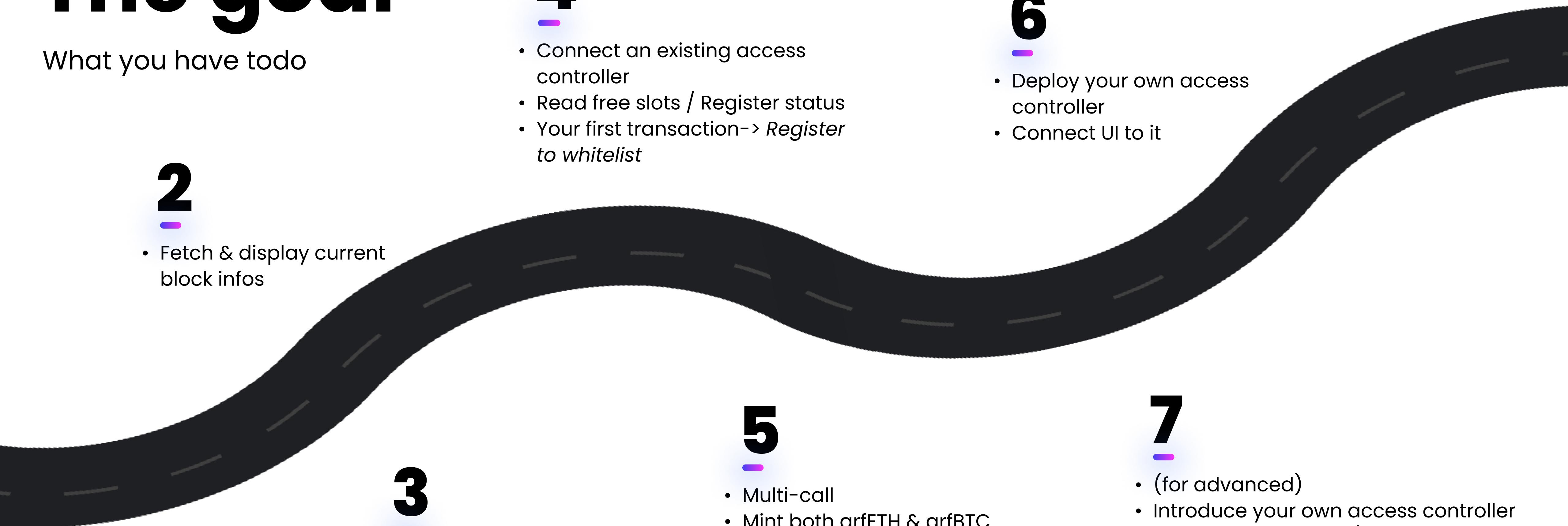
- Multi-call
- Mint both arfETH & arfBTC ERC20 in 1 Tx
- Fetch balances

6

- Deploy your own access controller
- Connect UI to it

7

- (for advanced)
- Introduce your own access controller feature on contract (*Remove access, list of whitelisted, ...*)
- Add a new UI component
- Connect all together



Sample

Some examples

Provider & contract

```
const starknet = await connect({ modalOptions: { theme: "dark" } });

const compiledContractABI = json.parse(
  JSON.stringify(contractABI)
);

const contract: Contract = new Contract(
  compiledContractABI,
  CONTRACT_ADDRESS
);

contract.connect(starknet.provider);

// Block call
provider.getBlock().then((newBlock) => {
  // Do something
});

accessControllerContract
  .invoke("someContractFunction", [])
  .then((response: AddTransactionResponse) => {
    // Do something
  })
  .catch((e) => {
    // Do something
  });

```

multicall

```
const multiCall: Invocation[] = something.map((value) => {
  return {
    contractAddress: contractAddress,
    entrypoint: "someContractFunction",
    calldata: [/*args*/],
  };
});

const account: AccountInterface = starknet.account;

account
  .execute(multiCall)
  .then((response: AddTransactionResponse) => {
    // Do something
  })
  .catch((e) => {
    // Do something
  });

```

9 | Full Starknet dApp

Your turn

Build the future

Clone <https://github.com/419Labs/workshop-arf-dapp>

Start by the README

Hello, 0x70...d4f6

Current Block

Hash: 0x4ba8b43b630239045321c85d8657e53b6a100ef988613272fa6bed4f807a664
Number: 273075
Gas price: 0x56187923

Transactions

0x1709e97803dfeacf72553bd8ad8aaaf3efefa931ea988ee03a2e6a7ae33a3111
ACCEPTED_ON_L2

0x7d009220dbe112a93b530cb1d2cbfa55e010f89b5fb9f64d343e49ba3e0ebbc
ACCEPTED_ON_L2

Register for whitelist

Access controller Contract:
0x027b91a18ed8c3ea77617c1818fb43ab0397144b956506181bf88ab9885a1338

Free slots: 28

Congrats! You are whitelisted

[Check whitelisted](#) [Register to whitelist](#)

Multi-call mint tokens

arfBTC balance: 3000.0 | Contract:
0x72df4dc5b6c4df72e4288857317caf2ce9da166ab8719ab8306516a2fddfff7

arfETH balance: 3000.0 | Contract:
0x3dd7b0db7cca8e8468d06d27b40ca9368754c30d76900fc19a65736fab9084

[Fetch arfBTC balance](#) [Fetch arfETH balance](#) [Mint arfBTC & arfETH in 1 TX](#)

Show in Github

Stars repo not found