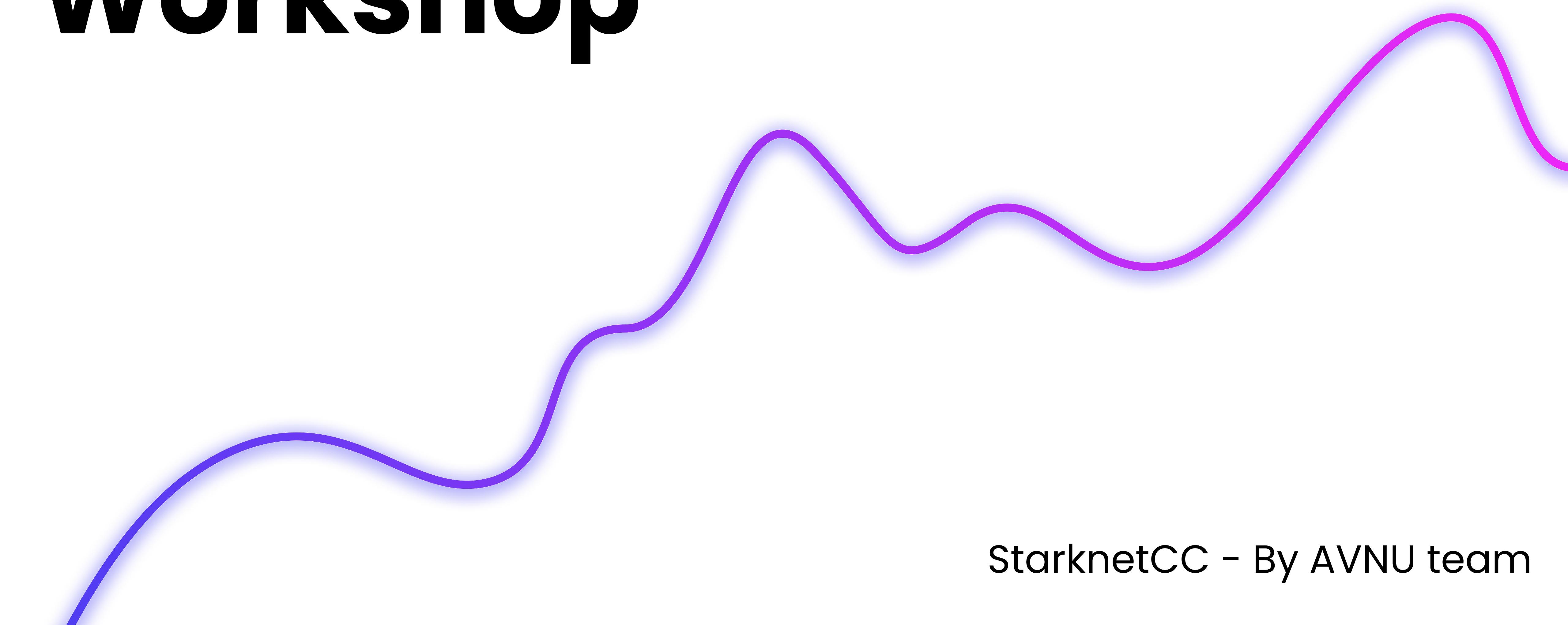


Full Starknet dApp Workshop



StarknetCC - By AVNU team

Who we are



Lenny Aebischer

Co-founder & CTO

Twitter: @0xlenny_eth



Florian Bellotti

Co-founder & Senior Software Engineer

Twitter: @FlorianBellotti



The workshop (1)

What's the purpose of this workshop

Use case

- Contract interactions
 - Start with existing contracts
 - Deploy & connect your own
- Read contract state
 - balanceOf
 - tokenURI
- Minting NFT
 - Simple
 - Multicall (send ETH)

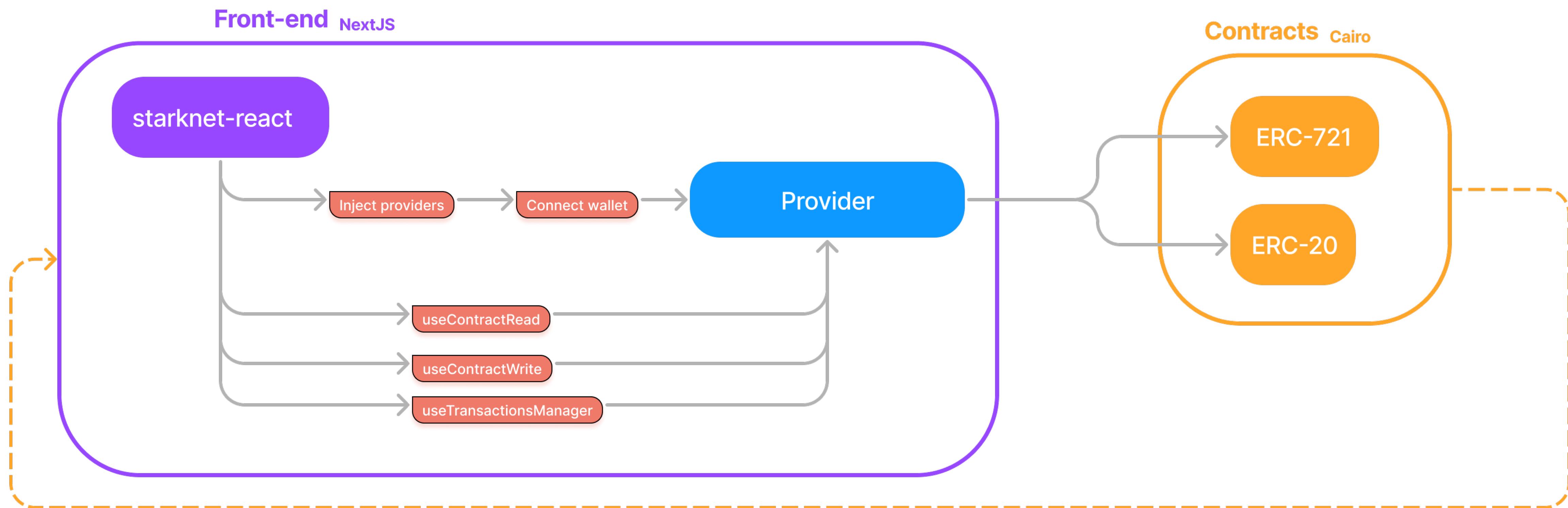
Global overview of a full dApp in Starknet

- 1h30 of workshop

Deploy on **Starknet Goerli Testnet**

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**

Starknet-react

- Collection of React providers and hooks for StarkNet

get-starknet (Argent)

- Support ArgentWebWallet

Cairo

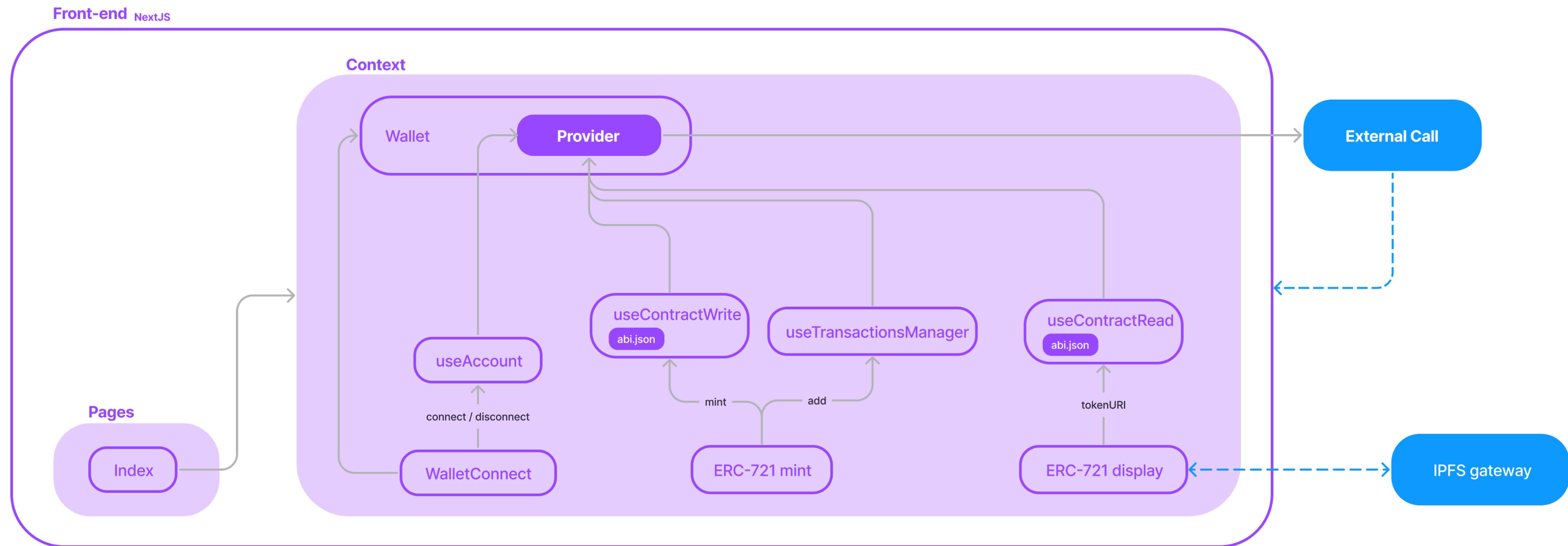
- Programming language for writing provable programs
- StarkNet contracts
- Use of OpenZeppelin standards & coding rules
- **If exists** – Cairo 1 transition

Tooling

- NPM or Yarn
- Scarb
- Starkli

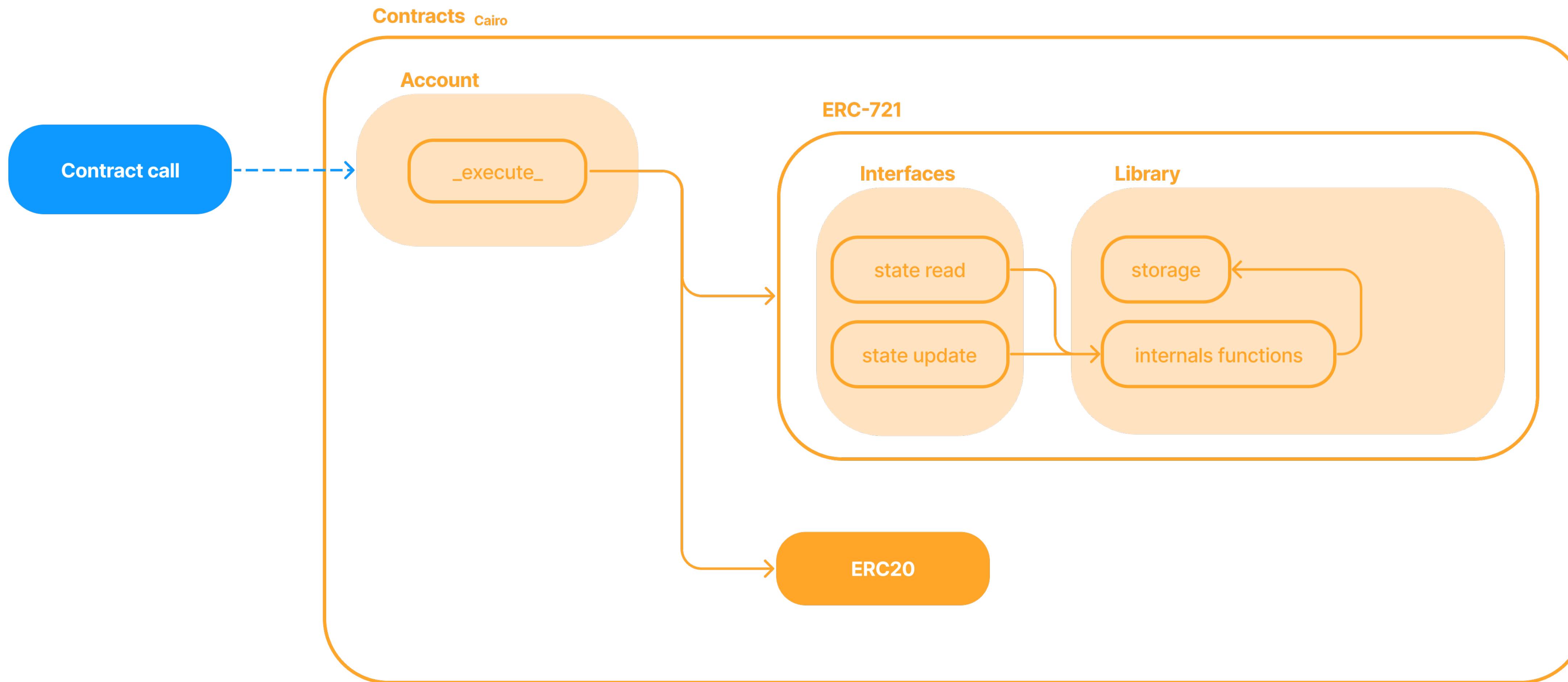
The architecture (1)

Front-end architecture implementation



The architecture (2)

Contracts architecture implementation



The goal

What you have todo

1

- Checkout project
- Tooling
- Readme on git

2

- Fetch & display current block infos

3

- Handling starknet-react
- Inject providers
- Connect wallet

4

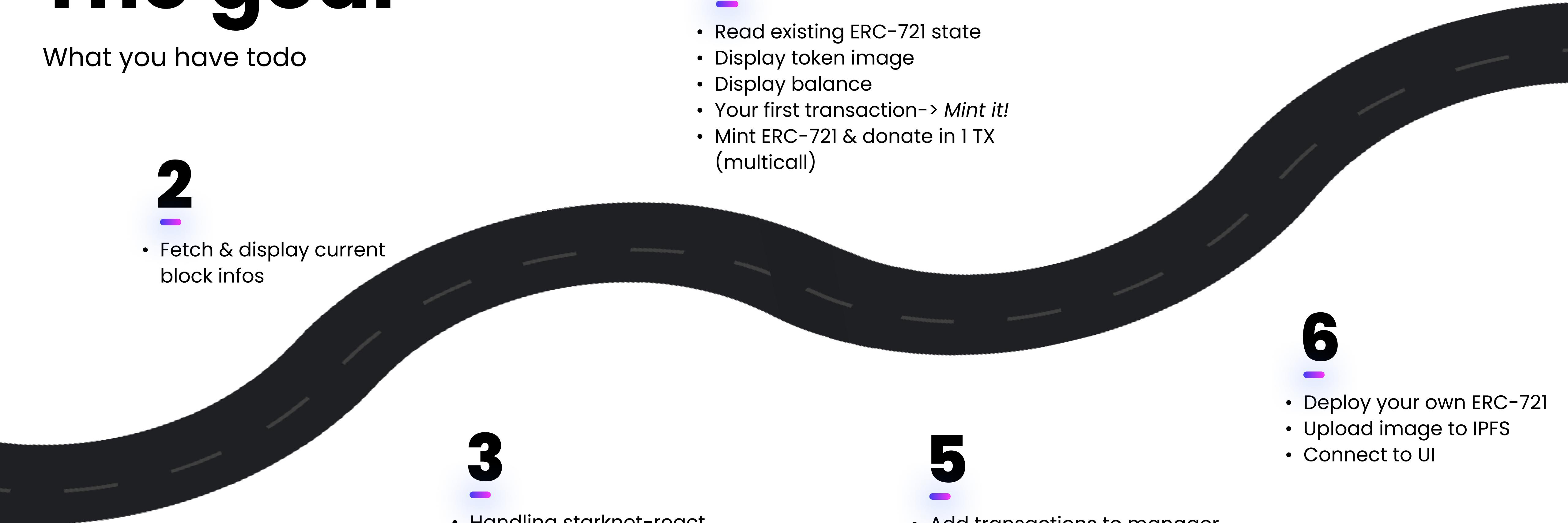
- Read existing ERC-721 state
- Display token image
- Display balance
- Your first transaction-> *Mint it!*
- Mint ERC-721 & donate in 1 TX (multicall)

5

- Add transactions to manager
- Watch transaction status in real time

6

- Deploy your own ERC-721
- Upload image to IPFS
- Connect to UI



Sample

Some examples

Handle connect

```
const { available, connect, refresh } = useConnectors();

useEffect(() => {
  let interval: NodeJS.Timer;
  if (available.length === 0) {
    interval = setInterval(refresh, 0);
  }
  return () => clearInterval(interval);
}, [available, refresh]);

const handleClickConnect = (connector: Connector) => () =>
{ connect(connector);
};
```

useContractWrite

```
const { address } = useAccount();
const { addTransaction } = useTransactionManager();
const ethAddress = "0x49d3...4dc7";
const { data: txData, write } = useContractWrite({
  calls: [
    {
      contractAddress: ethAddress,
      entrypoint: "transfer",
      calldata: [address, 1, 0],
    },
  ],
});

useEffect(() => {
  if (txData) addTransaction({ hash: txData.transaction_hash });
}, [txData, addTransaction]);
```

useContractRead

```
const compiledAvnuNft: Abi = json.parse(JSON.stringify(avnuNft)) as Abi;
const Nft = () => {
  const { data } = useContractRead({
    abi: compiledAvnuNft,
    address: environment.nftAddress,
    functionName: "tokenURI",
    args: [{ low: 1, high: 0 }],
  });
};
```

Your turn

Build the future

Clone <https://github.com/avnu-labs/avnu-dapp-workshop>

Start by the README

AVNU Workshop

Transaction part

Last transaction status

All transactions

ERC-721 part



Name	AVNU Early Adopters
Description	One day, the cosmic wind chance to embrace AVNU revolution intertwined with

Mint NFT

Mint and Donate :)

Balance: 1