# Braavos

Democratizing Banking

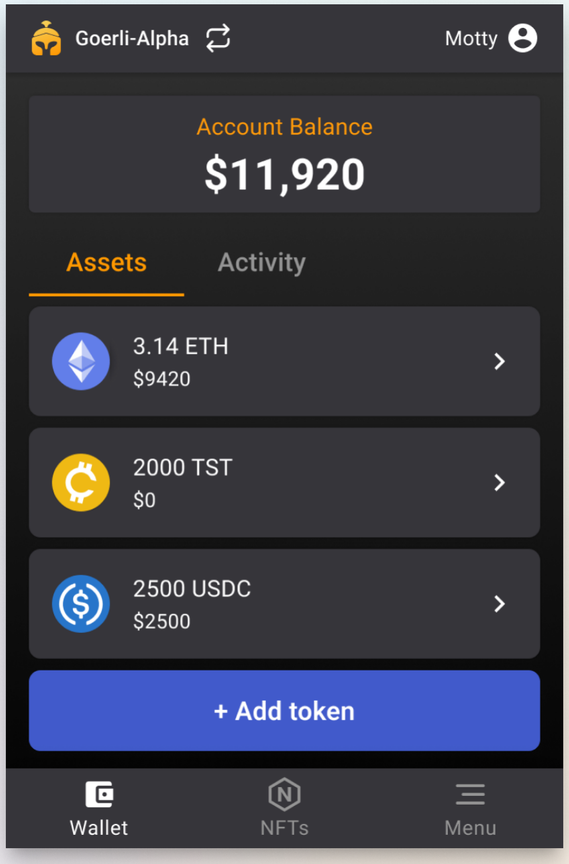
Braavos

# dApp <-> wallet integration

## Goals

By the end of this session you will know how to:

- → **Connect** your dApp with a wallet
- → **Call** your contract on chain
- → **Invoke** txs from your dApp to your contract
- → **Watch assets** e.g. add new tokens to user wallet
- → **Sign messages** to be used in your contracts



Braavos

# Smart Contracts

E2E **hands-on** walkthrough session for everything **wallet** <-> **smart contracts**

Braavos

# Connect -> get-starknet

StarkNet wallet <-> dApp bridge,
**ideation to implementation.**

tl;dr -
Seamless multi-wallet support,
no hassle, focus on coding 💻

Braavos

# Call

→ Call entry-points on your contract

```
const readTokenBalance = async (
    token_address: string = "0x049d36570d4e46f48e99674bd3fcc84644ddd6b96f7c741b1562b82f9e004dc7"
): Promise<string | undefined> => {
    const wallet = getStarknet();
    if (!wallet.isConnected) return undefined;

    const address = wallet.account.address;
    try {
        const result = await wallet.provider.callContract(
            {
                contractAddress: token_address,
                entrypoint: "balanceOf",
                calldata: [BigInt(address).toString()],
            },
            { blockIdentifier: "pending" }
        );
        const balance = composeUInt256(result.result[0], result.result[1]);
        return balance;
    } catch {
        return undefined;
    }
};
```

Braavos

# Invoke

→ mint tokens

```
const mint = async () => {
    const wallet = getStarknet();
    if (wallet.isConnected) {
        const erc20Contract = new Contract(
            Erc20Abi as Abi,
            "0x06e931246fbae79e0453f780ed58a4cb2ff91f7f1c702705c3c1de41a55d9e72",
            wallet.account
        );

        return erc20Contract.mint(
            wallet.account.address,
            parseInputAmountToUint256("100")
        );
    }
};
```

Braavos

# Watch Assets

→ Add new tokens to the wallet

```
const watchAsset = async (): Promise<undefined | boolean> => {
    const wallet = getStarknet();
    return !wallet.isConnected
        ? undefined
        : wallet.request({
            type: "wallet_watchAsset",
            params: {
                type: "ERC20", // The asset's interface, i.e. 'ERC20'
                options: {
                    // The hexadecimal Ethereum address of the token contract
                    address: "0x03e85bfbb8e2a42b7bead9e88e9a1b19dbccf661471061807292120462396ec9",
                    // A ticker symbol or shorthand, up to 5 alphanumerical characters
                    symbol: "DAI",
                    // The number of asset decimals
                    decimals: 18,
                    // A string url of the token logo (optional)
                    image: "...",
                },
            },
        });
};
```

Braavos

# Sign Messages

→ Sign Messages to be used by your contract (i.e. NFT listing for sell)

```
const sign = async (message: string): Promise<Signature | undefined> => {
    const wallet = getStarknet();
    return !wallet.isConnected
        ? undefined
        : wallet.account.signMessage({
            domain: {
                name: "StarkNet DApp",
                chainId:"SN_GOERLI",
                version: "0.0.1",
            },
            types: {
                StarkNetDomain: [
                    { name: "name", type: "felt" },
                    { name: "chainId", type: "felt" },
                    { name: "version", type: "felt" },
                ],
                Message: [{ name: "message", type: "felt" }],
            },
            primaryType: "Message",
            message: { message: "Braavos - first StarkNet mobile wallet" },
          });
};
```

Braavos

# Braavos ❤️ devs

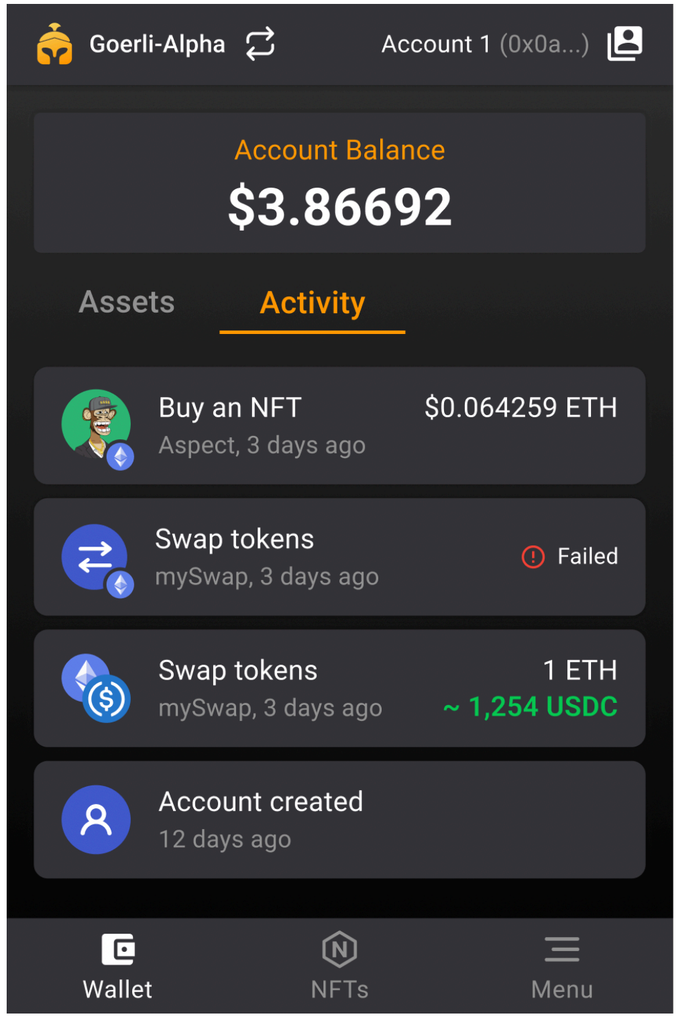
💪 **Preset cloud devnet env** instructions ->

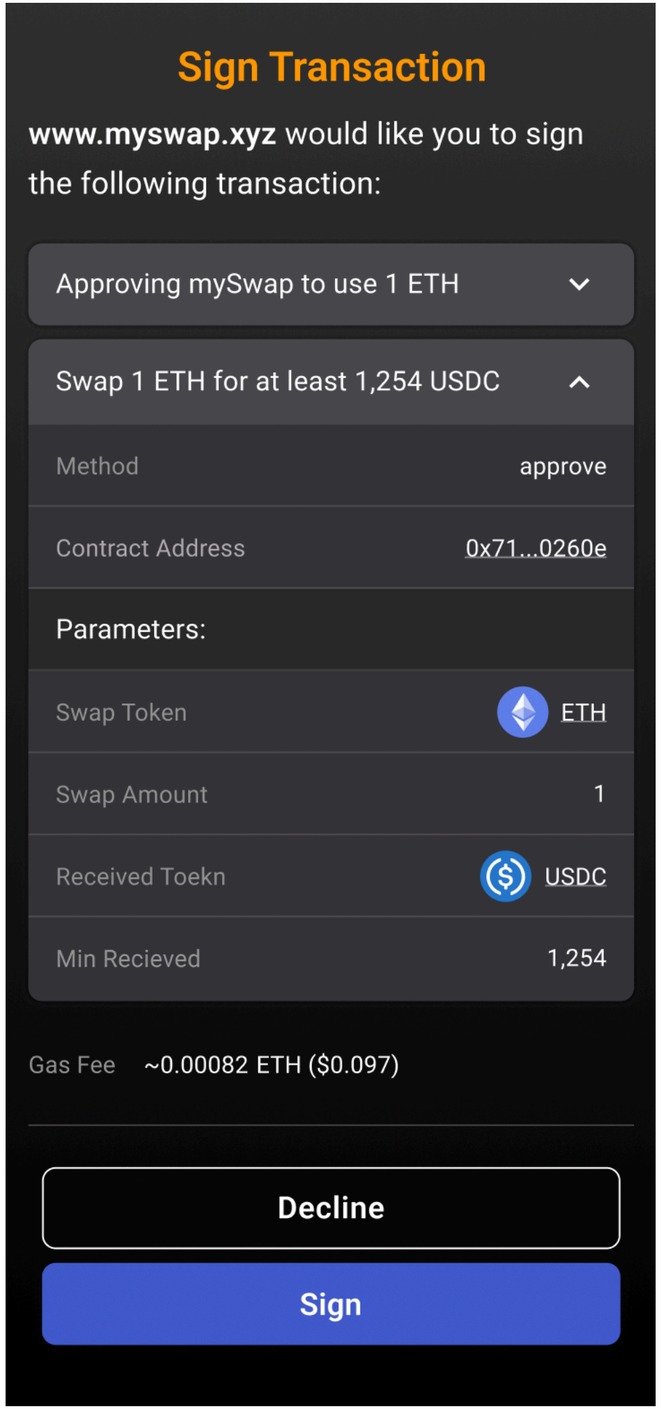
⛽ **mint** ETH inside the wallet

🕶️ **Headless** mode!

Braavos

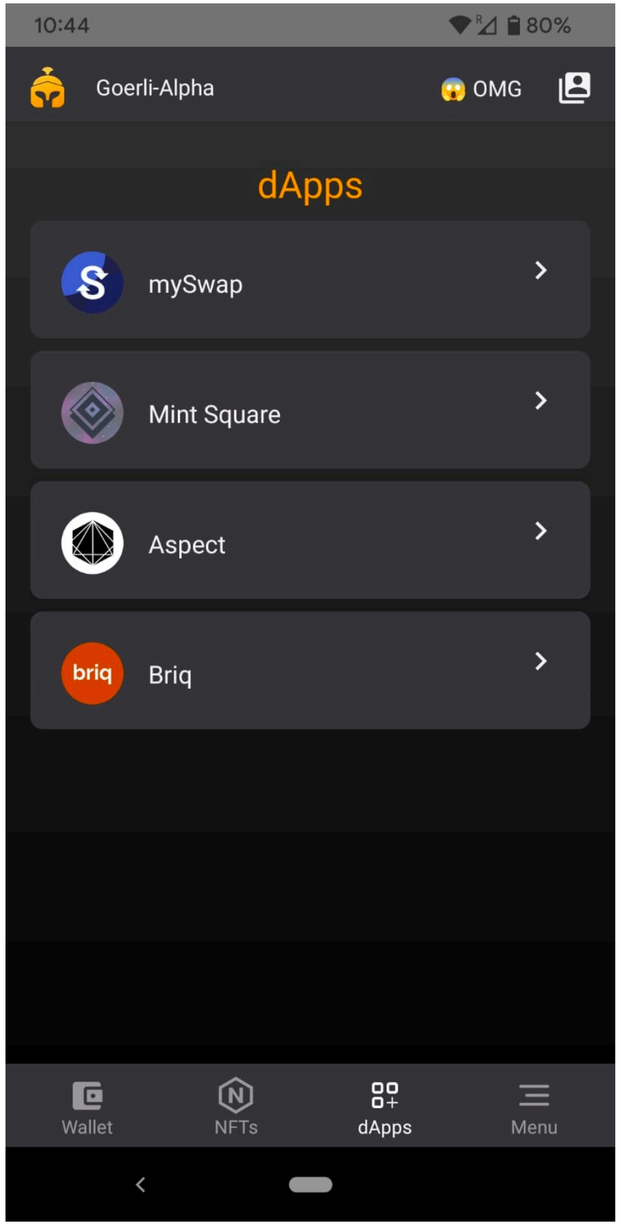
# Transactions -> Explained

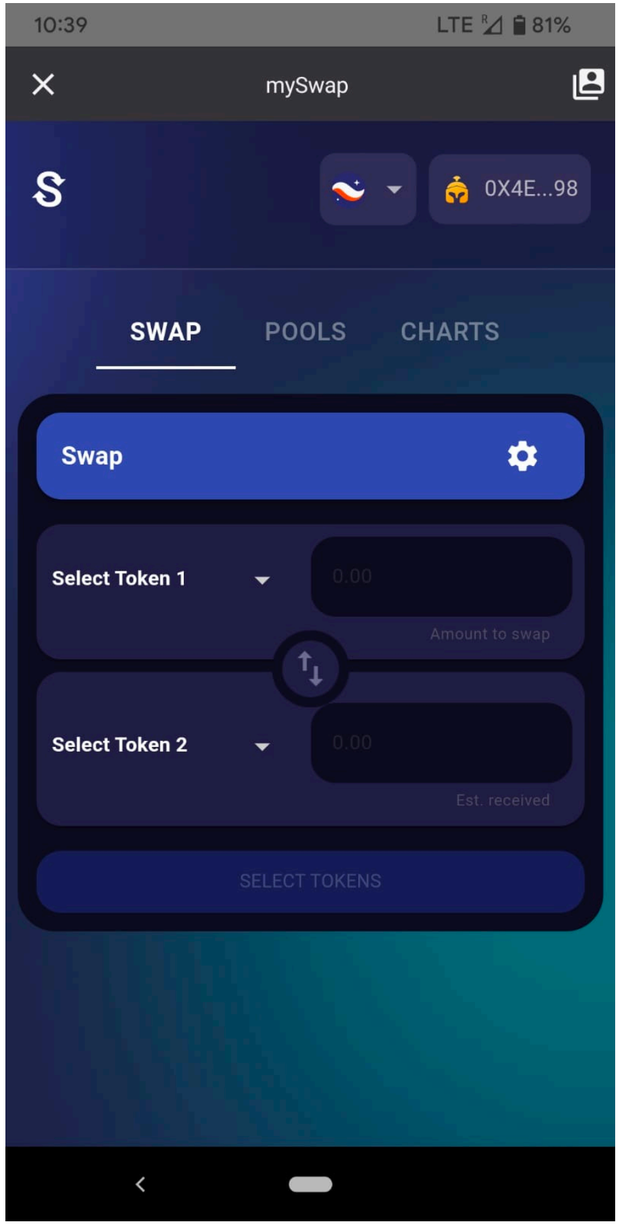


Braavos

# dApps on mobile!



Braavos

# THANK YOU