CESS Network

The Decentralized Data Infrastructure

Episode 9: DApp Development using Solidity Smart Contract

https://www.cess.network

Course Logistics

Course Website: https://course.cess.network/

Episode 1	CESS Network Introduction
Episode 2	CESS Architecture & Key Technologies
Episode 3	CESS Ecosystem and Applications
Episode 4	CESS Nodes & CESS Account Setup
Episode 5	Demo: Running a Consensus Node
Episode 6	Demo: Running a Storage Node
Episode 7	CESS DeOSS and DeOSS REST API
Episode 8	dApp Development using ink! Smart Contract
Episode 9	dApp Development using Solidity Smart Contract
Episode 10	Building Custom Pallet

Why Choose Solidity Smart Contract?

CESS

Extensive Developer Base

Most widely used language with a large and active developer community

Proven Track Record

A mature and battle-tested language with numerous successful projects and applications

Rich Ecosystem and Tooling

Developers benefit from rich ecosystem of development tools, libraries and frameworks

Strong Community Support

The Solidity community is vibrant and continuously contributing to its improvement

Interoperability with EVM Chains

Solidity smart contracts are compatible with any blockchain that supports the Ethereum Virtual Machine (EVM)

Comprehensive Documentation

Solidity has extensive and detailed documentation available for developers

Commonly Used Libraries

Name	Type	Description
Polkadot SDK	Substrate	An umbrella project encompassing three sub-projects: Substrate, Cumulus, and Polkadot.
Polkadot-js API	Substrate	Javascript/Typescript library to interact with Substrate-based blockchains, with utility libs on cryptographic functions.
ether.js	EVM Smart Contract	Library to interact with EVM-compatible smart contracts.
wagmi	EVM Smart Contract / React hook	React hook for EVM-compatible smart contract.

Substrate and EVM Addresses

CESS

CESS Address:

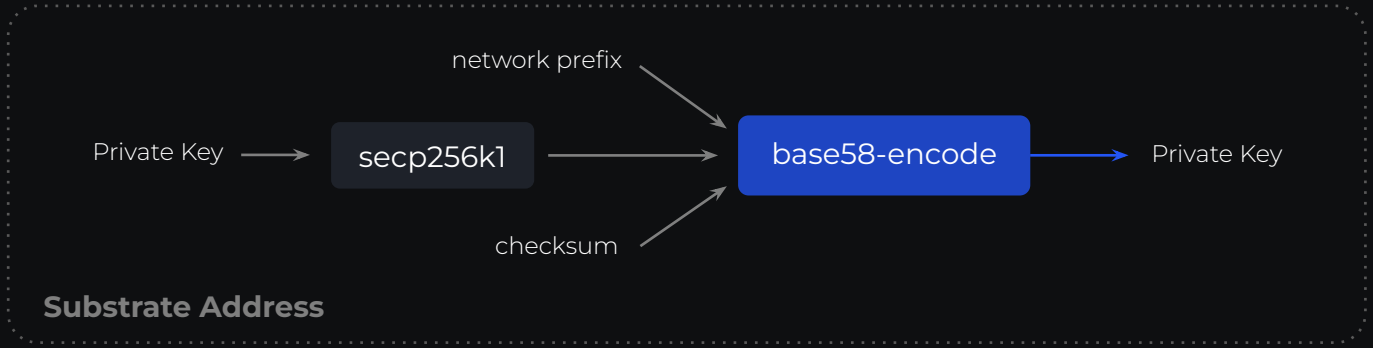
Base58 encoding

- SS58 address: cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef
- Base58 Mapping Table
- Decoded using Base58 encoder/decoder (36 bytes):
- 50acbe7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f85d0
- Pub. key (32 bytes): be7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f

```
                    :~/remote-builds/cess-core$ ./target/debug/cess-node key inspect "cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef"
Public Key URI `cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef` is account:
  Network ID/Version: cess-testnet
  Public key (hex):   0xbe7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f
  Account ID:         0xbe7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f
  Public key (SS58):  cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef
  SS58 Address:       cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef
```

- SS58 address: base58-encode(network prefix, public key, checksum)
- Network prefix: 0x50ac (decimal: 11330, due to additional conversion)
- Checksum: 0x85d0

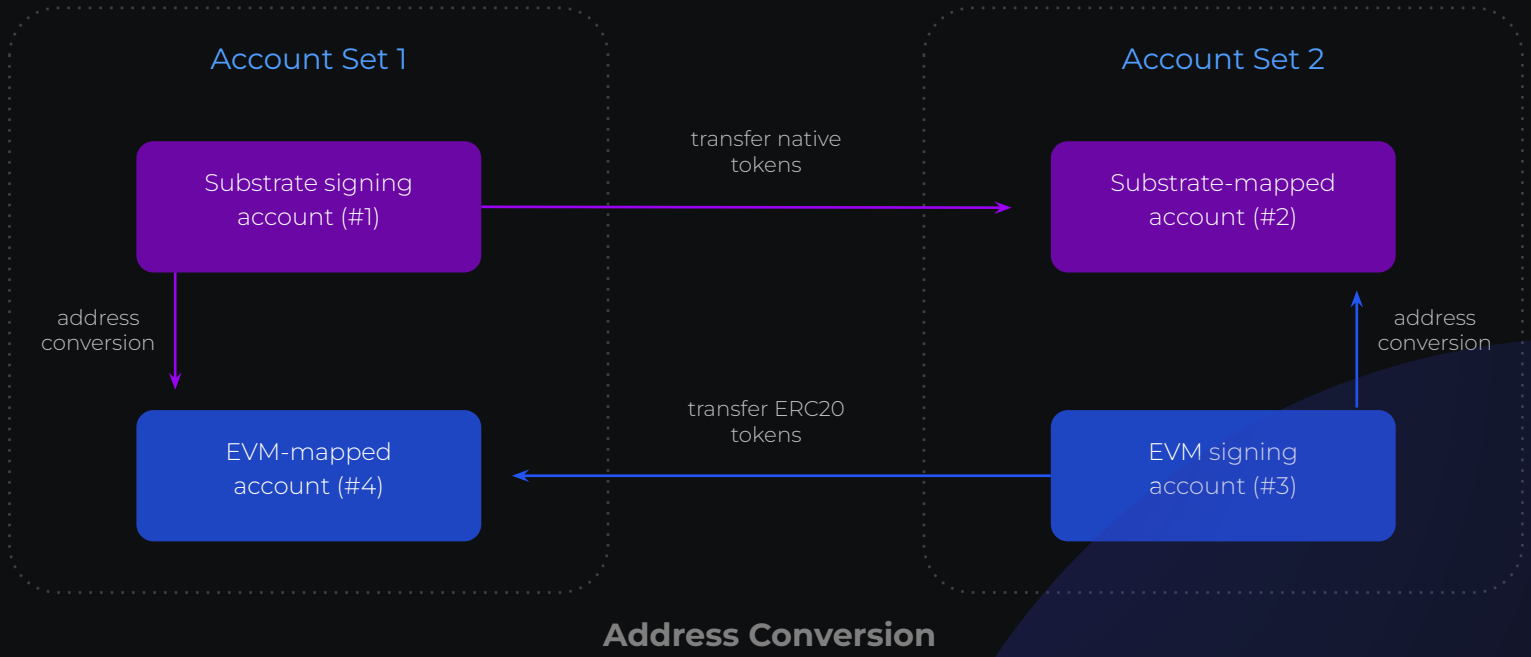
Substrate and EVM Address Generation



EVM address (H160 addr): 0xbe7c1553d878bcd97e5195aede2884c931cd5d28 (20 bytes)

Substrate and EVM Address Conversion

We need **TWO** account sets & an address conversion tool:



CESS

Demo

Transferring Funds Between Substrate and EVM

Demo 1: Transfer from Substrate Signing Acct to EVM Signing Acct

CESS

- CESS Explorer & Metamask installation
- The address conversion tool
- Convert EVM signing address to Substrate-mapped address
- Transfer from Substrate signing address to Substrate-mapped address
- Check account balance in Metamask

Demo 2: Transfer from EVM Signing Acct to Substrate Signing Acct

CESS

- The address conversion tool
- Convert Substrate signing address to EVM-mapped address
- Transfer from EVM signing address to EVM-mapped address
- Withdraw in Substrate signing address with an on-chain transaction
- Check account balance in the CESS Explorer

CESS

Demo

Deploy Solidity Contract on CESS Testnet

Demo: Deploy a Contract on CESS Testnet

- Flipper.sol in CESS example
- Configure *hardhat.config.ts* for CESS Testnet deployment
- Deploy with *hardhat deploy*
- Transfer from Substrate signing address to Substrate-mapped address
- Interact with the contract using Remix

CESS

CESS

Thank you for watching

Please Join Our Community

CESS

CESS Network - Episode 9

DApp Development using Solidity
Smart Contract

https://www.cess.network