

Pointer Contracts

Pointer Contracts enable tokens to be used interoperably in both EVM and CosmWasm environments. Intended to be efficient and quick to deploy, a pointer simply serves as an interpreter between the two "languages" used in either execution environment.

Why Pointer Contracts

A world without pointer contracts. Wallets and clients for feature-rich protocols typically support only a single execution environment. EVM wallets handle ERC-20 coins and ERC-721 NFTs but cannot interact with CosmWasm contracts due to different token standards and interaction methods. The same problem exists for clients built for other protocols (like CosmWasm) which cannot directly interact with Ethereum-based contracts or ERC tokens.

Pointer Contracts solve this by enabling interoperability of either protocol, regardless of the client interface.

With pointer contracts, all tokens can be accessed by wallets of either environment.

How it Works

Pointer contracts enable seamless interoperability. Each smart contract is limited to one associated pointer contract, which must be registered on chain when deploying. This prevents conflicts and provides a reference point for verifying the authenticity of the paired contract itself.

Deploying Pointer Contracts

This section guides you through the process of deploying a Pointer Contract for various token standards using the `seid` CLI tool.

For an in-depth understanding of Sei token standards, click [here](#).

Prerequisites

The list of requirements for deploying a pointer is fairly short:

- Sei light client (CLI/daemon) installed on your machine. You can follow the [installation guide](#)
- if needed.
- Address for the relevant contract (the "pointee")

When a pointer is registered for any given contract, the two are mapped and identifiable across both execution environments. This link is immutable and can only be changed by the original address that created it.

Before deploying a pointer, you should first check if a pointer for the token/contract you wish to deploy already exists by running the following query:

```
seid
```

```
q
```

```
evm
```

```
pointer TYPE POINTEE --evm-rpc=https://evm-rpc.arctic-1.seinetwork.io/ Arguments
```

- TYPE
 - : One of "ERC20", "ERC721", "NATIVE", "CW20", and "CW721". Note that type refers to the underlying token type, so a type of "ERC20" refers to a CosmWasm pointer pointing to an ERC-20 token.
- POINTEE
 - : the underlying token denom literal in the case of "NATIVE" type, or the underlying token contract address for all other types

Flags

- --evm-rpc
 - : The endpoint URL for the EVM RPC interface of the Sei blockchain. This URL is used by the `seid` command to interact with the Sei EVM.

Deploying a Pointer Contract

To deploy a Pointer Contract that links to a CosmWasm contract, you can use the following command:

seid

tx

evm

register-evm-pointer [pointer type] [cw-address] --gas-fee-cap= < cap

--gas-limit= < limit

--evm-rpc= < rpc

Arguments

- pointer type
- : The type of pointer you want to create, either CW20, CW721, or NATIVE.
- cw-address
- : The address of the CosmWasm contract you want to create an EVM pointer for.

Flags

- --gas-fee-cap
- : The maximum amount of gas fees that can be charged for the transaction.
- --gas-limit
- : The maximum amount of gas that can be consumed by the transaction.
- --evm-rpc
- : The endpoint URL for the EVM RPC interface of the Sei blockchain. This URL is used by theseid
- command to interact with the Sei EVM.

Confirmation

After initially registering a pointer, it is advisable to confirm registration by using the following command:

seid

query

evm

pointer [type] [pointee] [flags]

Arguments

- pointer type
- : The type of pointer you want to create, either CW20, CW721, or NATIVE.
- cw-address
- : The address of the CosmWasm contract you want to create an EVM pointer for.

Flags

- --height int
- : Use a specific height to query state at (this can error if the node is pruning state).
- --node string
- ::
- to Tendermint RPC interface for this chain
- --output string
- : Output format (text|json) (default "text").

For a quick-reference for flags/args and syntax, use the--help flag with any command. For example:

seid

tx

evm

--help

CosmWasm >> ERC Pointer

For ERC20 Tokens

To deploy a CW20 Pointer Contract that links to an ERC20 token, you can use the following command:

```
seid  
tx  
evm  
register-cw-pointer  
ERC20 ERC20_TOKEN_ADDRESS --from=SENDER  
--chain-id=arctic-1  
--broadcast-mode=block  
--gas=200000  
--fees=5000usei  
--node=https://rpc.arctic-1.seinetwork.io
```

Arguments

- ERC20_TOKEN_ADDRESS
- : The contract address of the ERC20 token you want to create a CW20 pointer for.

Flags

- --from
- : Keyfile (by name) from which the tx is sent. This address must have enough balance to cover transaction fees.
- --chain-id
- : Identifies the specific chain of the Sei network you're interacting with.arctic-1
- refers to the Sei devnet.
- --broadcast-mode
- : Determines how the transaction is broadcasted. Setting this to block
- means the transaction will wait to be included in a block before returning a response.
- --gas
- : Specifies the maximum amount of gas that can be consumed by the transaction.
- --fees
- : Indicates the transaction fee.
- --node
- : Points to the specific Sei node RPC URL you're connecting to for transaction submission.

i To check out the source code, please visit our [GitHub page\(opens in a new tab\)](#) .

For ERC721 Tokens

To deploy a CW721 Pointer Contract that links to an ERC721 token, you can use the following command:

```
seid  
tx  
evm  
register-cw-pointer  
ERC721 ERC721_TOKEN_ADDRESS --from=SENDER  
--chain-id=arctic-1  
--broadcast-mode=block  
--gas=200000
```

--fees=5000usei

--node=https://rpc.arctic-1.seinetwork.io

Arguments

- ERC721_TOKEN_ADDRESS
- : The contract address of the ERC721 token you want to create a CW721 pointer for.

Flags

- --from
- : Keyfile (by name) from which the tx is sent. This address must have enough balance to cover transaction fees.
- --chain-id
- : Identifies the specific chain of the Sei network you're interacting with.arctic-1
- refers to the Sei devnet.
- --broadcast-mode
- : Determines how the transaction is broadcasted. Setting this to block
- means the transaction will wait to be included in a block before returning a response.
- --gas
- : Specifies the maximum amount of gas that can be consumed by the transaction.
- --fees
- : Indicates the transaction fee.
- --node
- : Points to the specific Sei node RPC URL you're connecting to for transaction submission.

ERC >> CosmWasm Pointer

For CW20 Tokens

To deploy an ERC20 Pointer Contract that links to a CW20 token, you can use the following command:

seid

tx

evm

register-evm-pointer

CW20 CW20_TOKEN_ADDRESS --from=SENDER

--chain-id=arctic-1

--broadcast-mode=block

--gas=200000

--fees=5000usei

--node=https://rpc.arctic-1.seinetwork.io

Arguments

- CW20_TOKEN_ADDRESS
- : The contract address of the CW20 token you want to create an ERC20 pointer for.

Flags

- --from
- : Keyfile (by name) from which the tx is sent. This address must have enough balance to cover transaction fees.
- --chain-id
- : Identifies the specific chain of the Sei network you're interacting with.arctic-1
- refers to the Sei devnet.
- --broadcast-mode
- : Determines how the transaction is broadcasted. Setting this to block
- means the transaction will wait to be included in a block before returning a response.
- --gas
- : Specifies the maximum amount of gas that can be consumed by the transaction.

- --fees
- : Indicates the transaction fee.
- --node
- : Points to the specific Sei node RPC URL you're connecting to for transaction submission.

For CW721 Tokens

To deploy an ERC721 Pointer Contract that links to a CW721 token, you can use the following command:

```
seid
tx
evm
register-evm-pointer
CW721 CW721_TOKEN_ADDRESS --from=SENDER
--chain-id=arctic-1
--broadcast-mode=block
--gas=200000
--fees=5000usei
--node=https://rpc.arctic-1.seinetwork.io
```

Arguments

- CW721_TOKEN_ADDRESS
- : The contract address of the CW721 token you want to create an ERC721 pointer for.

Flags

- --from
- : Keyfile (by name) from which the tx is sent. This address must have enough balance to cover transaction fees.
- --chain-id
- : Identifies the specific chain of the Sei network you're interacting with.arctic-1
- refers to the Sei devnet.
- --broadcast-mode
- : Determines how the transaction is broadcasted. Setting this toblock
- means the transaction will wait to be included in a block before returning a response.
- --gas
- : Specifies the maximum amount of gas that can be consumed by the transaction.
- --fees
- : Indicates the transaction fee.
- --node
- : Points to the specific Sei node RPC URL you're connecting to for transaction submission.

Last updated onMay 27, 2024 [NFT Contracts Multi-Sig Accounts](#)