Table of Contents

Introduction

OpenContracts Smart Contract Protocol Draft

목표

비트코인 블록체인과 같은 smart contract가 embeded 되지 않은 p2p ledger에서 smart contract 개념을 표현하고, 사용하기 위한 protocol.

상세 기술

- OpenAssets과 같이 OP_RETURN을 활용한 transaction상 metadata 기반의 overlay protocol
- global computation & lazy evaluation 지원
  - as an overlay protocol and validation protocol
- Contract Issuance, Contract Execution, Contract Termination의 과정을 거쳐서 사용 됨
- based on
  - OpenAssets
    - https://github.com/OpenAssets/open-assets-protocol
  - multichain smart contract scheme
    - http://www.multichain.com/blog/2015/11/smart-contracts-good-bad-lazy/
  - BIP21 (bitcoin)
    - https://github.com/bitcoin/bips/blob/master/bip-0021.mediawiki
  - ERC-standard-uri-sceheme (ethereum)
    - https://www.bountysource.com/issues/30942695-erc-standard-uri-scheme-with-metadata-value-and-byte-code

Data Models

Blockchain Specific

- Chain ID
  - blockchain URI
    - expected to be standardized by W3C

      Contract Definition
- Contract ID
  - derived from signature (160 bits hash)
- Contract Pointer
  - URI and hash (sha256) of contract body
- Contract Body
  - chain상에 저장 (as metadata)
  - RESTful resource (URI)

Contract Resources

- Scalar values
  - with respect to
    - https://github.com/ethereum/wiki/wiki/Ethereum-Contract-ABI
- URIs
  - on-chain resource

        - contract instances (executed contracts) on-chain
        - transactions
    - RESTful resources

Contract Lifecycle

Contract Issuance

Contract Development

- Solidity와 같은 ethereum smart contract 개발 언어를 사용하여 contract를 개발
- Byte code로 compile

Contract Deployment

- Compile된 contract byte code를 repository에 저장
    - as RESTful resources or on-chain resources

      Contract Definition Transaction
- Repository에 저장된 contract byte code를 reference하는 contract pointer와 해당 contract에 대한 metadata, description등이 포함된 contract definition transaction을 생성
- 여기서 해당 contract의 contract ID가 부여됨
    - Issuance transaction의 첫번째 input 기반으로 contract ID가 상정 됨
- Contract definition transaction에는 OpenContracts protocol market output이 포함 되어야 함

Field	Description	Size
OCP Marker	Magic bytes for OCP (0x4f43)	2 bytes
Version	Version number for protocol (1 = 0x0100)	2 bytes
OP code	Issuance (0x0001)	2 bytes
Contract Pointer	URI and hash (sha256) pointer to contract body (details TBD)	Variable
Metadata	Arbitrary metadata. Can be empty. Should include ABI.	Variable

- Marker output 이후 output은 무시 됨

Contract Execution

- Contract ID가 첫번째 output으로 지정 되어야 함
- Contract ID 이후 첫번째 output으로 OpenContracts protocol marker output이 포함 되어야 함.

Field	Description	Size
OCP Marker	Magic bytes for OCP (0x4f43)	2 bytes
Version	Version number for protocol (1 = 0x0100)	2 bytes
OP code	Execution (0x0002)	2 bytes
Contract Pointer	URI and hash (sha256) pointer to contract body (details TBD)	Variable
Metadata	Arbitrary metadata. Can be empty. Should include ABI.	Variable

- Marker output 이후 output은 무시 됨

Contract Body

- JSON 형식으로 정의

Field	Description	Type
type	Contract executor type (ESC,LSC, ...)	string
hash	SHA256 hash of executable body	hex string
body	Base64 encoded executable code	base64 string

> { type: "ESC", hash: "d9fc201d89fdf94ba708022fe8540b026d5427f799849c8656f26d2eb29aa698", body: "ZDlmYzIwMWQ4OWZkZjk0YmE3MDgwMjJmZTg1NDBiMDI2ZDU0MjdmNzk5ODQ5Yzg2NTZmMjZkMmViMjlhYTY5OA==" }

Contract Termination

- Termination transaction의 첫번째 input 기반으로 contract ID가 상정 됨

Field	Description	Size
OCP Marker	Magic bytes for OCP (0x4f43)	2 bytes
Version	Version number for protocol (1 = 0x0100)	2 bytes
OP code	Termination (0x0003)	2 bytes

- Marker output 이후 output은 무시 됨