

Indy-sdk

NYM, Schema, delete wallet



Indy-sdk



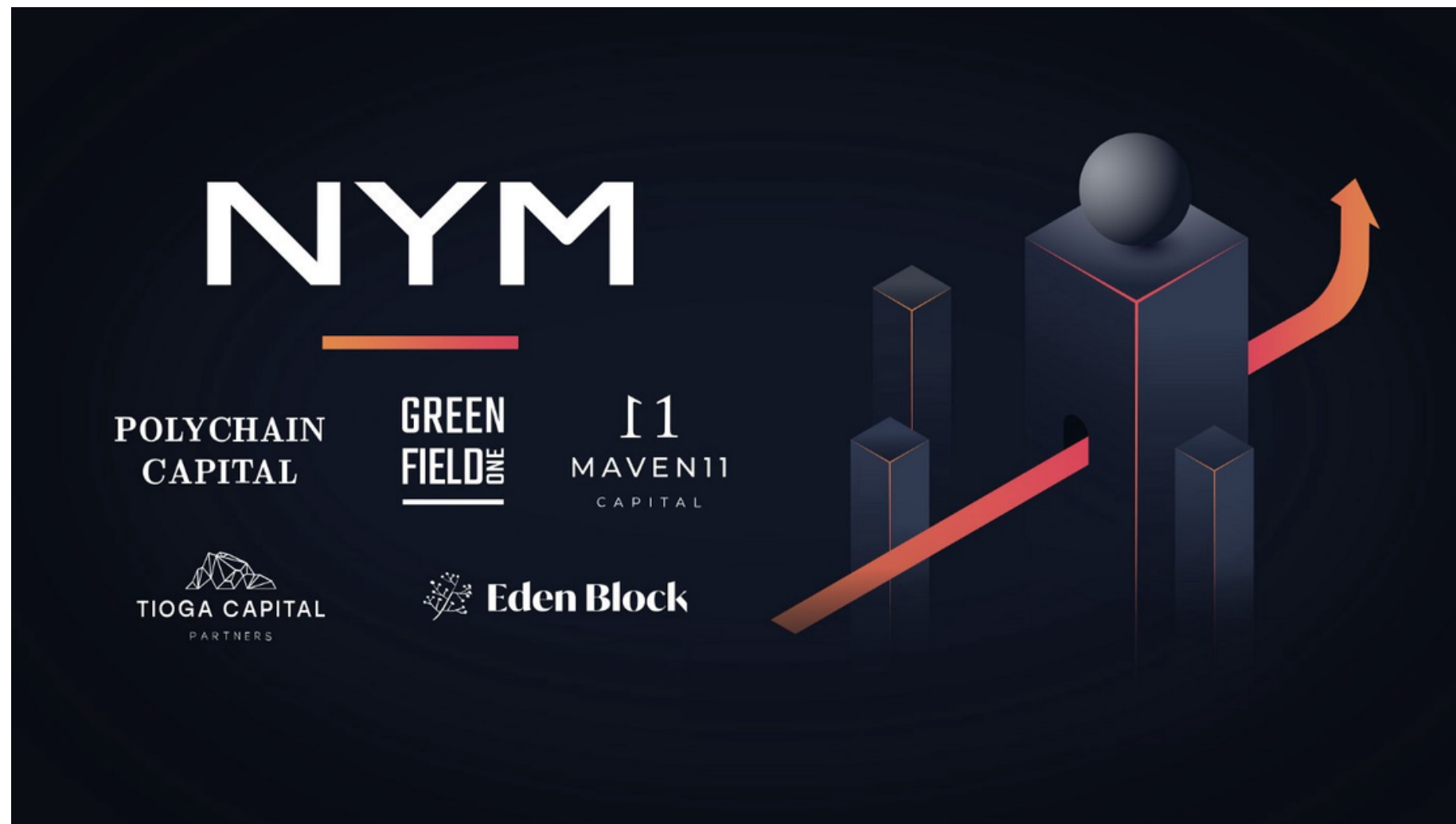
HYPERLEDGER
INDY

다른 주제

NYM

Schema

Deleting wallet



NYM

NYM은 개인 정보 보호와 함께 블록체인 기술을 결합한 분산 신원 관리(DID) 플랫폼으로서 블록체인 기술을 사용하여 사용자의 개인 정보를 안전하게 저장하고, 투명하고 보안성 높은 분산 신원 관리를 할수 있는 서비스 입니다.

1. NYM 생성 요청

```
build_nym_request(submitter_id=steward_id,  
                  target_id=trust_anchor_id,  
                  ver_key=trust_anchor_verkey,  
                  alias=None,  
                  role='TRUST_ANCHOR')
```

write_id_and_query_verkey.py

생성한 did와 verkey 입력

```
await do_call('indy_build_nym_request',  
              c_submitter_id,  
              c_target_id,  
              c_ver_key,  
              c_alias,  
              c_role,  
              build_nym_request.cb)
```

indy/ledger.py

do_call 함수로 실행

```
print_log('\n2. Open pool ledger and get handle from libindy\n')
pool_handle = await pool.open_pool_ledger(config_name=pool_name, config=None)
```

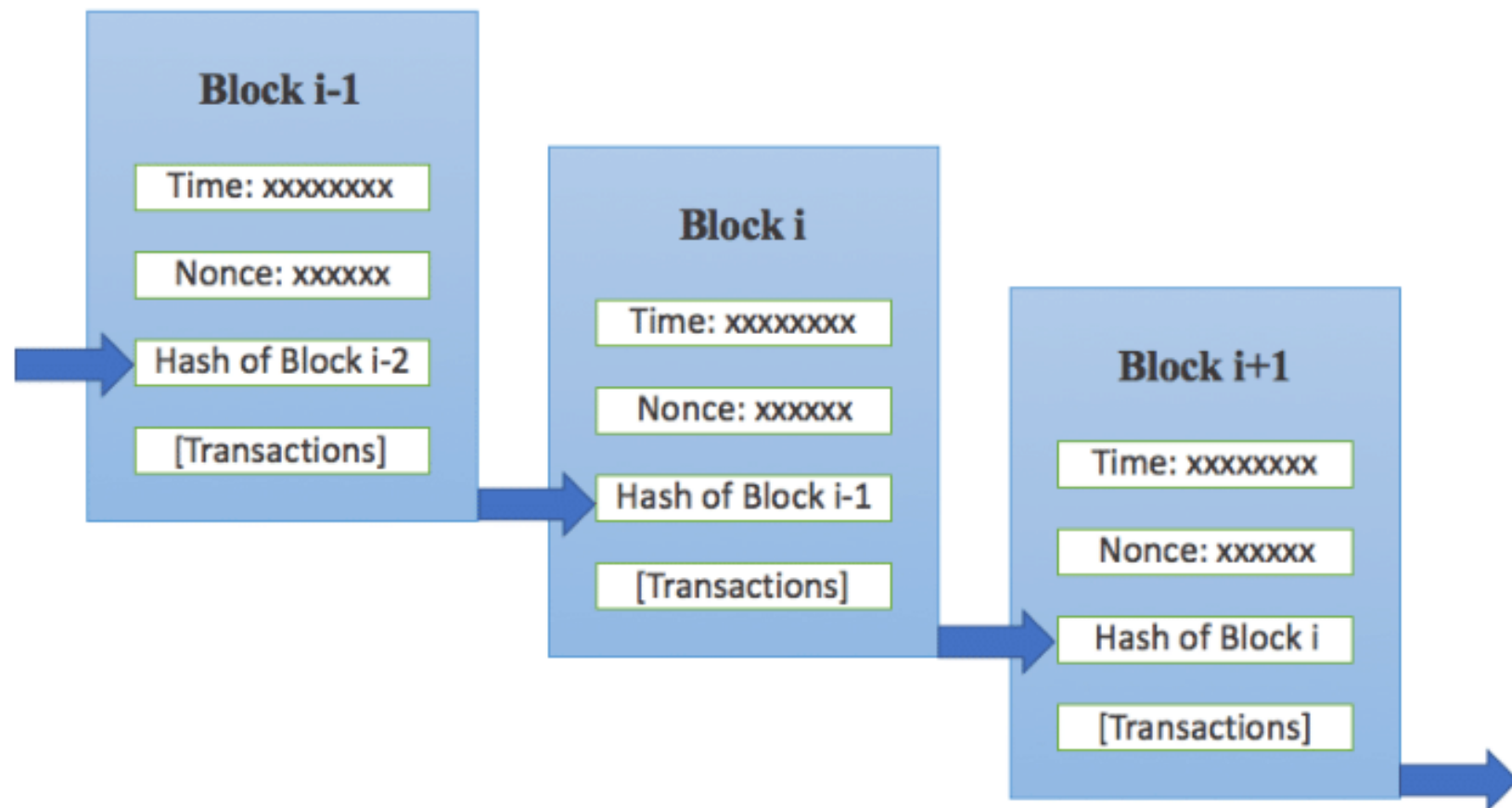
```
print_log('\n4. Open wallet and get handle from libindy\n')
wallet_handle = await wallet.open_wallet(wallet_config, wallet_credentials)
```

2. NYM 서명 및 전송 요청

```
ledger.sign_and_submit_request(pool_handle=pool_handle,
                               wallet_handle=wallet_handle,
                               submitter_id=steward_id,
                               request_json=nym_transaction_request)
```

write_did_and_query_verkey.py

위에서 생성한 핸들러, did를 변수
로 실행



Schema

블록체인의 스키마는 블록, 해시, 거래, 노드, 지갑, 스마트 컨트랙트로 이루어져 있습니다.

1. Schema 생성 및 요청

```
ledger.build_schema_request(steward_did, issuer_schema_json)
```

save-schema-and-cred-def.py

DID와 NYM의 거래요청으로 받은 schema_json을 이용하여 함수 실행

```
ledger.sign_and_submit_request(pool_handle,  
                                wallet_handle,  
                                steward_did,  
                                schema_request)
```

save-schema-and-cred-def.py

do_call 함수로 실행



Deleting wallet

작업 후 지갑을 삭제하고 pool을 닫는 경우는 보안 상의 이유, 불필요한 요소 제거, 업그레이드나 오류 수정 등의 여러 이유가 있습니다.

1. wallet, pool 종료

```
print_log('\n12. Closing wallet and pool\n')
await wallet.close_wallet(wallet_handle)
await pool.close_pool_ledger(pool_handle)
```

write_did_and_query_verkey.py

지갑을 열 때 생성한 핸들러를
이용하여 함수 실행

```
await do_call('indy_close_pool_ledger',
              c_handle,
              close_pool_ledger.cb)
```

```
await do_call('indy_close_wallet',
              c_handle,
              close_wallet.cb)
```

indy/ledger.py, wallet.py

do_call 함수로 실행

2. wallet, pool ledger 설정 삭제

```
# 14.  
print_log('\n14. Deleting created wallet\n')  
await wallet.delete_wallet(wallet_config, wallet_credentials)  
  
# 15.  
print_log('\n15. Deleting pool ledger config\n')  
await pool.delete_pool_ledger_config(pool_name)
```

write_did_and_query_verkey.py

1. 지갑 id와 key를 사용하여 삭제

2. pool 이름으로 삭제

```
await do_call('indy_delete_wallet',  
             c_config,  
             c_credentials,  
             delete_wallet.cb)
```

```
await do_call('indy_delete_pool_ledger_config',  
             c_config_name,  
             delete_pool_ledger_config.cb)
```

indy/ledger.py, wallet.py

do_call 함수로 실행