

Step1 : Connecting to Indy Nodes Pool

Steward

Ledger



Create pool config

After pool configuration is created we can connect to the nodes pool that this configuration describes

Function :

```
pool.create_pool_ledger_config(pool_['name'], pool_['config'])
```

Open pool ledger

Function :

```
pool_['handle'] = await pool.open_pool_ledger(pool_['name'], None)
```

Pool Handle is returned , which is used to reference this opened connection in future libindy calls.

Step 2 : Getting Ownership of Steward's Verinym

Steward's Agent

Wallet



Create wallet
Function :

```
wallet.create_wallet(steward['wallet_config'], steward['wallet_credentials'])
```

Open wallet

```
wallet.open_wallet(steward['wallet_config'], steward['wallet_credentials'])
```

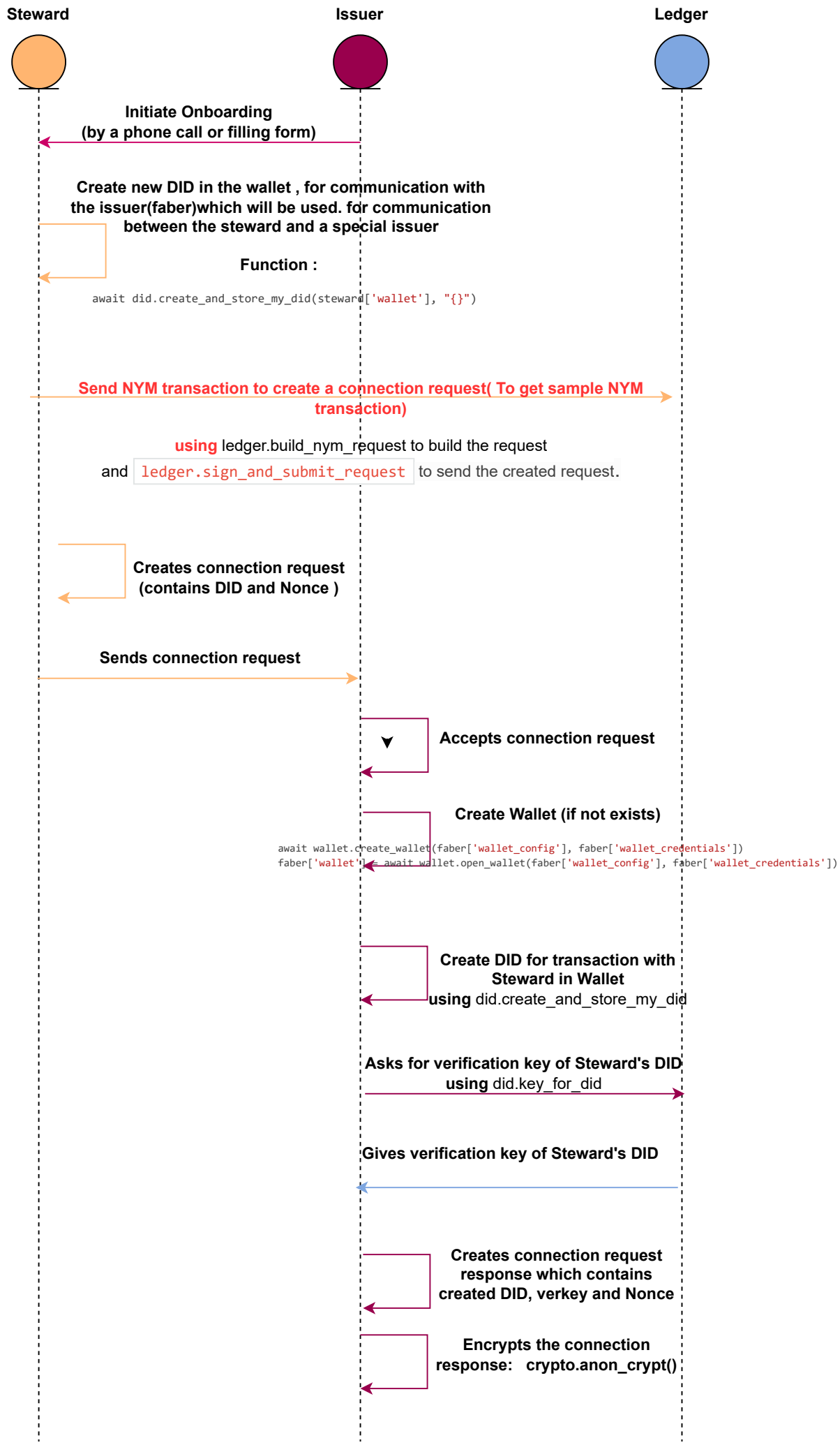
Wallet handle is returned by calling the function for open wallet , which is used to
reference this opened wallet for future calls

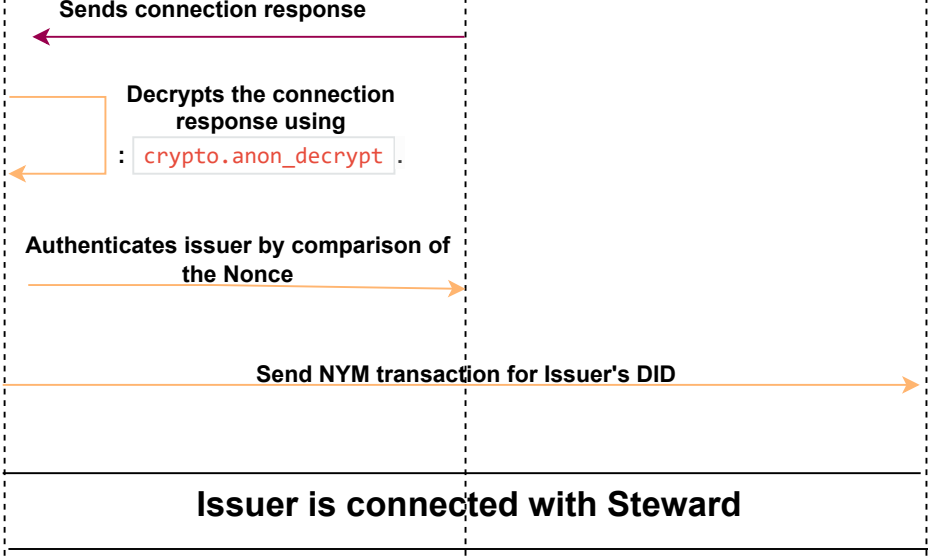
Create and store my DID

```
steward['did'], steward['key'] = await did.create_and_store_my_did(steward['wallet'], steward['did_info'])
```

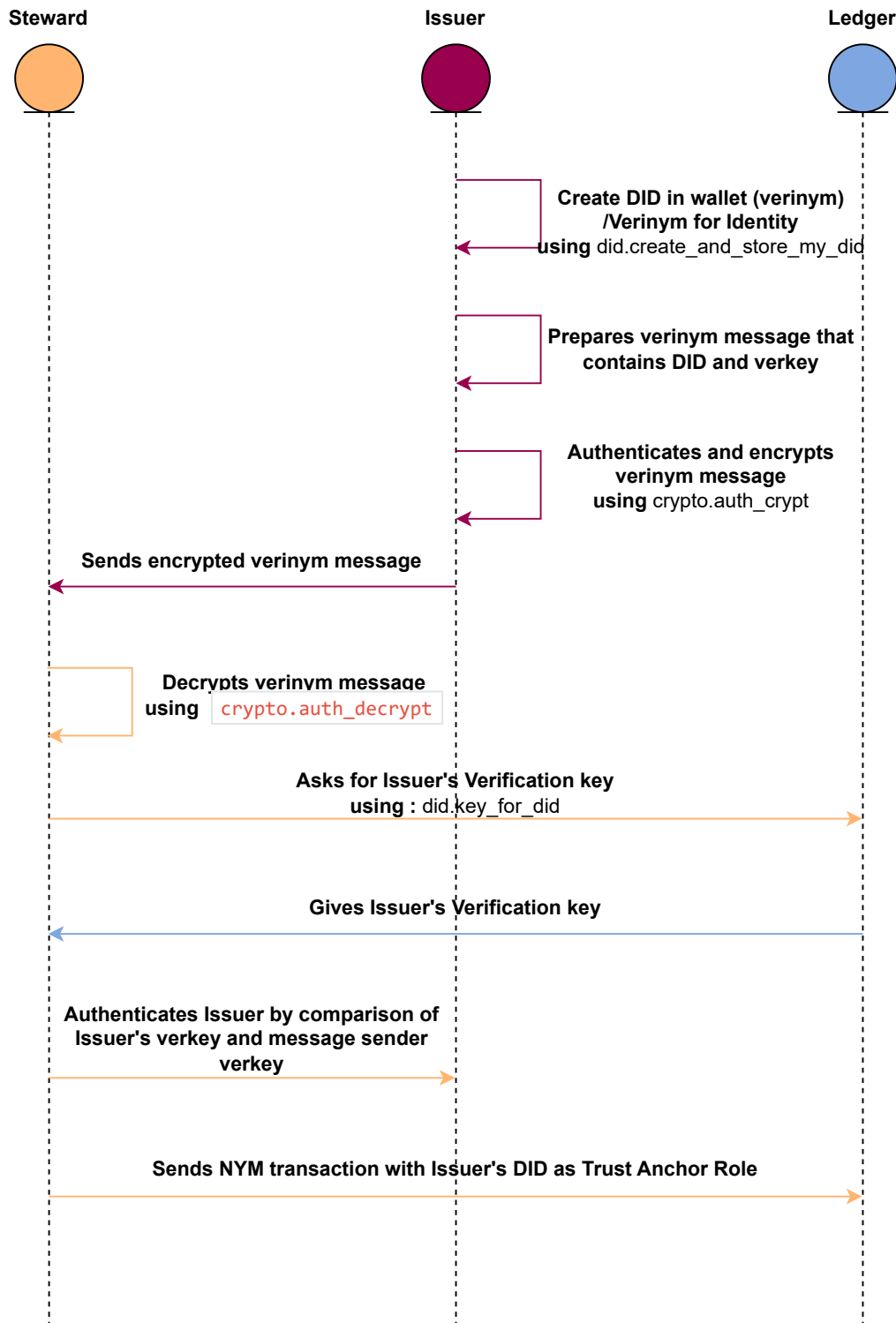
Store DID's signkey, verkey and
metadata

Step 3 : Onboarding of Steward and Issuer

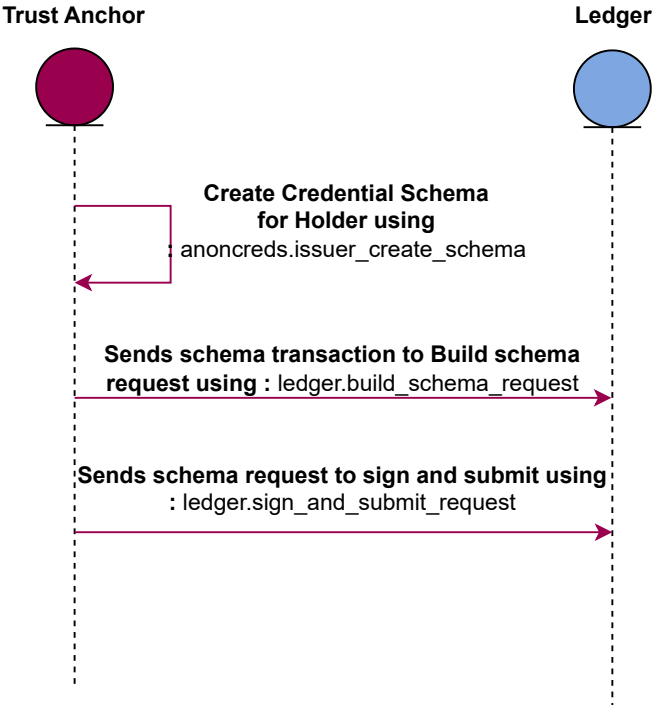




Step 4 : Getting Verinym and Trust Anchor Role for



Step 5: Creating Credential Schema



Step 6: Creating Credential Definition

Trust Anchor

Ledger



Request for Credential Schema using
: ledger.build_get_schema_request

Gives Schema in a format required by
Anoncreds API
using ledger.parse_get_schema_response

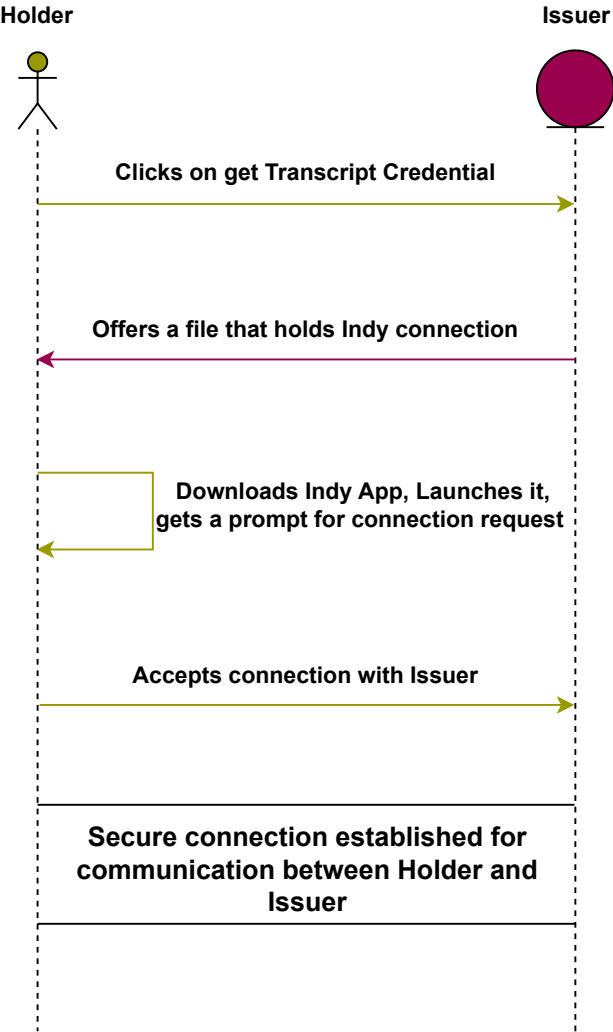
Creates Credential Definition related to Credential
Schema by
calling anoncreds.issuer_create_and_store_credential_def
, public credential definition is returned

Stores private credential definition in
wallet

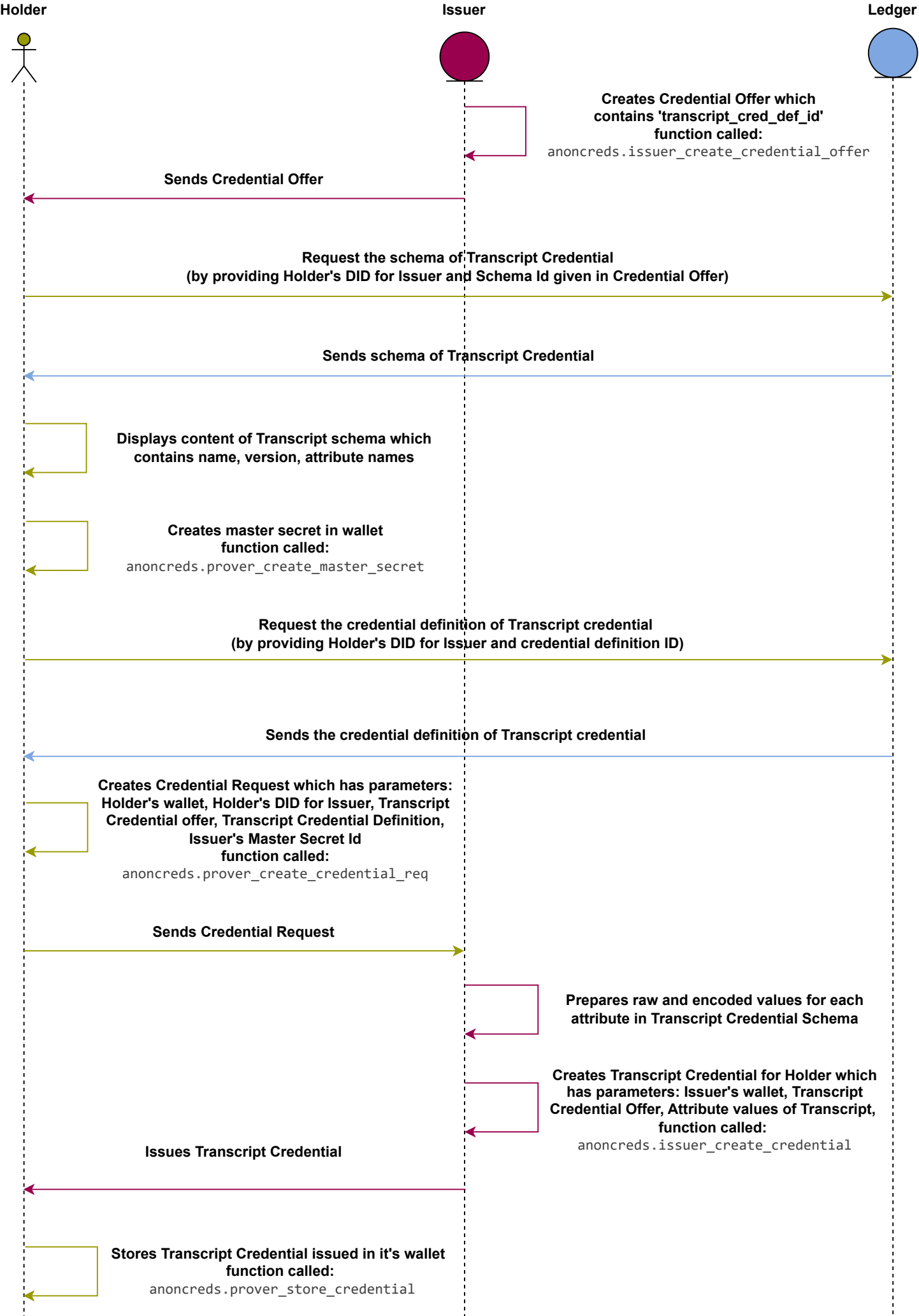
Sends CredDef transaction to build CredDef
request using ledger.build_cred_def_request
and sends the request
using ledger.sign_and_submit_request

Sends CredDef request to sign and submit
using the keys

Connecting Holder with Issuer



Credential Issuance



Generating Proof

Prover

Verifier



Sends proof request

Fetches credentials that match proof request which has two parameters:
Holder's wallet and proof request
function called:

`anoncreds.prover_get_credentials_for_proof_req`

Divides the attributes of matching credentials in three groups: revealed attributes, unrevealed attributes, attributes for which creating verifiable proof is not required

Gets credential schema and credential definition for used credentials.

Creates Proof which has parameters: Holder's wallet, proof request, requested credentials, master secret Id, schemas collection, credential definitions collection, etc.

function called:

`anoncreds.prover_create_proof`

Sends proof which contains revealed attributes, self attested attributes, empty unrevealed attributes and identifiers which has information about used credentials

Verifier Validating Proof

