inside stellar: features that power great developer tools

+ features for building developer tools & products for Stellar

leigh mcculloch

principal software engineer
stellar development foundation

what is great developer tooling?

"great developer tools are powerful *and* delightful. they give you access to everything you need while making you forget you're using a tool at all."

features

01

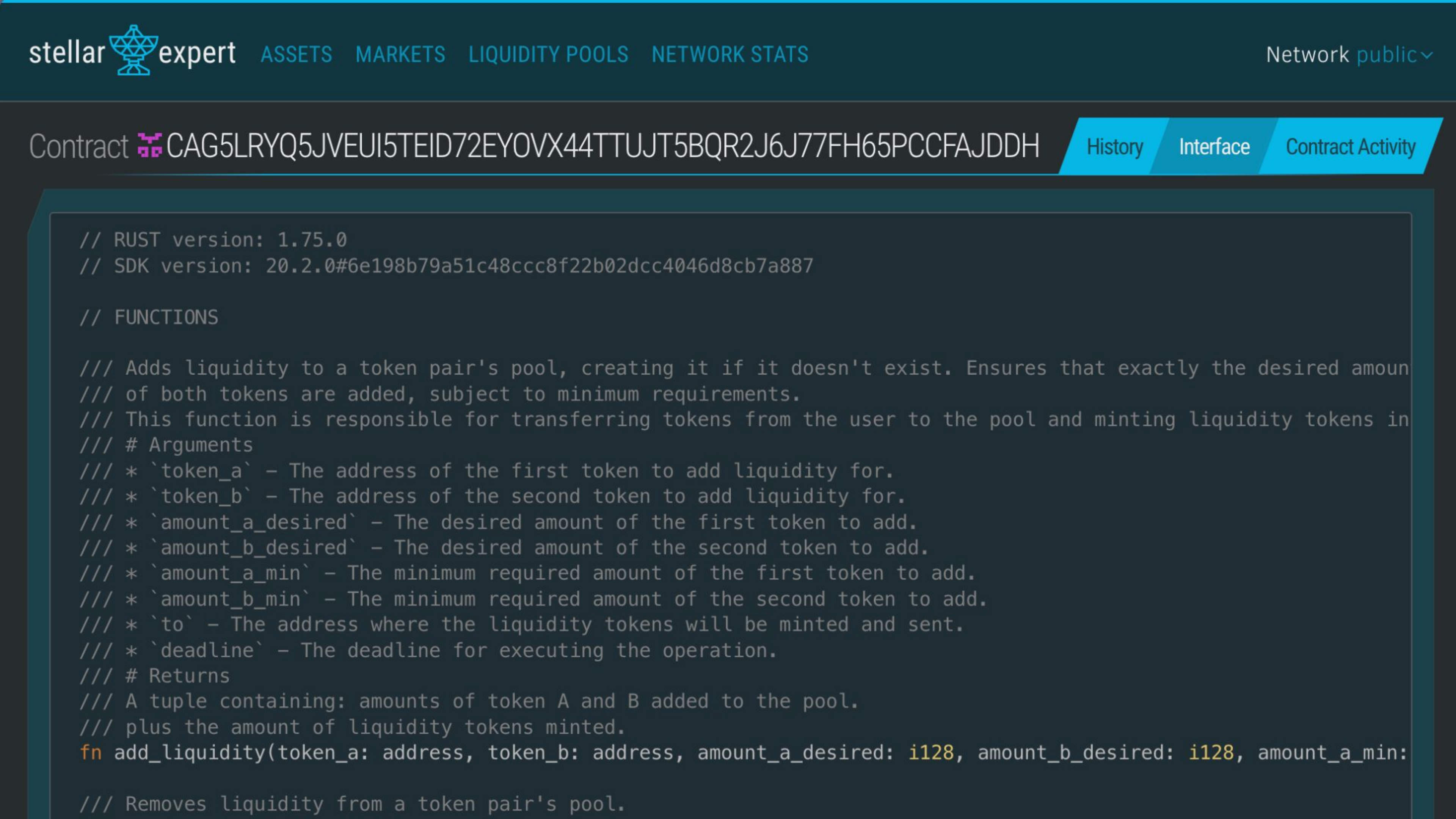
contract specs

02

soroban env

03

xdr json



Contract CAG5LRYQ5JVEUI5TEID72EYOVX44TTUJT5BQR2J6J77FH65PCCFAJDDH

```
// RUST version: 1.75.0
// SDK version: 20.2.0#6e198b79a51c48ccc8f22b02dcc4046d8cb7a887

// FUNCTIONS

/// Adds liquidity to a token pair's pool, creating it if it doesn't exist. Ensures that exactly the desired amoun
/// of both tokens are added, subject to minimum requirements.
/// This function is responsible for transferring tokens from the user to the pool and minting liquidity tokens in
/// # Arguments
/// * `token_a` - The address of the first token to add liquidity for.
/// * `token_b` - The address of the second token to add liquidity for.
/// * `amount_a_desired` - The desired amount of the first token to add.
/// * `amount_b_desired` - The desired amount of the second token to add.
/// * `amount_a_min` - The minimum required amount of the first token to add.
/// * `amount_b_min` - The minimum required amount of the second token to add.
/// * `to` - The address where the liquidity tokens will be minted and sent.
/// * `deadline` - The deadline for executing the operation.
/// # Returns
/// A tuple containing: amounts of token A and B added to the pool.
/// plus the amount of liquidity tokens minted.
fn add_liquidity(token_a: address, token_b: address, amount_a_desired: i128, amount_b_desired: i128, amount_a_min:

/// Removes liquidity from a token pair's pool.
```

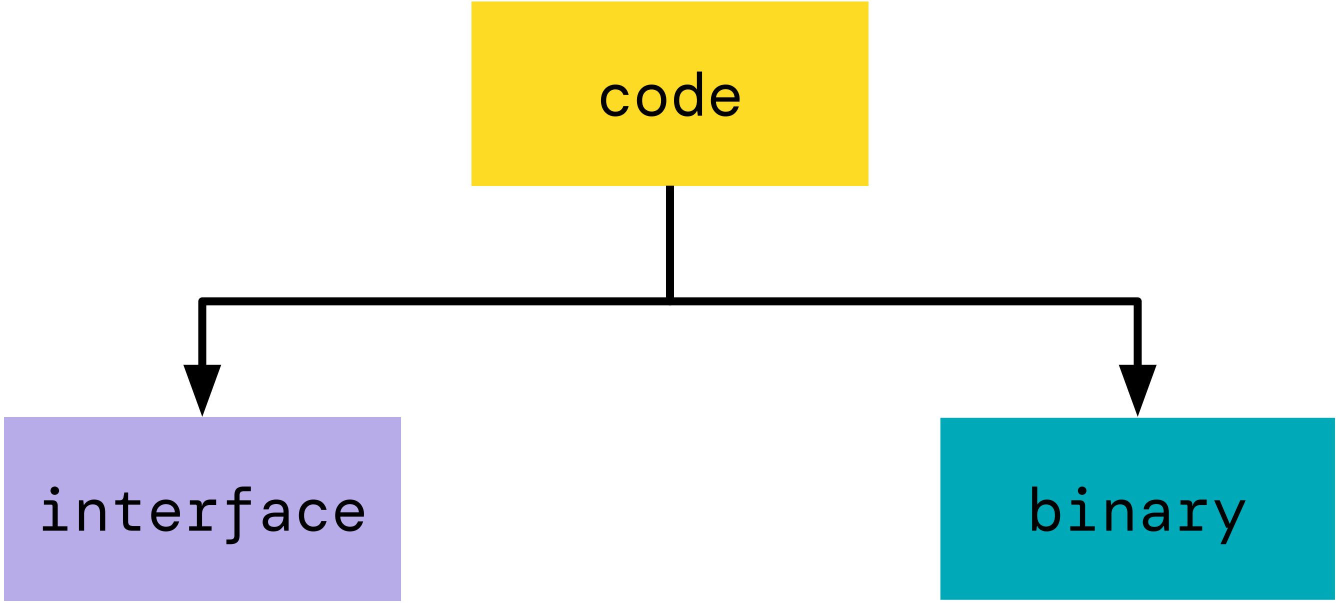
01 contract specs

stellar.org/protocol/sep-48

01

contract specs idl

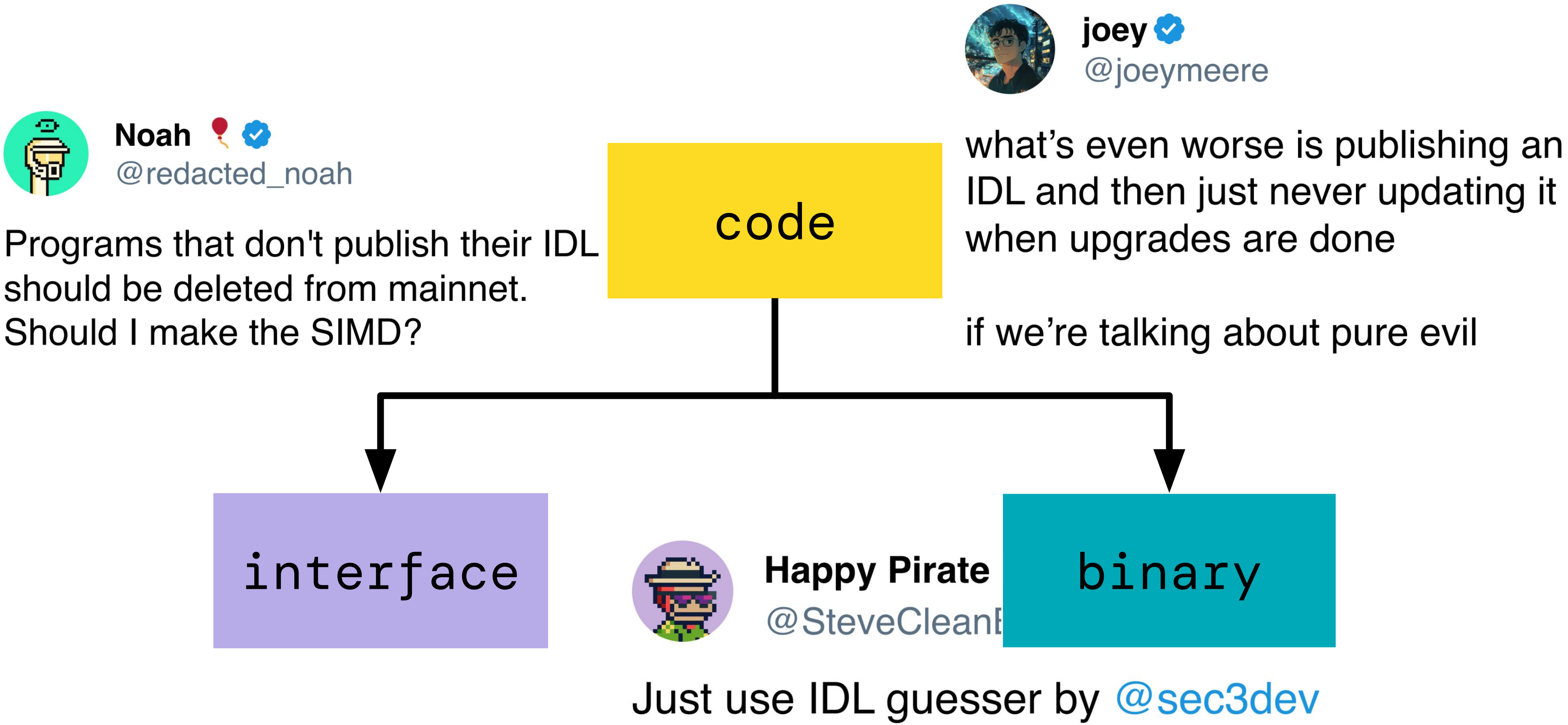
stellar.org/protocol/sep-48



code

interface

binary



Noah ✓
@redacted_noah

Programs that don't publish their IDL should be deleted from mainnet. Should I make the SIMD?

joey ✓
@joeymeere

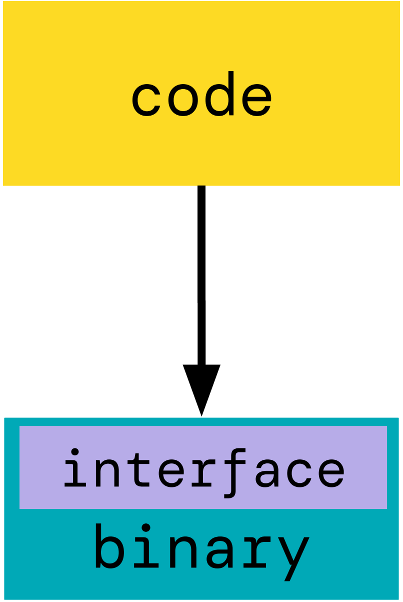
what’s even worse is publishing an IDL and then just never updating it when upgrades are done

if we’re talking about pure evil

Happy Pirate
@SteveClean

Just use IDL guesser by @sec3dev

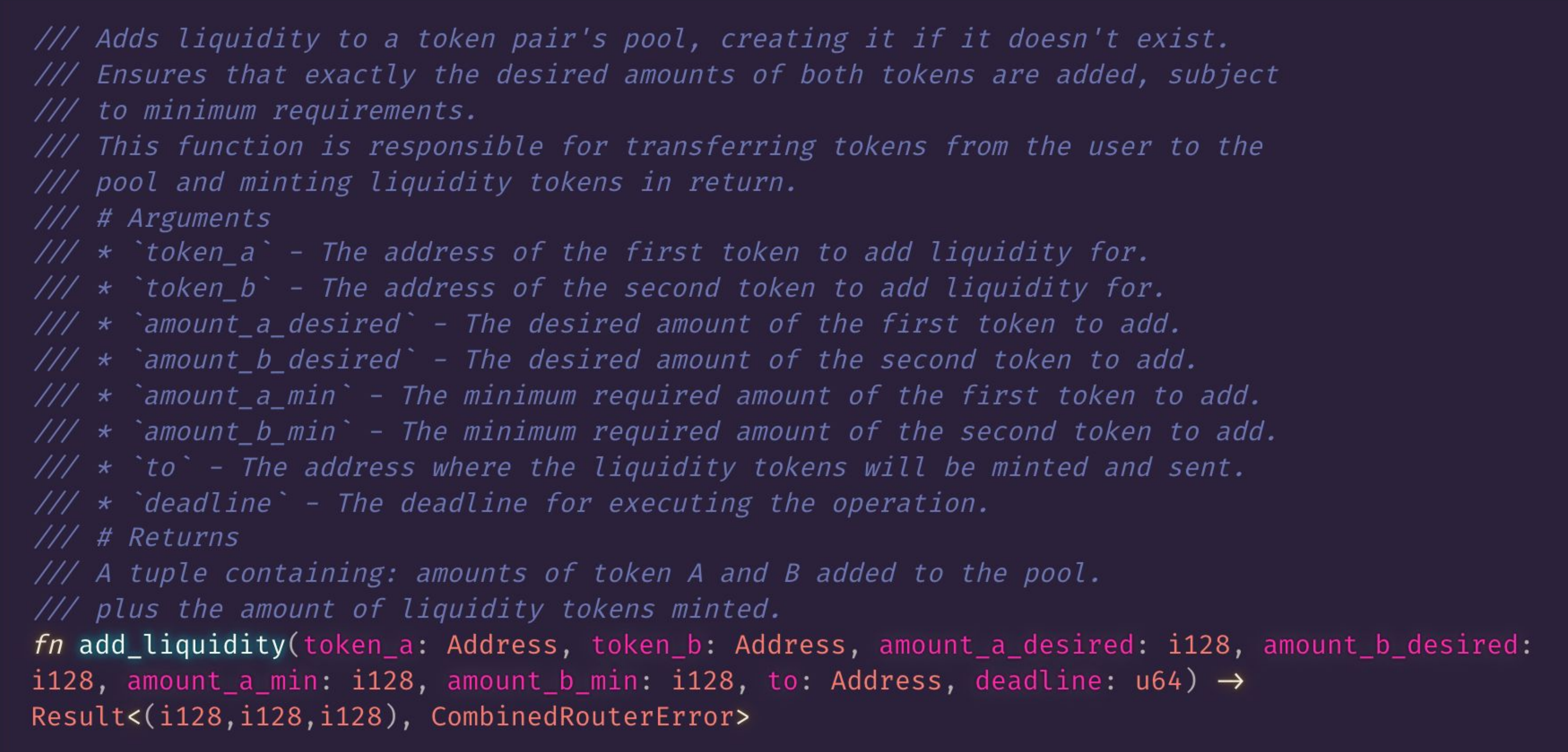
https://x.com/redacted_noah/status/1957498243572723846
https://blog.syndica.io/deep-dive-solana-on-chain-activity/
https://sec3.dev/blog/idl-guesser-recovering-instruction-layouts-from-closed-source-solana-programs



code

interface

binary



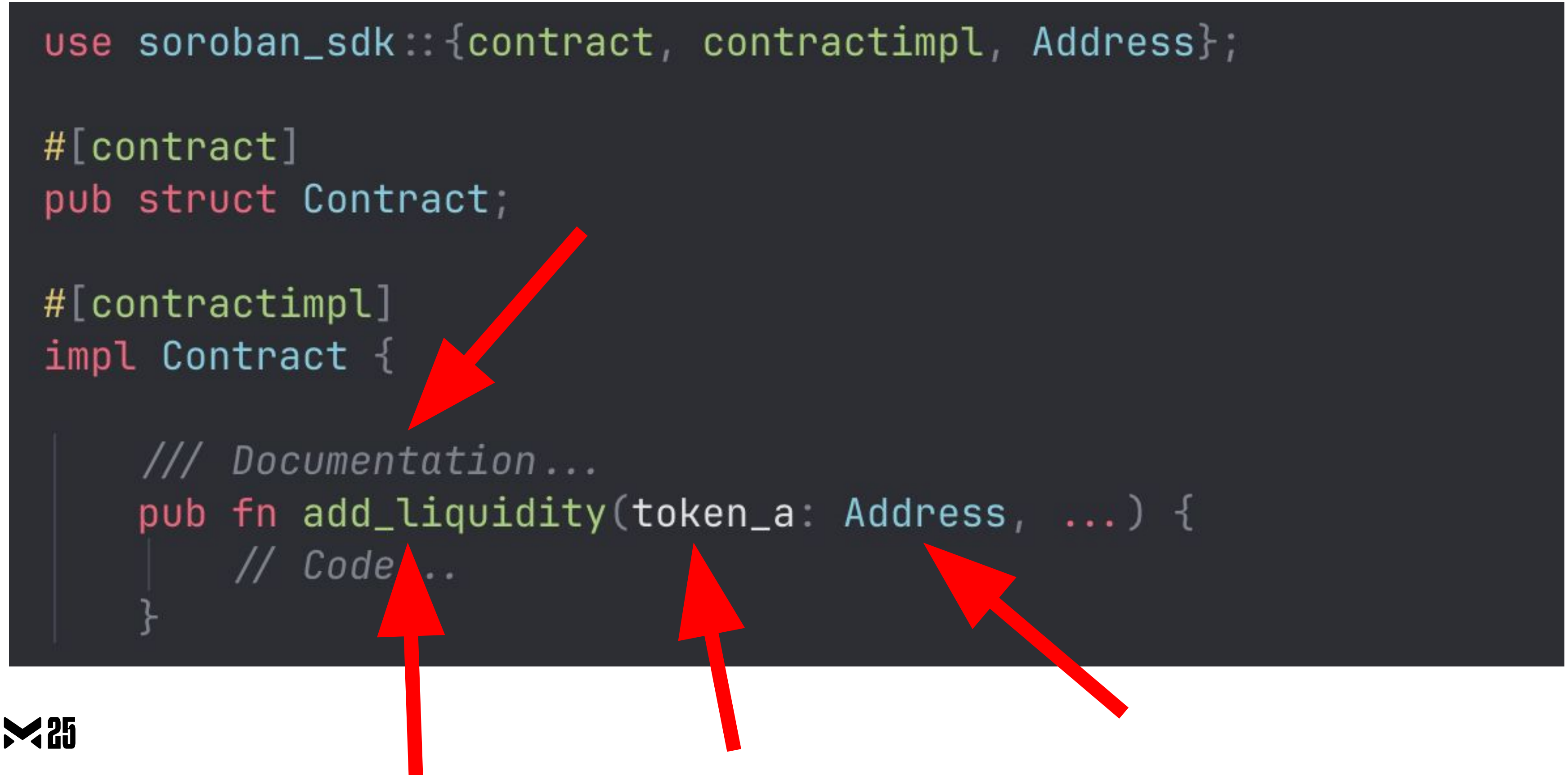
```rust
/// Adds liquidity to a token pair's pool, creating it if it doesn't exist.
/// Ensures that exactly the desired amounts of both tokens are added, subject
/// to minimum requirements.
/// This function is responsible for transferring tokens from the user to the
/// pool and minting liquidity tokens in return.
/// # Arguments
/// * `token_a` - The address of the first token to add liquidity for.
/// * `token_b` - The address of the second token to add liquidity for.
/// * `amount_a_desired` - The desired amount of the first token to add.
/// * `amount_b_desired` - The desired amount of the second token to add.
/// * `amount_a_min` - The minimum required amount of the first token to add.
/// * `amount_b_min` - The minimum required amount of the second token to add.
/// * `to` - The address where the liquidity tokens will be minted and sent.
/// * `deadline` - The deadline for executing the operation.
/// # Returns
/// A tuple containing: amounts of token A and B added to the pool.
/// plus the amount of liquidity tokens minted.
fn add_liquidity(token_a: Address, token_b: Address, amount_a_desired: i128, amount_b_desired:
i128, amount_a_min: i128, amount_b_min: i128, to: Address, deadline: u64) ->
Result<(i128,i128,i128), CombinedRouterError>
```

100%

of contracts on mainnet contain contract specs

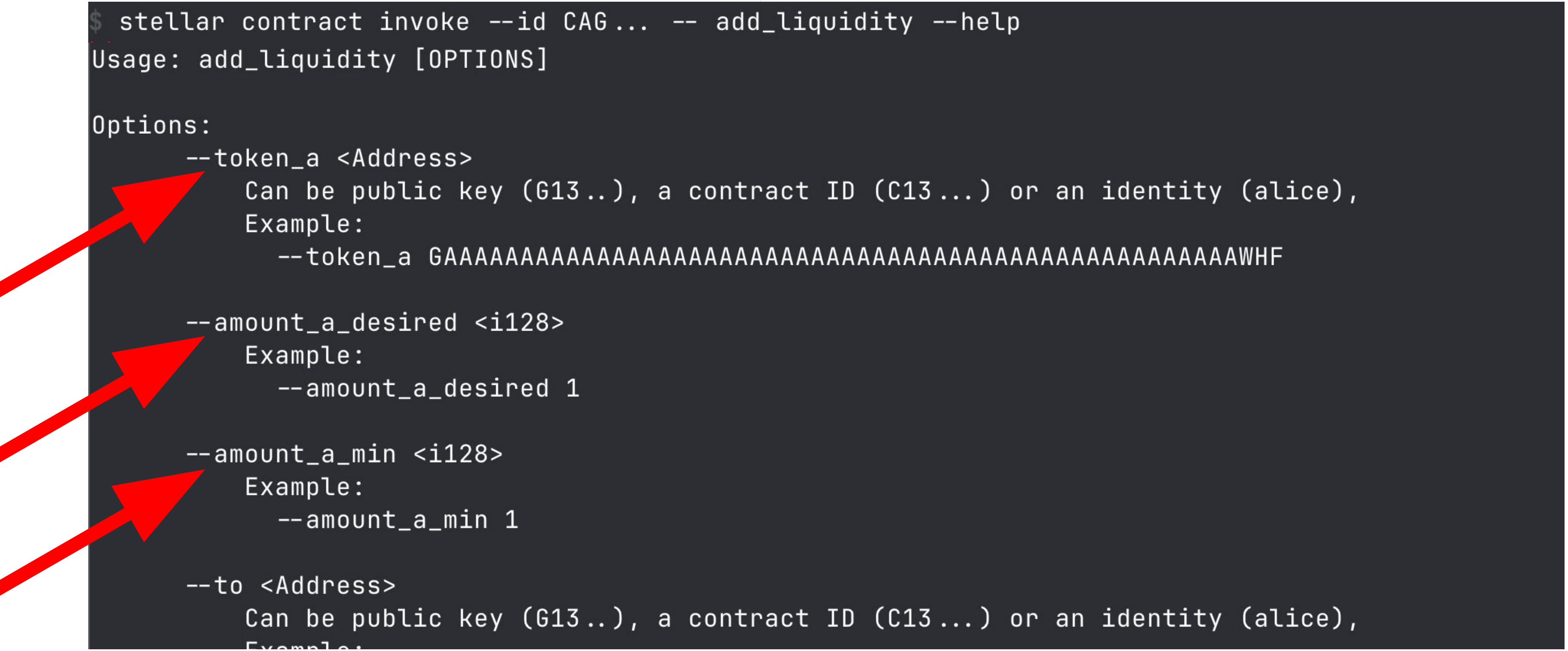
* excluding empty contracts

rust soroban-sdk



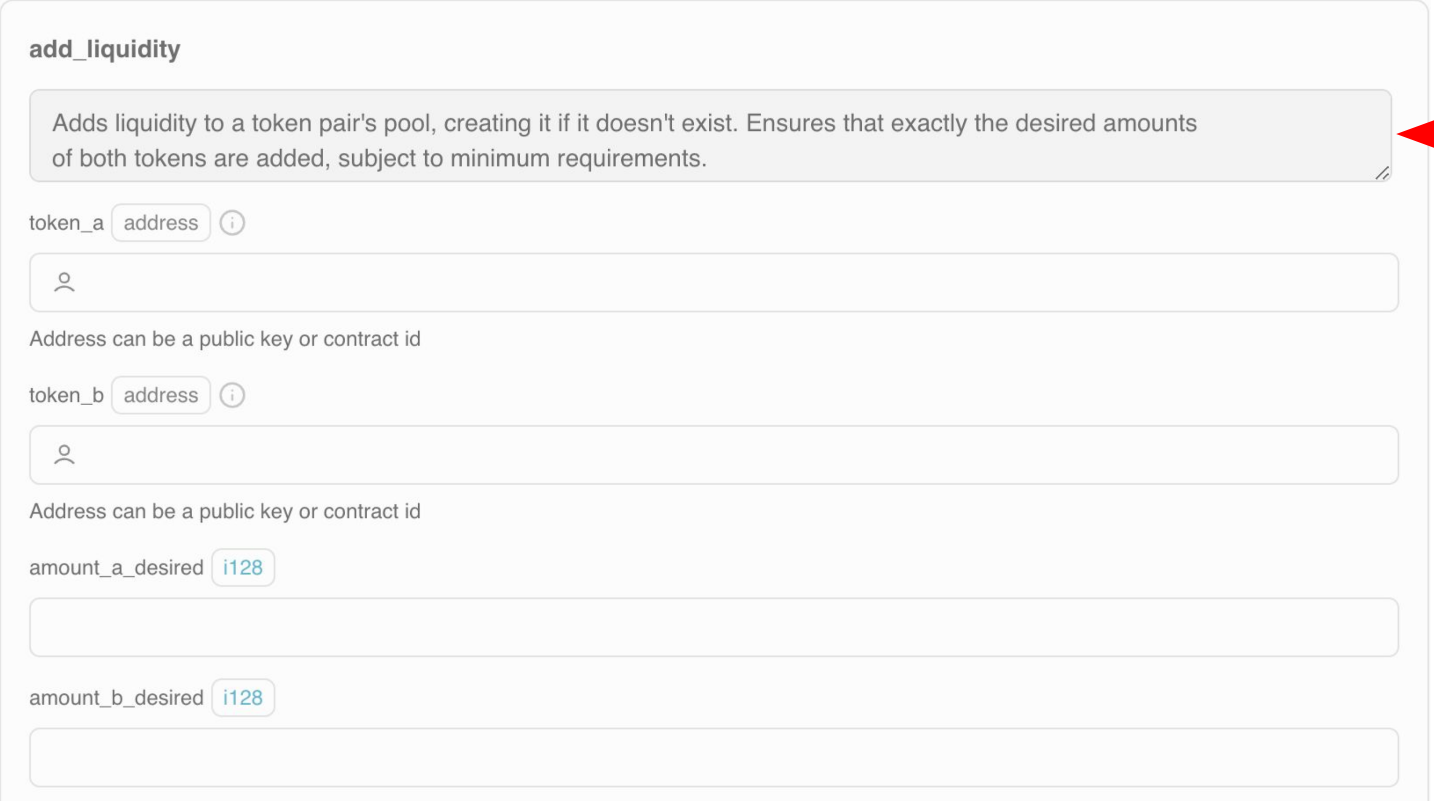
```rust
use soroban_sdk::{contract, contractimpl, Address};

#[contract]
pub struct Contract;

#[contractimpl]
impl Contract {

    /// Documentation...
    pub fn add_liquidity(token_a: Address, ...) {
        // Code...
    }
```

stellar-cli



```
$ stellar contract invoke --id CAG... -- add_liquidity --help
Usage: add_liquidity [OPTIONS]

Options:
      --token_a <Address>
          Can be public key (G13..), a contract ID (C13...) or an identity (alice),
          Example:
            --token_a GAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAWHF

      --amount_a_desired <i128>
          Example:
            --amount_a_desired 1

      --amount_a_min <i128>
          Example:
            --amount_a_min 1

      --to <Address>
          Can be public key (G13..), a contract ID (C13...) or an identity (alice),
          Example:
```

lab.stellar.org



add_liquidity

Adds liquidity to a token pair's pool, creating it if it doesn't exist. Ensures that exactly the desired amounts of both tokens are added, subject to minimum requirements.

token_a address

Address can be a public key or contract id

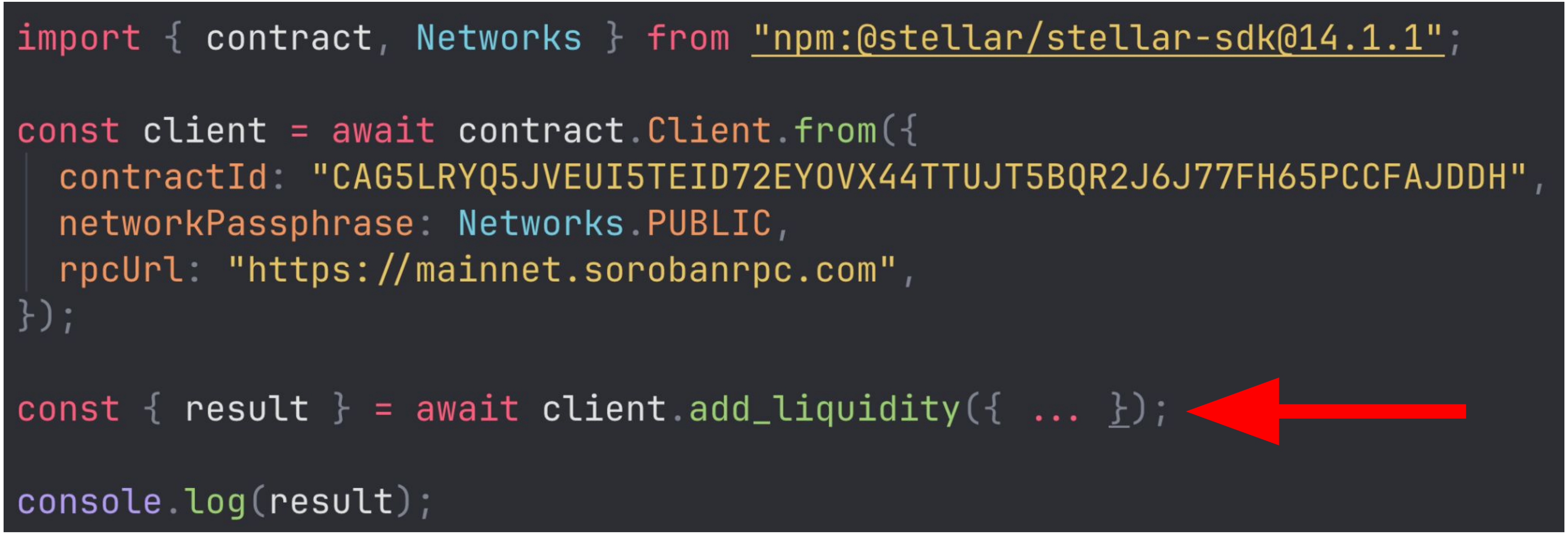
token_b address

Address can be a public key or contract id

amount_a_desired i128

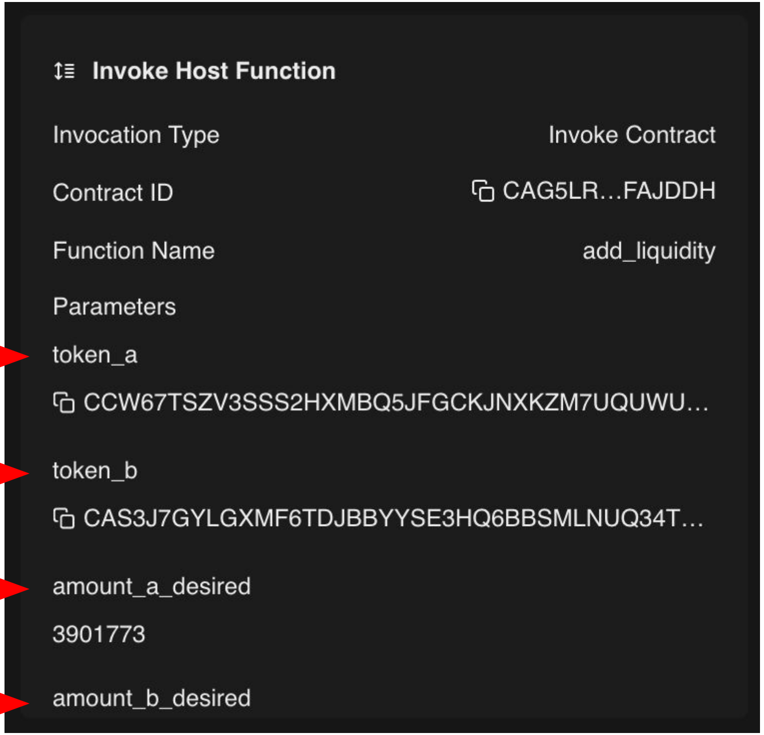
amount_b_desired i128

js stellar-sdk



```
import { contract, Networks } from "npm:@stellar/stellar-sdk@14.1.1";

const client = await contract.Client.from({
  contractId: "CAG5LRYQ5JVEUI5TEID72EYOVX44TTUJT5BQR2J6J77FH65PCCFAJDDH",
  networkPassphrase: Networks.PUBLIC,
  rpcUrl: "https://mainnet.sorobanrpc.com",
});

const { result } = await client.add_liquidity({ ... });

console.log(result);
```

freighter



Invoke Host Function

Invocation Type Invoke Contract

Contract ID CAG5LR…FAJDDH

Function Name add_liquidity

Parameters

token_a

CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWU…

token_b

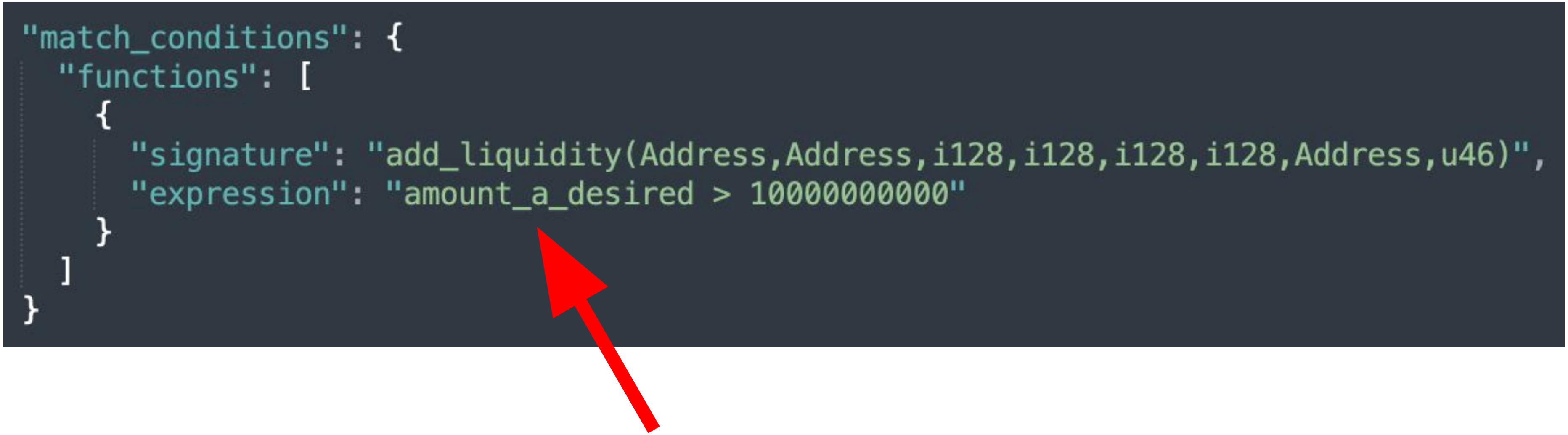
CAS3J7GYLGXMF6TDJBBYYSE3HQ6BBSMLNUQ34T…

amount_a_desired

3901773

amount_b_desired

open zeppelin monitor



```
"match_conditions": {
  "functions": [
    {
      "signature": "add_liquidity(Address,Address,i128,i128,i128,i128,Address,u46)",
      "expression": "amount_a_desired > 10000000000"
    }
  ]
}
```

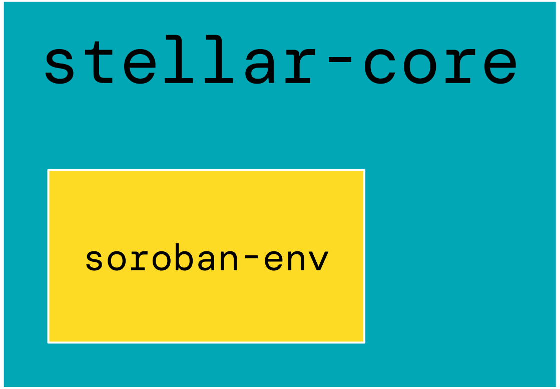
contract informed uxs

RUST	`soroban_spec::read::from_wasm(wasm)`
JS	`contract.Spec.fromWasm(wasm)`
PYTHON	`utils.get_specs_by_wasm_bytes(wasm)`
OR	wasm parser and xdr library and sep-48

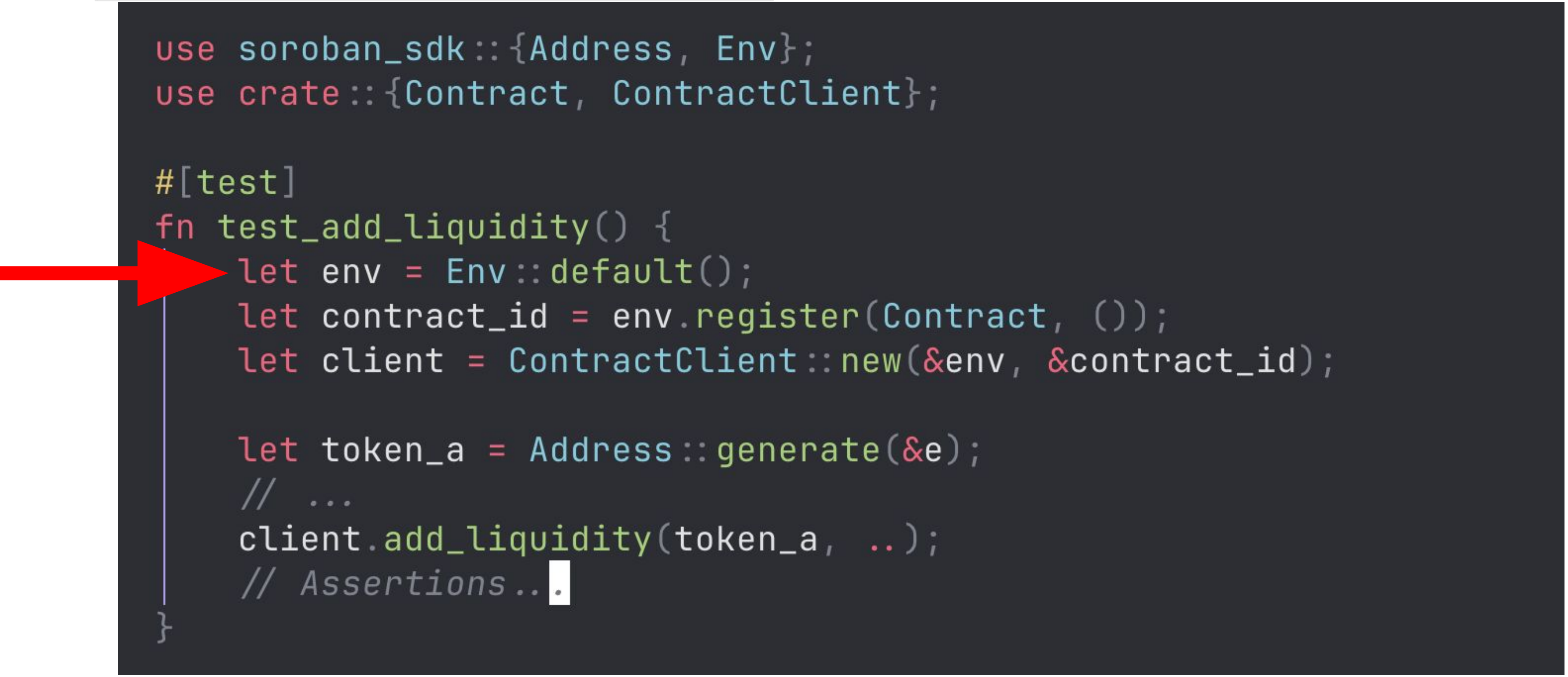
02 soroban env

github.com/stellar/rs-soroban-env

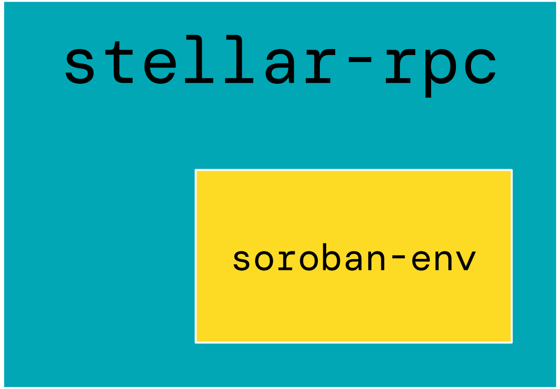
execution



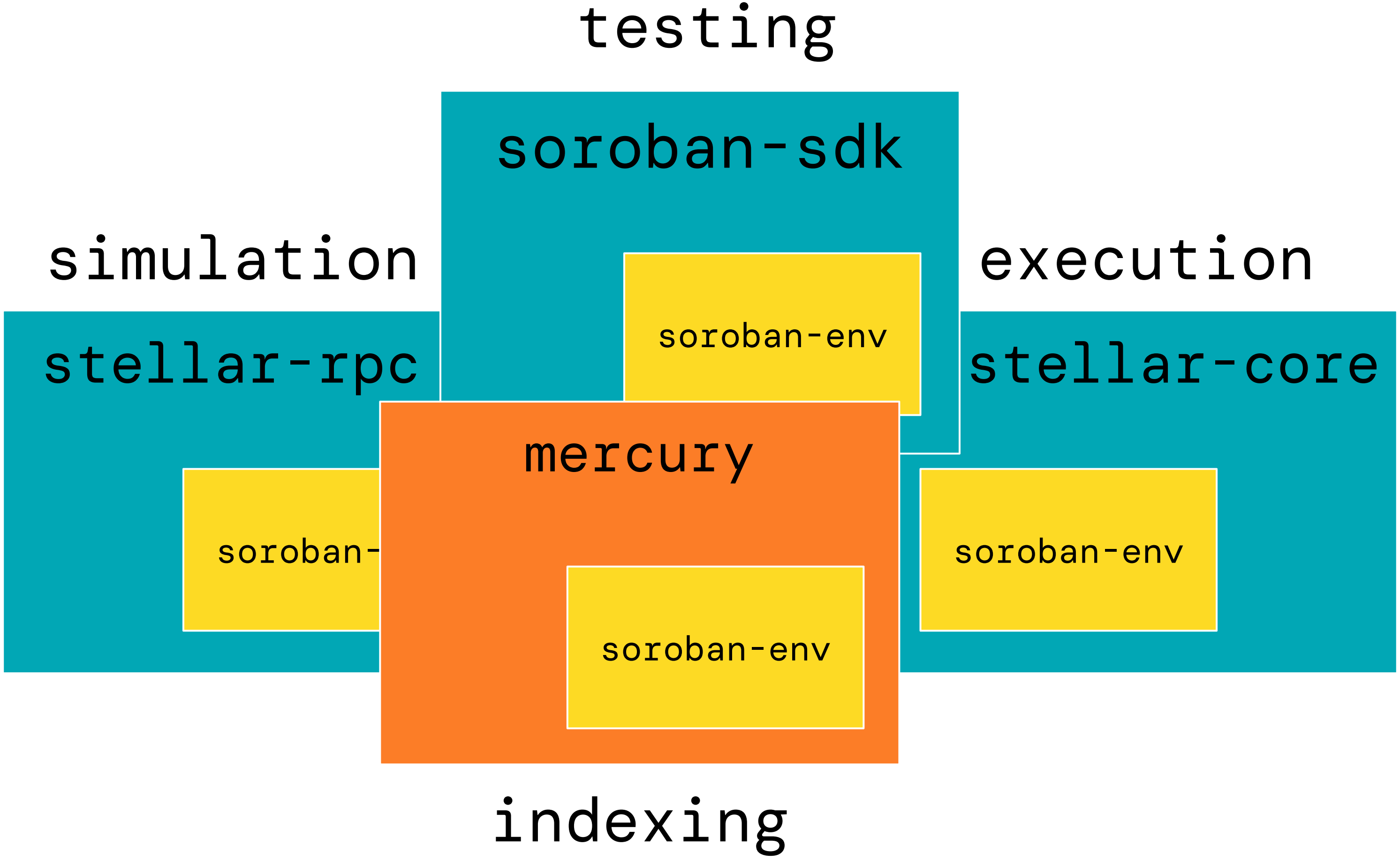
rust soroban-sdk



```rust
use soroban_sdk::{Address, Env};
use crate::{Contract, ContractClient};

#[test]
fn test_add_liquidity() {
    let env = Env::default();
    let contract_id = env.register(Contract, ());
    let client = ContractClient::new(&env, &contract_id);

    let token_a = Address::generate(&e);
    // ...
    client.add_liquidity(token_a, ..);
    // Assertions...
}
```



simulation without a simulator



testing

soroban-sdk

soroban-env

simulation

stellar-rpc

soroban-

execution

stellar-core

soroban-env

mercury

soroban-env

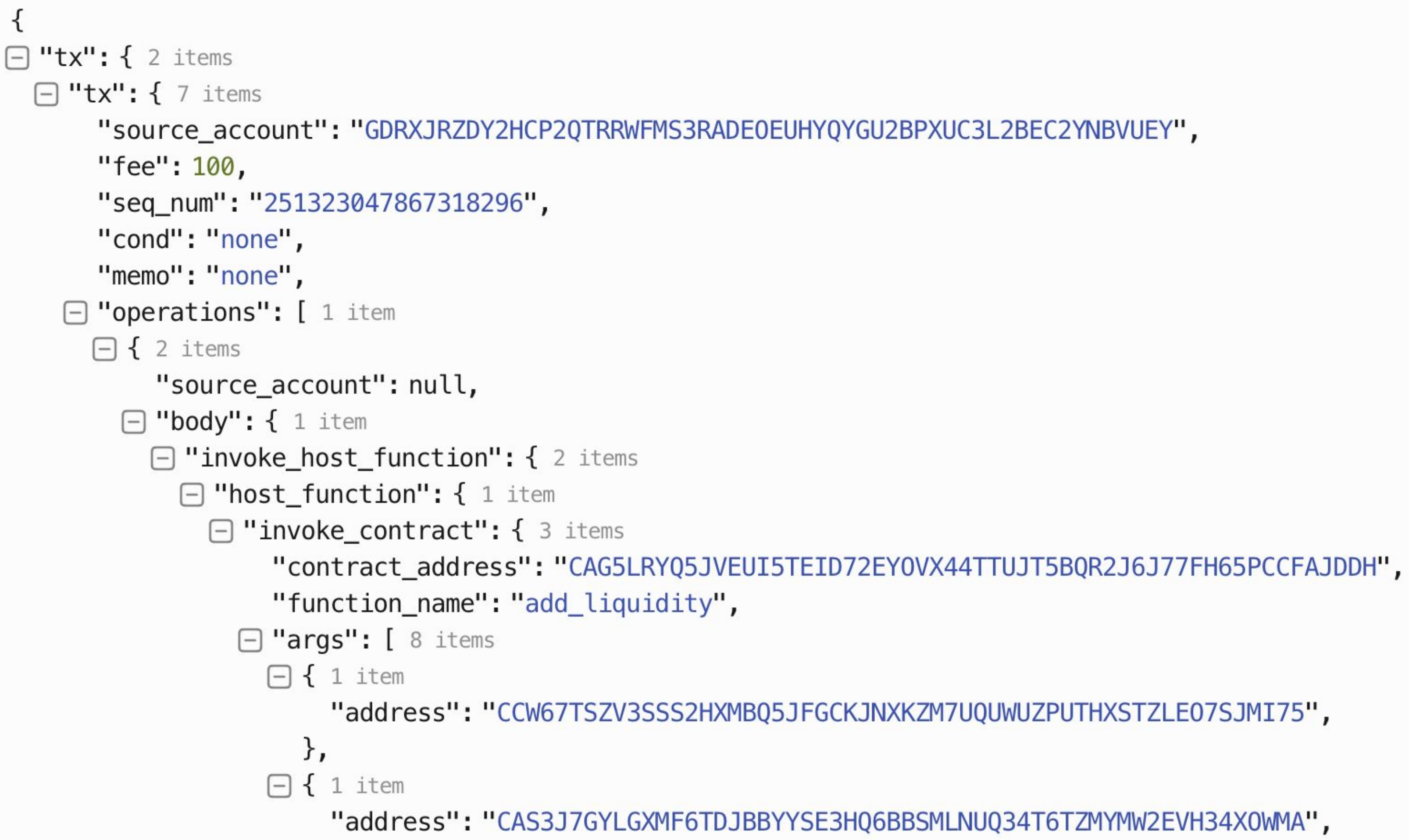
indexing

embed the soroban env

RUST soroban-env-host and soroban-simulation

03 xdr-json

stellar.org/protocol/sep-51



```
{
⊟ "tx": { 2 items
  ⊟ "tx": { 7 items
      "source_account": "GDRXJRZDY2HCP2QTRRWFMS3RADEOEUHYQYGU2BPXUC3L2BEC2YNBVUEY",
      "fee": 100,
      "seq_num": "251323047867318296",
      "cond": "none",
      "memo": "none",
    ⊟ "operations": [ 1 item
      ⊟ { 2 items
          "source_account": null,
        ⊟ "body": { 1 item
          ⊟ "invoke_host_function": { 2 items
            ⊟ "host_function": { 1 item
              ⊟ "invoke_contract": { 3 items
                  "contract_address": "CAG5LRYQ5JVEUI5TEID72EYOVX44TTUJT5BQR2J6J77FH65PCCFAJDDH",
                  "function_name": "add_liquidity",
                ⊟ "args": [ 8 items
                  ⊟ { 1 item
                      "address": "CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWUZPUTHXSTZLEO7SJMI75",
                    },
                  ⊟ { 1 item
                      "address": "CAS3J7GYLGXMF6TDJBBYYSE3HQ6BBSMLNUQ34T6TZMYMW2EVH34XOWMA",
```

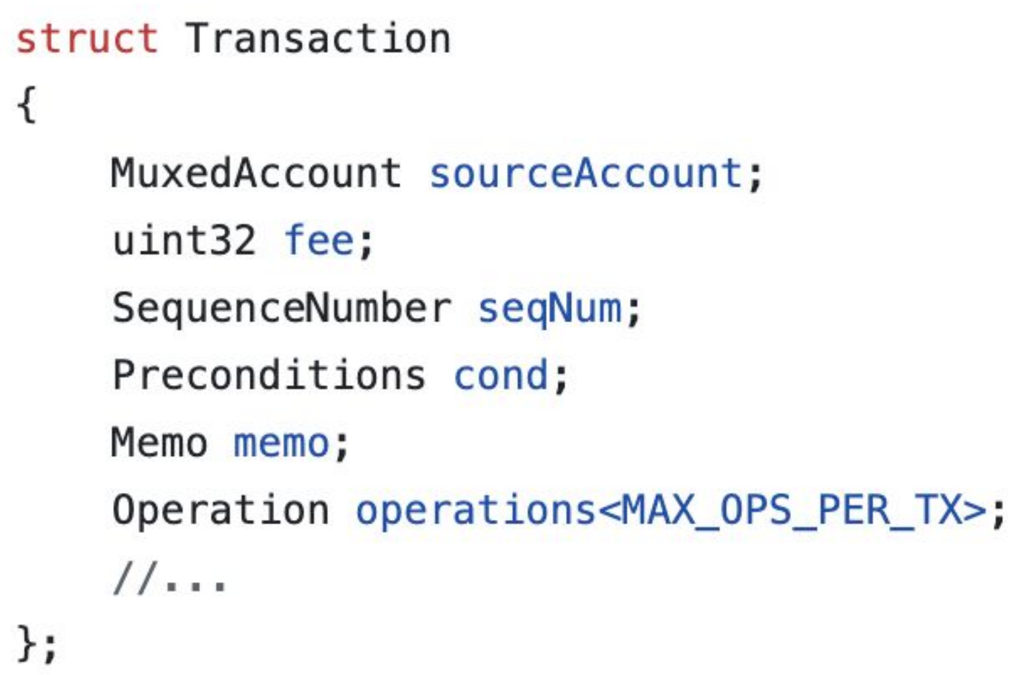
xdr

binary format

rfc4506

used on stellar

github.com/stellar/stellar-xdr



```
struct Transaction
{
    MuxedAccount sourceAccount;
    uint32 fee;
    SequenceNumber seqNum;
    Preconditions cond;
    Memo memo;
    Operation operations<MAX_OPS_PER_TX>;
    //...
};
```

	xdr
efficient	✅
simple	✅
no forward compatibility	✅
deterministic encoding	✅
readable by devs	❌

stellar-cli

```
$ stellar contract invoke --id CAG... --build-only -- add_liquidity …

AAAAAgAAAADjdMcjxo4n6hOMbFZLcQDI4lD4hg1NBfegtr0EgtYaGgAAAGQDfOD7AAA
AGAAAAAAAAAAAAAAAAQAAAAAAAAAAYAAAAAAAAAAAEN1ccQ6mpKI7MiB/0TDq35yc6Jn0
MI6T5P/lP7rxCKBAAAAA1hZGRfbGlxdWlkaXR5AAAAAAAAACAAAABIAAAABre/OWa7lK
Wj3YGHUlMJSW3Vln6QpamX0me8p5WR35JYAAAASAAAAASW0/NhZrsL6Y0hDjEibPDwQ
yYttIb5P08swy2iVPvl3AAAACgAAAAAAAAAAAAAAADuaygAAAAAKAAAAAAAAAAAAAAA
ApmaEjAAAAAoAAAAAAAAAAAAAAAA7msoAAAAACgAAAAAAAAAAAAAAAKZmhIwAAAASAA
AAAAAAAABZAD9wDhpaSCFAaXXAtE8a3GrSkiYGtjJw6mlMfRsCRAAAAAUAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA==
```

stellar-cli

```
$ stellar contract invoke --id CAG... --build-only -- add_liquidity …
    | base64 -d
    | xxd

00000000: 0000 0002 0000 0000 e374 c723 c68e 27ea  .........t.#..'.
00000010: 138c 6c56 4b71 00c8 e250 f886 0d4d 05f7  ..lVKq...P...M..
00000020: a0b6 bd04 82d6 1a1a 0000 0064 037c e0fb  ...........d.|..
00000030: 0000 0018 0000 0000 0000 0000 0000 0001  ................
00000040: 0000 0000 0000 0018 0000 0000 0000 0001  ................
00000050: 0dd5 c710 ea6a 4a23 b322 07fd 130e adf9  .....jJ#."......
00000060: c9ce 899f 4308 e93e 4ffe 53fb af10 8a04  ....C..>O.S.....
00000070: 0000 000d 6164 645f 6c69 7175 6964 6974  ....add_liquidit
00000080: 7900 0000 0000 0008 0000 0012 0000 0001  y...............
00000090: adef ce59 aee5 2968 f760 61d4 94c2 525b  ...Y..)h.`a...R[
000000a0: 7565 9fa4 296a 65f4 99ef 29e5 6477 e496  ue..)je...).dw..
000000b0: 0000 0012 0000 0001 25b4 fcd8 59ae c2fa  ........%...Y...
000000c0: 6348 438c 489b 3c3c 10c9 8b6d 21be 4fd3  cHC.H.<<...m!.O.
000000d0: cb30 cb68 953e f977 0000 000a 0000 0000  .0.h.>.w........
000000e0: 0000 0000 0000 0000 3b9a ca00 0000 000a  ........;.......
000000f0: 0000 0000 0000 0000 0000 0000 a666 848c  .............f..
00000100: 0000 000a 0000 0000 0000 0000 0000 0000  ................
```

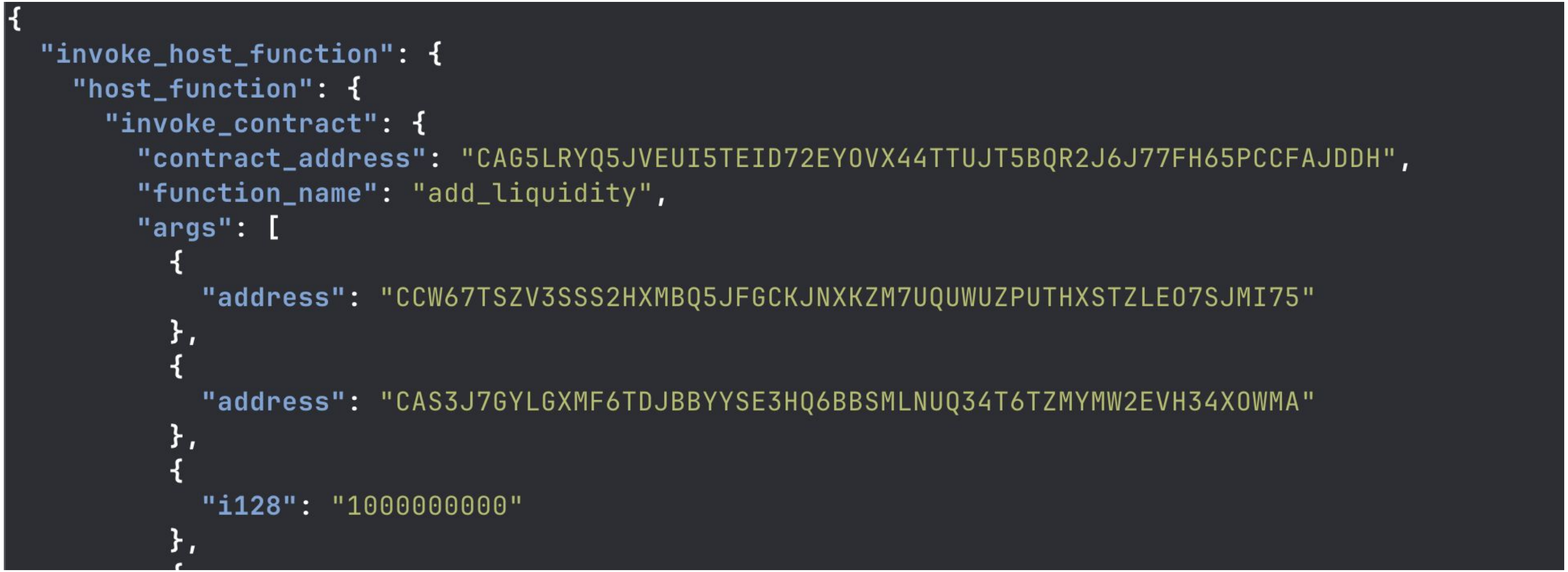
xdr-json

xdr → json

`xdr` ↔ `json`

stellar-cli

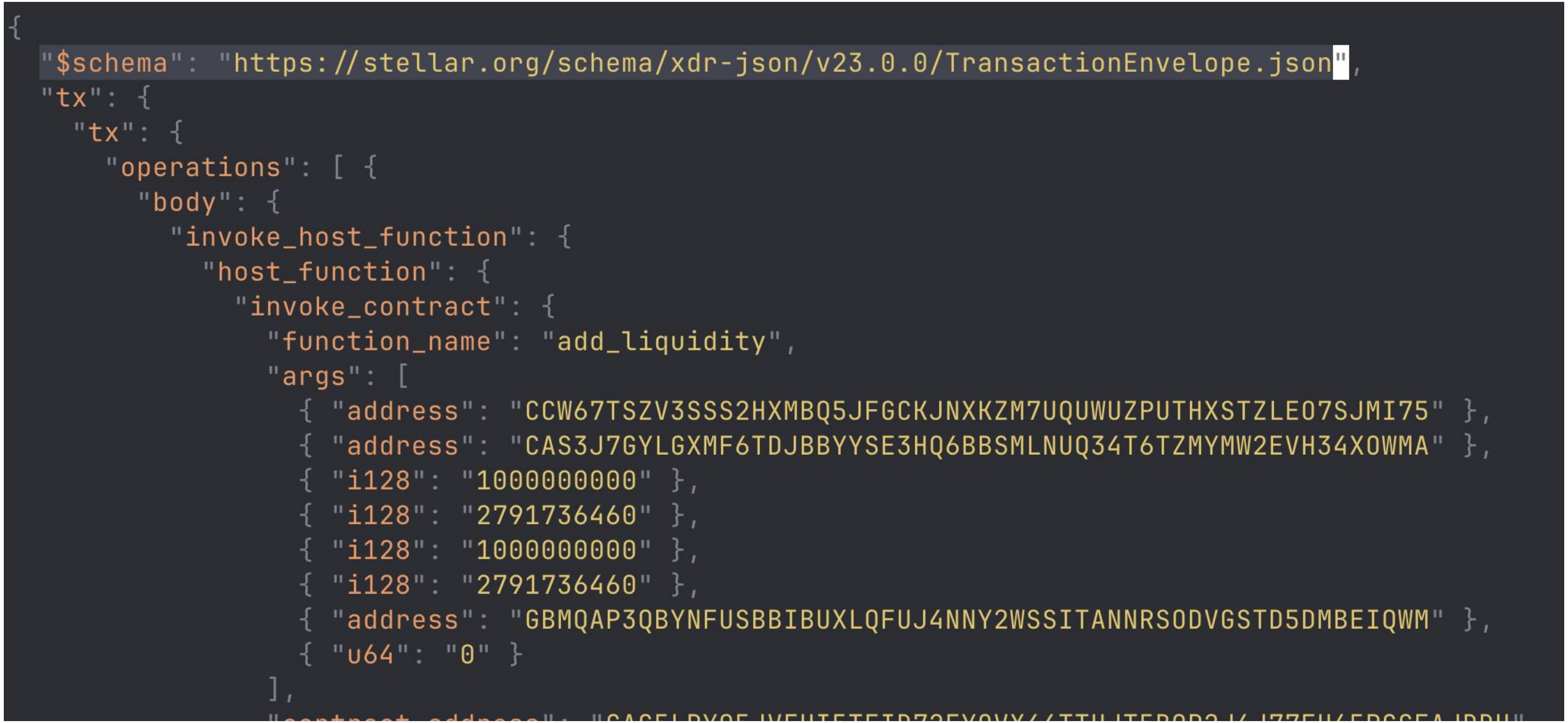
```
$ stellar contract invoke --id CAG... --build-only -- add_liquidity …
      | stellar tx decode
      | jq '.tx.tx.operations[0].body'
```



```
{
  "invoke_host_function": {
    "host_function": {
      "invoke_contract": {
        "contract_address": "CAG5LRYQ5JVEUI5TEID72EYOVX44TTUJT5BQR2J6J77FH65PCCFAJDDH",
        "function_name": "add_liquidity",
        "args": [
          {
            "address": "CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWUZPUTHXSTZLEO7SJMI75"
          },
          {
            "address": "CAS3J7GYLGXMF6TDJBBYYSE3HQ6BBSMLNUQ34T6TZMYMW2EVH34XOWMA"
          },
          {
            "i128": "1000000000"
          },
          {
```

stellar-cli

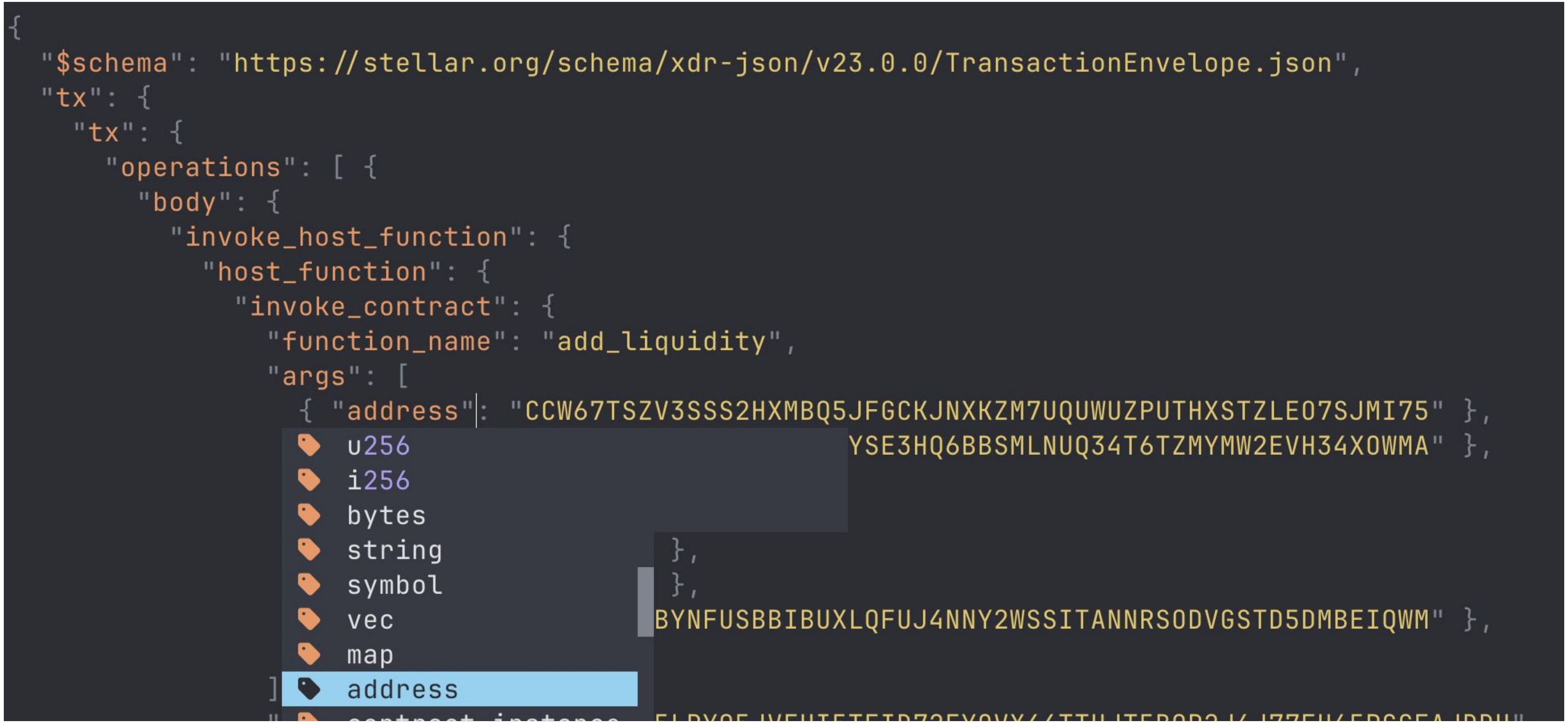
```
$ stellar contract invoke --id CAG... --build-only -- add_liquidity ... | stellar tx edit
```



```
{
  "$schema": "https://stellar.org/schema/xdr-json/v23.0.0/TransactionEnvelope.json",
  "tx": {
    "tx": {
      "operations": [ {
        "body": {
          "invoke_host_function": {
            "host_function": {
              "invoke_contract": {
                "function_name": "add_liquidity",
                "args": [
                  { "address": "CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWUZPUTHXSTZLEO7SJMI75" },
                  { "address": "CAS3J7GYLGXMF6TDJBBYYSE3HQ6BBSMLNUQ34T6TZMYMW2EVH34XOWMA" },
                  { "i128": "1000000000" },
                  { "i128": "2791736460" },
                  { "i128": "1000000000" },
                  { "i128": "2791736460" },
                  { "address": "GBMQAP3QBYNFUSBBIBUXLQFUJ4NNY2WSSITANNRSODVGSTD5DMBEIQWM" },
                  { "u64": "0" }
                ],
```

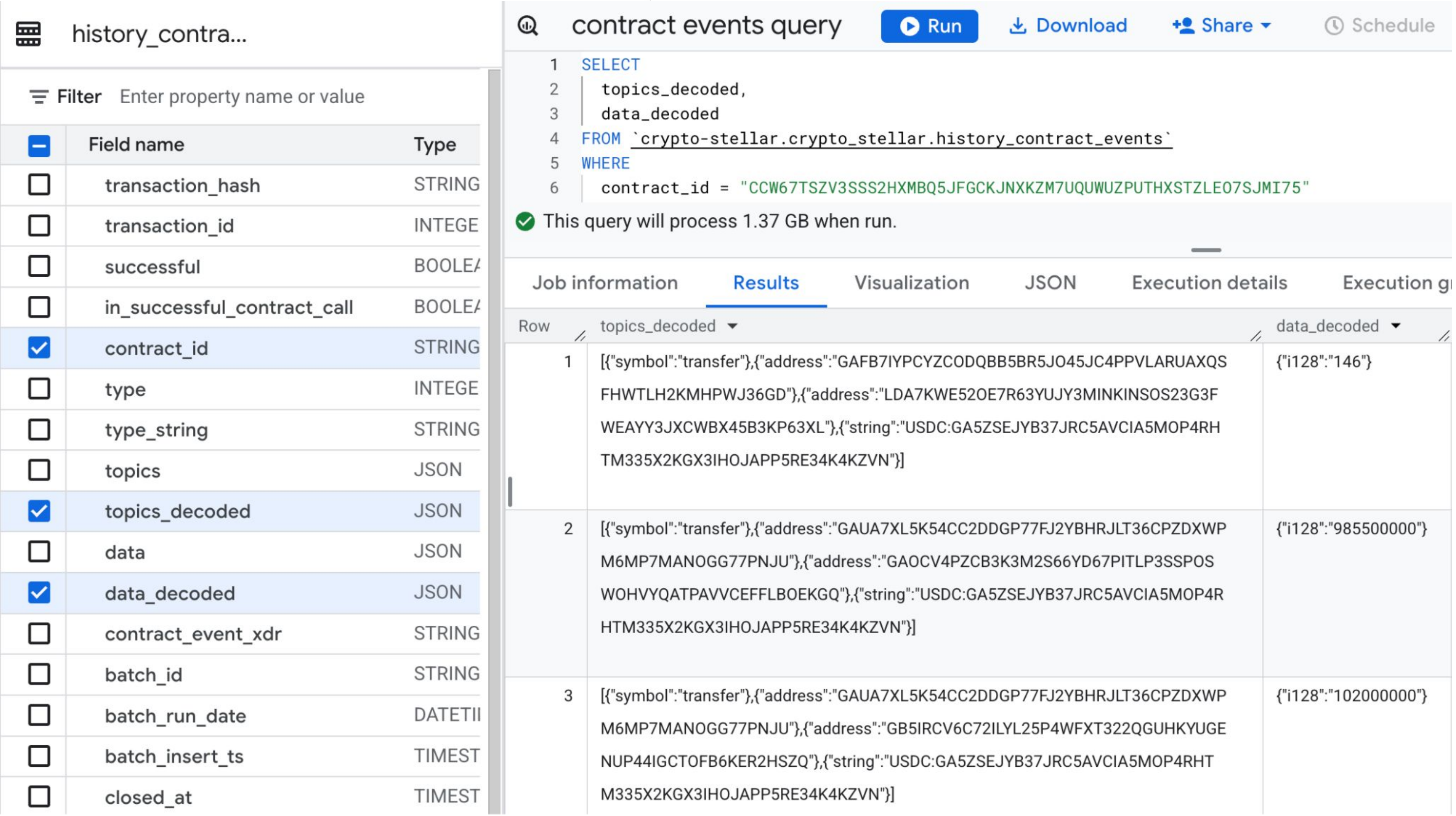
stellar-cli

```
$ stellar contract invoke --id CAG... --build-only -- add_liquidity ... | stellar tx edit
```



```
{
  "$schema": "https://stellar.org/schema/xdr-json/v23.0.0/TransactionEnvelope.json",
  "tx": {
    "tx": {
      "operations": [ {
        "body": {
          "invoke_host_function": {
            "host_function": {
              "invoke_contract": {
                "function_name": "add_liquidity",
                "args": [
                  { "address": "CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWUZPUTHXSTZLEO7SJMI75" },
                  u256                      YSE3HQ6BBSMLNUQ34T6TZMYMW2EVH34XOWMA" },
                  i256
                  bytes
                  string           },
                  symbol           },
                  vec              BYNFUSBBIBUXLQFUJ4NNY2WSSITANNRSODVGSTD5DMBEIQWM" },
                  map
                ] address
```

hubble (bigquery)



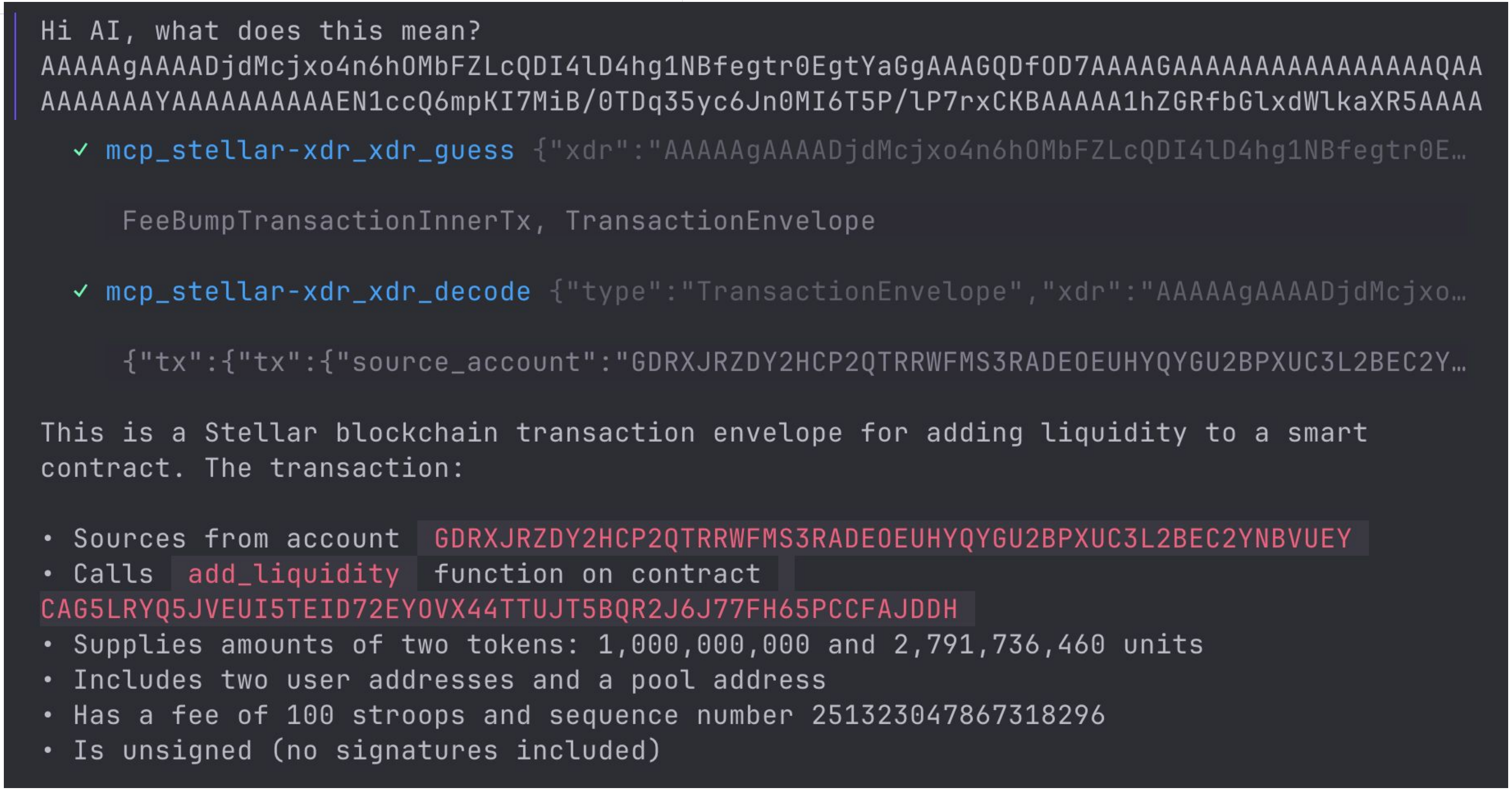
☐	Field name	Type
☐	transaction_hash	STRING
☐	transaction_id	INTEGE
☐	successful	BOOLEA
☐	in_successful_contract_call	BOOLEA
☑	contract_id	STRING
☐	type	INTEGE
☐	type_string	STRING
☐	topics	JSON
☑	topics_decoded	JSON
☐	data	JSON
☑	data_decoded	JSON
☐	contract_event_xdr	STRING
☐	batch_id	STRING
☐	batch_run_date	DATETII
☐	batch_insert_ts	TIMEST
☐	closed_at	TIMEST

```sql
SELECT
  topics_decoded,
  data_decoded
FROM `crypto-stellar.crypto_stellar.history_contract_events`
WHERE
  contract_id = "CCW67TSZV3SSS2HXMBQ5JFGCKJNXKZM7UQUWUZPUTHXSTZLEO7SJMI75"
```

This query will process 1.37 GB when run.

Row	topics_decoded	data_decoded
1	[{"symbol":"transfer"},{"address":"GAFB7IYPCYZCODQBB5BR5JO45JC4PPVLARUAXQSFHWTLH2KMHPWJ36GD"},{"address":"LDA7KWE52OE7R63YUJY3MINKINSOS23G3FWEAYY3JXCWBX45B3KP63XL"},{"string":"USDC:GA5ZSEJYB37JRC5AVCIA5MOP4RHTM335X2KGX3IHOJAPP5RE34K4KZVN"}]	{"i128":"146"}
2	[{"symbol":"transfer"},{"address":"GAUA7XL5K54CC2DDGP77FJ2YBHRJLT36CPZDXWPM6MP7MANOGG77PNJU"},{"address":"GAOCV4PZCB3K3M2S66YD67PITLP3SSPOSWOHVYQATPAVVCEFFLBOEKGQ"},{"string":"USDC:GA5ZSEJYB37JRC5AVCIA5MOP4RHTM335X2KGX3IHOJAPP5RE34K4KZVN"}]	{"i128":"985500000"}
3	[{"symbol":"transfer"},{"address":"GAUA7XL5K54CC2DDGP77FJ2YBHRJLT36CPZDXWPM6MP7MANOGG77PNJU"},{"address":"GB5IRCV6C72ILYL25P4WFXT322QGUHKYUGENUP44IGCTOFB6KER2HSZQ"},{"string":"USDC:GA5ZSEJYB37JRC5AVCIA5MOP4RHTM335X2KGX3IHOJAPP5RE34K4KZVN"}]	{"i128":"102000000"}

stellar-xdr mcp



Hi AI, what does this mean?

```
AAAAAgAAAADjdMcjxo4n6hOMbFZLcQDI4lD4hg1NBfegtr0EgtYaGgAAAGQDfOD7AAAAGAAAAAAAAAAAAAAAAQAA
AAAAAAAYAAAAAAAAAAEN1ccQ6mpKI7MiB/0TDq35yc6Jn0MI6T5P/lP7rxCKBAAAAA1hZGRfbGlxdWlkaXR5AAAA
```

```
✓ mcp_stellar-xdr_xdr_guess {"xdr":"AAAAAgAAAADjdMcjxo4n6hOMbFZLcQDI4lD4hg1NBfegtr0E…

    FeeBumpTransactionInnerTx, TransactionEnvelope

✓ mcp_stellar-xdr_xdr_decode {"type":"TransactionEnvelope","xdr":"AAAAAgAAAADjdMcjxo…

    {"tx":{"tx":{"source_account":"GDRXJRZDY2HCP2QTRRWFMS3RADEOEUHYQYGU2BPXUC3L2BEC2Y…
```

This is a Stellar blockchain transaction envelope for adding liquidity to a smart contract. The transaction:

- Sources from account `GDRXJRZDY2HCP2QTRRWFMS3RADEOEUHYQYGU2BPXUC3L2BEC2YNBVUEY`
- Calls `add_liquidity` function on contract `CAG5LRYQ5JVEUI5TEID72EYOVX44TTUJT5BQR2J6J77FH65PCCFAJDDH`
- Supplies amounts of two tokens: 1,000,000,000 and 2,791,736,460 units
- Includes two user addresses and a pool address
- Has a fee of 100 stroops and sequence number 251323047867318296
- Is unsigned (no signatures included)

give devs json

RUST stellar-xdr

JS @stellar/stellar-xdr-json

GO github.com/stellar/go-stellar-xdr-json

01

contract specs

02

soroban env

03

xdr json

leighmcculloch/m25