Moonbeam

Transact Remotely Through XCM

Presented by
Alberto Viera
Developer Relations - Moonbeam

Agenda:

XCM Overview

Instructions, Execution and Fees

Agenda:

XCM Overview

Instructions, Execution and Fees

XCM Tools

Decoding XCM Messages

Agenda:

XCM Overview

Instructions, Execution and Fees

XCM Tools

Monitors & *xcm-tools* repo

Remote Transact

Transact Remotely Through XCM

Agenda:

XCM Overview

Instructions, Execution and Fees

XCM Tools

Monitors & *xcm-tools* repo

Remote Transact

Transact Remotely Through XCM

Demo

Remote Transact Demo

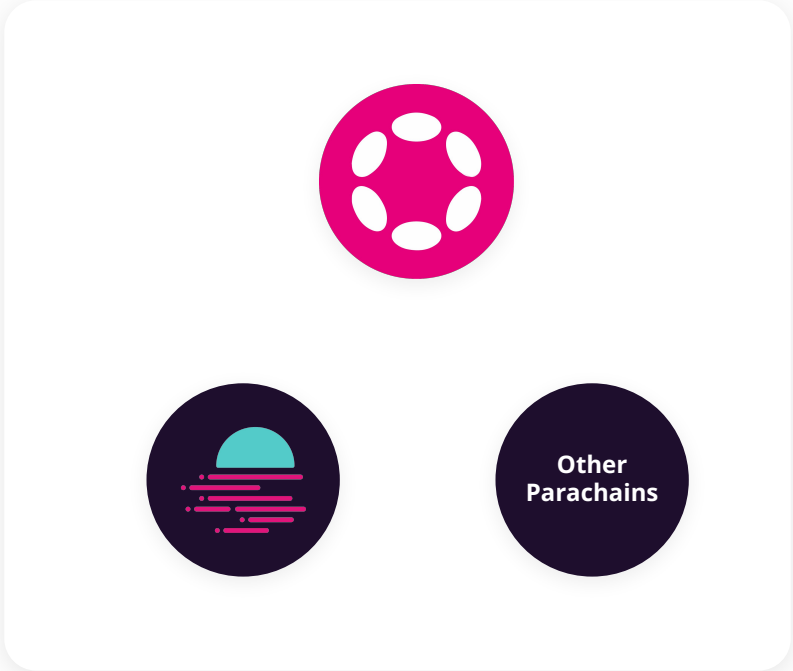

XCM Overview

https://github.com/paritytech/xcm-format
https://github.com/paritytech/polkadot/blob/master/xcm/xcm-executor/src/lib.rs

Polkadot & XCM
Native Intra-Communication Layer

XCM
Message with a given format and a set of instructions

XCVM
Virtual machine that interprets and executes the instructions in the XCM



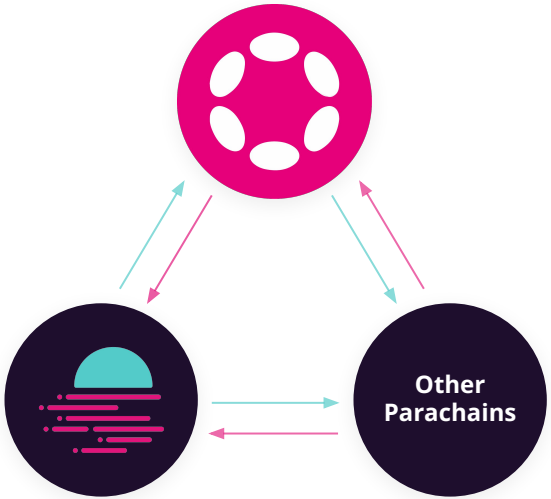
https://github.com/paritytech/xcm-format
https://github.com/paritytech/polkadot/blob/master/xcm/xcm-executor/src/lib.rs

Polkadot & XCM
Native Intra-Communication Layer

XCM
Message with a given format and a set of instructions

XCVM
Virtual machine that interprets and executes the instructions in the XCM

Communication Channels
They connect chain A with chain B (uni-directional). Allows message passing in that given direction



https://github.com/paritytech/xcm-format
https://github.com/paritytech/polkadot/blob/master/xcm/xcm-executor/src/lib.rs

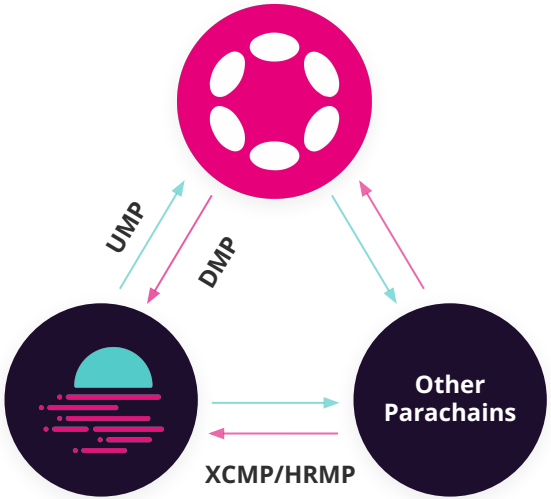
Polkadot & XCM
Native Intra-Communication Layer

XCM
Message with a given format and a set of instructions

XCVM
Virtual machine that interprets and executes the instructions in the XCM

Communication Channels
They connect chain A with chain B (uni-directional). Allows message passing in that given direction

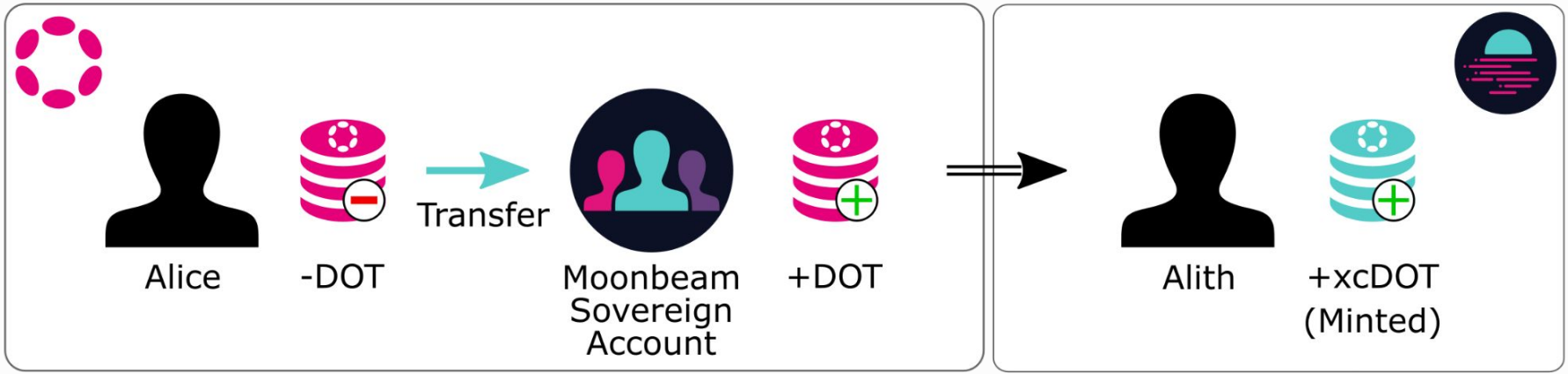
UMP/DMP & XCMP
How the message is passed around




Remote Token Transfers

Polkadot & XCM
XCM Example: Token Transfers

Asset Transfer from Polkadot (Alice) to Moonbeam (Alith)



https://www.shawntabrizi.com/substrate-js-utilities/
https://github.com/PureStake/xcm-tools/blob/main/scripts/calculate-sovereign-account.ts

Polkadot & XCM
Sovereign Account?

01 **Account Own by a Chain**
Relay Chain/Parachains own(s) an account on other chains

02 **Dispatch Origin & Holds Funds**
“Locking smart contract” on Ethereum bridges

03 **Address is Deterministic**
For Parachains, is determined by their parachain ID. For Relay Chain, it is encoding [“Parent” and trailing zeros]

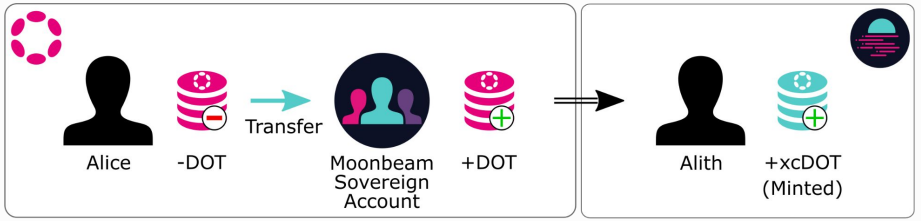
```
const sovAddressRelay = u8aToHex(`
    new Uint8Array([...new TextEncoder().encode('para'),
...targetParaId.toU8a()])
  ).padEnd(66, '0');

  const sovAddressPara = u8aToHex(
    new Uint8Array([...new TextEncoder().encode('sibl'),
...targetParaId.toU8a()])
  ).padEnd(66, '0');

  console.log(`Sovereign Account Address on Relay:
${sovAddressRelay}`);
  console.log(`Sovereign Account Address on other
Parachains (Generic): ${sovAddressPara}`);
  console.log(`Sovereign Account Address on Moonbase
Alpha: ${sovAddressPara.slice(0, 42)}`);
```

Polkadot & XCM
XCM Example: Token Transfers

TransferReserveAsset (URL)

Take funds from Alice and deposits them to Moonbeam's Sovereign Account

Sends an XCM message to destination that starts with *ReserveAssetDeposited* and *ClearOrigin*



Polkadot & XCM
XCM Example: Token Transfers

TransferReserveAsset (URL)

Take funds from Alice and deposits them to Moonbeam's Sovereign Account

Sends an XCM message to destination that starts with *ReserveAssetDeposited* and *ClearOrigin*

```
TransferReserveAsset { mut assets, dest, xcm } =>
{
  let origin =
self.origin.as_ref().ok_or(XcmError::BadOrigin)?;
  for asset in assets.inner() {
    Config::AssetTransactor::transfer_asset(asset,
origin, &dest)?;
  }
  let ancestry =
Config::LocationInverter::ancestry();
  assets.reanchor(&dest, &ancestry).map_err(|()|
XcmError::MultiLocationFull)?;
  let mut message =
vec![ReserveAssetDeposited(assets), ClearOrigin];
  message.extend(xcm.0.into_iter());
  Config::XcmSender::send_xcm(dest,
Xcm(message)).map_err(Into::into)
}
```

Polkadot & XCM
XCM Example: Token Transfers

TransferReserveAsset (URL)

Take funds from Alice and deposits them to Moonbeam's Sovereign Account

Sends an XCM message to destination that starts with *ReserveAssetDeposited* and *ClearOrigin*

```
TransferReserveAsset { mut assets, dest, xcm } =>
{
  let origin =
self.origin.as_ref().ok_or(XcmError::BadOrigin)?;
  for asset in assets.inner() {
    Config::AssetTransactor::transfer_asset(asset,
origin, &dest)?;
  }
  let ancestry =
Config::LocationInverter::ancestry();
  assets.reanchor(&dest, &ancestry).map_err(|()|
XcmError::MultiLocationFull)?;
  let mut message =
vec![ReserveAssetDeposited(assets), ClearOrigin];
  message.extend(xcm.0.into_iter());
  Config::XcmSender::send_xcm(dest,
Xcm(message)).map_err(Into::into)
}
```

Polkadot & XCM
XCM Example: Token Transfers

TransferReserveAsset (URL)

Take funds from Alice and deposits them to Moonbeam's Sovereign Account

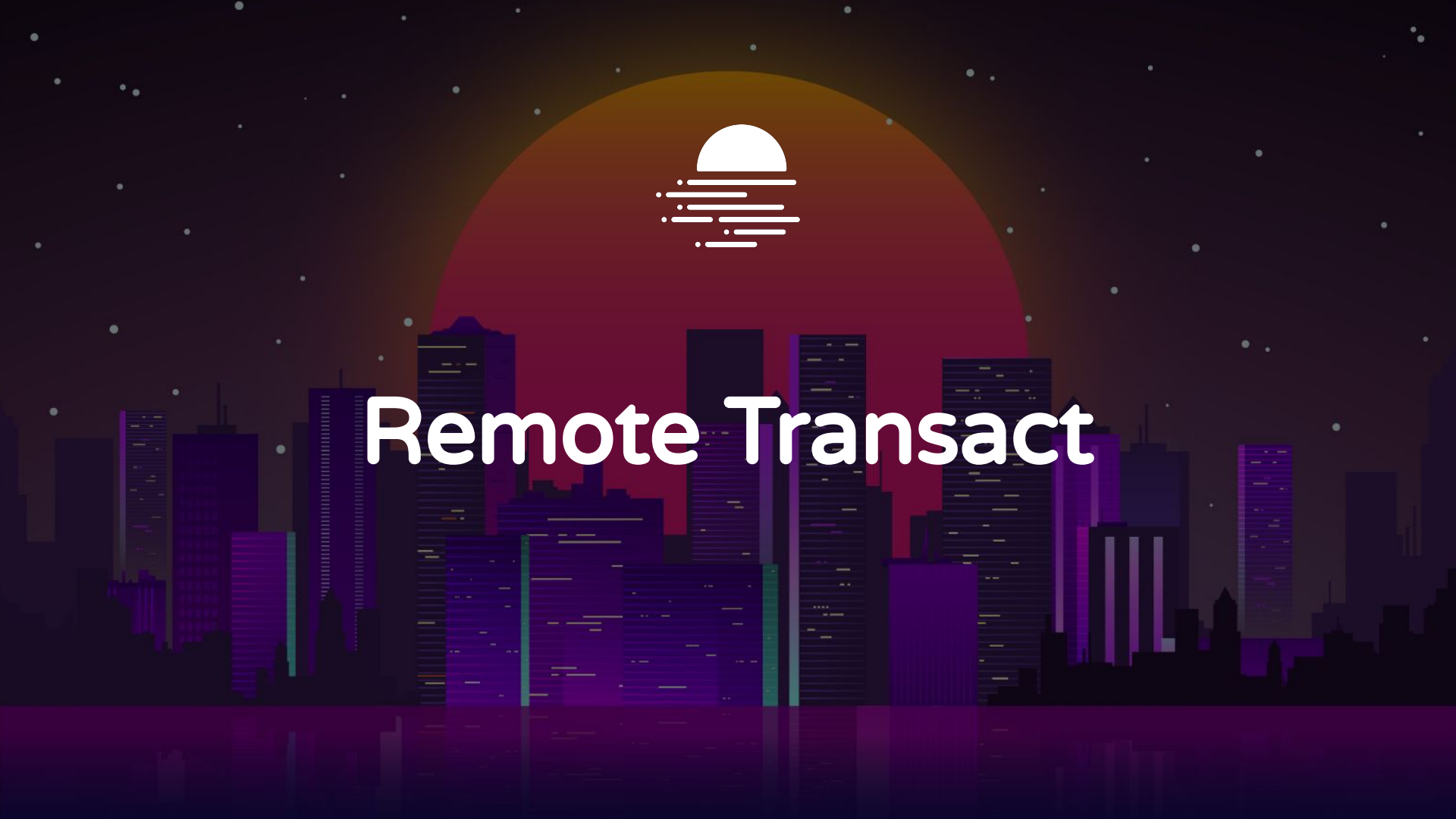
Sends an XCM message to destination that starts with *ReserveAssetDeposited* and *ClearOrigin*

ReserveAssetDeposited (URL)
Mints a representation of the asset (xcDOT) into holding

ClearOrigin (URL)
Removes the Origin register information

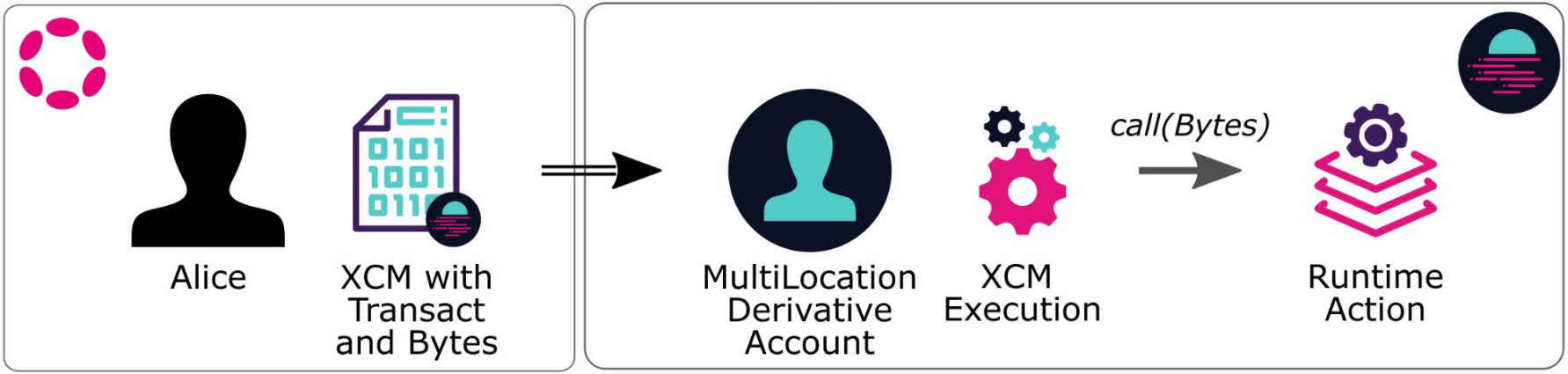
BuyExecution (URL)
Buys execution time, pays from holding

DepositAsset (URL)
Takes assets from holding and deposits to Alith


Remote Transact

Polkadot & XCM
XCM Example: Remote Transact

Remote Transact from Polkadot to Moonbeam



https://github.com/paritytech/xcm-format
https://github.com/paritytech/polkadot/blob/master/xcm/xcm-executor/src/lib.rs

Polkadot & XCM

XCM Example: Remote Transact

XCM Instruction (Optional)

You could include instructions to send assets to pay for the remote execution

https://github.com/paritytech/xcm-format
https://github.com/paritytech/polkadot/blob/master/xcm/xcm-executor/src/lib.rs

Polkadot & XCM

XCM Example: Remote Transact

XCM Instruction (Optional)

You could include instructions to send assets to pay for the remote execution

DescendOrigin (Optional) (URL)
Transforms the Origin with a given interior

WithdrawAsset (URL)
Removes assets from an account and put them in holding

BuyExecution (URL)
Buys execution time, pays from holding

Transact (URL)
Dispatches an encoded call data


Remote EVM Calls
Through XCM

https://docs.moonbeam.network/builders/xcm/remote-evm-calls/

Remote EVM Calls Through XCM

01

Ethereum-XCM Pallet

Middleware between the *Transact* XCM instruction and the Ethereum Pallet

02

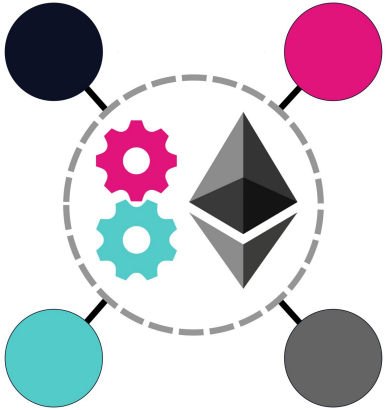
MultiLocation-Derivative Accs

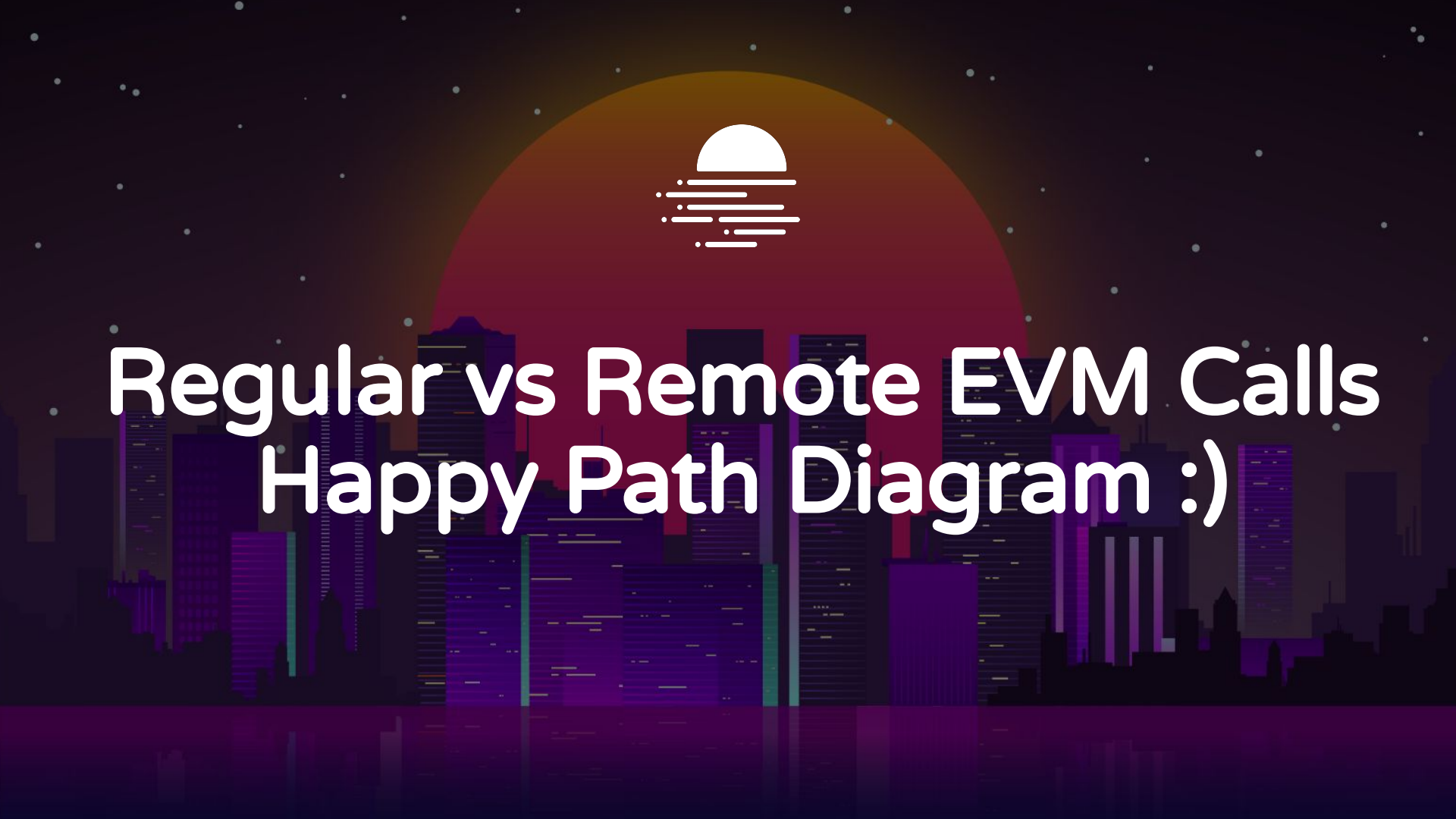
Keyless account controlled by another account from an external chain. Obtained from a *DescendOrigin* XCM instruction

03

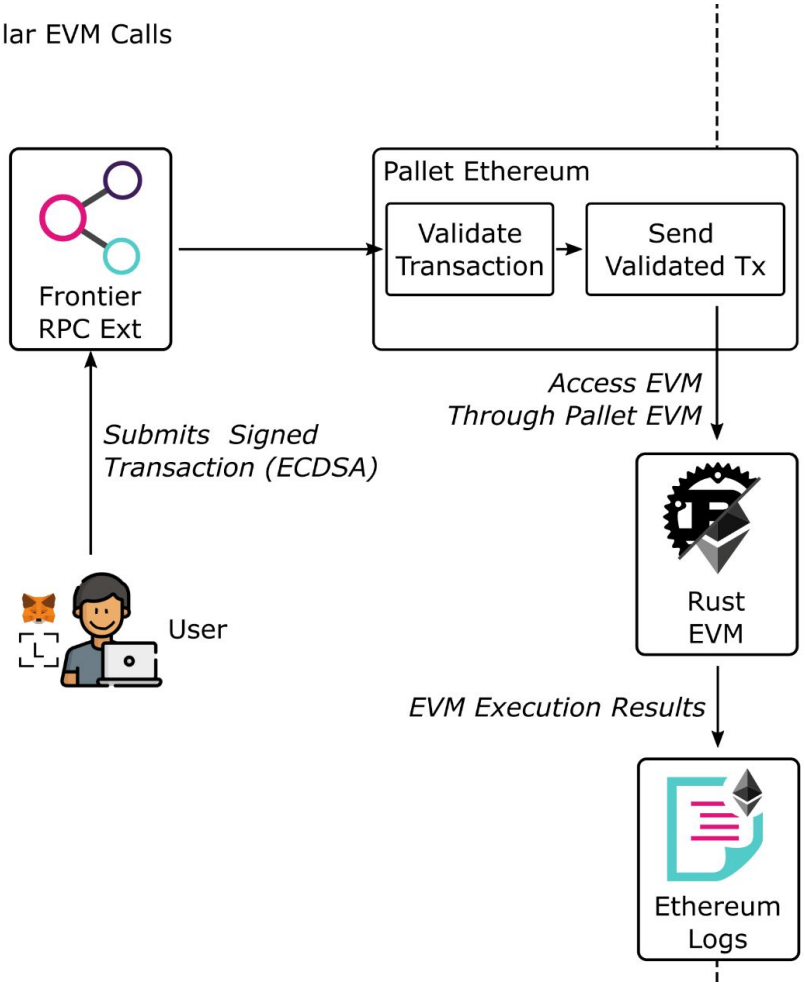
Proxied Execution

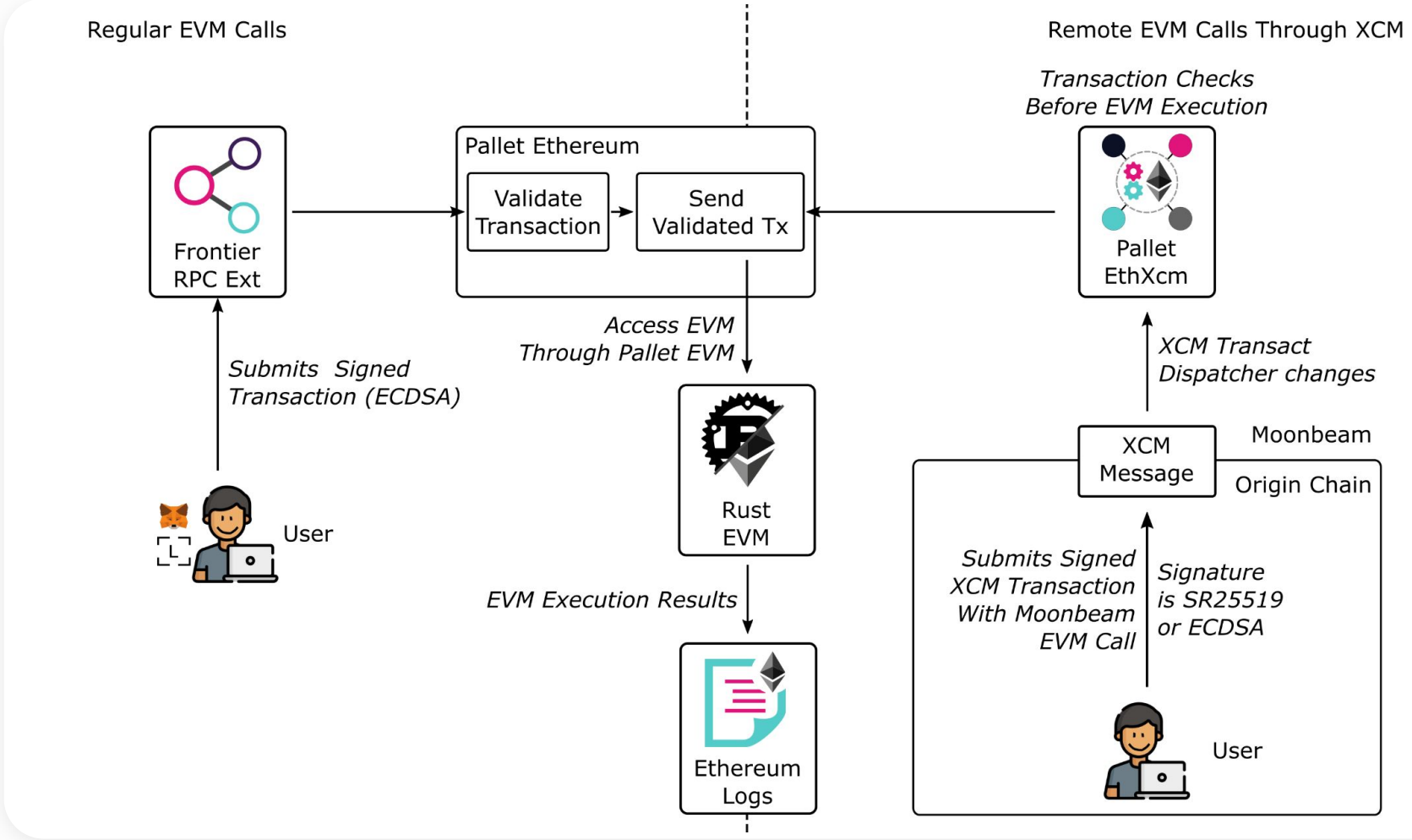
Define the *msg.sender* via a proxy




Regular vs Remote EVM Calls
Happy Path Diagram :)

Regular EVM Calls




Regular EVM Calls
Remote EVM Calls Through XCM
Transaction Checks
Before EVM Execution
Pallet Ethereum
Validate
Transaction
Send
Validated Tx
Frontier
RPC Ext
Pallet
EthXcm
Access EVM
Through Pallet EVM
Submits Signed
Transaction (ECDSA)
XCM Transact
Dispatcher changes
Rust
EVM
Moonbeam
XCM
Message
Origin Chain
User
EVM Execution Results
Submits Signed
XCM Transaction
With Moonbeam
EVM Call
Signature
is SR25519
or ECDSA
Ethereum
Logs
User

https://docs.moonbeam.network/builders/xcm/remote-evm-calls/

Remote EVM Calls
Main Differences

01

Global Nonce
Transactions with no signature can produce EVM Tx Hash collision

02

No Signature -> No ECRECOVER
The *from* is still the *msg.sender* and it is included in the transaction receipt and in the transaction by hash (ETH JSON RPC)

03

Gas Price is Zero
EVM Call is charged at an XCM level, not at an EVM level

```
curl --location --request POST
'https://rpc.api.moonbase.moonbeam.network' \
--header 'Content-Type: application/json' \
--data-raw '{
    "jsonrpc":"2.0",
    "id":1,
    "method":"eth_getTransactionByHash",
    "params":
["0x85735a6be6aa0b3ad5f6ce877d8b9048137876517d9ca5
b309bcd93ae997bf7a"]
  }
'
```

```
{
    "jsonrpc": "2.0",
    "result": {
      "hash": "0x85735a6be6aa0b3ad5f6ce877d8b9048137876517d9ca5b309bcd93ae997bf7a",
      "nonce": "0x1",
      "blockHash": "0xc4b573da6943cc94e55c2fb429160c5b24d91a9da6798102a28dd611c3b76cc0",
      "blockNumber": "0x2e7cf1",
      "transactionIndex": "0x0",
      "from": "0x4e21340c3465ec0aa91542de3d4c5f4fc1def526",
      "to": "0xa72f549a1a12b9b49f30a7f3aeb1f4e96389c5d8",
      "value": "0x0",
      "gasPrice": "0x0",
      "maxFeePerGas": "0x0",
      "maxPriorityFeePerGas": "0x0",
      "gas": "0x11558",
      "input": "0xd09de08a",
      "creates": null,
      "raw": "...",       // Value deleted so that it fits :)
      "publicKey": "...", // Value deleted so that it fits :)
      "chainId": "0x507",
      "standardV": "0x1",
      "v": "0x1",
      "r": "0x1",
      "s": "0x1",
      "accessList": [],
      "type": "0x2"
    },
    "id": 1
  }
```


MultiLocations

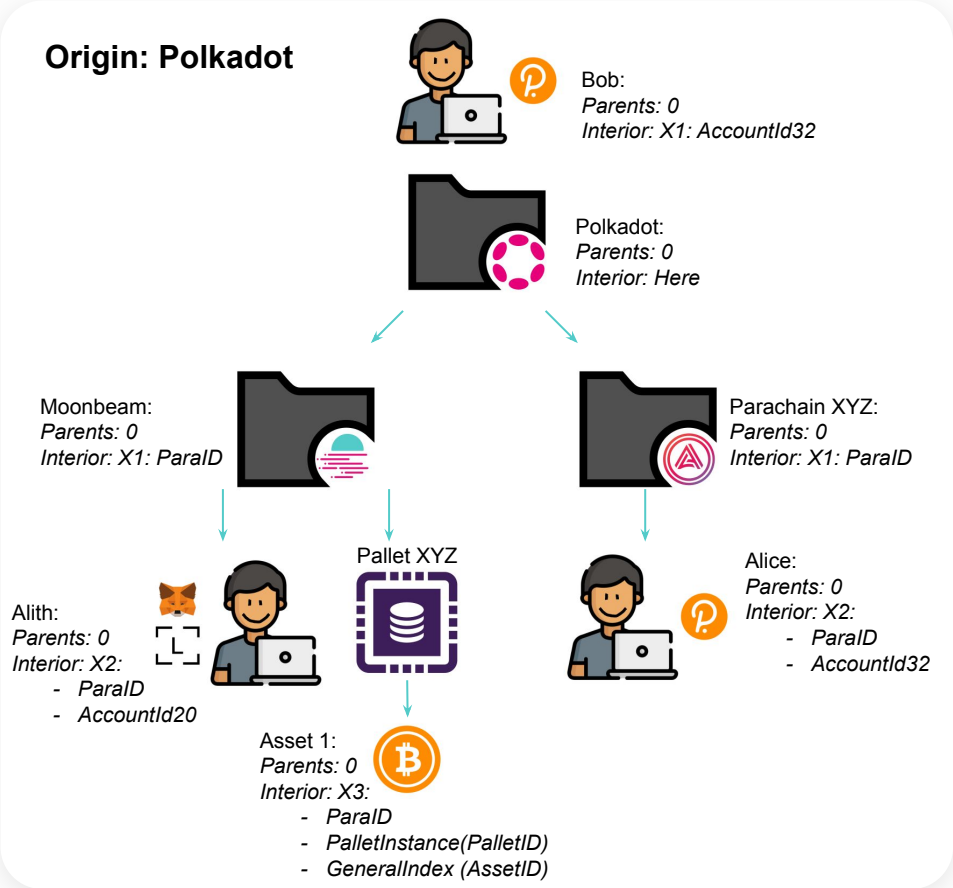
MultiLocation
Represent Locations in the Ecosystem

01 **Folder Path**
Define a target from an origin

02 **Relative to Origin**
MultiLocations are expressed in relative terms to the origin

03 **Represent Anything**
MultiLocations can define Chains, Accounts, Tokens... They contain:

- Parent: hops to an upper level
- Interior: content in that level



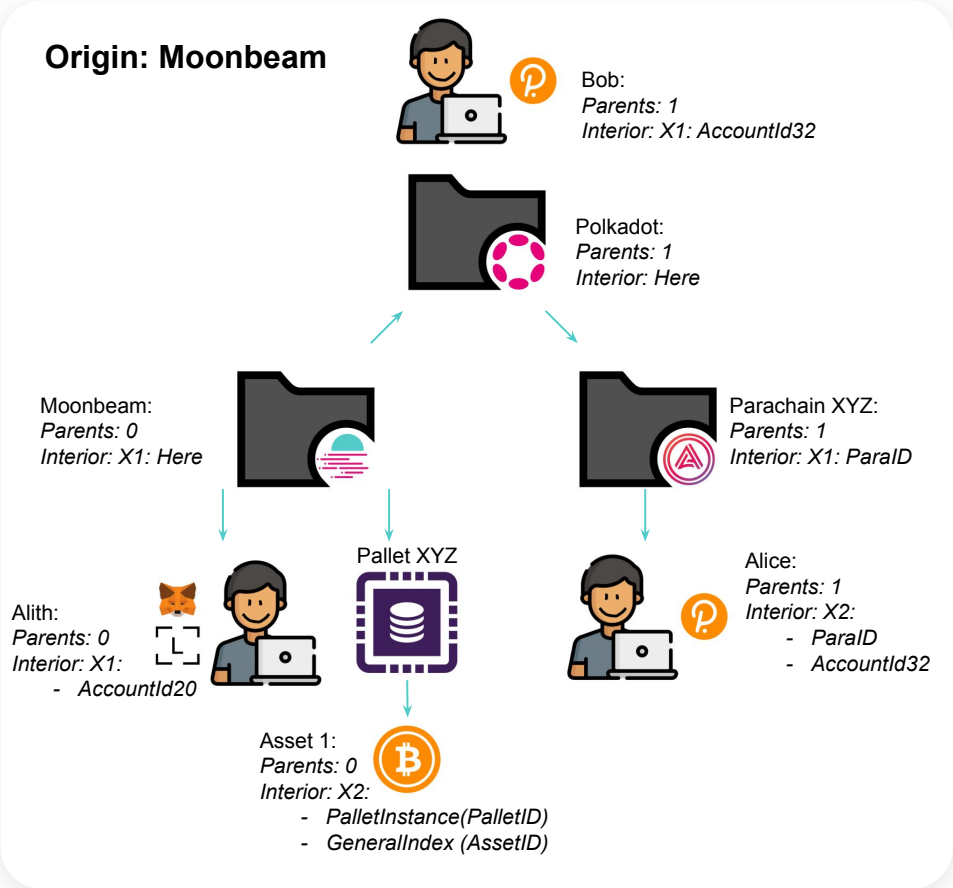
MultiLocation
Represent Locations in the Ecosystem

01 **Folder Path**
Define a target from an origin

02 **Relative to Origin**
MultiLocations are expressed in relative terms to the origin

03 **Represent Anything**
MultiLocations can define Chains, Accounts, Tokens... They contain:

- Parent: hops to an upper level
- Interior: content in that level



XCM Demo:
XCM Balance Transfer

XCM Balance Transfer Demo
What you'll need?

01
Moonbase Alpha Account
Account funded with DEV

02
MultiLocation Derivative Account
Calculate the multilocation-derivative account and funded with KMA

03
Calamari Call Data
SCALE Encoded Call data for a balance transfer

04
Moonbase Alpha Call Data
XCM-Transactor call data

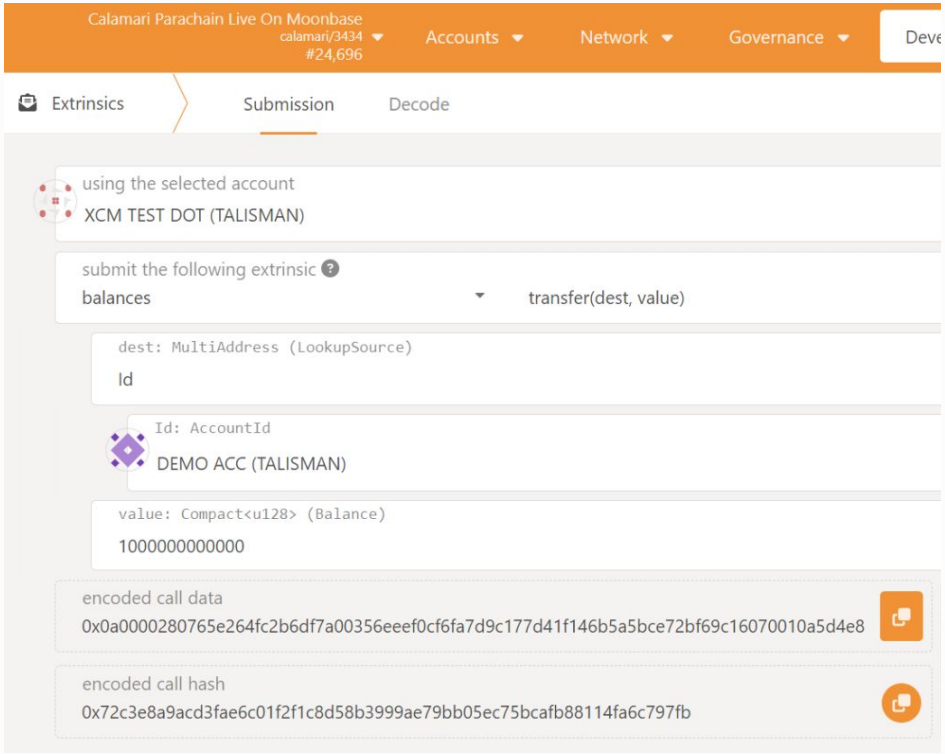
05
Check Tools
Moonscan explorer, Polkadot.js Apps explorer, etc.

06
Important Links
- Polkadot.js Moonbase Alpha
- Docs on Remote XCM Calls
- Docs on Remote EVM Calls
- XCM-tools Github Repo

XCM Balance Transfer Demo
Calamari Call Data

01

Balance Transfer
- Destination Address
- Amount



Calamari Balance Transfer SCALE Encoded Call Data:
0x0a0000280765e264fc2b6df7a00356eeef0cf6fa7d9c177d41f146b5a5bce72bf69c16070010a5d4e8

```
import { ApiPromise, WsProvider } from '@polkadot/api';
const provider = new WsProvider('WS_URL');
const dest= 'dmuqiUYzYsCZvtQNfAbiXP1xk7drm64Yg5d3zq7bXZnvPuv89';
const amount = 1000000000000;

const main = async () => {
  const api = await ApiPromise.create({ provider: provider });

  // Create a transfer extrinsic
  let tx = api.tx.balances.transfer(dest, amount);

  // Get SCALE Encoded Call Data
  let encodedCall = tx.toHex();
  console.log(`Encoded Call Data: ${encodedCall}`);
};
main();
```

Calamari Balance Transfer SCALE Encoded Call Data:
0x0a0000280765e264fc2b6df7a00356eeef0cf6fa7d9c177d41f146b5a5bce72bf69c16070010a5d4e8

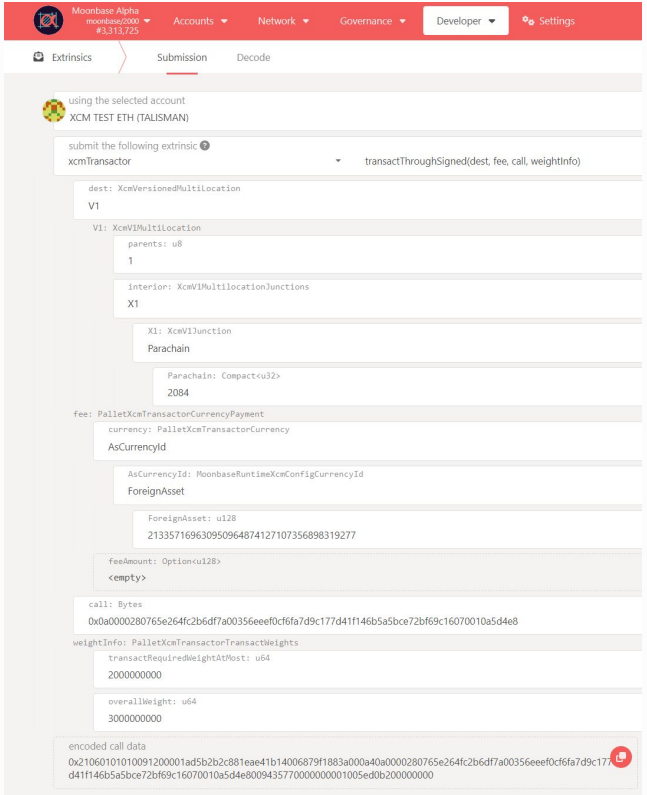
XCM Balance Transfer Demo
Moonbase Alpha Call Data

01

XCM-Transactor

transactThroughSigned:

- Destination Chain
- Fee Information
- Call
- Weight Information



Calamari Balance Transfer SCALE Encoded Call Data:
0x2106010101009120000 1ad5b2b2c881eae41b14006879f1883a000a40a0000280765e264fc2b6df7a00356eeef0cf6fa7d9c177d41f146b5a5bce72bf69c16070010a5d4e80094357700000000001005ed0b200000000

```
import { ApiPromise, WsProvider } from '@polkadot/api';
const provider = new WsProvider('wss://wss.api.moonbase.moonbeam.network');
const dest = { V1: { parents: 1, interior: { X1: { Parachain: 2084 } } } };
const fee = {
  currency: { AsCurrencyId: { ForeignAsset: '213357169630950964874127107356898319277' } },
  feeAmount: null,
};
const call = '0x0a0000280765e264fc2b6df7a00356eeef0cf6fa7d9c177d41f146b5a5bce72bf69c16070010a5d4e8';
const weightInfo = { transactRequiredWeightAtMost: 2000000000, overallWeight: 3000000000 };
const main = async () => {
  const api = await ApiPromise.create({ provider: provider });
  // Create a transfer extrinsic
  let tx = api.tx.xcmTransactor.transactThroughSigned(dest, fee, call, weightInfo);
  // Get SCALE Encoded Call Data
  let encodedCall = tx.toHex();
  console.log(`Encoded Call Data: ${encodedCall}`);
};
main();
```

Moonbase XCM-Transactor Transact Through Signed SCALE Encoded Call Data:
0x5d010421060101010091200001ad5b2b2c881eae41b14006879f1883a000a40a0000280765e264fc2b6df7a00356eeef0cf6fa7d9c177d41f146b5a5bce72bf69c16070010a5d4e80094357700000000001005ed0b200000000

How to Contact Us

If you want to know more…

https://moonbeam.network

https://docs.moonbeam.network/

@Moonbeam_Official

https://discord.gg/PfpUATX

@MoonbeamNetwork

Today's presentation :)

https://bit.ly/PoM2022Workshop