XRP LEDGER

EVM and the XRP Ledger: The Next Decade of DeFi

Jason Tigas - Developer Advocate, Ripple

Introduction

Jason Tigas
Developer Advocate, Ripple
Twitter: @jayCryp_TO
Github: jiveyTO

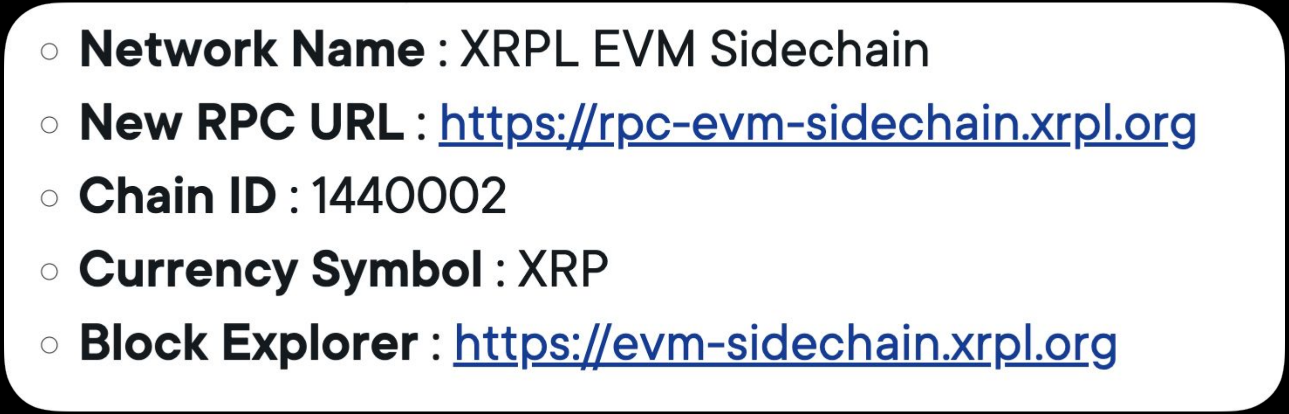
linktr.ee/jaycryp_to

Get the Code!

https://tinyurl.com/xrpl-evm-workshop

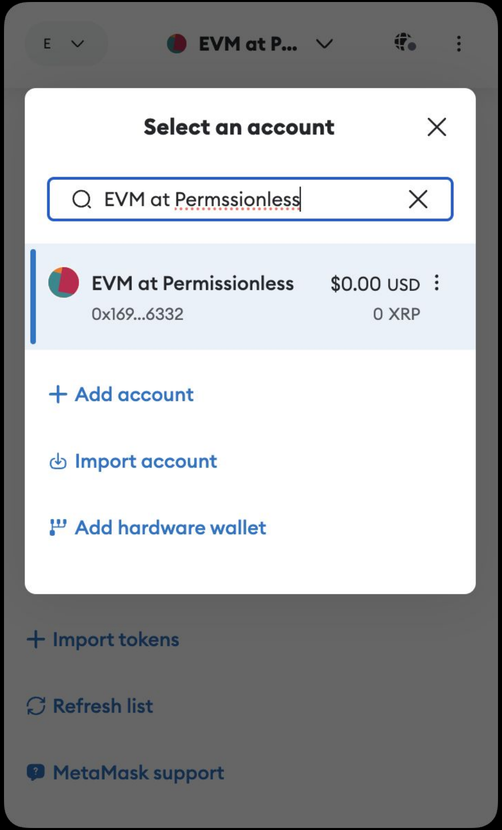
Setting up Metamask

Add a custom network using the details below:



- **Network Name** : XRPL EVM Sidechain
- **New RPC URL** : https://rpc-evm-sidechain.xrpl.org
- **Chain ID** : 1440002
- **Currency Symbol** : XRP
- **Block Explorer** : https://evm-sidechain.xrpl.org

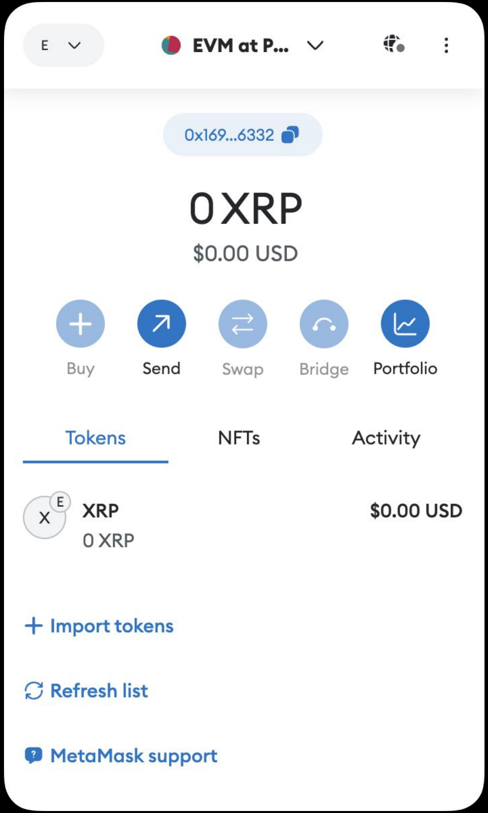
Add a new account



Select an account

EVM at Permssionless

EVM at Permissionless $0.00 USD
0x169...6332 0 XRP



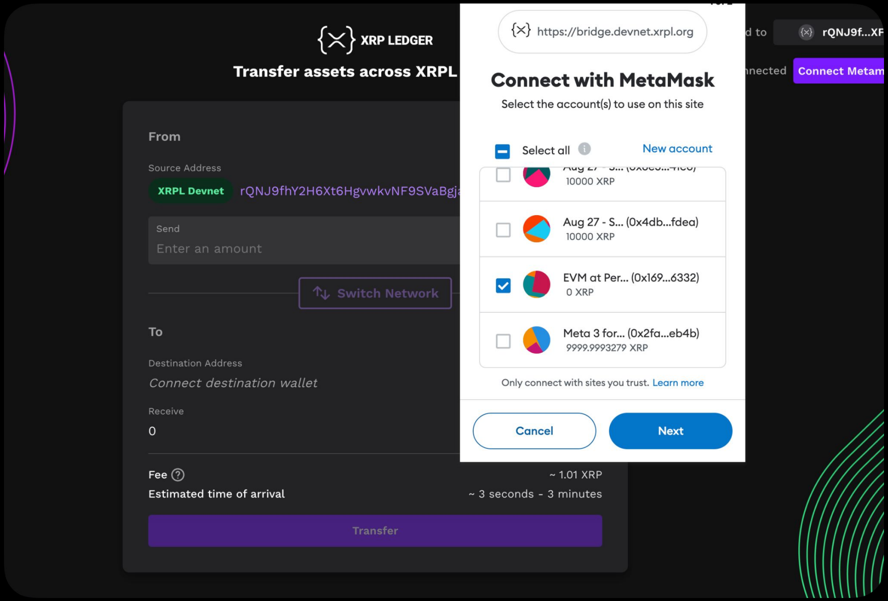
0x169...6332

0 XRP
$0.00 USD

XRP $0.00 USD
0 XRP

Bridge over some XRP (createAccount)

https://bridge.devnet.xrpl.org



XRP LEDGER

Transfer assets across XRPL

From

Source Address

XRPL Devnet rQNJ9fhY2H6Xt6HgvwkvNF9SVaBgj

Send

Enter an amount

Switch Network

To

Destination Address

Connect destination wallet

Receive

0

Fee ~ 1.01 XRP

Estimated time of arrival ~ 3 seconds - 3 minutes

Transfer

d to rQNJ9f...XF

nected Connect Metam

https://bridge.devnet.xrpl.org

Connect with MetaMask

Select the account(s) to use on this site

Select all New account

☐	Aug 27 - S... (0x0e0...4fc0)	10000 XRP
☐	Aug 27 - S... (0x4db...fdea)	10000 XRP
✓	EVM at Per... (0x169...6332)	0 XRP
☐	Meta 3 for... (0x2fa...eb4b)	9999.9993279 XRP

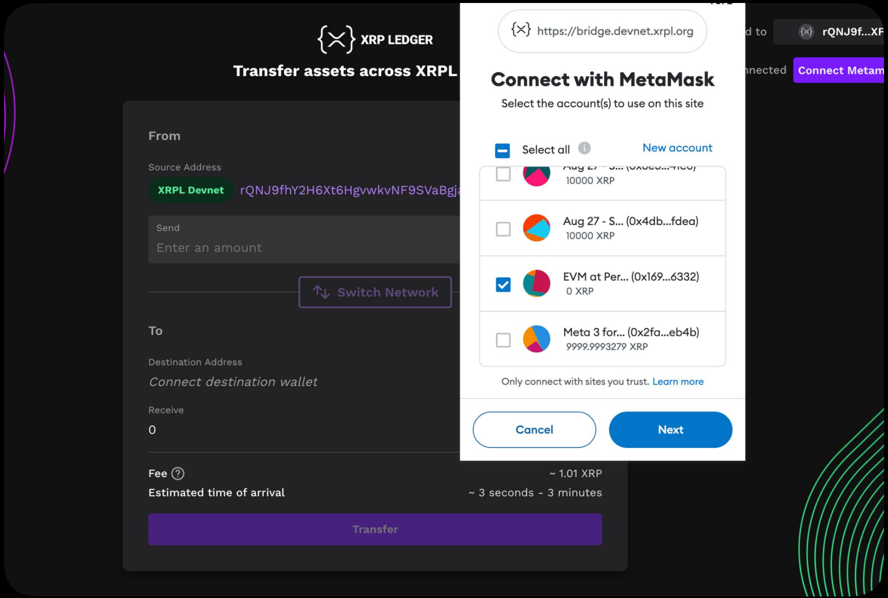
Only connect with sites you trust. Learn more

Cancel Next

XRPL Balance	EVM Balance
1,000,000	0
999,990 - 10 reserve	0
999,899 - 100 transfer - 1 reward	100 + 100 transfer

Bridge over some XRP (claim->commit)

https://bridge.devnet.xrpl.org



{X} XRP LEDGER

Transfer assets across XRPL

d to rQNJ9f...XF

nnected Connect Metam

From

Source Address

XRPL Devnet rQNJ9fhY2H6Xt6HgvwkvNF9SVaBgj

Send

Enter an amount

Switch Network

To

Destination Address

Connect destination wallet

Receive

0

Fee ~ 1.01 XRP

Estimated time of arrival ~ 3 seconds - 3 minutes

Transfer

https://bridge.devnet.xrpl.org

Connect with MetaMask

Select the account(s) to use on this site

Select all — New account

	Account	Balance
☐	Aug 27 - S... (0x0e3...41c0)	10000 XRP
☐	Aug 27 - S... (0x4db...fdea)	10000 XRP
✓	EVM at Per... (0x169...6332)	0 XRP
☐	Meta 3 for... (0x2fa...eb4b)	9999.9993279 XRP

Only connect with sites you trust. Learn more

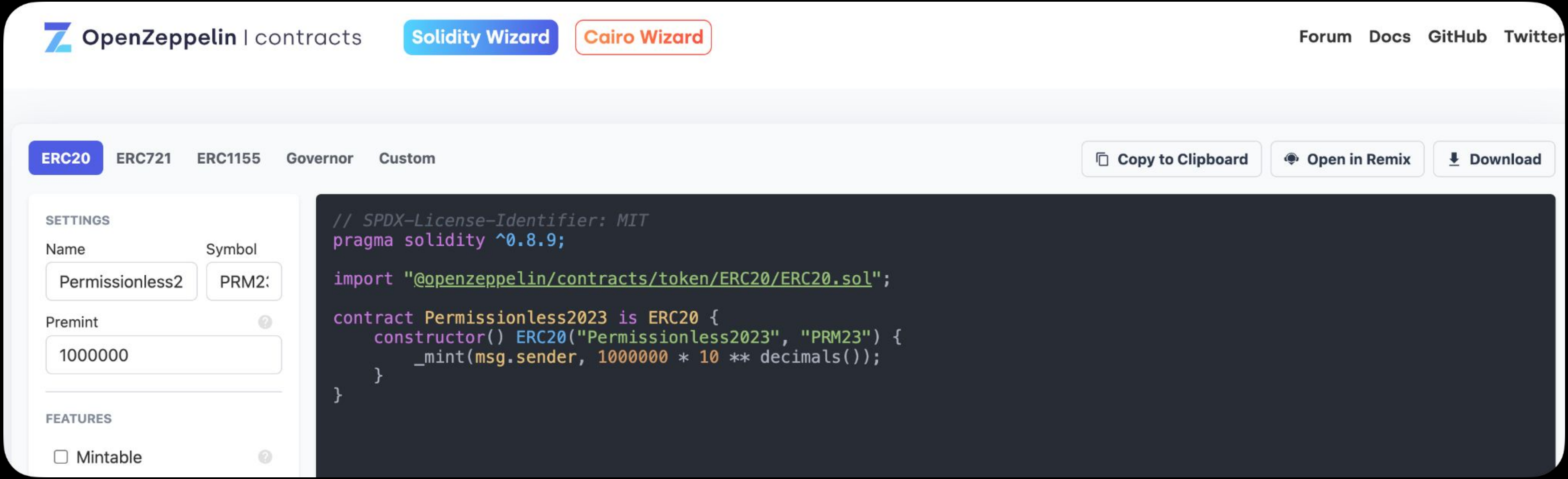
Cancel — Next

XRPL Balance	EVM Balance
999,899	100
999,799 - 100 transfer	99 -1 claimID
	199 + 100 transfer

* see Metamask for balance as explorer is laggy

Create your own ERC20 token

https://wizard.openzeppelin.com/#erc20



SETTINGS

Name: Permissionless2

Symbol: PRM2:

Premint: 1000000

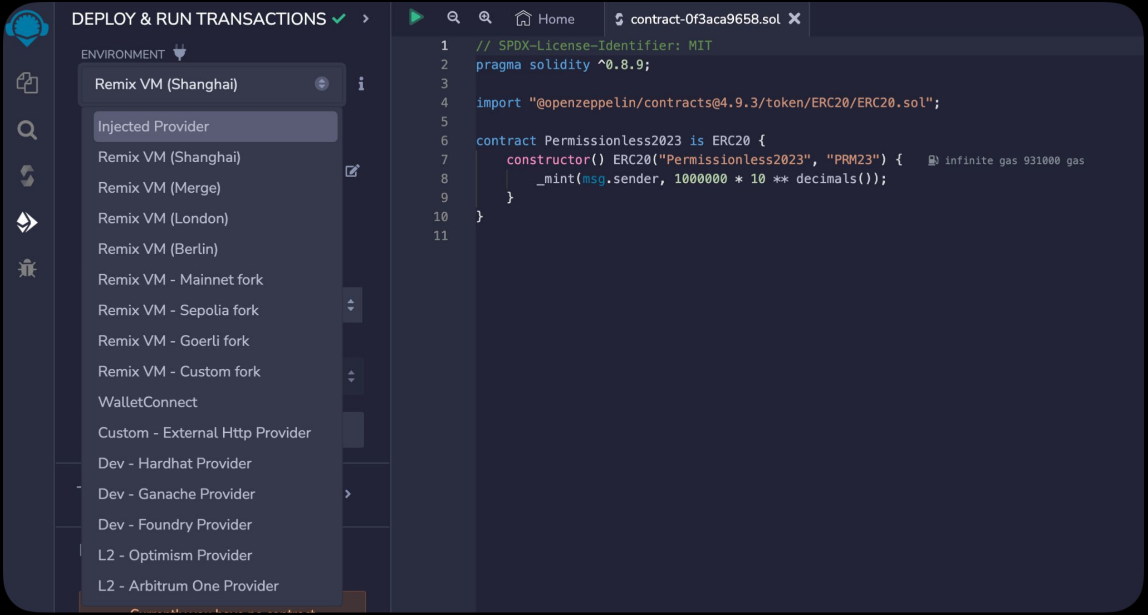
FEATURES

☐ Mintable

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract Permissionless2023 is ERC20 {
    constructor() ERC20("Permissionless2023", "PRM23") {
        _mint(msg.sender, 1000000 * 10 ** decimals());
    }
}
```

Open and deploy in Remix

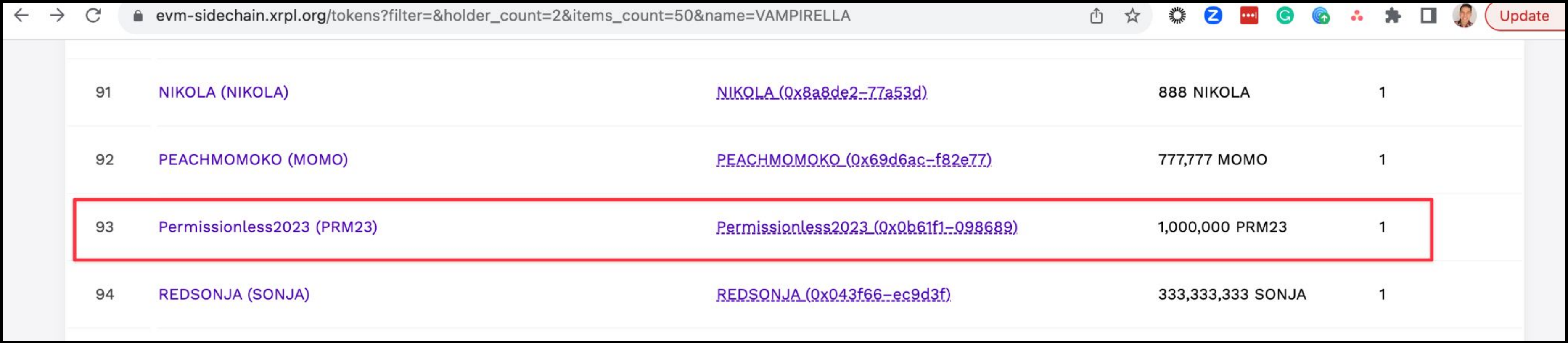
Use “Injected Provider” to compile and deploy



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts@4.9.3/token/ERC20/ERC20.sol";

contract Permissionless2023 is ERC20 {
    constructor() ERC20("Permissionless2023", "PRM23") {    infinite gas 931000 gas
        _mint(msg.sender, 1000000 * 10 ** decimals());
    }
}
```

View your token on the EVM explorer

https://evm-sidechain.xrpl.org/tokens



evm-sidechain.xrpl.org/tokens?filter=&holder_count=2&items_count=50&name=VAMPIRELLA

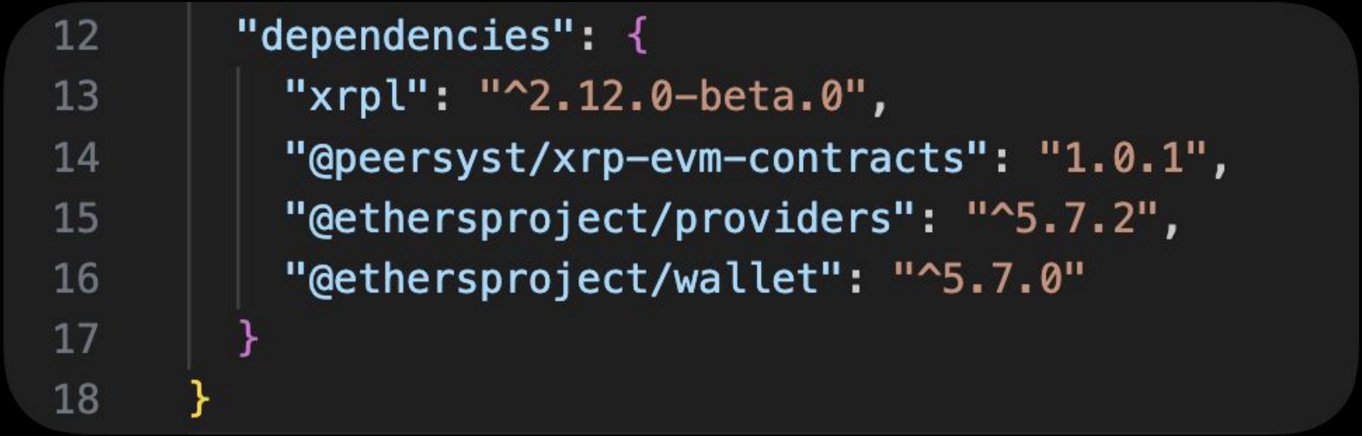
91	NIKOLA (NIKOLA)	NIKOLA (0x8a8de2–77a53d)	888 NIKOLA	1
92	PEACHMOMOKO (MOMO)	PEACHMOMOKO (0x69d6ac–f82e77)	777,777 MOMO	1
93	Permissionless2023 (PRM23)	Permissionless2023 (0x0b61f1–098689)	1,000,000 PRM23	1
94	REDSONJA (SONJA)	REDSONJA (0x043f66–ec9d3f)	333,333,333 SONJA	1

Let’s code

https://tinyurl.com/xrpl-evm-workshop

Setup your NPM project

1. Create a new directory and cd into it
2. npm init -y
3. Copy the dependencies from package.json
4. npm install



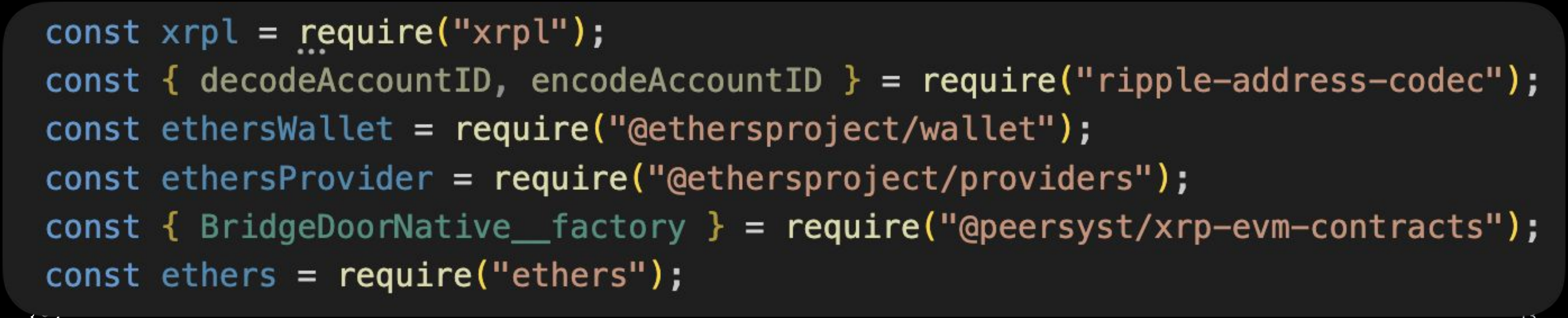
```
    "dependencies": {
      "xrpl": "^2.12.0-beta.0",
      "@peersyst/xrp-evm-contracts": "1.0.1",
      "@ethersproject/providers": "^5.7.2",
      "@ethersproject/wallet": "^5.7.0"
    }
}
```

tinyurl.com/xrpl-evm-workshop

Bridge.js

1. Create a new file called bridge.js
2. Import the packages

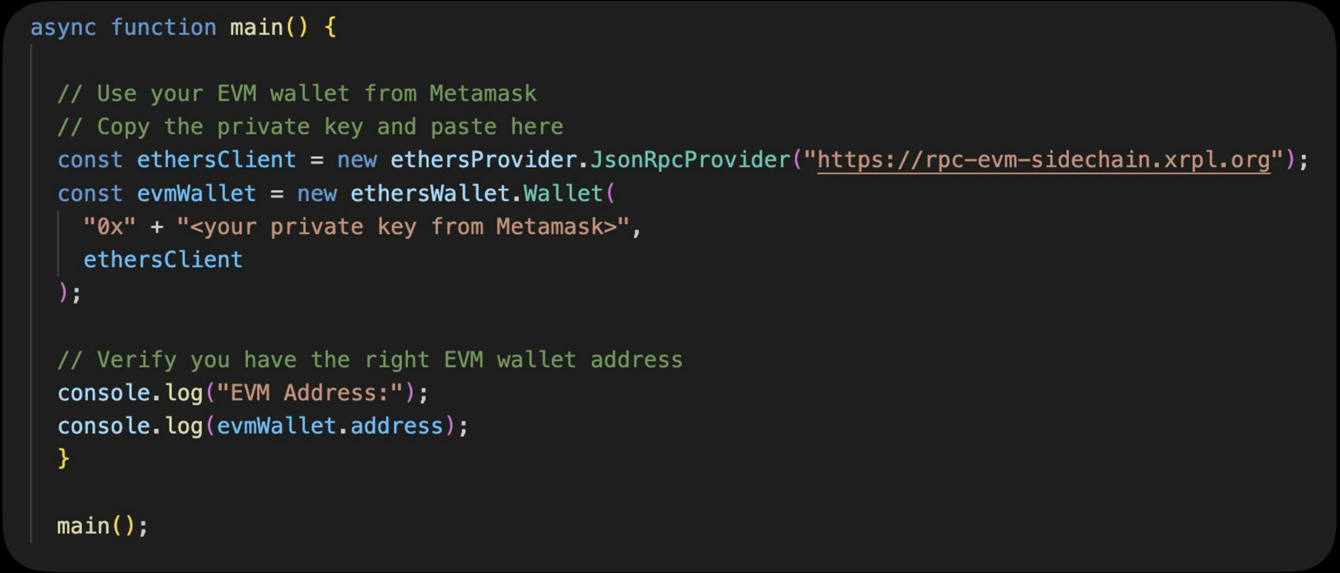
tinyurl.com/xrpl-evm-workshop



```javascript
const xrpl = require("xrpl");
const { decodeAccountID, encodeAccountID } = require("ripple-address-codec");
const ethersWallet = require("@ethersproject/wallet");
const ethersProvider = require("@ethersproject/providers");
const { BridgeDoorNative__factory } = require("@peersyst/xrp-evm-contracts");
const ethers = require("ethers");
```

Import your Metamask EVM wallet

1. Create a new account in Metamask
2. Get your private key Account Details -> Show Private Key
3. Paste your private key into the script



```javascript
async function main() {

  // Use your EVM wallet from Metamask
  // Copy the private key and paste here
  const ethersClient = new ethersProvider.JsonRpcProvider("https://rpc-evm-sidechain.xrpl.org");
  const evmWallet = new ethersWallet.Wallet(
    "0x" + "<your private key from Metamask>",
    ethersClient
  );

  // Verify you have the right EVM wallet address
  console.log("EVM Address:");
  console.log(evmWallet.address);
  }

  main();
```

Retrieve the bridge data

```javascript
// Retreive the bridge data
const lockingClient = new xrpl.Client(
  'wss://sidechain-net1.devnet.rippletest.net:51233',
)

await lockingClient.connect()
const lockingChainDoor = 'rEAjhZHotzo2jqPbjFpAEacgwc5XoUppgo'

const accountObjectsRequest = {
  command: 'account_objects',
  account: lockingChainDoor,
  type: 'bridge',
}

const bridgeData = await lockingClient.request(accountObjectsRequest)
const bridge = bridgeData.result.account_objects[0].XChainBridge
const bridgeDataSignatureReward = bridgeData.result.account_objects[0].SignatureReward
const bridgeDataMin = bridgeData.result.account_objects[0].MinAccountCreateAmount
console.log("Bridge  signature reward:")
console.log(bridgeDataSignatureReward)
console.log("Bridge account min:")
console.log(bridgeDataMin)
console.log("Bridge data:")
console.log(bridge)
```

Going forwards you need a disconnect line at the bottom of your script:

lockingClient.disconnect()

Create an XRPL devnet wallet with test XRP

1. Run once to get the private key
2. Comment out the fundWallet line and uncomment the fromSeed line

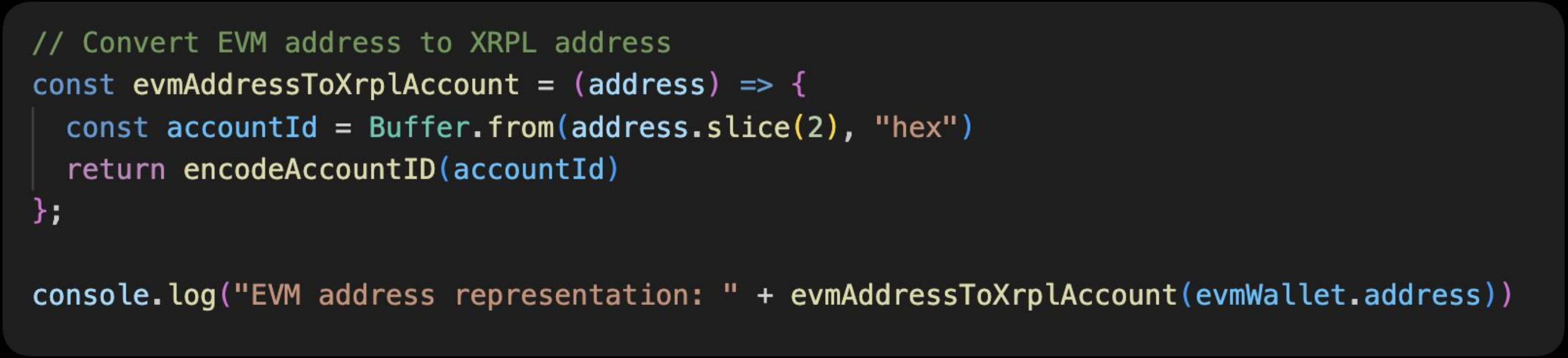
tinyurl.com/xrpl-evm-workshop

```
// Get an XRPL devnet wallet
const wallet1 = await lockingClient.fundWallet()
// const wallet1 = xrpl.Wallet.fromSeed("<XRPL devnet seed>")
// Note your address here: <XRPL devnet address>
// View it on the explorer:
// https://custom.xrpl.org/sidechain-net1.devnet.rippletest.net/accounts/<XRPL devnet address>

console.log("Wallet1:")
console.log(wallet1)
```

Convert EVM address to XRPL address

See the code for both functions:
evmAddressToXrplAccount and **xrplAccountToEvmAddress**

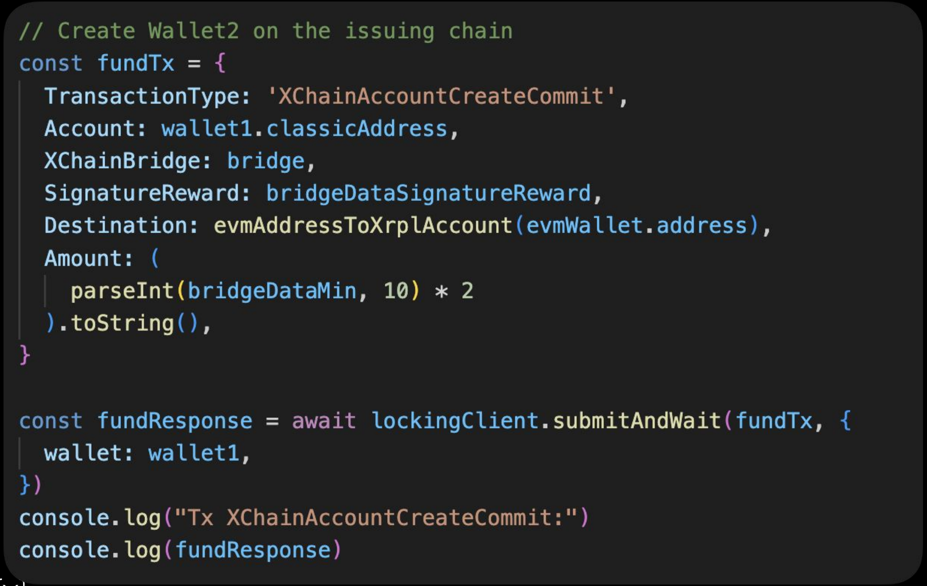
tinyurl.com/xrpl-evm-workshop



```javascript
// Convert EVM address to XRPL address
const evmAddressToXrplAccount = (address) => {
  const accountId = Buffer.from(address.slice(2), "hex")
  return encodeAccountID(accountId)
};

console.log("EVM address representation: " + evmAddressToXrplAccount(evmWallet.address))
```

Send XChainAccountCreateCommit

- Create the new account on the EVM sidechain
- Run and check Metamask
- Comment out submitAndWait



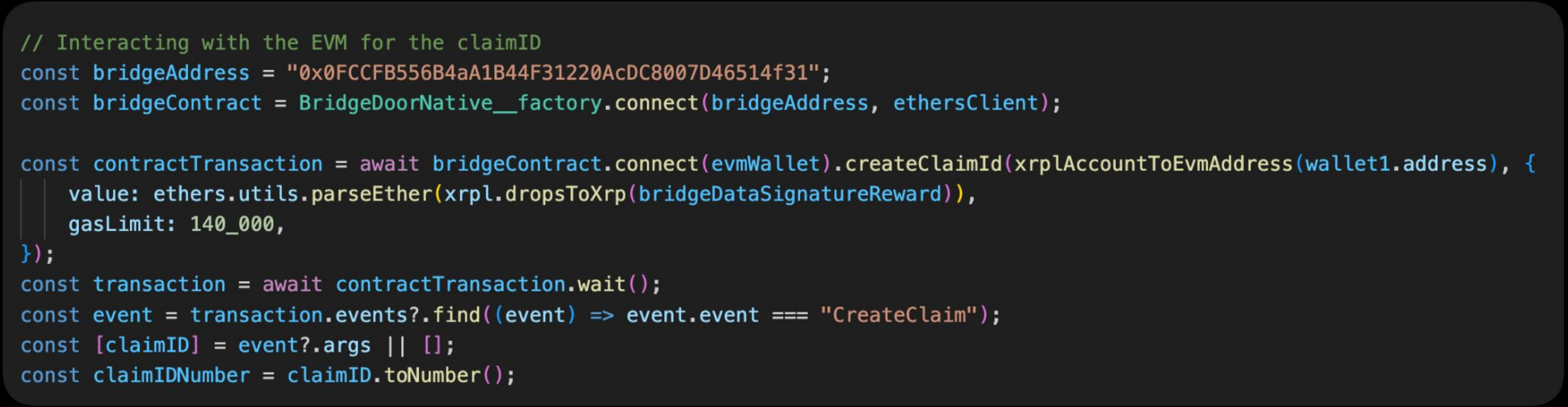
```
// Create Wallet2 on the issuing chain
const fundTx = {
  TransactionType: 'XChainAccountCreateCommit',
  Account: wallet1.classicAddress,
  XChainBridge: bridge,
  SignatureReward: bridgeDataSignatureReward,
  Destination: evmAddressToXrplAccount(evmWallet.address),
  Amount: (
    parseInt(bridgeDataMin, 10) * 2
  ).toString(),
}

const fundResponse = await lockingClient.submitAndWait(fundTx, {
  wallet: wallet1,
})
console.log("Tx XChainAccountCreateCommit:")
console.log(fundResponse)
```

tinyurl.com/xrpl-evm-workshop

Send XChainCreateClaimID

- Create the new account on the EVM sidechain
- Run and check Metamask

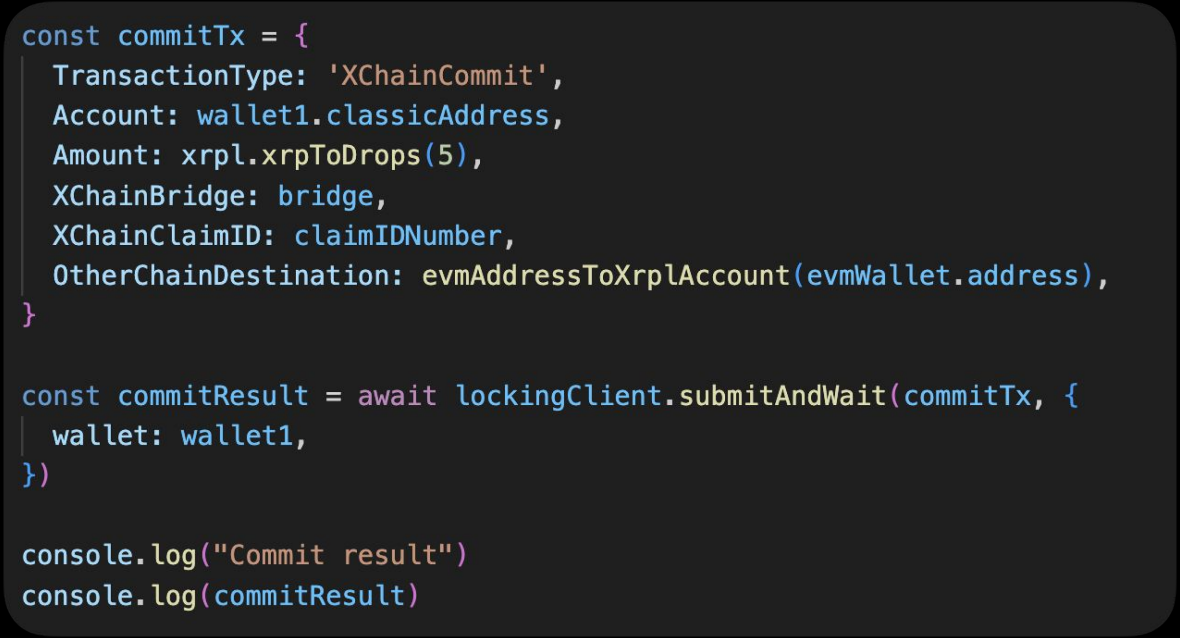
tinyurl.com/xrpl-evm-workshop



```
// Interacting with the EVM for the claimID
const bridgeAddress = "0x0FCCFB556B4aA1B44F31220AcDC8007D46514f31";
const bridgeContract = BridgeDoorNative__factory.connect(bridgeAddress, ethersClient);

const contractTransaction = await bridgeContract.connect(evmWallet).createClaimId(xrplAccountToEvmAddress(wallet1.address), {
    value: ethers.utils.parseEther(xrpl.dropsToXrp(bridgeDataSignatureReward)),
    gasLimit: 140_000,
});
const transaction = await contractTransaction.wait();
const event = transaction.events?.find((event) => event.event === "CreateClaim");
const [claimID] = event?.args || [];
const claimIDNumber = claimID.toNumber();
```

Send XChainCommit

- Commit the funds on the locking chain



```
const commitTx = {
  TransactionType: 'XChainCommit',
  Account: wallet1.classicAddress,
  Amount: xrpl.xrpToDrops(5),
  XChainBridge: bridge,
  XChainClaimID: claimIDNumber,
  OtherChainDestination: evmAddressToXrplAccount(evmWallet.address),
}

const commitResult = await lockingClient.submitAndWait(commitTx, {
  wallet: wallet1,
})

console.log("Commit result")
console.log(commitResult)
```

tinyurl.com/xrpl-evm-workshop