XRP LEDGER

Interoperability:

XRP Ledger and EVM

Introduction

Vito Tumas
Software Engineer, Ripple

LinkedIn: @vtumas
Twitter: @v_tumas
Github: Tapanito

Get the Code!

https://github.com/Tapanito/xrpl-evm-sidechain

Goals

1. Understand XRP Ledger
2. Understand Interoperability
3. Cross the blockchain bridge

XRP Ledger (XRPL) launched in 2012 to address limitations of crypto and fiat currencies for financial use cases, specifically payments

The differences between XRP Ledger, XRP, and Ripple

XRP LEDGER

Layer-1 Blockchain

The XRP Ledger is a secure, decentralized and public blockchain with ultra-low transaction fees.

XRP

Native Digital Asset

XRP is the native digital asset (token) of XRP Ledger, similar to what ETH is for Ethereum.

XRP is one of the only two cryptocurrencies with clear regulatory status in the US.

ripple

Crypto Solutions Company

Ripple is a technology company that builds crypto solutions for business.

Ripple is one of many developers building on and contributing to the XRP Ledger.

Source: https://ripple.com/xrp/, https://xrpl.org/faq.htm, https://ripple.com/insights/q1-2023-xrp-markets-report/l

XRPL Native Features

Payment

Transfers

Enables the transfer of XRP and the creation of currencies and other fungible tokens.

Buy/Sell

Exchange

Enables trading on the decentralized exchange by letting users place orders on an Order Book or swap against an Automated Market Maker.

NFT

NFT

Enables the creation and management of NFTs, including setting royalties for creators.

Clawback

Compliance

Enables issuers to choose the option of reclaiming issued assets through a clawback feature.

With over a decade of error-free performance XRP Ledger provides rock-solid foundations for innovation

100%

decentralized blockchain with 600+ nodes processing transactions and maintaining the ledger

1750+

unique apps and exchanges on mainnet built by a diverse set of global developers

4.5M+

active XRP wallet holders around the world

100+

validators operated by universities, exchanges, businesses, & individuals

2.6B+

transactions processed representing over $1T in value moved between counterparties

$30B+

market capitalization of XRP

Source: CoinMarketCap, July 2023, https://livenet.xrpl.org/, https://xrpscan.com/

EVM Sidechain

Why EVM Sidechain?

- Lack of general purpose smart contract support
- Connect with the EVM ecosystem developers

EVM Compatibility on different blockchains

Blockchain Ecosystem	**EVM Compatibility Solution/Project**
Ethereum	Native
Solana	Neon
Polkadot	Moonbeam
Cosmos	Evmos
Polygon	zkEVM
BNB Chain	BNB Smart chain
Avalanche	Avalanche C-chain
XRPL	**EVM Sidechain**

What is it?

The EVM Sidechain enables the ability to interact or deploy smart contracts written in Solidity with a secure bridge to XRPL Mainnet

XRPL Mainnet

Federated Byzantine Consensus

XRP Native Token

Rippled client

XRPL explorer

Example Compatible Wallets

← Secure Bridge →

EVM Sidechain

Proof-of-Authority Consensus

XRP Gas Token

Ethermint client

EVM Block explorer

Example Compatible Wallets

Why is that interesting?

EVM apps can now access and benefit from the XRPL ecosystem

01

Bridge to the XRPL ecosystem

Any Solidity app written for Ethereum / EVM can access liquidity and user base of XRPL Mainnet

02

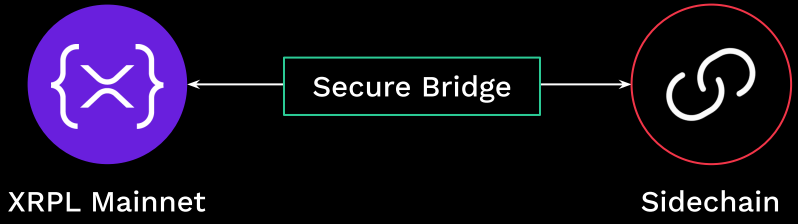
Optimized for DeFi

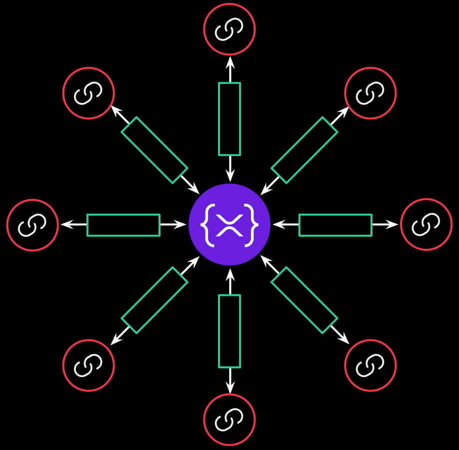
Secure bridges, enhanced scalability and fast transaction finality makes the EVM optimized for financial use cases, like DeFi and payments

03

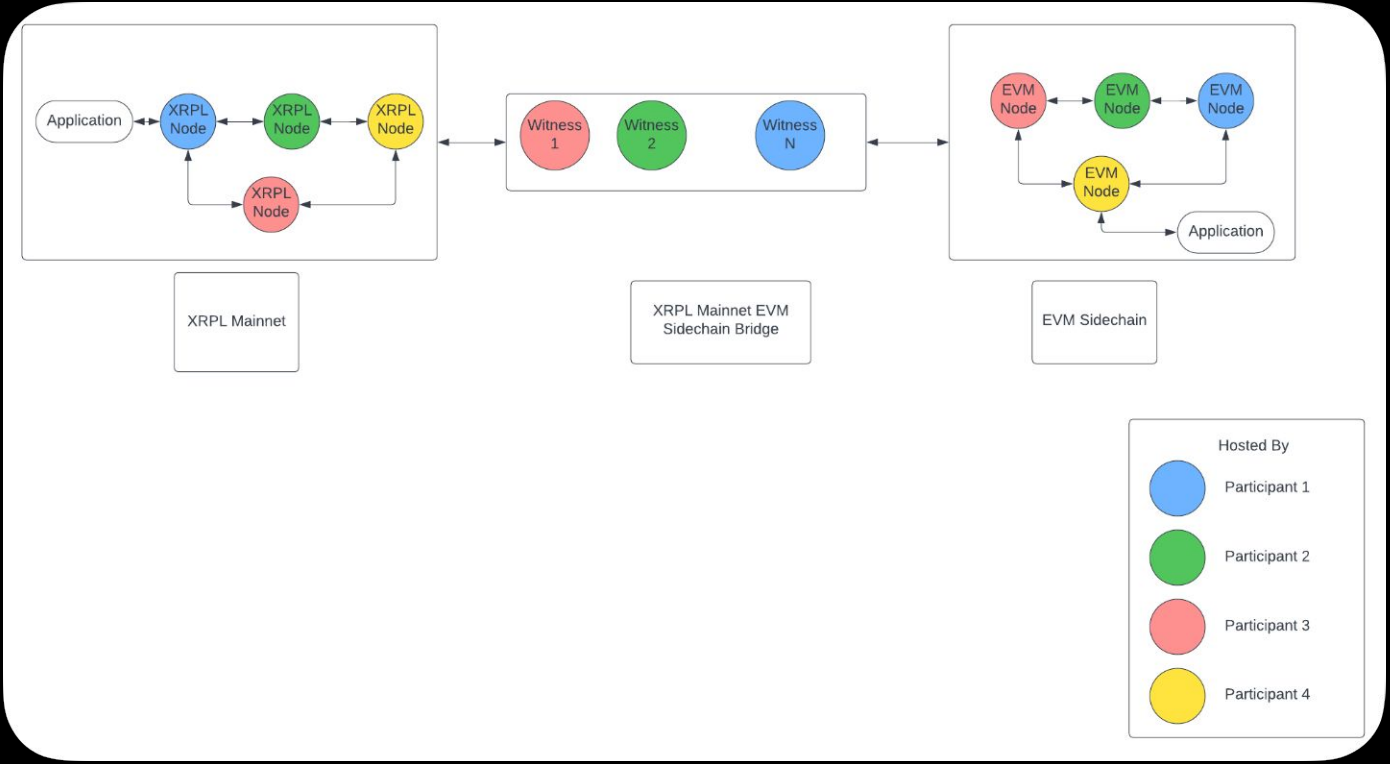
Easy to Build

Build using familiar Ethereum-based tools, wallets, explorers, and apps like MetaMask, Foundry, and Truffle

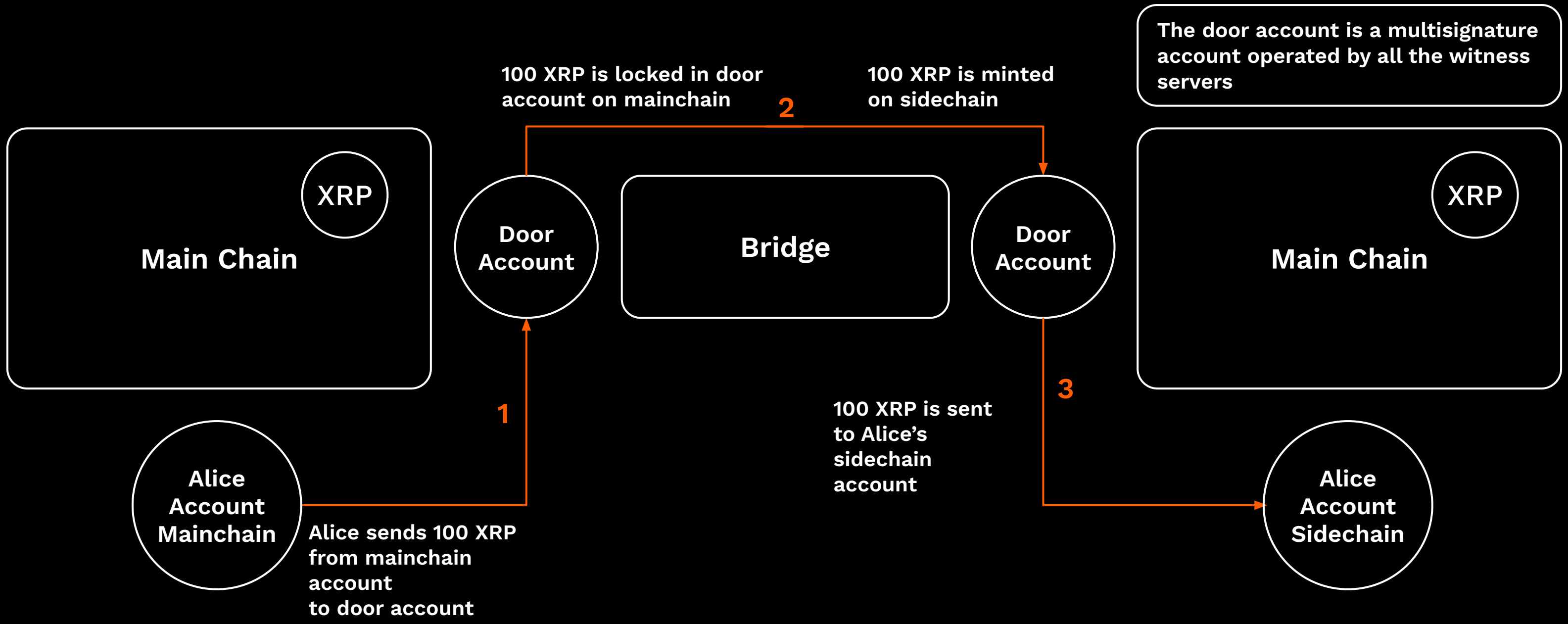

Secure Bridge
XRPL Mainnet
Sidechain



EVM Sidechain Concept



Sidechains - Flow of Funds Mainchain -> Sidechain



The door account is a multisignature account operated by all the witness servers
100 XRP is locked in door account on mainchain
2
100 XRP is minted on sidechain
XRP
Main Chain
Door Account
Bridge
Door Account
XRP
Main Chain
1
3
100 XRP is sent to Alice's sidechain account
Alice Account Mainchain
Alice sends 100 XRP from mainchain account to door account
Alice Account Sidechain

XRP LEDGER

EVM Sidechain

Elevate your DeFi app by
capturing 4.5M XRP holders

Join the Discord to get
started on Devnet

discord.gg/xrplevm

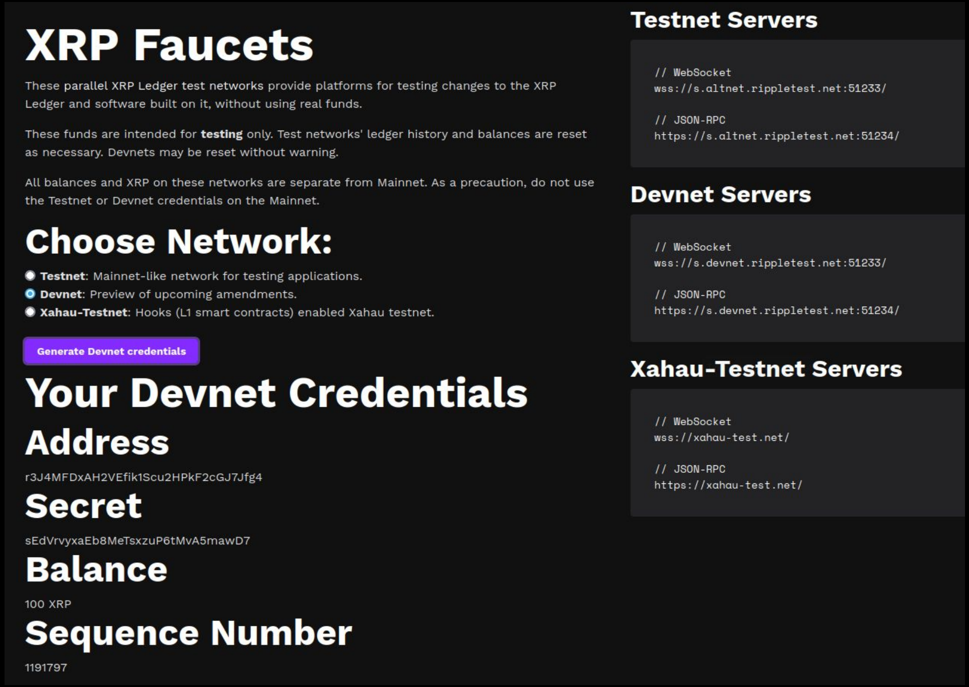
Developed by

ripple | peersyst

Setting up Wallets

Setting up XRP Ledger Devnet

https://xrpl.org/resources/dev-tools/xrp-faucets/



XRP Faucets

These parallel XRP Ledger test networks provide platforms for testing changes to the XRP Ledger and software built on it, without using real funds.

These funds are intended for **testing** only. Test networks' ledger history and balances are reset as necessary. Devnets may be reset without warning.

All balances and XRP on these networks are separate from Mainnet. As a precaution, do not use the Testnet or Devnet credentials on the Mainnet.

Choose Network:

- **Testnet**: Mainnet-like network for testing applications.
- [✓ **Devnet**: Preview of upcoming amendments.]
- **Xahau-Testnet**: Hooks (L1 smart contracts) enabled Xahau testnet.

Generate Devnet credentials

Your Devnet Credentials

Address

r3J4MFDxAH2VEfik1Scu2HPkF2cGJ7Jfg4

Secret

sEdVrvyxaEb8MeTsxzuP6tMvA5mawD7

Balance

100 XRP

Sequence Number

1191797

Testnet Servers

```
// WebSocket
wss://s.altnet.rippletest.net:51233/

// JSON-RPC
https://s.altnet.rippletest.net:51234/
```

Devnet Servers

```
// WebSocket
wss://s.devnet.rippletest.net:51233/

// JSON-RPC
https://s.devnet.rippletest.net:51234/
```

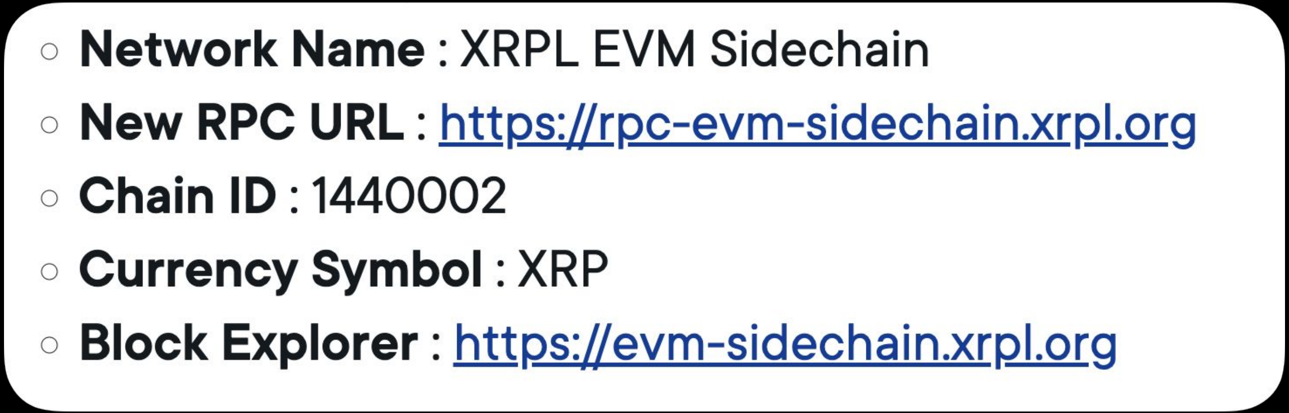
Xahau-Testnet Servers

```
// WebSocket
wss://xahau-test.net/

// JSON-RPC
https://xahau-test.net/
```

Setting up Metamask

https://metamask.io/

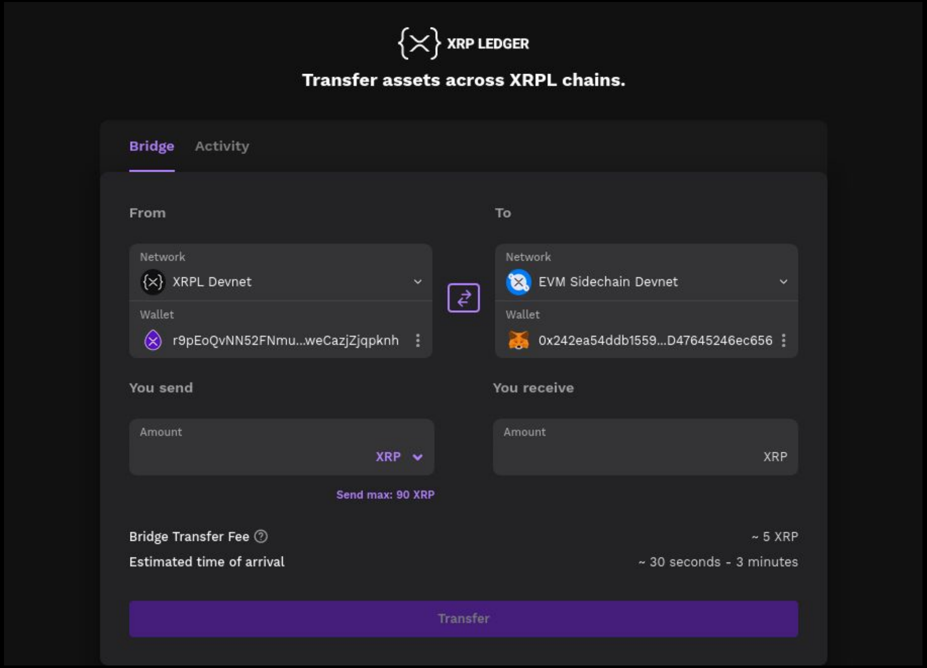
Add a custom network using the details below:



- **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

Bridge over some XRP

https://bridge.devnet.xrpl.org


XRP LEDGER

Transfer assets across XRPL chains.

Bridge Activity

From

Network: XRPL Devnet

Wallet: r9pEoQvNN52FNmu...weCazjZjqpknh

To

Network: EVM Sidechain Devnet

Wallet: 0x242ea54ddb1559...D47645246ec656

You send

Amount XRP

Send max: 90 XRP

You receive

Amount XRP

Bridge Transfer Fee ~ 5 XRP

Estimated time of arrival ~ 30 seconds - 3 minutes

Transfer

Get the Code!

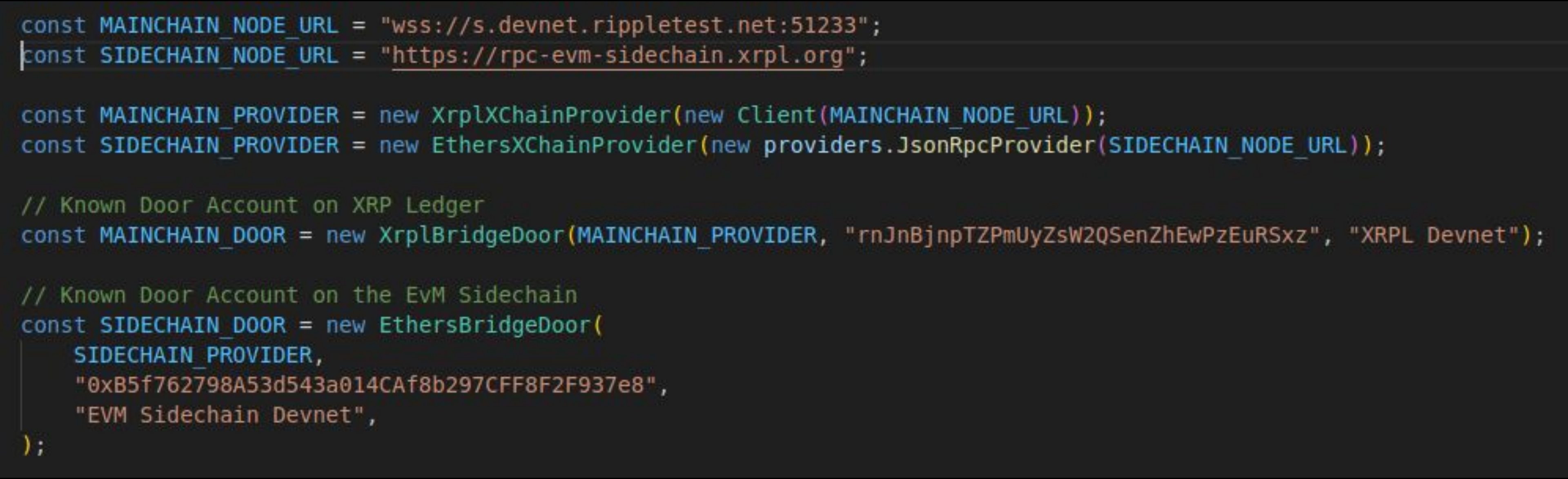
https://github.com/Tapanito/xrpl-evm-sidechain

Get the Code!

1. npm install
2. npm run compile

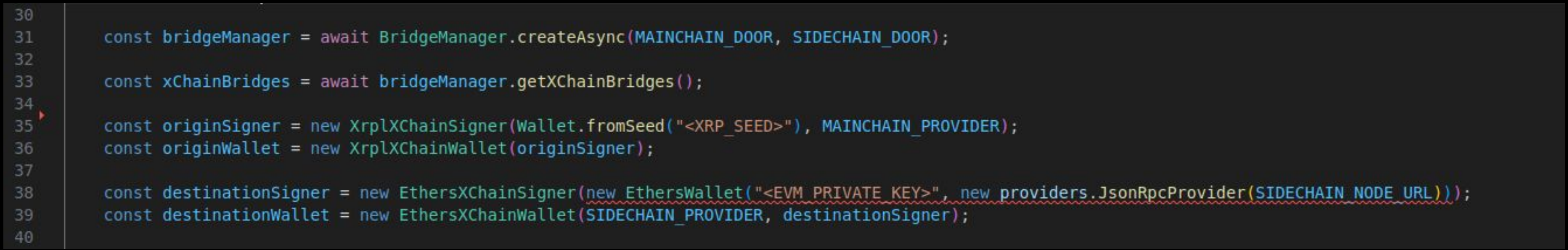
Code for Bridge

Identify the Doors

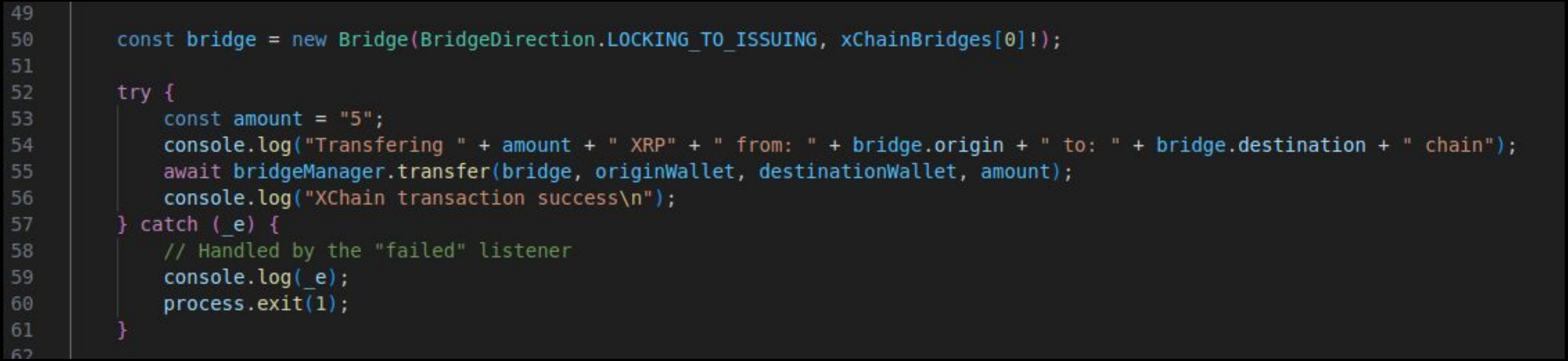


```
const MAINCHAIN_NODE_URL = "wss://s.devnet.rippletest.net:51233";
const SIDECHAIN_NODE_URL = "https://rpc-evm-sidechain.xrpl.org";

const MAINCHAIN_PROVIDER = new XrplXChainProvider(new Client(MAINCHAIN_NODE_URL));
const SIDECHAIN_PROVIDER = new EthersXChainProvider(new providers.JsonRpcProvider(SIDECHAIN_NODE_URL));

// Known Door Account on XRP Ledger
const MAINCHAIN_DOOR = new XrplBridgeDoor(MAINCHAIN_PROVIDER, "rnJnBjnpTZPmUyZsW2QSenZhEwPzEuRSxz", "XRPL Devnet");

// Known Door Account on the EvM Sidechain
const SIDECHAIN_DOOR = new EthersBridgeDoor(
    SIDECHAIN_PROVIDER,
    "0xB5f762798A53d543a014CAf8b297CFF8F2F937e8",
    "EVM Sidechain Devnet",
);
```

Setup Signers on both networks


```
    const bridgeManager = await BridgeManager.createAsync(MAINCHAIN_DOOR, SIDECHAIN_DOOR);

    const xChainBridges = await bridgeManager.getXChainBridges();

    const originSigner = new XrplXChainSigner(Wallet.fromSeed("<XRP_SEED>"), MAINCHAIN_PROVIDER);
    const originWallet = new XrplXChainWallet(originSigner);

    const destinationSigner = new EthersXChainSigner(new EthersWallet("<EVM_PRIVATE_KEY>", new providers.JsonRpcProvider(SIDECHAIN_NODE_URL)));
    const destinationWallet = new EthersXChainWallet(SIDECHAIN_PROVIDER, destinationSigner);

```

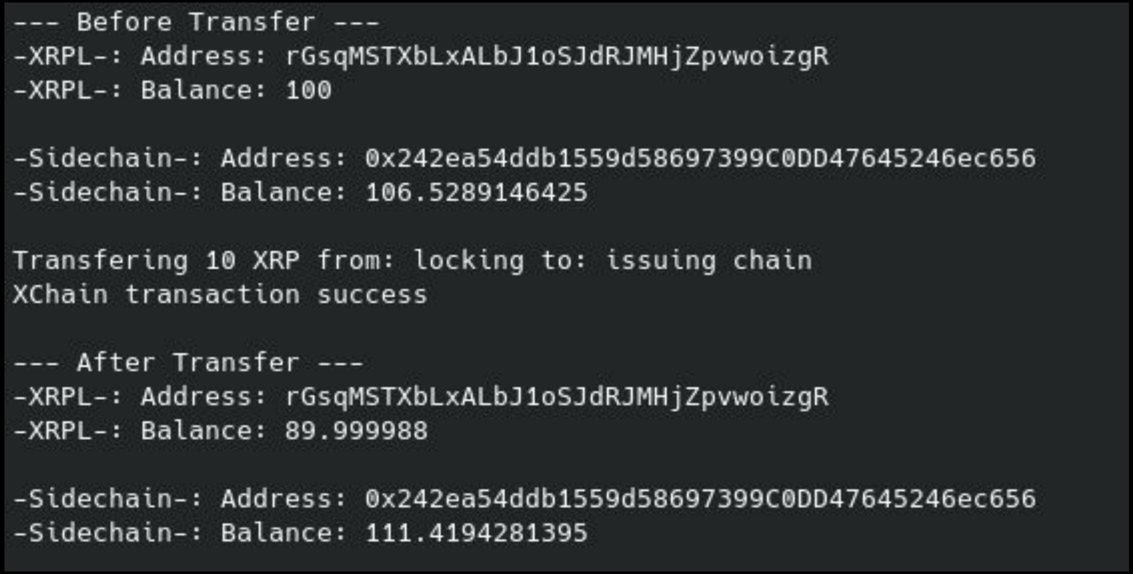
Setup Signers on both networks



```
const bridge = new Bridge(BridgeDirection.LOCKING_TO_ISSUING, xChainBridges[0]!);

try {
    const amount = "5";
    console.log("Transfering " + amount + " XRP" + " from: " + bridge.origin + " to: " + bridge.destination + " chain");
    await bridgeManager.transfer(bridge, originWallet, destinationWallet, amount);
    console.log("XChain transaction success\n");
} catch (_e) {
    // Handled by the "failed" listener
    console.log(_e);
    process.exit(1);
}
```

Setup Signers on both networks

npm run deploy-token

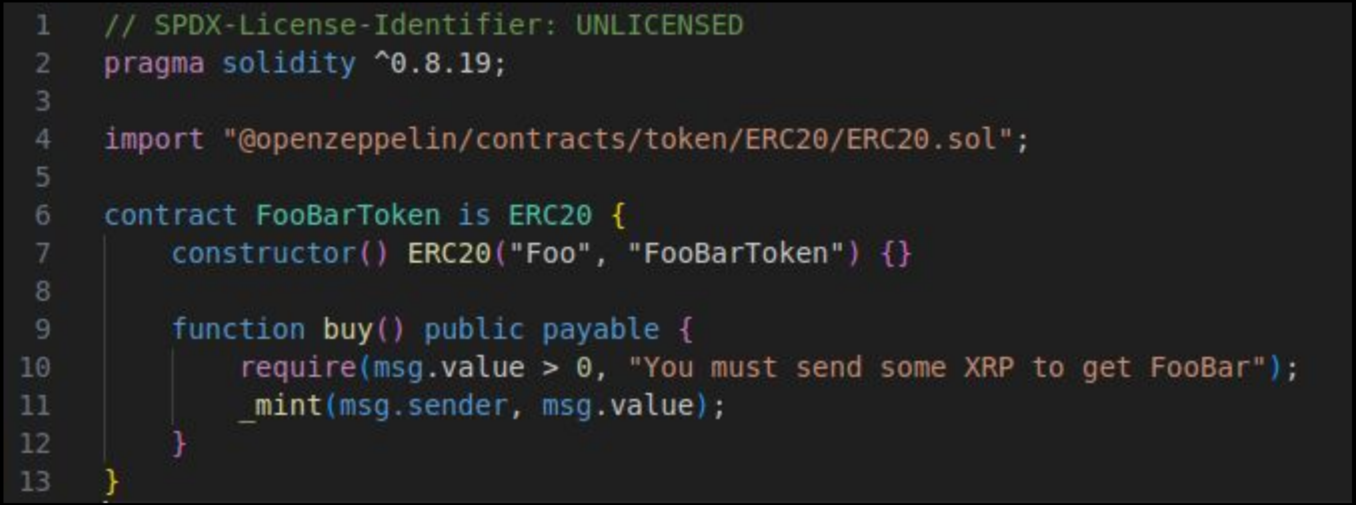


```
--- Before Transfer ---
-XRPL-: Address: rGsqMSTXbLxALbJ1oSJdRJMHjZpvwoizgR
-XRPL-: Balance: 100

-Sidechain-: Address: 0x242ea54ddb1559d58697399C0DD47645246ec656
-Sidechain-: Balance: 106.5289146425

Transfering 10 XRP from: locking to: issuing chain
XChain transaction success

--- After Transfer ---
-XRPL-: Address: rGsqMSTXbLxALbJ1oSJdRJMHjZpvwoizgR
-XRPL-: Balance: 89.999988

-Sidechain-: Address: 0x242ea54ddb1559d58697399C0DD47645246ec656
-Sidechain-: Balance: 111.4194281395
```

Mint ERC20

Create your own ERC20 token

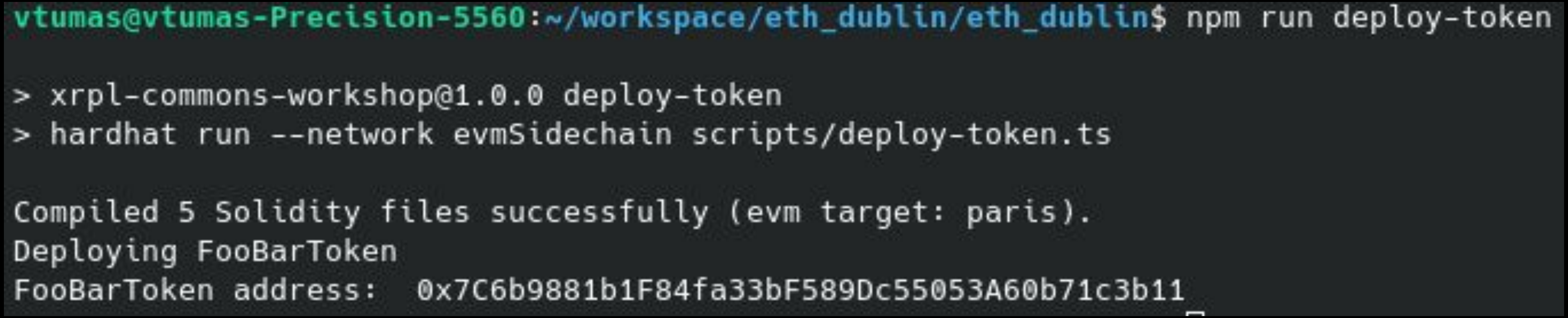
npm run deploy-token



```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.19;

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

contract FooBarToken is ERC20 {
    constructor() ERC20("Foo", "FooBarToken") {}

    function buy() public payable {
        require(msg.value > 0, "You must send some XRP to get FooBar");
        _mint(msg.sender, msg.value);
    }
}
```

Create your own ERC20 token

Keep the address



```
vtumas@vtumas-Precision-5560:~/workspace/eth_dublin/eth_dublin$ npm run deploy-token

> xrpl-commons-workshop@1.0.0 deploy-token
> hardhat run --network evmSidechain scripts/deploy-token.ts

Compiled 5 Solidity files successfully (evm target: paris).
Deploying FooBarToken
FooBarToken address:  0x7C6b9881b1F84fa33bF589Dc55053A60b71c3b11
```

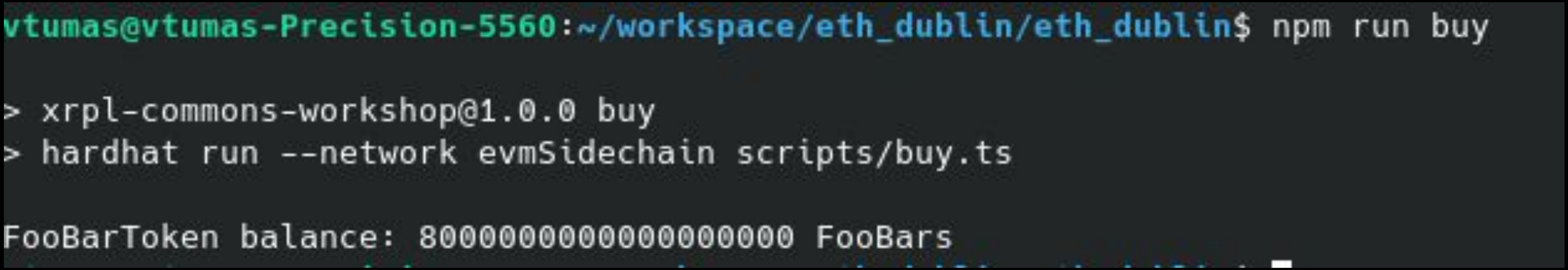
Buy FooBar

Buy FooBar

```
const main = async () => {
    const tokenFactory = await ethers.getContractFactory("FooBarToken");
    const tokenContract = tokenFactory.attach("<CONTRACT_ADDRESS>") as unknown;
    const fooBarContract = tokenContract as FooBarToken;

    const transaction = await fooBarContract.buy({
        value: ethers.parseEther("1"),
    });

    await transaction.wait(1);

    console.log("FooBarToken balance: " + await fooBarContract.balanceOf("<EVM_ACCOUNT_ADDRESS>" + " FooBars"));
}
```

Buy FooBar

npm run buy



```
vtumas@vtumas-Precision-5560:~/workspace/eth_dublin/eth_dublin$ npm run buy

> xrpl-commons-workshop@1.0.0 buy
> hardhat run --network evmSidechain scripts/buy.ts

FooBarToken balance: 800000000000000000000 FooBars
```