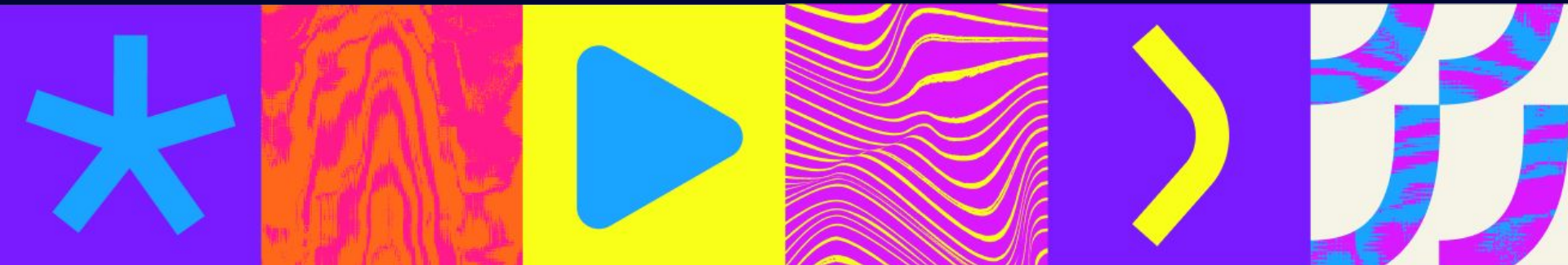
Commons

ECE
ÉCOLE D'INGÉNIEURS
ENGINEERING SCHOOL

# Workshop Introduction to the XRP Ledger



# Workshop I Introduction to the XRP Ledger

## Hack4Good Hackathon 2025

November 29th-30th 2025

XRPL Commons & ECE Paris

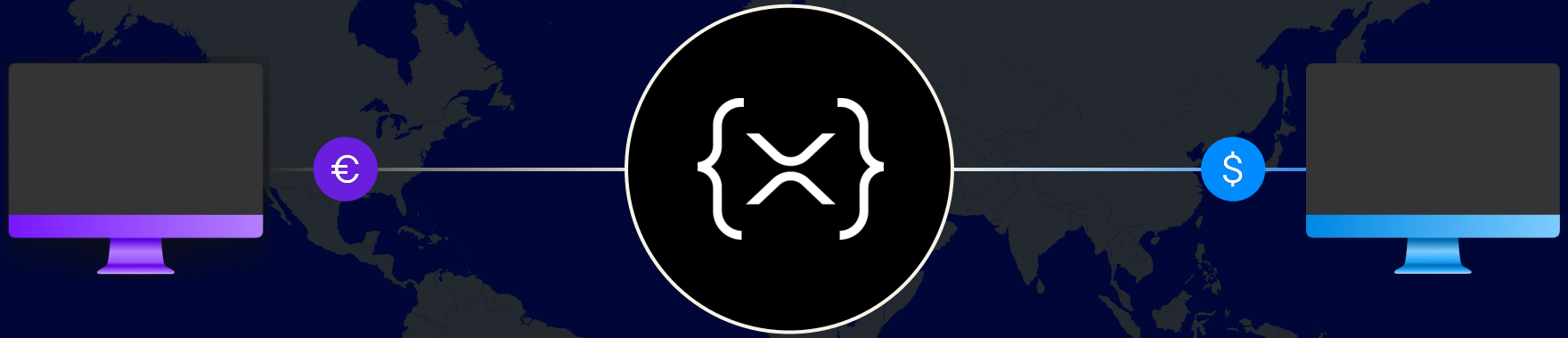
Commons | ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

# Agenda

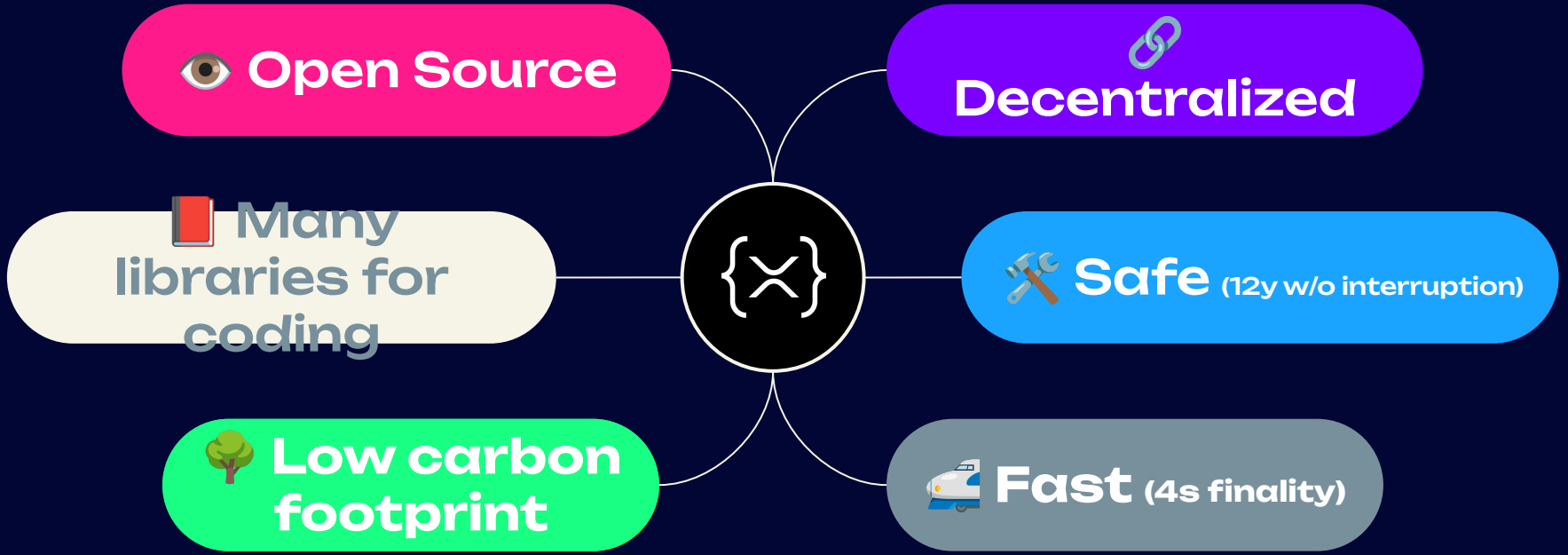
1. Introduction to the XRP Ledger
2. Interaction with XRPL
3. Introduction to Scaffold XRP
4. More documentation


# Introduction to the XRP Ledger

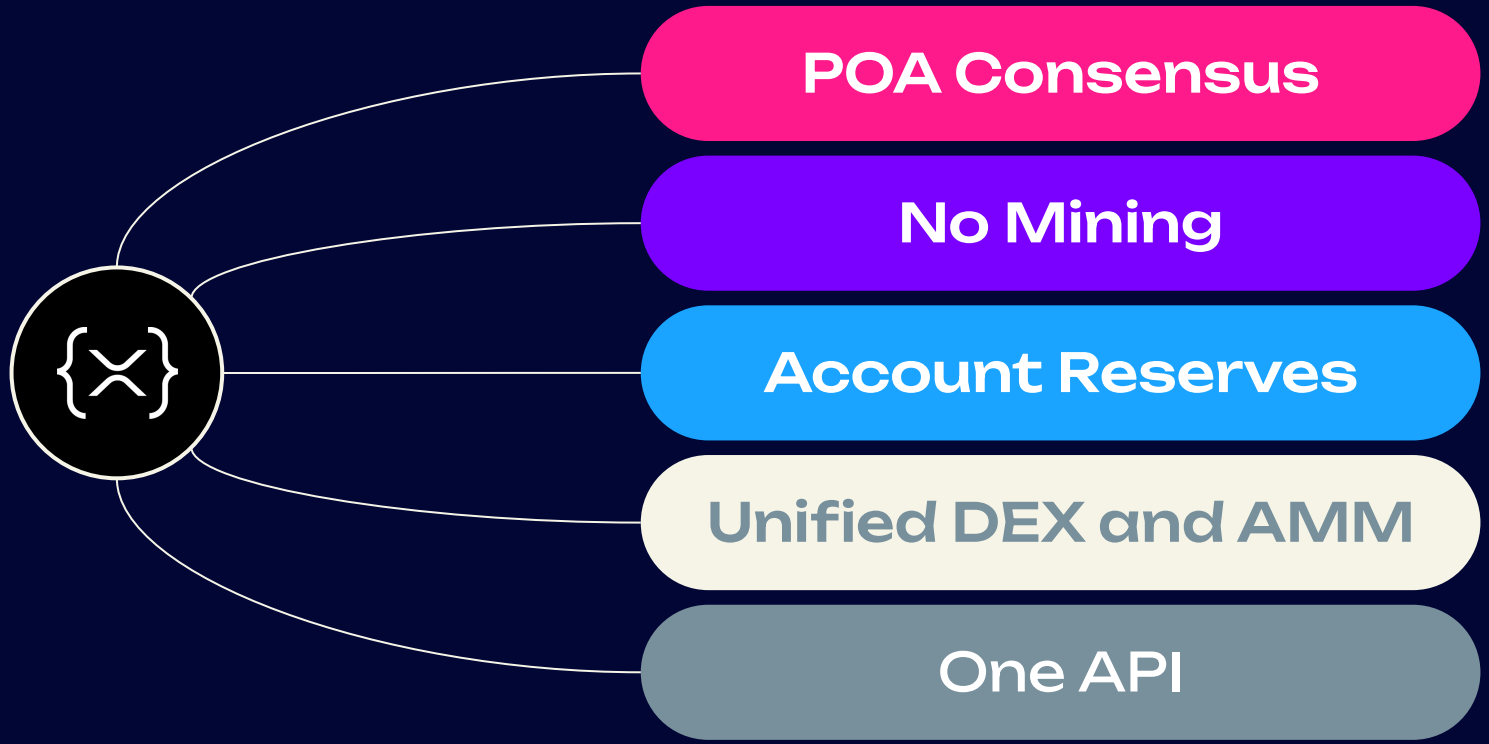
# XRP Ledger (XRPL) launched in 2012 to address limitations of crypto and fiat currencies for financial use cases, specifically payments



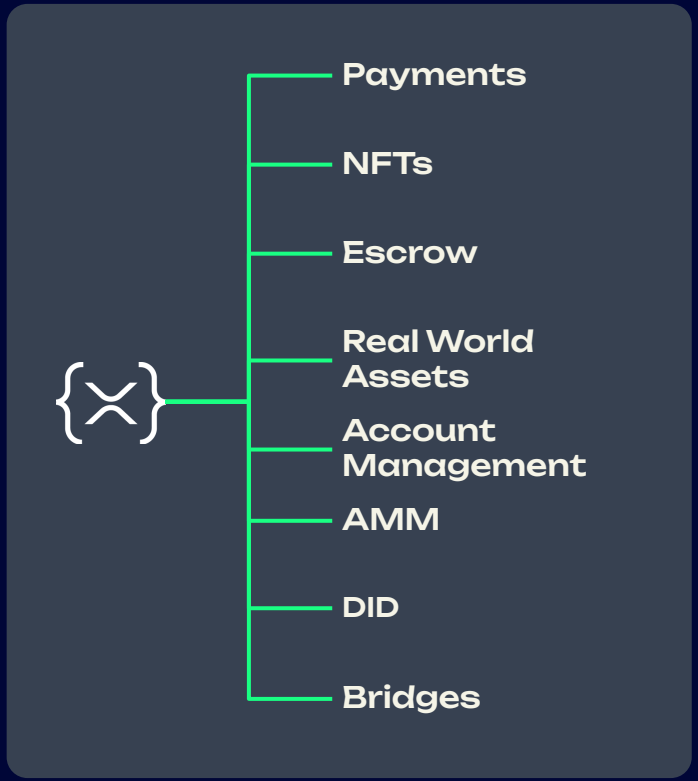
# The XRP Ledger



# Specificities of the XRPL



# One Endpoint to do all the things



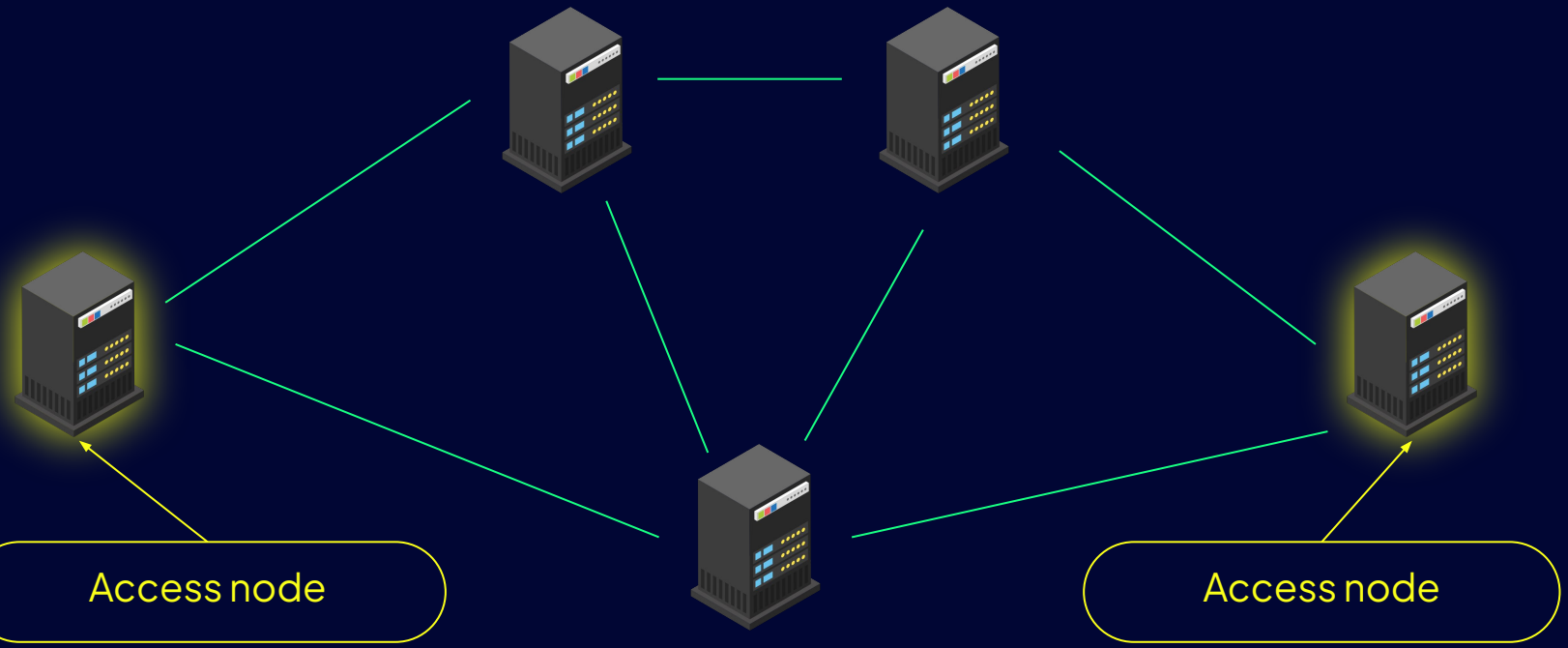

## Single API

No need to stitch together disparate systems or spend months integrating complex technology - simply connect into XRPL through a single API

## Minimal Code Required

Astonishingly simple, you can get up and running on the XRP Ledger in as little as few lines of code using familiar programming languages (JS, Python, Java, and many more)

# XRPL Network



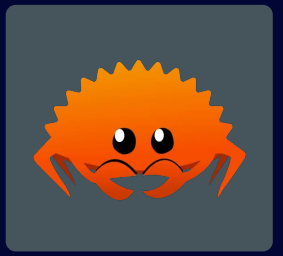

# Interaction with XRPL

# Languages with an XRPL SDK

Java


Typescript
Javascript

Python



Rust


Go

*Recommended: JavaScript/TypeScript or Python (most up to date)*

# Connect to a Node

```
import { Client } from "xrpl";

const client = new Client("wss://s.altnet.rippletest.net:51233");

async function main() {
  await client.connect();
  await client.disconnect();
}
```

# Create a Wallet

```
// Generate key pair and call the faucet
async function createWallet(client: xrpl.Client) {
  const { wallet, balance } = await client.fundWallet();
}
```

## 2. Interaction with XRPL

# Prepare & Send the Transaction

```
async function sendPaymentTx(
  client: xrpl.Client,
  wallet: xrpl.Wallet,
  address: string,
  amount: number,
) {
  const tx: xrpl.Payment = {
    TransactionType: "Payment",
    Account: wallet.classicAddress,
    Destination: address,
    Amount: xrpl.xrpToDrops(amount),
  };

  return await client.submitAndWait(tx, {
    autofill: true,
    wallet,
  });
}
```

# Get the Result




```
{
  api_version: 2,
  id: 22,
  result: {
    close_time_iso: "2024-12-06T13:39:10Z",
    ctid: "C02C0E9E00010001",
    hash: "A2D8910A45D19A91755F3BBC1E29F5FC97C438B0E117E3FBD042DAD1C9A94B98",
    ledger_hash: "8DAABD0B422F691F3E806EF0E78B8D3F5D7F7D831E97661754A6E31B5E7CEA7C",
    ledger_index: 2887326,
    meta: {
      AffectedNodes: [ { ModifiedNode: [Object] }, { CreatedNode: [Object] } ],
      TransactionIndex: 1,
      TransactionResult: "tesSUCCESS",
      delivered_amount: "10000000"
    },
    tx_json: {
      Account: "rELCjuVzgjBrBUexXoytR59gzEMTU3BwBc",
      DeliverMax: "10000000",
      Destination: "rf2ogfMAfxaSK8cRJ8b3HpSn11N4v5xD9h",
      Fee: "12",
      Flags: 0,
      LastLedgerSequence: 2887344,
      Sequence: 2887324,
      SigningPubKey: "ED598042DFC6F9C65B16002F042B57AA4372EC6B12B9F3862F772A4FCF7B331BBC",
      TransactionType: "Payment",
      TxnSignature: "4ABCFD76572234A1B51BE6509C6364835530EE44C2B59005D4A5ACE4C84B13F9EF0025CF84A5C1F922F500877E2BCFA
0C6696A434024FD2E3A776E0F7B485C07",
      date: 786807550,
      ledger_index: 2887326
    },
    validated: true
  },
  type: "response"
}
```

## 2. Interaction with XRPL

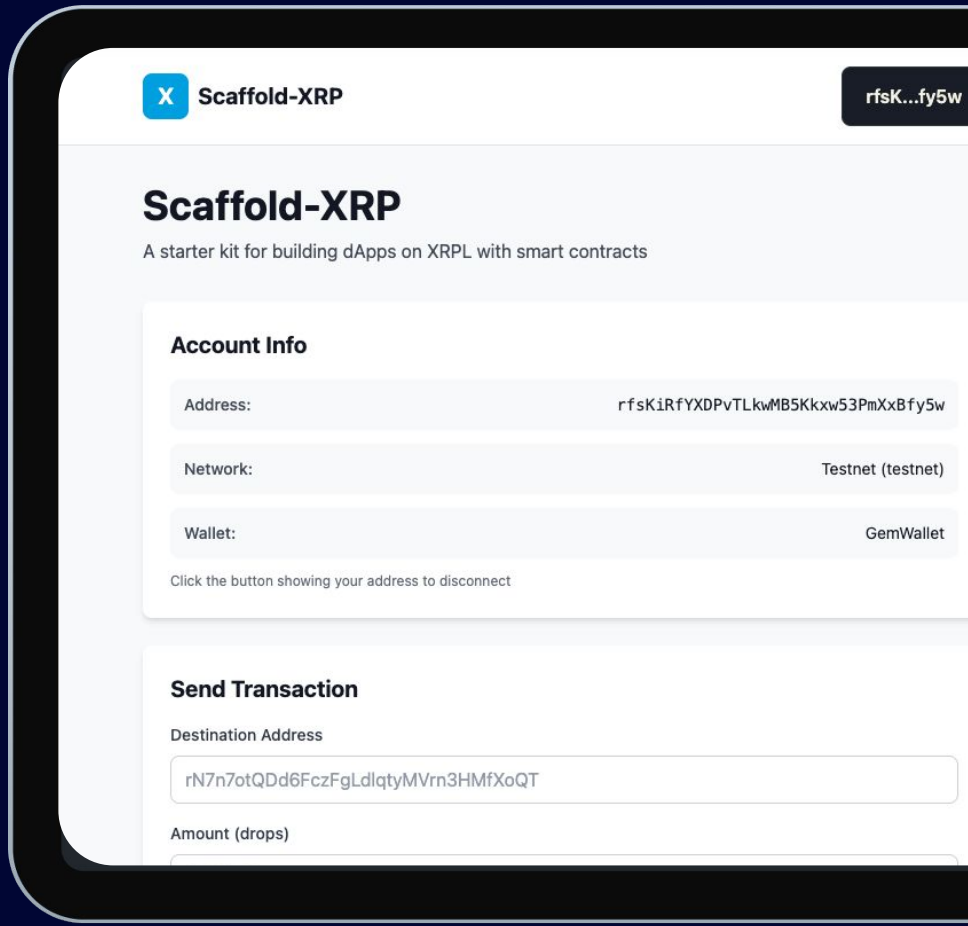
# The Entire Code

```
import xrpl from "xrpl";

const client = new xrpl.Client("wss://s.altnet.rippletest.net:51233");

async function main() {
  // 1. we connect to a node
  await client.connect();

  // 2. we create a wallet
  const wallet = await createWallet(client);

  // 3. we prepare and send the tx
  // Here we want to send 10 XRP to rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h
  const amount = 10;
  const tx = await sendPaymentTx(
    client,
    wallet,
    "rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h",
    amount,
  );

  //4. we get the result
  console.log(tx);
  await client.disconnect();
}
```


# Introduction to Scaffold XRP

# Scaffold XRP

https://xrpl.at/ece2025/scaffold



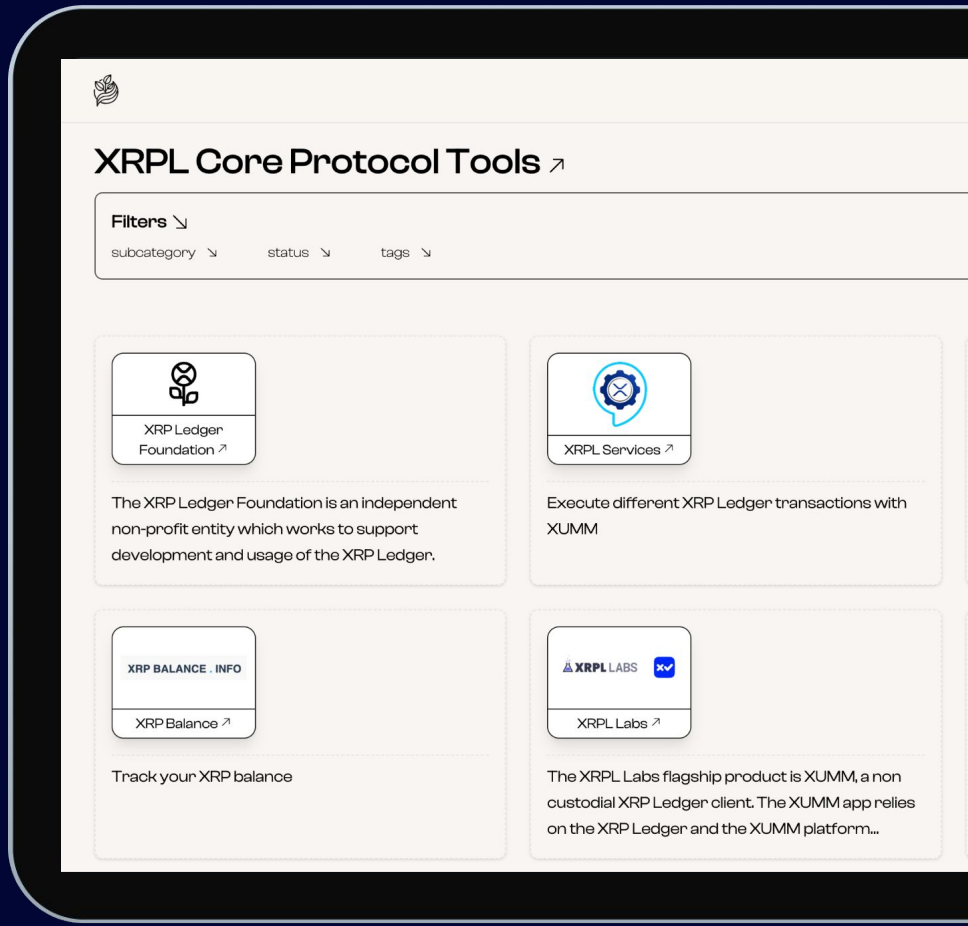
# Live Demo!


# More documentation

# 4. More documentation

## Ecosystem Map

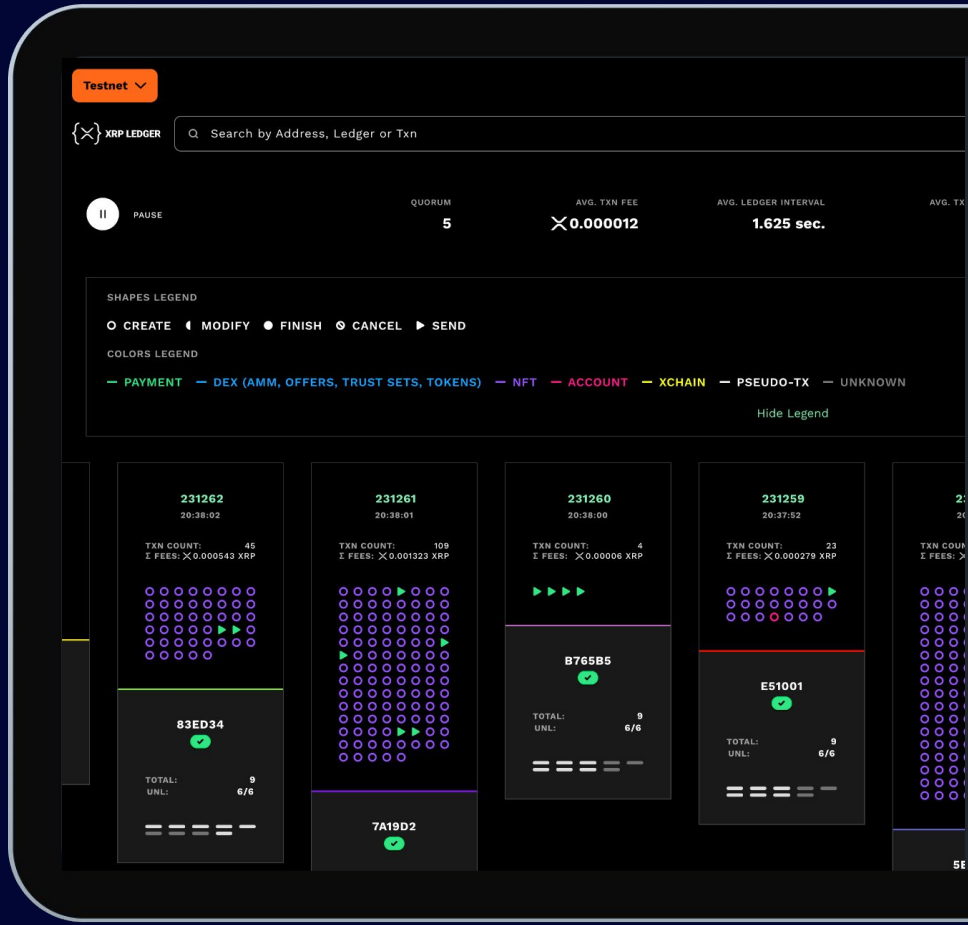
https://map.xrpl-commons.org



# 4. More documentation

## Block Explorer

https://testnet.xrpl.org

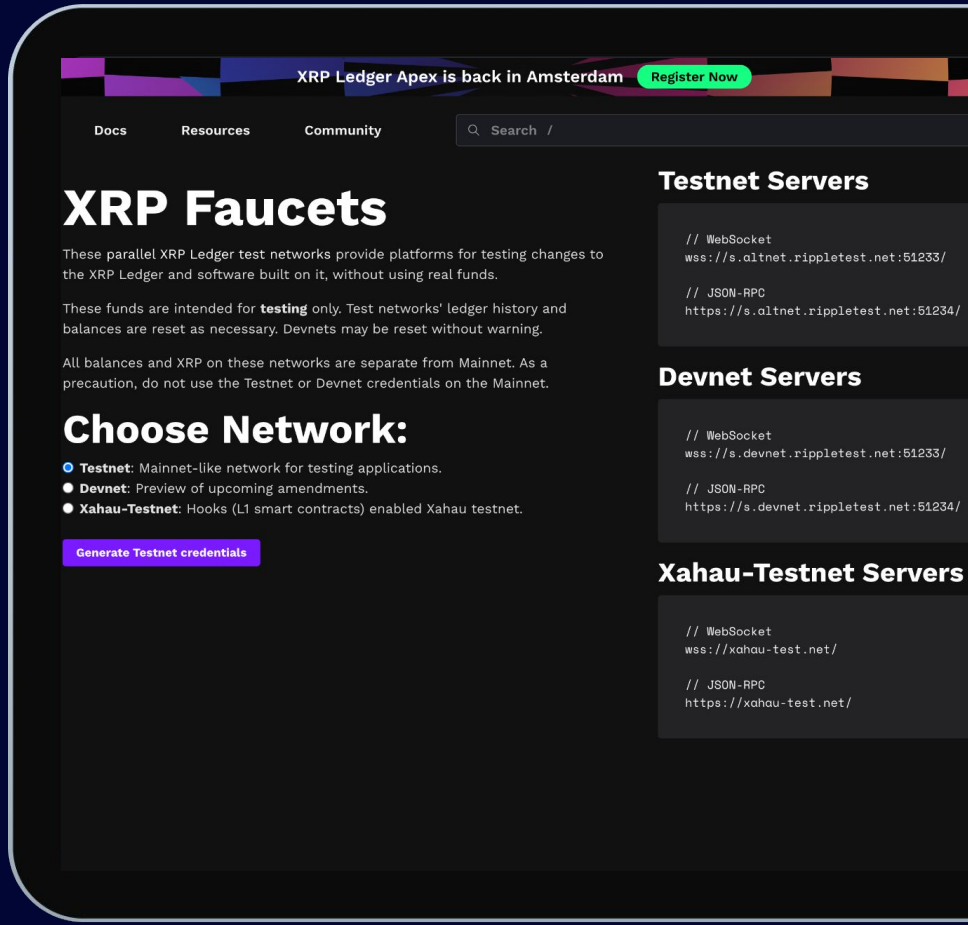



Commons

# 4. More documentation

## Faucets

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

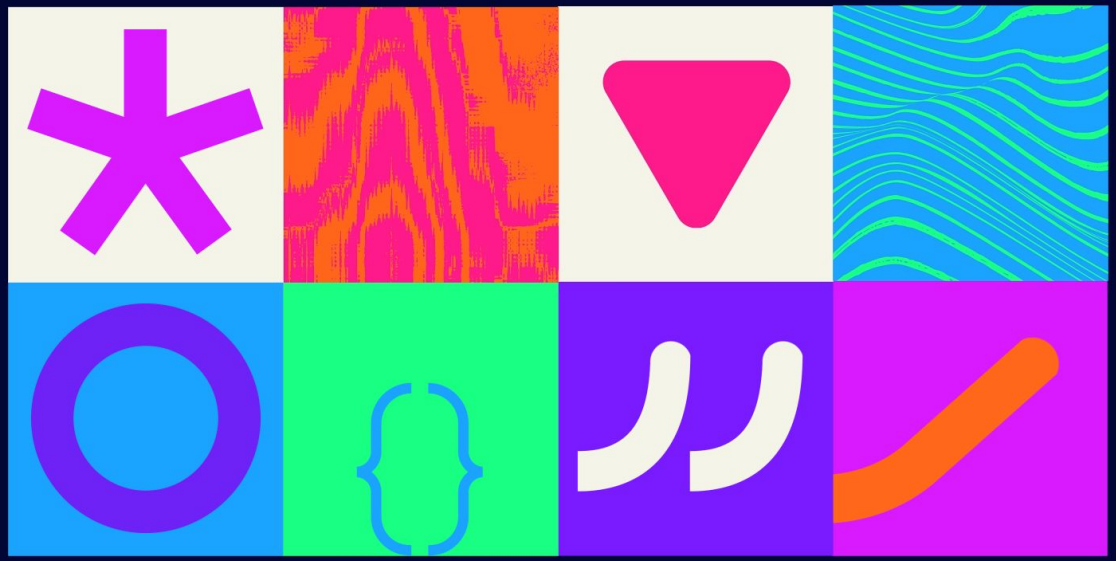


# 4. More documentation

## Hackathon GitHub

http://xrpl.at/ece2025

# Ask me anything!



commons

# Thank you!