🦺 Pour cette session, **Veuillez installer:**

xaman.app

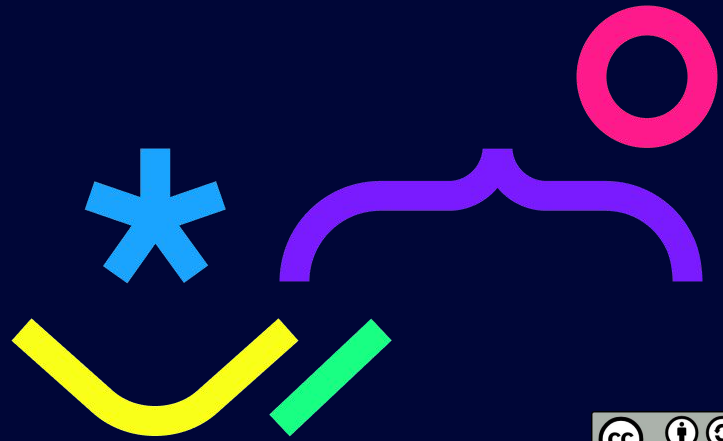
github.com

Commons

Introduction à la blockchain

Kryptosphere DeVinci · Student Cohort 2025

19 Septembre 2025



Florian Alonso, Product Manager @ XRPL Commons
Mathis Sergent, Research Engineer @ XRPL Commons

You are holding open source content.

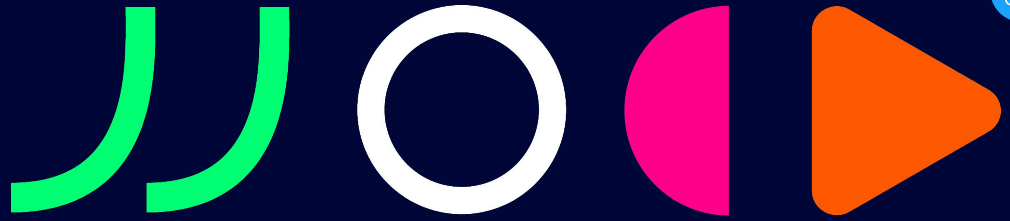
Here's how to handle it:

→ You can use this content in your work, adapt, and share it.

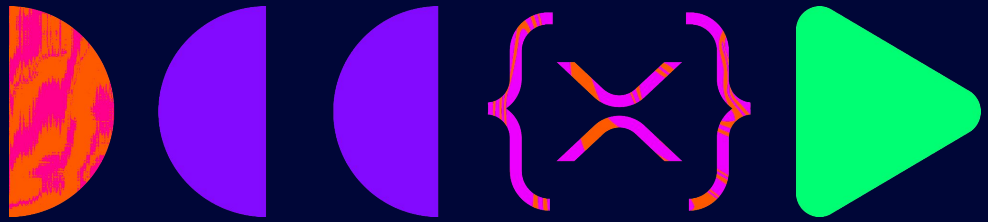
→ You must mention XRPL Commons and the Creative Commons license as a source.

Premiers pas sur la blockchain

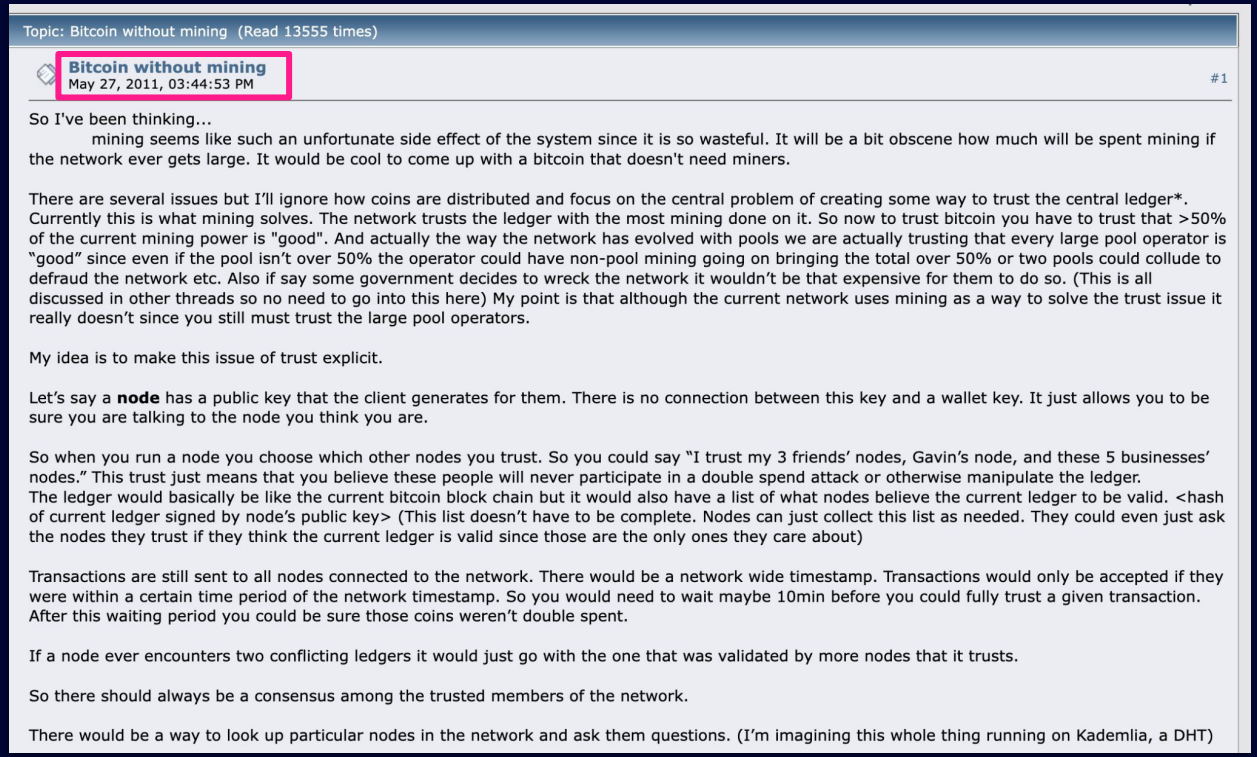
- 💙 Présentation d'XRPL Commons
- 🌐 Bases de la blockchain
- {X} Introduction au XRP Ledger
- 🪪 Créer mon wallet
- ↔ Exécuter un transfert
- 💸 Créer un token
- 💱 Acheter & Vendre un token



Introduction au XRP Ledger {⨯}



Les limitations de Bitcoin



Topic: Bitcoin without mining (Read 13555 times)

Bitcoin without mining
May 27, 2011, 03:44:53 PM

#1

So I've been thinking...
mining seems like such an unfortunate side effect of the system since it is so wasteful. It will be a bit obscene how much will be spent mining if the network ever gets large. It would be cool to come up with a bitcoin that doesn't need miners.

There are several issues but I'll ignore how coins are distributed and focus on the central problem of creating some way to trust the central ledger*. Currently this is what mining solves. The network trusts the ledger with the most mining done on it. So now to trust bitcoin you have to trust that >50% of the current mining power is "good". And actually the way the network has evolved with pools we are actually trusting that every large pool operator is "good" since even if the pool isn't over 50% the operator could have non-pool mining going on bringing the total over 50% or two pools could collude to defraud the network etc. Also if say some government decides to wreck the network it wouldn't be that expensive for them to do so. (This is all discussed in other threads so no need to go into this here) My point is that although the current network uses mining as a way to solve the trust issue it really doesn't since you still must trust the large pool operators.

My idea is to make this issue of trust explicit.

Let's say a **node** has a public key that the client generates for them. There is no connection between this key and a wallet key. It just allows you to be sure you are talking to the node you think you are.

So when you run a node you choose which other nodes you trust. So you could say "I trust my 3 friends' nodes, Gavin's node, and these 5 businesses' nodes." This trust just means that you believe these people will never participate in a double spend attack or otherwise manipulate the ledger.
The ledger would basically be like the current bitcoin block chain but it would also have a list of what nodes believe the current ledger to be valid. <hash of current ledger signed by node's public key> (This list doesn't have to be complete. Nodes can just collect this list as needed. They could even just ask the nodes they trust if they think the current ledger is valid since those are the only ones they care about)

Transactions are still sent to all nodes connected to the network. There would be a network wide timestamp. Transactions would only be accepted if they were within a certain time period of the network timestamp. So you would need to wait maybe 10min before you could fully trust a given transaction. After this waiting period you could be sure those coins weren't double spent.

If a node ever encounters two conflicting ledgers it would just go with the one that was validated by more nodes that it trusts.

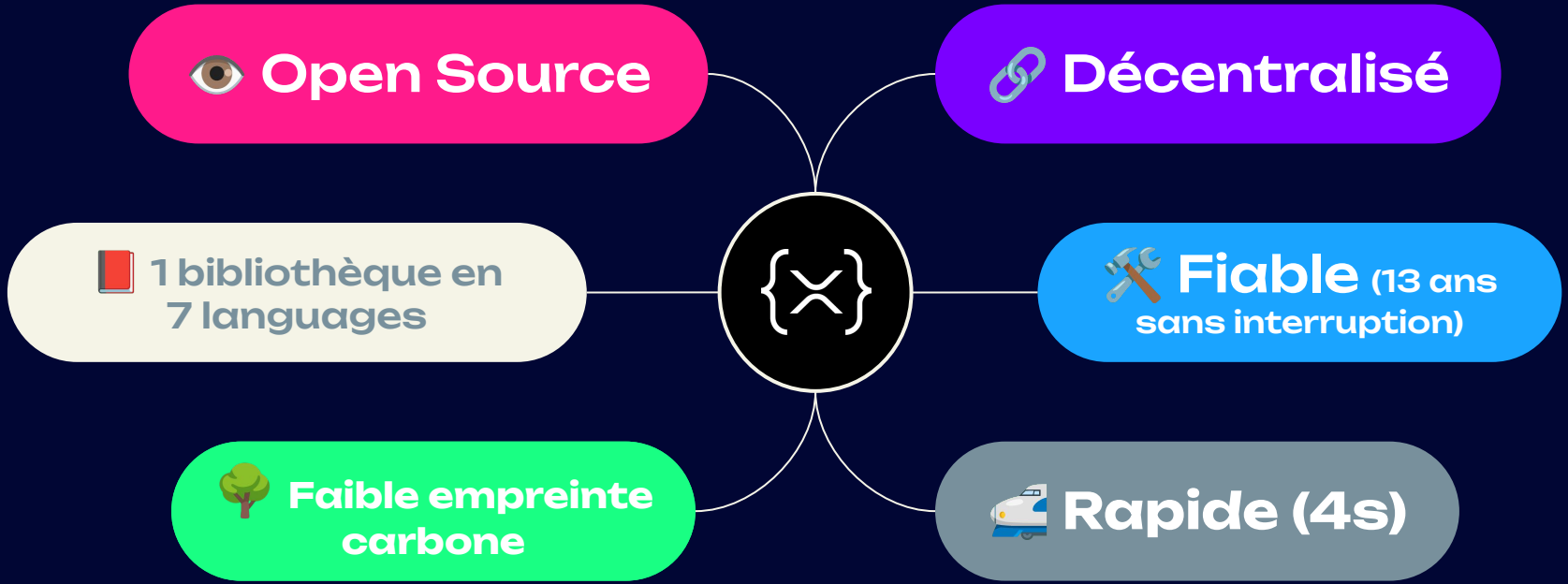
So there should always be a consensus among the trusted members of the network.

There would be a way to look up particular nodes in the network and ask them questions. (I'm imagining this whole thing running on Kademlia, a DHT)

Source: https://bitcointalk.org/index.php?topic=10193.0

En 2012, le XRP Ledger (XRPL) est lancé pour pallier aux limitations des monnaies fiat et des crypto-monnaies dans les paiements.

XRP Ledger



En chiffres, le XRPL est devenu l'une des blockchains layer-1 les plus robustes.

100%

décentralisée avec plus de 600 nœuds traitant les transactions et maintenant le réseau.

1,750+

applications sur le mainnet développés par une diversité de développeurs du monde entier.

5M+

wallet XRP actifs à travers le monde.

185+

validateurs Proof-of-Association gérés par des universités, des exchanges, des entreprises et des individus.

2.8B+

transactions traitées représentant plus de 1 trillion de dollars de valeur échangée entre contreparties.

~$125B+

capitalisation boursière de XRP, en faisant la troisième plus grande cryptomonnaie.

Source: CoinMarketCap, Mai 2025 https://livenet.xrpl.org/

Les différences entre le XRP Ledger, XRP et Ripple

XRP LEDGER

Blockchain de Layer-1

Le XRP Ledger est une blockchain publique décentralisée et open-source, portée par une communauté mondiale de développeurs.

Utilisé dans le monde entier pour la tokenisation, les paiements, les stablecoins, les MNBC (CBDCs) et plus encore.

XRP

Actif numérique natif

XRP est l'actif numérique natif du XRPL, comme l'ETH pour Ethereum ou le SOL pour Solana.

Il sert principalement à faciliter les transactions, à protéger le réseau contre le spam et à faire le lien entre devises sur le DEX du XRP Ledger.

ripple

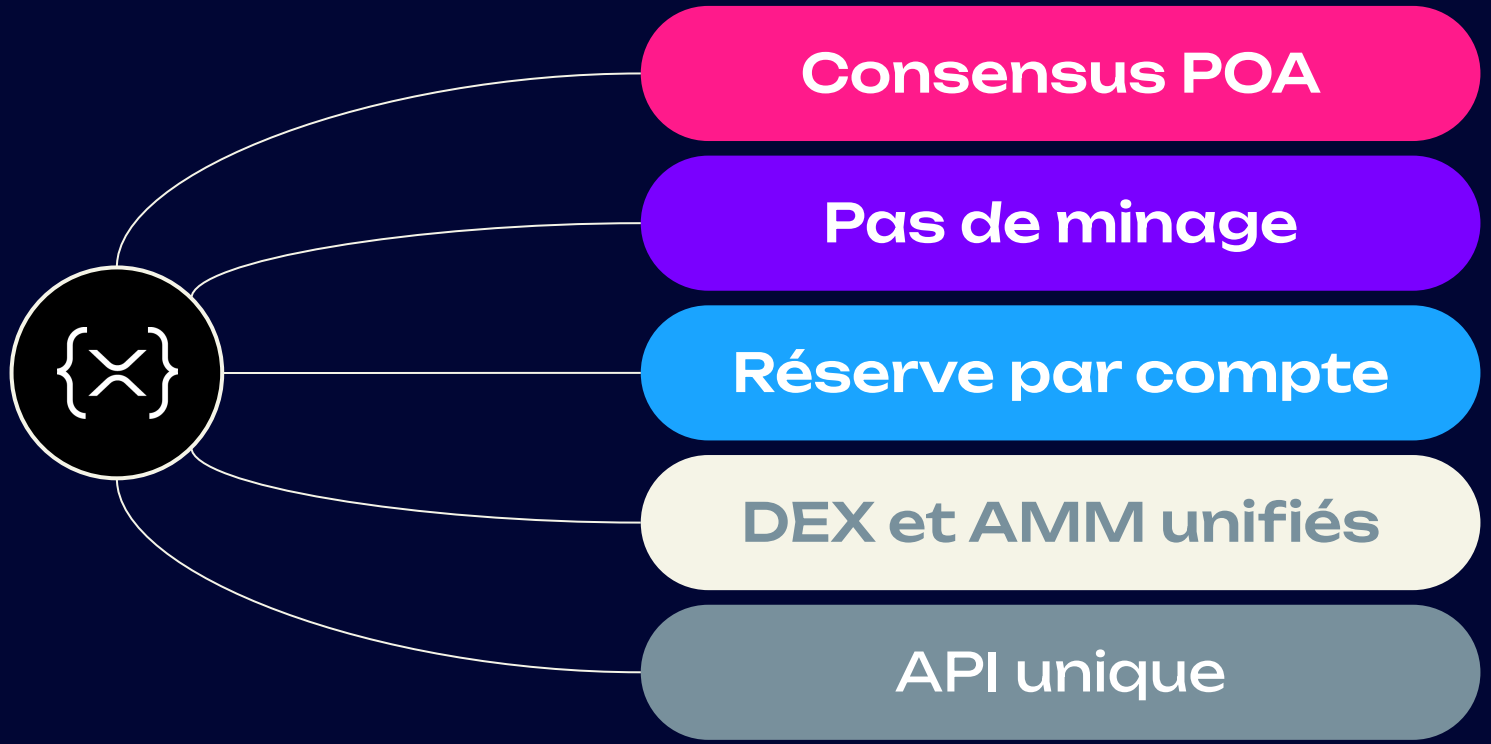
Entreprise crypto

Ripple est une entreprise tech qui développe des solutions crypto pour les entreprises, en utilisant le XRP comme monnaie de pont pour les paiements grâce à sa rapidité et sa fiabilité.

Elle fait partie des nombreux acteurs construisant sur le XRP Ledger.

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

Spécificités du XRPL



Consensus du XRP Ledger

Le consensus du XRPL est un type de **Proof of Association**

Confiance explicite.
Des centaines de nœuds validateurs participent au consensus. **35 nœuds spéciaux** figurent sur la UNL, qui liste les nœuds ayant le dernier mot. 80 % de la UNL doivent être d'accord pour valider un bloc.

Décentralisé
Aucune entité ne peut contrôler plus de 5 % de la UNL. Chaque membre de la UNL est une entité connue avec une transparence totale.

Gouvernance indépendante
La fondation XRPL veille à ce que les membres de la UNL respectent des directives strictes en matière de mises à jour et de disponibilité. Les membres de la UNL sont régulièrement audités et peuvent changer au fil du temps.

Un seul point d'entrée pour toutes les fonctionnalités

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)

Les contributeurs développent des fonctionnalités institutionnelles pour étendre les cas d'usage sur le XRPL

● Amendment voté

Identité décentralisée (DID)

Standard interopérable pour la gestion d'identité, l'authentification et la gestion des accès.

● Amendment voté

Sidechain XRPL

Personnalisez une version du XRPL pour répondre aux besoins d'une entreprise tout en restant interopérable.

● Amendment voté

EVM Sidechain

Apporter une programmabilité complète au XRPL dans des environnements EVM familiers.

● Amendment voté

Transactions unifiées

Regrouper les transactions pour une exécution totale ou nulle.

● In Development

Token multi-usages (MPT)

Jetons natifs polyvalents et programmable pour représenter différents types d'actifs.

● In Development

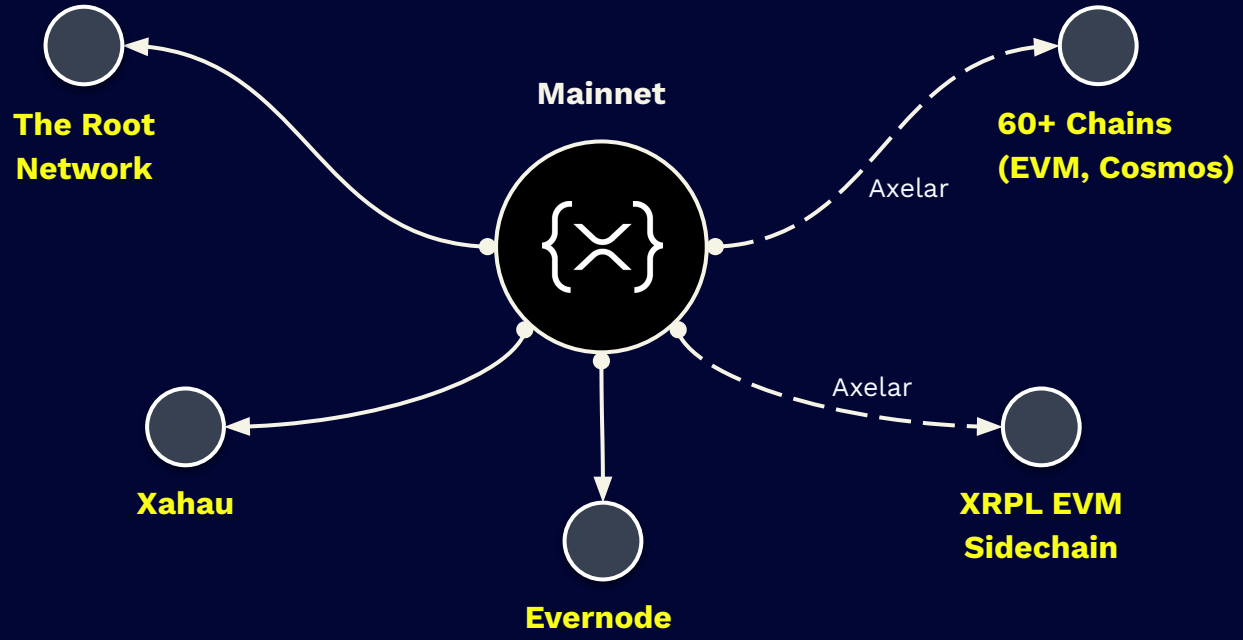
Prêt / Emprunt.

Fonctionnalité de prêt et d'emprunt native au protocole sans avoir besoin de contrats intelligents.

Source: Ripple Open Source Projects, https://hooks.xrpl.org/

L'écosystème XRP Ledger

Le réseau principal XRPL interagit avec les sidechains.



XRPL EVM Sidechain

Pourquoi une sidechain EVM ?

01

Smart Contracts

Les contrats intelligents polyvalents ouvrent notre écosystème à l'innovation et renforcent l'interopérabilité avec l'écosystème blockchain plus large.

02

Ouverture aux développeurs Solidity

La sidechain EVM est un moyen d'intégrer la grande communauté des développeurs EVM dans l' écosystème XRPL.

Pourquoi est-ce intéressant ?

Les applications EVM peuvent désormais bénéficier de l'écosystème XRPL

01

Pont vers l’écosystème XRPL.

Toute application Solidity écrite pour Ethereum / EVM peut accéder à la liquidité et à la base d’utilisateurs du XRPL Mainnet.

02

Optimisé pour la DeFi

Des ponts sécurisés, une scalabilité améliorée et une finalité rapide des transactions rendent l’EVM optimisé pour les cas d’usage financiers, comme la DeFi et les paiements.

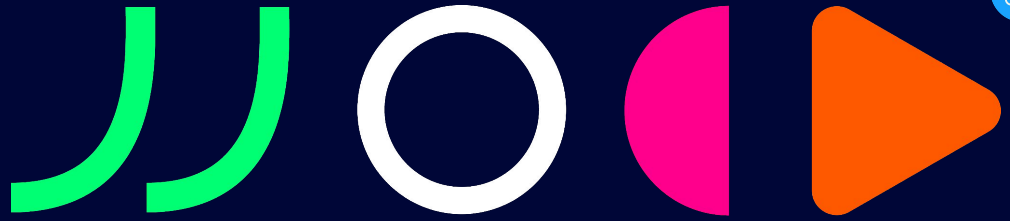
03

Facile à construire

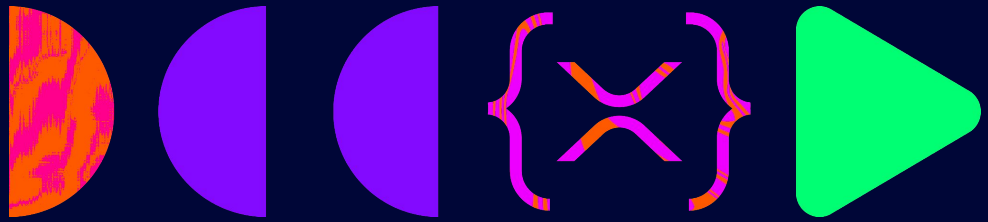
Construisez avec des outils, portefeuilles, explorateurs et applications basés sur Ethereum, comme MetaMask, Foundry et Truffle.

Challenge ta compréhension 🧠

Quiz



Wallets & Transactions 💳

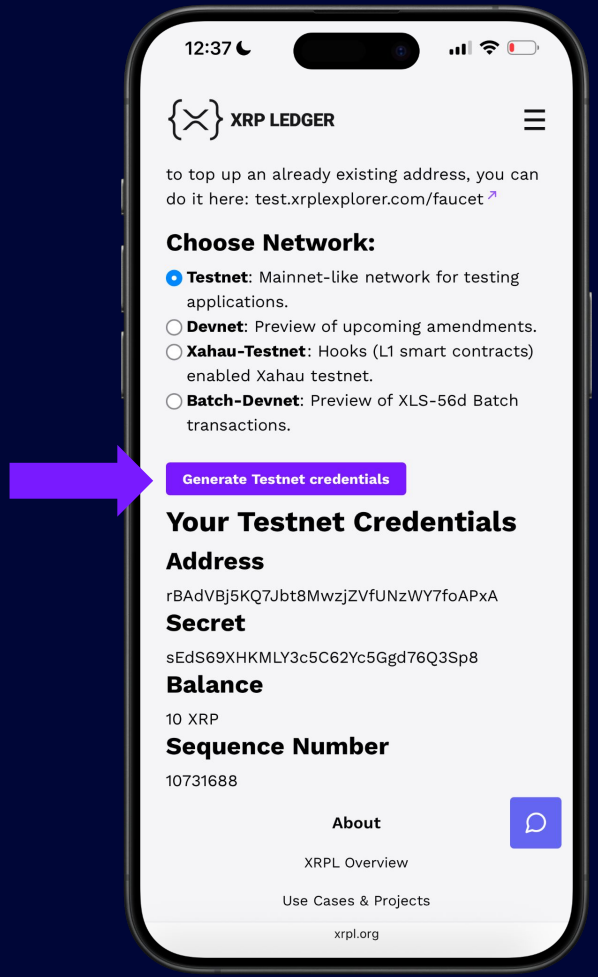


Levez la main si vous avez déjà un wallet.

Commons

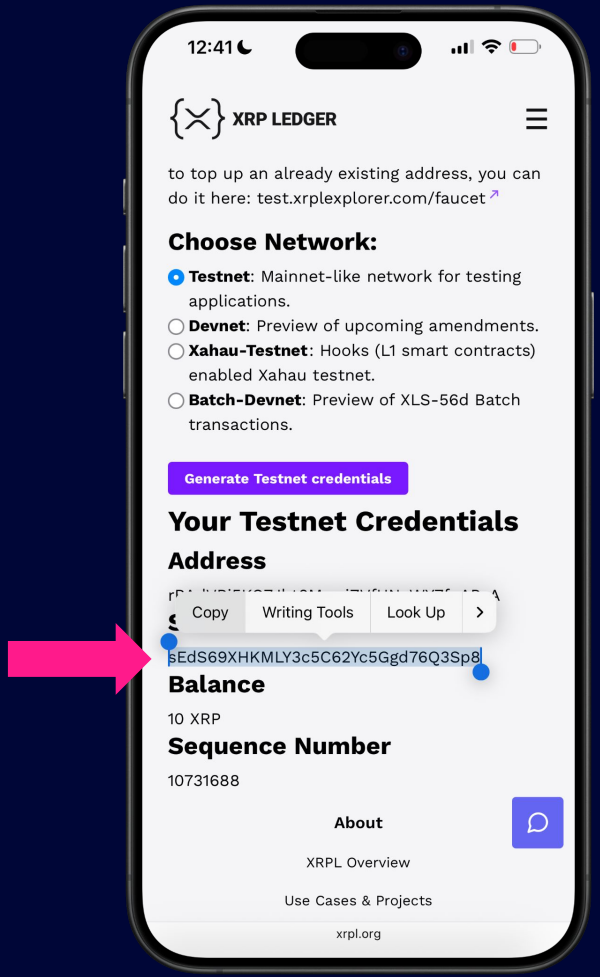
Ma première transaction 📲

À vos smartphones !

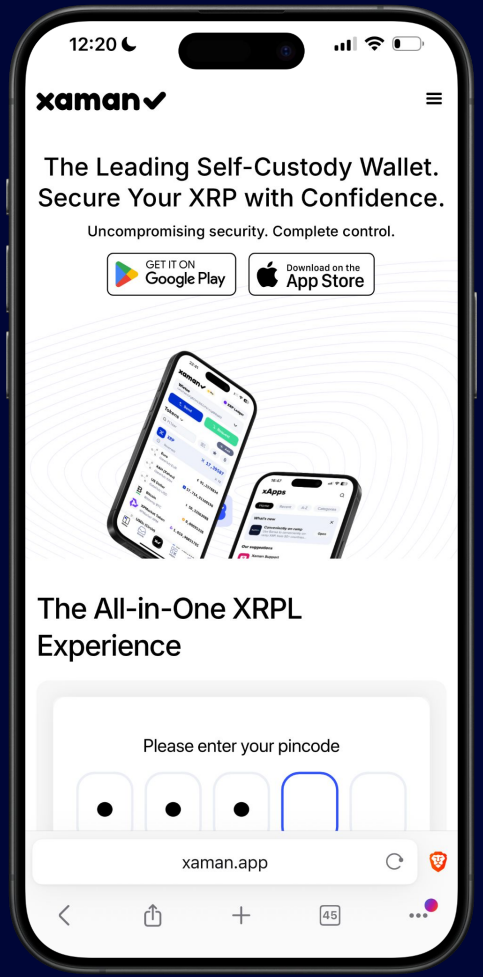


1. Créer un wallet

XRP Faucets

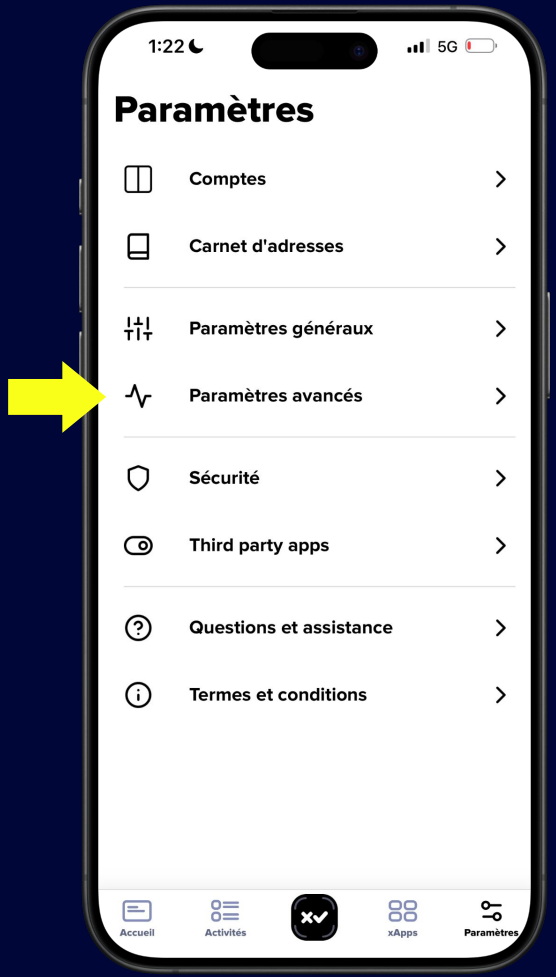


2. Copier la clé privée

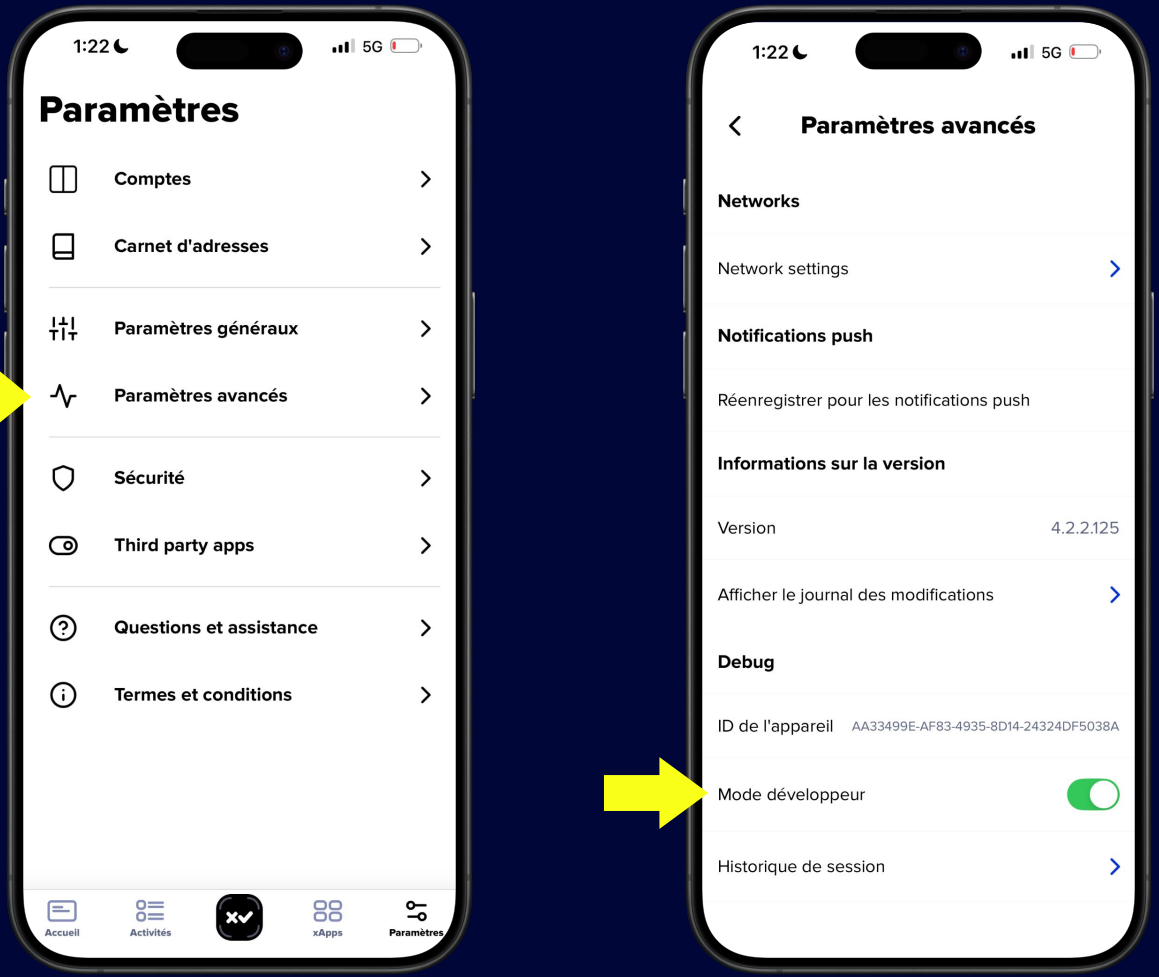


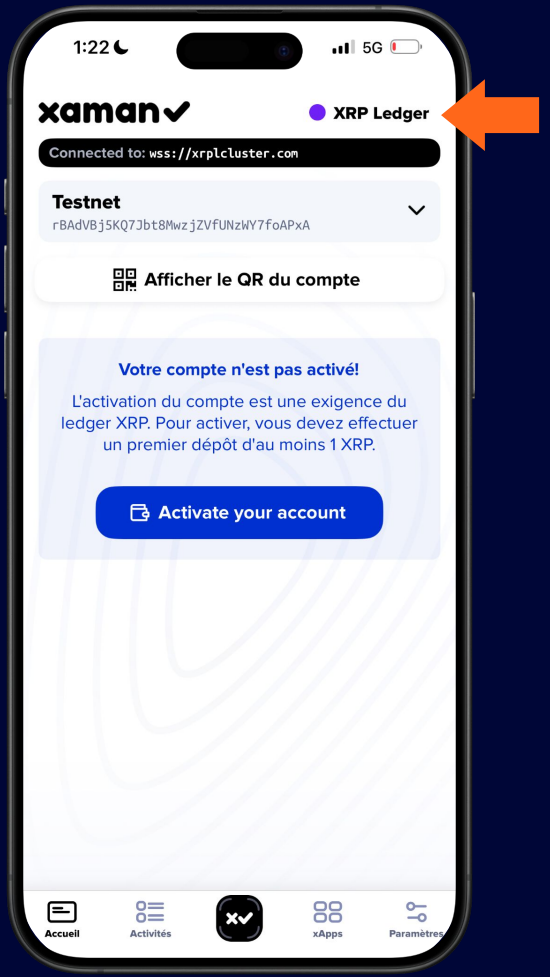
3. Télécharger Xaman

xaman.app

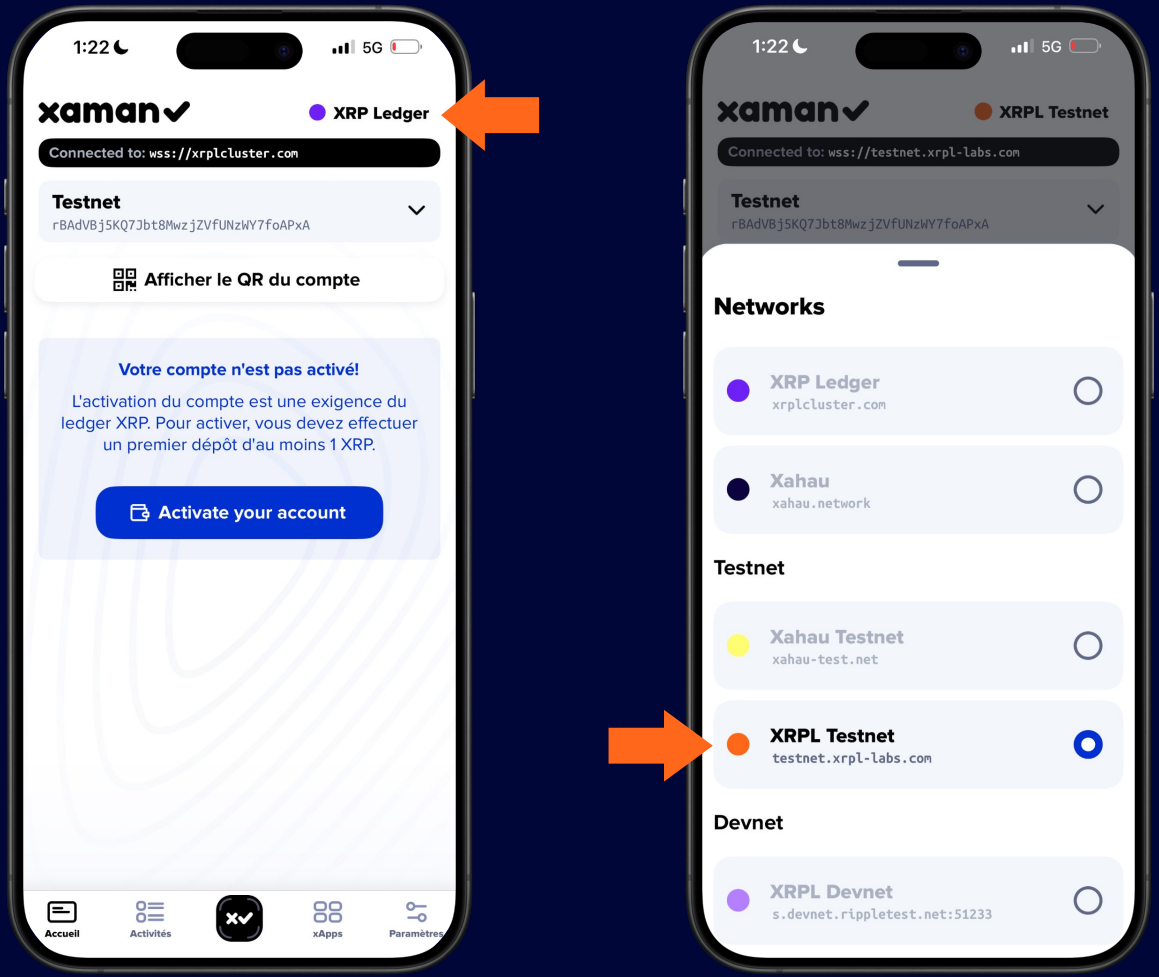


4. Activer le mode développeur


1:22
5G
Paramètres
Comptes
Carnet d'adresses
Paramètres généraux
Paramètres avancés
Sécurité
Third party apps
Questions et assistance
Termes et conditions
Accueil
Activités
xApps
Paramètres
1:22
5G
Paramètres avancés
Networks
Network settings
Notifications push
Réenregistrer pour les notifications push
Informations sur la version
Version
4.2.2.125
Afficher le journal des modifications
Debug
ID de l'appareil
AA33499E-AF83-4935-8D14-24324DF5038A
Mode développeur
Historique de session



5. Configurer le réseau Testnet

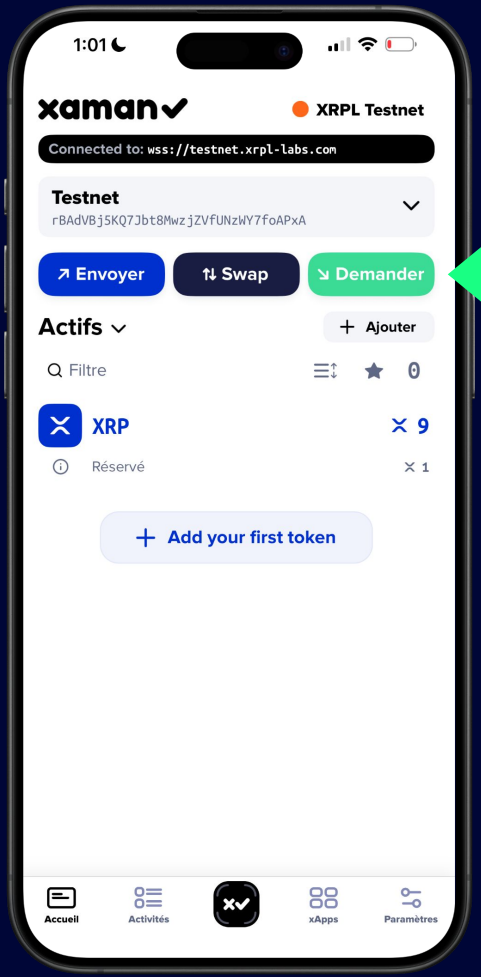

1:22
5G
xaman
XRP Ledger
Connected to: wss://xrplcluster.com
Testnet
rBAdVBj5KQ7Jbt8MwzjZVfUNzWY7foAPxA
Afficher le QR du compte
Votre compte n'est pas activé!
L'activation du compte est une exigence du ledger XRP. Pour activer, vous devez effectuer un premier dépôt d'au moins 1 XRP.
Activate your account
Accueil
Activités
xApps
Paramètres
1:22
5G
xaman
XRPL Testnet
Connected to: wss://testnet.xrpl-labs.com
Testnet
rBAdVBj5KQ7Jbt8MwzjZVfUNzWY7foAPxA
Networks
XRP Ledger
xrplcluster.com
Xahau
xahau.network
Testnet
Xahau Testnet
xahau-test.net
XRPL Testnet
testnet.xrpl-labs.com
Devnet
XRPL Devnet
s.devnet.rippletest.net:51233

6. Ajouter votre compte

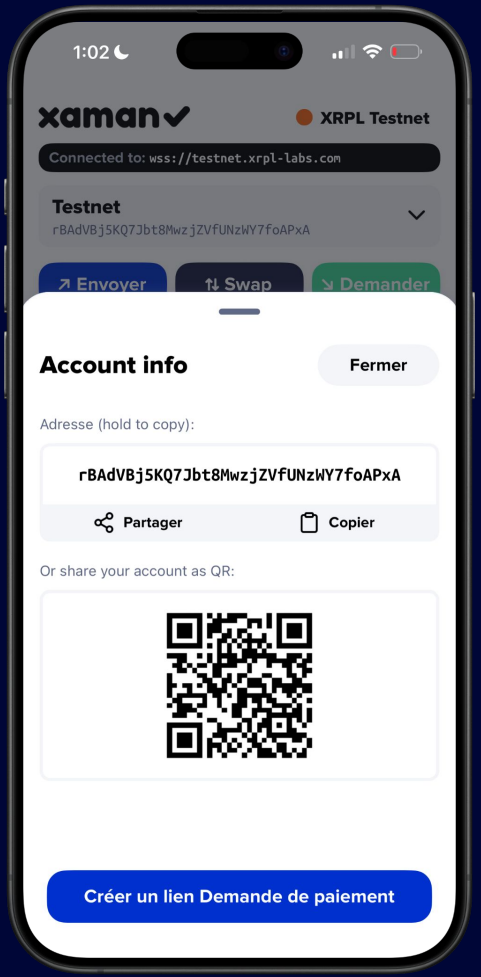

12:09
Ajouter un compte
Importer votre compte existant, ou créer un nouveau compte.
+ Créer un nouveau compte
Importer un compte existant
OU
Ajouter une carte Tangem


12:44
Type de compte
Veuillez sélectionner votre type de compte
Accès complet
Envoyer, signer et recevoir des transactions
Lecture seule
Voir le solde & recevoir des transactions
Vous pourrez par la suite et à tout moment mettre à jour votre compte en lecture seule à un accès complet
Retourner
Suivant

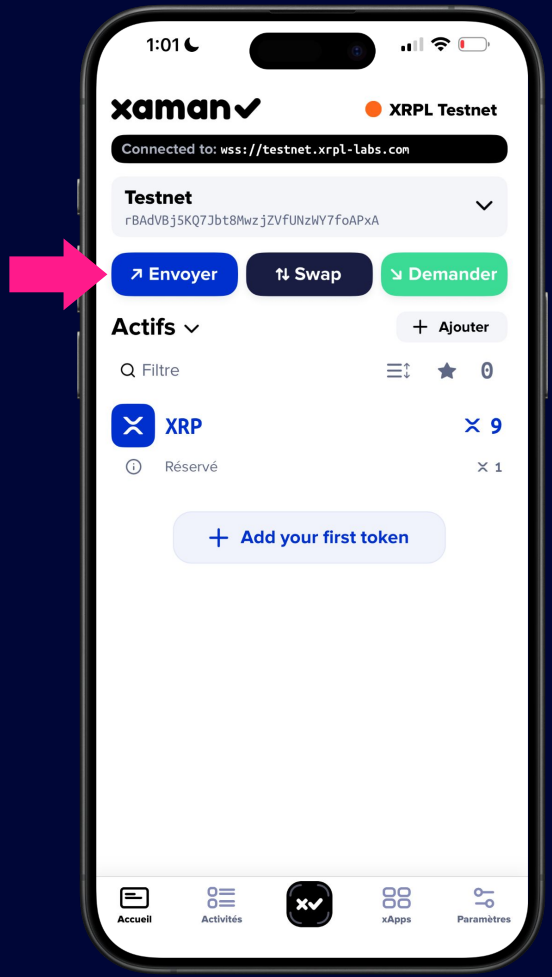

12:46
Type de secret
Veuillez sélectionner votre type de clé secrète
Numéros secrets
Type de compte préféré. 8 lignes de 6 chiffres
Graine familiale
Ressemble à sXXX1234XXX...
Mnémonique
Basé sur 12, 16 ou 24 mots
Retourner
Suivant



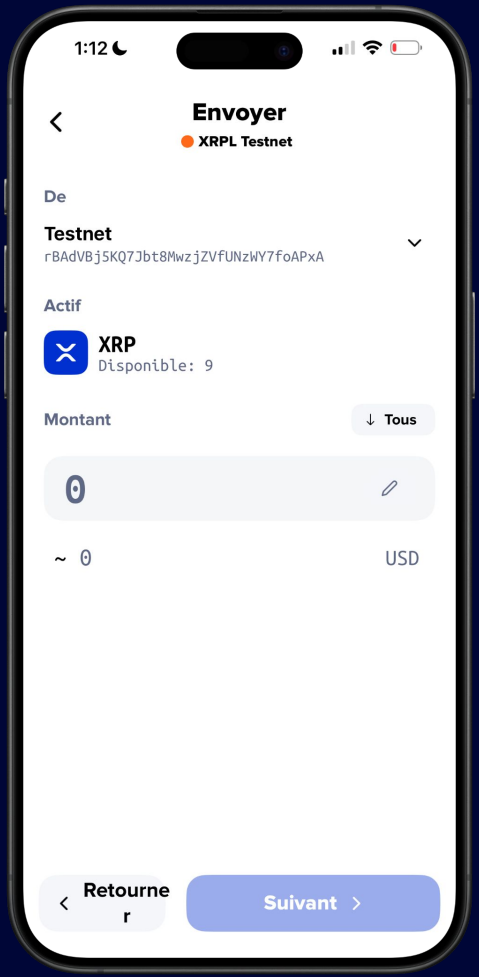
7. Recevoir un paiement



7. Recevoir un paiement



8. Faire un paiement



8. Faire un paiement

Première transaction 📲

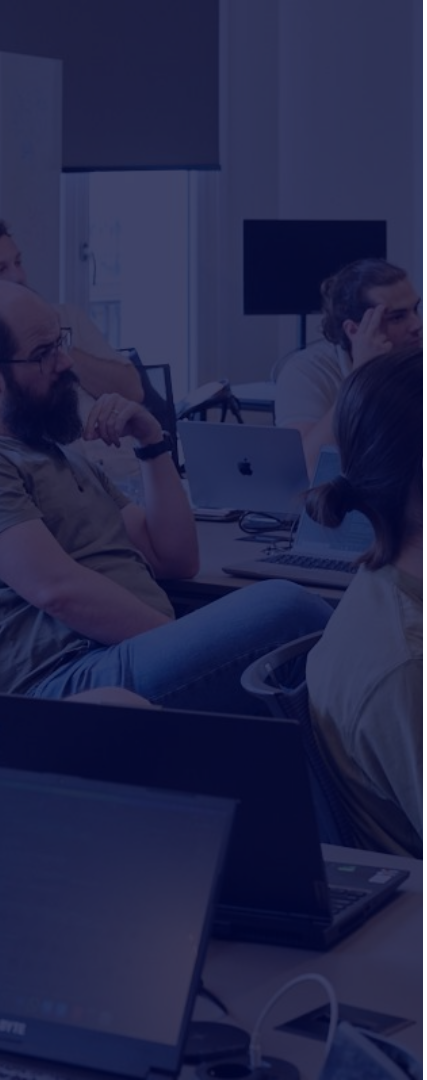
Avec votre voisin de droite 🤝

Pratiquer **par le code** 🧑‍💻

github.com/XRPL-Commons/cohort-devinci-2025

Parcours classique d'une transaction

1. Se connecter à un client
2. Obtenir un wallet
3. Préparer et signer la transaction
4. Obtenir le résultat



Adresses sur le XRP Ledger

Type	Commence par	Nombre de caractères
Adresse de compte	r	35
Clé publique de nœud	n	53
Clé privée	s	29

Les Bases d'une transaction XRPL

Chaque transaction a le même ensemble de **Champs Commun** plus des **Champs Supplémentaires** basés sur le **Transaction Type**

Some Common Fields:

- **Account**: L'adresse qui initie la transaction
- **TransactionType**: Définit le type de transaction (Payment, OfferCreate, etc.)
- **Fee**: Montant de XRP détruit comme coût de transaction
- **Sequence**: Numéro de séquence de transaction du compte

Transactions Types: Briques Fondamentales des Fonctionnalités du XRPL

Transaction Types = **Opérations spécifiques** ou actions sur le XRPL
Une feature peut nécessiter plusieurs types de transaction pour fonctionner complètement
Chaque transaction type a un but et un schéma spécifique

Examples:

La feature NFT inclut plusieurs types de transaction :

- ***NFTokenMint***: *Créer un nouveau NFT*
- ***NFTokenCreateOffer***: *Met un NFT en vente*
- ***NFTokenAcceptOffer***: *Finalise un transfer de NFT*

Les types de transaction sont les unités fondamentales de travail sur le ledger, et les fonctionnalités complexes sont construites en les combinant.

Flags: Customiser les Transactions

Flags = Options qui **modifient le comportement d'une transaction**
Permettent aux développeurs de personnaliser la logique de transaction sans nécessiter de nouveaux types de transaction

Peuvent être combinés (bitwise/bit à bit) pour créer des comportements complexes

Exemples:

NFTokenMint Flags:

- **tfTransferable**: Quand désactivé (0), rend le NFT non-transférable
- **tfBurnable**: Quand activé (1), permet au NFT d'être brûlé par l'émetteur

Les flags permettent un contrôle précis sur le comportement des transactions sans nécessiter de nouveaux types de transaction pour chaque variation.

1. Se connecter à un nœud

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

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

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

2. Créer un wallet

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

3. Préparer et envoyer la 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,
  });
}
```

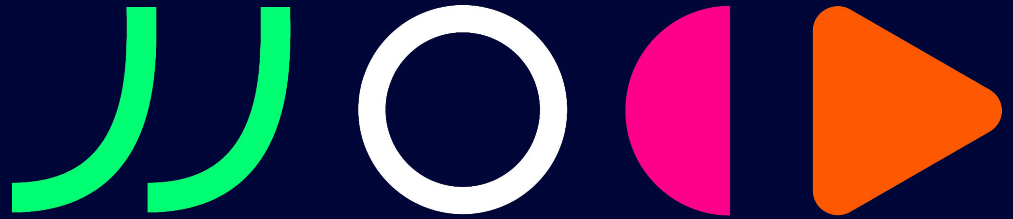
4. Afficher le résultat


```
{
  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: "rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h",
      Fee: "12",
      Flags: 0,
      LastLedgerSequence: 2887344,
      Sequence: 2887324,
      SigningPubKey: "ED598042DFC6F9C65B16002F042B57AA4372EC6B12B9F3862F772A4FCF7B331BBC",
      TransactionType: "Payment",
      TxnSignature: "4ABCFD76572234A1B51BE6509C6364835530EE44C2B59005D4A5ACE4C84B13F9EF0025CF84A5C1F922F500877E2BCFA
0C6696A434024FD2E3A776E0F7B485C07",
      date: 786807550,
      ledger_index: 2887326
    },
    validated: true
  },
  type: "response"
}
```

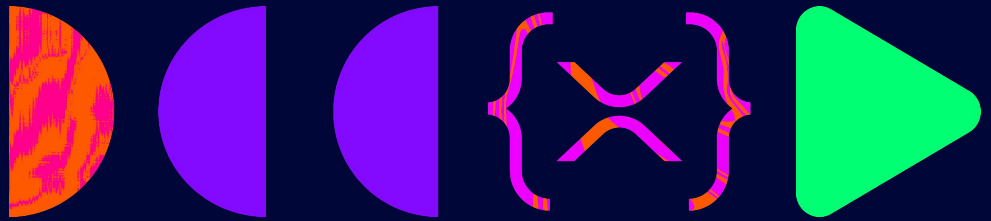
Code complet

```
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();
}
```

Commons



2. Créer un token 💸

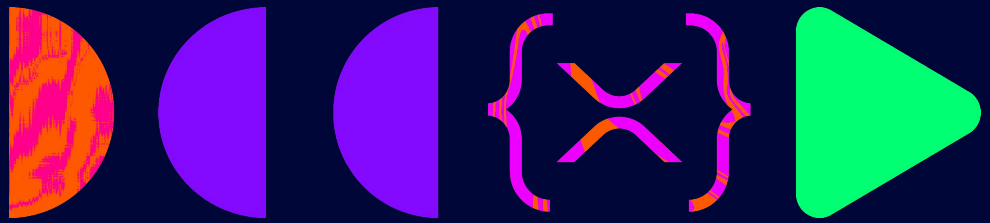


Multi-Purpose Token (MPT)

Create a token

```
const metadata: MPTokenMetadata = {
    ticker: "USDM",
    name: "USD Mathis",
    desc: "A stablecoin pegged to the US Dollar by Mathis",
    icon: "https://example.com/image.png",
    asset_class: "rwa",
    asset_subclass: "stablecoin",
    issuer_name: "Mathis",
};

const createTokenTx: MPTokenIssuanceCreate = {
    TransactionType: "MPTokenIssuanceCreate",
    Account: wallet.address,
    MaximumAmount: "10000000000000000",
    Flags: MPTokenIssuanceCreateFlags.tfMPTCanTransfer + MPTokenIssuanceCreateFlags.tfMPTCanTrade,
    MPTokenMetadata: convertStringToHex(JSON.stringify(metadata)),
};

const tokenCreateTxResult = await client.submitAndWait(createTokenTx, { wallet });
```

3. Acheter et vendre un token 💱



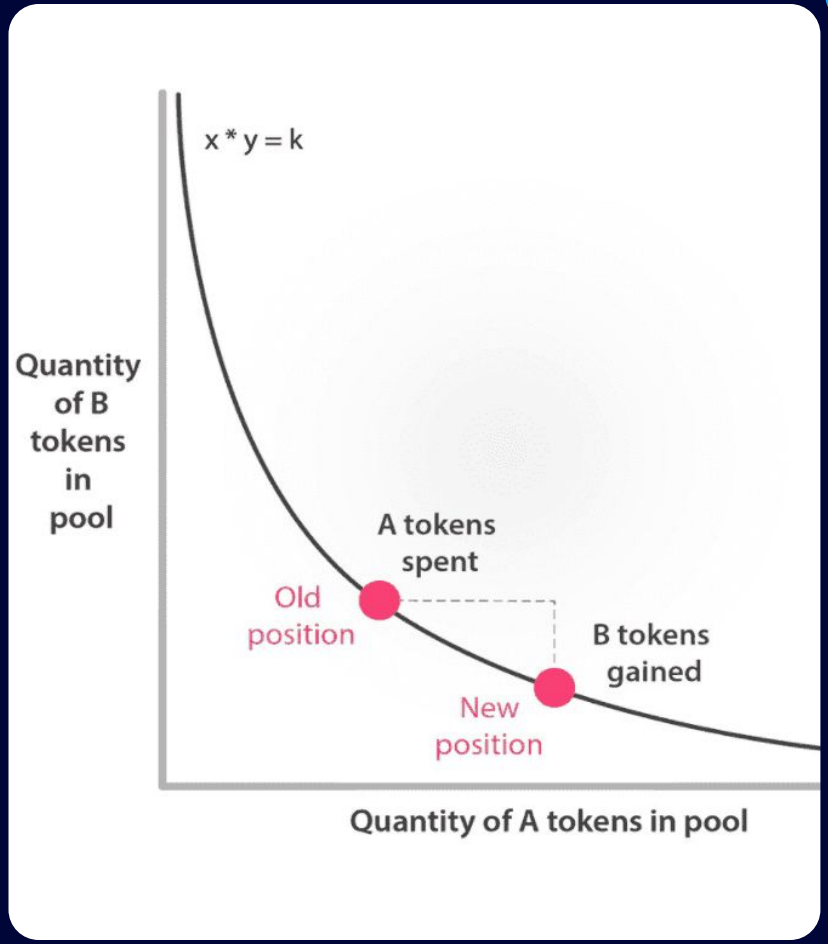
Plateformes d'échanges décentralisés (DEX):
CLOB & AMM

Central Limit OrderBook

- Présent depuis le premier jour
- Un système de matching pour les ordres d'achat et de vente à des prix spécifiques
- Les Ordres sont stockés sur le ledger jusqu'à ce qu'ils soient exécutés ou annulés
- Fonctionne comme une bourse traditionnelle avec des écarts bid/ask

Automated Market Maker

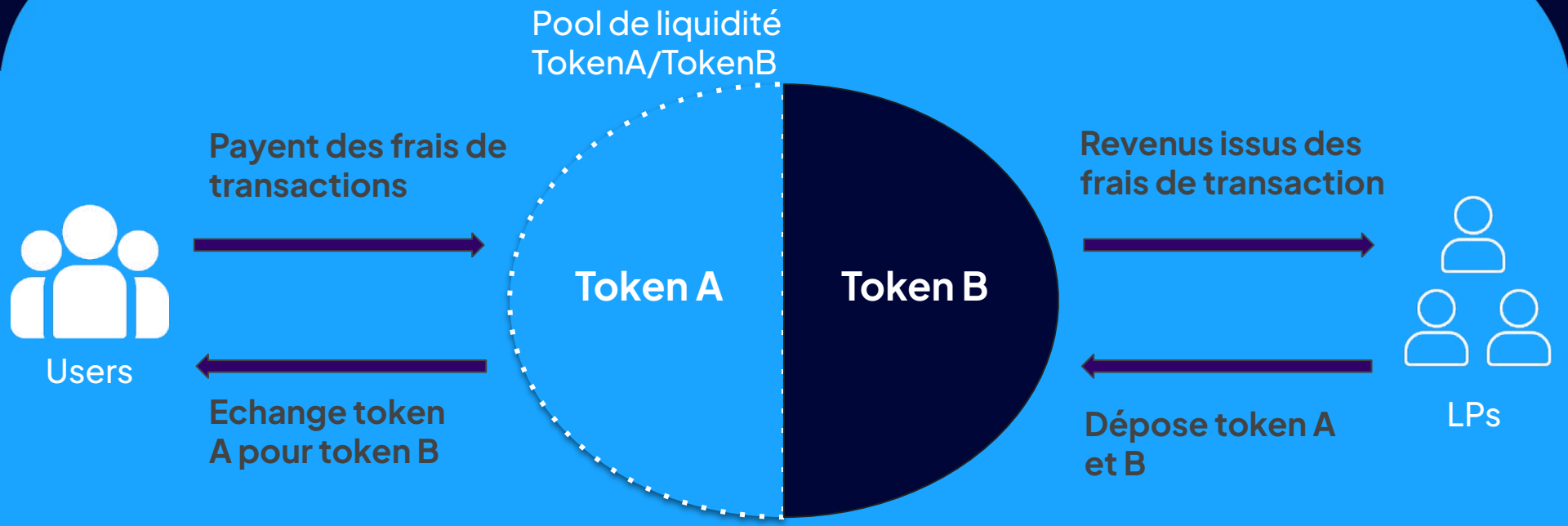
- Ajouté via xls-30d
- Prix déterminé par la formule de produit constant ($x \times y = k$)
- Aucune contrepartie nécessaire - échange contre le pool
- Les fournisseurs de liquidité gagnent des frais sur tous les échanges



Pourquoi utiliser un AMM ?

- Liquidité
  - Mécanisme simple pour les utilisateurs d'échanger des actifs
  - Permet une liquidité constante et une disponibilité sur les pools
  - Idéal pour les marchés moins liquides
  - Tous les ordres sont des ordres au marché
  - Slippage (dépend de la taille du pool)
- Farming
  - Créer et ajouter la liquidité initiale au pool
  - LP représente la proportion de propriété du pool
  - Provide liquidity to AMM
  - Fournir de la liquidité à l'AMM
  - Générer passivement des rendements sur la liquidité

AMM Flow


Pool de liquidité
TokenA/TokenB
Payent des frais de transactions
Users
Echange token A pour token B
Token A
Token B
Revenus issus des frais de transaction
LPs
Dépose token A et B

Constant Product Market Maker

$X * Y = K$

1000 XRP x	x	2000 USD y	=	2 000 000 k	Prix par XRP 2 USD
+100 XRP	SWAP	-181.9 USD			
1100 XRP x	x	1818.1 USD y	=	2 000 000 k	Prix par XRP 1,65 USD

Commons

Impermanent loss (IL)

L'Impermanent loss survient quand le prix des actifs d'un pool de liquidité évolue après leur dépôt, causant des pertes potentielles comparé au fait de garder les actifs hors du pool. Elle doit être compensée par les commissions du pool AMM.

1000 XRP | 2000 USD

Prix par XRP: 2$

Holdings assets: 4k$
LP assets: 4k$

1100 XRP | 1818.1 USD

Prix par XRP: 1.65$

Holdings assets: 3650$
LP assets: 3633.1$

Fees potentiels: 2$ (100XRP @1%)

➡️ En résumé

CLOB

- Gestion du capital plus efficace (vente ou achat)
- Nécessite un ajustement manuel de la liquidité lors du market making
- Aucun risque d'impermanent loss

AMM

- Façon la plus simple d'échanger des tokens
- Gagner des frais en fournissant de la liquidité
- Risque d'Impermanent loss

PathFinding

- Permet d'obtenir les meilleurs taux en trouvant la route optimale entre le CLOB et l'AMM

Créer une Pool sur l'AMM

```
async function createAMM(
  issuer: Wallet,
  receiver: Wallet,
  client: Client,
  tokenCode: string,
) {
  console.log("create AMM", { issuer, receiver, tokenCode });
  let createAmm: AMMCreate = {
    TransactionType: "AMMCreate",
    Account: receiver.address,
    TradingFee: 600,
    Amount: {
      currency: tokenCode,
      issuer: issuer.classicAddress,
      value: "2000000", // 2M tokens
    },
    Amount2: "50000000", // 50 XRP in drops
  };
  console.log(createAmm);

  const prepared = await client.autofill(createAmm);
  const signed = receiver.sign(prepared);
  const result = await client.submitAndWait(signed.tx_blob);

  console.log(result);
  console.log("Create amm tx: ", result.result.hash);

  return;
}
```

Challenge ta compréhension 🧠

Wrap-up! K!

Parle-nous de ton expérience 💬

xrpl.at/cohort-devinci

Thank you!

Commons