

# React



# STELLAR

 OCTOFOX

# REACT WITH STELLAR BLOCKCHAIN (DLT)

ฉบับจับมือเขียน

10.00 - 15.00 วันเสาร์ ที่ 12 มกราคม 2019  
ที่ OCTOFOX ชั้น 29 อาคาร 253 อโศก

โดย



นิพนธ์ ชินะธีมาตรมงคล  
Blockchain Developer



รังสิกร รังสิกวานิช  
Lead Full Stack Engineer

บัตรราคา 3,999 บาท

\*พิเศษ ฟรี!!!

\*สำหรับนักเรียน นักศึกษาและบุคลากร ติดต่อขอโค้ดเข้าร่วมฟรีได้ที่ Facebook: Octofox



# Outline

- Introduction to Stellar Network and DLT
- Stellar Lab
- **Workshop** “No Database Certification Platform”

# Introduction to Stellar Blockchain

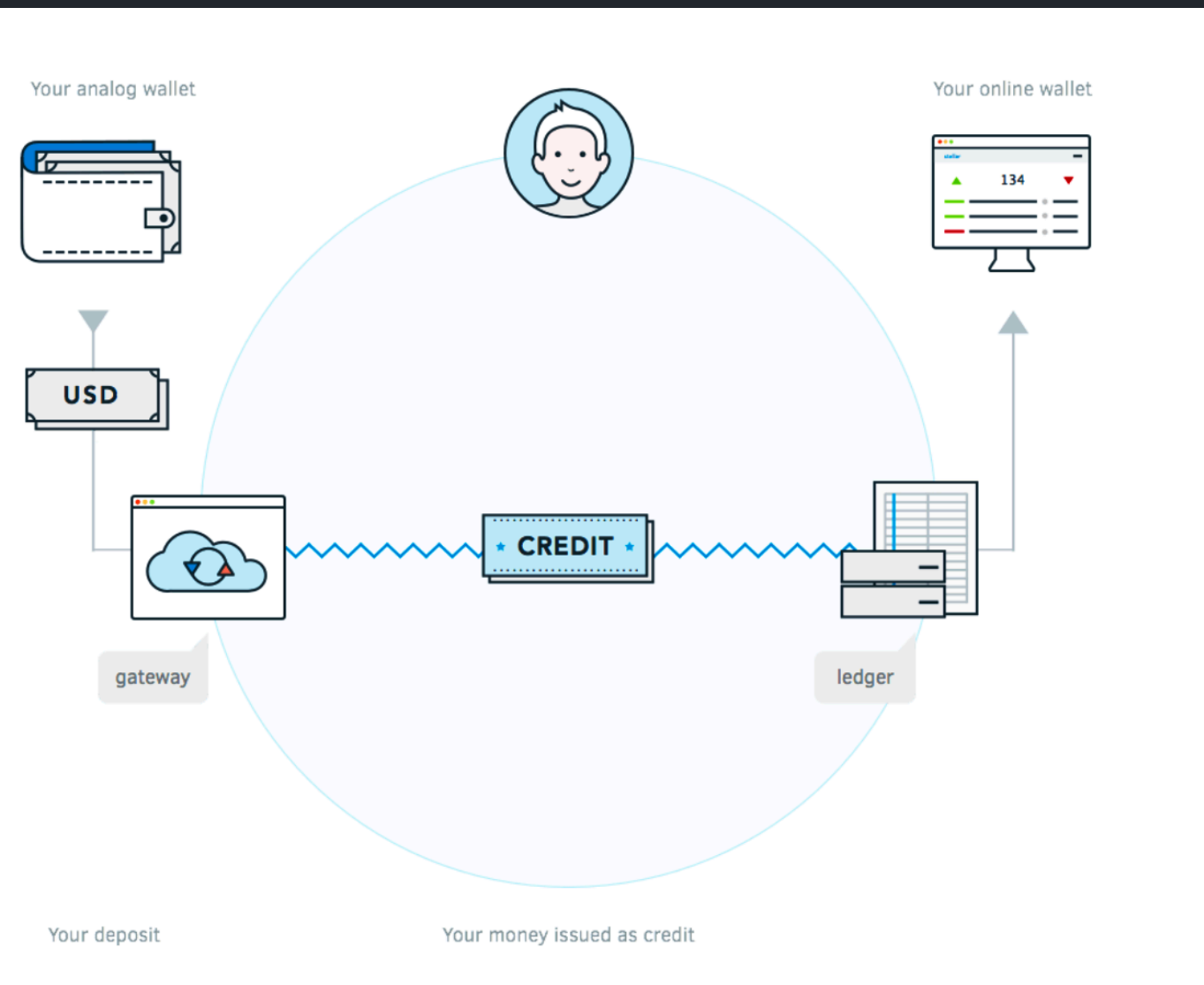
- Created by Jed McCaleb and Joyce Kim

“Stellar is a platform that connects banks, payments systems, and people. Integrate to move money quickly, reliably, and at almost no cost”

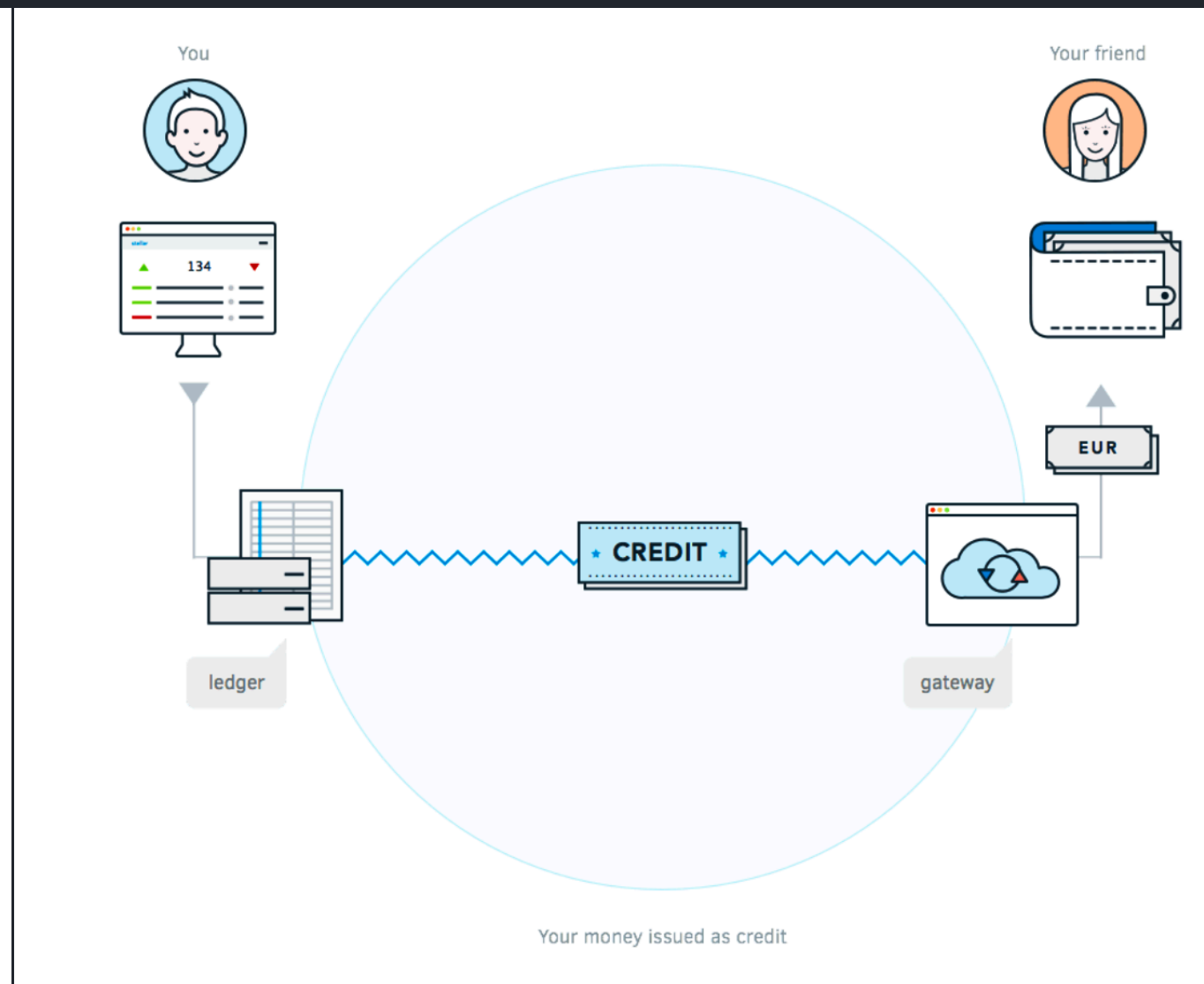
## Centralized money transfer



## \* Stellar - decentralized money transfer \*



Fiat to Token (cash-in)

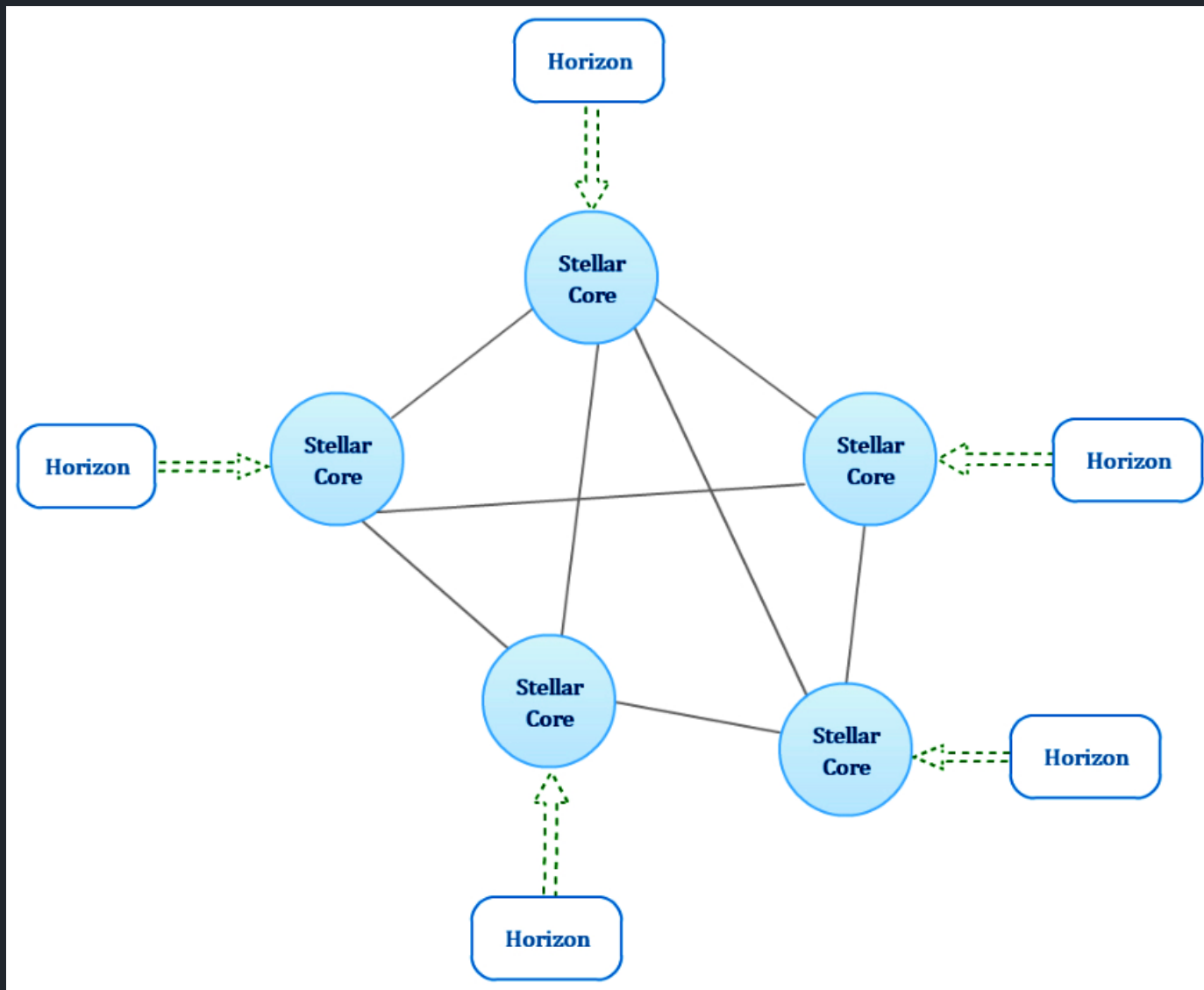


Token to Fiat (ash-out)

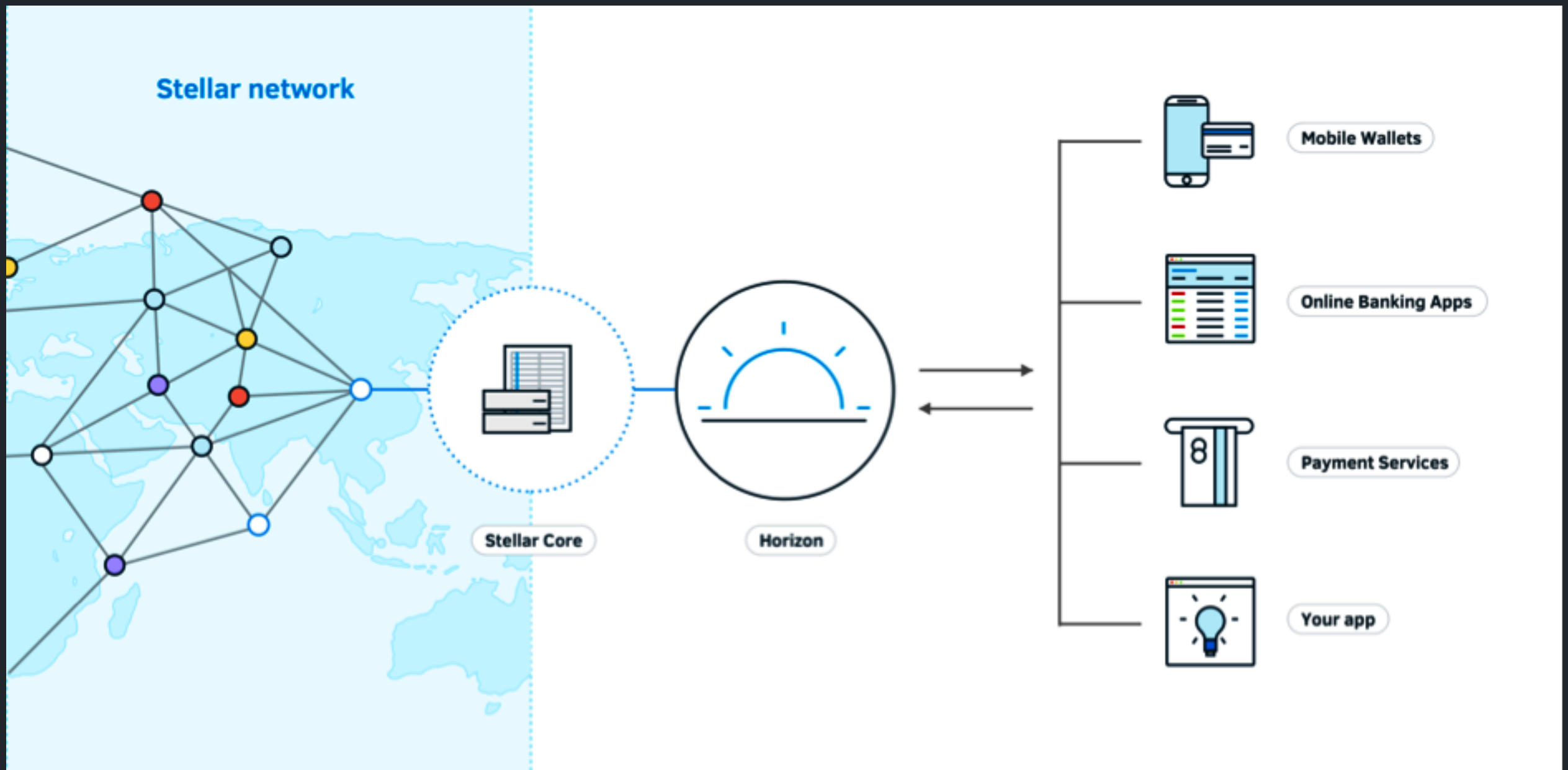
# Stellar Key Features

- Has a decentralized and open database
- Confirmation time: 3-5 seconds
- 1,000 transactions per second
- 0.00001 xlm/tx => 10,000 tx : 1 Baht
- Uses the Stellar Consensus Protocol
- Multi-Signatures and Smart Contracts
- The stellar token is called “Lumen” and denoted by “XLM” A 100 billion XLM has already been pre-mined.
- Has a 1% fixed annual inflation.

# Stellar Horizon



- A RESTful HTTP API server
- Allows users to:
  - Submit transactions
  - Check accounts / transactions
  - Subscribe to events
- Can be connected with various SDKs
  - Javascript SDK
  - Go SDK
  - Java SDK
  - Ruby SDK
  - Python SDK
  - C# SDK





# Stellar Core

- Backbone of stellar network
- Each horizon server connects to stellar core
- Process all transactions through Stellar Consensus Protocol (SCP)
- Hybrid datastore: Blockchain + Database

# Smart Contract in Stellar

- Stellar has no turing machine
- Stellar smart contract is controlled by
  - Set of operations
  - Sequence number
  - Multi-signatures
  - Time bound

# Lab

# lab 1: Create a stellar wallet

# lab 2: Transfer stellar lumens XLM

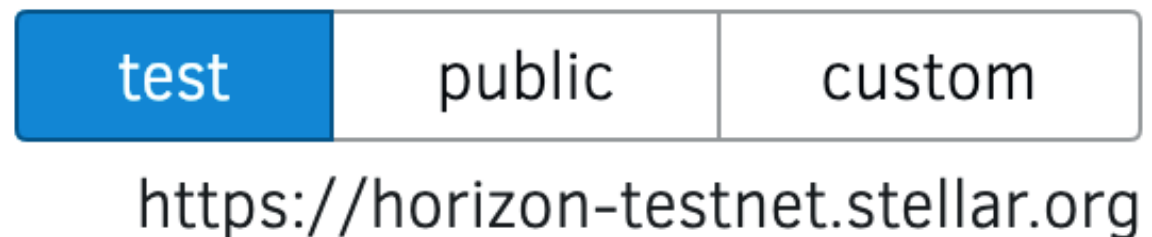
# lab 3: Create a custom asset (coin)

# lab 4: Transfer custom asset

# # lab 1: Create Stellar Account

1. Stellar Laboratory: <https://www.stellar.org/laboratory/>

2. Select test network



test public custom

<https://horizon-testnet.stellar.org>

## Method #1 Friendbot

3. Select **Account Creator** and click button **Generate Keypair**

### 1. Keypair generator

These keypairs can be used on the Stellar network where one is required. For example, it can be used as an account master key, account signer, and/or as a stellar-core node key.

Generate keypair

Public Key

GDRHWCWUZUAJNH2IRTQWLU2SEL4GDK5IOVIVPHF5BCCOZ5TD4HFQN7HE

Secret Key

SAHIVK2FVJJOFOFKC7BO25OSNAZ6JI3LXIIU6KHMEC6PKJDSOC3JDCUWD

[Fund this account on the test network using the friendbot tool below](#)

# # lab 1: Create Stellar Account (Cont.)

4. Copy **Public Key**, paste it in **friendbot** section 2 and click **Get test network lumens**

**2. Friendbot: Fund a test network account**  
The friendbot is a horizon API endpoint that will fund an account with 10,000 lumens on the test network.

GDRHWCWUZUAJNH2IRTQWLU2SEL4GDK5IOVIVPHF5BCCOZ5TD4HFQN7HE

Get test network lumens

Successfully funded GDRHWCWUZUAJNH2IRTQWLU2SEL4GDK5IOVIVPHF5BCCOZ5TD4HFQN7HE on the test network

5. Verify the account by [https://horizon-testnet.stellar.org/accounts/{public\\_key}](https://horizon-testnet.stellar.org/accounts/{public_key})

Or use Endpoint Explorer -> Accounts -> Single Account

# # lab 1: Create Stellar Account (Cont.)

Single Account ?

Account ID

☐ Server-Sent Events (streaming) mode ?

GET

JSON Response

```
{
  "_links": {
    "self": {
      "href": "https://horizon-testnet.stellar.org/accounts/GAVWZ5GFISGYTH4S7VCQCWWBF73BDXD2F"
    },
    "transactions": {
      "href": "https://horizon-testnet.stellar.org/accounts/GAVWZ5GFISGYTH4S7VCQCWWBF73BDXD2F",
      "templated": true
    },
    "operations": {
      "href": "https://horizon-testnet.stellar.org/accounts/GAVWZ5GFISGYTH4S7VCQCWWBF73BDXD2F",
      "templated": true
    }
  }
}
```

```
"sequence": "6774224282714112",
"subentry_count": 0,
"thresholds": {
  "low_threshold": 0,
  "med_threshold": 0,
  "high_threshold": 0
},
"flags": {
  "auth_required": false,
  "auth_revocable": false,
  "auth_immutable": false
},
"balances": [
  {
    "balance": "10000.0000000",
    "buying_liabilities": "0.0000000",
    "selling_liabilities": "0.0000000",
    "asset_type": "native"
  }
],
"signers": [
  {
    "public_key": "GAVWZ5GFISGYTH4S7VCQCWWBF73BDXD2FNQPTHHRMVNYFUPODOQPRQHU",
    "weight": 1,
    "key": "GAVWZ5GFISGYTH4S7VCQCWWBF73BDXD2FNQPTHHRMVNYFUPODOQPRQHU",
    "type": "ed25519_public_key"
  }
],
"data": {}
}
```

# # lab 1: Create Stellar Account (Cont.)

## Method #2 Create Account Operation

### 6. Go to Transaction Builder

6.1 Enter **Public Key** in Source Account

6.2 Click **Fetch next sequence...**

6.3 **Add Operation** and select **Create Account**

6.4 Enter a new **Public Key** in **Destination**

6.5 Enter **Starting Balance** (i.e. 2500)

*[Perform submit transaction]*

6.6 Verify that no error message shown

6.7 Click **Sign in Transaction Signer**

6.8 Enter **Secret Key** of source account in **Add Signer**

6.9 Click **Submit to Post Transaction endpoint**

6.10 Click **Submit** and verify result

The screenshot shows the Stellar Transaction Builder interface. A dropdown menu is open under the 'Operation Type' field, listing various operations. 'Create Account' is selected and highlighted in blue. Other visible options include Payment, Path Payment, Manage Offer, Create Passive Offer, Set Options, Change Trust, Allow Trust, Account Merge, Inflation, Manage Data, and Bump Sequence. The form fields below the dropdown are 'Destination', 'Starting Balance', and 'Source Account (optional)'. An example public key is provided: GCEXAMPLE5HWNK4AYSTEQ4UWDKHTCKA. At the bottom right, there is a '+ Add Operation' button.

# # lab2: Transfer Stellar Lumens (XLM)

Go to [Transaction Builder](#)

1. Enter sender's **Public Key** in Source Account
2. Click **Fetch next sequence...**
3. **Add Operation** and select **Payment**
4. Enter receiver's **Public Key** in **Destination**
5. Select **Native** and enter **Amount**

*[Perform submit transaction]*

6. Verify that no error message shown
7. Click **Sign in Transaction Signer**
8. Enter **Secret Key** in **Add Signer**
9. Click **Submit to Post Transaction endpoint**
10. Click **Submit** and verify result

1

Operation Type ?

Destination

Asset

Amount

Source Account (optional)

Select operation type

Create Account

✓ Payment

Path Payment

Manage Offer

Create Passive Offer

Set Options

Change Trust

Allow Trust

Account Merge

Inflation

Manage Data

Bump Sequence

et to a destination acco

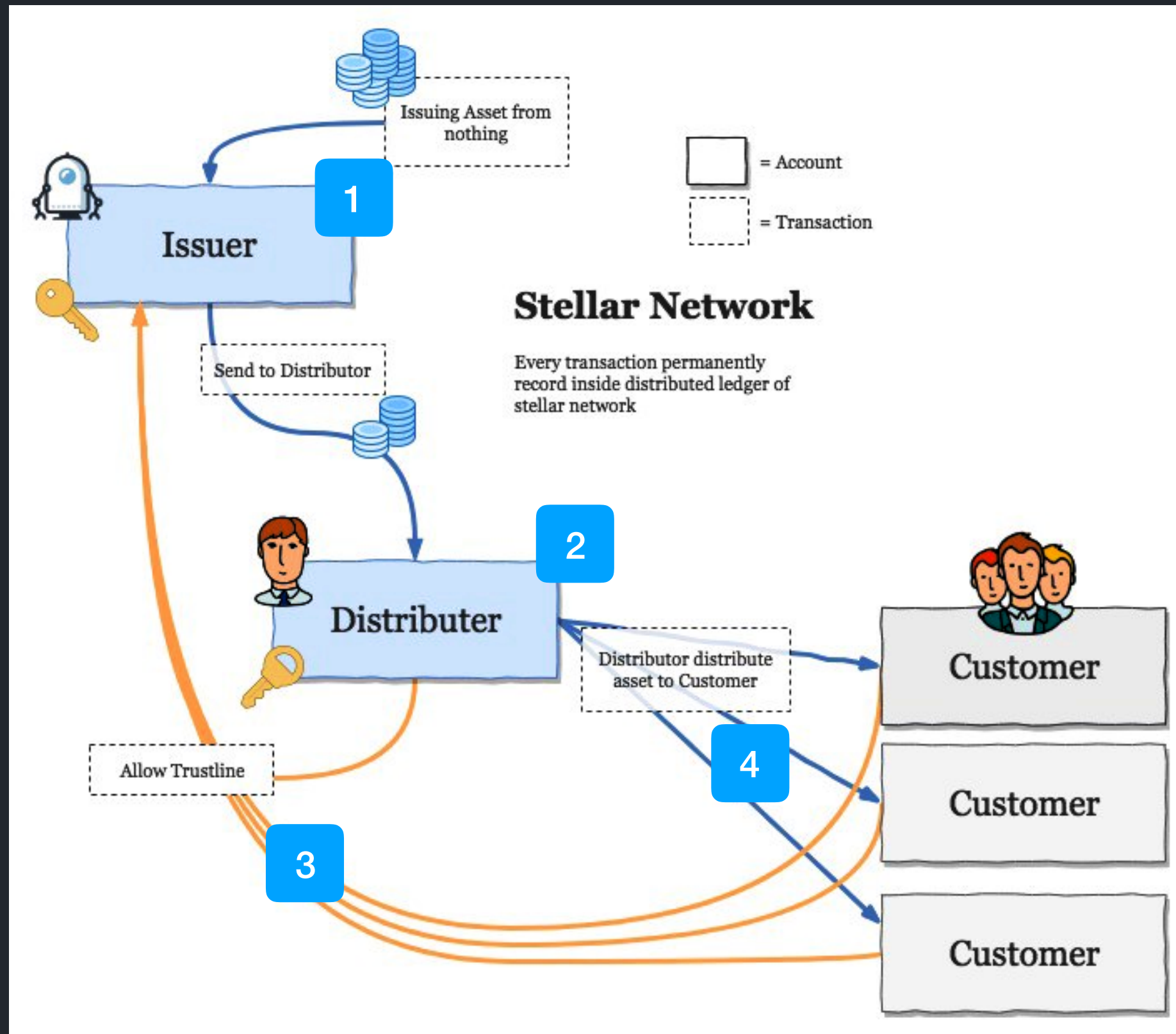
NK4AYSTEQ4UWDK

Alphanumeric 1

Example: GCEXAMPLE5HWNK4AYSTEQ4UWDK



# Create Custom Asset



# # Lab 3 : Create Custom Asset

1. Create 2 accounts with Friendbot, noted as **Issuer** and **Distributor**

2. Create trust line in Distributor account

1. Go to Transaction Builder
2. Source account: **Distributor's Public Key**
3. Add Operation: **Change Trust**
4. Asset: **Alphanumeric 4 or 12 and Enter asset code**
5. Enter **Issuer's Public Key** for **Issuer Account ID**
6. *[Perform submit transaction...]*

3. Transfer asset from Issuer to Distributor

1. Go to Transaction Builder
2. Source account: **Issuer's Public Key**
3. Add Operation: **Payment**
4. Destination: **Distributor's Public Key**
5. Asset: **Alphanumeric 4 or 12 and Asset Code and Issuer's Public Key**
6. Enter **Amount**
7. *[Perform submit transaction...]*

1

Operation Type ?

Change Trust

Creates, updates, or deletes a trustline.  
[See documentation for Change Trust](#)

Asset

Alphanumeric 4Alphanumeric 12

Trust Limit  
(optional)

Leave empty to default to the max int64.  
Set to 0 to remove the trust line.

Source Account  
(optional)

Example: GCEXAMPLE5HWNK4AYSTEQ4UWDK

+ Add Operation

# # Lab4 : Trust line and Transfer Custom Asset

## 1. Change trust line for wallet account

1. Go to Transaction Builder
2. Source account: **Wallet**
3. Add Operation: **Change Trust**
4. Asset: **Alphanumeric 4 or 12** and Enter asset code
5. Enter **Issuer Public Key** for Issuer Account ID
6. *[Perform submit transaction...]*

## 2. Transfer asset from distributor to wallet

1. Go to Transaction Builder
2. Source account: **Distributor Public Key**
3. Add Operation: **Payment**
4. Destination: Distributor Public Key
5. Asset: **Alphanumeric 4 or 12** and **Asset Code** and **Issuer Public Key**
6. Enter **Amount**
7. *[Perform submit transaction...]*