Exchange Use Case

By the end of this guide you should know how an exchange can integrate with the OMG Network.

- 1. Introduction
 - 1.1 Overview
 - 1.2 Requirements
 - User Personas
 - User Stories
- 2. Proposed Design
 - 2.1 Integration Scope
 - 2.1.1 Generate ETH Addresses
 - 2.1.2 Query Blockchain
 - 2.1.3 Make Deposits
 - 2.1.4 Make Withdrawals
 - 2.1.5 Merge UTXO
 - 2.1.6 Setup a Watcher
 - 2.1.7 Deposit Funds to the OMG Network
 - 2.1.8 Rebalance Wallets
 - 2.2 Integration UI

1. Introduction

1.1 Overview

The main goal of this document is to demonstrate how a token issuer or an exchange can add an OMG Network support for depositing and withdrawing a particular ERC20 token (e.g. USDt).

The integration with the OMG Network can significantly help the entire Ethereum ecosystem to offload a certain volume from the rootchain, as well as introduce faster and cheaper transactions with the same security guarantees as to the Ethereum network.

This document doesn't cover how exchange customers can deposit funds to the OMG Network. Please refer to the Web Wallet Quick Start for more details on this issue.

1.2 Requirements

User Personas

- 1. Customers participants and active users/traders of a given exchange.
- 2. Integrator development and management staff at an exchange that evaluate the requirements and deliver the integration to the OMG Network.
- 3. OMG Network a technical and management team within the OMG Network that supports the Integrator during the integration process.

User Stories

- 1. As a customer, I can deposit ERC20 tokens (e.g. USDt) from the OMG Network compatible wallet to my exchange account.
- 2. As a customer, I can use deposited funds to make trades on a given exchange.
- 3. As a customer, I can withdraw ERC20 tokens from my exchange account into the OMG Network wallet (web or mobile).

2. Proposed Design

The proposed design includes improvements to an existing exchange platform. However, there are no strict requirements one has to follow due to the distinct nature of given exchange infrastructure, development approach, UI/UX design, etc.

2.1 Integration Scope

The scope for integrating with the OMG Network by a given exchange includes the following steps:

2.1.1 Generate ETH Addresses

As an exchange, you need to be able to generate new Ethereum addresses for your customers. The OMG Network reflects the same addresses as the Ethereum network, thus you don't need to implement extra logic for your platform's wallet. If you already have Ethereum supported, you shouldn't have any issues.

Otherwise, you can create a new public/private key pair using <u>web3.eth.accounts.create</u> function provided by the web3 library. Note, if the keys are generated by a third-party provider such as a custodian, the integrator must be able to sign a transaction for it with <u>EIP-712</u> <u>signature</u> via <u>signTypedData_v3()</u> or <u>personalSign()</u>. See <u>eth-sig-util</u> for reference.

This document doesn't cover best practices for managing and storing keys due to a variety of methods different exchanges use to achieve a high level of security guarantees on their platform.

2.1.2 Query Blockchain

It's important to understand how to query data from the OMG Network. There are several things that might be important for any exchange:

Query transactions

- Transaction details use getTransaction(id) function by the <u>omg-js</u> or /transaction.get endpoint of the <u>Watcher Info API</u>.
- Transactions details filtered by an address, block number, or metadata use getTransactions(filters) function by the <u>omg-js</u> or /transaction.all endpoint of the <u>Watcher Info API</u>.

Query accounts

- Account balance use getBalance(address) function by the <u>omg-js</u> or /account.get_balance endpoint of the <u>Watcher Info API</u>. For more details, check <u>Retrieve Balances</u> guide.
- Account UTXO use getUtxos(address) function by the omg-js or /account.get_utxos endpoint of the Watcher Info API.
- Account transactions use getTransactions(filters) function by the <u>omg-js</u> or /account.get_transactions endpoint of the <u>Watcher Info API</u>.

Query network data

• Network fees - use getFees function by the omg-js or /fees.all endpoint of the Watcher Info API.

You can find more details about these functions using the following resources:

- omg-js library the official JavaScript library that allows implementing the required functionality
- omg-js documentation a list of all functions of the omg-js library
- omg_is-samples a JavaScript project that contains implementation for the core features of the OMG Network
- Tutorials installation process, code samples, and various tips for using the omg-js library
- Watcher Info API an API specification of the Watcher's Informational Service
- <u>Blockchain Explorer</u> a tool that allows viewing transactions and blocks that happen on the network

2.1.3 Make Deposits

The Deposit terminology for an exchange has a different meaning from a deposit on the OMG Network. It's very important to distinguish between those two:

- 1. OMG Network deposit represents an ETH transaction to the corresponding Vault smart contract on the Ethereum network. This creates a new deposit on the OMG Network and allows the depositor to transact on the network until a user decides to withdraw them back to Ethereum.
- 2. Exchange deposit represents a standard transaction on the OMG Network. A safe number of confirmations for a given exchange should equal to a safe number of confirmations on the Ethereum network. This is known as <u>deposit finality period</u> and is currently set to 10 blocks. The confirmations are counted on the Ethereum because the OMG Network relies on rootchain (Ethereum) security and creates blocks only when new transactions are being formed (i.e. on-demand).

While implementing a deposit functionality on a given exchange, you should refer to the latter definition. You can implement this step using the <u>Make a Transfer</u> guide.

Note, all of the code samples on the developer portal use Ropsten testnet as an example. The production version on the mainnet will have the same logic, except for the fees:

- Ropsten uses ETH as a fee
- Mainnet uses OMG token as a fee

You can replace the fee object when constructing a transaction body with the OMG contract as follows:

```
fee: {
    currency: 0xd26114cd6ee289accf82350c8d8487fedb8a0c07
}
```

Also, make sure to check for response type before signing a transaction. The existing implementation has two possible responses, one of them creates a merge transaction. You can find more details here. The same limitation applies to the /transaction.create endpoint of the Watcher Info API.

2.1.4 Make Withdrawals

Withdrawals on a given exchange also represent a standard transaction on the OMG Network. Therefore, you can use the step above.

2.1.5 Merge UTXO

The OMG Network uses the UTXO-based model for keeping track of balances on its chain. Each transaction can have up to 4 UTXO as inputs and can create up to 4 UTXO as outputs. This means that depositing smaller amounts of funds will create UTXO of low value, and may cause potential issues during funds withdrawal due to the following:

The first two outputs are used to cover the transferred value and transaction fee change, leaving only two outputs. Considering that it's very unlikely a user will withdraw the amount that equals a single available UTXO value, there will be another change output created. This means that the total number of UTXO you can use for a single withdrawal is 2. Therefore, it's recommended to check that your hot wallet has multiple UTXO of higher value during each withdrawal from an exchange.

In general, we suggest every exchange client merge UTXO more frequently or even automate this process as a background service. Merge operations are free because they benefit the network.

You can implement this functionality by following Merging UTXO guide.

2.1.6 Setup a Watcher

Watcher is a service that guarantees data availability, secures the network, and allows to maintain the trustlessness of the OMG Network. It's recommended for every client to run a separate Watcher instance to rely on their own information source about the incoming deposits, transactions, exits, and byzantine events that happen on the OMG Network.

It also means you fully trust the OMG Network operator with everything that occurs on the network. Running your own Watcher ensures that only valid transactions and blocks are created and confirmed by the childchain operator.

There are several ways to deploy a Watcher. You can use the one you prefer the most via this guide.

2.1.7 Deposit Funds to the OMG Network

After the core integration is done, you should deposit some funds to the OMG Network. This will allow customers to withdraw funds from an exchange account to the OMG Network compatible wallet.

You can deposit funds using the <u>Deposit Funds</u> guide or via one of our supported wallets. Refer to <u>Environments</u> to find the list of wallets you can work with right now.

2.1.8 Rebalance Wallets

Lastly, you need to understand how to rebalance the corresponding hot and cold wallets. You can do this by sending funds to another wallet directly or withdraw funds back to the Ethereum network first. This step is fully covered in <u>Start a Standard Exit</u> guide, as well as in the <u>Web Wallet Quick Start</u>.

Each withdrawal from the OMG Network requires to pass an exit period before you can retrieve your funds safely. Note, the exit period differs for each environment:

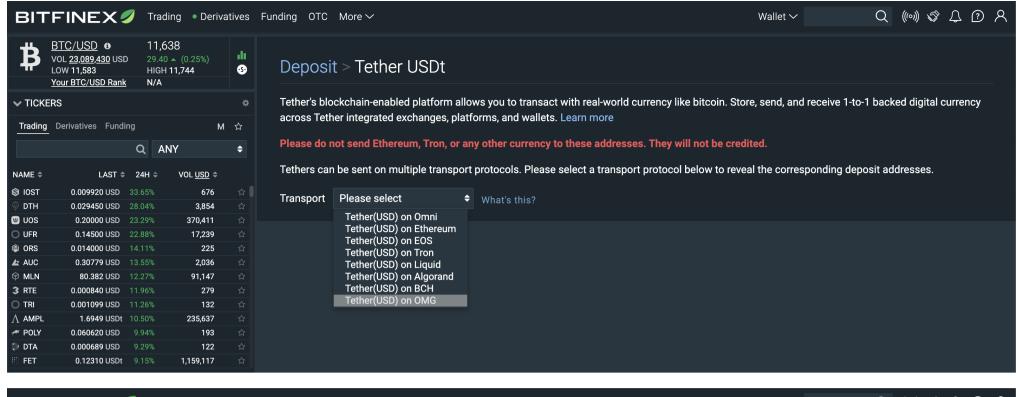
- Ropsten 86400 seconds (1 day)
- Mainnet 604800 seconds (1 week)

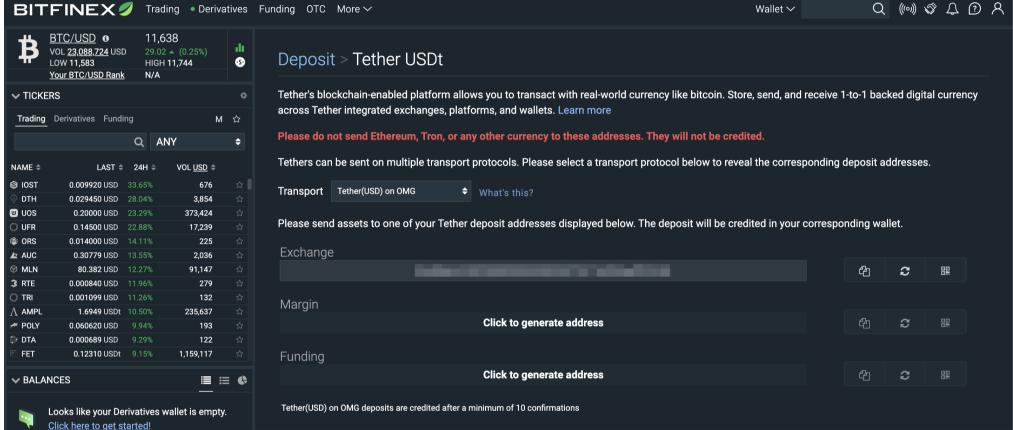
2.2 Integration UI

Each exchange maintains a specific set of design principles, brand book, color palette, etc. Therefore, it's recommended to follow the established user flow and UX principles the users are already familiar with.

Here's an example of the integration by one of our clients:

Deposit View





Withdrawal View

