



Real Estate Tokenization Solution Brief

Overview, Architecture and Demo

Dr. Pavel Kravchenko, Co-founder of Distributed Lab

pavel@distributedlab.com

Dmitry Haidashenko, CTO Distributed Lab

dh@distributedlab.com

Solution Overview

Current solution demo was created on top of the default TokenD (<https://docs.token.d.io/>) core by only modifying web-client and admin-panel applications. It can be used to get hands-on experience, show a live demo or develop your own solution for real estate tokenization.

Purpose of and expectation from the system

Real Estate Investment Trusts have to deal with complexity in managing investor agreements, ownership transfer operations, as well as cash and dividend flow. Blockchain technology can provide an immutable ledger of ownership records and enables easier, transparent, and efficient management of property and cash flows.

Necessary modules of the system: identity and account management, p2p transfers, cash and dividend management, role management and deposit / withdraw gateways.

The platform for tokenized REITs, which is flexible in terms of the legislation (regulatory adaptable) and which covers the whole range of functionality needed for the real estate investing operations. The goal is to provide users with an easy-to-use system which is self-sufficient in terms of their needs.



Workflow

* REIT – Real estate investment trust

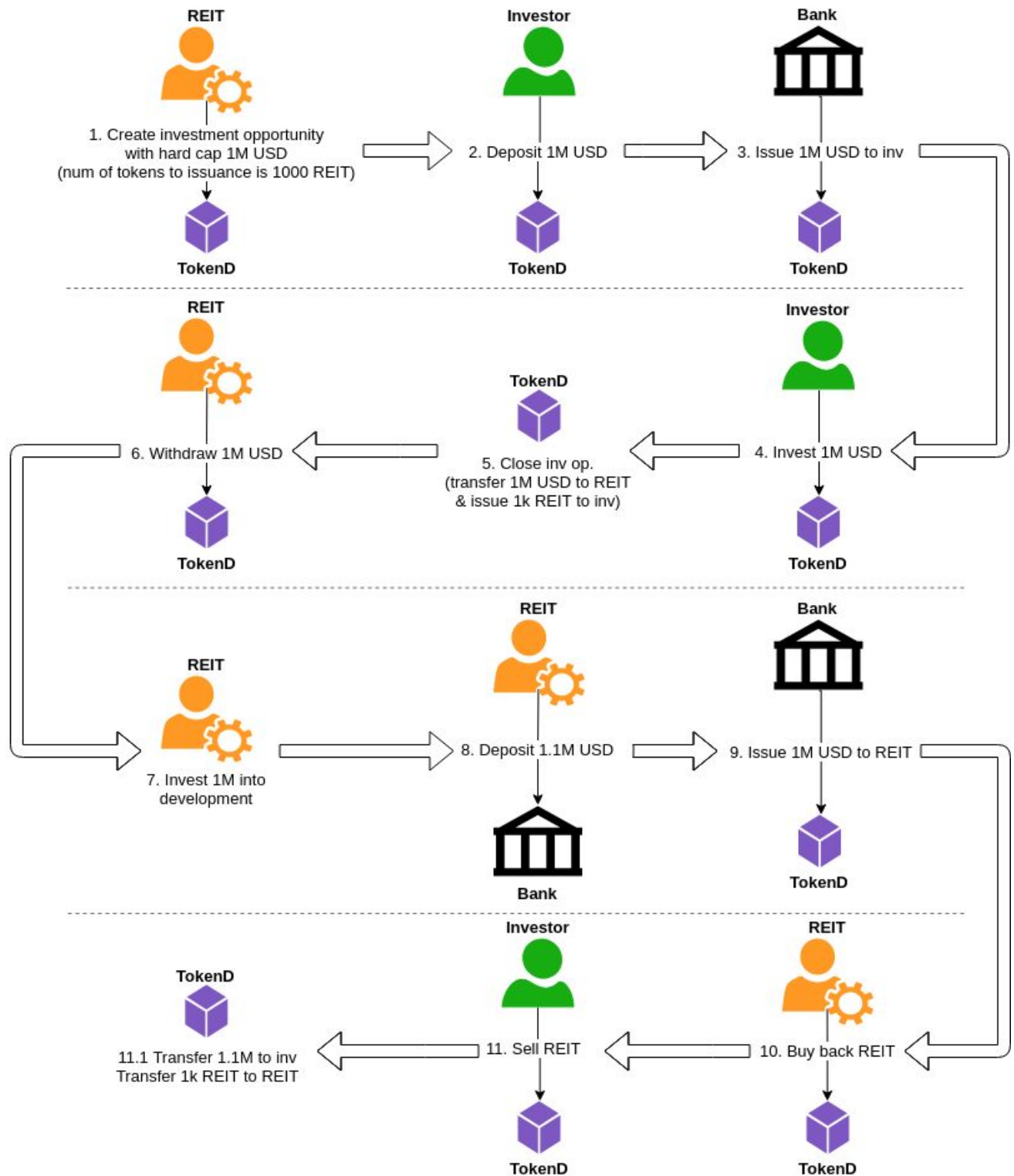


Figure 1. Real Estate tokenization workflow

Roles

RE entrepreneur is able to create their own REITs, invest in REITs of other entrepreneurs, and also trade them on the secondary market with other verified users.

Investor — is able to invest in REITs of entrepreneurs and also trade them on the secondary market with other verified users.

Admin is responsible for particular processes such as user verification, limits/fees imposition, REITs opportunity verification, etc. What is more, the roles of admins have to be differentiated to avoid, for example, bribery. This is mostly related to the roles performing user KYC and those revising *RE opportunities*. Also, ideally, there should be different admins who verify different types of users and also who impose user limits on the platform.

Types of REITS

Depending on their business logic, entrepreneurs should be able to create REIT opportunities of two types. These types differ according to how RE entrepreneurs will later settle with their investors: either through yearly dividends they would pay off or through buying back all the bonds with the promised extra margin after the maturity date.

Functionality requirements

- Match real estate entrepreneurs to investors.
- Real estate entrepreneurs can create their own REITs and accrue funds from investors.
- Investors can deposit fiat or crypto currencies and invest in real-estate properties in exchange for bonds or revenue shares.
- Investors can trade their assets on the secondary market.
- Investors can receive yearly dividends with revenue shares instrument or liquidate their assets to fiat currency with extra margin via bonds instruments.

User stories

EPIC: Legitimacy control

Story #1:

As a **platform owner**, I want to have **multiple independent admins** responsible for different verification processes to avoid the **possibility of bribery**.

Acceptance criteria:

- There is a possibility of granting different admins with different rights (e.g., one for the user identification, another for the real estate opportunity revision).

Story #2:

As an **admin**, I want to be able to **review user identities** and also **ask for additional KYC data** to make sure a **party is legitimate** before I grant them the corresponding access.

Acceptance criteria:

- Admin with corresponding permission can reject the user KYC request
- Admin can, at any time, ask for a repeat KYC review and request additional data.

Story #3:

As an **admin**, I want to be able to **revise opportunity requests** and also **ask for additional data** to make sure that the REIT opportunity is legitimate.

Acceptance criteria:

- During the verification process, an admin with the corresponding permission can ask for a repeat opportunity review and request additional data.

Story #4:

As an **admin**, I want to be able to **set limits** on the user accounts to make sure **users only perform what they have permission to perform**.

Acceptance criteria:

- Admins can set limits on:
 - Accounts
 - Assets



- Time bounds (daily, weekly, monthly)

EPIC: User functionality

Story #1:

As a **user**, I want to be able to be **both an investor and a real estate entrepreneur** to have **all** my real estate investment **activity within one platform**.

Acceptance criteria:

- There are two types of accounts: investor, RE entrepreneur
- To be able to invest, a user has to pass KYC for *investor*
- To be able to create opportunities, a user has to pass KYC for *RE entrepreneur*
- RE entrepreneur can do both create their opportunities and invest in those of other RE entrepreneurs.

Story #2:

As a **user**, I want to be able to **deposit/withdraw funds** via/on my credit card or using the bank transfer.

Story #3:

As a **user**, I want to be able to **deposit/withdraw my crypto funds** using my address/account.

Story #4:

As a **user**, I want to be able to transfer my funds to other users.

Acceptance criteria:

- Users can transfer funds to each other (an asset can only be transferred if neither the sender nor the receiver are imposed with corresponding limits).

Story #5:

As a **user**, I want to be able to see the **history of all my activity** to **keep control** of it.

Acceptance criteria:

- In the user account, in the tab *Activity*, a user can see all their activity performed (transfers, investing, trading, opportunities creation request, etc.)

EPIC: Investor functionality

Story #1:

As an **investor**, I want to be able to **cancel my investment (or reinvest)** at any time to **reduce risks** related to scam or fraud.

Story #2:

As an **investor**, I want to be able to **trade REITs on the secondary market**.

Acceptance criteria:

- If not imposed with corresponding limits, a user can trade a particular asset on the secondary market.

EPIC: RE entrepreneur functionality

Story #1:

As a **RE entrepreneur**, I want to be able to provide a wide **variety of information about my opportunity** request to make sure that potential **investors get interested**.

Acceptance criteria:

- RE entrepreneur can fill their opportunity request with photos, description, etc.

Story #2:

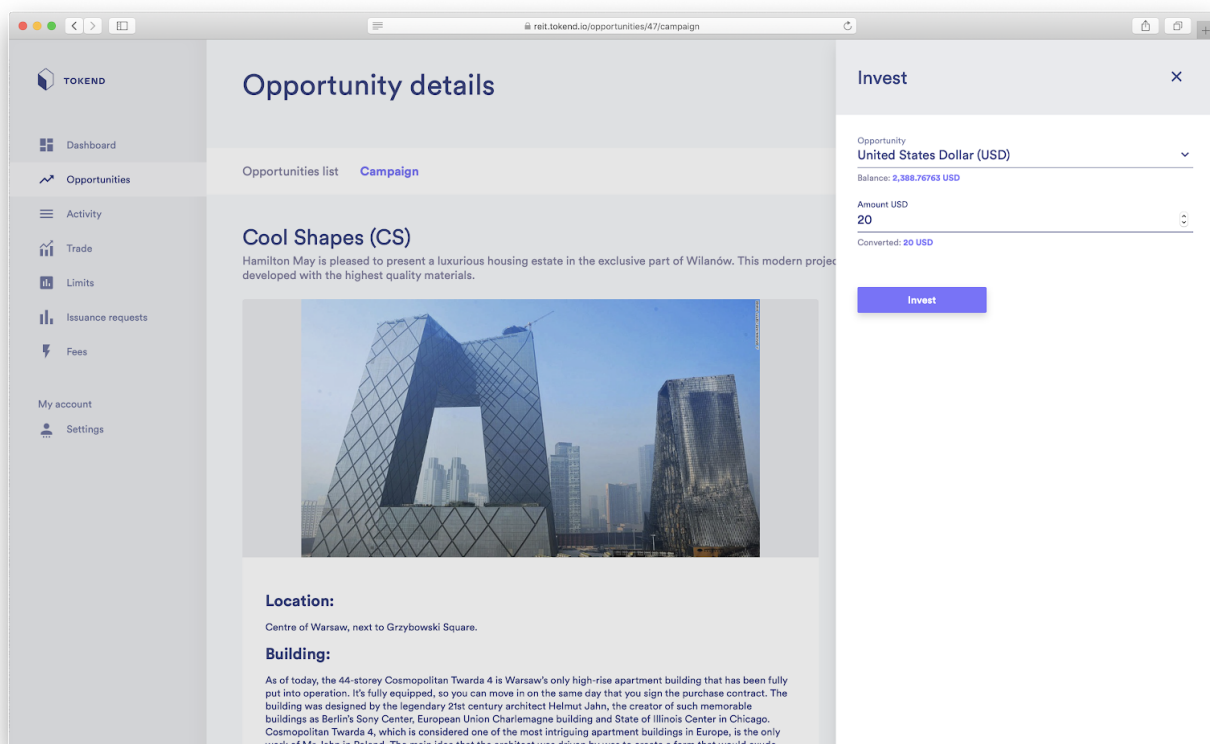
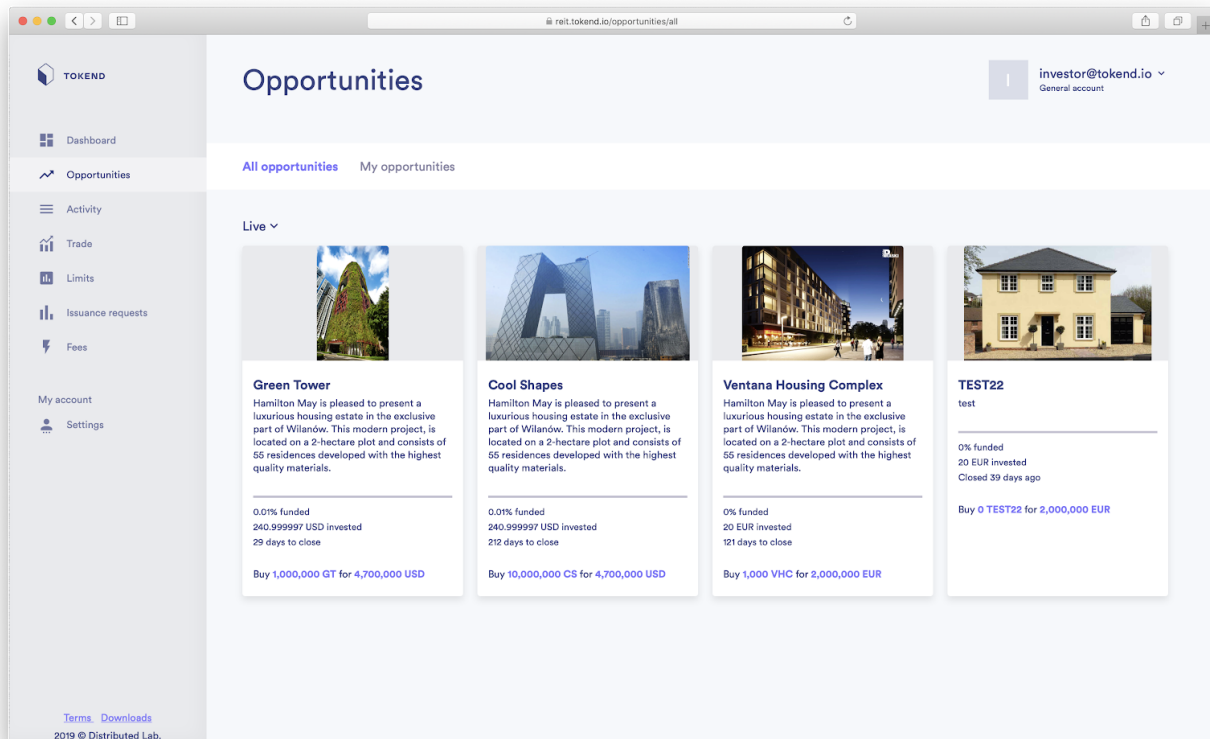
As a **RE entrepreneur**, I want to choose the **type of opportunity** I create to choose the specific way of **how I settle with my investors**.

Acceptance criteria:

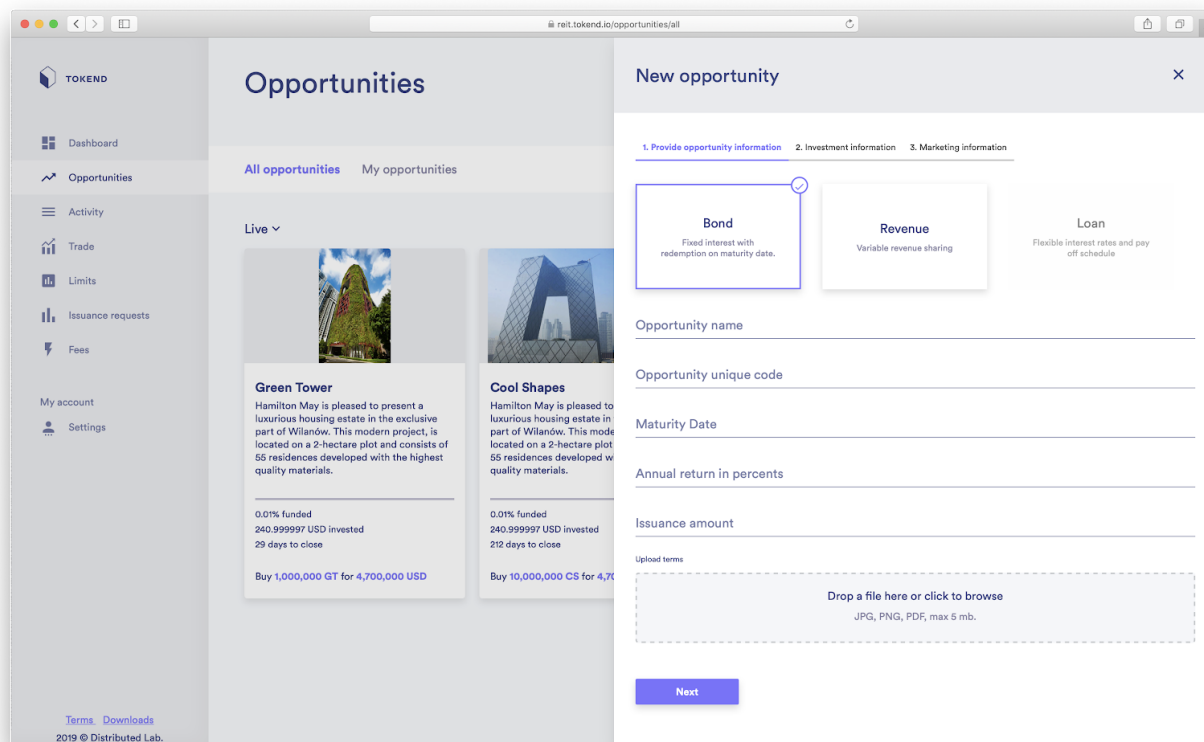
- Opportunities can be of two types: bond and revenue
- *Bond* opportunity means that after the maturity date, a RE entrepreneur buys back all investors' tokens with a promised extra charge
- *Revenue* opportunity means that each year, a RE entrepreneur pays off the promised dividends to the investors (in any desired system asset).

Screenshots

Investor

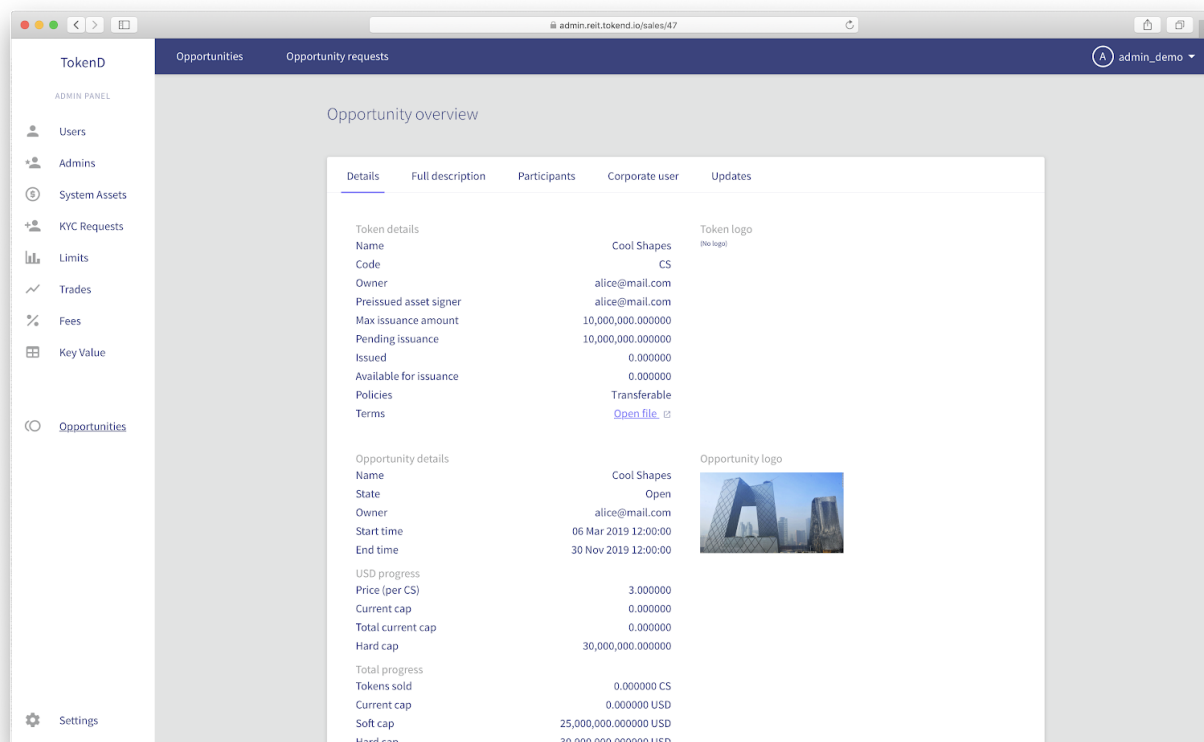


Real Estate Entrepreneur



The screenshot shows the 'Opportunities' page on the TokenD platform. The left sidebar contains navigation links: Dashboard, Opportunities, Activity, Trade, Limits, Issuance requests, Fees, My account, and Settings. The main content area displays two opportunity cards: 'Green Tower' and 'Cool Shapes'. The 'New opportunity' form is open on the right, showing three tabs: '1. Provide opportunity information', '2. Investment information', and '3. Marketing information'. The form includes fields for Opportunity name, Opportunity unique code, Maturity Date, Annual return in percents, and Issuance amount. It also has a section for 'Upload terms' with a file upload area and a 'Next' button.

Platform Administrator



The screenshot shows the 'Opportunity overview' page for the 'Cool Shapes' opportunity. The page is divided into two main sections: 'Token details' and 'Opportunity details'. The 'Token details' section includes fields for Name, Code, Owner, Preissued asset signer, Max issuance amount, Pending issuance, Issued, Available for issuance, Policies, and Terms. The 'Opportunity details' section includes fields for Name, State, Owner, Start time, End time, USD progress, Price (per CS), Current cap, Total current cap, Hard cap, Total progress, Tokens sold, Current cap, Soft cap, and Hard cap. The page also features a 'Token logo' and an 'Opportunity logo'.

Token details		Opportunity details	
Name	Cool Shapes	Name	Cool Shapes
Code	CS	State	Open
Owner	alice@mail.com	Owner	alice@mail.com
Preissued asset signer	alice@mail.com	Start time	06 Mar 2019 12:00:00
Max issuance amount	10,000,000.000000	End time	30 Nov 2019 12:00:00
Pending issuance	10,000,000.000000	USD progress	
Issued	0.000000	Price (per CS)	3.000000
Available for issuance	0.000000	Current cap	0.000000
Policies	Transferable	Total current cap	0.000000
Terms	Open file	Hard cap	30,000,000.000000
		Total progress	
		Tokens sold	0.000000 CS
		Current cap	0.000000 USD
		Soft cap	25,000,000.000000 USD
		Hard cap	30,000,000.000000 USD

Proof of Concept Deployment Guide

Steps:

1. Get the latest version of the platform from the TokenD Developer Edition:
<https://github.com/tokend/developer-edition>

If you are already using TokenD Developer Edition, make sure to do *git checkout master* && *git pull*

2. Add `MODULE_SCHEME_NAME: "reit"` to `configs/client.js`
3. In `docker-compose.yml`, set `admin_client` to `1.5.0-reitrc.0`

```
```admin_client:
 image: tokend/admin-client:1.5.0-reitrc.0```
```

4. Run the TokenD Developer Edition following README.md, Bootstrapping your environment section.

These actions will result in running a clean instance of TokenD Real Estate PoC. In the upcoming releases, we are going to add initialization scripts that create test users and demo data.

\*Please be aware that the TokenD Real Estate source code we are sharing is a proof-of-concept grade. It can be used for the reference, but it is not production-ready.