LockOrSwap (Next step)

This application is to apply for the next Booster Grant available (5000$).

Improvement

This is the minimum list of improvements needed to apply for the grant.

Improvement (Major)

Scrypto

- Current Reward for suppliers needs to be token-specific (now it is the same for all the tokens)
- Current Extra-Reward for tokenizers needs to be token-specific (now it is the same for all the tokens)

Architecture

- Setup the new OCI library to fetching interest rate changes for then storing in a database
- Setup a new API (Kotlin) for serving the interest rate values (filtered by start/end epoch and resource address)
- Setup a new frontend library for showing the graph about the interest rate movement

Position Summary & Operational Panel (Frontend)

- Show tabular info about supplied/locked tokens amount
- Show graph info about locked tokens (Create and event during tokenize, store data and create API)
- 'Position Summary' and 'Operational Panel' refinement
- 'Epoch' should be converted from 'epochs' to a date
- 'Operational Panel' -> Clicking on the token should highligth the corresponding line in the table below
- 'PopUp' for each function -> Clicking on 'Supply' or 'Tokenized' or any other button should open a PopUp (//mock not ready yet)
- 'Position Summary' should highlight amount of tokens supplied and locked

Terraform

- Create (copy) terraform script for updating the website

Business

- Study Pendle, Notional whitepaper and prepare a Tokenizer whitepaper

Improvement (Medium)

Architecture

- how to move dinamically the reward/extra reward (//not ready) based on amount supplied/locked

Improvement (Minor)

- Suppliers should be able to add liquidity without withdrawing first

Improvement (Future)

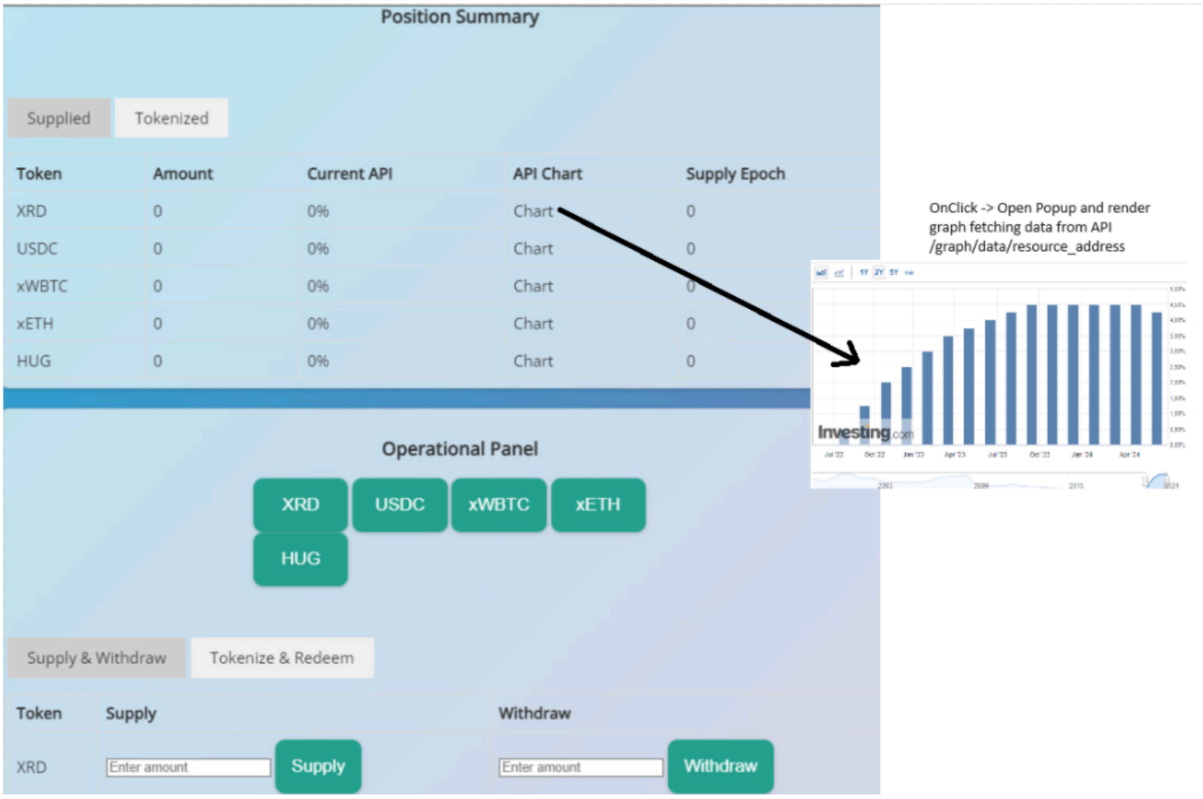
Architecture

- Create a function for Trade a locked position (This means burn the PT and YT and return a TKN-USDC for example, that is withdrawable directly from the platform)
- Create a function for Trade a locked indipendently a PT or a YT position
- Create an Order Book for trading PT or YT separately

Mock (Frontend)

Here we present a mock interface to be defined before continuing the project

Position Summary, Tab Supplied (Frontend)

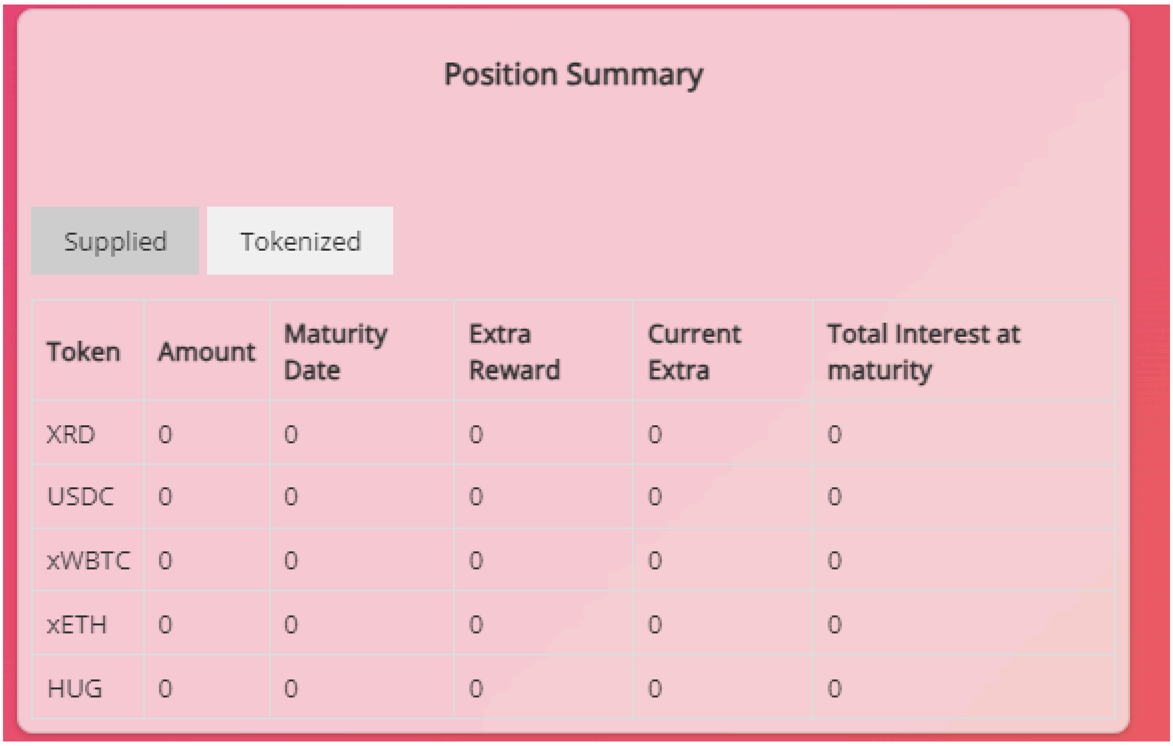


Token	Amount	Current API	API Chart	Supply Epoch
XRD	0	0%	Chart	0
USDC	0	0%	Chart	0
xWBTC	0	0%	Chart	0
xETH	0	0%	Chart	0
HUG	0	0%	Chart	0

OnClick -> Open Popup and render graph fetching data from API /graph/data/resource_address

Token	Supply		Withdraw	
XRD	Enter amount	Supply	Enter amount	Withdraw

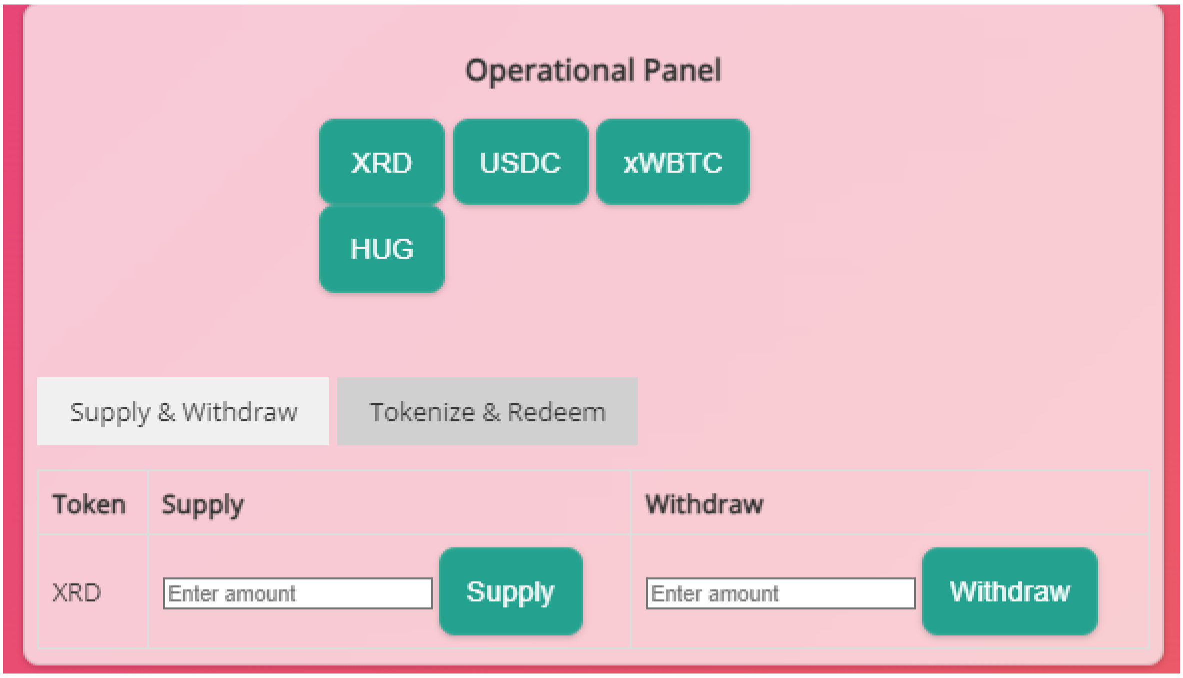
Position Summary, Tab Tokenized (Frontend)



Position Summary

Token	Amount	Maturity Date	Extra Reward	Current Extra	Total Interest at maturity
XRD	0	0	0	0	0
USDC	0	0	0	0	0
xWBTC	0	0	0	0	0
xETH	0	0	0	0	0
HUG	0	0	0	0	0

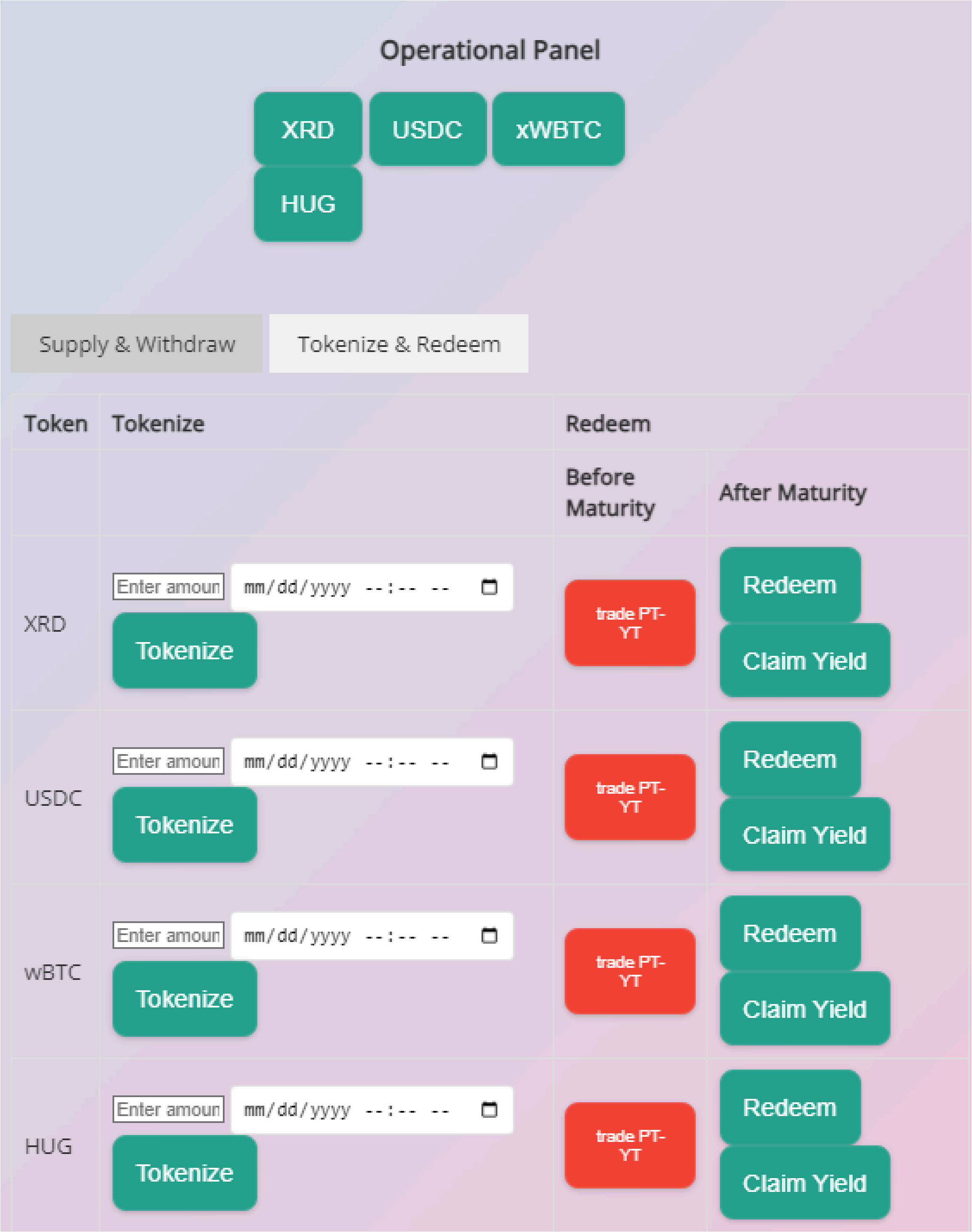
Operational Panel, Tab Supply & Withdraw (Frontend)



Operational Panel

Token	Supply	Withdraw
XRD	Enter amount Supply	Enter amount Withdraw

Operational Panel, Tab Tokenize & Reedem (Frontend)



Token	Tokenize	Redeem	
		Before Maturity	After Maturity
XRD	Enter amoun mm/dd/yyyy --:-- -- Tokenize	trade PT-YT	Redeem Claim Yield
USDC	Enter amoun mm/dd/yyyy --:-- -- Tokenize	trade PT-YT	Redeem Claim Yield
wBTC	Enter amoun mm/dd/yyyy --:-- -- Tokenize	trade PT-YT	Redeem Claim Yield
HUG	Enter amoun mm/dd/yyyy --:-- -- Tokenize	trade PT-YT	Redeem Claim Yield

Current Bug

- Extra amount coming from the yield claimed is not updated back as 'Liquidity Data' (This means that this amount remain locked in the contract)

Let's finally have a look at the Online demo dApp

You can also try the deployed dApp here https://zerocollateral.eu/