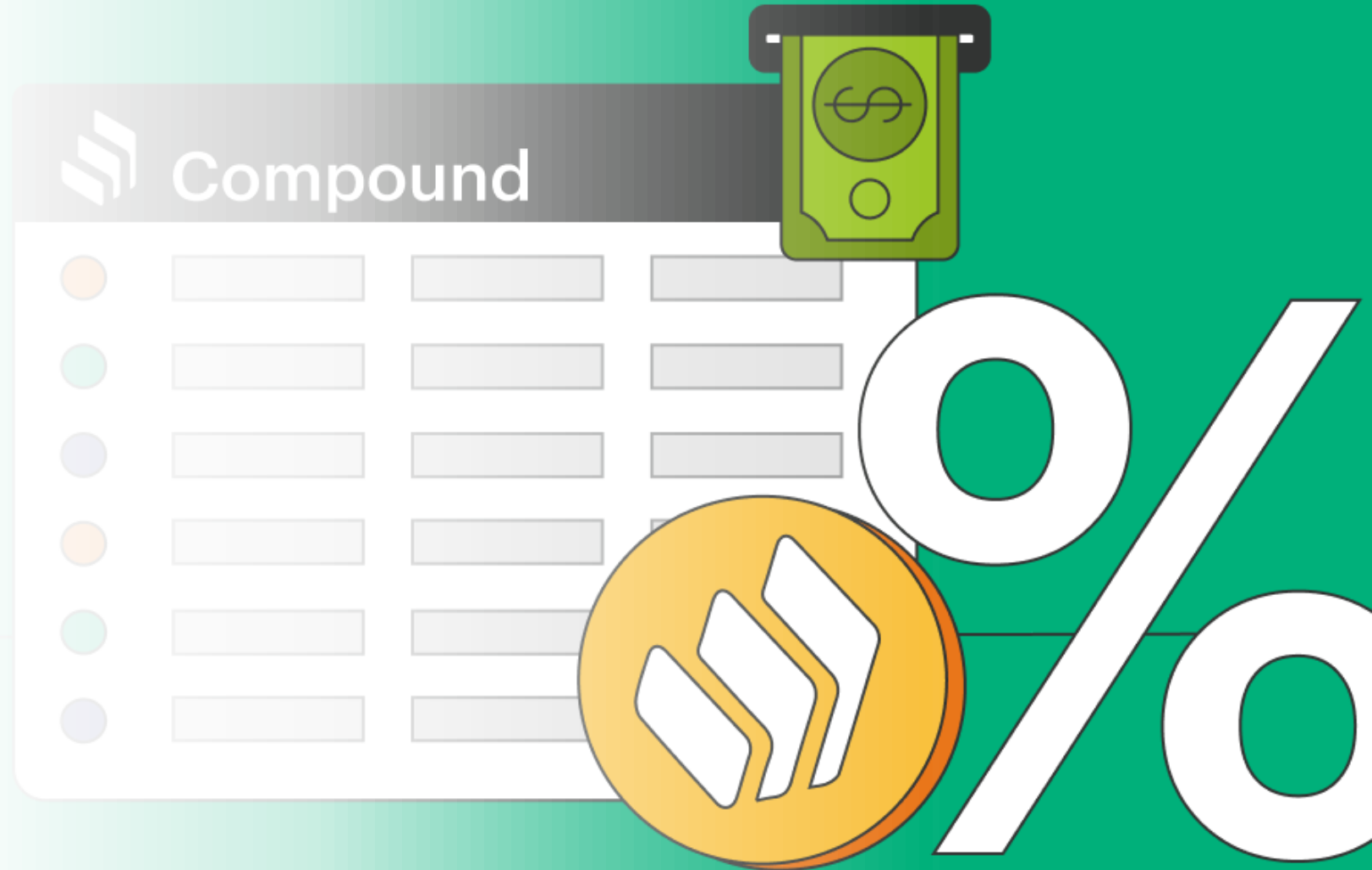


# Deploy Compound Protocol

Corso di

FINANCIAL COMPUTING E  
TECNOLOGIE PER L'HIGH  
FREQUENCY TRADING

PIERLUIGI LIGUORI  
FABIANO PRIORE  
MARIKA PIA SALVATO



master compound-protocol / contracts /

arjun-io and kbrizzle [RFP12] CToken Cleanup (#152)

# Repository Github del Protocollo

Governance [RFP12] CToken Cleanup (#152)

BasicRateModelV2.sol [RFP12] CToken Cleanup (#152)

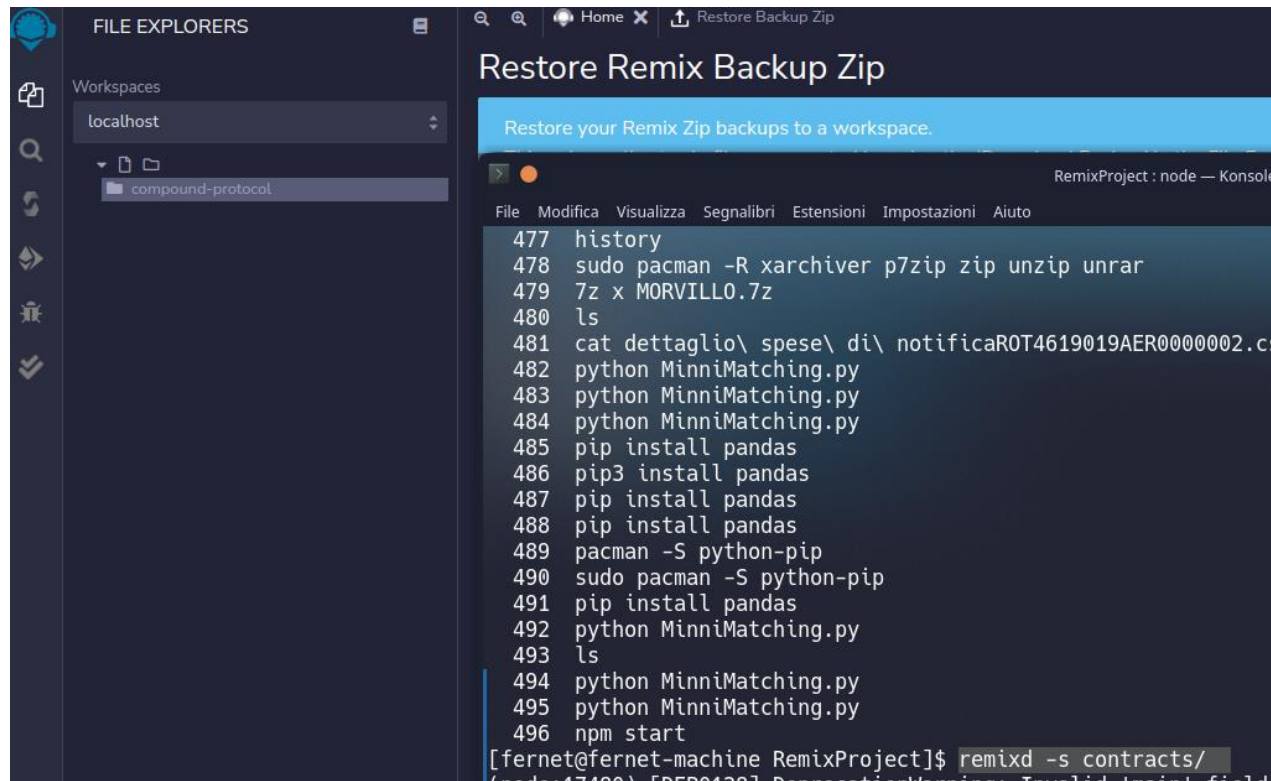
CDaiDelegate.sol [RFP12] CToken Cleanup (#152)

CErc20.sol [RFP12] CToken Cleanup (#152)

CErc20Delegate.sol [RFP12] CToken Cleanup (#152)

CErc20Delegator.sol [RFP12] CToken Cleanup (#152)

# Collegamento in locale con Remix



```
FILE EXPLORERS
Workspaces
localhost
  compound-protocol

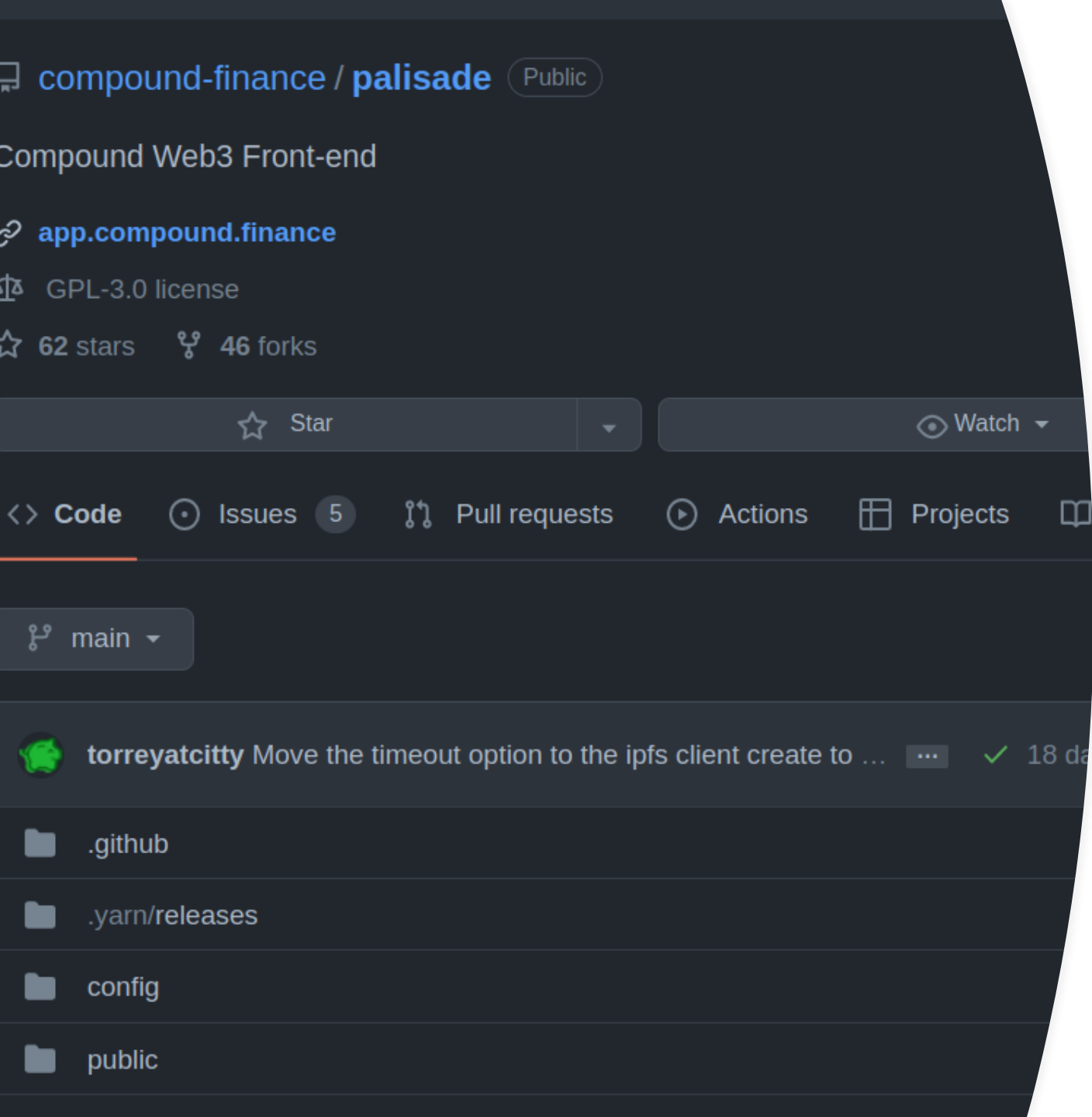
Restore Remix Backup Zip
Restore your Remix Zip backups to a workspace.

RemixProject : node — Konsole
File Modifica Visualizza Segnalibri Estensioni Impostazioni Aiuto
477 history
478 sudo pacman -R xarchiver p7zip zip unzip unrar
479 7z x MORVILLO.7z
480 ls
481 cat dettaglio\ spese\ di\ notificaROT4619019AER0000002.c
482 python MinniMatching.py
483 python MinniMatching.py
484 python MinniMatching.py
485 pip install pandas
486 pip3 install pandas
487 pip install pandas
488 pip install pandas
489 pacman -S python-pip
490 sudo pacman -S python-pip
491 pip install pandas
492 python MinniMatching.py
493 ls
494 python MinniMatching.py
495 python MinniMatching.py
496 npm start
[fernet@fernet-machine RemixProject]$ remixd -s contracts/
```

Dopo aver scaricato il codice del protocollo, lanciare da terminale:

```
npm install -g remixd
```

```
remixd -s contracts/
```



# Repository GitHub Frontend

Clonazione del repository in locale:

- git clone <https://github.com/compound-finance/palisade.git>
- cd palisade/
- yarn install
- yarn install --ignore-platform
- yarn watch-i18n
- yarn watch-css
- yarn start

## DETAILS

NAME\*

MyFirstProject

SAVE CHANGES

## KEYS

# Account Infura

PROJECT ID  
ca4d1c41e3aa460aa60aa5b200403ba5 

Procurarsi una key di progetto da sostituire in [palisade/config/env/development.json](#)

0bd83ca54147425c9591cdeed5124317 

# Note

- Vengono rideployate le funzionalità nella dashboard di <https://app.compound.finance/>
- Non vengono trattati i contratti relativi alla governance
- La TestNet usata è Kovan
- Alcuni contratti che riguardano oracoli e tokens già presenti non vengono rimpiazzati
- Ogni contratto rideployato deve essere rimpiazzato nel file [palisade/node\\_modules/compound-config/networks/kovan.json](#)



## Come avviare la nostra implementazione

---

- Sostituire il file `kovan.json` fornito in allegato
- Avviare il frontend con `npm start`



Supply Balance

€0

Net APY



...

Borrow Balance



€0

Limit 0%

## Supply Markets

Asset	APY	WalletCollateral
 Aave ...	5.05%	0 AAVE <input type="checkbox"/>
 Basic ...	0.08%	0 BAT <input type="checkbox"/>

## Borrow Markets

Asset	APY	WalletLiquidity
 Aave ...	15.23%	0 AAVE €650k
 Basic ...	3.37%	0 BAT€50.8...

# Istruzioni per deployare il compound protocol

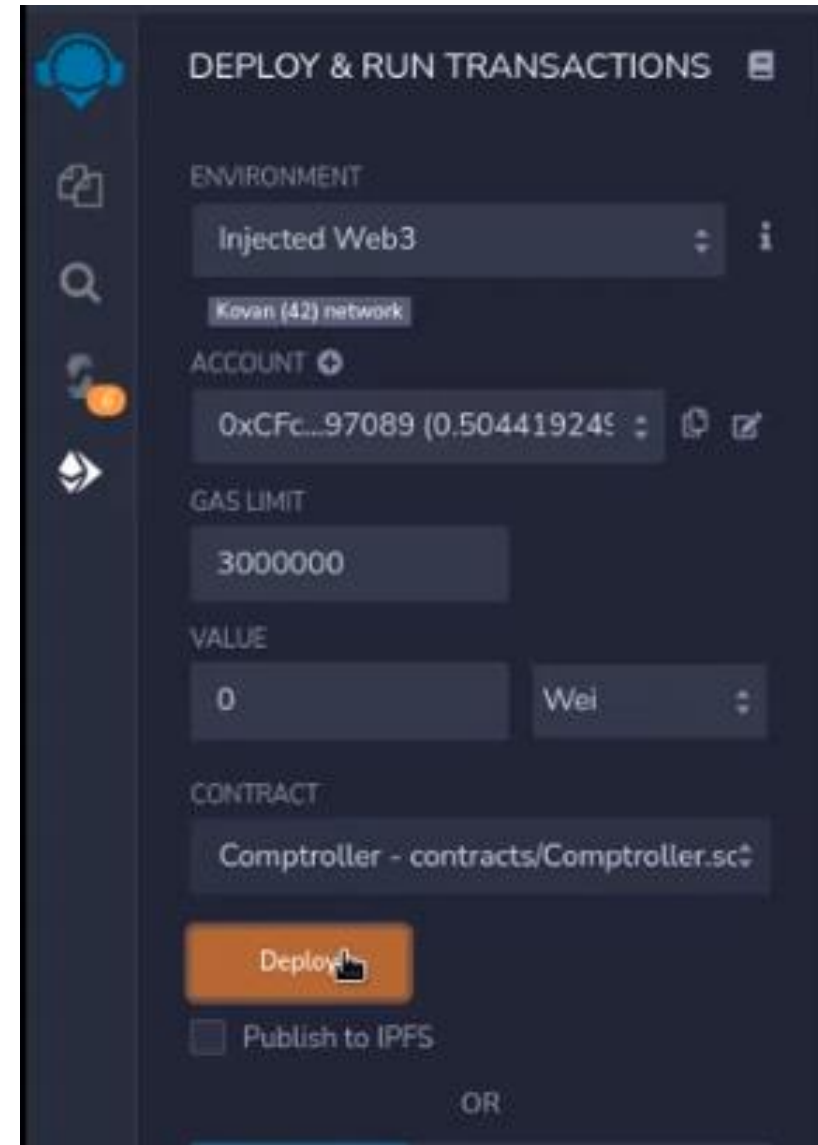


```
/**
 * @notice Return the address of the COMP token
 * @return The address of COMP
 */
function getCompAddress() virtual public view returns (address)
    return 0xc00e94Cb662C3520282E6f5717214004A7f26888;
}
}
```

- Copiare l'indirizzo del Contratto **Comp** (ricavabile dal file *kovan.json* corrente)
- Incollarlo nella funzione **getComAddress()** nel codice di **Comptroller.sol**

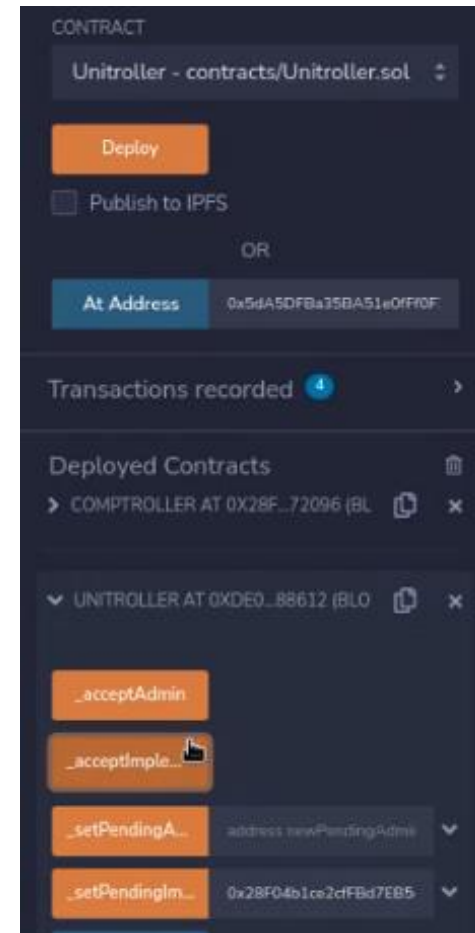
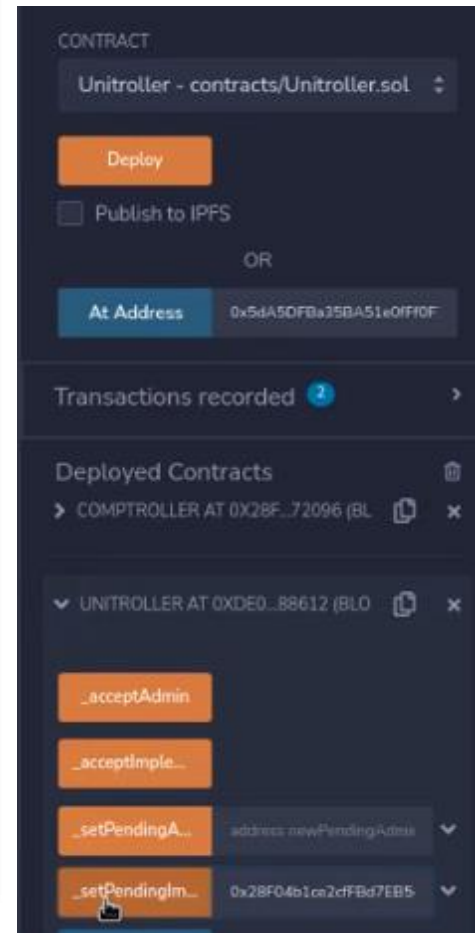
# Deploy di Comptroller.sol

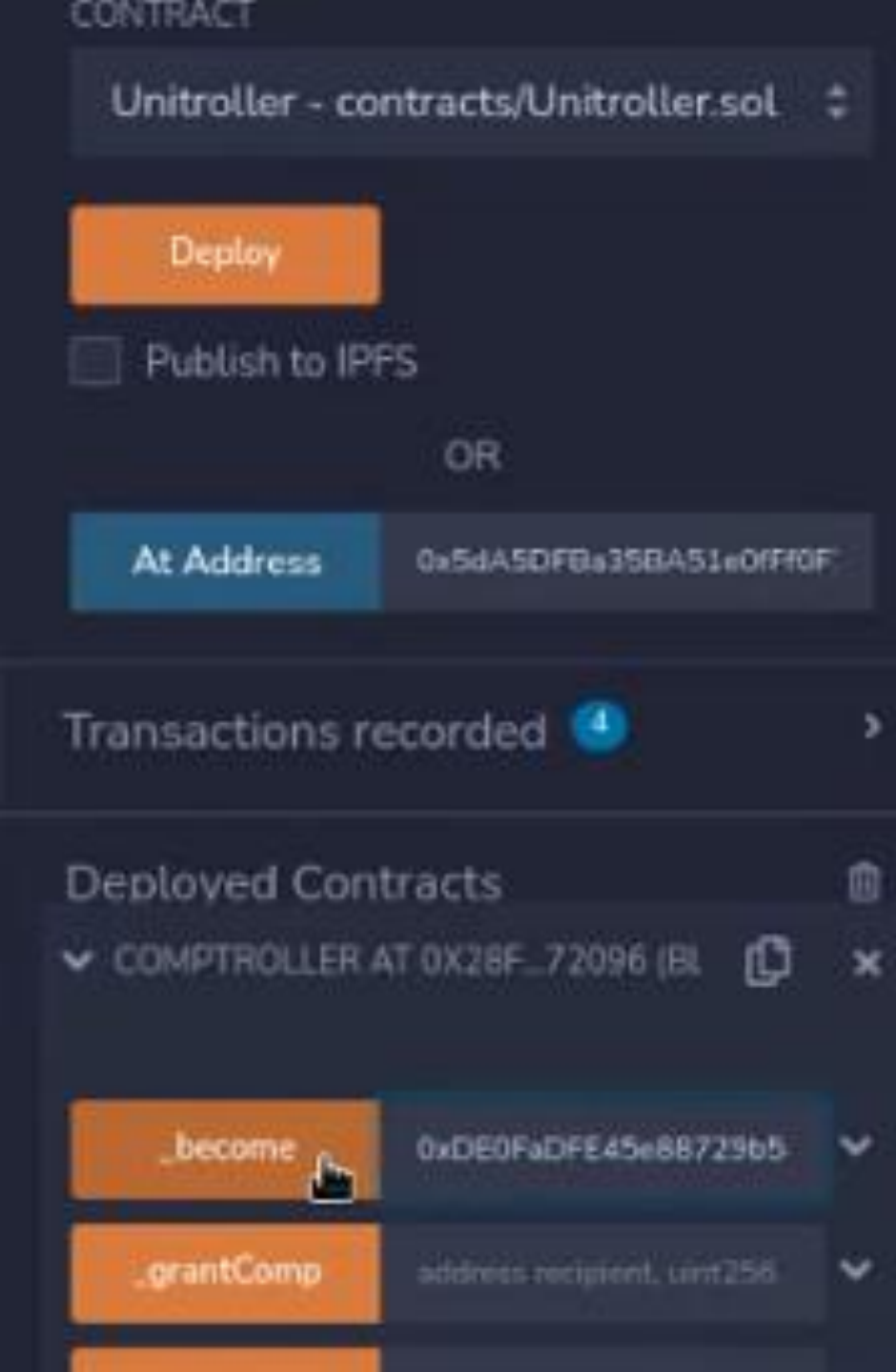
Da Remix, compilare [Comptroller.sol](#) e, dopo aver collegato *MetaMask*, deployare sulla TestNet pubblica



# Unitroller.sol

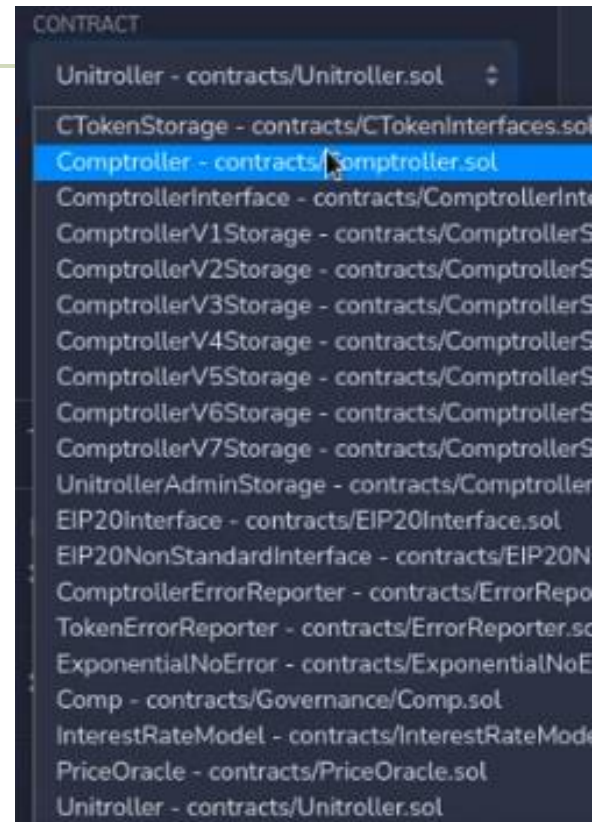
- Fare il deploy di **Unitroller.sol**
- Chiamare la funzione **\_setPendingImplementation()** inserendo come parametro l'indirizzo del Comptroller precedentemente creato
- Chiamare la funzione **\_acceptImplementation()**





# Unitroller.sol (II)

- Copiare l'indirizzo di Unitroller e darlo in input alla funzione `_become()` del Comptroller
- Reimportare l'Unitroller copiando il suo indirizzo sulla voce 'At Address' ma questa volta mascherandolo con l'implementazione di un Comptroller



# Oracolo



aggiungere oracolo (l'indirizzo si trova nel file *kovan.json*) a Unitroller (Comptroller) tramite `_setPriceOracle()`

# Deploy CEther

- Inserisci l'indirizzo del **Comptroller** creato
- Inserisci l'indirizzo dell' **InterestRateModel** esistente
- Specificare come **Admin** l'indirizzo del proprio Wallet
- Riempire gli altri campi come in figura

DEPLOY & RUN TRANSACTIONS

Kovan (42) network

ACCOUNT

0xCFc...97089 (0.4911412 ETH)

GAS LIMIT

3000000

VALUE

0 Wei

CONTRACT

CEther - contracts/CEther.sol

DEPLOY

COMPTROLLER: 0xDE0FaDfE45e8B729b58F

INTERESTRATEMODEL: 0xC18496A3a4Eee1

INITIALEXCHANGERATEMANTISSA: 200000

NAME: Compound ETH

SYMBOL: cETH

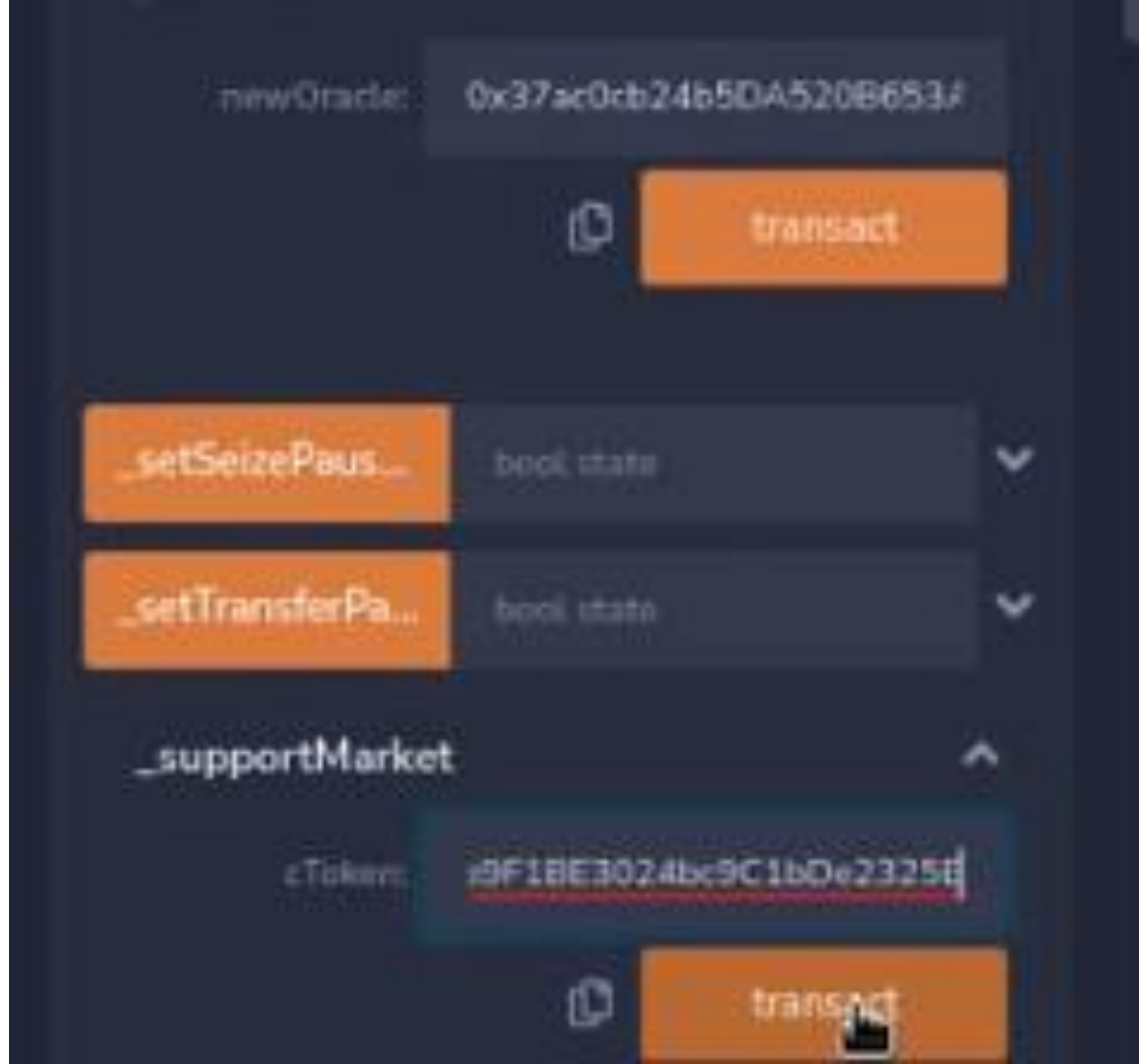
DECIMALS: 8

ADMIN: BB067303E9D1783B697089

transact

# Aggiungi cETH all' Unitroller

da Unitroller (Comptroller),  
chiamata `_supportMarket()` per  
cETH (appena creato)



# **\_setCollateralFactor()**

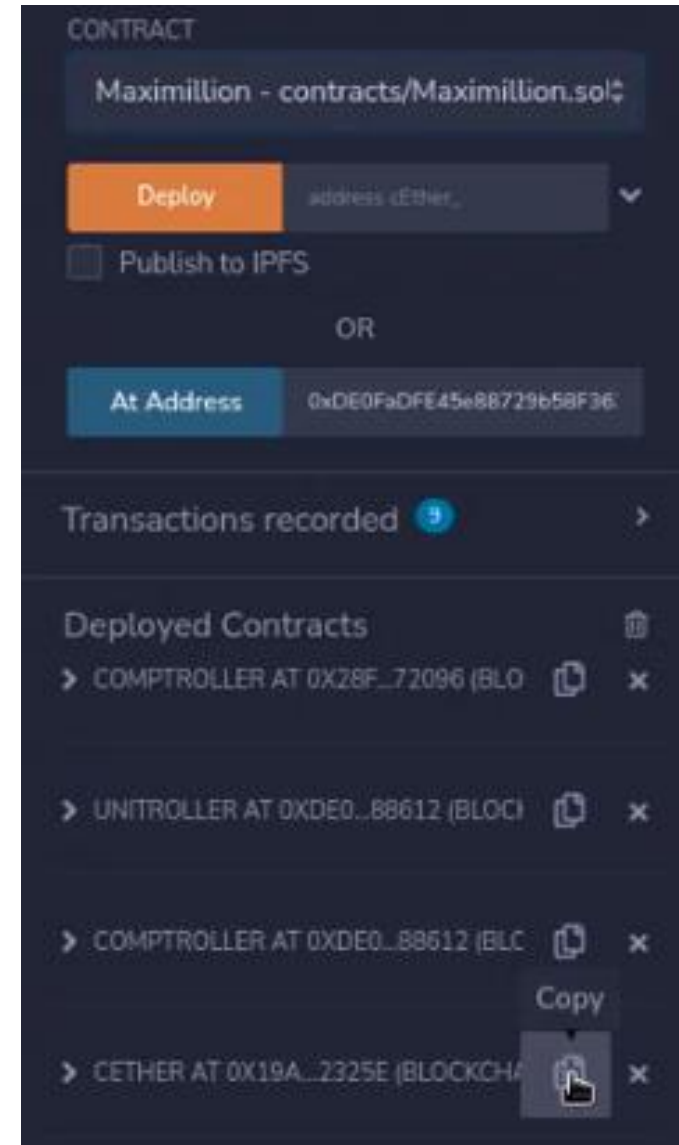
- da Unitroller (Comptroller), chiama **\_setCollateralFactor()** per cETH con valore  $75 \cdot 10^{18}$





# Maximillion

- Deployare un nuovo contratto da [Maximillion.sol](#)
- Come parametro, inserire l'indirizzo di [CEther](#)
- Maximillion è essenziale per la parte di repay e borrow



# Aggiunta dei Token

---



# CToken

- Copiare indirizzo di un **Token** già esistente (o crearne uno nuovo) da aggiungere al protocollo
- Accertarsi che provenga dall'implementazione **CErc20Delegator** o **CErc20Immutable**
- In figura viene mostrato l'esempio di deploy del **cToken** relativo a **BAT**
- **UNDERLYING\_**: bisogna specificare l'indirizzo del token da aggiungere
- Altri valori sono prelevabili da **kovan.json**

utable - contracts/CErc20Immutable.sol

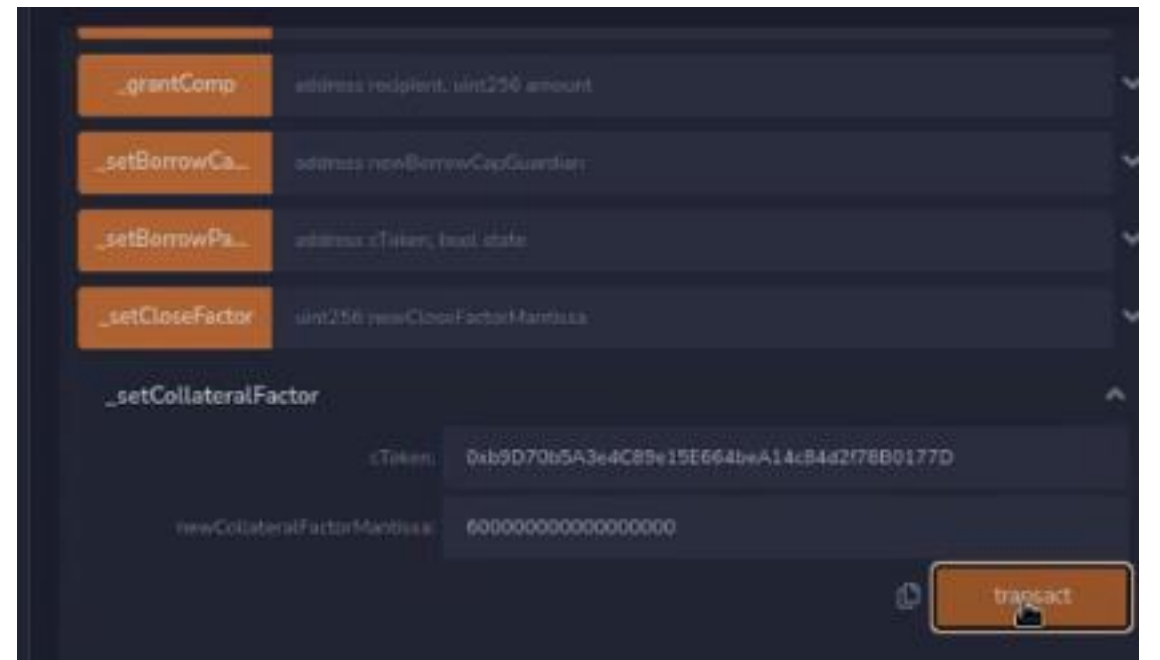
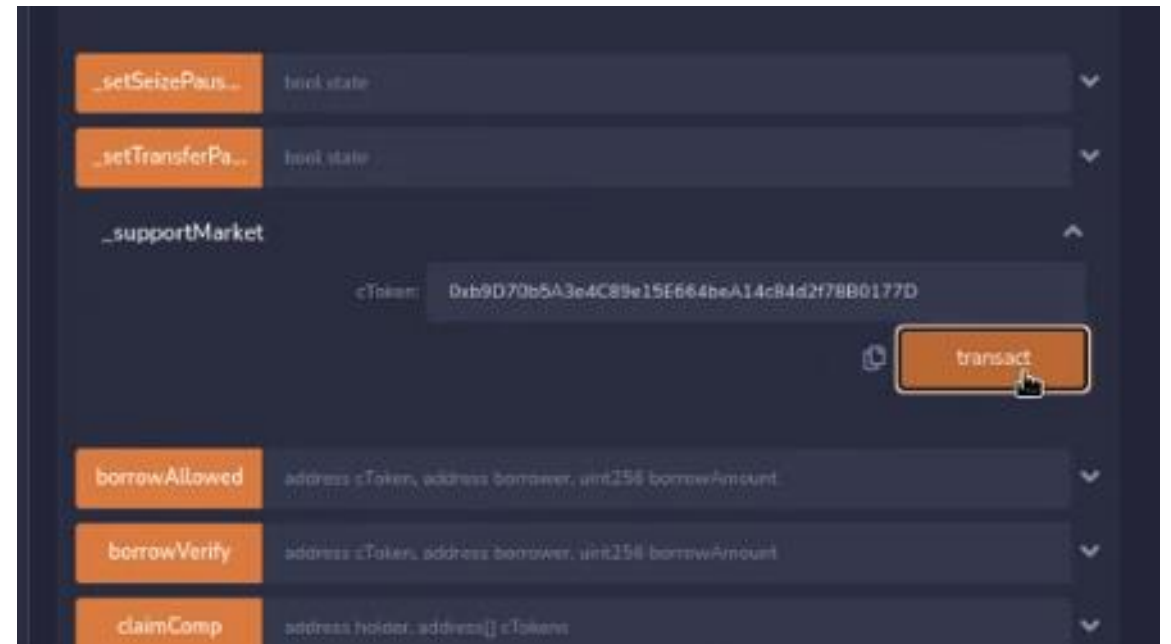
UNDERLYING_	0x482dC96B08111CB875109B075A40681E48aE02Cd
COMPTROLLER_	0xDE0FaDfE45e88723658F363ba5e542EcCE788612
INTERESTRATEMODEL_	0x7BDf14574c844cD9B75073f91Bb1aEAb44a80529
MANAGERATEMANTISSA_	20000000000000000000000000000000
NAME_	Compound BAT
SYMBOL_	cBAT
DECIMALS_	8
ADMIN_	0xCFc9195eaC52e74650BB067303E9D17838697089

transact

da **Unitroller** (Comptroller):

chiama **\_supportMarket** per il cToken appena creato

chiama **\_setCollateralFactor** per il cToken





# Esecuzione

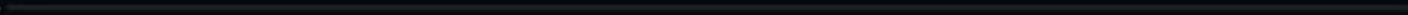
---






Supply Balance  
\$0.00000000

Net APY  
...

Borrow Balance  
\$0.00000000

Borrow Limit 0%  \$0.00

## Supply Markets

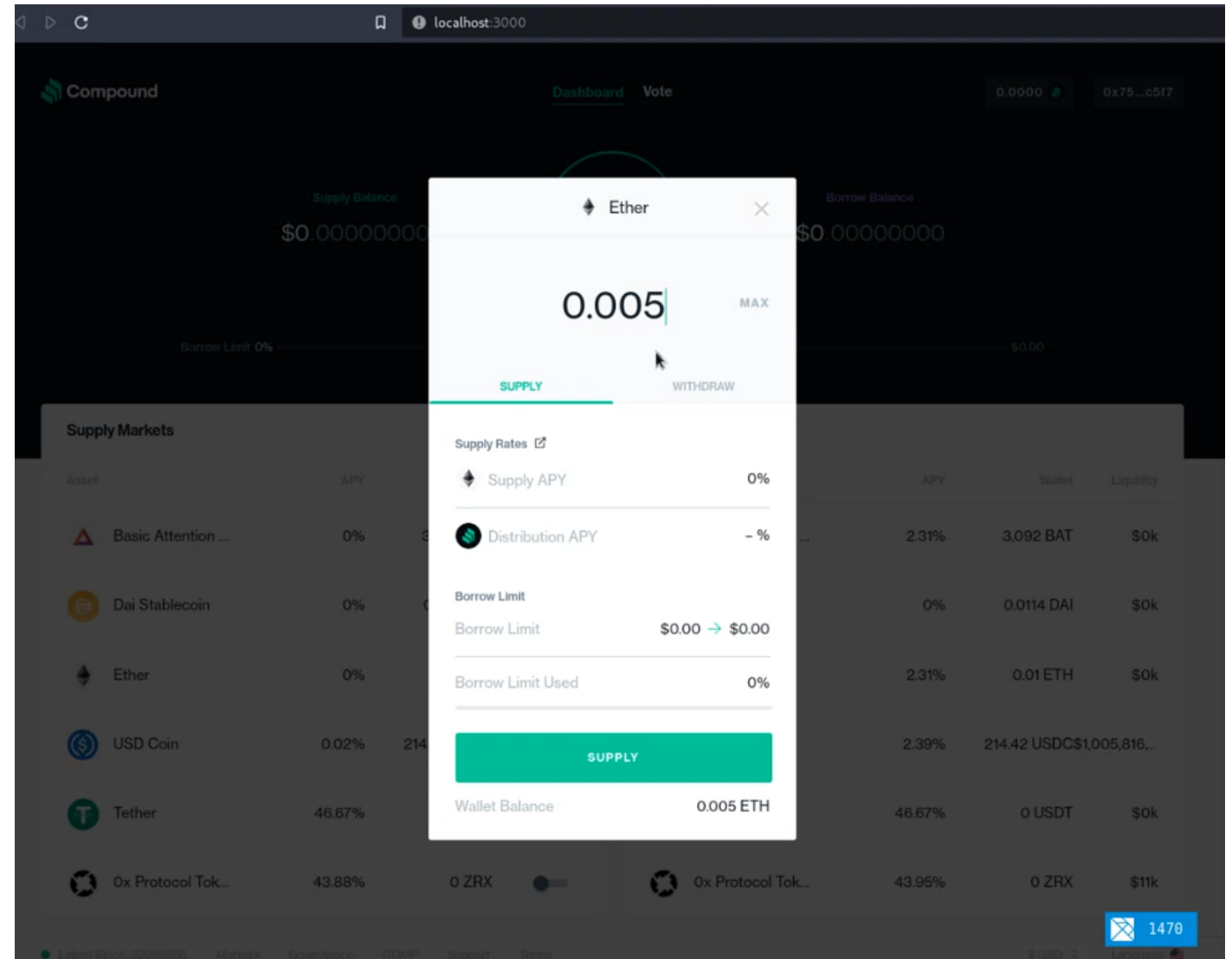
Asset	APY	Wallet	Collateral
 Basic Attention ...	0%	3,092 BAT	<input checked="" type="checkbox"/>
 Dai Stablecoin	0%	0.0114 DAI	<input checked="" type="checkbox"/>
 Ether	0%	0.01 ETH	<input checked="" type="checkbox"/>

## Borrow Markets

Asset	APY	Wallet	Liquidity
 Basic Attention ...	2.31%	3,092 BAT	\$0k
 Dai Stablecoin	0%	0.0114 DAI	\$0k
 Ether	2.31%	0.01 ETH	\$0k

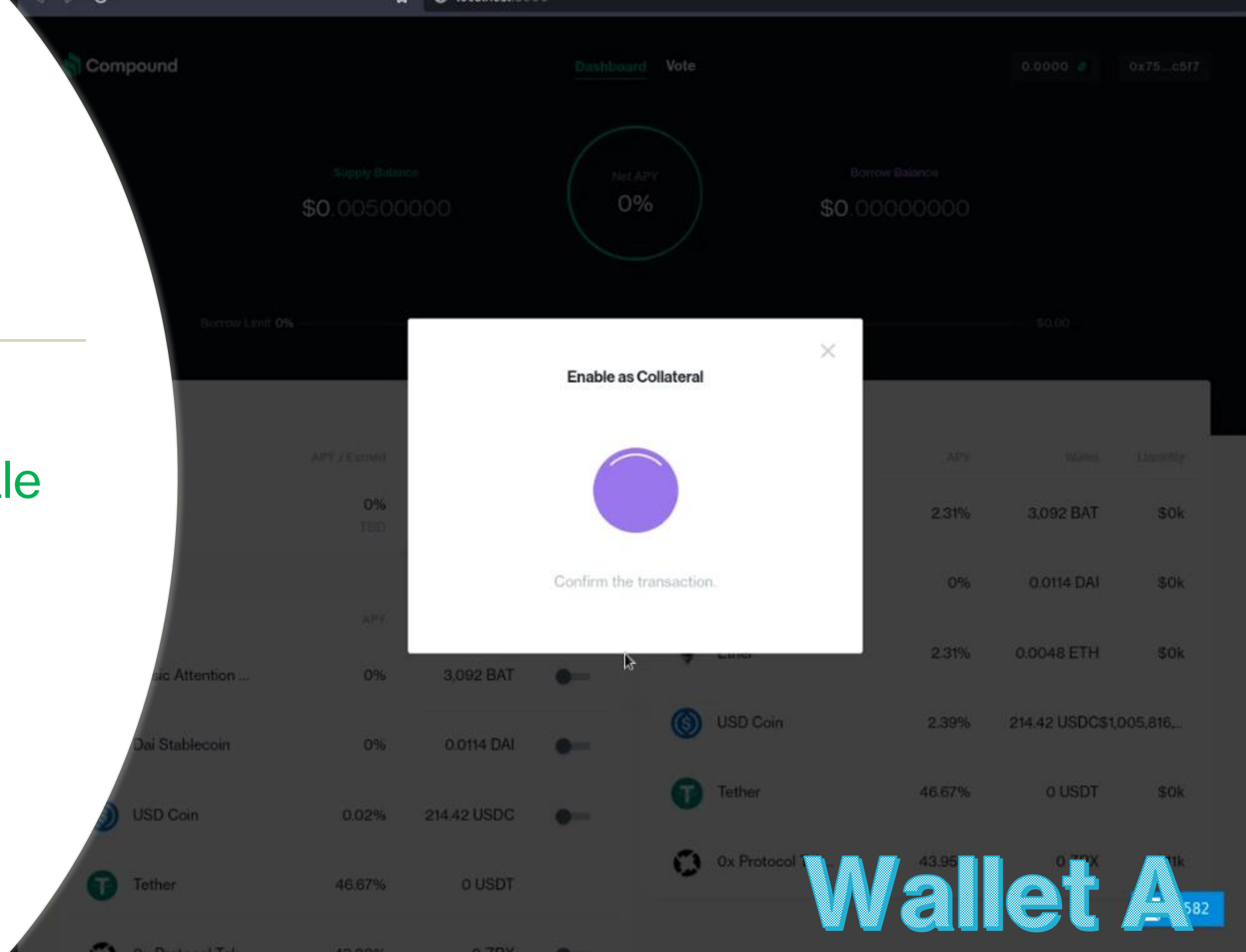
Wallet A

- Supply di Ether



Wallet A

- Abilita Ether come **collaterale**





Compound

DashboardVote

0.00000x75...c5f7

Supply Balance

\$0.00500000

Net APY

0%

Borrow Balance

\$0.00000000

Borrow Limit 0%

\$0.00

Supply

Asset	APY / Earned	Balance	Collateral
Ether	0% TBD	\$0.00 0.005 ETH	

All Markets ▴

Asset	APY	Wallet	Collateral
Basic Attention ...	0%	3,092 BAT	
Dai Stablecoin	0%	0.0114 DAI	
USD Coin	0.02%	214.42 USDC	
Tether	46.67%	0 USDT	

Borrow Markets

Asset	APY	Wallet	Liquidity
Basic Attention ...	2.31%	3,092 BAT	\$0k
Dai Stablecoin	0%	0.0114 DAI	\$0k
Ether	2.31%	0.0047 ETH	\$0k
USD Coin	2.39%	214.42 USDC\$1,005,816,...	
Tether	46.67%	0 USDT	\$0k
Ox Protocol Tok...	43.95%	0 ZRX	\$11k

1854

Wallet A

- Supply di DAI
- Abilita DAI come collaterale

Compound | Dashboard | Remix - Ethereum IDE | Contract Address 0xb... | CERC20Immutable | Confirmed transaction | Transaction 36 confirmed! View on Etherscan

localhost:3000

Compound Dashboard Vote 0.0000 0x75...c5f7

Supply Balance \$0.00502735 Net APY 0% Borrow Balance \$0.00000000

Borrow Limit 0% \$0.00

Supply			
Asset	APY / Earned	Balance	Collateral
Dai Stablecoin	0% TBD	\$0.00 0.0099 DAI	<input checked="" type="checkbox"/>
Ether	0% TBD	\$0.00 0.005 ETH	<input checked="" type="checkbox"/>

Borrow Markets			
Asset	APY	Wallet	Liquidity
Basic Attention ...	2.31%	3,092 BAT	\$0k
Dai Stablecoin	0%	0.0014 DAI	\$0k
Ether	2.31%	0.0044 ETH	\$0k
USD Coin	2.39%	214.42 USDC	\$1,005,816...
Tether	46.67%	0 USDT	\$0k
Ox Protocol Tok...	43.95%	0 ZRX	\$11k

All Markets			
Asset	APY	Wallet	Collateral
Basic Attention ...	0%	3,092 BAT	<input type="checkbox"/>
USD Coin	0.02%	214.42 USDC	<input type="checkbox"/>
Tether	46.67%	0 USDT	<input type="checkbox"/>

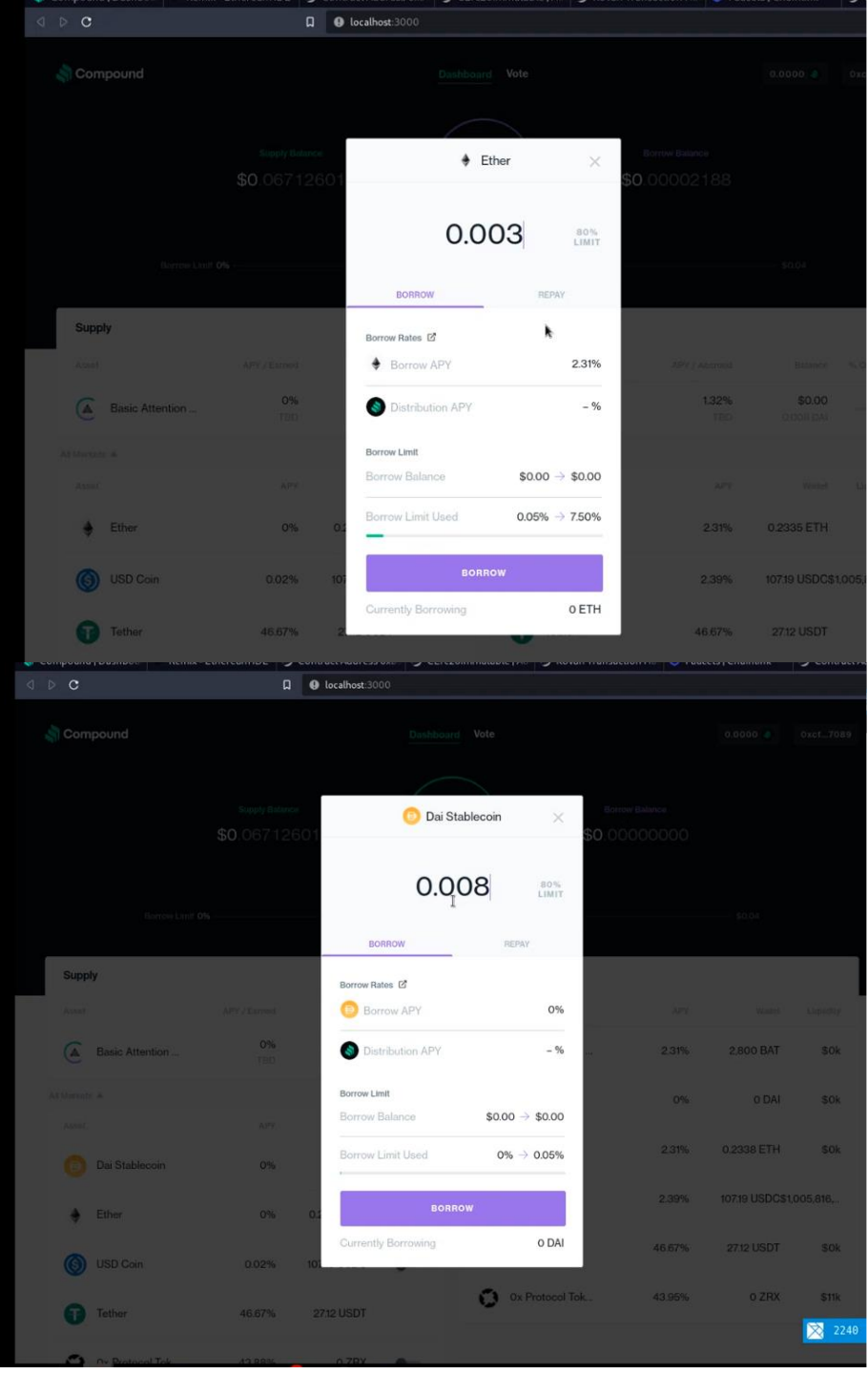
2122

Wallet A

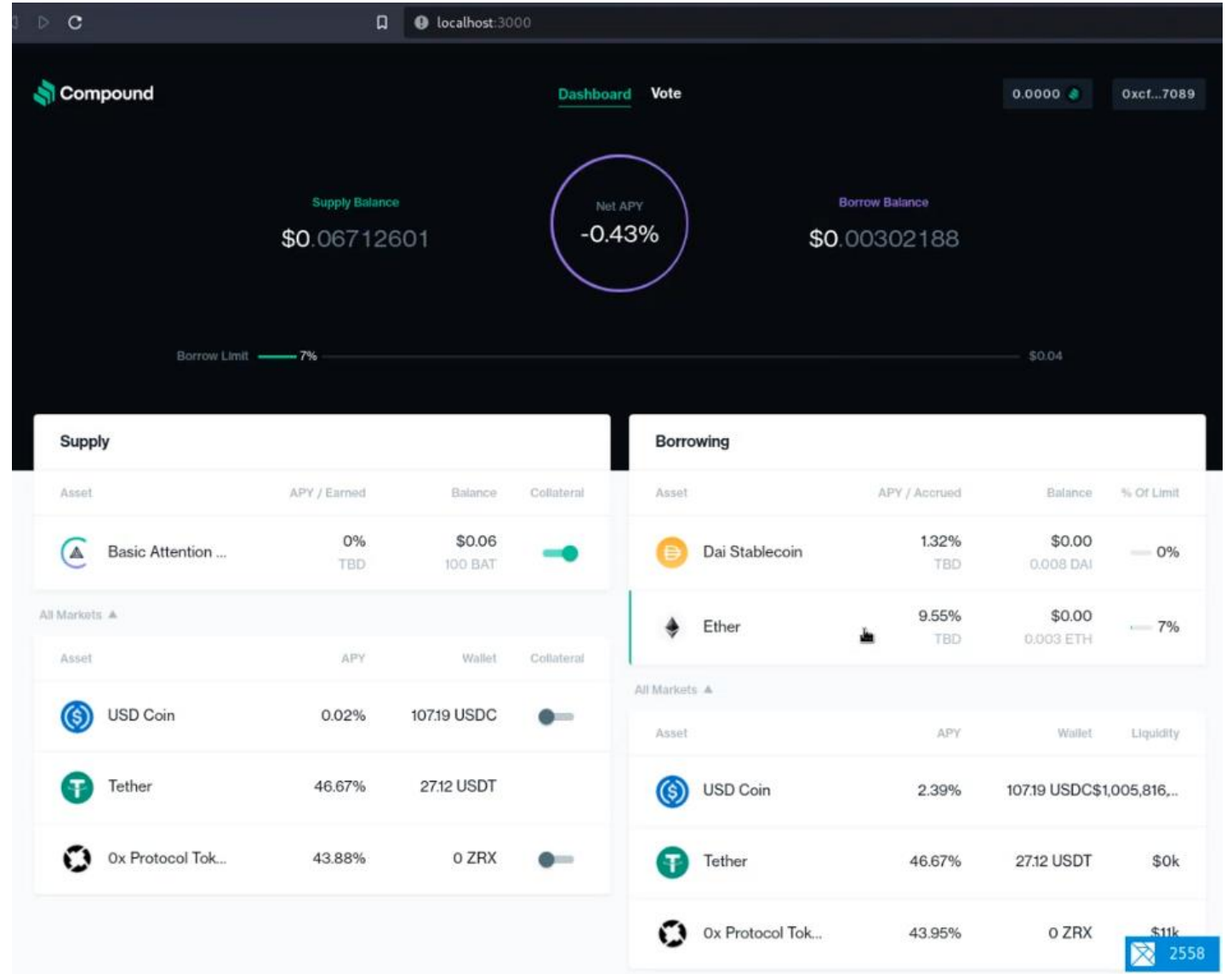
*NB: prima di fare operazioni di Borrow, Repay, e withdraw assicurarsi che ci sia liquidità*

- Borrow di DAI
- Borrow di Ether

Wallet B

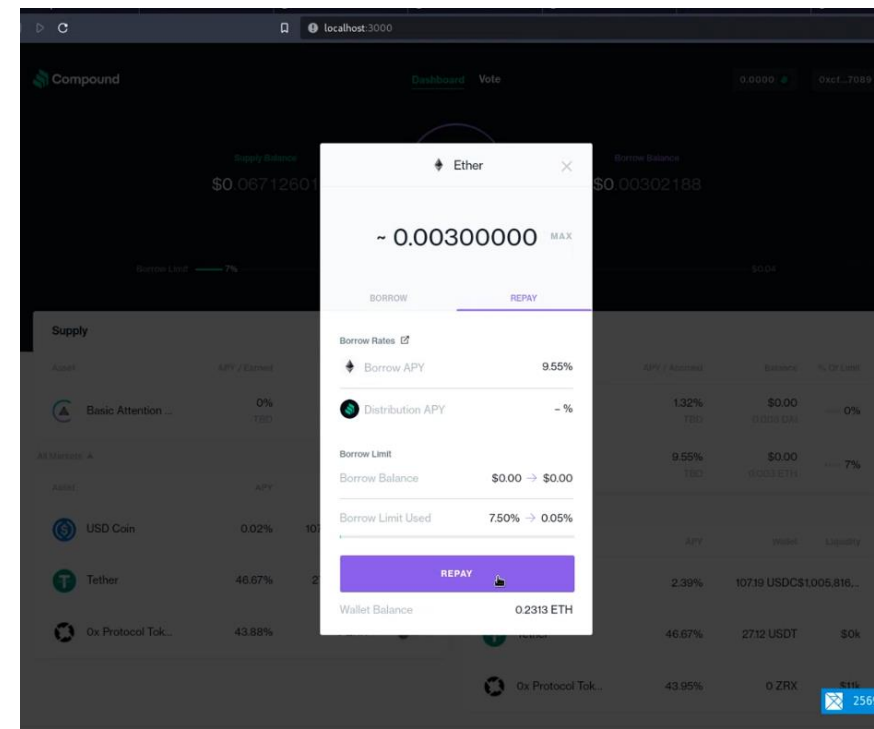
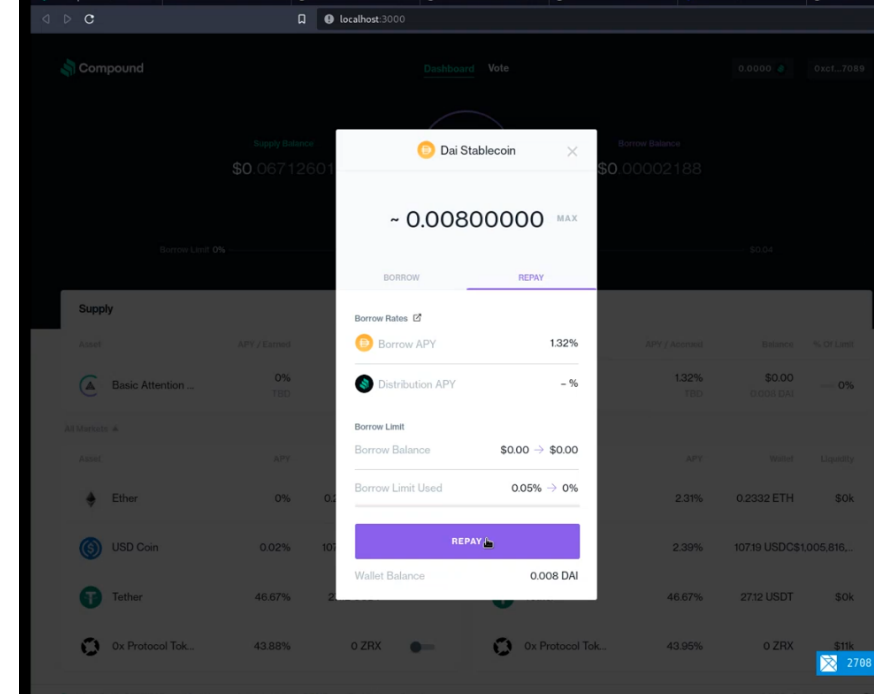


# Wallet B



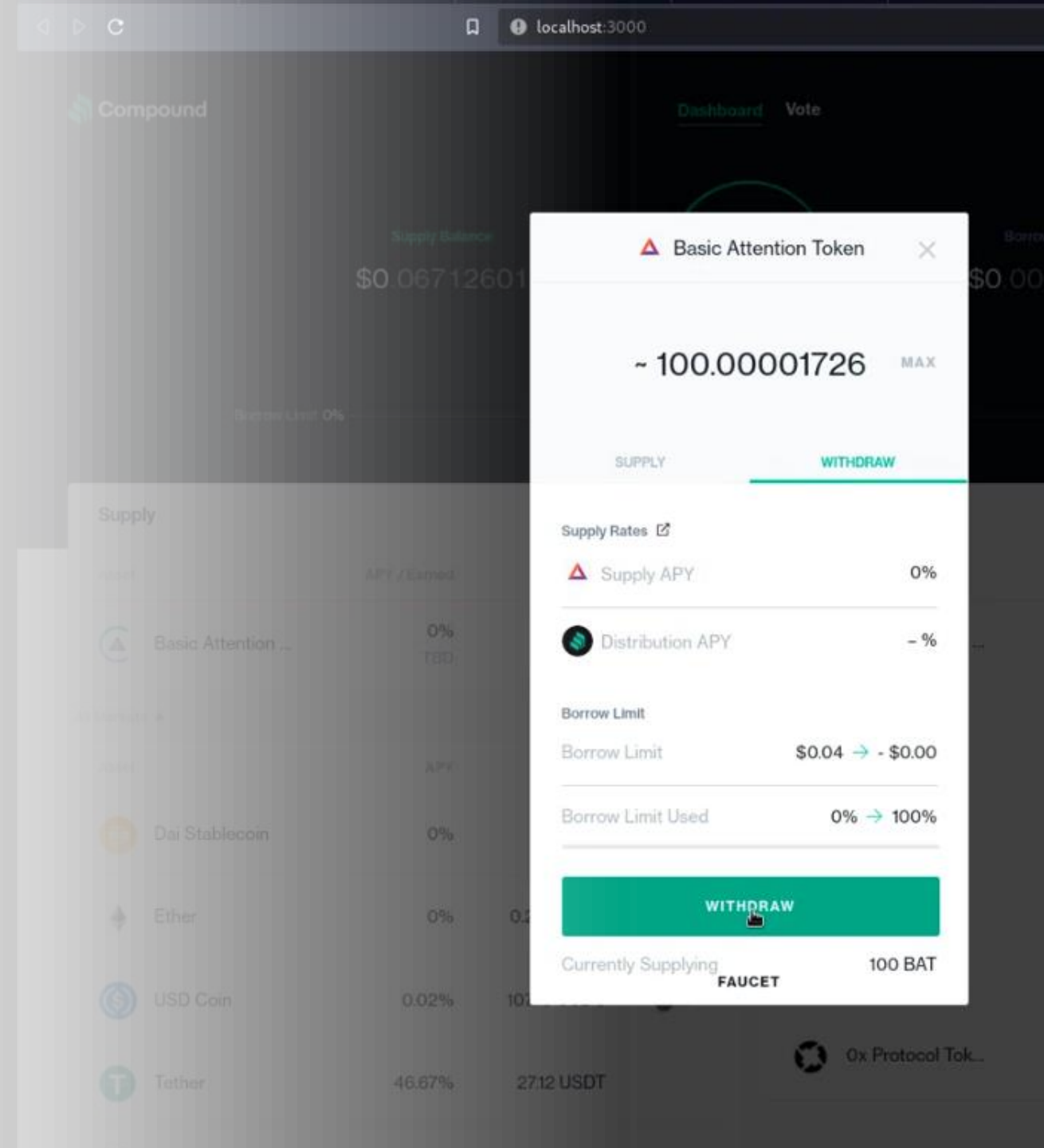
- Repay di Ether
- Repay di DAI

# Wallet B



- Withdraw di BAT

Wallet B



- Disattiva BAT come **collaterale**

# Wallet B

