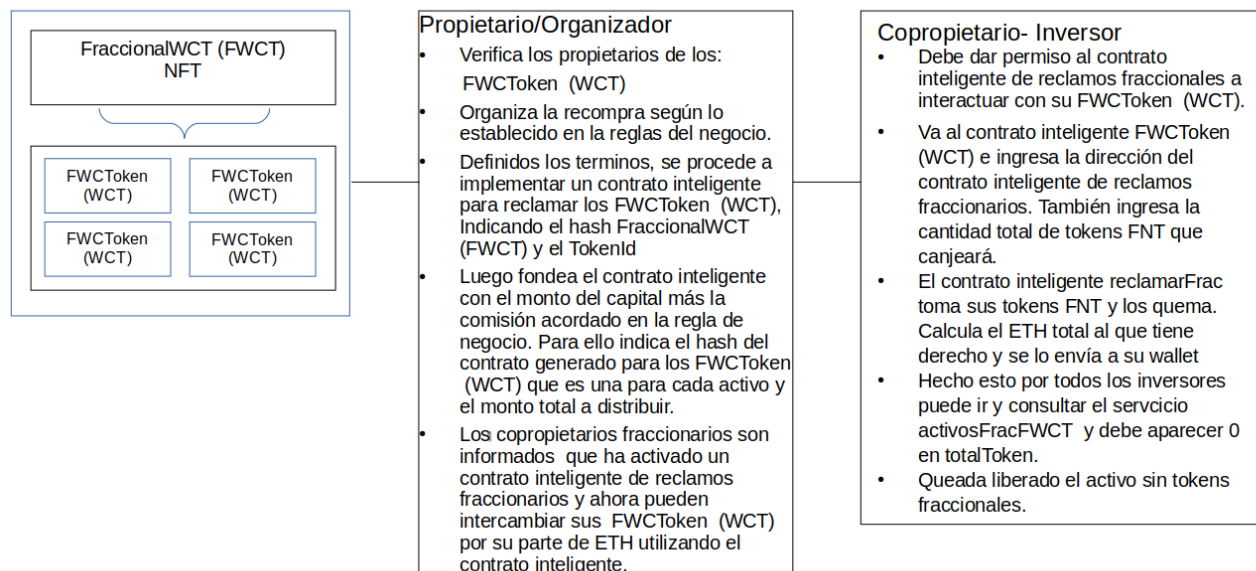


Con esto completamos lo que había denominado 1ra fase y podemos cubrir hasta los inversores y su envío de tokens.

La segunda fase es la siguiente:



Aquí se hace lo siguiente:

El propietario u organizador anuncia la recompra y para ello implementa un contrato inteligente para que los co-propietarios reclamen su inversión y su comisión según la reglas de negocio.

En este contrato para reclamar inversión el propietario coloca el monto total de inversión + comisiones.

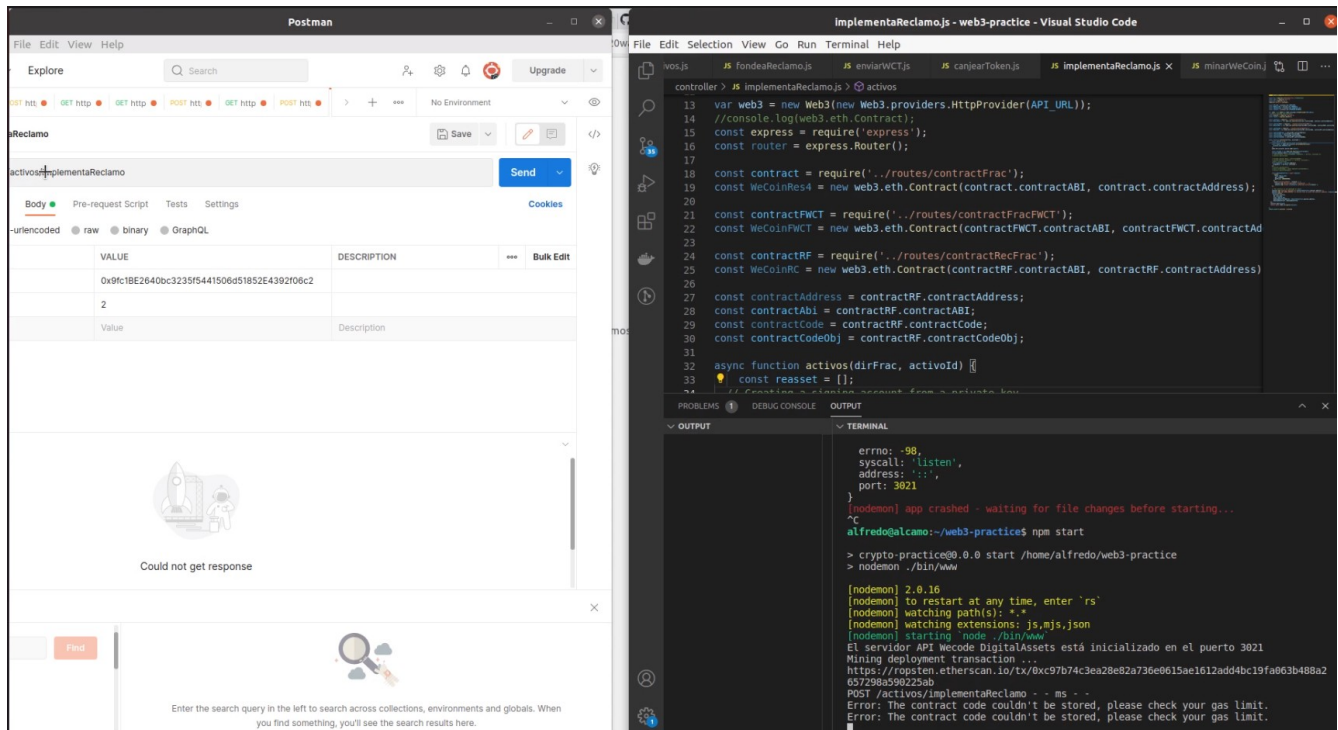
Luego cada uno de los co-propietarios reclaman su inversión y comisión y una vez hecho se libera el token y se quema.

Al hacerlo todos el token NFT queda libre sin co-propietarios y se puede vender o quemar.

Esta segunda parte esta programado y probado lo del contrato de reclamar inversion.

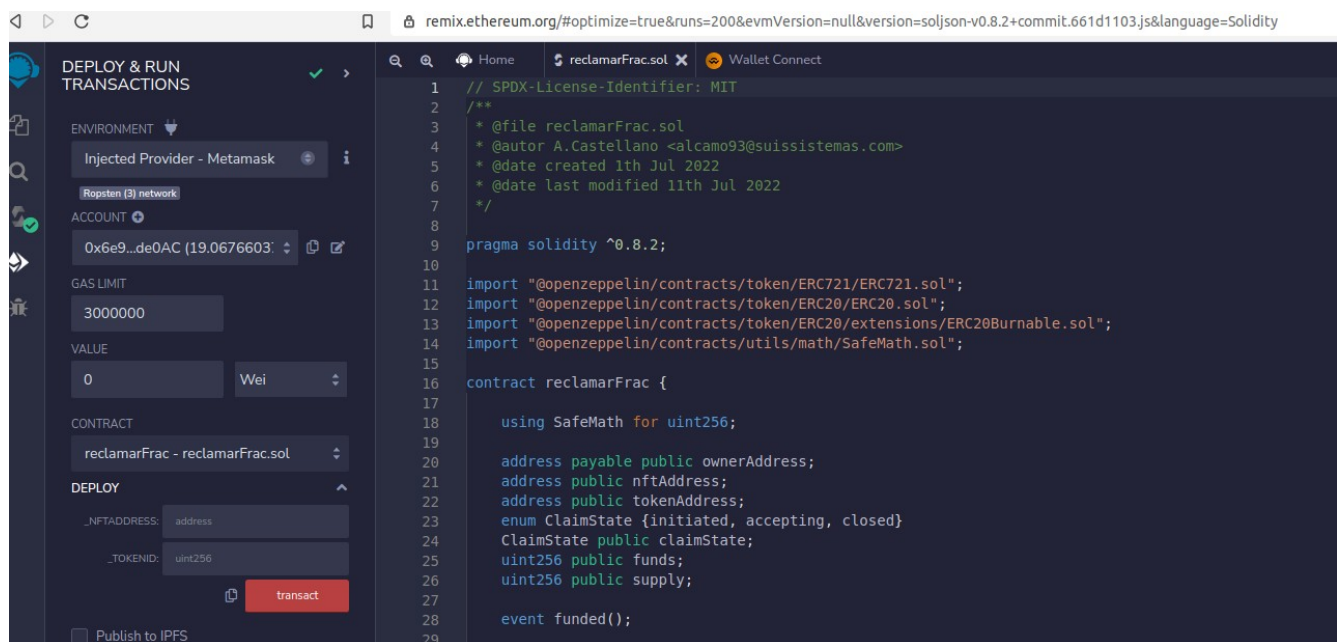
Estoy probando los servicios asociados al contrato y luego lo documento.

Se programo el servicio `implementaReclamo` para poder recomprar los FWCToken (WCT) es decir las partes del token NFT FraccionalWCT (FWCT) , para ello en este servicio debemos indicar el hash del FraccionalWCT (FWCT) y el TokenId del activo a recomprar como se ve en la siguiente pantalla:



El servicio `implementaReclamo` se esta programando como los anteriores, para efecto de nuestro ejemplo lo probaremos con Remix y Red Ropsten

Lo primero que vamos hacer es implementar el contrato inteligente `reclamoFrac` vía Remix, para luego fondear y que los inversores puedan canjear sus token por su dinero e intereses y liberen para quemar los tokens.



Lo que debemos hacer en esta pantalla de Remix es los siguiente:

Indicar el hash del FraccionalWCT (FWCT): **0x9fc1be2640bc3235f5441506d51852e4392f06c2**

y el TokenId del activo a recomprar: **2**

En Remix vamos al IDE en el navegador y conectamos por esta vía al nodo Ethereum y formamos las Tx a través de Metamask, esto me permite certificar y generar transacciones de los contratos en uso, esto lo había hecho inicialmente y funciona, como indique el error se genera al conectarnos via Infura para y no usar ni Remix, ni Metamask.

Acá el funcionamiento de las 3 (tres) funciones que se requieren para devolverle la inversión y el fee a los inversores:

implementaReclamo: Permite que el organizador/ propietario despliegue el contrato inteligente que su función es fondear el contrato del activo que se desea recomprar y permitir que el inversor reclame el capital y el fee de su inversión canjeando los token.

fondeaReclamo: Esta función permite al organizador/ propietario colocar el el monto y fee total a pagar a los inversores, acá indica el contrato del token que tiene las fracciones y que es uno por cada activo.

canjearToken: Permite al inversor retirar la inversión y el fee y devuelve la propiedad del activo al organizador/propietario, el inversor va al contrato de su fracción y le indica el contrato del reclamo y el monto de os token a canjear, al hacerlo se devuelve el monto +fee y se queman os token.

Como indicamos en Remix ingresamos

_NFTADDRESS: **0x9fc1be2640bc3235f5441506d51852e4392f06c2** y

_TOKENID: **2**

Y presionamos transact, en la pantalla a continuación vemos que nuestro contrato se implemento en la red de prueba roptsten.

remix.ethereum.org/#optimize=true&runs=200&evmVersion=null&version=soljson-v0.8.2+commit.661d1103.js&language=Solidity

DEPLOY & RUN
TRANSACTIONS

ENVIRONMENT
Injected Provider - Metamask
Ropsten (3) network
ACCOUNT
0x6e9...de0AC (19.0665318!)
GAS LIMIT
3000000
VALUE
0 Wei
CONTRACT
reclamarFrac - reclamarFrac.sol
DEPLOY
_NFTADDRESS: 0x9fc1be2640bc3235f5441506d5
_TOKENID: 2
transact
Publish to IPFS
OR
At Address Load contract from Address
Transactions recorded 1
Deployed Contracts
RECLAMARFRAC AT 0XB30...A9AB7

```
1 // SPDX-License-Identifier: MIT
2 /**
3  * @file reclamarFrac.sol
4  * @author A.Castellano <alcamo93@suissistemas.com>
5  * @date created 1th Jul 2022
6  * @date last modified 11th Jul 2022
7  */
8
9 pragma solidity ^0.8.2;
10
11 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
```

0 Listen on all transactions Search with transaction hash or address

- Select a Javascript file in the file explorer and then run 'remix.execute()' or 'remix.executeCurrent()' in the console
- Right click on a JavaScript file in the file explorer and then click 'Run'

The following libraries are accessible:

- web3 version 1.5.2
- ethers.js
- remix

Type the library name to see available commands.
creation of reclamarFrac pending...

[view on etherscan](#)

✓ [block:12636845 txIndex:2] from: 0x6e9...de0AC to: reclamarFrac.(constructor) value: 0 wei data: 0x608...00002

status	true Transaction mined and execution succeed
transaction hash	0x9cb7046ae44fe2ff5a3012de17603b76d1a6da7fb0a97ee4c69639bf5a8741d
from	0x6e927980222a26265a96891d2CbE79E1F4Ede0AC
to	reclamarFrac.(constructor)
gas	451409 gas
transaction cost	451409 gas
input	0x608...00002
decoded input	{ "address _nftAddress": "0x9fc1BE2640bc3235f5441506d51852E4392f06c2", "uint256 _tokenId": "2" }
decoded output	-
logs	[]
val	0 wei

Para verificar lo ocurrido vamos a Etherscan y revisamos la Tx, como se observa en la siguiente pantalla:

All Filters

Search by Address / Txn Hash / Block / Token / E

Ropsten Testnet Network

HomeBlockchain

Transaction Details

OverviewInternal TxnsState

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x9cb7046ae44fe2ff5a3012de17603b76d1a6da7fb0a97eea4c69639b5a8741d

Status:

Success

Block:

126368459 Block Confirmations

Timestamp:

1 min ago (Jul-21-2022 12:32:36 PM +UTC)

From:

0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To:

[Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7 Created]

Value:

0 Ether (\$0.00)

Transaction Fee:

0.001128522503159863 Ether (\$0.00)

Gas Price:

0.000000002500000007 Ether (2.500000007 Gwei)

Click to see More

0xb30426dbd82e270a12897984cc7dc424383a9ab7

El contrato creado:

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

All Filters

Search by Address / Txn Hash / Block / Token / Ens

Ropsten Testnet Network

HomeBlockchainTokensMiscRopsten

Contract 0xB30426dBd82e270A12897984cC7dC424383A9ab7

Contract Overview

Balance:

0 Ether

More Info

My Name Tag:

Not Available

Contract Creator:

0x6e927980222a26265a... at txn 0x9cb7046ae44fe2ff5a3...

Transactions

Internal TxnsErc20 Token TxnsContractEvents

Latest 1 from a total of 1 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x9cb7046ae44fe2ff5a3...	0x60806040	12636845	4 mins ago	0x6e927980222a26265a...	IN Contract Creation	0 Ether	0.00112852

[Download CSV Export]

A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

El propietario de la propiedad "comprará" los tokens FNT de los propietarios fraccionales. Para hacerlo, deposita en monto de lo invertido y el fee 1 ETH en el contrato inteligente de reclamos fraccionarios. Al hacerlo, proporciona la dirección del contrato inteligente del token FNT.

Para ello creamos el servicio **fondeaReclamo** y solicita los siguientes datos:

dirReclamo: Debemos proporcionar la dirección del contrato inteligente de reclamoFrac que es el que generamos en el punto anterior.

0xb30426dbd82e270a12897984cc7dc424383a9ab7

dirWCT: Debemos proporcionar la dirección del contrato inteligente FWCToken (WCT), dirección del token que es uno por cada activo y lo vemos acá: Para el ejemplo:

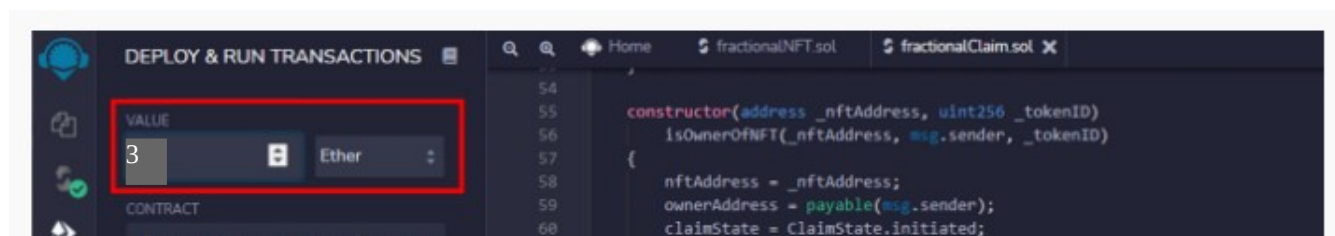
0x609e912E4755B202F43EF98dEC5D3287b21327E6

```
{
  "id": "2",
  "precio": "3000",
  "frac": "6",
  "tiempo": "12",
  "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
  "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
  "aprobado": "None",
  "dirToken": {
    "0": "2",
    "1": "0x609e912E4755B202F43EF98dEC5D3287b21327E6",
    "tokenId": "2",
    "fractionalToken": "0x609e912E4755B202F43EF98dEC5D3287b21327E6"
  },
  "totalToken": "6000000000000000000"
}
```

amount: Se indica el monto inversión + fee total repartir entre los tokens, se indica en Eth.

3

En Remix indicamos el valor, debemos considerar que el valor es en Ether.



Y luego en el contrato inteligente reclamoFrac colocamos el hash FWCToken (WCT), dirección del token que es uno por cada activo **0x609e912E4755B202F43EF98dEC5D3287b21327E6**

DEPLOY & RUN TRANSACTIONS

ACCOUNT 0x6e9...de0AC (19.066289f)

GAS LIMIT 3000000

VALUE 3 Ether

CONTRACT reclaimerFrac - reclaimerFrac.sol

DEPLOY

INFADRESS: 0xd61bcb240c32395d41506c

TOKENID: 2

transact

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 1 1

Deployed Contracts

RECLAMERFAC AT 0XB30...ABAE

claim address_token, val250

fund

tokens: 0x609e912647958202f43ef996

transact

claimState

funds

nftAddress

reclaimerFrac.sol

```
// SPDX-License-Identifier: MIT
2
3
4 * @file reclaimerFrac.sol
5 * @author A.Castellano <alcamo93@suissistemas.com>
6 * @date created 1th Jul 2022
7 * @date last modified 11th Jul 2022
8 */
9
10 pragma solidity ^0.8.2;
11
12 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
9
```

[illegible]

Al presionar transact, se fondea el contrato reclamarFrac que va a permitir al inversor canjear sus token, vemos la TX en Etherscan, en este caso coloque 3wei se debe colocar en Eth, para fin de la pruebas funciona.

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

Etherscan
Ropsten Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Ropsten

Transaction Details

Overview Logs (1) State

[This is a Ropsten Testnet transaction only]

Transaction Hash: 0x2383feca49316b4900cc6c5175bc61af30ac381c4a2161ae91d02ab04a313ce1

Status: Success

Block: 12636901 1 Block Confirmation

Timestamp: 4 secs ago (Jul-21-2022 12:44:24 PM +UTC)

From: 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To: Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7

Value: 3 wei (\$0.00)

Transaction Fee: 0.000241922500677383 Ether (\$0.00)

Gas Price: 0.000000002500000007 Ether (2.500000007 Gwei)

Click to see More

<https://ropsten.etherscan.io/address/0xb30426dbd82e270a12897984cc7dc424383a9ab7>

ropsten.etherscan.io/address/0xb30426dbd82e270a12897984cc7dc424383a9ab7

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

Etherscan
Ropsten Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Ropsten

Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7

Contract Overview

Balance: 3 wei

More Info

My Name Tag: Not Available

Contract Creator: 0x6e927980222a26265a... at txn 0x9cb7046ae44fe2ff5a3...

Transactions Internal Txns Erc20 Token Txns Contract Events

Latest 2 from a total of 2 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x2383feca49316b4900...	Fund	12636901	5 mins ago	0x6e927980222a26265a...	IN 0xb30426dbd82e270a12...	3 wei	0.00024192
0x9cb7046ae44fe2ff5a3...	0x60806040	12636845	17 mins ago	0x6e927980222a26265a...	IN Contract Creation	0 Ether	0.00112852

[Download CSV Export]

A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

Aqui vemos que nuestro contrato inteligente reclamarFrac a sido fondeado.

Canjear tokens FNT por ETH

Luego, el propietario informa a sus propietarios fraccionarios que ha activado un contrato inteligente de reclamos fraccionarios y ahora pueden intercambiar sus tokens FWCToken (WCT) por su parte de ETH utilizando el contrato inteligente. (Le debe indicar el Hash)

En el contrato inteligente de reclamos fraccionarios, ingresa la dirección del contrato inteligente del token FWCToken (WCT) y la cantidad que está canjeando.

El contrato inteligente Fractional Claims toma sus tokens FNT y los quema. Calcula el ETH total al que tiene derecho y se lo envía.

Se creo un servicio **canjearToken** se le debe indicar la siguiente información:

ProFrac: Dirección de la cuenta del propietario de la fracción.

ActivoId: TokenId que identifica el activo.

Spender: Hash contrato inteligente de reclamos fraccionarios

amount: Cantidad de tokens a canjear

En Remix nos vamos al contrato inteligente FWCToken (WCT) a la función approve, indicamos el hash del contrato inteligente reclamoFrac **0xb30426dbd82e270a12897984cc7dc424383a9ab7** y la cantidad de tokens a canjear, en este caso 6

The screenshot displays the Remix IDE interface with the following components:

- Left Panel (DEPLOY & RUN TRANSACTIONS):** Shows the 'approve' function being executed. The 'spender' field is set to '0xb30426dbd82e270a12897984cc7dc424383a9ab7' and the 'amount' is '6'. The 'Transact' button is highlighted.
- Center Panel (Solidity Code):** Displays the Solidity code for the 'reclamoFrac' contract, including a license identifier, author information, and the 'approve' function signature.
- Right Panel (Transaction Details):** Shows the transaction status as 'true Transaction mined and execution succeed'. It lists the transaction hash, from address, to address, gas used, and transaction cost.
- Bottom Panel (Logs):** Displays the transaction logs, including the 'approve' function call and the resulting output.
- MetaMask Notification:** A pop-up window on the right side of the screen asks for permission to access the WCT, showing the transaction hash and the amount.

Vemos en Etherscan la Tx approve

Transaction Details

Overview

Logs (1)

State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x7d6201ec558b50d5ad8c92dd219dfc657b2bfd86b8a8182051c8f6a1c9b1b1d

Status:

Success

Block:

12637004458 Block Confirmations

Timestamp:

1 hr 44 mins ago (Jul-21-2022 01:09:00 PM +UTC)

From:

0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To:

Contract 0x609e912e4755b202f43ef98dec5d3287b21327e6

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00011555000032354 Ether (\$0.00)

Gas Price:

0.000000002500000007 Ether (2.500000007 Gwei)

Etherscan

Ropsten Testnet Network

All FiltersSearch by Address / Txn Hash / Block / Token / Ens

HomeBlockchainTokensMiscRopste

Contract 0x609e912E4755B202F43EF98dEC5D3287b21327E6

Contract Overview

Balance:0 Ether

More Info

My Name Tag:Not Available

Contract Creator:0x6e927980222a26265a... at txn 0xf6cbbf83f7c80255081...

Token Tracker:FWCToken (WCT)

Transactions

Internal Txns

Erc20 Token Txns

Contract

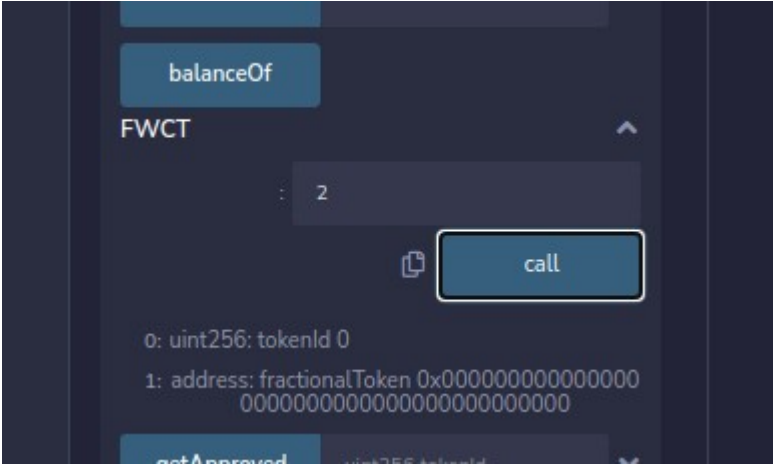
Events

Latest 7 from a total of 7 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x7d6201ec558b50d5ad...	Approve	12637004	1 hr 44 mins ago	0x6e927980222a26265a...	IN0x609e912e4755b202f4...	0 Ether	0.00011555

El contrato inteligente el contrato inteligente reclamoFrac toma sus tokens FNT y los quema. Calcula el ETH total al que tiene derecho y se lo envía al inversor.

Podemos verificar en Remix que ya to tenemos asociados token fraccionales asi:



GET ⌵ http://localhost:3021/activos/balanceWCT

Params Authorization Headers (9) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY
<input checked="" type="checkbox"/>	assetId
<input checked="" type="checkbox"/>	address
	Key

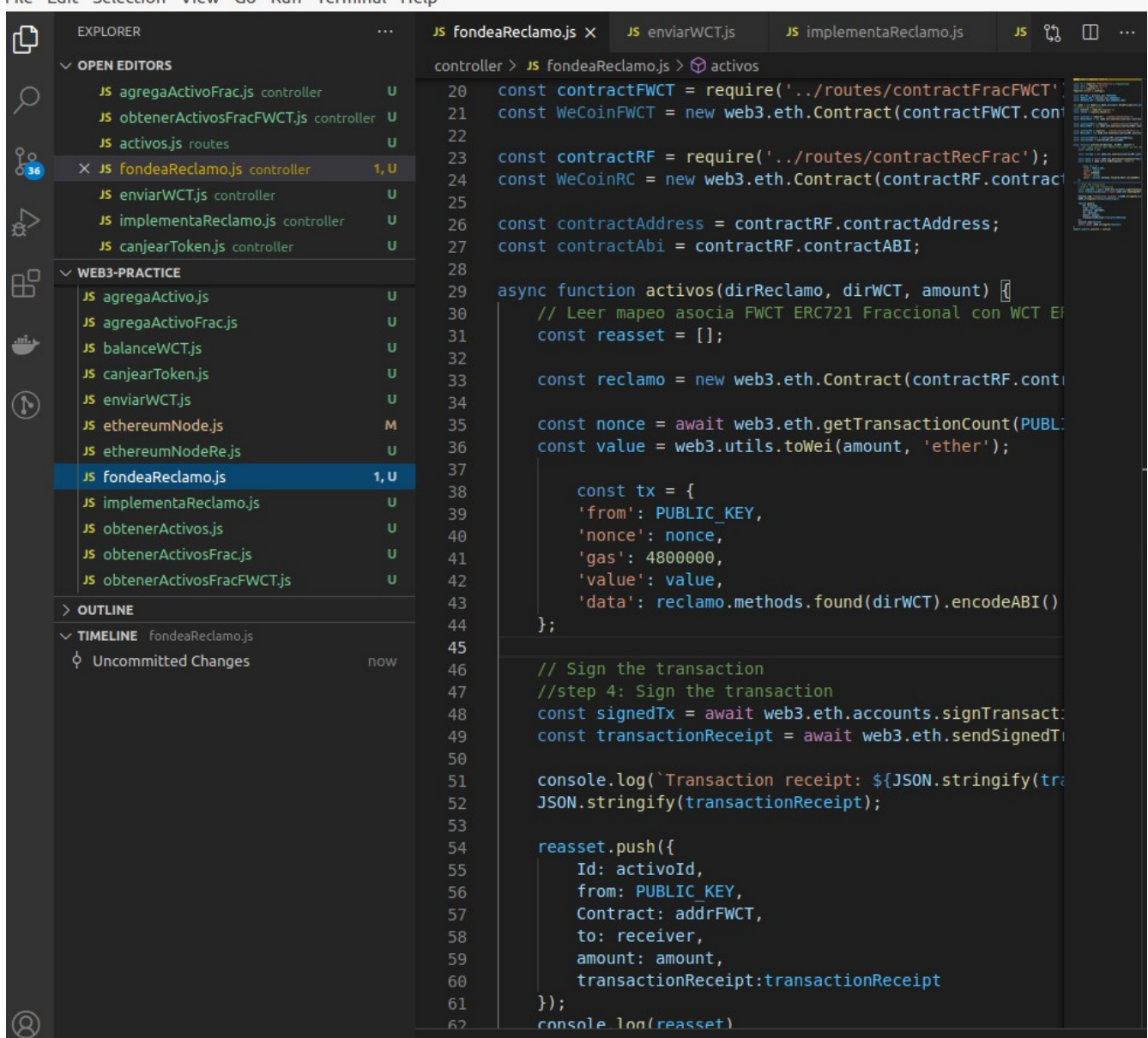
Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ⌵ ≡

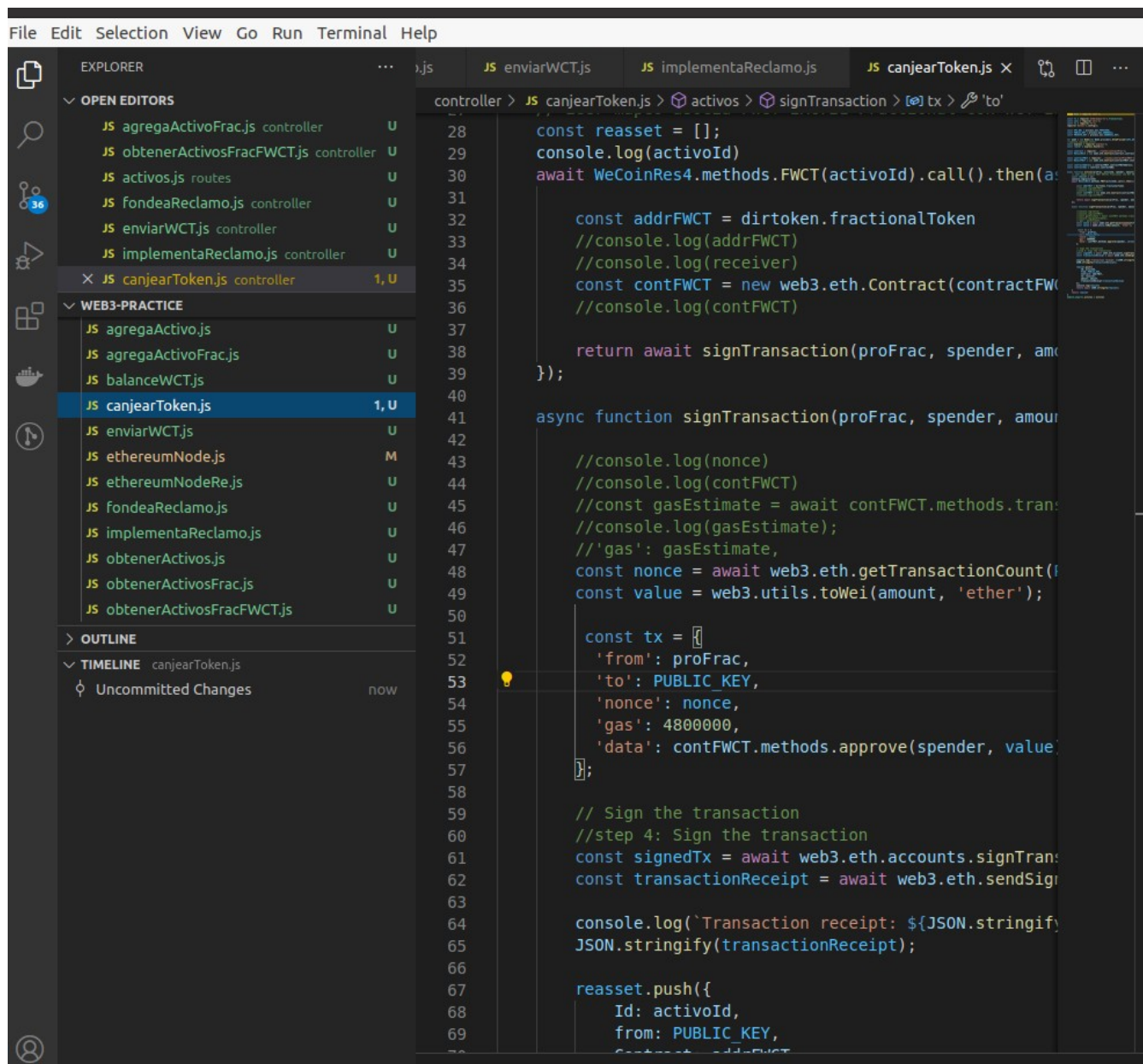
```
1 {
2   {
3     "id": "2",
4     "contractWCT": "0x609e912E4755B202F43EF98dEC5D3287b21327E6",
5     "walletAddress": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
6     "balanceWCT": "0"
7   }
8 }
```

Como vemos el contrato inteligente reclamoFrac funciona totalmente vía Remix, en la espera de que Infura ayude a determinar el error y si no habría que ver la opción de otro acceso al nodo Ethereum.

Los servicios asociados a reclamarFrac esta programados:



```
20 const contractFWCT = require('../routes/contractFracFWCT');
21 const WeCoinFWCT = new web3.eth.Contract(contractFWCT.contractABI);
22
23 const contractRF = require('../routes/contractRecFrac');
24 const WeCoinRC = new web3.eth.Contract(contractRF.contractABI);
25
26 const contractAddress = contractRF.contractAddress;
27 const contractAbi = contractRF.contractABI;
28
29 async function activos(dirReclamo, dirWCT, amount) {
30   // Leer mapeo asocia FWCT ERC721 Fraccional con WCT ERC20
31   const reaset = [];
32
33   const reclamo = new web3.eth.Contract(contractRF.contractABI, contractAddress);
34
35   const nonce = await web3.eth.getTransactionCount(PUBLIC_KEY, 'latest');
36   const value = web3.utils.toWei(amount, 'ether');
37
38   const tx = {
39     'from': PUBLIC_KEY,
40     'nonce': nonce,
41     'gas': 4800000,
42     'value': value,
43     'data': reclamo.methods.found(dirWCT).encodeABI()
44   };
45
46   // Sign the transaction
47   //step 4: Sign the transaction
48   const signedTx = await web3.eth.accounts.signTransaction(tx, PRIVATE_KEY);
49   const transactionReceipt = await web3.eth.sendSignedTransaction(signedTx.rawTransaction);
50
51   console.log(`Transaction receipt: ${JSON.stringify(transactionReceipt)}`);
52   console.log(JSON.stringify(transactionReceipt));
53
54   reaset.push({
55     Id: activoId,
56     from: PUBLIC_KEY,
57     Contract: addrFWCT,
58     to: receiver,
59     amount: amount,
60     transactionReceipt: transactionReceipt
61   });
62   console.log(reaset);
63 }
```



```
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
JS agregaActivoFrac.js controller U 28
JS obtenerActivosFracFWCT.js controller U 29
JS activos.js routes U 30
JS fondeaReclamo.js controller U 31
JS enviarWCT.js controller U 32
JS implementaReclamo.js controller U 33
X JS canjeaToken.js controller 1, U 34
WEB3-PRACTICE
JS agregaActivo.js U 37
JS agregaActivoFrac.js U 38
JS balanceWCT.js U 39
JS canjeaToken.js 1, U 40
JS enviarWCT.js U 42
JS ethereumNode.js M 43
JS ethereumNodeRe.js U 44
JS fondeaReclamo.js U 45
JS implementaReclamo.js U 46
JS obtenerActivos.js U 47
JS obtenerActivosFrac.js U 48
JS obtenerActivosFracFWCT.js U 49
OUTLINE
TIMELINE canjeaToken.js
Uncommitted Changes now

controller > JS canjeaToken.js > activos > signTransaction > tx > 'to'

const reasset = [];
console.log(activoId)
await WeCoinRes4.methods.FWCT(activoId).call().then(async () => {

  const addrFWCT = dirtoken.fractionalToken
  //console.log(addrFWCT)
  //console.log(receiver)
  const contFWCT = new web3.eth.Contract(contractFWCT, addrFWCT)
  //console.log(contFWCT)

  return await signTransaction(proFrac, spender, amount)
});

async function signTransaction(proFrac, spender, amount) {

  //console.log(nonce)
  //console.log(contFWCT)
  //const gasEstimate = await contFWCT.methods.transaction({
  //  'from': proFrac,
  //  'to': PUBLIC_KEY,
  //  'nonce': nonce,
  //  'gas': 4800000,
  //  'data': contFWCT.methods.approve(spender, value).encodeABI()
  //}).estimateGas({
  //  'from': proFrac,
  //  'to': PUBLIC_KEY,
  //  'nonce': nonce,
  //  'gas': 4800000,
  //  'data': contFWCT.methods.approve(spender, value).encodeABI()
  //});

  // Sign the transaction
  //step 4: Sign the transaction
  const signedTx = await web3.eth.accounts.signTransaction(transaction, privateKey);
  const transactionReceipt = await web3.eth.sendSignedTransaction(signedTx.rawTransaction);

  console.log(`Transaction receipt: ${JSON.stringify(transactionReceipt)}`);

  reasset.push({
    Id: activoId,
    from: PUBLIC_KEY,
    to: addrFWCT,
    amount: amount,
    nonce: nonce,
    gas: 4800000,
    data: contFWCT.methods.approve(spender, value).encodeABI()
  });
}
```

Como podemos ver a nivel de blockchain vemos que la prueba del concepto de token fraccional funciona y esto nos permite aplicarlo en diversos campos es infinito su uso.

En este ámbito de la fraccionalización el impacto en UK, artículo publicado la semana pasada 14-07-22. Esto en relación a la proyección y hacia donde se puede usar.

¿CAMBIO DE MODELO EN LOS FONDOS DE INVERSIÓN POR BLOCKCHAIN Y LA TOKENIZACIÓN?

https://www.linkedin.com/posts/alfredo-mu%C3%B1oz_investment-association-uk-investing-for-activity-6952524073998405633-tMhA?utm_source=linkedin_share&utm_medium=member_desktop_web

The Investment Association.pdf

La IA (The Investment Association) en UK, que agrupa a los Fondos de Inversión, defiende en este documento la necesidad de modernizar productos y servicios, por la inevitable innovación que supone la tecnología blockchain y los criptoactivos, que traerán beneficios en eficiencia operativa, menores costes con mejoras en prestación de servicios y un incremento y diversidad de los activos disponibles.

El fenómeno de la tokenización, claramente disruptivo, permitiría la propiedad fraccionada y supondría la apertura de mercados inicialmente accesibles con carácter institucional, facilitando el acceso y la liquidez, lo que podría generar inversiones hiperpersonalizadas.

Entienden que habría tres posibles escenarios que pueden producirse:

1. Mejoras sobre los fondos generando eficiencia sobre el funcionamiento actual, generando mejoras en la velocidad, la escala y la eficiencia, lo que exigiría la adopción de tokenización, permitiendo que las acciones de fondos y los activos subyacentes sean negociados de una manera mucho más efectiva.
2. Supone una evolución severa en la innovación con una personalización de la inversión, con bloques temáticos de inversión, ampliando la tokenización a empresas privadas, infraestructuras y activos digitales nativos
3. Y un tercer escenario, que supone un cambio radical, con una hiperpersonalización del inversor que disfrutaría de una experiencia interactiva y participativa, con exposiciones en acciones y valores individuales, en lugar de a través de un fondo. Habría un cambio en la participación del cliente y la relación pasaría de la delegación al fondo a una actuación como gestor de inversiones.

Por todo ello, la IA defiende tres ámbitos de colaboración entre los reguladores y la industria, que permitiera hacer más competitiva la industria de fondos en UK:

1. Innovación: Creando un marco para los fondos tokenizados; análisis de implicaciones de DeFi y consiguientes reformas.
2. Perímetro regulatorio: reglas para los criptoactivos, supervisión, control sobre las stablecoins y sus efectos
3. Cambio regulatorio: mejorar asesoramiento a los consumidores; suministro y divulgación de información digital a los inversores; generar reputación internacional de UK como centro financiero líder.