

Trazabilidad general

Estudiando el funcionamiento de trazabilidad



Trazabilidad general

Sobre la problemática a responder

- La idea es poder seguir una unidad cualquiera a través de su ciclo de movimiento y poder identificarlo unívocamente
- En el movimiento se involucran varias entidades, los participantes, que no necesariamente confían entre sí y que no tienen control uno sobre el otro
- Para cada movimiento de cada tracking, queremos saber qué participante hizo qué movimiento

Trazabilidad general

Sobre la problemática a responder

- El problema de la confianza entre partes hace la blockchain y smart contracts una buena solución, descentralizada y transparente
- Se proveerá un smart contract y una API que hará de abstracción sin quitar transparencia
- La API no tendrá persistencia, toda configuración a nivel operativa vivirá en el smart contract

Trazabilidad general

- Cada participante debe tener un nodo de ethereum en su infra
- Cada participante puede tener la API integrada en su infra
- Cada participante debe tener cuentas con saldo suficiente para operar
- El smart contract debe ser deployado por algún participante que será entonces el administrador del smart contract
- El administrador se debe encargarse de cargar los participantes y los roles en el smart contract
- El administrador se debe encargarse de cargar el workflow

Trazabilidad general

Configurando la API

- La API está hecha sobre Django, hay que crear un superuser que hará de admin de la aplicación en particular
- Se debe ingresar al admin de la aplicación con el superuser
- Hay que completar con el bytecode y el ABI del smart contract
- Completar la información del nodo, URL, account a usar y passphrase

Trazabilidad general

Configurando la API

Contract info			
NAME	DEFAULT	VALUE	IS MODIFIED
CONTRACT_ADDRESS Smart contract	0x00	<input type="text" value="0xb0B990876B689c5c33692790E1FeE9bC12F1A6Ce"/> Reset to default	✓
ABI Contract ABI		<input [{="" \"\"="" \"\",="" \"address\"="" \"currenthashstate\",="" \"inputs\":="" \"name\":="" \"outputs\":="" \"trackingaddress\",="" \"type\":="" constant\":="" true,="" type="text" value="[{ \" }],="" }]}]"=""/> Reset to default	✓
BYTECODE Smart Contract Bytecode	0x0	<input type="text" value="608060405234801561001057600080fd5b50336000806101000a81548173ffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffff160217905550"/> Reset to default	✓
FROM_BLOCK Start search from block	0	<input type="text" value="5298840"/> Reset to default	✓

Node info			
NAME	DEFAULT	VALUE	IS MODIFIED
NODE_URL Eth node url	http://127.0.0.1:8545	<input type="text" value="http://10.23.10.71:8501"/> Reset to default	✓
IS_POA Is Proof of Authority network	False	<input checked="" type="checkbox"/> Reset to default	✓
ACCOUNT Account	0x00	<input type="text" value="0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff"/> Reset to default	✓
PASSPHRASE Default account oassphrase		<input type="text" value="Clave123!"/>	✓

Trazabilidad general

Configurando datos operativos

- Ahora, hay que cargar los datos que corresponden al negocio en sí
- Habrá que cargar participantes y los roles que cumplen
- También es posible cargar el workflow lineal
- Estos se pueden cargar inicialmente y se pueden agregar más en cualquier momento

Trazabilidad general

Cargando los participantes por API

- El administrador del SC debe tener la API integrada en su infraestructura
- En la configuración de la API debe estar cargada la address del administrador
- El endpoint para cargar un participante es POST v1/participants
- El payload va por json
- Se deben pasar parámetros address y role
- Se puede usar directamente el swagger de la aplicación

Trazabilidad general

Cargando los participantes por API

POST

/v1/participants Lanza una transacción para agregar un participante con un rol

Lanza una transacción para agregar un participante con un rol

Precondiciones:

- address es un hexa válido para ethereum
- role es un str de como mucho 32 bytes
- el account configurado en esta aplicación es owner del smart contract configurado en esta aplicación

Parameters

Try it out

Name

Description

data

(body)

Example Value | Model

```
{
  "role": "END",
  "address": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff"
}
```

Parameter content type

application/json



Trazabilidad general

Cargando los participantes por API

- Ejecutar el webservice lanza la transacción si es válida según las precondiciones

Esto quiere decir que **no** deberíamos ver inmediatamente el resultado

- Se pueden verificar los participantes cargados con GET v1/participants
- Sobre redes tipo PoA, como la BFA, queda como responsabilidad del cliente esperar a que la transacción ya viva en un bloque en la cadena canónica

Trazabilidad general

Cargando los participantes por API

Code

Details

202

Undocumented

Response body

```
{
  "tx": "0xb1b6fbc20591c348cca9513f8b042ce4c4a08b638cdc91f310892dc5b890f92c",
  "message": "Participant adding transaction committed"
}
```

Download

Response headers

```
access-control-allow-origin: *
allow: GET, POST, HEAD, OPTIONS
content-length: 127
content-type: application/json
date: Tue, 02 Jul 2019 13:29:27 GMT
server: WSGIServer/0.2 CPython/3.5.7
vary: Accept, Cookie
x-frame-options: SAMEORIGIN
```

Trazabilidad general

Cargando el workflow por API

- El administrador debe tener la API integrada en su infraestructura
- El endpoint para cargar un participante es POST v1/workflowsteps
- El payload va por json
- Se deben pasar parámetros message y role
- Se puede usar directamente el swagger de la aplicacion
- Este endpoint también lanza una transacción
- Se pueden ver los pasos del workflow con GET v1/workflowsteps

Trazabilidad general

Cargando el workflow por API

POST

/v1/workflowsteps Lanza una transacción para agregar un paso al workflow

Lanza una transacción para agregar un paso al workflow

Precondiciones:

- role es un str de como mucho 32 bytes
- existe al menos un participante con ese role asignado
- el account configurado en esta aplicación es owner del smart contract configurado en esta aplicación

Parameters

Try it out

Name

Description

data

(body)

Example Value | Model

```
{
  "role": "string",
  "message": "string"
}
```

Parameter content type

application/json



Trazabilidad general

Ahora a los trackings

- Cargada ya la información operativa, podemos empezar con los trackings
- Cada tracking se representa con un identificador unívoco que elige el cliente
- Se puede pasar un estado en forma de diccionario, que se serializará y su hash será lo que se guarde en la blockchain

Trazabilidad general

Iniciando el track

Paso 1

Inicio

POST

/v1/trackings/start Lanza una transacción para iniciar un tracking con workflow

```
{
  "tracking_id": "mi_ejemplo",
  "state": {"atributo_optativo": "algun_valor"}
}
```

Trazabilidad general

Iniciando el track

- Cargada ya la información operativa, podemos empezar con los trackings
- Cada tracking se representa con un identificador unívoco que elige el cliente
- Se puede pasar un estado en forma de diccionario, que se serializará y su hash será lo que se guarde en la blockchain

Trazabilidad general

Iniciando el track

Paso 1

Inicio

Request URL

```
http://localhost:8030/v1/trackings/start
```

Server response

Code

Details

202

Undocumented

Response body

```
{
  "message": "Tracking start transaction accepted",
  "tx": "0xaca9b63eaab00d9760dba710be49ee9b6a7deaa95693599f886d06c73c0c566e"
}
```

Trazabilidad general

Iniciando el track

- Con GET v1/trackings se puede ver un listado completo paginable de todos los trackings que estan activos y que ya terminaron
- Con GET v1/trackings/{tracking_id} se puede ver un tracking particular con todo su historial

Trazabilidad general

Iniciando el track

Paso 1

Inicio

Request URL

`http://localhost:8030/v1/trackings`

Server response

Code

Details

200

Response body

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "block_number": 5556576,
      "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff",
      "datetime": "2019-07-03T18:09:08",
      "tracking_id": "mi_ejemplo",
      "tx": "0xaca9b63eaab00d9760dba710be49ee9b6a7deaa95693599f886d06c73c0c566e",
      "timestamp": 1562177348
    }
  ]
}
```

Trazabilidad general

Iniciando el track

Paso 1

Inicio

Request URL

`http://localhost:8030/v1/trackings/mi_ejemplo`

Server response

Code

Details

200

Response body

```
{
  "results": [
    {
      "block_number": 5556576,
      "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff",
      "datetime": "2019-07-03T18:09:08",
      "hash_state": "3c6a482b4e97c8f4dffa3bab0723508cc3165eeb5f26fa52050813ecd9470487",
      "message": "Se proveen materiales a fabricante de autopartes",
      "tx": "0xaca9b63eaab00d9760dba710be49ee9b6a7deaa95693599f886d06c73c0c566e",
      "timestamp": 1562177348
    }
  ]
}
```

Trazabilidad general

Moviendo el track



POST

/v1/trackings/move Lanza una transacción para mover al siguiente paso un tracking en su workflow

```
{
  "tracking_id": "mi_ejemplo",
  "state": {"atributo_optativo": "algun_valor"}
}
```

Trazabilidad general

Moviendo el track



Request URL	
<code>http://localhost:8030/v1/trackings/move</code>	
Server response	
Code	Details
202 <i>Undocumented</i>	Response body <pre>{ "message": "Forward tracking transaction committed", "tx": "0xa7088389b76bf4687ce420109ea69e512a9d475640e626ff0e507166e3dade57" }</pre>

Trazabilidad general

Moviendo el track

Request URL	
http://localhost:8030/v1/trackings/mi_ejemplo	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "results": [{ "block_number": 5556576, "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff", "datetime": "2019-07-03T18:09:08", "hash_state": "3c6a482b4e97c8f4dffa3bab0723508cc3165eeb5f26fa52050813ecd9470487", "message": "Se proveen materiales a fabricante de autopartes", "tx": "0xaca9b63eaab00d9760dba710be49ee9b6a7deaa95693599f886d06c73c0c566e", "timestamp": 1562177348 }, { "block_number": 5556704, "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff", "datetime": "2019-07-03T18:19:48", "hash_state": "3c6a482b4e97c8f4dffa3bab0723508cc3165eeb5f26fa52050813ecd9470487", "message": "Se reciben materiales de proveedor", "tx": "0x346a4aefcc96079835ae651f171df136df1c15e360874b46dad73475d2007811", "timestamp": 1562177988 }] }</pre>

Trazabilidad general

Nos adelantamos un poco hasta el final del tracking...



```
{
  "block_number": 5556876,
  "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff",
  "datetime": "2019-07-03T18:34:08",
  "hash_state": "3c6a482b4e97c8f4dffa3bab0723508cc3165eeb5f26fa52050813ecd9470487",
  "message": "Se envía auto a concesionaria",
  "tx": "0x15db8b4b2a5d76d369e33320336ffea82ca774d9a2f4a4c4e91a5e6bb16d6480",
  "timestamp": 1562178848
},
{
  "block_number": 5557148,
  "participant": "0x21ea59FC5cE54a827E20BC9b736FeeD8F9C880Ff",
  "datetime": "2019-07-03T18:56:48",
  "hash_state": "3c6a482b4e97c8f4dffa3bab0723508cc3165eeb5f26fa52050813ecd9470487",
  "message": "Se recibe auto",
  "tx": "0xa7088389b76bf4687ce420109ea69e512a9d475640e626ff0e507166e3dade57",
  "timestamp": 1562180208
}
]
```


Trazabilidad general

Trabajando sin un workflow definido



Trazabilidad general

Trabajando sin un workflow definido

- Si no hay workflow definido o no es necesariamente lineal se puede operar sin uno
- Hace falta especificar el mensaje en el momento de hacer el inicio del track o del movimiento
- Hace falta especificar el rol que tiene que tener el siguiente paso también

Trazabilidad general

Trabajando sin un workflow definido



POST

/v1/trackings/noworkflow/start Lanza una transacción para iniciar un tracking sin workflow

```
{
  "state": {"atributo_optativo": "algun_valor"},
  "tracking_id": "mi_ejemplo",
  "next_role": "AUTOPARTES",
  "message": "Se mandan materiales para fabricacion de autopartes"
}
```

Trazabilidad general

Trabajando sin un workflow definido

- Al no haber workflow definido, se define en el tiempo de vida del track cual es el siguiente paso
- Para definir que un paso es el último basta con no especificar el siguiente rol en el movimiento
- Más allá de eso, no hay diferencia en los listados de trackings con workflow o no

Otros patrones

Qué pasa si en el negocio hay varios tipos de producto?

- Supongamos que somos alguna empresa que le interesa tener control de trazabilidad sobre distintos tipos de productos
- Hay varias formas de modelar esto

Otros patrones

Qué pasa si en el negocio hay varios tipos de producto?

1) Se pueden identificar los tipos de producto desde el lado del cliente, y que en el tracking id se haga la diferenciación, esto significa que solo hace falta la API y el smart contract y la lógica del tracking id definida

Por ej: si tenemos por productos a las leches descremadas y enteras se puede tener la lógica de tracking id tipo

tracking_id: lote1_descremada

tracking_id: lote1_entera

Otros patrones

Qué pasa si en el negocio hay varios tipos de producto?

2) Como la API es muy minimalista en términos de infraestructura, se pueden instanciar varios smart contracts de negocio y varias API para cada tipo de producto, con cualquier herramienta de containers esto debería ser fácil de hacer

Volviendo al mismo ejemplo del punto anterior, se puede deployar un SC de negocio y una API por producto a seguir y se tiene todo bien distinguido desde el punto de vista de infraestructura

Otros patrones

Qué pasa si en el negocio hay varios tipos de producto?

3) Finalmente, se puede usar un atributo del estado de cada tracking para identificar el tipo de producto y tener así la distinción