Una criptomoneda centralizada

¿Cómo funciona Bitcoin?

Contenidos

En realidad vamos a definir dos:

- **Goofycoin (una moneda penca)**
- Scroogecoin (una moneda menos penca)

Goofycoin

Participantes en el sistema

Goofy

Goofycoin

Participantes en el sistema

Goofy

Alice

Bob

Charlie

Goofycoin

Participantes en el sistema

Goofy

(LS_G, LP_G)

Alice

(LS_A, LP_A)

Bob

(LS_B, LP_B)

Charlie

(LS_C, LP_C)

Goofycoin

Participantes en el sistema

Goofy

(LS_G, LP_G)

Alice

(LS_A, LP_A)

Bob

(LS_B, LP_B)

Charlie

(LS_C, LP_C)

No es necesario registrarse

Solo necesitamos saber la LP_G

Goofycoin

Tres reglas

1. Creación de un goofycoin
2. Transferencia de un goofycoin
3. Verificación que un goofycoin es valido

Regla 1

Creación de goofycoin

(LS_G, LP_G)

Regla 1

Creación de goofycoin

(LS_G, LP_G)

CreateCoin [UniqueCoinID]

Regla 1

Creación de goofycoin

(LS_G, LP_G)

Signed LS_G
CreateCoin [UniqueCoinID]

Regla 1

Creación de goofycoin

(LS_G, LP_G)

Signed LS_G
CreateCoin [UniqueCoinID]

Goofy es el dueño de cada goofycoin creado!!!

Regla 1

Creación de goofycoin

(LS_G, LP_G)

CreateCoin CoinID(27)

Regla 1

Creación de goofycoin

(LS_G, LP_G)

Firma del string "CreateCoin CoinID(27)" con LS_G
CreateCoin CoinID(27)

Regla 1

Creación de goofycoin

(LS_G, LP_G)

Firma del string "CreateCoin CoinID(27)" con LS_G
CreateCoin CoinID(27)

Goofy es el dueño del goofycoin CoinID(27)

Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)

Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)

Signed LS_G
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)

Coin27

Signed LS_G
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_A, LP_A)

Pay to LP_A : Hp(Coin27)

(LS_G, LP_G)

Coin27

Signed LS_G

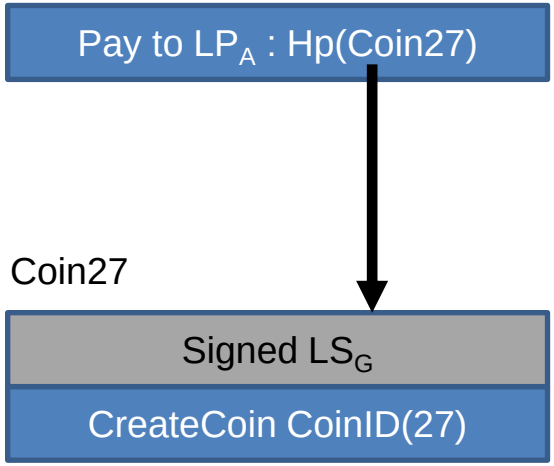
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)

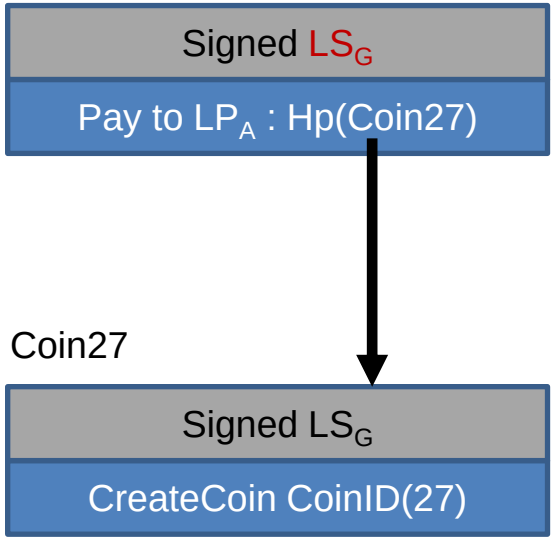


Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)

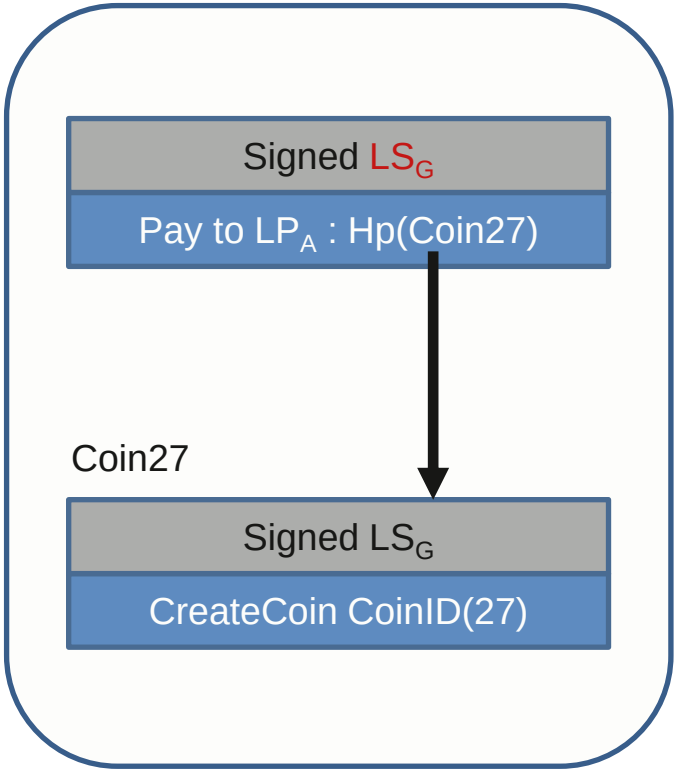


Regla 2

Transferencias

(LS_A, LP_A)

(LS_G, LP_G)



Alice es la dueña del goofycoin con el ID 27

Regla 2

Transferencias

(LS_A, LP_A)

Regla 2

Transferencias

(LS_A, LP_A)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G

CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)

Coin27[1]

Signed LS_G
Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)

Pay to LP_C : Hp(Coin27[1])

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G

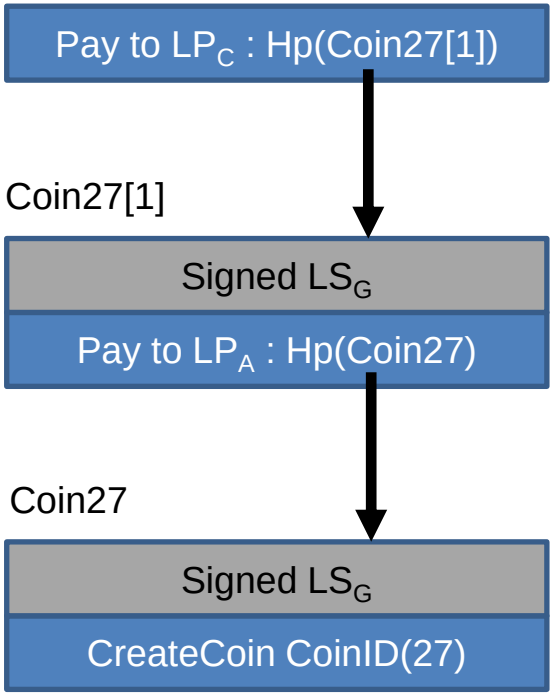
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)

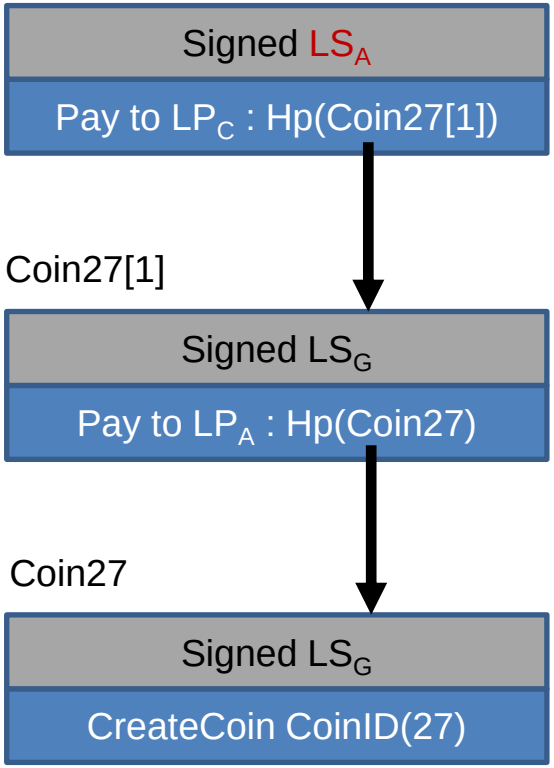


Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)



Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G

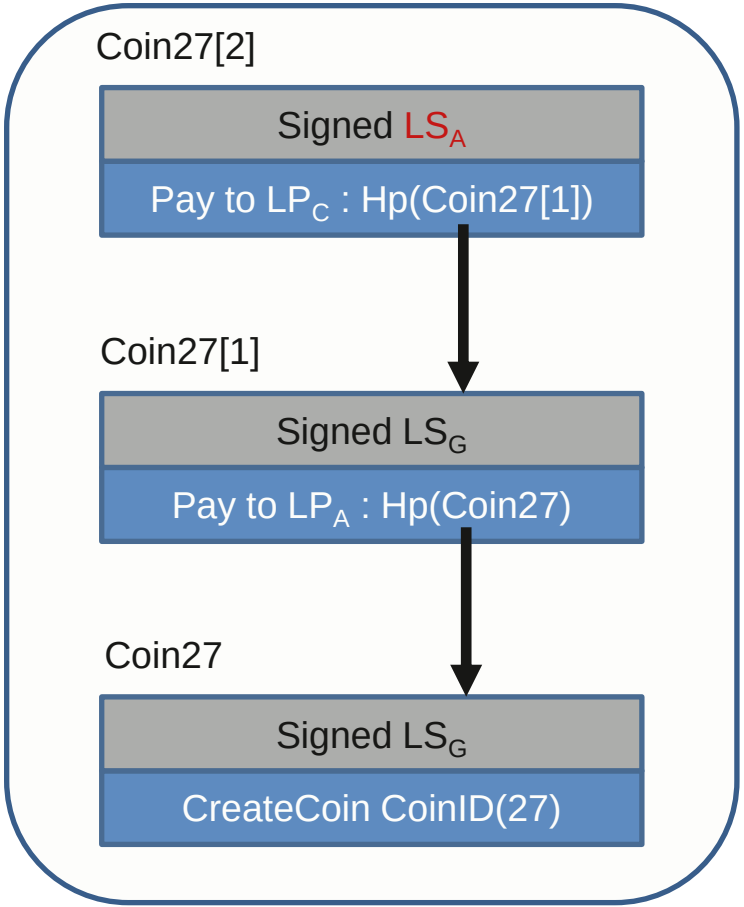
CreateCoin CoinID(27)

Regla 2

Transferencias

(LS_C, LP_C)

(LS_A, LP_A)



Regla 3

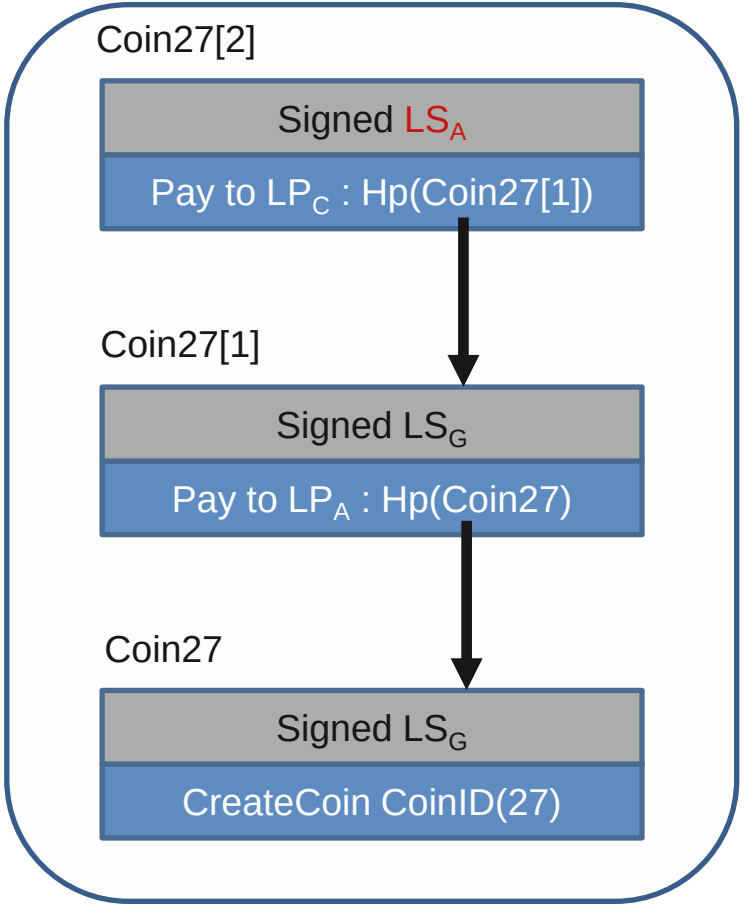
Verificación

(LS_C, LP_C)

Regla 3

Verificación

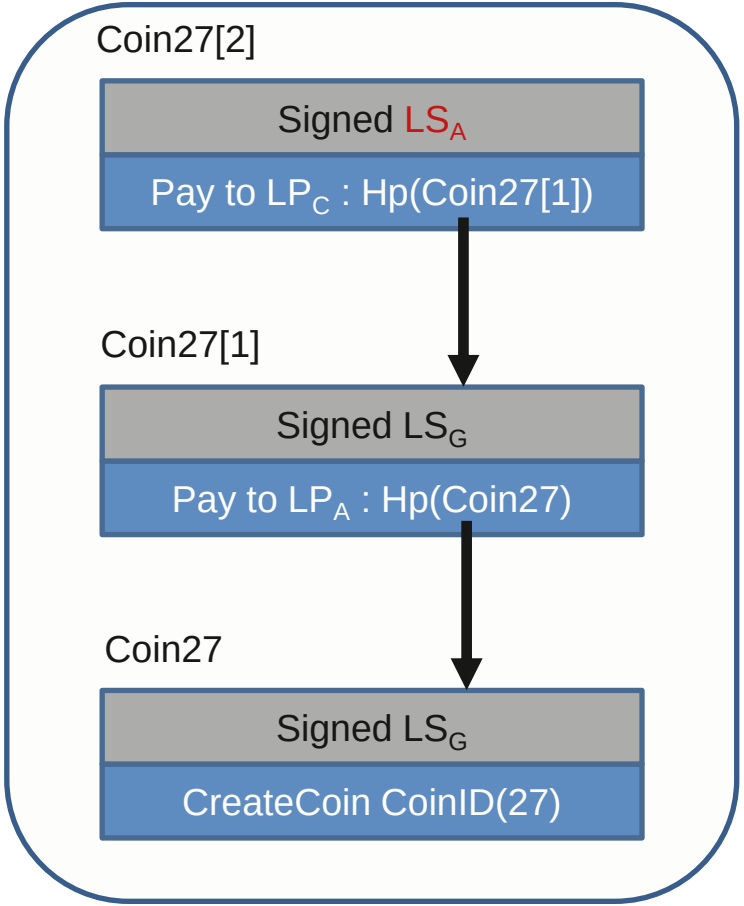
(LS_C, LP_C)



Regla 3

Verificación

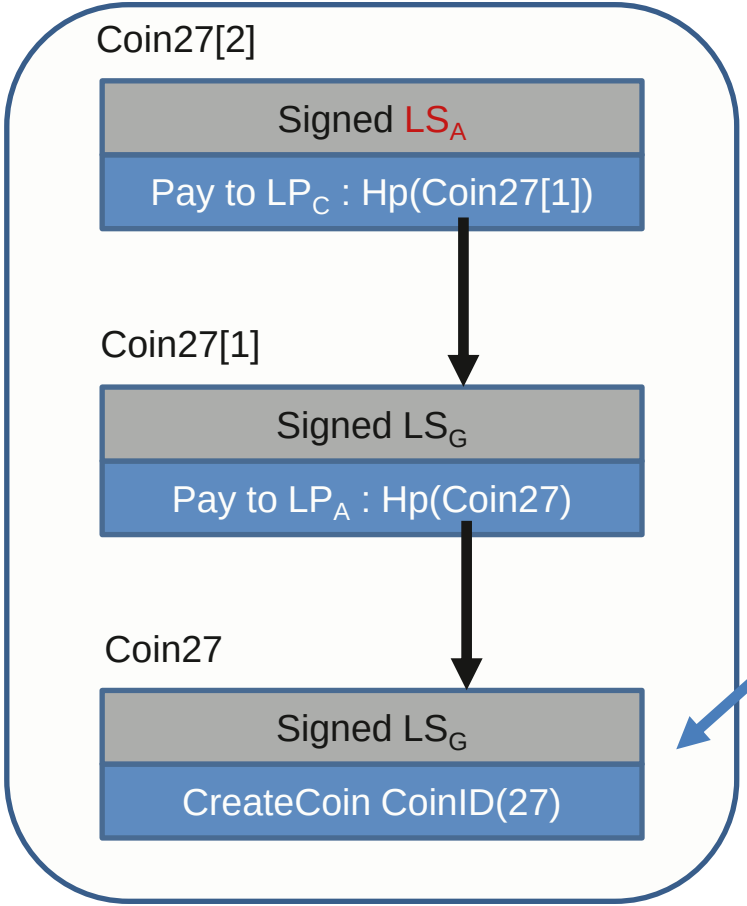
(LS_C, LP_C)



Regla 3

Verificación

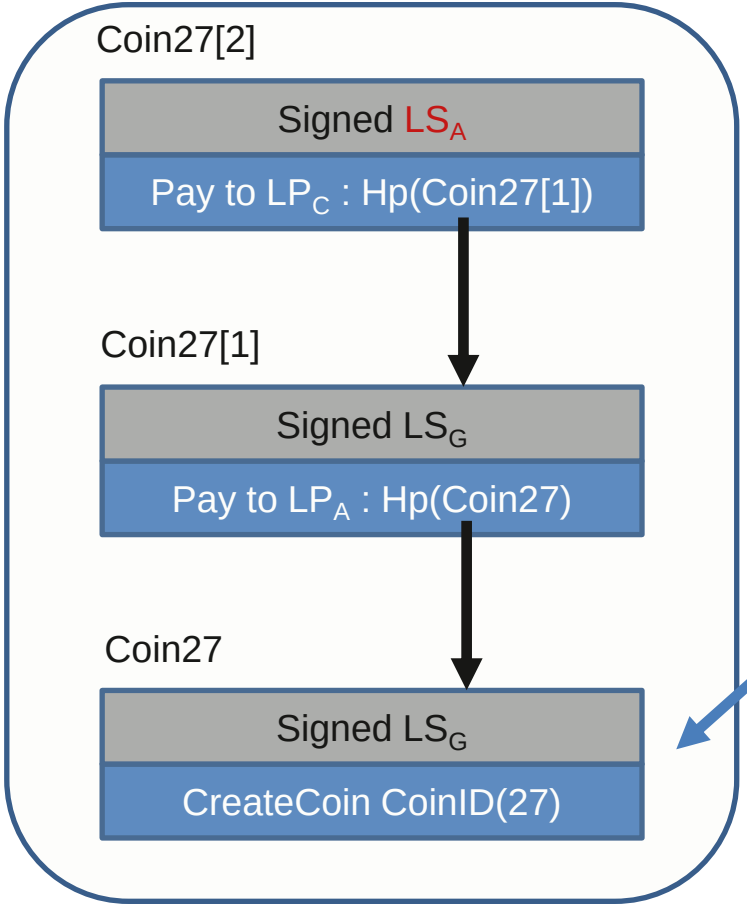
(LS_C, LP_C)



Regla 3

Verificación

(LS_C, LP_C)



Regla 3

Verificación

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G

CreateCoin CoinID(27)

Regla 3

Verificación

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

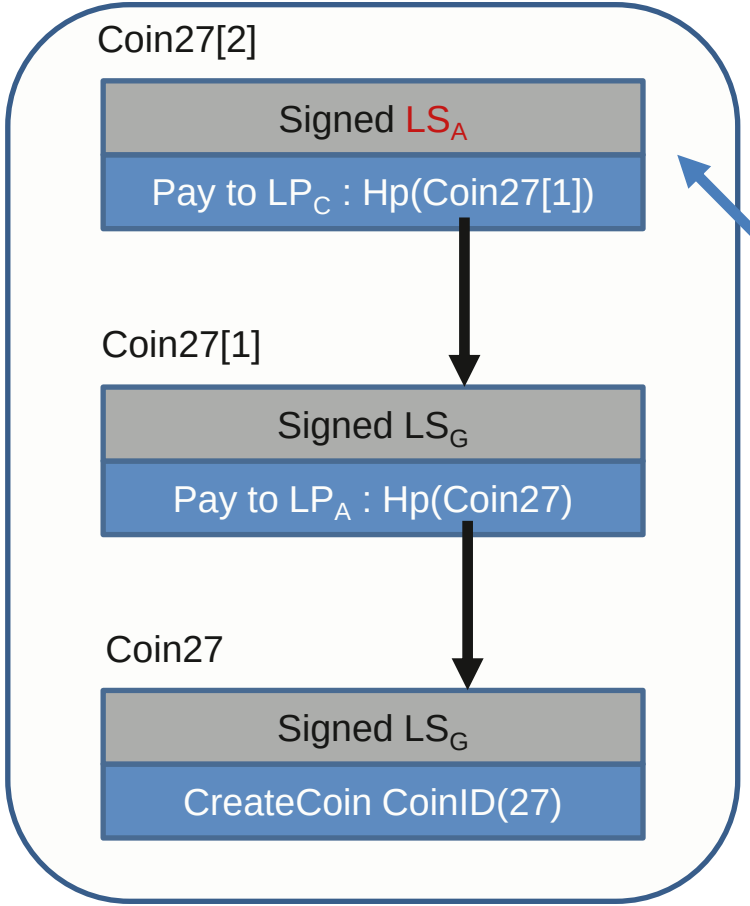
Signed LS_G

CreateCoin CoinID(27)

Regla 3

Verificación

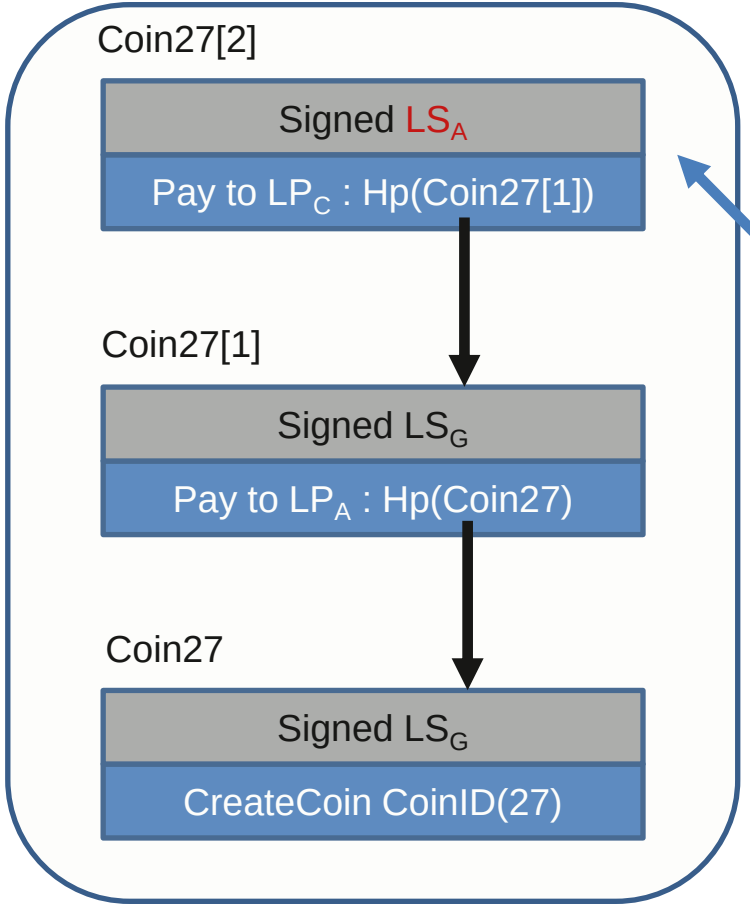
(LS_C, LP_C)



Regla 3

Verificación

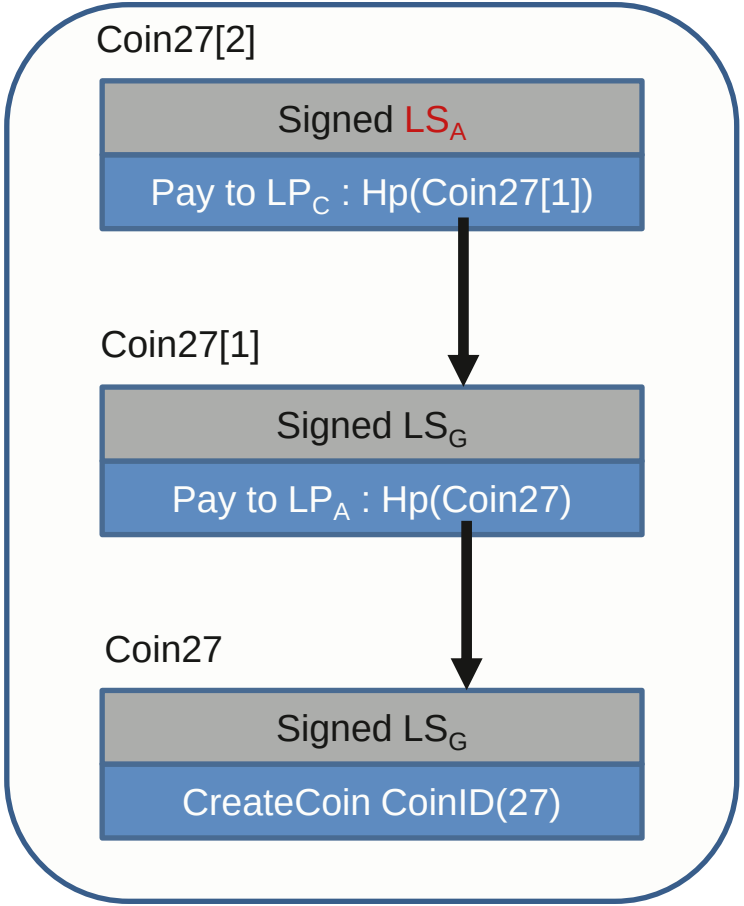
(LS_C, LP_C)



Regla 3

Verificación

(LS_C, LP_C)



Goofycoin

Tres reglas

1. Goofy crea los goofycoins
2. Transferencia de goofycoins
3. Verificación que un goofycoin es valido

Problema con Goofycoin

Double spending

(LS_A, LP_A)

Problema con Goofycoin

Double spending

(LS_A, LP_A)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

Coin27

Signed LS_G

CreateCoin CoinID(27)

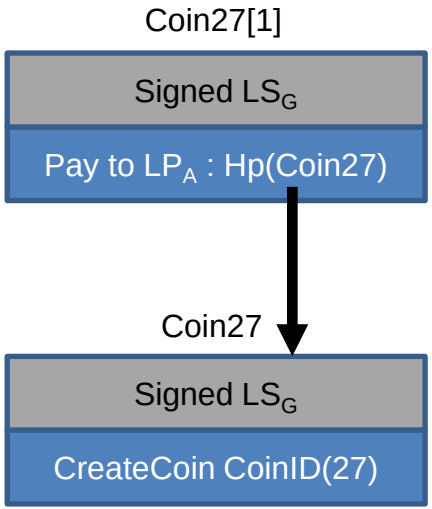
Problema con Goofycoin

Double spending

(LS_C, LP_C)

(LS_B, LP_B)

(LS_A, LP_A)



Problema con Goofycoin

Double spending

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

(LS_B, LP_B)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

(LS_A, LP_A)

Coin27

Signed LS_G

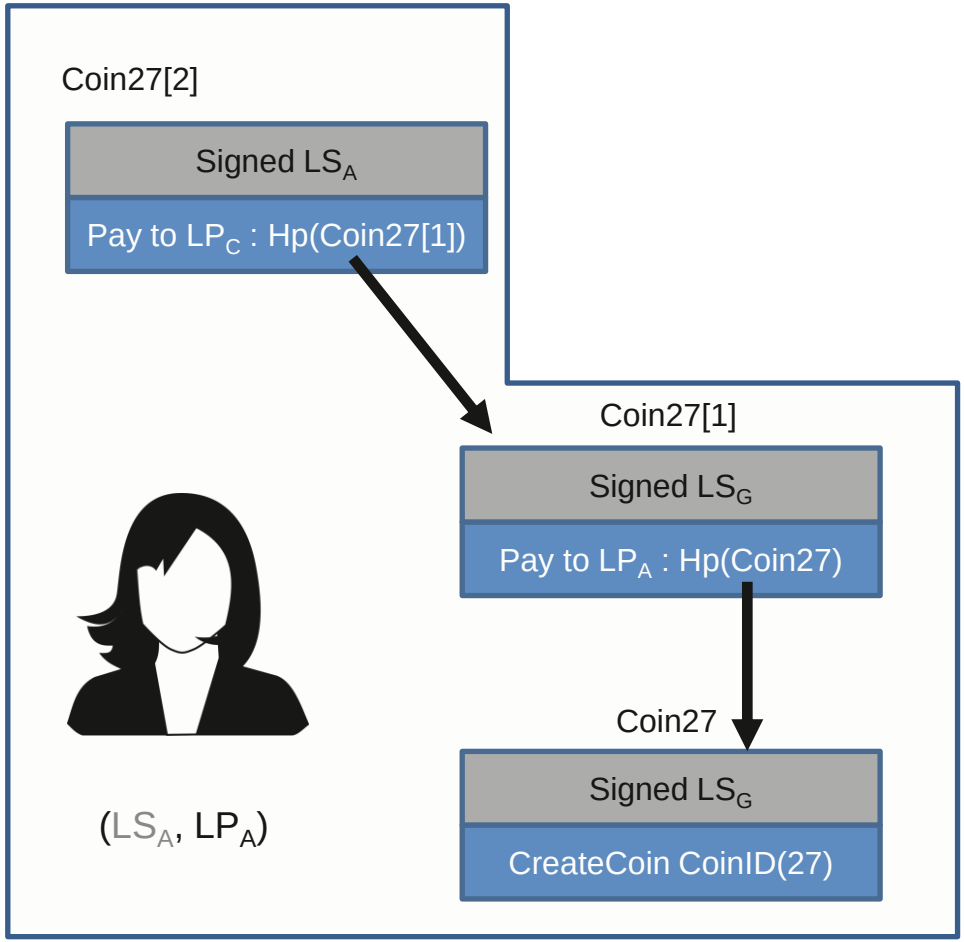
CreateCoin CoinID(27)

Problema con Goofycoin

Double spending

(LS_C, LP_C)

(LS_B, LP_B)



(LS_A, LP_A)

Problema con Goofycoin

Double spending

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

(LS_B, LP_B)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

(LS_A, LP_A)

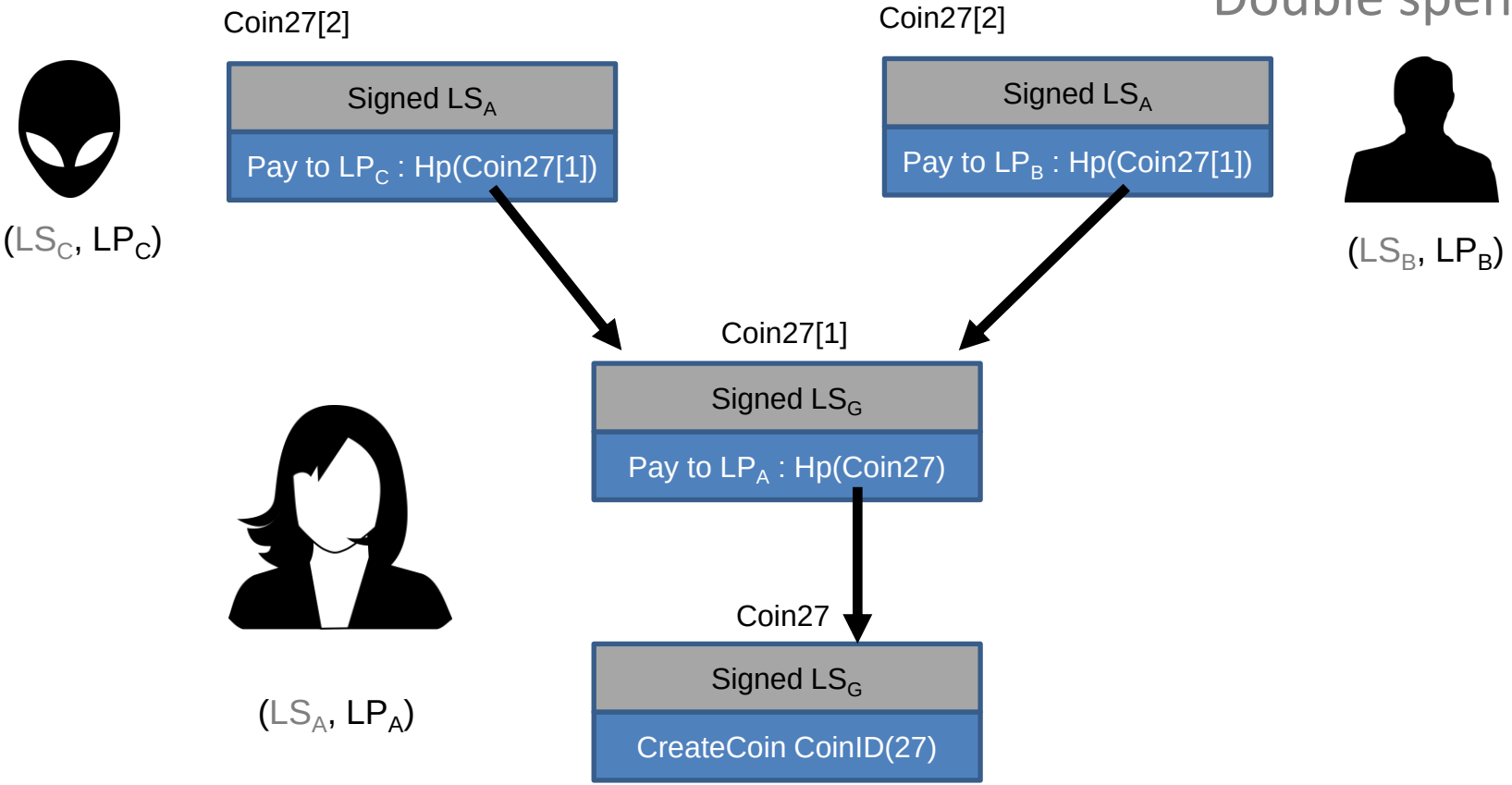
Coin27

Signed LS_G

CreateCoin CoinID(27)

Problema con Goofycoin

Double spending



Problema con Goofycoin

Double spending

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

Coin27[2]

Signed LS_A

Pay to LP_B : Hp(Coin27[1])

(LS_B, LP_B)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

(LS_A, LP_A)

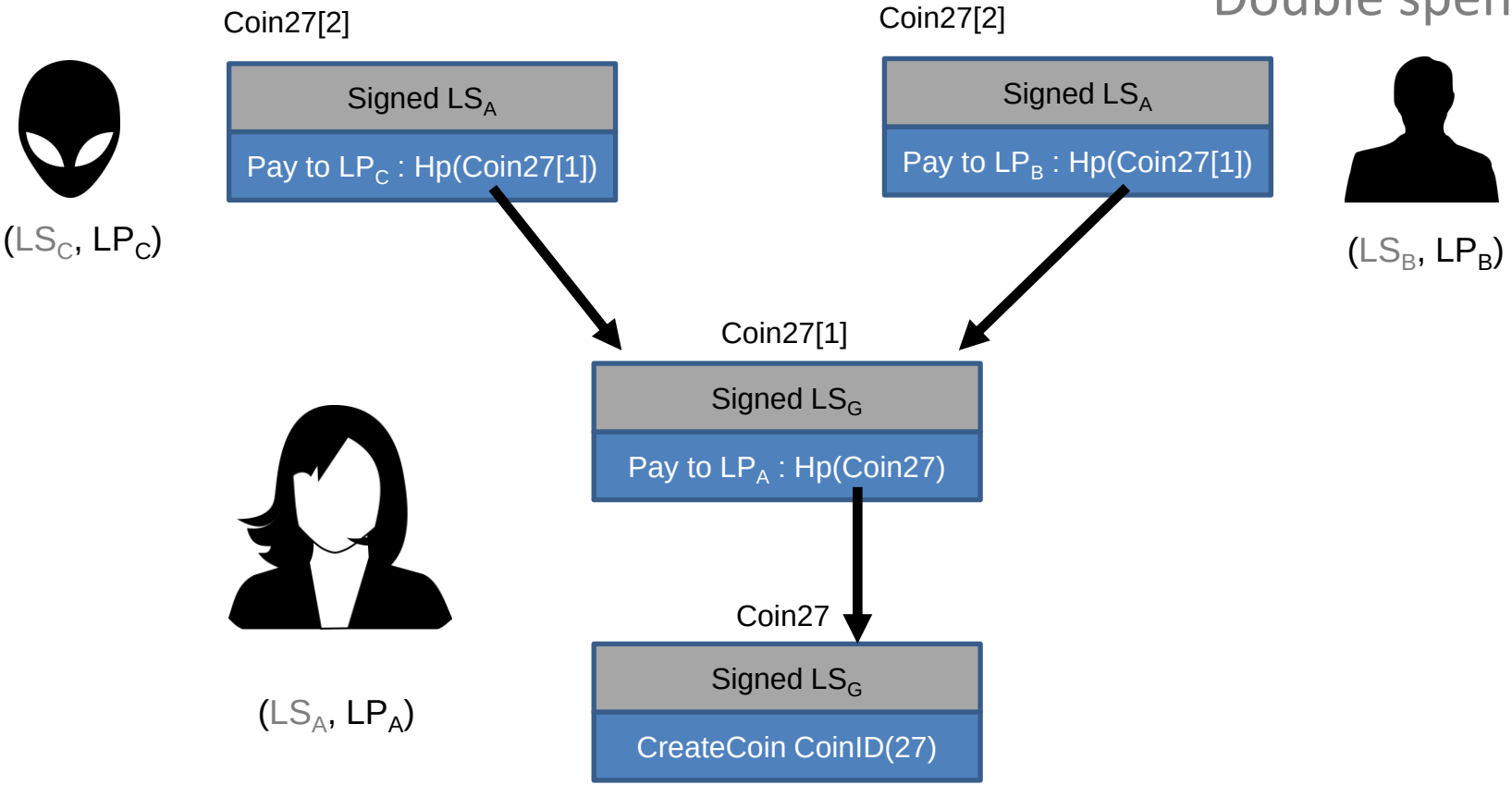
Coin27

Signed LS_G

CreateCoin CoinID(27)

Problema con Goofycoin

Double spending



Problema con Goofycoin

Double spending

(LS_C, LP_C)

Coin27[2]

Signed LS_A

Pay to LP_C : Hp(Coin27[1])

Coin27[2]

Signed LS_A

Pay to LP_B : Hp(Coin27[1])

(LS_B, LP_B)

Coin27[1]

Signed LS_G

Pay to LP_A : Hp(Coin27)

(LS_A, LP_A)

Coin27

Signed LS_G

CreateCoin CoinID(27)

Como resolver esto???

Una observación interesante

Bitcoin usa precisamente este modelo de transacciones:

- Lista de hash pointers desde la creación de un Bitcoin
- Obviamente, hay que resolver el problema de doble gasto

Scroogecoin

Scroogecoin

Scroogecoin

Scroogecoin

Libro contable
Scrooge crea $70 para Alice
Scrooge crea $50 para Bob
Alice paga Bob $50
Alice paga Charlie $20
Bob paga Charlie $100
Charlie paga Alice $120
…

Scroogecoin

Protección antes doble gastos

Scrooge publica las transacciones en un Libro contable público:

- Una vez hecha, la transacción se queda en el Libro (con un ID) para siempre
- Para asegurar el Libro contra cambios en el futuro, usaremos Blockchain

Con Blockchain:

- Uno puede verificar si el dinero fue gastado en el pasado

Scroogecoin

Estructura general

Scroogecoin

Estructura general

Scroogecoin

Estructura general

transID: 73

transaction data

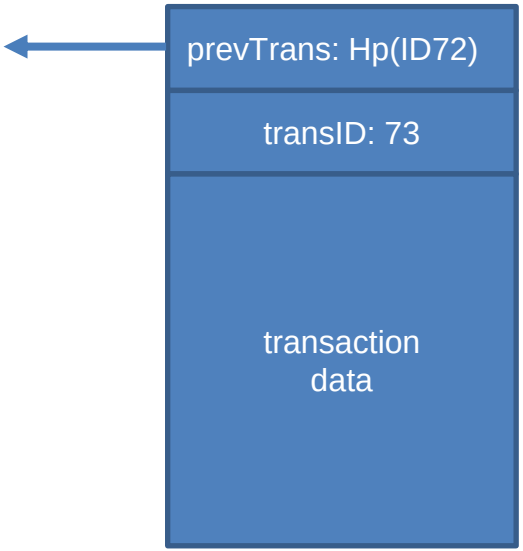
Scroogecoin

Estructura general

prevTrans: Hp(ID72)
transID: 73
transaction data

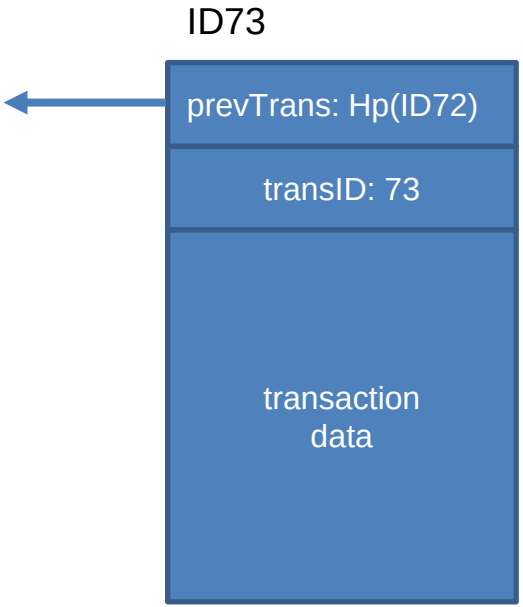
Scroogecoin

Estructura general



Scroogecoin

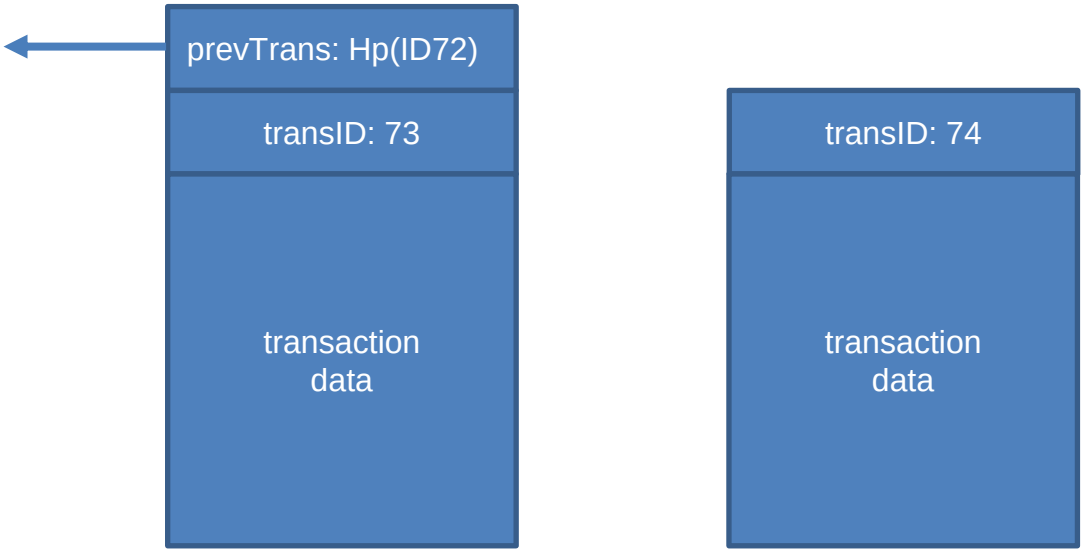
Estructura general



Scroogecoin

Estructura general

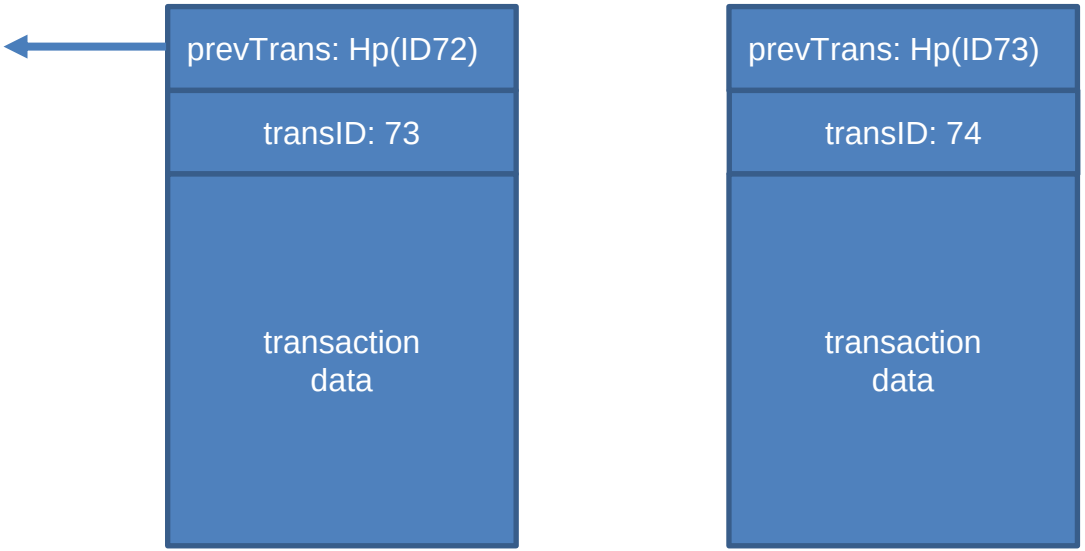
ID73



Scroogecoin

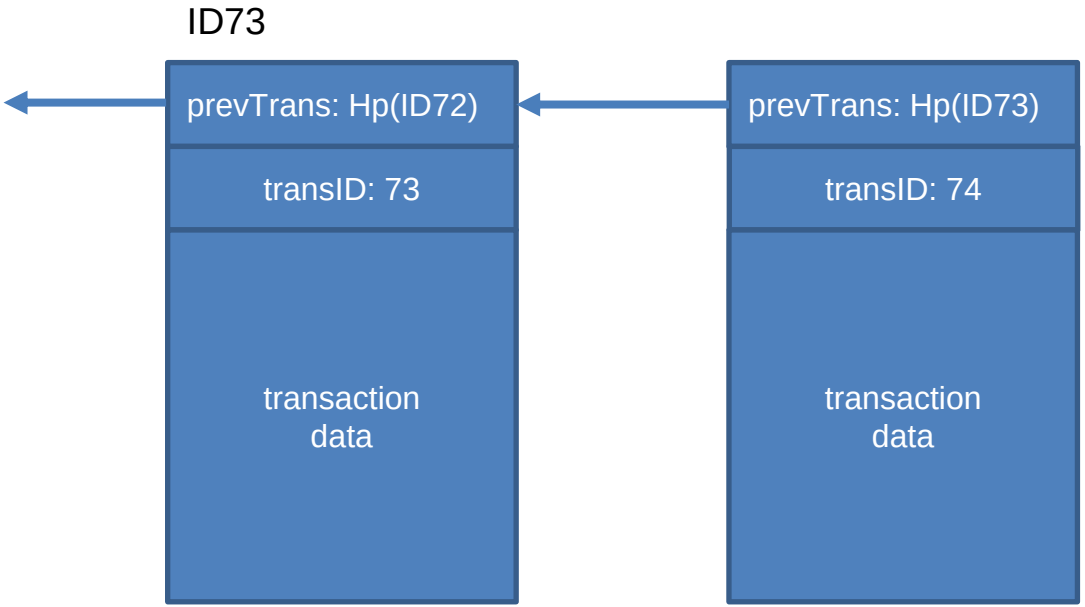
Estructura general

ID73



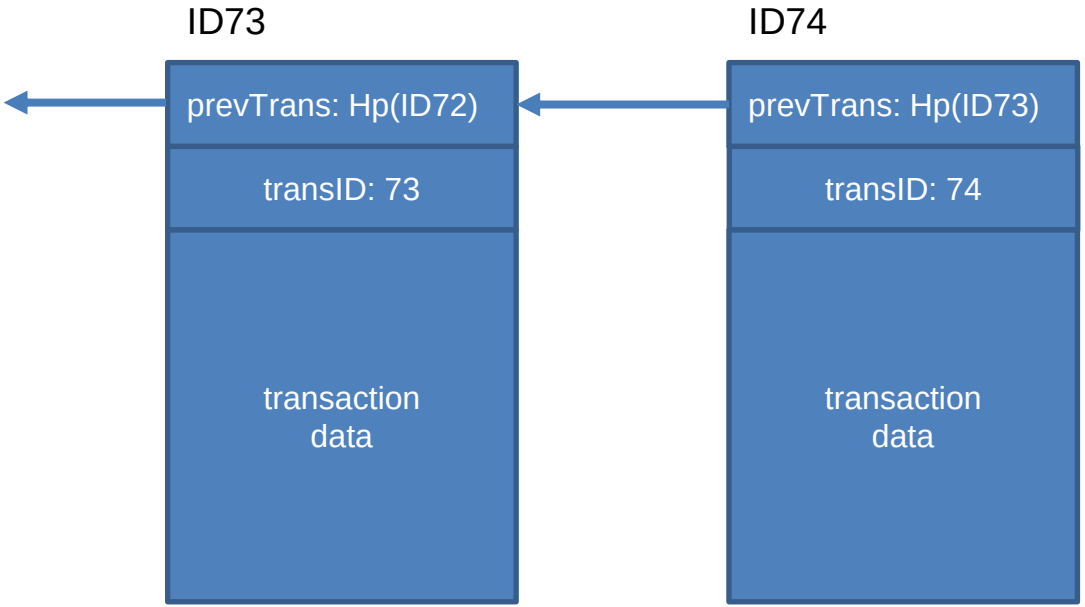
Scroogecoin

Estructura general



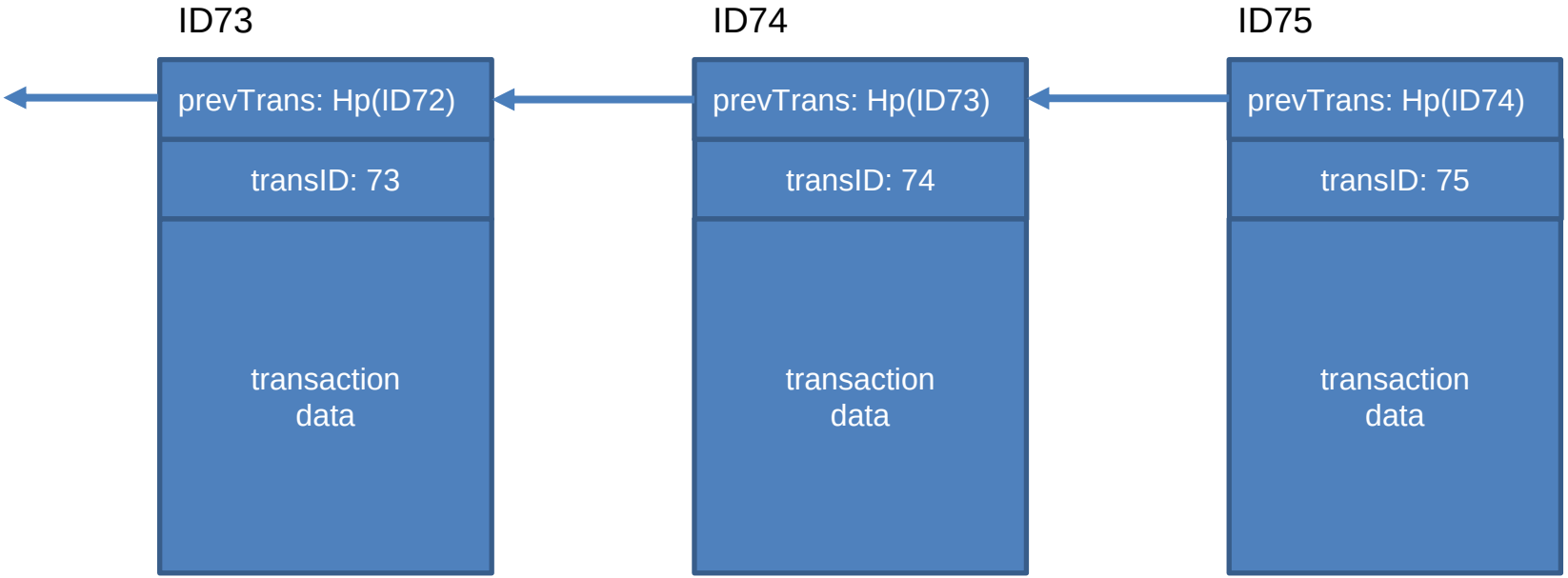
Scroogecoin

Estructura general



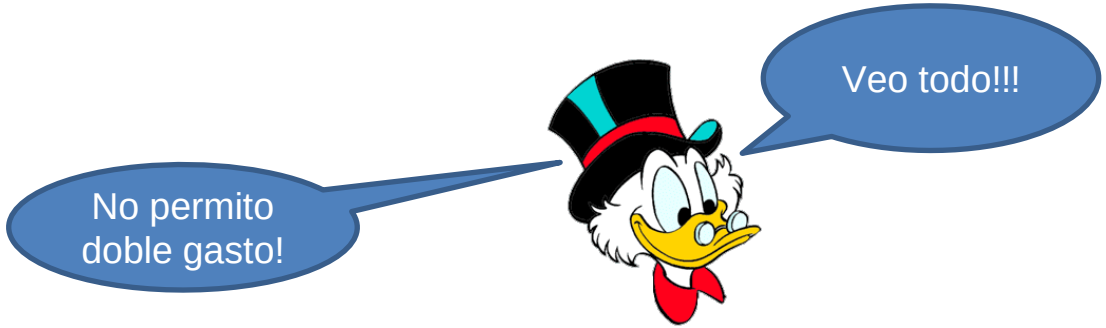
Scroogecoin

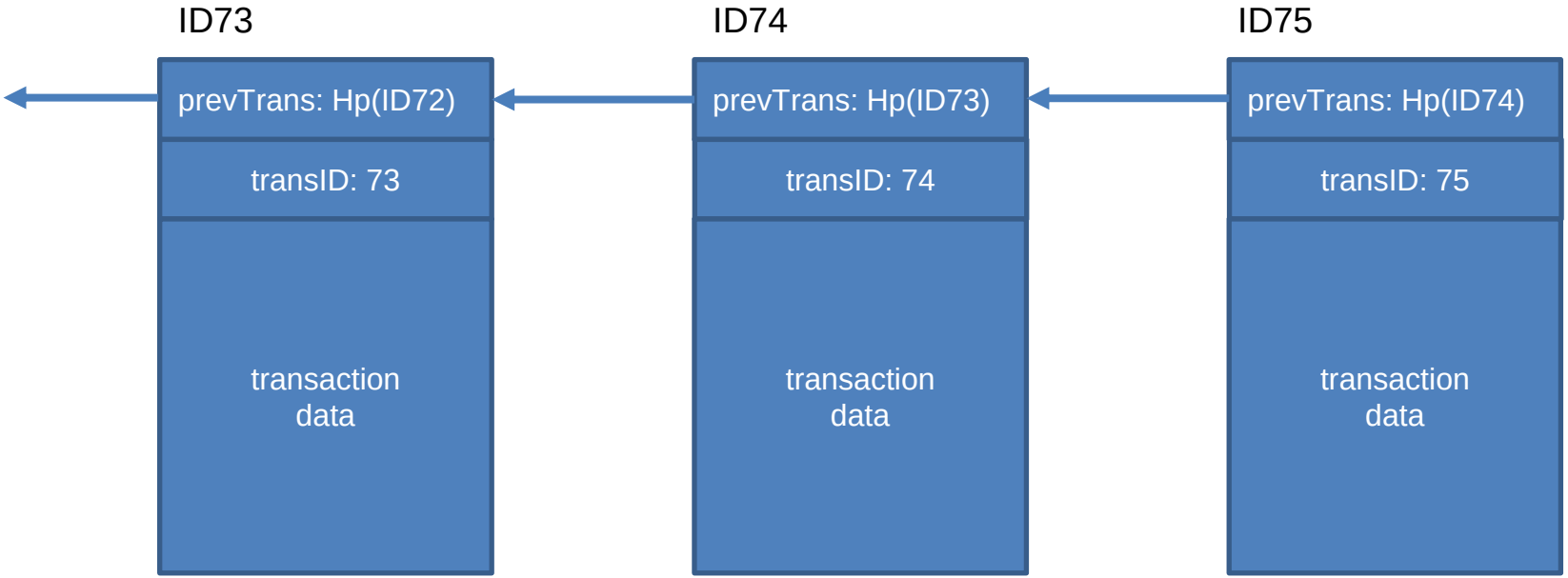
Estructura general



Scroogecoin

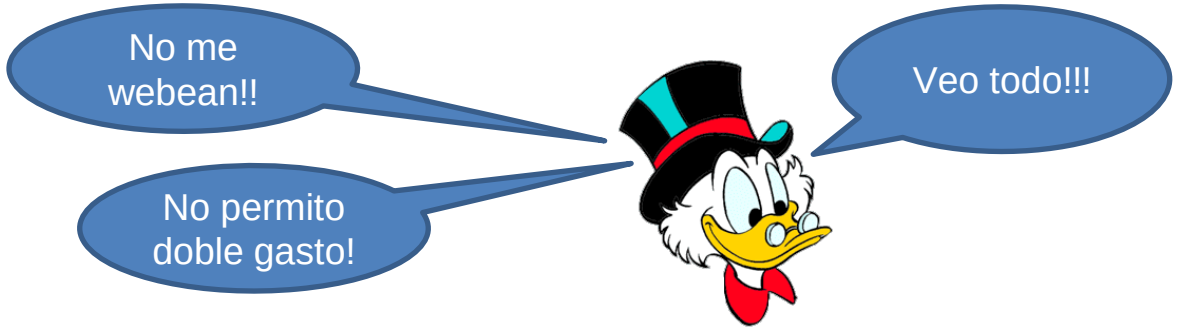
Estructura general

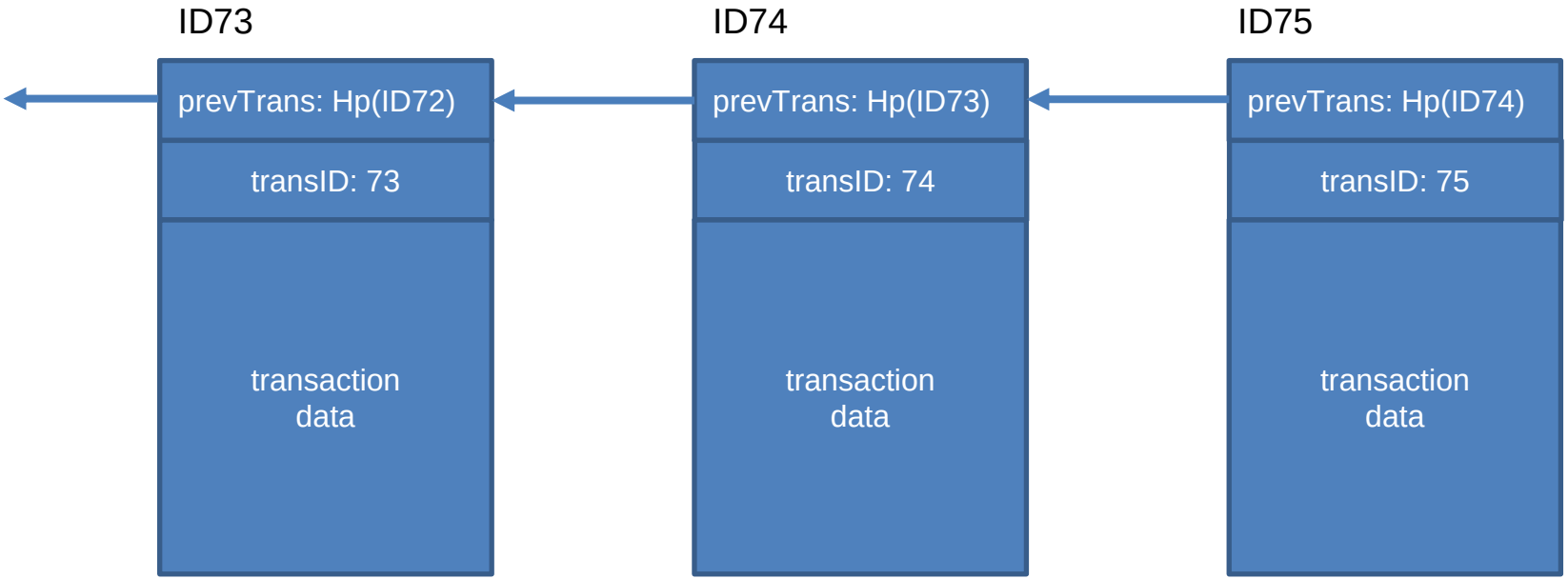




Scroogecoin

Estructura general



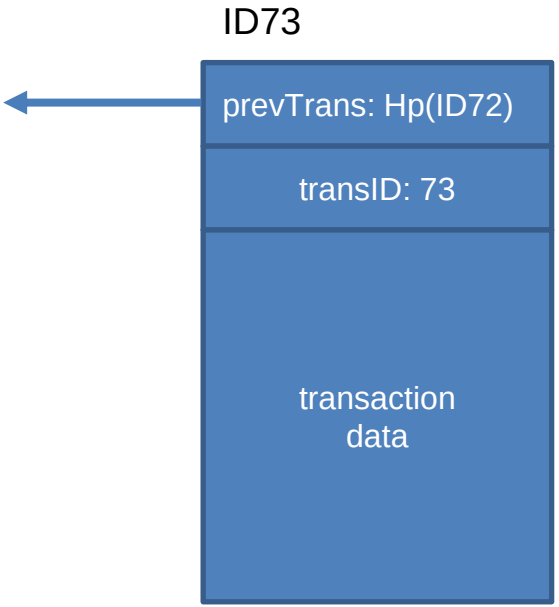


Importante

Scrooge firma los bloques

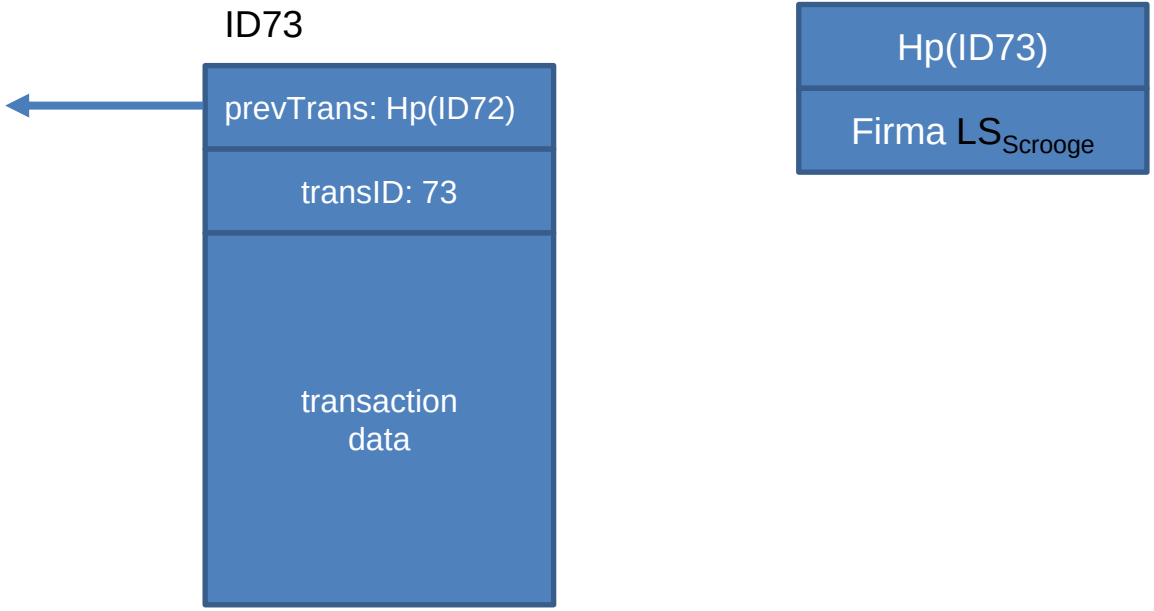
Importante

Scrooge firma los bloques



Importante

Scrooge firma los bloques



Scroogecoin

Reglas:

- Una transacción es valida solo si fue publicada en el Blockchain firmado por Scrooge
- Scrooge no incluye transacciones que intentan hacer el doble gasto
- Cada uno puede confirmar que la transacción es válida usando el Blockchain

Por qué necesitamos Blockchain? Basta si Scrooge firma solo transacciones?

Scroogecoin

Reglas:

- Una transacción es valida solo si fue publicada en el Blockchain firmado por Scrooge
- Scrooge no incluye transacciones que intentan hacer el doble gasto
- Cada uno puede confirmar que la transacción es válida usando el Blockchain

Por qué necesitamos Blockchain? Basta si Scrooge firma solo transacciones?

- Con Blockchain, podemos verificar que Scrooge no cambió transacciones nunca en el futuro (si tenemos algún hash pointer con su firma)

Scroogecoin

Casi como Goofycoin

Dos tipos de transacciones:

- CreateCoins (solo Scrooge)
- PayCoins (todos los participantes)
- Cada transacción tiene un ID único (su orden en el Libro contable)

Scrooge publica el Libro contable de la economía entera:

- Scrooge no va agregar una transacción que hace el doble gasto
- Vamos a usar Blockchain para asegurarse que Scrooge no está modificando las transacciones

Scroogecoin

CreateCoins transaction

Scroogecoin

CreateCoins transaction

transID: 73	type:CreateCoins

Scroogecoin

CreateCoins transaction

transID: 73	type:CreateCoins
coins created	

Scroogecoin

CreateCoins transaction

transID: 73	type:CreateCoins	
coins created		
num	value	recipient

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $\mathbf{LS}_{\mathbf{Scrooge}}$		

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

Creando más de un scroogecoin!!!

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

Para distintos destinatarios!!!

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

Con distinto valor!!!

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $\mathbf{LS}_{\mathbf{Scrooge}}$		

Firma de la transacción

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

coinID 73(0)

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

coinID 73(0)

coinID 73(1)

Scroogecoin

CreateCoins transaction

transID: 73		type:CreateCoins
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $LS_{Scrooge}$		

coinID 73(0)

coinID 73(1)

coinID 73(2)

Scroogecoin

CreateCoins transaction

CreateCoins transaction:

- **Válida si fue firmada por Scrooge**
- Scrooge puede crear cualquier cantidad de Scroogecoins en esta transacción
- Scrooge puede crear más que un Scroogecoin a mismo tiempo
- Cada Scroogecoin creado puede tener distinto destinatario (no solo Scrooge)

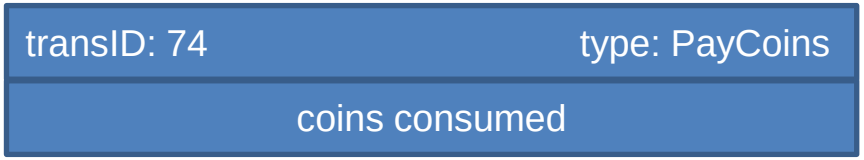
Scroogecoin

PayCoins transaction

transID: 74	type: PayCoins

Scroogecoin

PayCoins transaction



Scroogecoin

PayCoins transaction

transID: 74	type: PayCoins
coins consumed	
num	consumed coinID

Scroogecoin

PayCoins transaction

transID: 74	type: PayCoins
coins consumed	
num	consumed coinID
0	coinID 73(1)
1	coinID 73(2)

Scroogecoin

PayCoins transaction

transID: 74	type: PayCoins
coins consumed	
num	consumed coinID
0	coinID 73(1)
1	coinID 73(2)
coins created	

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…

Owned by Alice

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

Owned by Alice

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

Owned by Alice

Owned by Bob

Scroogecoin

PayCoins transaction

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
1	coinID 73(2)	
coins created		
num	value	recipient
0	3.2	0xf4…
1	1.7	0xa1…
2	4.6	0x55…
Signature by $\mathbf{LS_{Alice}}$		
Signature by $\mathbf{LS_{Bob}}$		

Owned by Alice

Owned by Bob

Scroogecoin

PayCoins transaction

PayCoins transaction:

- Consume Scroogecoins válidos (creados anteriormente en una transacción)
- Consume Scroogecoins que no han sido gastados anteriormente (doble gasto)
- El valor total de Scroogecoins consumidos es igual al valor de coins creados
- Todos los dueños de coins consumidos firmaron la transacción entera

Si se cumplen estas condiciones:

- Scrooge va a aceptar la transacción PayCoins
- Y la va a publicar en el Blockchain (y firmar el hash pointer)

Scroogecoin

Transacciones

Los dos tipos de transacciones en Scroogecoin:

- Crean nuevos Scroogecoins
- Consumen cero o más Scroogecoins
- Solo validas si eran publicadas en el Blockchain formado por Scrooge

Scroogecoin

Un Scroogecoin(de cualquier valor) es inmutable:

- Se crea una sola ves
- Se consume (gasta, destruye) una sola ves
- Después de esto existe solo como muestra de validez de otros Scroogecoins

Como dividir un Scroogecoin?

Scroogecoin

Un Scroogecoin(de cualquier valor) es inmutable:

- Se crea una sola ves
- Se consume (gasta, destruye) una sola ves
- Después de esto existe solo como muestra de validez de otros Scroogecoins

Como dividir un Scroogecoin?

- En una transacción que me paga a mi dos Scroogecoins del valor deseado

Scroogecoin

Qué logra Scroogecoin?

- Evita el doble gasto
- Las transacciones son inmutables (una ves hechas se quedan para siempre)
- No es necesario registrarse en el sistema (firmas digitales)
- Nadie puede falsificar un transacción (si no tiene nuestra llave privada)
- Nadie puede robarse nuestro Scroogecoin (sin mi llave privada)
- Nadie puede gastar plata que no tiene

Un problema de Scroogecoin

Scrooge:

- Un sistema completamente centralizado

Scrooge tiene mucho poder:

- No puede falsificar transacciones (blockchain)
- Puede crear cualquier cantidad de plata para si mismo
- Puede prohibir cierta gente participar en el sistema
- Puede demandar que cada transacción le pasa plata a él
- Pude aburrirse y abandonar el sistema

Qué es Bitcoin?

Bitcoin = Scroogecoin sin Scrooge

Qué es Bitcoin?

Nuestra próxima tarea:

- Remover al Scrooge
- **Descentralización**

Qué es Bitcoin?

Nuestra próxima tarea:

- Remover al Scrooge
- **Descentralización**

Pero antes de esto: hay que entender el Scroogecoin bien

- El mecanismo de Bitcoin es casi idéntico al mecanismo de Scroogecoin

UTXO

Unspent transaction output

Cuándo puedo gastar un Scroogecoin (en PayCoins):

- Si este Scroogecoin está válido
- Si mi Scroogecoin no fue gastado antes

Para verificar si el Scroogecoin está válido:

- Scrooge mantiene un pool de UTXOs
- Cuando le viene una transacción, chequea si los inputs están en el UTXO

UTXO

Unspent transaction output

En realidad (Bitcoin):

- Cada bloque contiene más de una transacción
- A Scrooge le llegan muchas para incluir en el próximo bloque
- Mucho más importante tener un pool de UTXOs
- Para chequear si todos los inputs están válidos

Si hay más transacciones en un bloque:

- Una puede referirse al otra (si están en el orden correcto) y gastar sus outputs
- Dos transacciones pueden intentar gastar el mismo Scroogecoin
- Cuál vamos a incluir?

UTXO

UTXO pool

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 73(1)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

UTXO

UTXO pool

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 73(1)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

Revisa todo el blockchain para ver si existe el coin y no fue gastado

UTXO

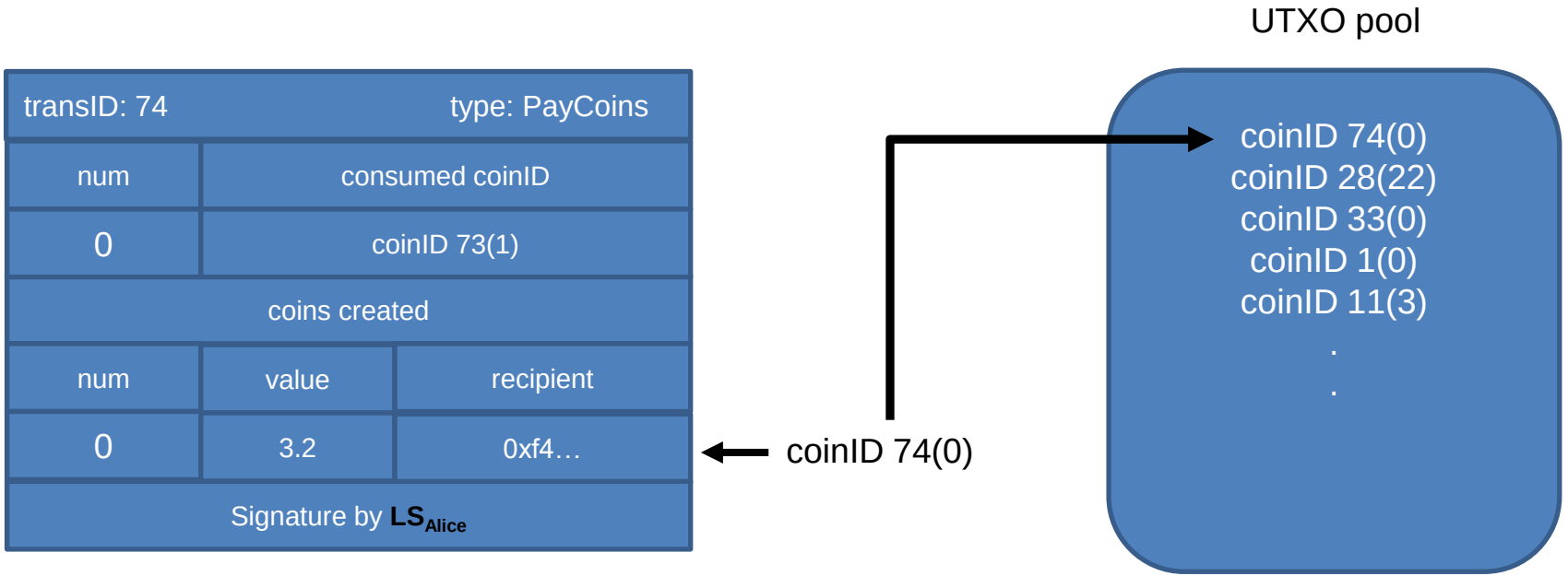
UTXO pool

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 73(1)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

← coinID 74(0)

UTXO

UTXO pool



Practice time!!!

Implementando Scroogecoin

Un UTXO pool:

coinID 74(0)
coinID 28(22)
coinID 33(0)
coinID 1(0)
coinID 11(3)
.
.
.

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 74(0)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by LS_{Alice}		

Practice time!!!

Implementando Scroogecoin

Un UTXO pool:

coinID 74(0)
coinID 28(22)
coinID 33(0)
coinID 1(0)
coinID 11(3)
.
.
.

Busca en el UTXO pool

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 74(0)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by LS_{Alice}		

Practice time!!!

Implementando Scroogecoin

Un UTXO pool:

coinID 74(0)
coinID 28(22)
coinID 33(0)
coinID 1(0)
coinID 11(3)
.
.
.

transID: 74		type: PayCoins
num	consumed coinID	
0	coinID 74(0)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by LS_{Alice}		

Busca en el UTXO pool

If OK, check the rest

Referencias

Libro Azul, capítulo 1.5