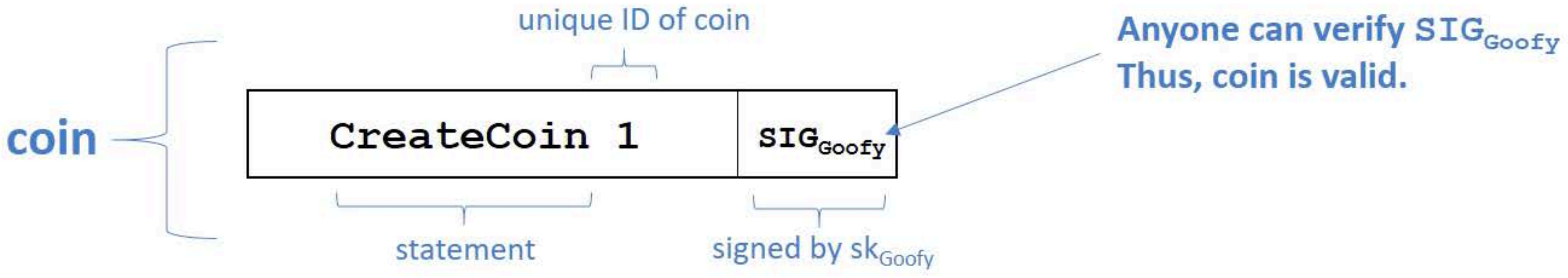
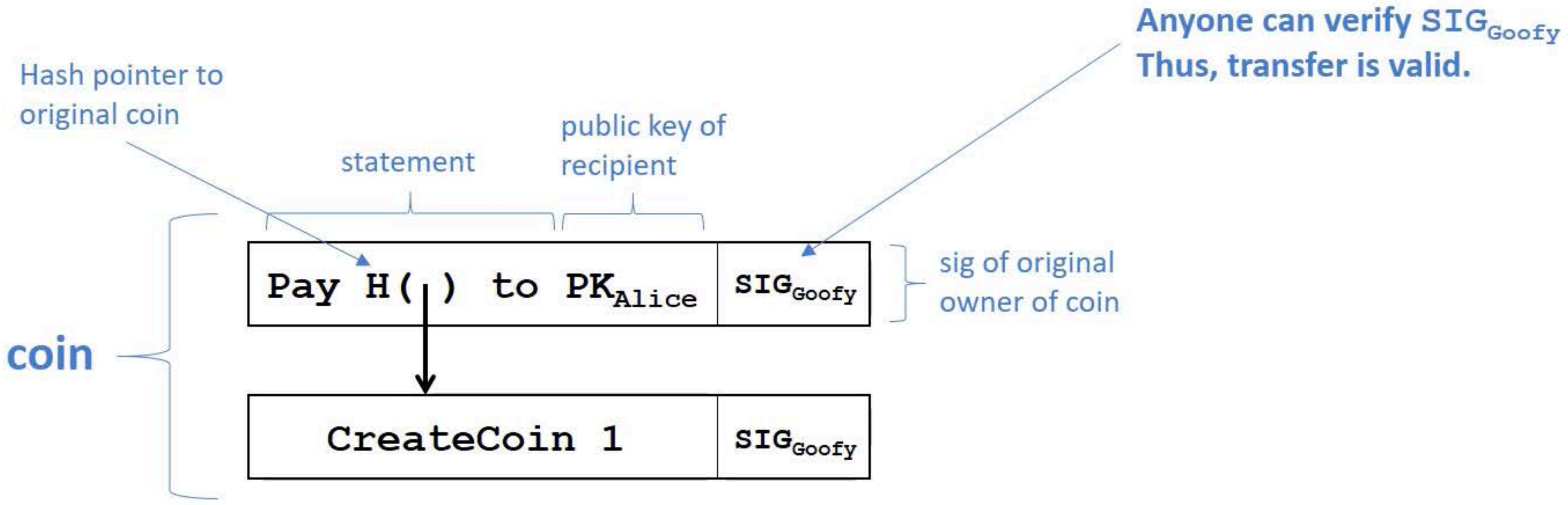
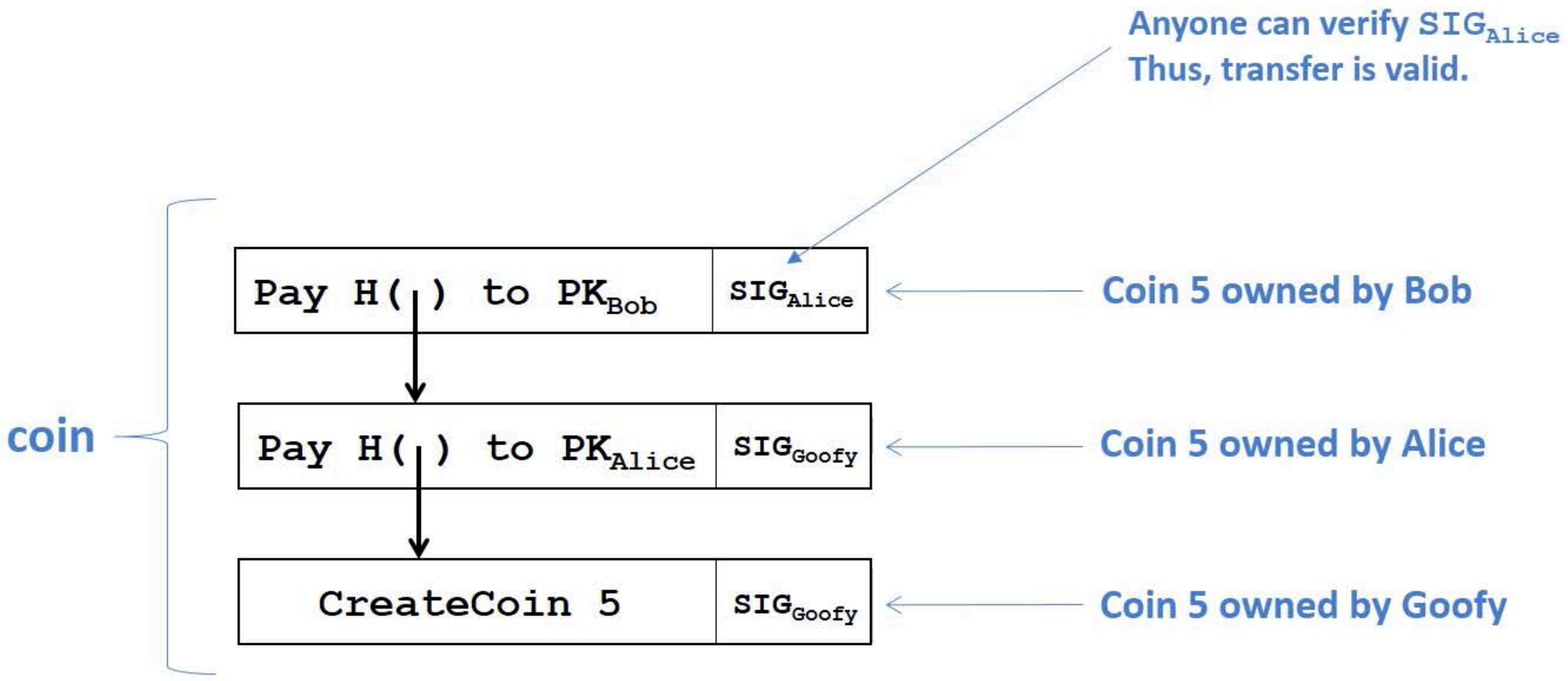
Goofy Coin
Simple Cryptocurrency

Rule 1: A designated entity, Goofy, can create new coins whenever he wants and these newly created coins belong to him.

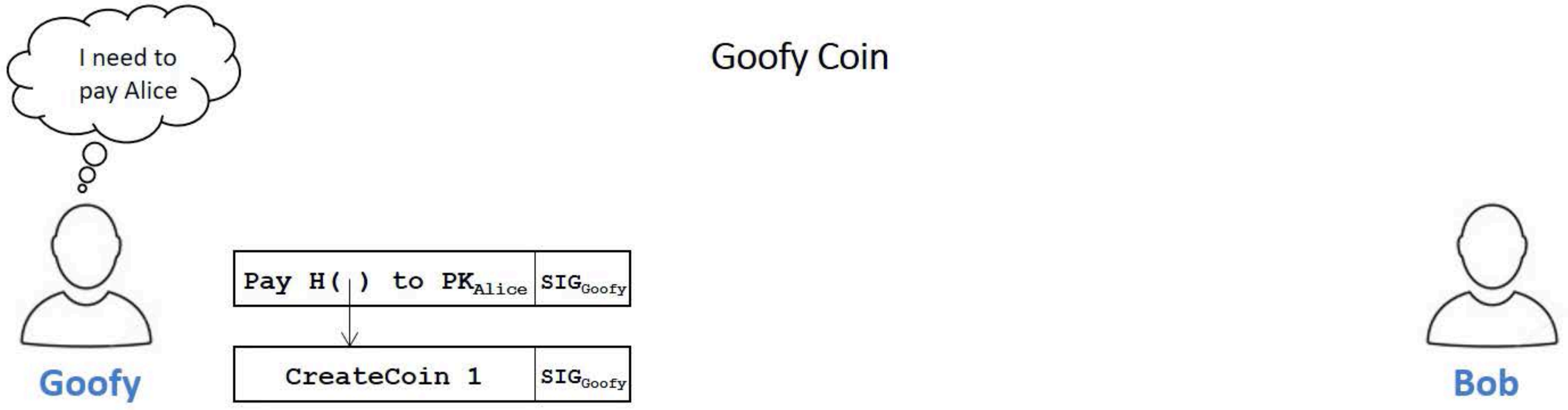

unique ID of coin
coin
CreateCoin 1
$\text{SIG}_{\text{Goofy}}$
statement
signed by sk_{Goofy}
Anyone can verify $\text{SIG}_{\text{Goofy}}$
Thus, coin is valid.

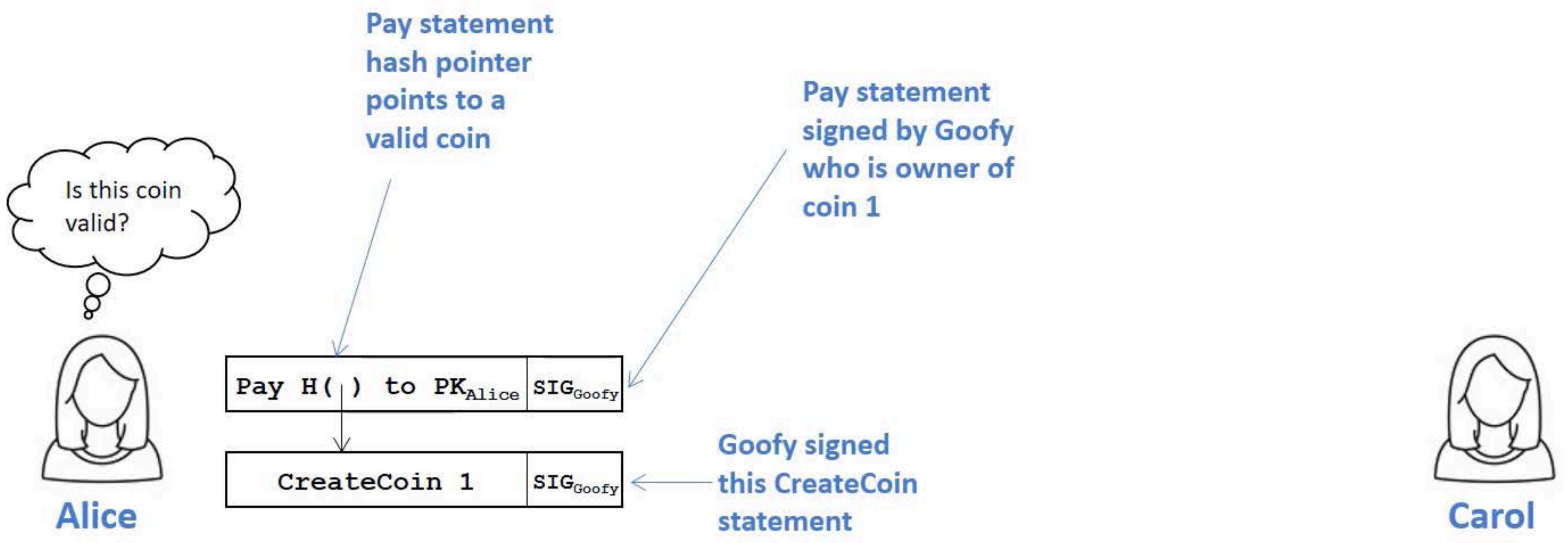
Rule 2: Whoever owns a coin can transfer it to someone else.


Anyone can verify $\text{SIG}_{\text{Goofy}}$
Thus, transfer is valid.
Hash pointer to original coin
statement
public key of recipient
Pay H( ) to PK_{Alice}
$\text{SIG}_{\text{Goofy}}$
sig of original owner of coin
coin
CreateCoin 1
$\text{SIG}_{\text{Goofy}}$

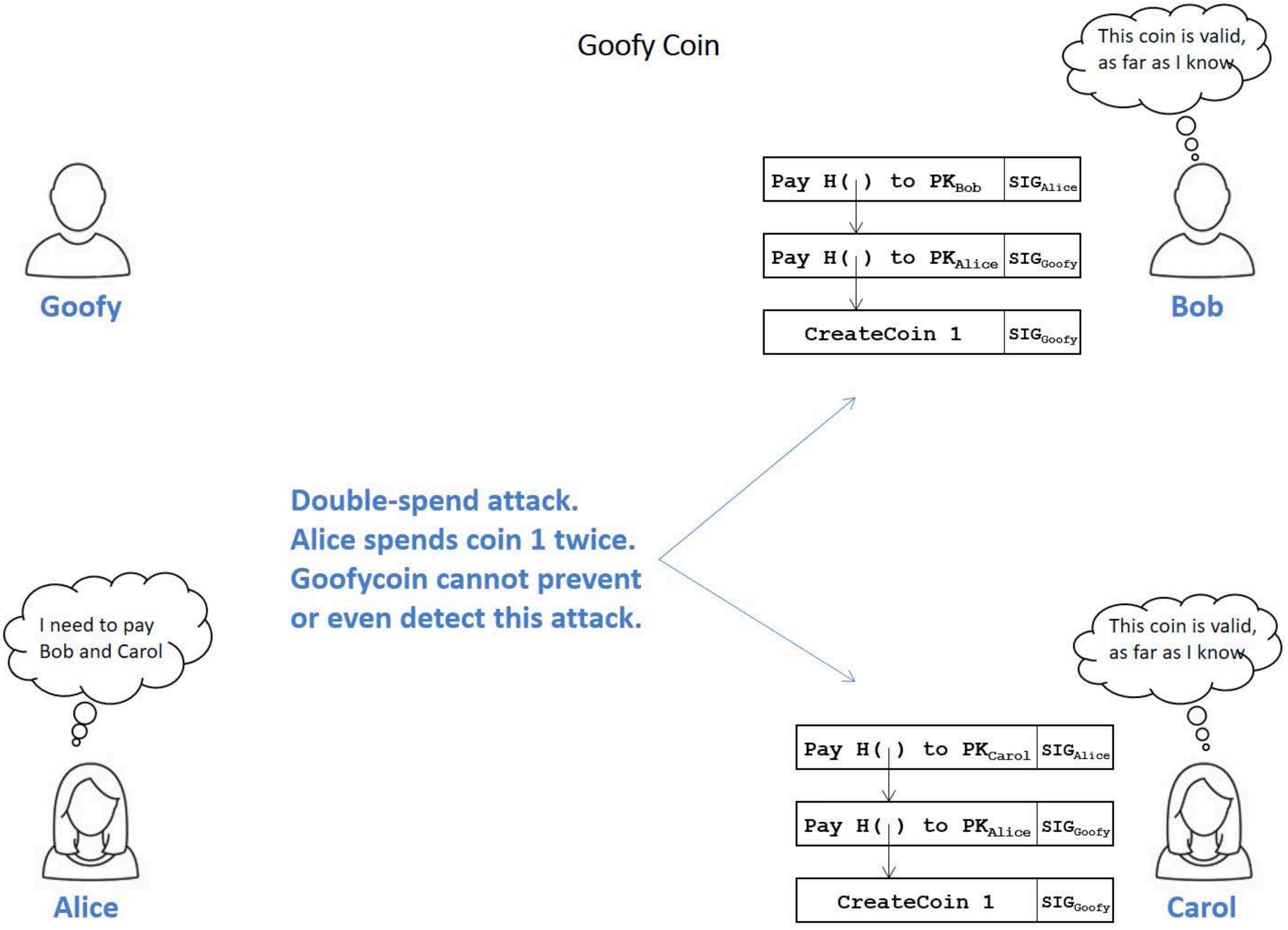

Anyone can verify SIG_Alice
Thus, transfer is valid.
Pay H( ) to PK_Bob
SIG_Alice
Coin 5 owned by Bob
coin
Pay H( ) to PK_Alice
SIG_Goofy
Coin 5 owned by Alice
CreateCoin 5
SIG_Goofy
Coin 5 owned by Goofy

Goofy Coin


I need to pay Alice
Pay H( ) to PK_Alice
SIG_Goofy
CreateCoin 1
SIG_Goofy
Goofy
Bob


Pay statement hash pointer points to a valid coin
Pay statement signed by Goofy who is owner of coin 1
Is this coin valid?
Pay H( ) to PK_Alice
SIG_Goofy
CreateCoin 1
SIG_Goofy
Goofy signed this CreateCoin statement
Alice
Carol

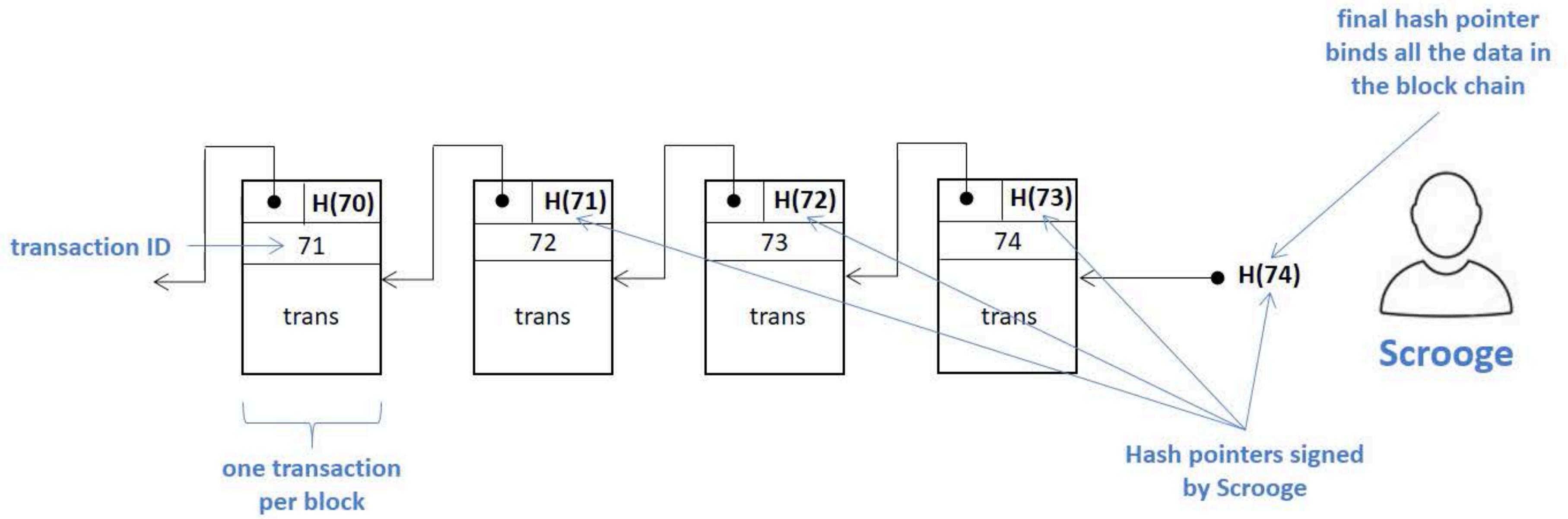
Goofy Coin


Goofy
Bob
This coin is valid, as far as I know
Pay H( ) to PK$_{Bob}$ | SIG$_{Alice}$
Pay H( ) to PK$_{Alice}$ | SIG$_{Goofy}$
CreateCoin 1 | SIG$_{Goofy}$
I need to pay Bob and Carol
Alice
Pay H( ) to PK$_{Carol}$ | SIG$_{Alice}$
Pay H( ) to PK$_{Alice}$ | SIG$_{Goofy}$
CreateCoin 1 | SIG$_{Goofy}$
This coin is valid, as far as I know
Carol

Double-spend attack.
Alice spends coin 1 twice.
Goofycoin cannot prevent or even detect this attack.

Scrooge Coin
Preventing double-spending attacks

- **Key Idea:** A designated entity, Scrooge, publishes an append-only ledger containing the history of all transactions.

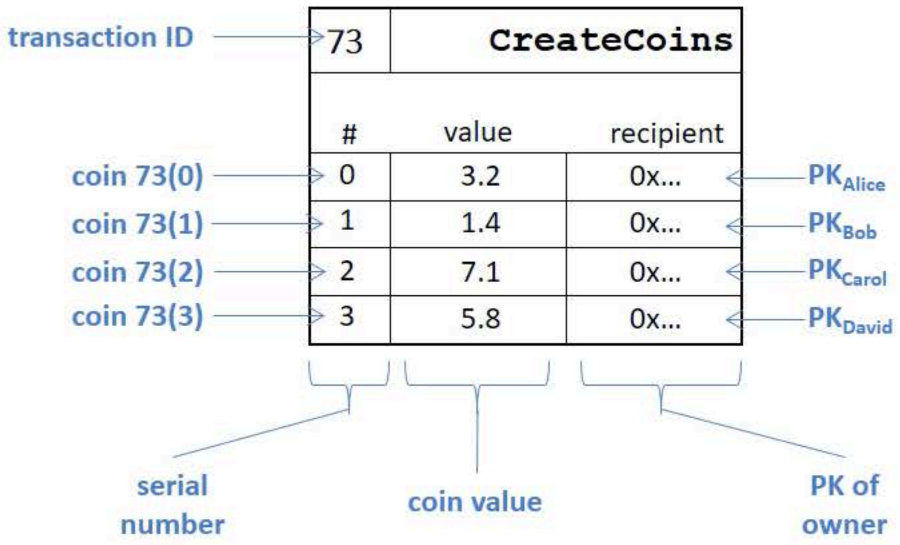


final hash pointer binds all the data in the block chain
H(70)
H(71)
H(72)
H(73)
transaction ID
71
72
73
74
H(74)
trans
trans
trans
trans
Scrooge
one transaction per block
Hash pointers signed by Scrooge

1. A transaction only counts if it is in the block chain signed by Scrooge.
2. Scrooge makes sure that he doesn't endorse a transaction that attempts to double-spend an already spent count.

Scrooge Coin
Transactions

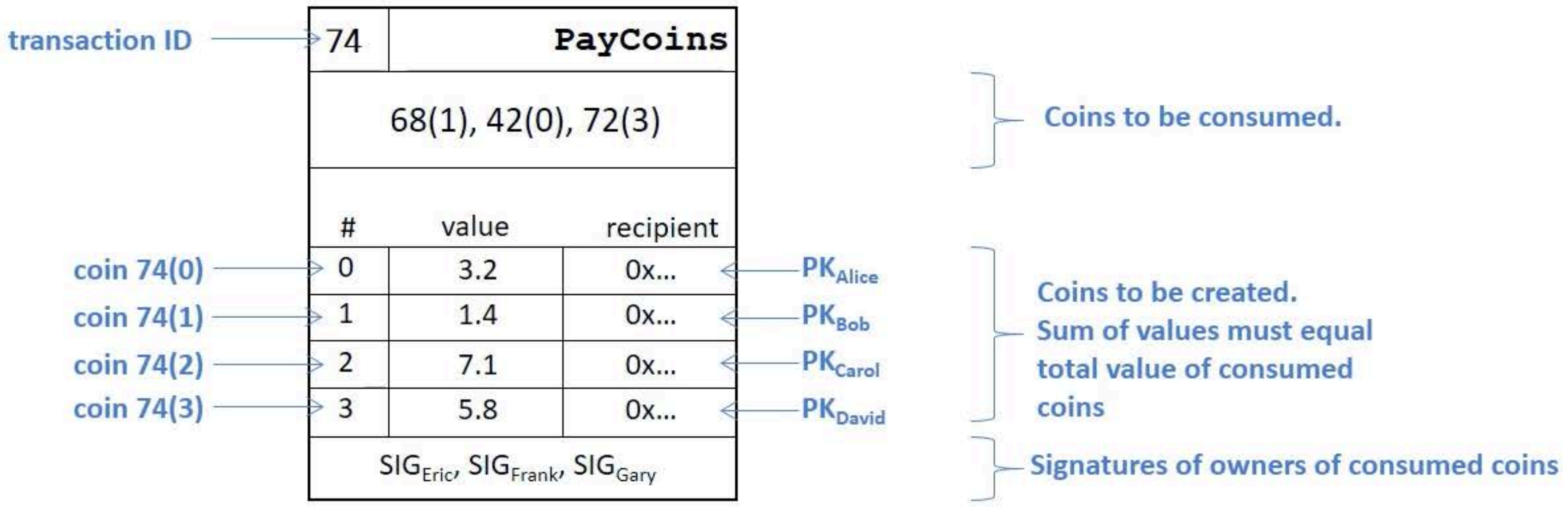
- `CreateCoins` transaction creates multiple coins and assigns each of them to a recipient. Only Scrooge can create coins.



73		CreateCoins
#	value	recipient
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...
3	5.8	0x...

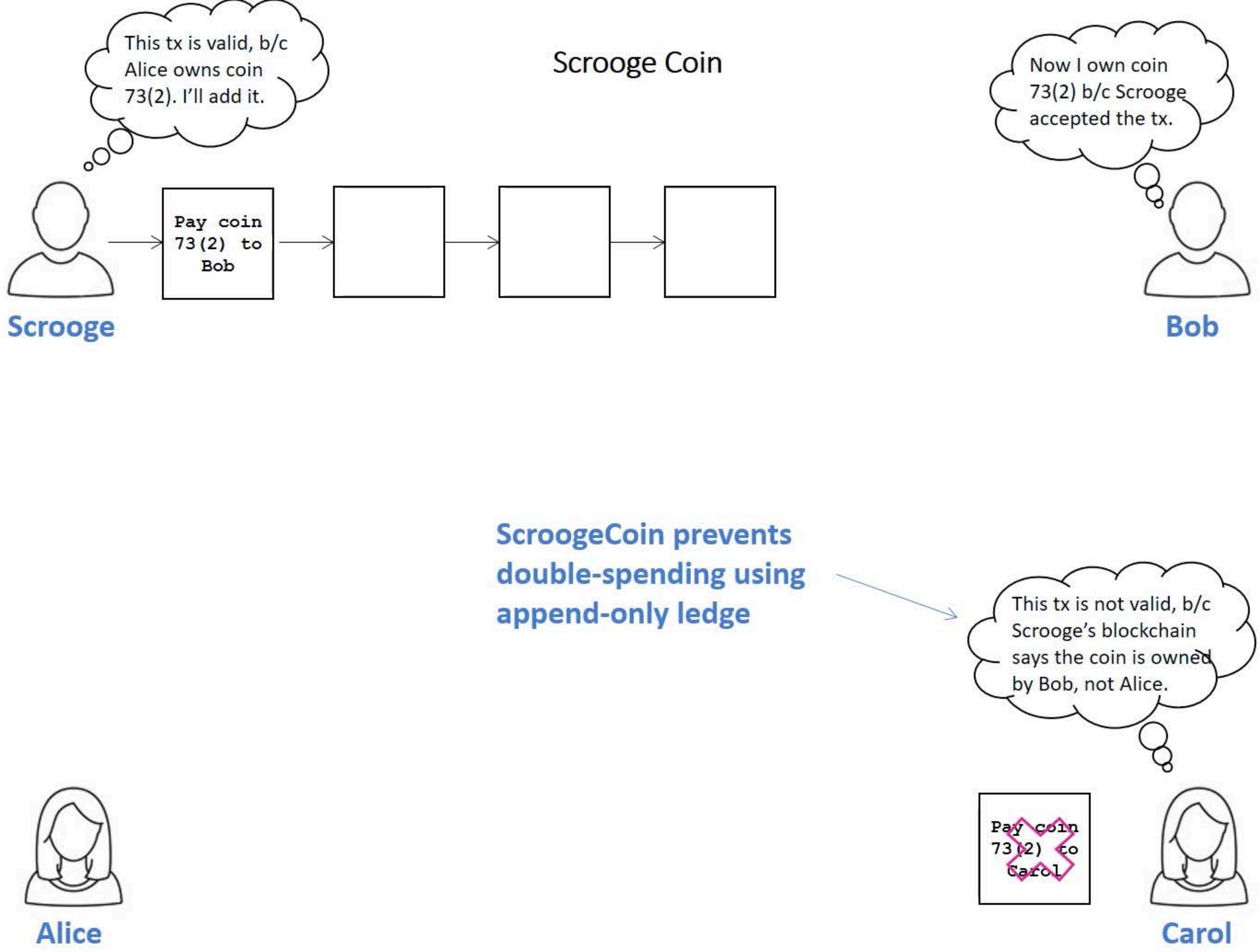
Scrooge Coin
Transactions

- `PayCoins` transaction consumes/destroys some coins and create new coins of the same total value. Transaction has to be signed by everyone who's paying in a coin.



- `PayCoins` transaction is valid if it satisfies four conditions
  1. The consumed are valid, that is they were created in previous transactions.
  2. The consumed coins have not already been consumed in some previous transaction.
  3. The total value of the coins that created is equal to the total value of the coins consumed.
  4. The transaction is signed by the owners of all coins consumed in transaction.

Scrooge Coin


This tx is valid, b/c Alice owns coin 73(2). I'll add it.
Pay coin 73(2) to Bob
Now I own coin 73(2) b/c Scrooge accepted the tx.
Scrooge
Bob
This tx is not valid, b/c Scrooge's blockchain says the coin is owned by Bob, not Alice.
Pay coin 73(2) to Carol
Alice
Carol

ScroogeCoin prevents double-spending using append-only ledge

Scrooge Coin
Evaluation

- Benefits of ScroogeCoin
  1. People can see which coins are valid
  2. Prevents double-spending
  3. Scrooge can't create fake transactions because he can't forge signatures

- Weakness of ScroogeCoin
  1. Scrooge has too much influence
  2. Scrooge could stop endorsing transactions from some users, making their coins unspendable.
  3. Scrooge could refuse to publish a transaction unless the user pays a transaction free.
  4. Scrooge could create as many new coins for himself as he wants.
  5. Scrooge could abandon the whole system and stop updating the blockchain completely.

- Key question
  - Is it possible to eliminate the need for a central trusted authority from a cryptocurrency?

Scenario A

Suppose the transactions 1 and 2 have been accepted by Scrooge and thus added to the ledger. If transaction 3 is submitted to Scrooge, will Scrooge accept it and add it to his ledger?

1	**CreateCoins**	
#	value	recipient
0	3.2	0x... PK_{Alice}
1	1.4	0x... PK_{Bob}
2	7.1	0x... PK_{Carol}
3	5.8	0x... PK_{David}

2	**PayCoins**	
1(1), 1(3)		
#	value	recipient
0	1.6	0x... PK_{Eric}
1	0.4	0x... PK_{Frank}
2	5.1	0x... PK_{Gary}
3	0.1	0x... PK_{Howard}
SIG_{Bob}, SIG_{David}		

3	**PayCoins**	
1(0), 2(0)		
#	value	recipient
0	3.4	0x... PK_{Ivan}
1	1.4	0x... PK_{Jack}
SIG_{Ivan}, SIG_{Jack}		

Already accepted by Scrooge on his ledger

Will Scrooge accept TX 3?

Scenario B

Suppose the transactions 1 and 2 have been accepted by Scrooge and thus added to the ledger. If transaction 3 is submitted to Scrooge, will Scrooge accept it and add it to his ledger?

1	CreateCoins	
#	value	recipient
0	3.2	0x… PK_{Alice}
1	1.4	0x… PK_{Bob}
2	7.1	0x… PK_{Carol}
3	5.8	0x… PK_{David}

2	PayCoins	
	1(1), 1(3)	
#	value	recipient
0	1.6	0x… PK_{Eric}
1	0.4	0x… PK_{Frank}
2	5.1	0x… PK_{Gary}
3	0.1	0x… PK_{Howard}
	SIG_{Bob}, SIG_{David}	

3	PayCoins	
	1(2), 2(3)	
#	value	recipient
0	6.8	0x… PK_{Ivan}
1	5.3	0x… PK_{Jack}
	SIG_{Carol}, SIG_{Howard}	

Already accepted by Scrooge on his ledger

Will Scrooge accept TX 3?

Scenario C

Suppose the transactions 1 and 2 have been accepted by Scrooge and thus added to the ledger. If transaction 3 is submitted to Scrooge, will Scrooge accept it and add it to his ledger?

1	CreateCoins	
#	value	recipient
0	3.2	0x... PK_{Alice}
1	1.4	0x... PK_{Bob}
2	7.1	0x... PK_{Carol}
3	5.8	0x... PK_{David}

2	PayCoins	
	1(1), 1(3)	
#	value	recipient
0	1.6	0x... PK_{Eric}
1	0.4	0x... PK_{Frank}
2	5.1	0x... PK_{Gary}
3	0.1	0x... PK_{Howard}
	SIG_{Bob}, SIG_{David}	

3	PayCoins	
	1(3), 2(1)	
#	value	recipient
0	6.2	0x... PK_{Ivan}
	SIG_{David}, SIG_{Frank}	

Already accepted by Scrooge on his ledger

Will Scrooge accept TX 3?

Revisited: Scenario C

Suppose the transactions 1 and 2 have been accepted by Scrooge and thus added to the ledger. If transaction 3 is submitted to Scrooge, will Scrooge accept it and add it to his ledger?

1	`CreateCoins`	
#	value	recipient
0	3.2	0x... PK_{Alice}
1	1.4	0x... PK_{Bob}
2	7.1	0x... PK_{Carol}
3	5.8	0x... PK_{David}

2	`PayCoins`	
	1(1), 1(3)	
#	value	recipient
0	1.6	0x... PK_{Eric}
1	0.4	0x... PK_{Frank}
2	5.1	0x... PK_{Gary}
3	0.1	0x... PK_{Howard}
	SIG_{Bob}, SIG_{David}	

3	`PayCoins`	
	1(3), 2(1)	
#	value	recipient
0	6.2	0x... PK_{Ivan}
	SIG_{David}, SIG_{Frank}	

Already accepted by Scrooge on his ledger

UTXO: {1(0), 1(2), 2(0), 2(1), 2(2), 2(3)}

Will Scrooge accept TX 3?

Must check that every input is in the UTXO set of the current ledger