paddyncl

Atomically Trading with Roger: Gambling on the success of a hardfork*

Patrick McCorry, Ethan Heilman, Andrew Miller

*Acknowledgement: We thank Roger Ver for letting us use his name in the paper’s title.

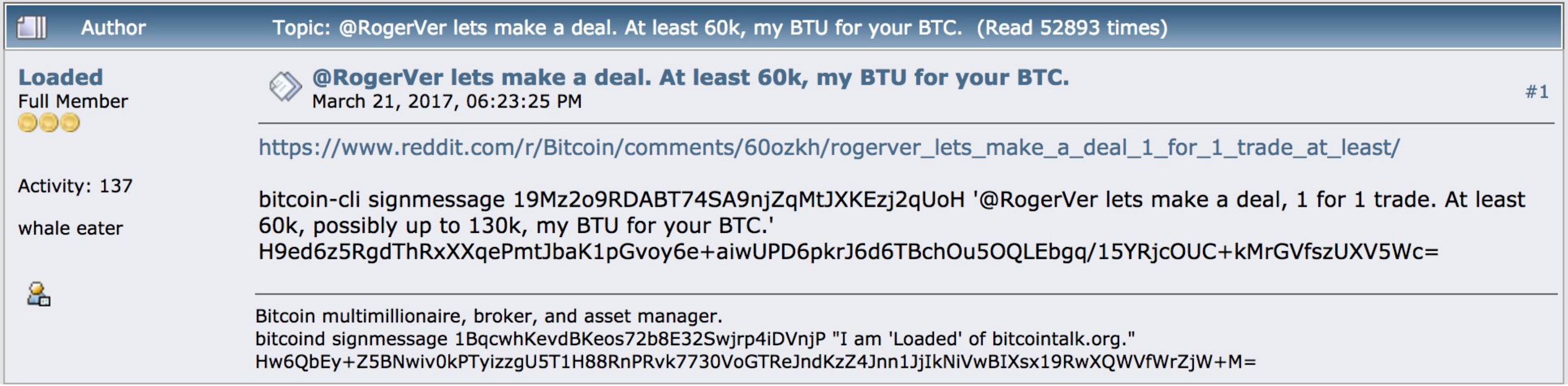
What is interesting in the paper?

- Brief history on soft and hard forks in Bitcoin/Ethereum.
- An overview of replay protection proposals (including a new one we call migration inputs)
- **This Talk: Hard Fork Atomic Trade Protocols for Bitcoin**
  - **How to set up trade prior to hardfork and perform it once hardfork occurs.**
  - **With and Without a transaction malleability fix!**
- Hard Fork Atomic Trade Protocol for Ethereum
  - How to use a Hardfork Oracle to set up and perform the atomic trade.

I hope to leave everyone with one message:

Transaction malleability *was* a pain in the ass and designing bitcoin contracts that accounts for malleability is non-trivial.

Loaded Challenges Roger (and he accepts)



Author | Topic: @RogerVer lets make a deal. At least 60k, my BTU for your BTC. (Read 52893 times)

Loaded
Full Member
●●●

Activity: 137

whale eater

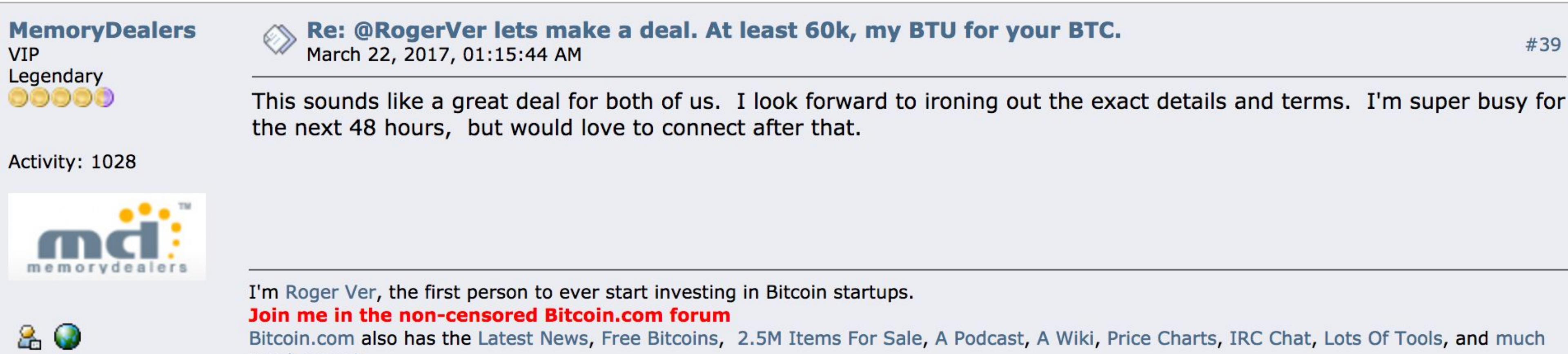
@RogerVer lets make a deal. At least 60k, my BTU for your BTC.
March 21, 2017, 06:23:25 PM

#1

https://www.reddit.com/r/Bitcoin/comments/60ozkh/rogerver_lets_make_a_deal_1_for_1_trade_at_least/

bitcoin-cli signmessage 19Mz2o9RDABT74SA9njZqMtJXKEzj2qUoH '@RogerVer lets make a deal, 1 for 1 trade. At least 60k, possibly up to 130k, my BTU for your BTC.'
H9ed6z5RgdThRxXXqePmtJbaK1pGvoy6e+aiwUPD6pkrJ6d6TBchOu5OQLEbgq/15YRjcOUC+kMrGVfszUXV5Wc=

Bitcoin multimillionaire, broker, and asset manager.
bitcoind signmessage 1BqcwhKevdBKeos72b8E32Swjrp4iDVnjP "I am 'Loaded' of bitcointalk.org."
Hw6QbEy+Z5BNwiv0kPTyizzgU5T1H88RnPRvk7730VoGTReJndKzZ4Jnn1JjIkNiVwBIXsx19RwXQWVfWrZjW+M=



MemoryDealers
VIP
Legendary
●●●●●

Activity: 1028

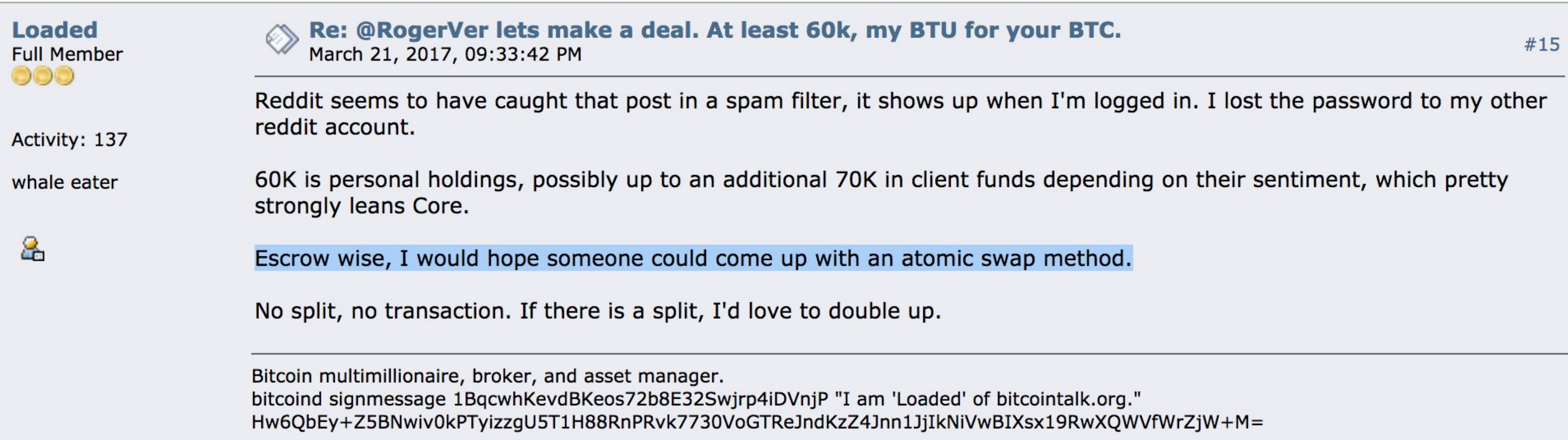
Re: @RogerVer lets make a deal. At least 60k, my BTU for your BTC.
March 22, 2017, 01:15:44 AM

#39

This sounds like a great deal for both of us. I look forward to ironing out the exact details and terms. I'm super busy for the next 48 hours, but would love to connect after that.

I'm Roger Ver, the first person to ever start investing in Bitcoin startups.
Join me in the non-censored Bitcoin.com forum
Bitcoin.com also has the Latest News, Free Bitcoins, 2.5M Items For Sale, A Podcast, A Wiki, Price Charts, IRC Chat, Lots Of Tools, and much

Loaded didn’t want to use an escrow.



Loaded
Full Member
●●●

Activity: 137

whale eater

Re: @RogerVer lets make a deal. At least 60k, my BTU for your BTC.
March 21, 2017, 09:33:42 PM

#15

Reddit seems to have caught that post in a spam filter, it shows up when I'm logged in. I lost the password to my other reddit account.

60K is personal holdings, possibly up to an additional 70K in client funds depending on their sentiment, which pretty strongly leans Core.

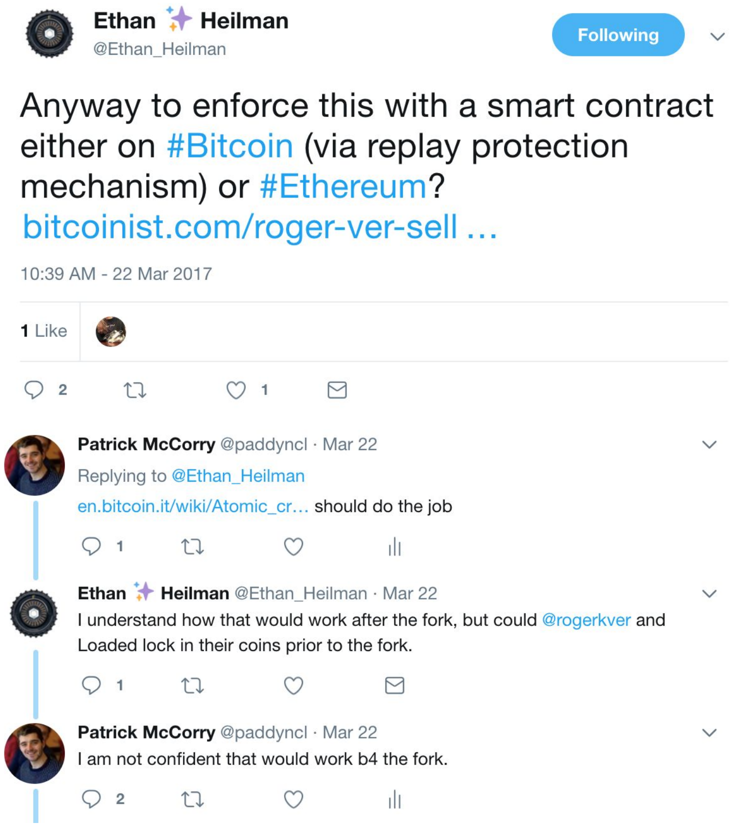
Escrow wise, I would hope someone could come up with an atomic swap method.

No split, no transaction. If there is a split, I'd love to double up.

Bitcoin multimillionaire, broker, and asset manager.
bitcoind signmessage 1BqcwhKevdBKeos72b8E32Swjrp4iDVnjP "I am 'Loaded' of bitcointalk.org."
Hw6QbEy+Z5BNwiv0kPTyizzgU5T1H88RnPRvk7730VoGTReJndKzZ4Jnn1JjIkNiVwBIXsx19RwXQWVfWrZjW+M=

paddyncl

Eventually.. I seen Ethan tweeting about the bet…



Ethan ✨ Heilman
@Ethan_Heilman

Anyway to enforce this with a smart contract either on #Bitcoin (via replay protection mechanism) or #Ethereum? bitcoinist.com/roger-ver-sell …

10:39 AM - 22 Mar 2017

1 Like

2 1

Patrick McCorry @paddyncl · Mar 22
Replying to @Ethan_Heilman
en.bitcoin.it/wiki/Atomic_cr… should do the job

1

Ethan ✨ Heilman @Ethan_Heilman · Mar 22
I understand how that would work after the fork, but could @rogerkver and Loaded lock in their coins prior to the fork.

1

Patrick McCorry @paddyncl · Mar 22
I am not confident that would work b4 the fork.

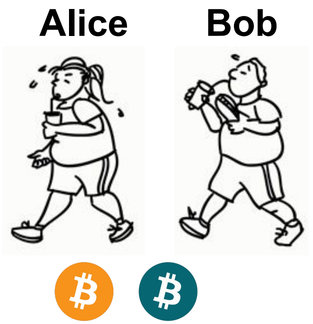
2

paddyncl

Atomically Trade across two forks

Alice **Bob**

Atomically Trade across two forks



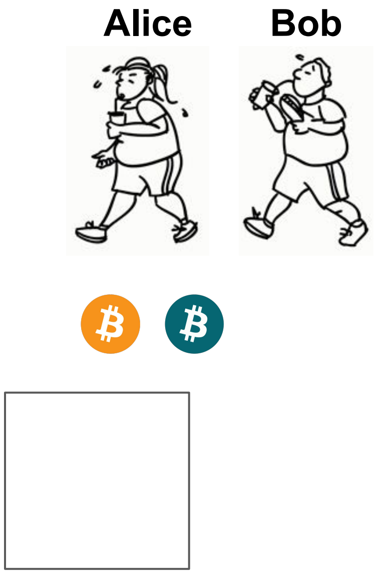
1. Deposit coins into a single transaction.

● Alices Deposit

● Bobs Deposit

paddyncl

Atomically Trade across two forks



1. Deposit coins into a single transaction.

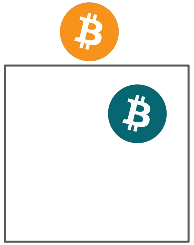
● Alices Deposit

● Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

₿ Alices Deposit

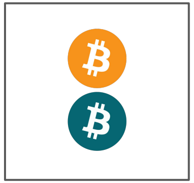
₿ Bobs Deposit

paddyncl

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

● Alices Deposit

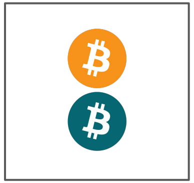
● Bobs Deposit

paddyncl

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

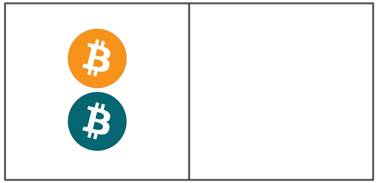
₿ Alices Deposit

₿ Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

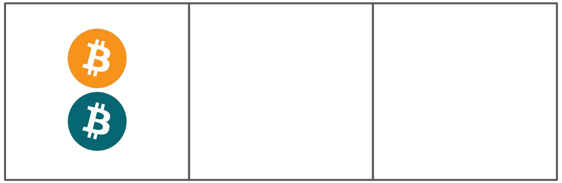
₿ Alices Deposit

₿ Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

₿ Alices Deposit

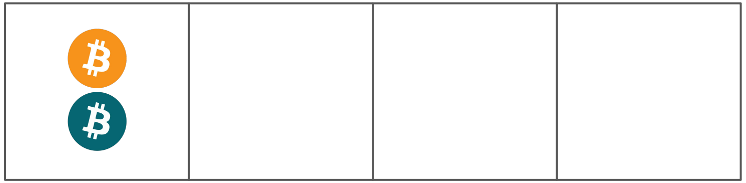
₿ Bobs Deposit

paddyncl

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.

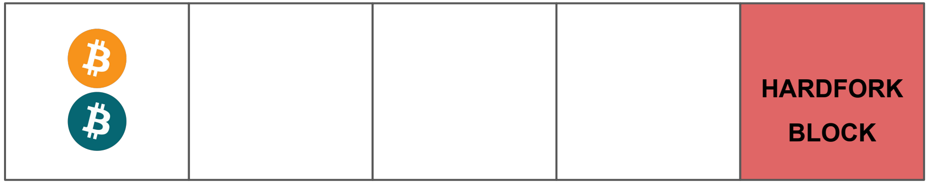
₿ Alices Deposit

₿ Bobs Deposit

Atomically Trade across two forks

Alice

Bob



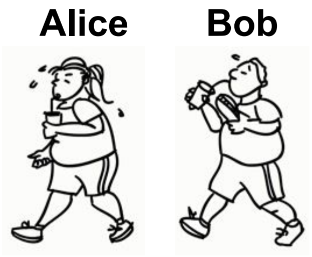
1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES

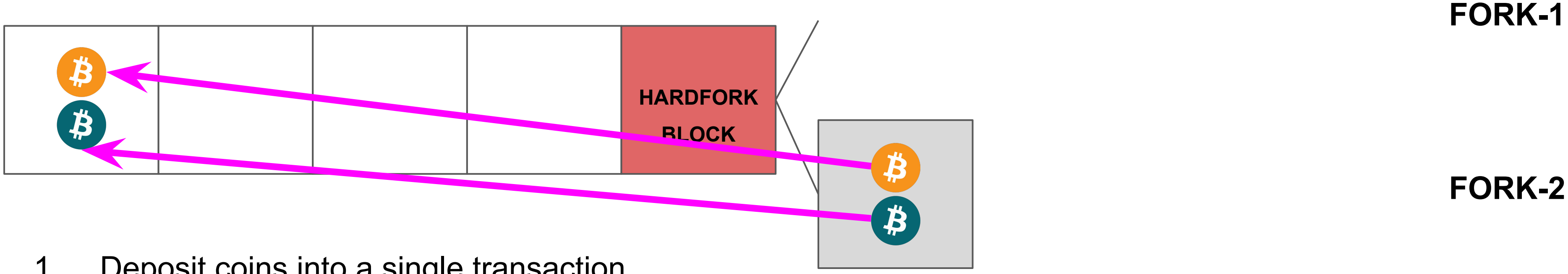
● Alices Deposit

● Bobs Deposit

paddyncl

Atomically Trade across two forks





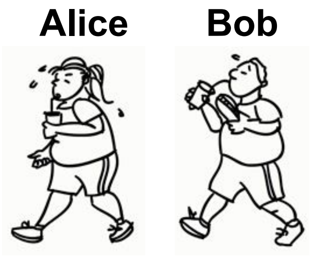
1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES
3. Alice withdraws both coins in FORK-2

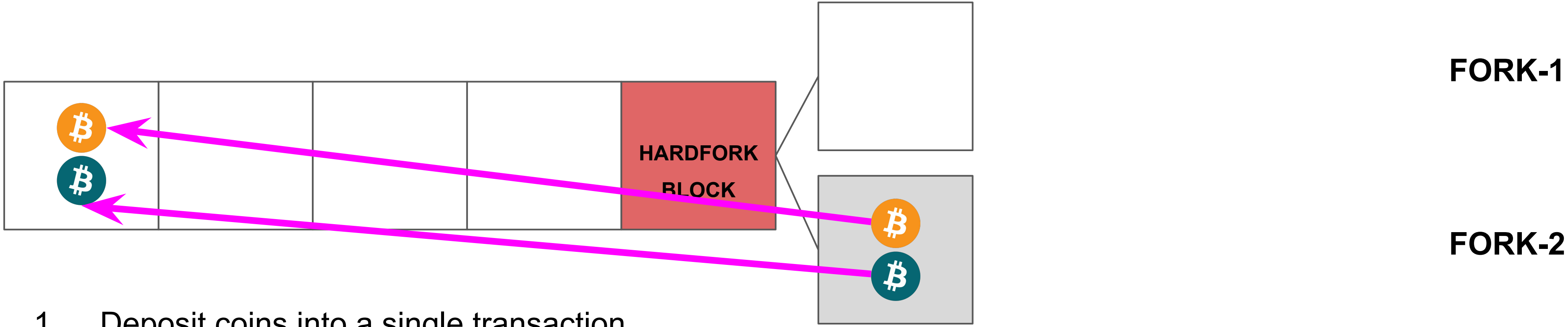
● Alices Deposit

● Bobs Deposit

paddyncl

Atomically Trade across two forks





1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES
3. Alice withdraws both coins in FORK-2
4. Bob withdraws both coins in FORK-1

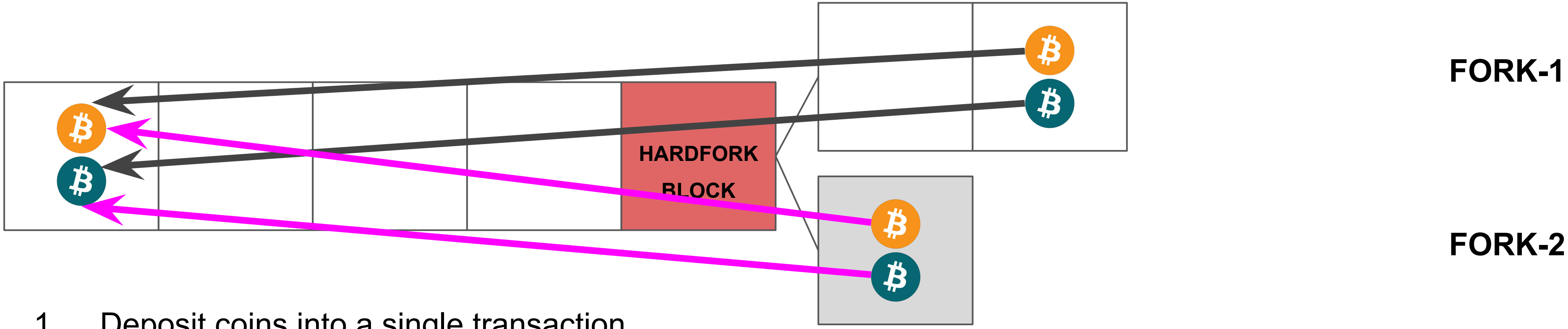
● Alices Deposit

● Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES
3. Alice withdraws both coins in FORK-2
4. Bob withdraws both coins in FORK-1

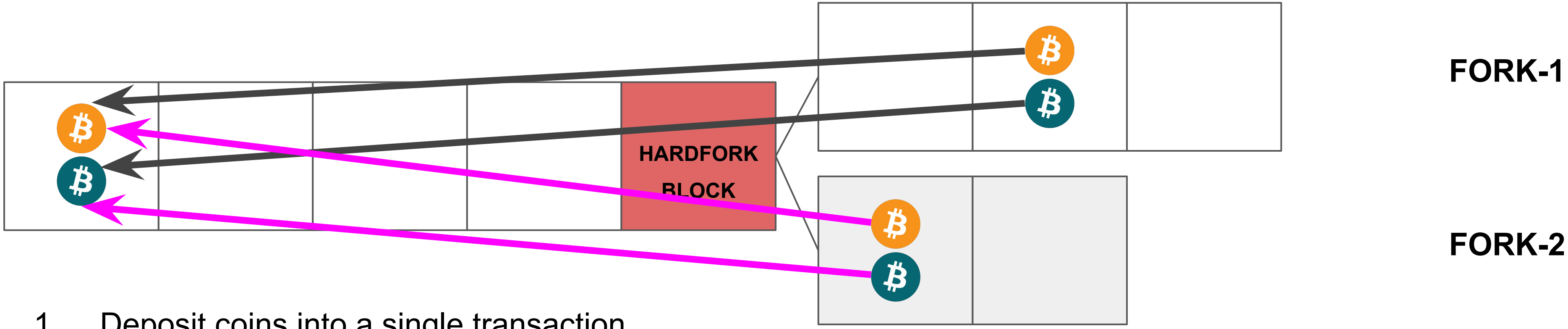
₿ Alices Deposit

₿ Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES
3. Alice withdraws both coins in FORK-2
4. Bob withdraws both coins in FORK-1

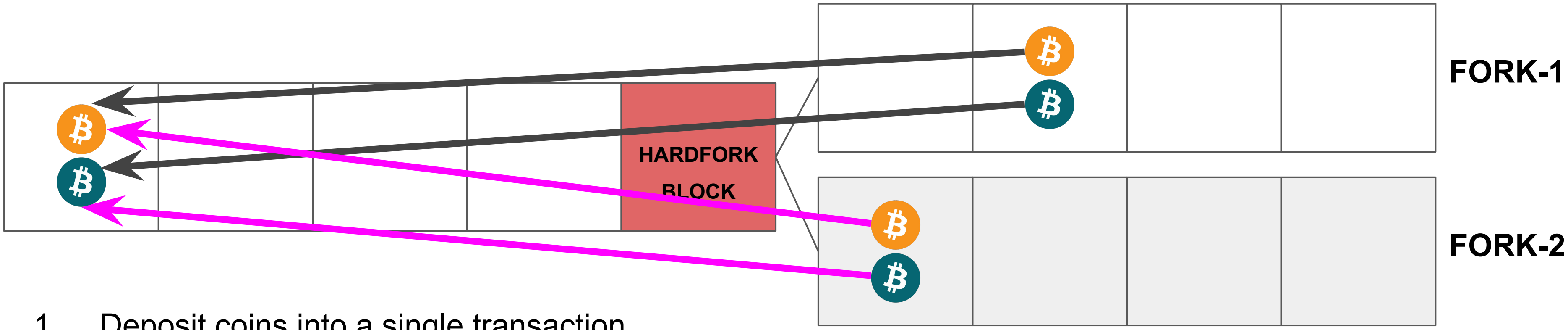
₿ Alices Deposit

₿ Bobs Deposit

Atomically Trade across two forks

Alice

Bob



1. Deposit coins into a single transaction.
2. HARDFORK ACTIVATES
3. Alice withdraws both coins in FORK-2
4. Bob withdraws both coins in FORK-1

● Alices Deposit

● Bobs Deposit

With and Without a Transaction Malleability fix

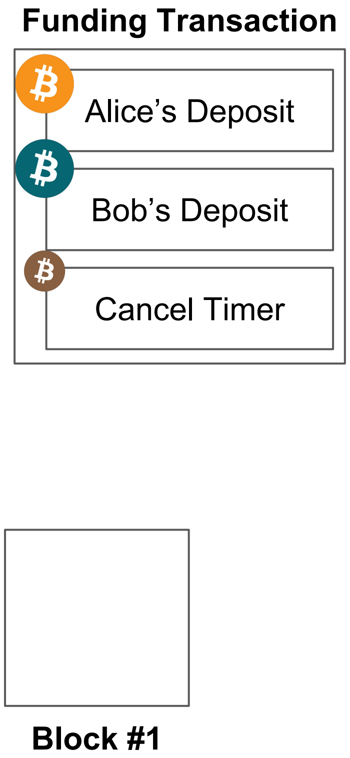
- Transaction malleability
  - The identification hash of a transaction (i.e. transaction id) can is malleable (i.e. changable) any time before it is accepted into the blockchain.
  - It is not safe to sign a chain of unconfirmed transactions.
- Without Transaction Malleability fix
  - Deposit must be stored in the blockchain - before both parties can sign atomic trade
- With Transaction Malleability Fix
  - All atomic trade transactions can be signed before the deposit is stored in the blockchain

… **Small difference? Huge implications for bitcoin contract design.**

Atomically Trade across two forks **without a fix for transaction malleability?**

- **Funding Stage**
  - Both parties deposit coins into the blockchain
- **Setup Cancellation:**
  - Bob will be able to cancel the atomic trade before Δ_{cancel}
- **Setup Atomic Trade:**
  - Both Alice and Bob exchange Transfer transactions.
  - Alice must reveal a secret R of H(R) after Δ_{fork} to trigger the trade
- **Setup Alice's Forfeit:**
  - Alice sets up a forfeit - if she does not reveal R before then Δ_B Bob can claim all the coins.
- **Commit to Trade**
  - Alice broadcasts a transaction after Δ_{cancel} that commits both parties to the atomic trade.
- **Atomic Trade**
  - Alice reveals R after Δ_{fork} and claims her coins in FORK-2
  - Bob finds R and claims his coins in FORK-1

Funding Stage



Funding Transaction

Alice’s Deposit

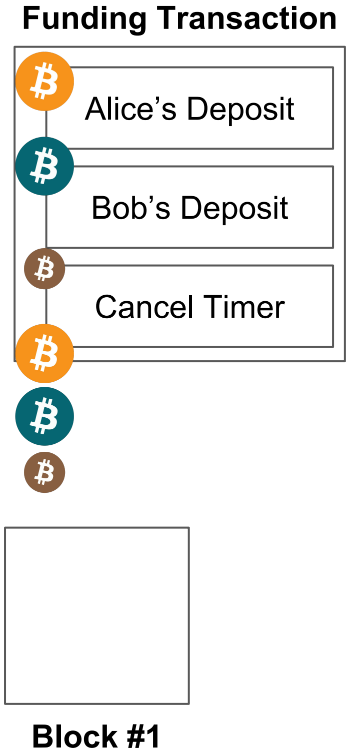
Bob’s Deposit

Cancel Timer

Block #1

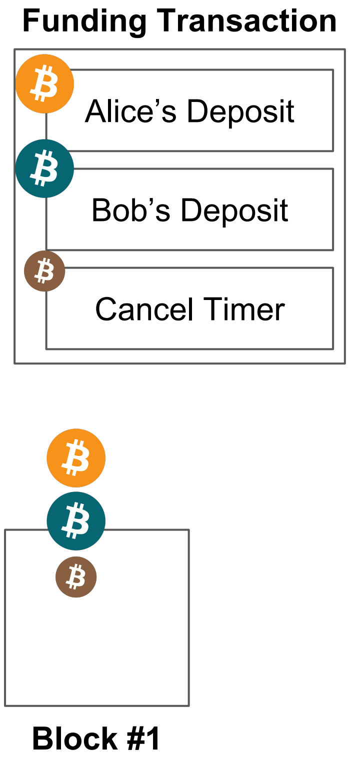
1. **Funding Transaction:** Stores deposit of both parties.

Funding Stage



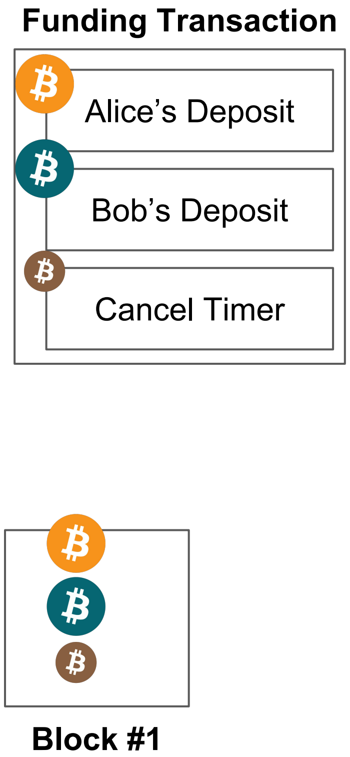
1. **Funding Transaction:** Stores deposit of both parties.

Funding Stage



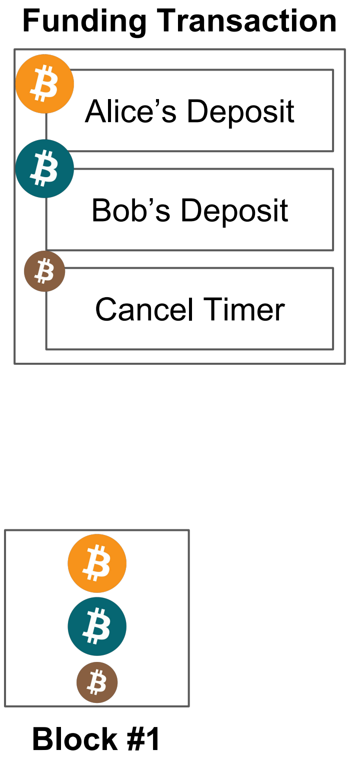
1. **Funding Transaction:** Stores deposit of both parties.

Funding Stage



1. **Funding Transaction:** Stores deposit of both parties.

Funding Stage



Funding Transaction

Alice’s Deposit

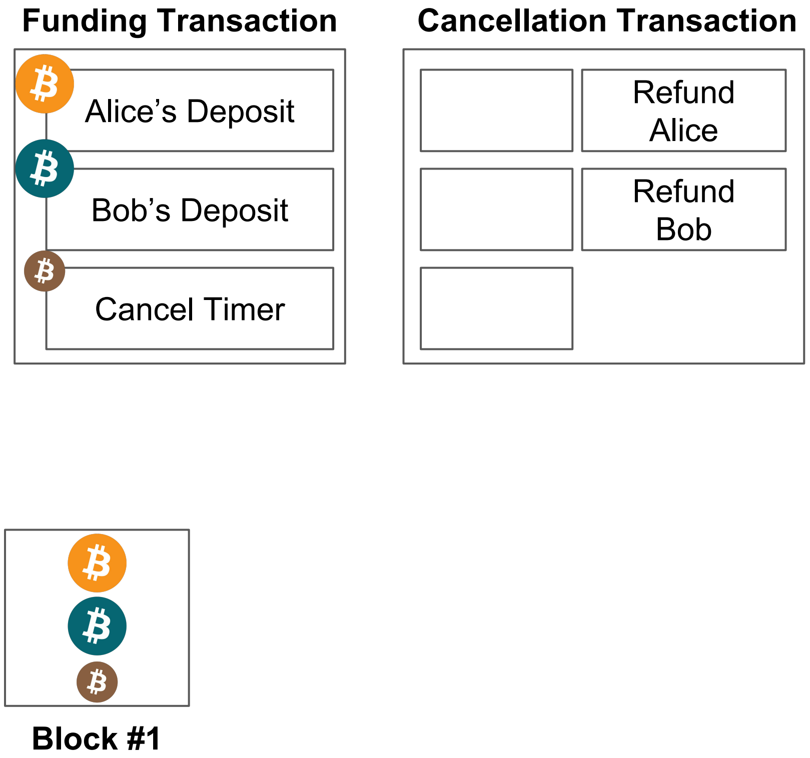
Bob’s Deposit

Cancel Timer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.

Setup Cancellation



Funding Transaction

Alice’s Deposit

Bob’s Deposit

Cancel Timer

Cancellation Transaction

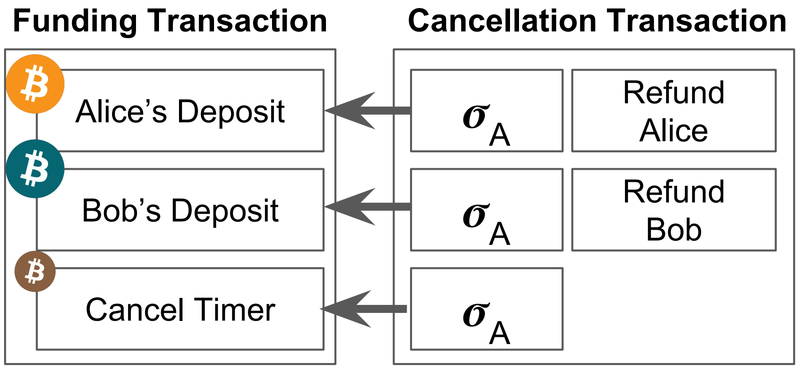
Refund Alice

Refund Bob

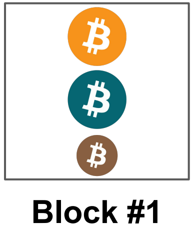
Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**

Setup Cancellation



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   - Signed by Alice and sent to Bob

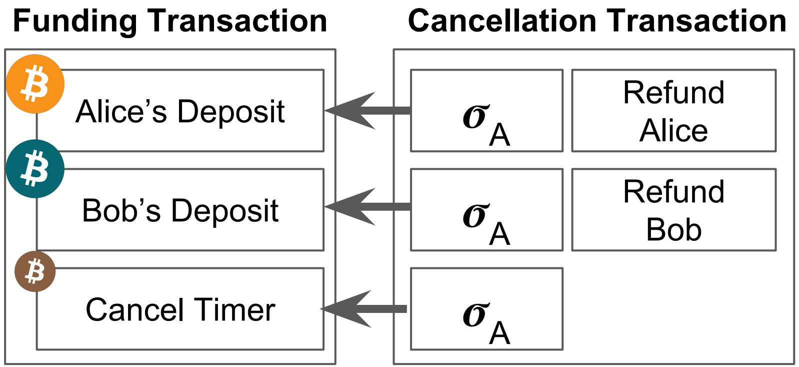


Why do we NEED a cancellation transaction?!

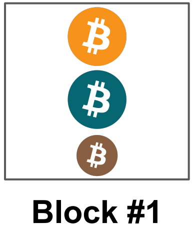
Later on, Alice will commit to reveal **R** of **H(R).**
If R is not revealed - she'll forfeit all coins to Bob.

If Alice refuses to make this commitment…
This transaction **lets Bob cancel the atomic trade altogether.**

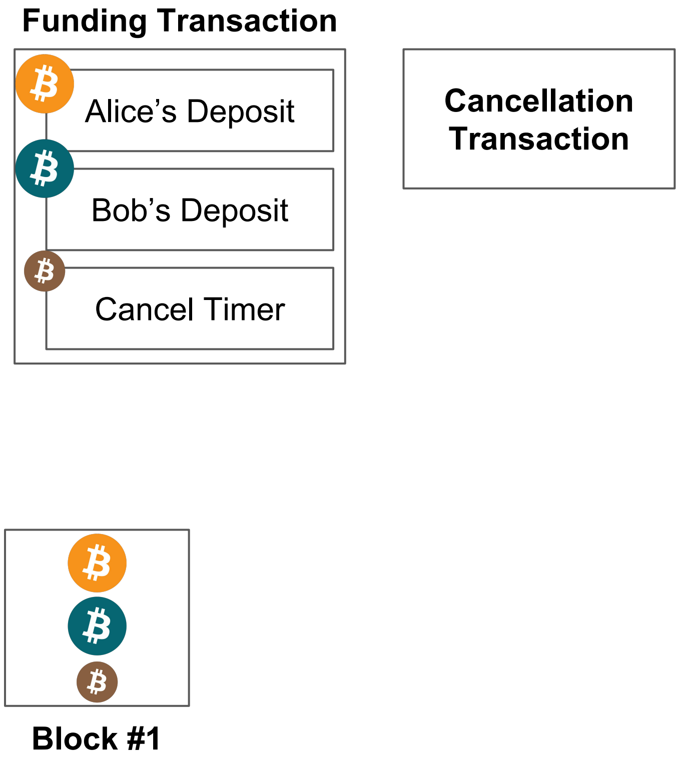
Setup Cancellation



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   - ○ Signed by Alice and sent to Bob

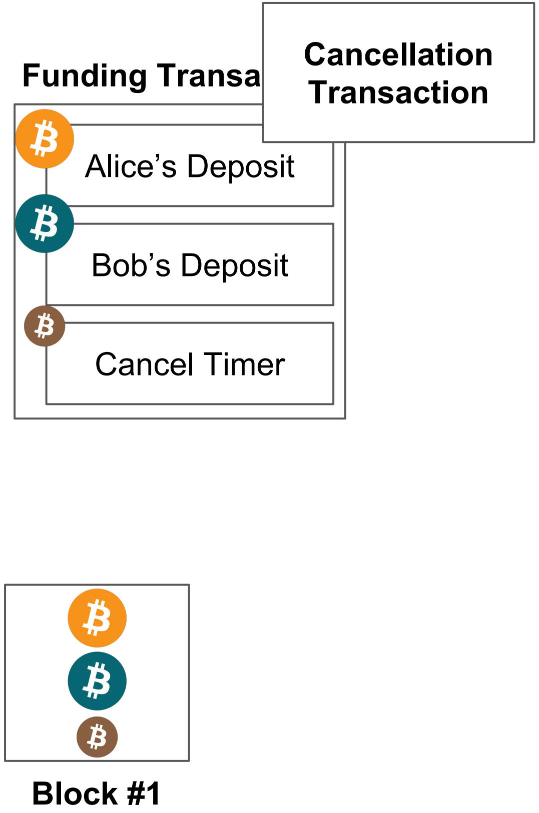


Setup Cancellation



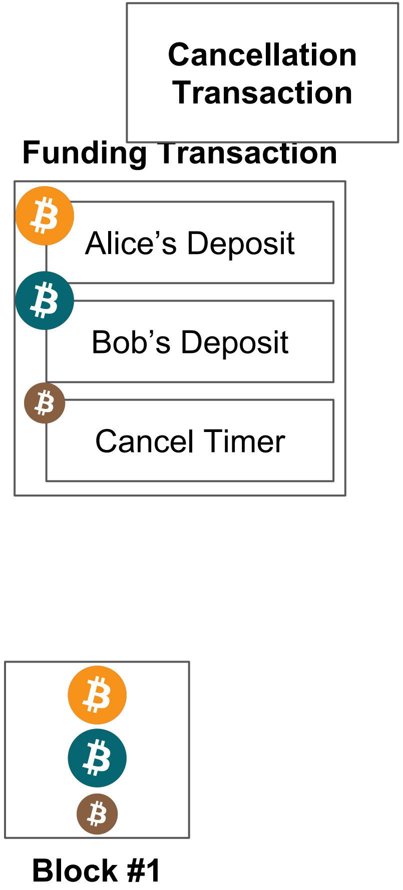
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   - Signed by Alice and sent to Bob

Setup Cancellation



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta}_{\textbf{cancel}}$ **= Block 3**
   - Signed by Alice and sent to Bob

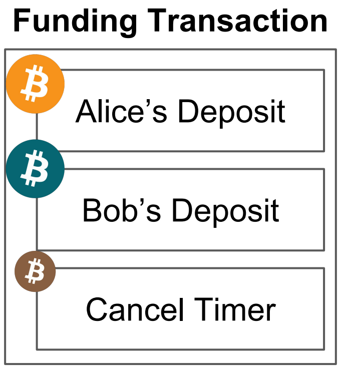
Setup Cancellation

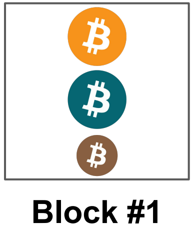


1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta_{\text{cancel}}}$ **= Block 3**
   - Signed by Alice and sent to Bob

Cancellation Transaction

Setup Cancellation

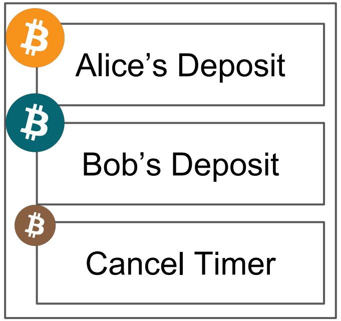




1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   - Signed by Alice and sent to Bob

Setup Atomic Trade

Cancellation Transaction

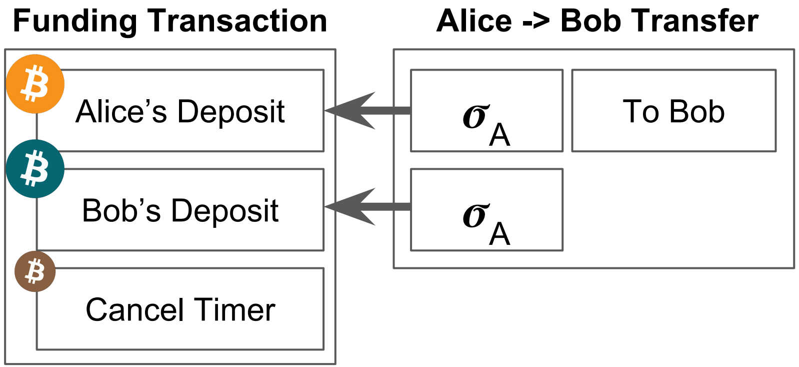

Funding Transaction

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.

Cancellation Transaction

Setup Atomic Trade

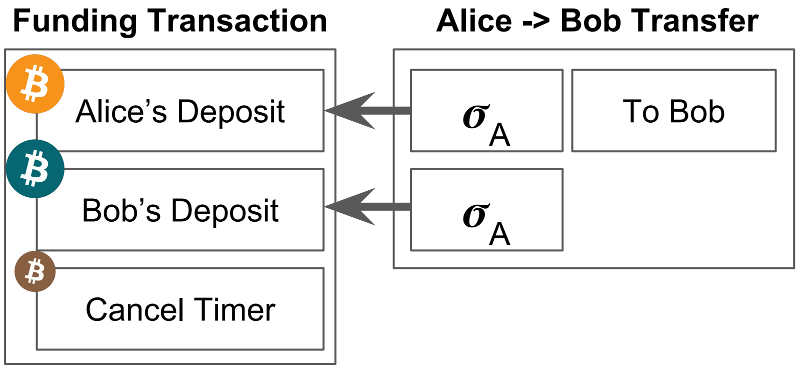


1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

Block #1

Cancellation Transaction

Setup Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

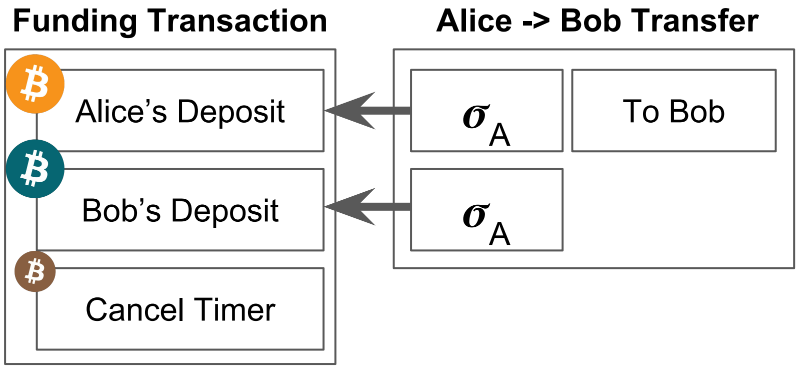
Block #1

Condition in Alice -> Bob Transfer:

Alice: "You can claim these coins Bob, if I reveal the **secret R of H(R)".**

Cancellation Transaction

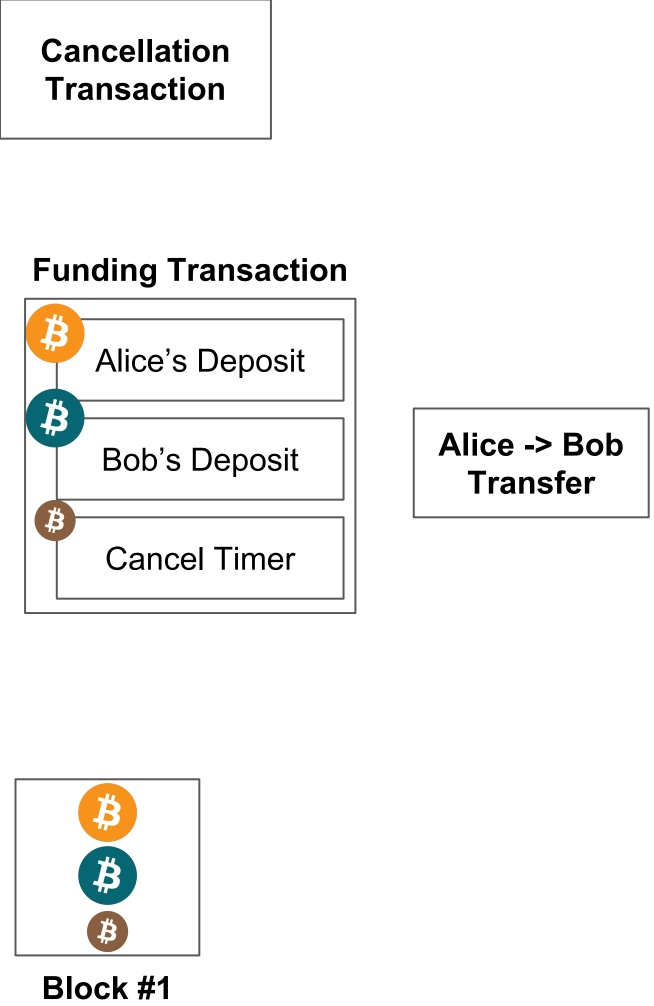
Setup Atomic Trade



Block #1

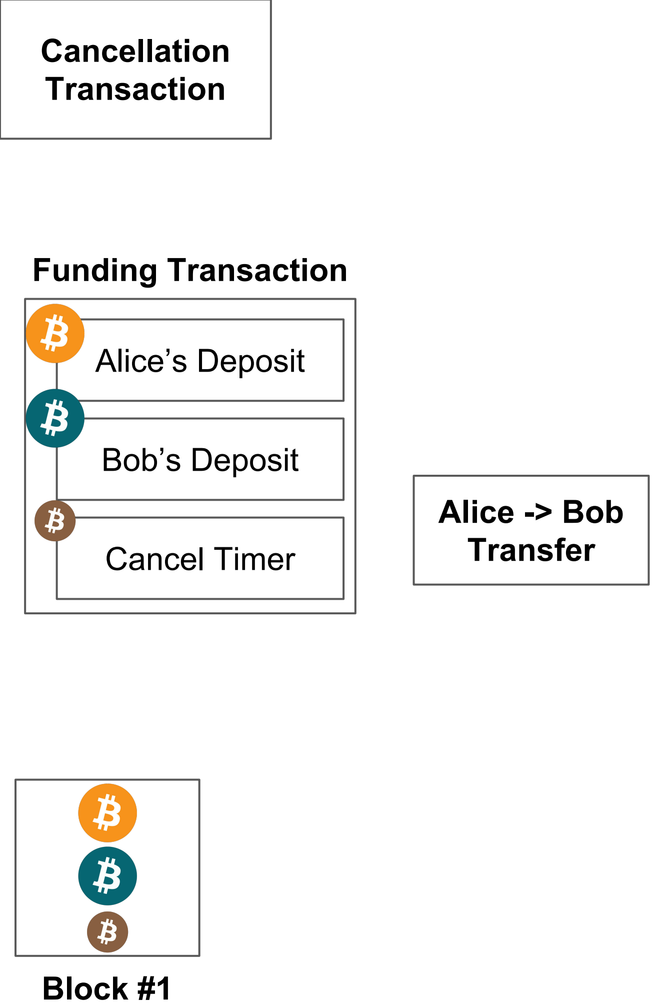
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

Setup Atomic Trade



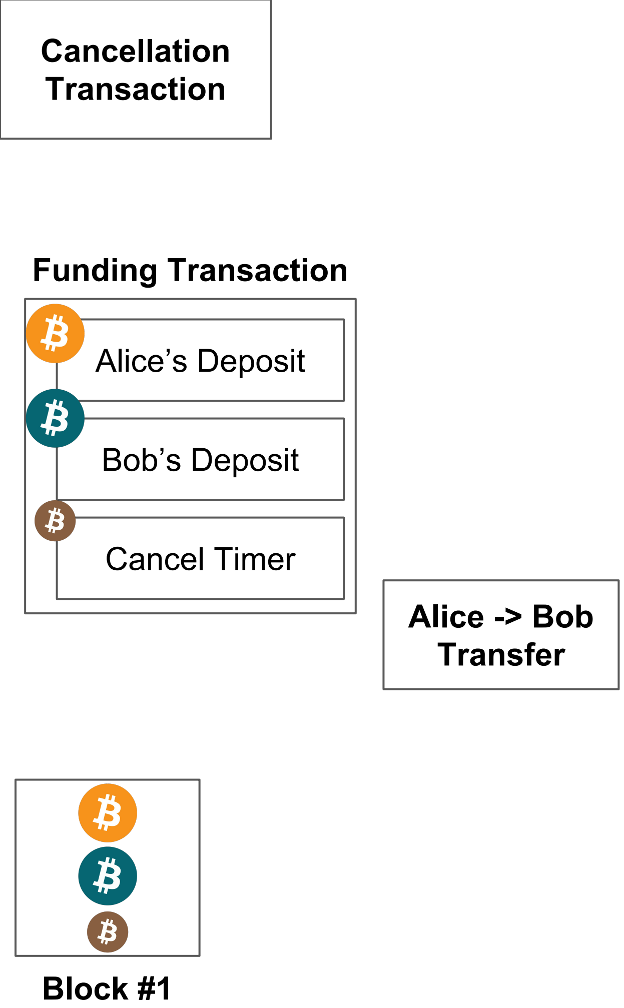
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
    a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
    a. Alice signs A->B and sends to Bob.

Setup Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
    a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
    a. Alice signs A->B and sends to Bob.

Setup Atomic Trade



Cancellation Transaction

Funding Transaction

Alice’s Deposit

Bob’s Deposit

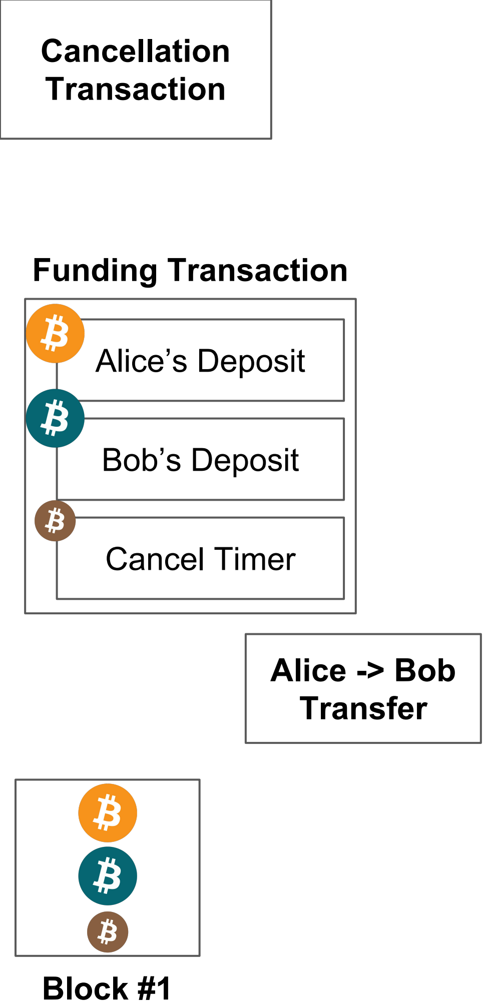
Cancel Timer

Alice -> Bob Transfer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta}_{\textbf{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

Setup Atomic Trade



Cancellation Transaction

Funding Transaction

Alice’s Deposit

Bob’s Deposit

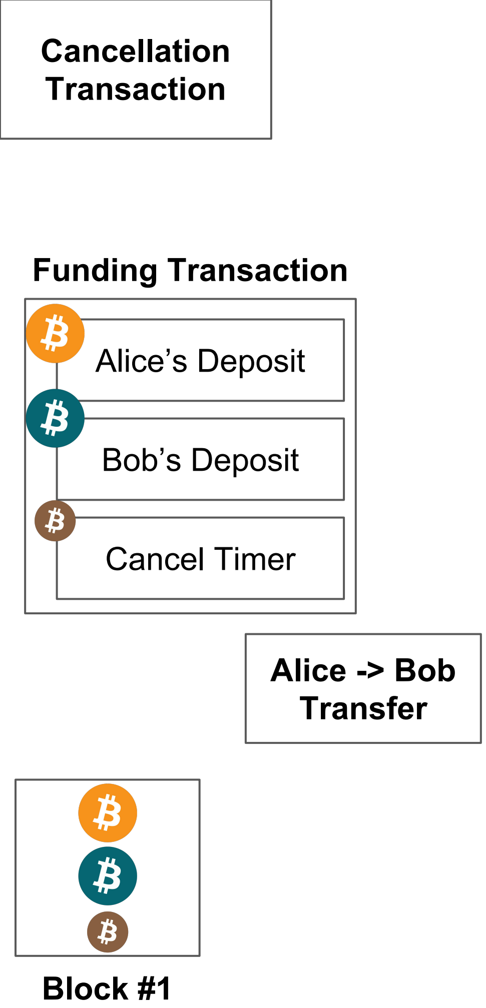
Cancel Timer

Alice -> Bob Transfer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

Setup Atomic Trade



Cancellation Transaction

Funding Transaction

Alice’s Deposit

Bob’s Deposit

Cancel Timer

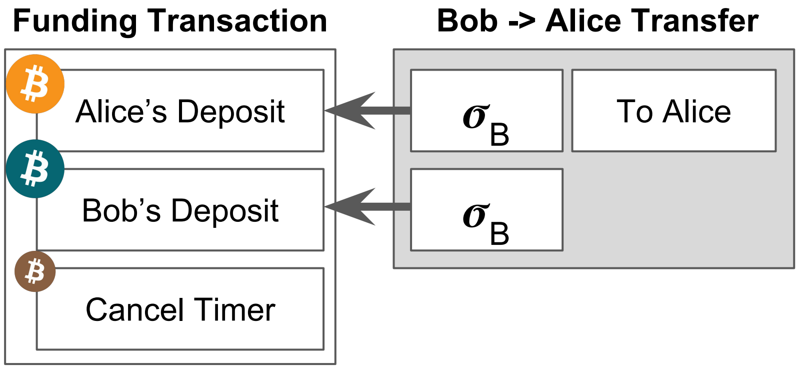
Alice -> Bob Transfer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta}_{\textbf{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.

Cancellation Transaction

Setup Atomic Trade



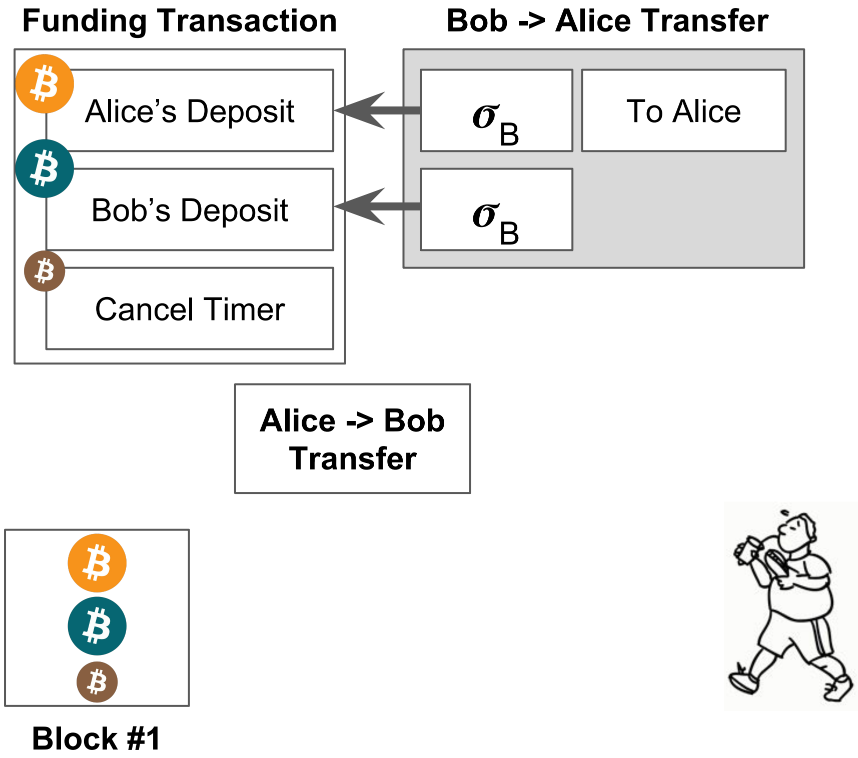
Alice -> Bob Transfer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

Cancellation Transaction

Setup Atomic Trade



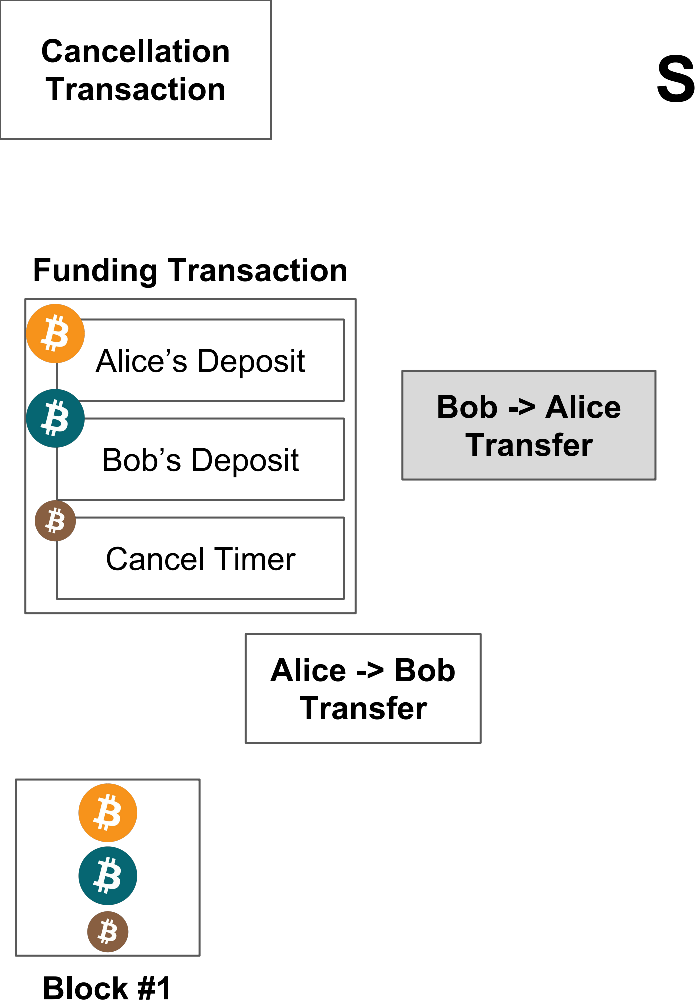
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta_{\text{cancel}}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

Condition in Bob -> Alice Transfer:

Bob: "You can claim these coins Alice, if I reveal the **secret R of H(R)".**

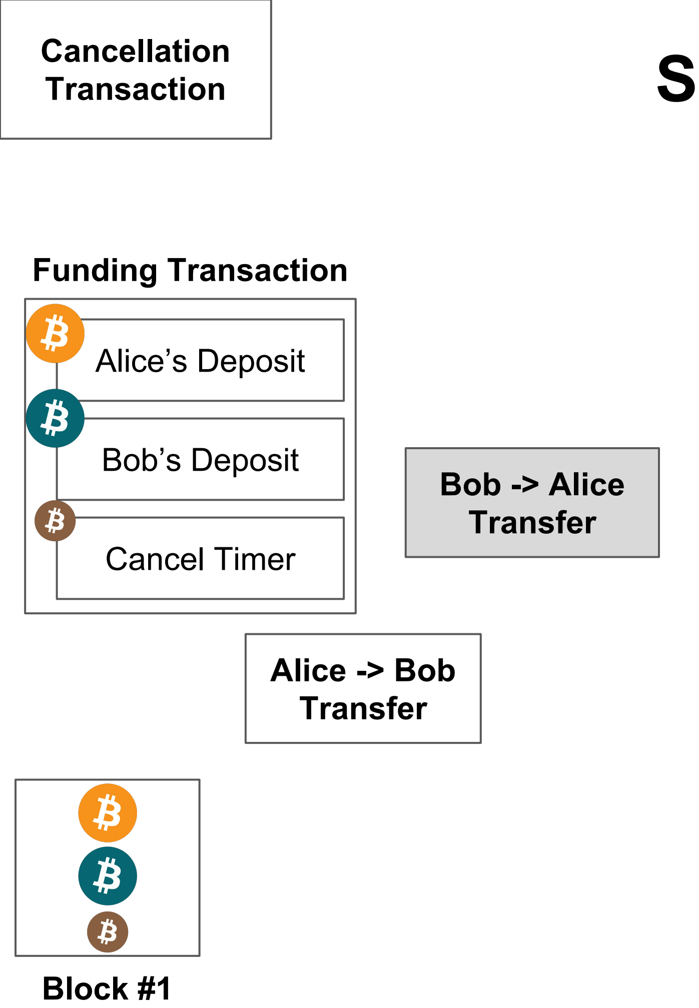
****REPLAY PROTECTION REQUIRED****

Setup Atomic Trade



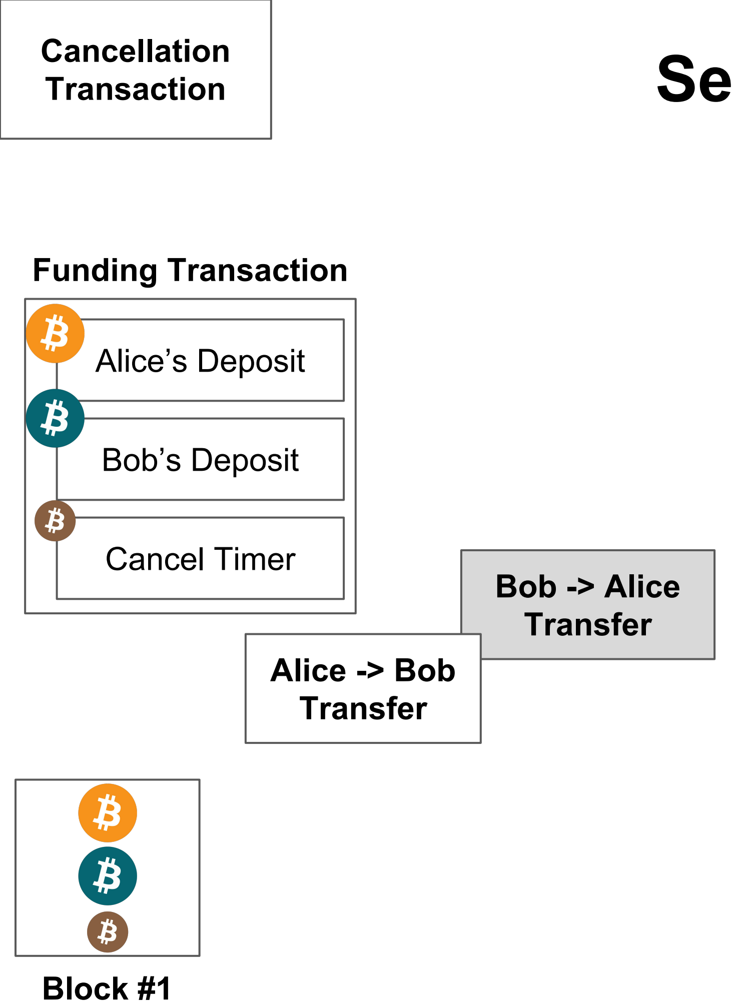
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

Setup Atomic Trade



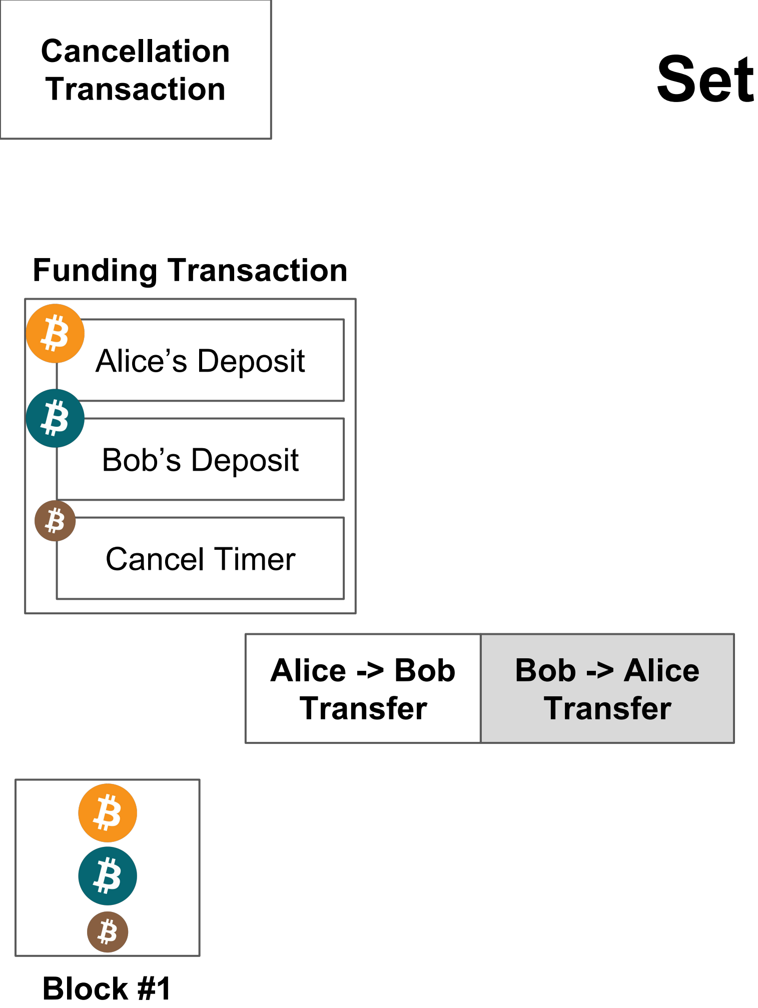
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

Setup Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\mathbf{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

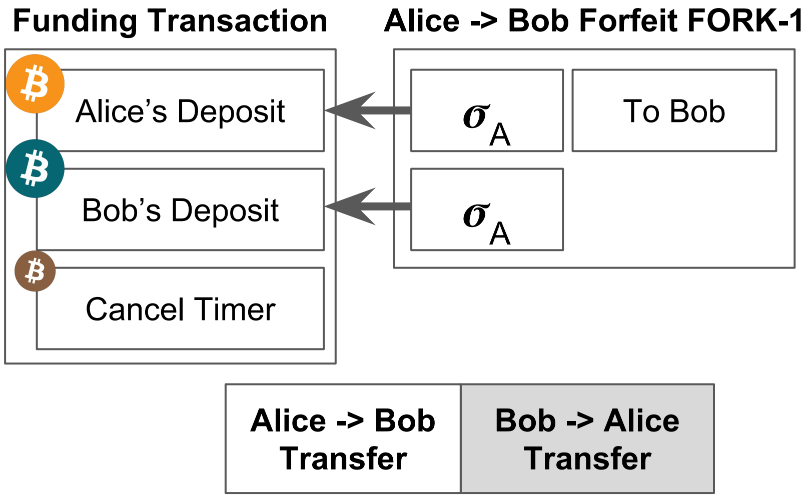
Setup Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.

Cancellation Transaction

Setup Alice's Forfeit



Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

Alice -> Bob Forfeit FORK-1

σ_A

To Bob

σ_A

Alice -> Bob Transfer

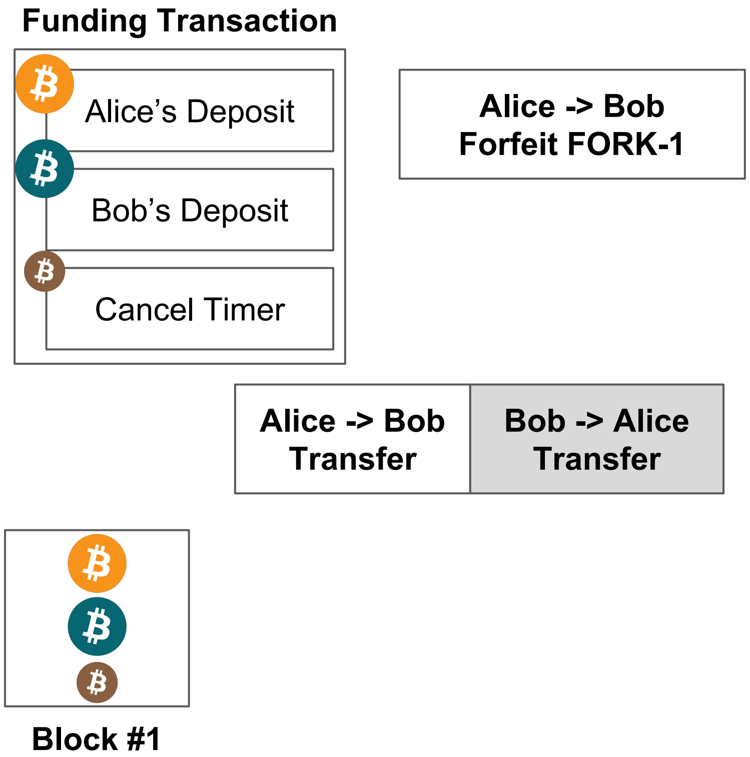
Bob -> Alice Transfer

Block #1

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

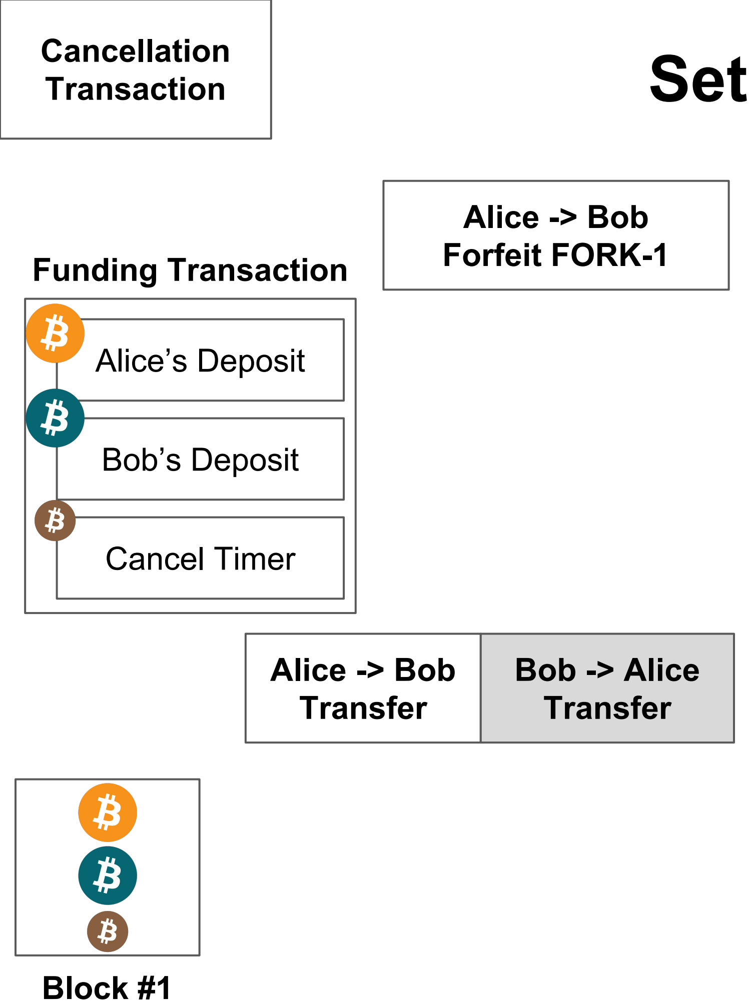
Cancellation Transaction

Setup Alice’s Forfeit



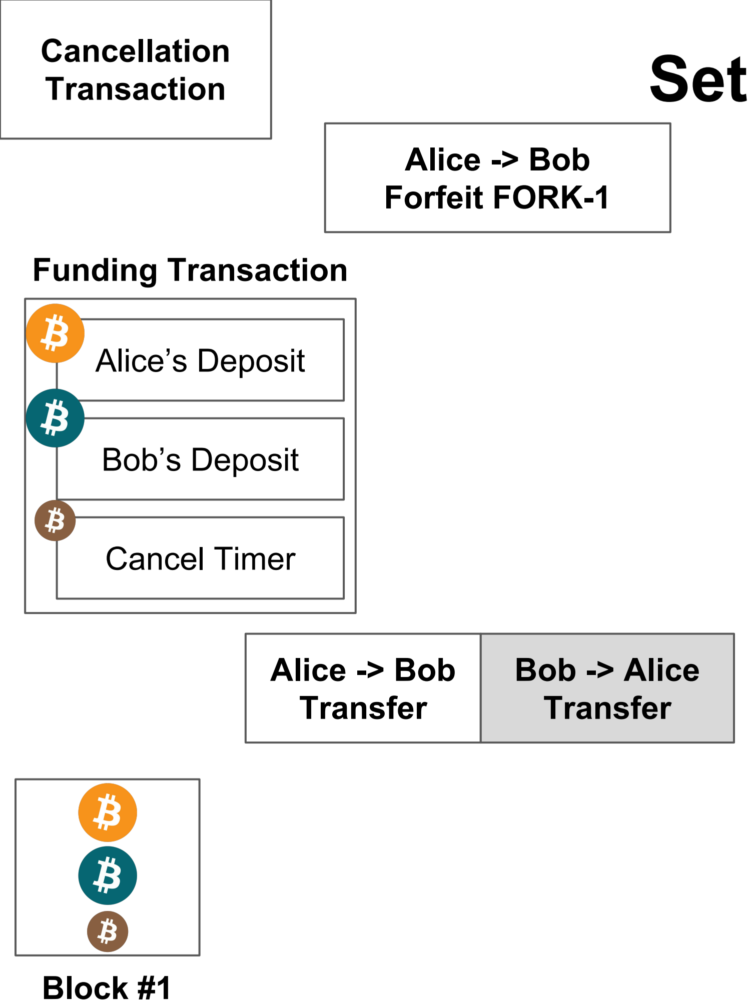
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Setup Alice’s Forfeit



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\mathbf{\Delta_{cancel}}$ **= Block 3**
    a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
    a. Alice signs A->B and sends to Bob.
    b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Setup Alice’s Forfeit

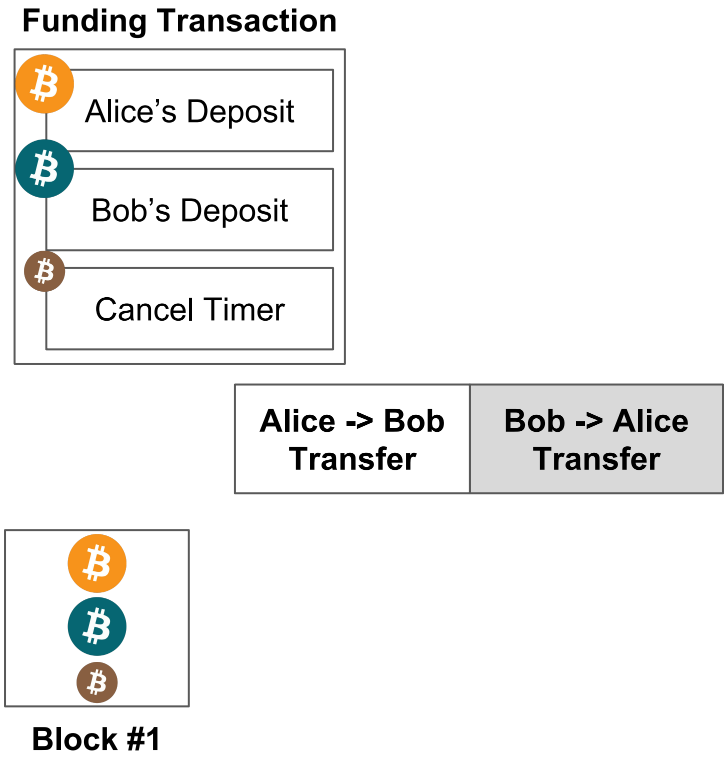


1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Cancellation Transaction

Alice -> Bob Forfeit FORK-1

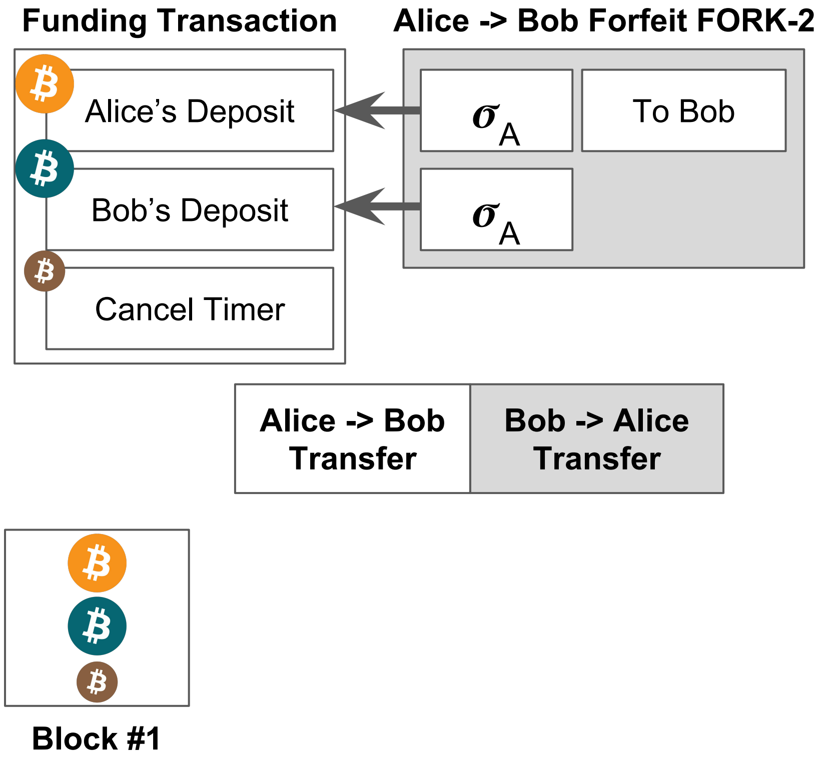
Setup Alice’s Forfeit



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\mathbf{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

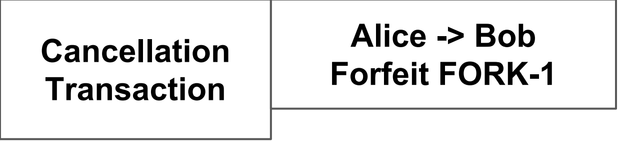
Cancellation Transaction
Alice -> Bob Forfeit FORK-1

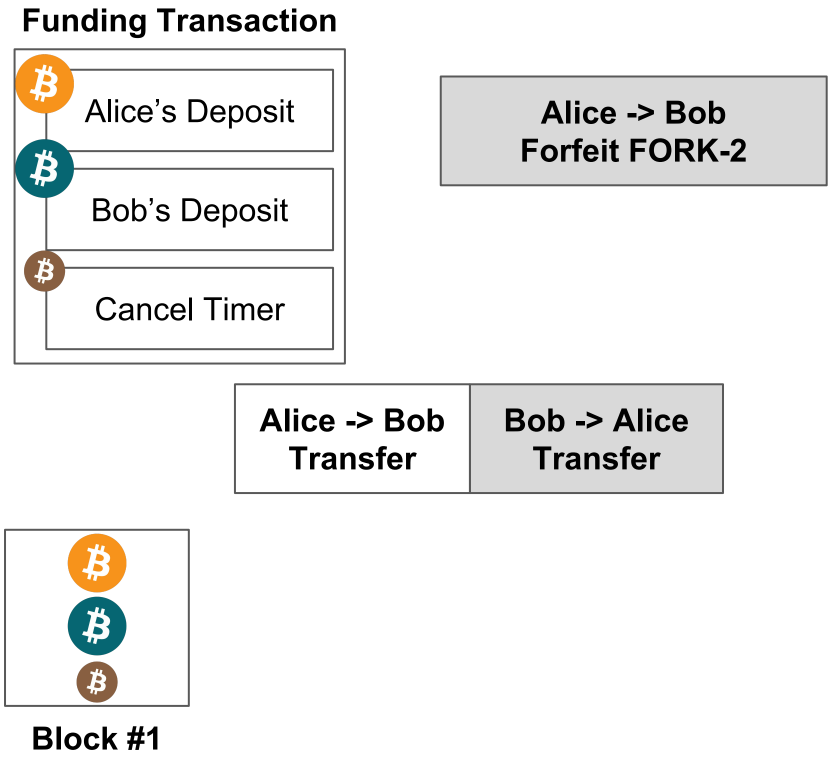
Setup Alice's Forfeit



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

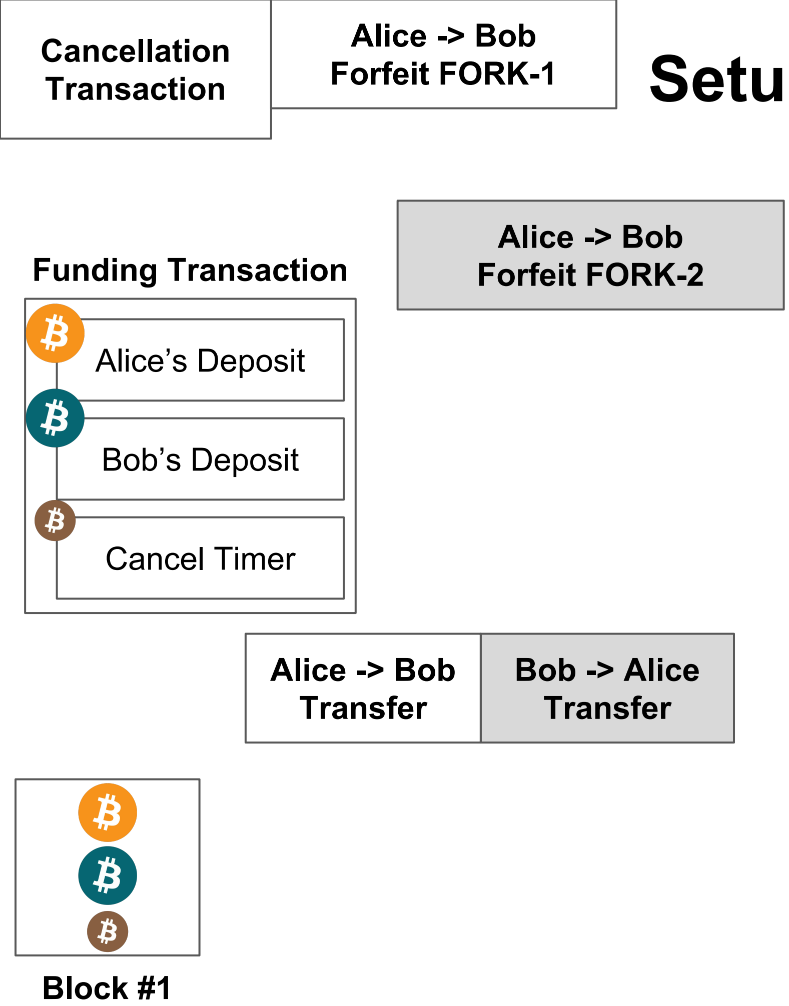
Setup Alice's Forfeit





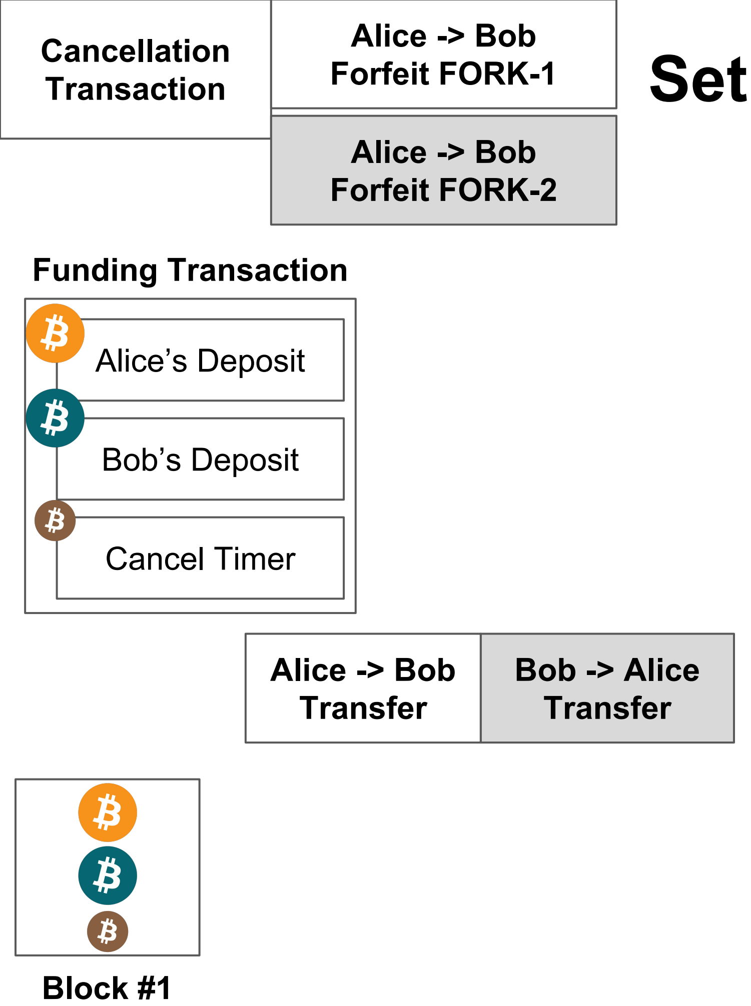
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Setup Alice’s Forfeit



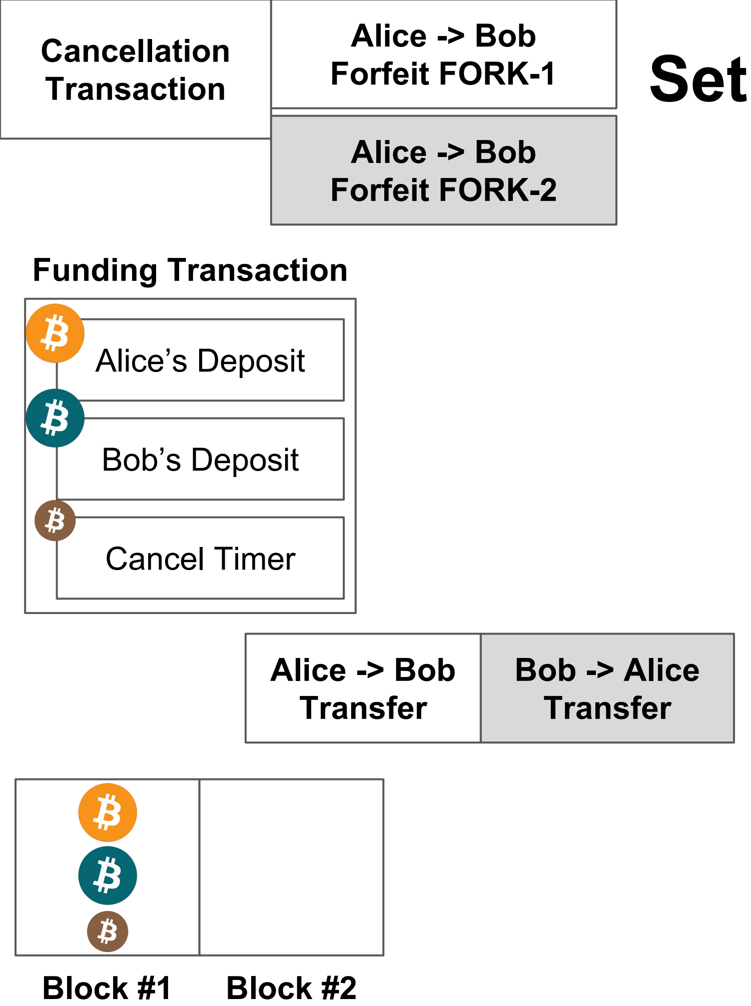
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Setup Alice’s Forfeit



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Setup Alice's Forfeit



Cancellation Transaction

Alice -> Bob Forfeit FORK-1

Alice -> Bob Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

Alice -> Bob Transfer

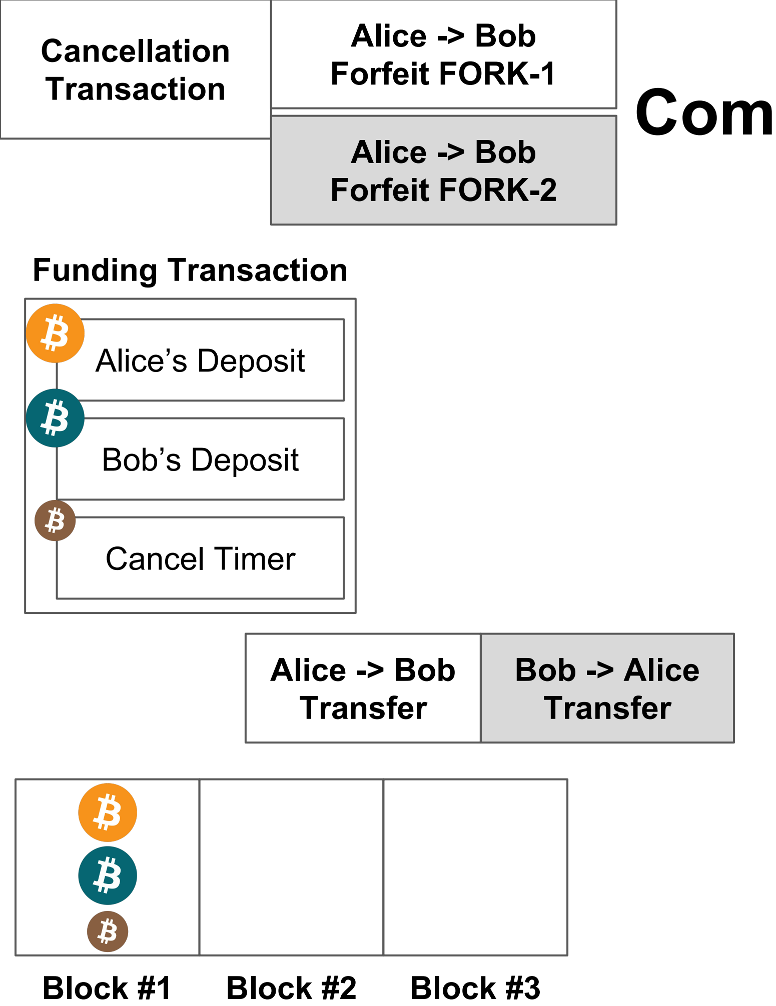
Bob -> Alice Transfer

Block #1

Block #2

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.

Both Parties
Commit To Atomic Trade



Cancellation Transaction

Alice -> Bob Forfeit FORK-1

Alice -> Bob Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

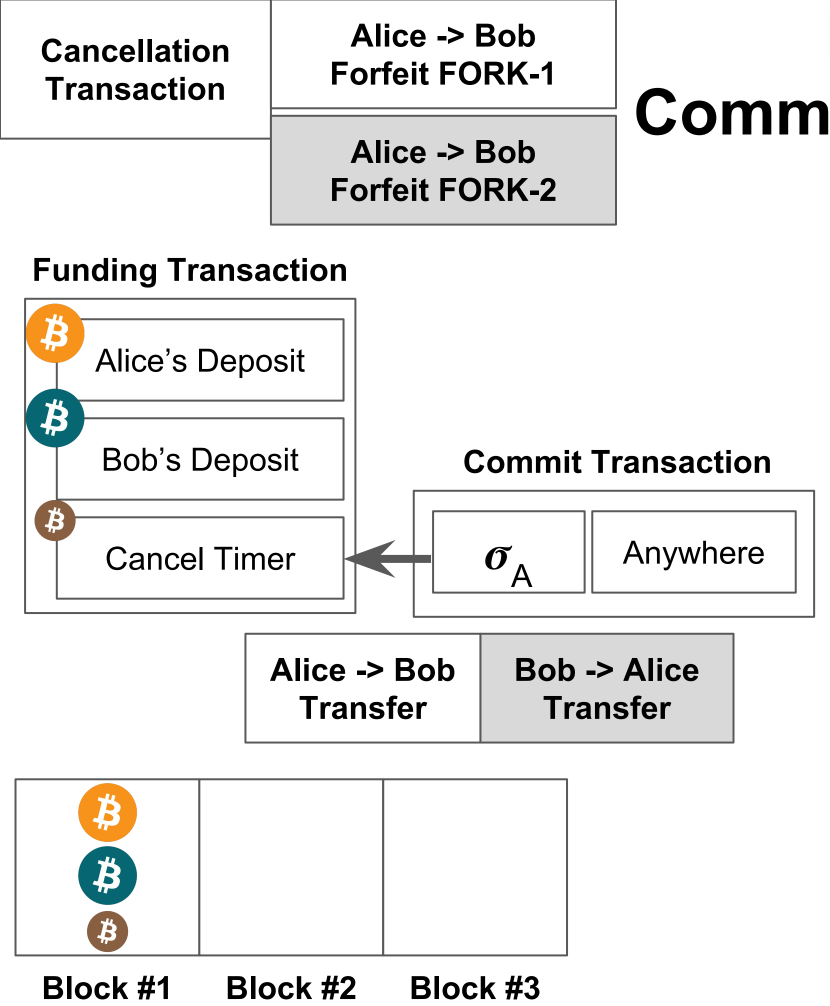
Alice -> Bob Transfer

Bob -> Alice Transfer

Block #1 **Block #2** **Block #3**

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after **Δ_{cancel} = Block 3**

Both Parties
Commit To Atomic Trade



Cancellation Transaction

Alice -> Bob Forfeit FORK-1

Alice -> Bob Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

Commit Transaction

σ_A

Anywhere

Alice -> Bob Transfer

Bob -> Alice Transfer

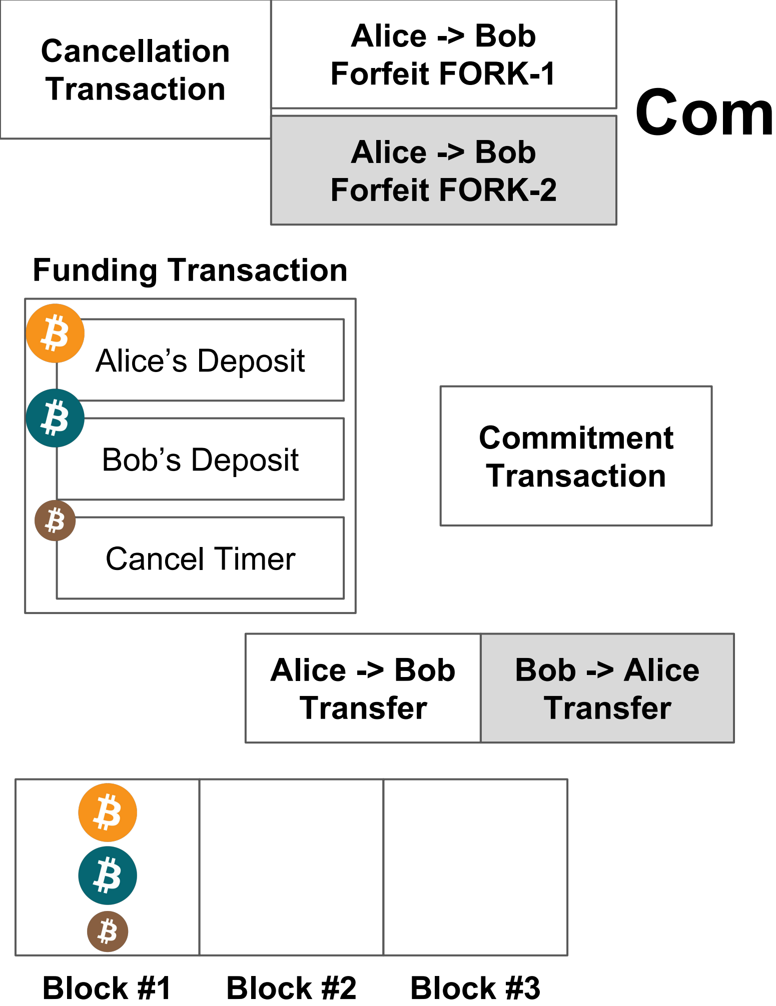
Block #1

Block #2

Block #3

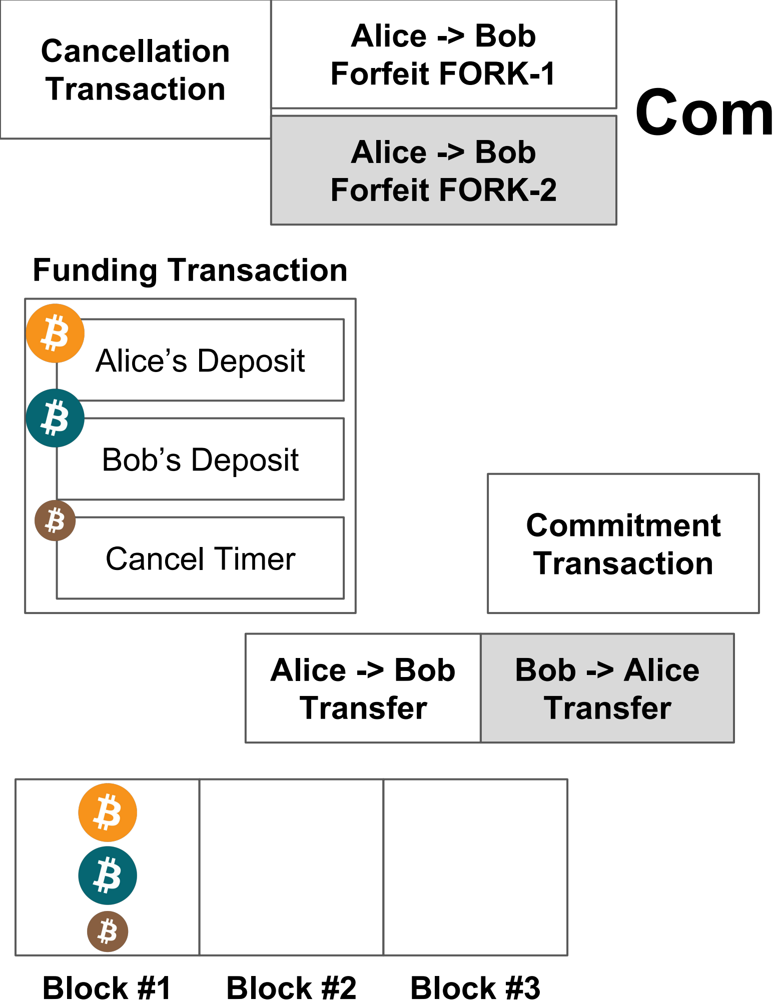
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\mathbf{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after $\mathbf{\Delta_{cancel}}$ **= Block 3**

Both Parties Commit To Atomic Trade



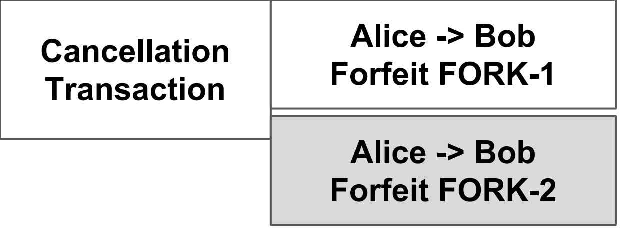
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after **Δ_{cancel} = Block 3**

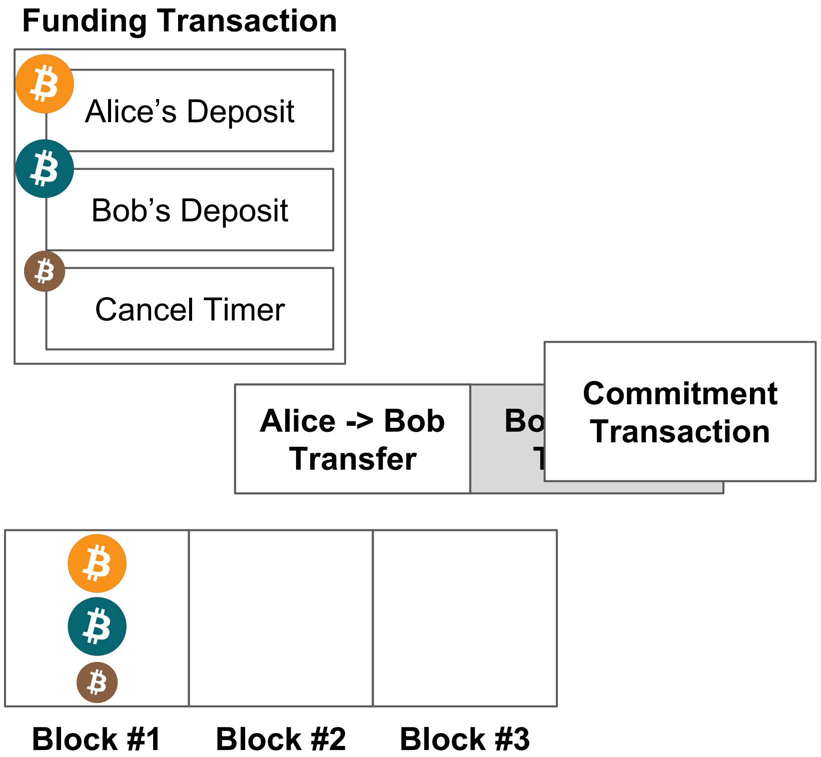
Both Parties Commit To Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\mathbf{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after $\mathbf{\Delta_{cancel}}$ **= Block 3**

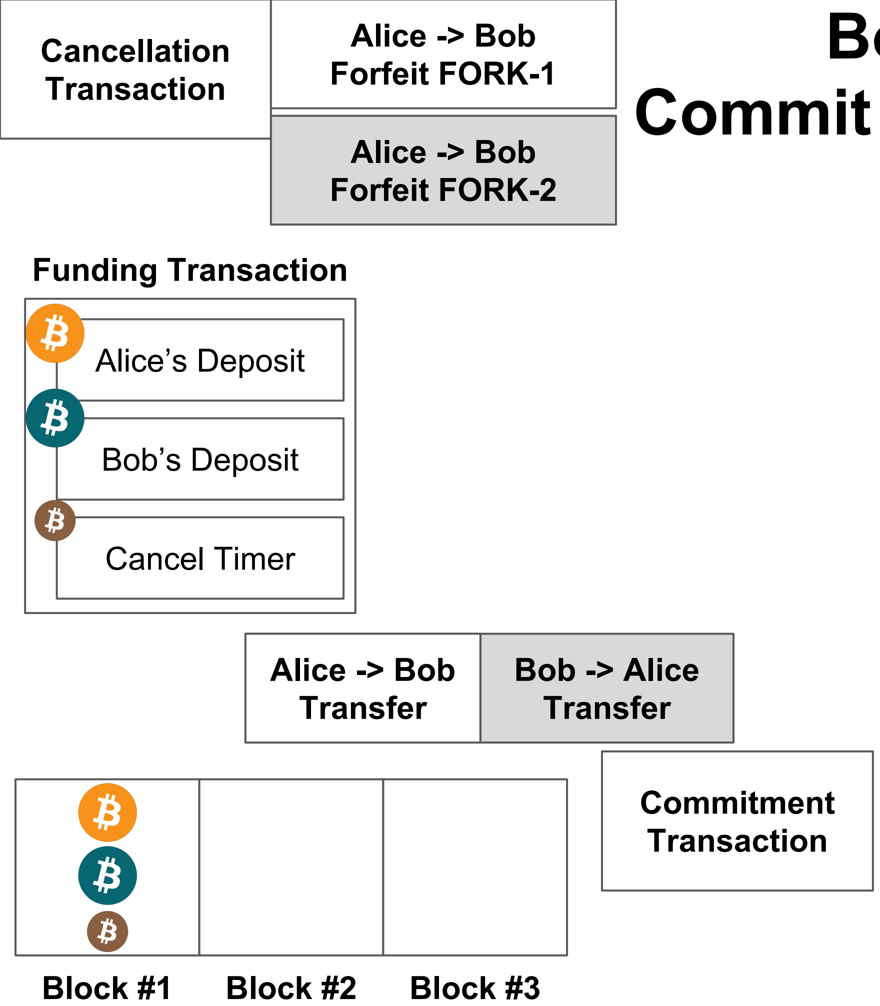
Both Parties Commit To Atomic Trade





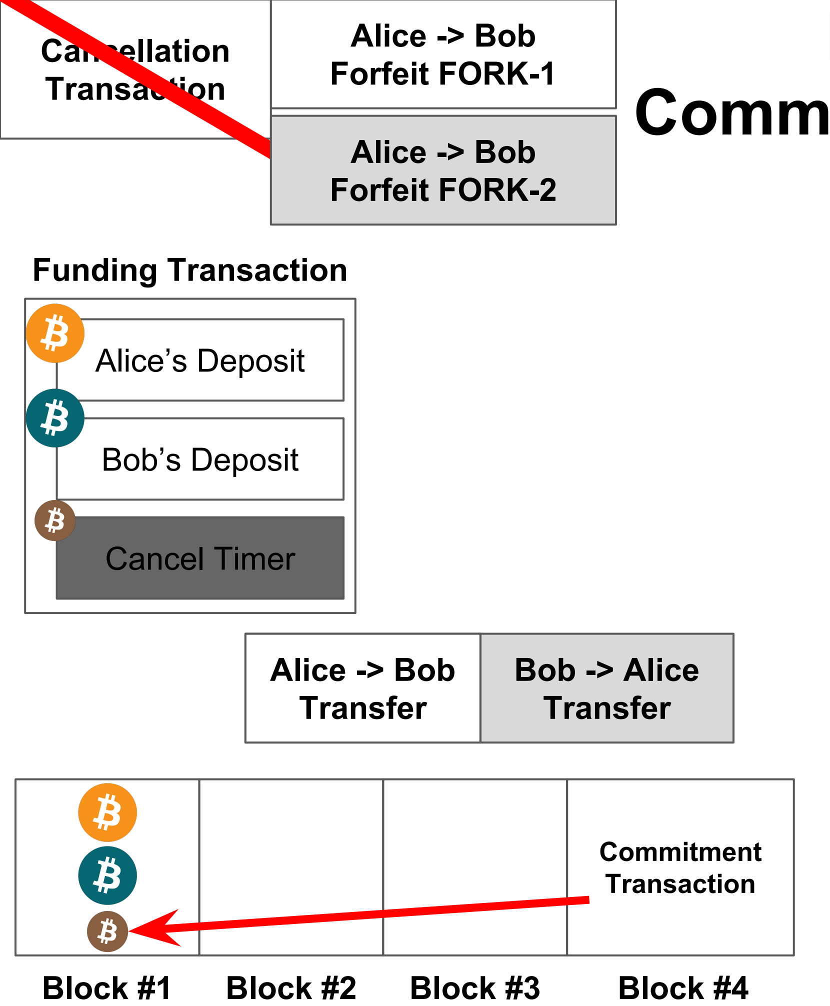
1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta_{cancel}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after $\boldsymbol{\Delta_{cancel}}$ **= Block 3**

Both Parties
Commit To Atomic Trade



1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before **Δ_{cancel} = Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after **Δ_{cancel} = Block 3**

Both Parties Commit To Atomic Trade



~~Cancellation Transaction~~

Alice -> Bob Forfeit FORK-1

Alice -> Bob Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

Alice -> Bob Transfer

Bob -> Alice Transfer

Commitment Transaction

Block #1

Block #2

Block #3

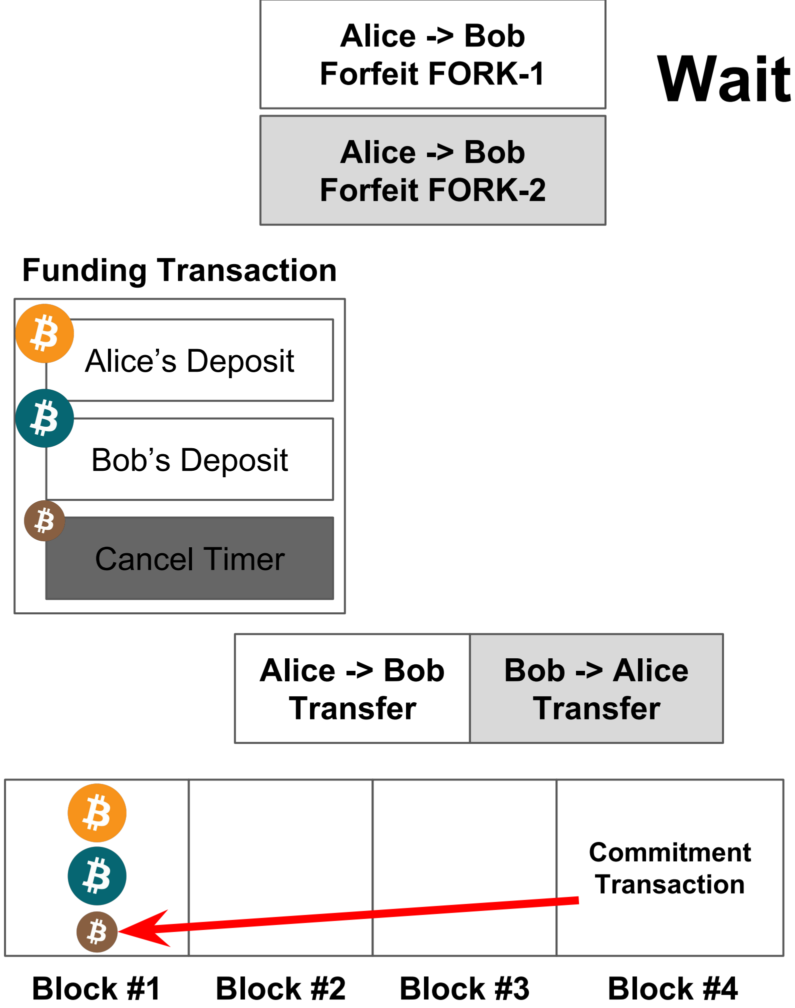
Block #4

1. **Funding Transaction:** Stores deposit of both parties.
2. **Cancellation Transaction**: Refunds all parties before $\boldsymbol{\Delta_{\text{cancel}}}$ **= Block 3**
   a. Signed by Alice and sent to Bob
3. **Transfer Transactions:** Sends each party coins in the respective fork if R of H(R) is revealed.
   a. Alice signs A->B and sends to Bob.
   b. Bob signs B->A and sends to Alice.
4. **Forfeit Transactions:** Alice promises to reveal pre-image r of H(R) before Δ_B = Block 7 otherwise Bob gets all coins.
5. **Commit Transaction:** Invalidates the cancellation transaction - and commits both parties to the trade! Only valid after $\boldsymbol{\Delta_{\text{cancel}}}$ **= Block 3**

Briefly what has happened so far…?

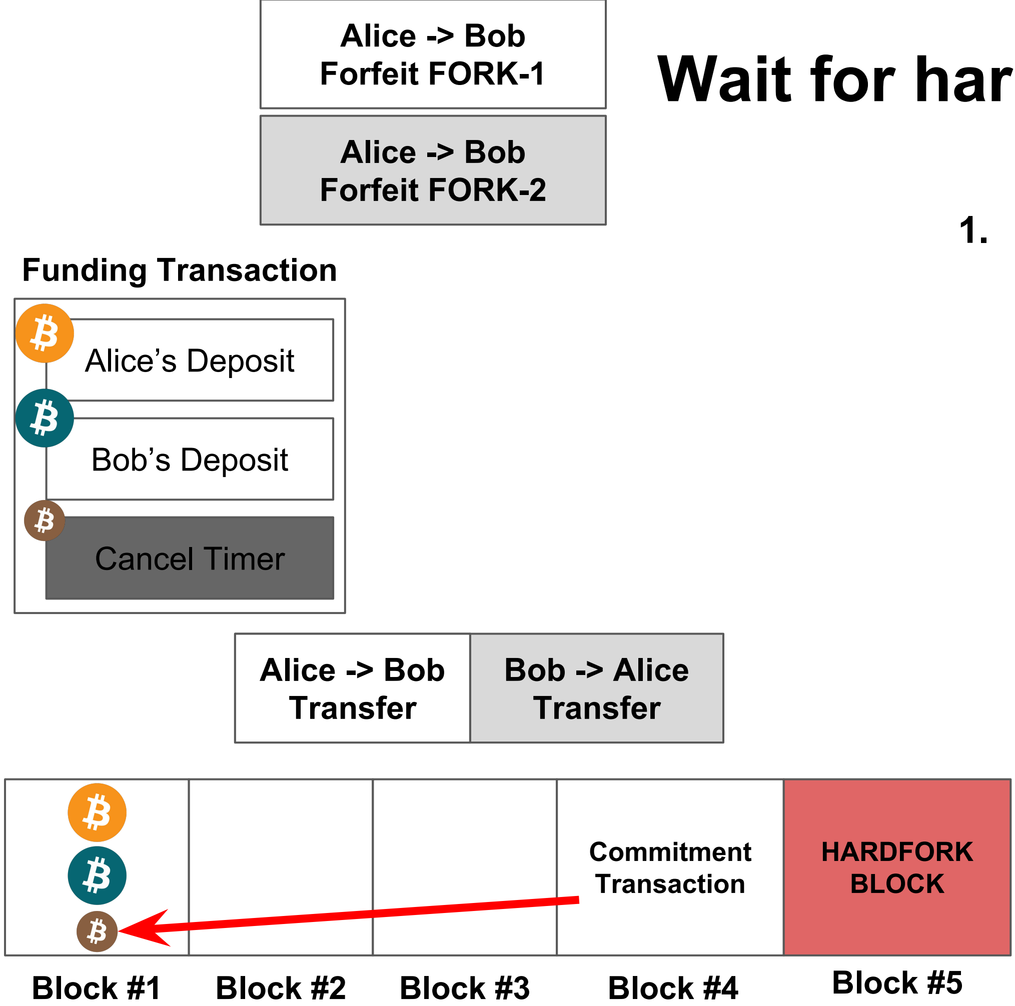
- **Funding Stage**
  - Both parties deposit coins into the blockchain
- **Setup Cancellation:**
  - Bob will be able to cancel the atomic trade before Δ_{cancel}
- **Setup Atomic Trade:**
  - Both Alice and Bob exchange Transfer transactions.
  - Alice must reveal a secret R of H(R) after Δ_{fork} to trigger the trade
- **Setup Alice's Forfeit:**
  - Alice sets up a forfeit - if she does not reveal R before then Δ_{B} Bob can claim all the coins.
- **Commit to Trade**
  - Alice broadcasts a transaction after Δ_{cancel} that commits both parties to the atomic trade.
- **Atomic Trade**
  - Alice reveals R after Δ_{fork} and claims her coins in FORK-2
  - Bob finds R and claims his coins in FORK-1

Wait for hardfork…



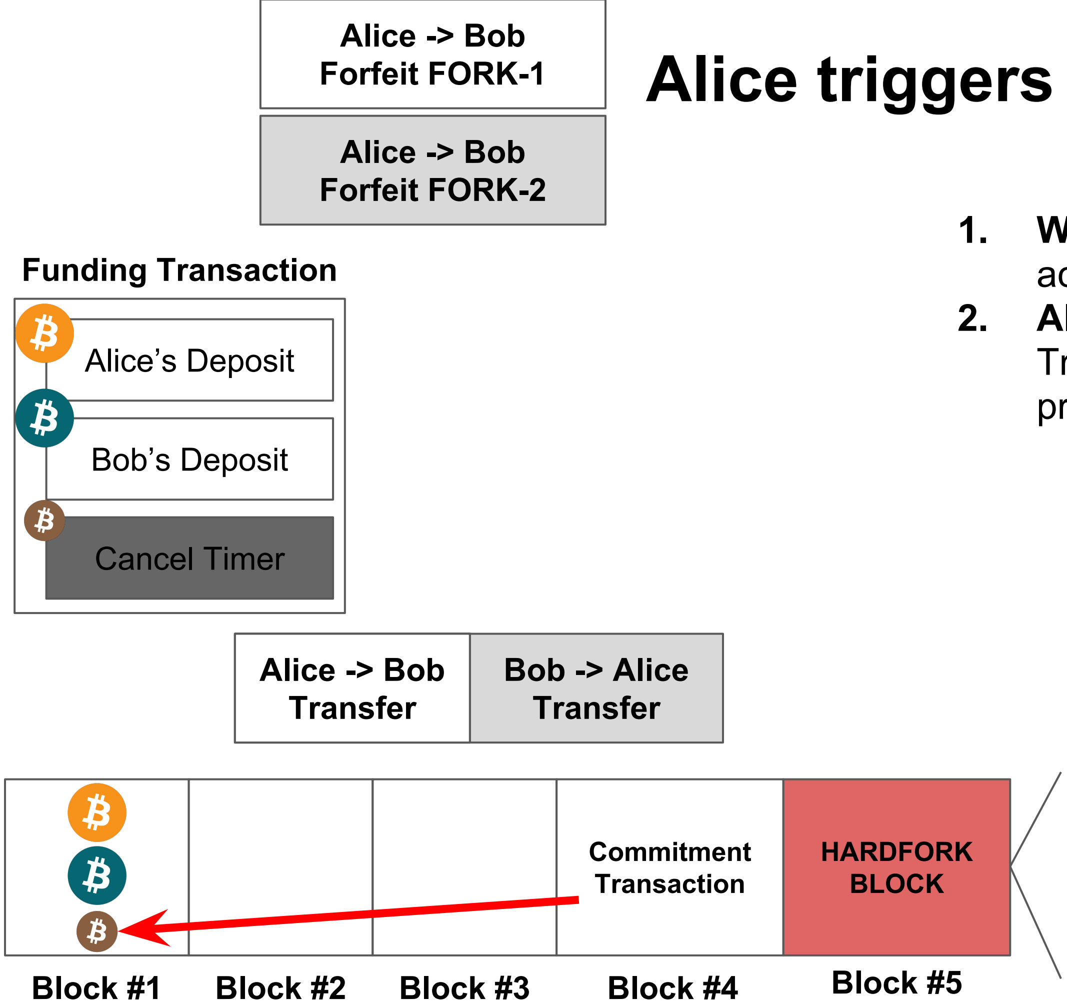
1. **Wait:** Both parties must wait until the hardfork activates.

Wait for hardfork…



1. **Wait:** Both parties must wait until the hardfork activates.

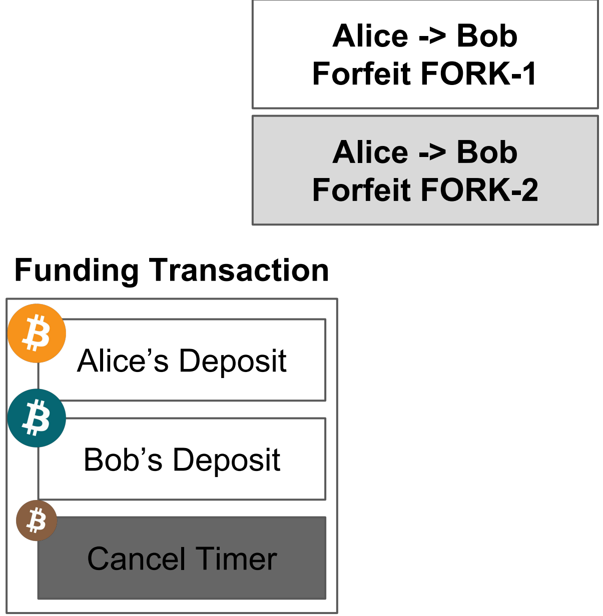
Alice triggers Trade



1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade.** She broadcasts Bob -> Alice Transfer Transaction which also reveals the pre-image R of H(R).

Alice triggers Trade

1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade.** She broadcasts Alice -> Bob transfer transaction and reveals pre-image R of H(R).



Alice -> Bob
Forfeit FORK-1

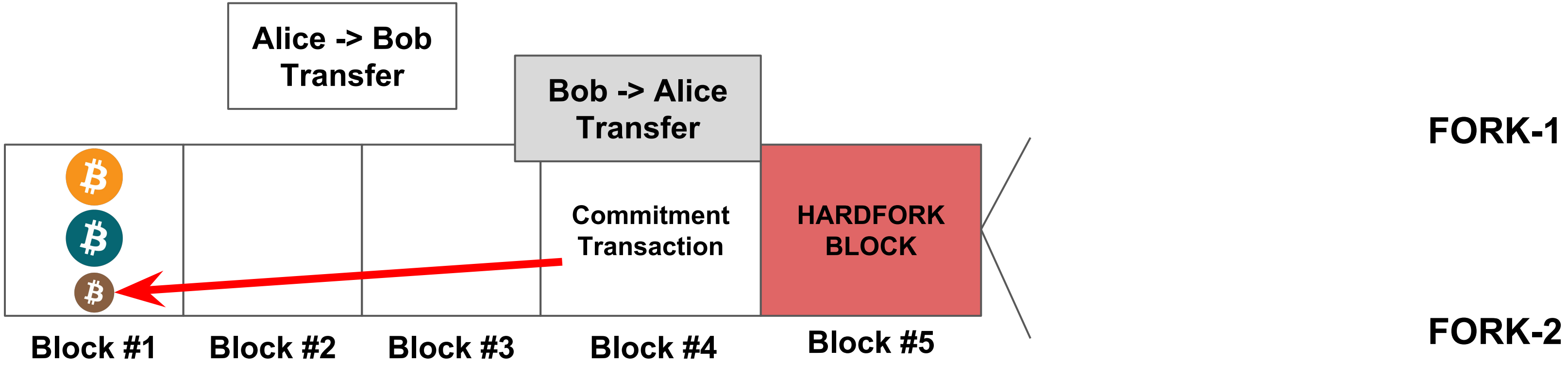
Alice -> Bob
Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer



Alice -> Bob
Transfer

Bob -> Alice
Transfer

Commitment
Transaction

HARDFORK
BLOCK

Block #1

Block #2

Block #3

Block #4

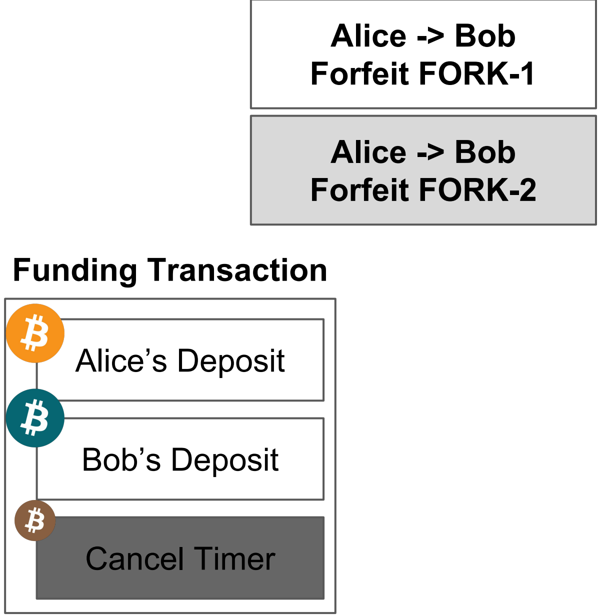
Block #5

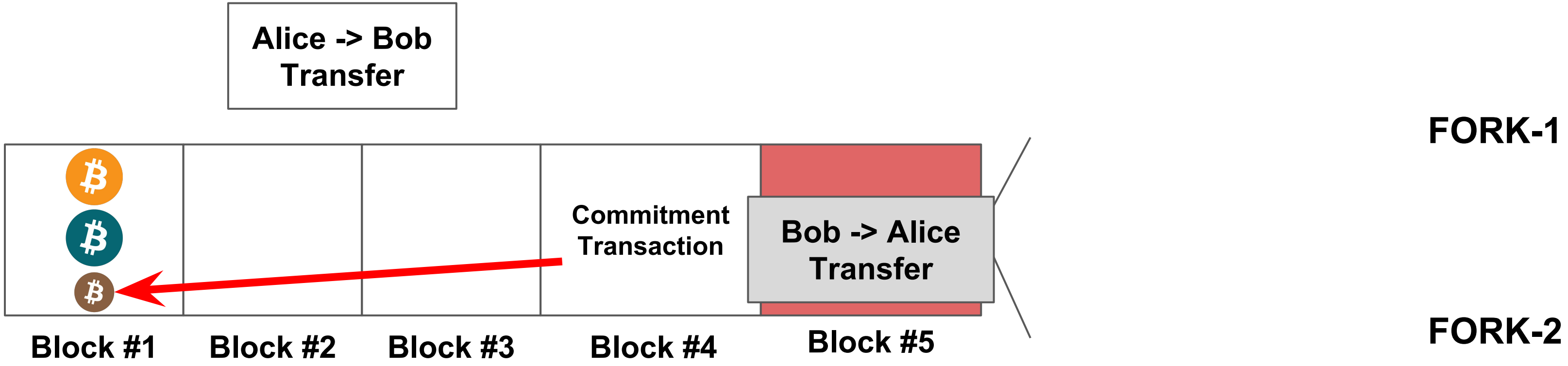
FORK-1

FORK-2

Alice triggers Trade

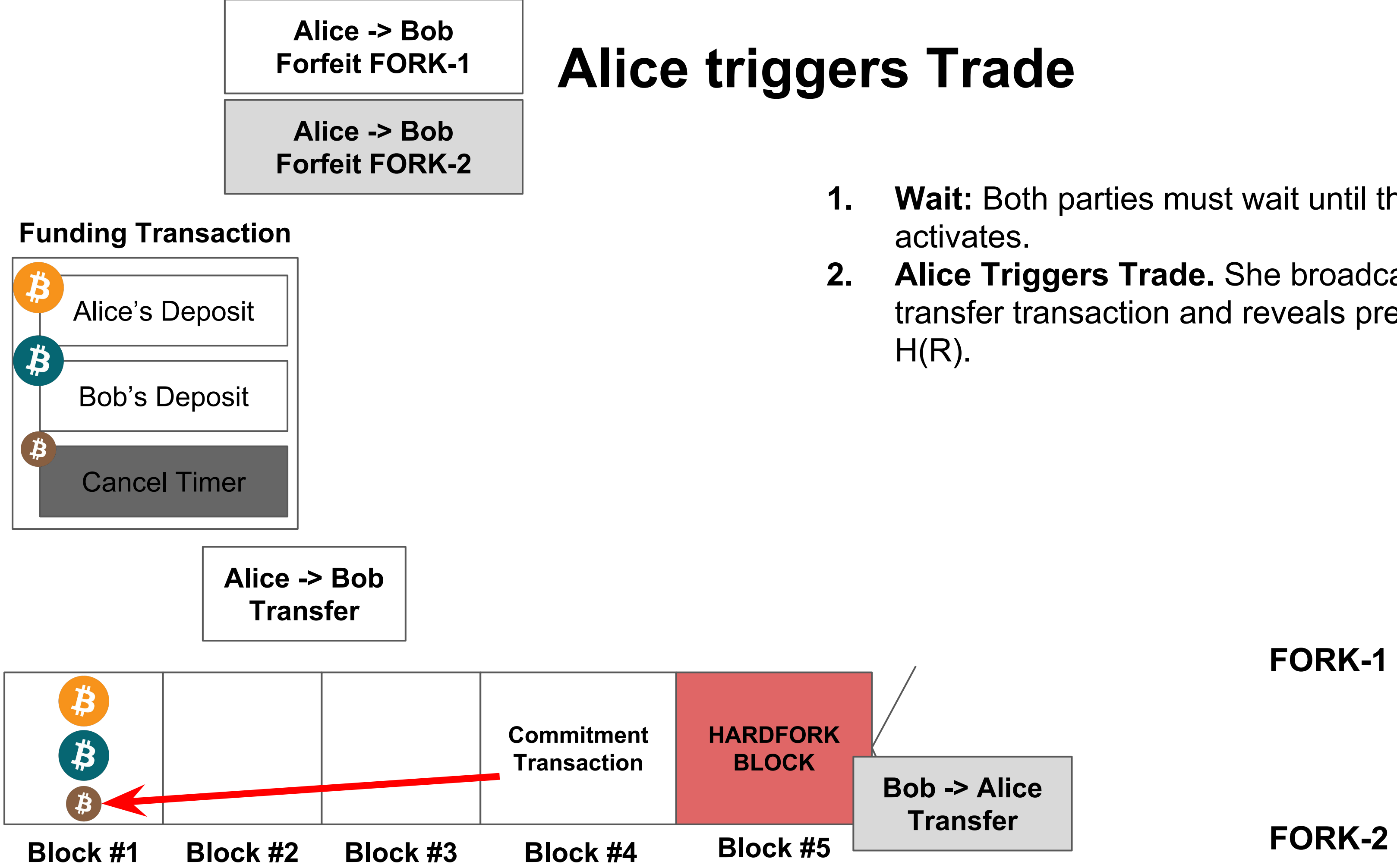
1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade.** She broadcasts Alice -> Bob transfer transaction and reveals pre-image R of H(R).





Alice triggers Trade

1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade.** She broadcasts Alice -> Bob transfer transaction and reveals pre-image R of H(R).



Alice -> Bob Forfeit FORK-1

Alice -> Bob Forfeit FORK-2

Funding Transaction

Alice's Deposit

Bob's Deposit

Cancel Timer

Alice -> Bob Transfer

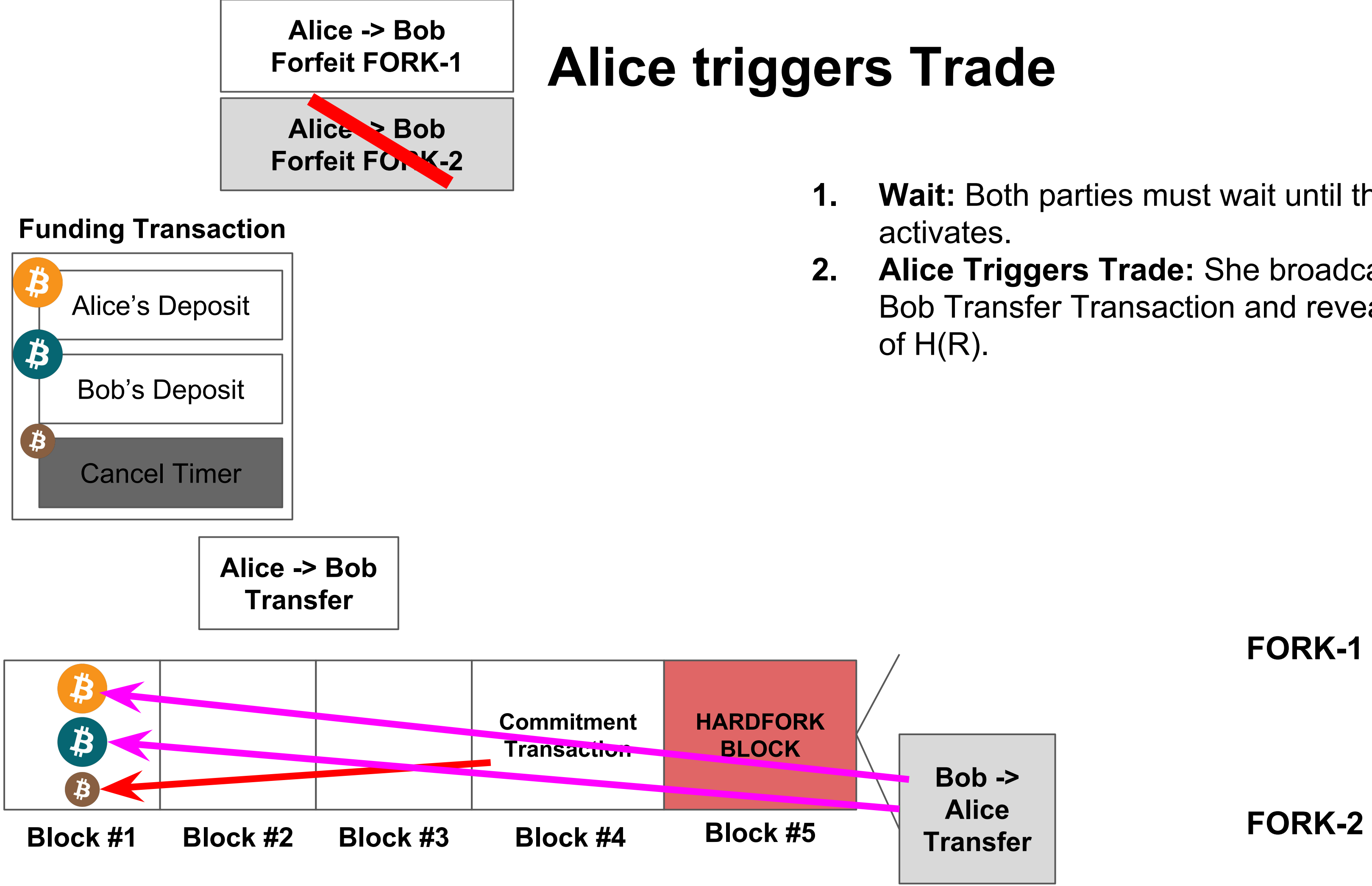
Commitment Transaction

HARDFORK BLOCK

Bob -> Alice Transfer

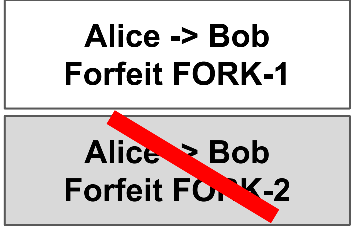
Alice triggers Trade

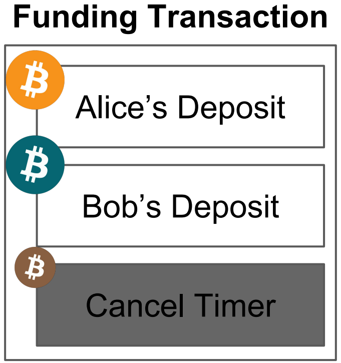
1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).

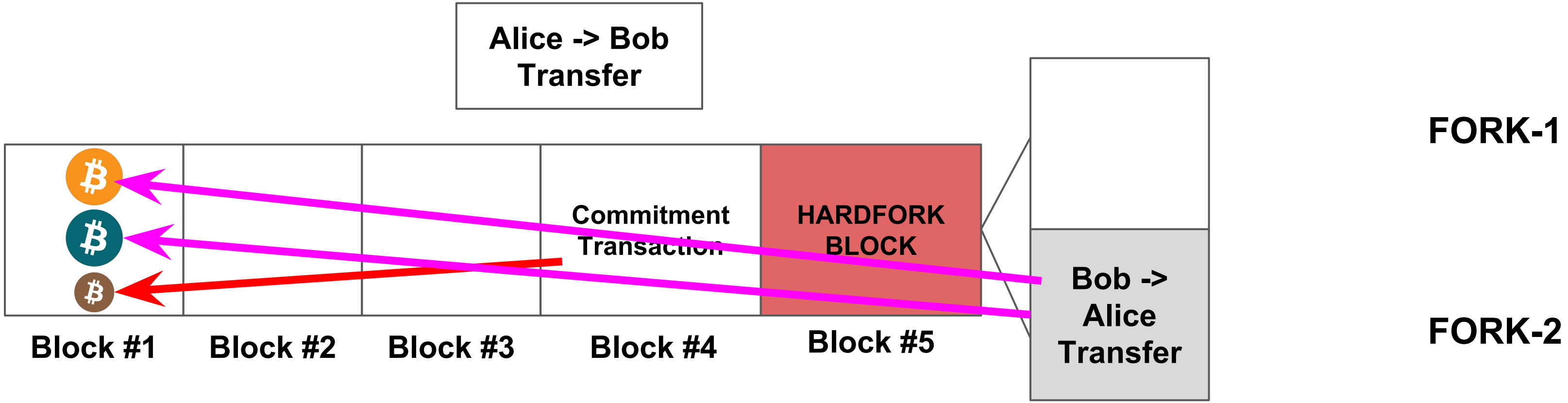


Bob claims his coins!

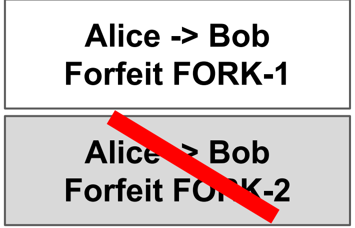
1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.

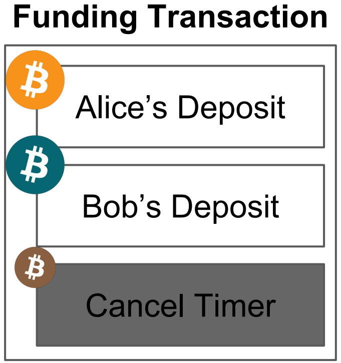




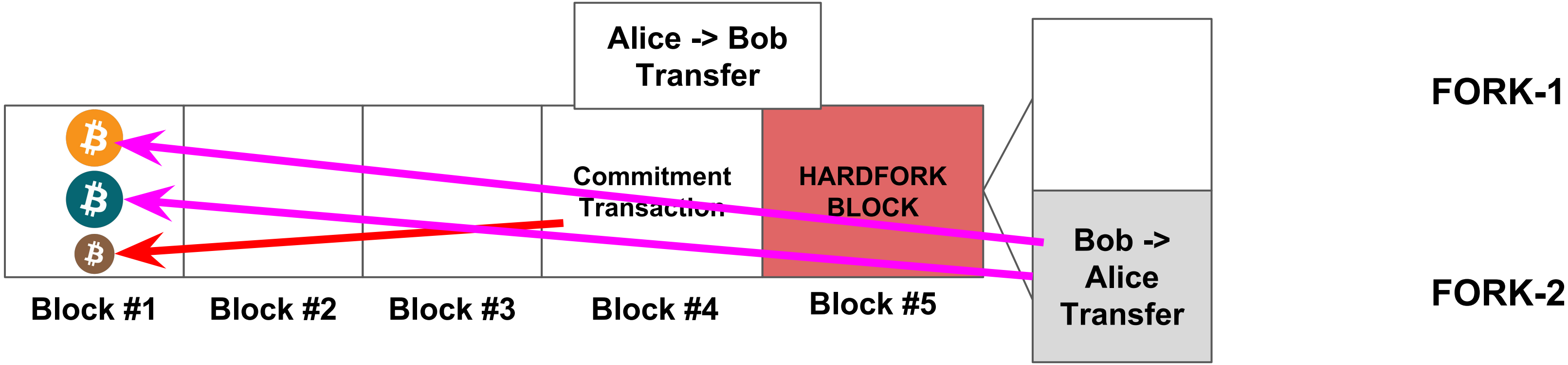


Bob claims his coins!

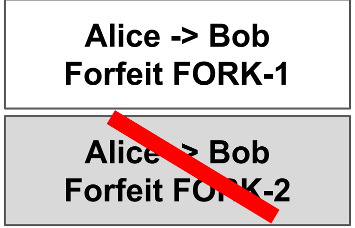




1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.



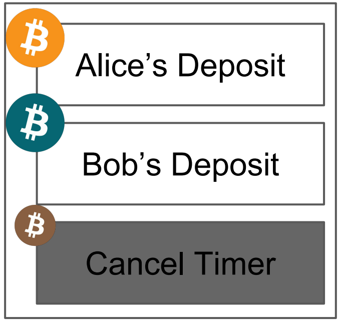
Bob claims his coins!



Alice -> Bob
Forfeit FORK-1

~~Alice -> Bob
Forfeit FORK-2~~

Funding Transaction

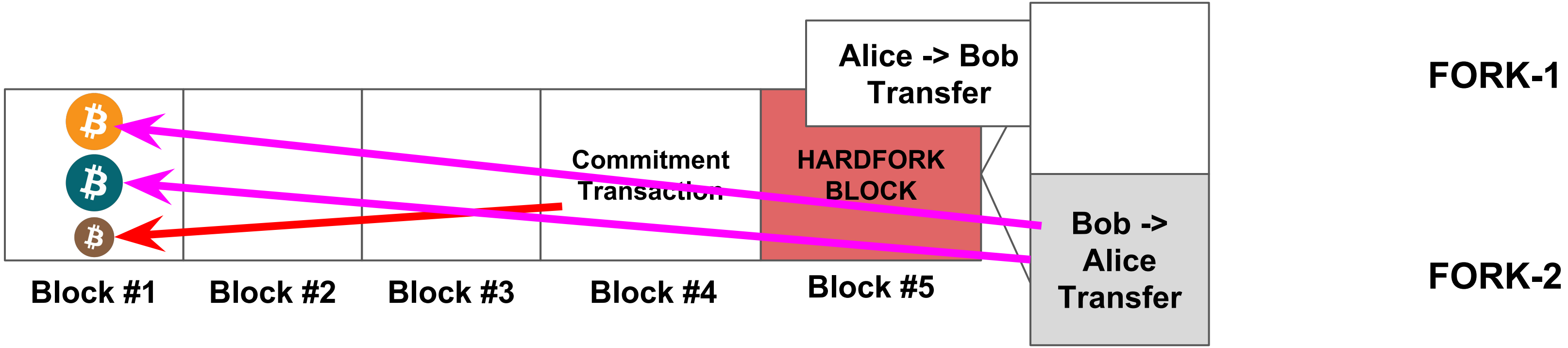


Alice's Deposit

Bob's Deposit

Cancel Timer

1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.



Alice -> Bob
Transfer

Commitment
Transaction

HARDFORK
BLOCK

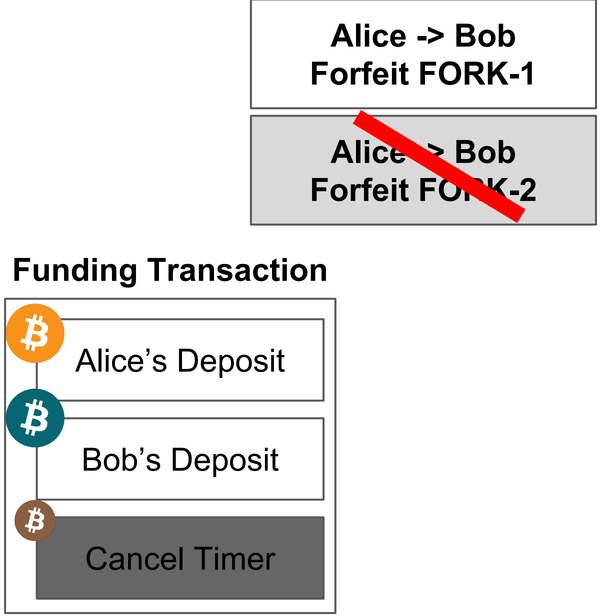
Bob ->
Alice
Transfer

FORK-1

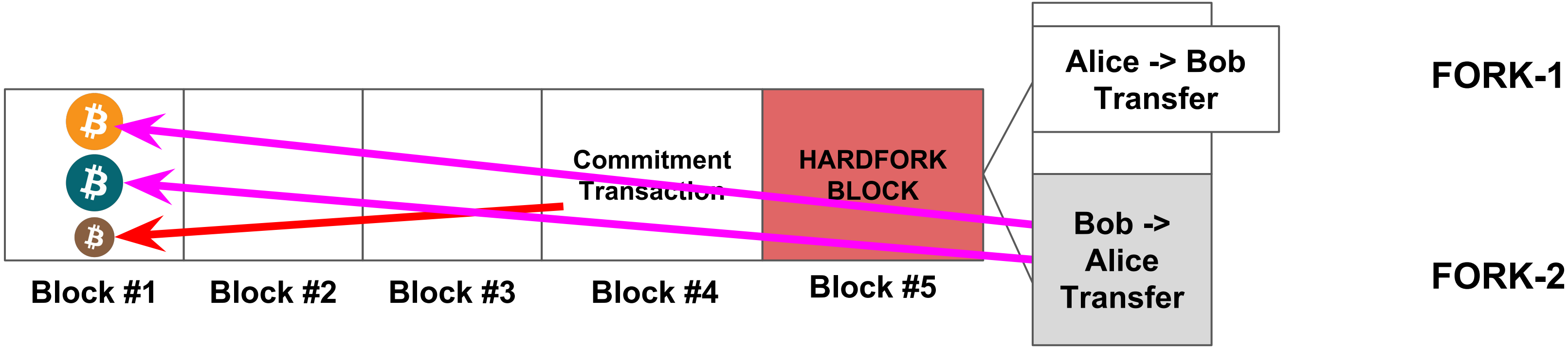
FORK-2

Block #1 Block #2 Block #3 Block #4 Block #5

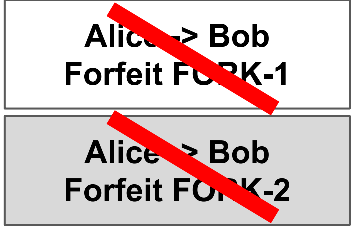
Bob claims his coins!



1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.



Bob claims his coins!

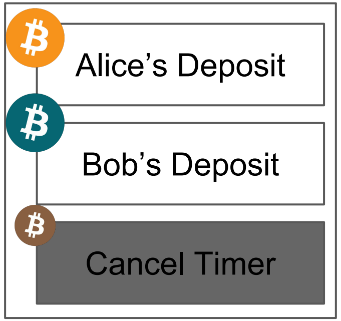


~~Alice -> Bob~~
~~Forfeit FORK-1~~

~~Alice -> Bob~~
~~Forfeit FORK-2~~

1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.

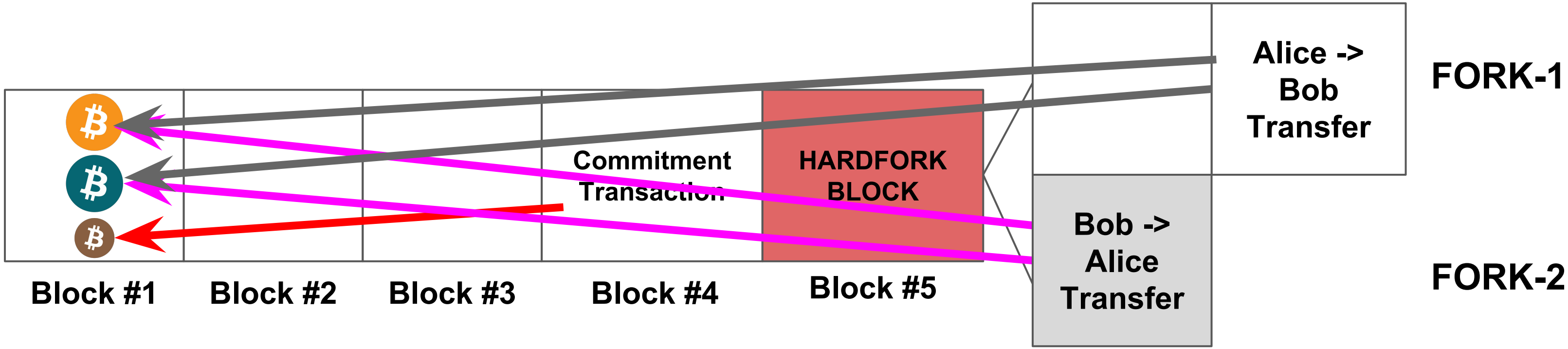
Funding Transaction



Alice's Deposit

Bob's Deposit

Cancel Timer



Block #1

Block #2

Block #3

Block #4

Commitment Transaction

HARDFORK BLOCK

Block #5

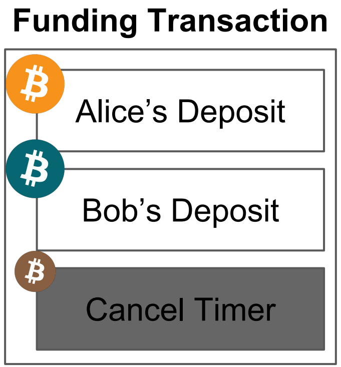
Alice -> Bob Transfer

FORK-1

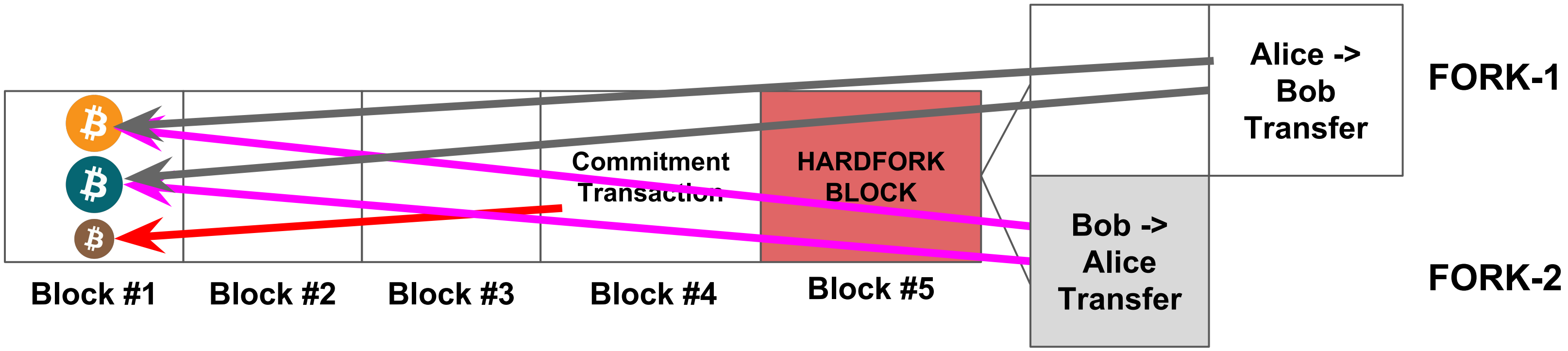
Bob -> Alice Transfer

FORK-2

All done!



1. **Wait:** Both parties must wait until the hardfork activates.
2. **Alice Triggers Trade:** She broadcasts Alice -> Bob Transfer Transaction and reveals pre-image R of H(R).
3. **Bob Claims Coins:** He finds R, and then broadcasts Bob -> Alice Transfer Transaction.
4. All done!



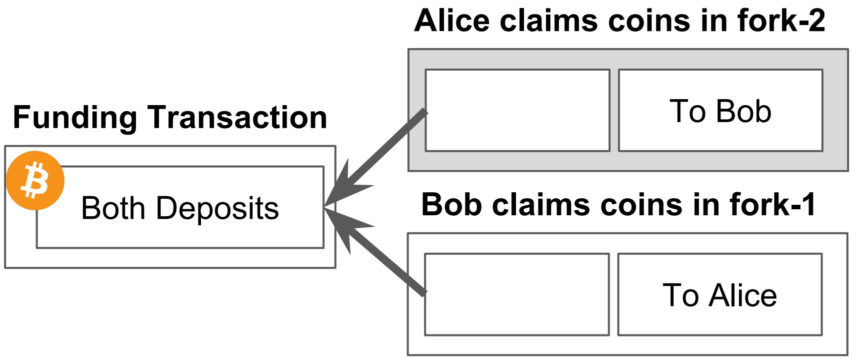
What are the problems?

- Elaborate
  - Four off-chain transaction required to set it up (and the bitcoin script is somewhat complex too)
- Potential to lock coins for long time
  - If Alice doesn’t sign cancellation transaction, then coins are locked up and eventually refunded after the hardfork.
- Hardfork Time must be FIXED.
  - If the hardfork is delayed after setup - Bob can potentially run away with all the coins!

…… What if Transaction Malleability is fixed?

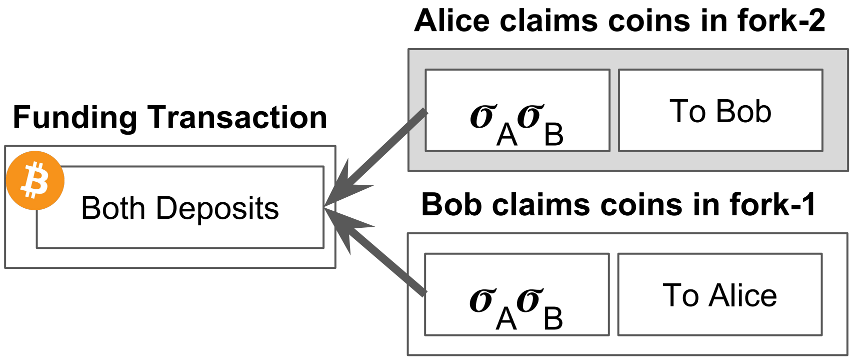
SEGWIT

Create 3 Transactions



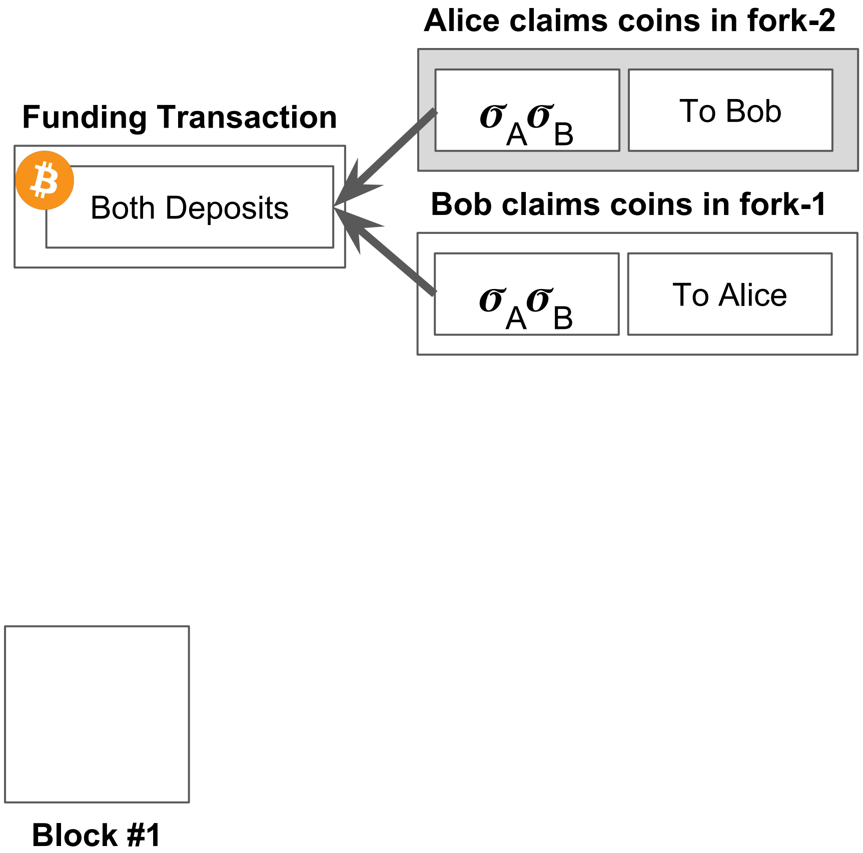
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.

Sign Transfer Transactions



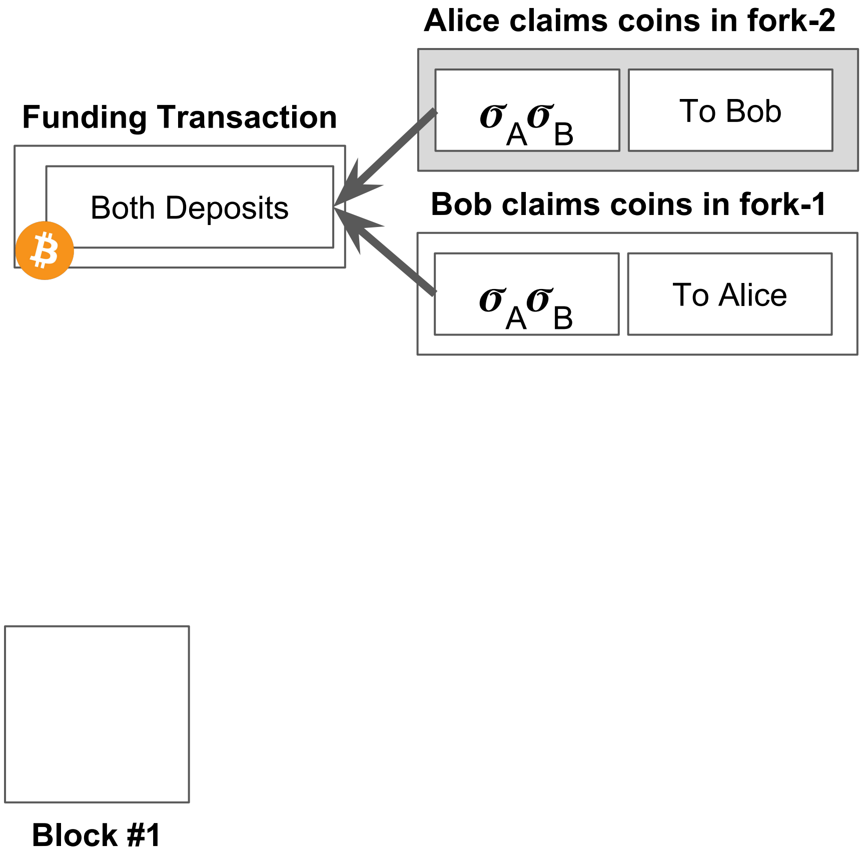
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.

Both Parties Sign and Publish Funding Tx



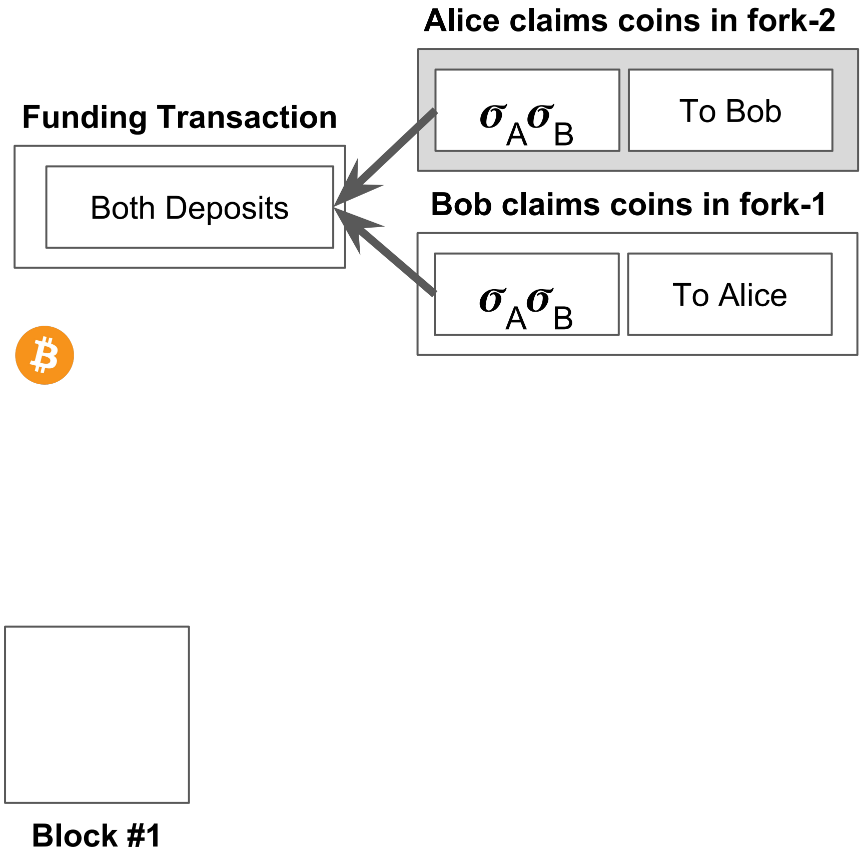
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.

Both Parties Sign and Publish Funding Tx



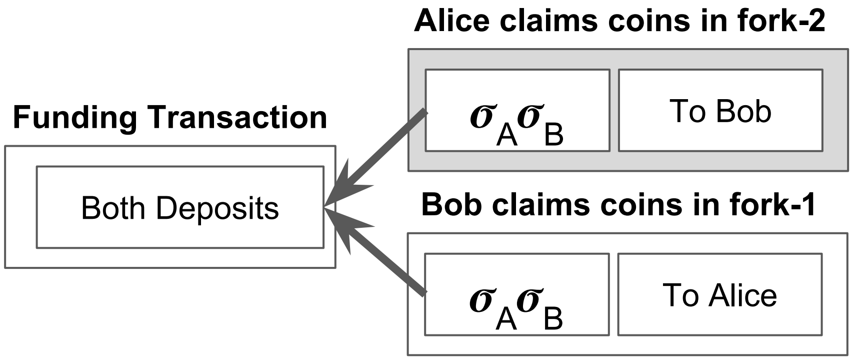
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.

Both Parties Sign and Publish Funding Tx

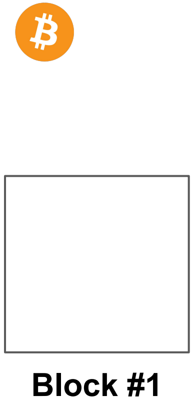


1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.

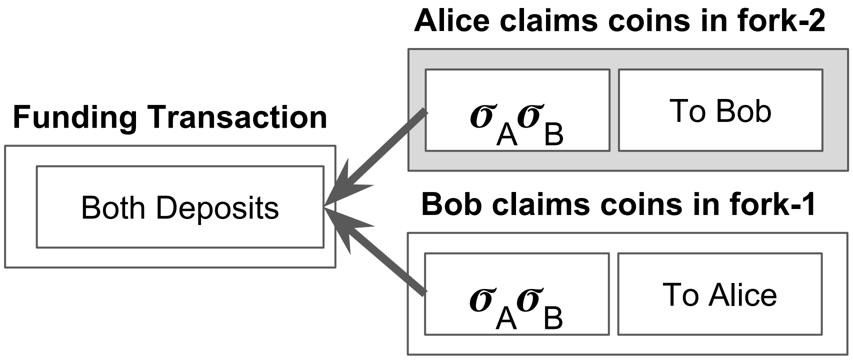
Both Parties Sign and Publish Funding Tx



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.



Both Parties Sign and Publish Funding Tx



Alice claims coins in fork-2

$\sigma_A\sigma_B$

To Bob

Funding Transaction

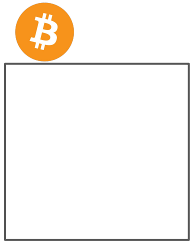
Both Deposits

Bob claims coins in fork-1

$\sigma_A\sigma_B$

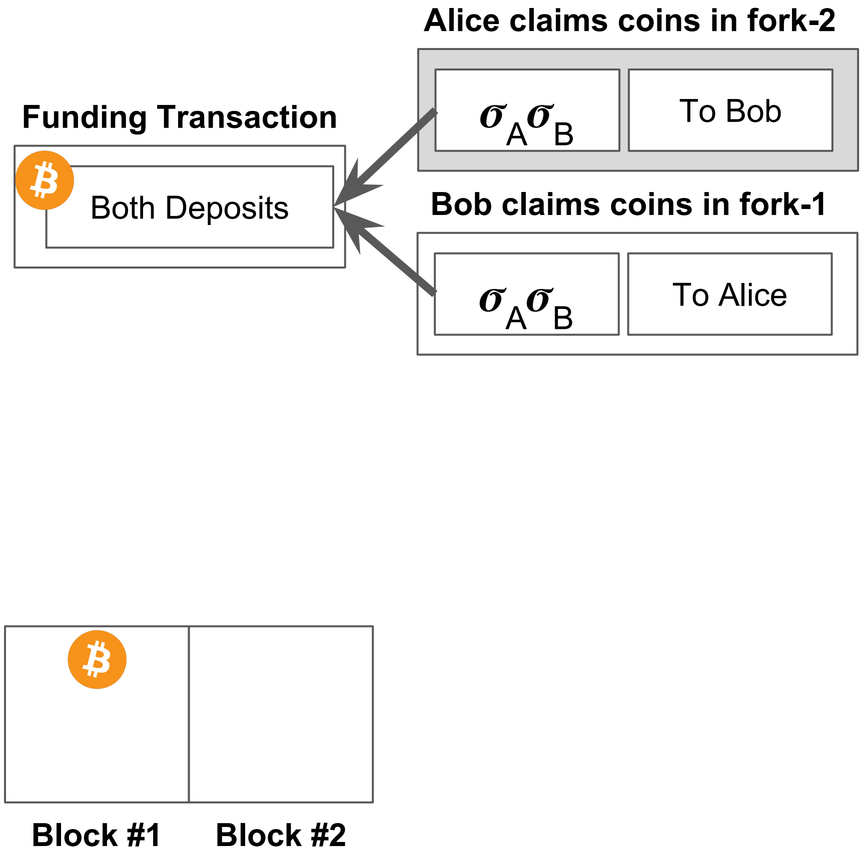
To Alice

1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.



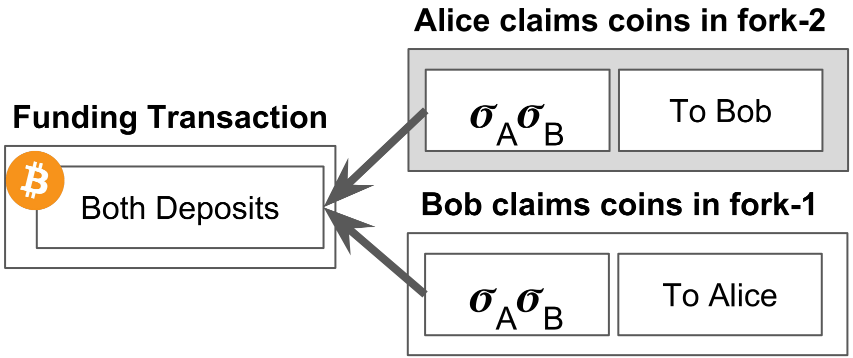
Block #1

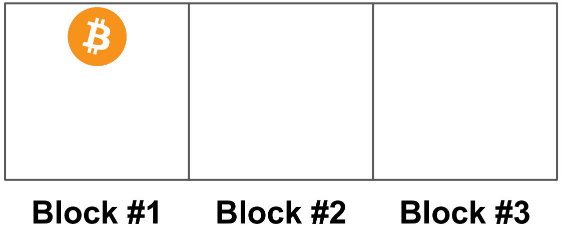
Wait for hardfork



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate

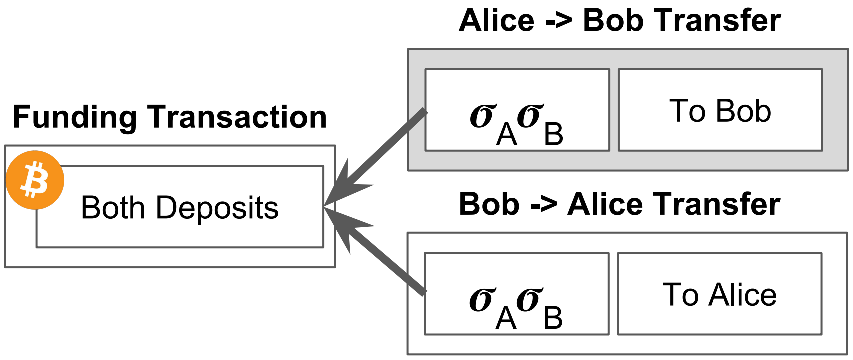
Wait for hardfork



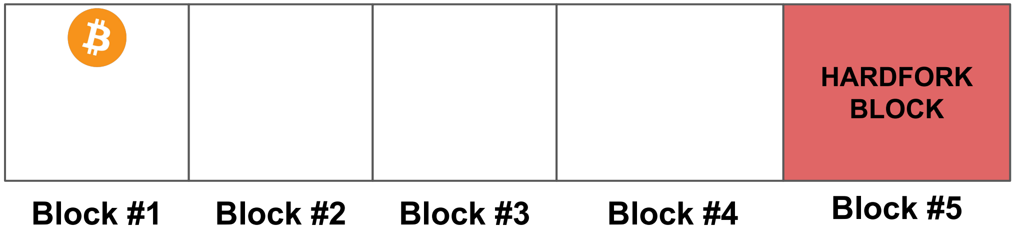


1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate

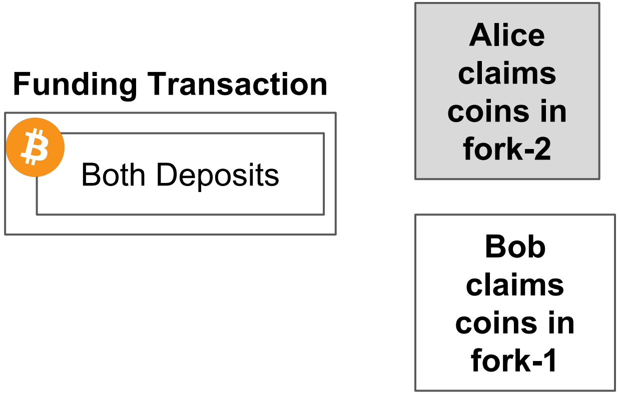
Wait for hardfork



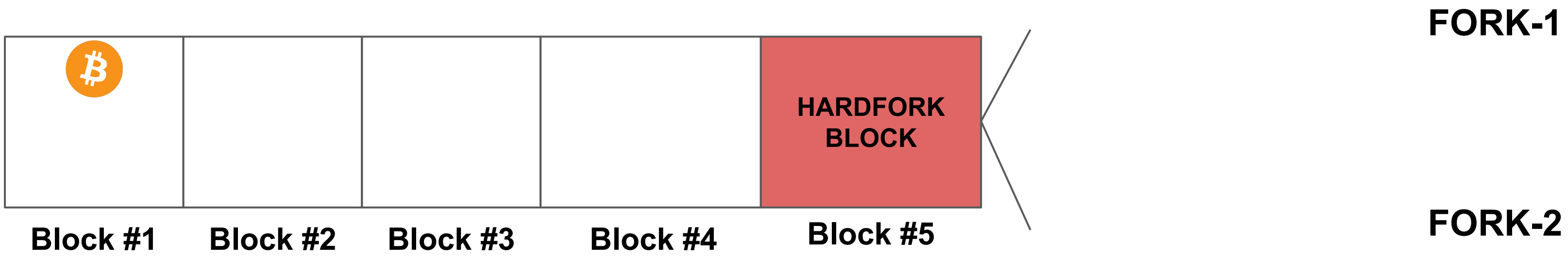
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate



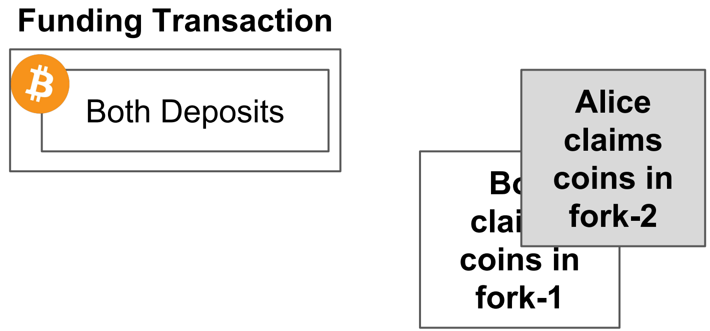
Both parties can claim after hardfork!



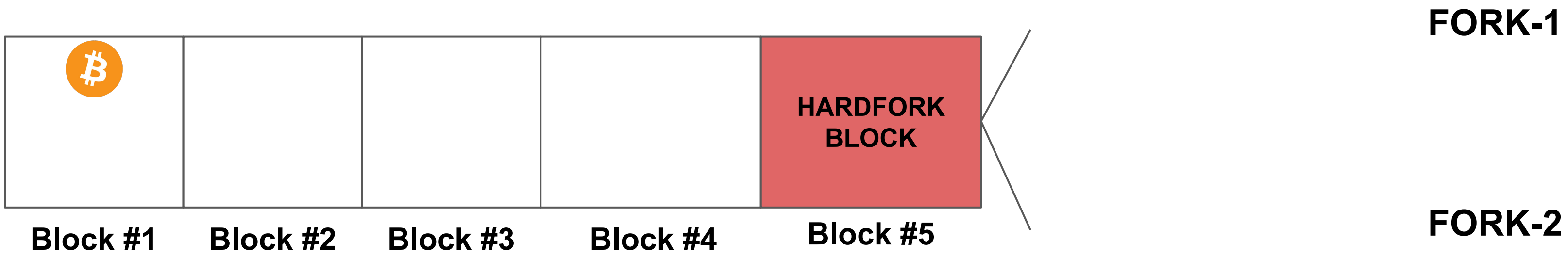
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.



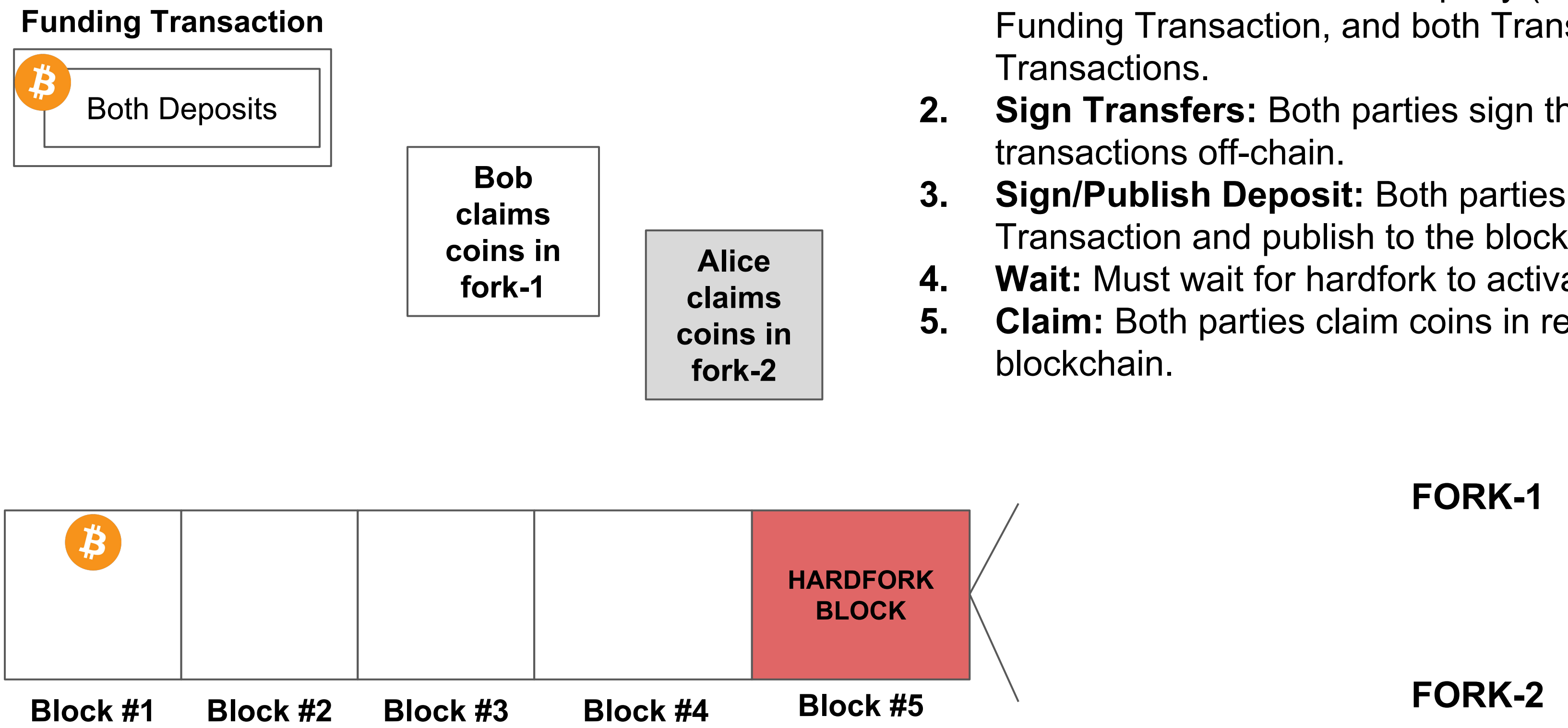
Both parties can claim after hardfork!



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.

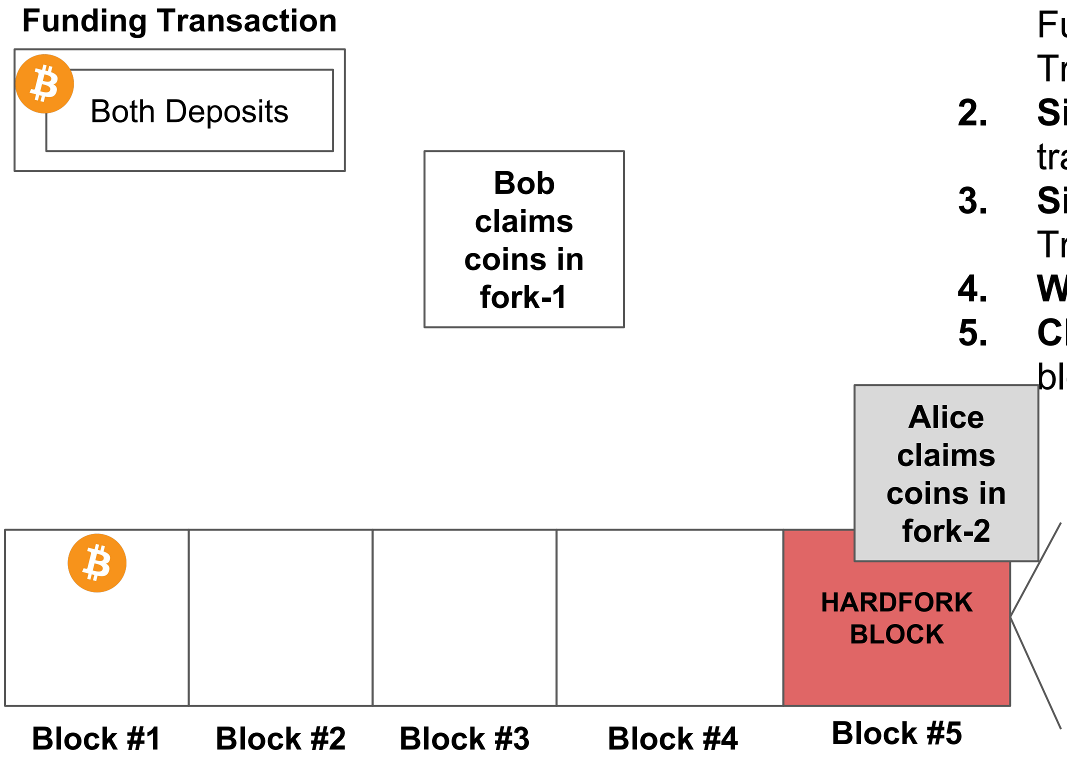


Both parties can claim after hardfork!



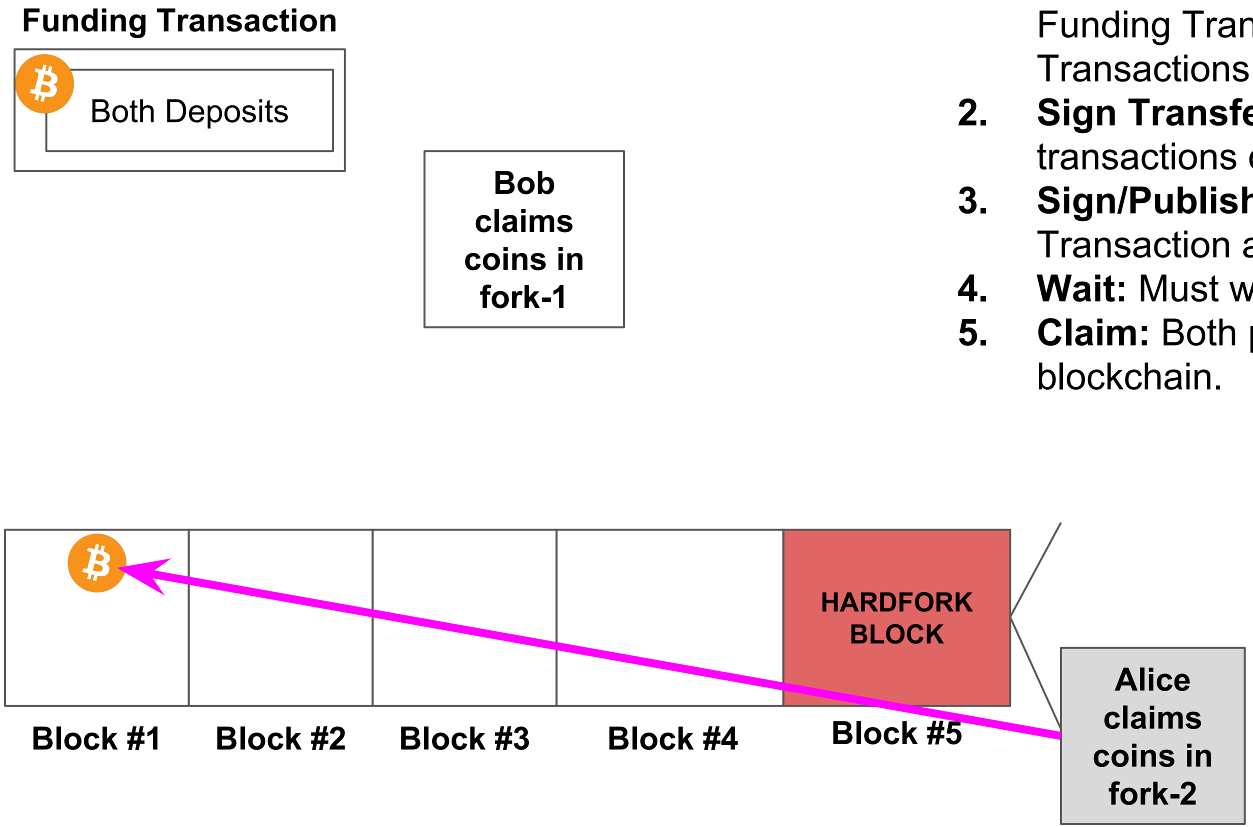
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.

Both parties can claim after hardfork!



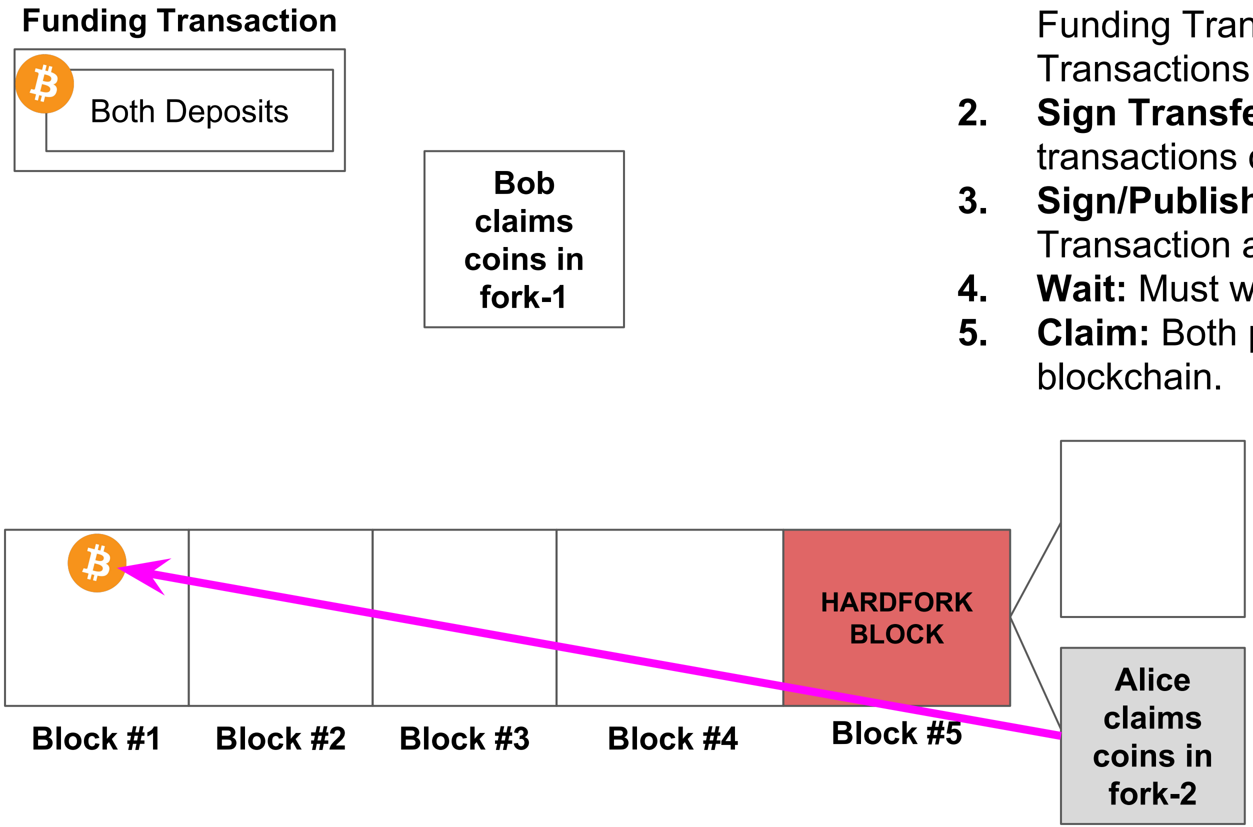
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.

Both parties can claim after hardfork!



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.

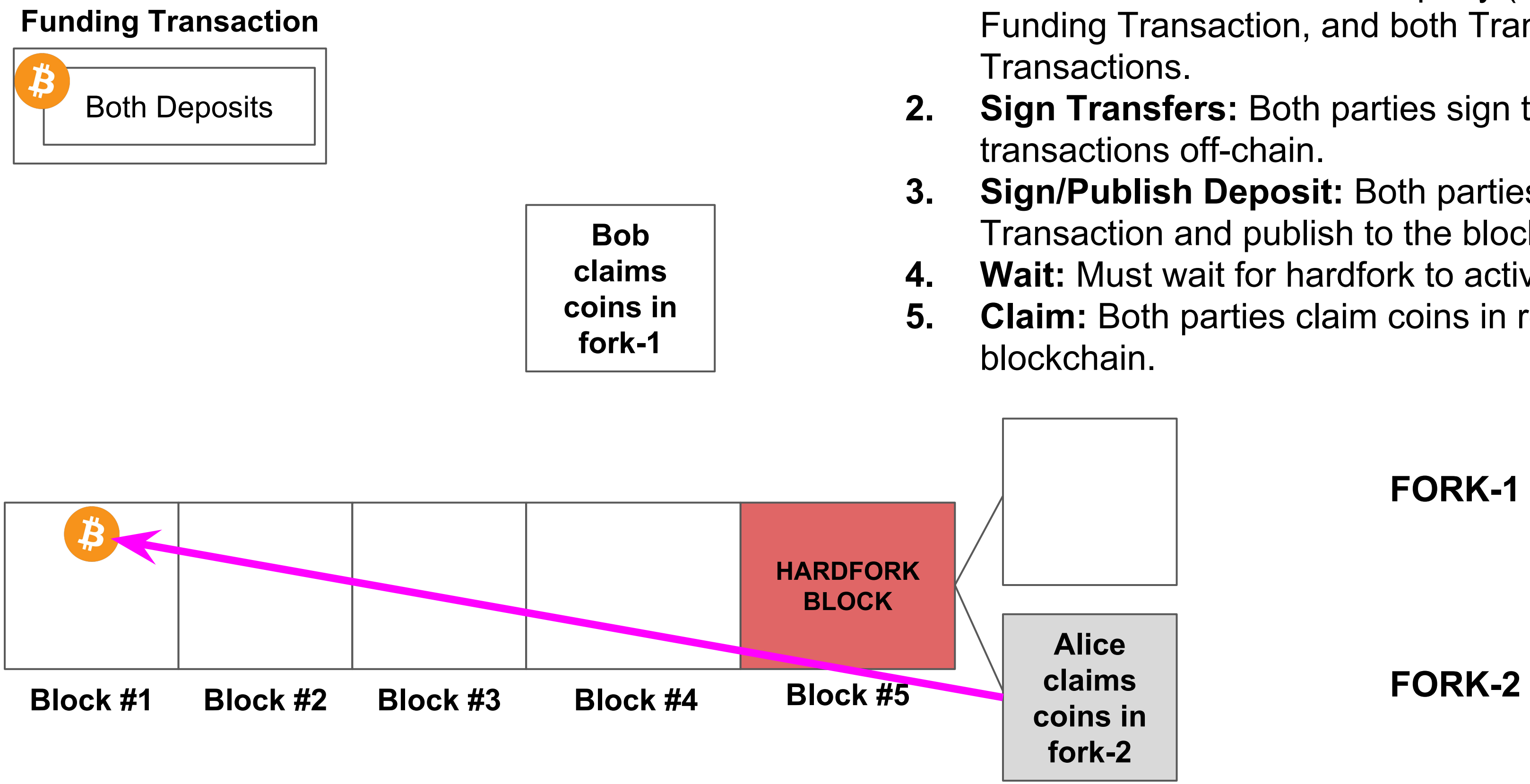
Both parties can claim after hardfork!



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.

Both parties can claim after hardfork!

1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.



Funding Transaction

Both Deposits

Bob claims coins in fork-1

HARDFORK BLOCK

Block #1 **Block #2** **Block #3** **Block #4** **Block #5**

Alice claims coins in fork-2

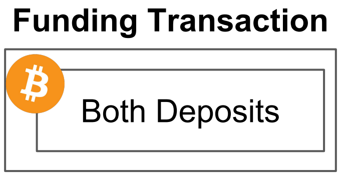
FORK-1

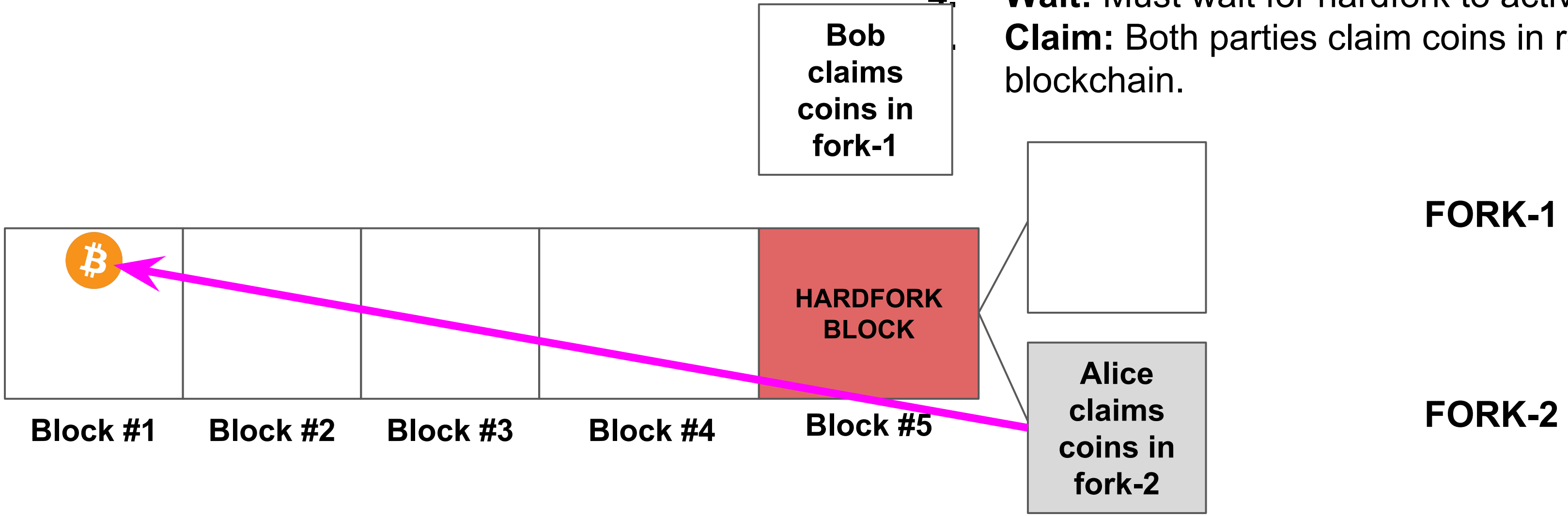
FORK-2

Both parties can claim after hardfork!

1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate

. **Claim:** Both parties claim coins in respective blockchain.



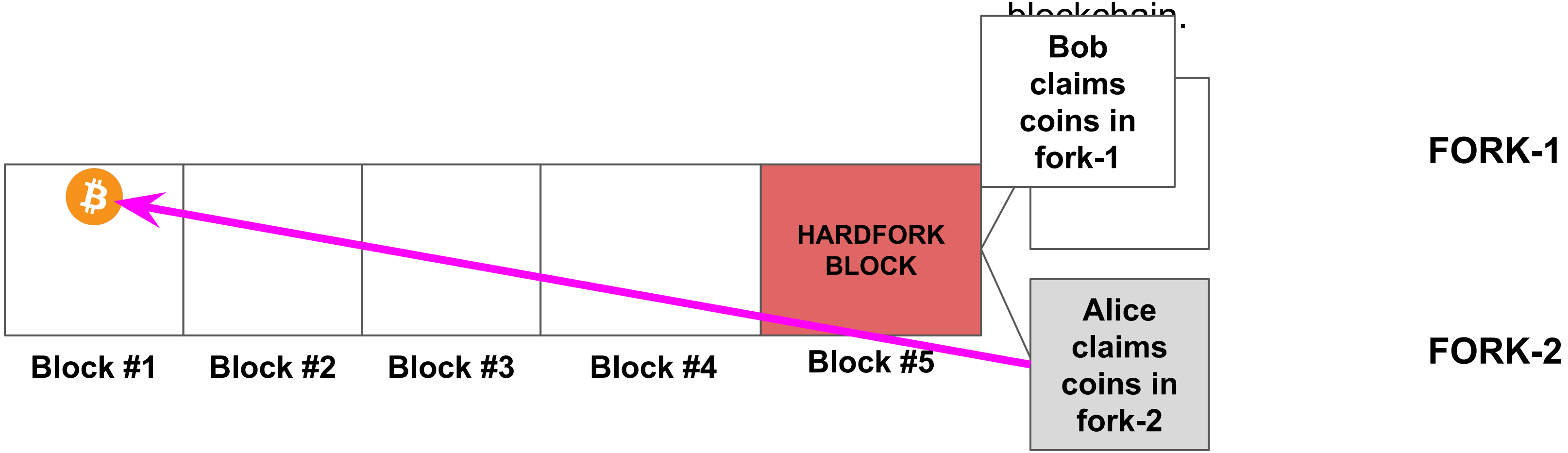


Both parties can claim after hardfork!

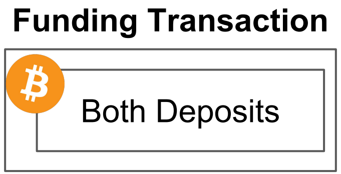
Funding Transaction

Both Deposits

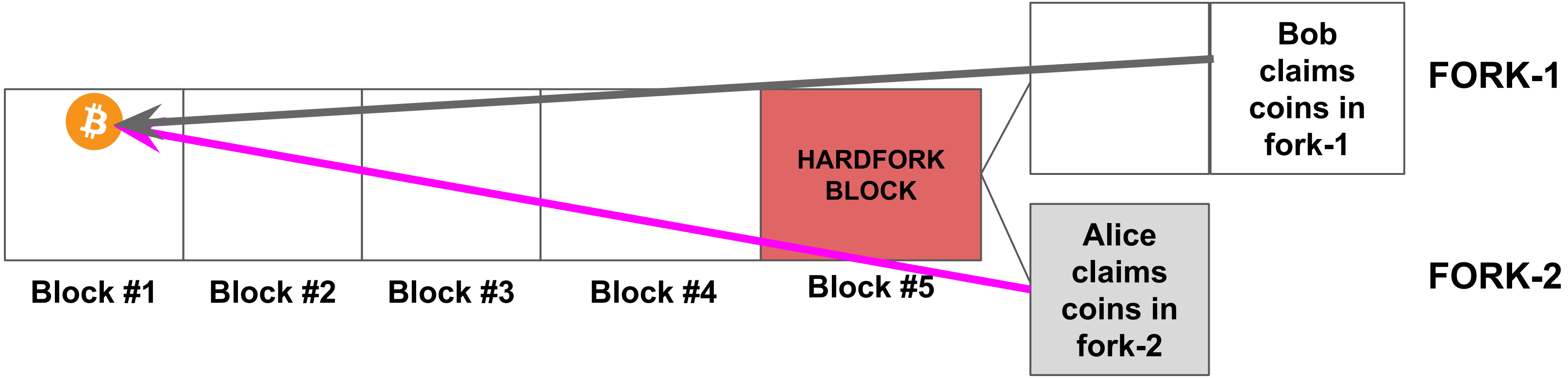
1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.



Both parties can claim after hardfork!



1. **Create Transactions:** One party (i.e. Alice) creates Funding Transaction, and both Transfer Transactions.
2. **Sign Transfers:** Both parties sign the transfer transactions off-chain.
3. **Sign/Publish Deposit:** Both parties sign Funding Transaction and publish to the blockchain.
4. **Wait:** Must wait for hardfork to activate
5. **Claim:** Both parties claim coins in respective blockchain.



How easy was that?

- Similar to establishing a basic payment channel
- No need for either party to trigger the exchange
- Hardfork time must still be FIXED… but no need for elaborate setup.
- Coins not locked for long time… (1 block after hardfork time).

… but when will this ***actually*** be useful?

...Segwit2x if replay protection is incorporated...

paddyncl http://homepages.cs.ncl.ac.uk/patrick.mc-corry/atomically-trading-roger.pdf