TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

S&P SECURITY & PRIVACY GROUP

Blitz: Secure Multi-Hop Payments Without Two-Phase Commits

Lukas Aumayr[1], Pedro Moreno-Sanchez[2], Aniket Kate[3], Matteo Maffei[1]

[1]TU Wien, [2]IMDEA Software Institute, [3]Purdue University

What's in store?

- 1) Drawbacks of LN
- 2) Blitz construction
- Discussion

USENIX Security '21

Blitz: Secure Multi-Hop Payments Without Two-Phase Commits

Lukas Aumayr
TU Wien
lukas.aumayr@tuwien.ac.at

Pedro Moreno-Sanchez
IMDEA Software Institute
pedro.moreno@imdea.org

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Abstract

Payment-channel networks (PCN) are the most prominent approach to tackle the scalability issues of current permissionless blockchains. A PCN reduces the load on-chain by allowing arbitrarily many off-chain multi-hop payments (MHPs) between any two users connected through a path of payment channels. Unfortunately, current MHP protocols are far from satisfactory. One-round MHPs (e.g., Interledger) are insecure as a malicious intermediary can steal the payment funds. Two-round MHPs (e.g., Lightning Network (LN)) follow the 2-phase-commit paradigm as in databases to overcome this issue. However, when tied with economical incentives, 2-phase-commit brings other security threats (i.e., wormhole attacks), staggered collateral (i.e., funds are locked for a time proportional to the payment path length) and dependency on specific scripting language functionality (e.g., Hash Time-Lock Contracts) that hinders a wider deployment in practice.

We present Blitz, a novel MHP protocol that demonstrates for the first time that we can achieve the best of the two worlds: a single round MHP where no malicious intermediary can steal coins. Moreover, Blitz provides the same privacy for sender and receiver as current MHP protocols do, is not prone to the wormhole attack and requires only constant collateral. Additionally, we construct MHPs using only digital signatures and a timelock functionality, both available at the core of virtually every cryptocurrency today. We provide the cryptographic details of Blitz and we formally prove its security. Furthermore, our experimental evaluation on a LN snapshot shows that (i) staggered collateral in LN leads to in between 4x and 33x more unsuccessful payments than the constant collateral in Blitz; (ii) Blitz reduces the size of the payment contract by 26%; and (iii) Blitz prevents up to 0.3 BTC (3397 USD in October 2020) in fees being stolen over a three day period as it avoids wormhole attacks by design.

1 Introduction

Permissonless cryptocurrencies such as Bitcoin enable secure payments in a decentralized, trustless environment. Transactions are verified through a consensus mechanism and all valid transactions are recorded in a public, distributed ledger, often called blockchain. This approach has inherent scalability issues and fails to meet the growing user demands: In Bitcoin, the transaction throughput is technically limited to tens of transactions per second and the transaction confirmation time is around an hour. In contrast, more centralized payment networks such as the Visa credit card network, can handle peaks of 47,000 transaction per second.

This scalability issue is an open problem in industry and academia alike [15, 31]. Among the approaches proposed so far, payment channels (PC) have emerged as one of the most promising solutions; implementations thereof are already widely used in practice, e.g., the Lightning Network (LN) [22] in Bitcoin. A PC enables two users to securely perform an arbitrary amount of instantaneous transactions between each other, while burdening the blockchain with merely two transactions, (i) for opening and (ii) for closing. In particular, following the unspent transaction output (UTXO) model, two users open a PC by locking some coins in a shared multi-signature output. By exchanging signed transactions that spend from the shared output in a peer-to-peer fashion, they can capture and redistribute their balances off-chain. Either one of the two users can terminate the PC by publishing the latest of these signed transactions on the blockchain.

As creating PCs requires locking up some coins, it is economically infeasible to set up a PC with every user one wants to interact with. Instead, PCs can be linked together forming a graph known as payment channel network (PCN) [19, 22]. In a PCN, a payment of α coins from a sender U_0 to a receiver U_n can be performed via a path $\{U_i\}_{i\in[0,n]}$ of intermediaries.

1.1 State-of-the-art PCNs

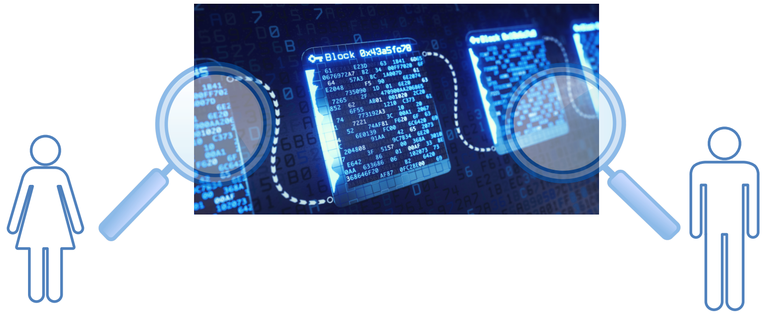
A possible way of achieving such a multi-hop payment (MHP) is an optimistic 1-round approach, e.g., Interledger [27]. Here, U_0 starts paying to its neighbor on the path U_1, who then pays to its neighbor U_2 and so on until U_n is reached. This protocol, however, relies on every intermediary behaving honestly, otherwise any intermediary can trivially steal coins by not forwarding the payment to its neighbor.

Full version: https://eprint.iacr.org/2021/176.pdf

1) Drawbacks of LN

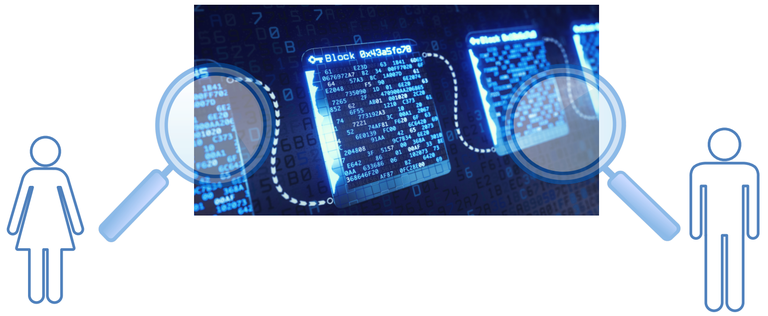
Scalability

Bitcoin's **transaction rate:** ~10 tx/sec
Visa's transaction rate: ~10K tx/sec



Scalability

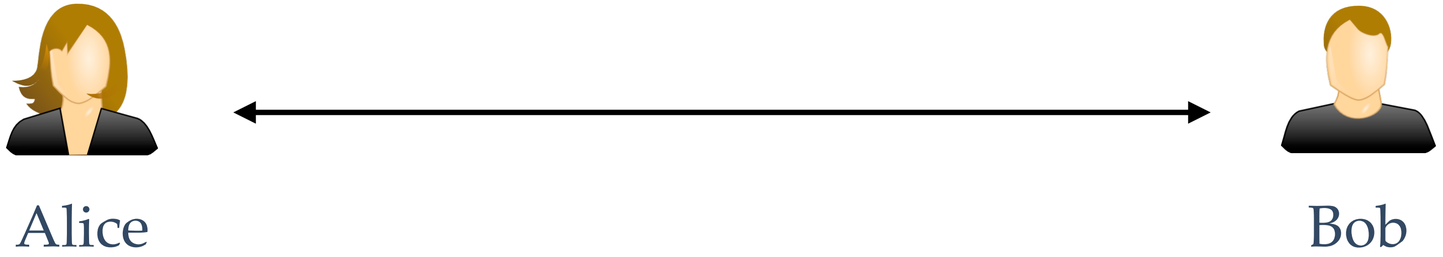
Bitcoin's **transaction rate:** ~10 tx/sec
Visa's transaction rate: ~10K tx/sec



- **On-chain, consensus layer** (tweak consensus)
  e.g., DAG Blockchain, sharding, ...
- **Off-chain, application layer** (local consensus, blockchain only for disputes)
  e.g., Payment Channel Networks

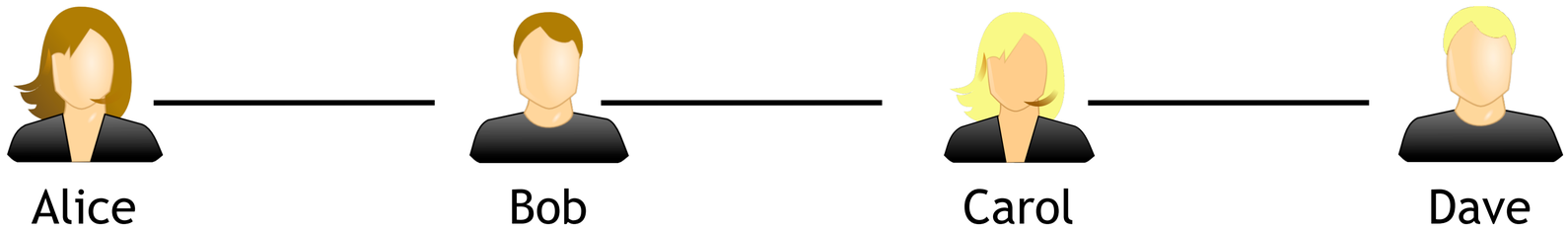
Lightning Network
(Bitcoin)

Payment Channels



- Arbitrarily many payments off-chain
- Only 2 transaction go on-chain

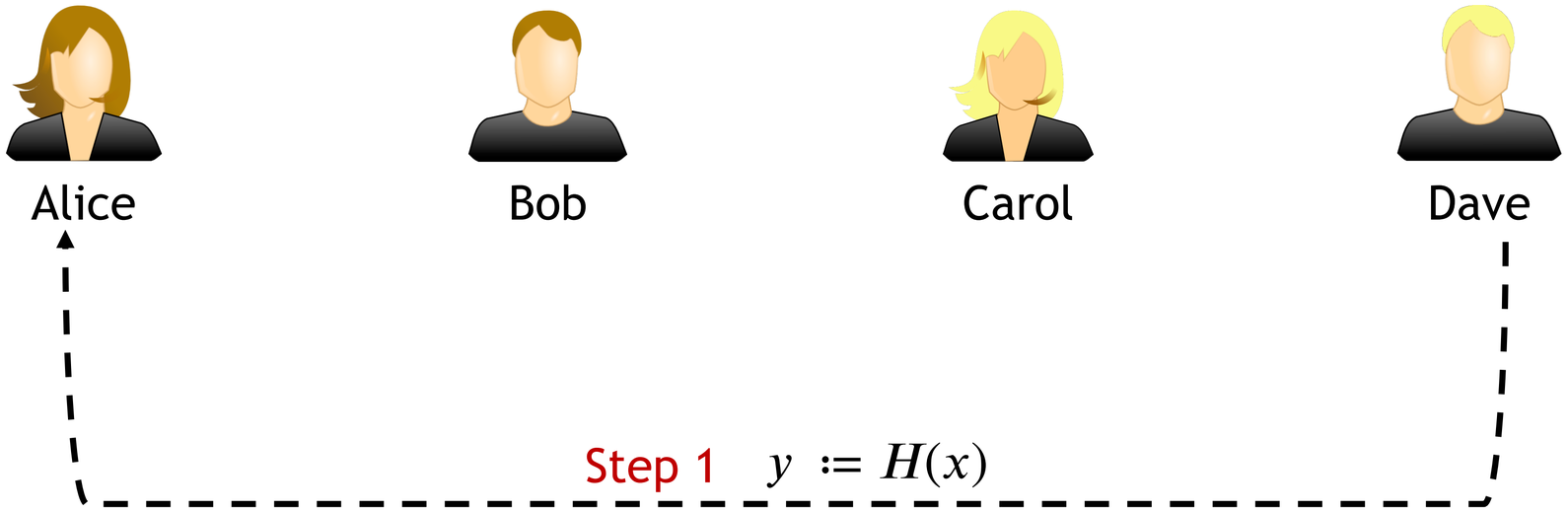
Payment Channel Network (PCN)



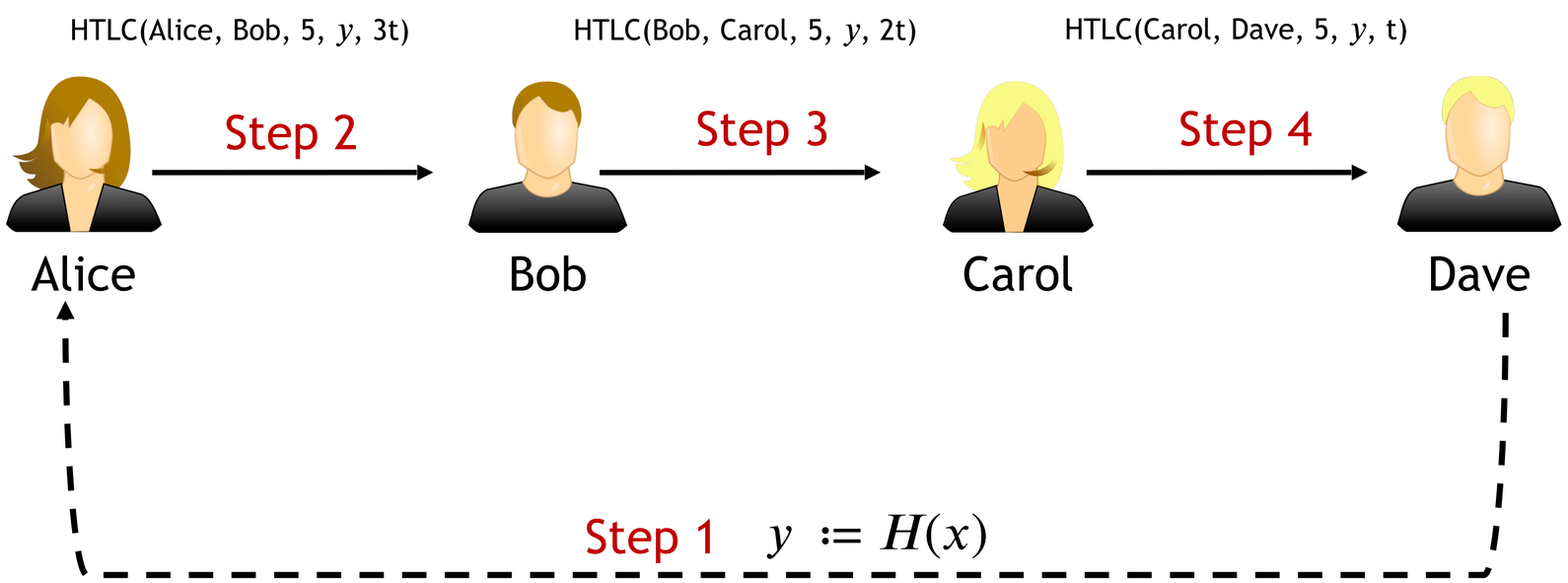
- Infeasible to open channels with everyone
- Link channels to form a PCN
- Multi-hop payments
- e.g., Lightning Network (LN) [1]

[1] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments," *2016*

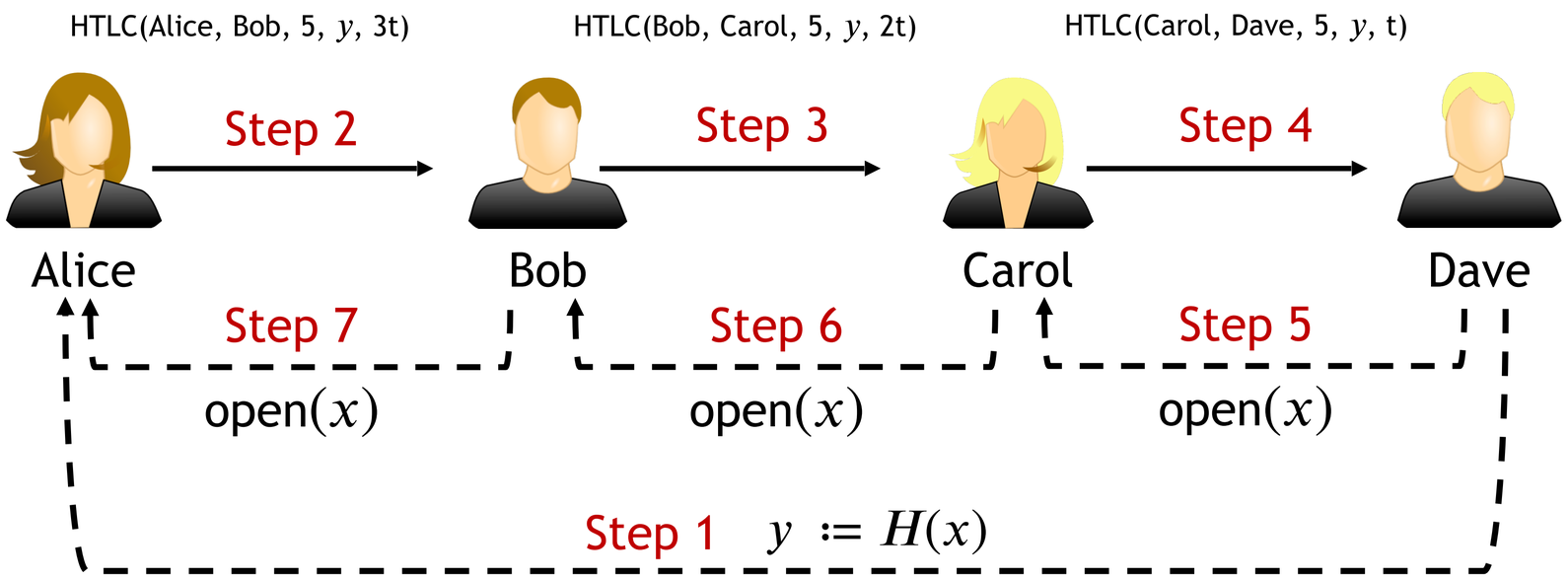
Multi-hop payments in the Lightning Network



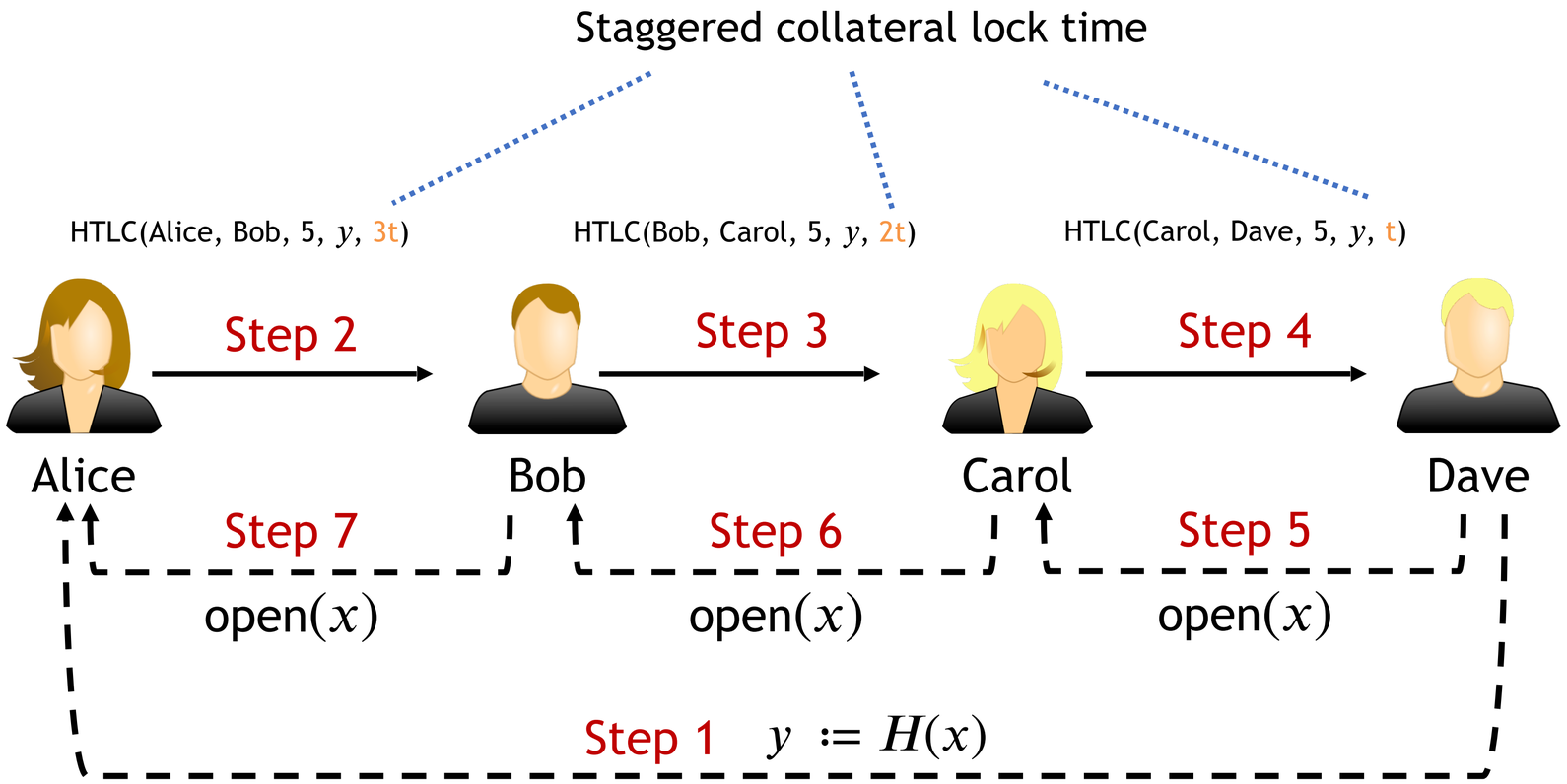
Multi-hop payments in the Lightning Network



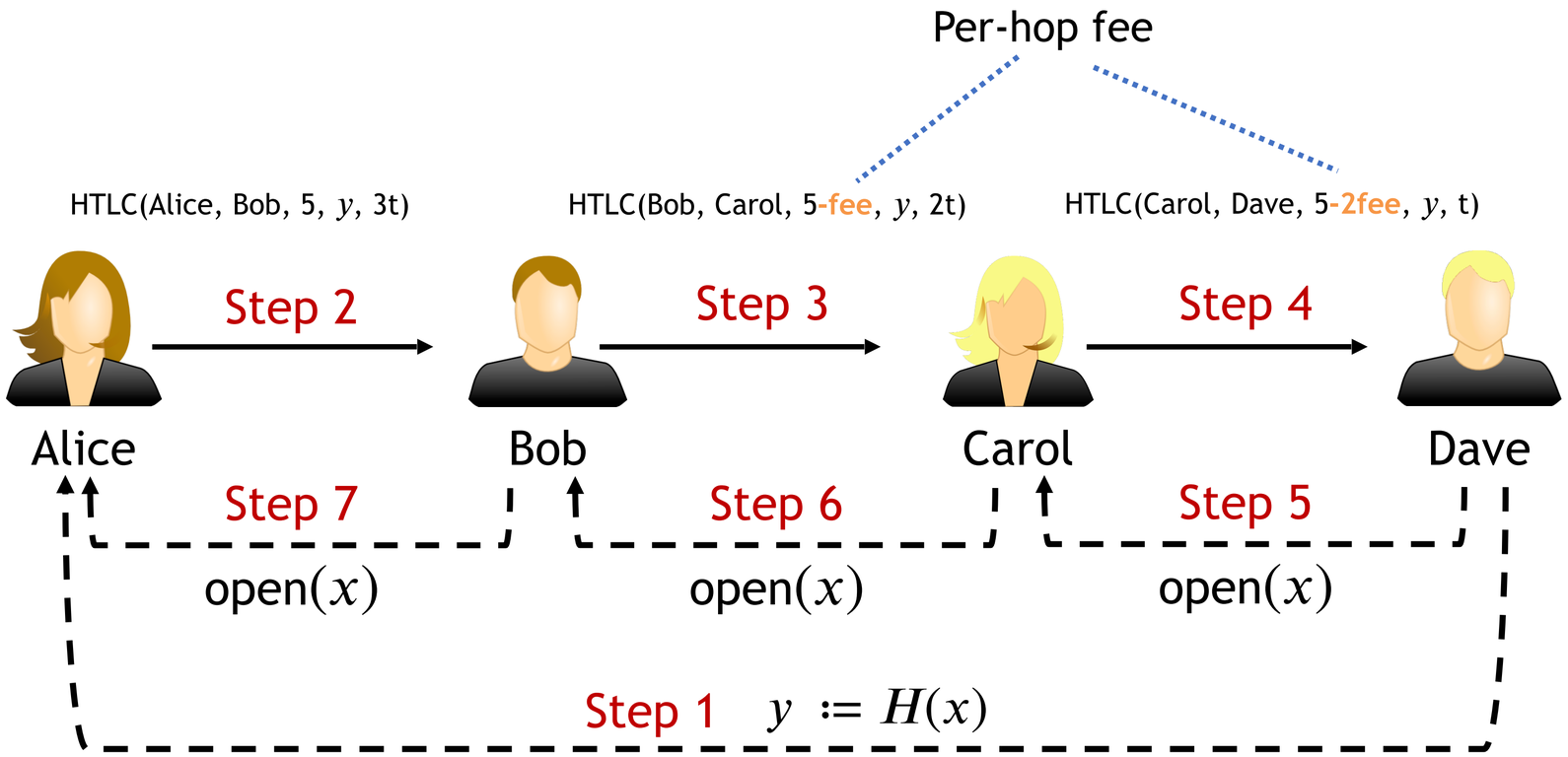
Multi-hop payments in the Lightning Network



Multi-hop payments in the Lightning Network



Multi-hop payments in the Lightning Network



Properties & drawbacks of Lightning payments

Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

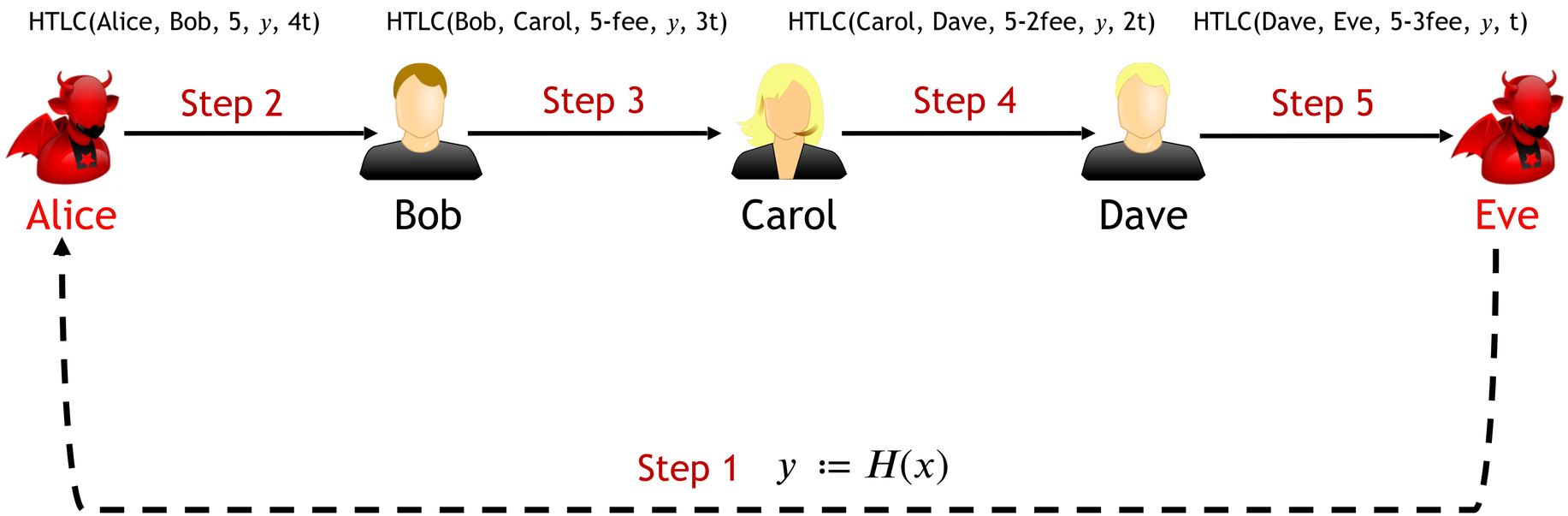
Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput

Griefing attack



- Attacker controls nodes Alice and Eve
- Set up payment to self
- Don’t complete payment, money locked

Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput

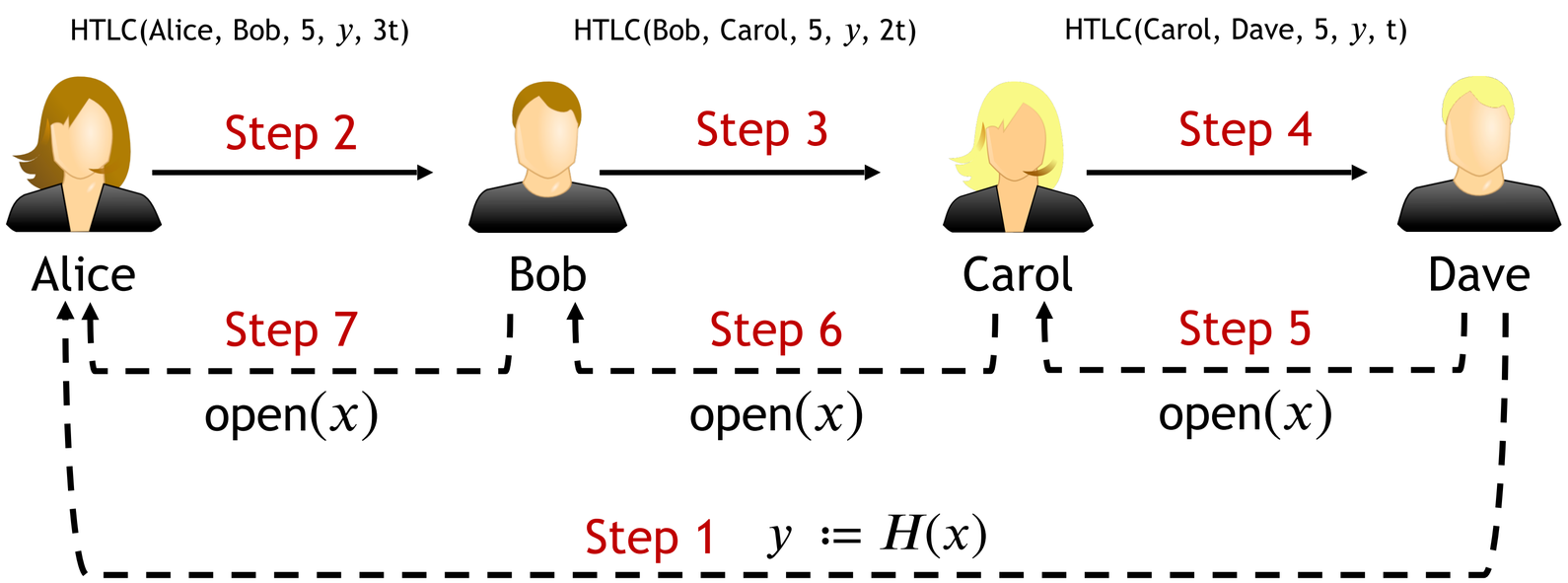
Properties & drawbacks of Lightning payments

- “Balance Security” ✔

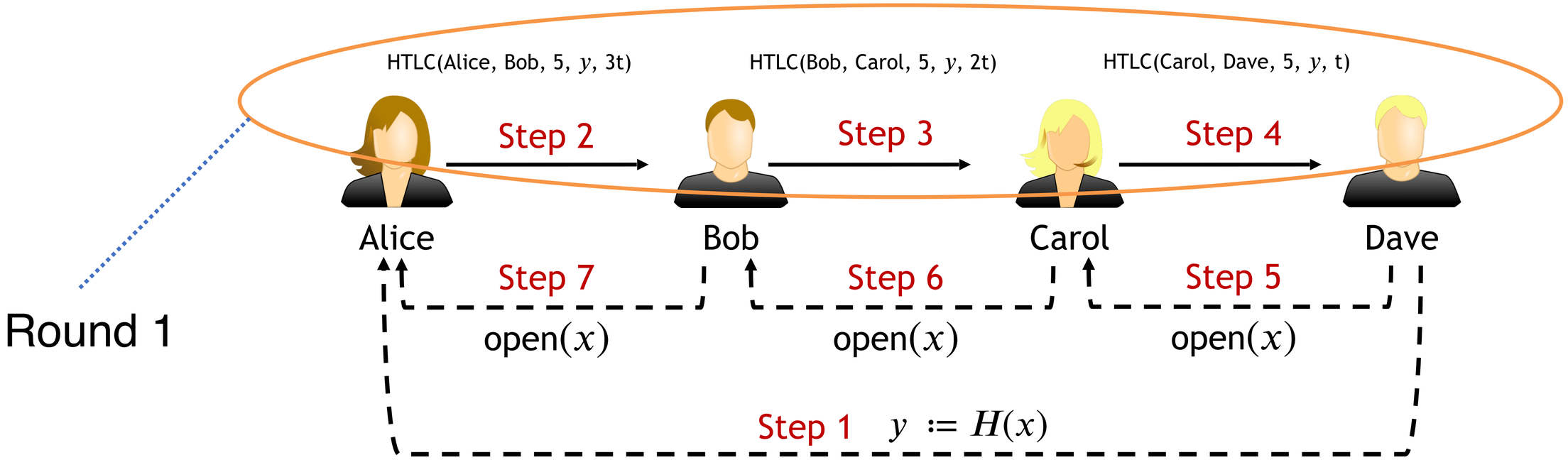
Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput
- Takes two rounds ✖

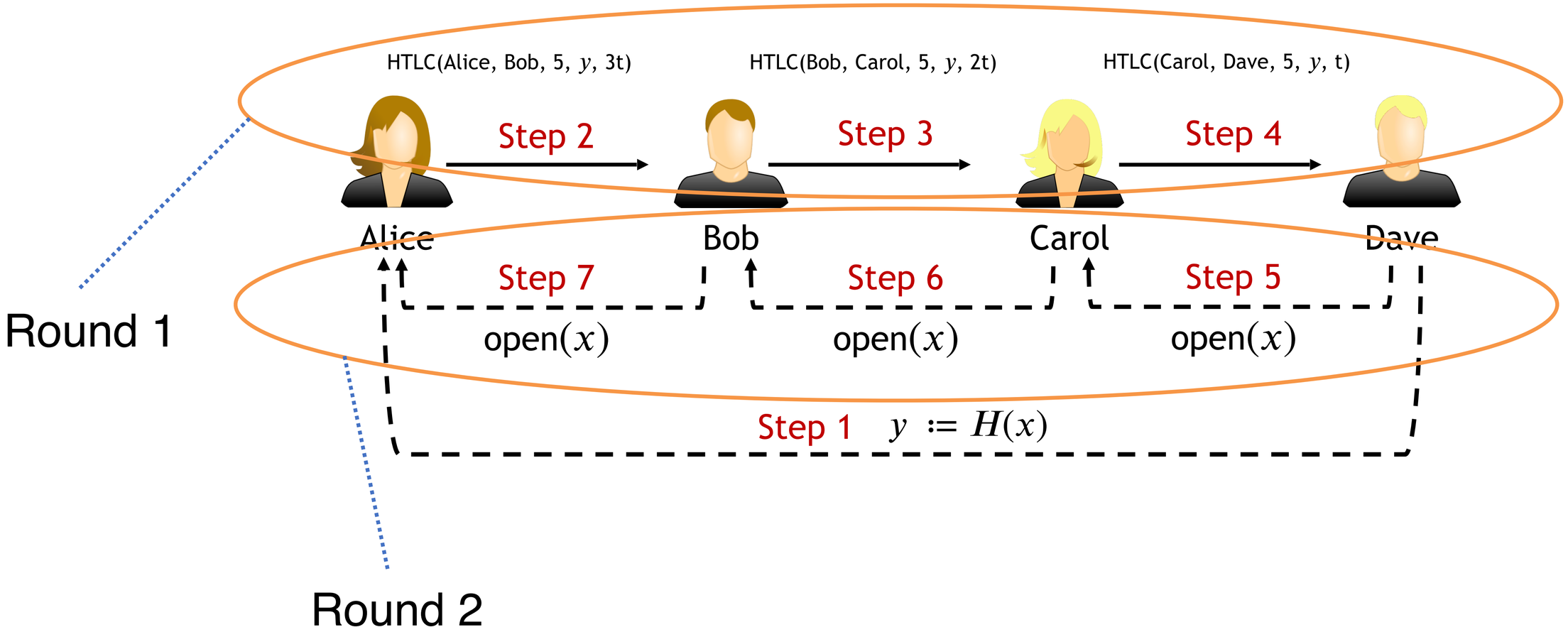
What do we mean by “round”?



What do we mean by “round”?



What do we mean by “round”?



Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput
- Takes two rounds ✖

Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput
- Takes two rounds ✖
- HTLC scripting requirements ✖

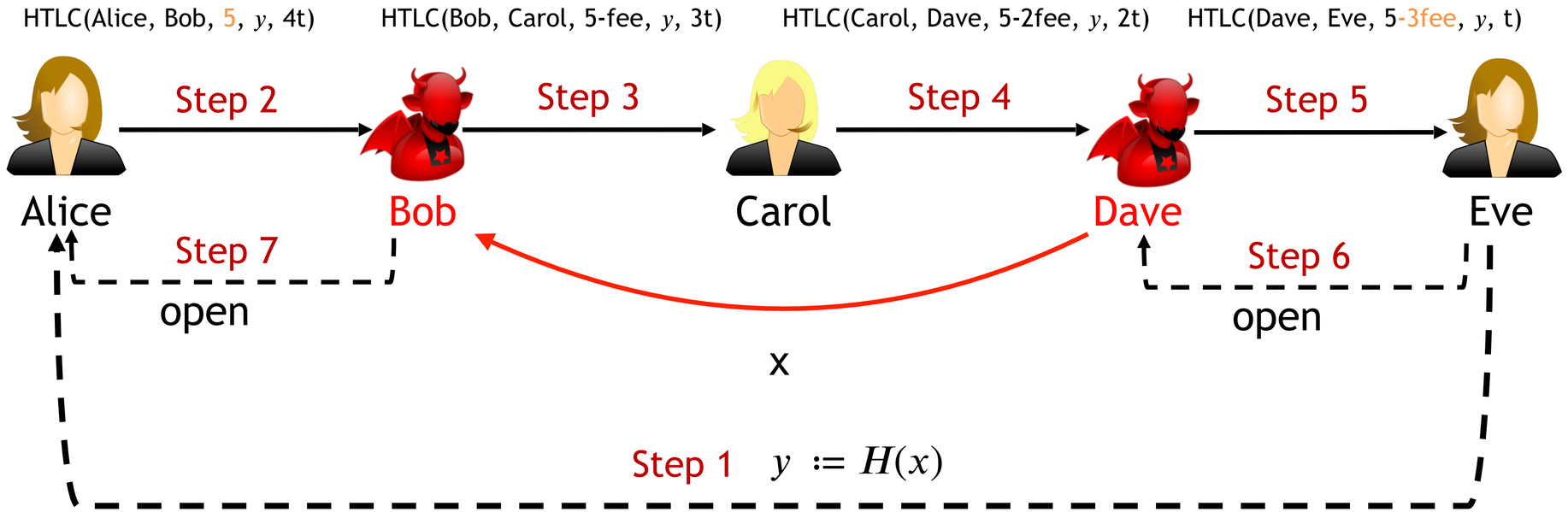
Properties & drawbacks of Lightning payments

- “Balance Security” ✔

Drawbacks:

- Staggered collateral lock time ✖
  - Decreases network throughput
- Takes two rounds ✖
- HTLC scripting requirements ✖
- Wormhole attack ✖

Wormhole attack [2]



- Bob and Dave steal the fee of honest Carol
- Carol's funds are still locked

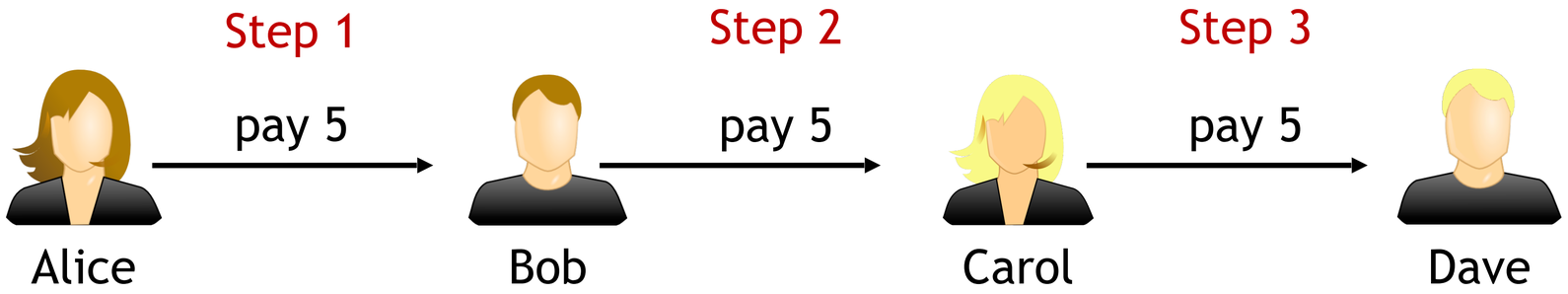
[2] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate and M. Maffei, "Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability," NDSS, *2019*

Goal

- Multi-hop payment with balance security
- One-round
- Constant collateral lock time
- Requires fewer scripting capabilities
- Resistance against wormhole attack

2) Blitz construction

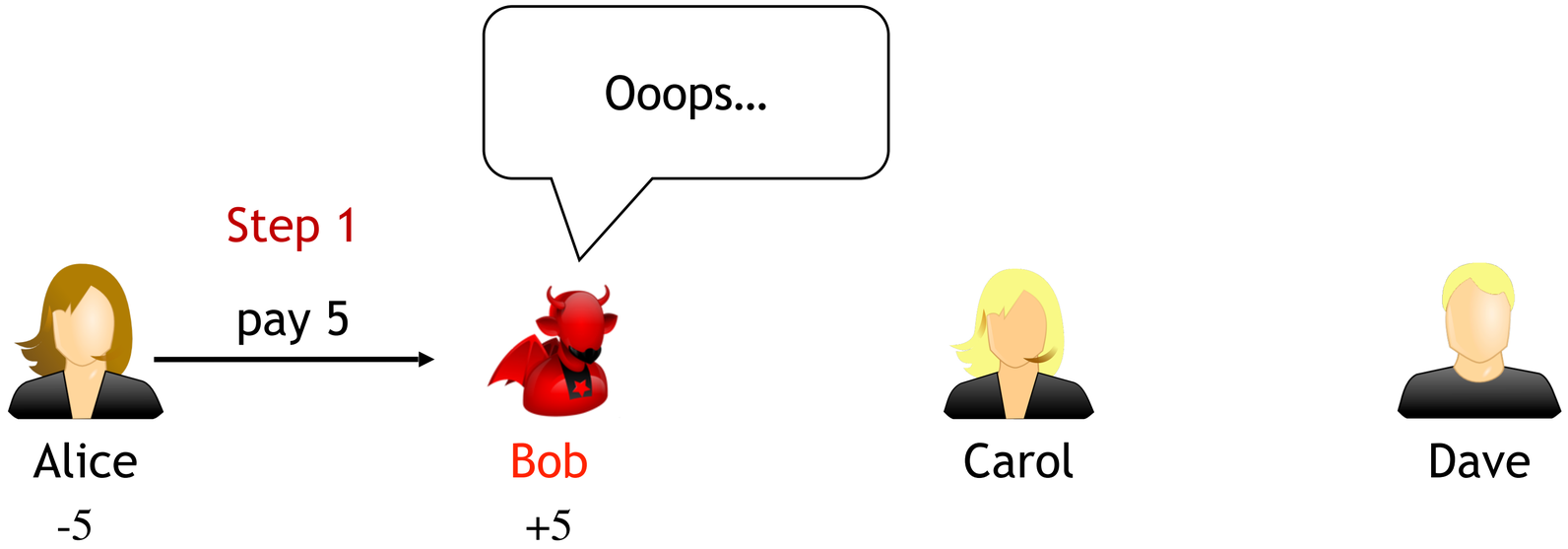
Multi-hop payments in one round: Attempt 1



=> Actually used in: Interledger [3]

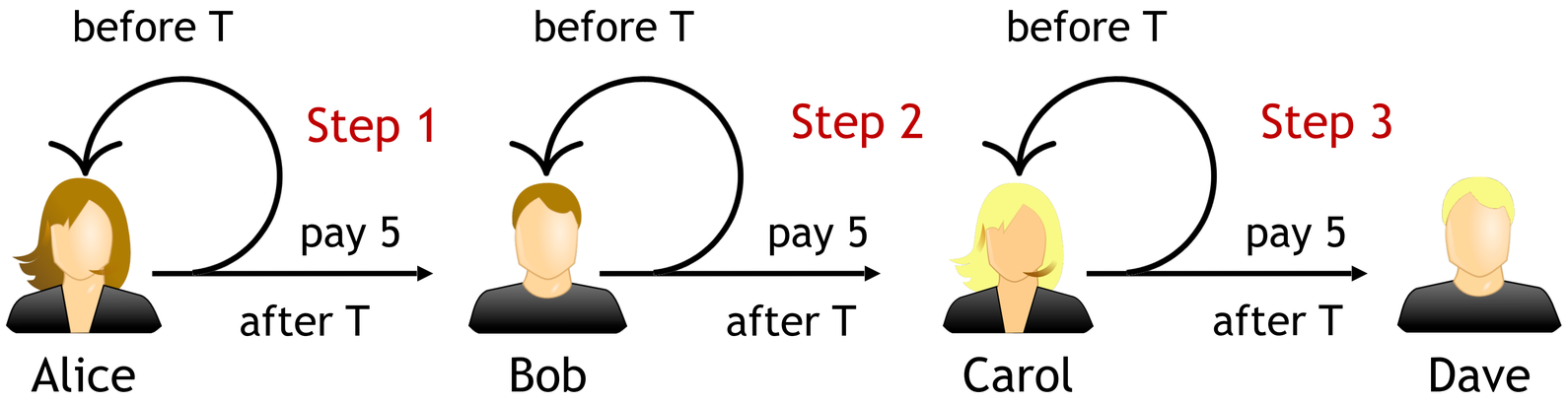
[3] S. Thomas and E. Schwartz, "A Protocol for Interledger Payments," *2015*

Multi-hop payments in one round: Attempt 1

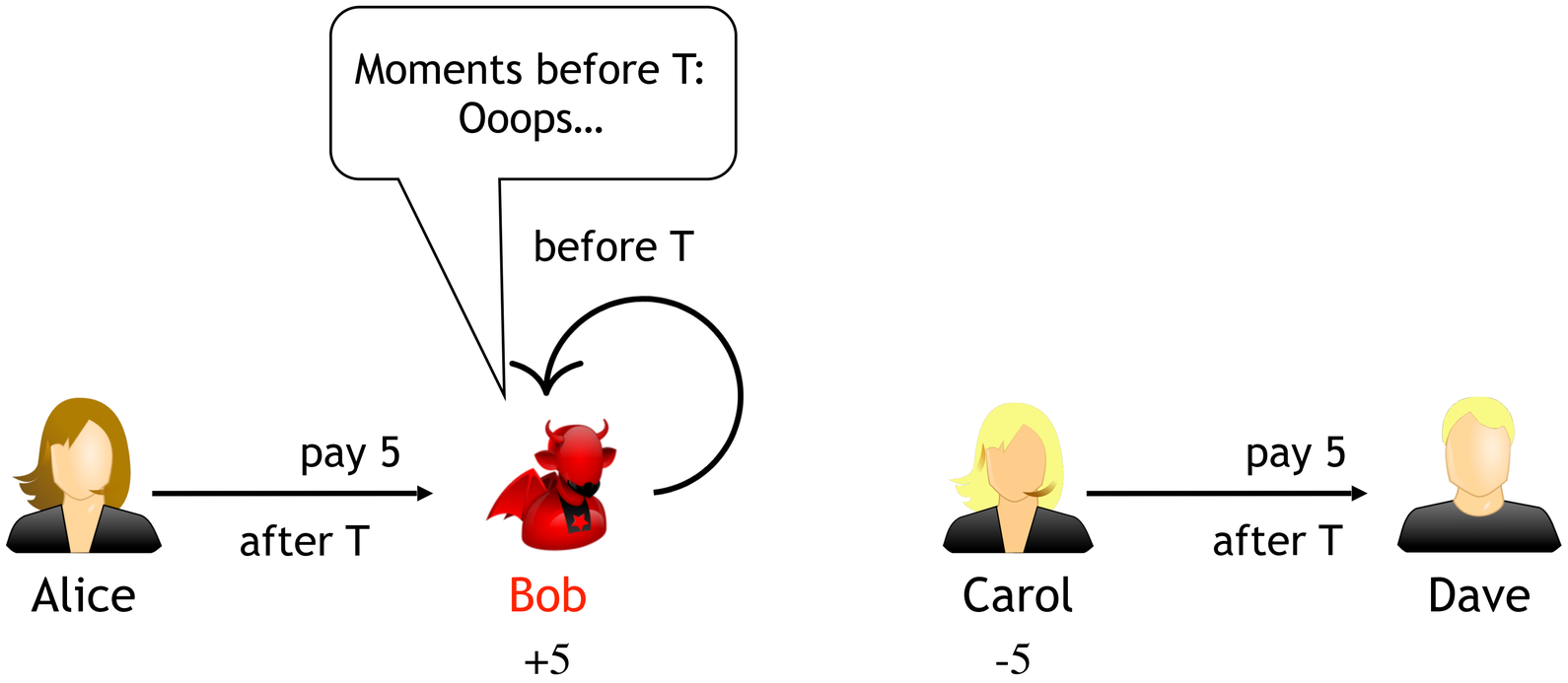


=> A malicious intermediary can stop the payment and effectively steal the 5 coins…

Multi-hop payments in one round: Attempt 2



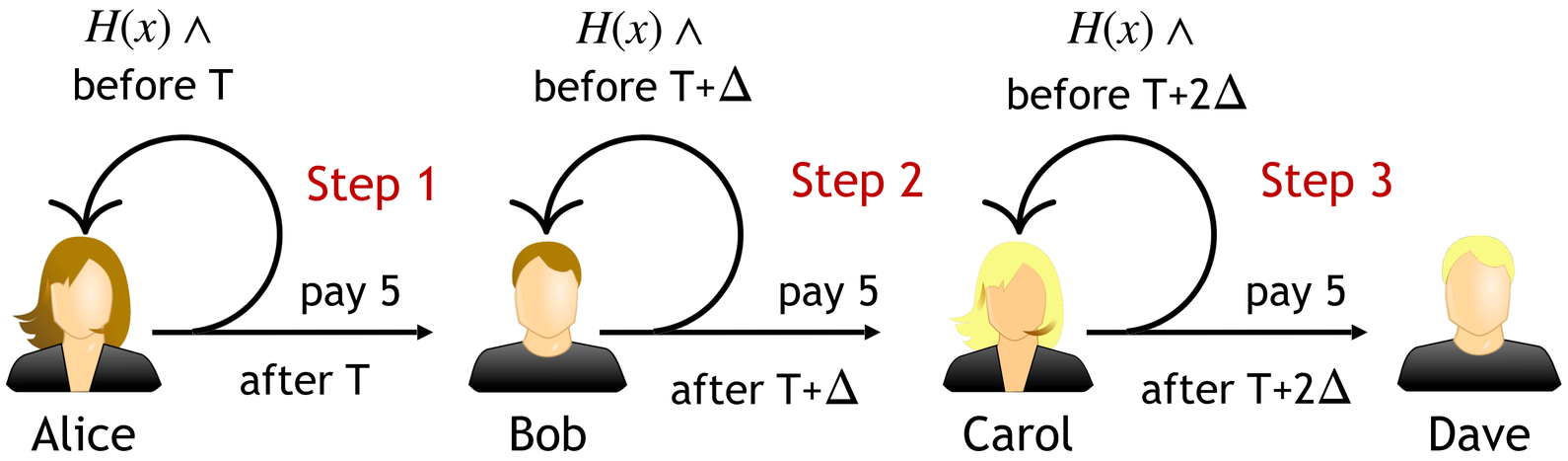
Multi-hop payments in one round: Attempt 2



- Bob refunds in the last moment
- Others won’t have time to react

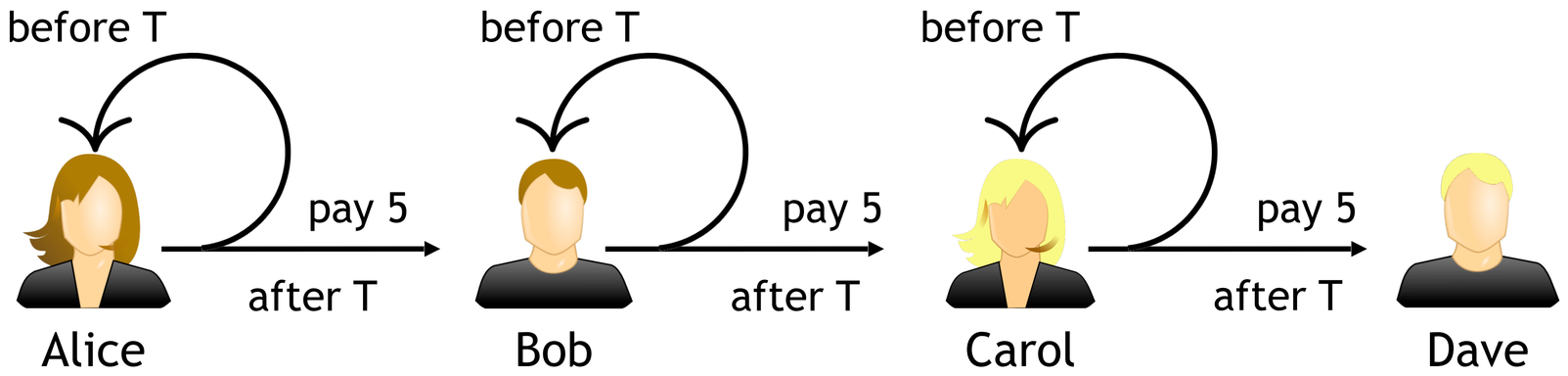
Multi-hop payments in one round: Attempt 3

x chosen by the sender

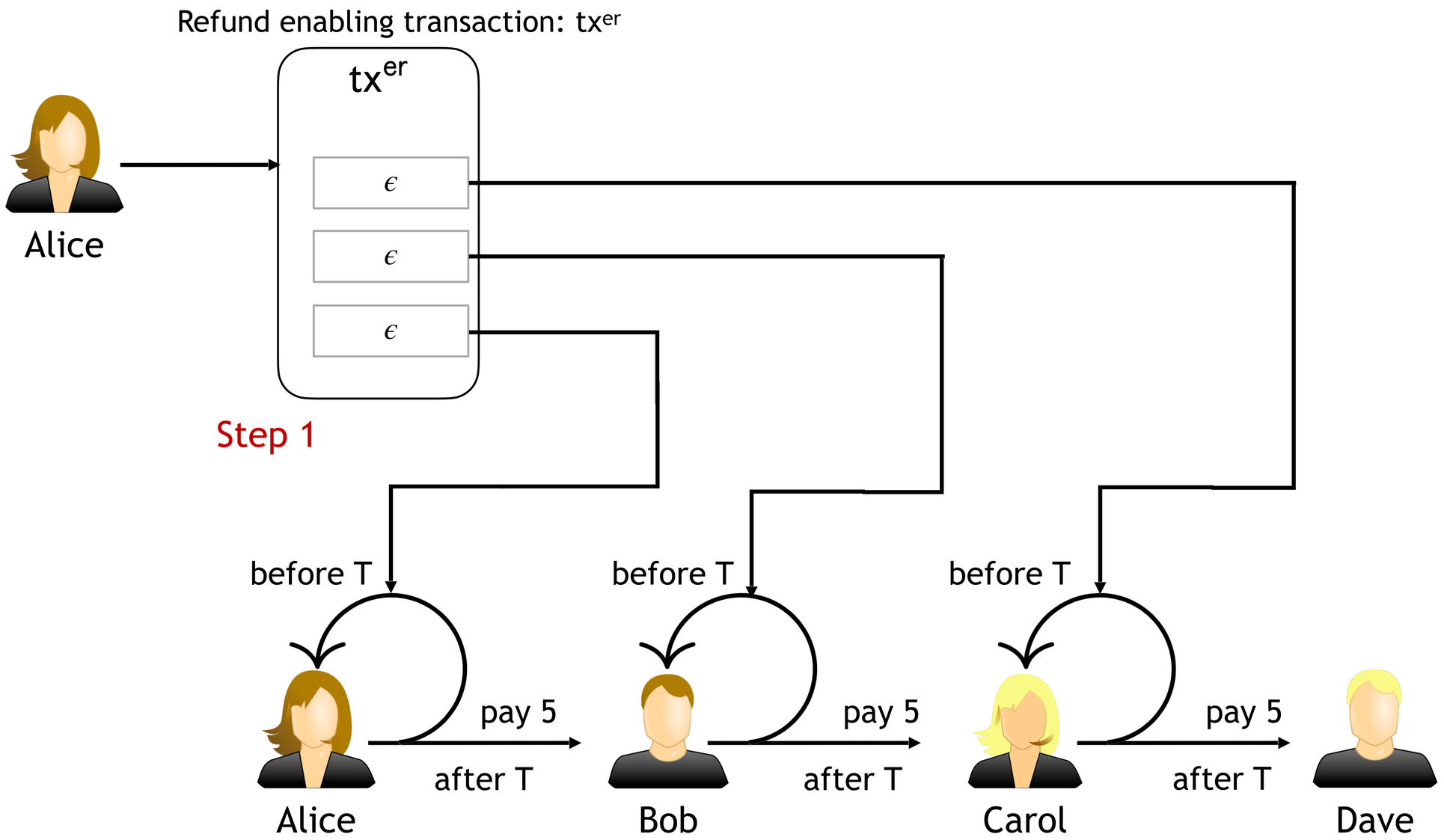


=> Similar to current Lightning multi-hop payments, has same scripting requirements as Lightning, collateral time grows linearly…

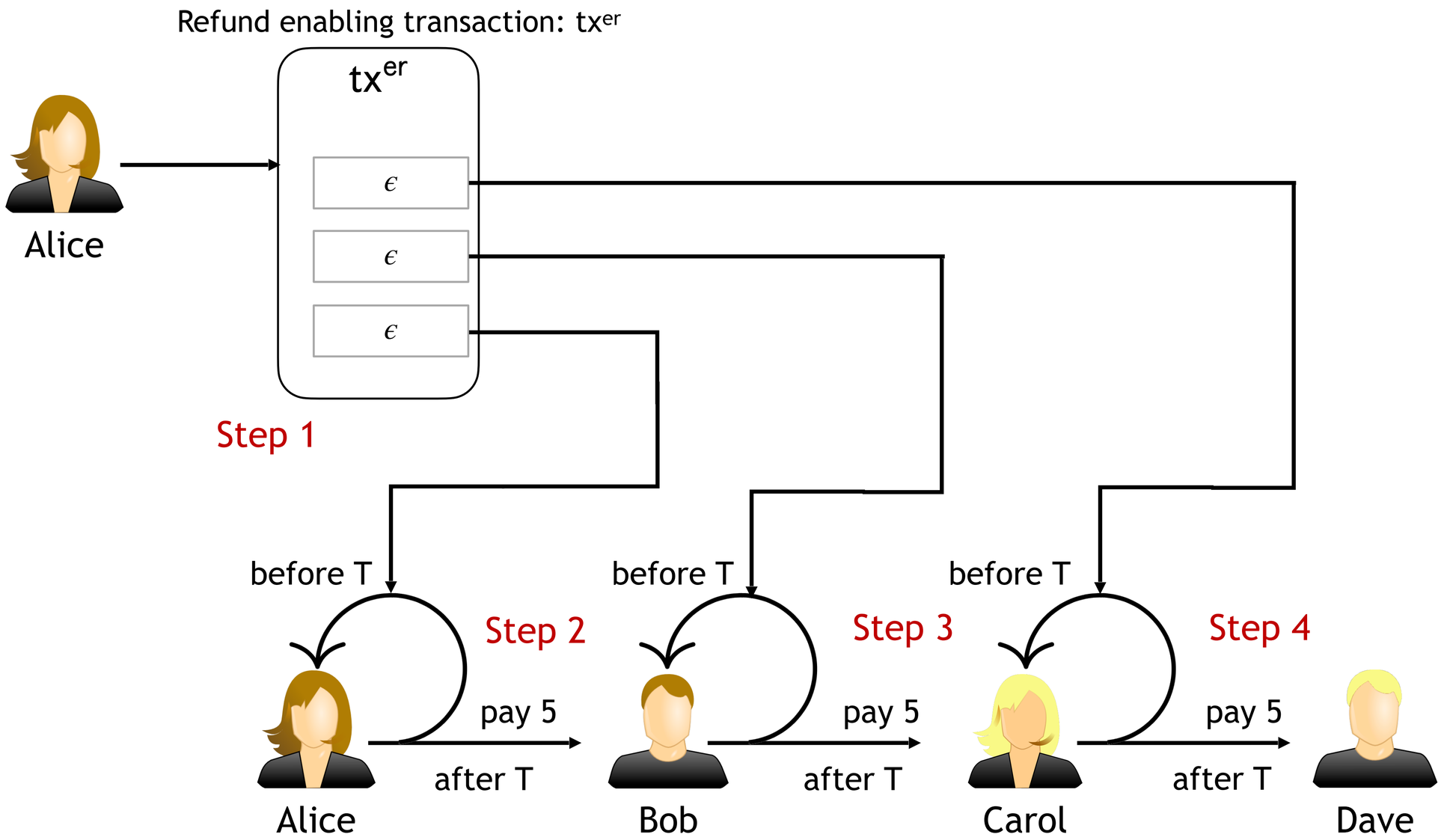
Blitz payments



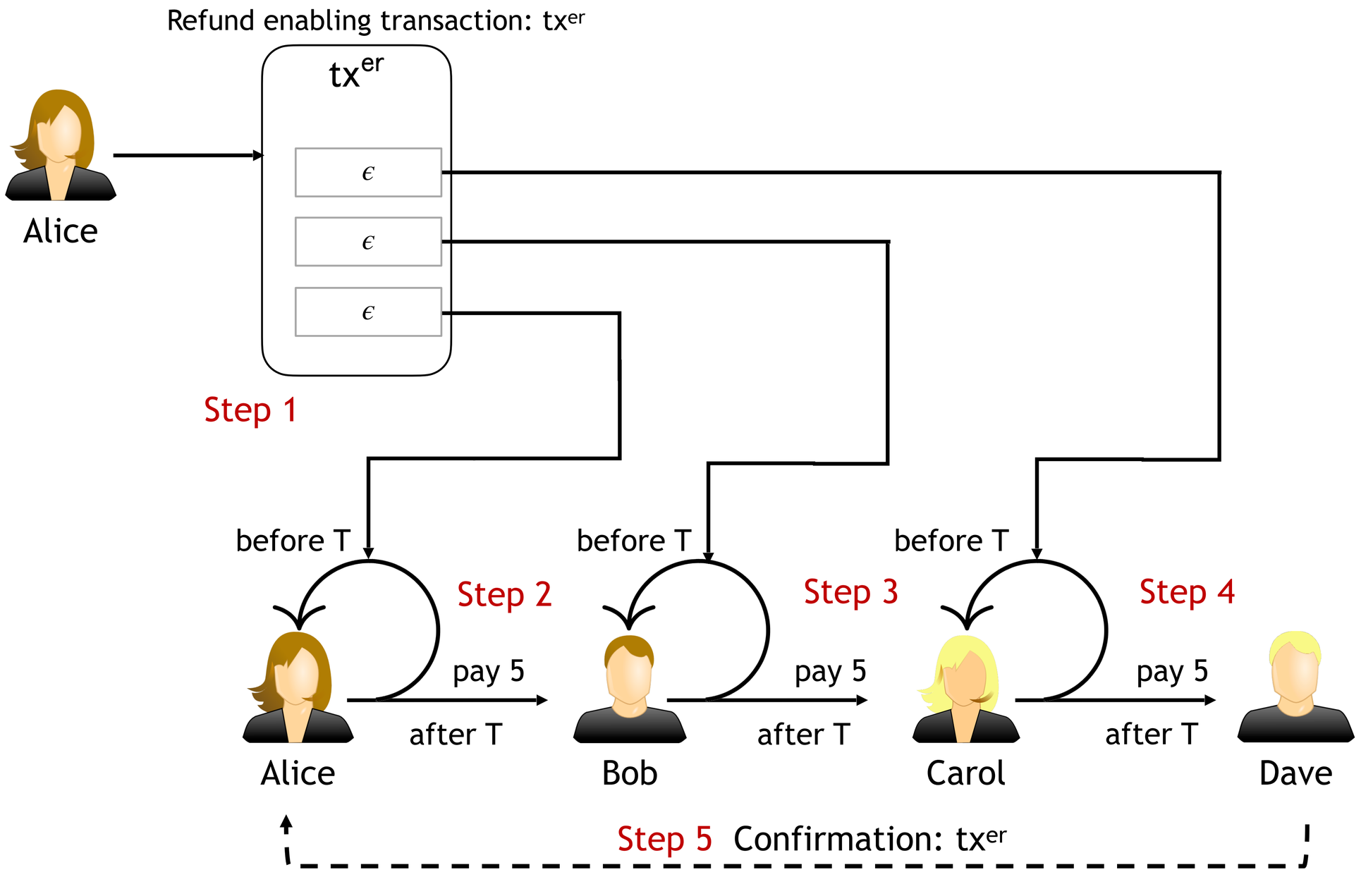
Blitz payments



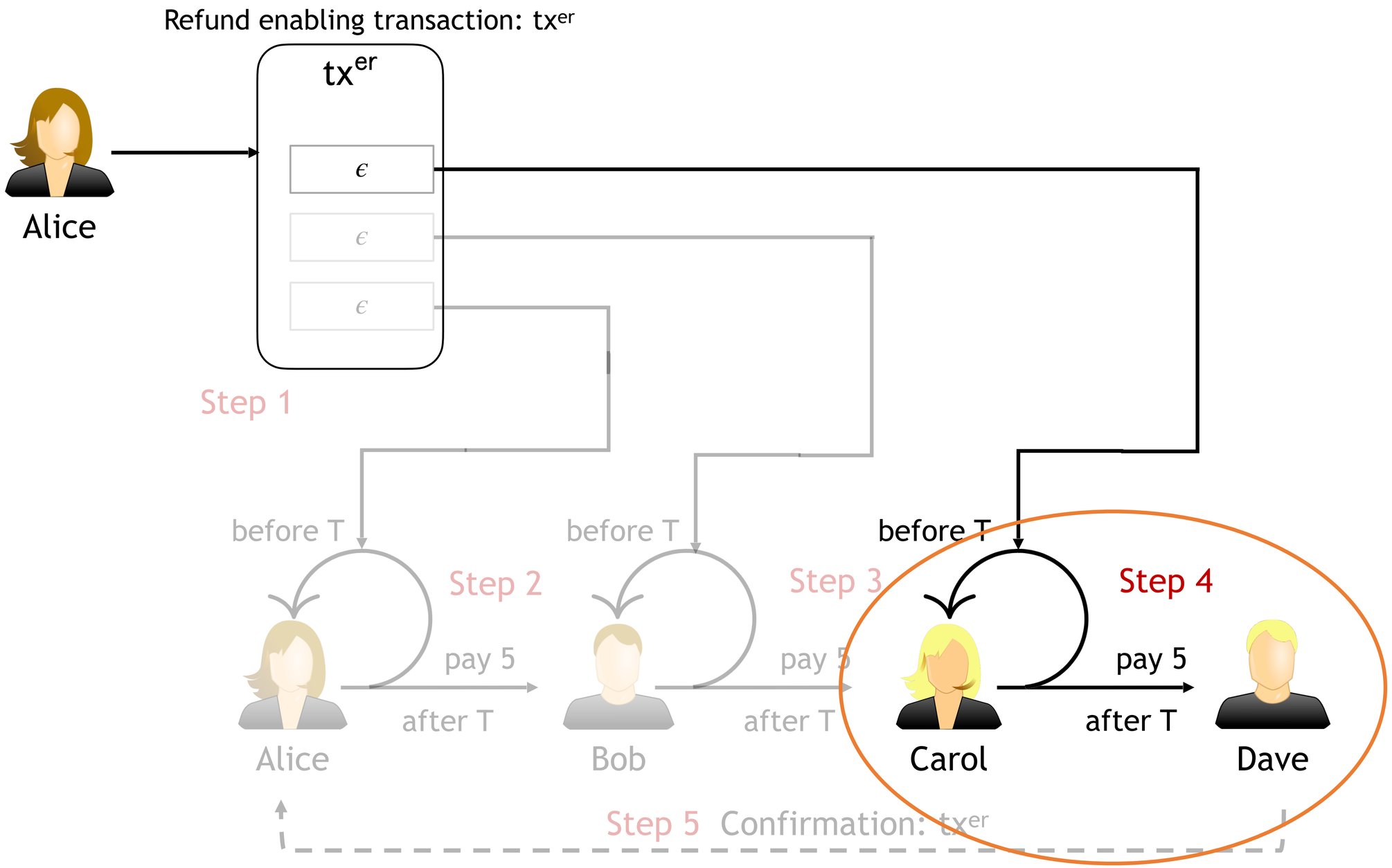
Blitz payments



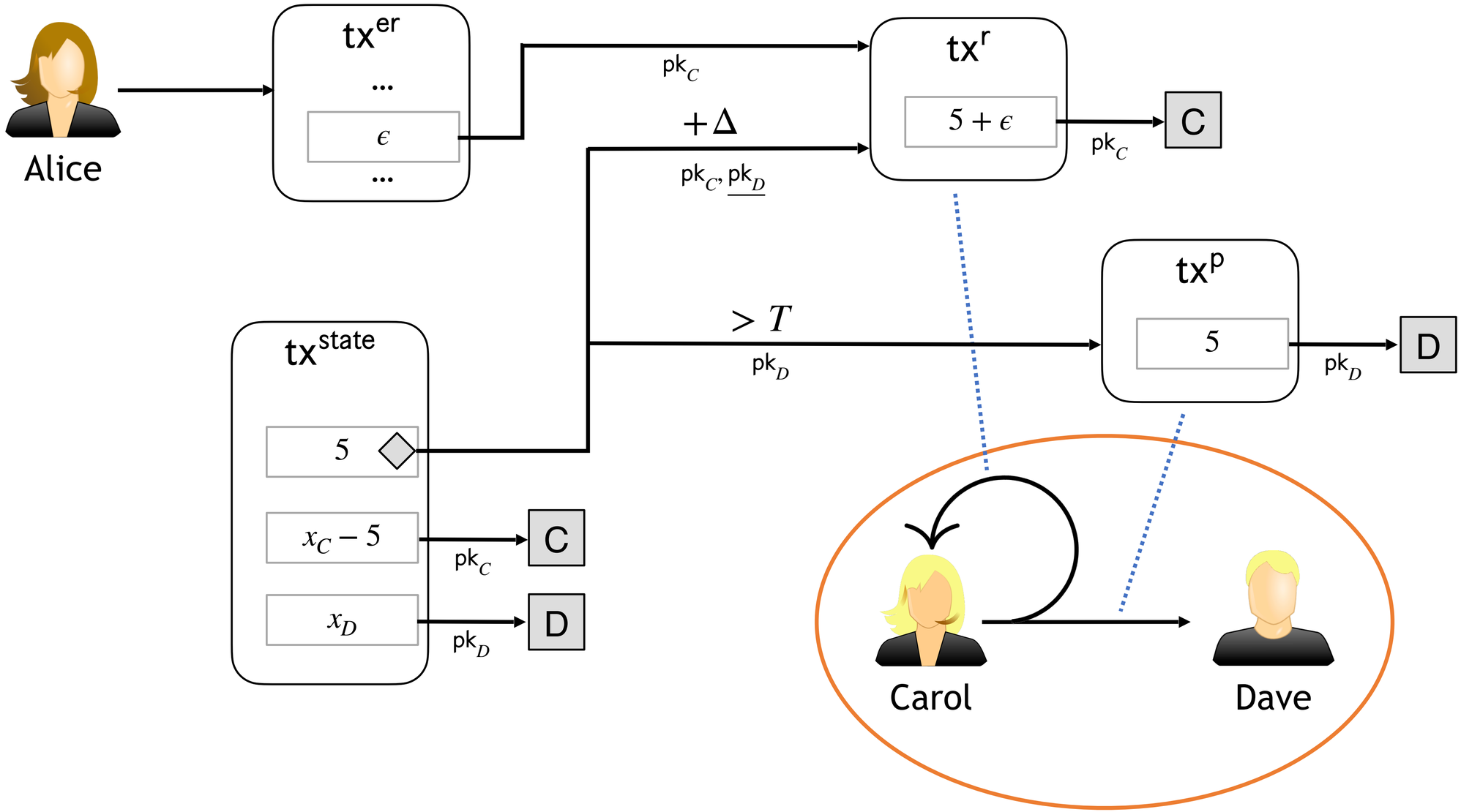
Blitz payments



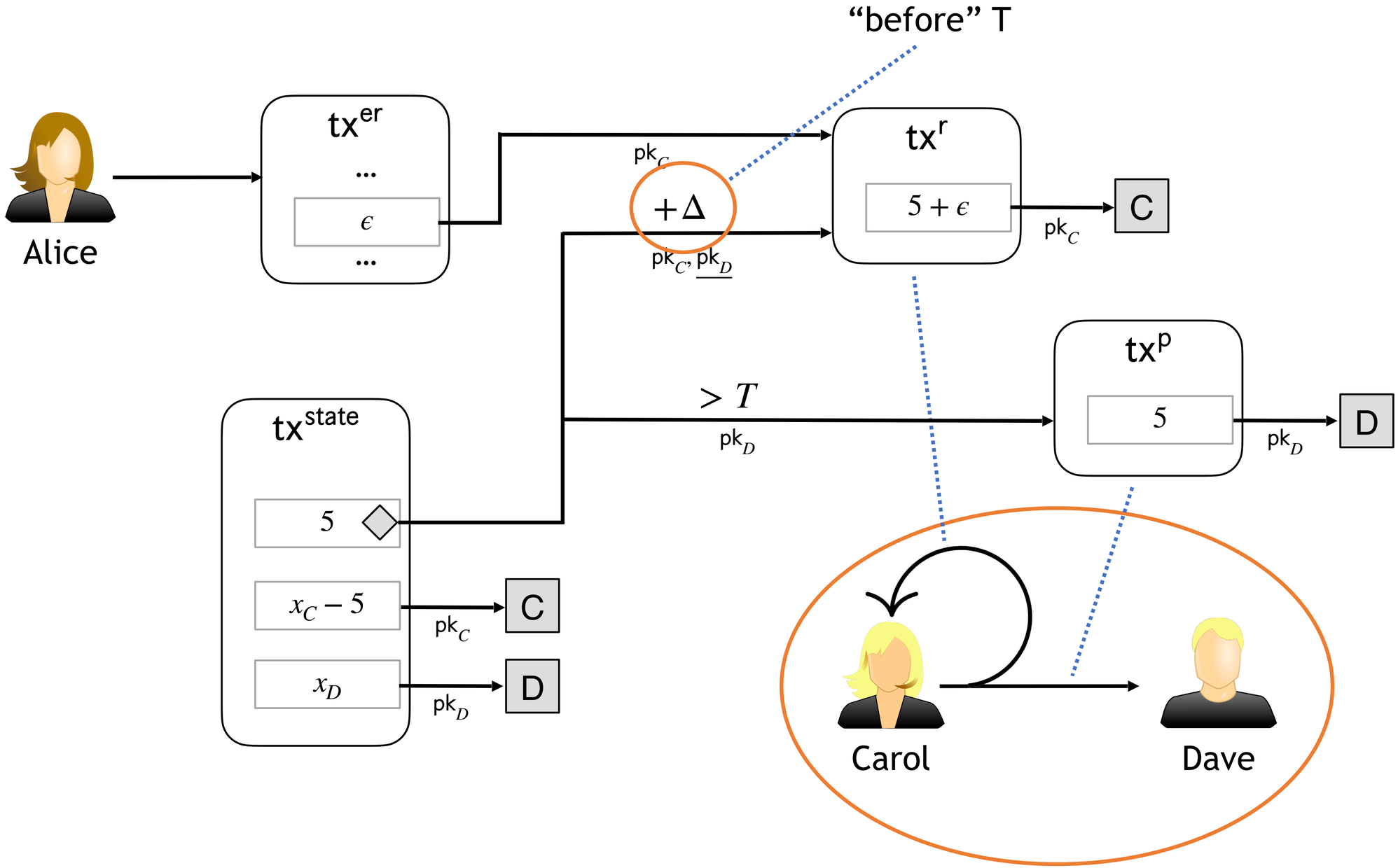
Blitz payments



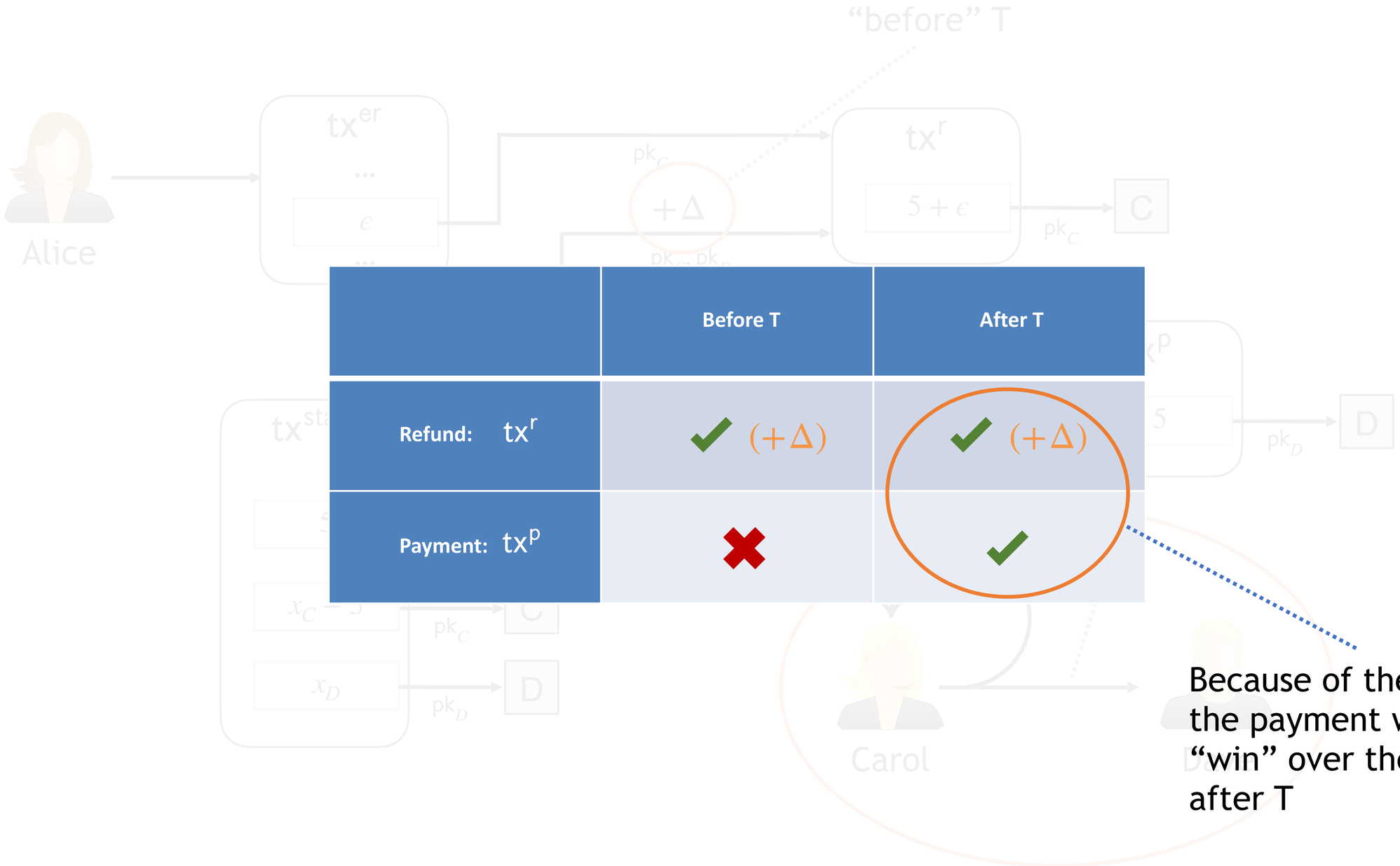
Blitz payments: details



Blitz payments: details



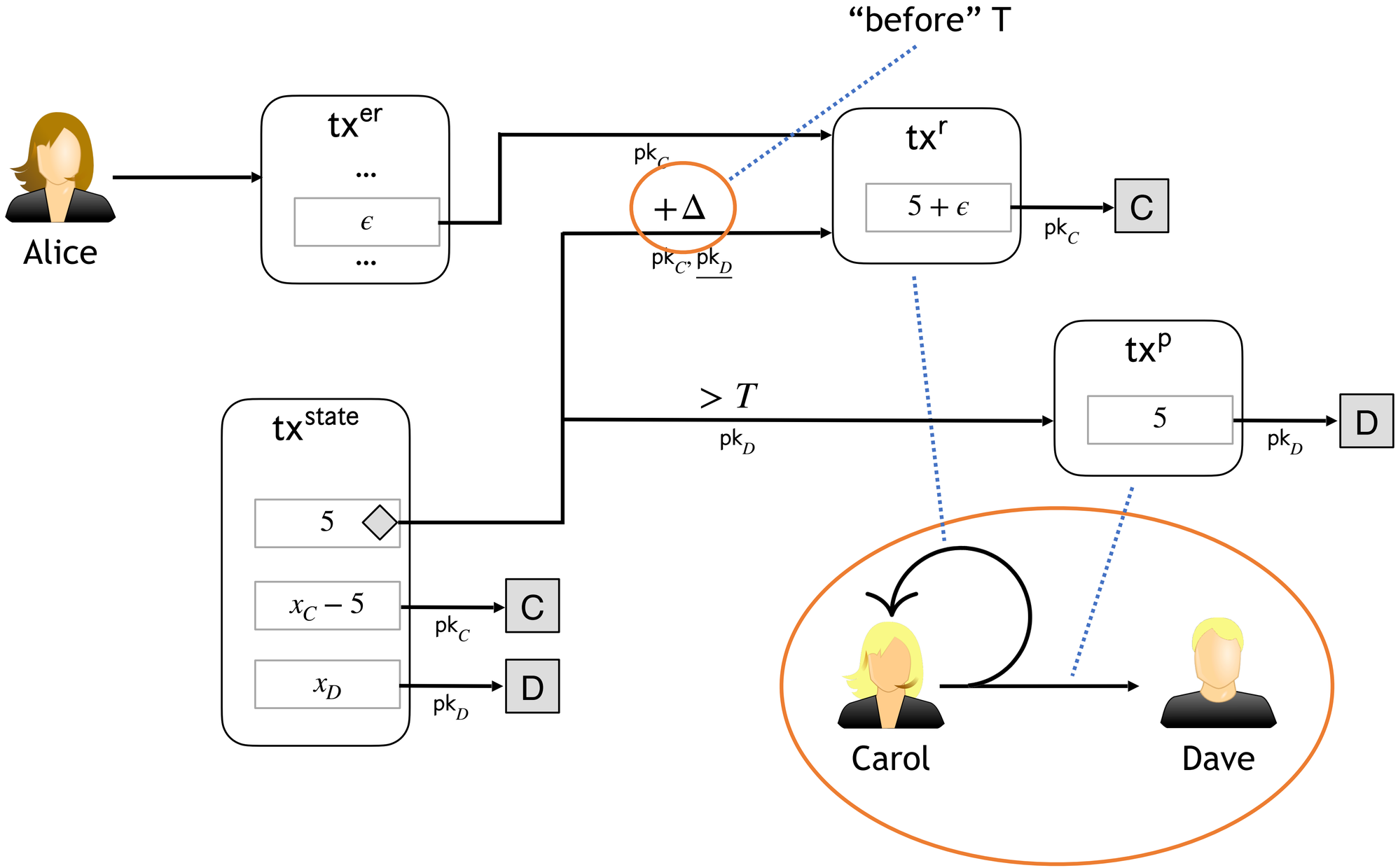
Before/After T



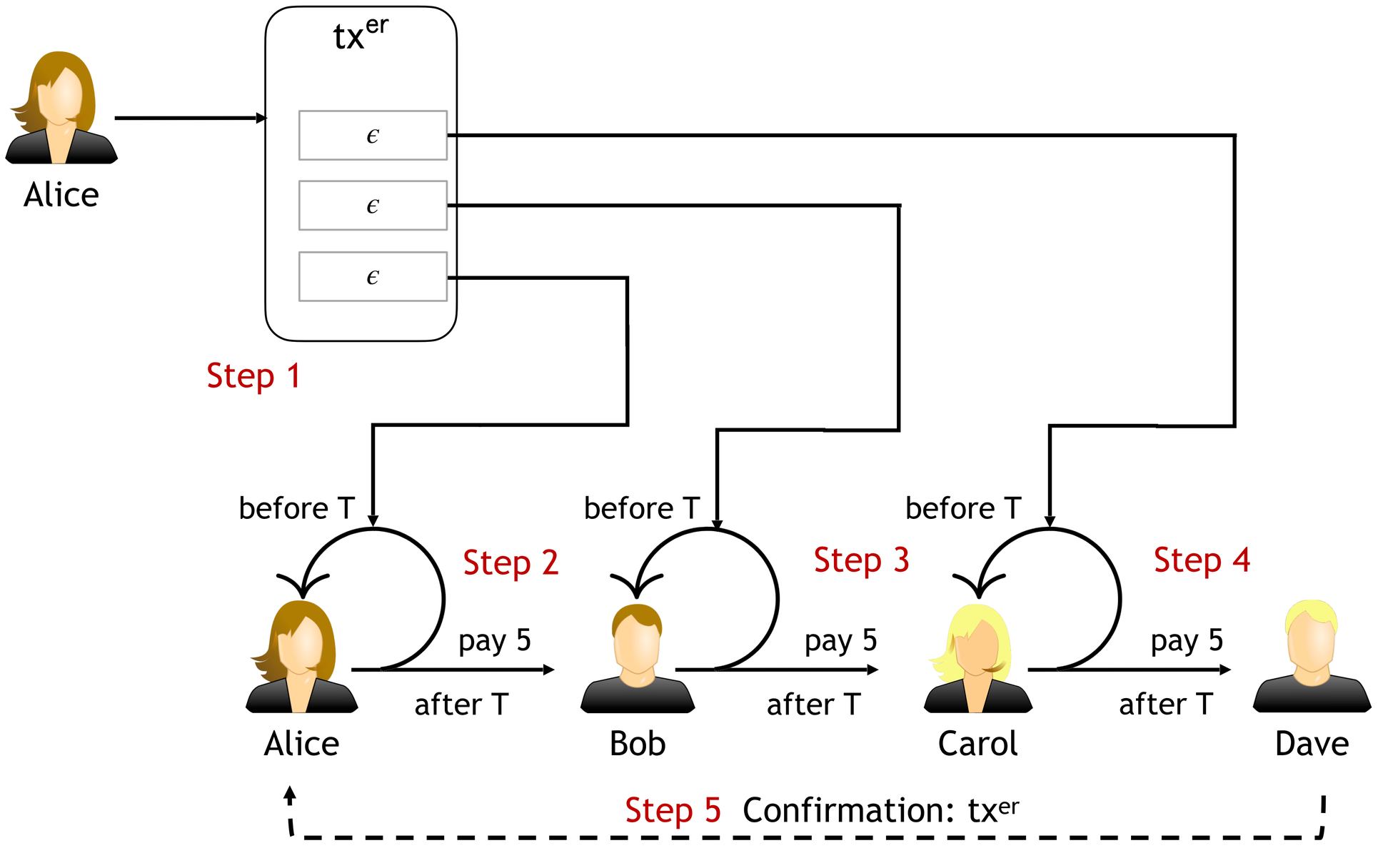
	Before T	After T
Refund: tx^r	✓ $(+\Delta)$	✓ $(+\Delta)$
Payment: tx^p	✗	✓

Because of the delay, the payment will always “win” over the refund after T

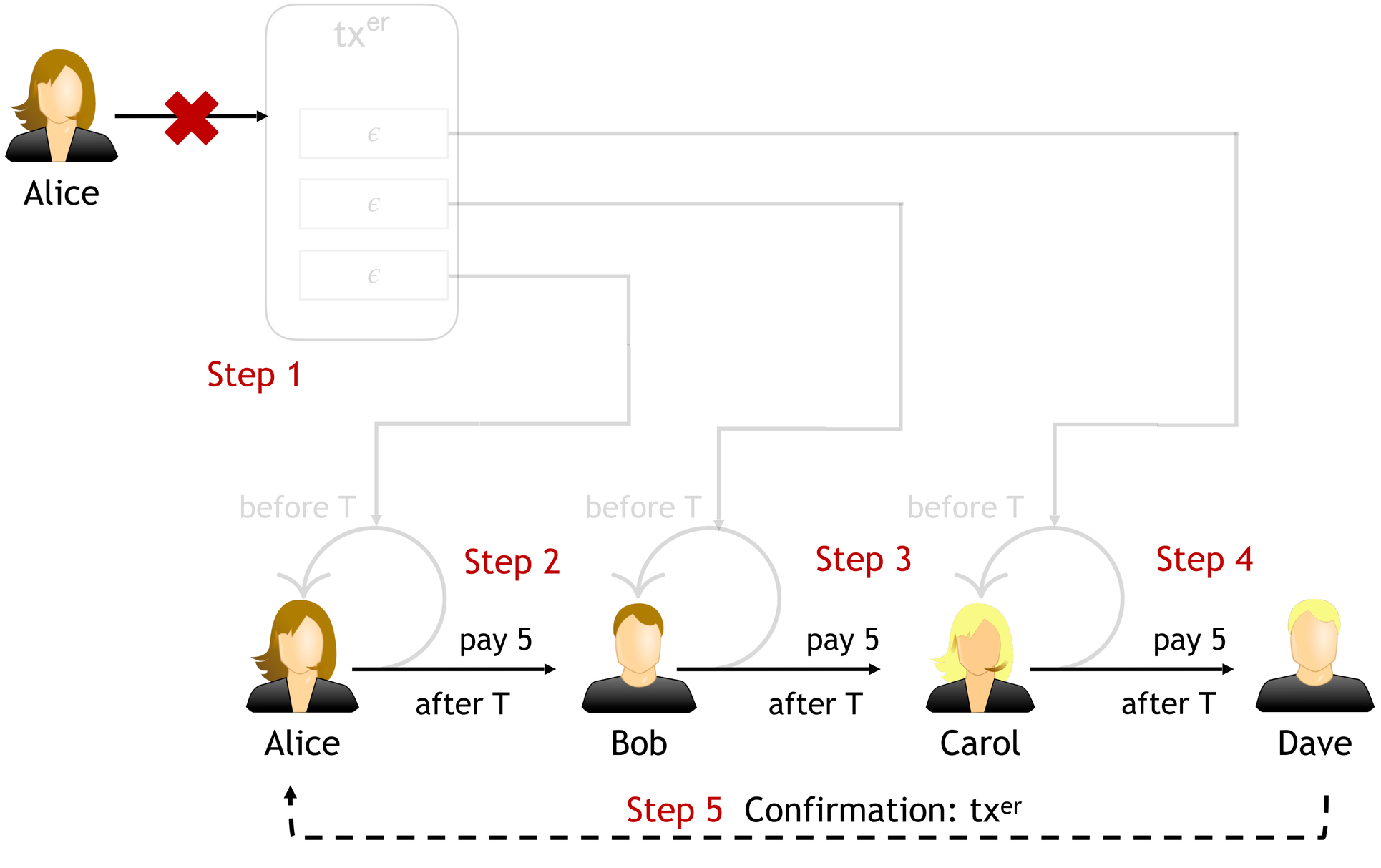
Before/After T



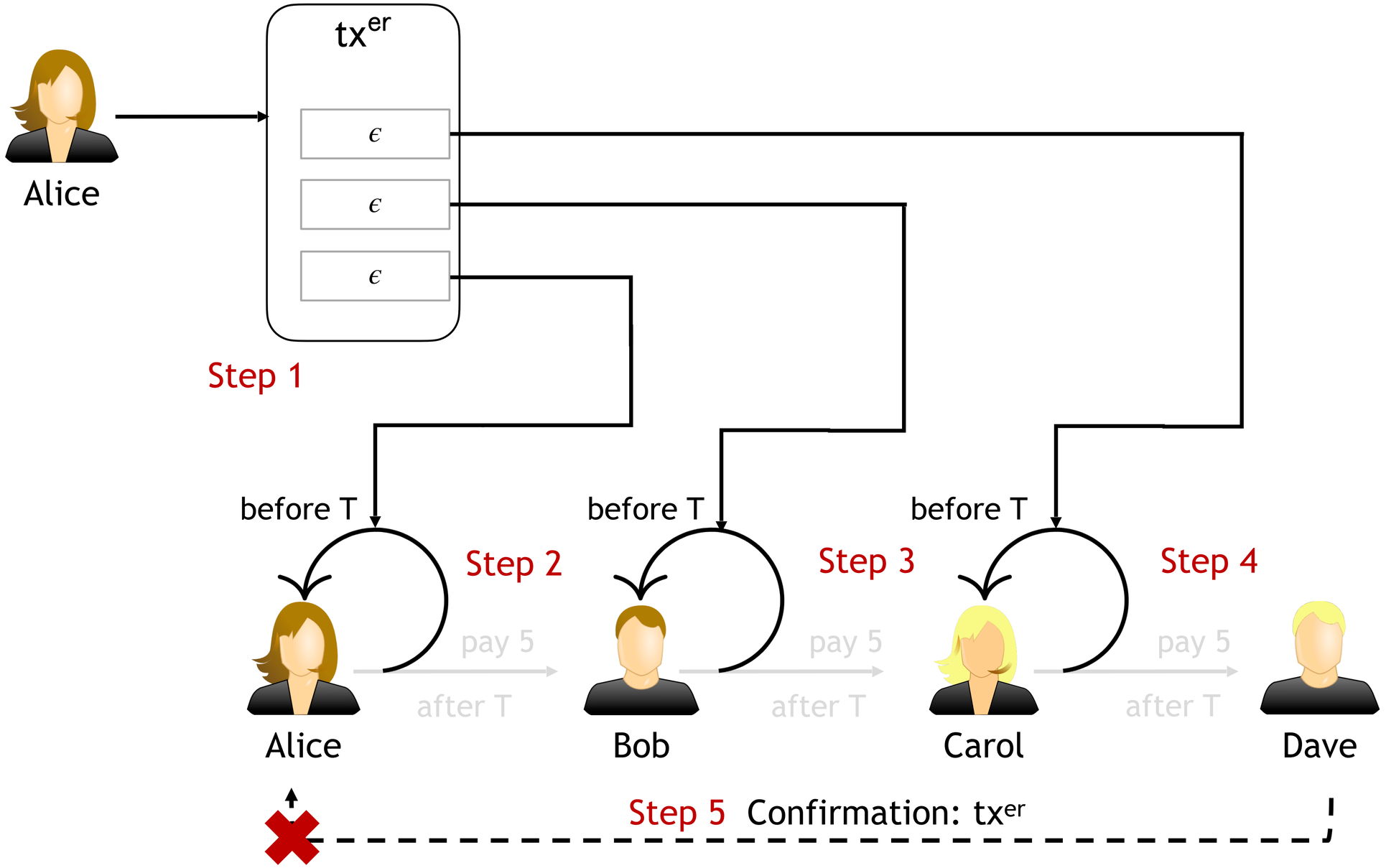
Blitz payments



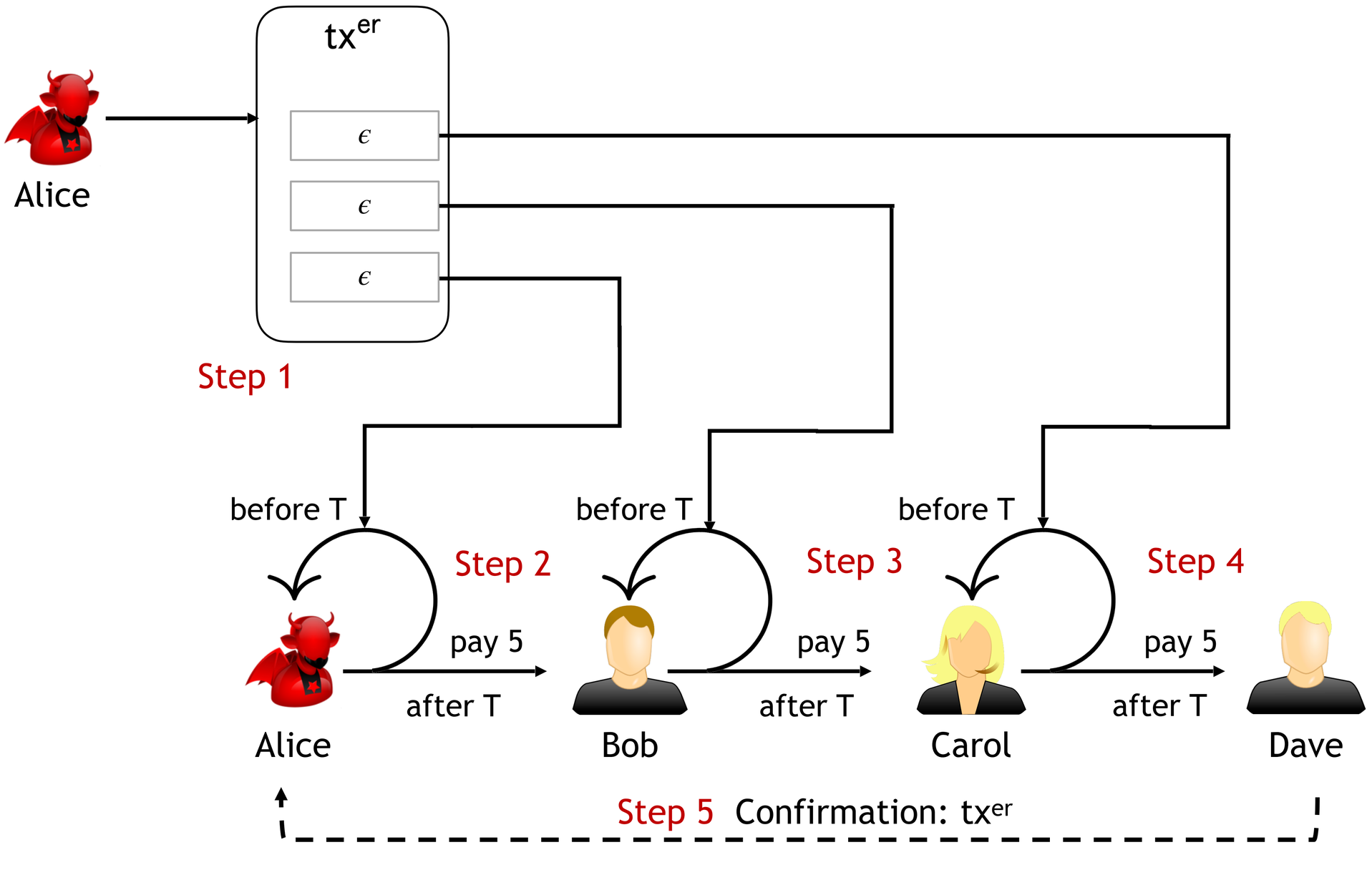
Successful payment



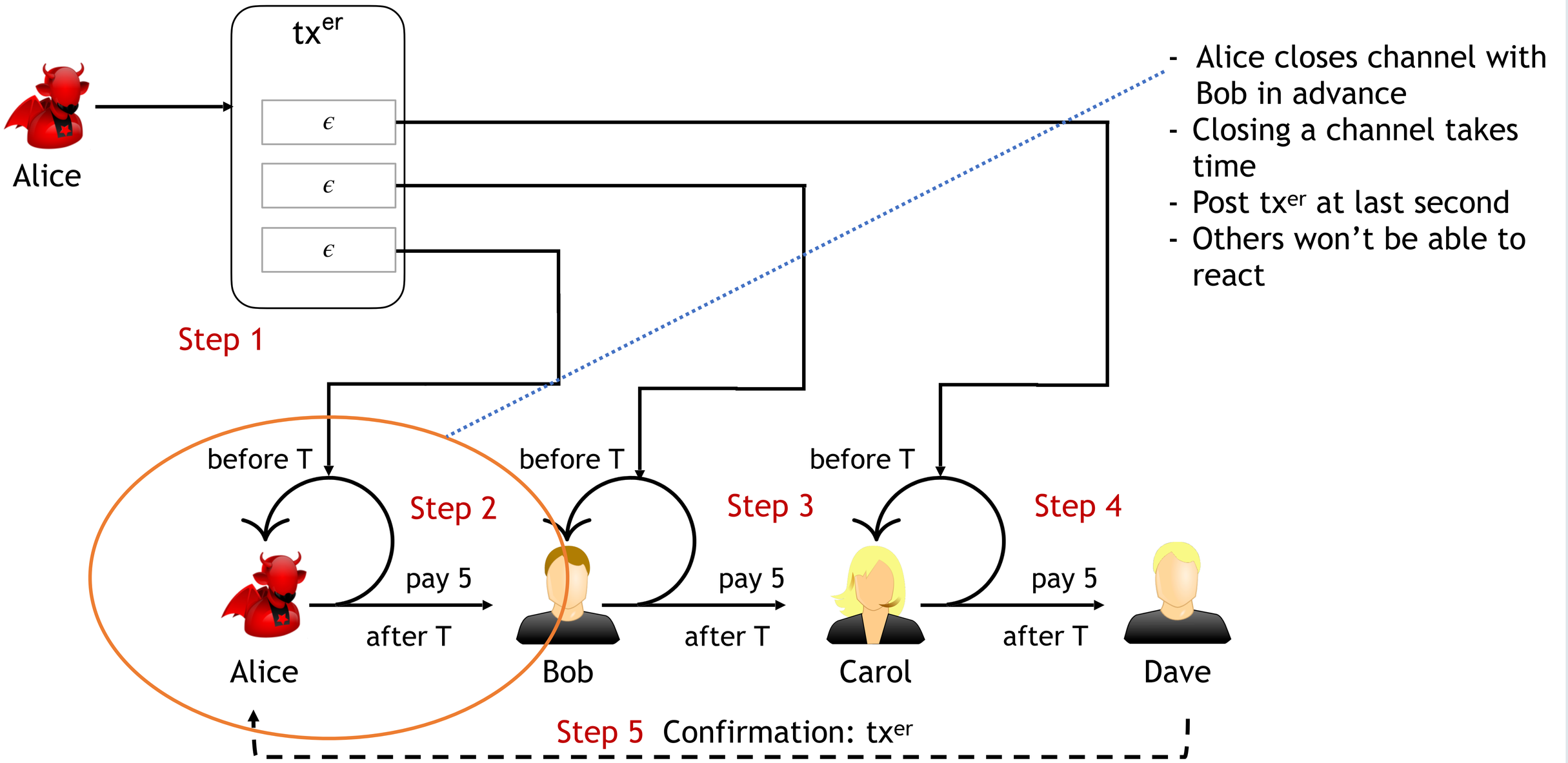
Refund


Alice
tx^er
ϵ
ϵ
ϵ
Step 1
before T
before T
before T
Step 2
Step 3
Step 4
pay 5
pay 5
pay 5
after T
after T
after T
Alice
Bob
Carol
Dave
Step 5 Confirmation: tx^er

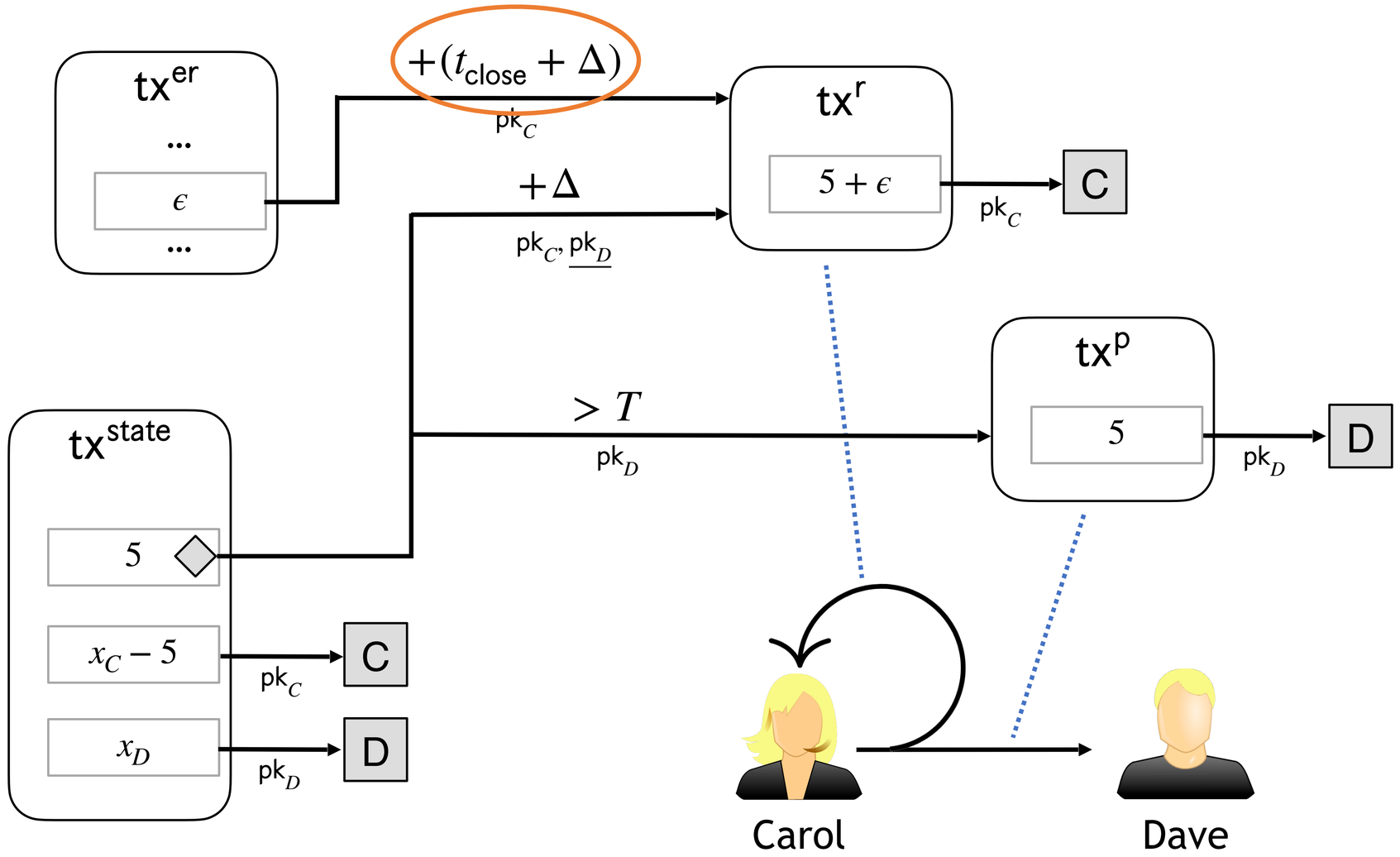
Sender advantage



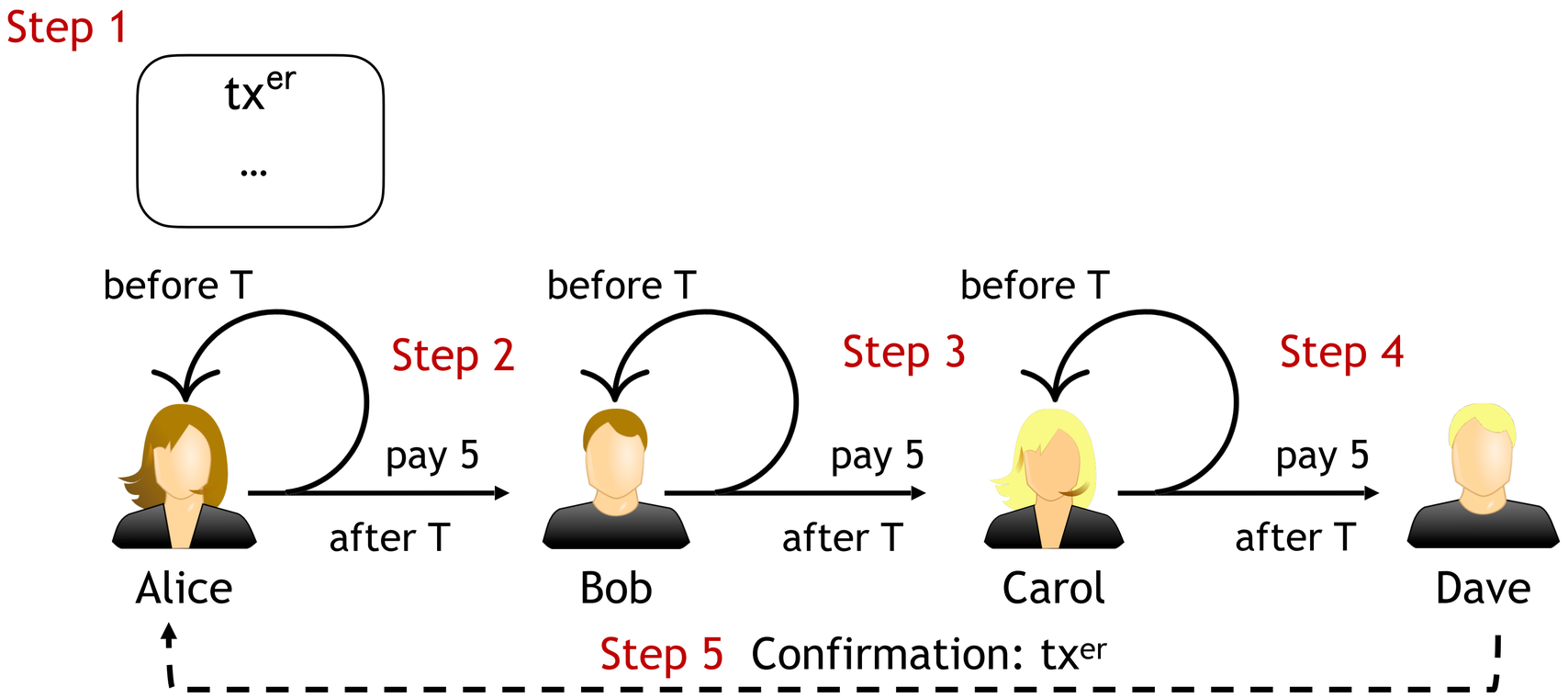
Sender advantage



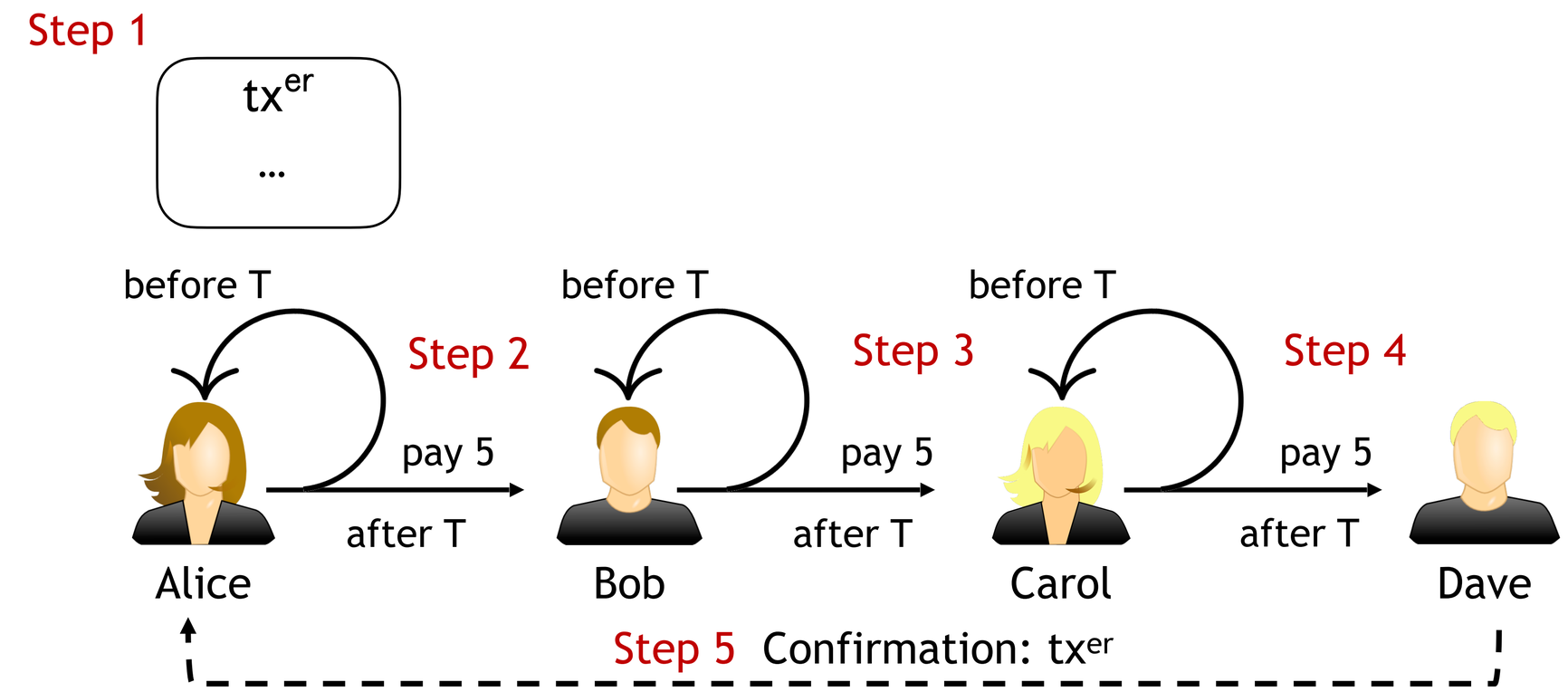
Remove sender advantage

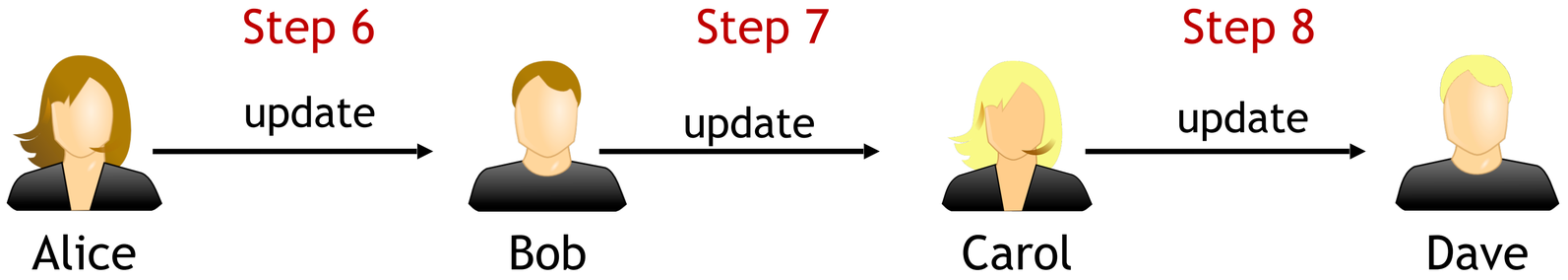


“Instant” payments in optimistic case?

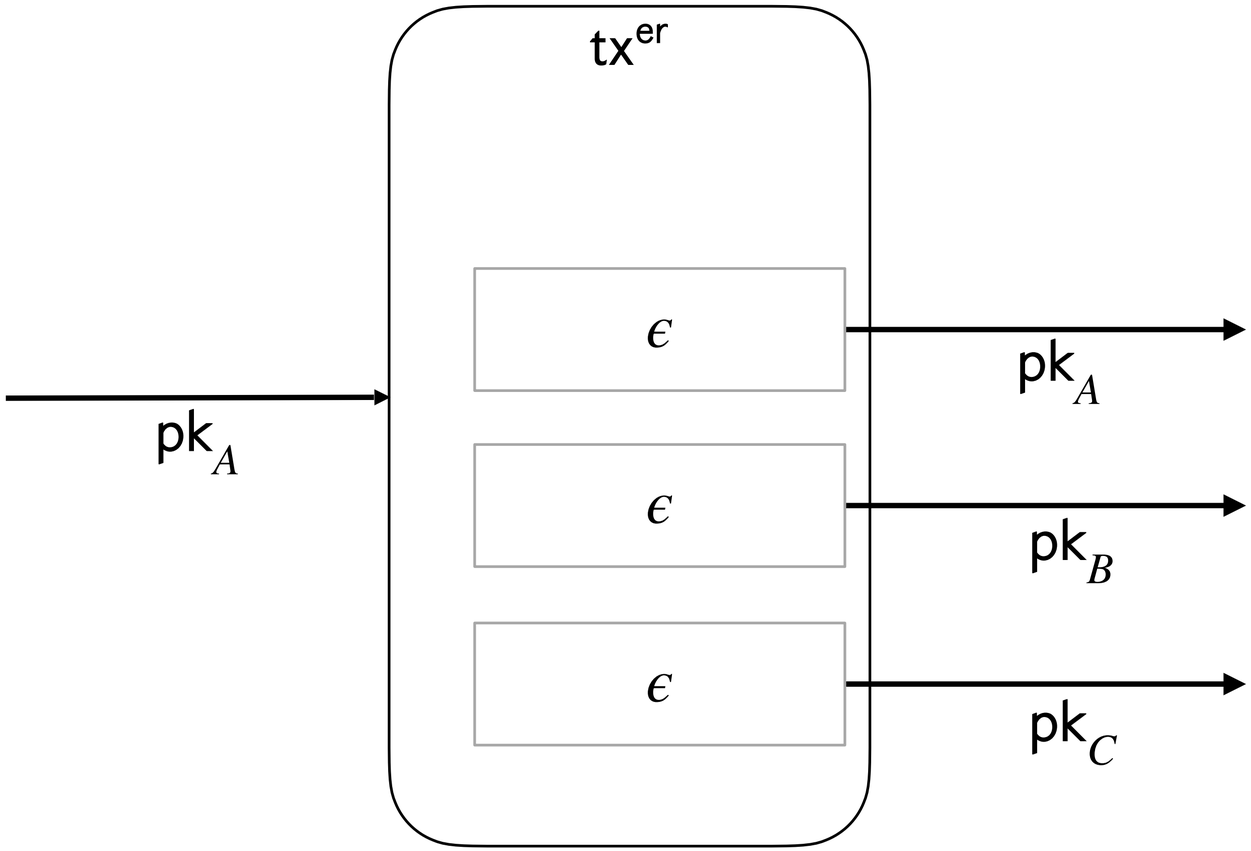


Fast track


Step 1
tx^er
...
before T
before T
before T
Step 2
Step 3
Step 4
pay 5
pay 5
pay 5
after T
after T
after T
Alice
Bob
Carol
Dave
Step 5 Confirmation: tx^er


Step 6
Step 7
Step 8
update
update
update
Alice
Bob
Carol
Dave

Privacy leakage through tx^{er}?



Privacy leakage through tx^{er}?

- Outputs of tx^{er} fresh addresses
- => Stealth addresses [4]
- Unlinkable to the users
- Input of tx^{er} unlinkable Alice
- Onion routing
- Users learn only about their direct neighbors
- Like LN: No relationship anonymity
  - Link payments by tx^{er} (Blitz) or hash value (LN)
- Like LN: No privacy for on-chain disputes

[4] N. Van Saberhagen, “CryptoNote v 2.0,” https://cryptonote.org/whitepaper.pdf, *2013*

Take Home

- Comparison to Lightning:

	Lightning	Blitz
Number of rounds	2	1 (2 for fast track)
Collateral lock time	Staggered (linear) $\Theta(n \cdot \zeta)$[1]	Constant $\Theta(\zeta)$
Wormhole attack	Susceptible	Secure
Scripting capabilities	Signatures, timelocks, hashlocks[2]	Signatures, timelocks

[1] In Lightning, a value of $\zeta = 144$ blocks (ca. 1 day) is currently used.

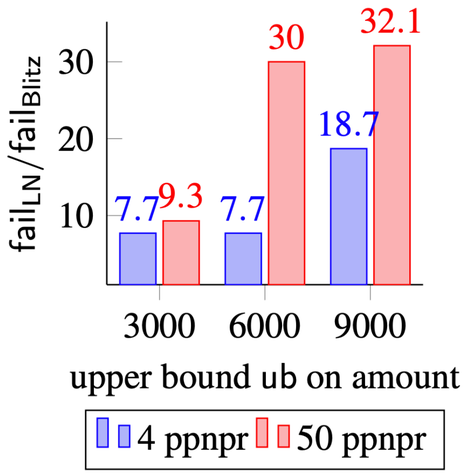
[2] Using constructions such as scriptless scripts, one could get rid of hashlocks.

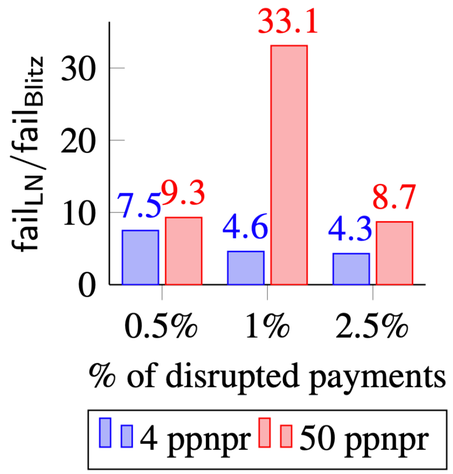
- Simulation showing practical advantage of constant collateral
- Formally modelled in UC framework, see paper.

Extra slides

Simulation

- Snapshot of Lightning Network, random payments in two phases
- Phase 1: Payments are disrupted, collateral locked up
  - In LN (staggered) and in Blitz (constant)
- Phase 2: Payments go through honest users in 3 rounds (days)
- Baseline: How many payments fail in Phase 2 without Phase 1 (e.g., not enough channel capacity)
- Failed payments in LN - baseline: $fail_{LN}$
- Failed payments in Blitz - baseline: $fail_{Blitz}$
- In Figure below: $fail_{LN}/fail_{Blitz}$





Parameters:

- ppnpr: payments per node and round
- Payment values sampled from [1000,ub]
- % of disrupted payments: how many payments get disrupted in Phase 1, compared to total number

Simulation code and results at: https://github.com/blitz-payments/simulation

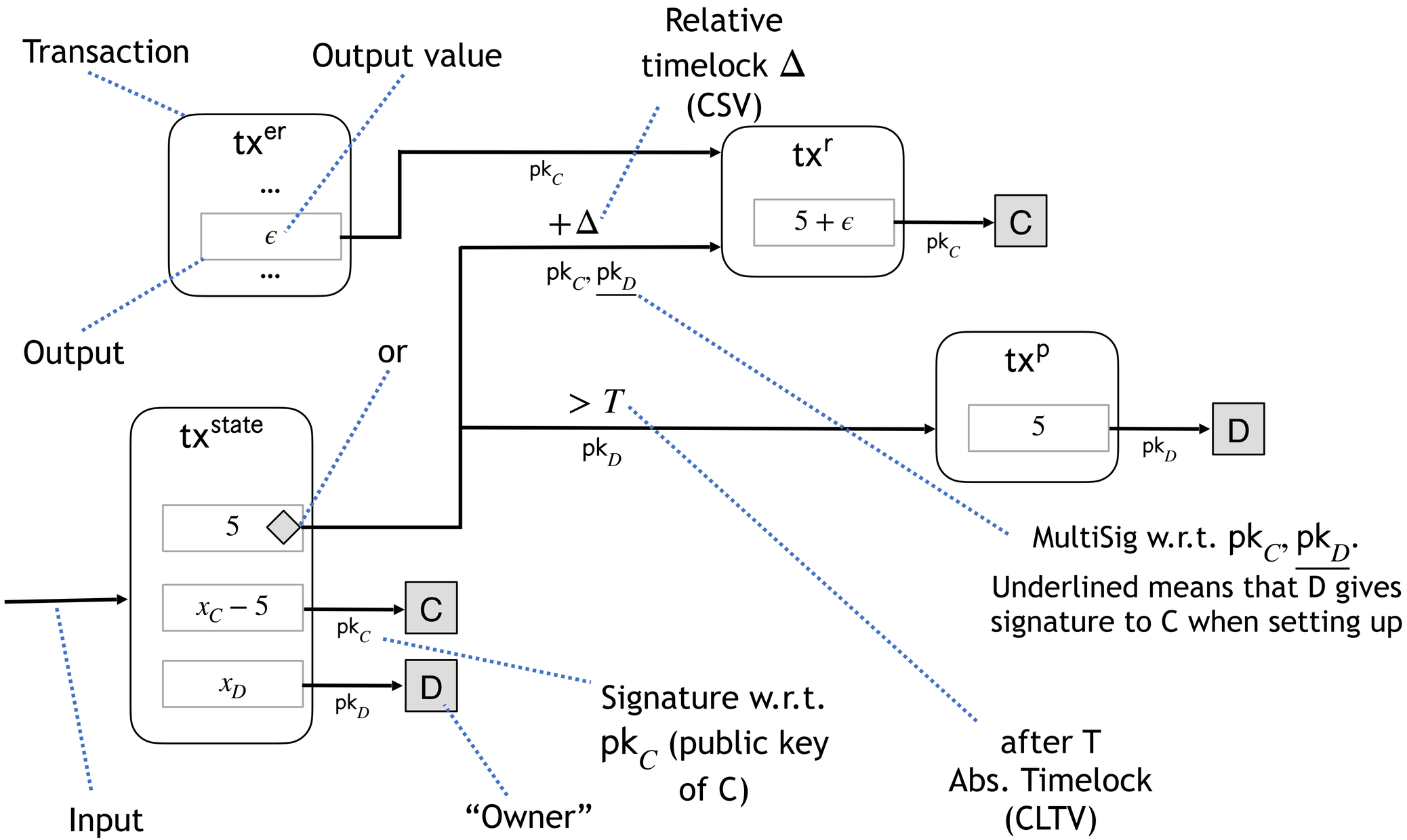
Other Evaluation

- Measure communication overhead in bytes
- Note that Blitz contract is 88 bytes compared to HTLC, which is 119 bytes
- State transactions could hold around 26% more Blitz in-flight payments than Lightning payments

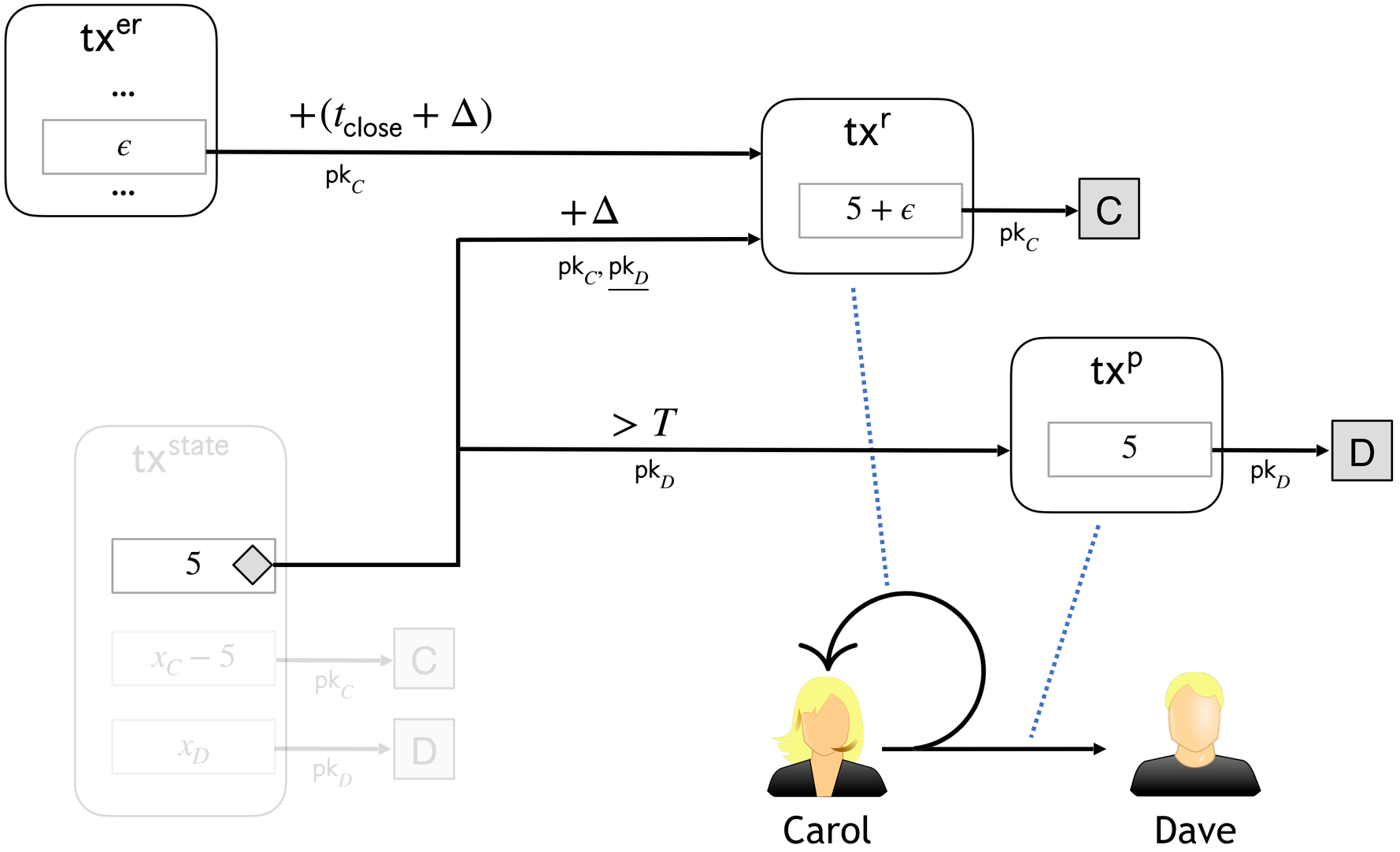
Cases	LN		Blitz	
	# txs	size	# txs	size
Pay (pessimistic)	1	192	1	158
Refund (pessimistic) per channel	1	158	1	307
Additional pess. refund cost for sender	0	0	1	$157 + 34 \cdot n$
Cost of p in-flight payments	1	$225 + 119 \cdot p$	1	$225 + 88 \cdot p$

Evaluation code at: https://github.com/blitz-payments/overhead

Notation


Transaction
Output value
Relative timelock Δ (CSV)
tx^er
...
ϵ
...
pk_C
+Δ
pk_C, pk_D
tx^r
5 + ϵ
C
pk_C
Output
or
tx^p
5
D
pk_D
> T
pk_D
tx^state
5
x_C − 5
x_D
C
pk_C
D
pk_D
MultiSig w.r.t. pk_C, pk_D.
Underlined means that D gives signature to C when setting up
Input
"Owner"
Signature w.r.t. pk_C (public key of C)
after T
Abs. Timelock
(CLTV)

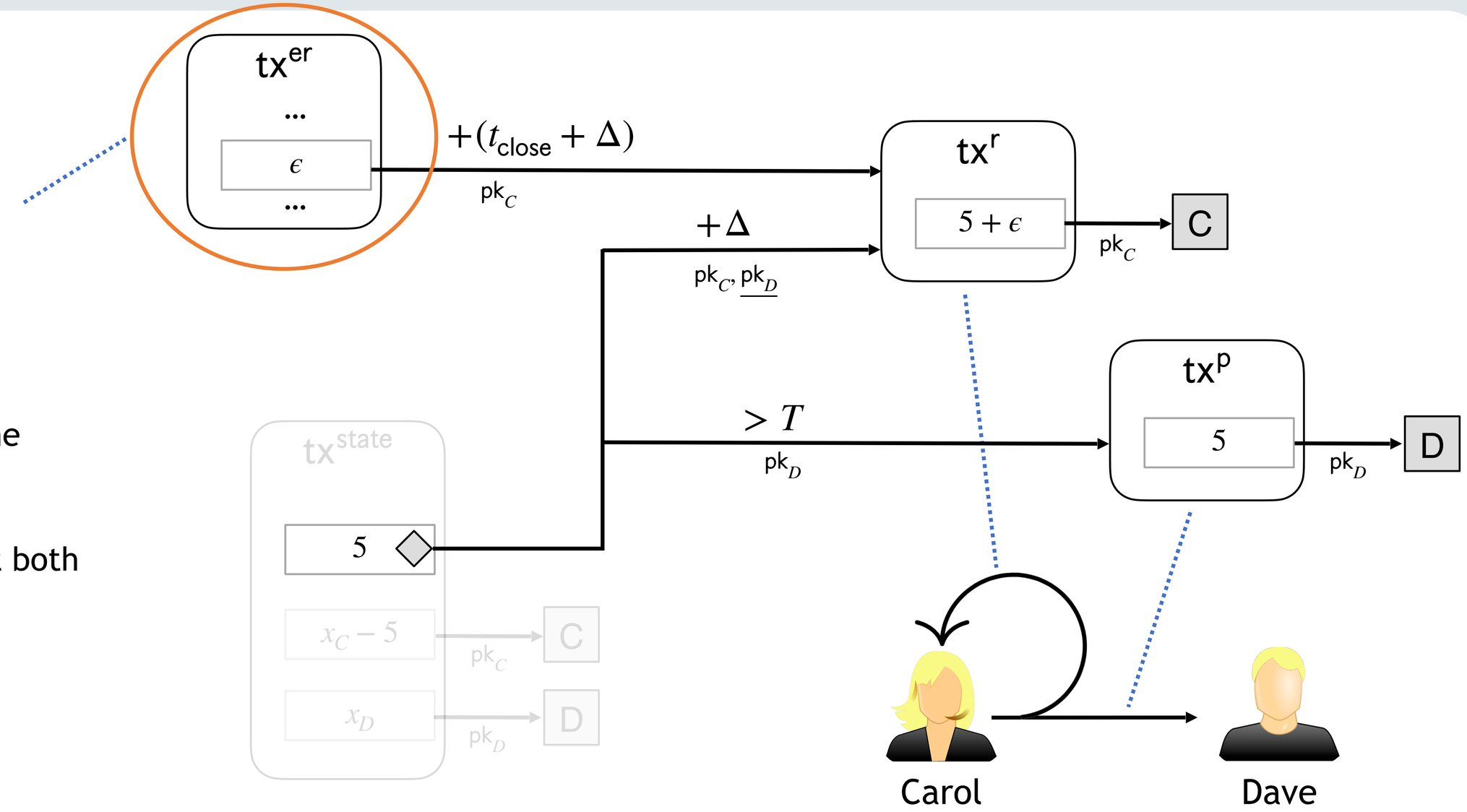
Irrational sender: Race condition



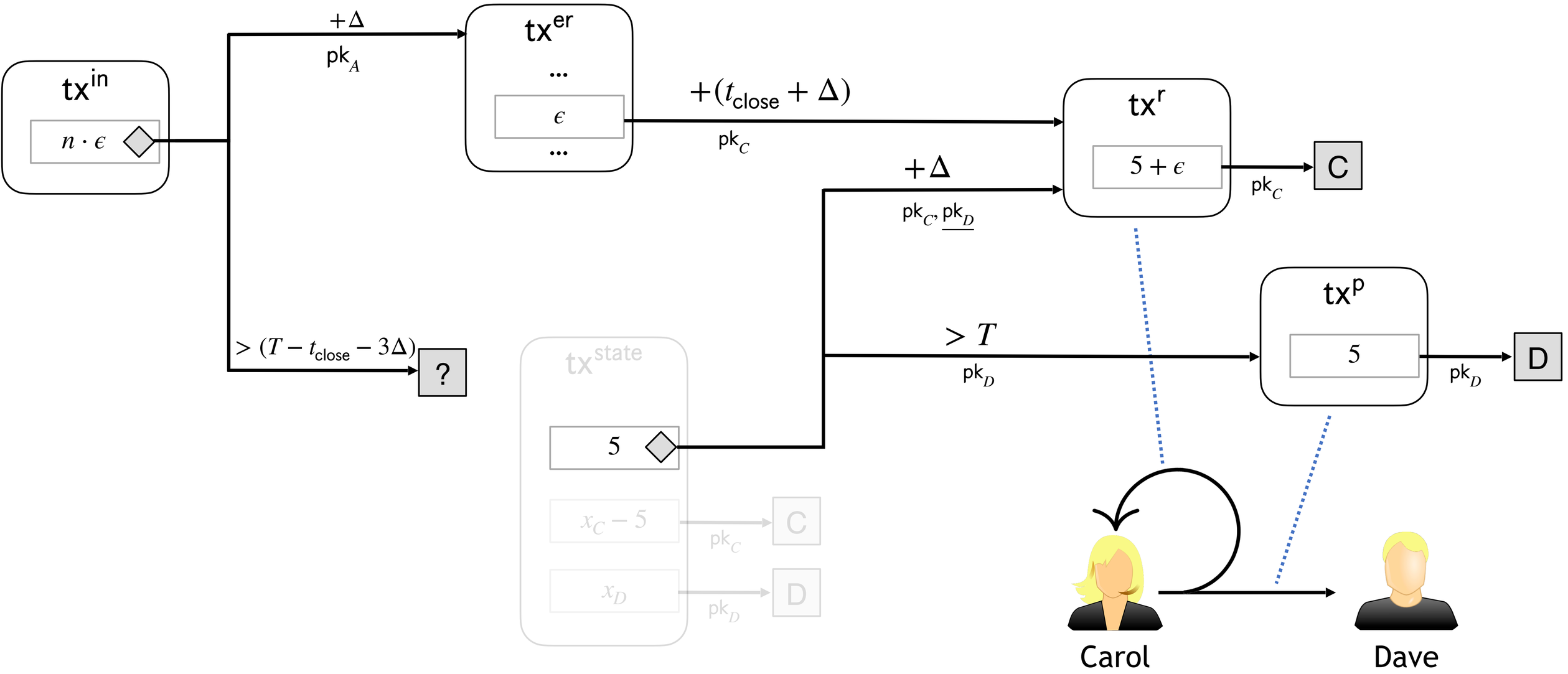
Irrational sender: Race condition

Sender posts tx^{er} at $(T - t_{\text{close}} - 2\Delta)$.
It gets accepted at $(T - t_{\text{close}} - \Delta)$
The time locks on the outputs expire at T.

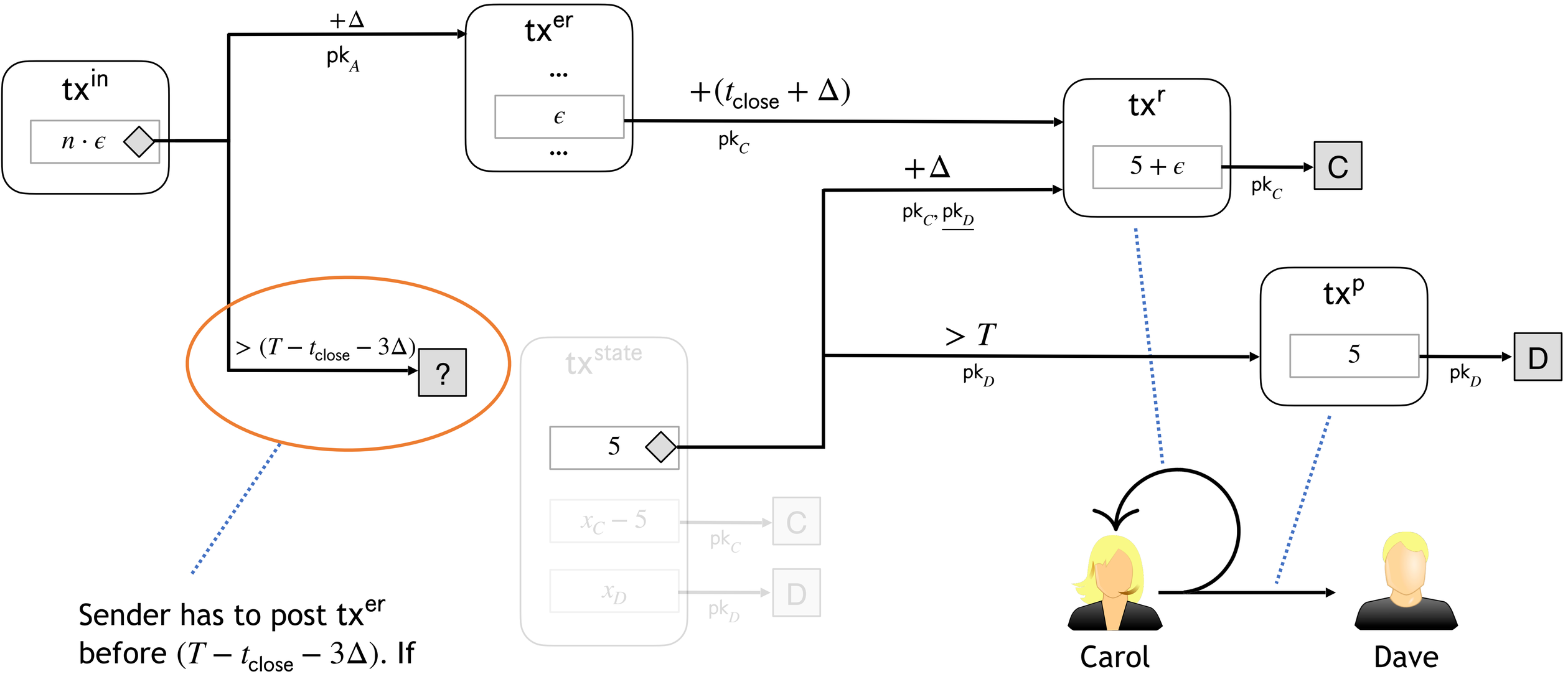
Refund and Payment both possible!



Prevent race condition

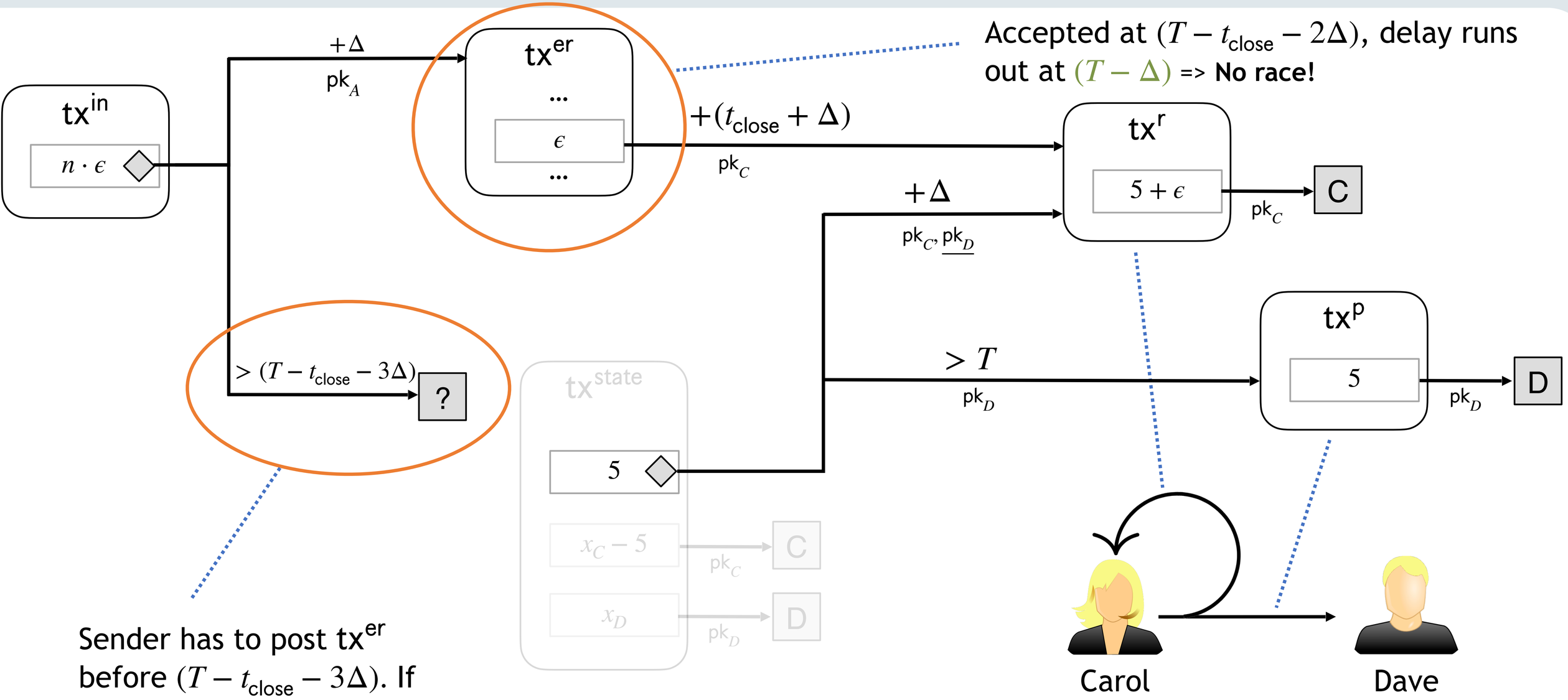


Prevent race condition



Sender has to post $\mathrm{tx}^{\mathrm{er}}$ before $(T - t_{\text{close}} - 3\Delta)$. If not, anyone can claim this output.

Prevent race condition



Sender has to post tx^{er} before $(T - t_{\text{close}} - 3\Delta)$. If not, anyone can claim this output.

Fast refund

Step 1

tx^{er}

...

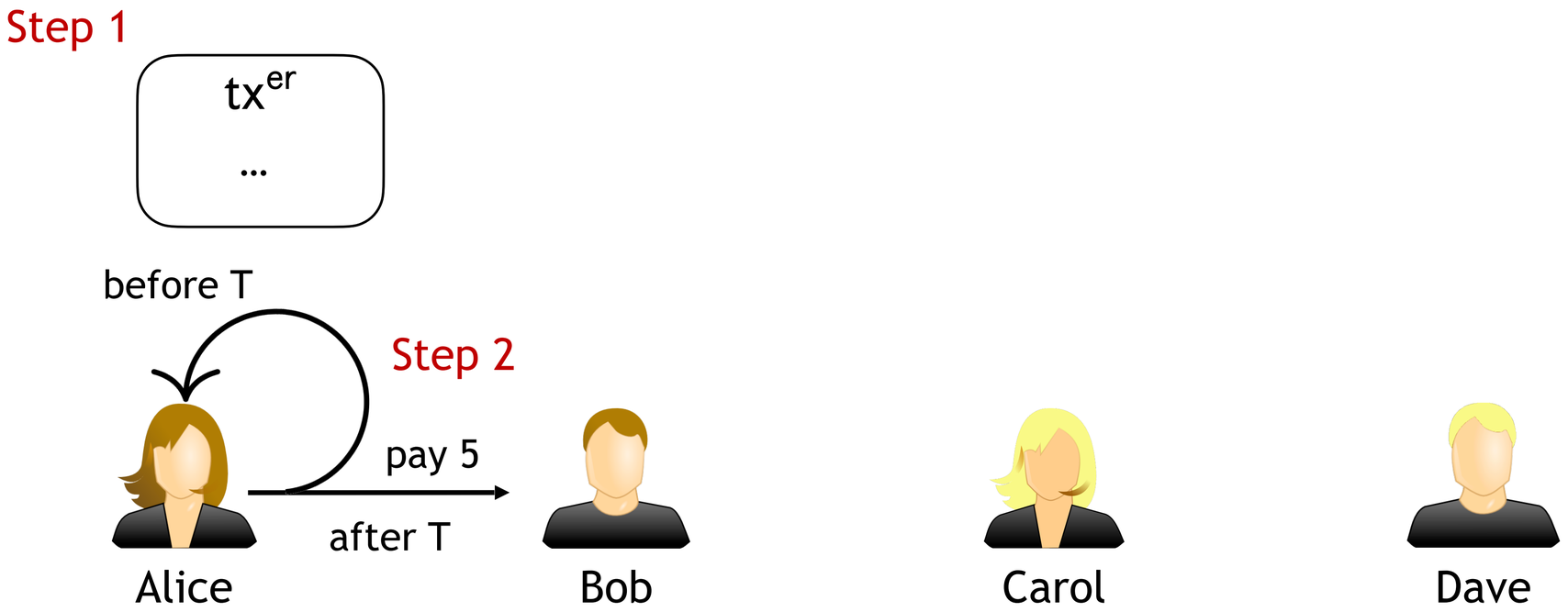
Alice

Bob

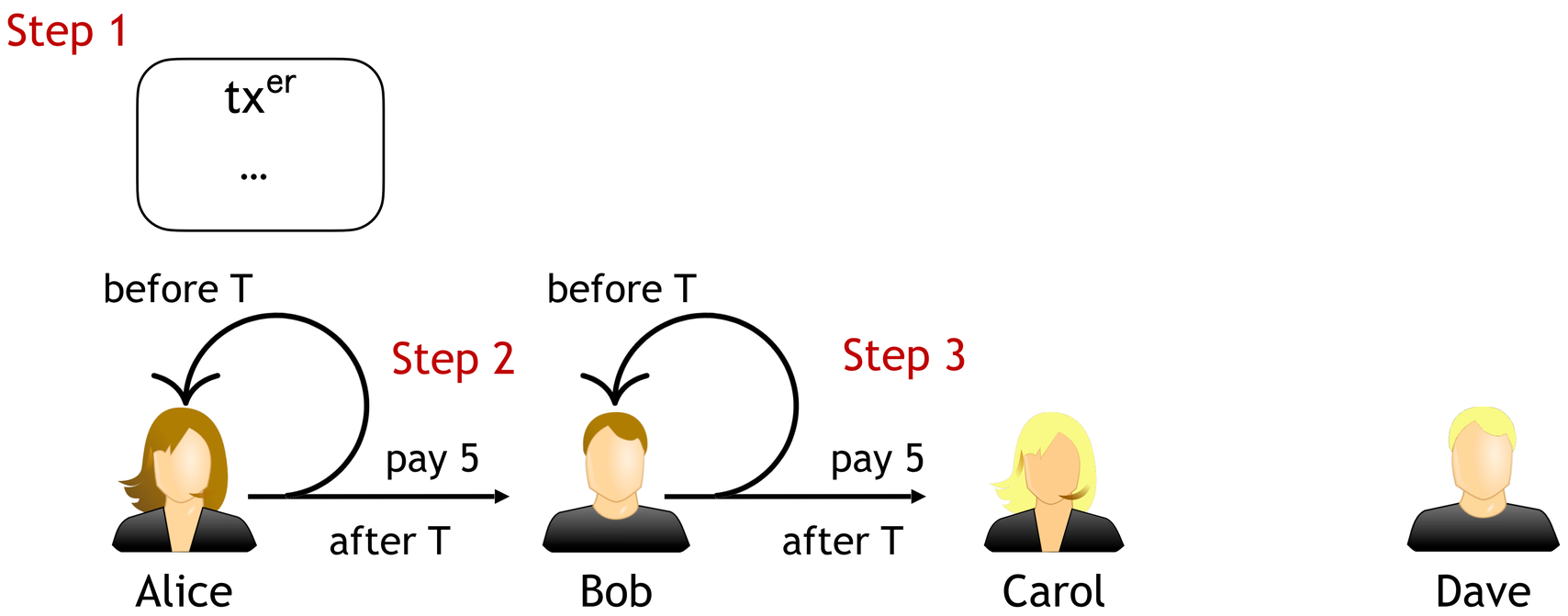
Carol

Dave

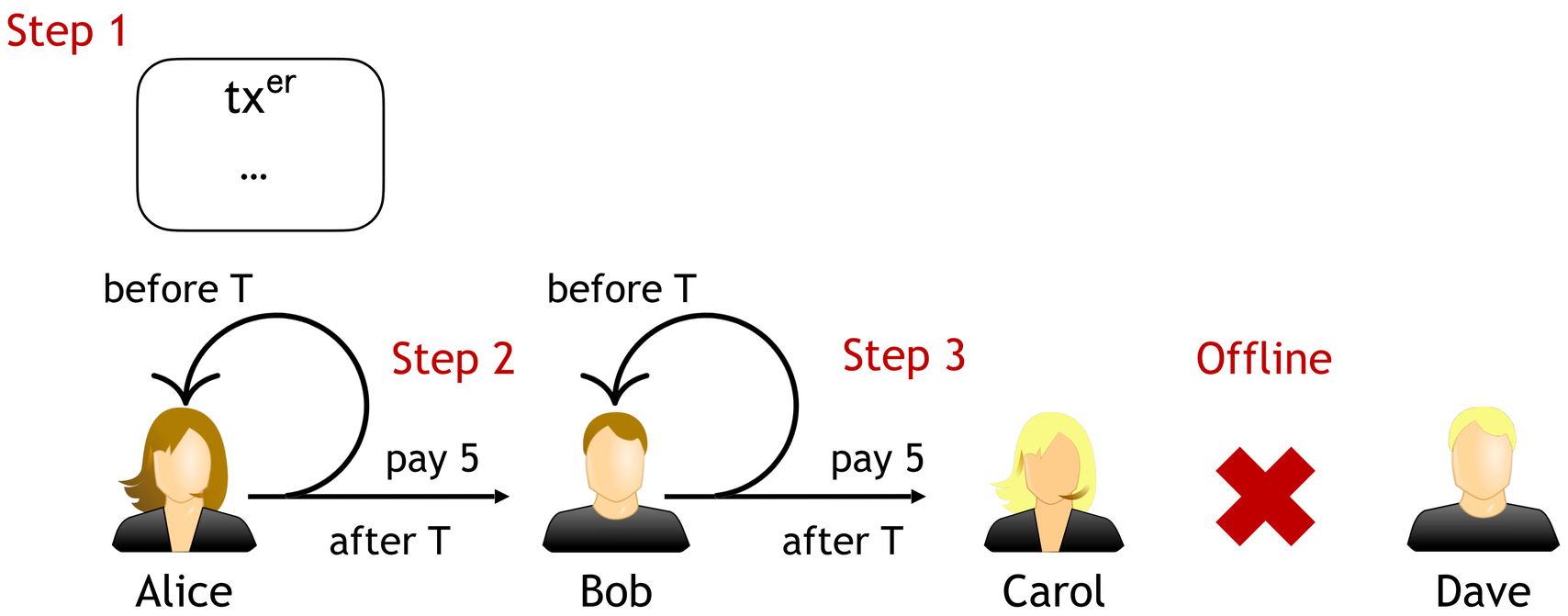
Fast refund



Fast refund

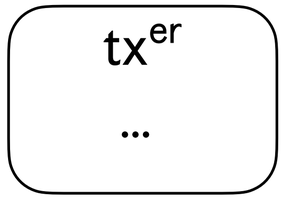

Step 1
tx^er
...
before T
Step 2
pay 5
after T
before T
Step 3
pay 5
after T
Alice
Bob
Carol
Dave

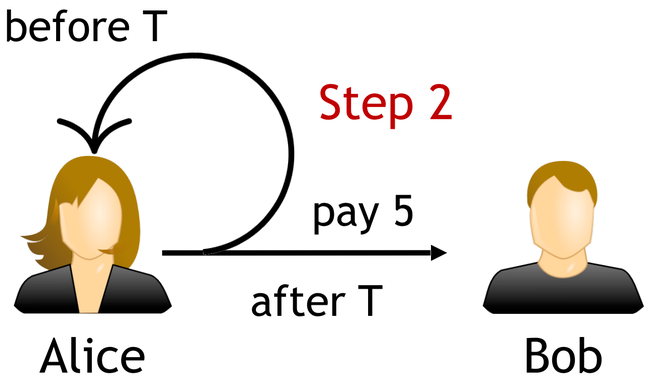
Fast refund


Step 1
tx^{er}
...
before T
Step 2
pay 5
after T
before T
Step 3
pay 5
after T
Offline
Alice
Bob
Carol
Dave

Fast refund

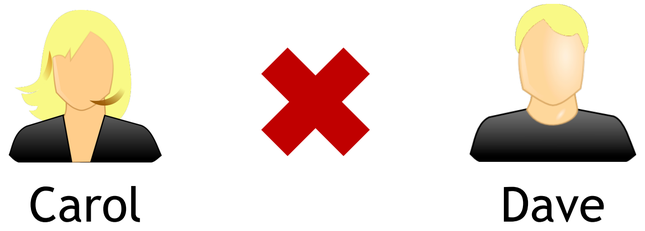
Step 1





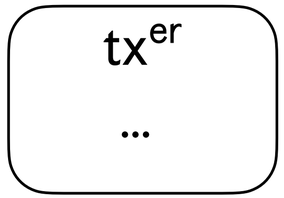
Step 4

Offline



Fast refund

Step 1



Step 5

Alice

Bob

Step 4

Carol

Offline

Dave

- Payment refunded without publishing tx^{er}
- Incentivize this behavior by giving fees to those participating