Scaling distributed ledgers and privacy-preserving applications

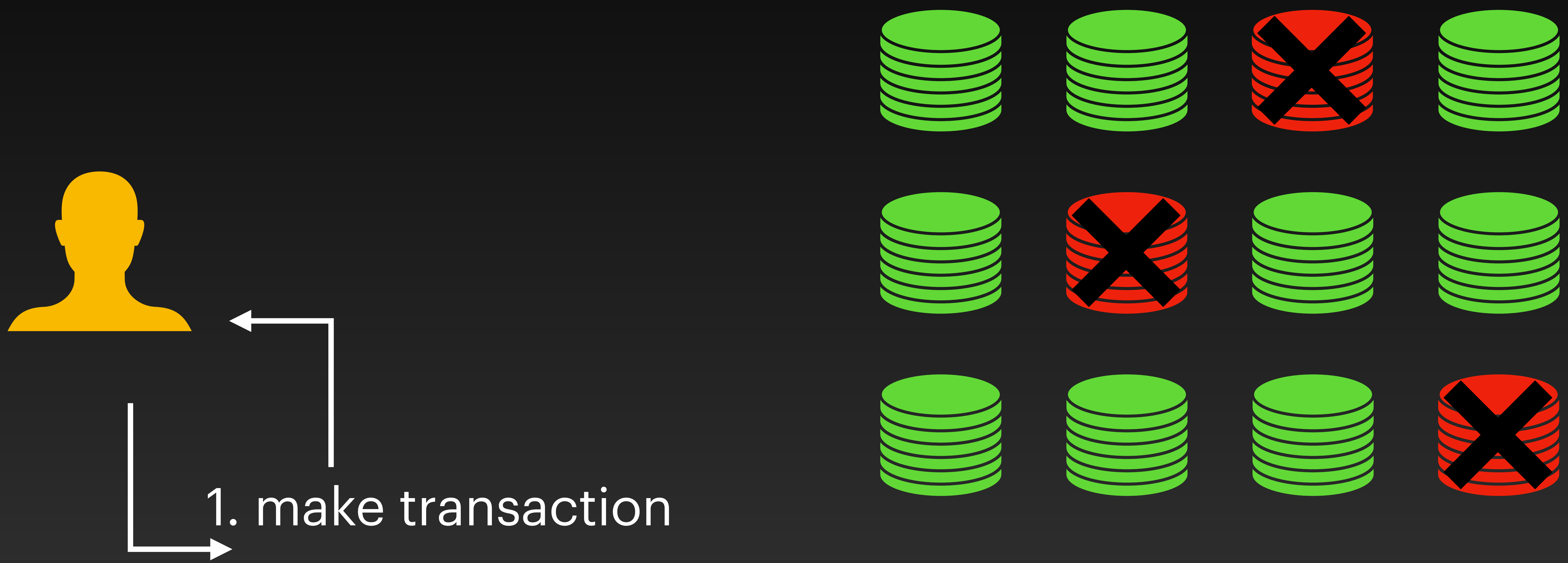
PhD Defense

Alberto Sonnino

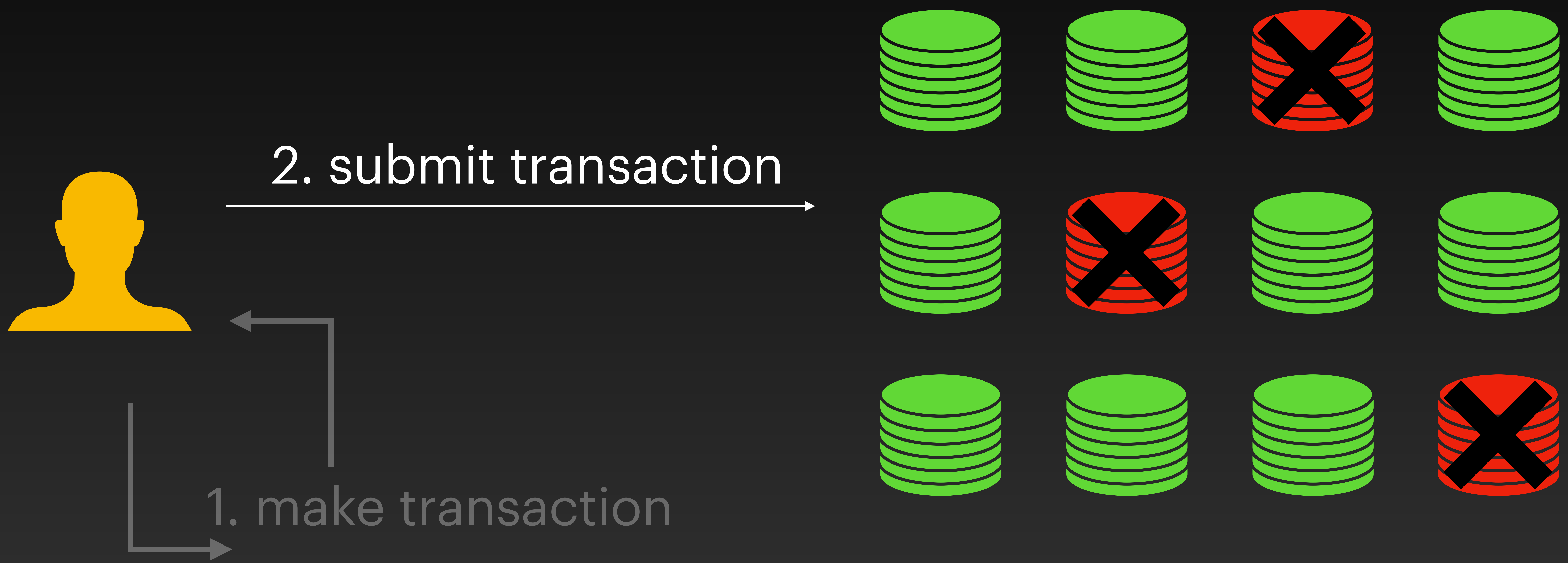
A set of nodes

... Some of which are bad

Blockchains

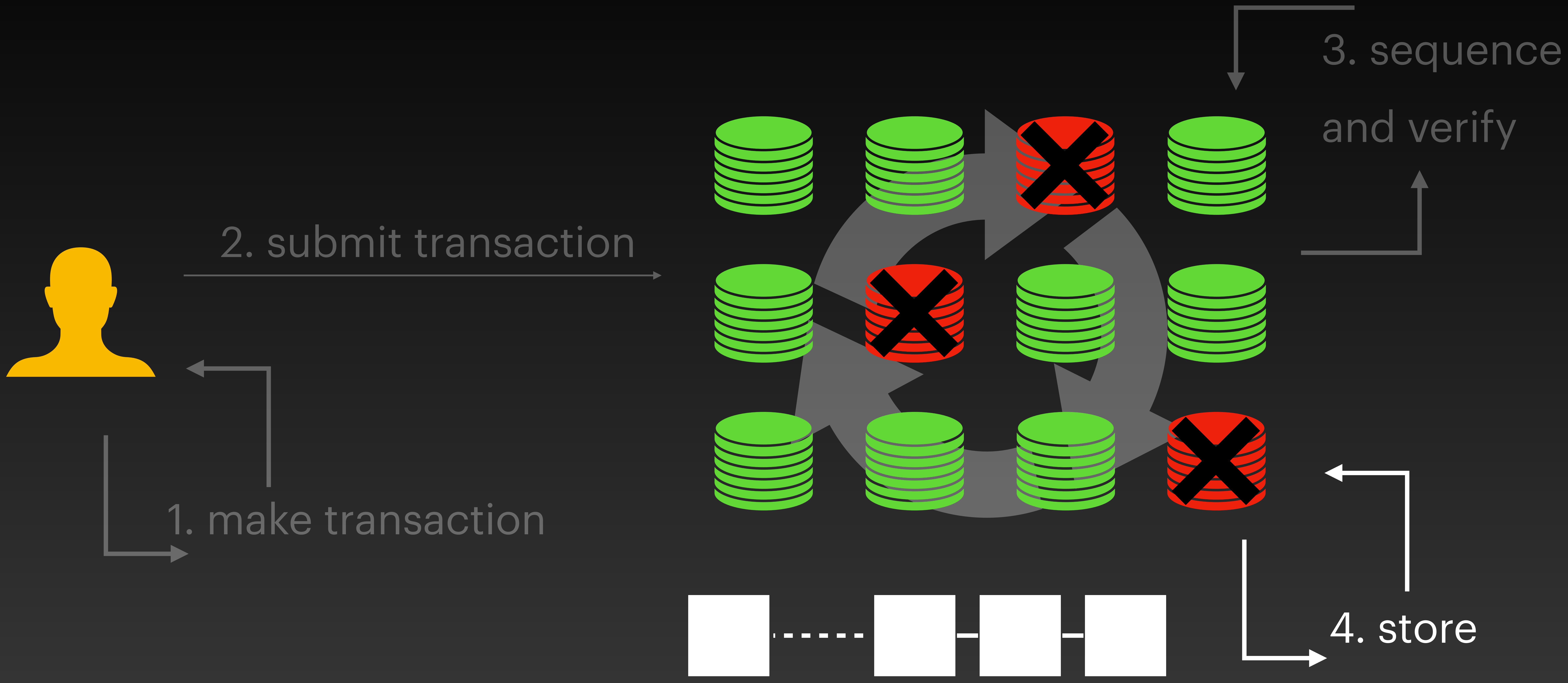


Blockchains



Blockchains

Blockchains



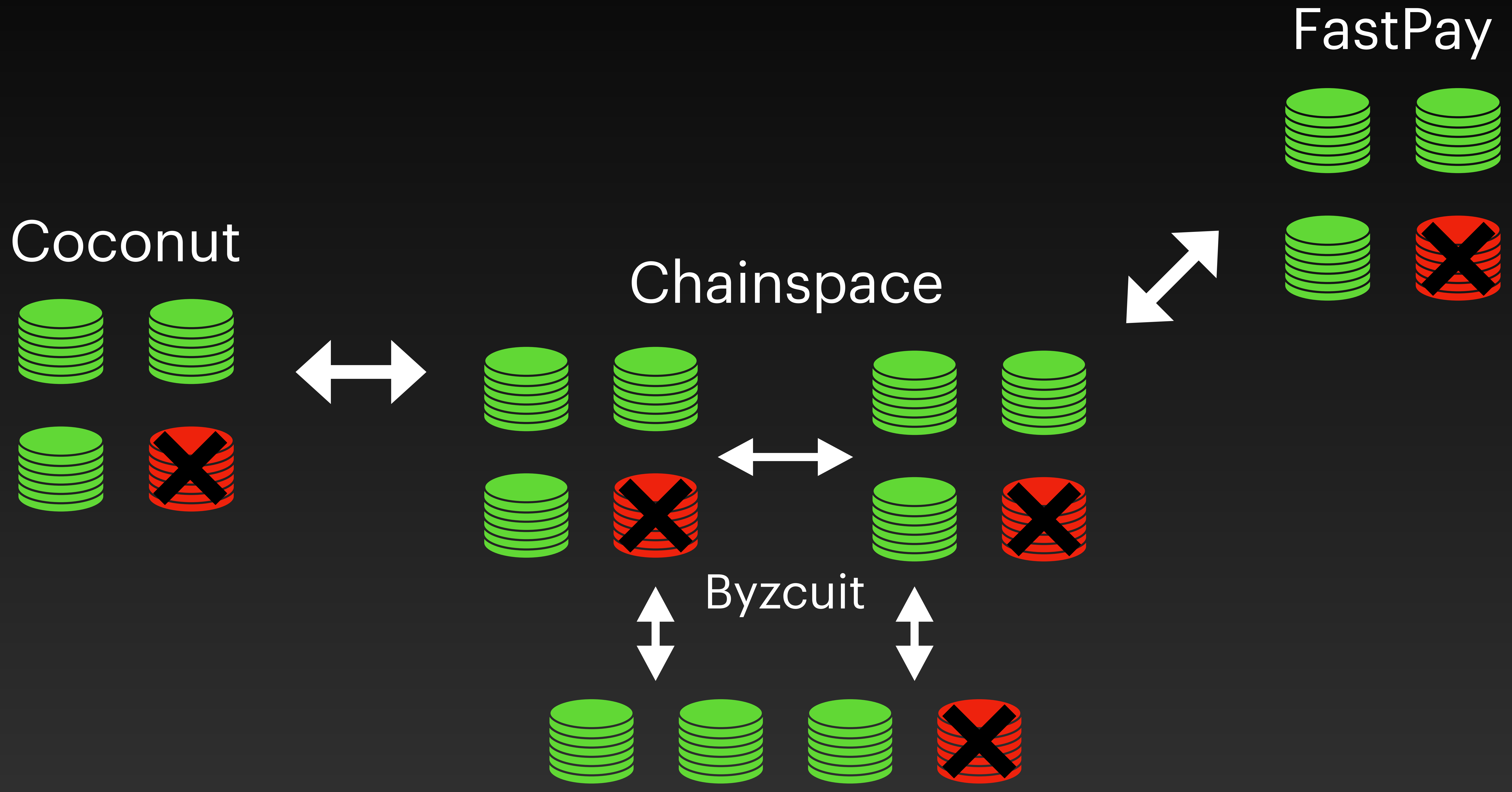
Low throughput

High latency

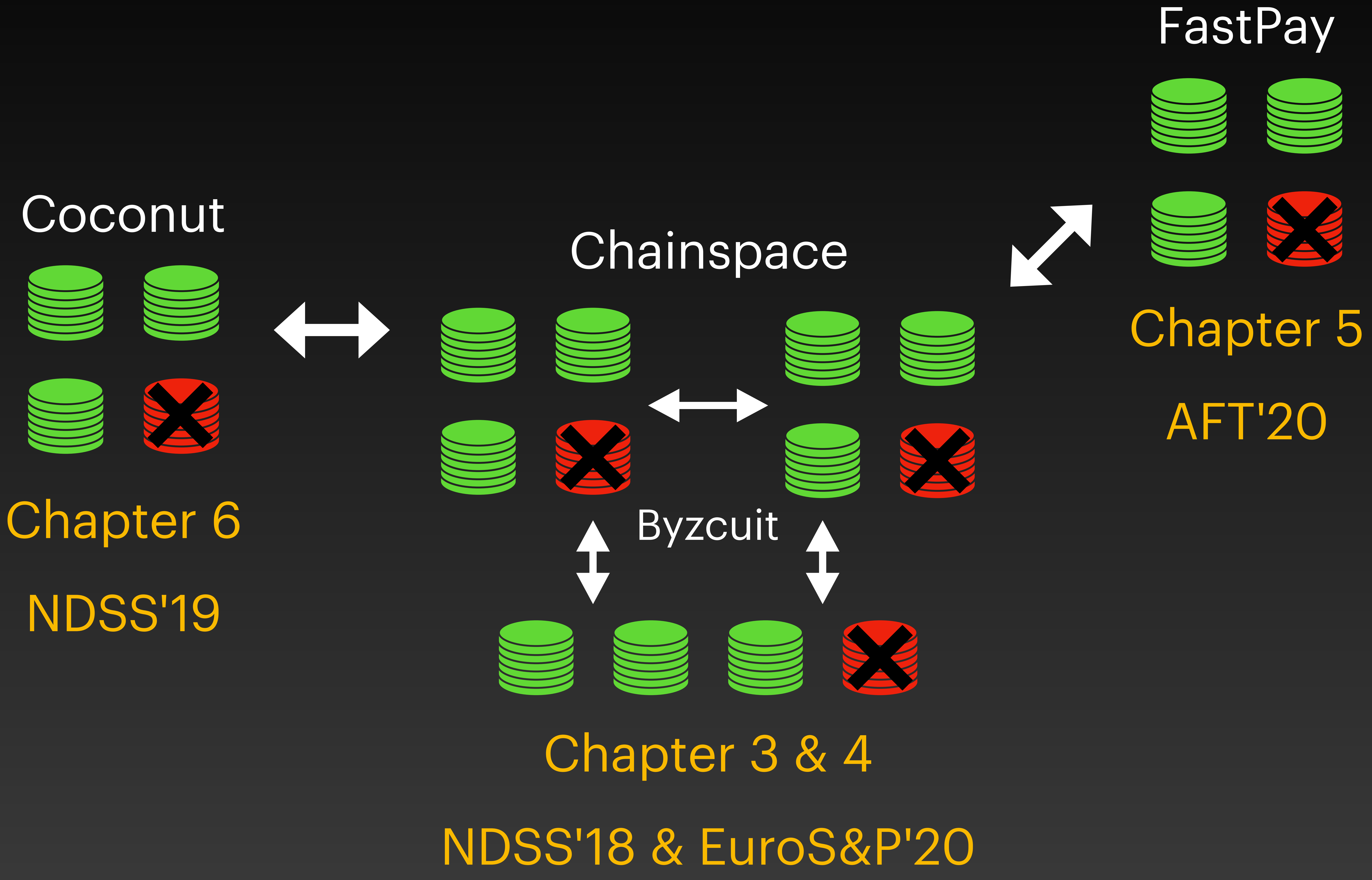
Slow finality

Poor privacy

Overview



Overview


FastPay
Chapter 5
AFT'20
Coconut
Chapter 6
NDSS'19
Chainspace
Byzcuit
Chapter 3 & 4
NDSS'18 & EuroS&P'20

Chainspace & Byzcuit

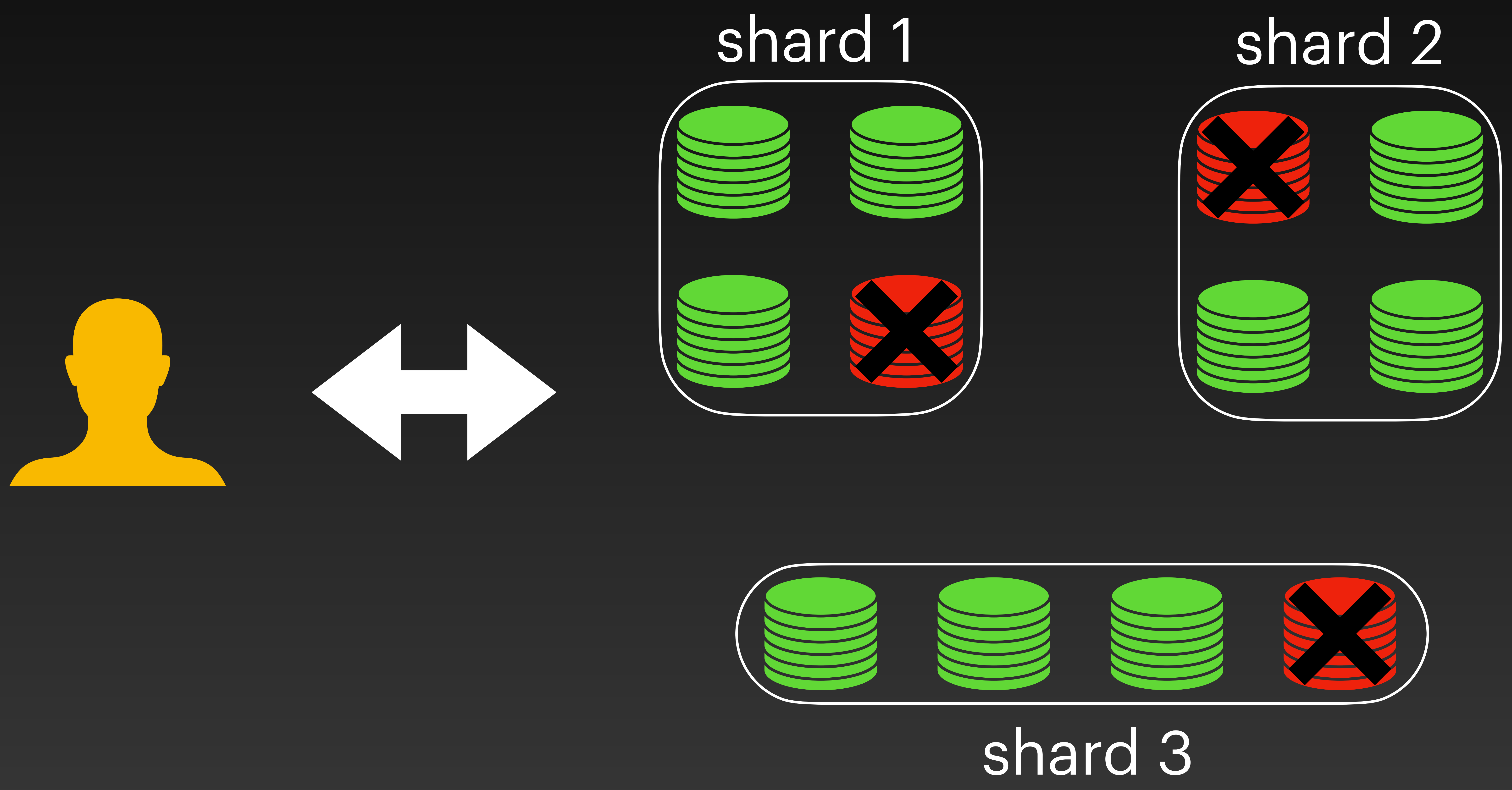
A scalable backbone with integrated privacy support

Chainspace

State sharding

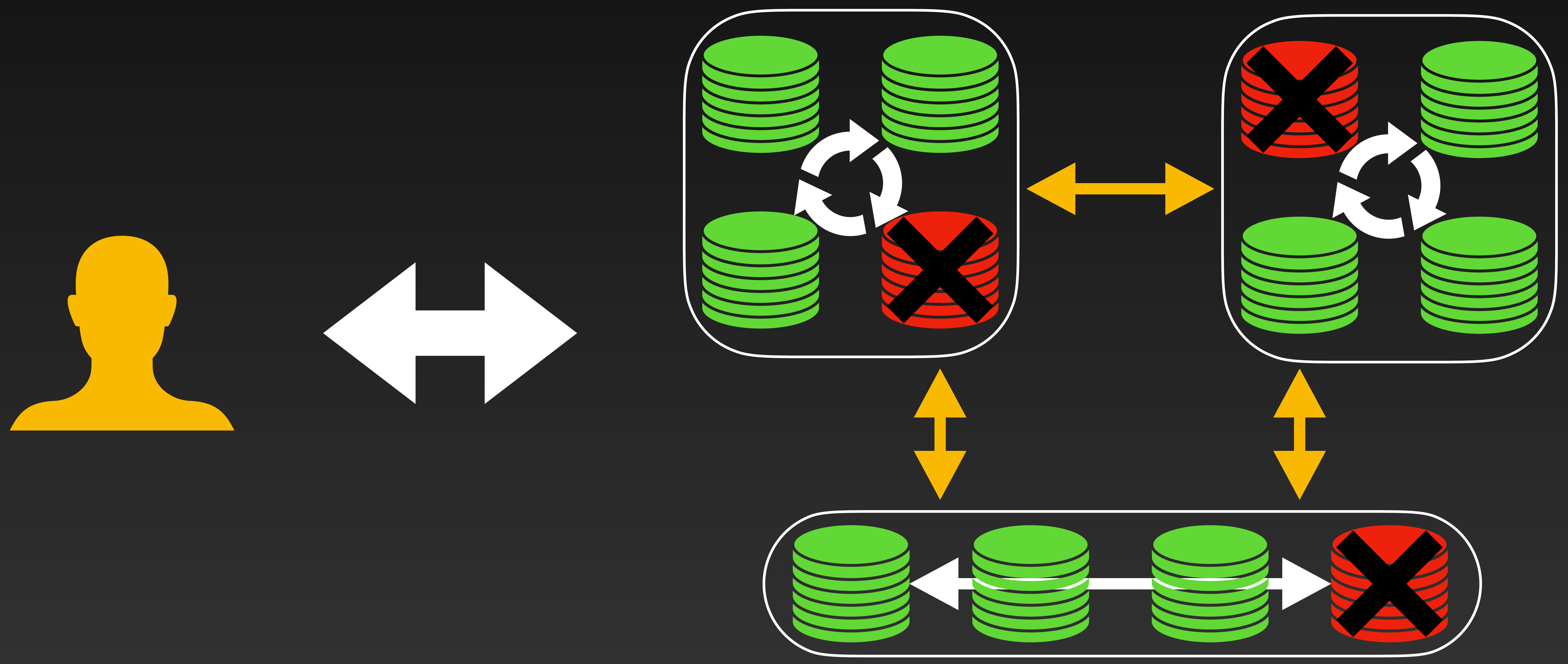
Chainspace

State sharding



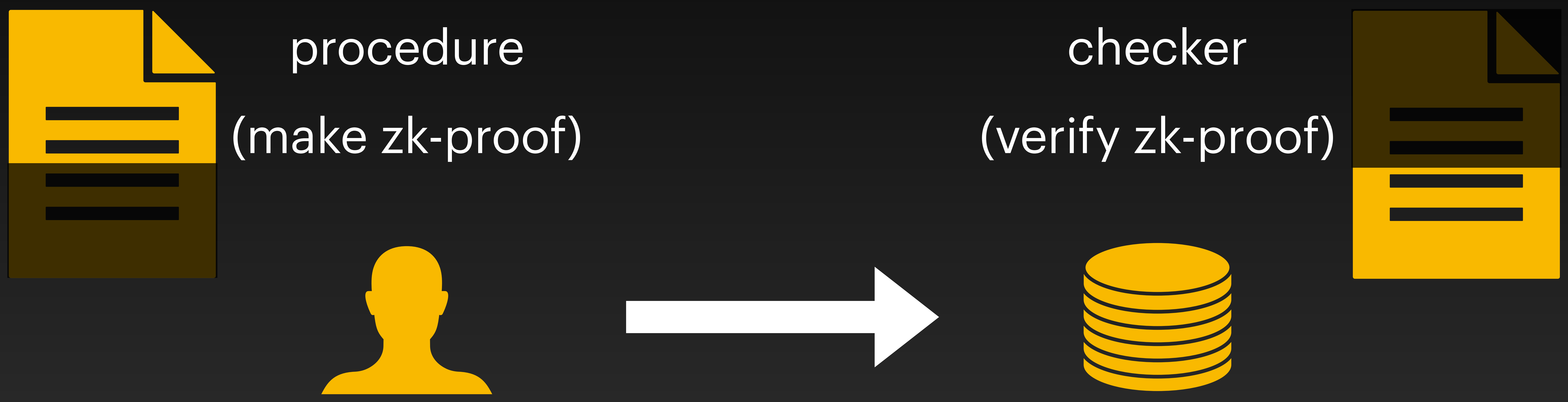
Byzcuit

Cross-shard consensus protocol



Privacy by Design

Privacy by Design



High throughput ✓

Low latency ✓

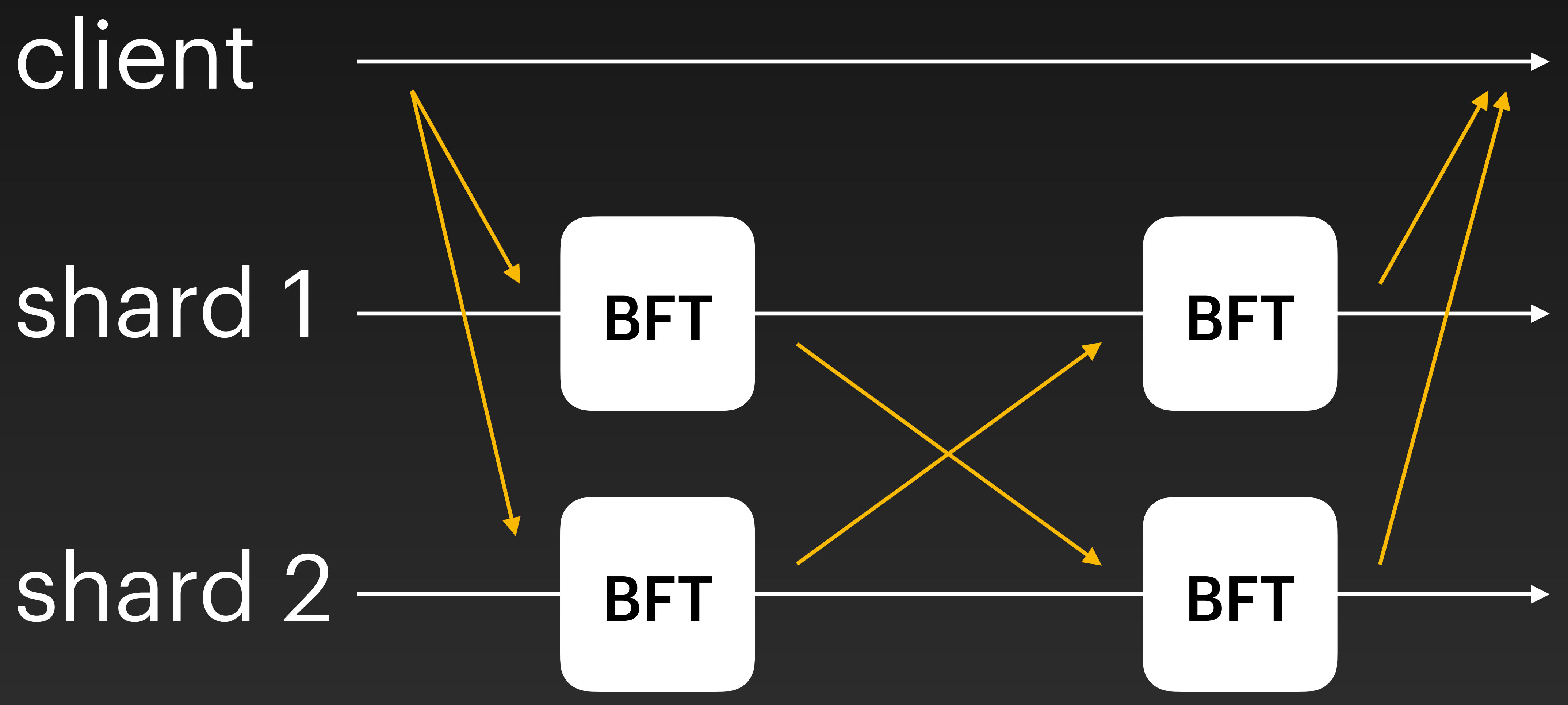
Fast finality ✓

Good privacy ✓

FastPay

A low-latency payment system

What we have so far



Total Latency:

slowest shard during phase 1

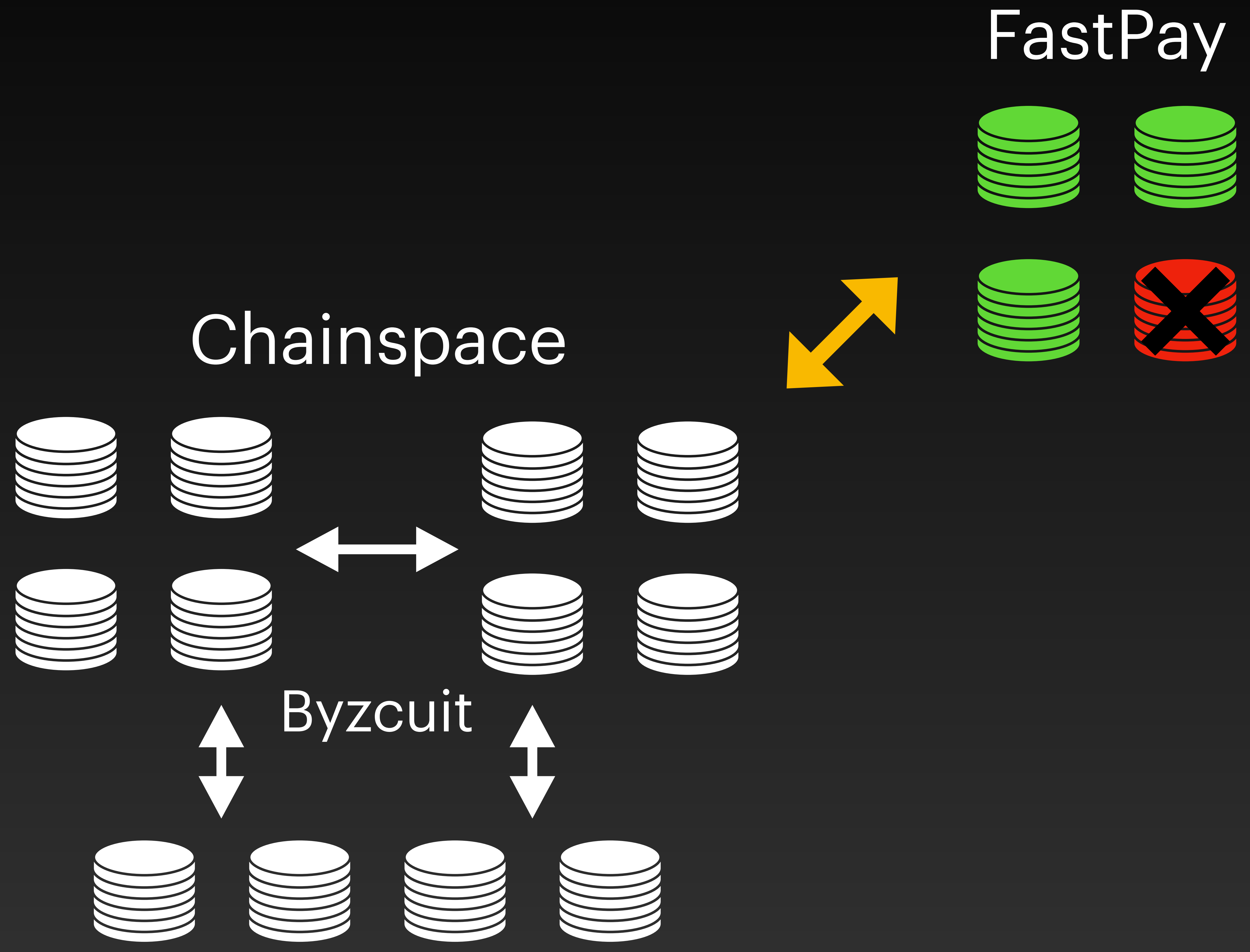
+

slowest shard during phase 2

+

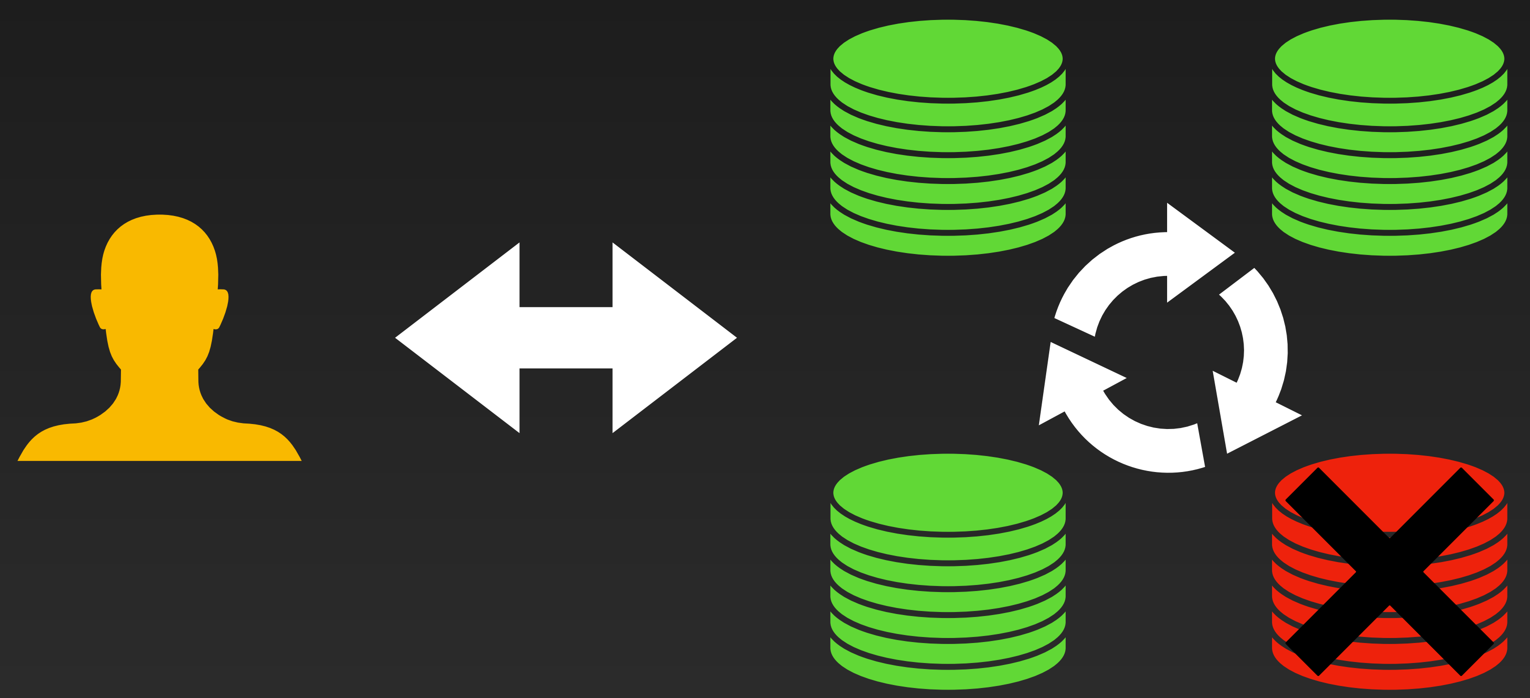
all communications

Overview



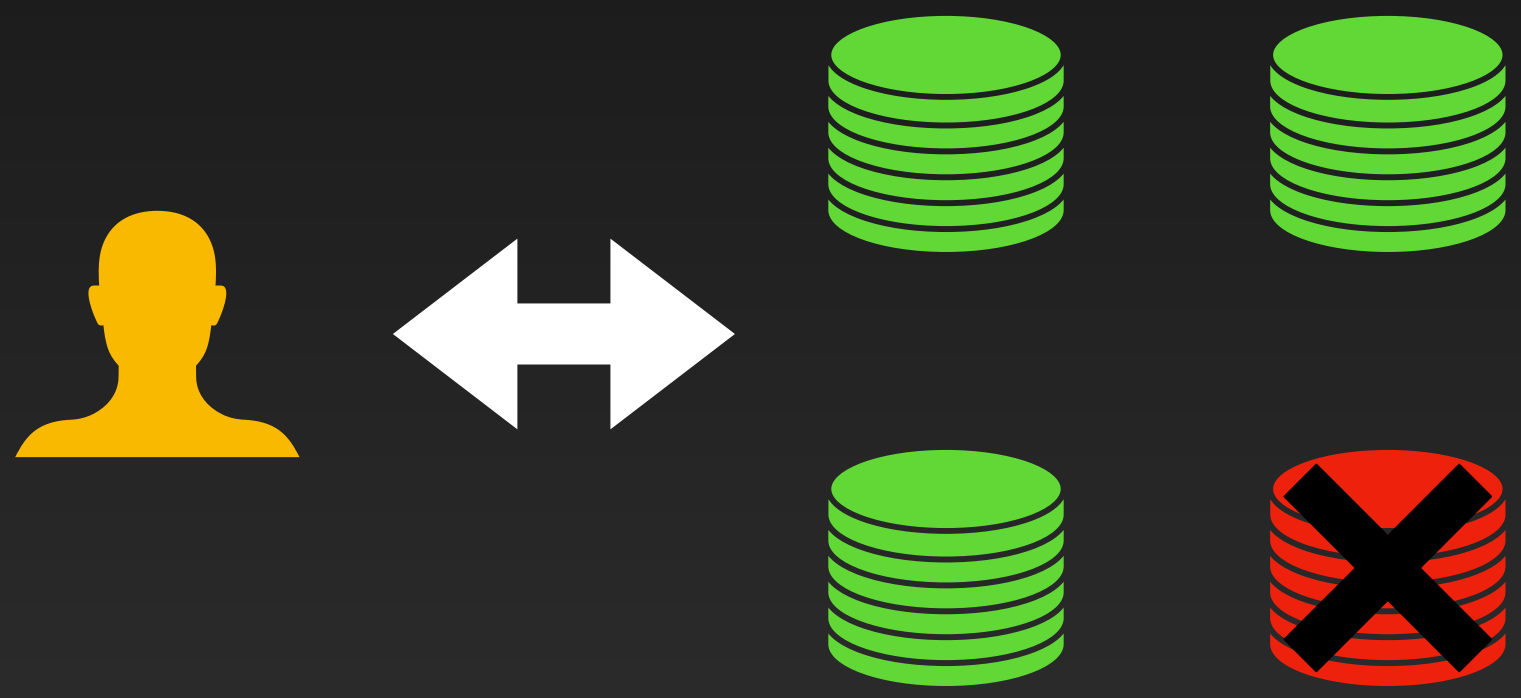
Difference with blockchains

Blockchains



Byzantine Consensus

FastPay



Byzantine Consistent Broadcast

High throughput ✓

Low latency ✓

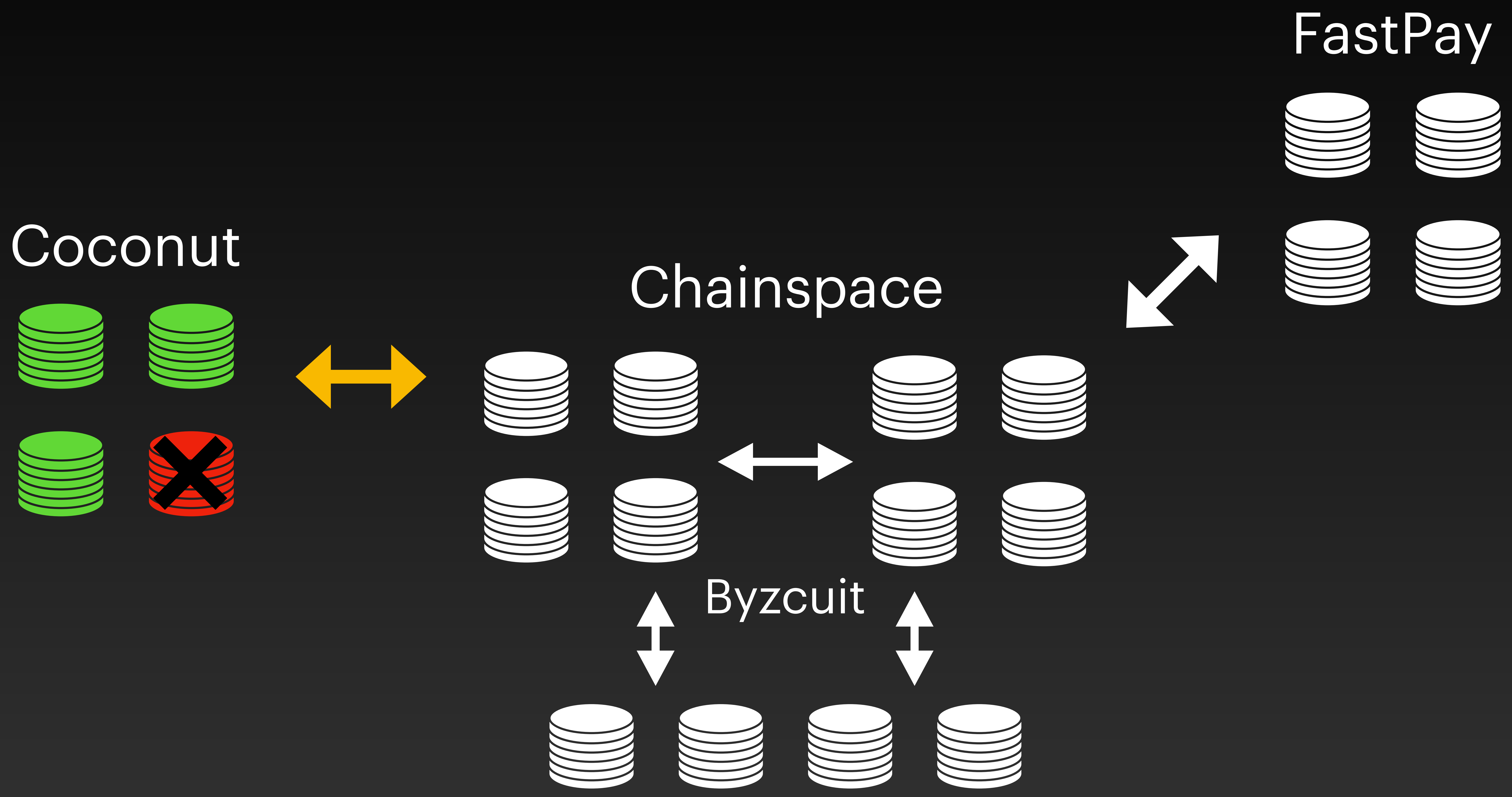
Fast finality ✓

Good privacy ✓

Coconut

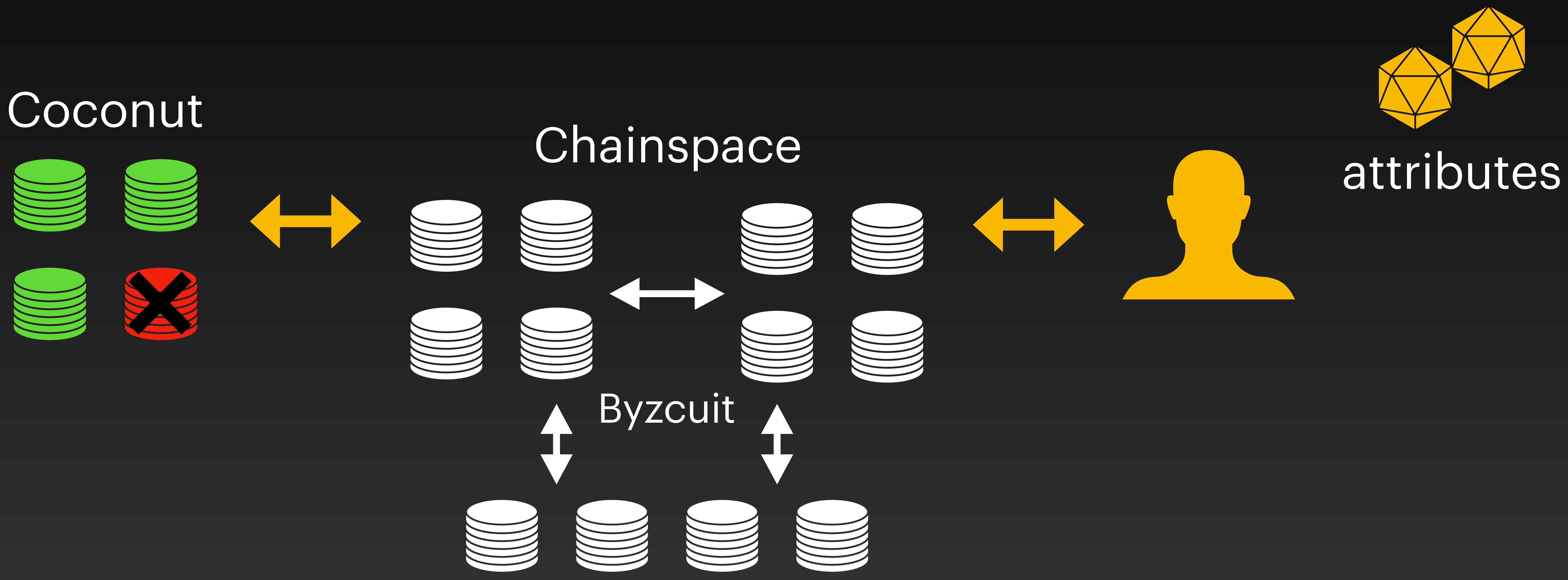
Privacy-preserving credentials for smart contract applications

Overview



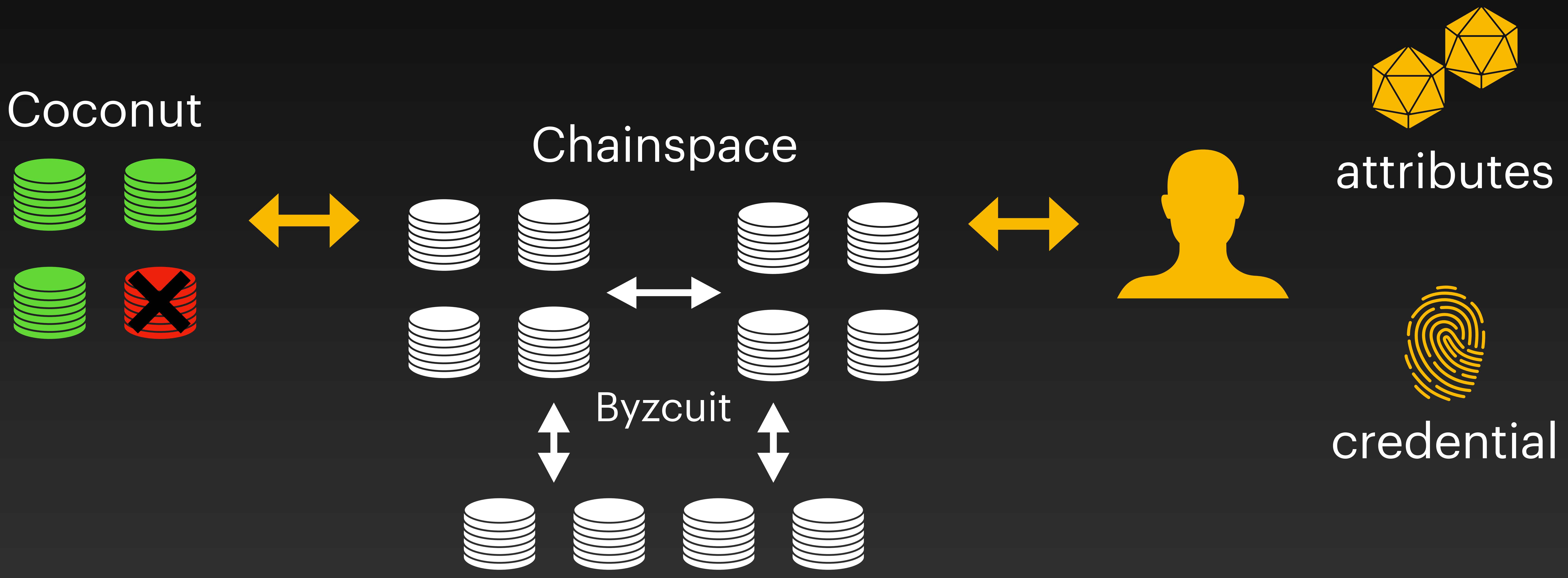
Coconut

Anonymous credentials in a blockchain setting



Coconut

Anonymous credentials in a blockchain setting



High throughput ✓

Low latency ✓

Fast finality ✓

Good privacy ✓

15. Fraud Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities

Mustafa Al-Bassam, **Alberto Sonnino**, Vitalik Buterin, Ismail Khoffi, Financial Cryptography and Data Security (FC), 2021

14. EL PASSO: Privacy-preserving, Asynchronous Single Sign-On

Zhiyi Zhang, Michał Król, **Alberto Sonnino**, Lixia Zhang, Etienne Rivière, International Symposium on Privacy Enhancing Technologies (PETs), 2021

13. Twins: White-Glove Approach for BFT Testing

Shehar Bano, **Alberto Sonnino**, Andrey Chursin, Dmitri Perelman, Zekun Li, Avery Ching, Dahlia Malkhi, ArXiv Preprint, 2020

12. FastPay: High-Performance Byzantine Fault Tolerant Settlement

Mathieu Baudet, George Danezis, **Alberto Sonnino**, ACM Conference on Advances in Financial Technologies (AFT), 2020

11. Replay Attacks and Defenses against Cross-shard Consensus in Sharded Distributed Ledgers

Alberto Sonnino, Shehar Bano, Mustafa Al-Bassam, George Danezis, IEEE European Symposium on Security and Privacy (EuroS&P), 2020

10. PASTRAMI: Privacy-preserving, Auditable, Scalable & Trust-worthy Auctions for Multiple Items.

Michał Król, **Alberto Sonnino**, Argyrios G. Tasiopoulos, Ioannis Psaras, Etienne Rivière ACM/IFIP Middleware, 2020

9. Location, location, location: Revisiting Modeling and Exploitation for Location-based Side Channel Leakages

Christos Andrikos, Lejla Batina, Lukasz Chmielewski, Liran Lerman, Vasilios Mavroudis, Kostas Papagiannopoulos, Guilherme Perin, Giorgos Rassias, **Alberto Sonnino**, Asiacrypt, 2019

8 FMPC: Secure Multiparty Computation from Fourier Series and Parseval's Identity

Alberto Sonnino, ArXiv Preprint, 2019

7. AStERISK: Auction-based Shared Economy ResolutIon System for blocKchain

Alberto Sonnino, Michał Król, Argyrios Tasiopoulos, Ioannis Psaras, Workshop on Decentralized IoT Systems and Security (DISS), 2019

6. SybilQuorum: Open Distributed Ledgers Through Trust Networks

Alberto Sonnino, George Danezis, ArXiv Preprint, 2019

5. Proof-of-Prestige: A Useful Work Reward System for Unverifiable Tasks

Michał Król, **Alberto Sonnino**, Mustafa Al-Bassam, Argyrios Tasiopoulos, Ioannis Psaras, IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2019

4. Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers

Alberto Sonnino, Mustafa Al-Bassam, Shehar Bano, Sarah Meiklejohn, George Danezis, Proceedings of the Network and Distributed System Security Symposium (NDSS), 2019

3. SoK: Consensus in the Age of Blockchains

Shehar Bano, **Alberto Sonnino**, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, George Danezis, ACM Conference on Advances in Financial Technologies (AFT), 2019

2. Airtnt: Fair Exchange Payment for Outsourced Secure Enclave Computations

Mustafa Al-Bassam, **Alberto Sonnino**, Michał Król, Ioannis Psaras, ArXiv preprint, 2018

1. Chainspace: A Sharded Smart Contracts Platform

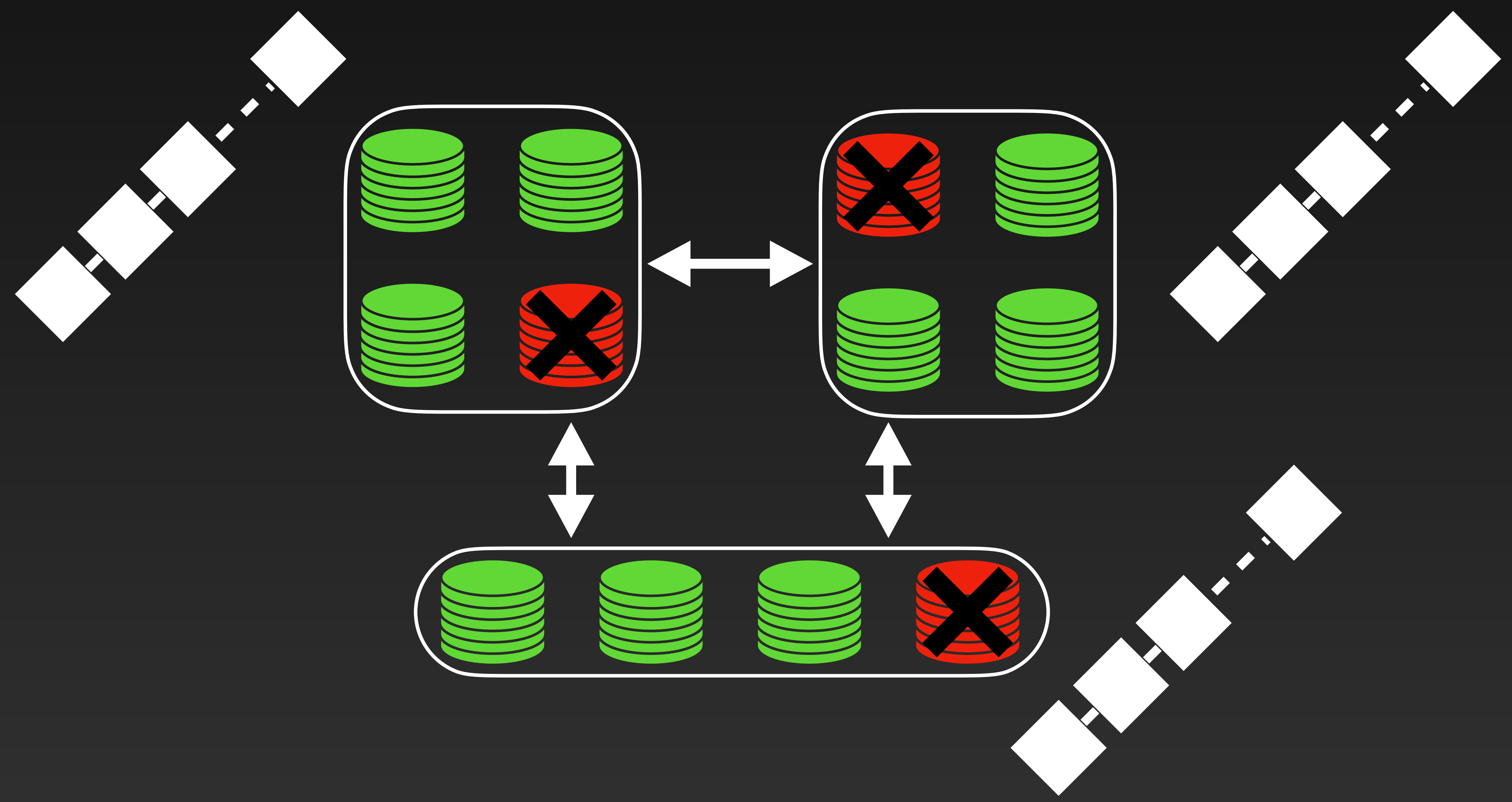
Mustafa Al-Bassam, **Alberto Sonnino**, Shehar Bano, Dave Hrycyszyn, George Danezis, Proceedings of the Network and Distributed System Security Symposium (NDSS), 2018

EXTRA

State Sharding

Traditional

Sharding



State Sharding

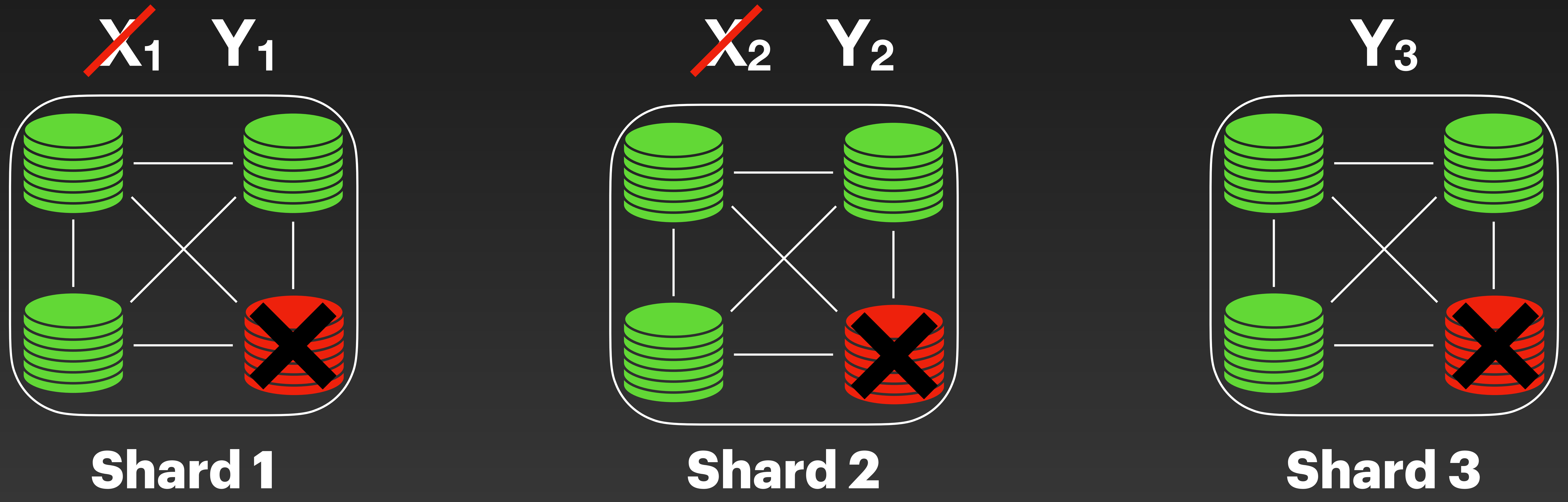
An example transaction

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$

State Sharding

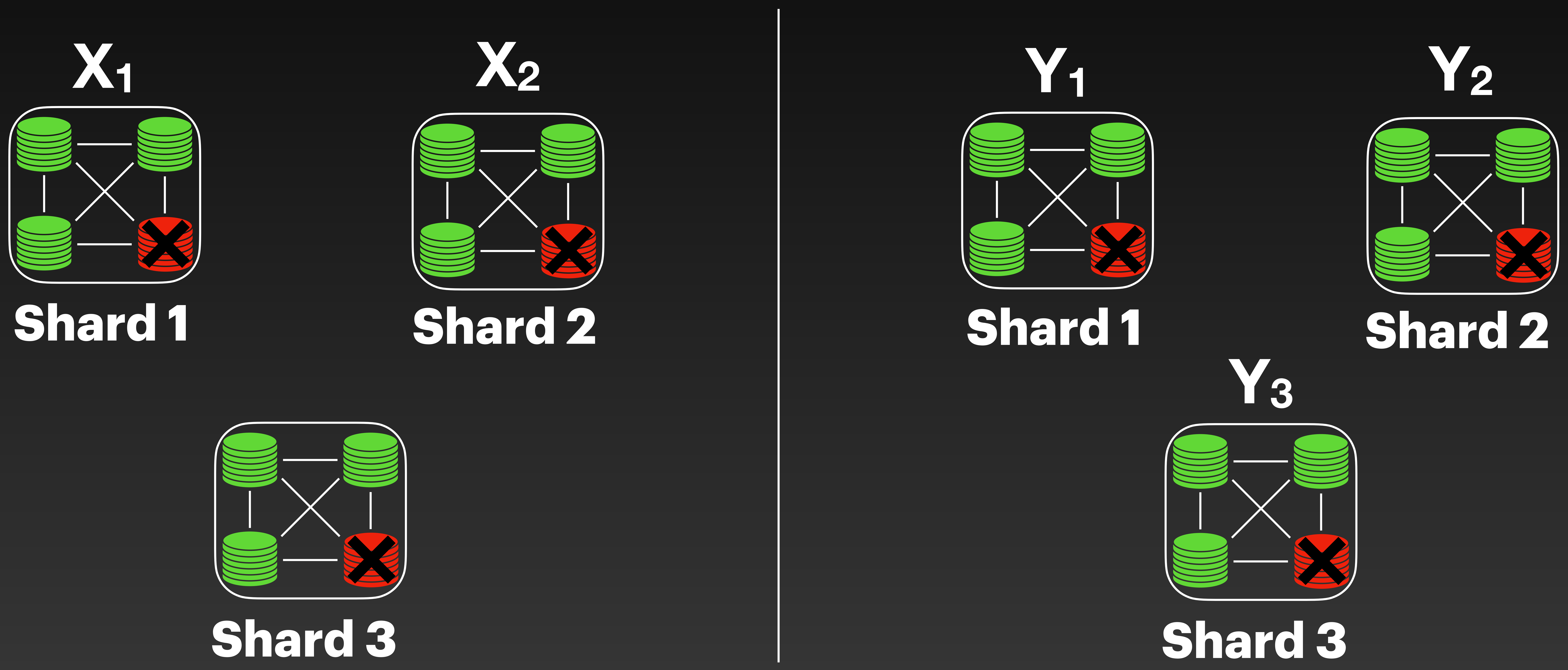
An example transaction

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



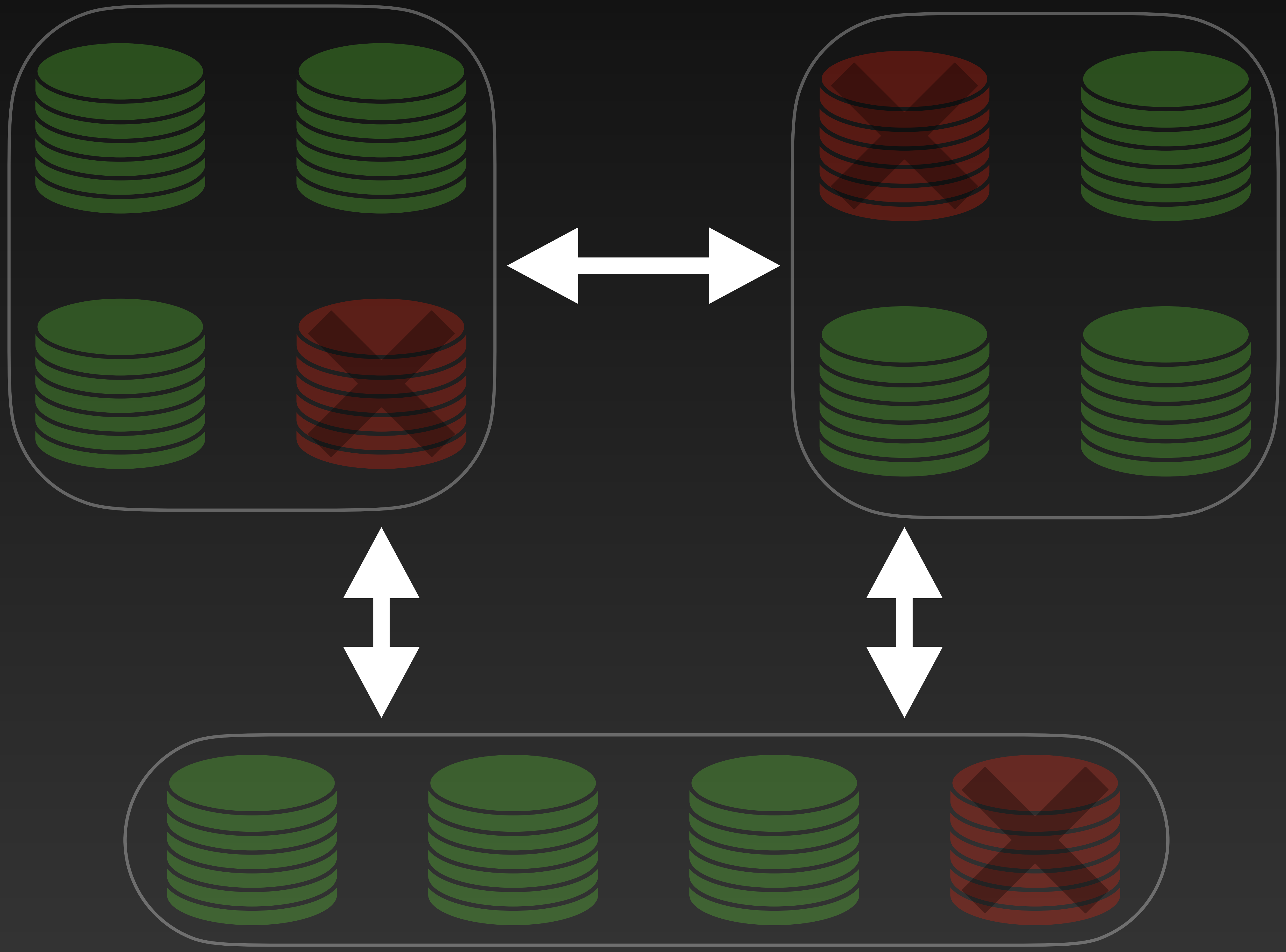
State Sharding

Only two acceptable final states



Cross-Shard Consensus

How do shards communicate with each other?



EXTRA

S-BAC Attacks

Attacks

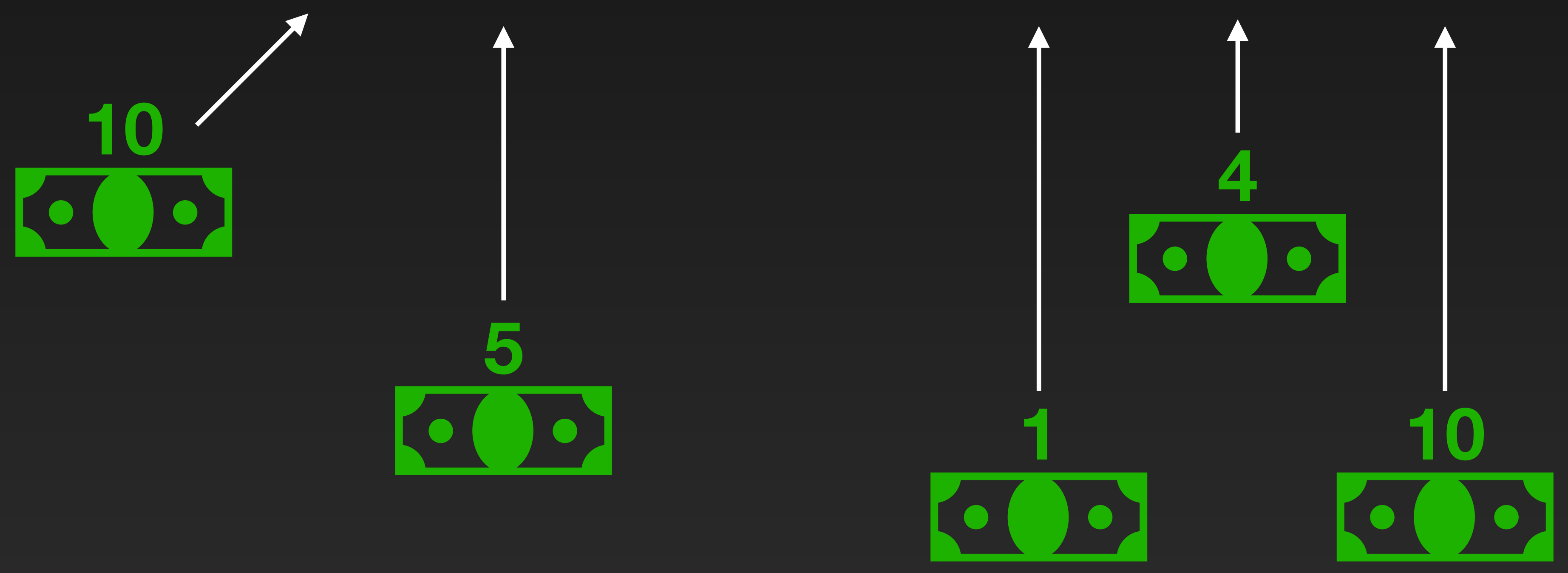
Double spend any object

- Does not need to collude with any node
- Acts as client or passive observer
- Re-orders network messages (not always needed)

Attack against S-BAC

Double-spend X_1

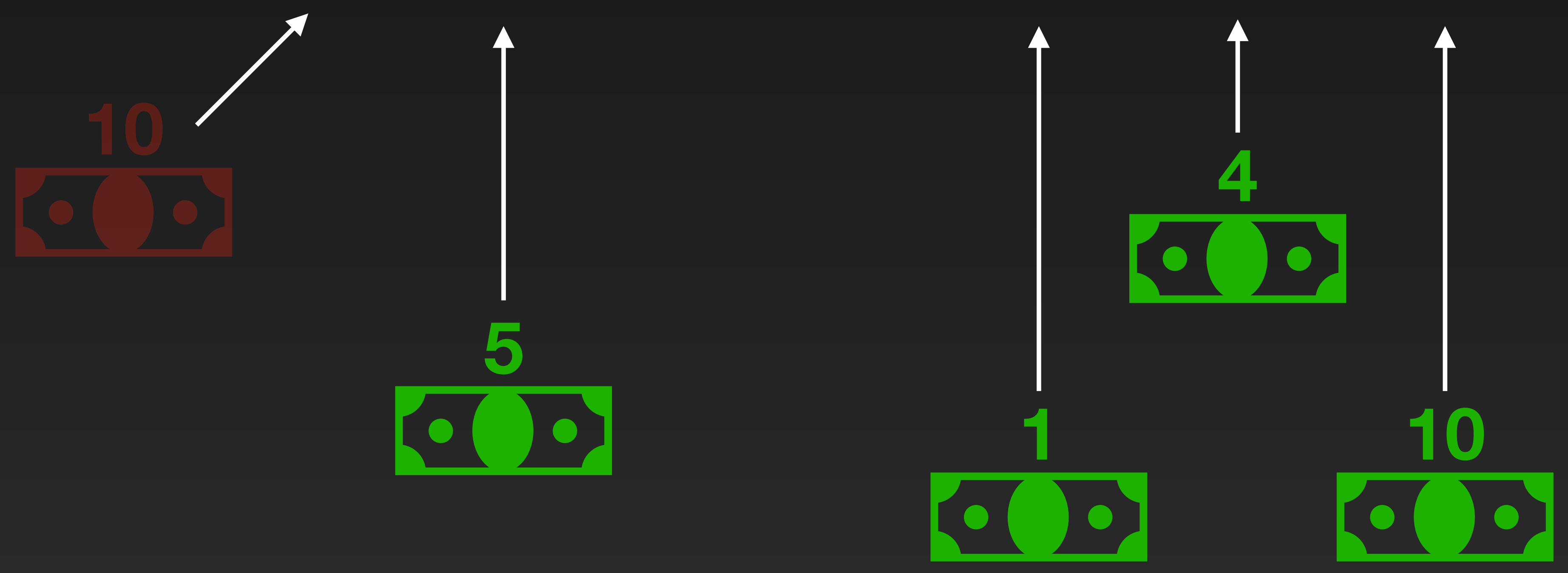
$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



Attack against S-BAC

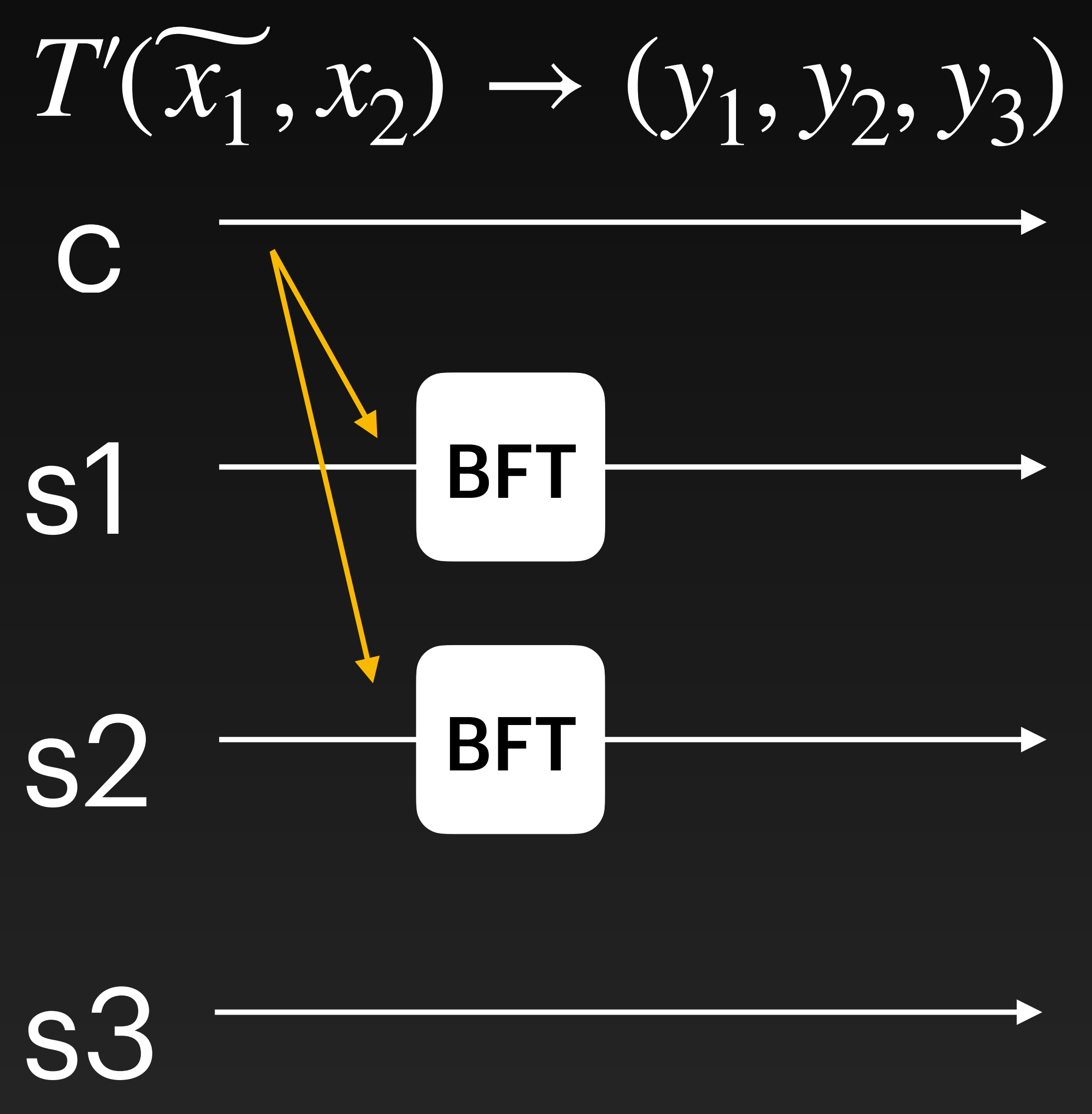
Double-spend X_1

$$T'(\widetilde{x_1}, x_2) \rightarrow (y_1, y_2, y_3)$$



Attack against S-BAC

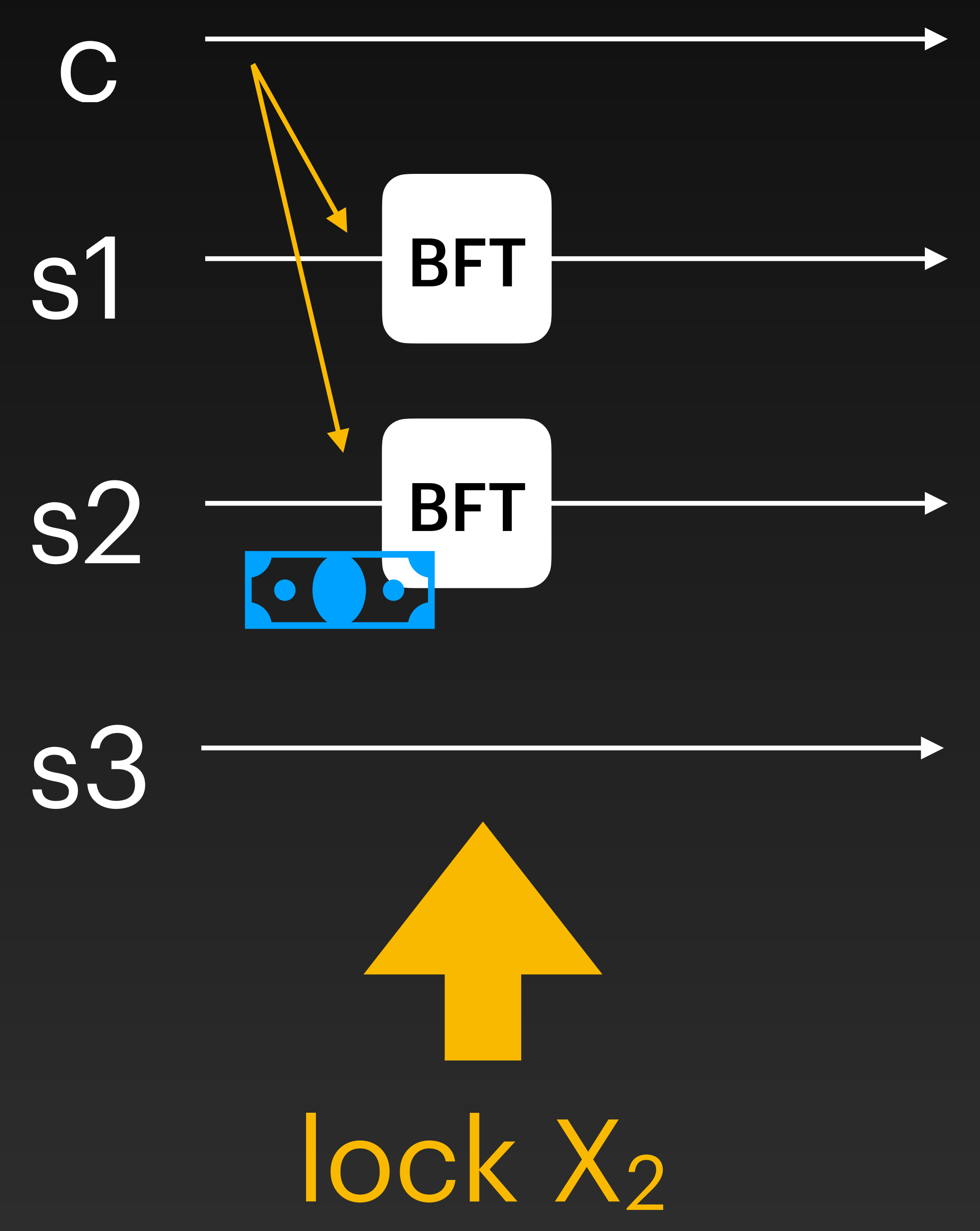
Double-spend X_1



Attack against S-BAC

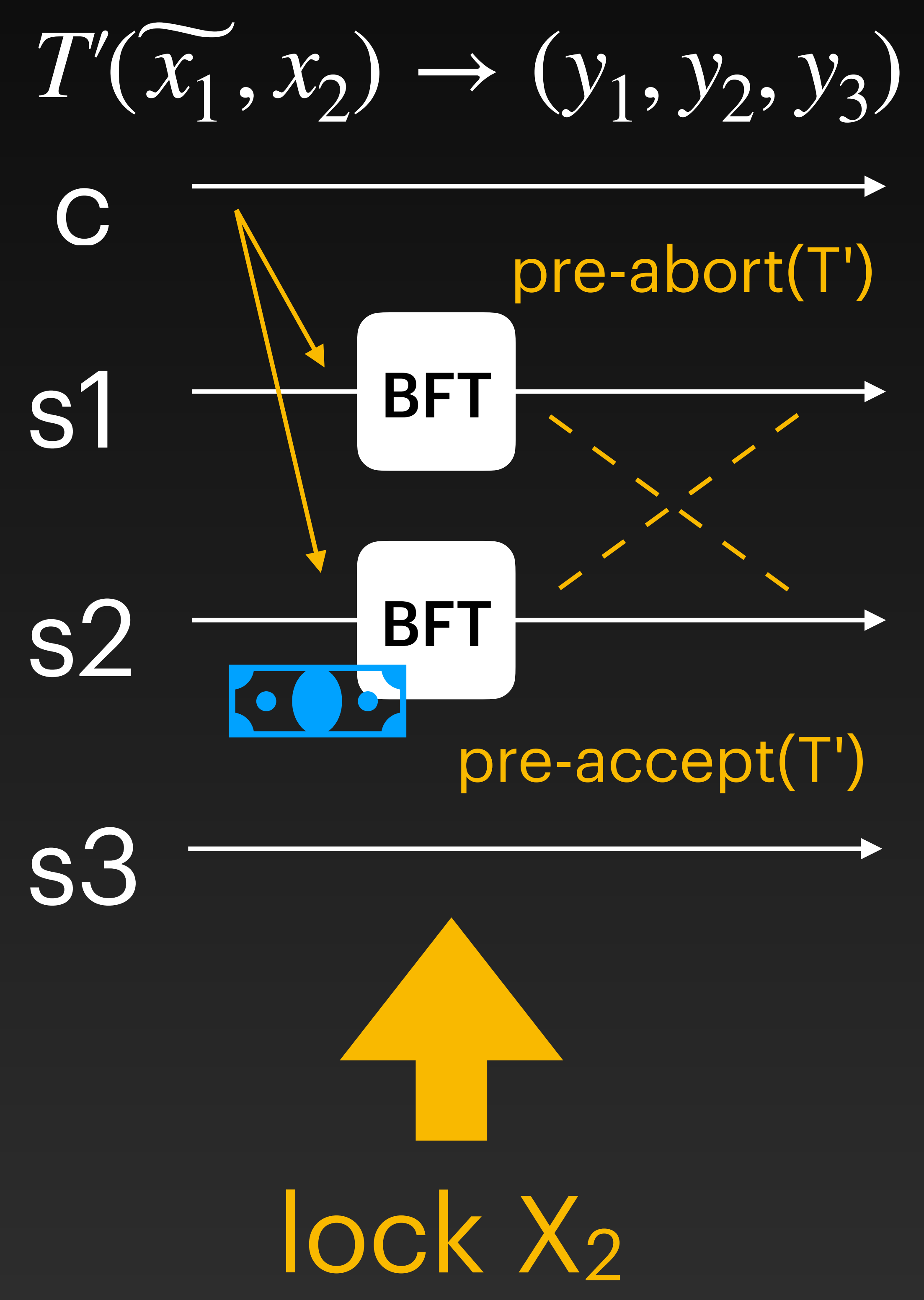
Double-spend X_1

$T'(\widetilde{x_1}, x_2) \rightarrow (y_1, y_2, y_3)$



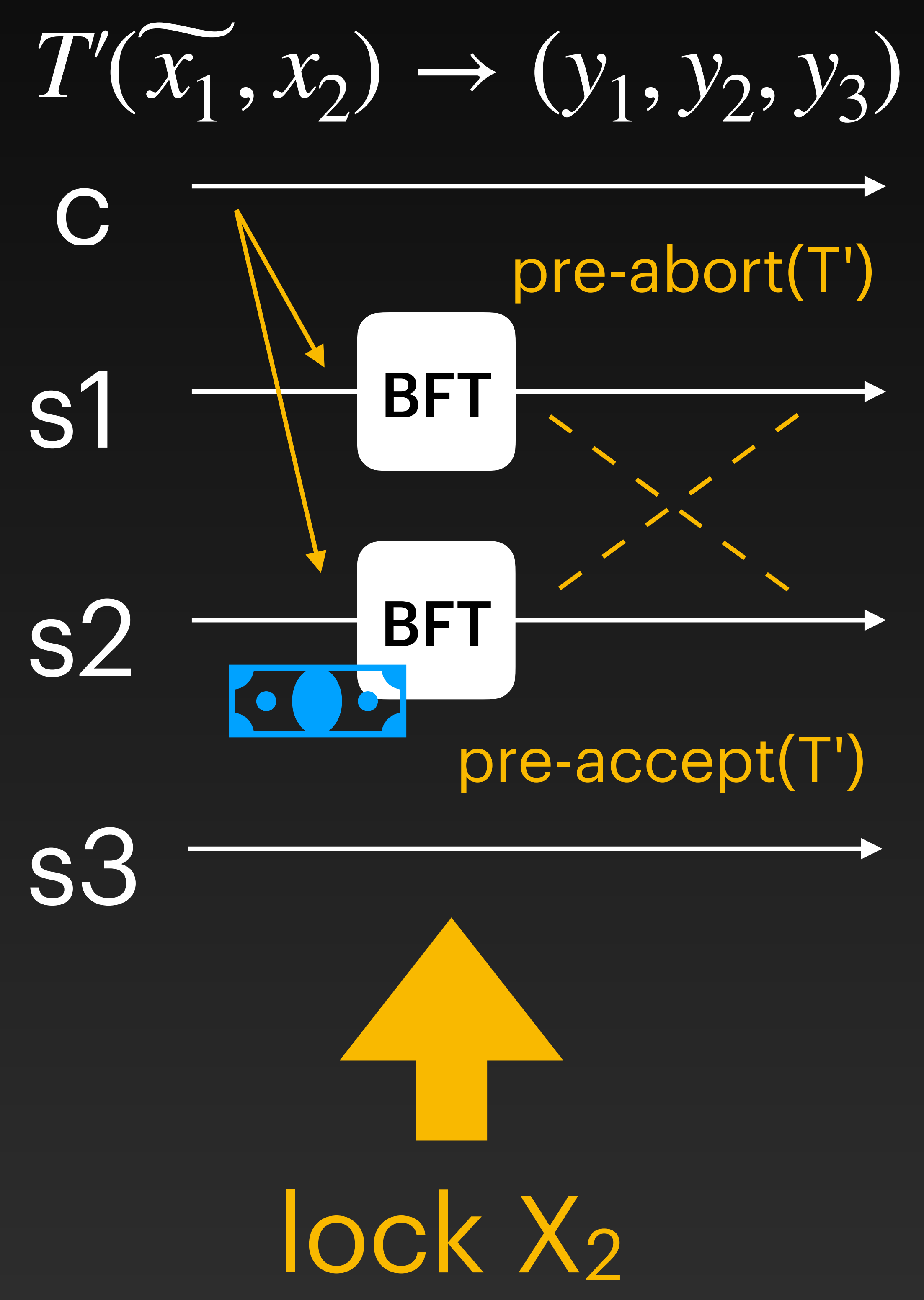
Attack against S-BAC

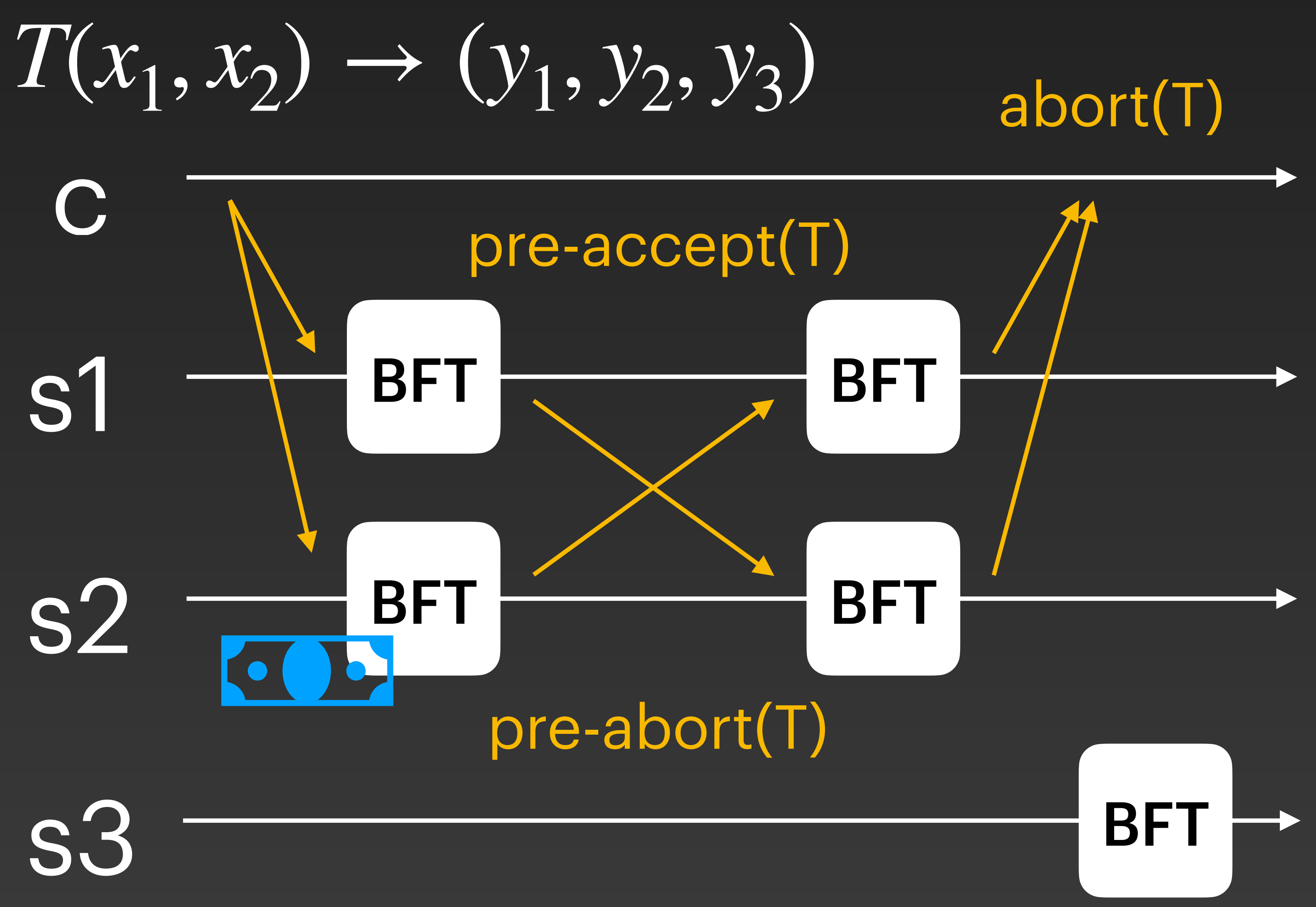
Double-spend X_1



Attack against S-BAC

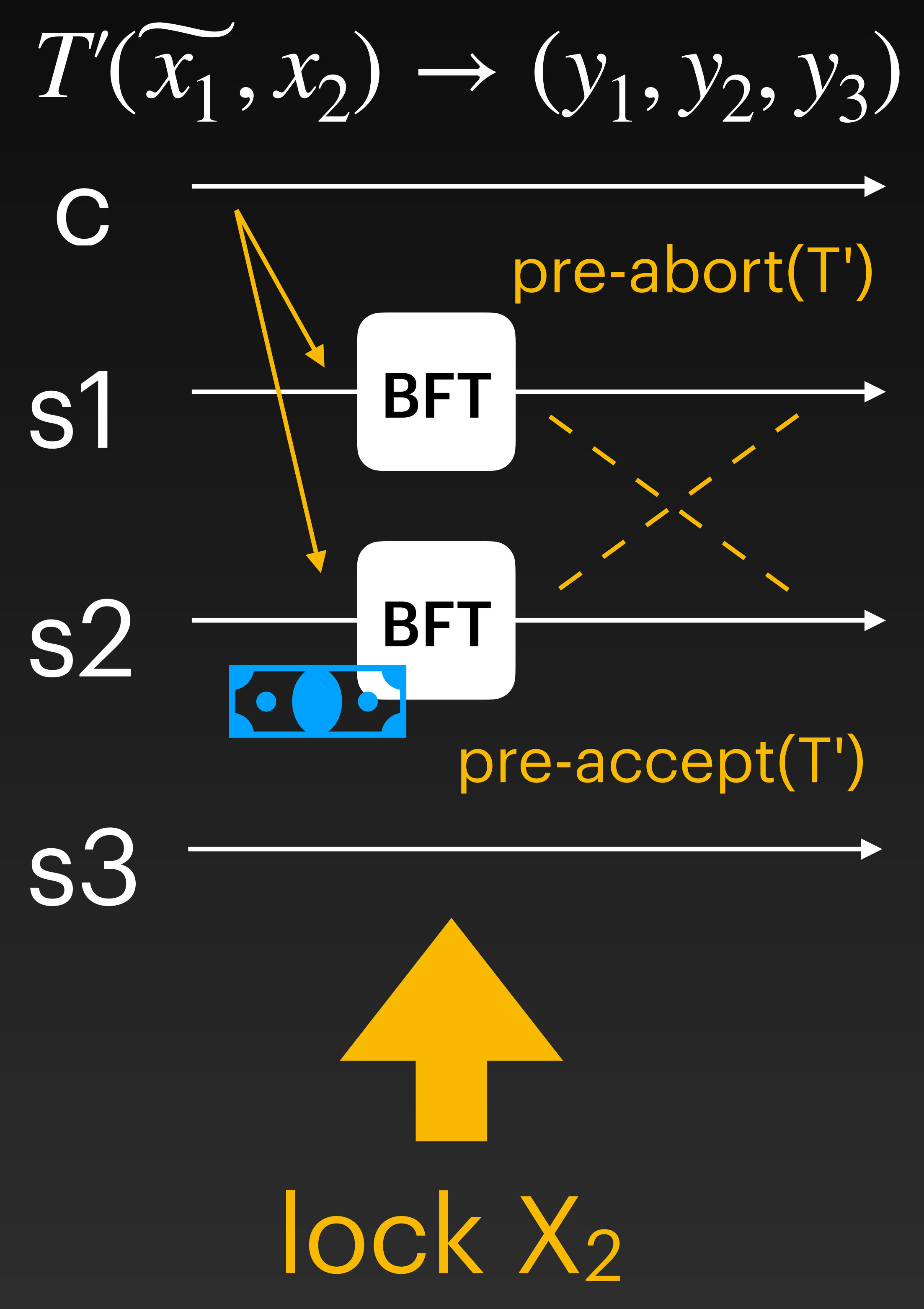
Double-spend X_1

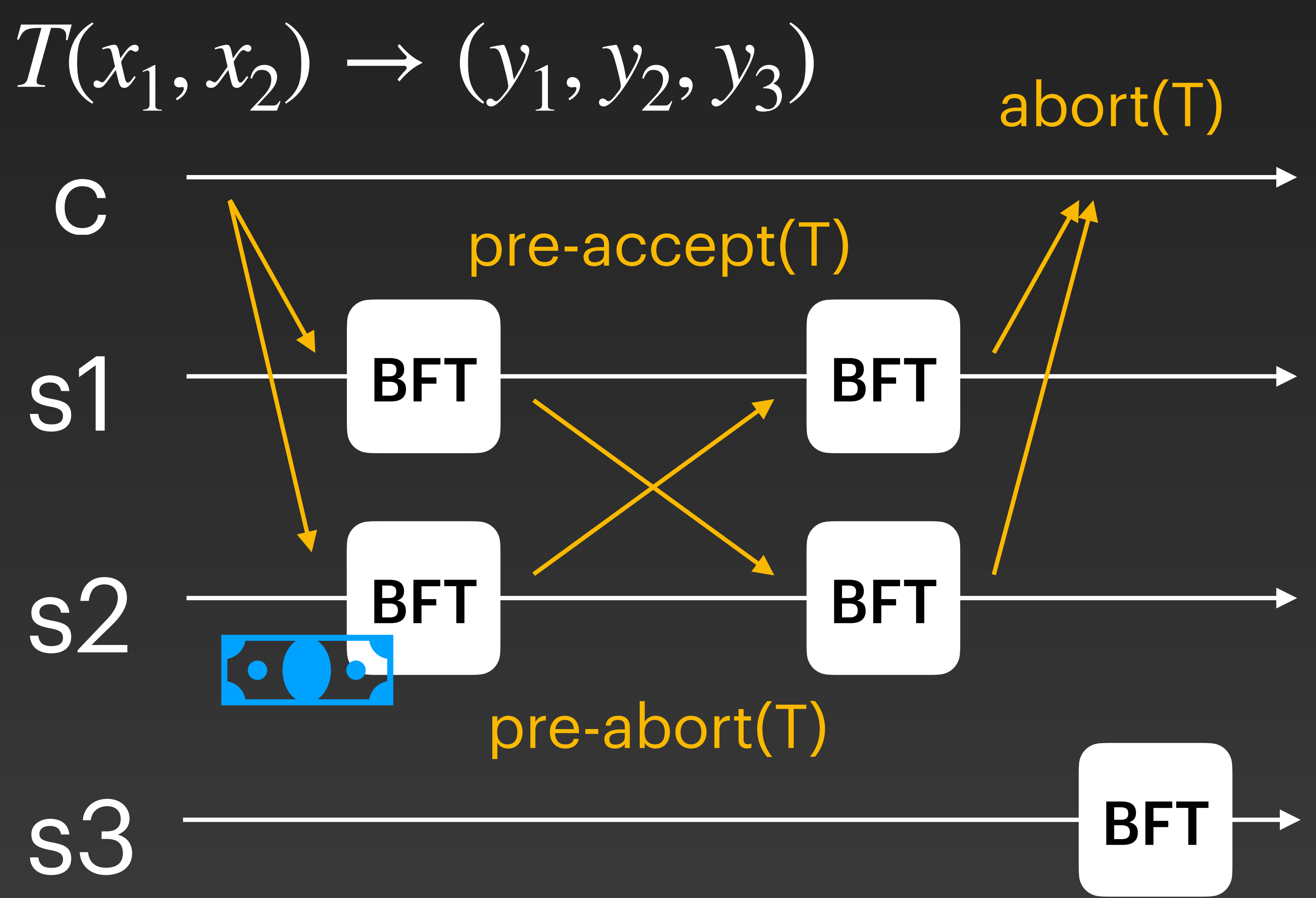




Attack against S-BAC

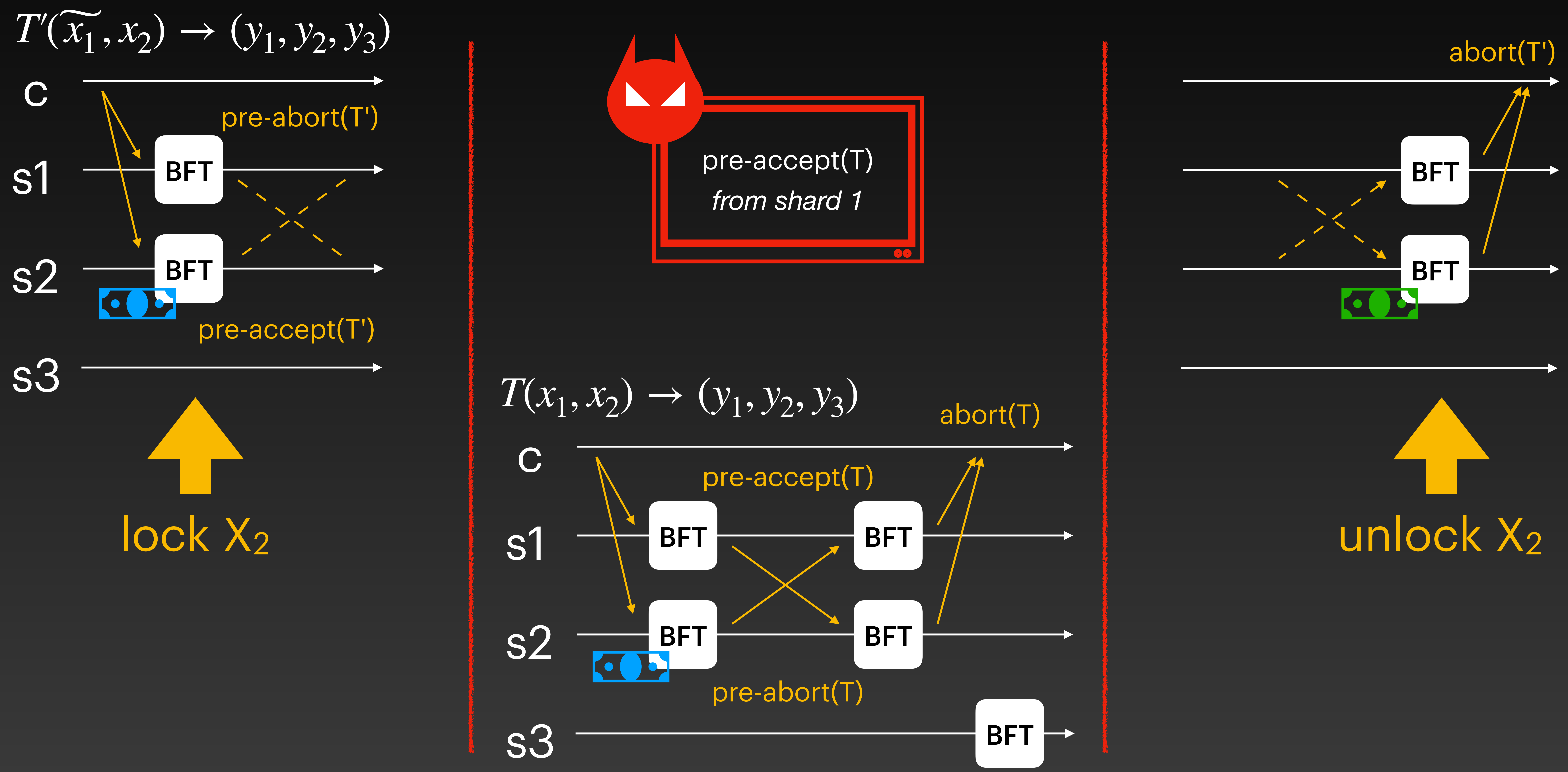
Double-spend X_1





Attack against S-BAC

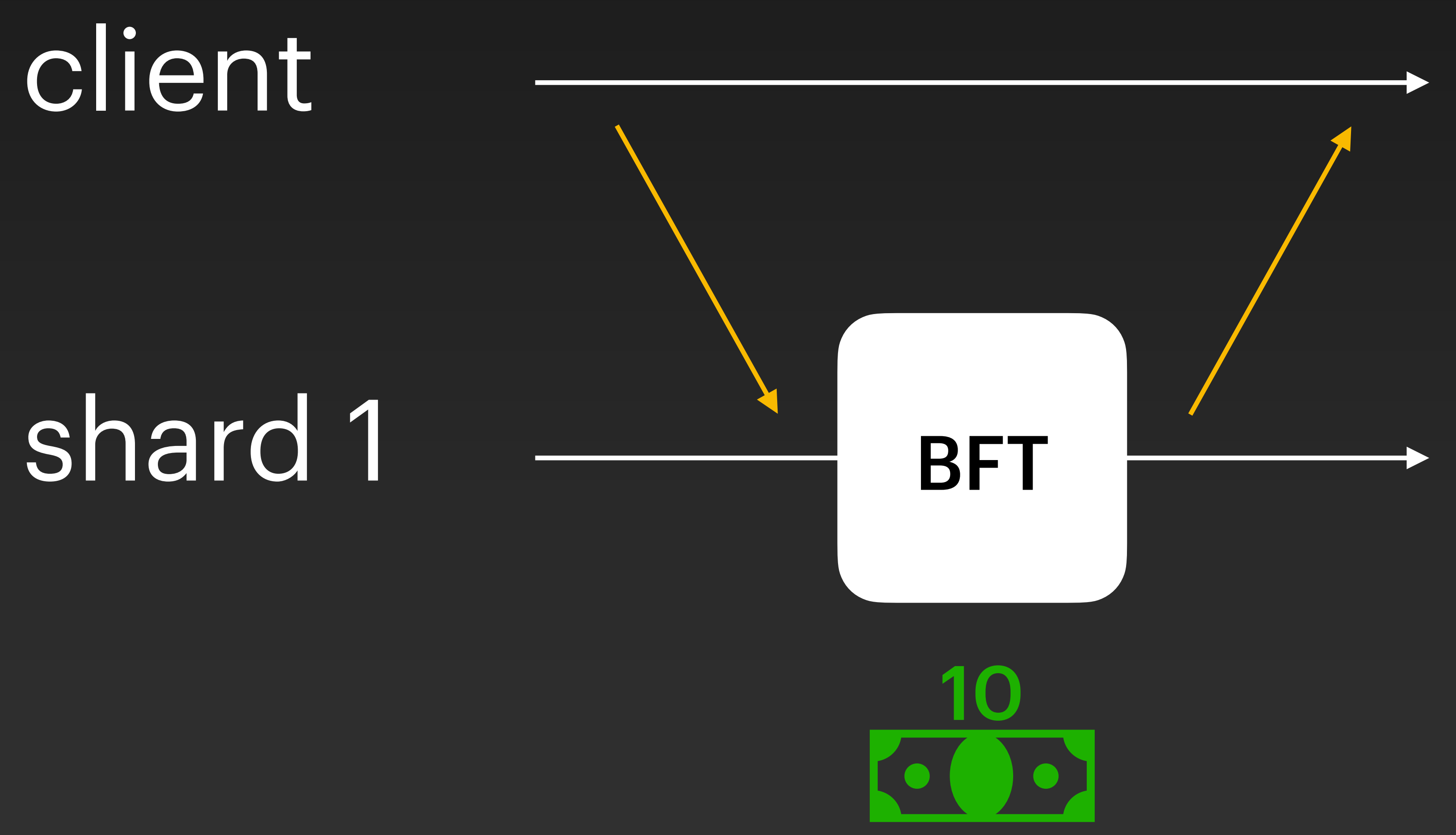
Double-spend X_1



Attack against S-BAC

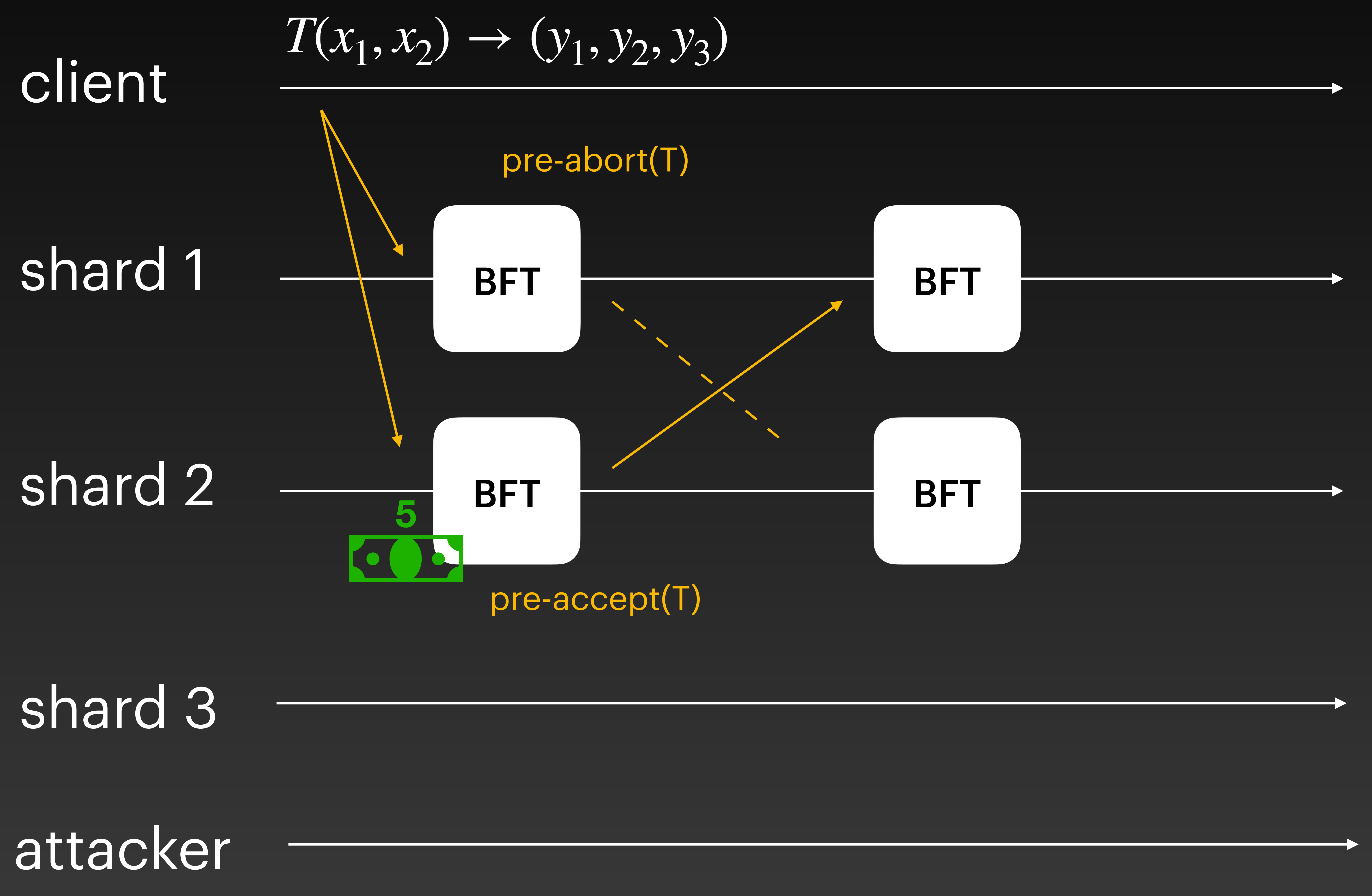
Double-spend X_1

$$T^*(x_1) \rightarrow (y_*)$$



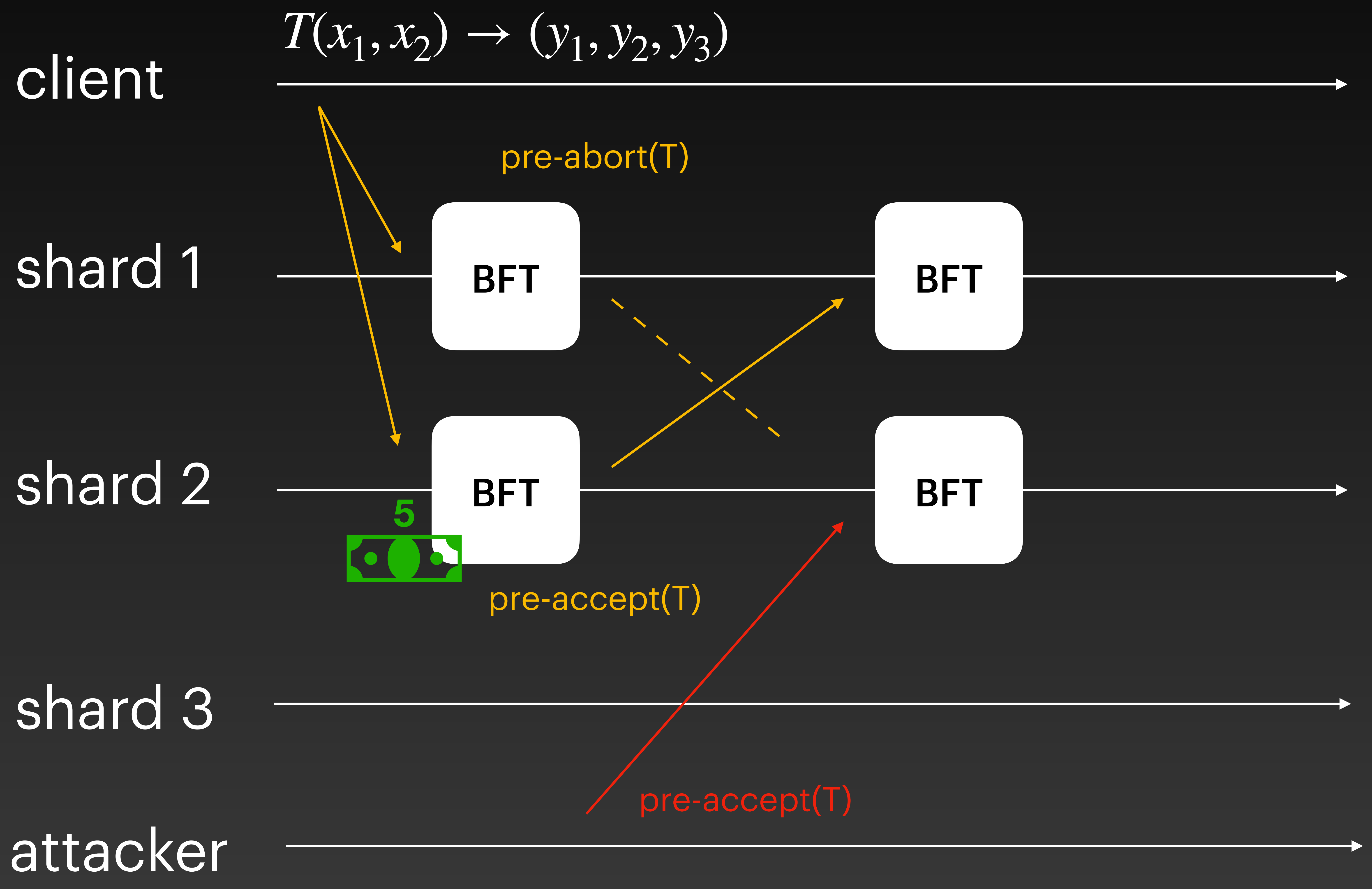
Attack against S-BAC

Double-spend X_1



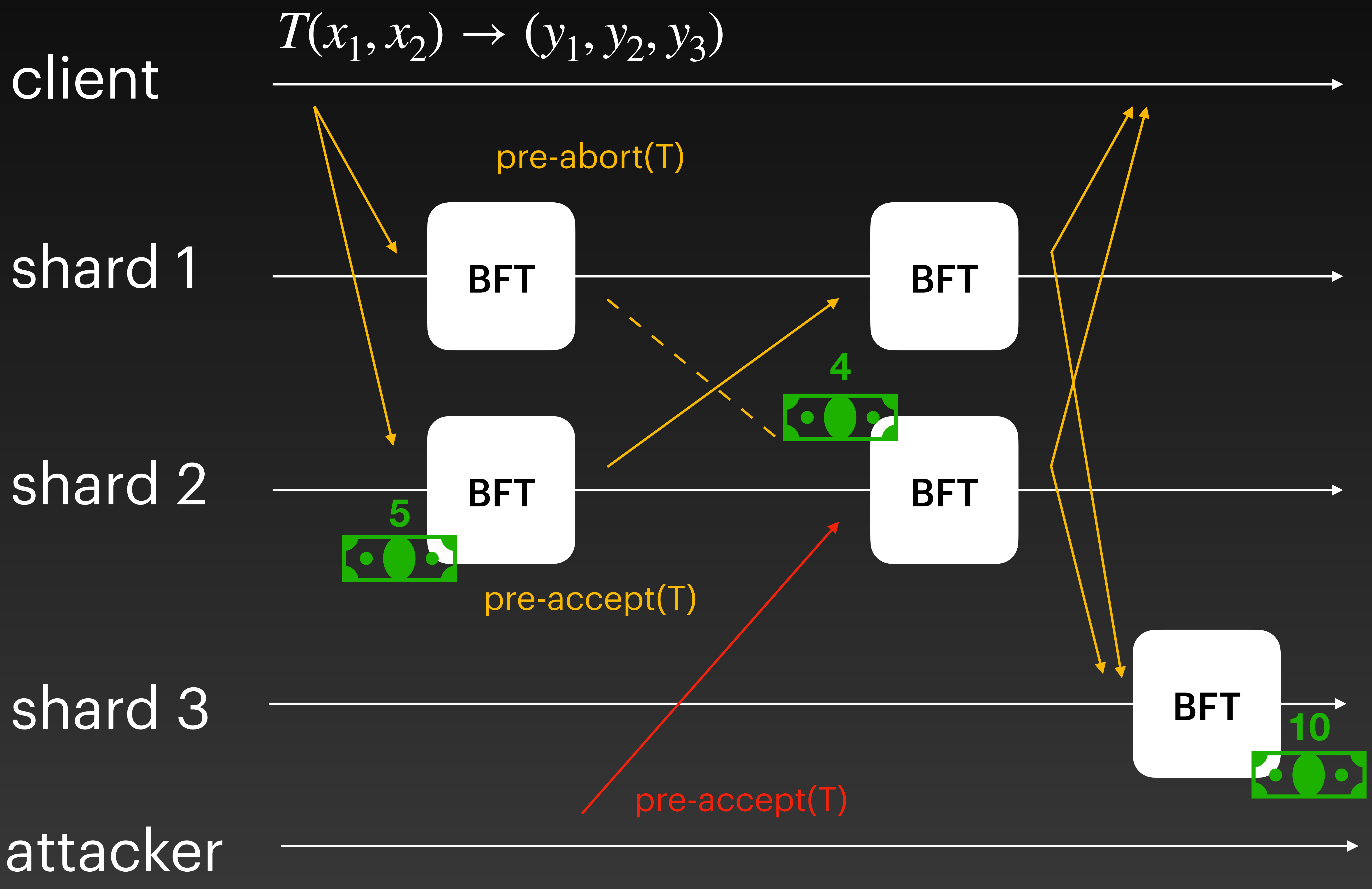
Attack against S-BAC

Double-spend X_1



Attack against S-BAC

Double-spend X_1



Attack against S-BAC

Double-spend X_1

Before attack

- X_1 — 10
- X_2 — 5

After attack

- Y_* — 10
- Y_2 — 4
- Y_3 — 10

What causes these issues?

Issue 1. Input shards cannot associate protocol messages to a specific protocol execution.

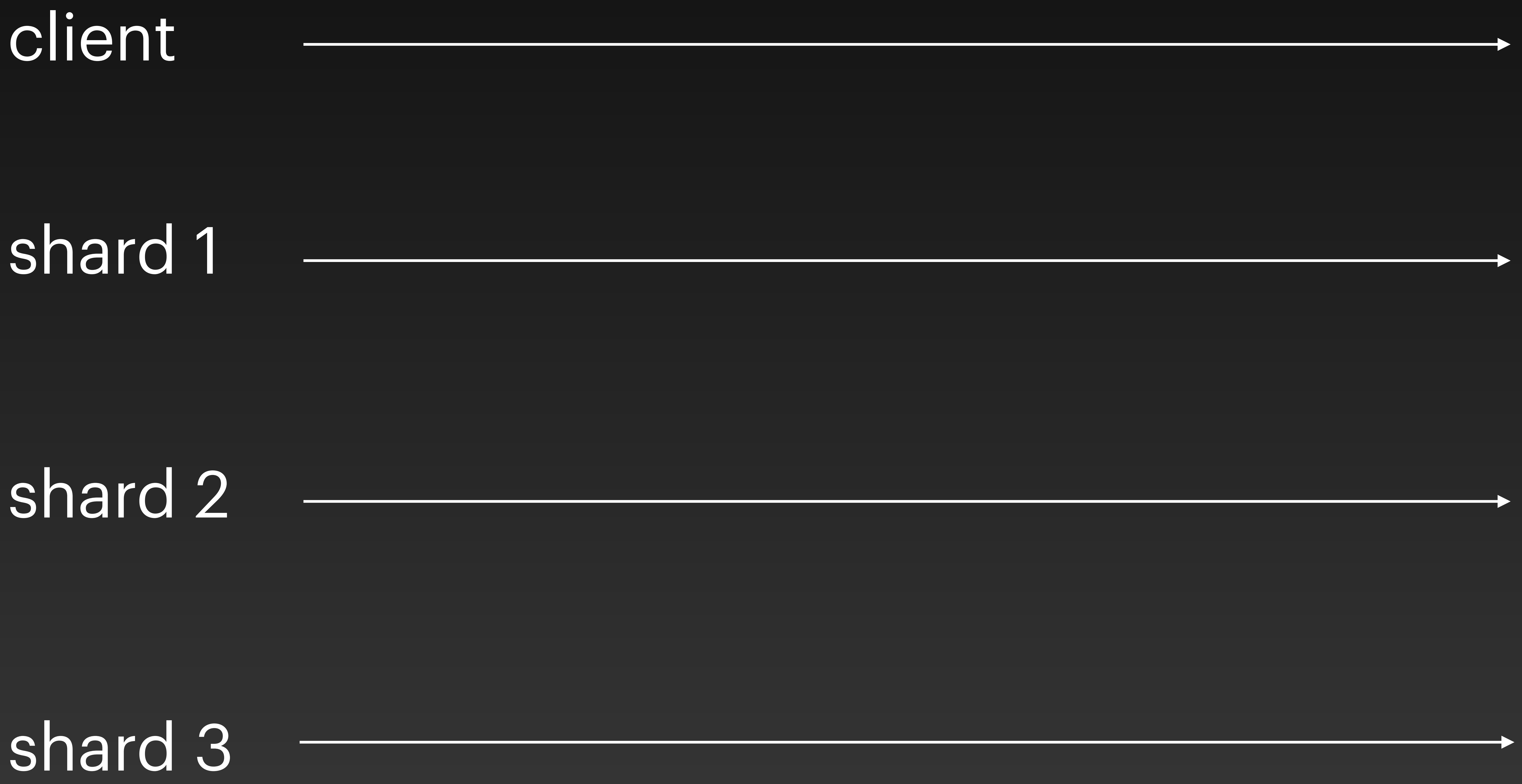
Issue 2. Output shards (that are not also input shards) do not experience the first phase of the protocol

EXTRA

Atomix Attacks

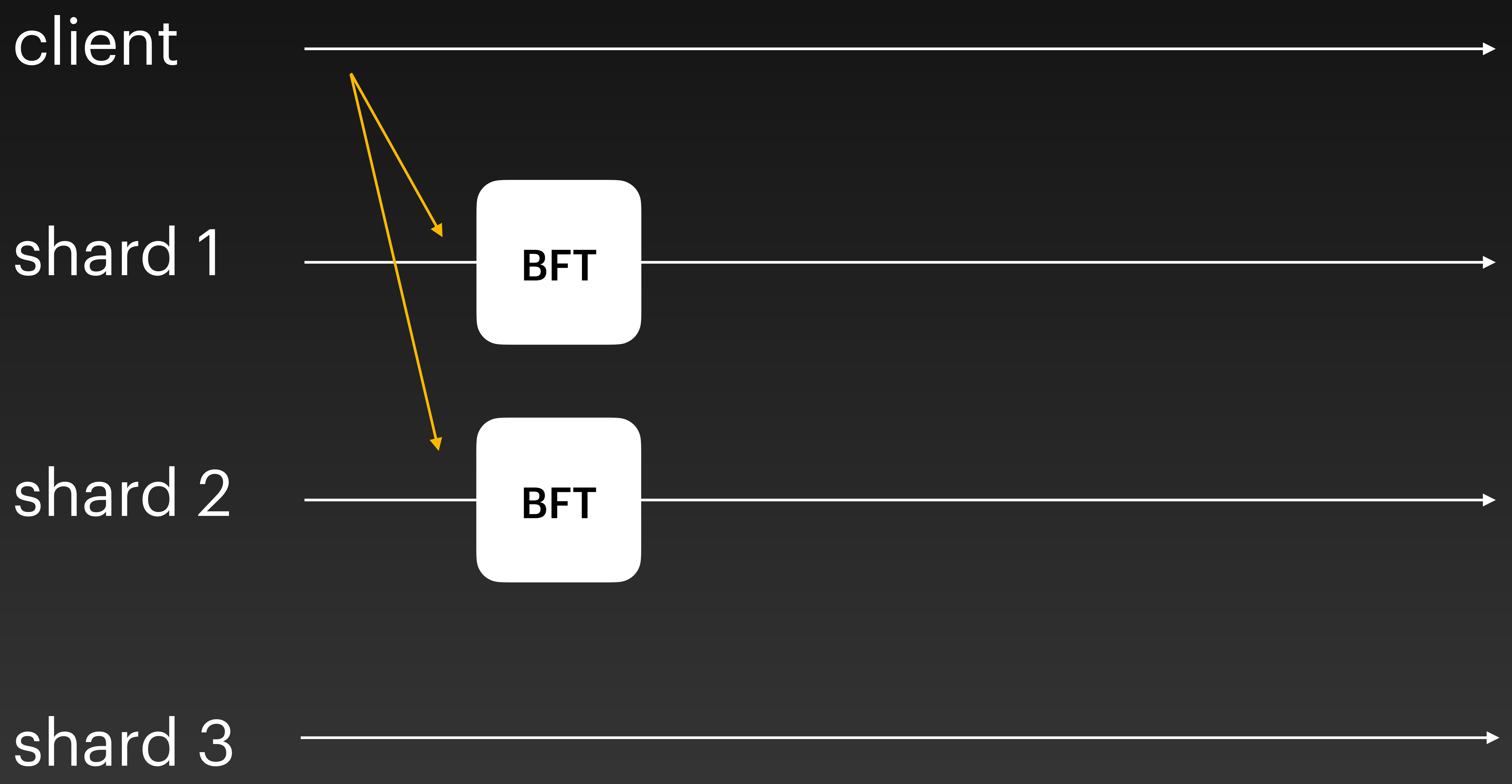
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



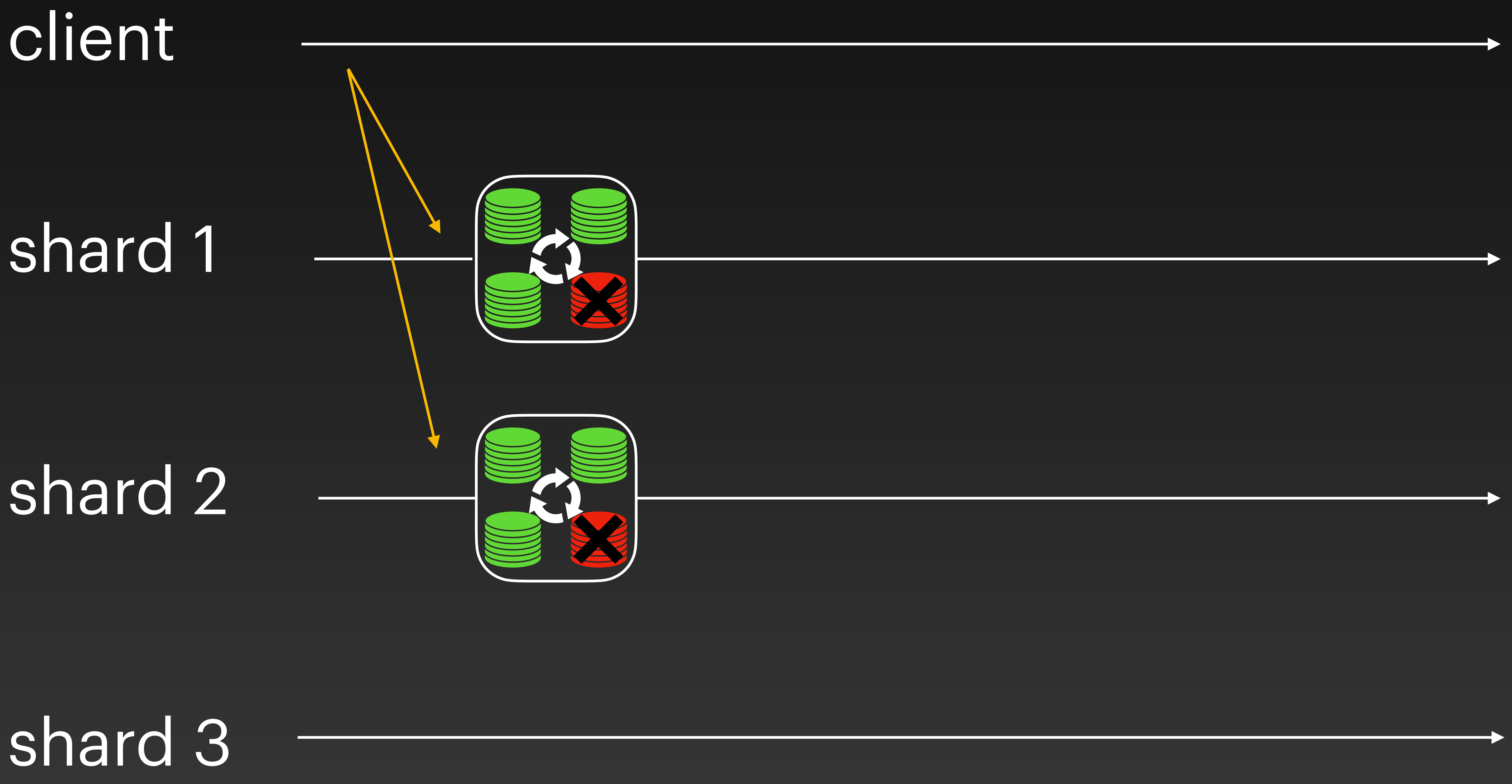
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



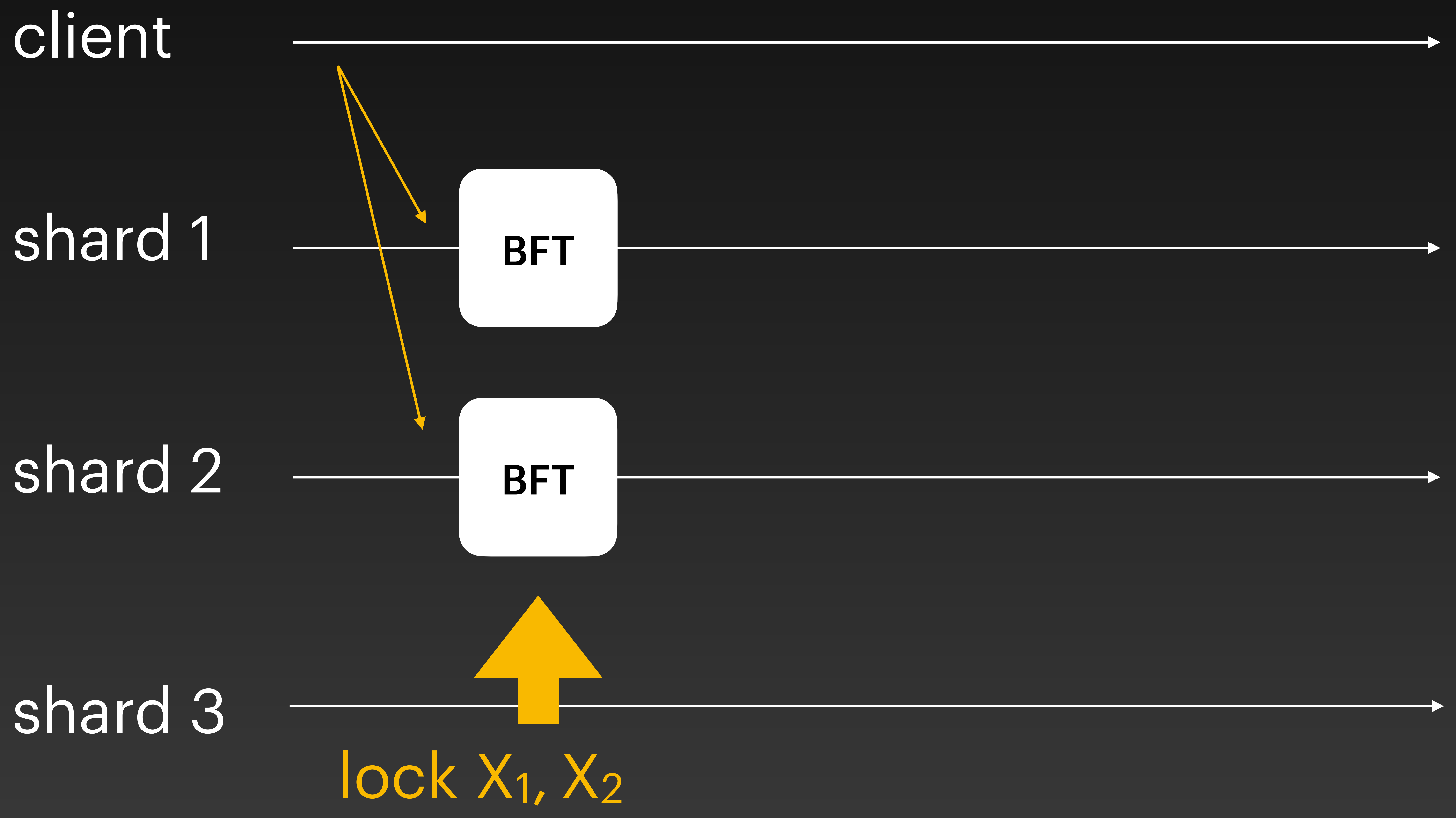
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



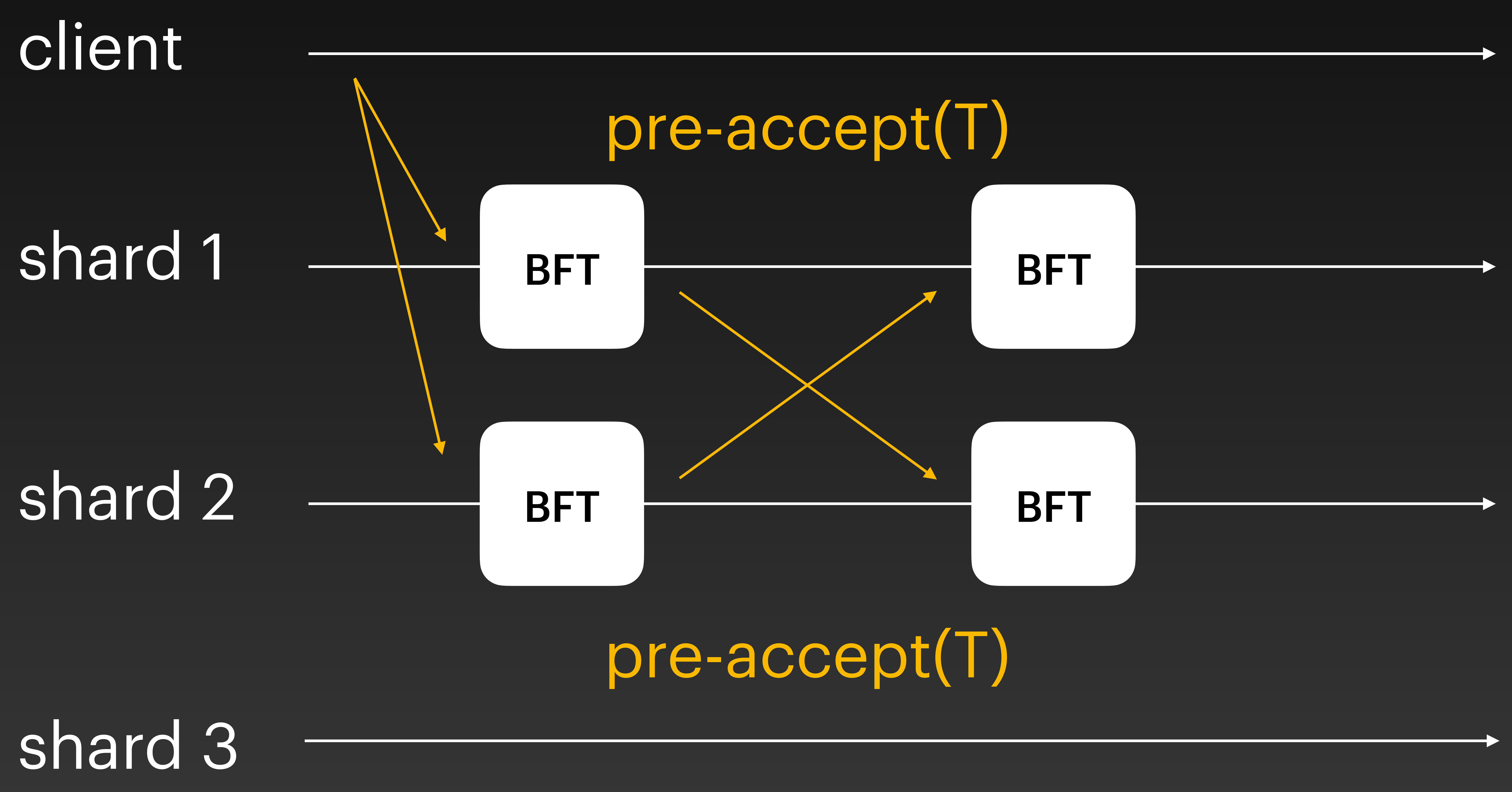
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



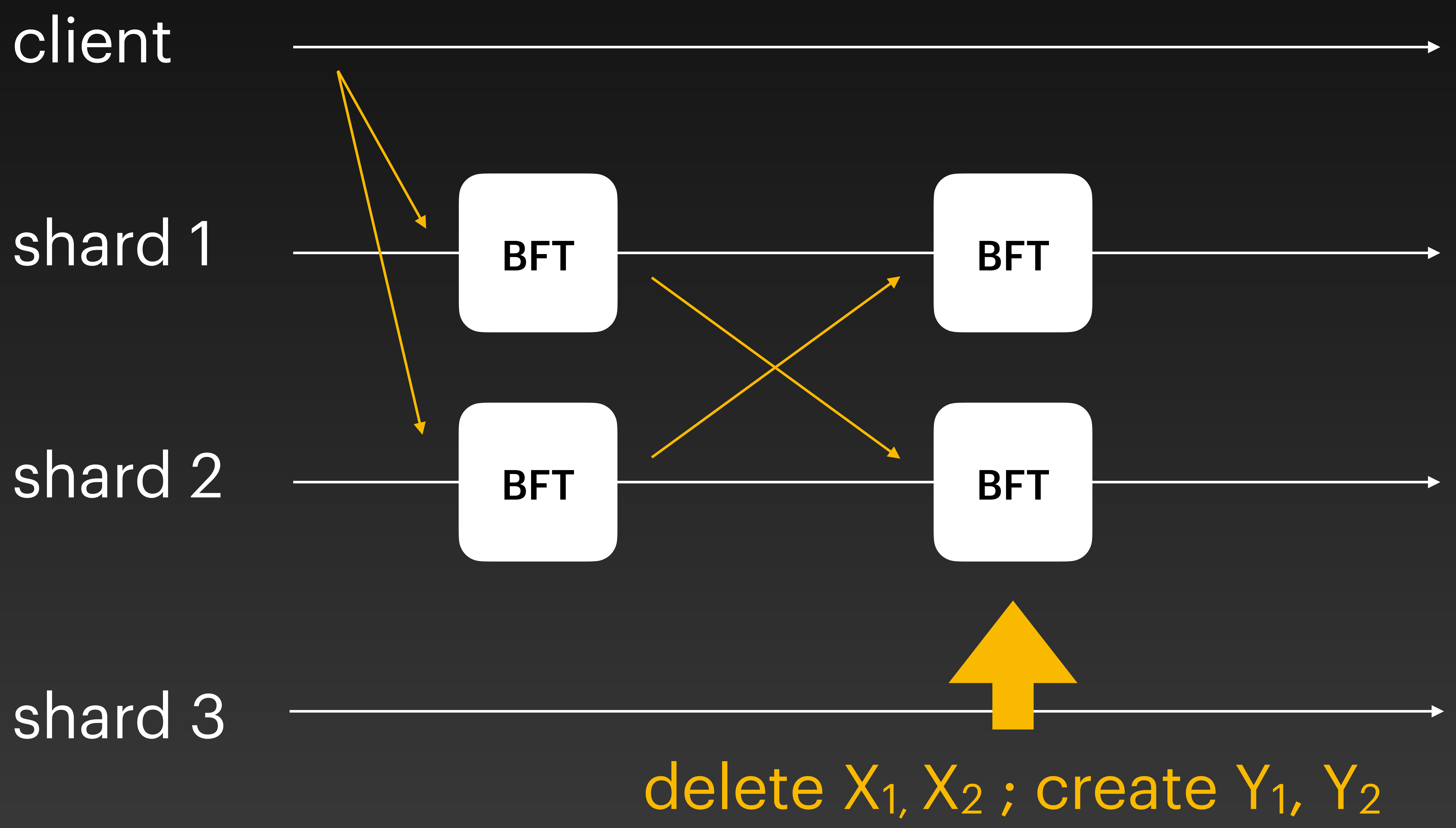
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



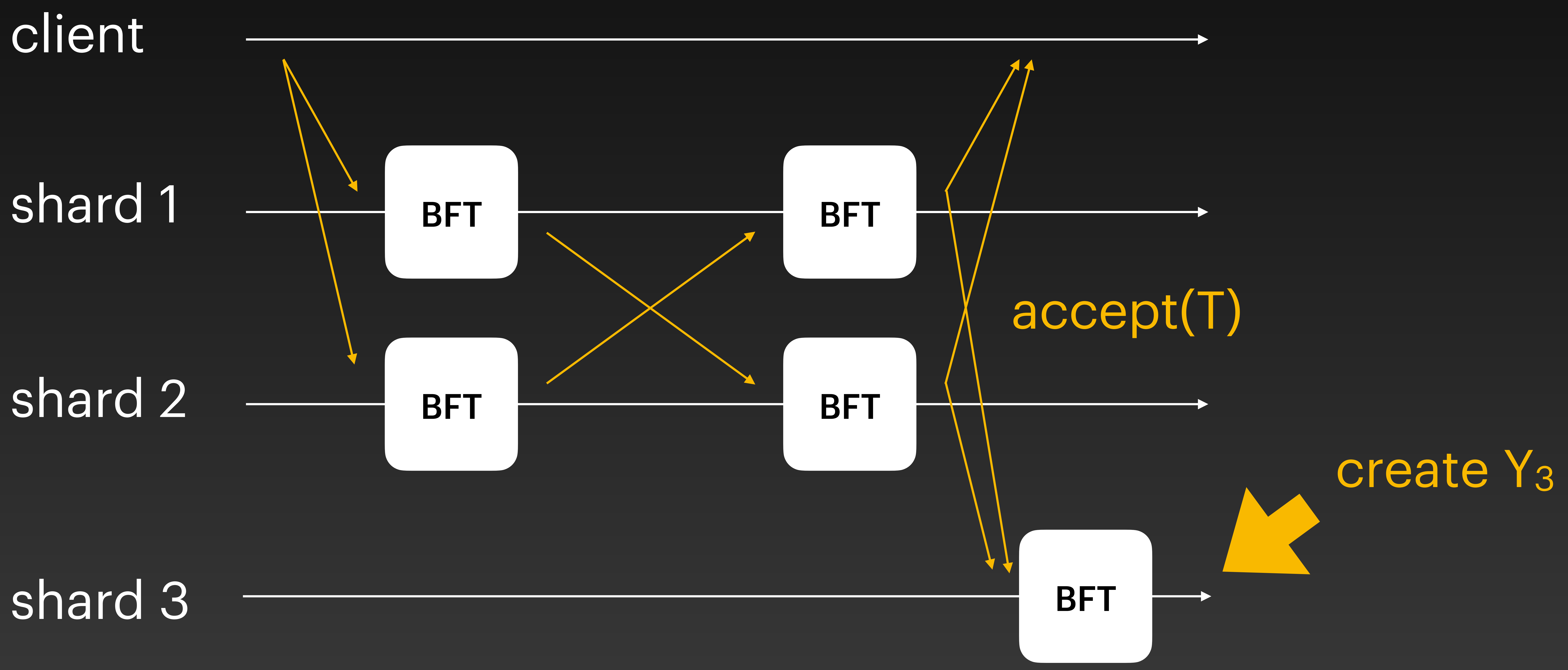
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



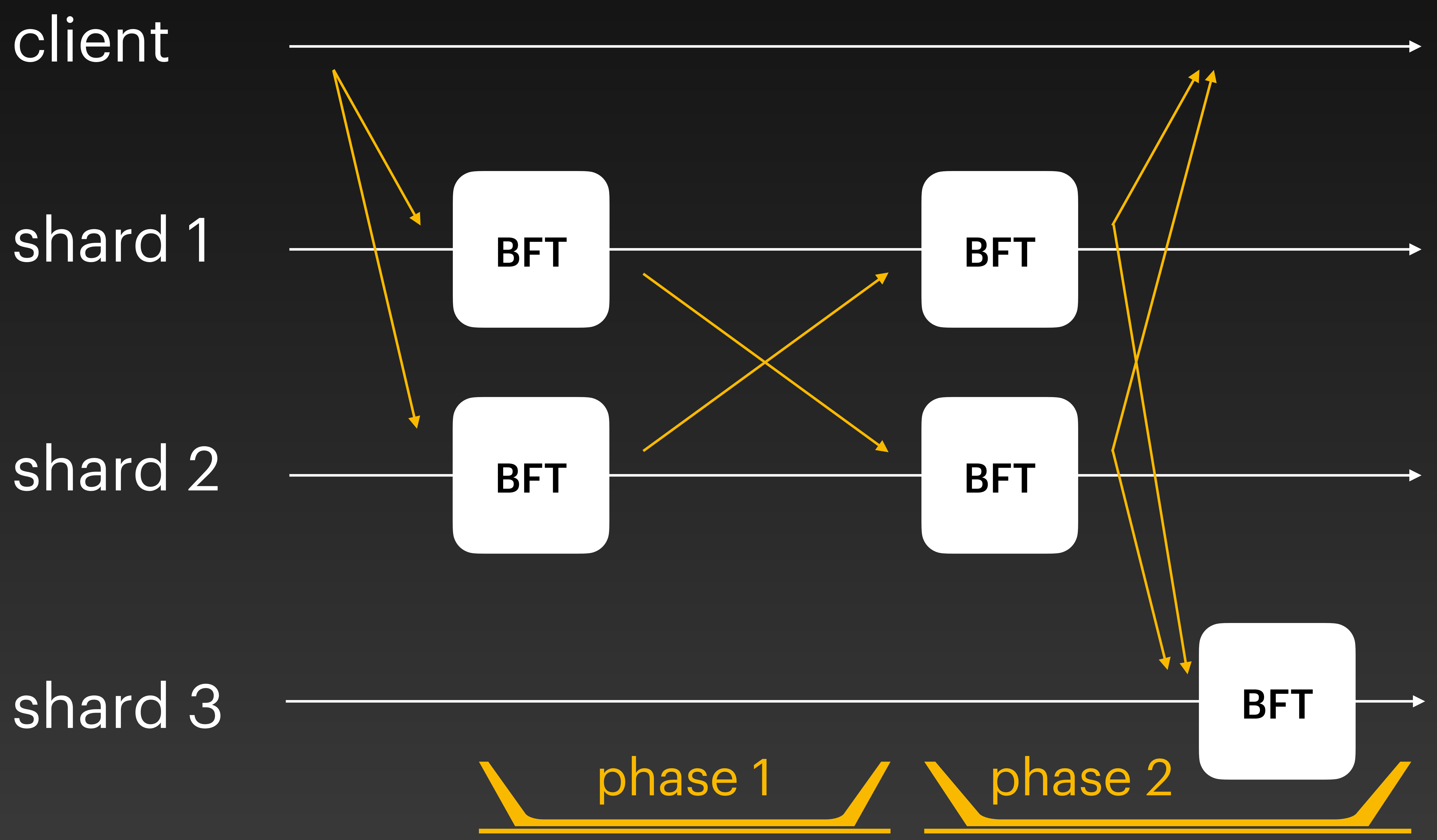
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



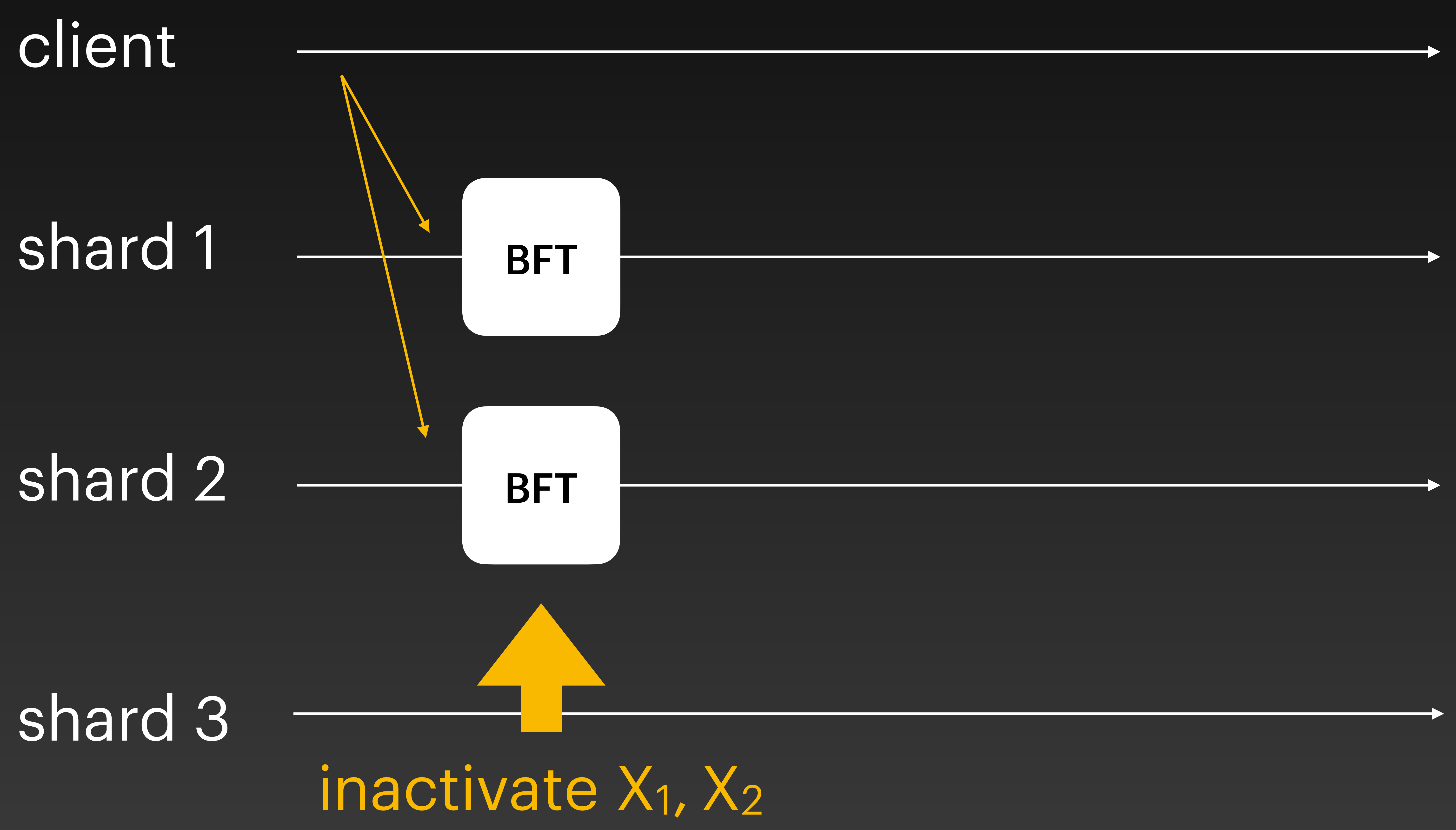
S-BAC

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



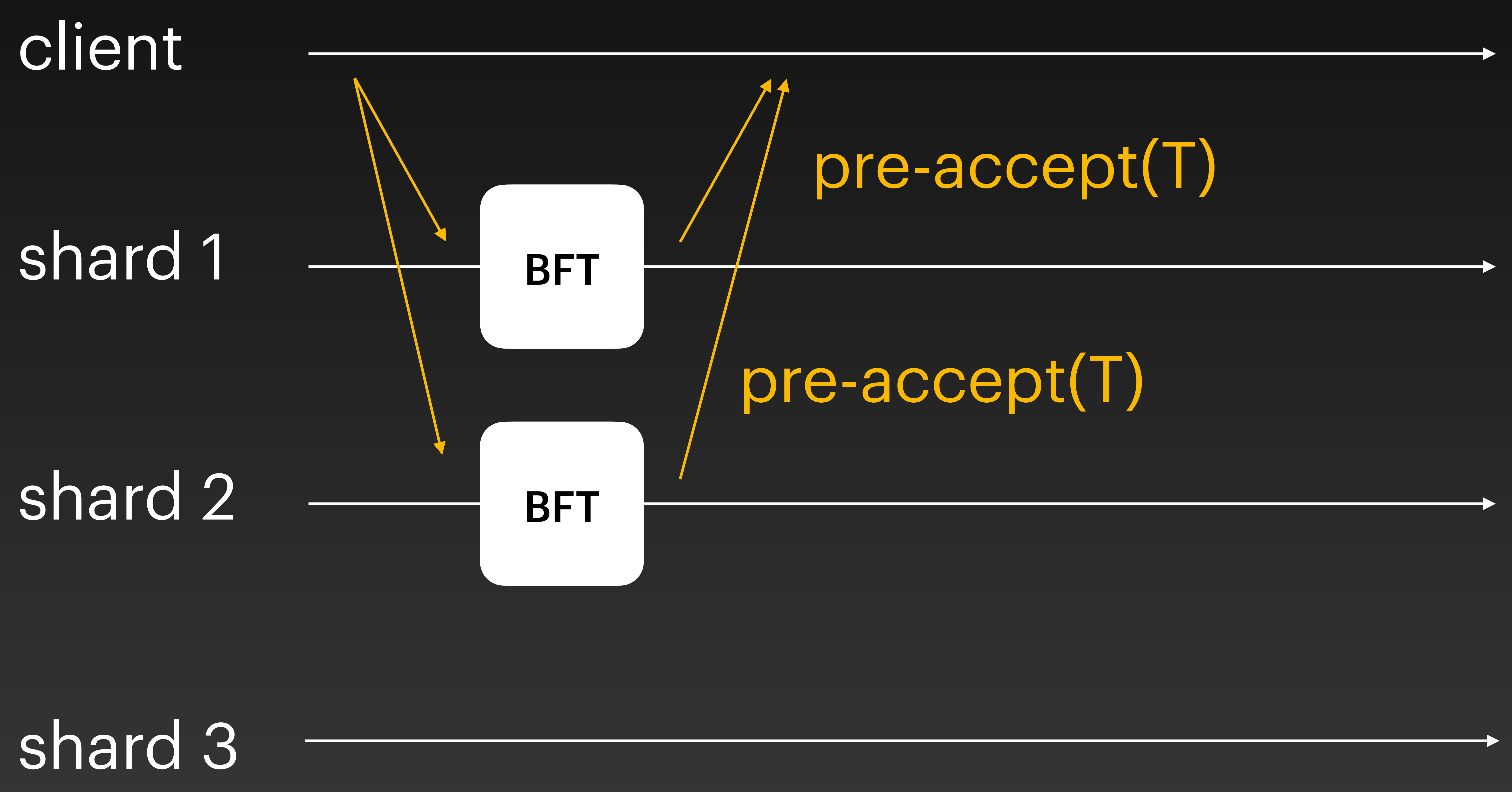
Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



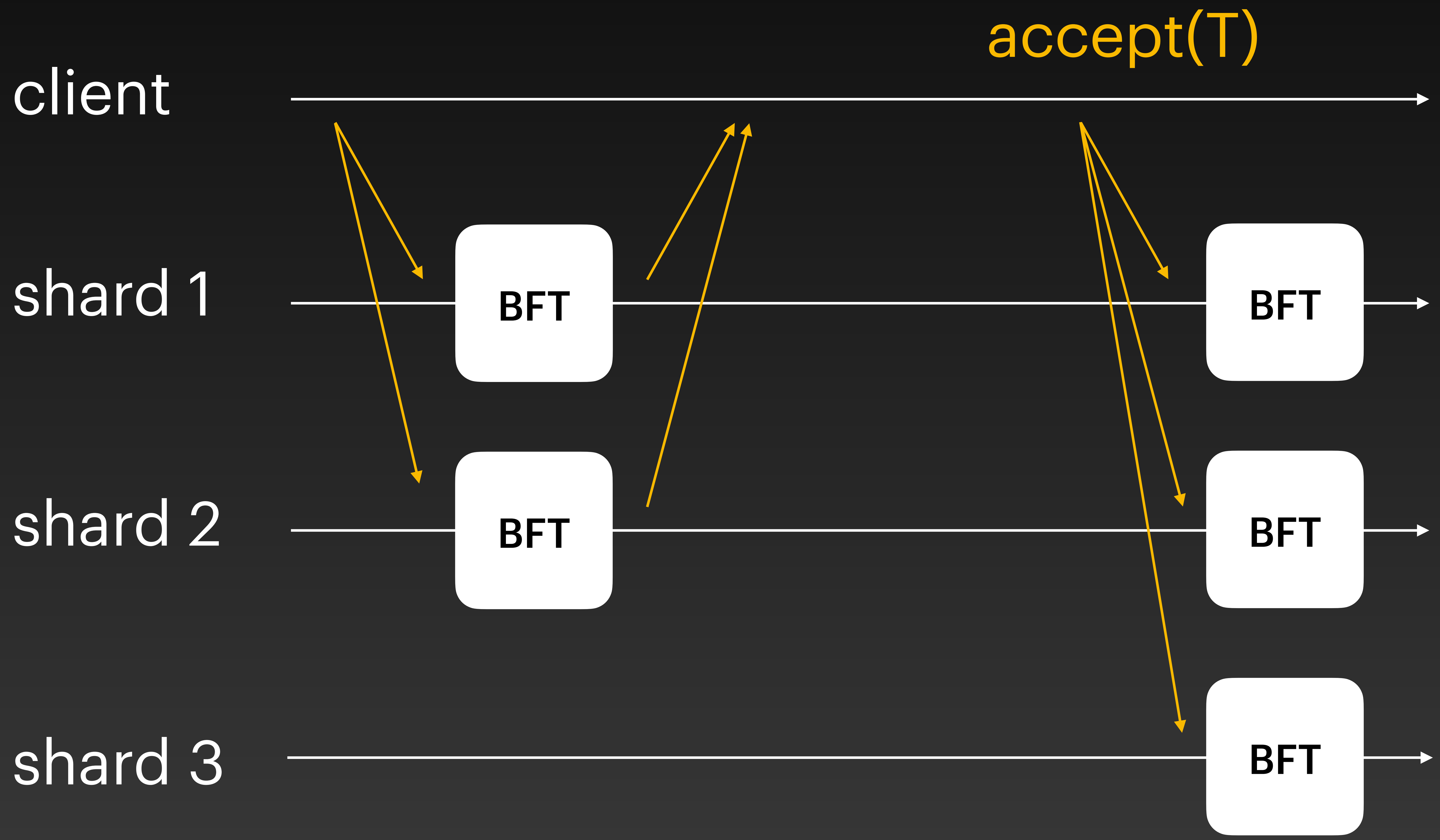
Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



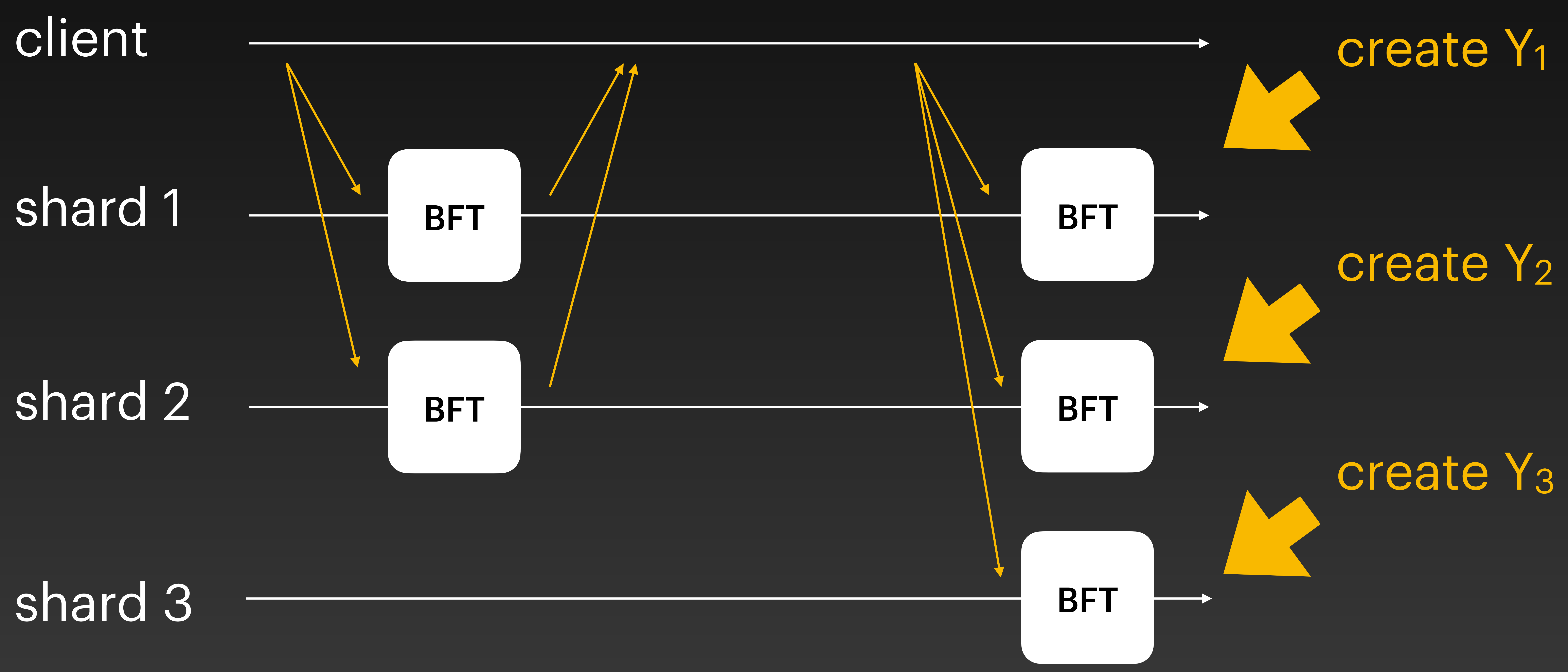
Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



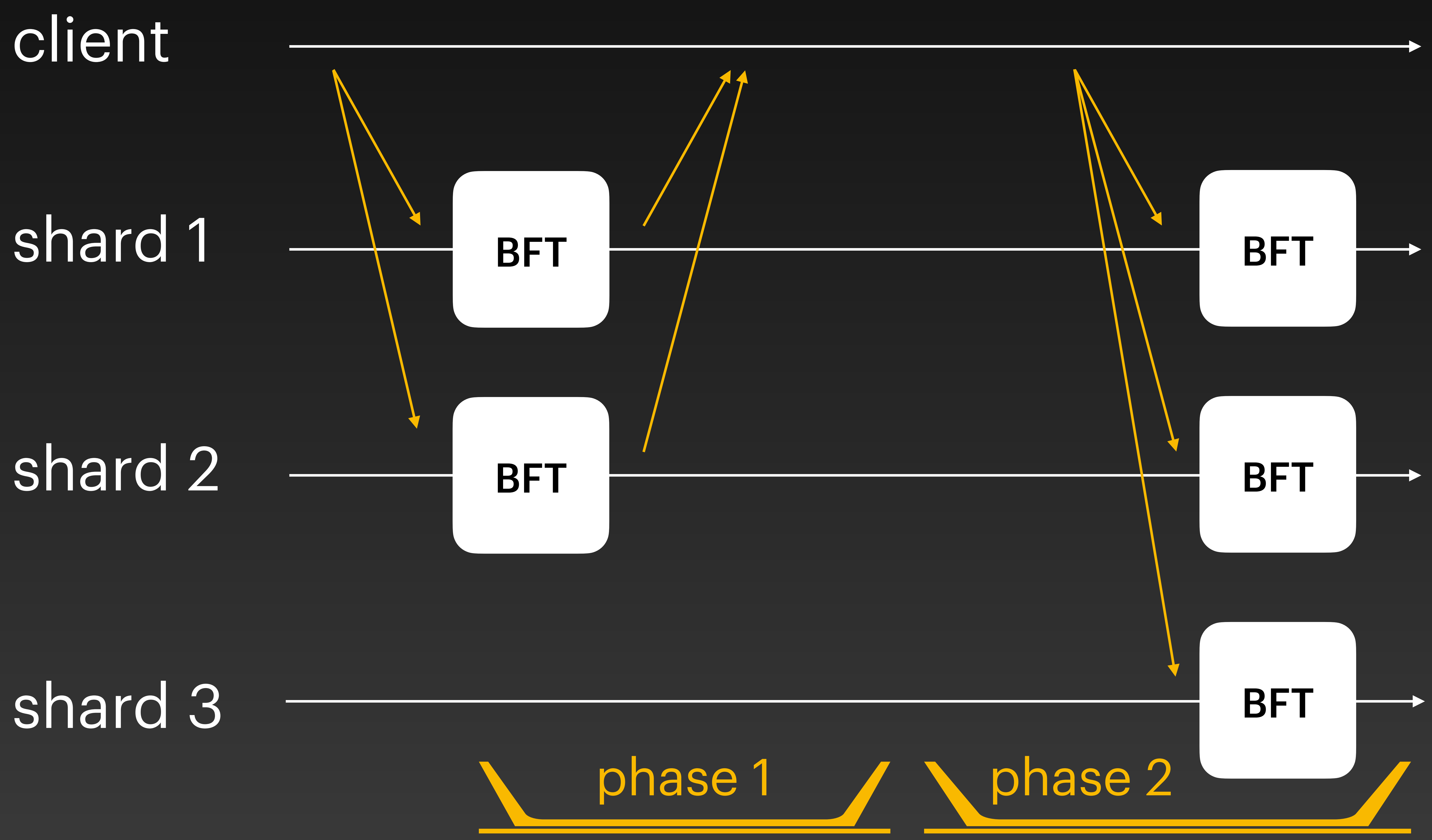
Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



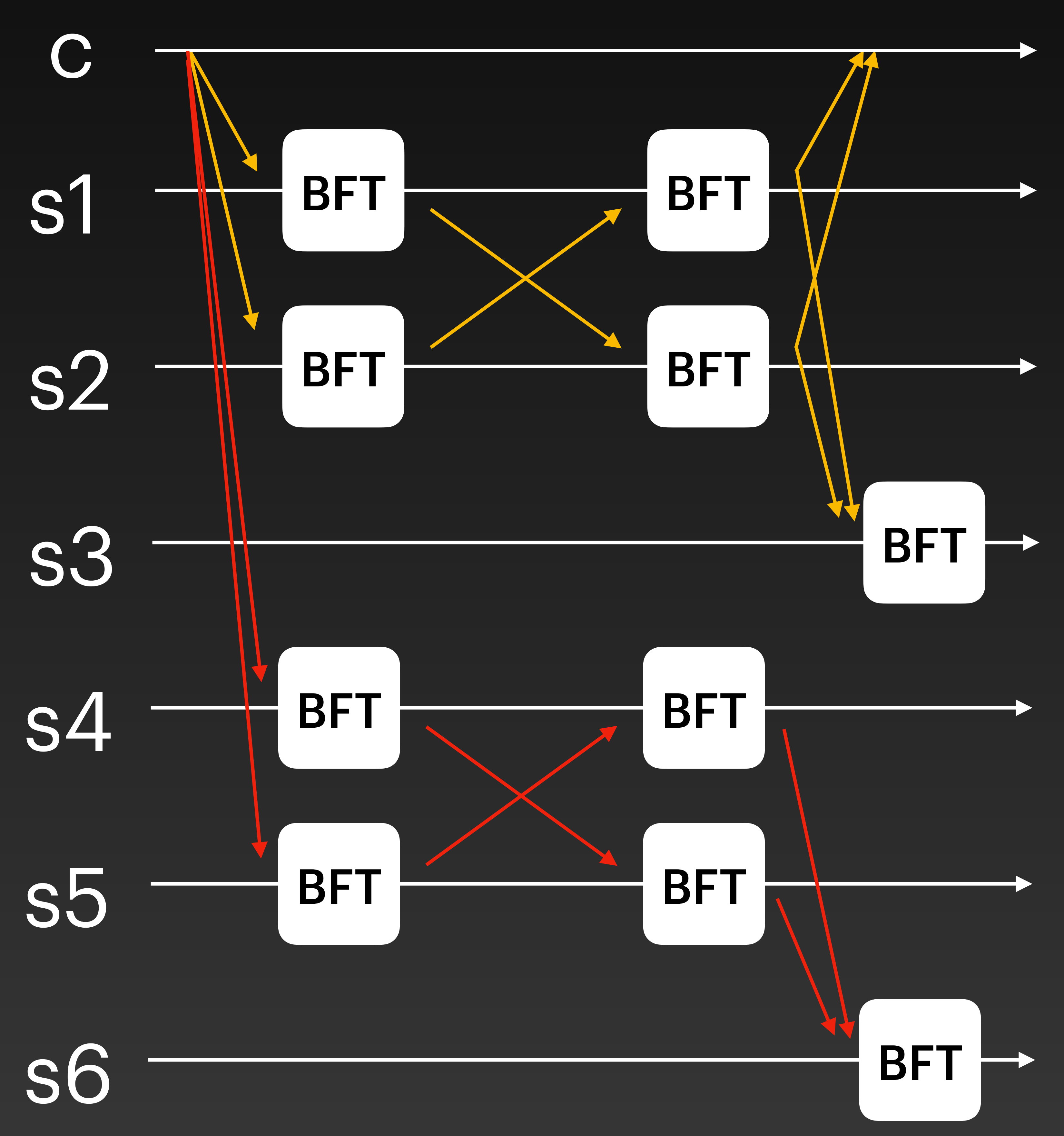
Atomix

$$T(x_1, x_2) \rightarrow (y_1, y_2, y_3)$$



Cross-Shard Consensus

How does it achieve linear scalability?



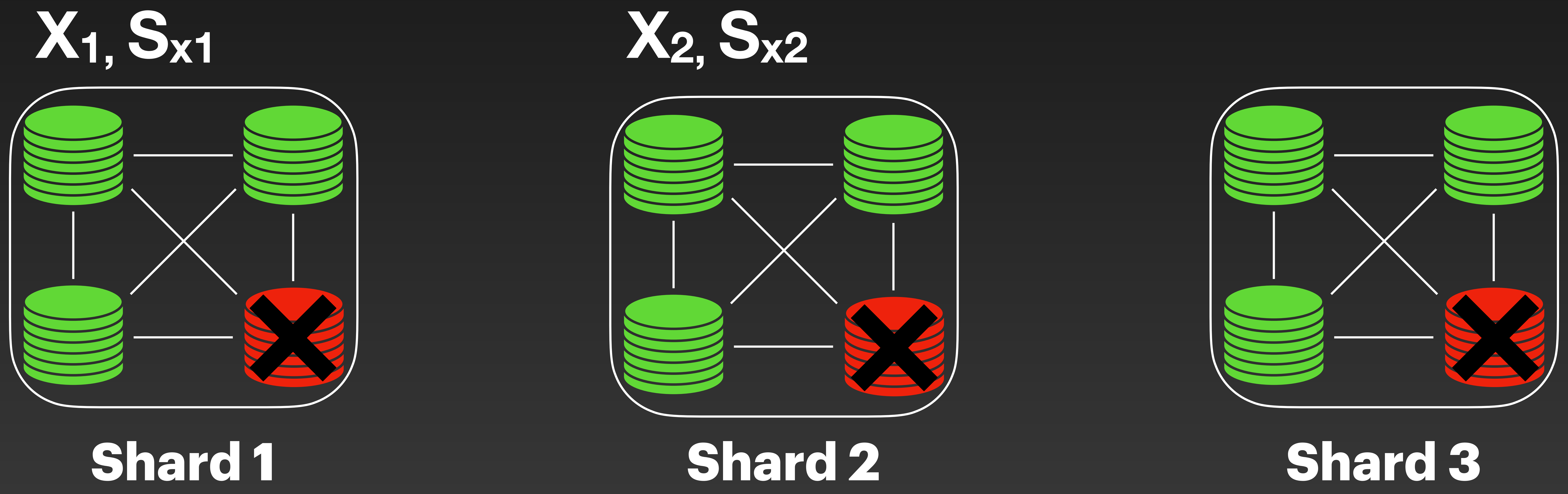
EXTRA

Byzcuit

Byzcuit

Fix issue 1

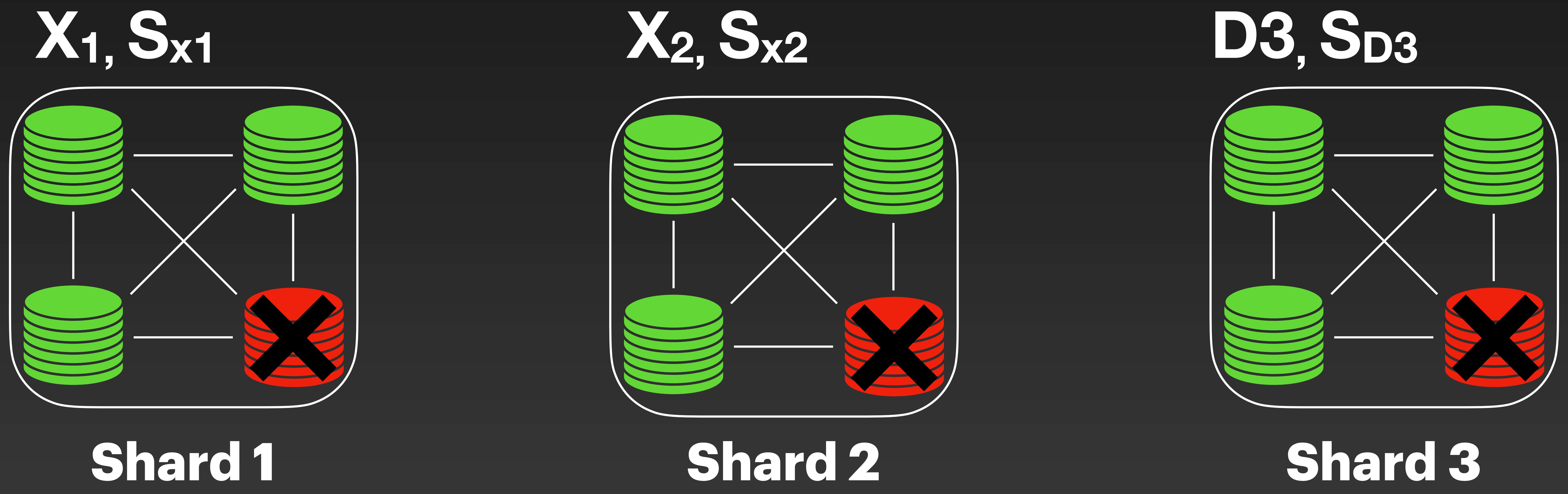
Add sequence numbers per object



Byzcuit

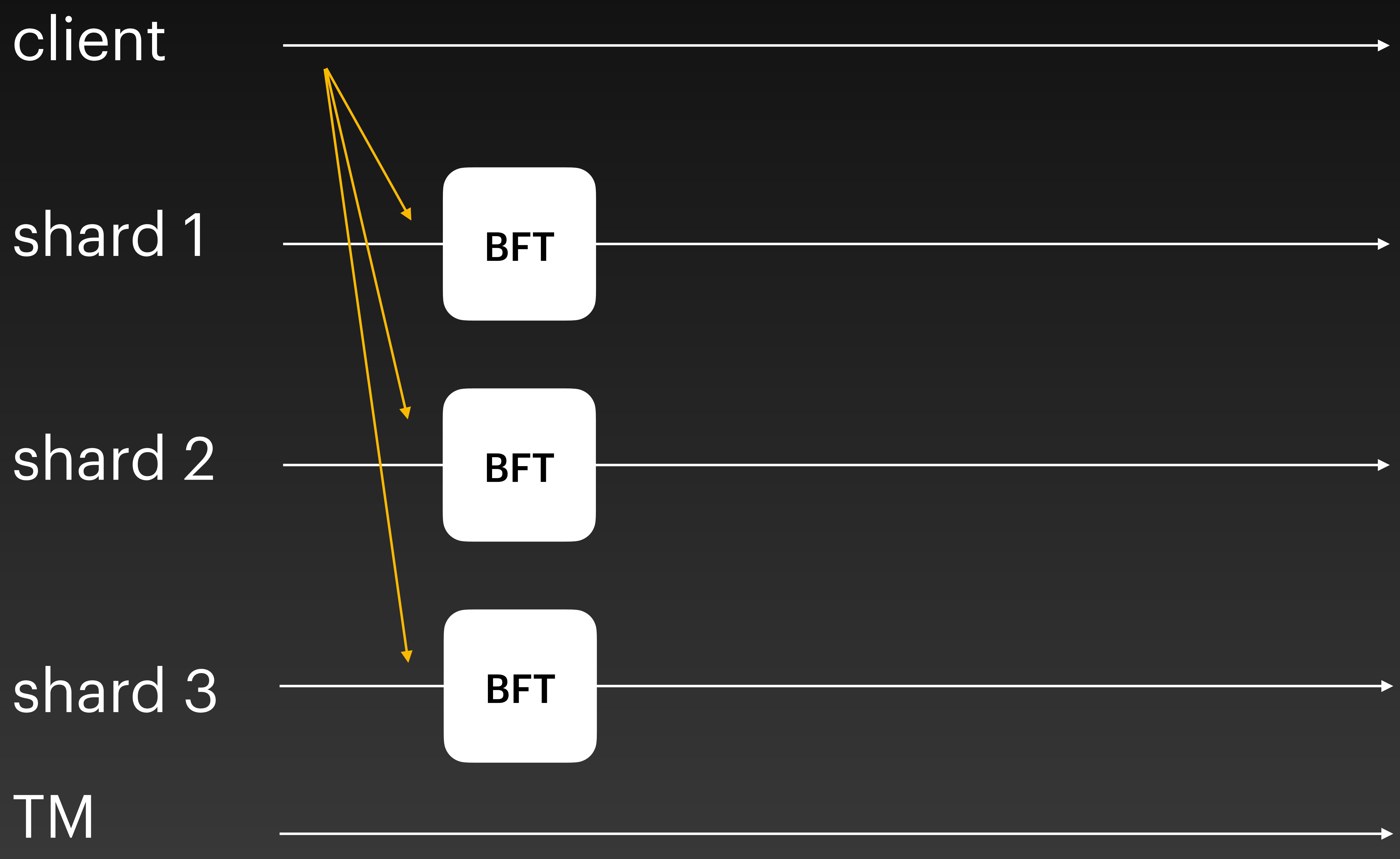
Fix issue 2

Dummy objects for output shards



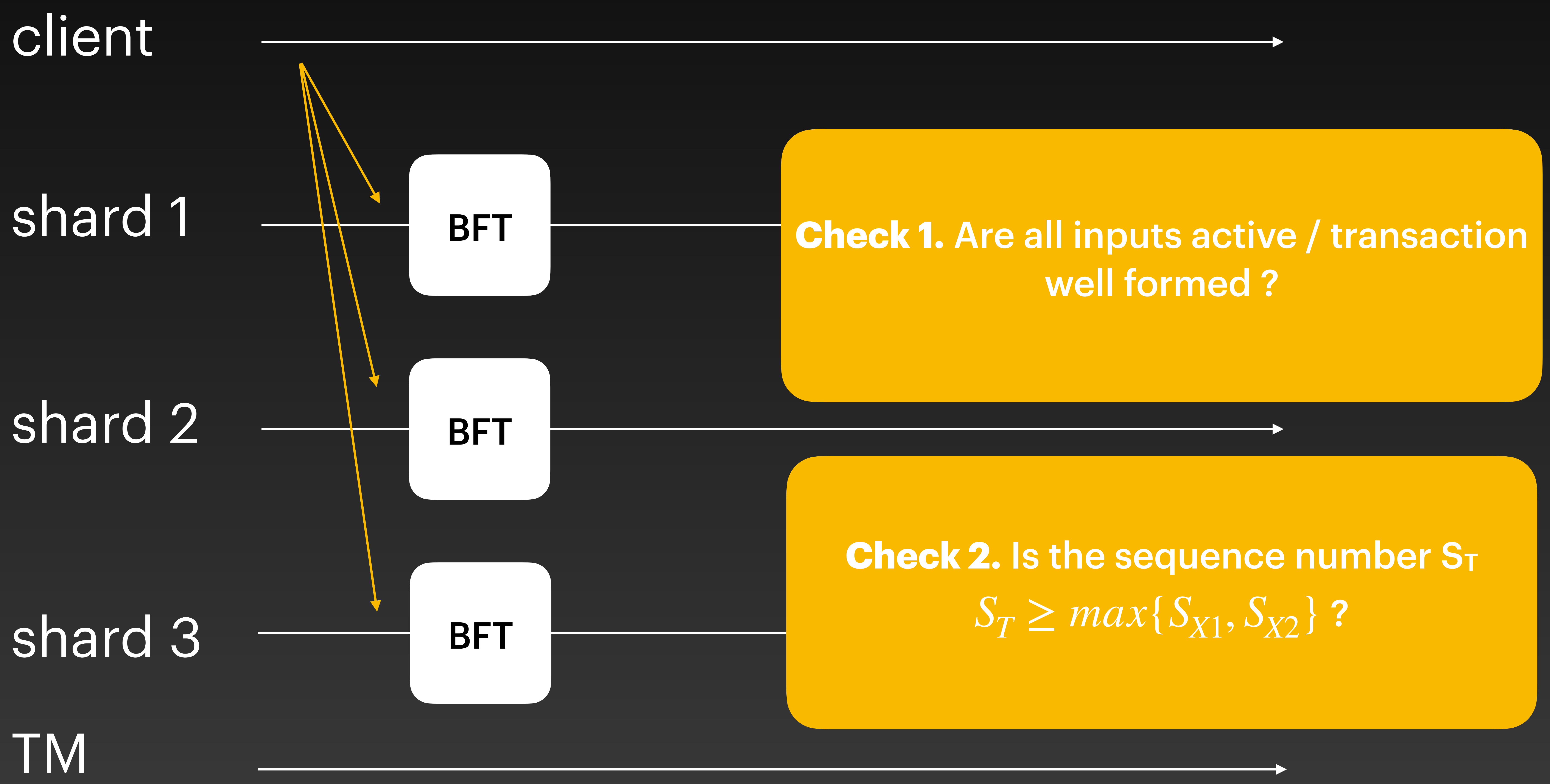
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



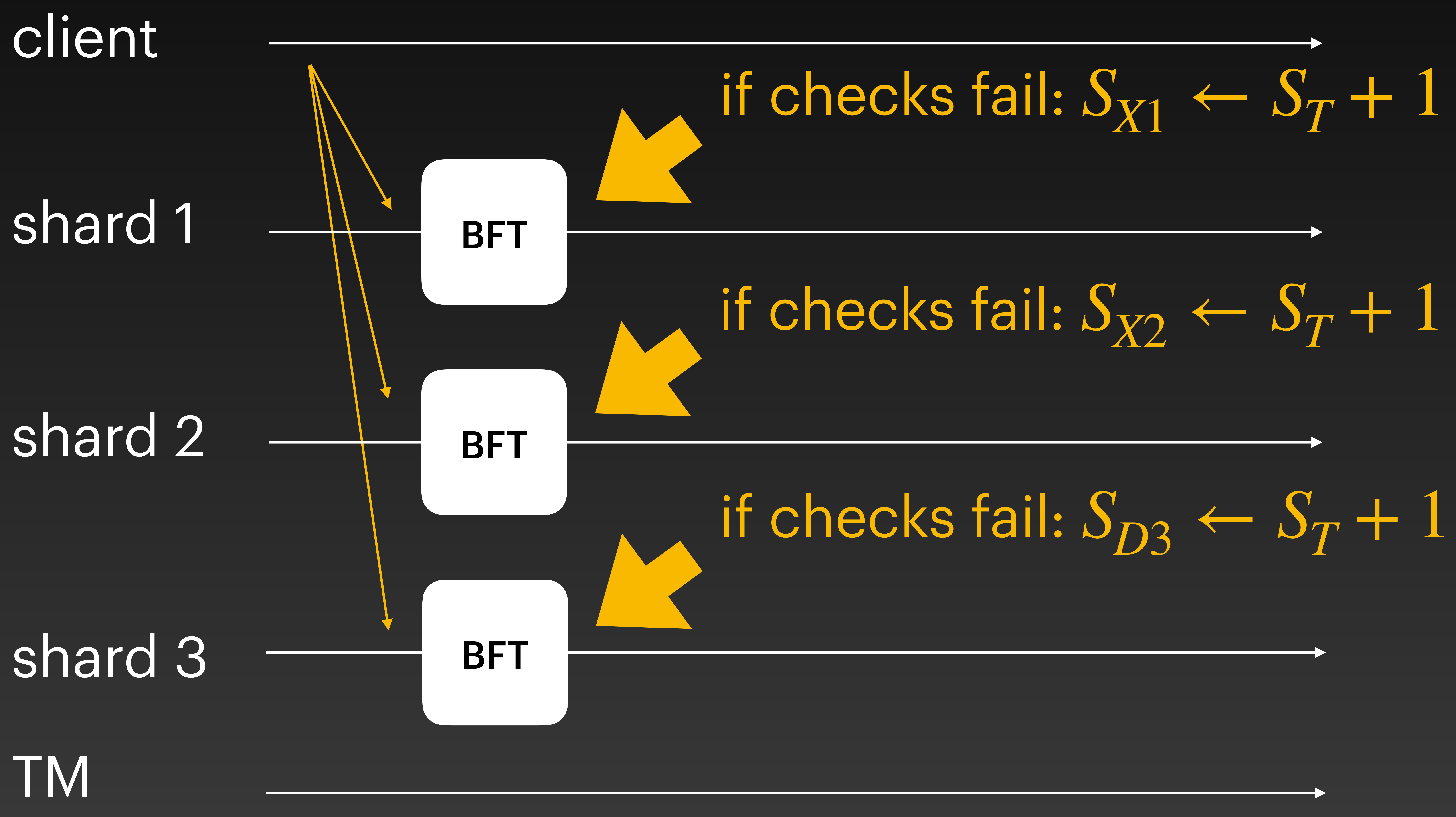
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



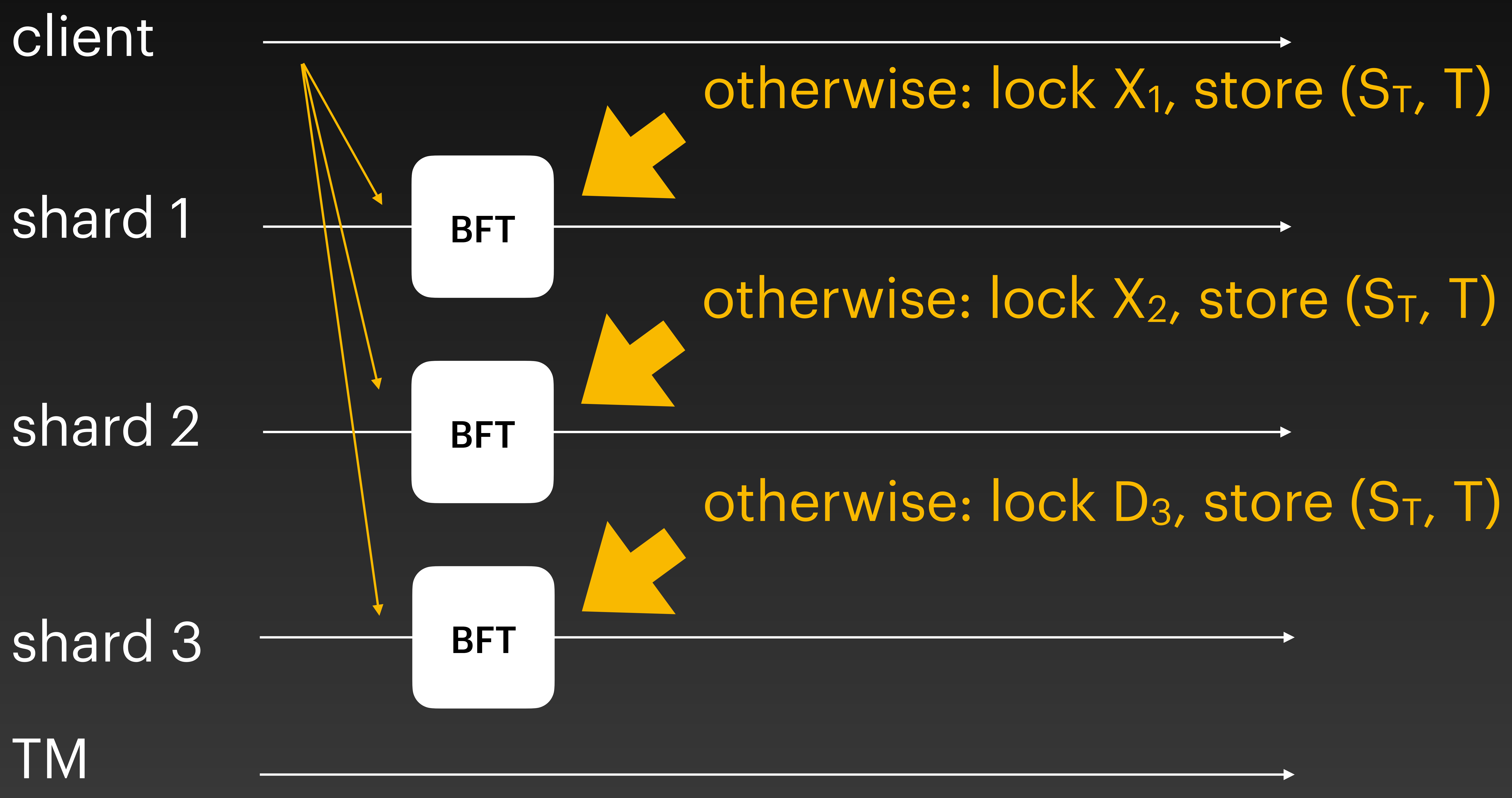
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



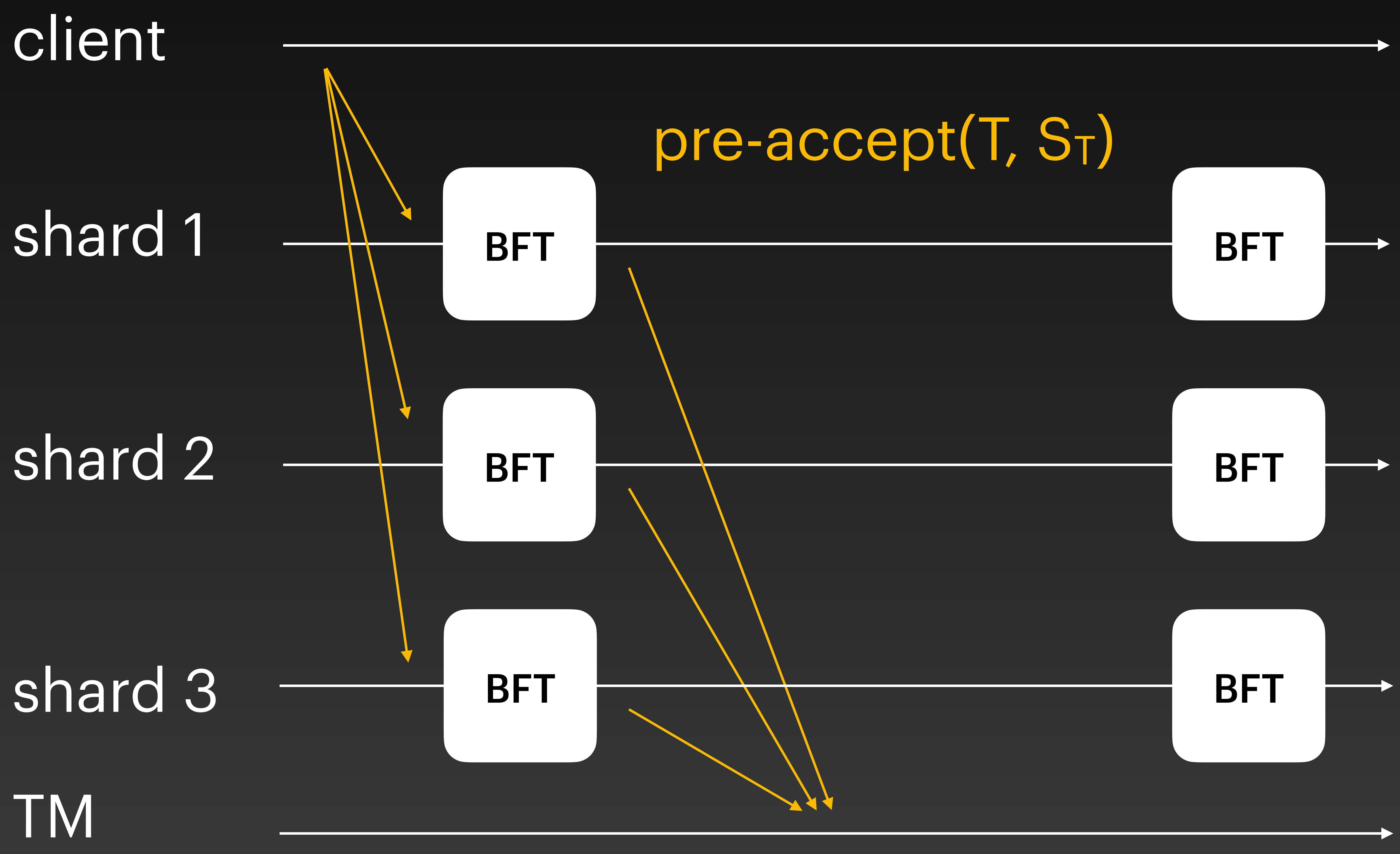
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



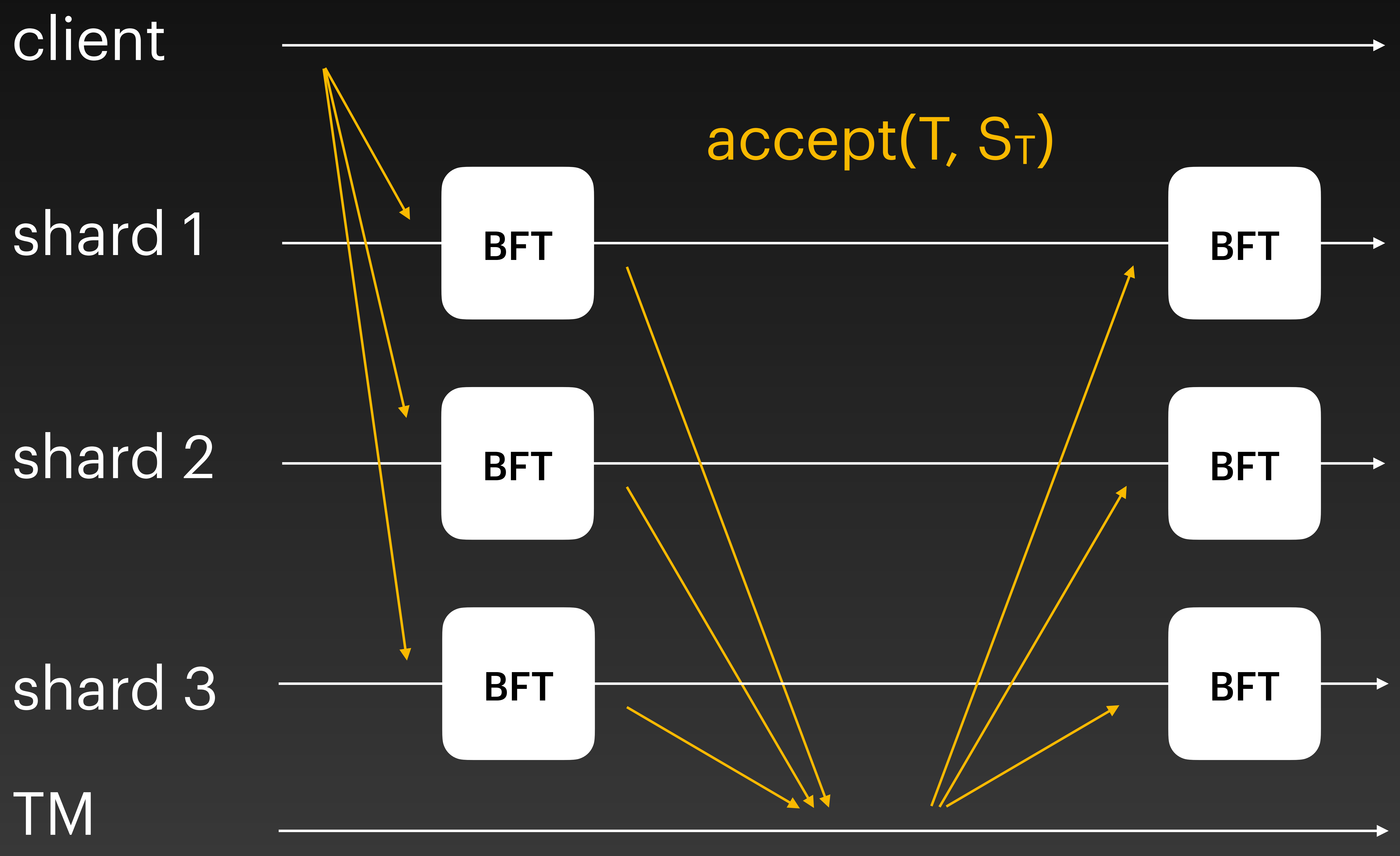
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



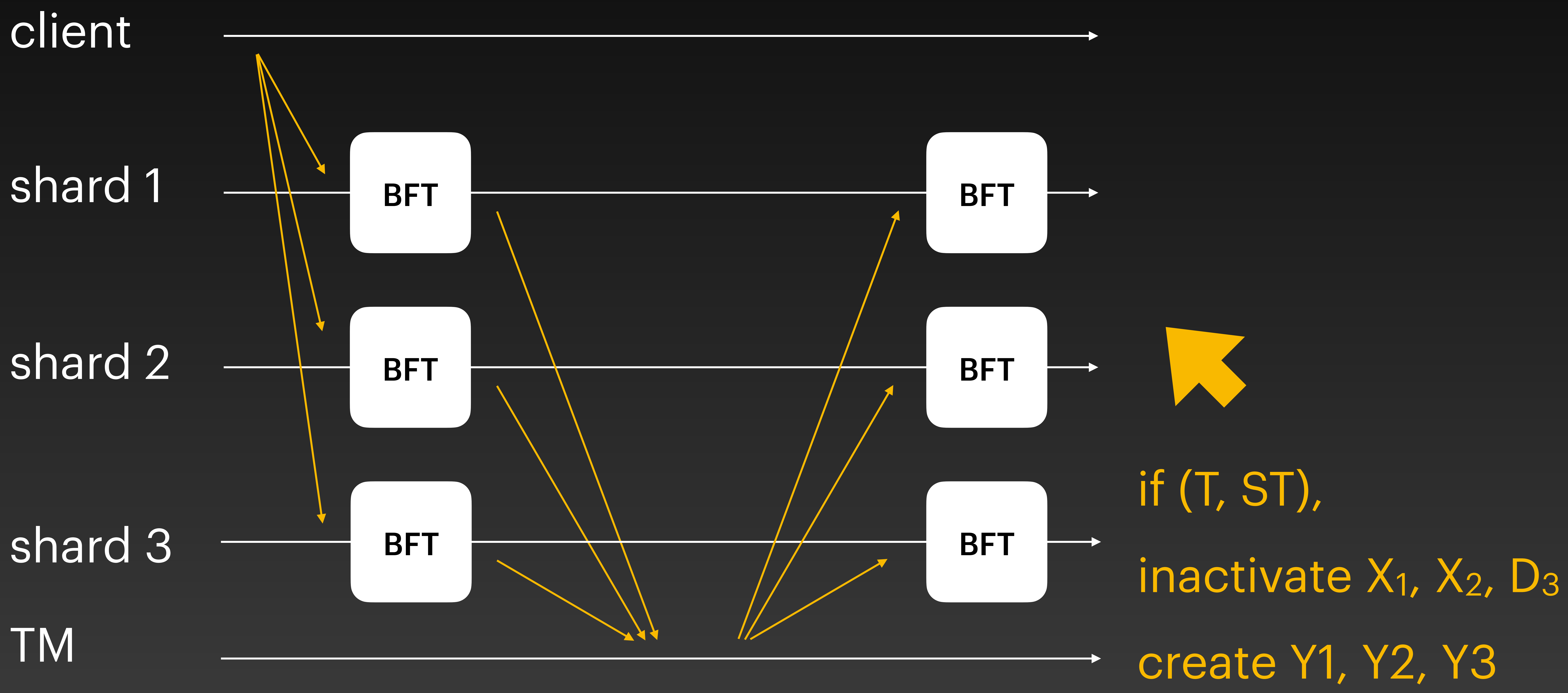
Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



Byzcuit

$$\{S_T, T(x_1, x_2, d_3) \rightarrow (y_1, y_2, y_3)\}$$



Why is Byzcuit secure?

Issue 1. Input shards cannot associate protocol messages to a specific protocol execution.

Sequence numbers:

act as session ID

Issue 2. Output shards (that are not also input shards) do not experience the first phase of the protocol

Dummy objects:

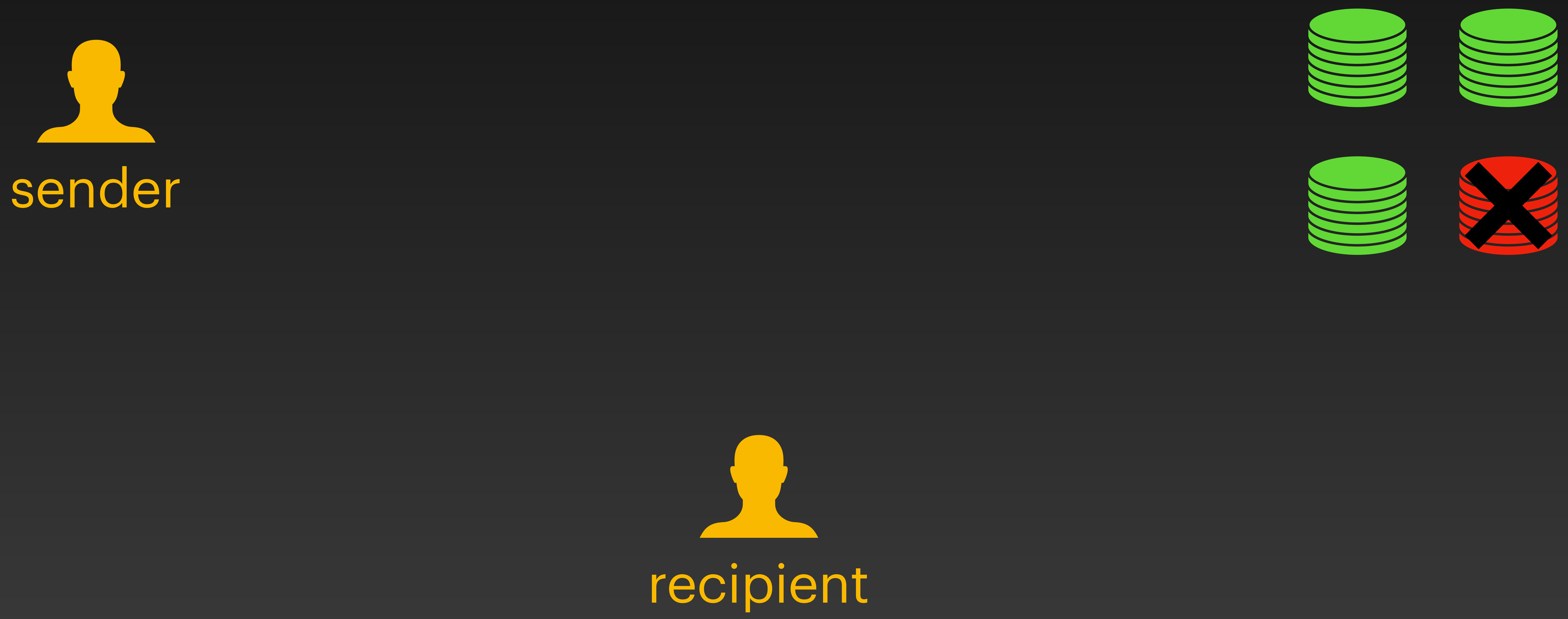
all shards experience the first phase of the protocol

EXTRA

FastPay Transfer

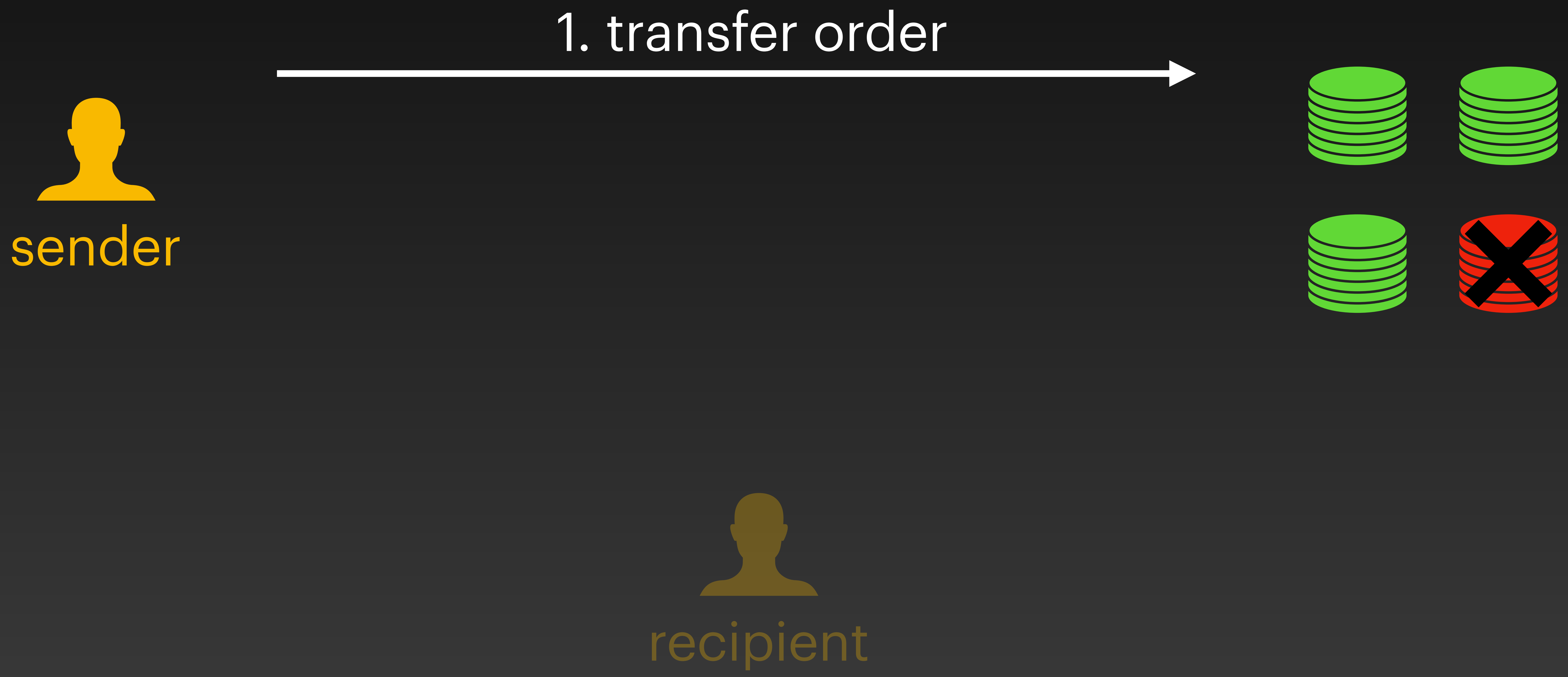
FastPay

How does it work?

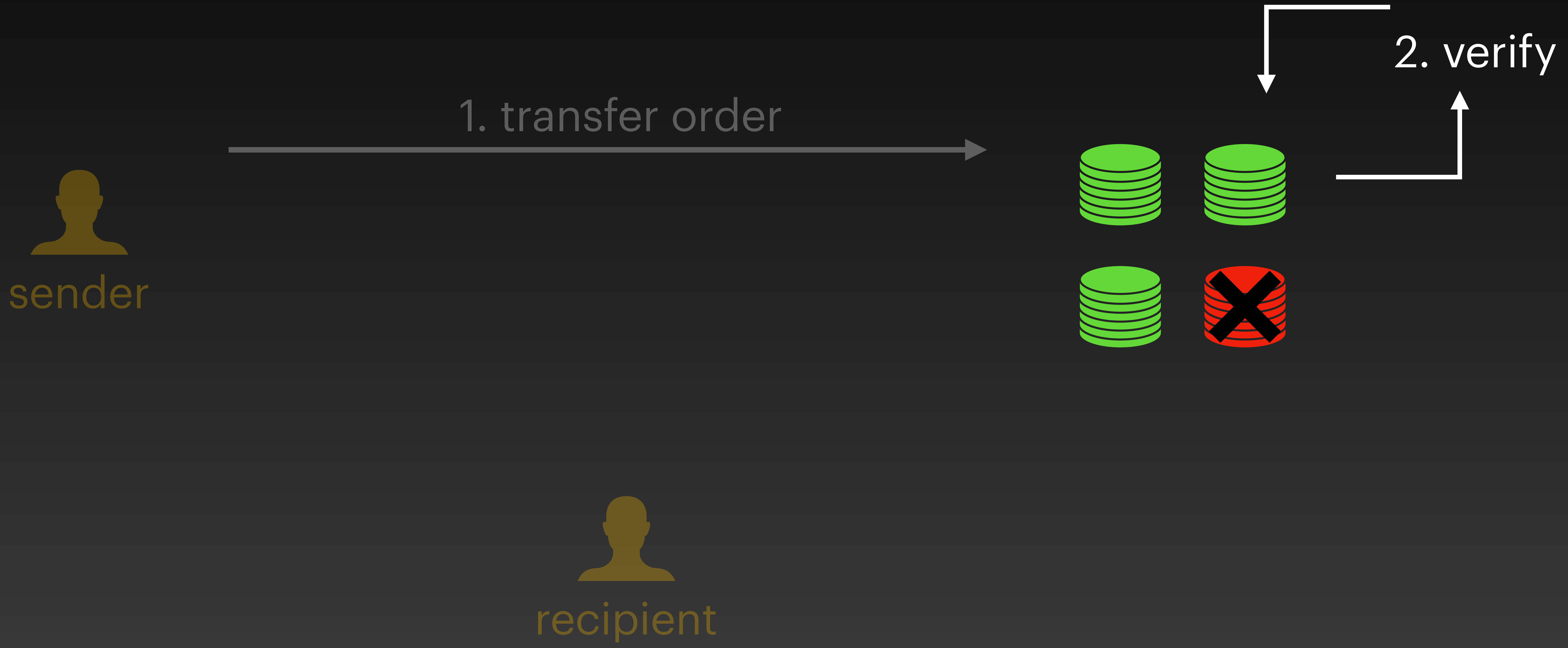


FastPay

How does it work?

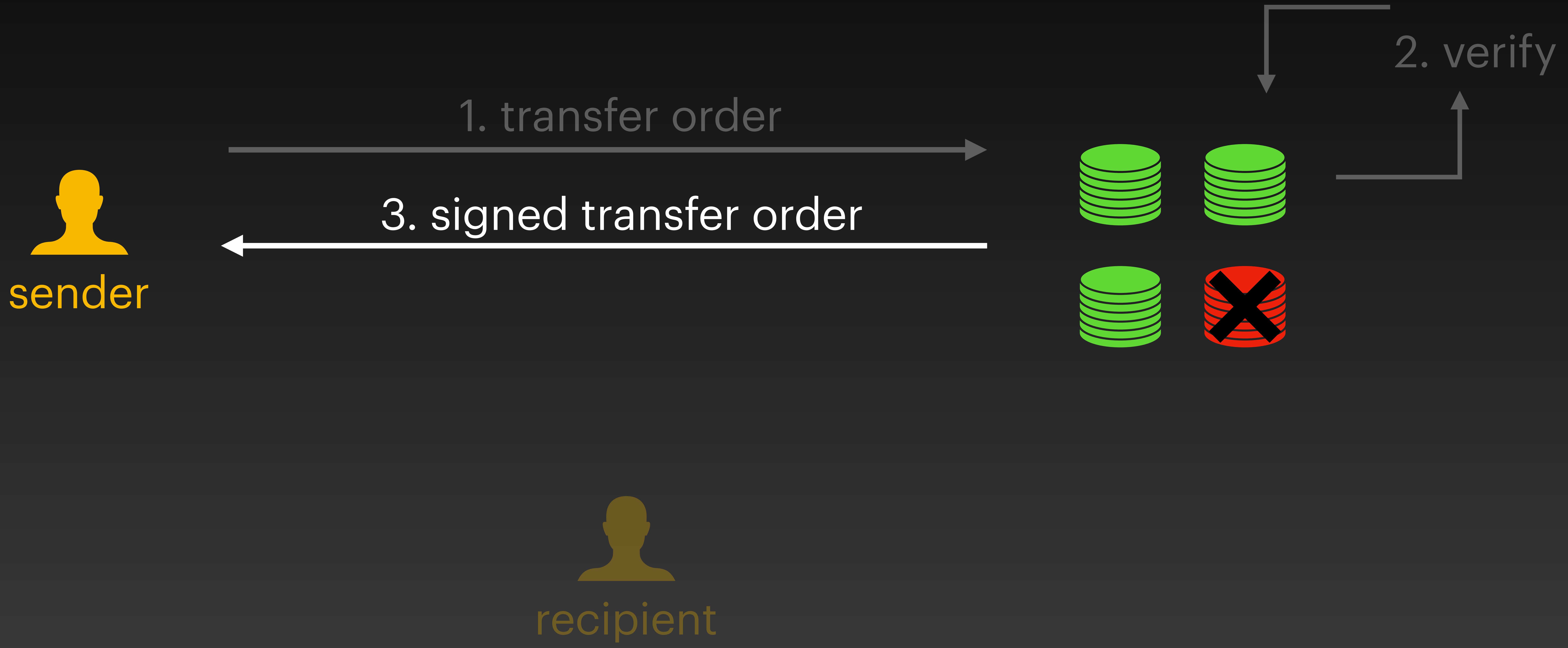


FastPay
How does it work?



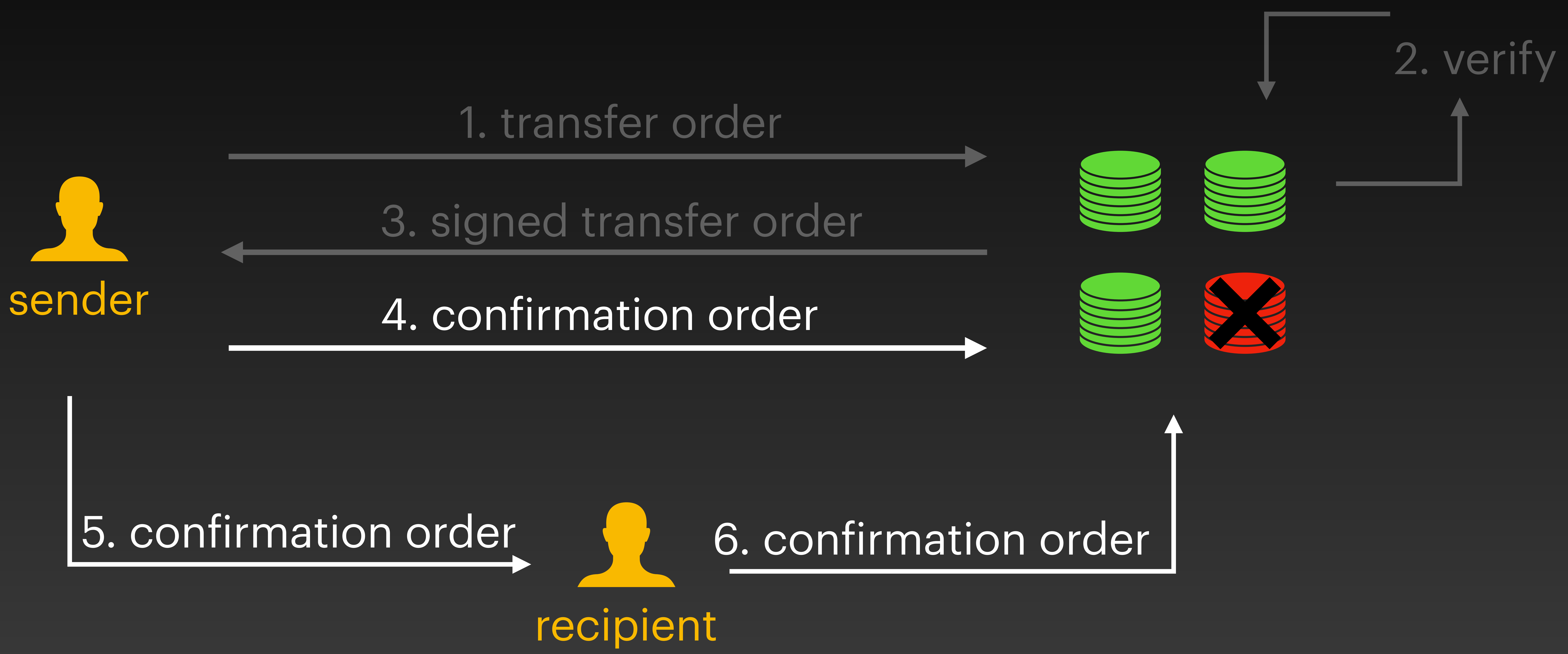
FastPay

How does it work?



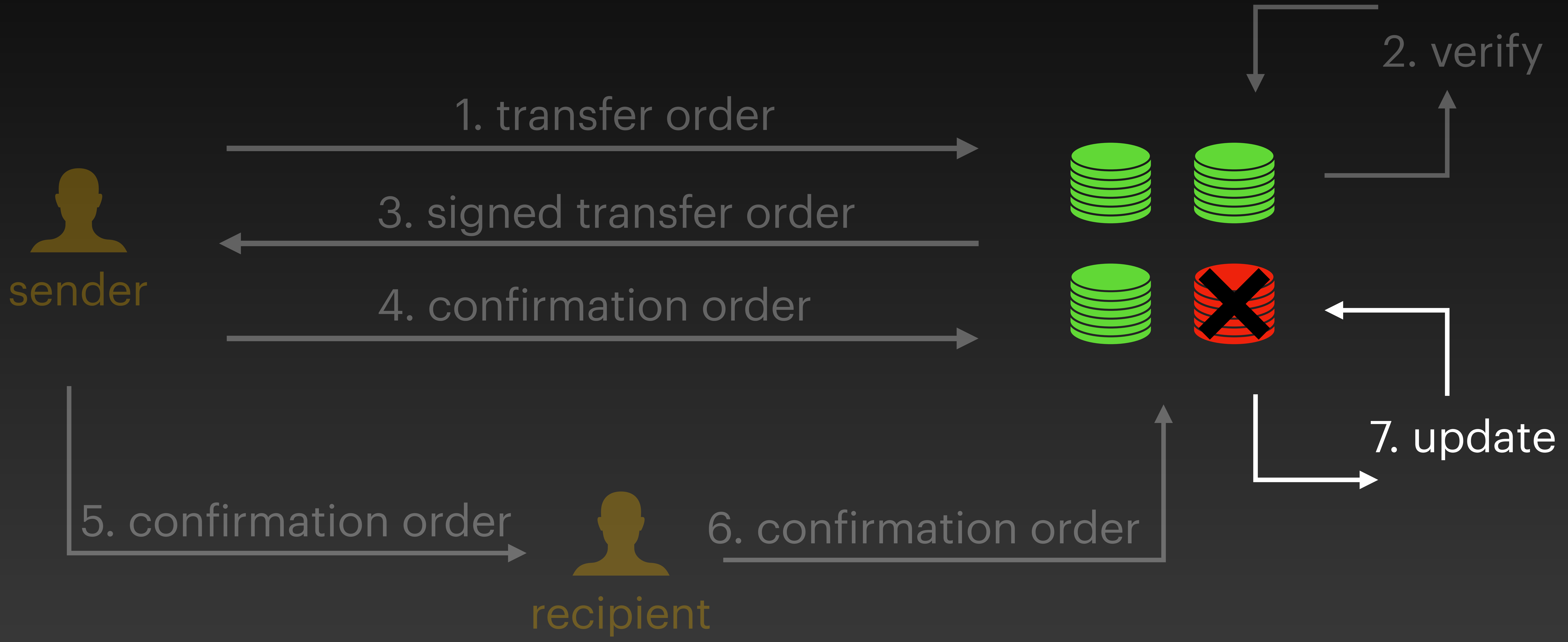
FastPay

How does it work?



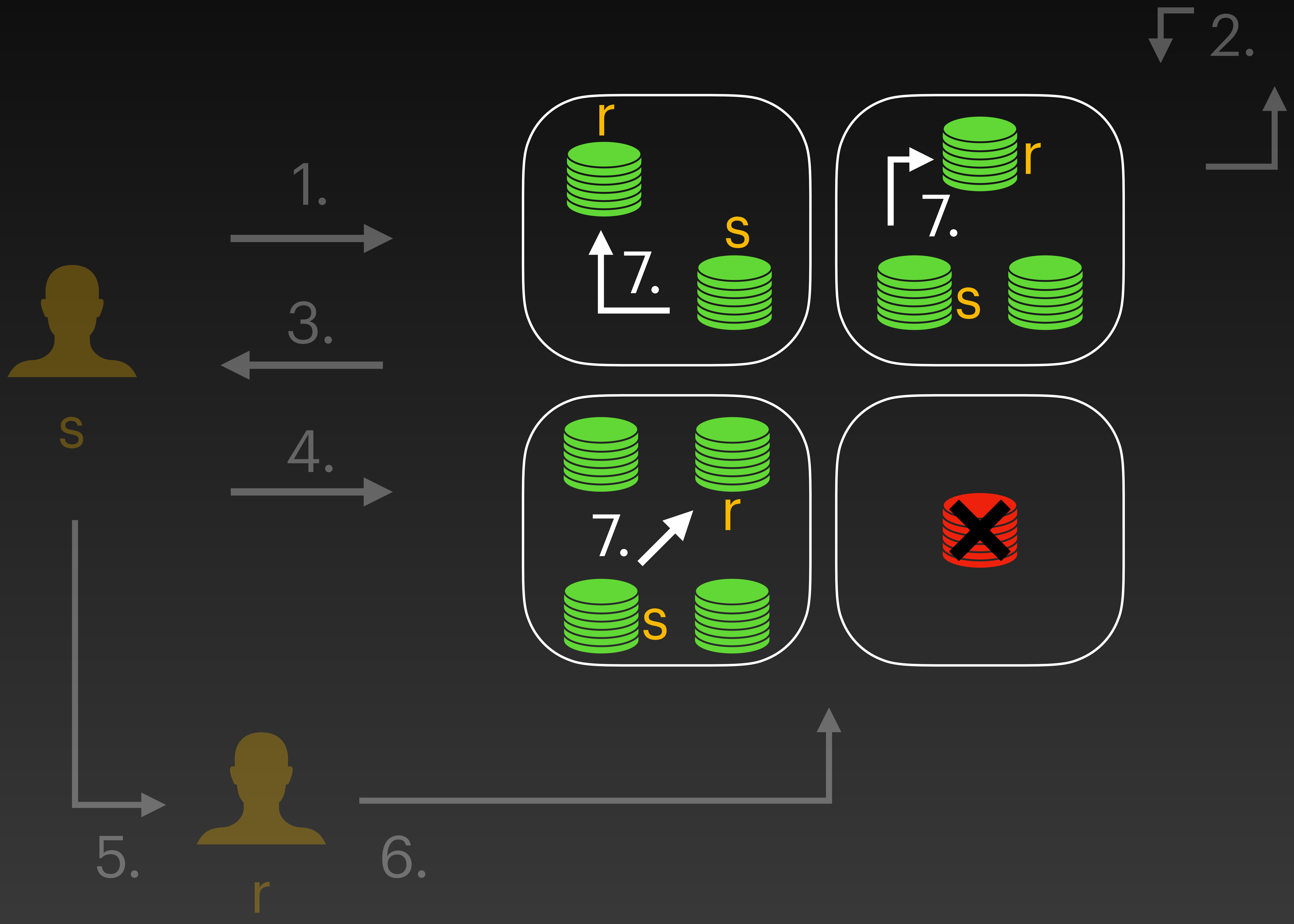
FastPay

How does it work?



FastPay

Increasing capacity



Byzantine Consistent Broadcast

Validity

No duplication

Integrity

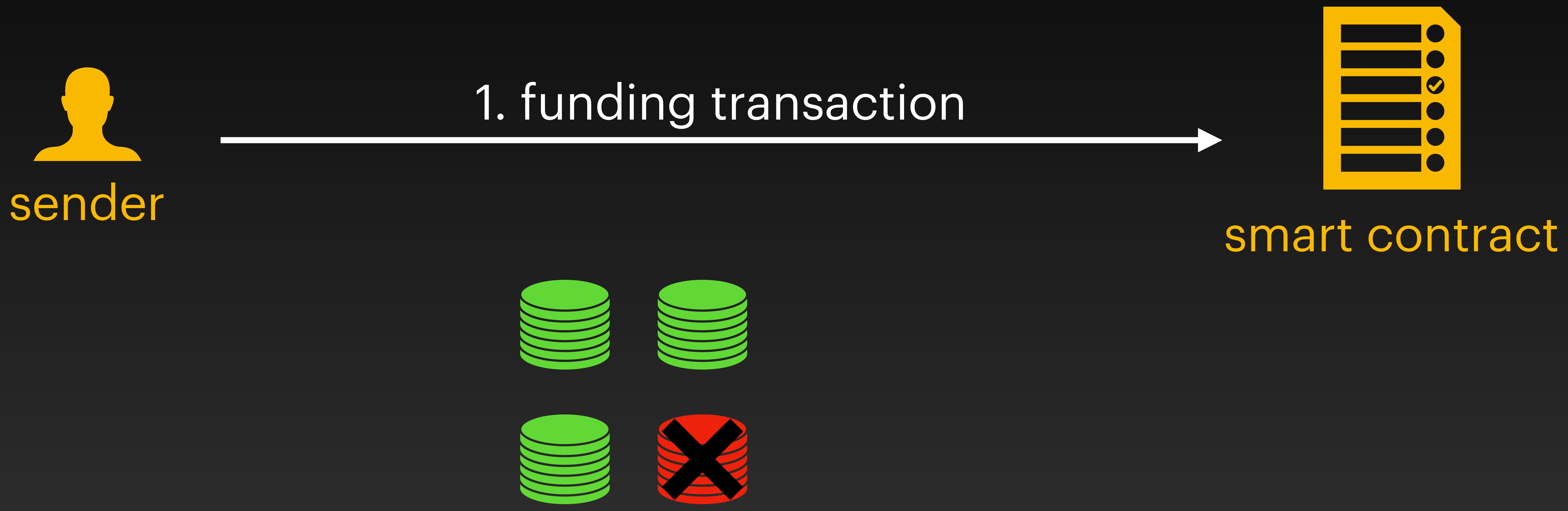
Consistency

EXTRA

FastPay Smart Contract Interface

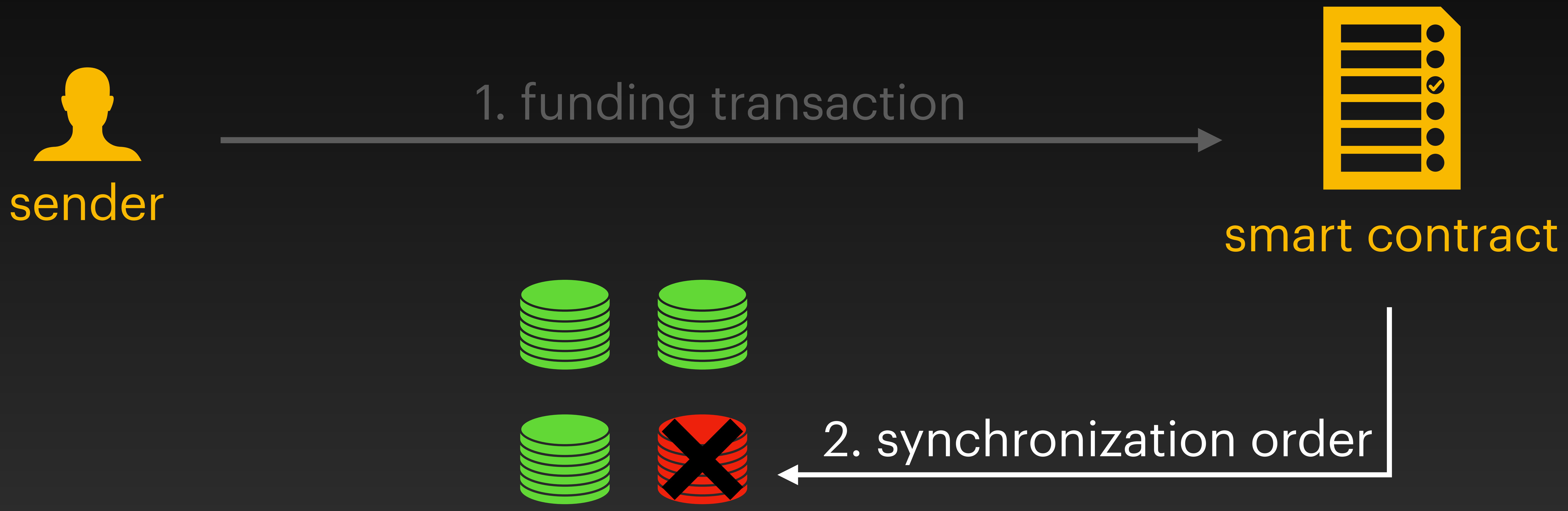
FastPay

From primary infrastructure to FastPay



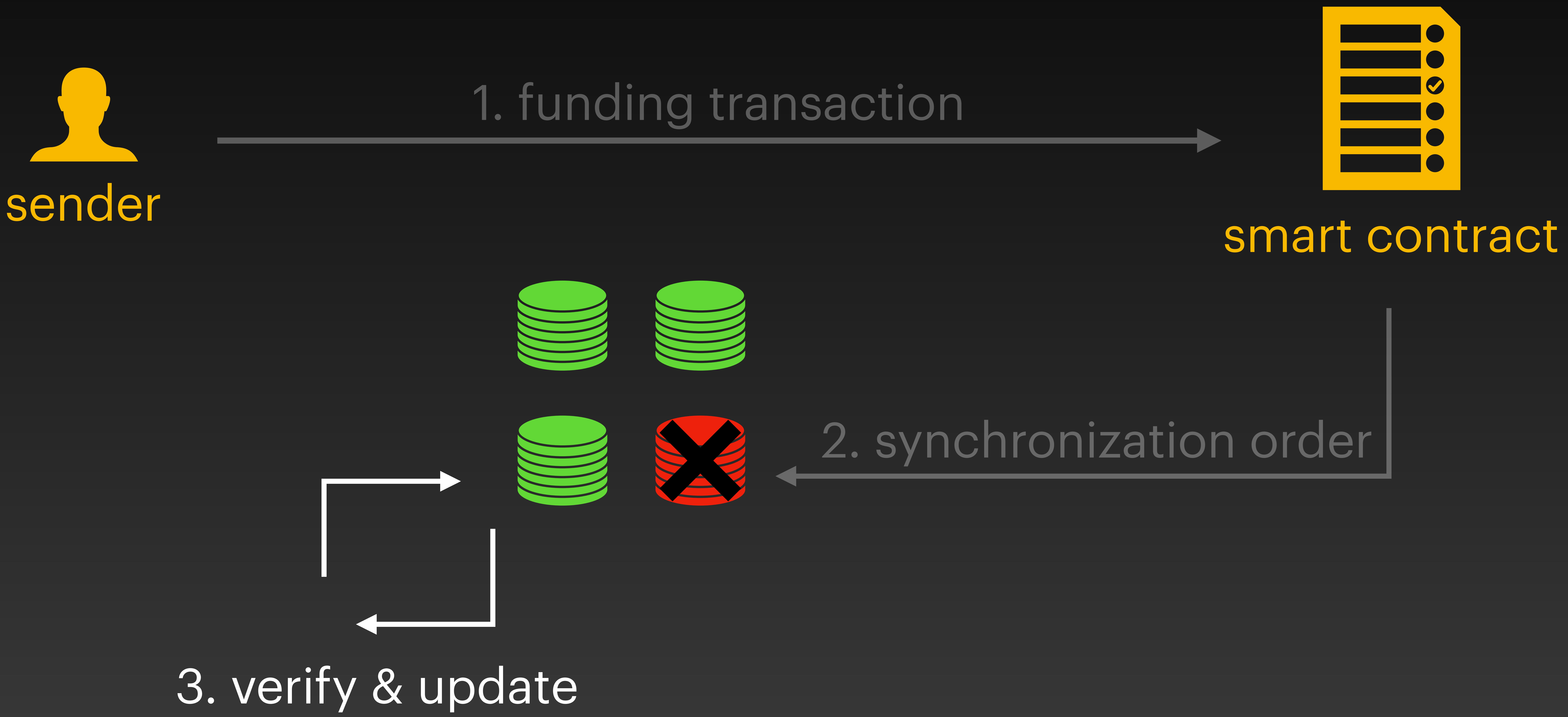
FastPay

From primary infrastructure to FastPay



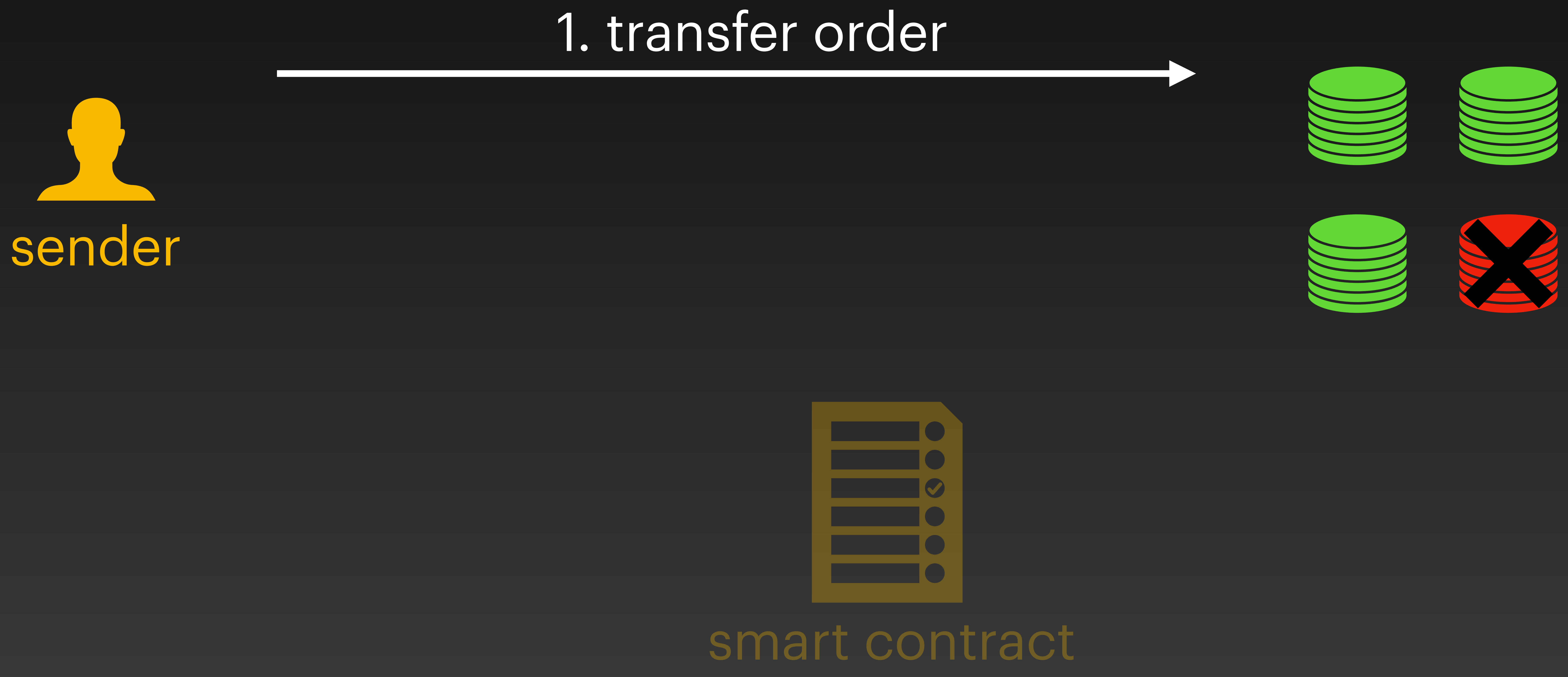
FastPay

From primary infrastructure to FastPay



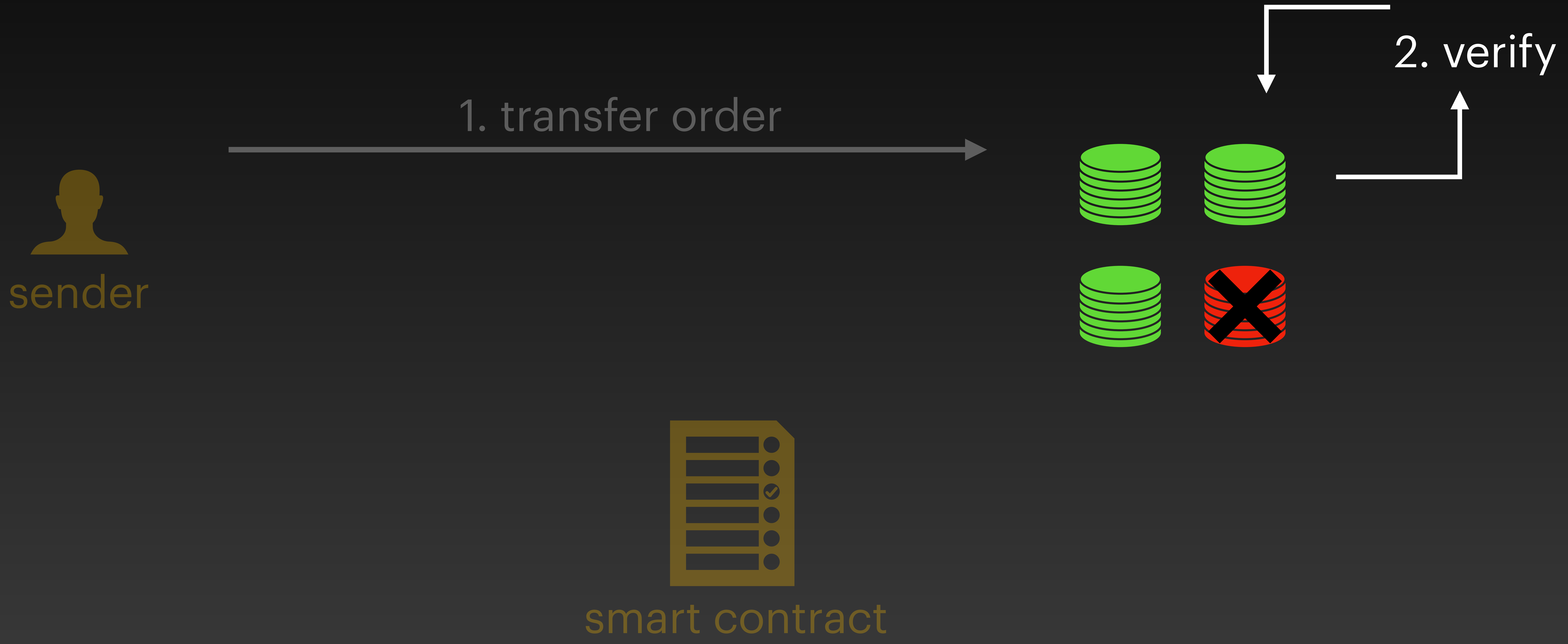
FastPay

From the primary infrastructure to FastPay



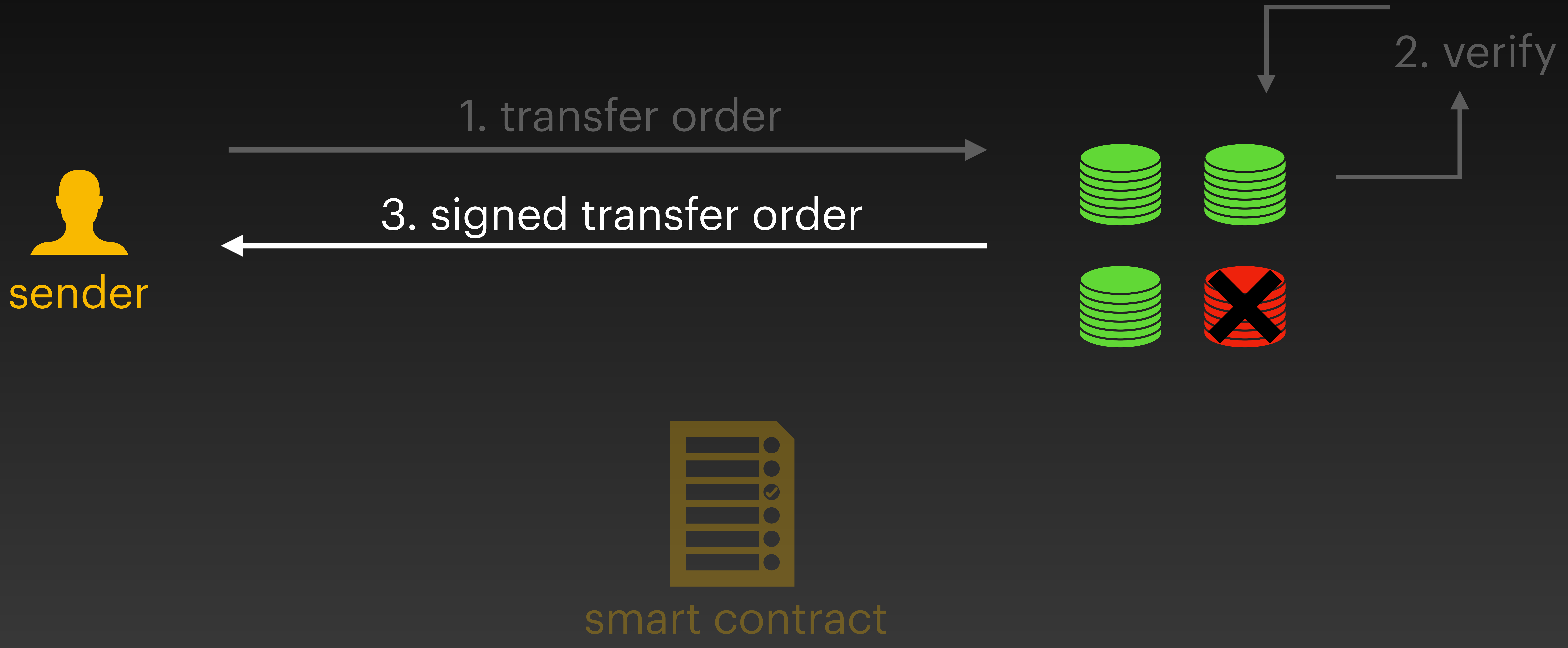
FastPay

From the primary infrastructure to FastPay



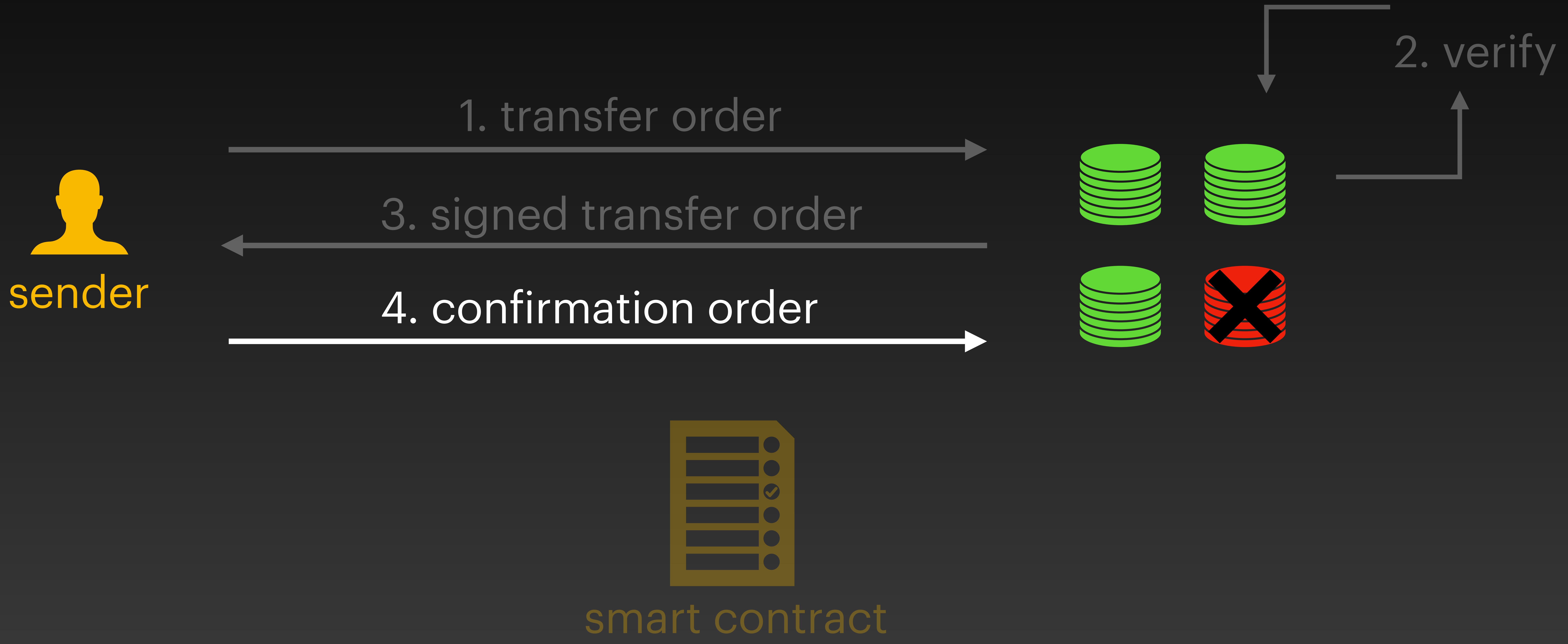
FastPay

From the primary infrastructure to FastPay



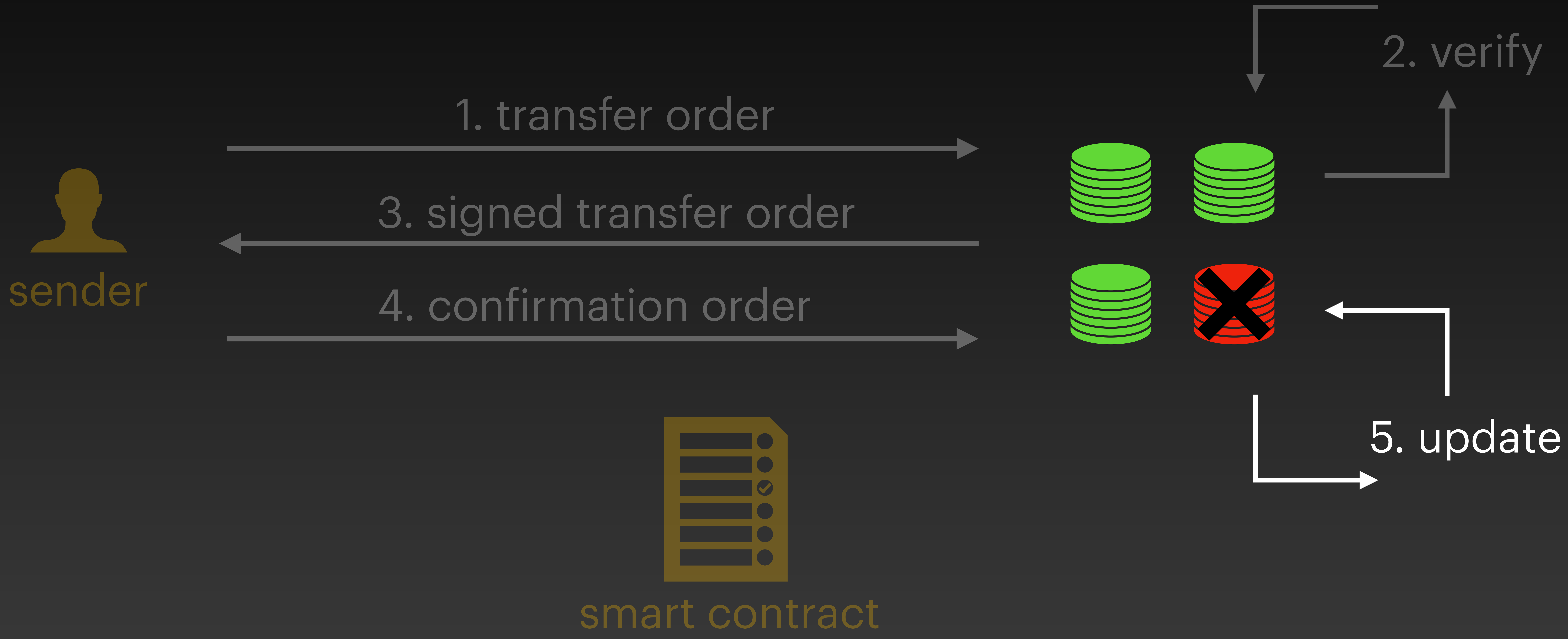
FastPay

From the primary infrastructure to FastPay



FastPay

From the primary infrastructure to FastPay



FastPay

From the primary infrastructure to FastPay