Hostile Actors and Attack Vectors

@DG Lab - Karl-Johan Alm

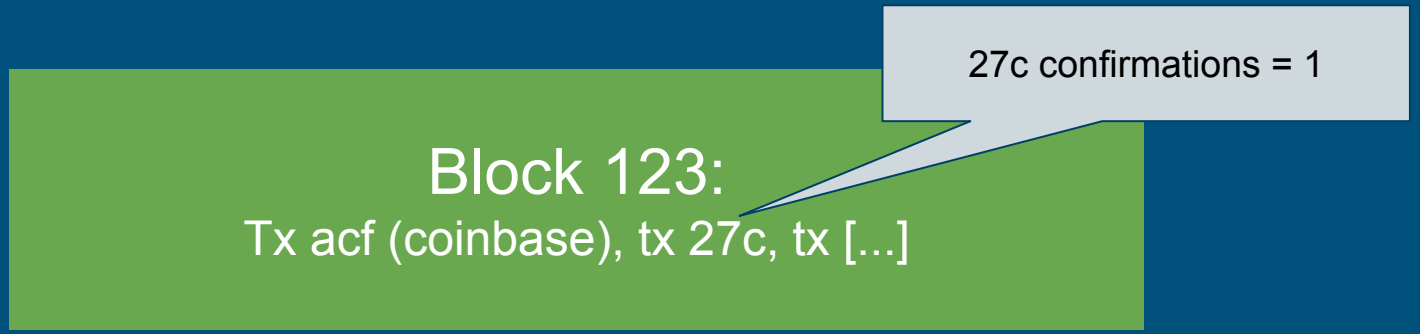
Confirmations, reorgs

Confirmations

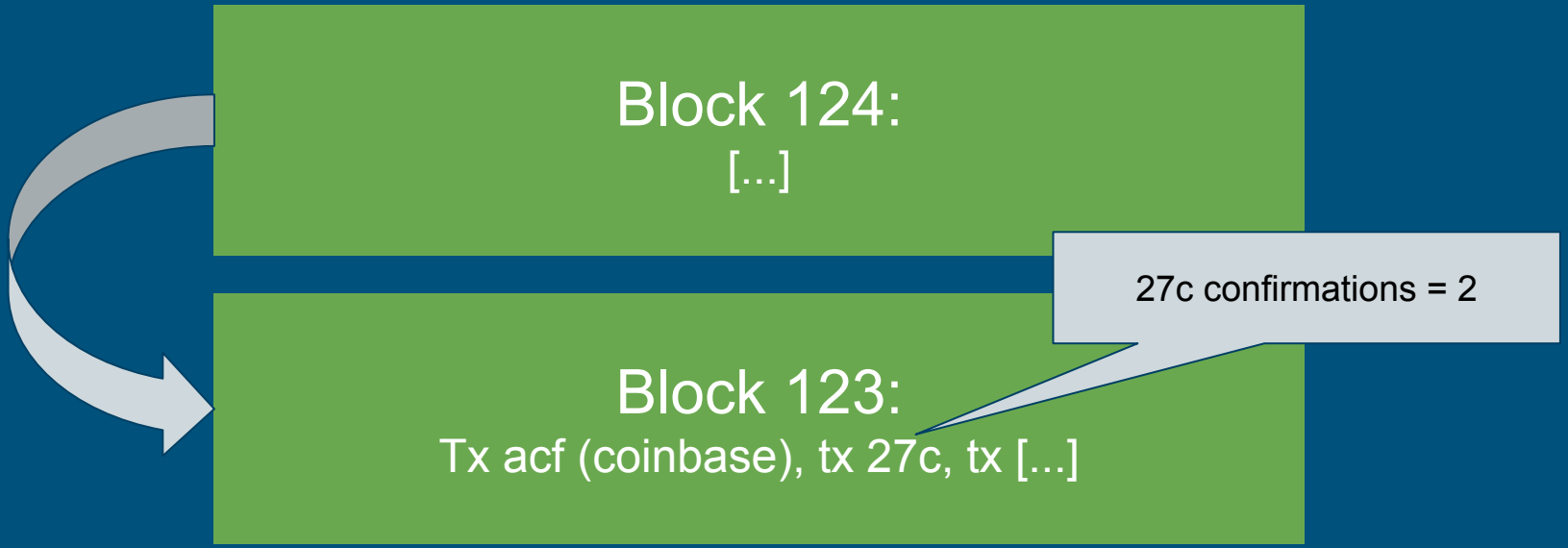
Block 123:

Tx acf (coinbase), tx 27c, tx [...]

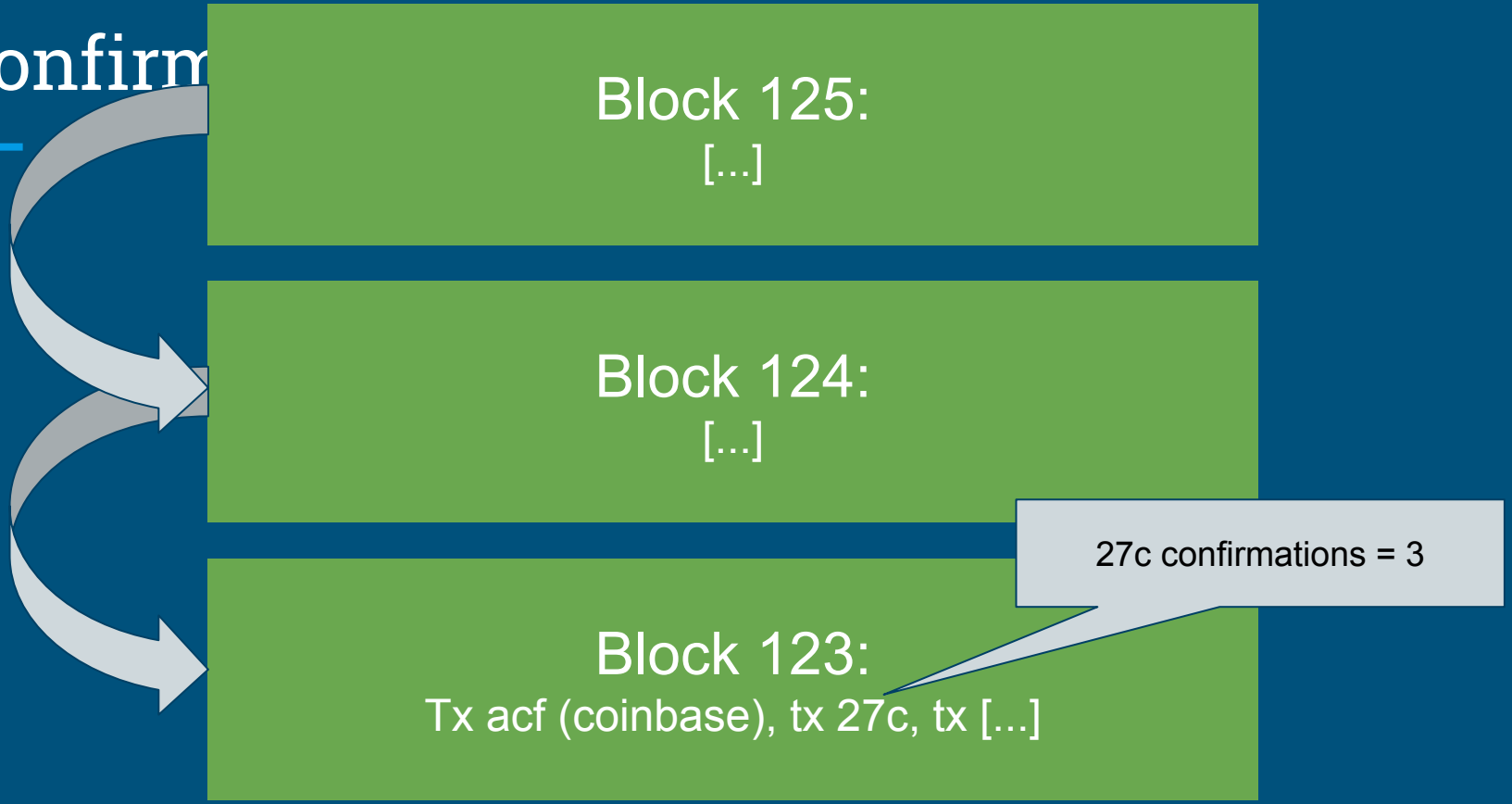
Confirmations



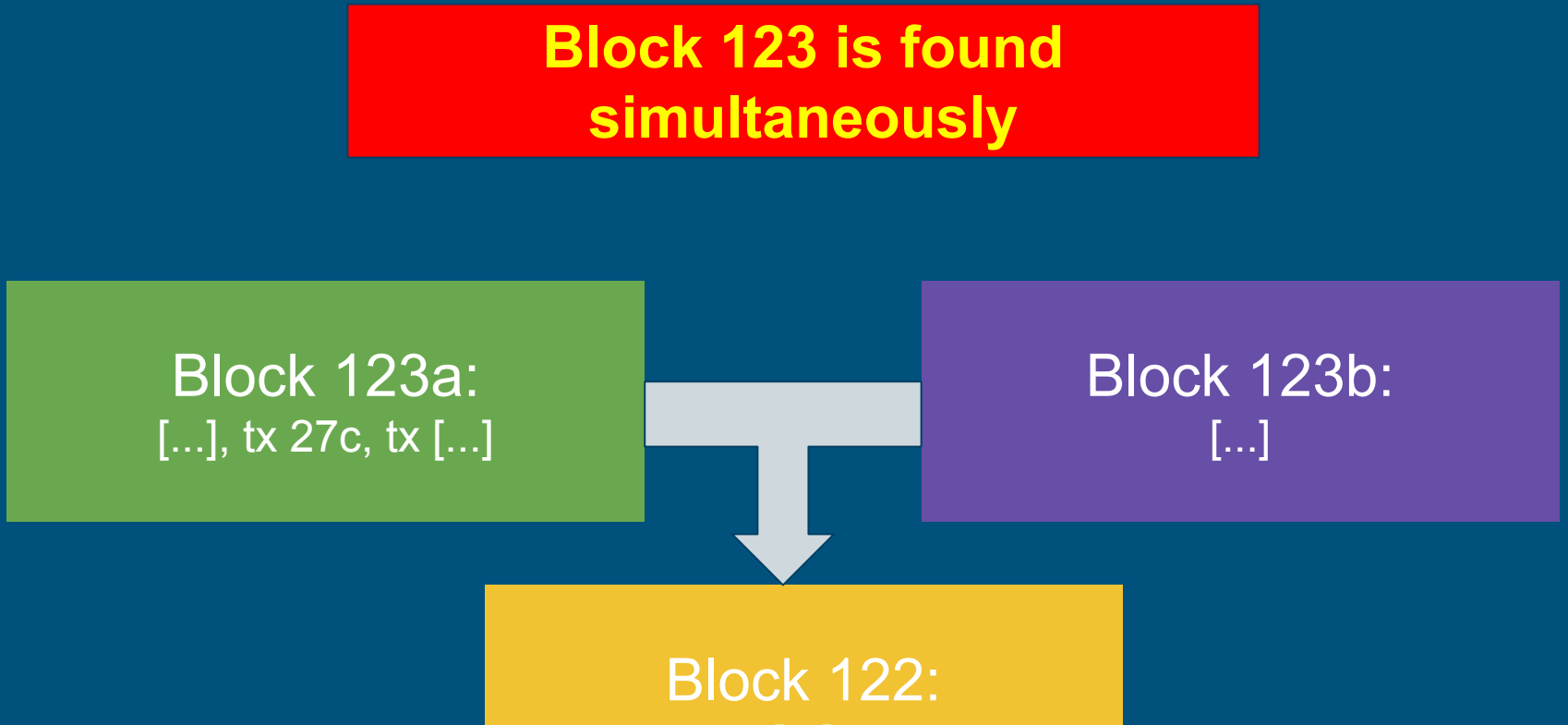
Confirmations



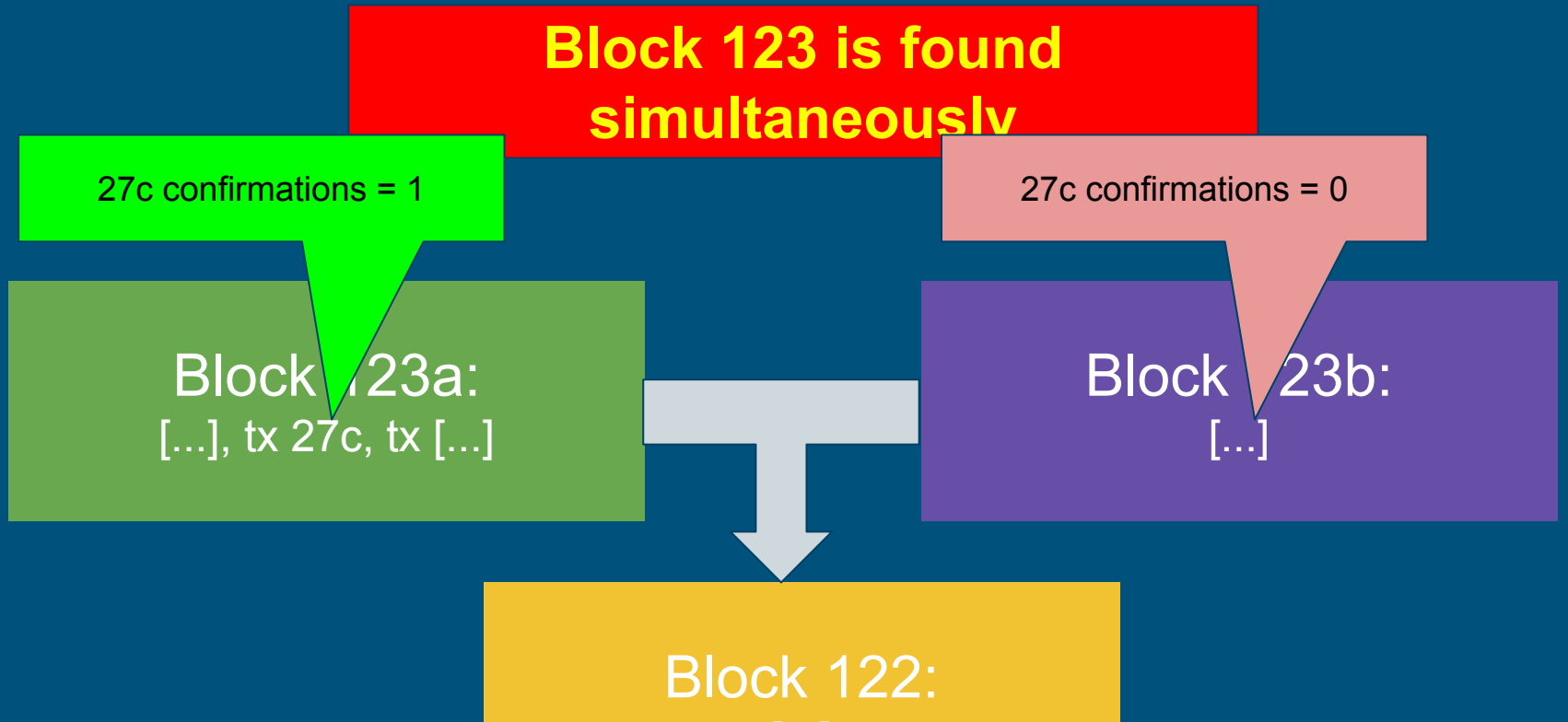
Confirm


Block 125:
[...]
Block 124:
[...]
Block 123:
Tx acf (coinbase), tx 27c, tx [...]
27c confirmations = 3

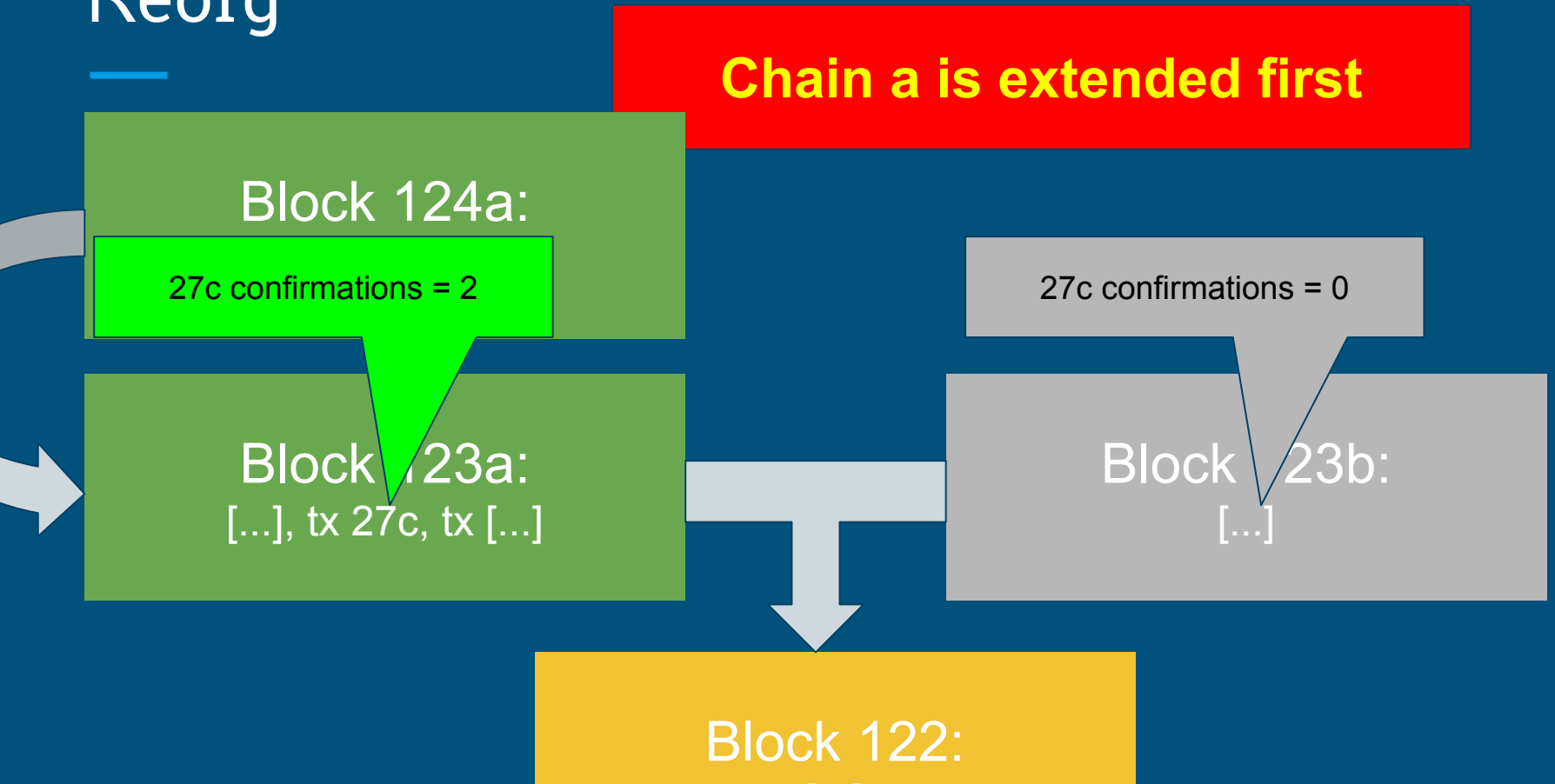
Reorg


Block 123 is found simultaneously
Block 123a:
[...], tx 27c, tx [...]
Block 123b:
[...]
Block 122:

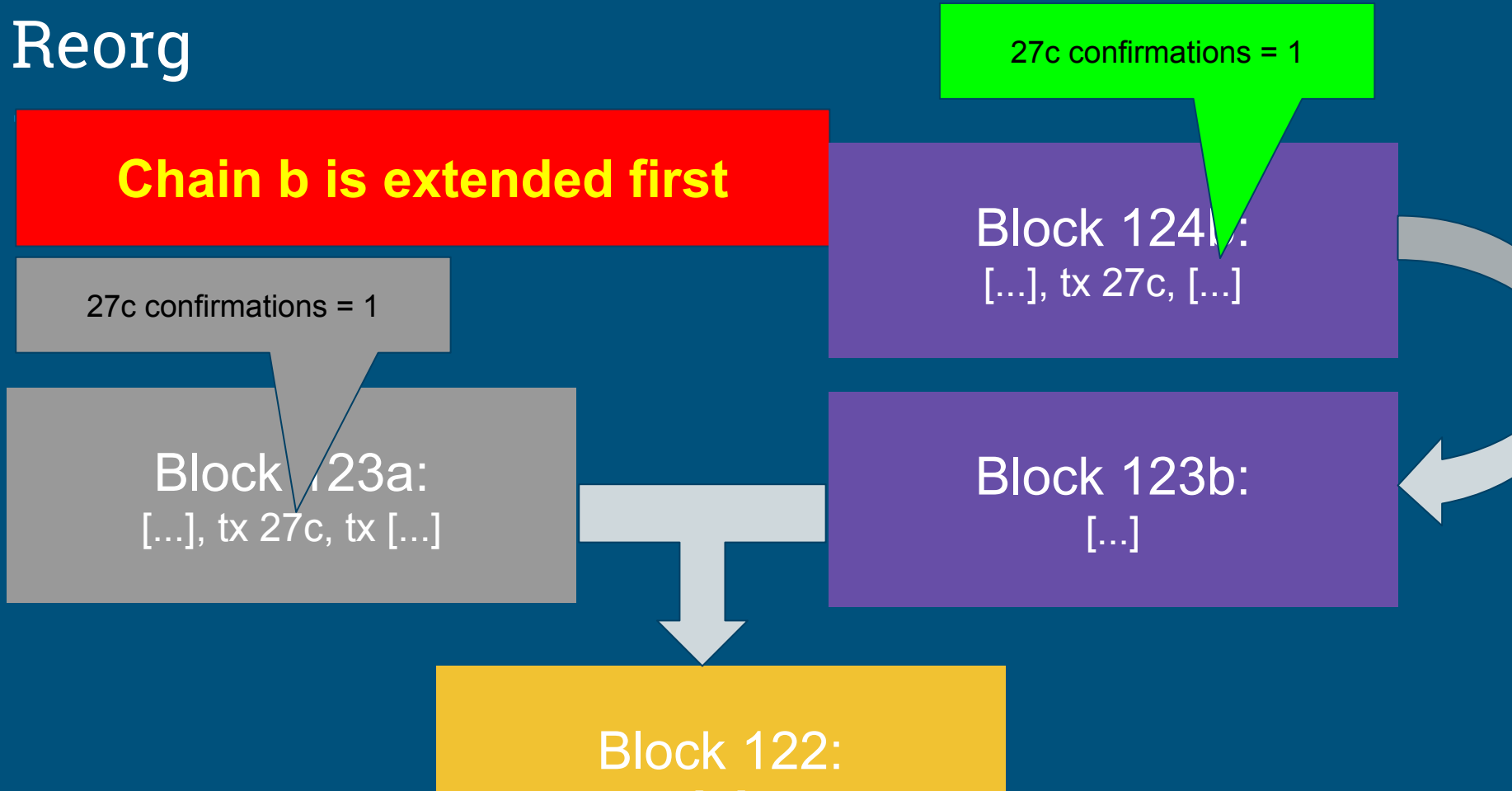
Reorg


Block 123 is found simultaneously
27c confirmations = 1
27c confirmations = 0
Block 123a:
[...], tx 27c, tx [...]
Block 123b:
[...]
Block 122:

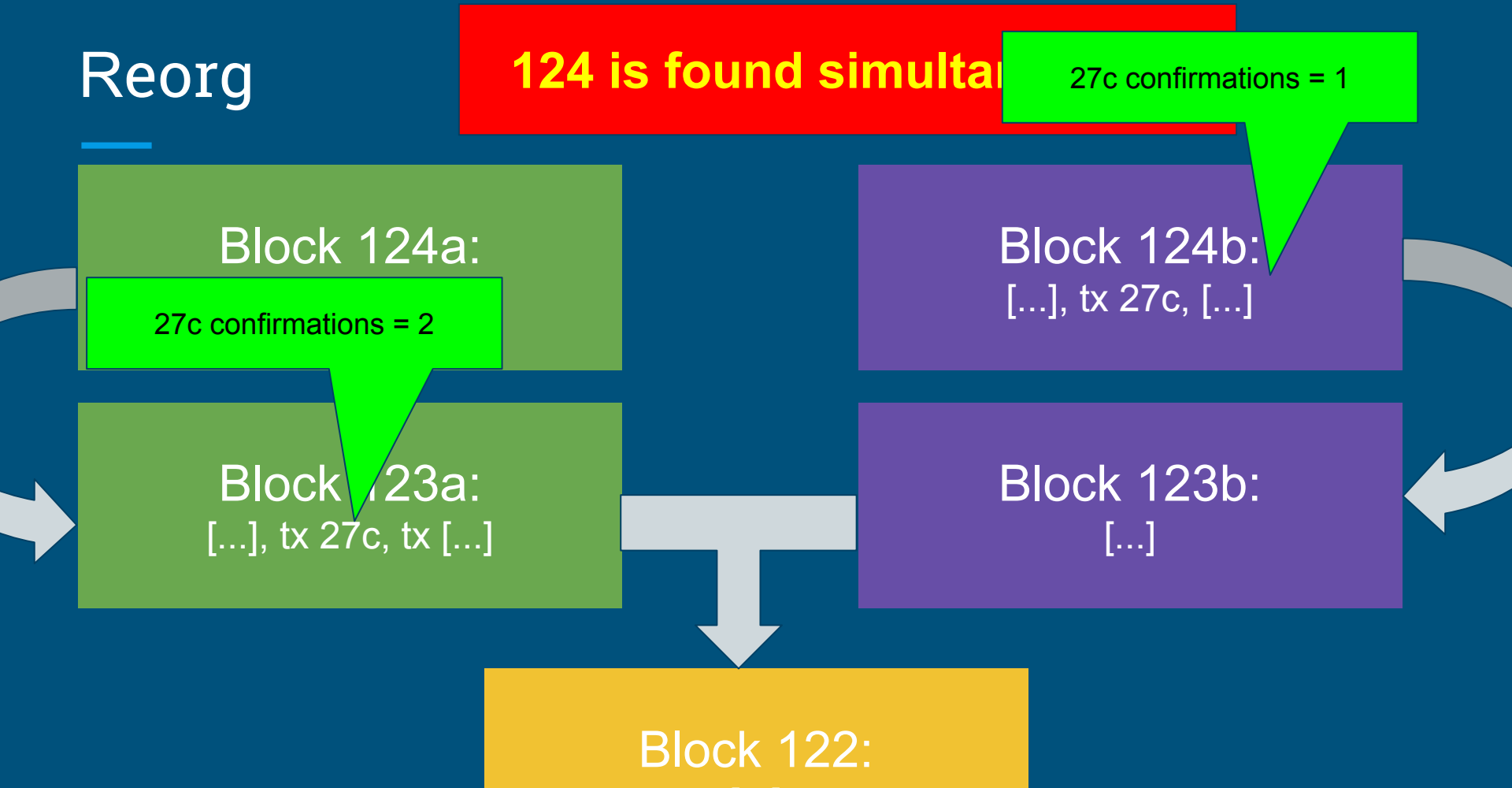
Reorg


Chain a is extended first
Block 124a:
27c confirmations = 2
Block 123a:
[...], tx 27c, tx [...]
27c confirmations = 0
Block 123b:
[...]
Block 122:

Reorg


27c confirmations = 1
Chain b is extended first
Block 124b:
[...], tx 27c, [...]
27c confirmations = 1
Block 123a:
[...], tx 27c, tx [...]
Block 123b:
[...]
Block 122:

Reorg

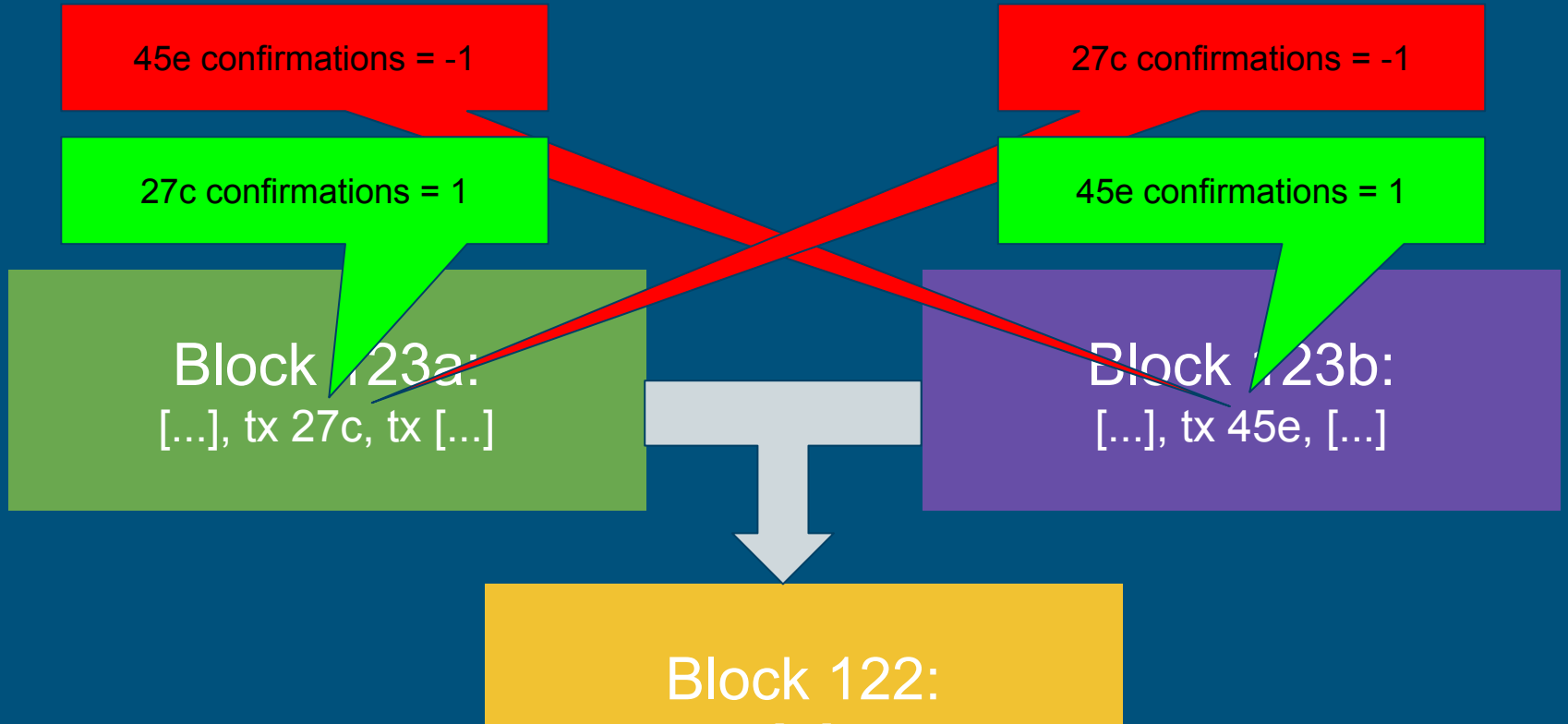

124 is found simulta
27c confirmations = 1
Block 124a:
27c confirmations = 2
Block 124b:
[...], tx 27c, [...]
Block 123a:
[...], tx 27c, tx [...]
Block 123b:
[...]
Block 122:

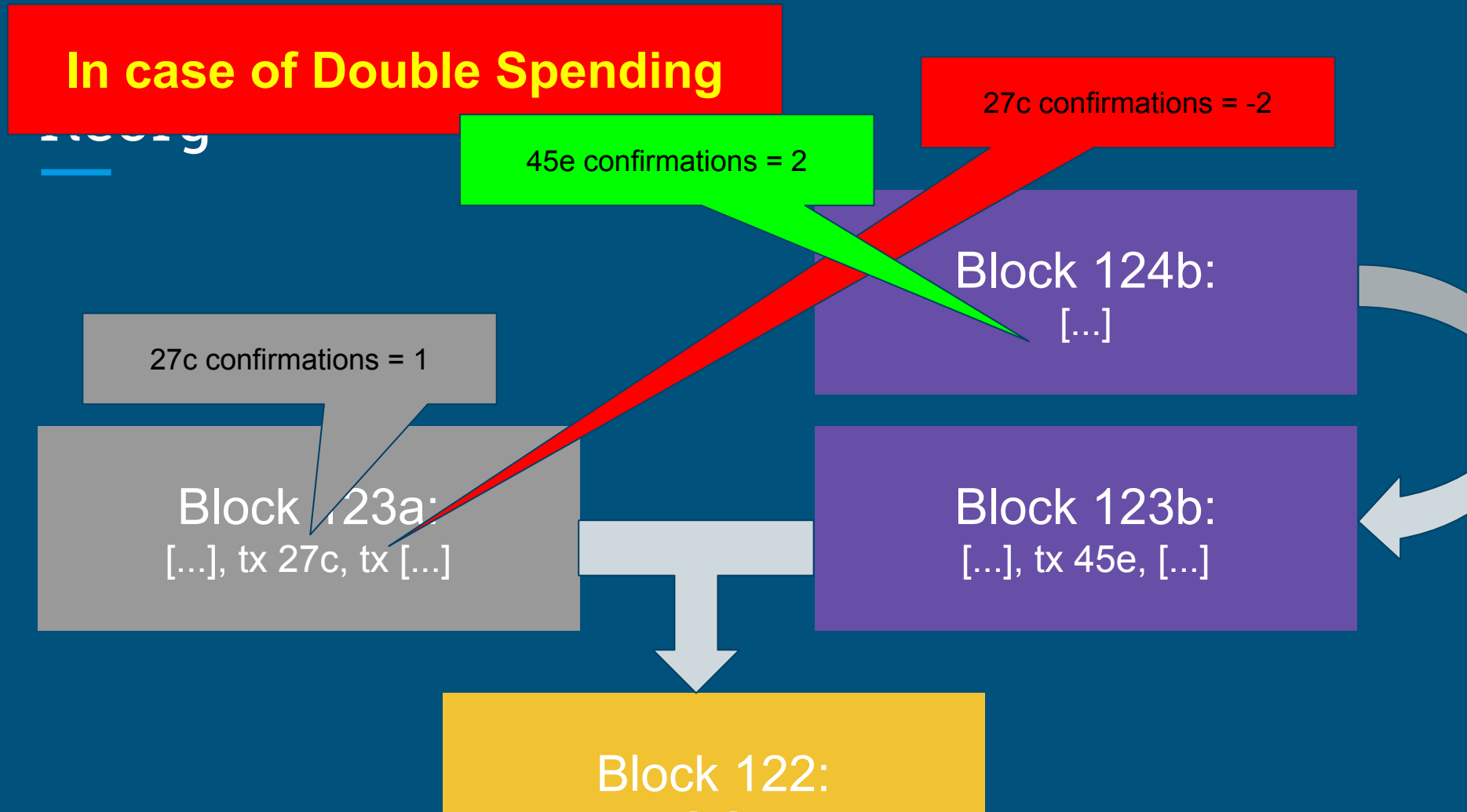
Reorg

TX 27c:
- **txin:**
  - **hash = 356**
  - **index = 0**
- txout:
  - addr = 1abc

TX 45e:
- **txin:**
  - **hash = 356**
  - **index = 0**
- txout:
  - addr = 1def

In case of Double Spending




In case of Double Spending
27c confirmations = -2
45e confirmations = 2
Block 124b:
[...]
27c confirmations = 1
Block 123a:
[...], tx 27c, tx [...]
Block 123b:
[...], tx 45e, [...]
Block 122:

Threat Model

Know your enemies

- Don’t worry too much for Proof-of-Concepts / Prototypes.
- But understand the dangers, and how to mitigate them.
- Of course it’s impossible to prevent everything!
- But nevertheless, know your enemies.

Example

A coin toss in a casino:

- HEADS you win $200, TAILS you lose $100

Probabilities might change depending whether the coin was produced by the government, or by the casino.

It’s a similar situation with Bitcoin development.

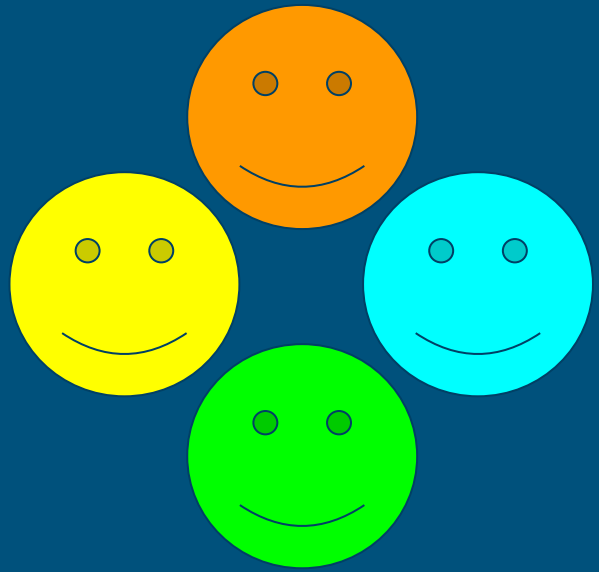
Threats on the Blockchain

- Many differences with general software development
- If you start developing without this understanding, **chances of problems occurring significantly increase**
- Beyond a level of “maybe it can happen”
- Currently tons of **unsafe software** is deployed!

Sybil Attacks

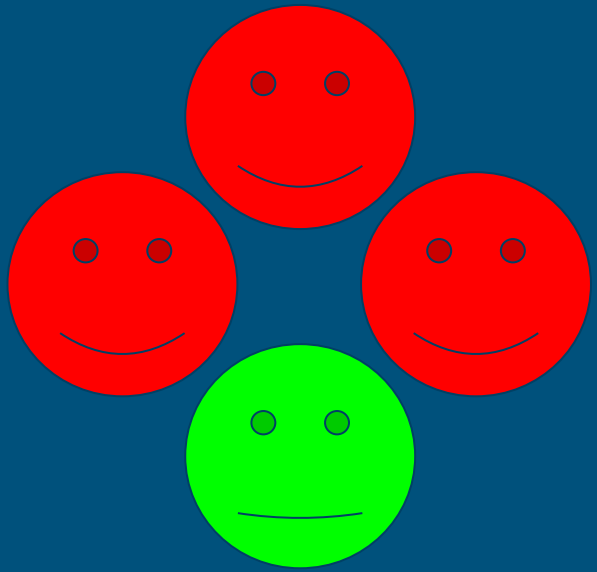
Example - Playing poker in a casino

Normally, everyone besides the dealer behaves for their own self-interest.

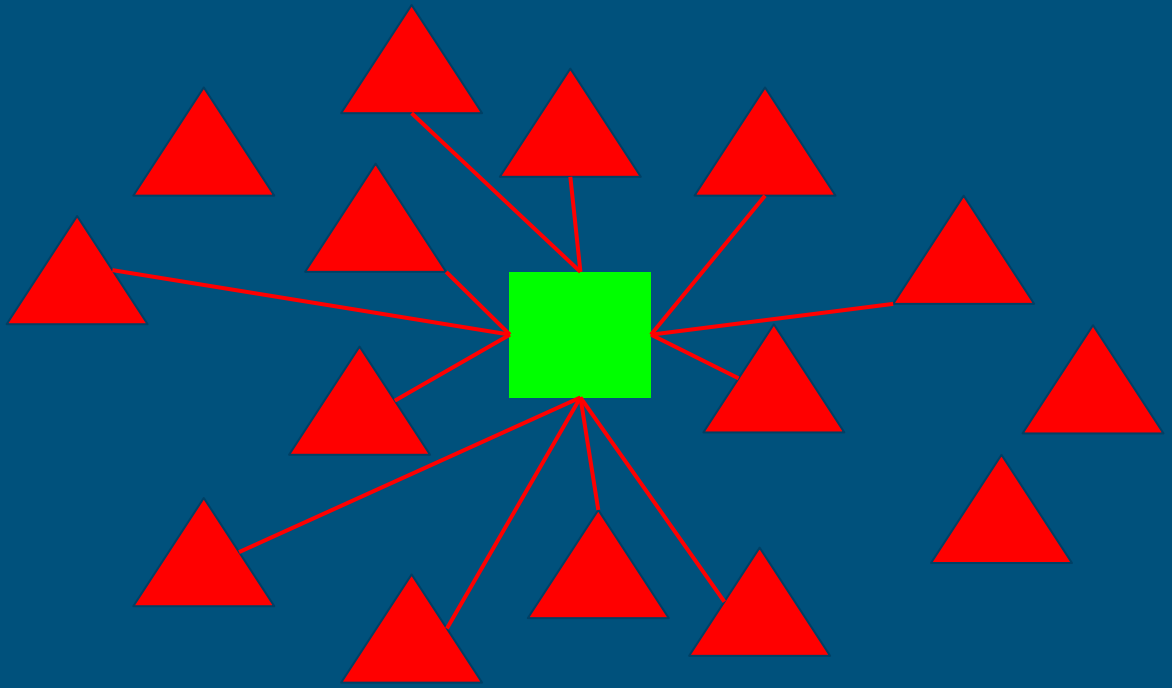


Example - Playing poker in a casino

What if everyone else (besides you) is employed by the casino.



On the Bitcoin Network



Sybil Attack - Types

- Attacking a single node
- Complete partition of the network
- Partial partitioning of the network

Sybil Attack - Methods

- Exhausting all the slots
- Setting up many nodes
- MITM (Man in the middle)

Sybil Attack - Prevention

- networkhashps dropping too much is suspicious
- Setup multiple nodes at multiple locations, and only connect to nodes you trust
    - Go through a VPN if possible
- BIP-150 (Peer Authentication)

The Double Spend problem

The Cheque example

You send out a request for a $300 cheque to an account (that already has $200 in it).

Combined, you assume you have $500 now. In actuality, that $300 cheque hasn't cleared yet.

Immediately, you want to send a cheque for $300, but the balance becomes -$100.

The Cheque example

If you check the cheque, there probably wouldn’t be a problem.

Bitcoin is also the same.

Double spending

It seems like you received BTC, but actually it didn’t happen.

Example: Create a Tx with a **Low Fee**, & a Tx with a **High Fee**.

You send the Target the Low Tx. When the Target gives you a confirmation, send the High Tx to the network.

（Real example）

Double spending

Double spending through malleability:

- Tx1 → Tx2 → Tx3 (connected, using 0-conf)
- Changing the Tx1 hash invalidates Tx2 & Tx3
- Anyone can do it (miner, relay node, …)

（solved with Segwit）

Double spending

The user’s wallet crashes.

Possible when trying to use a UTXO that was already used.

RBF (Replace By Fee)

In other words: base confirmation count on the value of the transaction. A $3 coffee can be 1 confirmation. A $1 mln transfer might be 144 ("one day").

Double Spending (under a Sybil Attack)

- Not relaying the actual Tx to target.
- Not relaying the block containing the actual Tx to target.
- Showing the fake Tx only to the target.

Business Logic problem

Business Logic problem

Unsafe environment

Access to Private Keys (Bitfinex wallet)

- Atomic operations

Web wallet Submit problem

- Programming error

Not-so-random random (using a nonce twice or more, like in recent WPA security issue)

Verification & Trust

Verification & Trust

- SPV Wallet
- Block Explorer

Cryptography fails

Cryptography fails

- Weakness in Elliptic Curve (EC)
- Address reuse → Bad
- Even if EC is completely busted, damage is limited to pub key hash, and there are still ways to make Bitcoin safer. (assuming hash algorithm is not broken (low chance))

Consensus problem

Consensus problem

- When the network forks. Example: BIP-66 (PPCOIN)
- No problem if you setup multiple nodes in multiple locations. And use multiple versions at the same time.
- In the case of a Fork, send Tx on both forks.

@DG Lab - Karl-Johan Alm

Twitter: @kallewoof