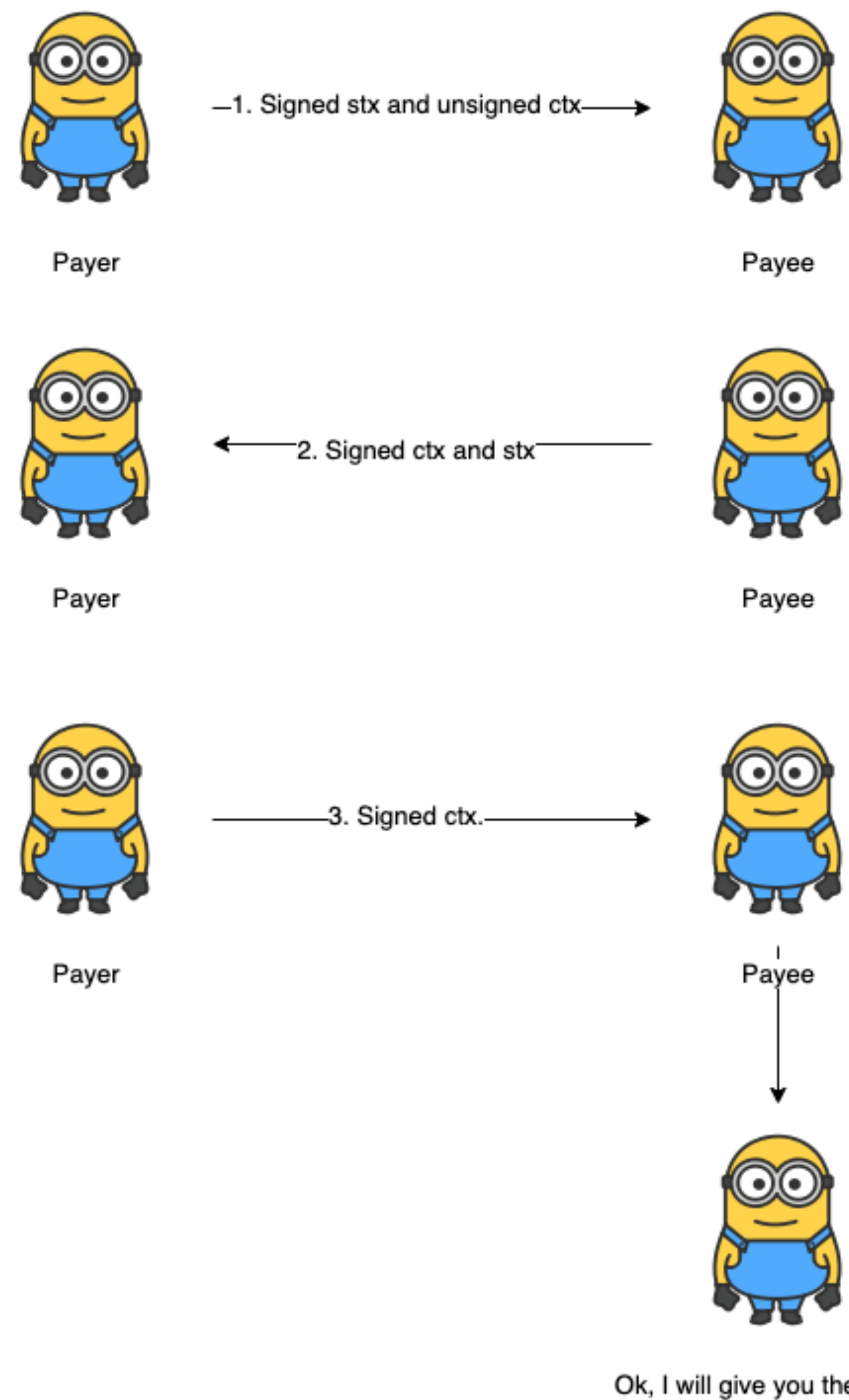


The update of GPC

Zhichun Lu

Make payment



Msg detail

Msg_send: 6

Require: lp, remote_port, id, amount

do:

- Construct the new stx and ctx.

Ensure:

- `msg = { id, 6 || "payment", ctx, stx, amount }`
- `doc = { id, key, fund_tx, status:7, ctx_pend, stx_pend. }`



Payer

—1. Signed stx and unsigned ctx—→



Payee

Msg detail

Msg_rcv: 6

Require:

do:

- The signature is valid.
- The information are right. (just create the info by myself and check).

Ensure:

- Whether the stx and ctx are valid. If not, just report errors (using msg type 0).



Payer

—1. Signed stx and unsigned ctx—→



Payee

Zhichun Lu

Msg detail

Msg_send: 7

Require: lp, remote_port.

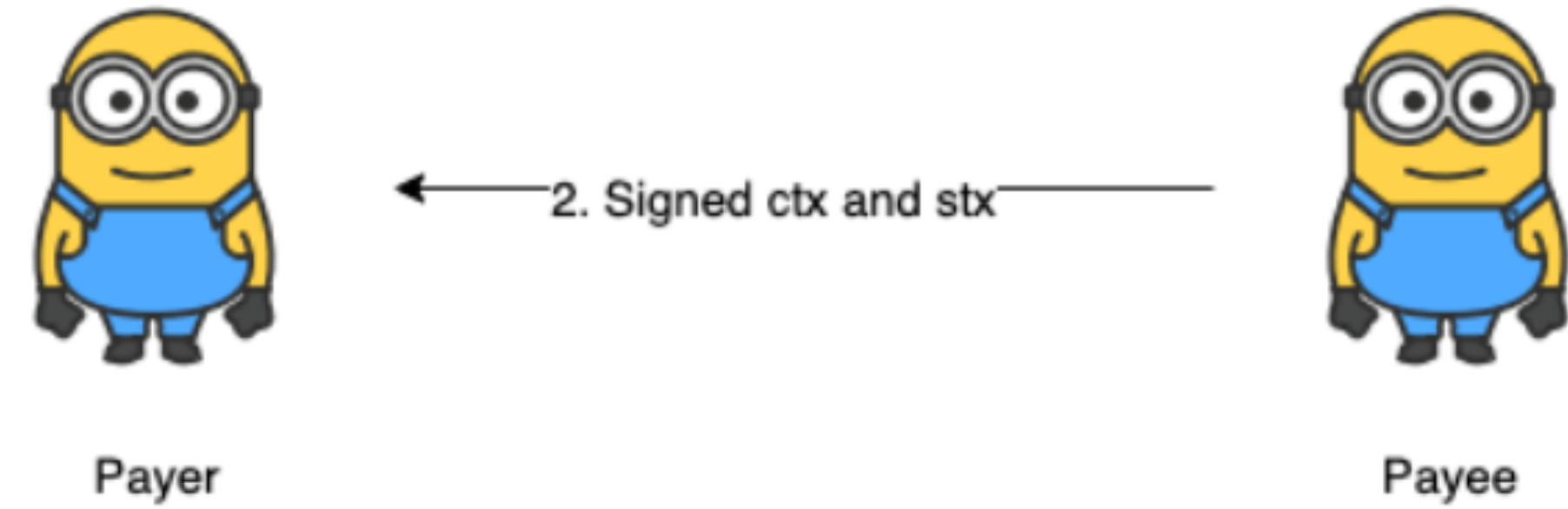
do:

- Sign the new stx and ctx.

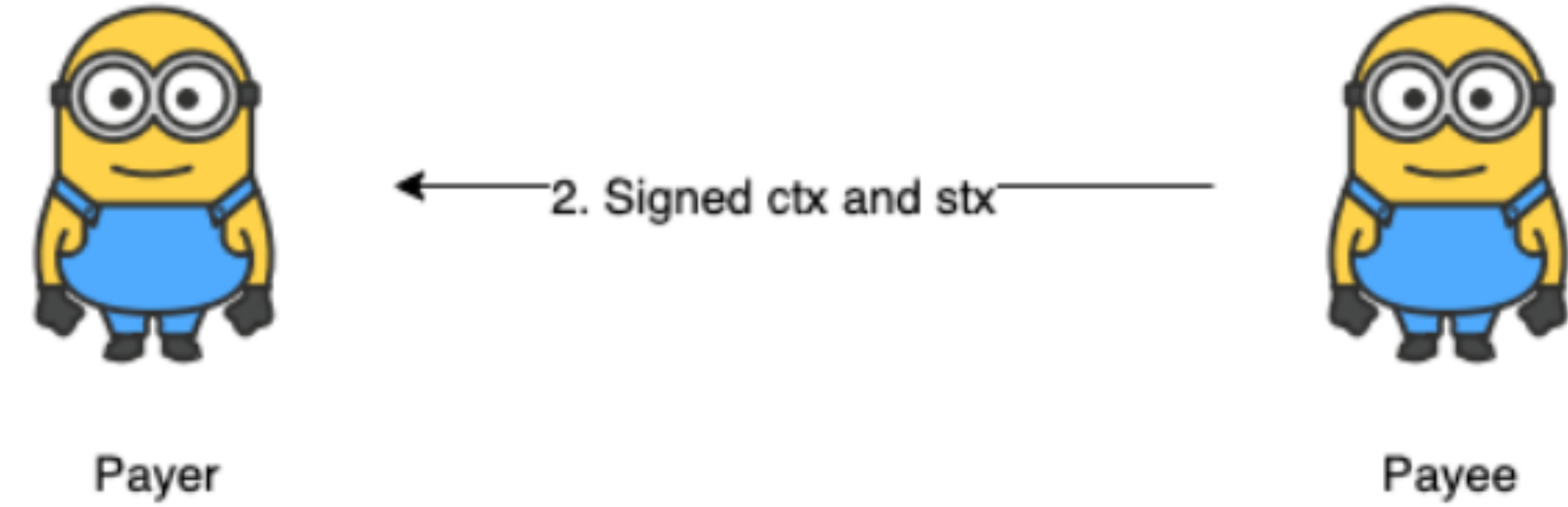
Ensure:

- `msg = { id, 7 || ctx, stx, amount }`
- `doc = { id, key, fund_tx, status:8, ctx_pend, stx_pend. }`

Zhichun Lu



Msg detail



Msg_rcv: 7

Require:

do:

- The signature is valid.
- The information are right. (just create the info by myself and check).

Ensure:

- Whether the stx and ctx are valid. If not, just report errors (using msg type 0).

Zhichun Lu

Msg detail

Msg_send: 8

Require: lp, remote_port.

do:

- Sign the ctx.

Ensure:

- $msg = \{ id, 8 \parallel ctx, amount \}$
- $doc = \{ id, status:6, ctx, stx. (updated) \}$



Payer

3. Signed ctx. →



Payee

Msg detail

Msg_rcv: 8

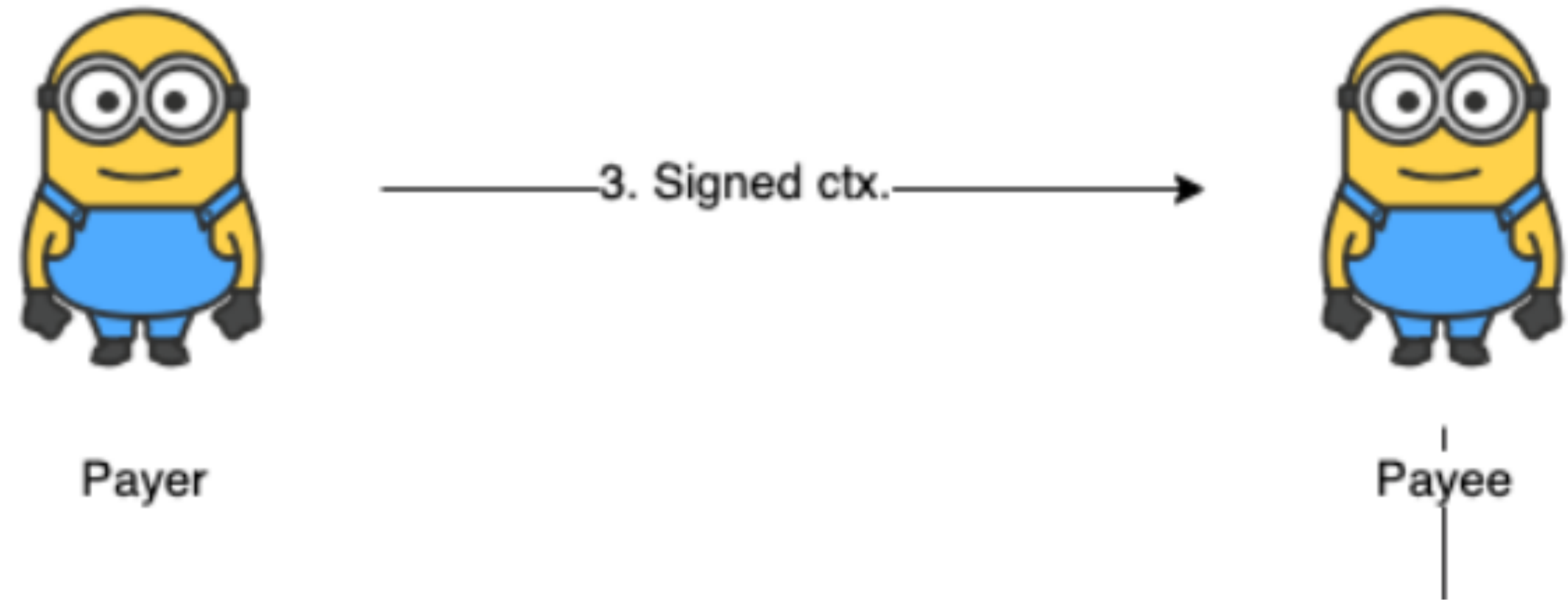
Require:

do:

- The signature is valid.
- The information are right. (just create the info by myself and check).

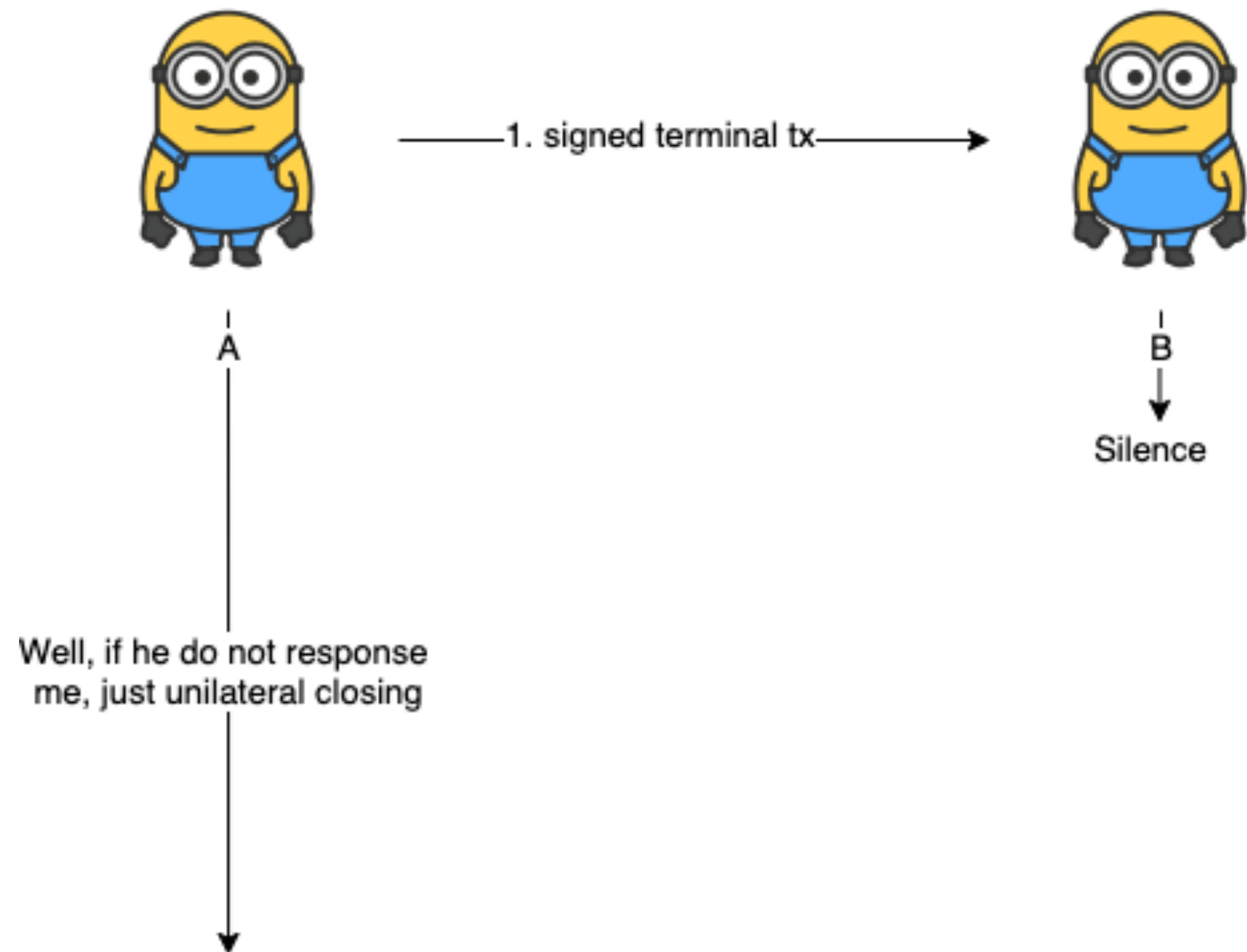
Ensure:

- Whether the ctx is invalid. If not, just report errors (using msg type 0), and then update the doc.



Zhichun Lu

The good case.



Zhichun Lu

Msg detail

Msg_send: 6

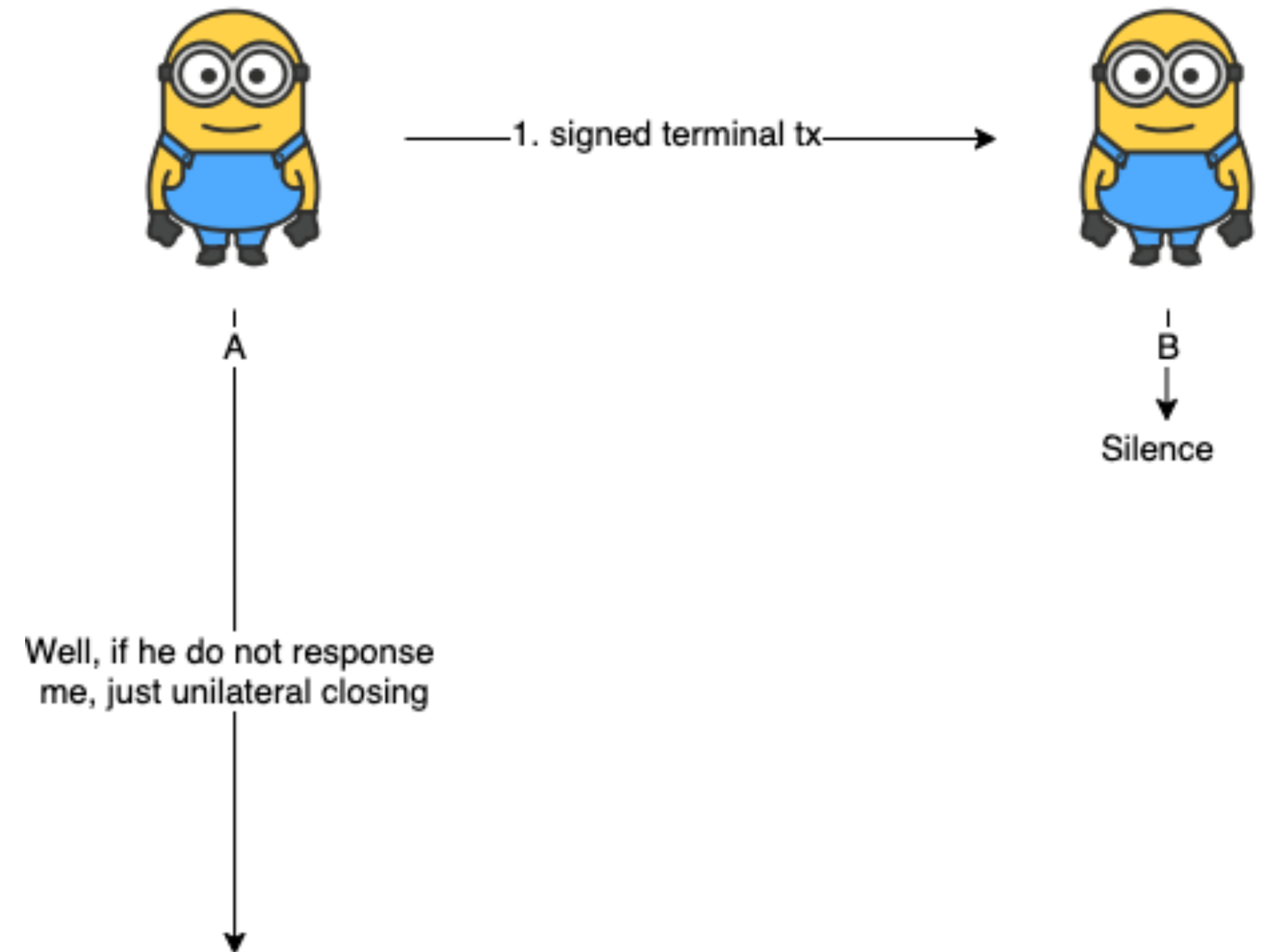
Require: Ip, remote_port, id

do:

- Construct terminal tx.

Ensure:

- `msg = { id, 6 || "closing", terminal_tx. }`
- `doc = { id, status:6, closing_time: current_height +100 }`



The monitor

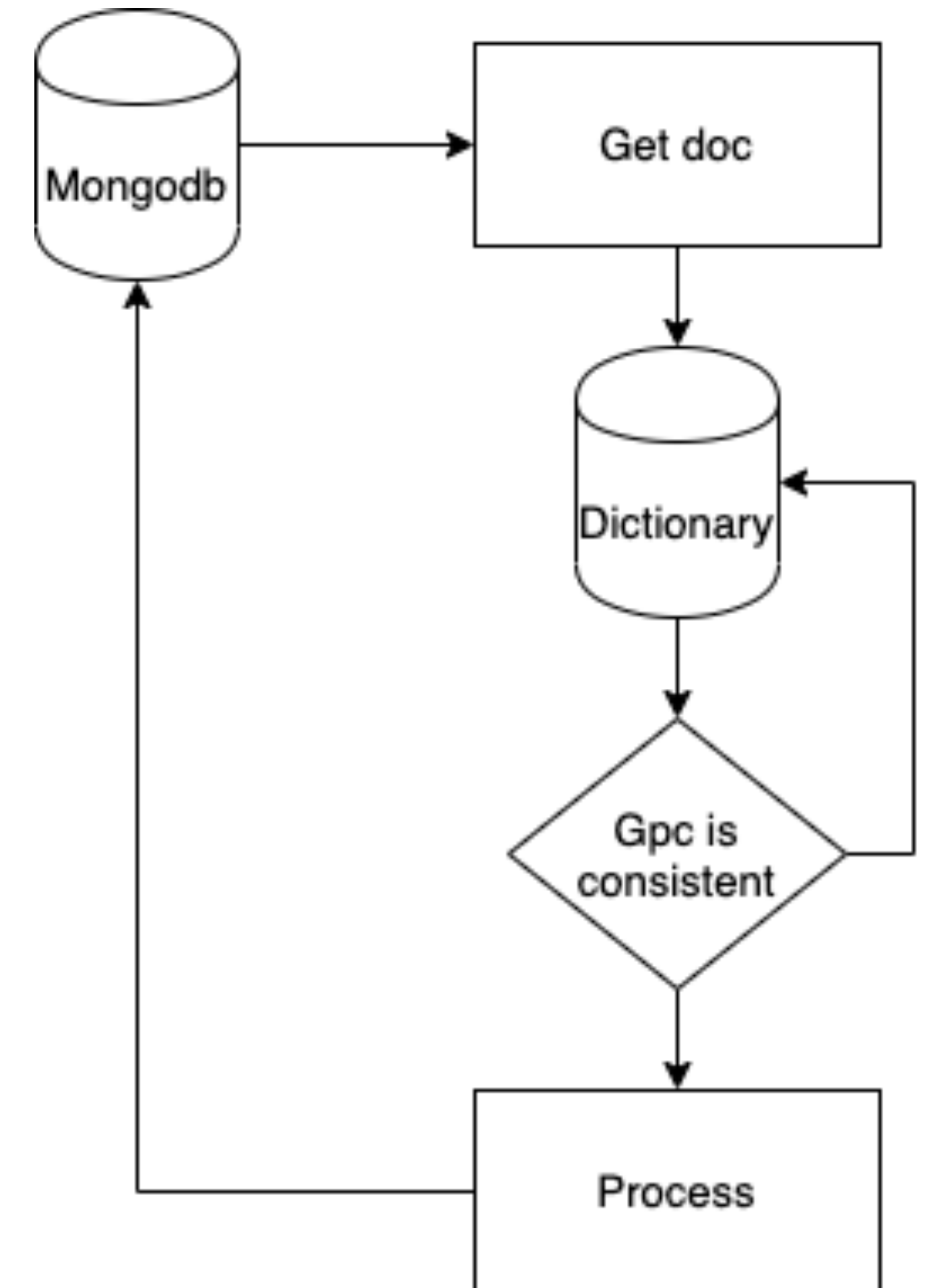
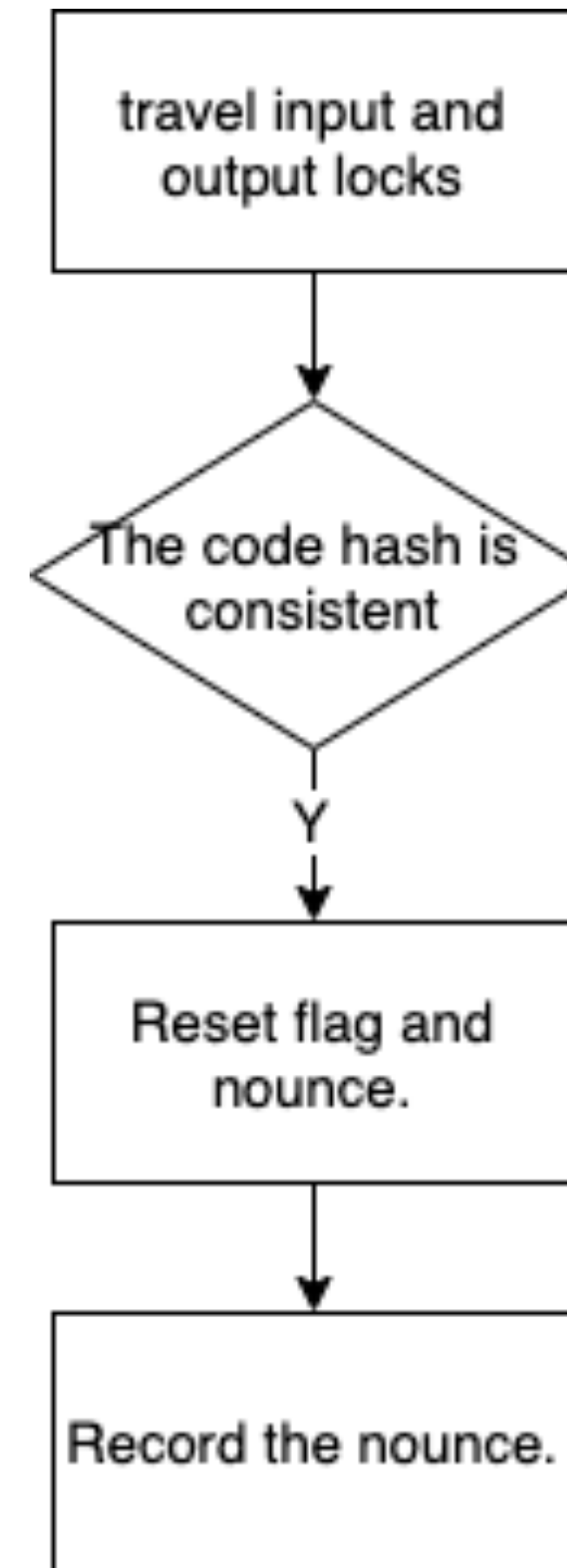
Stage.

0: Initial

1: Allow payments.

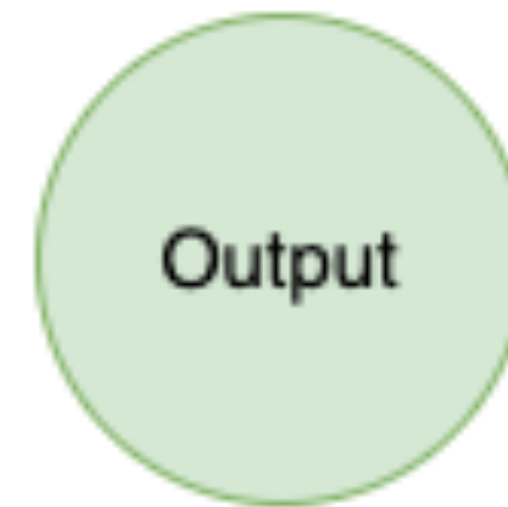
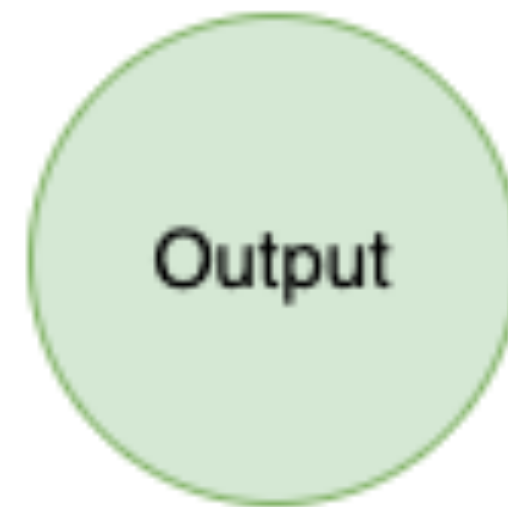
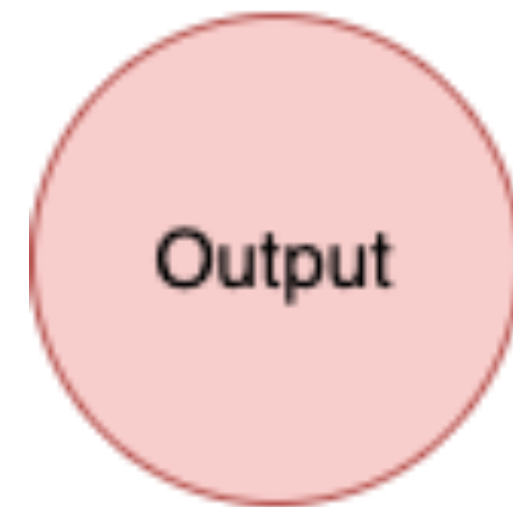
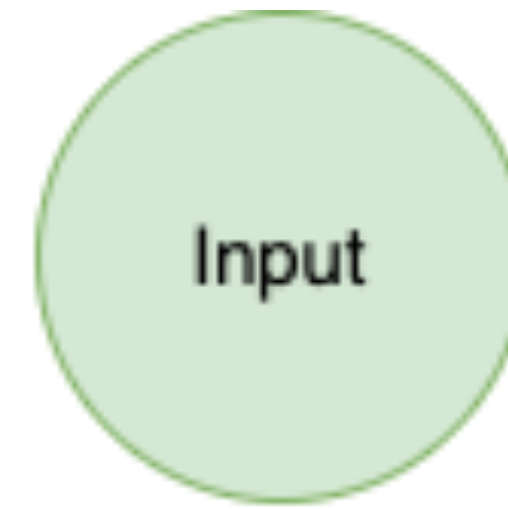
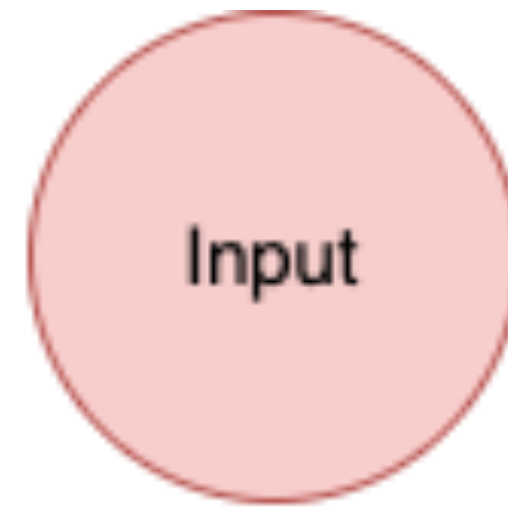
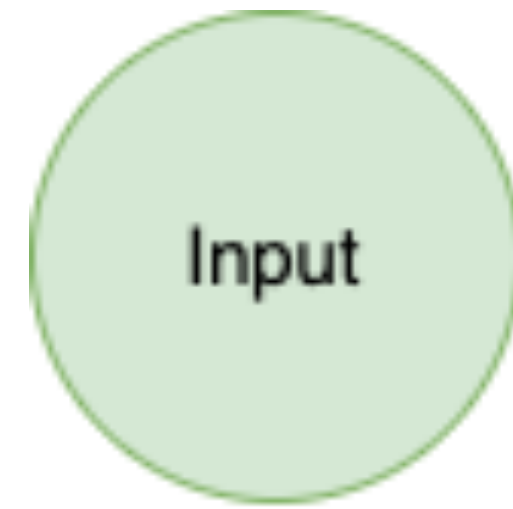
2: Racing

3. Settlement



Zhichun Lu

The monitor



Settlement

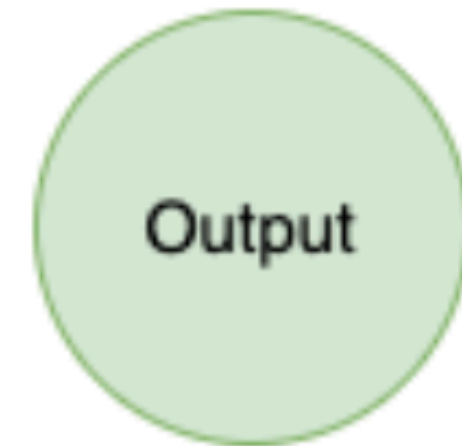
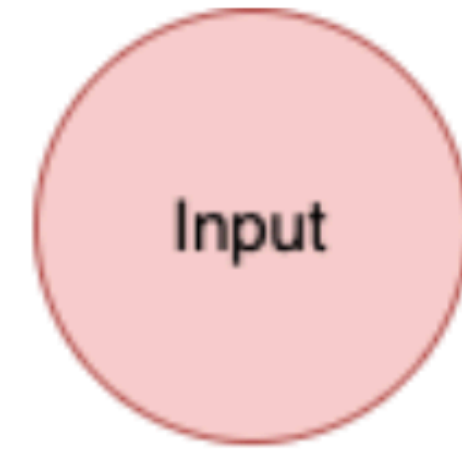
fund

Closing

Zhichun Lu

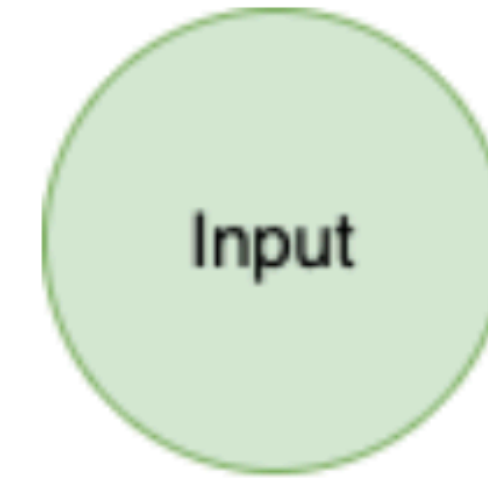
The monitor

Just set the stage to 1, and convert the output to input format, and store it.

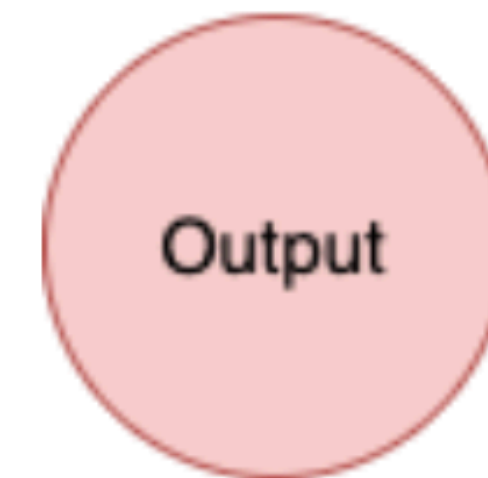


fund

The monitor



Just delete the doc.



Settlement

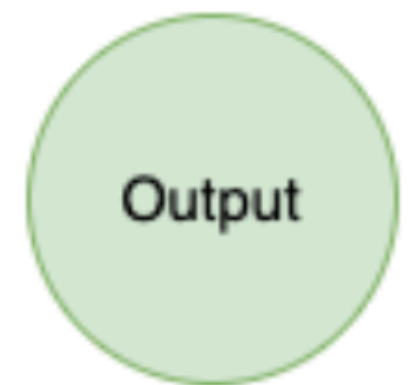
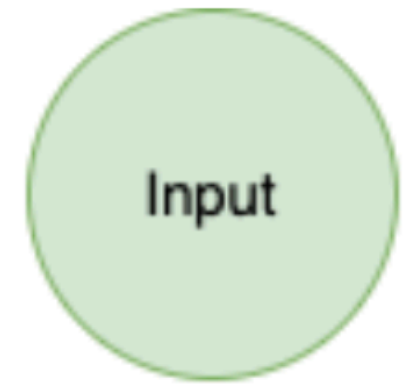
Zhichun Lu

The monitor

If `remote.output.nounce < nounce_local`:
convert the output and send latest ctx.

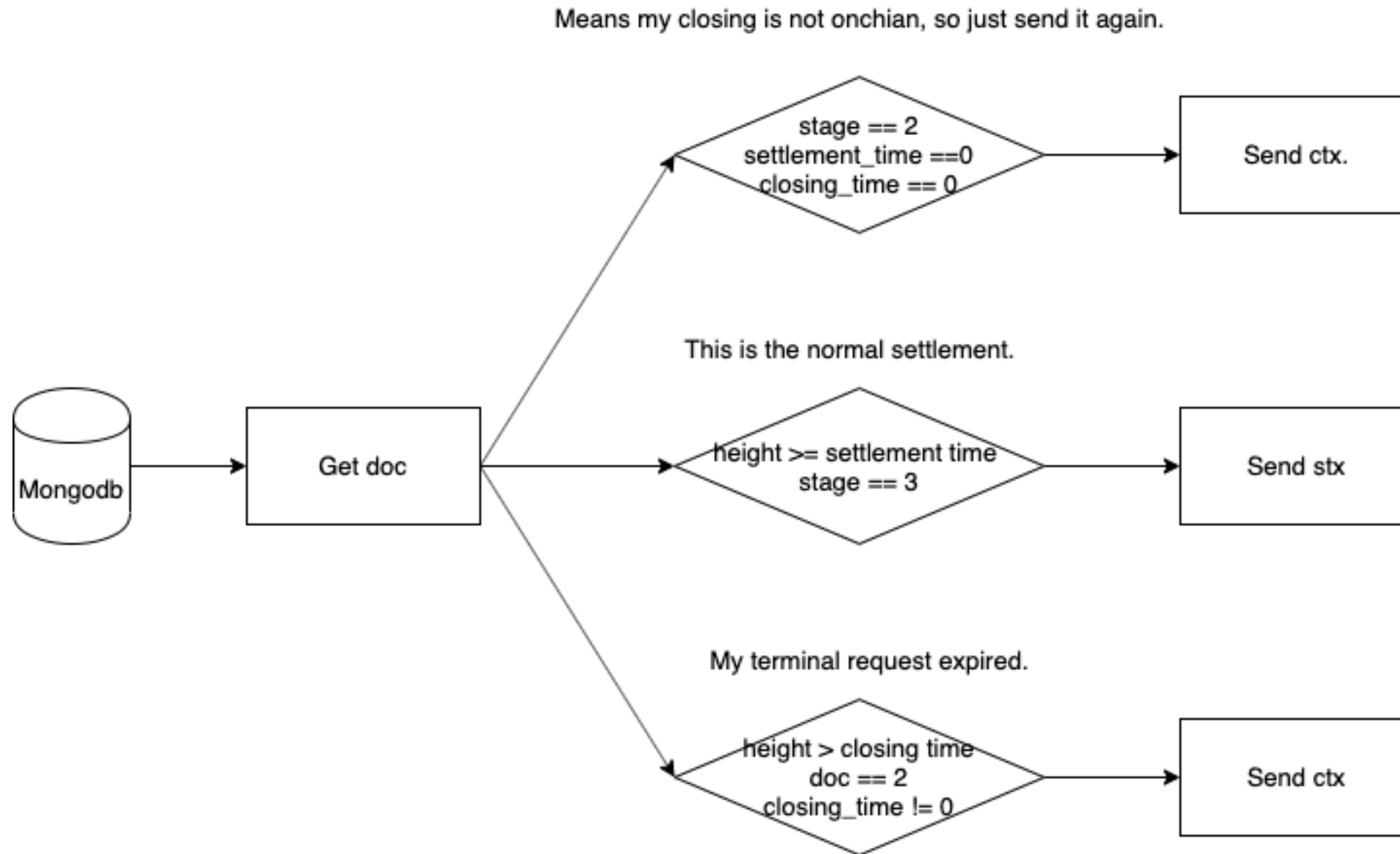
If `remote.output.noucne == nounce_local`:
calculate the effective date of settlement tx.
convert the output.

If `stx_pend != 0 && remote.output.nounce - nounce_local == 1`:
calculate the effective date of settlement tx.
convert the output.



Closing







The monitor



Overview

```
/Users/ZhiChunLu/prototype/libs/communication.rb:787: warning: assigned but unused variable - remote_pubkey
/Users/ZhiChunLu/prototype/libs/communication.rb:788: warning: assigned but unused variable - local_pubkey
/Users/ZhiChunLu/prototype/libs/communication.rb:792: warning: assigned but unused variable - ctx
/Users/ZhiChunLu/prototype/libs/communication.rb:825: warning: assigned but unused variable - iter
/Users/ZhiChunLu/prototype/libs/chain_monitor.rb:90: warning: assigned but unused variable - remote_input_nounce
/Users/ZhiChunLu/prototype/libs/chain_monitor.rb:206: warning: assigned but unused variable - nonsense
Commands:
  GPC closing_channel <pubkey> <id> ...
  GPC help [COMMAND] ...
  GPC init <private-key> ...
  GPC listen <pubkey> <port> ...
  GPC make_payment --pubkey <public key> --ip <ip> --port <port> --id <id> --amount <amount> --amount=AMOUNT --id=ID --ip=IP --port=PORT --pubkey=PUBKEY ...
  GPC monitor <public key> ...
  GPC send_closing_request --pubkey <public key> --ip <ip> --port <port> --id <id> --id=ID --ip=IP --port=PORT --pubkey=PUBKEY ...
  GPC send_establishment_request --pubkey <public key> --ip <ip> --port <port> --type <type script> --amount <amount> --fee <fee> --since <since> --amount=AMOUNT... if
→ client1 git:(master) ×
```

Commits on Jul 8, 2020

<div>update readme</div> <div> git authored and git committed 2 days ago</div>	<div> 5878894</div> <div><></div>
<div>ckb vertsion done</div> <div> git authored and git committed 2 days ago</div>	<div> 5e1500c</div> <div><></div>
<div>ckb version done</div> <div> git authored and git committed 2 days ago</div>	<div> 8ad9f32</div> <div><></div>

Zhichun Lu

Discussion

- One-way channel.
- Both witness and lock args needs id.
- I want to try to implement punishment, needs asymmetric commitments.
- I want to try to design multiple GPC output, maybe give a value to denote the number of GPC outputs.
- Type script verify.

Zhichun Lu

TODO

- Compatible with UDT version.
- Unit test (only the CLI? Or both CLI and customized attacker?)
- Punishment & multiple GPC outputs.
- Multi-hop payments.

Zhichun Lu