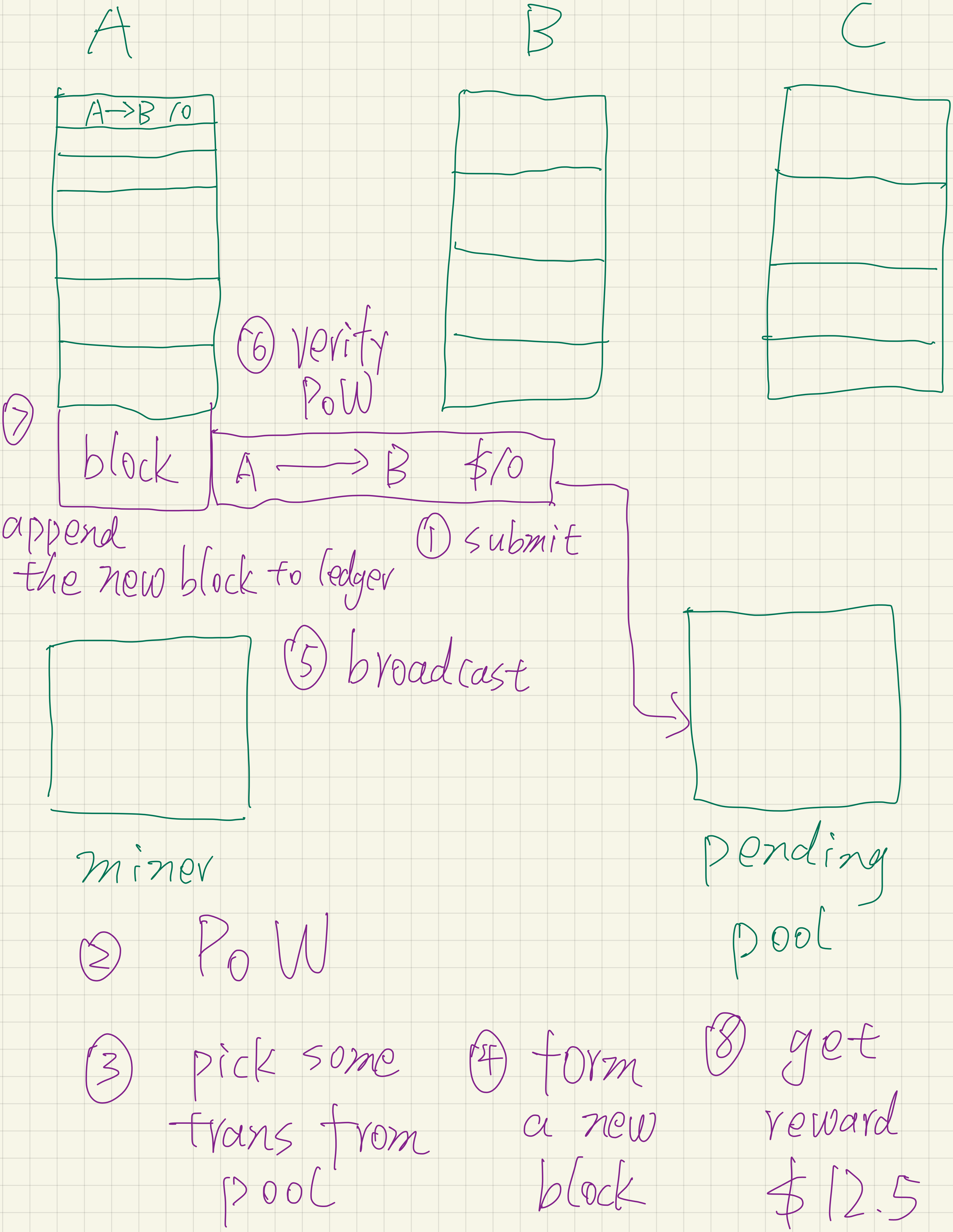
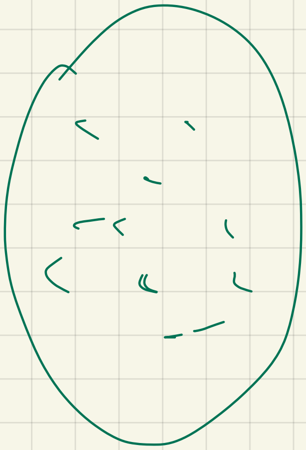

A
B
C
A→B 10
⑥ verify PoW
⑦
block
A → B $10
① submit
append the new block to ledger
⑤ broadcast
miner
pending pool
② PoW
③ pick some trans from pool
④ form a new block
⑧ get reward $12.5

PoW:

$h(x)$



$$\xrightarrow{h(x)} 16\text{-bits}$$

domain

$$0 \leq h(x) < 2^{16}$$

1 unit hash = work

$$h(\underset{\substack{\uparrow\\ \text{response}}}{x}) = \underbrace{0000\ \ 000000}_{\substack{\text{challenge}\\ \to 10}}\ \underbrace{\cdots\cdots}_{6}$$

$$2^{10} =$$

$$h(x) = \underbrace{0}_{1}\ \underbrace{-\ -\ -\ -\ -\ -}_{15}$$

$$2^{1} = 2$$

$$h(x) = \underbrace{00}_{2}\underbrace{\text{- - - -}}_{14}$$

$$2^2 = 4$$

10 zero $\Rightarrow 2^{10}$

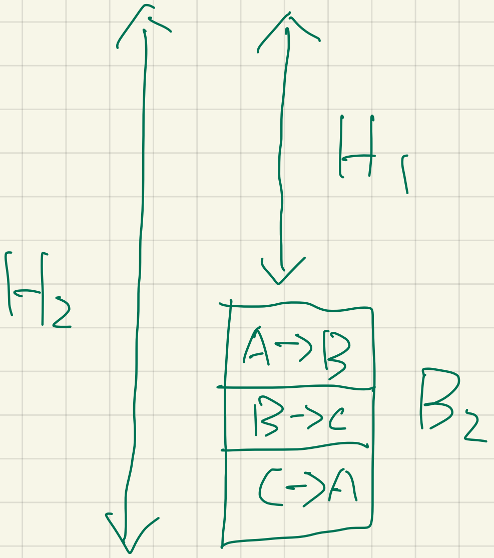
24 zero $\Rightarrow 2^{24} \Rightarrow \dfrac{10\text{ mins}}{\text{block}}$

6 zero $\Rightarrow 2^{6}$

hard to find a PoW

easy to verify a PoW

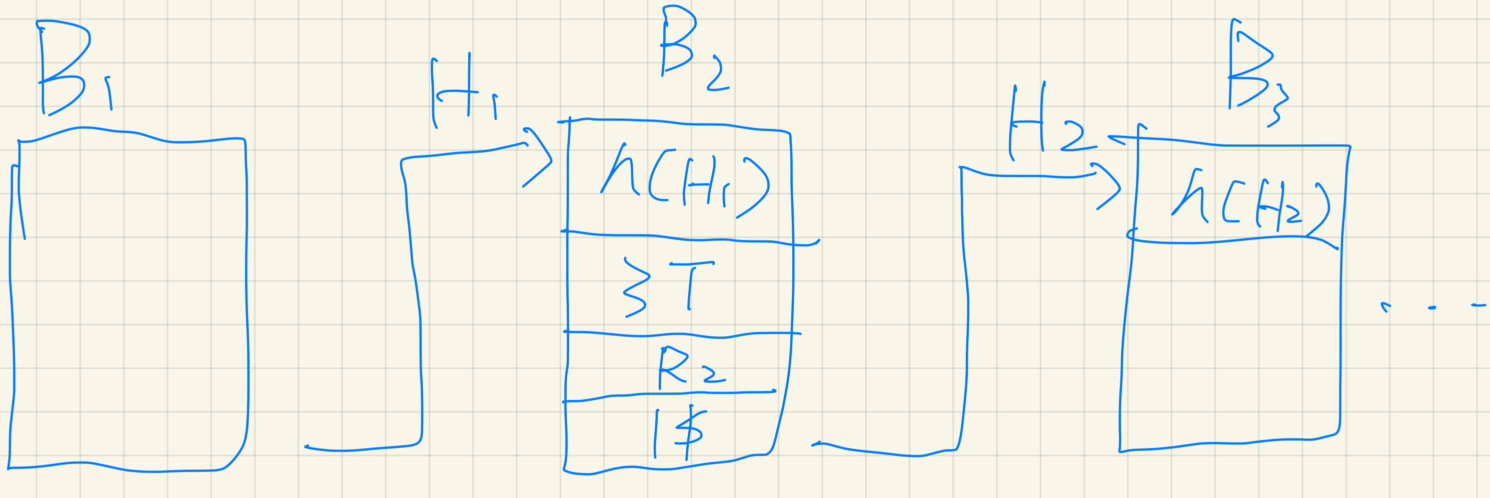
Block :



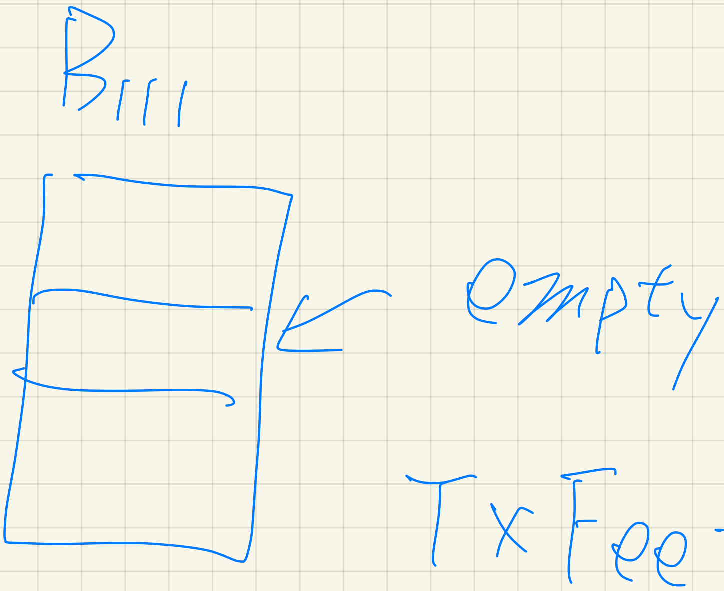
$$[h(H_1), H_2 - H_1] = C_2$$

☆

$$h(\underset{\uparrow}{C_2} \| \underset{\uparrow}{R_2}) = \underbrace{0 \ldots 0}_{80}$$



2008

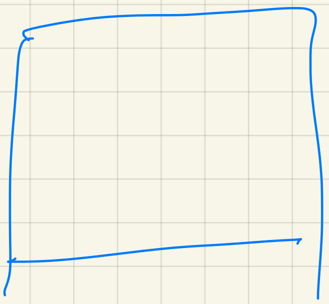


TxFee + $1

Starbucks

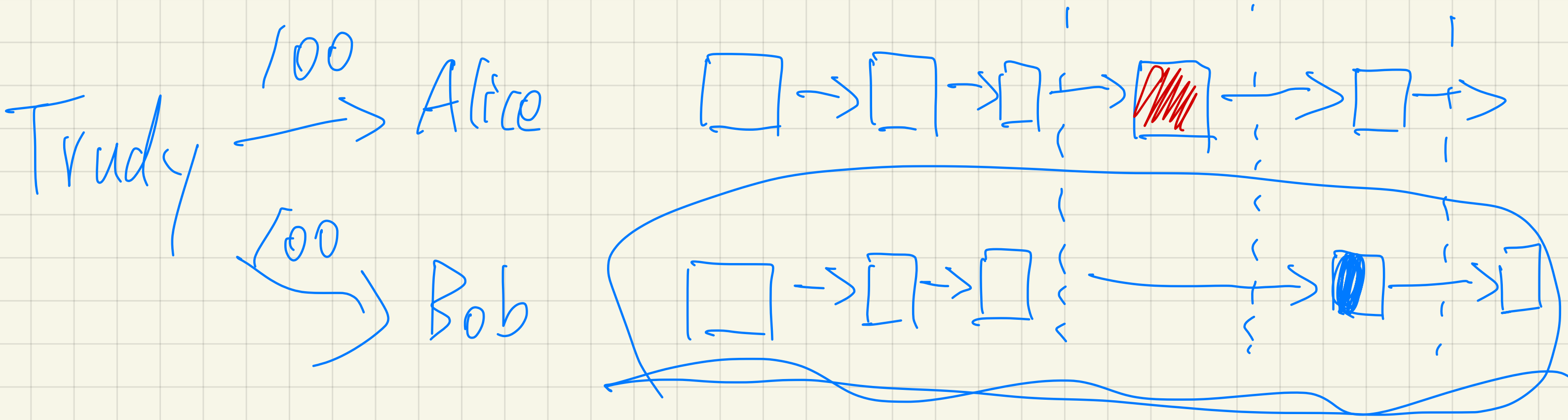
$3 + $5 { 2 ~ 3 blocks
25 mins

Tesla

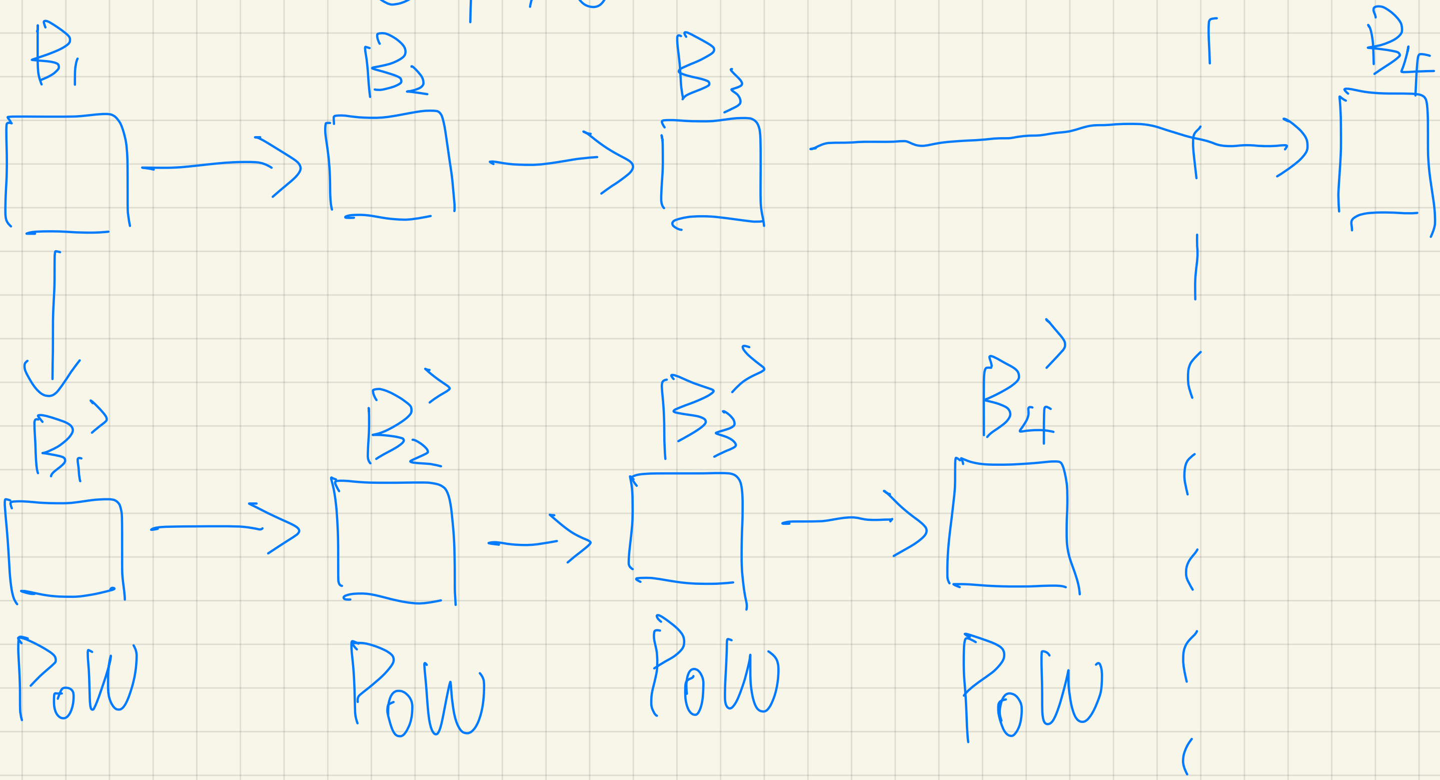


$$\frac{\sim \text{mb}}{\sim \text{K}}$$

double spending



51% attack:



$$T(3+1) < T(1)$$

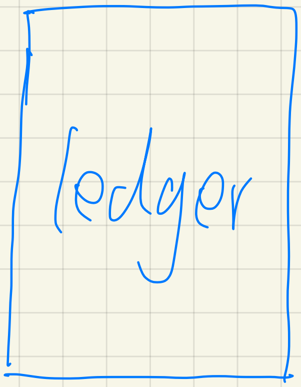
$$T(\underset{\sim}{3}+n) < T(n)$$

$$\boxed{\underline{T(n) < T(n)}}$$

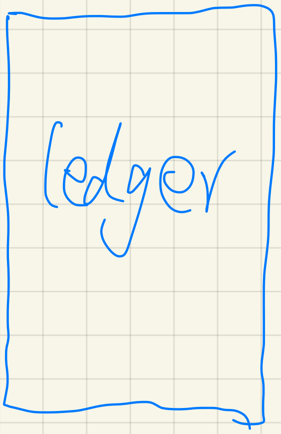
$$\underline{5\%}$$

$$\boxed{\underline{70\%}}$$

$$\begin{tikzcd}
 & C_1 \\
C \arrow[ru] \arrow[r] \arrow[rd] & C_2 \\
 & C_3
\end{tikzcd}$$



User
- key
- balance
- address



smart contract
- key
- balance
- address
- program
- data

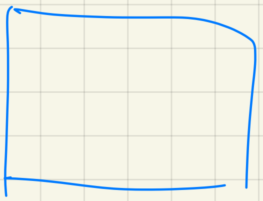
User
- normal user
- smart contract

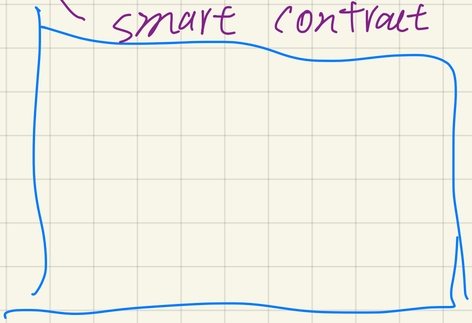
Transaction

send	receiver	$/0 ether	fee			
send	receiver	$/0	fee	add	2	3

send, receiver:
- normal
- smart contract

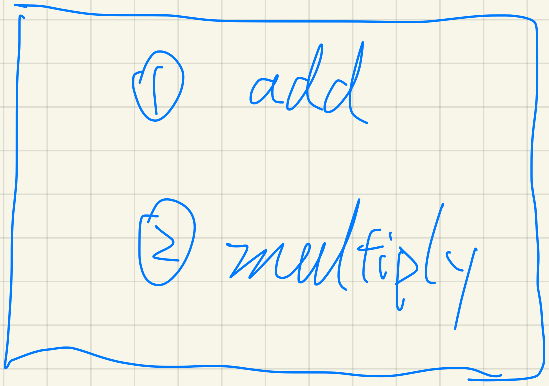
miner





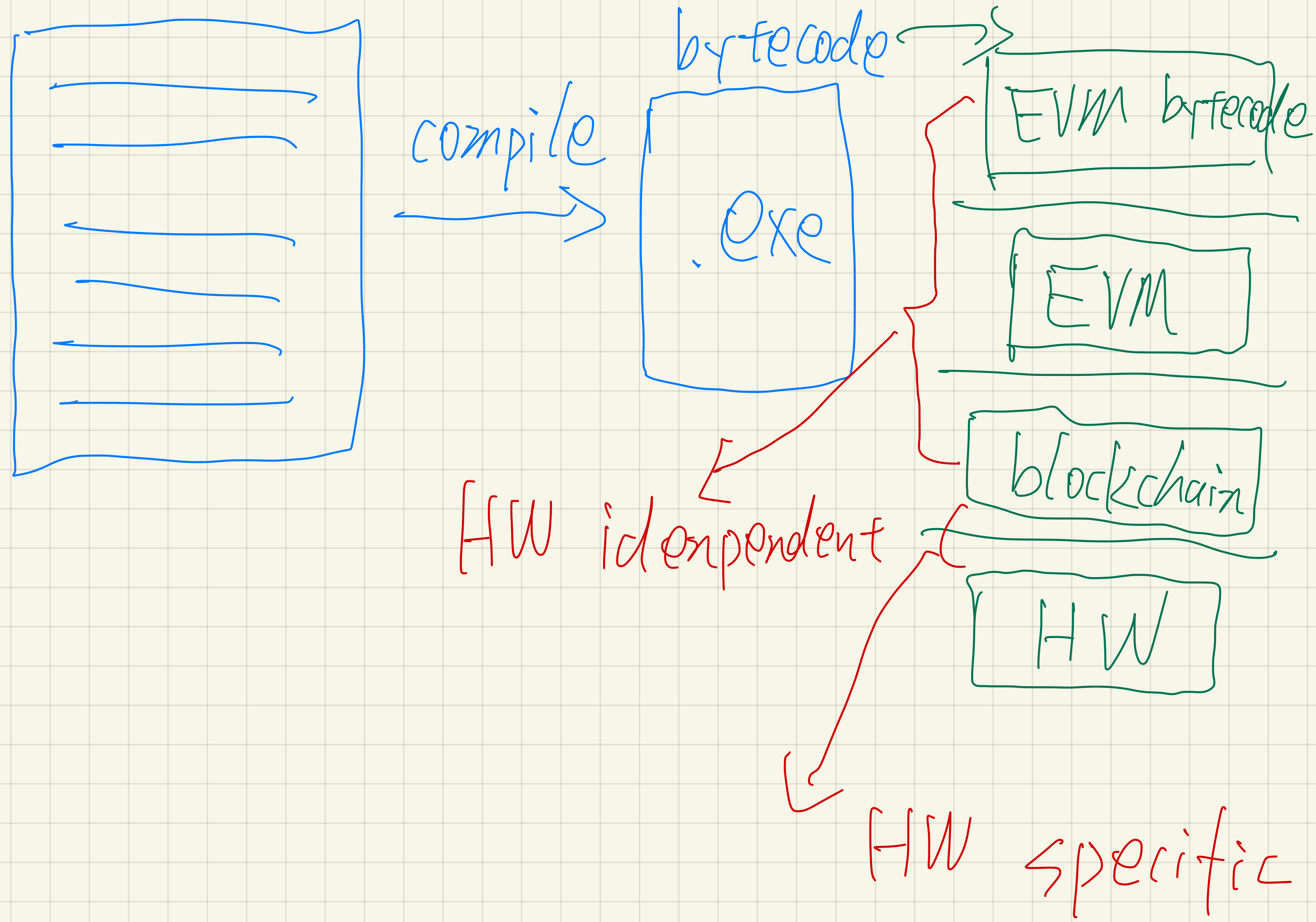
trans pending pool

Smart contract

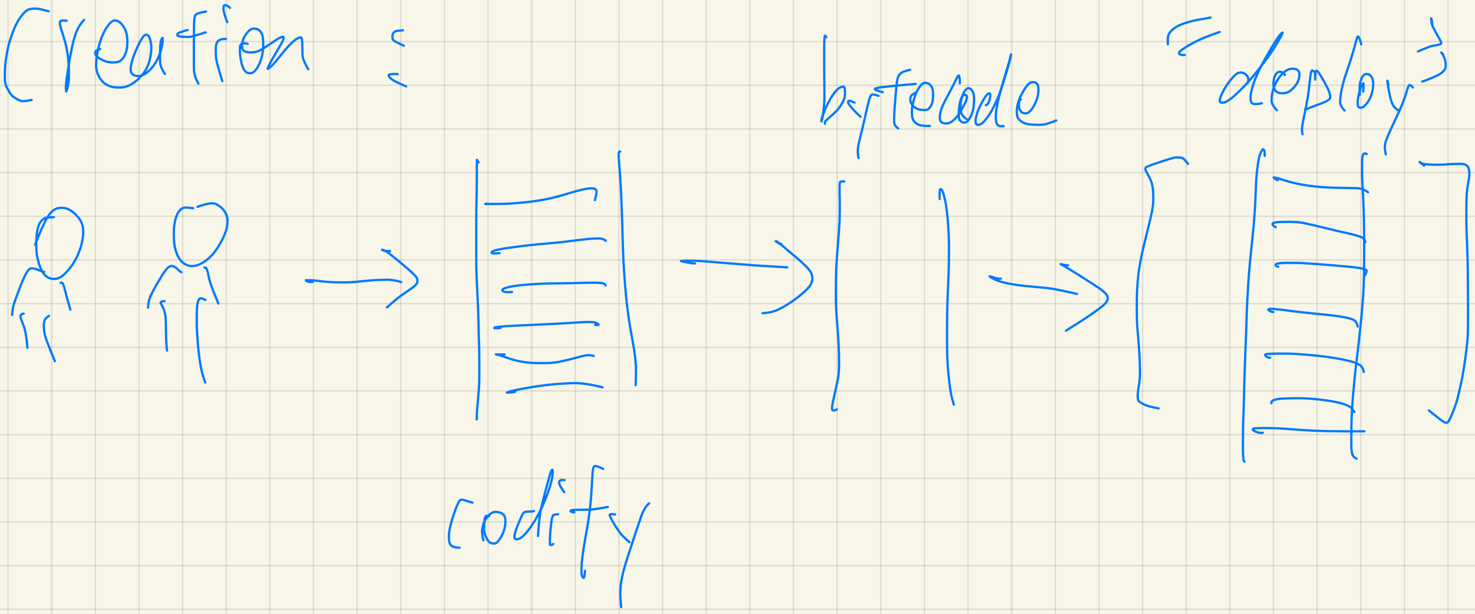


```
int add(a,b)
{
    return a+b;
}
```

solidity : ⟷ JS


bytecode
compile
.exe
EVM bytecode
EVM
blockchain
HW
HW idenpendent
HW specific

Creation :



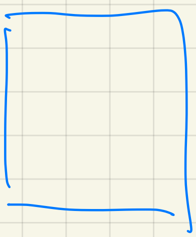
Pro : ① transparent ← every audit ; bytecode ↓ source code

② not tamperable

Con : cannot patch

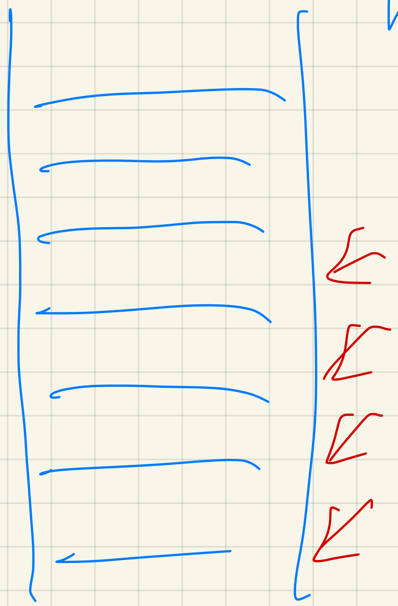
① destruct

② new



miner

PoW + execute contract

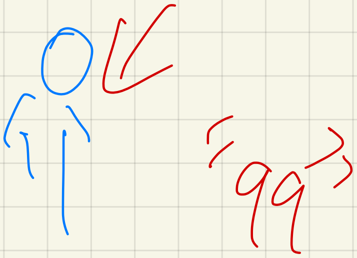


bomb

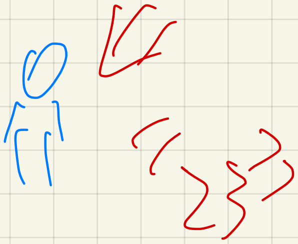
“infinite loop”

gas fee

fee + $10 + N×gas

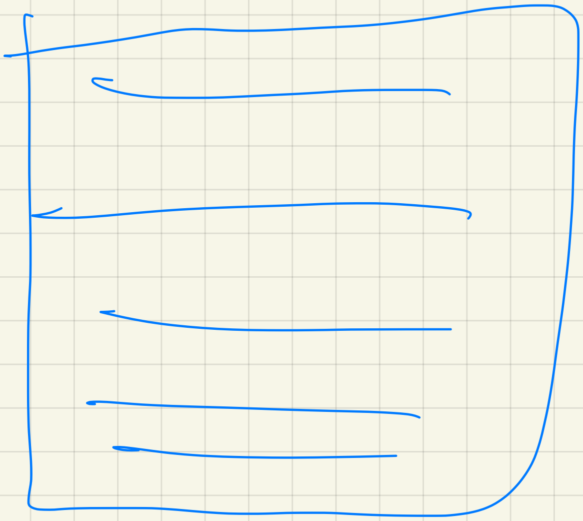


```
modify(99)

read( )
```



```
modify(23)

read( )
```

```
int a=1;
modify( b)
{
    a=b;
}
read(    )
{
    return a;
}
```



```
modify(99)
modify(23)
read( )

read( )
```

23

```
modify(23)
read( )
modify(99)

read( )
```

"99"

"race condition"