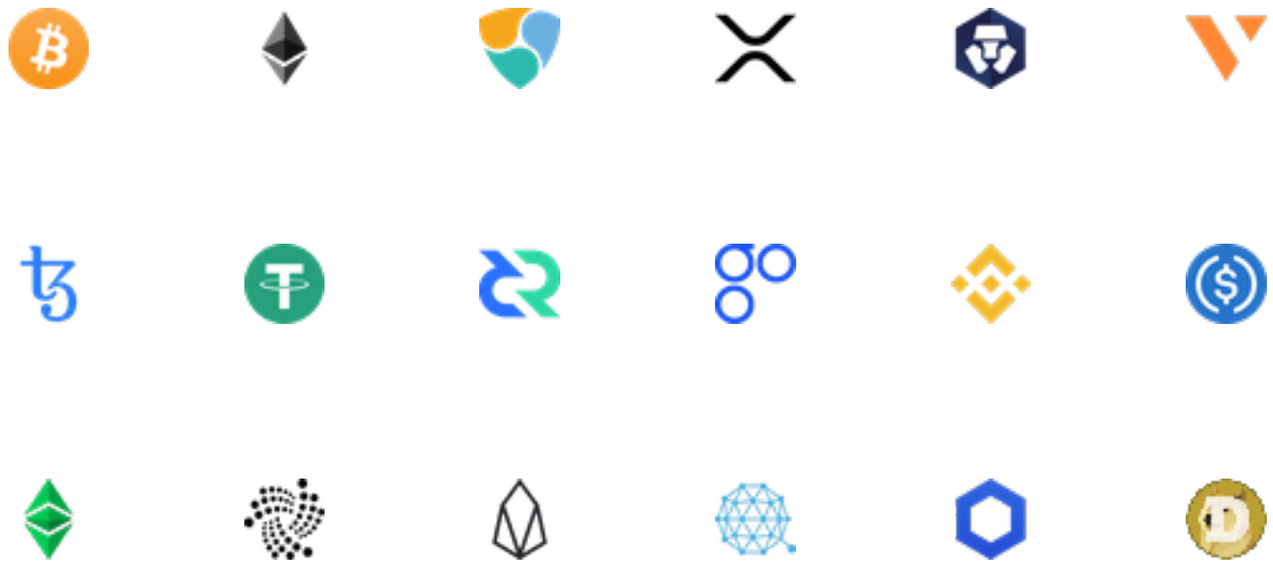
Who Spent My EOS?

On the (In)Security of Resource Management of EOS.IO

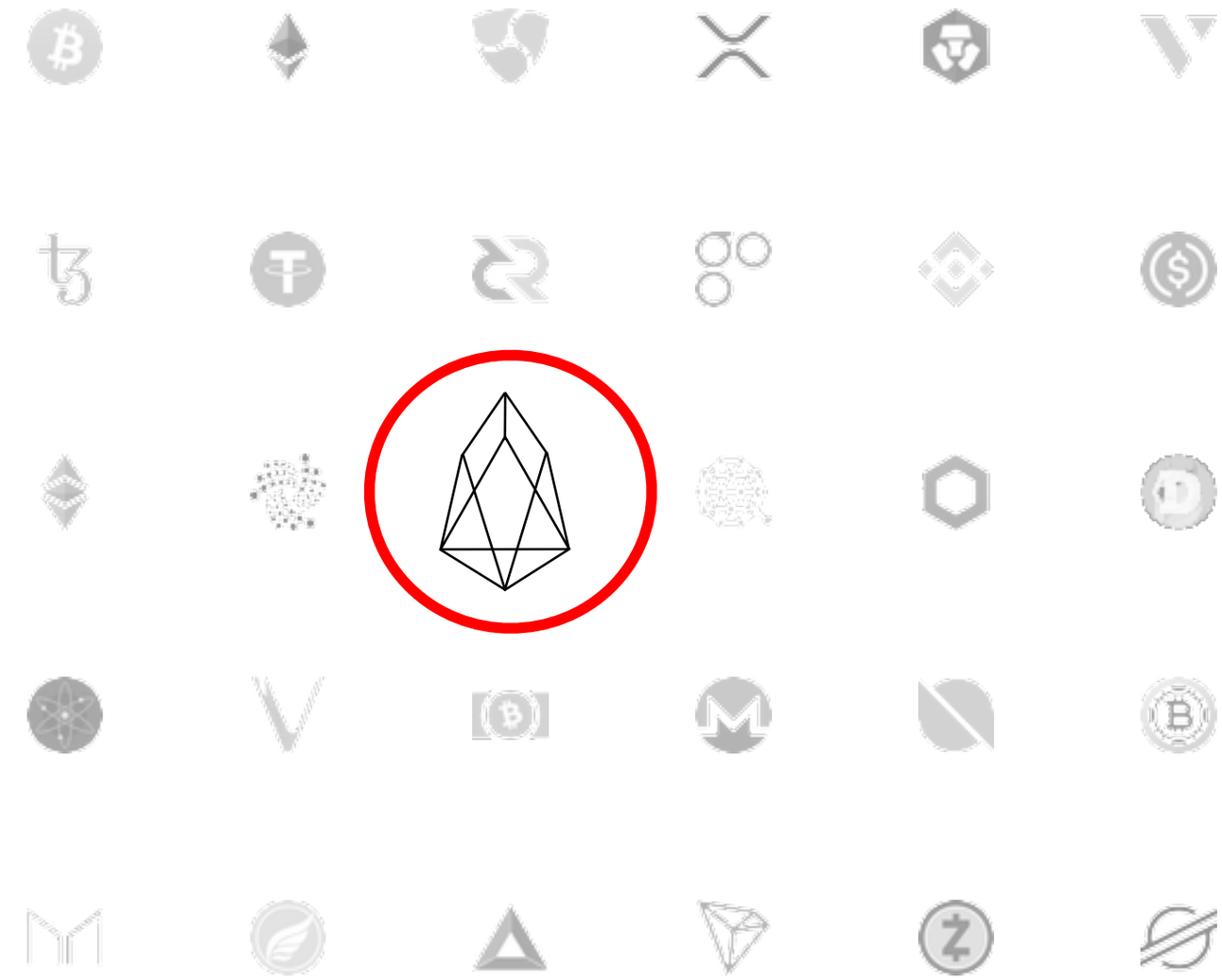
Sangsup Lee, **Daejun Kim**, Dongkwan Kim, Sooel Son, Yongdae Kim @KAIST

Abstract



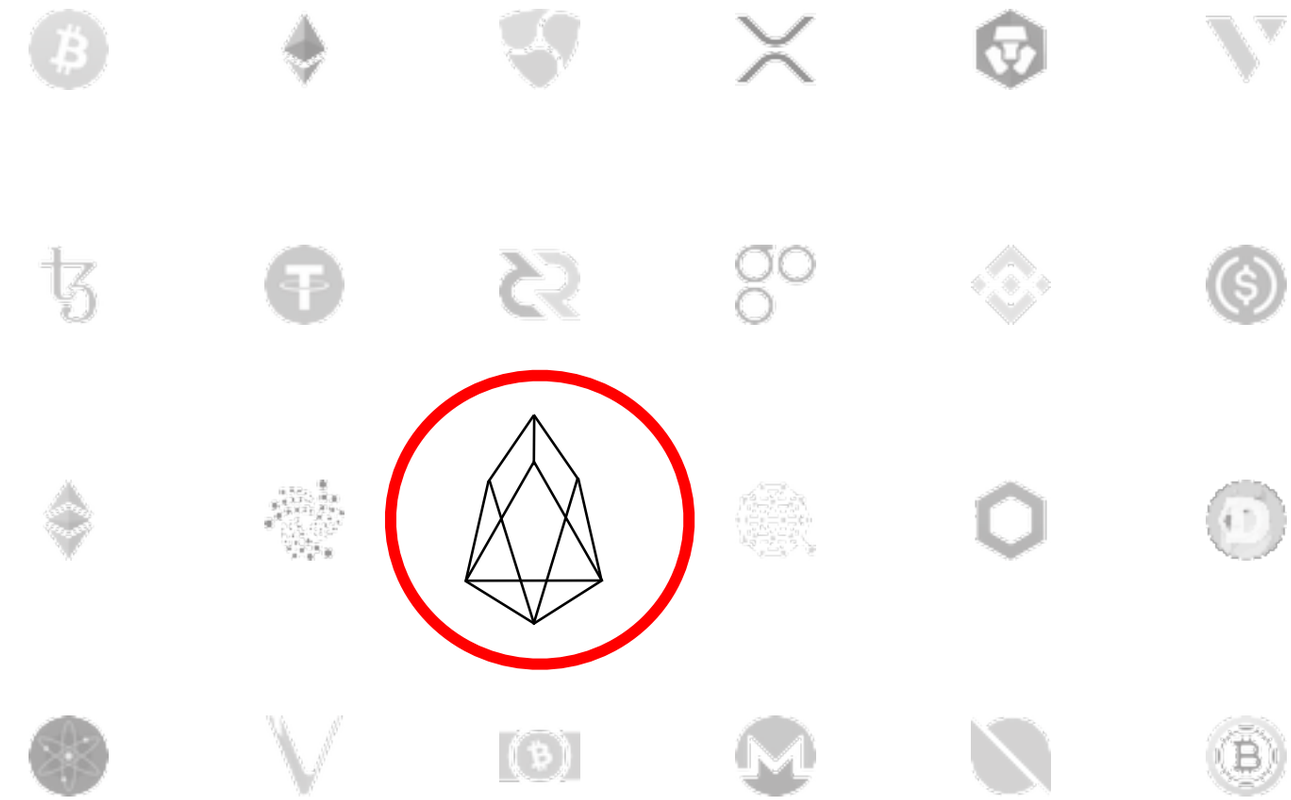
2K+ Cryptocurrencies

Abstract



Resource management of EOS.IO

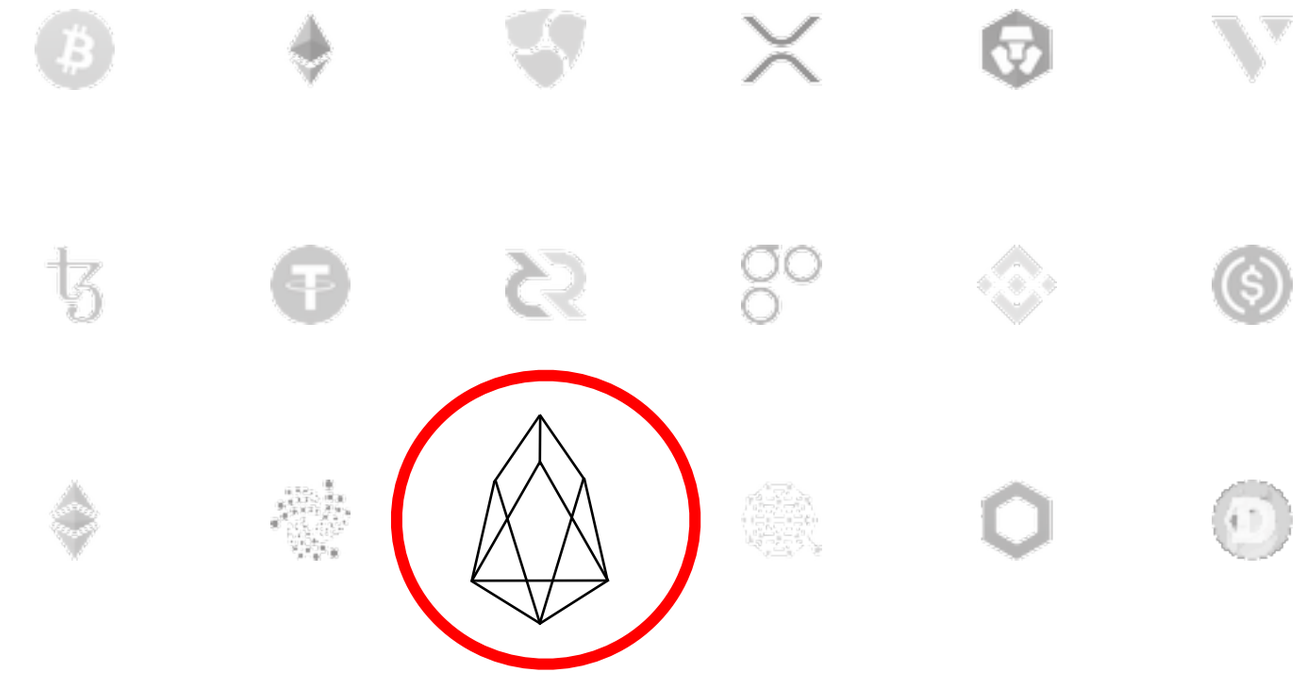
Abstract



4 unique vulnerabilities

Resource management of EOS.IO

Abstract



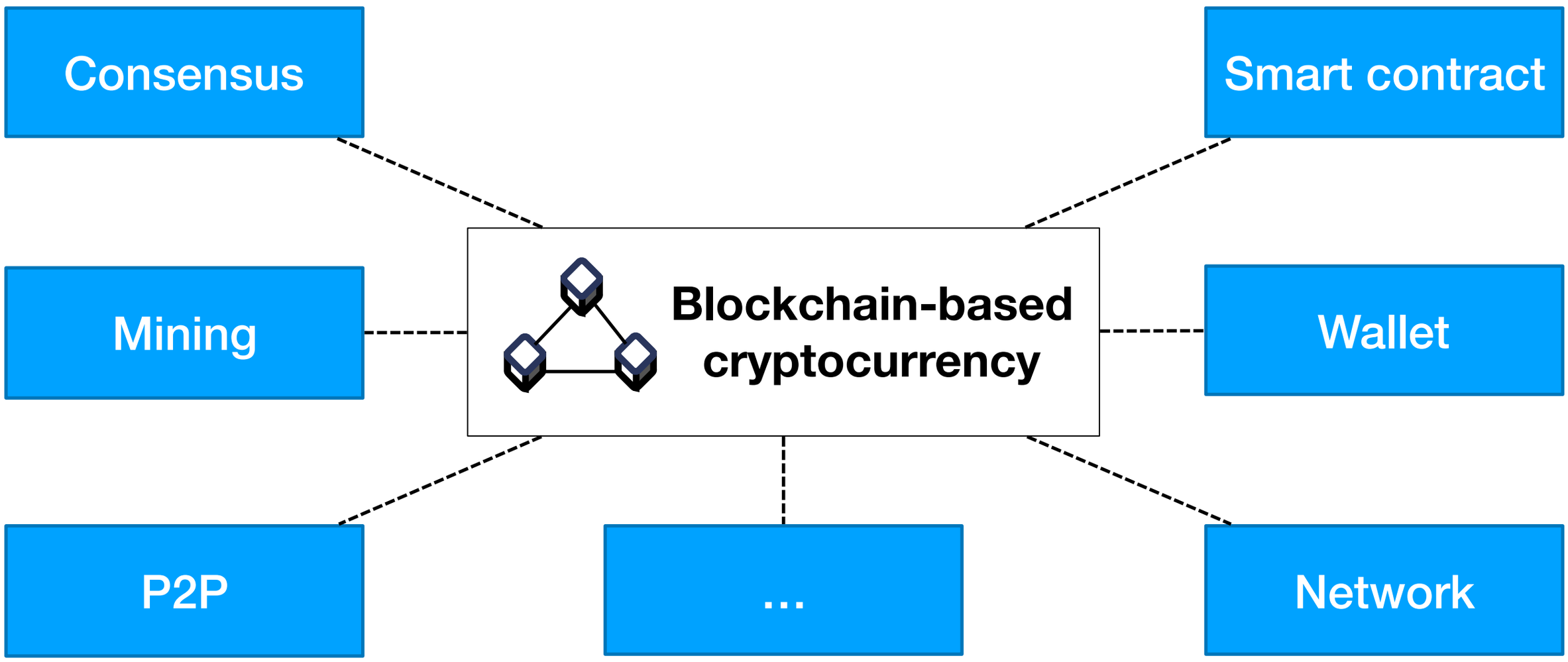
Evaluated the impact of each vulnerability

4 unique vulnerabilities

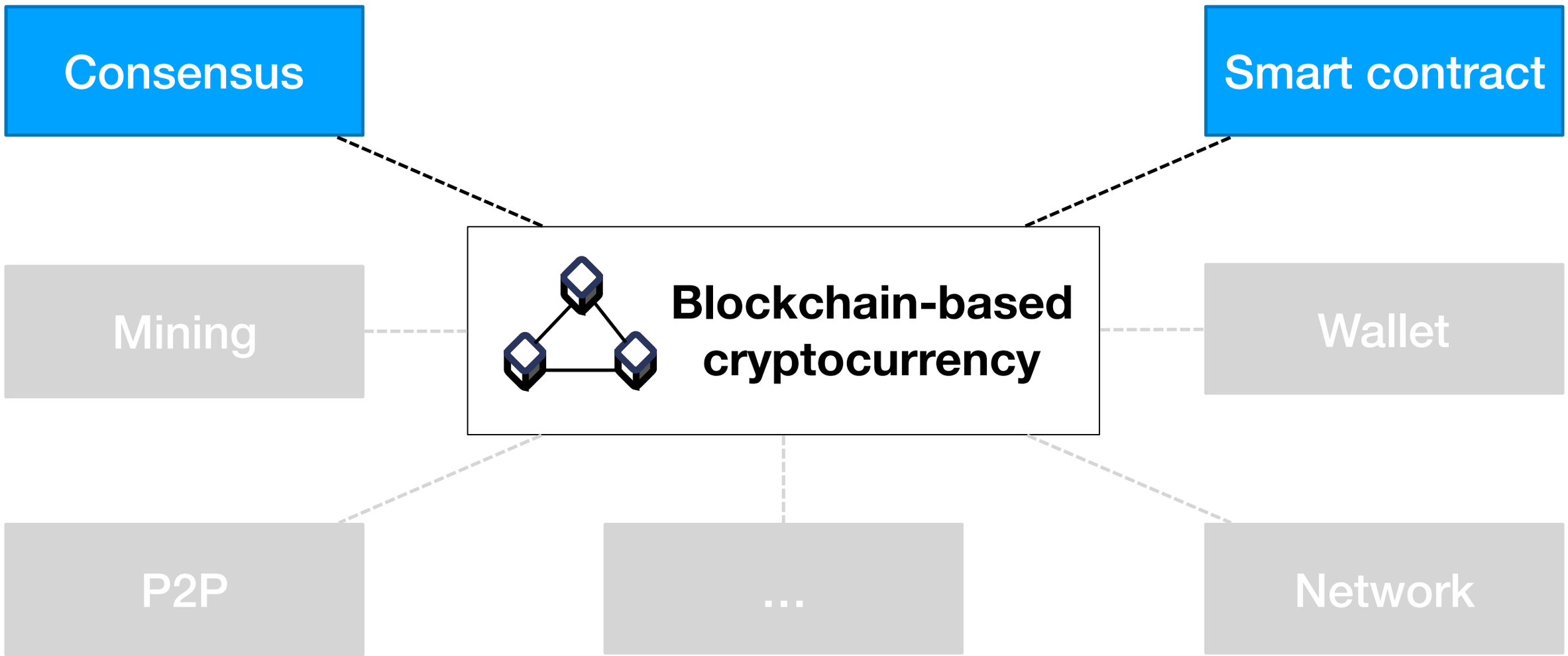
Resource management of EOS.IO

Background

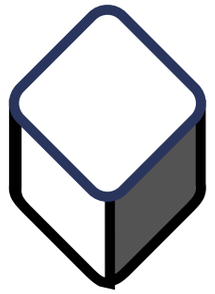
Overview of cryptocurrency components



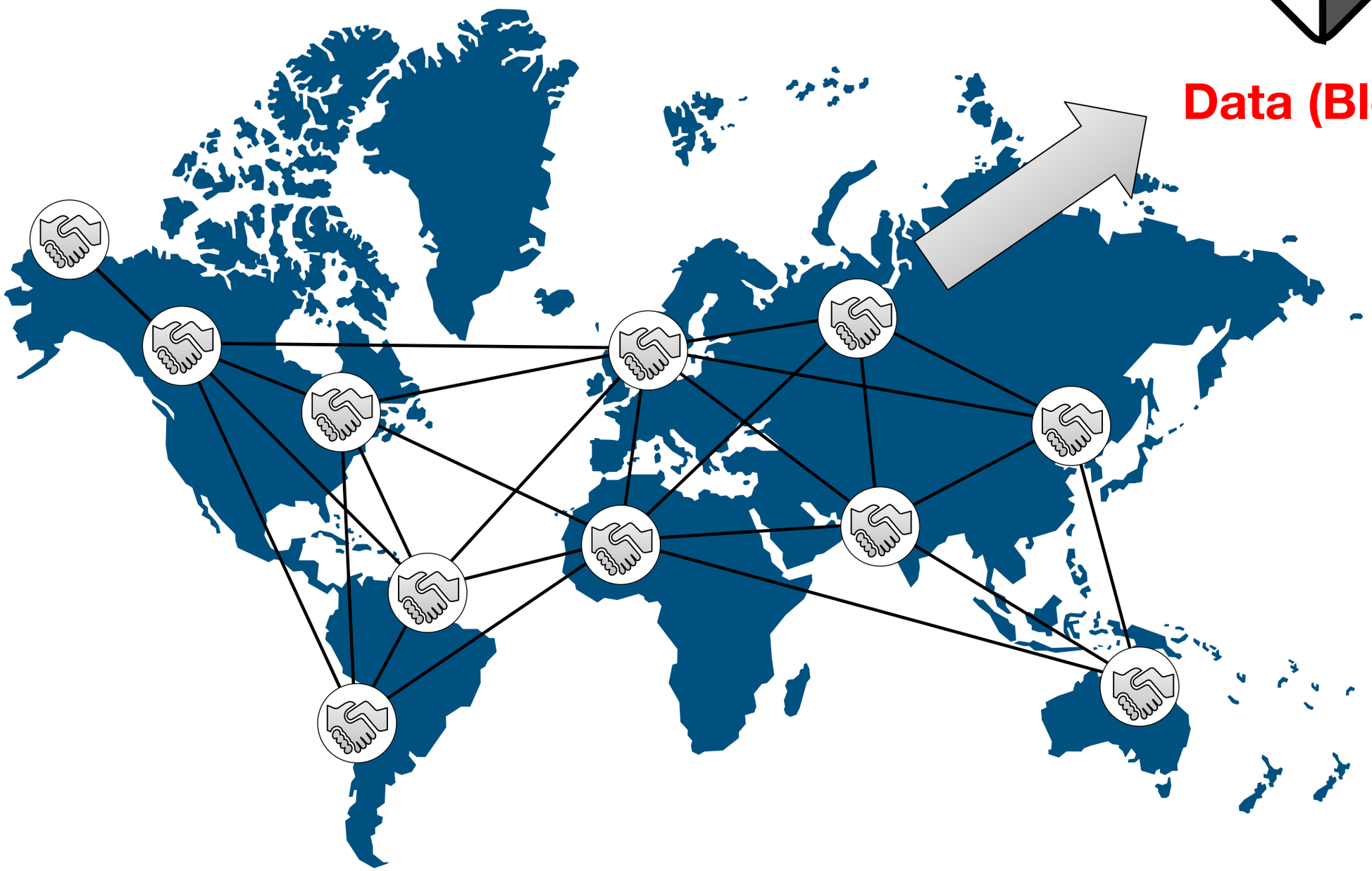
Key components



The fundamentals of blockchain

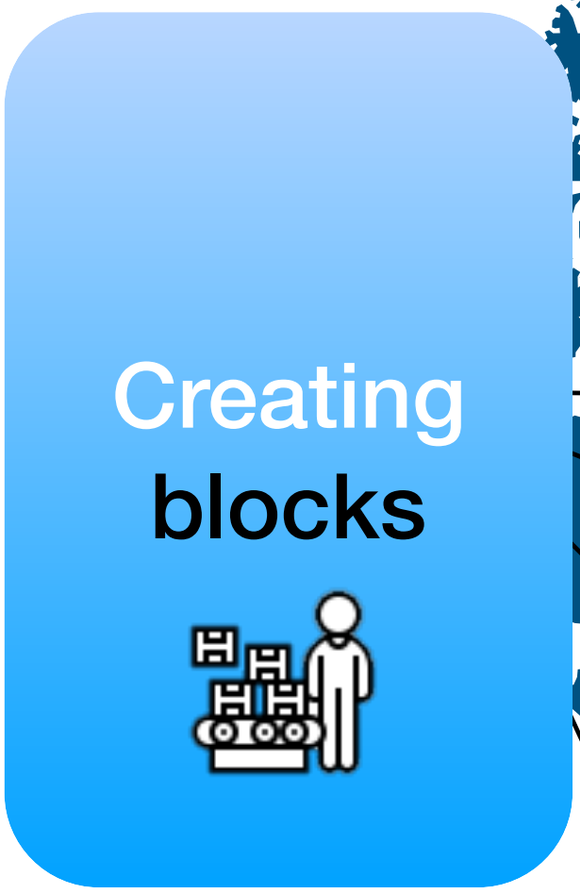


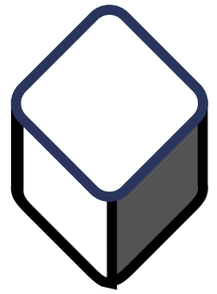
Data (Block)



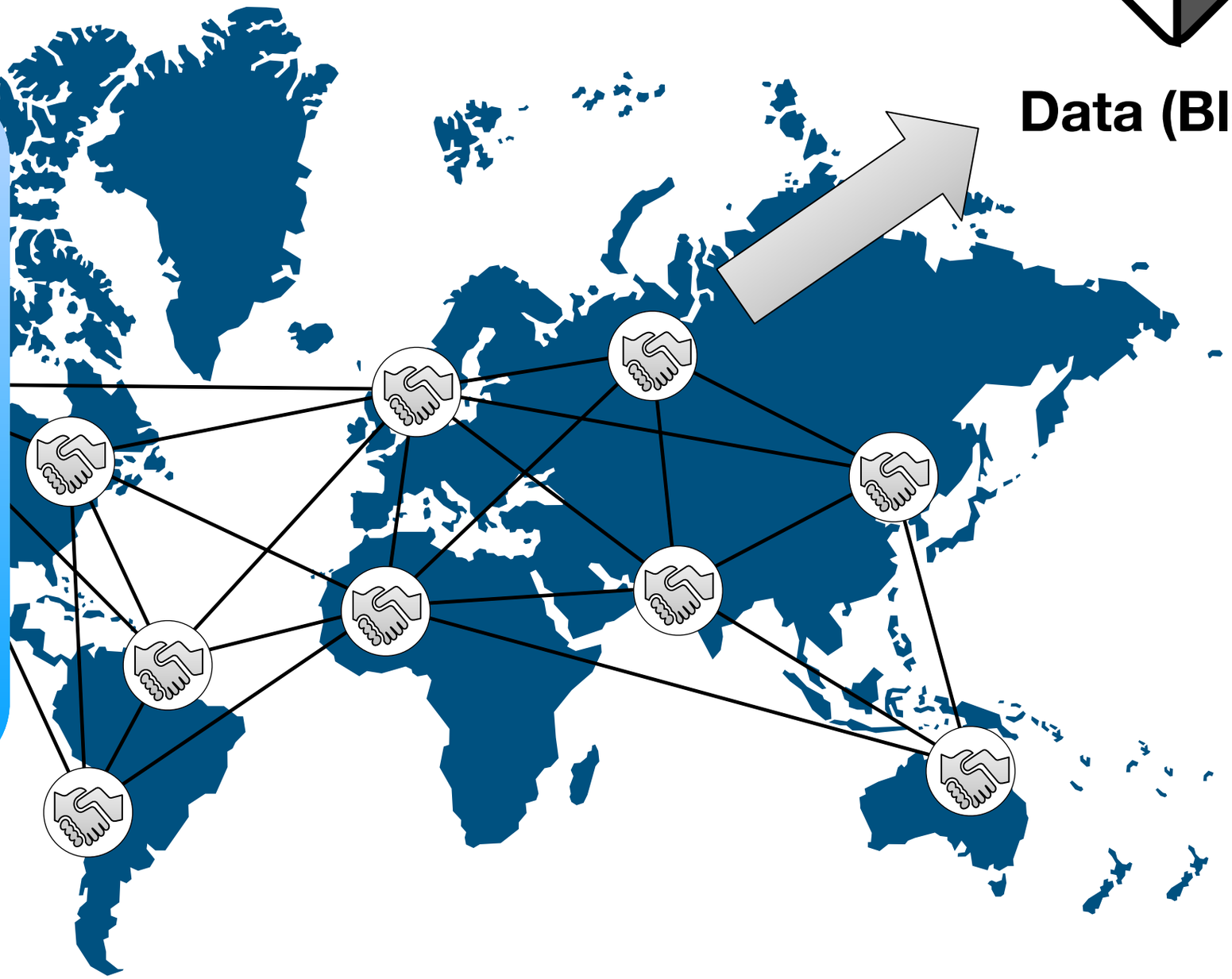
Consensus algorithm

Creating **blocks**

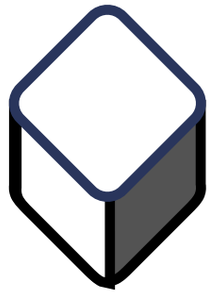




Data (Block)



Consensus algorithm

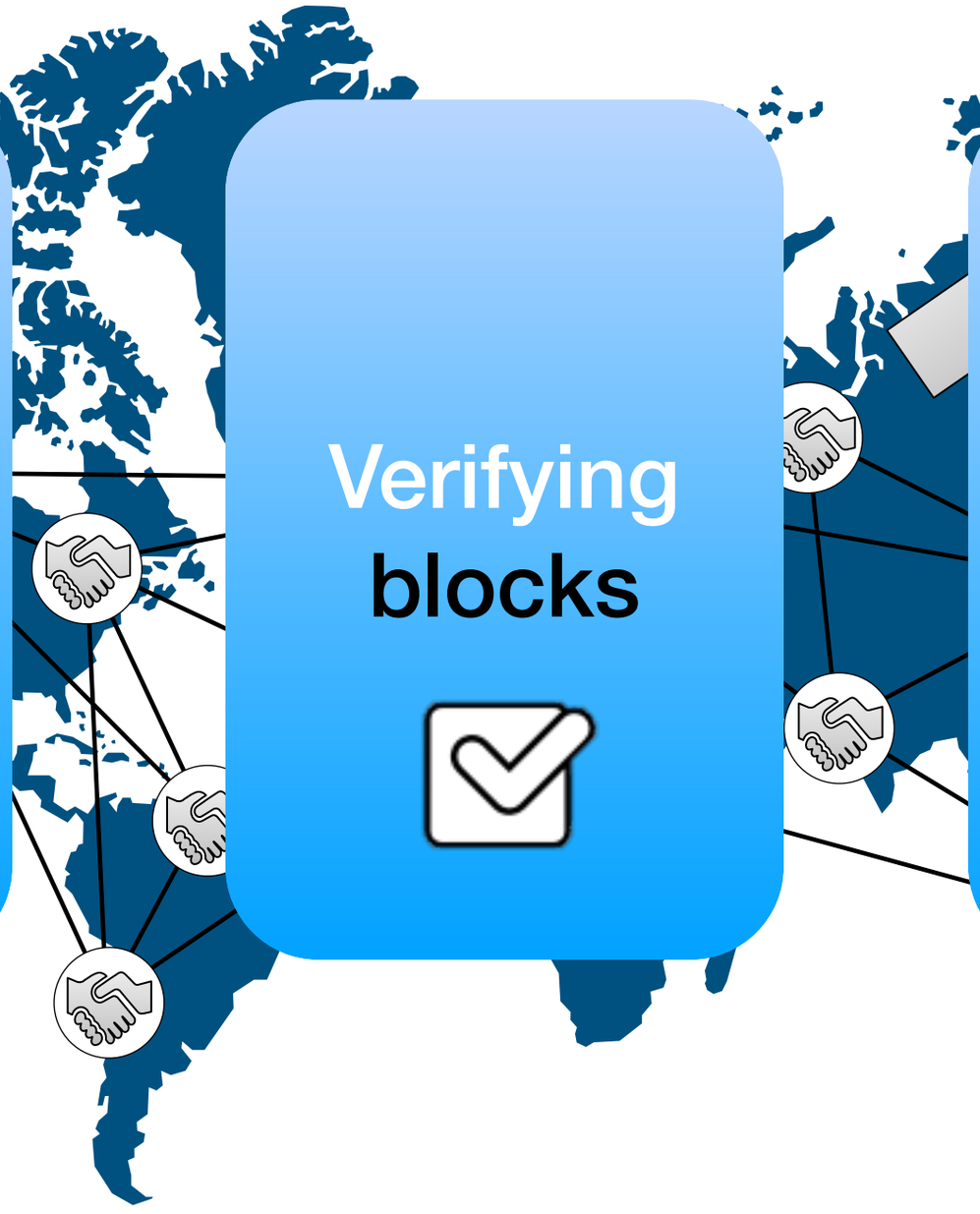


Data (Block)

Consensus algorithm

lock)

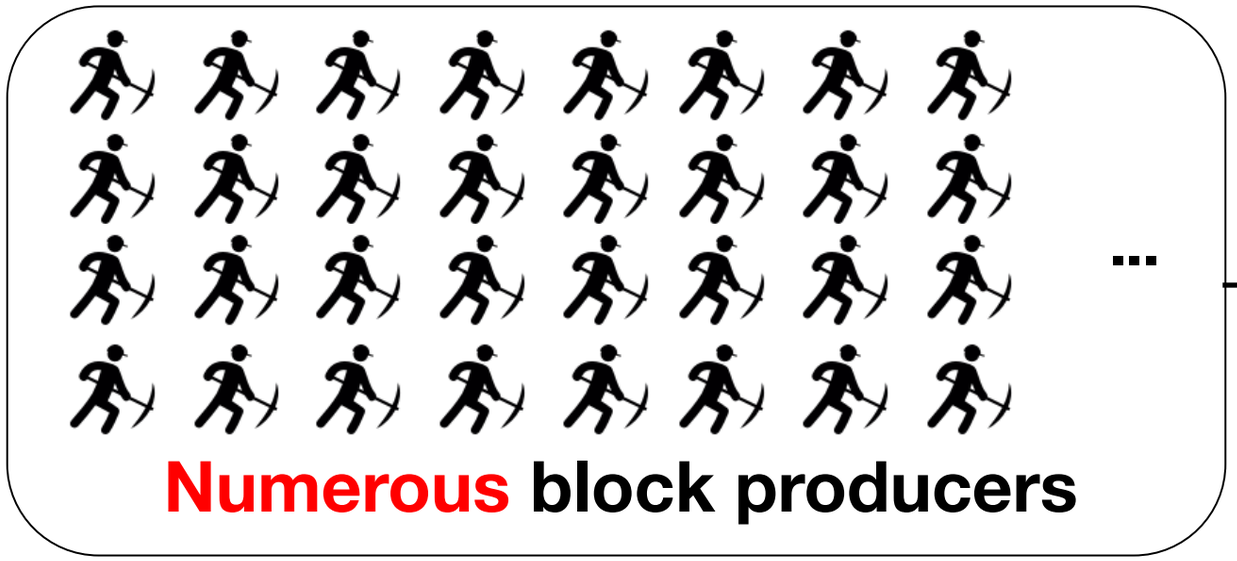
Creating blocks



Verifying blocks

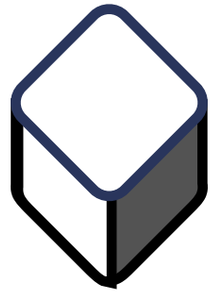
Agreement on blocks

Consensus algorithm (PoW)



Create

Agree



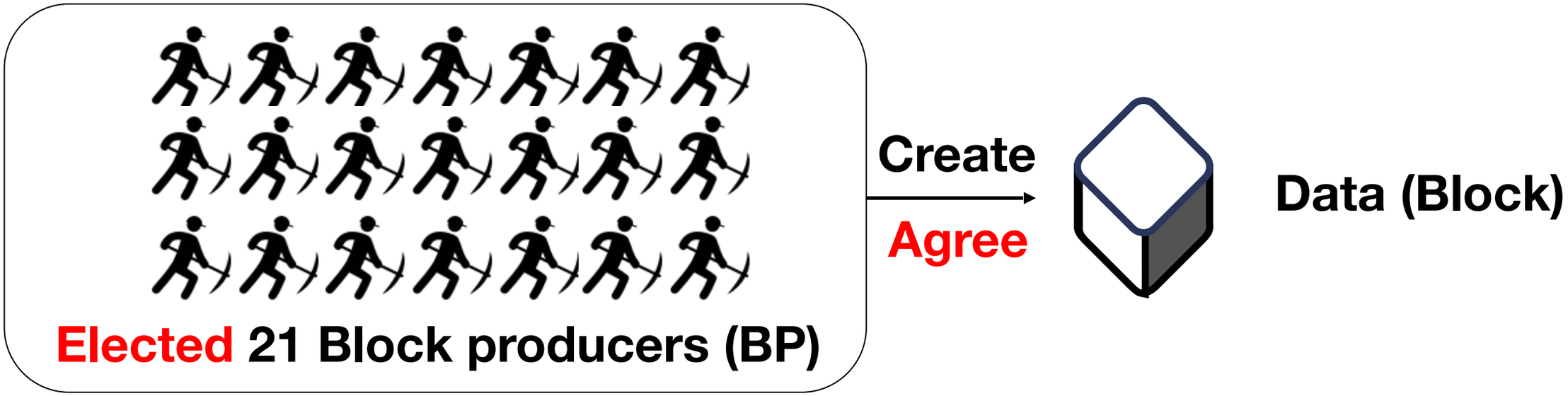
Data (Block)

Slow…

Bitcoin

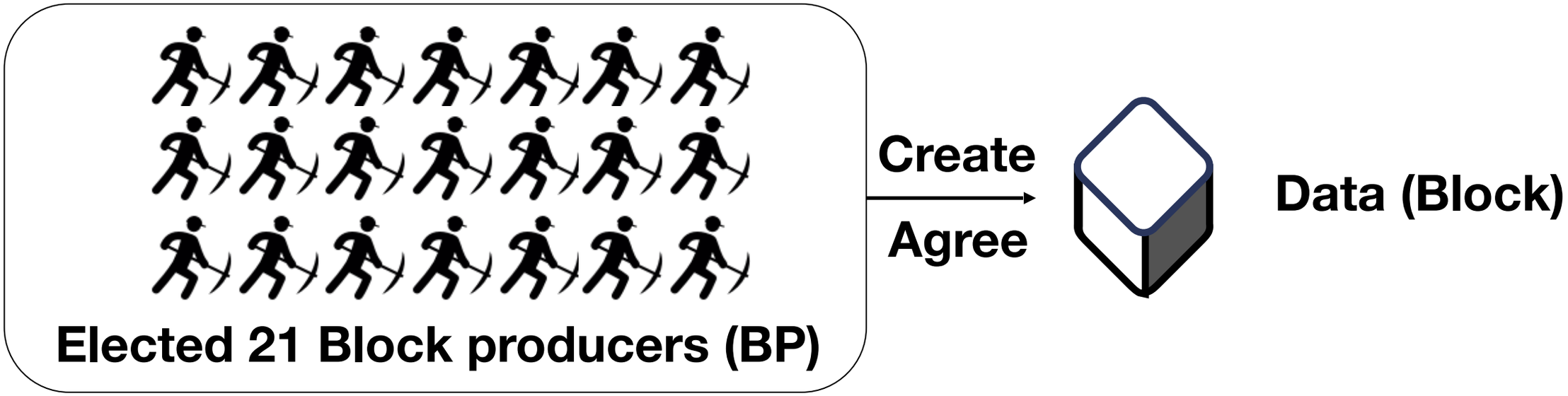
Ethereum

EOS.IO Consensus algorithm (DPoS)



FAST! (0.5 sec / block)

EOS.IO Consensus algorithm (DPoS)



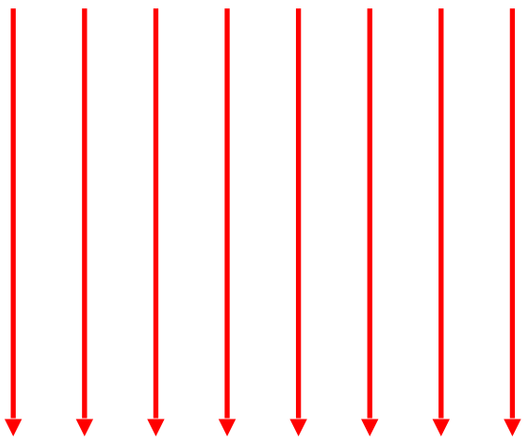
FAST! (0.5 sec / block)

But, resource management matters.

Resource management necessity

User

Transaction requests



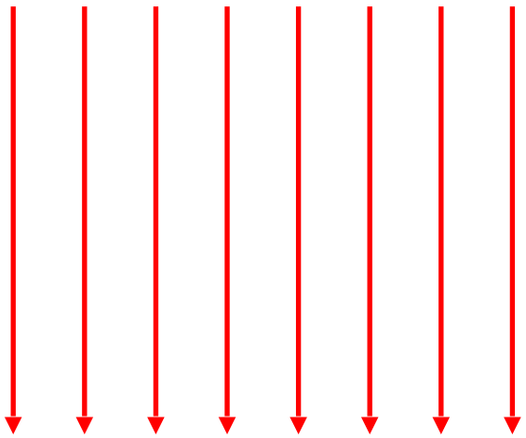
...

Blockchain

Resource management necessity

User

Transaction requests



...

Blockchain

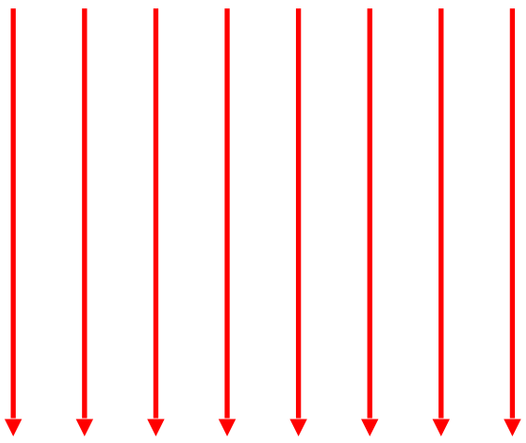
Elected 21 Block producers (BP)

Resource management necessity

...

User

Transaction requests



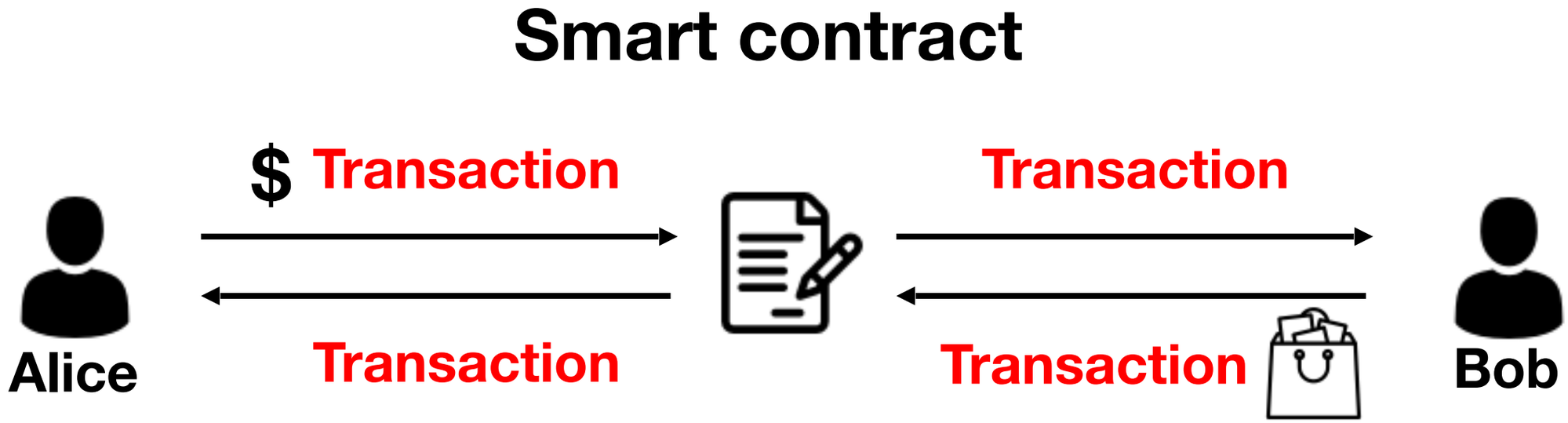
...

Overload problem

Properly process request

Blockchain

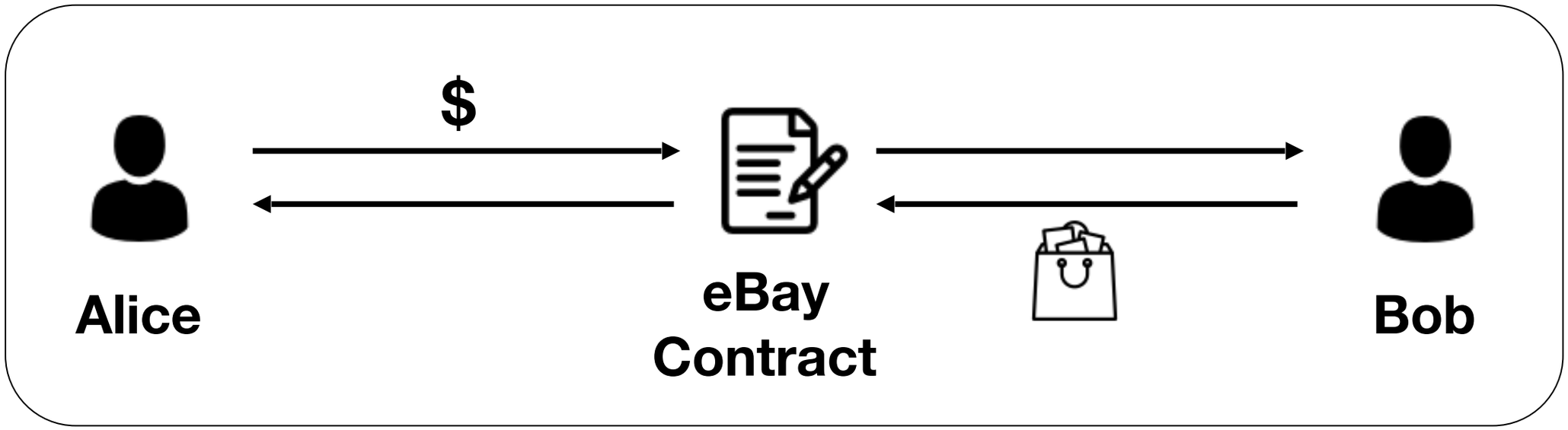
Elected 21 Block producers (BP)



Use Case

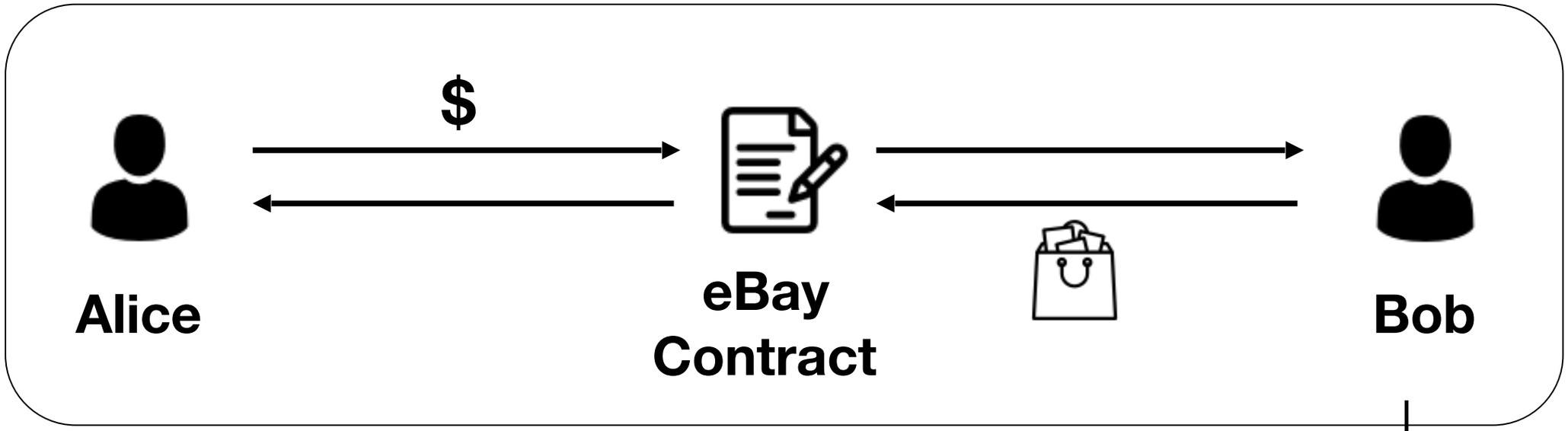
- Exchange
- Gambling
- Auction
- Funding
- Bank
- And so on.

Background: Smart contract on EOS.IO



- Target (Ex. eBay)
- Function (Ex. Bidding(), Selling())
- Permission (Ex. Alice@active)

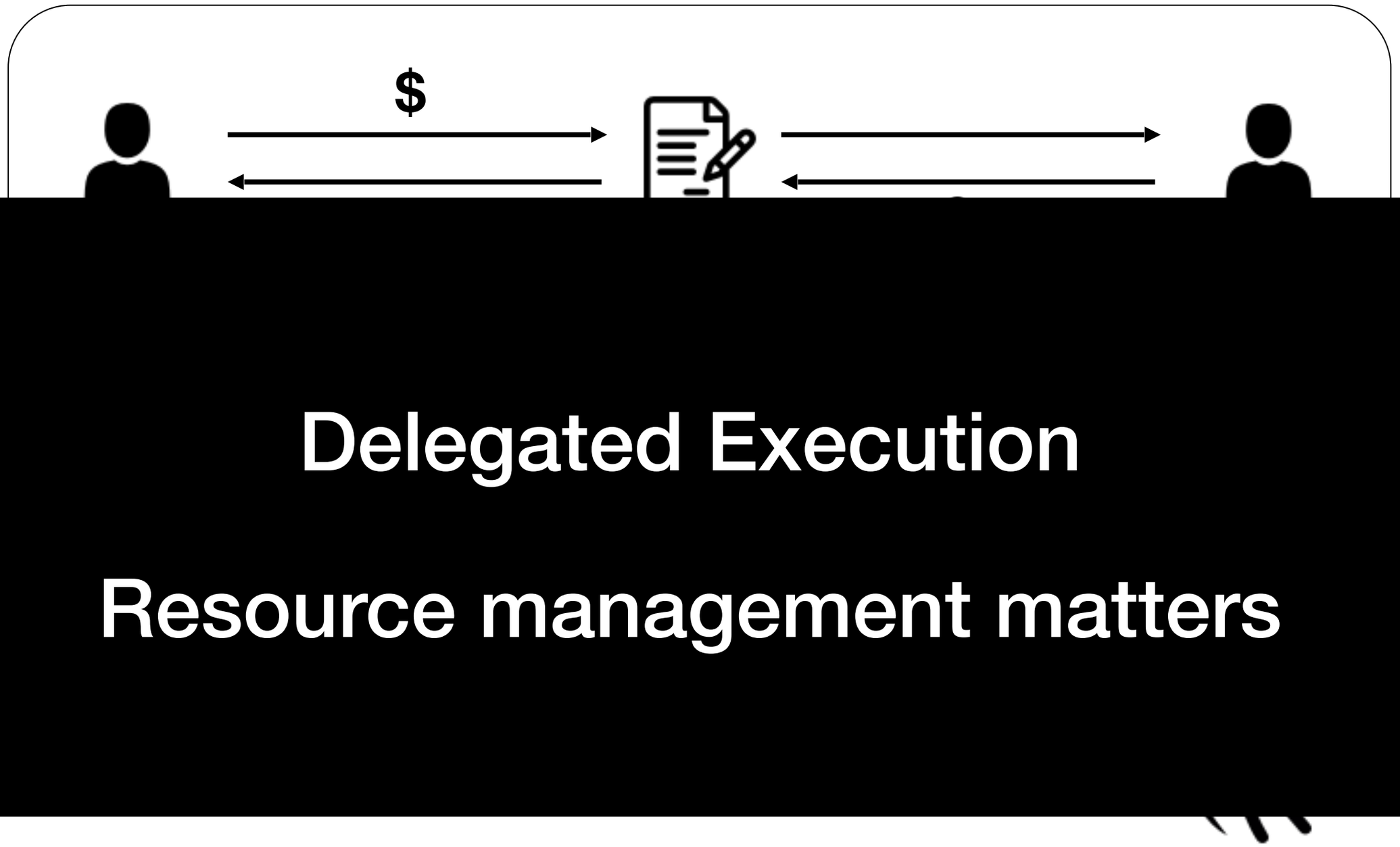
Background: Smart contract on EOS.IO



Delegated execution

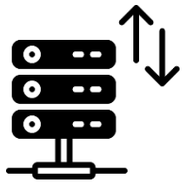
BP

Background: Smart contract on EOS.IO



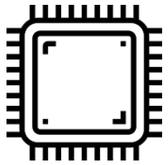
Background: Resource of EOS.IO

Transaction delivery



NET

Program execution



CPU

Data storing

RAM

Background: Resource of EOS.IO

Transaction delivery

NET

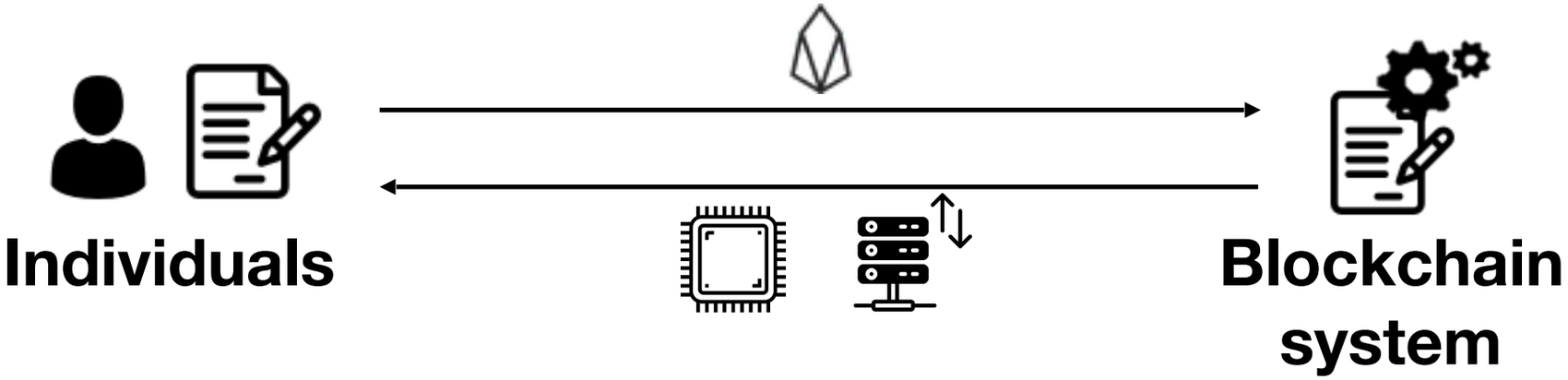
Program execution

CPU

Data storing

RAM

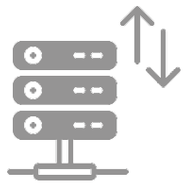
Staking



Refreshed every day

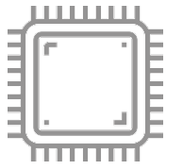
Background: Resource of EOS.IO

Transaction delivery



NET

Program execution

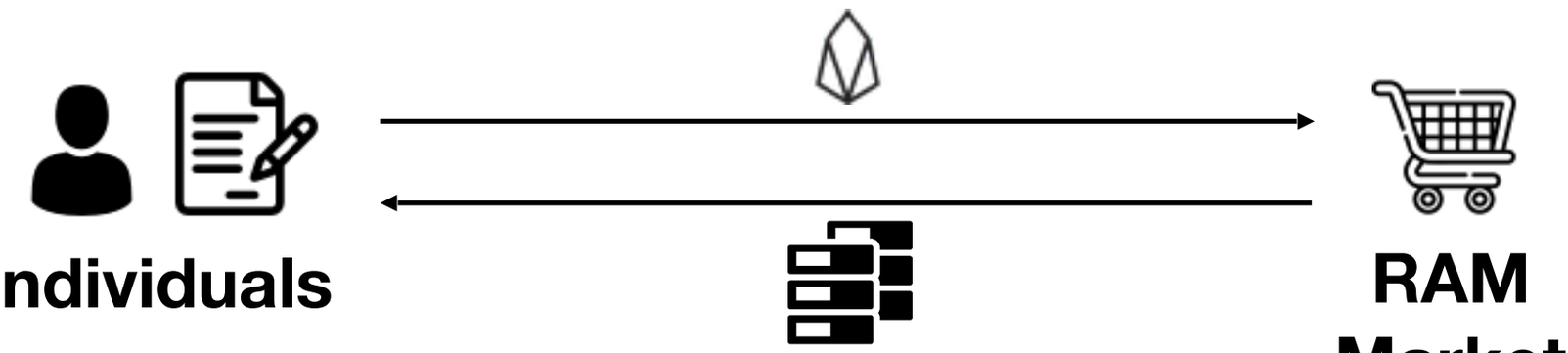


CPU

Data storing

RAM

Buy



Not refreshed every day.

Why EOS?

Why EOS?

Market cap

#1 #2

Smart contract research

Making smart contracts smarter
(ACM CCS '16)

ZEUS: Analyzing Safety of Smart Contracts
(NDSS '18)

teether: Gnawing at ethereum to automatically exploit smart contracts
(USENIX '18)

Consensus research

The miner's dilemma
(IEEE S&P '15)

Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin
(ACM CCS '17)

Publish or perish: A backward-compatible defense against selfish mining in bitcoin
(RSA '17)

Other research work

Porosity: A decompiler for blockchain-based smart contracts bytecode
(Defcon '17)

Hijacking bitcoin: Routing attacks on cryptocurrencies
(IEEE S&P '17)

Eclipse attacks on bitcoin's peer-to-peer network
(USENIX '15)

Why EOS?

Rank of marketcap	Name	Consensus algorithm	Smart contract platform
1	Bitcoin	PoW	X
2	Ethereum	PoW	O
3	Ripple	PoS	X
4	Litecoin	PoW	X
5	Bitcoin cash	PoW	X
6	Binance Coin	X	X
7	EOS	DPoS	O

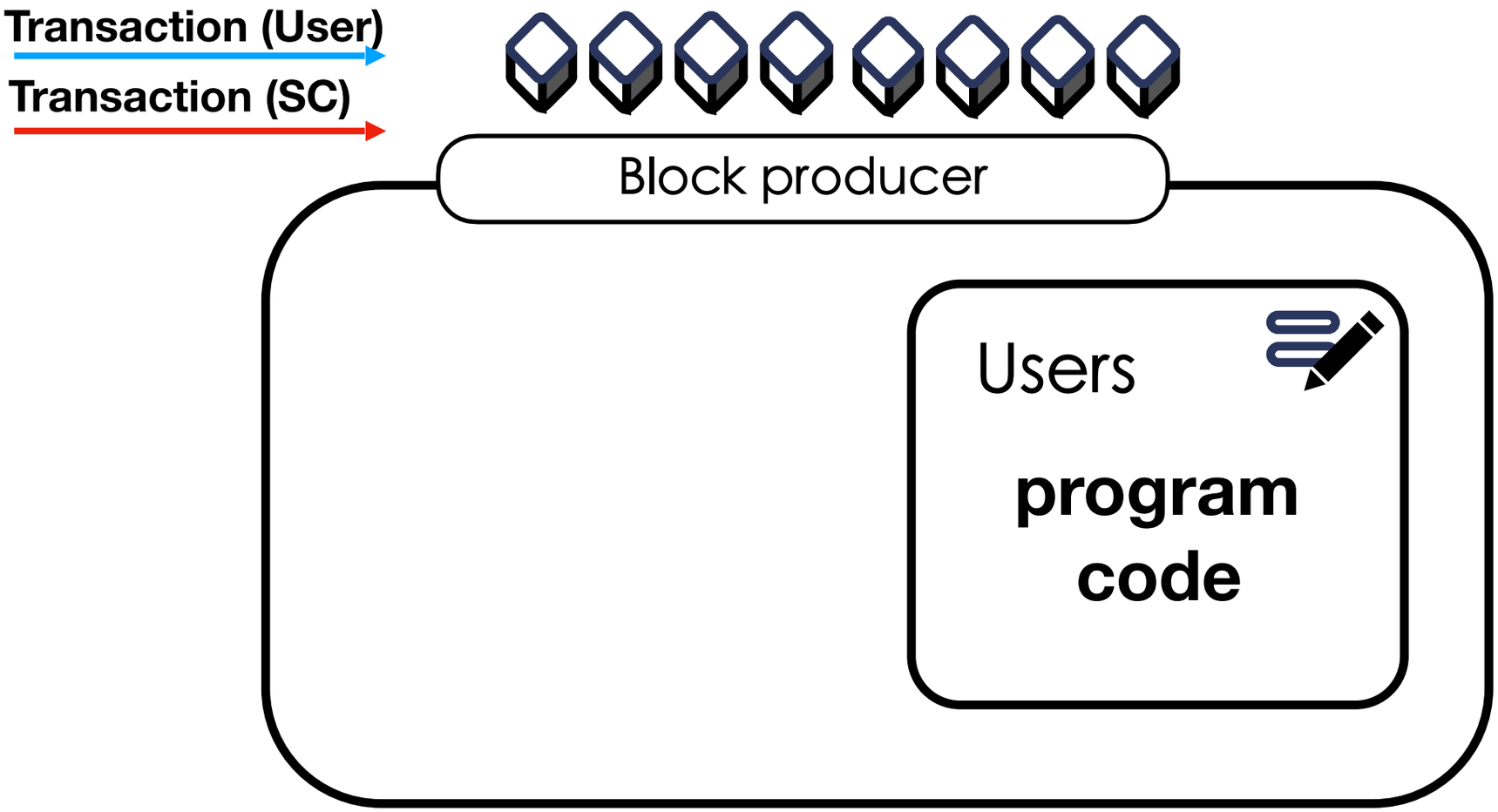
User accounts ≈ 1.3 M

But, no security research in academia.

In our paper…

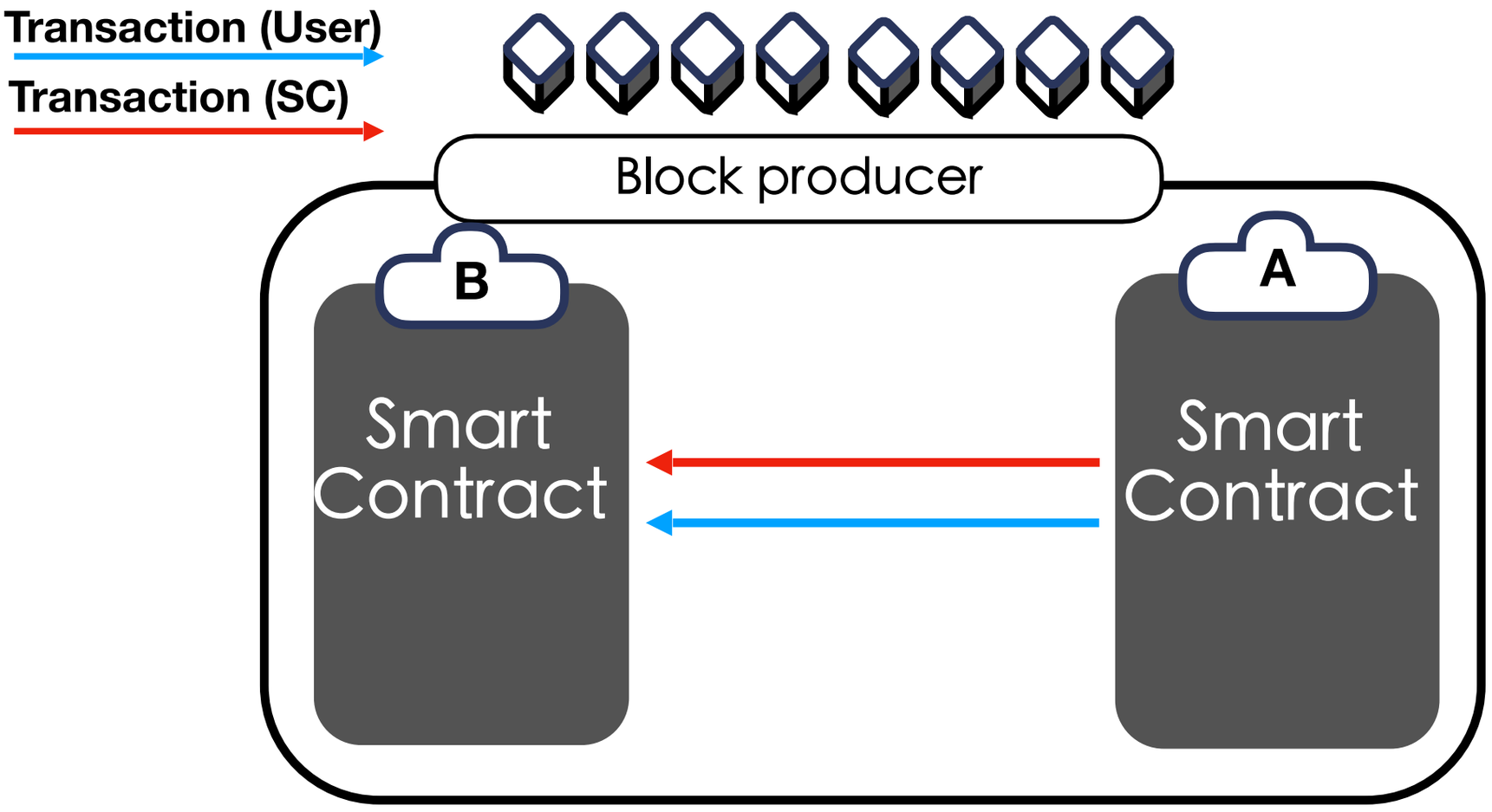
EOS structure

What are new attack targets?



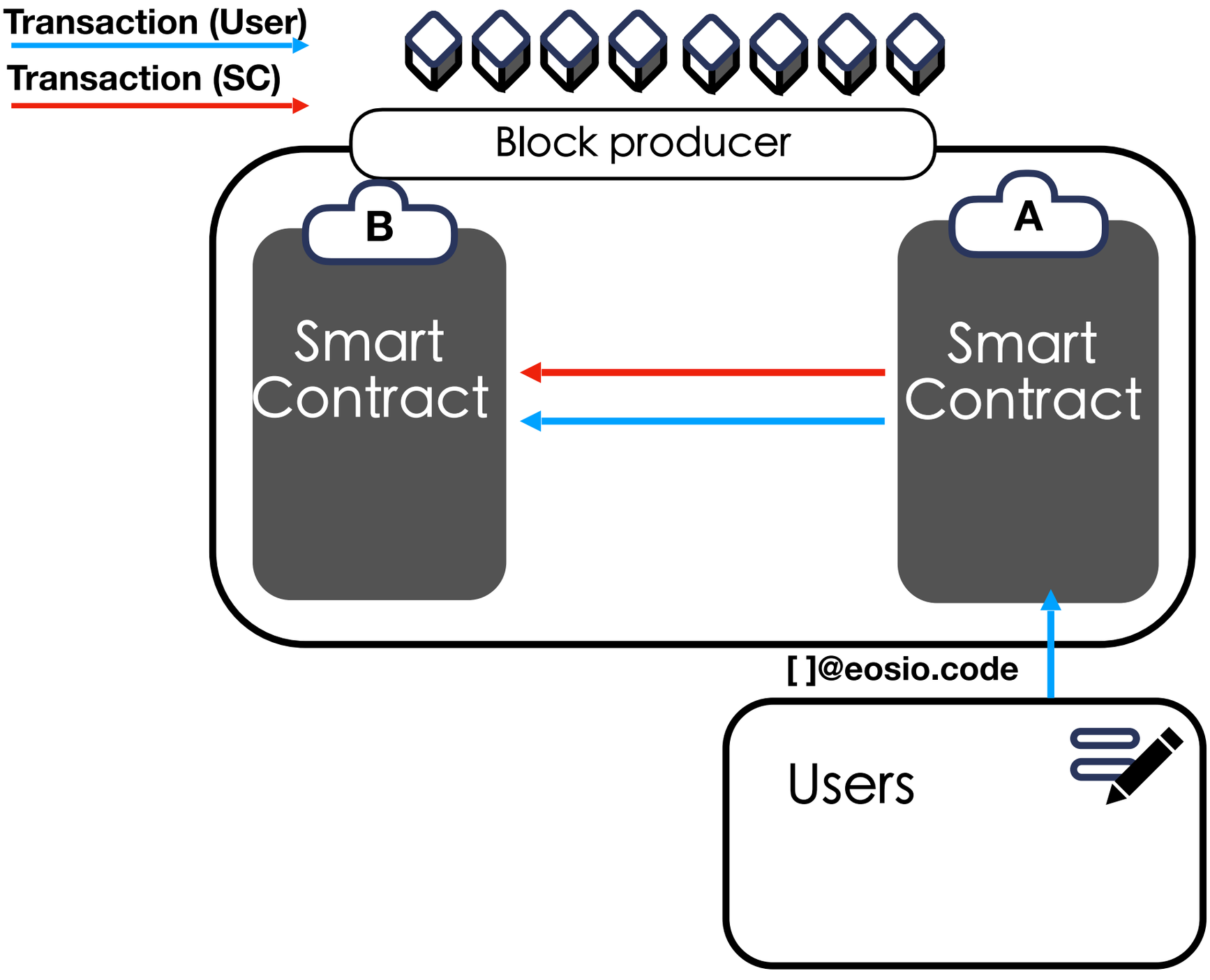
EOS structure

What are new attack targets?



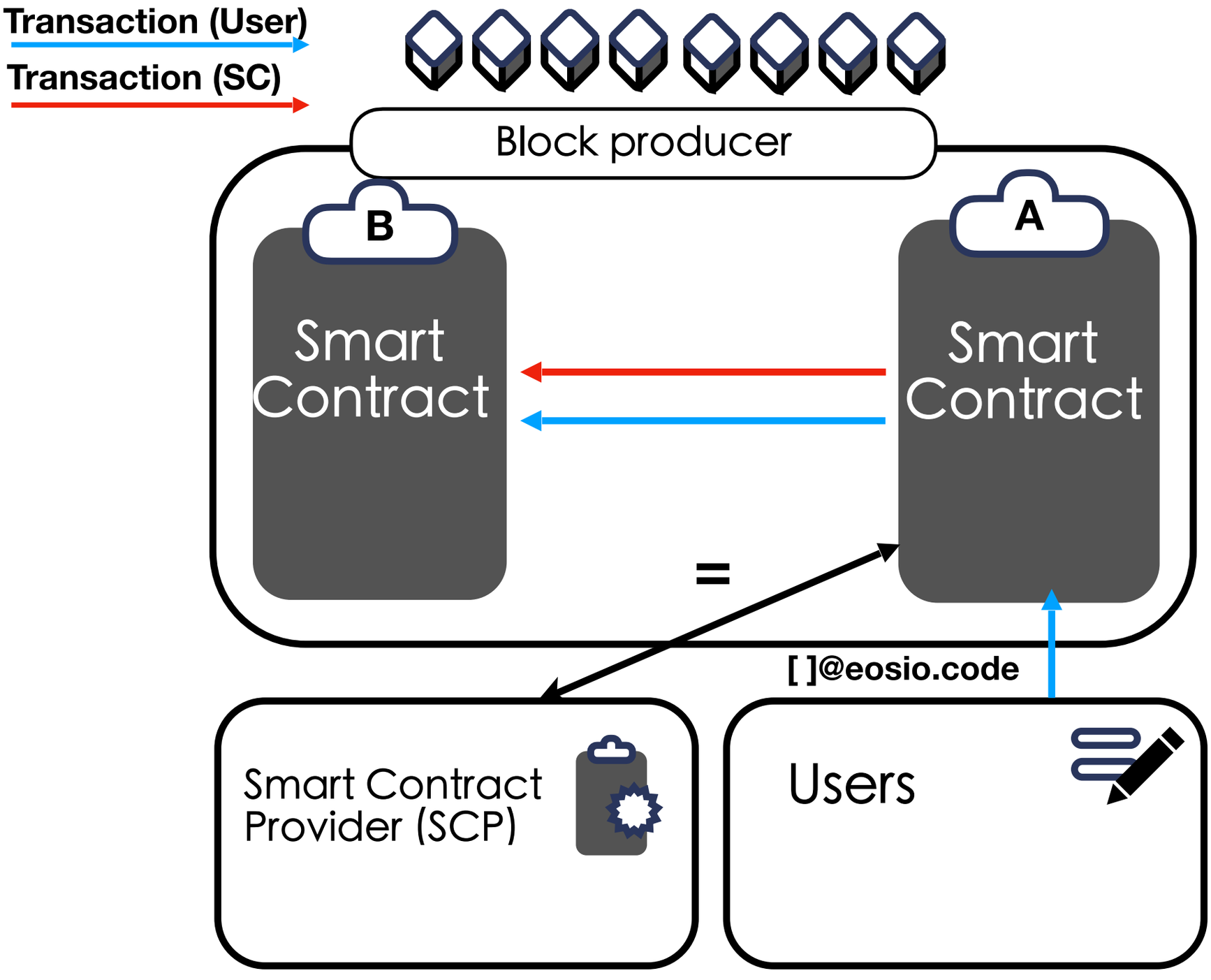
EOS structure

What are new attack targets?



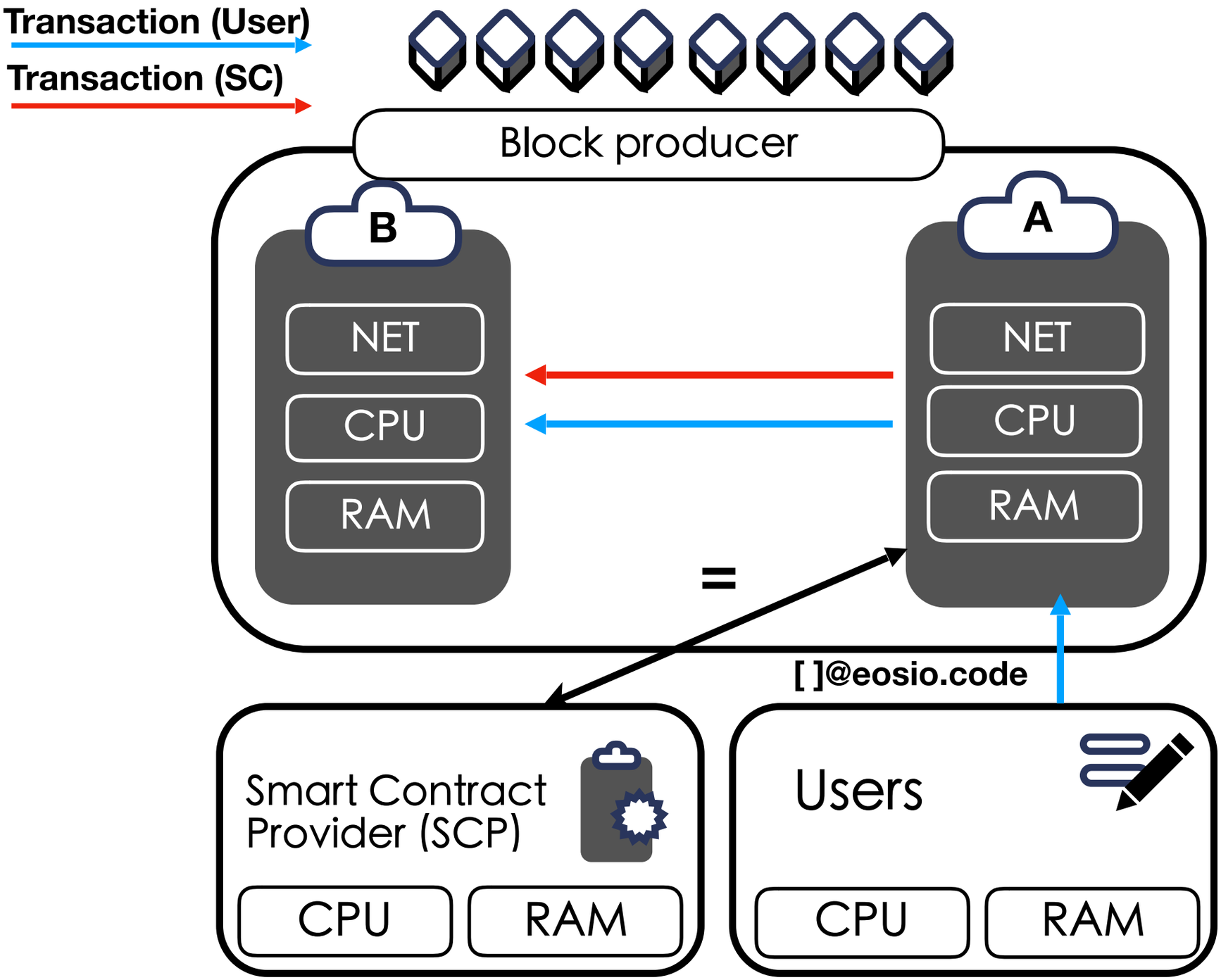
EOS structure

What are new attack targets?



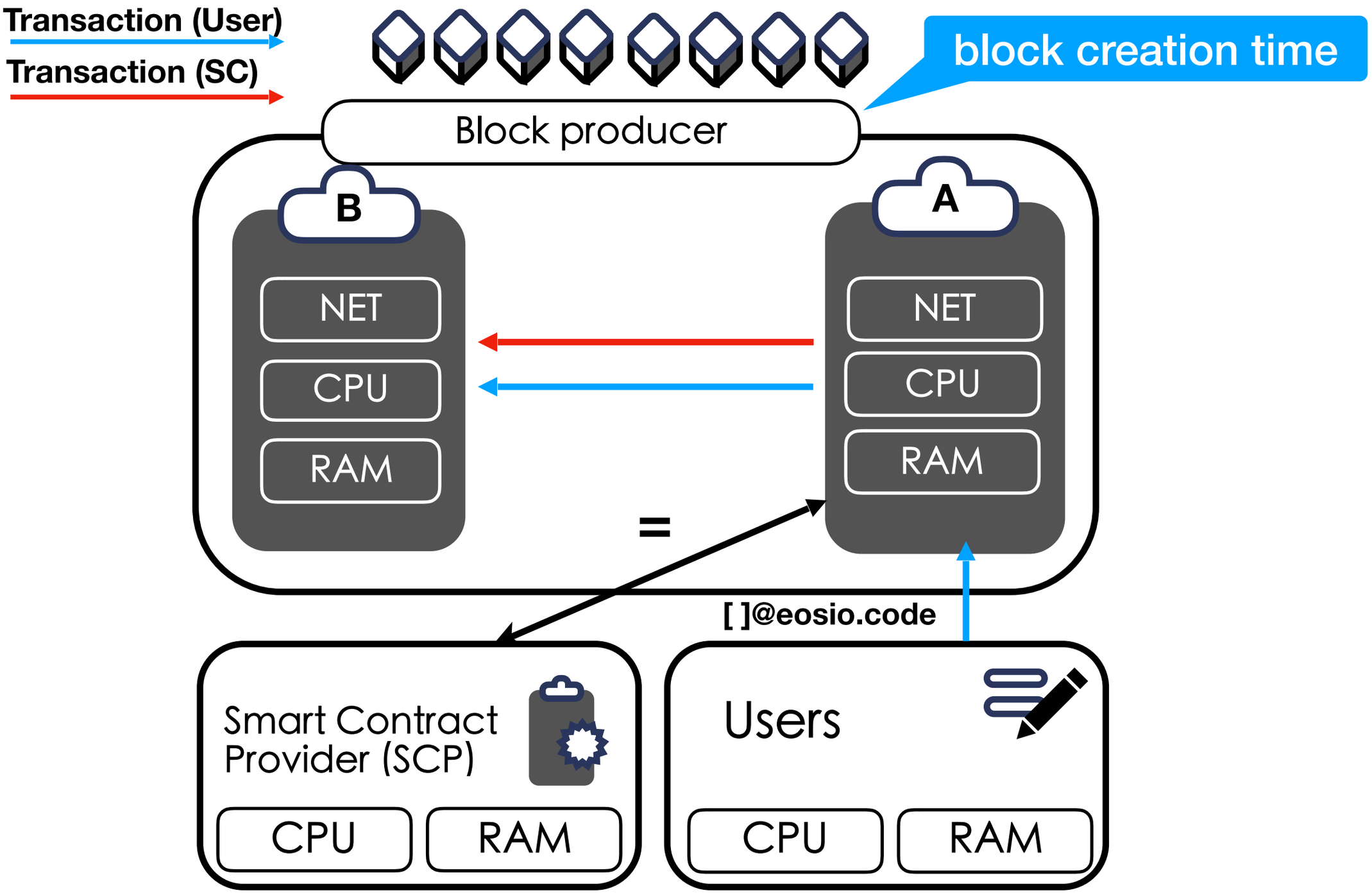
EOS structure

What are new attack targets?



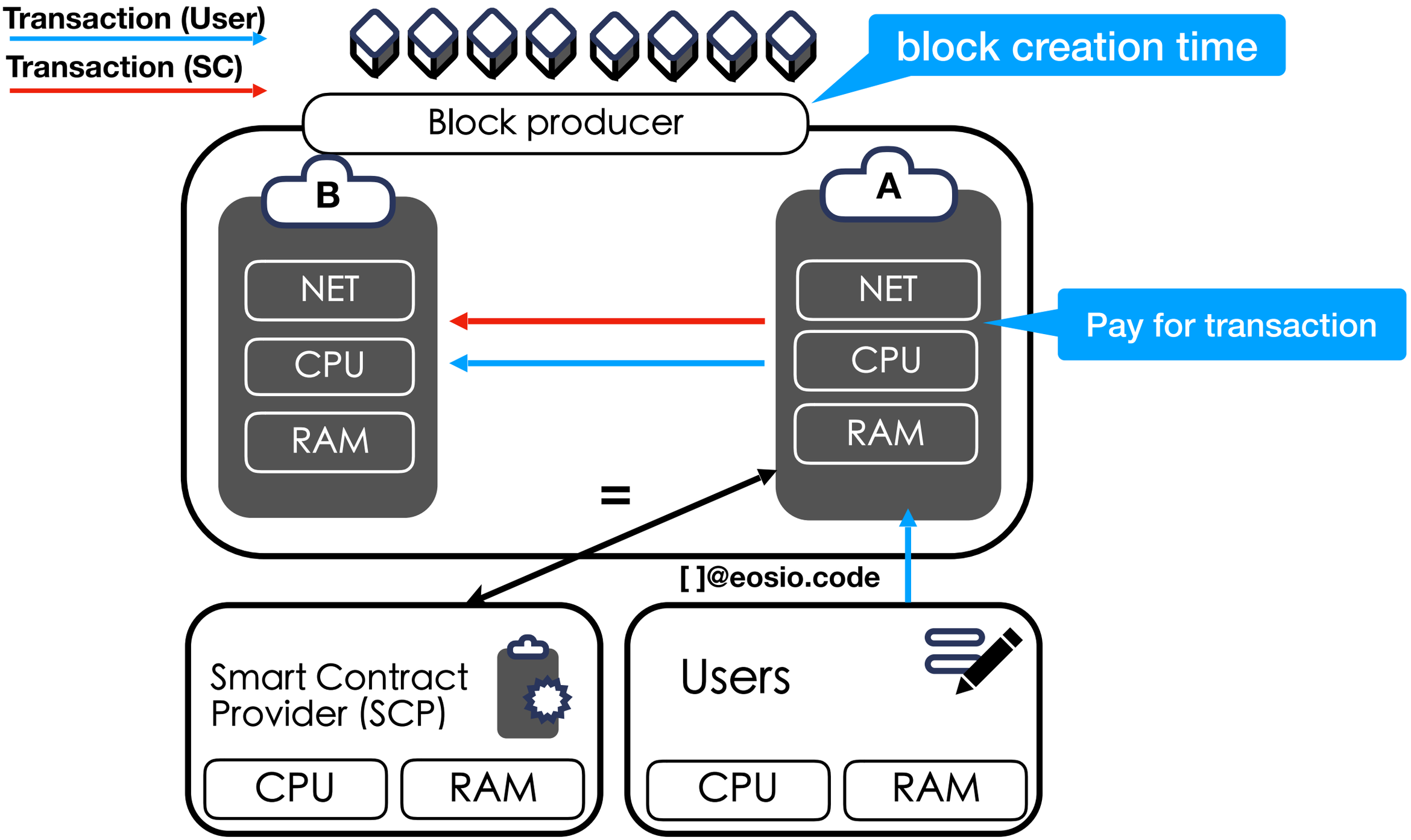
Attack Target

What are new attack targets?



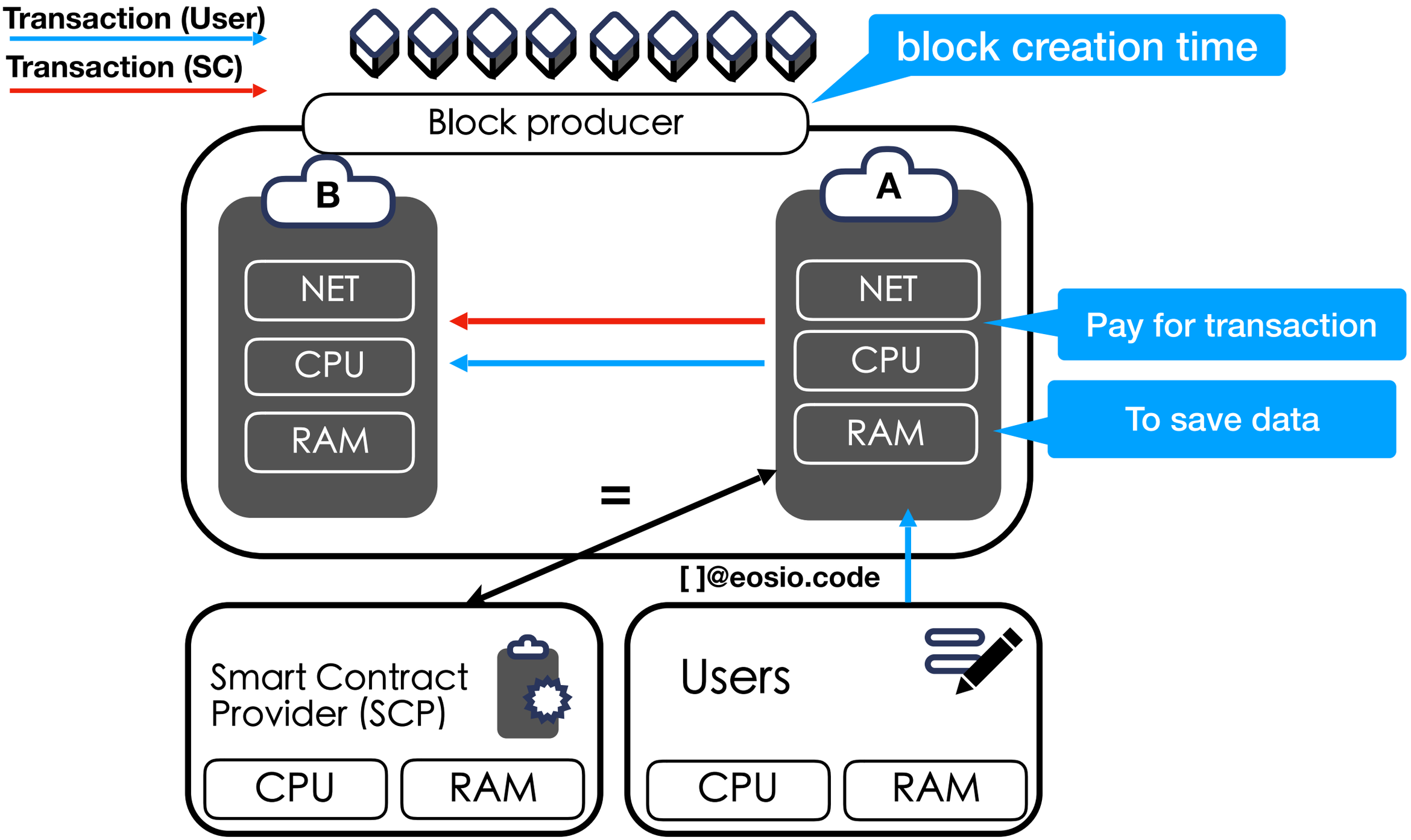
Attack Target

What are new attack targets?



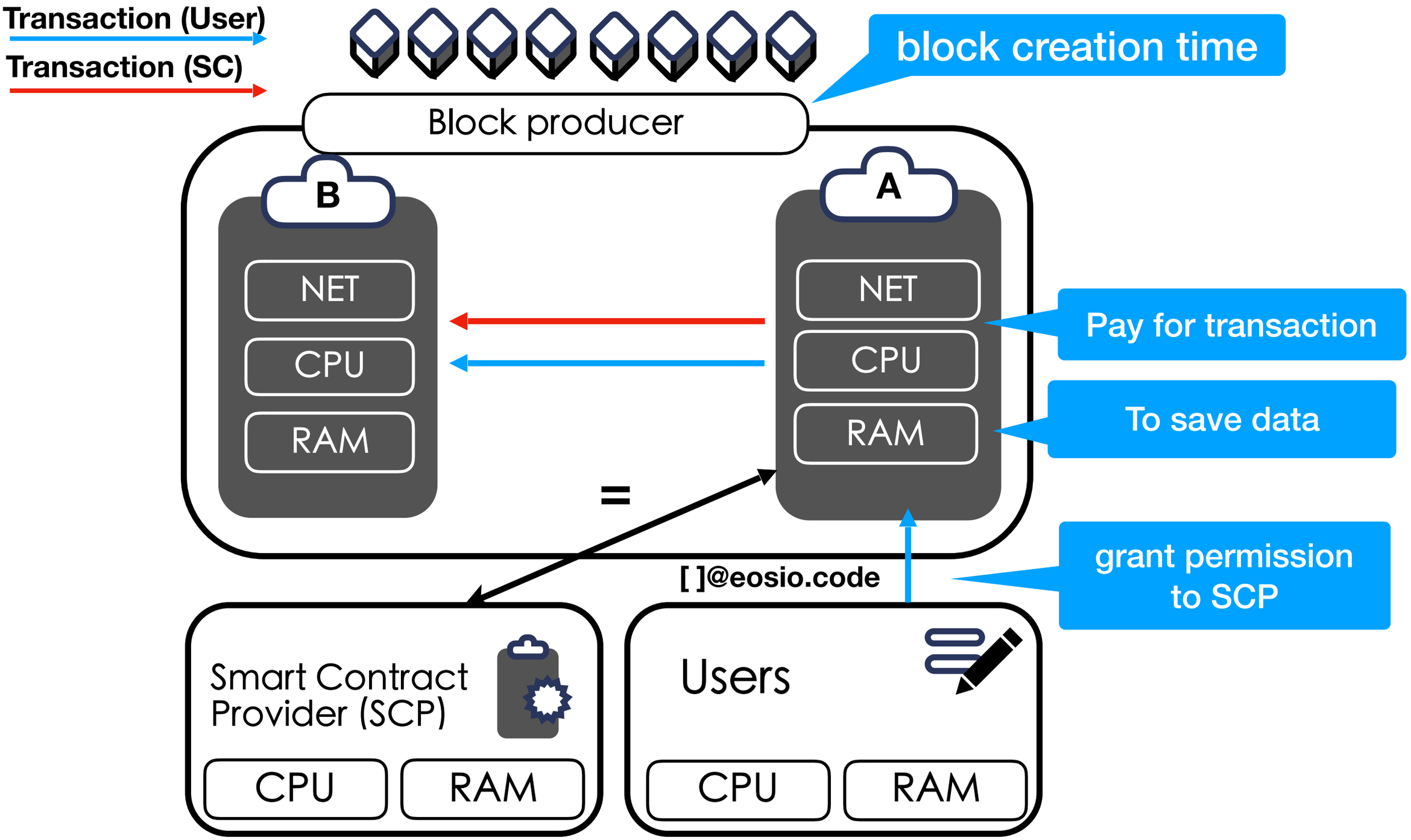
Attack Target

What are new attack targets?



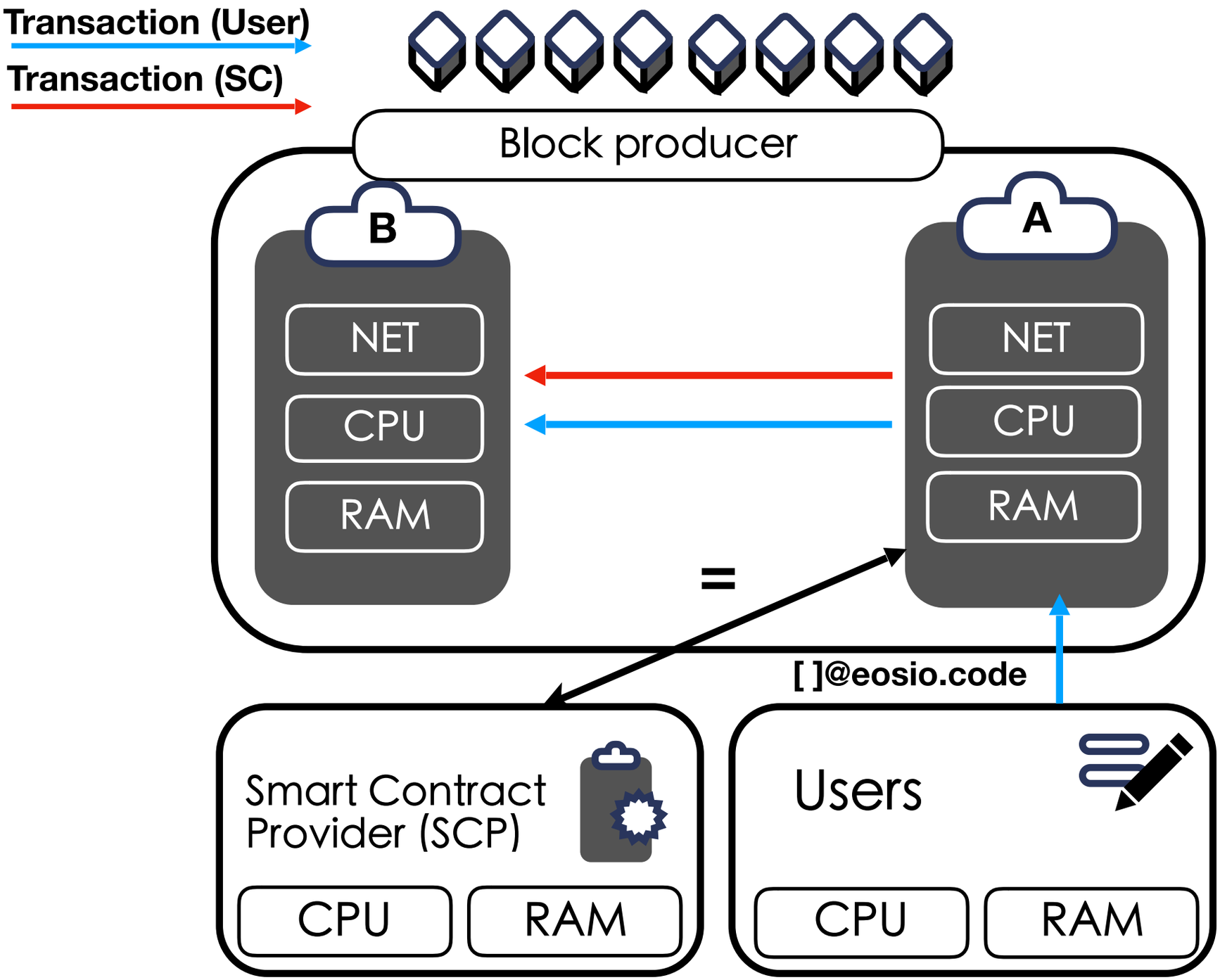
Attack Target

What are new attack targets?



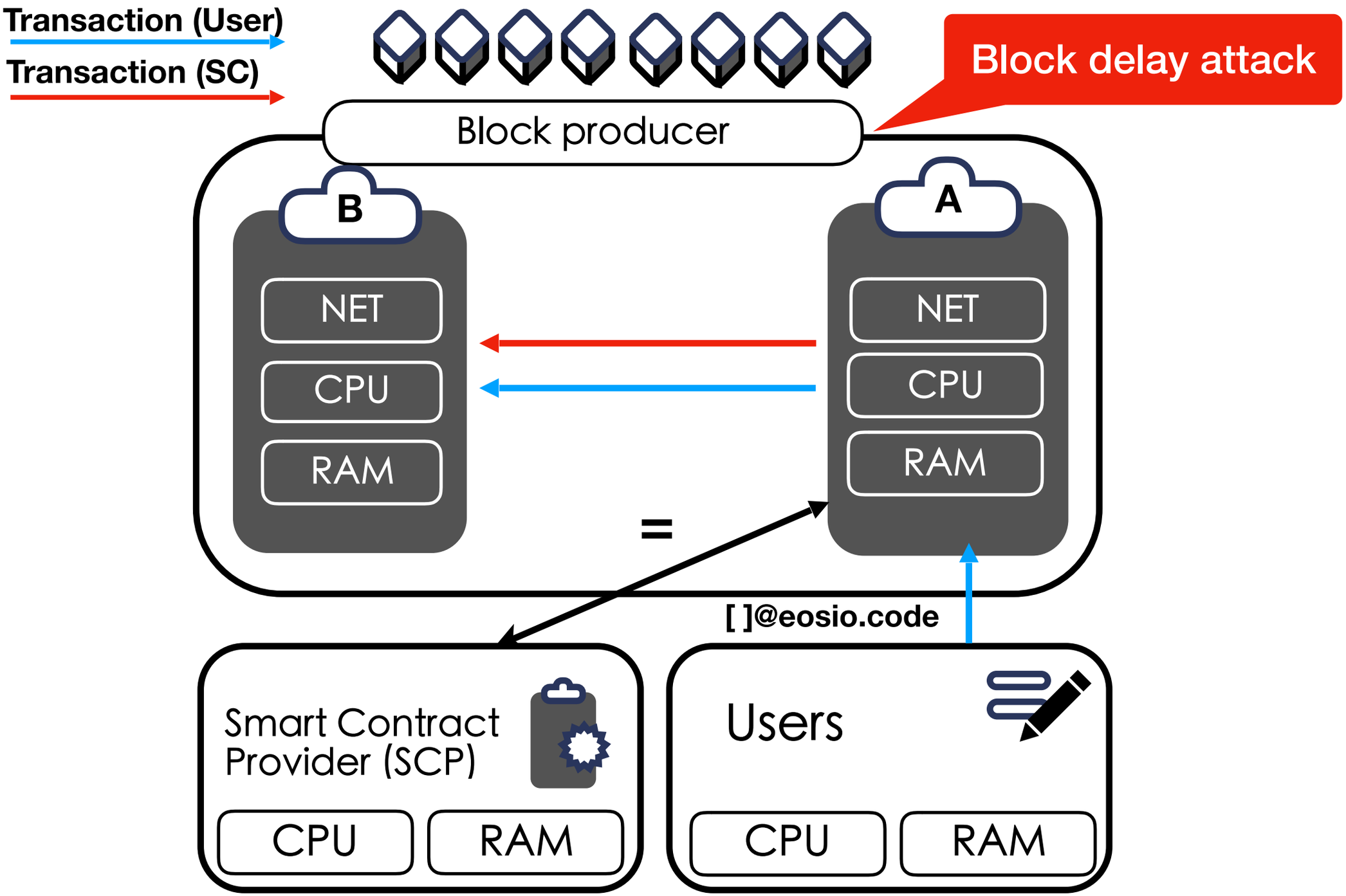
Attack Models & Threat Models & Attacks!

We found …



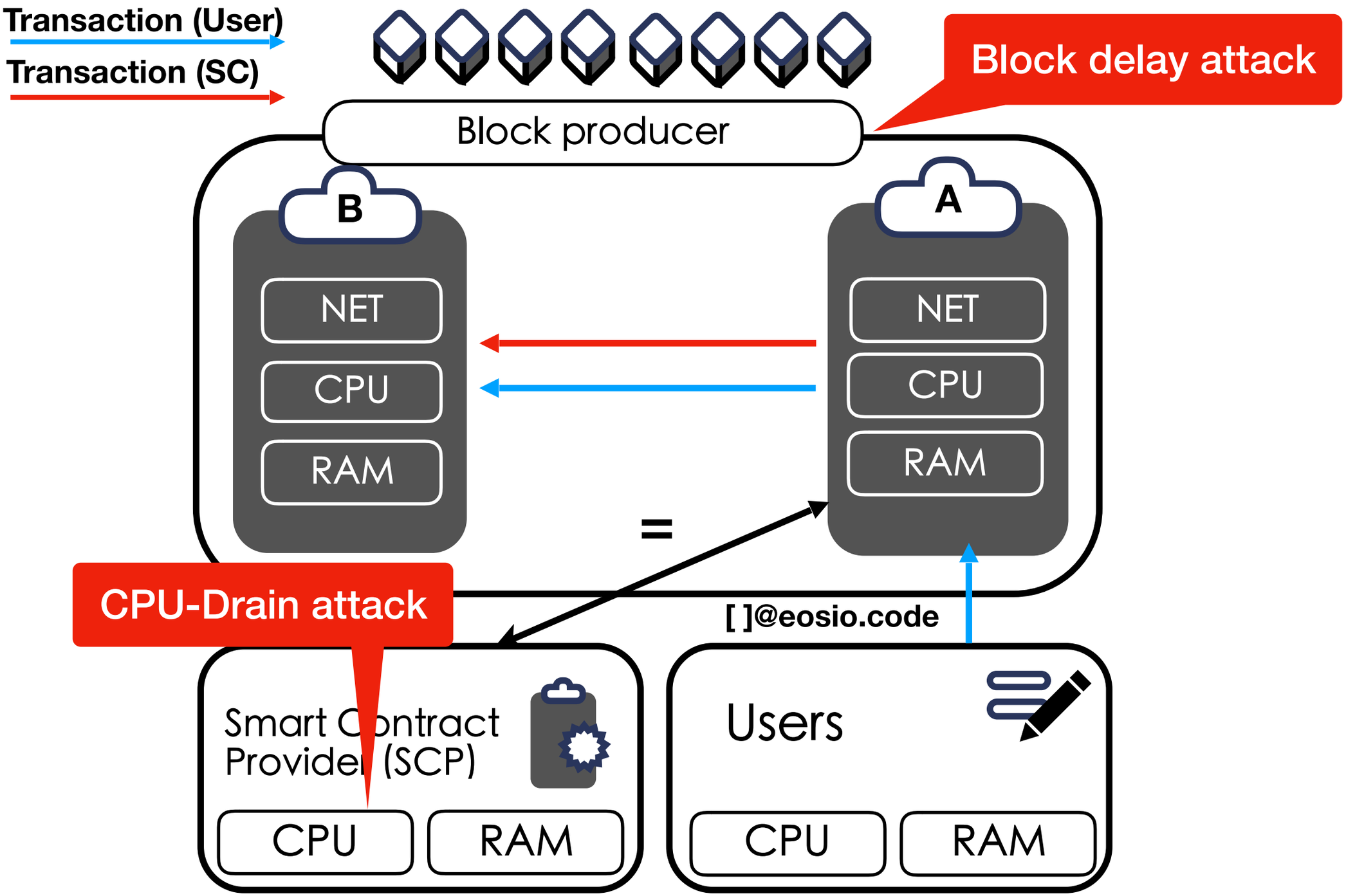
Attack Models & Threat Models & Attacks!

We found …



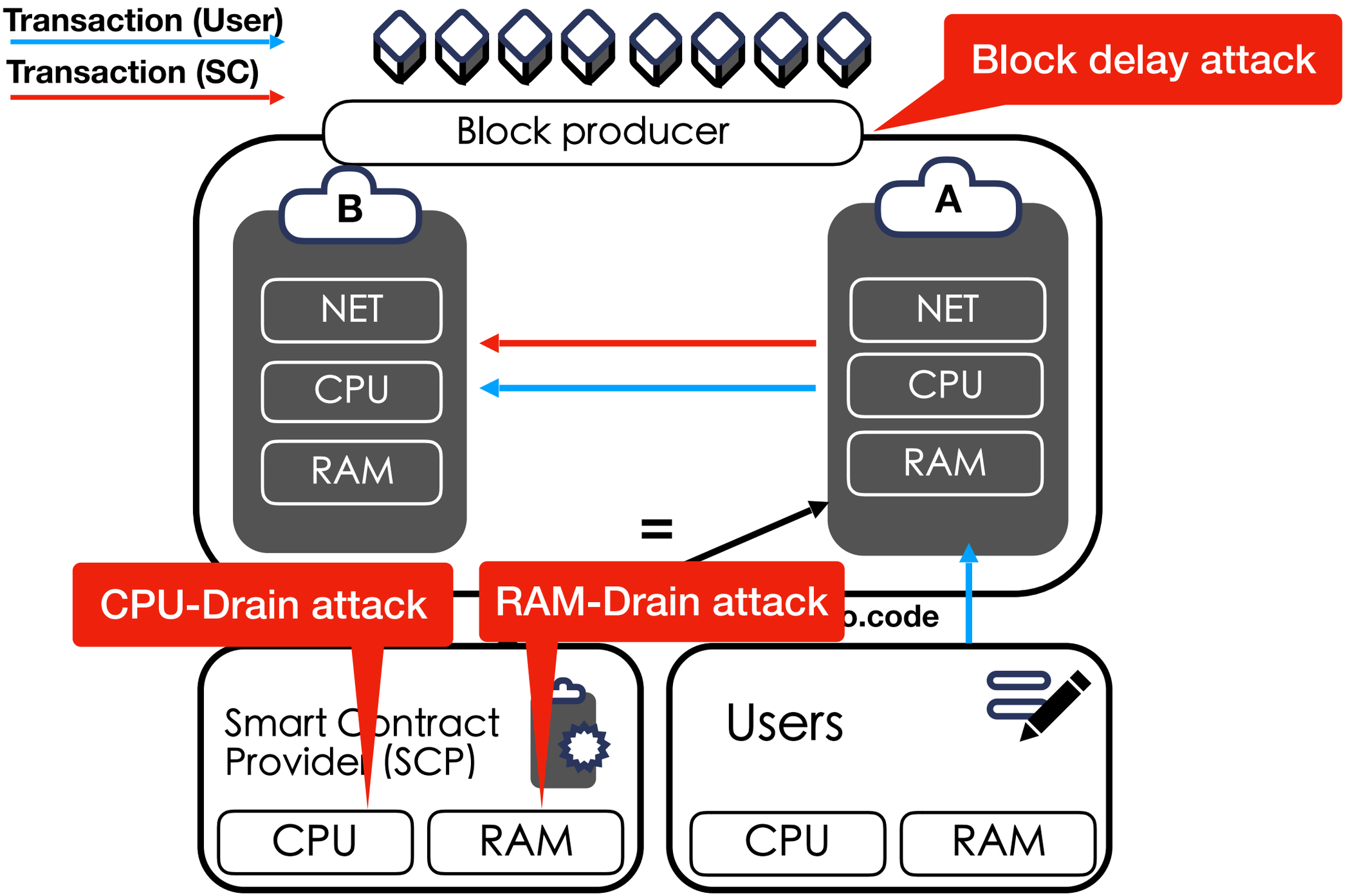
Attack Models & Threat Models & Attacks!

We found …



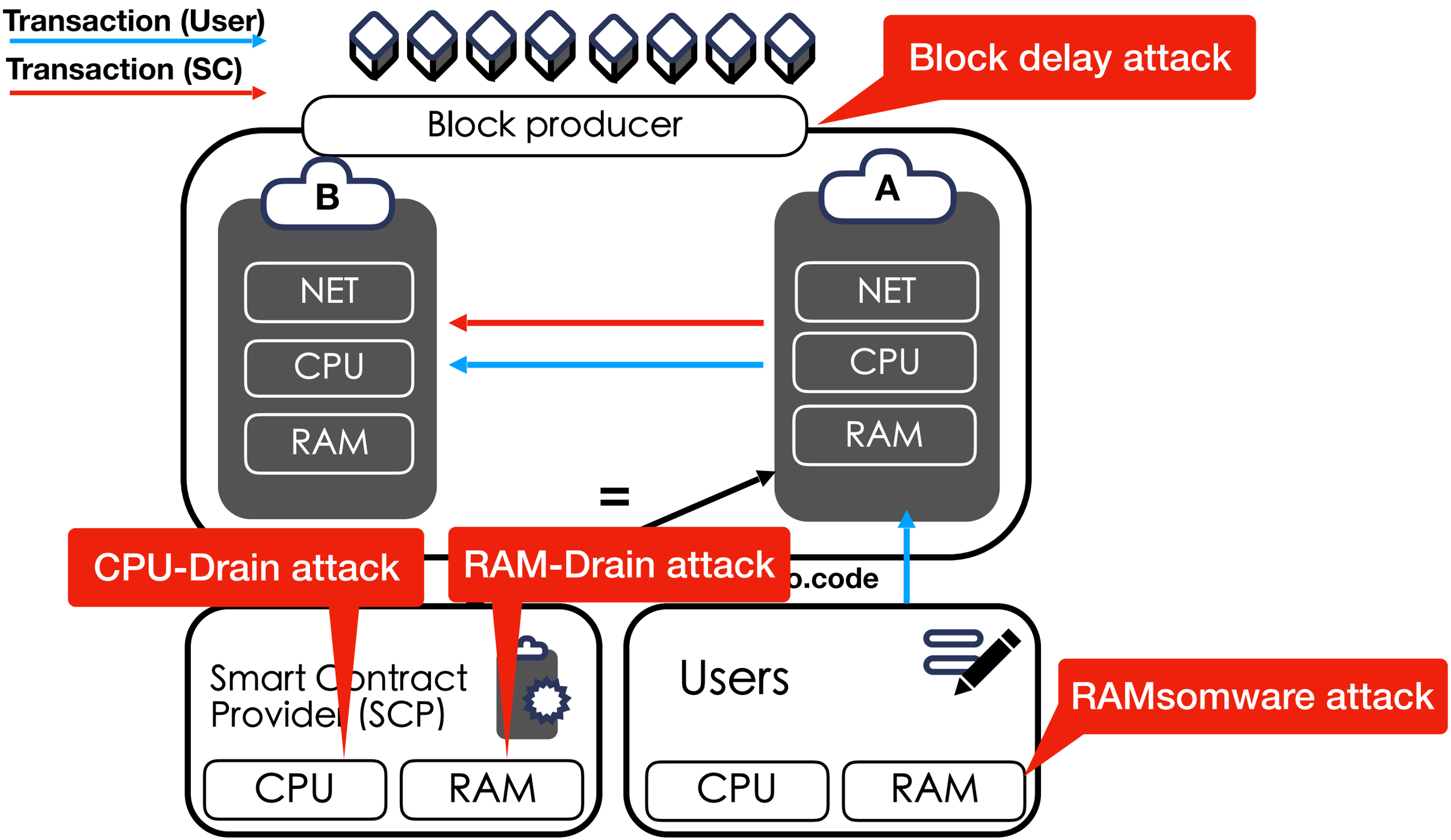
Attack Models & Threat Models & Attacks!

We found …



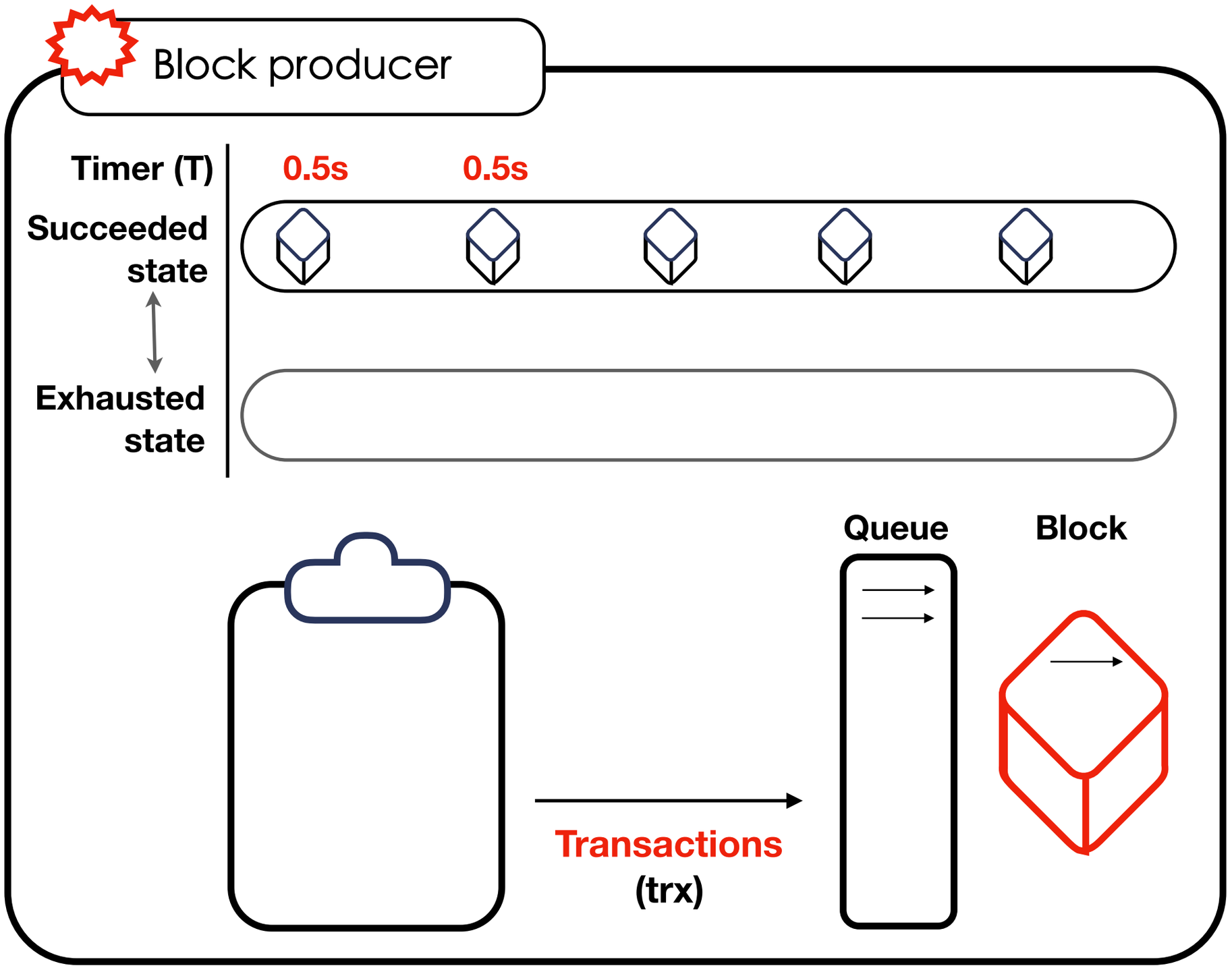
Attack Models & Threat Models & Attacks!

We found …

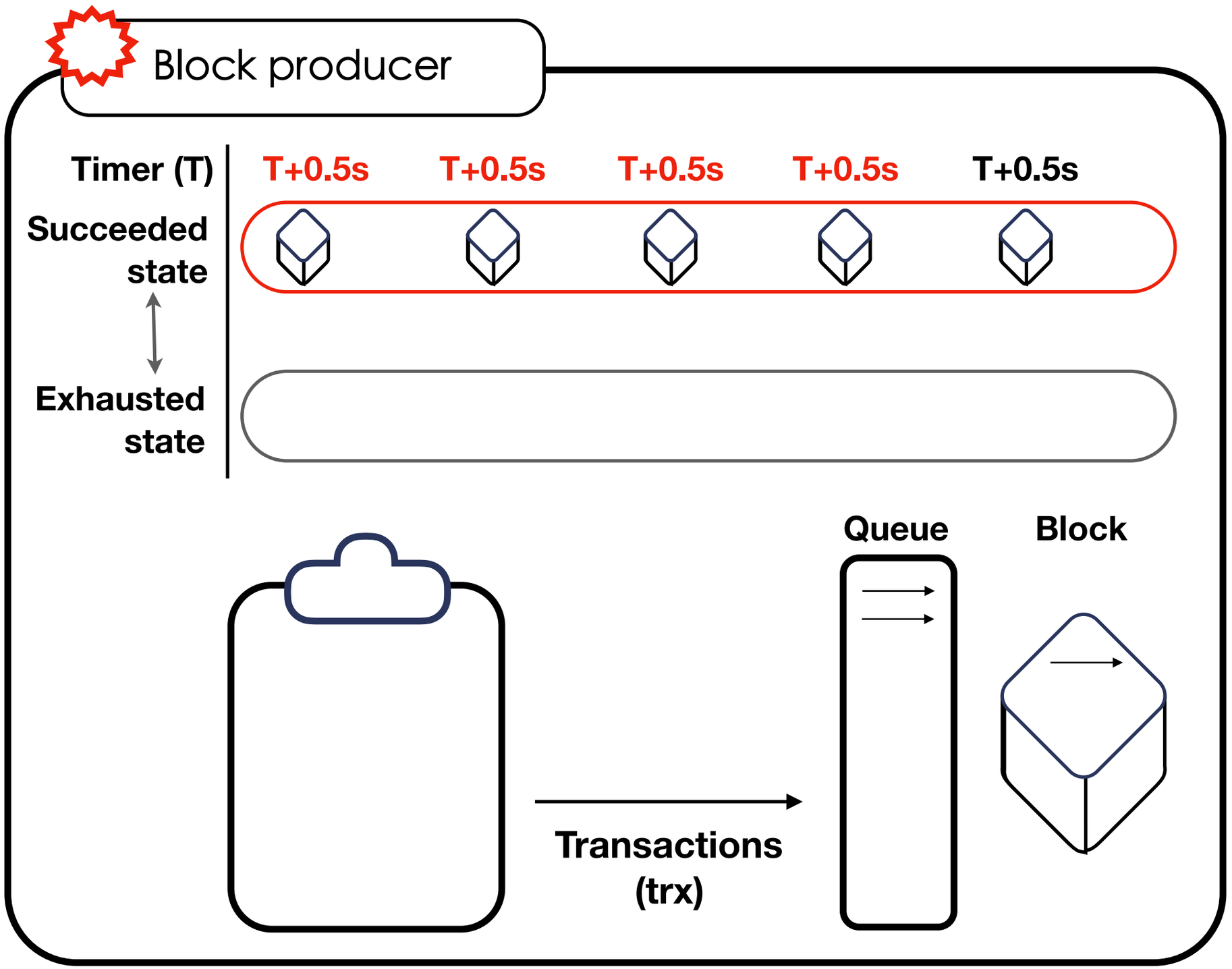


Attack

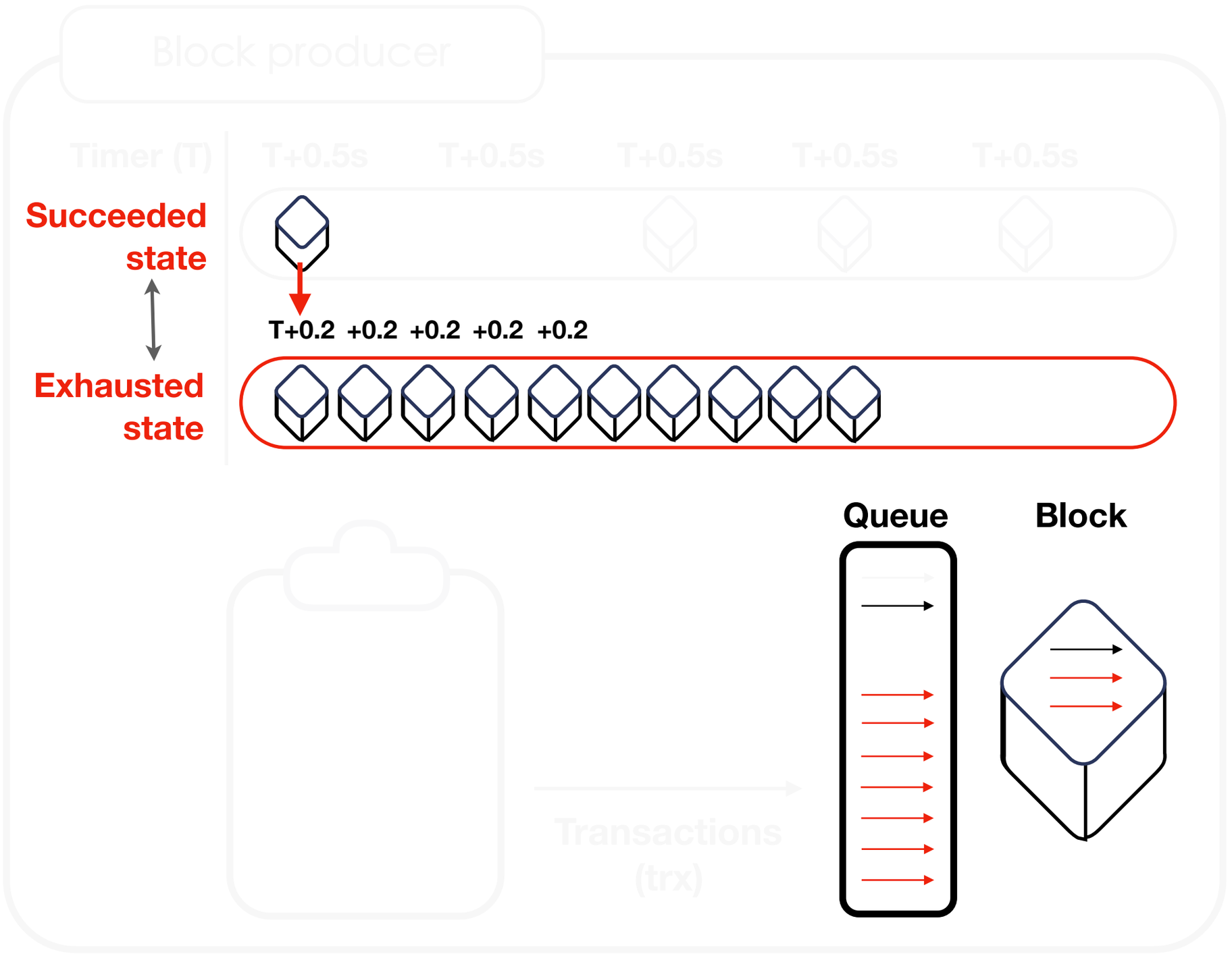
Block delay attack



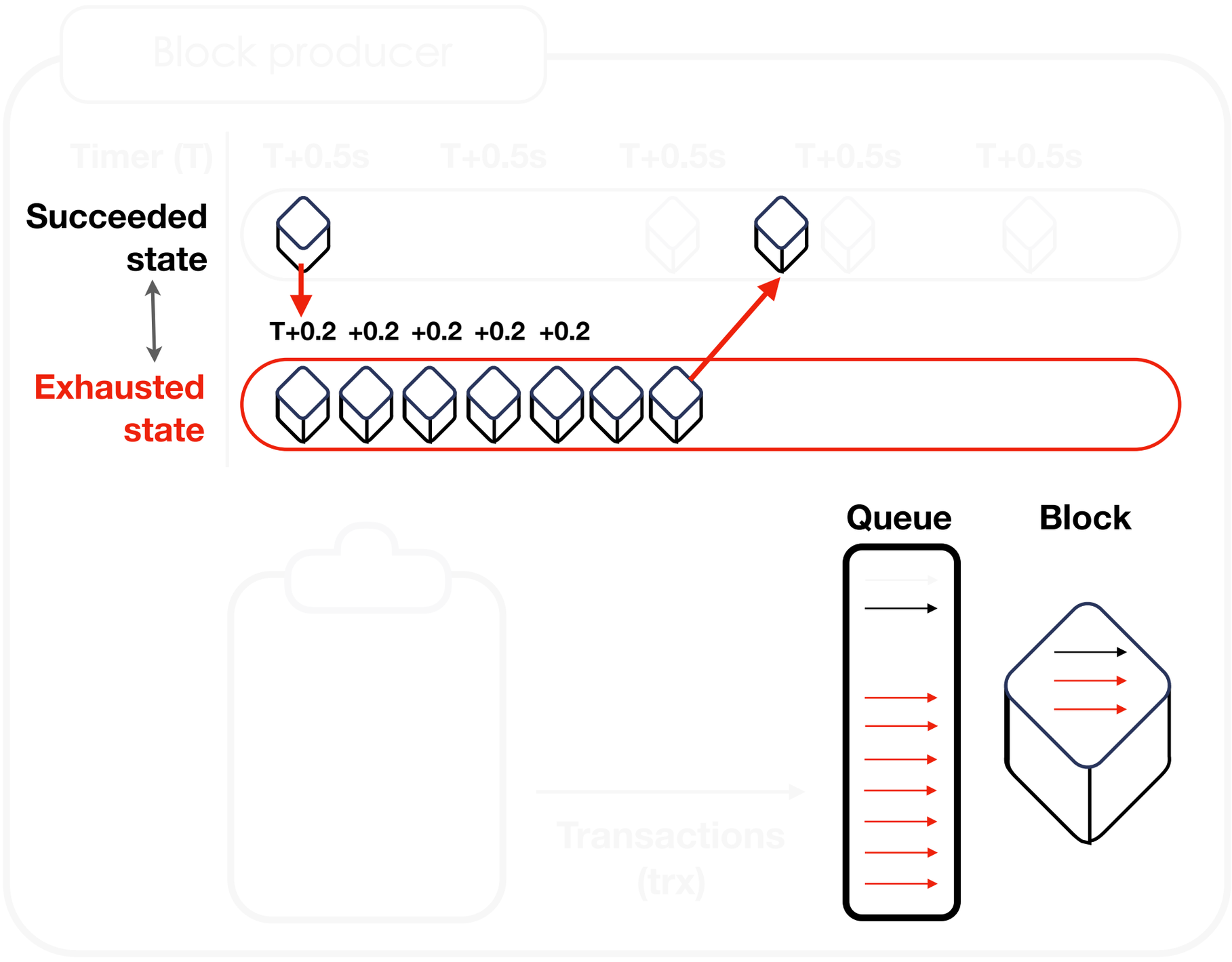
Block delay attack



Block delay attack

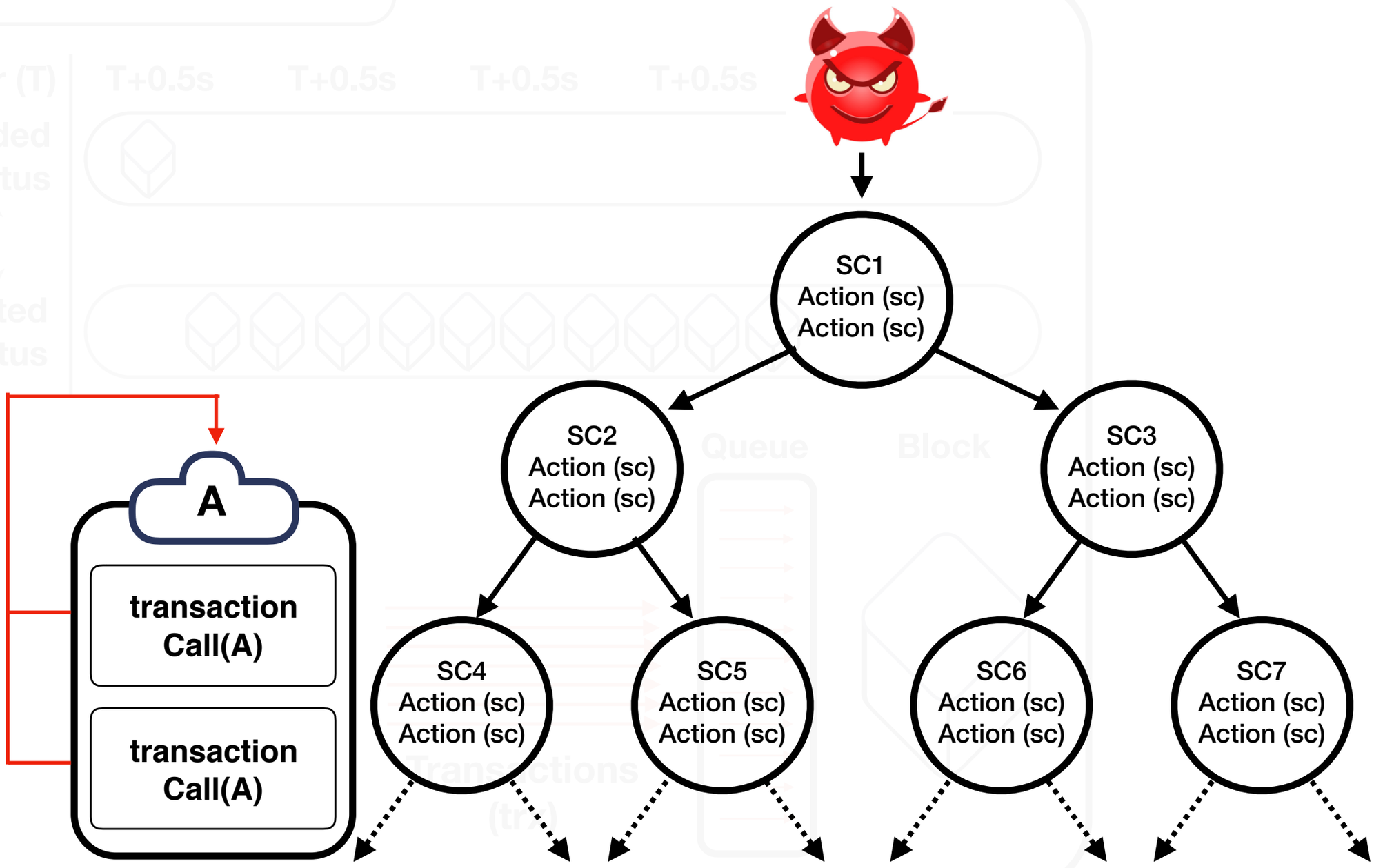


Block delay attack



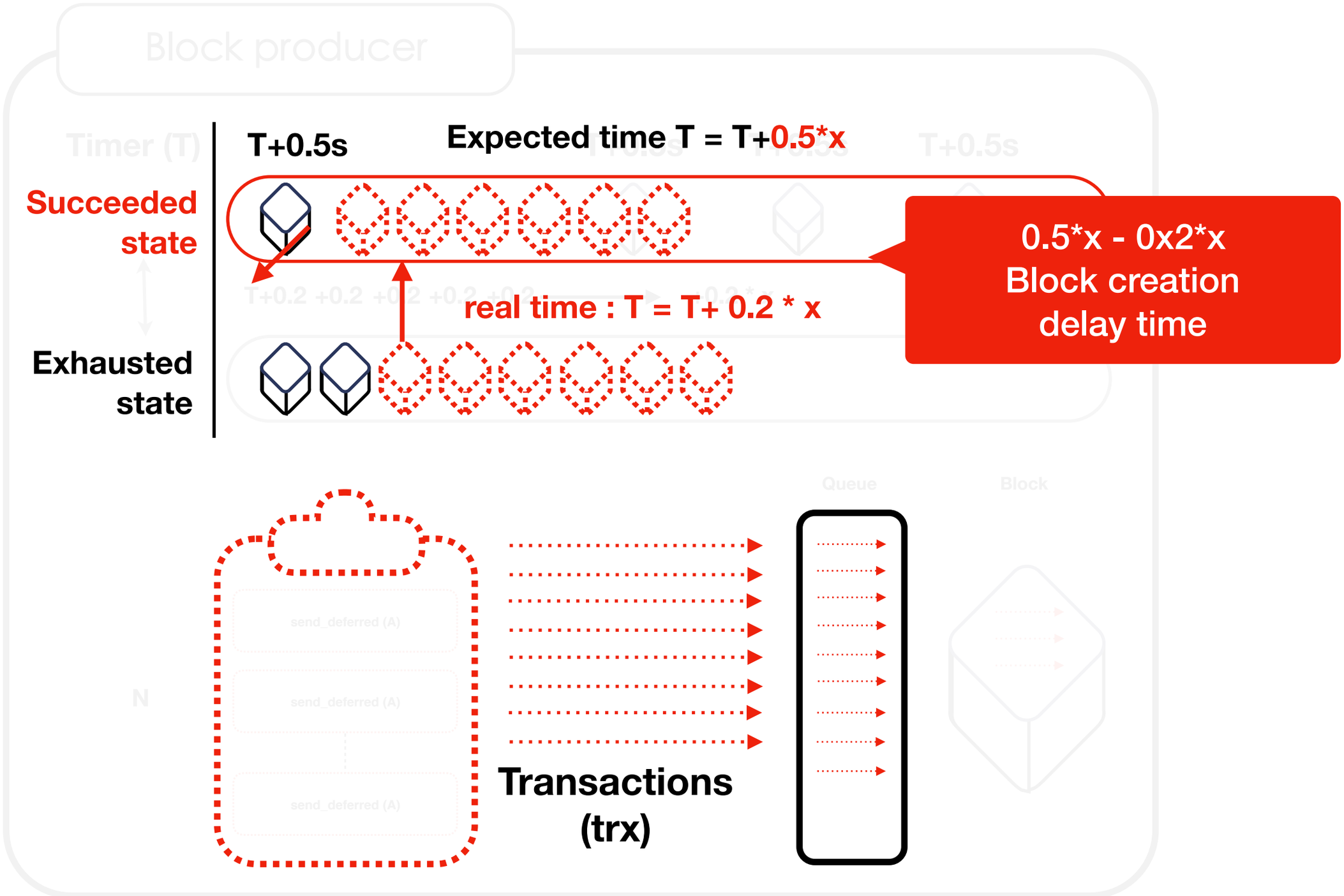
Block delay attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack



Block delay attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack



Block delay attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack

Estimated financial loss via block delay attack

	Attacker				Victim	
Block Count	Time (min)	Eos-CPU (min)	EOS-NET (MiB)	Cost (EOS)	Delay Time (min)	
376	0.92	1.23	16.13	480	2.05	
704	2.06	2.32	34.72	910	3.56	
1106	3.02	3.65	50.82	1,426	5.67	12,851
1471	4.00	4.85	65.53	1,894	7.46	148,478
1840	5.04(min)	6.07	79.69	2,368	9.12(min)	181,518

181,518 EOS == $880,000 USD (29/7/2019)

Average of EOS transfer volume (01/04/2019~30/04/2019)

Block delay attack

Estimated financial loss via block delay attack

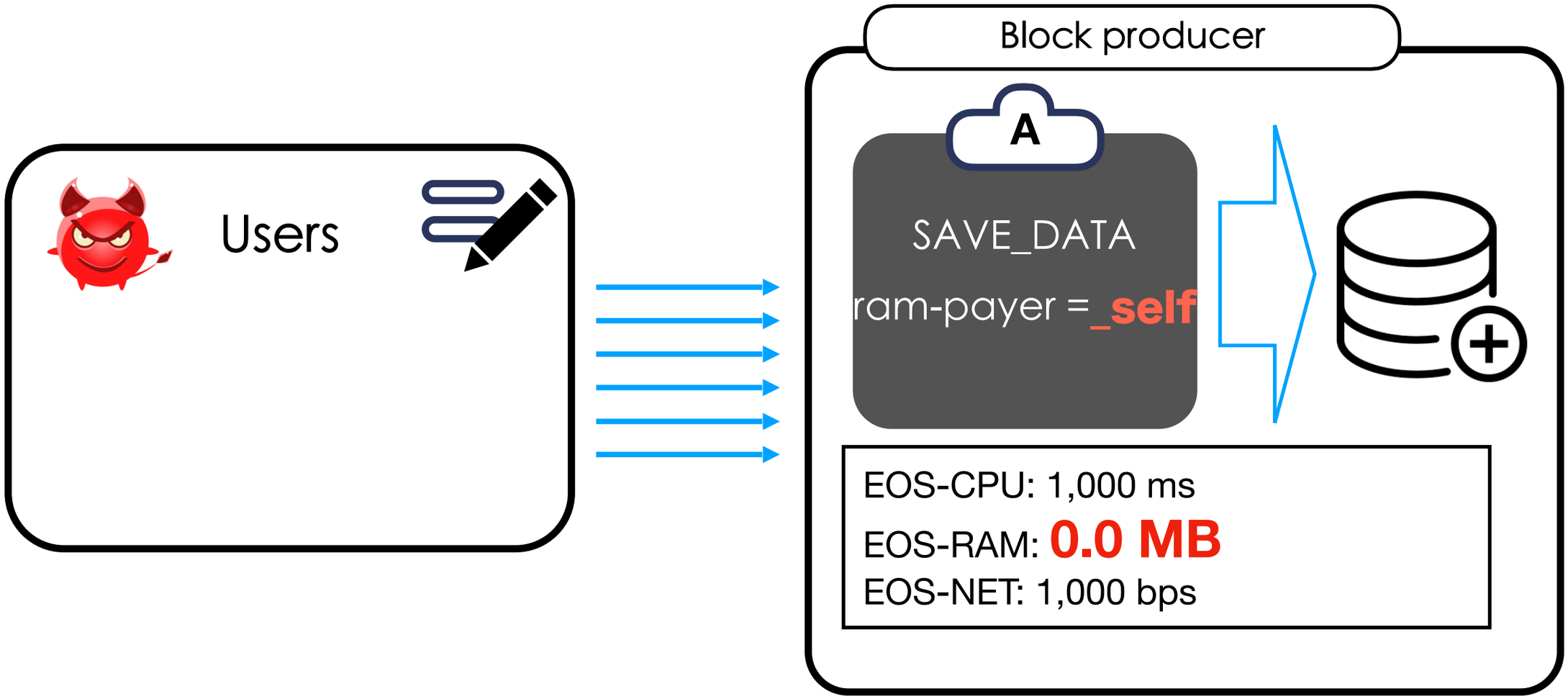
	Attacker				Victim	
Block Count	Time (min)	Eos-CPU (m		ost OS)	Delay Time (min)	Loss (EOS)
376	0.92	1			2.05	40,802
704	2.06				3.56	70,856
1106	3.02			6	5.67	112,851
1471	4.00	4.8		94	7.46	148,478
1840	5.04(min)	6.07	79.69	2,368	9.12(min)	181,518

We get maximum bug bounty
($10,000 USD)
From EOSIO foundation

표의 맨 아래쪽을 보면 5분동안 dummy transaction을 잔류 시키면, 약 9분간 block이 생성이 멈추는 결과를 얻을 수 있었다.

block 생성이 멈추면 EOS 의 모든 transaction 처리가 되지 않으며, 그 잠재적 손실은 191,518 EOS 이다.

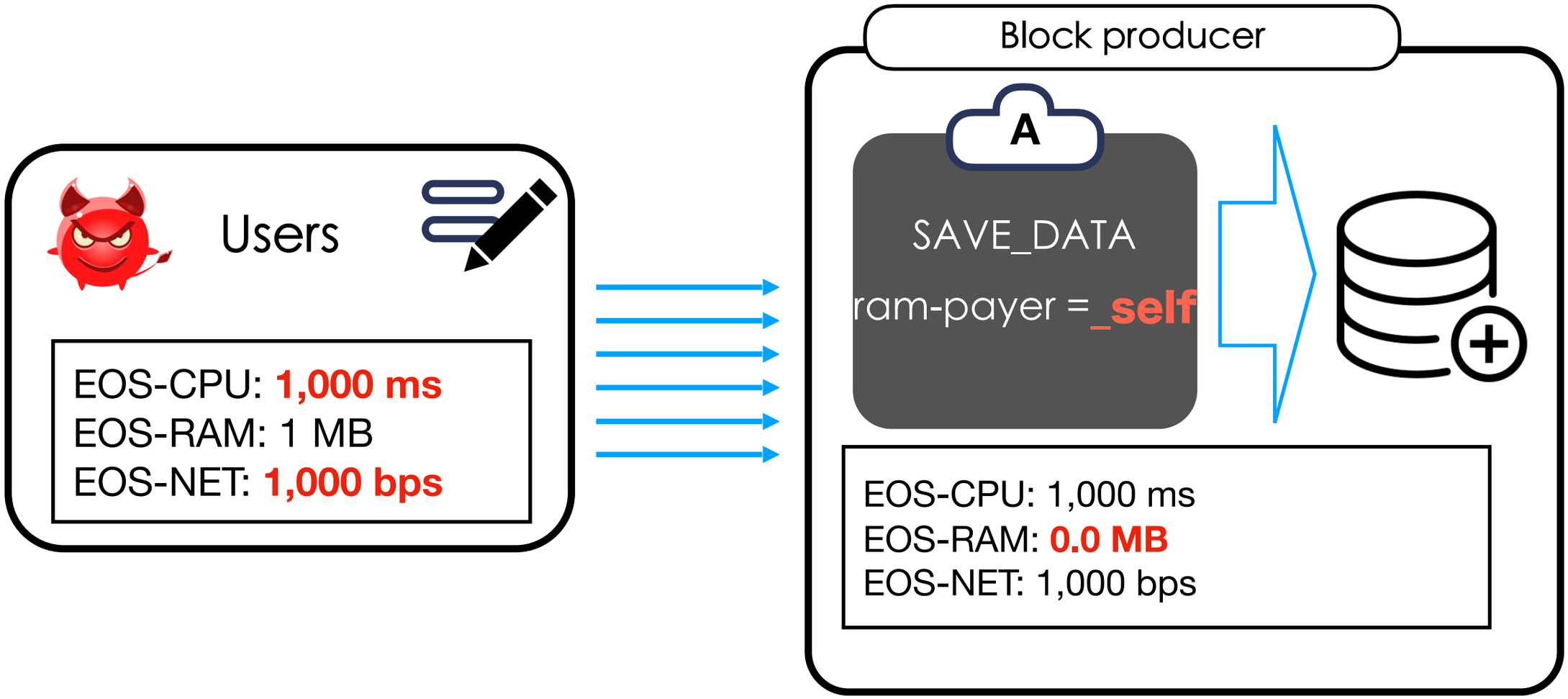
DoS by draining EOS resources: RAM-drain attack



EOS-RAM is purchase resource not stake,

so EOS-RAM doesn’t return until finishing their propose

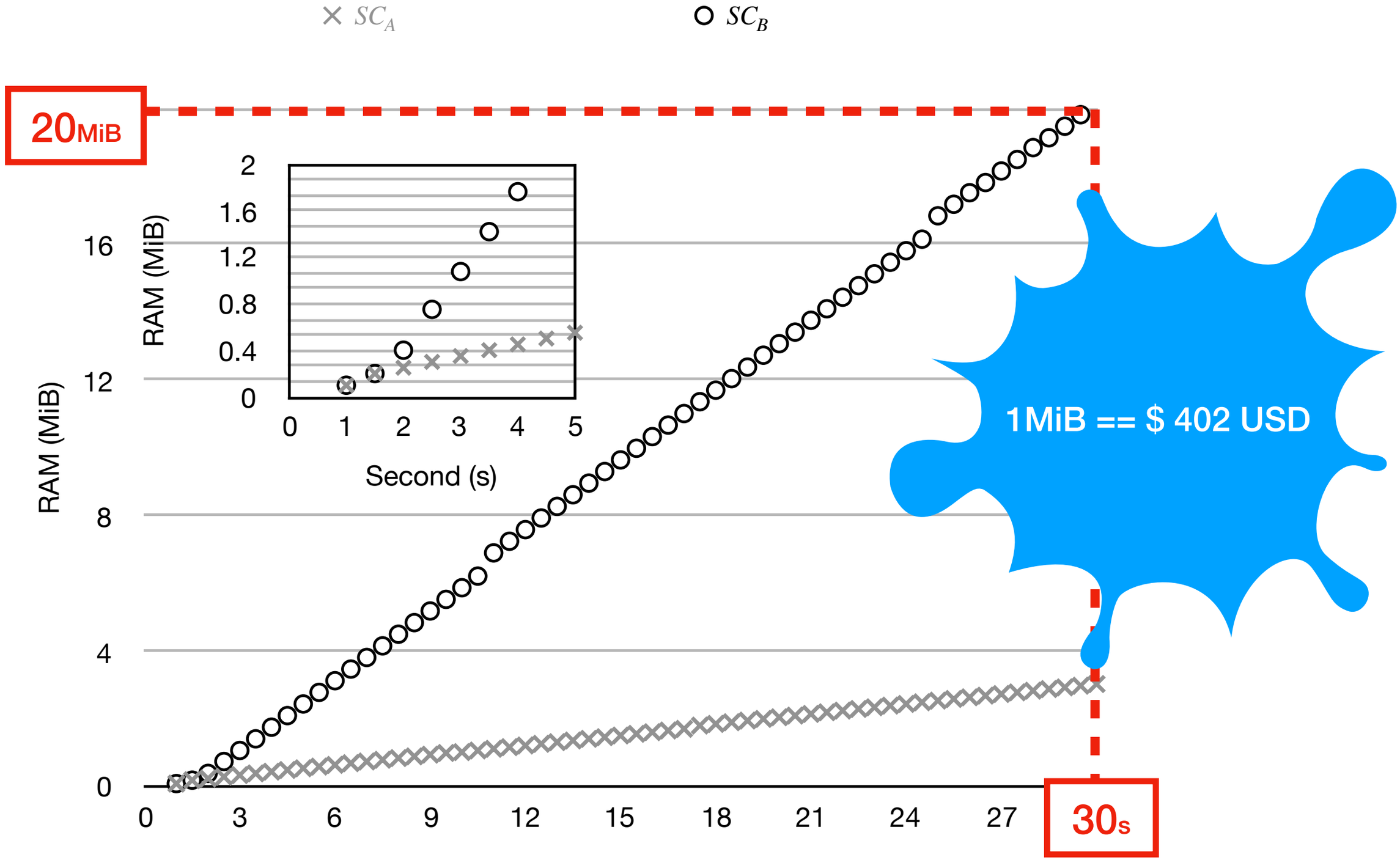
DoS by draining EOS resources: RAM-drain attack



EOS-RAM is purchase resource not stake,

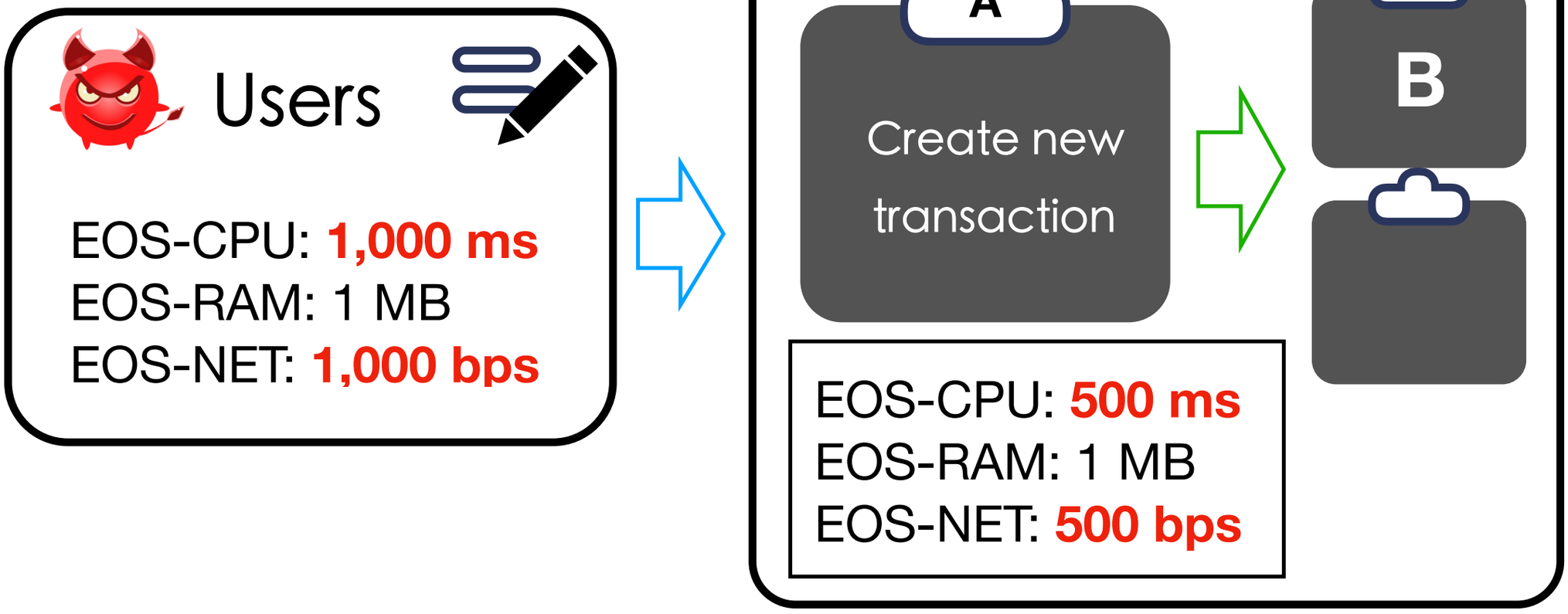
so EOS-RAM doesn’t return until finishing their propose

DoS by draining EOS resources: RAM-drain attack

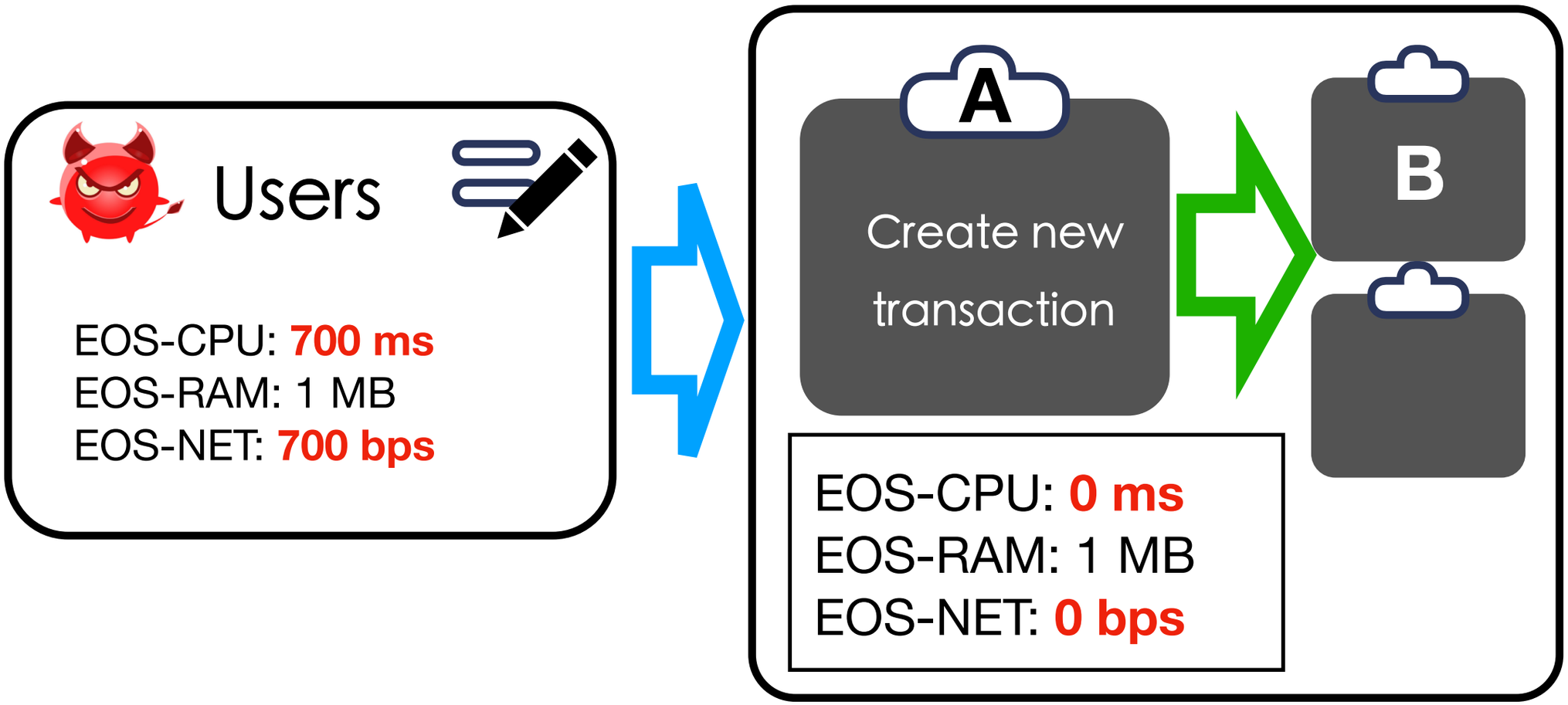


The Consuming time for RAM is depends on Smart contract's source code

DoS by draining EOS resources: CPU-drain attack



DoS by draining EOS resources: CPU-drain attack



Cost {EOS-CPU of $A} +
Cost {EOS-NET of $A}

<

Cost {EOS-CPU of $B} +
Cost {EOS-NET of $B}

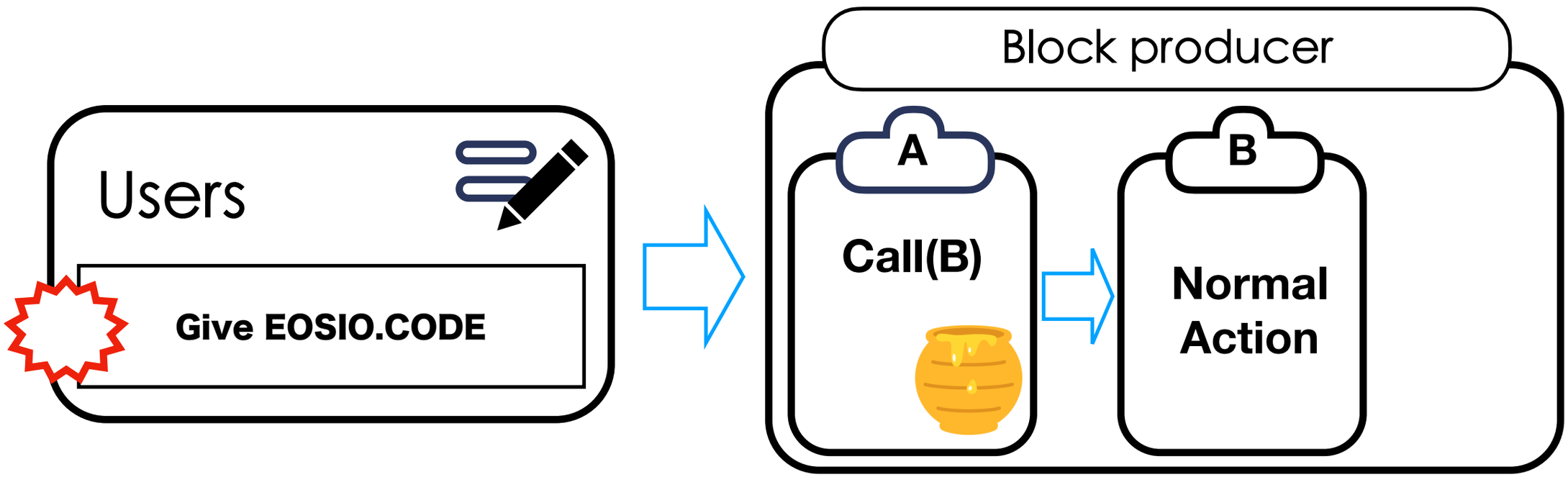
DoS by draining EOS resources: CPU-drain attack

	Attacker		m SC provider	
Attack Count	EOS-NET (KiB)			EOS-CPU (ms)
1	0.137			0.400
10	1.329			4.366
20	2.655	2.938	3.549	8.352
50	6.626	7.422	3.534	20.47
100	13.21	15.23	3.509	41.19

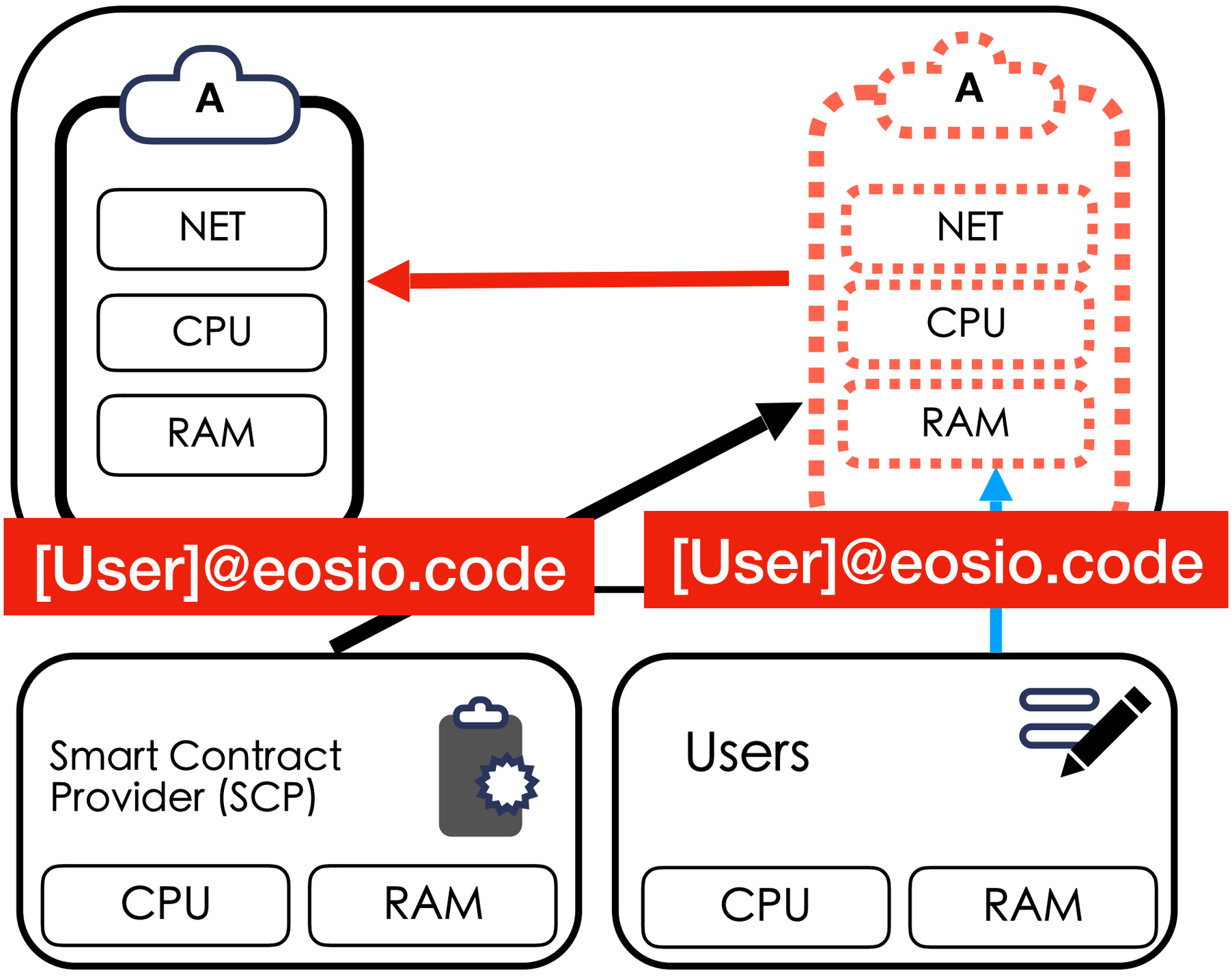
Attacker partially make DoS to victim while a day

Over x3 (times)

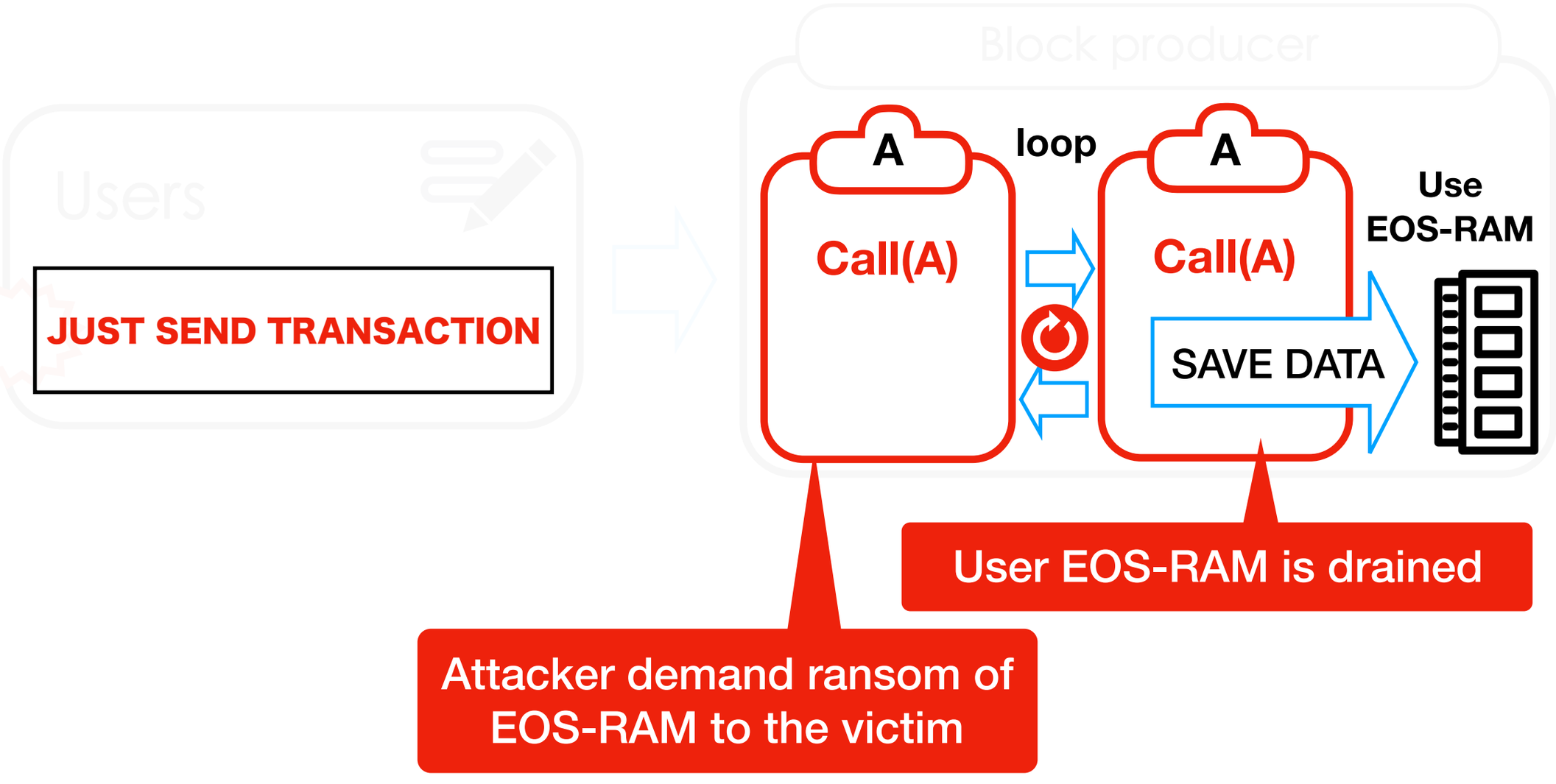
RAMsomware attack



RAMsomware attack

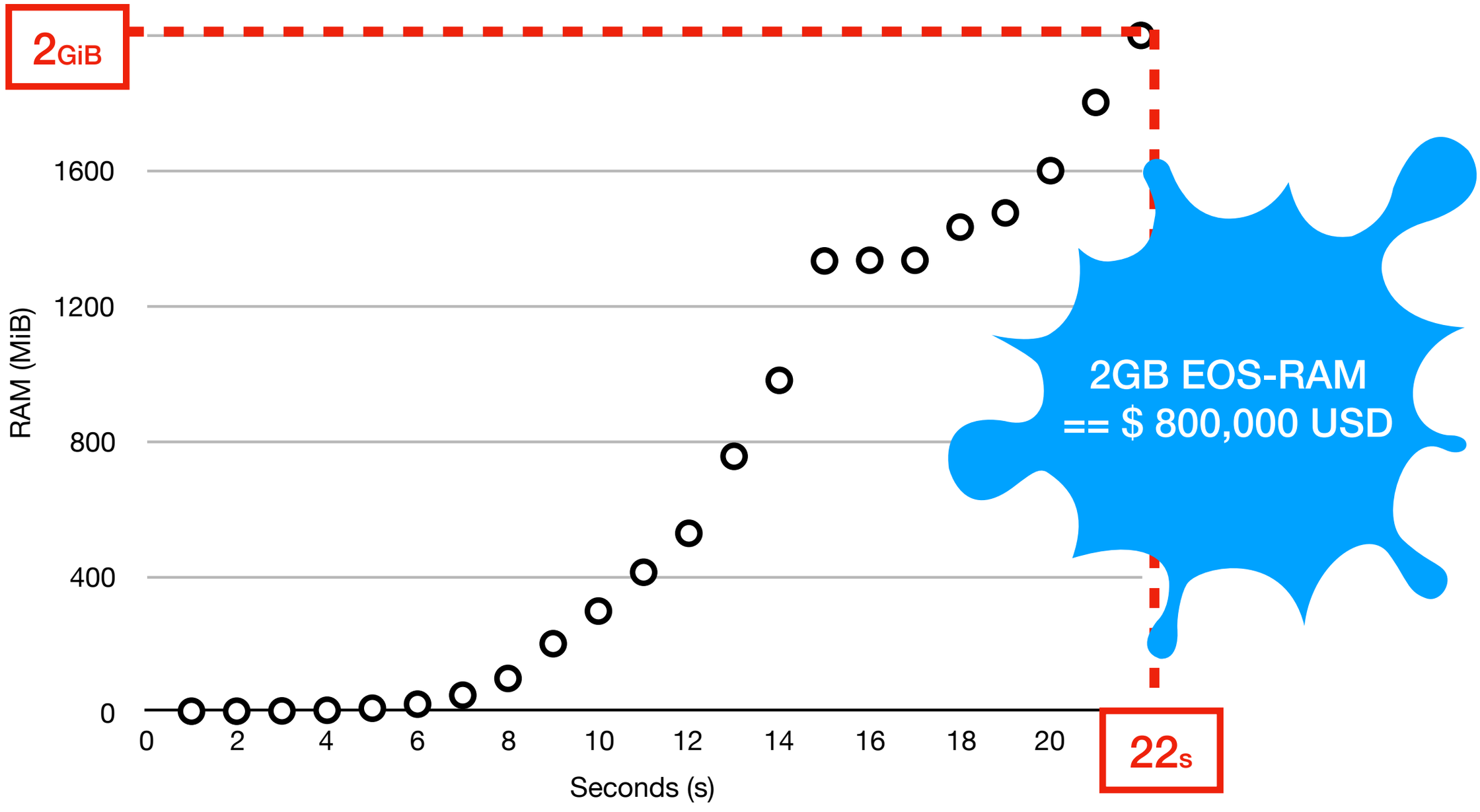


RAMsomware attack



RAMsomware attack

The user who have the largest EOS-RAM have 2GB EOS RAM



Defense

Defense

Trivial solution

- Block delay attack
- CPU/RAM drain attack
- RAMsomware attack

- Patched by EOSIO developers
- Do access control
- Do check smart contract version

Defense

Trivial solution

- Block delay attack
- CPU/RAM drain attack
- RAMsomware attack

- Patched by EOSIO developers
- Do access control
- Do check smart contract version

Design solution

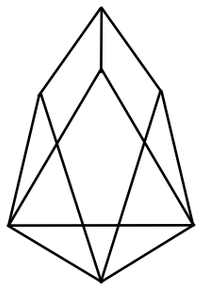
- Fine graind permission

: eosio.code Expire Time, Maximum EOS Coin per a transaction

: EOS-CPU permission, EOS-NET permission, EOS-RAM permission etc…

- Totally payment of transaction fee to the first transaction creator

: Every transaction that purpose a role, is payed by the users who start trx.



Conclusion & Future work

- **Conclusion**
  - Analyzed new threats from the view point of new resources in EOS.IO
  - Found 4 new attack methodologies and verified them
  - Proposed new security features to prevent our attacks
- **Future work**
  - Make an automatic auditing tool for our attacks
  - Design a web assembly analyzer

Thank you

{k1rh4, reset, dkay, sl.son, yongdaek}@kaist.ac.kr