Gas metering:

Appchain vs General Purpose Rollups

Felipe Argento

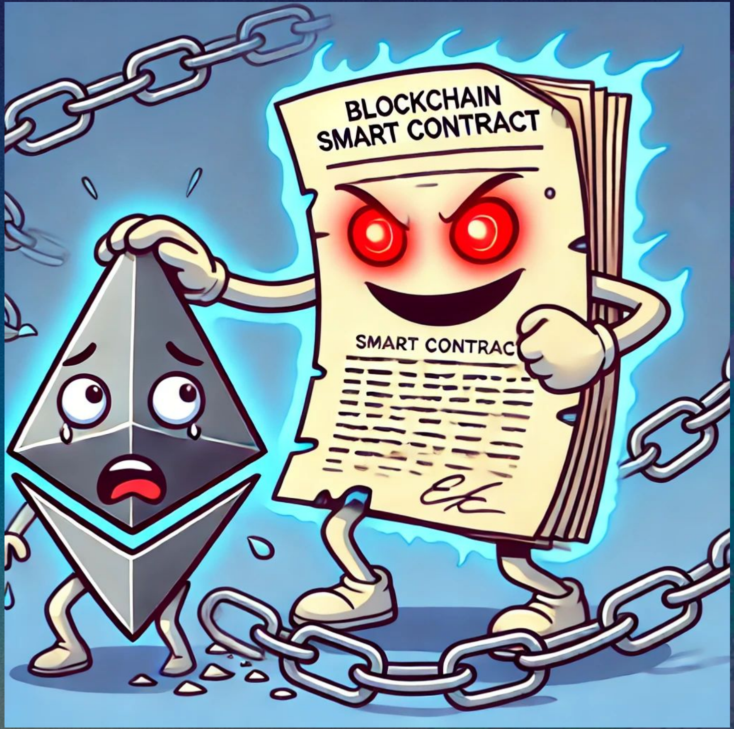
cofounder, Cartesi

We needed a defense system

A reason behind Gas is defense against **Resource Exhaustion attacks**

We're being attacked!

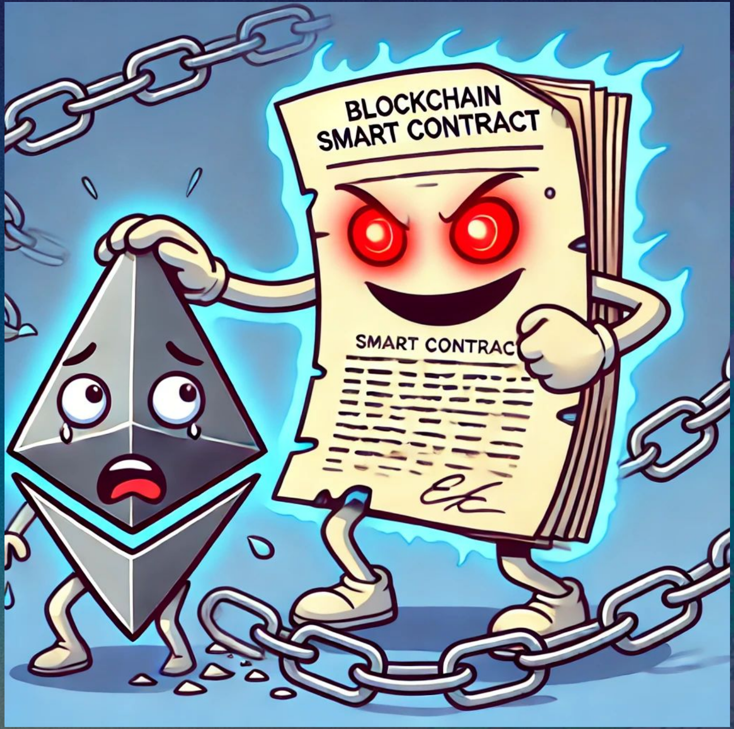
Imagine there is a transaction trying to make **the Ethereum network halt.** What should we do?



We're being attacked!

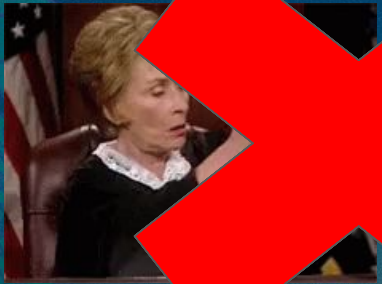
Imagine there is a transaction trying to make **the Ethereum network halt.** What should we do?

Look at the wall clock!

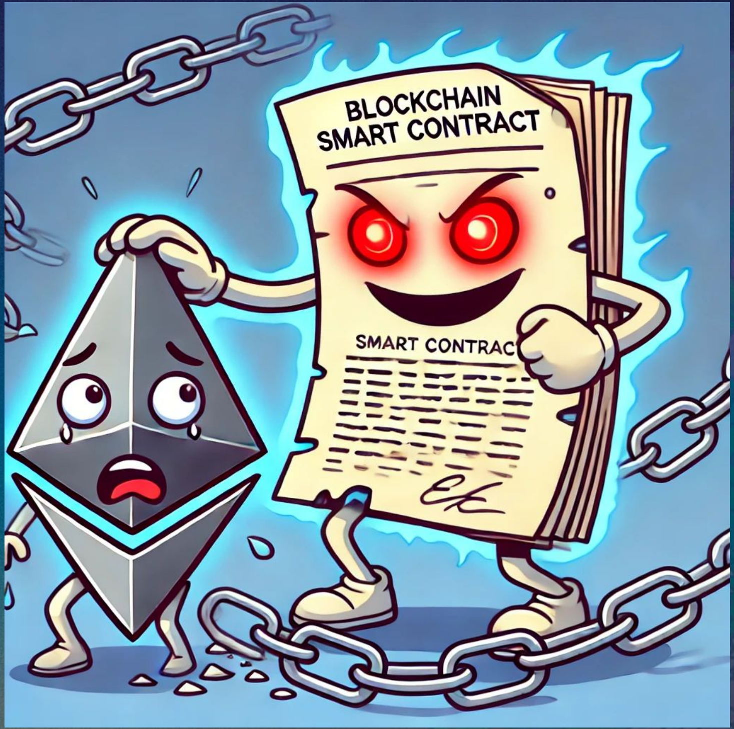


We're being attacked!

Imagine there is a transaction trying to make **the Ethereum network halt.** What should we do?



ok at the wall
k!



The gas metering system

- **Deterministic** measure of cost/time for every **opcode**
- Each **transaction** can spend an amount of gas
- Each **block** has an **upper gas limit**
- The **reference computer** is expected to be able to process a **block of gaslimit x** every **y time**

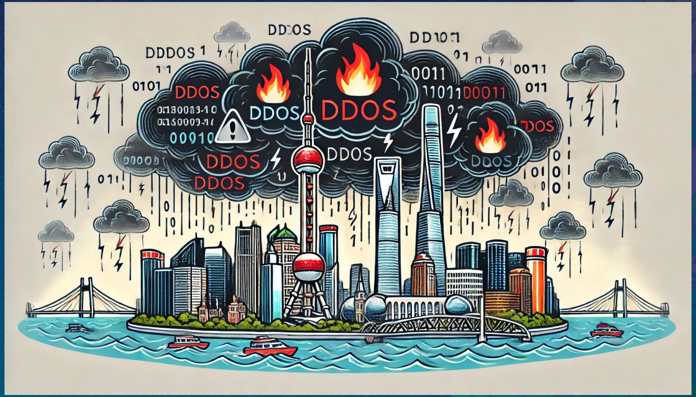
What can go wrong?

- These **estimations** are quite **hard** to do
- **Mispriced** opcodes are **dangerous**
- **Average case** is **VERY different** than **worst case**

What went wrong??

Shanghai Attacks, 2016:

- Miners had to reduce the block gas limit
- EIP 150 created to fight the bad guys



What made it worse:
Attackers were able to permissionlessly deploy code to trigger the badly estimated scenario

Shared Chains vs Appchains

Ethereum / Shared Rollups:

Users → Application → Chain

Application specific rollups:

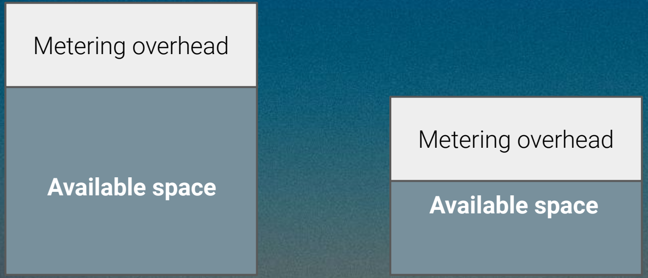
Users → Application 🫶 Chain

Shared Chains Metering

Adversarial metering

Granularity: opcode
Threat: data + custom code
Mismatch: worst case >>> average case

Blocksize:



Metering overhead

Available space

Average Case

Metering overhead

Available space

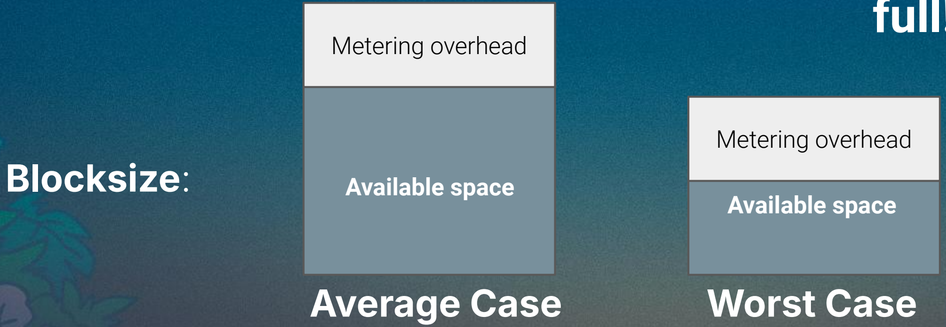
Worst Case

Shared Chains Metering

Adversarial metering

Granularity: opcode
Threat: data + custom code
Mismatch: worst case >>> average case

The reference computer is quite idle, even when the blocks are full!



Blocksize:

Metering overhead

Available space

Average Case

Metering overhead

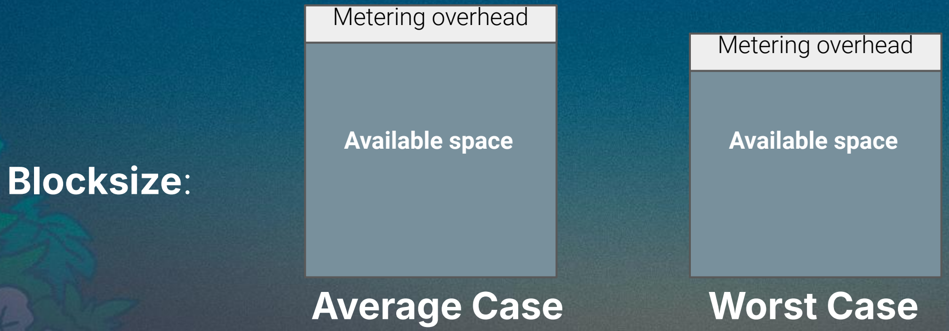
Available space

Worst Case

AppChains Metering

Cooperative metering

Granularity: Interactions (attack goblin)
Threat: only data!
Mismatch: worst case close to average case (data only)



Blocksize:

Metering overhead

Available space

Average Case

Metering overhead

Available space

Worst Case

Thank you!

Felipe Argento

Cofounder, Cartesi Foundation
felipe.argento@cartesi.foundation
@felipeargento

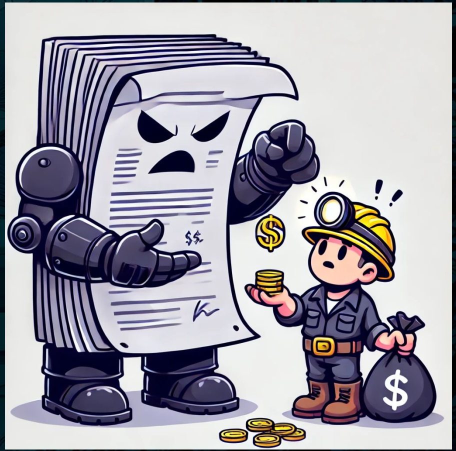
Gas Metering

- Deterministic measure of cost/time for every opcode
- Each transaction has a gas limit
- Each block has an upper gas limit
- The reference computer is expected to be able to process that gas limit faster than the blocktime

cartesi

What can go wrong??

- These estimations are quite hard to do
- Mispriced opcodes are dangerous
- Average case is VERY different than worst case

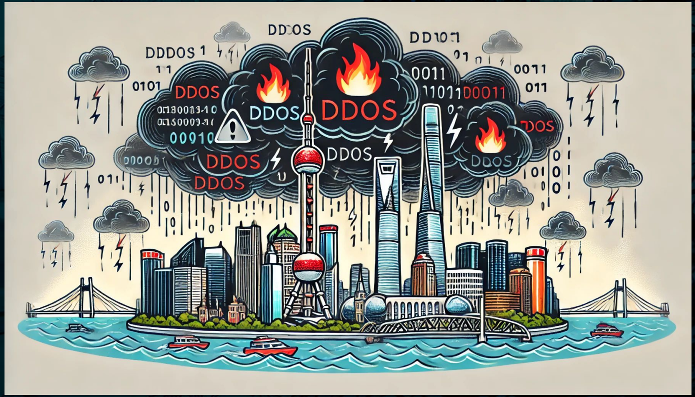


What went wrong??

Shanghai Attacks, 2016:

- Miners had to reduce the block gas limit
- EIP 150 created to fight the bad guys

What made it worse:
Attackers were able to permissionlessly deploy code to trigger the badly estimated scenario



Shared Chains vs Appchains

Ethereum / Shared Rollups:

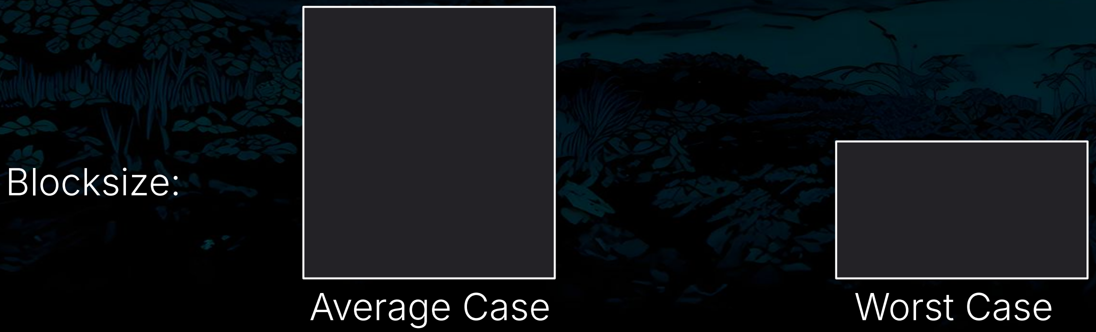
Users → Application → Chain

Application specific rollups:

Users → Application 🫶 Chain

Adversarial Metering

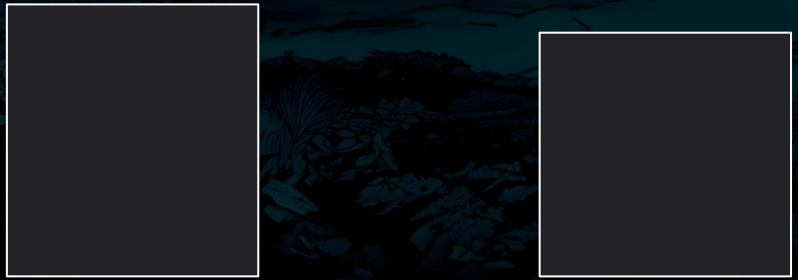
Granularity: opcode + details
Threat: data + custom code
Mismatch: worst case >>> average case



Cooperative Metering

Granularity: Interactions (attack goblin) + details
Threat: only data!
Mismatch: worst case close to average case (data only)

Blocksize:



Average Case

Worst Case

cartesi

Thanks!

Section 1 title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
    - Condimentum
        - Magna
        - Ligula