S.C.R.E.A.M

James Prestwich

James Prestwich

init4.technology

engineering
+ Razzlekhan Cosplayer
+ the guy from footloose
  that banned dancing

Twitter: @_prestwich
Github: prestwich

this talk is not well laid out
and will be an awful lot

This is the state (just roll with it)

State Contention: multiple parties

State Contention: re-ordering changes outcomes

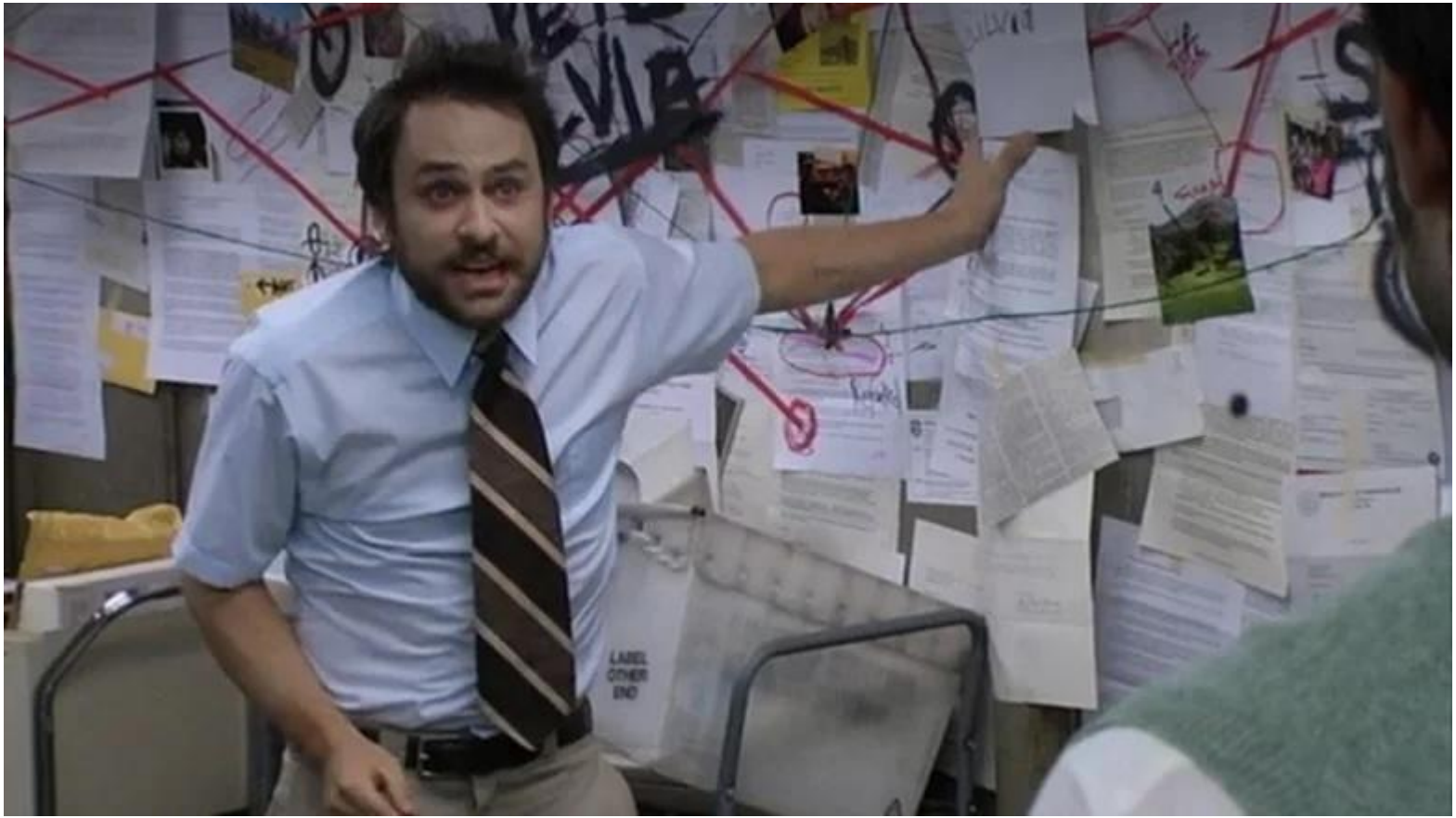
State Contention: re-ordering changes outcomes

(ranting about concurrency)

Why does this matter?

The OS and the CPU mediate concurrency

Ethereum’s OS has a knife and wants your wallet.


LABEL
OTHER
END

Let’s talk about the mail, Mac.

f()

A scoring function

perfect tree

f() =

10/10

no notes

My scoring function

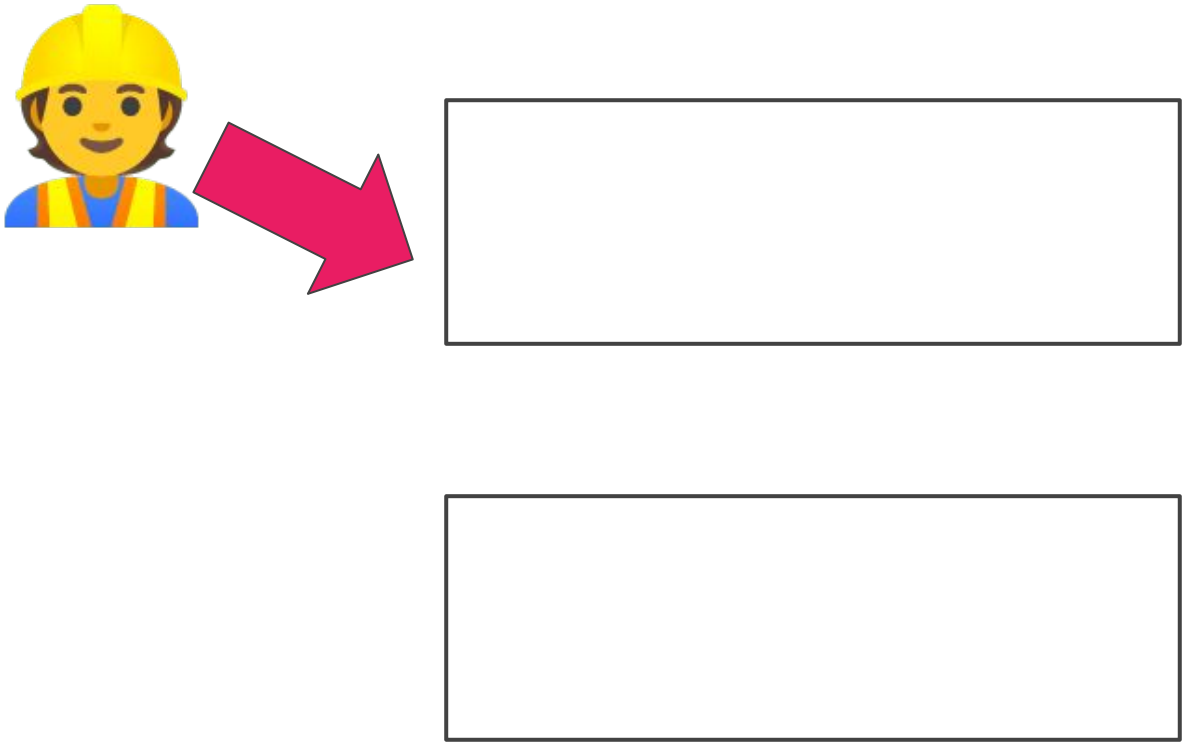
f()

f()

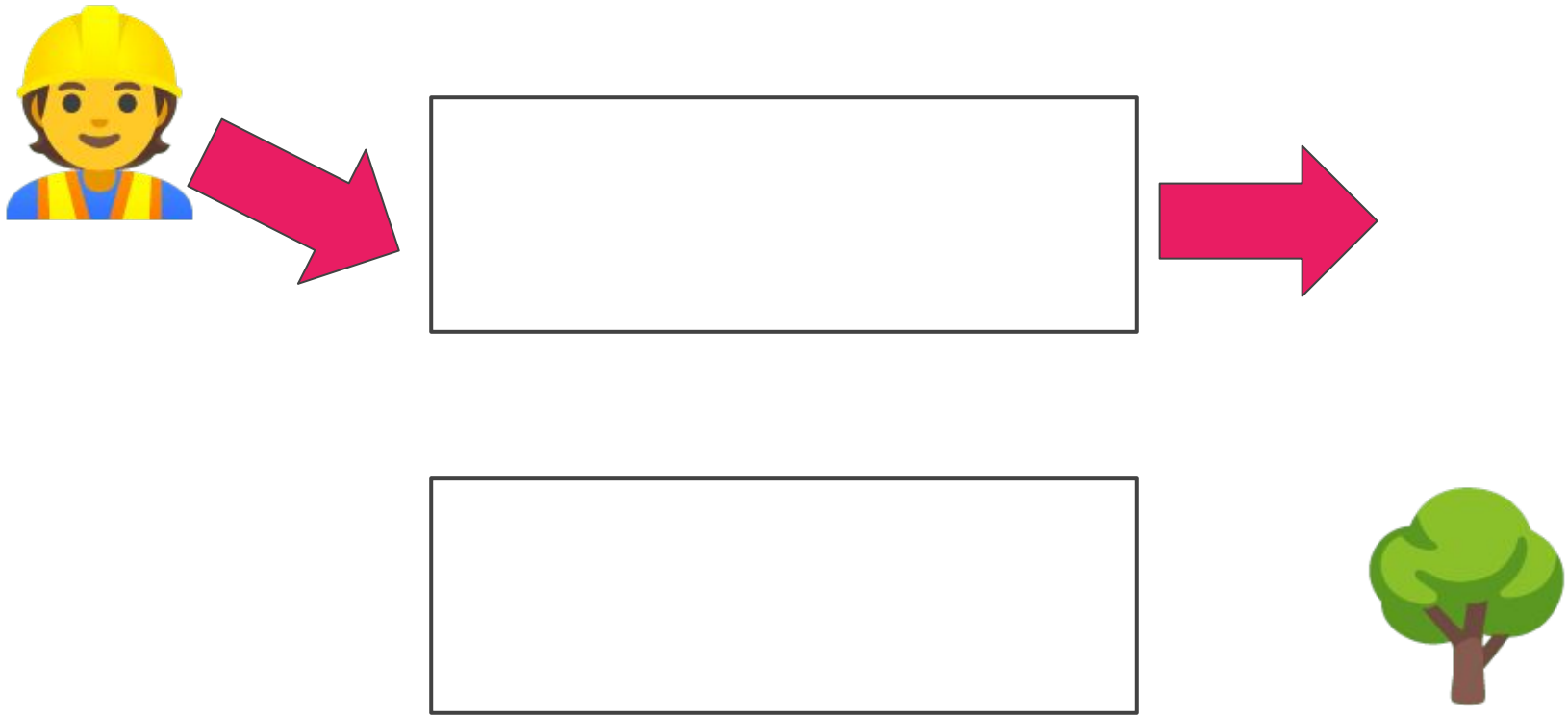
f()

Outcome scoring

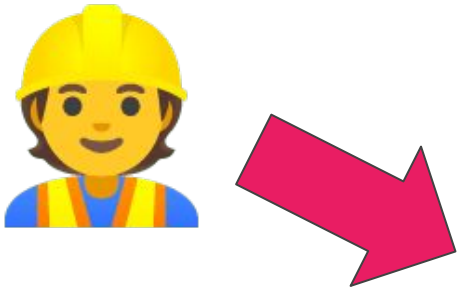
Builders



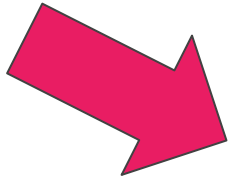
Builders decide order

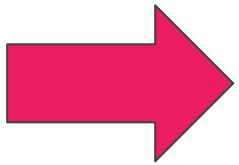


Order determines outcome



Therefore, builder decides outcome* (this is a really important *)

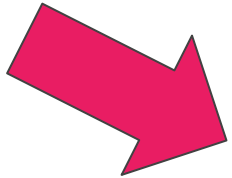




f()

f()

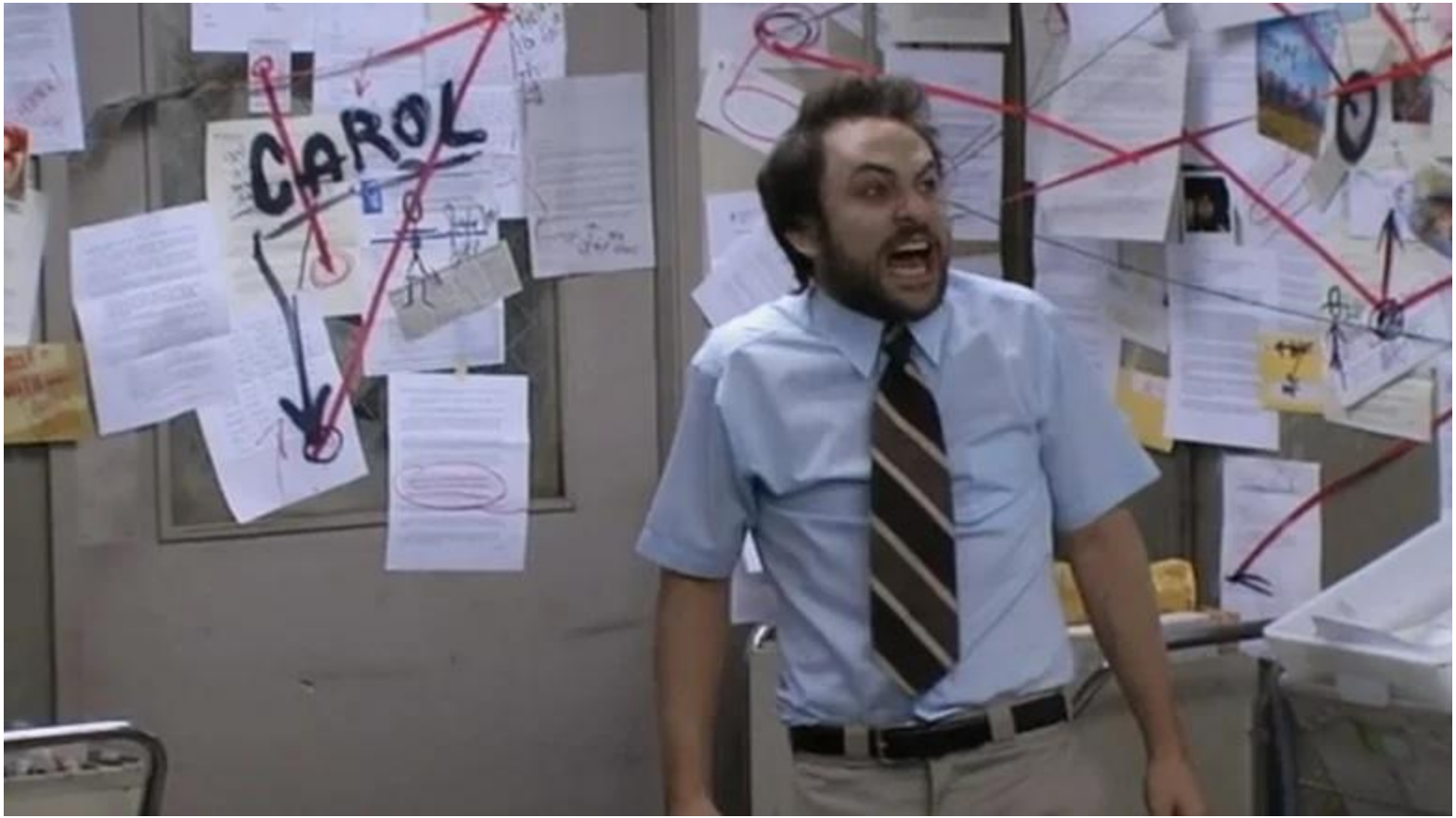
Outcome decides user happiness*



f(🌲)

f(🌳)

Therefore, the builder decides how happy you are*


CAROL

* and if they can affect the state

* if they can simulate things accurately

asterisk part 1: order invariance

f()

g()

h()

There are many choices of scoring function

h()

A different scoring function

h()

Perfect tree
10/10 no notes

h()

Perfect tree
10/10 no notes

Order invariant scoring functions

h()

Perfect tree
10/10 no notes

h()

Perfect tree
10/10 no notes

also me (in my lane, flourishing)

Order invariant scoring functions

okay so examples?

Order invariant

- I have to pay Yaz 1 ETH to defray his aragon court costs.

- I need to add more collateral to my maker vault

- I am transferring an nft pfp because i live like it is 2021 and this is still cool

h()

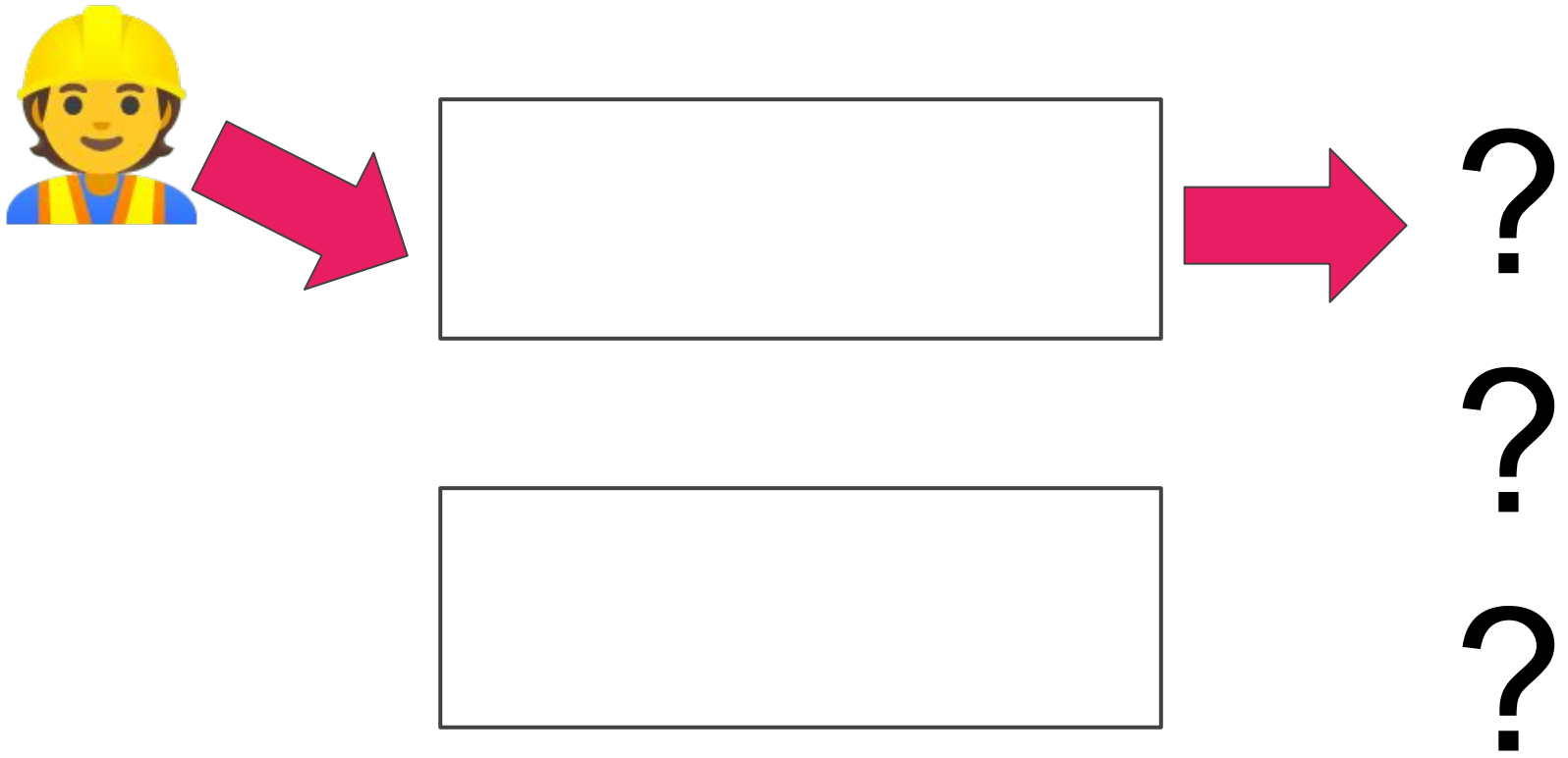
Perfect tree
10/10 no notes

h()

Perfect tree
10/10 no notes

takeaway: if the state is not contentious, the outcome may not change

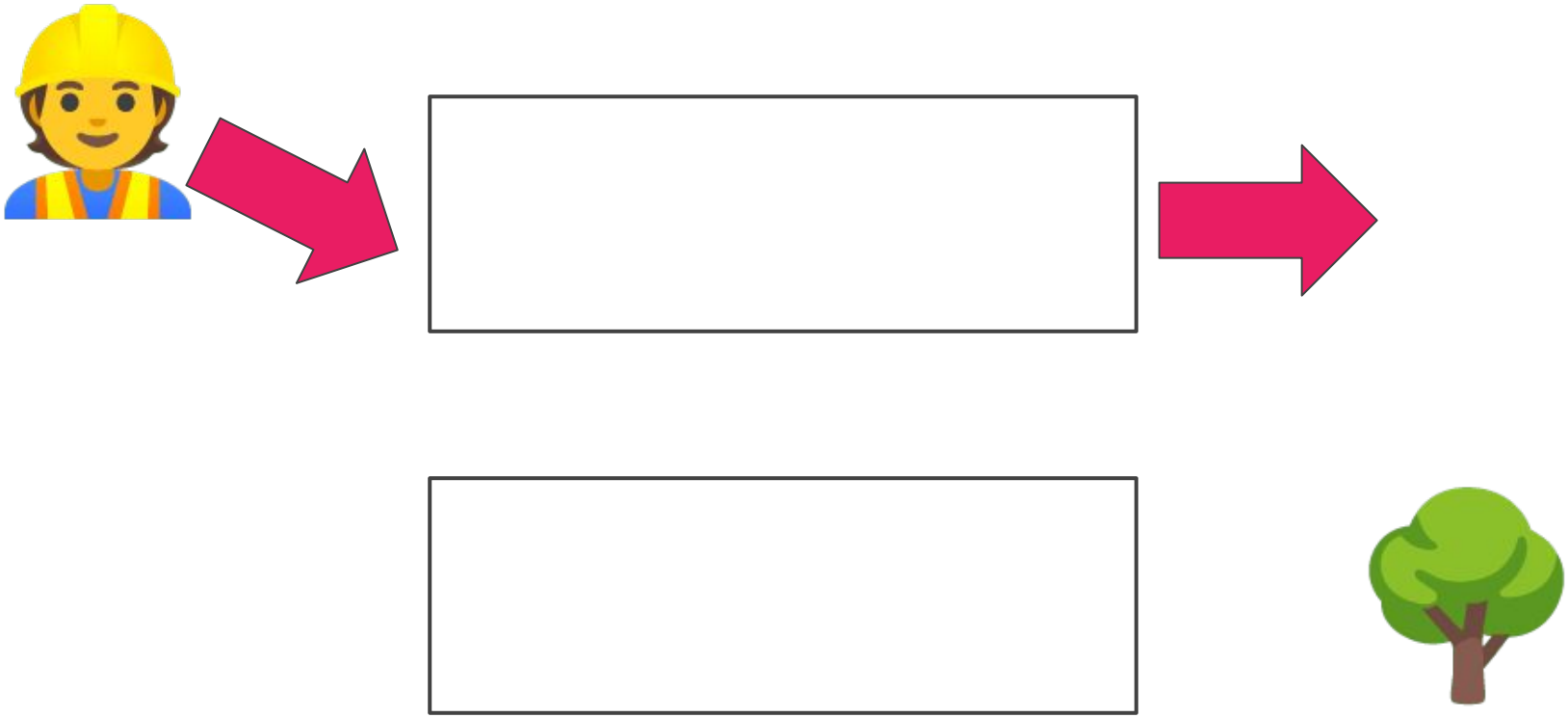
asterisk part 2: simulation



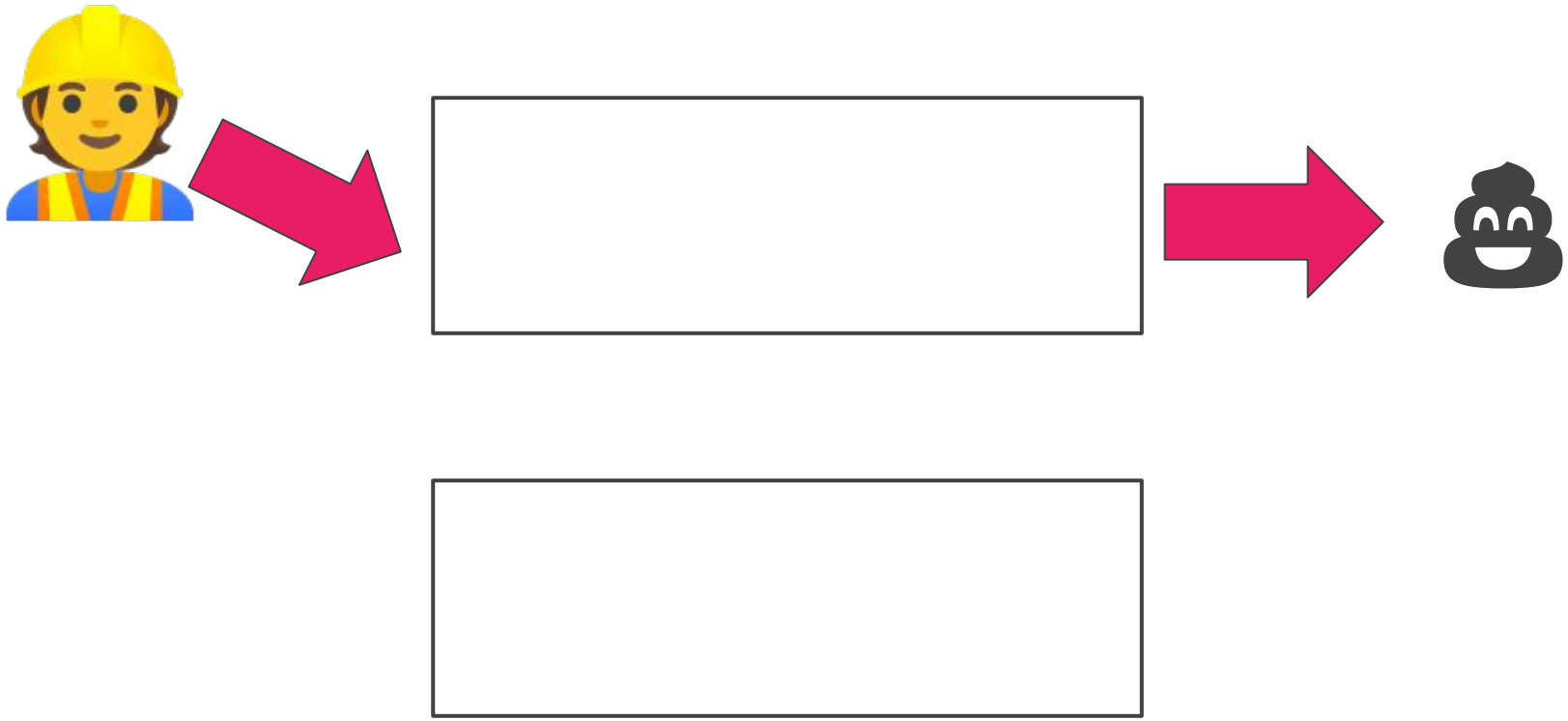
If the builder can’t determine the outcome, they don’t get to decide

Simulation relies on perfect knowledge

Simulation is also **GOOD** tho, actually



In order to get what you want, you pay the builder for it.

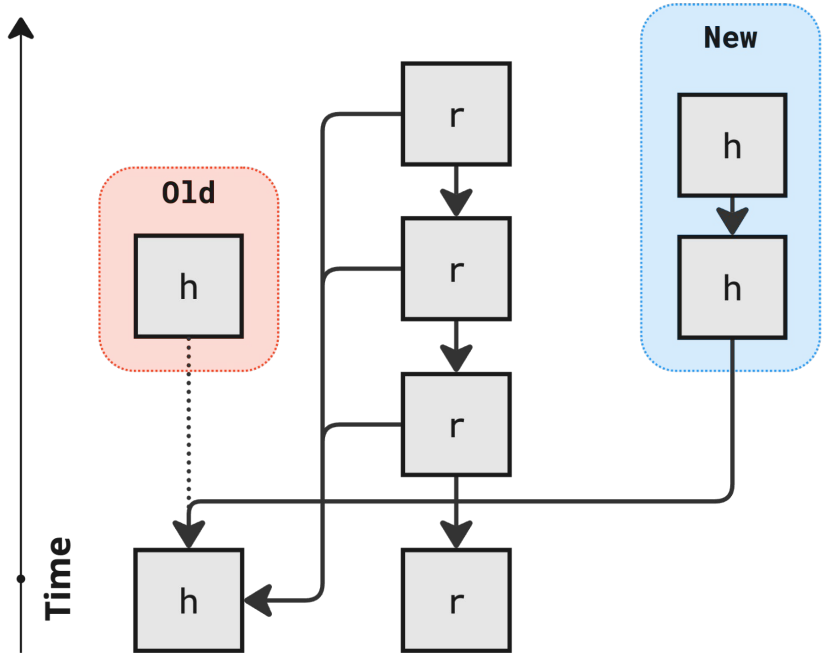


If the builder can’t determine the outcome, you may pay to get rekt

In what cases does a builder **NOT** have perfect knowledge?

referencing new, unpredictable information

Run-ahead, Host Reorg (no invalidation)

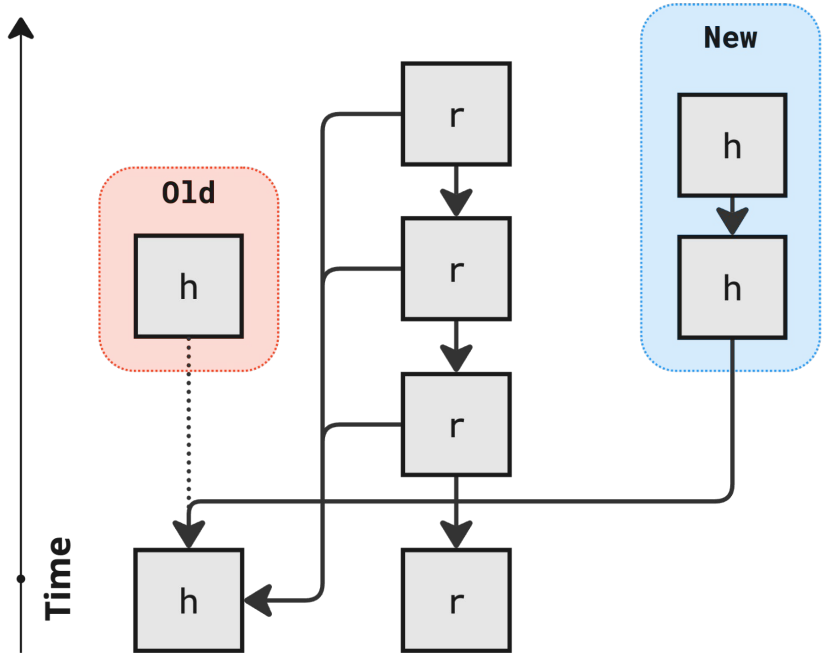


These slides looks very slick because my coworker Tom made the charts. My charts are trash

Run-a…no invalidation)

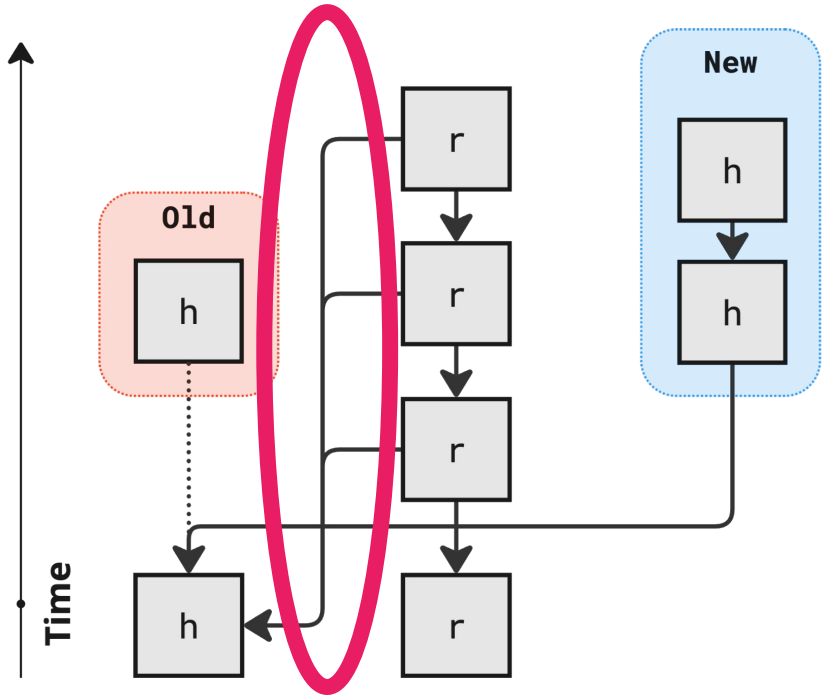
This is Tom

Run-ahead, Host Reorg (no invalidation)



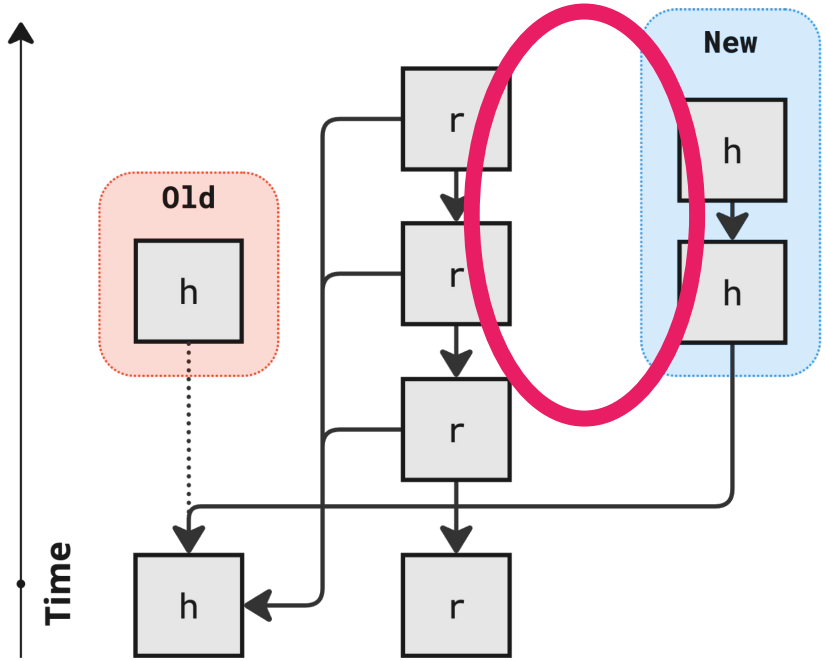
Speaking of unpredictable information. You did not expect Tom.

Run-ahead, Host Reorg (no invalidation)



The sequencer references **PAST** Ethereum blocks

Run-ahead, Host Reorg (no invalidation)



The sequencer cannot reference **CURRENT** Ethereum blocks

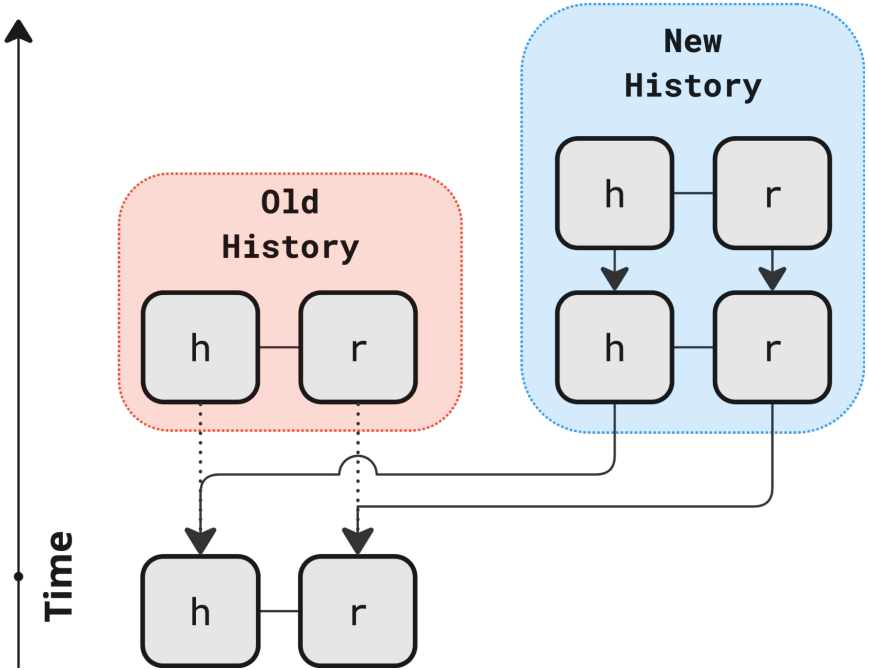
preconfs **REQUIRE** simulation

Optitrum timelags host references in order to preserve rollup simulation to preserve preconfs

Is there a tradeoff?

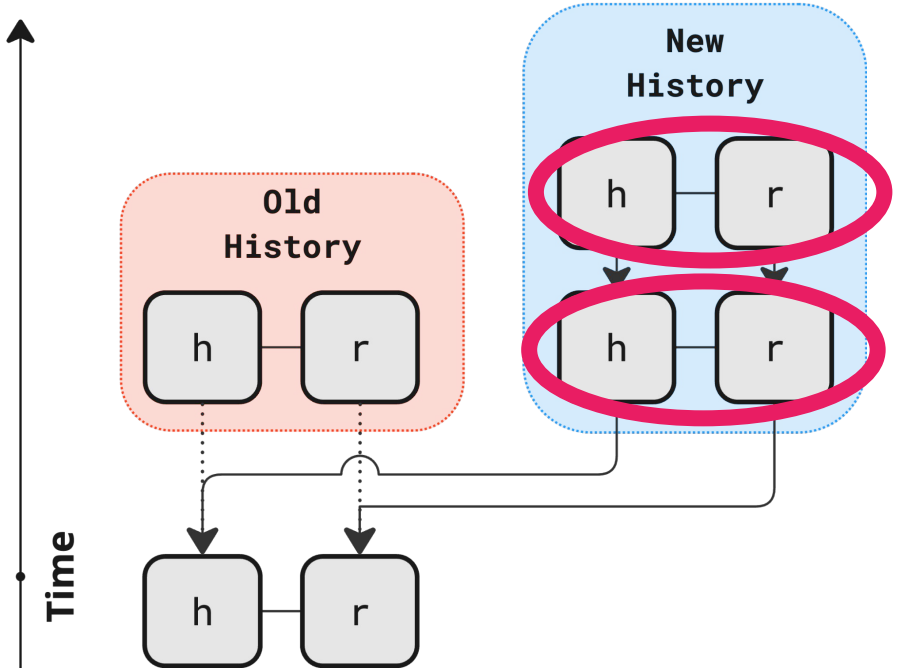
Of course there is!

Based, Host Reorg



This is a based rollup (ty based Tom)

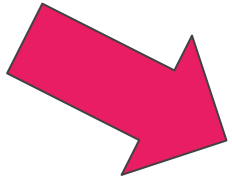
Based, Host Reorg



The rollup builder can reference current blocks *

* if they can simulate things accurately

Allow me to simply copy-paste some slides here thank you very much



f()

f()

Going allllllll the way back to 5 minutes of very fast talking ago

Order invariant

- I have to pay Yaz 1 ETH to defray his aragon court costs.

- I need to add more collateral to my maker vault

- I am transferring an nft pfp because i live like it is 2021 and this is still cool

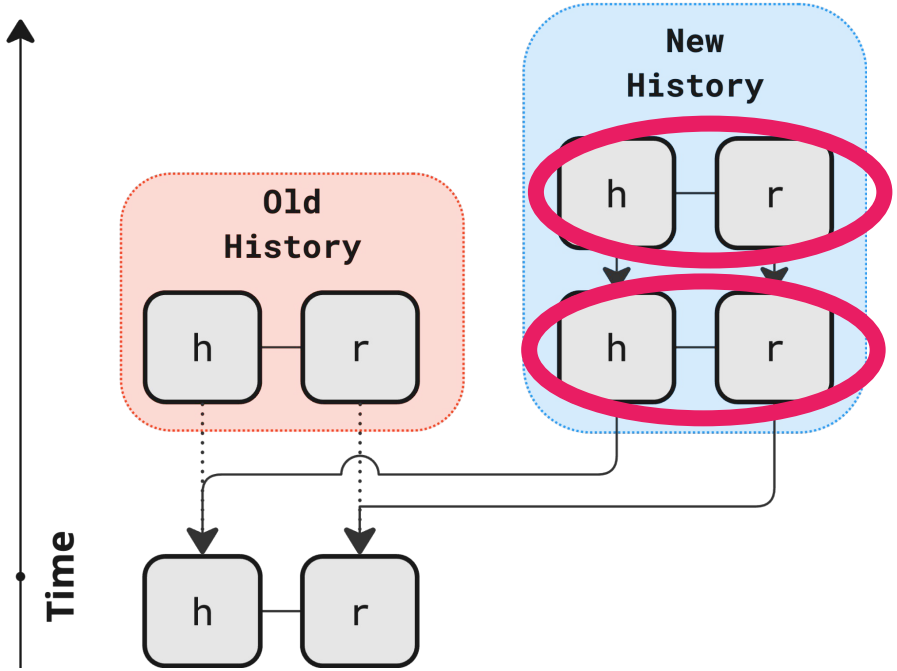
h()

Perfect tree
10/10 no notes

h()

Perfect tree
10/10 no notes

Based, Host Reorg



The rollup builder can reference current blocks *

When the rollup block is submitted atomically with order-invariant actions, the rollup state transition function can reference the output state of those actions and remain simulatable even tho the full host state is unknown

What did we learn here?

James had too much coffee today.

Simulation allows Censorship

Simulation is necessary for preconfs

Preconfs require delayed communication

Simulation is necessary for same-block host->rollup communication

Preconfs are not compatible with same-block communication

we are walking a mechanism tightrope

and every rollup has already fallen off :(

The end

Takeaways:

- TODO:
  give the talk a point

James Prestwich

engineering @ init4
+ Razzlekhan Cosplayer
+ hitting the gym at 4:30 AM
  because of jet lag

Twitter: @_prestwich
Github: prestwich

James Prestwich

init4.technology

engineering
+ Razzlekhan Cosplayer
+ hitting the gym at 4AM 💪
  (because of the jet lag)

Twitter: @_prestwich
Github: prestwich