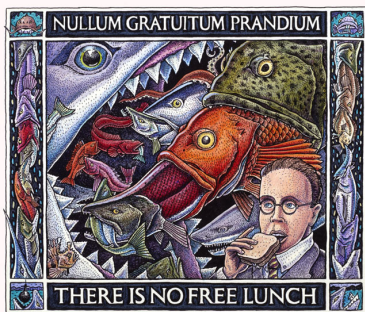


## No free lunch – an inclusion list design



mike neuder – ethereum foundation  
cce day 3 – december 8, 2023

# Outline

- Inclusion list design
- Free DA Problem
- No free lunch design
- Deniability



# Inclusion list design

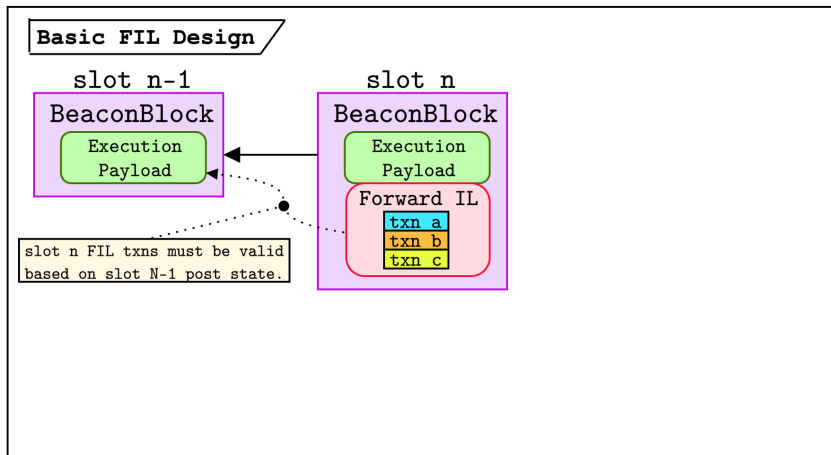
## Basic FIL Design

slot n-1

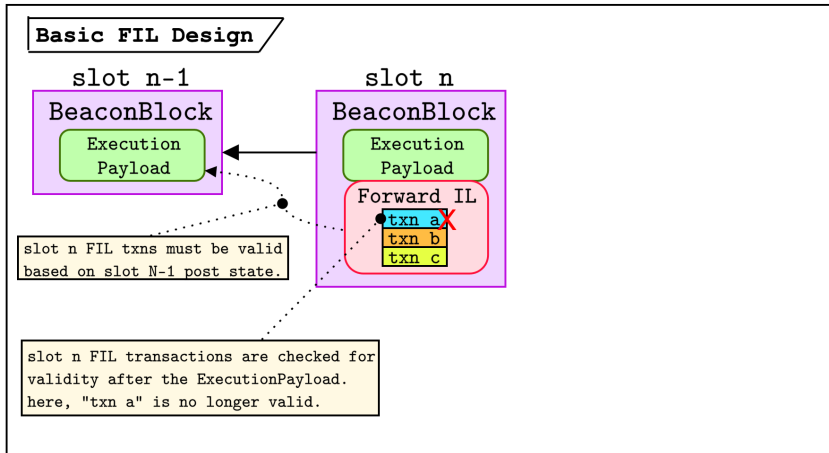
BeaconBlock

Execution  
Payload

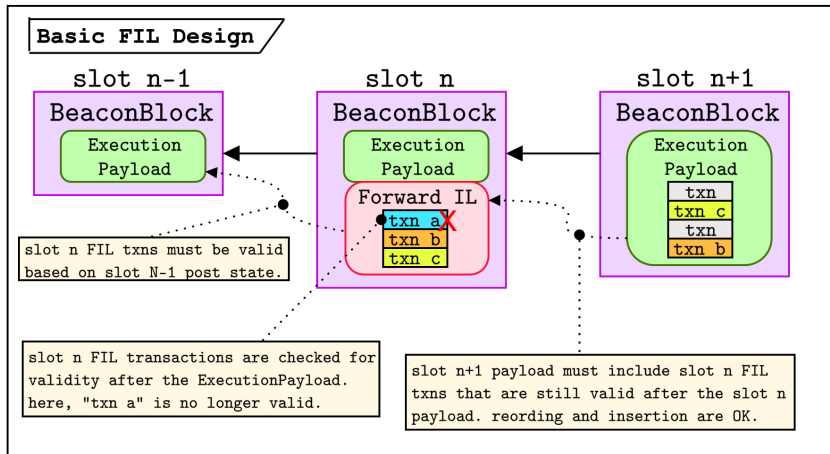
# Inclusion list design



# Inclusion list design



# Inclusion list design



# Free DA Problem

## Free DA problem

BeaconBlock

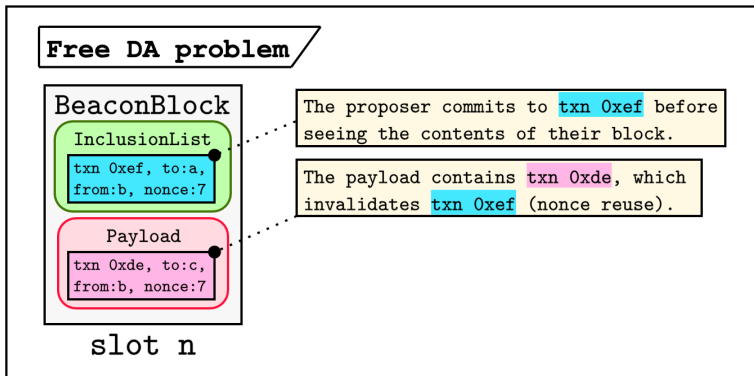
InclusionList

txn 0xef, to:a,  
from:b, nonce:7

The proposer commits to txn 0xef before seeing the contents of their block.

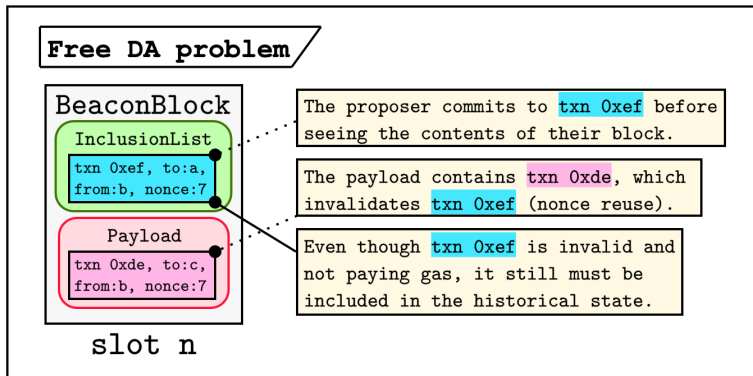
slot n

# Free DA Problem





# Free DA Problem

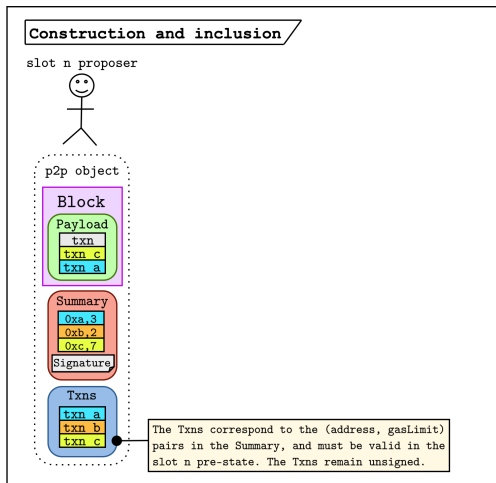


# No free lunch design (1/2)

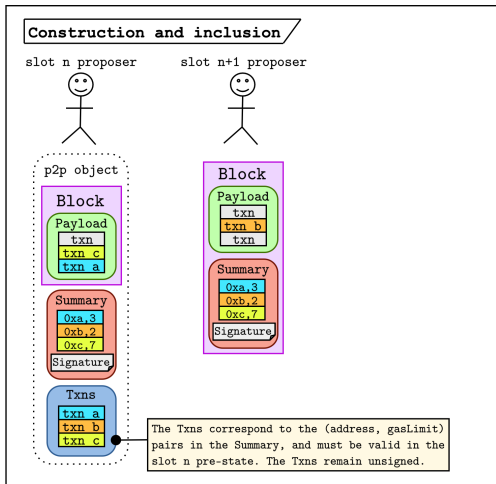
**(Observation 2)** *A transaction that is valid in the slot  $n$  pre-state will be invalid in the slot  $n$  post-state if*

- *the slot  $n$  payload includes at least one transaction from the same address (nonce reuse) or*
- *the `maxFeePerGas` is less than the base fee of the subsequent block.*

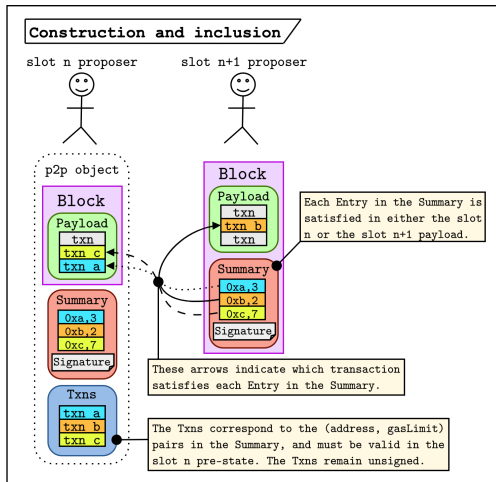
# No free lunch design (2/2)



# No free lunch design (2/2)



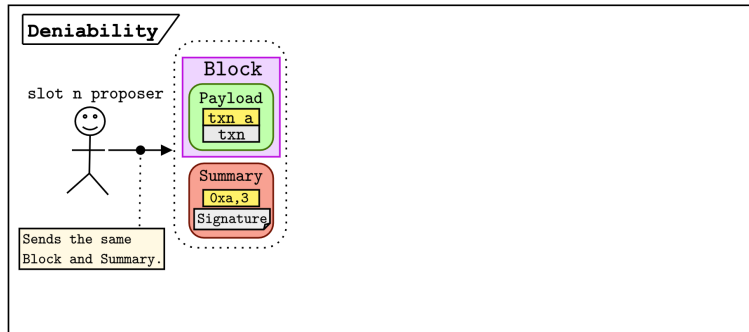
# No free lunch design (2/2)



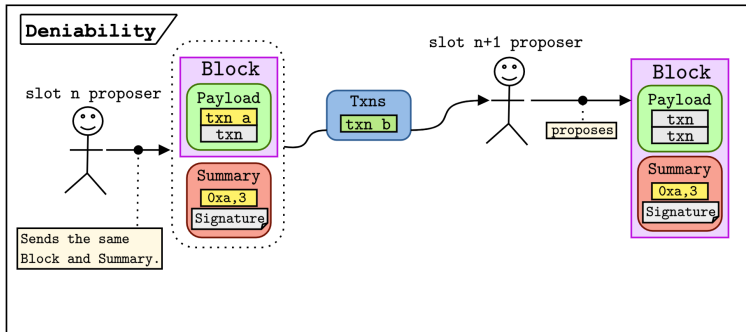
# Deniability (1/2)

**(Observation 4)** *If txn  $b$  aims to achieve free DA, then there exists a txn  $a$  such that txn  $a$  satisfies the same Entry in the Summary as txn  $b$ . Thus validators can safely deny having seen txn  $b$ , because they can claim to have seen txn  $a$  instead.*

# Deniability (2/2)

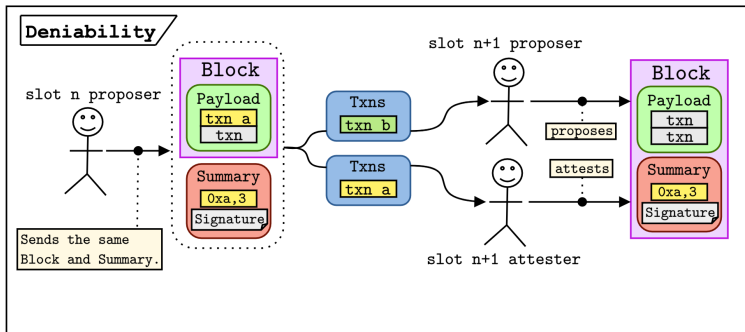


## Deniability (2/2)

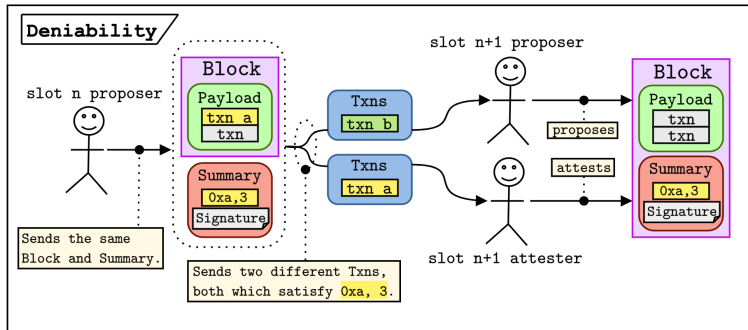




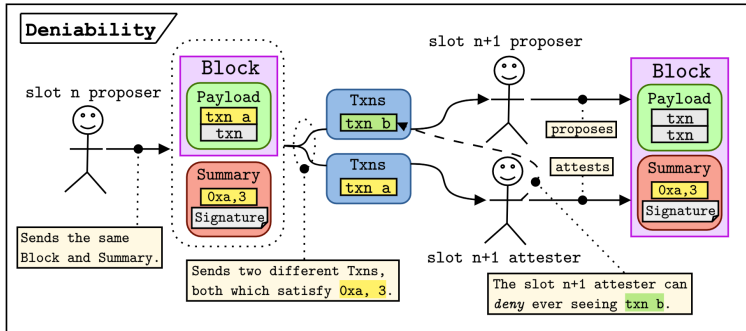
## Deniability (2/2)



## Deniability (2/2)

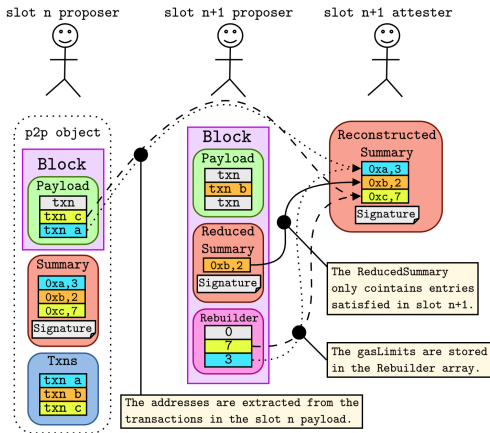


## Deniability (2/2)



# Extra credit

## Summary reconstruction



thanks! :-)

