TL;DR

- If there is a way to finalise state execution on each shard, we could use proofs of custody to ensure collators check the availability for a period of wind-back shards. Since cross-shard transactions are likely to require finalised state execution, it may be worth revisiting proofs of custody. The basic idea is to push availability checking to the state execution level as well.

Proofs of custody may be used even without finalised state execution-- so long as the slashing penalty is not too severe and collators lock funds on the appropriate shard chains – or alternatively we can simply use snarks/starks for fraud proofs.

A reminder of proof of custody schemes, similar toJustin's, but with extra padding:

1. When assigned to a shard, each collator commits to the SMC the hash of a random secret s

.

1. Before adding a new header to the SMC, the collator computes \text{hash}[b||s]

for the last 25 blocks b

, and includes the block numbers in the header. This is the challenge.

1. The collator reveals s, through a transaction to the SMC, which enables people to verify the challenge.

Slashing Conditions:

If we discover a block for which \text{hash}[b||s]

disagrees with the value in the collators header, we slash the collator. The question is how (I assume no starks for fraud proofs for security assumptions, although I think starks would be a great solution

, since they could just be used for fraud proofs which there would be few of, given the incentive structure):

With no execution-finalisation

, the collator posts a bond of X eth to the shard chain he is creating a header for, which he cannot spend for 50 blocks. In the event of an incorrect header, the executors adjust the collator's balance to 0.

To preserve transaction ordering: each proposer can hash his proposals in the order given. If this hash does not match up with the ordering given by the collator, the collator gets slashed as above.

If any node discovers s

before the block header is added to the SMC, they can submit s

to the SMC to steal the collator's deposit. This discourages outsourcing availability checking.

If there were a way of finalising state execution, then collators could also be slashed at the SMC level for a higher deposit as soon as a fraud proof was generated (merkle state root with a special symbol for fraud F stored under the validator in a finalised block). The question is how finalisation occurs.

This scheme may create perverse incentives with executors blackmailing collators, but if we take the truebit philosophy seriously and 10% of executors are honest, and false claims are punished, then it would be possible to push data availaibility checking to the execution layer.