following the thread for my proposal, and regarding a matter outlined in this pull request:ethereum/execution-apis#455 (comment) we are confident that our state provider with zero-knowledge proofs (zkp) can address the problem:

Our state provider incorporates a trustless RPC query system via the Helios p2p network. Additionally, it offers the provisioning of a ZK circuit (referred to as Axiom) and a p2p network to disseminate the zero-knowledge proof of the final state prior to execution.

ZK proofs of the last state can indeed help address the challenge mentioned. ZK proofs, specifically ZK-SNARKs, have the potential to provide proofs of complex computations and statements without revealing any sensitive data. Here's how we can apply to the scenario:

Challenge: Computing New Block State Root and State Changes Audit:

In Ethereum, auditing state changes that result from executing transactions is crucial for maintaining transparency and trust. However, as pointed out, state deletions can complicate matters, making it difficult to recreate the entire post-block trie structure and verify state changes against the state root in the block header.

ZK Proofs of Last State and State Changes:

Using ZK proofs, we can generate cryptographic evidence that certain computations were performed correctly without revealing the actual data involved. In the context of state changes in a blockchain:

1. Generating ZK Proofs for State Changes:
2. Before the block's execution, the transactions' state changes (including deletions) can be computed.
3. ZK-SNARKs can be employed to generate proof that these state changes were correctly computed without disclosing the specific details of the changes.
4. Proofs Auditing and Verification:
5. The ZK proofs generated can be included in the block or otherwise made available to the network participants.
6. Nodes in the network, including light clients, can independently verify these proofs against the post-block state root in the block header.
7. Efficiency and Privacy:
8. ZK-SNARKs allow for succinct proofs, which means that verifying the correctness of the state changes can be done with a compact proof size.
9. The private nature of ZK-SNARKs ensures that sensitive data, such as specific deleted items, is not revealed during the verification process.
10. Decentralization and Trust:
11. Because nodes can verify the correctness of the state changes without needing to recreate the entire trie structure, the challenge of missing internal nodes due to deletions is mitigated.
12. This approach enhances the trust and transparency of the blockchain while avoiding the need for all nodes to store complete trie data.
13. Auditing Historical Data:
14. Historical blocks' state changes can be audited and verified using the corresponding ZK proofs, allowing for comprehensive state change verification.