Currently, the Ethereum API only gives access to the last 256 blocks. This makes it hard and expensive for contracts to verify that something did indeed happen a long time in the past, though it isn't impossible, as demonstrated by Andrew Miller's and Kobi Gurkan's https://github.com/amiller/ethereum-blockhashes

My proposal for a future hard fork: add a function that provides logarithmic access to the entire blockchain history, whereby each block of number N offers direct access to the hash of the \lceil log_2 N\rceil

blocks of number (N-1) \& ~(-1 << k)

for k

from 0

to \lceil log_2 N\rceil-1

.

Then contracts can cheaply verify the presence of a transaction or log event in logarithmic time and space for both clients and servers. Validators don't need to maintain more than a logarithmic extra state, though full history servers may have to maintain O(N log N)

extra bits (or only O(N)

using some compression at the cost of recomputing O(log N)

hashes on demand).

PS: Shouldn't there be a topic for discussion of the contract API? Or is the API considered fixed in stone forever, even for backward-compatible extensions?