Currently setting up a new node takes quite a bit of time and requires a fair bit of data to be transferred. At 0xEb8B006Ee7ab4aB384C15DD8ceBaba9663B1e404 I stored JSZip.min.js, and you can access the ABI in the page source here (and test it out too!) or onGithub. For the file I used to generate this contract seehere.

By compressing data using an algorithm set at specific contract address (whether that be the algorithm I uploaded or a different one agreed upon later) and using hashes to verify a compressed representation of the chain, I think it would be possible for clients to transfer compressed data, and everyone would be able have access to immutable and standardized compression and decompression algorithms (since different implementations of zipping compress more or less efficiently, which would generate different hashes). This should speed up sync, and could even be used to compress older headers and data for a smaller disk size footprint. Since this is stored on chain, it could even work across different clients as well to ensure consensus is maintained. Essentially, the entire chain besides the contract containing the compression algorithm could be compressed to allow for smaller disc sizes and quicker transfers.

I tried digging through the github for Geth to see if anything like this was already implemented, but I'm unfamiliar with Go, so I may have missed it and it may be the case something like this is already implemented.