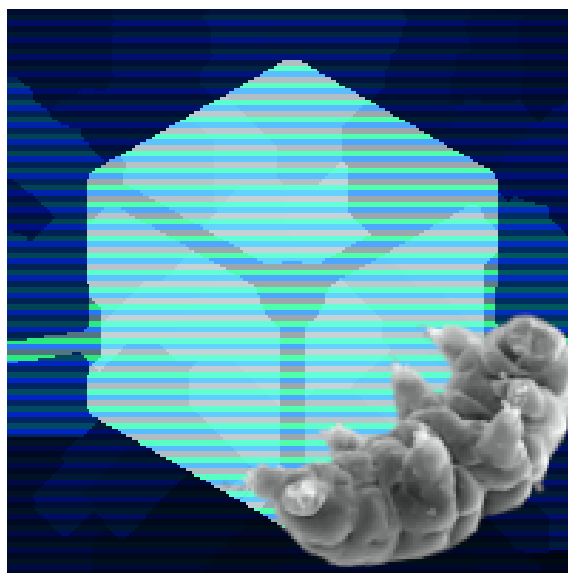# July 24 Update: Idealism and Pragmatism

Taylor Hulsmans

July 2022

Abstract

Another good week developing the pure vision hidden inside CAW's manifesto for a decentralized social net. Mostly boilerplate work for this one, going so far as to aid in the exposure of a bug in a recent ipfsjs upgrade to ESM modules. Perhaps my vite-react repo will aid in serving future devs in bootstraping a react typescript environment with tools like vite that are ipfsjs ready in the future.

0.1 Pragmatic Efficiency

At its heart, Tardigrade utilizes specialized forms of state channels to achieve efficiency. What makes the project innovative is that it does not try to provide liveness with cryptoeconomic miner/watcher incentives, security with fraud proofs or finality with challenge windows. Instead, the beneficiaries of a message chain has an incentive to pin on IPFS and distribute it over pubsub- ergo liveness

is now user driven. Security is proxied for reputation derived from the social graph by which the state channels receive initial state, it too is now user driven. Transaction finality is not given first class citizenship, as long as a signature chain is processed before the deadline, state update is attempted- Tardigrade just does the best it can. In short, Tardigrade doesn't try to be the secure and efficient scaling solution, it tries to be the perfect insecure and efficient scaling solution. Tardigrade isn't for sending millions trustlessly to a user across the world, its for sending cents inconsequentially to many people. So what if the homeless person skims the cafe tip jar, so what if one individual decides not to chip into the honour box for vegetables, so what if a patron forgets their bar tab one night? Hopefully, as reputation grows those cents can turn into dollars

0.2 Front end updates

IPFS has been in the process of upgrading their code base to Ecma script modules. When coupled with typescript and the lack of basic node packages in browsers, the vertical slice has gotten a bit, well, jagged. As such, I've opted to reduce the depth of the techstack and build directly into the react front end to begin with. A Tardigrade.js package can be teased out after it is built on the dashboard