

For internalising the costs of storage, [Gas Token](#) is a method of doing that by arbitraging gas prices, filling up the state with SSTORE with junk data (which obviously has disadvantages) when gas prices are low, and emptying the state with SLOAD when gas prices are high. Note that internalising costs are necessary for sustainability, however doing so can have adverse side effects. For instance in the case of internalising storage costs it increases the complexity for users and developers, or complexity can somehow be reduced with intermediaries that provide some improvements, e.g. storage providers that have a token, but in turn that may not be trustless, so you'd need ways to deal with that, e.g. slashing conditions. Nevertheless, these costs should be internalised, otherwise the protocol will suffer.

Also note that there are gas costs for storing data and a refund for clearing it, as defined in the [Yellow Paper](#):

[

Screenshot%20from%202018-03-20%2009-23-52

707×128 17.4 KB

](https://ethresear.ch/uploads/default/original/1X/9734b2ad90c18b132d37279458a9c28a3dad497c.png)

[

Screenshot%20from%202018-03-20%2009-32-49

796×95 28 KB

](https://ethresear.ch/uploads/default/original/1X/6df6785b1f7784c7bc094b1911296be54b7f0a9f.png)

With optional metering in eWASM these gas costs could be more cost-reflective. These gas fees and refunds internalise computation and I think also I/O or disk reads and writes.

It would also be good to consider how to internalise the cost of bandwidth, although I don't know how.

Internalising these costs could also somehow incentivise full nodes (which validate/execute transactions), which is also discussed here: [Incentives for running full Ethereum nodes](#).