

This is a simple proposed scheme for incentivizing the development of software libraries for ethereum (this can be implemented in an eth1 or eth2 context; the scheme is fairly agnostic). Example “ideal beneficiaries” of this include authors of:

- Wallet contracts
- Standard libraries
- [Weierstrudel](#) and other optimized implementations of algorithms

Both schemes assume [EIP 1559](#) is included, so there is a “burned fee” associated with every unit of gas consumed; this is important to prevent cheating by miners.

## Scheme 1

When a contract is published, the author of the contract (ie. the account that published it) is stored. Every time a contract call finishes, X% (eg. 33%) of the burned fee associated with the gas executed inside the callee is transferred to the creator of the callee.

## Scheme 2

Let  $T$   
be the previous total amount of ETH fees burned from gas spent executing a given contract with byte length  $L$   
, and  $N$   
be the burned fee associated with the gas spent executing a contract call to that contract. When that call finishes, transfer  $L * (f(\frac{T + N}{L}) - f(\frac{T}{L}))$

ETH to the author (notice that this is just  $f(T + N) - f(T)$

but stretched horizontally and vertically by a factor of  $L$

). In general,  $f$

should be a function with the following properties:

- $f(0) = 0$
- $\lim_{x \rightarrow \infty} f'(x) = \frac{1}{2}$
- $f'' \geq 0$

The idea is that we have a superlinear rebate, which starts off at zero, then grows over time and eventually approaches 50% of the burned fee. The inclusion of  $L$

in the calculation is there to “stretch out” the function for larger contracts, making the point at which the rebate approaches 50% proportional to the contract size; this prevents manipulation via combining or splitting contracts. One simple candidate for  $f$

is  $f(x) = \frac{x^2}{4M}$

for  $x < M$

and  $\frac{M}{4} + \frac{(x-M)^2}{2}$

for  $x \geq M$

:

The goal of the superlinear rebate is to add a check against copying a contract and replacing the developer’s rewards with your own. Such a tactic would not be profitable unless you can get many users using your contract.

## Notes

Note that Scheme 1 is, from a purely economic “tax/subsidy incidence” analysis, a no-op: developers could theoretically undercut each other by providing some portion of their rebates back to users through some extra-protocol mechanism, and the Bertrand competition equilibrium of this is zero returns to the developer. However, the goal is that this would be hard to do, and not worth it in the bulk of cases because the per-transaction fees involved are tiny, and the hope is that if the community agrees that such competition is harmful then there would not be effective infrastructure built to support it.

A long run sustainable fee level on ethereum today is [about 500 ETH/day](#). If 80% of total fees get burned through EIP 1559, and 70% of that is calling contracts ([gas used per day](#) - (21000 - 6800) \* [transactions per day](#) - 68 \* [blocks per day](#) \* [average block size](#) as a fraction of total gas used per day roughly gives this, and it's likely a low estimate due to zero bytes in blocks), and on average the rebate is 25% then this gives 70 ETH/day (\$14000 per day or \$5 million per year) to contract authors, at present fee levels

.