If a transaction without signature (all zeroes) is sent to a contract, the gas costs will be paid by the contract, but it is allowed to throw before some gas limit (e.g. 200 000) is reached without having the spent gas subtracted from its balance. Otherwise, if it does not throw before the gas limit is reached, it pays for all the gas used. It is up to the contract to verify eligibility, to collect tx fee within its own accounting, to protect against replays and so on. If they fail to do so, it is the contract's balance that will be depleted, miners/validators won't suffer any harm.

Thus, validators will always get paid for all the computational work they perform, except for a limited verification cost (O(1)

per transaction), but that is fundamentally no different from how it works today: signature verification is potentially "unpaid work" for validators, but it is bounded at O(1)

.

The amount of changes required for EVM to support this feature are minimal: no new opcodes, no special preambles.

One way to smoothly introduce this and avoid arbitrary magic numbers baked into the spec is to set this limit initially to zero and allow it to be modified similarly to block gas limit by validator voting. The introducing HF will have an initial target set to 200 000, which they will probably quickly reach if there is no strong opposition from the community.

Even this very simple first step will allow for anonimized tokens and a host of other interesting use cases currently limited by the requirement of tx senders having Ether on the same account.