I believe this is extremely important, as the proposed pay-to-stay scheme may disincentivize many current use cases of Ethereum, and instead relegate it to being a ledger for traditional businesses to run their traditional centralized apps on. Note, I do support this latter use case, but not exclusively, and my fear is that pay-to-stay would make all other uses extremely hard to implement.

Since I feel like I'm polluting thepay-to-stay pricing thread with this, I'm creating this topic for this discussion specifically.

As @vbuterin pointed out in that thread, community-owned registry contracts built on address -> {something} mappings (nb. the most common design pattern on today's Ethereum!) will be phased out in favor of single-owner, single-point-of-interest contracts.

This is an entirely new way of designing tokens and non-fungible assets, and I think deserves a thread to discuss ideas. A post pay-to-stay Ethereum would look a lot more like Bitcoin in my opinion, with UTXO-like contract structures, validated through provenance.

However, there are some use cases when mappings and shared-ownership registries still cannot be done away with… Like ENS, for example, users need a central address to use for querying based on name.

These contracts need to collect tax from registrants to pay for rent, otherwise new registrants will have to subsidize previous registrants, and as the state size grows, the cost of new registrations will grow proportionately (eventually becoming unfair or even untenable). However, to force old registrants to keep paying tax, their part of the state needs to be ejectable - for this records about tax payments needs to be kept, and non-payers need to be identifiable in O(log(n)) time - something I'm not sure is even possible in the EVM.