

Abstract

Taking a look back on 90s and beyond to pull a parallels between current state of Ethereum and bare-metal programming in Assembly and C back in a day, as well as today in embedded world.

The goal of this is to take the learnings and solve some problems we are seeing in Application layer security, such as highly complex vulnerability management procedures, difficulties to pre-arrange effective emergency response automations, and not least - very high level of stagnation caused by ERC standards track which just does not allows easily to keep dApps and contracts with the progress we need there.

Referring to the fact that Ethereum Virtual Machine its still only a Machine, Im trying to reason that in traditional programming and computer architecture many security issues of what we see today on application-layer on Ethereum smart contracts are effectively solved and we can learn from there "how".

My analogy is based on following illustration:

Think of an IoT enabled house. If there is an asset behind the door, which is controlled trough WiFi by chip in a computer, and we can declare, that the one who is able to open the doors is owning an asset. If we also understood section above that any peripheral is just an interface that can be accessed by calling right address on address bus, then we can map this equation out to the fact that from CPU standpoint, an asset ownership is equal to ability to set or unset a byte in some registry on some address of IO bus.

Based on this I'm making conclusions that

- a) Assets are not interfaces
- b) Secure architecture requires split between asset storage and asset driving interfaces
- c) Current asset model is not user centric and is causing not just stagnation and poor ux, but also centralisation risks
- d) The model of EVM seems correct for me, it can be competitive with what Move has to offer, but it must build a wrapper to abstract us away from "bare-evm-programming".
- e) ERC versioning is something we need to consider

Full article

peersky.xyz

[Fixing Ethereum UX with user-centric asset model](#)

How do we secure growth for ethereum for the next decades by designing secure, user centric architecture for assets and applications

RFC

This is request for comments. Im am looking for community feedback to incorporate this further.