Neutron: Modular Rolling-Governed Applications System

Internetbase.org - Anvil - Vesselin Tsukeff
ICDevs.org - Origyn.ch - Austin Fatheree

Abstract. Rolling governance will allow a web3 application to upgrade itself independently, without the need to rely on other services except the ones provided by the Internet Computer Protocol[1]. DAOs[2] provide enormous benefits when governing applications and services, but also have downsides. DAO-governed registries and services[3] that are connecting projects inside the ecosystem, but create for-profit gates that sometimes enforce premature standards that hinder innovation, and unnecessarily take control from users and ecosystem developers. In such a setting developers need to accept the architecture and standards imposed on them, seek permission from other organizations, and restrict themselves to what is technically allowed while joining a sub-ecosystem that is steering the wheel in an unknown direction.

The Internet Computer provides us with an open-to-everyone platform where everyone can deploy their own software, and create their own ledgers, architectures, standards, and DAOs. It's truly a world computer. But the freedom it provides is also holding us back when it comes to creating a network effect. We are proposing a solution that will improve the networking effect in a more decentralized and open manner that will not restrict the freedom of users and ecosystem developers. A solution that befits the Internet Computer.

The Neutron is a user-deployed canister[5]/smart contract. It's exclusively under the proprietor's control and is strategically designed to leverage the progressive advantages of the 'Snowball effect'. It has a web interface (frontend) with cryptographically secured contents[5], backend functions, and heartbeat[7]. This paradigm can only be accomplished securely on the Internet Computer. Its web interface allows users to upgrade it by choosing applications from an easy-to-use interface, or by adding them from an external URL. All applications are fetched assembled, and compiled inside it, then the new bigger Neutron is installed. The Neutron only relies on itself to upgrade and evolve without restriction. The applications it assembles can add functions to its backend and interfaces to its frontend. Among other ways, this can be achieved using Motoko[4], it is lightweight and allows frontends to compile the applications into one canister, while also providing a certain degree of module isolation.

This rolling governance will allow developers to create applications that allow users to transform and grow their Neutron into anything they like. These applications can add functionality such as an on-chain wallet, a multi-sig wallet, a DAO, peer-to-peer discovery, two-factor authentication/signing, trading bots, identity providers, file storage, and app stores. It starts with enough to facilitate the mechanism allowing self-upgrades and what happens after that is up to the Neutron owner and community of developers that offer applications. This will allow all projects to provide applications that integrate their unique

innovative services & applications. It will allow them to collaborate at a synchronous[1] level for the first time. Such a wallet can hold all possible crypto assets from any standards and any chains with Threshold ECDSA[9]. It can provide an identity[11], similar to Internet Identity[12]. It can store private data with VETKeys[13]. It will be able to do arbitrary computation on encrypted data[14]

Until now we have only had user canisters that aren't governed by the users themselves, but a DAO[10] (OpenChat, Hot Or Not, Ego). They require the DAO to approve any changes, thus increasing their DAO's value. Instead of being a restrictive gate, DAOs will be able to put a green checkmark that conveys their approval of a certain app to the end user. The Neutron paradigm allows us for the first time to allow users to govern their canisters, similar to how they already govern their personal computers - by adding and removing applications. Users are not all voting for a proposal that installs an application on all PCs globally. The Neutron paradigm allows the emergence of user-governed cryptographically secured operating systems.

The Neutron system simplifies compliance with regulations such as GDPR, HIPAA, and others by granting users ownership and control over their data. Vendors can deliver approved code to user canisters for computation, ensuring transparency and traceability. Applications, in this setting, hold only pointers to data, which significantly reduces data duplication and potential breaches. Furthermore, user ownership of data means they can delete their information at will, enhancing privacy and supporting regulatory requirements like 'the right to be forgotten' under GDPR. This decentralized data governance model provides a compliance-friendly framework for developers while preserving user privacy and control.

The open-source, algorithmic nature of the Neutron system allows vendors to run data summaries privately without exposing user data. Vetted and open-source roll-up canisters can distribute code to user data wallets, processing data in a way that maintains user privacy while still allowing vendors to extract valuable insights. These canisters execute computations directly on user data without moving or revealing the data itself, effectively implementing a "map-reduce" operation that enhances vendors' service offerings. The results are anonymized or aggregated to ensure no individual user's data can be identified. This maintains user privacy while enabling vendors to utilize the data for improving their services.

In the context of a client dApp retrieving data for multiple users, like a social graph where posts are held in user-owned canisters, the parallel nature of the Internet Computer's architecture becomes even more beneficial.

From the dApp's perspective, when a data retrieval request is made, it is broadcasted to all relevant user canisters simultaneously. Each canister independently processes the request and returns the relevant data (e.g., user posts in this case).

Since the retrieval operations are performed in parallel across the network, the total time to gather all necessary data is significantly reduced compared to sequential retrieval. This makes the process efficient, fast, and scalable, allowing the dApp to serve complex, multi-user requests effectively.

By leveraging the inherent parallelism of the Internet Computer, client dApps can optimally utilize resources, maintain swift performance, and ensure a responsive user experience, even with a large number of users or a high volume of data.

From an economic perspective, this leaves projects to decide how much of the generated value, they want to share with others inside the user-owned Neutron and how much they want to keep inside their own canisters. If they share too much - they will not be able to capture enough value to raise funding to develop their ideas. If they share too little or nothing they won't be able to effectively collaborate with other projects in the ecosystem.

References:

[1] Internet Computer Protocol https://internetcomputer.org/whitepaper.pdf
[2] Decentralized Autonomous Organizations https://internetcomputer.org/sns
[3] Such as DAB https://github.com/Psychedelic/dab and CAP https://github.com/Psychedelic/cap
[4] Motoko https://internetcomputer.org/docs/current/motoko/main/motoko
[5] Canister https://internetcomputer.org/docs/current/concepts/canisters-code
[6] https://internetcomputer.org/how-it-works/smart-contracts-serve-the-web/
[7] https://internetcomputer.org/docs/current/motoko/main/timers
[8] https://internetcomputer.org/docs/current/motoko/main/actors-async
[9] https://internetcomputer.org/how-it-works/chain-key-technology
[10] OpenChat https://oc.app/whitepaper
[10] Hot Or not
https://hotornot.notion.site/hotornot/Hot-or-Not-Whitepaper-c539179e51f44867979f4372e4635f59
[10] Ego https://forum.dfinity.org/t/proposal-of-personal-canister-framework-on-ic-ego/17899
[11] https://internetcomputer.org/docs/current/references/ic-interface-spec#canister-signatures
[12] https://internetcomputer.org/internet-identity
[13] https://internetcomputer.org/docs/current/blog/features/vetkey-primer
[14] https://www.zama.ai/