Twitter Thread by Robert Sasu

Robert Sasu
@SasuRobert

Thanks to the growing number of people who @getMaiar & engage with the @ElrondNetwork ecosystem, the number of accounts doubled to over 320k & we're over 2.5M TXs, an unprecedented adoption level for a 6 months-young mainnet.
#hypergrowth100 Day 40
Weekly #elrondtech ■■



1/ The Delegation Manager toolset has been released for testnet integrations, ahead of our imminent mainnet upgrade. It enables staking providers to set up & manage staking pools, nodes, fees & other parameters for their customers.
https://t.co/FTVIwYWKMh

2/ We updated the libp2p libraries to the latest release & found out that they work seamlessly with their old counterparts. Congrats @libp2p team for the effort put in backward compatibility. We know how complex that is. :handshake:
https://t.co/29I6YRtjud

3/ Redesigned the async call & token transfer syntax in our Arwen WASM VM. Interactions between contracts are now “return types”, which makes execution flow more intuitive. Callbacks have also been redesigned.
https://t.co/wh6BIBeDUd

4/ Arwen WASM VM updates continued:
- Integrated new Arwen and new features of Arwen into elrond-go. Added a set of integration tests for different scenarios.
- Fixed a bug in Arwen VM which did not let one contract to deploy multiple contracts.

5/ - Added unit and integration tests with a smart contract factory example.
- Testing the new Arwen async-calls & promises
- Arwen & elrond-wasm testing of various scenarios involving async calls and ESDT token transfers.

6/ - Made lots of fixes for Arwen's new multi-asyn call infrastructure (Promises).
- Continuous integration improvements - run tests on pull request (arwen-wasm-vm - added; elrond-wasm-rs - in progress)

7/ - Added new features for transferring value or transferring ESDT & calling other (multiple) smart contracts without the asynchronous call & callback functionality. This enables one contract to send & call functionalities from multiple contracts from the same execution

8/ - Improved the mechanism of sending back value to a contract through an asynchronous callback. This value can be any eGLD or ESDT. The treatment works both in and cross-shard
Learn more about our VM: https://t.co/U6R1xN4gb5

9/ Other notable achievements:
- Worked on the NFT standard for elrond-go. Sharded NFTs where users will keep their one NFTs at their own address. A lot of features here: like creating a brand, creating a marketplace without smart contract (will accept any value - ESDT / eGLD).

10/ - working on a wrapper over the API handler so when the node is syncing after a shuffle out, API endpoints could return messages that signal the node is currently syncing, instead of starting the API webserver only after the sync is finished.

11/ - Testnet release T1.1.28.0 was commissioned that fixes several bugs.
- Updated all other major 3-rd party libraries to their latest versions
- Fixed several tests that had nondeterministic execution paths leading to different code coverage values.

12/ - Finished a trie sync optimization that should decrease the time took by the node to get in sync with the rest of the network.
- Working on an ERC1155 smart contract
- Finalized unit tests for block execution invariant
- Refactored transaction coordinator with arguments

13/ - Tested the delegation manager and delegation SC.
- Ledger Nano S tests after deploying of the latest version.
- Added a response logger middleware on Elrond Proxy so we can have a better overview over failing requests

- Analyzed the runtime memory usage of the Elrond node.

14/ - Added a new endpoint that returns all the key-value pairs stored under a specified address
- Worked on support for Testnet BTC on our fork of Trust Wallet Core (this feature will be used on Maiar).
- Fixed a concurrency issue on Observers, within the VM Querying components

15/ - Minor fixes and extra tests in the wallet utilities of erdjs.
- Sketched a Python prototype for trivial secret splitting (involving one-time pads and basic modular arithmetic) - could be useful for Maiar in the future.
- Working on automatic elrond-sdk docs publishing

16/ - Optimizations on the node names and identities.
- Investigate failed relayed TXs from the backend.
- Refactored API resolvers and adapted the latest changes in economics to the soft restart feature branch.

17/ - Refactoring integrity checks to expose a single CheckIntegrity function
- Create logs parsing tool for bad transactions and used it to identify/debug some user registration failures in Maiar.
- Started to improve overall error logging in elrond-go

18/ - Implemented EIP 2334 javascript module that allows you to derive multiple bls keys from a 24 words mnemonic
- Work on TransactionConverter to replace function references in other components
- Tested different branches to ensure compatibility & protocol overall quality

19/ - Started to refactor different components from elrond-go to ensure clean code quality
- Implemented a disabled node facade for the new API refactoring & the unit tests to ensure code coverage
- Working on chainlink - aggregator smart contract

20/ - Added 2 new github action on the elrond-go repository, one to check if test coverage was increased or decreased and one to ensure if all our 3-rd party modules are up to date
- Continue to working on the task to move all the elastic indexer code in a new repository

21/ - Refactoring transaction status computer component
- Improve error handling and add unit tests

22/ "Stay Hungry Stay Foolish" and tune in soon for more of the #Elrondtech which powers the @ElrondNetwork ecosystem. Check our progress & get involved ■
https://t.co/YOoeEN0RIF