Engineering Weekly, 2024-05-02

Introduction

Migration from Go to Rust is done. New README.md will be prepared by tomorrow. Also, CONTRIBUTING.md will be added to provide steps to contribute and channels to join for new contributors.

P2P

Because of differences in implementation details between go-libp2p and rust-libp2p, now we need our bootstrap node. We're working on it, and once it's ready, we'll move on to Dockerfile and docker-compose so we can test `ramd` easily.

Live Object

`ramd` accepts wasm bytecode to create new live object. Execution is not straightforward as live objects have various function signatures. Solution is to lump all parameters, write it inside WASM instance, pass a pointer to it, and let WASM instance parse it.

A code for the last step is generated by Rust macro, abstracting it away from users. All users need to do is write a typical Rust library project, put live object SDK macros, and ta-da! The degree of abstraction varies depending on projects, but we want to provide good developer experience as much as we can. For more information, you can refer to CosmWasm SDK, Near SDK, etc.

We're working on writing sample live object code using SDK without macro, and then executing it via RPC request.

Conclusion

We’re pushing forward to make end-to-end functioning RAM network; users can create live objects, execute them, and user-sent messages are broadcasted across RAM nodes.

FAQ

Q: Any tasks I can contribute?

A: It’s hard to find independent tasks for now. Things are small so they’re intertwined. It could be a while to find some independent ones.

Q: What about non-coding contribution?

A: Sounds great! Tangible discussions are always welcomed. Good understanding of our code base and approaches would be prerequisite so feel free to ask questions if any. You can use our telegram chat group or discord.