Introduction

The open-source RChain project is building a *decentralized, economic, censorship-resistant, public compute infrastructure and blockchain*. It will host and execute programs popularly referred to as "smart contracts". It will be trustworthy, scalable, concurrent, with proof-of-stake consensus and content delivery.

Using smart contracts, a broad array of fully-scalable decentralized applications (dApps) can be built on the top of this platform. DApps may address areas such as identity, tokens, timestamping, financial services, monetized content delivery, Decentralized Autonomous Organizations (DAOs), exchanges, reputation, private social networks, marketplaces, and many more.

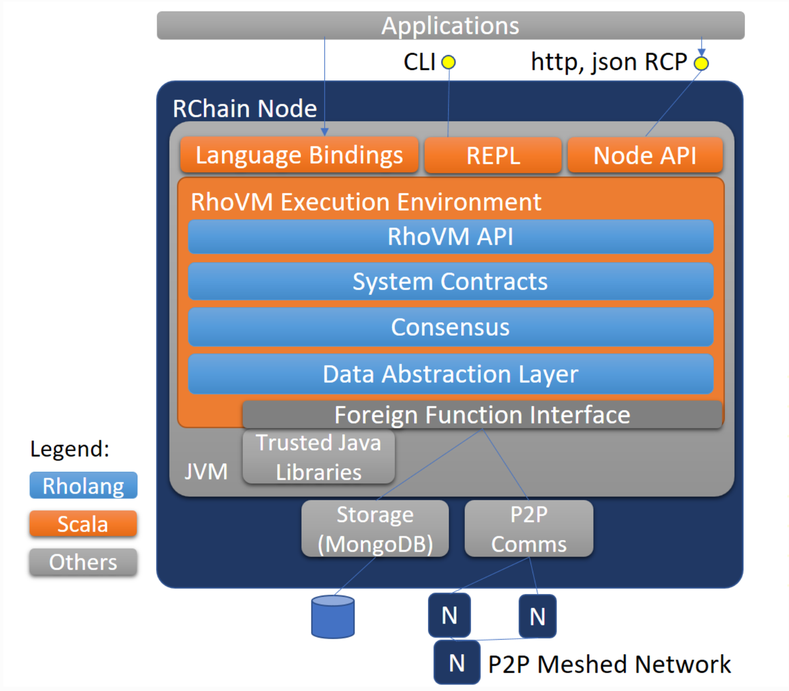

Figure: High-level RChain Architecture

The RChain Network implements direct node-to-node communication, where each node runs the RChain platform and a set of dApps on the top of it.

The heart of an RChain is the Rho Virtual Machine (RhoVM) Execution Environment, which runs multiple RhoVMs that are each executing a smart contract. These execute concurrently and are multi-threaded.

This concurrency, which is designed around on the formal models of mobile process calculi, along with an application of compositional namespaces, allows for what are in effect *multiple blockchains per node*. This multi-chain, independently executing virtual machine instances is in sharp contrast to a "global compute" design which constrains transactions to be executed sequentially, on a single virtual machine. In addition, each node can be configured to subscribe to and process the namespaces (blockchains) in which it is interested.

Like other blockchains, achieving consensus across nodes on the state of the blockchain is essential. RChain's protocol for replication and consensus is called Casper and is a proof-of-stake protocol. Similar to Ethereum, a contract starts out in one state, many nodes receive a signed transaction, and then their RhoVM instances execute that contract to its next state. An array of node operators, or "bonded validators" apply the consensus algorithm to crypto-economically verify that the entire history of state configurations and state transitions, of the RhoVM instance, are accurately replicated in a distributed data store.

The blockchain contracts (aka smart contracts, processes, or programs), including system contracts included in the installation are written in the RChain general-purpose language "**Rholang**" (Reflective higher-order language). Derived from the rho-calculus computational formalism, Rholang supports internal programmatic concurrency. It formally expresses the communication and coordination of many processes executing in parallel composition. Rholang naturally accommodates industry trends in code mobility, reactive/monadic API's, parallelism, asynchronicity, and behavioral types.

Since nodes are internally concurrent, and each need not run all namespaces (blockchains), the system will be *scalable*.

Since the contract language and its VM are build from the formal specifications of provable mathematics, and since the compiler pipeline and engineering approach is *correct by construction*, we expect the platform will be regarded as *trustworthy*.

Previous
Next