

## Designing a Transaction Broadcaster Service

The Transaction Broadcaster Service system that plays a crucial role in signing and sending transactions to an EVM-compatible blockchain network. It ensures that all the transactions are processed reliably. Moreover, it also keeps tracks of transaction statuses and allows manual retries if needed. The service consists of the several main components. The components will be API Layer, Transaction Processing Layer, Transaction Broadcaster, Retry Mechanism, Transaction and Status Tracker.

The API Layer provides an endpoint that allows other services to submit transactions for broadcasting. The endpoint is POST/transaction/broadcast. The API Layer accepts transaction data in a predefined format, then it validates the request and returns HTTP responses. The HTTP status code '200 OK' indicates that the request is valid, '4xx Client Errors' indicates that the request is invalid and '5xx Server Errors' indicates that there is a server-side error.

After request is accepted, the transaction enters the Transaction Processing Layer. The Transaction Queue stores incoming requests temporarily to ensure they are processed properly. Signing Module uses cryptographic key to sign the transaction securely before submission. The Transaction Formatter converts the transactions into correct format for broadcasting.

The Broadcasting Layer sends the signed transaction to blockchain network using JSON-RPC. It sends the eth\_sendRawTransaction RPC method to broadcast the signed transaction. It is also responsible for Response Handling. If successful, it will be marked as success, else it will be marked as fail. If by chance the response not received within a certain time period, the transaction is retried the maximum retry limit is reached.

The Retry Mechanism ensures that failed transactions aren't immediately marked as failures but instead are tried in a structured and efficient manner. To handle failures, the system includes Exponential Backoff, Manual Retry Feature and Failure Threshold. Exponential Backoff increases the time between retry attempts. Manual Retry Feature allows admins to manually retry failed transactions. For Failure Threshold, if a transaction continues to fail beyond a limit, it will be marked as failed.

The Transaction Status Tracker ensures that the status of each transaction is accurately tracked and updated as it progresses through the retry process. It records the status of each transaction of each stage. It also stores the number of retries attempted. This allows the system to know exactly what's occurring.

The transaction broadcaster service follows a structured workflow to ensure reliable processing. First, the API validates incoming transaction requests and stores them in a queue. The signing module then signs and formats the transaction before sending it to the blockchain via JSON-RPC. If successful, the transaction is marked as complete and if it fails, the system automatically retries using exponential backoff. If all retries fail, the transaction is recorded as failed. The system also tracks transaction statuses in a

database, allowing admins to monitor, review, and manually retry failed transactions when needed.

To ensure reliability, the system uses persistent storage to prevent transaction loss, proper shutdowns to save ongoing transactions, and redundancy with multiple blockchain nodes. For high transaction volumes, it employs asynchronous processing, a worker-based architecture, and batch processing to improve efficiency. To handle node failures, the system uses a circuit breaker to pause retries, fallback nodes for rerouting transactions, and rate limiting to prevent network overload.

The Transaction Broadcaster Service plays an important role in ensuring secure and reliable transaction processing for blockchain networks. By implementing asynchronous processing, automatic retries, and an admin dashboard, the system becomes fault-tolerant, scalable, and efficient. This design ensures that transactions are not lost, failures are handled gracefully, and admins have full control over transaction management. This service ensures reliable and high-performance transaction broadcasting for blockchain-based applications.