Yoda

Yoda is a program that is used by BandChain's validator nodes to automatically fulfill data for oracle requests.

Since a subset of validators who are selected for a data request must return the data they received from running the specified data source(s), each of them have to send aMsgReportData transaction to BandChain in order to fulfill their duty.

Although the transaction can be sent manually by user, it is not convenient, and would be rather time-consuming. Furthermore, most data providers already have APIs that can be used to query data automatically by another software. Therefore, we have developed Yoda to help validators to automatically query data from data providers by executing data source script, then submit the result to fulfill the request.

Yoda's execution flow consists of the follwing steps

1. Listening on incoming oracle requests and filter them.
2. Loading data source script based on received requests.
3. Execute the data source to get a result from data provider.
4. Collect result into messages and store in local pending list.
5. Send transaction based on pending list to fulfill oracle requests.

Yoda usually run as a process alongside the main BandChain daemon process. It holds another wallet (private key) account called areporter account , which is owned by validator account. The reporter account is used to help validator submit transactions of reporting data.

The reason of using reporter account instead of validator account is to maintain security of validator's private key. If Yoda's server is compromised, then reporter's private key may be exposed but not validator's private key. Moreover, most validators use hardware wallets, which is not designed for server that runs 24/7.

It's mechanism is designed as event-driven approach, which handle tasks concurrently to improve performance.

The image above represents high-level mechanism of Yoda.

The following section describes mechanism of the Yoda in each step.

1. Subscribe to Events and Filter Only Oracle Requests that should be responsible to

Firstly, Yoda listen to newly created transaction events occurred on BandChain via Tendermint's RPC Websocket. Then, it is extracted to check whether there is any event of newly created oracle request inside the transaction. If the event is found, then it is filtered to only select request that is assigned to the validator that yoda is responsible for.

2. Extract Oracle Request Events and Load Data Source Scripts

After oracle request events have been filtered, they are extracted to collect list of data source and calldata that need to be executed for this request.

Then, We have to also provide information for verifying request, which are as follow

- Chain ID of the BandChain
- Validator address
- Oracle request ID
- Data source's external ID (indicated by request)
- Signature of above content signed by Yoda's reporter account
- Public key of Yoda's reporter account for verify signature

These information can be used by data provider to ensure that incoming requests do really need the result to fulfill oracle requests and not to be used elsewhere or by someone else other than assigned validators.

3. Execute Data Source on Executor

Yoda executes those data source scripts by sending their script files along with necessary information for verifying oracle request to Yoda's executor environment, which helps running data source scripts and returns results to Yoda. More information about data source executor can be found inthis section .

4. Collect Results from Executor, Add to Local Pending List

Once results have been returned from Yoda's executor, Yoda constructsMsgReportData based on given answers and append to a pending message queue. As submitting a transaction takes some time and it is impossible to sign multiple transactions at the same time, the pending list can be accumulated.

5. Pack unsend ReportMsg(s) to a transaction and send to BandChain node

Once Yoda's reporter account is ready to sign a transaction, as a transaction can contains multiple messages, all messages in the pending list will be taken to construct the new transaction, then Yoda sign the transaction and send to BandChain to fulfill target oracle requests. Previous Become a Validator Next Introduction