Harmony RPC 框架结构

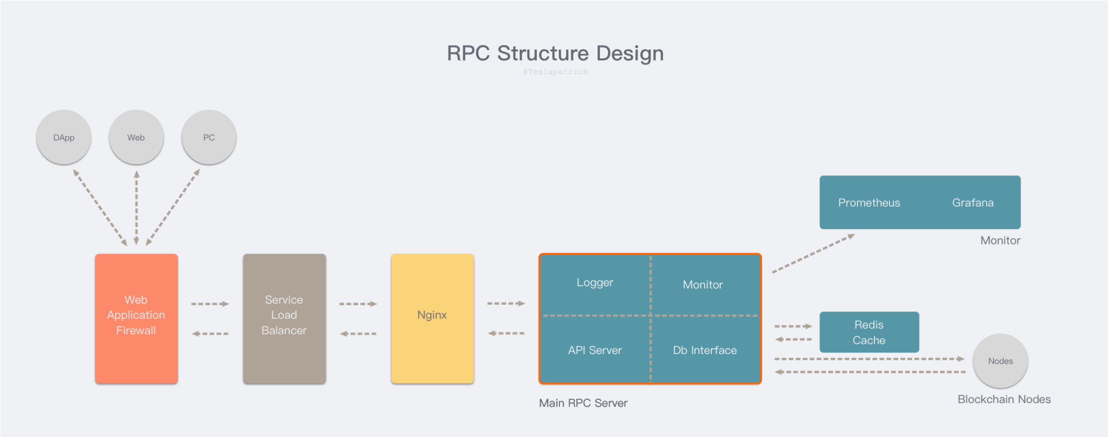
时间 / 版本	主要功能	修改人	备注
2022.02.09	RPC 框架 Specs	程华峥、谢浩耿	第一版

1 需求分析

基于测试的结果，Harmony 官方提供的 RPC Endpoint 主要存在的问题有：稳定性不能保证、接口 RPS（request per second）低（少量单个请求 / 并发请求）。针对 Harmony 的问题，我们提出了以下方案。

2 功能分析

基于讨论和以上需求分析的结果，针对 Harmony 的 RPC 接口加速的需求，RPC 加速的结构如下图所示。



DApp、Web、PC 端的请求，经过 Load Balancer，最终会到达我们的 RPC Server。

RPC Server 是主要的开发部分，包括 4 个部分：

a） API Server

b） Db Interface

c） Logger

d） Monitor

2.1API Server

API Server 是主要的 RPC 请求处理的入口，使用 RESTful POST 接口实现官方的接口。这块的功能分为以下几个部分：

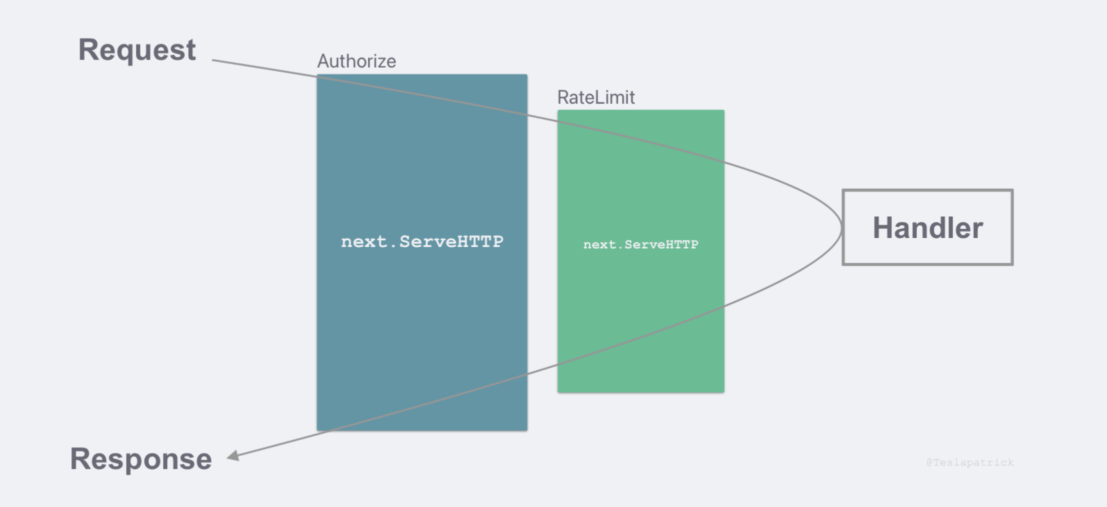
a）Harmony 官方接口功能；

b）JWT 鉴权功能（可选开发）

c）限流功能（可选开发）

Harmony 官方接口，直接复用官方的接口格式：Request、Response、错误代码。JWT 鉴权可以实现接口授权访问功能。限流是为了限制某个 IP 或是某个请求 method 的 rate。

请求的处理流程如图：



Handler 是最终处理 request 的模块，从 request 取出请求的 method、params，不同的 method 对应不同的 handler（详细的 handler 在第三章描述）。Handler 处理请求，从 Redis Cache、Nodes 中取得数据，同时将请求的结果返回给请求端。

2.2 Db Interface

Db Interface 主要是请求数据。数据的来源可以是 Redis Cache、Nodes 或是 SQL Server 等。由于目前的瓶颈主要在读取链上的数据，我们可以使用 Redis Cache 存储一些量大且不变的数据，如果是 Balance 等经常更新的数据，不适合存储在 Redis 中，同时要考虑 Redis 数据的读写分离，增加 Cache 系统的稳定性。

Redis Cache 的缓存策略，最贴近目前需求的是 Cache Aside Pattern：

* cache miss：DApp 先从 cache 取链上数据，如果没有得到，则从 Nodes 中取数据，成功返回后，放到 Redis 缓存中。

* cache hit：DApp 从 cache 中取链上数据，取到后返回。

* cache update：先把链上数据存到 Nodes 的本地数据库中，成功后，再让缓存中的数据失效，等待下一次 cache miss 来更新 cache。

这种缓存策略，可以先等 Nodes 同步完节点数据，再 cache miss 后，更新 redis。同时因为是同步所有的静态链上数据到 redis，对于 Harmony 来说，共识采用的是改进的 PBFT，则链上数据不会出现 re-org 的情况，则不会引发 cache update 的操作。

2.3 Logger

Logger 主要是打印 log 到文件中，记录 RPC Server 运行的流程，一些关键操作。方便日志查询归档等。Logger 可以考虑在基础 Log 功能的基础上，增加接口 Request 中的 Method 统计功能，方便向 Monitor 传递数据，展示接口调用状态。

2.4 Monitor

Monitor 是监听 RPC Server 的运行状态，Prometheus 可以订阅 Logger 的数据源，生成状态数据，交由 Grafana 展示监控状态数据。

3 详细设计

3.1 API Server

API Server 实现了 Harmony 的兼容 RPC，同时可以利用 Go 的 Interface 特性兼容其他类似的 RPC 服务。

现有支持 Harmony 的 RPC 接口以及对应的 request、response 值如下表：

标号	支持的 Json-RPC 接口	是否 Redis 索引	备注
	Account		
1	hmyv2_getBalance	否	有状态
2	hmyv2_getBalanceByBlockNumber	是	无状态
3	hmyv2_getStakingTransactionsCount	否	有状态
4	hmyv2_getStakingTransactionsHistory	待定	有状态

5	hmyv2_getTransactionsCount	否	有状态
6	hmyv2_getTransactionsHistory	待定	有状态
Blockchain / Blocks			
7	hmyv2_getBlocks	是	无状态
8	hmyv2_getBlockByNumber	是	无状态
9	hmyv2_getBlockByHash	是	无状态
10	hmyv2_getBlockSigners	是	无状态
11	hmyv2_getBlockSignersKeys	待定	无状态
12	hmyv2_getBlockTransactionCountByNumber	待定	有状态
13	hmyv2_getBlockTransactionCountByHash	待定	有状态
14	hmyv2_getHeaderByNumber	是	无状态
15	hmyv2_getLatestChainHeaders	否	有状态
16	hmyv2_latestHeader	否	有状态
Blockchain / Network			
17	hmyv2_blockNumber	是（定时失效）	有状态
18	hmyv2_getCirculatingSupply	是（定时失效）	有状态
19	hmyv2_getEpoch	是（定时失效）	有状态
20	hmyv2_getLastCrossLinks	是（定时失效）	有状态
21	hmyv2_getLeader	是（定时失效）	有状态
22	hmyv2_gasPrice	否	有状态
23	hmyv2_getShardingStructure	否	无状态
24	hmyv2_getTotalSupply	否	有状态
25	hmyv2_getValidators	是（定时失效）	有状态
26	hmyv2_getValidatorKeys	是（定时失效）	有状态
Blockchain / Node			
27	[WIP] hmyv2_getCurrentBadBlocks	否	
28	hmyv2_getNodeMetadata	是（定时失效）	有状态
29	hmyv2_protocolVersion	是	无状态
30	net_peerCount	否	有状态
Smart Contract			
31	hmyv2_call	待定	有状态
32	hmyv2_estimateGas	否	有状态
33	hmyv2_getCode	是	无状态
34	hmyv2_getStorageAt	待定	有状态
Staking / Delegation			
35	hmyv2_getDelegationsByDelegator	待定	有状态
36	hmyv2_getDelegationsByDelegatorByBlockNumber	是（定时失效）	有状态
37	hmyv2_getDelegationsByValidator	是（定时失效）	有状态
Staking / Validator			
38	hmyv2_getAllValidatorAddresses	是（定时失效）	有状态
39	hmyv2_getAllValidatorInformation	是（定时失效）	有状态
40	hmyv2_getAllValidatorInformationByBlockNumber	是	有状态
41	hmyv2_getElectedValidatorAddresses	是（定时失效）	有状态

42	hmyv2_getValidatorInformation	是（定时失效）	有状态
Staking / Network			
43	hmyv2_getCurrentUtilityMetrics	否	有状态
44	hmyv2_getMedianRawStakeSnapshot	否	有状态
45	hmyv2_getStakingNetworkInfo	否	有状态
46	hmyv2_getSuperCommittees	是（定时失效）	有状态
Transaction / Cross Shard			
47	hmyv2_getCXReceiptByHash	否	无状态
48	hmyv2_getPendingCXReceipts	待定	有状态
49	hmyv2_resendCx	否	
Transaction / Pool			
50	hmyv2_getPoolStats	是（定时失效）	有状态
51	hmyv2_pendingStakingTransactions	是（定时失效）	有状态
52	hmyv2_pendingTransactions	是（定时失效）	有状态
Transaction / Staking			
53	hmyv2_getCurrentStakingErrorSink	待定	有状态
54	hmyv2_getStakingTransactionByBlockNumberAndIndex	是	无状态
55	hmyv2_getStakingTransactionByBlockHashAndIndex	是	无状态
56	hmyv2_getStakingTransactionByHash	是	无状态
57	hmyv2_sendRawStakingTransaction	是	无状态
Transaction / Transfer			
58	hmyv2_getCurrentTransactionErrorSink	待定	有状态
59	hmyv2_getTransactionByBlockHashAndIndex	是	无状态
60	hmyv2_getTransactionByBlockNumberAndIndex	是	无状态
61	hmyv2_getTransactionByHash	是	无状态
62	hmyv2_getTransactionReceipt	是	无状态
63	hmyv2_sendRawTransaction	否	有状态

3.1.1 Account

接口功能：从最新状态获取地址余额		
Method： hmyv2_getBalance	**Request Parameters：** String - Wallet address	**Response Result：** Number - Wallet balance at given block in Atto
接口功能：从特定高度的状态获取地址余额		
Method： hmyv2_getBalanceByBlockNumber	**Request Parameters：** String - Wallet address Number - Block to get balance at	**Response Result：** Number - Wallet balance at given block in Atto
接口功能：		
Method： hmyv2_getStakingTransactionsCount	**Request Parameters：** String - Wallet address	**Response Result：** Number - Number of staking transactions

	String - Type of staking transaction (SENT, RECEIVED, ALL)	
接口功能：		
Method： hmyv2_getStakingTransactionsHistory	**Request Parameters：** Object - Transaction history args address - String : Wallet address pageIndex - Number : Optional, which page of transactinos to return, default 0 pageSize - Number : Optional, how many transactions to display per page, default 1000 fullTx - Bool : Optional, return full transaction data or just transaction hashes, default false txType - String: Optional, which type of transactions to display ("ALL", "RECEIVED", or "SENT"), default "ALL" order - String: Optional, sort transactions in ascending or descending order based on timestamp ("ASC" or "DESC"), default "ASC"	**Response Result：** If txType is true Array of Object : List of staking transactions blockHash - String: Block hash that transaction was finalized. null if the transaction is pending. blockNumber - Number: Block number that transaction was finalized. null if the transaction is pending. from - String: Wallet address of sender timestamp - Number : Timestamp in Unixtime when transaction was finalized gasPrice - Number: Gas price in Atto gas - Number: Gas limit in Atto hash - String: Transaction hash nonce - Number: Wallet nonce for the transaction transactionIndex - Number: Index of transaction in block. null if the transaction is pending. type - String : Type of staking transaction msg - Object: Staking transaction data If txType is false Array of String : List of staking transaction hashes
接口功能：		
Method： hmyv2_getTransactionsCount	**Request Parameters：** String - Wallet address String - Type of transaction (SENT, RECEIVED, ALL)	**Response Result：** Number - Number of transactions
接口功能：		
Method： hmyv2_getTransactionsHistory	**Request Parameters：** Object - Transaction history args address - String : Wallet address pageIndex - Number : Optional, which page of transactions to return, default 0	**Response Result：** If txType is true Array of Object

	pageSize - Number : Optional, how many transactions to display per page, default 1000 fullTx - Bool : Optional, return full transaction data or just transaction hashes, default false txType - String: Optional, which type of transactions to display ("ALL", "RECEIVED", or "SENT"), default "ALL" order - String: Optional, sort transactions in ascending or descending order based on timestamp ("ASC" or "DESC"), default "ASC"	blockHash - String: Block hash that transaction was finalized. null if the transaction is pending. blockNumber - Number: Block number that transaction was finalized. null if the transaction is pending. from - String : Wallet address timestamp - Number : Timestamp in Unix time when transaction was finalized gas - Number: Gas limit in Atto gasPrice - Number: Gas price in Atto hash - String: Transaction hash input - String: Transaction data, used for smart contracts nonce - Number: Wallet nonce for the transaction to - String: Wallet address of receiver transactionIndex - Number: Index of transaction in block. null if the transaction is pending. value - Number: Amount transfered in Atto shardID - Number : Shard where amount is from toShardID - Number : Shard where the amount is sent If txType is false Array of String : List of transaction hashes

3.1.2 Blockchain

接口功能：		
Method： hmyv2_getBlocks	**Request Parameters：** Number : Start block Number : End block Object： * withSigners - Bool : Include block signer wallet addresses * fullTx - Bool : Include full transaction data	**Response Result：** Array: List of blocks See hmyv2_getBlockByNumber for block structure

	* inclStaking - Bool : Include full staking transactions	
接口功能：		
Method： hmyv2_getBlockByNumber	**Request Parameters：** Number : Block number Object: * fullTx - Bool : Include full transaction data * inclTx - Bool : Include regular transactions * InclStaking - Bool : Include staking transactions	**Response Result：** Object difficulty - Number : Unused, legacy from Eth epoch - Number : Epoch number of block extraData - String : Hex representation of extra data in the block gasLimit - Number : Maximum gas that can be used for transactions in the block gasUsed - Number : Amount of gas used for transactions in the block hash - String : Block hash logsBloom - String : Bloom logs miner - String : Wallet address of the leader that proposed this block mixHash - String : Unused, legacy from Eth nonce - Number : Unused, legacy from Eth number - Number : Block number parentHash - String : Hash of parent block receiptsRoot - String : Hash of transaction receipt root size - Number : Block size in bytes stakingTransactions - JSON Array: List of staking transactions finalized in this block stateRoot - String : Hash of state root timestamp - Number : Unix timestamp of the block transactions - JSON Array : List of transactions finalized in this block transactionsRoot - String : Hash of transactions root uncles - JSON Array : Unused, legacy from Eth viewID - Number : View ID
接口功能：		

Method： hmyv2_getBlockByHash	**Request Parameters：** String : Block hash Object * fullTx - Bool : Include full transaction data * inclTx - Bool : Include regular transactions * InclStaking - Bool : Include staking transactions	**Response Result：** See hmyv2_getBlockByNumber for block structure.
接口功能：		
Method： hmyv2_getBlockSigners	**Request Parameters：** Number : Start block Number : End block Object： * withSigners - Bool : Include block signer wallet addresses * fullTx - Bool : Include full transaction data * inclStaking - Bool : Include full staking transactions	**Response Result：** Array : List of block signer wallet addresses
接口功能：		
Method： hmyv2_getBlockSignersKeys	**Request Parameters：** Number : Block number	**Response Result：** Array : List of block signer public BLS keys
接口功能：		
Method： hmyv2_getBlockTransactionCountByNumber	**Request Parameters：** Number : Block number	**Response Result：** Number : Number of transactions in block
接口功能：		
Method： hmyv2_getBlockTransactionCountByHash	**Request Parameters：** String : Block hash	**Response Result：** Number : Number of transactions in block
接口功能：		
Method： hmyv2_getHeaderByNumber	**Request Parameters：** Number : Block number	**Response Result：** See hmyv2_latestHeader for reply structure
接口功能：		
Method： hmyv2_getLatestChainHeaders	**Request Parameters：** None	**Response Result：** Object: beacon-chain-header - Object shard-id - Number : Shard ID block-header-hash - String : Block header hash

		block-number - Number : Block number view-id - Number : View ID epoch - Number : Epoch number shard-chain-header - Object shard-id - Number : Shard ID block-header-hash - String : Block header hash block-number - Number : Block number view-id - Number : View ID epoch - Number : Epoch number
接口功能：		
Method： hmyv2_latestHeader	**Request Parameters：** None	**Response Result：** Object: blockHash - String : Block hash blockNumber - Number : Block number shardID - Number : Shard ID leader - String : Wallet address of leader that proposed this block if prestaking, otherwise sha256 hash of leader's public bls key viewID - Number : View ID of the block epoch - Number : Epoch of block timestamp - String : Timestamp that the block was finalized unixtime - Number : Timestamp that the block was finalized in Unix time lastCommitSig - String : Hex representation of aggregated signatures of the previous block lastCommitBitmap - String : Hex representatino of the aggregated signature bitmap of the previous block
接口功能：		
Method： hmyv2_blockNumber	**Request Parameters：** None	**Response Result：** Number - Current block number
接口功能：		
Method： hmyv2_getCirculatingSupply	**Request Parameters：** None	**Response Result：** Number : Circulation supply of tokens in ONE
接口功能：		

Method： hmyv2_getEpoch	**Request Parameters：** None	**Response Result：** Number - Current block number
接口功能：		
Method： hmyv2_getLastCrossLinks	**Request Parameters：** None	**Response Result：** Array of Object hash - String : Parent block hash block-number - Number : Block number view-id - Number : View ID signature - String : Hex representation of aggregated signature signature-bitmap - String : Hex representation of aggregated signature bitmap shard-id - Number : Shard ID epoch-number - Number : Block epoch
接口功能：		
Method： hmyv2_getLeader	**Request Parameters：** None	**Response Result：** String - Wallet address of current leader
接口功能：		
Method： hmyv2_gasPrice	**Request Parameters：** None	**Response Result：** Number - Current average gas price of transactions
接口功能：		
Method： hmyv2_getShardingStructure	**Request Parameters：** None	**Response Result：** Array of Object current - Bool : If this node is currently on this shard ID http - String : HTTPS API endpoint for this shard ID shardID - Number : Shard ID ws - String : Websocket API endpoint for this shard ID
接口功能：		
Method： hmyv2_getTotalSupply	**Request Parameters：** None	**Response Result：** Number : Total number of pre-mined tokens
接口功能：		
Method： hmyv2_getValidators	**Request Parameters：** Number : Epoch number	**Response Result：** Object shardID - Number : Shard ID validators - Array of Object * address : Wallet address

		* balance : Balance of wallet
接口功能：		
Method： hmyv2_getValidatorKeys	**Request Parameters：** Number : Epoch Number	**Response Result：** Array : List of public BLS keys in the elected committee
接口功能：		
Method： [WIP] hmyv2_getCurrentBadBlocks	**Request Parameters：** None	**Response Result：** Array : List of bad blocks in node memory
接口功能：		
Method： hmyv2_getNodeMetadata	**Request Parameters：** None	**Response Result：** Object: blskey - Array : List of BLS keys on the node version - String : Harmony binary version network - String : Network name that the node is on (Mainnet or Testnet) chain-config – Object: * chain-id - Number : Chain ID for the network * cross-tx-epoch - Number : Epoch at which cross cross shard transactions were enabled * cross-link-epoch - Number : Epoch at which cross links were enabled * staking-epoch - Number : Epoch at which staking was enabled * prestaking-epoch - Number : Epoch at which pre-staking began * quick-unlock-epoch - Number : Epoch at which undelegations unlocked in one epoch * eip155-epoch - Number : Epoch at with EIP155 was enabled * s3-epoch - Number : Epoch at which Mainnet V0 was launched * receipt-log-epoch - Number : Epoch at which receipt logs were enabled is-leader - Bool : Whether the node is currently leader or not shard-id - Number : Shard that the node is on

		current-epoch - Number : Current epoch blocks-per-epoch - Number : Number of blocks per epoch (only available on Shard 0) role - String : Node type (Validator or ExplorerNode) dns-zone - String : Name of the DNS zone is-archival - Bool : Whether the node is currently in state pruning mode or not node-unix-start-time - Number : Start time of node in Unix time p2p-connectivity – Object: * total-known-peers - Number : Number of known peers * connected - Number : Number of connected peers * not-connected - Number : Number known peers not connected
接口功能：		
Method： hmyv2_protocolVersion	**Request Parameters：** None	**Response Result：** Number : Protocol version
接口功能：		
Method： net_peerCount	**Request Parameters：** None	**Response Result：** String - Number of peers represented as a Hex string

3.1.3 Smart Contract

接口功能：		
Method： hmyv2_call	**Request Parameters：** Object - Smart contract call object to - String : Smart contract address from - String : Wallet address, optional gas - Number : Gas to execute the smart contract call, optional gasPrice - Number : Gas price to execute smart contract call, optional value - Number : Value sent with the smart contract call, optional	**Response Result：** String - Return value of the executed smart contract

	data - String : Hash of smart contract method and parameters, optional Number: Block number	
接口功能：		
Method： hmyv2_estimateGas	**Request Parameters：** Object - Smart contract call object to - String : Smart contract address from - String : Wallet address, optional gas - Number : Gas to execute the smart contract call, optional gasPrice - Number : Gas price to execute smart contract call, optional value - Number : Value sent with the smart contract call, optional data - String : Hash of smart contract method and parameters, optional Number : Block number	**Response Result：** String - Hex of estimated gas price of smart contract call
接口功能：		
Method： hmyv2_getCode	**Request Parameters：** String : Smart Contract address Number : Block number	**Response Result：** String - Hex of smart contract code
接口功能：		
Method： hmyv2_getStorageAt	**Request Parameters：** String : Smart contract address String : Hex representation of storage location Number : Block number	**Response Result：** String - Data stored at the smart contract location

3.1.4 Staking

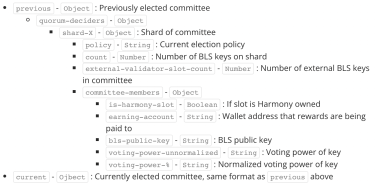
接口功能：		
Method： hmyv2_getDelegationsByDelegator	**Request Parameters：** String : Delegator address	**Response Result：** JSON Array of JSON Object: * validator_address - String : Validator wallet address * delegator_address - String : Delegator wallet address * amount - Number : Amount delegated in atto

		* reward - Number : Unclaimed rewards in Atto * Undelegations - JSON Array : List of pending undelegations
接口功能：		
Method： hmyv2_getDelegationsByDelegatorByBlockNumber	**Request Parameters：** String : Delegator wallet address Number : Block number	**Response Result：** See hmyv2_getDelegationsByDelegator for Delegator field descriptions
接口功能：		
Method： hmyv2_getDelegationsByValidator	**Request Parameters：** String : Validator wallet address	**Response Result：** Array of Object: * validator_address - String : Validator wallet address * delegator_address - String : Delegator wallet address * amount - Number : Amount delegated in Atto * reward - Number : Unclaimed rewards in Atto * Undelegations - JSON Array : List of pending undelegations
接口功能：		
Method： hmyv2_getAllValidatorAddresses	**Request Parameters：** None	**Response Result：** Array of String : List of wallet addresses that have created validators on the network
接口功能：		
Method： hmyv2_getAllValidatorInformation	**Request Parameters：** Number : Page to request (page size is 100), -1 for all validators	**Response Result：** Array of Object See hmyv2_getValidatorInformation for Validator field descriptions
接口功能：		
Method： hmyv2_getAllValidatorInformationByBlockNumber	**Request Parameters：** Number : Page number, -1 for all Number : Block number	**Response Result：** See hmyv2_getValidatorInformation for Validator field descriptions
接口功能：		
Method： hmyv2_getElectedValidatorAddresses	**Request Parameters：** None	**Response Result：** Array of String : List of wallet addresses that are currently elected
接口功能：		
Method： hmyv2_getValidatorInformation	**Request Parameters：** String : Validator wallet address	**Response Result：** Object: validator – Object:

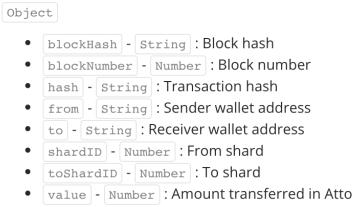
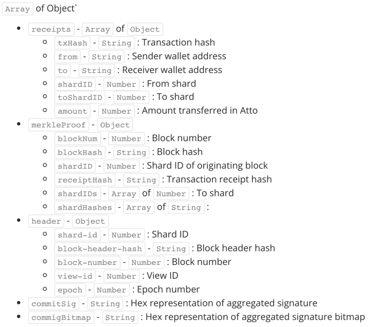
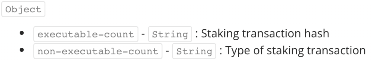
		* bls-public-keys - Array : List of public BLS keys associated with the validator wallet address * last-epoch-in-committee - Number : Last epoch any key of the validator was elected * min-self-delegation - Number : Amount that validator must delegate to self in Atto * max-total-delegation - Number : Total amount that validator will accept delegations until in Atto * rate - String : Current commission rate * max-rate - String : Max commission rate a validator can charge * max-change-rate - String : Maximum amount the commission rate can increase in one epoch * update-height - Number : Last block validator editted their validator information * name - String : Validator name, displayed on the Staking Dashboard * identity - String : Validator identity, must be unique * website - String : Validator website, displayed on the Staking Dashboard * security-contact - String : Method to contact the validator * details - String : Validator details, displayed on the Staking Dashboard * creation-height - Number : Block in which the validator was created * address - String : Validator wallet address * delegations - Array : List of delegations See hmyv2_getDelegationsByDelegator for delegation object format * metrics - Object : BLS key earning metrics for current epoch * by-bls-key - Array of Object * key – Object:

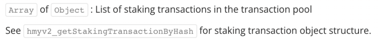
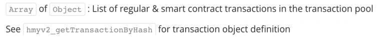
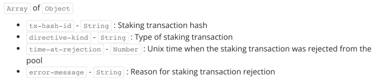
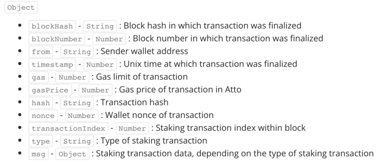
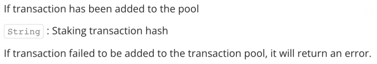
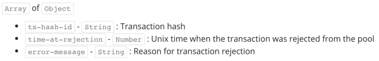
		* bls-public-key - String : BLS public key * group-percent - String : Key voting power in shard * effective-stake - String : Effective stake of key * raw-stake - String : Actual stake of key * earning-account - String : Validator wallet address * overall-percent - String : Percent of effective stake * shard-id - Number : Shard ID that key is on * earned-reward - Number : Lifetime reward key has earned * total-delegation - Number : Total amount delegated to validator * currently-in-committee - Bool : If key is currently elected * epos-status - String : Currently elected, eligible to be elected next epoch, or not eligible to be elected next epoch * epos-winning-stake - String : Total effective stake of the validator * booted-status - String : Banned status * active-status - String : Active or Inactive * lifetime - Object * reward-accumulated - Number : Lifetime reward accumulated by the validator * blocks – Object: * to-sign - Number : Number of blocks available to the validator to sign * signed - Number : Number of blocks the validator has signed * apr - String : Approximate Return Rate * epoch-apr - Array : List of APR per epoch * Epoch - Number : Epoch number

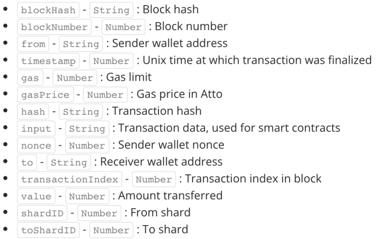
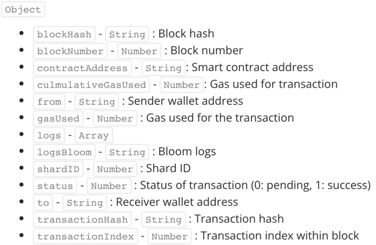
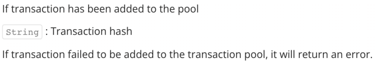
		* Value - String : Calculated APR for that epoch
接口功能：		
Method： hmyv2_getCurrentUtilityMetrics	**Request Parameters：** None	**Response Result：** AccumulatorSnapshot - Number : Total block reward given out in Atto CurrentStakedPercentage - String : Percent of circulating supply staked Deviation - String : Change in percent of circulating supply staked Adjustment - String : Change in circulating supply staked
接口功能：		
Method： hmyv2_getMedianRawStakeSnapshot	**Request Parameters：** None	**Response Result：** epos-median-stake - String : Effective median stake max-external-slots - Number : Number of available committee slots epos-slot-winners - Array of Object : Details for each slot winner * slot-owner - String : Wallet address of BLS key * bls-public-key - String : BLS public key * raw-stake - String : Actual stake * eposed-stake - String : Effective stake epos-slot-candidates - Array of Object : Details for each candidate * stake - Number : Actual stake in Atto * keys-at-auction - Array : List of BLS public keys * percentage-of-total-auction-stake - String : Percent of total network stake * stake-per-key - Number : Stake per BLS key in Atto * validator - String : Wallet address of validator
接口功能：		
Method： hmyv2_getStakingNetworkInfo	**Request Parameters：** None	**Response Result：** total-supply - String : Total number of pre-mined tokens circulating-supply - String : Number of tokens available in the network

		epoch-last-block - Number : Last block of epoch total-staking - Number : Total amount staked in Atto median-raw-stake - String : Effective median stake in Atto
接口功能：		
Method： hmyv2_getSuperCommittees	**Request Parameters：** None	**Response Result：**  • previous - Object : Previously elected committee ◦ quorum-deciders - Object ▪ shard-X - Object : Shard of committee ▪ policy - String : Current election policy ▪ count - Number : Number of BLS keys on shard ▪ external-validator-slot-count - Number : Number of external BLS keys in committee ▪ committee-members - Object ▪ is-harmony-slot - Boolean : If slot is Harmony owned ▪ earning-account - String : Wallet address that rewards are being paid to ▪ bls-public-key - String : BLS public key ▪ voting-power-unnormalized - String : Voting power of key ▪ voting-power-% - String : Normalized voting power of key • current - Ojbect : Currently elected committee, same format as previous above

3.1.5 Cross Shard

接口功能：		
Method： hmyv2_getCXReceiptByHash	**Request Parameters：** String : Cross shard receipt hash	**Response Result：**  Object • blockHash - String : Block hash • blockNumber - Number : Block number • hash - String : Transaction hash • from - String : Sender wallet address • to - String : Receiver wallet address • shardID - Number : From shard • toShardID - Number : To shard • value - Number : Amount transferred in Atto
接口功能：		
Method： hmyv2_getPendingCXReceipts	**Request Parameters：** None	**Response Result：**  Array of Object` • receipts - Array of Object ◦ txHash - String : Transaction hash ◦ from - String : Sender wallet address ◦ to - String : Receiver wallet address ◦ shardID - Number : From shard ◦ toShardID - Number : To shard ◦ amount - Number : Amount transferred in Atto • merkleProof - Object ◦ blockNum - Number : Block number ◦ blockHash - String : Block hash ◦ shardID - Number : Shard ID of originating block ◦ receiptHash - String : Transaction receipt hash ◦ shardIDs - Array of Number : To shard ◦ shardHashes - Array of String : • header - Object ◦ shard-id - Number : Shard ID ◦ block-header-hash - String : Block header hash ◦ block-number - Number : Block number ◦ view-id - Number : View ID ◦ epoch - Number : Epoch number • commitSig - String : Hex representation of aggregated signature • commigBitmap - String : Hex representation of aggregated signature bitmap
接口功能：		
Method： hmyv2_resendCx	**Request Parameters：** String : Cross shard receipt hash	**Response Result：** Bool : If cross shard receipt was successfully resent or not
接口功能：		
Method： hmyv2_getPoolStats	**Request Parameters：** None	**Response Result：**  Object • executable-count - String : Staking transaction hash • non-executable-count - String : Type of staking transaction
接口功能：		

Method： hmyv2_pendingStakingTransactions	**Request Parameters：** None	**Response Result：**  `Array` of `Object` : List of staking transactions in the transaction pool See `hmyv2_getStakingTransactionByHash` for staking transaction object structure.
接口功能：		
Method： hmyv2_pendingTransactions	**Request Parameters：** None	**Response Result：**  `Array` of `Object` : List of regular & smart contract transactions in the transaction pool See `hmyv2_getTransactionByHash` for transaction object definition
接口功能：		
Method： hmyv2_getCurrentStakingErrorSink	**Request Parameters：** None	**Response Result：**  `Array` of `Object` - `tx-hash-id` - `String` : Staking transaction hash - `directive-kind` - `String` : Type of staking transaction - `time-at-rejection` - `Number` : Unix time when the staking transaction was rejected from the pool - `error-message` - `String` : Reason for staking transaction rejection
接口功能：		
Method： hmyv2_getStakingTransactionByBlockNumberAndIndex	**Request Parameters：** Number : Block number Number : Staking transaction index	**Response Result：** See hmyv2_getStakingTransactionByHash for reply structure
接口功能：		
Method： hmyv2_getStakingTransactionByBlockHashAndIndex	**Request Parameters：** String : Block hash Number : Staking transaction index	**Response Result：** See hmyv2_getStakingTransactionByHash for reply structure
接口功能：		
Method： hmyv2_getStakingTransactionByHash	**Request Parameters：** String : Staking transaction hash	**Response Result：**  `Object` - `blockHash` - `String` : Block hash in which transaction was finalized - `blockNumber` - `Number` : Block number in which transaction was finalized - `from` - `String` : Sender wallet address - `timestamp` - `Number` : Unix time at which transaction was finalized - `gas` - `Number` : Gas limit of transaction - `gasPrice` - `Number` : Gas price of transaction in Atto - `hash` - `String` : Transaction hash - `nonce` - `Number` : Wallet nonce of transaction - `transactionIndex` - `Number` : Staking transaction index within block - `type` - `String` : Type of staking transaction - `msg` - `Object` : Staking transaction data, depending on the type of staking transaction
接口功能：		
Method： hmyv2_sendRawStakingTransaction	**Request Parameters：** String : Hex representation of signed staking transaction	**Response Result：**  If transaction has been added to the pool `String` : Staking transaction hash If transaction failed to be added to the transaction pool, it will return an error.
接口功能：		
Method： hmyv2_getCurrentTransactionErrorSink	**Request Parameters：** None	**Response Result：**  `Array` of `Object` - `tx-hash-id` - `String` : Transaction hash - `time-at-rejection` - `Number` : Unix time when the transaction was rejected from the pool - `error-message` - `String` : Reason for transaction rejection
接口功能：		
Method： hmyv2_getTransactionByBlockHashAndIndex	**Request Parameters：** String : Block hash Number : Transaction index	**Response Result：** See hmyv2_getTransactionByHash for reply structure
接口功能：		
Method： hmyv2_getTransactionByBlockNumberAndIndex	**Request Parameters：** Number : Block number Number : Transaction index	**Response Result：** See hmyv2_getTransactionByHash for reply structure
接口功能：		
Method：	**Request Parameters：**	**Response Result：**

hmyv2_getTransactionByHash	String : Transaction hash	 - `blockHash` - `String` : Block hash - `blockNumber` - `Number` : Block number - `from` - `String` : Sender wallet address - `timestamp` - `Number` : Unix time at which transaction was finalized - `gas` - `Number` : Gas limit - `gasPrice` - `Number` : Gas price in Atto - `hash` - `String` : Transaction hash - `input` - `String` : Transaction data, used for smart contracts - `nonce` - `Number` : Sender wallet nonce - `to` - `String` : Receiver wallet address - `transactionIndex` - `Number` : Transaction index in block - `value` - `Number` : Amount transferred - `shardID` - `Number` : From shard - `toShardID` - `Number` : To shard
接口功能：		
Method： hmyv2_getTransactionReceipt	**Request Parameters：** String : Transaction receipt	**Response Result：**  `Object` - `blockHash` - `String` : Block hash - `blockNumber` - `Number` : Block number - `contractAddress` - `String`: Smart contract address - `culmulativeGasUsed` - `Number`: Gas used for transaction - `from` - `String` : Sender wallet address - `gasUsed` - `Number` : Gas used for the transaction - `logs` - `Array` - `logsBloom` - `String` : Bloom logs - `shardID` - `Number` : Shard ID - `status` - `Number` : Status of transaction (0: pending, 1: success) - `to` - `String` : Receiver wallet address - `transactionHash` - `String` : Transaction hash - `transactionIndex` - `Number` : Transaction index within block
接口功能：		
Method： hmyv2_sendRawTransaction	**Request Parameters：** String : Hex representation of signed transaction	**Response Result：**  If transaction has been added to the pool `String` : Transaction hash If transaction failed to be added to the transaction pool, it will return an error.

3.1.6 错误处理

Method：	**Response Error：** error – Object: * code - Number: error code * message - String: error message

3.2Db Interface

3.2.1 Account

接口功能：	
Method： hmyv2_getBalance	**Redis：** <hash> balance(field:address, value:amount) [2s]
接口功能：	
Method： hmyv2_getBalanceByBlockNumber	**Redis：** <hash> balance:${blockNumber}(field:address, value:amount) [2s]
接口功能：	
Method： hmyv2_getStakingTransactionsCount	**Redis：** <hash> stakingTxCount:${txType}(field:address, value:count) [2s]

接口功能：	
Method： hmyv2_getStakingTransactionsHistory	**Redis：** 对应 fullTx=true 的情况下,可以通过 Nonce 检索： <zset> stakingTx:${address}(score:Nonce, member:txInfo)
接口功能：	
Method： hmyv2_getTransactionsCount	**Redis：** <hash> txCount(field:address, value:count) [2s]
接口功能：	
Method： hmyv2_getTransactionsHistory	**Redis：** 对应 fullTx=true 的情况下,可以通过 Nonce 检索： <zset> tx:${address}(score:Nonce, member:txInfo)

3.2.2 Blockchain

接口功能：	
Method： hmyv2_getBlocks	**Redis：** <hash> blockNumber(field:height, value:blockInfo) <zset> signers(score:height, member:signerInfo) <hash> tx:${height}(field:index, member:txInfo) <hash> stakingTx:${height}(field:index, member:stakingTxInfo) 后面三个可以换 set，不过每个高度都要 3 个 key，可能需要 expire
接口功能：	
Method： hmyv2_getBlockByNumber	**Redis：** <hash> blockNumber(field:height, value:blockInfo) <zset> signers(score:height, member:signerInfo) <hash> tx:${height}(field:index, member:txInfo) <hash> stakingTx:${height}(field:index, member:stakingTxInfo) 后面三个可以换 set，不过每个高度都要 3 个 key，可能需要 expire
接口功能：	
Method： hmyv2_getBlockByHash	**Redis：** <hash> blockHash(field:hash, value:blockInfo) <hash> blockHash(field:tx, value:txInfo) <hash> blockHash(field:stakingTx, value:stakingTxInfo)
接口功能：	
Method： hmyv2_getBlockSigners	**Redis：** <zset> signers(score:height, member:signerInfo) 可以换 set，不过每个高度都生成 1 个 key，可能需要 expire
接口功能：	
Method： hmyv2_getBlockSignersKeys	**Redis：** <zset> signersKey(score:height, member:signerInfo) 可以换 set，不过每个高度都生成 1 个 key，可能需要 expire
接口功能：	
Method：	**Redis：**

hmyv2_getBlockTransactionCountByNumber	<zset> tx(score:height, member:txInfo) 可以换set，不过每个高度都生成1个key，可能需要expire
接口功能：	
Method： hmyv2_getBlockTransactionCountByHash	**Redis：** <hash> blockHash(field:tx, value:txInfo)计数
接口功能：	
Method： hmyv2_getHeaderByNumber	**Redis：** <hash> blockNumber(value:height, value:blockInfo)
接口功能：	
Method： hmyv2_getLatestChainHeaders	**Redis：** <string> latestBeaconChainHeader beaconChainHeaderInfo [2s] <string> latestShardChainHeader shardChainHeaderInfo [2s]
接口功能：	
Method： hmyv2_latestHeader	**Redis：** <string> latestHeader headerInfo [2s]
接口功能：	
Method： hmyv2_blockNumber	**Redis：** <string> latestBlockNumber height [2s]
接口功能：	
Method： hmyv2_getCirculatingSupply	**Redis：** <string> circulatingSupply amount [时间待商榷]
接口功能：	
Method： hmyv2_getEpoch	**Redis：** <string> latestHeader headerInfo [2s]或者 <string> epoch epoch [时间待商榷]
接口功能：	
Method： hmyv2_getLastCrossLinks	**Redis：** <string> lastCrossLinkS1 CrossLinkS1Data [2s] <string> lastCrossLinkS2 CrossLinkS2Data [2s] <string> lastCrossLinkS3 CrossLinkS3Data [2s]
接口功能：	
Method： hmyv2_getLeader	**Redis：** <string> leader address [时间待商榷]
接口功能：	
Method： hmyv2_gasPrice	**Redis：** <string> gasPrice gasPrice [时间待商榷]
接口功能：	
Method： hmyv2_getShardingStructure	**Redis：** string或者list
接口功能：	
Method：	**Redis：**

hmyv2_getTotalSupply	<string> totalSupply totalSupply [时间待商榷]
接口功能：	
Method： hmyv2_getValidators	**Redis：** <set> validators:${height} validatorInfo
接口功能：	
Method： hmyv2_getValidatorKeys	**Redis：** <set> validatorKeys:${height} validatorKeysInfo
接口功能：	
Method： hmyv2_getCurrentBadBlocks	**Redis：**
接口功能：	
Method： hmyv2_getNodeMetadata	**Redis：**
接口功能：	
Method： hmyv2_protocolVersion	**Redis：** String protocolVersion [时间待商榷]
接口功能：	
Method： hmyv2_peerCount	**Redis：** String peerCount [时间待商榷]

3.2.3 Smart Contract

接口功能：	
Method： hmyv2_call	**Redis：**
接口功能：	
Method： hmyv2_estimateGas	**Redis：** <hash> estimateGas(field:address, value:gasUsed)
接口功能：	
Method： hmyv2_getCode	**Redis：** <hash> code(field:address, value:code)
接口功能：	
Method： hmyv2_getStorageAt	**Redis：** <hash> storageAt(field:address, value:data)

3.2.3 Staking

接口功能：	
Method： hmyv2_getDelegationsByDelegator	**Redis：** <set> delegationsByDelegator [时间待商榷]
接口功能：	
Method：	**Redis：**

hmyv2_getDelegationsByDelegatorByBlockNumber	<set> delegationsByDelegator:${address}:${height} [时间待商榷] 结果和上一个接口一样
接口功能：	
Method： hmyv2_getDelegationsByValidator	**Redis：** <set> delegationsByValidator:${address} [时间待商榷]
接口功能：	
Method： hmyv2_getAllValidatorAddresses	**Redis：** <set> validatorAddresses [时间待商榷]
接口功能：	
Method： hmyv2_getAllValidatorInformation	**Redis：** 太复杂，待讨论
接口功能：	
Method： hmyv2_getAllValidatorInformationByBlockNumber	**Redis：** 太复杂，待讨论
接口功能：	
Method： hmyv2_getElectedValidatorAddresses	**Redis：** <set> electedValidatorAddresses [时间待商榷]
接口功能：	
Method： hmyv2_getValidatorInformation	**Redis：** 太复杂，待讨论
接口功能：	
Method： hmyv2_getCurrentUtilityMetrics	**Redis：** <hash> currentUtilityMetric:${address} [时间待商榷]
接口功能：	
Method： hmyv2_getMedianRawStakeSnapshot	**Redis：** 太复杂，待讨论
接口功能：	
Method： hmyv2_getStakingNetworkInfo	**Redis：** <set> stakingNetWorkInfo [时间待商榷]
接口功能：	
Method： hmyv2_getSuperCommittees	**Redis：** 太复杂，待讨论

3.2.4 Transaction

接口功能：	
Method： hmyv2_getCXReceiptByHash	**Redis：** 待讨论
接口功能：	
Method：	**Redis：**

hmyv2_getPendingCXReceipts	待讨论
接口功能：	
Method： hmyv2_resendCx	**Redis：** <hash> resendCx(field: address, value:data)
接口功能：	
Method： hmyv2_getPoolStats	**Redis：** <hash> poolState(field: key, value:data) 或者用 string 存储，设置一个过期时间
接口功能：	
Method： hmyv2_pendingStakingTransactions	**Redis：** <set> pendingStakingTransactions StakingTransactionInfo[时间待商榷]
接口功能：	
Method： hmyv2_pendingTransactions	**Redis：** <set> pendingTransactions transaction [时间待商榷]
接口功能：	
Method： hmyv2_getCurrentStakingErrorSink	**Redis：** <set> CurrentStakingErrorSink data [时间待商榷]
接口功能：	
Method： hmyv2_getStakingTransactionByBlockNumberAndIndex	**Redis：** <hash> stakingTx:${height}(field:index, Value:stakingTxInfo)
接口功能：	
Method： hmyv2_getStakingTransactionByBlockHashAndIndex	**Redis：** <hash> stakingTx:${blockHash}(field:index, Value:stakingTxInfo)
接口功能：	
Method： hmyv2_getStakingTransactionByHash	**Redis：** <hash> stakingTx(field:txHash, Value:stakingTxInfo)
接口功能：	
Method： hmyv2_sendRawStakingTransaction	**Redis：**
接口功能：	
Method： hmyv2_getCurrentTransactionErrorSink	**Redis：** <set> currentTransactionErrorSink data
接口功能：	
Method： hmyv2_getTransactionByBlockHashAndIndex	**Redis：** <hash> tx:${blockHash}(field:index, Value:txInfo)
接口功能：	

Method： hmyv2_getTransactionByBlockNumberAndIndex	**Redis：** <hash> tx:${height}(field:index, Value:txInfo)
接口功能：	
Method： hmyv2_getTransactionByHash	**Redis：** <hash> tx(field:txHash, Value:txInfo)
接口功能：	
Method： hmyv2_getTransactionReceipt	**Redis：** <hash> receipt(field:receiptHash, Value:receiptInfo)
接口功能：	
Method： hmyv2_sendRawTransaction	**Redis：**

3.3 Logger

待补充。

3.4 Monitor

待补充。