CERTIK

Security Assessment

Stride

Sept 4th, 2022

Table of Contents

Summary

Overview

Project Summary

Audit Summary

Vulnerability Summary

Audit Scope

System Overview

Findings

GLOBAL-02 : Using Deprecated Method

ABI-01 : Hard Coded Variable

CAA-01 : Discussion on The Calculation of Reinvestment

CAA-02 : Lack of Sanity Check - Unlimited Commission Rate

DEO-01 : Unnecessary Condition

DEO-02 : Discussion on The DepositRecord ID And DepositRecord Count

HOS-01 : Lack of Sanity Check - Remove Validator

HOS-02 : Lack of Check on HostZone denom

IBC-01 : Incorrect Variable Used In `if` Statement

IBC-02 : Missing Important Validation Step

MSB-01 : Incorrect Calculation In Validator Rebalancing

MSU-01 : Incorrect Error Message

SBU-03 : Redundant Code

SBU-04 : Some TODOs Are Not Implemented

SBU-05 : Inconsistent Comment And Code

SBU-06 : Lack Of Input/Output Validation

SBU-07 : Unhandled Return Value

SBU-08 : Unused Parameters

SBU-09 : Comparison to A Boolean Constant

SLS-01 : Lack of Sanity Check - Prefix of `HostZone.IBCDenom`

SLS-02 : Double check the usage of methods

SLS-03 : Redundant Alias For Imported Packages

SLU-01 : Incorrect Key Used

SLU-02 : Missing Return Statement

SLU-03 : Unreasonable Receiver Type

SLU-06 : Naming Optimization

SLU-07 : Hard Coded Sensitive Values

SLU-08 : Redundant Type Conversion

SLU-09 : SafeMath Not Used

UNB-01 : Incorrect Comparison Condition

VAI-01 : Lack of Sanity Check - Zero Divisor

Optimizations

GLOBAL-01 : Lack of Gas Implementation

GEO-01 : Discussion on `UserRedemptionRecordCount`

IBC-03 : Discussion The Potential Redundant Statements

MSM-01 : Incorrect Format 'verb' Used

SBU-01 : Using Method `Error()` to Print Information Logs

SBU-02 : Using Method `Debug()` to Print Debug Logs

SLU-04 : Typo

SLU-05 : Using Method `Info()` to Print Error Logs

Appendix

Disclaimer

About

Summary

This report has been prepared for Stride to discover issues and vulnerabilities in the source code of the Stride project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Stride
Platform	CosmosSDK
Language	Golang
Codebase	https://github.com/Stride-Labs/stride
Commit	0a3b653ff55df54acc2daf43958d899db75698b0

Audit Summary

Delivery Date	Sept 04, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	4	0	0	0	0	0	4
● Medium	3	0	0	0	0	0	3
● Minor	6	0	0	0	0	0	6
● Informational	18	0	0	2	0	0	16
● Discussion	0	0	0	0	0	0	0

Audit Scope

ID	Repo	File	SHA256 Checksum
ABI	Stride-Labs/stride	interchainquery/keeper/abci.go	b2007cdb887e79c1597d07d9f1492427f8d9ee5a1c663eb07223e44eab9d5dc0
KEN	Stride-Labs/stride	interchainquery/keeper/keeper.go	2f53dd0e7a6a90c155e0dc832ee55ffd804f7466e2ac5366adf355d342293371
MSR	Stride-Labs/stride	interchainquery/keeper/msg_server.go	5fe56115ef820721911f6d6d161c0472958cc5dee8d1b5e9754979cd1751306e
QUK	Stride-Labs/stride	interchainquery/keeper/queries.go	9857043b8d9662cd745977aae557eea4aff9ff1f895383dac208b29b3c167097
CAB	Stride-Labs/stride	interchainquery/types/callbacks.go	96d3310a46f434ade9ae2608d8f10356ae0dd116b8f4a237c4a1a504ddda8ae5
COC	Stride-Labs/stride	interchainquery/types/codec.go	86fb558eda027038ee365daeeae1433c8b5afb8e8325bf4c5f6dba6676cd181b
ERT	Stride-Labs/stride	interchainquery/types/error.go	1bd00495221f712b0f0f20770035a1bd3fda10be31df8bdce5cf333ca0659c1e
EVT	Stride-Labs/stride	interchainquery/types/events.go	34bbaa818c01bcacc7d81f7ef5e3b3da06cca6f3a0425c0c96eeea41e4780e4d
GEY	Stride-Labs/stride	interchainquery/types/genesis.go	101469868fc1b9a8e919c0cc1aba22acdf346f73d8a9d4ee568d5646050aa751
KET	Stride-Labs/stride	interchainquery/types/keys.go	cf3d755527cae0017066e04519591d1a51ad2ad48841ad142f2a540def12ac51
MSY	Stride-Labs/stride	interchainquery/types/msgs.go	837ec079e042471f2dfd93c3cdbb001d9c7592080e504ed162a25152623e8e4e
GET	Stride-Labs/stride	interchainquery/genesis.go	4f64d321e497e10b3817d2674cc7c98596a5429fda9f1b8b564d93aeb6336b24
HAL	Stride-Labs/stride	interchainquery/handler.go	a0642a06c9c72c0a920ec1be61942df1b1d9d3134fdb6feba679ba199a96780b
MOI	Stride-Labs/stride	interchainquery/module.go	d9ca80cfdccf28710ff4007bef54d5ed822e6eb58577db92a33fb743d3358912

ID	Repo	File	SHA256 Checksum
ADD	Stride-Labs/stride	stakeibc/simulation/add_validator.go	e7d151881462235499e1eac6b4955acfd52a3121a2b8561c41304eebf0a890e2
CHA	Stride-Labs/stride	stakeibc/simulation/change_validator_weight.go	d2843804fae91c8c2dd843d6302b42a7f4ea4f6300b6f8b77d11b6e8319c4cd1
CLA	Stride-Labs/stride	stakeibc/simulation/claim_undelegated_tokens.go	a2b86edfd56b90dc646a109a0b3375471061ad78f28c815f7ba6055060ca552c
DEE	Stride-Labs/stride	stakeibc/simulation/delete_validator.go	519b05baa94d3deb10436202a54d5c6f606f4c2020982e67625a9ea8ef15ddb4
LIQ	Stride-Labs/stride	stakeibc/simulation/liquid_stake.go	d4ee0a16bcdd7e2920fac7aa6f7b19f951f2ffcd495533c4f00dcc45517e7a73
REB	Stride-Labs/stride	stakeibc/simulation/rebalance_validators.go	58c440baca6abfa959a614f442406dba6bbdb5695579c4a32032b280c04ef0b2
SII	Stride-Labs/stride	stakeibc/simulation/simap.go	86328d6f3ea15cf67f4671699284aff7732c80645a8552296e811021dbc54997
ABS	Stride-Labs/stride	stakeibc/abci.go	afe0ec95af93ece81fa45ecd75c3e165e3f3067c4c4f221c9240a86deb6ef887
GEA	Stride-Labs/stride	stakeibc/genesis.go	cca42c407158e632e37276016270fe1521d65a9f8636e6bd87aec302d6f553c3
HAE	Stride-Labs/stride	stakeibc/handler.go	f1c778698ad60ee17346c2d29c42305048030b4c1a347c651e0c9cee9ace2f6d
MOS	Stride-Labs/stride	stakeibc/module.go	a5e72bc8ed51778a6578aaa57407726a514d061699264856152d05b685ca9715
MOB	Stride-Labs/stride	stakeibc/module_ibc.go	15d00b32cdbbe505f88f316488d769f2122d64730016c0ab0d247b67e745f102
CAA	Stride-Labs/stride	stakeibc/keeper/callbacks.go	ff67e68d5c6ea7c03a1455960a57709724de26460df09805dd9a41b04eafe78e
DEL	Stride-Labs/stride	stakeibc/keeper/delegation.go	6c1328854c5e147ba52ebbc918d7c2d4c7d227940d097bc1c9584a29c7c2805d
EPH	Stride-Labs/stride	stakeibc/keeper/epoch_tracker.go	229de44ebd31fb951f13a5ccebc46b860dcf3563a7d96dda3f6cb71a1dda1176

ID	Repo	File	SHA256 Checksum
GRK	Stride-Labs/stride	stakeibc/keeper/grpc_query.go	35f44a3f0ddab8aa8ce9ce204089dbd37891d78138a644366211be7ba1462304
GRL	Stride-Labs/stride	stakeibc/keeper/grpc_query_delegation.go	cb57962a0c9a954aa84abe9640fa24c69f06ab4989c64fe9b3f79ef21161cf0b
GRO	Stride-Labs/stride	stakeibc/keeper/grpc_query_epoch_tracker.go	a04f0daf78b27dd7fab6b830e7e04f8e69bb886ec11f72b262449274abc0fb79
GRH	Stride-Labs/stride	stakeibc/keeper/grpc_query_host_zone.go	96efd6b00554477c6e02f58e4992241c34fdb4740aa3fd6e3dc17d2a3d96d6fa
GRI	Stride-Labs/stride	stakeibc/keeper/grpc_query_ica_account.go	eaed538935fded7e215ab3ee73a5e1826ef2c3c64703e625865824d53f6090c8
GRM	Stride-Labs/stride	stakeibc/keeper/grpc_query_min_validator_requirements.go	92556f74ccd8076bba0f7748d8b128fbfb9cb223edc963670b729dc02393c9ca
GRA	Stride-Labs/stride	stakeibc/keeper/grpc_query_module_address.go	8ae631e519a9b2621ccf8b56662e1225a4b95ac2ab3acb6a0c30152539e5c7c9
GRT	Stride-Labs/stride	stakeibc/keeper/grpc_query_params.go	863d0a0bfa1b7beafb6ad1bcab6f4ca79bacd1924f44f5494bd14913ecd2c561
GRG	Stride-Labs/stride	stakeibc/keeper/grpc_query_register_ica.go	c0a18d90b269b8637ad861aa76b620ae1d38c32570e64ce125c3f5762c5adca2
GRV	Stride-Labs/stride	stakeibc/keeper/grpc_query_validator.go	e91fc8e313507b9d23b318cfe29748e0f607de44951a64c5538fca74c4f546f6
HOO	Stride-Labs/stride	stakeibc/keeper/hooks.go	72c9eeed89623eebfb449a5c70a838e16dba98649d6667b038d1d4e086f947d4
HOS	Stride-Labs/stride	stakeibc/keeper/host_zone.go	7636bd79f4ca03683237124c0698a89f379038fa47fee0e6b6896f88b58153bf
IBC	Stride-Labs/stride	stakeibc/keeper/ibc_handlers.go	e40a71e2d61ea86686ddcec5d94fda73cc8cc6b968edc9592407081c9bad44fb
ICA	Stride-Labs/stride	stakeibc/keeper/ica_account.go	87b9aa73b2ef23c48511a26e4a721541dc2462b5d551a0dbf1e384e756d25f7d
KEA	Stride-Labs/stride	stakeibc/keeper/keeper.go	032d71e81d85559a083da0fc7a660efc8325f11a0a4a838c6640ecc8e8978eb6

ID	Repo	File	SHA256 Checksum
MIN	Stride-Labs/stride	stakeibc/keeper/min_validator_requirements.go	66c913403ebd5b2be3f04fc0ee850c963c395539932c88759042ee0063cfe625
MSV	Stride-Labs/stride	stakeibc/keeper/msg_server.go	6b2593cb72444bd18e7a6684c688d19a99dd8c5f19e095a827bbda6786378587
MSA	Stride-Labs/stride	stakeibc/keeper/msg_server_add_validator.go	ea13c8108b1630e2765fe02cbad806cb716b936bac2c4822140919ebf82cea41
MSC	Stride-Labs/stride	stakeibc/keeper/msg_server_change_validator_weight.go	b296b335620117ad4c520a7dba341622e520b312bf039630f7bb2d749f288e4a
MSL	Stride-Labs/stride	stakeibc/keeper/msg_server_claim_undelegated_tokens.go	737cee79c66d927a0a34dd8ebe6e394b030994e9bdaf39f6d2b22fde52ff945c
MSD	Stride-Labs/stride	stakeibc/keeper/msg_server_delete_validator.go	adfbd21771e7a905447cc48e1f97ff7a35e85bae0ba5525bd64c31dd2a0c4b56
MSI	Stride-Labs/stride	stakeibc/keeper/msg_server_liquid_stake.go	7ba79f6eccb2622a0f564d773cf31fb7d3b8ccd06df104b366ba2d6a08b426ad
MSB	Stride-Labs/stride	stakeibc/keeper/msg_server_rebalance_validators.go	1fc1395db534b33e3e44e5a80081fd99134b5cb2787920d722dffe8fd67a68f6
MSM	Stride-Labs/stride	stakeibc/keeper/msg_server_redeem_stake.go	f7b653b57bf990ad8818bca09ebd4222139278a413f002ca1c75c50505d6cd6a
MSH	Stride-Labs/stride	stakeibc/keeper/msg_server_register_host_zone.go	96c4414307d3212c5a4a2052bea48f590256c3281af54393b08619832039df31
MSK	Stride-Labs/stride	stakeibc/keeper/msg_server_register_ica.go	752855d2f1a15cf6fb6fb2d5a11ec8e20476913351e9ef03df4b8237dae39868
MSU	Stride-Labs/stride	stakeibc/keeper/msg_server_submit_tx.go	e8cea8c2b4a998ee4239e962ed90e777b16b72ed7342d9614c6af8da41a13a4c
PAS	Stride-Labs/stride	stakeibc/keeper/params.go	cc70135f26a221620d31f80d2bb2df06317d2b3b090c16a4a613898e41c449a0
UNB	Stride-Labs/stride	stakeibc/keeper/unbonding_records.go	c0000b218515d9b1e041516866f86f1747a14313377bce341ba4ac0d27a22407
VAL	Stride-Labs/stride	stakeibc/keeper/validator.go	d1b77f5625d9340122f86b1f3bf39031f1ff032b7402aa5a981cea4340e42e97

ID	Repo	File	SHA256 Checksum
VAI	Stride-Labs/stride	stakeibc/keeper/validator_selection.go	28fca25277f63dda6127ad1fe7d91348f1b82ebf982311543af53ca4b1fe74c9
COT	Stride-Labs/stride	stakeibc/types/codec.go	be8b15373451399f686ac2bbff157d9e1774effa588c82c913360c43f0d2aeb5
ERS	Stride-Labs/stride	stakeibc/types/errors.go	974e8007a582511a4a7b5f7dc856fabee6d25d8c0e32486d3d11089b04483f6b
EVS	Stride-Labs/stride	stakeibc/types/events_ibc.go	458a4ea8f96c5c447b90998d426a8a57b75dca3de65f06ad6fc38f273d617e2b
EXE	Stride-Labs/stride	stakeibc/types/expected_keepers.go	f717f2cee79272b965db61cca6e551f63bd524a260b653eb20998c9a27e5a231
GEP	Stride-Labs/stride	stakeibc/types/genesis.go	6f08bdd26673b0f26494a510e6691ecb2560212126042396cdbb129694a1c671
ICC	Stride-Labs/stride	stakeibc/types/ica_account.go	e9d29d9d7a24ffee27df80d4aa219a68dfd34273f2c4bd370eb77b31bed1d8ac
KEO	Stride-Labs/stride	stakeibc/types/key_epoch_tracker.go	8b28775f44bb47afb85f04ad89d0b4f6cd9df2fe5fad0520b5b9bfcc5677a58a
KEB	Stride-Labs/stride	stakeibc/types/keys.go	4479bb43690c2a827e117bddbffb6a2b9eb6c54a4f7838cbaea7696470f031ac
MEG	Stride-Labs/stride	stakeibc/types/message_add_validator.go	08bb7310b70ae3fb3a028ad6386c5f4478b07524f3ebb7ad7128d609fe74429c
MEE	Stride-Labs/stride	stakeibc/types/message_change_validator_weight.go	16e0a782630af121e150ec3c02b1c7c3edad1da46ca9d250e435655b1d045292
MEC	Stride-Labs/stride	stakeibc/types/message_claim_undelegated_tokens.go	c20dbe3e35f05a8e8022632b3f4d76953adb9f7fcb5bf92efca78a34f1821fcf
MED	Stride-Labs/stride	stakeibc/types/message_delete_validator.go	7ac40b256f60c19d4675bbec15d431bd282889f09e44ee382aed831050a10340
MEL	Stride-Labs/stride	stakeibc/types/message_liquid_stake.go	3e0e4eee5336190d671717efd2cb5b4d2dd74d7f59596824c4c663073c84e1ec
MER	Stride-Labs/stride	stakeibc/types/message_rebalance_validators.go	d2dddfbc0b7e734a699314a3962e9c6c3f1a396fc0b1a2715337c0d760738fbb

ID	Repo	File	SHA256 Checksum
MEM	Stride-Labs/stride	stakeibc/types/message_redeem_stake.go	736e676450cb5f039a3117882efb7f59968b9ff9636b1336913af59b9811ccae
MEI	Stride-Labs/stride	stakeibc/types/message_register_host_zone.go	2b2fb0edad883a5b9962a4b251361d670fc86326512d456973ced265f89e7ec4
MET	Stride-Labs/stride	stakeibc/types/message_register_ica.go	3a63f2bbdac86ce103e45908cd3410f5c45c582809bb1d097a2df7de287838a6
MEU	Stride-Labs/stride	stakeibc/types/message_submit_tx.go	8f6c1582e6ee07a04df5105506072e9683bec7e9cb51caf83898799eef6cf7c5
PAT	Stride-Labs/stride	stakeibc/types/params.go	25508b3a4ce283f815cf01ec85d0913c23daa31c5e0ce996b46294a30de9d489
QUX	Stride-Labs/stride	stakeibc/types/query_register_ica.go	7f984630c628cd78187c7898627595b172cbccb1d184ad7951bcac5bac9c352a
TYS	Stride-Labs/stride	stakeibc/types/types.go	7c347886dbeed39a02f9f23d860ffb46fa1da70151c2268a6289325c55acf415
QUL	Stride-Labs/stride	stakeibc/client/cli/query.go	f6515b1db1185a66bbd243e1a3818e308fefbea4b68d7ef79ff85f105a861292
QUN	Stride-Labs/stride	stakeibc/client/cli/query_delegation.go	bb1a843e7d7654dead4951463323ec5560afa8e46eee544148d19b9020c64f20
QU0	Stride-Labs/stride	stakeibc/client/cli/query_epoch_tracker.go	4299f4d8bf9709c06e3fae048d47bd91bde4c4ef68929032f6fb92bec41c3dae
QUZ	Stride-Labs/stride	stakeibc/client/cli/query_host_zone.go	dbf379c27060a764cdcfaed231a98708f4924902487c6cad8d01e12783ba5ffe
QU3	Stride-Labs/stride	stakeibc/client/cli/query_ica_account.go	3f38f29a0853e47da04cea554865bcd07122fb5b290c05adb955b473eed45ff5
QUQ	Stride-Labs/stride	stakeibc/client/cli/query_min_validator_requirements.go	d75654071f5dd2db6b7825cf94d9741a05c40b2bfa4c1e5488d3fa9ed4f56388
QU6	Stride-Labs/stride	stakeibc/client/cli/query_module_address.go	abfa58a1b0b81661957be1367800956a2a07669d3806ffe7afb61d88c73e5ae1
QU5	Stride-Labs/stride	stakeibc/client/cli/query_params.go	3732f22aabec7ac8623e7a22e7af4671209b89dcd5f6e8ae4153307c863b1b26

ID	Repo	File	SHA256 Checksum
QUF	Stride-Labs/stride	stakeibc/client/cli/query_register_ica.go	8ceb2a5b9a84d6163e010e35fd0bbf14685c052e0524f63c1195c1bb4ec186fa
QU4	Stride-Labs/stride	stakeibc/client/cli/query_validator.go	4294893162707b68330a5e92ccca753f533a0220668786d3a5cb4a9bd75dd2ca
TXT	Stride-Labs/stride	stakeibc/client/cli/tx.go	f1c564352e5a362acad8e8126a04e8cff343beb02852bdf1477d16388bcc24e0
TXV	Stride-Labs/stride	stakeibc/client/cli/tx_add_validator.go	cc3c60b8761072a6df5824eb0fc86eb620875c255e6830090ee528d1a9a32c8a
TXO	Stride-Labs/stride	stakeibc/client/cli/tx_change_validator_weight.go	c6b07bacefd3752d4cf5bd73d952babdd09a32d9de62081daeb7fbcf72db5710
TXM	Stride-Labs/stride	stakeibc/client/cli/tx_claim_undelegated_tokens.go	b92bc0d8fdbcf116451e43fcd69f2512d51702873ce6a38409d22c9687f9f9ca
TXK	Stride-Labs/stride	stakeibc/client/cli/tx_delete_validator.go	7fc10d4ce5caa493fd7bd3d9dbdd947e51342c69ea975f654940e74f13ffdc78
TXQ	Stride-Labs/stride	stakeibc/client/cli/tx_liquid_stake.go	fe729c97abb5a018faf21aecfda55ee327f91980ffad18a01154058f545da517
TXB	Stride-Labs/stride	stakeibc/client/cli/tx_rebalance_validators.go	2875e2f591cf757fae7976168b5c5ea4923c8758f7487558d301287ee4979c36
TXX	Stride-Labs/stride	stakeibc/client/cli/tx_redeem_stake.go	9102f25db461ccf878f4bb9f6231ef788e6c5ee7fa8decea5af7bb25a33abb0a
TXZ	Stride-Labs/stride	stakeibc/client/cli/tx_register_host_zone.go	2437f0aeba52fa834943d6532dc9b84cdbecc5a96a216ac792537171beb06018
TX0	Stride-Labs/stride	stakeibc/client/cli/tx_register_ica.go	5d96e198bbb2c89752c5fcae19a06112be44c35f17a69642ec2f7712e29d0cd3
TX3	Stride-Labs/stride	stakeibc/client/cli/tx_submit_tx.go	1f73c6f0bb03a8c9cc78b1c8c2d5f378f08e8b3a5f1da43aa4251fc38839ea5d
COY	Stride-Labs/stride	records/types/codec.go	ad9d77539aec89fe1ae23d12daf5573a1c85d649141cb8812f9e709ba5b6258d
ERY	Stride-Labs/stride	records/types/errors.go	38b216227cb7b59b7a42d200fff468d824dbda2fde09d4156c768b46c684411a

ID	Repo	File	SHA256 Checksum
EVI	Stride-Labs/stride	records/types/events_ibc.go	f74316c1f8438aa40bf66b152ecc624d218ee162d0e04e9b95be9a11d7471781
EXC	Stride-Labs/stride	records/types/expected_keepers.go	ea7436f7a9ae2c562f520ef7ac62fab2133e88726ef569e1f52a91a69778808d
GER	Stride-Labs/stride	records/types/genesis.go	43a5b9b8f7e9a66ec76b1f15fd29bb99e2be25f02e3ba8289f4292f20413c78e
KEC	Stride-Labs/stride	records/types/keys.go	fc388e532a07594109a981fce05650f1f8763eba42a20da661e4e3b2763e283f
PAY	Stride-Labs/stride	records/types/params.go	4bc10cd1ab904b638aeaffa3b92a56cb75cfb2a8aa06006ba7d3983fb3b0b89a
TYT	Stride-Labs/stride	records/types/types.go	7c347886dbeed39a02f9f23d860ffb46fa1da70151c2268a6289325c55acf415
SIT	Stride-Labs/stride	records/simulation/simap.go	86328d6f3ea15cf67f4671699284aff7732c80645a8552296e811021dbc54997
GEO	Stride-Labs/stride	records/genesis.go	efb27a6ae46e169a3932072a500d89d2990ed89cff900e12e7ba6baa2a696bc8
HAR	Stride-Labs/stride	records/handler.go	494e72ceecdbe24cd280adb921096ea45e74c2002f56f573834777c7e4b1634e
MOR	Stride-Labs/stride	records/module.go	7f0d0e807489bc5a3e897d9dfa3e96a69f01ace1014f76c4175e046470cf9fac
MOC	Stride-Labs/stride	records/module_ibc.go	e514b4c3775e706e9d2d876475d45b3ab26801bd777faa4cdaf2d4035afa9a34
MOM	Stride-Labs/stride	records/module_simulation.go	11eab50dd1c15b01ed37ebcc702dbac0c4c223e5287f851e8c2f25ea53b0b3d6
DEO	Stride-Labs/stride	records/keeper/deposit_record.go	501e1f7a998734877b5486e3bfac0802acaa238640c10fb6f7d1d19275d26d58
EPU	Stride-Labs/stride	records/keeper/epoch_unbonding_record.go	6e24399ca0cca36694ddf3a565701c8ddcf4d46d10499b4215865b80eb9d9141
GRX	Stride-Labs/stride	records/keeper/grpc_query.go	24215842d569558d91d972457e510bf814c08125d5ddaeb3909c5713cefde99b

ID	Repo	File	SHA256 Checksum
GR0	Stride-Labs/stride	records/keeper/grpc_query_deposit_record.go	06dfa56015b0b880d60f5c0c93ed8b795ca8ff77b706275aefe596c662cd6ad4
GRN	Stride-Labs/stride	records/keeper/grpc_query_epoch_unbonding_record.go	74bf69c7cc06c96221b2426bb98ab97e19fdcacd12d4070ef7274e4491b9a4e6
GR3	Stride-Labs/stride	records/keeper/grpc_query_params.go	7c15b1536bfbb01e014cbee2c8578d54363bb778b51655430c97a84c11df67dc
GRB	Stride-Labs/stride	records/keeper/grpc_query_user_redemption_record.go	ff4c62c04080db2123b4429c7beedc312167f9e57948afae861ee524ac82a657
KED	Stride-Labs/stride	records/keeper/keeper.go	028d57465ca93128b3d4b7d49a6c288f251adf5cf920536738fcf60cbcaeed4a
MSP	Stride-Labs/stride	records/keeper/msg_server.go	f7304af7def72b199978a570a55aa8db0ad9e6801f4fd614c7f52cf953b02a06
PAK	Stride-Labs/stride	records/keeper/params.go	1bdbc8e79ad71d29b77a1a82c5a62117c5d127e132e58d05472eb01c894f3b7b
USD	Stride-Labs/stride	records/keeper/user_redemption_record.go	2a9e6b0f1e3d3b79eb05a6271c480e6a5a083a1839b209e024deaf18369c0eaf
QU9	Stride-Labs/stride	records/client/cli/query.go	f1ce83285098b65b669580c2509f9eb6b34dc1211e5c8d95dc21bad2700bad56
QU8	Stride-Labs/stride	records/client/cli/query_deposit_record.go	b433772b074c4e91b50db8315e964e6c5473ff81aab7a66d8d7203bb50ab86f4
QU7	Stride-Labs/stride	records/client/cli/query_epoch_unbonding_record.go	d826af22fa6399a1ebb6ad95cced95fb2a4107dec08d95569286db99eaef079c
QER	Stride-Labs/stride	records/client/cli/query_params.go	5275f651434688776828f34ddc0a410de571b2e6c7cdf241c494b94850c3e994
QEY	Stride-Labs/stride	records/client/cli/query_user_redemption_record.go	00dc981ea20b74ce57c48dc4f9aad8a5058c39e5c0e33d4583cefd4f4594441f
TX6	Stride-Labs/stride	records/client/cli/tx.go	86b5f1bee09be6e37d50d096dec1552c2aaaff054ab2c2d64ab3df567fb917df

System Overview

Stride aims to provide a state-of-the-art multichain liquid staking experience where users can liquid stake their assets on any Cosmos chain. The project aims to resolve the problem that users are forced to choose between the rewards offered from staking and/or the yields offered from DeFi protocols. Staking tokens secure the network and earn passive yield, but requires users to lock up tokens; participating in DeFi lets users get a higher yield, but exposes users to more risk.

Strider works the following three steps.

1. Stake tokens on Stride from any Cosmos chain, and get rewards accumulated in real-time.
2. Users receive `stTokens` when they stake. These can be freely traded and redeemed with Stride at any time to receive original tokens.
3. Stride lets users use staked assets to compound yield.

Diagrams

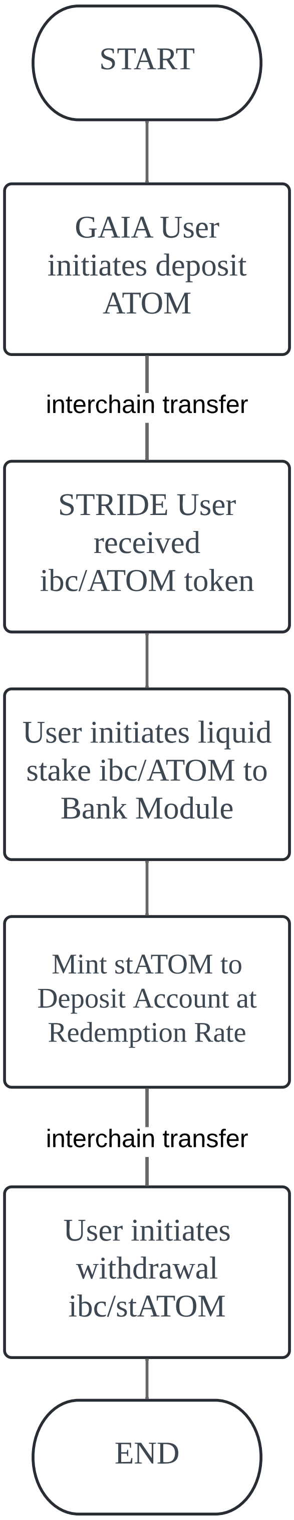
Flowchart

These flowcharts describe the separate steps of the process in this project, and what accounts will be flowed to, and how to reap compound interest.

Deposit and Liquid Stake

Users stake tokens on Stride from another Cosmos chain and then receive `stToken`s and rewards that accumulate in real-time.

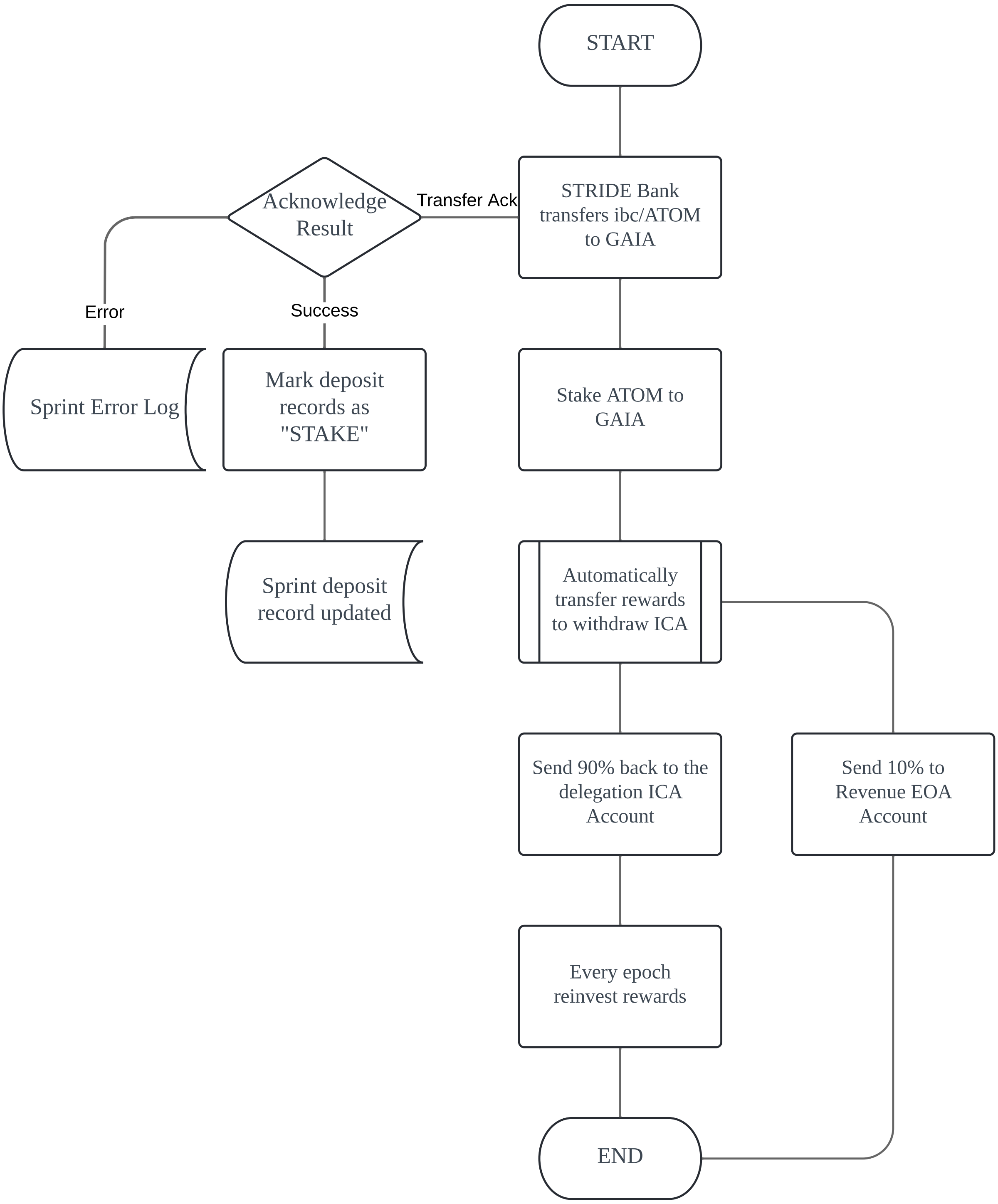
1. In each epoch, all the deposits will be recorded in one record, it will just update the total amount.
2. The status of the record will be `recordstypes.DepositRecord_TRANSFER`.
3. The total amount before a certain epoch will be accumulated that would be named `UndelegatedBalance`


START
GAIA User initiates deposit ATOM
interchain transfer
STRIDE User received ibc/ATOM token
User initiates liquid stake ibc/ATOM to Bank Module
Mint stATOM to Deposit Account at Redemption Rate
interchain transfer
User initiates withdrawal ibc/stATOM
END

$$RedemptionRate = \frac{(UnstakedATOM + StakedATOM + StrideModuleAccountBalance)}{stATOM}$$

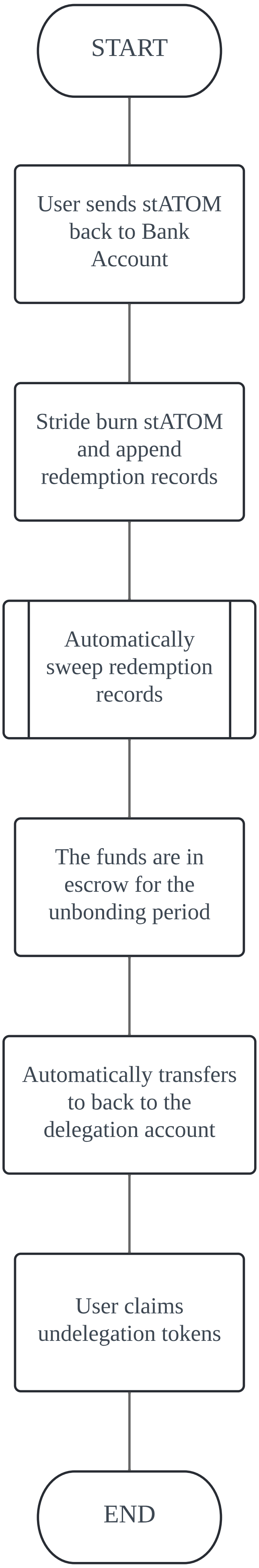
Epoch Delegation and Reinvestment

Stride automatically stakes and reinvests the liquid tokens and staked rewards at every epoch.


START
STRIDE Bank transfers ibc/ATOM to GAIA
Transfer Ack
Acknowledge Result
Error
Success
Sprint Error Log
Mark deposit records as "STAKE"
Sprint deposit record updated
Stake ATOM to GAIA
Automatically transfer rewards to withdraw ICA
Send 90% back to the delegation ICA Account
Send 10% to Revenue EOA Account
Every epoch reinvest rewards
END

Unbonding

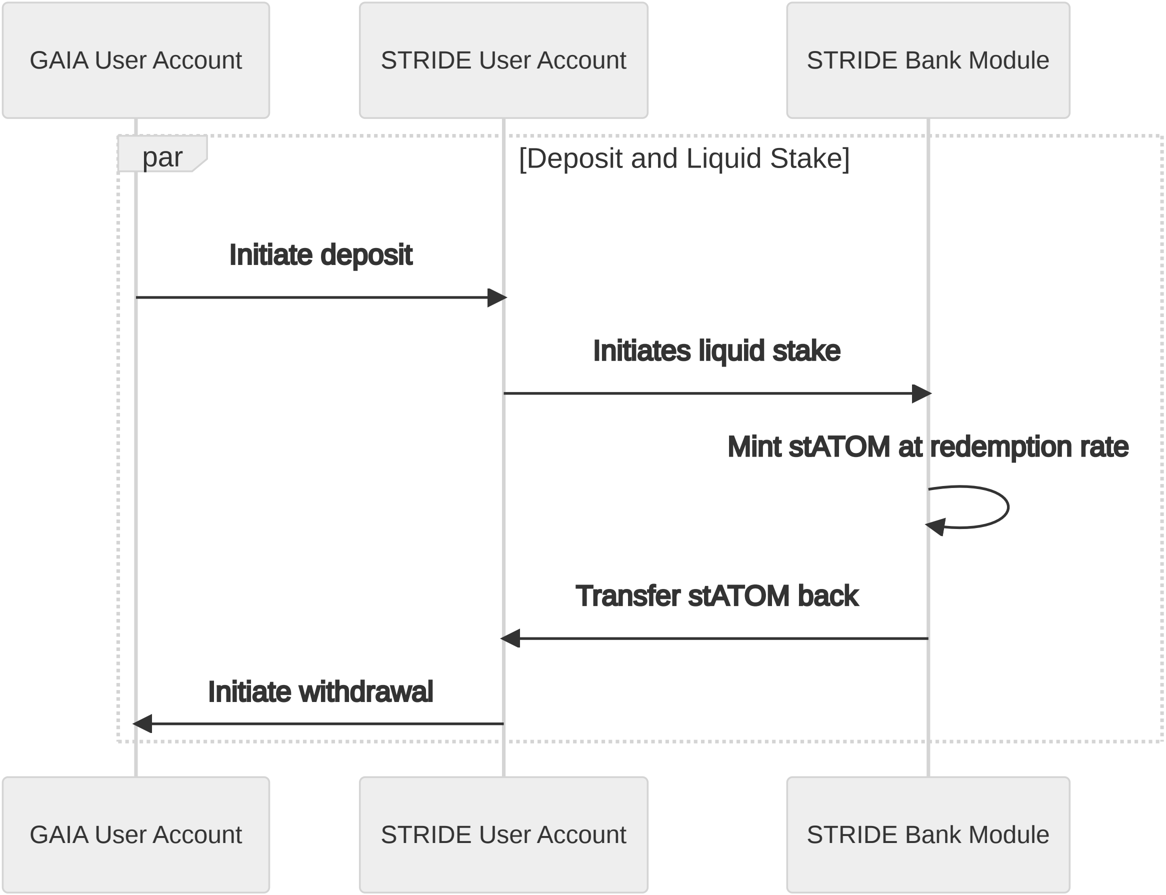
Users redeem the staked tokens from the `redemption account` in the HostZone by burning `stToken`. This process updates the `userRedemptionRecord` and `epochUnbondingRecord`. Users do not get the local token directly, triggered by `msg.ClaimUndelegatedTokens`, which is further processed at `ibc_handles.go/HandleSend()`.


START
User sends stATOM back to Bank Account
Stride burn stATOM and append redemption records
Automatically sweep redemption records
The funds are in escrow for the unbonding period
Automatically transfers to back to the delegation account
User claims undelegation tokens
END

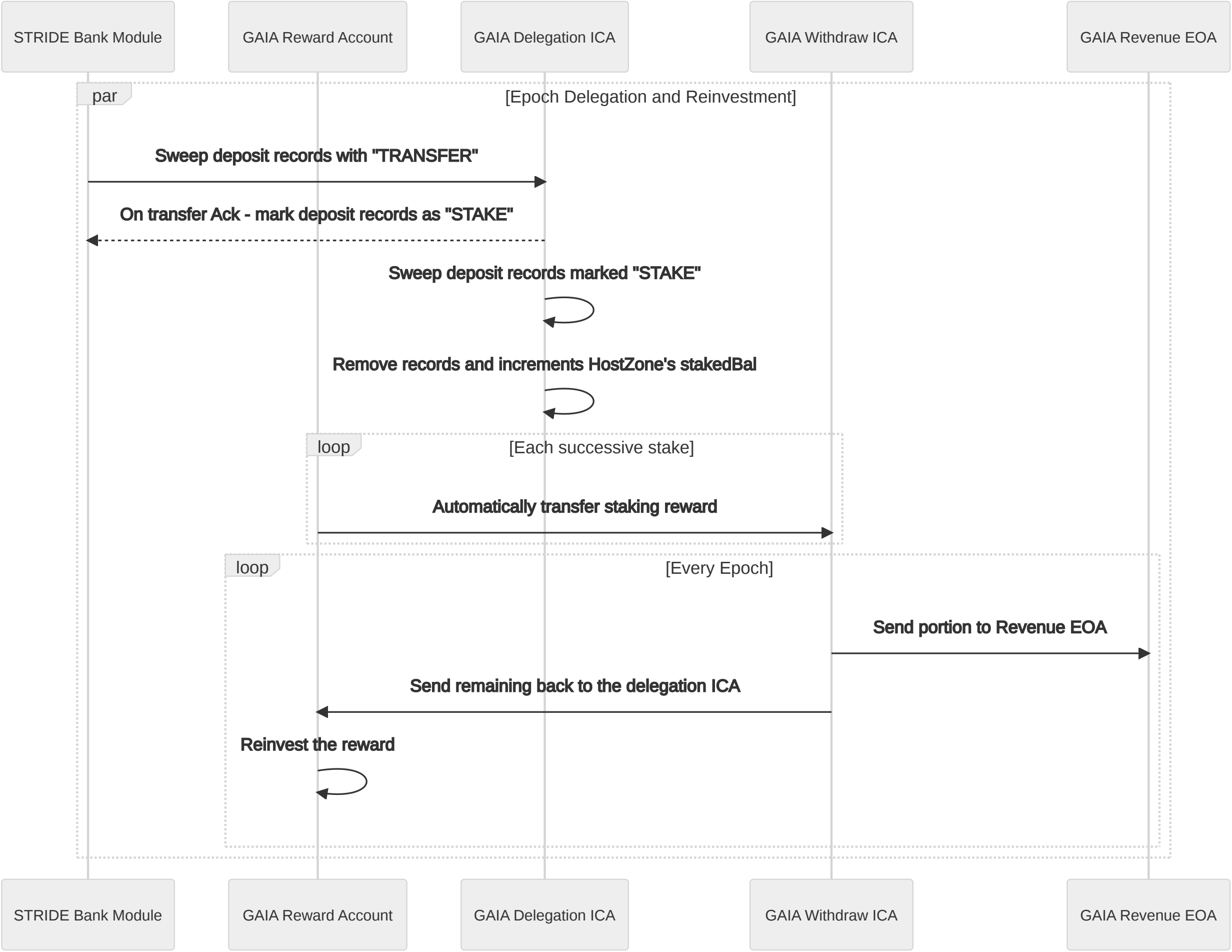
Instructions Sequence

The following graphs describe the flow of tokens in each transaction. And the GAIA ATOM is an example that has been supported by Stride now. After launch, Stride plans on rapidly expanding its reach throughout the Cosmos ecosystem.

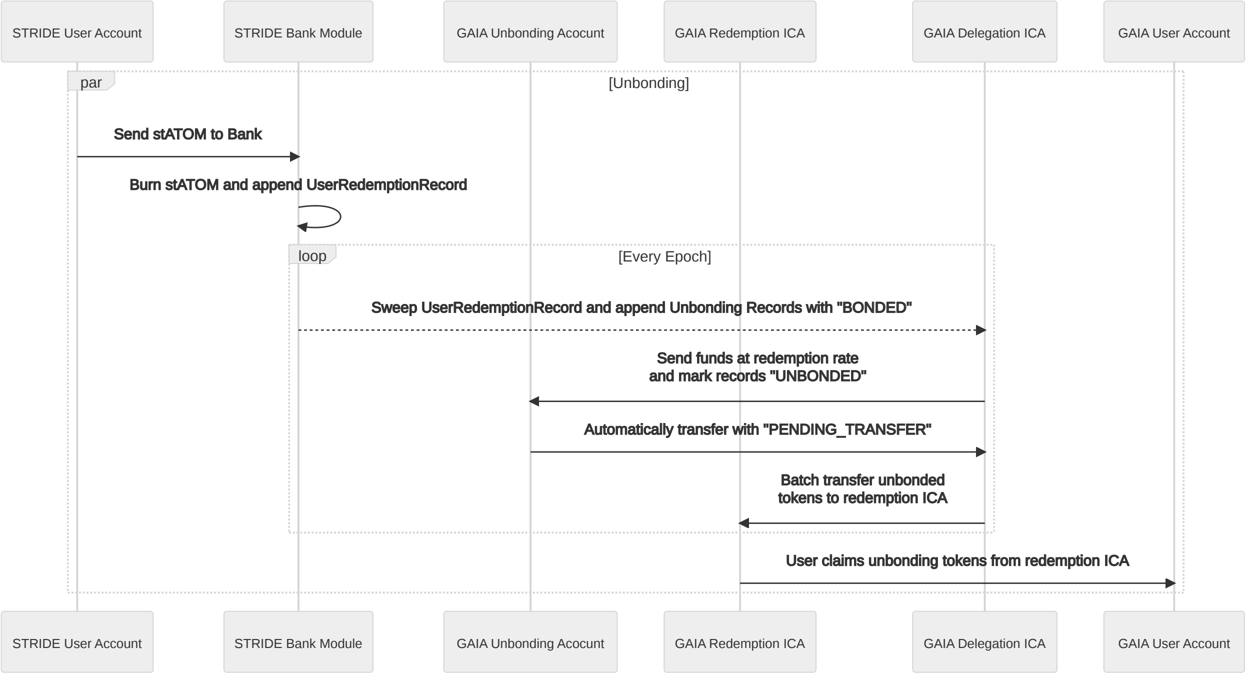
Deposit and Liquid Stake



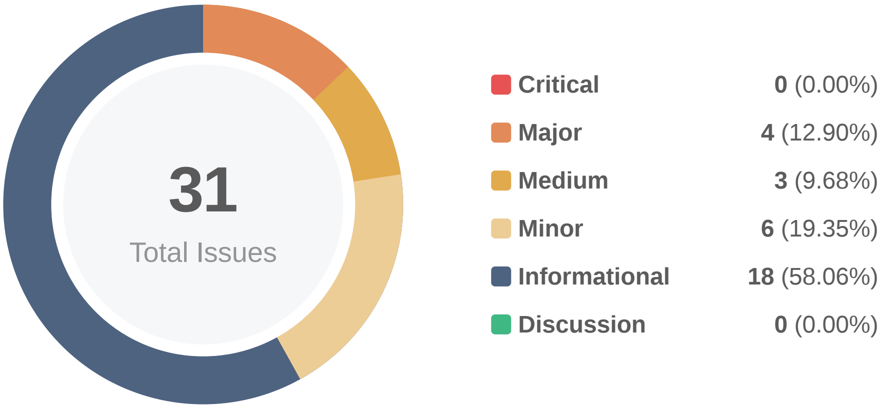
Epoch Delegation and Reinvestment


STRIDE Bank Module
GAIA Reward Account
GAIA Delegation ICA
GAIA Withdraw ICA
GAIA Revenue EOA
par
[Epoch Delegation and Reinvestment]
Sweep deposit records with "TRANSFER"
On transfer Ack - mark deposit records as "STAKE"
Sweep deposit records marked "STAKE"
Remove records and increments HostZone's stakedBal
loop
[Each successive stake]
Automatically transfer staking reward
loop
[Every Epoch]
Send portion to Revenue EOA
Send remaining back to the delegation ICA
Reinvest the reward
STRIDE Bank Module
GAIA Reward Account
GAIA Delegation ICA
GAIA Withdraw ICA
GAIA Revenue EOA

Unbonding


STRIDE User Account
STRIDE Bank Module
GAIA Unbonding Acocunt
GAIA Redemption ICA
GAIA Delegation ICA
GAIA User Account
par
[Unbonding]
Send stATOM to Bank
Burn stATOM and append UserRedemptionRecord
loop
[Every Epoch]
Sweep UserRedemptionRecord and append Unbonding Records with "BONDED"
Send funds at redemption rate
and mark records "UNBONDED"
Automatically transfer with "PENDING_TRANSFER"
Batch transfer unbonded
tokens to redemption ICA
User claims unbonding tokens from redemption ICA
STRIDE User Account
STRIDE Bank Module
GAIA Unbonding Acocunt
GAIA Redemption ICA
GAIA Delegation ICA
GAIA User Account

Findings



ID	Title	Category	Severity	Status
GLOBAL-02	Using Deprecated Method	Volatile Code	Informational	Resolved
ABI-01	Hard Coded Variable	Logical Issue	Informational	Resolved
CAA-01	Discussion On The Calculation Of Reinvestment	Volatile Code	Minor	Resolved
CAA-02	Lack Of Sanity Check - Unlimited Commission Rate	Volatile Code	Minor	Resolved
DEO-01	Unnecessary Condition	Coding Style	Informational	Resolved
DEO-02	Discussion On The DepositRecord ID And DepositRecord Count	Logical Issue	Informational	Acknowledged
HOS-01	Lack Of Sanity Check - Remove Validator	Logical Issue	Medium	Resolved
HOS-02	Lack Of Check On HostZone Denom	Volatile Code	Medium	Resolved
IBC-01	Incorrect Variable Used In `if` Statement	Logical Issue	Major	Resolved
IBC-02	Missing Important Validation Step	Volatile Code	Major	Resolved
MSB-01	Incorrect Calculation In Validator Rebalancing	Mathematical Operations	Major	Resolved
MSU-01	Incorrect Error Message	Coding Style	Informational	Resolved
SBU-03	Redundant Code	Coding Style	Informational	Resolved

ID	Title	Category	Severity	Status
SBU-04	Some TODOs Are Not Implemented	Control Flow	Informational	Resolved
SBU-05	Inconsistent Comment And Code	Inconsistency	Informational	Resolved
SBU-06	Lack Of Input/Output Validation	Volatile Code	Informational	Resolved
SBU-07	Unhandled Return Value	Logical Issue	Informational	Resolved
SBU-08	Unused Parameters	Coding Style	Informational	Resolved
SBU-09	Comparison To A Boolean Constant	Language Specific	Informational	Resolved
SLS-01	Lack Of Sanity Check - Prefix Of `HostZone.IBCDenom`	Volatile Code	Minor	Resolved
SLS-02	Double Check The Usage Of Methods	Control Flow	Informational	Resolved
SLS-03	Redundant Alias For Imported Packages	Coding Style	Informational	Resolved
SLU-01	Incorrect Key Used	Logical Issue	Major	Resolved
SLU-02	Missing Return Statement	Logical Issue	Medium	Resolved
SLU-03	Unreasonable Receiver Type	Volatile Code	Minor	Resolved
SLU-06	Naming Optimization	Coding Style	Informational	Resolved
SLU-07	Hard Coded Sensitive Values	Control Flow	Informational	Resolved
SLU-08	Redundant Type Conversion	Coding Style	Informational	Resolved
SLU-09	SafeMath Not Used	Mathematical Operations	Informational	Acknowledged
UNB-01	Incorrect Comparison Condition	Logical Issue	Minor	Resolved
VAI-01	Lack Of Sanity Check - Zero Divisor	Volatile Code	Minor	Resolved

GLOBAL-02 | Using Deprecated Method

Category	Severity	Location	Status
Volatile Code	● Informational		✓ Resolved

Description

We found that the method EmitEvents() is deprecated in CosmosSDK.

Reference : The code of `EmitEvents`

Recommendation

We recommend using the new API provided in the comment.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 7ded2ac332cf11b71ebbf3801ad58d67939cc9cc.

ABI-01 | Hard Coded Variable

Category	Severity	Location	Status
Logical Issue	● Informational	interchainquery/keeper/abci.go: 35	✓ Resolved

Description

In this function `EndBlocker()`, queryInfo.height is hardcoded as 0, which may be from input argument.

```
sdk.NewAttribute(types.AttributeKeyHeight, "0"),
```

Recommendation

We recommend double-checking this hardcoded argument to ensure it meets the design intent.

Alleviation

`[Stride]`: This is intentional. Passing “0” to interchainquery’s height makes the interchainquery query at the latest height on the HostZone. For more detail sees PR.

CAA-01 | Discussion On The Calculation Of Reinvestment

Category	Severity	Location	Status
Volatile Code	● Minor	stakeibc/keeper/callbacks.go: 119~133	✓ Resolved

Description

The protocol should ensure all balances are fully claimed by `delegatonAccount` and `REV_ACCT`, therefore it may be more convenient to obtain the reinvestment value by using $withdrawalBalance - strideClaimFloored$.

Recommendation

We recommend reviewing above mentioned methods.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 4a85ff28d2c7350e51dd2d0d3f58f45126311ea5.

CAA-02 | Lack Of Sanity Check - Unlimited Commission Rate

Category	Severity	Location	Status
Volatile Code	● Minor	stakeibc/keeper/callbacks.go: 119	✓ Resolved

Description

The variable `strideCommission` is used to store the stride's commission rate and if the value of `strideCommission` is set too large, uses would lose revenue.

Recommendation

We recommend client to add a more reasonable range check about `strideCommission`, and record this rate limit in the white paper.

Alleviation

`[Stride]`: This is not an issue. This is subjective and controlled through governance. Users might want to increase this down the line as the tokenomics change.

DEO-01 | Unnecessary Condition

Category	Severity	Location	Status
Coding Style	● Informational	records/keeper/deposit_record.go: 124~127	✓ Resolved

Description

The value of variable `hostZoneMatches` will always be true.

Recommendation

We recommend removing the `hostZoneMatches` from `if` condition.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit [752d48b355cdcb0986f46af3b3ef921666be4057].

DEO-02 | Discussion On The DepositRecord ID And DepositRecord Count

Category	Severity	Location	Status
Logical Issue	● Informational	records/keeper/deposit_record.go: 74~78	ⓘ Acknowledged

Description

Since the deposit record id and the deposit record count are stored with the same prefix `DepositRecordCountKey = "DepositRecord-count-"`. The RemoveDepositRecord() method will not reduce the DepositRecord count. Therefore, the `GetDepositRecordCount()` method will return a value greater than actual value.

We understand that the current solution is simple and easy to handle. But the team should be aware of this when using `GetDepositRecordCount()` method.

There is a similar logic in `epoch_unbonding_record.go`.

Recommendation

Consider revisiting above mentioned functions and variables.

Alleviation

Stride team acknowledged this finding.

HOS-01 | Lack Of Sanity Check - Remove Validator

Category	Severity	Location	Status
Logical Issue	● Medium	stakeibc/keeper/host_zone.go: 118~133	✓ Resolved

Description

When removing a validator with non-zero delegation amount from the set, those funds will be stranded until re-add this validator.

Recommendation

We recommend ensuring that `Validator.DelegationAmt` is ZERO before removing the target validator from a host zone.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 38bcb046587b69f0bed3e99995f8b84b18c8db41 and PR #139.

HOS-02 | Lack Of Check On HostZone Denom

Category	Severity	Location	Status
Volatile Code	● Medium	stakeibc/keeper/host_zone.go: 39~43	✓ Resolved

Description

According to the `GetHostZoneFromHostDenom()` method called by `LiquidStake()` method, the returned item should be a unique, specific HostZone. Therefore, we believe that the variable HostDenom should be uniquely validated when creating and modifying methods. In addition, the `GetHostZoneFromHostDenom()` method is called by the `MintStAsset()`, `HandleSend()`, `HandleDelegate()`, and `HandleUndelegate()` methods. All of these callers require a unique, specific HostZone.

- x/stakeibc/keeper/host_zone.go

```
// GetHostZoneFromHostDenom returns a HostZone from a HostDenom
func (k Keeper) GetHostZoneFromHostDenom(ctx sdk.Context, denom string) (*types.HostZone, error) {
  var matchZone types.HostZone
  inDenom := strings.ToUpper(denom)
  k.IterateHostZones(ctx, func(ctx sdk.Context, index int64, zoneInfo types.HostZone) error {
    zoneDenom := strings.ToUpper(zoneInfo.HostDenom)
    if zoneDenom == inDenom {
      matchZone = zoneInfo
      return nil
    }
    return nil
  })
  if matchZone.ChainId != "" {
    return &matchZone, nil
  }
  return nil, sdkerrors.Wrapf(sdkerrors.ErrUnknownRequest, "No HostZone for %s found", denom)
}
```

- x/stakeibc/keeper/msg_server_liquid_stake.go

```
func (k msgServer) LiquidStake(goCtx context.Context, msg *types.MsgLiquidStake) (*types.MsgLiquidStakeResponse, error) {
  ctx := sdk.UnwrapSDKContext(goCtx)

  // Init variables
  // deposit `amount` of `denom` token to the stakeibc module
  // NOTE: Should we add an additional check here? This is a pretty important line of code
  // NOTE: If sender doesn't have enough inCoin, this panics (error is hard to interpret)
  // check that hostZone is registered
  // strided tx stakeibc liquid-stake 100 uatom
  hostZone, err := k.GetHostZoneFromHostDenom(ctx, msg.HostDenom)
  if err != nil {
    k.Logger(ctx).Info("Host Zone not found for denom (%s)", msg.HostDenom)
    return nil, err
  }
  ......
```

Recommendation

We recommend adding validation to the `SetHostZone()` method to ensure that the `HostDenom` is unique.

Alleviation

`[Stride]`: The only place `hostZones` are created in `msg_register_host_zone`, so we added this check there, instead of doing the check-in `SetHostZone()`, because we can `SetHostZone()` often, so this iteration over all host zones will get expensive if there are many!

`[CertiK]`: Stride added a check on `HostDenom` in the `RegisterHostZone()` method in PR #142.

IBC-01 | Incorrect Variable Used In `if` Statement

Category	Severity	Location	Status
Logical Issue	● Major	stakeibc/keeper/ibc_handlers.go: 227	✓ Resolved

Description

In L227 in file `x/stakeibc/keeper/ibc_handler.go`, `epochUnbondingRecord.Id` is used to compared to the `dayEpochTracker.EpochNumber`. According to the design, it should be variable `epochUnbondingRecord.UnbondingEpochNumber` rather than `epochUnbondingRecord.Id`

Recommendation

We recommend correcting `epochUnbondingRecord.Id` to `epochUnbondingRecord.UnbondingEpochNumber` in both of `if` statements and messages in line 228.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit [1d779157131465086316bbb556381947224732d24].

IBC-02 | Missing Important Validation Step

Category	Severity	Location	Status
Volatile Code	● Major	stakeibc/keeper/ibc_handlers.go: 331~336	✓ Resolved

Description

The variable `undelegateAmt` ought to be positive, otherwise, the HostZone account may be in shortage. This is because a negative number would, unexpectedly, increase the value of HostZone StakedBal when the user undelegated his stAtom.

Recommendation

It is recommended to add a check to ensure that the variable `undelegateAmt` is greater than 0.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 1fb605dd07a1ef6092751c4e544acb1eb0b3b2f2.

MSB-01 | Incorrect Calculation In Validator Rebalancing

Category	Severity	Location	Status
Mathematical Operations	● Major	stakeibc/keeper/msg_server_rebalance_validators.go: 102	✓ Resolved

Description

According to the design, the variable `overWeightElem.deltaAmt` must be assigned with a negative value.

```
95    if overWeightElem.deltaAmt > 0 {
96      // if overWeightElem is positive, we're done rebalancing
97      break
98    }
```

In the process of rebalancing the delegation fund, the validator that is `underWeight` should get the delegation fund from the validator that is `overWeight`, and the value of `underWeightElem.deltaAmt` should be reduced in `underWeightElem.deltaAmt -= overWeightElem.deltaAmt`. But in the following calculation in `msg_server_rebalance_validators.go`(Line 102), the value of `underWeightElem.deltaAmt` will become greater if the value of `overWeightElem.deltaAmt` is correctly assigned with a negative value.

```
 99 if abs(underWeightElem.deltaAmt) > abs(overWeightElem.deltaAmt) {
100   // if the underweight element is more overweight than the overweight element
101   // we transfer stake from the underweight element to the overweight element
102   underWeightElem.deltaAmt -= overWeightElem.deltaAmt
103   overWeightIndex += 1
```

Recommendation

We recommend correcting the calculation in `msg_server_rebalance_validators.go`(Line 102) to

```
underWeightElem.deltaAmt -= abs(overWeightElem.deltaAmt)
```

or

```
underWeightElem.deltaAmt += overWeightElem.deltaAmt
```

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 9be5af827148db820c57eb08ef8c5eee6eba52d1.

MSU-01 | Incorrect Error Message

Category	Severity	Location	Status
Coding Style	● Informational	stakeibc/keeper/msg_server_submit_tx.go: 151	✓ Resolved

Description

Incorrect error message, ICA `WithdrawalAccount` is missing here but the account's name in error message becomes "delegation address".

Recommendation

We recommend correcting the error message to “Zone %s is missing a withdrawal address!”

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 6dd11dc4ec07a0ab6add2796906adbb8d4a8c12a.

SBU-03 | Redundant Code

Category	Severity	Location	Status
Coding Style	● Informational	interchainquery/keeper/abci.go: 14; interchainquery/types/error.go: 6; interchainquery/types/events.go: 8; interchainquery/types/msgs.go: 16~21; records/keeper/deposit_record.go: 103~106; records/keeper/epoch_unbonding_record.go: 125~128; records/types/errors.go: 11, 12; records/types/events_ibc.go: 5, 8; records/types/keys.go: 37; stakeibc/keeper/host_zone.go: 135~138; stakeibc/keeper/msg_server_register_host_zone.go: 18; stakeibc/types/codec.go: 32; stakeibc/types/errors.go: 11, 12, 16, 23, 25; stakeibc/types/events_ibc.go: 5, 15, 16, 20, 21, 22, 23; stakeibc/types/message_liquid_stake.go: 32~34; stakeibc/types/message_redeem_stake.go: 23~27; stakeibc/types/params.go: 18	✓ Resolved

Description

The linked variables, constants and functions are never used, and some linked statements do not affect the functionality of the codebase.

The file paths are as follows :

1. Unused Variables/Constants
   - x/interchainquery/keeper/abci.go, #L14 ~ 14
   - x/interchainquery/types/error.go, #L6 ~ 6
   - x/interchainquery/types/events.go, #L8 ~ 8
   - x/records/types/errors.go, #L11 ~ 11
   - x/records/types/errors.go, #L12 ~ 12
   - x/records/types/events_ibc.go, #L5 ~ 5
   - x/records/types/events_ibc.go, #L8 ~ 8
   - x/records/types/keys.go, #L37 ~ 37
   - x/stakeibc/types/errors.go, #L11 ~ 11
   - x/stakeibc/types/errors.go, #L12 ~ 12
   - x/stakeibc/types/errors.go, #L16 ~ 16
   - x/stakeibc/types/errors.go, #L23 ~ 23
   - x/stakeibc/types/errors.go, #L25 ~ 25
   - x/stakeibc/types/events_ibc.go, #L5 ~ 5
   - x/stakeibc/types/events_ibc.go, #L15 ~ 15
   - x/stakeibc/types/events_ibc.go, #L16 ~ 16
   - x/stakeibc/types/events_ibc.go, #L20 ~ 20

  - x/stakeibc/types/events_ibc.go, #L21 ~ 21
  - x/stakeibc/types/events_ibc.go, #L22 ~ 22
  - x/stakeibc/types/events_ibc.go, #L23 ~ 23
  - x/stakeibc/types/params.go, #L18 ~ 18

2. Unused Functions
   - x/interchainquery/types/msgs.go, #L16 ~ 21
   - x/records/keeper/deposit_record.go, #L103 ~ 106
   - x/records/keeper/epoch_unbonding_record.go, #L125 ~ 128
   - x/stakeibc/keeper/host_zone.go, #L135 ~ 138
   - x/stakeibc/types/message_liquid_stake.go, #L32 ~ 34
   - x/stakeibc/types/message_redeem_stake.go, #L23 ~ 27
3. Unused Statements
   - x/stakeibc/keeper/msg_server_register_host_zone.go, #L18 ~ 18
   - x/stakeibc/types/codec.go, #L32 ~ 32

Recommendation

We recommend the client to remove them if there is no plan for further usage.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 6702ae89814f71c305d0f9a1c40e498eddaf7254 and 067d66089e4508a966add4bfa075a2a4cdf8658a.

SBU-04 | Some TODOs Are Not Implemented

Category	Severity	Location	Status
Control Flow	● Informational	interchainquery/types/genesis.go: 15; interchainquery/types/msgs.go: 30~39; stakeibc/types: 57; stakeibc/types/message_add_validator.go: 45; stakeibc/types/message_change_validator_weight.go: 43; stakeibc/types/message_claim_undelegated_tokens.go: 41; stakeibc/types/message_liquid_stake.go: 57; stakeibc/types/message_register_host_zone.go: 45; stakeibc/types/message_submit_tx.go: 89	✓ Resolved

Description

The listed todos in `Validate()` and `ValidateBasic()` in the finding locations are not implemented.

Recommendation

Consider adding more checks and removing unnecessary todo tags.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 939fb5a53651cac3e2a9fbdafc893aa0d72c0dfa and b3bb1b414275ada7f946216aae6f9edc068a43d4.

SBU-05 | Inconsistent Comment And Code

Category	Severity	Location	Status
Inconsistency	● Informational	interchainquery/keeper/keeper.go: 86~88; records/keeper/epoch_unbonding_record.go: 84~86, 130~132; records/keeper/user_redemption_record.go: 49~50; stakeibc/keeper/host_zone.go: 178~180; stakeibc/keeper/msg_server_register_ica.go: 23~24; stakeibc/types/expected_keepers.go: 15~20	✓ Resolved

Description

The comment is inconsistent with the code.

In `x/records/keeper/user_redemption_record.go` :

Comment in line 49 is the doc comment for method `IterateUserRedemptionRecords()`, therefore the comment should have the following format **"IterateUserRedemptionRecords ..."**.

```
49  // IterateHostZones iterates zones
50  func (k Keeper) IterateUserRedemptionRecords(ctx sdk.Context,
```

In `x/stakeibc/keeper/msg_server_register_ica.go` :

The first return value is in type `*types.MsgRegisterAccountResponse` but it is recorded as "ICAAccount" in comment, the comment is incorrect.

```
23      // Return ICAAccount, err
24      return &types.MsgRegisterAccountResponse{}, nil
```

In `x/stakeibc/keeper/host_zone.go` :

Comment in line 178 is the doc comment for method `GetRedemptionAccount()`, therefore the comment should have the following format **"GetRedemptionAccount ..."**.

```
178  // GetHostZoneFromIBCDenom returns a HostZone from a IBCDenom
179  func (k Keeper) GetRedemptionAccount(ctx sdk.Context, hostZone types.HostZone)
(*types.ICAAccount, bool) {
```

In `x/interchainquery/keeper/keeper.go` :

Comment in line 86 is the doc comment for method `DeleteDatapoint()`, therefore the comment should have the following format **"DeleteDatapoint ..."**.

```
 86  // DeleteQuery delete datapoint
 87  func (k Keeper) DeleteDatapoint(ctx sdk.Context, id string) {
```

In `x/records/keeper/epoch_unbonding_record.go` :

Comment in line 84 is the doc comment for method `GetLatestEpochUnbondingRecord()`, therefore the comment should have the following format **"GetLatestEpochUnbondingRecord ..."**.

```
 84  // GetEpochUnbondingRecordByEpoch returns a epochUnbondingRecord from its epochNumber
 85  func (k Keeper) GetLatestEpochUnbondingRecord(ctx sdk.Context) (val types.EpochUnbondingRecord, found bool) {
```

Comment in line 130 is the doc comment for method `IterateEpochUnbondingRecords()`, therefore the comment should have the following format **"IterateEpochUnbondingRecords ..."**.

```
130  // IterateHostZones iterates zones
131  func (k Keeper) IterateEpochUnbondingRecords(ctx sdk.Context,
```

In `x/stakeibc/types/expected_keepers.go` :

In our understanding, both of the comments in line 17 and line 19 are used to describe the interface `BankKeeper`, therefore it will be better to move these comments immediately before declaration.

```
 15  // BankKeeper defines the expected interface needed to retrieve account balances.
 16  type BankKeeper interface {
 17     // BankKeeper interface: https://github.com/cosmos/cosmos-sdk/blob/main/x/bank/keeper/keeper.go
 18     SpendableCoins(ctx sdk.Context, addr sdk.AccAddress) sdk.Coins
 19     // Methods imported from bank should be defined here
 20     GetBalance(ctx sdk.Context, addr sdk.AccAddress, denom string) sdk.Coin
 21     SendCoinsFromModuleToAccount(ctx sdk.Context, senderModule string, recipientAddr sdk.AccAddress, amt sdk.Coins) error
 22     SendCoinsFromAccountToModule(ctx sdk.Context, senderAddr sdk.AccAddress, recipientModule string, amt sdk.Coins) error
```

```
    MintCoins(ctx sdk.Context, moduleName string, amt sdk.Coins) error
}
```

Recommendation

We recommend client to make sure the comments are consistent with codes.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 39116ce359fc9ebdf12f7821a18bf12410686acf.

SBU-06 | Lack Of Input/Output Validation

Category	Severity	Location	Status
Volatile Code	● Informational	interchainquery/keeper/keeper.go: 47, 55, 113; interchainquery/keeper/queries.go: 37; interchainquery/types/msgs.go: 18; stakeibc/genesis.go: 18	✓ Resolved

Description

There are some discussions on the lack of input or output validations:

1. `x/interchainquery/keeper/keeper.go`, should `GetDatapointForId()` only return 1 result rather than multiple results?
2. `x/interchainquery/keeper/queries.go`, `SetQuery()` does not check if there's been existing another query with same key = stakers & same id = 0, according to the reference: https://github.com/schnetzlerjoe/interchain-query-spec
3. `x/interchainquery/types/msgs.go`, `NewMsgSubmitQueryResponse()` does not check if a message has already been fullfilled nor not.
4. `x/interchainquery/keeper/keeper.go` `MakeRequest()`, should there be validations about the inputs values before GenerateQueryHash(connection_id, chain_id, query_type, request, module, height)?
5. `x/stakeibc/genesis.go`, `InitGenesis()` and `ExportGenesis()` are important methods, should they validate the input parameters?

Recommendation

We recommend adding validation for the above processes.

Alleviation

`[Stride]`:

1. It should only return one value, which we think it does.
2. The query ID is a hash of connection_id, chain_id, query_type, request, module and height, per `Queries.go:NewQuery`. Calling SetQuery should override existing queries that have the same hash. We check for this in `keeper.go:146`.
3. The result comes from the ICQ relayer which gets the result from the host chain query alongside the proof!
4. The issue has been fixed in this PR
5. The issue has been fixed in this PR

SBU-07 | Unhandled Return Value

Category	Severity	Location	Status
Logical Issue	● Informational	interchainquery/keeper/keeper.go: 71; interchainquery/keeper/queries.go: 53; records/keeper/deposit_record.go: 85; records/keeper/epoch_unbonding_record.go: 107, 136; records/keeper/user_redemption_record.go: 38, 55; records/module.go: 82; records/module_ibc.go: 57~66, 117, 128, 139, 150, 226; stakeibc/keeper/epoch_tracker.go: 54; stakeibc/keeper/host_zone.go: 85, 161; stakeibc/keeper/unbonding_records.go: 196; stakeibc/module.go: 83	✓ Resolved

Description

The returned `error` is unhandled in linked positions.

Recommendation

We recommend handling the `error` for improving maintainability.

Alleviation

`[Stride]` : in these functions, I'm not sure it makes to return an error because no line within the function can error.

- keeper.go#L71-71: 71
- queries.go#L53-53: 53
- deposit_record.go#L85-85: 85
- epoch_unbonding_record.go#L107-107: 107
- epoch_unbonding_record.go#L136-136: 136
- user_redemption_record.go#L38-38: 38
- user_redemption_record.go#L55-55: 55
- module.go#L82-82: 82
- module.go#L83-83: 83

`[CertiK]`: The team heeded the recommendation and partially resolved the finding in commit 37d19af5fc28bbc57177488b28f1e5fcb1e23710 and 40d00a8454771bb17ba1472774b3b8941f4f2487.

SBU-08 | Unused Parameters

Category	Severity	Location	Status
Coding Style	● Informational	interchainquery/types/msgs.go: 18; stakeibc/abci.go: 14; stakeibc/keeper/hooks.go: 320, 335; stakeibc/keeper/host_zone.go: 179; stakeibc/keeper/validator_selection.go: 57, 66	✓ Resolved

Description

The linked parameters are never used.

Recommendation

We recommend the client to remove them if there is no plan for further usage.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 6702ae89814f71c305d0f9a1c40e498eddaf7254.

SBU-09 | Comparison To A Boolean Constant

Category	Severity	Location	Status
Language Specific	● Informational	records/module_ibc.go: 204; stakeibc/keeper/ibc_handlers.go: 253	✓ Resolved

Description

Boolean constants can be used directly and do not need to be compared with true or false.

Recommendation

We advise the client to remove the comparison to the boolean constant.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 4713934233babcabbb237832c19dce2abed5683e.

SLS-01 | Lack Of Sanity Check - Prefix Of `HostZone.IBCDenom`

Category	Severity	Location	Status
Volatile Code	● Minor	stakeibc/keeper/msg_server_liquid_stake.go: 54; stakeibc/types/message_liquid_stake.go: 24~26	✓ Resolved

Description

In file `x/stakeibc/keeper/msg_server_liquid_stake.go`, we have found that the prefix of `HostZone.IBCDenom` is checked and the prefix must be "ibc/" :

```
24 func IsIBCToken(denom string) bool {
25     return strings.HasPrefix(denom, "ibc/")
26 }
```

```
  54    isIbcToken := types.IsIBCToken(ibcDenom) //
  55    if !isIbcToken {
  56        k.Logger(ctx).Info("invalid token denom")
  57        return nil, sdkerrors.Wrapf(types.ErrInvalidToken, "invalid token denom (%s)", ibcDenom)
  58    }
```

But the prefix of `HostZone.IBCDenom` is not checked when registering a new `HostZone`.

Recommendation

We recommend using the same context "ibc/" to check the prefix for `HostZone.IBCDenom` when registering a new HostZone.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 4a85ff28d2c7350e51dd2d0d3f58f45126311ea5.

SLS-02 | Double Check The Usage Of Methods

Category	Severity	Location	Status
Control Flow	● Informational	stakeibc/client/cli/tx.go: 34; stakeibc/client/cli/tx_register_host_zone.go: 15; stakeibc/client/cli/tx_register_ica.go: 15; stakeibc/client/cli/tx_submit_tx.go: 19; stakeibc/keeper/msg_server_register_host_zone.go: 14; stakeibc/keeper/msg_server_register_ica.go: 14; stakeibc/keeper/msg_server_submit_tx.go: 25	✓ Resolved

Description

`CmdRegisterHostZone`, `GetTxCmd()`, `CmdRegisterAccount()`, `CmdSubmitTx()`, `RegisterHostZone` and CmdRegisterHostZone() methods have a comment “Remove this pre-launch” . What is the plan for them during the launch?

Recommendation

We recommend reviewing these methods prior to launch.

Alleviation

`[Stride]`: We have since whitelisted these functions so they can only be invoked by the Stride Labs address. Public users should not be able to call them. And added this logic in this PR and refined it in this PR

SLS-03 | Redundant Alias For Imported Packages

Category	Severity	Location	Status
Coding Style	● Informational	stakeibc/keeper/hooks.go: 7; stakeibc/keeper/ibc_handlers.go: 6; stakeibc/types/genesis.go: 4; stakeibc/types/message_submit_tx.go: 4, 10; stakeibc/types/params.go: 4	✓ Resolved

Description

Package names and aliases are the same in linked positions.

Recommendation

We recommend removing redundant alias.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 603f79755e2c9da9cd0cf3a3829584341565bb5e.

SLU-01 | Incorrect Key Used

Category	Severity	Location	Status
Logical Issue	● Major	stakeibc/keeper/hooks.go: 76; stakeibc/keeper/msg_server_rebalance_validators.go: 78	✓ Resolved

Description

In the file `x/stakeibc/keeper/hooks.go`, the code on line 76 is used to get the time interval to update the redemption rate, where it would be more reasonable to use the key `types.KeyRedemptionRateInterval` than `types.KeyDepositInterval`.

```
75    // Update the redemption rate
76    redemptionRateInterval := int64(k.GetParam(ctx, types.KeyDepositInterval))
```

Moreover, in the file `x/stakeibc/keeper/msg_server_rebalance_validators.go`, the code on line 78 is used to get the rebalancing threshold while using the key `types.KeyValidatorRebalancingThreshold` would be more reasonable than `types.KeyDepositInterval`.

```
  78    rebalanceThreshold := float64(k.GetParam(ctx, types.KeyDepositInterval)) /
float64(10000)
```

Recommendation

We recommend correcting the key as described in the description.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 9be5af827148db820c57eb08ef8c5eee6eba52d1 and 4a85ff28d2c7350e51dd2d0d3f58f45126311ea5.

SLU-02 | Missing Return Statement

Category	Severity	Location	Status
Logical Issue	● Medium	stakeibc/keeper/msg_server_add_validator.go: 16~18; stakeibc/keeper/msg_server_submit_tx.go: 222~224, 242~244	✓ Resolved

Description

The error occurred at the linked position but no error was returned.

Recommendation

We recommend client to adding the return statement for catching errors. More further, we recommend client to make sure all of the errors would be handled.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit e1ea73b0b5cc91c7e3c1fa47e19dea25f26bb7ac and 95aff6452b70351c9e33c880511ade265d5f2e7a.

SLU-03 | Unreasonable Receiver Type

Category	Severity	Location	Status
Volatile Code	● Minor	stakeibc/keeper/msg_server_redeem_stake.go: 14; stakeibc/keeper/msg_server_register_host_zone.go: 15; stakeibc/keeper/msg_server_register_ica.go: 15; stakeibc/keeper/msg_server_submit_tx.go: 26	✓ Resolved

Description

Depending on the logic in `x/stakeibc/keeper/msg_server.go`, when implementing the methods defined in interface `types.MsgServer`, the type of methods' receiver should be `msgServer`, but the receiver of some methods have been specified as type `Keeper`.

```go
func NewMsgServerImpl(keeper Keeper) types.MsgServer {
    return msgServer{Keeper: keeper}
}

var _ types.MsgServer = msgServer{}
```

Recommendation

We recommend that when implementing the methods in interface `types.MsgServer`, the receiver of each method should be defined as type `msgServer`.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 18830b19176450715 75e4349472e59cfc9292f0a and b27541bad2072536c8a7cb43fdf89e2807e80638.

SLU-06 | Naming Optimization

Category	Severity	Location	Status
Coding Style	● Informational	stakeibc/keeper/msg_server_redeem_stake.go: 36~38; stakeibc/keeper/validator_selection.go: 41	✓ Resolved

Description

1. The map `targetWeight` is used to record the target delegation amount for each validator, the naming "targetWeight" would give the misconception that this variable is used to store weight for each validator.
2. The variable `stAmount` should be the amount of native token that the depositor should receive, perhaps it was improperly named.

Recommendation

We recommend renaming these variable for improving code readability.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in PR b5945681d992f7aff9ad60c86708ac29091e1b73.

SLU-07 | Hard Coded Sensitive Values

Category	Severity	Location	Status
Control Flow	● Informational	stakeibc/keeper/callbacks.go: 116; stakeibc/keeper/hooks.go: 264; stakeibc/keeper/msg_server_submit_tx.go: 157	✓ Resolved

Description

The linked codes are hard-coded in the code, these parameters make more sense if set in the configuration file

Recommendation

Recommended these parameters are set in configuration file.

Alleviation

`[CertiK]`: The team addressed the issue in `hooks.go` in commit f3f48a9f80c786302a580fd1e4510a99c34e7c1d, and the other two is intentionally designed.

1. The key in `msg_server_submit_tx.go` is not hardcoded, it's the name of a store on the host zone that will not change (ICQs use ABCIQuery which ingests store names)
2. The revenue account in `callbacks.go` is intentionally hardcoded and will remain that way, it's not meant to be a governance param!

SLU-08 | Redundant Type Conversion

Category	Severity	Location	Status
Coding Style	● Informational	stakeibc/keeper/callbacks.go: 62; stakeibc/keeper/ibc_handlers.go: 212; stakeibc/keeper/msg_server_submit_tx.go: 59, 209; stakeibc/keeper/unbonding_records.go: 102	✓ Resolved

Description

It is not necessary to convert the type in the linked positions.

Recommendation

We recommend removing redundant type conversion statements.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit f9515130ce512fa798265d43aba6d076d2434426.

SLU-09 | SafeMath Not Used

Category	Severity	Location	Status
Mathematical Operations	● Informational	stakeibc/keeper/hooks.go: 329, 344; stakeibc/keeper/msg_server_liquid_stake.go: 85; stakeibc/keeper/msg_server_redeem_stake.go: 88; stakeibc/keeper/unbonding_records.go: 51, 209	ⓘ Acknowledged

Description

SafeMath from `Cosmos-sdk` is not used in the linked functions which makes them possible for overflow/underflow and will lead to an inaccurate calculation result.

- x/stakeibc/keeper/msg_server_liquid_stake.go

```
85    depositRecord.Amount += int64(msg.Amount)
```

- x/stakeibc/keeper/unbonding_records.go

```
for _, unbondingRecord := range unbondingRecords {
    ......
    totalAmtTransferToRedemptionAcct += unbonding.Amount
    ......
}
```

Recommendation

We understand that overflow/underflow does not usually occur in normal processes, but we recommend considering extreme cases, especially since `msg.Amount` is an input parameter.

Alleviation

Stride team acknowledged this finding.

UNB-01 | Incorrect Comparison Condition

Category	Severity	Location	Status
Logical Issue	● Minor	stakeibc/keeper/unbonding_records.go: 219	✓ Resolved

Description

In the below if-statement, the first check condition should be the delegation account rather than the withdrawal account because the withdrawal account is not used in this function.

```
219    if (&zoneInfo).WithdrawalAccount != nil && (&zoneInfo).RedemptionAccount != nil {
// only process host zones once withdrawal accounts are registered
220        // get the delegation account and rewards account
221        delegationAccount := zoneInfo.GetDelegationAccount()
222        redemptionAccount := zoneInfo.GetRedemptionAccount()
```

Recommendation

It's recommended to correct `(&zoneInfo).WithdrawalAccount != nil` as `(&zoneInfo).delegationAccount != nil`.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 193a2e15b66b0bd25b14785ed9bf3e152eeeae1b.

VAI-01 | Lack Of Sanity Check - Zero Divisor

Category	Severity	Location	Status
Volatile Code	● Minor	stakeibc/keeper/validator_selection.go: 36, 48	✓ Resolved

Description

The variable `totalWeight` is used as a divisor but it could be ZERO.

Recommendation

We recommend adding a check to ensure that the value of `totalWeight` is not ZERO.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in PR 9be5af827148db820c57eb08ef8c5eee6eba52d1.

Optimizations

ID	Title	Category	Severity	Status
GLOBAL-01	Lack Of Gas Implementation	Logical Issue	● Optimization	✓ Resolved
GEO-01	Discussion On `UserRedemptionRecordCount`	Coding Style	● Optimization	✓ Resolved
IBC-03	Discussion The Potential Redundant Statements	Logical Issue	● Optimization	✓ Resolved
MSM-01	Incorrect Format 'Verb' Used	Logical Issue	● Optimization	✓ Resolved
SBU-01	Using Method `Error()` To Print Information Logs	Coding Style	● Optimization	✓ Resolved
SBU-02	Using Method `Debug()` To Print Debug Logs	Coding Style	● Optimization	✓ Resolved
SLU-04	Typo	Coding Style	● Optimization	✓ Resolved
SLU-05	Using Method `Info()` To Print Error Logs	Coding Style	● Optimization	✓ Resolved

GLOBAL-01 | Lack Of Gas Implementation

Category	Severity	Location	Status
Logical Issue	● Optimization		✓ Resolved

Description

Overview

In a blockchain system, we need to use Gas to prevent junk transactions and optimize TPS.

However, we did not find a specific implementation of Gas consumption in the Stride project for now. Given that Block Gaslimit has the role of regulating blockchain TPS, a complete Gas consumption mechanism is beneficial for tracking the node resources consumed in each message. We believe that in future upgrades, the project can make more detailed designs for the Gaslimit consumed during the processing of each message, in order to more accurately display the execution time required for each block in the form of gas. And then, we can optimize the Block Gaslimit to the optimal value, so that the blockchain TPS can be optimized.

Related Links

- Doc
    - Doc
    - Related Middleware Doc
    - Refund Doc
- Code
    - Gas Library Code
    - Kvstore library code
- Example
    - Example of Cosmos
    - Test Examples
    - Example of Shentu

Recommendation

We recommend using the following code to take Gas for the code that needs to be charged before the operation is executed in the future version.

```
ctx.GasMeter().ConsumeGas(fee, "NOTE")
```

We believe that a complete gas mechanism would help optimize TPS.

Alleviation

Stride team acknowledged this finding.

GEO-01 | Discussion On `UserRedemptionRecordCount`

Category	Severity	Location	Status
Coding Style	● Optimization	records/genesis.go: 18	✓ Resolved

Description

Is there any reason for keeping the commented-out logic about `UserRedemptionRecordCount` in the file `x/records/genesis.go`? If it will not be used in the launch.

Recommendation

We recommend removing it from the codebase.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 43aa67380d6cba7d859a89485f948d7666305d8f.

IBC-03 | Discussion The Potential Redundant Statements

Category	Severity	Location	Status
Logical Issue	● Optimization	stakeibc/keeper/ibc_handlers.go: 234~237, 348~351	✓ Resolved

Description

The variable `hostZoneUnbondings` is declared in the for loop of method `HandleUndelegate()`.However, this variable was never used in the following statement. Is there any logic missed? Or just a redundant statement here?

```
234 hostZoneUnbondings := epochUnbondingRecord.GetHostZoneUnbondings()
235 if len(hostZoneUnbondings) == 0 {
236   hostZoneUnbondings = make(map[string]*recordstypes.HostZoneUnbonding)
237 }
```

```
348 hostZoneUnbondings := epochUnbonding.GetHostZoneUnbondings()
349 if len(hostZoneUnbondings) == 0 {
350   hostZoneUnbondings = make(map[string]*recordstypes.HostZoneUnbonding)
351 }
```

There is a similar statement in the HandleSend() method.

Recommendation

We recommend revisiting above mentioned methods and logic.

Alleviation

`[CertiK]`: Stride team heeded the advice and resolved this finding in commit 4713934233babcabbb237832c19dce2abed5683e.

MSM-01 | Incorrect Format 'Verb' Used

Category	Severity	Location	Status
Logical Issue	● Optimization	stakeibc/keeper/msg_server_redeem_stake.go: 50	✓ Resolved

Description

Placeholder argument 'balance.Amount' is not type 'Int' therefore it is not correct to use verb `%d`.

```
50      k.Logger(ctx).Info(fmt.Sprintf("Redemption issuer IBCDenom balance: %d%s", balance.Amount, balance.Denom))
51      k.Logger(ctx).Info(fmt.Sprintf("Redemption requested redemotion amount: %v%s", inCoin.Amount, inCoin.Denom))
```

Recommendation

We recommend using verb `%v` in linked statement.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 5b29bed6fcc7961da88b44576a7f980a44ae43e8.

SBU-01 | Using Method `Error()` To Print Information Logs

Category	Severity	Location	Status
Coding Style	● Optimization	records/module_ibc.go: 195~196, 210~211; stakeibc/keeper/msg_server_claim_undelegated_tokens.go: 48~53	✓ Resolved

Description

In the linked positions, method `Logger.Error()` is used to print **INFORMATION** logs, which means that each information log will start with the identifier "E" instead of "I", and this will make the project more difficult to maintain.

Recommendation

We recommend client to use method `Logger.Info()` to print INFORMATION logs for improving code readability and maintainability.

Alleviation

`[CertiK]`: The team heeded the recommendation and resolved the finding in commit 6d5da1b96a45f80c2fef4c439183b8fcd23060ad and b9e1db5cafecac7fbbe1b14063017ce36b252fb3.

SBU-02 | Using Method `Debug()` To Print Debug Logs

Category	Severity	Location	Status
Coding Style	● Optimization	interchainquery/keeper/msg_server.go: 69, 76; stakeibc/keeper/ibc_handlers.go: 122	✓ Resolved

Description

In the linked positions, method `Logger.Debug()` is used to print **Debug** logs.

Recommendation

We recommend removing these statement if it’s used to track debug process, we recommend removing it before launch.

Or if these logs is necessary, we recommend using method `Logger.Info()` to print.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 453c2c1d01d6f3272f3e7ab4860d5645d7fcfd83.

SLU-04 | Typo

Category	Severity	Location	Status
Coding Style	● Optimization	stakeibc/keeper/hooks.go: 78, 109, 296; stakeibc/keeper/ibc_handlers.go : 294	✓ Resolved

Description

In `x/stakeibc/keeper/hooks.go` :

"Triggeting" in log should be "Triggering".

```
78              k.Logger(ctx).Info("Triggeting update redemption rate")
```

Variable naming "modeuleAcctBalance" should be "moduleAcctBalance".

```
296         modeuleAcctBalance, error := k.GetModuleAccountBalance(ctx, zoneInfo, depositRecords)
```

“host zome” in comment should be "host zone"

```
109                     // read clock time on host zome
```

In `x/stakeibc/keeper/ibc_handlers.go` :

"hostZome" in comment should be "hostZone".

```
294     // increment the stakedBal on the hostZome
```

Recommendation

We recommend correcting the typos for improving readability.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 16bcc1aafe55091a2118c4487fb05ee834838e4e.

SLU-05 | Using Method `Info()` To Print Error Logs

Category	Severity	Location	Status
Coding Style	● Optimization	stakeibc/keeper/hooks.go: 271~278; stakeibc/keeper/host_zone.go: 114; stakeibc/keeper/msg_server_add_validator.go: 16~30; stakeibc/keeper/msg_server_liquid_stake.go: 23~26, 28~32, 44~47, 55~58, 68~71, 75~78, 81~84, 94~97, 107~110, 116~119, 122~126; stakeibc/keeper/msg_server_redeem_stake.go: 94~97; stakeibc/keeper/msg_server_register_host_zone.go: 54~58, 61~65, 68~72, 75~79; stakeibc/keeper/msg_server_submit_tx.go: 222~225, 227~230, 242~244, 247~250	✓ Resolved

Description

In the linked positions, method `Logger.Info()` is used to print **ERROR** logs, which means that each error log will start with the identifier "I" instead of "E", and this will make the project more difficult to maintain.

Recommendation

We recommend client to use method `Logger.Error()` to print ERROR logs for improving code readability and maintainability.

Alleviation

`[CertiK]` : The team heeded the recommendation and resolved the finding in commit 82cd312dce0fb63c84ecf4bc6841eddbfe13898e and a09c92bda2e97a288c4c75c1fe767f006b26a158.

Appendix

Finding Categories

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND

“AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER’S OR ANY OTHER PERSON’S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER’S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK’S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER’S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED “AS IS” AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK’S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we’re able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.