Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project: Every Finance
Platform: Cross-Chain Network
Language: Solidity
Date: June 1st, 2023

Table of contents

Email: audit@EtherAuthority.io

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO THE PUBLIC AFTER ISSUES ARE RESOLVED.

Email: audit@EtherAuthority.io

Introduction

EtherAuthority was contracted by Every Finance to perform the Security audit of the Every Finance smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on June 1st, 2023.

The purpose of this audit was to address the following:

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

Project Background

- Every Finance Contract handles multiple contracts, and all contracts have different functions.
    - AssetBook: It allows the manager to add and remove assets to the investment portfolio.
    - HoldTime: It allows the update of the average hold time of the yield-bearing token.
    - Investment: Investors can deposit/withdraw funds, and the manager can validate investor requests.
    - Management: It allows the manager to set the different parameters of the product.
    - Proof: The Investor receives a deposit/withdrawal proof token when making a deposit/withdrawal request.
    - SafeHouse: It allows managing deposits, withdrawals, and transfers of assets.
    - Token: Implementation of the yield-bearing tokens {ERC20}.

- Every Finance Contract has functions like burn, mint, paused, receive, etc.
- There are 9 smart contracts, which were included in the audit scope. And there were some standard library codes, such as OpenZepelin, that were excluded. Because those standard library code is considered as time tested and community audited, so we can safely ignore them.

Audit scope

Name	Code Review and Security Analysis Report for Every Finance Smart Contracts
Platform	**Cross-Chain / Solidity**
File 1	AssetBook.sol
File 1 MD5 Hash	99557A38CC974D855E234339F7DE28F7
Updated File 1 MD5 Hash	0D9B40946FF97A9D429EC87BB7147C95
File 2	HoldTime.sol
File 2 MD5 Hash	935CEC2C923B41B1466E018238DA2283
File 3	Investment.sol
File 3 MD5 Hash	284F03D777EEA722B2C22747187BB2D6
Updated File 3 MD5 Hash	B49F08BECBE3FC2C40F28FEA29ACE6D8
File 4	Management.sol
File 4 MD5 Hash	324A30B4159A38EAE9F4BE822F8F08CC
Updated File 4 MD5 Hash	5FC3FA5B061D6CFF6588D6A41C897D7A
File 5	Proof.sol
File 5 MD5 Hash	4008B7C4E640D885327C2E2E7A357F75
Updated File 5 MD5 Hash	3FD0EA088CAE0E086B84631386F84144
File 6	SafeHouse.sol
File 6 MD5 Hash	2994C2399DC1BBA02019F61BAA060D0A
Updated File 6 MD5 Hash	A8D05782D310B6A84898342A5075B783
File 7	Token.sol
File 7 MD5 Hash	42663E2F880207925788FE893973C689
File 8	Treasury.sol
File 8 MD5 Hash	EBCC4679530F11E875CDBBA664E1332B
File 9	AssetBookAlpha.sol
File 9 MD5 Hash	999282F72E2ED9C5955AEF31D31101E9

File 10	DepositProofAlpha.sol
File 10 MD5 Hash	4AF38D74AA32DDEBA7F9251F0F1B7E98
File 11	HoldTimeAlpha.sol
File 11 MD5 Hash	A4E5C2D5EA090AA20E37B586B780EFDE
File 12	InvestmentAlpha.sol
File 12 MD5 Hash	2EA85EE156C11473FC70D46767ACE81F
File 13	ManagementAlpha.sol
File 13 MD5 Hash	A03AE1804DFAA92B3F9A8AFA211E43EF
File 14	SafeHouseAlpha.sol
File 14 MD5 Hash	FB40E347918429A813D33752B1C03C61
File 15	TokenAlpha.sol
File 15 MD5 Hash	5A393099245A9BCEA4F4CB958E693104
File 16	TreasuryAlpha.sol
File 16 MD5 Hash	C70F7752A9C5651A083E0F9B13314F91
File 17	WithdrawalProofAlpha.sol
File 17 MD5 Hash	18B17BA10980CB15E1A4B3CADD939CD0
File 18	AssetBookBeta.sol
File 18 MD5 Hash	12DC824BC59761CC5EAF7ADCD6495AC3
File 19	DepositProofBeta.sol
File 19 MD5 Hash	F91FB1EED9C31E80A34BB0E3081E13F1
File 20	HoldTimeBeta.sol
File 20 MD5 Hash	D1025794FF72B0CF1ABC4B181A6FD3CA
File 21	InvestmentBeta.sol
File 21 MD5 Hash	D2A1AE30AE8F52CC650242CA908CAEFD
File 22	ManagementBeta.sol
File 22 MD5 Hash	DA586D7706B40F0A35F63587DA2AE4BA
File 23	SafeHouseBeta.sol

File 23 MD5 Hash	426EC41E399684F6F2CA4B7642FBD6B9
File 24	TokenBeta.sol
File 24 MD5 Hash	F7C8256080657EE407CE7A2D44046473
File 25	TreasuryBeta.sol
File 25 MD5 Hash	F66222C671D8121AD7DBA67B801A016D
File 26	WithdrawalProofBeta.sol
File 26 MD5 Hash	592EFFDF5862BFE5FDD479E619957C29
File 27	AssetBookGamma.sol
File 27 MD5 Hash	468A834E7AB59E4EF8054171F5658663
File 28	DepositProofGamma.sol
File 28 MD5 Hash	4EE1323518D178640872D077F1D316B0
File 29	HoldTimeGamma.sol
File 29 MD5 Hash	36B388FF45264C5AD098804BC41E43E9
File 30	InvestmentGamma.sol
File 30 MD5 Hash	37FC3E9FEE69AF2E286CDCCCCC4AEB05
File 31	ManagementGamma.sol
File 31 MD5 Hash	51584D38EB8B649A3D3DB8CE1370CB2E
File 32	SafeHouseGamma.sol
File 32 MD5 Hash	BC7798A851B0F8EABCBC79887363987E
File 33	TokenGamma.sol
File 33 MD5 Hash	57D2301ADA2AA749CA5E4B98BAF5690D
File 34	TreasuryGamma.sol
File 34 MD5 Hash	49160734FEAE074E8BD3CE56194C279B
File 35	WithdrawalProofGamma.sol
File 35 MD5 Hash	73A897A70735F84DB7A4790052C49BA0
Audit Date	June 1st, 2023
Revised Audit Date	June 5th, 2023

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
File 1 AssetBook.sol **Admin has control over following functions:** ● Set asset address. ● Set Oracle address and price. **Other Specifications:** ● The manager can add and remove assets from the investment portfolio.	**YES, This is valid.**
File 2 HoldTime.sol **Admin has control over following functions:** ● Set the token address. ● Update the hold time. **Other Specifications:** ● The average hold time of a yield-bearing token can be updated.	**YES, This is valid.**
File 3 Investment.sol **Admin has control over following functions:** ● Set the manager addresses. ● Set the deposit proof addresses. ● Set the withdrawal proof address. ● Set the managementParity address. **Other Specifications:** ● It allows the investor to depositProof/withdraw funds and the manager to validate the depositProof/withdrawalProof investor requests.	**YES, This is valid.**

File 4 Management.sol **<u>Admin has control over following functions:</u>** ● Set the treasury address. ● Set the safeHouse address. **<u>Other Specifications:</u>** ● It allows the manager to set the different parameters of the product.	**YES, This is valid.**
File 5 Proof.sol **<u>Admin has control over following functions:</u>** ● Set the Investment addresses. ● Set the Metadata addresses ● Set the BaseURI. ● Set the tolerance value. ● Set the IsOnChainMetadata status. **<u>Other Specifications:</u>** ● The investor receives a deposit/withdrawal proof token when making a deposit/withdrawal request, which is validated by the manager.	**YES, This is valid.**
File 6 SafeHouse.sol **<u>Admin has control over following functions:</u>** ● Set the maximum withdrawal capacity in USD. ● Set the withdrawal capacity in USD. ● Set the price tolerance rate value. ● Set the assetBook address. ● Added a new vault address. **<u>Other Specifications:</u>** ● It allows the management of deposits, withdrawals and transfers of assets.	**YES, This is valid.**

File 7 Token.sol **<u>Admin has control over following functions:</u>** • Set the Investment addresses. • Set the hold time value. • Added and removed `account_` to `whitelist `. **<u>Other Specifications:</u>** • Implementation of the yield-bearing tokens.	**YES, This is valid.**
File 8 Treasury.sol • Treasury contracts have functions like: sendTo, receive.	**YES, This is valid.**
File 9 AssetBookAlpha.sol • The AssetBookAlpha contract inherited the AssetBook contract.	**YES, This is valid.**
File 10 DepositProofAlpha.sol • Name: DALPHA • Symbol: DALPHA • ID: 1	**YES, This is valid.**
File 11 HoldTimeAlpha.sol • The HoldTimeAlpha contract inherited the HoldTime contract.	**YES, This is valid.**
File 12 InvestmentAlpha.sol • The InvestmentAlpha contract inherited the Investment contract.	**YES, This is valid.**
File 13 ManagementAlpha.sol • The ManagementAlpha contract inherited the Management contract.	**YES, This is valid.**
File 14 SafeHouseAlpha.sol	**YES, This is valid.**

• The SafeHouseAlpha contract inherited the SafeHouse contract.	
File 15 TokenAlpha.sol • Name: ALPHA • Symbol: ALPHA	**YES, This is valid.**
File 16 TreasuryAlpha.sol • The TreasuryAlpha contract inherited the Treasury contract.	**YES, This is valid.**
File 17 WithdrawalProofAlpha.sol • Name: WALPHA • Symbol: WALPHA • ID: 0	**YES, This is valid.**
File 18 AssetBookBeta.sol • The AssetBookBeta contract inherited the AssetBook contract.	**YES, This is valid.**
File 19 DepositProofBeta.sol • Name: DBETA • Symbol: DBETA • ID: 1	**YES, This is valid.**
File 20 HoldTimeBeta.sol • The HoldTimeBeta contract inherited the HoldTime contract.	**YES, This is valid.**
File 21 InvestmentBeta.sol • The InvestmentBeta contract inherited the Investment contract.	**YES, This is valid.**
File 22 ManagementBeta.sol • The ManagementBeta contract inherited the	**YES, This is valid.**

Management contract.	
File 23 SafeHouseBeta.sol • The SafeHouseBeta contract inherited the SafeHouse contract.	**YES, This is valid.**
File 24 TokenBeta.sol • Name: BETA • Symbol: BETA	**YES, This is valid.**
File 25 TreasuryBeta.sol • The TreasuryBeta contract inherited the Treasury contract.	**YES, This is valid.**
File 26 WithdrawalProofBeta.sol • Name: WBETA • Symbol: WBETA • ID: 0	**YES, This is valid.**
File 27 AssetBookGamma.sol • The AssetBookGamma contract inherited the AssetBook contract.	**YES, This is valid.**
File 28 DepositProofGamma.sol • Name: DGAMMA • Symbol: DGAMMA • ID: 1	**YES, This is valid.**
File 29 HoldTimeGamma.sol • The HoldTimeGamma contract inherited the HoldTime contract.	**YES, This is valid.**
File 30 InvestmentGamma.sol • The InvestmentGamma contract inherited the Investment contract.	**YES, This is valid.**
File 31 ManagementGamma.sol	**YES, This is valid.**

• The ManagementGamma contract inherited the Management contract.	
File 32 SafeHouseGamma.sol • The SafeHouseGamma contract inherited the SafeHouse contract.	**YES, This is valid.**
File 33 TokenGamma.sol • Name: GAMMA • Symbol: GAMMA	**YES, This is valid.**
File 34 TreasuryGamma.sol • The TreasuryGamma contract inherited the Treasury contract.	**YES, This is valid.**
File 35 WithdrawalProofGamma.sol • Name: WGAMMA • Symbol: WGAMMA	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. Also, these contracts do contain owner control, which does not make them fully decentralized.

We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.
All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 0 medium, 2 low and 0 very low level issues.

We confirm that 2 low severity issues are acknowledged in the revised smart contract code.

Investors Advice: Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	N/A
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
	Other programming issues	Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Unused code	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: PASSED

Code Quality

This audit scope has 35 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in the Every Finance Protocol are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the Every Finance Protocol.

The Every Finance team has provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Code parts are well commented on smart contracts.

Documentation

We were given a Every Finance Protocol smart contract code in the form of a file. The hash of that code is mentioned above in the table.

As mentioned above, code parts are well commented. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Use of Dependencies

As per our observation, the libraries are used in this smart contracts infrastructure that are based on well known industry standard open source projects.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

AssetBook.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue
6	_revokeRole	internal	Passed	No Issue
7	updateAsset	external	Function input parameters lack of check	Refer to audit findings
8	removeAsset	external	access only Role	No Issue
9	getAsset	read	Passed	No Issue

AssetBookAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateAsset	external	access only Role	No Issue
3	removeAsset	external	access only Role	No Issue
4	getAsset	read	Passed	No Issue

AssetBookBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateAsset	external	access only Role	No Issue
3	removeAsset	external	access only Role	No Issue
4	getAsset	read	Passed	No Issue

AssetBookGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateAsset	external	access only Role	No Issue
3	removeAsset	external	access only Role	No Issue
4	getAsset	read	Passed	No Issue

HoldTime.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyOwner	modifier	Passed	No Issue
3	owner	read	Passed	No Issue
4	_checkOwner	internal	Passed	No Issue
5	renounceOwnership	write	access only Owner	No Issue
6	transferOwnership	write	access only Owner	No Issue
7	_transferOwnership	internal	Passed	No Issue
8	updateToken	external	access only Owner	No Issue
9	updateHoldTime	external	Passed	No Issue
10	getHoldTime	read	Passed	No Issue

Investment.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue
6	_revokeRole	internal	Passed	No Issue
7	whenNotPaused	modifier	Passed	No Issue
8	whenPaused	modifier	Passed	No Issue
9	paused	read	Passed	No Issue
10	_requireNotPaused	internal	Passed	No Issue
11	_requirePaused	internal	Passed	No Issue
12	_pause	internal	Passed	No Issue
13	_unpause	internal	Passed	No Issue
14	receive	external	Passed	No Issue
15	updateManagement	external	access only Role	No Issue
16	updateDepositProof	external	access only Role	No Issue
17	updateWithdrawalProof	external	access only Role	No Issue
18	updateManagementParity	external	access only Role	No Issue
19	updateToken	external	access only Role	No Issue
20	updateAsset	external	access only Role	No Issue
21	updateEventBatchSize	external	access only Role	No Issue
22	startNextEvent	external	access only Role	No Issue
23	validatedeposits	external	access only Role	No Issue
24	validatewithdrawalProofs	external	access only Role	No Issue
25	depositRequest	external	Passed	No Issue
26	canceldepositRequest	external	Passed	No Issue
27	withdrawaRequest	external	Passed	No Issue
28	cancelwithdrawalRequest	external	Passed	No Issue

29	sendToSafeHouse	external	access only Role	No Issue
30	mintPerformanceFee	external	access only Role	No Issue
31	mintManagementFee	external	access only Role	No Issue
32	mintOrBurnInvestmentFee	external	access only Role	No Issue
33	getdepositFee	read	Passed	No Issue
34	getTokenPrice	read	Passed	No Issue
35	isValidPrice	read	Passed	No Issue

Management.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue
6	_revokeRole	internal	Passed	No Issue
7	updateTreasury	external	access only Role	No Issue
8	updateSafeHouse	external	access only Role	No Issue
9	updateIsCancelDeposit	external	access only Role	No Issue
10	updateIsCancelWithdrawal	external	access only Role	No Issue
11	updateDepositFee	external	access only Role	No Issue
12	updateManagementFeeRate	external	access only Role	No Issue
13	updatePerformanceFeeRate	external	access only Role	No Issue
14	updateMinDepositAmount	external	access only Role	No Issue
15	updateTokenPrice	external	access only Role	No Issue
16	addWithdrawalFee	external	access only Role	No Issue
17	updateWithdrawalFee	external	access only Role	No Issue
18	deleteLastWithdrawalFee	external	access only Role	No Issue
19	calculateWithdrawalFeeRate	read	Passed	No Issue
20	getDepositFee	read	Passed	No Issue
21	getTokenPrice	read	Passed	No Issue
22	getWithdrawalFeeRate	read	Passed	No Issue
23	getWithdrawalFee	read	Passed	No Issue
24	getWithdrawalFeeSize	read	Passed	No Issue

Proof.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue
6	_revokeRole	internal	Passed	No Issue

7	onlyOwner	modifier	Passed	No Issue
8	owner	read	Passed	No Issue
9	_checkOwner	internal	Passed	No Issue
10	renounceOwnership	write	access only Owner	No Issue
11	transferOwnership	write	access only Owner	No Issue
12	_transferOwnership	internal	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	tokenOfOwnerByIndex	read	Passed	No Issue
15	totalSupply	read	Passed	No Issue
16	tokenByIndex	read	Passed	No Issue
17	_beforeTokenTransfer	internal	Passed	No Issue
18	_beforeConsecutiveTokenTransfer	internal	Passed	No Issue
19	_addTokenToOwnerEnumeration	write	Passed	No Issue
20	_addTokenToAllTokensEnumeration	write	Passed	No Issue
21	_removeTokenFromOwnerEnumeration	write	Passed	No Issue
22	_removeTokenFromAllTokensEnumeration	write	Passed	No Issue
23	updateInvestment	external	access only Role	No Issue
24	updateMetadata	external	access only Role	No Issue
25	setBaseURI	external	access only Role	No Issue
26	updateTolerance	external	access only Role	No Issue
27	updateIsOnChainMetadata	external	access only Role	No Issue
28	mint	external	Passed	No Issue
29	increasePendingRequest	external	access only Role	No Issue
30	decreasePendingRequest	external	access only Role	No Issue
31	preValidatePendingRequest	external	access only Role	No Issue
32	validatePendingRequest	external	access only Role	No Issue
33	tokenURI	read	Passed	No Issue
34	supportsInterface	read	Passed	No Issue
35	_increasePendingRequest	internal	Passed	No Issue
36	_decreasePendingRequest	internal	Passed	No Issue
37	burn	internal	Passed	No Issue
38	_decreaseTotalAmount	internal	Passed	No Issue

SafeHouse.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue

6	_revokeRole	internal	Passed	No Issue
7	whenNotPaused	modifier	Passed	No Issue
8	whenPaused	modifier	Passed	No Issue
9	paused	read	Passed	No Issue
10	_requireNotPaused	internal	Passed	No Issue
11	_requirePaused	internal	Passed	No Issue
12	_pause	internal	Passed	No Issue
13	_unpause	internal	Passed	No Issue
14	receive	external	Passed	No Issue
15	updateMaxWithdrawalCapacity	external	MaxWithdrawal limit is not set	Refer to audit findings
16	updateWithdrawalCapacity	external	access only Role	No Issue
17	updatePriceToleranceRate	external	access only Role	No Issue
18	updateAssetBook	external	access only Role	No Issue
19	addVault	external	access only Role	No Issue
20	removeVault	external	access only Role	No Issue
21	depositAsset	external	Function input parameters lack of check	Refer to audit findings
22	withdrawAsset	external	Function input parameters lack of check	Refer to audit findings
23	sendToVault	external	Function input parameters lack of check	Refer to audit findings
24	getLatestPrice	read	Passed	No Issue

Token.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	allowance	read	Passed	No Issue
3	approve	write	Passed	No Issue
4	transferFrom	write	Passed	No Issue
5	increaseAllowance	write	Passed	No Issue
6	decreaseAllowance	write	Passed	No Issue
7	_transfer	internal	Passed	No Issue
8	_mint	internal	Passed	No Issue
9	_burn	internal	Passed	No Issue
10	_approve	internal	Passed	No Issue
11	_spendAllowance	internal	Passed	No Issue
12	_beforeTokenTransfer	internal	Passed	No Issue
13	_afterTokenTransfer	internal	Passed	No Issue
14	totalSupply	read	Passed	No Issue
15	balanceOf	read	Passed	No Issue
16	transfer	write	Passed	No Issue

17	supportsInterface	read	Passed	No Issue
18	getRoleMember	read	Passed	No Issue
19	getRoleMemberCount	read	Passed	No Issue
20	_grantRole	internal	Passed	No Issue
21	_revokeRole	internal	Passed	No Issue
22	updateInvestment	external	access only Role	No Issue
23	updateHoldTime	external	access only Role	No Issue
24	addToWhiteList	external	access only Role	No Issue
25	removeFromWhiteList	external	access only Role	No Issue
26	mint	external	access only Role	No Issue
27	burn	external	access only Role	No Issue
28	getHoldTime	read	Passed	No Issue
29	_updateHoldTime	internal	Passed	No Issue
30	_beforeTokenTransfer	internal	Passed	No Issue

Treasury.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	getRoleMember	read	Passed	No Issue
4	getRoleMemberCount	read	Passed	No Issue
5	_grantRole	internal	Passed	No Issue
6	_revokeRole	internal	Passed	No Issue
7	receive	external	Passed	No Issue
8	sendTo	write	Function input parameters lack of check	Refer to audit findings

DepositProofAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue

13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue
15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

DepositProofBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue
15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

DepositProofGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue

15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

HoldTimeAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateToken	external	access only Owner	No Issue
3	updateHoldTime	external	Passed	No Issue
4	getHoldTime	read	Passed	No Issue

HoldTimeBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateToken	external	access only Owner	No Issue
3	updateHoldTime	external	Passed	No Issue
4	getHoldTime	read	Passed	No Issue

HoldTimeGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateToken	external	access only Owner	No Issue
3	updateHoldTime	external	Passed	No Issue
4	getHoldTime	read	Passed	No Issue

InvestmentAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateManagement	external	access only Role	No Issue
4	updateDepositProof	external	access only Role	No Issue
5	updateWithdrawalProof	external	access only Role	No Issue
6	updateManagementParity	external	access only Role	No Issue
7	updateToken	external	access only Role	No Issue
8	updateAsset	external	access only Role	No Issue

9	updateEventBatchSize	external	access only Role	No Issue
10	startNextEvent	external	access only Role	No Issue
11	validatedeposits	external	access only Role	No Issue
12	validatewithdrawalProofs	external	access only Role	No Issue
13	depositRequest	external	Passed	No Issue
14	canceldepositRequest	external	Passed	No Issue
15	withdrawaRequest	external	Passed	No Issue
16	cancelwithdrawalRequest	external	Passed	No Issue
17	sendToSafeHouse	external	access only Role	No Issue
18	mintPerformanceFee	external	access only Role	No Issue
19	mintManagementFee	external	access only Role	No Issue
20	mintOrBurnInvestmentFee	external	access only Role	No Issue
21	getdepositFee	read	Passed	No Issue
22	getTokenPrice	read	Passed	No Issue
23	isValidPrice	read	Passed	No Issue

InvestmentBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateManagement	external	access only Role	No Issue
4	updateDepositProof	external	access only Role	No Issue
5	updateWithdrawalProof	external	access only Role	No Issue
6	updateManagementParity	external	access only Role	No Issue
7	updateToken	external	access only Role	No Issue
8	updateAsset	external	access only Role	No Issue
9	updateEventBatchSize	external	access only Role	No Issue
10	startNextEvent	external	access only Role	No Issue
11	validatedeposits	external	access only Role	No Issue
12	validatewithdrawalProofs	external	access only Role	No Issue
13	depositRequest	external	Passed	No Issue
14	canceldepositRequest	external	Passed	No Issue
15	withdrawaRequest	external	Passed	No Issue
16	cancelwithdrawalRequest	external	Passed	No Issue
17	sendToSafeHouse	external	access only Role	No Issue
18	mintPerformanceFee	external	access only Role	No Issue
19	mintManagementFee	external	access only Role	No Issue
20	mintOrBurnInvestmentFee	external	access only Role	No Issue
21	getdepositFee	read	Passed	No Issue
22	getTokenPrice	read	Passed	No Issue
23	isValidPrice	read	Passed	No Issue

InvestmentGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateManagement	external	access only Role	No Issue
4	updateDepositProof	external	access only Role	No Issue
5	updateWithdrawalProof	external	access only Role	No Issue
6	updateManagementParity	external	access only Role	No Issue
7	updateToken	external	access only Role	No Issue
8	updateAsset	external	access only Role	No Issue
9	updateEventBatchSize	external	access only Role	No Issue
10	startNextEvent	external	access only Role	No Issue
11	validatedeposits	external	access only Role	No Issue
12	validatewithdrawalProofs	external	access only Role	No Issue
13	depositRequest	external	Passed	No Issue
14	canceldepositRequest	external	Passed	No Issue
15	withdrawaRequest	external	Passed	No Issue
16	cancelwithdrawalRequest	external	Passed	No Issue
17	sendToSafeHouse	external	access only Role	No Issue
18	mintPerformanceFee	external	access only Role	No Issue
19	mintManagementFee	external	access only Role	No Issue
20	mintOrBurnInvestmentFee	external	access only Role	No Issue
21	getdepositFee	read	Passed	No Issue
22	getTokenPrice	read	Passed	No Issue
23	isValidPrice	read	Passed	No Issue

ManagementAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateTreasury	external	access only Role	No Issue
3	updateSafeHouse	external	access only Role	No Issue
4	updateIsCancelDeposit	external	access only Role	No Issue
5	updateIsCancelWithdrawal	external	access only Role	No Issue
6	updateDepositFee	external	access only Role	No Issue
7	updateManagementFeeRate	external	access only Role	No Issue
8	updatePerformanceFeeRate	external	access only Role	No Issue
9	updateMinDepositAmount	external	access only Role	No Issue
10	updateTokenPrice	external	access only Role	No Issue
11	addWithdrawalFee	external	access only Role	No Issue
12	updateWithdrawalFee	external	access only Role	No Issue
13	deleteLastWithdrawalFee	external	access only Role	No Issue
14	calculateWithdrawalFeeRate	read	Passed	No Issue
15	getDepositFee	read	Passed	No Issue
16	getTokenPrice	read	Passed	No Issue

17	getWithdrawalFeeRate	read	Passed	No Issue
18	getWithdrawalFee	read	Passed	No Issue
19	getWithdrawalFeeSize	read	Passed	No Issue

ManagementBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateTreasury	external	access only Role	No Issue
3	updateSafeHouse	external	access only Role	No Issue
4	updateIsCancelDeposit	external	access only Role	No Issue
5	updateIsCancelWithdrawal	external	access only Role	No Issue
6	updateDepositFee	external	access only Role	No Issue
7	updateManagementFeeRate	external	access only Role	No Issue
8	updatePerformanceFeeRate	external	access only Role	No Issue
9	updateMinDepositAmount	external	access only Role	No Issue
10	updateTokenPrice	external	access only Role	No Issue
11	addWithdrawalFee	external	access only Role	No Issue
12	updateWithdrawalFee	external	access only Role	No Issue
13	deleteLastWithdrawalFee	external	access only Role	No Issue
14	calculateWithdrawalFeeRate	read	Passed	No Issue
15	getDepositFee	read	Passed	No Issue
16	getTokenPrice	read	Passed	No Issue
17	getWithdrawalFeeRate	read	Passed	No Issue
18	getWithdrawalFee	read	Passed	No Issue
19	getWithdrawalFeeSize	read	Passed	No Issue

ManagementGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateTreasury	external	access only Role	No Issue
3	updateSafeHouse	external	access only Role	No Issue
4	updateIsCancelDeposit	external	access only Role	No Issue
5	updateIsCancelWithdrawal	external	access only Role	No Issue
6	updateDepositFee	external	access only Role	No Issue
7	updateManagementFeeRate	external	access only Role	No Issue
8	updatePerformanceFeeRate	external	access only Role	No Issue
9	updateMinDepositAmount	external	access only Role	No Issue
10	updateTokenPrice	external	access only Role	No Issue
11	addWithdrawalFee	external	access only Role	No Issue
12	updateWithdrawalFee	external	access only Role	No Issue
13	deleteLastWithdrawalFee	external	access only Role	No Issue
14	calculateWithdrawalFeeRate	read	Passed	No Issue

15	getDepositFee	read	Passed	No Issue
16	getTokenPrice	read	Passed	No Issue
17	getWithdrawalFeeRate	read	Passed	No Issue
18	getWithdrawalFee	read	Passed	No Issue
19	getWithdrawalFeeSize	read	Passed	No Issue

SafeHouseAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateMaxWithdrawalCapacity	external	access only Role	No Issue
4	updateWithdrawalCapacity	external	access only Role	No Issue
5	updatePriceToleranceRate	external	access only Role	No Issue
6	updateAssetBook	external	access only Role	No Issue
7	addVault	external	access only Role	No Issue
8	removeVault	external	access only Role	No Issue
9	depositAsset	external	Passed	No Issue
10	withdrawAsset	external	access only Role	No Issue
11	sendToVault	external	access only Role	No Issue
12	getLatestPrice	read	Passed	No Issue

SafeHouseBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateMaxWithdrawalCapacity	external	access only Role	No Issue
4	updateWithdrawalCapacity	external	access only Role	No Issue
5	updatePriceToleranceRate	external	access only Role	No Issue
6	updateAssetBook	external	access only Role	No Issue
7	addVault	external	access only Role	No Issue
8	removeVault	external	access only Role	No Issue
9	depositAsset	external	Passed	No Issue
10	withdrawAsset	external	access only Role	No Issue
11	sendToVault	external	access only Role	No Issue
12	getLatestPrice	read	Passed	No Issue

SafeHouseGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	updateMaxWithdrawalCapacity	external	access only Role	No Issue
4	updateWithdrawalCapacity	external	access only Role	No Issue
5	updatePriceToleranceRate	external	access only Role	No Issue
6	updateAssetBook	external	access only Role	No Issue
7	addVault	external	access only Role	No Issue
8	removeVault	external	access only Role	No Issue
9	depositAsset	external	Passed	No Issue
10	withdrawAsset	external	access only Role	No Issue
11	sendToVault	external	access only Role	No Issue
12	getLatestPrice	read	Passed	No Issue

TokenAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateHoldTime	external	access only Role	No Issue
4	addToWhiteList	external	access only Role	No Issue
5	removeFromWhiteList	external	access only Role	No Issue
6	mint	external	access only Role	No Issue
7	burn	external	access only Role	No Issue
8	getHoldTime	read	Passed	No Issue
9	_updateHoldTime	internal	Passed	No Issue
10	_beforeTokenTransfer	internal	Passed	No Issue

TokenBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateHoldTime	external	access only Role	No Issue
4	addToWhiteList	external	access only Role	No Issue
5	removeFromWhiteList	external	access only Role	No Issue
6	mint	external	access only Role	No Issue
7	burn	external	access only Role	No Issue
8	getHoldTime	read	Passed	No Issue
9	_updateHoldTime	internal	Passed	No Issue
10	_beforeTokenTransfer	internal	Passed	No Issue

TokenGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateHoldTime	external	access only Role	No Issue
4	addToWhiteList	external	access only Role	No Issue
5	removeFromWhiteList	external	access only Role	No Issue
6	mint	external	access only Role	No Issue
7	burn	external	access only Role	No Issue
8	getHoldTime	read	Passed	No Issue
9	_updateHoldTime	internal	Passed	No Issue
10	_beforeTokenTransfer	internal	Passed	No Issue

TreasuryAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	sendTo	write	access only Role	No Issue

TreasuryBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	sendTo	write	access only Role	No Issue

TreasuryGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	receive	external	Passed	No Issue
3	sendTo	write	access only Role	No Issue

WithdrawalProofAlpha.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue
15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

WithdrawalProofBeta.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue
15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

WithdrawalProofGamma.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	updateInvestment	external	access only Role	No Issue
3	updateMetadata	external	access only Role	No Issue
4	setBaseURI	external	access only Role	No Issue
5	updateTolerance	external	access only Role	No Issue
6	updateIsOnChainMetadata	external	access only Role	No Issue
7	mint	external	access only Role	No Issue
8	increasePendingRequest	external	access only Role	No Issue
9	decreasePendingRequest	external	access only Role	No Issue
10	preValidatePendingRequest	external	access only Role	No Issue
11	validatePendingRequest	external	access only Role	No Issue
12	tokenURI	read	Passed	No Issue
13	supportsInterface	read	Passed	No Issue
14	_increasePendingRequest	internal	Passed	No Issue
15	_decreasePendingRequest	internal	Passed	No Issue
16	burn	internal	Passed	No Issue
17	_decreaseTotalAmount	internal	Passed	No Issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Email: audit@EtherAuthority.io

Audit Findings

Critical Severity

No Critical severity vulnerabilities were found.

High Severity

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

(1) MaxWithdrawal limit is not set: **SafeHouse.sol**

In the updateMaxWithdrawalCapacity, updateWithdrawalCapacity functions: admin can set the variables to any values. These values might one day be set to 100% to force transfers to go to the manager.

Resolution: We suggest checking the range limit for these function parameters.

Status: This is acknowledged in the revised smart contract code. Admin set this value to control the manager withdrawals

(2) Function input parameters lack of check:

Variable validation is not performed in below functions:

SafeHouse.sol

- sendToVault = asset_
- withdrawAsset = asset_
- depositAsset = asset_

Treasury.sol

- sendTo = asset_

AssetBook.sol

- updateAsset = asset_

Resolution: We advise to put validation: integer type variables should be greater than 0 and address type variables should not be address(0).
Status: This is acknowledged in the revised smart contract code. Assets can be ether, so it can be the zero address.

Very Low / Informational / Best practices:

No Informational severity vulnerabilities were found.

Centralization

This smart contract has some functions that can only be executed by the Admin (Owner). For the admin roles and the owners of the smart contract, multisignature is used to reduce the centralization risk. If the admin wallet private key were compromised, then it would create trouble.

Following are admin functions:

Ownable.sol

- renounceOwnership: Deleting ownership will leave the contract without an owner, removing any owner-only functionality.
- transferOwnership: Current owner can transfer ownership of the contract to a new account.
- _checkOwner: Thrown when the sender is not the owner.

AssetBook.sol

- updateAsset: The asset address, oracle address and price can be updated by the admin.

HoldTime.sol

- updateToken: The Token address can be updated by the owner.
- updateHoldTime:The Hold time can be updated by the owner.

Investment.sol

- updateManagement: The management address can be updated by the admin.
- updateDepositProof: The deposit proof address can be updated by the admin.
- updateWithdrawalProof: The withdrawal proof address can be updated by the admin.
- ManagementParity: The management parity address can be updated by the admin.
- updateToken: The admin should update the yield-bearing token address.
- updateAsset: The administrator can update the underlying asset that investors can deposit.
- updateEventBatchSize: Event Batch Size can be updated by the admin.

Management.sol

- updateTreasury: The treasury address can be updated by the admin.
- updateSafeHouse: The SafeHouse address can be updated by the admin.

Proof.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

SafeHouse.sol

- updateMaxWithdrawalCapacity: The administrator can set the maximum withdrawal capacity in USD.
- updateWithdrawalCapacity: The administrator can set the withdrawal capacity in USD.
- updatePriceToleranceRate: The administrator can set price tolerance rate value.
- updateAssetBook: AssetBook address can be updated by the admin.
- addVault: A new vault address can be added by the admin.

Token.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateHoldTime: The hold time value can be updated by the admin.
- addToWhiteList: Added `account_` to `whitelist ` by the admin.
- removeFromWhiteList: Remove `account_` from `whitelist ` by the admin.

AssetBookAlpha.sol

- updateAsset: The asset address, oracle address and price can be updated by the admin.

DepositProofAlpha.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

HoldTimeAlpha.sol

- updateToken: The Token address can be updated by the owner.
- updateHoldTime:The Hold time can be updated by the owner.

InvestmentAlpha.sol

- updateManagement: The management address can be updated by the admin.
- updateDepositProof: The Deposit Proof address can be updated by the admin.
- updateWithdrawalProof: WithdrawalProof address can be updated by the admin.
- ManagementParity: ManagementParity address can be updated by the admin.
- updateToken: The admin should update the yield-bearing token address.
- updateAsset: The administrator can update the underlying asset that investors can deposit.
- updateEventBatchSize:EventBatchSize can be updated by the admin.

ManagementAlpha.sol

- updateTreasury: The treasury address can be updated by the admin.

- updateSafeHouse: The SafeHouse address can be updated by the admin.

SafeHouseAlpha.sol

- updateMaxWithdrawalCapacity: The administrator can set the maximum withdrawal capacity in USD.
- updateWithdrawalCapacity: The administrator can set the withdrawal capacity in USD.
- updatePriceToleranceRate: The administrator can set price tolerance rate value.
- updateAssetBook: AssetBook address can be updated by the admin.
- addVault: A new vault address can be added by the admin.

TokenAlpha.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateHoldTime: The hold time value can be updated by the admin.
- addToWhiteList: Added `account_` to `whitelist ` by the admin.
- removeFromWhiteList: Remove `account_` from `whitelist ` by the admin.

WithdrawalProofAlpha.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

AssetBookBeta.sol

- updateAsset: The asset address, oracle address and price can be updated by the admin.

DepositProofBeta.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.

- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

HoldTimeBeta.sol

- updateToken: The Token address can be updated by the owner.
- updateHoldTime:The Hold time can be updated by the owner.

InvestmentBeta.sol

- updateManagement: The management address can be updated by the admin.
- updateDepositProof: The Deposit Proof address can be updated by the admin.
- updateWithdrawalProof: WithdrawalProof address can be updated by the admin.
- ManagementParity: ManagementParity address can be updated by the admin.
- updateToken: The admin should update the yield-bearing token address.
- updateAsset: The administrator can update the underlying asset that investors can deposit.
- updateEventBatchSize:EventBatchSize can be updated by the admin.

ManagementBeta.sol

- updateTreasury: The treasury address can be updated by the admin.
- updateSafeHouse: The SafeHouse address can be updated by the admin.

SafeHouseBeta.sol

- updateMaxWithdrawalCapacity: The administrator can set the maximum withdrawal capacity in USD.
- updateWithdrawalCapacity: The administrator can set the withdrawal capacity in USD.
- updatePriceToleranceRate: The administrator can set price tolerance rate value.
- updateAssetBook: AssetBook address can be updated by the admin.
- addVault: A new vault address can be added by the admin.

TokenBeta.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateHoldTime: The hold time value can be updated by the admin.
- addToWhiteList: Added `account_` to `whitelist ` by the admin.

- removeFromWhiteList: Remove `account_` from `whitelist ` by the admin.

WithdrawalProofBeta.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

AssetBookGamma.sol

- updateAsset: The asset address, oracle address and price can be updated by the admin.

DepositProofGamma.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

HoldTimeGamma.sol

- updateToken: The Token address can be updated by the owner.
- updateHoldTime:The Hold time can be updated by the owner.

InvestmentGamma.sol

- updateManagement: The management address can be updated by the admin.
- updateDepositProof: The Deposit Proof address can be updated by the admin.
- updateWithdrawalProof: WithdrawalProof address can be updated by the admin.
- ManagementParity: ManagementParity address can be updated by the admin.
- updateToken: The admin should update the yield-bearing token address.
- updateAsset: The administrator can update the underlying asset that investors can deposit.
- updateEventBatchSize:EventBatchSize can be updated by the admin.

ManagementGamma.sol

- updateTreasury: The treasury address can be updated by the admin.
- updateSafeHouse: The SafeHouse address can be updated by the admin.

SafeHouseGamma.sol

- updateMaxWithdrawalCapacity: The administrator can set the maximum withdrawal capacity in USD.
- updateWithdrawalCapacity: The administrator can set the withdrawal capacity in USD.
- updatePriceToleranceRate: The administrator can set price tolerance rate value.
- updateAssetBook: AssetBook address can be updated by the admin.
- addVault: A new vault address can be added by the admin.

TokenGamma.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateHoldTime: The hold time value can be updated by the admin.
- addToWhiteList: Added `account_` to `whitelist ` by the admin.
- removeFromWhiteList: Remove `account_` from `whitelist ` by the admin.

WithdrawalGamma.sol

- updateInvestment: Investment addresses can be updated by the admin.
- updateMetadata: Metadata addresses can be updated by the admin.
- setBaseURI: BaseURI can be set by the admin.
- updateTolerance: The tolerance value can be updated by the admin.
- updateIsOnChainMetadata: IsOnChainMetadata status can be set by the admin.

To make the smart contract 100% decentralized, we suggest renouncing ownership in the smart contract once its function is completed.

Conclusion

We were given a contract code in the form of a file. And we have used all possible tests based on given objects as files. We had observed 2 low severity issues in the smart contracts. We confirm that 2 low severity issues are acknowledged in the revised smart contract code. **So, the smart contracts are ready for the mainnet deployment**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high-level description of functionality was presented in the As-is overview section of the report.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

The security state of the reviewed contract, based on standard audit procedure scope, is **"Secured"**.

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review:

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis:

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results:

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

Suggested Solutions:

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

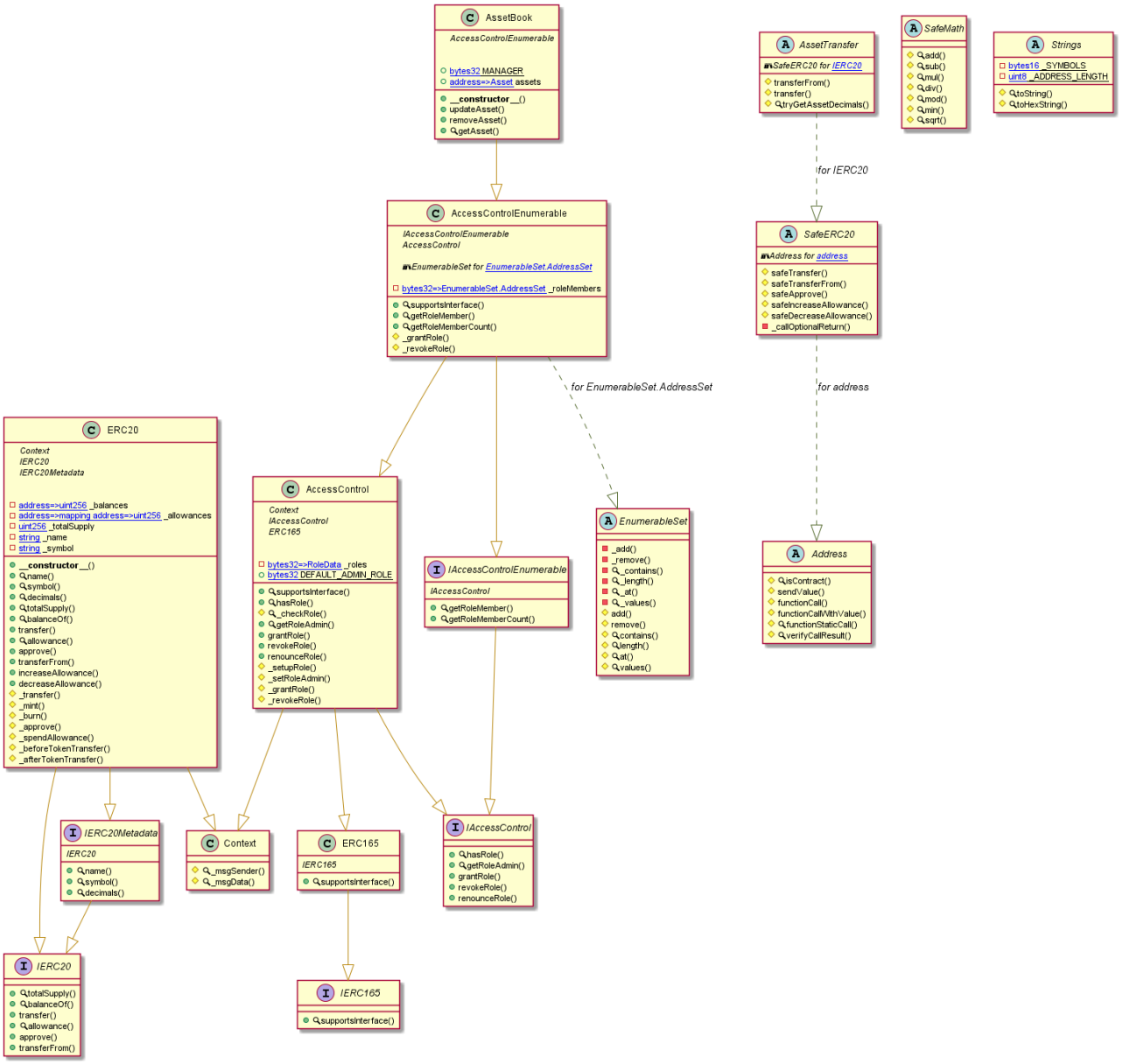
Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can't guarantee explicit security of the audited smart contracts.

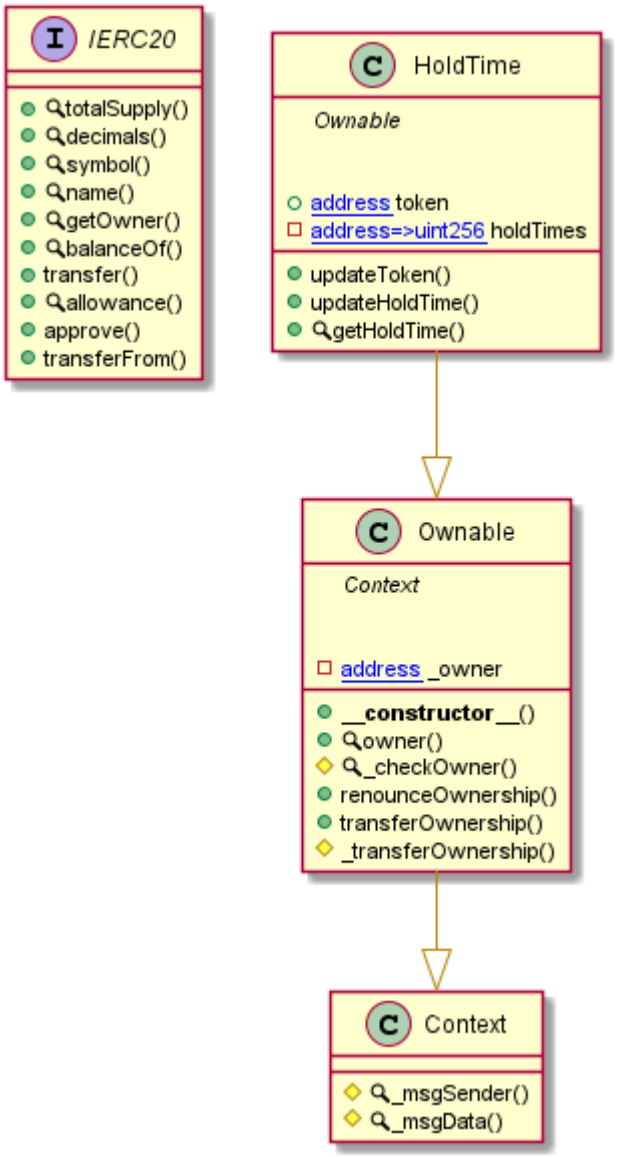
Appendix

Code Flow Diagram - Every Finance

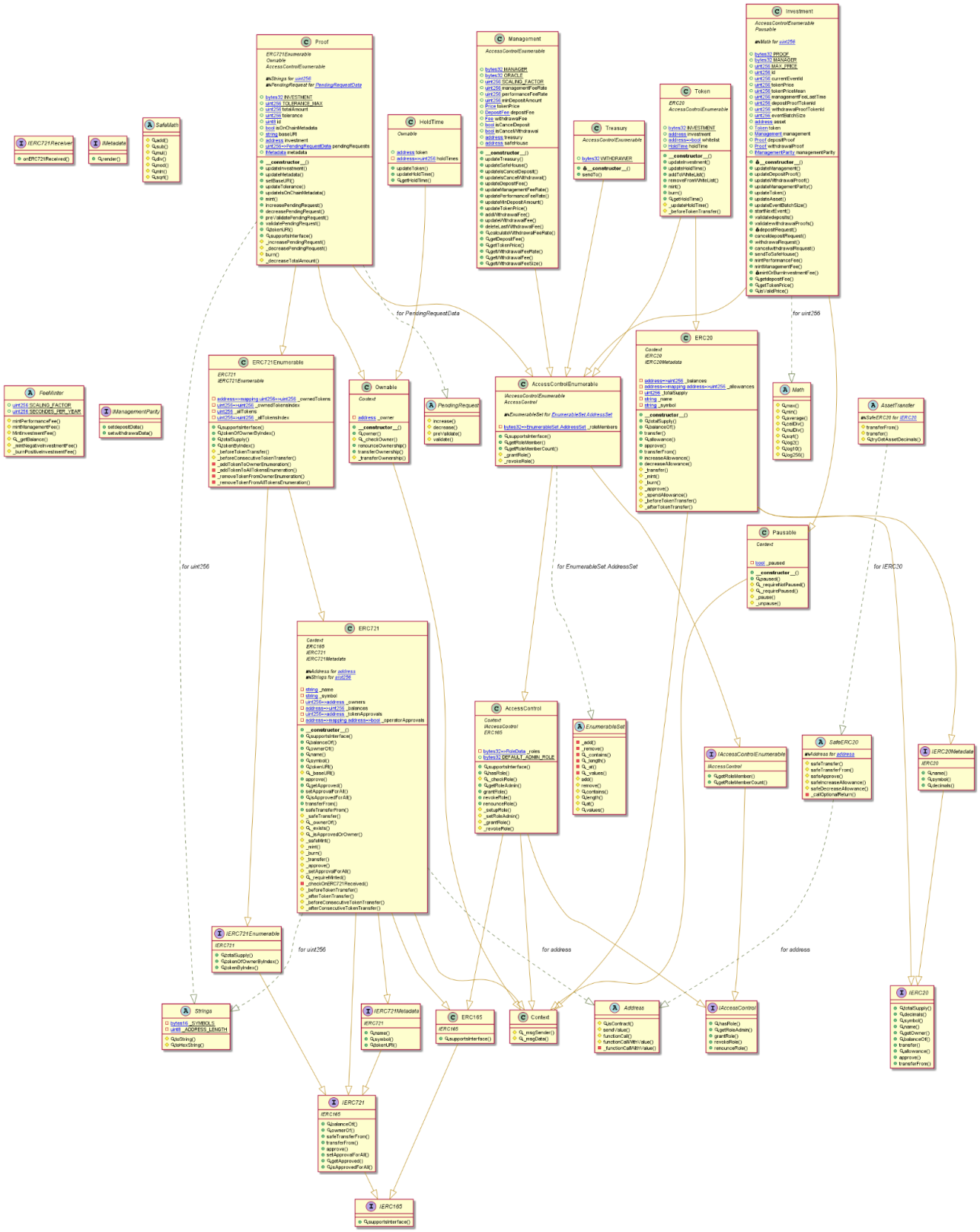
AssetBook Diagram



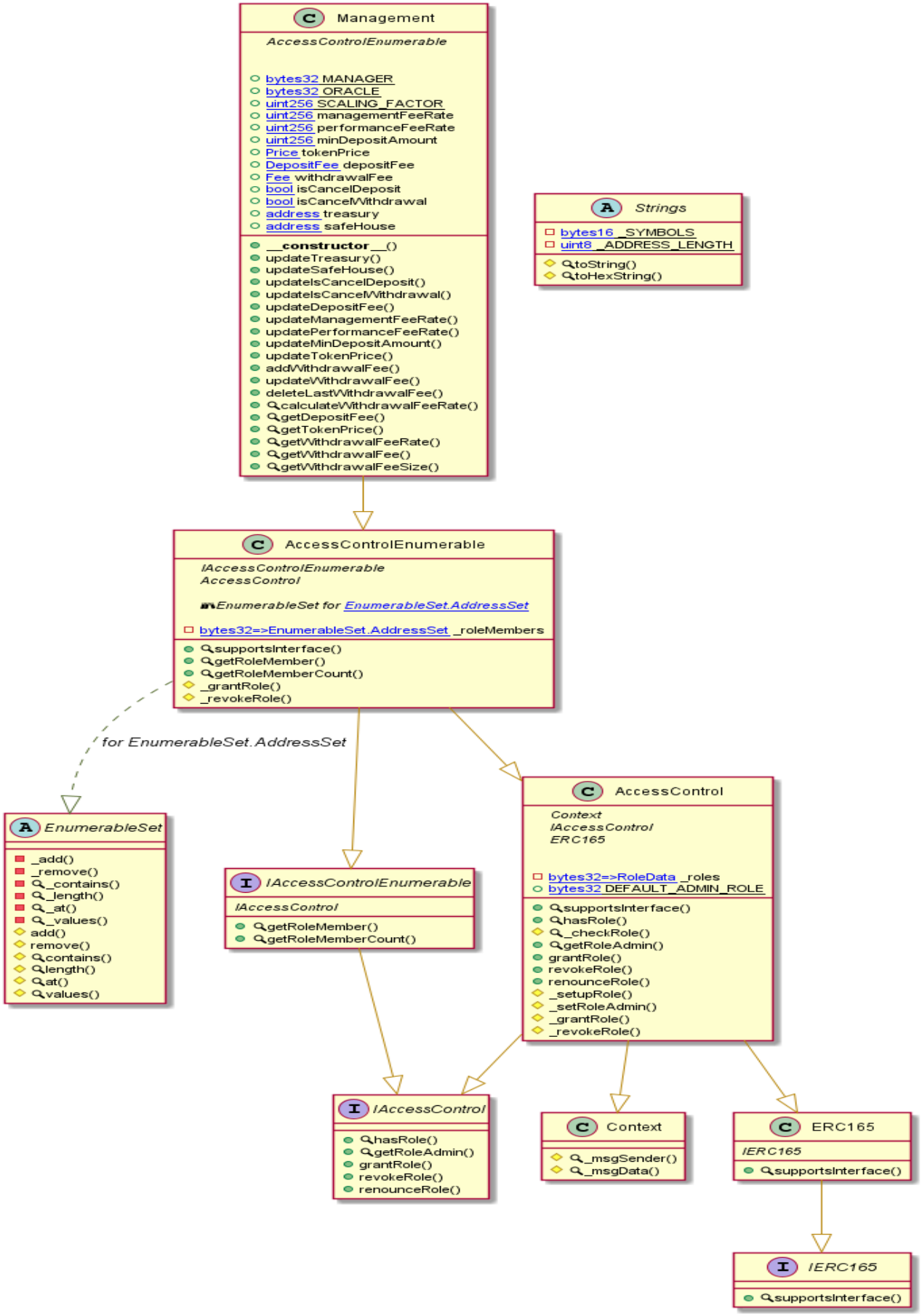
HoldTime Diagram



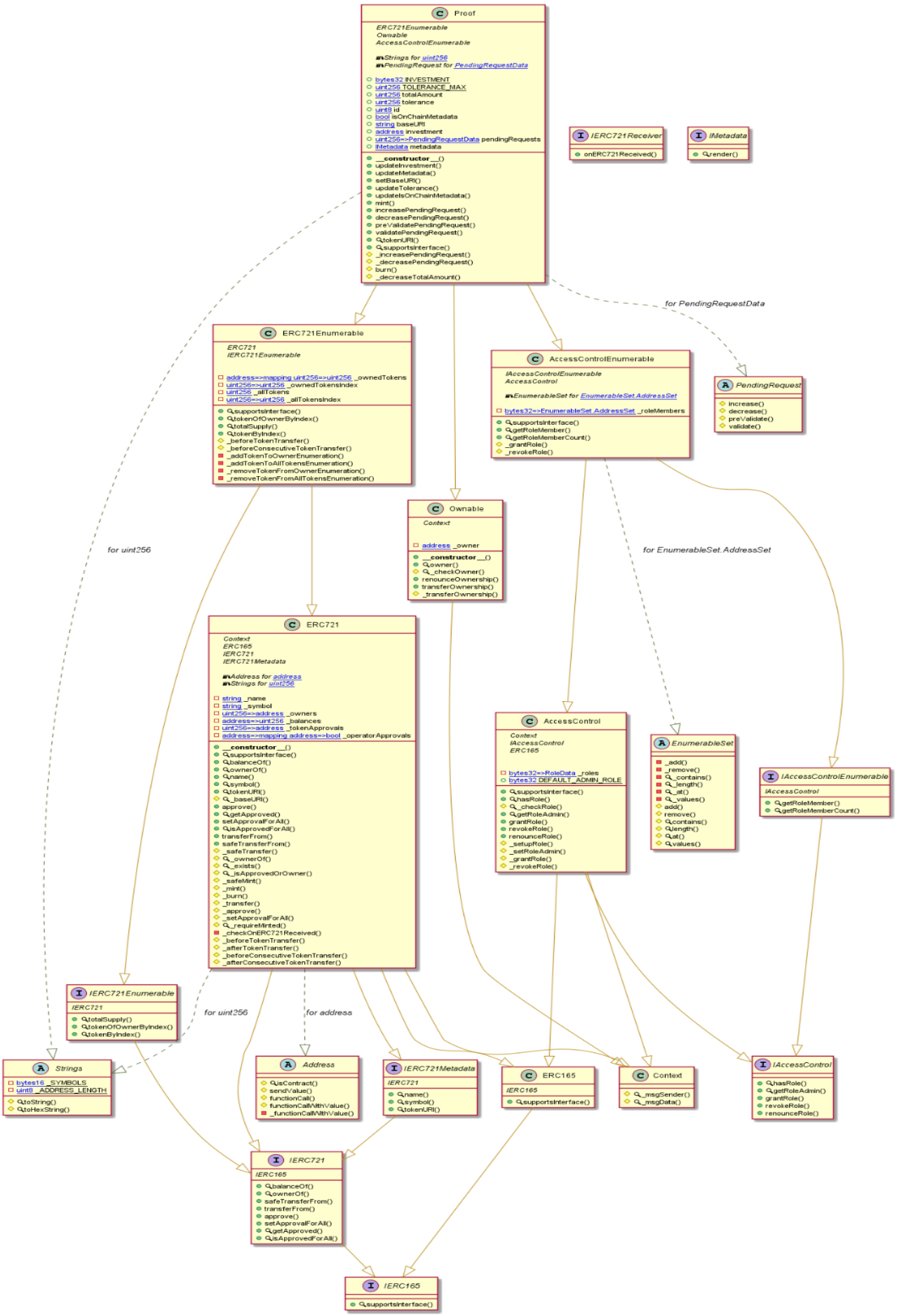
Investment Diagram



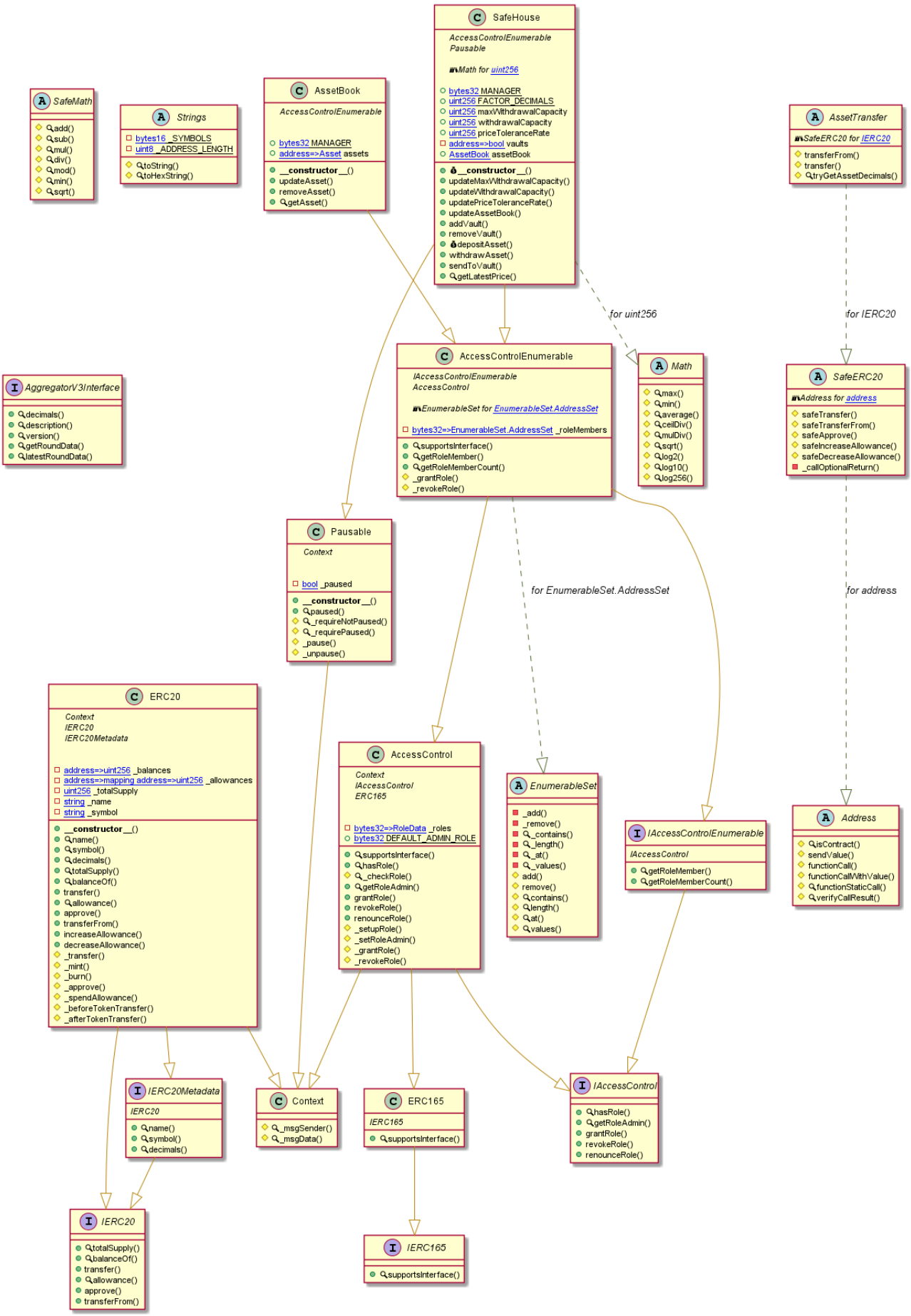
Management Diagram



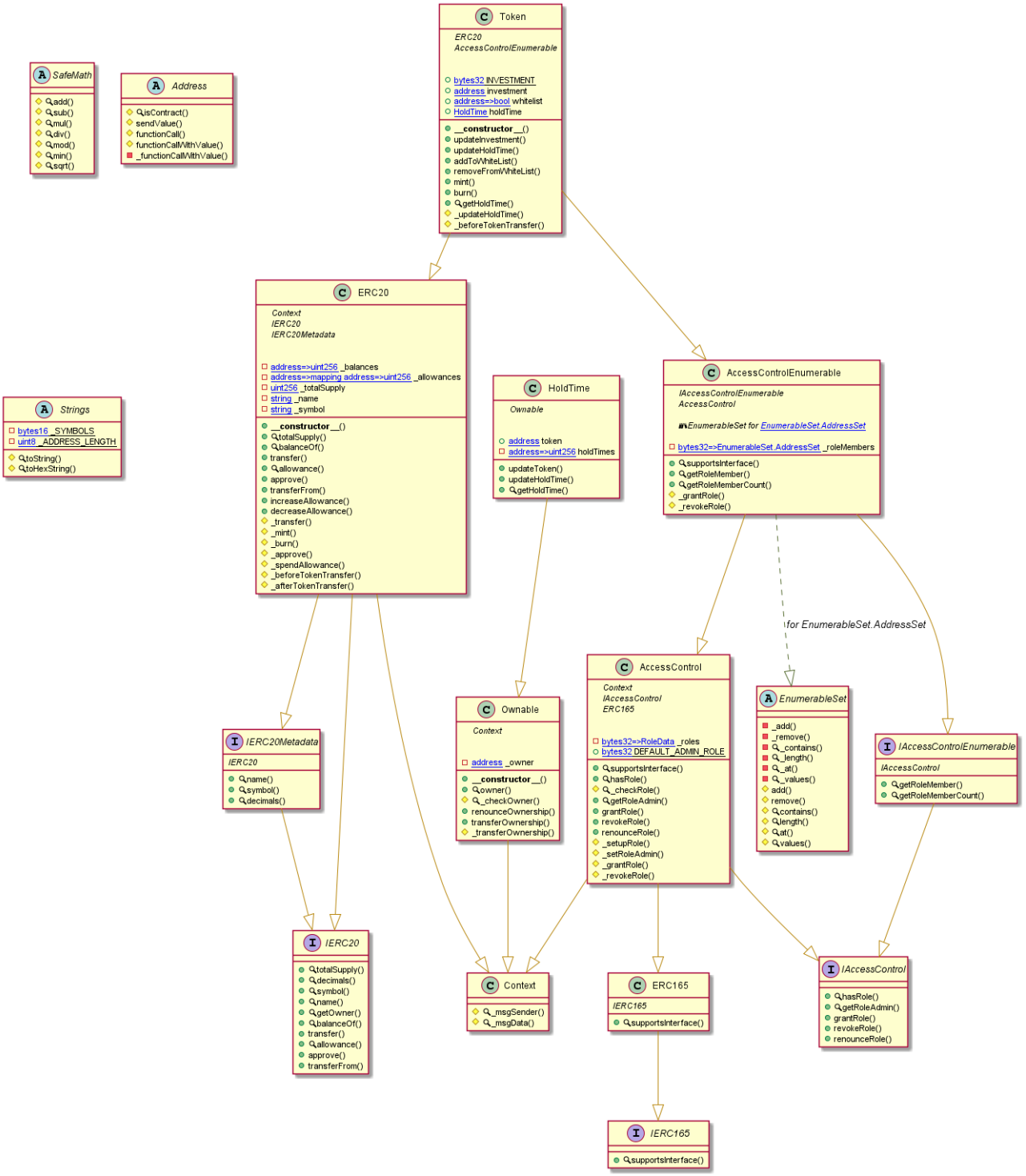
Proof Diagram



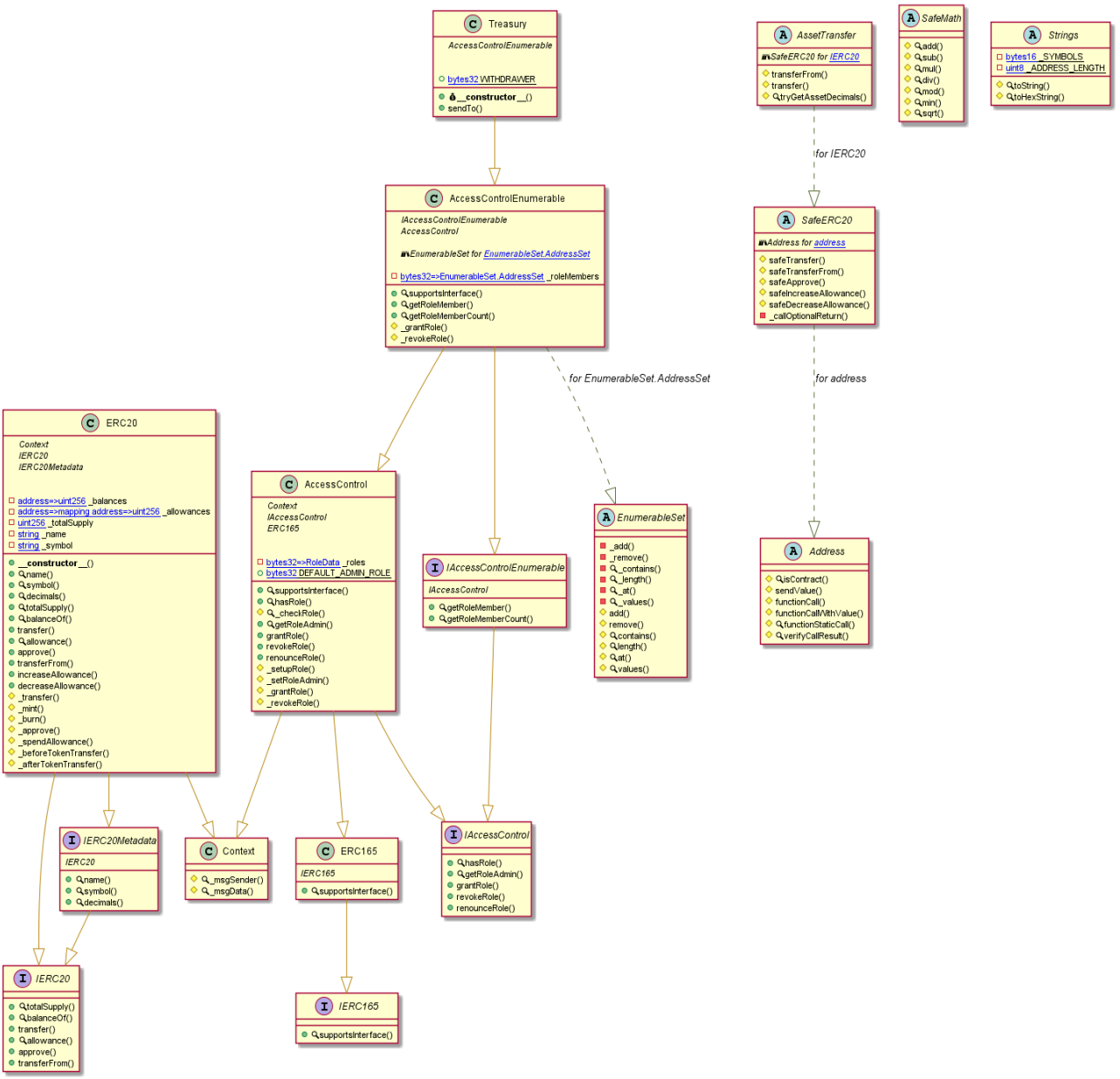
SafeHouse Diagram



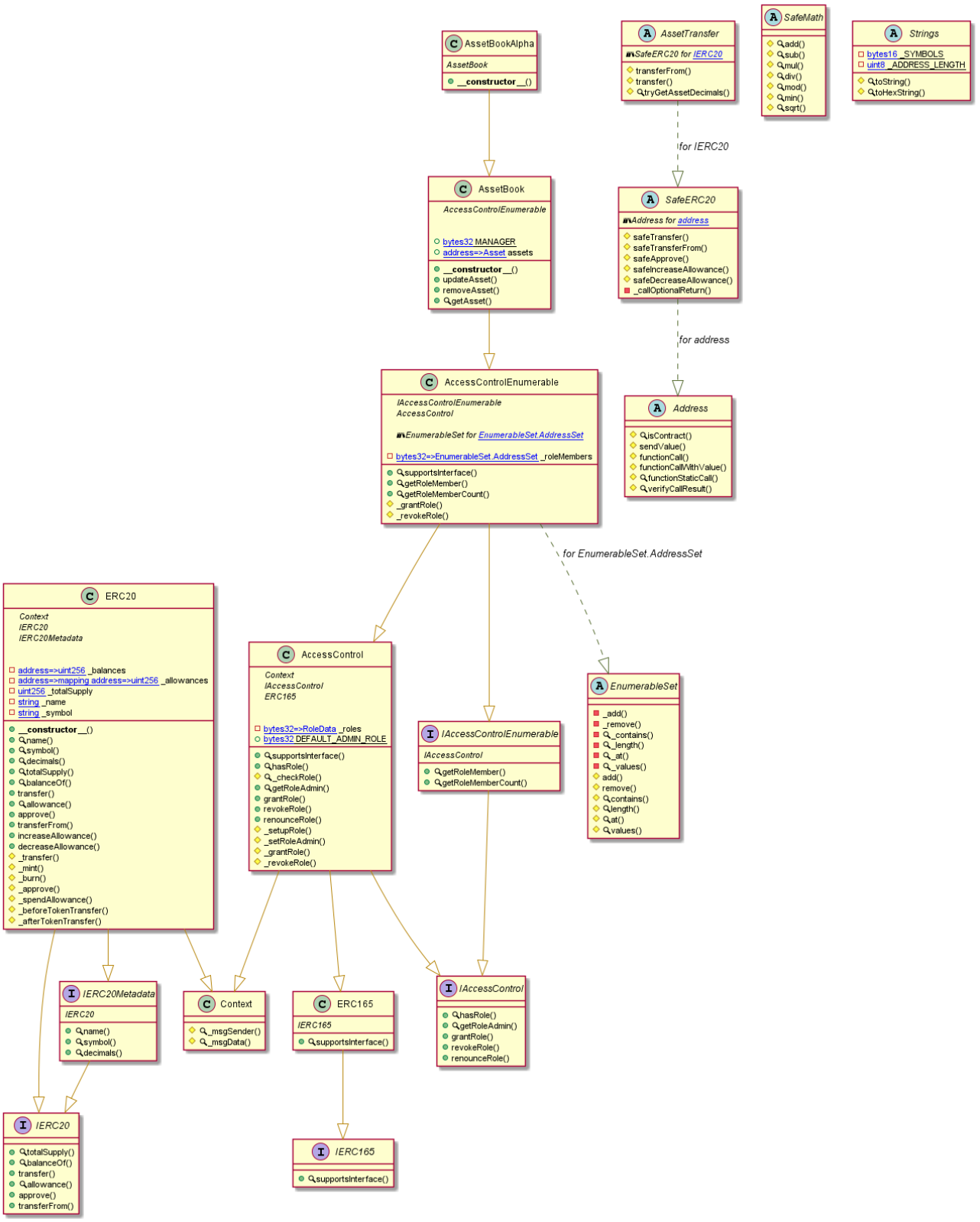
Token Diagram



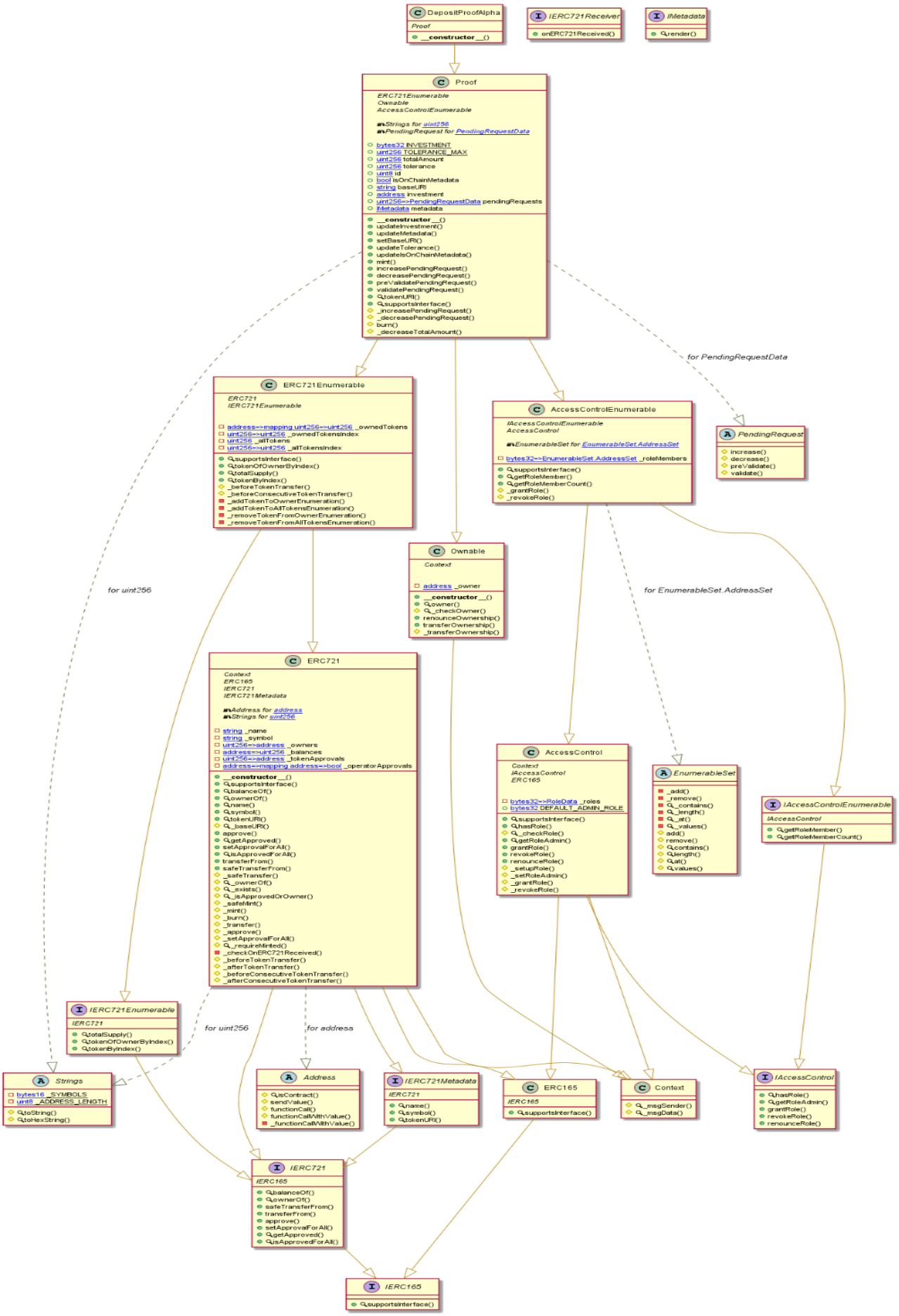
Treasury Diagram



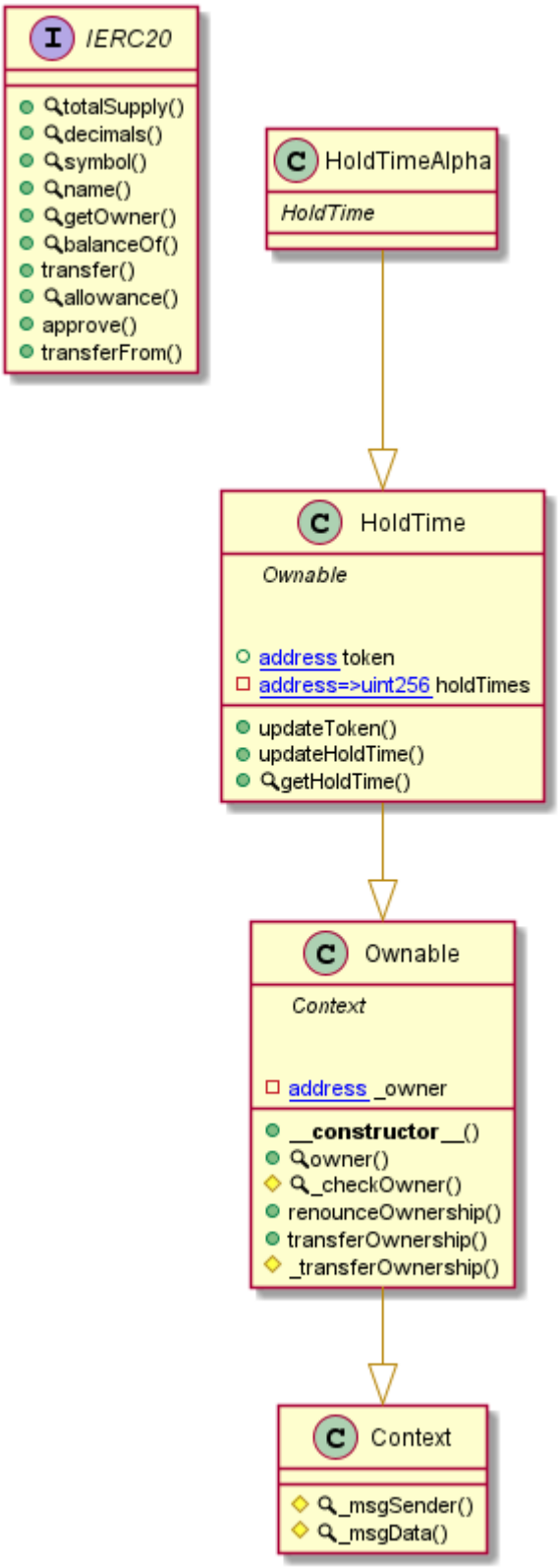
AssetBookAlpha Diagram



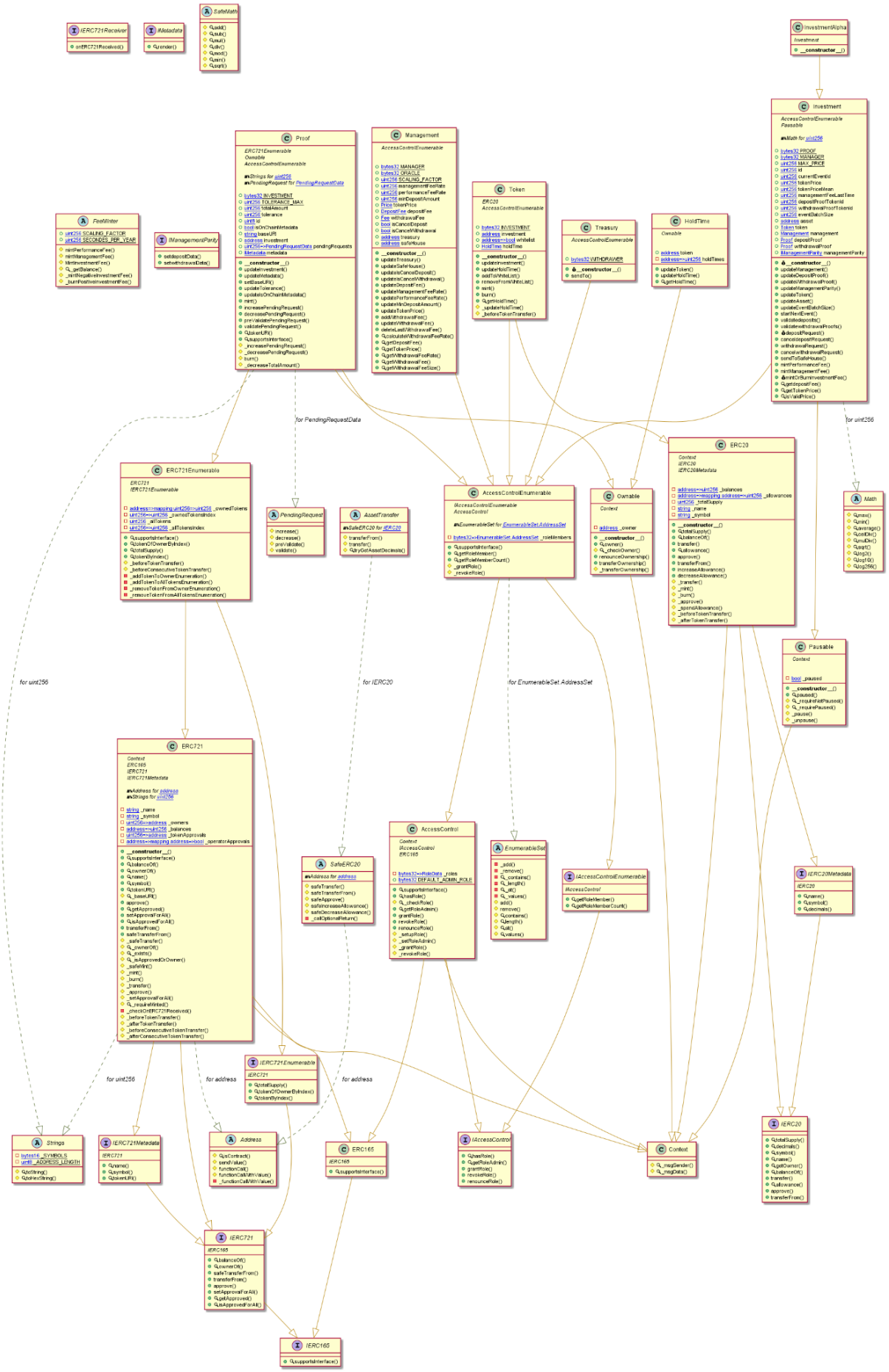
DepositProofAlpha Diagram



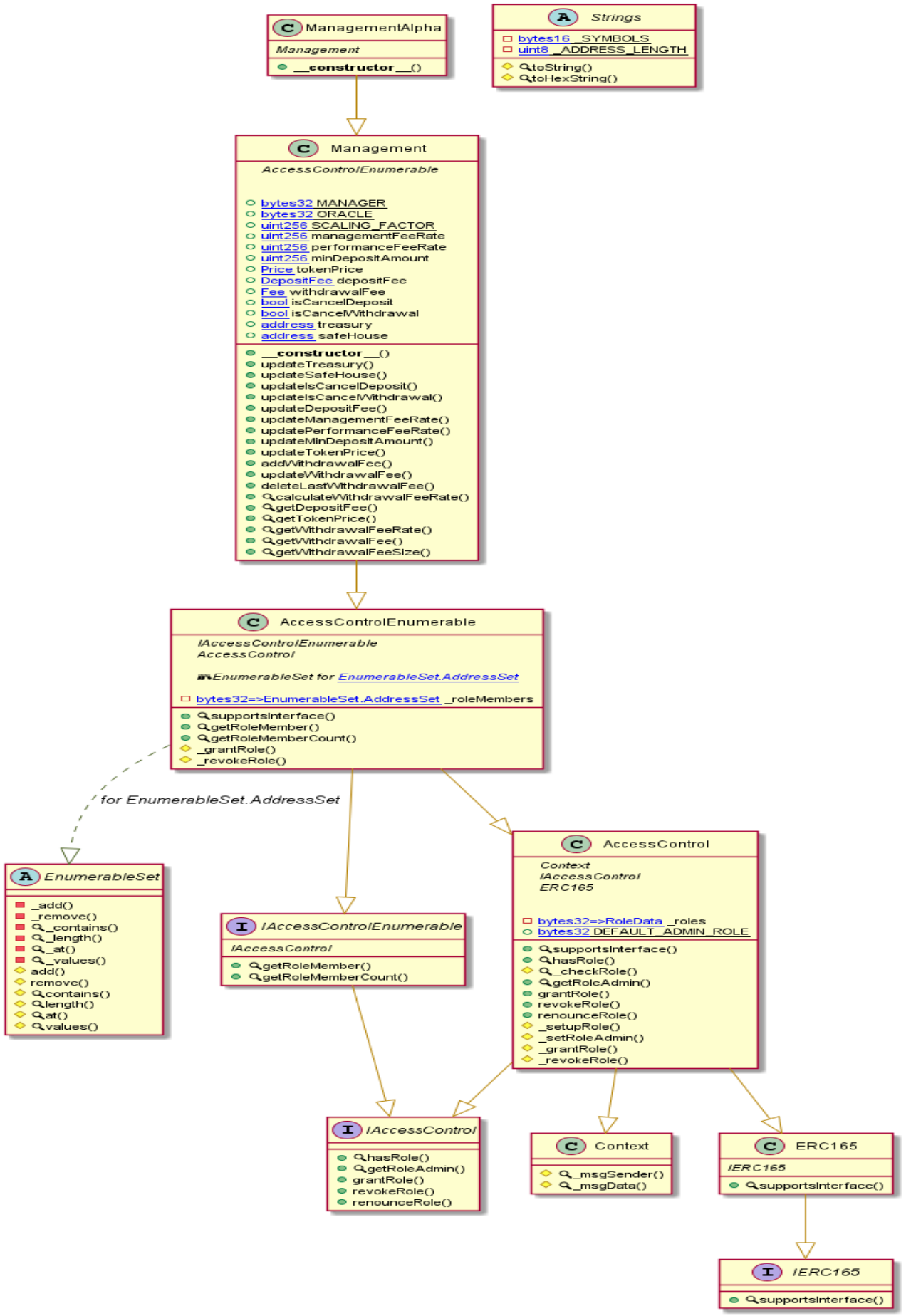
HoldTimeAlpha Diagram



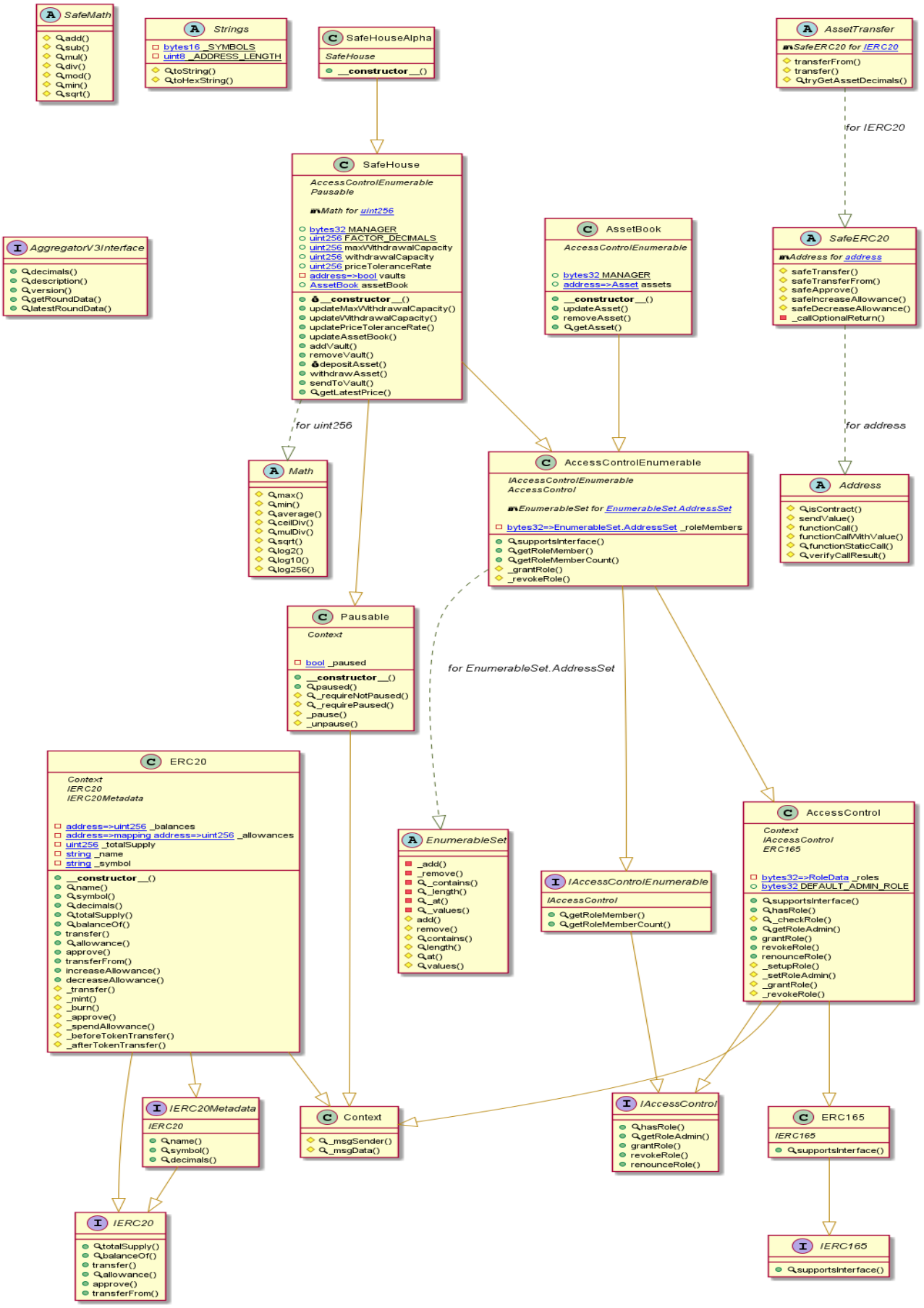
InvestmentAlpha Diagram



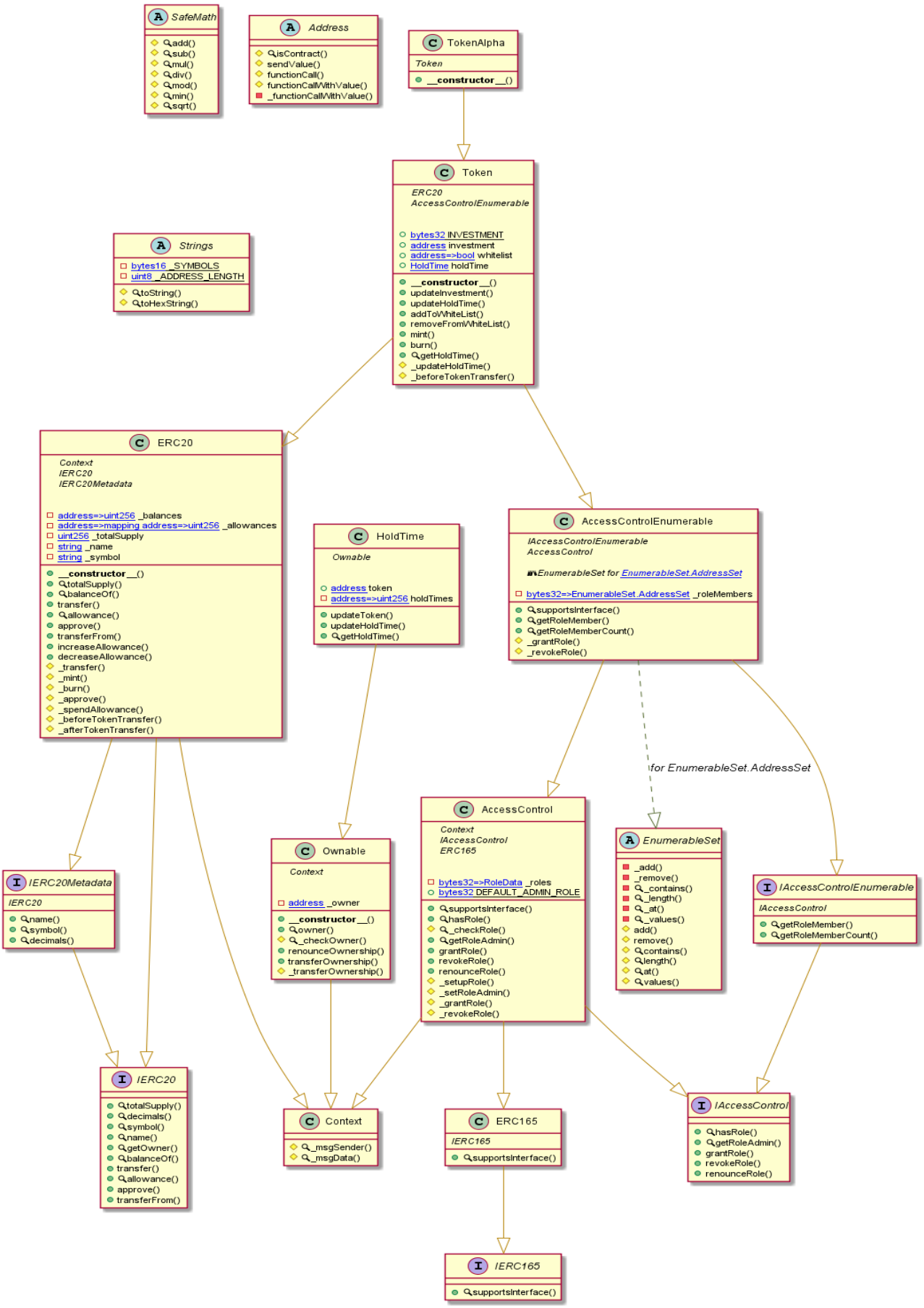
ManagementAlpha Diagram



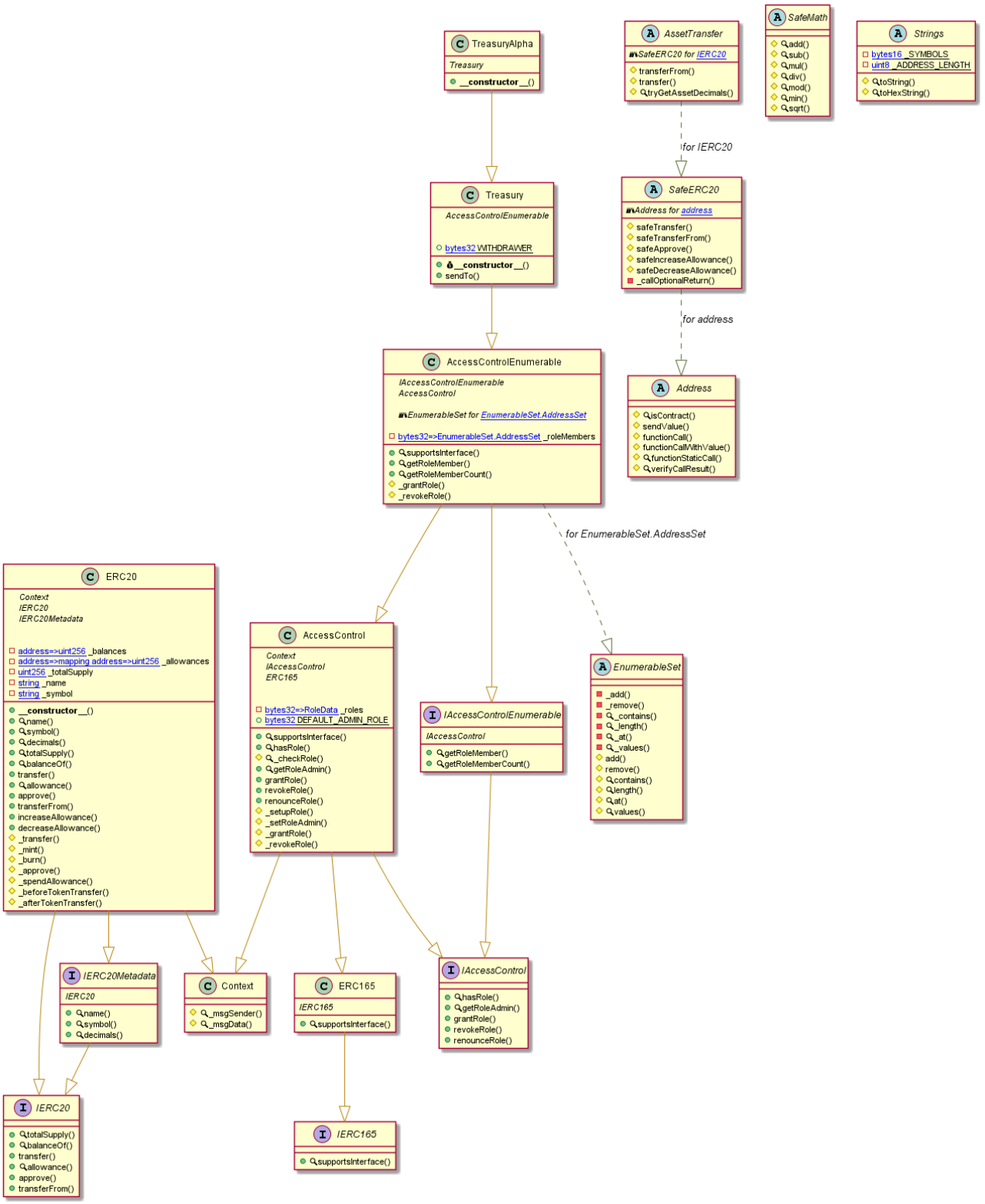
SafeHouseAlpha Diagram



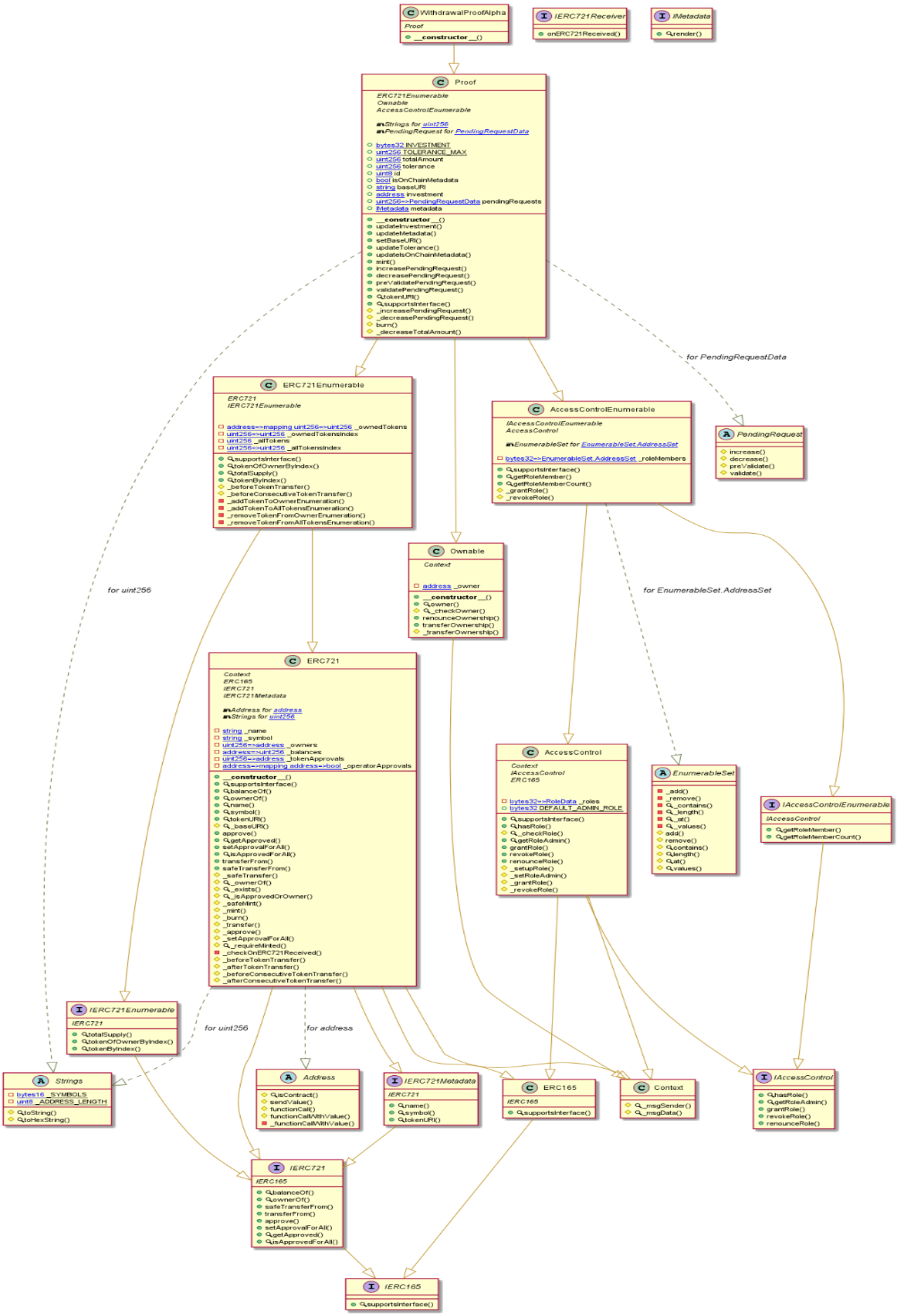
TokenAlpha Diagram



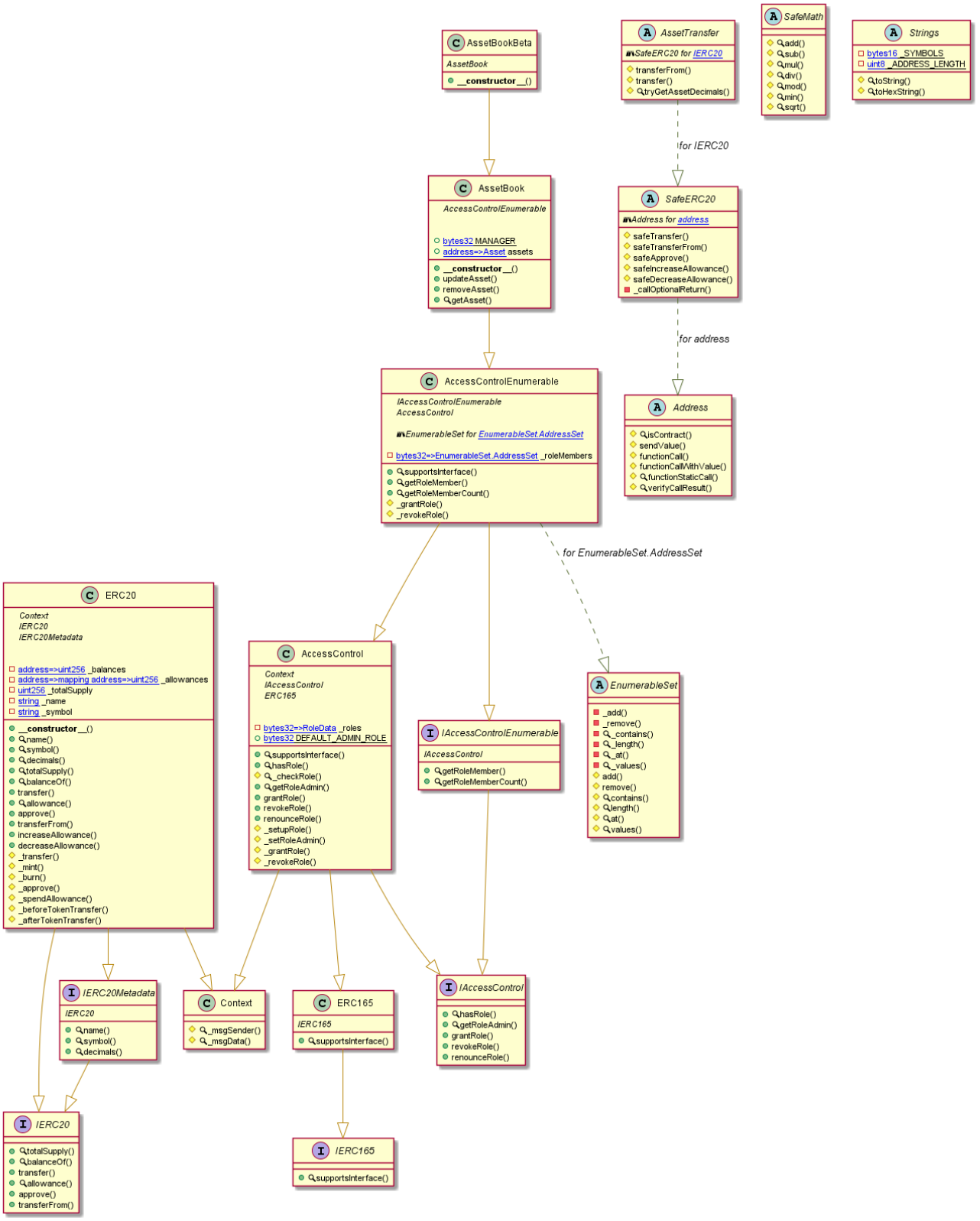
TreasuryAlpha Diagram



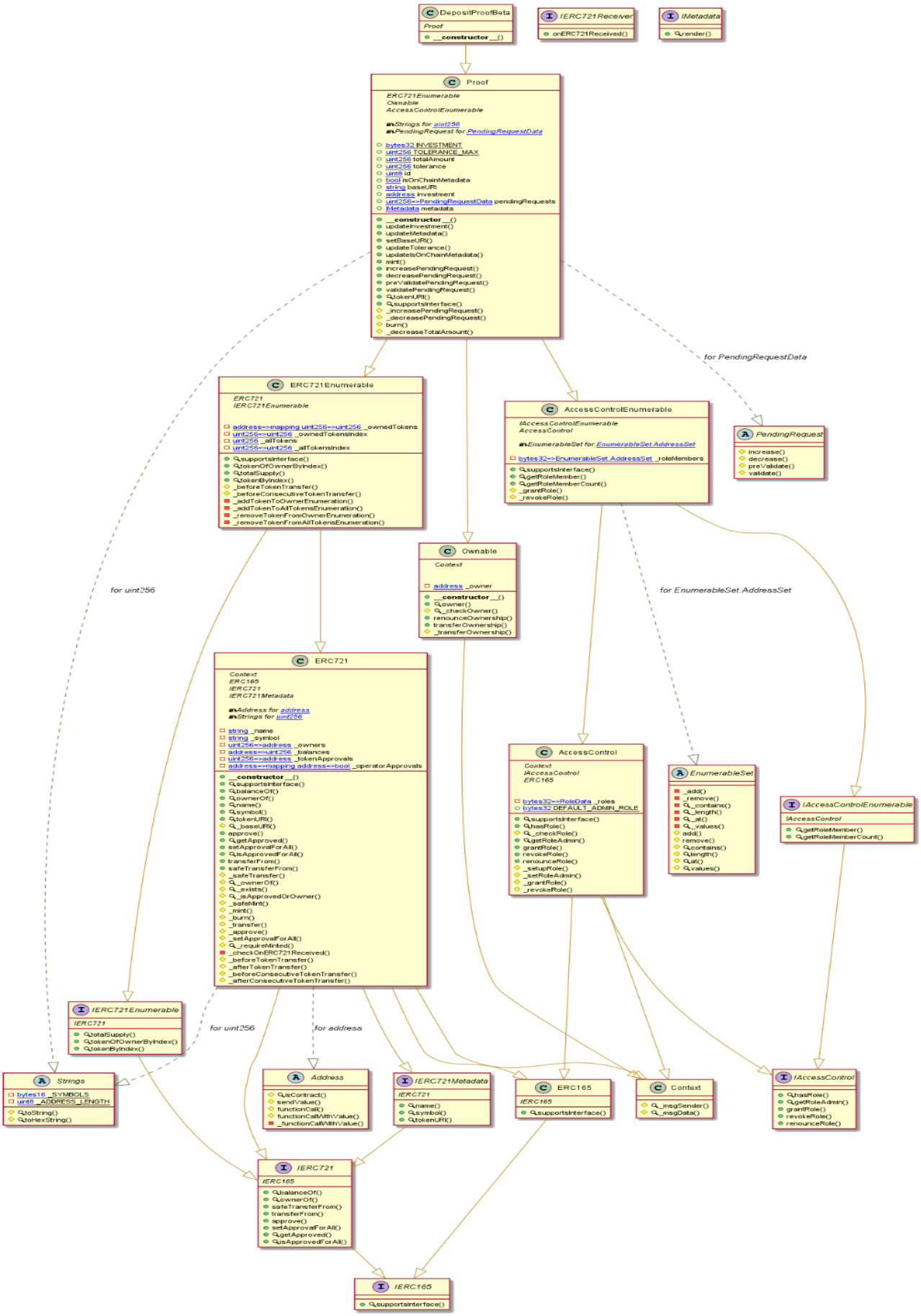
WithdrawalProofAlpha Diagram



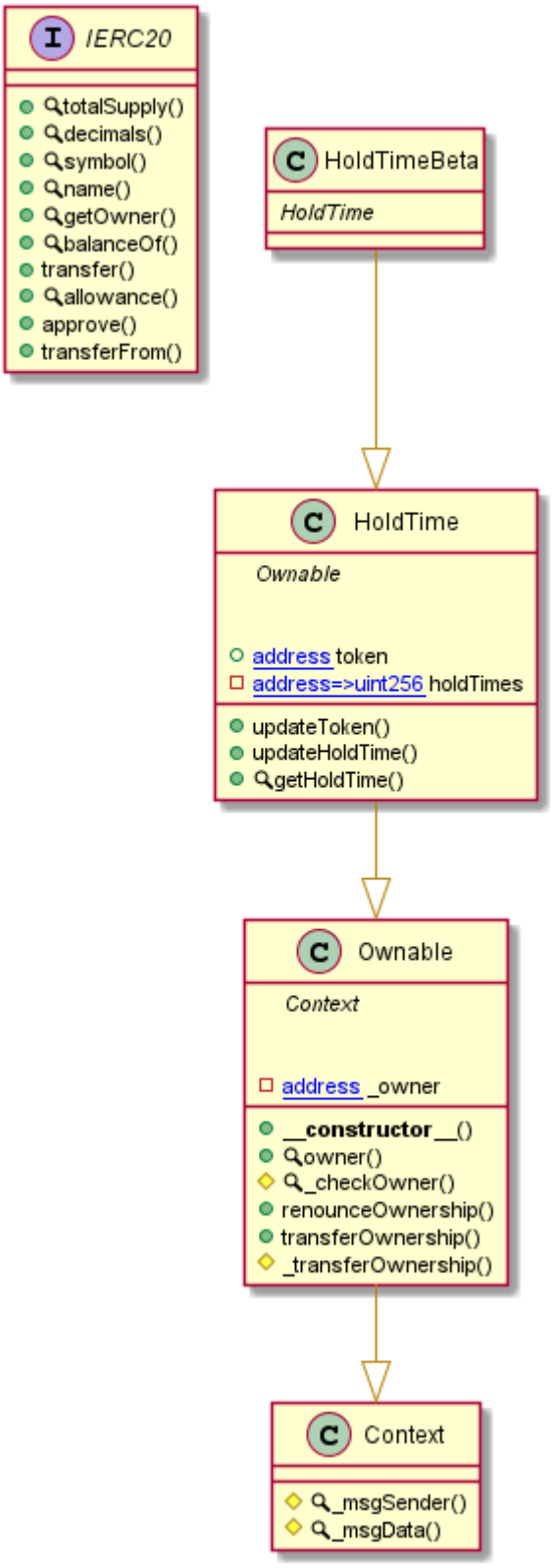
AssetBookBeta Diagram



DepositProofBeta Diagram

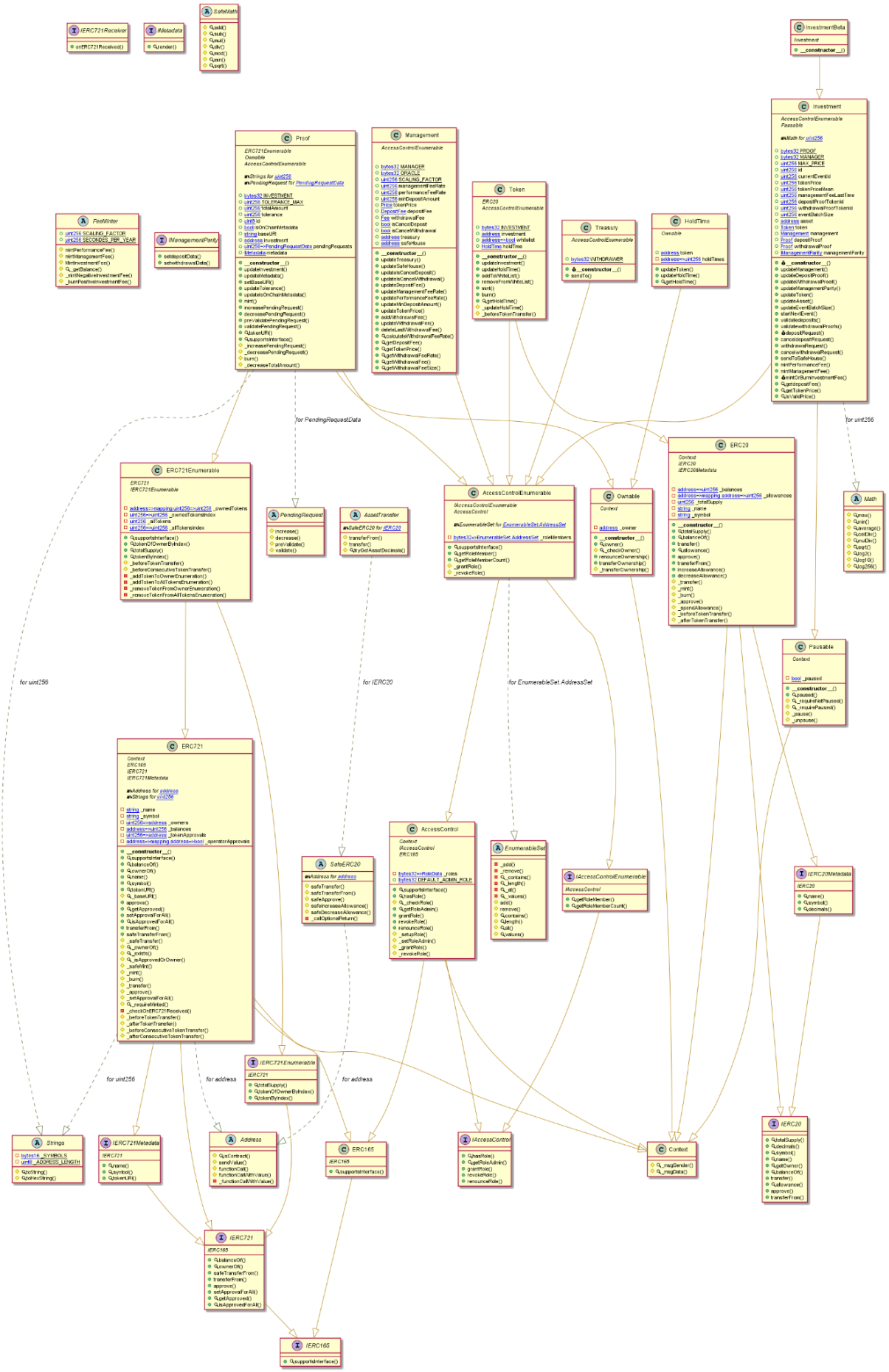


HoldTimeBeta Diagram

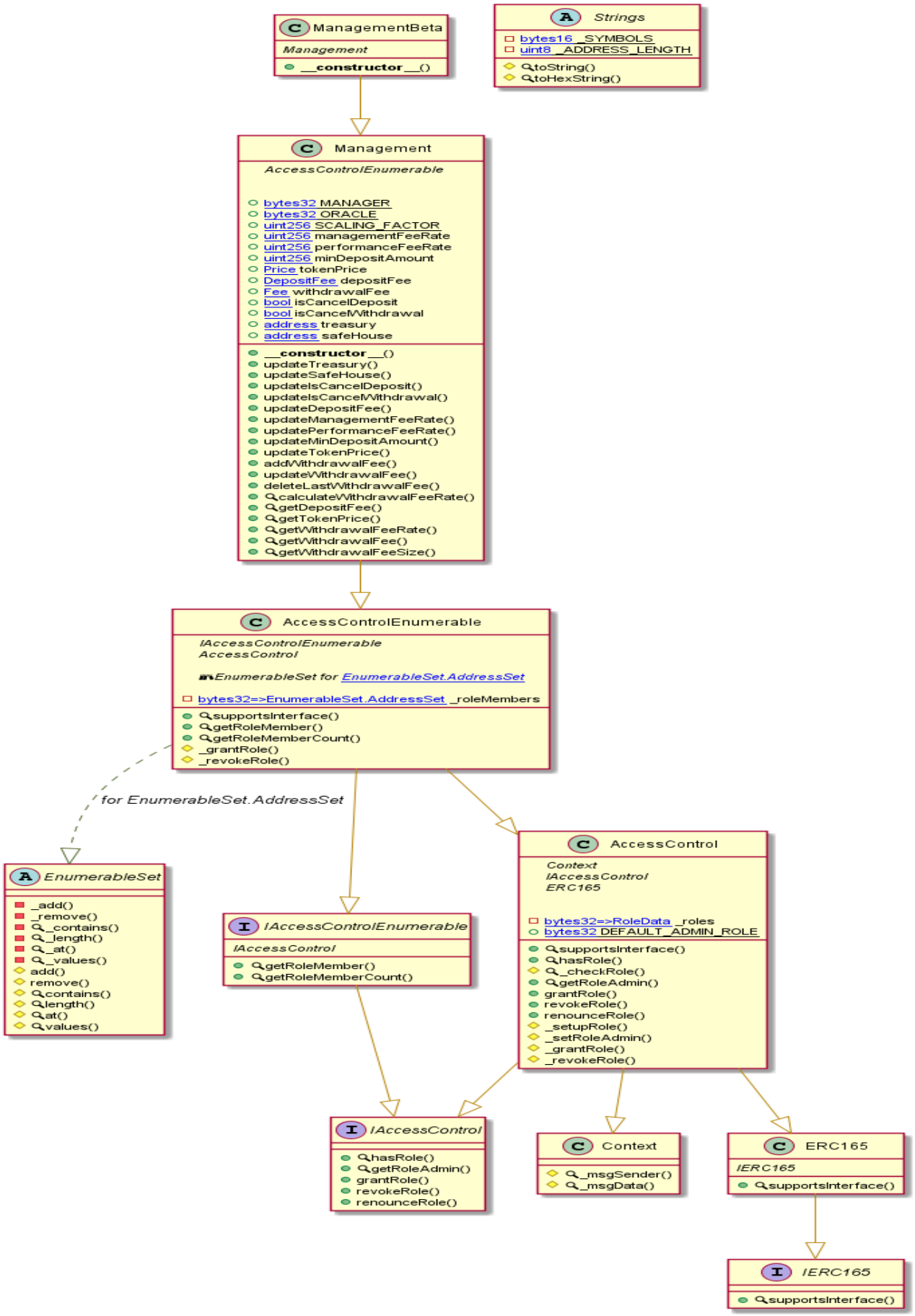


Email: audit@EtherAuthority.io

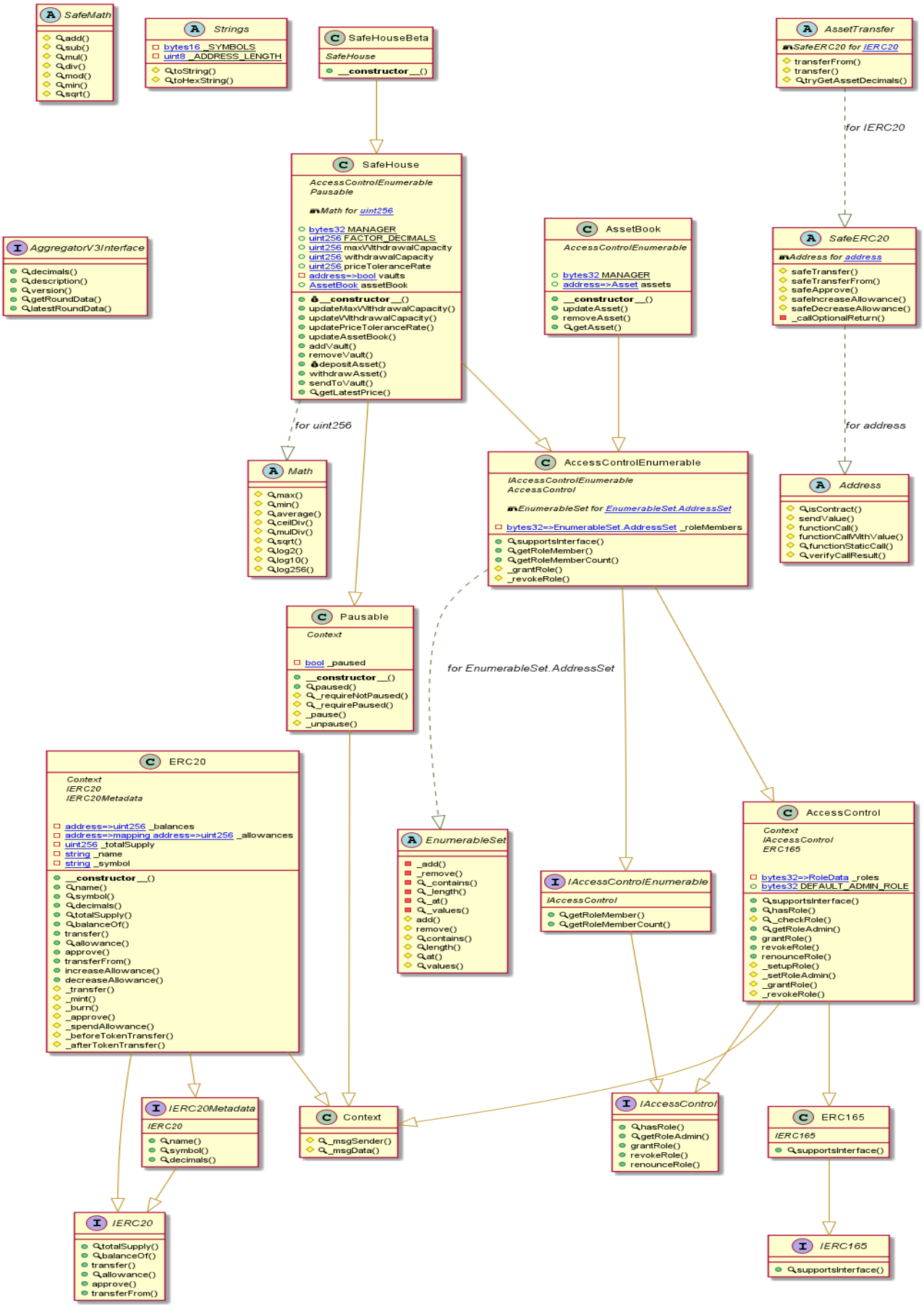
InvestmentBeta Diagram



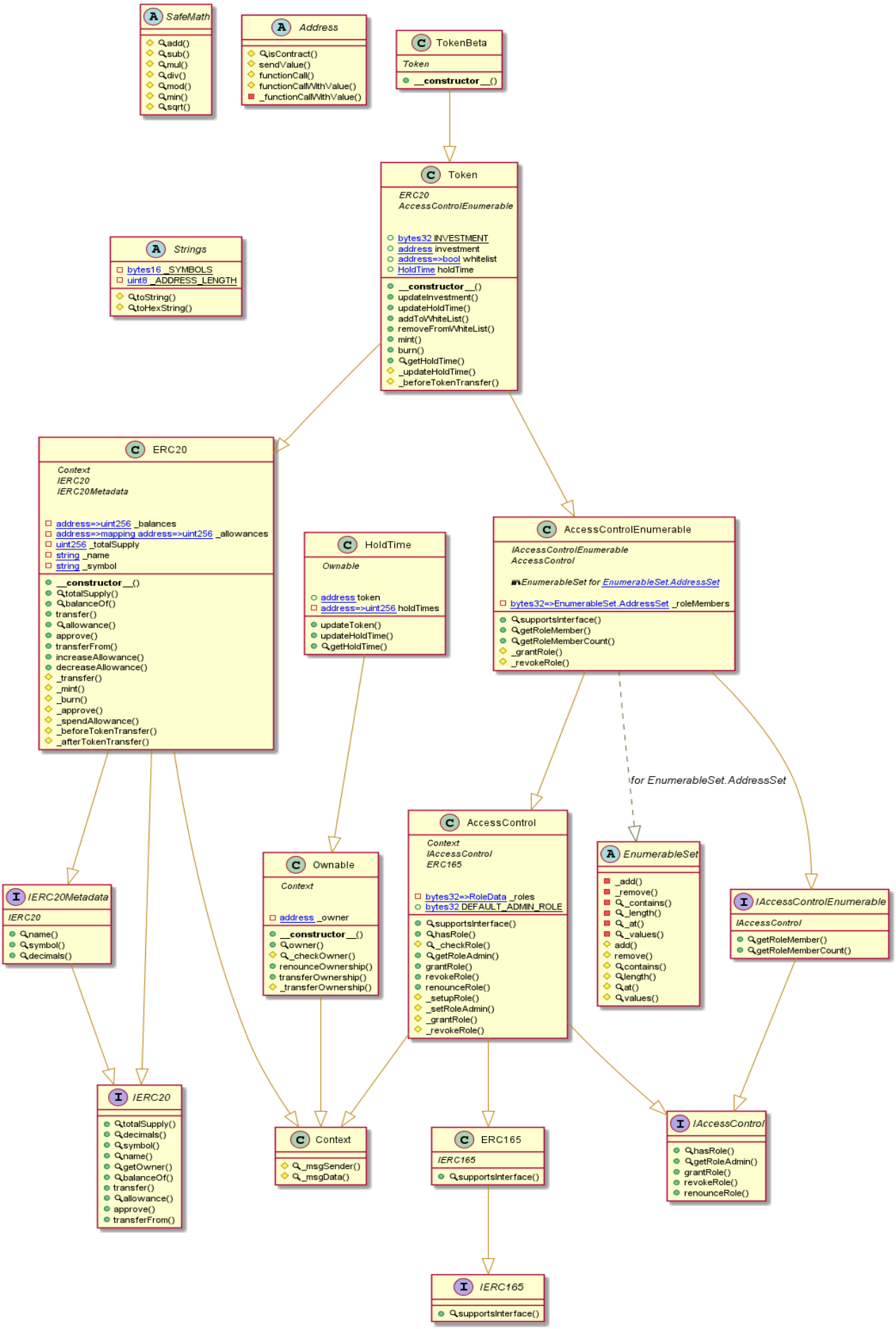
ManagementBeta Diagram



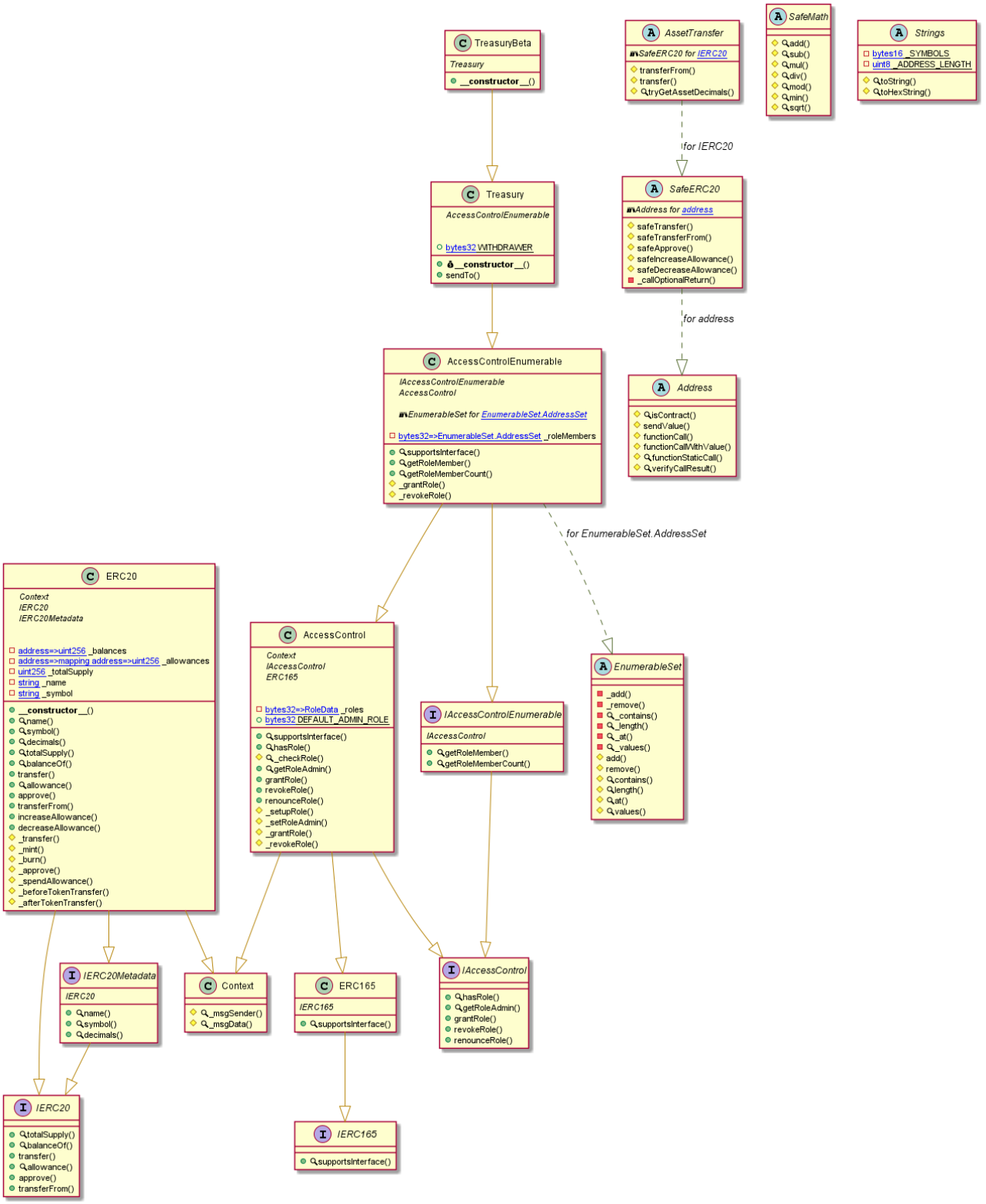
SafeHouseBeta Diagram



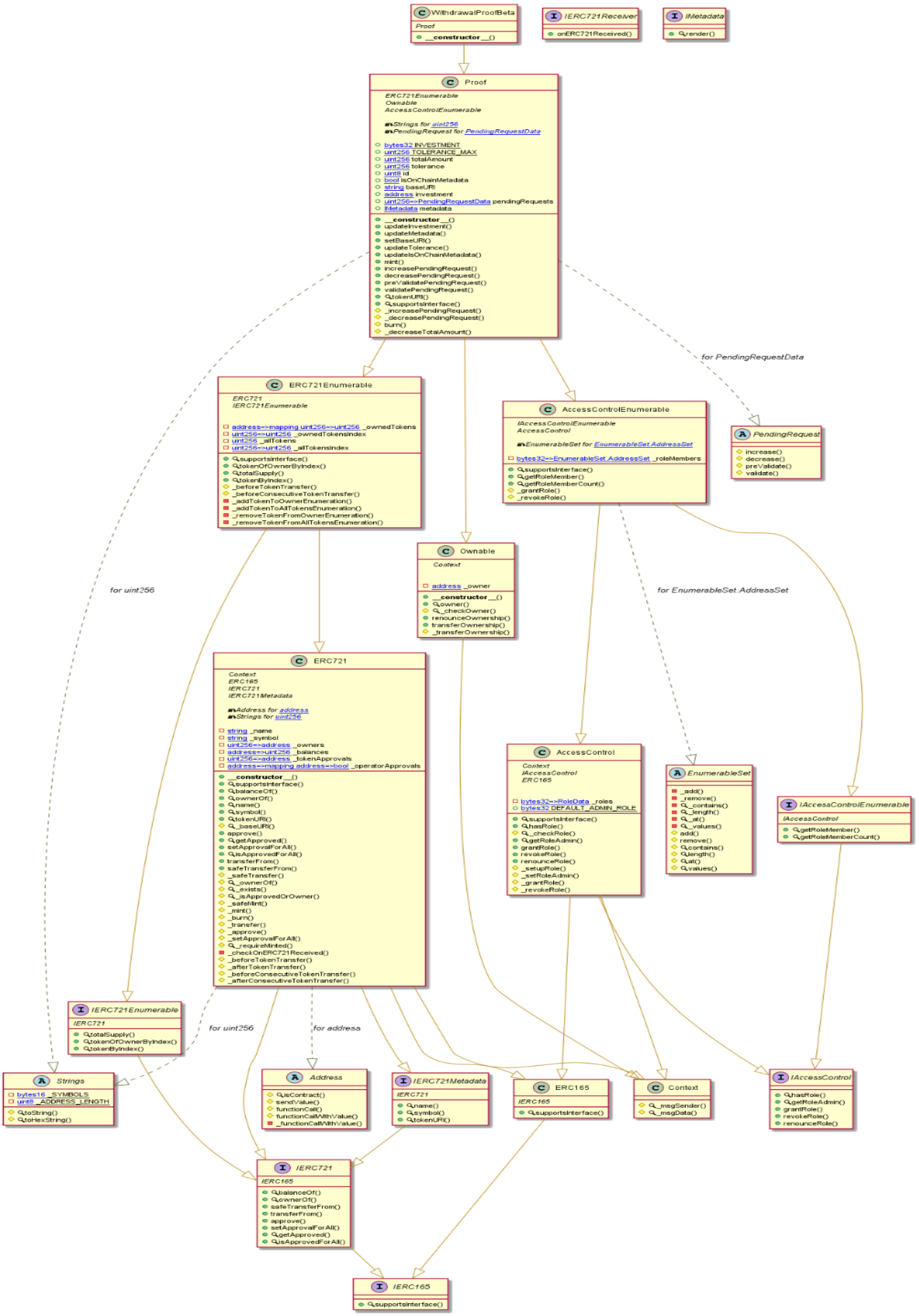
TokenBeta Diagram



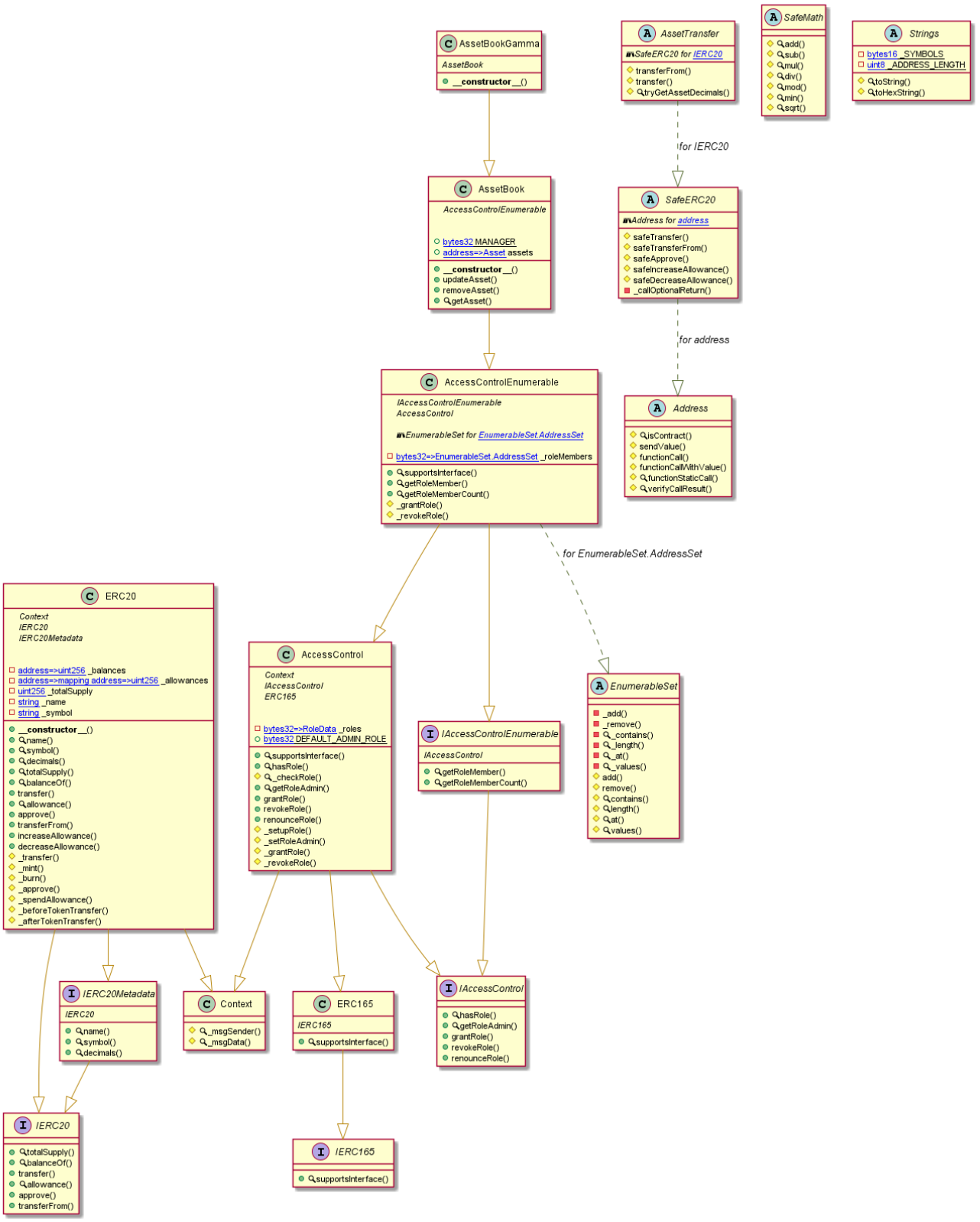
TreasuryBeta Diagram



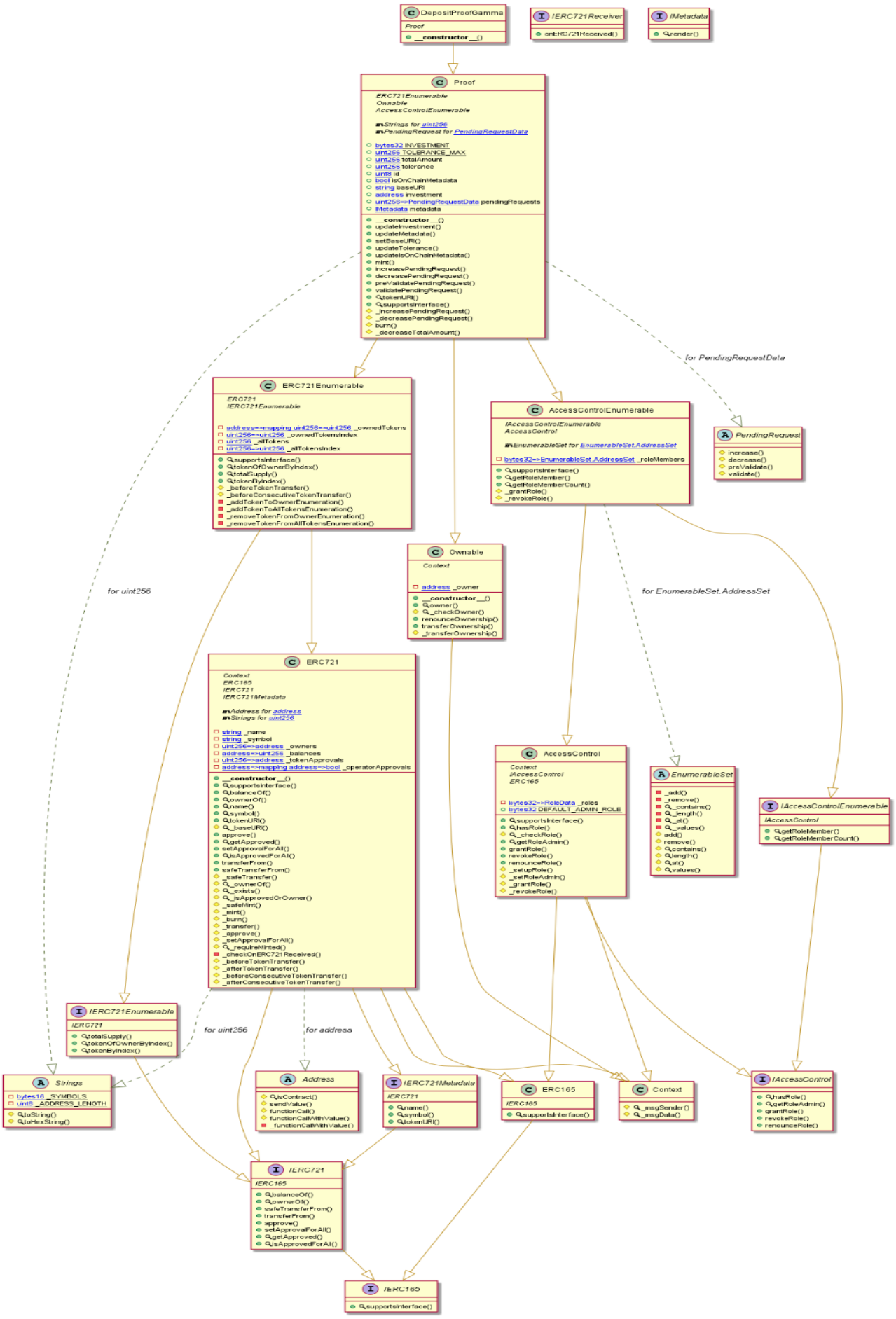
WithdrawalProofBeta Diagram



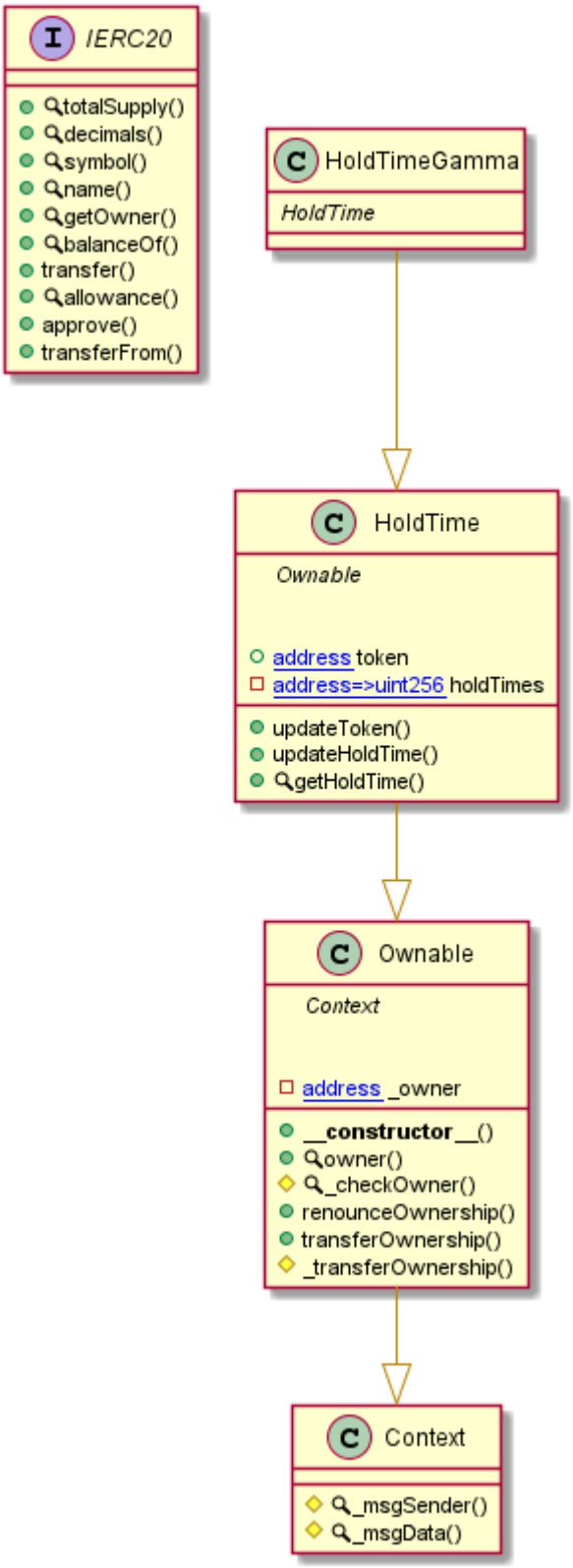
AssetBookGamma Diagram



DepositProofGamma Diagram

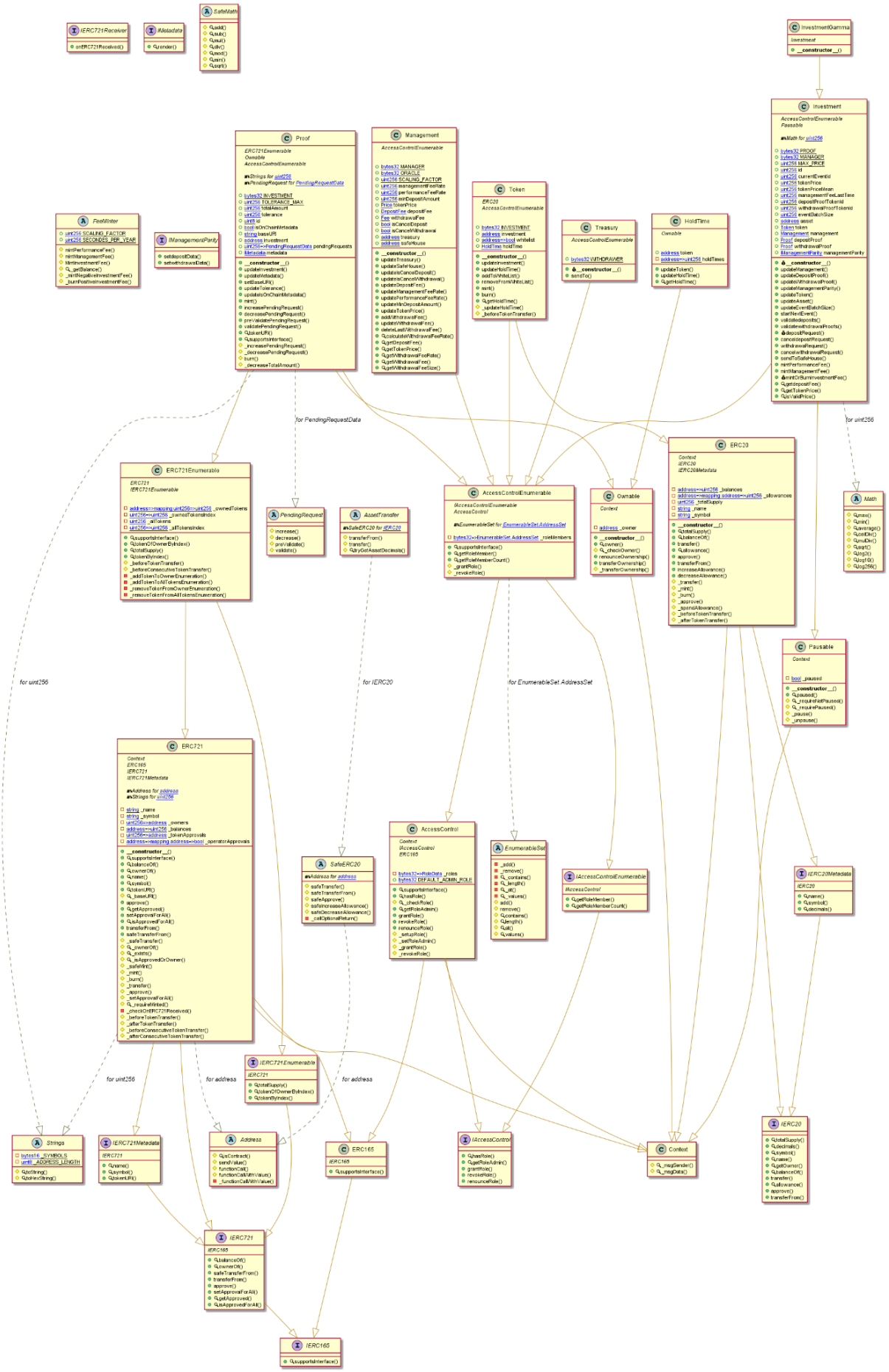


HoldTimeGamma Diagram

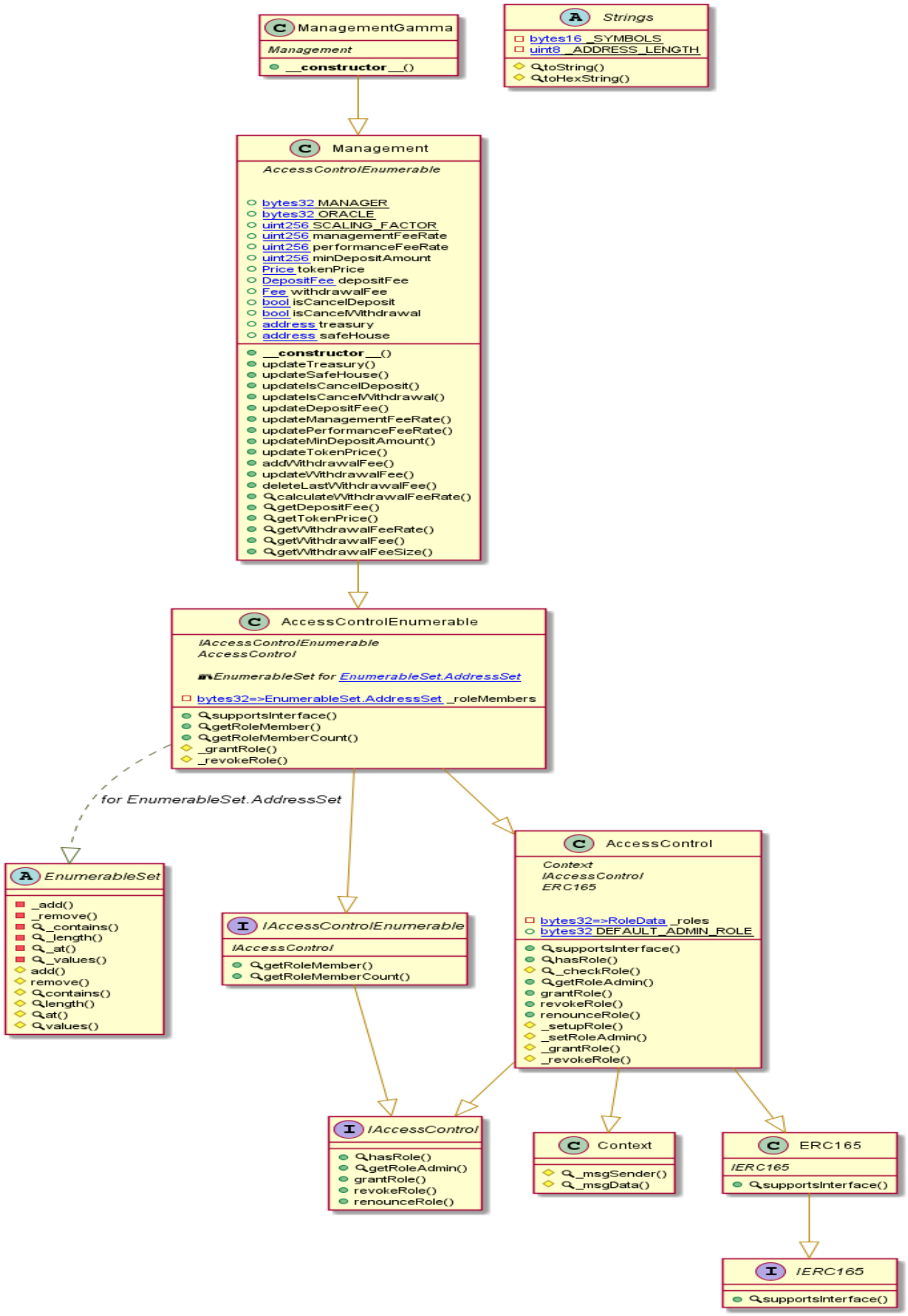


Email: audit@EtherAuthority.io

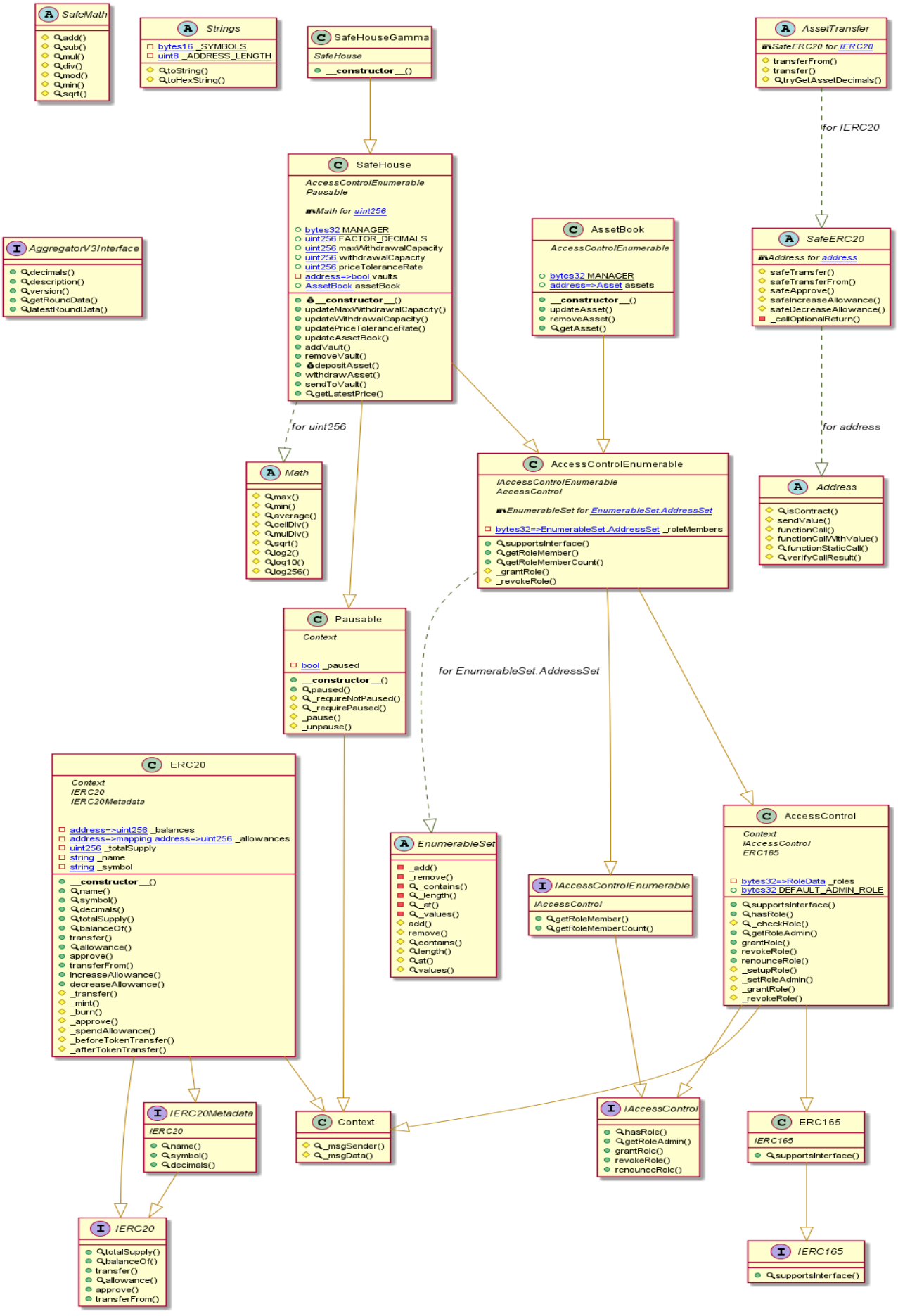
InvestmentGamma Diagram



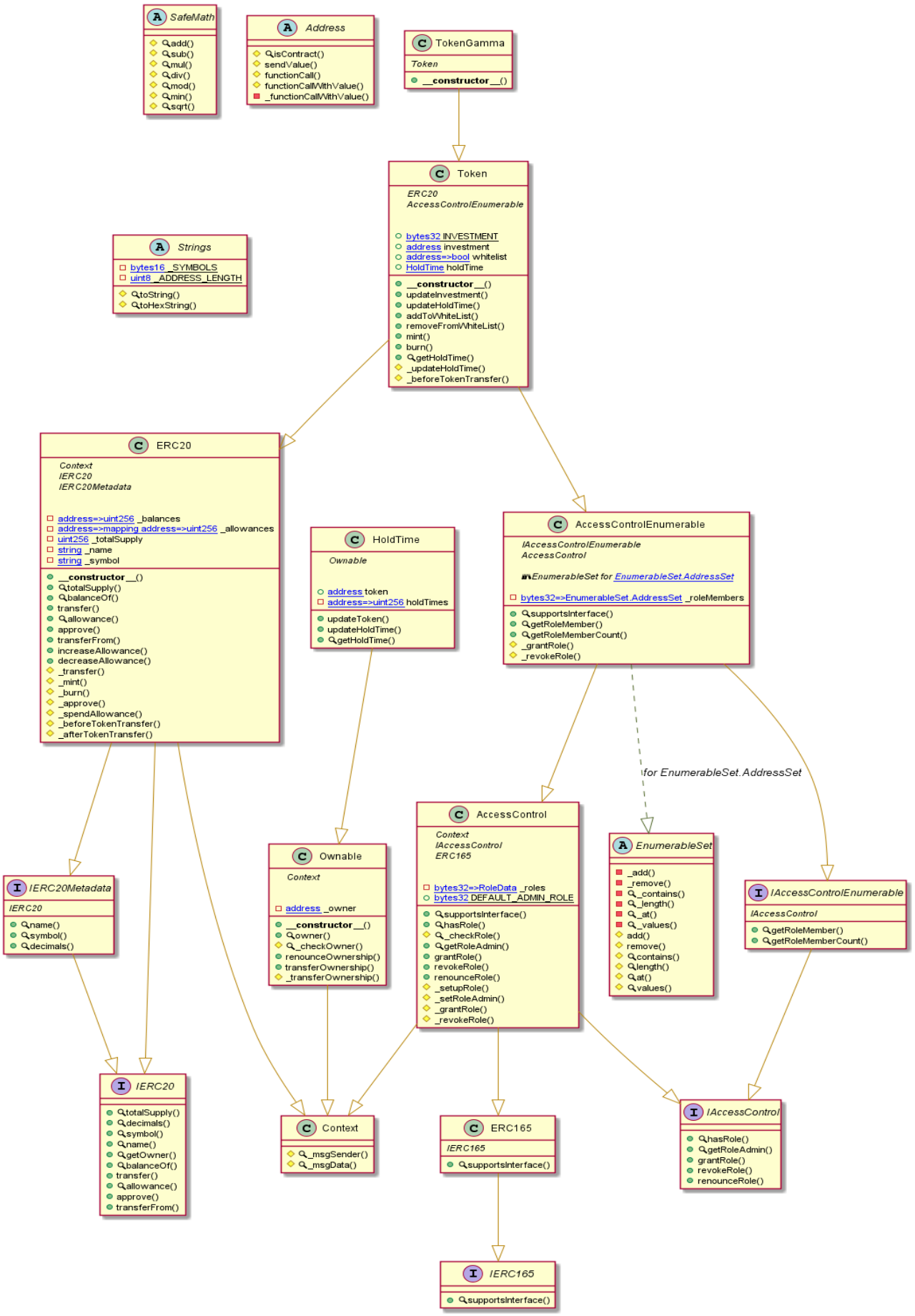
ManagementGamma Diagram



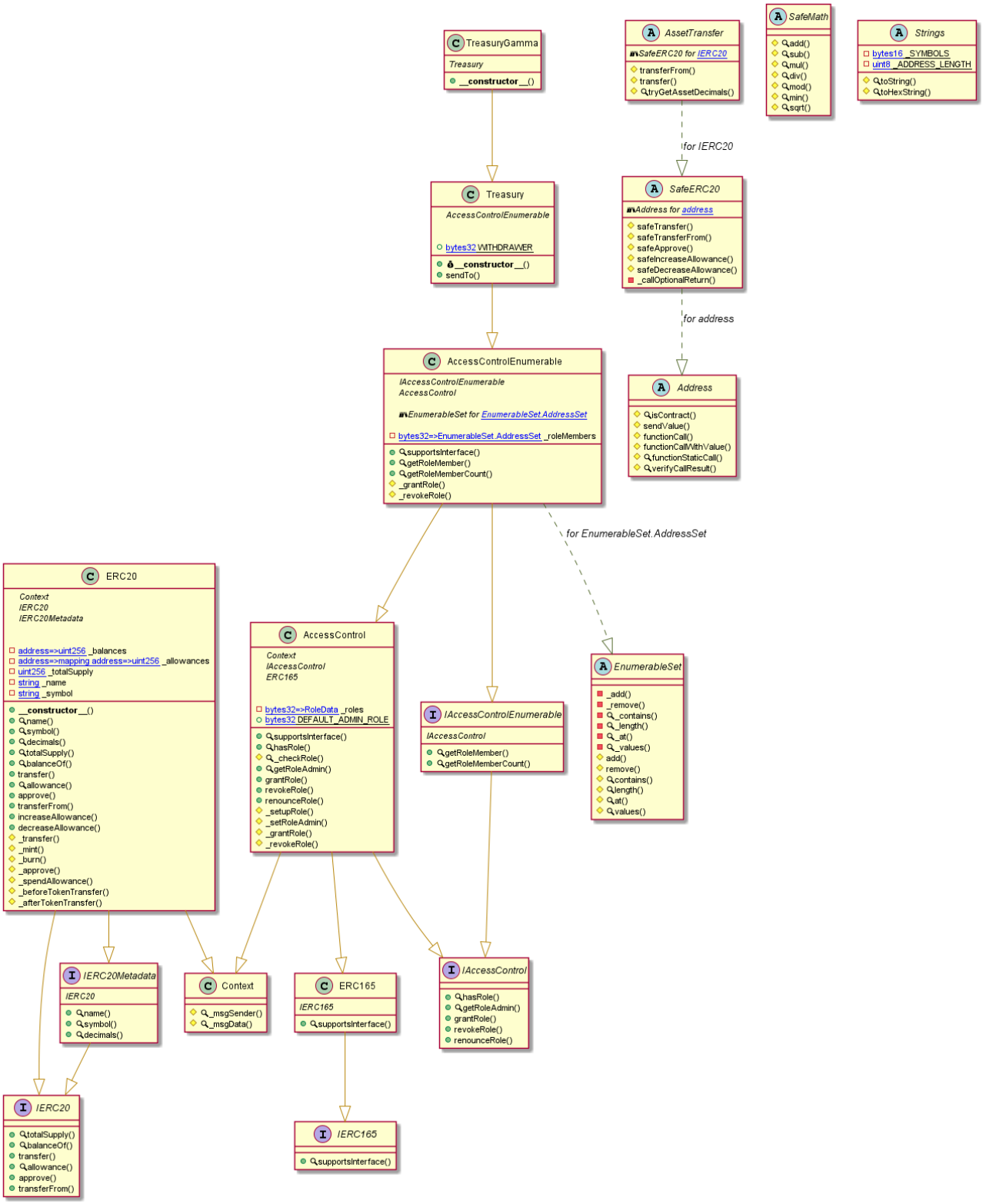
SafeHouseGamma Diagram



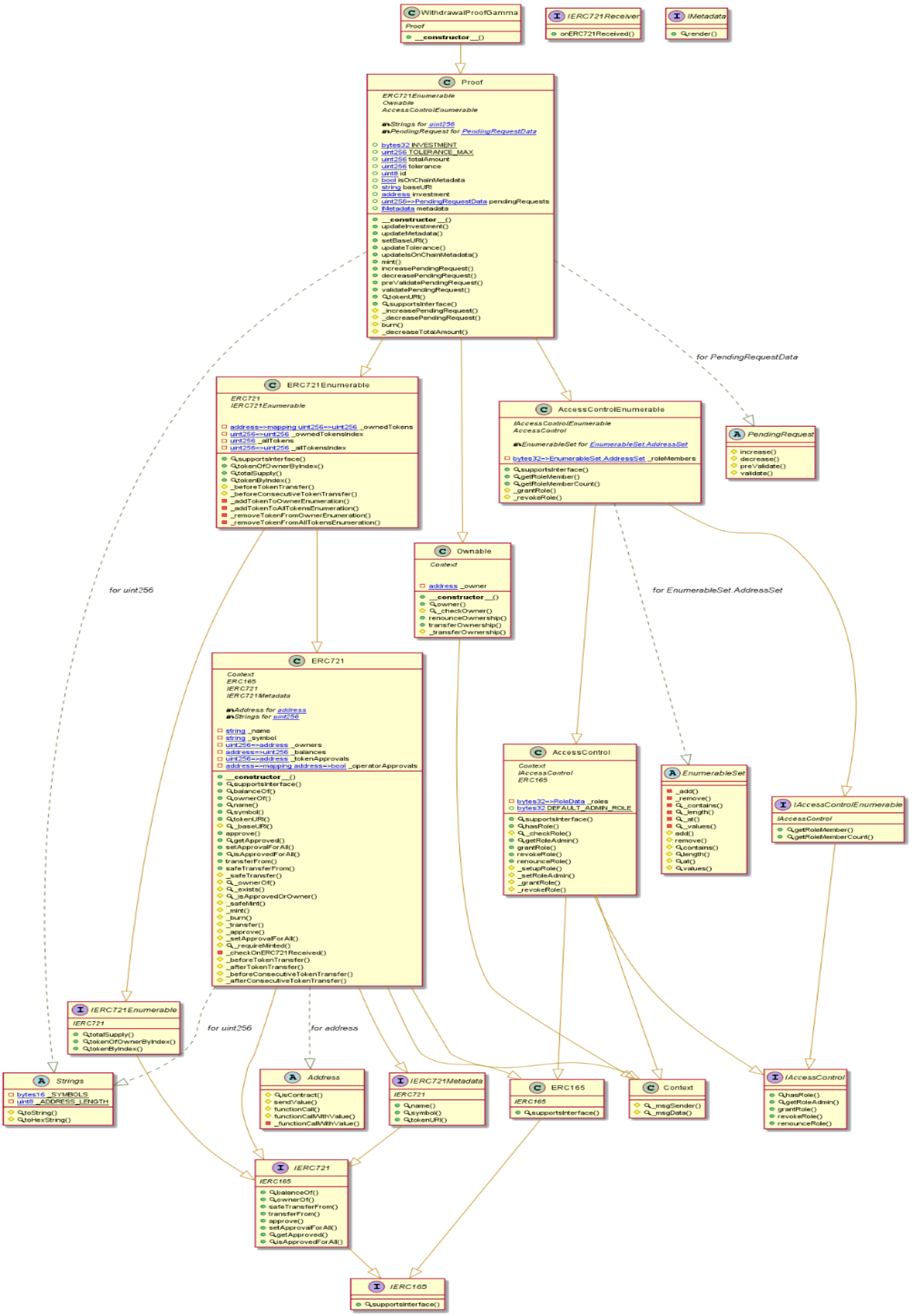
TokenGamma Diagram



TreasuryGamma Diagram



WithdrawalProofGamma Diagram



Slither Results Log

Slither log >> AssetBook.sol

```
Address.verifyCallResult(bool,bytes,string) (AssetBook.sol#90-108) uses assembly
        - INLINE ASM (AssetBook.sol#100-103)
EnumerableSet.values(EnumerableSet.Bytes32Set) (AssetBook.sol#559-568) uses assembly
        - INLINE ASM (AssetBook.sol#563-565)
EnumerableSet.values(EnumerableSet.AddressSet) (AssetBook.sol#595-604) uses assembly
        - INLINE ASM (AssetBook.sol#599-601)
EnumerableSet.values(EnumerableSet.UintSet) (AssetBook.sol#631-640) uses assembly
        - INLINE ASM (AssetBook.sol#635-637)
Strings.toString(uint256) (AssetBook.sol#646-660) uses assembly
        - INLINE ASM (AssetBook.sol#649-650)
        - INLINE ASM (AssetBook.sol#653-655)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.4 (AssetBook.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (AssetBook.sol#32-37):
        - (success) = recipient.call{value: amount}() (AssetBook.sol#35)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (AssetBook.sol#59-70):
        - (success,returndata) = target.call{value: value}(data) (AssetBook.sol#68)
Low level call in Address.functionStaticCall(address,bytes,string) (AssetBook.sol#76-85):
        - (success,returndata) = target.staticcall(data) (AssetBook.sol#83)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (AssetBook.sol#458-472):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (As
setBook.sol#461-464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (AssetBook.sol#260)" inContext (AssetBook.sol#255-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (AssetBook.sol#280) should be immutable
ERC20._symbol (AssetBook.sol#281) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
AssetBook.sol analyzed (17 contracts with 84 detectors), 66 result(s) found
```

Slither log >> HoldTime.sol

```
Context._msgData() (HoldTime.sol#39-42) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (HoldTime.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Redundant expression "this (HoldTime.sol#40)" inContext (HoldTime.sol#35-43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
HoldTime.sol analyzed (4 contracts with 84 detectors), 4 result(s) found
```

Slither log >> Investment.sol

```
Investment.isValidPrice(Proof,uint256) (Investment.sol#3051-3059) has external calls inside a loop: (minPrice_,maxPrice_) = pr
oof_.pendingRequests(tokenId_) (Investment.sol#3055-3057)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: depositProof.preVa
lidatePendingRequest(tokenId_,currentEventId) (Investment.sol#2704)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: (None,amountAsset_
,None,None,None) = depositProof.pendingRequests(tokenId_) (Investment.sol#2705)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: owner_ = depositPr
oof.ownerOf(tokenId_) (Investment.sol#2715)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: managementParity.s
etdepositData(amountToken_,amountAsset_,id) (Investment.sol#2717)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: token.mint(owner_,
amountToken_) (Investment.sol#2723)
Investment.validatedeposits(uint256[],uint256) (Investment.sol#2677-2748) has external calls inside a loop: depositProof.valid
atePendingRequest(tokenId_,amountAsset_,currentEventId) (Investment.sol#2726-2730)
Investment.validatewithdrawalProofs(uint256[],uint256) (Investment.sol#2750-2815) has external calls inside a loop: withdrawal
Proof.preValidatePendingRequest(tokenId_,currentEventId) (Investment.sol#2773)
Investment.validatewithdrawalProofs(uint256[],uint256) (Investment.sol#2750-2815) has external calls inside a loop: (None,amou
ntToken_,None,None,None) = withdrawalProof.pendingRequests(tokenId_) (Investment.sol#2774)
Investment.validatewithdrawalProofs(uint256[],uint256) (Investment.sol#2750-2815) has external calls inside a loop: owner_ = w
ithdrawalProof.ownerOf(tokenId_) (Investment.sol#2787)
Investment.validatewithdrawalProofs(uint256[],uint256) (Investment.sol#2750-2815) has external calls inside a loop: management
Parity.setwithdrawalData(amountAsset_,amountToken_,id) (Investment.sol#2789-2793)
Investment.validatewithdrawalProofs(uint256[],uint256) (Investment.sol#2750-2815) has external calls inside a loop: withdrawal
```

```
Pragma version^0.8.4 (Investment.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Investment.sol#19-24):
        - (success) = recipient.call{value: amount}() (Investment.sol#22)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (Investment.sol#56-78):
        - (success,returndata) = target.call{value: weiValue}(data) (Investment.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (Investment.sol#607-621) is not in mixedCase
Function FeeMinter.MintInvestmentFee(uint256,uint256,bool,address,address,address) (Investment.sol#2342-2376) is not in mixedC
ase
Event InvestmentwithdrawalRequest(address,uint256) (Investment.sol#2532) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Investment.sol#194)" inContext (Investment.sol#189-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Token (Investment.sol#1825-1926) does not implement functions:
        - IERC20.decimals() (Investment.sol#1342)
        - IERC20.getOwner() (Investment.sol#1348)
        - IERC20.name() (Investment.sol#1346)
        - IERC20.symbol() (Investment.sol#1344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
Investment.sol analyzed (35 contracts with 84 detectors), 127 result(s) found
```

Slither log >> Management.sol

```
Management.calculateWithdrawalFeeRate(uint256) (Management.sol#578-608) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= holdTime_,Transformative.Fi: max time) (Management.sol#582)
        - deltaTime_ <= withdrawalFee[0].time (Management.sol#589)
        - deltaTime_ > withdrawalFee[size_ - 1].time (Management.sol#591)
        - (deltaTime_ > time_) && (deltaTime_ <= fee_.time) (Management.sol#599)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
EnumerableSet.values(EnumerableSet.Bytes32Set) (Management.sol#87-96) uses assembly
        - INLINE ASM (Management.sol#91-93)
EnumerableSet.values(EnumerableSet.AddressSet) (Management.sol#123-132) uses assembly
        - INLINE ASM (Management.sol#127-129)
EnumerableSet.values(EnumerableSet.UintSet) (Management.sol#159-168) uses assembly
        - INLINE ASM (Management.sol#163-165)
Strings.toString(uint256) (Management.sol#174-188) uses assembly
        - INLINE ASM (Management.sol#177-178)
        - INLINE ASM (Management.sol#181-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setRoleAdmin(bytes32,bytes32) (Management.sol#318-322) is never used and should be removed
Context._msgData() (Management.sol#244-246) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (Management.sol#58-60) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (Management.sol#67-69) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (Management.sol#139-141) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (Management.sol#83-85) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (Management.sol#155-157) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (Management.sol#111-113) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (Management.sol#75-77) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (Management.sol#147-149) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (Management.sol#79-81) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (Management.sol#151-153) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (Management.sol#71-73) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (Management.sol#143-145) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (Management.sol#123-132) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (Management.sol#87-96) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (Management.sol#159-168) is never used and should be removed
Strings.toHexString(uint256) (Management.sol#190-193) is never used and should be removed
Strings.toString(uint256) (Management.sol#174-188) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (Management.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
Management.sol analyzed (10 contracts with 84 detectors), 28 result(s) found
```

Slither log >> Proof.sol

```
Reentrancy in Proof.mint(address,uint256,uint256,uint256,uint256,uint256) (Proof.sol#1112-1129):
        External calls:
        - _safeMint(account_,tokenId_) (Proof.sol#1120)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (Proof.sol#438-448)
        State variables written after the call(s):
        - _increasePendingRequest(tokenId_,amount_,minPrice_,maxPrice_,currentEventId_) (Proof.sol#1121-1127)
                - totalAmount += amount_ (Proof.sol#1214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
Pragma version^0.8.4 (Proof.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Proof.sol#17-22):
        - (success) = recipient.call{value: amount}() (Proof.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (Proof.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (Proof.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (Proof.sol#600-614) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Proof.sol#187)" inContext (Proof.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
Proof.sol analyzed (20 contracts with 84 detectors), 48 result(s) found
```

Slither log >> SafeHouse.sol

```
Reentrancy in Proof.mint(address,uint256,uint256,uint256,uint256,uint256) (Proof.sol#1112-1129):
        External calls:
        - _safeMint(account_,tokenId_) (Proof.sol#1120)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (Proof.sol#438-448)
        State variables written after the call(s):
        - _increasePendingRequest(tokenId_,amount_,minPrice_,maxPrice_,currentEventId_) (Proof.sol#1121-1127)
                - totalAmount += amount_ (Proof.sol#1214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
Pragma version^0.8.4 (SafeHouse.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SafeHouse.sol#33-38):
        - (success) = recipient.call{value: amount}() (SafeHouse.sol#36)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (SafeHouse.sol#60-71):
        - (success,returndata) = target.call{value: value}(data) (SafeHouse.sol#69)
Low level call in Address.functionStaticCall(address,bytes,string) (SafeHouse.sol#77-86):
        - (success,returndata) = target.staticcall(data) (SafeHouse.sol#84)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (SafeHouse.sol#459-473):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Sa
feHouse.sol#462-465)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (SafeHouse.sol#261)" inContext (SafeHouse.sol#256-264)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (SafeHouse.sol#281) should be immutable
ERC20._symbol (SafeHouse.sol#282) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SafeHouse.sol analyzed (21 contracts with 84 detectors), 83 result(s) found
```

Slither log >> Token.sol

```
Token.constructor(string,string,address,address)._name (Token.sol#777) shadows:
        - ERC20._name (Token.sol#208) (state variable)
Token.constructor(string,string,address,address)._symbol (Token.sol#778) shadows:
        - ERC20._symbol (Token.sol#209) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in Token.burn(address,uint256) (Token.sol#836-843):
        External calls:
        - _burn(from_,amount_) (Token.sol#841)
                - holdTime.updateHoldTime(account_,amount_) (Token.sol#852)
        Event emitted after the call(s):
        - Burn(from_,amount_) (Token.sol#842)
Reentrancy in Token.mint(address,uint256) (Token.sol#831-834):
        External calls:
        - _mint(to_,amount_) (Token.sol#832)
                - holdTime.updateHoldTime(account_,amount_) (Token.sol#852)
        Event emitted after the call(s):
        - Mint(to_,amount_) (Token.sol#833)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Pragma version^0.8.4 (Token.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Token.sol#96-101):
        - (success) = recipient.call{value: amount}() (Token.sol#99)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (Token.sol#133-155):
        - (success,returndata) = target.call{value: weiValue}(data) (Token.sol#141)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (Token.sol#192)" inContext (Token.sol#187-195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Token (Token.sol#763-864) does not implement functions:
        - IERC20.decimals() (Token.sol#160)
        - IERC20.getOwner() (Token.sol#166)
        - IERC20.name() (Token.sol#164)
        - IERC20.symbol() (Token.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
Token.sol analyzed (17 contracts with 84 detectors), 56 result(s) found
```

Slither log >> Treasury.sol

```
Pragma version^0.8.4 (Treasury.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Treasury.sol#32-37):
        - (success) = recipient.call{value: amount}() (Treasury.sol#35)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Treasury.sol#59-70):
        - (success,returndata) = target.call{value: value}(data) (Treasury.sol#68)
Low level call in Address.functionStaticCall(address,bytes,string) (Treasury.sol#76-85):
        - (success,returndata) = target.staticcall(data) (Treasury.sol#83)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (Treasury.sol#458-472):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Tr
easury.sol#461-464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (Treasury.sol#260)" inContext (Treasury.sol#255-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (Treasury.sol#280) should be immutable
ERC20._symbol (Treasury.sol#281) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
Treasury.sol analyzed (17 contracts with 84 detectors), 58 result(s) found
```

Slither log >> AssetBookAlpha.sol

```
Pragma version^0.8.4 (AssetBookAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (AssetBookAlpha.sol#32-37):
        - (success) = recipient.call{value: amount}() (AssetBookAlpha.sol#35)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (AssetBookAlpha.sol#59-70):
        - (success,returndata) = target.call{value: value}(data) (AssetBookAlpha.sol#68)
Low level call in Address.functionStaticCall(address,bytes,string) (AssetBookAlpha.sol#76-85):
        - (success,returndata) = target.staticcall(data) (AssetBookAlpha.sol#83)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (AssetBookAlpha.sol#458-472):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (As
setBookAlpha.sol#461-464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (AssetBookAlpha.sol#260)" inContext (AssetBookAlpha.sol#255-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (AssetBookAlpha.sol#280) should be immutable
ERC20._symbol (AssetBookAlpha.sol#281) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
AssetBookAlpha.sol analyzed (18 contracts with 84 detectors), 66 result(s) found
```

Slither log >> DepositProofAlpha.sol

```
Pragma version^0.8.4 (DepositProofAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DepositProofAlpha.sol#17-22):
        - (success) = recipient.call{value: amount}() (DepositProofAlpha.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (DepositProofAlpha.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (DepositProofAlpha.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (DepositProofAlpha.sol#600-614) is not in mixed
Case
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DepositProofAlpha.sol#187)" inContext (DepositProofAlpha.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
DepositProofAlpha.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Slither log >> HoldTimeAlpha.sol

```
Context._msgData() (HoldTimeAlpha.sol#39-42) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (HoldTimeAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Redundant expression "this (HoldTimeAlpha.sol#40)" inContext (HoldTimeAlpha.sol#35-43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
HoldTimeAlpha.sol analyzed (5 contracts with 84 detectors), 4 result(s) found
```

Slither log >> InvestmentAlpha.sol

```
Pragma version^0.8.4 (InvestmentAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (InvestmentAlpha.sol#19-24):
        - (success) = recipient.call{value: amount}() (InvestmentAlpha.sol#22)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (InvestmentAlpha.sol#56-78):
        - (success,returndata) = target.call{value: weiValue}(data) (InvestmentAlpha.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (InvestmentAlpha.sol#607-621) is not in mixedCa
se
Function FeeMinter.MintInvestmentFee(uint256,uint256,bool,address,address,address) (InvestmentAlpha.sol#2342-2376) is not in m
ixedCase
Event InvestmentwithdrawalRequest(address,uint256) (InvestmentAlpha.sol#2532) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (InvestmentAlpha.sol#194)" inContext (InvestmentAlpha.sol#189-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Token (InvestmentAlpha.sol#1825-1926) does not implement functions:
        - IERC20.decimals() (InvestmentAlpha.sol#1342)
        - IERC20.getOwner() (InvestmentAlpha.sol#1348)
        - IERC20.name() (InvestmentAlpha.sol#1346)
        - IERC20.symbol() (InvestmentAlpha.sol#1344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
InvestmentAlpha.sol analyzed (36 contracts with 84 detectors), 127 result(s) found
```

Slither log >> ManagementAlpha.sol

```
Management.calculateWithdrawalFeeRate(uint256) (ManagementAlpha.sol#578-608) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= holdTime_,Transformative.Fi: max time) (ManagementAlpha.sol#582)
        - deltaTime_ <= withdrawalFee[0].time (ManagementAlpha.sol#589)
        - deltaTime_ > withdrawalFee[size_ - 1].time (ManagementAlpha.sol#591)
        - (deltaTime_ > time_) && (deltaTime_ <= fee_.time) (ManagementAlpha.sol#599)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
AccessControl._setRoleAdmin(bytes32,bytes32) (ManagementAlpha.sol#318-322) is never used and should be removed
Context._msgData() (ManagementAlpha.sol#244-246) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (ManagementAlpha.sol#58-60) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (ManagementAlpha.sol#67-69) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (ManagementAlpha.sol#139-141) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (ManagementAlpha.sol#83-85) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (ManagementAlpha.sol#155-157) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (ManagementAlpha.sol#111-113) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (ManagementAlpha.sol#75-77) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (ManagementAlpha.sol#147-149) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (ManagementAlpha.sol#79-81) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (ManagementAlpha.sol#151-153) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (ManagementAlpha.sol#71-73) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (ManagementAlpha.sol#143-145) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (ManagementAlpha.sol#123-132) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (ManagementAlpha.sol#87-96) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (ManagementAlpha.sol#159-168) is never used and should be removed
Strings.toHexString(uint256) (ManagementAlpha.sol#190-193) is never used and should be removed
Strings.toString(uint256) (ManagementAlpha.sol#174-188) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (ManagementAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
ManagementAlpha.sol analyzed (11 contracts with 84 detectors), 28 result(s) found
```

Slither log >> SafeHouseAlpha.sol

```
Reentrancy in SafeHouse.depositAsset(address,uint256) (SafeHouseAlpha.sol#1270-1294):
        External calls:
        - AssetTransfer.transferFrom(msg.sender,address(this),amount_,IERC20(asset_)) (SafeHouseAlpha.sol#1286-1291)
        Event emitted after the call(s):
        - DepositAsset(msg.sender,asset_,amount_) (SafeHouseAlpha.sol#1293)
Reentrancy in SafeHouse.sendToVault(address,address,uint256) (SafeHouseAlpha.sol#1321-1331):
        External calls:
        - AssetTransfer.transfer(vault_,amount_,asset_) (SafeHouseAlpha.sol#1329)
        Event emitted after the call(s):
        - SendToVault(asset_,vault_,amount_) (SafeHouseAlpha.sol#1330)
Reentrancy in SafeHouse.withdrawAsset(address,uint256) (SafeHouseAlpha.sol#1296-1319):
        External calls:
        - AssetTransfer.transfer(msg.sender,amount_,asset_) (SafeHouseAlpha.sol#1317)
        Event emitted after the call(s):
        - WithdrawAsset(msg.sender,asset_,amount_) (SafeHouseAlpha.sol#1318)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Pragma version^0.8.4 (SafeHouseAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SafeHouseAlpha.sol#33-38):
        - (success) = recipient.call{value: amount}() (SafeHouseAlpha.sol#36)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (SafeHouseAlpha.sol#60-71):
        - (success,returndata) = target.call{value: value}(data) (SafeHouseAlpha.sol#69)
Low level call in Address.functionStaticCall(address,bytes,string) (SafeHouseAlpha.sol#77-86):
        - (success,returndata) = target.staticcall(data) (SafeHouseAlpha.sol#84)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (SafeHouseAlpha.sol#459-473):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Sa
feHouseAlpha.sol#462-465)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Redundant expression "this (SafeHouseAlpha.sol#261)" inContext (SafeHouseAlpha.sol#256-264)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (SafeHouseAlpha.sol#281) should be immutable
ERC20._symbol (SafeHouseAlpha.sol#282) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SafeHouseAlpha.sol analyzed (22 contracts with 84 detectors), 83 result(s) found
```

Slither log >> TokenAlpha.sol

```
Token.constructor(string,string,address,address)._name (TokenAlpha.sol#777) shadows:
        - ERC20._name (TokenAlpha.sol#208) (state variable)
Token.constructor(string,string,address,address)._symbol (TokenAlpha.sol#778) shadows:
        - ERC20._symbol (TokenAlpha.sol#209) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in Token.burn(address,uint256) (TokenAlpha.sol#836-843):
        External calls:
        - _burn(from_,amount_) (TokenAlpha.sol#841)
                - holdTime.updateHoldTime(account_,amount_) (TokenAlpha.sol#852)
        Event emitted after the call(s):
        - Burn(from_,amount_) (TokenAlpha.sol#842)
Reentrancy in Token.mint(address,uint256) (TokenAlpha.sol#831-834):
        External calls:
        - _mint(to_,amount_) (TokenAlpha.sol#832)
                - holdTime.updateHoldTime(account_,amount_) (TokenAlpha.sol#852)
        Event emitted after the call(s):
        - Mint(to_,amount_) (TokenAlpha.sol#833)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Pragma version^0.8.4 (TokenAlpha.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TokenAlpha.sol#96-101):
        - (success) = recipient.call{value: amount}() (TokenAlpha.sol#99)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (TokenAlpha.sol#133-155):
        - (success,returndata) = target.call{value: weiValue}(data) (TokenAlpha.sol#141)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TokenAlpha.sol#192)" inContext (TokenAlpha.sol#187-195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

TokenAlpha (TokenAlpha.sol#867-872) does not implement functions:
        - IERC20.decimals() (TokenAlpha.sol#160)
        - IERC20.getOwner() (TokenAlpha.sol#166)
        - IERC20.name() (TokenAlpha.sol#164)
        - IERC20.symbol() (TokenAlpha.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
TokenAlpha.sol analyzed (18 contracts with 84 detectors), 56 result(s) found
```

Slither log >> TreasuryAlpha.sol

```
Reentrancy in Treasury.sendTo(address,uint256,address) (TreasuryAlpha.sol#1593-1600):
        External calls:
        - AssetTransfer.transfer(to_,amount_,asset_) (TreasuryAlpha.sol#1598)
        Event emitted after the call(s):
        - SendTo(to_,amount_,asset_) (TreasuryAlpha.sol#1599)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Pragma version^0.8.4 (TreasuryAlpha.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TreasuryAlpha.sol#131-136):
        - (success) = recipient.call{value: amount}() (TreasuryAlpha.sol#134)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (TreasuryAlpha.sol#199-210):
        - (success,returndata) = target.call{value: value}(data) (TreasuryAlpha.sol#208)
Low level call in Address.functionStaticCall(address,bytes,string) (TreasuryAlpha.sol#228-237):
        - (success,returndata) = target.staticcall(data) (TreasuryAlpha.sol#235)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (TreasuryAlpha.sol#876-890):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Tr
easuryAlpha.sol#879-882)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TreasuryAlpha.sol#465)" inContext (TreasuryAlpha.sol#459-468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (TreasuryAlpha.sol#494) should be immutable
ERC20._symbol (TreasuryAlpha.sol#495) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
TreasuryAlpha.sol analyzed (18 contracts with 84 detectors), 58 result(s) found
```

Slither log >> WithdrawalProofAlpha.sol

```
Pragma version^0.8.4 (WithdrawalProofAlpha.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (WithdrawalProofAlpha.sol#17-22):
        - (success) = recipient.call{value: amount}() (WithdrawalProofAlpha.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (WithdrawalProofAlpha.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (WithdrawalProofAlpha.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (WithdrawalProofAlpha.sol#600-614) is not in mi
xedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (WithdrawalProofAlpha.sol#187)" inContext (WithdrawalProofAlpha.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
WithdrawalProofAlpha.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Slither log >> AssetBookBeta.sol

```
Pragma version^0.8.4 (AssetBookBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (AssetBookBeta.sol#32-37):
        - (success) = recipient.call{value: amount}() (AssetBookBeta.sol#35)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (AssetBookBeta.sol#59-70):
        - (success,returndata) = target.call{value: value}(data) (AssetBookBeta.sol#68)
Low level call in Address.functionStaticCall(address,bytes,string) (AssetBookBeta.sol#76-85):
        - (success,returndata) = target.staticcall(data) (AssetBookBeta.sol#83)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (AssetBookBeta.sol#458-472):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (As
setBookBeta.sol#461-464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (AssetBookBeta.sol#260)" inContext (AssetBookBeta.sol#255-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (AssetBookBeta.sol#280) should be immutable
ERC20._symbol (AssetBookBeta.sol#281) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
AssetBookBeta.sol analyzed (18 contracts with 84 detectors), 66 result(s) found
```

Slither log >> DepositProofBeta.sol

```
Pragma version^0.8.4 (DepositProofBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DepositProofBeta.sol#17-22):
        - (success) = recipient.call{value: amount}() (DepositProofBeta.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (DepositProofBeta.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (DepositProofBeta.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (DepositProofBeta.sol#600-614) is not in mixedC
ase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DepositProofBeta.sol#187)" inContext (DepositProofBeta.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
DepositProofBeta.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Slither log >> HoldTimeBeta.sol

```
Context._msgData() (HoldTimeBeta.sol#39-42) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (HoldTimeBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Redundant expression "this (HoldTimeBeta.sol#40)" inContext (HoldTimeBeta.sol#35-43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
HoldTimeBeta.sol analyzed (5 contracts with 84 detectors), 4 result(s) found
```

Slither log >> InvestmentBeta.sol

```
Pragma version^0.8.4 (InvestmentBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (InvestmentBeta.sol#19-24):
        - (success) = recipient.call{value: amount}() (InvestmentBeta.sol#22)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (InvestmentBeta.sol#56-78):
        - (success,returndata) = target.call{value: weiValue}(data) (InvestmentBeta.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (InvestmentBeta.sol#607-621) is not in mixedCas
e
Function FeeMinter.MintInvestmentFee(uint256,uint256,bool,address,address,address) (InvestmentBeta.sol#2342-2376) is not in mi
xedCase
Event InvestmentwithdrawalRequest(address,uint256) (InvestmentBeta.sol#2532) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (InvestmentBeta.sol#194)" inContext (InvestmentBeta.sol#189-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Token (InvestmentBeta.sol#1825-1926) does not implement functions:
        - IERC20.decimals() (InvestmentBeta.sol#1342)
        - IERC20.getOwner() (InvestmentBeta.sol#1348)
        - IERC20.name() (InvestmentBeta.sol#1346)
        - IERC20.symbol() (InvestmentBeta.sol#1344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
InvestmentBeta.sol analyzed (36 contracts with 84 detectors), 127 result(s) found
```

Slither log >> ManagementBeta.sol

```
Management.calculateWithdrawalFeeRate(uint256) (ManagementBeta.sol#578-608) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= holdTime_,Transformative.Fi: max time) (ManagementBeta.sol#582)
        - deltaTime_ <= withdrawalFee[0].time (ManagementBeta.sol#589)
        - deltaTime_ > withdrawalFee[size_ - 1].time (ManagementBeta.sol#591)
        - (deltaTime_ > time_) && (deltaTime_ <= fee_.time) (ManagementBeta.sol#599)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
AccessControl._setRoleAdmin(bytes32,bytes32) (ManagementBeta.sol#318-322) is never used and should be removed
Context._msgData() (ManagementBeta.sol#244-246) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (ManagementBeta.sol#58-60) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (ManagementBeta.sol#67-69) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (ManagementBeta.sol#139-141) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (ManagementBeta.sol#83-85) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (ManagementBeta.sol#155-157) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (ManagementBeta.sol#111-113) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (ManagementBeta.sol#75-77) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (ManagementBeta.sol#147-149) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (ManagementBeta.sol#79-81) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (ManagementBeta.sol#151-153) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (ManagementBeta.sol#71-73) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (ManagementBeta.sol#143-145) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (ManagementBeta.sol#123-132) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (ManagementBeta.sol#87-96) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (ManagementBeta.sol#159-168) is never used and should be removed
Strings.toHexString(uint256) (ManagementBeta.sol#190-193) is never used and should be removed
Strings.toString(uint256) (ManagementBeta.sol#174-188) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (ManagementBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
ManagementBeta.sol analyzed (11 contracts with 84 detectors), 28 result(s) found
```

Slither log >> SafeHouseBeta.sol

```
Pragma version^0.8.4 (SafeHouseBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SafeHouseBeta.sol#33-38):
        - (success) = recipient.call{value: amount}() (SafeHouseBeta.sol#36)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (SafeHouseBeta.sol#60-71):
        - (success,returndata) = target.call{value: value}(data) (SafeHouseBeta.sol#69)
Low level call in Address.functionStaticCall(address,bytes,string) (SafeHouseBeta.sol#77-86):
        - (success,returndata) = target.staticcall(data) (SafeHouseBeta.sol#84)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (SafeHouseBeta.sol#459-473):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Sa
feHouseBeta.sol#462-465)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (SafeHouseBeta.sol#261)" inContext (SafeHouseBeta.sol#256-264)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (SafeHouseBeta.sol#281) should be immutable
ERC20._symbol (SafeHouseBeta.sol#282) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SafeHouseBeta.sol analyzed (22 contracts with 84 detectors), 83 result(s) found
```

Slither log >> TokenBeta.sol

```
Token.constructor(string,string,address,address)._name (TokenBeta.sol#777) shadows:
        - ERC20._name (TokenBeta.sol#208) (state variable)
Token.constructor(string,string,address,address)._symbol (TokenBeta.sol#778) shadows:
        - ERC20._symbol (TokenBeta.sol#209) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in Token.burn(address,uint256) (TokenBeta.sol#836-843):
        External calls:
        - _burn(from_,amount_) (TokenBeta.sol#841)
                - holdTime.updateHoldTime(account_,amount_) (TokenBeta.sol#852)
        Event emitted after the call(s):
        - Burn(from_,amount_) (TokenBeta.sol#842)
Reentrancy in Token.mint(address,uint256) (TokenBeta.sol#831-834):
        External calls:
        - _mint(to_,amount_) (TokenBeta.sol#832)
                - holdTime.updateHoldTime(account_,amount_) (TokenBeta.sol#852)
        Event emitted after the call(s):
        - Mint(to_,amount_) (TokenBeta.sol#833)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Pragma version^0.8.4 (TokenBeta.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TokenBeta.sol#96-101):
        - (success) = recipient.call{value: amount}() (TokenBeta.sol#99)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (TokenBeta.sol#133-155):
        - (success,returndata) = target.call{value: weiValue}(data) (TokenBeta.sol#141)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TokenBeta.sol#192)" inContext (TokenBeta.sol#187-195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```

```
Redundant expression "this (TokenBeta.sol#192)" inContext (TokenBeta.sol#187-195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

TokenBeta (TokenBeta.sol#867-872) does not implement functions:
        - IERC20.decimals() (TokenBeta.sol#160)
        - IERC20.getOwner() (TokenBeta.sol#166)
        - IERC20.name() (TokenBeta.sol#164)
        - IERC20.symbol() (TokenBeta.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
TokenBeta.sol analyzed (18 contracts with 84 detectors), 56 result(s) found
```

Slither log >> TreasuryBeta.sol

```
Pragma version^0.8.4 (TreasuryBeta.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TreasuryBeta.sol#131-136):
        - (success) = recipient.call{value: amount}() (TreasuryBeta.sol#134)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (TreasuryBeta.sol#199-210):
        - (success,returndata) = target.call{value: value}(data) (TreasuryBeta.sol#208)
Low level call in Address.functionStaticCall(address,bytes,string) (TreasuryBeta.sol#228-237):
        - (success,returndata) = target.staticcall(data) (TreasuryBeta.sol#235)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (TreasuryBeta.sol#876-890):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Tr
easuryBeta.sol#879-882)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TreasuryBeta.sol#465)" inContext (TreasuryBeta.sol#459-468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (TreasuryBeta.sol#494) should be immutable
ERC20._symbol (TreasuryBeta.sol#495) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
TreasuryBeta.sol analyzed (18 contracts with 84 detectors), 58 result(s) found
```

Slither log >> WithdrawalProofBeta.sol

```
Pragma version^0.8.4 (WithdrawalProofBeta.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (WithdrawalProofBeta.sol#17-22):
        - (success) = recipient.call{value: amount}() (WithdrawalProofBeta.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (WithdrawalProofBeta.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (WithdrawalProofBeta.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (WithdrawalProofBeta.sol#600-614) is not in mix
edCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (WithdrawalProofBeta.sol#187)" inContext (WithdrawalProofBeta.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
WithdrawalProofBeta.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Slither log >> AssetBookGamma.sol

```
Pragma version^0.8.4 (AssetBookGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (AssetBookGamma.sol#32-37):
        - (success) = recipient.call{value: amount}() (AssetBookGamma.sol#35)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (AssetBookGamma.sol#59-70):
        - (success,returndata) = target.call{value: value}(data) (AssetBookGamma.sol#68)
Low level call in Address.functionStaticCall(address,bytes,string) (AssetBookGamma.sol#76-85):
        - (success,returndata) = target.staticcall(data) (AssetBookGamma.sol#83)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (AssetBookGamma.sol#458-472):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (As
setBookGamma.sol#461-464)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (AssetBookGamma.sol#260)" inContext (AssetBookGamma.sol#255-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (AssetBookGamma.sol#280) should be immutable
ERC20._symbol (AssetBookGamma.sol#281) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
AssetBookGamma.sol analyzed (18 contracts with 84 detectors), 66 result(s) found
```

Slither log >> DepositProofGamma.sol

```
Pragma version^0.8.4 (DepositProofGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DepositProofGamma.sol#17-22):
        - (success) = recipient.call{value: amount}() (DepositProofGamma.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (DepositProofGamma.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (DepositProofGamma.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (DepositProofGamma.sol#600-614) is not in mixed
Case
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DepositProofGamma.sol#187)" inContext (DepositProofGamma.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
sDepositProofGamma.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Slither log >> HoldTimeGamma.sol

```
Context._msgData() (HoldTimeGamma.sol#39-42) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.4 (HoldTimeGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Redundant expression "this (HoldTimeGamma.sol#40)" inContext (HoldTimeGamma.sol#35-43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
HoldTimeGamma.sol analyzed (5 contracts with 84 detectors), 4 result(s) found
```

Slither log >> InvestmentGamma.sol

```
Pragma version^0.8.4 (InvestmentGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (InvestmentGamma.sol#19-24):
        - (success) = recipient.call{value: amount}() (InvestmentGamma.sol#22)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (InvestmentGamma.sol#56-78):
        - (success,returndata) = target.call{value: weiValue}(data) (InvestmentGamma.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (InvestmentGamma.sol#607-621) is not in mixedCa
se
Function FeeMinter.MintInvestmentFee(uint256,uint256,bool,address,address,address) (InvestmentGamma.sol#2342-2376) is not in m
ixedCase
Event InvestmentwithdrawalRequest(address,uint256) (InvestmentGamma.sol#2532) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (InvestmentGamma.sol#194)" inContext (InvestmentGamma.sol#189-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Token (InvestmentGamma.sol#1825-1926) does not implement functions:
        - IERC20.decimals() (InvestmentGamma.sol#1342)
        - IERC20.getOwner() (InvestmentGamma.sol#1348)
        - IERC20.name() (InvestmentGamma.sol#1346)
        - IERC20.symbol() (InvestmentGamma.sol#1344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
InvestmentGamma.sol analyzed (36 contracts with 84 detectors), 127 result(s) found
```

Slither log >> ManagementGamma.sol

```
Management.calculateWithdrawalFeeRate(uint256) (ManagementGamma.sol#578-608) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= holdTime_,Transformative.Fi: max time) (ManagementGamma.sol#582)
        - deltaTime_ <= withdrawalFee[0].time (ManagementGamma.sol#589)
        - deltaTime_ > withdrawalFee[size_ - 1].time (ManagementGamma.sol#591)
        - (deltaTime_ > time_) && (deltaTime_ <= fee_.time) (ManagementGamma.sol#599)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
Pragma version^0.8.4 (ManagementGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
ManagementGamma.sol analyzed (11 contracts with 84 detectors), 28 result(s) found
```

Slither log >> SafeHouseGamma.sol

```
Pragma version^0.8.4 (SafeHouseGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SafeHouseGamma.sol#33-38):
        - (success) = recipient.call{value: amount}() (SafeHouseGamma.sol#36)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (SafeHouseGamma.sol#60-71):
        - (success,returndata) = target.call{value: value}(data) (SafeHouseGamma.sol#69)
Low level call in Address.functionStaticCall(address,bytes,string) (SafeHouseGamma.sol#77-86):
        - (success,returndata) = target.staticcall(data) (SafeHouseGamma.sol#84)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (SafeHouseGamma.sol#459-473):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Sa
feHouseGamma.sol#462-465)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (SafeHouseGamma.sol#261)" inContext (SafeHouseGamma.sol#256-264)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (SafeHouseGamma.sol#281) should be immutable
ERC20._symbol (SafeHouseGamma.sol#282) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SafeHouseGamma.sol analyzed (22 contracts with 84 detectors), 83 result(s) found
```

Slither log >> TokenGamma.sol

```
Pragma version^0.8.4 (TokenGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TokenGamma.sol#95-100):
        - (success) = recipient.call{value: amount}() (TokenGamma.sol#98)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (TokenGamma.sol#132-154):
        - (success,returndata) = target.call{value: weiValue}(data) (TokenGamma.sol#140)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TokenGamma.sol#191)" inContext (TokenGamma.sol#186-194)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

StableToken (TokenGamma.sol#367-377) does not implement functions:
        - IERC20.balanceOf(address) (TokenGamma.sol#167)
        - IERC20.decimals() (TokenGamma.sol#159)
        - IERC20.getOwner() (TokenGamma.sol#165)
        - IERC20.name() (TokenGamma.sol#163)
        - IERC20.symbol() (TokenGamma.sol#161)
        - IERC20.totalSupply() (TokenGamma.sol#157)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

StableToken.decimal (TokenGamma.sol#368) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
TokenGamma.sol analyzed (9 contracts with 84 detectors), 28 result(s) found
```

Slither log >> TreasuryGamma.sol

```
Pragma version^0.8.4 (TreasuryGamma.sol#3) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (TreasuryGamma.sol#131-136):
        - (success) = recipient.call{value: amount}() (TreasuryGamma.sol#134)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (TreasuryGamma.sol#199-210):
        - (success,returndata) = target.call{value: value}(data) (TreasuryGamma.sol#208)
Low level call in Address.functionStaticCall(address,bytes,string) (TreasuryGamma.sol#228-237):
        - (success,returndata) = target.staticcall(data) (TreasuryGamma.sol#235)
Low level call in AssetTransfer.tryGetAssetDecimals(IERC20) (TreasuryGamma.sol#876-890):
        - (success,encodedDecimals) = address(asset_).staticcall(abi.encodeWithSelector(IERC20Metadata.decimals.selector)) (Tr
easuryGamma.sol#879-882)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (TreasuryGamma.sol#465)" inContext (TreasuryGamma.sol#459-468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (TreasuryGamma.sol#494) should be immutable
ERC20._symbol (TreasuryGamma.sol#495) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
TreasuryGamma.sol analyzed (18 contracts with 84 detectors), 58 result(s) found
```

Slither log >> WithdrawalProofGamma.sol

```
Pragma version^0.8.4 (WithdrawalProofGamma.sol#2) allows old versions
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (WithdrawalProofGamma.sol#17-22):
        - (success) = recipient.call{value: amount}() (WithdrawalProofGamma.sol#20)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (WithdrawalProofGamma.sol#54-76):
        - (success,returndata) = target.call{value: weiValue}(data) (WithdrawalProofGamma.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Modifier PendingRequest.CheckPrice(PendingRequestData,uint256,uint256,uint256) (WithdrawalProofGamma.sol#600-614) is not in mi
xedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (WithdrawalProofGamma.sol#187)" inContext (WithdrawalProofGamma.sol#182-190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
WithdrawalProofGamma.sol analyzed (21 contracts with 84 detectors), 48 result(s) found
```

Solidity Static Analysis

AssetBook.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Address._functionCallWithValue(address,bytes,uint256,string): Could potentially lead to re-entrancy vulnerability.

more

Pos: 133:4:

ERC

ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type

more

Pos: 160:4:

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transfer(address,uint256,address) : Potentially should be constant/view/pure but is not.

more

Pos: 46:4:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

Pos: 109:12:

No return:

AssetTransfer.tryGetAssetDecimals(contract IERC20): Defines a return type but never explicitly returns a value.

Pos: 54:4:

Email: audit@EtherAuthority.io

HoldTime.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HoldTime.updateHoldTime(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 42:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 48:12:

Miscellaneous

Constant/View/Pure functions:

IERC20.transfer(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 41:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 43:8:

Investment.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 291:25:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 135:8:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

Pos: 300:8:

Management.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 201:35:

Miscellaneous

Similar variable names:

Management.updateIsCancelDeposit(bool) : Variables have very similar names "isCancelDeposit" and "isCancelDeposit_".
Pos: 100:32:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 58:8:

Proof.sol

Miscellaneous

Constant/View/Pure functions:

PendingRequest.increase(struct PendingRequestData,uint256,uint256,uint256,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 64:4:

Similar variable names:

Proof.updateMetadata(address) : Variables have very similar names "metadata" and "metadata_". Note: Modifiers are currently not considered by this static analysis.
Pos: 77:37:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 300:8:

SafeHouse.sol

Miscellaneous

Similar variable names:

SafeHouse.updateMaxWithdrawalCapacity(uint256) : Variables have very similar names "maxWithdrawalCapacity" and "maxWithdrawalCapacity_". Note: Modifiers are currently not considered by this static analysis.
Pos: 66:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 149:8:

Token.sol

Miscellaneous

Constant/View/Pure functions:

Token._updateHoldTime(address,uint256) : Potentially should be constant/view/pure but is not.
more
Pos: 136:4:

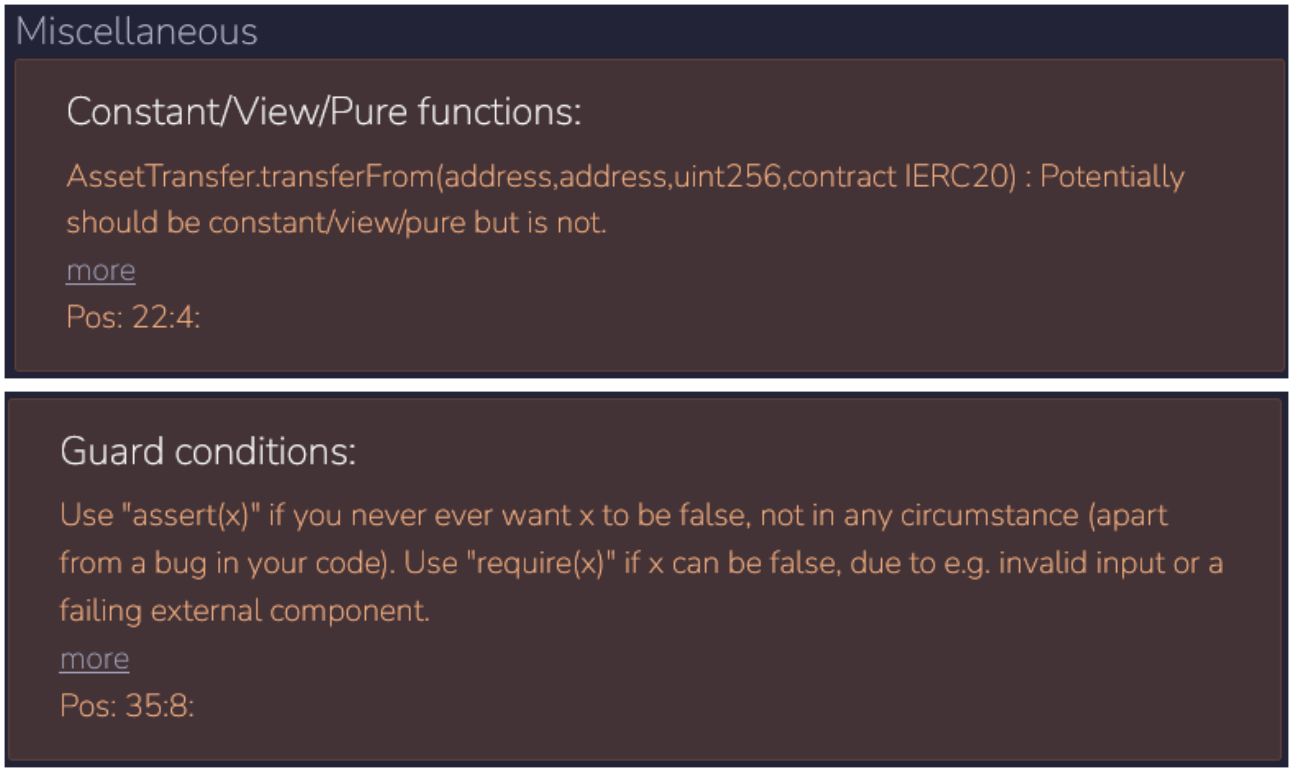
Similar variable names:

Token.updateInvestment(address) : Variables have very similar names "investment" and "investment_".
Pos: 45:16:

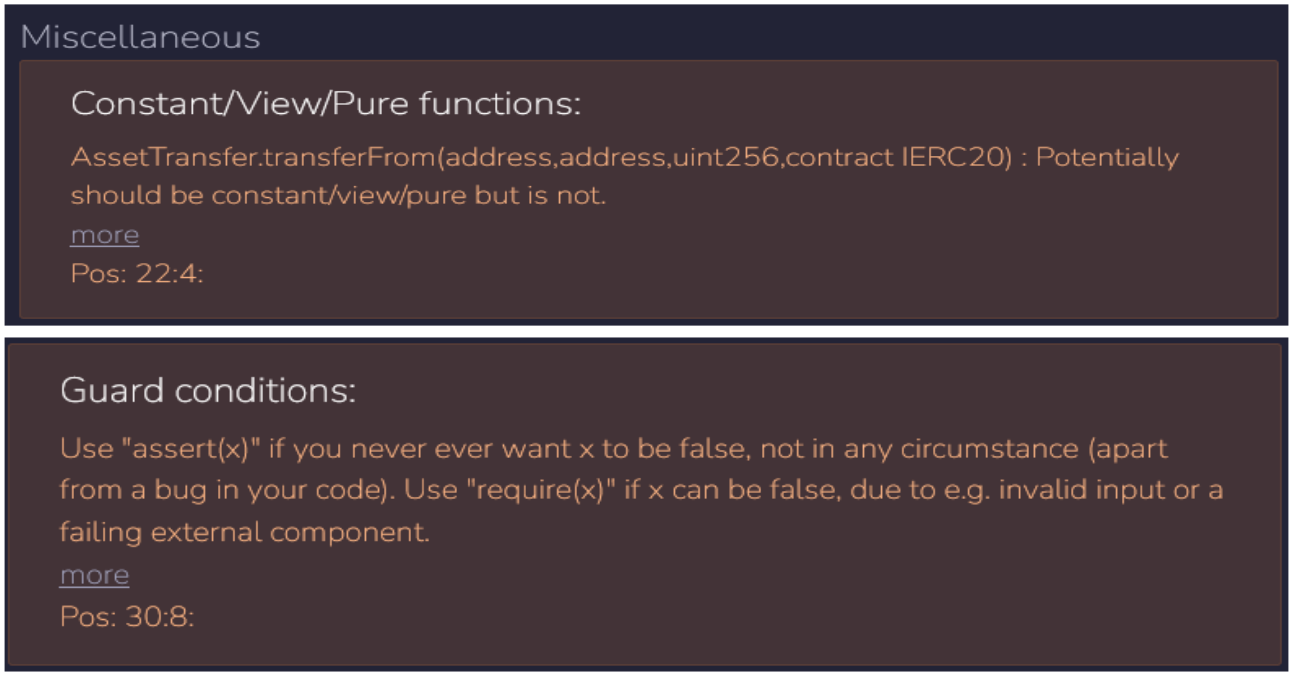
Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 137:8:

Treasury.sol



AssetBookAlpha.sol



DepositProofAlpha.sol

Miscellaneous

Constant/View/Pure functions:

PendingRequest.increase(struct PendingRequestData,uint256,uint256,uint256,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 64:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 321:8:

HoldTimeAlpha.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HoldTime.updateHoldTime(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 42:4:

Miscellaneous

Similar variable names:

HoldTime.updateToken(address) : Variables have very similar names "token" and "token_". Note: Modifiers are currently not considered by this static analysis.

Pos: 30:16:

InvestmentAlpha.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 291:25:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 321:8:

ManagementAlpha.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 201:35:

Miscellaneous

Similar variable names:

Management.(address,address,address) : Variables have very similar names "treasury" and "treasury_".

Pos: 59:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 239:8:

SafeHouseAlpha.sol

Miscellaneous

Similar variable names:

SafeHouse.addVault(address) : Variables have very similar names "vaults" and "vault_". Note: Modifiers are currently not considered by this static analysis.
Pos: 122:15:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 206:8:

TokenAlpha.sol

Miscellaneous

Constant/View/Pure functions:

Token._updateHoldTime(address,uint256) : Potentially should be constant/view/pure but is not.
more
Pos: 136:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 46:8:

TreasuryAlpha.sol

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transferFrom(address,address,uint256,contract IERC20) : Potentially should be constant/view/pure but is not.
more
Pos: 22:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 35:8:

WithdrawalProofAlpha.sol

Miscellaneous

Constant/View/Pure functions:

PendingRequest.increase(struct PendingRequestData,uint256,uint256,uint256,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 64:4:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

Pos: 300:8:

AssetBookBeta.sol

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transferFrom(address,address,uint256,contract IERC20) : Potentially should be constant/view/pure but is not.

more

Pos: 22:4:

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transferFrom(address,address,uint256,contract IERC20) : Potentially should be constant/view/pure but is not.

more

Pos: 22:4:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 34:18:

DepositProofBeta.sol

Miscellaneous

Constant/View/Pure functions:

PendingRequest.increase(struct PendingRequestData,uint256,uint256,uint256,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 64:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 77:8:

HoldTimeBeta.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HoldTime.updateHoldTime(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 42:4:

Gas & Economy

Gas costs:

Gas requirement of function HoldTime.updateHoldTime is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 42:4:

Miscellaneous

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 46:31:

InvestmentBeta.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 291:25:

Miscellaneous

Similar variable names:

Investment.(uint256,address,address,address,address,address,address) : Variables have very similar names "management" and "management_". Note: Modifiers are currently not considered by this static analysis.
Pos: 93:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 321:8:

ManagementBeta.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 201:35:

Miscellaneous

Similar variable names:

Management.(address,address,address) : Variables have very similar names "treasury" and "treasury_".
Pos: 59:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 200:8:

SafeHouseBeta.sol

Miscellaneous

Similar variable names:

SafeHouse.updateMaxWithdrawalCapacity(uint256) : Variables have very similar names "maxWithdrawalCapacity" and "maxWithdrawalCapacity_". Note: Modifiers are currently not considered by this static analysis.
Pos: 66:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 53:8:

TokenBeta.sol

Miscellaneous

Constant/View/Pure functions:

Token._updateHoldTime(address,uint256) : Potentially should be constant/view/pure but is not.
more
Pos: 136:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 61:8:

TreasuryBeta.sol

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transferFrom(address,address,uint256,contract IERC20) : Potentially should be constant/view/pure but is not.
more
Pos: 22:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 35:8:

WithdrawalProofBeta.sol

Miscellaneous

Constant/View/Pure functions:

PendingRequest.increase(struct PendingRequestData,uint256,uint256,uint256,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 64:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 106:8:

AssetBookGamma.sol

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transfer(address,uint256,address) : Potentially should be constant/view/pure but is not.

more

Pos: 46:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 35:8:

DepositProofGamma.sol

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 121:8:

HoldTimeGamma.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HoldTime.updateHoldTime(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 42:4:

Miscellaneous

Similar variable names:

HoldTime.updateToken(address) : Variables have very similar names "token" and "token_". Note: Modifiers are currently not considered by this static analysis.

Pos: 31:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 43:8:

InvestmentGamma.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 288:16:

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 135:8:

ManagementGamma.sol

Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 201:35:

Miscellaneous

Similar variable names:

Management.updateTreasury(address) : Variables have very similar names "treasury" and "treasury_".
Pos: 73:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 352:8:

SafeHouseGamma.sol

Miscellaneous

Similar variable names:

SafeHouse.updatePriceToleranceRate(uint256) : Variables have very similar names "priceToleranceRate" and "priceToleranceRate_". Note: Modifiers are currently not considered by this static analysis.
Pos: 95:38:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 120:8:

TokenGamma.sol

Miscellaneous

Similar variable names:

Token._updateHoldTime(address,uint256) : Variables have very similar names "account_" and "amount_".
Pos: 137:16:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 137:8:

TreasuryGamma.sol

Miscellaneous

Constant/View/Pure functions:

AssetTransfer.transfer(address,uint256,address) : Potentially should be constant/view/pure but is not.

more

Pos: 46:4:

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 34:18:

WithdrawalProofGamma.sol

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 321:8:

Solhint Linter

AssetBook.sol

```
AssetBook.sol:3:1: Error: Compiler version ^0.8.4 does not satisfy
the r semver requirement
AssetBook.sol:27:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
```

HoldTime.sol

```
HoldTime.sol:3:1: Error: Compiler version ^0.8.4 does not satisfy the
r semver requirement
HoldTime.sol:48:13: Error: Avoid to make time-based decisions in your
business logic
```

Investment.sol

```
Investment.sol:277:22: Error: Parse error: missing ';' at '{'
Investment.sol:289:22: Error: Parse error: missing ';' at '{'
Investment.sol:294:18: Error: Parse error: missing ';' at '{'
```

Management.sol

```
Management.sol:290:18: Error: Parse error: missing ';' at '{'
Management.sol:308:26: Error: Parse error: missing ';' at '{'
```

Proof.sol

```
Proof.sol:325:18: Error: Parse error: missing ';' at '{'
```

SafeHouse.sol

```
SafeHouse.sol:182:18: Error: Parse error: missing ';' at '{'
```

Token.sol

```
Token.sol:3:1: Error: Compiler version ^0.8.4 does not satisfy the r
semver requirement
Token.sol:26:5: Error: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Token.sol:126:80: Error: Code contains empty blocks
Token.sol:136:30: Error: Variable "account_" is unused
```

Treasury.sol

```
Treasury.sol:3:1: Error: Compiler version ^0.8.4 does not satisfy the
r semver requirement
Treasury.sol:17:5: Error: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
Treasury.sol:17:41: Error: Code contains empty blocks
Treasury.sol:21:32: Error: Code contains empty blocks
```

AssetBookAlpha.sol

```
AssetBookAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
AssetBookAlpha.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
AssetBookAlpha.sol:11:79: Error: Code contains empty blocks
```

DepositProofAlpha.sol

```
DepositProofAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
DepositProofAlpha.sol:10:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
DepositProofAlpha.sol:10:70: Error: Code contains empty blocks
```

HoldTimeAlpha.sol

```
HoldTimeAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
HoldTimeAlpha.sol:10:1: Error: Code contains empty blocks
```

InvestmentAlpha.sol

```
InvestmentAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
InvestmentAlpha.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
InvestmentAlpha.sol:18:81: Error: Code contains empty blocks
```

ManagementAlpha.sol

```
ManagementAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
ManagementAlpha.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
ManagementAlpha.sol:15:47: Error: Code contains empty blocks
```

SafeHouseAlpha.sol

```
SafeHouseAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
SafeHouseAlpha.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
SafeHouseAlpha.sol:15:44: Error: Code contains empty blocks
```

TokenAlpha.sol

```
TokenAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not satisfy
the r semver requirement
TokenAlpha.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
TokenAlpha.sol:14:57: Error: Code contains empty blocks
```

TreasuryAlpha.sol

```
TreasuryAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
TreasuryAlpha.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
TreasuryAlpha.sol:11:50: Error: Code contains empty blocks
```

WithdrawalProofAlpha.sol

```
WithdrawalProofAlpha.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
WithdrawalProofAlpha.sol:10:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
WithdrawalProofAlpha.sol:10:70: Error: Code contains empty blocks
```

AssetBookBeta.sol

```
AssetBookBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
AssetBookBeta.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
AssetBookBeta.sol:11:79: Error: Code contains empty blocks
```

DepositProofBeta.sol

```
DepositProofBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
DepositProofBeta.sol:10:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
DepositProofBeta.sol:10:68: Error: Code contains empty blocks
```

HoldTimeBeta.sol

```
HoldTimeBeta.sol:2:1: Error: Compiler version ^0.8.4 does not satisfy
the r semver requirement
HoldTimeBeta.sol:10:1: Error: Code contains empty blocks
```

InvestmentBeta.sol

```
InvestmentBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
InvestmentBeta.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
InvestmentBeta.sol:18:81: Error: Code contains empty blocks
```

ManagementBeta.sol

```
ManagementBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
```

```
ManagementBeta.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
ManagementBeta.sol:15:47: Error: Code contains empty blocks
```

SafeHouseBeta.sol

```
SafeHouseBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
SafeHouseBeta.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
SafeHouseBeta.sol:15:44: Error: Code contains empty blocks
```

TokenBeta.sol

```
TokenBeta.sol:2:1: Error: Compiler version ^0.8.4 does not satisfy
the r semver requirement
TokenBeta.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
TokenBeta.sol:14:55: Error: Code contains empty blocks
```

TreasuryBeta.sol

```
TreasuryBeta.sol:2:1: Error: Compiler version ^0.8.4 does not satisfy
the r semver requirement
TreasuryBeta.sol:11:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
TreasuryBeta.sol:11:50: Error: Code contains empty blocks
```

WithdrawalProofBeta.sol

```
WithdrawalProofBeta.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
WithdrawalProofBeta.sol:10:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
WithdrawalProofBeta.sol:10:68: Error: Code contains empty blocks
```

AssetBookGamma.sol

```
AssetBookGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
AssetBookGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
```

```
AssetBookGamma.sol:11:79: Error: Code contains empty blocks
```

DepositProofGamma.sol

```
DepositProofGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
DepositProofGamma.sol:10:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
DepositProofGamma.sol:10:70: Error: Code contains empty blocks
```

HoldTimeGamma.sol

```
HoldTimeGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
HoldTimeGamma.sol:10:1: Error: Code contains empty blocks
```

InvestmentGamma.sol

```
InvestmentGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
InvestmentGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
InvestmentGamma.sol:18:81: Error: Code contains empty blocks
```

ManagementGamma.sol

```
ManagementGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
ManagementGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
ManagementGamma.sol:15:47: Error: Code contains empty blocks
```

SafeHouseGamma.sol

```
SafeHouseAlphaGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
SafeHouseAlphaGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
SafeHouseAlphaGamma.sol:15:44: Error: Code contains empty blocks
```

TokenGamma.sol

```
TokenAlphaGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
TokenAlphaGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
TokenAlphaGamma.sol:14:57: Error: Code contains empty blocks
```

TreasuryGamma.sol

```
TreasuryAlphaGamma.sol:2:1: Error: Compiler version ^0.8.4 does not
satisfy the r semver requirement
TreasuryAlphaGamma.sol:11:5: Error: Explicitly mark visibility in
function (Set ignoreConstructors to true if using solidity >=0.7.0)
TreasuryAlphaGamma.sol:11:50: Error: Code contains empty blocks
```

WithdrawalProofGamma.sol

```
WithdrawalProofAlphaGamma.sol:2:1: Error: Compiler version ^0.8.4
does not satisfy the r semver requirement
WithdrawalProofAlphaGamma.sol:10:5: Error: Explicitly mark visibility
in function (Set ignoreConstructors to true if using solidity
>=0.7.0)
WithdrawalProofAlphaGamma.sol:10:70: Error: Code contains empty
blocks
```

Software analysis result:

These software reported many false positive results and some are informational issues. So, those issues can be safely ignored.

Email: audit@EtherAuthority.io