CMTAT - Audit-20210910

Introduction

This document concerned the audit report ABDK-CMTAT-audit-20210910.pdf which was made on the project CMTAT.

The audited version is that of commit cd3af7

Vocabulary

Header	Description
Id	Id of the concerned CVF
Severity	Same notation that in the report Color : Minor | Moderate | Major
Object	The concerned object [Functions, variables, event name, etc..]
Decision	Which decision has been taken and why
Status	[Full Fixed] => CVF has been fixed as recommended by the audit [Fixed] => CVF has been fixed, but there are some modifications compared to the audit [In progress] => CVF is being fixed (pull request / issue) [Partially fixed] => CVF is fixed in part, but there are still issues to be resolved [No better solution found] => CVF was resolved as best we could [Open] => CVF is NOT fixed, not processed

<table>
<tr><td></td><td>[No longer relevant] => CVF is no longer relevant, e.g., because the problematic functionality was removed.
[Closed] => A decision to not fix the CVF has been taken<table><tr><td>Full Fixed</td><td>Fixed</td><td>Partially fixed</td><td>No (better) solution found</td><td>Open</td><td>No longer relevant</td><td>Closed</td></tr></table></td></tr>
<tr><td>Commit / pull request fix</td><td>The concerned commit / pull request on github which fix the CVF</td></tr>
<tr><td>Remark</td><td>Any other relevant information, by example suggestions to fix the CVF.</td></tr>
</table>

Summary

ID	Severity	Module	Object [Functions, variables, event name, etc..]	Description (evt recommendation)	Decision	Status	Commit / pull request fix	Remark
1	Minor	SnapshotModule.sol All modules are concerned	pragma solidity	Solidity version	To avoid to change all the files for each little upgrade of solidity, we keep a floating pragma with the last tested version	Closed		See issues/68
2	Minor	SnapshotModule.sol	-	The version of the imported library should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	

3	Minor	SnapshotModule.sol	-	Provide a short function description and parameter meaning in the comments preceding the function.		Full Fixed	Commit	
4	Minor	SnapshotModule.sol	SNAPSHOTER_ROLE	The word “snapshoter” (12K results in Google) sounds odd.	The variable is renamed in snapshooter	Full Fixed	pull/13 Commit	
5	Minor	SnapshotModule.sol	_currentSnapshot	This variable is not initialized.	explicitly initializing to 0 Fixed	Full Fixed	pull/14 Commit	The commit fix broke the proxy construction. Commit fix
6	Moderate	SnapshotModule.sol	_scheduled Snapshots	Using an unordered list of scheduled snapshots and removing already created snapshots from it is suboptimal and have several important drawbacks [...]	-Use an ordered list of snapshots instead of an unordered list -We can still remove a scheduled snaphot (time situated in the future) from the list, but the order is maintained -We keep the principle to use the time to identify a snapshot (not the snapshot IDs) -We can only have one scheduled snapshot for a given time	Fixed	Pull/123 Main commit	
7	Minor	SnapshotM	scheduleSnap	Function scheduleSnapshot	No return value	Full	Pull/76	

		odule.sol	shot	always returns its argument.		Fixed	Commit	
8	Minor	SnapshotModule.sol	_rescheduleSnapshot	Here two passes over the array are made while just one suffices.	The second passage was removed during transformations to use an ordered list of snapshot	No longer relevant	Pull/123 Commit	
9	Minor	SnapshotModule.sol	_rescheduleSnapshot	This function always returns its second argument which is clearly redundant.	No return value	Full Fixed	Pull/76 Commit	
10	Minor	SnapshotModule.sol	_unscheduleSna pshot	function always returns its argument which is clearly redundant	No return value	Full Fixed	Pull/76 Commit	
11	Minor	SnapshotModule.sol	getNextSnapshots()	This method returns also uncleared snapshots from the past.	Fix by using an ordered list of snapshot + moving to the first next snapshot	Full Fixed	Pull/123 Main commit	
12	Minor	SnapshotModule.sol	snapshotBalanceOf snapshotTotalSupply	Is it desired behavior that the current balance is returned for an invalid snapshot time?	Yes, if no snapshots are found, it means that the value of the balance was never modified, we can return the current balance Same behavior as for OpenZeppelin erc20Snapshot	closed		
13	Moderat	SnapshotM	_setCurrentSn	This call reads the entire	Fix by using an ordered	Full	Pull/123	

	e	odule.sol	apshot	snapshot array, which is a significant overhead over each transfer.	list of snapshot (new complexity : O(1))	Fixed	Main commit	
14	Minor	SnapshotModule.sol	_beforeTokenTransfer	The most often used branch is the last one.	refactoring order of conditions as indicated in the audit report	Full Fixed	pull/16 Commit	
15	Minor	SnapshotModule.sol	_valueAt	Semantics of returned values is unclear	Add comments and descriptive names.	Full Fixed	Pull/115 Commit	
16	Minor	SnapshotModule.sol	_getCurrentSnapshot	This function is redundant as it just returns the value of a storage variable.	The function was removed	Full Fixed	Pull/115 Commit	
17	Minor	SnapshotModule.sol	Several functions (see report)	These loops do linear searches through the array whose length is unlimited. This makes the contract vulnerable to DoS attacks by scheduling a large number of snapshots.	Fix by using an ordered list of snapshot + we have an access control on the public function, so only authorized address can execute the function	Full Fixed	Pull/123 Main commit	
18	Minor	SnapshotModule.sol	_clearPastScheduled()	The "_scheduledSnapshots.length" value is read on every iteration.	cache scheduledSnapshot.length in a local variable	Full Fixed	Pull/115 Commit	
19	Minor	SnapshotModule.sol	_clearPastScheduled()	The element to be moved to this position will be once again read from storage	The function was removed	No longer relevant	Pull/123 Main commit	

				at the next iteration				
20	Minor	SnapshotModule.sol	_removeScheduledItem	This line executes even if index is the last element.	Add an if condition	Full Fixed	Pull/115 Commit	The modification is not tested in the tests!!!!
21	Minor	ValidationModule.sol	event RuleEngineSet	The parameter type should be “IRuleEngine”.	Proposed changes have been implemented	Full Fixed	Pull Commit	
22	Minor	ValidationModule.sol	IRule Engine pub lic rule Engine	This module should use “IRule” instead of “IRuleEngine” as it doesn’t need to know that the rule used is actually a composition of several nested rules.	We think it makes sens to keep a distinction between the RuleEngine and the rules. It is clearer.	Closed	It's clearer and functions specific to the ruleEngine can be added to this one	
23	Minor	ValidationModule.sol	IRule Engine public ruleEngine	The module doesn’t allow changing the rule engine after the deployment.	The module ValidationModule implements now a function setRuleEngine. This function was moved from CMTAT.sol to the concerned module	Full Fixed	pull/88 Commit	
24	Minor	ValidationModule.sol	_validateTransfer	This will revert in case the rule engine is not set.	A comment was added to explain the requirement.	Fixed	Pull Commit	
25	Minor	ValidationModule.sol	_messageForTransferRestriction	This will revert in case the rule engine is not set.	A comment was added to explain the requirement	Fixed	Pull Commit	

26	Minor	ValidationModule.sol	_detectTransferRestrictionc	This will revert in case the rule engine is not set.	A comment was added to explain the requirement	Fixed	Pull Commit	
27	Minor	CMTAT.sol	TRANSFER_OK	Defining restriction code constants in several placed is error-prone.	Define an enum in CMTAT.sol for the restriction code	Fixed	Pull/78 Commit	Messages are still defined in each file. Other concerned CVF : CVF-48, CVF-55
28	Moderate	CMTAT.sol	__CMTAT_init	There should be a call to the "__Validation_init_unchained" function somewhere after this call.	Proposed changes have been implemented	Full Fixed	pull/121 Commit	
29	Minor	CMTAT.sol	__CMTAT_init	There should be a call to the "__ERC165_init_unchained" function before this call.	Proposed changes have been implemented	Full Fixed	Pull/105 Commit	
30	Minor	CMTAT.sol	__CMTAT_init	This should be done before the "__Base_init_unchained" call as Base module inherits from ERC20 module.	Proposed changes have been implemented Later, the ERC20 functions have been moved from BaseModule to ERC20BaseModuleFixed	Full Fixed	Pull/105 Commit	
31	Minor	CMTAT.sol	__CMTAT_init	There should be a call to the	No more relevant	No		

				"__ERC2771Context_init_unchained" function before this call.	because __ERC2771Context_init_unchained" do not exist anymore in the recent version of OZ	longer relevant		
32	Minor	CMTAT.sol	mint	This event should be emitted inside the "_mint" function.	_mint is a function from OpenZeppelin library. It is not possible to modify it directly. We can override it but it is a bit overkill	Closed		
33	Minor	CMTAT.sol	BurnFrom (replaced by forceBurn in the version 2.1)	This would emit the "Approval" event (which is not desired) but will not emit the "Spend" event (which is actually desired).	No more relevant, the concerned code was removed by replacing the function burnFrom by forceBurn	No longer relevant		
34	Minor	CMTAT.sol	BurnFrom (replaced by forceBurn in the version 2.1)	This event should be emitted inside the "_burn" function.	_burn is a function from OpenZeppelin library. It is not possible to modify it directly. We can override it but it is a bit overkill	Closed		
35	Minor	CMTAT.sol	Several functions (see report)	With multiple inheritance, it is unclear what base contract is referred as "super" here.	The parent module / base contract is explicitly indicated.	Full Fixed	Pull/92 Commit pull/93 Commit	
36	Minor	CMTAT.sol	DetectTransfer	This logic should be moved to	The ValidationModule	Closed		

			Restriction messageForTransferRestriction	the “PauseModule” contract.	defines the different code of rejection, it is better to keep all the validation logic in the same module.			
37	Minor	CMTAT.sol	detectTransferRestriction	Should be “} else” for readability.	Proposed changes have been implemented	Full Fixed	Pull/17 Commit	
38	Minor	CMTAT.sol	DetectTransferRestriction messageForTransferRestriction	This logic should be moved to the “EnforcementModule” contact.	The ValidationModule defines the different code of rejection, it is better to keep all the validation logic in the same module.	Closed		
39	Minor	CMTAT.sol		This logic should be moved to the “ValidationModule”.	Proposed changes have been implemented	Full Fixed	Commit pull/89	
40	Minor	CMTAT.sol	setTokenId setTerms ~~setTrustedForwarder~~	These functions should log some event. The function setTrustedForwarder was removed from the code.	Define events setTokenId & setTerms	Full Fixed	Pull/80 Commit	
41	Minor	CMTAT.sol	kill	Due to various misuse cases, it is best practice to use selfdestruct only when multiple short-lived contracts are created	The function kill is a legal requirement, it can not be removed. Protections were added : - If the CMTAT is	Closed	pull/102	

					deployed with a proxy, the function kill can only be triggered through the proxy - The function is protected by an access control			
42	Minor	CMTAT.sol	_beforeToken Transfer	These checks should be the first to save gas.	Proposed changes have been implemented	Full Fixed	Pull/18 Commit	
43	Minor	CMTAT.sol	_beforeToken Transfer	This check should be done in the “PauseModule” smart contract.	The code calls a function from the PauseModule. We don't see what more we could do	Closed		
44	Minor	CMTAT.sol	_beforeToken Transfer	This check should be done in the “EnforcementModule” smart contract.	The code calls a function from the EnforcementModule. We don't see what more we could do	Closed		
45	Minor	CMTAT.sol	_beforeToken Transfer	This code should be moved to the “ValidationModule” smart contract.	Move the concerned code inside the ValidationModule	Full Fixed	Commit pull/89	

46	Minor	PauseModule.sol	-	The version of the imported library (OZ) should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	
47	Minor	PauseModule.sol	-	This contract should defines the “__Pause_init” and “__Pause_init_unchained” functions.	Define the functions PauseModule_init & PauseModule_init_unchained	Full Fixed	Pull/104 Commit	
48	Minor	PauseModule.sol	TRANSFER_REJECTED_PAUSED	Defining constants for different transfer rejection reasons in different contracts is error-prone	Define an enum in CMTAT.sol	Full Fixed	Pull/78 Commit	
49	Minor	AuthorizationModule.sol	-	The version of the imported library (OZ) should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	
50	Minor	AuthorizationModule.sol	-	It is a good practice to put a comment into an empty block to explain why it is empty.	The block is no longer empty.	No longer relevant		
51	Minor	AuthorizationModule.sol	-	This contract should define the “__Authorization_init” and “__Authorization_init_unchained” functions.	Define the functions Authorization_init & Authorization_init_unchained	Full Fixed	Pull/104 Commit	
52	Minor	MintModule.sol	-	This module doesn’t actually implement minting functionality.	Move the mint functionality inside the MintModule	Full Fixed	pull/88 Commit	

53	Minor	EnforcementModule.sol		The version of the imported library (OZ) should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	
54	Minor	EnforcementModule.sol	Freeze Unfreeze	These events should probably also have the “reason” parameter.	Add the parameters reasonIndexed & reason	Full Fixed	Pull/122 Commit	
55	Minor	EnforcementModule.sol	TRANSFER_REJECTED_FROZEN	Defining constants for different transfer rejection reasons in different contracts is error-prone.	Define an enum in CMTAT.sol	Full Fixed	Pull/78 Commit	
56	Moderate	EnforcementModule.sol	TEXT_TRANSFER_REJECTED_FROZEN	The message is exactly the same as in the “PauseModule” contract. Also, the message is misleading.	The text message has been changed	Full Fixed	Pull/79 Commit	
57	Minor	MetaTxModule.sol	-	The version of the imported library (OZ) should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	
58	Minor	MetaTxModule.sol	__MetaTx_init	This code relies on how the base contract is implemented.	No more relevant because __ERC2771Context_init_unchained” do not exist anymore in the recent version of OZ	No longer relevant		
59	Major	MetaTxModule.sol	__MetaTx_init_unchained	An unchained initializer is not supposed to call the base contract initializer.	The problematic function call was removed.	Full Fixed	Commit	
60	Minor	BurnModul	-	The version of the imported	The OpenZeppelin	Full	Pull/98/	

		e.sol		library (OZ) should be provided.	version is indicated in the file USAGE.md	Fixed	Commit	
61	Minor	BurnModule.sol	-	a) This module doesn't actually implement burning. b) Also, in inherits from the "Initializable" interface but doesn't have any initializer functions	- Move the burn functionality inside the BurnModule - Add initialize functions	Full Fixed	a) pull/88 Commit b) pull/104 Commit	
62	Minor	BaseModule.sol	-	The version of the imported library (OZ) should be provided.	The OpenZeppelin version is indicated in the file USAGE.md	Full Fixed	Pull/98/ Commit	
63	Minor	BaseModule.sol	-	This requirement is not present in the module documentation	The requirement was added to the documentation in base.md	Full Fixed	Pull/19 Commit	
64	Major	BaseModule.sol	transferFrom	The returned value is ignored.	Proposed changes have been implemented	Full Fixed	Commit	
65	Major	BaseModule.sol	approve	It would be better to call "super.approve" here. The current code relies on the knowledge of how the base contract is implemented.	Proposed changes have been implemented	Full Fixed	Commit	
66	Minor	IRuleEngin	-	It is common practice to	Proposed changes have	Full	Pull/74	

		e.sol		provide a short function description and parameter meaning in the comments preceding the function.	been implemented	Fixed	Commit	
67	Minor	IRuleEngine.sol	-	This interface is redundant. It basically defines a composite rule, i.e. a rule composed from other rules, but for those tokens that use such composite rule, it’s composite nature doesn’t not make any difference and is basically an implementation details.	Not fixed, it is no real necessary to fix this CVF, it is clearer to separate IRuleEngine and IRule. Thus, we have a clear separation between the engine and the rules.	Closed		
68	Minor	IRuleEngine.sol	setRules	Setting all rules at once effectively limits the maximum number of rules, as size of a transaction is limited by block gas limit.	The interface defines the minimal requirements for a ruleEngine. The developers are free to add more sophisticated functions to the ruleEngine An example can be found here: CMTA/RuleEngine	Closed		
69	Minor	IRuleEngine.sol	ValidateTransfer detectTransferRestriction messageForTr	These functions are very similar to the functions defined in the “IRule” interface.	Create two interfaces IERC1404 & IERC1404Wrapper (initially called IERC1404Common)	Full Fixed	Pull/74 Commit-1 Commit-2	

			ansferRestriction					
70	Minor	IRule.sol	-	It is common practice to provide a short function description and parameter meaning in the comments preceding the function.	Proposed changes have been implemented	Full Fixed	pull/74 Commit	
71	Minor	IRule.sol	IRule	The interface name is too generic. The interface is all about transfers, consider reflecting this in the name.	Not really a necessity	Closed		
72	Minor	IRule.sol	detectTransferRestriction canReturnTransferRestrictionCode messageForTransferRestriction	The semantics of these functions is unclear from their signatures.	Add a short description + create interface IERC1404	Full Fixed	pull/74 Commit	
73	Minor	IRule.sol	DetectTransferRestriction canReturnTransferRestrictionCode messageForTransferRestriction	Types narrower than 256 bits don't save gas when used with function arguments or return values, however they limit the range of possible values.	No change The original goal was to be compliant with EIP-1404: ethereum/EIPs#1404 and restrictionCode is uint8	Closed	See pull/70	Remark on the save gas part The parameter argument is linked with variable stored in storage and in storage use uint8 rather than 256 can

								save gas