

PURPOSE

The purpose of this document is to analyse the source code of the **AirNode** App and provide an opinion on the purpose and functionality (i.e., a description of what it does) of the code and also answer three specific questions relating to GDPR.

When operated as intended (i.e., by the API operator), **AirNode** DOES NOT:

- 1) Store any personal data in the solution
- 2) Store or cache any data processed through the **AirNode** solution in a way that would constitute a separate GDPR compliance consideration
- 3) Expose any data to any third parties outside of the API operator and the end data consumer, that would constitute a compliance consideration separately from the API operator and the end data consumer.

PACKAGE FILES ANALYSIS

The analysis in this report is from the Main Files of the code (They are found in the Package directory of the source code files) they end with *.sol file extensions. The folder structure was analysed to determine which files are being used in the final application and other files which are their purely for QA Testing, Online Repository Indexing and Project files used only by the development environment and have no relevance for this report (So I have not reviewed these files for this report). The full breakdown is tabulated in the **Source Code Structure** section towards the end of this document.

I have formatted the report to give an overview of the Source code with the 3 tests mentioned at the beginning of this report.

Each file from the Main Package has been individually accessed and the functions within each file has been documented to give the reader an overview of its purpose in the App.

AIRNODE PARAMETER STORE

2.0 AIRNODEPARAMETERSTORE.SOL

AirnodeParameterStore.sol

Called by the requester admin to create a request for the AirNode to send the funds kept in their designated wallet.

Function Name	Description	Review
setAirnodeParameters	Check authorisation for the current caller and set address if the caller has been authorised	Cached data is used here to for later use on withdrawal. No evidence of this stored or cached for longer than required and is disposed after use.
requestWithdrawal	Check if the wallet has the necessary funds for the withdrawal	*xPub Cached data is used here to fetch withdrawal. No evidence of this stored or cached for longer than is necessary and is disposed after use. *xPub is the master Public Key from a secure encrypted backend. It is secure in the AirNode App aswell.
fulfillWithdrawal	Fulfil the withdrawal with the generated request ID	No evidence for local Data store or caching here. The backend either fulfils or cancels the withdrawal.
checkAuthorizationStatus	Retrieve the AirNode ID, request ID, endpoint ID , Wallet ID & Client Address. Check if the necessary authorizations for this caller are authorised.	No evidence for Data store here. The API checks the backend for approval.

Function Name	Description	Review
getAirnodeParameters	Get the current caller Parameters using the AirNode ID as an index into the DB.	<p>AirNode ID is used as a reference here to pull data from the Backend Server.</p> <p>There is mention of memory which would be caching the data using the xPub for short term use.</p> <p>No evidence of this stored or cached for longer than required and is disposed after use.</p> <p>*xPub is the master Public Key from a secure encrypted backend. It is secure in the AirNode App aswell.</p>

REQUEST STORE

2.1 REQUESTSTORE.SOL

RequestStore.sol

This is used by requesters to manage their endorsements. A requester endorsing a client means that the client can request their requests to be fulfilled by the respective requester's designated wallets.

Function Name	Description	Review
createRequester	Creates the requesters with incrementing indices and initializes their withdrawal request nonces	No evidence for local Data store or caching here.
setRequesterAdmin	Sets the requester admin	No evidence for local Data store or caching here.
setClientEndorsementStatus	Sets the client endorsement status and initializes the client request nonce	No evidence for local Data store or caching here.

TEMPLATE STORE

2.2 TEMPLATESTORE.SOL

TemplateStore.sol

Templates created during runtime also allows information to be transferred using the get Template API.

Function Name	Description	Review
createTemplate	Creates template using the parameter table for the current user	No evidence for local Data store or caching here.
getTemplate	API to get template that was earlier by the create templates function	No evidence for local Data store or caching here.

AIRNODE RRP

2.3 AIRNODERRP.SOL

AirnodeRrp.sol

Clients use this module to make requests and AirNode uses it to fulfil them. In addition, it inherits from the modules that keep records, requesters and templates. It also includes some helper methods that AirNode use to reduce the number of calls they make to blockchain providers.

Function Name	Description	Review
makeRequest	Client is requesting to be endorsed	No evidence for local Data store or caching here.
makeFullRequest	Client is requesting to be Fully endorsed	No evidence for local Data store or caching here.
fulfil	Fulfillment parameters are requested	No evidence for local Data store or caching here.
fail	fails successfully	No evidence for local Data store or caching here.

AIRNODE RRP CLIENT

2.4 AIRNODERRPCLIENT.SOL

AirnodeRrpClient.sol

No Functions defined in this file, only a template pass through file.

CONVENIENCE

2.5 CONVENIENCE.SOL

Convenience.sol

The module that keeps the helper methods that AirNode use to make batch calls module is a helper method for the AirnodeParameterStore, TemplateStore, IConvenience {A helper method for the AirNode to set its parameters and forward the remaining funds in the master wallet to/via the AirNode).

Function Name	Description	Review
setAirnodeParametersAndForwardFunds	Sets the Airnode parameters and forwards funds. A helper method for the Airnode to set its parameters and forward the remaining funds in the master wallet to the Airnode admin.	No evidence for local Data store here.
getAirnodeParametersAndBlockNumber	Gets the Airnode parameters and the block number	No evidence for local Data store or caching here.
getTemplates	Gets templates	No evidence for local Data store or caching here.
checkAuthorizationStatuses	Returns authorisation statuses	No evidence for local Data store or caching here.

SELF AUTHORIZER

2.6 SELFAUTHORIZER.SOL

SelfAuthorizer.sol

Authorizer module where each AirNode is their own admin.

Function Name	Description	Review
setAdminStatus	Sets admin status Set super admin status Reverts if caller not Authorized to change admin status	No evidence for local Data store or caching here.
renounceAdminStatus	Renounces admin status Reverts if caller not Authorized to change admin status	No evidence for local Data store or caching here.
extendWhitelistExpiration	Extends whitelist expiration so the caller can spend longer in the App using the current time stamp. Some callers cannot extend its expiration For this call to be accepted the caller must be an admin or super admin user.	No evidence for local Data store or caching here.
setWhitelistExpiration	Sets whitelist expiration so the caller can spend longer in the App using the current time stamp. Caller is not authorised to set the whitelist. For this call to be accepted the caller must be an admin or super admin user.	No evidence for local Data store or caching here.
setBlacklistStatus	Sets blacklist users. Caller is not authorised to set the blacklist. For this call to be accepted the caller must be an admin or super admin user.	No evidence for local Data store or caching here.
isAuthorized	API to check the callers authorisation status for the current procedure	No evidence for local Data store or caching here.

API₃ AUTHORIZER2.7 API₃AUTHORIZER.SOLAPI₃Authorizer.sol

Authorizer module can use this to signal their type, admin sets the admin statuses of addresses and must have super admin privileges.

Function Name	Description	Review
setAdminStatus	Sets admin status Sets super admin status Reverts if caller not Authorized to change admin status	No evidence for local Data store or caching here.
renounceAdminStatus	Renounces admin status Reverts if caller not Authorized to change admin status	No evidence for local Data store or caching here.
extendWhitelistExpiration	Extends whitelist expiration so the caller can spend longer in the App using the current time stamp. Caller cannot extend its expiration For this call to be accepted the caller has to be an admin or super admin user.	No evidence for local Data store or caching here.
setWhitelistExpiration	Sets whitelist expiration so the caller can spend longer in the App using the current time stamp. Caller is not authorised to set the whitelist. For this call to be accepted the caller has to be an admin or super admin user.	No evidence for local Data store or caching here.
setBlacklistStatus	Sets blacklist users. Caller is not authorised to set the blacklist. For this call to be accepted the caller has to be an admin or super admin user.	No evidence for local Data store or caching here.
isAuthorized	API to check the callers authorisation status for the current procedure	No evidence for local Data store or caching here.

CONCLUSION

This report has been compiled by purely viewing and analysing the source code supplied from Tacita. The final/released **AirNode** Application hasn't been used to influence any part of this report.

The general overview is the **AirNode** App consist of APIs which are communicating with a backend server to authorise the user when making a fuel purchase. Once the user has been authorised from the backend a token is issued. From then on, the users access token is stored by the App to regulate the length of time the current user can access the backend using the token. The access token DOES NOT have any user related information. There is an inbuilt mechanism to expire the token after a X period where all active information is destroyed and the App login terminated.

There is also a mechanism to blacklist the user to prevent them from entering the App. This is fully controlled via the API Calls to the backend server.

All cached values if any are destroyed upon exit or termination of the App.

The App only communicates to the designated backend servers, there is no evidence the App connects to any 3rd Party servers on any other Apps on the installed device.

In final conclusion;

When operated as intended (i.e., by the API operator), **AirNode** DOES NOT:

- 1) Store any personal data in the solution
- 2) Store or cache any data processed through the **AirNode** solution in a way that would constitute a separate GDPR compliance consideration
- 3) Expose any data to any third parties outside of the API operator and the end data consumer, that would constitute a compliance consideration separately from the API operator and the end data consumer.

SOURCE CODE STRUCTURE

2.8 ROOT FOLDER STRUCTURE

The table below shows the Root folder structure of the source code and their relevance in the operation of the App.

Table 1 – Root Folder Source Code structure

Parent Directory	Sub Folders	Filename	Description
\airnode--master			<p>This directory mainly contains the folders for the Project and the GIT Repository (Git).</p> <p>This directory also contains QA Test related config Files for plugins used to test and compile the code before releasing the Application (App).</p>
\airnode--master		.dockerignore	Git repository related files
		.eslintignore	Git repository related files
		.eslintrc.js	Code QA config file for code formatting and layout
		.gitignore	Git repository related files
		.npmrc	Config file Node JS Plugin
		.prettierignore	Git repository related files
		.prettierrc	Git repository related files
		.solhint.json	Code QA config file for Solhint plugin
		.solhintignore	Git repository related files
		Docker.md	Java Docker API Client
		Dockerfile	Java Docker API Client
		jest.config.base.js	Discovering and configuring Test Specs for code QA
		lerna.json	Yarn and Package config file
		LICENSE	License Text File
		package.json	Yarn and Package Installer
		README.md	Git Readme Text File
		tsconfig.build.json	Build config file
		tsconfig.json	Build config file pointer
		yarn.lock	Autogenerated Lock file for the source code
	.github		Git and Code QA repository related folder

Parent Directory	Sub Folders	Filename	Description
	.husky		Git and Code QA repository related folder
	packages		This is where the Main package sits

2.9 PACKAGE FOLDER STRUCTURE

The table below shows the folder structure of the Package folder and their relevance in the operation of the App.

Table 2 – PACKAGE Source Code structure

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\	adapter	.gitignore jest.config.js package.json README.md tsconfig.build.json tsconfig.json yarn.lock	This directory mainly contains the folders for the Project and the GIT Repository (Git). This directory also contains QA Test related config Files for plugins used to test and compile the code before releasing the Application (App).
\airnode-master\packages\adapter	src	index.ts types.ts	This folder contains Index and Type definitions
\airnode-master\packages\adapter\src	clients	http.test.ts http.ts	This folder contains Index and Type definitions
\airnode-master\packages\adapter\src	Request-building	authentication.test.ts authentication.ts build-request.test.ts build-request.ts cookies.test.ts cookies.ts execution.test.ts execution.ts index.ts parameters.test.ts parameters.ts path-parser.test.ts path-parser.ts	This folder contains Index and Type definitions

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\adapter\src	Response-processing	casting.test.ts casting.ts encoding.test.ts encoding.ts extraction.test.ts extraction.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\	admin	.gitignore .npmignore cli-wrapper.js package.json package.publish.json publish.sh README.md tsconfig.build.json tsconfig.json yarn.lock	This folder contains Index and Type definitions
\airnode-master\packages\admin\	example	authorizers.json template.json	This folder contains Index and Type definitions
\airnode-master\packages\admin\	src	cli.ts evm.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\	airnode-abi	.gitignore .prettiignore jest.config.js package.json package.publish.json publish.sh README.md tsconfig.build.json tsconfig.json yarn.lock	This folder contains the publishing package
\airnode-master\packages\airnode-abi\	src	decoding.test.ts decoding.ts encoding.test.ts encoding.ts index.ts types.ts utils.ts	This folder contains Index and Type definitions

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\deployer\		.env.example .gitignore export-env.sh package.json README.md serverless.aws.yml tsconfig.build.json tsconfig.json yarn.lock	This folder contains Index and Type definitions
\airnode-master\packages\deployer\	src	commands.ts config.ts deployer-cli.ts io.ts util.ts	This folder contains definitions to assist the deployer module
\airnode-master\packages\deployer\src\	config-data	config.json.example security.json.example	This folder contains definitions to assist the deployer module
\airnode-master\packages\deployer\src\	evm	evm.ts	This folder contains definitions to assist the deployer module
\airnode-master\packages\deployer\src\	handlers\aws	index.ts	This folder contains Index and Type definitions
\airnode-master\packages\deployer\src\	infrastructure	serverless.ts ssm.ts terraform.ts	This folder contains definitions to assist the deployer module
\airnode-master\packages\deployer\	terraform	.gitignore main.tf outputs.tf variables.tf.workaround	This folder contains definitions to assist the deployer module
\airnode-master\packages\	node	.gitignore .prettiignore jest.config.js jest-e2e.config.js jest-unit.config.js package.json README.md tsconfig.build.json tsconfig.json yarn.lock	This folder contains Index and Type definitions
\airnode-master\packages\node\	__dev__	.env.example config.json.example invoke.ts	This folder contains Index and Type definitions

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\node\	src	constants.test.ts constants.ts index.ts logger.ts types.ts	
\airnode-master\packages\node\src	adapters\http	parameters.test.ts parameters.ts worker.test.ts worker.ts	
\airnode-master\packages\node\src	config	index.ts	This folder contains This folder contains Index and Type definitions
\airnode-master\packages\node\src	coordinator	state.test.ts state.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src	Coordinator\calls	aggregation.test.ts aggregation.ts coordinated-execution.test.ts coordinated-execution.ts disaggregation.test.ts disaggregation.ts index.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src	evm	abi-encoding.test.ts abi-encoding.ts evm-provider.test.ts evm-provider.ts gas-prices.test.ts gas-prices.ts index.ts initialization.test.ts initialization.ts networks.test.ts networks.ts transaction-counts.test.ts transaction-counts.ts utils.test.ts utils.ts wallet.test.ts wallet.ts workers.test.ts workers.ts	This folder contains definitions to assist the test modules

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\node\src\levm	authorization	authorization-application.test.ts authorization-application.ts authorization-fetching.test.ts authorization-fetching.ts index.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src\levm	contracts	airnodeRrp.test.ts airnodeRrp.ts index.ts types.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\src\levm	fulfilments	api-calls.test.ts api-calls.ts index.test.ts index.ts withdrawals.test.ts withdrawals.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\src\levm	handlers	fetch-pending-requests.test.ts fetch-pending-requests.ts index.ts initialize-provider.test.ts initialize-provider.ts process-transactions.test.ts process-transactions.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src\levm	requests	api-calls.test.ts api-calls.ts blocking.test.ts blocking.ts event-logs.test.ts event-logs.ts events.test.ts events.ts withdrawals.test.ts withdrawals.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src\levm	templates	index.ts template-application.test.ts template-application.ts template-fetching.test.ts template-fetching.ts template-verification.test.ts template-verification.ts	This folder contains definitions to assist the test modules

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\node\src\levm	verification	api-call-verification.test.ts api-call-verification.ts index.ts request-verification.test.ts request-verification.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src\	handlers	call-api.test.ts call-api.ts index.ts initialize-provider.test.ts initialize-provider.ts process-transactions.test.ts process-transactions.ts start-coordinator.test.ts start-coordinator.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\src\	providers	index.ts initialize.test.ts initialize.ts state.test.ts state.ts worker.test.ts worker.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\src\	requests	grouping.test.ts grouping.ts index.ts nonces.test.ts nonces.ts request.test.ts request.ts sorting.test.ts sorting.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\src\	utils	array-utils.test.ts array-utils.ts date-utils.test.ts date-utils.ts object-utils.test.ts object-utils.ts promise-utils.test.ts promise-utils.ts string-utils.test.ts string-utils.ts	This folder contains definitions to assist the test modules

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\node\src\	workers	index.test.ts index.ts local-handlers.test.ts local-handlers.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\src\	cloud-platforms	aws.test.ts aws.ts	This folder contains definitions to assist the packaging modules
\airnode-master\packages\node\test\	e2e	no-duplication.feature.ts request-status.feature.ts withdrawals.feature.ts	This folder contains definitions to assist the test modules
\airnode-master\packages\node\test\	fixtures	aggregated-api-call.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\ fixture\	config	config.ts index.ts node-settings.ts ois.ts worker-options.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\ fixture\	evm	airnodeRrp.ts event-logs.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\ fixture\	operation	deploy-config.ts index.ts requests.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\ fixture\	provider-states	evm.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\ fixture\	requests	api-calls.ts api-call-templates.ts index.ts metadata.ts withdrawals.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\	helpers	import.ts index.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\	setup\ e2e	deployment.ts index.ts utils.ts	This folder contains Index and Type definitions
\airnode-master\packages\node\test\	setup\ init	set-define-property.ts set-env-vars.ts	This folder contains definitions to assist the test modules

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\	ois	.gitignore package.json tsconfig.build.json tsconfig.json	This folder contains Index and Type definitions
\airnode-master\packages\ois	src	index.ts types.ts	This folder contains Index and Type definitions
\airnode-master\packages\ois\src\	validator	index.ts	This folder contains Index and Type definitions
\airnode-master\packages\	operation	.gitignore ecosystem.config.js hardhat.config.ts package.json README.md tsconfig.build.json tsconfig.json yarn.lock	This folder contains Index and Type definitions
\airnode-master\packages\operation\	src	index.ts server.ts types.ts	This folder contains Index and Type definitions
\airnode-master\packages\src\	config	evm-dev-config.json	This folder contains Index and Type definitions
\airnode-master\packages\src\	evm\deploy	account-assignment.ts airnodes.ts deployment.ts funding.ts index.ts requesters.ts state.ts	This folder contains Index and Type definitions
\airnode-master\packages\src\	evm\requests	index.ts make-requests.ts state.ts	This folder contains Index and Type definitions
\airnode-master\packages\src\	scripts	evm-dev-deploy.ts evm-dev-requests.ts	This folder contains Index and Type definitions

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\	protocol	.eslintignore .eslintrc.js .gitignore .npmignore .prettierrignore .solcover.js credentials.example.json hardhat.config.js index.js package.json package.publish.json publish.sh README.md yarn.lock	This folder contains Index and Type definitions
\airnode-master\packages\	protocol\contracts	AirnodeParameterStore.sol AirnodeRrp.sol AirnodeRrpClient.sol Convenience.sol RequesterStore.sol TemplateStore.sol	
\airnode-master\packages\protocol\contracts\	authorizers	Api3Authorizer.sol SelfAuthorizer.sol	Backend Authorization code
\airnode-master\packages\protocol\contracts\authorizers\	interfaces	IApi3Authorizer.sol IAuthorizer.sol ISelfAuthorizer.sol	Backend Authorization code
\airnode-master\packages\protocol\contracts\	mock	MockAuthorizerAlwaysFalse.sol MockAuthorizerAlwaysTrue.sol	Mock user testing code
\airnode-master\packages\protocol\	deploy	IAirnodeParameterStore.sol IAirnodeRrp.sol IConvenience.sol IRequesterStore.sol ITemplateStore.sol	The Protocol libraries for the Airnode App sits here
\airnode-master\packages\protocol\contracts\	mock	MockAirnodeRrpClient.sol	Mock Client Airnode access for testing
\airnode-master\packages\protocol\	deploy	1_deploy.js	Deploy account API caller
\airnode-master\packages\protocol\	test	AirnodeRrp.sol.js AirnodeRrpClient.sol.js Convenience.sol.js ProviderStore.sol.js RequesterStore.sol.js TemplateStore.sol.js	The Main Code for the Airnode App sits here

Parent Directory	Sub Folders	Filenames	Description
\airnode-master\packages\protocol\test\	authorizers	Api3Authorizer.sol.js SelfAuthorizer.sol.js	The Sub Code for the Airnode App sits here

APPENDIX

Ref ID	Ref Link
1	https://dzone.com/articles/introducing-amp-assimilating-caching-quick-read-fo