Reportable Situation API Specification

Version 2.0

26 Apr 2026

ASIC Business Contacts : **Scott Collins, Nathaniel Grant**

ASIC Technical Contacts : **Arul Murugan Kanagaraj, Yurong Wei (Jess)**

ASIC
Australian Securities & Investments Commission

Contents

Version History

Version	Date	Change Comments
1.0	29/8/2022	Published Version
1.1	28/9/2022	1. Corrections to Reportable Situations ((YAML) file - for detailed list of changes, refer **Appendix K: Modifications V1.1 to Reportable Situation (YAML)** - for updated Reportable Situation (YAML) file refer **Appendix A: Open API specification, Reportable Situation (YAML).** 2. Removed GetRecordofTransaction Sample Request & Response. 3. Clarification added to section: API Versioning
1.2	19/10/2022	1. Corrections to Reportable Situations ((YAML) file - for detailed list of changes, refer **Appendix L: Modifications V1.2 to Reportable Situation (YAML)** - for updated Reportable Situation (YAML) file refer **Appendix A: Open API specification, Reportable Situation (YAML).** 2. Further clarification added to section: API Versioning
1.3	24/11/2022	Updated sample with headers – Appendix B
1.4	26/04/2023	- updated Reportable Situation (YAML) file refer **Appendix A: Open API specification, Reportable Situation (YAML).** Included note for breach_since_occurred json field
1.5	18/01/2024	Updated **Appendix D** : Success response
1.6	07/03/2024	Updated yaml definition **Appendix A**
1.7	24/07/2024	Updated yaml definition **Appendix A**
1.8	19/08/2024	Updated Hours of Operation and Outage Windows
1.9	23/10/2024	Updated yaml definition **Appendix A**
2.0	05/03/2026	Updated yaml definition **Appendix A**
2.1	12/06/2026	Added Version Transitioning section

Introduction

Purpose

The purpose of this document is to outline the Reportable Situations Application Programming Interface (API) solution for external organisations to develop software notifying ASIC of reportable situations.

Audience

The primary audience of this document are the organisations external to ASIC who notify ASIC of reportable situations and their technology teams.

Background

From 1 October 2021, the breach reporting reforms for Australian financial services licensees (AFS licensees) require the notification of reportable situations to ASIC. This obligation was extended to include Australian credit licensees (ACLs).

The API provides a machine-to-machine interface solution for high-volume organisations to submit reportable situation transactions.

Further information

Information on the API, the process for applying for access and key documentation list below is available at https://asic.gov.au/online-services/application-programming-interface-api-for-reportable-situations/:

- The API specification document,
- The Reportable Situation API application form, and
- The Terms and Conditions ("user agreement").

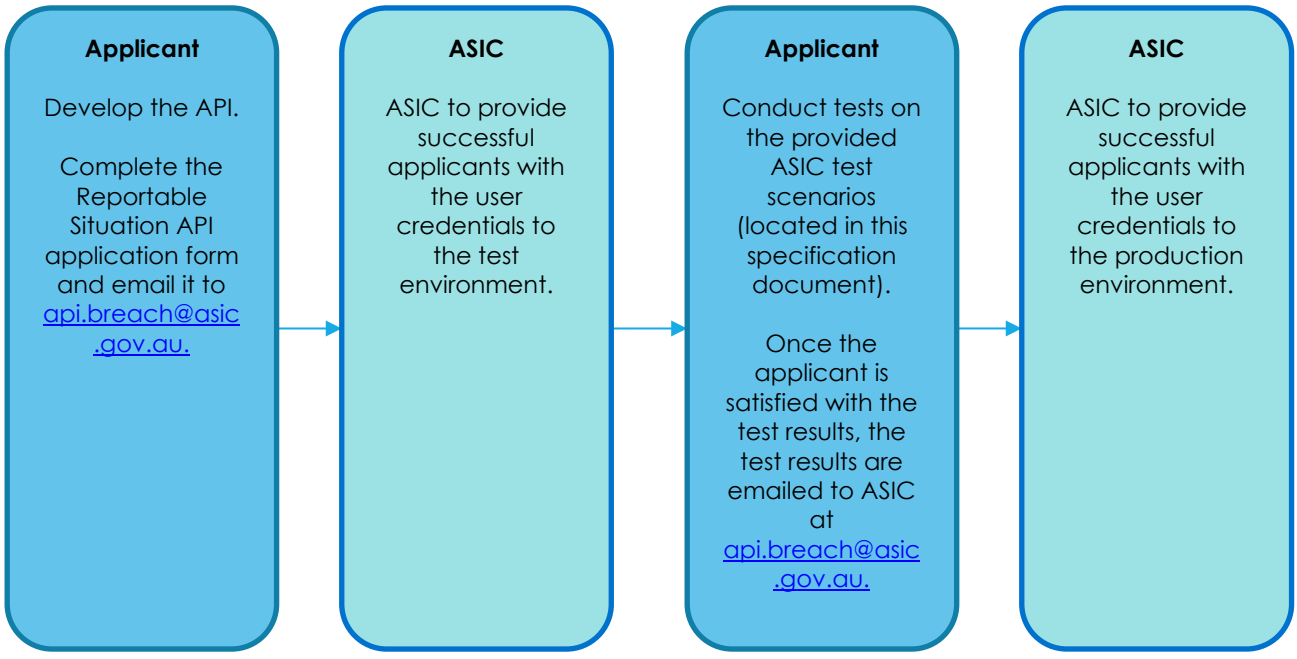
For further information, please email ASIC at api.breach@asic.gov.au.

As part of the BR API enhancement, a new version of the API (v2) will be available from June 2026. The current version of the API (v1) will remain available until ***Friday* March *12th 2027***. It is recommended that API (v2) be used for all new reportable situation submissions from June 2026. Previously lodged submissions will continue to be able to be updated during the transition period, however ***API major versions must not be mixed*** within a reportable situation lifecycle (i.e., initial submission and any subsequent update submission for the same reportable situation must use the same major version: v1 initial + v1 updates, or v2 initial + v2 updates). ***From Saturday 13th March 2027***, API (v1) will no longer be available to submit all reportable situation transactions and API (v2) will be the only supported version.

Glossary of terms

Term	Description
ACN	Australian Company Number
ABN	Australian Business Number
AFSL	Australian Financial Services Licensee
ACL	Australian Credit Licensee
ACR	Australian Credit Representative
API	Application Programming Interface
ROT	Record of Transaction
JSON	Java Script Object Notation
REST	Representative State Transfer

Registration process



Accessing the API

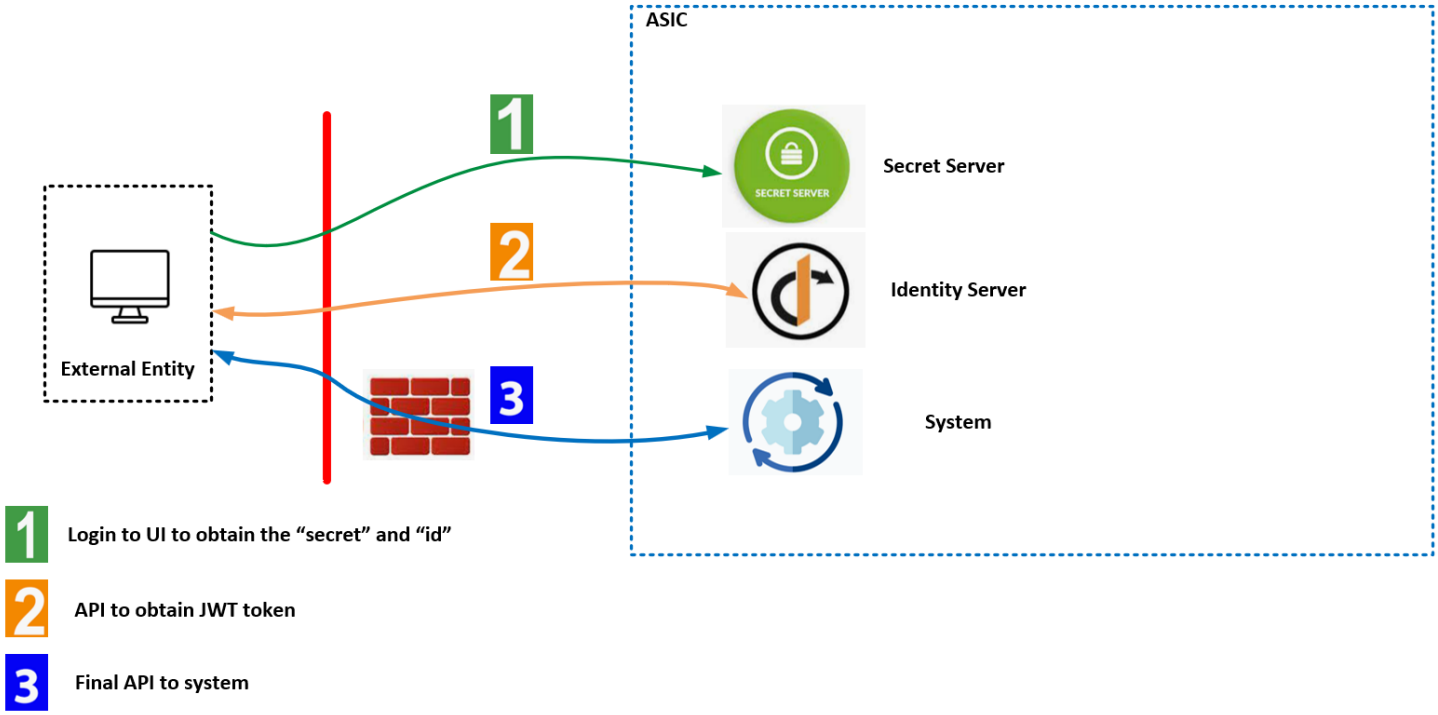
After the entity has successfully submitted onboarding information and ASIC has approved on-boarding, access to the API for testing can be provided.

1. Prior to making the first API call, the entity is required to login to a secret server User Interface (UI) to obtain their Client Id, Client Secret and API Key credentials.

   There are 2x APIs required for a transaction to be successfully authenticated and the reportable situation details submitted for validation and processing.

2. Using the details obtained from the secret server, an API is sent to ASIC's Identity server to obtain an access token.
3. Using the access token, an API is sent to the "createReportableSituation" endpoint with the payload details.

The following diagram summarises the three processes outlined above:



Accessing the Client Id, Client Secret and API Key

ASIC uses the SigBox User Interface (UI) solution to share the Client id, Client Secret & API Key with external parties.

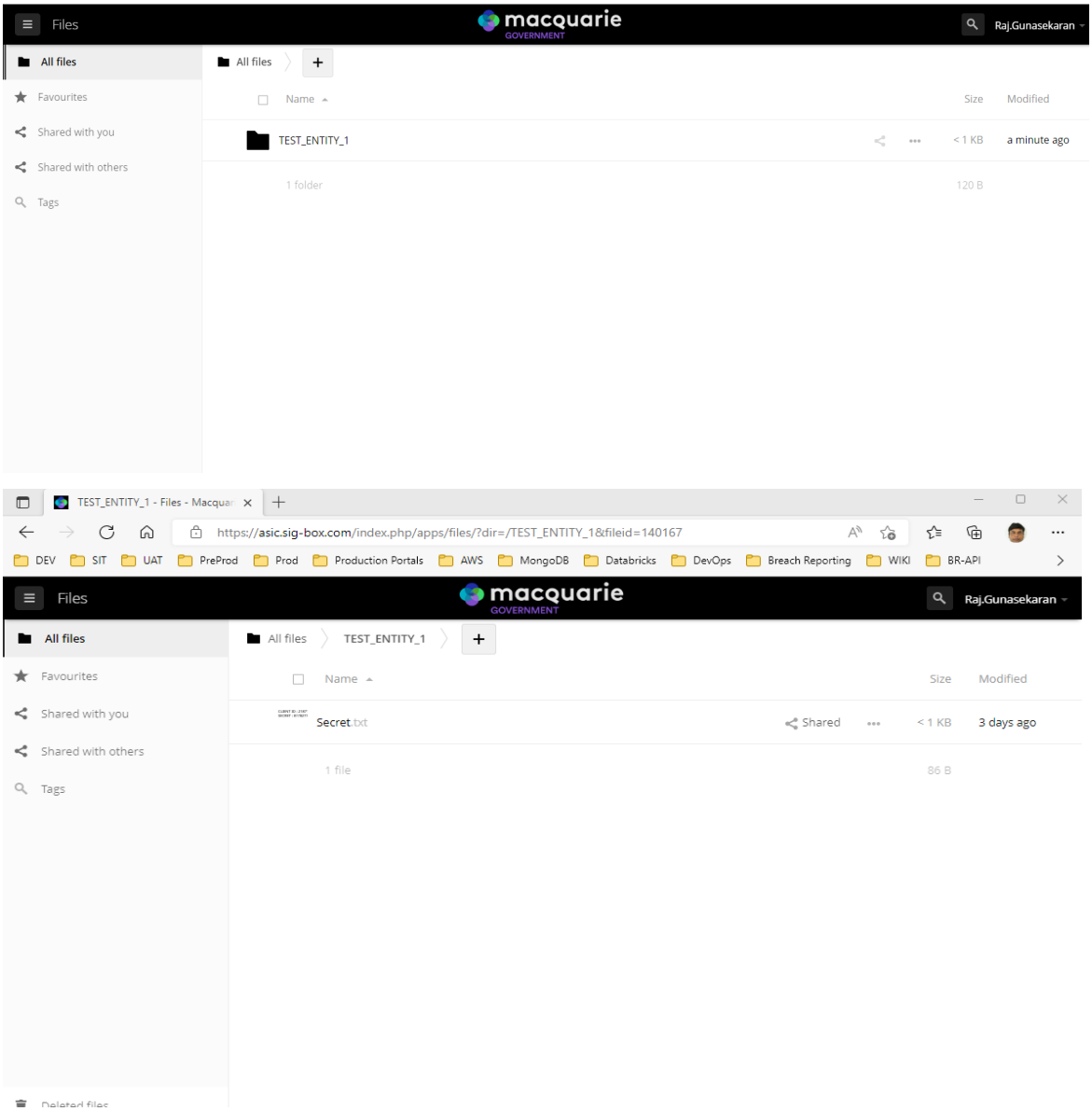
The steps entities are required to follow for accessing the UI & retrieving their credentials are:

1. Users will receive an email from the secret server/SigBox asking to reset the password for logging into the Secret server UI.

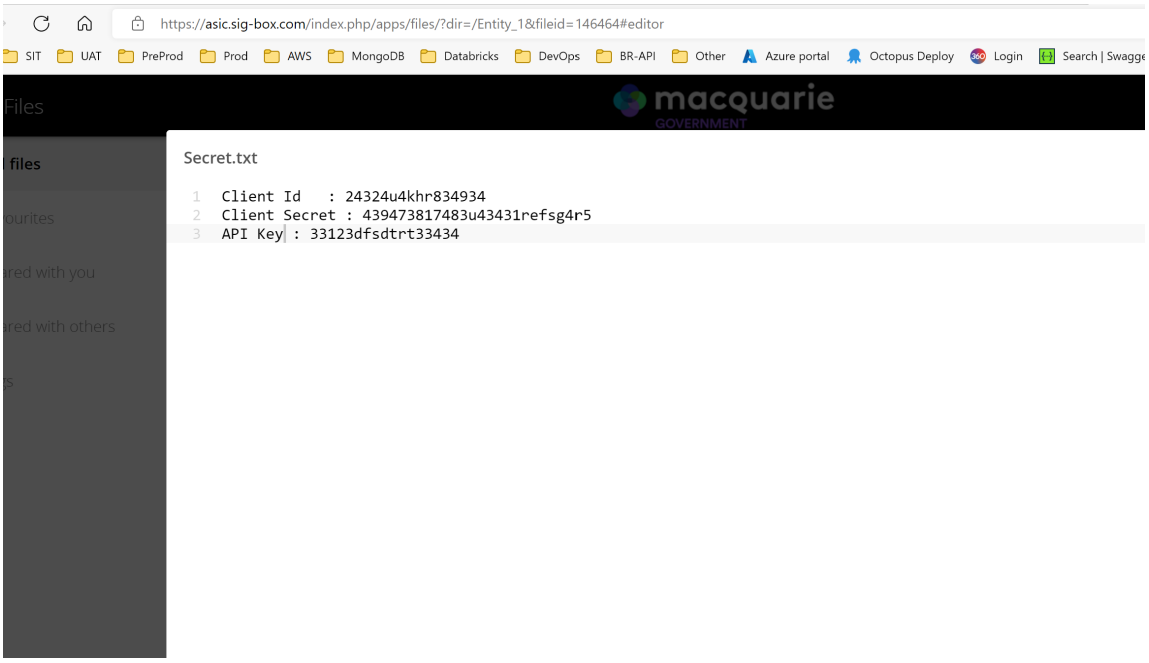
Note: The email address provided on the on-boarding form will be used for creating the account in secret server.

2. Once a user resets the password, they will be able to login into Secret server/SigBox UI

The User will be able to see a folder & a secret file under the folder (screenshots below)



The secret file contains the Client Id, Client Secret & API Keys.



Notes:

- A separate secret file will be created for Test and Production environments
- The expiration period of the client Id and client secret is around 6 months from the date of publishing. There is no expiration for the SigBox user account
- The email address used for SigBox registration will be the email advised by external entities as part of user on-boarding. A generic email address needs to be used for continuity.
- There are currently no plans to implement Multi-Factor Authentication (MFA) for accessing the SigBox application
- Any updates or changes to the Client Id, Client Secret and API Key will be communicated to Business and Technical teams via the email that was provided as part of on-boarding
- Client Id & Client Secret are unique for each entity onboarded
- The following are the rules for SigBox Password complexity:
- Minimum of 10 Characters:
    - At least one number is required
    - At least one lowercase is required
    - At least one uppercase is required
    - At least one special character is required.

Technical summary

Development of this interface has been guided by the Whole of Australian Government (WoAG) Application Programming Interfaces (API) standards. Refer https://www.api.gov.au.

This is a RESTful API service called over HTTPS and managed on API Gateway connecting to a collection of ASIC-Internal backend services that create the reportable situation.

There is a single endpoint URI for the client to call to create a reportable situation transaction. The single API endpoint request method is POST.

JSON is the standard used for transferring data to and from client and server sides (Header Content-Type = application/json). Payload requests sent by entities and server responses will be made using JSON.

The API conforms to REST architecture noting the following:

1. Client-server separation: Clients and servers can evolve independently and are not in any way coupled together, given that the interface (Swagger Doc) is unchanged. If any changes arise, they are communicated as an updated version.
2. Stateless requests: Server side does not store any context that links multiple consecutive API requests.
3. Resource Identifiers: Use of nouns for endpoint paths. The endpoint path for this API will be */v2/createReportableSituation*

API versioning

API's will be versioned such as:

- /v1/
- /v1.1/
- /v2/

Each updated version will be published and communicated with entities.

Production API version will align to the Web-form. Only one version of the API will be available in Production at one time.

Major version releases (eg. V1, V2) will be introduced when changes are required to the API causing previous versions to be incompatible. Implementation of these releases is mandatory.

The complexity of the changes will determine the time period provided for updates to be implemented, the dates for new version test system availability and production change dates. There are no plans to change the URL.

Minor versions (eg. V1.1, 1.2) will be introduced for bug fixes or additional enhancements which do not break the existing interface. ASIC will group together any bug fixes and additional enhancements and advise entities for their review/implementation as required for their solution.

Version Transitioning From v1 To v2

After releasing new version, the previous version is assigned a sunset date, after which that version will not be available. Here are some guidelines to help transition from old to the new version:

- Entities will have the option to continue using old or cut-over to new version, for example v1 and v2, until the sunset date.
- If the entity is ready to cut-over to use the new version before the sunset date, they can do so. However, in order to send updates to the in-flight RS reports using new version, they would need to inform ASIC by sending an email to api.breach@asic.gov in advance with below details:
    1. The date of cut-over
    2. In-flight RS Report details as per the CSV template specified in Appendix N

Once supplied with above information, ASIC will remove entity's access to the old version and they will need to use the new version from the cut-over date. Until the entity provides the in-flight RS report details, these transactions will not be available for updates using the new version.

- On the sunset date,
    1. Every entity's access to the old version will be removed;
    2. All entities must provide the in-flight transaction details as per the format specified in Appendix N, so that ASIC can migrate these transactions to the new version. Until this step, these transactions will not be available for updates using the new version.

Please note that after release of API version v2, new entities should only use API version v2 for RS reports submission.

Security standards

Government Standards

ASIC will follow whole of government standard and ISM controls for this API. These include:

- https://www.api.gov.au
- ISM Guideline for system hardening – Authentication
- ISM guideline for Cryptography: Cypher/Protocol requirements, encryption.

Transport Security

All communication between the requesting endpoint and the serving endpoint shall be encrypted in accordance with the prescribed specification below:

- Transport shall occur using HTTPS TLS 1.3
    - Key exchange
        - DH 3072 bit
        - ECDH 224 bit
        - RSA 3072 bit
    - Symmetric algorithm
        - AES-GCM 256 bit
    - Digital Signatures
        - DSA 2048 bit
        - RSA 3072 bit
    - Hashing algorithm
        - SHA2 348 bit

- Certificates shall be from SHA-2 (Secure Hash Algorithm 2) cryptographic hash functions with minimum key length of 2048.
- HTTP traffic will be rejected; there will be no redirect to HTTPS
- Unused HTTP methods are disabled and will return HTTP 405
- Each distinct API request must be authorised and validated
- ASIC reserves the right to reject certificates signed by certain Certification Authority (CAs).

Connection security

API request rate limiting will be enforced. An acceptable API request rate is around 500 transactions per day for an entity. Should restrictions constrict legitimate business usage, please email api.breach@asic.gov.au.

Input validation checks are performed, if the payload or query structure does not meet the specification it will be rejected.

Content validation check are performed, if any content not defined in the specification is received, the request will be rejected.

Authentication specification

Authentication & Authorisation

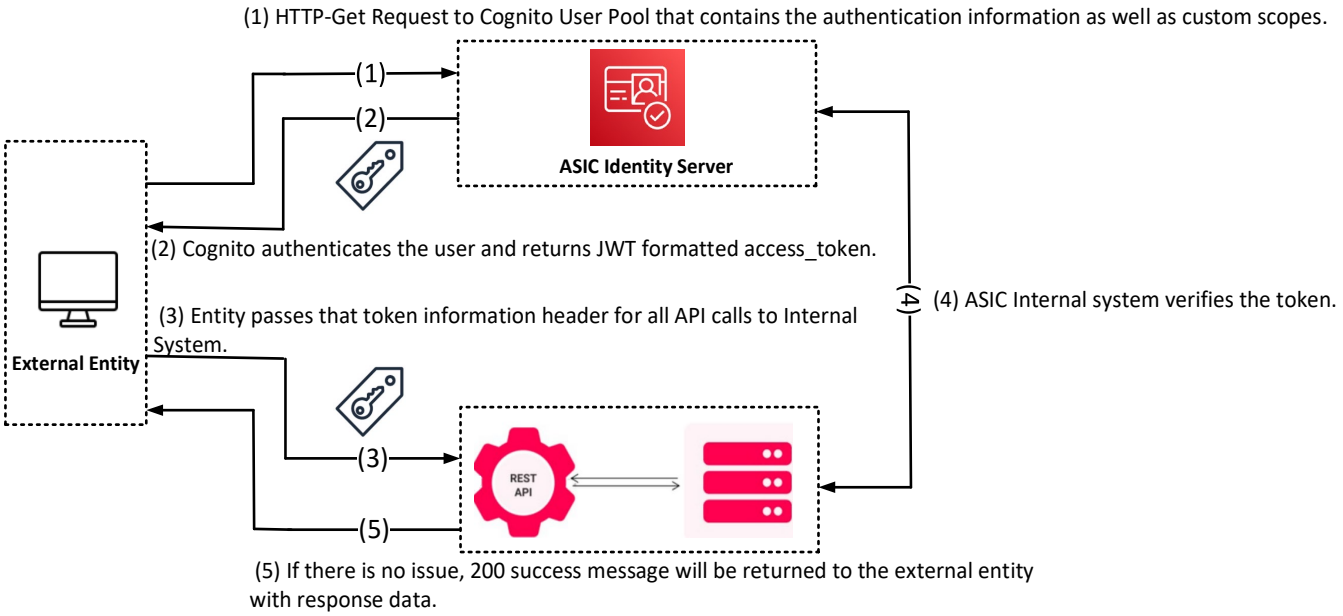
All API Consumers for a resource shall be authenticated to ensure that only authorised entities are granted permission access to the resource APIs. The following are the authentication steps that must be adhered to:

- Only Rest API is supported
- To call the API endpoint authorised entities must provide ASIC with a static IP (Intellectual Property) or static IP range no greater than a /24 subnet that will be added to an allow list. The entity client application request must originate from one of the addresses provided to ASIC
- ASIC uses the OAuth2 protocol (RFC-6749) for machine-to-machine M2M authentication
- M2M credentials is used to obtain a JWT access token from the ASIC's Identity Server
- M2M credentials shall expire no less frequently than once per year but should design in anticipation of quarterly key rotation
- All API Consumers must adhere to security requirements specified in the Terms & Conditions.

Client ID and Client Secret

OAuth 2.0 protocol agreed to be used between third-party entities and ASIC to authenticate and authorize each call.

The following diagram shows the flow for authentication and authorization:



(1) External entity makes a call to ASIC's Identity server. Refer Appendix J: Sample Request and Response for Identity Server API.

(2) Identity server authenticates the call and if successful would return the access token

(3) Subsequent API call needs to be made with the access token obtained in step (2)

(4) ASIC system checks/verifies the access token with Identity server to make sure they are valid. Once token validation is successful, the rest of the API logic / workflow is triggered

(5) Any success/error message is returned.

Note:

- The client must pass its client_id and client_secret in the authorization header through Basic HTTP authorization
- Examples of negative scenarios for ASIC Identity server are documented in the link

  Token endpoint - Amazon Cognito
- The authorization header string is Basic Base64Encode(client_id:client_secret).
- The following example is an authorization header for app client djc98u3jiedmi283eu928 with client secret abcdef01234567890, using the Base64-encoded version of the string djc98u3jiedmi283eu928:abcdef01234567890.

  Details to be passed for this API:

  Header:

Name	Value

Authorization	Basic Base64Encode(client_id:client_secret)

Body:

Name	Value
grant_type	client_credentials
scope	createReportableSituation/POST

Other information:

Name	Value
Method	POST
Access token expiry	60 Minutes

Payloads

Payload contents will be defined in an Open API specification (v3.0) and will be provided as a companion to this document. These are currently documented as appendices and a link will be provided when this is available from the ASIC website.

The mapping between the FDD and the API Specification is included in Appendix H: Mapping Document.

End point

This information will be provided as part of the on-boarding.

Schema validation

As there is complex form logic surrounding the notification of reportable situations, a complex schema validator has been implemented. This provides calling organisations with a reliable method of confirming payload validity.

New reports and Update Reports

There will be a single POST endpoint which will cater for both new reports and update reports.

Note: Do not mix API major versions for initial vs update submissions. If you create an initial reportable situation using v1, all subsequent updates must be submitted using v1. If you create an initial reportable situation using v2, all subsequent updates must be submitted using v2.

Body Structure, Conditional logic and Field properties and Error Handling

This information is provided in the Appendix A - Open API Specification, Reportable Situation (YAML).

Note: The reportable event status mapping is outlined in Appendix J: Reportable Event Status Calculation.

Examples of request payloads

Examples of successful and failed payloads are provided in Appendix B: Example Request Payload and Appendix C: Example Request Payload – Schema Failed

Example of response payloads

Examples of successful and failed responses are provided in Appendix D: Example Response Payload – Success; Appendix E: Example Response Payload – Schema Failed; and Appendix F: Example Response Payload – Internal Error.

Get Record of Transaction (GetRot) request payload

This is provided for test use only. This API can be used to download a record of transaction PDF to check successful submission and data mapping. Refer Appendix G: Open API Specification, Get Record of Transaction (YAML)

Validation process

Organisations ASIC has approved to on-board will be granted access to a test system, allowing them to validate their software against a non-Production API.

They will be provided with a list of tests that must be executed to demonstrate to ASIC that their software is functional, before being granted access to submit reportable situation notifications to ASIC Production systems. This list will be updated from time-to-time based on requirements of the relevant legislation.

Testing will also be required when new versions of the API are released.

External organisations are expected to execute their own test cases, but as a minimum will need to execute the ASIC mandated set of tests.

Examples of the types of tests that will be included in the mandatory set are:

- Submit Reportable situation event – for Initial Transaction
- Submit Reportable situation event – for Update Transaction
- Submit Reportable situation event – for Initial Transaction with multiple related entities (as required)
- Submit Reportable situation event – for Update Transaction with multiple related entities (as required)
- Validate all submissions made via APIs using the GetRoT API (available for testing only)
- Validate the PDF and the contents of the PDF from the GetRoT API (available for testing only)
- Submit Reportable situation event with invalid data and validate the error response.

Hours of Operation and Outage Windows

The API solution design and construction will involve a High-Availability (HA) design ensuring there are no single points of failure.

During planned / unplanned outage windows, the API service will not be available. Any calls to the service will receive an" Internal Server Error" with error code 500 indicating that the service is not available. all attempts during this time will need to be re-sent after the service becomes available at the end of the outage window.

As outage windows are planned, the outages will be posted on the ASIC website at Service availability | ASIC

Appendix A – Open API Specification, Reportable Situation (YAML)

ASIC-Create-Report
able-Situation-v2-sw

Appendix B – Example Request Payload

request_example.json

TestRecordOfTransac
tion.txt

TestReportableSituati
on.txt

**TestReportableSitu
ation_v2.txt**

Appendix C – Example Request Payload – Schema Failed

schema_failed_request.json

Appendix D –Example Response Payload - Success

success_response.json

Appendix E – Example Response Payload – Schema Failed

schema_failed_response.json

Appendix F –Example Response Payload – Internal Error

internal_error.json

Appendix G – Open API Specification, Get Record of Transaction (YAML)

getRot-external.yaml

Appendix H – Mapping document

ASIC%20-%20Breach%20Report%20API%2

Appendix I – Sample Request & Response for Identity Server API

Sample Request


```
  "Request Headers": {
   "authorization": "Basic
M3Fkcms3NTU1a2F0dW12cXBmMDNlbnFtc2M6MWFyODkzbXIwanAzZzBlbXVmNHA4YWg0ZnBjbzFvbWpvMGR
sc2h2MG1oYjIyb3Ztam9n",
   "user-agent": "PostmanRuntime/7.29.2",
   "accept": "*/*",
   "postman-token": "721c5918-62aa-474f-b3aa-ef876e4352e3",
   "host": "<ASIC HOST>",
   "accept-encoding": "gzip, deflate, br",
   "connection": "keep-alive",
   "content-type": "application/x-www-form-urlencoded",
   "content-length": "68"
  },
  "Request Body": {
   "grant_type": "client_credentials",
   "scope": "createReportableSituation/POST"
  }
```


Sample Response


```
  "Response Headers": {
   "date": "Wed, 03 Aug 2022 01:02:19 GMT",
   "content-type": "application/json;charset=UTF-8",
   "transfer-encoding": "chunked",
   "connection": "keep-alive",
   "set-cookie": "XSRF-TOKEN=d0b5159b-2e50-41c7-9e7d-75bffc4af341; Path=/; Secure; HttpOnly;
SameSite=Lax",
   "x-amz-cognito-request-id": "8289ee2f-cd14-471c-89e7-8f7489f22507",
   "x-application-context": "application:prod:8443",
   "x-content-type-options": "nosniff",
   "x-xss-protection": "1; mode=block",
   "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
   "pragma": "no-cache",
   "expires": "0",
   "strict-transport-security": "max-age=31536000 ; includeSubDomains",
   "x-frame-options": "DENY",
   "server": "Server"
  },
```

"Response Body":
"{\"access_token\":\"eyJraWQiOiJMQUdjUGJvdUtyWUZNemdTYkVWQ3JGQ2Rjd3VwenptWTFPSmVmUEZUMGh3PSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiIzcWRyazc1NTVrYXR1bXZxcGYwM2VucW1zYyIsInRva2VuX3VzZSI6ImFjY2Vzcy IsInNjb3BlIjoiY3JlYXRlUmVwb3J0YWJsZVNpdHVhdGlvblwvUE9TVCIsImF1dGhfdGltZSI6MTY1OTQ4ODUzOSwiaXNzIjoiaHR0cHM6XC9cL2NvZ25pdG8taWRwLmFwLXNvdXRoZWFzdC0yLmFtYXpvbmF3cy5jb21cL2FwLXNvdXRoZWFzdC0yX2JSYTVNZXlGciIsImV4cCI6MTY1OTQ4ODgzOSwiaWF0IjoxNjU5NDg4NTM5LCJ2ZXJzaW9uIjoyLCJqdGkiOiI5ZTdiM2QyNC04NWQ2LTQwNWQtODQ4ZC0xY2EwNmMzNzYxNTIiLCJjbGllbnRfaWQiOiIzcWRyazc1NTVrYXR1bXZxcGYwM2VucW1zYyJ9.ddEBokW6E5hLj-EtXxGHEw1FePZBTCTm3PqtJSqdAbCicLK8s14d-rpa0_X7wKPuDb_NbXEbhKMgoEgE5era6S2MVK5brM-vaHLY80IXoBsKszrEwZbWHnlOimoQy5gogHkL4IjT9b-NRTHziALHCD1H6RQqvmUcXVYTlJT5Lfk49o8pYnfMbcenMX6ve13fl8h054gXllfH3kQsG6vPc7FEfrhT3Yud4X7BBLyrE8aHBz2DhWGiN50IhB_wIkweTsh_2Hi1cBWRPKkb5ryAPq03s0ohJmt-LbgQR-vEgSsm3lEGFzH8_aiuIV7R3wen8EcZbuM7M7-0xoDpIe497A\",\"expires_in\":3600,\"token_type\":\"Bearer\"}"
}

Appendix J – Reportable Event Status Calculation

The types of reportable situation referred to in the table are:
A = Has breached
B = Unable to comply (likely breach)
C = Investigation commenced only (nothing found yet)
D = Investigation commenced, completed and no breach found
E = Gross negligence or serious fraud

Reportable situation event status	Type of reportable situation					Investigation complete?		No impact on clients	No client loss	Clients compensated	Started compensating	Intends to compensate?		Breach rectified	Has plan for rectifying	Preparing plan?		Addressed inability to comply	Taking measures re inability to comply	Will take measures to address?	
						P2-S1-6 or P2-S1-7b or P2-S1A-9 =		P2-S4-18 or P2B-S4A-18 =	P2-S4-33 or P2B-S4A-33 =	P2-S5-1 or P2B-S5A-1 =	P2-S5-3 or P2B-S5A-3 =	P2-S5-6 or P2B-S5A-6 =		P2-S5-11 or P2B-S5A-11 =	P2-S5-14 or P2B-S5A-14 =	P2-S5-16 or P2B-S5A-16- =		P2-S5-31aa or P2B-S5A-31aa =	P2-S5-31a or P2B-S5A-31a =	P2-S5-31c or P2B-S5A-31c =	
	A	B	C	D	E	Yes	No[1]	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Complete				X		X															
					X	X		X													
					X	X			X												
					X	X				X											
					X	X							X								
	X					X		X													

Reportable situation event status	Type of reportable situation					Investigation complete?		No impact on clients	No client loss	Clients compensated	Started compensating	Intends to compensate?		Breach rectified	Has plan for rectifying	Preparing plan?		Addressed inability to comply	Taking measures re inability to comply	Will take measures to address?	
						P2-S1-6 or P2-S1-7b or P2-S1A-9 =		**P2-S4-18 or P2B-S4A-18 =**	**P2-S4-33 or P2B-S4A-33 =**	**P2-S5-1 or P2B-S5A-1 =**	**P2-S5-3 or P2B-S5A-3 =**	**P2-S5-6 or P2B-S5A-6 =**		**P2-S5-11 or P2B-S5A-11 =**	**P2-S5-14 or P2B-S5A-14 =**	**P2-S5-16 or P2B-S5A-16- =**		**P2-S5-31aa or P2B-S5A-31aa =**	**P2-S5-31a or P2B-S5A-31a =**	**P2-S5-31c or P2B-S5A-31c =**	
	A	**B**	**C**	**D**	**E**	**Yes**	**No[1]**	**No**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**No**
	X					X			X					X							
	X					X				X				X							
	X					X							X	X							
	X					X		X									X				
	X					X			X								X				
	X					X				X							X				
	X					X							X				X				
		X				[X]												X				
		X				[X]																X
Investigation incomplete			X				X															
	X						X															
		X					X															
					X		X															
Remediation incomplete					X	X					X											
					X	X						X										
	X					X					X			X								
	X					X						X		X								
	X					X					X							X				
	X					X						X										

Reportable situation event status	Type of reportable situation					Investigation complete?		No impact on clients	No client loss	Clients compensated	Started compensating	Intends to compensate?		Breach rectified	Has plan for rectifying	Preparing plan?		Addressed inability to comply	Taking measures re inability to comply	Will take measures to address?	
						P2-S1-6 or P2-S1-7b or P2-S1A-9 =		P2-S4-18 or P2B-S4A-18 =	P2-S4-33 or P2B-S4A-33 =	P2-S5-1 or P2B-S5A-1 =	P2-S5-3 or P2B-S5A-3 =	P2-S5-6 or P2B-S5A-6 =		P2-S5-11 or P2B-S5A-11 =	P2-S5-14 or P2B-S5A-14 =	P2-S5-16 or P2B-S5A-16- =		P2-S5-31aa or P2B-S5A-31aa =	P2-S5-31a or P2B-S5A-31a =	P2-S5-31c or P2B-S5A-31c =	
	A	B	C	D	E	Yes	No[1]	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Rectification incomplete	X					X		X							X						
	X					X			X						X						
	X					X				X					X						
	X					X							X		X						
	X					X		X								X					
	X					X			X							X					
	X					X				X						X					
	X					X							X			X					
		X				[X]													X		
		X				[X]														X	
Remediation and rectification incomplete	X					X					X				X						
	X					X						X			X						
	X					X					X					X					
	X					X						X				X					
		X			X	[X]					X								X		
		X			X	[X]						X							X		
		X			X	[X]					X									X	
		X			X	[X]						X								X	

Appendix K – Modifications V1.1 to Reportable Situation (YAML)

Attached list of changes to the reportable situation (YAML) file listed under Appendix A.

Swagger%20changes_28092022.xlsx

Appendix L – Modifications V1.2 to Reportable Situation (YAML)

Attached list of V1.2 changes to the reportable situation (YAML) file listed under Appendix A.

Swagger%20changes_19102022_V1.2.xlsx

Appendix M – Modifications V2.0 to Reportable Situation (YAML)

Attached list of V2.0 changes to the reportable situation (YAML) file listed under Appendix A.

SwaggerChanges_05032026_v2.0.xlsx

Appendix N – Inflight Submissions (CSV)

Inflight_Submission_Data.csv