NICE Network Protocol Specification

Version 1.1

Scenera, Inc., Nikon Corporation, Sony Semiconductor Solutions Corporation, Wistron Corporation and Hon Hai Precision Industry Co., Ltd.(NICE Alliance Promoters) contributed to this document.

Edit History

Version	Date	Comments
1.1	26 Jan 2022	Initial Version 1.1 release

Table of Contents

1. Scope **5**

2. Overview **5**

3. Destination ID **5**

4. Management and Control Interface Message Format **5**

4.1. CMF (Common Message Format) *5*

4.1.1. CMFContainer Object 6

4.1.2. CMFRequest Object 6

4.1.3. CMFResponse Object 7

4.1.4. EncryptedPayload Object 8

5. Data Interface Message Format **8**

6. Transport Layer Security (TLS) **8**

6.1. Protocol Version *8*

6.2. Cipher Suites *9*

6.2.1. Key Exchange 9

6.2.2. Authentication 9

6.2.3. Encryption 9

7. WebAPI **10**

7.1. Protocol *10*

7.2. Version *10*

7.3. URL *10*

7.4. Redirection *10*

7.5. Management and Control Interface API *10*

7.6. Data Interface API *11*

7.7. HTTP Status Messages and Codes *11*

1. Scope

This document provides the network protocols utilized commonly in the NICE System except backend process. It also describes the message structure to transfer data between Entities in NICE.

2. Overview

The Entities in NICE System have the three kinds of Interfaces - **Management**, **Control** and **Data**. This document defines the following to use the Interfaces.

- Network protocols
- Mechanism to identify the Entities
- Message format to exchange data between Entities over the Interface.

The Management and Control Interfaces utilize the Common Message Format (CMF) to wrap the objects carried over the interface.

The Data Interface uses JSON directly over HTTPS.

3. Destination ID

- [VERSION]
  - NICE Specification version.
- [EndPointID]
  - Unique identifier of the destination Entity, which is assigned by NICE System. Format is defined in NICE Identifier Structure specification.
- [NODE_ID]
  - Unique identifier of the Node in the Entity, which is assigned by the Entity itself. Format is defined in NICE Identifier Structure specification.
- [PORT_ID]
  - Identifier to distinguish the stream data. It is equivalent to PortID of Node. Format is defined in NICE Identifier Structure specification.
- [API_NAME]
  - API name which Entity supports.

Interface	DestinationID
Management	/[VERSION]/[EndPointID]/**management**/[API_NAME]
Control	/[VERSION]/[EndPointID]/**control**/[NODE_ID]/[API_NAME]
Data	/[VERSION]/[EndPointID]/**data**/[NODE_ID]/[PORT_ID]/[API_NAME]

4. Management and Control Interface Message Format

4.1. CMF (Common Message Format)

Data transferred over Management and Control Interface shall be encapsulated into common message format.

4.1.1. CMFContainer Object

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "CMF Object",
    "properties": {
        "SignedCMF": {
            "type": "string",
            "description": "JWS Compact Serialization format string of a CMFRequest or
CMFResponse Object."
        }
    }
}
```

4.1.2. CMFRequest Object

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "CMFRequest Object",
    "description": "CMF(Common Message Format) Request Message",
    "properties": {
        "Version": {
            "type": "string",
            "enum": [
                "1.0"
            ]
        },
        "MessageType": {
            "type": "string",
            "enum": [
                "request"
            ]
        },
        "SourceEndPointID": {
            "type": "string",
            "description": "EndPointID of the API caller."
        },
        "DestinationEndPointID": {
            "type": "string",
            "description": "EndPointID of the API callee."
        },
        "DateTimeStamp": {
            "type": "string",
            "description": "UTC date time sent the request. Format is defined in NICE
Date Time Format specification."
        },
        "CommandType": {
            "type": "string",
            "description": "DestinationID defined in Network Protocol Specification.",
            "maxLength": 128
        },
        "Payload": {
            "type": "string",
```

```
            "description": "If there is no data then this property shall not be
presented. Otherwise, JWE Compact Serialization format string of EncryptedPayload
Object shall be set."
        }
    },
    "required": [
        "Version",
        "MessageType",
        "SourceEndPointID",
        "DestinationEndPointID",
        "CommandType",
        "DateTimeStamp"
    ]
}
```

4.1.3. CMFResponse Object

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "CMFResponse Object",
    "description": "CMF(Common Message Format) Response Message",
    "properties": {
        "Version": {
            "type": "string",
            "enum": [
                "1.0"
            ]
        },
        "MessageType": {
            "type": "string",
            "enum": [
                "response"
            ]
        },
        "SourceEndPointID": {
            "type": "string",
            "description": "Same value of DestinationEndPointID as in the CMFRequest
Object."
        },
        "DestinationEndPointID": {
            "type": "string",
            "description": "Same value of SourceEndPointID as in the CMFRequest
Object."
        },
        "DateTimeStamp": {
            "type": "string",
            "description": "UTC date time sent the Response. Format is defined in NICE
Date Time Format specification."
        },
        "ReplyStatusCode": {
            "type": "integer",
            "description": "0 if no error. The other error codes are defined per API."
        },
        "ReplyStatusMessage": {
            "type": "string",
            "maxLength": 256
        },
        "Payload": {
            "type": "string",
```

```
            "description": "If there is no data then this property shall not be
presented. Otherwise, JWE Compact Serialization format string of EncryptedPayload
Object shall be set."
        }
    },
    "required": [
        "Version",
        "MessageType",
        "SourceEndPointID",
        "DestinationEndPointID",
        "DateTimeStamp",
        "ReplyStatusCode"
    ]
}
```

4.1.4. EncryptedPayload Object

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "EncryptedPayload Object",
    "description": "Application Access Token and NICE object.",
    "properties": {
        "AccessToken": {
            "type": "string",
            "description": "This is an Application Access Token. It shall be set if
this object for the CMFRequest Object. Otherwise this property shall not be
presented."
        },
        "PayloadObject": {
            "type": "object",
            "description": "An arbitrary NICE object. If no object then this property
shall not be included."
        }
    }
}
```

5. Data Interface Message Format

Data Objects used in API calls on the Data Interface are inserted as JSON objects into the HTTP Request Body.

6. Transport Layer Security (TLS)

In order to achieve secure network transport the entities in NICE Eco System must support TLS as the underlying protocol of HTTP.

6.1. Protocol Version

Protocol	Version	Support Level
SSL	1.0, 2.0, 3.0	Prohibited

TLS	1.0, 1.1	Prohibited
TLS	1.2	Mandatory
TLS	1.3 or later	Optional

6.2. Cipher Suites

TLS server shall support at least one cipher suite in the table.

Cipher Suite Name	Key Exchange	Authentication	Encryption	MAC	PRF	Support Level
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE	ECDSA	AES-256-GCM	N/A	SHA384	Mandatory
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE	RSA	AES-256-GCM	N/A	SHA384	Mandatory
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE	ECDSA	AES-128-GCM	N/A	SHA256	Mandatory
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE	RSA	AES-128-GCM	N/A	SHA256	Mandatory
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	DHE	RSA	AES-256-GCM	N/A	SHA384	Mandatory

6.2.1. Key Exchange

PFS(Perfect Forward Security) is mandatory.

Less than 2048-bit of DH parameter must not use in case of DHE.

6.2.2. Authentication

Server authentication is mandatory.

6.2.3. Encryption

128-bit or better security level and AEAD(Authenticated Encryption with Associated Data) must be supported. AES GCM is mandatory.

7. WebAPI

7.1. Protocol

WebAPI server and client shall support HTTPS.

7.2. Version

Version	Support Level
HTTP/1.0	Prohibited
HTTP/1.1	Mandatory
HTTP/2	Optional

7.3. URL

URL Format	https://[Authority][DestinationID]	[Authority] shall be compliant with definition in RFC3986.

Interface	Example
Management	https://la.nicealliance.com/1.0/00000002-5cdd-280b-8002-000000000000/management/GetManagementObject
Control	https://as.nicealliance.com/1.0/00000002-5cdd-280b-8002-000000000000/control/0001/GetSceneMode
Data	https://ds.nicealliance.com/1.0/00000002-5cdd-280b-8002-000000000000/data/0001/0002/SetSceneData

7.4. Redirection

If HTTP response returned with the Status Code 3xx, then the HTTP client shall perform HTTP redirection according to the **Location:** in the response header. The maximum number of redirection is 5. If exceeded it, then HTTP client shall stop the crawling.

7.5. Management and Control Interface API

HTTP Method		POST	
HTTP Request	Header	Content-Type: "application/json"	Mandatory
		Authorization: *Network AccessToken*	Optional
	Body	CMFContainer Object containing CMFRequest Object shall be inserted as JSON string.	Mandatory
	Header	Content-Type: "application/json"	Mandatory

HTTP Response	Body	CMFContainer Object containing CMFResponse Object shall be inserted as JSON string.	Mandatory
API Status		ReplyStatusCode and ReplyStatusMessage in CMFResponse Object.	

7.6. Data Interface API

HTTP Method		POST	
HTTP Request	Header	Content-Type: "application/json"	Mandatory
		Authorization: *Network AccessToken*	Optional
	Body	SceneMark or DataSection Object is inserted as JSON string.	Mandatory
HTTP Response	Header	N/A	
	Body	N/A	
API Status		HTTP Status Message and Codes.	

7.7. HTTP Status Messages and Codes

Status Code	Message	Description
200	OK	Request successful
307	Temporary Redirect	Temporary redirection of resource.
308	Permanent Redirect	Permanent redirection of resource.
400	Bad Request	The request is malformed, such as HTTP body format error.
401	Unauthorized	Authentication failed.
403	Forbidden	Authentication succeeded but the client doesn't have permission to the request resource.
404	Not Found	When a non-existent resource is requested.
405	Method Not Acceptable	The error for an unexpected HTTP method.
406	Unacceptable	The client presented a content type in the Accept header which is not supported.
413	Payload too large	The request size exceeded the given limit.
415	Unsupported Media Type	The requested content type is not supported.
429	Too Many Requests	The error is used when there may be DOS attack detected or the request is rejected due to rate limiting.
500	Internal Server Error	An unexpected condition prevented the server from fulfilling the request.

501	Not Implemented	The requested operation is not implemented.
503	Service Unavailable	Temporarily unable to process the request.