NICE Device 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

This document describes the implementation of a NICE compliant Device. This includes how the device is managed and configured.

2. Device Overview

A NICE Device shall contain at least one Node. A Node is a basic unit of the Data Pipeline contains inputs or outputs, a sensor or actuator and a computer vision process. The Device shall manage the following functions on behalf the Nodes that it contains:

1. Network Connectivity
2. Security of access to the Device, the protection of connections to the Device.
3. Privacy of data that is generated by Nodes within the Device.

Each Device shall have a DeviceID which is a unique identifier for the Device. Each Node in the Device has a unique ID. The format of this ID and its relationship with the DeviceID is defined in the "NICE Identifier Structure" document. The Device shall validate Entities that are communicating with the Device and provide access to resources on the Device based on the Permission that has been provided by the Entity in communication with the Device. The Access Management for Devices is defined in the NICE Authentication Specification.

A Device shall be managed by one NICE Account Service at a time. The NICE License Authority is responsible for ensuring that the Device is allocated to a specific NICE Account Service. When adding a device to an account, the user is redirected from the BSS to the NICE LA, which requires the user to enter the Device ID and Password, which the NICE LA checks the correctness of and whether the Device is not already linked to an account. If these conditions are met then the NICE LA will issue a Device Management Object to the Device, which links the Device to the NICE AS.

The NICE Account Service enables Apps and Data Services to interact with the Device and to configure the Nodes that are housed in the Device using the DataPipelineController.

A Device shall have:

- One or more Nodes implemented.
- A Single management Interface.
- At least one IP based connection for Management, Control and Data.
- A Unique DeviceID, Private Key with an accompanying X.509 certificate available on the NICE Licensing Authority.

A Device shall maintain trusted time. See the section Device Trusted Time in NICE Authentication Specification for detail.

2.1. Interfaces

2.1.1. Management Interface

A **Management Interface** of a Device shall be used to configure the interconnections between Devices and Cloud Services and set up the security and privacy objects.
The interface shall be used for:

- Setting the Control and Data Protocols to be used for each connection.
- Setting the security credentials to enable secure communication between Devices, Cloud Services and App.

2.1.2. Control Interface

The **Control Interface** of a Node shall enable the DataPipelineController to manage the Node within a Device. The Node polls the DataPipelineController to fetch the SceneMode for the Node.

The Entities managing the Data Pipeline shall be provided credentials by the NICE Account Service that enable the Entity to communicate with the Device. The Entity shall use the API defined in the Data Pipeline Specification to manage the configuration of the Nodes on the Device.

2.1.3. Data Interface

The **Data Interface** of the Device enables the Device to exchange data with a DataPipeline.

The establishment of the Data Session shall be set by the SceneMode as defined in the Data Pipeline Specification.

3. Management Interface

3.1. GetManagementEndPoint

Function

The "GetManagementEndPoint" requests the ManagementEndPoint Object from the NICE LA.

Protocol(s) Used to Make Calls

WebAPI

Direction

Caller	DEVICE
Callee	NICELA

Request Parameters

Empty

Application AccessToken is not necessary to set.

Acknowledgement Parameters

ManagementEndPoint Object

3.2. GetManagementObject

Function

The "GetManagementObject" requests the Management Object from the NICE LA.

The Object shall always be encrypted under the Device Public Key and shall be Signed with a Private Key that is certified by the NICE LA for the purpose of being used to sign these Objects. In case the object is not signed or encrypted it shall be rejected by the Device. If the signature validation fails, the Object shall be rejected by the Device. If the object is incorrectly formed it shall be rejected by the Device.

Protocol(s) Used to Make Calls

WebAPI

Direction

Caller	DEVICE
Callee	NICELA

Request Parameters

Empty

Acknowledgement Parameters

Management Object

3.3. GetControlObject

Function

The "GetControlObject" requests the Control Object from the NICE AS.

Protocol(s) Used to Make Calls

WebAPI

Direction

Caller	DEVICE
Callee	NICEAS

Request Parameters

Empty

Acknowledgement Parameters

DeviceControl Object

4. Data Objects

4.1. Management Object

The Management Object is provided to a Device when it is linked to a User's Account. The Device polls the NICE LA for the current Management Object. When the Device has been added to the account the Management Object is generated. It is also updated as new Access Tokens are required. The Object contains information that allows the Device to verify messages sent to the Device by the NICE AS or Entities that the NICE AS authorizes to communicate with the Device.

The "AllowedTLSRootCertificates" provides a list of allowed root certificates for TLS communication. Each entry in the array contains a Root Certificate that is valid for TLS communication. Certificates in this list are appended to any existing listed TLS Root Certificates that are stored in the device. The list is explicitly provided here so as not required to have the actual certificate signed by the NICE LA. There is also an option to delete Certificates that have been provided previously to the Device.

4.1.1. JSON Object

Management Object

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "Management",
    "description": "The device shall first decrypt this object using its private key.
It then extracts and validates the NICELA working X.509 certificate and validates this
certificate. It then uses the public key from this certificate and validates the
message. The validate includes the entire decrypted document structure. ",
    "properties": {
        "Version": {
            "type": "string",
            "enum": [
                "1.0"
            ]
        },
        "DeviceID": {
            "type": "string",
```

```
            "description": "Permanent ID of the device. Readable by end user."
        },
        "NICEAS": {
            "type": "object",
            "description": "Management layer messages can only be sent and received
using these credentials to encrypt at an application level.",
            "properties": {
                "NICEASID": {
                    "type": "string",
                    "description": "Unique ID for the NICE Account Service Provider"
                },
                "NICEASName": {
                    "type": "string"
                },
                "NICEASEndPoint": {
                    "$ref": "Definitions.json#/definitions/EndPoint"
                }
            },
            "required": [
                "NICEASEndPoint",
                "NICEASID"
            ]
        },
        "AllowedTLSRootCertificates": {
            "$ref": "Definitions.json#/definitions/x5c"
        },
        "DeviceCertificate": {
            "$ref": "Definitions.json#/definitions/x5c"
        }
    },
    "required": [
        "NICEAS",
        "AllowedTLSRootCertificates",
        "DeviceID",
        "Version"
    ]
}
```

4.2. DeviceControl Object

This Object is provided by the NICE AS to the Device to manage the processing of Access Tokens that the NICE AS provides to Entities that are enabled to interact with the Device.

4.2.1. JSON Object

DeviceControl

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "DeviceControl",
    "description": "This object sets up the permissions and end points for the Control
communication.",
    "properties": {
        "Version": {
            "type": "string",
            "enum": [
                "1.0"
            ]
```

```
        },
        "DeviceID": {
            "type": "string"
        },
        "ControlEndPoints": {
            "type": "array",
            "uniqueItems": true,
            "items": {
                "$ref": "Definitions.json#/definitions/EndPoint"
            }
        },
        "AllowedTLSRootCertificates": {
            "$ref": "Definitions.json#/definitions/x5c"
        }
    },
    "required": [
        "ControlEndPoints",
        "DeviceID",
        "Version"
    ]
}
```

4.3. ManagementEndPoint Object

The ManagementEndPoint is provided by the NICE LA to the Device to indicate where it should get Management Object.

ManagementEndPoint

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "type": "object",
    "title": "ManagementEndPoint",
    "properties": {
        "Version": {
            "type": "string",
            "enum": [
                "1.0"
            ]
        },
        "NICELAEndPoint": {
            "$ref": "Definitions.json#/definitions/EndPoint"
        },
        "DeviceCertificate": {
            "$ref": "Definitions.json#/definitions/x5c"
        }
    },
    "required": [
        "Version",
        "NICELAEndPoint",
        "DeviceCertificate"
    ]
}
```