Device Implementation Guide

Version 1.0

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

Revision History

Version	Date	Comments
0.9rc1	13 Nov 2018	First draft
0.9rc2	25 Feb 2019	Second draft
0.9	25 Mar 2019	Final draft
1.0	22 May 2019	Final release

Contributors

Name	**Company**
Andrew Wajs	Scenera
Aviram Cohen	Scenera
Munehiro Shimomura	Sony
Hironori Miyoshi	Sony
Wendy Tin	Wistron

Table of Contents

1. Scope

This document describes the use cases for the Device and the guidelines for the security implementation of the Device.

2. Overview

This specification describes the life cycle for the Device from its manufacture to its usage by a User.

3. Use Cases

The Device undergoes the following events in its life cycle:

1. Manufacture. During the Device's manufacture the DeviceID and the Device Private Key shall be inserted into the Device. The DeviceID and Device Password shall be accessible to the User. The DeviceID and Device Password shall be used by the User to link the Device to the NICE Account Service. The Private Key is used to decrypt Access Tokens and data Objects that are sent to the device in an encrypted form and to sign objects that are sent from the Device.
2. First Setup.
    1. The Device is installed by the User and makes its first connection to the Internet.
    2. The Device is assigned to the User's account.
    3. The Device connects to Services and NICE Apps.
3. Power Reset. The Device is restarted.
4. The Device is configured and used for streaming of SceneMarks and SceneData or streaming of live data.
5. The Device is unlinked from a User's account and linked to a different User's account.
6. The Device has its firmware updated.

3.1. First Setup

3.1.1. Network Setup

3.1.1.1. Local Area Network Connection

When User sets up the device for the first time, it shall be connected to the Internet. The Device may be connected to the Internet via a local network. The method that is used to make the local connection is determined by the Device Manufacturer. Methods that may be used are:

1. The Device capturing an image that includes the SSID and Password for the local network either as text or as a QR code.
2. A Bluetooth or WiFi connection to an App running on a PC or Mobile Device. The App configures the SSID and Password using this connection.

Other methods are also permitted.

3.1.1.2. Internet Connection

When the Device has access to the Internet it shall establish a connection to the NICE_LA server using by MQTT and establish the MQTT Management Session. This session will always be kept.

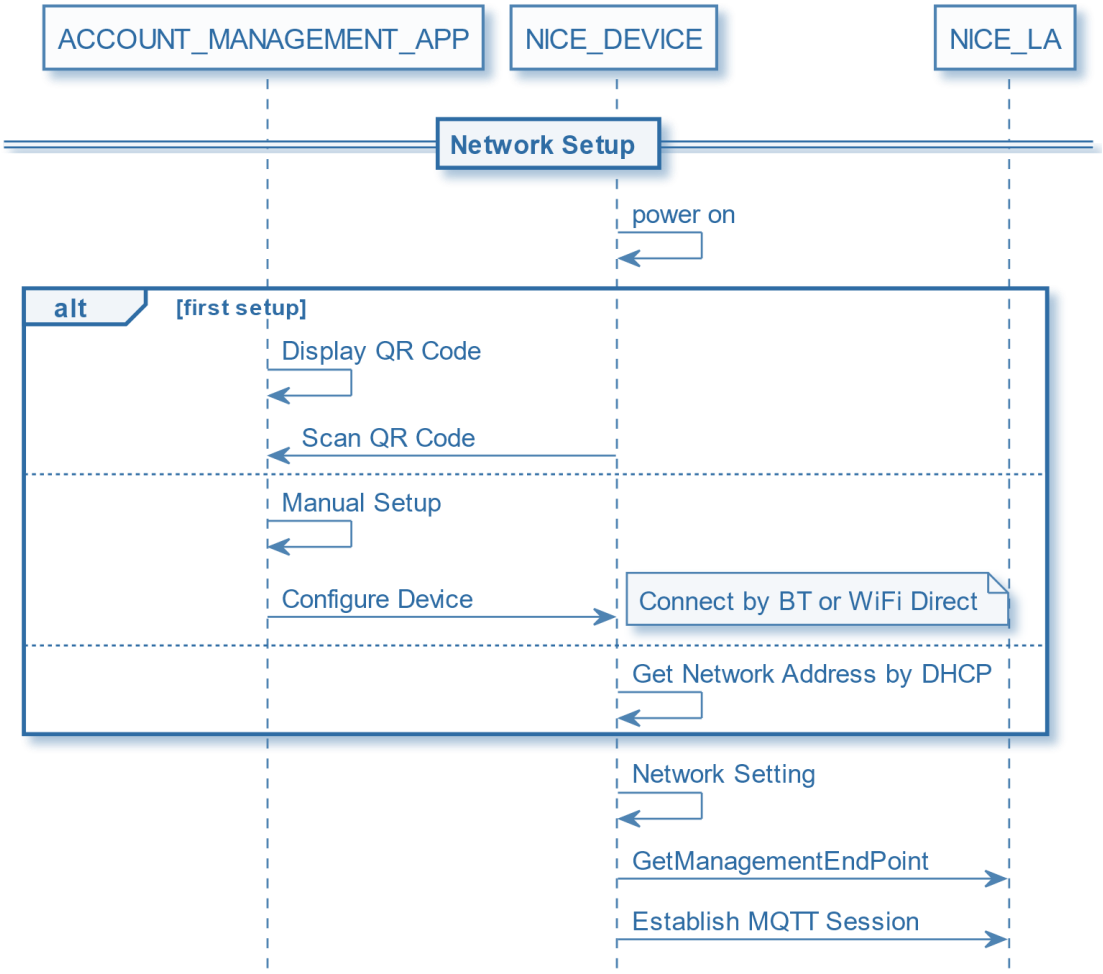

Figure 1. Establishing a first connection to the NICE License Authority

3.1.2. Assign Device to User Account

When the device is manufactured it is provisioned with device credentials including a DeviceID and a Password. These shall be accessible to the User. This may be through one of the following:

1. Display of the DeviceID and Password in a human readable form or readable by the NICE Account Management App.
2. Display of the DeviceID and Password in the form of a QR Code accessible to the User or the NICE Account Management App.
3. Accessible through a local connection with the NICE Account Management.

The User shall have an account with a NICE Account Service. The User shall initiate the linkage of the Device to their Account on the NICE Account Service. The NICE Account Management App for the NICE Account Service shall facilitate this assignment. The NICE Account Management App shall facilitate the entry of the DeviceID and Password with the NICE LA to obtain permission to link the Device to the Account on the NICE Account Service. The NICE Account Management App may use one of the following methods to facilitate the entry of this data:

1. Enable manual entry through an OAuth session with the NICE License Authority. The User reads the DeviceID and Password on the Device and enters it in an OAuth session with the NICE LA.
2. The NICE Account Management App scans the QR Code containing the DeviceID and Password.
3. The NICE Account Management App connects to the Device through a Bluetooth or some other local connection to read the Device ID and Password.

Once the Device has been associated with a User's account it shall only be released from the account if the User initiates the release from their Account. A subsequent entry of the DeviceID and Password will not be accepted unless the User has actively released the Device from their Account by logging into their Account and releasing the Device.

Once this link has been made in the back end, the NICE LA shall send the Device the Management Object. This shall configure the Device to be linked to the User's account on the NICE Account Service. The NICE Account Service shall send the Control Object to the Device to configure the device.

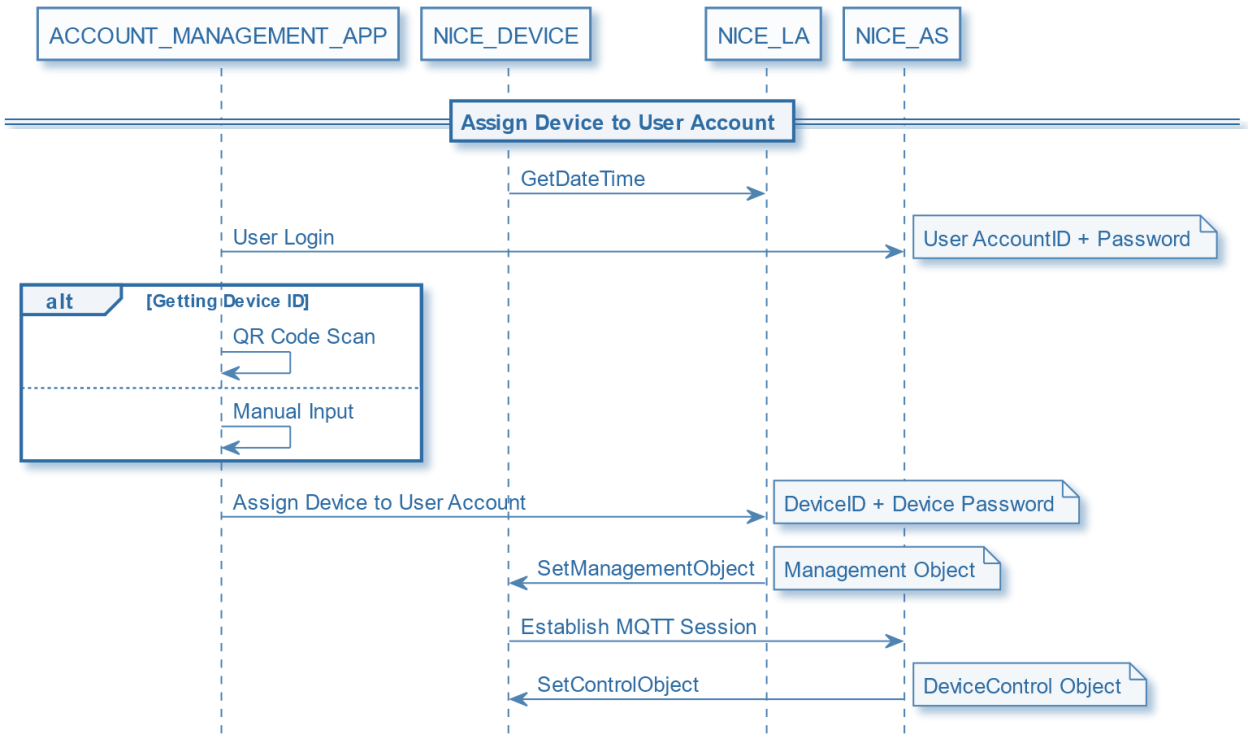

Figure 2. Assign a Device to a User's Account

After the Device is assigned to a User's account, the device shall connect to the NICE Account Service through an MQTT session. This session shall be permanently maintained.

After this session is established, the device shall become available for use. This connection shall be used by the NICE Account Service to deliver Control Objects to update connections that the Device may establish with other Entities.

3.2. Power Reset

When power is turned on, DEVICE shall perform the sequence of commands shown in Figure 3. The Device shall first get the Management Object from the NICE LA and then connect with the NICE AS as shown.

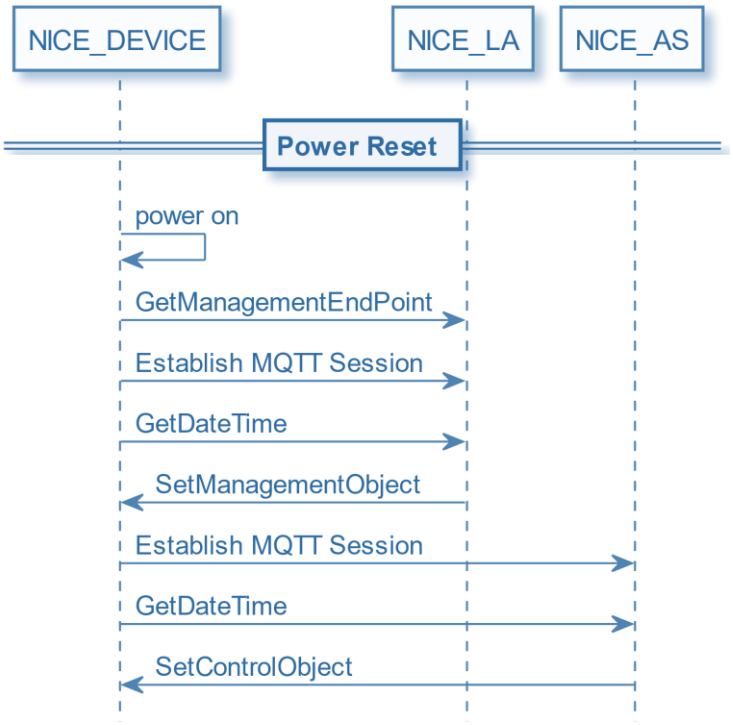

Figure 3. Power Reset Operation

3.3. Device Configuration

SetSceneMode shall be used for device control related settings such as SceneMode, Codec Parameter, Sensor Parameter etc.

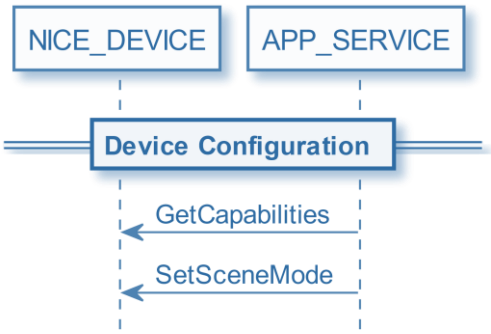

Figure 4. Device Configuration

3.4. Scene Mark and Data Streaming

When a Trigger condition as defined in the SceneMode is detected, the Device shall send a SceneMark, SceneData as defined in the SceneMode.

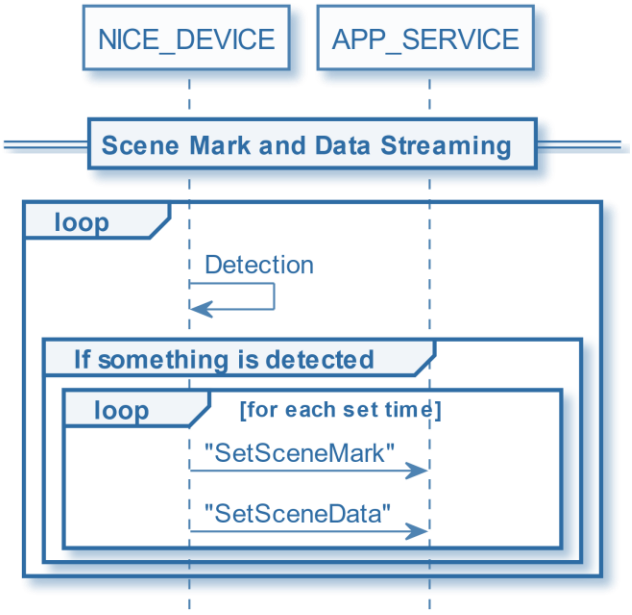

Figure 5 Publishing SceneMarks and SceneData

3.5. Live Streaming

The Device shall be capable of transmitting the current information (Video, Audio etc.) by Live Streaming over WebRTC.

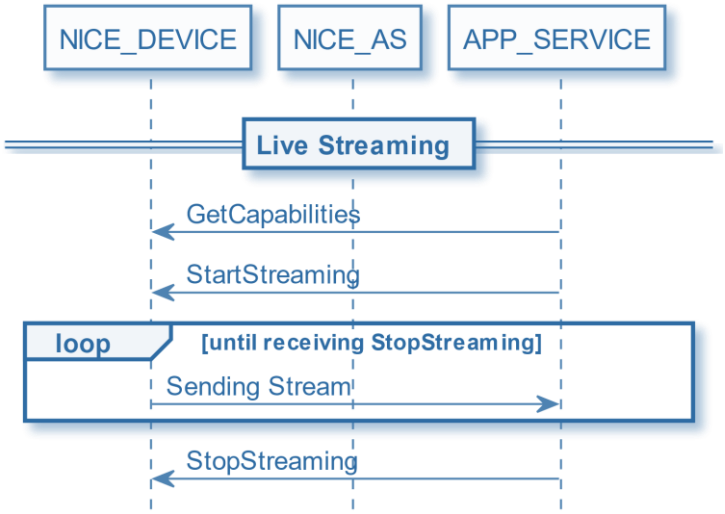

Figure 6. Live Streaming

3.6. Release Device

When the Device is in use, it shall be associated with owner's User Account. The Device may be passed to another owner. The first owner shall select the "release device" setting on their NICE account. This shall result in the Management Object being transferred to the Device and reset the Device back into its unassigned state. The Device can then be transferred to a different owner's User Account using the "Assign a Device to a User's Account" use case.

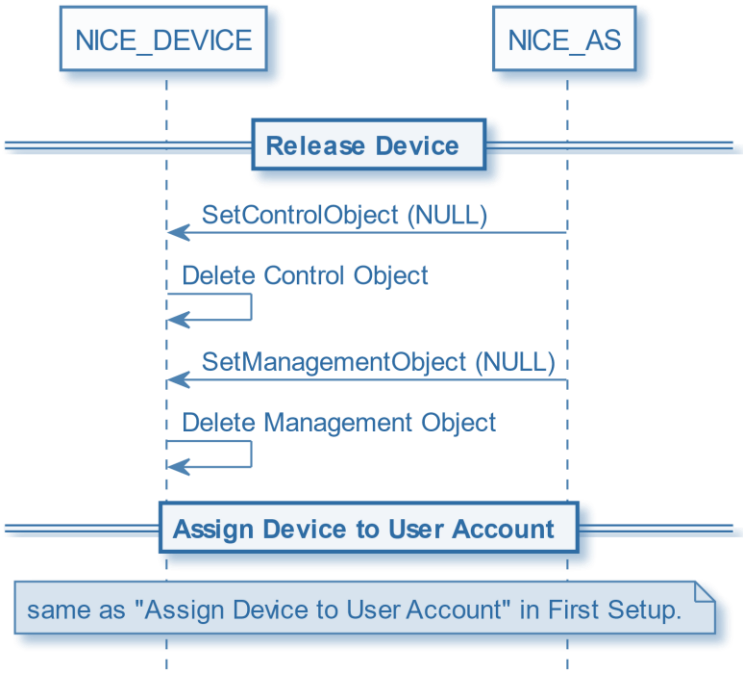

Figure 7. Release Device

3.7. Firmware Update

The Device shall be capable of updating its firmware. The Device shall receive a FirmwareUpdate Object. This may be pushed to the Device by either the Device Manufacturer, Device Seller or the NICE Account Service. The Device may also fetch the FirmwareUpdate Object. The Device fetch Firmware image from the URI in the FirmwareUpdate object. The Device shall use the hash in the FirmwareUpdate Object to validate that the correct Firmware that has been delivered to the Device before performing the update.

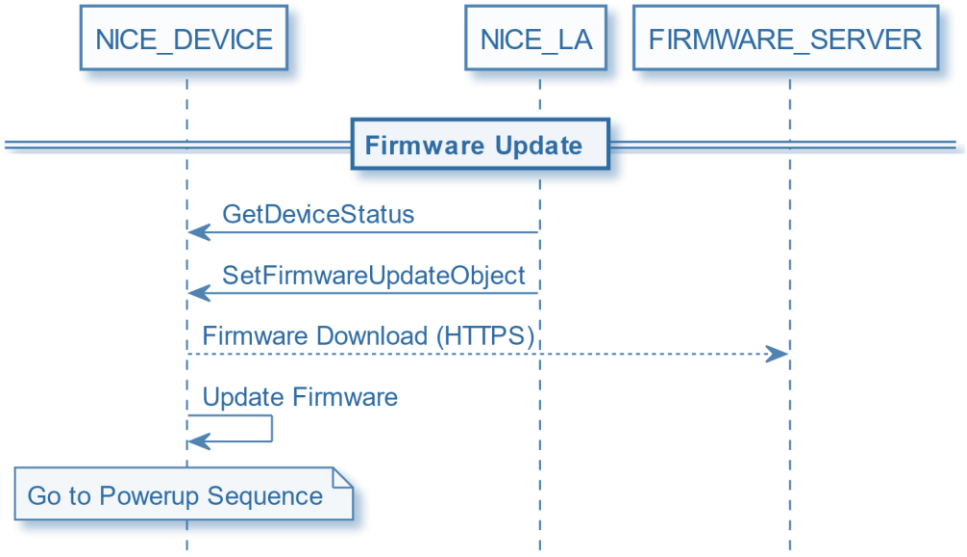

Figure 8. Firmware Update

4. Caching of Objects

The Device shall Cache Privacy Objects.

5. Security Requirements

The Device shall implement the following security features:

1. Secure Firmware Update as defined in the Device Specification.
2. Secure communication as defined in the Network Protocols Specification.
3. Secure processing of Access Tokens as defined in the Authentication Specification.
4. Secure processing of SceneData and SceneMarks as defined in the Security and Privacy Specification.
5. Trusted Time as defined in the Authentication Specification.

5.1. Security Levels

There are two permissible security levels for the implementation of the security functions outlined above.

5.1.1. Security Level 0 - Application Processor Only

The following functions are implemented on the Application Processor of the Device without any hardware separation of the security functions from the rest of the application processor software:

1. Secure Firmware Update
2. Secure communication
3. Secure processing of Access Tokens
4. Secure processing of SceneData and SceneMarks.
5. Trusted Time.

The random number generator function shall conform to that defined in SP800-90.

5.1.2. Security Level 1 - Trusted Execution Environment

A Trusted Execution Environment shall have the following:

1. Isolated execution environment. Software executing in the Trusted Execution Environment shall be isolated from any software executing on any other environment within the System on Chip including the Application Processor. Any software that is executed within this environment shall be verified before it is loaded into the environment. The software executing within the environment shall also remain confidential. I.e. the software shall not be exposed in unencrypted form outside of the Trusted Execution Environment.
2. Secure Storage. Data that is stored in Secure Storage shall not be accessible in unencrypted form by any process or software that is operating outside of the Trusted Execution Environment.
3. Random Number generation. The Trusted Execution Environment shall have a random number generation that is based on a physical source of randomness.
4. An AES engine may be optionally be included in the Trusted Execution Environment. The Application processor may have access to have data either encrypted or decrypted but shall not be capable of extracting Content Encryption Key from the AES engine.

The following data shall be secured within the Trusted Execution Environment:

1. Device Private Key
2. NICE License Authority Public Key.
3. Private Keys used for the TLS and DTLS communication.
4. Content Encryption Keys that are used to encrypt or decrypt data.
5. Time Clock. The Device shall use the Trusted Time Protocol to initiate the state of the Time Clock and shall maintain this Time Clock in the Trusted Execution Environment.
6. Manufacturer ID, Model Type, Seller ID, Current Software Version Number, Issue Date, MinimumalAcceptableVersion, OldestValidDate fields from the last accepted FirmwareUpdate Object. When first manufactured, these fields are set together with Device Private Key, DeviceID fields.
7. Tokens which have been revoked via the Management or Control Objects.

The following operations shall be performed in the Trusted Execution Environment:

1. Decryption of Objects that are encrypted with the Device Private Key.
2. Encryption or Decryption of Data Objects that is protected with a Content Encryption Key.
3. Authentication of Objects that are signed by keys that are certified by the NICE LA or NICE AS.
4. Signing of Objects using the Device Private Key.
5. Validation of Access Tokens that are used to access the device.
6. Validation of TimeStamps.

5.2. Implementation Guidelines

There shall be a Security Application which shall perform the security functions described below. The implementation of this Application is determined by the Security Level of the Device.

5.2.1. Secure Firmware Update

The FirmwareUpdate Object shall be received and processed within the Security Application.

The Object shall first be decrypted with the Device Private Key. The contents of the Object are validated by checking the signature of the Object using a Key that is certified by the NICE LA and that corresponds to the FirmwareSourceID. The following fields must match stored the stored values for Firmware Management:

1. ManufacturerID
2. ModelType
3. FirmwareSourceID

The following fields must be valid when compared with the following fields that are present in the Firmware Management Data structure:

1. OldestValidDate - the date of the update must be later than this date.
2. ValidDate - the date of the update must be before this date.

If these parameters pass then the HashValue field in the Firmware Update object may be checked against the hash value for the code image downloaded. The process file containing the update image shall be passed through the Security Application to enable the validation of the Hash value contained in the FirmwareUpdate Object.

If the hash value check passes, the following fields are updated using the new values that have been carried in the FirmwareUpdate Object:

1. MinimumAcceptableVersion - the version number for this Update must be greater than this version number.
2. OldestValidDate - the date of the update must be later than this date.
3. ValidDate - the date of the update must be before this date.

These values shall be used to check the next FirmwareUpdate Object that is submitted to the Device.

5.2.2. Secure Communication

All communication to the Device shall be protected using either TLS or DTLS. The messages that are used to set up the session key for a TLS session shall be processed within the secure execution environment, ensuring that the Private Key for the device does not leave the Security Application. The session key shall be transferred outside of the Security Application to enable the encryption or decryption of the data traffic to and from the Device.

5.2.3. Processing of Access Tokens

The Access Token for an Entity that is accessing the Device shall be submitted to the Security Application.

The Access Token shall be decrypted using the Device Private Key and authenticated using a Key that is certified by the NICE Account Service or NICE License Authority.

The Access Token fields shall be validated within the Security Application. The following fields in the decrypted token shall be validated:

1. "iss" the issuer field shall be either that corresponding to the NICE AS or NICE LA.
2. "sub" shall match the DeviceID for the device receiving the device.
3. "aud" shall match the Device or ApplicationID that is making the request for access to the device.
4. "exp" the date beyond which the token shall be rejected.
5. "nbf" the date before which the token shall be rejected.

The Security Application shall return whether the token has passed the authentication, format and field checks as well as the "Permission" field that defines the permissions that are allowed by the Token.

5.2.4. Secure SceneData and SceneMarks

The processing of Privacy Objects shall be performed by the Security Application. The Object shall be decrypted using the Device Private Key and authenticated using a Key that is certified by the NICE AS. The following parameters shall be verified before the object is further processed:

1. DeviceID shall match that of the Device.
2. The current time within the security process shall be within the StartDate&Time and EndDate&Time window.

The key that is carried in the Privacy Object shall be transferred to the process or hardware that is responsible for encrypting or decrypting data.

In addition to providing the key to the appropriate encryption or decryption function, the Security Application shall provide the usage rules to the processor that is handling the data.

5.2.5. Trusted Time

On power up of the Device, the Security Application shall use the random number generator to generate a TrustedTimeRequest object. The Security Application shall retain the random challenge and validate the TrustedTimeResponse object that is received from either the NICE LA or NICE AS time service.

Once the Security Application has validated the response from the NICE LA or NICE AS, the Security Application shall initiate the secure clock within the Security Application.

The requirement for accuracy of this clock is that the offset from the clock of either the NICE LA or NICE AS shall be no more than 1 second.

6. Minimum Encoding Requirements

Each NICE compliant shall implement the following at the minimum:

1. MPEG 4 Constrained Baseline Profile.
2. JPEG for still images.

7. Minimum Network Configuration Requirements

The Device shall support the following configuration:

1. Static IP address (netmask, default gateway).
2. Dynamic IP address (DHCP).
3. Address Port.
4. Proxy settings.
5. DNS settings.