OCF 2.3 – Firmware Update (SVR part) – Sec WG CR 2453

Legal Disclaimer

THIS IS A DRAFT SPECIFICATION DOCUMENT ONLY AND HAS NOT BEEN ADOPTED BY THE OPEN CONNECTIVITY FOUNDATION. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT OF THIS DRAFT DOCUMENT. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT DOCUMENT IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT DOCUMENT AT ANY TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT DOCUMENT IN ANY WAY, INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES. IMPLEMENTATION OF THIS DRAFT DOCUMENT IS DONE AT YOUR OWN RISK AMEND AND IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE. IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS TO THIS DRAFT DOCUMENT, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED HEREWITH INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL AS CLAIMS OF DETRIMENTAL RELIANCE.

Changes on top of v1.3.1

** Change 1: add bit for software check **

Table 55

Value	Device Mode	Description
bx0000,0001 (1)	Reset	Device reset mode enabling manufacturer reset operations
bx0000,0010 (2)	Take Owner	Device pairing mode enabling owner transfer operations
bx0000,0100 (4)	Not Applicable	
bx0000,1000 (8)	Security Management Services	Service provisioning mode enabling instantiation of Device security services and related credentials
bx0001,0000 (16)	Provision Credentials	Credential provisioning mode enabling instantiation of pairwise Device credentials using a management service of type urn:oic.sec.cms
bx0010,0000 (32)	Provision ACLs	ACL provisioning mode enabling instantiation of Device ACLs using a management service of type urn:oic.sec. ams
bx0100,0000 (64)	Initiate Software Version Validation	Software version validation requested/pending (1) Software version validation complete (0) Requires software download to verify integrity of software package
bx1000,0000 (128)	Initiate Secure Software Update	Secure software update requested/pending (1) Secure software update complete (0)
bx1 0000,0000 (256)	Initiate Software Availability Check	Checks if new software is available on remote endpoint. Does not require to download software. Methods used are out of bound.

Table 1 – Value Definition of the oic.sec.dpmtype Property (Low-Byte)

** Change 2: introduction section 14.4.2 **

Different manufacturers approach software update utilizing a collection of tools and strategies: over-the-air or wired USB connections, full or partial replacement of existing software, signed and verified code, attestation of the delivery package, verification of the source of the code, package structures for the software, etc.

It is recommended that manufacturers review their processes and technologies for compliance with industry best-practices that a thorough security review of these takes place and that periodic review continue after the initial architecture has been established.

This specification applies to software updates as recommended to be implemented by OCF Devices; it does not have any bearing on the above-mentioned alternative proprietary software update mechanisms. The described steps are being triggered by an OCF Client, the actual implementation of the steps and how the software package is downloaded and upgraded is vendor specific.

The triggers that can be invoked from OCF clients can perform:

1) Check if new software is available
2) Download and verify the integrity of the software package
3) Install the verified software package

The triggers are not sequenced, each trigger can be invoked individually.

The state of the transitions of firmware update is in figure XXX.

```
@startuml
[*] -down-> idle


idle -down-> new_version_check : pstat.tm@bit256 = 1 \n(Initiate Software Availability
Check)

idle --> upgrading : pstat.tm@bit128 = 1 \n (Initiate Secure Software Update)

idle : pstat.cm@bit256 = 0 (no new software version available)
idle : pstat.cm@bit64 = 0 (no valid software available)
idle : pstat.cm@bit128 = 0 (not upgraded)


new_version_check -up-> idle : pstat.cm@bit256 = 0 (no new version available)

new_version_check: pstat.cm@bit256 = 0 (New Software Available)
new_version_check: pstat.cm@bit64 = 0 (Valid Software Available)
new_version_check: pstat.cm@bit128 = 0 (upgrading)



new_version_check ->  version_validation : pstat.cm@bit256 = 1 (new version available)
idle ->  version_validation : pstat.tm@bit64 = 1 \n(Initiate Software Version Validation)

version_validation -> idle: pstat.cm@bit64 = 0 (no valid software available)
version_validation -> version_available: pstat.cm@bit64 = 1 (valid software available)
version_validation: pstat.cm@bit256 = 1 (New Software Available)
version_validation: pstat.cm@bit64 = 0 (Valid Software Available)
version_validation: pstat.cm@bit128 = 0 (upgrading)


version_available -> upgrading : pstat.tm@bit128 = 1 \n (Initiate Secure Software Update)
version_available: device will transition automatically
version_available: when validation check is finished



version_available : pstat.cm@bit256 = 1 (new version available)
version_available : pstat.cm@bit64 = 1 (valid version available)
version_available : pstat.cm@bit128 = 0 (not upgrading)
```

```
upgrading : dowloading and upgrading
upgrading : pstat.cm@bit256 = 1 (New Software Available)
upgrading : pstat.cm@bit64 = 1 (Valid Software Available)
upgrading : pstat.cm@bit128 = 1 (upgrading)

upgrading -> idle: pstat.cm@bit128 = 0 (upgrading finished)
@enduml
```

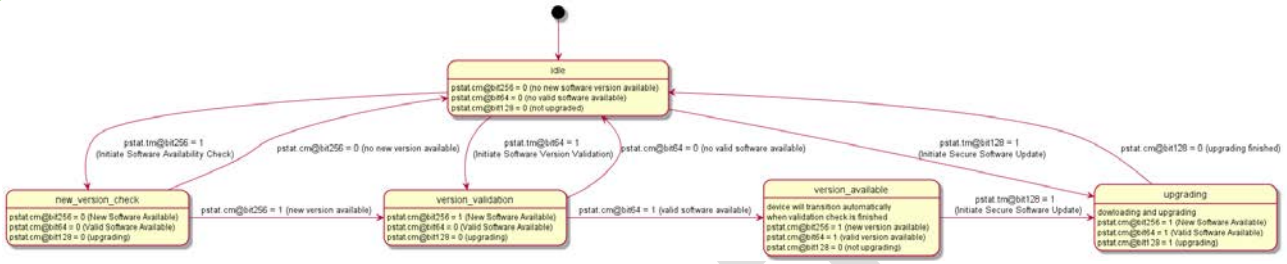

Figure 1. State transitioning diagram for firmware download

Table 2. Description of the software update bits

Bit	TM property	CM property
Bit 256	Initiate Software Availability Check	New Software Available
Bit 64	Initiate Software Version Validation	Valid Software Available
Bit 128	Initiate Secure Software Update	Upgrading

1.1.1.1 Checking availability of new firmware.

Setting the Initiate Software Availability Check bit in the /oic/sec/pstat.tm Property (see Table 52 of Section 13.7) indicates a request to initiate the process to check if new software is available, e.g. the process whereby the Device checks if a newer software version is available on the external endpoint. Once the Device has determined if an newer software version is available, it sets the Initiate Software Availability Check bit in the /oic/sec/pstat.cm Property to 1 (TRUE), indicating that new software is available or to 0 (FALSE) if no newer software version is available, See also table XXX where the bits in property TM indicates that the action is initiated and the CM bits are indicating the result of the action. Note that the Device receiving this trigger is not is not downloading and not validating the software to determine if new software is available. The version check is determined by the current software version and the software version on the external endpoint. The determination if a software package is newer is vendor defined.

1.1.2 Software Version Validation

Setting the Initiate Software Version Validation bit in the /oic/sec/pstat.tm Property (see Table 52 of Section 13.7) indicates a request to initiate the software version validation process, the process whereby the Device validates the software (including firmware, operating system, Device drivers, networking stack, etc.) against a trusted source to see if, at the conclusion of the check, the software update process will need to be triggered (see clause XXX). When the Initiate Software Version Validation bit of /oic/sec/pstat.tm is set to 1 (TRUE) by a sufficiently privileged Client, the Device sets the /oic/sec/pstat.cm Initiate Software Version Validation bit to 0 and initiates a software version check. Once the Device has determined if an valid software is available, it sets the Initiate Software

Version Validation bit in the /oic/sec/pstat.cm Property to 1 (TRUE) if an update is available or 0 (FALSE) if no update is available. To signal completion of the Software Version Validation process, the Device sets the Initiate Software Version Validation bit in the /oic/sec/pstat.tm Property back to 0 (FALSE).If the Initiate Software Version Validation bit of /oic/sec/pstat.tm is set to 0 (FALSE) by a Client, it has no effect on the validation process. Note that the Software Version Validation process can download the firmware from the external endpoint to verify the integrity of the software package

1.1.3 Software Update

Setting the Initiate Secure Software Update bit in the /oic/sec/pstat.tm Property (see Table 52 of Section 13.7) indicates a request to initiate the software update process. When the Initiate Secure Software Update bit of /oic/sec/pstat.tm is set to 1 (TRUE) by a sufficiently privileged Client, the Device sets the /oic/sec/pstat.cm Initiate Software Version Validation bit to 0 and initiates a software update process. Once the Device has completed the software update process, it sets the Initiate Secure Software Update bit in the /oic/sec/pstat.cm Property to 1 (TRUE) if/when the software was successfully updated or 0 (FALSE) if no update was performed. To signal completion of the Secure Software Update process, the Device sets the Initiate Secure Software Update bit in the /oic/sec/pstat.tm Property back to 0 (FALSE). If the Initiate Secure Software Update bit of /oic/sec/pstat.tm is set to 0 (FALSE) by a Client, it has no effect on the update process.

1.1.3.1 State of Device after software update.

The state of all resources implemented in the Device should be the same as after boot, meaning that the firmware update is not resetting user data and retaining a correct state.

User data of a Device is defined as:

- Retain the SVR states, e.g. the on boarded state, registered clients, etc..
- Retain all created resources
- Retain all stored data of a resource
    - For example the preferences stored for the brewing resource (oic.r.brewing).

1.1.4 Recommended Usage

The Initiate Secure Software Update bit of /oic/sec/pstat.tm should only be set by a Client after the Initiate Software Version Validation check is complete.

The process of updating Device software may involve state changes that affect the Device Operational State (/oic/sec/pstat.dos). Devices with an interest in the Device(s) being updated should monitor /oic/sec/pstat.dos and be prepared for pending software update(s) to affect Device state(s) prior to completion of the update.

Note that the Device itself may indicate that it is autonomously initiating a software version check/update or that a check/update is complete by setting the pstat.tm and pstat.cm Initiate Software Version Validation and Secure Software Update bits when starting or completing the version check or update process. As is the case with a Client-initiated update, Clients can be notified that an autonomous version check or software update is pending and/or complete by observing pstat resource changes.

Note that the oic.r.software update resource type specifies additional features to control the software update process see core specification.