CENTER for MEDICAL
INTEROPERABILITY

The Center for Medical Interoperability Specification
Automated Secure Update and Management Framework

CMI-SP-F-ASUM-D01-20190311

DRAFT

Notice

This document may contain references to other documents not owned or controlled by The Center. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document. To the extent this document contains or refers to documents of third parties, you agree to abide by the terms of any licenses associated with such third-party documents, including open source licenses, if any.

DISCLAIMER

This document is furnished on an "AS IS" basis and neither The Center nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and The Center and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

The Center reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by The Center or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from The Center, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CMI-SP-F-ASUM-D01-20190311			
Document Title:	Automated Secure Update and Management Framework			
Revision History:	D01			
Date:	March 11, 2019			
Status:	~~Work in Progress~~	**Draft**	~~Issued~~	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~The Center/Member~~	~~The Center/ Member/ NDA Vendor~~	**Public**

Key to Document Status Codes

Work in Progress An incomplete document designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through The Center.

Trademarks

Contents

Figures

Tables

1 Scope

1.1 Introduction and Purpose

This document establishes a foundational framework for Automated Secure Update and Management (ASUM) of clients with The Center's architecture. The purpose is to be establish a set of foundational requirements to be realized via different architectural paradigms that will specify the protocols, message exchanges, and related requirements.

1.2 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"SHALL" This word means that the item is an absolute requirement of this specification.

"SHALL NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product; for example, another vendor may omit the same item.

2 References

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification.

Notwithstanding, intellectual property rights may be required to use or implement such normative references.

All references are subject to revision, and parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

[CMI-DOC-TD] “Terms and Definitions”, Center for Medical Interoperability, Mar. 2019

https://medicalinteroperability.org/specifications/D01/CMI-DOC-TD-D01-20190311.pdf

[CMI-TR-F-SEC] “Security Considerations for Foundational Efforts”, Center for Medical Interoperability, Mar. 2019.

https://medicalinteroperability.org/specifications/D01/CMI-TR-F-SEC-D01-20190311.pdf

[CMI-SP-CDI-IHE-PCD-IST] “Clinical Data Interoperability using IHE PCD – Identity and Secure Transport Specification”, Center for Medical Interoperability, Mar. 2019.

https://medicalinteroperability.org/specifications/D01/CMI-SP-CDI-IHE-PCD-IST-D01-20190311.pdf

[HL7 FHIR] “Fast Healthcare Interoperability Resources”,

https://www.hl7.org/fhir/overview.html

[IETF-RFC2818] “HTTP over TLS”,

https://tools.ietf.org/html/rfc2818

[IETF-RFC5246] “The Transport Layer Security (TLS) Protocol Version 1.2”,

https://tools.ietf.org/html/rfc5246

[IHE-PCD] “Integrating the Healthcare Enterprise (IHE) Patient Care Device (PCD)”,

https://www.ihe.net/Patient_Care_Devices/

[OWASP] “Open Web Application Security Project (OWASP)”,

https://owasp.org/

[CMI-SP-F-ID] “Identity Specification”, Center for Medical Interoperability, Mar. 2019.

https://medicalinteroperability.org/specifications/D01/CMI-SP-F-ID-D01-20190311.pdf

2.2 Informative References

This specification does not provide any informative references.

2.3 Reference Acquisition

Center for Medical Interoperability, 8 City Boulevard, Suite 203 | Nashville, TN 37209;
Phone +1-615-257-6410; http://medicalinteroperability.org/

3 Terms and Definitions

This document relies on the terms and definitions specified in [CMI-DOC-TD]. In addition, it specifies the following additional terms.

ASUM Management Entity A logical network element that implements the non-client portion of the ASUM interface as described in this document.

Software Repository The logical network entity from which a client can obtain the software updates.

Vendor Software Update Server A logical network component that is part of the client manufacturer’s network that can provide information related to software updates.

4 Abbreviations and Acronyms

This document uses the following abbreviations:

API	Application Programming Interface
ASUM	Automated Software Update and Management
CMI	Center For Medical Interoperability
IP	Internet Protocol
NTP	Network Time Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

5 Overview

The Automated Secure Software Update and Management, or ASUM, framework describes the process by which software is updated on clients in an interoperable manner. Software may be updated for a variety of reasons: e.g., feature enhancements, bug fixes, to address cybersecurity threats, among others. The nomenclature, ASUM, reflects the direction of The Center's members for the process to be secure, automated, and interoperable across clients from different vendors.

Software update mechanisms currently range from manual processes to vendor-specific automation. A client's software architecture can also vary. Clients can use third-party operating systems (e.g., Microsoft® Windows®, Linux®), customized third-party operating systems, or vendor-specific operating systems. The client application software component – the portion that performs the medical functions – may be tightly coupled with the underlying operating system (e.g., co-compiled) or be a separate application. The application software component may be a monolithic image, be part of a mulit-component design, have various sub-components, etc. This client application software component may or may not support network connections (e.g., via Internet Protocol, or IP). Even when a network connection exists, there may be cases where it should not be updated (e.g., when the client is operational) or cannot be updated (e.g., failures that require manual intervention).

To address the myriad of possibilities of client architectures, the members have recommended an iterative approach to ensuring interoperable, secure, automated updates. Only clients that are networked and accessible via an IP network are in scope. Updates include both cyber security and functional updates of client application software components, whether coupled with operating systems or independent of them. This foundational specification applies to both devices and clients, though initial ASUM solution specifications will be focused on clients that are gateways. Updates particular to clients that are devices will be specified in the future.

Further, this document specifies an iteration of a normative framework for ASUM. The components of this iteration can be broadly categorized into: a) communications between a logical 'ASUM management entity' and the client; and, b) transport protocol to obtain the software update. The former includes mechanisms to obtain relevant information from the client, indicate the availability of software to the client, and for the client to report the status of an update attempt. The latter is the process by means of which the client securely obtains a software image.

Protocols for communication between the ASUM management entity and the client are specified as ASUM solutions and are documented separately. These will be specified using specific protocols – such as [IHE-PCD] - so that they can leverage relevant management protocols.

The requirements and a protocol for securely obtaining the software image for update are specified in this document. In this iteration, client implementations that are verified to meet the software update transport requirements are exempt from implementing this specific update mechanism. Client solutions that don't fully meet the requirements will be expected to implement this secure transport mechanism.

Finally, the mechanism for the health system to obtain available software updates from manufacturers for their deployed clients is an important step. However, it is deemed out of scope for this iteration.

5.1 Framework Components

The primary framework components are the ASUM management entity, the software repository, and the client. In addition, vendors may support a vendor software update server, a network component that provides automated communications to the healthcare facility. Outside of the client, all other network components are deemed to be logical in nature. These logical network components can be implemented within other backoffice components or as standalone solutions.

The interface requirements between the client and the ASUM management entity, and the client and the software repository, are specified in this document. The interactions between the healthcare facility and the vendor software update server are out of scope for this iteration. These are shown in Figure 1.

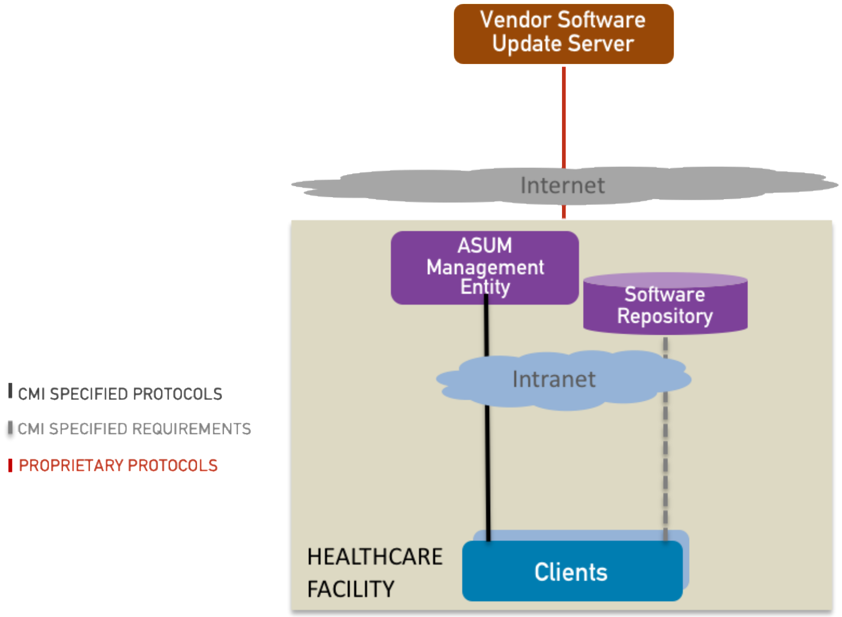

Figure 1-Framework Scope

5.2 Deployment Scenarios

The framework components and the interfaces in this document can support multiple deployment scenarios. A few deployment examples are illustrated in the following diagrams. There may be other options not covered here, e.g., hybrid option in a deployment. The framework is not prescriptive about which option should be used. The expectation is that it will be a hybrid of what is illustrated. Over time, members have indicated a desire to iteratively move towards the example shown in Figure 2. An ideal scenario envisions an update to this wherein all of the interfaces are consistent and interoperable. At this time, the interface between the healthcare facility and the vendor software update server is deemed to be out of scope.

The deployment example in Figure 2 illustrates the scenario where all the clients support the interoperable interface for communication with the ASUM management entity. It also assumes that these clients support an interoperable interface that can be realized via a single software repository. This may be based on compliance with the software update transport mechanism described by the framework or via a software repository that implements all of the various software update transport options.

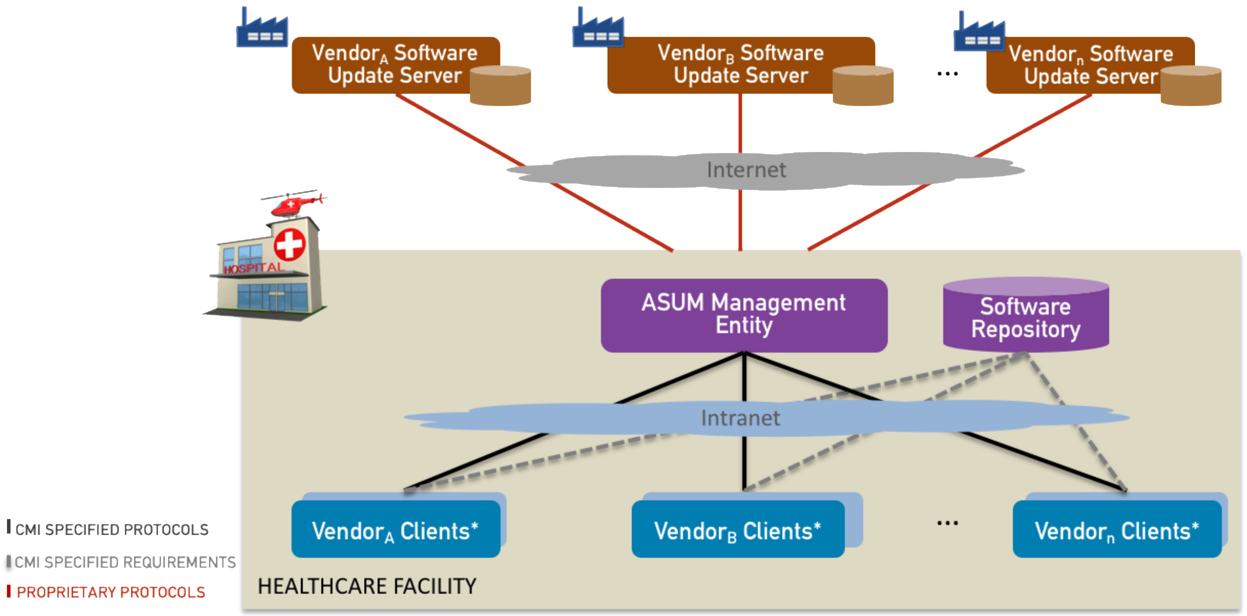

Figure 2-Deployment Option Example A

Figure 3 showcases the deployment scenario with a single ASUM management entity and vendor-specific software repositories that implement the corresponding client supported software update transport mechanism. There is a variation of this deployment scenario in which there is no software repository within the Healthcare Facility for a given vendor system.

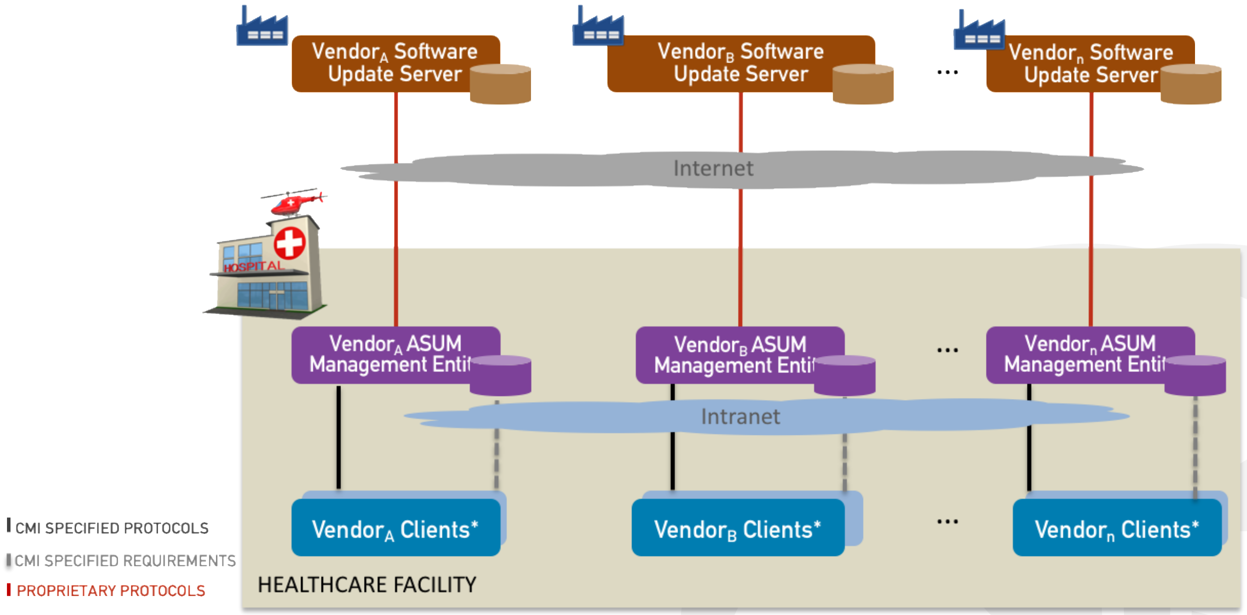
Vendor$_A$ Software Update Server
Vendor$_B$ Software Update Server
…
Vendor$_n$ Software Update Server
Internet
HOSPITAL
Vendor$_A$ ASUM Management Entit
Vendor$_B$ ASUM Management Entit
…
Vendor$_n$ ASUM Management Entit
Intranet
Vendor$_A$ Clients*
Vendor$_B$ Clients*
…
Vendor$_n$ Clients*
CMI SPECIFIED PROTOCOLS
CMI SPECIFIED REQUIREMENTS
PROPRIETARY PROTOCOLS
HEALTHCARE FACILITY

Figure 3- Deployment Option Example B

Figure 4 illustrates a deployment scenario with multiple management entities and associated vendor-specific software repositories that implement the corresponding client supported software update transport mechanisms.

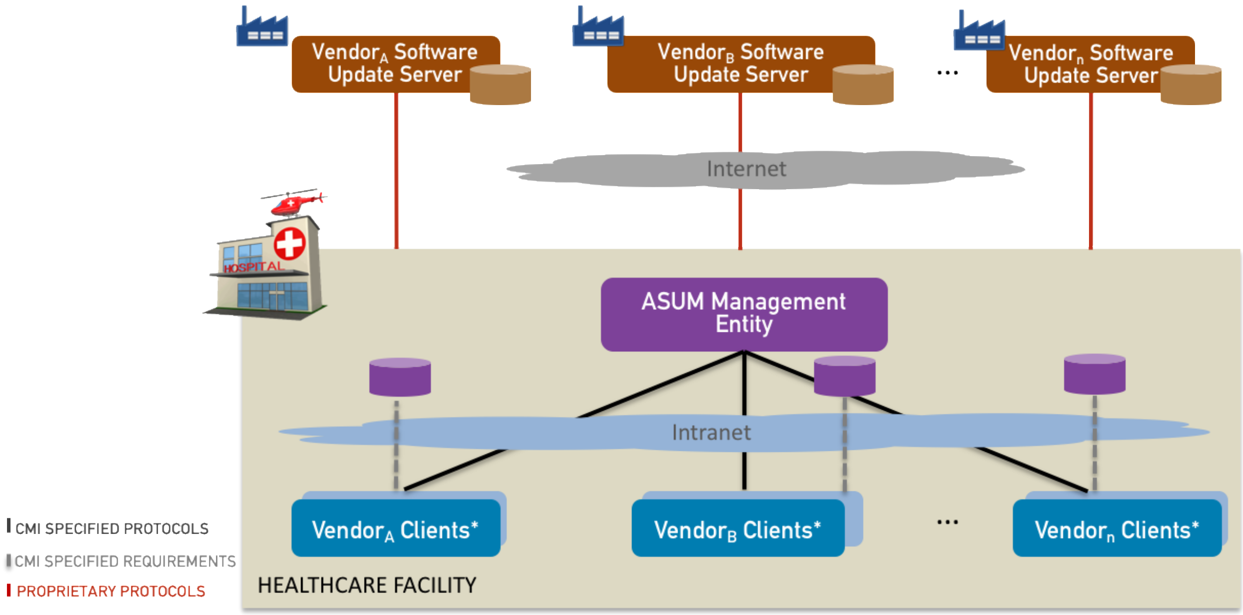

Figure 4- Deployment Option Example C

6 ASUM Framework Requirements

This section provides the normative framework requirements for CMI-specified ASUM solutions. For this iteration, requirements are provided for two interfaces: a) client and the ASUM management entity; and, b) client and a software repository. This section presents requirements that need to be met by ASUM solutions, and other relevant considerations related to the scope.

6.1 Trust Management

Code verification signing SHALL use identities (certificates and keys) issued by The Center and SHALL comply with [CMI-SP-F-ID]. Only CMI issued certificates SHALL be used. Code Verification Certificates (CVCs) SHALL be issued to vendors and hospitals for signing images. Clients and management entities SHALL use CMI medical device, platform, and server certificates to implement secure channels for automated secure update and management function, to include software transport.

Credentials (certificate and keys) SHALL NOT be changed through the ASUM processes. Any update that requires the client to change or renew identity is not in scope at this time.

6.2 Client and ASUM Management Entity Communications

ASUM solutions SHALL specify the protocol and interoperability requirements for communications between the ASUM management entity and the client. To initiate such communications, the ASUM solution SHALL specify an automated mechanism for the client to discover the ASUM management entity. All ASUM solution communications SHALL be authenticated and encrypted using keys and certificates issued by The Center.

Once discovered, the ASUM solution SHALL specify what the client needs to communicate (e.g., client software version) to the ASUM management entity. ASUM solutions SHALL specify how this information is shared from the client to the ASUM management entity. ASUM solutions MAY consider a management-pull, client-push, or both.

ASUM solutions SHALL also specify the notification mechanism by means of which the client is indicated to update software. Clients that comply with this Framework SHALL always use time obtained via NTP for time communications, unless NTP time is unavailable. If NTP time is unavailable, then Clients MAY use client-specific time. ASUM Management Entities SHALL also use time synchronized via NTP for time communications.

6.3 Client Information

For the purposes of inventory management, ASUM solutions SHALL require the clients to report the following to the ASUM management entity:

- Software version details (e.g., version number)
- Client model indicator
- Supported clinical data interoperability protocol version
- Operating system version
- Current status (operational | graceful shutdown in progress | etc.)

ASUM solutions SHALL specify when the above information is to be shared: anytime there is communication, when requested, etc. At a minimum, ASUM solutions SHALL require clients to report the above information to the ASUM Management Entity when it is initiated (e.g. bootup/power on/initialization including reset/reboot). ASUM solutions that don't support an ad-hoc management entity query SHALL report this info to the ASUM Management Entity during a graceful shutdown to enable the ASUM management entity to recognize the modified state of the client.

ASUM solutions SHALL provide mechanisms for clients to report any additional information, such as software or hardware dependencies and any vendor-specific info.

6.4 Software Update Notification

ASUM solutions SHALL ensure that an update notification includes: software image URL or URI, the authentication method (e.g. CVC), and related info (e.g. health system signature required).

For this iteration, ASUM solutions SHALL require clients to update within a specified timeframe. If the update does not happen within this timeframe, ASUM solutions SHALL require clients to refuse the update and respond with a failure status.

ASUM solutions SHALL allow clients to reject any notifications that are incomplete or incomprehensible. In addition Annex C provides a list of management codes that the clients will use to report status updates related to ASUM.

6.5 Secure Software Transport

ASUM solutions SHALL require clients to securely obtain the software image over a mutually authenticated channel that protects data integrity. One potential solution is provided in Annex A.

ASUM solutions SHALL allow vendor-specific software update solutions that comply with these requirements. In the absence of compliant vendor specific solutions, ASUM solutions SHALL require a mechanism as specified in this document, or specify a conformant alternative.

6.6 Software Image Verification

ASUM solutions SHALL require the ability to cryptographically verify the authenticated software update image as being sent by a valid source (e.g., manufacturer, health system, or both). Given the members emphasis on security and to leverage the strong digital identities framework within The Center's efforts, a mechanism is specified in Annex B. ASUM solutions SHALL use the mechanism in Annex B, or provide an equivalent alternative.

6.7 Software Update Status

ASUM solutions SHALL require clients to report the success or failure of software update attempts. ASUM solutions SHALL consider the following, at a minimum:

- Success: without issues | with warnings | with errors
- Failure

The ASUM solution SHALL allow the client to provide any additional information regarding errors and warnings. Here is a subset of errors to consider:

- Could not download image, incorrect URL
- Could not download image, mutual authentication failed
- Software image corrupted

- Software image could not be loaded
- Manufacturer signature (or equivalent indicator) not valid
- Manufacturer signature (or equivalent indicator) valid but not accepted according to provisioned policy (e.g., licensing)
- Health system signature (or equivalent indicator) not valid
- Health system signature (or equivalent indicator) valid but not accepted according to provisioned policy (e.g., licensing)
- Cause unknown

ASUM solutions SHALL recognize that there are cases when the client may require manual intervention due to irrecoverable issues.

6.8 Hardware Updates

Hardware updates SHALL be out of scope for ASUM solutions.

6.9 Operating Systems

In this iteration, ASUM solutions SHALL NOT consider or prohibit operating system updates to clients, except when they are tightly coupled with the application software component. In other words, third-party operating systems that are not co-compiled with the client application software component are out of scope. When tightly coupled, the vendor is responsible for including the operating system with software updates. Operating Systems that are not co-compiled with the application could still use this framework, but are not required to.

6.10 In-use Considerations

For this iteration, ASUM solutions SHALL NOT require clients to identify when they should, or should not, be updated. However, if clients have the ability to do so, ASUM solutions SHALL allow them to use these capabilities.

6.11 Client Considerations

ASUM solutions SHOULD consider clients that are resource constrained (e.g., battery operated clients) and in constrained environments (e.g., wireless clients).

Annex A. Secure Transport Mechanism (NORMATIVE)

This section specifies a solution for secure software updates that meets the ASUM framework requirements. ASUM solutions SHALL require clients to support this, or another conformant secure transport option.

A.1 Overview

The chosen secure transport mechanism is HTTP over TLS. The authentication leverages digital-certificate-based identities that clients and other network elements are required to support with The Center's architecture.

In addition to securing the transport, the digital certificates are also used to authenticate the software image. This is accomplished via a digital signature using the private key of a Manufacture or Health System digital certificate's private key. This signature is then verified by the client, using the associated public key. The digital certificates and keys used SHALL comply with The Center's identity requirements [CMI-SP-F-ID].

When the client is provided with a software update notification, the ASUM management entity provides a HTTPS URL and information regarding the certificate(s) to be used to authenticate the software image. The client then establishes a mutually authenticated HTTPS connection to retrieve the software image. It then verifies the digital signature prior to attempting an update. If the digital signature cannot be verified by the client, the client SHALL reject the software image and report the failure. If the digital signature is verified then a client SHALL attempt a software update. Further details on software image authentication (code signing and verification) is provided in Annex B.

A.2 Secure Transport Information in Update Notification

ASUM solutions that utilize this secure software transport mechanism SHALL specify the HTTPS URL from where the client can obtain the software update and the digital certificate to use for authentication during a software update notification. Clients SHALL reject any software update notification that does not provide an HTTPS URL. Clients SHALL also reject any update notification that does not specify the digital certificate to use for authentication.

A.3 Secure Transport Requirements

Clients SHALL establish a TLS 1.2 [IETF-RFC5246] session in accordance with [CMI-SP-CDI-IHE-PCD-IST] with the server indicated by the HTTPS URL the client received for software update. The client SHALL use the default TCP port of 443 unless reconfigured.

Clients SHALL only include HTTPS in the supported application protocols field of the ClientHello message of the TLS session. TLS sessions may be re-used. Clients SHALL initiate the file transfer and use HTTP GET method to obtain the software image. Clients SHALL NOT download the software image from any TLS session that is not mutually authenticated.

A.4 Software Repository Requirements

Software Repositories SHOULD follow [OWASP] guidelines for all implemented functionality. The software repository SHOULD NOT be implemented as a full web implementation. The Software Repository SHOULD implement the minimal functionality necessary to support this specification.

Annex B. Software Image Authentication (NORMATIVE)

To ensure that clients only download authorized software images, they are required to cryptographically authenticate the software provided even when it is over a mutually authenticated channel. This can help minimize issues due to security vulnerabilities at the software repository, or the process by means of which the repository obtains the software images. This section provides requirements for this verification.

B.1 Manufacturer's Preparation of the Software Image

The Client's Manufacturer SHALL sign the software image (file) using a digital certificate that chains up to The Center's Code Verification Certificate (CVC).

In addition to being signed, software files MAY be encrypted. Encryption of software at rest is not in the current scope of this specification, though that functionality may be considered in the future.

The transfer of this software image from the manufacturer to the health system's software repository is currently out of scope for the ASUM framework.

The software files (images) SHALL have n signed image header. Image signature header contains certificates (including revocation details), signatures, and possibly the file name of the software image. The signature header MAY be appended to the software image, or conveyed as a separate file. If conveyed separately, the signature header SHALL contain the file name of the associated software image.

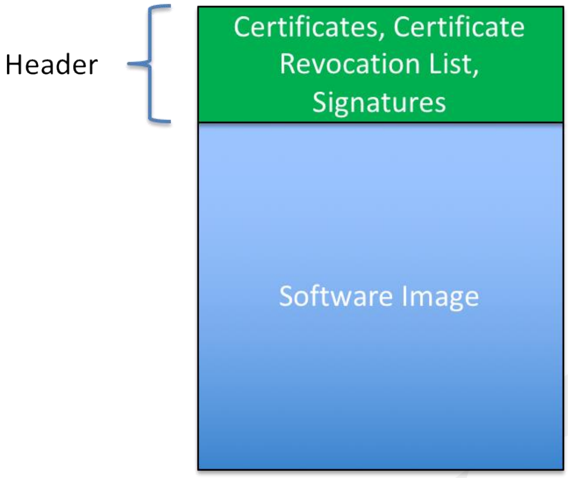

Figure 5: Example signed software image

Signatures SHALL be produced using SHA256. The file digest SHALL be signed using the CVC private key of the manufacturer (see [CMI-SP-F-ID]). Either RSA or ECC certificates and keys MAY be used.

B.2 Health Systems' Preparation of the Software Image

If the Health System desires to add an additional layer of authentication, then the Health System SHALL co-sign using a private key associated with a CVC digital certificate issued by the The Center's PKI. Accordingly, the Health System SHALL append appropriate certificates and signatures to the signed image header. The Health System will create a digest of the software image and sign the digest which will also be appended to the signed image header.

In addition to being signed, the health system MAY encrypt the software image. Encryption of software at rest is not in the current scope of this specification, though that functionality may be considered in the future.

B.3 Client Verification of Software Image

The Client SHALL verify the software image once it is downloaded before update is performed or any function is executed. Verification SHALL occur in the following order.

1. The certificates SHALL be verified (e.g., chained).
2. Revocation status SHALL be verified.
3. Digests of the associated software image SHALL be produced using SHA256.
4. If the Health System co-signed, Health System authorization SHALL be verified using the public key provided in the corresponding certificate and the compared against the digest. If verification is successful, the software update SHALL be deemed authorized and continue; otherwise, an unsuccessful message SHALL be sent accordingly and the update process SHALL terminate.
5. The manufacturer signature SHALL be verified using the public key provided in the corresponding certificate and the compared against the digest. If verification is successful, the software image SHALL be deemed authentic and the software update SHALL continue; otherwise, an unsuccessful message SHALL be sent accordingly and the update process SHALL terminate.

The Client SHALL attempt to update the system if the update is authorized and authentic.

B.4 Including Authentication and Encryption Information in the Software Update Notification

ASUM solutions SHALL accommodate for the authentication and encryption requirements in the software update notification. Specifically, ASUM solutions SHALL allow for the following: manufacturer authentication method, presence of the health system authentication.

Annex C. Management Codes (NORMATIVE)

This Annex specifies management codes that will be utilized by ASUM solutions. These are conditions that are reported by clients when they encounter specific states. ASUM solutions SHALL utilize these codes within their solutions. ASUM solutions MAY enhance these codes with additional codes.

Table 1 provides the code format, and Table 2 provides the specific codes to use for ASUM.

Table 1 – Code Format

FORMAT	< Specifier>-< type>-<category>-<identifier>-<Optional Text>
SPECIFICS	["CMI"|”CMI_V_<vendor-specific-identifier>”]-["S"|"E"|"W"|"I"]-["ASUM"|"CDI"|"ANC"]-[00000-99999]-[<Addl Info>]
DESCRIPTION	SPECIFIED BY ☞ CMI = CMI-specified codes| <vendor-specific unique identifier> (codes not specified by CMI) TYPE ☞ S = Success | E = Error | W = Warning | I = Informational CATEGORY ☞ ASUM = Automated Secure Update & Management IDENTIFIER ☞ XXXXX = number corresponding to a specific event ADDITIONAL INFORMATION ☞ Optional Text
EXAMPLE	CMI-S-ASUM-00000 refers to "Software Update SuccessFul"

Table 2 – ASUM Codes

	ID	DESCRIPTION	ADDITIONAL DETAILS	OPTIONAL TEXT	NEXT STEPS
S	**00000**	Software update successful	No errors or warnings	*none*	*none*
E	**00001**	ASUM management entity connection failure @ L2/L3	L2 or L3 error	*share addl info (e.g., ICMP error), if available*	*backoff/retry*
E	**00002**	ASUM management entity connection failure @ L4 Error	(i.e. firewall misconfiguration, etc)	*share addl info, if available*	*backoff/retry*
E	**00003**	ASUM management entity connection failure @ L4 NACK		*share addl info, if available*	*backoff/retry*
E	**00004**	ASUM management entity connection failure @ Auth	Authentication failed	*share authentication identity or other information, if supported*	*backoff/retry*
E	**00005**	ASUM management entity connection failure @ session establishment	Session establishment failed	*share addl details, if available*	*backoff/retry*
E	**00006**	ASUM management entity ACK failure	When Signaling (e.g., MEM DMC) ACK is required and is not received	*share addl details, if available*	*backoff/retry*
S	**00010**	Software update successfully requested	Software update request looks good	*share addl details, if available*	*none*
S	**00011**	Software update cancellation by ASUM management entity successful	ASUM Management entity cancelled software update	*none*	*none*
E	**00012**	Software update cancellation by ASUM management entity failed	Client could not process cancellation of software update	*share addl details, if available*	*none*
E	**00020**	Software update request rejected: URL/URI error	Client checks the format and determines errors (if supported)	*share addl details, if available*	*none*
E	**00021**	Software update request rejected: Authentication details erroneous	Client checks the authentication details and determines errors (if supported)	*share addl details, if available*	*none*
E	**00022**	Software update request rejected: unspecified error	Client determines errors not explicitly called out	*share addl details, if available*	*none*

W	**00030**	Software update not attempted; client in use		*share addl details, if available*	*none*
E	**00060**	Software update rejected during timeframe; client in use		*share addl details, if available*	*none*
E	**00080**	Image Download failure; incorrect URL/URI	Service unavailable	*share addl details, if available*	*none*
E	**00081**	Image Download failure; could not connect to software repository	Communication Error	*share addl details, if available*	*backoff/retry*
E	**00082**	Image Download failure; authentication error	Mutual authentication while attempting software image	*share addl details, if available*	*none*
E	**00083**	Image Download failure; download interrupted erroneously	Download failures, e.g., connection reset	*share addl details, if available*	*backoff/retry*
E	**00120**	System Update Failed; Software image corrupted	Software image could not be loaded	*share addl details, if available*	*backoff/retry*
E	**00121**	System Update Failed; manufacture authentication failed	e.g., signature invalid	*share addl details, if available*	*none*
E	**00122**	System Update Failed; health system authentication failed	e.g., signature invalid	*share addl details, if available*	*none*
E	**00123**	System Update Failed; license validation failed	client recognizes unuathorized image	*share addl details, if available*	*none*
E	**00124**	System Update Failed; Software image could not be loaded		*share addl details, if available*	*none*
E	**00125**	System Update Failed; Upgrade not compatible with system		*share addl details, if available*	*none*
E	**00126**	System Update Failed; System policy prevents upgrade		*share addl details, if available*	*none*
E	**00127**	System Update Failed; update availability timed out	Update could not be completed within the time window	*share addl details, if available*	*none*
E	**00128**	System Update Failed; unspecified error		*Share addl details specific info*	*vendor specific*
S	**00129**	System Update Success; no known operational errors	Everything A-OK	*share addl details, if available*	*none*
I	**00130**	System Update Success; system set to factory details	System reset to factory defaults	*share addl details, if available*	*none*

S	**00140**	Configuration interval change request successful	These are used when the ASUM solution uses a heartbeat approach, and a periodicity change is requested	*none*	*none*
E	**00141**	Configuration interval change request failed	These are used when the ASUM solution uses a heartbeat approach, and a periodicity change is requested	*none*	*none*
S	**00150**	Management entity change request successful	ASUM management entity modification request	*none*	*none*
E	**00151**	Management entity change request failed	ASUM management entity modification request	*none*	*none*

Appendix I. Manufacturer Considerations (INFORMATIVE)

Many clients use commercial operating systems. Updates of these systems include not only package installation, but also license activation. A typical software update process would involve many steps, including (not necessarily in this precise order):

- acquisition of the software update package (typically through download)
- validation of compatibility and other requirements
- installation of software on base system
- installation software on all clients (which may also include operating system updates)
- license verification of license key with registration process (online or offline)
- configuration of system and applications (required if new features are included in the update)
- user training
- acceptance testing

Consequently, the requirements scope is extensive. The updates will be done with a standard PC type installation process which usually requires an installer (of course, some software installers run resident on their systems, which may be performance impacting and can increase the attack surface of the system). System upgrades may require multiple installations before a system is ready for service. Licensing and CMI certificate handling must be included with the installer, as must handling of OS dependencies. Not all target systems can access on-line license servers, which means other manual mechanisms (license keys or files) must be handled.

I.1 Complex Embedded Device Software Upgrade

Many devices are complex systems. A patient monitor, for example, has multiple subsystems and interconnects to a support server. The monitor may include a measurement rack with a dedicated CPU and measurement modules, each with dedicated microcontrollers. Some systems have integrated multi-measurement modules.

In typical current care environments, the support server will have a repository of all software components, including any supported revisions. Using the support server, a typical process would involve many steps, including (not necessarily in this precise order):

- verify device and software inventory, including software, hardware, and firmware versions of directly and indirectly connected system components
- verify component compatibility with upgrade

- validate license entitlements including software versions and device options (often device and even customer specific)
- determine available update options
- user selects upgrade option and starts upgrade process
- upgrade all affected components
- configuration data may need to be preserved or it may be preferable to reset depending on the nature of the update
- license servers and systems inventories need to be updated
- Subsequent steps include formal acceptance testing, configuration updates (configure new features or even completely new configuration), and user training.

Clearly, this is more than simply a file transfer. The upgrade engine itself needs to be updated. Automation requires protocols to query internal components, track inventory and licensing, and manage upgrade states. Modular systems may need to update multiple packages at once or handle individual and possibly even dynamically built upgrades to single modules. Configuration updates and backups need to be included and some process for approving the device for safe use must be included.

I.2 Embedded Small Devices Upgrade

Small devices also need to be considered. These devices are not always networked. Those that are may have options beyond wired or wireless LAN such as Bluetooth, or even USB tethering to PC or other host. The bootloader on them may be very limited. Their memory and processing capability may be very minimal, as may be their battery capacity.

I.3 Summary of vendor considerations

There are many more variants than were discussed here. Software update mechanisms vary because of device requirements, architecture, and service strategy. Dependencies include:

- Device requirements – cost, size, energy
- Device architecture – modularity, system compatibility
- Service strategy – licensing, service entitlements

Consequently, a stepwise roadmap approach seems appropriate. CMI might constrain scope to defining security requirements such as code signing, secure transfer process, etc.,and defining update trigger events. Implementation can, at least initially, be left open.

Appendix II. Secure Update Recommendations (INFORMATIVE)

Securing the software update mechanism helps prevent the threat of malicious software being installed. Malicious code may change a device's behavior or functions or allow new functions to be performed. For example, a common malware strategy is to make a system available to support Denial of Service (DoS) attacks of other network devices. In the health system, access and distribution of a patient's personal information may be a goal of attackers. It is easy to envision scenarios for successful attacks that may cause harm or death of patients.

Learning from other industry experiences, requirements that should be considered are:

- The medical device must verify that a software update download comes from a trusted source and has not been tampered with.
- The medical device must verify that the software loaded for execution during boot up is valid.
- Keys used for software validation must be securely stored, making it difficult for a hacker to gain access.
- Hospital IT staff must have the option of controlling software updates for medical devices on their network.
- Download is to be triggered by medical device periodic polling and initiation by management command from hospital IT staff.

Some control methods to consider for secure software update include:

- Software image for download signed by manufacturer with PKCS #7 digital signature using certificate PKI (CMI PKI)
  - Medical device verifies signature before accepting download
  - Could also be used for boot up validation
- Download protocols include HTTP(S), SFTP, FTPS, TFTP, and FTP. The protocol used should be easy to implement and should include encryption.
- Software image is co-signed by hospital IT staff when they are controlling updates.
- Triggering mechanism
  - Management messaging (SNMP, TR-069)
  - Provisioning config file settings (includes image co-sign requirement).
  - Medical device periodically polls software image version status from network with proper client to server authentication and encryption.

Appendix III. Acknowledgements

The Center and its member companies would like to extend a heartfelt thanks to all those who participated in the development of this document.

Steve Georinger was the primary author, with technical and editorial contributions from **Sumanth Channabasappa**. Special thanks to those who were involved via a variety of discussions, reviews and input: **Ken Fuchs**, **Eldon Metz, Stuart Hoggan, Jeffrey Brown**, **Kai Hassing**, **Massimiliano Pala, Brionna Lopez** and **George Cragg.**

This effort was conducted within The Center's **Security, Architecture and Requirements,** and **Connectivity** Working Groups, whose members have included the following part-time and full-time participants during the time period that we discussed this version of the document:

WG Participant	**Company Affiliation**
Ali Nakoulima	Cerner
Barry Brown	Mortara
Bill Pelletier	GE
Brionna Lopez	CableLabs
Bruce Friedman	GE Healthcare
Corey Spears	Infor
David Niewolny	RTI
Doug Smith	Laird
Eldon Metz	Innovision Medical
George Cragg	Draeger
Ian Sherlock	TI
James Surine	Smiths Medical
Jay White	Laird
Jeffrey Brown	GE
JF Lancelot	Airstrip
Kai Hassing	Philips
Ken Fuchs	Draeger

WG Participant	Company Affiliation
Massimiliano Pala	CableLabs
Matthew Pekarske	GE
Mike Krajnak	GE
Neil Puthuff	RTI
Phil Raymond	Philips
Scott Coleman	Welch Allyn
Scott Eaton	Mindray
Song Chung	Welch Allyn
Soundharya Nagasubramanian	Welch Allyn
Stefan Karl	Philips
Stuart Hoggan	CableLabs

- *Steve Goeringer (Security Working Group Lead), David Fann, Trevor Pavey, Chris Riha, Paul Schluter, Bernie McKibben, John Williams, Katy Hoyer (Document Editor), Sumanth Channabasappa (Architecture & Requirements and Connectivity Working Group Lead); and, Ed Miller (CTO) -- The Center*