CENTER *for* MEDICAL
INTEROPERABILITY

# The Center for Medical Interoperability Specification Identity

## CMI-SP-F-ID-D02-2019-05-31

***Draft***

**Notice**

# 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.

# Table of Contents

# Document Status Sheet

| | |
|---|---|
| **Document Control Number:** | CMI-SP-ID |
| **Document Title:** | Identity |
| **Revision History:** | D02 IPR Review |
| **Date:** | 03/15/2019 |
| **Status:** | Draft |
| **Distribution Restrictions:** | 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 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.

# 1 Scope

## 1.1 Introduction and Purpose

The Center for Medical Interoperability is a 501(c)(3) organization led by members to change how medical technologies work together. Specifically, CMI aims to improve information flow and make technology function seamlessly in the background to achieve the best possible outcomes for patients. This goal of interoperability is in support of CMI's members' commitment to improve patient safety, care quality and outcomes, and reduce clinician burden and waste.

This document specifies identity of Connected Components. Connected Components include medical devices, gateways, platforms services, and other servers that connect to these CMI architecture elements as illustrated in Figure 1. These components may be hardware or software-based. Identity is the basis on which trusted connectivity and usage must be based. CMI identity will be based on a Public Key Infrastructure (PKI) certificate which will include a public key, unique identifier, and other information as defined in the CMI Certificate Policy. Certificate management will be rooted to the CMI Certificate Authority supported and facilitated by multiple subordinate certificate authorities.

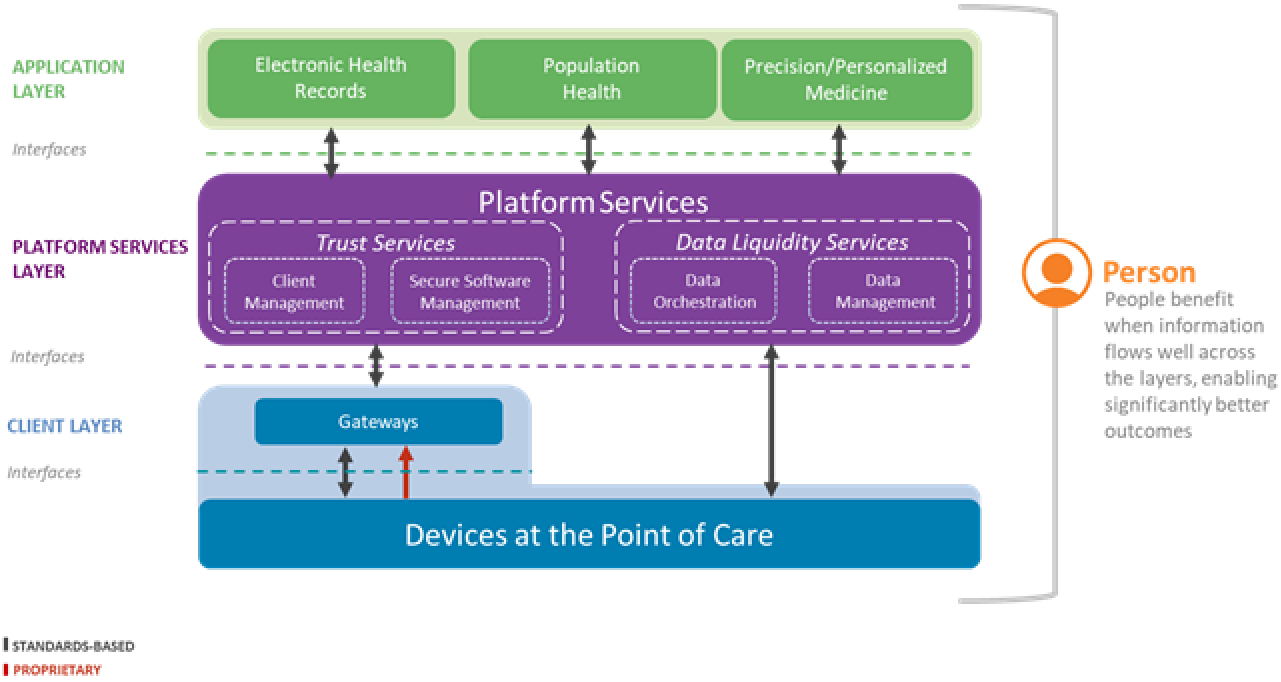

Figure 1: High-level Architecture

A core element of identity is use of a unique identifier. The unique identifier used for CMI identity comprised of a string, uniquely identifying both the requesting organization (typically a vendor) and the actual connected component. The identifier will be used in the CMI RSA and ECC Subscriber Certificates. When paired with a private key, the identifier and associated certificate creates an immutable identity which can be used by a variety of functions to enable secure interoperability.

## 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

This specification uses the following normative references:

**[CMI-SP-F-CP]** Certificate Policy

https://medicalinteroperability.org/specifications

**[CMI-SP-F-PF]** Provisioning Flows

https://medicalinteroperability.org/specifications

**[ITU-T-X.509]** Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.509-201610-I!!PDF-E&type=items

**[IETF-RFC5280]** Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

https://tools.ietf.org/html/rfc5280

**[FIPS 140-2]** Security Requirements for Cryptographic Modules, FIPS 140-2, May 25, 2001.

http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

**[IETF-RFC5272]** Certificate Management over CMS (CMC)

https://tools.ietf.org/html/rfc5272

**[IETF-RFC5273]** Certificate Management over CMS (CMC): Transport Protocols

https://tools.ietf.org/html/rfc5273

**[IETF-RFC5274]** Certificate Management Messages over CMS (CMC): Compliance Requirements

https://tools.ietf.org/html/rfc5274

**[IETF-RFC6402]** Certificate Management over CMS (CMC) Updates

https://tools.ietf.org/html/rfc6402

**[IETF-RFC5652]** Cryptographic Message Syntax (CMS)

https://tools.ietf.org/html/rfc5652

**[IETF-RFC4211]** Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)

https://tools.ietf.org/html/rfc4211

**[IETF-RFC2315]** PKCS #7: Cryptographic Message Syntax Version 1.5

https://tools.ietf.org/html/rfc2315

**[IETF-RFC2986]** PKCS #10: Certification Request Syntax Specification Version 1.7

https://tools.ietf.org/html/rfc2986

**[IETF-RFC6960]** X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

https://tools.ietf.org/html/rfc6960

**[SEMVER-2.0.0]** Semantic Versioning 2.0.0

https://semver.org/#semantic-versioning-200

## 2.2 Informative References

This specification uses the following informative reference:

**[CMI-DOC-TD]** Terms and Definitions

https://medicalinteroperability.org/specifications

**[CMI-TR-F-SEC]** Security Considerations for Foundational Efforts

https://medicalinteroperability.org/specifications

**[CMI-TR-CLC]** Considerations for Certificate Lifecycle

https://medicalinteroperability.org/specifications

**[IETF-RFC7030]** Enrollment over Secure Transport

https://tools.ietf.org/html/rfc7030

**[FDA-SBOM-1]** FDA In Brief: FDA proposes updated cybersecurity recommendations to help ensure device manufacturers are adequately addressing evolving cybersecurity threats (October 17, 2018)

https://www.fda.gov/NewsEvents/Newsroom/FDAInBrief/ucm623624.htm

**[IHE-PCD]** Integrating the Healthcare Enterprise (IHE) Patient Care Device (PCD)

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

**[BlueKrypt-Keylength-31]** "Cryptographic Key Length Recommendation" - v 31.0 - June 10, 2018

https://www.keylength.com/en/4/

**[CAB-CERT-LT]** "Ballot 193 – 825-day Certificate Lifetimes" March 2, 2017

https://cabforum.org/2017/03/17/ballot-193-825-day-certificate-lifetimes/

**[CMI-SP-F-ASUM]** Automated Secure Update and Management Framework

https://medicalinteroperability.org/specifications

**[NIAP-PPAS]** National Information Assurance Partnership Protection Profile for Application Software, Version: 1.2, April 22, 2016

https://www.niap-ccevs.org/MMO/PP/-394-/pp_app_v1.2.htm

## 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 specification uses the terms and definitions in [CMI-DOC-TD]

# 4 Abbreviations and Acronyms

This specification uses the following abbreviations and acronyms:

| | |
|---|---|
| ABAC | Attribute Based Access Control |
| API | Application Programming Interface |
| CA | Certificate Authority |
| CC | Connected Component |
| CMI | Center for Medical Interoperability |
| CMS | Cryptographic Message Syntax |
| CP | Certificate Policy |
| CRMF | Certificate Request Message Syntax |
| CSR | Certificate Signing Request |
| ECC | Elliptical Curve Cryptoptography |
| EST | Enrollment over Secure Transport |
| FQDN | Fully Qualified Domain Name |
| GSMA | Groupe Spéciale Mobile Association. |
| HIBCC | Health Industry Business Communications Council |

| | |
|---|---|
| ICCBBA | International Council for Commonality in Blood Banking Automation |
| IMEI | International Mobile Equipment Identity |
| MAC | Media Access Control |
| ME | Management Entity |
| NIAP | National Information Assurance Partnership |
| OUI | Organizational Unique Identifier |
| PKCS | Public Key Cryptographic Standard |
| PKI | Public Key Infrastructure |
| RA | Registration Authority |
| RBAC | Role Based Access Control |
| RS | Revocation Service (or Server) |
| RSA | Rivest Shamir Adleman |
| SCEP | Simple Certificate Enrollment Protocol |
| SN | Serial Number |
| TPM | Trusted Platform Module |
| UDI | Unique Medical Device Identification |
| UUID | Universally Unique Identifier |

# 5 Identity

CMI defines identity as “The set of characteristics, including PKI certificates, network addresses, and user accounts (user ID and password) by which and individual or device is uniquely recognizable.” This technical report overviews CMI’s approach to identity. Identity will be comprised of a public key and unique identifier that is included in a subscriber certificate as specified in [CMI-SP-F-CP]. The certificate may include other information as specified in the [CMI-SP-F-CP], such as network addresses, certain permanent configuration information, and other information. Identity will be applied to all network components that are part of the CMI architecture. Connected Components include medical devices, gateways, platforms services, and other servers that connect to these CMI architecture elements. These components may be hardware or software based.

The CMI identifier will be an assigned code that ensures an identity will be unique across the entire scope of CMI's trust system. Since the identifier is included in the subscriber certificate, the CMI scope includes both space and time. An identifier SHALL never be reused and any certificate containing an identifier SHALL be revocable and SHALL eventually expire. There may be other useful functions enabled by certificate based identity or the corresponding identifiers, such as indicating the manufacturer of the device or even care system or device type. These other uses are out of scope of this document.

Connected Components and servers to which they connect may participate in other trust systems. Consequently, these systems and end devices may have certificates in addition to CMI issued certificates for use by vendors and system operators (such as the hospital or care provider).

Identity management is deceptively complicated, particularly when considering long term life cycle support. [CMI-TR-CLC] provides explains important considerations and provides the design rational for certain requirements of this specification.

## 5.1 Identity Overview

Identity in the CMI trust ecosystem includes at minimum a unique identifier and a PKI public key that are included in a subscriber certificate as specified in the CMI certificate policy. The identifier uniquely identifies a Connected Component on the network and within the data liquidity scope in which it participates. Common uses of device identity include, but are not limited to:

- Network and service access authentication
- Device verification when performing authentication (is this the correct device?), including device-to-device authentication
- Identification of devices for management, provisioning, or patient association by Platform Services applications

CMI considered multiple options for device identifiers including MAC address, the FDA Unique Medical Device Identification (UDI), and GSMA International Mobile Equipment Identity (IMEI). The MAC address on medical devices may be associated with a module or network interface card that is replaceable and so is not suitable as a device identifier. Moreover, components with both wireless and wired interfaces will have multiple MACs. Not all CMI architecture elements are controlled under the FDA guidelines and so may not have a UDI. IMEIs are primarily used on cellular networks to identify mobile devices. However, most CMI devices will not be connecting to cellular networks and many will not be considered mobile.

Significant consideration was applied to development of the CMI identifier. Even selecting an element that could be used to identify vendors proved challenging. In 2013, the IEEE began restructuring their Registration Authority as shown in IETF Internet Draft OUI Registry Restructuring (https://tools.ietf.org/html/draft-ieee-rac-oui-restructuring-01). IEEE now issues a primer on their Registry Authority which can be viewed at https://standards.ieee.org/develop/regauth/tut/eui.pdf. Two alternatives were to use the IANA Private Enterprise Number (PEN) or a CMI issued company identifier. Ultimately, the choice was to use the most commonly available identifier already possessed by medical device manufacturers, namely, the IEEE MA-L (MAC Address Block Large) which is previously referred to as an OUI. The IEEE provides an alternative which is also accepted, the IEEE CID (Company Identifier).

Two certificate issuance models for identity management have been considered by the Center. One is a static certificate issuance process in which a certificate is installed at the point of manufacturer, or sometimes at the point of installation. Statically issued certificates often are tied to the life cycle of the device by the manufacturer. If the device is found to be compromised, the certificate should be revoked. If the certificate expires, the device is no longer authorized and will not be able to access the network or services.

There are multiple scenarios in which static issuance described above is very restrictive. This is particularly true for host systems (servers, desktops, laptops), software elements, or modular systems. Cryptographic processors and key stores (sometimes implemented as TPMs) may fail and need to be changed. Software based systems may not be bound to hardware and static issuance may rely on white box cryptography (obfuscation of the certificate and key store) which is more vulnerable than hardware based solutions. It may be beneficial to include certification or compliance information in the certificate (so it can be attested). But, since compliance levels may change over time, certificates containing such information should be updated.

Consequently, online certificate issuance solutions may be attractive. One established method of doing this include the Simple Certificate Enrollment Protocol (SCEP). A more recent approach evolving from SCEP is Enrollment of Secure Transport (EST). These types of identity management may provide significant benefits. They also significantly increase the complexity of identity management implementation. This complexity will increase the cost of certificate issuance and will certainly increase the attack surface of the CMI trust ecosystem. Dynamic issuance is still under consideration, but is not supported at this time.

To provide some flexibility in the life cycle of connected components and the associated life cycle of identity, the Center has incorporated a process for certificate renewal – a way for valid and necessary connected components to be issued new identities securely. This process does not leverage SCEP or EST; rather, it relies mostly on traditional message types for certificate signing requests and the associated fulfillment. Devices that leverage this process must be capable of generating new keys – this specification does not allow reuse of existing keys. Moreover, any subscriber requesting certificate renewal through the automated process must have a current (not expired) and valid (not revoked) certificate.

Certificate expiration and revocation introduces reliability concerns in the delivery of care. To minimize care disruption, messaging has been added to the certificate renewal and revocation process to manage the associate process. The goal this messaging is to allow management system and technical staff to proactively manage identity life cycle impacts.
Actual specification of initial issuance methods are outside the scope of this document. General notions are outlined in [CMI-SP-F-CP]. It is important to remember that the issuance method may impact the certificate lifetime (expiration period) and also the revocation method. These are informatively addressed in this document.

## 5.2 Recommendations

This document is a specification, anticipating that these recommendations will be included in CMI specifications (possibly in an evolved version of this document). Consequently, the recommendations below are presented as requirements using the normal CMI terminology for requirements.

# 6 Trust Considerations

## 6.1 Introduction

The CMI trust ecosystem will provide a basis for secure interoperability of care environments at the link, network, and data liquidity layers. The basis for this trust will be a PKI based certificate issuance process with a single root that will include unique identifiers and associated public and private key pairs. This will allow cryptography to support authentication, authorization, privacy, confidentiality, and attestation for multiple purposes. A PKI certificate, which includes the identifier and the public key, amongst other information indicated below, is the digital identity that will provide all trust and security actions.  Consequently, it is essential that identity, that is the PKI certificate, be issued, asserted, and used in accordance with CMI recommendations and requirements.

## 6.2 Connected Component Identity Requirement

Connected Components (devices, gateways, and platform services) SHALL all be issued an identity to connect to CMI compliant networks.

## 6.3 Software or Hardware Element Identity Requirement

Any other software or hardware element (application) that connects to a device, gateway, or platform services SHOULD also have an identifier and, for the purpose of CMI specifications, will also be considered a Connected Component.

## 6.4 Certificate Request Validation Requirement

Certificate requests SHALL be validated by an authority truly accountable for the outcome before signed certificates are issued by the certificate authority.

## 6.5 Certificate Expiration Requirement

Certificates SHALL not be issued indefinitely. They must have an expiration that is determined based on the likelihood of their associated private key being compromised as a function of time.

## 6.6 Certificate Compromised Revocation Requirement

Certificates whose keys are known or suspected of being compromised SHALL be revoked and authentication processes must validate whether keys are revoked prior to authorization.

## 6.7 Expired Certificate Requirement

Certificate expiration SHALL not be ignored. An expired certificate is not valid and access or authorization SHALL NOT be provided.

## 6.8 Dynamic Certificate Requirement

Finally, if an architecture leverages a PKI solution that allows for dynamic certificate issuance, or automated certificate renewal, those processes SHALL not circumvent any of the previous four principals.

These largely form the basis for how and why the [CMI-SP-F-CP] is written. As that specification is not normative to Connected Components, the principals are summarized here. Of course, there are many other factors that must also be included to successfully implement a trust system using PKI. However, those must be executed sympathetic to the principals above.

Clinician (user) and administrator identity is out of scope of these recommendations. However, care systems are highly encouraged to implement strong access control, preferably in accordance with NIST's role based access control (RBAC) or attribute based access control (ABAC) guidelines for authentication and authorization for staff access to CMI components. An application programming interface (API) for secure staff access to devices, gateways, and platform services may need to be specified in the future.

## 6.9 Identifier

The identifier unique identifies the device within the ecosystem. It uniquely identifies the entity that requested the Certificate and the device to which the Certificate is assigned. In its simplest form, this could be just a number. However, it may be useful for other security and management purposes to have a unique identifier that is attested. For example, an attested unique identifier may be useful for access control policies or inventory management. In some cases, this may even be used as a physical label on the device (though this is not required by CMI).

### 6.9.1 Identifier Requirement

The identifier SHALL be included in the component certificate as an X.509 field as indicated in the subscriber profiles later reviewed in this document. The identifier SHALL be a single UTF-8 string composed of four elements separated by a colon (":"). These elements will indicate the version of the identifier, the vendor identity, the type of component identity, and the component identity as summarized below with each element encoded in UTF-8 format:

[Version]:[VendorID]:[Type]:[ComponentID]

Details on each element are below.

#### *6.9.1.1 Version Requirement*

Version: Included to ensure future proofing of the identifier and SHALL be a three digit decimal number between "001" and "999". Connected Components identified in accordance with this release SHALL use version string "001".

#### *6.9.1.2 Vendor ID Requirement*

VendorID: Identifies the organization (typically a vendor or care entity) applying identity to the Connected Component. SHALL be a UTF-8 string corresponding to the 24-bit hex formatted representation of least significant bits of either an IEEE MA-L or a full IEEE CID. The MA-L or CID used SHALL be properly issued by the IEEE to the organization applying the identity to a Connected Component. Information on these registered identities is available at the following IEEE URLs:

- OUI restructuring -- https://tools.ietf.org/html/draft-ieee-rac-oui-restructuring-01
- Registry authority -- https://standards.ieee.org/develop/regauth/tut/eui.pdf
- IEEE MA-L (was OUI) -- https://standards.ieee.org/develop/regauth/oui/index.html
- IEEE CID -- https://standards.ieee.org/develop/regauth/cid/index.html

#### 6.9.1.3 Type Requirement

Type: The identifier may be useful for a wide range of security and management functions. The type field allows functions to determine the type of ComponentID included in the identifier. The Component IDs compliant with this document release are shown below. The appropriate Type SHALL be on of these valid types: “MAC”, “SN”, “UUID”, “HOST”, “FQDN”, “UDI”.

#### 6.9.1.4 ComponentID Requirement

ComponentID: The component of the identifier that ensures the CMI identifier is globally unique. It SHALL correspond to the Type as discussed above. The organization SHALL assure that all identifiers included in certificate requests are unique within their scope and the CA or sub-CA that issues certificates SHALL assure the identifiers of any certificates they issue are unique.

- “MAC” - SHALL be the most significant bits of a MAC address which is the remaining portion not used by the MA-L vender identity portion of the IEEE issued MA-L. That is, the portion that identifies the network component, not the organization asserting the identity. The MAC SHOULD be part of an address block properly issued by the IEEE and owned by the organization asserting the identity.
- “SN”-A serial number according to the needs of the asserting organization. Typically used by equipment manufacturers or software providers. This is probably the most flexible ComponentID type and SHOULD be the default type unless the ComponentID is being used for other security or management purposes as specified by the using medical organization.
- “UUID”-A Universally Unique Identifier provides a 128 bit unique name that can be used as a Uniform Resource Name. It is one way servers and clients that are based on software may be identified. UUIDs SHALL be compliant with IETF RFC 4122.
- “HOST”-Host names are commonly applied to software systems on installation by care facility IT staff. There is no universal approach to creating and asserting host names, but host names used for identifiers SHOULD comply with IETF RFCs 956, 1123, and 1178.
- “FQDN”-Fully Qualified Domain Names are used by DNS to map information resources on servers to IP address. Use of FQDNs SHALL be in compliance with IETF RFC 1035.
- “UDI”-Unique Device Identification issued by an FDA accredited UDI issuing agencies. Accredited issuing agencies at this time are G1, Health Industry Business Communications Council (HIBCC), and the International Council for Commonality in Blood Banking Automation (ICCBBA). (See

https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/UniqueDeviceIdentification/ChangesbetweenUDIProposedandFinalRules/default.htm.)

#### *6.9.1.5 Identifier Encoding Requirement*

The elements discussed above SHALL be encoded in the order shown and SHALL be delineated by a UTF-8 colon, ":".

### 6.9.2 Device Identity Certificate Policy Requirement

The device identity is reflected by the device certificate, which SHALL be issued according to [CMI-SP-F-CP].

### 6.9.3 Certificate Identifier Requirement

The certificate SHALL include the identifier (see Section 5.3.3).

### 6.9.4 Certificate compliance

The certificate SHALL be in compliance with [ITU-T-X.509] and [IETF-RFC5280].

## 6.10 Certificate PKI Hierarchy

The CMI PKI is a three tier infrastructure with a CMI Root CA at tier 1 that issues intermediate CA certificates (i.e., sub-CAs) at tier 2. The tier 2 sub-CAs issue compliant end-entity Subscriber certificates at tier 3 (see figure below). Three different CA chains anchored to a CMI Root CA have been identified: Manufacturer, Care Provider, and Code Verification. Additional CA claims may be added in the future. The CMI will make the Root CA and intermediate CA certificates available to Subscribers. (Note: Subscribers in this context is any element that requires a PKI certificate.)

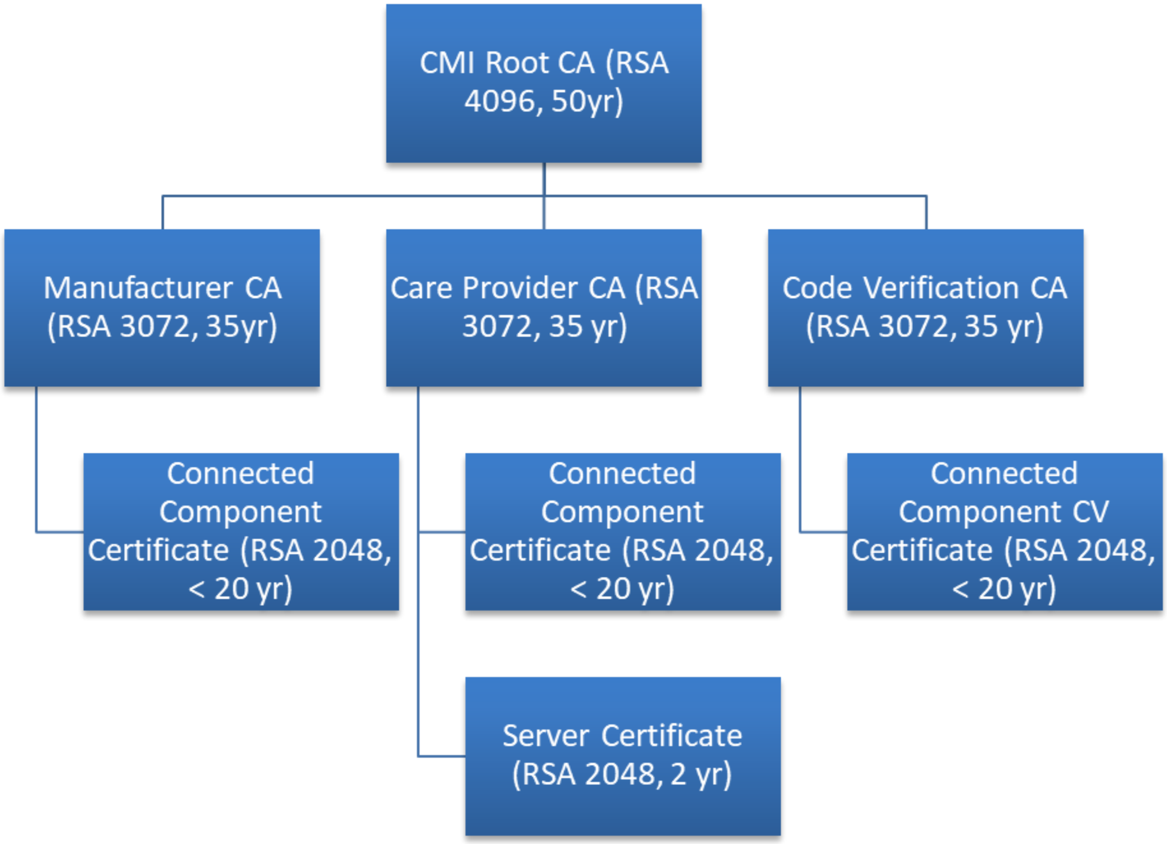
CMI Root CA (RSA 4096, 50yr)
Manufacturer CA (RSA 3072, 35yr)
Care Provider CA (RSA 3072, 35 yr)
Code Verification CA (RSA 3072, 35 yr)
Connected Component Certificate (RSA 2048, < 20 yr)
Connected Component Certificate (RSA 2048, < 20 yr)
Connected Component CV Certificate (RSA 2048, < 20 yr)
Server Certificate (RSA 2048, 2 yr)

*Figure 2: Certificate Hierarchy*

The CMI Root CA is the apex of its Root CA Domain. The Root CA will issue the sub-CA certificates to approved CA service providers. The sub-CAs will issue certificates to authorized Subscribers, which will embed the certificates in compliant devices.

Subscribers should install the CMI authorized Root CA certificate in the trust anchor store of their devices to validate received certificates. The end-entity certificate, its private key, and all sub-CA certificates for a given CA chain should also be installed on the device. During the TLS authentication messaging exchange the end-entity and all sub-CA chain certificates should be sent to the other end point.

The CMI certificate PKI is managed by CMI. CAs are hosted and secured by an experienced, trusted 3rd party approved by CMI. Sub-CAs are centralized and support end-entity subscriber certificate issuance to different medical device manufacturers and hospitals. Manufacturers and hospitals do not operate their own sub-CA unless given approval by CMI. This helps maintain the trust/assurance level of the CMI PKI.

Code Verification Certicates (CVCs) are issued to entities that are responsible for signing software images (including firmware). This supports secure software updates as specified in [CMI-SP-F-ASUM].

### 6.10.1 Software Based Element Certificate Protection Requirement

Software based elements that are issued certificates SHOULD use reasonable best practices to protect them.

### 6.10.2 Online Certificate Issuance Validity Requirement

Online issuance, such as Enrollment over Secure Transport ([IETF-RFC7030]), SHOULD issue certificates with relatively short validity periods, preferably 90 days and certainly not longer than two years.

### 6.10.3 Connected Component CVC Requirement

Connected Components SHALL NOT be issued their own CVCs as Subscribers (there is no function that benefits from this), but their trust store of certificates SHALL also contain whatever CVCs are necessary for them to validate software images.

## 6.11 Certificate Profiles

### 6.11.1 Certificate IETF-RFC5280 Conformance Requirement

CMI PKI Certificates SHALL conform to [IETF-RFC5280]: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.

### 6.11.2 Certificate Identity and Attribute Requirement

CMI PKI Certificates SHALL contain the identity and attribute data of a subject using the base certificate with applicable extensions. The base certificate SHALL contain the version number of the certificate, the certificate's identifying serial number, the signature algorithm used to sign the certificate, the issuer's distinguished name, the validity period of the certificate, the subject's distinguished name, information about the subject's public key, and extensions as defined in the following certificate profile tables.

### 6.11.2.1 Table 1: RSA Root CA Certificate Profile

| **Version** | **v3** |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=US<br>o=CMI<br>ou=RSA Root CA01<br>cn=CMI Root CA |
| **Subject DN** | c=US<br>o=CMI<br>ou= Root CA01<br>cn=CMI Root CA |
| **Validity Period** | 50 yrs |
| **Signature** | Sha512WithRSAEncryption (1.2.840.113549.1.1.13) |
| **Subject Public Key Info**<br>**algorithm**<br>**keysize**<br>**parameters** | RSA (1.2.840.113549.1.1.1)<br>4096-bits<br>NULL |

| **Extensions** | OID | Include | Criticality | Value |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **keyCertSign** | | | | Set |
| **cRLSign** | | | | Set |
| **basicConstraints** | {id-ce 19} | X | TRUE | |
| **cA** | | | | Set |
| **pathLenConstraint** | | | | Not set |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |

<Sub-CA Type> SHALL be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Member”, “Code Verification Certificate”.

<ID#>SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, “CA0001.”

### 6.11.2.2 Table 2: RSA Sub-CA Certificate Profile

| Version | v3 |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=US<br>o=CMI<br>ou=RSA Root CA01<br>cn=CMI RSA Root CA |
| **Subject DN** | c=<Country Code><br>o=<Organization Name><br>ou=RSA <Sub-CA Type> <ID#><br>cn=CMI RSA <Sub-CA Type> |
| **Validity Period** | 30 yrs |
| **Signature** | Sha512WithRSAEncryption (1.2.840.113549.1.1.13) |
| **Subject Public Key Info** | |
| **algorithm** | RSA (1.2.840.113549.1.1.1) |
| **keysize** | 3072-bits |
| **parameters** | NULL |

| **Extensions** | OID | Include | Criticality | Value |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **keyCertSign** | | | | Set |
| **cRLSign** | | | | Set |
| **basicConstraints** | {id-ce 19} | X | TRUE | |
| **cA** | | | | Set |
| **pathLenConstraint** | | | | 0 (zero) |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **authorityKeyIdentifier** | {id-ce 35} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |
| **certificatePolicies** | {id-ce 32} | X | FALSE | |
| **certPolicyId** | | | | <Certificate Policy OID, TBD> |
| **policyQualifiers** | | | | Not set |
| **authorityInfoAccess** | {id-pe 1} | X | FALSE | |
| **AccessDescription** | | | | |
| **accessMethod** | | | | OCSP |
| **accessLocation** | | | | Responder HTTP URI |

<Sub-CA Type> SHALL be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Member”, “Code Verification Certificate”.

<ID#>SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, “CA0001.”

### *6.11.2.3 Table 3: RSA Subscriber Certificate Profile*

| **Version** | **v3** |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=<Country Code><br>o=<Organization Name><br>ou=RSA <Sub-CA Type> <ID#><br>cn=CMI RSA <Sub-CA Type> |
| **Subject DN** | c=<Country Code><br>o=<Organization Name><br>ou=CMI <Device Type> Certificate<br>cn=<Device Identifier> |
| **Validity Period** | 20 yrs |
| **Signature** | Sha384WithRSAEncryption (1.2.840.113549.1.1.12) *or*, |
| **Subject Public Key Info**<br>**algorithm**<br>**keysize**<br>**parameters** | <br>RSA (1.2.840.113549.1.1.1)<br>2048-bits<br>NULL |

| **Extensions** | **OID** | **Include** | **Criticality** | **Value** |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **digitalSignature** | | | | Set |
| **keyEncipherment** | | | | Set |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **authorityKeyIdentifier** | {id-ce 35} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |
| **certificatePolicies** | {id-ce 32} | X | FALSE | |
| **certPolicyId** | | | | <Certificate Policy OID, TBD> |
| **policyQualifiers** | | | | Not set |
| **extKeyUsage** | {id-ce 37} | O | FALSE | |
| **id-kp-serverAuth** | | | | TLS server auth for platform services and other servers |
| **id-kp-clientAuth** | | | | TLS client auth for medical device and gateways |
| **cRLDistributionPoint** | | O | FALSE | |

| Version | | v3 | | |
|---|---|---|---|---|
| **authorityInfoAccess** | {id-pe 1} | X | FALSE | |
| **AccessDescription** | | | | |
| **accessMethod** | | | | OCSP |
| **accessLocation** | | | | Responder HTTP URI |

<Sub-CA Type> SHALL be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Member”, “Code Verification Certificate”.

<ID#> SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, “CA0001.”

<Device Type> SHALL indicate the purpose of the Connected Component to which the certificate is being issued. It SHOULD be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Platform Services”, “Gateway”, “Code Verification Certificate”. This requirement is left non-mandatory as there may be Connected Components that do not clearly meet these descriptors. It is anticipated additional descriptors will be added.

<Device Identifier> SHALL be included and is a globally unique identifier that is persistent as documented in Section 5.3.1. This field SHALL remain unchanged during certificate renewal (and it the basis of calling that process renewal rather than re-issuance).

<extKeyUsage> is optional but if the certificate supports a TLS/SSL client, client auth and server auth SHOULD be indicated as appropriate to the use of the certificate. If extKeyUsage is used, either or both extensions SHALL be used.

### 6.11.2.4 Table 4: ECC Root CA Certificate Profile

| Version | v3 |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=US<br>o=CMI<br>ou=ECC Root CA01<br>cn=CMI ECC Root CA |
| **Subject DN** | c=US<br>o=CMI<br>ou=ECC Root CA01<br>cn=CMI ECC Root CA |
| **Validity Period** | 50 yrs |
| **Signature** | ecdsa-with-Sha512 (1.2.840.10045.4.3.4) |
| **Subject Public Key Info**<br>**algorithm**<br>**parameters** | <br>EC (1.2.840.10045.2.1)<br>Secp521r1 (1.2.840.10045.3.1.35) |

| Extensions | OID | Include | Criticality | Value |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **keyCertSign** | | | | Set |
| **cRLSign** | | | | Set |
| **basicConstraints** | {id-ce 19} | X | TRUE | |
| **cA** | | | | Set |
| **pathLenConstraint** | | | | Not set |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |

<Sub-CA Type> SHALL be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Member”, “Code Verification Certificate”.

<ID#>SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, “CA0001.”

### 6.11.2.5 Table 5: ECC Sub-CA Certificate Profile

| Version | v3 |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=US<br>o=CMI<br>ou=ECC Root CA01<br>cn=CMI ECC Root CA |
| **Subject DN** | c=<Country Code><br>o=<Organization Name><br>ou=ECC <Sub-CA Type> <ID#><br>cn=CMI ECC <Sub-CA Type> |
| **Validity Period** | 30 yrs |
| **Signature** | ecdsa-with-Sha512 (1.2.840.10045.4.3.4) |
| **Subject Public Key Info**<br>**algorithm**<br>**parameters** | EC (1.2.840.10045.2.1)<br>Secp384r1 (1.2.840.10045.3.1.34) |

| Extensions | OID | Include | Criticality | Value |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **keyCertSign** | | | | Set |
| **cRLSign** | | | | Set |
| **basicConstraints** | {id-ce 19} | X | TRUE | |
| **cA** | | | | Set |
| **pathLenConstraint** | | | | 0 (zero) |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **authorityKeyIdentifier** | {id-ce 35} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |
| **certificatePolicies** | {id-ce 32} | X | FALSE | |
| **certPolicyId** | | | | <Certificate Policy OID, TBD> |
| **policyQualifiers** | | | | Not set |
| **authorityInfoAccess** | {id-pe 1} | X | FALSE | |
| **AccessDescription** | | | | |
| **accessMethod** | | | | OCSP |
| **accessLocation** | | | | Responder HTTP URI |

<Sub-CA Type> SHALL be one of the following values not including the quotes: “Medical Device”, “Enterprise Device”, “Member”, “Code Verification Certificate”.

<ID#>SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, “CA0001.”

### 6.11.2.6 Table 6: ECC Subscriber Certificate Profile

| Version | v3 |
|---|---|
| **Serial number** | Unique Positive Integer assigned by the CA and not longer than 20 octets. |
| **Issuer DN** | c=<Country Code><br>o=<Organization Name><br>ou=ECC <Sub-CA Type> <ID#><br>cn=CMI ECC <Sub-CA Type> |
| **Subject DN** | c=<Country Code><br>o=<Organization Name><br>ou=CMI <Device Type> Certificate<br>cn=<Device Identifier> |
| **Validity Period** | 20 yrs |
| **Signature** | ecdsa-with-Sha384 (1.2.840.10045.4.3.3) |
| **Subject Public Key Info**<br>**algorithm**<br>**parameters** | EC (1.2.840.10045.2.1)<br>Secp256r1 (1.2.840.10045.3.1.7) |

| Extensions | OID | Include | Criticality | Value |
|---|---|---|---|---|
| **keyUsage** | {id-ce 15} | X | TRUE | |
| **digitalSignature** | | | | Set |
| **keyAgreement** | | | | Set |
| **subjectKeyIdentifier** | {id-ce 14} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **authorityKeyIdentifier** | {id-ce 35} | X | FALSE | |
| **keyIdentifier** | | | | Calculated per Method 1 |
| **subjectAltName** | {id-ce 17} | O | FALSE | |
| **certificatePolicies** | {id-ce 32} | X | FALSE | |
| **certPolicyId** | | | | <Certificate Policy OID, TBD> |
| **policyQualifiers** | | | | Not set |
| **extKeyUsage** | {id-ce 37} | O | FALSE | |
| **id-kp-serverAuth** | | | | TLS server auth for platform services and other servers |

| Version | | v3 | | |
|---|---|---|---|---|
| **id-kp-clientAuth** | | | | TLS client auth for medical device and gateways |
| **cRLDistributionPoint** | | O | FALSE | |
| **authorityInfoAccess** | {id-pe 1} | X | FALSE | |
| **AccessDescription** | | | | |
| **accessMethod** | | | | OCSP |
| **accessLocation** | | | | Responder HTTP URI |

<Sub-CA Type> SHALL be one of the following values not including the quotes: "Medical Device", "Enterprise Device", "Member", "Code Verification Certificate".

<ID#> SHALL indicate the ID number of the CA and is populated when the CA certificate is issued. For Example, "CA0001."

<Device Type> SHALL indicate the purpose of the Connected Component to which the certificate is being issued. It SHOULD be one of the following values not including the quotes: "Medical Device", "Enterprise Device", "Platform Service", "Code Verification Certificate". This requirement is left non-mandatory as there may be Connected Components that do not clearly meet these descriptors. It is anticipated additional descriptors will be added.

<Device Identifier> SHALL be included and is a globally unique identifier that is persistent as documented in Section 5.3.1. This field SHALL remain unchanged during certificate renewal (and it the basis of calling that process renewal rather than re-issuance).

<extKeyUsage> is optional but if the certificate supports a TLS/SSL client, client auth and server auth SHOULD be indicated as appropriate to the use of the certificate. If extKeyUsage is used, either or both extensions SHALL be used.

## 6.12 Installation and Protection of Secrets and Certificates

### 6.12.1 Secrets and Certificate Protection Requirement

To ensure the integrity of the trust architecture, secrets and certificates SHALL be protected. The intent of protecting these cryptographic elements is to deter cloning or counterfeiting devices, tampering with devices to change their authorized functions or use, and to acquire credentials for the purpose of introducing unauthorized devices into trusted networks. Furthermore, it is important to protect any keys used to encrypt sensitive data whether at rest or in motion. Consequently, the Connected Component SHALL store the Connected Component Certificate private key in a manner that deters (makes difficult) unauthorized disclosure and modification. Installation and protection of secrets (keys) and certificates will depend on the nature of the component and the type of environment in which the Connected Component will operate. The current health industry state of the art does not specify, however, how keys and other secrets will be protected. Below are guidelines based on [FIPS 140-2] that are desired and may become mandatory in the future.

#### 6.12.1.1 Hardware based Connected Components in trusted environments requirement

- The Connected Component SHOULD meet [FIPS 140-2] security requirements for all instances of private and public permanent key storage.
- The Connected Component SHOULD meet [FIPS 140-2] level 1 physical security requirements (production grade enclosure) if it will operate in a trusted environment that is only accessible by authorized hospital staff.
- An ECC or RSA Connected Component certificate, private key, and issuing CA certificate as defined in The Center's Certificate Policy SHALL be securely installed in the Connected Component by the manufacturer.
- An ECC or RSA root CA certificate defined in The Center's Certificate Policy and authorized by The Center SHALL be installed in the Connected Component as a trust anchor for validating received certificates.

#### 6.12.1.2 Hardware based Connected Components in untrusted environments requirement

- The Connected Component SHOULD meet [FIPS 140-2] security requirements for all instances of private and public permanent key storage.
- The Connected Component SHOULD meet [FIPS 140-2] level 3 (tamper detection and key zeroization) or higher if it will operate in an untrusted environment where the public may have access. The Connected Component SHALL meet [FIPS 140-2] level 1 if it does not meet requirements of [FIPS 140-2] level 3.
- An ECC or RSA Connected Component certificate, private key, and issuing CA certificate as defined in The Center's Certificate Policy SHALL be securely installed in the Connected Component by the manufacturer.
- An ECC or RSA root CA certificate defined in The Center's Certificate Policy and authorized by The Center SHALL be installed in the Connected Component as a trust anchor for validating received certificates.

#### 6.12.1.3 Software based Connected Components requirement

- The Connected Component SHOULD store keys securely.
- The Connected Component SHOULD meet [FIPS 140-2] level 1 (cryptographic module to be executed on general purpose computing system).
- The Connected Component SHOULD implement security requirements specified in NIAP Protection Profile for Application Software (NIAP) [NIAP-PPAS]. In particular, storage of credentials SHOULD comply with FCS-STO-EXT.1.
- The Connected Component SHOULD use secure hardware such as a TPM Module.
- The Connected Component SHOULD apply access controls to protect certificates, private keys, and issuing CA certificates.

- A mitigating control, such as Intrusion Detection Systems, SHOULD be used to detect unauthorized access to certificates, private keys, and issuing CA certificates installed on the network component. Both external and internal mitigating controls SHOULD be used.
- An ECC or RSA Connected Component certificate, private key, and issuing CA certificate SHALL be securely installed in the Connected Component by trusted technical staff. Associated cryptographic material and software SHALL be controlled at all times.
- An ECC or RSA Connected Component certificate issued for use on software based Connected Components SHOULD have relatively short (<2 years) Certificate expiration periods.

It must be noted that while the use of white box cryptography is better than not addressing cryptographic security at all, existing solutions are known to be vulnerable to a wide variety of attacks. Consequently, it is highly recommended that hardware based security mechanisms be used. It is possible that on-line certificate issuance such as described by Enrollment over Secure Transport ([IETF-RFC7030]) may mitigate some vulnerabilities of software based Connected Components and may provide a more scalable, extensible trust ecosystem. Similarly, use of Hardware Security Module servers may provide benefits. These areas will be further studied by the Center.

The [FIPS 140-2] guidelines are based on protection profiles derived from the Common Criteria for Information Technology Security Evaluation. Excellent insight on secure implementation of cryptographic modules and their use can be found in the following National Information Assurance Partnership (NIAP) documents:

- NIAP Protection Profile for Application Software
- NIAP Protection Profile for General Purpose Operating Systems
- NIAP General Purpose Operating Systems Protection Profile / Mobile Device Fundamentals Protection Profile Extended Package (EP) Wireless Local Area Network (WLAN) Clients

# 7 Certificate Expiration Periods

## 7.1 Considerations in choosing expiration periods

An interesting aspect about the expiration period of PKI certificates is that the expiration is not really about the certificate. Rather, it's about the private key. Specifically, the expiration period is a set date in time reflecting how long the private key can be kept safely. There are two primary factors in determining this date. The first is how long might it take to factor (break) the key through brute force. For example, lets guess that today it might take up to 20 years to exhaustively factor a 2048-bit RSA key (this period of time is illustrative only). This is, however, a random process, and luck applies. So average luck would force the key in 10 years. So, perhaps the expiration period for a for an RSA based certificate for a 2048-bit key should be less than 10 years.

The second factor in determining the certificate expiration period must consider that the location in which the private keys are stored may be directly compromised. How likely this is depends on how securely the key is stored. A key stored in software is generally easier to compromise than something stored in hardware using, for example, a Trusted Platform Module (TPM).
In addition to the primary factors, we can also consider how long a given type of Connected Component should be authorized to access resources. For example, perhaps the given calibration of a networked sensor is only suitable for three years and the sensor will (should) be disposed of at that time. Deploying a corresponding certificate with an expiration period of three years seems prudent and will help ensure an adversary cannot use the key and certificate from the device if somehow compromised after disposal.

Finally, when certificates are used for authentication, part of the verification process includes checking the validity of the certificate of the issuing Certificate Authority (e.g., certificate chaining). Consequently, there is a relation in a given subscriber certificate and the validity period of the certificate for the issuing Certificate Authority. A Certificate Authority should not issue subscriber certificates valid beyond their own validity period.
There is a wide range of devices, uses, and environments used or encountered in health care. The cybersecurity risks of the key compromise vary accordingly as do the consequences of key compromises. In reality, the choice of a certificate expiration period is actually a bet of how long a given key can be protected for a given application. Factors against the desired outcome include:

- The ability to protect keys continually decreases over time because the ability to factor keys continually improves – non-linearly and non-predictably.
- Systemic faults in cryptography solutions are frequently discovered and operational errors in the distribution and management of keys may occur and be realized at some random future time.
- There are anticipated threats to cryptography, including use quantum computing to accelerate cryptanalysis, that will decrease the period of time a given key type and length can be protected.

It is important to accept that the bet made is not an “if a key will be compromised” but, rather, “how long will it be till a key is compromised”. And, that period of time decreases as adversaries develop exploits and as processing power increases which can be used to break keys faster. In other words, there is a design constraint that any specified certificate expiration period chosen today will, at some non-deterministic point of time in the future, be proven insufficient.
Consequently, the organizations that are responsible for the cybersecurity outcomes – hospitals and vendors – should choose appropriate key expiration. This should be done after diligent analysis of the risks for a specific environment, the specific systems used to provide care, and the use cases in which the keys will be used to protect patients’ interests. This choice should be made at the time Certificate Signing Requests are submitted to the PKI Certificate Authority.

## 7.2 Examples of certificate validity periods

While choice of expiration periods must remain the responsibility of security professionals at hospitals and vendors, it is useful to provide an example of possibly prudent certificate validity periods. Some samples are provided below. These are strictly informative (not normative). Furthermore, these examples should be considered in context of the time this report was written (September 2018) and represent the longest expiration periods that might be considered responsible at that time.

- Manually deployed and installed certificates on hardware based devices: As long as the anticipated life of the device or as long as the maintenance or refresh cycle of the device, not to exceed 20 years assuming reasonable protection of the private key.

- Manually deployed and installed certificates on software based deployment on devices with a TPM (or equivalent): Same as above, assuming verification is performed to validate that the TPM is, in fact, where the key and certificate are actually deployed. (Even on systems with TPM, applications must be coded to leverage the module.)

- Deployment on servers or software systems: As short as vendors and hospitals can operationally accommodate, not to exceed 2 years. This may seem a very short period of time, but is consistent with guidelines from both NIST [BlueKrypt-Keylength-31] and the CA/Browser Forum [CAB-CERT-LT] at the time of writing.

- If and when automatic renewal is used to support dynamic deployment of new certificates: Perhaps only 90 days. However, some science and engineering is required to fully understand the failure modes that may be introduced by such a strategy.

- Regardless of the deployment models above, certificate expiration should never be longer than the expiration date of the signing Certificate Authority.

# 8 Certificate Renewal

It essential that expired certificates be rejected during access and authorization attempts. To ensure valid Connected Components are able to perform necessary clinical functions with no or minimal risk related to certificate management, the Center has incorporated process for automatic renewal of certificates. This section overviews the basic notions of this process. Fundamentally, this process is derived from [IETF-RFC5272] and companion documents including [IETF-RFC5273], [IETF-RFC5274], [IETF-RFC6402], [IETF-RFC4211], [IETF-RFC2315], and [IETF-RFC2986]. In these documents, the equivalent process is referred to as certificate rekeying. Different terminology is used here because the CMI process introduces resiliency controls and mechanisms and also alerts and status messages to be consumed by the Management Entity.

## 8.1 Rationale

Intuitively, it might seem updating or renewing a certificate should occur in the same way certificates are issued. However, this is not actually completely necessary. We should establish an on-line, automated renewal process to allow Connected Components that are valid and necessary to receive new certificates when their current certificates are expiring. Moreover, this process should apply to systems that are hardware or software based regardless of whether the vendor or the hospital installed the certificate. Finally, since the Connected Component already has a current valid certificate, it actually seems more secure to do on-line automatic certificate renewal than on-line, automatic certificate issuance (using a process such as Enrollment over Secure Transport).

The process proposed automates certificate expiration management. Certificate renewal requests can be submitted prior to expiration by the Connected Component on which the expiring certificate is deployed. Corresponding alerts (alarms/notifications) of expiration can be triggered prior to expiration, and can even include escalation according to how soon the certificate will expire.

## 8.2 Certificate renewal management requirements

The Center has compiled the following requirements to minimize care disruptions associated with certificate expiration and renewal:

### 8.2.1 Certificate Renewal Request Requirement

- Certificate renewals SHALL be submitted 2 months prior to expiration by the device on which the certificate is expiring.
  - An information alert (“blue”) SHALL also be sent to the appropriate management servers.

### 8.2.2 Certificate Expiration Alert Requirement

- An alert SHALL be sent to the appropriate management servers 1 month prior to expiration by the device; this SHALL be a “yellow” alert and SHALL escalate to “red” as expiration nears.

  - A peer MAY optionally notify that a peer's certificate is expiring as an outcome of certificate validation during mutual authentication.

### 8.2.3 Certificate Renewal Request Vetting Requirement

- Certificate renewal requests SHALL be vetted and subsequently signed by an appropriate Registration Authority and forwarded to the appropriate Certificate Authority as per [CMI-SP-F-CP].

### 8.2.4 Certificate Renewal Issuance Requirement

- Certificate renewals SHALL be issued only for currently valid certificates (certificates included in renewal Certificate Signing Request SHALL NOT be expired and must chain to a valid Certificate Authority).

## 8.3 Process Overview

The easiest way to illustrate this process is to provide a sequence diagram. This is shown in Figure 3. The diagram shows interactions and functions performed by a Subscriber, Registration Authority (RA), Management Entity, and Certificate Authority (CA). The Subscriber is a Connected Component on which a CMI PKI certificate has previously been installed. The Registration Authority is responsible for ensuring that the Connected Component is valid and should receive a new certificate. The Registration Authority is likely within the hospital, but may alternatively be a function provided by a vendor or perhaps even the CMI. The Management Entity is an IT resource that provides a capability for managing certificate status – it might be a manager or managers or integrated into another resource (perhaps even a Gateway or Platform Services module). It's primary function here is to provide IT staff awareness of certificate life cycle functions. Finally, the Certificate Authority provides final signing that attests the validity of a new certificate.

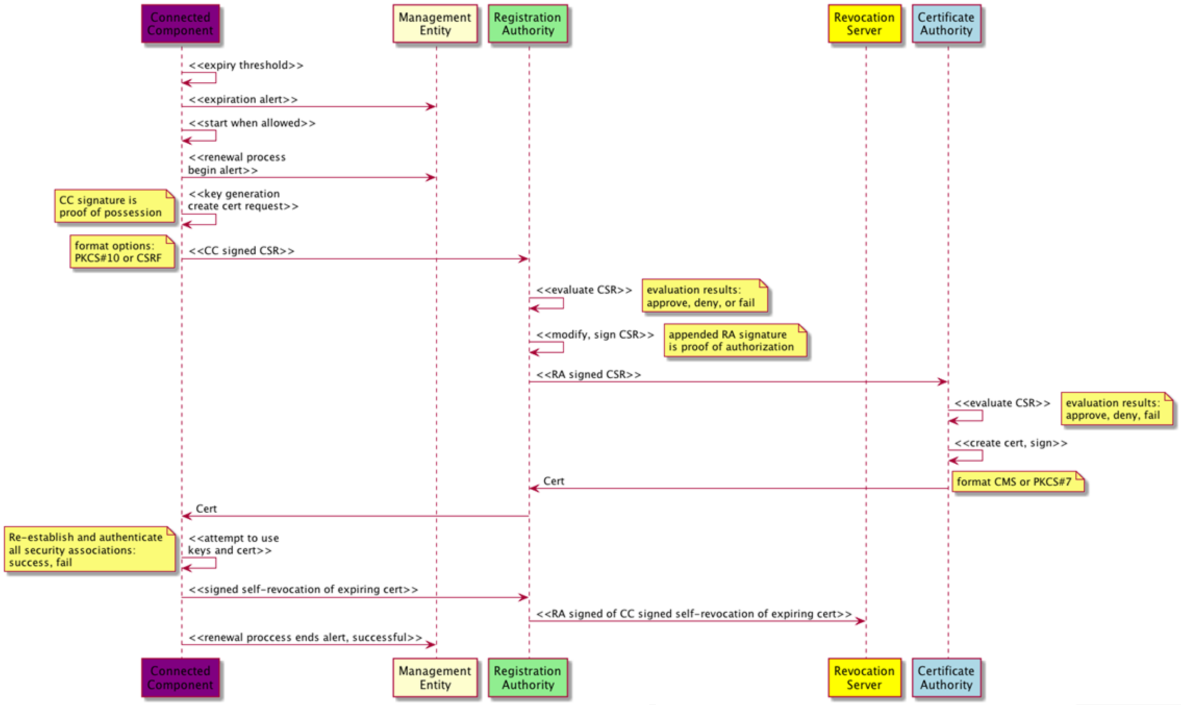
Connected Component
Management Entity
Registration Authority
Revocation Server
Certificate Authority
<<expiry threshold>>
<<expiration alert>>
<<start when allowed>>
<<renewal process begin alert>>
CC signature is proof of possession
<<key generation create cert request>>
format options: PKCS#10 or CSRF
<<CC signed CSR>>
<<evaluate CSR>>
evaluation results: approve, deny, or fail
<<modify, sign CSR>>
appended RA signature is proof of authorization
<<RA signed CSR>>
<<evaluate CSR>>
evaluation results: approve, deny, fail
<<create cert, sign>>
Cert
format CMS or PKCS#7
Cert
Re-establish and authenticate all security associations: success, fail
<<attempt to use keys and cert>>
<<signed self-revocation of expiring cert>>
<<RA signed of CC signed self-revocation of expiring cert>>
<<renewal proccess ends alert, successful>>
Connected Component
Management Entity
Registration Authority
Revocation Server
Certificate Authority

Figure 3: Certificate renewal sequence diagram, full success path

Certificate renewal messages will be conveyed using secure transport. For messages specifically conveying certificate requests and responses, these modify the sequence diagram above as shown in Figure 4.

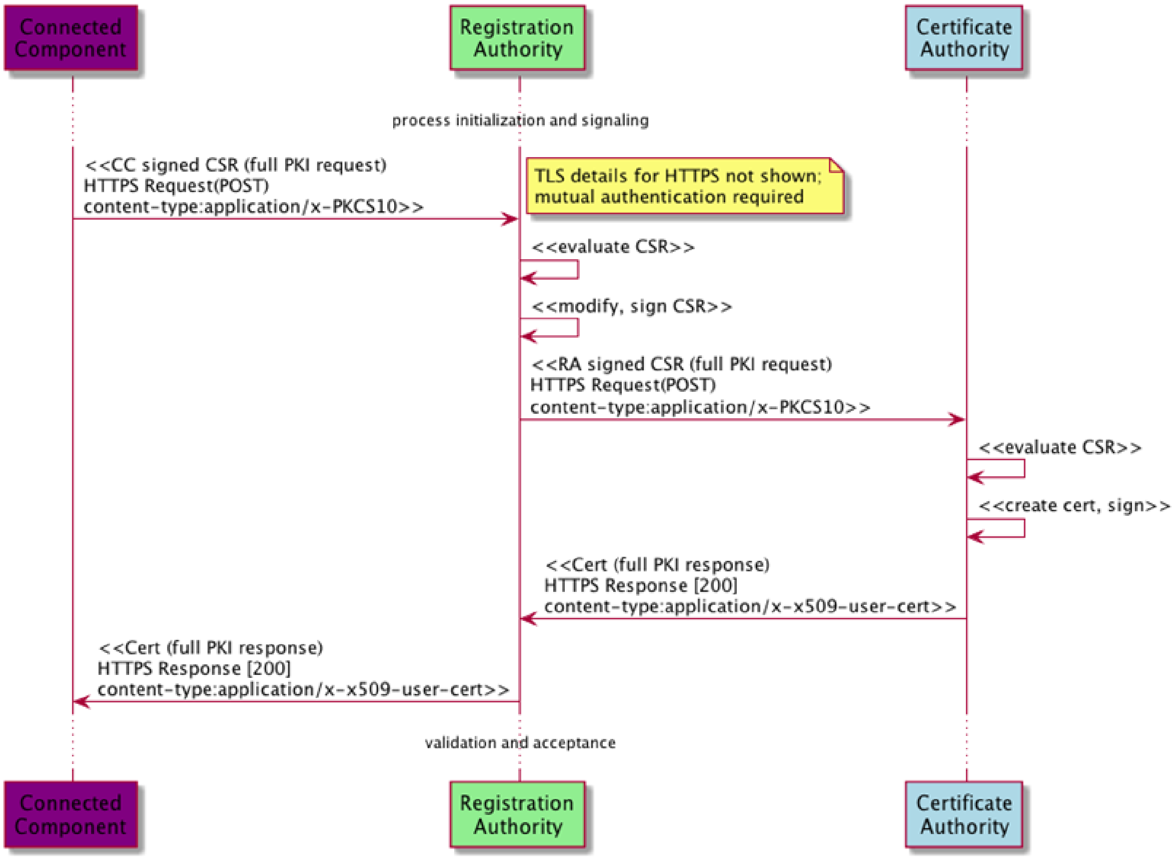
Connected Component
Registration Authority
Certificate Authority
process initialization and signaling
<<CC signed CSR (full PKI request)
HTTPS Request(POST)
content-type:application/x-PKCS10>>
TLS details for HTTPS not shown;
mutual authentication required
<<evaluate CSR>>
<<modify, sign CSR>>
<<RA signed CSR (full PKI request)
HTTPS Request(POST)
content-type:application/x-PKCS10>>
<<evaluate CSR>>
<<create cert, sign>>
<<Cert (full PKI response)
HTTPS Response [200]
content-type:application/x-x509-user-cert>>
<<Cert (full PKI response)
HTTPS Response [200]
content-type:application/x-x509-user-cert>>
validation and acceptance
Connected Component
Registration Authority
Certificate Authority

Figure 4: Sequence diagram for secure transport elements for certificate related messages

## 8.4 Resiliency of certificate renewal process

As reflected in the overview diagrams above, the certificate renewal process must consider deny and fail conditions at both RA certificate evaluation and CA certificate evaluation. Deny conditions occur when a certificate renewal request is determined by the RA or CA to be against policy, the certificate renewal requests are no longer necessary, or similar trust management considerations. Fail conditions occur when an RA or CA cannot process a certificate renewal request because the request cannot be read or is not correctly formatted. The CMC control messages to be used to convey information for denials or failure conditions are specified in [IETF-RFC5272]. Solution specific capabilities such as MEM PCD DMC can be developed.

Not all failures are in scope of this specification at this time. This includes network and message transport failures such as dropped messages and also software failures at the Subscriber or RA. Failures of the CA are also out of scope.

## 8.5 Future improvements

Several improvements are left to further research. This includes:

- Specific mechanisms for how Connected Components will recognize that their certificate is expiring. There are several options that can be implemented at the element on which the certificate is installed and also at peers and servers.
- Detailed message encoding and format details which may be dependent upon implementation.
- Detailed use of CMC Status Controls as specified in [IETF-RFC5272] and corresponding use of controls from solution specific capabilities such as MEM PCD DMC.
- Specific limitations or controls on how RAs can modify certificate renewal requests.
- Solution specific alerts to clinicians, clinical management systems, or management entities. At least one option here is to use solution specific capabilities such as provided in MEM PCD DMC.
- An entropy requirement for key generators needs to be specified.
- A process for key issuance and transport for constrained Connected Components can be investigated.
- It may be necessary for some security or legal compliance circumstances for keys to be escrowed. If so, a process for RAs to generate and store private/public key pairs can be specified using guidance from [IETF-RFC5272]. This is a significant security risk and may be illegal in some jurisdictions and so has not been included at this time.
- The current secure transport mechanism specified is HTTPS. [IETF-RFC5272] provides options including direct use of TCP/TLS but does not specify mapping. Moreover, HL7, FIHR, and DDS may provide options as well. These alternative mechanisms can be assessed if HTTPS is difficult to implement interoperability in the future.
- Secure transport mandates use of TLS1.2 at this time. When considered sufficiently secure, TLS1.3 can be specified.
- There are many options, some solution specific, for alerts and alarms to be conveyed to clinicians, clinical management servers, and the Management Entity. If SNMP is used, OIDs need to be defined and implementation of SNMPv3 using CMI certificates can be specified.

Implementation details and support from best practice organizations (such as NIST) can also be addressed in future iterations of this specification.

## 8.6 Normative requirements for certificate renewal requests from Connected Components

These requirements apply to any Connected Component that has been issued a Subscriber certificate. Connected Components (CC):

### 8.6.1 Connected Component ME Notification Requirement

- SHALL notify the Management Entity (ME) when a certificate is recognized to be expiring
    - Alerts MAY be sent as a technical alert in MEM DMC as an OBR
    - Alerts MAY be sent as an alarm from an SNMP trap

### 8.6.2 Connected Component Renewal Process Requirement

- SHALL start renewal process when clinically safe and SHALL notify the ME
    - Certificate update notifications MAY be sent as a technical alert in MEM DMC as an OBR
    - Certificate update notifications MAY be sent as an alarm from an SNMP trap

### 8.6.3 Connected Component Rekey Requirement

- SHALL rekey for all certificate renewals – the use of the keys corresponding to the expiring certificate is prohibited

### 8.6.4 Connected Component CSR Renewal Requirement

- SHALL use full PKI request for the CSR renewal as documented in [IETF-RFC5272] (recall that [IETF-RFC5272] refers to this process as rekeying)
    - The full PKI request SHALL be formatted as either a PKCS#10 or CMRF
    - The full message SHALL be signed by the CC using the legacy private key as proof of possession  and shall include the expiring certificate in the request

### 8.6.5 Connected Component Renewed Certificate Validation Requirement

- SHALL validate that it can use the renewed certificate
    - SHALL validate the renewed certificate prior to use as specified in [CMI-SP-F-PF].
    - SHALL re-establish secure channels using the renewed certificate. If establishing those channels fails, the CC SHALL revert to the expiring certificate and SHALL send a certificate renewal alert to the ME.
        - Alerts MAY be sent as a technical alert in MEM DMC as an OBR
        - Alerts MAY be sent as an alarm from an SNMP trap
        - The CC SHALL submit a self-revocation of the failed certificate to the RA and SHALL destroy the associated keys

## 8.7 Normative requirements for Registration Authorities

These requirements apply to Registration Authorities. Registration Authorities (RA):

### 8.7.1 Registration Authority Signature Verification Requirement

SHALL verify the signature of the PKI request as specified in [CMI-SP-F-PF].

### 8.7.2 Registration Authority CSR Evaluation Requirement

- SHALL evaluate the CSR
  - SHALL determine that the new cert request is signed using a key pair other than corresponding to the expiring certificate
  - SHALL verify the CC is authorized to receive a new certificate

### 8.7.3 Registration Authority Denial or Failure Notification Requirement

- SHALL signal the requested CC of DENY or FAIL
  - CMC control messages SHALL be used to convey information for denials or failure conditions are specified in [IETF-RFC5272].
  - SHALL notify the ME
    - Certificate update DENY or FAIL notifications MAY be sent as a technical alert in MEM DMC as an OBR
    - Certificate update DENY or FAIL notifications MAY be sent as an alarm from an SNMP trap

### 8.7.4 Registration Authority CSR Approval Requirement

- If a CSR is approved
  - MAY change any fields in the CSR, including removal of the original certificate
  - MAY append additional information as necessary
  - SHALL sign the CSR as proof of authorization

## 8.8 Normative requirements for secure transport of certificate related messages

Messages containing certificate renewal requests and responses SHALL comply with [IETF-RFC5273] and the following requirements.

### 8.8.1 Certificate Renewal Certificate Management Message HTTPS Requirement

- HTTPS SHALL be used used to transport all certificate renewal certificate management messages using TLS1.2
  - Mutual authentication SHALL be performed.
  - TCP port 443 SHALL be used by default and SHALL be re-configurable

### 8.8.2 Certificate Renewal Client POST Requirement

- Clients SHALL use the POST method to submit requests
  - Content-Type SHALL be application/PKCS10 (which includes CSRF formatting)
  - Body SHALL be binary value of the encoding of the PKCS10/CSRF full PKI request (CSR)

### 8.8.3 Certificate Renewal Server Response Requirement

- Servers SHALL use the 200 response code for successful responses
  - SHALL use appropriate HTTPS headers
  - Body SHALL be the BER (Basic Encoding Rules) for full PKI response

### 8.8.4 Certificate Renewal Status Message Requirement

- DENY and FAIL status messaging SHALL use HTTPS using TLS1.2

# 9 Certificate Revocation

Certificates are revoked by the CA when there is no longer confidence in the security of the keys associated with a certificate, such as when a device has been compromised. Certificates may also be revoked by the CA when the certificate and associated keys are no longer needed, such as when a device is no longer safe to use. This raises serious concerns for clinical reliability. Of course, revocation is an absolutely necessary process to ensure the security of any system using PKI. So the challenge in the clinical environment is to enable proactive management of certificate revocation in a manner that scales well for technical staff.

There are three circumstances of how revocation may occur. These are listed below:

- Self-revocation: A Connected Component may revoke its own certificate – self-revocation. This is useful for an expiring certificate that has been renewed as described above. It is also useful when the Connected Component has reached end-of-life and is no longer useful.
- Revocation by the RA: Hospital staff or vendors may recognize that a Connected Component is no longer useful or lose confidence in the component (because of suspected or known tampering or compromise). They may then have the Registration Authority responsible for the Connected Component revoke the certificate(s) associated with the component.
- Revocation by CA: The CA may be advised by a 3rd party that a Connected Component is unsafe or has been compromised (not directly responsible for the certificates issued to a Connected Component).

Traditionally, certificate revocation is a manually intensive deliberate process executed by staff at the CA. The CA receives a revocation request (on-line, through email, or even phone) by an entity vetted a priori (e.g., they have a business relationship and have been verified by the CA). The CA then does a rather exhaustive validation that the certificate has in fact been compromised with the responsible entity. Once confident the revocation requested is valid and warranted, the CA will execute the processes to add the revoked certificate to the revocation verification systems in use (typically CRLs and OCSP servers). The deliberate process here, which can take days to weeks, ensures that revocation cannot be leveraged as a denial of service attack vector.

However, this process does not meet the need of the health industry. Revocation occurs far too slowly and does not adequately advise clinical IT staff of changes in Connected Component status. A more automated process that includes some form of alerts of revocation is necessary.

The first two cases above – self-revocation and revocation by the RA – are being requested by trusted entities directly responsible for the outcome of their request. The associated revocation request can still be signed by a current and verifiable certificate – namely, that of the Connected Component or the RA. Therefore, signed revocation requests by self (the Connected Component) or the RA can be automatically processed. The RA handles both cases and is responsible for advising a management entity so IT staff can be aware of the revocation. The RA will relay self-singed and will send RA-signed revocation requests directly to an entity (server) managed by the CA that issued the certificate. It must be emphasized that this entity is part of the CA and the CA will implement appropriate procedures are practices necessary for security processing automat revocation request to prevent misuse.

In the case where a 3rd party requests revocation, manual processes requiring human verification at the CA and RA are still required. In some cases, the CA may be compelled to revoke certificates against the objection of the responsible RA (e.g., in the case the RA has been compromised or has not executed certificate responsibilities in accordance with the guiding Certificate Policy). However, when revocations occur in this way, the CA can send appropriate notifications to the RA and the RA can send corresponding alerts to the a management entity so IT staff can again be aware of the revocation.

## 9.1 Informative sequence diagram

The three processes discussed above are illustrated in the following figures.

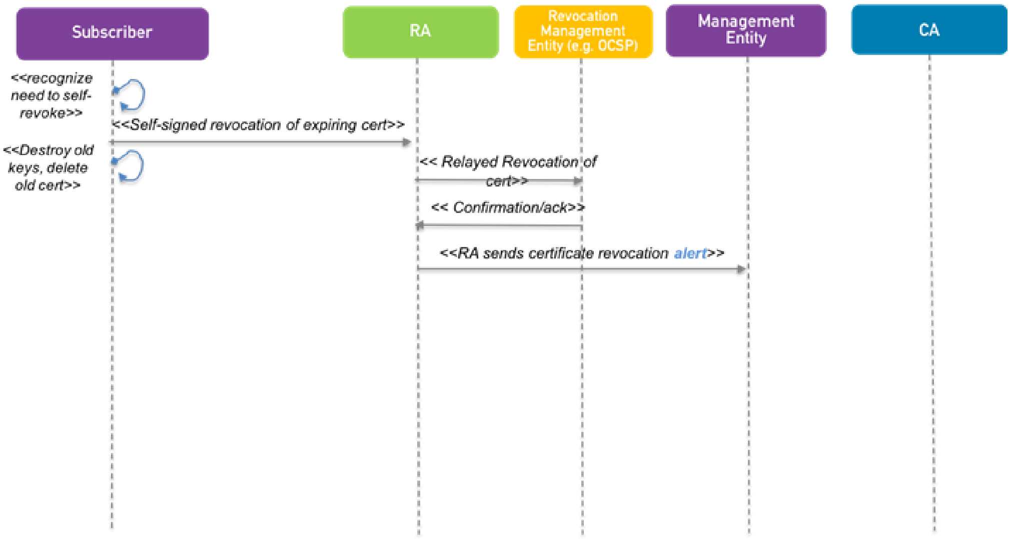
Subscriber
RA
Revocation Management Entity (e.g. OCSP)
Management Entity
CA
<<recognize need to self-revoke>>
<<Self-signed revocation of expiring cert>>
<<Destroy old keys, delete old cert>>
<< Relayed Revocation of cert>>
<< Confirmation/ack>>
<<RA sends certificate revocation alert>>

Figure 4: Automated certificate self-revocation with alerting

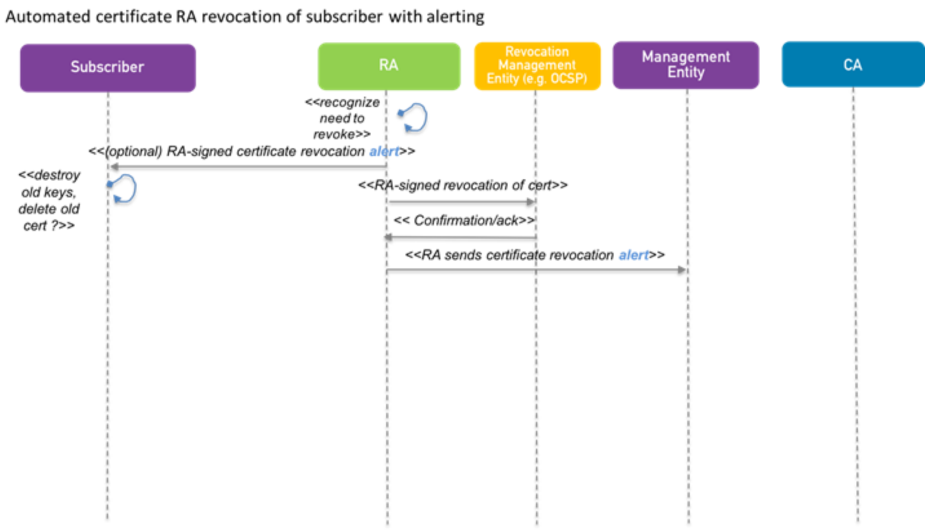
Automated certificate RA revocation of subscriber with alerting
Subscriber
RA
Revocation Management Entity (e.g. OCSP)
Management Entity
CA
<<recognize need to revoke>>
<<(optional) RA-signed certificate revocation alert>>
<<destroy old keys, delete old cert ?>>
<<RA-signed revocation of cert>>
<< Confirmation/ack>>
<<RA sends certificate revocation alert>>

Figure 5: Automated certificate RA revocation of subscriber with alerting

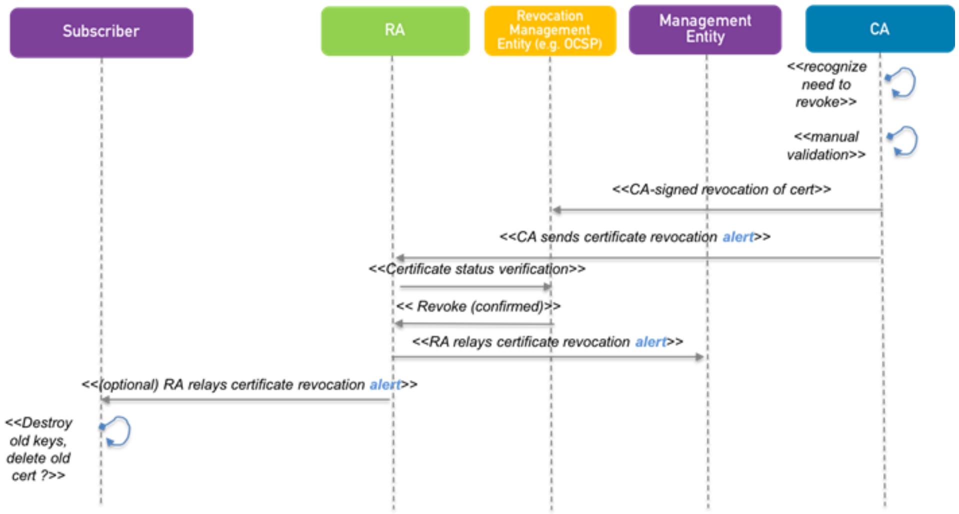
Subscriber
RA
Revocation Management Entity (e.g. OCSP)
Management Entity
CA
<<recognize need to revoke>>
<<manual validation>>
<<CA-signed revocation of cert>>
<<CA sends certificate revocation alert>>
<<Certificate status verification>>
<< Revoke (confirmed)>>
<<RA relays certificate revocation alert>>
<<(optional) RA relays certificate revocation alert>>
<<Destroy old keys, delete old cert ?>>

Figure 6: Certificate revocation by third party alert of subscriber with alerting

## 9.2 Normative Requirements

### 9.2.1 CA/RA Certificate Revocation Requirement

- Revocation SHALL be performed by the CA or RA in accordance with [IETF-RFC5280] and [IETF-RFC6960].

### 9.2.2 CA/RA Revocation Signaling Requirement

- The CA SHALL signal RAs and RAs SHALL signal MEs of revocations
  - CMC control messages SHALL be used to convey information for revocations as specified in [IETF-RFC5272] between CA and RA

- Certificate revocation notifications MAY be sent as a technical alert in MEM DMC as an OBR by RAs to MEs
- Certificate revocation notifications MAY be sent as an alarm from an SNMP trap by RAs to MEs

# 10 Connected Component Profile Framework

A Connected Component Profile (CCP or "profile") provides a mechanism for components to metadata to support automated compatibility recognition, protocol negotiation, and smooth communications. This profile is a machine-readable description of a component and its capabilities and is exchanged at run-time between connected components in various scenarios. For example, when a Client first connects with a Client Management Entity, the Client sends its profile, and the management entity responds with its own, enabling automated verification of communication compatibility and (potential) fallback to a mutually supported protocol.

Aside from enabling runtime communication of connected components, the CCP could also be used for other purposes, such as enabling automated testing as part of the certification process, or for procurement purposes, where a profile succinctly summarizes a component's capabilities.

This section defines the Connected Component Profile Framework, containing general requirements applicable to any use of a profile. Connected Component Profile Solutions are requirements implementing CCPs using specific protocols such as [IHE-PCD] and will be documented in other specifications.

## 10.1 Connected Component Profile Contents

The metadata associated with a connected component could be quite large. For efficiency, the profile is split into a Minimum Connected Component Profile (MCCP or "minimum profile"), which contains the elements needed for baseline interoperability, a Full Connected Component Profile (FCCP or "full profile"), which contains all other associated metadata. The MCCP is always exchanged when two components attempt communication; the MCCP contains a link to the CCP for run-time querying as needed.

This section defines the profile's contents, but its structure and serialization formats are defined in other specifications.

### 10.1.1 Minimum Connected Component Profile Contents

#### *10.1.1.1 MCCP Format Version*

The MCCP Format Version is included to ensure future proofing of the MCCP and SHALL be a three digit decimal number between "001" and "999". Connected Components identified in accordance with this release SHALL use version string "001".

#### *10.1.1.2 Connected Component Identifier*

This field uniquely identifies a Connected Component. Its format SHALL be equivalent to the Identifier defined in Section 5.3.1 of this specification. (While present in a component's security certificate, the identifier is also part of the MCCP for convenience at runtime.)

#### 10.1.1.3 Release Bundle Version

A *Release Bundle Version* (RBV) indicates which version of the CMI architecture a connected component complies with. It is that critical for components to communicate this information, because version differences may affect components' ability to communicate and might imply negotiation or compatibility mediation must take place.

The RBV SHALL use a "MAJOR.MINOR.PATCH" format. A new patch version indicates clarifications or corrections have been made, while a new major or minor version indicates requirements have been modified. A new patch or minor version indicates maintained interoperability, while a new major version indicates interoperability cannot be guaranteed. This approach aligns with standard semantic versioning practices (i.e. [SEMVER-2.0.0]).

Connected Components identified in accordance with this release SHALL be "1.0.0". In future iterations, the RBV may be specified in a separate document covering specification releases, versioning, and governance. Specification releases will be released as a self-consistent bundle, accompanied by a RBV, and Connected Components conforming to a given release bundle will use the RBV associated with that bundle.

#### 10.1.1.4 Current Component Status

This field indicates the current status of the Connected Component, such as "Operational" or "Graceful Shutdown in Progress". A full set of status codes is left to future iterations of this specification.

#### 10.1.1.5 Software Version

The Software Version of a Connected Component SHALL use a "MAJOR.MINOR.PATCH" format as defined in [SEMVER-2.0.0]. It is possible that future iterations of this specification will have software version be part of configuration data instead of the MCCP.

#### 10.1.1.6 Make and Model

The Make and Model a Connected Component SHALL each be a vendor-defined string. It is possible that future iterations of this specification will have make and mode be part of the full profile instead of the MCCP.

#### 10.1.1.7 Full Profile URI

The Full Profile is resolvable using the Full Profile URI. The format of the URI and mechanism by which it is resolved is left to future iterations of this specification. It is likely future iterations will specify the Full Profile URI to be a URL such as https://examplevendor.com/make_model_xyz.

#### 10.1.1.8 Configuration Data URI

Connected Component Configuration Data is resolvable using the Configuration Data URI. The format of the URI and mechanism by which it is resolved is left to future iterations of this specification.

##### 10.1.1.9 Extension for Optional Information

Extensions for optional information are allowable and will be more defined in future iterations of this specification.

### 10.1.2 Full Connected Component Profile Contents

The full profile contains metadata consistent for a type of Connected Component (e.g. for a given make and model of medical device). Examples of metadata that could be included in a full profile include:

- Manufacturer
- Component Type Identifier
- Supported Output Data Set
- etc.

The contents of the Full Connected Component Profile are left to future iterations of this specification. It is likely that the full profile will support the FDA's Software Bill of Materials [FDA-SBOM-1].

### 10.1.3 Configuration Data Contents

While two Connected Components may have identical full profiles, they might be configured differently. A Connected Component's Configuration Data contains this instance-specific data. Software version is one example of configuration data. Configuration data content requirements are left to future iterations of this specification.

## 10.2 Connected Component Profile Communication

### 10.2.1 General MCCP Communication

Any time a Connected Component establishes or reestablishes a secure connection on a CMI-defined interface, it SHALL send its MCCP as its first message, and the recipient SHALL respond with its MCCP.

Connected Component Profile solutions may allow MCCPs to be requested after a secure connection has been established.

### 10.2.2 Profile Updates

Any time a Connected Component's MCCP content changes, it SHALL send its MCCP to any recipient it has a secure connection with (and the recipient SHALL respond with its MCCP).

### 10.2.3 MCCP Format Negotiation

If an MCCP recipient supports the sender's MCCP format version, the recipient SHALL send its MCCP using that version. If an MCCP recipient does not support the sender's MCCP format version, the recipient SHALL respond with a list of its supported MCCP format versions. If the sender can support any of the recipient's supported MCCP format versions, the sender SHALL respond with its MCCP using the highest mutually supported format version.

### 10.2.4 Future Work (Informative)

In future iterations of this specification, CMI may digitally sign certified components' CCPs. Were that the case, a component would always sends its profile when first communicating across a CMI-specified interface, but the presence of a digital signature verifies the profile is accurate and the component has been certified.

In future iterations of this specification, a Connected Component such as a device gateway may serve as a proxy for multiple components, in which case it would send its own MCCP and that of the components it proxies.

## 11 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 Goeringer** authored this document. Special thanks to those who were directly involved a variety of discussions, reviews and input: **Stuart Hoggan, Jeffrey Brown, Eldon Metz, Guy Johnson** and **Dr. Massimiliano Pala.**

**Chris Riha** is the CMI Lead for this document. This document was primarily discussed and reviewed within The Center's Security Working Group, Working Group, with additional input from the Architecture & Requirements and Connectivity Working Groups. The part-time and full-time working group participants, additional offline reviewers, and their affiliations are listed below:

| Working Group Participants | Company Affiliation |
|---|---|
| **Aishwarya Muralidharan** | vTitan |
| **Alex Poiry** | Cerner |
| **Ali Nakoulima** | Cerner |
| **Andrew Meshkov** | 86Borders |
| **Brian Long** | Masimo |
| **Brian Scriber** | CableLabs |
| **Bruce Friedman** | GE Healthcare |
| **Corey Spears** | Infor |
| **Darshak Thakore** | CableLabs |
| **David Hatfield** | Becton Dickenson |
| **David Niewolny** | RTI |
| **Eldon Metz** | Innovision Medical |
| **George Cragg** | Draeger |
| **Guy Johnson** | Zoll |
| **Ian Sherlock** | Texas Instruments |
| **James Surine** | Smiths-Medical |
| **Jason Mortensen** | Bernoulli Health |

| Working Group Participants | Company Affiliation |
|---|---|
| **Jay White** | Laird |
| **Jeffrey Brown** | GE |
| **JF Lancelot** | Airstrip |
| **John Barr** | CableLabs |
| **John Hinke** | Innovision Medical |
| **John Williams** | FortyAU |
| **Kai Hassing** | Philips |
| **Ken Fuchs** | Draeger |
| **Logan Buchanan** | FortyAU |
| **M Prasannahvenkat** | vTitan |
| **Massimo Pala PhD** | CablelLabs |
| **Mike Krajnak** | GE |
| **Milan Buncick** | Aegis |
| **Neil Puthuff** | RTI |
| **Neil Seidl** | GE |
| **Ponlakshmi G** | vTitan |
| **Scott Eaton** | Mindray |
| **Stefan Karl** | Philips |
| **Travis West** | Bridge Connector |

- Sumanth Channabasappa (Chief Architect), Steve Goeringer (Security Architect), Chris Riha (Working Groups Lead), Paul Schluter, Bowen Shaner, Jacob Chadwell, David Fann, Spencer Crosswy, Dr. Richard Tayrien, Trevor Pavey; and, Ed Miller (CTO) - The Center