Cooperative-ITS Credential Management System Functional Analysis and Recommendations for Harmonization

Document HTG6-4

Version: 2015-09

EU-US ITS Task Force
Standards Harmonization Working Group
Harmonization Task Group 6

Harmonization Task Group 6 Membership	
Gianmarco Baldini	European Commission's Joint Research Centre
William Ball	Merriweather Advisors
Claire Barrett	US Department of Transportation
Norbert Bißmayer	Fraunhofer, SIT
Dominie Garcia	Booz Allen Hamilton
Dawn LaFrance-Linden	US Department of Transportation
Tom Lusco	Iteris
Vincent Mahieu	European Commission's Joint Research Centre
Catherine McGhee	Virginia Department of Transportation
Robert Rausch	Transcore
William Whyte	Security Innovation
Harmonization Task Group 6 Leadership	
Knut Evensen	Q-Free, European Commission
Peter Girgis	Transport Certification Australia
Wolfgang Höfs	European Commission DG Communication Networks, Content and Technology
Suzanne Sloan	US Department of Transportation
Steve Sill	US Department of Transportation

Contents

1 Introduction 5
1.1 Format of HTG6 Results ..... **Error! Bookmark not defined.**
2 Purpose of this Document 7
2.1 Background 7
2.2 Issue and Opportunity 8
2.3 Structure of the Analysis 9
2.4 Summary Observations and Conclusions 9
2.5 Recommended Future Actions: 11
3 Analysis 12
3.1 Objectives 12
3.2 Audience and Scope 13
4 Methodology 14
4.1 Define a Set of Minimal Functional Components for a CCMS 14
4.1.1 C-ITS Security Environment 15
4.1.2 End Entity Lifecycle 15
4.1.3 CCMS Components 16
4.1.4 Use cases 18
4.1.5 CCMS Component Details: Functions, Interfaces and Trust Chain Elements 20
4.2 Consider End Entity Lifecycle Requirements 60
4.2.1 Handling Credentials 61
4.2.2 Summary of End Entity CCMS Interfaces 63
4.3 Examine Existing Designs 64
4.3.1 PRESERVE 64
4.3.2 CAMP SCMS 65
4.4 Identify Interfaces between Interacting CCMS 66
4.4.1 Information Flow Diagrams for Inter-CCMS Communications 66
4.4.2 CCMS Coordination 69
4.4.3 CCMS Federation Scenarios 74
5 Conclusions and Recommendations 79
6 References 82
7 Terminology and Definitions 83

Figures

Figure 1: Scope of CCMS Operations 13
Figure 2: End Entity Security Status 16
Figure 3: CCMS Components 18
Figure 4: CCMS Provisioning 22
Figure 5:End Entity Enrollment 25
Figure 6: End Entity Authorization 35
Figure 7: Misbehavior Reporting and Action 44
Figure 8: Revocation 53
Figure 9: Application and Device Lifecycle 62
Figure 10: End Entity Interfaces 63
Figure 11: Baseline Analysis of PRESERVE PKI System 64
Figure 12: CCMS Baseline Analysis of CAMP SCMS 65
Figure 13: Interfaces that support Inter-CCMS Trust 67
Figure 14: Interfaces that support Inter-CCMS Trust (Continued) 68
Figure 15: Two CCMS trust each other on registration 76
Figure 16: Two CCMS trust each other on 77
Figure 17: Two CCMS trust each other on pseudonym certificate level 78

Tables

Table 1: Use Cases of the CCMS and End Entity Lifecycle 19
Table 2: Component Diagram Legend 20

1 Introduction

Harmonization Task Group 6 (HTG6) began in early 2014 as a cooperative effort between the European Commission, Transport Certification Australia, and the United States Department of Transportation to address multiregional Cooperative Intelligent Transportation Systems (C-ITS; also known as "Connected vehicle" or "CV" in the US). The focus of HTG6 was to develop a security policy framework for C-ITS collaboratively. The HTG6 work included:

- Identifying those policies and decisions that, if harmonized, offer significant public benefit;
- Analyzing the technical and policy/management elements of trust models associated with communications security systems; and
- Developing consensus on policy options that effectively and beneficially result in coordinated management policies and security approaches to the extent possible for C-ITS.

The body of work produced by HTG6 members presents the maximum set of common policy approaches with justification for the benefits for commonality. The work further recognizes policies and approaches that can differ regionally without impact.

HTG6 is comprised of acknowledged hands-on policy and technical experts who are deeply involved in policy analysis, C-ITS implementation, and security-related fields. HTG6 work was also informed by hands-on technical researchers, standards developers, communications engineers, certification experts, and original equipment manufacturers. The ability to engage with external experts was performed as a measure to ensure that the results can be viewed credibly by the broader standardization community.

1.1 Format of HTG6 Results

HTG6 results are presented in a series of reports. The primary reports include:

- **Executive Summary (HTG6-1).** This document is a high-level summary of the key results.
- **Summary of Results (HTG6-2).** This report summarizes the results across the body of HTG6 work for policy and decision makers.
- **Architecture Analysis (HTG6-3)**. This report identifies the primary elements of a Public Key Infrastructure (PKI) security system and compares these fundamental needs against four security architectures:
    - The EC's Joint Research Centre PKI for secure and confidential commercial vehicle digital tachograph regulation (an operational system)
    - TCA's Gatekeeper PKI for secure and confidential commercial vehicle regulation (an operational system)
    - EC's PRESERVE PKI architecture design for secure, authenticatable communications

- US's Security Credential Management System (SCMS) architecture design for secure, private, and authenticatable communications.

The comparative analysis yielded an understanding of the fundamental elements of a C-ITS Credential Management System (CCMS) that (a) are highly recommended for harmonization; (b) are recognized as beneficial if harmonized; or (c) do not require harmonization. The conclusions in this report highlight the areas for harmonization, describe them, and assign priority levels to harmonization decisions.

- **Functional Decomposition Analysis (HTG6-4; this document).** This report further analyzes the recommended areas that are identified as "highly beneficial" for harmonization in HTG6-3. To come to these results, the team decomposed a CCMS architecture and identified the interfaces and data flows where actions are needed to achieve harmonization. In many instances, the harmonization action requires a technical solution to establish inter-CCMS or intra-CCMS trust. In some instances, our team recognized that harmonization of language is also needed. If devices can communicate their level of security in manner that is clear and consistently defined, the actual devices or other technical harmonization actions are not needed.
- **Organizational Analysis (HTG6-5).** This report identifies how policies need to be harmonized to support trust models—both among the various entities that comprise a jurisdictional CCMS (intra-CCMS) as well as between jurisdictional CCMS (inter-CCMS). This report also describes the requirements for communication of those policies as a basis for harmonizing inter-CCMS and intra-CCMS trust, particularly across jurisdictional boundaries.
- **Risk Management Framework (HTG6-6)**. This report describes a process for implementers to identify and categorize their risks. This process leverages existing risk categorization processes from the National Institutes of Standards and Technology (NIST) and Common Criteria. The results support decision makers in identifying appropriate technical and policy controls to include in a CCMS architecture to mitigate or address risk. This report further identifies, at a high-level, some of the gaps that are not addressed by NIST/CC but that are needed for a cooperative security environment. Further analysis is needed to fully identify gaps.
- **Background Documents**. HTG6 also produced background documents to accompany these key deliverables. These documents include:
  - **PKI Primer (HTG6-7)**
  - **Primer on the Intelligent Transport Systems (ITS) Station concept and the Connected Vehicle Reference Implementation Architecture (CVRIA) (HTG6-8)**
  - **Glossary (HTG6-9)**

2 Purpose of this Document

The goal of HTG6 is to facilitate successful implementation of any jurisdictional C-ITS security system seeking to harmonize with adjacent systems by presenting a C-ITS security policy framework. This document, **HTG6-4: C-ITS Credential Management System Functional Analysis and Recommendations for Harmonization,** describes the functional components of a PKI system as they relate to C-ITS, and describes the functionality involved in enabling devices from different CCMS entities (either inter-CCMS or intra-CCMS) to trust one another (trust chain elements). In order to properly analyze the roles a CCMS must take, the HTG6 team decomposed the CCMS and its functions, the end entity devices and end entity applications, the lifecycle of those end entities, and requirements for access to resources in the ITS environment. The decomposition allowed the HTG6 team to:

- Describe four different trust models that can be implemented for a CCMS;
- Identify the key interfaces at which actions can be taken to establish trust through sharing of policies or by communicating levels of required security, privacy, or accessibility to critical resources that are allowed;
- Ascertain where there is a lack of specifications and recommend the development of future standards;
- Identify the role of compliance assurance or certification of applications and devices; and
- Identify where harmonized organizational policies play a key role.

2.1 Background

Any communications system needs to provide a mechanism to allow communicating partners to trust each other. In large systems, this is typically accomplished by cryptographic protection for individual communications, along with cryptographically secured credentials and a centralized credential management system with responsibility for ensuring that credentials are issued only to parties that are entitled to them. Each credential management system typically has a small number of nodes that serve as trust anchors, which can make statements themselves about the trustworthiness of end entity nodes or delegate the ability to make trust statements to other management nodes.

One typical way of instantiating a credential management system is via public key infrastructure (PKI). Currently in the C-ITS world there are at least two projects underway to build a C-ITS Credential Management System based on PKI: one in the US and one in Europe. These systems use slightly incompatible technical approaches and have not, to date, coordinated on exactly what criteria are used to determine that a device is trustworthy enough to be issued credentials. However, since the modern car market is global, it is likely that, at some point, devices authorized by one CCMS will have to interact with devices authorized by other CCMS. In addition, as CCMS need to be upgraded, the future CCMS will

need to be able to handle both existing and future devices and future devices will need to interact both with the original CCMS and with the newly implemented CCMS.

To establish an overview of the functionality involved in making it possible for devices from different CCMS to trust each other, the analysis described in this document considers the following questions:

- What functions may be required of a CCMS?
- What data must be exchanged between functions when users interact with the CCMS (end entities, or devices carrying and using crypto-material)?
- What data might be exchanged between CCMS?
- What functions exist in today's CCMS designs? What are the implications of trust levels between CCMS?
- Any CCMS must be able to adapt, to evolve to changing requirements induced through technological change or policy. How must the CCMS be constructed so as to enable this evolution?
- Similarly, any cryptographic algorithm may be made obsolescent, through technology change (e.g., quantum computing) or hostile action (e.g., finding a weakness in the algorithm and exploiting it). Any CCMS must be capable of changing the underlying cryptographic algorithms it uses. This concept is termed *crypto-agility*.

2.2 Issue and Opportunity

Public Key Infrastructure is the security solution widely identified as uniquely suited to the C-ITS environment and both known designs for C-ITS Credential Management Systems (CCMS) use it. However, adopting a particular PKI security solution for one CCMS, or even each CCMS, does not inherently provide a mechanism for devices of one CCMS to trust communications from devices native to another CCMS. The degree to which CCMS implement similar functionality and share information is dependent on and depends on the degree to which CCMS 'trust' one another.

Fundamentally this concept of inter-CCMS trust implies organizational trust between the entities managing the CCMS, but technically such trust is reflected in digital interactions between the CCMS. In the absence of a mandate for CCMS implementation, the C-ITS environment may end up with any number of CCMSs, which, for CCMS that do trust one another, leads to a potential for a great many inter-CCMS interfaces. It is crucial to future interoperability and extensibility that implementers understand the repercussions of inter-CCMS trust to inform their implementation decisions. To date,

there is little such summary information.

2.3 Structure of the Analysis

To determine the requirements involved in making it possible for devices from different CCMS to trust each other, and recommend actions for CCMS implementers, the analysis begins by defining a set of minimal functional components for a CCMS. Each functional component carries out a minimal set of operations, i.e. a set of operations such that two operations that do not need to be carried out by the same functional component are carried out by different components. In general, a minimal functional component as defined here has responsibility for maintaining a single set of data to support a single set of related operations. Since end-entities interact with CCMS in different ways at different phases of their lifecycle, that lifecycle is considered in this analysis.

The next step is to compare the required functionality identified in the first part of this analysis to actual emerging CCMS designs. This allows the functionalities for each design to be categorized as designed, partially identified, or not identified. Potential gaps in the CCMS designs can then be identified.

To address the different roles a CCMS must take, the analysis then identifies interfaces between interacting CCMS based on the level of trust between them, and considers access to C-ITS resources that may affect, or be affected by, security decisions.

Finally, the conclusions from this analysis are presented, along with recommendations intended to facilitate the successful implementation of any jurisdictional C-ITS security system seeking to interact with another.

2.4 Summary Observations and Conclusions

The functional analysis of the CCMS led to several additional conclusions about the nature of CCMS:

- Existing CCMS designs incorporate much of the required functionality of a comprehensive CCMS. However, neither surveyed design has addressed many of the inter-CCMS communications aspects identified in this analysis. Neither has the standards community described the relevant interfaces. ***This lack of interface specification is among the biggest issues confronting CCMS designs and the long term C-ITS deployment.***
- Equally important, ***device and application certification processes, whatever they may be, are linked with credential management, and as such must be considered concurrently with the architecting of security management systems and procedures.*** There are other issues however, such as ***resource management, CCMS policy management, and CCMS scoping,*** all of which are summarized in HTG6-5: Organizational Analysis.
- A CCMS operates in a three-dimensional environment: a CCMS's area of responsibility is defined

by geo-political boundaries, time, and the applications it supports. See section 2.2 (Audience and Scope) for an illustration of this concept.

- Given disaster recovery and resultant responses and upgrades, ***a CCMS must be defined to exist in an environment where it must communicate with at least one other CCMS***: the original and a CCMS implemented as part of disaster recovery, or as part of a system upgrade. See section 3.1 (Define a Set of Minimal Functional Components for a CCMS) for a brief discussion.
- Since a CCMS' components interact with end entities at different phases of the end entity's lifecycle, ***CCMS that have different institutional structures are possible***, ***so long as each independently managed component fulfills its requirements***, including interfaces to other components in that CCMS. See the section 4.1.2 (End Entity Lifecycle) for consideration of the end entity lifecycle in terms relative to the CCMS.
- ***There are aspects of the end-entity lifecycle that do not involve direct interaction between the end entity and the CCMS, but about which the CCMS must be informed, and may generate secondary interfaces between CCMS and other systems***. See section 4.2 (Consider End Entity Lifecycle Requirements) for aspects of the lifecycle that affect other related processes and procedures (such as certification or conformance testing) that may affect CCMS behavior but are not direct interactions between CCMS and end entities.
- ***CCMS components that manage policy must share their policies with end entities and also with other CCMS with which they have a trust relationship***. See section 4.1.5 (CCMS Component Details: Functions, Interfaces and Trust Chain Elements) for illustrations of policy sharing between CCMS and end entities, and section 4.4 (Identify Interfaces between Interacting CCMS) for a discussion of related inter-CCMS issues.
- ***There*** **are several distinct degrees to which CCMS may trust one another; establishment and maintenance of this inter-CCMS trust is contingent on the exchange of policy information**. See section 4.4.3 (CCMS Federation Scenarios) for a discussion of various inter-CCMS trust relationship possibilities.
- The long term viability of a given CCMS is dependent on its ability to evolve, in particular with regard to changes in cryptographic processes. Related, the *crypto-agility* of its supporting end entities much support such changes. ***Hardware-based crypto-processors should be discouraged, or designed so as to be easily replaced or upgraded***. Otherwise, end entities could be made permanently insecure in the event of a cryptographic failure.

2.5 Recommended Future Actions:

1. Most inter-CCMS interfaces are 1-to-many, so as the number of CCMS increase, the number of implemented interfaces increases in polynomial fashion. To mitigate against this complexity, ***the C-ITS community should minimize the number of CCMS*.**

2. There are a large number of inter-CCMS interfaces. In order to ensure consistency and scalability, the ***C-ITS community should develop standards to codify the interfaces between CCMS*.**

3. Certification of applications and devices, including the ways that ITS resources are accessed, is linked with the granting of digital credentials. Credential distribution and certification are distinctly different functions that may benefit from different operational, institutional structures, but since they are linked the requisite relationships are important to the overall success of C-ITS. ***A future harmonization activity should clarify the relationship between certification and credentials distribution.*** Absent harmonization, each implementer will have to perform this analysis, making inter-CCMS trust unlikely.

4. Devices and applications have a lifecycle, and in some phases of that lifecycle they have no relationship to the CCMS, i.e., prior to provisioning and subsequent to end-of-life. There are security-related policy concerns with how these end entities are handled in the times when they do not have a CCMS relationship. ***The C-ITS community should establish requirements defining how end entities must be developed and handled in order to obtain credentials, and establish shared requirements for proper disposal*.**

5. CCMS Components are responsible for managing the policies governing their operation, and disseminating that policy information to users. This sounds self-evident, but what it means is that the institutional framework governing the CCMS component is affected by and affects the technical design. Making changes to one or the other has a causative effect on the other. This also implies something about how CCMS enter in trust relationships with one another. Depending on how much CCMS' trust one another, their policy frameworks must be compatible, and may need to be identical. The ripple effect here is that if one CCMS changes policy, it jeopardizes the trust relationship between CCMS. ***The CCMS management community should collectively define CCMS policy frameworks.***

6. There are several likely-viable Inter-CCMS trust scenarios, and they have significantly different implications for the architecture and management of CCMS. ***Inter-CCMS trust scenarios need additional study.***

3 Analysis

This document describes the security systems required in the C-ITS environment, from a functional perspective, and how those systems interact with the end entities they secure. While an understanding of currently architected C-ITS credential management systems informed this analysis (from HTG6-3 report), what is presented here is a functional decomposition of a C-ITS Credential Management System and its functional internal and external interfaces. It is NOT a particular architecture or design, nor is it intended to imply one.

Through the course of this document, the following questions are considered:

- What functions may be required of a CCMS?
- What data must be exchanged between functions when users interact with the CCMS (end entities, or devices carrying and using crypto-material)?
- What data might be exchanged between CCMS? This considers:
    - When end entities interact, and
    - When CCMS' need to change something about the nature of their relationship with one another
- What functions exist in today's CCMS designs? (Current CCMS designs considered are the Crash Avoidance Metrics Partnership (CAMP) SCMS in the United States [8], [9], and the PRESERVE PKI in Europe [2], [3].)
- What are the implications of trust levels between CCMS?

In order to properly analyze the roles a CCMS must take, we must consider both the CCMS, the end entity devices and end entity applications.

3.1 Objectives

This report has these objectives:

- Define a set of minimal functional components for a CCMS;
- Analyze against lifecycle requirements;
- Examine existing CCMS designs;
- Identify interfaces between interacting CCMS or CCMS entities; and
- Describe a set of trust models and identify how harmonized policies that govern specific interfaces support different levels of trust.

3.2 Audience and Scope

This document is intended for anyone considering the development or operation of a CCMS. It may also be relevant to policy advisors, as many of the issues surrounding inter-jurisdictional cooperation reflect directly into inter-CCMS interactions. It focuses on the elemental functions that are needed to satisfy CCMS needs, based on current research. It does not include any formal needs or requirements statements, nor does it consider the internal functions needed as part of internal system operations, e.g. secure timing. The focus is on:

1. The delivery of services to end entities, including an understanding of what information needs to be communicated between CCMS and end entities;
2. The CCMS external interfaces that are required to facilitate inter-CCMS coordination; and
3. The CCMS internal interfaces that support the external interfaces defined in (1) and (2).

CCMS operational scope needs to be further considered across three axes (see below):

1. The geo-spatial area over which CCMS-granted credentials are relevant. This might be a political boundary, but will always be a geographic boundary (though unbound, e.g., all of earth, is possible if unlikely).
2. The time domain over which the CCMS operates and CCMS-granted credentials are relevant. While credentials of all types have a valid period from time X1 to time X2, a CCMS operating period, the time during which its functionality and interfaces are active, will be from time Y1 to Y2. While X1 cannot by definition be earlier than Y1, it is possible that X2 would be later than Y2 (in the case of a disaster event for example).
3. The application space in which CCMS-granted credentials are used. A given CCMS might supply credentials for only a limited subset of the applications functional at a particular time in a particular geo-political area.

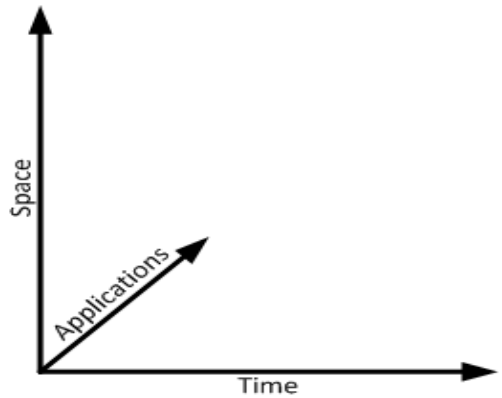


Figure 1: Scope of CCMS Operations

4 Methodology

The methodology for this functional analysis consists of these main tasks:

- **Define a set of minimal functional components for a CCMS**: In the absence of a mandate for CCMS implementation, the C-ITS environment may end up with any number of CCMS. It is crucial to future interoperability and extensibility that implementers have this information to inform their choices. To date, there is little such summary information.

- **Analyze against end entity lifecycle requirements:** The lifecycle of computer systems must be considered when performing any security analysis. The lifecycle of the end entity, be it mobile device, on-board equipment, or field equipment, must be part of the analysis. End entities interact with CCMS in different ways at different phases of their lifecycle.

- **Examine existing CCMS designs:** These designs do not illustrate all possible, likely, or practical CCMS implementations. Between them they share a great deal of functionality, but as will be seen in the bullet point below, both ignore certain likely necessary functions.

- **Identify interfaces between interacting CCMS:** Depending on the applications and the regulatory environment a CCMS must support, it may have different functions. The number of CCMS and their relative scopes will dictate the interfaces between CCMS. Previous published analyses and designs have largely ignored the concept of interacting CCMS, so the identification of these interfaces is a new thing, and one that may affect designs.

- **Describe a set of trust models**: These descriptions help identify how harmonized policies need to be applied to specific interfaces to support different levels of trust.

4.1 Define a Set of Minimal Functional Components for a CCMS

The aim of this functional analysis is to clarify information flows within and between CCMS. Each functional component identified in this decomposition carries out a minimal set of operations, i.e. a set of operations such that two operations that do not need to be carried out by the same functional component are carried out by different components. In general, a minimal functional component as defined here has responsibility for maintaining a single set of data to support a single set of related operations.

The functional components identified carry out specific roles within the credential management lifecycle. For correct and successful implementation, CCMS will also need a number of support services, including but not limited to:

- Secure timing;
- Secure storage;
- Trusted computing / processing of crypto material;
- Tamper protection;
- Reliable network connections; and
- Disaster recovery.

There may be other support services; this paper does not include an analysis of all the support functions that might be necessary. That said, the quality of these support services will affect the ability of different CCMS and CCMS components to trust each other. Just as within a CCMS there must be minimum performance requirements for these support services, there must also be minimum performance requirements harmonized between CCMS. We will not further address these harmonized minimum performance requirements, but they should be borne in mind by any implementer.

4.1.1 C-ITS Security Environment

The foundational element of any crypto-system is the functionality that enables security processes, namely the system that serves as trust anchor and the basis for crypto-processes such as trust verification, integrity protection, encryption, etc.

The C-ITS environment requires a foundational trust element that serves these needs: it must, at minimum, provide crypto-material that enables trust, both in the contents of messages, and the protection of data from unintended readers. The chosen solution depends on public-key infrastructure (PKI); however the systems currently under development in the US and the EU are somewhat different in their approach. Since the modern car market is global, and since the operable systems may indeed be different in at least two political environments, an understanding of just what the implications of differing trust anchors is warranted.

For the purpose of this analysis, the foundational trust anchor is referred to as a Cooperative ITS Credential Management System, or CCMS. At minimum the CCMS serves as root trust authority and provider of security credentials.

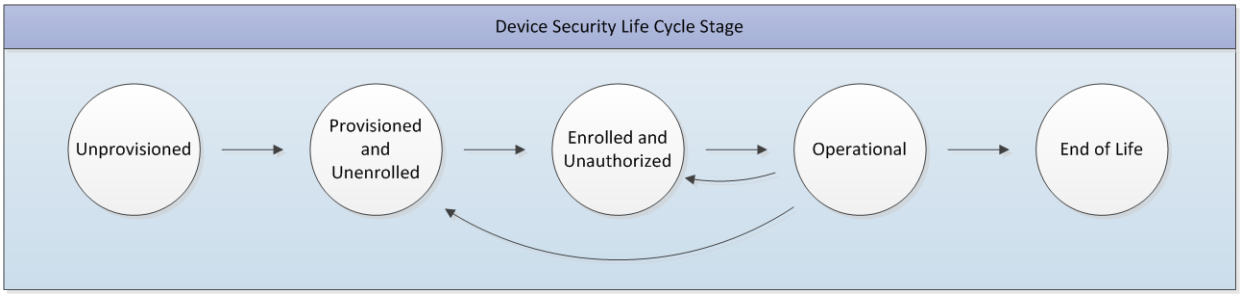
Applications and devices have a changing set of relationships depending on the lifecycle stage, as shown in 2 and 3.

4.1.2 End Entity Lifecycle

In order to frame the discussion of how the CCMS interacts with end entities, it is necessary to define the stages through which an end entity passes. A simplified version of the end entity security lifecycle is

depicted in Figure 2.

Figure 2: End Entity Security Status



Unprovisioned: The device does not have any of the crypto material or certificates necessary to interact with any parts of the CCMS other than the Provisioning components. Since the end entity is not part of the C-ITS at this stage, it cannot interact in trustworthy fashion with other end entities.

Provisioned and Unenrolled: The device has the crypto material and root certificates necessary to communicate with Enrollment components. At this stage the end entity is still not part of the C-ITS and cannot in trustworthy fashion interact with other end entities.

Enrolled and Unauthorized: The device has all the material it needs to communicate with Authorization components. It still cannot interact with other end entities in trustworthy fashion.

Operational: The device has all the material it needs to communicate with the Misbehavior components, Revocation components, and other operational end entities.

End-of-Life: The device is unable to communicate with any component of the CCMS or other end entities.

For a more detailed analysis of the end entity lifecycle, see Section 4.2. For a detailed explanation of mechanisms for provisioning/bootstrap, certification and type approval, see [10]**.**

4.1.3 CCMS Components

A CCMS has up to eight types of components:

1. <Root AuthZ/AuthN>: Establishes trust framework and serves as foundational trust authority for all elements of the CCMS and for all end entities. Responsible for managing trust relationships with other CCMS, and making that information available to other components of the CCMS (and by extension, end entities).

2. <Intermediate AuthZ/AuthN>: Underneath with Root, provides identical functionality but

potentially restricted to a smaller range of time / space / applications. Provides barrier between Root and CCMS components, and also enables scalability of root functions.

3. <Components: AuthZ/AuthN>: Underneath Intermediate, provides trust facilities to particular set of components.

4. Provisioning components provide the end entity with material that allows the end entity to move from the Unprovisioned state to the Unenrolled state.

5. Enrollment components provide the end entity with material that allows the end entity to move from the Unenrolled state to the Unauthorized state.

6. Authorization components provide the end entity with material that allows the end entity to move from the Unauthorized state to the Operational state.

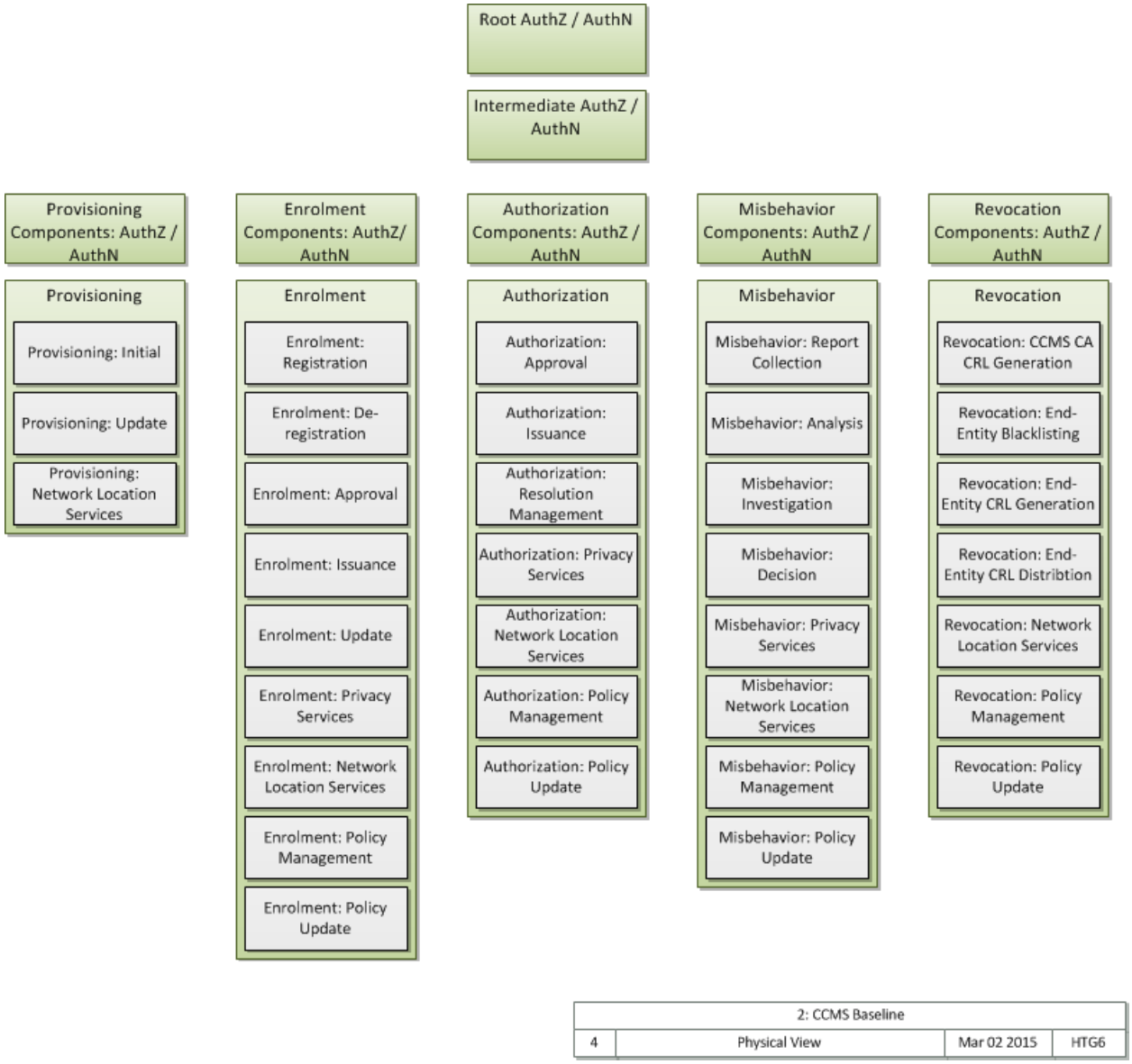
7. Misbehavior components collect and analyze misbehavior reports from end entities.

8. Revocation components act on misbehavior analyses, and move end entities to the Unauthorized or Unenrolled states.

Figure 3 shows the components of the CCMS organized in a logical fashion, with foundational trust elements toward the top, and end entity interaction elements moving in a progression akin to the default case for end entity interactions with the CCMS from left to right, where those phases of the end entity's life are defined in Section 4.1.2

Each component contains several sub components that are dedicated to perform the particular tasks of a component. For example, as shown in Figure 3 the provisioning component contains three high level functions in its sub components: Provisioning: Initial, Provisioning: Update, and Provisioning: Network Location Services.

Figure 3: CCMS Components



4.1.4 Use cases

In the lifecycle of the end entity, as described in Section 4.1.2, different functions need to be performed to go from one state to another state. The security use cases introduced in this section and listed in Table 1 are subsequently linked in Section 4.1.5 to the sub components as shown in Figure 3. A more detailed description of the use cases can be found in Chapter 8 (Lifecycle and Operation Issues) of the PRESERVE deliverable D5.1 [7].

Table 1: Use Cases of the CCMS and End Entity Lifecycle

Use case	Use case name
UC-Add-RCA	Installation of new Root
UC-Add-ICA	Installation of new Intermediate CA
UC-Add-PC	Installation of new Provisioning Component
UC-Add-EC	Installation of new Enrollment Component
UC-Add-AC	Installation of new Authorization Component
UC-Add-MC	Installation of new Misbehavior Component
UC-Add-RC	Installation of new Revocation Component
UC-Add-CS	Installation of new Certification Service
UC-Upd-RCA	Update of Root certificate
UC-Upd-ICA	Update of Intermediate CA certificate
UC-Upd-EC	Update of Enrollment Component certificate
UC-Upd-AC	Update of Authorization Component certificate
UC-Upd-MC	Update of Misbehavior Component certificate
UC-Upd-RS	Update of Revocation Component certificate
UC-Upd-CS	Update of Certification Service
UC-Rev-RCA	Revocation of Root certificate
UC-Rev-ICA	Revocation of Intermediate CA certificate
UC-Rev-EC	Revocation of Enrollment Component certificate
UC-Rev-AC	Revocation of Authorization Component certificate
UC-Rev-MC	Revocation of Misbehavior Component certificate
UC-Rev-CS	Revocation of Revocation Component certificate
UC-EoL-RCA	End of lifetime of Root certificate
UC-EoL-ICA	End of lifetime of Intermediate CA certificate
UC-EoL-EC	End of lifetime of Enrollment Component certificate
UC-EoL-AC	End of lifetime of Authorization Component certificate
UC-EoL-MC	End of lifetime of Misbehavior Component certificate
UC-EoL-RC	End of lifetime of Revocation Component certificate
UC-Rem-CS	Removal of Certification Service
UC-UP-RCA	Update policy of Root / CCMS
UC-UP-ICA	Update policy for Intermediate CA
UC-UP-EC	Update policy for Enrollment Component
UC-UP-AC	Update policy for Authorization Component
UC-UP-MC	Update policy for Misbehavior Component
UC-UP-RC	Update policy for Revocation Component
UC-Enrol-EE	Enrollment of end entity
UC-U-Enrol-EE	Update of enrollment of end entity
UC-Auth-EE	Authorization of end entity (Refill of Pseudonym Certificates)

UC-Approv-Auth-EE	Approval for authorization of end entity
UC-Add-ER	Add Enrollment: Registration
UC-Rem-ER	Remove Enrollment: Registration
UC-SW-Upd-EE	Software update of end entity's system
UC-HW-Upd-EE	Physical update or replacement of end entity elements (OBU, HSM)
UC-Upd-Sec	Change of security (crypto) format / protocol
UC-Dep-Sec	Deprecate security (crypto) format / protocol
UC-Upd-Cert	Change of certificate format
UC-Dep-Cert	Deprecate certificate format
UC-Rec-MR	Misbehavior report collection
UC-Proc-MR	Misbehavior analysis (evaluating the report content)
UC-Check-MR	Misbehavior investigation (ID resolution management)
UC-Dec-MR	Misbehavior decision

4.1.5 CCMS Component Details: Functions, Interfaces and Trust Chain Elements

For each CCMS component, a diagram is presented showing the functions included in the component and the component's interfaces to other components, trust chain elements and other CCMS. Objects and interfaces are coded according to the following scheme:

Table 2: Component Diagram Legend

Black: the interface is not secured through signature or encryption mechanisms
Blue: the interface primarily passes data encrypted for the target. Messages are not signed
Green: the interface primarily passes unencrypted data signed using the source's credentials
Red: the interface primarily passes data encrypted for the target and signed using the source's credentials

For each component function, the following aspects are defined:

- **Objective**: The objective of the function
- **Required Behavior**: The actions the function must undertake in order to accomplish its objective
- **Material Input**: Inputs from other components or end entity required in order to satisfy

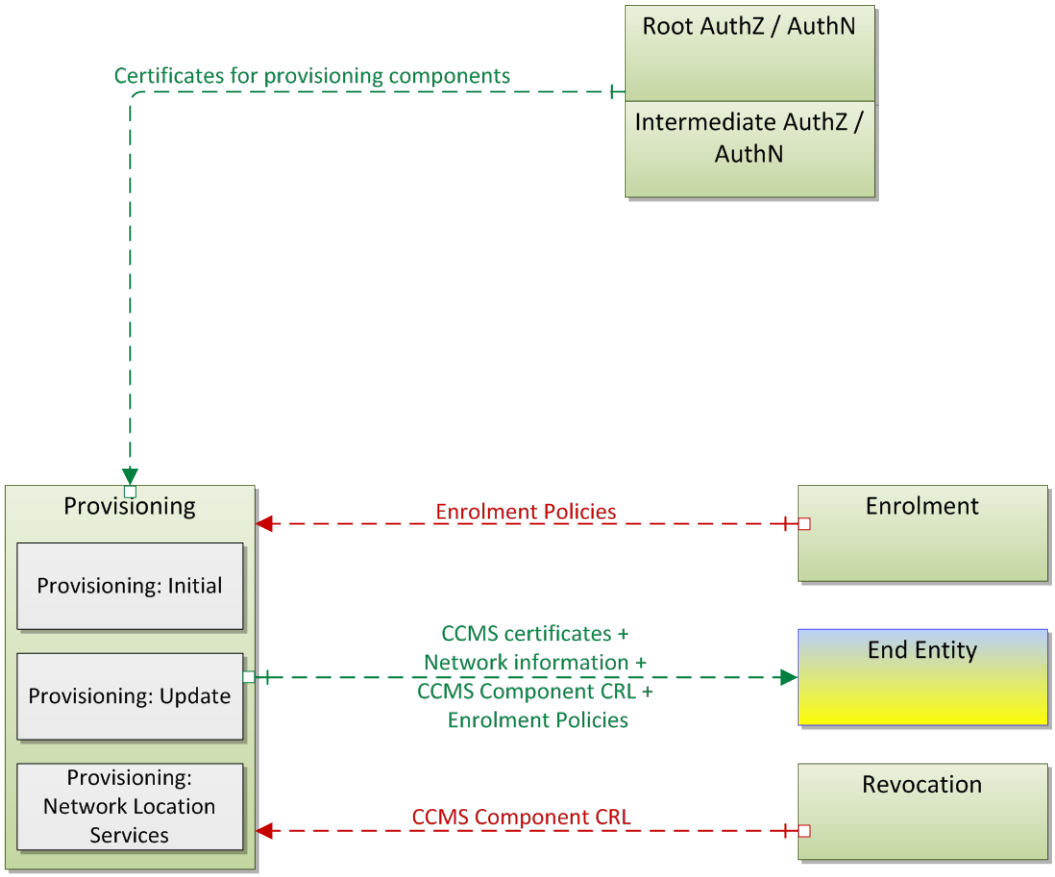
required behavior

- **Material Output**: Outputs generated and sent to other components, or end entity in order to satisfy required behavior
- **Interface to other CCMS**: Whether any inputs or outputs go to other CCMS. These may also relate to cross-certification, and discussed in more detail for some cases in **HTG6-3: C-ITS Credential Management System Architectural Analysis and Recommendations for Harmonization**
- **Use Cases Related to this Component**: the use cases that may utilize this component.
- **Existing Designs**: How the component function is reflected in the two existing designs, the CAMP SCMS and the PRESERVE PKI.

4.1.5.1 Provisioning Components

Provisioning components provide the end entity with material that allows it to enter the "Unenrolled" state. This consists of root certificates and the crypto material that allows it to communicate securely with the Enrollment system.

Figure 4: CCMS Provisioning



2: End Entity Provisioning			
4	Physical View	Feb 11 2015	HTG6

- **Provisioning: Initial**
  - Objective: To provide the end entity with the material required to move from the "unprovisioned" state to the "unenrolled" state.
  - Required behavior:
    - Ensure that the end entity meets requirements for this domain.
    - Provide the end entity with certificates of other CCMS components
    - Provide the end entity with relevant policy information related to provided material
  - Material Input:
    - Enrollment: Enrollment policies

    - Revocation: CCMS Component CRL
- Material Output:
    - End Entity: Root CA certificate
    - End Entity: CRL signer certificate
    - End Entity: Enrollment CA certificate
    - End Entity: Authorization CA certificate
    - End Entity: Misbehavior Authority certificate
    - End Entity: CCMS Component CRL
    - End Entity: Enrollment CSR policy and configuration such as max validity of enrollment certificate
- Interface to other CCMS: None
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Enrollment Component
Installation of new Authorization Component	Installation of new Misbehavior Component
Installation of new Revocation Component	Installation of new Certification Service
Update of Root certificate	Update of Intermediate CA certificate
Update of Enrollment Component certificate	Update of Authorization Component certificate
Update of Misbehavior Component certificate	Update of Revocation Component certificate
Update of Certification Service	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
Revocation of Authorization Component certificate	Revocation of Misbehavior Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Enrollment Component certificate
End of lifetime of Authorization Component certificate	End of lifetime of Misbehavior Component certificate
End of lifetime of Revocation Component certificate	Removal of Certification Service
Update policy of Root / CCMS	Update policy for Intermediate CA
Update policy for Enrollment Component	Update policy for Authorization Component
Update policy for Misbehavior Component	Update policy for Revocation Component
Physical update or replacement of end entity elements (OBU, HSM)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to DCM)
  - PRESERVE PKI: Root CA provides interface to download the root certificate, all certificates issued by this root, and the CCMS component CRL. The enrollment component provides interface to download the enrollment and authorization policy. Manufacturers of ITS-S are responsible to collect all relevant information for provisioning.

- **Provisioning: Update**
  - Objective: To provide the end entity with updated provisioning material
  - Required behavior: same as Provisioning: Initial
  - Material Input: same as Provisioning: Initial
  - Material Output: same as Provisioning: Initial
  - Interface to other CCMS: None
  - Use Cases Related to this Component: same as Provisioning: Initial
  - Existing designs: same as Provisioning: Initial

- **Provisioning: Network Location Services**
  - Objective: To provide the end entity with network information necessary for it to communicate with Provisioning and Enrollment components
  - Required behavior:
    - Provide the end entity with networking contact information
  - Material Input: None
  - Material Output:
    - End Entity: network information such as IP addresses, host names, or X.509 certificates of CCMS Provisioning and Enrollment components
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

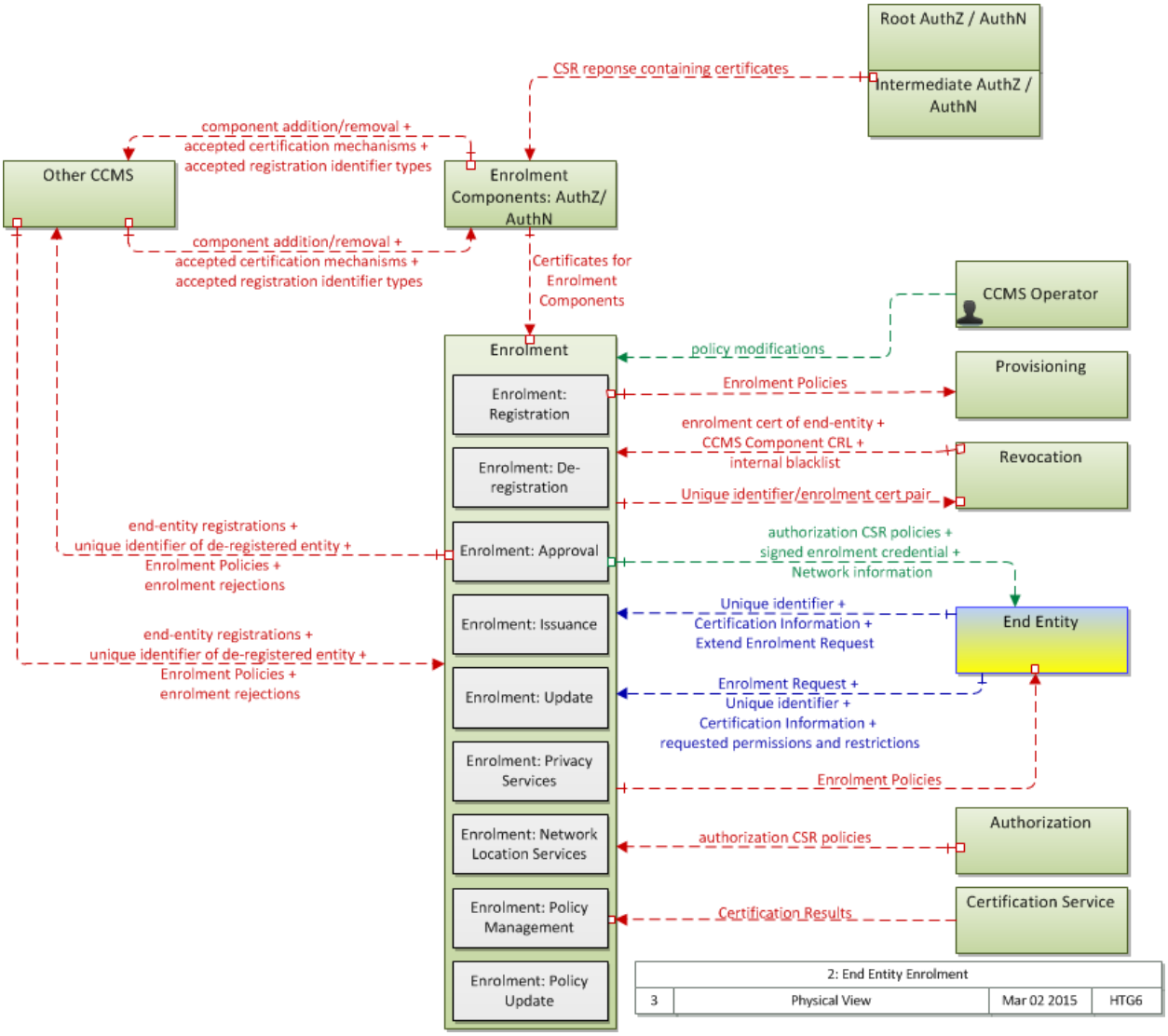
Installation of new Provisioning Component	Physical update or replacement of end entity elements (OBU, HSM)
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol

  - Existing designs:
    - CAMP SCMS: Device Configuration Manager (DCM)
    - PRESERVE PKI: Not identified

4.1.5.2 Enrollment Components

The end entity applies for and obtains enrollment credentials from Enrollment components. At the end of interaction with Enrollment components, the end entity has its own security credentials that can be used to communicate with other CCMS components, and is in the "Unauthorized" state. Enrollment is the sole CCMS function that can match an enrollment certificate to the unique identifier of a device, and so participates in de-registration processes through interaction with Revocation functions.

Figure 5:End Entity Enrollment



- **Enrollment Components: AuthZ/AuthN**
  - Objective: To enable trust of Enrollment components by end entities and other CCMS components
  - Required behavior:
    - Ensure that Enrollment components are implemented and operated correctly
    - Issue signed Enrollment component certificates
  - Material Input:
    - CSR response from Root CA containing certificates (suitable for Enrollment components; typically long-lived), signed by Root CA
  - Material Output:
    - Certificates for Enrollment components
  - Interface to other CCMS:
    - Notification of Enrollment Components of other CCMS that there is a new Enrollment Component that has to be contacted in the process Enrollment: Registration
    - Specification of accepted certification mechanism
    - Specification of identifier types used in registration/deregistration
  - Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Enrollment Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Enrollment Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Enrollment Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component
Enrollment of end entity	Update of enrollment of end entity
Approval for authorization of end entity	Add Enrollment: Registration
Remove Enrollment: Registration	Physical update or replacement of end entity elements (OBU, HSM)
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

  - Existing designs:
    - CAMP SCMS: Designed (corresponds to Root/Intermediate CA)
    - PRESERVE PKI: Designed (corresponds to Root CA)

- **Enrollment: Registration**
  - Objective: To add an end entity to the list of registered entities
  - Required behavior: Ensure that the end entity is entitled to be enrolled.
  - Material Input:
    - End Entity: Enrollment request including signature of Provisioning entity
    - End Entity: Device unique identifier
    - End Entity: Requested permissions and validity restrictions
    - End Entity: Information about certification of the end entity instance or class
  - Material Output:
    - End Entity: Rejection or acknowledgement
  - Interface to other CCMS:
    - Enrollment: Registration – Exchange of information about registered end entity if not private
  - Use Cases Related to this Component:

Installation of new Enrollment Component[1]	Installation of new Revocation Component
Installation of new Certification Service	Update of Enrollment Component certificate
Update of Revocation Component certificate	Update of Certification Service
Revocation of Enrollment Component certificate	End of lifetime of Enrollment Component certificate
Removal of Certification Service	Update policy for Enrollment Component [2]
Add Enrollment: Registration	Remove Enrollment: Registration
Physical update or replacement of end entity elements (OBU, HSM)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

  - Existing designs:
    - CAMP SCMS: Designed (corresponds to DCM)
    - PRESERVE PKI: Designed (corresponds to Long-Term CA and attestation of module key)

- **Enrollment: Deregistration**
  - Objective: To match an enrollment credential with a unique identifier to facilitate blacklisting of end entity at Enrollment: Approval and Authorization: Approval
  - Material Input:
    - Revocation: End entity enrollment credential
  - Material Output:

[1] May require interface to other CCMS if trust between CCMS is established at the enrollment certificate level
[2] *Ibid*

        - Revocation: Rejection or acknowledgement
        - Revocation: Unique identifier of end entity with enrollment certificate
    - Interface to other CCMS:
        - Revocation: End entity Backlisting, Enrollment: Deregistration – Unique identifier of deregistered end entity
    - Use Cases Related to this Component:

Installation of new Enrollment Component [3]	Revocation of Enrollment Component certificate
End of lifetime of Enrollment Component certificate	Update policy for Enrollment Component [4]
Add Enrollment: Registration	Remove Enrollment: Registration
Physical update or replacement of end entity elements (OBU, HSM)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

    - Existing designs:
        - CAMP SCMS: Not present
        - PRESERVE PKI: Designed (corresponds to Long-Term CA)
- **Enrollment: Approval**
    - Objective: To determine that the requesting end entity is entitled to an enrollment certificate
    - Required behavior:
        - Ensure that the end entity actually *is* entitled!
        - Interact with Revocation: End entity Backlisting to check that end entity is not blacklisted
    - Material Input:
        - Certification Service: Information about certification of the end entity instance or class (e.g. certificate of certification authority (might be OEM self-certification))
    - Material Output:
        - End Entity: Rejection (otherwise proceed to Enrollment: Issuance)
    - Interface to other CCMS:
        - Enrollment: Approval if trust between CCMS is established on enrollment certificate level
        - Revocation: End entity Blacklisting

[3] *Ibid*
[4] *Ibid*

- Use Cases Related to this Component:

Installation of new Enrollment Component [5]	Installation of new Revocation Component [6]
Update of Enrollment Component certificate	Update of Revocation Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Enrollment Component certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Root certificate	Update policy of Root / CCMS
End of lifetime of Enrollment Component certificate	Update policy for Enrollment Component [7]
Update policy for Intermediate CA	Update of enrollment of end entity [8]
Enrollment of end entity [9]	Deprecate security (crypto) format / protocol
Change of security (crypto) format / protocol	Deprecate certificate format
Change of certificate format	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to DCM)
  - PRESERVE PKI: Designed (corresponds to Long-Term CA)

- **Enrollment: Issuance**
  - Objective: To provide the requesting end entity with a signed enrollment certificate
  - Required behavior:
    - Verify that the request is approved by Enrollment: Approval
    - Sign the enrollment certificate
    - Provide the enrollment certificate to the requesting end entity
    - Store identifier of enrollment credential and associated unique identifier of end entity to which it was granted
  - Material Input:
    - CSR of enrollment certificate
    - Permitted permissions (e.g. ITS Application ID with Service Specific Permissions (ITS-AID-SSP), PSID, Trust Assurance Level)
    - Permitted validity restrictions concerning time and location
  - Material Output:
    - End Entity: Signed enrollment credential
    - End Entity: Authorization CSR policies (such as pseudonym CSR policy, max validity of authorization certificate)

[5] *Ibid*
[6] *Ibid*
[7] *Ibid*
[8] *Ibid*
[9] *Ibid*

- Interface to other CCMS: None
- Use Cases Related to this Component:

Update of Enrollment Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Enrollment Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component
Enrollment of end entity	Update of enrollment of end entity
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

- Existing designs:
  - CAMP SCMS: Designed (corresponds to ECA)
  - PRESERVE PKI: Designed (corresponds to Long-Term CA)

- **Enrollment: Update**
  - Objective: Verify that an end entity is entitled (not blacklisted) to request new enrollment certificates.
  - Required behavior:
    - Ensure that the previously blacklisted end entity is entitled to be re-enrolled.
    - Notify Enrollment: Approval of this change
    - Notify Revocation: End entity Blacklisting of this change
  - Material Input:
    - End Entity: Enrollment request including signature of Provisioning entity
    - End Entity: Device unique identifier
    - End Entity: Requested permissions and validity restrictions
    - End Entity: Information about certification of the end entity instance or class
    - Certification Service: Information about certification (e.g. certificate of certification authority (might be OEM self-certification) or certificate of repair shop)
  - Material Output:
    - End Entity: Rejection or new enrollment certificate provided by Enrollment: Issuance
    - Revocation: End entity Blacklisting: Notification that end entity is no longer blacklisted
  - Interface to other CCMS: None

- Use Cases Related to this Component:

Update of Enrollment Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Enrollment Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component
Update of enrollment of end entity	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to ECA)
  - PRESERVE PKI: Designed (corresponds to Long-Term CA)

- **Enrollment: Privacy services**
  - Objective: Protect the identity and contact information of the end entity from other components of the CCMS.
  - Required behavior: Dependent on privacy services needed.
  - Material Input:
    - End Entity: IP address of end entity or other identifiers in the communication
  - Material Output:
    - <Any internal>: anonymous / pseudonymous identifier of requesting end entity provided to any internal enrollment component.
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

Revocation of Enrollment Component certificate	End of lifetime of Enrollment Component certificate
Update policy for Enrollment Component	Enrollment of end entity
Update of enrollment of end entity	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

  - Existing designs:
    - CAMP SCMS: Not identified
    - PRESERVE PKI: Not identified

- **Enrollment: Network Location Services**
  - Objective: To provide the end entity with network information necessary for it to communicate with Enrollment and Authorization components
  - Required behavior:
    - Provide the end entity with networking contact information
  - Material Input: None
  - Material Output:
    - End Entity: Network information such as IP addresses or host names, and X.509 certificates of CCMS Enrollment and Authorization components
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

Installation of new Provisioning Component	Installation of new Enrollment Component
Update of Enrollment Component certificate	Revocation of Enrollment Component certificate
End of lifetime of Enrollment Component certificate	Update policy for Enrollment Component
Enrollment of end entity	Update of enrollment of end entity
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol

  - Existing designs:
    - CAMP SCMS: Device Configuration Manager (DCM)
    - PRESERVE PKI: Not identified

- **Enrollment: Policy Management**
  - Objective: To manage policy information related to Enrollment services
  - Required behavior:
    - Create and modify policy information related to enrollment functions. Example policies could be related to certification service minimum qualifications, audit requirements, enrollment certificate lifetime, types of end entities the CCMS will enroll, etc.
  - Material Input:
    - Policy information from CCMS Operator. Some policy changes may be based on inter-CCMS trust (see section 4.4), but most are likely to be manual changes.
  - Material Output: None
  - Interface to other CCMS:
    - None (see section 4.4).
  - Use Cases Related to this Component:

Installation of new Root[10]	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Enrollment Component
Installation of new Authorization Component	Update of Root certificate
Update of Intermediate CA certificate	Update of Enrollment Component certificate
Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
Revocation of Authorization Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Enrollment Component certificate
End of lifetime of Authorization Component certificate	Update policy of Root / CCMS[11]
Update policy for Intermediate CA	Update policy for Enrollment Component
Update policy for Authorization Component	Enrollment of end entity
Update of enrollment of end entity	Change of security (crypto) format / protocol[12]
Deprecate security (crypto) format / protocol[13]	Change of certificate format[14]
Deprecate certificate format[15]	

- Existing designs:
  - CAMP SCMS: ECA manages related policies
  - PRESERVE PKI: Long-Term CA manages related policies

- **Enrollment: Policy Update**
  - Objective: To distribute policy information related to Enrollment services to end-entities and other CCMS objects that need those policies
  - Required behavior:
    - Provide policy information related to enrollment functions to end entity.
  - Material Input: None
  - Material Output:
    - Provisioning: Policies relevant to Enrollment functions, such as certification/re-certification requirements, enrollment certificate lifetime, privacy policies to end entities and Provisioning: Initial/Update

[10] May require interface to other CCMS if trust between CCMS is established on enrollment certificate level or canonical / registration certificate level
[11] *Ibid*
[12] *Ibid*
[13] *Ibid*
[14] *Ibid*
[15] *Ibid*

- End Entity: Policies relevant to Enrollment functions, such as certification/re-certification requirements, enrollment certificate lifetime, privacy policies to end entities and Provisioning: Initial/Update

- Interface to other CCMS:
  - Provide Enrollment policies to other CCMS Enrollment: Approval if trust between CCMS is established on enrollment certificate level
- Use Cases Related to this Component:

Installation of new Root[16]	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Enrollment Component
Installation of new Authorization Component	Update of Root certificate
Update of Intermediate CA certificate	Update of Enrollment Component certificate
Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
Revocation of Authorization Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Enrollment Component certificate
End of lifetime of Authorization Component certificate	Update policy of Root / CCMS[17]
Update policy for Intermediate CA	Update policy for Enrollment Component
Update policy for Authorization Component	Enrollment of end entity
Update of enrollment of end entity	Change of security (crypto) format / protocol[18]
Deprecate security (crypto) format / protocol[19]	Change of certificate format[20]
Deprecate certificate format[21]	

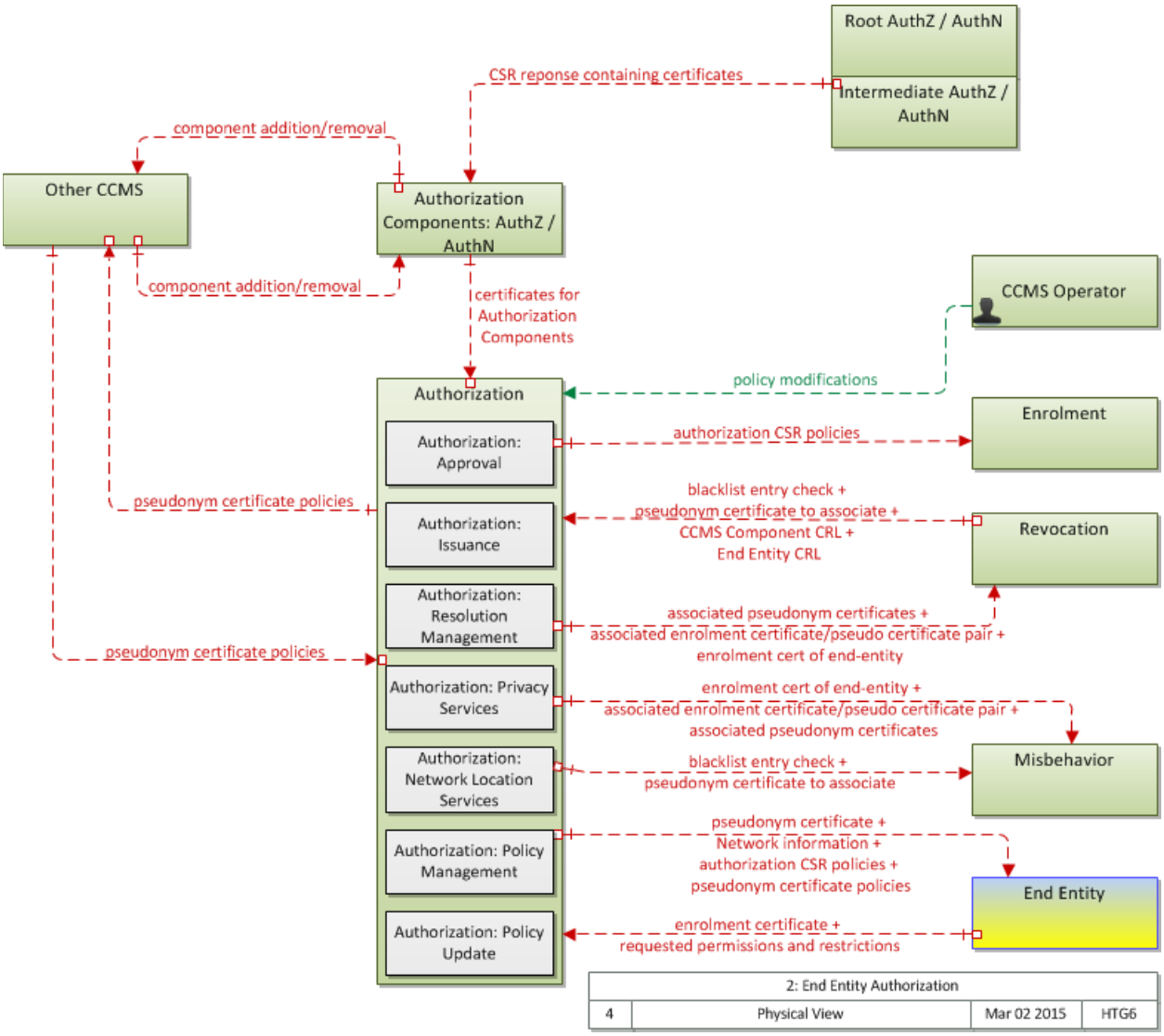
- Existing designs:
  - CAMP SCMS: Not identified
  - PRESERVE PKI: Long-Term CA provides elements of enrollment policy via public interfaces

[16] may require interface to other CCMS if trust between CCMS is established on enrollment certificate level or canonical / registration certificate level
[17] *Ibid*
[18] *Ibid*
[19] *Ibid*
[20] *Ibid*
[21] *Ibid*

4.1.5.3 Authorization Components

The end entity applies for and obtains authorization credentials. Pseudonym certificates are an example of authorization credentials. Once its interaction with the Authorization components is complete, the end entity is in the "Operational" state. While this section talks about a singular set of authorization credentials, authorization could just as easily deliver a batch of pseudonym certificates.

Figure 6: End Entity Authorization



- **Authorization Components: AuthZ/AuthN**
  - Objective: To enable trust of Authorization components by end entities and other CCMS

components

- Required behavior:
  - Ensure that Authorization components are implemented and operated correctly
  - Issue Authorization component certificates
- Material Input:
  - CSR response from Root CA containing certificates (suitable for Authorization components; typically long-lived), signed by Root CA
- Material Output:
  - Certificates for Authorization components
- Interface to other CCMS:
  - Notification of new Authorization components to Other CCMS: Misbehavior and Other CCMS: Revocation components (so the other CCMS knows that there is a new Authorization Component that can be contacted in the Misbehavior: Analysis processes)
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Authorization Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Authorization Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Authorization Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Authorization Component
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to Root/Intermediate CA)
  - PRESERVE PKI: Designed (corresponds to Root CA)

- **Authorization: Approval**
  - Objective: Verify that the requesting end entity is entitled to get an authorization certificate
  - Required behavior:

- Check that permissions requested match the permissions actually granted by the enrollment credentials
- Verify that the enrollment credentials are not blacklisted

- Material Input:
  - End Entity: Enrollment certificate of end entity
  - End Entity: Requested permissions (e.g. ITS Application ID with Service Specific Permissions (ITS-AID-SSP), PSID, Trust Assurance Level)
  - End Entity: Requested validity restrictions concerning time and location
  - Authorization: Issuance: Acknowledgement that end entity certificates have been issued and that the end entity has received the certificates. The end entity must not be able to repudiate the reception of certificates if they are successfully transmitted.
  - Revocation: Internal black list of revoked end-entities (managed by Revocation: CCMS End entity CRL Generation)
- Material Output: Rejection or acknowledgement
- Interface to other CCMS:
  - Enrollment: Approval if trust between CCMS is established on enrollment certificate level as discussed in Section 4.4.3.3. For this case the authorization component needs to know if the end entity is enrolled with another trusted CCMS.
  - Revocation: End entity Blacklisting. For this case the authorization component needs to know if the end entity is blacklisted by another trusted CCMS.
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Enrollment Component[22]	Installation of new Revocation Component[23]
Update of Root certificate	Update of Intermediate CA certificate
Update of Enrollment Component certificate	Update of Revocation Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Enrollment Component certificate	Revocation of Authorization Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Enrollment Component certificate
End of lifetime of Authorization Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component [24]

[22] may require interface to other CCMS if trust between CCMS is established on authorization approval level
[23] *Ibid*
[24] *Ibid*

Update policy for Authorization Component	Update policy for Revocation Component
Authorization of end entity (Refill of Pseudonym Certificates)	Approval for authorization of end entity [25]
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

- Existing designs:
  - CAMP SCMS: Designed (corresponds to Registration Authority)
  - PRESERVE PKI: Designed (corresponds to Long-Term CA)

- **Authorization: Issuance**
  - Objective: Signs and delivers the authorization certificate to the end entity
  - Required behavior:
    - Verify that the request is approved by Authorization: Approval
    - Sign the authorization certificate
    - Provide the authorization certificate to the end entity
  - Material Input:
    - Authorization: Approval: CSR for pseudonym certificate
    - Authorization: Approval: Permitted permissions (e.g. ITS Application ID with Service Specific Permissions (ITS-AID-SSP), PSID, Trust Assurance Level)
    - Authorization: Approval: Permitted validity restrictions concerning time and location
  - Material Output:
    - End Entity: Authorization (pseudonym) certificate
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Authorization Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Authorization Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Authorization Component
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

[25] *Ibid*

- Existing designs:
  - CAMP SCMS: Designed (corresponds to Pseudonym CA)
  - PRESERVE PKI: Designed (corresponds to Pseudonym CA)

- **Authorization: Resolution Management**
  - Objectives: Manage information that can be used to link pseudonym certificates to each other or to link pseudonym certificates to enrollment certificates for misbehavior report evaluation and revocation/removal purposes
  - Required behavior:
    - CCMS implementation-dependent
  - Material Input:
    - Revocation: Pseudonym certificate to associate
    - Others CCMS implementation-dependent
  - Material Output:
    - Revocation: Pseudonym association or identifier
    - Revocation: Pseudonym matching enrollment certificate
    - Revocation: Others CCMS implementation-dependent
  - Interface to other CCMS:
    - Revocation: CCMS CA CRL Generation - Authorization: Resolution Management has to know which component of other CCMS can be trusted.
    - Could include interface to foreign Misbehavior: Investigation or other components as well, depending on implementation
  - Use Cases Related to this Component:

Installation of new Misbehavior Component[26]	Update of Misbehavior Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Authorization Component certificate	Revocation of Misbehavior Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Authorization Component certificate	End of lifetime of Misbehavior Component certificate
Update policy of Root / CCMS	Update policy for Intermediate CA
Update policy for Authorization Component	Update policy for Misbehavior Component[27]
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	Misbehavior investigation (ID resolution management)[28]

[26] may require interface to other CCMS to exchange resolution information
[27] *Ibid*
[28] *Ibid*

- Existing designs:
  - CAMP SCMS: Designed (corresponds to Linkage Authorities)
  - PRESERVE PKI: Use Case identified (corresponds to request identifier shared between Pseudonym CA and Long-Term CA)

- **Authorization: Privacy services**
  - Objective: Protect the identity and contact information of the end entity from other components of the CCMS
  - Required behavior:
    - Dependent on privacy services needed by CCMS implementation
  - Material Input:
    - IP address of end entity or other identifiers in the communication is provided to any internal authorization component
  - Material Output:
    - Anonymous / pseudonymous identifier of requesting end entity is provided to any internal authorization component
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

Revocation of Authorization Component certificate	End of lifetime of Authorization Component certificate
Update policy for Authorization Component	Authorization of end entity (Refill of Pseudonym Certificates)
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

  - Existing designs:
    - CAMP SCMS: Designed (corresponds to LOP, Registration Authority “shuffle”, separate Linkage Authorities)
    - PRESERVE PKI: Use case identified (implementation left up to individual end-entities)

- **Authorization: Network Location Services**
  - Objective: To provide the end entity with network information necessary for it to communicate with Enrollment and Authorization components
  - Required behavior:
    - Provide the end entity with networking contact information
  - Material Input: None
  - Material Output:
    - End Entity: Network information such as IP addresses, host names, or X.509

certificates of CCMS Authorization and Misbehavior components

- Interface to other CCMS: None
- Use Cases Related to this Component:

Installation of new Provisioning Component	Installation of new Authorization Component
Update of Authorization Component certificate	Revocation of Authorization Component certificate
End of lifetime of Authorization Component certificate	Update policy for Authorization Component
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	

- Existing designs:
  - CAMP SCMS: Device Configuration Manager (DCM)
  - PRESERVE PKI: Not identified

- **Authorization: Policy Management**
  - Objective: To manage policy information related to Authorization services
  - Required behavior:
    - Create and modify policy information related to authorization functions. Example policies could be related pseudonym certificate lifetime, pseudonym certificate batch sizes, pseudonym certificate re-use policies, privacy policies, etc.
  - Material Input:
    - Policy information from CCMS Operator. Some policy changes may be based on inter-CCMS trust (see section 4.4), but most are likely to be manual changes.
  - Material Output: None
  - Interface to other CCMS: None (see section 4.4).
  - Use Cases Related to this Component:

Installation of new Root[29]	Installation of new Intermediate CA
Installation of new Enrollment Component	Update of Root certificate
Update of Intermediate CA certificate	Update of Enrollment Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Enrollment Component certificate	Revocation of Authorization Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Enrollment Component certificate	End of lifetime of Authorization Component certificate
Update policy of Root / CCMS[30]	Update policy for Intermediate CA

[29] may require interface to other CCMS if trust between CCMS is established on authorization approval level

Update policy for Enrollment Component	Update policy for Authorization Component
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol[31]
Deprecate security (crypto) format / protocol[32]	Change of certificate format[33]
Deprecate certificate format[34]	

- Existing designs:
  - CAMP SCMS: RA manages relevant policies
  - PRESERVE PKI: Root CA manages relevant policies

- **Authorization: Policy Update**
  - Objective: To distribute policy information related to Authorization services to end-entities and other CCMS objects that need those policies
  - Required behavior:
    - Provide policy information related to authorization functions to end entity.
  - Material Input: None
  - Material Output:
    - End Entity: Policies relevant to Authorization functions, such as pseudonym certificate lifetime, pseudonym certificate batch sizes, pseudonym certificate re-use policies, privacy policies, all to end entities.
    - End Entity: Policies relevant to pseudonym certificate request (CSR for authorization)
    - Enrollment: Policy Update: Policies relevant to pseudonym certificate request (CSR for authorization)
  - Interface to other CCMS:
    - Provide Pseudonym use policies to other CCMS if trust between CCMS is established on authorization approval level
  - Use Cases Related to this Component:

Installation of new Root[35]	Installation of new Intermediate CA
Installation of new Enrollment Component	Update of Root certificate
Update of Intermediate CA certificate	Update of Enrollment Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Enrollment Component	Revocation of Authorization Component

[30] *Ibid*
[31] *Ibid*
[32] *Ibid*
[33] *Ibid*
[34] *Ibid*
[35] may require interface to other CCMS if trust between CCMS is established on authorization approval level

certificate	certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Enrollment Component certificate	End of lifetime of Authorization Component certificate
Update policy of Root / CCMS[36]	Update policy for Intermediate CA
Update policy for Enrollment Component	Update policy for Authorization Component
Authorization of end entity (Refill of Pseudonym Certificates)	Change of security (crypto) format / protocol[37]
Deprecate security (crypto) format / protocol[38]	Change of certificate format[39]
Deprecate certificate format[40]	

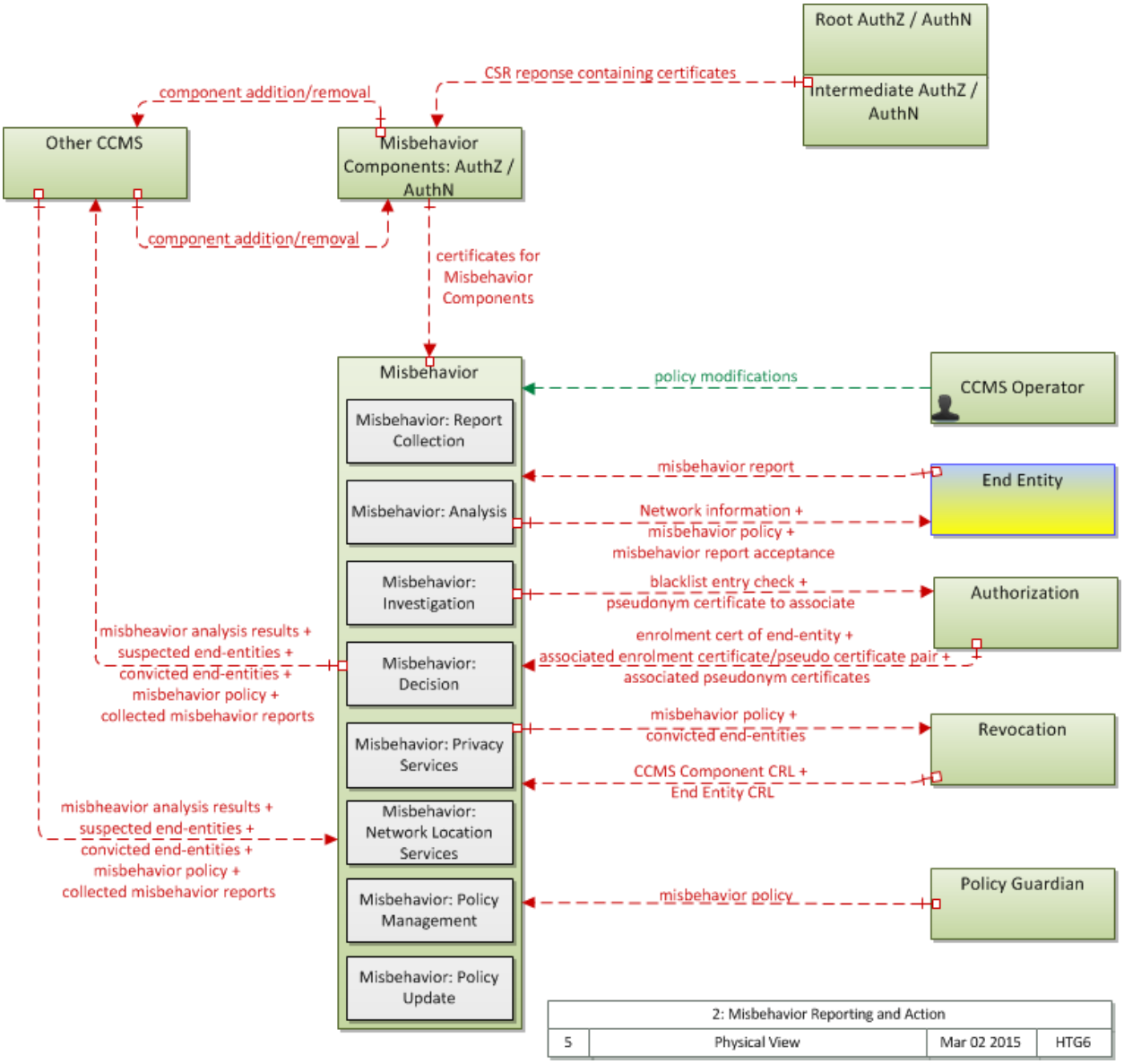
- Existing designs:
  - CAMP SCMS: Not identified
  - PRESERVE PKI: Not identified

4.1.5.4 Misbehavior Reporting Components

End-entities provide misbehavior reports and the Misbehavior component processes them. If revocation is warranted, this component provides information to Authorization or Revocation components to initiate revocation and/or blacklisting, as appropriate.

[36] *Ibid*
[37] *Ibid*
[38] *Ibid*
[39] *Ibid*
[40] *Ibid*

Figure 7: Misbehavior Reporting and Action



- **Misbehavior Components: AuthZ/AuthN**
  - Objective: To enable trust of Misbehavior components by end entities and other CCMS components
  - Required behavior:
    - Ensure that Misbehavior components are implemented and operated correctly
    - Issue Misbehavior component certificates

- Material Input:
  - CSR response from Root CA containing certificates (suitable for Misbehavior Components; typically long-lived), signed by Root CA
- Material Output:
  - Certificates for Misbehavior Components
- Interface to other CCMS: None- done within misbehavior component (report collection, analysis, investigation, decision)
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Misbehavior Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Misbehavior Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Misbehavior Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Misbehavior Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Misbehavior Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

- Existing designs:
  - CAMP SCMS: Designed (corresponds to Root/Intermediate CA)
  - PRESERVE PKI: Designed (corresponds to Root CA)

- **Misbehavior: Report Collection**
  - Objective: Collect, verify and aggregate misbehavior reports from end-entities
  - Required behavior:
    - Collect misbehavior reports
    - Verify cryptographic signatures of misbehavior reports
    - Perform plausibility checks on contents of misbehavior reports
    - Link related misbehavior reports together
  - Material Input:
    - End Entity: Misbehavior reports containing
      - Type (symptom / syndromes) and evidence of misbehavior event
      - Time and location of misbehavior event
      - ID of reporter
      - IDs of suspects
      - IDs of witnesses

- Confidence in correct detection

- Material Output:
  - End Entity: Rejection or acknowledgement of report
- Interface to other CCMS:
  - Provide reports to Other CCMS: Misbehavior Report Collection
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Authorization Component	Installation of new Misbehavior Component[41]
Installation of new Revocation Component	Update of Root certificate
Update of Intermediate CA certificate	Update of Authorization Component certificate
Update of Misbehavior Component certificate	Update of Revocation Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Authorization Component certificate	Revocation of Misbehavior Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Authorization Component certificate
End of lifetime of Misbehavior Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Authorization Component
Update policy for Misbehavior Component[42]	Update policy for Revocation Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format
Misbehavior report collection[43]	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to MA)
  - PRESERVE PKI: Not present

- **Misbehavior: Analysis**
  - Objective: Analyze reports to see if there is enough evidence to start an investigation
  - Required behavior:
    - Analyze reports and quantify evidence according to investigation metrics
    - Compare evidence metrics against investigation thresholds

[41] Interface to other CCMS: Misbehavior Component of other CCMS is informed about a new Misbehavior Component
[42] Interface to other CCMS to inform about the update
[43] Interface to other CCMS to inform about collected reports

- Provide reports that together exceed investigation thresholds to Misbehavior: Investigation

- Material Input:
  - Policy Guardian: Misbehavior policy which may contains information about
    - Types of reported misbehavior that can be processed
    - Rating / weighting of misbehavior types
    - Number of satisfying independent reporters
    - Number of confirmations of misbehavior provided by independent witnesses
    - Minimum level of misbehavior detection confidence
    - Rules to consider suspects as convicted end-entities
- Material Output:
  - End Entity: Misbehavior policy
  - Revocation: Misbehavior policy
- Interface to other CCMS:
  - Results of analysis to Other CCMS: Misbehavior: Analysis
- Use Cases Related to this Component:

Installation of new Misbehavior Component[44]	Update of Misbehavior Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Authorization Component certificate	Revocation of Misbehavior Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Authorization Component certificate
End of lifetime of Misbehavior Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Authorization Component
Update policy for Misbehavior Component[45]	Update policy for Revocation Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format
Misbehavior analysis (evaluating the report content)[46]	

- Existing designs:
  - CAMP SCMS: Designed (corresponds to MA)

[44] Interface to other CCMS: Misbehavior Component of other CCMS is informed about a new Misbehavior Component
[45] Interface to other CCMS to inform about the update
[46] Interface to other CCMS to inform about report analysis

- PRESERVE PKI: Not present

- **Misbehavior: Investigation**
  - Objective: Work with Authorization: Resolution Management to correlate misbehavior instances
  - Required behavior:
    - Investigate potentially related misbehavior instances to determine if they can be associated with the same end entity
    - Perform trend analysis to identify systemic issues in a class of devices
    - According to misbehavior policies, determine if either of the above investigations identify suspected end entities
  - Material Input: None
  - Material Output: None
  - Interface to other CCMS:
    - Other CCMS: Authorization: Resolution Management
    - Other CCMS: Misbehavior: Investigation
  - Use Cases Related to this Component:

Installation of new Misbehavior Component[47]	Update of Misbehavior Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Misbehavior Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Misbehavior Component certificate
Update policy of Root / CCMS	Update policy for Intermediate CA
Update policy for Misbehavior Component[48]	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	Misbehavior investigation (ID resolution management)[49]

  - Existing designs:
    - CAMP SCMS: Designed (corresponds to MA)
    - PRESERVE PKI: Not present

- **Misbehavior: Decision**
  - Objective: Determine that revocation or removal is necessary and take actions to facilitate that.

[47] Interface to other CCMS: Misbehavior Component of other CCMS is informed about a new Misbehavior Component
[48] Interface to other CCMS to inform about the update
[49] Interface to other CCMS to request resolution information

- Required behavior:
  - Interact with Authorization: Resolution Management to obtain information needed to issue the revocation/removal statement
- Material Input: None
- Material Output:
  - Revocation: Convicted end-entities
- Interface to other CCMS:
  - Other CCMS: Misbehavior: Decision
- Use Cases Related to this Component:

Installation of new Misbehavior Component[50]	Installation of new Revocation Component [51]
Update of Misbehavior Component certificate	Update of Revocation Component certificate
Revocation of Misbehavior Component certificate	Revocation of Revocation Component certificate
End of lifetime of Misbehavior Component certificate	Update policy for Misbehavior Component[52]
Update policy for Revocation Component[53]	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	Misbehavior decision[54]

- Existing designs:
  - Existing designs: CAMP SCMS: Designed (corresponds to MA)
  - PRESERVE PKI: Not present

- **Misbehavior: Privacy Services**
  - Objective: Provide privacy against CCMS components learning the identity (or other information such as location) of a reporting end entity
  - Required behavior:
    - Dependent on privacy services needed
  - Material Input:
    - IP address of reporting end entity or other identifiers in the communication
  - Material Output:
    - Anonymous / pseudonymous identifier of reporter
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

[50] Interface to other CCMS: Misbehavior Component of other CCMS is informed about a new Misbehavior Component
[51] *Ibid*
[52] Interface to other CCMS to inform about the update
[53] *Ibid*
[54] Interface to other CCMS to inform to about misbehavior decision

Revocation of Misbehavior Component certificate	End of lifetime of Misbehavior Component certificate
Update policy for Misbehavior Component	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	Misbehavior report collection

- Existing designs:
  - CAMP SCMS: Designed (corresponds to LOP)
  - PRESERVE PKI: Not present

- **Misbehavior: Network Location Services**
  - Objective: To provide the end entity with network information necessary for it to communicate with Misbehavior components
  - Required behavior:
    - Provide the end entity with networking contact information
  - Material Input: None
  - Material Output:
    - End Entity: network information such as IP addresses, host names, or X.509 certificates of CCMS Misbehavior components
  - Interface to other CCMS: None
  - Use Cases Related to this Component:

Installation of new Provisioning Component	Installation of new Misbehavior Component
Update of Misbehavior Component certificate	Revocation of Misbehavior Component certificate
End of lifetime of Misbehavior Component certificate	Update policy for Misbehavior Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Misbehavior report collection	

  - Existing designs:
    - CAMP SCMS: Device Configuration Manager
    - PRESERVE PKI: Not present

- **Misbehavior: Policy Management**
  - Objective: To manage policy information related to Misbehavior services
  - Required behavior:
    - Create and modify policy information related to misbehavior functions. Example policies could include criteria for misbehavior analysis or misbehavior report sharing

- Material Input:
  - Policy information from CCMS Operator. Some policy changes may be based on inter-CCMS trust (see section 4.4), but most are likely to be manual changes
- Material Output: None
- Interface to other CCMS: None (see section 4.4)
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Authorization Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Authorization Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Authorization Component certificate
End of lifetime of Misbehavior Component certificate	Update policy of Root / CCMS (may require interface to other CCMS if misbehavior information is exchanged between CCMS)
Update policy for Intermediate CA	Update policy for Authorization Component
Update policy for Revocation Component	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

- Existing designs:
  - CAMP SCMS: MA manages policies
  - PRESERVE PKI: Not identified

- **Misbehavior: Policy Update**
  - Objective: To distribute policy information related to Misbehavior services to end-entities and other CCMS objects that need those policies
  - Required behavior:
    - Provide policy information related to misbehavior functions to end entity.
  - Material Input: None
  - Material Output:
    - End Entity: Policies relevant to Misbehavior functions, such as pseudonym certificate lifetime, pseudonym certificate batch sizes, pseudonym certificate re-use policies, privacy policies, all to end entities
  - Interface to other CCMS:

- Other CCMS: Pseudonym use policies

- Use Cases Related to this Component:

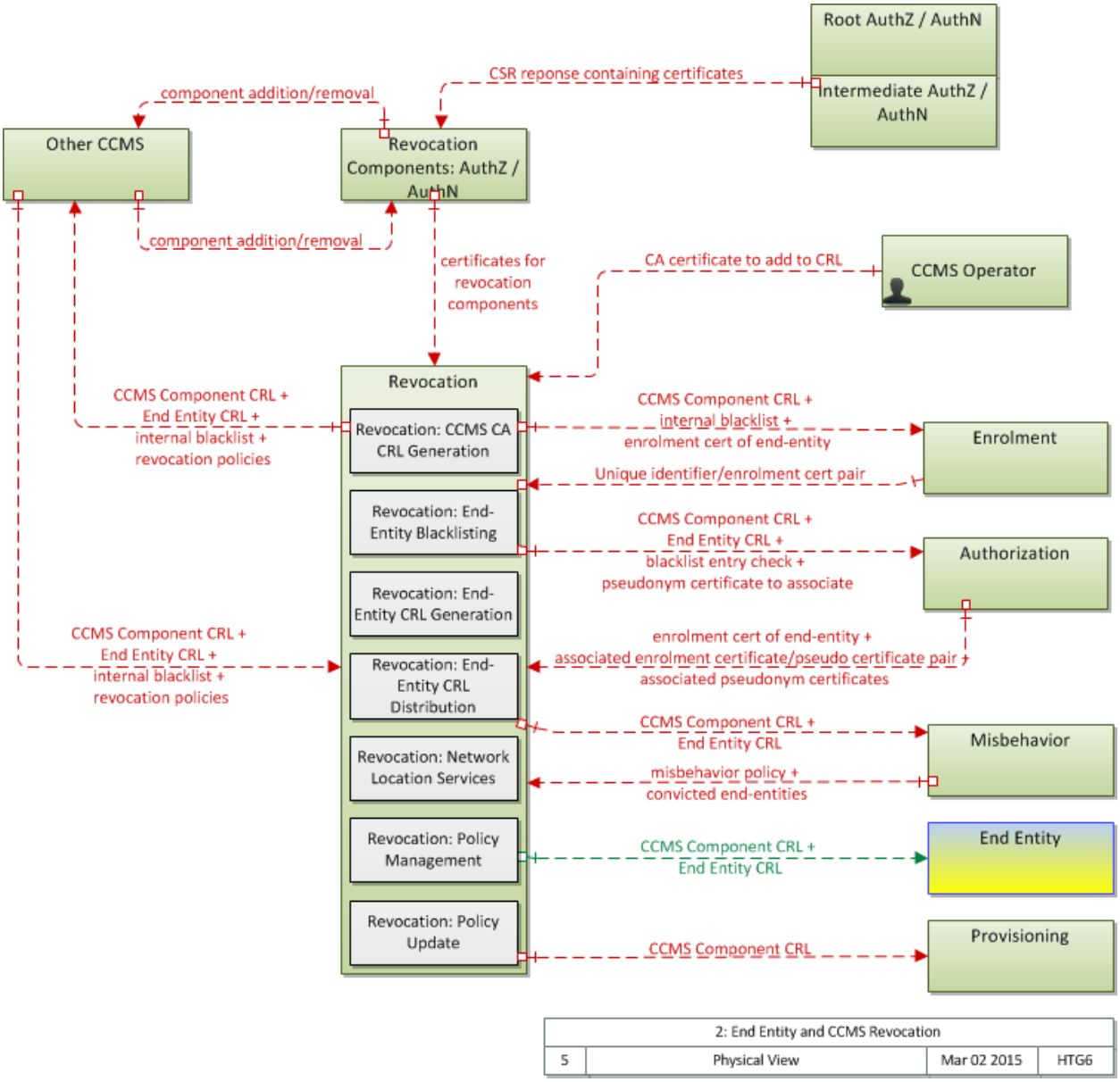
Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Authorization Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Authorization Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Authorization Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Authorization Component certificate
End of lifetime of Misbehavior Component certificate	Update policy of Root / CCMS (may require interface to other CCMS if misbehavior information is exchanged between CCMS)
Update policy for Intermediate CA	Update policy for Authorization Component
Update policy for Revocation Component	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

- Existing designs:
  - CAMP SCMS: Not identified
  - PRESERVE PKI: Not identified

4.1.5.5 Revocation Components

Revocation components generate the internal blacklist and CRL and distribute them to CCMS components and end entities respectively. Once placed on the CRL, an end entity is in the Unauthorized state. Once placed on the blacklist, an end entity is in the Unenrolled state.

Figure 8: Revocation



- **Revocation Components: AuthZ/AuthN**
  - Objective: To enable trust of Revocation components by end entities and other CCMS components
  - Required behavior:
    - Ensure that Revocation components are implemented and operated correctly

        - Issue Revocation component certificates
    - Material Input:
        - CSR response from Root CA containing certificates (suitable for Revocation Components; typically long-lived), signed by Root CA
    - Material Output:
        - Certificates for Revocation components
    - Interface to other CCMS: None - done within revocation component (CCMS component CRL generation, end entity blacklisting, etc.)
    - Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Installation of new Revocation Component
Update of Root certificate	Update of Intermediate CA certificate
Update of Revocation Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Revocation Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Revocation Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Revocation Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format

    - Existing designs:
        - CAMP SCMS: Designed (corresponds to Root/Intermediate CA)
        - PRESERVE PKI: Designed (corresponds to Root CA)

- **Revocation: CCMS CA CRL Generation**
    - Objective: Add certificates of CCMS Components that need to have trust revoked to the CCMA CA CRL
    - Required behavior:
        - Interacts with CCMS coordination (operators) to add CA certificates to CCMS CA CRL
    - Material Input:
        - CCMS Operator: CA certificates to be added to CRL
    - Material Output:
        - CCMS CA CRL
    - Interface to other CCMS:
        - Provide CCMS CA CRL to all other CCMS components via public interface

- Use Cases Related to this Component:

Installation of new Revocation Component[55]	Update of Revocation Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Enrollment Component certificate	Revocation of Authorization Component certificate
Revocation of Misbehavior Component certificate	Revocation of Revocation Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Revocation Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component
Update policy for Authorization Component	Update policy for Misbehavior Component
Update policy for Revocation Component	Enrollment of end entity
Update of enrollment of end entity	Authorization of end entity (Refill of Pseudonym Certificates)
Approval for authorization of end entity	Physical update or replacement of end entity elements (OBU, HSM)
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format
Misbehavior decision	

- Existing designs:
  - CAMP SCMS: Use case identified
  - PRESERVE PKI: Designed (Root CA)

- **Revocation: End entity Backlisting**
  - Objective: Add enrollment certificates to the internal blacklist.
  - Required Behavior:
    - Interact with Authorization: Resolution Management to add enrollment certificates to internal blacklist.
    - Interact with Enrollment: Deregistration to avoid enrollment certificate updates requested by revoked end-entities.
    - May interact with Enrollment: Update to remove end entity from blacklist
  - Material Input:
    - Authorization: Enrollment certificates to be added to the blacklist
  - Material Output:
    - Enrollment: Internal Blacklist
  - Interface to other CCMS:
    - Revocation: End entity Backlisting

[55] Interface to other CCMS to inform about the new CCMS revocation component

- Enrollment: Approval
- Authorization: Approval
- Misbehavior: Decision

o Use Cases Related to this Component:

Installation of new Enrollment Component[56]	Installation of new Misbehavior Component[57]
Installation of new Revocation Component[58]	Update of Misbehavior Component certificate
Update of Revocation Component certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Enrollment Component certificate
Revocation of Misbehavior Component certificate	Revocation of Revocation Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Revocation Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Enrollment Component[59]
Update policy for Misbehavior Component[60]	Update policy for Revocation Component[61]
Enrollment of end entity	Update of enrollment of end entity
Authorization of end entity (Refill of Pseudonym Certificates)	Approval for authorization of end entity[62]
Physical update or replacement of end entity elements (OBU, HSM)	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	Misbehavior decision[63]

o Existing designs:

- CAMP SCMS: Designed (Internal Blacklist Manager)
- PRESERVE PKI: Designed (Long-Term CA)

- **Revocation: End entity CRL Generation**
  - o Objective: Keep the end entity Certificate Revocation List (CRL) up-to-date
  - o Required Behavior:
    - Interact with Authorization: Resolution Management to add certificates to revocation list.

[56] may require interface to other CCMS if trust between CCMS is established on enrollment certificate level or canonical / registration certificate level to check the blacklist
[57] may require interface to other CCMS if misbehavior information is exchanged
[58] may require interface to other CCMS if end entity revocation information is exchanged
[59] may require interface to other CCMS if end entity revocation information is exchanged
[60] *Ibid*
[61] *Ibid*
[62] may require interface to other CCMS if trust between CCMS is established on authorization approval level
[63] may require interface to other CCMS if end entity revocation information is exchanged

- Remove expired certificates from revocation list.
- Interact with Enrollment: Deregistration to avoid enrollment certificate updates requested by revoked end-entities.

- Material Input:
  - Existing Certificate Revocation List,
  - Authorization: End entity certificates to be added to the CRL
- Material Output:
  - Authorization: Certificate Revocation List
  - Misbehavior: Certificate Revocation List
- Interface to other CCMS:
  - Misbehavior: Decision
- Use Cases Related to this Component:

Installation of new Misbehavior Component[64]	Installation of new Revocation Component[65]
Update of Misbehavior Component certificate	Update of Revocation Component certificate
Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Authorization Component certificate	Revocation of Misbehavior Component certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Revocation Component certificate
Update policy of Root / CCMS	Update policy for Intermediate CA
Update policy for Authorization Component	Update policy for Misbehavior Component[66]
Update policy for Revocation Component[67]	Physical update or replacement of end entity elements (OBU, HSM)
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol
Change of certificate format	Deprecate certificate format
Misbehavior decision[68]	

- Existing designs:
  - CAMP SCMS: Designed (CRL Generator)
  - PRESERVE PKI: Not present

- **Revocation: End entity CRL Distribution**
  - Objective: Provide end-entities with end entity Certificate Revocation List
  - Required behavior:

[64] May require interface to other CCMS if misbehavior information is exchanged
[65] May require interface to other CCMS if end entity revocation information is exchanged
[66] May require interface to other CCMS if end entity revocation information is exchanged
[67] *Ibid*
[68] *Ibid*

        - Distribute end entity CRL to end entities
    - Material Input: None
    - Material Output:
        - End Entity: Certificate Revocation List
    - Interface to other CCMS:
        - Revocation: End entity CRL Distribution
    - Use Cases Related to this Component:

Revocation of Root certificate	Revocation of Intermediate CA certificate
Revocation of Revocation Component certificate	End of lifetime of Root certificate
End of lifetime of Intermediate CA certificate	End of lifetime of Revocation Component certificate
Update policy of Root / CCMS	Update policy for Intermediate CA
Update policy for Revocation Component	Change of security (crypto) format / protocol
Deprecate security (crypto) format / protocol	Change of certificate format
Deprecate certificate format	

    - Existing designs:
        - CAMP SCMS: Designed (CRL Broadcast)
        - PRESERVE PKI: Not present

- **Revocation: Network Location Services**
    - Objective: To provide the end entity with network information necessary for it to communicate with Revocation components
    - Required behavior:
        - Provide the end entity with networking contact information
    - Material Input: None
    - Material Output:
        - End Entity: network information such as IP addresses, host names, or X.509 certificates of CCMS Revocation components
    - Interface to other CCMS: None
    - Use Cases Related to this Component:

Installation of new Provisioning Component	Installation of new Revocation Component
Update of Revocation Component certificate	Revocation of Revocation Component certificate
End of lifetime of Revocation Component certificate	Update policy for Revocation Component
Change of security (crypto) format / protocol	Deprecate security (crypto) format / protocol

- Existing designs:
  - CAMP SCMS: Device Configuration Manager (DCM)
  - PRESERVE PKI: Not present

- **Revocation: Policy Management**
  - Objective: To manage policy information related to Revocation services
  - Required behavior:
    - Create and modify policy information related to misbehavior functions. Example policies could include criteria for revocation
  - Material Input:
    - Policy information from CCMS Operator. Some policy changes may be based on inter-CCMS trust (see section 4.4), but most are likely to be manual changes.
  - Material Output: None
  - Interface to other CCMS: None (see section 4.4).
  - Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Update of Root certificate
Update of Intermediate CA certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Revocation Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Revocation Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Revocation Component
Change of security (crypto) format / protocol[69]	Deprecate security (crypto) format / protocol[70]
Change of certificate format[71]	Deprecate certificate format[72]

  - Existing designs:
    - CAMP SCMS: Policies managed by MA
    - PRESERVE PKI: Not identified

- **Revocation: Policy Update**
  - Objective: To distribute policy information related to Revocation services to end-entities and other CCMS objects that need those policies (for example, a change in revocation criteria)

[69] may require interface to other CCMS if end entity revocation information is exchanged
[70] *Ibid*
[71] may require interface to other CCMS if end entity revocation information is exchanged
[72] *Ibid*

- Required behavior:
  - Provide policy information related to Revocation functions to other parts of the CCMS
- Material Input: None
- Material Output:
  - Policies relevant to Revocation functions, such as revocation criteria, CRL distribution parameters
- Interface to other CCMS: None (see section4.4).
- Use Cases Related to this Component:

Installation of new Root	Installation of new Intermediate CA
Installation of new Provisioning Component	Update of Root certificate
Update of Intermediate CA certificate	Revocation of Root certificate
Revocation of Intermediate CA certificate	Revocation of Revocation Component certificate
End of lifetime of Root certificate	End of lifetime of Intermediate CA certificate
End of lifetime of Revocation Component certificate	Update policy of Root / CCMS
Update policy for Intermediate CA	Update policy for Revocation Component
Change of security (crypto) format / protocol[73]	Deprecate security (crypto) format / protocol[74]
Change of certificate format[75]	Deprecate certificate format[76]

- Existing designs:
  - CAMP SCMS: Not identified
  - PRESERVE PKI: Not identified

4.2 Consider End Entity Lifecycle Requirements

In order to consider the security state of an end entity, the CCMS requires certain actions or behavior on the part of the end entity. For instance, in Provisioning: Initial, the CCMS must "Ensure that the end entity meets requirements for this domain." This implies some other step in the device lifecycle: something after or in parallel with manufacturing, but prior to provisioning.

The definition of end entity used in this document, as relevant to a CCMS, is the entity that uses security credentials. Depending on the architecture of the end entity and the policies of the CCMS, these credentials may be used by one or many applications, either by the end entity on behalf of the application or by the application directly in the case where the application *is* the end entity.

[73] may require interface to other CCMS if end entity revocation information is exchanged
[74] *Ibid*
[75] may require interface to other CCMS if end entity revocation information is exchanged
[76] *Ibid*

Similarly, if an end entity is a device that operates multiple applications, those applications may have their own lifecycle.

4.2.1 Handling Credentials

There are four possible cases for the handling of credentials:

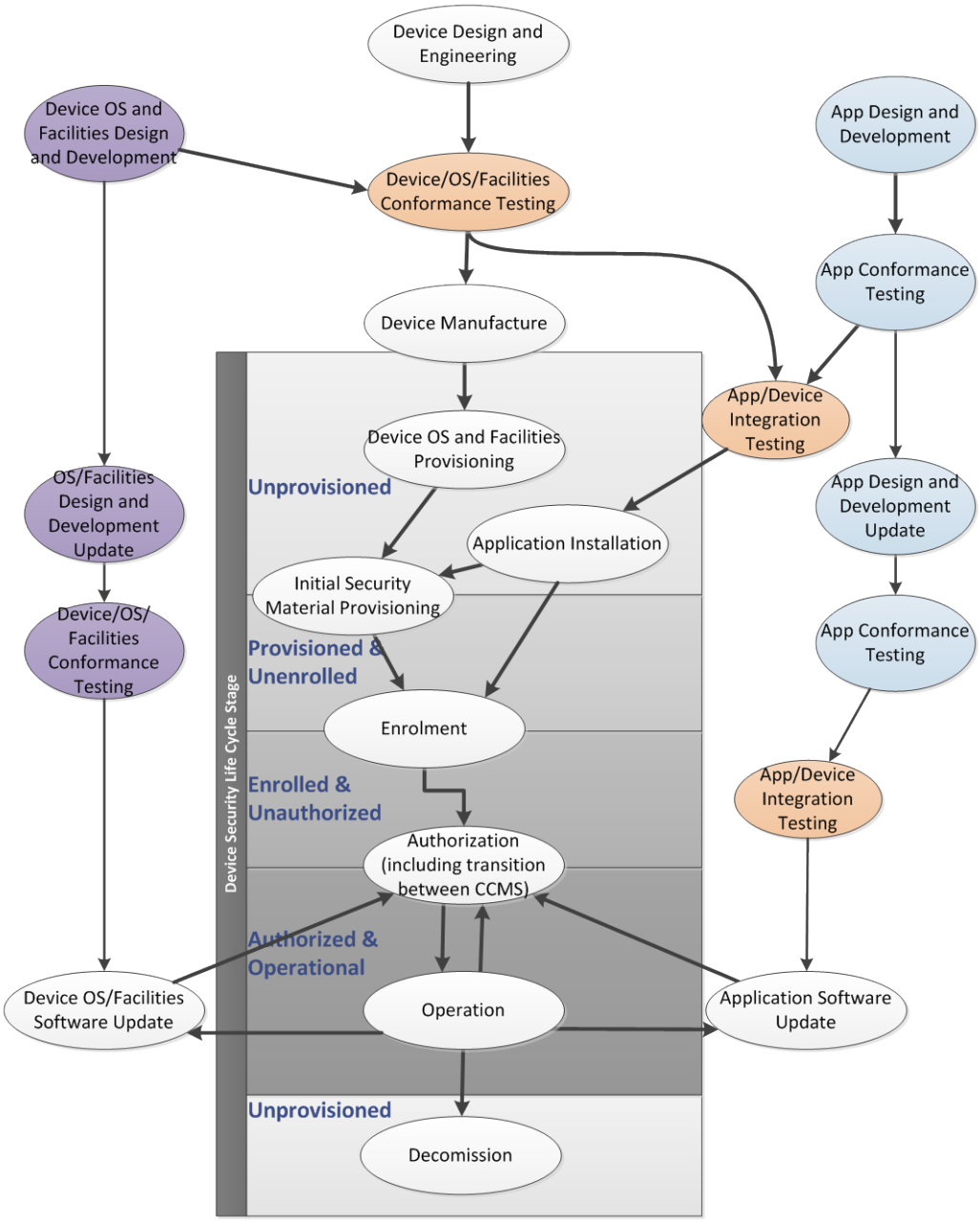
1. Credentials are associated with host devices only, in which case end entity = device
2. Credentials are associated with applications only, in which case end entity = application
3. Credentials are associated with host devices or applications, where a credentialed host may operate credentialed applications, so end entity = application *or* device
4. Credentials are associated with the particular combination of application *and* device

In all cases, one must consider the lifecycle of both applications and devices. Whether applications or devices are end entities affects the aspects of the lifecycle that the CCMS is concerned with. However, the detailed examination of system lifecycle processes in the C-ITS environment, while important to security policy analysis, is a subject that requires its own study. **For the purposes of security policy analysis, the HTG6 is basing its analysis on scenario 4, where both application and device are associated with a particular set of credentials.** Additionally, the lifecycle of both devices and applications are considered to begin with device/application design activities and end with end-of-life.

Figure 9 shows the progress of devices and applications through the end entity lifecycle, where the end entity represents some combination of device and application software object. Note that applications may be installed either before or after initial provisioning, **but must be installed prior to enrollment**. This is because the relationship between CCMS and certification occurs in the enrollment phase; to allow application installation later in the security lifecycle, the CCMS would have to support interaction between Authorization elements and certification bodies, and keep this relationship consistent with any relationships between certification and enrollment. Certification-Authorization relationships are not infeasible, but this relationship raises additional concerns about multiple CCMS interactions to the same external party and the resulting circle-of-trust. **Thus, it is recommended as a future study topic.**

In addition, Device, OS and application update procedures will have to be similarly studied and the required procedures and relationships identified. This view of the lifecycle **assumes** that once an application passes through the certification/conformance testing gate, its functionality remains unchanged, and maintenance and updates are limited to bug-fixes. Otherwise, the application (or OS, on the left side) must be effectively recertified as conformant, and that implies re-enrollment.

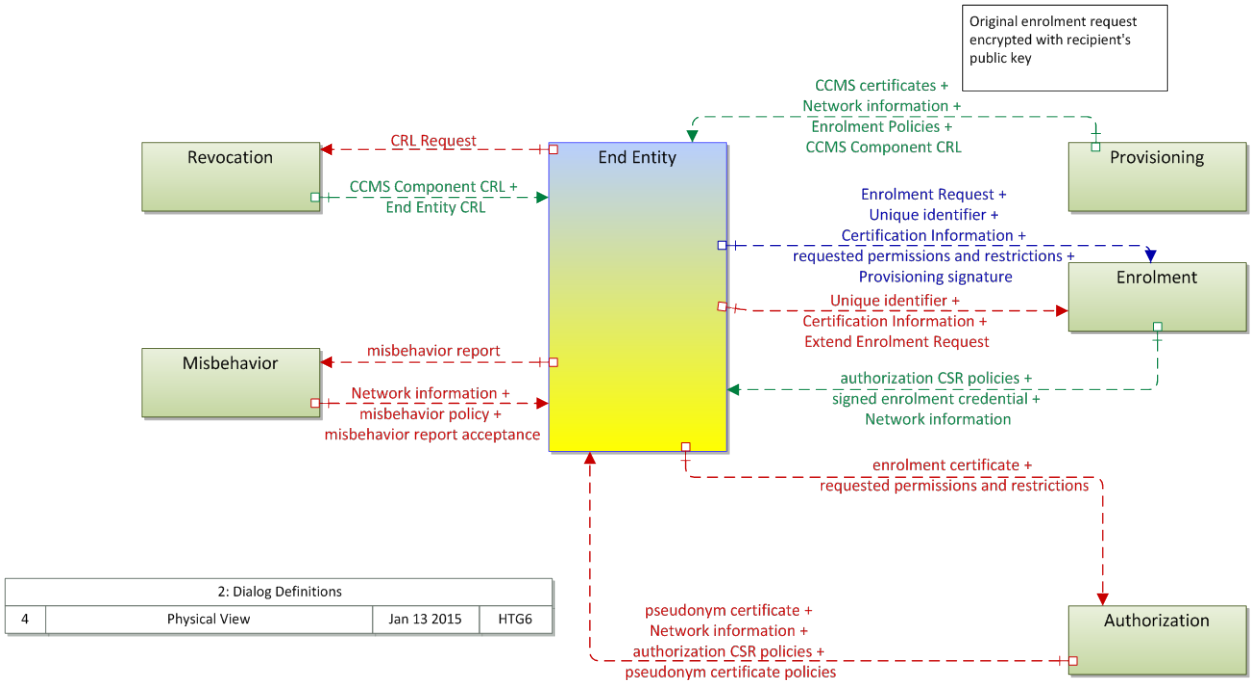
Figure 9: Application and Device Lifecycle



4.2.2 Summary of End Entity CCMS Interfaces

One of the results of the preceding CCMS analysis is the identification of all of the likely interfaces between end entities and CCMS. While not strictly relevant to the definition of CCMS, this by-product is highly relevant to anyone that might want to build a participating end entity. The following diagram is a synthesis of the preceding. Definitions of the information flows required here is a subject left for other efforts.

Figure 10: End Entity Interfaces



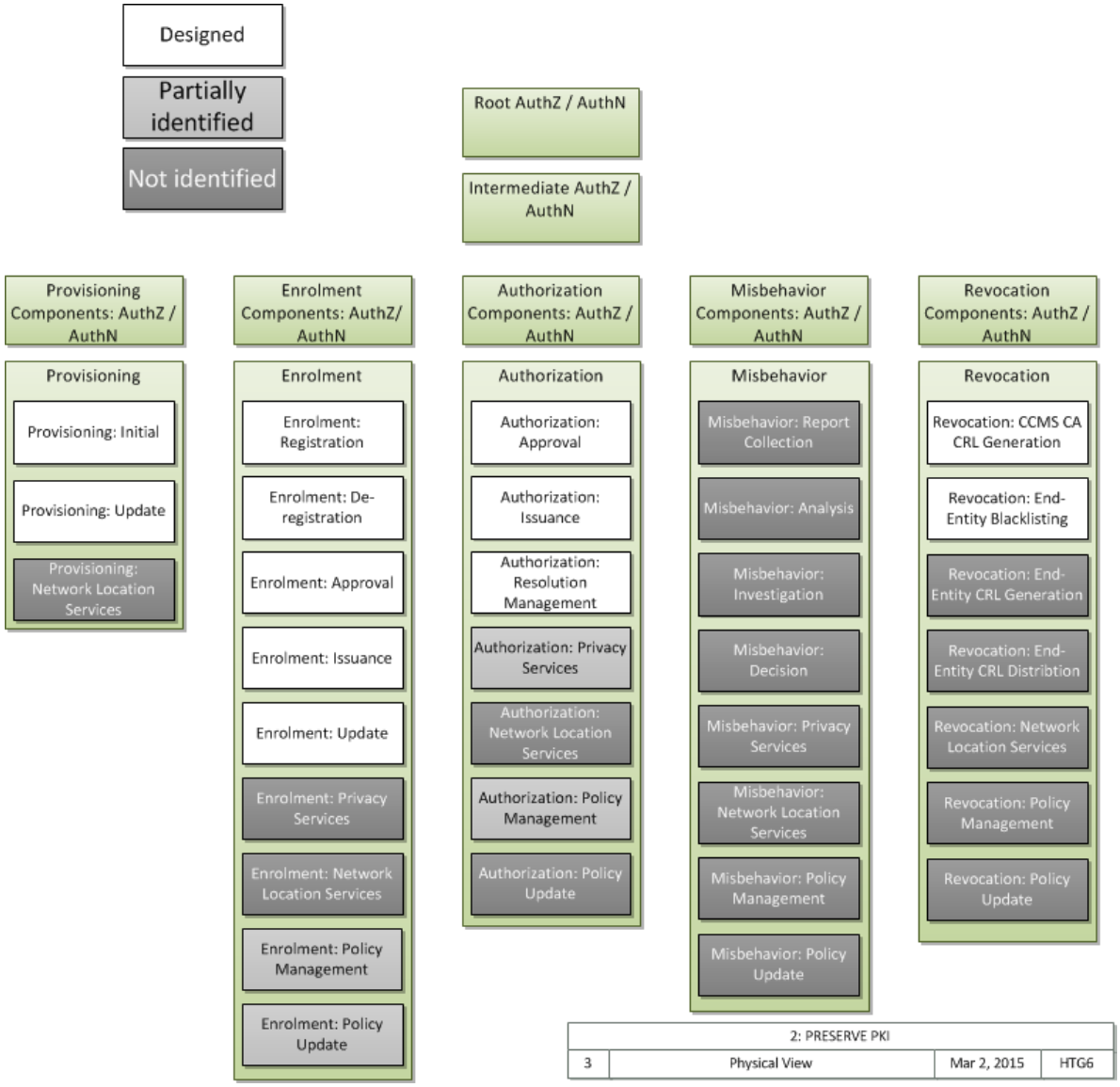
4.3 Examine Existing Designs

In this section we map the required functionality to what is known about existing designs based on the functionality described in Section 4.1.3

4.3.1 PRESERVE

The following diagram offers a baseline analysis of the PRESERVE PKI system.

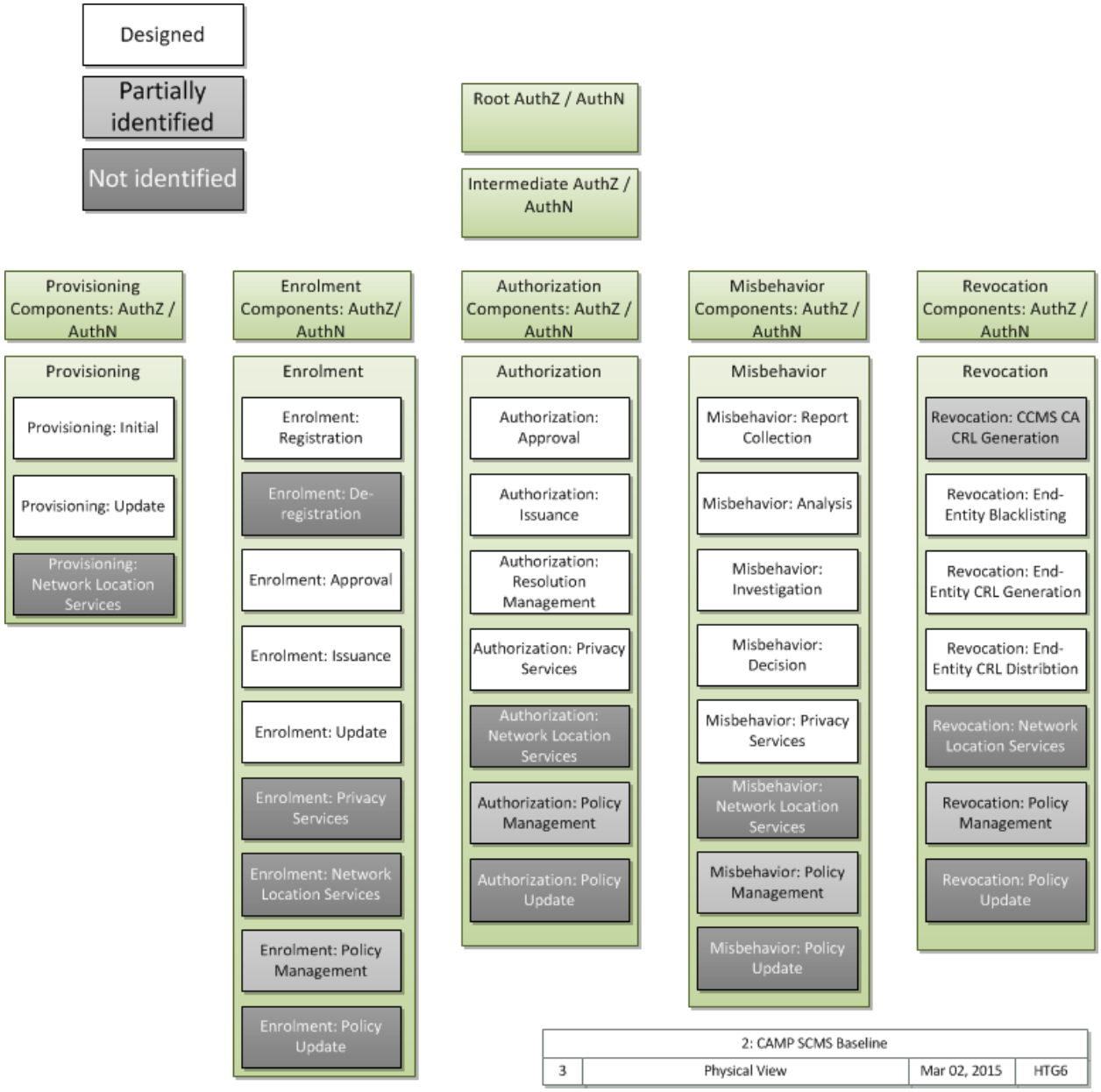
Figure 11: Baseline Analysis of PRESERVE PKI System



4.3.2 CAMP SCMS

The following diagram offers a baseline analysis of the SCMS PKI system developed by CAMP.

Figure 12: CCMS Baseline Analysis of CAMP SCMS



4.4 Identify Interfaces between Interacting CCMS

4.4.1 Information Flow Diagrams for Inter-CCMS Communications

The following two figures illustrate communications between CCMS. There are more than 70 information flows between CCMS, though the exact number may be more or less depending on the nature of trust between the CCMS.

Figure 13: Interfaces that support Inter-CCMS Trust

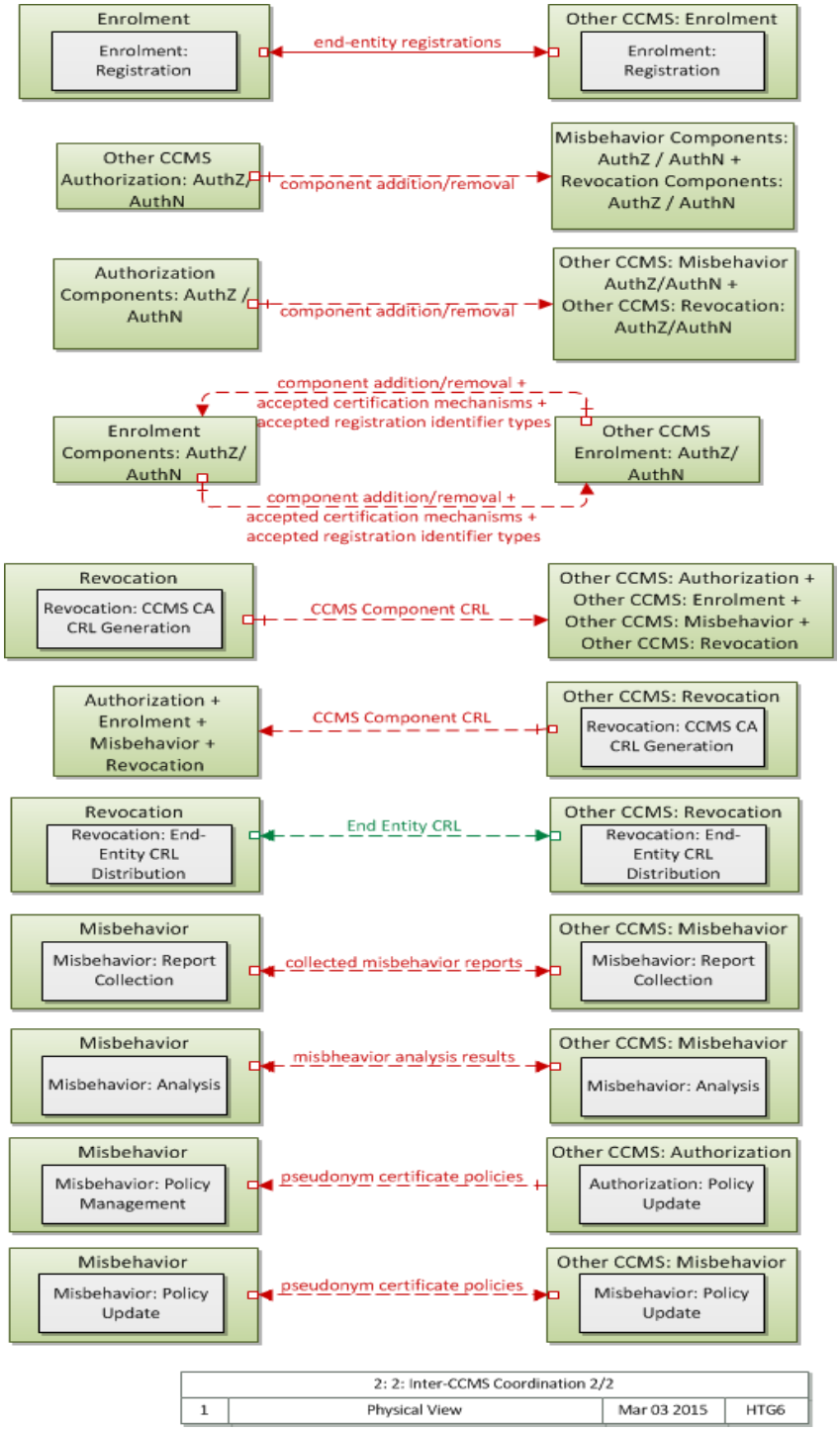
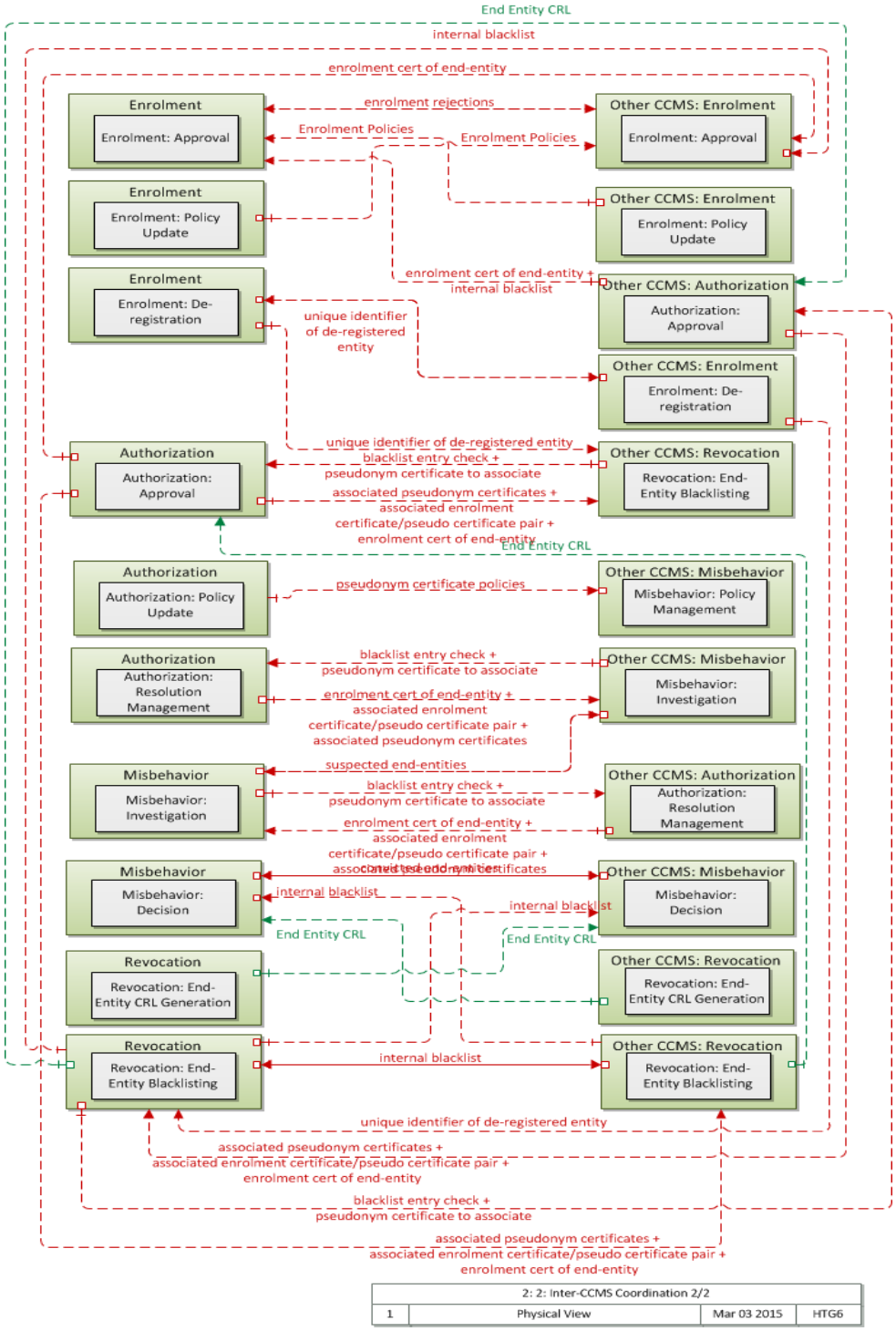


Figure 14: Interfaces that support Inter-CCMS Trust (Continued)



4.4.2 CCMS Coordination

In addition to the inter-CCMS communications noted above, there are use cases that require interaction between CCMS that are not covered under typical operations. The use cases identified below are more about managing a group of CCMS (e.g., modifying the top level domain's constituency or characteristics). As such, here the use cases are described singularly. These use cases will require some management and coordination functionality in addition to the functionality defined above, but the identification of that functionality is left as a future activity.

The presentation format is similar to section 4.1.5, except that the 'Use cases related' entry is omitted because these are entirely new use cases, not functions or components. The following use cases require additional inter-CCMS coordination *in addition to* what is noted above.

- **Add/Modify trust relationship to foreign CCMS**
  - Objective: To determine whether to add or change a trust relationship to another CCMS
  - Required behavior:
    - Receive information related to the CCMS to be added
    - Approve or reject to allow CCMS into the CCMS Federation
    - Provide vote to all other CCMS in the federation
    - Receive other CCMS votes
    - If foreign CCMS achieves enough votes, establish trust relationship
  - Material Input:
    - Foreign CCMS ID
    - Foreign CCMS privacy policy
    - Foreign CCMS certificate format
    - Foreign CCMS encryption algorithms supported
    - Foreign CCMS signing algorithms supported
    - Foreign CCMS root certificate if cross certification is supported
    - Other foreign CCMS policies
    - Rejection or approval from other federated CCMS
  - Material Output:
    - Rejection or approval of foreign CCMS trust modification
  - Existing designs:
    - CAMP SCMS: Not Present
    - PRESERVE PKI: Not Present
  - Applicable specifications or standards: None identified

- **Notification of certificate formats**
  - Objective: To exchange certificate formats and policies between CCMS, and to maintain

a record of that information

- Required behavior:
  - Provide certificate format information
  - Receive certificate format information
  - Provide policy information describing certificate issuance, certificate lifetimes, privacy protections including pseudonymity and insider defense protection, conditions of certificate revocation, CRL distribution frequency;
  - Receive and store record of all of same
- Material Input:
  - Certificate format information
  - Certificate issuance policies
  - Certificate revocation policies
  - Policy information describing certificate issuance
  - Certificate lifetimes
  - Privacy protections including pseudonymity and insider defense protection
  - Conditions of certificate revocation
  - CRL distribution frequency
- Material Output: same as material input
- Existing designs:
  - CAMP SCMS: Not Present
  - PRESERVE PKI: This functionality is provided by the Root CA and Enrollment CA of the PRESERVE PKI implementation, though not explicitly mentioned in the PRESERVE documents [2] [3] [7] or the PKI specification of the C2C-CC [1].
- Applicable specifications or standards: None identified

- **Notification of algorithms**
  - Objective: To exchange cryptographic algorithm information between CCMS, and to maintain a record of this information
  - Required behavior:
    - Provide algorithm description to other CCMS
    - Provide algorithm-specific parameters and similar distinct and specific relevant data to other CCMS
    - Receive and store record of all of same
  - Material Input:
    - Algorithm supported by this CCMS
    - Algorithm use; possible types:
      - Symmetric encryption/decryption

        - Asymmetric encryption/decryption
        - Hash algorithm
        - Signature generation/verification
        - Mechanisms for random number generation
    - Material Output: same as material input
    - Existing designs:
        - CAMP SCMS: Not Present
        - PRESERVE PKI: Not Present
    - Applicable specifications or standards: None identified

- **Notification of Certification Services**
    - Objective: To exchange certification service information between CCMS, and to maintain a record of this information
    - Required behavior:
        - Provide certification service credentials and applicability to other CCMS
        - Receive and store record of same
    - Material Input:
        - Credentials of certification service to add
        - Qualifications of certification service (what types of devices, what standards, when valid, performance references, etc.)
    - Material Output:
        - Same as material input
    - Existing designs:
        - CAMP SCMS: Not Present
        - PRESERVE PKI: Use Case identified (discussed in Protection Profile task group and Trust Assurance Level task group of C2C-CC [13])
    - Applicable specifications or standards: None identified

- **Notification of new CCMS component**
    - Objective: To exchange new component information between CCMS
    - Required behavior:
        - CCMS1 Enrollment/Authorization/Misbehavior/Revocation AuthZ/AuthN: compiles information about a new Enrollment/Authorization/Misbehavior/ Revocation component (see Material Input below) and transmits it to the relevant component(s) in CCMS 2
        - CCMS2 components: receive relevant information, validate it, store it, make it available for use if necessary

- Material Input:
  - Information about new enrollment component including: network location information, type of component, name, authorizations, certificate, particular policies if any, particular area of responsibility if any, time started, particular protocols used, restrictions on trust of that component
- Material Output: Information noted under 'material input' provided to components in CCMS2. For example, for a new Enrollment component:
  - relevant AuthZ/AuthN
  - End entity blacklisting: Allow the blacklisting of enrollment certificates from that enrollment component
  - Authorization: Inform authorization components that enrollment certificates from that enrollment component may be trusted
- Existing designs:
  - CAMP SCMS: Not Present
  - PRESERVE PKI: Not Present
- Applicable specifications or standards: None identified

- **Conditional pseudonym ID resolution**
  - Objective: To exchange sufficient information such that a pseudonym certificate identified by CCMS1 but issued by CCMS2 can be matched to an enrollment certificate issued by either CCMS1 or CCMS2
  - Required behavior:
    - CCMS1 and CCMS2 must have a trust relationship at either Enrollment or Authorization cert level
    - CCMS1 determine that a pseudonym cert needs to be associated with an Enrollment cert so that it may act on that Enrollment cert
    - CCMS1 complies resolution justification and pseudonym cert into a package for CCMS2
    - CCMS2 must associate pseudonym cert with Enrollment cert (if it issued Enrollment) or associate pseudonym cert with linkage values (if it provided Authorization and did not store Enrollment)
  - Material Input:
    - Pseudonym certificate to resolve (from CCMS1 to CCMS2)
    - Reason for resolution (from CCMS1 to CCMS2)
  - Material Output:
    - Matching enrollment certificate (from CCMS2 to CCMS1), if CCMS2 issued Enrollment cert

        - Matching authorization certificate linkage (could be linkage value, could be actual Enrollment cert, depending on technical implementation) information (from CCMS2 to CCMS1), if CCMS1 issued Enrollment cert
    - Existing designs:
        - CAMP SCMS: Not Present
        - PRESERVE PKI: Not Present
    - Applicable specifications or standards: None identified

- **Notify of performance metrics (exchanging information about aggregated registrations and other measures)**
    - Objective: To exchange information regarding aggregated registrations, misbehavior statistics and other related meta-data.
    - Required behavior:
        - Accumulate performance information
        - Receive and store foreign CCMS performance information
    - Material Input:
        - CCMS performance information
    - Material Output: same as material input
    - Existing designs:
        - CAMP SCMS: Not Present
        - PRESERVE PKI: Not Present
    - Applicable specifications or standards: None identified

- **Add Root CA Component**
    - Objective: To exchange information necessary for CCMS2 to trust a newly added CCMS1 root CA component
    - Required behavior:
        - CCMS1 Root CA: compiles information about a new Root component (see Material Input below) and transmits it to the relevant component(s) in CCMS 2
        - CCMS2 Root CA: receive relevant information, validate it, store it, make it available for use if necessary
    - Material Input:
        - Information about new root component including: network location information, certificate, particular policies if any, particular area of responsibility if any, time started, particular protocols used
    - Material Output: Information noted under 'material input' provided to components in CCMS2.
    - Existing designs:

        - CAMP SCMS: Not Present
        - PRESERVE PKI: Not Present
    - Applicable specifications or standards: None identified

- **Add Intermediate CA Component**
    - Objective: To exchange information necessary for CCMS2 to trust a newly added CCMS1 intermediate CA component
        - CCMS1 Root CA compiles information about a new Intermediate CA component (see Material Input below) and transmits it to the Root CA in CCMS 2
        - CCMS2 Root CA: receive relevant information, validate it, store it, make it available for use if necessary
    - Material Input:
        - Information about new Intermediate CA component including: network location information, name, authorizations, certificate, particular policies if any, particular area of responsibility if any, time started, particular protocols used, restrictions on trust of that component
    - Material Output: Information noted under 'material input' provided to components in CCMS2.
    - Existing designs:
        - CAMP SCMS: Not Present
        - PRESERVE PKI: Not Present
    - Applicable specifications or standards: None identified

4.4.3 CCMS Federation Scenarios

There several different levels of inter-CCMS could trust at, defined by what aspect of the foreign CCMS's end-entity management processes the CCMS trusts:

1. No trust
2. Registration level trust
3. Enrollment level trust
4. Pseudonym level trust
5. Root level trust (single root)

This information is particularly relevant for CCMS policy-setters and decision-makers. In order to answer the question, "who do I trust?", and the related "how much do I trust them?" questions, HTG6 considered the possible inter-CCMS trust relationships, their advantages and disadvantages. The following section describes the implications of the different levels of inter-CCMS trust.

Note that as an adjunct research area for HTG6, notional inter-CCMS trust scenarios need further review and enhancement. This was not a primary topic of study, but we believe it relevant for any scenario where C-ITS is deployed.

4.4.3.1 No Trust

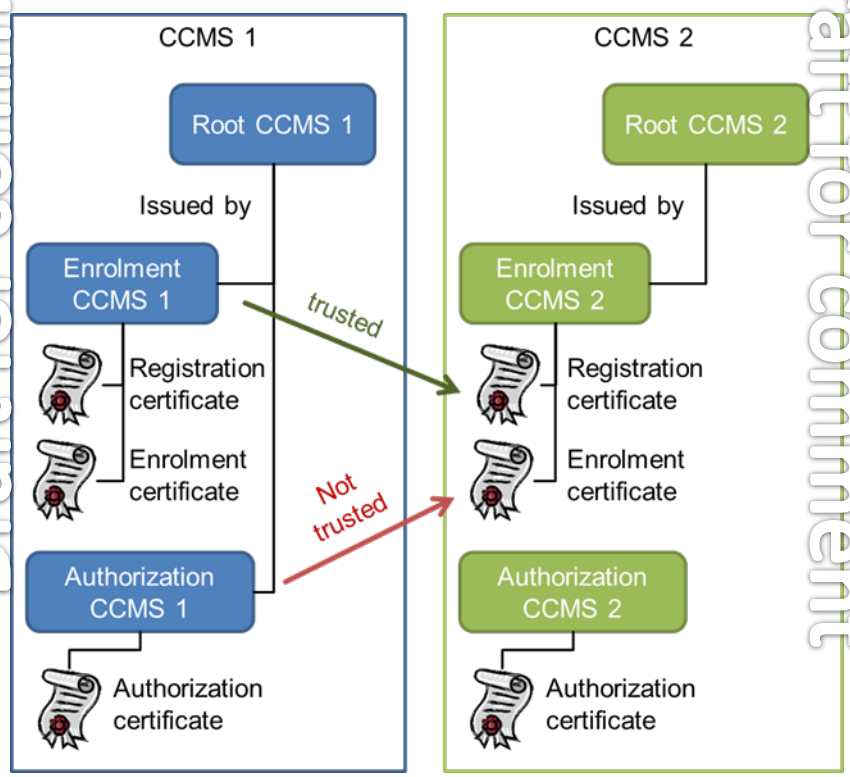
- No cross certification of roots
- End entity is registered in CCMS1 and CCMS2 with different canonical certificates
- End entity gets enrollment cert from CCMS1 and CCMS2
- End entity gets pseudonym certificates from CCMS1 and CCMS2
- CCMS1 do not trust certificates from CCMS2 and vice versa
- End entity needs to have root certificates of CCMS1 and CCMS2 in order to trust messages signed by pseudonym certificates from the other CCMS

4.4.3.2 Registration Level Trust

- No cross certification of roots
- End entity can be registered only in one CCMS, either CCMS1 or CCMS2. **The registration in one CCMS is trusted by the other CCMS.**
- End entity gets enrollment cert from both CCMS1 and CCMS2 based on the single canonical cert registered at one CCMS. **End entity needs to have two enrollment certificates.**
- End entity gets pseudonym certificates from both CCMS1 and CCMS2 based on the enrollment cert of the specific CCMS.
- CCMS1 only trusts canonical certificates from CCMS2 and vice versa
- End entity needs to have root certificates of CCMS1 and CCMS2 in order to trust messages signed by pseudonym certificates from the other CCMS

Draft for comment

Figure 15: Two CCMS trust each other on registration (canonical certificate) level



Draft for comment

4.4.3.3 Enrollment Level Trust

- No cross certification of roots
- End entity can be registered only in one CCMS, either CCMS1 or CCMS2
- End entity can have either CCMS1 or CCMS2. **The enrollment in one CCMS is trusted by the other CCMS. End entity needs to have only one enrollment cert from CCMS1 or CCMS2 but can have enrollment certificates from both CCMS.**
- End entity gets pseudonym certificates from CCMS1 and CCMS2 based on the enrollment cert of the CCMS that has issued the enrollment cert. In the process of requesting pseudonym certificates the CCMS can verify the enrollment cert when the enrollment component of the other CCMS is in the list of trusted entities.
    - If CCMS1 or CCMS2 have region restrictions then the enrollment cert cannot be used to authorize the request of pseudonym certificates for another region.
- CCMS1 only trusts enrollment certificates from CCMS2 and vice versa
- End entity needs to have root certificates of CCMS1 and CCMS2 in order to trust messages

signed by pseudonym certificates from the other CCMS

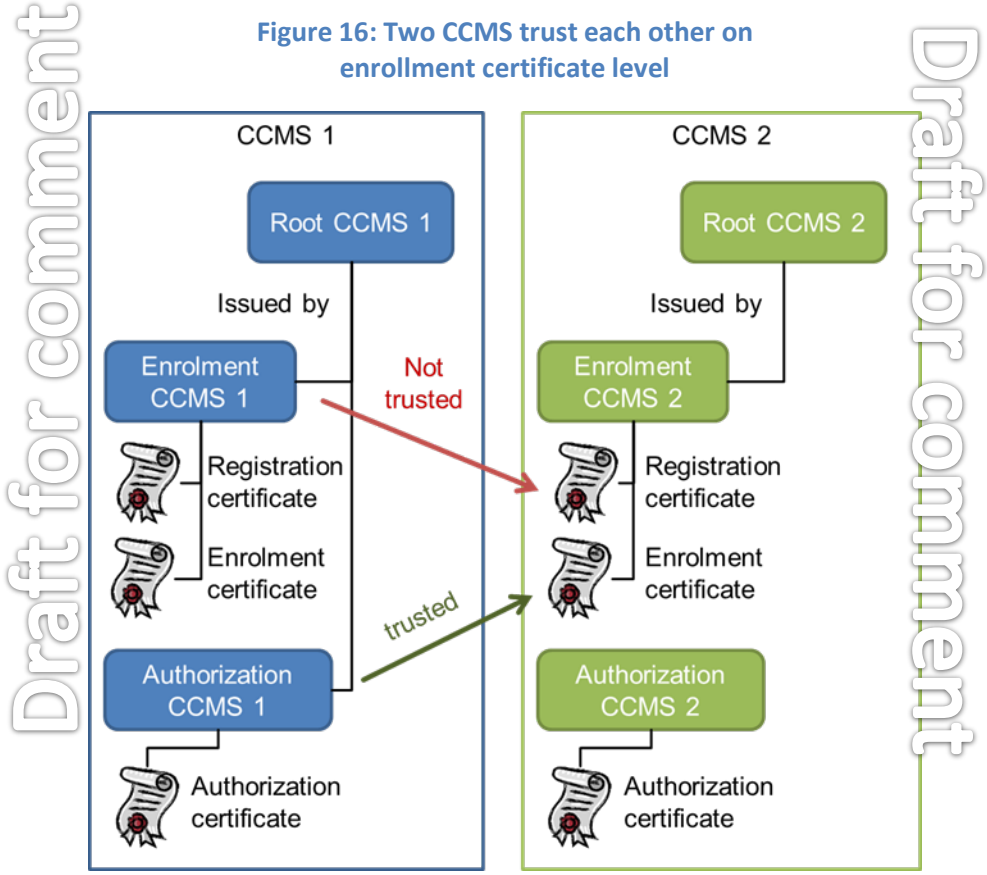


Figure 16: Two CCMS trust each other on enrollment certificate level

Draft for comment

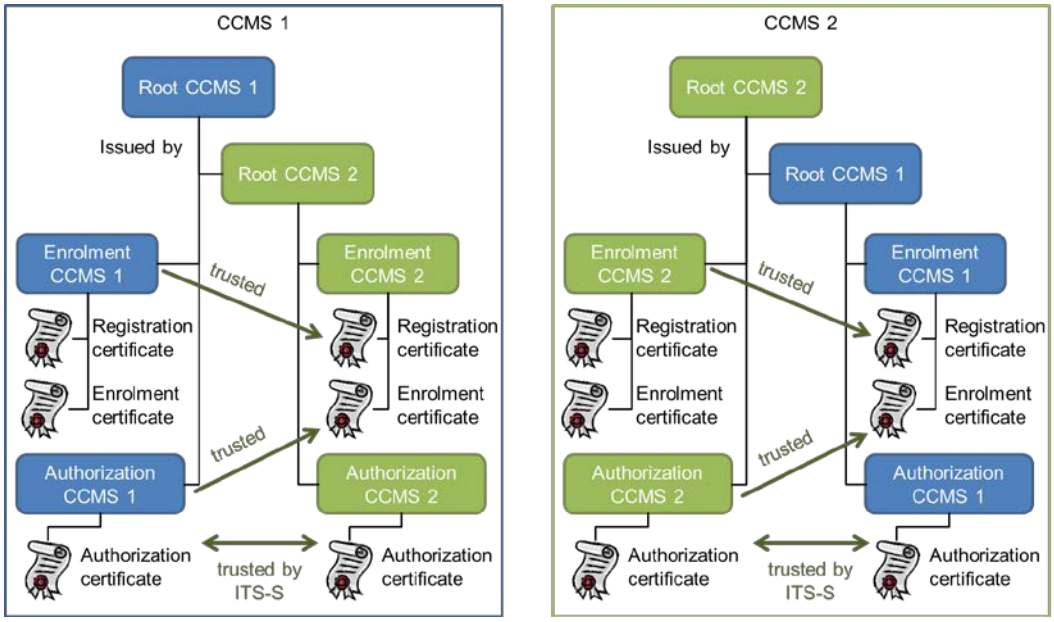
4.4.3.4 Pseudonym Level Trust

- Cross certification of roots or adding root cert of other CCMS to trust store
- It is required that the end entity is registered in one CCMS, either CCMS1 or CCMS2.
- It is required that the end entity has enrollment cert from one CCMS, either CCMS1 or CCMS2.
- End entity gets pseudonym certificates from CCMS1 or CCMS2. In the process of requesting pseudonym certificates the **CCMS can verify the enrollment cert of every CCMS due to the cross certification of roots or installation of root cert of other CCMS in trust store**.
    - NOTE: Policy might limit the number of pseudonym certificates that are valid for the same time period and location. If end entity is allowed to request pseudonym certificates from different CCMS then there has to be information exchange to ensure this policy.

- CCMS1 trusts **all** certificates from CCMS2 and vice versa due to the cross certification of the roots.
- End entity needs to have stored only one root cert of CCMS1 or CCMS2 in order to trust messages signed by pseudonym certificates issued by the other CCMS

Draft for comment

Figure 17: Two CCMS trust each other on pseudonym certificate level



Draft for comment

5 Conclusions and Recommendations

Existing CCMS designs architect much of the required functionality of a comprehensive CCMS. However, neither surveyed design has addressed many of the inter-CCMS communications aspects identified here. Neither has the standards community described the relevant interfaces. ***This lack of interface specification is among the biggest issues confronting CCMS designs and the long term C-ITS deployment.*** There are other issues however, all of which are summarized here along with HTG6's recommendations:

1. Most inter-CCMS interfaces are 1-to-many, so as the number of CCMS increase, the number of implemented interfaces increases in polynomial fashion. To mitigate against this complexity, ***the C-ITS community should minimize the number of CCMS*.**

2. There are a large number of inter-CCMS interfaces. In order to ensure consistency and scalability, the ***C-ITS community should develop standards to codify the interfaces between CCMS*.**

3. Certification of applications and devices, including the ways that ITS resources are accessed, is linked with the granting of digital credentials. Credential distribution and certification are distinctly different functions that may benefit from different operational, institutional structures, but since they are linked the requisite relationships are important to the overall success of C-ITS. ***A future harmonization activity should clarify the relationship between certification and credentials distribution.*** Absent harmonization, each implementer will have to perform this analysis, making inter-CCMS trust unlikely.

4. Devices and applications have a lifecycle, and in some phases of that lifecycle they have no relationship to the CCMS, i.e., prior to provisioning and subsequent to end-of-life. There are security-related policy concerns with how these end entities are handled in the times when they do not have a CCMS relationship. ***The C-ITS community should establish requirements defining how end entities must be developed and handled in order to obtain credentials, and establish shared requirements for proper disposal*.**

5. CCMS Components are responsible for managing the policies governing their operation, and disseminating that policy information to users. This sounds self-evident, but what it means is that the institutional framework governing the CCMS component is affected by and affects the technical design. Making changes to one or the other has a causative effect on the other. This also implies something about how CCMS enter in trust relationships with one another.

Depending on how much CCMS' trust one another, their policy frameworks must be compatible, and may need to be identical. The ripple effect here is that if one CCMS changes policy, it jeopardizes the trust relationship between CCMS. ***The CCMS management community should collectively define CCMS policy frameworks.***

6. There are several likely-viable Inter-CCMS trust scenarios, and they have significantly different implications for the architecture and management of CCMS. ***Inter-CCMS trust scenarios need additional study.***

The functional analysis of the CCMS led to several additional conclusions about the nature of CCMS:

- A CCMS operates in a three-dimensional environment: a CCMS's area of responsibility is defined by geo-political boundaries, time, and the applications it supports. See section 2.2 (Audience and Scope) for an illustration of this concept.
- Given disaster recovery and resultant responses and upgrades, ***a CCMS must be defined to exist in an environment where it must communicate with at least one other CCMS***: the original and a CCMS implemented as part of disaster recovery, or as part of a system upgrade. See section 3.1 (Define a Set of Minimal Functional Components for a CCMS) for a brief discussion.
- Since a CCMS' components interact with end entities at different phases of the end entity's lifecycle, CCMS that have different institutional structures are possible, so long as each independently managed component fulfills its requirements, including interfaces to other components in that CCMS. See the section 4.1.2 (End Entity Lifecycle) for consideration of the end entity lifecycle in terms relative to the CCMS.
- There are aspects of the end-entity lifecycle that do not involve direct interaction between the end entity and the CCMS, but about which the CCMS must be informed, and may generate secondary interfaces between CCMS and other systems. See section 4.2 (Consider End Entity Lifecycle Requirements) for aspects of the lifecycle that affect other related processes and procedures (such as certification or conformance testing) that may affect CCMS behavior but are not direct interactions between CCMS and end entities.
- CCMS components that manage policy must share their policies with end entities and also with other CCMS with which they have a trust relationship. See section 4.1.5 (CCMS Component Details: Functions, Interfaces and Trust Chain Elements) for illustrations of policy sharing between CCMS and end entities, and section 4.4 (Identify Interfaces between Interacting CCMS)

for a discussion of related inter-CCMS issues.

- There are several distinct degrees to which CCMS may trust one another; establishment and maintenance of this inter-CCMS trust is contingent on the exchange of policy information. See section 4.4.3 (CCMS Federation Scenarios) for a discussion of various inter-CCMS trust relationship possibilities.

- The long term viability of a given CCMS is dependent on its ability to evolve, in particular with regard to changes in cryptographic processes. Related, the *crypto-agility* of its supporting end entities much support such changes. ***Hardware-based crypto-processors should be discouraged, or designed so as to be easily replaced or upgraded***. Otherwise, end entities could be made permanently insecure in the event of a cryptographic failure.

6 References

[1]	Norbert Bißmeyer, Jan Peter Stotz, Hagen Stübing, Elmar Schoch, Stefan Götz, and Brigitte Lonc. A Generic Public Key Infrastructure for Securing Car-to-X Communication. In 18th World Congress on Intelligent Transportation Systems. ITS America, October 2011.
[2]	Norbert Bißmeyer, Sebastian Mauthofer, Jonathan Petit, Mirko Lange, Martin Moser, Daniel Estor, Michel Sall, Michael Feiri, Rim Moalla, Marcello Lagana, and Frank Kargl. PRESERVE D1.3 V2X Security Architecture v2. Deliverable, PRESERVE (PREparing SEcuRe VEhicleto-X Communication Systems) Consortium, January 2014.
[3]	Norbert Bißmeyer, Michael Feiri, Frank Kargl, Stelios Gisdakis, Martin Moser, and Panagiotis Papadimitratos. PRESERVE D5.2 Deployment Issues Report V2. Deliverable, PRESERVE (PREparing SEcuRe VEhicleto-X Communication Systems) Consortium, January 2013.
[4]	ETSI - European Telecommunications Standards Institute. Intelligent Transport Systems (ITS); Security; ITS Communications Security Architecture and Security Management. Technical Standard TS 102 940, ETSI, July 2012. v1.1.1.
[5]	ETSI - European Telecommunications Standards Institute. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. Technical Standard TS 102 941, ETSI, June 2012. v1.1.1.
[6]	ETSI - European Telecommunications Standards Institute. Intelligent Transport Systems (ITS); Security; Security Header and Certificate Formats. Technical Standard TS 103 097, ETSI, April 2013. v1.1.1.
[7]	Christophe Jouvray, Frank Kargl, Jonathan Petit, Panagiotis Papadimitratos, Michel Sall. PRESERVE D5.1 Deployment Issues Report V1. Deliverable, PRESERVE (PREparing SEcuRe VEhicleto-X Communication Systems) Consortium, August 2012.
[8]	W Whyte, A Weimerskirch, V Kumar, T Hehn , A security credential management system for v2v communications, Vehicular Networking Conference (VNC), 2013 IEEE, 2013
[9]	ITS Joint Program Office, Research and Innovative Technology Administration, U.S. Department of Transportation, Security Credential Management System Design: Security system design for cooperative vehicle-to-vehicle crash avoidance applications using 5.9 GHz Dedicated Short Range Communications (DSRC) wireless communications Draft Report —April 13, 2012. Available from http://www.its.dot.gov/meetings/pdf/Security_Design20120413.pdf
[10]	HTG6-9 Nomenclature document (Glossary)
[11]	HTG6-3: C-ITS Credential Management System Architectural Analysis and Recommendations for Harmonization
[12]	HTG6-6: Categorizing and Controlling Risk through Harmonized Security Policies
[13]	Alexander Kiening, Herve Seudie, Tyrone Stodart, Marko Wolf, Trust Assurance Levels of Cybercars in V2X Communication
[14]	Prepare Secure V2X Communication Systems (PRESERVE), http://www.preserve-project.eu, 2015

7 Terminology and Definitions

As mentioned in section 4.3 the content of this document is based on two CCMS designs (one in the EU, one in the USA). The following documents are the basis for the CCMS design in the EU.

- [1]: Describes fundamentals of the basic Public Key Infrastructure (PKI) as used by the European Car-to-Car Communication Consortium
- [2]: Describes the PKI as used in the European research project PRESERVE [14]. This PKI is based on the C2C-CC PKI described in [1] and is harmonized with the Pilot PKI implementation of the C2C-CC.
- [3], [7]: Describes the security lifecycle issues related to V2X communications as used in the European research project PRESERVE [14].
- [4], [5], [6]: Describes the PKI as described by the European Telecommunications Standards Institute (ETSI)

The following documents are the basis for the CCMS design in the USA, where it is known as the Security Credential Management System (SCMS).

- [9] is a specification for the credential management protocols as implemented within the Safety Pilot Model Deployment.
- [8] is an academic paper describing the key elements of the final proposal for the design, which differs in some respects from the Model Deployment design.

The following documents are other Harmonization Task Group 6 products, and are referenced throughout this report as appropriate.

- [10] is a list of terms and definitions that apply throughout all HTG 6 products.
- [11] is a comparison of existing domain-relevant Public-Key Infrastructure systems and a categorization of harmonization priorities by topical area.
- [12] is a risk analysis document that addresses mechanisms for defining the security needs of deployed applications and devices.

Selected definitions of critical nature to the reader of this report:

Application Ecosystem: A collection of components communicating to satisfy a business process and deliver a user service.

AuthZ = Authorization: being granted the ability to access resources.

AuthN = Authentication: verifying the user.

Cooperative ITS Credential Management System = CCMS: The components of the C-ITS environment that provide trust services: enabling authenticability, maintaining message integrity, service authorization and the exchange of data that must be kept confidential.

End Entity: A participant in C-ITS, this is a device that runs applications that are part of a C-ITS Application Ecosystem.