

TTA-15099-SD
발행일 : 2015년 11월 30일



oneM2M 어플리케이션 개발자 가이드 표준 **해설서**

Telecommunications Technology Association



한국정보통신기술협회
Telecommunications Technology Association



oneM2M 어플리케이션 개발자 가이드 표준 해설서

Telecommunications Technology Association

발간	한국정보통신기술협회 (TTA)
저자	송재승 (세종대학교) 최성찬 (한국전자부품연구원) Martin Ting Miao (한국전자부품연구원) 박현철 (세종대학교) 김재호 (한국전자부품연구원)
감수	류창호 (한국정보통신기술협회 프로젝트 매니저)

oneM2M 어플리케이션 개발자 가이드 표준 **해설서**



한국정보통신기술협회
Telecommunications Technology Association

본 문서에 대한 저작권은 TTA에 있으며, TTA와 사전 협의 없이 이 문서의 전체 또는 일부를 상업적 목적으로 복제 또는 배포해서는 안 됩니다.

Copyright© Telecommunications Technology Association 2015. All Rights Reserved

CONTENTS

제1장 소개	6
1.1 사물인터넷(Internet of Things, IoT)	7
1.2 oneM2M이란?	15
제2장 유스케이스	23
2.1 oneM2M 유스케이스 소개	23
2.2 원격 조명 제어 유스케이스	25
제3장 아키텍처 및 기능	28
3.1 원격 조명 제어 시스템 구조	28
3.2 구독(Subscription) 기능	29
3.3 그룹 관리 기능	30
3.4 검색 기능	31
제4장 프로시저	33
4.1 소개	33
4.2 메시지 흐름	34
4.3 원격 제어 시나리오	40
제5장 구현	42
5.1 소개	42
5.2 전제 조건	42
5.3 엔티티에 대한 주소 지정	43
5.4 조명 상태 데이터 모델링	43
5.5 리소스 구조	44
5.6 엔티티의 역할	46
5.7 구현 절차	47

제6장 결론	68
제7장 참고 문헌	69
제8장 Annex A Reading 리소스	71
8.1 Annex A.1 CSE resources	71
8.2 Annex A.2 게이트웨이 디바이스 어플리케이션 MN-AE	73
8.3 Annex A.3 조명 디바이스 어플리케이션	73
8.4 Annex A.4 스마트폰 어플리케이션 IN-AE	75
8.5 Annex A.5 접근 제어 정책	76
8.6 Annex A.6 컨테이너	77
8.7 Annex A.7 ContentInstances	79
8.8 Annex A.8 구독(Subscriptions)	81
8.9 Annex A.9 그룹	82

제1장 소개

사물인터넷에 대한 국제 표준을 개발하는 oneM2M 에서 Release 1 표준 문서들이 2015년 1월에 개발이 된 뒤에, 많은 사물인터넷 제조업체, 사업자, 개발자들이 oneM2M을 이용한 사물인터넷 어플리케이션 및 서비스의 개발에 관심을 보이고 있으며, 실제 상용화 서비스 및 제품들이 생산이 되어지기 시작했다. 하지만, oneM2M 표준 문서들의 경우 수백페이지의 분량으로 개발 되어있어, 사물인터넷 또는 oneM2M 국제표준을 처음 접하는 개발자들이 표준 문서들을 직접 보고 개발을 시작하는 것은 쉽지 않은 작업이다.

따라서, 본 문서는 oneM2M 규격의 서비스 플랫폼에서 제공되는 기능을 사용하여 어플리케이션을 개발하는 개발자들이 본격적인 개발을 하기에 앞서, 아주 간단한 사물인터넷 어플리케이션을 만들고 테스트해 볼 수 있는 해설서로써 다음의 내용들을 포함하고 있다.

- >> 제 1 장 : oneM2M의 기본적인 개요에 대해서 설명한다.
- >> 제 2 장 : oneM2M의 여러 유스케이스 중 간단하게 원격 조명 유스케이스를 다룬다.
- >> 제 3 장 : 원격 조명 제어 유스케이스의 아키텍처 및 기능에 대해서 간단하게 다룬다.
- >> 제 4 장 : 원격 조명 제어 유스케이스를 구현할 때 필요한 프로시저에 대해서 다룬다.
- >> 제 5 장 : 조명 제어 유스케이스의 구현에 필요한 절차에 대해서 다룬다.
- >> 제 6 장 : 본 해설서의 결론에 대해서 다룬다.
- >> 제 7 장 : 참고 문헌
- >> 제 8 장 : Annex A Reading 리소스

본 해설서에서 기준으로 삼은 oneM2M 표준 문서들의 버전은 oneM2M release1.0이며 작성할 때 참고한 각 규격들은 다음 표 <1-1>에 간단한 설명들과 함께 볼 수 있다. 요구사항 / 아키

텍처 / 프로토콜 / 보안 / 장치관리, 추상화 및 시맨틱 5개의 실무반에서 작성한 총 11개의 문서들을 참조하였으며, 각 표준 문서들은 oneM2M 웹사이트로부터 다운받아서 사용할 수 있다. 또한 본 해설서는 2015.11월에 개최된 oneM2M TP #20의 변경사항들을 반영하였으며, 특히 TS-0001 v1.4.0, TS-0009 v1.4.0 그리고 TS-0013 v1.2.0의 버전을 참조하여 작성되었다.

표 <1-1> oneM2M Release 1 표준 규격

워킹 그룹	기술규격	설명
요구사항 (WG1)	TS 0002 - Requirements TS [4]	요구사항
	TS 0011 - Common Terminology [13]	일반적인 용어 설명
아키텍처 (WG2)	TS 0001 - Architecture TS [3]	아키텍처 및 기능
프로토콜 (WG3)	TS 0004 - Core Protocol TS [6]	핵심 프로토콜을 비롯하여 oneM2M에서 사용되는 다양한 프로토콜
	TS 0008 - CoAP Protocol Binding TS [10]	
	TS 0009 - HTTP Protocol Binding TS [11]	
	TS 0010 - MQTT Protocol Binding TS [12]	
보안 (WG4)	TS 0003 - Security Solutions TS [5]	보안 및 프라이버시
장치관리, 추상화 및 시맨틱 (WG5)	TS 0005 - Management Enablement (OMA) TS [7]	장치관리, 추상화 및 시맨틱 가능성에 대한 연구
	TS 0006 - Management Enablement (BBF) TS [8]	
	TR 0007 - Study of Abstraction and Semantics Enablements [9]	

자료 출처 : www.onem2m.org/technical/published-documents

1.1 사물인터넷(Internet of Things, IoT)

본 절에서는 먼저 1.1.1 사물인터넷(IoT)이란 무엇인지에 대한 개념을 소개하고 이어서, 1.1.2 현재의 산업 동향 및 서비스에 대한 설명을 통해 실질적으로 다양한 산업에서 어떠한 서비스들이 개발되는 지에 대한 이해를 하고, 1.1.3 그에 따른 여러가지 표준 플랫폼들에 대해서 알아보고자 한다.

1.1.1 사물인터넷이란?

사물인터넷, 즉 IoT란 그림 <1-1>에서 보이는 것처럼 인터넷과 같은 연결성이 제공된 환경에서 각종 센서를 탑재한 디바이스들이 상호간 통신을 하면서 새로운 서비스를 창출하고 유용한 가치를 제공하는 기술을 의미한다. 사물인터넷과 유사한 개념으로 사물지능통신 (Machine to Machine,

M2M)이란 용어가 사용되어지기도 하는데, 이는 사물인터넷을 통신 관점에서 볼 경우 사물들간의 통신을 통해서 여러가지 데이터들이 교환되어 다양한 사물인터넷 서비스들이 제공되어질 수 있다는 기술 활용적 측면에서 발생한 것며, 기술적인 차이는 거의 없는 것으로 간주된다.

사물인터넷 실현을 위한 핵심 기술로는 네트워크 인프라 기술, 서비스 플랫폼 기술 등이 있다. 먼저, 네트워크 인프라 기술이란 Zigbee, Bluetooth, WiFi, 유선망, 이동통신망 등으로 구성된 네트워크 환경에서 연결성 보장 및 효율적 망 관리를 가능하게 하는 네트워크 계층의 기술을 의미한다. 특히, 사물인터넷을 위해 기존의 이동통신 서비스에서 사용되어지던 기술 이외에도 Low Power Wide Area (LPWA)와 같은 제한적인 디바이스들에 사용 가능한 네트워크 기술들이 최근들어 많이 소개되어지고 있다. 다음으로 서비스 플랫폼 기술이란 서비스 제공자가 사용자에게 다양한 사물인터넷 서비스가 가능하도록 하는 공통의 기반 기능들으로써 어플리케이션 계층의 기술이다.

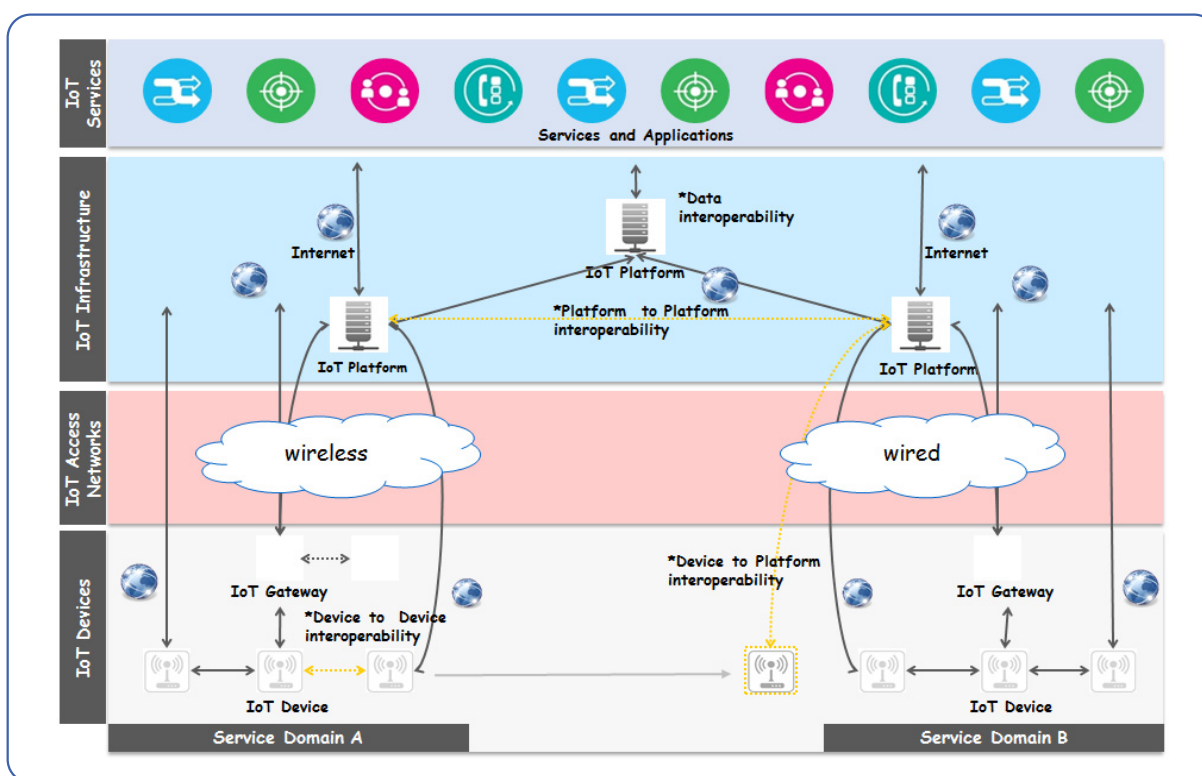


그림 <1-1> 사물인터넷의 개요

1.1.2 산업 동향 및 서비스

전세계적으로 인터넷과 연결되는 디바이스의 급격한 증가로 인해 사물인터넷 환경이 빠르게 확산될 것이라는 전망이 나오면서 글로벌 기업들은 경쟁적으로 사물인터넷 시장에 진출하고 있

으며 초기 시장 선점을 하기 위해 노력하고 있다. 이에 따라 한국, 미국, 일본, 중국 유럽 등 Information and Communications Technologies (ICT) 강국을 중심으로 사물인터넷 분야를 미래 핵심 성장 동력으로 전망하고 이의 육성을 위해 국가적 차원에서 지원하고 있다.

Gartner는 아래 표 <1-2>에서와 같이 2014년 37.5억대에 이르던IoT 기기가 2015년에는 48.8억대에, 2020년에는 250억대에 이를 것으로 예상하였다. 특히, 2020년에는 IoT 기기의 절반 이상이 소비자 부문에서 발생할 것으로 전망하였다.

표 <1-2> 부문별 IoT 기기 전망 (백만 대)

부문	2014년	2015년	2020년
자동차	189.6	372.3	3,511.1
소비자	2,244.6	2,874.9	13,172.5
포괄적(Generic) 비즈니스	479.4	623.9	5,158.6
수직적(Vertical) 비즈니스	836.5	1,009.4	3,164.4
총 기기 수	3,750.0	4,880.6	25,006.6

자료 : 미래창조과학부 정보통신정보화 및 정책지원 사업(IT통계조사 및 동향분석)의 연구결과 [1]

참고 : 자동차(서브시스템), 소비자(정보·엔터테인먼트, 홈오트메이션 등), 포괄적 비즈니스(에너지시스템, 자동시스템 등), 수직적 비즈니스(제조, 유틸리티, 운송 등)

Cisco 또한 2020년에 이르면 25억 명의 사람과 370억 개 이상의 사물이 인터넷으로 연결되고, 2030년에는 500억 개의 사물들이 연결되는 만물인터넷(Internet of Everything, IoE)으로 진화할 것으로 예측하였다.

이러한 전망을 바탕으로 국내·외 기업들 또한 다음과 같이IoT 기술을 여러 분야에서 활용하고 있다. 먼저 국내에서는 SKT, KT, LG U+ 등 이동통신 3사의 예를 들 수 있다.

>> SKT

SK텔레콤은 5개 분야 13개 아이템을 선보였으며, IoT 전용망 ‘저전력 장거리 통신 기술 (LPWAN : Low Power Wide Area Networks)’에 대한 시연 등도 준비하였다. LPWAN은 적은 전력으로 안정적 데이터 전송이 필요한 분야에 사용이 가능한 네트워크 기술이다. 또한 2015년 6월에는 통합형 사물인터넷 플랫폼인 씽플러그(ThingPlug)를 선보였다. 이는 IoT 국제표준인 oneM2M 표준을 준수하여 개발되었기 때문에 동일 표준을 따르는 수많은 기기 및 어플리케이션과 연동이 가능하며 기업과 개발자가 사물인터넷 서비스를 쉽게 이용할 수 있는 서비스 플랫폼 기능을 지원한다.

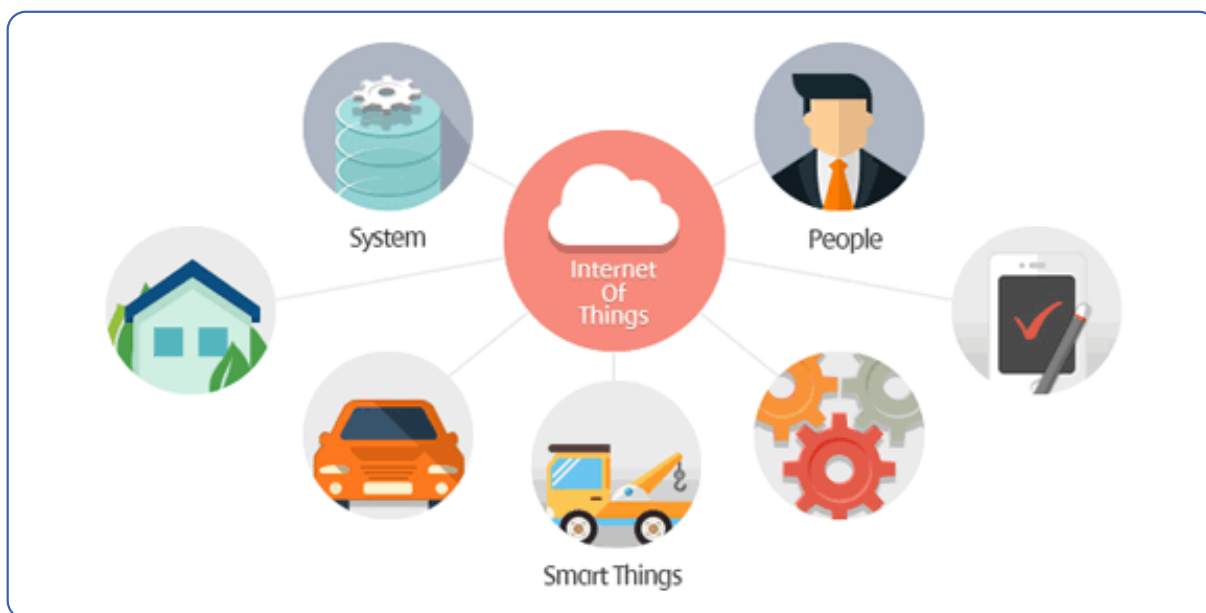


그림 <1-2> 통합형 IoT 플랫폼 ThingPlug

>> KT

KT는 ‘올레 기가 IoT 홈캠’ 서비스를 출시하였고, 스마트폰 앱으로 홈캠 단말이 촬영하는 영상을 실시간 모니터링할 수 있다. 위급 상황 시 앱의 ‘긴급출동’ 버튼을 누르면 안전요원이 출동한다.

>> LG유플러스

LG유플러스는 2014년에 이미 사물인터넷 상용 서비스를 시작하였으며, 특히 2015년 5월에는 IoT@Home 사업을 통해 신규 IoT 서비스 6종을 선보였다. 또한, 스마트폰으로 문을 열고 닫을 수 있는 유플러스 도어락을 개발하였고, 외부에서 충격이 있으면 알려주는 보안 기능까지 첨가했다. 산업 IoT 분야에서는 ‘유플러스 비즈 라이브컨트롤’을 공개했다. 고화질 영상을 실시간으로 전송해 원격지에 위치한 장비나 시설물을 관리할 수 있다. 긴급 장애 상황 등을 신속히 대응할 수 있는 것이 특징이다.

이 외에도 여러 기업들이 IoT기술을 통해 다음과 같이 서비스 중이거나 개발 중에 있다. 다음은 해외 IoT 기술을 가지고 서비스 및 플랫폼들에 대한 개발을 진행하고 있는 업체들에 대한 정보들을 간략하게 소개한다.

>> IBM 스마트 플래닛

IBM은 IoT의 일환으로 Smarter Planet 캠페인을 시작하였다. 이 캠페인은 세상의 모든 사물들을 기능화하고 서로 연결하여 지능화 함으로써 전 지구적으로 드러나고 있는 각 분야의 비효율과 낭비를 해결할 수 있고, 세계 경제의 성장 동력을 다시 회복하는데 기여하는 것이 목적이다.

>> Current Cost

Current Cost 사는 가정용 단말기(Envi)와 트랜스미터(Transmitter)를 이용하여 가정 내에서 사용하는 전기를 측정하여 이를 단말기 상으로 보여주는 In-home display를 개발하였다. 이 장치를 통해 사용자에게 사용하는 전기량 뿐만 아니라 한달 간 예상되는 전기요금, 전기를 절약하는 경향 등을 직접적으로 확인할 수 있는 기능을 제공한다.

>> AlertME

Alertme는 ZigBee 통신과 인터넷을 통하여 사용자가 스마트폰 및 원격지에서 PC를 이용하여 홈 네트워크에 접속하여 모니터링 할 수 있는 스마트 모니터링 기술과 가정에서 사용하는 에너지를 관리 및 제어할 수 있는 스마트 에너지 기술을 통하여 서비스를 제공하는 어플리케이션이다. 홈 네트워크에 접속하기 위한 소프트웨어(안드로이드, 아이폰, Window 계열 지원)와 홈 네트워크를 구성하고 있는 하드웨어(스마트 플러그, 스마트 미터리더, 스마트 카메라, 스마트 컨택트 센서 등)로 나뉜다.

>> Pachube("Patch-bay")

Pachube는 Usman Hague에 의해 인터랙티브 환경에서 센서들로부터 들어오는 실시간 데이터들을 관리할 목적으로 처음 소개되었다. 사람들과 사물, 어플리케이션, 그리고 IoT를 서로 이어주기 위한 목적으로 개발되었으며, 웹 기반 서비스를 통해 전세계의 데이터를 실시간으로 관리할 수 있고 등록된 사람들에게 전세계로부터 수집한 정보들을 공유하고 협업할 수 있는 환경을 제공해준다.

>> Touchatag

Touchatag는 RFID, 2D 태그, NFC, QR 코드 등을 이용하여 컴퓨터에 연결하는 근거리 USB

리더 및 리더 카드(스티커 타입)을 이용하여 사용자들이 물리적인 공간에서 사용하는 오브젝트들을 온라인 공간에 연결시켜 웹서핑, PC 연결 텍스트 메시지 전송 등을 실시간으로 사용할 수 있는 서비스를 제공하는 기술이다.

>> Nabaztag:tag

Nabaztag:tag는 Violet사의 Rafi Haladjian and Olivier Mevel가 개발한 ambient display device로서 WiFi를 통하여 인터넷 접근을 가능하게 하며 이를 통하여 일기예보 확인, 이메일 확인 등의 서비스를 제공하는 기술이다. MP3 파일과 메시지를 주고 받거나 재생 및 읽을 수 있는 기능, 기상정보, 주식시장 보고서, 뉴스 헤드라인, 알람시계, 이메일 알람, RSS 피딩, MP3 스트리밍 등의 온라인 서비스를 제공한다.

>> Mir:ror

Mir:ror는 Nabaztag:tag를 개발한 Violet의 개인용 RFID리더로서 개인용 컴퓨터의 USB포트로 연결하여 사용하도록 개발하였다. RFID 태그인 ztamp를 내장한 사용자 친화형 태그인 Nabaztag이나 ztamp를 Mir:ror에 가까이 하면 다양한 어플리케이션과 멀티미디어 콘텐츠를 실행하도록 연결하는 서비스를 제공한다.

>> Twine

MIT Media Lab의 David Carr와 John Kestner가 개발한 Twine은 사물을 사용자에게 문자 메시지, 이메일, 트위터로 연결해주는 가장 간단한 방법이다. 2.5인치 정사각형 모양의 육면체로 무선WiFi 이더넷 연결이 가능하며 내/외부 센서, 두 개의 AAA 배터리로 수개월을 독립적으로 작동할 수 있다.

>> GPS Shoes

GTX Corp은 알츠하이머나 치매환자들의 위치를 확인하고 모니터링할 수 있는 이동형 2-way GPS tracking이 가능한 신발을 개발하였다. GPS Shoes는 신발 안에 모션 센서를 장착하여 움직임이 없을 때는 스스로 파워-절약 상태로 전환되며 신발의 위치를 확인하고 모니터링하기 위한 포털을 GTX Tracking Portal페이지에 구축하고 구글 맵에 통합된 형태의 유료 서비스를 제

공한다. 또한 신발이 특정 위치에서 일정 거리 이상 벗어나면 자동으로 관련 사람에게 스마트폰이나 이메일로 상황을 알려주는 등 다양한 헬스케어 산업과 관련된 서비스를 제공한다.

>> Vitality GlowCap

Vitality GlowCap은 AT&T와 협력하여 개발한 지능형 약뚜껑으로 불빛, 오디오, 전화, SMS 메시지 등을 통해 정기적으로 약을 복용하고 있는 환자에게 정확한 시간에 약을 복용할 수 있도록 도와주는 IoT서비스의 일환이다. 사용자가 약뚜껑을 열게 되면 관련 데이터가 3G AT&T 통신 네트워크를 통해 Vitality로 보고가 되고 이는 연결되어진 의사 또는 데이터를 수신할 수 있도록 허용되어진 그룹에게 리포트 형태의 서비스를 제공한다.

>> People Power

People Power는 오픈 소스 기술을 이용하여 사용자들을 IoT 기반의 사물에 연결함으로써 사용자들이 사용하는 에너지를 절감하는 솔루션을 개발하였다. People Power가 제공하는 Energy Service Platform(ESP)는 클라우드 기반 소프트웨어 솔루션으로서 에너지의 실시간 사용 정보 및 이력을 IoT를 통해 연결된 기기들로부터 읽어들이어 사용자들에게 에너지를 절감하기 위한 방법을 제공한다.

1.1.3 사물인터넷 표준

앞 절에서 본 것처럼 여러 기업들이 IoT 기술을 이용한 서비스를 시행 중이거나 준비 중이지만 기존에 상용화되어진 대부분의 플랫폼들은 국제 표준을 준용하지 않고 독자적으로 개발하였기 때문에, 호환성이라는 문제점이 존재하였다. 이는 새로운 사업 또는 서비스를 개바르할 경우, 기존의 솔루션 및 플랫폼을 재사용하지 못하고 새롭게 다시 개발해야 했으며, 제조업체들 역시 사업자들 별로 새로운 사물인터넷 디바이스들을 개발해야 하는 문제점을 야기 했다. 따라서, oneM2M을 비롯하여 IEEE, IIC, AllSeen Alliance, OIC 등 다양한 표준 단체들은 사물인터넷의 표준화를 위하여 다음과 같은 표준 활동을 진행 중이다.

>> oneM2M

2012년 8월 글로벌 사물인터넷 서비스 플랫폼 표준 개발을 목표로 설립되었으며, 사물인터넷 표준화 기구로는 세계 최대 규모이다. 본 해설서에서 다루는 Release 1을 개발하고 난 뒤에, 시맨

틱스, 다양한 Interworking 기술들, 개발자 가이드 등을 포함하는 Release 2 표준 개발을 현재 한창 진행중에 있다. 또한, ITU-T, IEEE, W3C 등과 밀접하게 협력을 통해 사물인터넷 글로벌 표준으로 점차 자리 잡아가고 있으며, 다양한 상용 서비스 및 디바이스들이 개발되어지고 있다. 본 해설서에서 다룰 표준 플랫폼이기도 하며 더 자세한 내용은 다음 절에서 보다 다뤄보도록 하겠다.

>> IEEE 표준 그룹

2014년 6월, IEEE는 P2413 프로젝트를 공식 개시함으로써 IoT 아키텍처 구축을 통해 더욱 다양한 산업과 기술 영역으로서의 확장에 나섰다. 프로젝트에 참여 중인 개발업체와 조직은 Cisco Systems, Huawei, GE, Oracle, Qualcomm, ZigBee Alliance 등이 있다.

IEEE P2413 프로젝트는 홈 자동화 및 산업 시스템 등 향후 IoT가 적용될 것으로 기대되는 부문의 커넥티드 기기와 어플리케이션들 간의 상호운용성을 담보하는 상위 수준의 아키텍처 프레임워크를 구성하는 것을 목표로 표준을 개발하고 있다.

>> IIC(Industrial Internet Consortium)

IIC는 IoT의 산업용 사용을 위한 표준 개발을 목표로 AT&T, Cisco, GE, Intel 및 IBM에 의해 2014년 3월에 설립되었다. 현실 세계와 디지털 세계의 통합을 개선, 빅데이터에 대한 더 나은 액세스를 지원하기 위하여 테크놀로지 사일로 장벽을 허무는데 초점을 두고 있다.

>> AllSeen Alliance

2013년 12월, Qualcomm과 Linux Foundation은 Cisco, Microsoft, LG전자, HTC, Panasonic, AT&T등과 함께 AllSeen Alliance라는 표준 단체를 결성하였다. AllSeen Alliance는 IoT용 프로덕트, 시스템 및 서비스의 광범위한 채택을 위한 오픈소스 IoT를 제공하는 오픈 소스 비영리 컨소시엄으로, 소비자 전자 제조업체, 자동차 업체, IoT 클라우드 공급업체, 혁신적 신생 기업, 칩셋 제조업체, 서비스 제공업체, 소프트웨어 개발업체 등 기업이 참여 중이다.

최근 AllJoyn을 이용한 상용 제품이 연이어 출시되고 있는데, LG전자와 Panasonic이 일부 TV와 무선 스피커 시스템에 AllJoyn을 적용했으며, 스마트 조명 회사인 LIFX도 AllJoyn을 채택한 제품을 개발하는 등 자동차, TV, 음향시스템, 스마트빌딩, 모바일 어플리케이션 등 20여 개의 제품에 AllJoyn이 적용, 기출시 혹은 출시될 예정이다.

>> OIC(Open Interconnect Consortium)

OIC는 Intel, Atmel, Dell, 삼성전자, Wind River등이 2014년 7월 27일 설립하였다. 비영리기관인 OIC는 IoT를 구성할 수십억 개의 기기 간의 연결 요건 정의 및 상호운용성 보장을 목표로 하고 있다. 이를 위해 펌팩터, 운영시스템 및 서비스 공급자에 관계없이 기기 간의 정보 흐름을 무선으로 연결하고 지능적으로 관리할 수 있도록 연결성에 중점을 둔 표준을 개발할 계획을 가지고 있다.

>> OASIS

OASIS는 2013년 MQTT를 M2M/IoT를 위한 표준 프로토콜로 선정하고 해당 스펙을 공개하였다. MQTT란 Message Queue Telemetry Transport의 약자로 지연 및 손실이 심한 네트워크 환경에서 검침기, 센서 등 작은 기기들의 신뢰성 있는 메시지 전달을 위해 IBM에서 1999년 개발한 메시지 프로토콜이다.

원격 장치 모니터링을 위해 데이터 수집 목적으로 개발되었으며 동작 시에는 브로커라는 중계서버를 기반으로 M2M/IoT 기기들 간의 Pub/Sub관계를 맺고, 이를 바탕으로 기기들 사이에 전달되는 메시지를 손실 복구 기능을 통해 전달하는 구조를 가지고 있다. [4]

1.2 oneM2M이란?

본 절에서는 1.2.1 oneM2M의 개요, 1.2.2 기존의 플랫폼과 비교했을 때 oneM2M표준 플랫폼의 장점, 1.2.3 약어 및 정의에 대해서 다루고자 한다.

1.2.1 oneM2M 개요

2015년 현재 사물인터넷 국제 표준의 중심에 서 있는 것이 oneM2M 표준이다. oneM2M은 각국의 표준화 기관들이 모여 글로벌 사물인터넷 표준을 위해 2012년에 결성된 표준 협의체로, 한국의 TTA를 비롯하여 미국의 TTA 및 ATIS, 유럽의 ETSI, 일본의 TTC 및 ARIB, 중국의 CCSA까지 총 7개의 표준 단체가 참여하고 있고, 전 세계의 주요 회사들이 표준단체를 통해 함께 참여하고 있다. 협약에 의하면 참석한 각 표준단체는 각 지역의 개별 사물인터넷 또는 M2M 표준을 만들지 않고, oneM2M 표준만을 준수하기로 하여 그 효용을 높이고 있다. oneM2M은 스마트 홈, 스마트 카, 스마트 헬스케어 등 다양한 사물인터넷 응용 서비스를 지원할 수 있는 공통

플랫폼(Common Services Platform)을 정의하는 것이 주된 역할이다. oneM2M에서는 공통 서비스 기능이 포함된 엔티티를 하부 네트워크 기술과 독립적으로 설계하고, 그들과의 인터페이스에 대해 정의하고 있다. 지원하는 프로토콜에는 HTTP, CoAP, MQTT, XMPP, Websocket 등이 있으며, 2014년 8월에 Release 1.0이, 2015년 1월에 Release 1.0.1이 발표되었다.

사물인터넷 서비스에서 공통으로 필요로 되어지는 기능들을 모아놓은 엔티티인 Common Service Entity (CSE)는 물리적으로 서비스 프로바이더가 제공하는 서버 인프라 시스템뿐만 아니라 센서, 디바이스에 올라가는 미들웨어 스택 및 액세스 네트워크와 연결하는 게이트웨이의 미들웨어 스택으로 구현이 가능하다. 즉 oneM2M의 표준화 대상은 서버, 게이트웨이, 디바이스에 탑재될 수 있는 미들웨어 스택에 대한 표준화를 진행하는 것이고 이를 통해서 서버, 게이트웨이, 디바이스 전 영역을 기반으로 사물인터넷 서비스를 제공하고자 하는 것이다.

공통 플랫폼을 소프트웨어 레이어의 관점에서 보면 해당 Common Service Entity(CSE)는 M2M 어플리케이션 프로그램과 기기의 오퍼레이팅 시스템 소프트웨어 사이에서 동작되는 소프트웨어로 정의할 수 있다. 다양한 버티컬 어플리케이션에서 요구되는 기능들을 제공하기 위한 수평적 플랫폼의 역할을 담당하며 해당 기능으로서는 원격 설정, 동작 지시, 연결, 데이터 수집, 데이터 보관, 디바이스 관리, 보안 등의 공통적인 기능을 제공함으로써 M2M 디바이스 및 M2M 어플리케이션 사이에서 서비스 제공을 위한 종단간 데이터 전달 및 컨트롤 서비스를 지원한다.

1.2.2 oneM2M 표준 플랫폼의 장점

기존의 서비스 프로바이더 및 어플리케이션에 종속적인 플랫폼들과 비교했을때, oneM2M 표준을 따르는 플랫폼이 제공해주는 장점은 다음과 같다.

- » 기존에 각 서비스 프로바이더 별, 어플리케이션 별로 종속적 개발되었던 사물인터넷 플랫폼의 파편화를 방지할 수 있다. 그리고 수평적 공통 플랫폼의 도입으로 규모의 경제를 통한 사물인터넷 산업의 활성화를 제공할 수 있다.
- » 수평적 공통 플랫폼의 도입은 사물인터넷 어플리케이션 서비스 개발의 복잡성을 낮추어 주는 효과가 있다. 이는 공통 플랫폼을 통해서 어플리케이션이 요구하는 다양한 기능들을 공급할 수 있기 때문이고 이를 통해서 서비스 프로바이더는 서비스 개발 시에 개발 비용을 감소시킬 수 있다.
- » 수평적 공통 플랫폼은 농업, 물류, 의료, 자동차, 가전 등 서비스 및 산업 환경에 구애받지 않으며 소프트웨어적으로 구현된 컴포넌트를 재사용할 수 있기 때문에 이를 통한 개발 비용과 더불어 Trouble Shooting을 위한 운영비용을 감소시킬 수 있다.

- » 표준화된 수평적 공통 플랫폼은 하부의 네트워크를 액세스하고 해당 네트워크에서 제공하는 서비스를 활용하는 부분에서 전세계 네트워크를 유용하도록 설계되므로 서비스 및 사업기회의 확장성을 제공해준다.
- » 사물인터넷 공통 플랫폼에 대한 운영적 측면에서 표준화된 기술은 관련 산업 및 서비스 프로바이더 관계자들에게 공통의 이해도를 제공하고 따라서 기술에 대한 운영 측면에서 문제발생에 대한 대처 및 문제에 대한 보편화된 솔루션을 제공하므로 운영 비용 감소를 가져올 수 있다.

즉, oneM2M 표준 플랫폼을 통한 장점은 다음 세 가지 키워드로 정리할 수 있다. 사물인터넷 플랫폼의 파편화 방지, 이를 통한 개발 및 운영 비용 감소와 그에 따른 산업 활성화이다. 그리고 이러한 장점을 기반으로 oneM2M 사물인터넷 공통 서비스 표준 플랫폼이 개발된 것으로 볼 수 있다.[2]

1.2.3 oneM2M 아키텍처

oneM2M 공통 플랫폼은 레이어 아키텍처 스타일로서 표현하면 상단의 M2M 어플리케이션 레이어와 하단의 M2M 네트워크 서비스 레이어 사이에 위치하고 있으며 네트워크 서비스 레이어가 제공하는 네트워크 서비스를 유용하면서 M2M 기반 다양한 어플리케이션들에게 공통 기능을 제공하기 위한 공통 서비스 레이어로 표현될 수 있다.

또한 그림 <1-2>와 같이 oneM2M 플랫폼 아키텍처는 기능 엔티티 모델로 표현이 가능하며 센서, 액추에이터, 디바이스, 게이트웨이가 위치하는 필드 도메인 영역과 서비스 프로바이더의 서버가 위치하는 인프라스트럭처 도메인 영역으로 구분이 가능하다. 또한 각각의 도메인에서 계층적 아키텍처 모델에서 살펴본 것과 같이 어플리케이션 레이어에 위치하는 Application Entity (AE), 공통 서비스 플랫폼 레이어에 위치하는 Common Service Entity (CSE), 그리고 네트워크 서비스 레이어의 Network Service Entity (NSE)로 분리하여 기능 모델을 정의하고 있다. 그림 <1-3>에서와 같이 필드 도메인 및 인프라스트럭처 도메인에서 위치하는 기능 엔티티라고 명명되는 각각의 엔티티에 대한 정의는 다음과 같다.[2], [5]

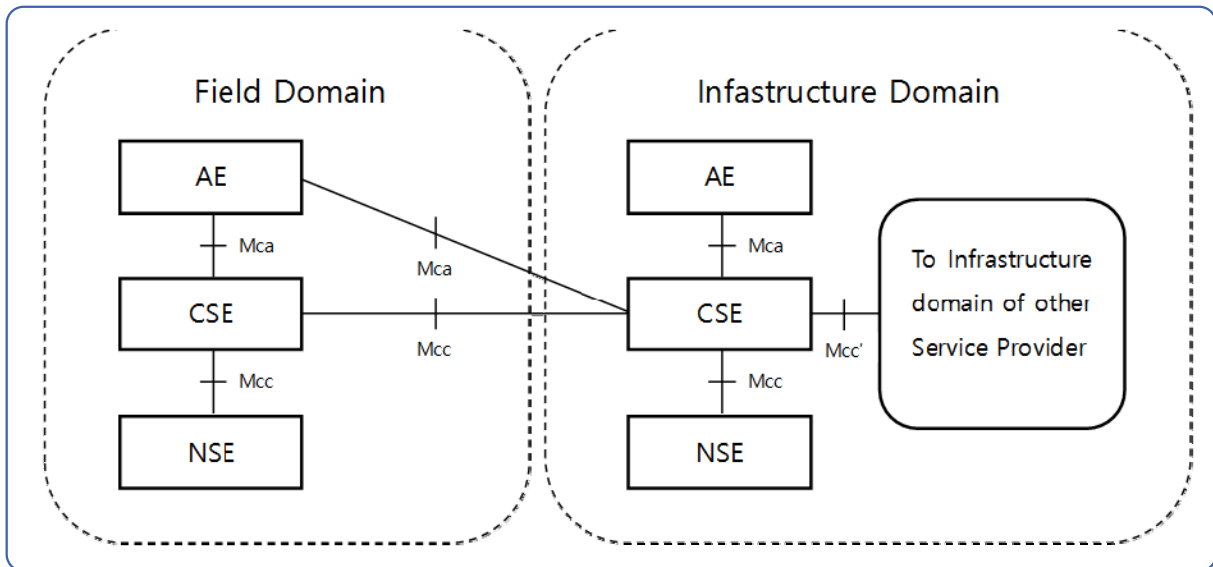


그림 <1-3> oneM2M 공통 플랫폼 기능 모델

» **Application Entity (AE)** : M2M 서비스를 제공하기 위한 어플리케이션 기능 로직을 포함하는 논리적인 엔티티를 의미하며 각각의 AE는 유일한 AE 식별자인 AE-ID로 구별된다. AE에 대한 예로서는 관제 시스템, 스마트그리드 시스템, 헬스케어 시스템을 위한 어플리케이션 등이 고려되어질 수 있다.

» **Common Service Entity (CSE)** : oneM2M 서비스 플랫폼에서 공통적으로 제공되어야 하는 공통 서비스 기능을 제공하는 부분으로서, 컴퓨터 시스템에서 미들웨어 소프트웨어에 해당한다. oneM2M 에서 정의한 CSE에는 총 12개의 Common Service Function (CSF) 공통 서비스 기능을 포함하고 있다. CSF는 CSE에서 제공되어야 하는 기능들 중 비슷한 기능들을 그룹화 한 논리적인 그룹으로, 각각의 CSF 들은 이후에 설명되어질 리소스를 통해서 외부에 노출되어 서비스를 제공할 수 있다. 해당 CSE는 AE를 구별하는 방식과 마찬가지로 CSE의 식별자인 CSE-ID를 통해서 유일하게 식별 가능하다.

» **Network Service Entity (NSE)** : CSE가 위치한 미들웨어의 하부 네트워크 서비스에 대한 추상화 영역으로 CSE에게 네트워크 서비스를 제공한다. 제공 가능한 네트워크 서비스의 예로서는 디바이스 관리, 위치관리, 3GPP 이동통신망에 연결되어있는 IoT 디바이스 트리거링 서비스 등이며 해당 서비스를 위해서는 네트워크 서비스가 관여되어진다.

또한 그림 <1-3>에서 나타난 각각의 엔티티들 간의 인터페이스를 의미하는 참조 포인트는 'Mca, Mcc, Mcn, Mcc' 으로 4개의 참조 포인트가 정의되고 있으며, 해당 참조 포인트를 통해서 AE, CSE, NSE 들 간의 메시지 교환과 같은 통신이 이뤄진다.

» **Mca (M2M Communication with AE) 참조 포인트** : AE 와 CSE 간의 포인트를 가리키며, 해당 AE 가 CSE에서 제공하는 공통 서비스 기능을 이용하기 위한 API 의 연결 포인트이고, CSE 와 AE간의 통신을 위한 연결 포인트이다.

- » **Mcc (M2M Communication with CSE) 참조 포인트** : 두 개의 CSE간의 포인트를 가리키며, CSE와 다른 CSE간의 서비스 공개 및 통신을 가능하게 하는 연결 포인트이다.
- » **Mcn (M2M Communication with NSE) 참조 포인트** : CSE와 NSE간의 포인트를 가리키며, CSE가 NSE에서 제공되는 네트워크 서비스 기능을 이용할 수 있는 연결 포인트이면서 네트워크 망으로의 데이터 전달 연결 포인트이다.
- » **Mcc'(M2M Communication with CSE of different M2M Service Provider) 참조 포인트** : 서로 다른 서비스 프로바이더에 종속적인 CSE간의 포인트를 가리키며, 서비스 프로바이더 간 CSE사이의 서비스 공개 및 통신을 지원하는 연결 포인트이다.

oneM2M 아키텍처 모델은 필드 도메인, 인프라스트럭처 도메인에 위치하는 CSE와 AE의 역할 및 기능에 대한 구분을 기반으로 4종류의 노드타입을 정의하고 있다. 이는 해당 그림 <1-4>에 oneM2M 노드 구성모델에 표시되어 있다.

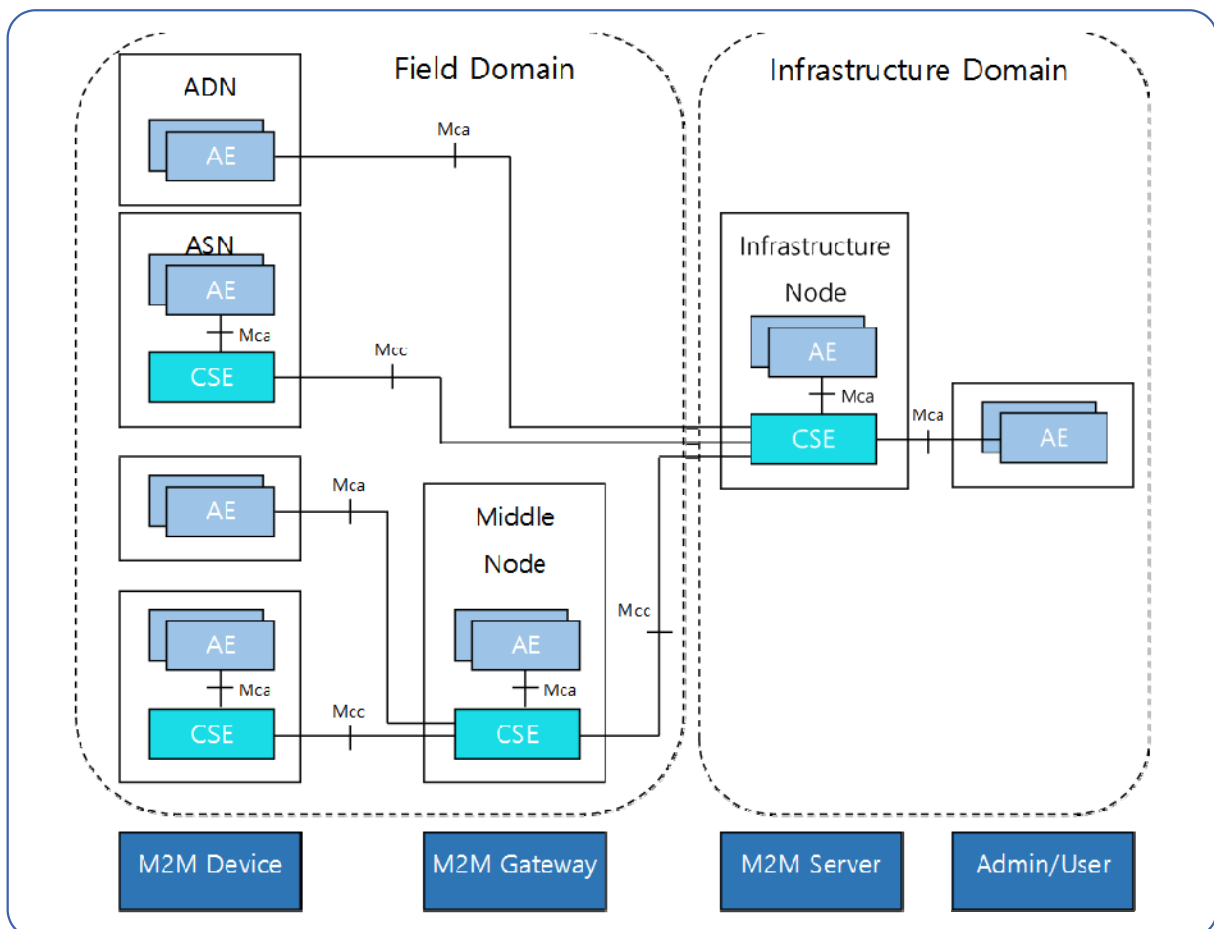


그림 <1-4> oneM2M 노드 구성 모델

- » **Infrastructure Node (IN)** : 인프라스트럭처 도메인에 위치하고 있는 IN-CSE를 포함하는 서버단에 위치하는 기기를 의미한다. 해당 IN은 서비스 프로바이더 당 한 개의 IN을 지원하는 것으로 정의되며 IN은 한 개의 CSE로만 구성이 되거나 1개 이상의 AE를 포함하는 형태로 구성되어질 수 있다. 논리적 기기인 IN에 매칭되는 물리적 기기로는 서버를 예로 들 수 있다. IN은 Mcc 참조 포인트를 통해서 한 개 이상의 MN과 한 개 이상의 ASN과 연동되며 Mca 참조 포인트를 통해서 한 개 이상의 ADN과 연동될 수 있다. Mcc' 을 통한 연동은 다른 서비스 프로바이더 영역에 위치한 IN노드와 연동된다.
- » **Middle Node (MN)** : 필드 도메인에 위치한 MN-CSE를 포함하는 논리적 기기로서 일반적으로 여러 센서나 액추에이터들이 연결되는 게이트웨이가 이에 해당한다고 보면 된다. MN은 한 개의 CSE로 구성이 되거나, 하나의 CSE에 1개 이상의 AE 를 포함하는 형태로 구성되어질 수 있다. MN은 Mcc 참조 포인트를 통해서 적어도 하나의 IN 또는 MN과 연동되며 Mcc 참조 포인트를 통해서 ASN과 연동 및 Mca 참조 포인트를 통한 ADN과 연동될 수 있다.
- » **Application Service Node (ASN)** : 필드 도메인에 위치한 ASN-CSE와 ASN-AE를 포함하고 있는 논리적 기기이다. 해당 ASN은 한 개의 CSE와 1개 이상의 AE를 포함하는 형태로 구성되며 논리적 ASN에 매칭되는 물리적 기기로는 M2M 디바이스를 예로 들 수 있다. ASN 은 Mcc 참조 포인트를 통해서 한 개의 MN 또는 한 개의 IN에 연동된다.
- » **Application Dedicated Node (ADN)** : 필드 도메인에 위치한 ADN-AE를 포함하고 CSE를 포함하지 않는 논리적 기기이다. 즉 해당 ADN는 CSE가 없고, 1개 이상의 AE를 포함한다. 논리적 ADN에 매칭되는 물리적 기기로는 센서 및 액추에이터와 같은 자원제약적인 M2M 디바이스를 예로 들 수 있다. ADN은 Mca 참조 포인트를 통해서 MN 또는 IN과 연동되는 구조를 갖는다.

1.2.4 oneM2M 공통 서비스 기능 작업

앞서 설명한 바와 같이 oneM2M 공통 서비스 플랫폼 개발과 관련하여 아키텍처 표준 문서에서 다루고 있는 부분은 Common Service Entity (CSE)에 대한 기능들을 표준화하는 것이다. 그리고 해당 기능들은 oneM2M이 지향하고 있는 리소스 기반 아키텍처 구조를 기반으로 각각의 리소스 타입으로 해당 공통 서비스 기능들이 표현되고 있다. oneM2M 표준화 작업 진행방식은 전체적으로 유스케이스 및 요구사항을 다루고, 이를 통해 필요한 기능들을 도출해 내어 시스템 아키텍처를 디자인하고, 상세 프로토콜을 개발하는 3GPP 표준화 방식에서 취한 Stage 기반의 접근 방식으로 표준 개발을 진행하였다.

이러한 방식을 바탕으로 Stage 1에서는 공통 플랫폼이 지원해야 하는 다양한 서비스 어플리케이션에 대한 유스케이스를 여러 산업군으로부터 추출하여 취합하고, 이들 유스케이스들로 부터 공통 기능에서 제공해야 하는 요구사항들을 도출하는 작업들이 진행되었다. Stage 2에서는

Stage1에서 도출된 요구사항을 만족할 수 있는 기능 요소들을 정의하고, 이를 통해 사물인터넷 서비스 계층 아키텍처를 표준화 하였으며, 여러 공통의 기능을 표현할 수 있는 리소스 타입 및 상위 수준의 메시지 흐름을 정의하는 작업들이 진행되었다. Stage 3에서는 아키텍처를 고려한 RESTful 스타일의 서비스 계층 프로토콜 디자인 및 서비스 계층과 하위 계층 간의 프로토콜 바인딩을 정의하였다. 그리고 보안 및 디바이스 관리, 추상화, 시맨틱스 작업은 Stage 1,2,3 전반에 걸쳐 표준화가 진행되었다.

1.2.5 oneM2M 공통 서비스 기능

앞장에서 살펴본 oneM2M의 표준 작업 결과 Stage 2에서 oneM2M 공통 서비스 기능이 도출되었으며, 해당 공통 서비스 기능을 oneM2M 리소스 정보 모델 기반 아키텍처에 따라 리소스 타입으로서 표현, 개발하였다. oneM2M 공통 서비스 기능은 총 12개로 정의되며, 해당 기능에 대한 이해와 이에 대한 리소스 타입에 대한 이해가 oneM2M 아키텍처 표준 문서의 핵심부분이다. 표 <1-2>에서는 oneM2M CSE에 위치한 12개의 공통 서비스 기능(Common Service Function)을 테이블화 한 것이며 각각의 기능은 리소스 타입으로 표현되고 Mcc, Mca, Mcn 참조 포인트를 통해서 서비스가 제공된다.

표 <1-3> oneM2M 플랫폼의 공통 서비스 기능

Common Service Function (CSF)	기능
Registration (REG)	REG CSF는 AE와 CSE 또는 CSE와 CSE간의 등록을 담당하며, 이러한 등록 관계를 통해서 oneM2M 엔티티간의 접속 및 접근이 가능하며 oneM2M 엔티티간의 데이터 전달을 통한 oneM2M 서비스 구성이 가능해진다.
Discovery (DIS)	DIS CSF는 기본적으로 oneM2M 리소스 및 어트리뷰트에 담긴 서비스 정보에 대한 검색 기능을 제공한다. 기본적으로 텍스트 기반의 검색이 제공되어지며, 그 이외에도 특징 기반의 검색 등이 제공되어질 수 있다.
Security (SEC)	SEC CSF는 oneM2M 공통 서비스 플랫폼의 공통 기능으로서 접근권한, 키 관리와 같은 보안 메커니즘을 제공하는 기능을 담당한다.
Group Management (GMG)	GMG CSF는 리소스들을 그룹으로 관리할 수 있도록 하는 기능을 담당한다. 이를 통해, 비슷한 역할을 하거나, 공동으로 관리되어야 하는 리소스들을 그룹으로 관리할 수 있다.
Data Management & Repository (DMR)	DMR CSF는 기본적으로 데이터 저장소의 기능을 제공하는 것이다. 또한 데이터의 타입, 시맨틱 정보, 시간, 위치와 관련한 데이터의 분류 및 데이터 포맷의 변경 및 데이터 처리 기능을 제공한다.

Common Service Function (CSF)	기능
Subscription & Notification (SUB)	SUB CSF는 리소스에 대한 구독정보를 관리하고 리소스 및 어트리뷰트에 대한 업데이트 발생, Child 리소스의 생성, 삭제 등 해당 리소스에 대한 트래킹 및 관련된 변화 정보를 통지하는 역할을 담당한다.
Device Management (DMG)	DMG CSF는 디바이스 관리 기능을 제공하는 공통 서비스 기능이다. 디바이스 관리라고 함은 디바이스에 설치된 펌웨어에 대한 관리, 디바이스 하드웨어 리소스 관리, 디바이스 동작 설정 관리, 진단을 포함한다.
Application & Service Management (ASM)	ASM CSF는 ADN, ASN, MN, IN에 위치한 AE와 CSE 소프트웨어에 대한 관리 기능을 담당한다. 해당 기능은 AE와 CSE의 재설치, 업데이트, 고장 탐지, 설정에 관한 기능을 제공한다.
Communication Management & Delivery Handling (CMDH)	CSE간, AE와 CSE간, 데이터 전달 시에 NSE를 통한 데이터 전달 서비스를 제공하는 기능을 담당한다. CMDH CSF는 메시지 전달을 위해서 언제 보낼건지, 어떤 네트워크 연결을 활용하여 보낼지를 결정한다.
Network Service Exposure, Service Execution & Triggering (NSSE)	NSSE CSF는 기저 네트워크와 관련된 통신을 관리하며 Mcn 참조 포인트를 통한 네트워크 접근 서비스를 제공한다.
Location (LOC)	LOC CSF는 AE의 위치 정보 요청에 대해서 ASN, MN과 관련된 위치정보를 획득하기 위한 방법을 제공하는 CSF이다.
Service Charging & Accounting (SCA)	SCA CSF는 oneM2M 공통 서비스 플랫폼을 통해서 제공되는 서비스에 대한 과금 체계 및 방법에 대한 기능을 제공한다.

제2장 유스케이스

본 장에서는 Release 1.0에 포함된 유스케이스를 소개하고자 한다. 2.1에서는 oneM2M 유스케이스들을 간단하게 소개하고 2.2에서는 유스케이스들 중 가장 간단한 유스케이스인 원격 조명 제어 유스케이스를 시나리오와 함께 설명한다.

2.1 oneM2M 유스케이스 소개

본 절에서는 oneM2M으로부터 제공되는 서비스의 관점에서의 유스케이스를 간단히 설명한다. oneM2M의 유스케이스는 에너지, Enterprise, 헬스케어, 공공서비스, 주거형 서비스, 차량 서비스, 기타 서비스 등 총 7개의 산업분야에서 33개의 유스케이스가 다양한 산업군으로 부터 제안되어 표준 문서에 포함되었다. 이들 중 본 해설서에서 다루게 될 원격 조명 제어 유스케이스가 포함되어지는, 주거형 서비스 분야에서의 유스케이스를 살펴보면 다음과 같다. [2], [4]

2.1.1 홈 에너지 관리 서비스

홈 에너지 관리 서비스 유스케이스는 집에서 사용되어지는 각종 가전 기기들의 에너지 소비를 관리하기 위한 것으로, 사용자들은 매일 가정에서 사용하는 에너지량에 대해 알 수 있고, 원격 제어 기능을 통해 다양한 가전기기를 제어할 수 있다. 특히, 집에서의 에너지 정보 수집과 이를 IoT 서버 관리 시스템에 전달하는 EGW (Energy Gateway)에 대해서도 다루고 있다. 본 유스케이스에서 도출된 공통의 요구사항은 다음과 같다.

- » 데이터 수집 및 보고(reporting)기능
- » IoT/M2M 기기 원격 제어
- » 정보 수집 및 다수의 어플리케이션으로의 정보 전송
- » 데이터 저장 및 공유
- » Privacy, 보안, 인증
- » 데이터 안전성
- » IoT/M2M 기기 및 게이트웨이에 대한 제어 및 관리
- » 서비스 가입 기능
- » 트래픽 흐름제어

2.1.2 HEHS(Home Energy Management System)

가정의 에너지 관리를 제공하는 Home Energy Management System (HEHS)에 기반하는 이 유스케이스는 여러 제조사의 다양한 가전들이 Local Area Network (LAN) 또는 Personal Area Network (PAN)을 통해 연결되고 게이트웨이를 통해 제어되는 시나리오를 다룬다. 게이트웨이 기기는 가정에 설치되어진 각종 가전기기의 상태 정보를 수집하고 이를 관리 서버에 전송하며, 새로운 가전기기를 지원할 수 있도록 자동 업그레이드 기능이 제공된다. 이로부터 도출된 요구사항은 다음과 같다.

- » 새로 설치되는 가전기기에 대한 자동 감지
- » 가전기기를 원격에서 자동으로 설정할 수 있는 pre-provisioning
- » 추상적 객체 모델을 이용하여 다양한 판매자의 가전제품 지원
- » 사용자가 직접 조작할 경우 HEMS의 제어를 override할 수 있는 기능

2.1.3 Plug-in Electrical Charging Vehicles and Power Feed in Home Scenario

본 유스케이스는 전기차량을 가정에서 충전하는 경우, 서로 다른 이해관계자들에 대한 상호작용 시나리오를 보여주고 있다. 본 유스케이스에서 언급되고 있는 이해관계자들로는 Electricity-Network service provider (Electricity-N/W-SP), Electric Vehicle Charging SP (EVC-SP), PEV service provider (PEV SP)등이 있다. PEV는 에너지 소비 매체 또는 전력 저장소 (power storage)로 사용될 수 있다. 이로부터 도출된 요구사항은 다음과 같다.

- » SW 업그레이드
- » PEV 상태 정보 수집
- » 과금 정보 수집
- » 수집된 데이터를 하나 이상의 어플리케이션에 전송
- » 데이터 저장 / 공유
- » 프라이버시

2.1.4 시맨틱 홈 제어

본 유스케이스는 서로 다른 두 개의 어플리케이션들 간에 리소스의 의미를 포함하고 있는 시맨틱 정보를 이용한 협업 (co-operation)에 대한 내용을 다루고 있으며, 이를 Building Management

System (BEMS)과 Home Energy Management System (HEMS)의 사례를 들어 설명하고 있다. 이로부터 도출된 요구사항은 다음과 같다.

- » 범용 시맨틱 데이터 모델 (예: Ontology) 지원
- » 시맨틱 정보에 기반한 IoT/M2M 리소스 검색 기능
- » 실제 생활 주변의 다양한 엔티티(예: 방, 창문 등) 검색
- » 추상화 디바이스와 실제 물리적 디바이스 간의 제어 명령 매핑

2.1.5 시맨틱 디바이스 플러그 앤 플레이

본 유스케이스는 새로운 기기가 등록될 경우, IoT/M2M 시스템의 시맨틱 정보를 이용하여 자체적으로 자신의 특징을 발견하고, 주변의 기기와 그 외 사물과의 관계를 설정하는 내용을 포함하고 있다. 예를 들어, 새로 설치된 기기들의 M2M 서비스 제공자에게 자신의 역할과 기능 그리고 시맨틱 정보를 제공하고, 새로 설치된 기기들의 시맨틱 정보와 기존 기기들의 시맨틱 정보를 비교하여 M2M 시스템 상에서 서로 간의 관계 (relationship)를 설정하고 이를 기반으로 새로운 서비스를 생성하거나 시맨틱 기반의 검색에 사용되어질 수 있게된다. 도출된 요구사항은 다음과 같다.

- » 버티컬 산업에 범용적으로 적용될 수 있는 시맨틱 데이터 모델
- » M2M 객체는 M2M 시스템에게 자신의 시맨틱 정보(description)을 제공
- » 사물 간의 시맨틱 관계 설정

다음 절에서 다룰 유스케이스는 위에서 설명한 주거형 서비스 분야 중에 모두 포함이 되어질 수 있으며, 주거형 사물인터넷 서비스의 가장 대표적이면서도 간단한 원격 조명 제어 유스케이스를 설명한다. 나머지 유스케이스들에 대해서는 2014년에 TTA에서 발행한 oneM2M 서비스 플랫폼 표준 해설서에 자세히 나와있으니 참고하기 바란다.

2.2 원격 조명 제어 유스케이스

본 해설서는 개발자들이 oneM2M 서비스 플랫폼으로부터 제공되는 기능을 사용하여 어플리케이션을 개발하는 것을 용이하게 하기 위하여 복잡한 유스케이스가 아닌, 가장 기본적이면서도

간단한 유스케이스를 제시한다. 이를 우리의 일상생활에서 찾아보면 다음과 같은 시나리오를 생각해볼 수 있다.

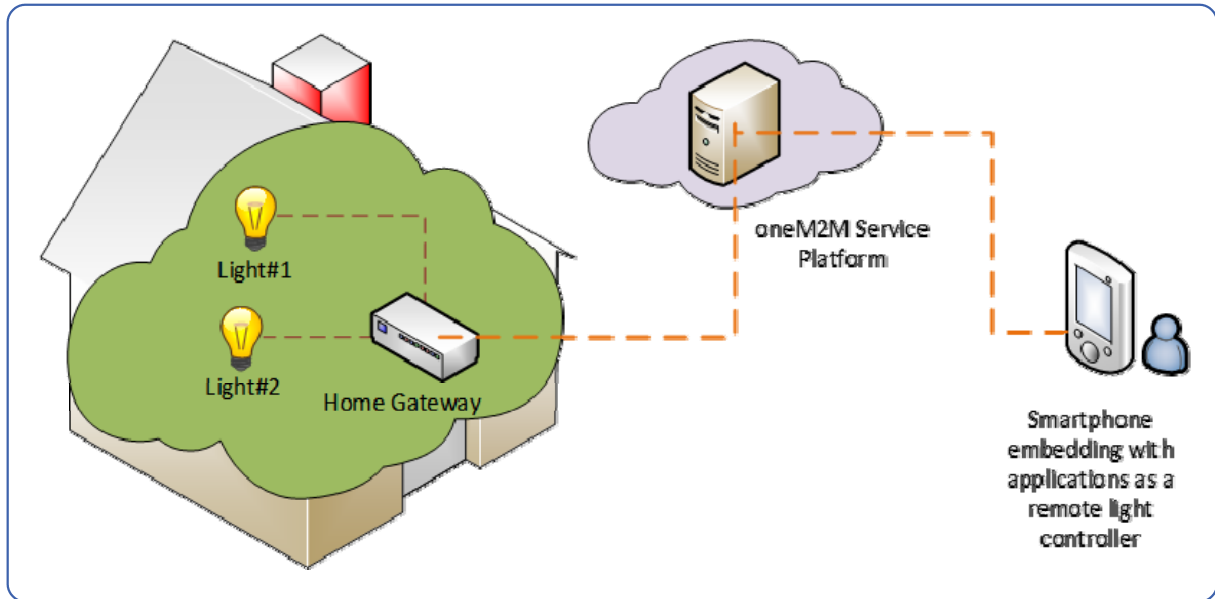


그림 <2-1> 원격 조명 제어 유스케이스의 개요

- » 원세종 군은 하루 일과를 마친 후 지친 몸을 이끌고 집으로 귀가했다.
- » 혼자 살고 있기 때문에 귀가했을 때 집 안에 있는 불은 모두 꺼져있는 상황이다.
- » oneM2M 사물인터넷 서비스 사용하지 않았던 한달 전까지만해도, 원세종군은 귀가후에 거실, 주방, 화장실 등의 불을 일일이 켜고, 취침 전에는 다시 모든 실내등들을 일일이 찾아다니며 꺼야만 했다.
- » 그러나, 한달전 한국 서비스에서 새롭게 선보인 oneM2M기반의 스마트홈 서비스에 가입하면서 부터는, 문앞에서부터 스마트 폰을 이용한 사물인터넷 원격 조명 제어 서비스를 이용함으로써, 집에 들어가 손가락 하나로 집 안의 모든 조명을 원격으로 제어할 수 있게 된다.
- » 이제 원세종 군이 집에 귀가할 때는 어두운 집이 아닌 밝은 집이 그를 반겨줄 수 있을 것이다.

본 유스케이스는 oneM2M 규격 서비스 플랫폼에 접근 가능한 어플리케이션이 내장된 스마트폰이나 스마트 탭을 통하여 원격 조명 제어를 가능하게 한다. 그림 <2-1>은 원격 조명 제어의 유스케이스 개요를 나타내며 주요 구성 요소는 다음과 같다.

- ① 본 유스케이스에 보이는 조명은 집에 배치되어 있고 홈 게이트웨이에 연결되어 있다. 조명은 무선 접근 인터페이스를 통해 oneM2M 플랫폼과 상호 작용할 수 있다.

- ② 홈 게이트웨이는 조명을 검색 하여 이를 자기 자신에 연결하고, 조명과 oneM2M 서비스 플랫폼 사이에서 조명의 최신 상태를 교환하고 저장할 수 있도록, oneM2M 서비스 플랫폼과 조명사이에 위치해 통신 기능을 제공하도록 구성되어 있다.
- ③ oneM2M 서비스 플랫폼은 집, 차량, 산업 같은 각각 다른 분야의 영역을 대상으로 차별화된 어플리케이션 서비스를 제공한다. 또한 등록, 검색, 데이터 관리, 그룹 관리, 신청/알림 등의 공통 서비스 기능을 지원한다.
- ④ 스마트폰 어플리케이션은 스마트폰에 내장되어 다음과 같은 기능을 통해 원격 조명 제어디바이스 역할을 한다.
 - 홈 게이트웨이에 연결되어져 있는 조명 기기들에 대한 검색
 - 조명의 상태를 변경시키는 명령어 전송. 즉, On/Off. 추가로 조명의 조도 등에 대한 변경도 가능하다
 - 조명 상태 검색

제3장 아키텍처 및 기능

본 장에서는 oneM2M 엔티티 역할에 의해 표시되는 구성 요소와 함께 구현된 유스케이스의 아키텍처 및 기능을 설명한다. 예를 들어, 물리적 장치는 ADN-AE로 모델링 될 수 있으며 oneM2M 서비스 시스템은 IN-CSE 등으로 모델링 될 수 있다. [3]

먼저 3.1 장에서는 본 해설서에서 다루고 있는 원격 조명 제어 시스템의 구조에 대해서 논하며, 이후 3.2에서 3.4장까지는 원격 조명 제어 시스템 시나리오에서 사용되어지는 Subscription 기능, 그룹관리 기능, 그리고 검색 기능을 순차적으로 설명한다.

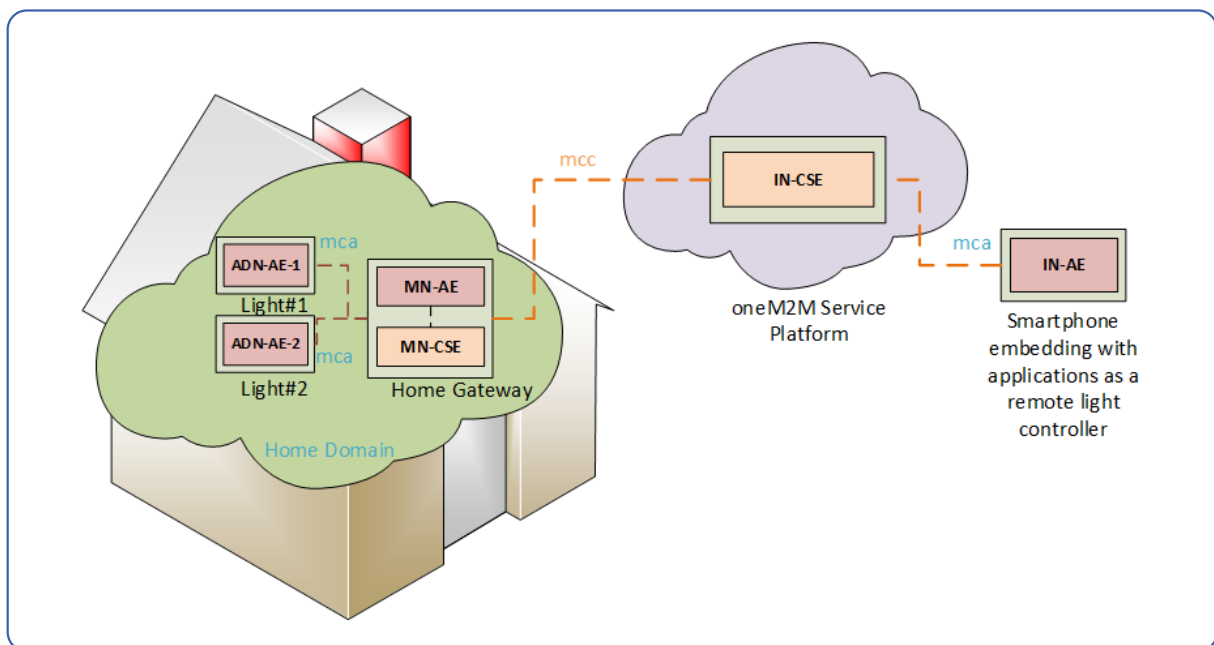


그림 <3-1> 원격 조명 제어 유스케이스의 oneM2M 기능 아키텍처

3.1 원격 조명 제어 시스템 구조

그림 <2-1>에서 보여주고 있는 원격 조명 제어 유스케이스는 그림 <3-1>에 보여지는 oneM2M 기능 아키텍처로 표현되어질 수 있다.

oneM2M 기능 아키텍처에서는 두 개의 entity의 역할이 정해져 있다. AE와 CSE가 바로 그것이다. 예를 들어 조명 같은 어플리케이션 전용 디바이스는 어플리케이션에 대한 로직을 포함하

고 있으므로, 보통 ADN-AE으로 구현되어질 수 있다. 스마트폰 어플리케이션은 스마트폰 디바이스에 내장되어 oneM2M 서비스 플랫폼과 직접 통신이 가능하도록 ADN-AE 역할을 맡는다. oneM2M 서비스 플랫폼은 각종 디바이스들을 리소스로 등록시키므로 IN-CSE의 역할을, 홈 게이트웨이는 여러 조명 기기들이 연결되어 서비스 플랫폼에 데이터에 대한 전달 등을 담당하므로 MN-CSE의 역할을 한다.

oneM2M 기능 구조에서 정의된 두 개의 참조 포인트인 mca와 mcc는 본 유스케이스에서 각각 AE와 CSE사이, 두 개의 CSE사이에서 사용되어진다. 그림 <3-1>에서 보는 것처럼, 조명 어플리케이션 (ADN-AE-1 또는 ADN-AE-2)과 홈 게이트웨이 MN-CSE사이에서 사용되는 참조 포인트가 mca, 홈 게이트웨이와 oneM2M 서비스 플랫폼 사이에서 사용되는 참조 포인트가 mcc로 해석되어질 수 있다.

요약하자면, 본 유스케이스에 사용되는 어플리케이션은 다음과 같이 분류된다.

- ① **ADN-AE1** : 조명 *Light#1*에 내장된 어플리케이션으로서 조명을 제어하고 mca 참조 포인트를 통하여 홈 게이트웨이 MN-CSE와 상호작용하는 기능을 가지고 있다.
- ② **ADN-AE2** : 조명 *Light#2*에 내장된 어플리케이션으로서 조명을 제어하고 mca 참조 포인트를 통하여 홈 게이트웨이 MN-CSE와 상호작용하는 기능을 가지고 있다.
- ③ **IN-AE** : 스마트폰 디바이스에 내장된 어플리케이션으로서 mcc 참조 포인트를 통해서 oneM2M 서비스 플랫폼인 IN-CSE와 직접 상호작용하여 조명 *Light#1*과 *Light#2*를 원격으로 제어하는 기능을 가지고 있다.
- ④ **MN-AE** : 홈 게이트웨이 MN-CSE에 내장되어 있으며 mca 참조 포인트를 통해서 MN-CSE와 상호작용하는 게이트웨이 어플리케이션이다. [6]

3.2 구독(Subscription) 기능

구독 (Subscription)은 어플리케이션간의 데이터 교환에 유용하게 쓰이는 기능이다. CSE내부에는 많은 데이터/서비스가 리소스의 형태로 저장되어 있고, 어플리케이션은 이러한 리소스를 통하여 데이터들을 읽고 쓰는 등의 사용을 할 수 있다. 어플리케이션이 특정 리소스에 관심이 있어 해당 리소스가 상태가 변경이 될 경우 자동으로 이러한 정보를 알림 받고자 할 경우 CSE가 제공하는 구독 기능을 이용할 수 있다. 이때 구독할 수 있는 정보는 구독 대상 리소스에 대한 다른 엔티티의 접근 내역, 리소스의 속성 변경 및 자식 리소스의 변경을 포함한다. 검색 기능과 비슷하게 특정 조건을 만족하는 변경 사항만 알림 받기 위해 알림 이벤트 조건 (notification-EventCriteria)을 설정할 수 있도록 정의되어져 있다.

구독 리소스의 속성은 크게 구독 속성 (예 : expirationCounter), 알림 속성 (예 : notificationEventCat), 알림 주소 (예 : notificationURI) 및 알림 이벤트 필터 (예 : notificationEventCriteria)로 나눌 수 있다. 다음 표 <3-1>은 구독 리소스의 주요 속성 정보에 대한 설명을 보여주고 있다.

표 <3-1> 구독(Subscription) 리소스의 주요 속성 정보

속성	설명
<i>eventNotificationCriteria</i>	Subscribed-to 리소스에 발생하는 Event 중 특정 조건을 만족하는 Notification을 수신하고자 할 때 설정
<i>expirationCounter</i>	해당 Counter 만큼의 Notification 전송 시 Subscription 리소스 삭제됨
<i>notificationURI</i>	Notification이 전송되는 URI
<i>batchNotify</i>	특정 시간/개수의 Notification을 통합하여 Batch Notification으로 수신하고자 할 때 설정
<i>rateLimit</i>	특정 시간 동안 특정 개수 이상의 Notification 수신을 제한하기 위해 설정.
<i>notificationEventCat</i>	Notification에 설정되는 QoS 카테고리 정보
<i>pendingNotification</i>	Reachability 및 Schedule에 의해 전송되지 못한 Notification을 재전송 하기 위한 Policy 정보
<i>notificationStoragePriority</i>	Notification이 바로 전송되지 않고 저장될 때 저장 우선 순위 정보
<i>notificationContentType</i>	Notification 메시지에 포함되는 데이터 구성. (E.g., Modified Attribute Only, Whole Resource)

3.3 그룹 관리 기능

그룹 기능은 그룹 멤버인 하나 이상의 리소스에 대한 간편한 접근 기능을 제공한다. 예를 들어 100개 리소스에 대한 100번의 획득 요청을 해당 그룹 리소스에 대한 단일 획득 요청으로 대신하여 100개 리소스의 정보를 한번의 트랜잭션을 통해 획득하는 것이다. 따라서 이러한 그룹 멤버 단일 접근 기능을 이용하기 위해서는 우선 관련있는 멤버 리소스들을 정의할 수 있는 그룹 리소스를 생성해야 한다.

그룹 리소스는 기본적으로 멤버 목록과 그룹 요청을 전송할 수 있는 가상 리소스로 구성된다. 그룹 기능을 이용하려는 요청자는 우선 그룹 멤버 리소스의 목록과 그룹 속성을 명시하여 그룹 리소스 생성을 요청한다. 성공적으로 그룹 리소스가 생성되면 가상 리소스 (fanOut)가 자동적으로 생성되며 해당 리소스에 생성/획득/갱신/삭제/알림 요청을 전송하면 그룹을 가진 CSE는 이 요청을 해당 그룹의 모든 멤버 리소스로 전파(fan-out)한다. 이때 전파 요청의 요청자 정보(fr)은 본래 요청자의 정보가 유지된다.

전파된 요청을 수신한 멤버들은 이를 본래 요청자가 전송한 요청으로 간주하고 접근 제어를 거쳐 해당 요청을 수행한다. 이후 요청을 수행 결과는 다시 그룹 소유 CSE로 전달되며, 그룹 소유 CSE는 멤버로부터 전달된 응답 메시지를 종합하여 본래 요청자에 전달한다.

그룹을 통해 그룹 멤버 리소스에 자식 리소스를 생성할 때 구독 리소스에 대해서는 추가적인 기능을 제공한다. 요청자는 그룹을 통해 모든 멤버에 동일한 구독 리소스를 생성할 수 있고 이를 통해 그룹 구독 생성자는 그룹 멤버의 모든 변경 사항을 동일한 알림 주소로 수신할 수 있다. 이때 각 멤버로부터의 개별 통지 메시지는 그룹 소유 CSE가 종합하여 전달해줄 수 있다. 다음의 표 <3-2>는 그룹 리소스가 가질 수 있는 자식 리소스와 여러 속성들에 대한 내용을 보여주고 있다.

표 <3-2> 그룹 리소스의 자식 리소스 및 속성 정보

자식 리소스	설명
<i><fanOutPoint></i>	모든 그룹 멤버에 Batch Request를 수행하기 위한 주소
속성	설명
<i>memberType</i>	멤버 리소스의 리소스 타입, 동일하지 않을 경우 "mixed"
<i>membersIDs</i>	멤버 리소스에 대한 링크
<i>currentNrOfMembers</i>	현재 멤버 수
<i>maxNrOfMembers</i>	최대 멤버 수
<i>membersAccessControlPolicyIDs</i>	fanOut 리소스를 통해 모든 멤버에 Request를 fan-out하기 위한 권한 정보
<i>memberTypeValidated</i>	모든 멤버의 Resource Type이 Validation되면 TRUE
<i>consistencyStrategy</i>	memberTypeValidated가 FALSE인 경우 그룹 관리 방법 (E.g., Validation 실패한 멤버 삭제)

3.4 검색 기능

검색 (Discovery) 기능 (혹은 발견 기능)은 특정 CSE가 가진 리소스에 대한 검색을 제공하는 기능이다. 즉, 검색의 대상이 엔티티가 아닌 리소스라는 특징을 가진다. oneM2M은 리소스 기반의 아키텍처를 채용하고 있고, 서비스를 리소스로 형상화하므로 특정 서비스를 발견하는 동작은 해당 리소스를 검색하는 동작으로 구성되어질 수 있다. 요청자는 특정 리소스를 검색하기 위한 조건을 명시하여 요청 메시지를 전송하고, 수신자는 조건에 맞는 리소스를 자신이 가진 리소스 중에 검사하여 해당 리소스의 목록을 반환한다.

다른 기능과는 다르게 검색 기능을 위한 별도의 리소스 타입은 존재하지 않는다. 즉 요청자는 리소스 발견을 위한 요청을 “검색” 리소스에 전송하지 않고 임의의 리소스에 전송할 수 있다. 이 리소스의 위치는 리소스 검색의 시작점이라고 할 수 있다. 앞서 살펴본 것과 같이 CSE는 리소스를 트리 (tree) 형식으로 저장하고 있다. 따라서 여기서 말하는 검색 시작점은 리소스 트리 상의 특정 리소스를 의미하고 리소스 검색의 범위는 시작점이 가리키는 리소스 및 해당 리소스에 대한 모든 하위 리소스를 포함한다.

아래 예시에서 확인할 수 있듯이 검색 기능은 획득 동작 (RETRIEVE operation)을 사용한다. 검색 기능에는 특정 조건을 만족하는 자원을 검색할 수 있도록 필터링 기능을 제공하는데 이는 요청 파라미터 중 fc (filter criteria)를 통해서 사용할 수 있다. fc는 다양한 필터 조건으로 구성되어 있다. 예시처럼 특정 자원 타입을 검색할 수도 있고 특정 시간 이후에 생성된 자원 만을 검색할 수도 있다. 그리고 검색 결과를 제한할 수도 있다.

필터 조건은 여러 개를 동시에 사용할 수 있도록 되어있어, 어플리케이션이 원하는 모든 조건들을 필터를 통해서 전달할 수 있다. 동일한 조건을 여러 개 사용하는 경우는 AND 또는 OR 논리 연산을 적용하여 결과를 반환한다. 예를 들어 “resourceType=AE & resourceType=container”인 경우 AE 자원 타입 그리고 컨테이너 자원 타입의 자원을 검색한다. 일반적인 자원 획득 기능과 발견 기능 모두 획득 동작을 사용하므로 이를 구분하기 위하여 발견 기능 사용시에는 filter criteria의 filterUsage를 Discovery로 설정한다. [2]

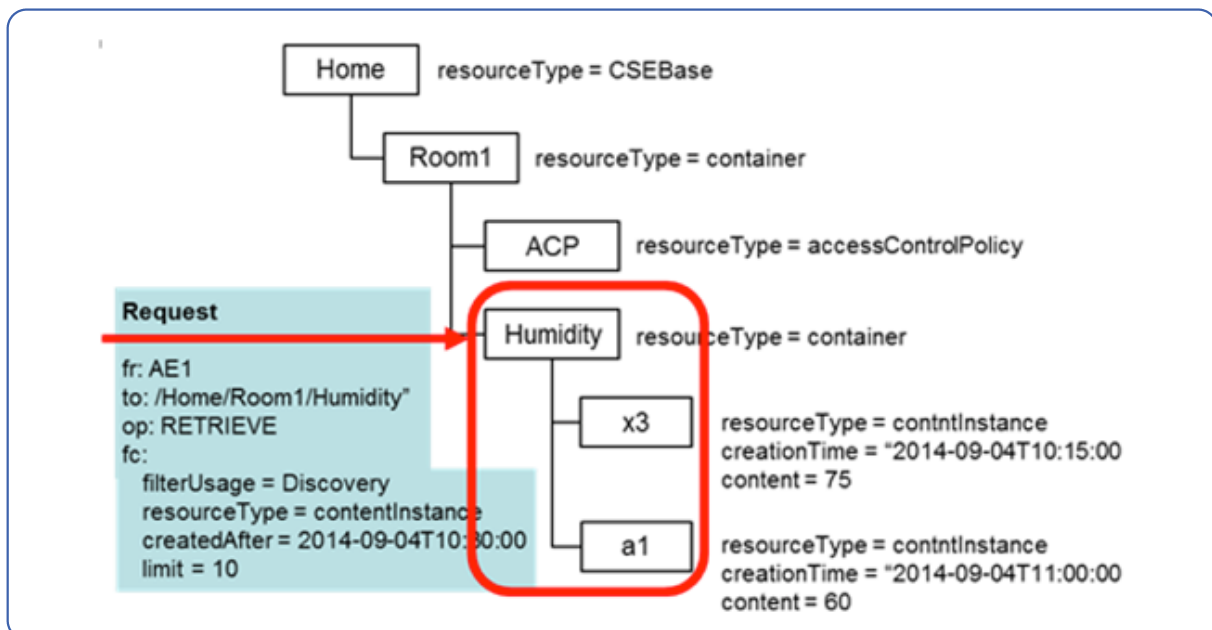


그림 <3-2> 발견 기능 예시

제4장 프로시저

이 장에서는 원격 조명 제어 유스케이스를 구현하기 위해 필요한 상세 프로시저에 대해서 설명하고자 한다. 4.1장에서는 원격 조명 제어 유스케이스가 실제로 oneM2M 아키텍처를 구성하는 엔티티들 간에 요구되어지는 프로시저를 소개하고, 그 뒤 4.2장에서는 각각의 프로시저에서 사용되어지는 메시지 흐름에 대해서 알아본다. 마지막으로 4.3장에서는 원격 제어 시나리오를 통해 단일 조명일 때와 다중 조명일 때의 프로시저를 분리하여 설명한다.

4.1 소개

본 장에서는 원격 조명 제어 유스케이스를 구현하기 위해 필요한 프로시저들을 도출하여 소개한다. 본 유스케이스를 지원하기 위해 oneM2M 표준에서 정의하고 있는 기능을 이용할 경우, 다음과 같은 프로시저들이 필요로 되어진다.

- ① **등록** : 해당 프로시저는 조명에 대한 등록, 게이트웨이의 어플리케이션 등록, 그리고 데이터 저장 리소스에 접근하는 서비스를 위한 접근 제어 정책 리소스 생성을 포함한다.
- ② **최초의 리소스 생성** : 본 프로시저는 그룹 리소스 생성, 특정 접근 제어 정책을 가진 컨테이너 리소스 생성, 초기 조명 상태를 가진 contentInstance 리소스 생성, 알람을 위한 가입 리소스 생성을 포함한다.
- ③ **검색 및 그룹 리소스 업데이트** : 특정 필터 기준을 가진 모든 컨테이너들은 게이트웨이 어플리케이션으로부터 발견될 수 있고 그룹 리소스의 멤버로 제공된다.
- ④ **발견 및 조명상태 검색** : 특정 필터 기준을 가진 모든 컨테이너들은 스마트폰 어플리케이션을 통해 리소스 ID를 사용하여 발견, 검색될 수 있다. oneM2M 서비스 플랫폼으로의 접근이 증가하고 content 정보 또한 검색될 수 있다.
- ⑤ **단일 조명 스위치 on/off** : 스마트폰 어플리케이션으로부터 검색되어 연결된 모든 조명은 스마트폰 어플리케이션을 통해 on/off 스위치가 가능하다.
- ⑥ **다중 조명 스위치 on/off** : 스마트폰 어플리케이션으로부터 검색되어 연결된 하나 이상의 조명들은 스마트폰 어플리케이션을 통해 on/off 스위치가 가능하다.

4.2 메시지 흐름

본 절에서는 유스케이스를 통해 소개되어진 각 절차에 관한 메시지 흐름을 설명한다. 4.2.1장에서는 등록하는 과정을, 4.2.2장에서는 초기 리소스 생성 과정, 그리고 4.2.3장에서 그룹 리소스의 발견 및 업데이트, 마지막으로 4.2.4장에서는 contentInstance의 발견 및 검색 순으로 알아보도록 한다.

4.2.1 등록

oneM2M 기반의 모든 디바이스들은 서비스를 통해서 사용되어지기 위해, 전원이 켜지게 되면, 해당 디바이스가 등록되어야 하는 게이트웨이 또는 서버를 찾아 등록을 하게 되며 이러한 과정을 통해 oneM2M 서비스를 이용할 수 있게 된다.

등록 단계에 관한 메시지 흐름은 그림 <4.2.1-1>에 보여지며 다음과 같은 순서로 진행되어진다.

- 1) oneM2M 서비스 플랫폼 (IN-CSE)에 게이트웨이 (MN-CSE) 등록
- 2) 게이트웨이 (MN-CSE)에 게이트웨이 어플리케이션 (MN-AE) 등록
- 3)-1 게이트웨이 (MN-CSE)에 조명 어플리케이션 (ADN-AE-1) 등록
- 3)-2 게이트웨이 (MN-CSE)에 조명 어플리케이션 (ADN-AE-2) 등록
- 4)-1 oneM2M 서비스 플랫폼 (IN-CSE)에 스마트폰 어플리케이션(IN-AE)을 등록
- 4)-2 다음 IN-CSE는 게이트웨이 (MN-CSE)에 스마트폰 어플리케이션 리소스 (IN-AE)를 알린다.
- 5) 게이트웨이 어플리케이션 (MN-AE)은 특정 필터 기준을 가진 게이트웨이 (MN-CSE)로부터 스마트폰 어플리케이션 (IN-AE)을 검색한다.
- 6) 게이트웨이 어플리케이션 (MN-AE)은 모든 조명 어플리케이션 (ADN-AE-1 및 ADN-AE-2)에 대한 접근 권한을 부여하는 스마트폰 어플리케이션(IN-AE)을 위해 접근 제어 정책 리소스를 생성한다.

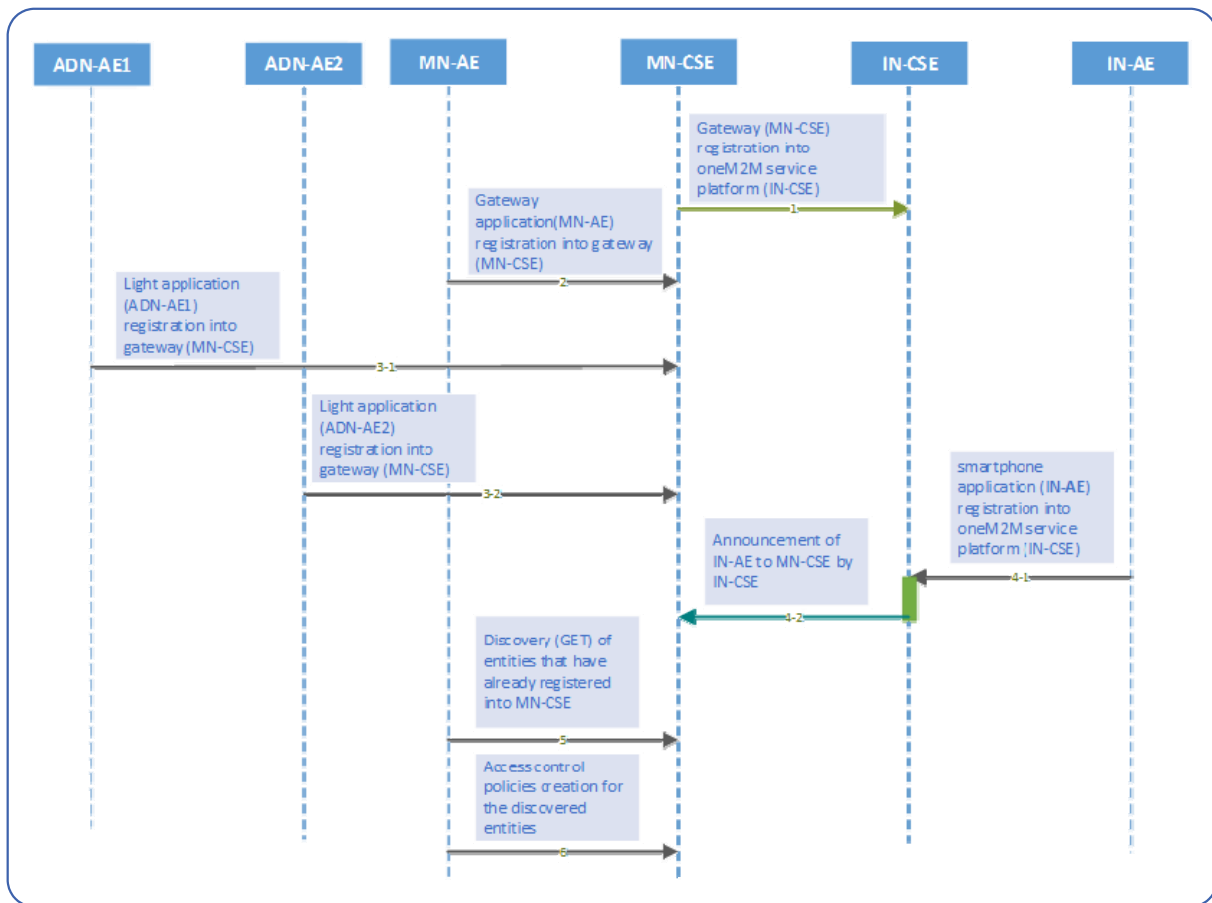


그림 <4.2.1-1> 등록 단계의 메시지 흐름

4.2.2 초기 리소스 생성

조명의 상태를 검색하고 업데이트 하기 위해서는 이러한 값들을 저장할 수 있는 초기 리소스에 대한 생성이 필요하다. 초기 리소스 생성에는 접근 제어 정책, 그룹 등의 내용들이 고려되어진다. 초기 리소스 생성 단계에 관한 메시지 흐름은 그림 <4.2.2-1>에 보여지며 순서는 다음과 같다.

- 1) 게이트웨이 어플리케이션 (MN-AE)은 게이트웨이 (MN-CSE)에 특정한 접근 제어 정책을 통해 두 개의 그룹을 생성한다. 그룹 멤버는 MN-AE로부터 초기화된 검색 과정을 통해 발견된 컨테이너 리소스의 목록으로부터 더해질 수 있다.

- 1)-1 하나는 *GROUP_FOR_LIGHT_STATES_UPDATE* 라는 이름의 조명 상태를 업데이트하기 위한 그룹이고,
- 1)-2 다른 하나는 *GROUP_FOR_LIGHT_STATES_RETRIEVAL* 라는 이름의 조명 상태를 검색하기 위한 그룹이다.

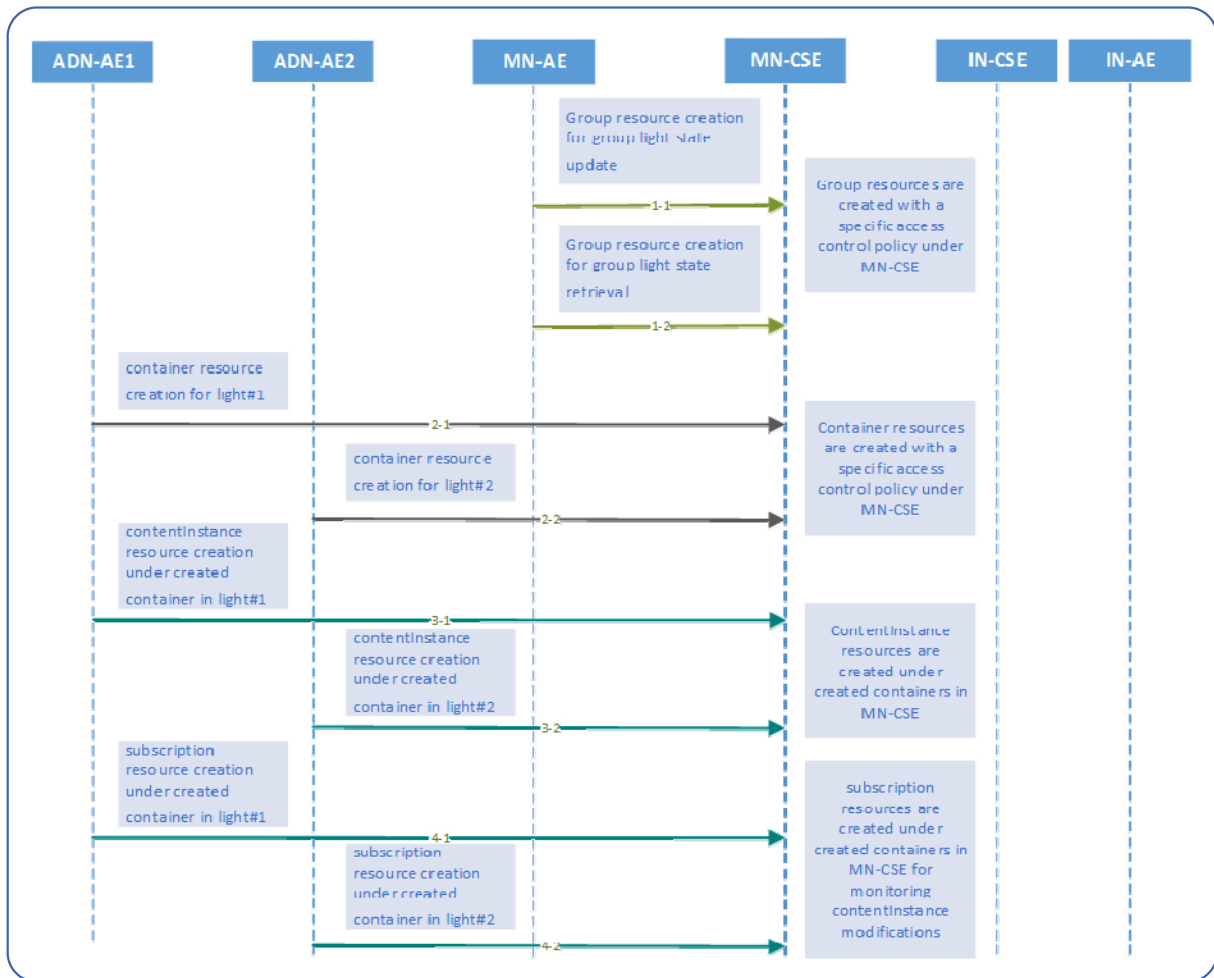


그림 <4.2.2-1> 초기 리소스 생성 단계의 메시지 흐름

2) 두 개의 컨테이너 리소스들이 각각 등록된 조명 어플리케이션에서 조명 상태를 저장하기 위해 게이트웨이 (MN-CSE)에 생성된다. 컨테이너들은 같은 접근 제어 정책을 사용하여 생성된다.

2-1) 조명 *light#1*에 대한 컨테이너 리소스가 MN-CSE에 생성된다.

2-2) 조명 *light#2*에 대한 컨테이너 리소스가 MN-CSE에 생성된다.

3) contentInstance 리소스는 제어되는 조명 상태를 나타내기 위해 생성된 각 컨테이너 아래에 위치하게 되며, 조명 어플리케이션에 의해 생성된다.

3-1) 조명 *light#1*에서 생성된 컨테이너 아래에 contentInstance 리소스 생성

3-2) 조명 *light#2*에서 생성된 컨테이너 아래에 contentInstance 리소스 생성

4) 새로운 contentInstance가 MN-AE이나 IN-AE에 의해 업데이트 될 때마다 가입자, 즉 조명 어플리케이션이 알림 받을 수 있도록 게이트웨이(MN-CSE)의 컨테이너 아래에서 등록 리소스가 생성된다.

4-1) 조명 *light#1*에서 생성된 컨테이너 아래 구독(Subscription) 리소스 생성

4-2) 조명 *light#2*에서 생성된 컨테이너 아래 구독 리소스 생성

4.2.3 그룹 리소스의 발견 및 업데이트

그룹 리소스의 발견 및 업데이트는 조명과 관련된 여러 데이터를 그룹 단위로 발견하고 업데이트 하기 위한 기능이다. 그림 <4.2.3-1>에 나와있는 그룹 리소스 상의 발견 및 업데이트에 관한 메시지 흐름은 다음과 같다.

- 1) 게이트웨이 어플리케이션 (MN-AE)은 *FilterCriteria* 파라미터에 주어진 특정 필터 기준과 관련된 GET 요청을 주기적으로 보낼 수 있다. *FilterCriteria* 파라미터는 게이트웨이의 MN-CSE에 저장된 컨테이너 리소스의 발견을 위한 요청 메시지에 들어있다. oneM2M 기능 구조에서 정의된 동작을 검색하기 위한 필터 기준 조건의 그룹 (*createdBefore*, *createdAfter*, *modifiedSince*, *unmodifiedSince*, *label*, *creator*, *expireAfter*, *resourceType* 등)은 검색 과정에서 사용될 수 있다.
- 2) MN-AE로 부터 수신되어진 요청 메시지를 게이트웨이 (MN-CSE)가 분석해본 결과, 만약 메시지에 필터 기준이 포함되어 있다면 그 기준에 따라 게이트웨이 어플리케이션 (MN-AE)으로 발견된 컨테이너 리소스의 URI를 응답한다.
- 3) 게이트웨이 어플리케이션 (MN-AE)은 최신 또는 예전의 *contentInstance* 리소스 검색과 *contentInstance*의 생성을 위한 발견된 컨테이너들이 제공하는 URI의 그룹 멤버 목록을 업데이트하기 위한 POST 요청을 보낸다. 발견된 멤버 URI는 앞에서 생성된 그룹 리소스 *GROUP_FOR_LIGHT_STATES_UPDATE*와 *GROUP_FOR_LIGHT_STATES_RETRIEVAL*에 들어있다.

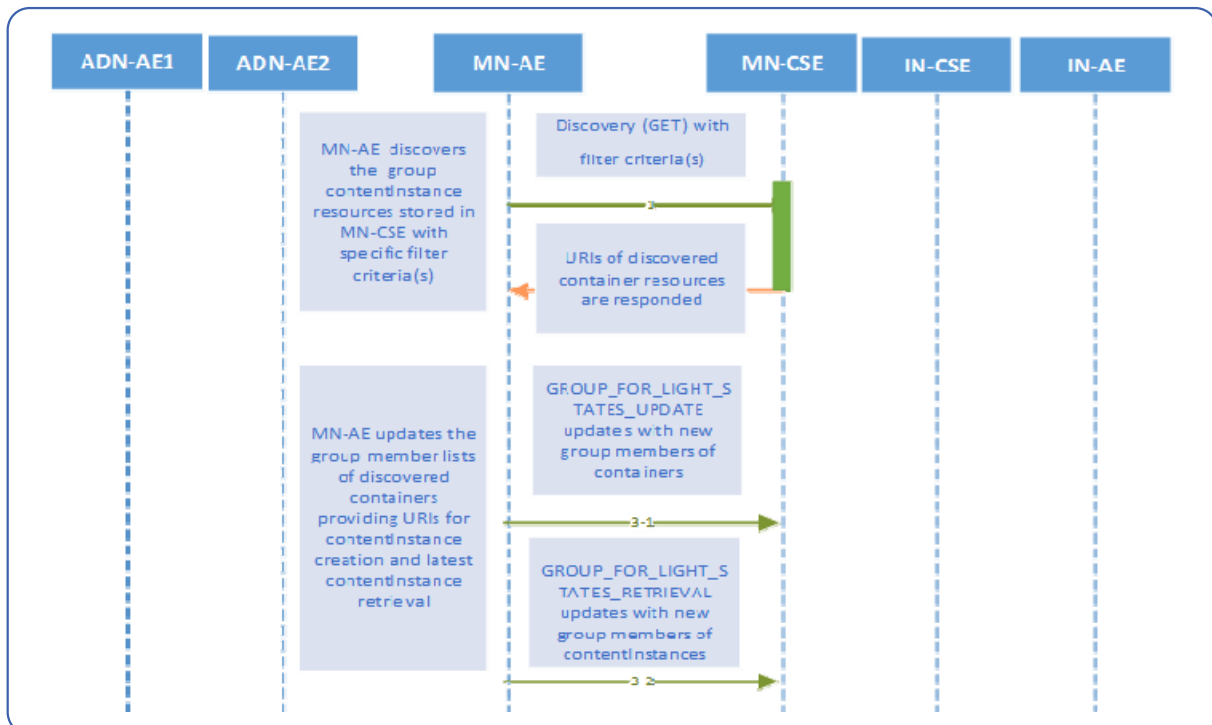


그림 <4.2.3-1> 발견 및 그룹 조명 상태 업데이트 상에서의 메시지 흐름

4.2.4 contentInstance 리소스의 발견 및 검색

스마트폰 어플리케이션으로부터 최신 업데이트된 contentInstance를 검색하기 위하여 content-Instance 리소스의 발견 및 검색 단계가 필요하다.

그림 <4.2.4-1>과 <4.2.4-2>에서 나타내는 contentInstance 리소스 상의 발견 및 검색에 관한 메시지 흐름의 순서는 다음과 같다.

- 1-1,1-2) 스마트폰 어플리케이션 (IN-AE)이 *FilterCriteria*에서 주어지는 특정 필터 기준과 관련된 GET 요청을 보낸다. 여기서 *FilterCriteria*는 MN-CSE에 저장된 컨테이너 리소스의 발견을 위한 요청 메시지에 포함된다. 또한 IN-AE은 MN-CSE에 저장된 그룹 리소스의 발견을 위해 MN-CSE로 GET 요청을 보낼 수 있다.
- 2-1,2-2) 게이트웨이 (MN-CSE)는 만약 ADN-AE-1과 ADN-AE-2에서 요청이 있다면 요청된 컨테이너 리소스의 URI를 스마트폰 어플리케이션 (IN-AE)으로 응답한다. IN-AE이 모든 그룹 멤버 리소스에 발견 요청을 보내는 경우에, MN-CSE은 IN-AE과 컨테이너의 발견된 그룹 리소스의 URI를 함께 응답한다.
- 3-1-1~3-2-2) 각 컨테이너 ADN-AE-1과 ADN-AE-2의 URI에 대해서, IN-AE은 단일 조명 어플리케이션으로부터 가장 마지막으로 업데이트된 contentInstance의 검색을 위한 GET 요청을 보낸다. contentInstance 리소스 그룹 멤버의 검색의 경우에는, IN-AE이 content-Instances의 그룹을 검색하기 위한 GET 요청을 보낸다.
- 4-1-1~4-2-2) 스마트폰 어플리케이션(IN-AE)은 요청했던 컨테이너의 조명 상태로 응답한다.

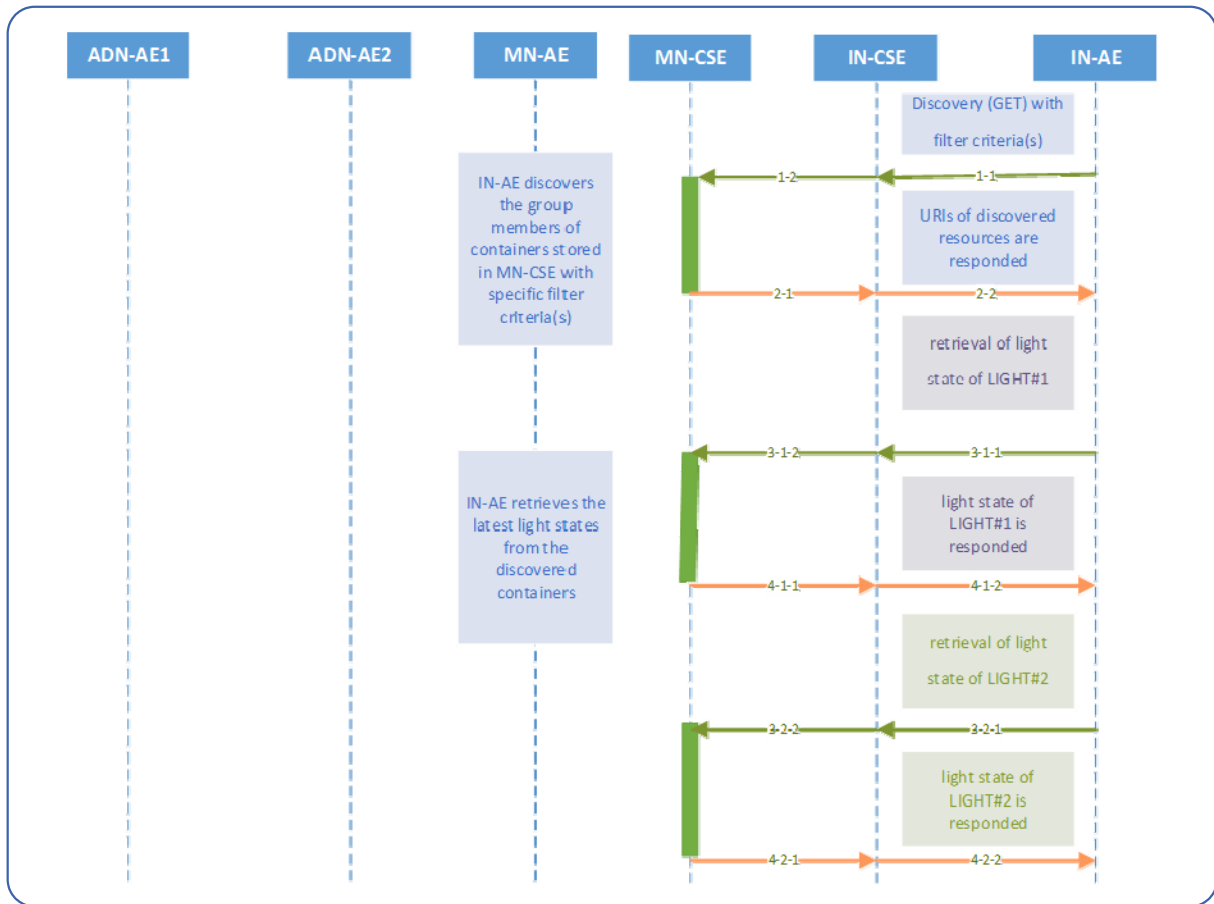


그림 <4.2.4-1> 발견 및 단일 조명 검색 상의 메시지 흐름

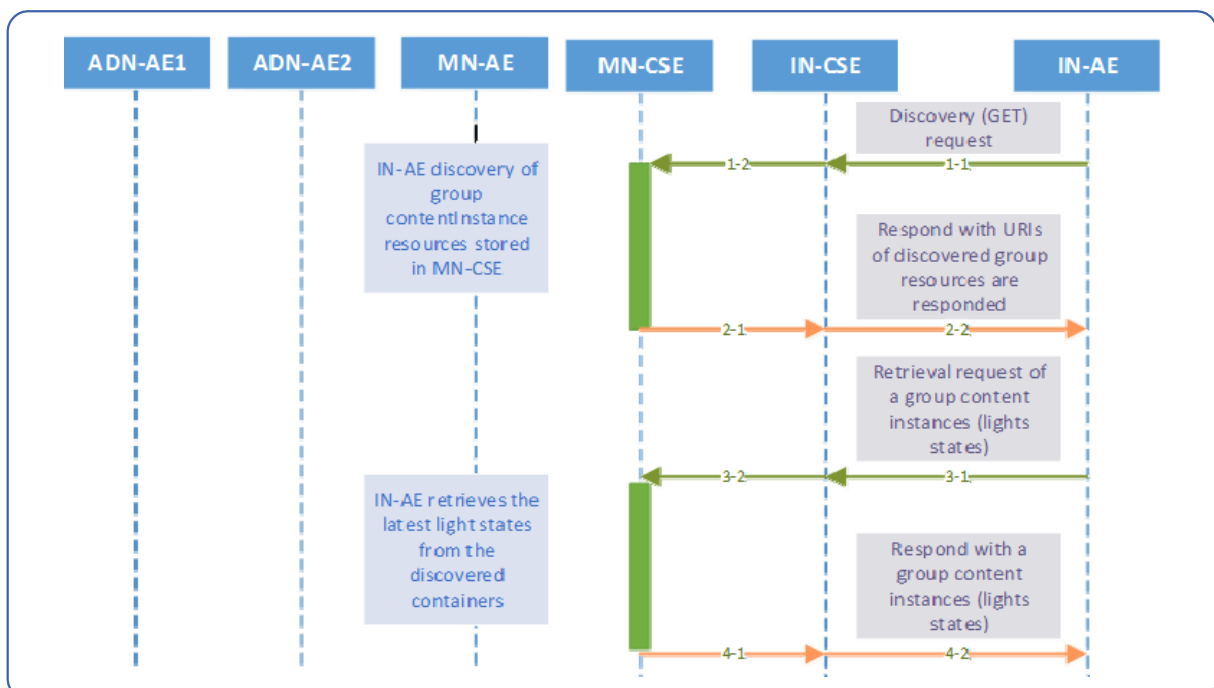


그림 <4.2.4-2> 발견 및 조명 그룹 검색 상의 메시지 흐름

4.3 원격 제어 시나리오

스마트폰과 조명이 서로 다른 네트워크에 연결되어 있는 경우도 실제 사물인터넷 운용 환경에서는 종종 발생한다. 예를 들어, 스마트폰의 경우 사업자 A를 통해 서비스를 받고 있으면서, 조명 사물인터넷 서비스는 사업자 B를 통하여 받는 운용 환경을 고려해 볼 수 있다. 이 경우 oneM2M 서비스 플랫폼에 접근 권한이 있는 어플리케이션이 내장된 스마트폰을 통하여 타 사업자 망에 연결되어있는 조명들을 원격으로 제어할 수 있다. 본 절에서 소개될 두 개의 시나리오는 다음과 같이 4.3.1 단일 조명 제어, 4.3.2 다중 조명 제어로 나뉜다.

4.3.1 단일 조명 제어

*Light#1*과 *Light#2*를 포함하는 어떤 단일 조명은 스마트폰 어플리케이션(IN-AE)을 통하여 사용자에게 원격으로 제어될 수 있다. 그림 <4.3.1-1>에서 보여지는 단일 조명 제어의 메시지 흐름 순서는 다음과 같다.

- 1-1) 사용자가 자신의 스마트폰에서 *light#1* 조명 상태를 업데이트 할 때, IN-AE은 IN-CSE에 *contentInstance* 생성을 하기 위한 요청을 보낸다.
- 1-2) IN-CSE는 IN-AE로 부터 수신받은 메시지를 해석하고, MN-CSE에 저장된 ADN-AE1의 대상 컨테이너 아래에 새 조명 상태를 나타내는 새로운 *content-Instance*를 생성한다.
- 2-1) *contentInstance*가 성공적으로 만들어졌다면, MN-CSE는 제어된 *light#1*이 실행된 최신 조명 상태 알림을 ADN-AE1에 보낸다.
- 1-3) 사용자가 자신의 스마트폰에서 *light#2* 조명 상태를 업데이트 할 때, IN-AE은 IN-CSE에 *contentInstance* 생성을 하기 위한 요청을 보낸다.
- 1-4) IN-CSE는 IN-AE로 부터 수신받은 메시지를 해석한뒤, MN-CSE에 저장된 ADN-AE2의 대상 컨테이너 아래에 새 조명 상태를 나타내는 새로운 *content-Instance*를 생성한다.
- 2-2) *contentInstance*가 성공적으로 만들어졌다면, MN-CSE는 제어된 *light#1*이 실행된 최신 조명 상태 알림을 ADN-AE2에 보낸다.

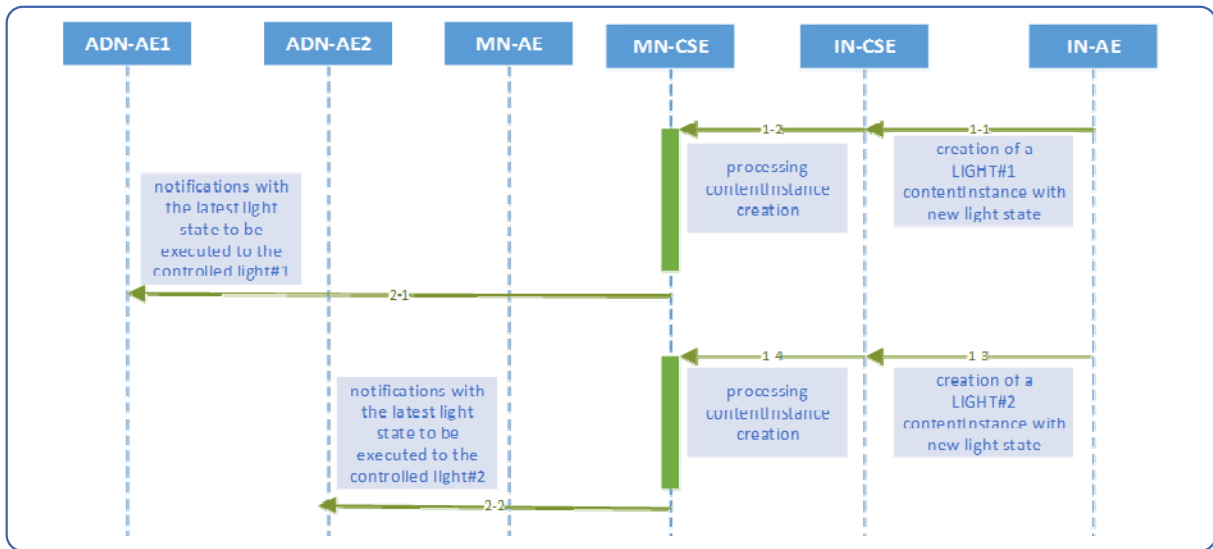


그림 <4.3.1-1> 단일 조명 원격 제어 메시지 흐름

4.3.2 다중 조명 제어

다중 조명은 가정에 하나 이상의 조명들이 산재해 있는 환경에서, 제어 과정을 용이하게 하기 위하여 원격으로 가정에 설치된 모든 조명들을 한번에 제어하는 것을 가능하게 해준다.

조명 원격 제어는 사용자가 스마트폰 어플리케이션 (IN-AE)을 통하여 oneM2M 서비스 플랫폼 (IN-CSE)으로 단일 조명 제어 명령을 다중 조명들이 등록되어 있는 그룹 리소스에 보냄으로써 여러 개의 조명을 원격으로 제어하는 것이 가능하다.

1-1,1-2) 사용자가 자신의 스마트폰에서 조명 상태의 그룹을 업데이트 시킬 때, IN-AE은 MN-CSE에 저장된 대상 컨테이너의 아래에서 새로운 조명 상태의 그룹을 나타내는 새로운 contentInstance 들의 그룹을 생성한다.

2-1-1) contentInstance가 성공적으로 생성되었다면, MN-CSE가 ADN-AE2에 알림을 보낸다.

2-1-2) contentInstance가 성공적으로 생성되었다면, MN-CSE가 ADN-AE1에 알림을 보낸다.

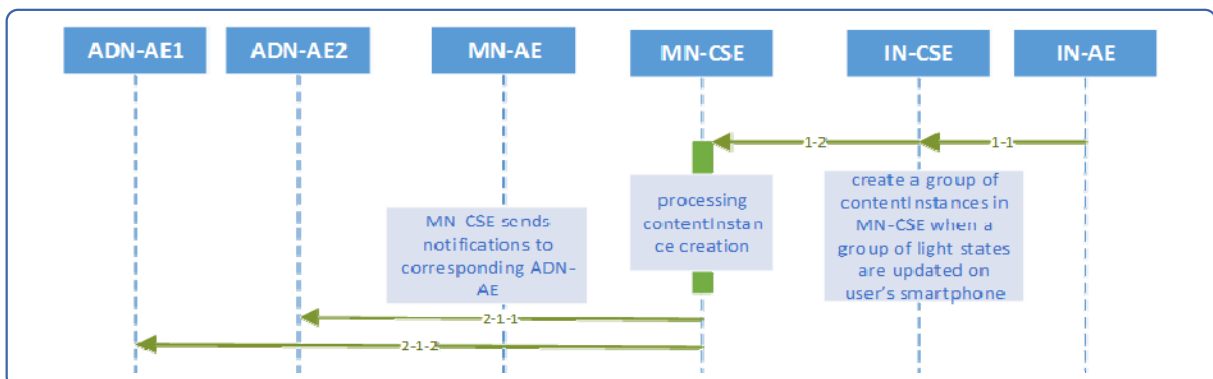


그림 <4.3.2-1> 다중 조명 원격 제어 메시지 흐름

제5장 구현

본 장에서는 앞에서 소개한 유스케이스의 구현에 대해서 설명한다. 5.1에서는 간단한 소개를, 5.2에서는 전제 조건, 5.2에서 엔티티에 대한 주소 지정, 5.3에서 조명 상태 데이터 모델링, 5.4에서 리소스 구조, 5.5에서 엔티티의 역할, 마지막으로 5.6에서 구현 절차 순으로 알아본다.

5.1 소개

유스케이스의 올바른 구현을 위한 전제조건을 만족하고 리소스 트리 등을 포함한 원격 조명 제어 유스케이스의 구현에 필요한 절차를 제시한다.

5.2 전제 조건

원격 조명 제어 유스케이스가 제대로 구현될 수 있도록 보장하기 위한 전제 조건은 다음과 같다.

- ① 모든 어플리케이션은 서버 사용이 가능하다.
- ② 디바이스와 어플리케이션 엔티티는 DNS 네트워크 서비스에 의해 해결된 호스트 네임으로 독립적인 자체 주소를 가진다.
- ③ 호스트 포트 번호 8080은 oneM2M 서비스를 위해 예약되어 있다.
- ④ 본 유스케이스에서 보안은 고려되지 않는다.
- ⑤ oneM2M 프리미티브의 HTTP 바인딩은 본 유스케이스에 사용된다.
- ⑥ oneM2M 프리미티브의 XML 직렬화는 본 유스케이스에 사용된다.
- ⑦ 본 유스케이스에서 옵션 헤더는 선택적으로 사용되는데 반해 모든 필수 HTTP 헤더는 HTTP 요청에서 제시된다.
- ⑧ 본 유스케이스에서 옵션 리소스 속성들이 선택적으로 사용되는 반면에 본 유스케이스에 존재하는 모든 필수 리소스 속성은 HTTP 요청에서 제시된다.
- ⑨ 본 유스케이스에 관련되어 있는 IN-CSE와 MN-CSE는 같은 oneM2M 서비스 프로바이더 도메인 내에 배포된다.

- ⑩ 본 유스케이스에 관련되어 있는 모든 AE들은 처음 CSE에 등록되고 AE의 식별자가 AE의 Registrar CSE에 의해 할당됨에 따라, 'C' 언어로 시작한다.
- ⑪ 본 유스케이스에서 생성된 모든 리소스는 *Hierarchical address* 의 oneM2M 리소스 식별자 형식으로 주소지정 될 수 있다.
- ⑫ 기본 접근 제어 정책은 이미 IN-CSE 아래 생성되었으며 이는 IN-CSE에 MN-CSE를 등록하는 데 사용된다.
- ⑬ 리소스와 속성들의 표현을 위한 약어들은 본 유스케이스에서 사용된다.
- ⑭ 모든 요청 발신자는 CSE에 있는 리소스에 접근하기 위해 *Blocking Requests* 을 보낸다.

5.3 엔티티에 대한 주소 지정

AE 및 CSE를 포함한 각 oneM2M 엔티티는 올바른 호스트 주소로 주소 지정 될 수 있다. 그것은 oneM2M 표준에 명시되어 있는 주소 지정 규칙에 따라 DNS 네트워크 서비스에 의해 IP 주소일 수 있거나 FQDN 주소가 IP 주소로 확인될 수 있다.

본 유스케이스에 나타나는 모든 엔티티는 다음의 URI들로 주소를 가질 수 있다.

>> IN-CSE : `/in-cse`

>> MN-CSE : `/mn-cse`

In-cse의 경우에는 IN-CSE의 SP-relative-CSE-ID이고 mn-cse는 MN-CSE의 SP-relative-CSE-ID이다.

5.4 조명 상태 데이터 모델링

contentInstance의 content에 저장된 조명 상태 ON/OFF는 contentInstance 리소스의 XML 표현 내부 문자열로 모델링 된다. 예를 들면 다음과 같다.

`<content> ON </content>`

혹은

`<content> OFF </content>`

5.5 리소스 구조

oneM2M 어플리케이션의 개발은 IN-CSE 및 MN-CSE 서비스 기능 계층의 리소스 트리 디자인을 포함한다. 리소스 트리는 4장 프로시저에서 제시된 절차에 따라 생성된 자식(하위) 리소스를 포함한다. 리소스 트리에 표시된 모든 자식 리소스들은 원격 조명 제어의 유스케이스를 구현하기 위해 필요하다.

5.5.1 IN-CSE의 리소스 구조

IN-CSE의 리소스 트리는 그림 <5.5.1-1>에 표시된 **in_cse**라는 이름의 CSEBase로 시작한다. 루트 CSEBase는 직접 자식 리소스로 구성되어 있으며, *mn_cse* 라는 이름의 *remoteCSE* 하나와 *in_ae*라는 이름의 AE 하나. mn-cse의 자식 리소스는 다음 절 5.5.2에서 소개된다.

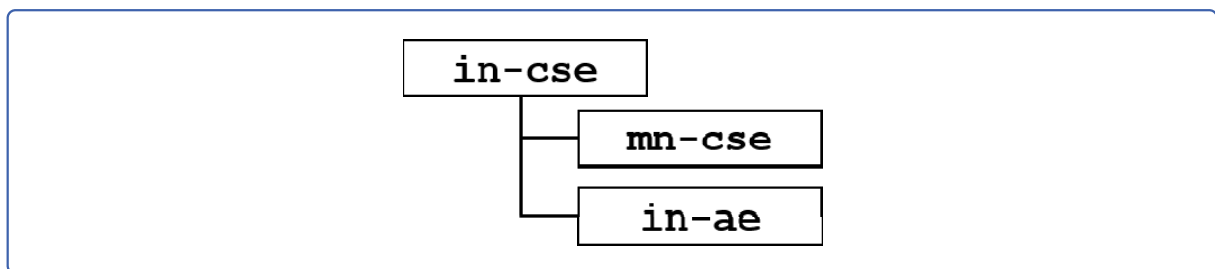


그림 <5.5.1-1> IN-CSE 리소스 구조

5.5.2 MN-CSE의 리소스 구조

MN-CSE의 리소스 트리는 그림 <5.5.2-1>에 표시된 *mn-cse*라는 이름의 *remoteCSE*로 시작한다.

MN-CSE의 리소스 트리는 다음과 같은 하위 리소스로 구성된다.

- >> *acp1*이라는 이름의 *accessControlPolicy*
- >> *adn-ae1*이라는 이름의 ADN-AE은 *light1*이라 이름 지어진 container의 하위 리소스와 여러 *contentInstance*를 포함한다.
- >> *adn-ae2*이라는 이름의 ADN-AE은 *light2*이라 이름 지어진 container의 하위 리소스와 여러 *contentInstance*를 포함한다.
- >> *mn-ae*이라는 이름의 MN-AE은 두 개의 그룹 리소스를 포함한다
- >> *subscription* 리소스는 각각 컨테이너 *light1*, *light2*의 하위 리소스로 표시되고,

» *group1* 과 *group2* 라는 이름의 두 그룹 리소스는 각각 MN-CSE에서 생성되는 컨테이너들의 멤버와 모든 컨테이너에서 최신 *contentInstance*로 생성된다(즉, 컨테이너 *light1*, *light2*). 접근 정책이 부여된 스마트폰 어플리케이션 사용자는 모든 조명 상태를 업데이트하기 위해 만들어진 새로운 *contentInstance*를 포함한 *fanOutPoint* 요청을 보낼 수 있다. 모든 최신 조명 상태는 그룹 *group2*로 GET 요청을 보냄으로써 검색될 수 있다.

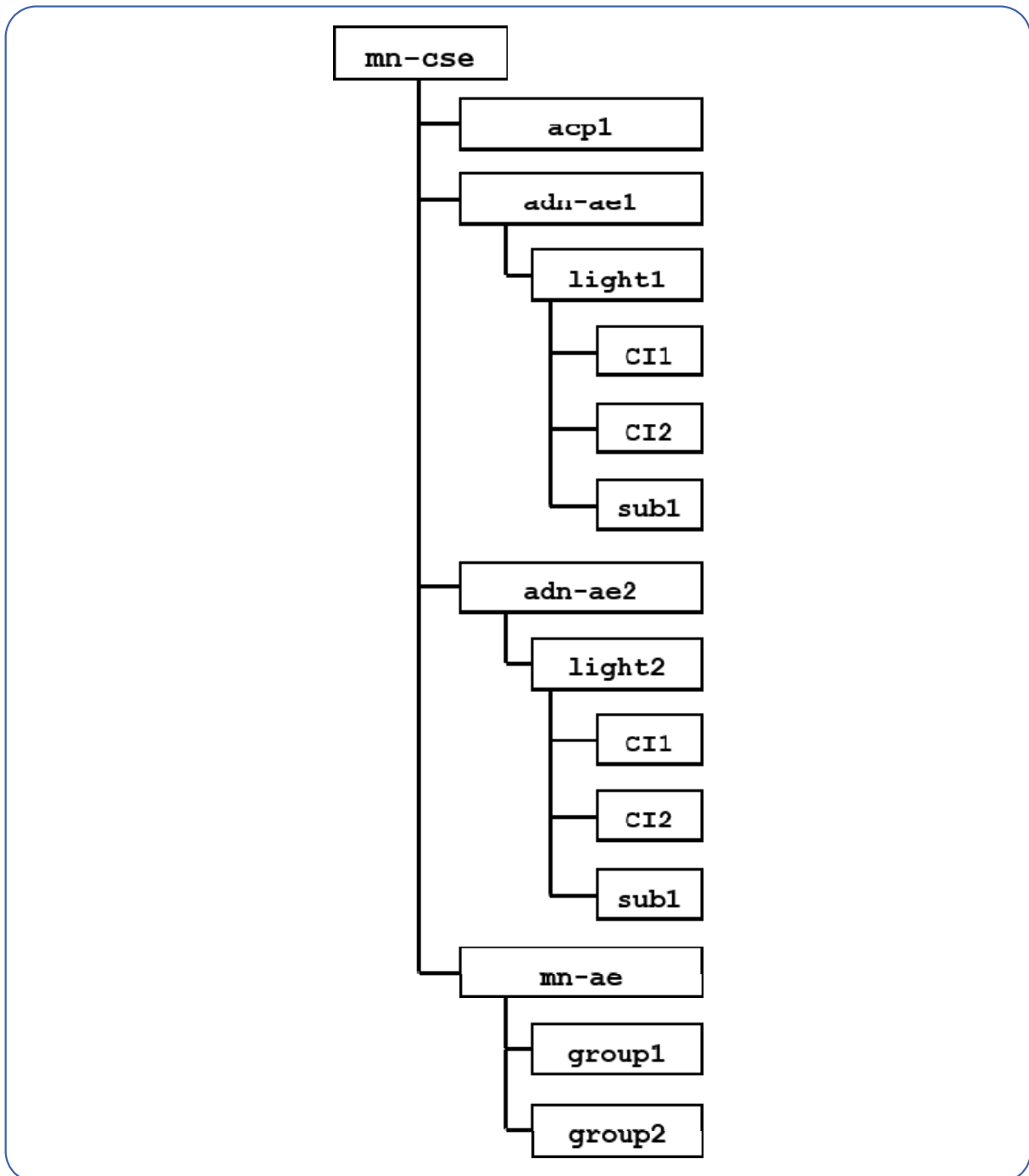


그림 <5.5.2-1> MN-CSE 리소스 구조

5.6 엔티티의 역할

본 절에서는 본 유스케이스에 포함된 모든 엔티티들의 역할을 제시한다.

5.6.1 oneM2M 서비스 플랫폼 (IN-CSE)

oneM2M 서비스 플랫폼은 루트 CSEBase 리소스 *in-cse* 의 표현과 함께 IN-CSE로 모델링 된다. CSEBase *in-cse* 은 다음에 대한 책임이 있다.

- » 각각 스마트폰 어플리케이션인 *in-ae* 와 홈 게이트웨이인 *mn-cse* 로부터 등록 요청을 처리하고,
- » remoteCSE인 *mn-cse* 에게 *in-ae* 을 알림.

5.6.2 홈 게이트웨이 어플리케이션 (MN-AE)

홈 게이트웨이 어플리케이션은 다음 과정에 의해 MN-AE로 모델링 된다.

- » 홈 게이트웨이 디바이스 초기화
- » MN-CSE에 홈 게이트웨이 어플리케이션 등록
- » MN-CSE에서 접근 제어 정책 리소스 *acp1* 생성
- » MN-CSE에 접근 제어 정책 *acp1*과 함께 그룹 리소스 생성
- » MN-CSE에 등록된 디바이스 어플리케이션과 MN-CSE에 위치한 그룹 멤버 리소스를 발견
- » 모든 조명 상태 표시

5.6.3 조명 어플리케이션(ADN-AE1 과 ADN-AE2)

두 개의 어플리케이션은 다음과 같은 과정에 의해 ADN-AE로 모델링 된다.

- » 조명 제어 장치 초기화
- » MN-CSE에 조명 장치 등록
- » 컨테이너 리소스 *light1*와 *light2*, 접근 제어 정책 *acp1*을 각각 생성
- » 컨테이너 *light1*과 *light2* 아래에 subscription 리소스 *sub1*과 *sub2*을 각각 생성
- » 컨테이너 *light1*과 *light2* 아래에 초기 조명 상태를 가진 각각의 *contentInstance* 리소스 생성

5.6.4 스마트폰 어플리케이션 (IN-AE)

스마트폰 어플리케이션은 oneM2M 서비스 플랫폼인 IN-CSE와 직접 통신할 수 있는 IN-AE로 다음과 같은 과정을 통해 모델링 된다.

- » 스마트폰 조명 제어 어플리케이션 초기화
- » IN-CSE에 스마트폰 어플리케이션을 등록
- » 컨테이너 light1과 light2 발견
- » 발견된 조명 상태 디스플레이
- » 스마트폰 어플리케이션 사용자로부터 조명 상태 변경 명령을 받아,
- » 단일 및 다중 조명에 대한 상태 변경 명령을 실행.

5.7 구현 절차

본 유스케이스에서 사용되는 구현 절차는 HTTP 바인딩과 참조 포인트 mca와 mcc를 설명하는 표준 API를 따르는 oneM2M 프리미티브의 XML 직렬화에 매핑된다. 또한, 리소스와 속성을 표현하는 short name이 구현 절차에 사용된다.

5.7.1 MN-CSE 등록

구현 절차는 다음과 같은 과정을 통해 IN-CSE에 MN-CSE를 등록하는 것으로 시작한다.

HTTP Request:

```
POST ~/in-cse/server?rcn=0 HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/mn-cse
Content-Type:application/vnd.onem2m-res+xml;ty=16
X-M2M-RI: incse/88643

<?xml version="1.0" encoding="UTF-8"?>
<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="home_gateway">
<csi>mn-cse</csi>
  <cb>mn.provider.com/mn-cse</cb>
  <rr>TRUE</rr>
<poa>http://mn.provider.com:8080</poa>
```

```
<cst>2</cst>
<acpi>/~/in-cse/acp-666957710</acpi>
</m2m:csr>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: incse/88643
Content-Location: /in-cse/csr-299409504
```

5.7.2 접근 제어 정책 생성

사용자가 접근 제어 정책 리소스를 생성할 때, 사용자는 *acor* 필드에 *accessControlOriginator*와 *acop* 필드에서 접근 제어 발신자에 대해 부여된 *operations*의 이름을 명확히 명시해야 한다.

우리가 *acop*의 값을 63으로 명시하여 지정된 발신자는 *CREATE*, *RETRIEVE*, *UPDATE*, *DELETE*, *DISCOVERY*, 그리고 *NOTIFY* operation을 수행하는 것이 허용됨을 나타낸다.

MN-CSE에서 접근 제어 정책 *acp1*의 생성은 다음과 같은 절차에 따라 구현된다.

HTTP Request:

```
POST /~/mn-cse/home_gateway?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
Content-Type:application/vnd.onem2m-res+xml;ty=1
X-M2M-RI: mncse/62948

<m2m:acp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway_acp">

  <pv>
    <acr>
      <acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse
/Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>
      <acop>63</acop>
    </acr>
  </pv>
<pvs>
  <acr>
```



```

<acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse/
Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>
<acop>51</acop>
</acr>
</pvs>
</m2m:acp>

```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/62948
Content-Location: /mn-cse/acp-805496226

```

접근 제어 정책 리소스는 해당하는 어플리케이션을 부여하는 데 사용될 수 있다. 접근 권한은 특정 리소스에 대한 접근 및 특정 작업을 수행하기 위해 MN-CSE로 등록된다. 어플리케이션 리스트는 필터 기준 조건을 이용한 발견 절차로 얻을 수 있다. 발견 절차에 대한 더 자세한 내용은, 5.6.8 절에서 볼 수 있다. 여기서 우리는 접근 제어 정책 리소스를 만들기 전에 발견된 어플리케이션의 리스트가 이미 MN-CSE에 등록됐다고 가정한다.

5.7.3 어플리케이션 엔티티 등록

① 조명 어플리케이션 ADN-AE1

MN-CSE에 ADN-AE1을 등록하는 것은 다음과 같은 절차에 따라 구현될 수 있다. ADN-AE1에 할당된 접근 제어 정책 식별자(비구조화된 CSE-relative resourceID)가 `/mn-cse/acp-805496226` 인 것에 주목한다.

HTTP Request:

```

POST /~/mn-cse/home_gateway?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/C
Content-Type:application/vnd.onem2m-res+xml;ty=2
X-M2M-RI: mncse/92345

<?xml version="1.0" encoding="UTF-8"?>

```

```

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light_ae1">
<api>A01.com.company.lightApp1</api>
  <rr>TRUE</rr>
  <poa>http://192.168.0.10:9090</poa>
<acpi>/~/mn-cse/acp-805496226</acpi>
</m2m:ae>

```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/92345
Content-Location: /mn-cse/ae-CAE340304071

```

② 조명 어플리케이션 ADN-AE2

MN-CSE에 ADN-AE2을 등록하는 것은 다음과 같은 절차에 따라 구현될 수 있다. ADN-AE2에 할당된 접근 제어 정책 식별자(비구조화된 CSE-relative resourceID)가 [/mn-cse/acp-805496226](#)인 것에 주목한다.

HTTP Request:

```

POST /~/mn-cse/home_gateway?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/C
Content-Type:application/vnd.onem2m-res+xml;ty=2
X-M2M-RI: mncse/18346

<?xml version="1.0" encoding="UTF-8"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light_ae2">
<api>A01.com.company.lightApp2</api>
  <rr>TRUE</rr>
  <poa>http://192.168.0.20:9090</poa>
<acpi>/~/mn-cse/acp-805496226</acpi>
</m2m:ae>

```

HTTP Response:

```

201 Created

```

```
X-M2M-RSC: 2001
X-M2M-RI: mncse/18346
Content-Location: /mn-cse/ae-CAE340304042
```

③ 홈 게이트웨이 어플리케이션 MN-AE

MN-CSE에 MN-AE를 등록하는 것은 다음과 같은 절차에 따라 구현될 수 있다. MN-AE에 할당된 접근 제어 정책 식별자(비구조화된 CSE-relative resourceID)가 `/mn-cse/acp-805496226`인 것에 주목한다.

HTTP Request:

```
POST ~/mn-cse/home_gateway?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/C
Content-Type:application/vnd.onem2m-res+xml;ty=2
X-M2M-RI: mncse/19347

<?xml version="1.0" encoding="UTF-8"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway_ae">
<api>A01.com.company.gatewayApp</api>
<rr>FALSE</rr>
<acpi>/~/mn-cse/acp-805496226</acpi>
</m2m:ae>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/19347
Content-Location: /mn-cse/ae-CAE340303271
```

④ 스마트폰 어플리케이션 IN-AE

IN-CSE에 IN-AE를 등록하는 것은 다음과 같은 절차에 따라 구현될 수 있다. MN-AE에 할당된 접근 제어 정책 식별자(비구조화된 CSE-relative resourceID)가 `/in-cse/acp-666957710`인 것에 주목한다.

HTTP Request:

```
POST ~/in-cse/server?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/C
Content-Type:application/vnd.onem2m-res+xml;ty=2
X-M2M-RI: incse/16346

<?xml version="1.0" encoding="UTF-8"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="smartphone_ae">
<api>A01.com.company.lightControlApp</api>
<rr>FALSE</rr>
<acpi>/~/in-cse/acp-666957710</acpi>
</m2m:ae>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: incse/16346
Content-Location: /in-cse/ae-CAE340304178
```

5.7.4 컨테이너 생성

① ADN-AE1의 컨테이너 생성

ADN-AE1에 대한 컨테이너 리소스의 생성은 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```
POST ~/mn-cse/home_gateway/light_ae1?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae1
Content-Type:application/vnd.onem2m-res+xml;ty=3
X-M2M-RI: mncse/13345

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">
</m2m:cnt>
```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/13345
Content-Location: /mn-cse/cnt-582759912

```

② ADN-AE2의 컨테이너 생성

ADN-AE2에 대한 컨테이너 리소스의 생성은 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```

POST ~/mn-cse/home_gateway/light_ae2?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae2
Content-Type:application/vnd.onem2m-res+xml;ty=3
X-M2M-RI: mncse/62345

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">
</m2m:cnt>

```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/62345
Content-Location: /mn-cse/cnt-582769893

```

5.7.5 ContentInstance 생성

① ADN-AE1의 contentInstance 생성

초기 content OFF를 가진 ADN-AE1의 컨테이너 아래에 contentInstance 리소스의 생성은 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```

POST ~/mn-cse/home_gateway/light_ae1/light?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae1
Content-Type:application/vnd.onem2m-res+xml;ty=4

```

X-M2M-RI: mncse/24345

```
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
<con>OFF</con>
<cnf>text/plain:0</cnf>
</m2m:cin>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/24345
Content-Location: /mn-cse/cin-394798749
Content-Type: application/vnd.onem2m-res+xml
```

② ADN-AE2의 contentInstance 생성

초기 content OFF를 가진 ADN-AE2의 컨테이너 아래에 contentInstance 리소스의 생성은 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```
POST ~/mn-cse/home_gateway/light_ae2/light?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae1
Content-Type:application/vnd.onem2m-res+xml;ty=4
X-M2M-RI: mncse/22345

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
<con>OFF</con>
<cnf>text/plain:0</cnf>
</m2m:cin>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/22345
Content-Location: /mn-cse/cin-256599578
```

5.7.6 그룹 생성

① 모든 조명 상태를 업데이트하기 위한 그룹 생성

MN-AE에서 그룹 리소스 *group1* 의 생성은 다음과 같은 절차에 따라 구현될 수 있다. 그룹 리소스는 두 개의 초기 멤버 id 0과 1로 생성된다.

HTTP Request:

```
POST /~/mn-cse/home_gateway/gateway_ae?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
Content-Type:application/vnd.onem2m-res+xml;ty=9
X-M2M-RI: mncse/76905

<?xml version="1.0" encoding="UTF-8"?>
<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="containers_grp">
    <mt>3</mt>

    <mid>/~/mn-cse/cnt-582759912 /~/mn-cse/cnt-582769893</mid>
<mnrm>10</mnrm>
</m2m:grp>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/76905
Content-Location: /mn-cse/grp-977978327
```

② 모든 최신 조명 상태의 검색을 위한 그룹 생성

MN-AE에서 그룹 리소스 *group2* 의 생성은 다음과 같은 절차에 따라 구현될 수 있다. 그룹 리소스는 두 개의 초기 멤버 id 0과 1로 생성된다.

HTTP Request:

```
POST /~/mn-cse/home_gateway/gateway_ae?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
Content-Type:application/vnd.onem2m-res+xml;ty=9
```

X-M2M-RI: mncse/12980

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstates_grp">
  <mt>4</mt>
  <mid>/~/mn-cse/cnt-582759912/1a /~/mn-cse/cnt-582769893/1a</mid>
  <mnrm>10</mnrm>
</m2m:grp>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/12980
Content-Location: /mn-cse/grp-677974227
```

5.7.7 구독(Subscription) 생성

① ADN-AE1의 contentInstance를 구독(Subscription)

사용자가 구독 리소스를 생성할 때, 파라미터 notification content type (줄여서 nct)은 구독된 리소스의 모든 속성들이 구독자에게 알려질 것을 나타내기 위해 값이 1로 설정된다.

ADN-AE1은 아래의 절차에 따라 ADN-AE1에서 컨테이너의 contentInstance에 변경 사항이 있을 때마다 ADN-AE1이 알림을 받을 수 있도록 ADN-AE1의 리소스 식별자로 설정된 알림 URI를 포함하여 구독 리소스를 생성한다.

HTTP Request:

```
POST /~/mn-cse/home_gateway/light_ae1/light?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae1
Content-Type:application/vnd.onem2m-res+xml;ty=23
X-M2M-RI: mncse/67891

<?xml version="1.0" encoding="UTF-8"?>
<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate_sub1">
  <enc>
<net>3</net>
```



```

</enc>
<nu>/~/mn-cse/ae-CAE340304071</nu>
<nct>1</nct>
</m2m:sub>

```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/67891
Content-Location: /mn-cse/sub-856593979

```

② ADN-AE2의 contentInstance를 구독(Subscription)

사용자가 구독 리소스를 생성할 때, 파라미터 *notification content type* (줄여서 *nct*)은 구독된 리소스의 모든 속성들이 구독자에게 알려질 것을 나타내기 위해 값이 1로 설정된다.

ADN-AE2은 아래의 절차에 따라 ADN-AE2에서 컨테이너의 contentInstance에 변경 사항이 있을 때마다 ADN-AE2이 알림을 받을 수 있도록 ADN-AE2의 리소스 식별자로 설정된 알림 URI를 포함하여 구독 리소스를 생성한다.

HTTP Request:

```

POST /~/mn-cse/home_gateway/light_ae2/light?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Clight_ae2
Content-Type:application/vnd.onem2m-res+xml;ty=23
X-M2M-RI: mncse/29387

<?xml version="1.0" encoding="UTF-8"?>
<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate_sub2">
  <enc>
    <net>3</net>
  </enc>
  <nu>/~/mn-cse/ae-CAE340304042</nu>
<nct>1</nct>
</m2m:sub>

```

HTTP Response:

```

201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/29387
Content-Location: /mn-cse/sub-856463728

```

5.7.8 발견(Discovery)

oneM2M 서비스 플랫폼에서 발견 기능은 하나 이상의 필터 기준에 따라 RETREIEVE operati-on 을 사용하여 구현된다. 리소스 발견을 위한 retrieve operation을 가능하게 하기 위해, 파라미터 *filterUsage* (줄여서 *fu*)는 쿼리 문자열로 RETRIEVE 요청에 제시되어야 한다. 또한, 파라미터 *resource type* (줄여서 *rty*)는 단일 조명 및 그룹 조명 멤버의 발견을 위한 필터 기준 조건으로 사용될 수 있다. 파라미터 *discovery result type* (줄여서 *drt*)은 URIList의 요소 형식이 구조화되지 않은 것을 나타내기 위해 2로 설정된다.

① MN-CSE에 등록된 단일 조명 발견

MN-CSE에 등록된 단일 조명 발견은 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```

GET /~/mn-cse/home_gateway?fu=1&rty=3&drt=2 HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/99882
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
X-M2M-RI: mncse/99882
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols">
    /~/mn-cse/cnt-582759912
    /~/mn-cse/cnt-582769893
</m2m:uril>

```

사용자가 사용하는 스마트폰 어플리케이션은 응답 메시지에서부터 MN-CSE에 등록된 컨테이너를 나타내는 URI의 목록을 검색할 수 있다. 예를 들면, /mn-cse/cnt-582759912 는 ADN-AE1에 생성된 컨테이너의 URI 이다.

② MN-CSE에 위치한 그룹의 발견

MN-CSE에 위치한 그룹의 발견은 *in-ae* 에 의해 다음과 같은 절차에 따라 구현될 수 있다.

HTTP Request:

```
GET ~/mn-cse/home_gateway?fu=1&rty=9&drt=2 HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/15001
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: mncse/15001
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols">
    ~/mn-cse/grp-977978327
    ~/mn-cse/grp-677974227
</m2m:uril>
```

사용자는 응답 메시지에서부터 MN-CSE에 위치한 그룹 리소스를 나타내는 URI의 목록을 검색할 수 있다. 예를 들면, /mn-cse/grp-977978327 그룹 group1의 URI 목록이 있다. 발견된 그룹 리소스의 검색된 URI는 그룹 멤버 업데이트 operation에 사용될 수 있다.

5.7.9 최신 contentInstance 검색

사용자는 가상 리소스 *latest* (줄여서 *la*)가 부착된 최신 contentInstance RETRIEVE 요청 보

내기를 통해서 최신 조명 상태를 검색할 수 있다. 또한 사용자는 가상 리소스 *fanOutPoint* (줄여서 *fopt*)가 부착된 그룹 RETRIEVE 요청 보내기를 통해 최신 조명 상태의 그룹을 검색할 수 있다.

① ADN-AE1의 최신 contentInstance 검색

ADN-AE1에서 컨테이너 *light1* 의 contentInstance는 다음과 같은 절차를 통해 검색될 수 있다.

HTTP Request:

```
GET /~/mn-cse/home_gateway/light_ae1/light/la HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/11223
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: mncse/11223
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin
  xmlns:m2m="http://www.onem2m.org/xml/protocols"
  rn="cin-394798749">
  <ty>4</ty>
  <ri>cin-394798749</ri>
  <pi>cnt-181049109</pi>
  <ct>20150925T045938</ct>
  <lt>20150925T045938</lt>
  <et>20151107T154802</et>
  <st>0</st>
  <cnf>text/plain:0</cnf>
  <cs>3</cs>
  <con>OFF</con>
</m2m:cin>
```

② ADN-AE2의 최신 contentInstance 검색

ADN-AE2에서 컨테이너 *light* 의 contentInstance는 다음과 같은 절차를 통해 검색될 수 있다.

HTTP Request:

```
GET /~/mn-cse/home_gateway/light_ae2/light/la HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/22336
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC:2000
X-M2M-RI: mncse/22336
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin
  xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-256599578">
  <ty>4</ty>
<ri>cin-256599578</ri>
  <pi>cnt-790965889</pi>
  <ct>20150925T050515</ct>
  <lt>20150925T050515</lt>
<et>20151107T154802</et>
<st>0</st>
<cnf>text/plain:0</cnf>
  <cs>3</cs>
  <con>OFF</con>
</m2m:cin>
```

③ 모든 조명 상태를 위한 최신 contentInstance의 그룹 검색

최신 contentInstance의 그룹은 다음과 같은 절차에 따라 *group* RETRIEVE 요청 보냄으로써 얻을 수 있다.

HTTP Request:

```
GET /~/mn-cse/home_gateway/gateway_ae/lightstates_grp/fopt HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/55667
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: mncse/55667
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:agr
  xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <m2m:rsp>
    <rsc>2000</rsc>
    <pc>
      <m2m:cin rn="cin-394798749">
        <ty>4</ty>
        <ri>cin-394798749</ri>
        <pi>cnt-181049109</pi>
        <ct>20150925T045938</ct>
        <lt>20150925T045938</lt>
      <et>20151107T154802</et>
      <st>0</st>
      <cnf>text/plain:0</cnf>
      <cs>3</cs>
      <con>OFF</con>
    </m2m:cin>
  </pc>
  <to>/~/in-cse/ae-CAE340304178</to>
  <fr>/~/mn-cse/cnt-582759912/la</fr>
</m2m:rsp>
```

```

<m2m:rsp>
  <rsc>2000</rsc>
  <pc>
    <m2m:cin rn="cin-256599578">
      <ty>4</ty>
      <ri>cin-256599578</ri>
      <pi>cnt-790965889</pi>
      <ct>20150925T050515</ct>
      <lt>20150925T050515</lt>
    <et>20151107T154802</et>
    <st>0</st>
  <cnf>text/plain:0</cnf>
  <cs>3</cs>
  <con>OFF</con>
</m2m:cin>
</pc>
<to>/~/in-cse/ae-CAE340304178</to>
<fr>/~/mn-cse/cnt-582769893/la</fr>
</m2m:rsp>
</m2m:agr>

```

5.7.10 조명 상태 변경

일단 스마트폰 어플리케이션 in-ae가 IN-CSE에 등록되면, in-ae는 스마트폰 어플리케이션 사용자가 조명 상태를 변경하기 위한 조명 제어 명령어를 보낼 수 있도록 IN-CSE에 위치한 컨테이너를 포함한 리소스에 접근하기 위해 허용될 수 있다.

스마트폰 어플리케이션 사용자가 스마트폰 사용자 인터페이스를 통해 조명 상태를 변경할 때, in-ae는 새로운 상태를 나르는 새로운 contentInstance 생성 절차를 수행한다

단일 조명 상태의 변경은 특정 컨테이너에 대한 새로운 contentInstance 리소스의 생성을 통해 구현될 수 있다. 예를 들면, 모든 조명 상태의 변경을 통해 접근 제어 정책 `acp1(gateway_acp)`을 가진 컨테이너는 접근 제어 정책 `acp1(gateway_acp)`을 그룹(`lightstates_grp`)의 각 멤버에 대한 새로운 contentInstance를 생성함으로써 구현될 수 있다. 후자인 경우의 구현은 그룹1(`lightstates_grp`)에 대한 새로운 contentInstance 리소스의 content를 가진 <fanOutPoint> 리소스를 생성하는 것이다. 그럼으로써, 그룹1의 모든 멤버의 content는 다같이 구현될 수 있다.

① ADN-AE1의 컨테이너 아래 contentInstance 생성

HTTP Request:

```
POST ~/mn-cse/home_gateway/light_ae1/light?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
Content-Type:application/vnd.onem2m-res+xml;ty=4
X-M2M-RI: mncse/11123

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <con>ON</con>
  <cnf>text/plain:0</cnf>
</m2m:cin>
```

HTTP Response:

```
201 Created
X-M2M-RSC: 2001
X-M2M-RI: mncse/11123
Content-Location: /mn-cse/cin-789356234
```

② ADN-AE2의 컨테이너 아래 contentInstance 생성

HTTP Request:

```
POST ~/mn-cse/mn-cse-nm/adn-ae2/light2?rcn=0 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Cin-ae
Content-Type:application/vnd.onem2m-res+xml;ty=4
X-M2M-RI: light2/12347
X-M2M-NM:CI6

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <con>ON</con>
  <cnf>text/plain:0</cnf>
</m2m:cin>
```

HTTP Response:

```
201 Created
```



```
X-M2M-RSC: 2001
X-M2M-RI: light2/12347
Content-Location: /mn-cse/Cadn-ae2/light2/CI6
```

③ 그룹 fanout을 사용하여 모든 조명의 상태를 업데이트

HTTP Request:

```
POST ~/mn-cse/home_gateway/gateway_ae/lightstates_grp/fopt HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
Content-Type:application/vnd.onem2m-res+xml
X-M2M-RI: mncse/33344
```

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
<con>ON</con>
<cnf>text/plain:0</cnf>
</m2m:cin>
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: mncse/33344
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:agr
  xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <m2m:rsp>
    <rsc>2001</rsc>
  <rqi>ae/33344</rqi>
  <pc/>
  <to>/~/in-cse/ae-CAE340304178</to>
  <fr>/~/mn-cse/cnt-582759912</fr>
  </m2m:rsp>
  <m2m:rsp>
    <rsc>2001</rsc>
```

```

        <rqi>ae/33344</rqi>
    <pc/>
    <to>/~/in-cse/ae-CAE340304178</to>
    <fr>/~/mn-cse/cnt-582769893</fr>
</m2m:rsp>
</m2m:agr>

```

5.7.11 알림(Notifications)

contentInstance가 ADN-AE의 컨테이너 아래에 생성될 때마다, 생성된 전체 contentInstance를 포함하는 최신 알림(Notifications)이 대상 구독자에게 게시된다. 즉, AE 어플리케이션 adn-ae1 이나 adn-ae2 는 각각 알림으로 수신 받은 조명의 새로운 상태를 정확히 할 수 있다.

① ADN-AE1에 알림 게시(Post a notification)

HTTP Request:

```

POST / HTTP/1.1
Host: http://192.168.0.10:9090
X-M2M-Origin: /mn-cse
X-M2M-RI: notif/12345
Content-Type: application/vnd.onem2m-ntfy+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:sgn
  xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <nev>
    <rep>
      <cin>
        <con>ON</con>
        <cnf>text/plain:0</cnf>
      </cin>
      <rss>201</rss>
    </rep>
  </nev>
  <sur>
    /~/mn-cse/sub-856593979
  </sur>
</m2m:sgn>

```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: notif/12345
```

② ADN-AE2에 알림 게시(Post a notification)

HTTP Request:

```
POST / HTTP/1.1
Host: http://192.168.0.20:9090
X-M2M-Origin: /mn-cse
X-M2M-RI: notif/12346
Content-Type: application/vnd.onem2m-ntfy+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:sgn
  xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <nev>
    <rep>
      <cin>
        <con>ON</con>
        <cnf>text/plain:0</cnf>
      </cin>
      <rss>201</rss>
    </rep>
  </nev>
</sur>
/~mn-cse/sub-856463728
</sur>
</m2m:sgn>
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
X-M2M-RI: notif/12346
```

제6장 결론

본 해설서에서 설명한 원격 조명 제어 유스케이스는 oneM2M 서비스 플랫폼에 등록되어있는 리소스에 접근하도록 허용되는 스마트폰 어플리케이션을 사용하여 원격으로 조명 램프를 제어하는 기본적인 상황에 대해서 개발자들이 손쉽게 구현해 볼 수 있도록 구성되었다. 본 유스케이스는 스마트 디바이스의 등록과 같은 상위 수준의 프로시저에 따라 수행되는 절차들을 상세히 실제 예제와 함께 설명하였다. 예를 들면, oneM2M 서비스 플랫폼의 게이트웨이, container와 contentInstances 생성, 그룹 리소스 생성, contentInstance 검색, contentInstances 등록 및 알림 등이 본 해설서에서 상세히 설명되어졌다.

본 해설서에서 설명되어진 원격 조명 제어 유스케이스를 oneM2M 서비스 플랫폼에 매핑하는 방법은 다음과 같이 몇 가지 주의할 점이 있다.

- » oneM2M 프리미티브의 CoAP바인딩과 JSON 직렬화는 본 해설서에서 다루지 않고 있으며, 해당 유스케이스는 별도의 기술 문서로 oneM2M Test Working Group 에서 개발되어질 예정이다.
- » 본 유스케이스는 2015년 11월에 개최된 oneM2M Technical Plenary #20 미팅에서 합의된 여러 내용들을 기반으로 작성이 되었기 때문에, 이후 표준에 반영되는 여러 내용들 및 oneM2M Release 2.0에 대한 내용들은 매 Technical Plenary 미팅 후에 업데이트 되는 내용들을 확인하고 개발을 진행하기를 바란다.

제7장 참고 문헌

- [1] 이효은, 홍승표, 『IoT현황 및 주요 이슈』, 정보통신기술진흥센터, page 7, 2014년 12월.
- [2] 송재승, 이규명, 서정욱, 강남희, 김재호, 최성찬, 양현석, 최희동, 정승명, 김성윤, 안홍범, 『oneM2M 서비스 플랫폼 표준 해설서』, 한국정보통신기술협회, pages 26-40, 2014년 11월.
- [3] oneM2M, "TS-0001 Functional Architecture", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [4] oneM2M, "TS-0002 Requirements", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [5] oneM2M, "TS-0003 Security Solutions", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [6] oneM2M, "TS-0004 Core Protocol", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [7] oneM2M, "TS-0005 Management Enablement (OMA)", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [8] oneM2M, "TS-0006 Management Enablement (BBF)", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [9] oneM2M, "TS-0007 Study of Abstraction and Semantics Enablements", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [10] oneM2M, "TS-0008 CoAP Protocol Binding", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [11] oneM2M, "TS-0009 HTTP Protocol Binding", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [12] oneM2M, "TS-0010 MQTT Protocol Binding", oneM2M TECHNICAL SPECIFICATION, January 2015.

- [13] oneM2M, "TS-0011 Common Terminology", oneM2M TECHNICAL SPECIFICATION, January 2015.
- [14] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M," IEEE Wireless Commun. Mag., vol.21, no.3, pp.20-26, June 2014.
- [15] J.Song, A. Kunz, M. Schmidt, and P. Szczytowski. "Connecting and managing m2m devices in the future internet," Mobile Networks and Applications, pages 1-14, 2013.
- [16] S. Husain, A. Kunz, J. Song, T. Koshimizu, "Interworking Architecture Between oneM2M Service Layer and Underlying Networks," Third IEEE Workshop on Telecommunication Standards: From Research to Standards, 8 Dec. 2014, Part of IEEE Globecom 2014, Austin, TX, USA.

제8장 Annex A Reading 리소스

본 유스케이스의 각 요청 흐름 과정에서 만들어진 리소스의 정보는 스마트폰 어플리케이션 IN-AE이 다음과 같은 대상 리소스에 RETRIEVE 요청 받기 시작하는 것을 통해서 검색될 수 있다.

본 장에서는 8.1에서 CSE resources, 8.2에서는 게이트웨이 디바이스 어플리케이션인 MN-AE, 8.3에서는 조명 디바이스 어플리케이션인 ADN-AE, 8.4에서는 스마트폰 어플리케이션인 IN-AE, 8.5에서는 접근 제어 정책, 8.6에서는 컨테이너, 8.7에서는 contentInstance, 8.8에서는 구독(Subscriptions), 8.9에서는 그룹 관리에 대한 구현 절차를 나타내었다.

8.1 Annex A.1 CSE resources

8.1.1 Annex A1.1 IN-CSE

HTTP Request:

```
GET /~/in-cse/server HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: incse/12345
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:cb
  xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="server">
  <ty>5</ty>
</ri>/~/in-cse</ri>
  <ct>20150925T045938</ct>
```

```

<lt>20150925T045938</lt>
<acpi>/~/in-cse/acp-666957710</acpi>
<cst>1</cst>
<csi>in-cse</csi>
<srt>1 2 3 4 5 9 14 15 16 17 23</srt>
<poa>http://in.provider.com:8080</poa>
</m2m:cb>

```

8.1.2 Annex A.1.2 MN-CSE

HTTP Request:

```

GET /~/in-cse/server/home_gateway HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: incse/12346
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="home_gateway">
  <ty>16</ty>
  <ri>csr-299409504</ri>
  <pi>/~/in-cse</pi>
  <ct>20150925T045938</ct>
  <lt>20150925T045938</lt>
  <et>20171005T105550</et>
  <acpi>/~/in-cse/acp-666957710</acpi>
  <poa>http://mn.provider.com:8080</poa>
  <cb>mn.provider.com/mn-cse</cb>
  <csi>/mn-cse</csi>
  <rr>TRUE</rr>
</m2m:csr>

```


8.2 Annex A.2 게이트웨이 디바이스 어플리케이션 MN-AE

HTTP Request:

```
GET /~/mn-cse/home_gateway/gateway_ae HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/12347
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway_ae">
  <ty>2</ty>
  <ri>ae-CAE340303271</ri>
  <pi>/~/mn-cse</pi>
  <ct>20150925T052438</ct>
  <lt>20150925T052438</lt>
  <et>20171005T105550</et>
  <acpi>/~/mn-cse/acp-805496226</acpi>
  <api>A01.com.company.gatewayApp</api>
  <aei>CAE340303271</aei>
  <rr>FALSE</rr>
</m2m:ae>
```

8.3 Annex A.3 조명 디바이스 어플리케이션

8.3.1 Annex A.3.1 ADN-AE1

HTTP Request:

```
GET /~/mn-cse/home_gateway/light_ae1 HTTP/1.1
Host:http://mn.provider.com:8080
```

X-M2M-Origin:/mn-cse/Cgateway_ae

X-M2M-RI: mncse/12347

Accept:application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

Content-Type: application/vnd.onem2m-res+xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light_ae1">
  <ty>2</ty>
<ri>ae-CAE340304071</ri>
  <pi>/~/mn-cse</pi>
  <ct>20150925T052455</ct>
  <lt>20150925T052455</lt>
  <et>20171005T105550</et>
<acpi>/~/mn-cse/acp-805496226</acpi>
  <api>A01.com.company.lightAppl</api>
<aei>CAE340304071</aei>
<rr>TRUE</rr>
</m2m:ae>
```

8.3.2 Annex A.3.2 ADN-AE2

HTTP Request:

GET /~/mn-cse/home_gateway/light_ae2 HTTP/1.1

Host:http://mn.provider.com:8080

X-M2M-Origin:/mn-cse/Cgateway_ae

X-M2M-RI: mncse/12348

Accept:application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

Content-Type: application/vnd.onem2m-res+xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light_ae2">
  <ty>2</ty>
  <ri>ae-CAE340304042</ri>
  <pi>/~/mn-cse</pi>
  <ct>20150925T052542</ct>
  <lt>20150925T052542</lt>
  <et>20171005T105550</et>
  <acpi>/~/mn-cse/acp-805496226</acpi>
  <api>A01.com.company.lightApp2</api>
  <aei>CAE340304042</aei>
  <rr>TRUE</rr>
</m2m:ae>
```

8.4 Annex A.4 스마트폰 어플리케이션 IN-AE

HTTP Request:

```
GET /~/in-cse/server/smartphone_ae HTTP/1.1
Host:http://in.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: incse/12349
Accept:application/xml
```

Resonse status:

```
200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml
```

Resonse message:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="smartphone_ae">
  <ty>2</ty>
  <ri>ae-CAE340304178</ri>
  <pi>/~/in-cse</pi>
```

```

    <ct>20150925T052622</ct>
    <lt>20150925T052622</lt>
    <et>20171005T105550</et>
<acpi>/~/in-cse/acp-666957710</acpi>
    <api>A01.com.company.lightControlApp</api>
    <aei>CAE340304178</aei>
<rr>FALSE</rr>
</m2m:ae>

```

8.5 Annex A.5 접근 제어 정책

```
GET /mn-cse/acp1 HTTP/1.1
```

HTTP Request:

```

GET /~/mn-cse/home_gateway/gateway_acp HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12350
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<m2m:acp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway_acp">
<ty>1</ty>
<ri>acp-805496226</ri>
    <pi>/~/mn-cse</pi>
    <ct>20150925T050238</ct>
    <lt>20150925T050238</lt>
    <et>20171005T105550</et>
<pv>
<acr>
    <acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse/
Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>

```

```

        <acop>63</acop>
    </acr>
</pv>
<pvs>
    <acr>
        <acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse/
Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>
        <acop>51</acop>
    </acr>
</pvs>
</m2m:acp>

```

8.6 Annex A.6 컨테이너

8.6.1 Annex A.6.1 ADN-AE1 아래 컨테이너

```

GET /mn-cse/Cadn-ae1/light1 HTTP/1.1
Host:http://m2m.provider.com:8080
X-M2M-Origin:/mn-cse/Cmn-ae
X-M2M-RI: Cmn-ae/12351
Accept:application/xml

```

Resonse status:

200 OK

Content-Type: application/vnd.onem2m-res+xml; charset=utf-8

Resonse message:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light1">
    <ty>3</ty>
    <ri>cnt-58275991</ri>
    <pi>/mn-cse/Cadn-ae1</pi>
    <ct>20150925T052955</ct>
    <lt>20150925T052955</lt>
    <et>20171005T105550</et>
    <acpi>/mn-cse/acp-80549622</acpi>

```

```

<lb1/>
<st>0</st>
<at/>
<aa/>
<cr/>
<mni/>
<mbs/>
<mia/>
<cni>1</cni>
<cbs>3</cbs>
<li/>
<or/>
</m2m:cnt>

```

8.6.2 Annex A.6.2 ADN-AE2 아래 컨테이너

```
GET /mn-cse/acpl HTTP/1.1
```

HTTP Request:

```

GET /~/mn-cse/home_gateway/gateway_acp HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12350
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<m2m:acp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway_acp">
<ty>1</ty>
<ri>acp-805496226</ri>
  <pi>/~/mn-cse</pi>
  <ct>20150925T050238</ct>
  <lt>20150925T050238</lt>
  <et>20171005T105550</et>
<pv>

```

```

<acr>
    <acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse/
Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>
    <acop>63</acop>
</acr>
</pv>
<pvs>
    <acr>
    <acor>/~/in-cse/home_gateway /~/mn-cse/Cgateway_ae /~/mn-cse/
Clight_ae1 /~/mn-cse/Clight_ae2 /~/in-cse/Csmartphone_ae</acor>
    <acop>51</acop>
    </acr>
</pvs>
</m2m:acp>

```

8.7 Annex A.7 ContentInstances

8.7.1 Annex A.7.1 ADN-AE1에서 최신 contentInstance

HTTP Request:

```

GET /~/mn-cse/home_gateway/light_ae1/light/la HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12353
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin
  xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-394798749">
  <ty>4</ty>

```

```

<ri>cin-394798749</ri>
<pi>/~/mn-cse/cnt-582759912</pi>
  <ct>20150925T053225</ct>
  <lt>20150925T053225</lt>
<et>20171005T105550</et>
<st>0</st>
<cnf>text/plain:0</cnf>
<cs>2</cs>
  <con>ON</con>
</m2m:cin>

```

8.7.2 Annex A.7.2 ADN-AE2에서 최신 contentInstance

HTTP Request:

```

GET /~/mn-cse/home_gateway/light_ae2/light/la HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12354
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin
  xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-256599578">
  <ty>4</ty>
<ri>cin-256599578</ri>
  <pi>/~/mn-cse/cnt-582769893</pi>
  <ct>20150925T053425</ct>
  <lt>20150925T053425</lt>
<et>20171005T105550</et>
<st>0</st>
<cnf>text/plain:0</cnf>

```



```

<cs>2</cs>
    <con>ON</con>
</m2m:cin>

```

8.8 Annex A.8 구독(Subscriptions)

8.8.1 Annex A.8.1 ADN-AE1에서 컨테이너 구독

HTTP Request:

```

GET /~/mn-cse/home_gateway/light_ae1/light/lightstate_sub1 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12355
Accept:application/xml

```

HTTP Response:

```

200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate_sub1">
    <ty>23</ty>
    <ri>sub-856593979</ri>
<pi>/~/mn-cse/cnt-582759912</pi>
    <ct>20150926T052955</ct>
    <lt>20150926T052955</lt>
    <et>20171005T105550</et>
    <acpi>/~/mn-cse/acp-805496226</acpi>
    <enc>
<net>3</net>
</enc>
    <nu>/~/mn-cse/ae-CAE340304071</nu>
    <nct>1</nct>
</m2m:sub>

```

8.8.2 Annex A.8.2 ADN-AE2에서 컨테이너 구독

HTTP Request:

```
GET /~/mn-cse/home_gateway/light_ae2/light/lightstate_sub2 HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/mn-cse/Cgateway_ae
X-M2M-RI: mncse/12356
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate_sub2">
  <ty>23</ty>
  <ri>sub-856463728</ri>
  <pi>/~/mn-cse/cnt-582759912</pi>
  <ct>20150926T053055</ct>
  <lt>20150926T053055</lt>
  <et>20171005T105550</et>
  <acpi>/~/mn-cse/acp-805496226</acpi>
  <enc>
    <net>3</net>
  </enc>
  <nu>/~/mn-cse/ae-CAE340304042</nu>
  <nct>1</nct>
</m2m:sub>
```

8.9 Annex A.9 그룹

8.9.1 Annex A.9.1 그룹 1

HTTP Request:

```
GET /~/mn-cse/home_gateway/gateway_ae/containers_grp HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/12357
Accept:application/xml
```

HTTP Response:

```
200 OK
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+xml

<?xml version="1.0" encoding="UTF-8"?>
<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="containers_grp">
  <ty>9</ty>
  <ri>grp-977978327</ri>
  <pi>/~/mn-cse/ae-CAE340303271</pi>
  <ct>20151004T045954</ct>
  <lt>20151004T045954</lt>
  <et>20171005T105550</et>
  <acpi>/~/mn-cse/acp-805496226 /~/mn-cse/acp-805496226</acpi>
  <mt>3</mt>
  <cnm>2</cnm>
  <mnrm>10</mnrm>
  <mid>/~/mn-cse/cnt-582759912 /~/mn-cse/cnt-582769893</mid>
  <mtv>true</mtv>
  <csy>1</csy>
  <fopt>/~/mn-cse/grp-977978327/fopt</fopt>
</m2m:grp>
```

8.9.2 Annex A.9.2 그룹 2

HTTP Request:

```
GET /~/mn-cse/home_gateway/gateway_ae/lightstates_grp HTTP/1.1
Host:http://mn.provider.com:8080
X-M2M-Origin:/in-cse/Csmartphone_ae
X-M2M-RI: mncse/12358
```

Accept:application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

Content-Type: application/vnd.onem2m-res+xml

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstates_grp">
  <ty>9</ty>
  <ri>grp-677974227</ri>
  <pi>/~/mn-cse/ae-CAE340303271</pi>
  <ct>20151004T044954</ct>
  <lt>20151004T044954</lt>
  <acpi>/~/mn-cse/acp-805496226 /~/mn-cse/acp-805496226</acpi>
  <et>20171005T105550</et>
  <mt>4</mt>
  <cnm>2</cnm>
  <mn>10</mn>
  <mid>/~/mn-cse/cnt-582759912/1a /~/mn-cse/cnt-582769893/1a</mid>
  <mtv>true</mtv>
  <csy>1</csy>
  <fopt>/~/mn-cse/grp-677974227/fopt</fopt>
</m2m:grp>
```

oneM2M 어플리케이션 개발자 가이드 표준 해설서

발행인 : 임차식

발행처 : 한국정보통신기술협회

(13591) 경기도 성남시 분당구 분당로 47

Tel : 031-724-0114, Fax : 031-724-0109

발행일 : 2015. 11. 30

1. 이 자료는 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 발간된 자료입니다.
2. 이 자료집의 무단 복제를 금하며, 내용을 인용할 시에는 반드시 정부기금사업의 결과임을 밝혀야 합니다.

설문조사

■ 표준해설서명 :

먼저 본 표준해설서를 이용해 주신 여러분께 감사드리며, 향후 발행할 표준해설서에 활용하기 위하여 다음과 같이 설문을 요청하오니, 협조하여 주시면 감사하겠습니다.

1. 본 표준해설서에 대해 어느 정도 만족하십니까? ()
① 매우불만족 ② 불만족 ③ 보통 ④ 만족 ⑤ 매우만족
2. 본 표준해설서의 난이도는 어떻다고 생각하십니까? ()
① 매우 어렵다 ② 어렵다 ③ 보통이다 ④ 쉽다 ⑤ 매우 쉽다
3. 본 표준해설서를 어떤 경로로 입수하시게 되었습니까? ()
① 우편배송 ② TTA 홈페이지 ③ TTA ICT Standard Weekly ④ TTA e-mail
⑤ 주위 동료나 상사의 권유 ⑥ 기타 ()
4. 본 표준해설서에 대한 추가, 보완 및 개선사항이 있다면 아래에 기재하여 주십시오.

--

5. 표준해설서 발간을 희망하는 표준이 있다면 아래에 기재하여 주십시오.

- TTA표준번호 :

- TTA표준명 :

*표준검색 방법 : TTA홈페이지(www.tta.or.kr)→자료마당→자료검색

본 설문조사에 응해주신 분께는 추첨을 통해 소정의 기념품을 드리겠습니다.

 기념품을 받기위한 주소

(우편번호:)

 보내실 곳 : spyu@tta.or.kr

◇ 작성해 주셔서 감사합니다. ◇

