Wi-Fi 내 Direct 통신 프레임워크

이름	학번	이메일	전화번호
김세호	201011317	seho00@naver.com	010-9568-8280
박대규	201011329	qkreorb0321@naver.com	010-9552-1624
차금옥	201011369	lovecko123@naver.com	010-9437-0713

지도 교수 : 민덕기 교수님 (인)

목차

[그림 목차]

1. 개발 동기

최근 스마트폰의 개발과 발달로 인해 인간의 삶은 더욱 윤택해졌다. 손쉽고 빠르게 많은 정보를 얻을 수 있으며, 또한 어플리케이션을 통해 다양하게 활용이 가능하다. 스마트폰과 같은 IT 기술의 진화는 인간의 삶에 많은 변화를 만들어냈다. PC가 등장하고, 이후 스마트폰이 등장하며 새로운 혁신이 만들어진 것처럼, 스마트 폰 이후에도 또 다른 진화와 혁신이 만들어질 것이다. 이러한 맥락에서, IoT라는 새로운 패러다임이 주목을 받고 있다.

IoT(Internet Of Thing)는 최근 IT분야에서 가장 주목 받고 있는 기술 중 하나로, 사물간에 Network를 통하여 서로 상호작용하는 기술을 말한다. 이러한 Mechanism은 추후 더욱 발전되어 인간의 생활에 필수적인 기술이 될 것이라 예상한다. 그러므로 사물간의 네트워크를 필요로 하는 어플리케이션을 구현하고는 개발자들에게 좀 더 손쉽게 프로그램을 구현할 수 있도록 프레임워크를 개발하고자 한다. 특별히 Wi-Fi 내의 Device간 Direct Communication을 가능하게 하는 기능을 제공하여, Application Level에서 TCP Communication을 고려하지 않도록 편의성을 제공한다.

Windows 플랫폼 뿐만 아니라 Android나 다른 플랫폼 상에서도 쉽게 사용할 수 있는 cross-platform framework를 구현하고 Window과 Android를 기반으로 사용할 수 있는 Framework 개발을 목표로 한다. 가장 먼저 고려했던 것은 Android 플랫폼을 탑재한 스마트폰과 windows나 여타 플랫폼을 기반으로 한 laptop 간의 Network이다. IoT의 대표적인 기술 Alljoyn Framework를 분석하고 좀 더 보완된 framework를 구현하고자 한다. 이 Framework를 통하여 Wi-Fi network에 접속된 device간의 손쉬운 communication을 가능하도록 한다.

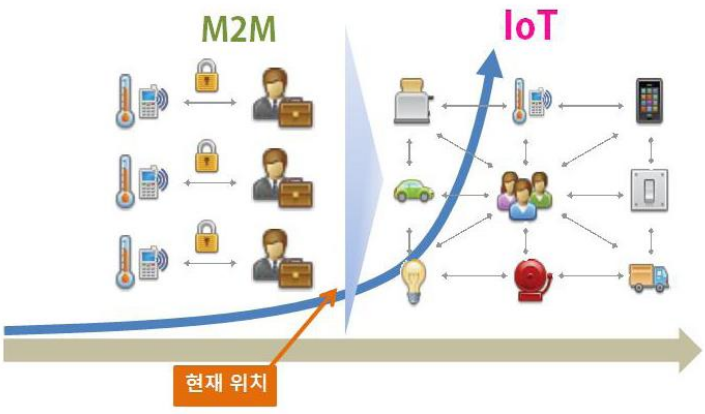
이러한 Framework를 개발한 후 그 Framework를 기반으로 하는 Application을 제작한다. 통합 Device 관리 Service인 Device Cloud를 분석하여 사용자 중심의 IoT Application을 구현하고자 한다. Device를 통합적으로 관리하고 제어할 수 있는 Application 구현을 목표로 한다.

2. 관련 연구

2.1. IoT(Internet of Thing)

[그림1] IoT 개념도

- 사람과 사물, 서비스 세가지 분산된 환경 요소에 대해 인간의 명시적 개입 없이 상호 협력적으로 sensing, networking, information processing 등 지능적 관계를 형성하는 사물 공간 연결 망을 말한다.
- IoT의 주요 구성 요소인 사물(Thing)은 유무선 네트워크에서의 end-device 뿐만 아니라, 인간 차량, 교량, 각종 전자장비, 문화재, 자연 환경을 구성하는 물리적 사물 등이 포함된다.
- 이동통신망을 이용하여 사람과 사물, 사물과 지능통신을 할 수 있는 M2M의 개념을 인터넷으로 확장하여 사물은 물론, 현실과 가상세계의 모든 정보와 상호작용하는 개념으로 진화하고 있다.



[그림2] M2M과 IoT의 개념변화

2.1.1. IoT 3대 주요 기술

1) Sensing Technology

- 전통적인 온도/습도열/가스/조도/초음파 센서 등에서부터 원격 감지, SAR, 레이더, 위치, 모션, 영상 센서 등 유형 사물과 주위 환경으로부터 정보를 얻을 수 있는 물리적 센서를 포함한다.

- 물리적인 센서는 응용 특성을 좋게 하기 위해 표준화된 인터페이스와 정보 처리 능력을 내장한 스마트 센서로 발전하고 있으며, 또한, 이미 sensing한 데이터로부터 특정 정보를 추출하는 virtual sensing 기능도 포함되며 virtual sensing 기술은 실제 IoT 서비스 인터페이스에 구현되고 있다.

- 기존의 독립적이고 개별적인 센서보다 한 차원 높은 다중(다분야) 센서기술을 사용하기 때문에 한층 더 지능적이고 고차원적인 정보를 추출할 수 있다.

2) 유무선 통신 및 네트워크 인프라 기술

- IoT의 유무선 통신 및 네트워크 장치로는 기존의 WPAN, WiRi, 3G/4G/LTE, Bluetooth, Ethernet, BcN, 위성통신, Microware, 시리얼 통신, PLC 등, 인간과 사물, 서비스를 연결시킬 수 있는 모든 유·무선 네트워크를 의미 한다.

  ※ WPAN(Wireless Personal Area Networks)

  ※ 시리얼 통신: 일반적으로 컴퓨터 기기를 접속하는 방법의 하나로, 접속하는 선의 수를 줄이고, 원거리까지 신호를 보낼 수 있도록 한 통신 방식이다.

3) IoT 서비스 인터페이스 기술

- IoT서비스 인터페이스는 IoT의 주요 3대 구성 요소(인간 · 사물 · 서비스)를 특정 기능을 수행하는 응용서비스와 연동하는 역할을 한다.

- IoT 서비스 인터페이스는 네트워크 인터페이스의 개념이 아니라 여러 서비스 제공(정보를 sensing, 가공/추출/처리, 저장, 판단, 상황 인식, 인지, security/privacy protection, 인증/인가, discovery, 객체 정형화, ontology 기반의 semantic, open senser API, 가상화, 위치확인, 프로세스 관리, 오픈 플랫폼 기술, 미들웨어 기술, 데이터 마이닝 기술, 웹 서비스 기술, social network 등)을 위해 인터페이스(저장, 처리, 변환 등) 역할을 수행하는 것을 말한다.

2.1.2. IoT 기술 동향 - 개방형 IoT 디바이스 및 표준플랫폼 기술 동향

2.1.2.1. 국내 IoT/M2M 기술 동향

- 2012년 국내 최초로 전자부품연구원에서는 건물/빌딩 에너지 통합 관제 및 제어기술을 IoT를 Open Web기반 기술들을 중심으로 정의한 WoT(Web of Things)기술로 개발하였음. 본 기술은 다양한 스마트 에너지 관리 기술에 적용이 가능한 수준이며, 향후 클라우드 및 분산환경에 적용이 가능함. 현재 본 기술은 Oracle에서도 매우 큰 관심을 보이고 있으며, 활발한 후속 연구 기획이 진행중임

- 2010년 국내 대표적인 통신기업인 KT에서 M2M 플랫폼을 개발하여 사물인터넷을 통한 USN기술이 통신사 서비스플랫폼에 연결되는 결과를 만든바 있으며, 이후 Open API기반의 KT의 KHub를 통해 유비쿼터스 미들웨어 기반 기술도 확보한바 있음

- 2012년 SKT에서는 국제 표준 규격을 준수한 서버와 단말 플랫폼으로 구성되는 개방형 M2M플랫폼을 개발하여, 상용화 서비스를 제공하고 있음. 이는 현재 국내에서 배포한 상용 M2M플랫폼 중 가장 완성도가 높은 상용 시스템으로 향후 국내 M2M시장의 성장을 크게 견인할 것으로 평가 받고 있음

- 전체적으로 국내의 경우, M2M 단말 및 플랫폼 기술은 국책 연구를 넘어서 상용 화 단계로 추진되고 있지만, IoT 플랫폼은 연구 초기 단계임. 삼성전자, LG전자 등 국내업체들이 글로벌 가전시장을 양분하고 있으나 IoT를 활용한 장기적 융합전략 부재로 IoT 시장에서 큰 성과를 보이지 못하고 있음

- 따라서, 규격화된 통신 모듈 및 다양한 센서의 생산과 보급, 무선네트워크 기술 발전에 따른 표준화 적극 참여, IoT 플랫폼 관련 개방성 확보 등의 노력을 적극적으로 추진하는 등의 IoT 서비스 활성화 노력이 연속적으로 필요함

2.1.2.2. 국내 IoT/M2M 표준화 방향

- 우리나라의 경우 국책연구소 및 학교, 그리고 이동통신 사업자(KT, SKT와 LG U+)들과 삼성, LG 등 단말기사업자를 중심으로 M2M 기술 및 서비스 전략을 개발하고 있으며, 2009년에는 방송통신위원회 산하에 IoT/M2M 포럼이 신설되어 표준화 활동을 추진하고 있음

- 2011년에는 정보통신기술협회(TTA) 이동통신 기술위원회(TC7) 산하에 사물지능통신 프로젝트 그룹(PG708)이 생성되어 국가차원의 M2M 표준기술 개발을 기대할 수 있게 되었음

2.1.2.3. 국외 IoT 기술 동향

- 기업 IT 인프라가 인터넷을 기반으로 한 시스템으로 전환되면서 인터넷 아키텍처(Internet Oriented Architecture)에 대한 관심이 증대되고 있으며 IBM은 스마트 그리드 서비스/관리 플랫폼 분야, Intel은 GE와 스마트 그리드 표준화 분야, Cisco는 보안 통신 인프라분야, 구글은 GE와 스마트미터 협업을 통한 Power Meter 서비스 및 웹-에너지 IoT정보 플랫폼 분야에서 두각을 나타내고 있음. Comverge는 스마트계량기 기반 가정에너지 사용관리 분야, EnerNoc은 가정용 웹전력시스템 분야에 활발한 제품개발을 추진 중에 있음 ISSUE 11 개방형 IoT 플랫폼 기술 동향

- 2012년 윈드리버는 M2M 어플리케이션에 특화된 소프트웨어 개발 환경인 '윈드리버 인텔리전트 디바이스 플랫폼'을 발표하여, M2M 디바이스 개발에 필요한 다양한 소프트웨어 모듈을 상용 컴포넌트 형태로 제공하여 개발 시간을 줄일 수 있도록 하였음

- 개방형 IoT 디바이스로는 Arduino, Nanode가 있음. Arduino는 오픈소스 Wiring 플랫폼의 연장으로 개발된 오픈소스 싱글보드 마이크로컨트롤러로서 이종프로젝트(multi-disciplinary projects)에서 사용되는 전자기기들의 접근성을 용이하게 하기위한 목적으로 개발되었음. Nanode는 유무선 이더넷 인터페이스를 기반으로한 웹 접근성이 기본으로 구현이 된 Arduino와 같은 오픈소스형 하드웨어임

- 대표적인 실시간 개방형 IoT 플랫폼 및 응용서비스 구축 사례로는 사람들과 사물, 어플리케이션, 그리고 IoT에 이어주기 위한 목적으로 개발된 세계최초 상용화 IoT프랫폼 서비스인 Pachube가 있음. 영국에 본사가 있는 Pachube는 인터넷 기반 서비스를 통해 전세계의 데이터를 실시간으로 관리할 수 있으며 등록한 사람들에게 전세계로부터 수집한 정보들을 공유하고 협업할 수 있는 환경을 제공해 줌

2.1.1.4. 국외 IoT 표준 동향

- IoT 기술 표준은 유럽을 중심으로 다양한 연구 및 기술개발을 진행하고 있고, EC는 정책적으로 확산에 주력하고 있는 가운데, 미래정보화 시대의 비전으로 자리매김하고 있음. ITU-T를 중심으로 IoT에 관련된 표준화 활동이 2010년에 제안되었으며, 2011년 2월 새로운 표준화 영역으로 설정하여 ITU-T의 관련 표준화 그룹들이 한 곳에 모여 협력 개발할 수 있도록 2011년 5월 IoT-GSI(Global Standards Initiative)를 설치하여 운영하기 시작하였음

2.1.3. IoT 발전 방향

- IoT 시장을 빠르게 확대하기 위해서는 소비자 수요, 투자, 그리고 관련 정책계획의 균형적 추진이 필수적이므로, 수요중심으로의 패러다임 전환에 적합한 시장 환경을 마련해야 한다.

- 국제표준 기반의 개방형 IoT 플랫폼 개발 및 확보 : Network Connectivity와 IoT 응용서비스에 대해 표준/개방형 인터페이스 제공 필요하다.

- 모듈/단말 제조업체, 솔루션 사업자 등의 직접적 이해당사자뿐만 아니라 대학 및 연구소, 정부 등의 협력을 통한 IoT 사업 선 순환 체계 형성 필요하다

- 크리티컬 매스에 도달하기 위해서는 수익성 확보와 함께 관련 업체들 간의 협업이 중요하며, 협업의 구심점이 되는 IoT 플랫폼을 장악하는 기업이 IoT 시장의 리더가 될 것으로 예상된다.

- 국내에서도 IoT 디바이스, 네트워크, 애플리케이션뿐만 아니라 IoT 플랫폼역량을 가진 업체가 등장해야 하며, 그렇지 못할 경우 해외 업체의 플랫폼에 종속될 가능성이 있다.

- IoT 지원 Access망 확대, 서비스 범위 확장, 서비스 유형 확대, 지원 부가서비스 확대를 목표로 Multi-Access 기반 개방형 IoT 플랫폼으로 고도화 추진중이다.

2.2 유사 Application

2.2.1. alljoyn

[그림3] AllJoyn

- AllJoyn은 퀄컴(QIC)에서 만든 Converged Solution으로, Allseen Alliance라는 컨소시엄

을 통해 공개하였다. Quacomm 산하 Qualcomm Inovation Center(QuIC)가 개발한 오픈 소스 IoT Language이다.

- 최근 AllJoyn은 IoT의 바람을 타고 국내외 ICT업계에서 본격적으로 검토되고 있는 중요 플랫폼으로 성장하고 있다. 실제 2013년도 국내 전자 제조사에서 TV에 적용이 되었고, 통신사에서도 Converged Home Service로 활발히 검토 중이다.

- Alljoyn은 근거리 기반의 기기간 peer-to-peer기술로 중계서버(termidiary server)없이 device와 device가 direct로 통신할 수 있도록 하는 Framework이다. Windows, linux, android 환경에서 alljoyn에 기반을 둔 application을 개발할 수 있다. Alljoyn 기술은 같은 AP안의 Wi-Fi에 연결 되어있는 Device간의 통신을 하도록 한다.

- AllJoyn을 통해 서로 다른 플랫폼을 가진 모든 스마트기기들은 서로의 변화를 감지, 제어할 수 있고, 그들간의 자원(resources)을 공유할 수 있다. 스마트기기들 간의 자원을 이용한 다양한 어플리케이션을 생각해볼 수 있다. 예를 들어 TV를 통해 냉장고의 온도를 조작하고, 커피머신을 작동시키고, 초인종 카메라를 볼 수 있는 기능을 생각할 수 있다.

- Alljoyn는 Application 개발자가 Network Layer에서 어떤 통신 기술을 사용하는지 고려하지 않아도 된다. 개념적으로는 Alljoyn 프레임워크가 알아서 Wi-Fi 혹은 Bluetooth를 상황에 맞게 선택하나 현실적으로는 동일 Wi-Fi 내에 접속 중인 Device끼리 통신한다.

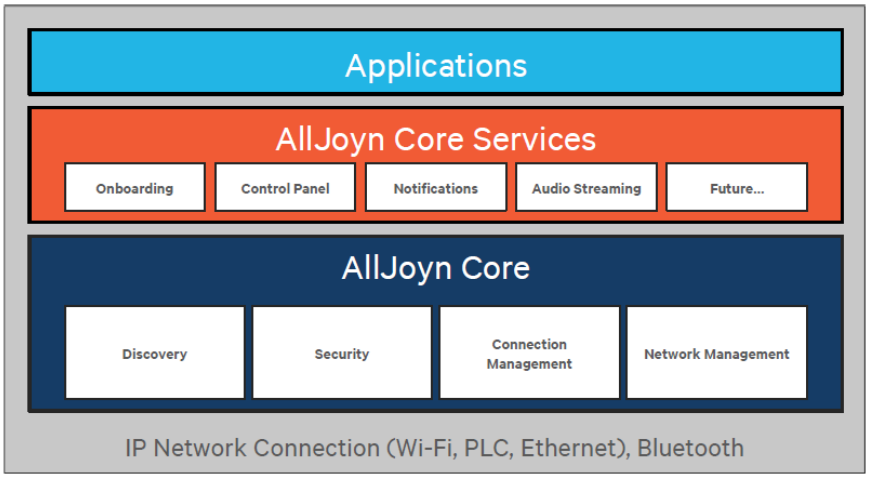
[그림4] AllJoyn Converged Home Service

-연결된 Device들은 RMI(Remote Method Invocation)방식으로 데이터를 전달한다. Annotaion을 추가한 interface 파일을 서로 공유하고, proxy를 통해 해당 interface의 method를 호출하는 방식이다. 소켓 프로그래밍처럼 주고 받을 데이터의 순서와 타입에 대

한 약속을 하고 예외처리를 할 필요 없이 method를 정의하고 호출하는 방식으로 동작한다.

-Alljoyn Daemon : Alljoyn기반의 Application이 정상적으로 다른 Device와 통신하기 위해서는 각 device에 한 개 이상의 Alljoyn daemon이 동작하고 있어야 한다. Application 개발자는 Alljoyn library를 링크하고 여기에 정의된 API를 호출하는 방식으로 Daemon에 Service를 등록하거나 Client로서 Service를 찾고 연결하여 method를 호출할 수 있다.

-Daemon이 설치된 Device가 같은 네트워크 안에 연결되면, Daemon이 서로의 존재를 인지하고 정보를 교환한다. 교환하는 정보는 자신에게 연결된 Service의 종류, 자신에게 연결된 Client와 해당 Client가 요청하는 Service 등이 있다. 개념적으로는 같은 네트워크 안에 연결된 Device가 하나로 연결되어 가상의 Alljoyn Bus가 만들어지는 형태이다. Device의 경계는 사라지고 Alljoyn Bus안에서 Client가 제공하는 Service의 종류와 Client가 요청하고 있는 Service의 종류가 중요해졌다.



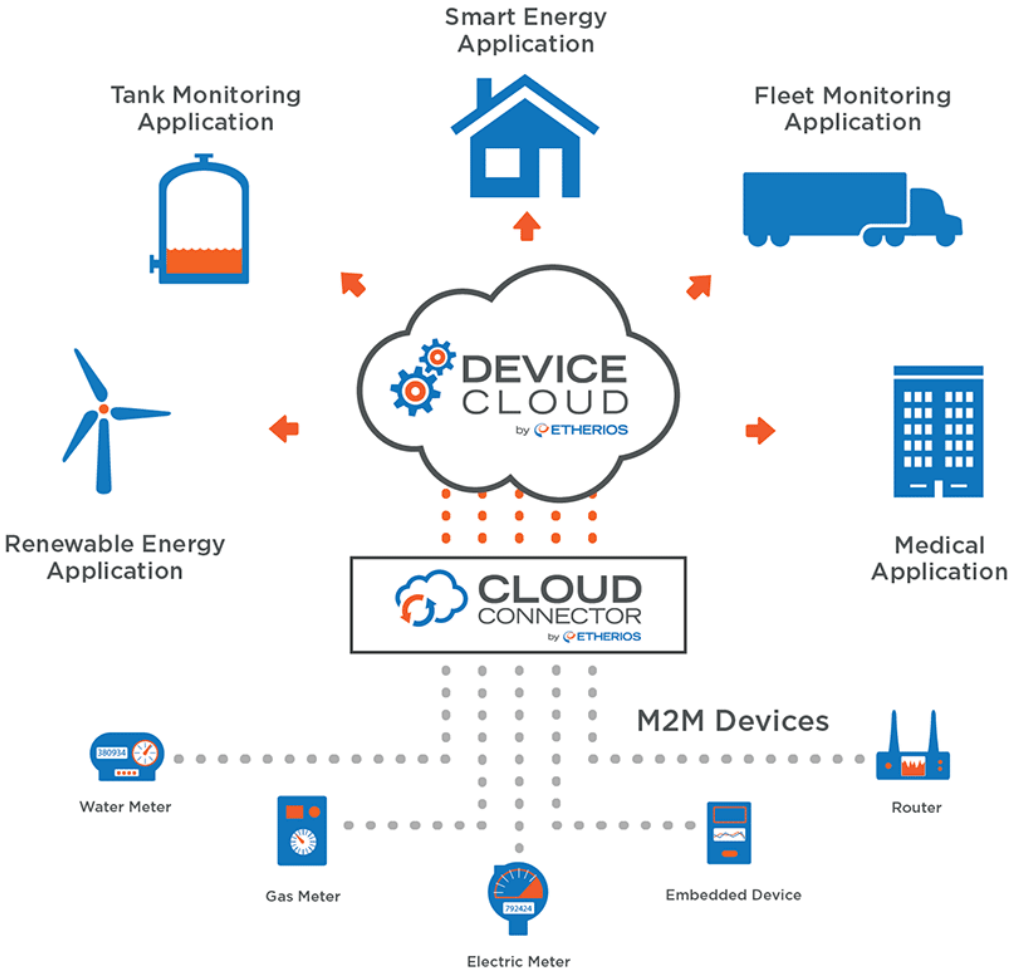
[그림5] AllJoyn Framework Structure

2.2.2 Device Cloud

[그림6] Device Cloud

Device Cloud는 Etherios라는 회사에서 제작하였으며, Cloud Connector로 연결된 모든 디바이스의 데이터를 언제, 어디서나 접근할 수 있도록 하는 Cloud System을 말한다.

Cloud Connector는 Android, Embedded, Kinetics, Java ME, Java SE 플랫폼 등을 지원하며, 각 디바이스의 자원을 관리하는 기능을 제공한다.



[그림7] Device Cloud 개요

2.2.2.1. Device Cloud 특징

- Data Stream : 긴 시간 동안의 data를 순서대로 저장한다. 시간 순서대로 저장된 data는 data point와 data stream의 두 가지 concept를 포함한다.

- SCI(Server Command Interface) : SCI는 User가 그들의 device과 연관된 정보에 접근하거나 명령을 수행할 수 있도록 하는 web service이다.

- SM/UDP(Short Message/User Datagram Protocol) : Device cloud의 이 특징은

Device cloud의 UDP 기반의 SM protocol을 통해 매우 작은 data 공간을 활용할 수 있도록 한다. 이러한 protocol은 현재 SMS와 Iridium Satellite messaging에 사용되고 있다.

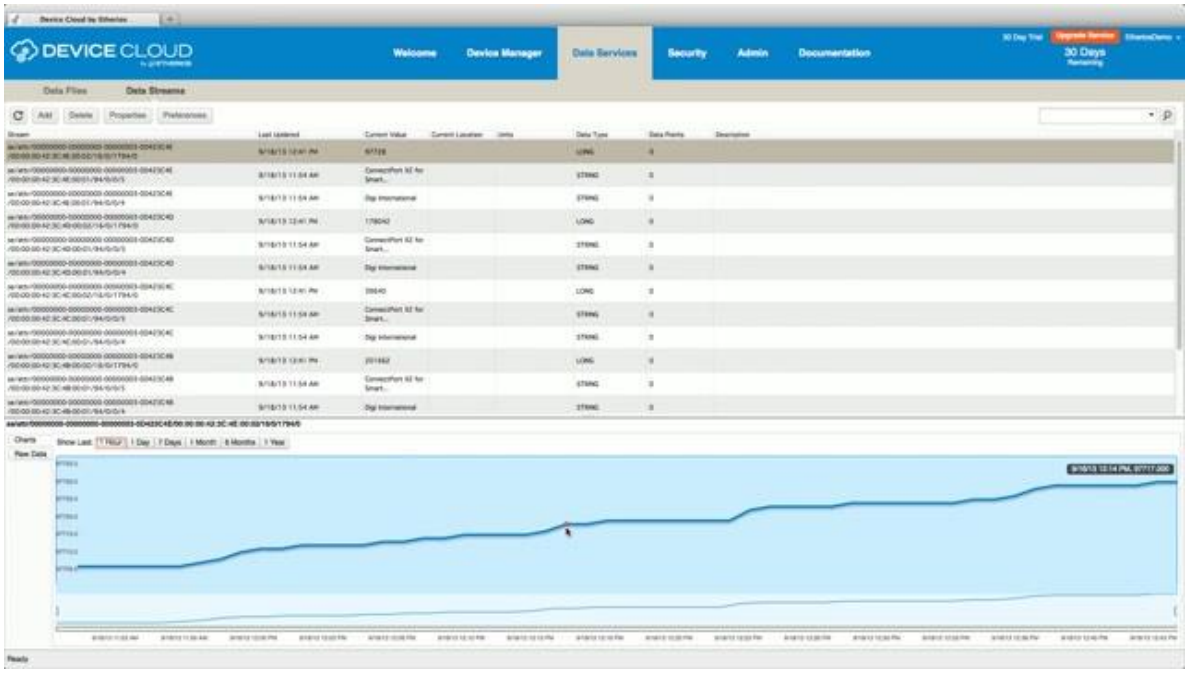
- Monitor(Push Notification) : Device Cloud를 통한 여러 Event에 대하여 User Application이 Asynchronous notification을 등록할 수 있다. 즉, Device Cloud를 통하여 Device에 대한 알림을 받거나 모니터링 용도로 사용할 수 있다.

2.2.2.2. Device Cloud의 장점

- 작은 메모리 용량
- 다양한 디바이스 플랫폼 지원
- Full-Duplex Messaging and Control
- 디바이스 관리 및 Troubleshooting Tool
- 원격 파일 시스템 관리
- Secure Connection

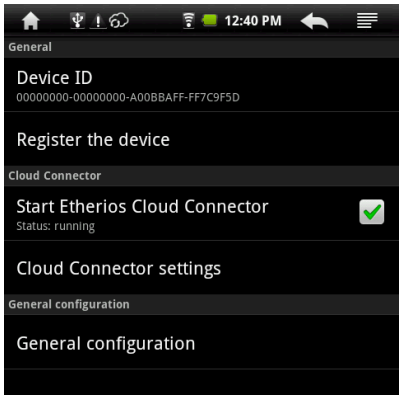
[그림8] Security Service of Device Cloud

Device Cloud는 상용화된 시스템이므로 Security 측면도 상당히 고려되었다. 국제표준인 ISO27002를 비롯하여 각종 수많은 정보보호에 관한 Certification을 통과하였다.



[그림9] Device Cloud Service 화면 - web

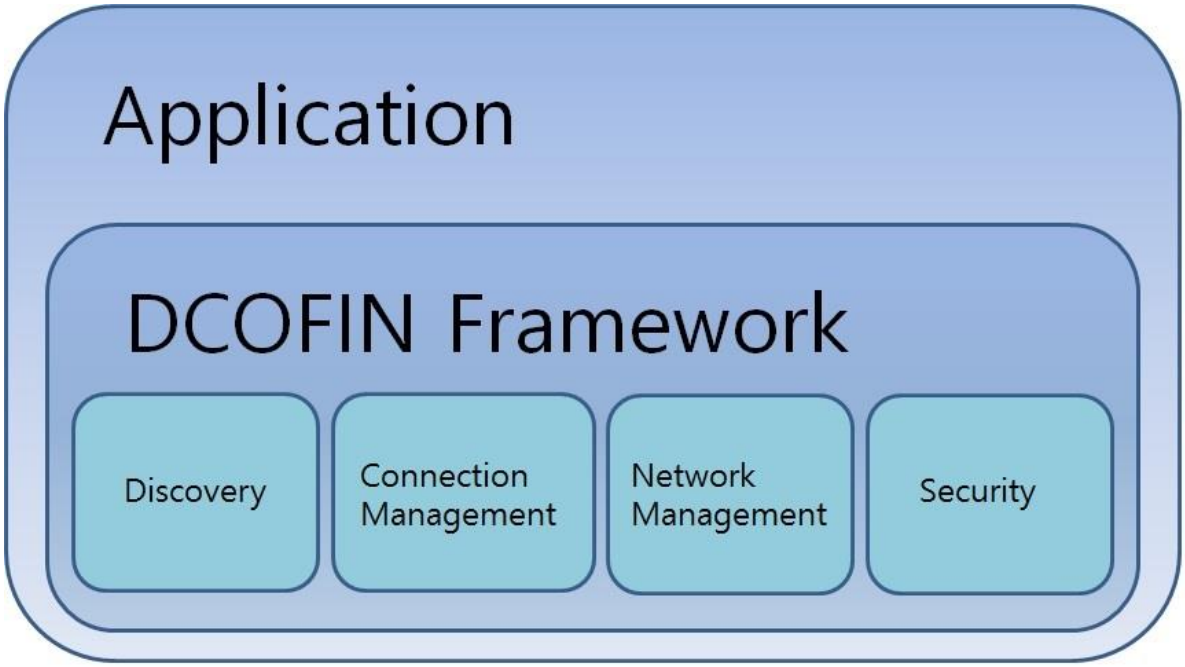
Device Management 기능을 통해 실시간으로 접속된 전체 디바이스를 제어하고, 일간, 월간 디바이스들의 네트워크 정보들을 확인할 수 있다. 추가적으로 예약 Reboot, Firmware Update 기능과 특정 디바이스가 어떤 상태에 도달했을 때 알림을 주는 Alarm 기능 등을 제공한다.



[그림10] Device Cloud Service 화면 - android

3. Wi-Fi 내 Direct 통신 프레임워크(DCOFIN)

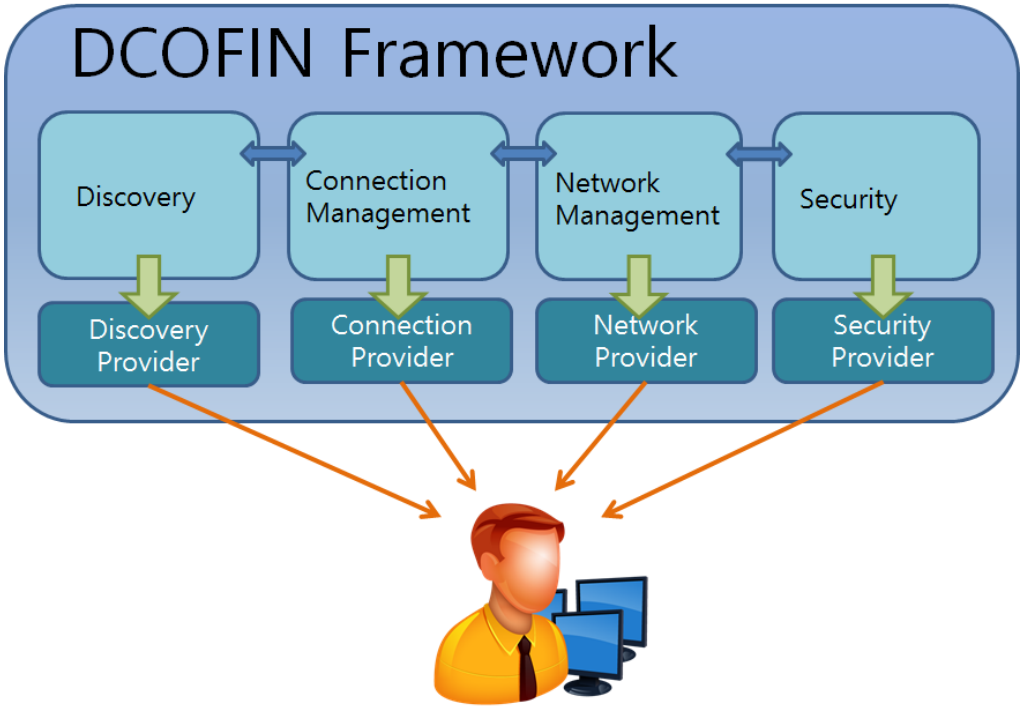
3.1. 전체 프레임워크 구조도



[그림11] DCOFIN 전체 구조도

Direct Communication Framework In wireless Network(DCOFIN)은 크게 네 종류의 모듈들로 구성된다. Discovery, Connection Management, Network Management, Security로 구성되며, 각각의 기능을 담당한다. Discovery module은 Wi-Fi내에 접속했을 시 Daemon의 작동과 정보를 교환하는 기능을 한다. Network Management module은 TCP Socket Communication에 관련된 기능을 담당한다. Connection Manager는 Daemon과 Application과의 연결을 담당한다. Security module은 개체 인증, 암호화 등의 기능을 한다. 각각의 module은 서로 상호작용하며 기능을 수행한다. Programmer는 module내부의 모든 class들을 직접 사용하지 않으며, Provider class를 통하여 보다 더 손쉽게 Framework 기반의 application을 구현할 수 있다.

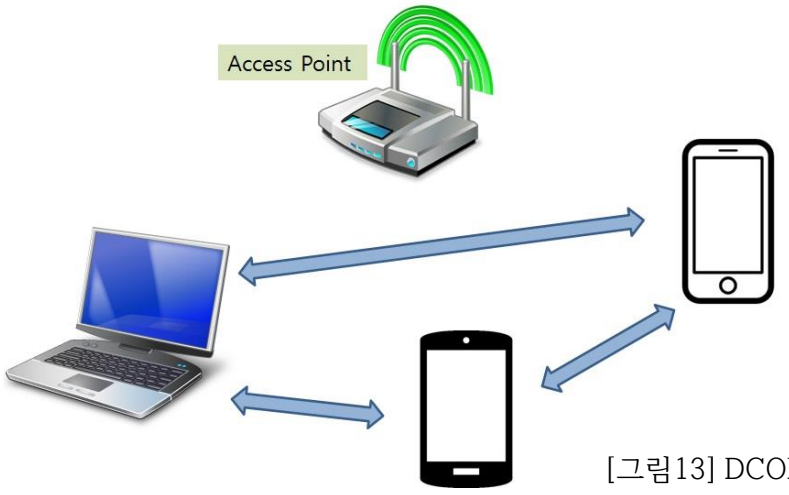
Discovery Provider, Connection Provider, Network Provider, Security Provider는 각 모듈의 기능을 편리하게 제공하기 위한 Wrapper Class이다. Programmer는 Provider Class를 사용하여 DCOFIN Framework 기반의 Application을 손쉽게 제작할 수 있다.



[그림12] DCOFIN Framework 흐름도

내부의 구조를 모두 알 필요 없이, Provider에서 제공하는 기능만으로 핵심 기능을 사용할 수 있게 한다.

3.2. 프레임워크 개요

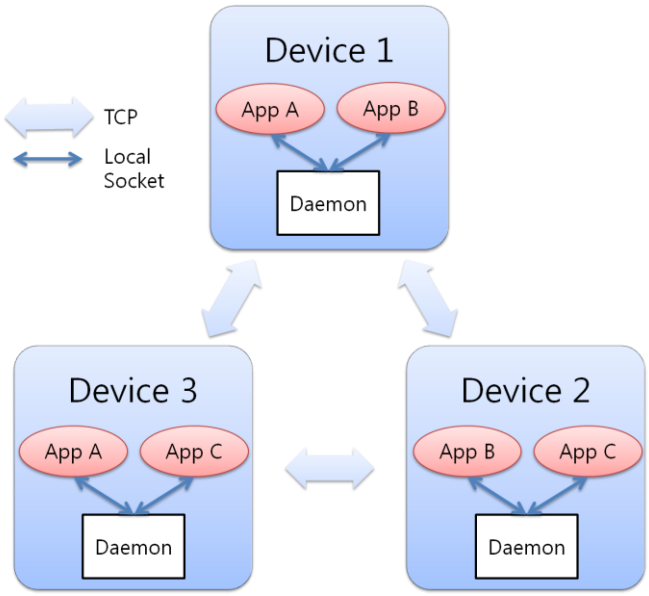


[그림13] DCOFIN 개요

DCOFIN Framework는 동일 Access Point(Wi-Fi 공유기)에 접속한 Device간에 Direct Communication을 가능하게 하는 Framework이다.

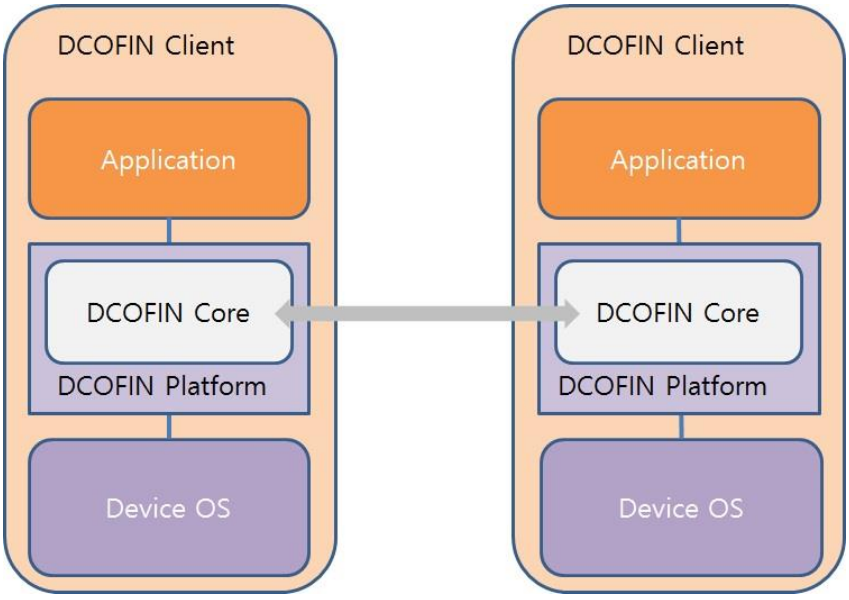
각 디바이스, 즉 Wi-Fi 연결이 가능한 Mobile 기기의 응용프로그램들은 해당 플랫폼의

Daemon과 통신을 하게 되고, Daemon은 응용프로그램이 다른 디바이스와 주고 받는 데이터를 Routing하는 역할을 한다. 각 디바이스는 Wi-Fi에 연결된 상태에서 TCP 통신을 하고 응용프로그램과 Daemon은 Local socket 통신을 한다.



[그림14] Daemon간 통신 구조도

Daemon은 디바이스 내 응용프로그램들이 전송하는 데이터 Packet이 거쳐가는 일종의 Gateway 역할을 하는데, 디바이스 간 Direct 통신을 제공하는 주체이다. 각 디바이스의 Daemon들은 같은 네트워크(Wi-Fi)에 접속된 Daemon들 간의 정보를 주기적으로 교환한다. 해당 Daemon이 네트워크 상에 아직 존재하는지 Health checking을 하는 것이다.

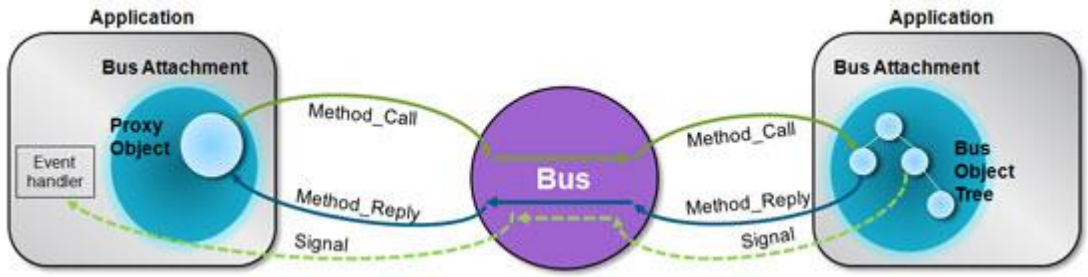


[그림15] Device간 통신 구조도

각 Daemon들은 데이터를 주고 받는 메시지 타입이 4가지로 정의된다.

- Signal (Async.) : 시그널은 Broadcast, Multicast, Point-to-Point 전송에 사용한다.
- Method call (Sync.) : RPC(Remote Procedure Call)를 할 때 사용한다.
- Method reply (Sync.) : RPC의 리턴 메시지를 보낼 때 사용한다.
- Error : RPC에 대한 에러 메시지를 리턴할 때 사용한다.

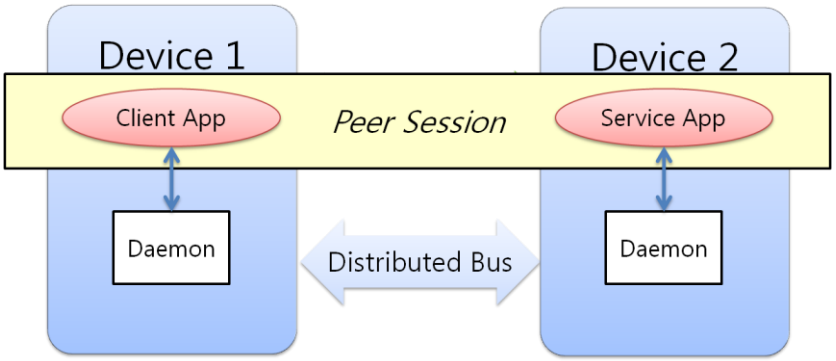
Method call과 reply는 Synchronous 이벤트이므로 response가 올 때까지 Block된다.



[그림16] Daemon간 메시지 전송

3.3. 프레임워크 세부 구조도

3.3.1. Discovery Module

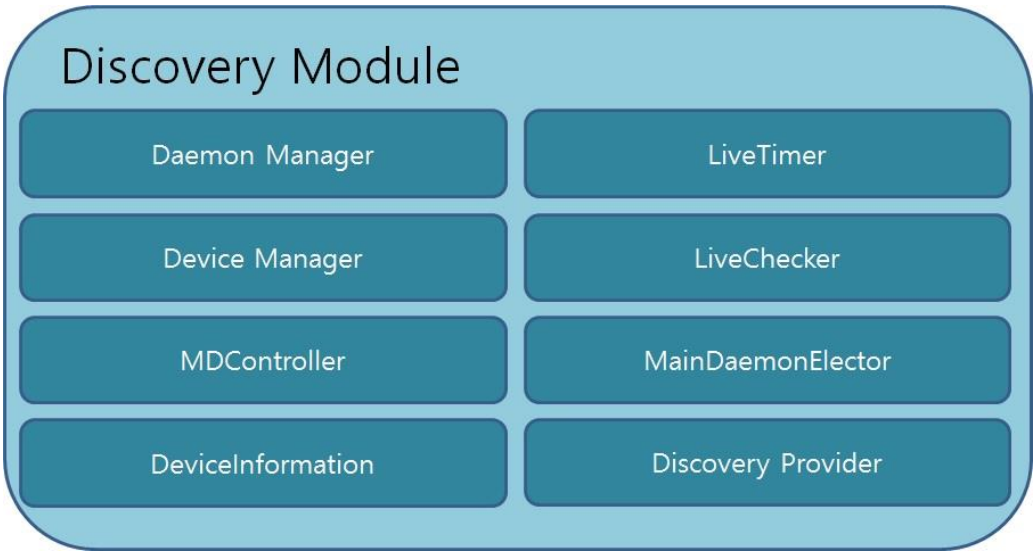


[그림17] Application Peer Session

Discovery Module은 Wireless Network(Access Point)에 접속했을 경우 Daemon을 실행하고 Daemon의 작업에 관한 정보를 교환하는 역할을 한다. Daemon은 background에서 실행되며, 초기화 단계에서 Wi-Fi내의 Device에 Broadcasting을 통하여 접속을 알린다. Broadcasting의 기능은 Network Management module에서 담당한다. 접속을 알리면,

Daemon가운데 대표 역할을 하는 Main Daemon이 새로 접속한 Daemon(Device)과의 인증을 시도한다. 새로운 Daemon의 인증은 Security module에서 담당한다. 만약 Main Daemon이 존재하지 않을 경우(Network에 접속한 첫 번째 Device일 경우) 스스로 Main Daemon이 되며, 그렇지 않을 경우 Main Daemon을 그대로 유지하거나 새롭게 선출한다.

3.3.1.1. Discovery module 구조도



[그림18] Discovery Module 구조도

-DaemonManager class : Daemon의 실행과 Live 상태 체크, Main Daemon역할 수행 여부 등 Daemon의 기능에 관한 전반적인 역할을 담당한다. 위의 기능을 담당하는 Class등의 객체를 생성하여 기능을 수행한다.

-DeviceManager : Main daemon과 나머지 Device들의 정보를 저장하고 관리하는 역할을 수행한다.

-DeviceInformation : Wi-Fi Network에 있는 Device들의 정보를 저장하는 Class이다. 객체 하나당 하나의 Device정보를 저장한다.

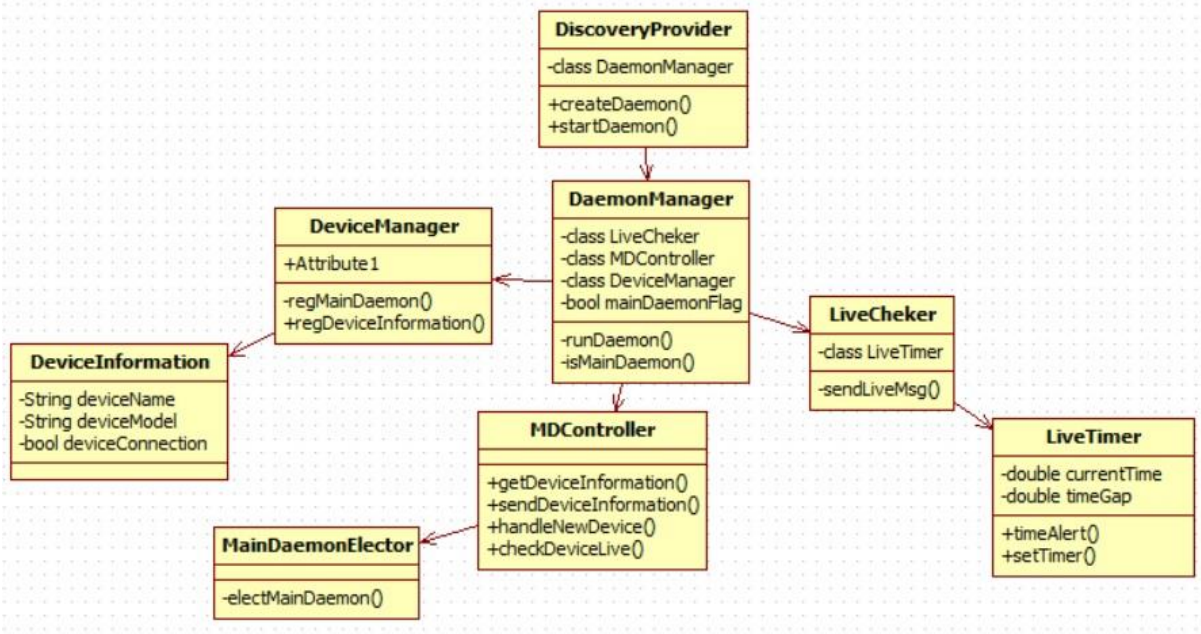
-MDController : Main Daemon으로 선정될 시, Main Daemon의 기능을 수행한다. 새로운 Device가 접속하여 Broadcasting message를 보낼 시 Main Daemon이 답을 하여 통신하는 기능, 새로운 Device를 Authenticate하는 기능, 새로운 Device에게 Network내의 Device정보들을 넘겨주는 기능들을 한다.

-LiveChecker : Network내의 Device들의 접속여부를 확인하기 위하여 주기적으로 Main

Daemon에게 live Message를 보내는 역할을 한다.

-LiveTimer : 주기적으로 Live message를 보내기 위하여 해당하는 시간을 측정하는 역할을 한다.

3.3.1.2. Discovery module Class Diagram



[그림19] Discovery Module Class Diagram

3.3.2. Connection Management Module

Connection Management module은 Application와 Daemon간의 연결을 담당한다. 앞서 보았듯이 Daemon간의 Communication은 TCP Socket을 이용하고 Daemon과 Application간의 Communication은 local socket을 이용한다. Connection Management module은 Daemon과 연결된 Application을 namespace로 관리한다. 즉, 각 Daemon들은 응용프로그램에 대한 namespace를 유지하고 있다가, 세션이 연결되면 각 Daemon들은 namespace를 공유한다. 세션이 유지되는 동안 디바이스 간 통신 상태는 살아있음을 의미한다. Multicast의 경우 세션에 연결된 모든 Peer에게 보낼 수 있다.

3.3.2.1. Connection Management Module 구조도

[그림20] Connection Management Module 구조도

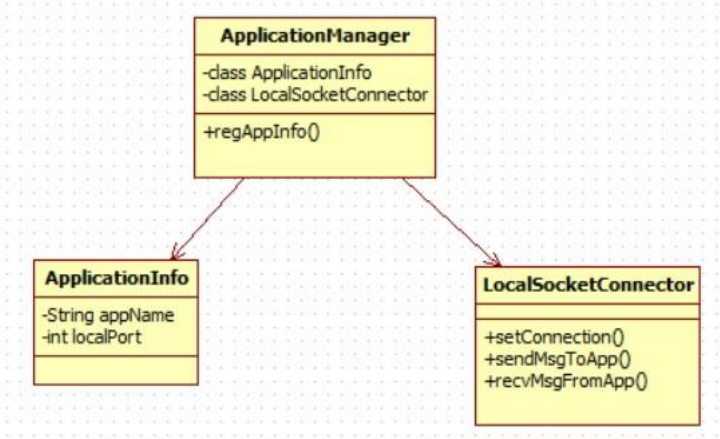
-ApplicationManager : Daemon과 연결되는 Application을 통합적으로 관리한다. Application class와 LocalSocketConnector class의 객체를 생성하여 관리한다.

-ApplicationInfo : 객체를 생성하여 Application 정보를 저장한다. 해당 Application의 namespace와 local에서 사용할 port number를 저장한다. 앞서 보았듯이, device내의 통신은 local socket 통신을 통하여 이루어진다. 따라서 해당 application의 port 번호를 저장한다.

-LocalSocketConnector : Local socket을 통하여 Application과 Daemon간에 연결을 설정하며 Message를 주고받는다.

-ConnectionProvider : 프로그래머에게 손쉽게 사용할 수 있도록 Connection Manager module을 wrapping한다.

3.3.2.2. Connection Management Module Class Diagram

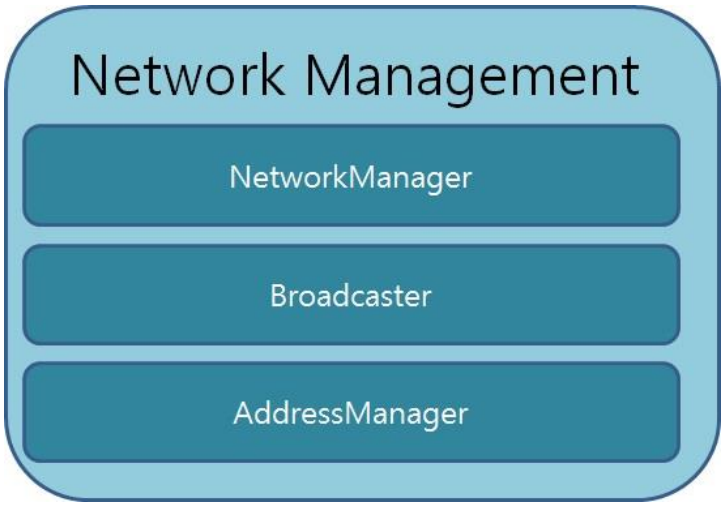


[그림21] Connection Management Module 구조도

3.3.3. Network Management Module

Network Management Module은 TCP Socket Communication과 관련된 전반적인 기능을 담당한다. DCOFIN Framework에서 Daemon들은 TCP Socket을 이용하여 서로 통신한다. 따라서 Wi-Fi내에서 TCP Socket Communication을 담당하는 module이 따로 필요하다. Network Management Module은 Wi-Fi 내에서 Private IP를 이용하여 Device를 구분하고, Message를 Transfer한다. Application Level에서 Private IP를 고려할 필요 없이, DCOFIN Framework를 이용하여 Device의 Private IP에 쉽게 접근이 가능하다. Network Management Module은 Private IP를 통한 Device간의 Communication 기능을 제공하기 위하여 해당 Network의 Gateway Address, Subnet Mask등을 이용하여 Network Address를 계산하는 기능도 제공한다.

3.3.3.1. Network Management Module 구조도



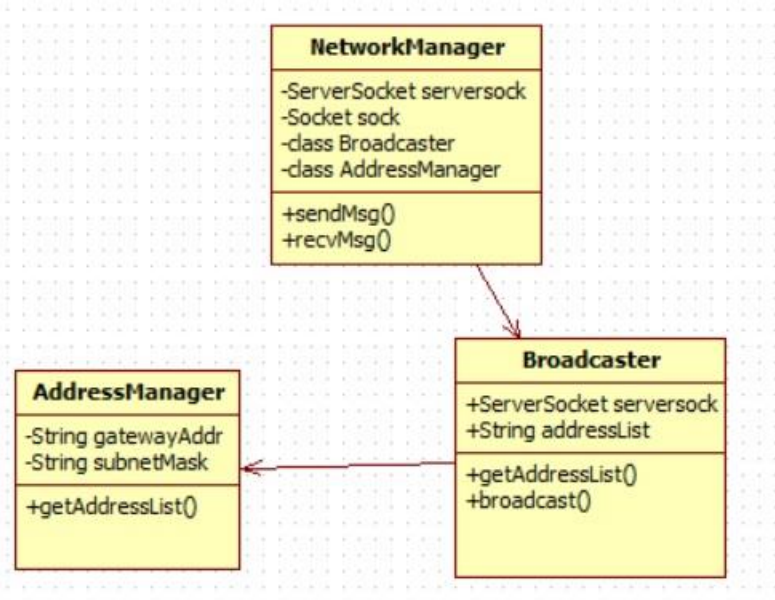
[그림22] Network Management Module 구조도

-NetworkManager : Socket Communication에 관한 전반적인 기능을 담당한다. Server socket을 생성하여 다른 device에서 오는 Message를 수신하며 다른 daemon의 socket address를 통하여 Message를 송신한다.

-Broadcaster : DCOFIN Framework에서는 device간의 존재유무를 서로 판단하기 위하여 Wi-Fi내의 private address를 통하여 Broadcasting을 한다. Broadcaster class는 AddressManager class로부터 private address정보를 받아와 broadcasting한다.

-AddressManager : 해당 Wi-Fi Network의 Gateway address와 subnet mask를 통하여 Network의 private address를 생성한다.

3.3.3.2. Network Management Module Class Diagram



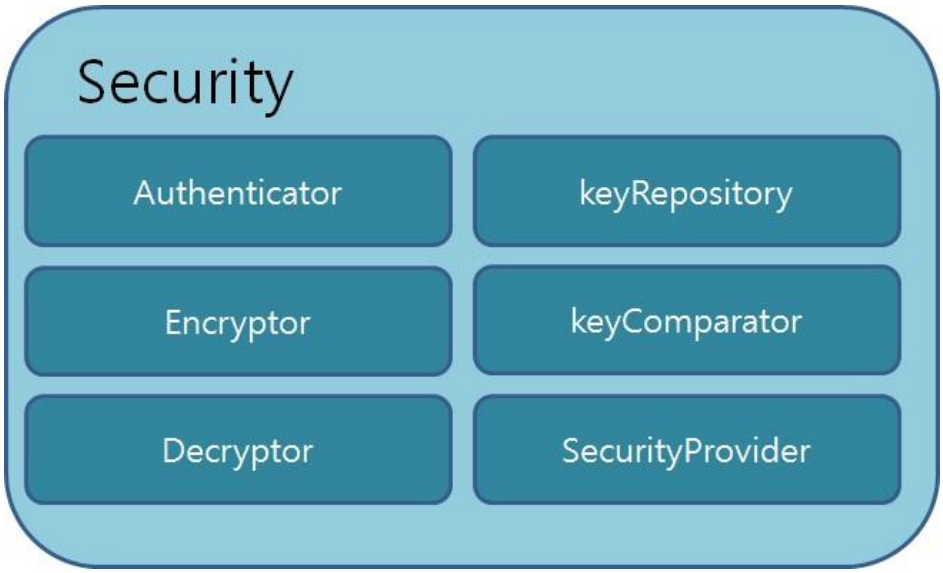
[그림23] Network Management Module Class Diagram

3.3.4. Security Module

Security Module 은 Device 의 Authentication 과 Communication 의 Encryption 에 관련된 전반적인 기능을 담당한다. Wi-Fi Network 내에서의 Authentication 은 Security 측면에서 굉장히 중요한 이슈이다. Daemon 을 통하여 Application 간의 Direct Communication 을 하는 DCOFIN 에서의 Authentication 은 더욱 중요하고 치명적이라 할 수 있다. 따라서 Security module 에서는 기존의 EAP-TLS, EAP-TTLS, EAP-AKA, PEAP 방식 등을 응용하여 안전한 Device Authentication Mechanism 을 개발한다.

Security module에서 제공하는 다른 기능은 Encryption이다. Daemon간에 전송되는 Message들을 암호화함으로써 안전한 정보 전송을 제공한다. 즉, 각 응용프로그램들은 중요한 정보에 대한 암호화 키를 유지하고 있으며 Packet에 대한 필요한 정보를 추출하기 위해 유지하고 있던 키로 인증을 한다. 보안 및 비 보안 Method call 인터페이스를 제공한다.

3.3.4.1. Security Module 구조도



[그림24] Security Module 구조도

-SecutiryProvider : Security module의 기능을 Wrapping하여 programmer에게 제공한다. Authenticator, Decryptor, Encryptor의 객체를 생성하여 기능을 사용한다.

-Authenticator : Wi-Fi network에 새로 접속한 Device의 인증을 담당한다. device 각각의 daemon들은 application마다 다른 key값을 가진다. Authenticator class는 이러한 key값을 KeyRepository class로부터 전달받아 KeyComparator 객체의 기능을 통해 인증을 수행한다.

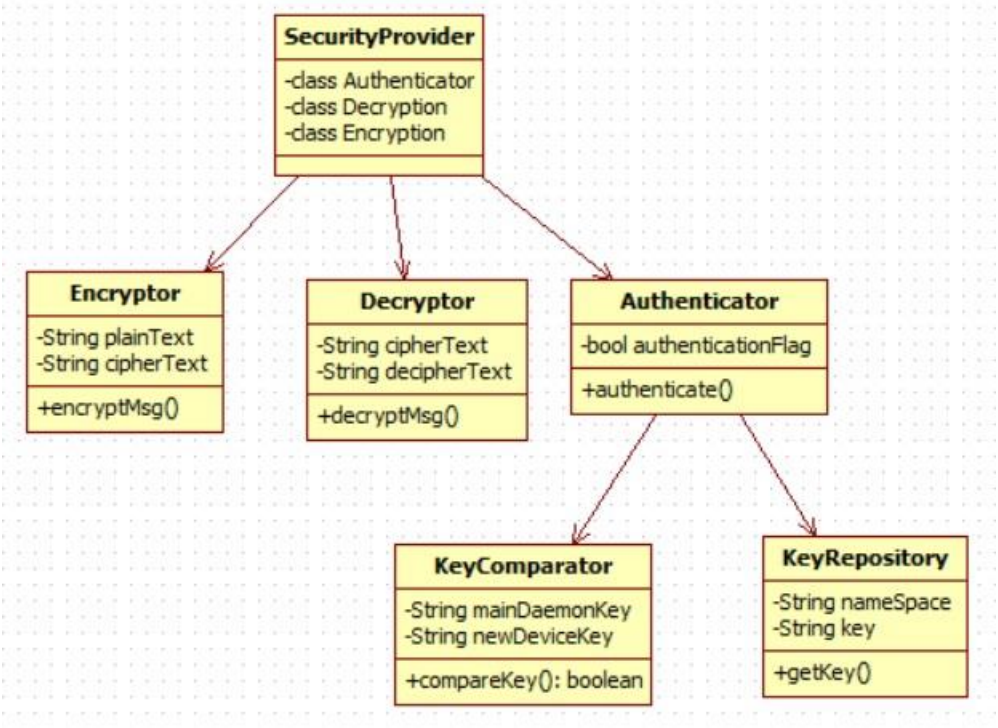
-KeyRepository : Application의 namespace와 key값을 저장하는 class이다. 객체 하나당 하나의 key값 정보를 저장한다.

-KeyComparator : MainDaemon으로부터의 key값과 새로운 Device로부터의 Key값을 비교하여 결과를 return한다.

-Encryptor : plain text를 입력 받아 cipher text로 변환하는 역할을 한다.

-Decryptor : cipher text를 입력 받아 decipher text로 변환한다.

3.3.4.2. Security Module Class Diagram



[그림25] Security Module Class Diagram

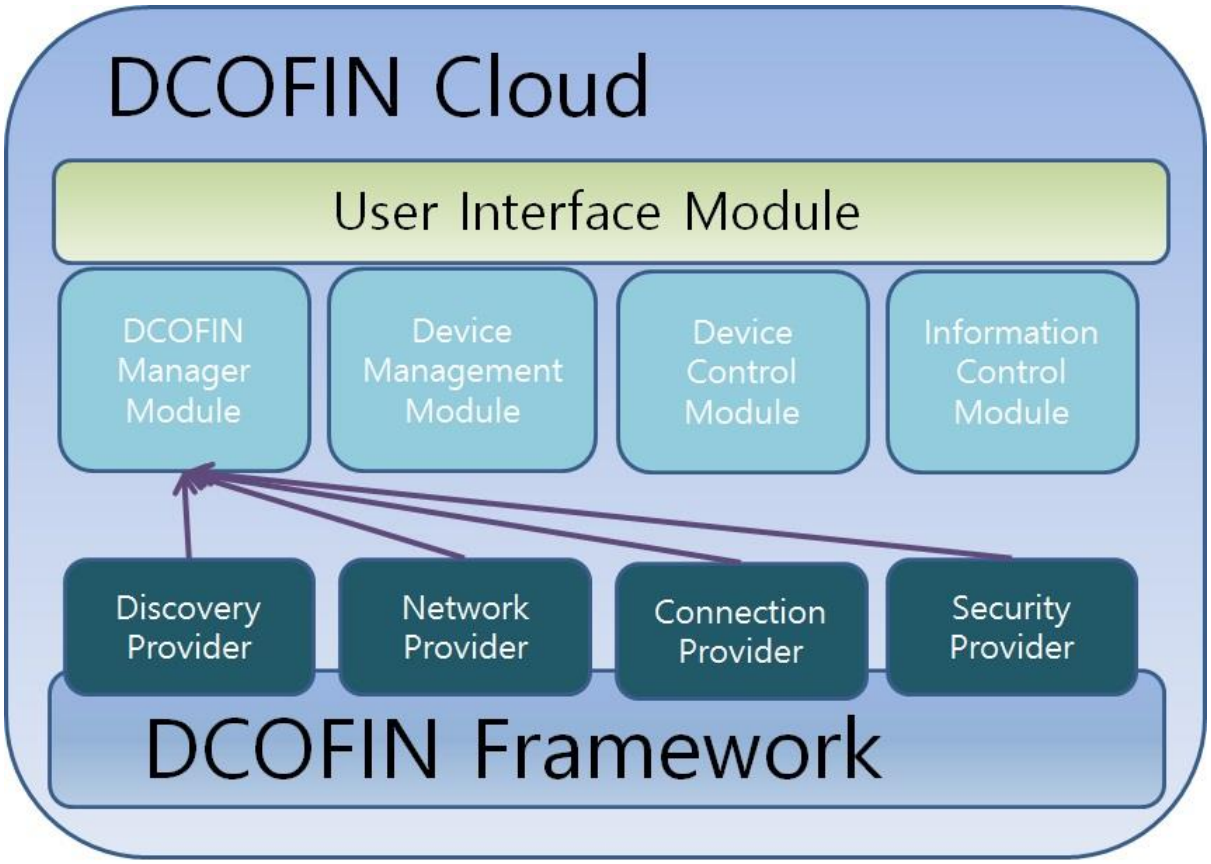
4. DCOFIN Cloud

[그림26] DCOFIN Cloud Logo

-DCOFIN Cloud는 DCOFIN Framework를 기반으로 개발하는 통합 Device 관리 system 이다.

-기존의 Device 관리 system은 최소 하나 이상의 server가 필요한 반면에, DCOFIN Cloud는 DCOFIN Framework를 사용하여 Network에 접속한 어떤 Device던지 다른 Device를 관리할 수 있도록 한다.

4.1. Application 전체 구조도



[그림27] DCOFIN Cloud 전체 구조도

-DCOFIN Cloud는 다양한 Platform상에서의 User Interface를 제공한다. Windows(java) 기반과 Android기반의 UI를 구현한다.

-DCOFIN Framework를 통한 Application 구현으로 Wi-Fi내의 Device의 정보를 direct로 전송 받는다.

-기존 device cloud와의 차이점은 서버가 따로 존재하는 것이 아닌 어떠한 device에서든지 직접 다른 device의 정보를 direct로 전송 받고, direct로 제어할 수 있다는 것이다

5. 장점

5.1. 편의성

원격제어, 파일전송 등 제공하는 기능을 통해 사용자의 편의성을 높여준다.

5.2. 범용성

운영체제(OS)나 제조사에 구애 받지 않고 쉽게 사용 할 수 있다.

5.3. 재사용성

프레임워크를 재사용 할 수 있어, 설계 비용이 절감 된다.

5.3. 활용성

Wi-Fi Network를 사용하는 다양한 Application에 활용이 가능하다.

6. 기대 효과

6.1. DCOFIN Framework

- DCOFIN Framework를 사용하여 Application을 구현함으로써 Wi-Fi내의 Device간의 보다 더 효율적인 Communication이 가능하다.

- Application Level에서 Device의 Network Address에 관하여 직접 관여하지 않아도 되므로 프로그래머에게 편의성을 제공한다.

- Wi-Fi 내에서 Socket Communication을 사용하는 Application(예를 들면 원격제어, 파일 전송)의 개발에 유용한 Tool이 될 수 있다.

- windows, android 등 여러 플랫폼에서 사용할 수 있는 framework를 제공하여 개발자와 사용자에게 편의성을 제공한다.

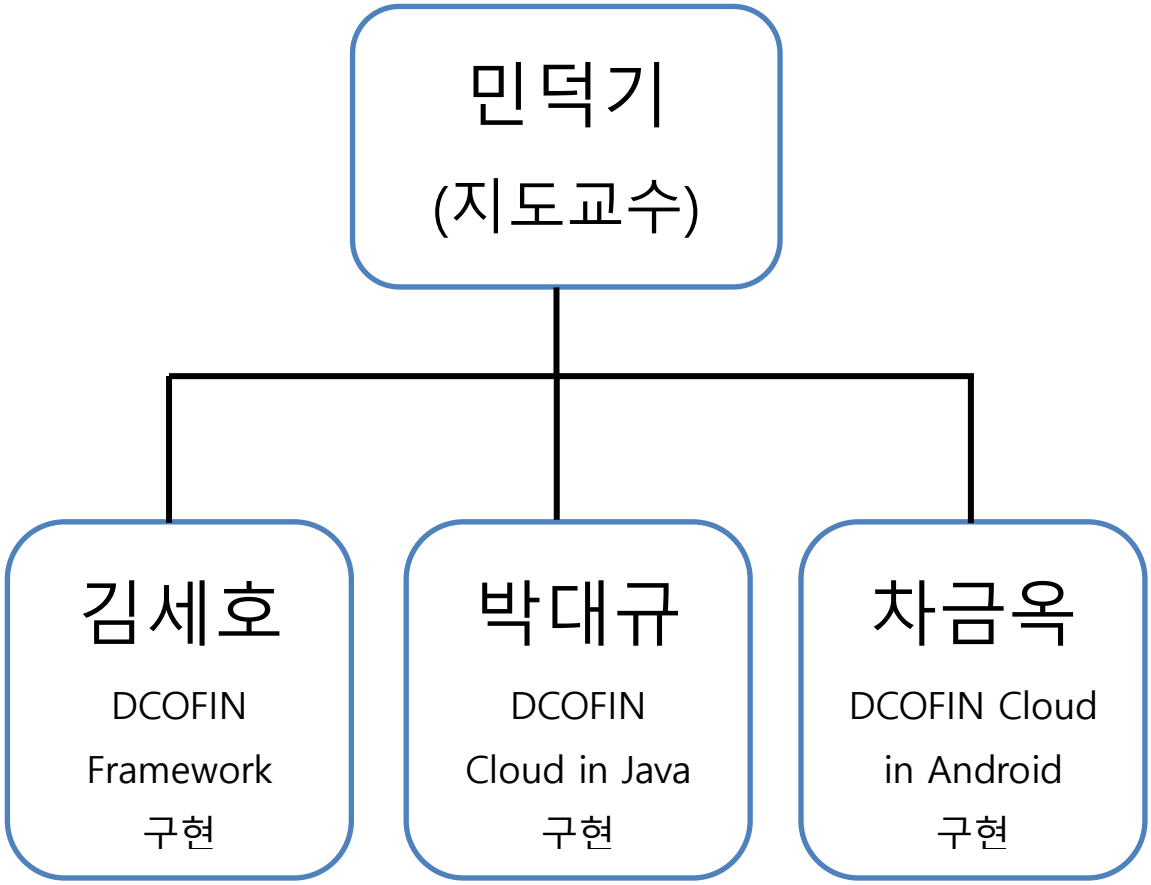
6.2. DCOFIN Cloud

- DCOFIN Cloud를 통하여 Wi-Fi Network에 접속한 Device를 통합적으로 관리하여 사용자에게 손쉽게 IoT Service를 제공한다.

- Device Information을 관리하며 원격제어 등의 서비스를 제공, 특정 Device가 아닌 Wi-Fi Network에 접속한 임의의 Wireless Device를 통하여 Network 내의 Device를 관리할 수 있게 한다.

-Wi-Fi Network 내의 Device 관리 기능을 통해 Network 보안을 강화할 수 있다.

7. 역할 분담



[그림28] 역할 분담

7. 개발 일정

일련 번호	주요 내용	추진 일정												소요 기간
		4		5		6		7		8		9		10		
1	자료조사 및 분석															6주
2	제안서 작성															6주
3	프레임워크 구조 설계															10주
4	DCOFIN Framework 개발															8주
5	DCOFIN Cloud 개발															8주
6	모듈 통합 및 통합 테스트															4주
7	프레임워크 활용성 검증															4주
8	최종 테스트 및 보고서 작성															4주