

Open API 개발 기술 현황

(State of Art on Open API Development)

김태영, 권동우, 김현우, 주홍택

계명대학교 컴퓨터공학과

{ty.kim, dwkwon, hwkim84, juht}@kmu.ac.kr

요 약

사물인터넷은 최근 IT 분야에서 가장 각광받는 기술이며, 이를 이용하여 다양한 사물을 연계한 매쉬업 서비스의 중요성이 점차 증가하고 있다. 다양한 사물을 연계하여 기존에 존재하지 않던 새로운 서비스를 가능케 하는 매쉬업 서비스의 기반 기술 중 하나가 Open API 이다. Open API 는 인터넷 상에서 서비스, 정보, 데이터를 외부 개발자 혹은 사용자가 사용할 수 있도록 공개한 것으로 본 논문에서는 매쉬업의 기반 기술인 Open API 를 개발하기 위해 Open API 구성 기술에 대해 설명한다. 뿐만 아니라 Open API 개발 기술 현황에 대해서도 분석하고, Open API 를 개발하기 위한 방법에 대해 소개한다.

Keywords: Open API 개발 방법, Open API 구성 기술, Open API development, RESTful, SoAP, XML, JSON

1. 서론

최근 인터넷의 발달과 다양한 스마트 디바이스의 등장으로 관리자의 개입 없이 스마트 디바이스 간에 통신을 수행하는 사물인터넷 시대가 도래하였다[1]. 그에 따라 인터넷과 스마트 디바이스 간 통신을 기반으로 기존에 존재하지 않았던 새로운 융합 매쉬업 서비스가 출현하였다. 이러한 매쉬업 서비스는 스마트 디바이스를 운영하는 플랫폼에 독립적으로 동작해야 하며, 데이터 전송 시 네트워크 부하가 적고, 다양한 형태의 서비스를 요구하는 사용자들을 포용해야 한다.

Open API(Application Programming Interface)는 인터넷 상의 서비스, 정보, 데이터를 외부의 사용자 및 개발자가 사용할 수 있도록 개방해 놓은 API 이다. Open API 는 다양한 개발자들이 사용 가능한 형태로 배포되며, 네트워크 부하가 낮고, HTTP 프로토콜 상위에서 동작하기 때문에 플랫폼에 독립적인 특징을 지닌다. 이러한 특징들로 인해 Open API 는 다양한 매쉬업 서비스의 구현에 핵심 기반이 된다.

Open API 는 개별적으로 매우 단일 기능을 수행하지만 여러 Open API 를 조합하여 복잡하고 다양한 매쉬업 서비스의 제공이 가능하다. 그 예로 현재 미국, EU, 뉴질랜드, 한국 등 다양한 국가에서 행정, 재난, 교육 등 다양한 분야의 공공 데이터를 간단한 Open API 로 제공하고 Open API 를 통해 복잡한 매쉬업 서비스를 개발하도록 장려하고 있다[2]-[4]. 뿐만 아니라 Google, Yahoo, amazon, Facebook 등 글로벌 기업에서도 다양한 기능을 수행하는 Open API 를 배포하여 새로운 매쉬업 서비스 개발을 독려하고 있다. [5]-[8].

새로운 매쉬업 서비스의 개발에 Open API 를 사용하면 개발자가 복잡한 기능을 구현하지 않고 Open API 를 통해 그 결과만을 활용하기 때문에 간편하게 서비스 개발이 가능하며, 그로 인해 서비스 개발 시간을 단축할 수 있는 장점이 있다. Open API 를 이용하여 새로운 매쉬업 서비스를 개발할 때 고려해야 할 점은 첫째, 현재 배포하고 있는 Open API 를 용도에 맞게 어떻게 선택할 것인가와 둘째, 적절한 Open API 가 없을 시 어떻게 Open API 를 개발할 것인가에 있다.

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신-방송 연구개발사업의 일환으로 수행하였음. [R0126-15-1009, ICBMS 플랫폼 간 정보 모델 연동 및 서비스 매쉬업을 위한 스마트 중재 기술 개발]

논문번호 : KNOM2015-01-04, Received April 27, 2015; Revised June 12, 2015; Accepted August 9, 2015

Open API 를 용도에 맞게 선택하려면 현재 배포중인 Open API 를 전수 조사하고 분석하는 절차가 필요하지만, 사실상 Open API 의 전수 조사, 분석은 시간적으로 불가능하다. 이에 본 논문은 Open API 를 조사 및 분석하는 절차는 고려하지 않고 있으며, 용도에 맞는 Open API 가 없을 시 어떻게 Open API 를 개발 할 것인가에 초점을 두고 있다.

본 논문에서는 효과적인 Open API 개발을 위해 목적에 맞는 Open API 의 서비스 형태, 데이터 타입, Open API 사용을 위한 인증 등 Open API 구성 기술을 살펴보고, 살펴 본 기술들을 비교 분석하였다. 그리고 Open API 개발 동향을 분석하고, Open API 를 효과적으로 개발 하기 위해 방법과 Open API 개발 도구를 조사하였으며, 이를 비교 분석하고 그 결과를 제시한다.

본 논문은 다음과 같은 순서로 기술한다. 2 장에서는 다양한 Open API 구성기술에 대해서 설명하고, 3 장에서는 Open API 를 개발하기 위한 기술 현황에 대해 살펴본다. 마지막으로 4 장에서는 결론 및 향후 연구에 대해 논의한다.

2. Open API 구성 기술

이 장에서는 Open API 개발 시 결정해야 하는 Open API 구성 기술에 대해 소개한다. 특히 웹 서비스를 위한 Open API 접근 방식을 살펴보고, Open API 의 동작 수행 시 어떤 데이터를 송수신 하는지에 대한 Open API 데이터 형식, Open API 인증에 대해서 각각 알아본다. 실제 Open API 개발 시에는 앞으로 살펴볼 웹 서비스 방식과 데이터 형식 각각의 특징을 고려하여 Open API 를 설계해야 하며, 다양한 Open API 사용자를 위한 인증까지도 고려 대상으로 포함시켜야 한다.

2.1 Open API 웹 서비스 방식

Open API 는 일반적으로 웹 서비스 형태로 배포되며 대표적인 Open API 웹 서비스 방식으로 SoAP(Simple Object Access Protocol), JSON(JavaScript Object notation)-RPC(Remote Procedure Call), XML(Extensible Markup Language)-RPC, RESTful(Representational State Transfer), BPEL(Business Process Execution Language) 등이 있다. 이 절에서는 가장 많이 사용되고 있는 SoAP 와 RESTful 웹 서비스 방식에 대해서 알아본다.

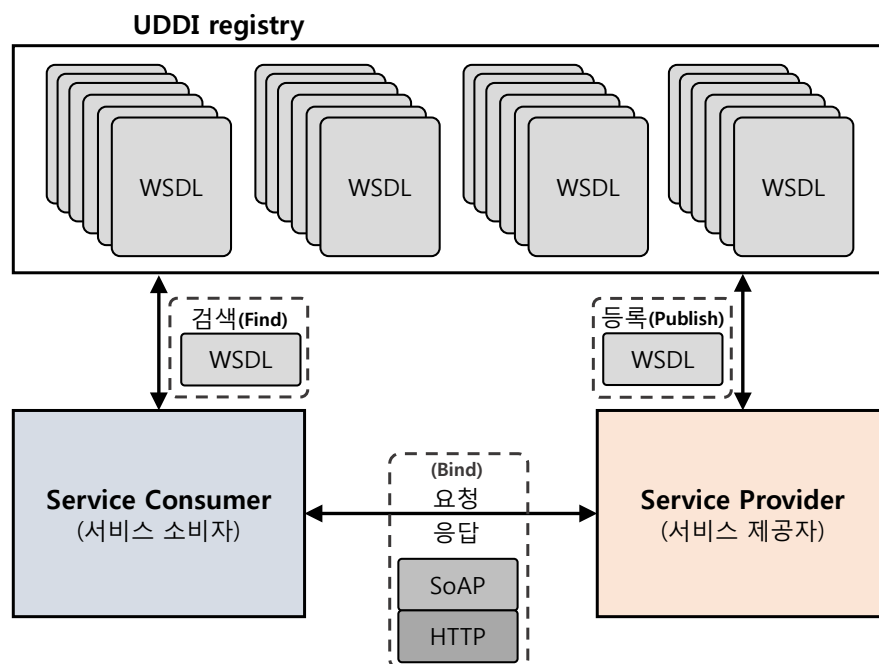


그림 1. SoAP 개념도

먼저 분산 컴퓨팅 환경에서 사용하기 위해 설계 된 SoAP[9]를 살펴보면 HTTP, HTTPS, SMTP 를 이용하여 XML 기반의 메시지를 교환하는 프로토콜로써 방화벽의 우회가능하고, 프로그래

밍 언어 및 디바이스 운영 플랫폼에 독립적이다. 그림 1 은 SoAP 의 동작을 설명한다. 웹 서비스 제공자는 UDDI(Universal Description Discovery and Integration) 레지스트리에 웹 서비스를 WSDL(Web Services Description Language) 형태로 등록한다. 이후 웹 서비스 소비자는 UDDI 레지스트리에서 사용하고자 하는 WSDL 형태의 웹 서비스를 검색하고 웹 서비스 제공자에 서비스를 요청하고, 요청에 대한 응답을 수신한다. 웹 서비스 제공자와 웹 서비스 소비자의 요청 및 응답 절차에는 SoAP 를 이용하여 이루어진다[10].

이에 반해 RESTful 은 웹의 장점을 활용하기 위한 네트워크 기반의 아키텍처로 Roy Fielding 에 의해 정의 되었다[11]. RESTful 은 웹 서비스 개발자나 사용자에게 편리하게 데이터를 제공하는데 있어 매우 간편한 방법이며 HTTP 를 통해 JSON 이나 XML 같은 데이터의 전송을 수행하기 때문에 SoAP 와 마찬가지로 방화벽의 우회나 프로그래밍 언어 및 디바이스 운영 플랫폼에 독립적이다. RESTful 의 동작은 그림 2 와 같다. 웹 서비스 소비자는 배포된 URI 를 이용하여 직접 웹 서비스 제공자에 서비스 요청 및 응답을 수행한다.

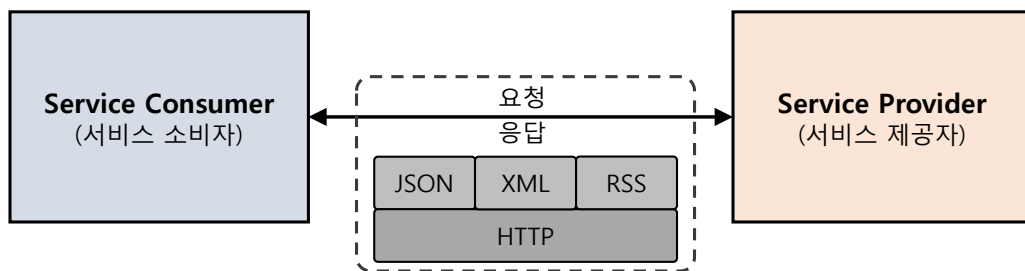


그림 2. RESTful 개념도

표 1 은 SoAP 와 RESTful 을 비교하고 요약한 결과이다. SoAP 와 RESTful 의 대표적인 차이점은 표준의 유무와 동작의 간결성이다. SoAP 는 잘 완성된 표준에 따라 Open API 설계와 보안, 에러 처리 등 다양한 기능을 제공한다. 하지만 복잡한 표준에 따라 Open API 를 개발해야 하기 때문에 RESTful 에 비해 구현에 어려움이 있으며, 동작에 많은 오버헤드를 지닌다는 단점이 있다. RESTful 은 표준의 부재로 Open API 설계에 많은 어려움이 따르지만, 구현의 용이성과 빠른 동작이라는 장점을 지닌다.

표 1. SoAP 와 RESTful 기술 비교

| | | SoAP | RESTful |
|-------|---------|--|---|
| 특징 | | <ul style="list-style-type: none"> •분산 컴퓨팅 환경을 가정하여 설계 •Stateful 기반 •POST 방식 사용 | <ul style="list-style-type: none"> •P2P 통신 환경을 가정하여 설계 •Stateless 기반 •GET, POST, PUT, DELETE 방식 사용 |
| 강점 | | <ul style="list-style-type: none"> •표준 스펙이 존재 <ul style="list-style-type: none"> : Open API 설계 시 RESTful 에 비해 유리 : 보안 및 에러 처리 가능 | <ul style="list-style-type: none"> •SoAP 에 비해 간결함 <ul style="list-style-type: none"> : 동작상 구조가 간단함 •개발이 용이함 <ul style="list-style-type: none"> : 추가적인 메시지 처리 없음 |
| 약점 | | <ul style="list-style-type: none"> •개발이 어려움 <ul style="list-style-type: none"> : 표준 스펙에 따라 구현 : 복잡한 구조에 따른 오버헤드 존재 •개발을 도와주는 추가적인 도구 필요 | <ul style="list-style-type: none"> •표준의 부재 <ul style="list-style-type: none"> : 개발 시 자체 표준으로 설계 진행 : 보안 및 에러 처리 절차 없음 •HTTP 통신 모델에 의존 •P2P 이외에 환경에서는 유용하지 않음 |
| 기반 기술 | 서비스 명세 | WSDL | JSON, XML, RSS 등 |
| | 레지스트리 | UDDI | X |
| | 기반 프로토콜 | HTTP, HTTPS, SMTP | HTTP |

2.2 Open API 데이터 형식

Open API 개발 시 목적, 용도에 따라 적합한 데이터 형식을 선택해야 하며, 이 절에서는 Open API 개발 시에 적합한 데이터 형식의 선택을 위해서 현재 가장 많이 사용되고 있는 Open API 데이터 형식에 대해서 소개한다. Open API 에서 사용하는 데이터 형식으로는 XML, JSON, RSS 등이 있으며 현재 가장 널리 사용된 XML 과 최근에 가장 많이 사용되고 있는 JSON 에 대해서 살펴본다.

먼저 XML[12]은 W3C(World Wide Web Consortium)에서 제안된 언어로 가장 널리 사용된 데이터 형식이다. XML 은 태그 구조를 사용하기 때문에 작성하기 간편하며 태그 정보로 인해 사람이 이해하기가 편리하고, DTD(Document type Declaration)[13]를 정의하여 다양한 방식으로 확장이 가능하다. 하지만 태그 구조로 인해 반복 데이터 삽입 시 불필요한 데이터가 계속 나타나며, 이로 인해 데이터의 파싱 시간이 길어지는 단점을 가진다.

JSON[14]은 최근 가장 많이 사용되고 있으며, 경량의 데이터 형식으로 값에 대한 표현을 키와 값 쌍으로 하여 데이터 내용을 함축적으로 저장한다. JSON 에서 채택한 데이터 구조는 C, C++, C#, Java, Perl, Python 등 다양한 프로그래밍 언어에서도 지원하고 있으며, 효율적인 데이터 구성이 가능하고, 데이터의 함축성으로 인해 컴퓨터가 이해하기 쉽고 데이터 추출 시간이 빠르다는 장점이 있다. 그림 3 은 XML 과 JSON 의 파일 구조의 예를 보여준다.

일반적으로 JSON 은 XML 보다 데이터를 함축적으로 저장하기 때문에 동일 내용을 인코딩 했을 때, JSON 으로 인코딩 된 데이터가 XML 로 인코딩 된 데이터보다 더 적은 용량을 필요로 하며 이에 따라 전송 속도역시 JSON 이 더 빠르다[15]. 또한 전송 데이터 항목 개수 변화에 따라 XML 의 전송 시간은 큰 차이를 보이는 반면, JSON 은 전송하는 항목의 개수에 따른 전송 시간의 변화는 거의 일어나지 않는다. 다만, JSON 으로 인코딩 된 데이터의 경우, XML 로 인코딩 된 데이터를 처리 할 때보다 더 많은 CPU 자원이 필요하다.

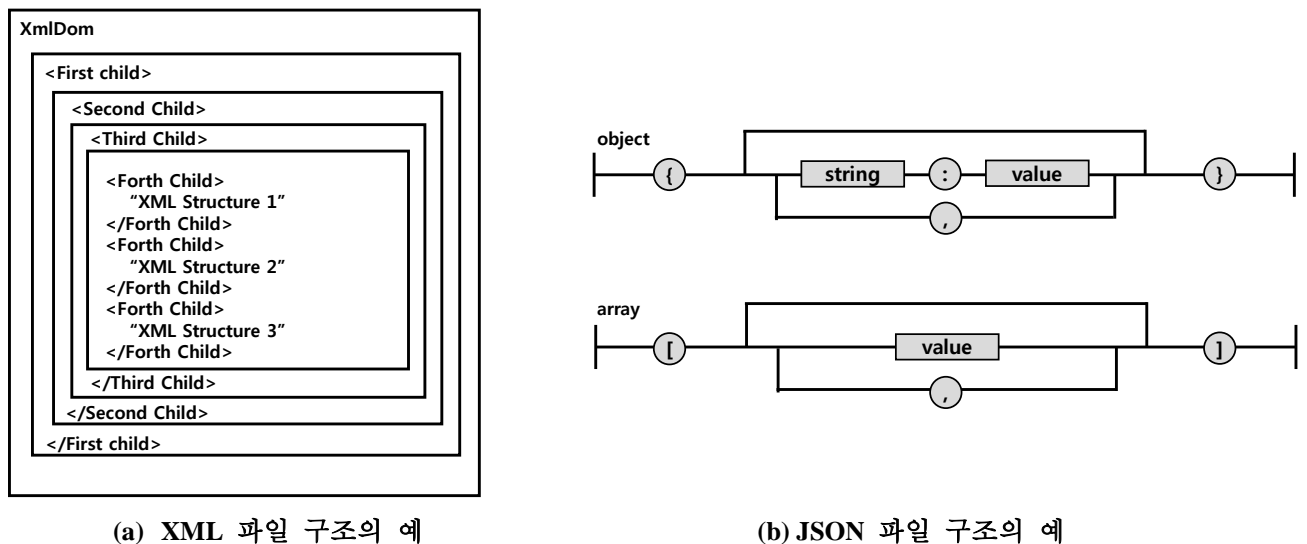


그림 3. XML, JSON 파일 구조의 예

2.3 Open API OAuth 인증

OAuth 는 기존 Google 의 AuthSub, Yahoo 의 BBAuth 등과 같은 다양한 인증 방식을 통합하여 표준화한 인증 방식으로 현재 Open API 인증에는 대부분 OAuth 가 사용된다. OAuth 는 보안상 취약한 아이디와 패스워드 기반의 인증체계를 개선한 방법으로 아이디와 패스워드 없이 인증을 수행하는 방법이다. OAuth 는 인증과 권한의 역할 또한 수행하는데 OAuth 인증에는 Request_Token 이 사용되며, 권한에는 Access_Token 이 사용된다. OAuth 는 1.0 을 시작으로 OAuth 1.0 보안 결함을 개선 한 OAuth 1.0a 이 있으며, 최근에 OAuth 2.0 이 발표되었다[16],[17].

OAuth 2.0 은 OAuth 1.0a 을 개선한 인증처럼 보이지만 상호 호환이 되지 않고, 실제로는 사용된 용어부터 규격까지 차이가 많다. OAuth 2.0 은 OAuth 1.0a 보다 다양한 애플리케이션의 인증을

제공하며, HTTPS 를 사용하여 Signature 를 간소화하고, Access-Token 의 유효기간 기능을 제공한다. 표 2 는 OAuth 1.0a 와 OAuth 2.0 을 비교한 표이다.

표 2. OAuth 1.0a 와 OAuth 2.0 비교 분석표

| | OAuth 1.0a | OAuth 2.0 |
|-------------|--|-----------------------------------|
| 목적 | •웹 애플리케이션의 인증 | •웹 애플리케이션 외 다양한 애플리케이션 인증 |
| 토큰 요청방법 | •HMAC 암호화 인증방식 : Signature 생성 | •HTTPS 사용 : Signature 간소화 |
| Token 인증 기간 | •인증 유효기간이 없음 : 자체 구현 | •인증 유효기간 설정 : Refresh Token 사용 |
| 기타 | •OAuth 1.0a 에서 발생한 인증과 API 사용 이슈로 인해 OAuth 2.0 에서는 API 서버와 인증 서버를 분리하는 규격 정의 | |

3. Open API 개발 기술 현황

이 장에서는 효과적으로 Open API 를 개발하기 위한 방법들에 대해서 살펴 본다. 먼저 Open API 의 일반적인 개발 방법에 대해서 알아 보고, 효과적인 Open API 의 개발을 도와주는 Open API 개발 도구들에 대해서도 알아본다.

3.1 Open API 개발 기술

Open API 개발 기술에는 Open API 개발과 Open API 배포가 포함되며, Open API 개발자는 두 가지 과정을 모두 수행해야 한다. 먼저 Open API 개발에는 다양한 방법이 존재한다. Open API 개발 방법에 영향을 미치는 요소로는 개발자의 역량이나 개발 성향, Open API 의 사용 목적, 용도에 따라 다양하게 나뉠 수가 있다. 그래서 이 절에서는 효과적으로 Open API 를 개발 하기 위한 정형화된 Open API 개발 방법론을 설명한다.

Open API 개발은 크게 5 단계로 나눌 수 있다[18]. Open API 개발을 위한 가장 첫 번째 단계는 Open API 의 목적이나 용도에 맞도록 Open API 요구사항을 정의 하는 것이다. Open API 정의 시의 고려 요소는 Open API 사용형태, Open API 사용 패턴, 규제사항이다. 쉽게 말해 모바일 앱 용 API, 웹 용 API 등이 Open API 사용형태를 말하며, 사용 권한 있는 사용자가 Open API 를 사용하게 할지 아니면 사용 권한이 없는 사용자도 사용이 가능하게 할지는 사용패턴을 의미한다. 또한, SoAP 나 RESTful 등 Open API 서비스 프로토콜에 따른 제약이 규제 사항에 포함될 수 있다.

두 번째 단계에서는 Open API 요구사항 정의에서 Open API 사용을 특정기간 내 몇 번으로 제약할 지에 대한 사용 관점과 어떤 콘텐츠까지 허용해 줄지에 대한 관점, 어떤 기능까지 제공할 지에 대한 기능적 관점에서의 제약 사항을 설정하는 것이다. 지도 Open API 를 예를 들면 사용 관점에서는 특정 사용자 앱에서 하루에 500 번까지로 Open API 호출을 제한하고, 콘텐츠 관점에서는 지도 내의 정부 시설에 대한 정보는 표시하지 않으며, 기능 관점에서는 기능적으로 축척률을 1:1,000 까지로 지원을 제한한다.

세 번째 단계에서는 앞선 두 단계에서 생성한 Open API 요구사항 정의와 Open API 제약 사항을 이용하여 Open API 명세서를 작성하는 것이다. Open API 요구사항 명세서 작성의 관점은 어떻게 요구사항을 만족 시키는가가 아니라 어떤 요구사항이 있는가에 맞춰져야 한다.

Open API 개발의 네 번째 단계는 Open API 명세서의 내용에 따라 적합한 구현 방법을 선정하고 설계하는 것이다. 지도 서비스를 위한 Open API 설계로 예를 들면 자바 언어를 사용하여 Open API 서비스 제공 방식은 RESTful 로 선정하고, 데이터 형식은 JSON 을 사용하며, Open API 사용 시 인증 방법은 OAuth 2.0 을 사용하도록 설계하는 것이다.

마지막으로 다섯 번째 단계에는 Open API 설계에 따라 구체적인 프로그램 언어를 사용하여 Open API 를 구현하는 것이다. 그림 4 는 앞서 설명한 Open API 개발 절차를 보여준다.

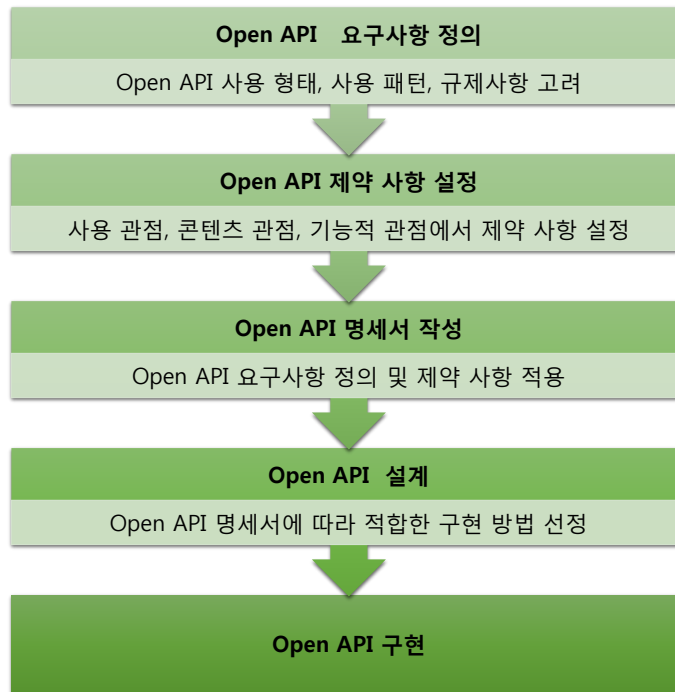


그림 4. Open API 개발 절차

뿐만 아니라 Open API 사용자들을 위한 커뮤니티 기능을 제공하여 사용자들로 하여금 개발한 Open API 사용을 유도한다. 두 번째 절차는 Open API 포털에 Open API 수행 동작 및 버전 등의 정보가 기술된 매뉴얼 문서를 사용자가 이용 가능한 형태로 업로드 하는 것이다. 세 번째 절차는 Open API 포털의 회원(Open API 사용자 및 개발자)에게 Open API 사용을 위한 인증키를 발급하고 Open API의 원활한 사용을 허용하는 것이다. 마지막 절차는 앞선 두 번째, 세 번째 단계를 반복하면서 Open API 포털 관리를 수행하여 사용자들로 하여금 개발한 Open API 사용을 장려하는 것이다.

3.2 Open API 개발을 위한 도구

Open API의 중요성이 증가하고 Open API의 개발이 정부, 기업, 개인으로 점차 늘어감에 따라 Open API 개발을 도와주는 도구들이 출현하였다. Open API 개발을 도와주는 도구들은 Open API 개발적인 측면에서 배포적인 측면까지 Open API 개발을 전반적으로 도와준다. 먼저 Open API 개발 측면에서 쉽게 Open API 개발을 달성할 수 있도록 웹이나 클라우드, 애플리케이션의 UI를 통해서 Open API 개발을 도와준다. 뿐만 아니라 규격화 된 Open API 개발을 통해 개발 시간 단축 및 개발 양을 줄여준다. 그리고 Open API 배포를 위해 기본적인 Open API 포털 기능부터 복잡한 Open API 포털 기능까지 제공해준다. 이 절에서는 현재까지 출시된 Open API 도구들 중 가장 많이 사용되고 있는 3가지 도구에 대해서 조사하고 설명한다.

CA Technology에서는 Open API를 개발 및 배포하기 위한 도구 모음으로 CA API Developer Portal, CA API Gateway, CA API Gateway를 제어하기 위한 Policy Manager를 출시했다[19]. CA API Developer Portal은 외부 개발자들이 웹 기반으로 접속하여 Open API를 생성할 수 있도록 도와주며, CA Policy Manager는 CA API Gateway를 관리하며 관리자로서 Open API를 개발하고 Portal을 관리하는 역할을 수행한다. CA API Gateway는 back-end 서버들에 Open API 서비스를 요청하고, CA API Developer Portal로부터 수신된 메시지를 처리한다. 뿐만 아니라 back-end 서버로의 부하 분산과 사용자별 Open API 이용 로그 수집, 사용자의 Open API 사용 제어, OAuth Token 인증 등을 수행한다. CA Technology의 Open API 개발 도구들은 외부 사용자에게 Open API를 배포하기 위해 로컬 네트워크 내에 CA Open API 개발 도구를 설치해야 하며 다양한 기능을 Open API 개발자에게 제공한다. CA Technology의 Open API 개발 도구는 Open API 사용자 관리, SoAP 기반 API 생성, SoAP 기반 API의 배포를 위한 WSDL 생성, 웹 API 생성, RESTful API 생성, Key(Token) 관리, 사용자 별

통계, 계층화 데이터 관리를 위한 LDAP(Lightweight Directory Access Protocol) 생성 기능 등을 UI 로 수행한다. 그림 5 는 CA Technology 의 Open API 개발 도구의 개념도를 보여준다.

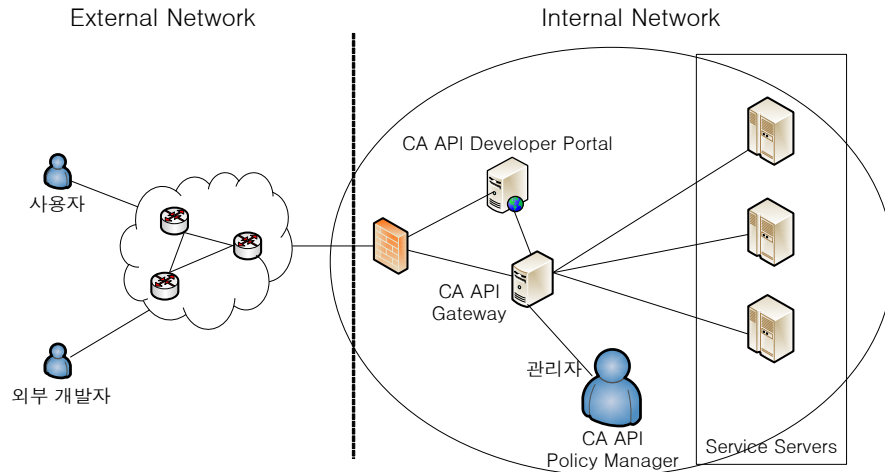


그림 5. CA Technology 의 Open API 개발 도구 개념도

다른 Open API 개발을 위한 도구로 Mulesoft 의 Anypoint 플랫폼이 있다[20]. Mulesoft 의 Anypoint 는 별도의 설치 없이 클라우드 환경에 접속하여 Open API 개발과 배포를 도와주는 Open API 개발 도구이다. Anypoint 는 SOAP, RESTful 기반의 Open API 를 설계, Build, 관리, Open API 인증을 할 수 있는 기능을 제공한다. 추가적으로 Open API 개발을 위한 빌드 도구도 지원한다. 뿐만 아니라 개발한 Open API 배포를 위한 간단한 포털을 제공하고 Open API 버전 관리 탭과 웹 매뉴얼 기능, 사용자 Open API 통계 기능도 제공한다. 그림 6 은 MuleSoft 의 Anypoint 의 개념도를 보여준다.

마지막으로 살펴볼 Open API 개발 도구는 Apigee 의 Apigee Edge API management 이다[21]. Apigee Edge 역시 설치가 필요 없는 클라우드 기반의 Open API 관리 도구로써 RESTful 기반 Open API 의 설계 및 구현을 도와주며, 사용자별 Open API 사용 제어, Open API 인증, 메시지 변환, 사용자별 Open API 통계 기능을 제공한다. 그림 7 은 Apigee 의 Apigee Edge 의 개념도를 보여준다.

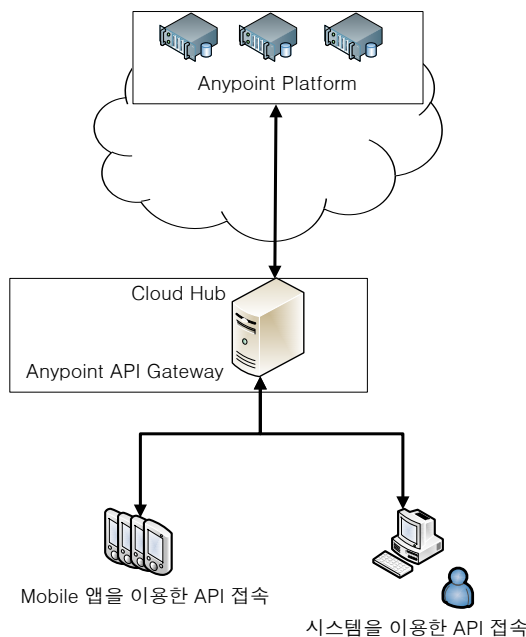


그림 6. Anypoint 플랫폼 개념도

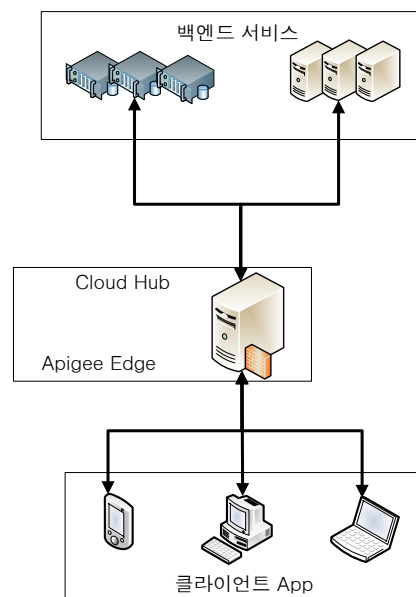


그림 7. Apigee Edge 개념도

표 3. Open API 개발 툴 비교

| | CA API Gateway & CA API Developer Portal (CA) | Anypoint (Mulesoft) | Apigee Edge (Apigee) |
|----|---|---|--|
| 기능 | <ul style="list-style-type: none"> •Open API 설계 •SoAP API 생성 •RESTful API 생성 •Web API 생성 •SLA(Service Level Agreement) •Open API 포털 사용자 관리 •Open API Key 관리 •Open API 인증 •메시지 변환 •Open API 포털 지원 •사용자 별 이용 로그 수집 •사용자 별 통계 지원 •LDAP Provider 생성 •WSDL 생성 •인증서 관리 •Open API 사용자 관리 | <ul style="list-style-type: none"> •Open API 설계 •SoAP API 생성 •RESTful API 생성 •사용자 별 통계 지원 •Open API 포털 지원 •Open API 인증 •클라우드 기반으로 동작 | <ul style="list-style-type: none"> •Open API 설계 •RESTful API 생성 •SLA(Service Level Agreement) •LDAP Provider 생성 •사용자 별 통계 지원 •Open API 포털 지원 •Open API 인증 •메시지 변환 •클라우드 기반으로 동작 |
| 장점 | <ul style="list-style-type: none"> •Open API 개발 외에 다양한 기능을 수행 •프로젝트 수행 시 수행 목적에 따라 상세 설정이 가능함 | <ul style="list-style-type: none"> •설치 없이 사용 가능 •빠른 API 포털 구축 가능 •무료 | <ul style="list-style-type: none"> •설치 없이 사용 가능 •빠른 API 포털 구축 가능 |
| 단점 | <ul style="list-style-type: none"> •CA의 Open API 관리 관련 도구 모음의 설치 필요 •내부 네트워크에 설치함으로써 추가적인 보안 관리 필요 •유료 | <ul style="list-style-type: none"> •프로젝트 수행 시 수행 목적에 따라 상세 설정이 어려움 •CA Technology 제품에 비해 기능이 제한적임 | <ul style="list-style-type: none"> •프로젝트 수행 시 수행 목적에 따라 상세 설정이 어려움 •CA Technology 제품에 비해 기능이 제한적임 •유료 |

표 3은 조사한 CA Technology의 Open API 개발 도구와 Mulesoft의 Anypoint, Apigee의 Apigee Edge를 비교 분석한 것이다. 프로젝트 수행 범위가 작아 Open API의 상세한 설정을 하지 않아도 되는 경우나 프로젝트의 빠른 결과가 필요할 시에는 Mulesoft의 Anypoint, Apigee의 Apigee Edge를 이용하는 것이 더 유리하다. 하지만 프로젝트의 규모가 커서 프로젝트의 운영 정책이나 Open API의 상세 설정이 필요하거나 다양한 기능이 필요할 경우에는 CA Technology의 Open API 개발 도구들을 사용하는 것이 더 유용하다.

물론 Open API 개발을 처음부터 도구를 사용하지 않고 직접 Open API 구현과 배포를 수행할 수도 있다. 하지만 시간적 측면, 투입 인력 측면, 비용 측면에서 Open API 개발 도구를 이용하는 것이 효율적인 방법이다.

4. 결론 및 향후 연구

Open API는 앞으로 다가올 사물인터넷과 매쉬업 서비스 시대의 기반 기술로써 Open API의 중요성과 Open API의 개발 및 배포는 꾸준히 증가하고 있다. Open API 개발에는 다양한 기술과 방법이 이용되고 있으며 본 논문에서는 Open API 개발을 위한 구성 기술로 Open API 웹 서비스 방식과 Open API 데이터 형식, Open API 인증에 대해서 살펴보았다. 뿐만 아니라 Open API 개발을 위한 절차와 Open API 배포 절차 그리고 개발 소요시간 측면, 개발 용이성 측면, 개발 비용 측면에서 효율적인 Open API 개발 도구에 대해서도 조사 및 분석하였다.

향후에는 현재까지 조사되지 않은 Open API 개발 도구에 대해서 추가적으로 조사 분석을 수행할 것이다. 또한, 현재 ICT 분야에서 주목 받고 있는 IoT, Big Data, Cloud, Mobile, Security 플랫폼들을 융합한 새로운 매쉬업 서비스 구현을 위한 Open API 개발을 진행할 것이다. 먼저 특정 분야의 매쉬업 서비스에 대한 시나리오를 작성한 후, 작성한 시나리오를 이용하여 본 논문에서 기술된 Open API 개발방법에 따라 요구사항을 도출하여 제약사항을 설정한 후 요구사항 명세서를 작성하고, Open API 설계 및 구현을 수행할 것이다.

5. 참고 문헌

- [1] J. Höller et al., From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st ed.: Elsevier, 2014.
- [2] 공공데이터 포털, (2015, July 7). 공공데이터 포털(한국). [Online]. Available <http://www.data.go.kr>
- [3] U.S. Government's open data, (2015, July 7). The home of the U.S. Government's open data. [Online]. Available <http://www.data.gov>
- [4] Data.GOV.UK, (2015, July 7). Data.GOV.UK Opening up Government. [Online]. Available <http://data.gov.uk>
- [5] Google, (2015, July 12). Google Developer. [Online]. Available <https://developers.google.com/?hl=ko>
- [6] Yahoo, (2015, July 12). Yahoo Developer Network. [Online]. Available <https://developer.yahoo.com>
- [7] Amazon, (2015, July 11). Amazon offerings for developers. [Online]. Available <https://developer.amazon.com>
- [8] Facebook, (2015 July 11). Facebook Developers. [Online]. Available <https://developers.facebook.com>
- [9] W3C, (2015, July 11) SOAP Specifications. [Online]. Available <http://www.w3.org/TR/soap>
- [10] 박유미 et al., "SoAP 기반 웹 서비스와 RESTful 웹서비스 기술 비교", ETRI 전자통신 동향분석, 제 25 권 제 2 호, 2010 년, pp. 112-120.
- [11] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Dept. Inform. Comput. Sci., University of California, IRVINE, U.S.A., 2000.
- [12] W3C, (2015, July 15). Extensible Markup Language(XML). [Online]. Available <http://www.w3.org/XML>
- [13] W3C, (2015, July 15). Guide to the W3C XML Specification ("XMLspec") DTD, Version 2.1. [Online]. Available <http://www.w3.org/XML/1998/06/xmlspec-report-v21.htm>
- [14] JavaScript Object notation(JSON), (2015, July 15) JSON 개요, <http://json.org/json-ko.html>
- [15] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of json and xml data interchange formats: A case study," in ISCA 22nd Intl Conf. on Comp. and their App. in Industry and Engg, San Francisco, California, 2009, pp. 157-62.
- [16] The OAuth 1.0 Protocol, RFC 5849. April 2010.
- [17] The OAuth 2.0 Authorization Framework, RFC 6749, October 2012.
- [18] *Akana Building Successful APIs*, Akana, Inc., Los Angeles, CA, 2012.
- [19] CA Technology, (2015, July 16). CA Technology API Management. [Online]. Available <http://www.ca.com/us/products/api-management.aspx?intcmp=headernav>
- [20] Mulesoft, (2015, July 17). Mulesoft Anypoint Platform. [Online]. Available <https://www.mulesoft.com/platform/enterprise-integration>
- [21] Apigee, (2015, July 16). Apigee Intelligent API Management. [Online]. Available <https://apigee.com/about/products/api-management>



김 태 영

2010 계명대학교 컴퓨터공학과 학사

2010 ~ 2012 계명대학교 컴퓨터공학과 석사

2012 ~ 2014 (주)스맥 시스템개발 전임연구원

2014 ~ 현재 계명대학교 컴퓨터공학과 박사과정

<관심분야> 네트워크 관리 및 보안, IoT 관리, Open API 개발



권 동 우

2010 계명대학교 컴퓨터공학과 학사

2010 ~ 2012 계명대학교 컴퓨터공학과 석사

2013 ~ 현재 계명대학교 컴퓨터공학과 박사과정

<관심분야> 네트워크 관리 및 보안, 미디어 스트리밍, SDN 관리, 인터넷 침입예측



김 현 우

2010 계명대학교 컴퓨터공학과 학사

2010 ~ 2012 계명대학교 컴퓨터공학과 석사

2012 ~ 현재 계명대학교 컴퓨터공학과 박사과정

<관심분야> 네트워크 관리 및 보안, 방화벽 정책 관리



주 홍 택

1989 한국과학기술원 전자계산학과 학사

1989 ~ 1991 포항공과대학교 컴퓨터공학과 석사

1991 ~ 1997 대우통신종합연구소 선임연구원

1997 ~ 2002 포항공과대학교 컴퓨터공학과 박사

2002 ~ 현재 계명대학교 컴퓨터공학과 교수

<관심분야> 네트워크 관리 및 보안, IoT 관리, Open API 개발