SaaS 플랫폼을 위한 사용자 권한 관리 시스템 설계

한재일, 정기용, 성정욱, 한창훈, 손영수, 김영만
국민대학교 컴퓨터공학부
{jhan, admin, bogerking, asura349, alcmzl, ymkim}@kookmin.ac.kr

Design of User Authority Management System for SaaS Platform

Jaeil Han, Kiyong Jung, Jungwook Sung, Changhoon Han, Youngsoo Son, Youngman Kim
School of Computer Science, Kookmin University

요 약

SaaS 환경에서는 다수의 사용자가 동일한 어플리케이션을 이용하게 되므로 사용자에 따라 특정 자원 혹은 기능에 대한 접근을 제어하는 보안 기능의 중요성이 더욱 부각되고 있다. 이러한 기능을 갖춘 SaaS 플랫폼의 중요성과 그 가치는 매우 크다 할 수 있으나 국내에서는 이에 대한 많은 연구가 보이지 않고 있다. 본 논문에서는 계층적 그룹 구조, 접속 시간·IP 주소·MAC ID 등에 따른 역할 부여 등 풍부한 기능을 제공하는 SaaS 플랫폼을 위한 사용자 권한 관리 시스템을 설계한다.

1. 서론

최근 소프트웨어 시장에서 떠오르고 있는 키워드로 클라우드 컴퓨팅, 가상화, SOA, RIA 등을 꼽을 수 있다. 이들의 공통점은 서비스에 있으며 서비스는 시장의 주된 성장동력으로 자리매김하고 있다[1].

SaaS(Software as a Service)는 소프트웨어를 배포하는 새로운 방식으로, 클라우드 컴퓨팅의 핵심이 되는 어플리케이션 가상화 기술이다[2]. 이는 기존에도 존재하는 유통 방식이었으나, 클라우드 컴퓨팅, 가상화 등 제반 기술과 네트워크 환경의 발전으로 인해 장점이 부각되면서 더욱 주목받고 있다.

이러한 SaaS 환경에서는 다수의 사용자가 동일한 환경에서 동일한 어플리케이션을 이용하게 된다. 때문에 사용자에 따라 특정 자원 혹은 기능에 대한 접근을 제어하는 보안 기능의 중요성이 더욱 부각되고 있으며, 이러한 기능을 모두 갖춘 SaaS 플랫폼의 중요성과 그 가치는 매우 크다 할 수 있다.

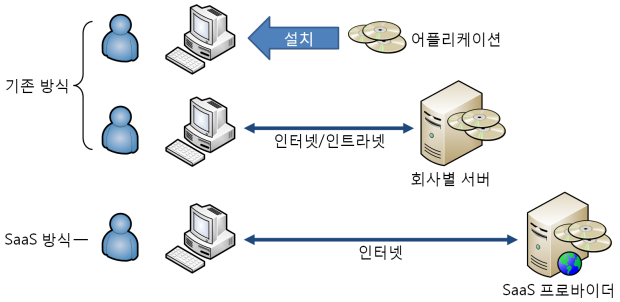
본 논문에서는 SaaS 플랫폼을 위한 사용자 권한 관리 시스템을 설계한다. 2장에서는 관련 연구를, 3장에서는 설계한 시스템 구조 및 권한 부여 메커니즘을 기술하고 5장에서 결론을 맺는다.

2. 관련 연구 및 문제점

2.1. SaaS

2.1.1. 개요

전통적 소프트웨어 비즈니스 모델과 SaaS의 중요한 차이점 중 하나는 제품 소유의 여부인데, 기존 기업용 소프트웨어는 기업 내부의 서버 등 장비에 저장해 이용한다는 점에서 고객이 소유권을 갖고 있었지만, SaaS는 소프트웨어가 제품이 아닌 서비스, 즉 빌려 쓰는 모델이라는 점에서 기존 라이센스 모델과 확연히 구분된다[3]. (그림 1)은 사용자의 설치 또는 회사별 서버 유지가 필요한 종전의 소프트웨어 배포방식과 SaaS 방식을 비교한 것이다.



(그림 1) 소프트웨어 배포방식

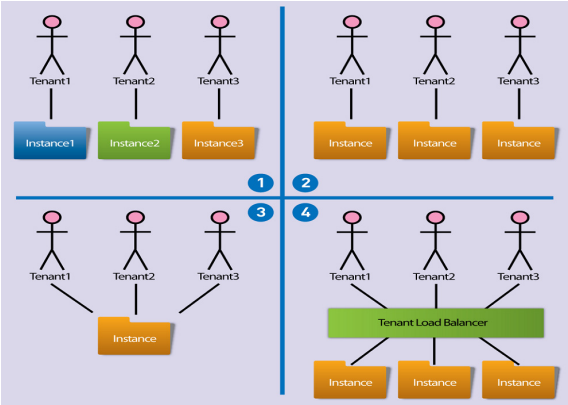
또한 기반 기술인 클라우드 컴퓨팅, 가상화를 비롯해 웹 서비스 및 SOA 기술의 성숙과 Ajax 등과 같은 신개발방식의 보편화에 따라 점차 널리 행해지는 소프트웨어 배포 모델이 되고 있다. 또한 광대역 서비스가 확대됨에 따라 세계적으로 더 많은 지역에서 사용자들이 이러한 서비스에 접근할 수 있게 되었다[4].

2.1.2. SaaS 성숙도 모델

SaaS 모델은 그 성숙도에 따라 네 단계로 나뉠 수 있다. (그림 2)는 SaaS 모델의 성숙도를 도식화한 것이다. 1단계는 고객 각각에게 커스터마이징된 인스턴스를 제공하는 방식으로, ASP와 유사하다. 2단계는 고객마다 인스턴

* 본 연구는 중소기업청의 "PIR/IPR/PDR 센서 기반의 물체 추적 시스템 소프트웨어 개발" (A2010-0164) 과제의 지원을 받아 수행한 결과임.

스를 제공하는 것은 1단계와 동일하지만 하나의 범용 인스턴스를 설정하여 이용한다는 점이 다르다. 3단계는 멀티테넌트 환경을 실현하여 여러 사용자가 단 하나의 인스턴스를 공유하는 모델로, 본 논문에서 다루고자 하는 방식이다. 마지막 4단계는 로드 밸런서를 이용하여 자유롭게 확장 가능하도록 지원하는 모델로, SaaS의 궁극적 목표라 할 수 있다[5].



(그림 2) SaaS 성숙도 모델[2]

2.1.3. 국내외 현황

SaaS 시장은 해를 거듭할수록 발전하고 있는데, 가트너는 2010년 7월 당해 SaaS 시장의 규모를 2009년 대비 14.% 성장한 85억달러 이상으로 전망하였으며[6], 또한 IDC는 전세계 SaaS 시장규모가 매년 21% 증가하여 2015년에는 320억 달러에 달할 것으로 예측하였다[2].

국내시장의 경우 글로벌 기업의 국내진출로 인해 시장 확대 및 경쟁이 전망되나, 국내의 SaaS 기술은 초보 수준으로 세계적인 추세에 1~2년 가량 뒤쳐져 있는 실정이다. Salesforce, Microsoft, Oracle등이 성숙도 3단계 이상의 SaaS를 구현 및 서비스하는 것에 비해 국내 주요 SaaS 공급업체의 SaaS 성숙도는 2단계에 머물러 있다[7]. SaaS 성숙도 3단계로 올라서기 위해 필수적인 요소인 멀티테넌트 환경에 대한 연구 및 개발이 미진하기 때문이다.

2.2. SaaS 플랫폼의 사용자 권한 관리 사례

2.2.1. Force.com

Force.com은 세계 SaaS 시장에서 선두를 달리고 있는 Salesforce의 SaaS 플랫폼이다. Salesforce는 1999년에 설립되어 CRM 솔루션을 시작으로 SaaS에서 큰 성공을 거두었으며, 이를 바탕으로 클라우드 컴퓨팅 분야로 사업을 확장하여 진출하고 있다.

Force.com은 공유정책을 통해 상세한 권한 설정을 지원하는데, 데이터베이스의 자원에 대해 그룹, 역할, 개인 등에 대한 권한을 설정할 수 있다. 공유의 대상이 되는 자원 또한 개개의 레코드부터 특정한 레코드들의 집합까지 세부적으로 설정 가능하다. 전체적인 공유정책도 다수의 자료, 다수의 사용자로부터 하나의 자료, 하나의 사용자에 이르기까지 이를 점점 세분화 시켜 설정할 수 있다. 또한, 실제 조직도를 기반으로 부서간의 계층관계에 따라 공유권한이 상속되는 등의 구조를 지원한다[8].

2.2.2. SaaSpia

SaaSpia는 ETRI에서 2009년부터개발중인 SaaS 플랫폼으로, 고성능 메타데이터 처리 기술, 고성능 멀티테넌트 애플리케이션 실행엔진 기술, 테넌트별 데이터의 무결성을 보장하는 보안기술의 개발에 중점을 두고 있다[9].

SaaSpia에서 사용자의 권한은 사용자-권한-메뉴간의 매핑관계를 통해 부여된다. 즉, 각각의 메뉴에 ID가 부여되며, 플랫폼이 사용자와 메뉴간의 관계를 관리한다[10].

3. 시스템 설계

3.1. 목표 및 요구사항

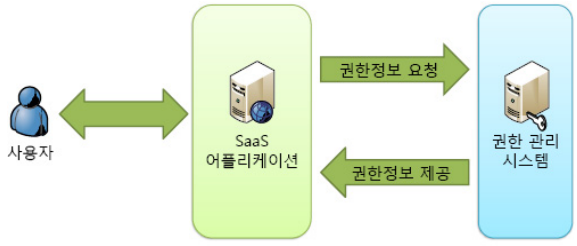
본 시스템의 목표는 사용자가 SaaS 어플리케이션의 특정 기능을 사용할 수 있는지에 대한 권한 정보를 관리하는 것이다.

이 때 사용자에게 권한은 다양한 경로로 부여될 수 있다. 회사원을 예로 들면 소속된 부서·직급·근무지 등에 따라 다양한 권한이 부여될 수 있으며, 임시로 특정 업무를 대리하는 등의 각종 예외 상황이 존재할 수 있다. 또한 업무 시간에 사내에서 접속한 경우만 특정 기능을 이용할 수 있도록 하는 등 사용 환경에 따라 다른 권한을 부여하는 경우도 생각해 볼 수 있다.

권한 관리 시스템은 이 같은 다양한 경우를 자연스럽게 관리할 수 있어야 하며, 한편 권한 부여 또는 수정 시에 모든 사용자에게 권한을 일일이 부여 또는 재부여하는 등의 중복 정의는 최소화되어야 한다.

3.2. 시스템 구조

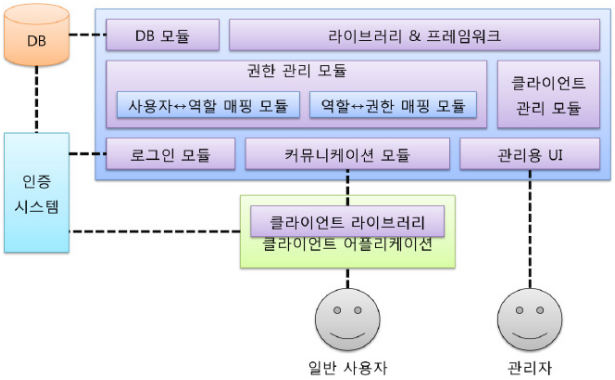
본 시스템은 (그림 3)에 나타난 것처럼 SaaS 어플리케이션으로부터 사용자의 권한을 요청받아 이를 제공하는 서버 역할을 수행한다. 따라서 클라이언트 측에 제공하는 API와 라이브러리를 함께 구현하여야 한다.



(그림 3) 어플리케이션과 시스템간 상호작용

전체 시스템의 구조는 (그림 4)와 같다. 권한 관리 모듈을 비롯해 DB모듈, SaaS 어플리케이션의 식별자와 기능 목록을 관리하는 클라이언트 관리 모듈, SaaS 어플리케이션의 요청을 받아 정보를 제공하는 커뮤니케이션 모듈, 이에 대응하여 SaaS 어플리케이션 측에서 동작하는 클라이언트 라이브러리 등이 존재한다. 이 중 핵심 기능을 수행

하는 권한 관리 모듈의 내부 매커니즘에 대해서는 3.3.에서 자세히 기술한다.



(그림 4) 시스템 구조

또한 사용자의 권한을 편집하는 관리용 UI가 존재하는데, 이는 본 시스템에 대한 클라이언트 어플리케이션으로서 동작한다. 즉, 관리용 UI를 사용하기 위한 권한 역시 본 시스템의 내부에서 관리하게 된다.

한편 사용자 인증을 위한 시스템은 별도로 구현하지 않고 기존에 존재하는 외부 시스템의 것을 그대로 이용하며, 본 시스템은 외부 사용자 인증 시스템의 클라이언트로서 동작한다. 이렇게 하면 플랫폼에 적합한 인증 프로토콜을 사용하는 인증 시스템을 이용하거나 SSO을 구현한 시스템 등을 선택적으로 사용할 수 있어 보다 유연한 구성이 가능하다.

3.3. 권한 부여 매커니즘

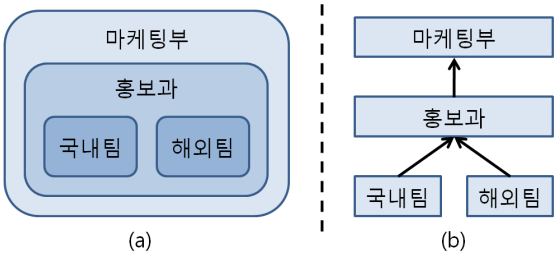
본 시스템은 3.1.에서 기술한 요구사항을 만족시키기 위해 사용자-그룹-조건-역할-권한 매핑 관계를 이용한다. 이 구조를 통해 다양한 조건에 따른 역할 정의를 효율적으로 표현할 수 있으며 중복 정의를 최소화할 수 있다.

여기서 권한은 특정 어플리케이션의 한 기능에 대한 접근 권한을 나타낸 것이다. 즉, 어떤 기능을 사용할 수 있는지 없는지 여부를 나타내는 가장 작은 단위의 정보가 된다. 역할은 권한의 집합으로, 그룹 또는 사용자에게 부여된다. 핵심이 되는 요소인 그룹과 조건에 대해서는 이어지는 절에서 상세히 기술한다.

3.3.1. 그룹

그룹은 사용자 또는 그룹의 집합으로, 역할이 부여되는 대상이다. 사용자는 동시에 여러 그룹에 소속될 수 있으며, 그룹은 최대 한 그룹에만 소속될 수 있다.

이 때 한 그룹의 원소는 해당 그룹이 가지고 있는 모든 역할을 부여받게 된다. 즉, 그룹 간에는 상속 관계가 발생하게 되며, 이를 이용하면 회사 조직과 같은 구조를 나타낼 수 있다. 예를 들어 마케팅부 내에 홍보과, 홍보과 내에 국내팀이 있는 회사에서, 홍보과에 소속된 사용자는 마케팅부의 모든 역할을, 국내팀에 소속된 사용자는 마케팅부와 홍보부의 모든 역할을 상속받는다. (그림 5)의 (a)는 이 같은 구조를 다이어그램으로 나타낸 것이며, (b)는 이들 그룹 간의 관계를 상하 구조로 다시 나타낸 것이다.



(그림 5) 그룹 계층 구조

또한 한 사용자가 동시에 여러 그룹에 소속될 수 있으므로 태스크포스나 면접관 같은 특별한 조직을 자연스럽게 나타낼 수 있다.

3.3.2. 조건

3.3.에서 기술한 것처럼 한 그룹에는 여러 역할이 부여될 수 있으나, 특정한 한 시점에는 한 개의 역할만이 할당된다. 즉, 사용자의 접속 시간 및 접속 환경 등 여러 조건에 따라 한 개의 역할이 선택되게 되는데, 이러한 조건으로는 접속 시간, IP 주소, MAC ID 등이 있다.

3.3.3. 충돌 회피

한 사용자에게 최종적으로 권한이 부여되는 경로가 다양하기 때문에 권한 간 충돌이 발생할 수 있다. 예를 들면 한 사용자에게 역할 1과 2가 할당되었는데, 역할 1에서는 기능 A를 허용하고 역할 2에서는 이를 차단할 수 있다.

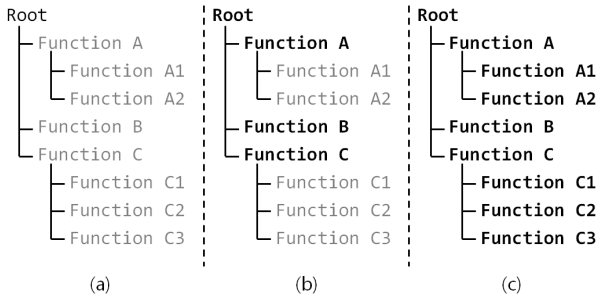
이처럼 권한이 충돌하는 경우 우선적으로 권한을 허용하도록 한다. 즉, 사용자에게 부여된 여러 역할 중 단 하나라도 특정 기능을 허용하면 사용자는 해당 기능을 이용할 수 있게 된다. 이는 모든 권한이 기본적으로 차단된 상태이고 관리자가 특정 사용자에게 허용하는 식으로 시스템을 설정하는 것이 일반적이므로 자연스럽다.

또한 이와는 별도로 역할 간에 정수로 된 우선순위를 부여할 수 있도록 해 우선순위가 높은 역할이 존재할 경우 무조건 해당 역할에서 정의한 권한을 부여하도록 한다. 단 우선순위가 높은 역할에서 정의하지 않은 기능은 다음으로 우선순위가 높은 역할에서 부여한 내용을 따르도록 하는 식으로 권한을 부여한다.

3.4. 프로토콜

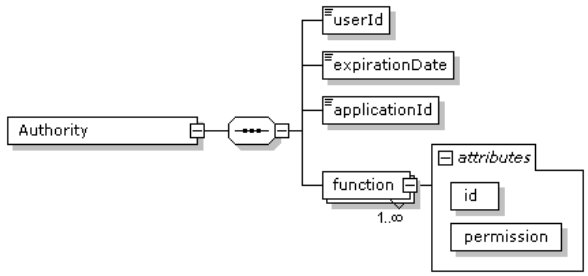
사용자가 SaaS 어플리케이션에 최초로 접속하면 SaaS 어플리케이션은 권한 관리 시스템에 사용자의 권한정보를 요청하게 되고, 권한 관리 시스템은 해당 사용자의 권한정보를 제공해야 한다. 이와 같은 요청/응답 과정에서는 HTTP 프로토콜을 이용하며, 응답 데이터는 XML로 표현한다. 즉 SaaS 어플리케이션은 미리 약속된 권한 관리 시스템의 URL에 접근해 본 시스템에 해당 사용자의 권한정보를 요청하게 된다.

이 때 파라메터로 자신의 식별자와 인증 키, 사용자의 식별자, 인증 정보가 필요한 기능의 식별자와 깊이를 전송한다. 여기서 깊이는 특정 기능을 기준으로 하위 몇 단계 기능까지의 정보를 가져올 것인지를 나타낸다. 예를 들면, (그림 6)과 같은 기능 계층 구조를 가진 SaaS 어플리케이션의 Root 기능에 대해 권한 정보를 요청했을 때 깊이가 0일 경우 (그림 6)의 (a), 1일 경우 (b), 2일 경우 (c)의 굵은 글씨로 표현된 기능에 대한 권한 정보가 제공된다.

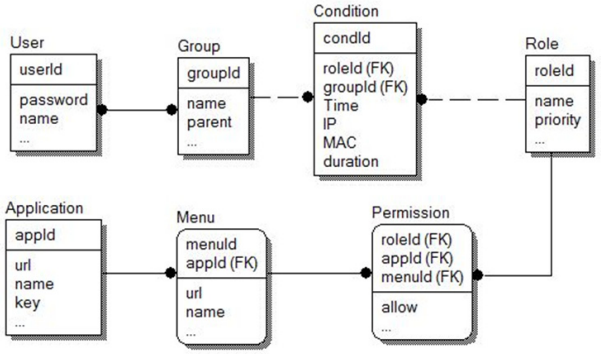


(그림 6) 기능 계층 구조 예

(그림 7)은 권한 정보를 기술하는 응답 XML의 포맷을 나타낸 것이다. <applicationId>는 정보를 요청한 어플리케이션의 식별자를, <userId>는 해당 파일이 기술하는 사용자의 식별자를, <expirationDate>는 이 정보가 만료되는 시점을 나타낸다. <function> 엘리먼트는 각각의 기능에 대한 권한을 나타내는데, id 속성은 기능의 식별자를, permission속성은 allow, deny 등 해당 기능에 대한 권한을 나타낸다. 이 때 <function> 엘리먼트는 중첩되어 계층 구조를 나타낼 수 있다.



(그림 7) XML 포맷



(그림 8) DB 구조

3.4. DB 구조

(그림 8)은 시스템에 반드시 필요한 데이터만을 간략하게 나타낸 ER 다이어그램이다. 매핑에 사용되는 각각의 요소가 테이블로 표현되어 있다. 이 때 사용자의 정보를 담는 User 테이블은 외부 인증 시스템의 것을 공유한다.

4. 결론

본 논문에서는 SaaS 플랫폼에 반드시 필요한 요소 중 하나인 사용자 권한 관리 시스템을 설계하였다.

본 논문에서 설계한 시스템은 사용자의 권한 관리를 위해 계층적 그룹 구조, 접속 시간・IP 주소・MAC ID 등에 따른 역할 부여 등 풍부한 기능을 제공한다. 또한 플랫폼은 물론 인증 시스템에 독립적인 구조이므로 다양한 SaaS 플랫폼에 적용할 수 있으며, 비단 SaaS 플랫폼만이 아니라 대규모 사용자를 대상으로 하는 어플리케이션에 폭넓게 활용 가능하다.

추후에는 본 논문에서 설계한 시스템을 실제로 구현함과 더불어 사용자-권한 매핑 과정에서 가해지는 DB 부하를 최소화하기 위한 연구를 진행할 계정이다.

참고문헌

[1] 신현석, “MS 클라우드 컴퓨팅과 애저 서비스 플랫폼 이해,” ZDNet Korea, 2008

[2] 김영만, “웹기반 SaaS 플랫폼,” 2008

[3] “서비스형 소프트웨어”, TTA 정보통신 용어사전

[4] “SaaS (software as a service); 서비스형 소프트웨어)”, http://terms.co.kr/SaaS.htm

[5] Frederick Chong 외, “Architecture Strategies for Catching the Long Tail”, MSDN

[6] “올 전세계 SaaS시장, 전년보다 14% 성장 85억달러”, 전자신문, http://www.etnews.co.kr/news/detail.html?id=201007230120

[7] TTA, “ICT 중점기술 표준화전략맵 Ver. 2011”, 2011

[8] “Salesforce Security Implementation Guide”, http://na1.salesforce.com/help/doc/en/salesforce_security_impl_guide.pdf

[9] 김형환 외, “SaaS 기술 개발 동향”, 전자통신동향분석 제24권 제4호, 2009

[10] ETRI, “SaaSpia 권한 관리”, 2010