

이해당사자의 관점에서 분류한 SaaS 플랫폼 역량

이지현* 허성진*

*한국전자통신연구원

e-mail : jihyun@etri.re.kr, sjheo@etri.re.kr

A Study on Capability of Software as a Service Platform perceived from SaaS Stakeholders

Jihyun Lee* Sung Jin Hur*

*Electronics and Telecommunications Research Institute

요 약

어플리케이션에 대해 사용한 후 비용을 지불하는 use first, pay later 서비스 개념의 SaaS의 구현 성숙도 모델이 Microsoft 사, Dharmesh Shah, Forrester 사에서 정리되어 발표되었다. 하지만 기존의 SaaS 성숙도 모델은 SaaS 생태계에 존재하는 이해당사자들의 관점 중 서비스 제공자의 관점만 반영되어 SaaS 어플리케이션 사용자, 개발자, 테넌트 관리자의 측면에서 고려된 기술 성숙 방향이 반영되어 있지 않다. 본 논문은 SaaS 기본 이론과 업계 제품에서 조사된 이론에 입각해 SaaS 생태계 전반의 SaaS 이해당사자들 관점에서 SaaS 플랫폼 역량의 발전을 분류할 수 있는 체계를 아키텍처에 기반하여 제시하고자 한다.

1. 서론

소프트웨어의 개념이 영구적으로 제품을 구매하여 설치하여 사용하는 데에서 벗어나 웹 상에서 필요할 때 빌려 쓰는 개념으로 사용자에게 점차적으로 다가서면서 소프트웨어의 개념이 변화해 가고 있다. 특히 기업의 필요에 맞춰진 맞춤형 전용 어플리케이션의 경우에 소프트웨어 유지보수 비용 측면에서 외부에서 아웃소싱할 때 자체 개발하여 유지 보수하는 것보다 비용 절감 효과가 있다. 이러한 배경으로 소프트웨어의 서비스화는 떠오르는 기술로 관심을 모아 가고 있다. 이와 같이 차세대 IT 소프트웨어를 서비스로 개발하는 기술이 'Software as a Service' 일명 SaaS 기술이다. SaaS의 중요 개념인 멀티테넌시(multitenancy)는 하나의 소프트웨어 실행 서비스로 여러 클라이언트 회사(multitenant)의 요구 사항에 맞춰진 서비스를 제공하는 것을 의미한다. 하나의 실행 인스턴스로 이를 실현하기 위해 각 회사는 자사의 필요에 알맞은 조건을 직접 지정할 수 있어야 한다. 이를 충족시키기 위해 소프트웨어가 실행되는 환경은 똑같은 UI, 똑같은 데이터베이스 디자인, 똑같은 비즈니스 로직을 제공해서는 안되고 각 각의 항목 별로 차별화된 환경을 설정할 수 있는 기반을 주어야 함을 의미한다. 이것이 바로 SaaS의 구성가능성(configurability)이다. 본 논문에서는 SaaS 플랫폼의 멀티테넌시와 구성가능성을 지원 하는 아키텍처 별 구현 기술과 이해당사자 관점의 SaaS 플랫폼 역량 차이에 대해 분류하고자 한다.

2. SaaS 이해당사자 및 도입 효과

SaaS 생태계를 구성하는 이해당사자는 서비스 사용자와 서비스 제공자의 두 분류로 나뉘며 서비스 사용자에는 테넌트 사용자와 테넌트 관리자가 포함된다. 서비스 제공자에는 서비스 제공자와 SaaS 플랫폼 제공자가 포함되고 이해당사자 별 역할은 다음과 같다.

- (1) 테넌트 사용자
 - 회사에서 제공되는 웹 상의 어플리케이션을 사용하는 최종 사용자(end user)
- (2) 테넌트 관리자
 - 서비스 제공자에게 서비스 사용 권한을 얻은 후 자사의 요구 사항과 조직 내 접근 권한에 맞게 어플리케이션 구성을 담당
 - 최종 사용자가 사용한 서비스의 사용량에 대해 청구된 비용을 지불
- (3) 서비스 제공자
 - 멀티테넌트용 SaaS 어플리케이션을 개발하여 다중 클라이언트 회사가 사용할 수 있도록 배치
 - 테넌트에게 제공한 서비스의 사용을 모니터링하고 이에 대한 비용을 청구함
- (4) SaaS 플랫폼 제공자
 - SaaS 어플리케이션이 실행되는 OS, DBMS 등의 환경을 제공함

SaaS 플랫폼의 도입 효과는 테넌트가 소프트웨어를 구매하는 비용의 일부를 소프트웨어 대여 비용으로 지불할 때 발생한다. 여러 테넌트를 지원하기 위한 소프트웨어를 서비스 제공자가 일일이 변경하여 만들

본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 일환으로 수행하였음. [2009-S-033-01, 중소기업 SW 서비스를 위한 SaaS 플랫폼 개발]

어 주기 보다는 테넌트 마다의 기능 구성 여지를 제공하여 자체적으로 필요에 맞는 변경이 이뤄질 수 있도록 하기 때문에 비용 절감 효과가 있는 것이다.

3. SaaS 구현 기술의 성숙 모델과 국내 도입 단계

SaaS 에 대한 구현 기술 수준은 초기 단계에서부터 단계별로 차츰 향상된 중간 단계를 거쳐 최고의 기술 성숙 상태에 이른다. 세 가지 특징 모델에 관해 소개한다.

- (1) Microsoft 사 관점[1]
- (2) Dharmesh Shah 의 관점[2]
- (3) Forrester 사의 관점[3]

Microsoft 는 확장성, 멀티테넌시, 구성가능성으로 정의되는 SaaS 의 3 대 품질 속성 측면에서의 SaaS 기술 성숙 레벨을 5 단계로 정리하였다. 레벨 0~1 은 멀티테넌트를 지원하지 않으나 레벨 1 은 같은 버전의 소프트웨어를 수행한다는 차이가 있다. 레벨 2 는 테넌트 별로 UI, 비즈니스 규칙, 데이터에 대한 구성을 변경하여 맞춤 서비스를 만든다. 레벨 3 은 레벨 2 의 기능 외 동적으로 로드 밸런싱하여 새로운 소프트웨어 인스턴스를 추가할 수 있는 차이가 있다.

Dharmesh Shah 는 SaaS 를 위한 기준이 없는 무질서의 상태에서부터 서로 다른 버전의 소프트웨어를 각 인스턴스로 제공하는 SaaS 유토피아로의 발전까지 총 5 개 단계로 정의한다.

Forrester 사는 기업의 업무 프로그램을 데이터 센터로부터 제공, 관리, 업그레이드 해 주던 ASP(Application Service Provider)과 같이 아웃소싱 사업이 발전된 것이 SaaS 라고 정의하며 SaaS 의 이상적인 단계는 동적으로 비즈니스 어플리케이션 전체 영역에 대해 서비스화 하는 것이라고 총 6 개의 단계로 정의한다.

각 관점의 단계 별 구분은 SaaS 기술 성숙 모델은 레벨을 정리한 정도나 시각에 따른 차이가 있지만 전반적으로 상당히 겹친다. 현재 국내의 SaaS 기술은 Microsoft 의 관점에 따라 단계 2 의 실현한 후 단계 3 로 진화하기 위한 과정에 있다.

4. 참조 아키텍처로 분류된 SaaS 플랫폼 역량

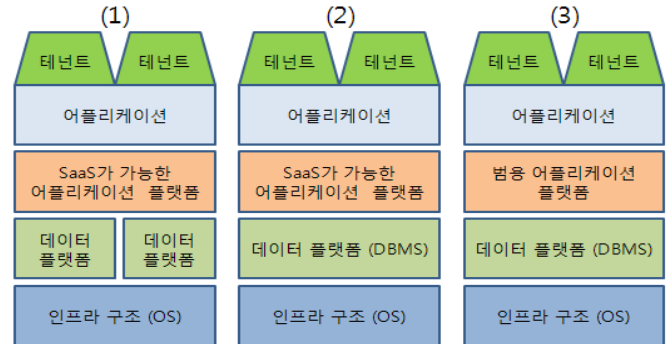
어플리케이션이 서비스로 동작하도록 개발되어야 하는 기술들을 스택으로 표시하고 각 스택 별로 서로 다른 구성 요소를 갖는 총 3 가지 참조 아키텍처를 살펴보자 [4]. 각 아키텍처 별 특징을 서술한다.

첫째, 그림 1 의 (1)에서 보이는 공유 프로세스 기반의 아키텍처는 다수 테넌트의 트랜잭션이 하나의 특정 어플리케이션 서버에서 수행되도록 하고 데이터베이스(일명 데이터 플랫폼)는 물리적으로 분리되어 사용하는 구조이다. 이러한 구조에서 멀티테넌트 지원 아키텍처는 어플리케이션을 개발하는 환경과 어플리케이션을 실행하는 환경으로 나뉜다. 테넌트마다 어플리케이션을 수행하기 위해 각 테넌트의 트랜잭션은 메타데이터에 기반한 해석기를 사용하여 해석함으로써 동적으로 구성 변경된 UI, 비즈니스 로직, 데이터

에 대한 처리를 수행한다.

둘째, 그림 1 의 (2)에서 보이는 전체 공유 기반의 아키텍처는 앞서 설명된 공유 프로세스 기반의 아키텍처에서 보여지듯 테넌트 별로 분리된 물리적 데이터베이스의 사용이 아니라 논리적인 형태의 하나의 데이터베이스를 사용하여 모든 기술을 공유하며 멀티테넌트를 지원한다는 차이가 있다.

셋째, 그림 1 의 (3)에서 보이는 맞춤 기반의 멀티테넌시 지원 아키텍처는 어플리케이션 코드를 구현 수준으로 멀티테넌시를 지원하는 방법으로 이를 위해서 어플리케이션 설계 및 구현에 구조변경(configuration) 알고리즘을 반영하여야 하고 이러한 설계 및 구현 요건들이 해석되는 범용적인 SaaS 플랫폼(즉, 범용 어플리케이션 플랫폼)이 제공되어야 한다. 이 아키텍처는 세 가지의 SaaS 플랫폼 중 설계 및 구현 수준이 가장 높은 아키텍처이다.



(그림 1) SaaS 플랫폼 참조 아키텍처

5. 이해당사자 별 역량 활용 정도

SaaS 플랫폼 이해당사자의 관점에서 SaaS 플랫폼의 역량을 정리해본다.

테넌트 사용자는 SaaS 어플리케이션의 최종 사용자로서 SaaS 플랫폼을 직접 사용하지 않지만 높은 역량을 제공하는 SaaS 플랫폼일수록 보다 복잡한 어플리케이션을 제공받을 수 있다.

테넌트 관리자의 SaaS 어플리케이션 구성변경성은 SaaS 플랫폼 특히 어플리케이션 개발 통합 환경에서 제공하는 구성변경 역량의 수준에 따라 차이가 발생할 수 있다. SaaS 플랫폼의 기능 및 지원 기술에 따라 서비스 제공자가 개발하는 멀티테넌트용 어플리케이션의 종류와 배포(deployment) 방식은 차이가 발생한다. 또한 SaaS 플랫폼 제공자 관점은 확장 가능 방법에 따른 인프라구조 구축 기술에 차이가 발생한다.

참고문헌

- [1] Gianpaolo Carraro, Fred Chong, "Software as a Service (SaaS): An Enterprise Perspective," MSDN, Oct. 2006.
- [2] Dharmesh Shah, "SaaS Maturity Model, "SaaS Maturity Model," InfoQ, Feb. 2009.
- [3] Stefan Ried, "Forrester's SaaS Maturity Model," Forrester Report, Aug. 2008.
- [4] Yefim V. Natis, "Reference Architecture for Multitenancy: Enterprise Computing in the Cloud," Dec. 2008.