論 文
5-4-13

Design and Implementation of Security Framework for Application Server with Components

김행곤(Haeng-Kon Kim)[1] 강전근(Jeon-Geun Kang) [2]

요 약

웹 서비스 애플리케이션의 개발은 변화하는 다양한 이질적인 시스템간의 상호 운영성과 유지가 필수적이고, 사용자 측면의 변경 없이 비즈니스 환경의 변경이 가능하도록 시스템의 확장성과 유연성 및 기존 웹 서비스 애플리케이션을 이용한 재사용성이 제공되어져야 한다. 따라서 웹 서비스 애플리케이션 개발을 위한 CBD(Component Based Development) 적용은 시스템 구축을 위한 자연스런 기술 및 방법론으로 기존의 설계, 구조, 유지보수의 문제점의 해결책으로 제시되고, 웹 분산 환경을 기반한 웹 서비스를 동적으로 빠른 시간에 릴리즈 가능하게 한다. 본 논문에서는 웹 응용 서버 및 모바일 응용 서버 시스템을 위해 개발된 보안 프레임워크의 설계 및 구현과 관련된 컴포넌트를 식별하여 아키텍처에 맞게 계층화시키고, 기존의 UML을 기반으로 웹서비스의 특성을 반영하여 적용 가능한 모델링을 제시한다. 제안된 보안 프레임워크는 레거시 보안 시스템과의 연동, J2EE 보안 지원, JAAS 지원, Kerberos 지원 등 응용 서버 시스템들이 요구하는 다양한 보안 기능을 제공하도록 컴포넌트화 특징을 가진다.

Abstract

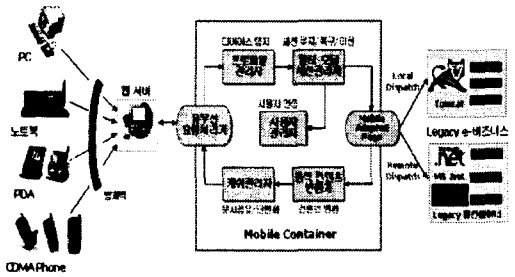
Development of Web service application requires the interoperability between various heterogeneous systems, extensibility to allow modification of business environment not of user interface, maintainability, flexibility and reusability. As the properties of CBD (Component Based Development) have gradually become clear, attention has started to turn the smooth technology and methodology to solve the existing problems and issues the dynamic responds for the distributed web environments. In this paper, we identify some of the major architectural affecting CBD and describe the Design and Implementation of Security Framework for Application Server with Components. We identify the candidate components, model it using UML and layer it on the architecture. The frameworks will provide the various security functions, such as incorporating with legacy security systems, supporting of J2EE, JAAS and Kerberos and assisting in increasing the tailorability of component.

1) 정회원 : 대구카톨릭대학교 컴퓨터공학부 교수
2) 정회원 : 아산정보기능대학 멀티미디어과 교수

논문접수 : 2004. 4. 19.
심사완료 : 2004. 4. 28.

1. 서론

웹 서비스 애플리케이션은 모든 웹 서비스 컴포넌트를 상호 연동 가능하도록 비즈니스간의 통합을 가속화하고 있으며 다양한 비즈니스 요구의 신속한 대응과 이기종 운영 환경의 효과적인 연계 및 이질적인 애플리케이션에 대한 일관적인 인터페이스를 제공하고, 기존 웹 애플리케이션과 서비스를 이용한 재사용을 제공하기 위한 표준 기반의 인터넷 서비스로 주목되어지고 있다[1,2]. 또한 인터넷 환경에서 다양한 응용의 구축과 실행을 지원하도록 개발된 미들웨어 시스템인 응용 서버 시스템은 그림1과 같이 응용 구축을 위한 다양한 지원 도구 및 기능들을 포함하고 있으며, 그 중 응용의 보안을 담당하기 위한 보안 서비스 프레임워크는 응용 개발자의 노력을 최소로 하여, 인터넷 환경에서 필수적으로 요구되는 보안 기능을 응용서버 및 응용 서비스 로직에 제공하는 것을 목표로 하고 있다. 본 논문에서는 웹 응용 서버 및 모바일 응용 서버 시스템을 위해 개발된 보안 프레임워크의 설계 및 구현과 관련된 컴포넌트를 식별하여 아키텍처에 맞게 계층화시키고, 기존의 UML을 기반으로 웹서비스의 특성을 반영하여 적용 가능한 모델링을 제시한다. 제안된 보안 프레임워크는 레거시 보안 시스템과의 연동, J2EE 보안 지원, JAAS 지원, Kerberos 지원 등 응용 서버 시스템들이 요구하는 다양한 보안 기능을 제공하도록 컴포넌트화 특징을 가진다.

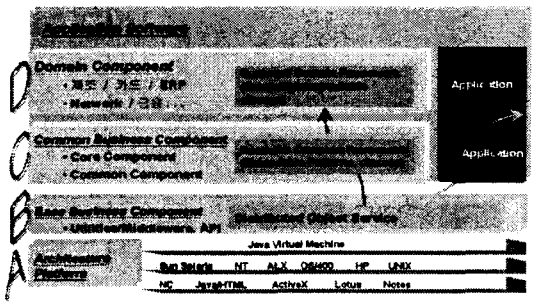


(그림 1) 응용서버 시스템 구조

2. 관련 연구

2.1 컴포넌트 아키텍처

컴포넌트 기반 개발에서 컴포넌트의 생산과 활용을 위한 지침으로 컴포넌트 참조 아키텍처 모델이 필요하다. 컴포넌트 아키텍처는 관련된 다양한 컴포넌트들을 연관시키기 위한 표준 계층으로 컴포넌트의 획득, 이해 및 조립을 위한 레이아웃을 제시함으로써 사용자들이 필요로 하는 컴포넌트들을 식별, 검색하고, 커스터마이즈하여 조립할 수 있는 가이드라인을 제공한다. (그림 2)는 본 연구에서 정의한 ABCD 컴포넌트 아키텍처로서 4 계단(4 tier)으로 구성되어 있으며 인터넷 모바일 응용서비스 컴포넌트 개발과 조립시 참조된다. 도메인별로 별도의 아키텍쳐를 구성할 수 있으나 본 연구에서는 레거시 보안 시스템과의 연동, J2EE 보안 지원, JAAS 지원 등 응용 서버 시스템들이 요구하는 다양한 보안 기능을 제공하는 아키텍쳐를 가진다[3,4]

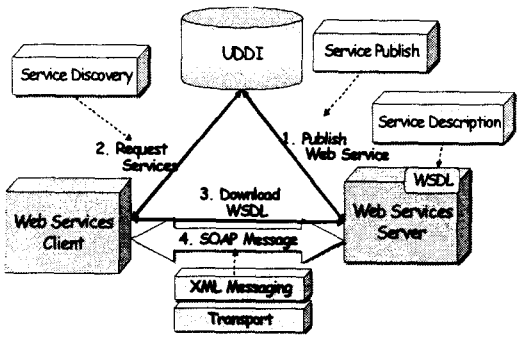


(그림 2) ABCD 컴포넌트 아키텍처

22.2 웹 서비스 개념 및 구조

웹서비스 정의는 크게 기술적 측면과 비즈니스 측면으로 접근할 수 있다. 기술적 측면의 웹 서비스는 인터넷 상에서 표준화된 기술을 사용하여 정의된 소프트웨어 애플리케이션이다. 비즈니스 차원에서는 기업들이 다양한 비즈니스를 발견하여 운영할 수 있게 해줌으로 웹 서비스 자체를 하나의 비즈니스 로직으로 정의할 수 있다. 따라서 웹 서비스는 이 두가지 측면을 통합하여 정의된다. 웹 서비스의 특징은 플랫폼 독립적이고, 디바이스 및 위치 독립적이며, 동적인 기능과 기존 시

스템이 적용 가능하다. 웹 서비스에 기반이



(그림 3) 웹 서비스 구조

되는 표준들로는 SOAP, UDDI, WSDL, XML이 있다. (그림 3)은 이들을 이용한 웹 서비스 아키텍처를 나타낸다[5].

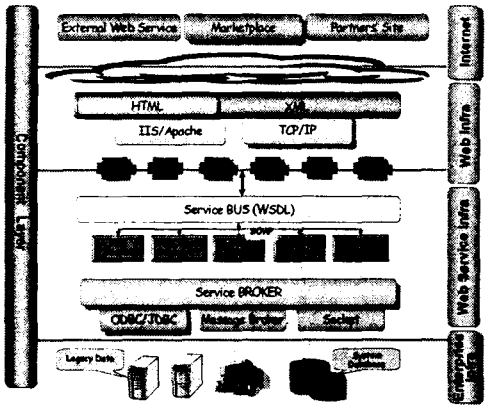
3. CBD 기반 보안 프레임워크 구축

3.1 모바일 응용서버

무선 인터넷 서비스를 제공하기 위한 플랫폼인 모바일 응용 서버는 휴대폰, PDA 등 다양한 무선 단말을 대상으로 한번 저작된 동일한 컨텐츠를 단말기 종류에 상관없이 제공할 수 있도록 설계 개발되었다. 무선 마크업 언어는 회사별로 다양하게 존재하는데, 한번의 저작으로 다양한 무선 마크업 언어를 지원하기 위해, 모바일 응용 서버 시스템은 PWML이란 가상의 단일 무선 마크업 언어에 대한 문서 변환 기능을 통해 다양한 이통사의 단말기를 지원할 수 있게 하였다. 또한 모바일 응용 서버 위에 구축된 비즈니스 업무들은 무선 인터넷의 간헐적 단절성, 업무의 긴급성, 입출력 장치의 제약, 이동성 등을 고려해 단일 사용자가 단말기의 종류를 변경하여 접속하더라도 이전에 처리하였던 내용을 계속 이어 받아 업무를 진행시킬 수 있도록 멀티 모달을 지원한다. 그림 1은 무선 단말을 통해 사용자의 서비스 요청이 접수되었을 때 모바일 응용 서버에서 그 요청을 처리하여 응답하는 과정 동안의 개괄적인 흐름도이다. 사용자의 요청은 무선 요청 처리기를 통해 접수되며, 프로파일 관리자는 접수된 요청의 HTTP 헤더정보로부터 단말기의 특징 정보를 추출한다. 멀티모달 세션 관리자는 단말 특징 정보와 사용자 관리자를 통해 인증된 사용자를 기반으로 응용의 세션을 관리하거나 복구하여 사용중인 세션을 유지시키는 기능을 수행한다. 동적 컨텐츠 변환기는 응용 로직이 수행되고 난 후의 컨텐츠를 단말의 프로파일 정보를 가지고 사용자 단말 환경에 적합하도록 변환한다. 캐쉬 관리자는 변환된 페이지에 대한 정보를 캐쉬하여, 이후 같은 페이지에 대한 요청이 있을 때 캐쉬된 페이지를 사용하여 응답 속도를 개선한다. 모바일 응용서버의 기능 모듈들 중, 그림 4의 사용자 관리자는 모바일 사용자의 인증 및 모바일 응용서버 내 자원들에 대한 접근 권한 관리, 모바일 특화된 보안 서비스를 제공하도록 설계 되었으며, 본 논문에서 설명하는 보안 프레임워크는 사용자 관리자에게 보안과 관련된 API를 제공하여, 모바일 응용서버 시스템의 사용자 보안을 강화하도록 하였다. 또한 모바일 응용 서버 시스템의 소스 기반 보안 및 자원 보호를 위한 플랫폼을 제공한다.

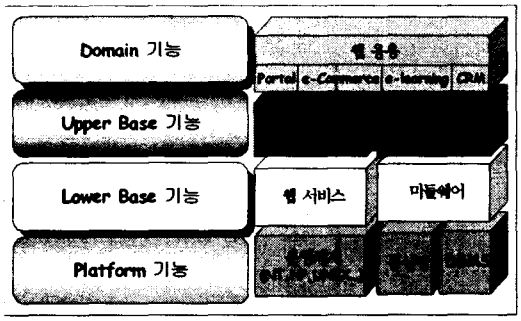
3.2 웹 서비스 애플리케이션 아키텍처

최근 웹의 발전으로 기존의 클라이언트/서버 방식은 대용량 대규모의 비즈니스 응용에서 서버 과부하와 다중 서버의 등장으로 인한 트랜잭션 관리의 문제점을 극복하기 힘들고, 시스템 확장시 클라이언트 프로그램의 유지보수가 매우 어렵다. 본 논문에서는 컴포넌트 기반의 웹 애플리케이션 아키텍처를 제안한다. 아키텍처는 표현 로직 부분은 클라이언트에, 비즈니스 로직과 데이터 접근 로직을 중간 계층과 서버 계층에서 구현하는 계층으로 분리된 형태를 따름으로써 시스템 운영의 유연성, 확장성 및 보안성을 보장할 수 있다. 특히 각 계층의 구현은 컴포



(그림 5) 웹 서비스 응용아키텍쳐

넌트를 사용하여 구현함으로써 다양한 시스템의 구성과 운영의 변화를 수용하고, 또한 클라이언트들은 무결성을 유지하면서 서버에 접근 가능하며, 서버는 확장이 용이하여 응용에 적합한 특징을 제공할 수 있다. 그림 5 는 웹 서비스 애플리케이션 아키텍처를 나타낸다. 웹 인프라 계층은 N-tier 아키텍처의 첫 번째 계층인 프리젠테이션 계층을 표현하고 방화벽, 브라우저 또는 모바일 장치에 표현하기 위한 Web Server 확장성을 위한 로드 발랜스를

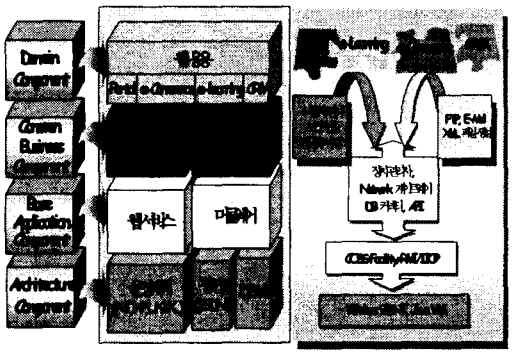


(그림 6) 웹 서비스 애플리케이션 구조

포함하고 웹 서비스에 대한 변경이나 관련된 표준은 포함하지 않는다. 웹 서비스 인프라 계층은 N-tier 아키텍처의 비즈니스 로직과 애플리케이션 서버에서 지원되는 객체를 포함하고 웹 서비스의 표준을 사용한다[6,7]. 이렇게 정의된 웹 서비스 애플리케이션 아키텍처를 기반으로 실제적인 응용 개발을 위해 필요로 하는 기술들의 계층이 정의되어져야 한다. 그림 6은 웹 서비스 애플리케이션 개발에 요구되어지는 기술들을 계층적으로 표현한 것이다. 이들 계층들은 웹 애플리케이션 개발에 필요한 기술의 적용 영역과 범위 및 구현 환경과의 제약 사항들을 기준으로 분류된다.

3.3 컴포넌트 식별 및 아키텍처 매핑

컴포넌트 기반 웹 서비스 애플리케이션 도메인 영역에서 식별된 컴포넌트이다. 식별된 단일 형태의 컴포넌트는 다양한 의미로 다른 응용에 사용될 수 있고, 또한 이들 컴포넌트끼리의 조립을 통해 새로운 응용을 생성할 수 있다. 따라서 실제 이들 컴포넌트는 웹 서비스 애플리케이션 개발을 위해 계층별로 커스터마이즈하여 조립, 확장될 수 있다. 웹 서비스 애플리케이션 개발을 위해 요구되는 기술의 계층적 구조와 참조 아키텍처에 기반하여 그림7과 같은 컴포넌트 계층을 정의하고 컴포넌트

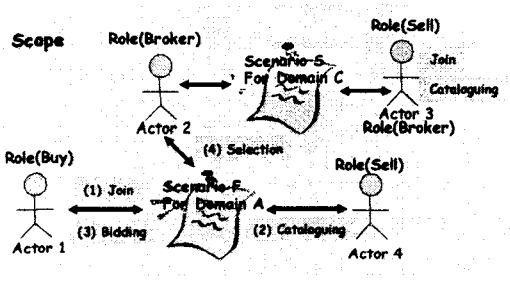


(그림 7) 보안 프레임워크 컴포넌트 매핑

의 매핑을 통해 조립을 위한 계층적 위상을 보여준다. 즉, 다양한 웹 서비스 애플리케이션 개발을 위해 컴포넌트를 식별하고 계층화함으로써 컴포넌트 조립을 위한 시나리오를 제시한다.

3.5. 모바일 응용서버모델링

다양한 웹 서비스 애플리케이션 개발 방법론이 등장하면서 그에 따른 설계 방법 또한 단순히 웹사이트의 GUI에 대한 설계에서 전체 논리적 설계를 위한 접근과 네비게이션 설계 및 동적인 웹사이트 지원을 위한데이터베이스 접근이 제공되고 있다. 하지만 이들 접근의 대부분은 단순히 UML의 기본적이고 단순한 표현을 지원하고 있고, 웹이라는 특성을 간과하고 단지 기존의 일반

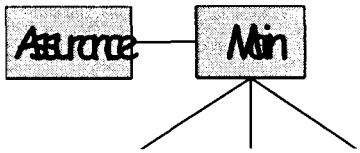


(그림 8) 유스케이스 시나리오 다이어그램

적인 애플리케이션의 특성만을 수용하고 있다. 따라서 본 논문의 웹 서비스 애플리케이션 모델링은 기존의 요구사항 분석 방법을 통해 분석되어진 결과를 바탕으로 전체 웹 서비스 애플리케이션의 유스케이스 시나리오 설계와 레이아웃 설계, 웹 서비스 협력 설계, 네비게이션 설계 및 페이지-서비스 구현 설계를 제시한다. 또한 이들 설계들은 UML의 확장 메카니즘을 사용하여 표현된다[8,9].

(1) 유스케이스 시나리오 설계 : 유스케이스는 웹 서비스 애플리케이션의 기능적인 요구사항을 정의하고, 무엇을 수행하는지에 대한 명확하고 일관성 있는 정의를 제공한다. 즉, 유스케이스를 사용하여 시스템 요소의 행위를 모델링함으로써 시스템에 대한 이해와 테스팅을 할 수 있는 기반을 제공한다.

(2) 레이아웃 설계 : 요구사항 분석 결과를 바탕으로 웹 서비스 애플리케이션의 구조를 설계하는 단계로 객체와 컴포넌트간의 관계와 이들간의 비즈니스 로직을 제공하는 인터페이스들을 모델링 한다. 즉 애플리케이션의 도메인 모델로 식별된 도메인 영역별 기능들을 구분하여 각 기능에 따른 수평, 수직 형태로 관계를 결정한다.



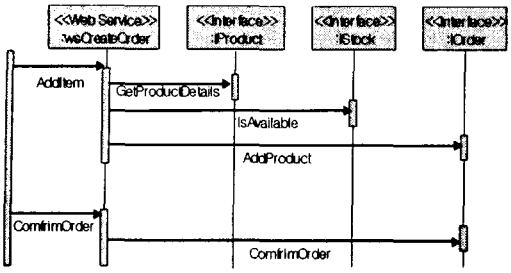
(그림 9) 레이아웃 다이어그램

(3) 웹서비스 협력 설계 : 레이아웃 설계를 기반으로 비즈니스 서비스와 웹 서비스를 식별하여 이들이 제공하는 인터페이스들을 시간적 순서대로 모델링 하고 기술한다. 시스템의 동적인 면을 표현하기 위해 상호작용을 이용하고 일련의 요소간에 교환되는 메시지로 구성되는 행위이다. 그들의 상호작용은 메시지의 시간 순서를 강조하고, 메시지
를 주고받는 요소의 구조적 구성을 표현한다.

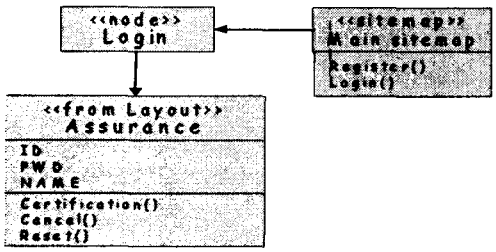
(4) 네비게이션 설계 : 웹 서비스 애플리케이션의 네비게이션 구조를 보여주는 것으로 페이지들 사이의 연결관계와 데이터 이동을 설계하는 단계로 네비게이션의 잠재적인 구조와 클래스와의 관계를 고려하여 네비게이션 트리를 생성한다.

(5) 페이지-서비스 구현 설계 : 웹 서비스 애플리케이션을 구성하는 페이지를 표현하고 웹 페이지를 상호 연결을 상세하게 설계하는 단계이다. 웹 서비스 애플리케이션 구성 요소는 웹 페이지와 일대일로 대응되고, 각 페이지의 기능에 따라 컴포넌트 형태의 표기를 사용하여 표현한다.

본 논문에서는 위의 설계를 기반으로 웹 서비스 협력(그림 10)과 네비게이션 설계(그림 11)을 제시한다.



(그림 10) 웹 서비스 협력 다이어그램



(그림 11) 네비게이션 다이어그램

3.6. J2EE 에서 모바일 응용서버구현

J2EE 아키텍처를 적용한 웹 어플리케이션은 네 가지로 구분할 수 있다[8].

1) 비 분산 아키텍처

- 비즈니스 컴포넌트 인터페이스를 사용하는 웹 어플리케이션
- 로컬 EJB에 접근하는 웹 어플리케이션

2) 분산 아키텍처

- 원격 EJB를 사용하는 분산 어플리케이션
- 웹 서비스 인터페이스를 공개하는 웹 어플리케이션

다음은 '원격 EJB를 사용하는 분산 어플리케이션'의 아키텍처를 보여준다.

이 아키텍처에서 Web Container의 객체와 EJB Container의 객체는 서로 원격 호출(RMI : Remote Method Invocation)을 하게 된다. 원격 호출은 객체를 네트워크로 전송하기 위해 마샬링/언마샬링을 해야 한다. 따라서, 원격 호출은 로컬 호출에 비해서 매우 느리다. 성능 향상을 위해서 원격 호출을 최대한 줄이기 위해 엔터프라이즈 빈은 큰 단위의 (Coarse-grained) 서비스를 제공하거나Web Container의 객체에서 원격 호출 참조를 저장(Cashing)함으로써 원격 호출을 줄일 수 있다. 이와 같은 성능 향상을 위한 전략을 패턴에서 찾을 수 있었다.

3.7 성능 향상을 위한 J2EE아키텍처 패턴

3.7.1 J2EE 패턴

패턴은 특정 상황(Context)의 반복되는 설계 문제에 대한 검증된 해결 방법이다[9]. 디자인 패턴은 성공적이고 증명된 설계와 아키텍처의 쉬운 재사용을 가능하게 하고, 새로운 시스템을 개발할 때 개발자들이 보다 쉽게 접근할 수 있게 한다. 현재 많은 디자인 패턴들이 나와 있는데, J2EE디자인 패턴에는 두 개의 큰 카테고리가 있다.

TheServerSide.com패턴[10]과 Sun Java Center의 J2EE 패턴[11]이다. 이 패턴 카테고리에 나온 패턴들 중에서 성능 개선을 위한

J2EE 패턴	TheServerSide.com 패턴
Service Locator	Service Locator
Session Façade	Session Façade
Transfer Object	Data Transfer Object

패턴은 다음과 같다.

1) Session Façade

네트워크 호출을 감소 시키기 위해서 큰 단위의(Coarse-grained) 인터페이스를 구현한다.

2) Service Locator

반복적인 JNDI 룩업(lookup)을 피하기 위해 InitialContext 객체의 참조를 저장하는 캐싱 메커니즘을 제공한다.

3) Transfer Object

모든 데이터를 직렬화 가능한 하나의 객체로 패키징(Packaging)함으로써 원격 호출의 수를 줄여준다.

3.7.2 성능향상을 위한 아키텍처 패턴

J2EE패턴 TrasferObject와 TheServerSide .com패턴의 DataTransfer Object는 유사한 패턴이므로 J2EE 패턴의 명칭을 사용한다. 따라서, 위 3가지 패턴의 조합으로 엔터프라이즈 어플리케이션을 만들 수 있는 두 가지의 아키텍처를 만들었다. 이는 패턴 간의 관계를 고려했다.

- 첫 번째 모델에 적용한 패턴

패턴명	종류
Session Façade	Façade
Transfer Object	POJO

- 두 번째 모델에 적용한 패턴

패턴명	종류
Session Façade	Façade
Service Locator	Singleton
Transfer Object	POJO

보안 프레임워크는 다양한 응용 플랫폼에서 독립적으로 보안 서비스를 제공할 수 있도록 설계개발 되었다. 그리고 기업 내 이미 존재하는 보안 시스템의 통합을 고려하여 JAAS를 지원하며, J2EE 보안 스펙지원, Kerberos, DB, LDAP, File 기반의 인증 서비스 지원, 역할 기반의 사용자 권한 관리, 인증된 사용자에 대한 정보 제공 기능을 제공하여, 다양한 응용의 보안 요구를 만족시킬 수 있는 기술들이 내포되어 있다.

보안 프레임워크는 모바일 응용서버 시스템 및 웹 응용서버 시스템에 적용되어 테스트 되었다. 보안 프레임워크의 모든 서비스는 보안 프레임워크의 보안 관리자에 등록되며, 응용별로 보안 도메인을 구성하여, 해당 보안 정책과 Realm을 관리하도록 하였다. 이를 위해, 인증 방법 및 역할에 따른 접근 제어 방법을 환경 파일에 기술하여, 응용 구축자가 원하는 형태로 보안 서비스가 이루어지도록 하였다.

(1) Add-on Module

보안 프레임워크의 모바일 응용 서버는 JSP와 Servlet을 지원하기 위해 Tomcat 서버와 함께 구성되는데, Java Servlet 스펙[2]에서 요구하는 보안 요구 사항을 만족시키기 위해, Tomcat의 보안 서비스와 연동하여야 하여야 한다. 보안 프레임워크는 이를 위해, Tomcat에 Add-on 되는 모듈들을 제공하여, 선언적 보안을 제공하기 위한 Security Realm 어댑터와 명령적 보안을 제공하기 위한 Security Valve를 둘 수 있도록 하였다.

Security Realm 어댑터는 보안 프레임워크 내에 등록된 Realm를 선언적인 방법으로Tomcat과 연동하도록 구현 하였으며, Security Valve는 응용 로직 상에서 구현된 보안 로직이 Tomcat의 보안 서비스와 연동할 수 있도록 세션과 사용자 Subject를 사용하여 구현하였다.

(2) Security Realms

Security Realm 은 하나의 공통된 보안 체계 안에 속하게 되는 사용자들과 그룹, 역할 등의 목록을 구성하고 있는 단위로, 시스템에 접근하려는 사용자가 인증된 사용자인지를 확인하고 해당 사용자가 어떠한 그룹에 속하며 어떤 역할을 가지고 있는지, 또 사용자나 그룹이 어떤 자원에 접근이 가능한지를 판단할 수 있도록 하였다. 또한 인증된 사용자의 기본 정보를 제공하여 응용 개발자가 사용자 정보에 접근할 수 있도록 하였다. 아래 텍스트 상자는 'WebDomain'이라는 이름으로 보안 도메인을 구성한 예제이며, 보안 도메인 내에 JDBCRealm을 두어, Oracle 데이터베이스에 사용자 그룹 및 역할들에 대해 정의하고 사용자 기본 정보를 관리하도록 하였다.

(3) Configurations

보안 환경 파일은 응용 서버의 전체 보안 정책, 도메인 별 보안 정책, 응용 별 보안 도메인 구성 방법 등을 기술하며, 각각 server.xml, security.policy, web-security .xml 파일로 저장되는데, 보안 관리 도구를 사용하여, 응용 구축자가 직접 관리할 수 있도록 하였다. 보안 관리 도구는 현재, 시스템 및 응용을 위한 환경 설정 파일을 생성.기술하는 도구로서 구현

되었으며, 사용자.그룹 관리에 필요한 기능은 제공되지 않는다. 단, 보안 프레임워크는 사용자.그룹 정보를 응용에서 활용할 수 있도록 User 인터페이스를 제공한다.

(4) Session

사용자 세션의 관리는 응용서버 별로 다양한 형태로 제공될 수 있지만, 기본적으로 인증된 사용자 정보가 그 사용자 세션에 등록되어 응용서버의 보안 서비스에 접근할 수 있는 방법을 제공할 것이다. 본 논문에서 설명하는 보안 프레임워크는 Add-on 모듈을 통해, 사용자 세션에, 인증된 사용자의 보안 정보를 등록하도록 하였다. 세션에 등록된 인증 정보는 응용 개발자나 시스템에 의해 활용할 수 있으나 정보의 변경 및 읽기에 제약을 둘 수 있도록 하였다. 그리고 시간 제한에 의한 세션의 파기 시에, 세션 리스너(Listener)와 연동하여, 인증된 사용자는 자동적으로 로그아웃처리 되도록 하였다.

(5) Cluster

응용서버가 클러스터링을 지원하는 시스템에서는 각각의 클러스터 노드가 인증된 사용자에 대한 인증 정보와 세션을 멀티캐스트 프레임워크를 통해 공유하고 있어야 한다. 보안 프레임워크는 인증된 사용자가 클러스터링 환경에서 또다시 인증 작업을 수행하지 않게 하기 위해, 쿠키 또는 커버로스 티켓을 사용하여 단일 인증을 지원하고 있다. 또한 각각의 클러스터 노드들에 대한 상호 인증 및 전송 데이터 보안을 위한 API를 제공하여, 클러스터링 환경에서의 보안을 지원한다.

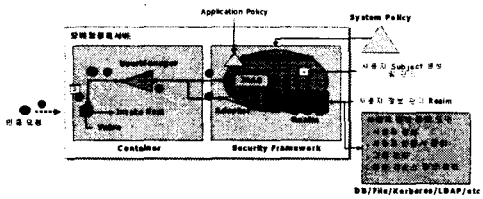
(6) SSO(Single Sign-On)

보안 프레임워크는 SSO를 제공하기 위해, 쿠키 또는 커버로스를 사용하고 있는데, 쿠키의 보안 데이터 설정은 사용자 관리 시스템에서 이루어지며, 서버 시스템의 보안 키로 암호화되어 처리된다.

사용자 인증과 권한 관리는 보안 프레임워크에서 제공하는 기본적인 보안 서비스로, Java2 Security Manager를 통해 소스 기반 접근 제어를 담당하도록 하였으며, JAAS를 통한 인증 및 역할 기반 보안, ACL을 통한 접근 제어, 등으로 구성되어 인증 및 권한 관리를 수행하도록 하였다.

3.8 사용자 인증 및 권한 관리

웹 응용 서버 시스템에서 제공하는 가장 기본적인 인증 방법은 브라우저가 사용자 신원 정보를 수집하여 웹 응용 서버에 이를 전달하는 선언적 인증 방법이다. 선언적 인증은 Basic, Form-based, Client-certificate, Digest 인증 방법이 있으며, 브라우저가 사용자 신원 정보를 수집하는 방법과 웹 컨테이너에 신원 정보를 전달하는 방법에 따라 분류된다. 선언적 인증은 응용 개발자가 선택할 수 있는 가장 쉬운 인증 방법이지만, 응용에서 필요한 완벽한 보안 체계를 제공하지는 않는다. 본 보안 프레임워크에서는 선언적 인증을 위해, 웹 응용 서버에 전달된 사용자 인증 정보를 Add-on 모듈에서 가로채, 보안 프레임워크를 통해 인증 받도록 하였으며, 이러한 방법으로 보안 프레임워크에서 제공하는 다양한 인증 방법을 사용할 수 있도록 하였다. 그리고 개발자가 직접 응용 로직 상에서 인증을 제어할 수 있도록 하기 위해, 프로그램 인증에 필요한 API를 제공되었다.



(그림 11) 사용자 인증 절차

JAAS(Java Authentication and Authorization Service)는 사용자 인증 및 권한 할당을 위한 프레임워크와 표준 프로그래밍 인터페이스를 제공하며, 사용자 기반 인증 및 권한 부여를 지원하도록 Java2 플랫폼의 액세

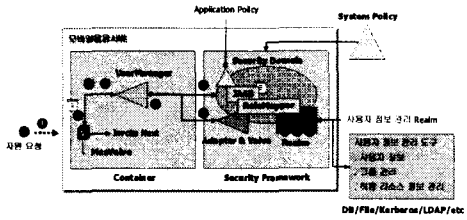
스 제어 구조를 확장한 PAM(Pluggable Authentication Module) 표준을 기반으로 한다. 보안 프레임워크는 JAAS를 완벽히 지원하며, 기본적으로 Kerberos, DB, 인증서 기반의 인증을 위한 LoginModule을 제공한다. 또한 LoginModule SPI(Service Provider Interface)를 통해 특정 LoginModule를 제공하는 보안 서비스 제공자가 보안 프레임워크와 연동할 수 있는 방법을 제공한다.

그림 11은 모바일 응용 서버 시스템에서 사용자 인증을 처리하는 과정을 설명한 개괄적인 흐름도이다. 선언적으로 보호된 모바일 응용 서버의 자원에 접근하려는 사용자가 있을 때, 응용 로직 또는 모바일 응용 서버는 사용자에게 인증을 요구하게 된다. 사용자 인증 정보가 모바일 응용 서버에 접수되면, Tomcat Valve 및 모바일 사용자 관리자를 거쳐, 보안 프레임워크로 전달된다.

전달된 인증 정보는 초기 응용의 배포 시에 기술한 인증 방법을 통해 사용자 인증을 수행하고 그 결과를 되돌려 준다.

인증된 사용자의 인증 정보는 사용자의 세션에 등록되어 관리되는데, 인증 정보는 사용자의 Subject 객체로 표현된다. Subject 객체에 대한변경은 인증이 일어나는 시점에만 가능하며, 특별한 상황에서는 보안 프레임워크를 개시한 시스템Subject에 의해서만 변경이 가능하도록 하였다. 사용자가 응용서버의 자원 및 서비스에 접근하기 위해서는 해당 자원 및 서비스에 대한 권한을 가지고 있어야 한다. 또한 응용서버의 보안을 위해서, 응용 개발자의 코드가 불법적으로 시스템 자원에 접근하는 것을 방지하도록 설계되어야 한다. 보안 프레임워크는 이러한 권한 관리 및 접근제어를 위해, JAAS 기반 하의 인증된 사용자의 Principal을 통한 역할 기반 보안, 자원에 대한 ACL(Access Control List) 관리 서비스를 제공하며, 코드 기반의 접근 보안을 위해 Java의 Security Manager를 이용한다.

응용 개발자는 웹 서버의Request 객체에 대한 *isUserInRole* 함수를 통해 기본적인 접근 제어를 수행할 수 있으며, 보안 프레임워크



(그림 12)모바일 응용서버자원 처리 흐름도

에서 제공하는 권한 관리 서비스 API를 통해 보다 정밀한 접근 제어 로직을 개발할 수 있다. 그림12 는 모바일 응용서버 시스템에서 사용자가 어떤 모바일 응용서버 자원 또는 서비스를 요청하였을 경우에 이를 처리하는 과정을 설명한 개괄적인 흐름도이다. 선언적으로 보호된 모바일 응용 서버의 자원에 접근하려는 사용자가 있을 때, 먼저 인증 과정을 요구하며, 인증이 완료된 이후에 사용자의 Subject 객체를 통해 사용자 Principal 혹은 Group Principal을 가져오며, 이를 보안 프레임워크에 전달하여 JAAS 혹은ACL를 통해 사용자 접근 권한을 체크하고, 접근 권한을 가진 사용자에게 자원 및 서비스에 대한 접근을 허용한다.

응용 로직 상에서 접근 제어를 수행하기 위해서, 보안 프레임워크는 응용이 사용하는 Realm에 사용자 Principal과 요구되는 역할을 기반으로 접근 권한을 검사하는 함수를 제공하고 있다.

6. 결론 및 향후 연구

응용서버 시스템은 응용이 필요로 하는 다양한 기능들을 제공하여야 하며, 그 중 보안 기능은 인터넷 환경과 같은 공개형 네트워크에서 서비스를 제공하고자 하는 응용에서는 필수적으로 요구된다. 웹 응용이 제공하는 서비스는 다양하기 때문에 그에 따른 보안의 강도와 시스템 또한 다양하며, 응용 서버 내 보안 프레임워크는 이를 만족할 수 있는 기능을 제공하여야 한다. 또한 기존 레거시 보안 시스템과 연동할 수 있는 방법을 제공하여야 한다.

최근 컴포넌트를 기반한 웹 서비스 응용 시스템 개발이 필요하게 되면서 표준 기술, 체계적 명세와 모델링 그리고 통합에 대한 연구가 추가적으로 요구되고 있다. 즉, 웹 서비스 애플리케이션 개발에 대한 사용자의 의도를 융통성 있게 수용하고, 실질적인 재사용 요소로서 뿐 아니라 응용으로의 생성과 운용을 위한 생산성을 CBD를 통해 얻을 수 있다.

본 논문에서는 레거시 보안 시스템과의 연동, J2EE 보안 지원, JAAS 지원, Kerberos 지원 등 응용 서버 시스템들이 요구하는 다양한 보안 기능을 제공하도록 컴포너트 기반 보안 프레임워크를 개발한다. 보안 프레임워크는 현재 모바일응용서버 및 웹 응용 서버 시스템에서 테스트되었으며, 커버로스 서버와 통신하는 통신 모듈을 개발하여 보다 강화된 인증 서비스가 가능하도록 하였다. 이후 시스템의 활용도를 높이기 위하여, 보안 프레임워크에서 요구하는 보안 환경 설정을 응용 구축자가 편리하게 작성할 수 있도록 응용서버 구축 도구와 연계한 보안 환경 설정 및 사용자 정보 등록 도구를 개발할 필요가 있다. 고수준의 컴포넌트 재사용을 통해 실질적인 애플리케이션 개발 표준화와 생산성 향상을 목표로 한다.

참고 문헌

[1] 정부연, "웹 서비스의 현황 및 비즈니스 모델의 변화", 정보통신정책, 제14권 15호, 2002

[2] David Sprott, Componet AND Services, CBDi forum, Insight for Web Service and Software Component Practice, 2002

[3] 김행곤 외, 컴포넌트 저장소 형상 관리 시스템 개발, ETRI 최종보고서, 2000

[4] Rasesh Trivedi, "Web Services Architecture Model", RCG Information Technology, 2002

[5] Blake, M.B., An agent-based ross-organizational workflow architecture in support of Web services" WET ICE 2002 Proceeding, Eleventh IEEE International Workshops, pp.176~181, 2002

[6] David Piper, "Web Services Conceptualization to Design", SELECT Business Solutions, 2002

[7] Jim Conallen, "Modeling Web Application with UML", Conallen, Inc., 1999

[8] "Java2 Platform Enterprise Edition Specification, Version 1.4"

[9] "Java Servlet Specification Version 2.4"

[10] 김수형, 이경호, 김중배, "Clustered EJB서의 멀티캐스트 보안 연구," 정보과학회 2003 춘계 학술대회

[11] "Java Authentication and Authorization Service(JAAS)," http://java.sun.com/products/jaas/