(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0024014
(43) 공개일자 2017년03월06일

(51) 국제특허분류(Int. Cl.)
G06F 21/62 (2013.01) *G06F 17/30* (2006.01)
H04L 12/911 (2013.01) *H04L 29/06* (2006.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
G06F 21/62 (2013.01)
G06F 17/30584 (2013.01)
(21) 출원번호 10-2017-7002052
(22) 출원일자(국제) 2015년06월23일
심사청구일자 없음
(85) 번역문제출일자 2017년01월23일
(86) 국제출원번호 PCT/US2015/037270
(87) 국제공개번호 WO 2015/200379
국제공개일자 2015년12월30일
(30) 우선권주장
62/016,058 2014년06월23일 미국(US)
62/054,912 2014년09월24일 미국(US)

(71) 출원인
오라클 인터내셔날 코포레이션
미국, 캘리포니아 94065, 레드우드 시티, 오라클 파크웨이 500
(72) 발명자
홉킨스 윌
미국 94065 캘리포니아 레드우드 쇼어스 엠/에스 5오피7 오라클 파크웨이 500
페레즈 크래이그
미국 94065 캘리포니아 레드우드 쇼어스 엠/에스 5오피7 오라클 파크웨이 500
(뒷면에 계속)
(74) 대리인
박장원

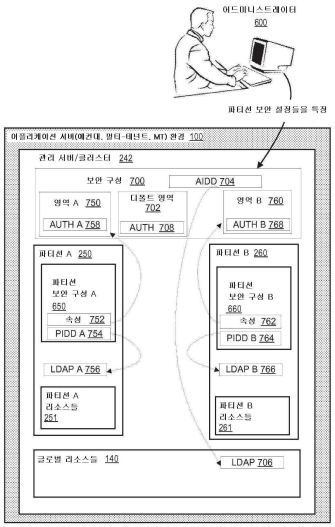
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **멀티테넌트 어플리케이션 서버 환경에서 보안을 지원하는 시스템 및 방법**

(57) 요 약

하나의 실시예에 따르면, 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 시스템 및 방법이 본 명세서에 기술된다. 하나의 실시예에 따르면, 파티션 당 보안 구성(per-partition security configuration)은, 파티션 당 보안 영역(per-partition security realm)(인증, 인가(authorization), 크리덴셜 매핑, 감사(auditing), 패스워드 검증, 증명서 검증 및 사용자 락아웃에 대한 구성을 포함), SSL 구성(키들, 증명서들 및 다른 구성 속성들 포함), 및 파티션 및 글로벌 리소스들에 대한 액세스 제어를 포함한다. 어드미니스트레이터(administrator)는 룰들의 허여(grant)를 통해 하나 이상의 파티션 사용자들을 파티션 어드미니스트레이터들로서 지정할 수 있다.

대 표 도 - 도7



(52) CPC특허분류
G06F 21/604 (2013.01)
H04L 47/70 (2013.01)
H04L 63/08 (2013.01)
H04L 63/105 (2013.01)
H04L 67/02 (2013.01)
H04L 67/10 (2013.01)
G06F 2221/2141 (2013.01)
(72) 발명자
가이 데이비드
미국 94065 캘리포니아 레드우드 쇼어스 엠/에스 5오피7 오라클 파크웨이 500
보워 피터
미국 03049 뉴 햄프셔 홀리스 사우스 게이트 로드 20
리 주안
미국 02494 매사추세츠 니드햄 데이비드 로드 22
탄칠 제프
미국 94065 캘리포니아 레드우드 쇼어스 엠/에스 5오피7 오라클 파크웨이 500
스리람마드헤시칸 크리쉬나
미국 95014 캘리포니아 쿠퍼티노 파리쉬 플래이스 10163

명 세 서

청구범위

청구항 1

복수의 파티션들, 복수의 파티션 리소스들 및 복수의 글로벌 리소스들을 포함하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법으로서,

어드민 보안 영역(admin security realm), 제1 보안 영역 및 제2 보안 영역을 포함하는 복수의 보안 영역들을 정의하는 단계와;

상기 복수의 파티션 리소스들 중 복수의 제1 파티션 리소스들을 가지도록 상기 복수의 파티션들 중 제1 파티션을 구성하는 단계와;

상기 복수의 파티션 리소스들 중 복수의 제2 파티션 리소스들을 가지도록 상기 복수의 파티션들 중 제2 파티션을 구성하는 단계와;

상기 제1 보안 영역과 상기 제1 파티션을 관련시키는 제1 보안 구성을 제공하는 단계와;

상기 제2 보안 영역과 상기 제2 파티션을 관련시키는 제2 보안 구성을 제공하는 단계와;

상기 제1 파티션과 제1 프라이머리 신원 도메인(primary identity domain)을 관련시키는 단계 - 상기 제1 프라이머리 신원 도메인은 제1 테넌트와 관련된 복수의 제1 사용자들을 나타냄 - 와;

상기 제2 파티션과 제2 프라이머리 신원 도메인을 관련시키는 단계 - 상기 제2 프라이머리 신원 도메인은 제2 테넌트와 관련된 복수의 제2 사용자들을 나타냄 - 와;

상기 복수의 파티션 리소스들 및 상기 복수의 글로벌 리소스들에게로의 액세스에 대한 인증(authentication) 및 인가(authorization)를 제어하기 위해 상기 어드민 보안 영역, 제1 보안 영역 및 제2 보안 영역 각각을 런타임 시 동시에 동작시키는 단계를 포함하고,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들이 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지되, 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지지 않고,

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들은 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지되, 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지지 않는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 2

제1항에 있어서,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들의 제1 표시를 저장하기 위해 제1 신원 저장소를 참조하도록 상기 제1 프라이머리 신원 도메인을 구성하는 단계와; 그리고

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들의 제2 표시를 저장하기 위해, 상기 제1 신원 저장소와는 다른 제2 신원 저장소를 참조하도록 상기 제2 프라이머리 신원 도메인을 구성하는 단계를 더 포함하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 3

제1항에 있어서,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들의 제1 표시를 저장하기 위해 신원 저장소의 제1 부분을 참조하도록 상기 제1 프라이머리 신원 도메인을 구성하는 단계와; 그리고

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들의 제2 표시를 저장하기 위해 상기 신원 저장소의 제2 부분을 참조하도록 상기 제2 프라이머리 신원 도메인을 구성하는 단계를 더 포함하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 4

선행하는 어느 청구항에 있어서,

상기 멀티테넌트 어플리케이션 서버 환경과 어드민 신원 도메인을 관련시키는 단계를 더 포함하고, 상기 어드민 신원 도메인은 상기 멀티테넌트 어플리케이션 서버 환경의 복수의 시스템 어드미니스트레이터들을 나타내고,

상기 멀티테넌트 어플리케이션 서버 환경과 관련된 상기 복수의 시스템 어드미니스트레이터들은 상기 복수의 글로벌 리소스들에게로의 액세스를 가지는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 5

선행하는 어느 청구항에 있어서,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들을 인증하고 그리고 상기 복수의 제1 사용자들 중 하나 이상과 조합하여 상기 제1 프라이머리 신원 도메인을 식별하는 제1 서명된 프린서플들(principals)을 생성하도록 구성된 제1 인증 서비스를 제공하는 단계를 더 포함하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 6

선행하는 어느 청구항에 있어서,

상기 제1 프라이머리 신원 도메인과 상기 복수의 제1 리소스들 각각을 관련시키는 단계와;

상기 제2 프라이머리 신원 도메인과 상기 복수의 제2 리소스들 각각을 관련시키는 단계와;

인가 서비스를 제공하는 단계를 포함하고, 상기 인가 서비스는 리소스에 액세스하기 위해 사용자로부터 콜(call)을 수신함에 응답하여, 상기 사용자와 관련된 프라이머리 신원 도메인을 상기 리소스와 관련된 프라이머리 신원 도메인과 비교하고 그리고 상기 사용자와 관련된 프라이머리 신원 도메인이 상기 리소스와 관련된 프라이머리 신원 도메인과 매치되는 경우에만 상기 리소스에게로의 액세스를 인가하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 7

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들을 인증하고 그리고 상기 복수의 제1 사용자들 중 하나 이상과 조합하여 상기 제1 프라이머리 신원 도메인을 식별하는 제1 서명된 프린서플들을 생성하도록 구성된 제1 인증 서비스를 제공하는 단계와;

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들을 인증하고 그리고 상기 복수의 제2 사용자들 중 하나 이상과 조합하여 상기 제2 프라이머리 신원 도메인을 식별하는 제2 서명된 프린서플들을 생성하도록 구성된 제2 인증 서비스를 제공하는 단계와;

상기 제1 프라이머리 신원 도메인과 상기 복수의 제1 리소스들 각각을 관련시키는 단계와;

상기 제2 프라이머리 신원 도메인과 상기 복수의 제2 리소스들 각각을 관련시키는 단계와;

인가 서비스를 제공하는 단계를 포함하고, 상기 인가 서비스는 리소스에 액세스하기 위해 프린서플과 관련된 콜을 수신함에 응답하여, 상기 프린서플에서 식별된 프라이머리 신원 도메인을 상기 리소스와 관련된 프라이머리 신원 도메인과 비교하고 그리고 상기 프린서플과 관련된 프라이머리 신원 도메인이 상기 리소스와 관련된 프라이머리 신원 도메인과 매치되는 경우에만 상기 리소스에게로의 액세스를 인가하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 방법.

청구항 8

머신 판독가능 포맷의 프로그램 명령어들을 포함하는 컴퓨터 프로그램으로서, 상기 프로그램 명령어들은 컴퓨터 시스템에 의해 실행될 때, 상기 컴퓨터 시스템으로 하여금 청구항 제1항 내지 제7항 중 어느 한 항의 방법을 수행하도록 하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 9

비일시적 머신 판독가능 데이터 저장 매체에 저장된 제8항의 컴퓨터 프로그램을 포함하는 컴퓨터 프로그램.

청구항 10

복수의 파티션 리소스들 및 복수의 글로벌 리소스들을 갖는 복수의 파티션들을 포함하는 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하기 위한 명령어들이 저장된 비일시적 컴퓨터 판독가능 매체로서, 명령어들은 실행될 때, 시스템으로 하여금 단계들을 수행하도록 하며, 상기 단계들은:

어드민 보안 영역, 제1 보안 영역 및 제2 보안 영역을 포함하는 복수의 보안 영역들을 정의하는 단계와;

상기 복수의 파티션 리소스들 중 복수의 제1 파티션 리소스들을 가지도록 상기 복수의 파티션들 중 제1 파티션을 구성하는 단계와;

상기 복수의 파티션 리소스들 중 복수의 제2 파티션 리소스들을 가지도록 상기 복수의 파티션들 중 제2 파티션을 구성하는 단계와;

상기 제1 보안 영역과 상기 제1 파티션을 관련시키는 제1 보안 구성을 제공하는 단계와;

상기 제2 보안 영역과 상기 제2 파티션을 관련시키는 제2 보안 구성을 제공하는 단계와;

상기 제1 파티션과 제1 프라이머리 신원 도메인을 관련시키는 단계 - 상기 제1 프라이머리 신원 도메인은 제1 테넌트와 관련된 복수의 제1 사용자들을 나타냄 - 와;

상기 제2 파티션과 제2 프라이머리 신원 도메인을 관련시키는 단계 - 상기 제2 프라이머리 신원 도메인은 제2 테넌트와 관련된 복수의 제2 사용자들을 나타냄 - 와;

상기 복수의 파티션 리소스들 및 상기 복수의 글로벌 리소스들에게로의 액세스에 대한 인증 및 인가를 제어하기 위해 상기 어드민 보안 영역, 제1 보안 영역 및 제2 보안 영역 각각을 런타임 시 동시에 동작시키는 단계를 포함하고,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들이 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지되, 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지지 않고,

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들은 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지되, 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지지 않는 것을 특징으로 하는 비일시적 컴퓨터 판독가능 매체.

청구항 11

멀티테넌트 어플리케이션 서버 환경 시스템으로서,

복수의 마이크로프로세서들 및 메모리를 포함하는 어플리케이션 서버 환경과;

상기 어플리케이션 서버 환경 상에 구성된 복수의 파티션들과;

상기 어플리케이션 서버 환경에 제공되는 복수의 파티션 리소스들 및 복수의 글로벌 리소스들과;

상기 어플리케이션 서버 환경에서 구성된 어드민 보안 영역, 제1 보안 영역 및 제2 보안 영역을 포함하는 복수의 보안 영역들과;

상기 복수의 파티션 리소스들 중 복수의 제1 파티션 리소스들을 가지도록 구성된 상기 복수의 파티션들 중 제1 파티션과;

상기 복수의 파티션 리소스들 중 복수의 제2 파티션 리소들을 가지도록 구성된 상기 복수의 파티션들 중 제2 파티션과;

상기 제1 보안 영역과 상기 제1 파티션을 관련시키는 제1 보안 구성과;

상기 제2 보안 영역과 상기 제2 파티션을 관련시키는 제2 보안 구성과;

상기 제1 파티션과 관련된 제1 프라이머리 신원 도메인 - 상기 제1 프라이머리 신원 도메인은 제1 테넌트와 관

련된 복수의 제1 사용자들을 나타냄 - 과;

상기 제2 파티션과 관련된 제2 프라이머리 신원 도메인을 포함하고, 상기 제2 프라이머리 신원 도메인은 제2 테넌트와 관련된 복수의 제2 사용자들을 나타내고,

상기 어드민 보안 영역, 제1 보안 영역 및 제2 보안 영역은 상기 복수의 파티션 리소스들 및 상기 복수의 글로벌 리소들에게로의 액세스에 대한 인증 및 인가를 제어하기 위해 런타임 시 동시에 동작하도록 구성되고,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들이 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지되, 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지지 않고,

상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들은 상기 제2 파티션의 상기 복수의 제2 파티션 리소스들에게로의 액세스를 가지되, 상기 제1 파티션의 상기 복수의 제1 파티션 리소스들에게로의 액세스를 가지지 않는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

청구항 12

제11항에 있어서,

상기 제1 프라이머리 신원 도메인은 상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들의 제1 표시를 저장하기 위해 제1 신원 저장소를 참조하도록 구성되고; 그리고

상기 제2 프라이머리 신원 도메인은 상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들의 제2 표시를 저장하기 위해, 상기 제1 신원 저장소와는 다른 제2 신원 저장소를 참조하도록 구성된 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

청구항 13

제11항에 있어서,

상기 제1 프라이머리 신원 도메인은 상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들의 제1 표시를 저장하기 위해 신원 저장소의 제1 부분을 참조하도록 구성되고; 그리고

상기 제2 프라이머리 신원 도메인은 상기 제2 테넌트와 관련된 상기 복수의 제2 사용자들의 제2 표시를 저장하기 위해 상기 신원 저장소의 제2 부분을 참조하도록 구성된 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

청구항 14

제11항 내지 제13항 중 어느 한 항에 있어서,

상기 멀티테넌트 어플리케이션 서버 환경과 관련된 어드민 신원 도메인을 더 포함하고, 상기 어드민 신원 도메인은 상기 멀티테넌트 어플리케이션 서버 환경의 복수의 시스템 어드미니스트레이터들을 나타내고; 그리고

상기 멀티테넌트 어플리케이션 서버 환경과 관련된 상기 복수의 시스템 어드미니스트레이터들은 상기 복수의 글로벌 리소스들에게로의 액세스를 가지는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

청구항 15

제11항 내지 제14항 중 어느 한 항에 있어서,

상기 제1 테넌트와 관련된 상기 복수의 제1 사용자들을 인증하고 그리고 상기 복수의 제1 사용자들 중 하나 이상과 조합하여 상기 제1 프라이머리 신원 도메인을 식별하는 제1 서명된 프린서플들을 생성하도록 구성된 제1 인증 서비스를 제공하도록 구성된 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

청구항 16

제11항 내지 제15항 중 어느 한 항에 있어서,

상기 복수의 제1 리소스들은 상기 제1 프라이머리 신원 도메인과 관련되고;

상기 복수의 제2 리소스들은 상기 제2 프라이머리 신원 도메인과 관련되고;

상기 시스템은 인가 서비스를 더 포함하고, 상기 인가 서비스는 리소스에 액세스하기 위해 사용자로부터 콜을 수신함에 응답하여, 상기 사용자와 관련된 프라이머리 신원 도메인을 상기 리소스와 관련된 프라이머리 신원 도메인과 비교하고 그리고 상기 사용자와 관련된 프라이머리 신원 도메인이 상기 리소스와 관련된 프라이머리 신원 도메인과 매치되는 경우에만 상기 리소스에게로의 액세스를 인가하는 것을 특징으로 하는 멀티테넌트 어플리케이션 서버 환경 시스템.

발명의 설명

기 술 분 야

[0001] 저작권 공지

[0002] 본 명세서에서 개시된 부분은 저작권 보호를 받는 내용을 포함한다. 저작권자는 미국특허상표청의 특허 파일 또는 기록에 나타난 대로 본 특허 문서 또는 특허 개시내용을 어느 누군가가 팩시밀리 재생하는 것은 반대하지 않지만, 그 밖의 모든 것은 저작권으로 보호된다.

[0003] 기술분야

[0004] 본 발명의 실시예들은 일반적으로, 어플리케이션 서버들 및 클라우드 플랫폼 환경들에 관한 것이며, 특히 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 시스템 및 방법에 관한 것이다.

배 경 기 술

[0005] 소프트웨어 어플리케이션 서버들 - 이들의 예들은 오라클 웹로직 서버(WLS) 및 Glassfish를 포함함 - 은 일반적으로, 기업 소프트웨어 어플리케이션들을 실행하기 위한 관리 환경을 제공한다. 최근, 클라우드 환경들에서 이용하기 위한 기술들이 또한 개발되어 왔는 바, 이 기술들은 사용자들 또는 테넌트들이 상기 클라우드 환경 내에서 자신의 어플리케이션들을 개발 및 실행할 수 있게 하고 상기 환경에 의해 제공되는 분산 리소스들의 장점을 취할 수 있게 한다.

발명의 내용

[0006] 하나의 실시예에 따르면, 멀티테넌트 어플리케이션 서버 환경에서 보안을 제공하는 시스템 및 방법이 본 명세서에 기술된다. 하나의 실시예에 따르면, 파티션 당 보안 구성(per-partition security configuration)은, 파티션 당 보안 영역(per-partition security realm)(인증(authentication), 인가(authorization), 크리덴셜 매핑, 감사(auditing), 패스워드 검증, 증명서 검증 및 사용자 락아웃에 대한 구성을 포함), SSL 구성(키들, 증명서들 및 다른 구성 속성들 포함), 및 파티션 및 글로벌 리소스들에 대한 액세스 제어를 포함한다. 어드미니스트레이터(administrator)는 룰들의 허여(grant)를 통해 하나 이상의 파티션 사용자들을 파티션 어드미니스트레이터들로서 지정할 수 있다.

도면의 간단한 설명

[0007] 도 1은 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시(multi-tenancy)를 지원하는 시스템을 도시한다.

도 2는 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하는 시스템을 더 도시한다.

도 3은 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하는 시스템을 더 도시한다.

도 4는 하나의 실시예에 따른 예시적인 리소스 그룹 템플릿의 이용을 도시한다.

도 5는 하나의 실시예에 따른 예시적인 멀티-테넌트 환경을 도시한다.

도 6은 하나의 실시예에 따른 멀티-테넌트 환경에서 보안을 지원하는 시스템 및 방법의 일반적인 기능을 도시한다.

도 7은 하나의 실시예에 따른 복수의 영역들(realms)의 이용을 위해 구성된 멀티-테넌트 환경을 도시한다.

도 8은 하나의 실시예에 따른 복수의 신원 도메인들을 갖는 멀티-테넌트 환경에서 로그인 모듈들에 의해 구현되

는 인증 방법을 도시한다.

도 9는 하나의 실시예에 따른 복수의 신원 도메인들을 갖는 멀티-테넌트 환경에서 구현되는 인가 서브시스템을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 하나의 실시예에 따르면, 본 발명은 멀티-테넌트 환경에서 인증, 인가, 크리덴셜 매핑, 감사, 패스워드 검증, 증명서 검증 및 사용자 락아웃에 대한 구성을 포함하는 보안 서비스들의 파티션 당 구성을 제공한다. 본 발명의 실시예들은 또한, 특정한 파티션에 전개된 어플리케이션들이 상기 특정한 파티션의 사용자들에게만 액세스가능하게 하고, 특정한 파티션에 대한 파티션 구성이 오직, 상기 특정한 파티션의 어드미니스트레이터(및 웹로직 어드미니스트레이터)에게 이용가능하게 하고, 그리고 파티션 어드미니스트레이터들이 오직, 글로벌하게 가시적인 리소스들 및 구성에 대한 판독 전용 액세스를 가지도록, 파티션 및 글로벌 리소스들에 대한 액세스 제어를 제공한다. 도 1 내지 5 및 첨부 텍스트는 멀티-테넌트 어플리케이션 서버 환경을 기술한다. 도 6 내지 8 및 첨부 텍스트는 예컨대, 도 1 내지 5에 관하여 기술되는 바와 같이 멀티-테넌트 서버 환경들에서 보안을 지원하는 시스템 및 방법을 기술한다.

[0009] 다음의 설명에서, 본 발명은 첨부 도면들의 도해들에서 제한이 아닌 예로서 예시된다. 본 명세서에서 다양한 실시예에 대한 참조들은 반드시 동일한 실시예에 대한 것은 아니며, 이러한 참조들은 적어도 하나를 의미한다. 특정한 구현들이 논의되지만, 이는 단지 예시적인 목적들로 제공됨이 이해된다. 이 기술 분야의 숙련자는 다른 컴포넌트들 및 구성들이 본 발명의 범위 및 사상으로부터 벗어남이 없이 이용될 수 있음을 인지할 것이다.

[0010] 더욱이, 특정 예들에서, 다수의 특정한 세부사항들이 본 발명의 철저한 설명을 제공하기 위해 제시될 것이다. 그러나, 본 발명이 이러한 특정한 세부사항들이 없이도 실시될 수 있음이 이 기술 분야의 숙련자들에게 분명해질 것이다. 다른 예들에서, 잘 알려진 특징들은 본 발명을 모호하게 하지 않기 위해 매우 세부적으로 기술되지 않는다.

[0011] 공통적인 도면 번호들은 도면들 및 상세한 설명 전반에서 유사한 요소들을 나타내기 위해 이용되며, 그러므로, 도해에서 이용되는 도면 번호들은 요소가 다른 곳에서 기술되는 경우 그러한 도해에 특정적인 상세한 설명에서 참조될 수 있거나 혹은 참조되지 않을 수 있다. 세 자리 도면 번호 중 첫째 자리는 요소가 처음 나타나는 도해들의 시리즈를 나타낸다.

[0012] 어플리케이션 서버(예컨대, 멀티-테넌트, MT) 환경

[0013] 도 1은 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하는 시스템을 도시한다.

[0014] 도 1에 도시된 바와 같이, 하나의 실시예에 따르면, 어플리케이션 서버(예컨대, 멀티-테넌트, MT) 환경(100) 또는, 소프트웨어 어플리케이션들의 개발 및 실행을 할 수 있게 하는 다른 컴퓨팅 환경은 어플리케이션 서버 도메인을 정의하기 위해 런타임 시 이용되는 도메인(102) 구성을 포함하고 이에 따라 동작하도록 구성될 수 있다.

[0015] 하나의 실시예에 따르면, 어플리케이션 서버는 런타임 시 이용하기 위해 정의되는 하나 이상의 파티션들(104)을 포함할 수 있다. 각각의 파티션은 글로벌하게 고유한 파티션 식별자(ID) 및 파티션 구성과 관련될 수 있고, 또한 리소스 그룹 템플릿(126)에 대한 참조와 함께 하나 이상의 리소스 그룹들(124) 및/또는 파티션별(partition-specific) 어플리케이션들 및 리소스들(128)을 포함할 수 있다. 도메인-레벨 리소스 그룹들, 어플리케이션들 및/또는 리소스들(140)은 또한, 옵션에 따라서는 리소스 그룹 템플릿을 참조하여 도메인 레벨에서 정의될 수 있다.

[0016] 각각의 리소스 그룹 템플릿(160)은 하나 이상의 어플리케이션들 A(162), B(164), 리소스들 A(166), B(168) 및/또는 다른 전개가능한 어플리케이션들 또는 리소스들(170)을 정의할 수 있고, 리소스 그룹에 의해 참조될 수 있다. 예를 들어, 도 1에 도시된 바와 같이, 파티션(104) 내의 리소스 그룹(124)은 리소스 그룹 템플릿(160)을 참조(190)할 수 있다.

[0017] 일반적으로, 시스템 어드미니스트레이터는 파티션들, 도메인-레벨 리소스 그룹들 및 리소스 그룹 템플릿들 및 보안 영역(security realms)을 정의할 수 있고, 파티션 어드미니스트레이터는 예컨대, 파티션-레벨 리소스 그룹들을 생성, 파티션에 어플리케이션들을 전개(deploy) 또는 파티션에 대한 특정 영역을 참조함으로써, 자신만의 파티션의 양상들을 정의할 수 있다.

[0018] 도 2는 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하는 시스템을 더 도시한다.

[0019] 도 2에 예시된 바와 같이, 하나의 실시예에 따르면, 파티션(202)은 예컨대, 리소스 그룹 템플릿(210)에 대한 참조(206)를 포함하는 리소스 그룹(205), 가상 타겟(예컨대, 가상 호스트) 정보(204) 및 플러그가능 데이터베이스(PDB: pluggable database) 정보(208)를 포함할 수 있다. 리소스 그룹 템플릿(예컨대, 210)은 예컨대, 자바 어드미니스트레이터 서버(JMS) 서버(213), 축적 전송(SAF: store-and-forward) 에이전트(215), 메일 세션 컴포넌트(216) 또는 자바 데이터베이스 연결(JDBC) 리소스(217)과 같은 리소스들과 함께 복수의 어플리케이션들 A(211) 및 B(212)을 정의할 수 있다.

[0020] 도 2에 도시된 리소스 그룹 템플릿이 예로서 제공되며, 다른 실시예들에 따르면 다른 타입의 리소스 그룹 템플릿들 및 요소들이 제공될 수 있다.

[0021] 하나의 실시예에 따르면, 파티션(예컨대, 202) 내의 리소스 그룹이 특정한 리소스 그룹 템플릿(예컨대, 210)을 참조(220)할 때, 특정한 파티션과 관련된 정보는 파티션별 정보(230), 예컨대 파티션별 PDB 정보를 나타내기 위해, 참조된 리소스 그룹 템플릿과 조합하여 이용될 수 있다. 그 다음, 파티션별 정보는 파티션에 의해 사용하기 위한 리소스들, 예컨대 PDB 리소스를 구성하기 위해 어플리케이션 서버에 의해 이용될 수 있다. 예를 들어, 파티션(202)과 관련된 파티션별 PDB 정보는 그 파티션에 의해 이용하기 위한 적절한 PDB(238)를 갖는 컨테이너 데이터베이스(CDB)(236)를 구성(232)하기 위해 어플리케이션 서버에 의해 이용될 수 있다.

[0022] 마찬가지로, 하나의 실시예에 따르면, 특정한 파티션과 관련된 가상 타겟 정보는 파티션에 의해 이용하기 위한 파티션별 가상 타겟(240)(예컨대, baylandurgentcare.com)을 정의(239)하기 위해 이용될 수 있는 바, 이는 그 다음, URL(uniform resource locator), 예컨대, http://baylandurgentcare.com을 통해 액세스가능하게 될 수 있다.

[0023] 도 3은 하나의 실시예에 따른 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하는 시스템을 더 도시한다.

[0024] 하나의 실시예에 따르면, config.xml 구성 파일과 같은 시스템 구성이 파티션을 정의하기 위해 이용되는 바, 이 시스템 구성은 그 파티션 및/또는 다른 파티션 속성들과 관련된 리소스 그룹들에 대한 구성 요소들을 포함한다. 속성 이름/값 쌍들을 이용하여 파티션 당 값들이 특정될 수 있다.

[0025] 하나의 실시예에 따르면, 복수의 파티션들은 관리 서버/클러스터(242) 내에서 또는 CDB(243)에게로의 액세스를 제공할 수 있고 웹 티어(web tier)(244)를 통해 액세스가능한 유사한 환경 내에서 실행될 수 있다. 이는 예컨대, 도메인 또는 파티션이 (CDB의) PDB들 중 하나 이상과 관련되도록 한다.

[0026] 하나의 실시예에 따르면, 이 예시적인 파티션 A(250) 및 파티션 B(260)에서 복수의 파티션들 각각은 그 파티션과 관련된 복수의 리소스들을 포함하도록 구성될 수 있다. 예를 들어, 파티션 A는 PDB A(259)와 관련된 데이터소스 A(257)와 함께 어플리케이션 A1(252), 어플리케이션 A2(254) 및 JMS A(256)를 포함하고 리소스 그룹(251)을 포함하도록 구성될 수 있고, 파티션은 가상 타겟 A(258)를 통해 액세스가능하다. 마찬가지로, 파티션 B(260)는 PDB B(269)와 관련된 데이터소스 B(266)와 함께 어플리케이션 B1(262), 어플리케이션 B2(264) 및 JMS B(266)를 포함하는 리소스 그룹(261)을 포함하도록 구성될 수 있고, 파티션은 가상 타겟 B(268)을 통해 액세스가능하다.

[0027] 상기 여러 예들은 CDB 및 PDB들의 이용을 예시하지만, 다른 실시예들에 따르면 다른 타입의 멀티테넌트 또는 비-멀티테넌트(non-multi-tenant) 데이터베이스들이 지원될 수 있고, 예컨대 스키마들(schemas)의 이용 또는 서로 다른 데이터베이스들의 이용을 통해 각각의 파티션에 대해 특정한 구성이 제공될 수 있다.

[0028] 리소스들

[0029] 하나의 실시예에 따르면, 리소스는 환경의 도메인에 대해 전개될 수 있는 시스템 리소스, 어플리케이션 또는 다른 리소스 또는 객체이다. 예를 들어, 하나의 실시예에 따르면, 리소스는 서버, 클러스터 또는 다른 어플리케이션 서버 타겟에 대해 전개될 수 있는 어플리케이션, JMS, JDBC, 자바메일, WLDF, 데이터 소스 또는 다른 시스템 리소스 또는 다른 타입의 객체일 수 있다.

[0030] 파티션들

[0031] 하나의 실시예에 따르면, 파티션은, 파티션 식별자(ID) 및 구성과 관련될 수 있고 어플리케이션들을 포함하고

그리고/또는 리소스 그룹들 및 리소스 그룹 템플릿들의 이용을 통해 도메인 범위의(domain-wide) 리소스들을 참조할 수 있는 도메인의 런타임 및 어드미니스트레이티브 서브디비전(administrative subdivision) 또는 슬라이드이다.

[0032] 일반적으로, 파티션은 자신만의 어플리케이션들을 포함하고, 리소스 그룹 템플릿들을 통해 도메인 범위의 어플리케이션들을 참조하며, 자신만의 구성을 가질 수 있다. 파티션가능한 엔티티들은 리소스들, 예컨대 JMS, JDBC, 자바메일, WLDF 리소스들, 및 JNDI 이름공간, 네트워크 트래픽, 작업 관리자들 및 보안 정책 및 영역들과 같은 다른 컴포넌트들을 포함할 수 있다. 멀티테넌트 환경의 맥락에서, 시스템은 테넌트와 관련된 파티션들의 어드미니스트레이티브 및 런타임 양상들(administrative and runtime aspects)에게 테넌트 액세스를 제공하도록 구성될 수 있다.

[0033] 하나의 실시예에 따르면, 파티션 내의 각각의 리소스 그룹은 옵션에 따라서는, 리소스 그룹 템플릿을 참조할 수 있다. 파티션은 복수의 리소스 그룹들을 가질 수 있고, 이들 각각은 리소스 그룹 템플릿을 참조할 수 있다. 각각의 파티션은 파티션의 리소스 그룹들이 참조하는 리소스 그룹 템플릿들에 특정되지 않는 구성 데이터에 대한 속성들을 정의할 수 있다. 이는 파티션이 그 파티션으로 이용할 특정값들에 대한 리소스 그룹 템플릿에 정의된 전개가능한 리소스들의 바인딩으로서 역할을 하게 한다. 일부 경우들에서, 파티션은 리소스 그룹 템플릿에 의해 특정되는 구성 정보를 오버라이드(override)할 수 있다.

[0034] 하나의 실시예에 따르면, 예컨대 config.xml 구성 파일에 의해 정의된 파티션 구성은 복수의 구성 요소들, 예컨대, 파티션을 정의하는 속성들 및 차일드 요소들을 포함하는 "partition", 상기 파티션에 대해 전개되는 어플리케이션들 및 리소스들을 포함하는 "resource-group", 해당 템플릿에 의해 정의되는 어플리케이션들 및 리소스들을 포함하는 "resource-group-template", 데이터베이스별 서비스 이름, 사용자 이름 및 패스워드를 포함하는 "jdbc-system-resource-override" 및 리소스 그룹 템플릿들 내의 매크로 교체(macro replacement)를 위해 이용될 수 있는 속성 키 값들을 포함하는 "partition-properties"을 포함할 수 있다.

[0035] 시동 시, 시스템은 리소스 그룹 템플릿으로부터 각각의 리소스에 대한 파티션별 구성 요소들을 생성하기 위해 구성 파일에 의해 제공되는 정보를 이용할 수 있다.

[0036] 리소스 그룹들

[0037] 하나의 실시예에 따르면, 리소스 그룹은 도메인 또는 파티션 레벨에서 정의될 수 있고 리소스 그룹 템플릿을 참조할 수 있는 전개가능한 리소스들의 명명된, 완전히 적격한 집합(named, fully-qualified collection)이다. 리소스 그룹 내의 리소스들은 어드미니스트레이터가 시작 또는 이 리소스들에 연결하기 위해 필요한 모든 정보, 예컨대 데이터 소스에 연결하기 위한 크리덴셜들 또는 어플리케이션에 대한 타겟팅 정보를 제공했다는 점에서 완전히 적격한(fully-qualified) 것으로 고려된다.

[0038] 시스템 어드미니스트레이터는 도메인 레벨에서 또는 파티션 레벨에서 리소스 그룹들을 선언할 수 있다. 도메인 레벨에서, 리소스 그룹은 그룹 관련 리소스들에 편리한 방식을 제공한다. 시스템은 비그룹화된 리소스들과 동일하게 도메인-레벨 리소스 그룹에서 선언된 리소스들을 관리할 수 있어서, 상기 리소스들은 시스템 시동 동안 시작되고, 시스템 셧-다운 동안 정지될 수 있다. 어드미니스트레이터는 또한, 그룹 내의 리소스를 개별적으로 정지, 시작 또는 제거할 수 있고, 그룹 상에서 동작함으로써 묵시적으로 상기 그룹 내의 모든 리소스들에 영향을 줄 수 있다. 예를 들어, 리소스 그룹을 정지시키는 것은 아직 정지되지 않은 그룹 내의 리소스들 모두를 정지시키고, 리소스 그룹을 시작시키는 것은 아직 시작되지 않은 그룹 내의 어떤 리소스들을 시작시키며, 리소스 그룹을 제거하는 것은 그룹에 포함된 리소스들 모두를 제거한다.

[0039] 파티션 레벨에서, 시스템 또는 파티션 어드미니스트레이터는 어떤 보안 제약들을 겪는 파티션 내의 0개 이상의 리소스 그룹들을 구성할 수 있다. 예를 들어, SaaS 사용 경우에서, 다양한 파티션-레벨 리소스 그룹들은 도메인-레벨 리소스 그룹을 참조할 수 있고, PaaS 사용 경우에서, 리소스 그룹 템플릿을 참조하는 것이 아니라, 해당 파티션 내에서만 이용가능해지는 어플리케이션들 및 이들의 관련 리소스들을 표시하는 파티션-레벨 리소스 그룹들이 생성될 수 있다.

[0040] 하나의 실시예에 따르면, 리소스 그룹화는 어플리케이션들 및 이 어플리케이션들이 도메인 내에서 별개의 어드미니스트레이티브 유닛으로서 이용하는 리소스들을 함께 그룹화하기 위해 이용될 수 있다. 예를 들어, 하기 기술되는 의료 기록들(MedRec) 어플리케이션에서, 리소스 그룹화는 MedRec 어플리케이션 및 이의 리소스들을 정의한다. 복수의 파티션들은 각각 파티션별 구성 정보를 이용하여 동일한 MedRec 리소스 그룹을 실행할 수 있어서, 각각의 MedRec 인스턴스의 일부인 어플리케이션들은 각각의 파티션에 특정적이게 된다.

[0041] 리소스 그룹 템플릿들

[0042] 하나의 실시예에 따르면, 리소스 그룹 템플릿은 리소스 그룹으로부터 참조될 수 있는 도메인 레벨에서 정의되는 전개가능한 리소스들의 집합이며, 이의 리소스들을 활성화시키기 위해 요구되는 정보의 일부는 파티션 레벨 구성의 사양을 지원할 수 있도록 템플릿 자체의 일부로서 저장되지 않을 수 있다. 도메인은 어떤 수의 리소스 그룹 템플릿들을 포함할 수 있고, 이 템플릿들 각각은 예컨대, 하나 이상의 관련 자바 어플리케이션들 및 이 어플리케이션들이 의존하는 리소스들을 포함할 수 있다. 이러한 리소스들에 관한 정보의 일부는 모든 파티션들에 걸쳐 동일할 수 있고, 다른 정보는 파티션 마다 다양할 수 있다. 모든 구성이 도메인 레벨에서 특정될 필요는 없는바 - 대신 파티션 레벨 구성이 매크로들 또는 속성 이름/값 쌍들의 이용을 통해 리소스 그룹 템플릿에 특정될 수 있다.

[0043] 하나의 실시예에 따르면, 특정한 리소스 그룹 템플릿은 하나 이상의 리소스 그룹들에 의해 참조될 수 있다. 일반적으로, 어떤 소정 파티션 내에서, 리소스 그룹 템플릿은 오직 한 번에 하나의 리소스 그룹에 의해 참조될 수 있는 바, 즉 동일한 파티션 내에서 복수의 리소스 그룹들에 의해 동시에 참조될 수 없다. 그러나, 리소스 그룹 템플릿은 다른 파티션 내의 다른 리소스 그룹에 의해 동시에 참조될 수 있다. 리소스 그룹을 포함하는 객체, 예컨대 도메인 또는 파티션은 리소스 그룹 템플릿 내의 어떤 토큰들의 값을 설정하기 위해 속성 이름/값 할당들을 이용할 수 있다. 시스템이 참조 리소스 그룹을 이용하여 리소스 그룹 템플릿을 활성화시킬 때, 이는 이 토큰들을 리소스 그룹 포함 객체에 설정된 값들과 교체할 수 있다. 일부 경우들에서, 시스템은 또한, 각각의 파티션/템플릿 조합에 대해 런타임 구성을 생성하기 위해 통계적으로 구성된 리소스 그룹 템플릿들 및 파티션들을 이용할 수 있다.

[0044] 예를 들어, SaaS 사용 경우에서, 시스템은 동일한 어플리케이션들 및 리소스들을 복수번 활성화시킬 수 있는바, 이는 이 어플리케이션들 및 리소스들을 이용할 각각의 파티션에 대해 한번씩 활성화시키는 것을 포함한다. 어드미니스트레이터가 리소스 그룹 템플릿을 정의할 때, 이들은 다른 곳에 공급될 정보를 표시하기 위해 토큰들을 이용할 수 있다. 예를 들어, CRM-관련 데이터 리소스에 연결할 시 이용하기 위한 사용자이름이 \${CRMDataUsername}로서 리소스 그룹 템플릿에 표시될 수 있다.

[0045] 테넌트들

[0046] 하나의 실시예에 따르면, 멀티-테넌트(MT) 어플리케이션 서버 환경과 같은 멀티-테넌트 환경에서, 테넌트는 하나 이상의 파티션들 및/또는 하나 이상의 테넌트-인지 어플리케이션들에 의해 표시될 수 있거나 또는 그렇지 않으면 이와 관련될 수 있는 엔티티이다.

[0047] 예를 들어, 테넌트들은 서로 다른 외부의 회사들 또는 특정 기업 내의 서로 다른 부서들(예컨대, HR 및 재무 부서들)과 같은 별개의 사용자 조직들을 나타낼 수 있고, 이들 각각은 서로 다른 파티션과 관련될 수 있다. 글로벌 하게 고유한 테넌트 신원(테넌트 ID)는 특정한 시점에 특정한 테넌트와의 특정한 사용자의 관련성이다. 시스템은 예컨대, 사용자 신원 저장소를 참조함으로써 특정한 사용자가 어느 테넌트에 속하는지를 사용자 신원으로부터 도출할 수 있다. 사용자 신원은 시스템으로 하여금 이들로만 한정되는 것은 아니지만, 사용자가 속할 수 있는 테넌트를 포함하여 사용자가 수행하도록 인가된 그러한 동작(action)들을 실시할 수 있게 한다.

[0048] 하나의 실시예에 따르면, 시스템은 서로 다른 테넌트들의 관리 및 런타임이 서로 격리되게 한다. 예를 들어, 테넌트들은 자신의 어플리케이션들 및 이들이 액세스할 수 있는 리소스들의 일부 거동들을 구성할 수 있다. 시스템은 특정한 테넌트가 다른 테넌트에 속하는 아티팩트들을 어드미니스터(administer)할 수 없게 할 수 있고, 런타임 시 특정한 테넌트 대신 작동하는 어플리케이션들이 그 테넌트와 관련된 리소스들만 참조하고 다른 테넌트들과 관련된 리소스들을 참조하지 못하게 할 수 있다.

[0049] 하나의 실시예에 따르면, 테넌트-비인지 어플리케이션은 테넌트들을 분명하게(explicitly) 다루는 로직을 포함하지 않는 어플리케이션이어서, 상기 어플리케이션이 이용하는 어떤 리소스들은 어떤 사용자가 상기 어플리케이션이 응답하는 요청을 제출했는지에 관계없이 액세스가능할 수 있다. 이와는 대조적으로, 테넌트-인지 어플리케이션은 테넌트들을 분명하게 다루는 로직을 포함한다. 예를 들어, 사용자의 신원에 기초하여, 어플리케이션은 사용자가 속하는 테넌트를 도출할 수 있고, 그 정보를 테넌트별 리소스들에 액세스하기 위해 이용할 수 있다.

[0050] 하나의 실시예에 따르면, 시스템은 사용자들이 테넌트-인지형이도록 명시적으로 작성되는 어플리케이션들을 전개할 수 있게 하여서, 어플리케이션 개발자들은 현재의 테넌트의 테넌트 ID를 획득할 수 있다. 그 다음, 테넌트-인지 어플리케이션은 상기 어플리케이션의 단일 인스턴스를 이용하는 복수의 테넌트들을 처리하기 위해 테넌트 ID를 이용할 수 있다.

[0051] 예를 들어, 단일 의사의 사무실 또는 병원을 지원하는 MedRec 어플리케이션은 두 개의 서로 다른 파티션들 또는 테넌트들, 예컨대 Bayland Urgent Care 테넌트 및 Valley Health 테넌트에 노출될 수 있는 바, 이들 각각은 기저 어플리케이션 코드를 변경함이 없이 예컨대, 별개의 PDB들과 같은 별개의 테넌트별 리소스들에 액세스할 수 있다.

[0052] 예시적인 도메인 구성 및 멀티-테넌트 환경

[0053] 하나의 실시예에 따르면, 어플리케이션들은 도메인 레벨에서 리소스 그룹 템플릿에 대해 또는 파티션 범주의(scoped to) 또는 도메인 범위의 리소스 그룹에 대해 전개될 수 있다. 어플리케이션 구성은 어플리케이션 당 또는 파티션 당 특정되는 전개 계획들을 이용하여 오버라이드될 수 있다. 전개 계획들은 또한, 리소스 그룹의 일부로서 특정될 수 있다.

[0054] 도 4는 하나의 실시예에 따른 예시적인 멀티-테넌트 환경에서 이용하기 위한 도메인 구성을 도시한다.

[0055] 하나의 실시예에 따르면, 시스템이 파티션을 시작할 때, 이는 제공되는 구성에 따라 각각의 데이터베이스 인스턴스들에 대해 가상 타겟들(예컨대, 가상 호스트들) 및 연결 풀들을 생성하는 바, 이러한 생성은 각각의 파티션 당 하나를 생성하는 것을 포함한다.

[0056] 전형적으로는, 각각의 리소스 그룹 템플릿은 하나 이상의 관련 어플리케이션들 및 이 어플리케이션들이 의존하는 리소스들을 포함할 수 있다. 각각의 파티션은 파티션과 관련된 특정 값들에 대한 리소스 그룹 템플릿들 내의 전개가능한 리소스들의 바인딩을 제공함으로써 상기 파티션이 참조하는 리소스 그룹 템플릿에 특정되지 않은 구성 데이터를 제공할 수 있는 바, 이는 일부 경우들에서, 상기 리소스 그룹 템플릿에 의해 특정된 특정 구성 정보를 오버라이드하는 것을 포함한다. 이는 시스템이 각각의 파티션이 정의한 속성 값들을 이용하여 각각의 파티션에 대해 서로 다르게 리소스 그룹 템플릿에 의해 표시되는 어플리케이션을 활성화시킬 수 있게 한다.

[0057] 일부 예들에서, 파티션은, 리소스 그룹 템플릿들을 참조하지 않거나 또는 자신만의 파티션 범주의 전개가능한 리소스들을 직접적으로 정의하는 리소스 그룹들을 포함할 수 있다. 파티션 내에 정의되는 어플리케이션들 및 데이터 소스들은 일반적으로, 그 파티션에게만 이용가능하다. 리소스들은 파티션들, <partitionName>/<resource JNDI name>, 또는 domain:<resource JNDI name>를 이용하여 파티션들에 걸쳐 액세스될 수 있도록 전개될 수 있다.

[0058] 예를 들어, MedRec 어플리케이션은 복수의 자바 어플리케이션들, 데이터 소스, JMS 서버 및 메일 세션을 포함할 수 있다. 복수의 테넌트들에 대해 MedRec 어플리케이션을 실행하기 위해, 시스템 어드미니스트레이터는 단일 MedRec 리소스 그룹 템플릿(286)을 정의할 수 있고, 상기 템플릿에 그러한 전개가능한 리소스들을 선언할 수 있다.

[0059] 도메인 레벨의 전개가능한 리소스들과는 대조적으로, 리소스 그룹 템플릿에 선언된 전개가능한 리소스들은 템플릿에 완전히 구성되지 않을 수 있거나 또는 있는 그대로(as-is) 활성화되지 못할 수 있는 바, 그 이유는 이 리소스들에 일부 구성 정보가 결여되어 있기 때문이다.

[0060] 예를 들어, MedRec 리소스 그룹 템플릿은 어플리케이션들에 의해 이용되는 데이터 소스를 선언할 수 있지만, 이는 데이터베이스에 연결하기 위한 URL을 특정하지 않을 수 있다. 서로 다른 테넌트들과 관련된 파티션들, 예컨대 파티션 BUC-A(290)(Bayland Urgent Care, BUC) 및 파티션 VH-A(292)(Valley Health, VH)는 MedRec 리소스 그룹 템플릿을 참조(296, 297)하는 MedRec 리소스 그룹(293, 294)을 각각 포함함으로써 하나 이상의 리소스 그룹 템플릿들을 참조할 수 있다. 그 다음, 상기 참조는 Bayland Urgent Care 테넌트가 사용하기 위한 BUC-A 파티션과 관련된 가상 호스트 baylandurgentcare.com(304) 및 Valley Health 테넌트가 사용하기 위한 VH-A 파티션과 관련된 가상 호스트 valleyhealth.com(308)를 포함하는, 각각의 테넌트에 대한 가상 타겟들/가상 호스트들을 생성(302, 306)하기 위해 이용될 수 있다.

[0061] 도 5는 하나의 실시예에 따른 예시적인 멀티-테넌트 환경을 더 도시한다. 도 5에 도시된 바와 같이 그리고 하나의 실시예에 따라 두 파티션들이 MedRec 리소스 그룹 템플릿을 참조하는 상기 예를 계속 들어, 서블릿 엔진(310)이 복수의 테넌트 환경들, 이 예에서는 Bayland Urgent Care Physician 테넌트 환경(320) 및 Valley Health Physician 테넌트 환경(330)을 지원하기 위해 이용될 수 있다.

[0062] 하나의 실시예에 따르면, 각각의 파티션(321, 331)은 테넌트 환경에 대해 유입 트래픽을 수락할 서로 다른 가상 타겟, 및 파티션 및 이의 리소스들(324, 334)(이 예에서는 bayland urgent care 데이터베이스 또는 valley health 데이터베이스를 각각 포함)에 연결하기 위한 서로 다른 URL(322, 332)를 정의할 수 있다. 상기 데이터베

이스 인스턴스들은 호환가능한 스키마를 이용할 수 있는 바, 그 이유는 동일한 어플리케이션 코드가 두 데이터베이스들에 대해 실행될 것이기 때문이다. 시스템이 파티션들을 시작할 때, 이는 가상 타겟들, 및 각각의 데이터베이스 인스턴스들에게로의 연결 풀들을 생성할 수 있다.

[0063] 파티션 보안

[0064] 상기 기술된 파티션 특징들은 개별적인 파티션들 간의 그리고 파티션들과 이 파티션들을 포함하는 멀티테넌트 시스템 간의 격리를 제공한다. 상기 기술된 바와 같이, 각각의 파티션은 자신만의 어플리케이션들을 포함하고, 리소스 그룹 템플릿들을 통해 도메인 범위의 어플리케이션들을 참조하며, 자신만의 구성을 가질 수 있다. 파티션가능한 엔티티들은 리소스들, 예컨대 JMS, JDBC, 자바메일, WLDF 리소스들 및 JNDI 네임스페이스, 네트워크 트래픽, 작업 관리자들과 같은 다른 컴포넌트들을 포함할 수 있다. 멀티테넌트 시스템의 보안 기능들이 파티션된/멀티-테넌트 환경에서 어플리케이션들 및 상기 시스템을 보안하기 위해 이러한 격리를 모델링하고 액세스 제어들을 적절하게 시행(enforce)하는 것이 바람직하다. 본 발명의 하나의 실시예에 따르면, 상기 시스템은 멀티테넌트 어플리케이션 서버 환경에서 보안에 대한 지원을 포함할 수 있다. 상기 시스템은 멀티테넌트 환경 및 상기 기술된 파티션 특징들의 지원에 있어서 보안 서비스들을 제공한다.

[0065] 본 발명의 실시예들은 인증, 인가, 크리덴셜 매핑, 감사, 패스워드 검증, 증명서 검증 및 사용자 락아웃에 대한 구성을 포함하는 보안 서비스들의 파티션 당 구성을 제공한다. 본 발명의 실시예들은 또한, 특정한 파티션에 전개된 어플리케이션들이 상기 특정한 파티션의 사용자들에게만 액세스가능하게 하고, 특정한 파티션에 대한 파티션 구성이 오직 상기 특정한 파티션의 어드미니스트레이터(및 웹로직 어드미니스트레이터)에게 이용가능하게 하고 그리고 파티션 어드미니스트레이터들이 오직 글로벌하게 가시적인 리소스들 및 구성에 대한 판독 전용 액세스를 가지도록, 파티션 및 글로벌 리소스들에 대한 액세스 제어를 제공한다.

[0066] 도 6은 하나의 실시예에 따른 멀티-테넌트 환경에서 보안을 지원하는 시스템 및 방법의 일반적인 기능을 예시한다. 도 6에 도시된 바와 같이, 파티션 A(250) 및 파티션 B(262) 각각은 분리된 파티션 보안 구성 A(650) 및 파티션 보안 구성 B(660)를 제공받을 수 있다. 파티션 보안 구성 A(650)는 파티션 A의 사용자들 및 어드미니스트레이터들에게 파티션 A 리소스들(251)에 대한 보안된 액세스를 제공할뿐만 아니라 글로벌 리소스들(140)에 대한 제한된 액세스를 제공하고 이 어드미니스트레이터들 및 사용자들에 의한 파티션 B 리소스들(261)에 대한 액세스를 방지한다. 도메인 어드미니스트레이터(600)는 특정한 파티션에 전개된 어플리케이션들이 상기 특정한 파티션의 사용자들 및 어드미니스트레이터들에게만 액세스가능하게 하도록 파티션 및 글로벌 리소스들에 대한 액세스 제어를 보장하기 위해 각각의 도메인에 대한 파티션 보안 구성을 초기화하도록 허용된다.

[0067] 본 발명의 실시예들은 개별적인 프린서플들(principals)의 관리, 이들의 인증, 인가 및 시스템 및 파티션 경계들 내에서의 또는 이들에 걸친 특권들(privileges)의 제어를 제공하는 강화된 플러그가능 신원 관리(IdM) 시스템을 지원하고 활용한다. 본 발명의 특정한 실시예들은 또한, 하나 이상의 파티션 사용자들을 특정한 파티션에 대한 파티션 어드미니스트레이터들로서 지정하는 능력을 제공하는 어드미니스트레이션 서비스들을 제공한다.

[0068] 멀티테넌트 서버 환경에서 보안을 위해 제공되는 지원은, 영역들(realms) 및 영역 기반 서비스들에 주로 관련된 다수의 특징들 및 거동들을 포함하는 바, 이는 다음을 포함한다.

[0069] • 복수의 영역들: 복수의 활성 영역들에 대한 지원은 각각의 파티션이 서로 다른 영역에 대해 실행될 수 있게 함으로써 영역 기반 서비스들에 대한 파티션 당 구성을 가능하게 한다. 파티션들은 또한, 보안 영역을 공유하는 것을 선택할 수 있는 바, 이는 보안 구성 및 메타데이터에 관하여 독립성 및 격리성의 결과적인 손실을 갖는다.

[0070] • 신원 도메인들: 신원 도메인은, 전형적으로 물리적 저장소에 있는 사용자들 및 그룹들의 개별 세트를 나타내는 사용자들 및 그룹들에 대한 논리적 네임스페이스이다. 신원 도메인들은 특정한 파티션들과 관련된 사용자들을 식별하기 위해 그리고 다른 목적들로 사용된다.

[0071] • 파티션-인지 보안 서비스들: 파티션-인지 보안 서비스들은 자신들이 실행되는 파티션 컨텍스트를 이해하여서, 이들은 예를 들어 리소스를 소유하는 파티션에 기초하여 리소스에 대한 액세스를 제어할 수 있다. 정의상(by definition), 파티션-인지 서비스들은 또한 신원-도메인-인지형이다.

[0072] 이러한 특징들 및 기능적 구역(area)들에 대한 추가의 세부사항들이 하기에 기술된다.

[0073] 복수의 영역 지원

[0074] 영역 또는 보안 영역은 시스템 내의 보안 서비스들에 대한 명명된 구성(named configuration)이다. 영역들은 인

증, 인가, 롤 매핑, 크리덴셜 매핑, 감사 및 멀티테넌트 서버 환경에서 이용하기 위한 다른 서비스들을 구성하기 위해 이용된다. 본 발명의 실시예들은 인증, 인가, 크리덴셜 매핑, 감사, 패스워드 검증, 증명서 검증 및 사용자 락아웃에 대한 구성을 포함하는 보안 서비스들의 파티션 당 구성을 제공한다. 시스템은 복수의 활성 영역들을 지원한다. 각각의 활성 파티션은 서로 다른 보안 영역과 관련될 수 있다. 따라서, 본 발명의 실시예들에 의해 제공되는 멀티테넌트 시스템에서, 복수의 영역 런타임들에 대응하고 복수의 서로 다른 파티션들에 대응하는 복수의 활성 구성들이 존재할 수 있다.

[0075] 시스템에 의해 제공되는 보안 서비스들은 "영역 기반(realm-based)" 또는 "비 영역 기반(non-realm-based)"을 특징으로 할 수 있다. 영역 기반 서비스들은 영역 객체에 의해 구성에 표시되는 바와 같이 보안 "영역"에 대해 구성되는 그러한 서비스들이다. 영역 기반 서비스들은 인증, 인가, 크리덴셜 매핑, 감사 및 여러 다른 서비스들을 포함한다. 비 영역 기반 서비스들은 영역 객체 상에서 구성되지 않는 그러한 서비스들이다. 이들은 보안 구성 객체(도메인 객체의 차일드, 영역 객체들에 대한 피어런트 컨테이너) 상에서 구성되는 도메인 범위의 서비스들 및 설정들, 및 다양한 다른 서비스들을 포함한다.

[0076] 도 7은 복수의 영역들의 이용을 위해 구성된 멀티테넌트 환경을 도시한다. 파티션 보안은 복수의 활성 영역들을 지원하기 위해 보안 서비스를 활용한다. 복수의 영역 객체 인스턴스들은 보안 구성 객체 상의 어레이 속성으로서 관리된다. 도 7에 도시된 바와 같이, 보안 구성 객체(700)는 세 개의 영역 객체들 즉, 영역 A(750), 디폴트 영역(702) 및 영역 B(760)를 포함한다. 디폴트 영역(702)은 도메인/글로벌 런타임에 의해 이용되는 영역을 나타낸다. 그러나, 영역 A(750) 및 영역 B(760)는 또한, 런타임 시 활성이다. 디폴트 영역은 또한, "어드민 영역(admin realm)"으로 지칭될 수 있는 바, 그 이유는 디폴트 영역이 시스템 및 어드미니스트레이티브 리소스들에 대한 인가 체크들을 수행하기 위해 그리고 다른 어드미니스트레이티브 목적들을 위해 이용되기 때문이다.

[0077] 각각의 파티션 보안 구성은 도메인 구성의 보안 구성 객체 상에서 구성되는 영역들 중 하나를 참조하는 영역 속성을 가진다. 따라서, 파티션 보안 구성 A(650)는 영역 A(750)를 참조하는 영역 속성(752)을 포함하고, 파티션 보안 구성 A(660)는 영역 B(760)를 참조하는 영역 속성(762)을 포함한다. 파티션에 대한 보안 영역을 구성하는 것을 지원하기 위한 메소드들이 파티션 리소스들에 추가된다. 각각의 파티션 보안 구성은 (영역이 파티션에 의해 참조되는 경우 상기 영역이 삭제될 수 없음을 또한 암시하는) 유효한 영역 참조를 가져야 한다. 오직 시스템 어드미니스트레이터(600)가 파티션의 영역 참조를 변경할 수 있는 바, 파티션 어드미니스트레이터들은 이를 행할 수 없다. 파티션은 자신의 영역 참조가 변경되기 전에 정지되어야 한다.

[0078] 영역들에 대한 파티션들의 매핑은 완전히 플렉서블한 바, 매 파티션(every partition)은 서로 다른 영역을 참조할 수 있거나, 모든 파티션들(all partitions)은 동일한 영역을 공유할 수 있거나 또는 일부 파티션들은 서로 다른 영역들을 참조할 수 있고 다른 파티션들은 영역을 공유할 수 있다. 어떤 조합이 가능하다. 각각의 파티션에 대해 개별적인 영역을 구성하는 것은 파티션에 대한 가장 큰 독립성 및 격리성을 제공한다. 복수의 파티션들 간에 영역을 공유하는 것은 구성을 간략화할 수 있고, 복수의 파티션들 간에 영역을 공유하는 것은 구성을 간략화할 수 있고, 서로 신뢰하고 유사한 보안 구성 요건들을 가지는 관련된 파티션들에게 양호한 선택일 수 있다. 그러나, 독립성 및 격리성이 감소되고, 하나의 파티션의 어드미니스트레이터들은 다른 공유 파티션들에 영향을 끼치는 동작(action)을 취할 수 있다. 디폴트/어드민 영역을 공유하는 것이 허용되지만, 권고되지는 않는다. 디폴트 영역이 공유되는 경우, 파티션 어드미니스트레이터들에 의한/대한 특권 확대(escalation)를 회피하기 위한 주의가 기울여져야 하는 바, 그 이유는 디폴트 정책들이 상기 파티션 어드미니스트레이터들로 하여금 디폴트 영역에서 롤 매핑 및 인가 정책들을 수정할 수 있게 할 것이기 때문이다.

[0079] 영역은, 상기 영역을 나타내는 영역 객체가 인스턴스화되어 시스템 어드미니스트레이터에 의해 보안 구성 객체 내의 영역 어레이에 추가될 때, 생성된다. 새로운 영역의 생성은 온라인으로 또는 오프라인으로 발생될 수 있다. 영역 구성은 변경이 저장/활성화될 때마다, 또는 오프라인 변경들의 경우, 서버가 부팅될 때, 검증된다. 디폴트 영역은 서버가 부팅되기 위해 구성되고 유효해야 한다. 영역을 생성하는 것은 영역의 런타임 수명 주기(lifecycle)를 관리하는 데 필수적인 런타임 객체들의 생성을 트리거한다. 일단 생성되면, 영역은 이용가능해진다. 상기 영역은 파티션들에 의해 또는 도메인/글로벌 런타임에 의해 참조될 수 있고, 런타임 코드는 자신의 서비스들을 요청할 수 있다. 생성된 영역은 사용 패턴들 및 구성 변경들에 응답하여 어떤 횟수 시작, 정지 또는 재시작될 수 있다.

[0080] 영역은 요구에 따라(on demand), 상기 영역이 서비스 요청들에 필요할 때, 또는 관리 목적들을 위해 시작된다. 특히, 파티션이 시작될 때, (파티션의 영역이 이미 실행중이지 않은 경우) 파티션의 영역이 또한 시작된다. 영역 서비스들에 대한 런타임 참조들은 영역이 재시작되는 경우에도 상기 영역이 존재하는 한 유효하게 유지되어

야 한다. 존재하지 않는 영역에 대해 서비스를 획득하거나 인보크하는 것을 시도하는 것은 결과적으로, 에러 또는 예외(exception)를 야기할 것이다. 존재하는 영역에 대해 서비스를 획득하거나 또는 인보크하는 것을 시도하는 것은, 상기 영역이 요청을 만족시키기 위해 시작되어야 하는 경우에도 그리고 서비스 참조가 획득된 후 상기 영역이 시작, 정지 또는 재시작된 경우에도, 항상 성공적이어야 한다.

[0081] 파티션된 환경에서, 복수의 활성 영역들이 존재할 수 있다. 따라서, 영역 서비스를 요청할 때 영역 이름이 항상 특정된다. 상기 특정된 영역 이름은 특정한 영역 또는 디폴트 영역을 식별할 수 있다. 영역 이름 파라미터는, 영역 - 이 영역에 대해 서비스가 요청됨 - 을 식별하는 스트링 값이다. 대부분의 서비스 인보케이션들을 위해 이용할 정확한 영역은 콜러(caller)의 파티션 컨텍스트, 인보크되는 서비스 및 메소드, 및 콜(call)의 파라미터들에 좌우된다. 디폴트 영역이 특정되는 경우, 로직이 각각의 콜의 컨텍스트를 평가하는 "서비스 프록시들"에 의해 정확한 영역을 선택하고, 위임할 정확한 영역을 결정하고 그리고 그 영역에 적절한 서비스를 인보크하기 위해 적용된다. 이러한 방식으로 영역 선택을 캡슐화(encapsulating)하는 것은, 복잡한 로직을 구현하거나 또는 복수의 영역들로부터 서비스들에 대한 참조들을 캐시하기 위한 코드를 콜할 필요를 회피한다. 런타임 코드는 보안 서비스 관리자로부터 영역 기반의 보안 서비스들에 대한 참조들을 획득한다. 대부분의 콜러들은 상기 콜러들이 필요로 하는 서비스들에 대한 참조들을 상기 콜러들이 초기화할 때 한번 획득하고, 상기 콜러들이 필요로 하는 한은 이 참조들을 유지한다.

[0082] 보안 서비스 관리자는 영역 서비스들에 대한 직접적인 참조들 대신 서비스 프록시들을 리턴한다. 서비스 프록시 거동은 서비스를 요청할 때 주어지는 영역 이름에 기초하여 다양할 것이다. 기존 영역의 실제 이름이 특정될 때, 리턴된 프록시는 그 특정한 영역에 위임(delegate)하도록 유선 연결(hard-wired)될 것이다. 유선 연결된 프록시 서비스들은 구성된 영역에 모든 요청들을 위임한다. 하기 코드 스니펫은 콜러가 프린서플 인증자 서비스(Principal Authenticator service)에 대한 참조를 어떻게 얻을 수 있는지를 예시한다.

```
import weblogic.security.service.PrincipalAuthenticator;
import weblogic.security.service.SecurityServiceManager;
PrincipalAuthenticator pa =
SecurityServiceManager.getPrincipalAuthenticator(kernelID, realmName);
```

[0083]

[0084] 예시된 바와 같이, 콜러는 식별된 영역 이름에 대한 프린서플 인증자 서비스를 요청한다. 시스템 보안 서비스는 특정된 영역에 의존하는 영역 서비스 프록시 참조를 리턴한다.

[0085] "디폴트" 영역이 보안 서비스 요청에 특정될 때, 서비스에 대한 리턴된 프록시는 리턴할 정확한 프록시를 선택하기 위해, 컨텍스트-감응적(context-sensitive)(또는 "자동 선택") 거동을 제공할 것이다. 자동 선택 프록시 서비스들은 현재의 파티션에 대해 구성된 영역에 일부 요청들을 위임하고, 디폴트/어드민 영역에 다른 요청들을 위임한다. 인증 서비스들은 항상 로컬하며, 현재의 파티션의 영역에 위임된다. 인가 서비스들은 때때로 로컬하며, 예컨대, Serviet 또는 EJB 리소스들과 같은 파티션 리소스들에 대한 현재의 파티션의 영역에 위임된다. 인가 서비스들은 때때로 글로벌하며, 예컨대, 시스템 리소스들과 같은 글로벌 리소스들 예컨대, JMX 및 어드민 리소스들에 대한 디폴트/어드민 영역에 위임된다. 감사 서비스(audit service)들은 항상 로컬하며, 현재의 파티션의 영역에 위임된다. 영역 서비스들은 스스로 감사하기 위해 자신들의 영역으로부터의 감사 서비스를 이용한다. 크리덴셜 매핑 서비스들, 증명서 빌딩/검증, 패스워드 검증 및 사용자 락아웃은 항상 로컬하며, 현재의 파티션의 영역에 위임된다.

[0086] 신원 도메인들

[0087] 신원 도메인(IDD)은 사용자들 및 그룹들에 대한 로직적 네임스페이스이다. 신원 도메인들은 사용자들의 서로 다른 세트들을 식별하고 구별하기 위해 이용된다. 신원 도메인은 특정한 회사(예컨대, "Acme Corp" IDD)로부터의 또는 그 회사 내의 부서(예컨대, "HR 부서" IDD)로부터의 사용자들을 나타낼 수 있다. 클라우드 환경에서, 신원 도메인들은 커스토머 서비스 대표들(representatives)로부터 시스템 어드미니스트레이터들을 구별할 수 있고, 이들을 테넌트 사용자들로부터 구별할 수 있다. 신원 도메인들은 서로 다른 파티션들로부터의 사용자들을 구별하고 그리고 리소스들의 소유권(ownership)을 귀속시키는(attribute) 데 필수적이다. 그러므로, 이들은 파티션들 간을 구별할 수 있는 롤 매핑 및 인가 정책들을 위한 그리고 서로 다른 파티션들로부터의 사용자들 또는 리소스들 구별해야 하는 다른 서비스들을 위한 구현 기술(enabling technology)이다. 따라서, 신원 도메인들은 필수적으로 모든 파티션된 환경들에서 이용되도록 의도된다.

[0088] 신원 도메인에 의해 표시되는 로직적 네임스페이스는, 대응하는 사용자 저장소의 구조/토폴로지에서 물리적 발현(physical manifestation)을 가진다. 신원 도메인의 물리적 표시는 타겟 사용자 저장소 기술에 의해 지원되는 어떤 것 - 예를 들어, 각각의 신원 도메인에 대한 개별 LDAP 인스턴스 또는 데이터베이스의 사용자 레코드들에 추가된 신원 도메인 필드일 수 있다. 공유 IdM/Oracle Public Cloud(OPC)의 경우, 신원 도메인은 단일 Oracle Internet Directory(OID) 인스턴스의 사용자들 및 그룹 계층들에서 서로 다른(distinct) 서브-트리로서 표시된다.

[0089] 멀티테넌트 서버 환경에서, 신원 도메인들은 "테넌트들"과 "파티션들" 사이의 연결/정렬 포인트로서 서빙(serving)한다. 단일 신원 도메인은 테넌트의 사용자들을 표시한다(즉, 테넌트들과 신원 도메인들 간에 1-1 매핑이 존재한다). 이는, 특정한 파티션이 테넌트와 관련된 신원 도메인을 이용하도록 구성되는 경우, 효과적으로 테넌트에 할당되거나 또는 테넌트에 의해 소유될 수 있음을 의미한다. 즉, 테넌트의 사용자들은 파티션에 액세스할 수 있게 되고, 다른 테넌트의 사용자들은 상기 파티션에 액세스할 수 없게 된다.

[0090] 파티션 프라이머리 신원 도메인들(Partition Primary Identity Domains) - 신원 도메인들은 파티션된 환경에서 여러 목적들을 서빙한다. 맨 먼저, 이들은 서로 다른 파티션들과 관련된 사용자들을 구별한다. 각각의 파티션은 상기 파티션과 관련된 사용자들의 세트를 식별하는 프라이머리 신원 도메인(PIDD)으로 구성된다. 디폴트 액세스 정책들은 이 사용자들 - 그러나 다른 신원 도메인들로부터의 사용자들이 아닌 사용자들 - 이 파티션에 액세스할 수 있게 한다. 복수의 파티션들이 동일한 프라이머리 신원 도메인을 구성하는 것이 가능하지만, 이렇게 하는 것의 효과는 이 파티션들의 사용자들 간의 어떤 구별(distinction)을 제거하는 것이다. 공유된 신원 도메인으로부터의 사용자들은 공유 파티션들 모두에 액세스할 수 있게 된다. 프라이머리 신원 도메인은 또한, 롤 매핑 및 인가 정책들을 작성하기에 편리한 형태로 파티션의 리소스들의 "소유권"을 나타내도록 서빙한다. 파티션들의 리소스들을 마치 이들이 파티션의 신원 도메인에 의해 소유되는 것처럼 취급하는 것은 사용자의 신원 도메인과 리소스의 신원 도메인 간을 쉽게 비교할 수 있게 한다. 신원 도메인들이 사용중일 때, 하나의 신원 도메인은 도메인에 대한 어드미니스트레이티브 신원 도메인(AIDD: Administrative Identity Domain)으로서 지정되어야 한다. 이는 효과적으로는, 도메인에 대한 프라이머리 신원 도메인이다. 이는, 시스템 어드미니스트레이터들이 속하고 그리고 시스템 및 어드미니스트레이티브 리소스들의 "소유권"이 귀속되는 신원 도메인이다.

[0091] 도 7은 멀티-테넌트 서버 환경(100)을 도시하며, 이 멀티-테넌트 서버 환경에서 신원 도메인은 사용중이다. 도시된 바와 같이, 도 7에서, 파티션 A(250)의 구성(650)은 프라이머리 신원 도메인 PIDD A(754)의 식별을 포함한다. PIDD A(754)는 LDAP A(756)와 관련된다. 파티션 B(260)의 구성(660)은 프라이머리 신원 도메인 PIDD B(764)의 식별을 포함한다. PIDD B(764)는 LDAP B(766)와 관련된다. 도메인 레벨 보안 구성은 어드미니스트레이티브 신원 도메인 AIDD(704)를 참조한다. AIDD(704)는 LDAP(706)와 관련된다. 주목할 점으로서, 도 7에는 독립적인 LDAP 디렉토리들을 이용하여 예시되지만, 각각의 신원 도메인에 대한 사용자 저장소는 테넌트 필요들 또는 테넌트에 의해 제공되는 기존 사용자 저장소에 따라 구성가능하다. 사용자 저장소는 멀티-테넌트 환경(100) 내부에 있거나 혹은 외부에 있을 수 있다. 대안적으로는, PIDD A(754) 및 PIDD B(766)는 예컨대, LDAP(706)의 개별 서브트리들 또는 데이터베이스 내의 사용자 레코드들에 추가되는 신원 도메인 필드를 나타낼 수 있다. 프라이머리 신원 도메인 참조는 어떤 사용자 저장소가 프라이머리 신원 도메인과 관련되었든지 간에 콜을 처리하는 플러그가능한 인터페이스의 구성을 가능하게 한다. 서로 다른 파티션들은 서로 다른 타입의 사용자 저장소들을 이용할 수 있다.

[0092] 신원 도메인들은 적어도 하나의 신원 도메인이 멀티테넌트 서버 환경에서 구성되는 경우 "사용 중"인 것으로 고려된다. 신원 도메인들이 사용중일 때 - 프린서플들이 항상 신원 도메인 필드를 지닌(carry)다는 의미에서 신원 도멘인들은 항상 "인에이블(enable)"되고, 인증 제공자들은 항상 비-널(non-null)/비어있지 않은(non-empty) 신원 도메인 값들을 갖는 프린서플들을 생성할 수 있다. 신원 도메인들은 보안 구성 객체의 어드미니스트레이티브 신원 도메인 속성, 시스템 내의 각각의 파티션 객체의 프라이머리 신원 도메인 속성 및 시스템에 구성된 신원-도메인-인지 인증 제공자(들)의 신원 도메인 속성에 구성된다. 이 속성들 중 어느 것이 비-널이고, 비어있지 않은 값으로 설정되면, 신원 도메인들은 사용중인 것으로 고려된다.

[0093] 신원 도메인들이 사용중이면, 어드미니스트레이티브 신원 도메인은 설정(구성 검증에 의해 체크)되어야 하며, 매 파티션(every partition)은 (구성 검증에 의해 체크된) 프라이머리 신원 도메인을 구성해야 한다. 어떤 영역에 대해 시스템에 구성된 모든 롤 매핑, 인가, 크리덴셜 매핑 및 감사 제공자들은 (영역 시작 시간에 체크되는) 신원 도메인 인지 제공자 마커 인터페이스를 구현해야 한다. 인증 제공자들은 디폴트/어드민 영역 및 적절한 신원 도메인들로부터의 사용자들을 인증할 수 있는 각각의 파티션의 영역(이는 구성 검증에 의해 체크되지 않음)에 구성된다. 따라서, 도 7에 도시된 예를 이용하여, 영역 A(750)는 PIDD A(754)로부터의 사용자들을 인증할 수

있는 인증 제공자를 구성해야 하고, 영역 B(760)는 PIDD B(764)로부터의 사용자들을 인증할 수 있는 인증 제공자를 구성해야 한다. 마찬가지로, 디폴트 영역(702)은 AIDD(704)로부터의 사용자들을 인증할 수 있는 인증 제공자를 구성해야 한다.

[0094] 신원-도메인-인지 제공자들의 이용은 또한, 보안 구성 객체의 신원 도메인 인지 제공자 요구 속성(Identity Domain Aware Providers Required attribute)을 설정함으로서 강제될 수 있다. 이 속성을 참(true)으로 설정하는 것은 신원 도메인들이 시스템에 구성되지 않는 경우에도, 모든 롤 매핑, 인가, 크리덴셜 매핑 및 감사 제공자들이 신원 도메인 인지 제공자 인터페이스를 지원하게 할 것이다.

[0095] 신원 제공자(SSPI) 인터페이스 제공자들은 파티션된 환경에서 - 또는, 더욱 정확하게는, 신원 도메인들이 시스템에 구성되는 환경에서, 정확하게 기능하도록 신원-도메인-인지형(identity-domain-aware)이어야 한다. 예를 들어, 신원 도메인들을 이해하지 못하는 인가 제공자는 동일한 이름을 갖되 서로 다른 신원 도메인들을 갖는 두 사용자들 간을 정확하게 구별할 수 없으며, 그러므로, 유효한 인가 결정들을 내릴 수 없다. 신원 도메인들이 사용중일 때 또는 신원 도메인 인지 제공자 요구 속성이 참으로 설정될 때, 다음의 타입들의 모든 제공자들은 신원-도메인-인지형이 되어야 하고, 마커 인터페이스를 구현해야 한다.

[0096] • 롤 매핑

[0097] • 인가

[0098] • 크리덴셜

[0099] • 매핑 감사

[0100] 마커 인터페이스를 구현함과 함께, 신원-도메인-인지 제공자들은, 동일성에 대한 테스트를 할 때 신원 도메인을 고려하는 것(accounting for)과 그리고 신원 도메인을 고려하는 맵 키들을 구성하는 것을 포함하여, 사용자 및 그룹 프린서플들이 신원 도메인 값들을 지닐(carry) 때 상기 사용자 및 그룹 프린서플들을 적절하게 처리한다.

[0101] 인증

[0102] 상기 기술된 바와 같이, 신원 도메인(IDD)은 사용자들 및 그룹들에 대한 로직적 네임스페이스이다. 신원 도메인들은 사용자들의 서로 다른 세트들을 식별하고 구별하기 위해 이용된다. 신원 도메인들은 서로 다른 파티션들로부터의 사용자들을 구별하고 리소스들의 소유권을 귀속시키는 데 필수적이다. 그러므로, 이들은 파티션들 간을 구별할 수 있는 롤 매핑 및 인가 정책들을 위한 그리고 서로 다른 파티션들로부터의 사용자들 또는 리소스들 구별해야 하는 다른 서비스들을 위한 구현 기술이다. 따라서, 사용자들이 인증될 때, 이 사용자들이 특정한 신원 도메인에 관하여 인증되는 것이 필수적이다. 일례로서, 인증 메커니즘은, 파티션 B의 존 스미스(John Smith) 사용자로부터 구별되어야 하는 파티션 A의 존 스미스 사용자를 구별해야만 한다. 따라서, 사용자들은 사용자 이름, 및 사용자가 속하는 신원 도메인에 관하여 인증된다.

[0103] 시스템에 의해 제공되는 보안 서비스들은 "영역 기반" 또는 "비 영역 기반"을 특징으로 할 수 있다. 영역 기반 서비스들은 보안 영역 객체에 의한 구성에 표시되는 바와 같이 보안 "영역"에 대해 구성되는 그러한 서비스들이다. 인증은 영역 객체 상에 구성된 영역 기반 보안 서비스들이다.

[0104] 본 발명의 실시예들에서, 신원 도메인들을 활용하는 멀티테넌트 환경에서 프린서플들의 인증을 제공하기 위한 시스템이 제공된다. 신원 도메인들을 적절하게 나타내는 기본적인 타입들(콜백, 프린서플들)이 제공된다. 컨테이너들 및 어플리케이션들로 하여금 사용자의 신원 도메인들을 특정할 수 있게 하는 어플리케이션 프로그래밍 인터페이스들이 제공된다. 추가적으로, 서비스들 및 제공자들은 신원 도메인들에 대한 인증을 지원하는 특징들 및 결과적인 주체(subject)들 및 프린서플들의 적절한 처리를 포함한다.

[0105] 신원 도메인들을 활용하는 멀티테넌트 환경에서, 프린서플들은 사용자 이름 및 사용자가 등록되어 있는 신원 도메인에 의해 정의된다. 인터페이스는 신원 도메인 정보를 지니는 프린서플들을 식별하고, 관련된 신원 도메인을 획득하기 위해 메소드를 선언한다. 클래스가 이 인터페이스를 구현하기 위해 이용되고, 신원 도메인 정보를 지닌다. 프린서플의 이름 및 신원 도메인 모두가 비교들에 이용된다. 프린서플 서명들은 이름 및 신원 도메인 필드들 모두를 커버한다. 프린서플 팩토리 클래스(Principal Factory class)는 신원 도메인 값들을 갖는 사용자 및 그룹 프린서플들을 생성하는 것을 지원한다.

[0106] 두 개의 콜백들 즉, 신원 도메인 이용 콜백 및 신원 도메인 그룹 콜백이 제공되며, 이들은 각각, 관련 신원 도

메인을 갖는 사용자 이름 및 관련된 신원 도메인을 각각 갖는 그룹 이름들의 세트를 표시할 수 있다. 콜백 처리자들(callback handlers)은 새로운 콜백들을 처리하는 것을 지원하기 위해 제공된다. 함께 고려하면, 콜백들 및 콜백 처리자들은 어플리케이션들 및 컨테이너들이 인증하는 사용자들의 신원 도메인들을 특정할 수 있게 하고, 인증 서비스들 및 제공자들이 인증 동안 사용자들의 신원 도메인 정보에 액세스할 수 있게 한다.

[0107] 사용자 크리덴셜들을 콜백 처리자로서 취하는 어플리케이션 프로그래밍 인터페이스들은 관련된 신원 도메인을 갖는 사용자 이름을 표시할 수 있는 신원 도메인 사용자 콜백을 처리할 수 있는 콜백 처리자를 이용하여 신원 도메인들을 쉽게 지원할 수 있다. 콜백들 대신 사용자이름 스트링들을 이용하는 레거시 인증 API들은 신원 도메인 사용자 콜백 및 신원 도메인 그룹 콜백을 지원하기 위해 콜백 처리자 인수(argument)를 이용할 수 있도록 수정된다.

[0108] 서블릿 컨테이너(도 3 참조)의 폼 로그인 구현은 로그인 폼들(login forms)이 사용자의 신원 도메인을 수집(또는 그렇지 않으면 결정)하고 이를 표준 사용자 및 패스워드와 함께 컨테이너에 패스할 수 있도록 새로운 신원 도메인 파라미터를 지원하기 위해 확장된다. 따라서, 예를 들어 테넌트 A 환경(320)에 액세스하는 사용자는 사용자 ID 및 패스워드를 요하는 로그온 폼을 제공받을 수 있다. 서블릿 컨테이너는 디폴트 값(즉, 현재의 파티션의 프라이머리 신원 도메인 값)을 제공한다. 따라서, 예를 들어, 로그온 폼이 테넌트 A 환경(320)에 제공되기 때문에, 로그온 데이터는 파티션 A(250)의 PIDD A(754)와 자동으로 관련될 수 있다(도 7 참조). 사용자는 신원 도메인을 특정하도록 요구되지 않을 수 있다. 그러나, 서로 다른 신원 도메인 값이 특정되면, 이는 컨테이너에 의해 공급되는 디폴트 값을 오버라이드하는 바 - 그러나, 파티션 A 내의 로그온 폼은, 파티션에 대한 디폴트 신원 도메인이 아닌 신원 도메인의 선택을 허용하도록 구성되지 않을 수 있다.

[0109] 스트링 사용자이름 파라미터를 취하는 여러 어플리케이션 인증 인터페이스와 함께 컨테이너 인증은 파티션된 환경에서 콜될 때 신원 도메인을 자동으로 공급한다. 이러한 거동은 파티션에 대해 설정된 프라이머리 신원 도메인 값에 의존하며, 레거시 어플리케이션들이 파티션에서 실행될 때 정확하게 인증할 수 있게 한다. 어플리케이션들에 대한 변경은 요구되지 않는다. 디폴트 신원 도메인 거동이 적절하게 작동하게 하기 위해, 컨테이너 인증은 파티션이 구별되고 모든 프로토콜들/컨테이너들에 대하여 실행 컨텍스트에 이용가능해질 때까지 시도되지 않을 수 있다.

[0110] 신원 도메인들을 활용하는 멀티테넌트 시스템에서 주체들을 인증하기 위한 프로세싱 모델은 종래의 인증 기능과 유사하다. 따라서, 종래의 인증 시스템들 및 방법들은 인증 제공자들이 신원-도메인-인지형임을 보장하기 위해 추가적인 특징들과 조합하여 활용될 수 있다. 인증 제공자들은 자신이 인증할 수 있는 신원 도메인들을 인지하고 다른 신원 도메인들에 대해 인증하기 위한 요청들을 무시해야만 한다. 예를 들어, "Acme" 신원 도메인에 대해 인증하도록 구성된 제공자는 오직, Acme 사용자들을 인증하기 위한 요청에 응답하고, 다른 신원 도메인들로부터의 사용자들을 무시해야 한다. 전형적으로, 지원되는 신원 도메인들의 리스트는 제공자의 구성으로부터 비롯될 것이지만 - 따라서 어드미니스트레이터들은 시스템에 가시적인 신원 도메인들에 걸쳐 제어를 할 수 있음 - 이는 프로세싱 모델의 요건은 아니다. 단일 제공자는 하나의 신원 도메인 또는 많은 신원 도메인들에 대해 인증할 수 있으며, 또한 신원 도메인을 갖지 않는 사용자들을 인증할 수 있고 그러하도록 구성된 경우, 이 사용자들을 인증할 수 있다. 복수의 인증 제공자들이 구성될 수 있고, 이들 각각은 하나 이상의 신원 도메인들에 대해 각각 인증할 수 있다.

[0111] 도 7을 참조하면, 예를 들어, 영역 A(750), 영역 B(760) 및 디폴트 영역(702) 각각은 자신들의 인증 제공자를 구성할 수 있다. 도시된 바와 같이, 영역 A(750)는 인증 제공자 Auth A(758)를 구성하고, 영역 B(760)는 인증 제공자 Auth B(768)를 구성하고, 디폴트 영역(702)은 인증 제공자 Auth(708)를 구성한다. 또는, 예를 들어, 단일 인증 제공자가 구성될 수 있고, 정확한 신원 도메인에 관하여 사용자들이 인증되게 하기 위해 신원 도메인 정보를 수신하고 이용한다.

[0112] Auth A(758) 및 Auth B(768)과 같은 인증 제공자들은 자신들이 지원하는 신원 도메인(들)을 식별한다. 인증 제공자들은 구성된 인증 제공자 인스턴스 당 단 하나의 신원 도메인을 지원한다. SSPI 인터페이스 - 신원 도메인 인증자 - 는 각각의 인증 제공자가 자신들이 지원하는 신원 도메인(들)을 구성/선언할 수 있게 한다. 신원 도메인 사용자들을 인증할 수 있는 인증 제공자들은 인터페이스를 구현하고 신원 도메인 콜백들을 처리한다.

[0113] 인증 제공자들은 자바 인증 및 인가 서비스(JAAS) 프로세싱 모델을 활용하여 구현될 수 있다. JAAS는 두 가지 목적들 즉, 코드가 어플리케이션, 애플릿, 빈 또는 서블릿으로서 실행되는지에 관계없이 누가 현재 자바 코드를 실행하고 있는지를 신뢰성있게 그리고 보안되게 결정하기 위한 사용자들의 인증을 위해, 그리고 사용자들이 수행되는 동작들을 행하기 위해 요구되는 액세스 제어권들(허락들)을 가지게 하기 위한 사용자들의 인가를 위해

이용될 수 있다. JAAS 인증은 플러그가능한 방식으로 수행된다. 이는 자바 어플리케이션들이 기저 인증 기술들과는 독립적으로 유지되게 한다. 새로운 또는 갱신된 기술들이 어플리케이션 자체에 대한 수정들을 요함이 없이 플러그 인 될 수 있다. 이용될 특정 인증 기술에 대한 구현은 런타임 시 결정된다. 구현은 로그인 구성 파일에 특정된다.

[0114] 인증 제공자들은, 영역 상에 구성되고 각각의 인증 제공자에 대해 설정된 JAAS 제어 플래그들와 일관된 순서로 콜된다. 상기 기술된 프로세싱 모델을 고려하여 볼 때, 신원 도메인 인증자들에 대한 대부분의 적절한 구성은 일반적으로, 각각의 제공자가 a) 신원 도메인들의 개별적인, 비-중첩 세트를 처리하고, b) "SUFFICIENT" JAAS 제어 플래그와 함께 구성되는 것이다. 다른 구성들이 허용되지만, 타겟 신원 도메인에 기초하여 인증 요청들을 처리 또는 무시하는 것을 선택할 수 있는 인증 제공자들과 JAAS 제어 플래그들 사이의 인터랙션(interaction)에 비추어 신중하게(carefully) 고려되어야 한다.

[0115] 도 8은 하나의 실시예에 따른 복수의 신원 도메인들을 갖는 멀티-테넌트 환경에서 로그인 모듈들에 의해 구현되는 방법을 도시한다. 복수의 신원 도메인들을 갖는 멀티테넌트 서버 환경들에서 인증 제공자들의 로그인 모듈들은, 도 8에 도시된 방법을 이용하여 콜백들을 처리하는 것을 시도함으로써 인증 요청을 처리할지를 결정한다. 단계(810)에서, 콜백은 영역 상에 구성된 제1 인증 제공자에게 제공된다. 단계(812)에서, 인증 제공자는 신원 도메인 사용자 콜백을 처리하는 것을 시도한다. 단계(814)에서, 신원 도메인 사용자 콜백이 리턴되는 경우, 인증 제공자는 이로부터 사용자의 신원 도메인을 획득한다. 단계(816)에서, 사용자의 신원 도메인이 이 인증자에 의해 처리되는 경우, 사용자를 인증한다. 패스워드 콜백이 사용자의 패스워드를 획득하기 위해 이용된다. 단계(818)에서, 사용자의 신원 도메인이 이 인증자에 의해 처리되지 않는 경우, 요청을 무시하기 위해 "거짓"을 리턴한다. 단계(820)에서, 인증 제공자가 신원 도메인을 갖지 않는 사용자들을 처리하도록 구성되는 경우, 인증 제공자는 이름 콜 백을 처리하는 것을 시도한다. 단계(822)에서, 이름이 발견되지 않으면, 요청을 무시하기 위해 "거짓"을 리턴한다. 단계(824)에서, 이름이 발견되면, 사용자를 인증한다. 패스워드 콜백은 사용자의 패스워드를 획득하기 위해 이용된다. 단계(826)에서, 인증 제공자가 사용자를 인증할 수 없는 경우, 시스템은 영역 상에 구성된 다음 인증 제공자를 (존재하는 경우) 트라이한다. 제1 로그인 모듈이 요청을 무시하는 경우, 요청은 상기 요청을 처리하는 것을 시도하는 다른 로그인 모듈에 패스된다. 인증 제공자들은 요청이 처리되거나 또는 처리 실패(fail)할 때까지 영역 상에 구성된 순서로 콜된다.

[0116] 신원 어서션 제공자들(identity assertion providers)은 사용자이름 및 패스워드가 아닌 토큰들 또는 크리덴셜들, 예컨대 SAML 어서션들, Kerberos 티켓들 또는 쿠키 기반 세션 토큰들에 기초하여 사용자들을 인증하는 인증 제공자들이다. 이들은 콜백을 처리하지 않는다. 대신, 이들은 사용자의 토큰을 검증하고, 이의 컨텐츠를 검사함으로써 사용자의 신원 - 사용자이름, 그룹 멤버쉽, 등등 - 을 결정한다. 획득된 사용자 정보는 콜백 처리자의 형태로 리턴되는 바, 이는 궁극적으로는 사용자의 주체에 대해 적절한 프린서플들을 채우기(population) 위해 로그인 모듈 체인에 패스된다.

[0117] 다른 인증 제공자들과 마찬가지로, 파티션된 환경에서 이용되는 신원 어서션 제공자들은 신원-도메인-인지형이 되도록 구성되고 자신의 신원 도메인 지원을 선언하기 위해 적절한 신원 도메인 인증자 인터페이스를 구현한다. 이들은, 구성으로부터 또는 어서트된(asserted) 토큰으로부터, 사용자에 대해 (그리고 잠재적으로는, 어떤 어서트된 그룹들에 대해) 어서트할 신원 도메인을 결정할 수 있다. 신원 도메인을 갖는 사용자를 어서트할 때, 리턴된 콜백 처리자는 신원 도메인 사용자 콜백 및, 옵션에 따라서는 신원 도메인 그룹 콜백을 처리하도록 구성된다.

[0118] 로그인 모듈들은 바람직하게는, 프린서플들을 생성하기 위해 프린서플 팩토리를 이용한다. 팩토리는 신원 도메인 정보를 갖는 프린서플들을 생성하기 위한 메소드들을 제공한다. 프린서플들을 직접적으로 인스턴스화하는 경우, 신원 도메인은 적절한 구성자(constructor) 또는 설정자(setter) 메소드(들)을 이용하여 제공되어야 한다. 프린서플 검증 서비스는 인증 시간에 모든 프린서플에 서명하고, 프린서플들이 로컬 도메인에 의해 본래 인증되었음을 보장하기 위해 시스템으로 유입되는 프린서플들의 서명들을 검증한다. 실제 서명/검증은 플린서플 검증자 제공자에 위임된다.

[0119] 프린서플 검증은 영역 기반 서비스인 바, 이는 서로 다른 영역들이 서로 다르게 프린서플들을 검증할 수 있음을 의미한다. 그렇게 하는 것은 하나의 영역에 의해 인증된 프린서플들이 다른 영역에 의해 인증된 프린서플들과 구별될 수 있게 한다. 이러한 거동이 잠재적으로는 많은 시나리오들에서 유용할 수 있지만, 이는 필수적인 것은 아니다. 예를 들어, 멀티테넌트 시스템은 도메인 상에 구성되고 따라서 범주가 도메인 범위인 단일 서명 키 - "도메인 크리덴셜" - 을 이용할 수 있다. 따라서, 프린서플들은 영역이 이들을 인증했는지에 관계 없이 동일한

키로 서명된다. 대안적인 실시예들에서, 서로 다른 서명 키들이 서로 다른 영역들에 의해 대신 활용될 수 있다.

[0120] 프린서플 및 신원 어서션 캐시들. 각각의 영역은 프린서플 캐시 및 신원 어서션 캐시를 유지한다. 전자는, 서명이 상대적으로 고가의 동작이기 때문에 동일한 프린서플을 반복적으로 서명하는 것을 회피하기 위해 이용되는 서명된 프린서플들의 캐시이다. 캐시 키는 바람직하게는, 서로 다른 신원 도메인들로부터의 프린서플들을 혼동하지 않도록 사용자 이름 및 신원 도메인이다. 후자는 신원 어서션으로부터 비롯되는 주체들의 캐시이다. 이 캐시는 동일한 사용자 신원이 반복적으로 어서트될 때 사용자 정보(그룹 멤버쉽, 등등)를 페치하기 위해 반복적으로 LDAP(또는 다른 사용자 저장소들)에 쿼리(query)하는 것을 회피하도록 구현된다. 프린서플 캐시와 마찬가지로, 캐시 키는 바람직하게는, 서로 다른 신원 도메인들로부터의 프린서플들을 혼동하지 않도록 사용자의 이름 및 사용자의 신원 도메인이다.

[0121] 인가

[0122] 하나의 실시예에서, 멀티테넌트 서버 환경은, 롤 매핑 서비스 및 인가 서비스로 구성된 인가 서브시스템에 의해 지원되는 롤 기반 액세스 제어(RBAC) 모델을 구현한다. 두 서비스들은 대체로 독립적이지만, 인가 서비스는 특정한 식별된 리소스에 관한 인가 결정을 내릴 때 사용자의 롤들을 결정하기 위해 롤 매핑 서비스를 콜한다. 롤 매핑 서비스 및 인가 서비스 모두는 실제 롤 매핑 및 인가 기능을 제공하는 보안 지원 제공자들에게 위임한다.

[0123] 시스템에 의해 제공되는 보안 서비스들은 "영역 기반" 또는 비 영역 기반"을 특징으로 할 수 있다. 영역 기반 서비스들은 보안 영역 객체에 의한 구성에 표시되는 바와 같은 보안 "영역"에 대해 구성되는 그러한 서비스들이다. 인가는 영역 객체 상에 구성되는 영역 기반 보안 서비스들이다.

[0124] 보안 지원 제공자들은 롤 매핑 및 인가 룰(rule)들에 이용될 수 있는 다수의 정책 "서술부들(predicates)"을 구현한다. 이 서술부들은 특정한 조건이 충족되면, 예컨대 주체의 프린서플들이 특정한 사용자 또는 그룹 이름과 매치되거나 또는 특정한 IP 어드레스로부터 비롯되는 요청과 매치되면, 롤을 허여(grant)하거나 또는 리소스에 대한 액세스를 직접적으로 허여한다. 서술부들은 논리적 AND, OR 및 NOT 시멘틱들을 이용하여 조합될 수 있다. RBAC에 대한 지원이 사용자가 특정된 롤을 가지는 경우 액세스를 허여하기 위해 인가 룰들에 이용되는 롤 서술부에 의해 제공된다. RBAC 모델에서, 액세스의 프라이머리 결정요인(determinant)은 사용자가 소정 롤을 가지는지 또는 가지지 않는지의 여부이다.

[0125] 롤 매핑에 가장 자주 이용되는 서술부는 그룹 서술부이며, 이는 특정한 그룹 내의 멤버쉽에 기초하여 롤을 허여한다. 웹로직 어드미니스트레이티브 롤 매핑이 이러한 서술부를 이용하고, 예컨대, "어드미니스트레이터들" 그룹의 멤버들에게 "어드민" 롤을 허여한다. (특정 사용자 이름과 매칭되는) 사용자 서술부의 이용은 덜 일반적이다. IP 어드레스 및 하루 중의 시간과 같은 추가적인 서술부는 전형적으로 컨텍스트에 기초하여 롤을 추가로 제약(constrain)하기 위해 이용된다. 예를 들어, "Bank Teller" 롤은 "Tellers" 그룹의 멤버들에게 허여될 수 있지만, 오직 은행이 업무를 위해 열려 있는 시간 동안에만 허여된다.

[0126] 파티션된 환경에서, 액세스 제어는 추가적인 국면을 띈다. 사용자들, 그룹들 및 리소스들은 이제 파티션들과 관련되고, 액세스 결정들은, 사용자의 신원 및 롤에 추가적으로, 사용자의 신원 도메인 및 리소스가 속한 파티션 모두를 고려한다. 두 개의 새로운 특징들/거동들은 파티션-인지 액세스 제어를 가능하게 한다. 첫째로, 신원 도메인을 고려하는 사용자들 및 그룹들을 비교하는 새로운 서술부들이 제공된다. 둘째로, 신원 도메인들은 리소스의 파티션 소유권을 나타내기 위해 이용된다. 따라서, 사용자들 및 리소스들 모두가 신원 도메인들과 관련된다. 따라서, 인가 결정이 특별한 사용자가 신원 도메인과 관련된 리소스에 액세스하기 위해 정확한 신원 도메인의 정확한 롤을 가지는지를 판단하는 인가 결정이 인가 제공자에 의해 이루어질 수 있다.

[0127] 리소스들은 예컨대, 도 3에 도시된 바와 같이, 실제로 파티션들에 속하며 신원 도메인들에 속하지 않는다. 그러나, 리소스 소유권을 신원 도메인으로서 표시하는 것은 장점적인 바, 그 이유는 이렇게 하는 것이 인가 결정을 내릴 때 리소스 신원 도메인들과 사용자/그룹 신원 도메인들 간의 직접적인 비교를 제공하기 때문이다. 단순한 스트링 비교가 두 신원 도메인들이 매치하는지 매치하지 않는지를 결정할 수 있다. 파티션들이 "매치"를 결정하기 위해 신원 도메인들에 대해 비교된 접근법은 제약적인 명명 규약들(naming conventions), 매핑 테이블들 또는 두 가지 모두를 요할 수 있고, 런타임 시 덜 효율적일 수 있다. 파티션에 대한 신원 도메인 매핑은 파티션 객체의 프라이머리 신원 도메인 속성 상에 구성된다(예컨대, 도 7의 PIDD A(754) 및 PIDD B(764) 참조). 리소스 소유권을 나타내기 위해 신원 도메인을 이용하는 것의 추가적인 장점은 모델을 파티션들에 결부시키지 않는다는 점이며 - 따라서 신원 도메인들은 또한 비 파티션된 환경에서 리소스 인가의 제어를 위해 이용될 수 있다.

[0128] 다음 서술부들은 신원-도메인-인지 사용자 및 그룹 매칭에 대한 지원을 제공한다. 서술부들은 롤 매핑 및 인가

룰들에 이용가능하지만, 웹로직의 RBAC 모델과 일관된 롤 매핑을 위해 주로 이용되도록 기대된다.

표 1

[0129]

서술부	룰 표현	설명
IDDUser(user, idd)	User.name == user && User.idd == idd	특정된 신원 도메인으로부터 특정된 사용자와 매치. 기존 사용자 서술부와 유사하되, 오직 사용자가 특정된 신원 도메인으로부터의 사용자인 경우, 매치
IDDGroup(group, idd)	Group.name == group && Group.idd == idd	특정된 신원 도메인으로부터 특정된 그룹과 매치. 기존 그룹 서술부와 유사하되, 오직 그룹이 특정된 신원 도메인으로부터의 도메인인 경우, 매치
OwnerIDDUser(user)	User.name == user && User.idd == Resource.idd	사용자의 신원 도메인이 리소스 소유자의 신원 도메인과 매치되는 경우 특정된 사용자와 매치
OwnerIDDGroup(group)	Group.name == group && Group.idd == Resource.idd	그룹의 신원 도메인이 리소스 소유자의 신원 도메인과 매치되는 경우 특정된 그룹과 매치
AdminIDDUser(user)	User.name == user && User.idd == Admin.idd	사용자의 신원 도메인이 어드미니스트레이티브 신원 도메인과 매치되는 경우 특정된 사용자와 매치
AdminIDDGroup(group)	Group.name == group && Group.idd == Admin.idd	그룹의 신원 도메인이 어드미니스트레이티브 신원 도메인과 매치되는 경우 특정된 그룹과 매치

[0130] 도 9는 하나의 실시예에 따르면 복수의 신원 도메인들을 갖는 멀티-테넌트 환경에서 구현되는 인가 서브시스템을 도시한다. 도 9에 도시된 바와 같이, 신원 도메인(912)과 관련된 사용자(910)는 신원 도메인(916)과 관련된 리소스(914)에 액세스하는 것을 시도한다. 랩퍼 클래스(wrapper class)는 이 값들을 특정하기 위해 이용된다. 랩퍼 클래스는 필요한 경우 현재의 파티션을 결정하고, 공급된 소유권 값을 롤 매핑/인가 제공자들에 의해 이용하기 위한 리소스 신원 도메인 값으로 변환할 것이다. 랩퍼는 또한, 정확한 리소스 신원 도메인 값이 공급됨을 보장한다. 리소스에게로의 액세스는 롤 매핑 서비스(920) 및 인가 서비스(930)를 포함하는 인가 서브시스템(900)에 의해 제어된다. 롤 매핑 서비스(920) 및 인가 서비스(930)는 실제 롤 매핑 및 인가 기능을 제공하는 보안 지원 제공자들(940)에게 위임한다. 보안 지원 제공자들(940)은 룰들을 제공하는 서술어들(942)을 구현하는 바, 이 서술어들은 사용자가 콜되는 리소스에 액세스하기 위한 롤을 하는지의 여부뿐만 아니라 사용자 IdD(912)가 리소스 IdD(916)와 매치되는지의 여부를 결정할 수 있게 한다. 리소스를 콜하는 프로세스들은 리소스 - 이 리소스에 대한 액세스가 요청됨 - 의 소유권을 식별하는 리소스 신원 도메인 값을 제공한다. 따라서, 인가 제공자들은 신원 도메인 인지 서술어들을 이용하여 인가 결정을 내리기 위해 사용자 신원, 사용자 신원 도메인 및 리소스 및 리소스 신원 도메인에 대한 사용자 롤을 비교하도록 된다.

[0131] 파티션 보안은 하나의 새로운 메소드, 세 개의 새로운 컨텍스트 처리자 속성들 및 컨텍스트 처리자 랩퍼 클래스를 인가 및 롤 매핑 API들에 추가한다. 롤 관리자 인터페이스 상의 새로운 User In Role() 메소드는 리소스 및 컨텍스트 처리자를 쿼리에 대한 입력으로서 패스하여 콜러로 하여금 사용자가 특정한 롤을 하는지를 직접적으로 결정할 수 있게 한다. 세 개의 새로운 컨텍스트 처리자 속성들은 콜러로 하여금 인가 및 롤 매핑 API들을 콜할 때 리소스들에 대한 리소스 소유권 정보를 특정할 수 있게 하는 바, RESOURCE_PARTITION (파티션 이름)은 리소스에 대한 소유 파티션을 나타낸다. RESOURCE_OWNERSHIP_DEFAULT는 콜러 컨텍스에 의해 결정된 현재의 파티션이 리소스를 소유함을 나타낸다. RESOURCE_IDENTITY_DOMAIN는 콜러로 하여금 리소스 소유자로서 파티션 신원 도메인을 특정할 수 있다. 새로운 컨텍스트 처리자 랩퍼 클래스는 리소스 IDD 컨텍스트 랩퍼이고, 이는 콜러들이 적절한 신원 도메인 소유권 정보를 갖는 기존 컨텍스트 처리자 인스턴스를 데코레이션(decoration)하기 위해 이용할 수 있다.

[0132] 복수의 영역들의 논의에서 주목할 점으로서, 컨테이너들은 런타임 시 정확한 영역에 요청들을 안내(direct)하는 프록시 서비스들을 통해 보안 서비스들과 인터랙션한다. 인가 관리자 및 롤 관리자의 경우, 프록시는 리소스 타입 및 소유권을 고려하는 로직을 이용하여 정확한 영역을 선택할 것이다. 현재의 파티션에 의해 소유되는 리소스들 - 예컨대, 어플리케이션 리소스들 - 에 대해, "로컬한" 영역이 이용될 것이다. 예컨대, 시스템 리소스들에 대해, 디폴트/글로벌 영역이 이용될 것이다. 그 다음, 이는 리소스 타입에 따라 로컬 파티션의 영역에 대해 또는 디폴트/글로벌 영역에 대해 인가/롤 매핑 서비스를 콜할지를 결정할 것이다.

[0133] 신원-도메인-인지 서술어들의 이용은 가능한 범위의 롤(possible scoped role)들 - 특정한 신원 도메인의 컨텍스트에서만 유효한 롤들 - 을 만든다. 소유자 IDD 그룹 서술어를 이용하여, 파티션 어드민 롤은 리소스 - 이 리소스에 대한 액세스가 요청됨 - 를 소유하는 신원 도메인에 대한 파티션 어드미니스트레이터 그룹의 멤버들인 사용자들에게 허여될 수 있다. (리소스는 롤 쿼리들뿐만 아니라 인가 쿼리들에 대한 파라미터이다). 테넌트 어드민 롤이 유사한 방식으로 허여된다. 다음의 새로운 롤 정책들은 파티션된 도메인들에서 영역들에 제공(provision)될 것인 바, 주목할 점으로서 테넌트 기반의 롤 매핑들은 이들을 필요로 하는 계층화된 컴포넌트들에 의해 제공될 수 있다.

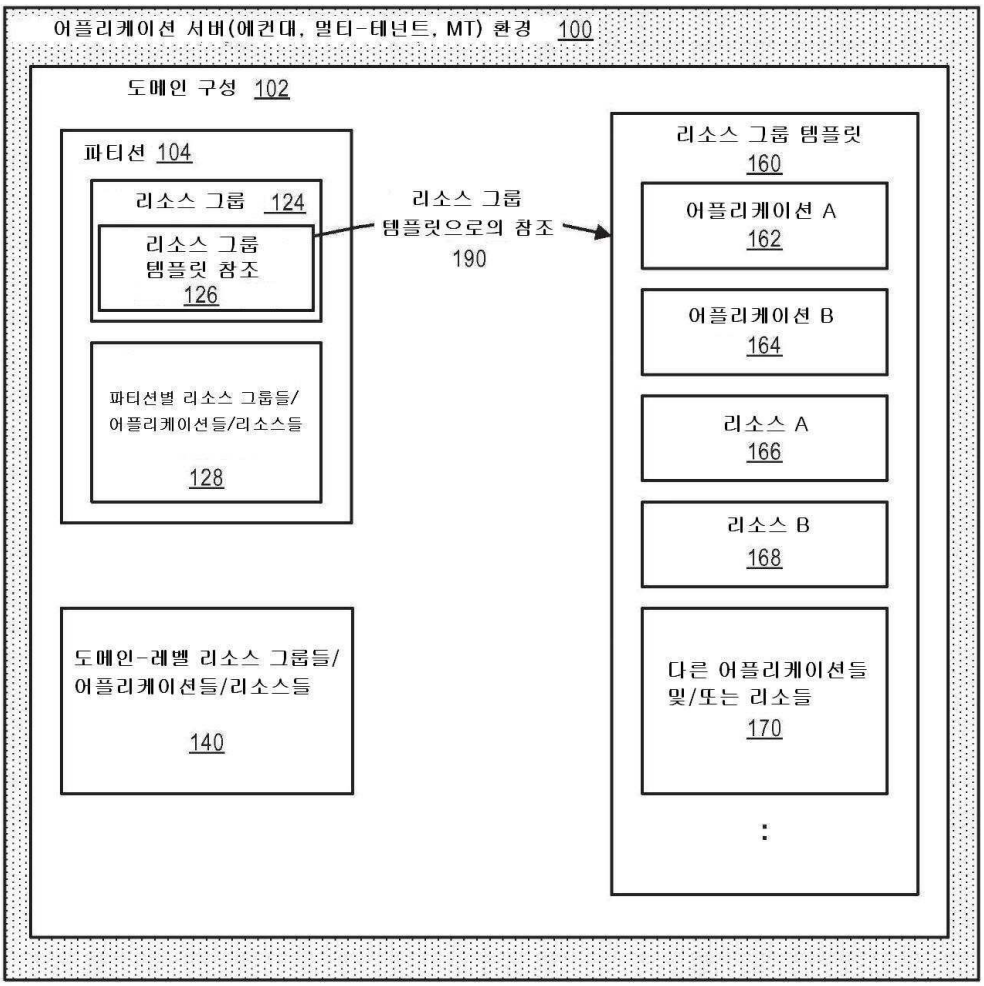
[0134] 본 발명은 본 발명의 교시들에 따라 프로그램된 하나 이상의 프로세서들, 메모리 및/또는 컴퓨터 판독가능 저장 매체들을 포함하여 하나 이상의 종래의 범용 또는 특수용 디지털 컴퓨터, 컴퓨팅 디바이스, 머신 또는 마이크로프로세서를 이용하여 편리하게 구현될 수 있다. 적절한 소프트웨어 코딩이 소프트웨어 분야의 숙련자들에게 분명할 바와 같이 본 발명의 교시들에 기초하여 숙련된 프로그래머들에 의해 쉽게 준비될 수 있다.

[0135] 일부 실시예들에서, 본 발명은 컴퓨터 프로그램 물을 포함하는 바, 상기 컴퓨터 프로그램 물은 명령어들이 저장된/본 발명의 프로세스들 중 어느 것을 수행하도록 컴퓨터를 프로그래밍하기 위해 이용될 수 있는 저장 매체 또는 컴퓨터 판독가능 매체(매체들)이다. 저장 매체는 이들로만 한정되는 것은 아니지만, 플로피 디스크(disk)들, 광학 디스크(disc)들, DVD, CD-ROM들, 마이크로드라이브 및 자기-광학 디스크(disk)들을 포함하는 어떤 타입의 디스크, ROM들, RAM들, EPROM들, EEPROM들, DRAM들, VRAM들, 플래시 메모리 디바이스들, 자기 또는 광학 카드들, (분자 메모리 IC들을 포함하는)나노시스템들 또는, 명령어들 및/또는 데이터를 저장하기에 적절한 어떤 타입의 매체 또는 디바이스를 포함할 수 있다.

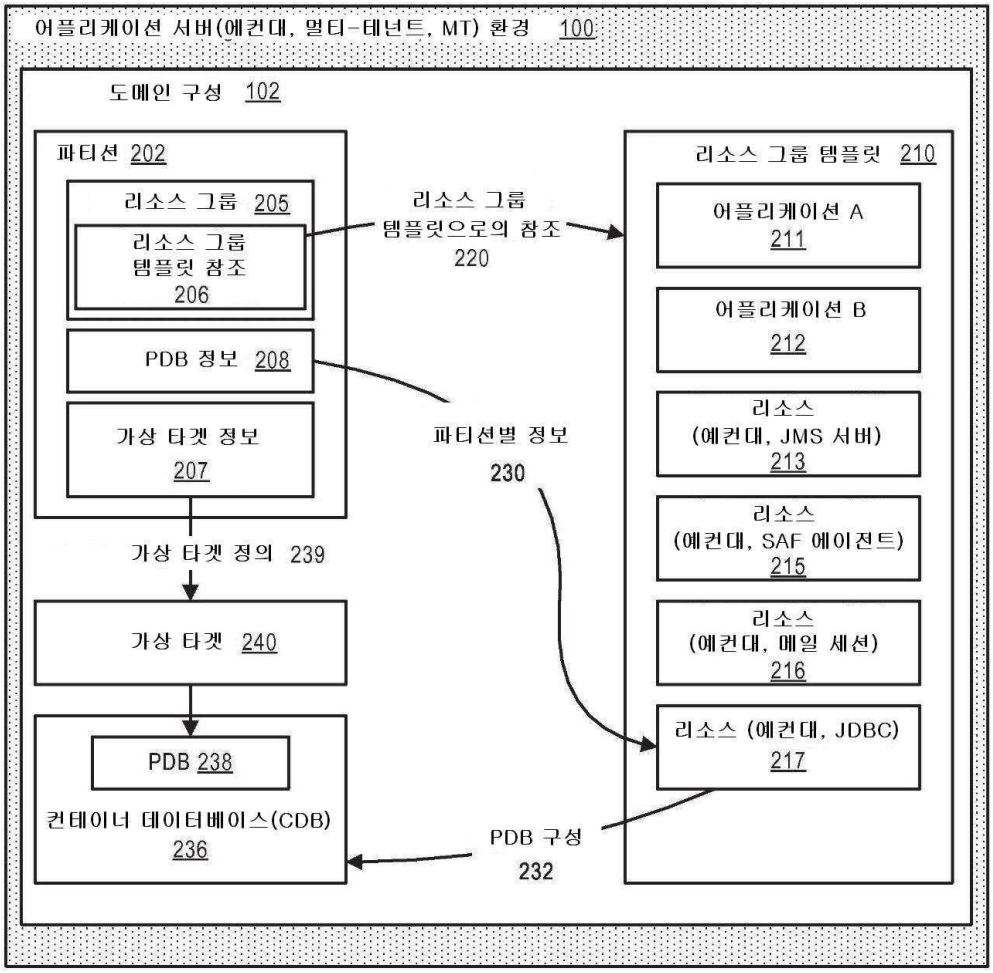
[0136] 본 발명의 상기 설명은 예시 및 설명을 목적으로 제공되었다. 본 설명은 완전한 것(exhaustive)으로 의도되거나 정확히 개시된 형태들로만 본 발명을 제한하고자 의도된 것이 아니다. 많은 수정들 및 변형들이 이 기술분야의 숙련자에게 분명할 것이다. 위 실시예들은 본 발명의 원리 및 이의 실용적 응용을 가장 잘 설명하기 위해 선택 및 기술되었으며, 그럼으로써 이 기술분야의 숙련자들은 본 발명에 대한 다양한 실시예들 및 고려되는 특별한 사용에 적합한 다양한 수정들을 이해할 수 있다. 본 발명의 범위는 다음의 특허 청구 범위 및 이의 균등물에 의해 한정되어야 함이 의도된다.

도면

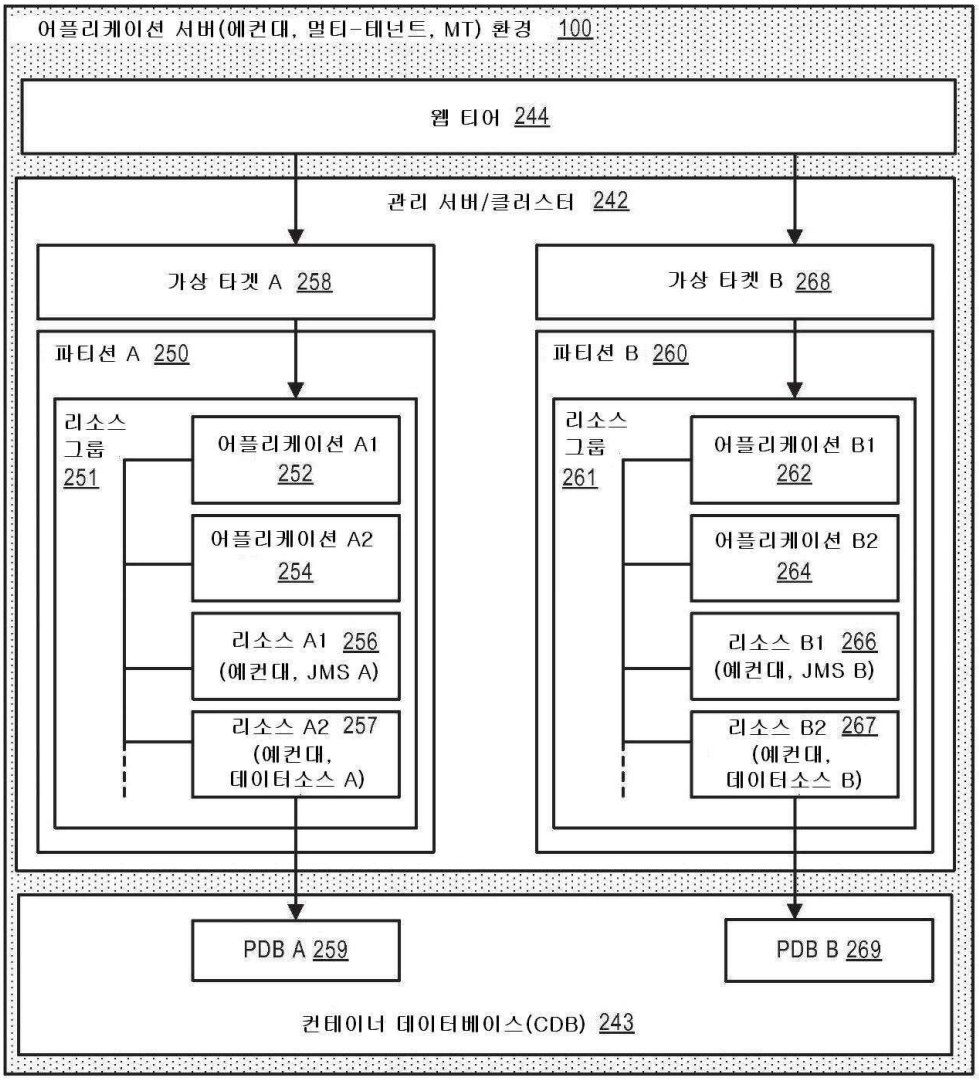
도면1


어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
도메인 구성 102
파티션 104
리소스 그룹 124
리소스 그룹 템플릿 참조 126
파티션별 리소스 그룹들/어플리케이션들/리소스들 128
도메인-레벨 리소스 그룹들/어플리케이션들/리소스들 140
리소스 그룹 템플릿으로의 참조 190
리소스 그룹 템플릿 160
어플리케이션 A 162
어플리케이션 B 164
리소스 A 166
리소스 B 168
다른 어플리케이션들 및/또는 리소스들 170
:

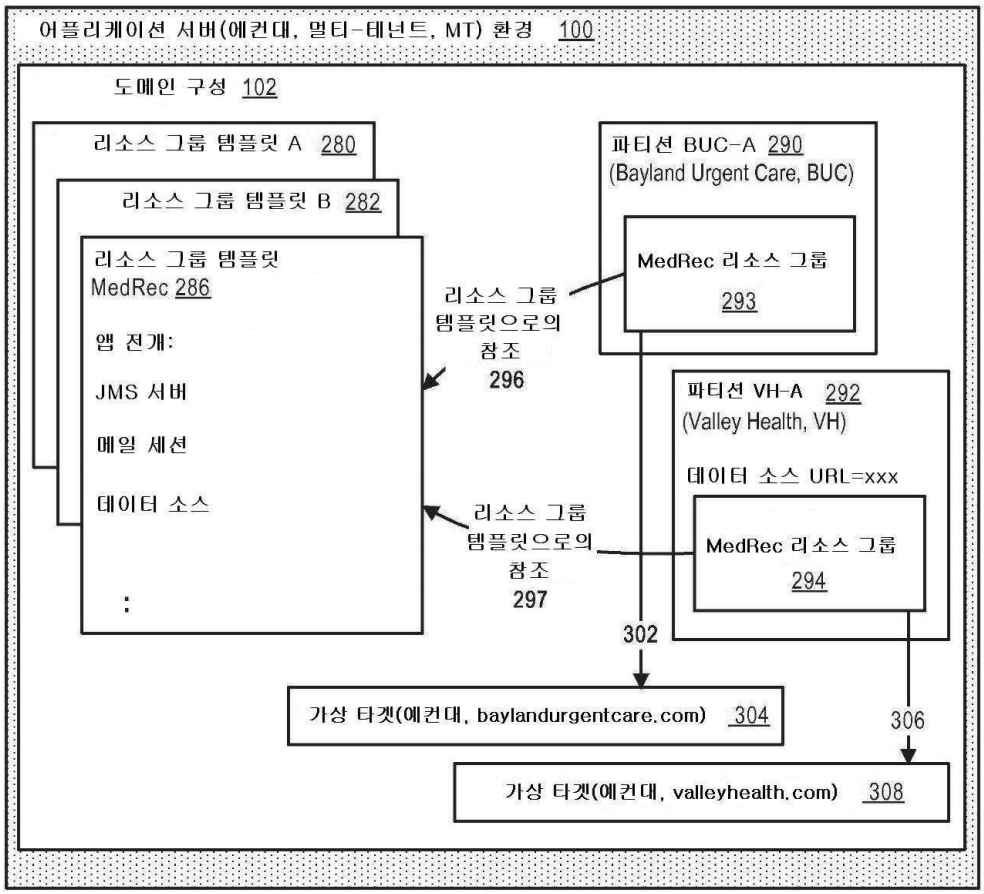
도면2


어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
도메인 구성 102
파티션 202
리소스 그룹 205
리소스 그룹 템플릿 참조 206
PDB 정보 208
가상 타겟 정보 207
가상 타겟 정의 239
가상 타겟 240
PDB 238
컨테이너 데이터베이스(CDB) 236
리소스 그룹 템플릿으로의 참조 220
파티션별 정보 230
PDB 구성 232
리소스 그룹 템플릿 210
어플리케이션 A 211
어플리케이션 B 212
리소스 (에컨대, JMS 서버) 213
리소스 (에컨대, SAF 에이전트) 215
리소스 (에컨대, 메일 세션) 216
리소스 (에컨대, JDBC) 217

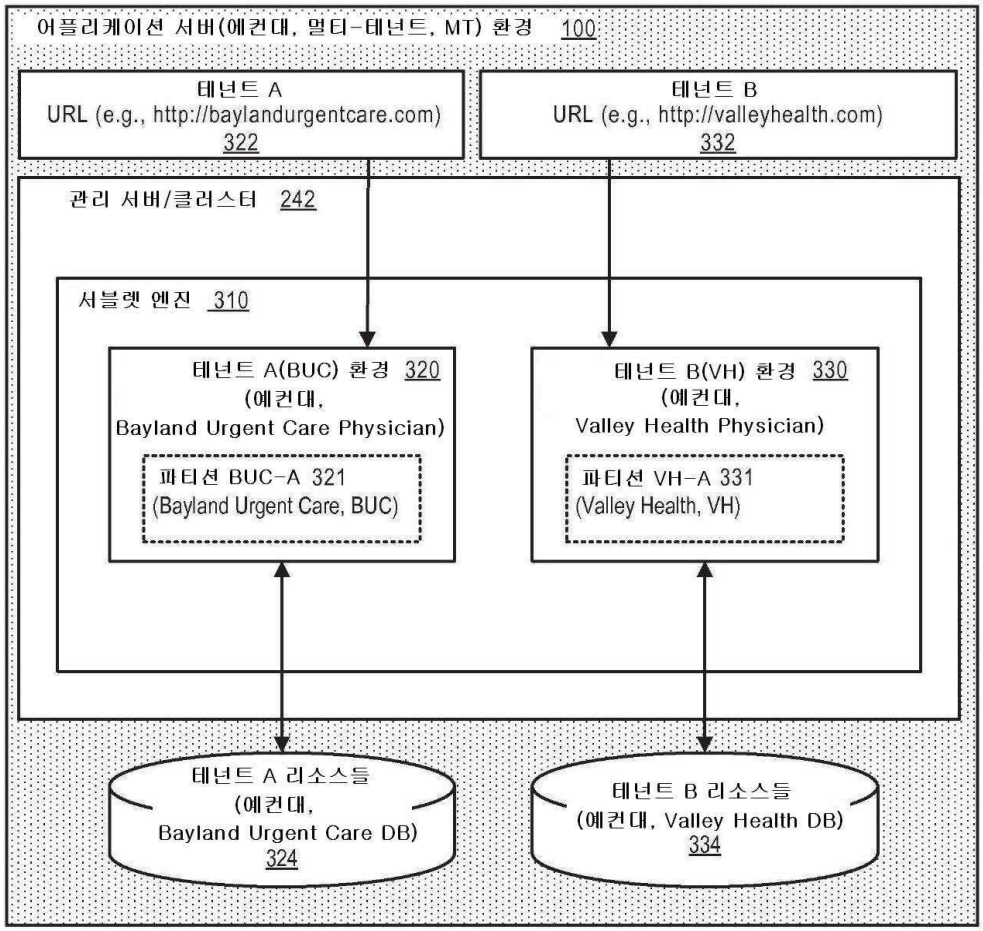
도면3


어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
웹 티어 244
관리 서버/클러스터 242
가상 타겟 A 258
가상 타겟 B 268
파티션 A 250
파티션 B 260
리소스 그룹 251
어플리케이션 A1 252
어플리케이션 A2 254
리소스 A1 256 (에컨대, JMS A)
리소스 A2 257 (에컨대, 데이터소스 A)
리소스 그룹 261
어플리케이션 B1 262
어플리케이션 B2 264
리소스 B1 266 (에컨대, JMS B)
리소스 B2 267 (에컨대, 데이터소스 B)
PDB A 259
PDB B 269
컨테이너 데이터베이스(CDB) 243

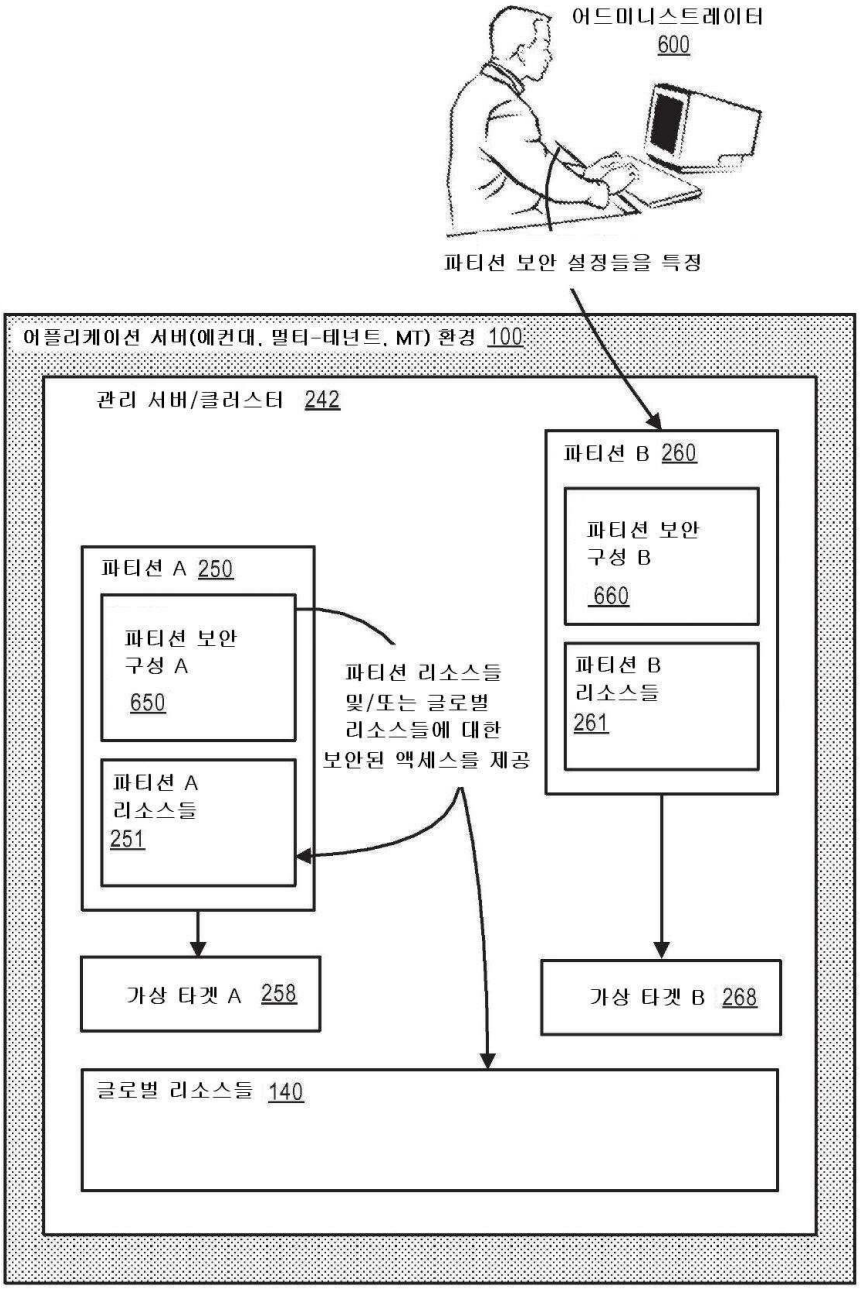
도면4


어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
도메인 구성 102
리소스 그룹 템플릿 A 280
리소스 그룹 템플릿 B 282
리소스 그룹 템플릿
MedRec 286
앱 전개:
JMS 서버
메일 세션
데이터 소스
:
리소스 그룹
템플릿으로의
참조
296
파티션 BUC-A 290
(Bayland Urgent Care, BUC)
MedRec 리소스 그룹
293
파티션 VH-A 292
(Valley Health, VH)
데이터 소스 URL=xxx
MedRec 리소스 그룹
294
리소스 그룹
템플릿으로의
참조
297
302
306
가상 타겟(에컨대, baylandurgentcare.com) 304
가상 타겟(에컨대, valleyhealth.com) 308

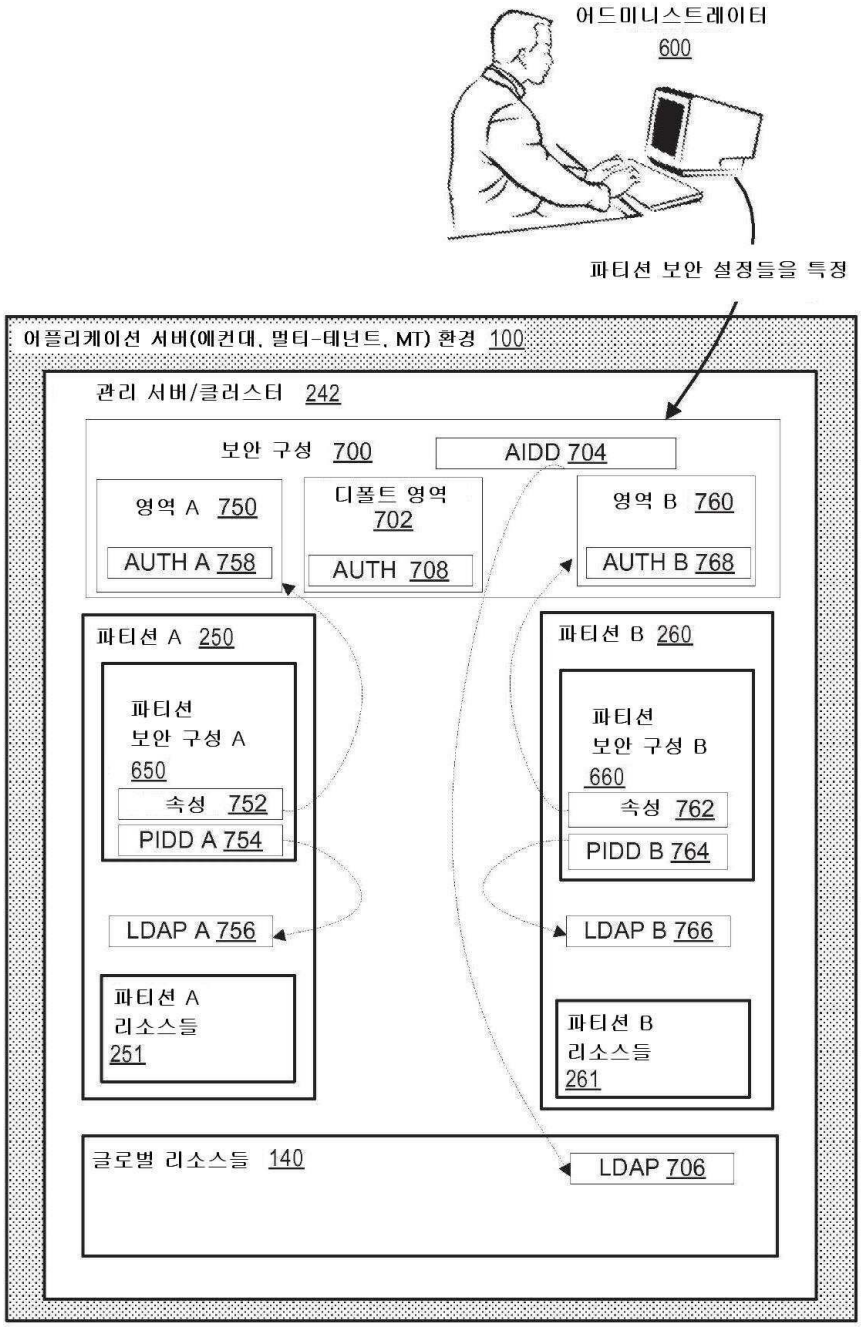
도면5


어플리케이션 서버(예컨대, 멀티-테넌트, MT) 환경 100
테넌트 A
URL (e.g., http://baylandurgentcare.com)
322
테넌트 B
URL (e.g., http://valleyhealth.com)
332
관리 서버/클러스터 242
서블렛 엔진 310
테넌트 A(BUC) 환경 320
(예컨대,
Bayland Urgent Care Physician)
파티션 BUC-A 321
(Bayland Urgent Care, BUC)
테넌트 B(VH) 환경 330
(예컨대,
Valley Health Physician)
파티션 VH-A 331
(Valley Health, VH)
테넌트 A 리소스들
(예컨대,
Bayland Urgent Care DB)
324
테넌트 B 리소스들
(예컨대, Valley Health DB)
334

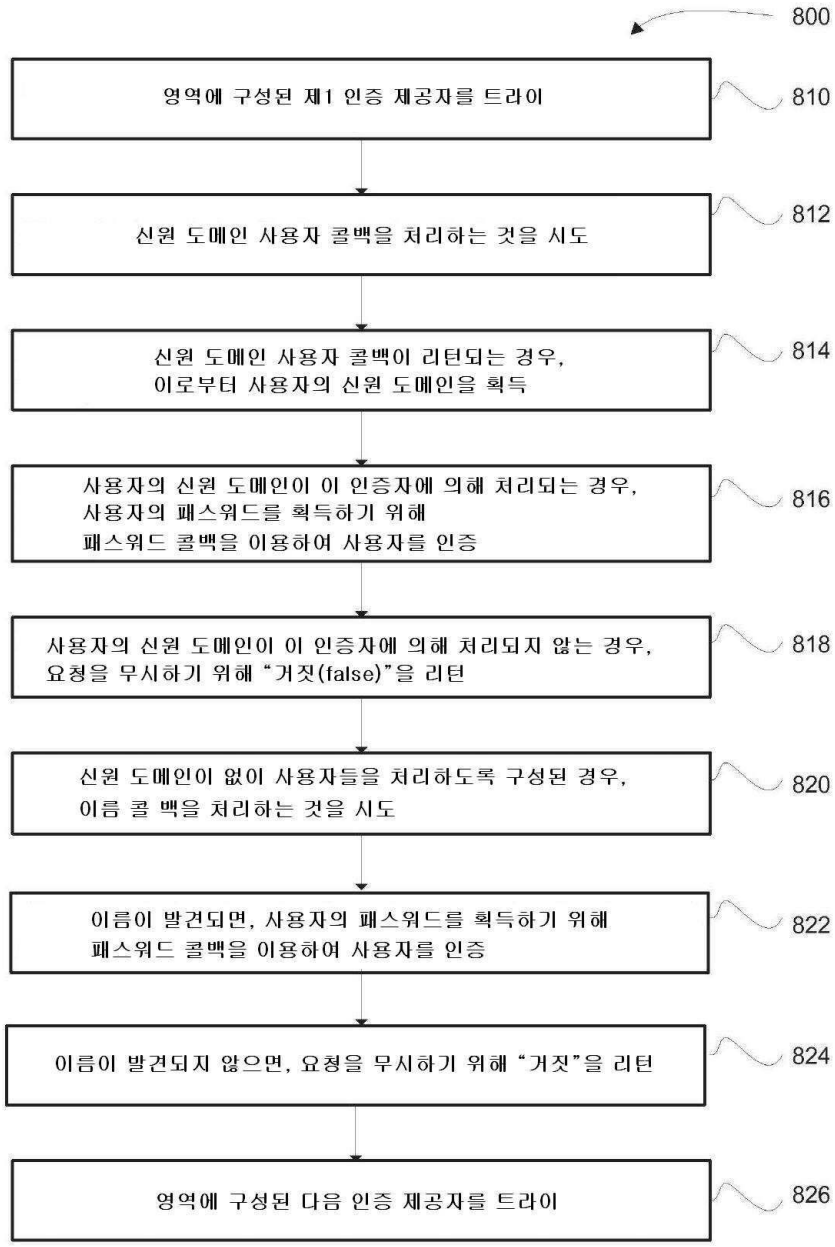
도면6


어드미니스트레이터
600
파티션 보안 설정들을 특정
어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
관리 서버/클러스터 242
파티션 B 260
파티션 보안
구성 B
660
파티션 A 250
파티션 보안
구성 A
650
파티션 B
리소스들
261
파티션 리소스들
및/또는 글로벌
리소스들에 대한
보안된 액세스를 제공
파티션 A
리소스들
251
가상 타겟 A 258
가상 타겟 B 268
글로벌 리소스들 140

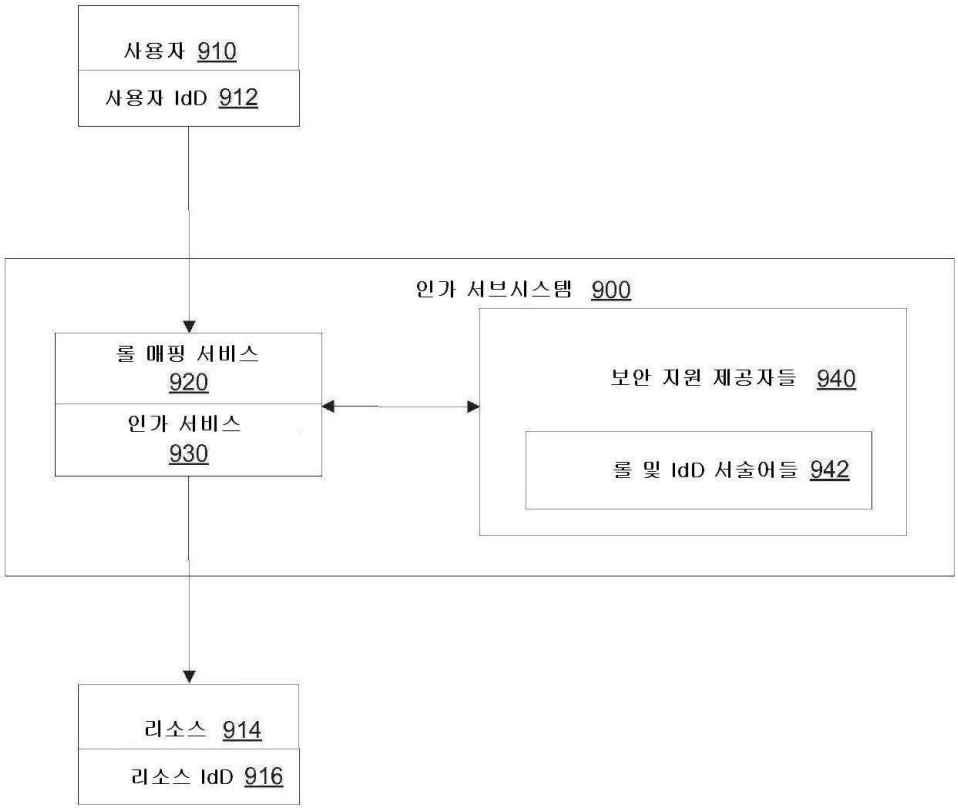
도면7


어드미니스트레이터
600
파티션 보안 설정들을 특정
어플리케이션 서버(에컨대, 멀티-테넌트, MT) 환경 100
관리 서버/클러스터 242
보안 구성 700
AIDD 704
영역 A 750
AUTH A 758
디폴트 영역
702
AUTH 708
영역 B 760
AUTH B 768
파티션 A 250
파티션
보안 구성 A
650
속성 752
PIDD A 754
LDAP A 756
파티션 A
리소스들
251
파티션 B 260
파티션
보안 구성 B
660
속성 762
PIDD B 764
LDAP B 766
파티션 B
리소스들
261
글로벌 리소스들 140
LDAP 706

도면8


800
영역에 구성된 제1 인증 제공자를 트라이
810
신원 도메인 사용자 콜백을 처리하는 것을 시도
812
신원 도메인 사용자 콜백이 리턴되는 경우,
이로부터 사용자의 신원 도메인을 획득
814
사용자의 신원 도메인이 이 인증자에 의해 처리되는 경우,
사용자의 패스워드를 획득하기 위해
패스워드 콜백을 이용하여 사용자를 인증
816
사용자의 신원 도메인이 이 인증자에 의해 처리되지 않는 경우,
요청을 무시하기 위해 "거짓(false)"을 리턴
818
신원 도메인이 없이 사용자들을 처리하도록 구성된 경우,
이름 콜 백을 처리하는 것을 시도
820
이름이 발견되면, 사용자의 패스워드를 획득하기 위해
패스워드 콜백을 이용하여 사용자를 인증
822
이름이 발견되지 않으면, 요청을 무시하기 위해 "거짓"을 리턴
824
영역에 구성된 다음 인증 제공자를 트라이
826

도면9


사용자 910
사용자 IdD 912
인가 서브시스템 900
롤 매핑 서비스 920
인가 서비스 930
보안 지원 제공자들 940
롤 및 IdD 서술어들 942
리소스 914
리소스 IdD 916