(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년09월30일
(11) 등록번호 10-2449753
(24) 등록일자 2022년09월27일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) *H04L 65/40* (2022.01)
(52) CPC특허분류
G06F 9/5077 (2013.01)
H04L 67/10 (2022.05)
(21) 출원번호 10-2017-7003960
(22) 출원일자(국제) 2015년07월14일
심사청구일자 2020년06월16일
(85) 번역문제출일자 2017년02월13일
(65) 공개번호 10-2017-0033350
(43) 공개일자 2017년03월24일
(86) 국제출원번호 PCT/US2015/040407
(87) 국제공개번호 WO 2016/011051
국제공개일자 2016년01월21일
(30) 우선권주장
62/024,364 2014년07월14일 미국(US)

(56) 선행기술조사문헌
Sun Microsystems, Inc.'JNDI API',[online],1999.07.14.,[2021.11.08.검색]*
Carla Sadtler 외.'WebSphere Application Server V6.1:System Management and Configuration'.ibm.com/redbooks.[online]2006.11.02.,[2021.11.08.검색]
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
오라클 인터내셔날 코포레이션
미국, 캘리포니아 94065, 레드우드 쇼어스 엠에스 5오피7, 오라클 파크웨이 500
(72) 발명자
장 레이멩
중국 베이징 100193 하이디안 디스트릭트 종관춘 소프트웨어 파크 빌딩 넘버 24
샨 꾸오준
중국 베이징 100193 하이디안 디스트릭트 종관춘 소프트웨어 파크 빌딩 넘버 24
(뒷면에 계속)
(74) 대리인
박장원

전체 청구항 수 : 총 22 항 심사관 : 이후락

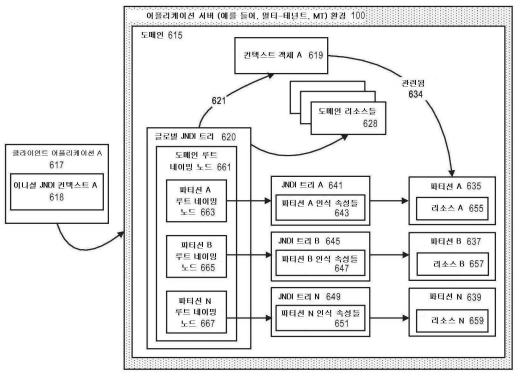
(54) 발명의 명칭 **멀티 테넌트 어플리케이션 서버 환경에서 네임스페이스들을 지원하기 위한 시스템 및 방법**

(57) 요 약

본 명세서에 설명된 것은 일 실시예에 따른, 멀티-테넌트 어플리케이션 서버 환경에서 네임스페이스들을 지원하기 위한 시스템 및 방법이다. 상기 어플리케이션 서버 환경은 복수의 파티션들을 갖는 도메인을 포함할 수 있다. 도메인-레벨 리소스들에 바인딩된 글로벌 네임스페이스들 또는 JNDI 트리는 각각의 파티션 루트 노드가 파티션

(뒷면에 계속)

대 표 도



JNDI 트리의 루트 노드인 파티션 루트 노드들의 컬렉션을 유지할 수 있다. 파티션 JNDI 트리의 각 노드는 특정 파티션에 특정한 속성들을 포함함으로써 파티션-인식형으로 만들어진다. 파티션에 대한 초기 컨텍스트는 리소스 룩업 요청들을 디스패칭하는데 사용하기 위해 생성될 수 있으며, 다른 어플리케이션들에 의해 상기 파티션 내의 리소스들에 액세스하는 데에 재사용 될 수 있다.

(52) CPC특허분류
H04L 67/141 (2022.05)
H04L 67/34 (2022.05)

(72) 발명자
샤넌 빌
미국 캘리포니아 94065 레드우드 쇼어스 엠/에스 5오피7 오라클 파크웨이 500
몰다니 라지브
미국 캘리포니아 94089 써니베일 린즈 테라스 1038
페이젠 로렌스
미국 뉴저지 07069 와청 레드몬트 로드 104
린 윤
중국 베이징 100193 하이디안 디스트릭트 종관춘 소프트웨어 파크 빌딩 넘버 24
첸 춘타오
중국 베이징 100193 하이디안 디스트릭트 종관춘 소프트웨어 파크 빌딩 넘버 24

명 세 서

청구범위

청구항 1

어플리케이션 서버 환경에서 다수의 자바(Java) 네이밍 및 디렉토리 인터페이스(JNDI) 트리를 지원하기 위한 시스템으로서,

하나 이상의 마이크로 프로세서를 각각 포함하는 하나 이상의 컴퓨터; 및

컴퓨터들 상의 어플리케이션 서버 환경을 포함하고,

상기 어플리케이션 서버 환경은,

도메인의 복수의 파티션과, 각 파티션은 각각의 리소스 그룹이 애플리케이션 또는 리소스를 정의하는 리소스 그룹 탬플릿에 대한 참조를 포함하는 하나 이상의 리소스 그룹을 포함하고, 그리고

각각의 파티션 루트 네이밍 노드가 파티션 JNDI 트리와 관련된 파티션 루트 네이밍 노드들의 컬렉션을 유지하는 글로벌 JNDI 트리를 포함하며,

상기 시스템은 특정 파티션 내의 애플리케이션 또는 리소스에 액세스하기 위한 요청을 어플리케이션으로부터 수신할 때,

특정 파티션과 관련된 초기 JNDI 컨텍스트를 생성하고, 그리고

JNDI 동작들을 애플리케이션으로부터 특정 파티션과 관련된 JNDI 트리에 위임하도록 동작하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 2

제1항에 있어서,

상기 시스템은 파티션 JNDI 트리를 식별하기 위해 요청으로부터의 정보 및 파티션 루트 네이밍 노드들의 컬렉션을 검사하도록 더 동작하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 3

제1항 또는 제2항에 있어서,

요청으로부터의 정보는 특정 파티션에 대한 정보를 지정하는 제공자 URL을 포함하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 4

제1항에 있어서,

파티션에 배치된 리소스들은 관련 파티션 JNDI 트리에 바인딩(binding)되고, 관련 파티션 JNDI 트리 내의 각 노드는 파티션 인식형(partition-aware)인 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 5

제1항에 있어서,

초기 JNDI 컨텍스트는 일단 생성되면 하나 이상의 다른 어플리케이션에 의한 재사용을 위해 컨텍스트 객체로서 유지되고, 애플리케이션으로부터의 JNDI 동작들은 컨텍스트 객체와 관련된 파티션으로 전달되는(directed) 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 6

제5항에 있어서,

컨텍스트 객체와 파티션 간의 관계는 컨텍스트 객체를 파티션의 네임 스페이스(name space)와 관련시킴으로써 유지되는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 7

제1항에 있어서,

JNDI 동작들이 파티션의 컴포넌트 호출(component invocation) 컨텍스트에서 처리될 수 있도록 하기 위해 복수의 콜백 기능이 JNDI 서브시스템에 제공되는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 시스템.

청구항 8

어플리케이션 서버 환경에서 다수의 자바 네이밍 및 디렉토리 인터페이스(JNDI) 트리를 지원하기 위한 방법으로서,

어플리케이션 서버 환경에서 도메인을 구성하는 단계와, 상기 도메인은 복수의 파티션과, 파티션 루트 네이밍 노드들의 컬렉션을 유지하는 글로벌 JNDI 트리를 포함하고, 각 파티션은 각각의 리소스 그룹이 애플리케이션 또는 리소스를 정의하는 리소스 그룹 탬플릿에 대한 참조를 포함하는 하나 이상의 리소스 그룹을 포함하고, 그리고 각 파티션 루트 네이밍 노드는 파티션에 특정한 파티션 JNDI 트리와 관련되며;

특정 파티션 내의 애플리케이션 또는 리소스에 액세스하기 위한 요청을 어플리케이션으로부터 수신하는 단계와;

특정 파티션과 관련된 초기 JNDI 컨텍스트를 생성하는 단계와; 그리고

JNDI 동작들을 애플리케이션으로부터 특정 파티션과 관련된 JNDI 트리에 위임하는 단계를 포함하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 9

제8항에 있어서,

상기 방법은 파티션 JNDI 트리를 식별하기 위해 요청으로부터의 정보 및 파티션 루트 네이밍 노드들의 컬렉션을 검사하는 단계를 더 포함하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 10

제9항에 있어서,

요청으로부터의 정보는 특정 파티션에 대한 정보를 지정하는 제공자 URL을 포함하는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 11

제8항에 있어서,

파티션에 배치된 리소스들은 관련 파티션 JNDI 트리에 바인딩되고, 관련 파티션 JNDI 트리 내의 각 노드는 파티션 인식형인 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 12

제8항에 있어서,

초기 JNDI 컨텍스트는 일단 생성되면 하나 이상의 다른 어플리케이션에 의한 재사용을 위해 컨텍스트 객체로서 유지되고, 애플리케이션으로부터의 JNDI 동작들은 컨텍스트 객체와 관련된 파티션으로 전달되는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 13

제12항에 있어서,

컨텍스트 객체와 파티션 간의 관계는 컨텍스트 객체를 파티션의 네임 스페이스와 관련시킴으로써 유지되는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 14

제8항에 있어서,

JNDI 동작들이 파티션의 컴포넌트 호출 컨텍스트에서 처리될 수 있도록 하기 위해 복수의 콜백 기능이 JNDI 서브시스템에 제공되는 것을 특징으로 하는 JNDI 트리를 지원하기 위한 방법.

청구항 15

컴퓨터 시스템에 의해 실행될 때 컴퓨터 시스템으로 하여금 제8항 내지 제14항 중 어느 한 항에 따른 방법을 수행하게 하는 기계-판독가능 포멧의 명령어들을 포함하는 비-일시적 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 16

제15항의 컴퓨터 프로그램을 저장하는 비-일시적 기계 판독가능 데이터 저장 매체.

청구항 17

하나 이상의 컴퓨터에 의해 판독되고 실행될 때 하나 이상의 컴퓨터로 하여금 단계들을 수행하게 하는 저장된 명령어들을 포함하는 비-일시적 컴퓨터 판독가능 저장 매체로서, 상기 단계들은,

애플리케이션 서버 환경에서 도메인을 구성하는 단계와, 상기 도메인은 복수의 파티션, 및 파티션 루트 네이밍 노드들의 컬렉션을 유지하는 글로벌 JNDI 트리를 포함하고, 각 파티션은 각각의 리소스 그룹이 애플리케이션 또는 리소스를 정의하는 리소스 그룹 탬플릿에 대한 참조를 포함하는 하나 이상의 리소스 그룹을 포함하고, 그리고 각 파티션 루트 네이밍 노드는 파티션에 특정한 파티션 JNDI 트리와 관련되며;

특정 파티션 내의 애플리케이션 또는 리소스에 액세스하기 위한 요청을 어플리케이션으로부터 수신하는 단계와;

특정 파티션과 관련된 초기 JNDI 컨텍스트를 생성하는 단계와; 그리고

JNDI 동작들을 애플리케이션으로부터 특정 파티션과 관련된 JNDI 트리에 위임하는 단계를 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 18

제17항에 있어서,

상기 단계들은 파티션 JNDI 트리를 식별하기 위해 요청으로부터의 정보 및 파티션 루트 네이밍 노드들의 컬렉션을 검사하는 단계를 더 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 19

제17항 또는 제18항에 있어서,

요청으로부터의 정보는 특정 파티션에 대한 정보를 지정하는 제공자 URL을 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 20

제17항에 있어서,

파티션에 배치된 리소스들은 관련 파티션 JNDI 트리에 바인딩되고, 관련 파티션 JNDI 트리 내의 각 노드는 파티션 인식형인 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 21

제17항에 있어서,

초기 JNDI 컨텍스트는 일단 생성되면 하나 이상의 다른 어플리케이션에 의한 재사용을 위해 컨텍스트 객체로서 유지되고, 애플리케이션으로부터의 JNDI 동작들은 컨텍스트 객체와 관련된 파티션으로 전달되는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 22

제21항에 있어서,

컨텍스트 객체와 파티션 간의 관계는 컨텍스트 객체를 파티션의 네임 스페이스와 관련시킴으로써 유지되는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 저장 매체.

발명의 설명

기 술 분 야

[0001] [저작권 공지]

[0002] 본 특허 문서에 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 않으나, 그 밖의 모든 것은 저작권으로 보호된다.

[0003] [기술분야]

[0004] 본 발명의 실시예들은 일반적으로 어플리케이션 서버들 및 클라우드 플랫폼 환경들과 관련되며, 멀티 테넌트(multitenant) 어플리케이션 서버 환경에서 네임 스페이스들을 지원하기 위한 시스템 및 방법과 특히 관련된다.

배 경 기 술

[0005] 소프트웨어 어플리케이션 서버들은 - 예시로서 Oracle WebLogic Server(WLS) 및 Glassfish를 포함한다 - 일반적으로 엔터프라이즈 소프트웨어 어플리케이션들을 실행하기 위해 관리되는 환경을 제공한다. 최근, 클라우드 환경 내에서 사용자들 또는 테넌트들이 그들의 어플리케이션들을 개발 및 실행하고, 그 환경에 의해 제공된 분산 리소스들(distributed resources)를 활용할 수 있게 하는 클라우드 환경에서 사용하기 위한 기술 또한 개발되었다.

발명의 내용

[0006] 본 명세서에 설명된 것은 일 실시예에 따른, 멀티 테넌트 어플리케이션 서버 환경에서 네임 스페이스들을 지원하기 위한 시스템 및 방법이다. 이 어플리케이션 서버 환경은 복수의 파티션들(partitions)을 갖는 도메인을 포함할 수 있다. 도메인-레벨 리소스들에 바인딩(binding)된 글로벌 네임스페이스(namespace) 또는 JNDI 트리는, 각각의 파티션 루트 노드(partition root node)가 파티션 JNDI 트리의 루트 노드인 파티션 루트 노드들의 컬렉션(collection of partition root nodes)을 유지할 수 있다. 파티션 JNDI 트리의 각 노드는 특정 파티션에 특정한(specific) 속성을 포함함으로써 파티션-인식(partition-aware)형으로 만들어진다. 파티션에 대한 이니셜 컨텍스트(initial context)는 리소스 룩업(lookup) 요청들을 상기 파티션에 디스패칭(dispatching)하는 데 사용하기 위해 생성될 수 있고, 다른 어플리케이션들에 의해 상기 파티션의 리소스들에 액세스 하는 데에 재사용될 수 있다.

도면의 간단한 설명

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

도 2는 일 실시예에 따른, 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하기 위한 시스템을 더 도시한다.

도 3은 일 실시예에 따른, 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하기 위한 시스템을 더 도시한다.

도 4는 일 실시예에 따른, 멀티-테넌트 환경과 함께 사용하기 위한 대표적인 도메인 구성을 도시한다.

도 5는 일 실시예에 따른, 대표적 멀티-테넌트 환경을 더 도시한다.

도 6은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서, 도메인 내의 다중 파티션 JNDI 트리들을 지원하기 위한 시스템을 도시한다.

도 7은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서, 도메인 내의 다중 파티션 JNDI 트리들을 지원하기 위한 시스템을 더 도시한다.

도 8은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서, 도메인 내의 다중 파티션 JNDI 트리들을 지원하기 위한 방법을 도시한다.

도 9는 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서, 도메인 내의 다중 파티션 JNDI를 지원하기 위한 대표적인 시스템을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 본 명세서에 설명된 것은 일 실시예에 따른, 멀티 테넌트 어플리케이션 서버 환경에서 네임 스페이스들을 지원하기 위한 시스템 및 방법이다. 이 어플리케이션 서버 환경은 복수의 파티션들을 갖는 도메인을 포함할 수 있다. 도메인-레벨 리소스들에 바인딩된 글로벌 네임스페이스 또는 JNDI 트리는, 각각의 파티션 루트 노드가 파티션 JNDI 트리의 루트 노드인 파티션 루트 노드들의 컬렉션을 유지할 수 있다. 파티션 JNDI 트리의 각 노드는 특정 파티션에 특정한 속성들을 포함함으로써 파티션-인식형으로 만들어진다. 파티션에 대한 이니셜 컨텍스트는 리소스 룩업 요청들을 상기 파티션에 디스패칭하는 데 사용하기 위해 생성될 수 있고, 다른 어플리케이션들에 의해 상기 파티션의 리소스들에 액세스 하는 데에 재사용될 수 있다.

[0009] **어플리케이션 서버(예를 들어, 멀티-테넌트, MT) 환경**

[0010] 도 1은 일 실시예에 따른, 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하기 위한 시스템을 도시한다.

[0011] 도 1에 도시된 바와 같이, 일 실시예에 따르면, 소프트웨어 어플리케이션들의 배포 및 실행을 가능하게 하는 어플리케이션 서버(예를 들어, 멀티-테넌트, MT) 환경(100) 또는 다른 컴퓨팅 환경은, 런타임에서 어플리케이션 서버 도메인을 정의하는 데에 사용되는 도메인(102) 구성에 따라 포함 및 작동하도록 구성될 수 있다.

[0012] 일 실시예에 따르면, 어플리케이션 서버는 런타임에서 사용하기 위해 정의되는 하나 이상의 파티션들(104)을 포함할 수 있다. 각 파티션은 파티션 식별자(identifier, ID) 및 파티션 구성과 관련될 수 있고, 리소스 그룹 템플릿들(resource group templates, 126) 및/또는 파티션-특정(partition-specific) 어플리케이션들 또는 리소스들(128)에 대한 참조와 함께 하나 이상의 리소스 그룹들(resource group, 124)을 더 포함할 수 있다. 도메인-레벨 리소스 그룹들, 어플리케이션들 및/또는 리소스들(140) 또한 도메인 레벨에서 정의될 수 있으며, 선택적으로(optionally) 리소스 그룹 템플릿에 대한 참조와 함께 정의된다.

[0013] 각 리소스 그룹 템플릿(160)은 하나 이상의 어플리케이션들 A(162), B(164), 리소스들 A(166), B(168), 및/또는 다른 배포 가능한(deployable) 어플리케이션들 또는 리소스들(170)을 정의할 수 있으며, 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.

[0014] 일반적으로, 시스템 관리자는 파티션들, 도메인-레벨 리소스 그룹들 및 리소스 그룹 템플릿들, 그리고 보안 렐름들(realms)을 정의할 수 있다; 반면, 파티션 관리자는 예를 들어, 파티션-레벨 리소스 그룹들을 생성하거나, 파티션에 어플리케이션들을 배포하거나, 또는 파티션에 대한 특정 렐름을 참조함으로써 각자의 파티션의 양태(aspect)들을 정의할 수 있다.

[0015] 도 2는 일 실시예에 따른, 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하기 위한 시스템을 더 도시한다.

[0016] 일 실시예에 따르면, 도 2에 도시된 바와 같이, 파티션(202)은 예를 들어, 리소스 그룹 템플릿(210)에 대한 참조(206)를 포함하는 리소스 그룹(205), 가상 타겟(예를 들어, 가상 호스트) 정보(207), 그리고 플러거블(pluggable) 데이터베이스(PDB) 정보(208)를 포함할 수 있다. 예를 들어, 리소스 그룹 템플릿(예를 들어, 210)은, 예컨대 Java Message Server(JMS) 서버(213), store-and-forward(SAF) 에이전트(215), 메일 세션 컴포넌트(216), 또는 Java Database Connectivity(JDBC) 리소스(217)와 같은 리소스들과 함께 복수의 어플리케이션들 A(211), B(212)를 정의할 수 있다.

[0017] 도 2에 도시된 리소스 그룹 템플릿은 예시로서 제공된다; 다른 실시예들에 따라서, 다양한 유형들의 리소스 그룹 템플릿들 및 엘리먼트(element)들이 제공될 수 있다.

[0018] 일 실시예에 따르면, 파티션(예를 들어, 202)이 특정 리소스 그룹 템플릿(예를 들어, 210)을 참조(220)할 때, 특정 파티션과 관련된 정보와 참조된 리소스 그룹 템플릿이 결합하여 사용되어, 예를 들어 파티션-특정 PDB 정보와 같은 파티션-특정 정보(230)를 표시할 수 있다. 그 다음, 이 파티션-특정 정보는 어플리케이션 서버에 의해 사용되어 상기 파티션에 의해 사용되는 리소스들 예를 들어, PDB 리소스를 구성할 수 있다. 예를 들어, 파티션(202)과 관련된 파티션-특정 PDB 정보는 어플리케이션 서버에 의해 사용되어, 해당 파티션에 의해 사용되기 위해 적절한 PDB와 함께 컨테이너 데이터베이스(CDB)(236)를 구성(232)할 수 있다.

[0019] 유사하게, 일 실시예에 따르면, 특정 파티션과 관련된 가상 타겟 정보(virtual target information)는 그 파티션(예를 들어, baylandurgent.com)이 사용하기 위한 파티션-특정 가상 타겟(240)을 정의(239)하는 데에 사용될 수 있으며, 그 후 이 파티션-특정 가상 타겟은 URL(Uniform Resource Locator)을 통해 액세스 가능하게 될 수 있다(예를 들어, http://baylandurgentcare.com).

[0020] 도 3은 일 실시예에 따른, 어플리케이션 서버, 클라우드 또는 다른 환경에서 멀티-테넌시를 지원하기 위한 시스템을 더 도시한다.

[0021] 일 실시예에 따르면, 예컨대 config.xml 구성 파일과 같은 시스템 구성은 한 파티션- 이 파티션은 자신과 관련된 리소스 그룹들에 대한 구성 엘리먼트들을 포함한다 - 및/또는 기타 파티션 속성들을 정의하는데 사용된다. 값들은 속성 이름/값 페어(pair)들을 사용하여 파티션별로 지정될 수 있다.

[0022] 일 실시예에 따르면, 복수의 파티션들은, CDB(243)에 대한 액세스를 제공할 수 있고 웹 티어(web tier)(244)를 통해 액세스 가능한 관리되는 서버/클러스터(242) 또는 유사한 환경 내에서 실행될 수 있다. 이는 예를 들어, 도메인이 CDB와 관련되고, 각각의 파티션이 (CDB의) 하나 이상의 PDB들과 암시적으로 관련될 수 있게 한다.

[0023] 일 실시예에 따르면, 복수의 파티션들 중 각각은 - 이 예시에서는 파티션 A(250) 및 파티션 B(260) - 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), 그리고 JMS A(256)를 포함하는 리소스 그룹(251)을 포함하도록 구성될 수 있으며, 이 파티션은 가상 타겟 A(258)을 통해 액세스 가능하다. 유사하게, 파티션 B(260)는 PDB B(269)와 관련된 데이터소스 B(267)와 함께, 어플리케이션 B1(262), 어플리케이션 B2(264), 그리고 JMS B(266)를 포함하는 리소스 그룹(261)을 포함하도록 구성될 수 있으며, 이 파티션은 가상 타겟 B(268)를 통해 액세스 가능하다.

[0024] 상기 몇 가지 예시들이 CDB 및 PDB들의 사용을 도시하고 있는 반면, 다른 실시예들에 따라서는 다른 유형의 멀티-테넌트 또는 비-멀티-테넌트 데이터베이스들이 지원될 수 있으며, 예를 들어 스키마들의 사용 또는 다양한 데이터베이스들의 사용을 통해 각각의 파티션들에 대한 특정 구성이 제공될 수 있다.

[0025] **리소스들**

[0026] 일 실시예에 따르면, 리소스는 환경의 도메인에 배포될 수 있는 시스템 리소스, 어플리케이션 또는 다른 리소스 또는 객체이다. 예를 들어, 일 실시예에 따르면, 리소스는 어플리케이션, JMS, JDBC, JavaMail, WLDF, 데이터소스이거나, 또는 서버, 클러스터, 또는 다른 어플리케이션 서버 타겟에 배포될 수 있는 다른 시스템 리소스 또는 다른 유형의 객체일 수 있다.

[0027] **파티션들**

[0028] 일 실시예에 따르면, 파티션은 파티션 식별자(ID) 및 구성과 관련될 수 있는 도메인의 런타임 및 관리적 서브디비전(administrative subdivision) 또는 슬라이스이며, 어플리케이션들을 포함하고 그리고/또는 리소스 그룹들 및 리소스 그룹 템플릿들의 사용을 통해서 도메인-범위 리소스들(domain-wide resources)을 참조할 수 있다.

[0029] 일반적으로, 파티션은 자체 어플리케이션들을 포함하고, 리소스 그룹 템플릿들을 통해 도메인 범위 어플리케이션들을 참조하며, 자체의 구성을 가질 수 있다. 파티션 가능한 엔티티(partitionable entity)들은 예를 들어 JMS, JDBC, JavaMail, WLDF 리소스들, 그리고 다른 구성요소들(예컨대 JNDI 네임스페이스, 네트워크 트래픽, 작업 관리자들, 그리고 보안 정책들 및 렐름들)과 같은 리소스들을 포함할 수 있다. 멀티-테넌트 환경의 컨텍스트에서, 시스템은 테넌트와 관련된 파티션들의 관리적 및 런타임 양태들에의 테넌트 액세스를 제공하도록 구성될 수 있다.

[0030] 일 실시예에 따르면, 파티션 내의 각 리소스 그룹은 하나 이상의 리소스 그룹 템플릿들을 참조할 수 있다. 각 파티션은 해당 파티션의 리소스 그룹들이 참조하는 리소스 그룹 템플릿들에 특정되지 않은 구성 데이터에 대한 속성들을 정의할 수 있다. 이는 파티션이 리소스 그룹 템플릿에 정의된 배포 가능한 리소스들과, 그 파티션과

사용되기 위한 특정 값들의 바인딩으로서의 역할을 할 수 있도록 한다. 경우에 따라서는, 파티션이 리소스 그룹 템플릿에 의해 특정된 구성 정보를 오버라이드(override) 할 수 있다.

[0031] 일 실시예에 따르면, 예를 들어 config.xml 파티션 구성 파일에 의해 정의된 것과 같은 파티션 구성은 복수의 구성 엘리먼트들, 예를 들면: 파티션을 정의하는 속성들(attributes) 및 자식 엘리먼트들을 포함하는 "파티션"; 상기 파티션에 배포된 어플리케이션들 및 리소스들을 포함하는 "리소스-그룹"; 템플릿에 의해 정의되는 어플리케이션들 및 리소스들을 포함하는 "리소스-그룹-템플릿"; 데이터베이스-특정 서비스 이름, 사용자 이름, 및 패스워드를 포함하는 "jdbc-시스템-리소스-오버라이드"; 그리고 리소스 그룹 템플릿들에서 매크로 교체(macro replacement)에 사용될 수 있는 속성 키 값(property key values)들을 포함하는 "파티션-속성들"을 포함할 수 있다.

[0032] 스타트업(startup) 시, 시스템은 구성 파일에 의해 제공된 정보를 사용하여 리소스 그룹 템플릿으로부터 각 리소스에 대한 파티션-특정 구성 엘리먼트들을 생성할 수 있다.

[0033] **리소스 그룹들**

[0034] 일 실시예에 따르면, 리소스 그룹은 도메인이나 파티션 레벨에서 정의될 수 있는 배포 가능한 리소스들의 명명된, 완전히-자격을 갖춘 컬렉션(fully-qualified collection)이며, 리소스 그룹은 리소스 그룹 템플릿을 참조할 수 있다. 리소스 그룹 내의 리소스들은 관리자가 그 리소스들을 시작하거나 또는 그 리소스들에 연결하기 위해 필요한 모든 정보 - 예를 들어, 데이터 소스에 연결하기 위한 자격(credentials)들 또는 어플리케이션에 대한 타겟팅 정보- 를 제공한다는 점에서 완전히-자격을 갖춘 것으로 고려된다.

[0035] 시스템 관리자는 도메인 레벨 또는 파티션 레벨에서 리소스 그룹들을 선언할 수 있다. 도메인 레벨에서, 리소스 그룹은 관련된 리소스들을 그룹핑(grouping)하기 위한 편리한 방식을 제공한다. 시스템은 도메인-레벨 리소스 그룹에서 선언된 리소스들을 그룹핑되지 않은 리소스들과 동일하게 관리하여, 이 리소스들이 시스템 스타트-업 동안에 시작되고 시스템 셧-다운 동안에 중지될 수 있게 한다. 관리자 또한 그룹 내의 리소스를 개별적으로 중지, 시작, 또는 제거할 수 있으며, 그룹 상에서 동작함으로써 그 그룹 내의 모든 리소스들에 암시적으로 영향을 줄 수 있다(act on). 예를 들어, 리소스 그룹을 중지하면 그 그룹 내의 아직 중지되지 않은 모든 리소스들이 중지되고, 리소스 그룹을 시작하면 그 그룹 내의 아직 시작되지 않은 모든 리소스들이 시작되고, 그리고 리소스 그룹을 제거하면 그 그룹에 포함된 모든 리소스들이 제거된다.

[0036] 파티션 레벨에서, 시스템 또는 파티션 관리자는 임의의 보안 제약들의 대상이 되는, 파티션 내의 하나 이상의 리소스 그룹들을 구성할 수 있다. 예를 들어, SaaS 유스 케이스(use case)에서는, 다양한 파티션-레벨 리소스 그룹들이 도메인-레벨 리소스 그룹 템플릿들을 참조할 수 있는 반면, PaaS 유스 케이스에서는 파티션-레벨 리소스 그룹들이 생성될 수 있는바, 이 파티션-레벨 리소스 그룹들은 리소스 그룹 템플릿을 참조하지 않고, 대신에 그 파티션 내에서만 이용 가능한 어플리케이션들과 그들의 관련 리소스들을 나타낸다.

[0037] 일 실시예에 따르면, 어플리케이션들과 그들이 도메인 내에서 별개의 관리적 유닛으로서 사용하는 리소스들을 함께 그룹핑 하기 위해 리소스 그룹핑이 사용될 수 있다. 예를 들어, 아래에 설명된 의료 기록(MedRec) 어플리케이션에서, 리소스 그룹핑은 MedRec 어플리케이션과 이 어플리케이션의 리소스들을 정의한다. 다수의 파티션들이 동일한 MedRec 리소스 그룹을 실행할 수 있고, 각각의 파티션은 파티션-특정 구성 정보를 사용하여 각 MedRec 인스턴스(instance)의 일부인 어플리케이션들이 각각의 파티션에 특정되도록 만들어진다.

[0038] **리소스 그룹 템플릿들**

[0039] 일 실시예에 따르면, 리소스 그룹 템플릿은 도메인 레벨에서 정의되는 배포 가능한 리소스들의 컬렉션이고, 리소스 그룹으로부터 참조될 수 있으며, 리소스 그룹 템플릿의 리소스들을 활성화하는데 요구되는 정보 중 일부는 템플릿이 파티션 레벨 구성의 세부사양(specification)을 지원하도록 템플릿 자체의 일부로서 저장되지 않을 수 있다. 도메인은 임의의 수의 리소스 그룹 템플릿들을 포함할 수 있으며, 각 리소스 그룹 템플릿은 예를 들어, 하나 이상의 관련 Java 어플리케이션들과 이들 어플리케이션들이 의존하는 리소스들을 포함할 수 있다. 이러한 리소스들에 관한 일부 정보는 모든 파티션들에 걸쳐 동일할 수 있으나, 다른 정보는 파티션마다 다를 수 있다. 모든 구성이 도메인 레벨에서 특정될 필요는 없다 - 대신 파티션 레벨 구성이 매크로들 또는 속성 이름/값 페어들의 사용을 통해 리소스 그룹 템플릿에 지정될 수 있다.

[0040] 일 실시예에 따르면, 특정 리소스 그룹 템플릿은 하나 이상의 리소스 그룹들에 의해 참조될 수 있다. 리소스 그룹을 포함하는 객체(예를 들어, 도메인 또는 파티션)는 속성 이름/값 할당(assignments)들을 사용하여 리소스 그룹 템플릿 내의 임의의 토큰들의 값을 설정할 수 있다. 참조하는 리소스 그룹을 사용하여 시스템이 리소스 그

룹 템플릿을 활성화할 때, 시스템은 리소스 그룹의 포함하는 객체에 설정된 값들로 이들 토큰들을 대체할 수 있다. 일부 경우들에서, 시스템은 또한 정적으로 구성된 리소스 그룹 템플릿들 및 파티션들을 사용하여 각각의 파티션/템플릿 조합에 대한 런타임 구성을 생성할 수 있다.

[0041] 예를 들어, SaaS 유스 케이스에서 시스템은 동일한 어플리케이션들 및 리소스들을 여러 번 - 이들 어플리케이션들 및 리소스들을 사용할 파티션 하나당 한 번씩을 포함 - 활성화할 수 있다. 관리자가 리소스 그룹 템플릿을 정의할 때, 이들은 토큰들을 사용하여 다른 곳에 제공될 정보를 나타낼 수 있다. 예를 들어, CRM-관련 데이터 리소스에 연결하는데 사용할 사용자이름(username)은 리소스 그룹 템플릿에서 /${CRMDataUsername}으로서 표시될 수 있다.

[0042] **테넌트들**

[0043] 일 실시예에 따르면, 예컨대 멀티-테넌트(MT) 어플리케이션 서버 환경과 같은 멀티-테넌트 환경에서, 테넌트는 하나 이상의 파티션들 및/또는 하나 이상의 테넌트-인식 어플리케이션들에 의해 표현될 수 있는, 또는 그들과 관련될 수 있는 엔티티이다.

[0044] 예를 들어, 테넌트들은 예컨대 서로 다른 외부 회사들 또는 특정 기업 내 서로 다른 부서들(예를 들어, 인사 부 또는 재무 부서들)과 같은 별개의 사용자 조직들을 을 나타낼 수 있으며, 각각의 테넌트는 서로 다른 파티션과 관련될 수 있다. 테넌트 아이덴티티(테넌트 ID)는 시간상 특정 순간에서 특정 사용자와 특정 테넌트의 관계(association)이다. 시스템은 특정 사용자가 어느 테넌트에 속하는지 유저 아이덴티티로부터 - 예를 들어, 유저 아이덴티티 스토어를 참조함으로써- 도출할 수 있다. 유저 아이덴티티는 시스템으로 하여금 사용자가 수행할 수 있도록 권한이 부여되는 액션들 - 사용자가 어느 테넌트에 속하는가를 포함하되 이에 국한되지 않음 - 을 시행(enforce)할 수 있게 한다.

[0045] 일 실시예에 따르면, 시스템은 서로 다른 테넌트들의 관리와 런타임의 격리를 가능하게 한다. 예를 들어, 테넌트들은 그들의 어플리케이션들의 일부 동작들(behaviors) 및 그들이 액세스 권한을 갖는 리소스들을 구성할 수 있다. 시스템은 특정 테넌트가 다른 테넌트에 속하는 아티팩트(artifacts)들을 관리할 수 없도록, 그리고 특정 테넌트를 대신하여 작업하는 어플리케이션들이 런타임에서 그 테넌트와 관련된 리소스들만을 참조하고 그 외 테넌트들과 관련된 리소스들은 참조하지 않도록 할 수 있다.

[0046] 일 실시예에 따르면, 테넌트-미인식 어플리케이션은 어플리케이션이 사용하는 임의의 리소스들이 그 어플리케이션이 응답하는 요청을 어떤 사용자가 제출하였는지에 관계없이 액세스 가능할 수 있도록, 명시적으로 테넌트들을 다루는 로직을 포함하지 않는 것이다. 반면에 테넌트-인식 어플리케이션은 테넌트들을 명시적으로 다루는 로직을 포함한다. 예를 들어, 어플리케이션은 사용자의 아이덴티티에 기초하여, 사용자가 속하는 테넌트를 도출하고 그 정보를 사용하여 테넌트-특정 리소스들에 액세스할 수 있다.

[0047] 일 실시예에 따르면, 시스템은 사용자들로 하여금 테넌트-인식형으로 명시적으로 작성된 어플리케이션들을 배포할 수 있도록 하여, 어플리케이션 개발자들이 현재 테넌트의 테넌트 ID를 획득할 수 있게 한다. 그 다음, 테넌트-인식 어플리케이션은 이 테넌트 ID를 사용하여 해당 어플리케이션의 단일 인스턴스를 사용하는 다수의 테넌트들을 핸들링(handling)할 수 있다.

[0048] 예를 들어, 단일 의사의 오피스 또는 병원을 지원하는 MedRec 어플리케이션은 예를 들어, Bayland Urgent Care 테넌트 및 Valley Health 테넌트와 같은 두 개의 서로 다른 파티션들 또는 테넌트들에 노출될 수 있으며, 이들 각각은 근본적인(underlying) 어플리케이션 코드를 변경하지 않고 별개의 테넌트-특정 리소스들(예컨대 별개의 PDB들)에 액세스할 수 있다.

[0049] **대표적인 도메인 구성 및 멀티-테넌트 환경**

[0050] 일 실시예에 따르면, 어플리케이션들은 도메인 레벨의 리소스 그룹 템플릿에 배포될 수 있거나, 파티션 또는 도메인에 스코핑(scoping)된 리소스 그룹에 배포될 수 있다. 어플리케이션 구성은 어플리케이션 별로 또는 파티션 별로 특정되는 배포 플랜(deployment plan)들을 사용하여 오버라이드될 수 있다. 또한, 배포 플랜들은 리소스 그룹의 일부로서 특정될 수 있다.

[0051] 도 4는 일 실시예에 따른, 멀티-테넌트 환경과 함께 사용하기 위한 대표적인 도메인 구성을 도시한다.

[0052] 일 실시예에 따르면, 시스템이 파티션을 시작할 때, 시스템은 제공된 구성에 따라 각각의 데이터베이스 인스턴스들에 대해 가상 타겟들(예를 들어, 가상 호스트들) 및 연결 풀(pool)들 - 각 파티션별 하나씩을 포함함- 을

생성한다.

[0053] 일반적으로, 각 리소스 그룹 템플릿은 하나 이상의 관련된 어플리케이션들 및 이들 어플리케이션들이 의존하는 리소스들을 포함할 수 있다. 리소스 그룹 템플릿들 내의 배포 가능한 리소스들을 파티션과 관련된 특정 값에 바인딩함을 제공함으로써 - 경우에 따라, 리소스 그룹 템플릿에 의해 특정된 어떤 구성 정보를 오버라이드하는 것을 포함함 - 각 파티션은 그 파티션이 참조하는 리소스 그룹 템플릿에 특정되지 않은 구성 데이터를 제공할 수 있다. 이는 시스템으로 하여금 각 파티션이 정의한 속성 값들을 사용하여 리소스 그룹 템플릿에 의해 각 파티션마다 다르게 표현되는 어플리케이션을 활성화할 수 있게 한다.

[0054] 일부 인스턴스들에서, 파티션은 리소스 그룹 템플릿들을 참조하지 않는, 또는 리소스 그룹들 자체의 파티션-스코핑된 배포 가능한 리소스들을 직접적으로 정의하는 리소스 그룹들을 포함할 수 있다. 파티션 내에 정의된 어플리케이션들 및 데이터 소스들은 일반적으로 그 파티션에서만 사용 가능하다.

[0055] 예를 들어, MedRec 어플리케이션은 복수의 Java 어플리케이션들, 데이터 소스, JMS 서버 및 메일 세션을 포함할 수 있다. 다수의 테넌트들을 위한 MedRec 어플리케이션을 실행하기 위해, 시스템 관리자는 템플릿 내의 상기 배포 가능한 리소스들을 선언하는 단일 MedRec 리소스 그룹 템플릿(286)을 정의할 수 있다.

[0056] 도메인-레벨 배포 가능한 리소스들과는 반대로, 리소스 그룹 템플릿에 선언된 배포 가능한 리소스들은 템플릿에 완전히 구성되지 않거나, 또는 일부 구성 정보가 없기 때문에 그대로 활성화될 수 없다.

[0057] 예를 들어, MedRec 리소스 그룹 템플릿은 어플리케이션들에 의해 사용되는 데이터 소스를 선언할 수 있으나, 데이터베이스에 연결하기 위한 URL을 지정하지 않을 수 있다. 예를 들어, 파티션 BUC-A(290)(Bayland Urgent Care, BUC) 및 파티션 VH-A(Valley Health, VH)와 같은, 서로 다른 테넌트들에 관련된 파티션들은 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, 이 참조는 Bayland Urgent Care 테넌트에 의해 사용되기 위한 BUC-A 파티션과 관련된 가상 호스트 baylandurgentcare.com(304)와, Valley Health 테넌트에 의해 사용되기 위한 VH-A 파티션과 관련된 가상 호스트 valleyhealth.com(308)을 포함하는 각 테넌트에 대한 가상 타겟들/가상 호스트들을 생성(302, 306)하는데 사용될 수 있다.

[0058] 도 5는 일 실시예에 따른 대표적인 멀티-테넌트 환경을 더 도시한다. 도 5에 도시된 바와 같이, 그리고 두 파티션들이 MedRec 리소스 그룹 템플릿을 참조하는 전술된 예시를 계속하면, 일 실시예에 따라 복수의 테넌트 환경들 - 이 예시에서는 Bayland Urgent Care Physician 테넌트 환경(320) 및 Valley Health Physician tenant 환경(330) - 을 지원하기 위해 서블렛(servlet) 엔진(310)이 사용될 수 있다.

[0059] 일 실시예에 따르면, 각 파티션(321, 331)은 그 테넌트 환경에 대한 착신 트래픽을 수용할 서로 다른 가상 타겟과, 파티션과 파티션의 리소스들(324, 334) - 이 예시에서는 bayland urgent care 데이터 베이스 또는 valley health 데이터베이스를 포함함 - 에 연결하기 위한 서로 다른 URL(322, 332)을 정의할 수 있다. 데이터베이스 인스턴스들은 동일한 어플리케이션 코드가 두 데이터베이스들 모두에 대해 실행할 것이므로 호환 가능한 스키마(compatible schema)들을 사용할 수 있다. 시스템이 파티션들을 시작할 때, 시스템은 각각의 데이터베이스 인스턴스들에 대한 가상 타겟들 및 연결 풀들을 생성할 수 있다.

[0060] **다수의 파티션 JNDI 트리에 대한 지원**

[0061] 어플리케이션 서버 환경 예를 들어, 멀티테넌트, 클라우드 또는 다른 환경들은 다수의 파티션들을 포함할 수 있다. 예를 들어, 데이터 소스, JMS 또는 메일 세션과 같은 리소스는 하나 이상의 파티션에 배포될 수 있다.

[0062] 일반적으로, 리소스는 어플리케이션 서버의 네이밍 서비스를 통해 리소스 이름에 맵핑(mapping)되는 하나 이상의 객체들을 포함할 수 있어, 어플리케이션이 이러한 객체들에 액세스해야 할 때, 어플리케이션은 리소스의 이름별로 객체들을 위치할 수 있다.

[0063] 일 실시예에 따르면, 이상적으로 네이밍 서비스는 파티션-인식형이며, 특정 파티션에 리소스 룩업 요청을 디스패칭 하기 위해 네이밍 서비스 내에서 파티션의 격리를 제공한다.

[0064] 본 명세서에 설명된 것은 일 실시예에 따른 멀티테넌트 어플리케이션 서버 환경에서 네임스페이스들을 지원하기 위한 시스템 및 방법이다. 어플리케이션 서버 환경은 복수의 파티션들을 갖는 도메인을 포함할 수 있다. 도메인-레벨 리소스들에 바인딩된 글로벌 네이밍스페이스 또는 JNDI 트리는 각각의 파티션 루트 노드가 파티션 JNDI 트리의 루트 노드인 파티션 루트 노드들의 컬렉션을 유지할 수 있다. 파티션 JNDI 트리의 각 노드는 특정 파티션에 특정한 속성들을 포함함으로써 파티션-인식형으로 만들어진다. 파티션에 대한 이니셜 컨텍스트는 파티션에

리소스 룩업 요청들을 디스패칭 하는데 사용하기 위해 생성될 수 있으며, 다른 어플리케이션들에 의해 파티션 내의 리소스들에 액세스하는 데에 재사용될 수 있다.

[0065] 일 실시예에 따르면, 파티션 JNDI 트리의 라이프-사이클은 파티션 JNDI 트리의 관련 파티션의 라이프-사이클과 동일하다. 파티션이 생성될 때, 파티션 JNDI 트리가 생성될 수 있다. 파티션 정보, 예를 들어 파티션 이름 및 파티션 ID는 생성된 파티션 JNDI 트리에 바인딩 될 수 있다. 파티션이 파괴되면, 그 파티션에 대한 파티션 JNDI 트리가 파괴된다.

[0066] 일 실시예에 따르면, 각각의 파티션 JNDI 트리는 관련 파티션 내의 리소스들에 액세스하기 위해 어플리케이션에 의해 사용될 수 있다. 어플리케이션은 도메인의 독립형 어플리케이션 클라이언트, 도메인의 관리되는 서버에 있는 어플리케이션, 파티션 내에 배포된 어플리케이션, 또는 다른 파티션에 배포된 어플리케이션 중 하나일 수 있다.

[0067] 전술된 어플리케이션들의 각각이 특정 파티션으로부터 리소스들을 요청하면, 이 어플리케이션은 해당 파티션과 관련된 JNDI 컨텍스트를 이니시에이트(initiate)할 수 있으며, 이 JNDI 컨텍스트를 사용하여 상기 어플리케이션으로부터의 리소스 요청들을 파티션 JNDI 트리에 위임(delegate)할 수 있다.

[0068] 일 실시예에 따르면, 일단 생성되면 JNDI 컨텍스트는 그 파티션과 계속 관련되도록 유지(remain)될 수 있어, JNDI 컨텍스트 내 모든 후속(subsequent) JNDI 동작들이 관련 파티션에 위임될 수 있고, 그 파티션의 컨텍스트 내에서 수행될 수 있다. JNDI 컨텍스트는 이 JNDI 컨텍스트를 생성하는데 사용된 스레드(thread)와는 다른 스레드들 상의 하나 이상의 어플리케이션들에 의해 사용될 수 있다.

[0069] 도 6은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서 도메인 내의 다수의 파티션 JNDI 트리들을 지원하기 위한 시스템을 도시한다.

[0070] 도 6에 도시된 바와 같이, 도메인 또는 글로벌 JNDI 트리(620)는 도메인 루트 네이밍 노드(661)을 포함할 수 있으며, 도메인 루트 네이밍 노드는 파티션 루트 네이밍 노트들의 컬렉션 예를 들어, 파티션 A 루트 네이밍 노드(663), 파티션 B 루트 네이밍 노드(665), 그리고 파티션 N 루트 네이밍 노드(667)를 더 포함한다. 각각의 파티션 루트 네이밍 노드들은 파티션 JNDI 트리 예를 들어, JNDI 트리 A(641), JNDI 트리 B(645), JNDI 트리 N(649)에 대한 루트 네이밍 노드일 수 있다.

[0071] 일 실시예에 따르면, 글로벌 JNDI 트리는 도메인 레벨 리소스들(628)에 바인딩된다. 각 파티션 JNDI 트리는 파티션(예를 들어, 파티션 A(635), 파티션 B(637), 또는 파티션 C(639))과 관련되고; 그 파티션에 배포되는 리소스들(예를 들어, 리소스 A(655), 리소스 B(657) 및 리소스 N(659))에 바인딩된다.

[0072] 어플리케이션이 특정 파티션 내 리소스에 액세스해야 할 때, 어플리케이션은 파티션 JNDI 트리의 시작점으로서 이니셜 JNDI 컨텍스트를 생성할 수 있다. 요청하는 어플리케이션이 어디에 위치했는지에 따라, 이니셜 JNDI 컨텍스트를 생성하는데 사용되는 파라미터(parameter)로서 제공자 URL이 제공될 수 있다. 만약 요청하는 어플리케이션이 요청되는 리소스와 동일한 파티션에 있을 경우, 제공자 URL은 필요하지 않다. 그렇지 않을 경우에는, 제공자 URL이 제공되어야 한다; 예를 들어, 요청하는 어플리케이션이 어플리케이션 서버 도메인에 대한 클라이언트 어플리케이션인 경우.

[0073] 도 6에 도시된 바와 같이, 클라이언트 어플리케이션(617)이 파티션 A 내의 리소스에 액세스해야 하는 경우, 파티션을 특정하는 제공자 URL과 함께 이니셜 JNDI 컨텍스트 A(618)가 생성될 수 있다.

[0074] 리스팅 1은 제공되는 제공자 URL 과 함께 초기 JNDI 컨텍스트를 생성하는 예시를 나타낸다.

```
Hashtable h = new Hashtable();
h.put(Context.INITIAL_CONTEXT_FACTORY,
"weblogic.jndi.WLInitialContextFactory");
h.put(Context.PROVIDER_URL, "t3://server-2:7001/partition-A");
h.put(Context.SECURITY_PRINCIPAL, "partition-A-user");
h.put(Context.SECURITY_CREDENTIALS, "partition-A-password");
Context ctx = null;
try {
      ctx = new InitialContext(h);
} catch (NamingException e) {
      e.printStackTrace();
} finally {
      try {
            ctx.close();
      } catch (NamingException e) {
            e.printStackTrace();
      }
}
```

[0075]

[0076] **리스팅 1**

[0077] 도시된 바와 같이, 리스팅 1은 파티션 A에 배포되는 리소스들을 요청하고, 그 파티션을 파라미터로서 특정하는 제공자 URL과 함께 이니셜 JNDI 컨텍스트를 생성하는 어플리케이션을 나타낸다.

[0078] 일 실시예에 따르면, JNDI 서브시스템이 이니셜 JNDI 컨텍스트를 생성하라는 요청들을 수신하면, JNDI 서브시스템은 제공자 URL을 사용하여 예를 들어, 특정된 파티션의 정보를 어느 파티션 루트 네이밍 노드가 포함하는지 검사함으로써, 제공자 URL에 지정된 대로 이니셜 컨텍스트를 파티션과 관련시킬 수 있다. 만약 제공자 URL이 제공되지 않으면, JNDI 서브시스템은 현재 컴포넌트 호출 컨텍스트(component invocation context)를 이니셜 JNDI 컨텍스트과 관련시킬 수 있다. 현재 컴포넌트 호출 컨텍스트는 요청하는 어플리케이션이 실행되고 있는 현재 파티션과 관련될 수 있다.

[0079] 일 실시예에 따르면, 일단 이니셜 JNDI 컨텍스트가 생성되면, 요청하는 어플리케이션에 의해 이니시에이트된 JNDI 동작들은 이니셜 JNDI 컨텍스트와 관련된 파티션 JNDI 트리 내에서 수행될 수 있다. 이니셜 JNDI 컨텍스트로부터의 요청들은 그 파티션과 관련된 JNDI 리소스들에 위임될 수 있다.

[0080] 또한, 이니셜 컨텍스트 및 파티션 A와의 관계(634)는 다른 어플리케이션들에 의한 재사용을 위해 도메인 내의 컨텍스트 객체 A(619)로서 유지(621)될 수 있다.

[0081] 도 7은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서 도메인 내의 다수의 파티션 JNDI 트리들을 지원하기 위한 시스템을 더 도시한다.

[0082] 도 7에 도시된 바와 같이, 위에서 생성된 컨텍스트 객체는 복수의 어플리케이션들 예를 들어, 다른 클라이언트 어플리케이션 B(717), 파티션 B 내의 어플리케이션(737), 그리고 도메인의 관리되는 서버 내의 어플리케이션(761)에 의한 재사용(743, 745, 747)을 위해 유지될 수 있다.

[0083] 일 실시예에 따르면, 이들 어플리케이션들은 파티션 A내의 리소스들에 액세스 할 때 컨텍스트 객체를 호출(call)할 수 있다. 이들 어플리케이션들로부터의 JNDI 동작들은 파티션 A와 관련된 JNDI 트리에 자동적으로 위임될 수 있고; 파티션 A의 컨텍스트 내에서 수행될 수 있다.

[0084] 일 실시예에 따르면, 컨텍스트 객체와 파티션 사이의 관계는 컨텍스트 객체를 파티션의 네임스페이스과 관련시킴으로써 유지될 수 있다. 현재 컴포넌트 호출 컨텍스트 대신에, 컨텍스트 객체와 관련된 파티션의 컴포넌트 호출 객체에서 JNDI 동작들이 프로세스되거나 또는 수행될 수 있도록 복수의 콜백(callback) 기능들이 JNDI 서브시스템에 제공될 수 있다.

[0085] 도 8은 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서 도메인 내의 다수의 파티션

JNDI 트리들을 지원하기 위한 방법을 도시한다.

[0086] 단계(811)에서, 어플리케이션 서버 환경 내의 도메인이 제공되고, 상기 도메인은 복수의 파티션들과 글로벌 JNDI 트리를 포함하고, 상기 글로벌 JNDI 트리는 파티션 루트 네이밍 노드들의 컬렉션을 유지하며, 각 파티션 루트 네이밍 노드는 파티션에 특정한 파티션 JNDI 트리와 관련된다.

[0087] (813)에서, 도메인 내의 JDNI 서브시스템에서 어플리케이션으로부터 이니셜 JNDI 컨텍스트를 생성하기 위한 요청이 수신된다.

[0088] 단계(815)에서, JNDI 서브시스템은 요청으로부터의 정보 및 파티션 루트 네이밍 노드들의 컬렉션을 검사하여, 특정 파티션과 관련된 이니셜 JNDI 컨텍스트를 생성한다.

[0089] (817)에서, 어플리케이션으로부터의 JNDI 동작들은 특정 파티션과 관련된 JNDI 트리에 위임된다.

[0090] 도 9는 일 실시예에 따른, 어플리케이션 서버 환경, 클라우드 또는 다른 환경에서 도메인 내의 다수의 파티션 JNDI 트리들을 지원하기 위한 대표적 시스템을 도시한다.

[0091] 일 실시예에 따르면, 어플리케이션 서버 환경은 파티션-인식 JNDI 룩업들 및 파티션에 배포된 리소스들에 대한 바인딩들을 수행할 수 있다. 이러한 환경에서, 어플리케이션은 오직 이 어플리케이션이 배포된 파티션의 범위(scope)에 있는 JNDI 리소스들에만 액세스할 수 있다. 예를 들어, 하나의 파티션에 배포된 어플리케이션은 다른 파티션의 JNDI 리소스들 또는 도메인의 JNDI 리소스들에 액세스할 수 없다.

[0092] 일 실시예에 따르면, 시스템은 각각의 파티션 루트 네이밍 노드들이 파티션 네임 스페이스와 관련된 파티션 루트 네이밍 노드들의 컬렉션(예를 들어, 파티션 A 네임스페이스(935) 및 파티션 B 네임스페이스(937))을 유지함으로써 JNDI 리소스들을 격리시킬 수 있다.

[0093] 일 실시예에 따르면, 파티션 네임스페이스(예를 들어, 파티션-레벨 JNDI 트리) 내의 각 노드는 파티션에 특정한 하나 이상의 값들에 바인딩함으로써, 파티션-인식형으로 만들어질 수 있다. 다른 파티션들에 대한 다른 네임스페이스들에 동일한 JNDI 이름이 바인딩 될 수 있다.

[0094] 예를 들어, 동일한 JNDI 리소스(예를 들어, 세션 리소스)가 파티션 A 네임스페이스, 파티션 B 네임스페이스, 그리고 글로벌 네임스페이스(920)에 바인딩 될 수 있다. 리소스에 대한 JNDI 룩업은 파티션 A 또는 파티션 B 내에 배포된 어플리케이션으로부터 수신될 수 있다. 파티션 A에 배포된 어플리케이션의 경우, 세션(945)이 반환될 수 있다. 파티션 B에 배포된 어플리케이션의 경우, 세션(955)이 반환될 수 있다.

[0095] 일 실시예에 따르면, JNDI 리소스가 파티션의 네임스페이스에 바인딩 되지 않은 경우, 파티션에 배치된 어플리케이션으로부터 수신된 리소스 룩업들은 예를 들어 NameNotFoundException과 같은 오류를 생성할 수 있다.

[0096] 도 9에 더 도시된 바와 같이, 글로벌 네임스페이스(920)는 도메인 루트 노드(930) 및 복수의 리소스 노드들을 포함할 수 있다. 글로벌 JNDI 트리 내의 노드들은 파티션-인식형이 아니다. 즉, 파티션-특정 정보들에 바인딩 되지 않는다.

[0097] 일 실시예에 따르면, 파티션-인식 네이밍 노드 및 파티션-미인식 노드는 동일한 JNDI 이름을 가질 수 있으나(예를 들어, 글로벌 네임스페이스의 세션 노드(975) 및 파티션 A 네임스페이스의 세션 노드(945)), 이들은 서로 다른 리소스들을 반환할 수 있다.

[0098] 이와 같이, 파티션-인식 노드들은 멀티테넌트 어플리케이션 서버 환경의 필요조건들을 충족시키기 위해, 향상된 룩업/바인드/리바인드(rebind)/언바인드(unbind) 기능들을 제공한다.

[0099] 본 발명은 하나 이상의 종래의 범용 또는 특수 디지털 컴퓨터, 컴퓨팅 디바이스, 머신, 또는 마이크로 프로세서를 사용하여 편리하게 구현될 수 있으며, 본 발명의 교시에 따라 프로그래밍된 하나 이상의 프로세서들, 메모리 및/또는 컴퓨터 판독가능 저장 매체를 포함한다. 본 발명의 교시에 기초하여 소프트웨어 분야의 통상의 기술자에게 명백하도록, 숙련된 프로그래머들에 의해 적절한 소프트웨어 코딩이 손쉽게 준비될 수 있다.

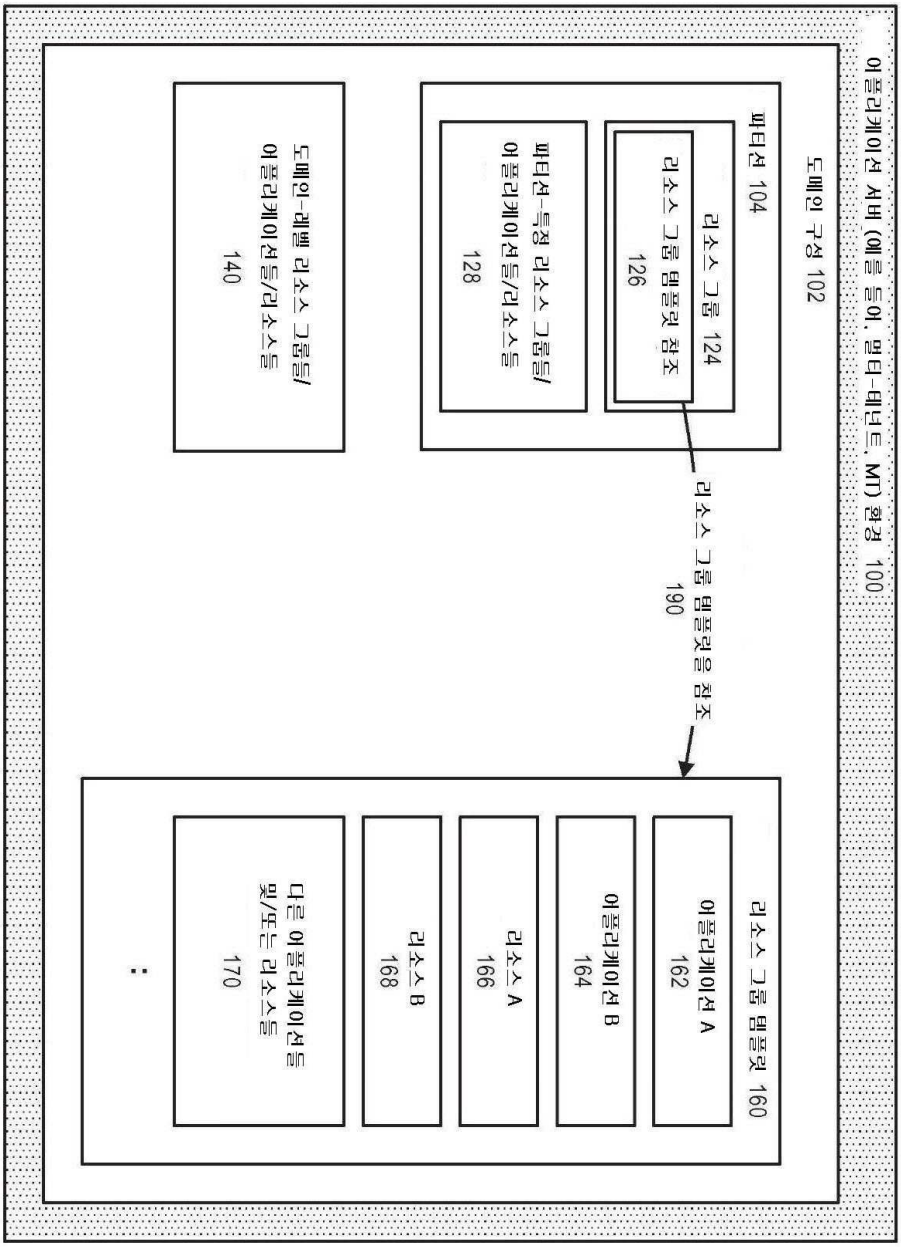
[0100] 일부 실시예들에서, 본 발명은 그에/그 안에 명령어들이 저장된 비-일시적인 저장 매체 또는 컴퓨터 판독가능 매체(들)인 컴퓨터 프로그램 제품을 포함하며, 명령어들은 본 발명의 프로세스들 중 임의의 것을 수행하기 위해 컴퓨터를 프로그래밍하는 데에 사용될 수 있다. 저장 매체는 플로피 디스크들, 광학 디스크들, DVD, CD-ROM들, 마이크로드라이브 및 자기-광학 디스크들, ROM들, RAM들, EPROM들, EEPROM들, DRAM들, VRAM들, 플래시 메모리 디바이스들, 자기 또는 광학 카드들, 나노시스템들(분자 메모리 IC 포함), 또는 명령어들 및/또는 데이터를 저

장하기에 적합한 임의의 형태의 매체 또는 디바이스를 포함할 수 있지만, 이에 제한되지 않는다.

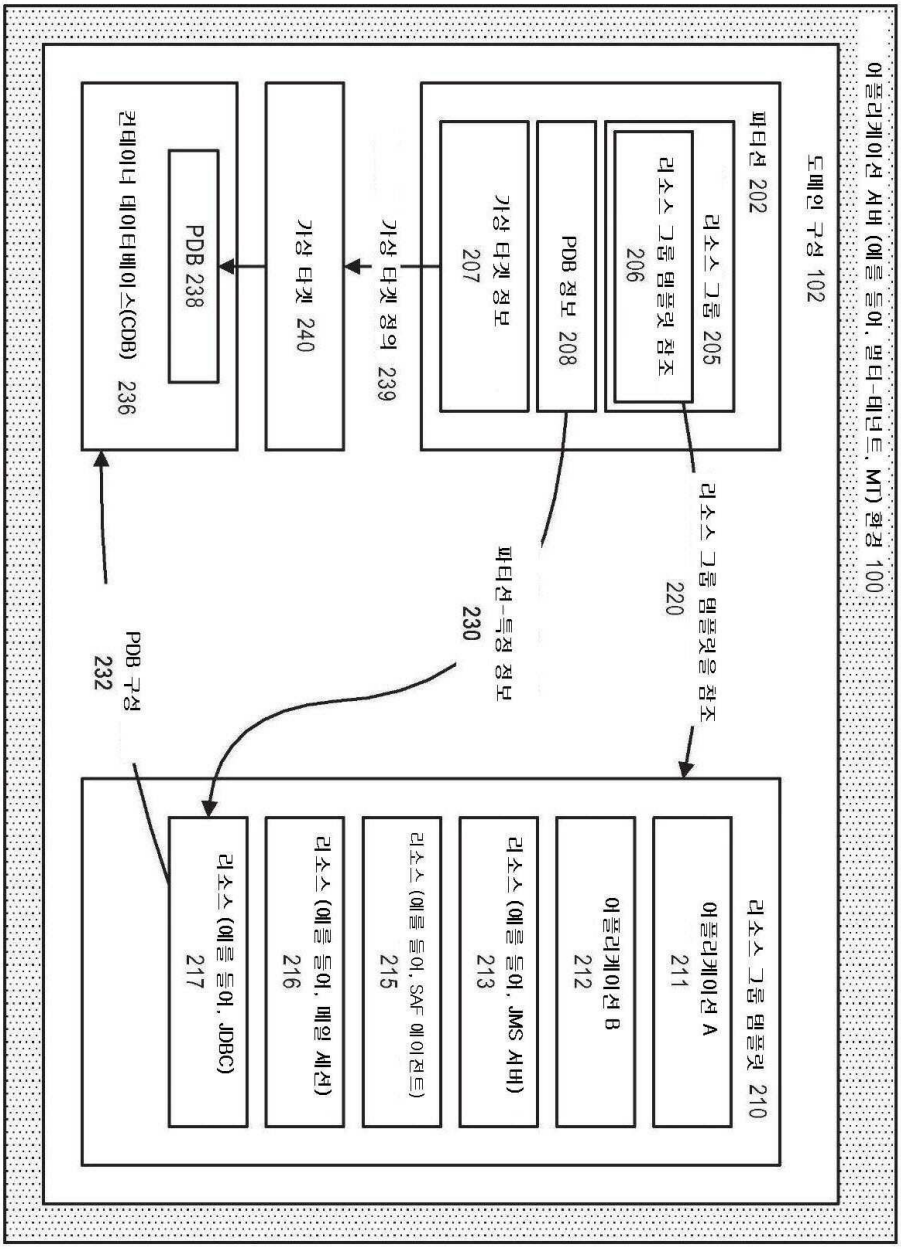
[0101] 본 발명의 전술된 설명은 예시 및 설명의 목적으로 제공되었다. 이는 완전하다거나 본 발명을 개시된 명확한 형태들로 제한하도록 의도되지 않았다. 많은 수정들 및 변형들이 통상의 기술자들에게 명백할 것이다. 본 발명의 원리들 및 실질적인 적용들을 가장 잘 설명하기 위해 실시예들이 선택되고 설명되었으며, 이에 의해 다른 통상의 기술자들이 다양한 실시예들에 대한 고려된 특정한 사용에 적합한 다양한 수정들과 함께 본 발명을 이해할 수 있도록 한다. 본 발명의 범위는 다음의 청구항들 및 그들에 상응하는 것들에 의해 정의되도록 의도된다.

도면

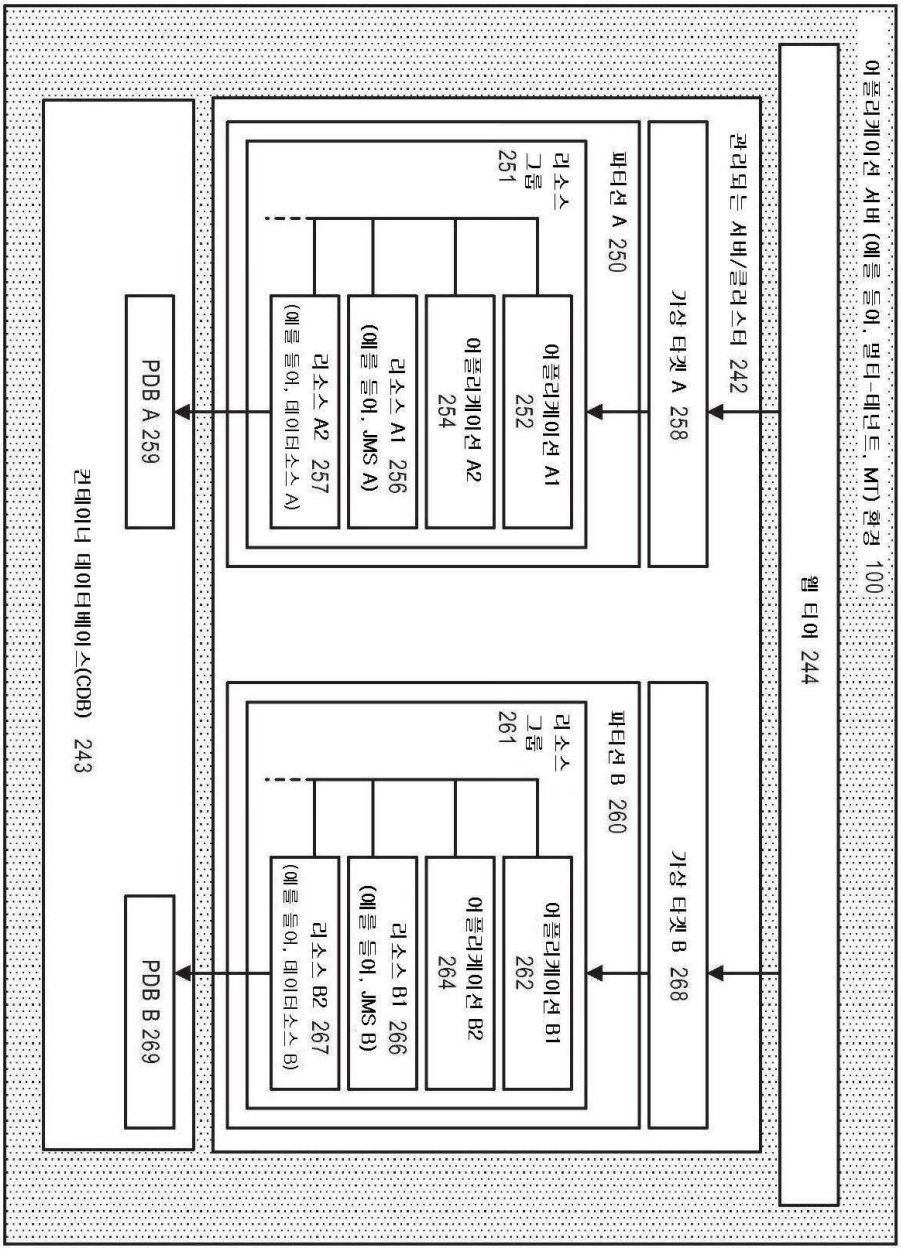
도면1



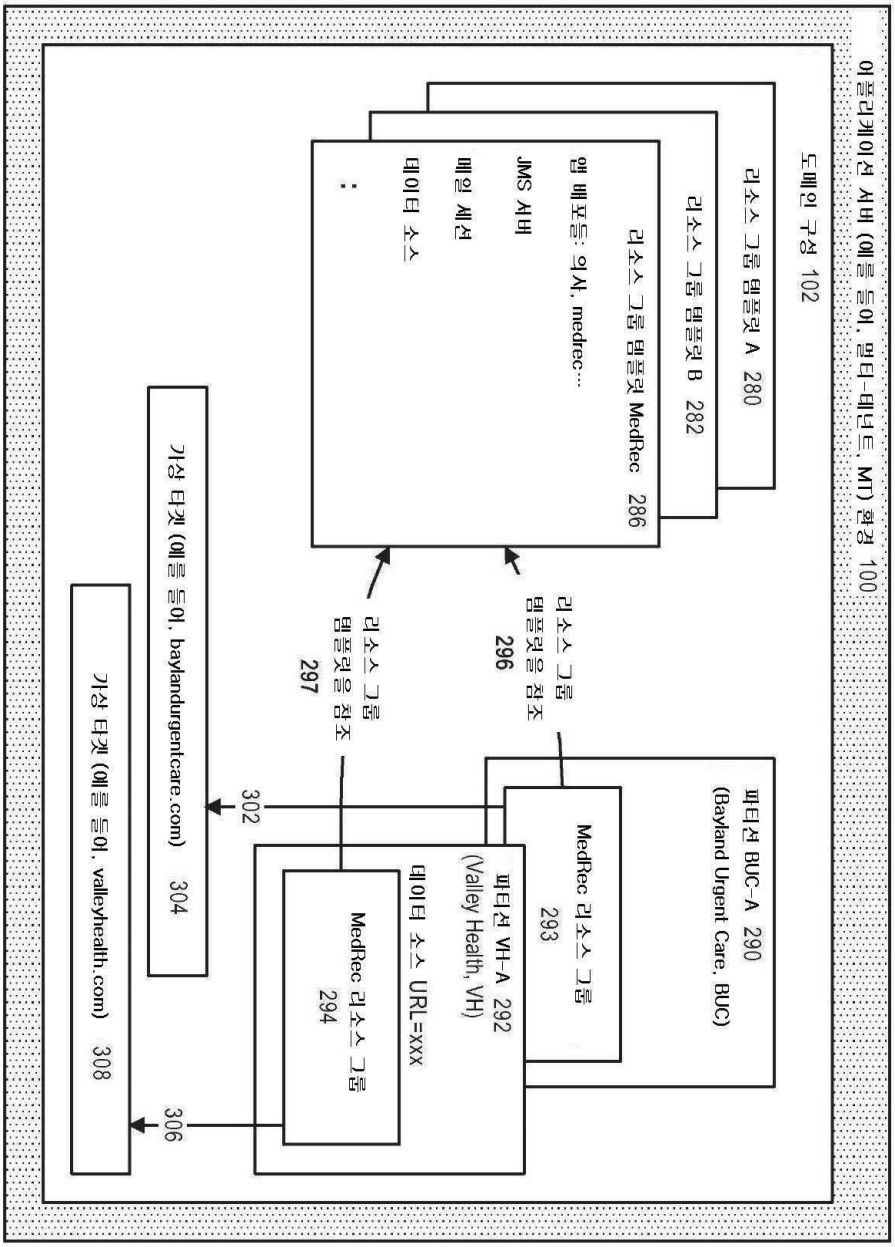
도면2


어플리케이션 서버 (예를 들어, 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
컨테이너 데이터베이스(CDB) 236
PDB 238
리소스 그룹 템플릿을 참조 220
파티션-특정 정보 230
PDB 구성 232
리소스 그룹 템플릿 210
어플리케이션 A 211
어플리케이션 B 212
리소스 (예를 들어, JMS 서버) 213
리소스 (예를 들어, SAF 에이전트) 215
리소스 (예를 들어, 메일 세션) 216
리소스 (예를 들어, JDBC) 217

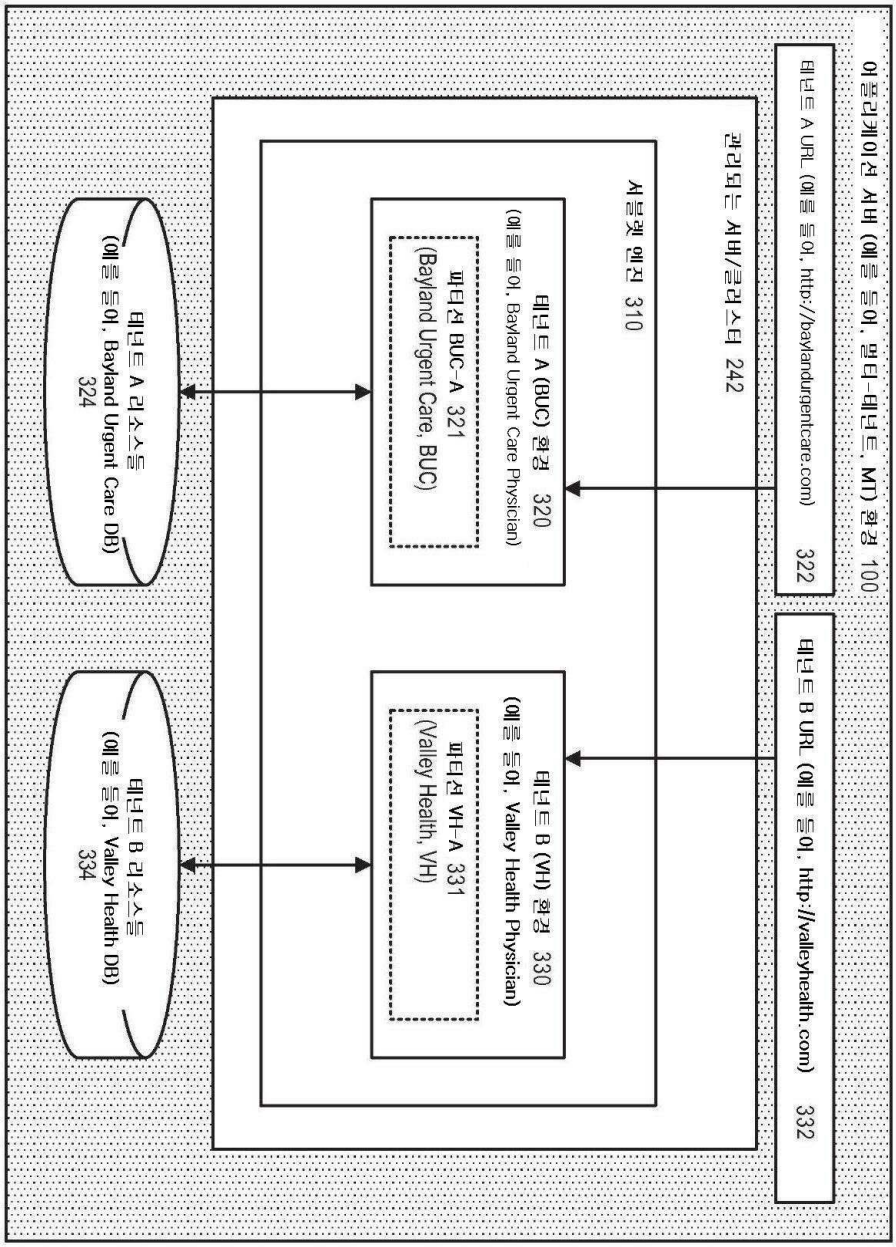
도면3


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

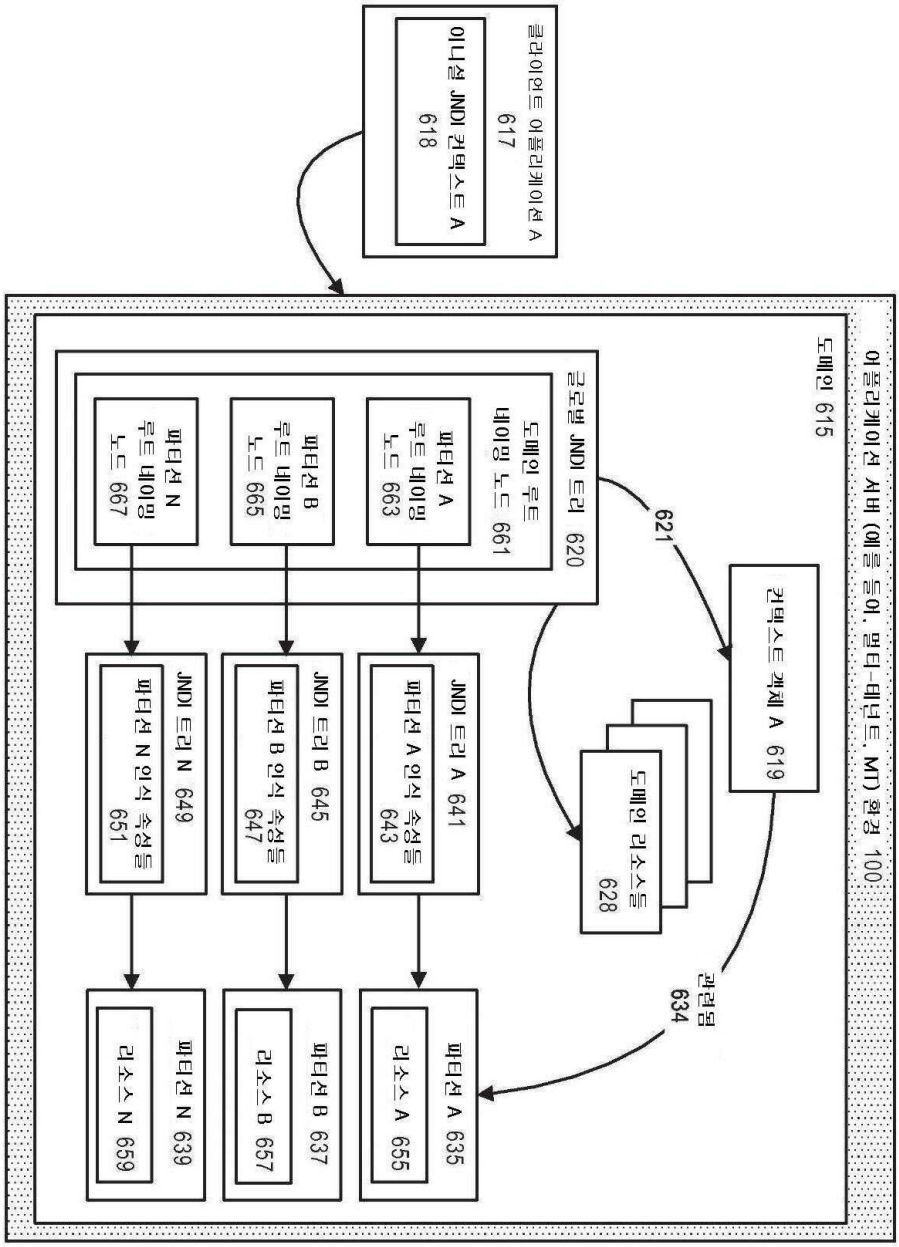
도면4


어플리케이션 서버 (예를 들어, 멀티-테넌트, MT) 환경 100
도메인 구성 102
리소스 그룹 템플릿 A 280
리소스 그룹 템플릿 B 282
리소스 그룹 템플릿 MedRec 286
앱 배포들: 의사, medrec…
JMS 서버
메일 세션
데이터 소스
:
리소스 그룹 템플릿을 참조 296
리소스 그룹 템플릿을 참조 297
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
302
306
가상 타겟 (예를 들어, baylandurgentcare.com) 304
가상 타겟 (예를 들어, valleyhealth.com) 308

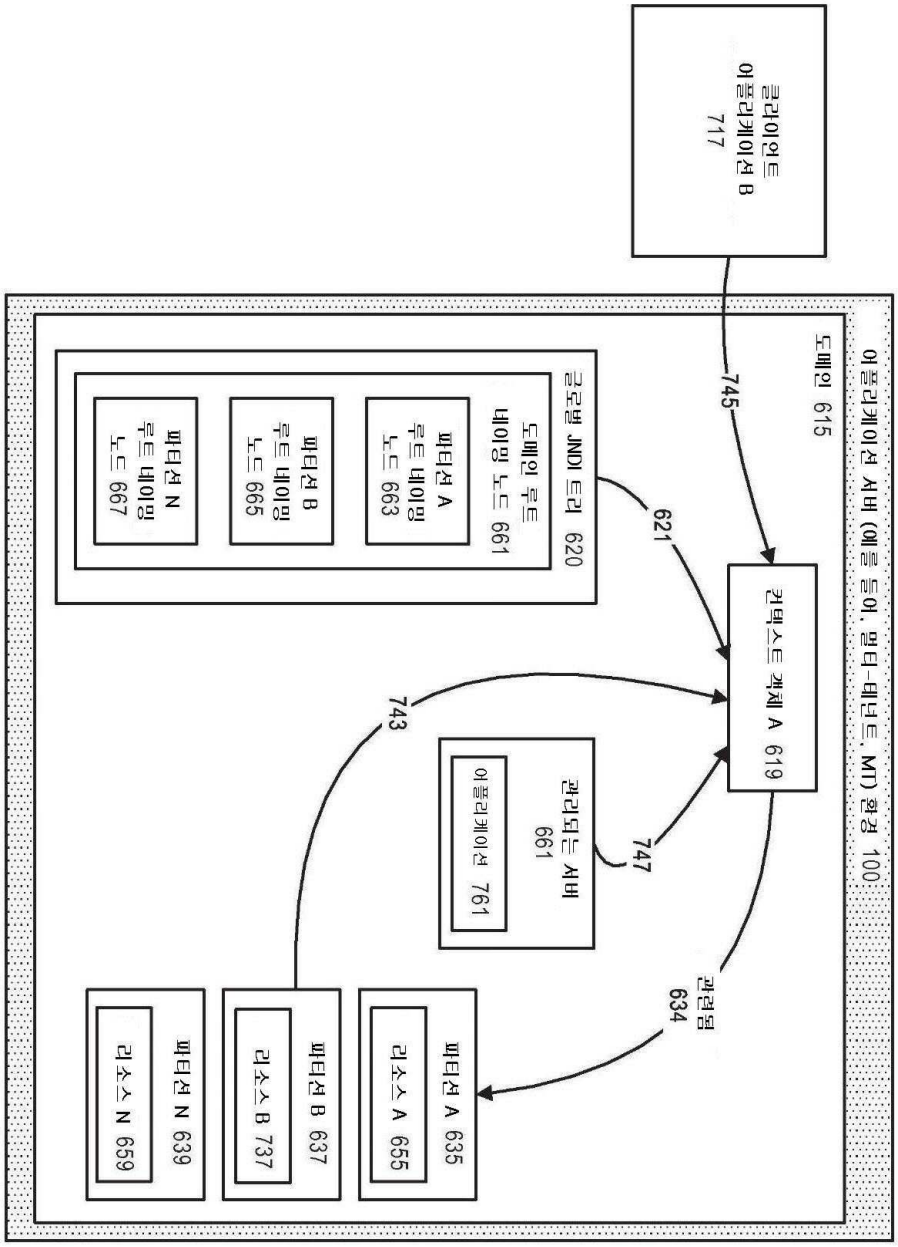
도면5


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

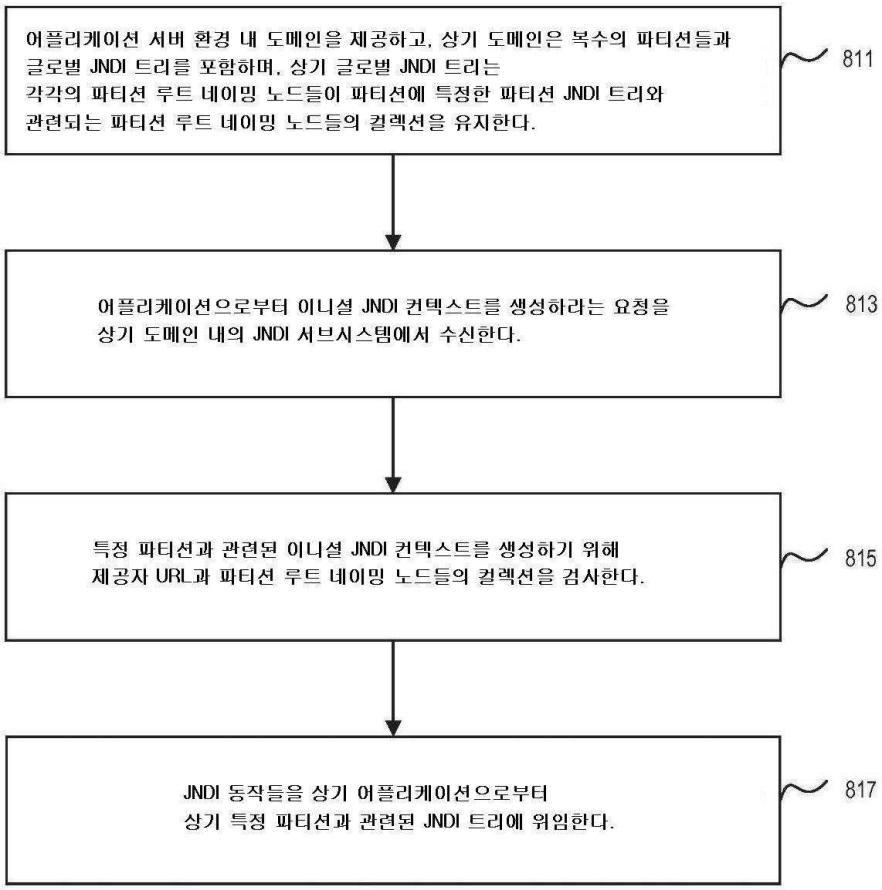
도면6


애플리케이션 서버 (예를 들어, 멀티-테넌트, MT) 환경 100
도메인 615
클라이언트 애플리케이션 A 617
이니셜 JNDI 컨텍스트 A 618
글로벌 JNDI 트리 620
621
도메인 루트 네이밍 노드 661
파티션 A 루트 네이밍 노드 663
파티션 B 루트 네이밍 노드 665
파티션 N 루트 네이밍 노드 667
컨텍스트 객체 A 619
도메인 리소스들 628
관련됨 634
JNDI 트리 A 641
파티션 A 인식 속성들 643
JNDI 트리 B 645
파티션 B 인식 속성들 647
JNDI 트리 N 649
파티션 N 인식 속성들 651
파티션 A 635
리소스 A 655
파티션 B 637
리소스 B 657
파티션 N 639
리소스 N 659

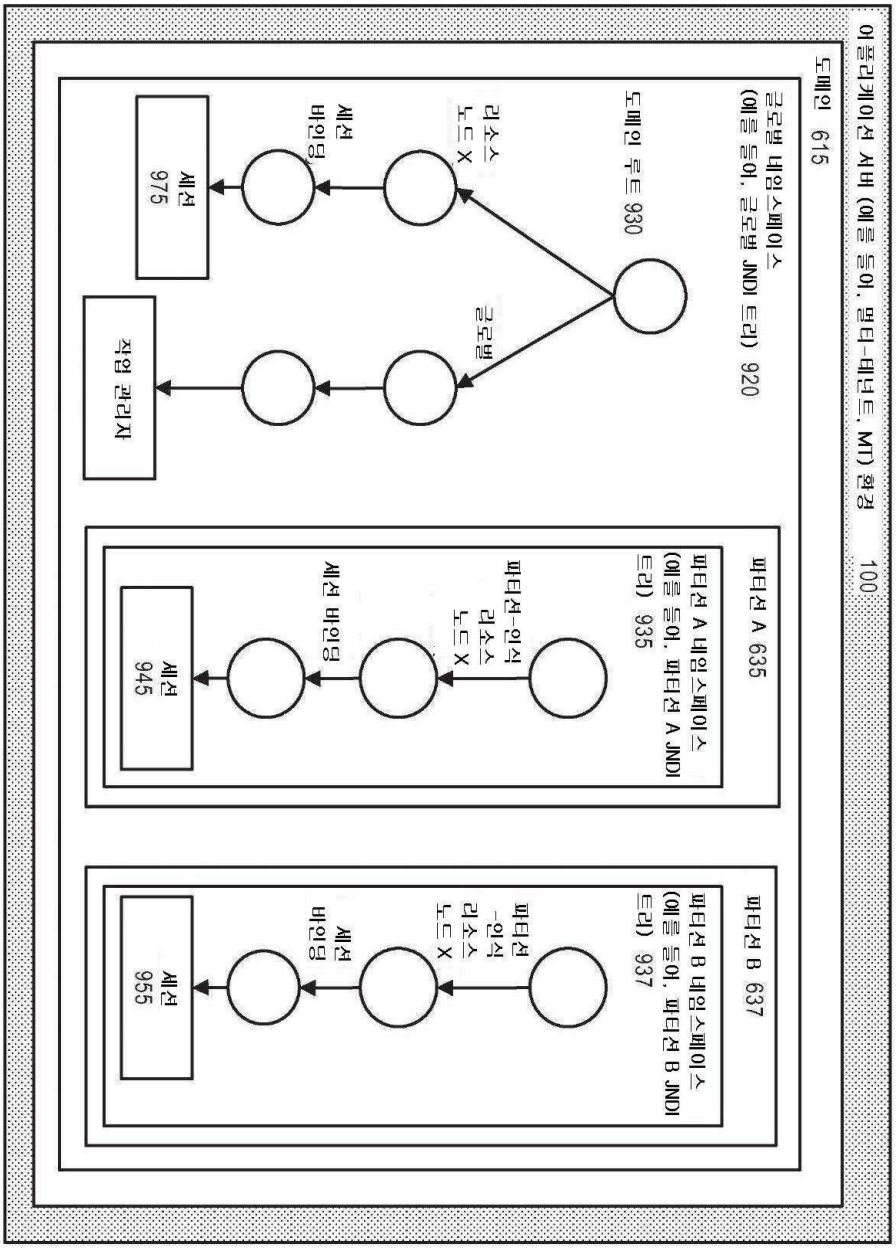
도면7


클라이언트
어플리케이션 B
717
745
어플리케이션 서버 (예를 들어, 멀티-테넌트, MT) 환경 100
도메인 615
글로벌 JNDI 트리 620
도메인 루트
네이밍 노드 661
파티션 A
루트 네이밍
노드 663
파티션 B
루트 네이밍
노드 665
파티션 N
루트 네이밍
노드 667
621
컨텍스트 객체 A 619
743
관리되는 서버
661
어플리케이션 761
747
관련됨
634
파티션 A 635
리소스 A 655
파티션 B 637
리소스 B 737
파티션 N 639
리소스 N 659

도면8


어플리케이션 서버 환경 내 도메인을 제공하고, 상기 도메인은 복수의 파티션들과 글로벌 JNDI 트리를 포함하며, 상기 글로벌 JNDI 트리는 각각의 파티션 루트 네이밍 노드들이 파티션에 특정한 파티션 JNDI 트리와 관련되는 파티션 루트 네이밍 노드들의 컬렉션을 유지한다.
811
어플리케이션으로부터 이니셜 JNDI 컨텍스트를 생성하라는 요청을 상기 도메인 내의 JNDI 서브시스템에서 수신한다.
813
특정 파티션과 관련된 이니셜 JNDI 컨텍스트를 생성하기 위해 제공자 URL과 파티션 루트 네이밍 노드들의 컬렉션을 검사한다.
815
JNDI 동작들을 상기 어플리케이션으로부터 상기 특정 파티션과 관련된 JNDI 트리에 위임한다.
817

도면9


어플리케이션 서버 (예를 들어, 멀티-테넌트, MT) 환경 100
도메인 615
글로벌 네임스페이스 (예를 들어, 글로벌 JNDI 트리) 920
도메인 루트 930
리소스 노드 X
세션 바인딩
세션 975
글로벌
작업 관리자
파티션 A 635
파티션 A 네임스페이스 (예를 들어, 파티션 A JNDI 트리) 935
파티션-인식 리소스 노드 X
세션 바인딩
세션 945
파티션 B 637
파티션 B 네임스페이스 (예를 들어, 파티션 B JNDI 트리) 937
파티션 -인식 리소스 노드 X
세션 바인딩
세션 955