



등록특허 10-2251803



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년05월14일
(11) 등록번호 10-2251803
(24) 등록일자 2021년05월07일

- (51) 국제특허분류(Int. Cl.)
H04L 29/08 (2006.01) *H04L 29/06* (2006.01)
- (52) CPC특허분류
H04L 67/28 (2013.01)
G06F 16/957 (2019.01)
- (21) 출원번호 10-2017-7010513
- (22) 출원일자(국제) 2015년04월27일
심사청구일자 2020년03월18일
- (85) 번역문제출일자 2017년04월18일
- (65) 공개번호 10-2017-0060070
- (43) 공개일자 2017년05월31일
- (86) 국제출원번호 PCT/US2015/027763
- (87) 국제공개번호 WO 2016/048419
국제공개일자 2016년03월31일
- (30) 우선권주장
62/054,613 2014년09월24일 미국(US)
14/696,432 2015년04월25일 미국(US)

- (56) 선행기술조사문현
US20080140857 A1
US20120210335 A1
US20110161477 A1
JP2012516112 A

전체 청구항 수 : 총 17 항

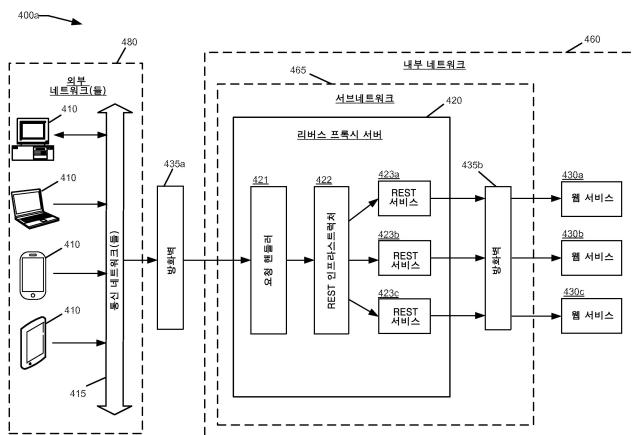
심사관 : 이준석

(54) 발명의 명칭 리버스 프록시 서버들 내의 서비스들

(57) 요 약

본 발명의 실시예들은 컴퓨터 네트워크들 사이에서 송신되는 메시지들을 프로세싱하기 위한 기술들을 제공한다. 웹 서비스들 및 다른 웹 컨텐트에 대한 클라이언트 디바이스들로부터의 요청들과 같은 메시지들이 복수의 컴퓨터 네트워크들 사이에서 송신될 수 있다. 프록시 서버들과 같은 중간 디바이스들 또는 애플리케이션들이 통신 엔드

(뒷면에 계속)

대 표 도

포인트들 사이에서 메시지들을 수신하고, 프로세싱하며, 송신할 수 있다. 일부 실시예들에 있어서, 리버스 프록시 서버는 리버스 프록시 서버 내에서 동적으로 표현 상태 전환(Representational State Transfer; REST) 서비스들 및 REST 자원들을 생성하도록 구성될 수 있다. 리버스 프록시 서버 내의 REST 서비스들 및 REST 자원들은 클라이언트 디바이스들로부터 들어오는 요청들을 처리하고 백엔드 웹 서비스들을 호출할 수 있으며, 그럼으로써 리버스 프록시 서버 상의 다양한 보안 정책들의 설계 추상화 및/또는 집행을 가능하게 한다.

(52) CPC특허분류

H04L 63/0209 (2013.01)

H04L 63/0281 (2013.01)

H04L 63/083 (2013.01)

H04L 63/105 (2013.01)

H04L 63/20 (2013.01)

H04L 65/105 (2013.01)

H04L 67/02 (2013.01)

H04L 67/2823 (2013.01)

명세서

청구범위

청구항 1

컴퓨터 네트워크들 사이에서 웹 서비스 요청들을 송신하는 방법으로서, 내부 컴퓨터 네트워크와 통신하는 프록시 서버에서, 상기 내부 컴퓨터 네트워크로부터 분리된 외부 컴퓨터 네트워크 내의 클라이언트 디바이스로부터 웹 서비스 요청을 수신하는 단계; 상기 웹 서비스 요청 내의 제 1 요청된 자원을 식별하는 단계; 상기 제 1 요청된 자원이 상기 프록시 서버 내의 제 1 표현 상태 전환(Representational State Transfer; REST) 웹 서비스에 의해 노출된다는 것을 결정하는 단계; 상기 제 1 REST 웹 서비스에 의해 노출되는 상기 제 1 요청된 자원이 상기 프록시 서버 내에 존재하지 않는다는 것을 결정하는 단계; 상기 제 1 요청된 자원이 상기 프록시 서버 내에 존재하지 않는다고 결정하는 것에 응답하여, 상기 프록시 서버 내에서 상기 제 1 요청된 자원을 생성하는 단계로서, 상기 제 1 요청된 자원은 상기 웹 서비스 요청이 상기 클라이언트 디바이스로부터 수신된 이후에 생성되는, 단계; 상기 프록시 서버 내에서 상기 제 1 REST 웹 서비스를 호출하고, 상기 웹 서비스 요청을 상기 생성된 제 1 요청된 자원으로 포워딩(forward)하는 단계; 및 상기 프록시 서버 내에서의 상기 제 1 REST 웹 서비스 내의 상기 생성된 제 1 요청된 자원의 실행 동안, 상기 내부 컴퓨터 네트워크 내의 컴퓨터 서버 내에서 제 2 웹 서비스를 호출하는 단계를 포함하는, 방법.

청구항 2

삭제

청구항 3

청구항 1에 있어서, 상기 방법은, 상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내에서 상기 제 2 웹 서비스에 의해 제공되는 자원들의 세트를 기술하는 웹 애플리케이션 기술 언어(Web Application Description Language; WADL) 파일을 액세스하는 단계; 및 상기 제 2 웹 서비스에 의해 제공되는 상기 자원들의 세트의 상기 WADL 파일 내의 설명을 사용하여, 상기 프록시 서버 내에서 상기 제 1 REST 웹 서비스 내에 하나 이상의 자원들을 생성하는 단계를 더 포함하는, 방법.

청구항 4

청구항 3에 있어서, 상기 프록시 서버 내에서 상기 제 1 REST 웹 서비스 내에 상기 제 1 요청된 자원을 생성하는 단계는, 상기 WADL 파일 내의 하나 이상의 자원 설명들을 수정하는 단계; 상기 수정된 자원 설명들에 기초하여 하나 이상의 REST 자원들을 생성하는 단계; 및 상기 프록시 서버 내에서 상기 제 1 REST 웹 서비스 내에 상기 REST 자원들의 각각을 배포하는 단계를 포함하는, 방법.

청구항 5

청구항 1, 청구항 3 또는 청구항 4 중 어느 한 항에 있어서,

상기 제 2 웹 서비스는 상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내의 REST 웹 서비스인, 방법.

청구항 6

청구항 5에 있어서,

상기 프록시 서버 내의 상기 제 1 REST 웹 서비스는, 상기 제 2 REST 웹 서비스를 호출하도록 구성된 적어도 하나의 자원을 포함하며 상기 내부 컴퓨터 네트워크 내의 상이한 컴퓨터 서버에 의해 노출되는 제 3 REST 웹 서비스를 호출하도록 구성된 적어도 하나의 자원을 포함하는 복수의 자원들을 노출시키는, 방법.

청구항 7

청구항 5에 있어서,

상기 제 2 REST 웹 서비스는 복수의 자원들을 노출시키며, 상기 프록시 서버 내의 상기 제 1 REST 웹 서비스는 상기 제 2 REST 웹 서비스에 의해 노출되는 상기 복수의 자원들의 서브셋트를 노출시키는, 방법.

청구항 8

청구항 1에 있어서,

상기 제 2 웹 서비스는 상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내의 단순 객체 접근 프로토콜 (Simple Object Access protocol; SOAP) 웹 서비스인, 방법.

청구항 9

청구항 1에 있어서,

상기 방법은,

상기 프록시 서버 내에서 상기 웹 서비스 요청에 대한 미리 결정된 프로세싱 흐름 내의 현재 포인트를 결정하는 단계;

상기 웹 서비스 요청에 대한 상기 미리 결정된 프로세싱 흐름 내의 상기 결정된 현재 포인트에 기초하여, 하나 이상의 보안 정책들을 검색하는 단계; 및

상기 보안 정책들에 따라 상기 웹 서비스 요청을 프로세싱하는 단계로서, 상기 웹 서비스 요청은 상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버에 의해 노출되는 상기 제 2 웹 서비스를 호출하기 이전에 프로세싱되는, 단계를 더 포함하는, 방법.

청구항 10

시스템으로서,

하나 이상의 프로세서들을 포함하는 프로세싱 유닛; 및

상기 프로세싱 유닛과 결합되며 이에 의해 판독이 가능하고, 상기 프로세싱 유닛에 의해 실행될 때, 상기 프로세싱 유닛이 하기의 동작들을 수행하게끔 하는 명령어들의 세트를 저장하는 메모리로서, 상기 동작들은,

외부 컴퓨터 네트워크 내의 클라이언트 디바이스로부터 웹 서비스 요청을 수신하는 동작으로서, 상기 시스템은 상기 외부 컴퓨터 네트워크로부터 분리된 내부 컴퓨터 네트워크의 서브네트워크 내에서 동작하도록 구성되는, 동작;

상기 웹 서비스 요청 내의 제 1 요청된 자원을 식별하는 동작;

상기 제 1 요청된 자원이 상기 시스템 내의 제 1 표현 상태 전환(Representational State Transfer; REST) 웹 서비스에 의해 노출된다는 것을 결정하는 동작;

상기 제 1 REST 웹 서비스에 의해 노출되는 상기 제 1 요청된 자원이 상기 시스템 내에 존재하지 않는다는 것을 결정하는 동작;

상기 제 1 요청된 자원이 상기 시스템 내에 존재하지 않는다고 결정하는 것에 응답하여, 상기 시스템 내에서 상기 제 1 요청된 자원을 생성하는 동작으로서, 상기 제 1 요청된 자원은 상기 웹 서비스 요청이 상기 클라이언트 디바이스로부터 수신된 이후에 생성되는, 동작;

상기 시스템 내에서 상기 제 1 REST 웹 서비스를 호출하고, 상기 웹 서비스 요청을 상기 생성된 제 1 요청된 자원으로 포워딩하는 동작; 및

상기 시스템 내에서의 상기 제 1 REST 웹 서비스 내의 상기 생성된 제 1 요청된 자원의 실행 동안, 상기 내부 컴퓨터 네트워크 내의 컴퓨터 서버 내에서 제 2 웹 서비스를 호출하는 동작을 포함하는, 시스템.

청구항 11

삭제

청구항 12

청구항 10에 있어서,

상기 메모리는 상기 프로세싱 유닛에 의해 실행될 때 상기 프로세싱 유닛이 하기의 동작들을 수행하게끔 하는 명령어들을 추가로 저장하며, 상기 동작들은,

상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내에서 상기 제 2 웹 서비스에 의해 제공되는 자원들의 세트를 기술하는 웹 애플리케이션 기술 언어(Web Application Description Language; WADL) 파일을 액세스하는 동작; 및

상기 제 2 웹 서비스에 의해 제공되는 상기 자원들의 세트의 상기 WADL 파일 내의 설명을 사용하여, 상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 하나 이상의 자원들을 생성하는 동작을 포함하는, 시스템.

청구항 13

청구항 12에 있어서,

상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 상기 제 1 요청된 자원을 생성하는 동작은,

상기 WADL 파일 내의 하나 이상의 자원 설명들을 수정하는 동작;

상기 수정된 자원 설명들에 기초하여 하나 이상의 REST 자원들을 생성하는 동작; 및

상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 상기 REST 자원들의 각각을 배포하는 동작을 포함하는, 시스템.

청구항 14

청구항 10, 청구항 12, 또는 청구항 13 중 어느 한 항에 있어서,

상기 제 2 웹 서비스는 상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내의 REST 웹 서비스인, 시스템.

청구항 15

청구항 14에 있어서,

상기 시스템 내의 상기 제 1 REST 웹 서비스는, 상기 제 2 REST 웹 서비스를 호출하도록 구성된 적어도 하나의 자원을 포함하며 상기 내부 컴퓨터 네트워크 내의 상이한 컴퓨터 서버에 의해 노출되는 제 3 REST 웹 서비스를 호출하도록 구성된 적어도 하나의 자원을 포함하는 복수의 자원들을 노출시키는, 시스템.

청구항 16

청구항 14에 있어서,

상기 제 2 REST 웹 서비스는 복수의 자원들을 노출시키며, 상기 시스템 내의 상기 제 1 REST 웹 서비스는 상기 제 2 REST 웹 서비스에 의해 노출되는 상기 복수의 자원들의 서브셋트를 노출시키는, 시스템.

청구항 17

프로세서에 의해 실행될 때, 상기 프로세서가 하기의 동작들을 수행하게끔 하는 저장된 명령어들의 세트를 포함하는 비-일시적인 컴퓨터-판독가능 메모리로서, 상기 동작들은,

내부 컴퓨터 네트워크의 서브네트워크 내에서 동작하도록 구성된 시스템에서 웹 서비스 요청을 수신하는 동작으로서, 상기 웹 서비스 요청은 상기 내부 컴퓨터 네트워크로부터 분리된 외부 컴퓨터 네트워크 내의 클라이언트 디바이스로부터 수신되는, 동작;

상기 웹 서비스 요청 내의 제 1 요청된 자원을 식별하는 동작;

상기 제 1 요청된 자원이 상기 시스템 내의 제 1 표현 상태 전환(Representational State Transfer; REST) 웹 서비스에 의해 노출된다는 것을 결정하는 동작;

상기 제 1 REST 웹 서비스에 의해 노출되는 상기 제 1 요청된 자원이 상기 시스템 내에 존재하지 않는다는 것을 결정하는 동작;

상기 제 1 요청된 자원이 상기 시스템 내에 존재하지 않는다고 결정하는 것에 응답하여, 상기 시스템 내에서 상기 제 1 요청된 자원을 생성하는 동작으로서, 상기 제 1 요청된 자원은 상기 웹 서비스 요청이 상기 클라이언트 디바이스로부터 수신된 이후에 생성되는, 동작;

상기 시스템 내에서 상기 제 1 REST 웹 서비스를 호출하고, 상기 웹 서비스 요청을 상기 생성된 제 1 요청된 자원으로 포워딩하는 동작; 및

상기 시스템 내에서의 상기 제 1 REST 웹 서비스 내의 상기 생성된 제 1 요청된 자원의 실행 동안, 상기 내부 컴퓨터 네트워크 내의 컴퓨터 서버 내에서 제 2 웹 서비스를 호출하는 동작을 포함하는, 컴퓨터-판독가능 메모리.

청구항 18

삭제

청구항 19

청구항 17에 있어서,

상기 컴퓨터-판독가능 메모리는 상기 프로세서에 의해 실행될 때, 상기 프로세서가 하기의 동작들을 수행하게끔 하는 저장된 명령어들 추가로 포함하며, 상기 동작들은,

상기 내부 컴퓨터 네트워크 내의 상기 컴퓨터 서버 내에서 상기 제 2 웹 서비스에 의해 제공되는 자원들의 세트를 기술하는 웹 애플리케이션 기술 언어(Web Application Description Language; WADL) 파일을 액세스하는 동작; 및

상기 제 2 웹 서비스에 의해 제공되는 상기 자원들의 세트의 상기 WADL 파일 내의 설명을 사용하여, 상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 하나 이상의 자원들을 생성하는 동작을 포함하는, 컴퓨터-판독가능 메모리.

청구항 20

청구항 19에 있어서,

상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 상기 자원들을 생성하는 동작은,

상기 WADL 파일 내의 하나 이상의 자원 설명들을 수정하는 동작;

상기 수정된 자원 설명들에 기초하여 하나 이상의 REST 자원들을 생성하는 동작; 및

상기 시스템 내에서 상기 제 1 REST 웹 서비스 내에 상기 REST 자원들의 각각을 배포하는 동작을 포함하는, 컴퓨터-판독가능 메모리.

발명의 설명

기술 분야

[0001]

관련 출원들

[0002]

본 출원은, "MOBILE SECURITY ACCESS SERVER(MSAS)"라는 명칭으로 2014년 09월 24일자로 출원된 미국 특허 출원번호 제62/054,613호의 이익 및 우선권을 주장하는, "SERVICES WITHIN REVERSE PROXY SERVERS"이라는 명칭으로 2015년 04월 25일자로 출원된 미국 정규 특허 출원번호 제14/696,432호의 이익 및 우선권을 주장한다. 이 상에서 식별된 특허 출원들의 전체 내용들은 모든 목적들을 위하여 본원에 참조로서 통합된다.

배경 기술

[0003]

본 개시는 전반적으로 컴퓨터 네트워크들에 걸쳐 보안을 제공하기 위한 시스템들, 방법들 및 컴퓨터-판독가능 매체들에 관한 것이다. 보다 더 구체적으로, 본 개시는, 클라이언트 디바이스들과 백엔드(backend) 웹 애플리케이션들 및 서비스들 사이에 구현되는 프록시 서버에서 보안 서비스들 및 다른 기능들을 제공하기 위한 시스템들, 방법들, 및 컴퓨터-판독가능 매체들에 관한 것이다. 이러한 보안 서비스들은, 다른 것들 중에서도 특히 인증, 인가, 감사, 싱글 사인 온(single sign on), 보안 정책 집행, 키 관리 및 분배, 보안 통신, 보안 데이터 저장, 및 보안 데이터 공유를 포함할 수 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0004]

본원에서 설명되는 측면들은 컴퓨터 네트워크들 사이에서 송신되는 메시지들을 프로세싱하기 위한 다양한 기술들을 제공한다. 일부 실시예들에 있어서, 백엔드 웹 서비스들, 애플리케이션들, 및 다른 웹 컨텐트에 대한 클라이언트 디바이스들로부터의 요청들과 같은 메시지들이 복수의 컴퓨터 네트워크들 사이에서 송신될 수 있다. 물리적인 또는 논리적인 서브네트워크(subnetwork) 내에 구현된 프록시 서버(proxy server)들과 같은 하나 이상의 중간 디바이스들 또는 애플리케이션들이 통신 엔드포인트(endpoint)들 사이에서 메시지들을 수신하고, 프로세싱 하며, 송신할 수 있다. 일부 실시예들에 있어서, 리버스 프록시 서버는 리버스 프록시 서버 내에서 동적으로 표현 상태 전환(Representational State Transfer; REST) 서비스들 및 REST 자원들을 생성하도록 구성될 수 있다. 리버스 프록시 서버 내의 REST 서비스들 및 REST 자원들은 클라이언트 디바이스들로부터 들어오는

(incoming) 요청들을 처리하고 백엔드 웹 서비스들을 호출할 수 있으며, 그럼으로써 리버스 프록시 서버 상의 다양한 보안 정책들의 설계 추상화 및/또는 집행을 가능하게 한다.

[0005] 본원에서 설명되는 특정 측면들에 따르면, 프록시 서버는 프록시 서버 내의 REST 웹 서비스에 의해 노출되는 특정 자원에 대한 웹 서비스 요청을 수신할 수 있다. 프록시 서버의 REST 웹 서비스 내의 적절한 자원이 호출될 수 있으며, 결과적으로 백엔드 웹 서비스를 호출할 수 있다. 일부 경우들에 있어서, 리버스 프록시 서버는, 다양한 백엔드 웹 서비스들을 가상화하고 숨기는 REST 웹 서비스들의 세트를 노출시킬 수 있다. 예를 들어, 리버스 프록시 서버는, 신뢰할 수 없는 네트워크들 상의 클라이언트 디바이스들이 기초 백엔드 웹 서비스들을 보거나 또는 이에 대한 지식을 갖지 않을 수 있도록 오로지 가상 URL(uniform resource locator)들만을 노출시킬 수 있다.

[0006] 본원에서 설명되는 추가적인 측면들에 따르면, 클라이언트 디바이스들로부터 수신된 REST 요청들 중 일부 또는 전부를 처리하기 위하여 REST 서비스들 및 REST 자원들이 리버스 프록시 서버 내에서 생성될 수 있다. 이러한 REST 서비스들/자원들이 동적으로 생성될 수 있으며, 리버스 프록시 서버 내에서 REST 요청들을 처리하거나 및/또는 대응하는 백엔드 웹 서비스들의 세트를 호출하도록 구성될 수 있다. REST 웹 서비스들/자원들을 동적으로 생성하고 관리하기 위하여, REST 인프라스트럭처(infrastructure) 및/또는 REST 애플리케이션 엔진이 리버스 프록시 서버 내에 구현될 수 있다. 추가적으로, 일부 실시예들에 있어서, 리버스 프록시 서버 내의 REST 자원들은 백엔드 웹 서비스 호출들을 생성하고 다양한 정책들을 집행하기 위하여 이를 정책 집행 엔진으로 제공할 수 있다.

도면의 간단한 설명

[0007] 도 1은 그 내부에서 본 발명의 다양한 실시예들이 구현될 수 있는 예시적인 분산형 시스템의 컴포넌트들을 예시하는 블록도이다.

도 2는, 본 발명의 실시예들에 의해 제공되는 서비스들이 이에 의해 클라우드 서비스들로서 제공될 수 있는 시스템 환경의 컴포넌트들을 예시하는 블록도이다.

도 3은 그 내부에서 본 발명의 다양한 실시예들이 구현될 수 있는 예시적인 컴퓨터 시스템을 예시하는 블록도이다.

도 4a 및 도 4b는, 본 발명의 하나 이상의 실시예들에 따른, 컴퓨팅 디바이스들 및/또는 시스템들 사이에서 메시지들을 프로세싱하고 송신하기 위한 리버스 프록시 서버를 포함하는 컴퓨팅 환경들의 예들을 고-레벨로 예시하는 블록도들이다.

도 5는 본 발명의 하나 이상의 실시예들에 따른, REST 요청들을 수신하고 프로세싱하기 위한 리버스 프록시 서버를 고-레벨로 예시하는 다른 블록도이다.

도 6a 내지 도 6b는, 본 발명의 하나 이상의 실시예들에 따른, 클라이언트 디바이스들로부터 REST 요청들을 수신하고 프로세싱하며, 백엔드 웹 서비스들을 결정하고 호출하기 위한 프로세스를 예시하는 순서도이다.

도 7은 본 발명의 하나 이상의 실시예들에 따른, 동적으로 REST 서비스들 및 REST 자원들을 생성하기 위한 리버스 프록시 서버를 고-레벨로 예시하는 다른 블록도이다.

도 8은 본 발명의 하나 이상의 실시예들에 따른, 리버스 프록시 서버 상에서 REST 서비스들 및 REST 자원들을 생성하고, 빌드(build)하며, 및 배포(deploy)하기 위한 프로세스를 예시하는 순서도이다.

도 9는, 본 발명의 하나 이상의 실시예들에 따른, 정책 집행 엔진을 포함하는 리버스 프록시 서버를 고-레벨로 예시하는 다른 블록도이다.

도 10은 본 발명의 하나 이상의 실시예들에 따른, 결정된 메시지 프로세싱 정책들을 사용하여 메시지들을 수신하고 프로세싱하기 위한 프로세스를 예시하는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0008] 다음의 설명에 있어서, 예시의 목적들을 위하여, 다수의 특정 세부사항들이 본 발명의 다양한 실시예들의 완전한 이해를 제공하기 위하여 기술된다. 그러나, 본 발명의 실시예들이 이러한 특정 상세내용들 중 일부가 없이 실시될 수 있다는 것이 당업자에게 자명할 것이다. 다른 사례들에 있어, 잘 알려진 구조들 및 디바이스들은 블

록도의 형태로 도시된다.

[0009] 다음의 설명은 오로지 예시적인 실시예들만을 제공하며, 본 개시의 범위, 적용가능성, 또는 구성을 제한하도록 의도되지 않는다. 오히려, 예시적인 실시예들의 다음의 설명은 당업자들에게 예시적인 실시예를 구현하기 위한 사용 가능한 설명을 제공할 것이다. 첨부된 청구항들에 기술되는 바와 같은 본 발명의 사상 및 범위로부터 벗어나지 않고 엘리먼트들의 기능 및 배열에 있어서 다양한 변화들이 이루어질 수 있다는 것이 이해되어야만 한다.

[0010] 다음의 설명에 있어서, 다수의 특정 세부사항들이 실시예들의 완전한 이해를 제공하기 위하여 주어진다. 그러나, 실시예들이 이러한 특정 세부사항들 없이 실시될 수 있다는 것이 당업자에게 이해될 것이다. 예를 들어, 회로들, 시스템들, 네트워크들, 프로세스들, 및 다른 컴포넌트들은 불필요한 세부사항으로 실시예들을 모호하지 않게 하기 위하여 블록도 형태의 컴포넌트들로서 도시될 수 있다. 다른 경우들에 있어서, 잘 알려진 회로들, 프로세스들, 알고리즘들, 구조들, 및 기술들은 실시예들을 모호하게 하는 것을 회피하기 위하여 불필요한 세부사항 없이 도시될 수 있다.

[0011] 또한, 개별적인 실시예들이 순서도, 흐름도, 데이터 흐름도, 구조도, 또는 블록도로서 도시되는 프로세스로서 설명될 수 있다는 것을 주의해야 한다. 순서도가 순차적인 프로세스로서 동작들을 설명할 수 있지만, 동작들 중 다수는 병렬로 또는 동시에 수행될 수 있다. 이에 더하여, 동작들의 순서는 재-배열될 수 있다. 그것의 동작이 완료될 때 프로세스가 종료되지만, 이는 도면에는 포함되지 않은 추가적인 단계들을 가질 수 있다. 프로세스는, 함수, 절차, 서브루틴, 서브프로그램 등에 대응할 수 있다. 프로세스가 함수에 대응할 때, 프로세스의 종료는 호출 함수 또는 메인 함수에 대한 함수의 반환(return)에 대응할 수 있다.

[0012] 용어 "컴퓨터-관독가능 매체"는 비제한적으로, 휴대용 또는 고정형 저장 디바이스들, 광 저장 디바이스들, 및 명령어(들) 및/또는 데이터를 저장, 포함 또는 운반할 수 있는 다양한 다른 매체들과 같은 비-일시적인 매체들을 포함한다. 코드 세그먼트(code segment) 또는 컴퓨터-실행가능 명령어들은, 절차, 함수, 서브프로그램, 프로그램, 루틴, 서브루틴, 모듈, 소프트웨어 패키지, 클래스, 또는 명령어들, 데이터 구조들, 또는 프로그램 명령문들의 임의의 조합을 나타낼 수 있다. 코드 세그먼트는, 정보, 데이터, 인수들, 파라미터들, 또는 메모리 컨텐츠를 전달하거나 및/또는 수신함으로써 다른 코드 세그먼트 또는 하드웨어 회로에 결합될 수 있다. 정보, 인수들, 파라미터들, 데이터 등은, 메모리 공유, 메시지 전달, 토큰 전달, 네트워크 송신 등을 포함하는 임의의 적절한 수단을 통해 전달되거나, 포워딩되거나, 또는 송신될 수 있다.

[0013] 추가로, 실시예들은 하드웨어, 소프트웨어, 펌웨어, 미들웨어, 마이크로코드, 하드웨어 기술 언어, 또는 이들의 임의의 조합에 의해 구현될 수 있다. 소프트웨어, 펌웨어, 미들웨어 또는 마이크로코드로 구현될 때, 필요한 태스크(task)들을 수행하기 위한 프로그램 코드 또는 코드 세그먼트들은 기계 관독가능 매체에 저장될 수 있다. 프로세서(들)는 필요한 태스크들을 수행할 수 있다.

[0014] 컴퓨터 네트워크들 사이에서 송신되는 메시지들을 프로세싱하기 위한 다양한 기술들(예를 들어, 방법들, 시스템들, 하나 이상의 프로세스들에 의해 실행가능한 복수의 명령어들을 저장하는 비-일시적 컴퓨터-관독가능 저장 메모리, 등)이 본원에서 설명된다. 일부 실시예들에 있어서, 백엔드 웹 서비스들, 애플리케이션들, 및 다른 웹 컨텐트에 대한 클라이언트 디바이스들로부터의 요청들과 같은 메시지들이 복수의 컴퓨터 네트워크들 사이에서 송신될 수 있다. 물리적인 또는 논리적인 서브네트워크 내에 구현된 프록시 서버들과 같은 하나 이상의 중간 디바이스들 또는 애플리케이션들이 통신 엔드포인트들 사이에서 메시지들을 수신하고, 프로세싱하며, 송신할 수 있다. 일부 실시예들에 있어서, 리버스 프록시 서버는 리버스 프록시 서버 내에서 동적으로 표현 상태 전환(REST) 서비스들 및 REST 자원들을 생성하도록 구성될 수 있다. 리버스 프록시 서버 내의 REST 서비스들 및 REST 자원들은 클라이언트 디바이스들로부터 들어오는 요청들을 처리하고 백엔드 웹 서비스들을 호출할 수 있으며, 그럼으로써 리버스 프록시 서버 상의 다양한 보안 정책들의 설계 추상화 및/또는 집행을 가능하게 한다.

[0015] 일부 실시예들에 있어서, 프록시 서버는 프록시 서버 내의 REST 웹 서비스에 의해 노출되는 특정 자원들에 대한 웹 서비스 요청을 수신할 수 있다. 프록시 서버의 REST 웹 서비스 내의 적절한 자원이 호출될 수 있으며, 결과적으로 백엔드 웹 서비스를 호출할 수 있다. 일부 경우들에 있어서, 리버스 프록시 서버는, 다양한 백엔드 웹 서비스들을 가상화하고 숨기는 REST 웹 서비스들의 세트를 노출시킬 수 있다. 예를 들어, 리버스 프록시 서버는, 신뢰할 수 없는 네트워크들 상의 클라이언트 디바이스들이 기초 백엔드 웹 서비스들을 보거나 또는 이에 대한 지식을 갖지 않을 수 있도록 오로지 가상 URL들만을 노출시킬 수 있다. 추가적인 측면들에 따르면, 클라이언트 디바이스들로부터 수신된 REST 요청들 중 일부 또는 전부를 처리하기 위하여 REST 서비스들 및 REST 자원들이 리버스 프록시 서버 내에서 생성될 수 있다. 이러한 REST 서비스들/자원들이 동적으로 생성될 수 있으며, 리버스 프록시 서버 내에서 REST 요청들을 처리하거나 및/또는 대응하는 백엔드 웹 서비스들의 세트를 호출

하도록 구성될 수 있다. REST 웹 서비스들/자원들을 동적으로 생성하고 관리하기 위하여, REST 인프라스트럭처 및/또는 REST 애플리케이션 엔진이 리버스 프록시 서버 내에 구현될 수 있다. 추가적으로, 특정 실시예들에 있어서, 리버스 프록시 서버 내의 REST 자원들은 백엔드 웹 서비스 호출들을 생성하고 다양한 정책들을 집행하기 위하여 이를 정책 집행 엔진으로 제공할 수 있다.

[0016] 도 1은 그 내부에서 본 발명의 다양한 실시예들이 구현될 수 있는 예시적인 분산형 시스템의 컴포넌트들을 예시하는 블록도이다. 예시된 실시예에 있어서, 분산형 시스템(100)은, 하나 이상의 네트워크(들)(110)를 통해 웹 브라우저, 사유 클라이언트(proprietary client)(예를 들어, 오라클 폼스(Oracle Forms)), 또는 유사한 것과 같은, 클라이언트 애플리케이션을 실행하고 동작시키도록 구성된 하나 이상의 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및 108)을 포함한다. 서버(112)는 네트워크(110)를 통해 원격 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및 108)과 통신가능하게 결합될 수 있다.

[0017] 다양한 실시예들에 있어서, 서버(112)는 시스템의 하나 이상의 컴포넌트들에 의해 제공되는 하나 이상의 서비스들 또는 소프트웨어 애플리케이션들을 실행하도록 맞춰질 수 있다. 일부 실시예들에 있어서, 이러한 서비스들은 웹-기반으로 또는 클라우드 서비스들로서 또는 서비스형 소프트웨어(Software as a Service; SaaS) 모델 하에서 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및/또는 108)의 사용자들에게 제공될 수 있다. 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및/또는 108)을 조작하는 사용자들은 결과적으로 이러한 컴포넌트들에 의해 제공되는 서비스들을 사용하기 위하여 서버(112)와 상호작용하는 하나 이상의 클라이언트 애플리케이션들을 사용할 수 있다.

[0018] 도면에 도시된 구성에 있어서, 시스템(100)의 소프트웨어 컴포넌트들(118, 120 및 122)은 서버(112) 상에서 구현되는 것으로 도시된다. 다른 실시예들에 있어서, 시스템(100)의 컴포넌트들 및/또는 이러한 컴포넌트들에 의해 제공되는 서비스들 중 하나 이상이 또한 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및/또는 108) 중 하나 이상에 의해 구현될 수 있다. 그러면, 클라이언트 컴퓨팅 디바이스들을 조작하는 사용자들은 이러한 컴포넌트들에 의해 제공되는 서비스들을 사용하기 위하여 하나 이상의 클라이언트 애플리케이션들을 사용할 수 있다. 이러한 컴포넌트들은 하드웨어, 펌웨어, 소프트웨어, 또는 이들의 조합들로 구현될 수 있다. 분산형 시스템(100)과 상이할 수 있는 다양하고 상이한 시스템 구성들이 가능하다는 것이 이해되어야만 한다. 따라서, 도면에 도시된 실시예는 일 실시예의 시스템을 구현하기 위한 분산형 시스템의 일 예이며, 제한하는 것으로 의도되지 않는다.

[0019] 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및/또는 108)은, Microsoft Windows Mobile® 및/또는 iOS, Windows Phone, Android, BlackBerry 10, Palm OS, 및 유사한 것과 같은 다양한 모바일 운영 시스템들과 같은 소프트웨어를 실행하며, 인터넷, e-메일, 단문 메시지 서비스(short message service; SMS), Blackberry®, 또는 다른 통신 프로토콜 가능형(enabled)일 수 있는, 휴대용 핸드헬드 디바이스들(예를 들어, iPhone®, 셀룰러 전화기, iPad®, 컴퓨팅 태블릿, 개인용 정보 단말기(personal digital assistant; PDA)) 또는 웨어러블(wearable) 디바이스들(예를 들어, Google Glass® 머리 착용형 디스플레이(head mounted display))일 수 있다. 클라이언트 컴퓨팅 디바이스들은, 예로서, 다양한 버전들의 Microsoft Windows®, Apple Macintosh®, 및/또는 리눅스 운영 시스템들을 실행하는 개인용 컴퓨터들 및/또는 랩톱 컴퓨터들을 포함하는 범용 개인용 컴퓨터들일 수 있다. 클라이언트 컴퓨팅 디바이스들은, 비제한적으로, 예를 들어, 구글 크롬 OS와 같은 다양한 GNU/리눅스 운영 시스템들을 포함하는, 다양한 상용-이용가능 UNIX® 또는 UNIX-유사 운영 시스템들 중 임의의 것을 실행하는 워크스테이션 컴퓨터들일 수 있다. 대안적으로 또는 추가적으로, 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및 108)은, 네트워크(들)(110)를 통해 통신할 수 있는 임의의 다른 전자 디바이스, 예컨대 씬-클라이언트(thin-client) 컴퓨터, 인터넷-가능형 게이밍 시스템(예를 들어, Kinect® 제스처 입력 디바이스를 갖는 또는 이를 갖지 않는 마이크로소프트 Xbox 게이밍 콘솔), 및/또는 개인용 메시징 디바이스일 수 있다.

[0020] 예시적인 분산형 시스템(100)이 4개의 클라이언트 컴퓨팅 디바이스들과 함께 도시되지만, 임의의 수의 클라이언트 컴퓨팅 디바이스들이 지원될 수 있다. 센서들 등을 갖는 디바이스들과 같은 다른 디바이스들이 서버(112)와 상호작용할 수 있다.

[0021] 분산형 시스템(100)의 네트워크(들)(110)는, 비제한적으로, TCP/IP(transmission control protocol/Internet protocol), SNA(systems network architecture), IPX(Internet packet exchange), 애플 토크(AppleTalk), 및 유사한 것을 포함하는 다양한 상용-이용가능 프로토콜들 중 임의의 것을 사용하여 데이터 통신을 지원할 수 있는 당업자들에게 익숙한 임의의 유형의 네트워크일 수 있다. 단지 예로서, 네트워크(들)(110)는 근거리 네트워크(LAN), 예컨대 이더넷, 토큰-링 및/또는 유사한 것 기반의 근거리 네트워크일 수 있다. 네트워크(들)(110)는

광역 네트워크 및 인터넷일 수 있다. 이는, 비제한적으로 가상 사설 네트워크(virtual private network; VPN)를 포함하는 가상 네트워크, 인트라넷, 엑스트라넷, 공중 교환 전화 네트워크(public switched telephone network; PSTN), 적외선 네트워크, 무선 네트워크(예를 들어, 전기 전자 기술자 협회(Institute of Electrical and Electronics; IEEE) 802.11 프로토콜들의 묶음(suite), Bluetooth®, 및/또는 임의의 다른 무선 프로토콜 중 임의의 것 하에서 동작하는 네트워크); 및/또는 이들 및/또는 다른 네트워크들의 임의의 조합을 포함할 수 있다.

[0022] 서버(112)는, 하나 이상의 범용 컴퓨터들, 특수 서버 컴퓨터들(예로서, PC(personal computer) 서버들, UNIX® 서버들, 중급 서버(mid-range server)들, 메인프레임 컴퓨터들, 랙-장착형 서버들, 등을 포함함), 서버 팜(server farm)들, 서버 클러스터(server cluster)들, 또는 임의의 다른 적절한 배열 및/또는 조합으로 구성될 수 있다. 다양한 실시예들에 있어서, 서버(112)는 이상의 개시에서 설명된 하나 이상의 서비스들 또는 소프트웨어 애플리케이션들을 실행하도록 맞춰질 수 있다. 예를 들어, 서버(112)는 본 개시의 일 실시예에 따른 이상에서 설명된 프로세싱을 수행하기 위한 서버에 대응할 수 있다.

[0023] 서버(112)는, 임의의 상용 이용가능 서버 운영 시스템뿐만 아니라 이상에서 논의된 것들 중 임의의 것을 포함하는 운영 시스템을 실행할 수 있다. 서버(112)는 또한, HTTP(hypertext transport protocol) 서버들, FTP(file transfer protocol) 서버들, CGI(common gateway interface) 서버들, JAVA® 서버들, 데이터베이스 서버들, 및 유사한 것을 포함하는 다양한 추가적인 서버 애플리케이션들 및/또는 중간-계층(mid-tier) 애플리케이션들 중 임의의 것을 실행할 수 있다. 예시적인 데이터베이스 서버들은 비제한적으로, 오라클, 마이크로소프트, 사이베이스, IBM(International Business Machines), 및 유사한 것으로부터 상용적으로 이용가능한 것들을 포함한다.

[0024] 일부 구현예들에 있어서, 서버(112)는, 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및 108)의 사용자들로부터 수신되는 데이터 피드(data feed)들 및/또는 이벤트 업데이트들을 분석하고 통합하기 위한 하나 이상의 애플리케이션들을 포함할 수 있다. 일 예로서, 데이터 피드들 및/또는 이벤트 업데이트들은 비제한적으로, Twitter® 피드들, Facebook® 업데이트들 또는 하나 이상의 제 3 자 정보 소스들 및 연속적인 데이터 스트림들로부터 수신되는 실-시간 업데이트들을 포함할 수 있으며, 이들은 센서 데이터 애플리케이션들, 금융 시세표시기들, 네트워크 성능 측정 툴들(예를 들어, 네트워크 모니터링 및 트래픽 관리 애플리케이션들), 클릭스트림(clickstream) 분석 툴들, 자동차 트래픽 모니터링, 및 유사한 것과 연관된 실-시간 이벤트들을 포함할 수 있다. 서버(112)는 또한 클라이언트 컴퓨팅 디바이스들(102, 104, 106, 및 108)의 하나 이상의 디스플레이 디바이스들을 통해 데이터 피드들 및/또는 실-시간 이벤트들을 디스플레이하기 위한 하나 이상의 애플리케이션을 포함할 수 있다.

[0025] 분산형 시스템(100)은 또한 하나 이상의 데이터베이스들(114 및 116)을 포함할 수 있다. 데이터베이스들(114 및 116)은 다양한 위치들에 존재할 수 있다. 예로서, 데이터베이스들(114 및 116) 중 하나 이상은 서버(112)에 대하여 로컬적인 (및/또는 그 내부에 존재하는) 비-일시적인 저장 매체 상에 존재할 수 있다. 대안적으로, 데이터베이스들(114 및 116)은 서버(112)로부터 원격에 존재할 수 있으며, 네트워크-기반 또는 전용 연결을 통해 서버(112)와 통신할 수 있다. 실시예들의 하나의 세트에 있어서, 데이터베이스들(114 및 116)은 저장-영역 네트워크(storage-area network; SAN) 내에 존재할 수 있다. 유사하게, 서버(112)에 기인하는 기능들을 수행하기 위하여 필요한 임의의 파일들이 적절하게 서버(112) 상에 로컬적으로 및/또는 원격적으로 저장될 수 있다. 실시예들의 하나의 세트에 있어서, 데이터베이스들(114 및 116)은, SQL-포맷형 명령들에 응답하여 데이터를 저장, 업데이트, 및 검색하도록 맞춰진 오라클에 의해 제공되는 데이터베이스들과 같은 관계형 데이터베이스를 포함할 수 있다.

[0026] 도 2는, 본 발명의 실시예들에 의해 제공되는 서비스들이 이에 의해 클라우드 서비스들로서 제공될 수 있는 시스템 환경의 컴포넌트들을 예시하는 블록도이다. 예시된 실시예에 있어서, 시스템 환경(200)은, 클라우드 서비스들을 제공하는 클라우드 인프라스트럭처 시스템(202)과 상호작용하기 위해 사용자들에 의해 사용될 수 있는 하나 이상의 클라이언트 컴퓨팅 디바이스들(204, 206, 및 208)을 포함한다. 클라이언트 컴퓨팅 디바이스들은, 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들을 사용하기 위하여 클라이언트 컴퓨팅 디바이스의 사용자에 의해 클라우드 인프라스트럭처 시스템(202)과 상호작용하기 위해 사용될 수 있는 웹 브라우저, 사유 클라이언트 애플리케이션(예를 들어, 오라클 폼스), 또는 어떤 다른 애플리케이션과 같은 클라이언트 애플리케이션을 동작시키도록 구성될 수 있다.

[0027] 도면에 도시된 클라우드 인프라스트럭처 시스템(202)이 도시된 것들과는 다른 컴포넌트들을 가질 수 있다는 것이 이해되어야만 한다. 추가로, 도면에 도시된 실시예는 본 발명의 일 실시예를 통합할 수 있는 클라우드 인프

라스트러처 시스템의 단지 하나의 예이다. 일부 다른 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)은 도면에 도시된 것보다 더 많거나 또는 더 적은 컴포넌트들을 가질 수 있거나, 2개 이상의 컴포넌트들을 결합할 수 있거나, 또는 컴포넌트들의 상이한 구성 또는 배열을 가질 수 있다.

[0028] 클라이언트 컴퓨팅 디바이스들(204, 206, 및 208)은 102, 104, 106, 및 108에 대하여 이상에서 설명된 것들과 유사한 디바이스들일 수 있다.

[0029] 예시적인 시스템 환경(200)이 3개의 클라이언트 컴퓨팅 디바이스들과 함께 도시되지만, 임의의 수의 클라이언트 컴퓨팅 디바이스들이 지원될 수 있다. 센서들 등을 갖는 디바이스들과 같은 다른 디바이스들이 클라우드 인프라스트럭처 시스템(202)과 상호작용할 수 있다.

[0030] 네트워크(들)(210)는 클라이언트들(204, 206, 및 208)과 클라우드 인프라스트럭처 시스템(202) 사이의 데이터의 교환 및 통신을 가능하게 할 수 있다. 각각의 네트워크는, 네트워크(들)(110)에 대하여 이상에서 설명된 것들을 포함하여, 다양한 상용-이용가능 프로토콜들 중 임의의 프로토콜을 사용하여 데이터 통신을 지원할 수 있는 당업자들에게 익숙한 임의의 유형의 네트워크일 수 있다.

[0031] 클라우드 인프라스트럭처 시스템(202)은 서버(112)에 대하여 이상에서 설명된 것들을 포함할 수 있는 하나 이상의 서버들 및/또는 컴퓨터들을 포함할 수 있다.

[0032] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템에 의해 제공되는 서비스들은, 온 디맨드(on demand)식으로 클라우드 인프라스트럭처 시스템의 사용자들이 이용할 수 있게 되는 서비스들, 예컨대 온라인 데이터 저장 및 백업(backup) 솔루션들, 웹-기반-이메일 서비스들, 호스팅형(hosted) 오피스 제품군(suite)들 및 문서 협업 서비스들, 데이터베이스 프로세싱, 관리형 기술 지원 서비스들, 및 유사한 것의 호스트를 포함할 수 있다. 클라우드 인프라스트럭처 시스템에 의해 제공되는 서비스들은 그 사용자들의 요구들을 충족시키기 위하여 동적으로 스케일링(scale)할 수 있다. 클라우드 인프라스트럭처 시스템에 의해 제공되는 서비스의 특정 인스턴스 생성(instantiation)은 본원에서 "서비스 인스턴스"로서 지칭된다. 일반적으로, 클라우드 서비스 제공자의 시스템으로부터 인터넷과 같은 통신 네트워크를 통해 사용자가 이용할 수 있게 된 임의의 서비스는 "클라우드 서비스"로서 지칭된다. 전형적으로, 공중 클라우드 환경에 있어서, 클라우드 서비스 제공자의 시스템을 구성하는 서버들 및 시스템들은 고객의 자체적인 사내(on-premise) 서버들 및 시스템들과는 상이하다. 예를 들어, 클라우드 서비스 제공자의 시스템은 애플리케이션을 호스팅할 수 있으며, 사용자는, 인터넷과 같은 통신 네트워크를 통해 애플리케이션을 온 디맨드식으로, 주문 및 사용할 수 있다.

[0033] 일부 예들에 있어서, 컴퓨터 네트워크 클라우드 인프라스트럭처 내의 서비스는, 클라우드 판매사에 의해 사용자에게 제공되는 저장소, 호스팅형 데이터베이스, 호스팅형 웹 서버, 소프트웨어 애플리케이션, 또는 다른 서비스, 또는 달리 당업계에 알려진 바와 같은 서비스에 대한 보호형(protected) 컴퓨터 네트워크 액세스를 포함할 수 있다. 예를 들어, 서비스는 인터넷을 통한 클라우드 상의 원격 저장소에 대한 패스워드-보호형 액세스를 포함할 수 있다. 다른 예로서, 서비스는, 네트워크화된 개발자들에 의한 사적인 사용(private use)을 위한 웹 서비스-기반 호스팅형 관계형 데이터베이스 및 스크립트-언어 미들웨어 엔진을 포함할 수 있다. 또 다른 예로서, 서비스는 클라우드 판매사의 웹 사이트 상에서 호스팅되는 이메일 소프트웨어 애플리케이션에 대한 액세스를 포함할 수 있다.

[0034] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)은, 셀프-서비스(self-service)의, 가입-기반(subscription-based)의, 탄력적 스케일러블(scalable)의, 신뢰할 수 있는, 고도로 이용가능하며, 안전한 방식으로 고객에게 전달되는 애플리케이션들, 미들웨어, 및 데이터베이스 서비스 제공(offering)들의 묶음을 포함할 수 있다. 이러한 클라우드 인프라스트럭처 시스템의 일 예는 본 양수인에 의해 제공되는 오라클 퍼블릭 클라우드(Oracle Public Cloud)이다.

[0035] 다양한 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)은, 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들에 대한 고객의 가입(subscription)을 자동으로 프로비저닝(provision)하고, 관리하며, 추적하도록 맞춰질 수 있다. 클라우드 인프라스트럭처 시스템(202)은 상이한 배포(deployment) 모델들을 통해 클라우드 서비스들을 제공할 수 있다. 예를 들어, 서비스들은, 클라우드 인프라스트럭처 시스템(202)이 클라우드 서비스 판매 조직에 의해 소유되며(예를 들어, 오라클에 의해 소유되며) 서비스들이 일반적인 공중 또는 상이한 산업 기업들에게 이용가능하게 되는 공중 클라우드 모델 하에서 제공될 수 있다. 다른 예로서, 서비스들은, 클라우드 인프라스트럭처 시스템(202)이 오로지 단일 조직을 위해서만 운용되며, 조직 내의 하나 이상의 엔티티(entity)들에 대하여 서비스들을 제공할 수 있는 사설 클라우드 모델 하에서 제공될 수 있다. 클

라우드 서비스들은 또한, 클라우드 인프라스트럭처 시스템(202) 및 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들이 연관된 커뮤니티(community) 내의 몇몇 조직들에 의해 공유되는 커뮤니티 클라우드 모델 하에서 제공될 수 있다. 클라우드 서비스들은 또한 2개 이상의 상이한 모델들의 조합인 하이브리드(hybrid) 클라우드 모델 하에서 제공될 수 있다.

[0036] 일부 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들은, 서비스형 소프트웨어(SaaS) 카테고리, 서비스형 플랫폼(Platform as a Service; PaaS) 카테고리, 서비스형 인프라스트럭처(Infrastructure as a Service; IaaS) 카테고리, 또는 하이브리드 서비스들을 포함하는 서비스들의 다른 카테고리들 하에서 제공되는 하나 이상의 서비스들을 포함할 수 있다. 가입 주문을 통해, 고객은 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 하나 이상의 서비스들을 주문할 수 있다. 그러면, 클라우드 인프라스트럭처 시스템(202)은 고객의 가입 주문의 서비스들을 제공하기 위하여 프로세싱을 수행한다.

[0037] 일부 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들은, 비제한적으로, 애플리케이션 서비스들, 플랫폼 서비스들 및 인프라스트럭처 서비스들을 포함할 수 있다. 일부 예들에 있어서, 애플리케이션 서비스들은 클라우드 인프라스트럭처 시스템에 의해 SaaS 플랫폼을 통해 제공될 수 있다. SaaS 플랫폼은 SaaS 카테고리에 속하는 클라우드 서비스들을 제공하도록 구성될 수 있다. 예를 들어, SaaS 플랫폼은 통합형 개발 및 배포 플랫폼 상에 온-디맨드 애플리케이션들의 묶음을 구축하고 전달하기 위한 능력을 제공할 수 있다. SaaS 플랫폼은 SaaS 서비스들을 제공하기 위한 기초 소프트웨어 및 인프라스트럭처를 관리 및 제어할 수 있다. SaaS 플랫폼에 의해 제공되는 서비스들을 이용함으로써, 고객들은 클라우드 인프라스트럭처 시스템 상에서 실행되는 애플리케이션들을 이용할 수 있다. 고객들이 별개의 라이센스들 및 지원을 구매할 필요 없이, 고객들이 애플리케이션 서비스들을 획득할 수 있다. 다양하고 상이한 SaaS 서비스들이 제공될 수 있다. 예들은, 비제한적으로, 판매 실적 관리, 기업 통합, 더 큰 조직들을 위한 비지니스 유연성을 위한 솔루션들을 제공하는 서비스들을 포함한다.

[0038] 일부 실시예들에 있어서, 플랫폼 서비스들은 클라우드 인프라스트럭처 시스템에 의해 PaaS 플랫폼을 통해 제공될 수 있다. PaaS 플랫폼은 PaaS 카테고리에 속하는 클라우드 서비스들을 제공하도록 구성될 수 있다. 플랫폼 서비스들의 예들은, 비제한적으로, 조직들(예컨대 오라클)이 기존의 애플리케이션들을 공유되는 공통 아키텍처 상에 통합하는 것을 가능하게 하는 서비스들뿐만 아니라, 플랫폼에 의해 제공되는 공유되는 서비스들을 레버리징(leverage)하는 새로운 애플리케이션들을 구축하기 위한 능력을 포함할 수 있다. PaaS 플랫폼은 PaaS 서비스들을 제공하기 위한 기초 소프트웨어 및 인프라스트럭처를 관리 및 제어할 수 있다. 고객들이 별개의 라이센스들 및 지원을 구매할 필요 없이, 고객들이 클라우드 인프라스트럭처 시스템에 의해 제공되는 PaaS 서비스들을 획득할 수 있다. 플랫폼 서비스들의 예들은, 비제한적으로, 오라클 자바 클라우드 서비스(Java Cloud Service; JCS), 오라클 데이터베이스 클라우드 서비스(Database Cloud Service; DBCS), 및 다른 것들을 포함한다.

[0039] PaaS 플랫폼에 의해 제공되는 서비스들을 이용함으로써, 고객들은 클라우드 인프라스트럭처 시스템에 의해 지원되는 프로그래밍 언어들 및 툴들을 이용할 수 있으며, 또한 배포된 서비스들을 제어할 수 있다. 일부 실시예들에 있어서, 클라우드 인프라스트럭처 시스템에 의해 제공되는 플랫폼 서비스들은, 데이터베이스 클라우드 서비스들, 미들웨어 클라우드 서비스들(예를 들어, 오라클 퓨전 미들웨어(Fusion Middleware) 서비스들), 및 자바 클라우드 서비스들을 포함할 수 있다. 일 실시예에 있어서, 데이터베이스 클라우드 서비스들은, 조직들이 데이터베이스 자원들을 풀링(pool)하고 고객들에게 데이터베이스 클라우드의 형태로 서비스형 데이터베이스를 제공하는 것을 가능하게 하는 공유형 서비스 배포 모델들을 지원할 수 있다. 클라우드 인프라스트럭처 시스템 내에서, 미들웨어 클라우드 서비스들은 고객들이 다양한 비지니스 애플리케이션들을 개발 및 배포하기 위한 플랫폼을 제공할 수 있으며, 자바 클라우드 서비스들은 고객들이 자바 애플리케이션들을 배포하기 위한 플랫폼을 제공할 수 있다.

[0040] 다양하고 상이한 인프라스트럭처 서비스들이 클라우드 인프라스트럭처 시스템 내에서 IaaS 플랫폼에 의해 제공될 수 있다. 인프라스트럭처 서비스들은, SaaS 플랫폼 및 PaaS 플랫폼에 의해 제공되는 서비스들을 사용하는 고객들을 위한 저장소, 네트워크들, 및 다른 기본 컴퓨팅 자원들과 같은 기초 컴퓨팅 자원들의 관리 및 제어를 가능하게 한다.

[0041] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)은 또한, 클라우드 인프라스트럭처 시스템의 고객들에게 다양한 서비스들을 제공하기 위하여 사용되는 자원들을 제공하기 위한 인프라스트럭처 자원들(230)을 포함할 수 있다. 일 실시예에 있어서, 인프라스트럭처 자원들(230)은, 서버들, 저장소와 같은 하드웨어, 및 PaaS 플랫폼 및 SaaS 플랫폼에 의해 제공되는 서비스들을 실행하기 위한 네트워킹 자원들의 사전-통합되고 최적

화된 조합들을 포함할 수 있다.

[0042] 일부 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202) 내의 자원들은 다수의 사용자들에 의해 공유될 수 있으며, 요구마다 동적으로 재-할당될 수 있다. 추가적으로, 자원들은 상이한 시간 구간(time zone)들에서 사용자들에게 할당될 수 있다. 예를 들어, 클라우드 인프라스트럭처 시스템(230)은, 제 1 시간 구간에서의 제 1 세트의 사용자들이 지정된 시간 동안 클라우드 인프라스트럭처 시스템의 자원들을 사용하는 것을 가능하게 하고, 그런 다음 상이한 시간 구간에 위치된 다른 세트의 사용자들에게 동일한 자원들의 재-할당을 가능하게 할 수 있으며, 그럼으로써 자원들의 사용을 최대화한다.

[0043] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)의 상이한 컴포넌트들 또는 모듈들에 의해 그리고 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들에 의해 공유되는 복수의 내부 공유형 서비스들(232)이 제공될 수 있다. 이러한 내부 공유형 서비스들은, 비제한적으로, 보안 및 신원(identity) 서비스, 통합 서비스, 기업 저장소 서비스, 기업 관리자 서비스, 바이러스 스캐닝 및 화이트 리스트(white list) 서비스, 고 이용가능성, 백업 및 복원 서비스, 클라우드 지원을 가능하게 하기 위한 서비스, 이메일 서비스, 통지 서비스, 파일 전송 서비스, 및 유사한 것을 포함할 수 있다.

[0044] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(202)은 클라우드 인프라스트럭처 시스템 내에서 클라우드 서비스들(예를 들어, SaaS, PaaS, 및 IaaS 서비스들)의 광범위한 관리를 제공할 수 있다. 일 실시예에 있어서, 클라우드 관리 기능은, 클라우드 인프라스트럭처 시스템(202)에 의해 수신되는 고객의 가입을 프로비저닝, 관리 및 추적, 및 유사한 것을 하기 위한 성능들을 포함할 수 있다.

[0045] 일 실시예에 있어서, 도면에 도시된 바와 같이, 클라우드 관리 기능은, 주문 관리 모듈(220), 주문 편성(order orchestration) 모듈(222), 주문 프로비저닝 모듈(224), 주문 관리 및 모니터링 모듈(226), 및 신원 관리 모듈(228)과 같은 하나 이상의 모듈들에 의해 제공될 수 있다. 이러한 모듈들은, 범용 컴퓨터들, 특수 서버 컴퓨터들, 서버 패드, 서버 클러스터들, 또는 임의의 다른 적절한 배열 및/또는 조합일 수 있는 하나 이상의 컴퓨터들 및/또는 서버들을 포함하거나 또는 이들을 사용하여 제공될 수 있다.

[0046] 예시적인 동작(234)에 있어서, 클라이언트 디바이스(204, 206 또는 208)와 같은 클라이언트 디바이스를 사용하는 고객은, 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 하나 이상의 서비스들을 요청하고 클라우드 인프라스트럭처 시스템(202)과 상호작용할 수 있다. 특정 실시예들에 있어서, 고객은 클라우드 사용자 인터페이스(User Interface; UI), 클라우드 UI(212), 클라우드 UI(214) 및/또는 클라우드 UI(216)를 액세스할 수 있으며, 이러한 UI들을 통해 가입을 주문할 수 있다. 고객이 주문하는 것에 응답하여 클라우드 인프라스트럭처 시스템(202)에 의해 수신되는 주문 정보는, 고객이 가입을 의도하는 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 하나 이상의 서비스들 및 고객을 식별하는 정보를 포함할 수 있다.

[0047] 고객에 의해 주문된 이후에, 주문 정보가 클라우드 UI들(212, 214 및/또는 216)을 통해 수신된다.

[0048] 동작(236)에서, 주문은 주문 데이터베이스(218)에 저장된다. 주문 데이터베이스(218)는 클라우드 인프라스트럭처 시스템(218)에 의해 동작되는 그리고 다른 시스템 엘리먼트들과 관련되어 동작되는 몇몇 데이터베이스들 중 하나일 수 있다.

[0049] 동작(238)에서, 주문 정보가 주문 관리 모듈(220)로 포워딩된다. 일부 경우들에 있어서, 주문 관리 모듈(220)은, 주문을 검증하는 것, 및 검증 시에 주문을 예약(book)하는 것과 같은 주문과 관련된 과금 및 회계 기능들을 수행하도록 구성될 수 있다.

[0050] 동작(240)에서, 주문에 관한 정보가 주문 편성 모듈(222)로 통신된다. 주문 편성 모듈(222)은, 고객에 의해 이루어진 주문에 대한 자원들 및 서비스들의 프로비저닝을 편성하기 위하여 주문 정보를 사용할 수 있다. 일부 경우들에 있어서, 주문 편성 모듈(222)은 가입된 서비스들을 지원하기 위하여 주문 프로비저닝 모듈(224)의 서비스들을 사용하여 자원들의 프로비저닝을 편성할 수 있다.

[0051] 특정 실시예들에 있어서, 주문 편성 모듈(222)은 각각의 주문과 연관된 비지니스 프로세스들의 관리를 가능하게 하며, 주문이 프로비저닝으로 진행되어야만 하는지 여부를 결정하기 위하여 비지니스 로직을 적용한다. 동작(242)에서, 새로운 가입에 대한 주문의 수신 시에, 주문 편성 모듈(222)은 자원들을 할당하고 가입 주문을 이행하기 위해 요구되는 이러한 자원들을 구성하기 위한 요청을 주문 프로비저닝 모듈(224)로 전송한다. 주문 프로비저닝 모듈(224)은 고객에 의해 주문된 서비스들에 대한 자원들의 할당을 가능하게 한다. 주문 프로비저닝 모듈(224)은, 요청된 서비스들을 제공하기 위한 자원들을 프로비저닝하기 위해 사용되는 물리적인 구현 계층과 클

라우드 인프라스트럭처 시스템(200)에 의해 제공되는 클라우드 서비스들 사이에 소정의 레벨의 추상화를 제공한다. 따라서, 주문 편성 모듈(222)은, 예컨대 서비스들 및 자원들이 실제로 즉각적으로(on the fly) 프로비저닝 되는지 또는 사전-프로비저닝되고 요청 시에 할당(allocate)되고/배정(assign)되는지 여부와 같은 구현 세부사항들로부터 분리될 수 있다.

[0052] 동작(244)에서, 일단 서비스들 및 자원들이 프로비저닝되면, 제공되는 서비스의 통지가 클라우드 인프라스트럭처 시스템(202)의 주문 프로비저닝 모듈(224)에 의해 클라이언트 디바이스들(204, 206, 및/또는 208) 상의 고객들에게 전송될 수 있다.

[0053] 동작(246)에서, 고객의 가입 주문은 주문 관리 및 모니터링 모듈(226)에 의해 관리되고 추적될 수 있다. 일부 경우들에 있어서, 주문 관리 및 모니터링 모듈(226)은, 사용되는 저장소의 양, 전송되는 데이터의 양, 사용자들의 수, 및 시스템 사용 시간(system up time)과 시스템 정지 시간(system down time)의 양과 같은 가입 주문의 서비스들에 대한 사용 통계자료를 수집하도록 구성될 수 있다.

[0054] 특정 실시예들에 있어서, 클라우드 인프라스트럭처 시스템(200)은 신원 관리 모듈(228)을 포함할 수 있다. 신원 관리 모듈(228)은, 클라우드 인프라스트럭처 시스템(200) 내에서 액세스 관리 및 인가 서비스들과 같은 신원 서비스들을 제공하도록 구성될 수 있다. 일부 실시예들에 있어서, 신원 관리 모듈(228)은 클라우드 인프라스트럭처 시스템(202)에 의해 제공되는 서비스들을 사용하기를 원하는 고객들에 관한 정보를 제어할 수 있다. 이러한 정보는, 이러한 고객들의 신원들을 인증하는 정보 및 이러한 고객들이 다양한 시스템 자원들(예를 들어, 파일들, 디렉토리들, 애플리케이션들, 통신 포트들, 메모리 세그먼트들, 등)과 관련하여 수행하도록 인가되는 액션들을 설명하는 정보를 포함할 수 있다. 신원 관리 모듈(228)은 또한 각각의 고객에 대한 설명 정보(descriptive information) 및 설명 정보가 어떻게 그리고 누구에 의해 액세스되고 수정될 수 있는지에 대한 관리를 포함할 수 있다.

[0055] 도 3은 그 내부에서 본 발명의 다양한 실시예들이 구현될 수 있는 예시적인 컴퓨터 시스템을 예시하는 블록도이다. 시스템(300)은 이상에서 설명된 컴퓨터 시스템들 중 임의의 것을 구현하기 위하여 사용될 수 있다. 도면에 도시된 바와 같이, 컴퓨터 시스템(300)은 버스 서브시스템(302)을 통해 복수의 주변 서브시스템들과 통신하는 프로세싱 유닛(304)을 포함한다. 이러한 주변 서브시스템들은 프로세싱 가속 유닛(306), I/O 서브시스템(308), 저장 서브시스템(318) 및 통신 서브시스템(324)을 포함할 수 있다. 저장 서브시스템(318)은 유형적인 컴퓨터-판독가능 저장 매체(322) 및 시스템 메모리(310)를 포함한다.

[0056] 버스 서브시스템(302)은 컴퓨터 시스템(300)의 다양한 컴포넌트들 및 서브시스템들이 의도된 바와 같이 서로 통신하는 것을 가능하게 하기 위한 메커니즘을 제공한다. 버스 서브시스템(302)이 단일 버스로서 개략적으로 도시되었지만, 버스 서브시스템의 대안적인 실시예들은 복수의 버스들을 사용할 수 있다. 버스 서브시스템(302)은, 다양한 버스 아키텍처들 중 임의의 것을 사용하는 메모리 버스 또는 메모리 제어기, 주변기기 버스, 및 로컬 버스를 포함하는 몇몇 유형들의 버스 구조들 중 임의의 버스 구조일 수 있다. 예를 들어, 이러한 아키텍처들은, 산업 표준 아키텍처(Industry Standard Architecture; ISA) 버스, 마이크로 채널 아키텍처(Micro Channel Architecture; MCA) 버스, 개량 ISA(Enhanced ISA; EISA) 버스, 비디오 전자공학 표준 위원회(Video Electronics Standards Association; VESA) 로컬 버스, 및 주변 컴포넌트 상호연결(Peripheral Component Interconnect; PCI) 버스를 포함할 수 있으며, 이들은 IEEE P1386.1 표준에 대하여 제조되는 메자닌 버스(Mezzanine bus)로서 구현될 수 있다.

[0057] 하나 이상의 집적 회로들(예를 들어, 통상적인 마이크로프로세서 또는 마이크로제어기)로서 구현될 수 있는 프로세싱 유닛(304)은 컴퓨터 시스템(300)의 동작을 제어한다. 하나 이상의 프로세서들이 프로세싱 유닛(304) 내에 포함될 수 있다. 이러한 프로세서들은 단일 코어 또는 다중코어 프로세서들을 포함할 수 있다. 특정 실시예들에 있어서, 프로세싱 유닛(304)은 각각의 프로세싱 유닛 내에 포함된 단일 또는 다중코어 프로세서들을 갖는 하나 이상의 독립적인 프로세싱 유닛들(332 및/또는 334)로서 구현될 수 있다. 다른 실시예들에 있어서, 프로세싱 유닛(304)은 또한 2개의 듀얼-코어 프로세서들을 단일 칩 내에 통합함으로써 형성된 쿼드-코어 프로세싱 유닛으로서 구현될 수 있다.

[0058] 다양한 실시예들에 있어서, 프로세싱 유닛(304)은 프로그램 코드에 응답하여 다양한 프로그램들을 실행할 수 있으며, 프로그램들 또는 프로세스들의 동시 다중 실행을 유지할 수 있다. 임의의 주어진 시점에, 실행될 프로그램 코드의 전부 또는 일부가 프로세서(들)(304) 내에 및/또는 저장 서브시스템(318) 내에 상주할 수 있다. 적절한 프로그래밍을 통하여, 프로세서(들)(304)는 이상에서 설명된 다양한 기능들을 제공할 수 있다. 컴퓨터 시스템(300)은, 디지털 신호 프로세서(digital signal processor; DSP), 특수-목적 프로세서, 및/또는 유사한 것을

포함할 수 있는 프로세싱 가속 유닛(306)을 추가적으로 포함할 수 있다.

[0059] I/O 서브시스템(308)은 사용자 인터페이스 입력 디바이스들 및 사용자 인터페이스 출력 디바이스들을 포함할 수 있다. 사용자 인터페이스 입력 디바이스들은, 키보드, 포인팅 디바이스들 예컨대 마우스 또는 트랙볼, 터치패드 또는 디스플레이 내에 통합된 터치 스크린, 스크롤 휠, 클릭 휠, 다이얼, 버튼, 스위치, 키패드, 음성 명령 인식 시스템들을 가진 오디오 입력 디바이스들, 마이크들, 및 다른 유형들의 입력 디바이스들을 포함할 수 있다. 사용자 인터페이스 입력 디바이스들은, 예를 들어, 사용자들이 제스처들 및 구두(spoken) 명령들을 사용하는 자연스러운 사용자 인터페이스를 통해, Microsoft Xbox® 360 게임 제어기와 같은, 입력 디바이스를 제어하고 이와 상호작용하는 것을 가능하게 하는 Microsoft Kinect® 모션 센서와 같은 모션 센싱 및/또는 제스처 인식 디바이스들을 포함할 수 있다. 사용자 인터페이스 입력 디바이스들은 또한, 사용자들로부터 눈 움직임(eye activity)(예를 들어, 사진을 찍는 동안의 및/또는 메뉴를 선택하는 동안의 '깜박임')을 검출하고 눈 제스처들을 입력 디바이스(예를 들어, Google Glass®) 내로의 입력으로서 변환하는 Google Glass® 눈 깜박임 검출기와 같은 눈 제스처 인식 디바이스들을 포함할 수 있다. 추가적으로, 사용자 인터페이스 입력 디바이스들은, 사용자들이 음성 명령들을 통하여 음성 인식 시스템(예를 들어, Siri® 네비게이터(navigator))과 상호작용하는 것을 가능하게 하는 음성 인식 센싱 디바이스들을 포함할 수 있다.

[0060] 사용자 인터페이스 입력 디바이스들은 또한, 비제한적으로, 3차원(3D) 마우스들, 조이스틱들 또는 포인팅 스틱들, 게임패드들 및 그래픽 태블릿들, 및 음향/시각 디바이스들 예컨대 스피커들, 디지털 카메라들, 디지털 캠코더들, 휴대용 매체 플레이어들, 웹캠들, 이미지 스캐너들, 평거프린트 스캐너들, 바코드 리더 3D 스캐너들, 3D 프린터들, 레이저 거리계들, 및 시선 추적 디바이스들을 포함할 수 있다. 추가적으로, 사용자 인터페이스 입력 디바이스들은, 예를 들어, 의료 이미징 입력 디바이스들 예컨대 컴퓨터 단층촬영, 자기 공명 이미징, 양전자 방출 단층촬영, 의료 초음파 검사 디바이스들을 포함할 수 있다. 사용자 인터페이스 입력 디바이스들은 또한, 예를 들어, 오디오 입력 디바이스들 예컨대 MIDI 키보드들, 디지털 악기들, 및 유사한 것을 포함할 수 있다.

[0061] 사용자 인터페이스 출력 디바이스들은 디스플레이 서브시스템, 표시등들, 또는 비-시각적 디스플레이들 예컨대 오디오 출력 디바이스들, 등을 포함할 수 있다. 디스플레이 서브시스템은 음극선관(cathode ray tube; CRT), 액정 디스플레이(liquid crystal display; LCD) 또는 플라즈마 디스플레이를 사용하는 것과 같은 평면-패널 디바이스, 프로젝션 디바이스, 터치 스크린, 및 유사한 것일 수 있다. 일반적으로, 용어 "출력 디바이스"의 사용은 컴퓨터 시스템(300)으로부터 사용자 또는 다른 컴퓨터로 정보를 출력하기 위한 모든 가능한 유형들의 디바이스들 및 메커니즘들을 포함하도록 의도된다. 예를 들어, 사용자 인터페이스 출력 디바이스들은, 비제한적으로, 시각적으로 텍스트, 그래픽들 및 오디오/비디오 정보를 전달하는 다양한 디스플레이 디바이스들, 예컨대 모니터들, 프린터들, 스피커들, 헤드폰들, 자동차 네비게이션 시스템들, 플로터(plotter)들, 음성 출력 디바이스들, 및 모뎀들을 포함할 수 있다.

[0062] 컴퓨터 시스템(300)은, 시스템 메모리(310) 내에 현재 위치되어 있는 것으로 도시되는 소프트웨어 엘리먼트들을 포함하는 저장 서브시스템(318)을 포함할 수 있다. 시스템 메모리(310)는, 프로세싱 유닛(304) 상에 로딩 가능하며 실행 가능한 프로그램 명령어들뿐만 아니라 이러한 프로그램들의 실행 동안 생성되는 데이터를 저장할 수 있다.

[0063] 컴퓨터 시스템(300)의 구성 및 유형에 따라서, 시스템 메모리(310)는 (랜덤 액세스 메모리(random access memory; RAM)와 같은) 휘발성일 수 있거나 및/또는 (판독-전용 메모리(read-only memory; ROM), 플래시 메모리, 등과 같은) 비-휘발성일 수 있다. RAM은 전형적으로 프로세싱 유닛(304)에 의해 현재 실행되고 동작되고 있거나 및/또는 이에 즉시 액세스 가능한 데이터 및/또는 프로그램 모듈들을 포함한다. 일부 구현예들에 있어서, 시스템 메모리(310)는 복수의 상이한 유형들의 메모리들, 예컨대 정적 랜덤 액세스 메모리(static random access memory; SRAM) 또는 동적 랜덤 액세스 메모리(dynamic random access memory; DRAM)를 포함할 수 있다. 일부 구현예들에 있어서, 예컨대 기동 동안에 컴퓨터 시스템(300) 내의 엘리먼트들 사이에서 정보를 전송하는 것을 돋는 기본 루틴들을 포함하는 기본 입력/출력 시스템(basic input/output system; BIOS)은 전형적으로 ROM 내에 저장될 수 있다. 예로서 그리고 비제한적으로, 시스템 메모리(310)는 또한, 클라이언트 애플리케이션들, 웹 브라우저들, 중간-계층 애플리케이션들, 관계형 데이터 베이스 관리 시스템(relational database management system; RDBMS)들, 등을 포함할 수 있는 애플리케이션 프로그램들(312), 프로그램 테이터(314), 및 운영 시스템(316)을 예시한다. 예로서, 운영 시스템(316)은, 다양한 버전들의 Microsoft Windows®, Apple Macintosh®, 및/또는 리눅스 운영 시스템들, (비제한적으로 다양한 GNU/리눅스 운영 시스템들, Google Chrome ® OS, 및 유사한 것을 포함하는) 다양한 상용-이용가능 UNIX® 또는 UNIX-유사 운영 시스템들 및/또는 모바일 운영 시스템들 예컨대 iOS, Windows® Phone, Android® OS, BlackBerry® 10 OS, 및 Palm® OS 운영 시스템들

을 포함할 수 있다.

[0064] 저장 서브시스템(318)은 또한 일부 실시예들의 기능을 제공하는 기본 프로그래밍 및 데이터 구성물들을 저장하기 위한 유형의 컴퓨터-판독가능 저장 매체를 제공할 수 있다. 프로세서에 의해 실행될 때 이상에서 설명된 기능을 제공하는 소프트웨어(프로그램들, 코드 모듈들, 명령어들)가 저장 서브시스템(318) 내에 저장될 수 있다. 이러한 소프트웨어 모듈들 또는 명령어들이 프로세싱 유닛(304)에 의해 실행될 수 있다. 저장 서브시스템(318)은 또한 본 발명에 따라 사용되는 데이터를 저장하기 위한 저장소를 제공할 수 있다.

[0065] 저장 서브시스템(300)은 또한, 추가적으로 컴퓨터-판독가능 저장 매체(322)에 연결될 수 있는 컴퓨터-판독가능 저장 매체 리더(320)를 포함할 수 있다. 시스템 메모리(310)와 함께 그리고 선택적으로 이와 조합되어, 컴퓨터-판독가능 저장 매체(322)는, 컴퓨터-판독가능 정보를 일시적으로 및/또는 더 영구적으로 포함하고, 저장하며, 송신하고, 및 검색하기 위한 저장 매체들에 더하여 원격, 로컬, 고정, 및/또는 착탈가능 저장 디바이스들을 포함적으로 나타낼 수 있다.

[0066] 코드, 또는 코드의 부분들을 포함하는 컴퓨터-판독가능 저장 매체(322)는 또한, 비제한적으로, 정보의 저장 및/또는 송신을 위하여 임의의 방법 또는 기술로 구현된 휘발성 및 비-휘발성, 착탈가능 및 비-착탈가능 매체와 같은 저장 매체 및 통신 매체를 포함하는 당업계에서 알려지거나 또는 사용되는 임의의 적절한 매체를 포함할 수 있다. 이는 비-일시적이고 유형의 컴퓨터-판독가능 저장 매체, 예컨대 RAM, ROM, 전기 소거가능 프로그램가능 ROM(electronically erasable programmable ROM; EEPROM), 플래시 메모리 또는 다른 메모리 기술품, CD-ROM, 디지털 다기능 디스크(digital versatile disk; DVD), 또는 광 저장장치, 자기 카세트들, 자기 테이프, 자기 디스크 저장장치 또는 다른 자기 저장 디바이스들, 또는 다른 유형의 컴퓨터 판독가능 매체를 포함할 수 있다. 이는 또한 비유형적인 컴퓨터-판독가능 매체, 예컨대 데이터 신호들, 데이터 송신들, 또는 희망되는 정보를 송신하기 위해 사용될 수 있으며 컴퓨터 시스템(300)에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다.

[0067] 예로서, 컴퓨터-판독가능 저장 매체(322)는, 비-착탈가능 비휘발성 자기 매체로부터 판독하거나 또는 이에 기입하는 하드 디스크 드라이브, 착탈가능 비휘발성 자기 디스크로부터 판독하거나 또는 이에 기입하는 자기 디스크 드라이브, 및 착탈가능 비휘발성 광 디스크 예컨대 CD ROM, DVD, 및 Blu-Ray® 디스크, 또는 다른 광 매체로부터 판독하거나 또는 이에 기입하는 광 디스크 드라이브를 포함할 수 있다. 컴퓨터-판독가능 저장 매체(322)는, 비제한적으로, Zip® 드라이브들, 플래시 메모리 카드들, 범용 직렬 버스(universal serial bus; USB) 플래시 드라이브들, 보안 디지털(secure digital; SD) 카드들, DVD 디스크들, 디지털 비디오 테이프, 및 유사한 것을 포함할 수 있다. 컴퓨터-판독가능 저장 매체(322)는 또한, 비-휘발성 메모리 기반 고체-상태 드라이브(solid-state drive; SSD)들 예컨대 플래시-메모리 기반 SSD들, 기업 플래시 드라이브들, 고체 상태 ROM, 및 유사한 것, 휘발성 메모리 기반 SSD들 예컨대 고체 상태 RAM, 동적 RAM, 정적 RAM, DRAM-기반 SSD들, 자기저항성 RAM(magnetoresistive RAM; MRAM) SSD들, 및 DRAM 및 플래시 메모리 기반 SSD들의 조합을 사용하는 하이브리드 SSD들을 포함할 수 있다. 디스크 드라이브들 및 그들의 연관된 컴퓨터-판독가능 매체는 컴퓨터-판독가능 명령어들, 데이터 구조들, 프로그램 모듈들, 및 컴퓨터 시스템(300)에 대한 다른 데이터의 비-휘발성 저장을 제공할 수 있다.

[0068] 통신 서브시스템(324)은 다른 컴퓨터 시스템들 및 네트워크들에 대한 인터페이스를 제공한다. 통신 서브시스템(324)은 컴퓨터 시스템(300)으로부터 다른 시스템들로 데이터를 송신하고 이로부터 데이터를 수신하기 위한 인터페이스로서 역할한다. 예를 들어, 통신 서브시스템(324)은 컴퓨터 시스템(300)이 인터넷을 통해 하나 이상의 디바이스들에 연결하는 것을 가능하게 할 수 있다. 일부 실시예들에 있어서, 통신 서브시스템(324)은 (예를 들어, 셀룰러 전화기 기술, 진보된 데이터 네트워크 기술, 예컨대 3G, 4G 또는 EDGE(enhanced data rates for global evolution), WiFi(IEEE 802.11 패밀리 표준들, 또는 다른 모바일 통신 기술들, 또는 이들의 임의의 조합)을 사용하여) 무선 음성 및/또는 데이터 네트워크들에 액세스하기 위한 라디오 주파수(radio frequency; RF) 트랜시버 컴포넌트들, 위성 위치확인 시스템(global positioning system; GPS) 수신기 컴포넌트들, 및/또는 다른 컴포넌트들을 포함할 수 있다. 일부 실시예들에 있어서, 통신 서브시스템(324)은 무선 인터페이스에 더하여 또는 그 대신에 유선 네트워크 연결(예를 들어, 이더넷)을 제공할 수 있다.

[0069] 일부 실시예들에 있어서, 통신 서브시스템(324)은 또한 컴퓨터 시스템(300)을 사용할 수 있는 하나 이상의 사용자들을 대표하여 구조화된 및/또는 구조화되지 않은 데이터 피드들(326), 이벤트 스트림들(328), 이벤트 업데이트들(330), 및 그와 유사한 것의 형태로 입력 통신을 수신할 수 있다.

[0070] 예로서, 통신 서브시스템(324)은 소셜 네트워크들 및/또는 다른 통신 서비스들의 사용자들로부터의 실-시간 데이터 피드들(326) 예컨대 Twitter® 피드들, Facebook® 업데이트들, 웹 피드들 예컨대 리치 사이트 서머리

(Rich Site Summary; RSS) 피드들, 및/또는 하나 이상의 제 3 자 정보 소스들로부터의 실-시간 업데이트들을 수신하도록 구성될 수 있다.

[0071] 추가적으로, 통신 서브시스템(324)은 또한, 사실상 명시적인 종료를 갖지 않는 제한이 없거나 또는 연속적일 수 있는 실-시간 이벤트들 및/또는 이벤트 업데이트들(330)의 이벤트 스트림들(328)을 포함할 수 있는 연속적인 데이터 스트림들의 형태로 데이터를 수신하도록 구성될 수 있다. 연속적인 데이터를 생성하는 애플리케이션들의 예들은, 예를 들어, 센서 데이터 애플리케이션들, 금융 시세표시기들, 네트워크 성능 측정 툴들(예를 들어, 네트워크 모니터링 및 트래픽 관리 애플리케이션들), 클릭스트림 분석 툴들, 자동차 트래픽 모니터링, 및 유사한 것을 포함할 수 있다.

[0072] 통신 서브시스템(324)은 또한, 구조화되거나 및/또는 구조화되지 않은 데이터 피드들(326), 이벤트 스트림들(328), 이벤트 업데이트들(330), 및 유사한 것을 컴퓨터 시스템(300)에 결합된 하나 이상의 스트리밍 데이터 소스 컴퓨터들과 통신할 수 있는 하나 이상의 데이터베이스들로 출력하도록 구성될 수 있다.

[0073] 컴퓨터 시스템(300)은, 핸드헬드 휴대용 디바이스(예를 들어, iPhone® 셀룰러 폰, iPad® 컴퓨팅 태블릿, PDA), 웨어러블 디바이스(예를 들어, Google Glass® 머리 착용형 디스플레이), PC, 워크스테이션, 메인프레임, 키오스크, 서버 랙, 또는 임의의 다른 데이터 프로세싱 시스템을 포함하는 다양한 유형들 중 임의의 유형일 수 있다.

[0074] 컴퓨터들 및 네트워크들의 계속해서 변화하는 성질에 기인하여, 도면에 도시된 컴퓨터 시스템(300)의 설명은 오로지 특정한 일 예로서만 의도된다. 도면에 도시된 시스템보다 더 많거나 또는 더 적은 컴포넌트들을 갖는 다수의 다른 구성들이 가능하다. 예를 들어, 커스텀화된 하드웨어가 또한 사용될 수 있거나 및/또는 특정 엘리먼트들이 하드웨어, 펌웨어, (애플릿(applet)들을 포함하는) 소프트웨어, 또는 이들의 조합으로 구현될 수 있다. 추가로, 네트워크 입력/출력 디바이스들과 같은 다른 컴퓨팅 디바이스들에 대한 연결이 이용될 수 있다. 본원에 제공되는 개시 및 교시들에 기초하여, 당업자는 다양한 실시예들을 구현하기 위한 다른 방식들 및/또는 방법들을 인식할 것이다.

[0075] 이상에서 소개된 바와 같이, 본 발명의 실시예들은 컴퓨터 네트워크들 사이에서 송신되는 메시지들을 프로세싱하기 위한 다양한 기술들을 제공한다. 더 구체적으로, 물리적인 또는 논리적인 서브네트워크 내에 구현된 프록시 서버들과 같은 중간 디바이스들 또는 애플리케이션들이 통신 엔드포인트들 사이에서 메시지들을 수신하고, 프로세싱하며, 송신할 수 있다. 일부 실시예들에 있어서, 리버스 프록시 서버는 리버스 프록시 서버 내에서 동적으로 표현 상태 전환(REST) 서비스들 및 REST 자원들을 생성하도록 구성될 수 있다. 리버스 프록시 서버 내의 REST 서비스들 및 REST 자원들은 클라이언트 디바이스들로부터 들어오는 요청들을 처리하고 백엔드 웹 서비스들을 호출할 수 있으며, 그럼으로써 리버스 프록시 서버 상의 다양한 보안 정책들의 설계 추상화 및/또는 집행을 가능하게 한다.

[0076] 다양한 실시예들에 있어서, 프록시 서버는 프록시 서버 내의 REST 웹 서비스에 의해 노출되는 특정 자원들에 대한 웹 서비스 요청을 수신할 수 있다. 프록시 서버의 REST 웹 서비스 내의 적절한 자원이 호출될 수 있으며, 결과적으로 백엔드 웹 서비스를 호출할 수 있다. 일부 경우들에 있어서, 리버스 프록시 서버는, 다양한 백엔드 웹 서비스들을 가상화하고 숨기는 REST 웹 서비스들의 세트를 노출시킬 수 있다. 예를 들어, 리버스 프록시 서버는, 신뢰할 수 없는 네트워크들 상의 클라이언트 디바이스들이 기초 백엔드 웹 서비스들을 보거나 또는 이에 대한 지식을 갖지 않을 수 있도록 오로지 가상 URL들만을 노출시킬 수 있다. 추가적인 측면들에 따르면, 클라이언트 디바이스들로부터 수신된 REST 요청들 중 일부 또는 전부를 처리하기 위하여 REST 서비스들 및 REST 자원들이 리버스 프록시 서버 내에서 생성될 수 있다. 이러한 REST 서비스들/자원들이 동적으로 생성될 수 있으며, 리버스 프록시 서버 내에서 REST 요청들을 처리하거나 및/또는 대응하는 백엔드 웹 서비스들의 세트를 호출하도록 구성될 수 있다. REST 웹 서비스들/자원들을 동적으로 생성하고 관리하기 위하여, REST 인프라스트럭처 및/또는 REST 애플리케이션 엔진이 리버스 프록시 서버 내에 구현될 수 있다. 추가적으로, 특정 실시예들에 있어서, 리버스 프록시 서버 내의 REST 자원들은 백엔드 웹 서비스 호출들을 생성하고 다양한 정책들을 집행하기 위하여 이를 정책 집행 엔진으로 제공할 수 있다.

[0077] 도 4a 및 도 4b는, 다양한 컴퓨터 네트워크들 내의 컴퓨팅 디바이스들 및/또는 시스템들 사이에서 메시지들을 프로세싱하고 송신하기 위한 리버스 프록시 서버(420)를 포함하는 컴퓨팅 환경들(400a 및 400b)의 컴포넌트들을 예시하는 블록도들이다. 이러한 예에 예시된 컴퓨팅 환경들(400a 및 400b)(집합적으로, 400)은, 다양한 클라이언트 디바이스들(410)에게 웹 애플리케이션들 및 웹 서비스들(430)과 같은 백엔드 컴퓨팅 자원들에 대한 액세스를 제공하도록 설계된 고-레벨 컴퓨터아키텍처에 대응할 수 있다. 다양한 실시예들에 있어서, 컴퓨팅 환경(40)

0)은, 다양한 조직들의 컴퓨팅 요구들을 지원하기 위하여 소형의 단순한 컴퓨팅 시스템으로부터 이러한 다른 시스템들과 통합되도록 설계된 하드웨어, 소프트웨어, 및 네트워크 컴퓨팅 환경(400)은 대형의 고도로 복잡한 시스템에 이르는 범위일 수 있다. 컴퓨팅 환경(400)은 다-계층 컴퓨터 아키텍처로서 구현될 수 있으며, 이는 웹-기반 및/또는 클라우드-기반 구현예들을 포함할 수 있고, 그 내부에서 다양한 엔드포인트 디바이스들(예를 들어, 사용자 디바이스들(410), 웹 서비스 제공자들(430), 등)이 하나 이상의 중간-계층 시스템들을 통해 상호작용한다. 추가적으로, 컴퓨팅 환경(400) 내에 도시된 각각의 컴퓨팅 환경(400)은, 하드웨어, 소프트웨어, 및/또는 네트워크 컴퓨팅 환경(400) 내에 도시된 복수의 컴퓨팅 환경(400)들은 결합된 컴퓨팅 시스템들 내에서 동작하는 논리적인 서브컴퓨팅 환경(400)으로서 구현될 수 있다. 다른 경우들에 있어서, 컴퓨팅 환경(400) 내에 도시된 복수의 컴퓨팅 환경(400)들은 결합된 컴퓨팅 시스템들 내에서 동작하는 논리적인 서브컴퓨팅 환경(400)으로서 구현될 수 있다.

[0078] 도 4a 및 도 4b에 도시된 바와 같이, 이러한 컴퓨팅 환경들(400)은, 그 내부에서 클라이언트 디바이스들(410)이 하나 이상의 컴퓨터 네트워크(들)(415), 하나 이상의 방화벽(들)(435), 리버스 프록시 서버(420), 및/또는 다른 중간 네트워크 디바이스들을 통해 하나 이상의 백엔드 웹 서비스들(430)로 요청들을 송신할 수 있는 클라이언트-서버 시스템에 대응할 수 있다. 웹 서비스들(430)은, 비제한적으로, 단순 객체 접근 프로토콜(Simple Object Access protocol; SOAP) 웹 서비스들 또는 API들, 표현 상태 전환(Representational State Transfer; REST) 웹 서비스들 또는 API들, 및/또는 하이퍼 텍스트 전송 프로토콜(Hypertext Transfer Protocol; HTTP) 또는 HTTP 보안 프로토콜들을 통해 노출되는 웹 컨텐트를 포함하는, 다양한 시스템들(430)에 의해 노출되는 임의의 애플리케이션 프로그래밍 인터페이스(application programming interface; API)들, 서비스들, 애플리케이션들, 및 임의의 다른 정보 자산들을 포함할 수 있다. 이러한 경우들에 있어서, 리버스 프록시 서버(420)는 클라이언트 디바이스들(410)과 백엔드 웹 서비스들(430) 사이에 보안 계층을 제공할 수 있다. 예를 들어, 프록시 서버(420)는, 백엔드 웹 서비스들(430)과 연관된 다양한 보안 및 관리 정책들의 집행 및 서비스 가상화와 함께, 백엔드 웹 서비스들(430)에 대한 중심 액세스 포인트를 제공할 수 있다. 리버스 프록시 서버(420)는 또한 이러한 서비스들(430)을 가상화하고 가리면서 백엔드 웹 서비스들(430)을 노출시킬 수 있다. 예를 들어, 리버스 프록시 서버(420)는, 신뢰할 수 없는 네트워크들 상의 클라이언트 디바이스들(410)이 기초 백엔드 웹 서비스들(430)을 보거나 또는 이에 대한 지식을 갖지 않을 수 있도록 오로지 가상 URL들만을 노출시킬 수 있다.

[0079] 추가적으로 또는 대안적으로, 컴퓨팅 환경들(400)은 반대 방향으로 요청들-응답들을 프로세싱하고 송신하기 위한 클라이언트-서버 시스템으로서 구성될 수 있다. 예를 들어, 일부 실시예들에 있어서, 내부 컴퓨터 네트워크(460) 내에서 동작하는 하나 이상의 클라이언트 디바이스들(미도시)이 프록시 서버(420) 및 방화벽(들)(435a) 너머의 다양한 외부 컴퓨터 시스템들 및 네트워크들(480) 상에서 동작하는 웹 서비스들 또는 애플리케이션들(미도시)로 요청들을 송신할 수 있다. 따라서, 서버(420)가 본원에서 리버스 프록시 서버(420)로서 지정되지만, 이는 또한 포워드(forward) 프록시 서버로서 역할할 수 있으며, 내부 클라이언트 디바이스들(440)과 외부 백엔드 웹 서비스들 또는 애플리케이션들 사이에 보안 계층을 제공할 수 있다는 것이 이해되어야만 한다. 리버스 프록시(즉, 리버스 프록시 모드)의 기능 또는 포워드 프록시(즉, 포워드 프록시 모드)의 기능 중 하나를 수행할 때, 리버스 프록시 서버(420)는 SOAP 웹 서비스들, REST 웹 서비스들, HTTP/HTTPS 웹 컨텐트, 및 유사한 것으로의 네트워크 요청들 및 이들로부터의 네트워크 응답들을 처리할 수 있다. 프록시 서버(420)가 포워드 프록시 서버로서 동작할 때, 내부 클라이언트 디바이스들은 외부 백엔드 웹 서비스들/애플리케이션들에 관해 이미 알고 있을 수 있으며, 이러한 백엔드 서비스들/애플리케이션들은 클라이언트 측 상에서 구성된 직접 송신들을 프록시 서버(420)로부터 수신할 수 있다. 이러한 경우들에 있어서, 프록시 서버(420)는 임의의 보안 또는 통신 관리 정책들을 사용하여 포워드 프록시 URI(uniform resource identifier) 엔드포인트들에 대한 보안성을 제공할 수 있다. 포워드 프록시 모드 또는 리버스 프록시 모드 각각에 있어서, 프록시 서버(420)는 커베로스 케이이니트-기반 인증(Kerberos Kinit-based authentication), 커베로스 퍼케이이니트-기반 인증(Kerberos Pkinit-based authentication), 인증 프로토콜 버전 2.0에 대한 개방 표준(open standard for authorization protocol version 2.0; OAuth2) 기반 인증, TLP-기반 인증, 심플 앤 프로텍티드 GSSAPI 교섭 메커니즘(Simple and Protected GSSAPI Negotiation Mechanism; SPNEGO) 토큰들, 윈도우즈 NT 랜 관리자(WINDOWS NT LAN Manager; NTLM) 토큰들, 보안 어써션 마크업 언어(Security Assertion Markup Language; SAML) 토큰들, 및 유사한 것을 사용하는 백엔드 서비스들의 세션 토큰 생성 및/또는 도전-기반 인증과 같은 다양한 보안 및 인증 특징들을 지원할 수 있다.

[0080] 클라이언트 디바이스들(410)은, 도 1 내지 도 3의 예시적인 컴퓨팅 시스템들의 이상에서 논의된 하드웨어, 소프트웨어, 및 네트워킹 컴퓨팅 환경(400)들 중 일부 또는 전부를 포함하는, 데스크탑 또는 랩톱 컴퓨터들, 모바일 디바이스들, 및 다른 다양한 컴퓨팅 디바이스들/시스템들을 포함할 수 있다. 일부 실시예들에 있어서, 클라이언트 디

바이스들(410)은 백엔드 웹 서비스들(430)로 데이터를 요청하고 이들로부터 데이터를 수신하도록 구성된 하나 이상의 클라이언트 소프트웨어 애플리케이션들(예를 들어, 웹 브라우저들)을 포함할 수 있다. 클라이언트 디바이스들(410)은 또한, 네트워크 인터페이스들, 보안 및 인증 성능들, 및 라이브(live) 컨텐트를 수신하고 이를 실시간으로(또는 거의 실시간으로) 사용자들에게 제공하기 위한 컨텐트 캐싱(caching) 성능들을 수립하기 위해 필요한 하드웨어 및 소프트웨어 컴포넌트들을 포함할 수 있다.

[0081] 통신 네트워크들(415)은 본원에서 설명된 컴퓨터 네트워크들 및 다른 통신 네트워크들의 임의의 조합을 포함할 수 있다. 예를 들어, 네트워크들(415)은 근거리 네트워크(local area network; LAN)들, 광역 네트워크(wide area network; WAN)들(예를 들어, 인터넷), 및 다양한 무선 전기통신 네트워크들과 같은 전송 제어 프로토콜/인터넷 프로토콜(transmission control protocol/Internet protocol; TCP/IP) 네트워크들을 포함할 수 있다. 추가적으로, 통신 네트워크(415)가 클라이언트 디바이스들(410)을 백엔드 웹 서비스들(430)로부터 분리하는 다수의 상이한 물리적 및 논리적 네트워크들의 조합들을 나타낼 수 있다는 것이 이해되어야만 한다. 하나 이상의 방화벽들(435)에 더하여, 웹 서버들, 인증 서버들과 같은 다양한 서버들, 및/또는 방화벽들, 라우터들, 게이트웨이들, 로드 밸런서(load balancer)들, 및 유사한 것과 같은 특수 네트워킹 컴포넌트들이 클라이언트 디바이스들(410)과 백엔드 웹 서비스들(430) 사이의 통신을 가능하게 할 수 있다.

[0082] 일부 실시예들에 있어서, 리버스 프록시 서버(420)는 도 4b에 도시된 바와 같은 분리된 DMZ(470) 내의 분리된 컴퓨터 시스템(예를 들어, 프록시 컴퓨터 서버)로서 구현될 수 있거나 또는 특수 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들을 포함하는 컴퓨터 다중 컴퓨팅 시스템들의 조합으로서 구현될 수 있다. 대안적으로 또는 추가적으로, 리버스 프록시 서버(420)는 신뢰되는 네트워크(460) 내의 컴퓨터 서버 또는 네트워크 디바이스(예를 들어, 웹 서버 또는 방화벽(435b)) 내에서 실행되는 프록시 서버 소프트웨어 애플리케이션일 수 있다. 따라서, 리버스 프록시 서버(420)는 도 4a에 도시된 바와 같이 내부 컴퓨터 네트워크(460)의 물리적인 또는 논리적인 서브네트워크(465) 내에 존재할 수 있다. 어느 경우에든, 리버스 프록시 서버(420)는 신뢰되는 내부 네트워크(460) 상의 클라이언트들/서버들과 신뢰되지 않는 외부 네트워크들(480) 상의 클라이언트들/서버들 사이의 매개체로서 역할할 수 있다. 추가적으로, 리버스 프록시 서버(420) 내의 컴포넌트들(421-423)(뿐만 아니라 도 5, 도 7, 및 도 9를 참조하여 논의되는 다른 프록시 서버 컴포넌트들)의 각각은 리버스 프록시 서버(420)와 통신하도록 구성된 별개의 컴퓨팅 시스템들로서 구현될 수 있거나, 또는 리버스 프록시 서버(420)와 동일한 컴퓨터 서버 내에 통합된 논리적인 서브컴포넌트들로서 동작할 수 있다. 어느 경우에든, 각각의 컴포넌트(421-423)(뿐만 아니라 도 5, 도 7, 및 도 9를 참조하여 논의되는 다른 프록시 서버 컴포넌트들)은 본원에서 설명되는 기술들을 수행하기 위한 특수 하드웨어, 소프트웨어, 네트워크, 및 메모리 서브 시스템들을 사용하여 구현될 수 있다.

[0083] 이러한 예에 있어서, 리버스 프록시 서버(420)는 통신 네트워크들(415) 및/또는 방화벽들(435a)을 통해 클라이언트 디바이스들(410)로부터 메시지들을 수신하도록 구성된 메시지 핸들러(message handler)(421)를 포함한다. 일부 실시예들에 있어서, 메시지 핸들러(421)는 임의의 외부 네트워크들로부터 백엔드 웹 서비스들(430)로의 모든 TCP, UDP, HTTP, 및 HTTPS 트래픽에 대한 진입 포인트일 수 있다. 메시지 핸들러(421)는 또한 백엔드 웹 서비스들(430)로부터의 응답을 수신하고, 응답들을 클라이언트 디바이스들(410)로 송신하도록 구성될 수 있다. 일부 예들에 있어서, 메시지 핸들러(421)는 로드 밸런서들, 캐시들, 및/또는 메시지 조절기(throttler)들과 같은 하나 이상의 특수 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들을 포함할 수 있다.

[0084] 메시지들을 수신하고 파싱한 이후에, 메시지 핸들러(421)는 메시지들을 (예를 들어, 자바 네이티브 인터페이스(Java Native Interface; JNI) 또는 .NET 프로그래밍 프레임워크, 등을 통해) 적절한 웹 서비스 프레임워크로 송신할 수 있다. 예를 들어, 리버스 프록시 서버(420)에서 수신된 SOAP 요청들이 SOAP 웹 서비스 프레임워크(미도시)로 송신될 수 있으며, 반면 REST 요청들은 REST 웹 서비스 프레임워크(예를 들어, REST 인프라스트럭처(422))로 송신될 수 있다. 웹 컨텐트 요청들은, 예를 들어, 요청들을 파싱하고 URL 가상화 컴포넌트 또는 서비스와 같은 다양한 컴포넌트들로 송신함으로써, 메시지 핸들러(421)에 의해 유사하게 처리될 수 있다. 일부 경우들에 있어서, 메시지 핸들러(421)는 또한, SOAP 대 REST 및 REST 대 SOAP 메시지 변환들뿐만 아니라 자바스크립트 객체 표기법(JavaScript Object Notation; JSON) 대 XML 또는 JSON 대 SOAP, 및 이의 역들과 같은 프로토콜 변환들을 수행하도록 구성될 수 있다.

[0085] 프록시 서버(420)는 또한 표현 상태 전환(REST) 인프라스트럭처(422) 및 하나 이상의 REST 웹 서비스들(또는 REST API들)(423)을 포함할 수 있다. 도 4에 도시된 바와 같이, REST 인프라스트럭처(422)는 요청 핸들러(421)로부터 REST 웹 서비스들에 대한 요청들을 수신할 수 있다. 이하에서 더 상세하게 설명되는 바와 같이, REST 인프라스트럭처(422)는 요청들을 요청을 처리하기 위한 적절한 REST 서비스(423)로 포워딩하기 이전에 REST 요청들을 분석하고 프로세싱할 수 있다. 각각의 REST 서비스(423)는 요청을 처리하고 요청에 기초하여 대응하는 백

엔드 웹 서비스들(430)을 호출하도록 구성된 REST 자원들을 포함할 수 있다. 추가적으로, 일부 실시예들에 있어서, REST 인프라스트럭처(422)는 요청들을 처리하기 위하여 동적으로 REST 서비스들(423) 및 개별적인 REST 자원들을 생성할 수 있다.

[0086] REST 인프라스트럭처(422)는 (RESTful 웹 서비스들, RESTful API들, 등으로도 지칭되는) REST 웹 서비스들을 개발하고 실행하기 위한 웹 서비스 프레임워크를 포함할 수 있다. 일부 실시예들에 있어서, REST 인프라스트럭처(422)는 JAX-RS의 JERSEY 참조 구현(reference implementation)과 같은 JAX-RS(RESTful 웹 서비스들에 대한 Java API) 명세의 구현 예일 수 있다. JAX-RS 및 JERSEY 웹 서비스/API 인프라스트럭처와 관련하여 이하에서 다수의 예들이 논의되지만, 다른 프레임워크들 및 기술들이 다른 예들에서 사용될 수 있다. 예를 들어, 다른 REST 웹 프레임워크들/API 서비스들, 예컨대 SCALA, BOWLING FINCH/FINAGLE 프레임워크들이 JERSEY 및 JAX-RS에 대하여 대안적으로 또는 이에 더하여 사용될 수 있다.

[0087] 다양한 실시예들에 있어서, REST 인프라스트럭처(422)는 REST 웹 서비스들을 구현하기 위한 라이브러리(library)를 (예를 들어, JERSEY JAVA 서블릿 컨테이너(servlet container) 내에) 포함할 수 있으며, 또한 REST 자원들을 식별하기 위해 미리 정의된 클래스들을 스캔하도록 구성된 애플리케이션(예를 들어, JERSEY JAVA 서블릿)을 제공할 수 있다. REST 인프라스트럭처(422)는 또한 REST 웹 서비스들(423 및/또는 430)과 통신하기 위한 클라이언트 라이브러리를 제공할 수 있다. 클라이언트 디바이스들로부터의 요청들을 수신할 때, REST 인프라스트럭처 내의 애플리케이션(예를 들어, JERSEY JAVA 서블릿)은 들어오는 HTTP 요청들을 분석하고 요청에 응답하기 위한 정확한 자원 및 메소드를 선택할 수 있다. 특정 유형들의 웹 서비스들과 달리, REST 웹 서비스들은 XML 또는 임의의 다른 특정 데이터 포맷으로 통신해야 할 필요가 없다. 따라서, REST 인프라스트럭처(422)는, XML, 자바스크립트 객체 표기법(JSON), 콤마-구분 값들(comma-separated values; CSV), 또는 다양한 다른 데이터 포맷들로 데이터를 생성하는 것을 지원할 수 있다.

[0088] 도 4a 및 도 4b에 도시된 바와 같이, 리버스 프록시 서버(420)는 2개 이상의 컴퓨터 네트워크들 사이의, 예를 들어, 웹 서비스들(430)을 제공하는 신뢰되는 내부 네트워크(460)와 이를 통해 다양한 신뢰되지 않는 클라이언트 디바이스들(410)이 내부 웹 서비스들(430)을 액세스할 수 있는 신뢰되지 않는 외부 네트워크(480)(예를 들어, 인터넷) 사이의 중간 네트워크 디바이스 내에 구현될 수 있다. 도 4a에 도시된 바와 같이, 리버스 프록시 서버(420)는 내부 컴퓨터 네트워크(460)에 대한 통신 관리 및 보안의 초기 계층을 제공하기 위하여 내부 컴퓨터 네트워크(460)의 서브네트워크(465) 내에서 동작할 수 있다. 예를 들어, 보안 내부 네트워크(460)는 다양한 다른 서버들 및 클라이언트 디바이스들과 함께 복수의 웹 서비스들 및 애플리케이션들(430)을 포함할 수 있다. 리버스 프록시 서버(420) 및/또는 추가적인 네트워크 또는 컴퓨팅 디바이스들은 동일한 내부 네트워크(460)의 부분일 수 있지만, 이는 하나 이상의 게이트웨이들, 방화벽들(435b), 등에 의해 내부 컴퓨터 네트워크로부터 분리된 내부 컴퓨터 네트워크의 물리적인 서브네트워크(465) 내에서 동작할 수 있다. 일부 예들에 있어서, 리버스 프록시 서버(420)는 내부 컴퓨터 네트워크(460)의 논리적인 서브네트워크(465)(그렇지만 물리적인 서브네트워크는 아닌) 내에서 실행되는 프록시 서버 애플리케이션으로서 구현될 수 있다. 따라서, 리버스 프록시 서버(420)는 백엔드 웹 서비스들(430) 중 하나 이상과 동일한 컴퓨팅 시스템들 또는 내부 컴퓨터 네트워크(460) 내의 다른 컴퓨터 시스템들 상에 상주할 수 있다.

[0089] 추가적으로, 일부 실시예들에 있어서, 리버스 프록시 서버(420)는 신뢰되는 내부 네트워크(460)와 신뢰되지 않는 외부 네트워크(480) 사이의 비무장 지대(demilitarized zone; DMZ) 네트워크(470) 내에서 동작할 수 있다. 도 4b에 도시된 바와 같이, DMZ 네트워크(470)는, 내부 네트워크(460) 및 외부 네트워크(480) 둘 모두로부터 분리된, 전용 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들을 갖는 완전히 별개의 컴퓨팅 시스템으로서 구현될 수 있다. 대안적으로, 도 4a에 도시된 바와 같이, DMZ는 신뢰되는 내부 네트워크(460)의 물리적인 또는 논리적인 서브네트워크(465)로서 구현될 수 있으며, 그럼으로써 DMZ는 클라이언트 디바이스들(410) 및/또는 백엔드 웹 서비스들(430)에서 제공되는 엔드포인트 보안과는 별개의 통신 관리 및 보안의 제 1 계층을 제공한다. 어느 경우에든, DMZ 네트워크는 2개의 방화벽들(435a 및 435b) 사이에 구현될 수 있거나, 또는 단일 방화벽을 사용하여 또는 DMZ 네트워크(470) 또는 서브네트워크(465)를 신뢰되는 내부 네트워크(460) 및 신뢰되지 않는 외부 네트워크(415) 둘 모두로부터 물리적으로 또는 논리적으로 분리하는 네트워크 디바이스들의 다른 다양한 구성들을 사용하여 구현될 수 있다. 리버스 프록시 서버(420)와 같은 DMZ 내의 모든 컴퓨터 서버들 및 다른 디바이스들은 내부 네트워크(460) 내의 디바이스들의 특정한 서브세트(예를 들어, 웹 서비스들(430)을 호스팅하는 특정 서버들)에 대한 제한된 연결을 가질 수 있다. 이러한 연결은 특정 호스트들, 포트들, 프로토콜들, 및 유사한 것에 기초하여 제한될 수 있다. 유사하게, 외부의 신뢰되지 않는 네트워크(예를 들어, 네트워크(415) 및 클라이언트 디바이스들(410))와 통신할 때 제한된 연결의 정책들이 DMZ 내의 디바이스들에 대하여 집행될 수 있다. 리버스

프록시 서버(420)를 DMZ 내에서 동작시키는 것에 더하여, 특정 실시예들에 있어서, 백엔드 웹 서비스들(430) 중 하나 이상이 DMZ 내에서 동작할 수 있다. 예를 들어, 외부 시스템들(예를 들어, 웹 서버들, 이메일 서버들, 도메인 네임 시스템(Domain Name System; DNS) 서버들, 등)로부터의 공격들에 더 취약하거나 공격들을 당하기 쉬운 특정 컴퓨터 서버들/서비스들이 프록시 서버(420)를 갖는 DMZ 내로 이동될 수 있다.

[0090] 도 5는 하나 이상의 실시예들에 따른 리버스 프록시 서버의 특정 엘리먼트들 및 시스템들을 예시하는 블록도이다. 이러한 예에 있어서, 리버스 프록시 서버(520)는 리버스 프록시(420)에 대응할 수 있으며, 리버스 프록시 서버(520)의 다양한 서브컴포넌트들(521-523)은 이상에서 설명된 대응하는 서브컴포넌트들(421-423)과 유사하거나 또는 동일할 수 있다. 추가적으로, 리버스 프록시 서버(520)는 컴퓨팅 환경(400) 또는 다른 유사한 환경들 내에서 동작할 수 있으며, 클라이언트 컴퓨팅 디바이스들과 백엔드 웹 서비스들 사이에서 메시지들을 프로세싱하고 송신하도록 구성될 수 있다. 따라서, 리버스 프록시 서버(520)는 도 4a 및 도 4b를 참조하여 이상에서 논의된 하드웨어, 소프트웨어, 및 네트워킹 컴포넌트들 및 기능들 중 일부 또는 전부를 포함할 수 있다.

[0091] REST 인프라스트럭처(522)는, REST 서비스들 및 자원들을 개발하고, 배포하며, 실행하고, 액세스하기 위한 다양한 컴포넌트들(예를 들어, 애플리케이션들 및 라이브러리들)을 갖는 하나 이상의 REST 웹 서비스 프레임워크들을 포함할 수 있다. 다양한 실시예들에 있어서, REST 인프라스트럭처(522)는 JAX-RS 명세의 JERSEY 구현예 및/또는 다른 REST 웹 서비스 프레임워크 구현예들을 포함할 수 있다. REST 인프라스트럭처(522)는 REST 웹 서비스들을 구현하기 위한 하나 이상의 라이브러리들, REST 자원들을 식별하기 위하여 미리 정의된 클래스들을 스캔하기 위한 애플리케이션들, 및 REST 웹 서비스들과 통신하기 위한 클라이언트 라이브러리들, 등을 제공할 수 있다.

[0092] 이상에서 논의된 바와 같이, 요청 핸들러(521)는 REST 요청들을 REST 인프라스트럭처(522)로 포워딩할 수 있으며, 이는 요청들을 리버스 프록시 서버(520) 내의 하나 이상의 REST 서비스들로 포워딩하기 이전에 요청들을 분석하고 프로세싱할 수 있다. 이러한 예에 있어서, REST 인프라스트럭처(522)는 REST 스택(stack)(524), 정적 REST 애플리케이션(525)(예를 들어, 빌드-타임(build-time) 동안 생성된 JERSEY 애플리케이션), 및 REST 제공자 자원(526)을 포함한다. 이하에서 더 상세하게 논의되는 바와 같이, 리버스 프록시 서버(520)는 동적으로(즉, 런타임 동안) REST 스택(524)을 초기화하고 정적 REST 애플리케이션(525)을 등록할 수 있다. 정적 REST 애플리케이션(525)은, 요청을 처리하기 위하여 적절한 REST 서비스(523)의 루트(root) 자원을 식별하는 것을 담당할 수 있는 REST 제공자 자원(526)을 반환할 수 있다.

[0093] 도 5에 도시된 바와 같이, REST 제공자 자원(526)은 요청들을 리버스 프록시 서버(520) 내의 임의의 REST 서비스들(523)로 포워딩할 수 있다. 이러한 예에 있어서, 요청이 REST 서비스(523b)로 포워딩되었으며, 그 REST 서비스 내의 루트 자원(527)에 의해 수신되었다. 이러한 예에 도시된 바와 같이, REST 서비스(523)는, 그 내부에서 루트 자원(527)이 서비스(523) 내의 (예를 들어, 서비스의 "/" 경로에 매핑되는) 최상위-레벨 자원이며 상이한 요청들을 처리할 서브-자원들(528)(또는 자식 자원들)을 식별하는 것을 담당하는 자원 계층들로서 설계되고 구현될 수 있다. 루트 자원(527) 및/또는 모든 자식 자원들(528) 둘 모두는 이하에서 더 상세하게 논의되는 바와 같이 동적 REST 자원들(예를 들어, 런타임 동안 생성되는 자원들)로서 생성될 수 있다. 추가적으로, 이러한 예에서 단순한 2-레벨 계층이 도시되지만, 다수의 상이한 레벨들(예를 들어, 3-레벨, 4-레벨, 5-레벨, 등)의 복잡한 계층들이 다른 예들에서 구현될 수 있다.

[0094] 이제 도 6a 내지 도 6b(집합적으로 "도 6")를 참조하면, 리버스 프록시 서버들을 통해 REST 웹 서비스 요청들을 수신하고 프로세싱하기 위한 프로세스를 예시하는 순서도가 도시된다. 이하에서 설명되는 바와 같이, 이러한 프로세스 내의 단계들은 리버스 프록시 서버(420)(및/또는 리버스 프록시 서버들(520, 720, 및 920)) 및 그 내부에 구현된 다양한 서브시스템들/서브컴포넌트들과 같은 컴퓨팅 환경(400) 내의 하나 이상의 컴포넌트들에 의해 수행될 수 있다. 추가적으로, 일부 실시예들에 있어서, 이러한 프로세스 내의 특정 단계들은 클라이언트 디바이스들(410), 백엔드 웹 서비스들(430) 내에서, 및/또는 다른 다양한 중간 디바이스들에 의해서 수행될 수 있다. 메시지들을 수신하고 분석하는 것, 메시지 프로세싱 정책들을 선택하는 것 및 메시지들을 프로세싱하는 것을 포함하는 본원에서 설명되는 기술들이 이상에서 설명된 특정 시스템 및 하드웨어 구현예들에 반드시 한정되는 것이 아니라, 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들의 다른 조합들을 포함하는 다른 하드웨어 및 시스템 환경들 내에서 수행될 수 있다는 것이 추가로 이해되어야만 한다.

[0095] 단계(601)에서, 웹 서비스 요청이 리버스 프록시 서버(420)와 같은 중간 컴퓨팅 시스템 또는 애플리케이션에 의해 수신될 수 있다. 이상에서 언급된 바와 같이, 리버스 프록시 서버(420)는 신뢰되는 내부 네트워크(460)와 하나 이상의 신뢰되지 않는 외부 네트워크들 사이의 중간 서버 디바이스 및/또는 애플리케이션으로서 구현될 수

있다. 따라서, 리버스 프록시 서버(420)는 클라이언트 엔드포인트들(예를 들어, 클라이언트 디바이스들(410))에 의해 송신되고 몇몇 엔드포인트 디바이스들(예를 들어, 백엔드 웹 서비스들 및/또는 애플리케이션들(430)을 호스팅하는 컴퓨터 서버)에 대해 의도된 메시지 또는 이의 역인 메시지를 인터셉트할 수 있다.

[0096] 일부 실시예들에 있어서, 내부 네트워크(460)에 진입하거나 또는 이를 떠나는 모든 네트워크 트래픽이 리버스 프록시 서버(420)를 통해 라우팅될 수 있다. 다른 경우들에 있어서, 리버스 프록시 서버(420)는 특정 유형들 또는 프로토콜들의 네트워크 메시지들, 예를 들어, SOAP, REST, 또는 URL 자원들에 대한 클라이언트 디바이스들(410)로부터의 HTTP 요청들, 및 다시 클라이언트 디바이스들로 향하는 SOAP, REST, 또는 URL 웹 서비스들/애플리케이션들(430)로부터의 HTTP 응답들을 인터셉트하도록 구성될 수 있다. 따라서, 단계(601)에서 수신되는 웹 서비스 요청은, 예를 들어 그리고 비제한적으로, TCP 메시지, HTTP 또는 HTTPS 메시지, 단순 메일 전송 프로토콜(Simple Mail Transport Protocol; SMTP), 사용자 데이터그램 프로토콜(User Datagram Protocol; UDP) 메시지, 및/또는 자바 메시지 서비스(Java Message Service; JMS) 메시지일 수 있다. 일부 경우들에 있어서, 웹 서비스 요청은 클라이언트 디바이스(410)로부터 백엔드 웹 서비스(430)로의 SOAP, REST, 또는 웹 컨텐트 요청에 대응할 수 있거나, 또는, 클라이언트 디바이스(410)로부터의 SOAP, REST, 또는 웹 컨텐트 요청에 대한 백엔드 웹 서비스(430)에 의한 응답에 대응할 수 있다.

[0097] 단계(602)에서, 리버스 프록시 서버(420)는 요청이 리버스 프록시 서버(420) 내의 REST 서비스(423/523)에 의해 노출된 REST 자원(528)으로 향하는 것인지 아닌지 여부를 결정하기 위하여 단계(601)에서 수신된 요청을 분석할 수 있다. 본원에서 사용되는, 프로그래밍 객체 또는 데이터 객체와 같은 자원을 "노출시키는 것"은 자원에 대한 액세스를 제공하는 인터페이스를 제공한다는 것을 나타낼 수 있다. 예를 들어, 웹 서비스들 및 API들은, 클라이언트 애플리케이션들이 객체들을 액세스하거나 및/또는 조작하는 것을 가능하게 하는 메소드들 및 동작들을 제공함으로써 자원들을 노출시킬 수 있다.

[0098] 이상에서 논의된 바와 같이, 단계(601)에서 수신된 REST 웹 서비스 요청은 백엔드 웹 서비스(430)에 대하여 의도될 수 있다(및/또는 궁극적으로 이로 포워딩될 수 있다). 그러나, 다양한 실시예들에 있어서, 클라이언트 디바이스들(410)로부터 수신된 REST 요청들 중 일부 또는 전부를 처리하기 위하여 REST 서비스들 및 REST 자원들이 리버스 프록시 서버(420) 내에서 생성될 수 있다. 예를 들어, 도 4a 및 도 4b에 도시된 바와 같이, REST 서비스들(423)의 세트가 REST 요청들을 처리하기 위하여 리버스 프록시 서버(420) 내에서 생성될 수 있으며, 잠재적으로 백엔드 웹 서비스들(430)의 대응하는 세트를 호출할 수 있다. 따라서, 리버스 프록시 서버(420) 내의 REST 서비스들(423/523)의 세트는 백엔드 웹 서비스들(430)에 의해 노출되는 동일한 자원들의 일부 또는 전부를 노출시킬 수 있다. 일부 경우들에 있어서, 리버스 프록시 서버(420) 내의 이러한 REST 서비스들(423)은 "가상 서비스들"로서 지칭될 수 있다. 이하에서 더 상세하게 설명되는 바와 같이, 이러한 REST 서비스들(423)은 정확하게 백엔드 웹 서비스들(430)에 대응하거나 또는 대응하지 않을 수 있다. 예를 들어, 단일 REST 서비스(423)가 복수의 상이한 백엔드 서비스들(430)을 호출하는 REST 자원들(528)을 노출시킬 수 있거나, 또는 역으로, 복수의 상이한 REST 서비스들(423)이 동일한 백엔드 서비스(430)를 호출할 수 있다. 추가적으로, 리버스 프록시 서버(420) 내의 REST 서비스들(423)은 백엔드 웹 서비스들(430)에 의해 지원되지 않는 새로운 또는 상이한 동작들 또는 방법들, 파라미터들, 데이터 유형들, 및 유사한 것에 대한 지원을 포함할 수 있거나, 또는 이의 역할 수 있다.

[0099] 추가로, 일부 실시예들에 있어서 특정 REST 웹 서비스들(423) 및/또는 REST 자원들(527-528)이 리버스 프록시 서버(420) 내에서 동적으로 생성될 수 있다. 예를 들어, 리버스 프록시 서버(420) 내의 REST 인프라스트럭처(422)(및/또는 522, 722, 및 922) 및/또는 다른 컴포넌트들은 런타임 동안 다양한 REST 서비스들 및/또는 REST 자원들을 생성하거나, 삭제하거나, 대체하거나 또는 재생할 수 있다. 이러한 경우들에 있어서, 심지어 특정 REST 자원들(528)이 특정 시간에 리버스 프록시 서버(420) 상에 존재하지 않더라도, 그럼에도 불구하고 이러한 자원들이 동시에 리버스 프록시 서버(420)에 의해 계속해서 노출될 수 있다. 예를 들어, 웹 애플리케이션 기술 언어(Web Application Description Language; WADL) 파일이 REST 웹 서비스(423/523)에 의해 노출되는 자원들의 전부를 기술하기 위하여 생성될 수 있다. WADL 파일은, 심지어 이러한 자원들(528) 중 일부가 아직 생성되지 않은 경우에도, 어떠한 REST 자원들(528)이 REST 서비스(423)에 의해 노출될지를 결정하기 위해 사용될 수 있다. 리버스 프록시 서버(420) 내에서 다양한 REST 서비스들(423/523) 및 REST 자원들(528)을 동적으로 생성하고 재생하기 위한 기술들이 도 7 및 도 8을 참조하여 이하에서 더 상세하게 설명된다.

[0100] 단계(602)에서 요청이 리버스 프록시 서버(420) 내의 REST 서비스(423/523)에 의해 노출된 REST 자원(528)으로 향하는 것인지 아닌지 여부를 결정하기 위하여, 요청은 요청 목적지 및 메시지 헤더 및/또는 바디(body)의 관련된 부분들을 식별하기 위하여 파싱되고 분석될 수 있다. 예를 들어, 요청이 REST 서비스(423)에 대한 WADL 파일

(또는 다른 웹 서비스 설명 데이터) 내의 REST 자원에 대응하는 URL로 향해지는 HTTP 요청인 경우, 요청 핸들러(421)는, 요청이 리버스 프록시 서버(420)에 의해 노출된 REST 자원(528)으로 향해진다고 결정할 수 있다(602: 예). 대안적으로, 요청이 무효인 요청, SOAP 요청, 일반 웹 컨텐트에 대한 요청, 또는 리버스 프록시 서버(420)에 의해 액세스가 가능하지 않은 REST 자원으로 향해지는 요청인 경우, 요청 핸들러(421)는, 요청이 리버스 프록시 서버(420)에 의해 노출된 REST 자원(528)으로 향해지지 않는다고 결정할 수 있다(602: 아니오).

[0101] 단계(603)에서, 단계(601)에서 수신된 REST 웹 서비스 요청은 리버스 프록시 서버(420) 내의 REST 인프라스트럭처(422)로 포워딩될 수 있다. 예를 들어, 단계(602)에서 웹 서비스 요청이 리버스 프록시 서버(420)에 의해 노출된 REST 자원에 대한 요청이라고 결정한 이후에, 요청 핸들러(421)는 요청을 REST 인프라스트럭처(422) 내의 REST 스택(524)으로 포워딩할 수 있으며, 그러면 REST 스택(524)이 요청을 정적 REST 애플리케이션(525)으로 포워딩할 수 있다. 이상에서 논의된 바와 같이, 일부 실시예들에 있어서, REST 스택(524)은 JAX-RS 스택일 수 있으며, 정적 REST 애플리케이션은 빌드-타임 동안 생성된 JERSEY 애플리케이션일 수 있다. 일부 경우들에 있어서, 리버스 프록시 서버(520) 내의 컴포넌트들은 동적으로 JAX-RS 스택(524)을 초기화하고 정적 JERSEY 애플리케이션(525)을 등록하도록 구성될 수 있다.

[0102] 단계(604)에서, 리버스 프록시 서버(420)(및/또는 리버스 프록시 서버(520, 720 및 920))는 요청된 REST 자원(들)이 현재 리버스 프록시 서버(420) 상에 존재하는지 여부를 결정할 수 있다. 이상에서 언급된 바와 같이, 일부 실시예들에 있어서, REST 웹 서비스들(423)은 동적으로 생성될 수 있으며, 즉, 리버스 프록시 서버(420) 내의 REST 인프라스트럭처(422)(및/또는 522, 722, 및 922) 및/또는 다른 컴포넌트들에 의해 런타임 동안 생성될 수 있다. 따라서, 단계(604)에서, REST 인프라스트럭처(422) 및/또는 REST 서비스들(423)은 요청된 REST 자원들이 현재 리버스 프록시 서버(420) 상에 존재하는지 여부를 결정할 수 있다. 일부 경우들에 있어서, REST 요청의 수신 시에(602: 예), REST 스택(524)(예를 들어, JAX-RX 스택)은 요청을 수신하고 정적 REST 애플리케이션(525)(예를 들어, JERSEY 애플리케이션)을 호출할 수 있으며, 이는 등록된 REST 제공자 자원(526)을 호출할 수 있다. REST 제공자 자원(526)은 요청을 적절한 REST 서비스(423) 및/또는 가상 REST 서비스들(423) 내의 적절한 REST 자원들(428)에 위임할 수 있다. REST 제공자 자원(526) 및/또는 호출된 REST 서비스의 루트 자원(527)이 요청된 자원이 이전에 생성되었다는 것을 결정하는 경우(604: 예), 요청은 단계(606)에서 REST 서비스(523)의 루트 자원(527)으로 포워딩될 수 있으며, 그런 다음 단계(607)에서 요청을 처리하기 위한 적절한 REST 자원(528)으로 포워딩될 수 있다. 일부 실시예들에 있어서, 제공자 자원(526)은 요청의 URI를 사용하여 적절한 REST 루트 자원(527)을 결정할 수 있으며, REST 루트 자원(527)은 자원의 "/" 경로에 매핑될 수 있다. 그런 다음, REST 루트 자원(527)은, 요청을 처리하기 위한 리버스 프록시 서버(420) 내의 실제 REST 자원(528)의 위치를 찾기 위하여 서브-자원(sub-resource) 로케이터(locator)를 사용할 수 있다.

[0103] 대안적으로, REST 인프라스트럭처(526) 내의 하나 이상의 컴포넌트들 및/또는 호출된 REST 서비스(523)가 요청된 REST 요청이 아직 생성되지 않았다는 것(또는 재생성되어야 할 필요가 있다는 것)을 결정하는 경우(604: 아니오), 요청을 처리하기 위한 요청된 적절한 REST 자원이 단계(605)에서 생성될 수 있다. 리버스 프록시 서버(420) 내의 가상 REST 웹 서비스들(423)의 생성이 이하에서 도 7 내지 도 8을 참조하여 더 상세하게 논의된다. 단계(605)에서 요청을 처리하기 위한 적절한 REST 자원이 생성된 이후에, 그런 다음 단계(606)에서 요청이 REST 서비스(523)의 루트 자원(527)으로 포워딩될 수 있으며, 그런 다음 단계(607)에서 새로이 생성된 REST 자원(528)으로 포워딩될 수 있다.

[0104] 단계(608)에서, 요청의 처리 동안, 예를 들어, 리버스 프록시 서버(420) 내의 REST 자원(528)의 실행 동안, 하나 이상의 백엔드 웹 서비스 호출들이 결정될 수 있다. 일부 실시예들에 있어서, 소프웨어 후크(hook) 및/또는 다른 맞춤화된 소프트웨어 코드가 자원이 생성될 때 REST 자원(528) 내에 삽입될 수 있다. 이러한 소프트웨어 후크들 및/또는 맞춤화된 소프트웨어 코드는 백엔드 REST 서비스들(430) 및 자원들뿐만 아니라, 개별적인 동작들(또는 메소드들), 파라미터들, 및 백엔드 웹 서비스(430)에 대한 호출 내에서 송신될 수 있는 다른 데이터를 식별할 수 있다. 요청을 처리하기 위하여 리버스 프록시 서버(420) 상의 REST 자원(528)이 호출될 때, 소프트웨어 후크 및/또는 맞춤화된 소프트웨어 코드가 실행될 수 있으며, 백엔드 웹 서비스(430)에 대한 호출이 생성될 수 있다.

[0105] 일부 경우들에 있어서, 단계(608)에서 결정된 백엔드 웹 서비스 호출은 리버스 프록시 서버(420) 상의 REST 자원(528)에 의해 처리되는 REST 요청과 유사하거나 또는 동일할 수 있다. 예를 들어, 이하에서 더 상세하게 설명되는 바와 같이, 리버스 프록시 서버(420) 내의 REST 서비스들(423)은, (예를 들어, 웹 애플리케이션 기술 언어(WADL) 파일에 기초하는) 동일한 자원 정의들을 가지며, 메소드들 및 파라미터들, 등을 지원하는 백엔드 REST 서비스들(430)의 복사본(copy)들로서 생성될 수 있다. 이러한 경우들에 있어서, 리버스 프록시 서버(420) 상의

REST 자원(528)은, 동일한 REST 자원들을 호출하도록 구성된 단순한 소프트웨어 후크를 포함할 수 있으며, 대응하는 백엔드 REST 웹 서비스(430) 내의 파라미터들 및/또는 바디 컨텐트 등을 갖는 동일한 메소드들을 호출할 수 있다.

[0106]

다양한 다른 경우들에 있어서, 단계(608)에서 결정된 백엔드 웹 서비스 호출은 리버스 프록시 서버(420) 상의 REST 자원(528)에 의해 처음에 처리되는 REST 요청과 상이할 수 있다. 이러한 경우들에 있어서, 리버스 프록시 서버(420)에 의해 클라이언트 디바이스들(410)에 노출되는 REST 서비스들(423/523) 및 자원들(528)은 백엔드 REST 서비스들(430)에 의해 노출되는 REST 자원들에 대응해야 할 필요가 전혀 없다. 예를 들어, 도 4a 및 도 4b를 간략하게 참조하면, 가상 REST 서비스(423a) 및 그것의 대응하는 백엔드 REST 서비스(430a)는 (예를 들어, 동일한 자원들을 포함하며, 동일한 메소드들, 파라미터들, 데이터 유형들, 등을 지원하는 유사한(또는 동일한) WADL 파일을 갖는) 일부 경우들에 있어서 유사하게 구현될 수 있으며, 다른 경우들에 있어서는, 가상 REST 서비스들(423) 및 대응하는 백엔드 서비스들(430)이 전혀 상이하게 구현될 수 있다. 예시하기 위하여, 단계(601)에서 클라이언트 디바이스(410)로부터 수신된 REST 요청은, 메시지 헤더들 또는 바디 내에 REST 자원(528)에 대한 URL 경로, HTTP 메소드, 하나 이상의 URL 파라미터들 및/또는 추가적인 파라미터들 또는 데이터를 포함할 수 있다. 그러나, 메시지를 처리하기 위하여 REST 자원(528)이 실행될 때, 이는 상이한 URL 경로(예를 들어, 상이한 서브-자원 경로 또는 자원 명칭), 상이한 HTTP 메소드, 상이한 URL 파라미터들, 및/또는 상이한 헤더 또는 바디 데이터를 갖는 백엔드 웹 서비스 호출들을 생성할 수 있다. 추가적으로, 일부 실시예들에 있어서, 백엔드 웹 서비스들(430)은 REST 서비스들이어야 할 필요가 없으며, SOAP 서비스들, URL들, 또는 다른 유형들의 웹 서비스들, 애플리케이션들, 웹 컨텐트 등을 호출하도록 구성될 수 있다. 따라서, 클라이언트 디바이스들(410)은 오로지 리버스 프록시 서버(420) 내의 REST 서비스들(423/523)에 의해 노출되는 REST 자원들(528)의 지식만을 가질 수 있으며, 기초 백엔드 웹 서비스들(430)의 설계 또는 구조에 대한 어떠한 인식도 갖지 않을 수 있다. 가상 REST 서비스들(423) 및 그들의 대응하는 백엔드 서비스들(430)에 대한 상이한 웹 서비스/자원 설계들 및 구현 예들의 사용은, 일부 실시예들에 있어서, 예를 들어, 신뢰되지 않은 클라이언트 디바이스들(410)로부터 백엔드 서비스들(430)의 기초 설계를 가리기 위하여, 그리고 백엔드 웹 서비스들(430) 사이에 더 용이한 통합, 호환성, 및 확장성(scalability)을 제공하기 위하여 유익할 수 있다.

[0107]

백엔드 웹 서비스 호출이 생성된 이후에, 단계(609)에서, 리버스 프록시 서버(420)는 백엔드 웹 서비스 호출과 연관된 다양한 보안 정책들(및/또는 다른 통신 관리 정책들)을 집행할 수 있다. 그런 다음, 단계(610)에서, 신뢰되는 내부 네트워크(460) 내의 하나 이상의 백엔드 웹 서비스들(430)을 호출하기 위하여 호출이 리버스 프록시 서버(420) 내에서 실행될 수 있다. 도 9 내지 도 10을 참조하여 이하에서 더 상세하게 설명되는 바와 같이, 리버스 프록시 서버(420) 내의 REST 자원들(428)은, 호출들을 프로세싱하고 다양한 정책들을 집행할 수 있는 정책 집행 엔진(960)으로 백엔드 웹 서비스 호출들을 제공할 수 있다. 특정 실시예들에 있어서, 보안 정책들(및/또는 다른 통신 관리 정책들)은, 클라이언트(410)로부터 백엔드 웹 서비스들(430)로의 및 그 역으로의 요청-응답의 엔드-투-엔드(end-to-end) 프로세싱 흐름 내의 다양하고 상이한 부속(attachment) 포인트들(예를 들어, OnRequest, OnInvoke, OnResponse, MessageTransformation, OnError, 등)에서 집행될 수 있다.

[0108]

도 7은 하나 이상의 실시예들에 따른 리버스 프록시 서버의 특정 엘리먼트들 및 시스템들을 예시하는 블록도이다. 이러한 예에 있어서, 리버스 프록시 서버(720)는 리버스 프록시 서버(420) 및/또는 리버스 프록시 서버(520)에 대응할 수 있으며, 리버스 프록시 서버(720)의 다양한 서브컴포넌트들(722-723)은 이상에서 설명된 대응하는 서브컴포넌트들과 유사하거나 또는 동일할 수 있다. 추가적으로, 리버스 프록시 서버(520)는 컴퓨팅 환경(400) 또는 다른 유사한 환경들 내에서 동작할 수 있으며, 클라이언트 컴퓨팅 디바이스들과 백엔드 웹 서비스들 사이에서 메시지들을 프로세싱하고 송신하도록 구성될 수 있다. 따라서, 리버스 프록시 서버(720)는 도 4a 내지 도 4b 및/또는 도 5를 참조하여 이상에서 논의된 하드웨어, 소프트웨어, 및 네트워킹 컴포넌트들 및 기능들 중 일부 또는 전부를 포함할 수 있다.

[0109]

도 7에 예시된 특정 컴포넌트들은 리버스 프록시 서버(720) 내에서 REST 웹 서비스들(723)을 생성하도록 구성될 수 있다. 이상에서 언급된 바와 같이, 일부 실시예들에 있어서, REST 서비스들(723)에 의해 노출되거나 그 안에 포함되는 REST 자원들(728)과 함께 REST 서비스들(723)이 리버스 프록시 서버(720) 내에서 동적으로 생성될 수 있다. 이러한 실시예들에 있어서, 리버스 프록시 서버(720) 내의 REST 인프라스트럭처(722), REST 애플리케이션 엔진(740), 게이트웨이 관리 시스템(750) 및/또는 다른 컴포넌트들은 런타임 동안 리버스 프록시 서버 내에서 다양한 REST 서비스들(723) 및/또는 REST 자원들(728)을 생성하거나, 삭제하거나, 대체하거나 또는 갱신하도록

구성될 수 있다.

[0110] 이러한 예에 있어서, REST 인프라스트럭처(722)는 REST 스택(724), 정적 REST 애플리케이션(725)(예를 들어, 빌드-타임 동안 생성된 JERSEY 애플리케이션), 및 REST 제공자 자원(726)을 포함한다. 리버스 프록시 서버(720)는 런타임 동안 REST 스택(724)을 동적으로 초기화할 수 있으며, REST 제공자 자원(726)을 반환할 수 있는 정적 REST 애플리케이션(725)을 등록할 수 있다. REST 애플리케이션(725) 및/또는 제공자 자원(726)은, 리버스 프록시 서버(720)에 의해 노출되는 어떠한 REST 서비스들(723) 및 자원들(728)이 리버스 프록시 서버(720) 상에서 생성되었는지를 결정하도록 구성될 수 있다. 이러한 예에 있어서, REST 애플리케이션(725)은, 예를 들어, 클라이언트 디바이스(410)로부터의 REST 요청을 처리하기 위하여 REST 웹 서비스(723) 또는 REST 자원(728)이 생성되어야만 하는 때를 결정할 수 있다. 그런 다음, REST 애플리케이션(725)은 새로운 REST 서비스들(723) 및/또는 자원들(728)을 생성하도록 REST 애플리케이션 엔진(740)에 명령할 수 있다.

[0111] 도 7은 리버스 프록시 서버(720) 내의 REST 애플리케이션 엔진(740), 및 리버스 프록시 서버(720)와 통신하도록 구성된 (예를 들어, 내부 네트워크(460) 내의) 게이트웨이 관리 서버(750)를 포함한다. REST 애플리케이션 엔진(740)은 리버스 프록시 서버(720) 내에 REST 서비스들(723) 및/또는 자원들(728)을 생성하도록 구성된 하나 이상의 소프트웨어 툴들 또는 플랫폼들을 포함할 수 있다. 이하에서 논의되는 바와 같이, REST 애플리케이션 엔진(740)은, 예를 들어, 게이트웨이 관리 서버(750)로부터 수신되는 WADL 파일을 통해 REST 서비스 및/또는 자원 설명 데이터를 수신할 수 있다. 그런 다음, REST 애플리케이션 엔진(740)은 리버스 프록시 서버(720) 내에서 REST 서비스들(723) 및 자원들(728)을 생성하고, 빌드하며, 배포할 수 있다.

[0112] 이제 도 8을 참조하면, 리버스 프록시 서버 내에서 REST 서비스들 및/또는 자원들을 동적으로 생성하기 위한 프로세스를 예시하는 순서도가 도시된다. 이하에서 설명되는 바와 같이, 이러한 프로세스 내의 단계들은 이상에서 논의된 하나 이상의 컴포넌트들, 예컨대 리버스 프록시 서버(420)(및/또는 리버스 프록시 서버들(520, 720, 및 920)), REST 애플리케이션 엔진(740), 게이트웨이 관리 서버(750), 및/또는 그 내부에 구현된 다양한 서브시스템들 및 서브컴포넌트들에 의해 수행될 수 있다. 추가적으로, 일부 실시예들에 있어서, 이러한 프로세스 내의 특정 단계들은 클라이언트 디바이스들(410), 백엔드 웹 서비스들(430) 내에서, 및/또는 다른 다양한 중간 디바이스들에 의해서 수행될 수 있다. 웹 서비스 설명 데이터를 송신하고 수신하는 것, 및 동적으로 REST 서비스들 및 자원들을 생성하는 것을 포함하는 본원에서 설명되는 기술들이 이상에서 설명된 특정 시스템 및 하드웨어 구현예들에 반드시 한정되는 것이 아니라, 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들의 다른 조합들을 포함하는 다른 하드웨어 및 시스템 환경들 내에서 수행될 수 있다는 것이 추가로 이해되어야만 한다.

[0113] 이상에서 언급된 바와 같이, 단계들(801-804)은 프록시 서버 내에서 REST 서비스들 및 자원들을 동적으로 생성하기 위한 프로세스들과 관련되어 수행될 수 있다. 예를 들어, REST 서비스들(723) 및 REST 자원들(728)은 빌드 타임 동안이 아니라 런타임 동안 리버스 프록시 서버(720) 내에서 생성될 수 있다. REST 서비스 또는 자원의 동적인 생성은 런타임 동안 리버스 프록시 서버(720)에 의해 수신되는 다양한 정보에 기초하여 개시(또는 트리거(trigger))될 수 있다. 예를 들어, 클라이언트 디바이스(410)로부터 수신된 REST 요청이, 요청된 REST 자원이 아직 리버스 프록시 서버(720) 내에서 생성되지 않은 경우에 리버스 프록시 서버(720) 내에서의 동적인 REST 서비스/자원 생성 프로세스를 개시할 수 있다. 다른 예로서, 새로운 백엔드 웹 서비스(730)의 배포, 또는 기존의 백엔드 웹 서비스(730)에 대한 수정이, 리버스 프록시 서버(720) 내의 REST 서비스들(723)을 백엔드 웹 서비스들(730)과 동기화시키기 위하여 리버스 프록시 서버(720) 내에서의 동적인 REST 서비스/자원 생성 프로세스를 개시할 수 있다.

[0114] 단계(801)에서, 하나 이상의 백엔드 웹 서비스들(730)을 설명하는(및/또는 정의하는) 데이터가 리버스 프록시 서버(720) 내에서 수신될 수 있다. 일부 실시예들에 있어서, 단계(801)에서 수신되는 설명 데이터는 백엔드 웹 서비스들(730) 및/또는 신뢰되는 게이트웨이 관리 서버(750)에 의해 생성된 하나 이상의 웹 애플리케이션 기술 언어(WADL) 파일들을 포함할 수 있다. REST 서비스(730)에 대한 WADL 파일은, 자원 URL들, 지원되는 메소드들, 파라미터들, 데이터 유형들, 및 유사한 것을 포함하는, REST 서비스에 의해 노출되는 자원들의 전부를 설명할 수 있다. 이상에서 언급된 바와 같이, 백엔드 웹 서비스들(730)은 REST 웹 자원들(730)뿐만 아니라 다른 유형들의 웹 서비스들(730), 예컨대 SOAP 웹 서비스들 및 다른 유형들의 백엔드 웹 서비스들, 애플리케이션들, 웹 컨텐트, 등을 포함할 수 있다. 따라서, SOAP 웹 서비스들(730)에 대한 웹 서비스 기술 언어(Web Service Description Language; WSDL)가 WADL 대신에 사용될 수 있으며, 다양한 다른 데이터 포맷들이 백엔드 웹 서비스들(730)의 유형들에 의존하여 다른 예들에서 사용될 수 있다.

[0115] 수신된 설명 데이터의 유형 또는 포맷과 무관하게, 데이터는 백엔드 웹 자원들(730)의 구조 및 지원되는 동작들

을 결정하기 위하여 수신 디바이스에 의해 파싱될 수 있는 기계-판독가능 포맷으로 송신될 수 있다. 일부 실시 예들에 있어서, 설명 데이터(예를 들어, WADL 파일들)은 개별적인 백엔드 웹 서비스들(730)에 의해 생성되고, 게이트웨이 관리 서버(750)에 제공되며, 그런 다음 REST 애플리케이션 엔진(740)으로 송신될 수 있다. 예를 들어, 게이트웨이 관리 서버(750)는, 새로운 백엔드 웹 서비스(730)의 배포 또는 기존의 백엔드 웹 서비스(730)에 대한 수정에 응답하여 자동으로 갱신된 설명 데이터를 REST 애플리케이션 엔진(740)으로 송신할 수 있다. 다른 경우들에 있어서, REST 애플리케이션 엔진(740)은, 주기적으로 또는 리버스 프록시 서버(720) 내에서 발생하는 이벤트에 응답하여, 게이트웨이 관리 서버(750)로부터(또는 직접적으로 백엔드 웹 서비스들(730)로부터) 설명 데이터를 검색할 수 있다.

[0116] 단계들(802-804)에서, 리버스 프록시 서버(720) 내의 REST 애플리케이션 엔진(740) 및/또는 다른 컴포넌트들이 REST 서비스들(723) 및/또는 REST 자원들(728)을 리버스 프록시 서버(720) 내에서 생성하고, 빌드하며, 배포할 수 있다. 이러한 예에 있어서, 단계(802)에서, REST 애플리케이션 엔진(740)은 백엔드 웹 서비스들(730)에 의해 노출되는 자원들에 기초하여 REST 서비스 설명 데이터를 수정하거나 및/또는 맞춤화할 수 있다. 단계(803)에서, REST 애플리케이션 엔진(740)은 단계(802)에서 수정된/맞춤화된 REST 서비스 설명 데이터를 사용하여 리버스 프록시 서버(720) 내에 하나 이상의 더미(dummy) 레스트(rest) 자원들을 생성할 수 있다. 마지막으로, 단계(804)에서, REST 애플리케이션 엔진(740) 및/또는 REST 인프라스트럭처(722)는 단계(803)에서 생성된 REST 서비스들(723) 및/또는 자원들(728)을 리버스 프록시 서버(720) 내에 배포할 수 있다.

[0117] 일부 실시 예들에 있어서, REST 서비스들을 생성하고 빌드하는 단계들은, 예를 들어, REST 애플리케이션 엔진(740) 및/또는 REST 인프라스트럭처(722)에 의해 리버스 프록시 서버(720) 내에서 수행될 수 있다. 그러나, 다른 예들에 있어서, REST 서비스들(723) 및 REST 자원들(728)은, 리버스 프록시 서버(720)에 제공되고 배포되기 이전에, 리버스 프록시 서버(720)로부터 원격적으로(예를 들어, 내부 네트워크(460) 내에서) 설계되거나, 생성되거나, 및/또는 빌드될 수 있다.

[0118] REST 자원들(728)에 대한 수정들 및/또는 맞춤화들은 하나 이상의 백엔드 웹 서비스들(730)을 호출하기 위하여 더미 REST 자원 내로 삽입되는 소프트웨어 후크 및/또는 다른 맞춤화된 소프트웨어 코드의 형태를 취할 수 있다. 이러한 소프트웨어 수정들/맞춤화들은, 예를 들어, 백엔드 REST 서비스들(730) 및 자원들뿐만 아니라, 개별적인 동작들(또는 메소드들), 파라미터들, 및 백엔드 웹 서비스(730)에 대한 호출 내에서 송신될 수 있는 다른 데이터를 식별할 수 있다. REST 자원(728)이 리버스 프록시서버(720) 상에서 빌드되고 배포된 이후에, 백엔드 웹 서비스(730)를 호출하기 위하여 소프트웨어 후크 및/또는 맞춤화된 소프트웨어 코드가 실행될 수 있다. 이상에서 논의된 바와 같이, 일부 경우들에 있어서, 단계(803)에서 REST 자원들(728) 내로 삽입된 소프트웨어 코드는, 동일한 REST 자원들을 호출하도록 구성된 단순한 소프트웨어 후크를 포함할 수 있으며, 대응하는 백엔드 REST 웹 서비스(730) 내의 파라미터들 및/또는 바디 컨텐트 등을 갖는 동일한 메소드들을 호출할 수 있다. 다른 경우들에 있어서, 단계(803)에서 수정된/맞춤화된 코드는 상이한 URL 경로들(예를 들어, 상이한 서브-자원 경로 또는 자원 명칭), 상이한 HTTP 메소드들, 상이한 URL 파라미터들, 및/또는 상이한 헤더 또는 바디 데이터를 사용하여 하나 이상의 상이한 백엔드 웹 서비스들(730)을 호출하도록 설계될 수 있다. 추가적으로, 이상에서 논의된 바와 같이, 백엔드 웹 서비스들(730)은 REST 서비스들이어야 할 필요가 없으며, SOAP 서비스들, URL들, 또는 다른 유형들의 웹 서비스들, 웹 애플리케이션들, 또는 웹 컨텐트일 수 있다. 이러한 경우들에 있어서, 단계(803)에서의 소프트웨어 맞춤화/수정은 SOAP 및 URL 호출들을 생성하기 위한 및/또는 다양하고 상이한 유형들의 백엔드 웹 서비스들, 애플리케이션들, 웹 컨텐트 등을 호출하기 위한 맞춤 코드를 포함할 수 있다.

[0119] 단계(804)에서 리버스 프록시 서버(720) 내에서 REST 자원들(728)을 배포할 때, 각각의 자원(728)은 자원(728)의 URL에 따라서 REST 서비스(723) 내에서 계층적으로 배포될 수 있다. 이상에서 논의된 바와 같이, REST 서비스들은, 서비스(723) 내의 (예를 들어, 서비스의 "/" 경로에 매핑되는) 최상위-레벨 자원이며 상이한 요청들을 처리할 서브-자원들(728)(또는 자식 자원들)을 식별하는 것을 담당하는 루트 자원을 가질 수 있다.

[0120] 도 9는 정책 집행 엔진(960)을 포함하는 리버스 프록시 서버(920)의 특정 엘리먼트들 및 시스템들을 예시하는 블록도이다. 이러한 예에 있어서, 리버스 프록시 서버(920)는 리버스 프록시 서버(420)(및/또는 520 및 720)에 대응할 수 있으며, 리버스 프록시 서버(920)의 다양한 서브컴포넌트들(921-923)은 이상에서 설명된 대응하는 서브컴포넌트들과 유사하거나 또는 동일할 수 있다. 추가적으로, 리버스 프록시 서버(920)는 컴퓨팅 환경(400) 또는 다른 유사한 환경들 내에서 동작할 수 있으며, 클라이언트 컴퓨팅 디바이스들과 백엔드 웹 서비스들 사이에서 메시지들을 프로세싱하고 송신하도록 구성될 수 있다. 따라서, 리버스 프록시 서버(920)는 도 4a 내지 도 4b, 도 5, 및/또는 도 7을 참조하여 이상에서 논의된 하드웨어, 소프트웨어, 및 네트워킹 컴포넌트들 및 기능들

중 일부 또는 전부를 포함할 수 있다.

[0121] 요청 핸들러(921), REST 구현부(922), 및 REST 서비스들(923)에 더하여, 이러한 예는 리버스 프록시 서버(920) 내의 정책 집행 엔진(960)을 예시한다. 이러한 예에 있어서, REST 서비스들(923) 및 그들의 개별적인 REST 자원들(428)은 호출될 임의의 백엔드 웹 서비스 호출들을 정책 집행 엔진(960)에 제공할 수 있다. 정책 집행 엔진(960)은, 백엔드 웹 서비스들(930)을 호출하기 이전에 및/또는 이후에, 백엔드 웹 서비스 호출들을 프로세싱하고 다양한 보안 정책들 및 다른 통신 관리 정책들을 집행할 수 있다.

[0122] 정책 집행 엔진(960)은 보안 정책들을 구현하도록 구성된 다양한 보안 시스템들 또는 컴포넌트들뿐만 아니라, 다른 통신 관리 정책들을 리버스 프록시 서버(920) 내에 포함할 수 있다. 이러한 예에 있어서, 정책 집행 엔진(960)은 메시지 조절 시스템(961), 인증 및 인가 시스템들(962), 키 관리 시스템(963), 및 토큰 중재 시스템(964)을 포함한다. 정책 집행 엔진(960) 내의 이러한 시스템들 및 보안 컴포넌트들은, 클라이언트 디바이스들(410)로부터의 메시지들을 인증하고, 보안 토큰 중재를 제공하며, API 키 관리를 수행하고, 미세 단위(fine grained) 인가 및/또는 데이터 치환(data redaction)을 수행하며, 기밀성 및 무결성을 지원하고, 위협-기반 인증을 수행하며, 모바일 클라이언트 디바이스들(410)에 대한 디바이스-기반 보안을 수행하고, 비무장 지대(DMZ) 위협 보호를 지원하며, 프로토콜 및 페이로드(payload) 중재를 수행하고, 및 유사한 것을 수행할 수 있다. 예를 들어, 인증/인가 시스템들(962)은 서비스 거부(Denial of Service; DoS) 공격들을 방지하고, 기형 메시지들을 검출 및 필터링하며, SQL, 자바스크립트, 및/또는 XPath/XQuery 주입 공격들을 검출 및 방지하고, 악의적인 컨텐트로부터 보호하기 위하여 메시지 검증(예를 들어, 메시지 첨부물 내의 바이러스들의 검출, XML 및 JSON 데이터 구조들의 검증, 폼 및 쿼리(query) 파라미터들의 검증, 등)을 수행하기 위한 서브시스템들을 포함할 수 있다. 토큰 중재 시스템(964)은 지정된 클라이언트 디바이스들(410)과 백엔드 웹 서비스들(930) 사이에서 인증 토큰들을 변환하도록 구성될 수 있다. 보안 시스템들(961-964)이 또한, 동작들을 제거함으로써, 예를 들어, 복수의 백엔드 API들 또는 서비스들을 집성함으로써 편성을 지원할 수 있으며, 자동 중재 또는 화해(composition)를 수행할 수 있다.

[0123] 추가적으로, 이러한 예에 있어서, 정책 집행 엔진(960)은 메시지 프로세싱 정책들(965)의 데이터 저장부를 포함한다. 메시지 프로세싱 정책들은 다양한 형태의 컴퓨터-판독가능 매체, 예컨대 XML, 자바스크립트, 또는 다른 유형들의 실행가능 소프트웨어 컴포넌트들로 저장될 수 있다. 이하에서 더 상세하게 논의되는 바와 같이, 메시지 프로세싱 정책들(965)은 리버스 프록시 서버(920) 내에서 보안 정책들 및 다른 통신 관리 정책들을 집행하기 위하여 사용될 수 있다. 데이터 저장부(965)는, 개별적인 메시지들에 대한 엔드-투-엔드 프로세싱 흐름 동안 다양한 단계들에서 검색되고 개별적인 메시지들에 적용될 수 있는 개별적인 메시지 프로세싱 정책들을 포함할 수 있다. 메시지 프로세싱 정책 데이터 저장부(965)는, 이러한 예에 도시된 바와 같이 리버스 프록시 서버(920) 내에 상주할 수 있거나, 또는 신뢰되는 내부 컴퓨터 네트워크(460)의 백엔드 서버 또는 보안 제 3 자 서버, 또는, 유사한 것 내에 상주할 수 있다.

[0124] 이제 도 10을 참조하면, 메시지 프로세싱 정책들을 결정하고 REST 요청들 및 다른 백엔드 웹 서비스 호출들과 같은 메시지들을 프로세싱하기 위한 프로세스를 예시하는 순서도가 도시된다. 이하에서 설명되는 바와 같이, 메시지 프로세싱 정책들의 결정 및 집행은, 그 내부에 구현된 다양한 서브시스템들/서브컴포넌트들과 함께 리버스 프록시 서버(920)(및/또는 420, 520, 및 72) 내의 하나 이상의 컴포넌트들에 의해 수행될 수 있다. 추가적으로, 일부 실시예들에 있어서, 이러한 프로세스 내의 특정 단계들은 클라이언트 디바이스들(410), 백엔드 웹 서비스들(930) 내에서, 및/또는 다른 다양한 중간 디바이스들에 의해서 수행될 수 있다. 엔드-투-엔드 메시지 프로세싱 흐름들을 모니터링하는 것, 메시지 프로세싱 정책들을 결정하는 것, 및 메시지 프로세싱 정책들을 집행하는 것을 포함하는 본원에서 설명되는 기술들이 이상에서 설명된 특정 시스템 및 하드웨어 구현예들에 반드시 한정되는 것이 아니라, 하드웨어, 소프트웨어, 및 네트워크 컴포넌트들의 다른 조합들을 포함하는 다른 하드웨어 및 시스템 환경들 내에서 수행될 수 있다는 것이 추가로 이해되어야만 한다.

[0125] 단계(1001)에서, 정책 집행 엔진(960)은, 메시지에 대한 목적지 백엔드 서비스(930)뿐만 아니라 메시지를 개시한 클라이언트 디바이스(410)를 결정하기 위하여 REST 서비스(923)로부터 수신된 메시지를 분석할 수 있다. 메시지의 목적지는, REST 서비스(923)에 의해 생성된 백엔드 웹 서비스 호출(예를 들어, REST 요청, SOAP 요청, 또는 웹 컨텐트 요청)을 파싱하고 분석함으로써 결정될 수 있다. 예를 들어, REST 또는 SOAP 요청의 URI(uniform resource identifier), 또는 웹 서비스 또는 애플리케이션의 식별자 및/또는 메시지 바디 내의 동작 식별자들이 내부 네트워크(460)에 의해 제공되는 백엔드 웹 서비스/애플리케이션(930) 또는 웹 컨텐트에 대응할 수 있다. 이러한 예에 있어서, 프록시 서버(920)는 메시지 헤더 및 컨텐트에 기초하여 메시지가 내부 네트워크(460) 내의 웹 서비스(930)를 호스팅하는 특정 컴퓨터 서버에 대하여 의도된다는 것을 결정할 수 있다. 예

시지의 송신기를 식별하는 메시지 내의 정보, 예컨대 소스 IP 어드레스 또는 호스트 명칭 식별자가 또한 메시지의 의도된 목적지를 결정하기 위하여 사용될 수 있다. 메시지의 의도된 목적지를 결정하는 것에 더하여, 프록시 서버(420)는 이로부터 메시지가 개시된 클라이언트 디바이스(410)를 결정할 수 있다.

[0126] 단계(1002)에서, 정책 집행 엔진(960)은 백엔드 웹 서비스(930)로 송신될 메시지에 대한 미리 결정된 프로세싱 흐름 내의 현재 포인트를 결정할 수 있다. 메시지 프로세싱 흐름은, 프록시 서버(920)에 의한 클라이언트 디바이스(410)로부터의 메시지의 수신으로 시작하여, 프록시 서버(920)에 의한 클라이언트 디바이스(410)로의 응답의 송신으로 종료하는 프록시 서버(920)에 의해 실행될 엔드-투-엔드 메시지 프로세싱 흐름을 지칭할 수 있다. 이하에서 논의되는 바와 같이, 메시지에 대한 미리 결정된 프로세싱 흐름 내의 현재 포인트를 결정하는 단계는, 메시지와 연관된 정책 모델을 식별하는 단계, 및 프로세싱 모델 내의 현재 프로세싱 위치를 결정하는 단계를 포함할 수 있다.

[0127] 일부 실시예들에 있어서, 메시지들에 대한 미리 결정된 메시지 프로세싱 흐름들은 정책 모델들에 의해 정의될 수 있다. 정책 모델은, 메시지의 엔드-투-엔드 메시지 프로세싱 흐름 동안 다양한 포인트들에서 메시지를 프로세싱하기 위하여 정책 집행 엔진(960)에 의해 적용될 수 있는 정책들(예를 들어, 보안 정책들, 통신 관리 정책들, 등)의 세트를 정의하는 데이터를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름을 정의하는 정책 모델들 및 개별적인 메시지 프로세싱 정책들 둘 모두는, 다양한 형태의 컴퓨터-판독가능 매체, 예컨대 XML, 자바스크립트, 또는 다른 유형들의 실행가능 소프트웨어 컴포넌트들일 수 있다. 정책 모델들 및/또는 메시지 프로세싱 정책들은 정책 집행 엔진(960) 내에, 예를 들어, 데이터 저장부(965)에, 또는 내부 네트워크(460) 내의 어딘가에 저장될 수 있다.

[0128] 이상에서 언급된 바와 같이, 정책 모델들은, 정책 집행 엔진(960)이 메시지의 엔드-투-엔드 프로세싱 흐름의 다양한 포인트들에서 메시지에 적용할 수 있는 메시지 프로세싱 정책들의 세트를 정의할 수 있다. 일부 실시예들에 있어서, 정책 집행 엔진(960)은 REST 서비스들(923)로부터 수신된 메시지의 특성들에 의존하여 단계(1002)에서 상이한 정책 모델들을 적용할 수 있다. 예를 들어, 정책 집행 엔진(960)에 의해 검색되고 적용되는 특정 정책 모델은 메시지의 의도된 목적지 및/또는 요청을 개시한 클라이언트 디바이스(410)에 의존할 수 있다. 추가적으로, 정책 집행 엔진(960)에 의해 검색되고 적용되는 정책 모델은, 메시지를 송신하기 위해 사용된 네트워크 프로토콜들 및/또는 메시지의 요청 유형 또는 클라이언트 유형에 의존할 수 있다. 예를 들어, REST 요청들, SOAP 요청들, 웹 컨텐트(URL) 요청들, 및 유사한 것에 대하여 상이한 정책 모델들이 사용될 수 있다.

[0129] 일부 예들에 있어서, 정책 모델들은 XML로 또는 다른 기계-판독가능 포맷으로 구현될 수 있다. 정책 모델들은 ("어쩌션(assertion)들"로서 지칭될 수 있는) 프로세싱 흐름 내의 다양한 포인트들의 태그들 또는 식별자들, 및 프로세싱 포인트들/어쩌션들의 각각에 대한 하나 이상의 정책 식별자들을 포함할 수 있다. 예를 들어, 정책 모델은 요청이 수신될 때 ("on-request" 태그 내의) 수행될 정책들, ("message-transformation" 태그들 내의) 메시지 변환을 수행하는 정책들, 및 백엔드 웹 서비스가 호출될 때 ("on-invocation" 태그 내의) 수행될 정책들을 식별한다.

[0130] 일부 실시예들에 있어서, 정책 집행 엔진(960)은 서비스 레벨(또는 URL 레벨)에서 및/또는 동작 레벨(또는 메소드 레벨)에서 정책들을 적용할 수 있다. 따라서, 백엔드 웹 서비스들(930)을 호출할 때, 정책 집행 엔진이 정책 모델 내에서 식별된 정책들을 집행할 수 있기 이전에 정책 집행 엔진(960)은 (SOAP에 대한) 동작 또는 (REST 및 URL에 대한) 메소드를 먼저 결정할 수 있다.

[0131] REST 서비스(923)로부터 수신된 메시지와 연관된 정책 모델(또는 프로세싱 흐름을 정의하는 다른 데이터)를 식별한 이후에, 정책 집행 엔진(960)은 정책 또는 프로세싱 흐름에 따른 메시지의 프로세싱에 있어서의 현재 포인트를 결정할 수 있다. 메시지 프로세싱 흐름 내의 현재 포인트는 메시지 자체의 특성들에 의해서뿐만 아니라 메시지의 이전의 프로세싱에 관하여 이전에 저장된 데이터에 기초하여 결정될 수 있다. 이상에서 언급된 바와 같이, 미리 결정된 프로세싱 흐름은, 클라이언트 디바이스(410)에 의한 초기 요청으로부터, 클라이언트 디바이스(410)로 다시 송신되는 응답에 이르는, 메시지에 대한 엔드-투-엔드 프로세싱을 적용할 수 있다. 따라서, 메시지가 클라이언트 디바이스(410)로부터 수신된 초기 요청인지, 클라이언트 디바이스로부터의 추가적인 데이터(예를 들어, 인증 증명서들 또는 요청과 관련된 추가적인 데이터)의 송신인지, 백엔드 웹 서비스(930)로부터의 응답인지, 또는 백엔드 서버 또는 디바이스로부터의 추가적인 데이터(예를 들어, 싱글 사인-온 또는 토큰 번역(transliteration) 서비스로부터의 데이터)의 송신인지 여부를 결정하는 것은, 정책 집행 엔진(960)이 엔드-투-엔드 메시지 프로세싱 흐름 내의 메시지 프로세싱의 현재 포인트를 결정하는 것을 가능하게 할 수 있다. 추가적으로, 정책 집행 엔진(960)은, 정책 집행 엔진(960)이 메시지에 적용해야만 하는 다음의 메시지 프로세싱 정책을 결정

하기 위하여, 메시지 또는 다른 관련된 메시지들에 대해 수행된 이전의 프로세싱과 관련된 데이터, 예컨대 이전의 메시지 변환들의 결과들, 서비스들의 호출들, 경험된 프로세싱 오류들을 저장할 수 있다.

[0132] 다음의 단락들은, 여기에서 메시지 프로세싱 정책들이 적용될 수 있는, 정책 모델 또는 다른 메시지 프로세싱 흐름 내의 가능한 ("어쩌션들"로서도 지칭될 수 있는) 포인트들의 몇몇 예들을 포함한다. 이러한 예들이 오로지 예시적일 뿐이며, 완전한 리스트일 필요가 없다는 것이 이해되어야만 한다. 또한, 본원에서 설명되는 어쩌션 명칭들(예를 들어, OnRequest, OnInvoke, OnResponse, OnError, MessageTransformation, 등)뿐만 아니라 어쩌션들 및 정책들에 대해 사용되는 XML 구조들 및 태그 명칭들이 다양한 다른 실시예들에서 변화될 수 있다.

[0133] 단계(1002)에서 정책 모델 또는 다른 미리 결정된 메시지 프로세싱 흐름 내의 현재 포인트를 결정하는 단계의 제 1 예는, 수신된 메시지가 외부 컴퓨터 네트워크 내의 클라이언트 디바이스(410)로부터의 요청에 대응한다는 것을 결정하는 단계를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름의 개시에서의 이러한 포인트는 "OnRequest" 어쩌션 또는 유사한 것으로서 지칭될 수 있다. 이하에서 더 상세하게 논의되는 바와 같이, OnRequest 어쩌션은 가상 서비스들, 프록시 서비스들, 및/또는 웹 애플리케이션들을 안전하게 하기 위하여 적용될 수 있는 정책들에 대한 참조(reference)들을 포함할 수 있다. 예를 들어, OnRequest 어쩌션은, 정책 집행 엔진(960)이 외부 클라이언트 디바이스들(410)로부터 수신된 새로운 웹 서비스/애플리케이션/컨텐츠 요청들에 대하여 집행해야만 하는 보안 정책들을 나타내는 URI들 또는 다른 식별자들을 포함할 수 있다. OnRequest 어쩌션들은 또한 다른 정책들을 참조하게 할 수 있거나 및/또는 다른 어쩌션들을 포함할 수 있다. 일부 경우들에 있어서, OnRequest 어쩌션들은 오로지 리버스 프록시 모드에서만 동작할 수 있으며, 즉, 이들은 오로지 내부 웹 서비스들(930)에 대한 외부 클라이언트 디바이스들(410)로부터의 요청들만을 처리할 수 있다.

[0134] 단계(1002)에서 일어날 수 있는 현재 메시지 프로세싱 포인트의 다른 결정은, 클라이언트 디바이스(410)로부터 요청을 수신하는 단계 이후에, 프록시 서버(920)가 백엔드 웹 애플리케이션 또는 웹 서비스(930)로 요청을 송신해야만 한다는 것을 결정하는 단계를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트는 "OnInvoke" 어쩌션 또는 유사한 것으로서 지칭될 수 있다. OnRequest 어쩌션과 유사하게, 일부 실시예들에 있어서, OnInvoke 어쩌션은 오로지, 초기 요청이 내부 네트워크(460) 내의 백엔드 웹 서비스들/애플리케이션들(930)을 호출하기 위하여 클라이언트 디바이스(410)로부터 수신되었던 리버스 프록시 사용 경우들에서만 적용될 수 있다. OnInvoke 어쩌션들은, 정책 집행 엔진(960)이 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트 동안 집행해야만 하는 정책들을 나타내는 URI들 또는 다른 식별자들을 포함할 수 있다. 복수의 정책 식별자들(또는 참조들)은, 예를 들어, 복수의 XML "Policy URI" XML 엘리먼트들을 사용함으로써 OnInvoke 내에 포함될 수 있다. 추가적으로, OnInvoke 어쩌션들은 클라이언트의 자원 패턴을 사용하는 것으로부터 클라이언트 세부사항들을 고유하게 식별할 수 있다. OnInvoke 어쩌션들에 대해 사용되는 클라이언트 유형(예를 들어, REST 클라이언트, SOAP 클라이언트, URL/웹 클라이언트, 등)은, OnInvoke 어쩌션 내에 구성된 값들에 기초하여 런타임 시에 정책 집행 엔진(960)에 의해 결정될 수 있다. OnInvoke 어쩌션들은 또한 다른 정책들을 참조하게 할 수 있거나 및/또는 다른 어쩌션들을 포함할 수 있다.

[0135] 클라이언트 디바이스(410)로부터 요청을 수신하는 단계 이후에 그리고 백엔드 웹 서비스(430)를 호출하는 단계 이후에, 현재 메시지 프로세싱 포인트를 결정하는 단계의 다른 예는, 프록시 서버(920)가 클라이언트 디바이스(410)로 응답을 송신해야만 한다는 것을 결정하는 단계를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트는 "OnResponse" 어쩌션 또는 유사한 것으로서 지칭될 수 있다. OnRequest 및 OnInvoke 어쩌션들과 유사하게, 일부 실시예들에 있어서, OnResponse 어쩌션은 오로지, 초기 요청이 내부 네트워크(460) 내의 백엔드 웹 서비스(930)를 호출하기 위하여 클라이언트 디바이스(410)로부터 수신되었던 리버스 프록시 사용 경우들에서만 적용될 수 있다. OnResponse 어쩌션들은, 정책 집행 엔진(960)이 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트 동안 집행해야만 하는 정책들을 나타내는 URI들 또는 다른 식별자들을 포함할 수 있다. 복수의 정책 식별자들(또는 참조들)이 OnResponse 내에 포함될 수 있으며, OnResponse 어쩌션들이 또한 다른 정책들을 참조하게 할 수 있거나 및/또는 다른 어쩌션들을 포함할 수 있다.

[0136] 현재 메시지 프로세싱 포인트를 결정하는 단계의 다른 예는, 엔드-투-엔드 프로세싱 흐름 동안 어떤 포인트에서, 프록시 서버(920)가 메시지를 하나의 메시지 유형으로부터 다른 메시지 유형으로 변환해야만 한다는 것을 결정하는 단계를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트는 "MessageTransformation" 어쩌션 또는 유사한 것으로서 지칭될 수 있다. 예를 들어, 프록시 서버(920)는 제 1 메시지 유형(예를 들어, REST 요청)을 갖는 메시지를 수신할 수 있으며, 메시지를 분석하여 메시지가 오로지 제 2 메시지 유형(예를 들어, 백(back) SOAP 서비스)만을 수락하는 백엔드 서비스 또는 애플리케이션에 대해 의도된다는 것을 결정할 수 있다. 이러한 결정 이후에, 정책 집행 엔진(960)은, 변환된 메시지를 의도된 목적지로

전송하기 이전에 메시지에 대하여 적절한 MessageTransformation 어쩌션을 실행할 수 있다. 정책 집행 엔진(960)에 의해 지원될 수 있는 변환 정책들의 예들은, 비제한적으로, XML 대 자바스크립트 객체 표기법(JSON) 및 JSON 대 XML 정책들, XML 대 SOAP 및 SOAP 대 XML 정책들, 및 JSON 대 SOAP 및 SOAP 대 JSON 정책들을 포함할 수 있다. 다른 잘-알려진 매체 유형들 사이의 변환이 다양한 실시예들에서 지원될 수 있다. 정책 집행 엔진(960)은 자동적으로 백엔드 서비스 가상화의 시점에 적절한 변환 정책들을 첨부(attach)할 수 있으며, 변환들은 프록시 서버(920)에 또는 컴퓨팅 환경 내의 어딘가에 설치된 하나 이상의 번역 프레임워크들을 사용하여 수행될 수 있다.

[0137]

현재 메시지 프로세싱 포인트를 결정하는 단계의 또 다른 예는, 메시지에 대한 엔드-투-엔드 프로세싱 흐름 동안 어떤 포인트에서 오류가 발생하였다는 것을 결정하는 단계를 포함할 수 있다. 메시지의 엔드-투-엔드 프로세싱 흐름 내의 이러한 포인트는 "OnError" 어쩌션 또는 유사한 것으로서 지정될 수 있다. 메시지에 대한 OnError 어쩌션을 트리거하는(예를 들어, 메시지와 연관된 OnError 어쩌션 내의 식별된 하나 이상의 정책들의 실행을 트리거하는) 오류들은 리버스 프록시 서버(920)에 의해 이루어진 프로세싱 내에서 발생하는 오류들 및/또는 백엔드 컴퓨터 서버 또는 디바이스로부터 리버스 프록시 서버(920)에 의해 수신된 오류일 수 있다. 예를 들어, 리버스 프록시 서버(920)는, 인증 서비스, 토큰 번역 서비스, 또는 백엔드 웹 서비스(930)와 같은 메시지의 프로세싱 흐름 동안 호출된 백엔드 컴퓨터 서버로부터 오류 표시를 수신할 수 있다. 추가적으로, 리버스 프록시 서버(920)는, 메시지 프로세싱 태스크들을 수행하는 동안, 메시지를 파싱하거나 또는 검증함에 있어서의 오류들, 또는 메시지 변환 정책을 실행할 때의 오류들과 같은 오류들을 식별하거나 또는 생성할 수 있다. 따라서, 특정 메시지 프로세싱 정책들이 적용될 수 있는 프로세싱 흐름들 내의 포인트들("어쩌션들"로도 지칭됨)의 이전의 예들의 일부와 다르게, OnError 어쩌션들은 조건적일 수 있다. 즉, 메시지의 엔드-투-엔드 프로세싱 흐름 동안, 정책 집행 엔진(960)은, 프로세싱 동안 발생할 수 있는 오류들의 수 및 유형에 의존하여 OnError 어쩌션으로부터의 정책을 한번 적용하거나, 복수 회 적용하거나, 또는 전혀 적용하지 않을 수 있다.

[0138]

단계(1003)에서, 단계(501)에서 수신된 메시지를 프로세싱하기 위한 하나 이상의 특정 정책들이 정책 집행 엔진(960)에 의해 결정될 수 있다. 이상에서 논의된 바와 같이, 리버스 프록시 서버(920)에 의해 선택되고 메시지들에 적용되는 특정 정책들은 보안 정책들뿐만 아니라 임의의 다른 유형들의 통신 관리 정책들을 포함할 수 있다. 예를 들어 그리고 비제한적으로, 이러한 정책들은, 다른 것들 중에서도 특히 인증, 인가, 감사, 싱글 사인 온, 보안 정책 집행, 키 관리 및 분배, 보안 통신, 보안 데이터 저장, 및 보안 데이터 공유와 관련된 기능들을 수행할 수 있다.

[0139]

정책들은, 먼저 메시지와 연관된 엔드-투-엔드 프로세싱 흐름(들)(예를 들어, 정책 모델들)을 검색하고 그런 다음 엔드-투-엔드 흐름 내의 현재 포인트에서 메시지에 적용될 특정 정책들을 식별하기 위하여 단계(1002)에서 결정된 엔드-투-엔드 프로세싱 흐름들 내의 현재 포인트(예를 들어, 어쩌션)를 사용함으로써 정책 집행 엔진(960)에 의해 단계(1003)에서 선택될 수 있다. 예를 들어, 메시지가 클라이언트 디바이스(410)로부터의 백엔드 웹 서비스(430)에 대한 요청인 경우, 정책 집행 엔진(960)은 메시지에 대한 엔드-투-엔드 프로세싱 흐름을 지배하는 정책 모델의 "on-request" 태그 내에서 식별되는 임의의 정책들을 검색할 수 있다.

[0140]

단계(1004)에서, 정책 집행 엔진(960)은 단계(1003)에서 선택된 정책들을 사용하여 메시지를 프로세싱할 수 있다. 이상에서 논의된 바와 같이, 정책 집행 엔진(960)은, 메시지에 대한 미리 결정된 엔드-투-엔드 프로세싱 흐름으로부터 URI들 또는 다른 정책 식별자들을 식별함으로써 메시지에 적용될 적절한 정책들을 결정할 수 있다. 일부 실시예들에 있어서, 정책 모델은, 엔드-투-엔드 프로세싱 흐름 내의 현재 포인트에 대응하는 어쩌션들에 의존하여 적용될 정책들에 대한 정책 URI들(또는 다른 식별자들)을 포함할 수 있다. 이러한 정책 URI들은 정책들의 저장 위치들을 참조할 수 있다. 다른 예들에 있어서, 정책 식별자들이 URI들로서 표현될 필요가 없으며, 이들은 API 또는 서비스 식별자들, 함수 명칭들, 메소드 명칭들, 및/또는 동작 명칭들, 및 유사한 것과 같은 다른 식별 데이터를 포함할 수 있다. 어떤 경우에든지, 정책 식별자들은 메시지 프로세싱 정책들에 대한 저장 위치 또는 다른 액세스 정보를 식별할 수 있다. 정책들 자체가 다양한 형태의 컴퓨터-판독가능 매체, 예컨대 XML, 자바스크립트, 또는 다른 유형들의 실행가능 소프트웨어 컴포넌트들로 저장될 수 있다.

[0141]

메시지 프로세싱 정책들은 데이터 저장부들, 예컨대 컴퓨팅 환경의 내의 다양하고 상이한 서버들 또는 디바이스들 내에 위치된 데이터베이스들 및/또는 파일-기반 저장 시스템들에 저장될 수 있다. 예를 들어, 메시지 변환 정책들, 메시지 조절 정책들, 로드 밸런싱 정책들, 및 상대적으로 변화하지 않을 수 있으며 보안 데이터가 없는 다른 정책들과 같은 특정 정책들은 프록시 서버(920) 내에(예를 들어, 메시지 프로세싱 정책 데이터 저장부(965) 내에) 로컬적으로 저장될 수 있다. 사용자 인증/인가 정책들 및 빈번하게 변화될 수 있거나 또는 보안 데이터를 포함할 수 있는 다른 정책들과 같은 다른 정책들은 신뢰되는 내부 컴퓨터 네트워크(460)의 보안 서버 또

는 저장 시스템 내에 저장될 수 있다. 다른 경우들에 있어서, 특정 정책들은 외부 네트워크들 내의 보안 제 3자 서버들 또는 클라이언트 디바이스들(410) 상에 저장될 수 있다. 정책 집행 모델(960)은 단계(1004)에서 이러한 다양한 위치들 중 임의의 위치로부터 정책들을 검색하고 적용하도록 구성될 수 있다.

[0142] 단계(1004)에서의 다양한 보안 정책들 및/또는 다른 통신 관리 정책들을 사용하여 메시지를 프로세싱하는 단계 이후에, 단계(1005)에서, 리버스 프록시 서버(920)는 프로세싱된 메시지를 그것의 의도된 목적지로 송신할 수 있다. 이상에서 논의된 바와 같이, 의도된 목적지는 메시지 헤더 및/또는 메시지 바디의 부분들을 파싱하고 분석함으로써 단계(1001)에서 결정될 수 있다. 메시지의 의도된 목적지는, 백엔드 웹 서비스/(930)에 대한 요청과 같이 내부 네트워크(460) 내에 존재할 수 있다. 대안적으로, 메시지의 의도된 목적지는, 외부 웹 서비스 또는 애플리케이션에 대한 요청, 또는 클라이언트 디바이스(410)로의 응답 또는 다른 송신과 같이 외부 네트워크 내에 존재할 수 있다.

[0143] 이상에서 논의된 바와 같이, 프록시 서버(920) 내에서 메시지를 프로세싱하기 위한 특정 정책들의 선택 및 적용은, 메시지에 대한 미리 결정된 엔드-투-엔드 프로세싱 흐름과 함께 엔드-투-엔드 흐름 내의 메시지에 대한 현재 프로세싱 포인트의 결정에 의해 결정될 수 있다. 이상에서 소개된 정책 모델들은, 정책 집행 엔진(960)이 메시지의 엔드-투-엔드 프로세싱 흐름의 다양한 포인트들에서 메시지에 적용할 수 있는 메시지 프로세싱 정책들의 세트를 정의할 수 있다. 일부 실시예들에 있어서, 엔드-투-엔드 프로세싱 흐름들을 정의하기 위한 정책 모델들 및 다른 기술들은 정책 템플릿(template)들의 세트를 사용하여 생성될 수 있다. 이러한 템플릿들은 특정 어찌션들에 대응할 수 있으며, 정책 모델 엔드-투-엔드 프로세싱 흐름들을 생성하기 위하여 사용될 수 있다. 예를 들어, 하나 이상의 어찌션 템플릿들이 복사될 수 있으며, 적절한 정책 URI들이 각각의 템플릿 복사본 내에 삽입될 수 있다. 그런 다음, 맞춤화된 템플릿들이 엔드-투-엔드 프로세싱 흐름 동안 실행될 수 있는 정책들을 정의하기 위하여 적절한 정책 모델들에 부가될 수 있다.

[0144] 엔드-투-엔드 프로세싱 흐름 동안 실행될 정책들 및 어찌션들을 정의하는 것에 더하여, 정책 집행 엔진(960)에 의해 집행되는 정책 모델들(및 다른 형태들의 미리 결정된 엔드-투-엔드 프로세싱 흐름들)이 또한 특정 정책들이 수행되거나 또는 수행되지 않을 수 있는 조건들을 정의할 수 있다. 일부 실시예들에 있어서, 정책 모델은 정책 모델 내에서 참조되는 정책들의 각각을 수행하기 위한 조건들을 구현하기 위한 논리적 명령어들의 세트를 포함할 수 있다. 예를 들어, 정책 모델은, 프록시 서버(920)에게 일부 메시지 유형들(예를 들어, SOAP, REST, 또는 URL 메시지들)에 대하여 특정 정책이 반드시 수행되어야만 하지만 다른 메시지 유형들에 대해서는 그렇지 않다는 것을 지시하는 조건들을 포함할 수 있다. 추가적으로, 이상에서 논의된 바와 같이, 정책 모델들은 서비스/애플리케이션 레벨에서 및/또는 일부 경우들에 있어서는 동작/메소드 레벨에서 선택적으로 정책들을 적용할 수 있으며, 따라서, 특정 정책들의 적용은 호출되는 백엔드 웹 서비스(930)뿐만 아니라 서비스(930) 내에서 호출되는 특정 동작들 또는 메소드들에 의존할 수 있다. 다양한 추가적인 실시예들에 있어서, 일부 정책 모델들은, 정책 집행 엔진(960)에게 특정 정책이 일부 사용자들에 대해서 반드시 실행되어야 하지만 다른 사용자들에 대해서는 그렇지 않거나, 일부 클라이언트 디바이스 유형들에 대해서 반드시 실행되어야 하지만 다른 클라이언트 디바이스 유형들에 대해서는 그렇지 않거나, 일부 백엔드 웹 서비스들/애플리케이션들에 대해서 반드시 실행되어야 하지만 다른 백엔드 웹 서비스들/애플리케이션들에 대해서는 그렇지 않거나, 및/또는 메시지와 관련된 임의의 다른 특성들을 지시하는 조건들을 포함할 수 있다.

[0145] 이상의 예들이 예시하는 바와 같이, 본원에서 설명되는 다양한 실시예들은, 메시지의 엔드-투-엔드 프로세싱 흐름의 전체에 걸친 다양하고 상이한 프로세싱 포인트들에서 DMZ 또는 다른 논리적인 또는 물리적인 서브네트워크 내에서, 상이한 보안 정책들 및 다른 통신 관리 정책들이 적용될 수 있는 동적 정책 모델들을 지원할 수 있다. 이러한 동적 정책 모델 프레임워크는 악의적인 외부 컴퓨팅 시스템들로부터의 공격들을 방지하기 위한 추가적인 보안성을 구축 및 구현하기 위하여 사용될 수 있으며, 라스트 마일(last mile) 보안 인프라스트럭처 내에서(예를 들어, 백엔드 웹 서비스들/애플리케이션들(430) 내에서) 가능하지 않거나 또는 바람직하지 않을 수 있었던 추가적인 유형들의 보안 정책들을 구현할 수 있다. 추가적으로, 강건한 인증 및 인가 시스템들이, 토큰 번역 및/또는 싱글-사인-온 액세스 제어 시스템들과 같은 본원에서 설명된 동적 정책 모델을 사용하여 구현될 수 있다. 예를 들어, 클라이언트 디바이스(410)는 사용자명칭/패스워드 또는 다른 사용자 증명서들을 통해 인증할 수 있으며, 미리 결정된 엔드-투-엔드 프로세싱 흐름은, 상이한 유형들의 다양하고 상이한 액세스 토큰들(예를 들어, 커베로스 토큰들, SPNEGO 토큰들, 사용자명칭 토큰들, NTLM 토큰들, SAML 토큰들, 등)을 검색하거나 또는 이를 생성하기 위하여, 내부 네트워크(460) 내의 신뢰되는 인증/인가 서비스들로부터 토큰 검색 및 검증을 수행하는 프록시 서버(420) 내에서 실행될 수 있다. 따라서, 사용자가 유효한 증명서들의 하나의 세트를 제공하고 성공적으로 인증 및 인가된 이후에, 프록시 서버(420) 내의 다양한 정책 모델들이, 사용자에 의해 그 이후에 액세스되

는 다양하고 상이한 백엔드 웹 서비스들/애플리케이션들(430)에 대한 대응하는 토큰 유형을 검색하거나 또는 이를 생성함으로써 싱글-사인-온 액세스 제어 시스템을 구현하기 위하여 사용될 수 있다.

[0146] 본 출원의 일 실시예에 따르면, 프로세싱 유닛 및 통신 유닛을 포함하는 시스템이 제공된다. 이러한 시스템은, 본 발명의 원리들을 수행하기 위한 하드웨어, 소프트웨어, 또는 하드웨어 및 소프트웨어의 조합에 의해 구현될 수 있다. 프로세싱 유닛 및 통신 유닛은 도 3에 예시된 컴포넌트들과 같은 이상에서 설명된 컴포넌트들에 의해 구현될 수 있다는 것이 당업자들에게 이해될 것이다. 동시에, 프로세싱 유닛 및 통신 유닛이 이상에서 설명된 바와 같은 본 발명의 원리들을 구현하기 위하여 결합되거나 또는 서브-유닛들로 분리될 수 있다는 것이 당업자들에게 이해될 것이다. 따라서, 본원의 설명은 본원에서 설명된 기능적 유닛들의 임의의 가능한 조합 또는 분리 또는 추가적인 정의를 지원할 수 있다.

[0147] 이상의 실시예의 일 예에 있어서, 프로세싱 유닛 및 통신 유닛을 다음의 동작들을 수행하기 위하여 협업할 수 있다: 외부 컴퓨터 네트워크 내의 클라이언트 디바이스로부터 웹 서비스 요청을 수신하는 동작으로서, 시스템은 외부 컴퓨터 네트워크로부터 분리된 내부 컴퓨터 네트워크의 서브네트워크 내에서 동작하도록 구성되는, 동작; 웹 서비스 요청 내의 제 1 자원을 식별하는 동작; 제 1 자원이 시스템 내에서 제 1 표현 상태 전환(REST) 웹 서비스에 의해 노출된다는 것을 결정하는 동작; 시스템 내에서 제 1 REST 웹 서비스를 호출하는 동작; 및 시스템 내에서의 제 1 REST 웹 서비스의 실행 동안, 내부 컴퓨터 네트워크 내의 컴퓨터 서버 내에서 제 2 웹 서비스를 호출하는 동작.

[0148] 다른 예에 있어서, 프로세싱 유닛 및 통신 유닛은 다음의 동작들을 수행함으로써 REST 웹 서비스를 호출하기 위하여 협업할 수 있다: 제 1 REST 웹 서비스에 의해 노출된 제 1 자원이 프로세싱 유닛과 연관된 메모리 내에 존재하지 않는다는 것을 결정하는 동작; 및 메모리 내에 제 1 자원을 생성하는 동작으로서, 제 1 자원은 클라이언트 디바이스로부터 웹 서비스 요청이 수신된 이후에 생성되는, 동작.

[0149] 다른 예에 있어서, 프로세싱 유닛 및 통신 유닛은 다음의 동작들을 추가로 수행하기 위하여 협업할 수 있다: 내부 컴퓨터 네트워크 내의 컴퓨터 서버 내에서 제 2 웹 서비스에 의해 제공되는 자원들의 세트를 설명하는 웹 애플리케이션 기술 언어(WADL) 파일을 액세스하는 동작; 및 제 2 웹 서비스에 의해 제공되는 자원들의 세트의 WADL 파일 내의 설명을 사용하여 제 1 REST 웹 서비스 내의 하나 이상의 자원들을 생성하는 동작.

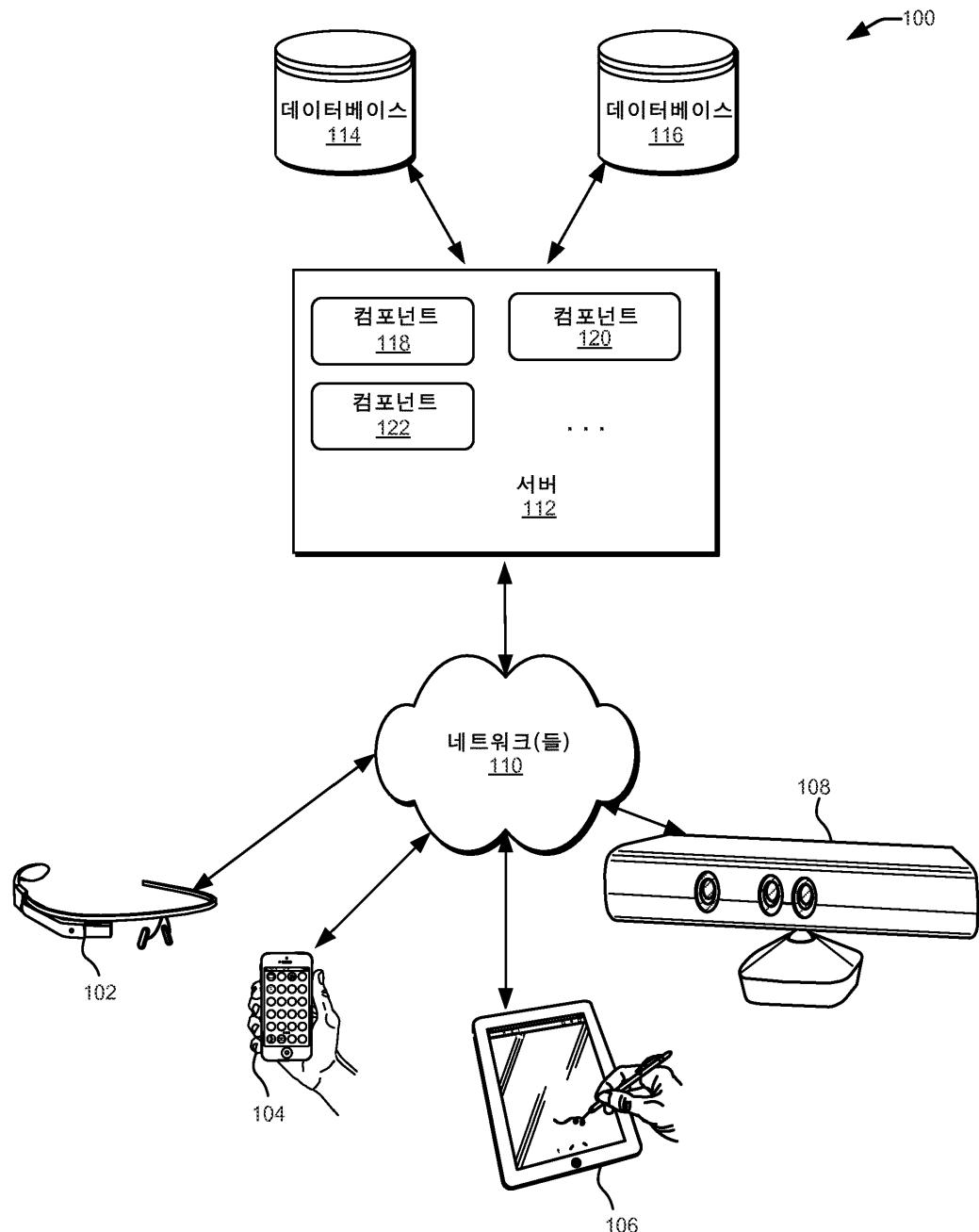
[0150] 또 다른 예에 있어서, 프로세싱 유닛 및 통신 유닛은 다음의 동작들을 수행함으로써 시스템 내에서 제 1 REST 웹 서비스 내의 자원들을 생성하기 위하여 협업할 수 있다: WADL 파일 내의 하나 이상의 자원 설명들을 수정하는 동작; 수정된 자원 설명들에 기초하여 하나 이상의 REST 자원들을 생성하는 동작; 및 시스템 내에서 제 1 REST 웹 서비스 내의 REST 자원들의 각각을 배포하는 동작.

[0151] 이상의 설명에 있어서, 예시의 목적들을 위하여, 방법들이 특정한 순서로 설명되었다. 대안적인 실시예들에 있어서, 방법들은 설명된 것과는 상이한 순서로 수행될 수 있다는 것이 이해되어야만 한다. 이상에서 설명된 방법들은 하드웨어 컴포넌트들에 위해 수행될 수 있거나 또는, 범용 또는 전용 프로세서 또는 로직 회로들과 같은 기계가 방법들을 수행하기 위한 명령어들로 프로그래밍되게끔 하기 위하여 사용될 수 있는 기계-실행가능 명령어들의 시퀀스들로 구현될 수 있다. 이러한 기계-실행가능 명령어들은 하나 이상의 기계 판독가능 매체들 또는 메모리 디바이스들, 예컨대 CD-ROM들, 또는 다른 유형의 광 디스크들, 플로피 디스켓들, ROM들, RAM들, EPROM들, EEPROM들, 자기 또는 광 카드들, 플래시 메모리, 또는 전자적 명령어들을 저장하기에 적절한 다른 유형들의 기계-판독가능 매체들 또는 메모리 디바이스들 상에 저장될 수 있다. 대안적으로, 방법들은 하드웨어 및 소프트웨어의 조합에 의해 수행될 수 있다.

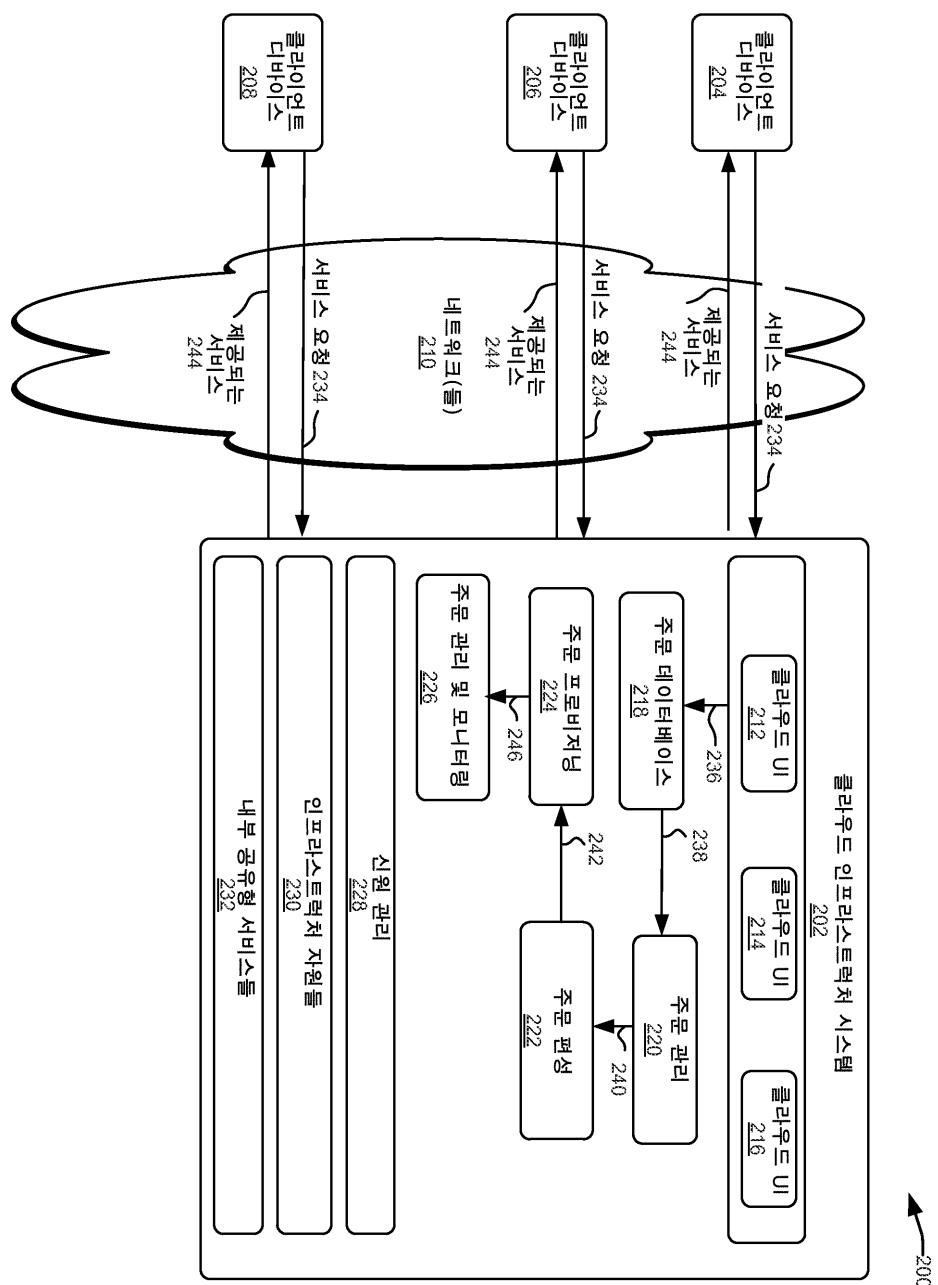
[0152] 본 발명의 예시적이고 지금 선호되는 실시예들이 본원에서 상세하게 설명되었지만, 발명적인 개념들이 달리 다양하게 구현되고 이용될 수 있으며, 첨부된 청구항들이 종래 기술에 의해 제한되는 것을 제외하고는 이러한 변형예들을 포함하는 것으로 해석되도록 의도된다는 것이 이해될 것이다.

도면

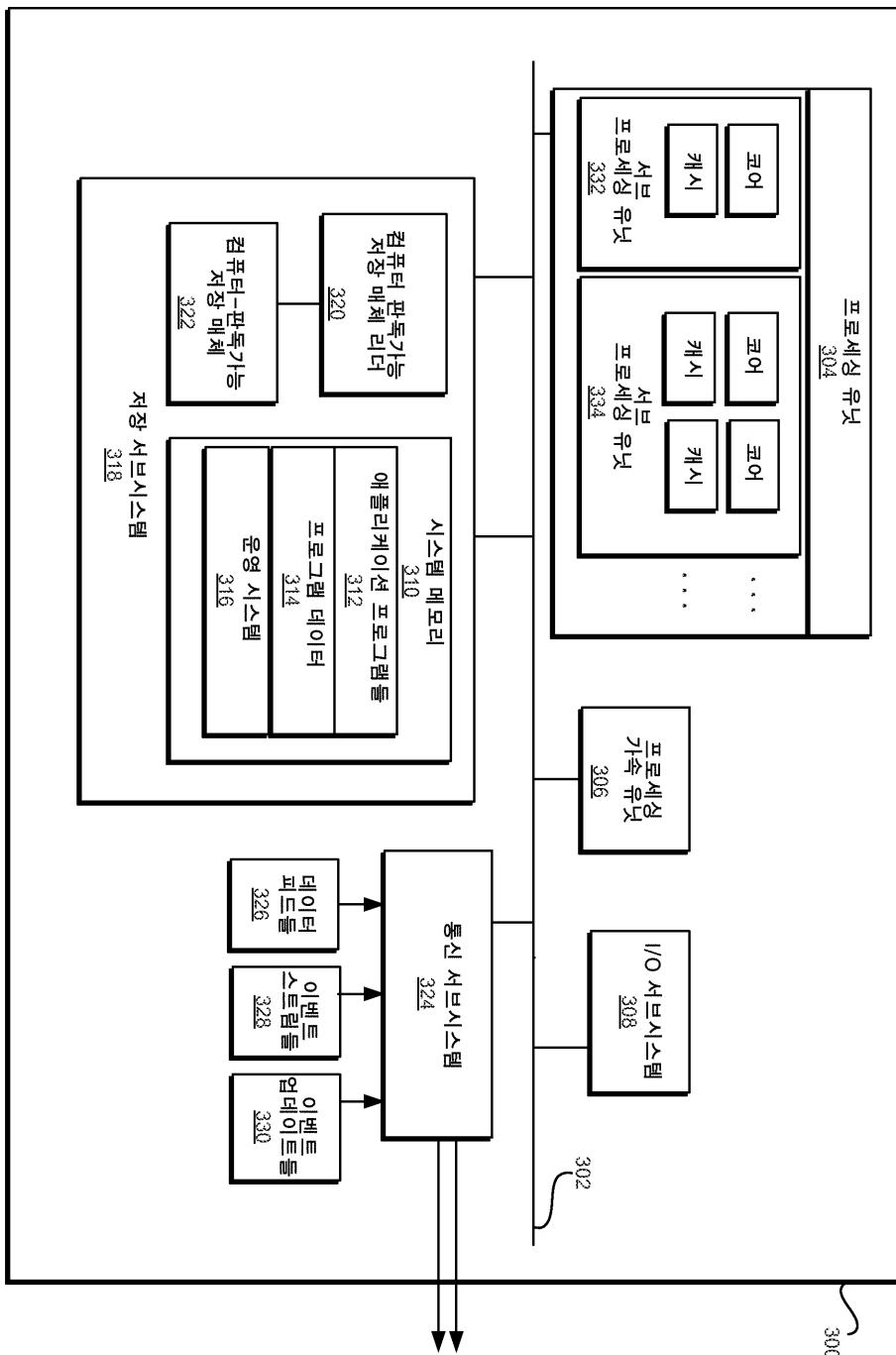
도면1



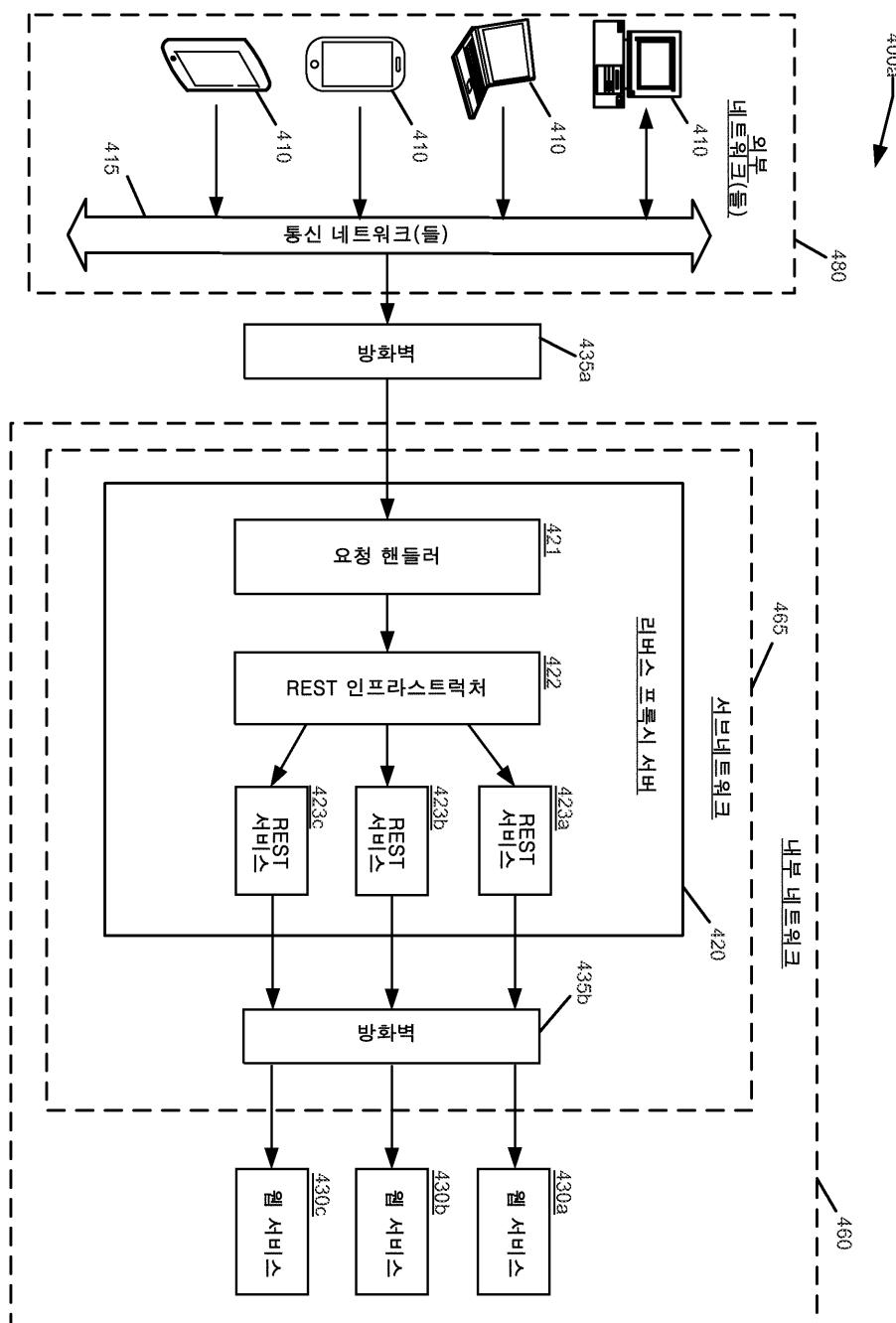
도면2



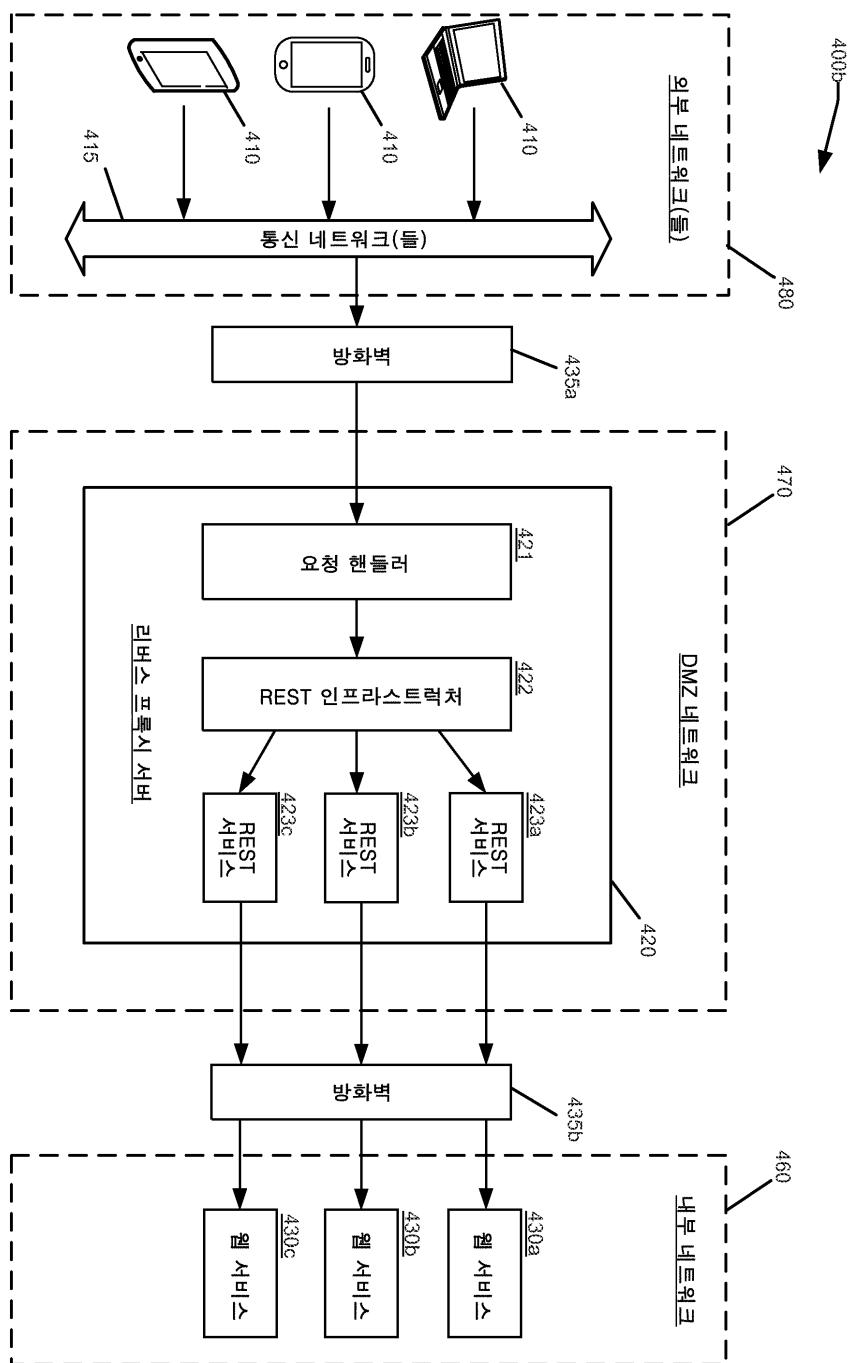
도면3



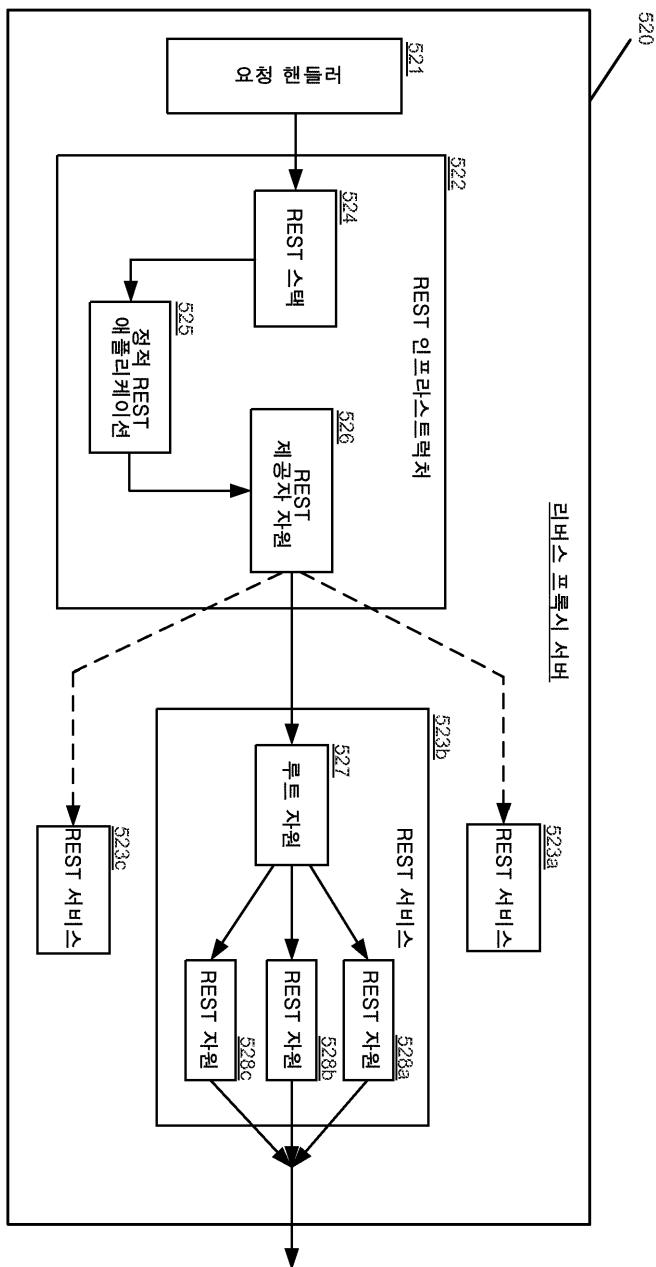
도면4a



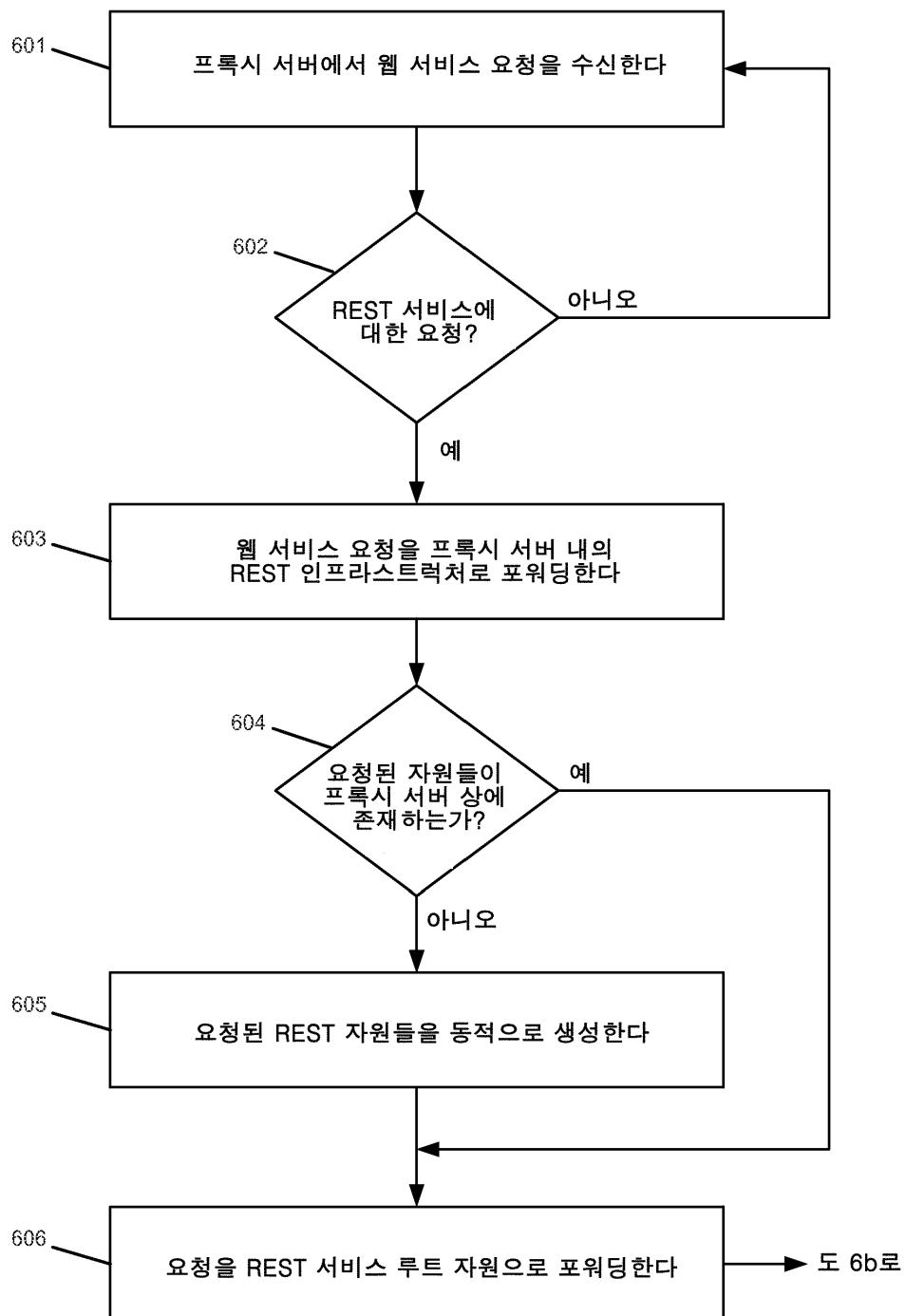
도면4b



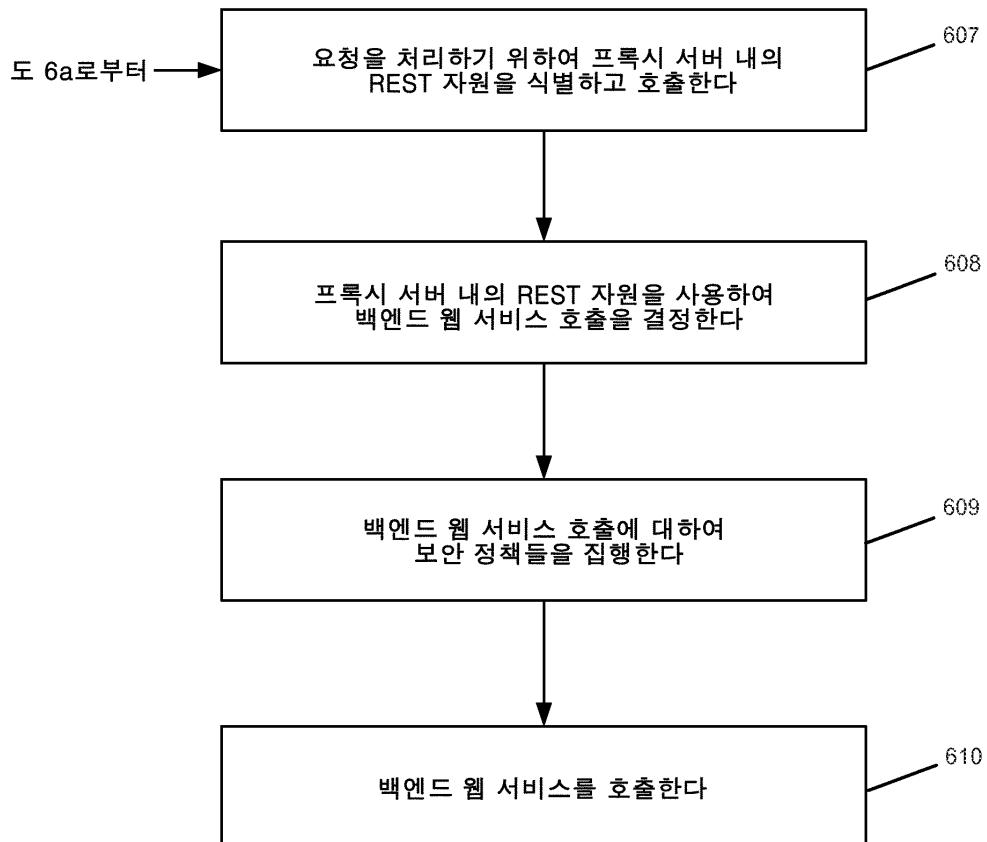
도면5



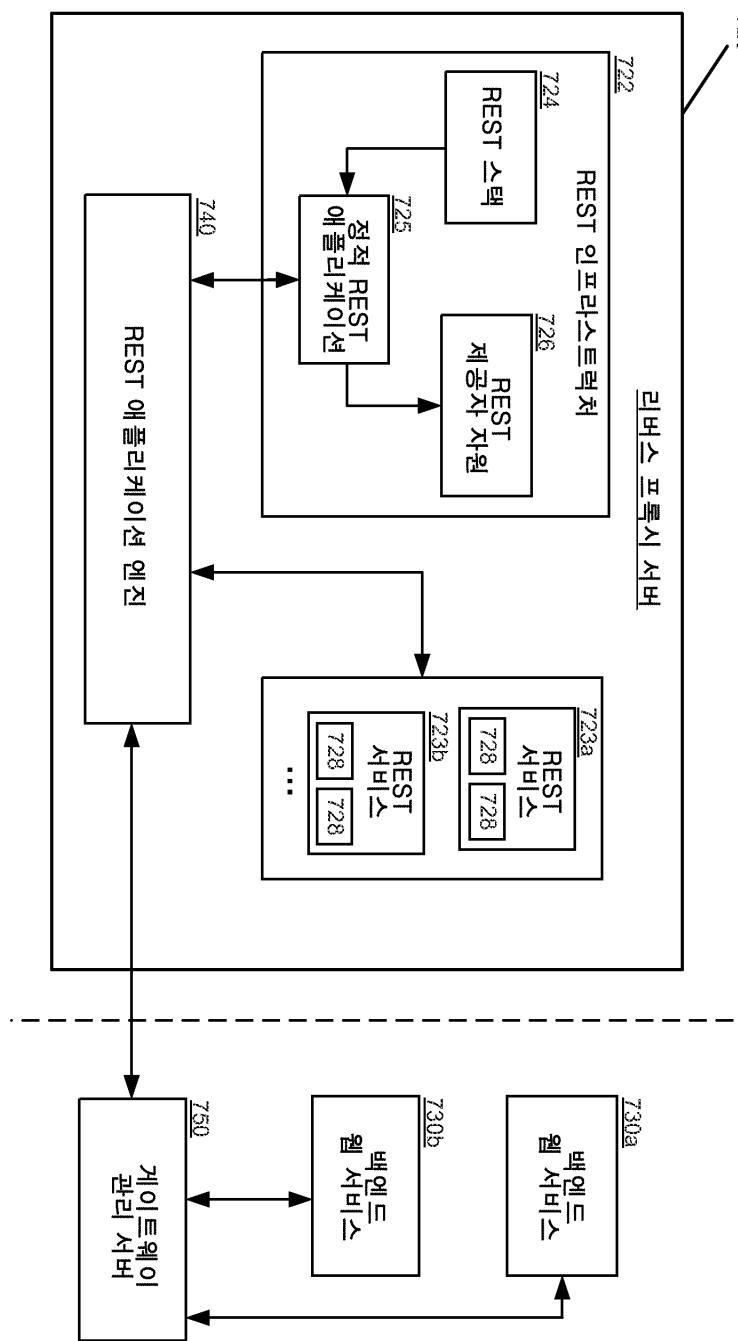
도면6a



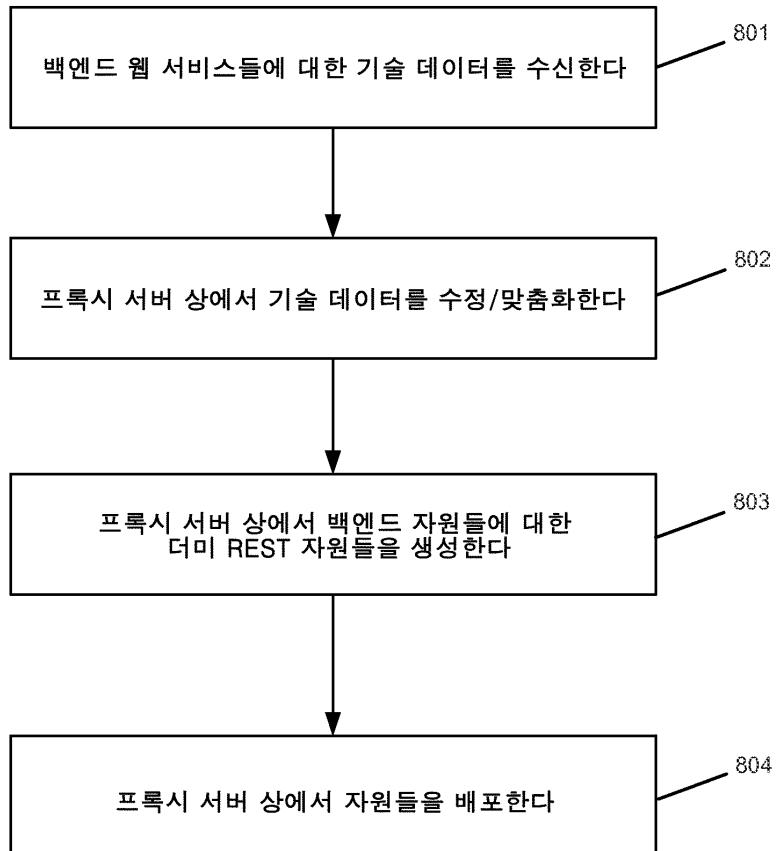
도면6b



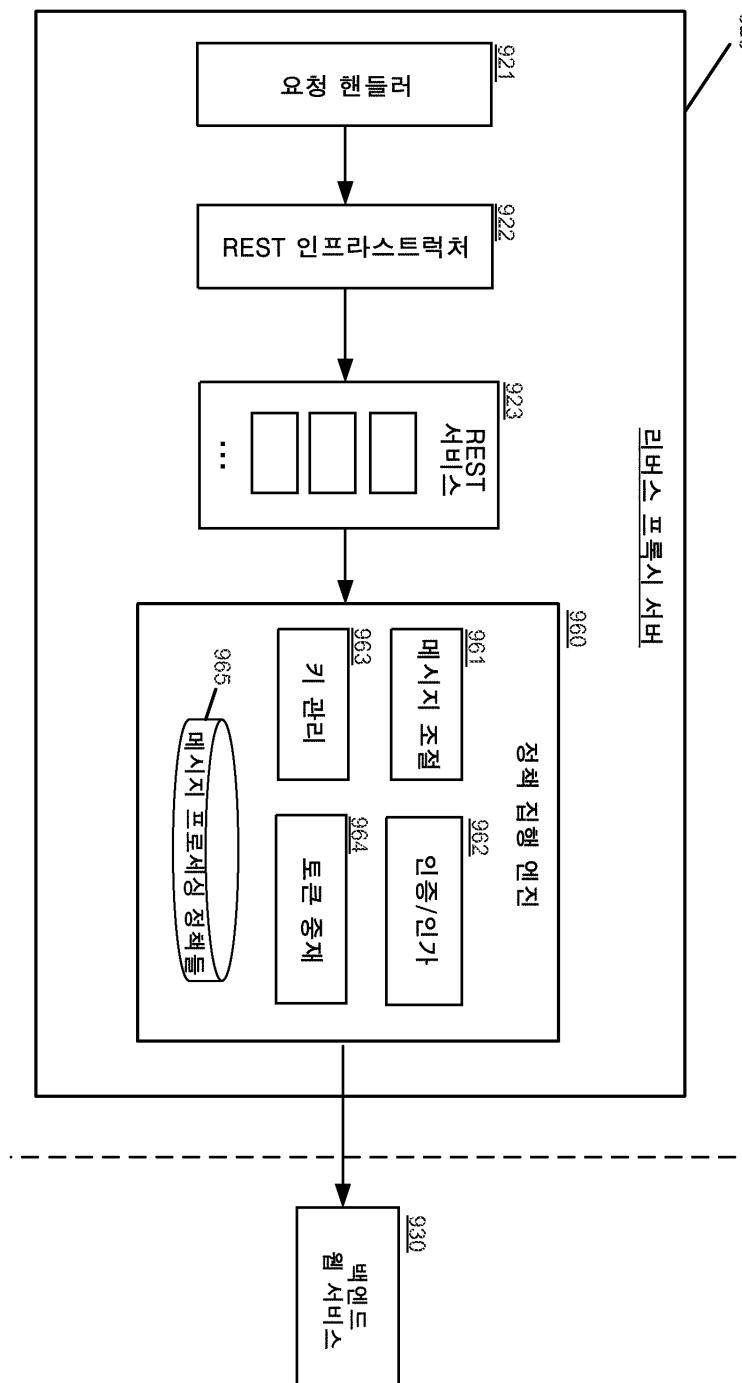
도면7



도면8



도면9



도면10

