



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2016년03월10일

(11) 등록번호 10-1602099

(24) 등록일자 2016년03월03일

(51) 국제특허분류(Int. Cl.)

H04L 12/24 (2006.01)

(52) CPC특허분류

H04L 41/0226 (2013.01)

H04L 41/0213 (2013.01)

(21) 출원번호 10-2015-0014818

(22) 출원일자 2015년01월30일

심사청구일자 2015년01월30일

(56) 선행기술조사문헌

KR101157039 B1*

KR1020140121534 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

충남대학교산학협력단

대전광역시 유성구 대학로 99 (궁동, 충남대학교)

(72) 발명자

이규철

대전 유성구 배울2로 133, 205동 203호 (용산동, 경남아너스빌2단지)

임형준

대전광역시 대덕구 계족산로 136, 502동904호(송촌동, 선비마을5단지아파트)

(뒷면에 계속)

(74) 대리인

특허법인태동

전체 청구항 수 : 총 13 항

심사관 : 김종권

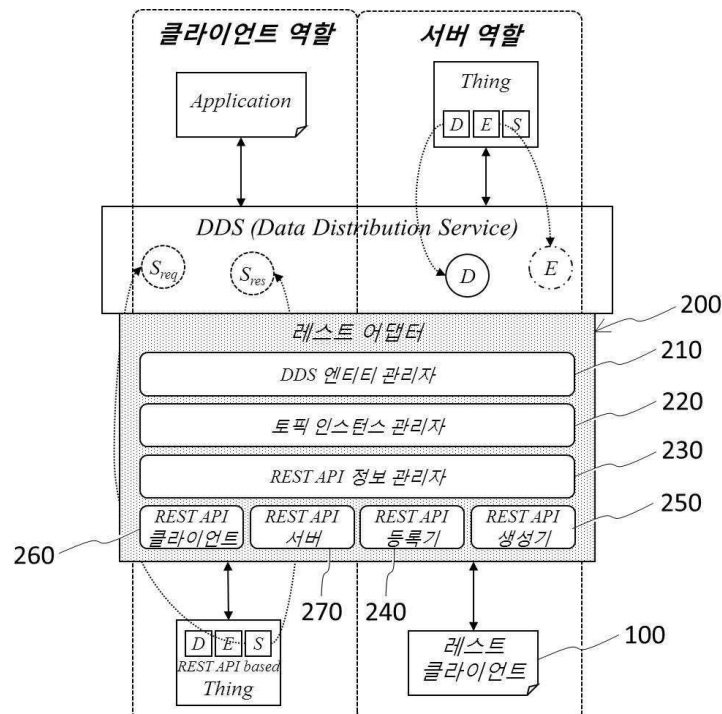
(54) 발명의 명칭 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법

(57) 요약

본 발명은 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법에 관한 것으로서, 발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS)를 기반으로 하여 사물 인터넷의 디바이스 서비스와 외부 서비스를 연동하기 위한 시스템에 있어서, 사물 인터넷의 디바이스 서비스와 관련한 레스트 API(Application

(뒷면에 계속)

대표도 - 도3



Program Interface)를 동기/비동기 방식으로 호출하는 레스트 클라이언트; 및 사물 인터넷의 사물에서 외부 서비스 중 레스트 API를 호출하는 경우에, 상기 레스트 API(Application Program Interface)를 서비스 토픽의 형태에 맞게 타입 변환하여 DDS 내에 발간하고, 상기 레스트 클라이언트의 호출에 따라 DDS 내에서 서비스 토픽으로 명세된 디바이스 서비스를 구독하여 레스트 API 속성에 맞게 변환하여 제공하는 레스트 어댑터를 포함한다. 따라서, 본 발명은 레스트 어댑터에서 DDS 기반의 서비스 토픽을 이용하여 디바이스 서비스와 외부 서비스의 양방향 서비스 연동을 지원할 수 있고, DDS 기반의 토픽을 이용하여 서비스를 일관적으로 표현하여 레스트 어댑터를 통해 상호 운용성을 제공할 수 있으며, 기존의 HTTP 프로토콜을 이용하여 데이터를 전달하기 때문에 부수적인 레이아웃이 필요하지 않으며, HTTP의 기본 메소드를 이용하여 일관된 레스트 API 오퍼레이션(Operations)으로 제어할 수 있다.

(72) 발명자

백문기

충북 제천시 용두대로7길 25, 107동 602호 (하소동, 그린코아루아파트)

김인아

경기 안산시 상록구 호동로1길 15, 102호 (일동, 아산빌라)

이 발명을 지원한 국가연구개발사업

과제고유번호 2012R1A1A4A01015696

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 일반연구자지원사업 지역대학우수과학자

연구과제명 사물 인터넷의 이질적 데이터/이벤트/서비스 통합을 위한 연동 및 조합 기법 개발

기 여 율 1/1

주관기관 충남대학교 산학협력단

연구기간 2012.05.01 ~ 2015.04.30

공지예외적용 : 있음

명세서

청구범위

청구항 1

발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS)를 기반으로 하여 사물 인터넷의 디바이스 서비스와 외부 서비스를 연동하기 위한 시스템에 있어서,

사물 인터넷의 디바이스 서비스와 관련한 레스트 API(Application Program Interface)를 동기/비동기 방식으로 호출하는 레스트 클라이언트; 및

사물 인터넷의 사물에서 외부 서비스 중 레스트 API를 호출하는 경우에, 상기 레스트 API(Application Program Interface)를 서비스 토픽의 형태에 맞게 타입 변환하여 DDS 내에 발간하고, 상기 레스트 클라이언트의 호출에 따라 DDS 내에서 서비스 토픽으로 명세된 디바이스 서비스를 구독하여 레스트 API 속성에 맞게 변환하여 제공하는 레스트 어댑터를 포함하고,

상기 서비스 토픽은 서비스를 요청하는 서비스 요청 토픽과 서비스 요청에 따른 결과값을 알려주는 서비스 응답 토픽으로 구분하여 명세하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 시스템.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 서비스 요청 토픽은 서비스 호출을 위한 서비스 요청 ID, 서비스 이름, 인자값 이름, 인자값을 포함하고, 상기 서비스 응답 토픽은 서비스 요청한 서비스 요청 ID, 서비스 ID, 서비스 이름, 결과값 이름, 결과값을 포함하며,

상기 서비스 요청 토픽은 상기 서비스 요청 ID와 서비스 이름을 조합하여 키를 구성하고, 상기 서비스 응답 토픽은 서비스 ID와 서비스 이름을 조합하여 키를 구성하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 시스템.

청구항 4

제3항에 있어서,

상기 레스트 API는 주소, 이름, 파라미터를 포함하고,

상기 레스트 API의 주소는 상기 레스트 API를 리소스 형태로 등록하고, 상기 리소스에 체계적인 URI를 부여하며,

상기 레스트 API의 이름은 서비스 토픽의 서비스 이름을 이용하며,

상기 파라미터는 서비스 토픽의 인자값을 이용하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 시스템.

청구항 5

제1항에 있어서,

상기 레스트 어댑터는,

도메인 참여자, 발간자, 구독자, 데이터라이터, 데이터리더를 생성하는 DDS 엔티티를 관리하며, 서비스 요청/응답 기능을 수행하는 DDS 엔티티 관리자;

DDS 내에서 전달되는 서비스 토픽의 정보를 레스트 API로 타입 변환하고, 외부 서비스에서 전달되는 레스트 API의 정보를 서비스 토픽으로 타입 변환하는 토픽 인스턴스 관리자;

상기 레스트 API의 속성들을 저장하고, 상기 레스트 API의 정보에 대한 등록, 읽기, 수정, 삭제를 포함하는 관리 기능을 수행하는 레스트 API 정보 관리자;

상기 레스트 API를 리소스 형태로 등록하는 레스트 API 등록기;

상기 서비스 토픽으로 명세된 디바이스 서비스를 레스트 API로 생성하는 레스트 API 생성기;

상기 서비스 토픽을 이용하여 웹 상에 존재하는 레스트 API를 호출하도록 클라이언트 역할을 수행하는 레스트 API 클라이언트 모듈; 및

상기 레스트 클라이언트의 호출 요청을 처리하여 상기 레스트 어댑터 내에 존재하는 레스트 API를 호출하도록 서버 역할을 수행하는 레스트 API 서버 모듈을 포함하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 시스템.

청구항 6

제1항에 있어서,

상기 레스트 어댑터는 HTTP 메소드를 통해 레스트 API를 호출하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 시스템.

청구항 7

발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS)를 기반으로 하여 사물 인터넷의 디바이스 서비스와 외부 서비스를 연동하는 방법에 있어서,

상기 디바이스 서비스를 레스트(REST) 기반의 외부 서비스와 연동되도록 레스트 API(Application Program Interface)와 DDS 기반의 토픽으로 명세된 디바이스 서비스를 상호 변환하는 레스트 어댑터를 설계하는 설계 단계;

레스트 기반의 사물이 웹상의 레스트 API를 호출하는 경우에, 상기 레스트 API의 속성들을 저장하고, 상기 레스트 API를 서비스 토픽의 형태에 맞게 타입 변환하여 발간하는 클라이언트 역할 실행 단계; 및

외부의 레스트 클라이언트가 상기 디바이스 서비스를 동기/비동기 방식으로 호출하는 경우에, DDS 내에서 서비스 토픽으로 명세된 디바이스 서비스를 구독하여 레스트 API 속성에 맞게 변환한 후 상기 변환한 레스트 API를 리소스로 등록하는 서버 역할 실행 단계를 포함하고,

상기 서비스 토픽은 서비스를 요청하는 서비스 요청 토픽과 서비스 요청에 따른 결과값을 알려주는 서비스 응답 토픽으로 구분하여 명세하고, 키로 구분되는 인스턴스와 전송 단위인 샘플을 통해 송수신되는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 8

삭제

청구항 9

제7항에 있어서,

상기 서비스 요청 토픽은 서비스 호출을 위한 서비스 요청 ID, 서비스 이름, 인자값 이름, 인자값을 포함하고, 상기 서비스 응답 토픽은 서비스 요청한 서비스 요청 ID, 서비스 ID, 서비스 이름, 결과값 이름, 결과값을 포함

하며,

상기 서비스 요청 토픽은 상기 서비스 요청 ID와 서비스 이름을 조합하여 키를 구성하고, 상기 서비스 응답 토픽은 서비스 ID와 서비스 이름을 조합하여 키를 구성하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 10

제9항에 있어서,

상기 레스트 API는 주소, 이름, 파라미터를 포함하고,

상기 레스트 API의 주소는 상기 레스트 API를 리소스 형태로 등록하고, 상기 리소스에 체계적인 URI를 부여하며,

상기 레스트 API의 이름은 서비스 토픽의 서비스 이름을 이용하며,

상기 파라미터는 서비스 토픽의 인자값을 이용하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 11

제10항에 있어서,

상기 디바이스 서비스와 레스트 API를 명세한 서비스 토픽은 동일한 데이터 타입을 가지고, 상기 서비스 ID와 서비스 이름을 조합한 키가 구성된 다수의 인스턴스에 의해 구분되는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 12

제11항에 있어서,

상기 레스트 어댑터는 HTTP 메소드를 통해 레스트 API를 호출하고, 상기 레스트 클라이언트는 레스트 API의 주소를 이용해 접근하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 13

제7항에 있어서,

상기 서버 역할 실행 단계는,

상기 레스트 API를 리소스로 등록하기 이전에 해당 레스트 API의 등록 여부를 확인하여 중복 저장되지 않도록 하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 14

제9항에 있어서,

상기 서버 역할 실행 단계는,

상기 레스트 클라이언트가 상기 레스트 어댑터에 동기 방식으로 서비스 요청한 경우에, 상기 레스트 어댑터는 DDS 내에서 해당 서비스를 서비스 요청 토픽에 작성하여 발간하고, 상기 서비스 요청의 처리 결과를 서비스 응답 토픽으로 구독하여 제공하는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

청구항 15

제12항에 있어서,

상기 서버 역할 실행 단계는,

상기 레스트 클라이언트가 비동기 방식으로 서비스를 요청한 경우에, 상기 레스트 어댑터는 DDS 내에 요청한 서비스를 전달하고, 상기 서비스 요청의 처리 결과에 대한 응답 메시지가 없거나 HTTP 상태 코드를 전달받는 것을 특징으로 하는 사물 인터넷에서 레스트 기반의 서비스 연동 방법.

발명의 설명

기술 분야

[0001] 본 발명은 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법에 관한 것으로, 보다 상세하게는 사물인터넷의 디바이스 서비스와 외부 서비스의 레스트 API를 양방향으로 연동할 수 있는 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법에 관한 것이다.

배경 기술

[0002] 사물인터넷(Internet of Things, IoT)은 유비쿼터스 센서 네트워크(Ubiquitous Sensor Network, USN)나 사물지능통신(Machine-to-Machine, M2M)에서 발전하여 시간과 장소의 제약 없이 사물이 인터넷에 연결되는 환경을 의미한다.

[0003] 이러한 사물인터넷은 물리적 또는 가상의 사물(Things)을 인터넷으로 연결하여 센싱(sensing), 액추에이팅(actuating), 데이터 수집, 저장, 처리 등의 서비스를 제공하는 인프라로 정의될 수 있다.

[0004] 사물인터넷은 사람뿐만 아니라 사물들이 통신을 통해 정보를 주고받고, 이를 활용한 다양한 서비스가 가능하다. 이를 위해 동적으로 수많은 사물이 네트워크에 연결되는 확장성(scalability), 사물인터넷을 구성하는 네트워크 프로토콜, 데이터 형식, 서비스 호출 방식이 서로 다른 이질성(heterogeneity)에 대한 상호운용성(interoperability), 무선 환경 사물의 이동성(mobility), 실시간 통신, 신뢰성 기반의 인식 및 적응성(awareness & adaptability) 등의 특징을 지원해야 한다.

[0005] 사물은 센서와 액추에이터 기능을 담당하는 모듈과 2G/3G, 이더넷, WiFi, 지그비(Zigbee), RFID 등의 통신모듈, 파워모듈로 구성된다. 사물이 주고받는 정보는 센싱을 통해 얻은 측정값에서부터 특정 기능을 제어하는 것에 따라 데이터, 이벤트, 서비스로 나뉘어진다. 데이터는 사물에서 단순히 수집하여 전달하는 측정값을 의미하며, 이벤트는 특정 사건이 발생하는 것을 의미한다. 서비스의 경우에는 디바이스의 기능을 제어하는 디바이스 서비스와 정보 시스템이나 클라우드 서비스에서 제공하는 외부 서비스로 구분된다. 이에 따라 데이터, 이벤트 및 서비스를 유기적으로 모두 연계할 수 있어야 한다.

[0006] 한편, 레스트(Representational State Transfer, REST)는 웹을 이용하여 분산 시스템을 구축하기 위한 소프트웨어 프로토콜의 구조의 한 형식이다. 이러한 레스트는 네트워크 애플리케이션 개발에 사용되는 인터페이스 스타일로 복잡한 프로토콜 대신에 간단하고 명료한 HTTP 프로토콜을 이용한다. 레스트는 리소스를 구별할 수 있는 식별자와 리소스를 수행하기 위한 CRUD(CREATE, RETRIEVE, UPDATE, DELETE) 문법, 리소스 실행 결과에 대한 상태 코드, 다른 리소스를 연결할 수 있는 하이퍼미디어 링크를 통해 사용할 수 있다.

[0007] 레스트의 CRUD 연산은 HTTP 프로토콜을 통해 사용하게 되므로 표 1과 같이 HTTP 메소드와 매핑해야 한다.

표 1

CRUD	HTTP	Description
CREATE	PUT/	PUT: Creates a resource
	POST	POST: Inserts a resource
RETRIEVE	GET	Retrieves a resource
UPDATE	PUT/	PUT: Updates or replaces a resource
	POST	POST: Updates, or extends a resource; may change the state of other resource
DELETE	DELETE	Deletes a resource

[0008]

[0009]

CREATE는 PUT을 통해 리소스를 생성하거나 POST를 이용하여 새로운 리소스를 삽입할 수 있는데, PUT의 경우에는 그래픽 디자이너가 포토샵 프로그램에 동일한 이미지를 계속 추가하는 것과 같이 사용하고, POST는 편집자가 제목, 표지, 목차 등의 템플릿을 이용하여 이미지를 생성하는 것과 같이 사용한다. RETRIEVE와 DELETE는 GET과 DELETE로 각각 매핑할 수 있고, UPDATE의 경우에는 CREATE와 같이 PUT이나 POST로 표현할 수 있다.

[0010]

비특허논문1에서는 레스트와 사물인터넷의 디바이스 서비스를 연동하기 위해, 센서, 액츄에이터 등의 사물인터넷 디바이스 서비스를 REST API(Application Program Interface)로 변환하여 제공하며, 비특허논문2에서는 블루투스, Zigbee 디바이스를 REST API로 변환하여 외부에서 디바이스 서비스를 접근할 수 있도록 제공한다.

[0011]

그러나, 비특허논문1 및 비특허논문2는 외부의 REST API를 디바이스 서비스로 변환하여 제공하지 않기 때문에 양방향 서비스 연동을 지원하지 못할 뿐만 아니라 디바이스의 프로토콜마다 드라이버나 번들이 필요하다는 문제점이 있다.

선행기술문헌

비특허문헌

[0012]

(비특허문헌 0001) 비특허문헌1 : Qiang Li, Weijun Qin, Bing Han, Ruicong Wang, Limin Sun, "A Case Study on REST-Style Architecture for Cyber-Physical Systems: Restful Smart Gateway," Journal of Computer Science & Information Systems, vol.8, Issue 4, pp.1317-1329, Oct. 2011.

(비특허문헌 0002) 비특허문헌2 : Architecting the Internet of Things, "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices," pp.97-129, 2011.

발명의 내용

해결하려는 과제

[0013]

본 발명은 DDS 기반의 서비스 토픽을 이용하여 사물 인터넷의 디바이스 서비스는 레스트 API로 변환되어 레스트 클라이언트가 호출할 수 있도록 하고, 외부 서비스의 레스트 API는 서비스 토픽으로 변환되어 사물 인터넷에서 호출하여 서비스를 제공할 수 있도록 하는 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법을 제공한다.

과제의 해결 수단

[0014]

실시예들 중에서, 사물인터넷에서 레스트 기반의 서비스 연동 시스템은, 발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS)를 기반으로 하여 사물 인터넷의 디바이스 서비스와 외부 서비스를 연동하기 위한 시스템에 있어서, 사물 인터넷의 디바이스 서비스와 관련한 레스트 API(Application Program Interface)를

동기/비동기 방식으로 호출하는 레스트 클라이언트; 및 사물 인터넷의 사물에서 외부 서비스 중 레스트 API를 호출하는 경우에, 상기 레스트 API(Application Program Interface)를 서비스 토픽의 형태에 맞게 타입 변환하여 DDS 내에 발간하고, 상기 레스트 클라이언트의 호출에 따라 DDS 내에서 서비스 토픽으로 명세된 디바이스 서비스를 구독하여 레스트 API 속성에 맞게 변환하여 제공하는 레스트 어댑터를 포함하는 것을 특징으로 한다.

[0015] 상기 서비스 토픽은 서비스를 요청하는 서비스 요청 토픽과 서비스 요청에 따른 결과값을 알려주는 서비스 응답 토픽으로 구분하여 명세하는 것을 특징으로 한다.

[0016] 상기 서비스 요청 토픽은 서비스 호출을 위한 서비스 요청 ID, 서비스 이름, 인자값 이름, 인자값을 포함하고, 상기 서비스 응답 토픽은 서비스 요청한 서비스 요청 ID, 서비스 ID, 서비스 이름, 결과값 이름, 결과값을 포함하며, 상기 서비스 요청 토픽은 상기 서비스 요청 ID와 서비스 이름을 조합하여 키를 구성하고, 상기 서비스 응답 토픽은 서비스 ID와 서비스 이름을 조합하여 키를 구성하는 것을 특징으로 한다.

[0017] 상기 레스트 API는 주소, 이름, 파라미터를 포함하고, 상기 레스트 API의 주소는 상기 레스트 API를 리소스 형태로 등록하고, 상기 리소스에 체계적인 URI를 부여하며, 상기 레스트 API의 이름은 서비스 토픽의 서비스 이름을 이용하여, 상기 파라미터는 서비스 토픽의 인자값을 이용하는 것을 특징으로 한다.

[0018] 상기 레스트 어댑터는, 도메인 참여자, 발간자, 구독자, 데이터라이터, 데이터리더를 생성하는 DDS 엔티티를 관리하며, 서비스 요청/응답 기능을 수행하는 DDS 엔티티 관리자; DDS 내에서 전달되는 서비스 토픽의 정보를 레스트 API로 타입 변환하고, 외부 서비스에서 전달되는 레스트 API의 정보를 서비스 토픽으로 타입 변환하는 토픽 인스턴스 관리자; 상기 레스트 API의 속성들을 저장하고, 상기 레스트 API의 정보에 대한 등록, 읽기, 수정, 삭제를 포함하는 관리 기능을 수행하는 레스트 API 정보 관리자; 상기 레스트 API를 리소스 형태로 등록하는 레스트 API 등록기; 상기 서비스 토픽으로 명세된 디바이스 서비스를 레스트 API로 생성하는 레스트 API 생성기; 상기 서비스 토픽을 이용하여 웹 상에 존재하는 레스트 API를 호출하도록 클라이언트 역할을 수행하는 레스트 API 클라이언트 모듈; 및 상기 레스트 클라이언트의 호출 요청을 처리하여 상기 레스트 어댑터 내에 존재하는 레스트 API를 호출하도록 서버 역할을 수행하는 레스트 API 서버 모듈을 포함하는 것을 특징으로 한다.

[0019] 상기 레스트 어댑터는 HTTP 메소드를 통해 레스트 API를 호출하는 것을 특징으로 한다.

[0020] 실시예들 중에서, 사물인터넷에서 레스트 기반의 서비스 연동 방법은, 발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS)를 기반으로 하여 사물 인터넷의 디바이스 서비스와 외부 서비스를 연동하는 방법에 있어서, 상기 디바이스 서비스를 레스트(REST) 기반의 외부 서비스와 연동되도록 레스트 API(Application Program Interface)와 DDS 기반의 토픽으로 명세된 디바이스 서비스를 상호 변환하는 레스트 어댑터를 설계하는 설계 단계; 레스트 기반의 사물이 웹상의 레스트 API를 호출하는 경우에, 상기 레스트 API의 속성들을 저장하고, 상기 레스트 API를 서비스 토픽의 형태에 맞게 타입 변환하여 발간하는 클라이언트 역할 실행 단계; 및 외부의 레스트 클라이언트가 상기 디바이스 서비스를 동기/비동기 방식으로 호출하는 경우에, DDS 내에서 서비스 토픽으로 명세된 디바이스 서비스를 구독하여 레스트 API 속성에 맞게 변환한 후 상기 변환한 레스트 API를 리소스로 등록하는 서버 역할 실행 단계를 포함하는 것을 특징으로 한다.

[0021] 상기 서비스 토픽은 서비스를 요청하는 서비스 요청 토픽과 서비스 요청에 따른 결과값을 알려주는 서비스 응답 토픽으로 구분하여 명세하고, 키로 구분되는 인스턴스와 전송 단위인 샘플을 통해 송수신되는 것을 특징으로 한다.

[0022] 상기 서비스 요청 토픽은 서비스 호출을 위한 서비스 요청 ID, 서비스 이름, 인자값 이름, 인자값을 포함하고, 상기 서비스 응답 토픽은 서비스 요청한 서비스 요청 ID, 서비스 ID, 서비스 이름, 결과값 이름, 결과값을 포함하며, 상기 서비스 요청 토픽은 상기 서비스 요청 ID와 서비스 이름을 조합하여 키를 구성하고, 상기 서비스 응답 토픽은 서비스 ID와 서비스 이름을 조합하여 키를 구성하는 것을 특징으로 한다.

[0023] 상기 레스트 API는 주소, 이름, 파라미터를 포함하고, 상기 레스트 API의 주소는 상기 레스트 API를 리소스 형태로 등록하고, 상기 리소스에 체계적인 URI를 부여하며, 상기 레스트 API의 이름은 서비스 토픽의 서비스 이름을 이용하여, 상기 파라미터는 서비스 토픽의 인자값을 이용하는 것을 특징으로 한다.

[0024] 상기 디바이스 서비스와 레스트 API를 명세한 서비스 토픽은 동일한 데이터 타입을 가지고, 상기 서비스 ID와 서비스 이름을 조합한 키가 구성된 다수의 인스턴스에 의해 구분되는 것을 특징으로 한다.

[0025] 상기 레스트 어댑터는 HTTP 메소드를 통해 레스트 API를 호출하고, 상기 레스트 클라이언트는 레스트 API의 주소를 이용해 접근하는 것을 특징으로 한다.

- [0026] 상기 서버 역할 실행 단계는, 상기 레스트 API를 리소스로 등록하기 이전에 해당 레스트 API의 등록 여부를 확인하여 중복 저장되지 않도록 하는 것을 특징으로 한다.
- [0027] 상기 서버 역할 실행 단계는, 상기 레스트 클라이언트가 상기 레스트 어댑터에 동기 방식으로 서비스 요청한 경우에, 상기 레스트 어댑터는 DDS 내에서 해당 서비스를 서비스 요청 토픽에 작성하여 발간하고, 상기 서비스 요청의 처리 결과를 서비스 응답 토픽으로 구독하여 제공하는 것을 특징으로 한다.
- [0028] 상기 서버 역할 실행 단계는, 상기 레스트 클라이언트가 비동기 방식으로 서비스를 요청한 경우에, 상기 레스트 어댑터는 DDS 내에 요청한 서비스를 전달하고, 상기 서비스 요청의 처리 결과에 대한 응답 메시지가 없거나 HTTP 상태 코드를 전달받는 것을 특징으로 한다.

발명의 효과

- [0029] 본 발명의 사물인터넷에서 레스트 기반의 서비스 연동 시스템 및 그 방법은, 레스트 어댑터에서 DDS 기반의 서비스 토픽을 이용하여 사물 인터넷의 디바이스 서비스는 레스트 API로 변환되어 레스트 클라이언트가 호출할 수 있도록 하고, 외부 서비스의 레스트 API는 서비스 토픽으로 변환되어 사물 인터넷에서 호출하여 서비스를 제공할 수 있도록 하여 양방향 서비스 연동을 지원할 수 있는 효과가 있다.
- [0030] 또한, 본 발명은 통신에 참여하는 응용 프로그램의 위치나 존재에 상관없이 데이터를 실시간 전송할 수 있도록 복잡한 네트워크 프로그래밍을 단순화하고, DDS 기반의 토픽을 이용하여 서비스를 일관적으로 표현하여 레스트 어댑터를 통해 상호 운용성을 제공할 수 있으며, 기존의 HTTP 프로토콜을 이용하여 데이터를 전달하기 때문에 부수적인 레이어가 필요하지 않으며, HTTP의 기본 메소드를 이용하여 일관된 레스트 API 오퍼레이션(Operations)으로 제어할 수 있는 효과가 있다.

도면의 간단한 설명

- [0031] 도 1은 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반으로 실시간 데이터, 이벤트 및 서비스를 통합 연동하는 미들웨어를 설명하는 도면이다.
- 도 2는 도 1의 DDS 토픽을 설명하는 도면이다.
- 도 3은 본 발명의 일 실시예에 따른 사물 인터넷에서 레스트 기반의 서비스 연동 시스템을 설명하는 도면이다.
- 도 4는 본 발명의 일 실시예에 따른 사물 인터넷에서 레스트 기반의 서비스 연동 방법을 설명하는 순서도이다.
- 도 5는 도 4의 레스트 어댑터에 대한 클라이언트 역할을 실행하는 상태를 설명하는 도면이다.
- 도 6은 도 4의 레스트 어댑터에 대한 동기/비동기 방식의 클라이언트 역할 실행 과정을 설명하는 도면이다.
- 도 7은 도 4의 레스트 어댑터에 대한 서버 역할을 실행하는 상태를 설명하는 도면이다.
- 도 8은 도 4의 레스트 어댑터에 대한 동기/비동기 방식의 서버 역할 실행 과정을 설명하는 도면이다.
- 도 9는 사물 인터넷의 디바이스 서비스를 레스트 API로 변환하는 과정을 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0032] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0033] 한편, 본 발명에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0034] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2

구성요소도 제1 구성요소로 명명될 수 있다.

- [0035] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0036] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0037] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0038] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 발명에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0039] 도 1은 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반으로 실시간 데이터, 이벤트 및 서비스를 통합 연동하는 미들웨어를 설명하는 도면이고, 도 2는 도 1의 DDS 토픽을 설명하는 도면이다.
- [0040] 도 1을 참고하면, 사물 인터넷의 미들웨어는, OMG(Object Management Group) 표준인 DDS(Data Distribution Service)를 기반으로 사물 인터넷의 특징을 제공하면서 DDS 토픽(Topic)을 이용하여 실시간으로 데이터, 이벤트 및 서비스를 통합/ 연동한다.
- [0041] 이와 같이, DDS 토픽으로 표현되는 데이터, 이벤트 및 서비스는 디바이스 서비스 어댑터, 외부 서비스 어댑터, 데이터 어댑터 및 이벤트 어댑터 등의 각 어댑터를 이용하여 상호운용성을 제공한다.
- [0042] 이때, DDS는 발간/구독 모델에 기반하여 분산 환경을 위한 데이터 중심의 프로그램 모델에 대한 표준화의 필요성에 의해 만들어진 실시간 데이터 통신 미들웨어 표준이다. DDS 표준은 DDS API 표준을 규정한 DCPS(Data-Centric Publish-Subscribe)와 네트워크 계층 통신 프로토콜을 규정한 RTPS(Real Time Publish-Subscribe)로 구성되어 있다.
- [0043] DCPS는 발간/구독 모델에 기반한 데이터 교환 기능을 제공하는 인터페이스 규격으로, 도 2에 도시된 바와 같이, 데이터리더(read())/ 데이터라이터(write())의 API를 제공하며, 발간자(publisher)는 전송할 데이터를 생성하고 배포하는 기능을 담당하고, 구독자(subscriber)는 이 데이터를 수신하여 활용하게 된다.
- [0044] RTPS는 발간/구독 모델에 기반한 구현 측면의 데이터 전송 프로토콜로서 UDP/IP와 같은 신뢰성 없는 전송계층 위에서도 동작 가능하도록 설계되어 있다. DDS를 이용한 실시간 통신은 도메인으로 규정한 장치들간에 가상의 데이터 전송 채널인 토픽을 생성하여 발간/구독이 가능하다.
- [0045] DDS 토픽은 DDS 도메인 내에 유일한 이름으로 구분되며, 타입서포트(TypeSupport)를 통해 송수신하는 데이터의 타입을 명세한다.
- [0046] 사물 인터넷의 서비스는 사물이 제공하는 디바이스 서비스와 클라우드 서비스나 정보 시스템에서 제공하는 외부 서비스로 구분할 수 있다. 디바이스 서비스는 사물의 대표적인 네트워크 프로토콜인 지그비, RFID, 블루투스 등에서 제공하는 서비스를 지칭하고, 외부 서비스는 공개 API, 웹서비스, 클라우드 서비스, 엔터프라이즈 서비스 등으로 볼 수 있다.
- [0047] 도 3은 본 발명의 일 실시예에 따른 사물 인터넷에서 레스트 기반의 서비스 연동 시스템을 설명하는 도면이다.

- [0048] 도 3을 참고하면, 레스트를 기반으로 사물 인터넷 서비스인 디바이스 서비스와 외부 서비스의 연동을 위한 사물 인터넷에서 레스트 기반의 서비스 연동 시스템은 레스트 클라이언트(100)와 레스트 어댑터(200)를 포함한다.
- [0049] 레스트 클라이언트(100)는 사물 인터넷의 서비스와 관련한 레스트 API를 동기/비동기 방식으로 호출한다.
- [0050] 레스트 어댑터(200)는 사물 인터넷의 서비스 연동을 위해 클라이언트 역할과 서버 역할을 모두 수행하는 것으로서, 레스트 API와 DDS 기반의 서비스 토픽을 상호 변환하는 동작을 수행한다.
- [0051] 이러한 레스트 어댑터(200)는 DDS 엔티티 관리자(210), 토픽 인스턴스 관리자(220), 레스트 API 정보 관리자(230), 레스트 API 등록기(240), 레스트 API 생성기(250), 레스트 API 클라이언트 모듈(260) 및 레스트 API 서버 모듈(270)을 포함한다.
- [0052] DDS 엔티티 관리자(DDS Entity Manager)(210)는 DDS 엔티티인 도메인참여자(DomainParticipant), 발간자, 구독자, 데이터라이터(DataWriter), 데이터 리더(DataReader)를 포함하며, 서비스를 요청하고 응답을 수락하는 기능을 담당한다. 이러한 DDS 엔티티 관리자(210)는 레스트 어댑터(200)가 DDS를 이용하기 위하여 DDS의 엔티티를 생성할 수 있는 API를 가진 컴포넌트로서, 서비스 토픽을 발간/구독하기 위해 레스트 어댑터(200)의 시작과 동시에 메모리상에 싱글톤 패턴(Singleton Pattern)으로 구현된 전역적인 쓰레드로 존재한다.
- [0053] DDS 엔티티 관리자(210)는 발간자와 데이터라이터의 생성, 구독자와 데이터리더 생성을 통한 DDS 엔티티 생성의 역할, 서비스 토픽에 대한 리스너와 웨이트셋 생성을 통한 토픽 인스턴스 발간/구독/삭제의 역할을 수행한다.
- [0054] 토픽 인스턴스 관리자(TopicInstance Manager)(220)는 DDS 엔티티 관리자(210)를 통해 DDS로 보낼 정보를 생성하는 부분으로서, 서비스 토픽의 샘플에 대한 직렬화와 역직렬화한 정보를 REST API 정보 관리자(230)에 전달한다.
- [0055] 토픽 인스턴스 관리자(220)는 레스트 어댑터(200)와 DDS 사이에 존재하여 DDS에서 전달되는 정보를 레스트 API로 타입 변환을 하거나 레스트 어댑터(200) 사이에서 전달되는 정보를 DDS 기반의 토픽 정보로 타입을 변환한다.
- [0056] REST API 정보 관리자(230)는 외부의 REST API를 호출하기 위한 정보를 관리한다. 이러한 레스트 API 정보 관리자(230)는 레스트 API의 속성들을 테이블에 저장하며 CRUD 연산에 의해 레스트 API들을 등록/읽기/수정/삭제하는 데이터 베이스 관련 API들을 가지고 있다.
- [0057] REST API 등록기(REST API Register)(240)는 외부의 REST API를 등록하여 서비스 토픽으로 변환하는 기능을 담당한다. 레스트 API 등록기(240)에 의해 등록된 레스트 API들은 레스트 API 정보 관리자(230)에 전송되어 데이터베이스에 저장되며, 토픽 인스턴스 관리자(220)가 토픽 정보로 변환하여 DDS 엔티티 관리자(210)가 DDS 내에 발간하여 수행이 완료되도록 한다.
- [0058] REST API 생성기(REST API Generator)(250)는 사물인터넷의 디바이스 서비스를 REST API로 변환하는 기능을 담당한다. 즉 레스트 API 생성기(250)는 DDS내에서 발간된 서비스 토픽을 구독하고, 이 서비스 토픽 정보를 레스트 API로 변환한다. 이때, 레스트 API는 리소스빌더(Resource.Builder) 객체를 이용하여 웹서버에 등록될 리소스를 생성하여 데이터베이스에 저장하고, 리소스빌더 객체는 URI와 HTTP 메소드, 레스트 클라이언트(100)의 요청시 처리하게 될 클래스를 설정한다. 레스트 API 생성기(250)에서 생성된 레스트 API는 웹 서버 실행 중에 동적으로 등록/삭제가 가능하다.
- [0059] REST 클라이언트 모듈(260)과 REST API 서버 모듈(270)은 사물인터넷의 디바이스 서비스와 외부의 REST API를 양방향으로 연동하기 위한 호출을 처리한다.
- [0060] 또한, 레스트 어댑터(200)는 레스트 API의 속성들 또는 리소스들 포함한 각종 정보를 저장하는 데이터베이스를 포함하고 있다.
- [0061] 도 4는 본 발명의 일 실시예에 따른 사물 인터넷에서 레스트 기반의 서비스 연동 방법을 설명하는 순서도이다.
- [0062] 도 4를 참고하면, 사물 인터넷에서 레스트 기반의 서비스 연동 방법은, 사물인터넷의 서비스 연동을 위해 양방향의 레스트 어댑터(200)를 설계하는 과정(S10), 레스트 어댑터(200)가 클라이언트 역할을 실행하는 과정(S20) 및 레스트 어댑터(200)가 서버 역할을 실행하는 과정(S30)으로 진행된다.
- [0063] 먼저, 양방향의 레스트 어댑터(200)를 설계하는 과정(S10)은 서비스 토픽으로 명세된 디바이스 서비스를 레스트

API로 변환하거나, 레스트 API를 웹서버에 등록하여 서비스 토픽으로 변환한다.

[0064] 레스트 어댑터(200)가 클라이언트 역할을 실행하는 과정(S20) 및 레스트 어댑터(200)가 서버 역할을 실행하는 과정(S30)은 서비스 토픽과 레스트 API의 변환 동작을 수행하면서 레스트 API를 삭제 또는 사물인터넷의 디바이스 서비스가 제어된 경우에 대한 동작을 수행한다.

[0065] 도 5는 도 4의 레스트 어댑터에 대한 클라이언트 역할을 실행하는 상태를 설명하는 도면이고, 도 6은 도 4의 레스트 어댑터에 대한 동기/비동기 방식의 클라이언트 역할 실행 과정을 설명하는 도면이다.

[0066] 도 5 및 도 6을 참고하면, 레스트 클라이언트 모듈(260)은 레스트 어댑터(200)가 클라이언트 역할을 수행하기 위해 서비스 토픽을 이용하여 웹 상에 존재하는 레스트 API를 호출하고, 등록된 레스트 API 정보를 레스트 API 정보 관리자(230)에서 가져올 수 있다.

[0067] 이때, 레스트 클라이언트 모듈(260)은 동기/비동기 방식으로 레스트 API를 호출하는데, 동기 방식의 서비스 요청 토픽(S Req)을 호출할 경우에 웹상의 레스트 API를 호출하여 그 응답(S Res)을 전달받을때까지 기다린다. 레스트 클라이언트 모듈(260)의 비동기 방식의 호출은 응답 메시지가 없는 경우와 HTTP 상태 코드만 오는 경우를 처리하도록 한다.

[0068] 서비스 요청 토픽은 표 2에 나타나 있듯이, DEST_ID에는 레스트 API 주소, 서비스 이름에는 레스트 API 이름, 파라미터에는 인자값을 이용하고, HTTP 메소드를 통해 레스트 API를 호출한다.

표 2

```

“ServiceRequestTopic.idl”

struct ServiceRequestTopic {

    string requestID;

    string serviceID;

    string name;

    string parameters;

    string values;

}; #pragma keylist ServiceRequestTopic serviceID name // two keys
    
```

[0069]

[0070] 레스트 API의 응답이 오면 표 3에 나타난 서비스 응답 토픽에 결과값(VALUE)을 넣어 보낸다.

표 3

```

“ServiceResponseTopic.idl”

struct ServiceResponseTopic {

    string requestID;

    string name;

    string parameters;

    string values;

}; #pragma keylist ServiceResponseTopic    requestID name // two keys

```

[0071]

[0072]

서비스의 특성 상 서비스 사용자는 어떤 동작을 요청하고, 서비스 제공자가 처리한 서비스의 결과값을 받기 때문에 서비스를 요청하는 토픽과 서비스 요청에 따른 처리 결과를 알려주는 토픽으로 구분된다. 표 2와 표 3은 서비스 요청 토픽과 서비스 요청에 따른 결과값을 알려주는 서비스 응답 토픽을 IDL(Interface Definition Language)로 정의한 것이다.

[0073]

표 2 및 표 3에서 정의된 토픽은 도메인 내의 발간자는 데이터라이터(DataWriter)를 통해 지속적으로 토픽을 발간하고, 구독자는 미리 정한 토픽을 지속적으로 구독한다.

[0074]

서비스 요청 토픽은 호출하려는 서비스 요청 ID, 서비스 이름, 인자값 이름, 인자값을 넣어 데이터라이터를 통해 DDS 도메인 내에 발간한다. 서비스 요청 ID와 서비스 이름이 복합적으로 키를 구성되기 때문에 디바이스 서비스 어댑터는 데이터리더를 통해 해당 토픽을 수신하면 요청한 서비스를 확인하여 수행하도록 한다. 서비스의 응답 토픽은 데이터라이터를 통해 서비스 응답 토픽에 요청한 서비스 요청 ID, 서비스 ID, 서비스 이름, 결과값 이름, 결과값을 함께 보내면 레스트 어댑터(200)가 수신하여 서비스의 결과값을 확인할 수 있고, 서비스 요청 ID와 서비스 이름이 복합적으로 키를 구성한다. 이와 같은 서비스 요청은 디바이스 서비스 어댑터가 레스트 API를 요청하는 경우에도 동일하게 동작한다.

[0075]

이와 같이, 레스트 어댑터(200)는 기존의 HTTP 프로토콜을 이용하여 데이터를 전달하기 때문에 부수적인 레이어가 필요하지 않으며, HTTP의 기본 메소드인 GET/PUT/POST/DELETE를 이용하여 일관된 레스트 API 오퍼레이션(Operations)으로 제어가 가능하다. 그로 인해 사물인터넷의 수많은 사물들을 하나의 고유한 URI로 구별하여 표현할 수 있으며 사물에서 발생하는 정보들을 HTTP의 메소드로 제어가 가능하며, 레스트 어댑터(200)는 처리 결과를 HTTP 상태 코드(Status code)로 전달받게 된다

[0076]

도 7은 도 4의 레스트 어댑터에 대한 서버 역할을 실행하는 상태를 설명하는 도면이고, 도 8은 도 4의 레스트 어댑터에 대한 동기/비동기 방식의 서버 역할 실행 과정을 설명하는 도면이다.

[0077]

도 7 및 도 8을 참고하면, 레스트 API 서버 모듈(270)은 레스트 클라이언트(100)의 서비스 요청을 처리하기 위해 레스트 API 정보를 서비스 토픽으로 변환하여 DDS 내에 발간하고, 서비스 토픽을 발간한 후에 DDS 내에서 응답을 구독하기 위해 웨이트셋(WaitSet)을 이용하여 현재 쓰레드를 블록 상태에서 응답을 기다리며, 웨이트셋이 응답을 전달받으면 서비스 토픽을 레스트 클라이언트(100)가 요청한 응답으로 타입 변환하여 레스트 클라이언트(100)에 전달하도록 한다.

[0078]

레스트 어댑터(200)는 서비스 토픽으로 명세된 사물인터넷의 디바이스 서비스를 레스트 API로 변환하여 레스트 클라이언트(100)에 제공하기 때문에 레스트 클라이언트(100)는 레스트 API 서버 모듈(270)을 통해 제공하는 서비스로 인식할 수 있다.

- [0079] DDS 내에서 구독한 서비스 토픽을 레스트 API로 변환하는 과정은 레스트 API를 서비스 토픽으로 변환하는 과정과 다르게 수많은 사물들이 등장하기 때문에 서비스 토픽이 더욱 빈번하게 일어난다. 레스트 어댑터(200)는 구독한 서비스 토픽을 레스트 API 속성에 맞게 변환해주고, 레스트 어댑터(200)가 가진 웹 서버에 리소스(Resource) 형태로 등록된다. 이러한 토픽을 레스트 API로 변환하는 과정은 웹 서버가 실행 중에도 일어나게 되므로 웹 서버는 레스트 API를 실행 중에도 동적으로 등록가능 해야 한다.
- [0080] 도 9는 사물 인터넷의 디바이스 서비스를 레스트 API로 변환하는 과정을 설명하는 도면이다.
- [0081] 도 9를 참고하면, 레스트 어댑터(200)의 DDS 엔티티 관리자(210)는 DDS 내에서 비동기적으로 전달되는 발간된 토픽(Publish Topic)을 구독한다. 이를 위해 DDS 엔티티 관리자(210)는 발간토픽 리스너(Publish Topic Listener)를 생성하여 토픽의 발간 여부를 확인(Notify) 받는다. DDS 엔티티 관리자(210)에 의해 생성된 데이터 리더(Data Reader)가 해당 토픽을 읽는다.(1~5) 데이터리더는 데이터라이터를 통해 DDS 엔티티 관리자(210)에게 토픽 인스턴스를 전달하고, DDS 엔티티 관리자(210)는 토픽 인스턴스를 포함하는 토픽의 정보를 구성하여 토픽 인스턴스 관리자(220)에게 전달한다. (6, 7)
- [0082] 서비스 토픽은 데이터를 주고받는 가상의 채널로 실제 데이터를 송수신하기위해서 키로 인스턴스를 구분한다. 즉 디바이스 서비스와 레스트 API를 명세한 서비스 토픽은 동일한 데이터 타입을 가지며, 서비스 ID와 서비스 이름을 쌍으로 키가 구성된 인스턴스가 여러개 생성된다.
- [0083] 이때, 서비스 토픽의 수가 증가하면 인스턴스의 수도 증가하지만, 동일한 데이터 타입을 가지므로 이를 발간/구독하는 데이터라이터와 데이터리더의 수는 증가하지 않는다.
- [0084] 레스트 어댑터(200)는 인스턴스의 전송단위인 샘플을 통해 정보를 송수신한다. 샘플은 네트워크 전송에 적합하도록 직렬화되어 있어 레스트 API로 변환하기 위해 정보를 추출하는 역직렬화를 수행한다.(8)
- [0085] 토픽 인스턴스 관리자(220) 및 레스트 API 생성기(250)는 데이터리더를 통해 읽어온 서비스 토픽의 정보를 REST API 속성에 맞게 타입 변환하고, 타입 변환된 정보에서 REST API를 생성한다. 즉, 역직렬화를 통해 추출한 서비스 이름과 인자값을 이용하여 레스트 API 서버 모듈(270)에 등록한다.(9~15)
- [0086] 서비스 이름은 레스트 API 이름으로 동일하게 사용되고, 레스트 API 주소가 생성되어 외부의 레스트 클라이언트(100)가 호출할 수 있도록 한다. 예를 들면, 디바이스 서비스 이름이 “lightonoff” 일 경우에, 레스트 어댑터(200)를 통해 레스트 API 이름으로 사용하는 동시에 “http://restadaptor.cnu.ac.kr/lightonoff”의 주소로 레스트 API를 표현할 수 있다.
- [0087] 이렇게 생성된 디바이스 서비스의 레스트 API는 레스트 어댑터(200)에서 레스트 API 주소, HTTP 메소드, 인자값 등을 확인할 수 있다. 레스트 API의 정보는 추후 변환된 레스트 API 호출이 있는 경우에 DDS를 통해 디바이스 서비스를 호출하도록 정보를 유지한다. 레스트 어댑터(200)는 레스트 API를 등록하기 이전에 레스트 API 서버 모듈(270)에 이미 등록되어 있는지를 확인하여 동일한 레스트 API가 중복저장되지 않도록 한다.
- [0088] 서비스 토픽의 정보는 서비스 ID인 발간자(타 어댑터의 이름), 타 어댑터에서 사용 가능한 서비스 이름, 서비스의 파라미터의 이름과 파라미터 타입을 JSON포맷으로 나타낸 파라미터(PARAMETERS)를 포함한다. 그리고, 서비스 토픽과 레스트 API 속성의 매핑 정보는 다음과 같다.
- [0089] API 이름은 서비스 토픽의 서비스 이름, HTTP 메소드는 GET 메소드를 이용하여 호출, 레스트 API 주소는 웹 서버에서 등록된 URI 주소, 파라미터는 서비스 토픽의 파라미터 값, 파라미터 타입은 GET 메소드를 이용하기 위해 쿼리(Query Parameter)를 이용, 응답 타입은 XML과 JSON 응답포맷을 각각 이용한다.
- [0090] 레스트 어댑터(200)는 서비스 토픽을 레스트 API로 변환하는 과정이 완료되면 웹 서버에 변환된 REST API들을 리소스(Resource)로 등록하고, 해당 등록 작업이 완료되어 웹 서버가 동작하면 웹 상에 존재하는 레스트 클라이언트(100)가 레스트 어댑터(200)에게 서비스에 해당하는 레스트 API를 호출 할 수 있게 된다. 레스트 어댑터(200)는 레스트 클라이언트(100)의 동기/비동기 방식의 호출을 받을 수 있다.
- [0091] 동기 방식의 호출의 경우 레스트 클라이언트(100)가 레스트 어댑터(200)에 서비스를 요청하면 레스트 어댑터(200)는 서버 역할을 하여 DDS내에 해당 하는 서비스를 서비스 요청 토픽에 작성하여 발간하고, 처리결과를 서비스 응답 토픽으로 구독하여 레스트 어댑터(200)가 레스트 클라이언트(100)에게 전달한다.
- [0092] 비동기 방식의 호출의 경우에, 레스트 클라이언트(100)가 레스트 어댑터(200)에 서비스를 요청하면, DDS 내에

요청한 서비스를 전달하고 레스트 어댑터(200)는 처리 결과를 HTTP 상태 코드로 전달한다.

[0093] DDS는 레스트 어댑터가 DDS내에서 서비스에 대한 요청을 토픽 정보로 받게 되고, DDS내에서 서비스의 요청은 리스너를 이용하여 비동기적으로 요청이 가능하도록 한다. 리스너가 서비스 토픽이 도착했음을 확인하면 데이터리더는 읽기(Read)오퍼레이션을 수행한다. 이러한 정보를 서비스 토픽에서 레스트 API로 타입 변환하여 레스트 API 클라이언트 모듈(260)에게 전달한다. 레스트 API 클라이언트 모듈(260)은 전달받은 정보를 이용하여 해당 레스트 API에 요청을 하게 된다. 레스트 API 클라이언트 모듈(260)이 응답을 받으면 그 응답 결과를 서비스 토픽에 맞게 타입 변환한 뒤 DDS 내에 발간한다.

[0094] 레스트 어댑터는 레스트 클라이언트(100)에서 레스트 API를 요청하면, 웹 서버가 요청 쓰레드를 생성하고 DDS 내에 쓰기(Write) 오퍼레이션으로 해당 요청 정보를 담은 서비스 토픽을 발간한다. 그리고, 웹 서버내의 요청한 요청 쓰레드가 응답을 받기 위해서 웨이트셋(WaitSet)을 생성하고 쿼리조건(QueryCondition)에 응답 받으려는 서비스 토픽의 정보를 조건으로 설정하여 해당 서비스 토픽이 도착하기를 블록(Block)상태에서 기다린다. 웨이트셋(WaitSet)은 해당 서비스 토픽이 도착하면 언블록(Unblock)상태가 되어 데이터리더가 읽기 오퍼레이션을 수행하도록 한다. 레스트 어댑터는 서비스 응답 토픽을 레스트 클라이언트(100)에게 전달한다.

[0095] 사물인터넷은 사물들이 유무선 네트워크로 연결되어 유기적으로 정보를 주고받으면서 상호작용하는 개념으로 수많은 사물을 접근하고 정보를 송수신하기 위한 클라우드 기반의 사물인터넷(CoT; Cloud of Things) 구축이 필요하다. 클라우드 기반의 사물인터넷은 네트워크로 연결된 수많은 콘텐츠, 서비스와 사물은 대량으로 발생하는 정보를 효율적으로 관리하고 결합할 수 있어야 하며, 현재 구성된 시스템에 영향을 주지 않고 새로운 사물들을 동적으로 네트워크에 적용시켜 바로 통신 또는 서비스로 제공해야 한다. 또한, 대부분이 무선 환경의 사물이기 때문에 네트워크에 추가/변경/삭제가 동적으로 가능해야 하고, 신뢰성 및 실시간성 등의 메시지 송수신에 필요한 서비스 품질(QoS; Quality of Services)을 보장해야 한다.

[0096] 상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

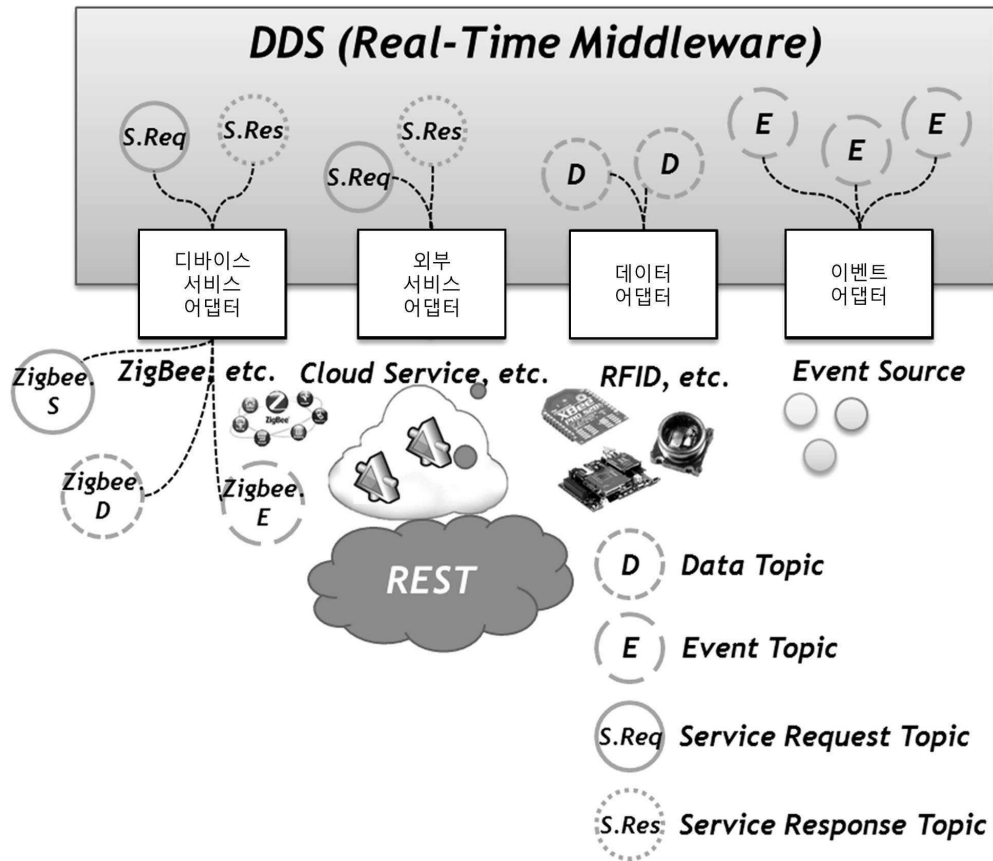
부호의 설명

[0097]

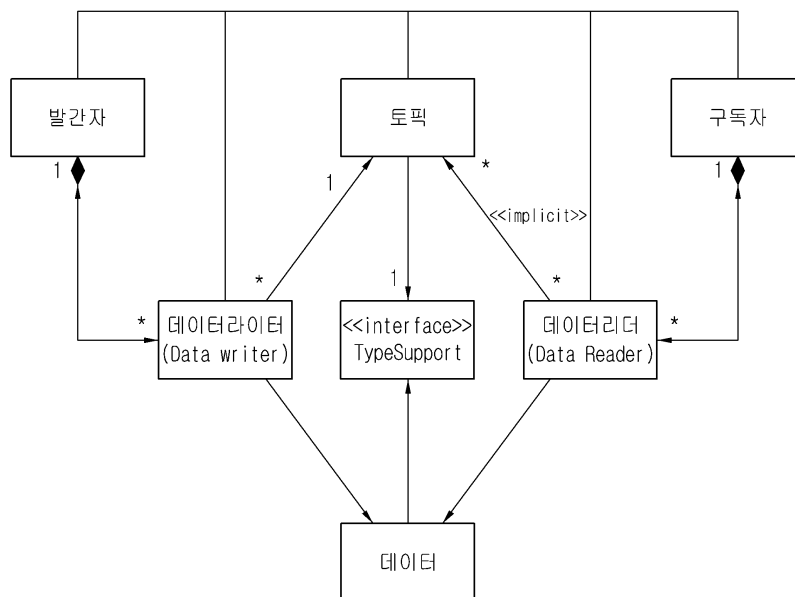
100 : 레스트 클라이언트	200 : 레스트 어댑터
210 : DDS 엔티티 관리자	220 : 토픽 인스턴스 관리자
230 : 레스트 API 정보 관리자	240 : 레스트 API 등록기
250 : 레스트 API 생성기	260 : 레스트 API 클라이언트 모듈
270 : 레스트 API 서버 모듈	

도면

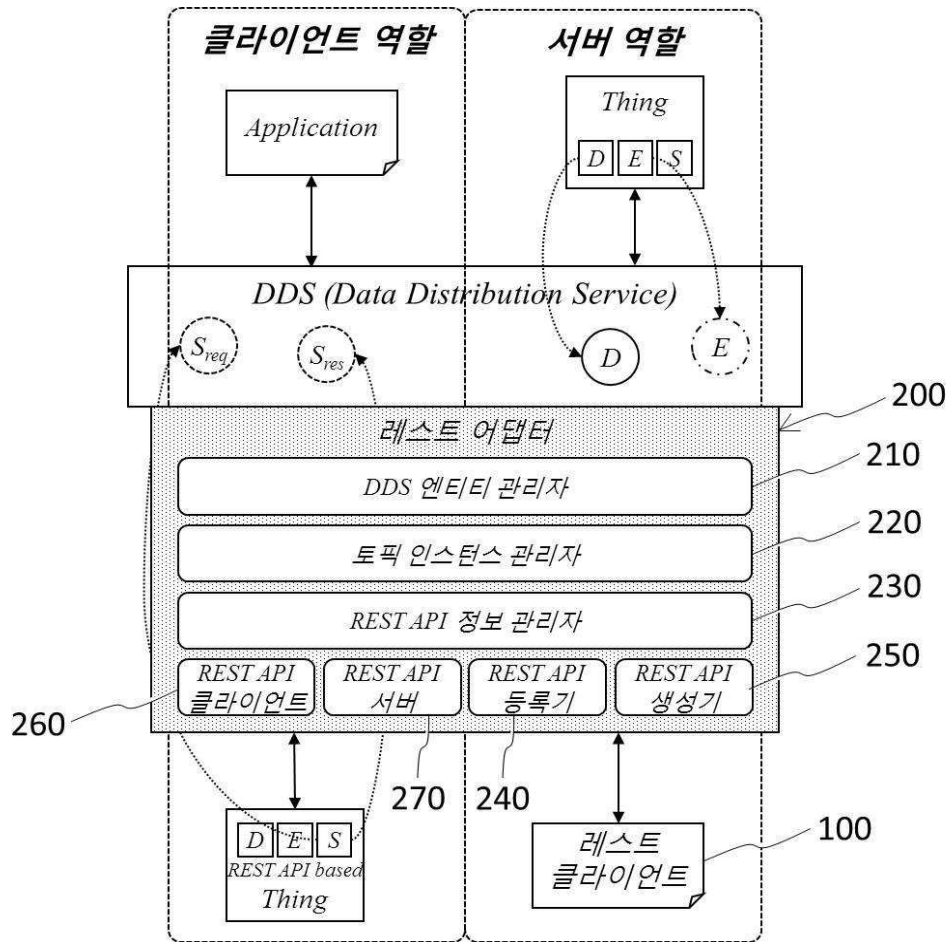
도면1



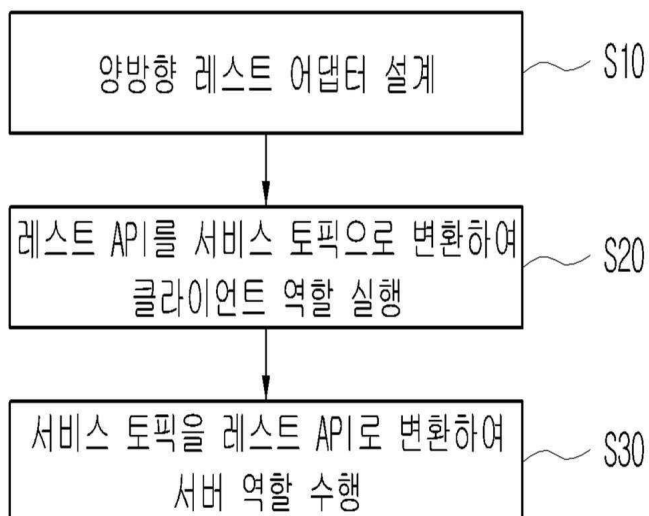
도면2



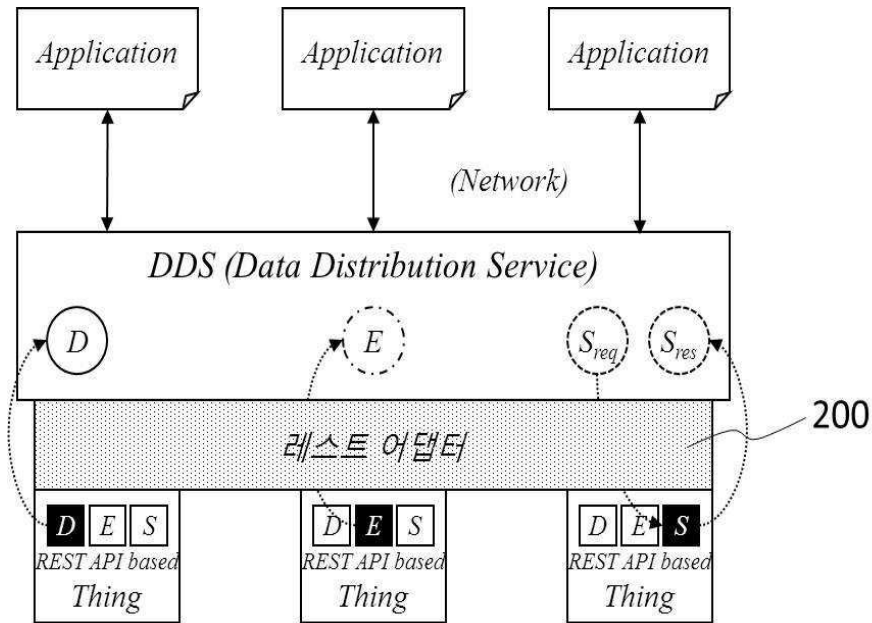
도면3



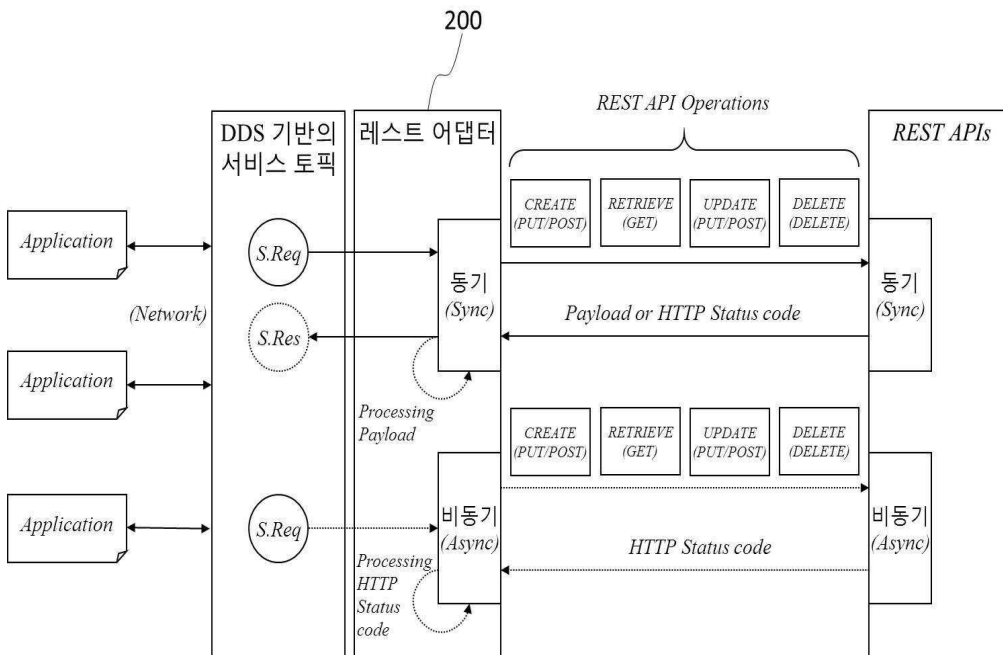
도면4



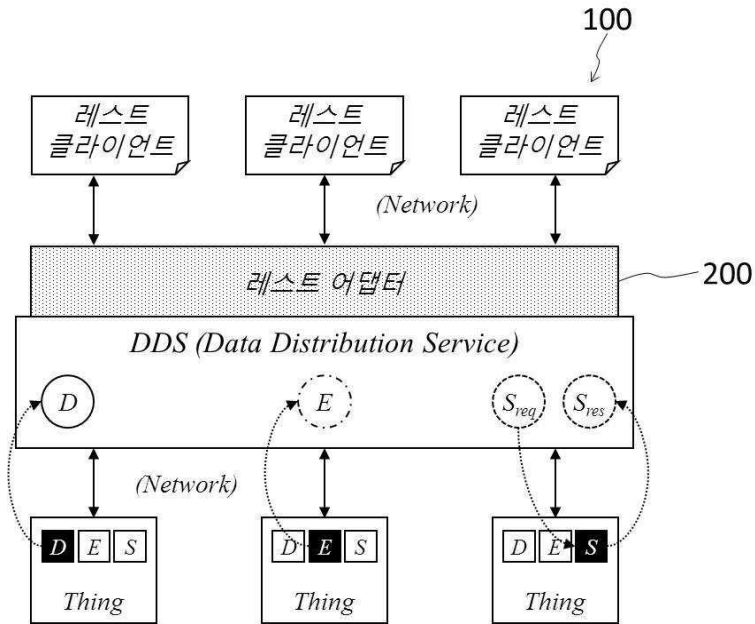
도면5



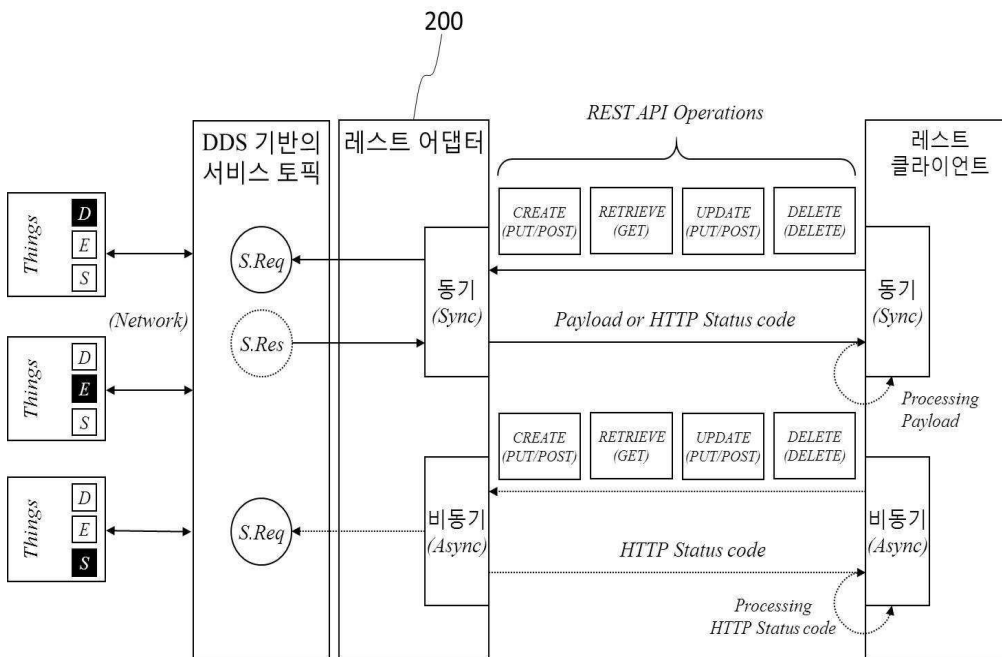
도면6



도면7



도면8



도면9

