



등록특허 10-2307976



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년09월30일
(11) 등록번호 10-2307976
(24) 등록일자 2021년09월27일

- (51) 국제특허분류(Int. Cl.)
G06F 3/16 (2018.01) *G10L 15/22* (2006.01)
H04M 1/27 (2006.01) *H04M 3/493* (2006.01)
H04N 21/239 (2011.01) *H04N 21/41* (2011.01)
H04N 21/4147 (2011.01) *H04N 21/422* (2016.01)
H04N 21/47 (2016.01) *H04N 21/4722* (2011.01)
- (52) CPC특허분류
G06F 3/167 (2013.01)
G10L 15/22 (2013.01)
- (21) 출원번호 10-2020-7029660(분할)
(22) 출원일자(국제) 2017년05월10일
심사청구일자 2020년10월27일
(85) 번역문제출일자 2020년10월15일
(65) 공개번호 10-2020-0121915
(43) 공개일자 2020년10월26일
(62) 원출원 특허 10-2018-7033471
원출원일자(국제) 2017년05월10일
심사청구일자 2018년12월10일
(86) 국제출원번호 PCT/US2017/032002
(87) 국제공개번호 WO 2017/197010
국제공개일자 2017년11월16일
(30) 우선권주장
62/334,434 2016년05월10일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
US20150006182 A1*
(뒷면에 계속)
- 전체 청구항 수 : 총 20 항

심사관 : 서광훈

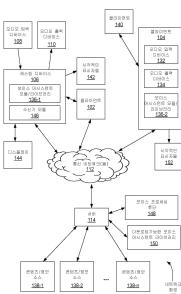
(54) 발명의 명칭 디바이스들 상의 보이스 어시스턴트에 대한 구현들

(57) 요약

오디오 입력 시스템을 갖는 전자 디바이스에서의 방법은, 디바이스에서 구두 입력을 수신하는 단계; 구두 입력을 프로세싱하는 단계; 요청을 원격 시스템에 송신하는 단계 - 요청은 구두 입력에 기반하여 결정된 정보를 포함함 -; 요청에 대한 응답을 수신하는 단계 - 응답은 구두 입력에 기반한 정보에 따라 원격 시스템에 의해 생성됨

(뒷면에 계속)

대 표 도 - 도1



—; 및 응답에 따라 동작을 수행하는 단계를 포함하며, 여기서, 수신하는 단계, 프로세싱하는 단계, 송신하는 단계, 수신하는 단계 및 수행하는 단계 중 하나 이상은 전자 디바이스 상에서 실행되는 보이스 어시스턴트 라이브러리의 하나 이상의 보이스 프로세싱 모듈들에 의해 수행되고, 보이스 프로세싱 모듈들은 하나 이상의 애플리케이션 프로그램들에 액세스가능한 복수의 보이스 프로세싱 동작들을 제공하고 그리고/또는 전자 디바이스 상에서 실행되거나 또는 실행가능한 소프트웨어를 동작시킨다.

(52) CPC특허분류

HO4M 1/271 (2013.01)
HO4M 3/493 (2013.01)
HO4N 21/2393 (2013.01)
HO4N 21/4104 (2020.08)
HO4N 21/42203 (2013.01)
HO4N 21/42204 (2020.08)
HO4N 21/42206 (2013.01)
HO4N 21/47 (2020.08)
G10L 2015/223 (2013.01)

(56) 선행기술조사문헌

JP2014507030 A
 US20040128137 A1
 KR1020120137425 A
 KR101579292 B1*
 KR1020150029974 A*

*는 심사관에 의하여 인용된 문헌

(30) 우선권주장

62/336,566 2016년05월13일 미국(US)
 62/336,551 2016년05월13일 미국(US)
 62/336,565 2016년05월13일 미국(US)
 62/336,569 2016년05월13일 미국(US)

명세서

청구범위

청구항 1

제1 디바이스 타입을 갖고, 그리고 오디오 입력 시스템, 하나 이상의 프로세서들, 및 상기 하나 이상의 프로세서들에 의한 실행을 위한 하나 이상의 프로그램들을 저장한 메모리를 포함하는 전자 디바이스에서:

상기 전자 디바이스가 상기 제1 디바이스 타입을 갖는 것에 기초하여 상기 전자 디바이스 상에서의 실행을 위해 디바이스-독립적 보이스 어시스턴트 라이브러리(device-agnostic voice assistant library)를 구성하는 단계
- 상기 보이스 어시스턴트 라이브러리는, 상기 제1 디바이스 타입을 포함하는, 복수의 상이한 전자 디바이스 타입들에 걸친 실행을 위해 동작가능하고; 그리고 상기 보이스 어시스턴트 라이브러리는 복수의 보이스 프로세싱 모듈들을 포함하고, 상기 보이스 프로세싱 모듈들 각각은 상기 상이한 전자 디바이스 타입들 상에서 실행되거나 또는 실행가능한 애플리케이션 프로그램들에 액세스가능한 하나 이상의 보이스 프로세싱 동작들을 제공함 -;

상기 구성하는 단계 이후에, 오디오 콘텐츠에 대한 원격 위치를 식별하는 단계;

상기 원격 위치로부터 상기 오디오 콘텐츠를 스트리밍하는 단계;

스피커를 통해, 제1 볼륨 레벨로 상기 오디오 콘텐츠로부터 청취가능(audible) 출력을 생성하는 단계;

상기 청취가능 출력을 생성하는 동안:

상기 전자 디바이스에 대한 사용자 입력을 수신하는 단계;

상기 전자 디바이스 상에서 실행되는 상기 보이스 어시스턴트 라이브러리를 이용하여 상기 사용자 입력을 프로세싱하는 단계;

상기 사용자 입력이, 사용자가 상기 전자 디바이스에 대한 구두(verbal) 입력을 제출하려 한다는 표시에 대응한다는 결정에 응답하여, 상기 제1 볼륨 레벨로부터, 상기 제1 볼륨 레벨보다 작은, 제2 볼륨 레벨로 상기 청취가능 출력을 조정하는 단계; 및

상기 조정하는 단계에 후속하여:

상기 사용자로부터 상기 구두 입력을 수신하는 단계; 및

상기 구두 입력을 프로세싱하는 단계

를 포함하는, 방법.

청구항 2

제1항에 있어서,

상기 전자 디바이스는 마이크로폰을 포함하고, 그리고 상기 전자 디바이스에 대한 상기 사용자 입력은 상기 사용자가 말하는 오디오 핫워드(hotword)를 상기 마이크로폰에 의해 검출하는 것을 포함하는, 방법.

청구항 3

제1항에 있어서,

상기 전자 디바이스에 대한 상기 사용자 입력은 상기 전자 디바이스 상의 사용자 인터페이스 엘리먼트의 활성화를 포함하는, 방법.

청구항 4

제3항에 있어서,

상기 사용자 인터페이스 엘리먼트는 어시스턴트 인터페이스 버튼을 포함하는, 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 오디오 콘텐츠는 서드-파티(third-party) 오디오시각적(audiovisual) 콘텐츠를 포함하는, 방법.

청구항 6

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 오디오 콘텐츠는 캐스팅되는 콘텐츠를 포함하는, 방법.

청구항 7

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 구두 입력을 수신한 후에, 상기 제2 볼륨 레벨로부터 상기 제1 볼륨 레벨로 상기 청취가능 출력을 조정하는 단계를 더 포함하는, 방법.

청구항 8

제1 디바이스 타입을 갖는 전자 디바이스로서,

스피커;

하나 이상의 프로세서들; 및

상기 하나 이상의 프로세서들에 의해 실행될 하나 이상의 프로그램들을 저장하는 메모리를 포함하고,

상기 하나 이상의 프로그램들은:

상기 전자 디바이스가 상기 제1 디바이스 타입을 갖는 것에 기초하여 상기 전자 디바이스 상에서의 실행을 위해 디바이스-독립적 보이스 어시스턴트 라이브러리를 구성하고 — 상기 보이스 어시스턴트 라이브러리는, 상기 제1 디바이스 타입을 포함하는, 복수의 상이한 전자 디바이스 타입들에 걸친 실행을 위해 동작가능하고; 그리고 상기 보이스 어시스턴트 라이브러리는 복수의 보이스 프로세싱 모듈들을 포함하고, 상기 보이스 프로세싱 모듈들 각각은 상기 상이한 전자 디바이스 타입들 상에서 실행되거나 또는 실행가능한 애플리케이션 프로그램들에 액세스가능한 하나 이상의 보이스 프로세싱 동작들을 제공함 —;

상기 구성하는 것 이후에, 오디오 콘텐츠에 대한 원격 위치를 식별하고;

상기 원격 위치로부터 상기 오디오 콘텐츠를 스트리밍하고;

상기 스피커를 통해, 제1 볼륨 레벨로 상기 오디오 콘텐츠로부터 청취가능 출력을 생성하고; 그리고

상기 청취가능 출력을 생성하는 동안:

상기 전자 디바이스에 대한 사용자 입력을 수신하고;

상기 전자 디바이스 상에서 실행되는 상기 보이스 어시스턴트 라이브러리를 이용하여 상기 사용자 입력을 프로세싱하고;

상기 사용자 입력이, 사용자가 상기 전자 디바이스에 대한 구두 입력을 제출하려 한다는 표시에 대응한다는 결정에 응답하여, 상기 제1 볼륨 레벨로부터, 상기 제1 볼륨 레벨보다 작은, 제2 볼륨 레벨로 상기 청취가능 출력을 조정하고; 그리고

상기 조정하는 것에 후속하여:

상기 사용자로부터 상기 구두 입력을 수신하고; 그리고

상기 구두 입력을 프로세싱하기 위한

명령들을 포함하는, 전자 디바이스.

청구항 9

제8항에 있어서,

상기 하나 이상의 프로그램들은, 하나 이상의 애플리케이션 프로그래밍 인터페이스(API)들을 갖는 보이스 어시스턴트 애플리케이션을 포함하는, 전자 디바이스.

청구항 10

제9항에 있어서,

상기 하나 이상의 API들은 상기 디바이스의 오디오 출력들의 볼륨을 더킹(duck)하기 위한 API 기능을 포함하는, 전자 디바이스.

청구항 11

제9항 또는 제10항에 있어서,

상기 하나 이상의 API들은 상기 보이스 어시스턴트 애플리케이션과 하나 이상의 운영 시스템들 간의 호환 인터페이스를 제공하기 위한 API를 포함하는, 전자 디바이스.

청구항 12

제9항 또는 제10항에 있어서,

상기 하나 이상의 API들은 상기 보이스 어시스턴트 애플리케이션과 상기 디바이스의 하드웨어 기능성 간의 인터페이스를 제공하기 위한 API를 포함하는, 전자 디바이스.

청구항 13

제12항에 있어서,

하나 이상의 발광 다이오드(LED)들을 더 포함하고, 그리고

상기 디바이스의 상기 하드웨어 기능성은 상기 하나 이상의 LED들의 동작을 포함하는, 전자 디바이스.

청구항 14

제9항 또는 제10항에 있어서,

상기 구두 입력은 상기 디바이스 상의 제2 애플리케이션으로 지향되는 사용자 커맨드를 포함하고; 그리고

상기 하나 이상의 프로그램들은, 상기 사용자 커맨드에 따라 상기 하나 이상의 API들의 API 기능을 호출하기 위한 명령들을 더 포함하는, 전자 디바이스.

청구항 15

하나 이상의 프로그램들을 저장한 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 하나 이상의 프로그램들은, 제1 디바이스 타입을 갖고 그리고 하나 이상의 프로세서들, 메모리, 및 스피커를 포함하는 오디오 디바이스에 의해 실행될 때, 상기 오디오 디바이스로 하여금 동작들을 수행하게 하는 명령들을 포함하고,

상기 동작들은:

상기 오디오 디바이스가 상기 제1 디바이스 타입을 갖는 것에 기초하여 상기 오디오 디바이스 상에서의 실행을 위해 디바이스-독립적 보이스 어시스턴트 라이브러리를 구성하는 것 – 상기 보이스 어시스턴트 라이브러리는, 상기 제1 디바이스 타입을 포함하는, 복수의 상이한 전자 디바이스 타입들에 걸친 실행을 위해 동작가능하고; 그리고 상기 보이스 어시스턴트 라이브러리는 복수의 보이스 프로세싱 모듈들을 포함하고, 상기 보이스 프로세싱 모듈들 각각은 상기 상이한 전자 디바이스 타입들 상에서 실행되거나 또는 실행가능한 애플리케이션 프로그램들에 액세스가능한 하나 이상의 보이스 프로세싱 동작들을 제공함 –;

상기 구성하는 것 이후에, 오디오 콘텐츠에 대한 원격 위치를 식별하는 것;

상기 원격 위치로부터 상기 오디오 콘텐츠를 스트리밍하는 것;

상기 스피커를 통해, 제1 볼륨 레벨로 상기 오디오 콘텐츠로부터 청취가능 출력을 생성하는 것; 및
상기 청취가능 출력을 생성하는 동안:

상기 오디오 디바이스에 대한 사용자 입력을 수신하는 것;

상기 사용자 입력이, 사용자가 상기 오디오 디바이스에 대한 구두 입력을 제출하려 한다는 표시에 대응한다는 결정에 응답하여, 상기 제1 볼륨 레벨로부터, 상기 제1 볼륨 레벨보다 작은, 제2 볼륨 레벨로 상기 청취가능 출력을 조정하는 것;

상기 조정하는 것에 후속하여:

상기 사용자로부터 상기 구두 입력을 수신하는 것; 및

상기 구두 입력을 프로세싱하는 것

을 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 16

제15항에 있어서,

상기 하나 이상의 프로그램들은 보이스 어시스턴트 애플리케이션 및 추가적인 애플리케이션을 포함하고;

상기 제1 볼륨 레벨은 상기 추가적인 애플리케이션에 의해 설정되고; 그리고

상기 청취가능 출력은 상기 보이스 어시스턴트 애플리케이션을 통해 상기 제1 볼륨 레벨로부터 제2 볼륨 레벨로 조정되는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 17

제15항 또는 제16항에 있어서,

상기 하나 이상의 프로그램들은:

상기 구두 입력을 프로세싱함으로써 질의(query)를 결정하고;

원격 서버 시스템에 상기 질의를 전송하고;

상기 원격 서버 시스템으로부터 상기 질의에 대한 응답을 수신하고; 그리고

상기 응답에 따라 동작을 수행하기 위한

명령들을 더 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 18

제15항 또는 제16항에 있어서,

상기 하나 이상의 프로그램들은:

상기 구두 입력을 수신한 후에, 상기 제2 볼륨 레벨로부터 상기 제1 볼륨 레벨로 상기 청취가능 출력을 조정하기 위한

명령들을 더 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 19

제15항에 있어서,

상기 오디오 디바이스는 보이스 어시스턴트 애플리케이션 및 하나 이상의 추가적인 애플리케이션들을 저장하고; 그리고

상기 청취가능 출력은 상기 하나 이상의 추가적인 애플리케이션들 중 하나로부터의 출력을 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 20

제15항 또는 제16항에 있어서,

상기 오디오 디바이스는 마이크로폰을 포함하고, 그리고 상기 오디오 디바이스에 대한 상기 사용자 입력은 상기 사용자가 말하는 오디오 핫워드를 상기 마이크로폰에 의해 검출하는 것을 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 일반적으로, 디바이스들에 대한 보이스 어시스턴트(voice assistant)들 및 관련된 라이브러리들을 포함하지만 이에 제한되지는 않는 컴퓨터 기술에 관한 것이다.

배경 기술

[0002] 오디오/보이스 입력 및 출력을 통해 사용자와 상호작용하는 보이스-기반 어시스턴트들은 인터넷 및 클라우드 컴퓨팅의 성장과 함께 인기를 얻고 있다. 이를 어시스턴트들은, 몇몇 예들을 들자면 뉴스, 스포츠 스코어들, 날씨, 및 주식들을 포함하는 다양한 타입들의 정보를 제공할 뿐만 아니라 디지털 미디어의 소비를 위한 인터페이스를 제공할 수 있다.

[0003] 사용자는 보이스-기반 어시스턴트 기능이 바람직한 다수의 디바이스들을 가질 수 있다. 다양한 디바이스들에 걸쳐 구현 및 사용될 수 있고, 다양한 디바이스들에 걸쳐 일관된 경험을 제공할 수 있으며, 특정 디바이스에 특정한 기능을 지원할 수 있는 보이스-기반 어시스턴트를 갖는 것이 바람직하다.

발명의 내용

[0004] 본 명세서에 설명된 구현들은, 광범위하게 다양한 운영 시스템 플랫폼들에 대한 로컬 디바이스의 제어를 가능하게 하는 방식으로 임베디드(embeded) 시스템들 및/또는 디바이스들에 보이스 어시스턴트를 임베딩하거나 또는 포함시키는 것에 관한 것이다.

[0005] 몇몇 구현들에 따르면, 얇고 낮은-리소스-사용 디바이스-측 라이브러리는 오디오 데이터의 로컬 프로세싱, 웨이크워드(wakeword)들 또는 핫워드(hotword)들을 청취하는 것, 및 사용자 요청들을 전송하는 것을 포함하는 특성들을 갖는다. 부가적인 특성들은 클라우드 브레인(cloud brain)에 대한 연결, 확장가능한 보이스 액션 제어 시스템, 많은 별개의 동작 환경들로의 통합을 허용하는 이식성 계층(portability layer), 및 클라이언트 소프트웨어의 나머지와 비동기식으로 업데이트되기 위한 능력을 포함한다.

[0006] 설명된 구현들은 많은 상이한 디바이스들에 걸쳐 보이스 어시스턴트와 상호작용하기 위한 유사한 사용자 경험을 제공하는 이점을 갖는다.

[0007] 설명된 구현들은, 디바이스 그 자체로부터 가능하게 되는 혁신들로부터 보이스 어시스턴트 능력들의 디커플링된 혁신을 가능하게 하는 다른 이점을 갖는다. 예컨대, 개선된 인식 파이프라인이 생성되었다면, 그 파이프라인은 디바이스들로 푸시 아웃(push out)될 수 있지만, 디바이스 제조사는 그것을 수신하기 위해 어떠한 것도 행할 필요는 없으며, 이전의 보이스 커맨드들로부터 여전히 이득을 얻을 수 있다.

[0008] 몇몇 구현들에 따르면, 오디오 입력 시스템, 하나 이상의 프로세서들, 및 하나 이상의 프로세서들에 의한 실행을 위한 하나 이상의 프로그램들을 저장한 메모리를 갖는 전자 디바이스에서의 방법은, 디바이스에서 구두 입력(verbal input)을 수신하는 단계; 구두 입력을 프로세싱하는 단계; 요청을 원격 시스템에 송신하는 단계 – 요청은 구두 입력에 기반하여 결정된 정보를 포함함 –; 요청에 대한 응답을 수신하는 단계 – 응답은 구두 입력에 기반한 정보에 따라 원격 시스템에 의해 생성됨 –; 및 응답에 따라 동작을 수행하는 단계를 포함하며, 여기서, 수신하는 단계, 프로세싱하는 단계, 송신하는 단계, 수신하는 단계 및 수행하는 단계 중 하나 이상은 전자 디바이스 상에서 실행되는 보이스 어시스턴트 라이브러리의 하나 이상의 보이스 프로세싱 모듈들에 의해 수행되고, 보이스 프로세싱 모듈들은 하나 이상의 애플리케이션 프로그램들에 액세스가능한 복수의 보이스 프로세싱 동작들을 제공하고 그리고/또는 전자 디바이스 상에서 실행되거나 또는 실행가능한 소프트웨어를 동작시킨

다.

[0009] 몇몇 구현들에서, 오디오 입력 시스템을 포함하는 전자 디바이스들에 대한 디바이스-독립적인(device-agnostic) 보이스 어시스턴트 라이브러리는, 복수의 상이한 전자 디바이스 타입들 상에서 구현되는 공통 동작 시스템 상에서 실행되도록 구성되는 하나 이상의 보이스 프로세싱 모듈들을 포함하며, 보이스 프로세싱 모듈들은 애플리케이션 프로그램들에 액세스 가능한 복수의 보이스 프로세싱 동작들을 제공하고 전자 디바이스들 상에서 실행되는 소프트웨어를 동작시켜, 그에 의해, 보이스 프로세싱 동작들 중 하나 이상과 상호작용하도록 구성된 보이스-인에이를 애플리케이션들의 이식성을 가능하게 한다.

[0010] 몇몇 구현들에서, 전자 디바이스는 오디오 입력 시스템, 하나 이상의 프로세서들, 및 하나 이상의 프로세서들에 의해 실행될 하나 이상의 프로그램들을 저장한 메모리를 포함한다. 하나 이상의 프로그램들은, 디바이스에서 구두 입력을 수신하고; 구두 입력을 프로세싱하고; 요청을 원격 시스템에 송신하고 – 요청은 구두 입력에 기반하여 결정된 정보를 포함함 –; 요청에 대한 응답을 수신하며 – 응답은 구두 입력에 기반한 정보에 따라 원격 시스템에 의해 생성됨 –; 그리고 응답에 따라 동작을 수행하기 위한 명령들을 포함하며, 여기서, 수신, 프로세싱, 송신, 수신 및 수행 중 하나 이상은 전자 디바이스 상에서 실행되는 보이스 어시스턴트 라이브러리의 하나 이상의 보이스 프로세싱 모듈들에 의해 수행되고, 보이스 프로세싱 모듈들은 하나 이상의 애플리케이션 프로그램들에 액세스 가능한 복수의 보이스 프로세싱 동작들을 제공하고 그리고/또는 전자 디바이스 상에서 실행되거나 또는 실행 가능한 소프트웨어를 동작시킨다.

[0011] 몇몇 구현들에서, 비-일시적인 컴퓨터 판독가능 저장 매체는 하나 이상의 프로그램들을 저장한다. 하나 이상의 프로그램들은, 오디오 입력 시스템 및 하나 이상의 프로세서들을 갖는 전자 디바이스에 의해 실행될 경우, 전자 디바이스로 하여금, 디바이스에서 구두 입력을 수신하게 하고; 구두 입력을 프로세싱하게 하고; 요청을 원격 시스템에 송신하게 하고 – 요청은 구두 입력에 기반하여 결정된 정보를 포함함 –; 요청에 대한 응답을 수신하게 하며 – 응답은 구두 입력에 기반한 정보에 따라 원격 시스템에 의해 생성됨 –; 그리고 응답에 따라 동작을 수행하게 하는 명령들을 포함하며, 여기서, 수신, 프로세싱, 송신, 수신 및 수행 중 하나 이상은 전자 디바이스 상에서 실행되는 보이스 어시스턴트 라이브러리의 하나 이상의 보이스 프로세싱 모듈들에 의해 수행되고, 보이스 프로세싱 모듈들은 하나 이상의 애플리케이션 프로그램들에 액세스 가능한 복수의 보이스 프로세싱 동작들을 제공하고 그리고/또는 전자 디바이스 상에서 실행되거나 또는 실행 가능한 소프트웨어를 동작시킨다.

도면의 간단한 설명

[0012] 도 1은 몇몇 구현들에 따른 예시적인 네트워크 환경을 예시한 블록 다이어그램이다.

[0013] 도 2는 몇몇 구현들에 따른 예시적인 보이스 어시스턴트 클라이언트 디바이스를 예시한 다이어그램이다.

[0014] 도 3은 몇몇 구현들에 따른 예시적인 서버 시스템을 예시한 다이어그램이다.

[0015] 도 4는 몇몇 구현들에 따른, 보이스 어시스턴트 라이브러리들의 기능적인 뷰를 예시한 블록 다이어그램이다.

[0016] 도 5는 몇몇 구현들에 따른, 디바이스 상에서 구두 입력들을 프로세싱하기 위한 방법의 흐름도이다.

[0017] 유사한 참조 부호들은 도면들 전반에 걸쳐 대응하는 부분들을 지칭한다.

발명을 실시하기 위한 구체적인 내용

[0018] 이제, 다양한 구현들에 대한 참조가 상세히 이루어질 것이며, 다양한 구현들의 예들은 첨부한 도면들에 예시되어 있다. 다음의 상세한 설명에서, 다수의 특정한 세부사항들이 본 발명 및 설명된 구현들의 완전한 이해를 제공하기 위해 기재된다. 그러나, 본 발명은 이러한 특정한 세부사항들 없이 실시될 수 있다. 다른 예시들에서, 잘-알려진 방법들, 절차들, 컴포넌트들, 및 회로들은 구현들의 양상들을 불필요하게 모호하게 하지 않기 위해 상세히 설명되지 않는다.

[0019] 몇몇 구현들에서, 보이스 어시스턴트의 목적은, 다양한 디바이스들에 걸쳐 이용가능하고 광범위하게 다양한 사용 경우들을 가능하게 하는 개인화된 보이스 인터페이스를 사용자에게 제공하여, 사용자의 하루 동안 일관된 경험을 제공하는 것이다. 보이스 어시스턴트 및/또는 관련된 기능은 퍼스트-파티(first party) 및 서드-파티(third-party) 제품들 및 디바이스들에 통합될 수 있다.

- [0015] [0020] 예시적인 사용 경우는 미디어를 수반한다. 보이스 커맨드들은 보이스를 통한 음악, 라디오, 팟캐스트들, 뉴스, 및 다른 오디오 미디어의 재생 및 제어를 개시하는 데 사용될 수 있다. 예컨대, 사용자는 다양한 타입들의 오디오 미디어를 재생 또는 제어하기 위해 보이스 커맨드들(예컨대, "재즈 음악 재생", "107.5FM 재생", "다음 노래로 스kip", "'연속물(Serial)' 재생")을 말할 수 있다. 추가로, 그러한 커맨드들은 다양한 소스들, 이를테면 지상파 라디오 스테이션들의 온라인 스트리밍, 음악 가입 서비스들, 로컬 저장소, 원격 저장소 등으로부터의 오디오 미디어를 재생하는 데 사용될 수 있다. 추가로, 보이스 어시스턴트는 부가적인 콘텐츠를 지원하기 위해 캐스팅(casting) 디바이스들과 함께 이용가능한 통합들을 이용할 수 있다.
- [0016] [0021] 다른 예시적인 사용 경우는 원격 재생을 수반한다. 사용자는 보이스 어시스턴트 기능을 포함하는 캐스팅 디바이스에 보이스 커맨드를 이슈할 수 있으며, 보이스 커맨드에 따라, 커맨드에서 특정된 디바이스 상에서, 하나 이상의 디바이스들의 특정된 그룹 내의 디바이스들 상에서, 또는 커맨드에서 특정된 영역 내의 하나 이상의 디바이스들 상에서 미디어가 재생(예컨대, 그 디바이스들로 캐스팅)된다. 사용자는 또한, 커맨드에서 일반적인 카테고리들 또는 특정한 콘텐츠를 특정할 수 있으며, 커맨드 내의 특정된 카테고리 또는 콘텐츠에 따라 적절한 미디어가 재생된다.
- [0017] [0022] 추가의 예시적인 사용 경우는 비-미디어, 이를테면 생산성 특성들(예컨대, 타이머들, 알람 시계들, 캘린더), 흡 자동화, 검색 엔진(예컨대, 검색 쿼리들)에 의해 작동된 질문들 및 대답들, 재미(예컨대, 어시스턴트 개성, 농담들, 게임들, 이스터 에그들), 및 일상적인 태스크들(예컨대, 교통, 내비게이션, 음식, 금융, 선물들 등)이다.
- [0018] [0023] 몇몇 구현들에서, 보이스 어시스턴트는 캐스팅 디바이스의 선택적인 특성으로서 제공되며, 보이스 어시스턴트 기능은 캐스팅 디바이스의 일부로서 업데이트될 수 있다.
- [0019] [0024] 몇몇 구현들에서, 사용자들로부터의 보이스 커맨드들 및 구두 입력들 내의 핫워드들 또는 키워드들의 검출은 애플리케이션 프로세서에 의해 수행(예컨대, 사용자가 보이스 커맨드 또는 구두 입력을 말한 클라이언트 디바이스 또는 캐스팅 디바이스에서 수행)된다. 몇몇 구현들에서, 핫워드들의 검출은 외부 디지털 신호 프로세서에 의해 수행(예컨대, 사용자가 보이스 커맨드 또는 구두 입력을 말한 클라이언트 또는 캐스팅 디바이스와는 대조적으로 보이스 커맨드들을 프로세싱하는 서버 시스템에 의해 수행)된다.
- [0020] [0025] 몇몇 구현들에서, 보이스 어시스턴트 특성을 갖는 디바이스는, 원거리 지원, "푸시 투 어시스턴트(push to assist)" 또는 "푸시 투 토크(push to talk)"(예컨대, 보이스 어시스턴트 기능을 개시하기 위한 버튼), 및 AC 전력 중 하나 이상을 포함한다.
- [0021] [0026] 몇몇 구현들에서, 보이스 어시스턴트는, 오디오 입력(예컨대, 마이크로폰, 진행중인 재생을 위한 미디어 루프백(loopback)), 마이크로폰 상태(예컨대, 온/오프), 더킹(ducking)(예컨대, 어시스턴트가 핫워드 또는 푸시 투 토크 중 어느 하나를 통해 트리거링될 경우 모든 출력들의 볼륨을 감소시키는 것), 및 새로운 어시스턴트 이벤트들 및 상태 메시지들(예컨대, 어시스턴트가 트리거링되었음(예컨대, 핫워드를 들었음, 어시스턴트 버튼을 눌렀음), 스피치를 청취함, 서버 상에서 대기함, 응답, 응답 완료됨, 알람/타이머가 재생중임) 중 하나 이상에 대한 애플리케이션 프로그래밍 인터페이스(API)들을 포함한다.
- [0022] [0027] 몇몇 구현들에서, 보이스 어시스턴트 기능을 갖는 디바이스는, 디바이스 상에서 보이스 어시스턴트의 기능을 인에이블링시키거나 용이하게 하도록(예컨대, 디바이스 상에서 보이스 어시스턴트 기능을 셋업하고, 튜토리얼들을 사용자에게 제공하도록) 구성 목적들을 위해 (예컨대, 스마트폰 상의 구성 애플리케이션을 이용하여) 다른 디바이스와 통신할 수 있다. 구성들 또는 셋업들은, 디바이스 위치를 특정하는 것, 사용자 계정과의 연관성, 보이스 제어에 대한 사용자 옵트-인(opt-in), 미디어 서비스들(예컨대, 비디오 스트리밍 서비스들, 음악 스트리밍 서비스들)에 연결되고 미디어 서비스들을 우선순위화하는 것, 흡 자동화 구성들 등을 포함할 수 있다.
- [0023] [0028] 몇몇 구현들에서, 보이스 어시스턴트를 갖는 디바이스는 사용자에 대한 하나 이상의 사용자 인터페이스 엘리먼트들 또는 표시들을 포함할 수 있다. 사용자 인터페이스 엘리먼트들 중 하나 이상은 (예컨대, 하나 이상의 LED들을 사용하여 디스플레이되는 광 패턴들로서, 스피커에 의해 출력된 사운드 패턴들로서) 물리적이며, 핫워드에 의존하지 않는 "푸시 투 어시스턴트" 또는 "푸시 투 토크" 트리거, "마이크로폰 음소거" 트리거 및 시각적인 상태 표시, 즉 "핫워드 상태 대기"의 시각적인 표시, "핫워드 검출"의 시각적인 표시, 일정 거리(예컨대, 15 피트(feet))에서 가시적인 "어시스턴트가 능동적으로 청취중임"의 시각적인 표시, "어시스턴트가 작동중/생각중임"의 시각적인 표시, "보이스 메시지/통지가 이용가능함"의 시각적인 표시, "볼륨 레벨"의 제어 방법 및 상태 표시자, 및 "일시정지/재개"의 제어 방법 중 하나 이상을 포함할 수 있다. 몇몇 구현들에서, 이를 물리적인 사

용자 인터페이스 엘리먼트들은 클라이언트 디바이스 또는 캐스팅 디바이스에 의해 제공된다. 몇몇 구현들에서, 보이스 어시스턴트는 상이한 디바이스들에 걸친 경험의 일관성을 위해 상이한 디바이스들에 걸쳐 공통 세트의 사용자 인터페이스 엘리먼트들 또는 표시들을 지원한다.

[0024] 몇몇 구현들에서, 보이스 어시스턴트는 디바이스-특정 커맨드들 및/또는 핫워드들 뿐만 아니라 표준화되고 미리 정의된 세트의 커맨드들 및/또는 핫워드들을 지원한다.

[0025] [0030] 도 1은 몇몇 구현들에 따른 네트워크 환경(100)을 예시한다. 네트워크 환경(100)은 캐스팅 디바이스(106) 및/또는 보이스 어시스턴트 클라이언트 디바이스(104)를 포함한다. 캐스팅 디바이스(106)(예컨대, 구글 사의 크롬캐스트)는 오디오 입력 디바이스(108)(예컨대, 마이크로폰) 및 오디오 출력 디바이스(110)(예컨대, 하나 이상의 스피커들)에 직접 또는 그렇지 않으면 통신가능하게 커플링된다. 몇몇 구현들에서, 오디오 입력 디바이스(108) 및 오디오 출력 디바이스(110) 둘 모두는 캐스팅 디바이스(106)에 통신가능하게 커플링된 디바이스(예컨대, 스피커 시스템, 텔레비전, 사운드 바)의 컴포넌트들이다. 몇몇 구현들에서, 오디오 입력 디바이스(108)는 캐스팅 디바이스(106)의 컴포넌트이고, 오디오 출력 디바이스(110)는 캐스팅 디바이스(106)가 통신가능하게 커플링된 디바이스의 컴포넌트이며, 그 역도 가능하다. 몇몇 구현들에서, 오디오 입력 디바이스(108) 및 오디오 출력 디바이스(110)는 캐스팅 디바이스(106)의 컴포넌트들이다.

[0026] [0031] 몇몇 구현들에서, 캐스팅 디바이스(106)는 클라이언트(102)에 통신가능하게 커플링된다. 클라이언트(102)는 보이스 어시스턴트 특성들을 포함하는 캐스팅 디바이스(106)의 구성을 용이하게 하는 애플리케이션 또는 모듈(예컨대, 캐스팅 디바이스 셋팅 앱)을 포함할 수 있다.

[0027] [0032] 몇몇 구현들에서, 캐스팅 디바이스(106)는 디스플레이(144)에 커플링된다.

[0028] [0033] 몇몇 구현들에서, 캐스팅 디바이스(106)는 하나 이상의 시각적인 표시자들(142)(예컨대, LED 광들)을 포함한다.

[0029] [0034] 몇몇 구현들에서, 캐스팅 디바이스(106)는 수신기 모듈(146)을 포함한다. 몇몇 구현들에서, 수신기 모듈(146)은, 예컨대 하드웨어 기능들을 포함하고 콘텐츠 소스와 통신하는 캐스팅 디바이스(106)를 동작시킨다. 몇몇 구현들에서, 상이한 콘텐츠 소스들에 대해 캐스팅 디바이스(106)에 상이한 수신기 모듈들(146)이 존재한다. 몇몇 구현들에서, 수신기 모듈(146)은 상이한 콘텐츠 소스들에 대한 각각의 서브-모듈들을 포함한다.

[0030] [0035] 보이스 어시스턴트 클라이언트 디바이스(104)(예컨대, 구글 사의 구글 어시스턴트, 구글 사의 구글 홈을 갖는 스마트폰, 랩톱 또는 데스크톱 컴퓨터, 태블릿 컴퓨터, 보이스 커맨드 디바이스, 모바일 디바이스 또는 차량내 시스템)는 오디오 입력 디바이스(132)(예컨대, 마이크로폰) 및 오디오 출력 디바이스(134)(예컨대, 하나 이상의 스피커들, 헤드폰들)를 포함한다. 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104)(예컨대, 구글 사의 구글 어시스턴트, 구글 사의 구글 홈을 갖는 보이스 커맨드 디바이스, 모바일 디바이스 또는 차량내 시스템)는 클라이언트(140)(예컨대, 스마트폰, 태블릿 디바이스)에 통신가능하게 커플링된다. 클라이언트(140)는 보이스 어시스턴트 특성들을 포함하는 보이스 어시스턴트 클라이언트 디바이스(104)의 구성을 용이하게 하는 애플리케이션 또는 모듈(예컨대, 보이스 커맨드 디바이스 셋팅 앱)을 포함할 수 있다.

[0031] [0036] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104)는 하나 이상의 시각적인 표시자들(152)(예컨대, LED 광들)을 포함한다. 시각적인 표시자들(예컨대, LED 광들)을 갖는 보이스 어시스턴트 클라이언트 디바이스의 일 예는, 발명의 명칭이 "LED Design Language for Visual Affordance of Voice User Interfaces"으로 2016년 5월 13일자로 출원되었고, 그 전체가 본 명세서에 인용에 의해 포함되는 미국 출원 제 62/336,566호의 도 4a에 예시된다.

[0032] [0037] 캐스팅 디바이스(106) 및 보이스 어시스턴트 클라이언트 디바이스(104)는 보이스 어시스턴트 모듈 또는 라이브러리(136)의 각각의 인스턴스들을 포함한다. 보이스 어시스턴트 모듈/라이브러리(136)는 다양한 디바이스들(예컨대, 캐스팅 디바이스(106), 보이스 어시스턴트 클라이언트 디바이스(104))에 걸쳐 보이스 어시스턴트 기능을 구현하는 모듈/라이브러리이다. 보이스 어시스턴트 기능은 디바이스-특정 특성들(예컨대, 보이스 어시스턴트를 통해 디바이스-특정 특성들을 제어하기 위한 지원)을 여전히 허용하면서 다양한 디바이스들에 걸쳐 일관적이다. 몇몇 구현들에서, 보이스 어시스턴트 모듈 또는 라이브러리(136)는 디바이스들에 걸쳐 동일하거나 또는 유사하며; 동일한 라이브러리의 인스턴스들은 다양한 디바이스들에 포함될 수 있다.

[0033] [0038] 몇몇 구현들에서, 디바이스의 탑재에 의존하여, 보이스 어시스턴트 모듈/라이브러리(136)는 디바이스에 설치된 애플리케이션에, 디바이스 운영 시스템에 포함되거나, 또는 디바이스에 임베딩(예컨대, 펌웨어에

임베딩)된다.

- [0034] [0039] 몇몇 구현들에서, 캐스팅 디바이스(106)의 보이스 어시스턴트 모듈/라이브러리(136-1)는 보이스 어시스턴트 동작들을 수행하기 위해 수신기 모듈(146)과 통신한다.
- [0035] [0040] 몇몇 구현들에서, 캐스팅 디바이스(106)의 보이스 어시스턴트 모듈/라이브러리(136-1)는 시각적인 표시자들(142)을 제어하거나 또는 그렇지 않으면 시각적인 표시자들(142)에 영향을 줄 수 있다.
- [0036] [0041] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104)의 보이스 어시스턴트 모듈/라이브러리(136-2)는 시각적인 표시자들(152)을 제어하거나 또는 그렇지 않으면 시각적인 표시자들(152)에 영향을 줄 수 있다.
- [0037] [0042] 캐스팅 디바이스(106) 및 보이스 어시스턴트 클라이언트 디바이스(104)는 하나 이상의 통신 네트워크들(112)(예컨대, 로컬 영역 네트워크들, 광역 네트워크들, 인터넷)을 통해 서버 시스템(114)에 통신가능하게 커플링된다. 보이스 어시스턴트 모듈/라이브러리(136)는, 오디오 입력 디바이스(108/132)에 의해 꺽업(pick up)된(예컨대, 캡처된) 구두 입력을 검출(예컨대, 수신)하고, (예컨대, 핫워드들을 검출하기 위해) 구두 입력을 프로세싱하며, 프로세싱된 구두 입력 또는 프로세싱된 구두 입력의 인코딩을 서버(114)에 송신한다. 서버(114)는 프로세싱된 구두 입력 또는 그의 인코딩을 수신하고, 수신된 구두 입력을 프로세싱하여, 구두 입력에 대한 적절한 응답을 결정한다. 적절한 응답은 기능 또는 동작을 수행하기 위한 캐스팅 디바이스(106) 또는 보이스 어시스턴트 클라이언트 디바이스(104)에 대한 콘텐츠, 정보, 또는 명령들 또는 커맨드들 또는 메타데이터일 수 있다. 서버(114)는 응답을 캐스팅 디바이스(106) 또는 보이스 어시스턴트 클라이언트 디바이스(104)에 전송하며, 여기서, 콘텐츠 또는 정보가 출력되고(예컨대, 오디오 출력 디바이스(110/134)를 통해 출력되고) 그리고/또는 기능이 수행된다. 프로세싱의 일부로서, 서버(114)는 응답에 대한 콘텐츠 또는 정보, 또는 그에 대한 참조들을 획득하도록 하나 이상의 콘텐츠 또는 정보 소스들(138)과 통신할 수 있다. 몇몇 구현들에서, 콘텐츠 또는 정보 소스들(138)은, 예컨대, 검색 엔진들, 데이터베이스들, 사용자의 계정과 연관된 정보(예컨대, 캘린더, 태스크 리스트, 이메일), 웹사이트들, 및 미디어 스트리밍 서비스들을 포함한다. 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104) 및 캐스팅 디바이스(106)는 서로 통신하거나 또는 상호작용할 수 있다. 그러한 통신 또는 상호작용의 예들 뿐만 아니라 보이스 어시스턴트 클라이언트 디바이스(104)(예컨대, 구글 사의 구글 홈)의 예시적인 동작들은, 발명의 명칭이 "LED Design Language for Visual Affordance of Voice User Interfaces"으로 2016년 5월 13일자로 출원된 미국 출원 제 62/336,566호, 발명의 명칭이 "Voice-Controlled Closed Caption Display"으로 2016년 5월 13일자로 출원된 미국 출원 제 62/336,569호, 및 발명의 명칭이 "Media Transfer among Media Output Devices"으로 2016년 5월 13일자로 출원된 미국 출원 제 62/336,565호에 설명되어 있으며, 이들 출원들 모두는 그들 전체가 본 명세서에 인용에 의해 포함된다.
- [0038] [0043] 몇몇 구현들에서, 보이스 어시스턴트 모듈/라이브러리(136)는 오디오 입력 디바이스(108/132)에 의해 캡처된 구두 입력을 수신하고, (어떠한 프로세싱도 없거나 프로세싱이 거의 없는) 구두 입력 또는 그의 인코딩을 서버(114)에 송신한다. 서버(114)는 구두 입력을 프로세싱하여 핫워드들을 검출하고, 적절한 응답을 결정하며, 응답을 캐스팅 디바이스(106) 또는 보이스 어시스턴트 클라이언트 디바이스(104)에 전송한다.
- [0039] [0044] 구두 입력이 캐스팅 디바이스(106) 또는 보이스 어시스턴트 클라이언트 디바이스(104)가 기능을 수행하기 위한 커맨드를 포함한다고 서버(114)가 결정하면, 서버(114)는 응답으로, 캐스팅 디바이스(106) 또는 보이스 어시스턴트 클라이언트 디바이스(104)가 그 기능을 수행하도록 명령하는 명령들 또는 메타데이터를 송신한다. 기능은 디바이스에 특정할 수 있으며, 보이스 어시스턴트에서 그러한 기능들을 지원하기 위한 능력은 보이스 어시스턴트 모듈/라이브러리(136)에 부가된 또는 연결된 맞춤형 모듈 또는 기능으로서 캐스팅 디바이스(106) 또는 클라이언트(104)에 포함될 수 있다.
- [0040] [0045] 몇몇 구현들에서, 서버(114)는, 구두 입력 프로세싱 동작들을 수행하고 구두 입력들에 대한 응답들을 결정하는 보이스 프로세싱 후단(148)을 포함하거나 그에 커플링된다.
- [0041] [0046] 몇몇 구현들에서, 서버(114)는 다운로딩가능한 보이스 어시스턴트 라이브러리(150)를 포함한다. 다운로딩가능한 보이스 어시스턴트 라이브러리(150)(예컨대, 보이스 어시스턴트 라이브러리(136)와 동일한 것, 또는 그의 업데이트)는 새로운 특성들 및 기능 또는 업데이트들을 포함할 수 있으며, 보이스 어시스턴트 라이브러리를 디바이스에 부가하거나 또는 보이스 어시스턴트 라이브러리(136)를 업데이트하기 위해 다운로딩될 수 있다.
- [0042] [0047] 도 2는 몇몇 구현들에 따른, 네트워크 환경(100)의 예시적인 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)를 예시한 블록 다이어그램이다. 보이스 어시스턴트 클라이언트 디바이스

(104)의 예들은, 모바일 폰, 태블릿 컴퓨터, 램프 컴퓨터, 데스크톱 컴퓨터, 무선 스피커(예컨대, 구글 사의 구글 홈), 보이스 커맨드 디바이스(예컨대, 구글 사의 구글 홈), 텔레비전, 사운드바, 캐스팅 디바이스(예컨대, 구글 사의 크롬캐스트), 미디어 스트리밍 디바이스, 가정용 전자기기, 소비자 전자 디바이스, 차량내 시스템, 및 웨어러블 개인용 디바이스를 포함하지만 이에 제한되지는 않는다. 보이스 어시스턴트 클라이언트 디바이스(104)(예컨대, 구글 사의 구글 홈, 구글 어시스턴트 능력을 갖는 모바일 디바이스) 또는 캐스팅 디바이스(106)(예컨대, 구글 사의 크롬캐스트)는 통상적으로, 하나 이상의 프로세싱 유닛(CPU)들(202), 하나 이상의 네트워크 인터페이스들(204), 메모리(206), 및 이를 컴포넌트들을 상호연결시키기 위한 하나 이상의 통신 버스들(208)을 포함(종종 칩셋으로 지칭됨)한다. 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는, 오디오 입력 디바이스(108 또는 132)(예컨대, 보이스-커맨드 입력 유닛 또는 마이크로폰) 및 선택적으로는 다른 입력 디바이스들, 이를테면 키보드, 마우스, 터치 스크린 디스플레이, 터치-감응형 입력 패드, 제스처 캡처 카메라, 또는 다른 입력 버튼들 또는 제어들을 포함하여 사용자 입력을 용이하게 하는 하나 이상의 입력 디바이스들(210)을 포함한다. 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(102)는 키보드를 보완하거나 또는 대체하기 위해 마이크로폰 및 보이스 인식, 또는 카메라 및 제스처 인식을 사용한다. 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 또한, 오디오 출력 디바이스(110 또는 134)(예컨대, 하나 이상의 스피커들, 헤드폰들 등) 및 선택적으로는 사용자 인터페이스들의 프리젠테이션을 가능하게 하고 콘텐츠 및 정보를 디스플레이하는 하나 이상의 시각적인 디스플레이들(예컨대 디스플레이(144)) 및/또는 하나 이상의 시각적인 표시자들(142 또는 152)(예컨대, LED들)을 포함하는 하나 이상의 출력 디바이스들(212)을 포함한다. 선택적으로, 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)의 위치를 결정하기 위한 위치 검출 유닛(214), 이를테면 GPS(글로벌 포지셔닝 위성) 또는 다른 지오-로케이션(geo-location) 수신기를 포함한다. 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 선택적으로 또한, 다른 오브젝트들(예컨대, 웨어러블 개인용 디바이스의 경우에는 사용자-착용자)에 대한 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)의 근접도를 결정하기 위한 근접도 검출 디바이스(215), 예컨대 IR 센서를 포함할 수 있다. 선택적으로, 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 센서(들)(213)(예컨대, 가속도계, 자이로스코프 등)를 포함한다.

[0043]

[0048] 메모리(206)는 고속 랜덤 액세스 메모리, 이를테면 DRAM, SRAM, DDR RAM, 또는 다른 랜덤 액세스 솔리드 스테이트 메모리 디바이스들을 포함하고; 그리고 선택적으로는, 비-휘발성 메모리, 이를테면 하나 이상의 자기 디스크 저장 디바이스들, 하나 이상의 광 디스크 저장 디바이스들, 하나 이상의 플래시 메모리 디바이스들, 또는 하나 이상의 다른 비-휘발성 솔리드 스테이트 저장 디바이스들을 포함한다. 메모리(206)는 선택적으로, 하나 이상의 프로세싱 유닛들(202)로부터 원격으로 로케이팅된 하나 이상의 저장 디바이스들을 포함한다. 메모리(206) 또는 대안적으로는 메모리(206) 내의 비-휘발성 메모리는 비-일시적인 컴퓨터 판독가능 저장 매체를 포함한다. 몇몇 구현들에서, 메모리(206) 또는 메모리(206)의 비-일시적인 컴퓨터 판독가능 저장 매체는 다음의 프로그램들, 모듈들, 및 데이터 구조들, 또는 이들의 서브세트 또는 슈퍼세트를 저장한다:

[0044]

- 다양한 기본적인 시스템 서비스들을 핸들링하고 하드웨어 종속 테스크들을 수행하기 위한 절차들을 포함하는 운영 시스템(216);

[0045]

- 하나 이상의 네트워크 인터페이스들(204)(유선 또는 무선) 및 하나 이상의 네트워크들(112), 이를테면 인터넷, 다른 광역 네트워크들, 로컬 영역 네트워크들, 대도시 영역 네트워크들 등을 통해 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)를 다른 디바이스들(예컨대, 서버 시스템(114), 클라이언트(102, 140), 다른 보이스 어시스턴트 클라이언트 디바이스들(104) 또는 캐스팅 디바이스(106))에 연결시키기 위한 네트워크 통신 모듈(218);

[0046]

- 하나 이상의 출력 디바이스들(212)(예컨대, 디스플레이들, 스피커들 등)을 통해 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)에서 정보의 프리젠테이션을 가능하게 하기 위한 사용자 인터페이스 모듈(220);

[0047]

- 하나 이상의 입력 디바이스들(210)에 의해 캡처 또는 수신된 하나 이상의 사용자 입력들 또는 상호작용들을 프로세싱하고 입력 또는 상호작용을 해석하기 위한 입력 프로세싱 모듈(222);

[0048]

- 구두 입력들을 프로세싱하고, 구두 입력들을 서버(114)에 제공하고, 서버(114)로부터 응답들을 수신하며, 응답들을 출력하기 위한 보이스 어시스턴트 모듈(136); 및

[0049]

- 보이스 어시스턴트 모듈(136)과 연관된 데이터를 적어도 저장하고, 다음을 포함하는 클라이언트 데이터

(226):

[0050] ○ 보이스 어시스턴트 모듈(136) 및 보이스 어시스턴트 기능에 대한 설정들 및 구성들과 연관된 정보를 저장하기 위한 보이스 어시스턴트 설정들(228);

[0051] ○ 콘텐츠 또는 정보의 미리 정의된 그리고/또는 사용자-특정된 소스들 및 카테고리들을 저장하기 위한 콘텐츠/정보 소스들(230) 및 카테고리들(232);

[0052] ○ 보이스 어시스턴트 모듈(136)의 동작 및 사용, 이를테면, 수신된 커맨드들 및 요청들, 커맨드들 및 요청들에 대한 응답들, 커맨드들 및 요청들에 대한 응답으로 수행된 동작들 등과 연관된 정보(예컨대, 로그들)를 저장하기 위한 사용 이력(234);

[0053] ○ 콘텐츠/정보 소스들(230)에서 사용자들의 각각의 계정들에 액세스하기 위한 하나 이상의 사용자들의 인가들 및 인증 정보, 및 그 인가된 계정들에 대한 계정 정보를 저장하기 위한 사용자 계정들 및 인가들(236); 및

[0054] ○ 재생을 위한 콘텐츠를 수신하기 위해 콘텐츠 소스들과 통신하는 것을 포함하여, 캐스팅 디바이스(106)의 캐스팅 기능을 동작시키기 위한 수신기 모듈(146).

[0049] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 보이스 어시스턴트 및 관련된 기능에 대한 하나 이상의 라이브러리들 및 하나 이상의 애플리케이션 프로그래밍 인터페이스(API)들을 포함한다. 이들 라이브러리들은 보이스 어시스턴트 모듈(136) 또는 수신기 모듈(146)에 포함되거나 또는 그들에 의해 연결될 수 있다. 라이브러리들은, 보이스 어시스턴트 기능과 연관된 모듈들 또는 보이스 어시스턴트 기능을 용이하게 했던 다른 기능들을 포함한다. API들은 보이스 어시스턴트 기능을 용이하게 하는 하드웨어 및 다른 소프트웨어(예컨대, 운영 시스템, 다른 애플리케이션들)에 대한 인터페이스들을 제공한다. 예컨대, 보이스 어시스턴트 클라이언트 라이브러리(240), 디버깅 라이브러리(242), 플랫폼 API들(244), 및 포직스(POSIX) API들(246)이 메모리(206)에 저장될 수 있다. 이들 라이브러리들 및 API들은 도 4를 참조하여 아래에서 추가로 설명된다.

[0050] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는, 보이스 어시스턴트 클라이언트 라이브러리(240)의 모듈들 및 기능들을 사용하는 보이스 애플리케이션(250), 및 선택적으로는 디버깅 라이브러리(242), 플랫폼 API들(244), 및 포직스 API들(246)을 포함한다. 몇몇 구현들에서, 보이스 애플리케이션(250)은 보이스 어시스턴트 클라이언트 라이브러리(240) 등의 사용을 통해 보이스-인에이블되는 퍼스트-파티 또는 서드-파티 애플리케이션이다.

[0051] 위에서 식별된 엘리먼트들 각각은 이전에 언급된 메모리 디바이스들 중 하나 이상에 저장될 수 있으며, 위에서 설명된 기능을 수행하기 위한 명령들의 세트에 대응한다. 위에서 식별된 모듈들 또는 프로그램들(즉, 명령들의 세트들)은 별개의 소프트웨어 프로그램들, 절차들, 모듈들 또는 데이터 구조들로서 구현될 필요는 없으며, 따라서, 이들 모듈들의 다양한 서브세트들은 다양한 구현들에서 조합되거나 또는 그렇지 않으면 재배열될 수 있다. 몇몇 구현들에서, 메모리(206)는 선택적으로, 위에서 식별된 모듈들 및 데이터 구조들의 서브세트를 저장한다. 더욱이, 메모리(206)는 선택적으로, 위에서 설명되지 않은 부가적인 모듈들 및 데이터 구조들을 저장한다.

[0052] 도 3은 몇몇 구현들에 따른, 네트워크 환경(100)의 예시적인 서버 시스템(114)을 예시한 블록 다이어그램이다. 서버(114)는 통상적으로, 하나 이상의 프로세싱 유닛(CPU)들(302), 하나 이상의 네트워크 인터페이스들(304), 메모리(306), 및 이를 컴포넌트들을 상호연결시키기 위한 하나 이상의 통신 버스들(308)을 포함(종종 칩셋으로 지칭됨)한다. 서버(114)는 선택적으로, 사용자 입력을 용이하게 하는 하나 이상의 입력 디바이스들(310), 이를테면 키보드, 마우스, 보이스-커맨드 입력 유닛 또는 마이크로폰, 터치 스크린 디스플레이, 터치-감응형 입력 패드, 제스처 캡처 카메라, 또는 다른 입력 버튼들 또는 제어들을 포함한다. 더욱이, 서버(114)는 키보드를 보완하거나 또는 대체하기 위해 마이크로폰 및 보이스 인식 또는 카메라 및 제스처 인식을 사용할 수 있다. 몇몇 구현들에서, 서버(114)는 선택적으로, 예컨대, 전자 디바이스 상에 인쇄된 그래픽 시리즈 코드들의 이미지들을 캡처하기 위한 하나 이상의 카메라들, 스캐너들, 또는 포토 센서 유닛들을 포함한다. 서버(114)는 선택적으로 또한, 사용자 인터페이스들의 프리젠테이션을 가능하게 하고 콘텐츠를 디스플레이하며, 하나 이상의 스피커들 및/또는 하나 이상의 시작적인 디스플레이들을 포함하는 하나 이상의 출력 디바이스들(312)을 포함한다.

[0053] 메모리(306)는 고속 랜덤 액세스 메모리, 이를테면 DRAM, SRAM, DDR RAM, 또는 다른 랜덤 액세스 솔리드

스테이트 메모리 디바이스들을 포함하고; 그리고 선택적으로는, 비-휘발성 메모리, 이를테면 하나 이상의 자기 디스크 저장 디바이스들, 하나 이상의 광 디스크 저장 디바이스들, 하나 이상의 플래시 메모리 디바이스들, 또는 하나 이상의 다른 비-휘발성 솔리드 스테이트 저장 디바이스들을 포함한다. 메모리(306)는 선택적으로, 하나 이상의 프로세싱 유닛들(302)로부터 원격으로 로케이팅된 하나 이상의 저장 디바이스들을 포함한다. 메모리(306) 또는 대안적으로는 메모리(306) 내의 비-휘발성 메모리는 비-일시적인 컴퓨터 판독가능 저장 매체를 포함한다. 몇몇 구현들에서, 메모리(306) 또는 메모리(306)의 비-일시적인 컴퓨터 판독가능 저장 매체는 다음의 프로그램들, 모듈들, 및 데이터 구조들, 또는 이들의 서브세트 또는 슈퍼세트를 저장한다:

- [0060] ● 다양한 기본적인 시스템 서비스들을 핸들링하고 하드웨어 종속 태스크들을 수행하기 위한 절차들을 포함하는 운영 시스템(316);
- [0061] ● 하나 이상의 네트워크 인터페이스들(304)(유선 또는 무선) 및 하나 이상의 네트워크들(112), 이를테면 인터넷, 다른 광역 네트워크들, 로컬 영역 네트워크들, 대도시 영역 네트워크들 등을 통해 서버 시스템(114)을 다른 디바이스들(예컨대, 보이스 어시스턴트 클라이언트 디바이스(104), 캐스팅 디바이스(106), 클라이언트(102), 클라이언트(140))에 연결시키기 위한 네트워크 통신 모듈(318);
- [0062] ● 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)의 위치 정보에 기반하여 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)의 근접도 및/또는 위치를 결정하기 위한 근접도/위치 결정 모듈(320);
- [0063] ● 보이스 어시스턴트 구두 입력들(예컨대, 보이스 어시스턴트 클라이언트 디바이스들(104) 및 캐스팅 디바이스들(106)로부터 수신된 구두 입력들)을 프로세싱하고, 다음 중 하나 이상을 포함하는 보이스 어시스턴트 후단(116):

 - 구두 입력들에서 커맨드들 및 요청들을 식별하도록 구두 입력들을 프로세싱하기 위한 구두 입력 프로세싱 모듈(324);
 - 커맨드들 및 요청들에 대한 응답으로 콘텐츠 및 정보를 수집하기 위한 콘텐츠/정보 수집 모듈(326); 및
 - 커맨드들 및 요청들에 대한 응답으로 구두 출력들을 생성하고, 응답 콘텐츠 및 정보로 구두 출력들을 채우기 위한 응답 생성 모듈(328);

- [0067] ● 보이스 어시스턴트 플랫폼의 동작과 연관된 데이터를 적어도 저장하고 다음을 포함하는 서버 시스템 데이터(330):

 - 보이스 어시스턴트 플랫폼의 사용자들과 연관된 정보를 저장하고, 다음을 포함하는 사용자 데이터(332);
 - 보이스 어시스턴트 셋팅들(228)에 대응하는 보이스 어시스턴트 셋팅 정보, 및 콘텐츠/정보 소스들(230) 및 카테고리들(232)에 대응하는 정보를 저장하기 위한 사용자 보이스 어시스턴트 셋팅들(334);
 - 커맨드들 및 요청들 및 대응하는 응답들의 이력들을 포함하는 보이스 어시스턴트와 관련된 사용자들의 이력들(예컨대, 로그들)을 저장하기 위한 사용자 이력(336); 및
 - 콘텐츠/정보 소스들(230)에서 사용자들의 각각의 계정들에 액세스하기 위한 사용자들의 인가들 및 인증 정보, 및 사용자 계정들 및 인가들(236)에 대응하는 그 인가된 계정들에 대한 계정 정보를 저장하기 위한 사용자 계정들 및 인가들(338).

- [0072] [0054] 위에서 식별된 엘리먼트들 각각은 이전에 언급된 메모리 디바이스들 중 하나 이상에 저장될 수 있으며, 위에서 설명된 기능을 수행하기 위한 명령들의 세트에 대응한다. 위에서 식별된 모듈들 또는 프로그램들(즉, 명령들의 세트들)은 별개의 소프트웨어 프로그램들, 절차들, 모듈들 또는 데이터 구조들로서 구현될 필요는 없으며, 따라서, 이들 모듈들의 다양한 서브세트들은 다양한 구현들에서 조합되거나 또는 그렇지 않으면 재배열될 수 있다. 몇몇 구현들에서, 메모리(306)는 선택적으로, 위에서 식별된 모듈들 및 데이터 구조들의 서브세트를 저장한다. 더욱이, 메모리(306)는 선택적으로, 위에서 설명되지 않은 부가적인 모듈들 및 데이터 구조들을 저장한다.
- [0073] [0055] 몇몇 구현들에서, 보이스 어시스턴트 모듈(136)(도 2)은 하나 이상의 라이브러리들을 포함한다. 라이브러리는 각각의 기능들을 수행하는 모듈들 또는 서브-모듈들을 포함한다. 예컨대, 보이스 어시스턴트 클라이언트

트 라이브러리는 보이스 어시스턴트 기능들을 수행하는 모듈들을 포함한다. 보이스 어시스턴트 모듈(136)은 또한, 특정 하드웨어(예컨대, 클라이언트 또는 캐스팅 디바이스 상의 하드웨어), 특정 운영 소프트웨어, 또는 원격 시스템들과 협업하기 위한 하나 이상의 애플리케이션 프로그래밍 인터페이스(API)들을 포함할 수 있다.

[0074] [0056] 몇몇 구현들에서, 라이브러리는 예컨대, 대역통과, 필터링, 삭제, 및 핫워드 검출을 포함하는 오디오 신호 프로세싱 동작들을 지원하는 모듈들을 포함한다. 몇몇 구현들에서, 라이브러리는 후단(예컨대, 서버-기반) 스피치 프로세싱 시스템들에 연결되기 위한 모듈들을 포함한다. 몇몇 구현들에서, 라이브러리는 디버깅(예컨대, 스피치 인식 디버깅, 하드웨어 이슈 디버깅, 자동화된 테스팅)을 위한 모듈들을 포함한다.

[0075] [0057] 도 4는 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)에 저장되고, 보이스 어시스턴트 모듈(136) 또는 다른 애플리케이션에 의해 구동될 수 있는 라이브러리들 및 API들을 예시한다. 라이브러리들 및 API들은 보이스 어시스턴트 클라이언트 라이브러리(240), 디버깅 라이브러리(242), 플랫폼 API(244), 및 포직스 API들(246)을 포함할 수 있다. 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)의 애플리케이션들(예컨대, 보이스 어시스턴트 모듈(136), 또는 보이스 어시스턴트와의 협업을 지원하기를 원할 수 있는 다른 애플리케이션들)은 애플리케이션에서 보이스 어시스턴트 기능을 제공 또는 지원하기 위해 라이브러리들 및 API들을 포함하거나, 그들에 연결되거나, 또는 그들을 구동시킬 수 있다. 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240) 및 디버깅 라이브러리(242)는 별개의 라이브러리들이며; 보이스 어시스턴트 클라이언트 라이브러리(240) 및 디버깅 라이브러리(242)의 라이브러리들을 별개로 유지하는 것은, 이를 라이브러리들의 상이한 보안 함의(implication)들을 고려하는 상이한 릴리즈 및 업데이트 절차들을 용이하게 한다.

[0076] [0058] 몇몇 구현들에서, 라이브러리들은 유연하며; 라이브러리들은 다수의 디바이스 타입들에 걸쳐 사용되고, 동일한 보이스 어시스턴트 기능을 통합할 수 있다.

[0077] [0059] 몇몇 구현들에서, 라이브러리들은 표준 공유 오브젝트들(예컨대, 표준 리눅스 공유 오브젝트들)에 의존하며, 따라서, 이들 표준 공유 오브젝트들을 사용하는 상이한 운영 시스템들 또는 플랫폼들(예컨대, 임베디드 리눅스의 다양한 리눅스 배포들 및 특색들)과 호환가능하다.

[0078] [0060] 몇몇 구현들에서, 포직스 API들(246)은 다양한 운영 시스템들과의 호환성을 위해 표준 API들을 제공한다. 따라서, 보이스 어시스턴트 클라이언트 라이브러리(240)는 상이한 포직스-호환가능 운영 시스템들의 디바이스들에 포함될 수 있으며, 포직스 API들(246)은 보이스 어시스턴트 클라이언트 라이브러리(240)와 상이한 운영 시스템들 사이에 호환 인터페이스를 제공한다.

[0079] [0061] 몇몇 구현들에서, 라이브러리들은 보이스 어시스턴트를 구현하는 상이한 타입들의 디바이스들에 걸쳐 이용가능한 기본 사용 경우들(예컨대, 타이머들, 알람들, 볼륨 제어)을 지원하고 용이하게 하기 위한 모듈들을 포함한다.

[0080] [0062] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240)는, 보이스 어시스턴트를 시작하고, 구성하며, 보이스 어시스턴트와 상호작용하기 위한 기능들 또는 모듈들을 포함하는 제어기 인터페이스(402)를 포함한다. 몇몇 구현들에서, 제어기 인터페이스(402)는, 디바이스에서 보이스 어시스턴트를 시작하기 위한 "Start()" 기능 또는 모듈(404); (예컨대, 액션을 보이스 어시스턴트를 통해 실행가능할 수 있도록) 액션을 보이스 어시스턴트에 등록하기 위한 "RegisterAction()" 기능 또는 모듈(406); 업데이트된 셋팅들로 보이스 어시스턴트를 재구성하기 위한 "Reconfigure()" 기능(408); 및 기본적인 이벤트들에 대한 기능들의 세트를 어시스턴트에 등록하기 위한 "RegisterEventObserver()" 기능(410)을 포함한다.

[0081] [0063] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240)는 특정한 보이스 어시스턴트 기능과 연관된 다수의 기능들 또는 모듈들을 포함한다. 예컨대, 핫워드 검출 모듈(412)은 핫워드들을 검출하기 위해 보이스 입력들을 프로세싱한다. 스피치 프로세싱 모듈(414)은 보이스 입력들 내의 스피치를 프로세싱하고, 스피치를 텍스트로 변환하거나 텍스트를 스피치로 변환한다(예컨대, 워드들 및 어구들을 식별, 스피치-텍스트 데이터 변환, 텍스트 데이터-스피치 변환). 액션 프로세싱 모듈(416)은 구두 입력들에 대한 응답으로 액션들 및 동작들을 수행한다. 로컬 타이머들/알람들/볼륨 제어 모듈(418)은 디바이스에서의 알람 시계, 타이머, 및 볼륨 제어 기능 및 보이스 입력에 의한 그들의 제어(예컨대, 디바이스에서 타이머들, 시계들, 알람 시계들을 유지하는 것)를 용이하게 한다. 로깅/메트릭 모듈(420)은 보이스 입력들 및 응답들을 레코딩(예컨대, 로깅)할 뿐만 아니라, 관련 메트릭들(예컨대, 응답 시간, 유휴 시간 등)을 결정 및 레코딩한다. 오디오 입력 프로세싱 모듈(422)은 보이스 입력들의 오디오를 프로세싱한다. MP3 디코딩 모듈(424)은 MP3-인코딩된 오디오를 디코딩한다.

오디오 입력 모듈(426)은 오디오 입력 디바이스(예컨대, 마이크로폰)를 통해 오디오를 캡처한다. 오디오 출력 모듈(428)은 오디오 출력 디바이스(예컨대, 스피커)를 통해 오디오를 출력한다. 이벤트 큐잉(queueing) 및 상태 추적 모듈(430)은 디바이스에서 보이스 어시스턴트와 연관된 이벤트들을 큐잉하고, 디바이스에서 보이스 어시스턴트의 상태를 추적한다.

[0082] [0064] 몇몇 구현들에서, 디버깅 라이브러리(242)는 디버깅을 위한 모듈들 및 기능들을 제공한다. 예컨대, HTTP 서버 모듈(432)은 연결 이슈들의 디버깅을 용이하게 하고, 디버그 서버/오디오 스트리밍 모듈(434)은 오디오 이슈들을 디버깅한다.

[0083] [0065] 몇몇 구현들에서, 플랫폼 API(244)는 보이스 어시스턴트 클라이언트 라이브러리(240)와 디바이스의 하드웨어 기능 사이에 인터페이스를 제공한다. 예컨대, 플랫폼 API는, 디바이스 상에서 버튼 입력들을 캡처하기 위한 버튼 입력 인터페이스(436), 루프백 오디오를 캡처하기 위한 루프백 오디오 인터페이스(438), 메트릭들을 로깅 및 결정하기 위한 로깅 및 메트릭 인터페이스(440), 오디오 입력을 캡처하기 위한 오디오 입력 인터페이스(442), 오디오를 출력하기 위한 오디오 출력 인터페이스(444), 및 보이스 어시스턴트와 상호작용할 수 있는 다른 서비스들로 사용자를 인증하기 위한 인증 인터페이스(446)를 포함한다. 도 4에 도시된 보이스 어시스턴트 클라이언트 라이브러리 구성의 이점은, 그 구성이, 동일하거나 또는 유사한 보이스 프로세싱 기능이 일관된 API들 및 보이스 어시스턴트 기능들의 세트들을 갖는 넓은 범위의 보이스 어시스턴트 디바이스 타입들 상에서 제공될 수 있게 한다는 것이다. 이러한 일관성은 보이스 어시스턴트 애플리케이션들의 이식성 및 보이스 어시스턴트 동작들의 일관성을 지원하며, 이는 차례로, 상이한 디바이스 타입들 상에서 실행되는 보이스 어시스턴트 애플리케이션들 및 기능들에 대해 일관된 사용자 상호작용들 및 익숙함을 촉진시킨다. 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240)의 전부 또는 일부는 서버-기반 보이스 어시스턴트 애플리케이션들(예컨대, 프로세싱을 위해 서버(114)로 송신된 보이스 입력들에 대해 동작되는 서버 애플리케이션들)을 지원하기 위해 서버(114)에서 제공될 수 있다.

[0084] [0066] 제어기(402)("Controller") 및 관련된 클래스들에 대응하는 클래스들 및 기능들의 예시적인 코드가 아래에 나타난다. 이들 클래스들 및 기능들은 다양한 디바이스들 상에서 실행가능한 애플리케이션들에 의하여 공통 API들을 통해 이용될 수 있다.

[0085] [0067] 아래의 클래스 "ActionModule"은 보이스 어시스턴트 서버에 의해 제공되는 커맨드들을 핸들링하기 위해 애플리케이션이 그 자신의 모듈들을 등록하는 것을 용이하게 한다:

```

// The application can register its own software modules
// to handle commands provided by the voice assistant server.

class ActionModule {
public:

// An action result describes if an action was executed successfully.

class Result {
public:
    virtual ~Result() = default;

// Set the action result to indicate success.

    virtual void SetOk() = 0;

// Set the action result to the given response code and human
// readable string.

    virtual void SetError(int response_code,
                          const std::string& str);
};

// Arguments to an action handler.

class Args {
public:
    virtual ~Args() = 0;

// Get the serialized protobuf data for a given typed action handler
// argument.

    virtual bool GetProtobufDataFromType
        (std::string type, std::string* data) = 0;
};

    virtual ~ActionModule() = 0;

// Returns the name of this module.

    virtual std::string GetName() = 0;

```

// Handles the given |action_name| with its |args|, and updates result
 [0086] // according to the outcome of the action execution.

```

    virtual void Handle(std::string action_name,
                       std::unique_ptr<Args> args, Result* result) = 0;

// Sets a named protobuf to given serialized data to indicate to the
// voice assistant the local state of this module.

    virtual bool GetModuleContext(std::string* protobuf_type,
                                 std::string* protobuf_data) = 0;
};

```

[0088] [0068] 아래의 클래스 "BuildInfo"는 (예컨대, 애플리케이션, 플랫폼, 및/또는 디바이스의 식별자들 또는 버전
 넘버들을 이용하여) 보이스 어시스턴트 클라이언트 라이브러리(240) 또는 보이스 어시스턴트 클라이언트 디바이
 스(104) 그 자체를 구동시키는 애플리케이션을 설명하기 위해 사용될 수 있다:

```

// Build information used for describing the application
// running the voice assistant client library. For dedicated voice
// assistant devices, this should describe the device.
// This object will be returned from CreateDefaultBuildInfo, can be
// modified, and then set back on a Settings object.

class BuildInfo {
public:
    virtual ~BuildInfo() = default;

// Sets the application version.
    virtual void SetApplicationVersion(const std::string&
                                         application_version) = 0;

// Sets the install identifier. This must be a device-specific identifier
// that should not be the same as any other device or user identifier.
    virtual void SetInstallId(const std::string&
                             install_id) = 0;

// Sets the platform identifier.
    virtual void SetPlatformId(const std::string&
                             platform_id) = 0;

// Sets the platform version.
    virtual void SetPlatformVersion(const std::string&
                                 platform_version) = 0;

// Sets the device model. Optional.
    virtual void SetDeviceModel(const std::string&
                             device_model) = 0;
};

[0089]

```

[0090] [0069] 아래의 클래스 "EventDelegate"는 기본적인 이벤트들과 연관된 기능들, 이를테면 스피치 인식의 시작, 보이스 응답을 출력하는 보이스 어시스턴트의 시작 및 완료 등을 정의한다:

```

// Receives events from the Assistant library.

class EventDelegate {
public:
    class RecognizedSpeechChangedEvent {
public:
    virtual ~RecognizedSpeechChangedEvent() {}

// Indicates the updated recognized text from the voice assistant. If part
// of OnRecognizedSpeechFinishedEvent, this indicates the final
// recognized text.

    virtual std::string
        GetRecognizedSpeech() = 0;
};

    virtual ~EventDelegate() {}

// Indicates the voice assistant client library is booting up.

    virtual void OnBootingUpEvent() = 0;

// Indicates the hotword was heard.

    virtual void OnHeardHotwordEvent() = 0;

// Indicates recognition of speech has started. Speech recognition will
// continue until OnRecognizingSpeechFinishedEvent is received.

    virtual void OnRecognizingSpeechStartedEvent() = 0;

// Indicates a change to the current hypothesis of recognized speech has
// occurred. |event| indicates the new hypothesis.

    virtual void OnRecognizedSpeechChangedEvent(
        const RecognizedSpeechChangedEvent& event) = 0;

// Indicates the final speech recognition has occurred.
// |event| indicates the final value.

    virtual void OnRecognizingSpeechFinishedEvent(
        const RecognizedSpeechChangedEvent& event) = 0;

```

[0091]

```

// Indicates the voice assistant is starting to respond by voice.
// The voice assistant will be responding until OnRespondingFinishedEvent
// is received.
    virtual void OnRespondingStartedEvent() = 0;

// Indicates the voice assistant has finished responding by voice.
    virtual void OnRespondingFinishedEvent() = 0;

// Indicates the alarm has started sounding. The alarm will continue
// sounding until OnAlarmSoundingFinishedEvent is received.
    virtual void OnAlarmSoundingStartedEvent() = 0;

// Indicates the alarm has finished sounding.
    virtual void OnAlarmSoundingFinishedEvent() = 0;

// Indicates the timer has started sounding. The timer will continue
// sounding until OnTimerSoundingFinishedEvent is received.
    virtual void OnTimerSoundingStartedEvent() = 0;

// Indicates the timer has finished sounding.
    virtual void OnTimerSoundingFinishedEvent() = 0;

// Indicates a volume change has occurred to the default volume (which
// occurs when a user says "turn up the volume" for instance, without
// specifying alarm or another specific volume kind.) |new_volume|
// indicates the new default volume from 0.0 to 1.0.
    virtual void OnDefaultVolumeChangeEvent(float
        new_volume) = 0;

// Indicates the voice assistant client library is out of date with the
// server and needs to be updated. When this occurs, the client will
// no longer interact with the server.
    virtual void OnClientLibraryOutOfDateEvent() = 0;

```

};

[0092] [0070] 아래의 클래스 "DefaultEventDelegate"는 특정한 이벤트들에 대한 어느 것도 행하지 않는 오버라이드 (do-nothing override)들에 대한 기능들을 정의한다:

```

// Provides a default do-nothing implementation of an EventDelegate,
// useful for overriding only those functions that are interesting.
class DefaultEventDelegate : public EventDelegate {
    public:

```

[0094]

```

        void OnBootingUpEvent() override {}
        void OnHeardHotwordEvent() override {}
        void OnRecognizingSpeechStartedEvent() override {}
        void OnRecognizedSpeechChangedEvent(const
            RecognizedSpeechChangedEvent& event) override {}
        void OnRecognizingSpeechFinishedEvent(const
            RecognizedSpeechChangedEvent& event) override {}
        void OnRespondingStartedEvent() override {}
        void OnRespondingFinishedEvent() override {}
        void OnAlarmSoundingStartedEvent() override {}
        void OnAlarmSoundingFinishedEvent() override {}
        void OnTimerSoundingStartedEvent() override {}
        void OnTimerSoundingFinishedEvent() override {}
        void OnDefaultVolumeChangeEvent(float new_volume)
            override {}
        void OnClientLibraryOutOfDateEvent() override {}
    };

```

[0095]

[0071] 아래의 클래스 "Settings"은 제어기(402)에 제공될 수 있는 설정들(예컨대, 로케일(locale), 지오로케이션, 파일 시스템 디렉토리)을 정의한다:

```

// Assistant settings that are provided to the controller. They must be
// provided to the controller when starting the Assistant. They can also
// be updated and then provided to the Reconfigure function to take effect.
// The embedding app should not create its own classes derived from this.

```

```

class Settings {
public:
    virtual ~Settings() {}

    // Creates a default BuildInfo object.
    virtual std::unique_ptr<BuildInfo>
        CreateDefaultBuildInfo() = 0;

    // Sets the geolocation of the device. Optional.
    virtual void SetGeolocation(const Geolocation&
        geolocation) = 0;

    // Sets the build information of the device.
    virtual void SetBuildInfo(const BuildInfo&
        build_info) = 0;

    // Sets a file system directory the voice assistant client library can use.
    // This directory should be cleared whenever the voice assistant client
    // library should lose all previous context, such as when a factory data
    // reset occurs.

    virtual void SetAssistantDirectory(const
        std::string& path) = 0;

    // Sets the UserAgent to pass in connections to the server.
    virtual void SetUserAgent(const std::string&
        user_agent) = 0;

    // Sets the locale of the device.
    virtual void SetLocaleInfo(const LocaleInfo&
        locale_info) = 0;
};

```

[0098]

[0072] 아래의 클래스 "Controller"는 제어기(402)에 대응하고, Start(), Reconfigure(), RegisterAction(), 및 RegisterEventObserver() 기능들은 기능들 Start()(404), Reconfigure()(408), RegisterAction()(406), 및

RegisterEventObserver()(410)에 각각 대응한다.

```

// Controller class of the assistant.
class Controller {
public:
    virtual ~Controller() {}

// Creates a new default settings object that the application
// should configure and then pass to Start.
    virtual std::unique_ptr<Settings>
    CreateDefaultSettings() = 0;

// Starts the assistant and returns immediately. Returns true on success,
// false on failure. Will only succeed once per process. |settings| are the
// settings for the Assistant module. These are passed by const reference
// so it is clear the caller retains the Settings object and that any
// subsequent changes have no effect unless passed to Reconfigure. This
// function will fail if any required settings are not set.
    virtual bool Start(const Settings& settings) = 0;

// Reconfigures the running assistant and returns immediately. Returns
// false on failure including if the assistant is not yet started.
// |settings| are the new settings for the voice assistant module. This
// function will fail if any required settings are not set.
    virtual bool Reconfigure(const Settings& settings) = 0;

// Registers an action |module|. Fails if already registered.
    virtual bool RegisterAction(std::unique_ptr<ActionModule>
        module) = 0;

// Registers the EventDelegate to receive all assistant events.
    virtual void RegisterEventObserver(
        std::unique_ptr<EventDelegate> delegate) = 0;

// Call this function to create the controller class which controls the
// Assistant. |platform| must be set to a pointer to the platform API that
// the assistant will use. Returns nullptr on error.
    static ASSISTANT_EXPORT std::unique_ptr<Controller>
    Create(std::unique_ptr<PlatformApi> platform_api);
};

[0101]

```

[0073] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 디바이스(104) 또는 캐스팅 디바이스(106)는 플랫폼(예컨대, 동일한 플랫폼을 사용하여 다른 디바이스들과 통신하기 위한 인터페이스들의 세트, 및 인터페이스들의 세트를 지원하도록 구성된 운영 시스템)을 구현한다. 아래의 예시적인 코드는, 보이스 어시스턴트 클라이언트 라이브러리(402)가 플랫폼과 상호작용하기 위한 인터페이스와 연관된 기능들을 예시한다.

[0103] [0074] 아래의 클래스 "Authentication"는 특정한 계정들로 보이스 어시스턴트의 사용자를 인증하기 위한 인증토큰을 정의한다:

```

// Authentication provider for the platform.

class Authentication {
public:

// Returns the authentication scope of an authentication token.
    virtual std::string GetGoogleOAuth2Scopes() = 0;

// Returns the authentication token.
    virtual bool GetGoogleOAuth2Token(std::string* token) = 0;

protected:
    virtual ~Authentication() = default;
};

[0104]

```

[0075] 아래의 클래스 "OutputStreamType"는 오디오 출력 스트림들의 타입들을 정의한다:

```

// The possible types of audio output stream.

enum class OutputStreamType {
    kTts,
    kAlarm,
    kCalibration,
};

[0106]

```

[0076] 아래의 클래스 "SampleFormat"은 지원된 오디오 샘플 포맷들(예컨대, PCM 포맷들)을 정의한다:

```

// The supported PCM sample formats.

enum class SampleFormat {
    kInterleavedS16,           // Interleaved signed 16-bit integer.
    kInterleavedS32,           // Interleaved signed 32-bit integer.
    kInterleavedF32,           // Interleaved 32-bit float.
    kPlanarS16,                // Planar signed 16-bit integer.
    kPlanarS32,                // Planar signed 32-bit integer.
    kPlanarF32,                // Planar 32-bit float.
};

[0108]

```

[0077] 아래의 "BufferFormat"은 디바이스의 오디오 버퍼에 저장된 데이터의 포맷을 정의한다:

```

// Information about the format of data stored in an audio buffer.

struct BufferFormat {
    int sample_rate;
    SampleFormat sample_format;
    int num_channels;
};

[0109]

```

[0078] 클래스 "AudioBuffer"는 오디오 데이터에 대한 버퍼를 정의한다:

```

// Buffer class for input/output audio data.

class AudioBuffer {
public:

```

```

// Returns the format of the data in the buffer.
    virtual BufferFormat GetFormat() const = 0;

// Immutable data; used by the AudioInput delegate to read the incoming
// data.
    virtual const char* GetData() const = 0;

// Writable data; used by the AudioOutput delegate to write more
// data for output.
    virtual char* GetWritableData() const = 0;

// Returns the number of audio frames contained in
// GetData()/GetWritableData().
    virtual int GetFrames() const = 0;

protected:
    virtual ~AudioBuffer() {}

};

[0113]

```

[0114] [0079] 아래의 클래스 "AudioOutput"는 오디오 출력에 대한 인터페이스를 정의한다:

```

// Interface for audio output.
class AudioOutput {
public:
    enum Error {
        kFatalError,
        kUnderrun,
    };

    class Delegate {
public:

// Called when more output audio data is needed. The delegate
// implementation must fill data into the |buffer| as soon as possible,
// calling the |done_cb| once some data has been written. Note that
// the delegate may partially fill the buffer, but the number of
// |bytes_written| must be a multiple of the frame size. The delegate
// does not take ownership of the |buffer|.
// Note that this method must not block. If there is no data available
// to fill the buffer immediately, the buffer can be filled
// asynchronously by any thread, and then the |done_cb| must be called.
// The |done_cb| must not be called after the stream has been stopped
};

[0115]

```

```

// by a call to Stop(). If the end of stream has been reached, the
// delegate must call the |done_cb| with 0 |bytes_written|.
    virtual void FillBuffer(AudioBuffer* buffer,
                           const std::function<void(int
                           frames_written)>& done_cb) = 0;

// Called to indicate that the end of stream (ie, the point where the
// delegate passed 0 |bytes_written| to the |done_cb| of FillBuffer()) has
// been played out. Once this is called, it is safe to call Stop() without
// risk of discarding any unplayed data.
    virtual void OnEndOfStream() = 0;

// Called when an output error occurs.
    virtual void OnError(Error error) = 0;

// Called once the output has been stopped. Once this method has been
// called, there will be no more calls to any delegate methods unless the
// output is started again.
    virtual void OnStopped() = 0;

protected:
    ~Delegate() {}

};

virtual ~AudioOutput() {}

// Returns the stream type of this output, which was specified when the
// output was created.
    virtual OutputStreamType GetType() = 0;

// Starts audio output. This will begin requesting buffers in the given
// |format| by calling the |delegate|'s FillBuffer() method.
    virtual void Start(const BufferFormat& format,
                      Delegate* delegate) = 0;

// Stops audio output, putting this interface in a state when Start() can
// safely be called again with a new audio format and delegate. Any
// unplayed data provided by the delegate should be discarded when Stop()
// is called.
// Once the stop is complete and no further calls to the delegate will be
// made, the delegate's OnStopped() method will be called.
    virtual void Stop() = 0;

// Sets the volume range for this output stream. The volume of this stream
// should track the default volume as long as that volume is within the
// |min_volume| <= volume <= |max_volume| range (so, use the default
[0116]                                         volume, but clamped to the given range). The |min_volume| and
// |max_volume| are values 0.0 <= v <= 1.0, and represent a fraction of
// the total possible output volume of the system.
    virtual void SetVolume(float min_volume, float
                           max_volume) = 0;
};

[0117]

```

[0118]

[0080] 아래의 클래스 "AudioInput"은 오디오 입력을 캡처하기 위한 인터페이스를 정의한다:

```

// Interface for capturing audio input. When started, this should capture
// audio from all microphones, and provide data from each microphone as a
// separate channel in the buffers provided to the delegate's
// OnBufferAvailable() method.

class AudioInput {
public:
    enum Error {
        kFatalError,
        kOverrun,
    };

    class Delegate {
public:
    // Called when more input audio data is available. The |timestamp| is
    // the time in microseconds (relative to CLOCK_MONOTONIC_RAW epoch) that
    // the data in |buffer| was captured (for loopback audio, it is the
    // timestamp when the data was expected to be played).
    virtual void OnBufferAvailable(
        const AudioBuffer& buffer,
        int64_t timestamp) = 0;

    // Called when an error occurs on the AudioInput.
    virtual void OnError(Error error) = 0;

    // Called once the input has been stopped. Once this method has been
    // called, there will be no more calls to any delegate methods unless the
    // input is started again.
    virtual void OnStopped() = 0;
};

    virtual ~AudioInput() {}

    // Starts capturing audio input and passing it to the |delegate|'s
    // OnBufferAvailable() method.
    virtual void Start(Delegate* delegate) = 0;

    // Stops capturing audio input. Once the input is stopped and no more calls
    // to any delegate methods will be made, the delegate's OnStopped() method
    // will be called.
    virtual void Stop() = 0;
};

```

[0119]

```

[0120]

```

[0121] [0081] 아래의 클래스 "Resources"는 시스템 리소스들에 대한 액세스를 정의한다:

```
// Access to system resource files.

class Resources {
public:
    using ResourceLoadingCallback = std::function<void(
        const std::string& output)>;
    Resources() {}
    virtual ~Resources() {}
    virtual bool GetBuiltInHotwordData(
        const LocaleInfo* locale,
        const ResourceLoadingCallback& callback) = 0;
    virtual bool GetAlarmMp3(const ResourceLoadingCallback&
        callback) = 0;
    virtual bool GetTimerMp3(const ResourceLoadingCallback&
        callback) = 0;
    virtual bool GetCalibrationMp3(const
        ResourceLoadingCallback& callback) = 0;
    virtual bool GetVolumeChangeMp3(const
        ResourceLoadingCallback& callback) = 0;
    virtual bool GetSpeechRecognitionErrorMp3(
        const LocaleInfo* locale,
        const ResourceLoadingCallback& callback) = 0;
    virtual bool GetSpeechRecognitionStoppedMp3(
        const LocaleInfo* locale,
        const ResourceLoadingCallback& callback) = 0;
    virtual bool GetNoInternetMp3(const LocaleInfo* locale,
        const ResourceLoadingCallback& callback) = 0;
};

[0122]
```

[0123] [0082] 아래의 클래스 "PlatformApi"는 보이스 어시스턴트 클라이언트 라이브러리(240)에 대한 플랫폼 API(예컨대, 플랫폼 API(244))를 특정한다:

```
// Platform API to be used the voice assistant.

class PlatformApi {
public:
    virtual ~PlatformApi() {}

    // Returns an audio output interface for the desired |type| of stream.
    // This is owned by the PlatformApi.
    virtual std::unique_ptr<AudioOutput> GetAudioOutput(
        OutputStreamType type) = 0;

    // Returns an interface for capturing audio input.
    virtual std::unique_ptr<AudioInput> GetAudioInput() = 0;

    // Returns an interface for capturing loopback audio. This is an
    // "audio input" where the captured data is audio data that was about to
    // be played out.

    // The loopback audio may be captured after all mixing and
    // postprocessing has been done, as soon as possible before sending to
    // the output hardware.
    virtual std::unique_ptr<AudioInput>
    GetLoopbackInput() = 0;

    virtual Authentication& GetAuthentication() = 0;
};

[0124]
```

[0125] [0083] 몇몇 구현들에서, 볼륨 제어는 보이스 어시스턴트 클라이언트 라이브러리(240) 외부에서 핸들링될 수 있다. 예컨대, 시스템 볼륨은 보이스 어시스턴트 클라이언트 라이브러리(240)의 제어부의 외부의 디바이스에 의해 유지될 수 있다. 다른 예로서, 보이스 어시스턴트 클라이언트 라이브러리(240)는 볼륨 제어를 여전히 지원

할 수 있지만, 보이스 어시스턴트 클라이언트 라이브러리(240)에 대한 볼륨 제어를 위한 요청들은 디바이스로 안내된다.

- [0127] [0084] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240) 내의 알람 및 타이머 기능은 사용자에 의해 디스에이블링되거나 또는 디바이스에서 라이브러리를 구현할 경우 디스에이블링될 수 있다.
- [0128] [0085] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240)는 또한, 디바이스 LED들 상의 LED 애니메이션들의 디스플레이를 용이하게 하기 위해 디바이스 상에서 LED들에 대한 인터페이스를 지원한다.
- [0129] [0086] 몇몇 구현들에서, 보이스 어시스턴트 클라이언트 라이브러리(240)는 캐스팅 디바이스(106)의 캐스팅 수신기 모듈(예컨대, 수신기 모듈(146))에 포함되거나 또는 그에 연결될 수 있다. 보이스 어시스턴트 클라이언트 라이브러리(240)와 수신기 모듈(146) 사이의 연결은, 예컨대, 부가적인 액션들(예컨대, 로컬 미디어 재생)에 대한 지원 및 캐스팅 디바이스(106) 상의 LED들의 제어를 위한 지원을 포함할 수 있다.
- [0130] [0087] 도 5는 몇몇 구현들에 따른, 디바이스 상에서 구두 입력들을 프로세싱하기 위한 방법(500)의 흐름도를 예시한다. 방법(500)은, 오디오 입력 시스템(예컨대, 오디오 입력 디바이스(108/132)), 하나 이상의 프로세서들(예컨대, 프로세싱 유닛(들)(202)), 및 하나 이상의 프로세서들에 의한 실행을 위한 하나 이상의 프로그램들을 저장한 메모리(예컨대, 메모리(206))를 갖는 전자 디바이스(예컨대, 보이스 어시스턴트 클라이언트 디바이스(104), 캐스팅 디바이스(106))에서 수행된다. 몇몇 구현들에서, 전자 디바이스는, 오디오 입력 시스템(예컨대, 오디오 입력 디바이스(108/132)), 하나 이상의 프로세서들(예컨대, 프로세싱 유닛(들)(202)), 및 하나 이상의 프로세서들에 의해 실행될 하나 이상의 프로그램들을 저장한 메모리(예컨대, 메모리(206))를 포함하며, 하나 이상의 프로그램들은 방법(500)을 수행하기 위한 명령들을 포함한다. 몇몇 구현들에서, 비-일시적인 컴퓨터 판독 가능 저장 매체는 하나 이상의 프로그램들을 저장하며, 하나 이상의 프로그램들은, 오디오 입력 시스템(예컨대, 오디오 입력 디바이스(108/132)) 및 하나 이상의 프로세서들(예컨대, 프로세싱 유닛(들)(202))을 갖는 전자 디바이스에 의해 실행될 경우, 전자 디바이스로 하여금 방법(500)을 수행하게 하는 명령들을 포함한다. 방법(500)을 수행하기 위한 프로그램들 또는 명령들은 도 2 내지 도 4를 참조하여 위에서 설명된 모듈들, 라이브러리들 등에 포함될 수 있다.
- [0131] [0088] 디바이스는 디바이스에서 구두 입력을 수신한다(502). 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 사용자에 의해 말해진 구두 입력(예컨대, 보이스 입력)을 캡처한다.
- [0132] [0089] 디바이스는 구두 입력을 프로세싱한다(504). 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 구두 입력을 프로세싱한다. 프로세싱은, 핫워드 검출, 텍스트 데이터로의 변환, 및 사용자에 의해 제공된 커맨드들, 요청들, 및/또는 파라미터들에 대응하는 워드들 및 어구들의 식별을 포함할 수 있다. 몇몇 구현들에서, 프로세싱은 최소일 수 있거나, 또는 어떠한 프로세싱도 전혀 존재하지 않을 수 있다. 예컨대, 프로세싱은 서버(114)로의 송신을 위해 구두 입력 오디오를 인코딩하는 것, 또는 서버(114)로의 송신을 위해 구두 입력의 캡처된 로우(raw) 오디오를 준비하는 것을 포함할 수 있다.
- [0133] [0090] 디바이스는 요청을 원격 시스템에 송신하며, 요청은 구두 입력에 기반하여 결정된 정보를 포함한다(506). 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 구두 입력으로부터 요청 및 하나 이상의 연관된 파라미터들을 식별하도록 구두 입력을 프로세싱함으로써 구두 입력으로부터 요청을 결정한다. 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 결정된 요청을 원격 시스템(예컨대, 서버(114))에 송신하며, 여기서, 원격 시스템은 요청에 대한 응답을 결정 및 생성한다. 몇몇 구현들에서, 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 구두 입력을 (예컨대, 인코딩된 오디오로서, 로우 오디오 데이터로서) 서버(114)에 송신하며, 서버(114)는 요청 및 연관된 파라미터들을 결정하기 위해 구두 입력을 프로세싱한다.
- [0134] [0091] 디바이스는 요청에 대한 응답을 수신하며, 여기서, 응답은 구두 입력에 기반한 정보에 따라 원격 시스템에 의해 생성된다(508). 원격 시스템(예컨대, 서버(114))은 요청에 대한 응답을 결정 및 생성하고, 응답을 클라이언트 디바이스(104)/캐스팅 디바이스(106)에 송신한다.
- [0135] [0092] 디바이스는 응답에 따라 동작을 수행한다(510). 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 수신된 응답에 따라 하나 이상의 동작들을 수행한다. 예컨대, 응답이 오디오에 의해 특정한 정보를 출력하기 위한 디바이스에 대한 커맨드이면, 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 정보를 리트리브하고, 정보를 스피치 오디오 출력으로 변환하며, 스피커를 통해 스피치 오디오를 출력한다. 다른 예로서, 응답이 미디어 콘텐츠를 재생하기 위한 디바이스에 대한 커맨드이면, 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 미디어 콘텐츠를 리트리브하고, 미디어 콘텐츠를 재생한다.

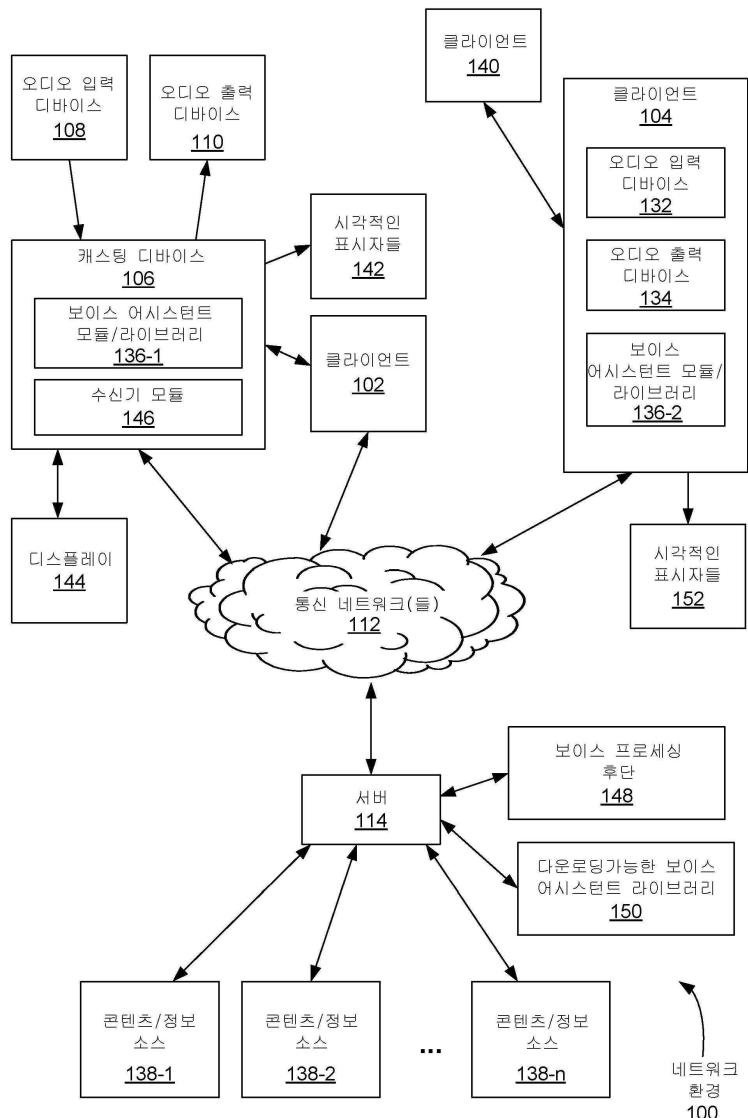
- [0136] [0093] 수신, 프로세싱, 송신, 수신 및 수행 중 하나 이상은 전자 디바이스 상에서 실행되는 보이스 어시스턴트 라이브러리의 하나 이상의 보이스 프로세싱 모듈들에 의해 수행되며, 보이스 프로세싱 모듈들은 하나 이상의 애플리케이션 프로그램들에 액세스가능한 복수의 보이스 프로세싱 동작들을 제공하고 그리고/또는 전자 디바이스 상에서 실행되거나 또는 실행가능한 소프트웨어를 동작시킨다(512). 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 수신, 프로세싱, 송신, 수신, 및 수행 단계들 중 하나 이상을 수행하기 위한 기능들 및 모듈들을 포함하는 보이스 어시스턴트 클라이언트 라이브러리(240)를 가질 수 있다. 보이스 어시스턴트 클라이언트 라이브러리(240)의 모듈들은, 라이브러리(240)를 포함하거나 그에 연결되는 (예컨대, 라이브러리(240) 및 관련된 API들을 구동시키는) 클라이언트 디바이스(104)/캐스팅 디바이스(106)의 애플리케이션들, 운영 시스템들, 플랫폼 소프트웨어에 액세스가능한 다수의 보이스 프로세싱 및 어시스턴트 동작들을 제공한다.
- [0137] [0094] 몇몇 구현들에서, 보이스 프로세싱 모듈들과 연관된 적어도 몇몇 보이스 프로세싱 동작들은, 광역 네트워크를 통해 전자 디바이스와 상호연결된 원격 시스템 상에서 수행된다. 예컨대, 요청을 결정하기 위한 구두 입력의 프로세싱은, 네트워크(들)(112)를 통해 클라이언트 디바이스(104)/캐스팅 디바이스(106)와 연결된 서버(114)에 의해 수행될 수 있다.
- [0138] [0095] 몇몇 구현들에서, 보이스 어시스턴트 라이브러리는, 복수의 상이한 디바이스 타입들 상에서 동작가능한 공통 운영 시스템 상에서 실행가능하며, 그에 의해, 보이스 프로세싱 동작들 중 하나 이상과 상호작용하도록 구성된 보이스-인에이블 애플리케이션들의 이식성을 가능하게 한다. 보이스 어시스턴트 클라이언트 라이브러리(240)(및 관련된 라이브러리들 및 API들, 예컨대, 디버깅 라이브러리(242), 플랫폼 API(244), 포직스 API(246))는 미리 정의된 운영 시스템(예컨대, 리눅스)의 표준 엘리먼트들(예컨대, 오브젝트들)을 사용하며, 따라서, 미리 정의된 운영 시스템의 배포 또는 특색(예컨대, 상이한 리눅스 또는 리눅스-기반 배포들 또는 특색들)을 구동시키는 다양한 디바이스들 상에서 동작가능하다. 이러한 방식으로, 보이스 어시스턴트 기능은 다양한 디바이스들에 이용가능하며, 보이스 어시스턴트 경험은 다양한 디바이스들에 걸쳐 일관된다.
- [0139] [0096] 몇몇 구현들에서, 요청 및 응답은 디바이스에서 핸들링될 수 있다. 예컨대, 디바이스에 로컬일 수 있는 기본적인 기능들, 이를테면 타이머들, 알람 시계들, 시계들, 및 볼륨 제어에 대해, 클라이언트 디바이스(104)/캐스팅 디바이스(106)는 구두 입력을 프로세싱하고, 요청이 이들 기본적인 기능들 중 하나에 대응한다고 결정하고, 디바이스에서 응답을 결정하며, 응답에 따라 하나 이상의 동작들을 수행할 수 있다. 디바이스는 로깅 목적들을 위해 요청 및 응답을 여전히 서버(114)에 리포팅할 수 있다.
- [0140] [0097] 몇몇 구현들에서, 오디오 입력 시스템을 포함하는 전자 디바이스들에 대한 디바이스-독립적인 보이스 어시스턴트 라이브러리는, 복수의 상이한 전자 디바이스 타입들 상에서 구현되는 공통 동작 시스템 상에서 실행되도록 구성되는 하나 이상의 보이스 프로세싱 모듈들을 포함하며, 보이스 프로세싱 모듈들은 애플리케이션 프로그램들에 액세스가능한 복수의 보이스 프로세싱 동작들을 제공하고 전자 디바이스들 상에서 실행되는 소프트웨어를 동작시켜, 그에 의해, 보이스 프로세싱 동작들 중 하나 이상과 상호작용하도록 구성된 보이스-인에이블 애플리케이션들의 이식성을 가능하게 한다. 보이스 어시스턴트 클라이언트 라이브러리(240)는, 라이브러리와 동일한 미리 정의된 운영 시스템을 공유하는 다양한 디바이스들 상에서 구동될 수 있는 라이브러리이며(예컨대, 라이브러리 및 디바이스 운영 시스템은 리눅스-기반임), 따라서 라이브러리는 디바이스-독립적이다. 라이브러리(240)는 다양한 디바이스들에 걸쳐 애플리케이션들에 액세스가능한 보이스 어시스턴트 기능에 대해 다수의 모듈들을 제공한다.
- [0141] [0098] 몇몇 구현들에서, 보이스 프로세싱 모듈들과 연관된 적어도 몇몇 보이스 프로세싱 동작들은, 광역 네트워크를 통해 전자 디바이스들과 상호연결된 후단 서버 상에서 수행된다. 예컨대, 라이브러리(240)는 요청을 결정하기 위한 프로세싱을 위해 구두 입력을 서버(114)에 송신하도록 서버(114)와 통신하는 모듈들을 포함한다.
- [0142] [0099] 몇몇 구현들에서, 보이스 프로세싱 동작들은 전자 디바이스들과 (예컨대, 직접적으로 또는 통신가능하게) 커플링된 디바이스들을 제어하도록 구성되는 디바이스-특정 동작들을 포함한다. 라이브러리(240)는 클라이언트 디바이스(104)/캐스팅 디바이스(106)에 커플링된 다른 디바이스들(예컨대, 무선 스피커들, 스마트 텔레비전 등)을 제어하기 위한 기능들 또는 모듈들을 포함할 수 있다.
- [0143] [00100] 몇몇 구현들에서, 보이스 프로세싱 동작들은, 전자 디바이스들의 사용자에게 또는 전자 디바이스들과 (예컨대, 직접적으로 또는 통신가능하게) 커플링된 디바이스들 상에서 요청된 정보 및/또는 미디어 콘텐츠를 제공하도록 구성되는 정보 및 미디어 요청 동작들을 포함한다. 라이브러리(240)는 정보 또는 미디어를 리트리브하고, 클라이언트 디바이스(104)/캐스팅 디바이스(106) 또는 커플링된 디바이스 상에서 정보 또는 미디어를 제공(예컨대, 이메일을 큰소리로 읽거나, 뉴스 기사들을 큰 소리로 읽거나, 스트리밍 음악을 재생함)하기 위한 기

능들 또는 모듈들을 포함할 수 있다.

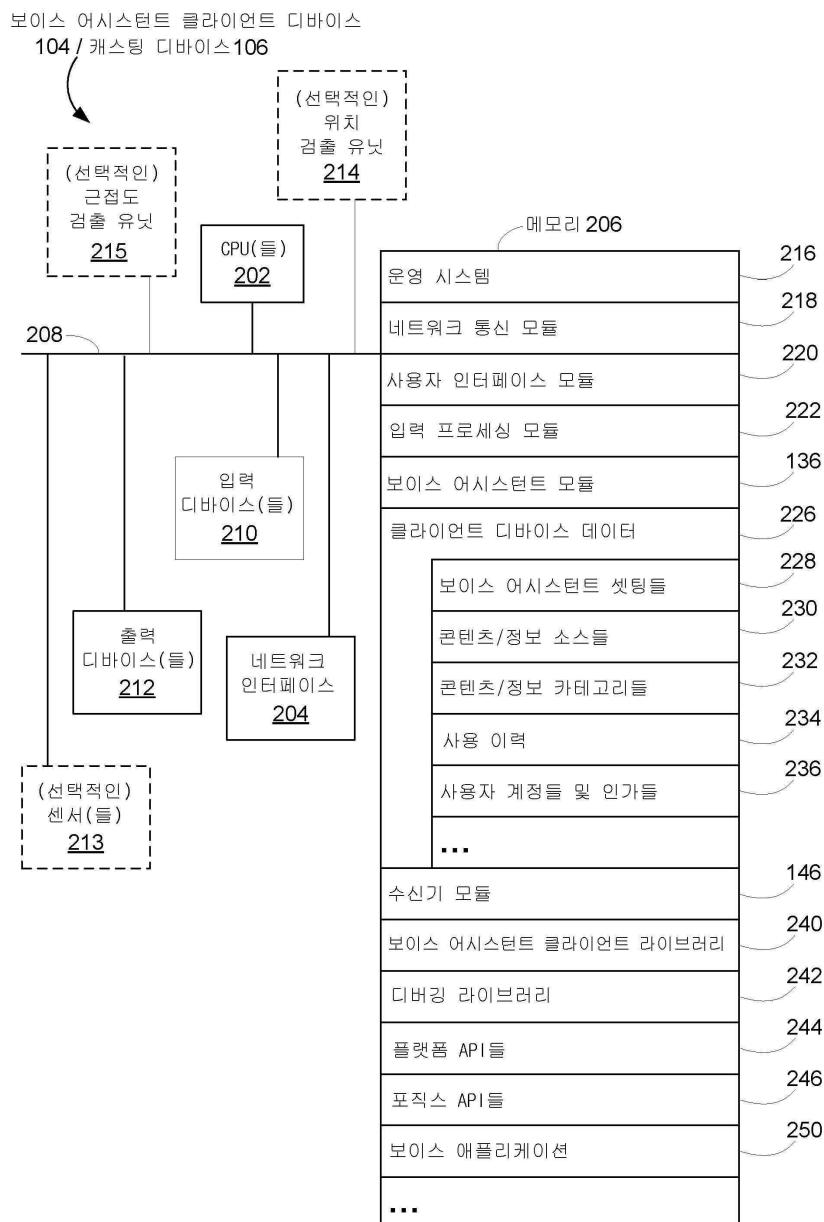
- [0144] [00101] 용어들 "제1", "제2" 등이 다양한 엘리먼트들을 설명하기 위해 본 명세서에서 사용될 수 있지만, 이들 엘리먼트들이 이들 용어들에 의해 제한되지 않아야 함을 이해할 것이다. 이들 용어들은 하나의 엘리먼트를 다른 엘리먼트와 구별하기 위해서만 사용된다. 예컨대, 제1 접촉은 제2 접촉으로 지칭될 수 있고, 유사하게 제2 접촉은 제1 접촉으로 지칭될 수 있으며, 이는, "제1 접촉"의 모든 발생들이 일관되게 이름변경되고 제2 접촉의 모든 발생들이 일관되게 이름변경되는 한 설명의 의미를 변경시킨다. 제1 접촉 및 제2 접촉 둘 모두는 접촉이지만, 그들은 동일한 접촉은 아니다.
- [0145] [00102] 본 명세서에서 사용된 용어는 특정한 구현들만을 설명하려는 목적을 위한 것이며, 청구범위를 제한하도록 의도되지 않는다. 구현들의 설명 및 첨부된 청구범위에서 사용된 바와 같이, 단수형들은, 문맥상 명확하게 달리 표시되지 않으면, 복수형들을 또한 포함하도록 의도된다. 본 명세서에서 사용된 바와 같이, 용어 "및/또는"은 연관된 항목들 중 하나 이상의 항목들의 임의의 그리고 모든 가능한 조합들을 지칭하고 포함한다는 것이 또한 이해될 것이다. 본 명세서에서 사용되는 경우 용어들 "구비" 및/또는 "구비하는"이 언급된 특성들, 정수들, 단계들, 동작들, 엘리먼트들, 및/또는 컴포넌트들의 존재를 특정하지만, 하나 이상의 다른 특성들, 정수들, 단계들, 동작들, 엘리먼트들, 컴포넌트들, 및/또는 그들의 그룹들의 존재 또는 부가를 배제하지는 않는다는 것이 추가로 이해될 것이다.
- [0146] [00103] 본 명세서에서 사용된 바와 같이, 용어 "~라면"은 문맥에 의존하여, 언급된 조건 선례가 참인 "~경우" 또는 "~시에" 또는 "결정하는 것에 대한 응답으로" 또는 "결정에 따라" 또는 "검출하는 것에 대한 응답으로"를 의미하도록 해석될 수 있다. 유사하게, 어구 "[언급된 조건 선례가 참]이라고 결정되면" 또는 "[언급된 조건 선례가 참이면]" 또는 "[언급된 조건 선례가 참]인 경우"는 문맥에 의존하여, 언급된 조건 선례가 참인 것으로 "결정할 시에" 또는 "결정하는 것에 대한 응답으로" 또는 "결정에 따라" 또는 "검출할 시에" 또는 "검출하는 것에 대한 응답으로"를 의미하도록 해석될 수 있다.
- [0147] [00104] 이제, 다양한 구현들에 대한 참조가 상세히 이루어질 것이며, 다양한 구현들의 예들은 첨부한 도면들에 예시되어 있다. 다음의 상세한 설명에서, 다수의 특정한 세부사항들이 본 발명 및 설명된 구현들의 완전한 이해를 제공하기 위해 기재된다. 그러나, 본 발명은 이러한 특정한 세부사항들 없이 실시될 수 있다. 다른 예시들에서, 잘-알려진 방법들, 절차들, 컴포넌트들, 및 회로들은 구현들의 양상들을 불필요하게 모호하게 하지 않기 위해 상세히 설명되지 않는다.
- [0148] [00105] 전술한 설명은 설명의 목적을 위해 특정한 구현들을 참조하여 설명되었다. 그러나, 위의 예시적인 논의들은, 포괄적이거나 본 발명을 개시된 정확한 형태로 제한하도록 의도되지 않는다. 많은 변형들 및 변경들이 위의 교시들의 관점에서 가능하다. 본 발명의 원리들 및 본 발명의 실제 애플리케이션들을 최상으로 설명하고, 그에 의해, 당업자들이 고려된 특정 사용에 적합하게 본 발명 및 다양한 구현들을 다양한 변형들과 함께 최상으로 이용할 수 있게 하기 위해 구현들이 선정되고 설명되었다.

도면

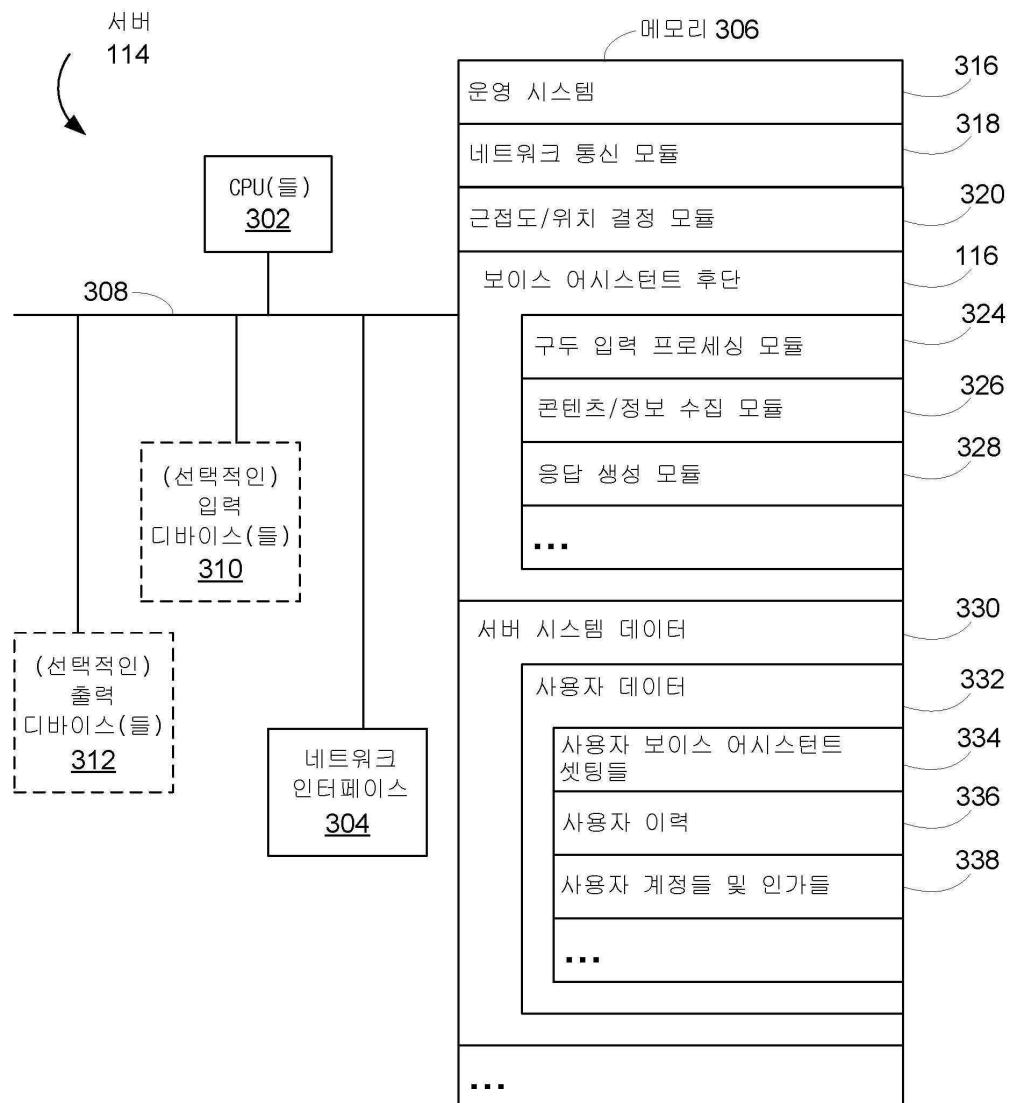
도면1



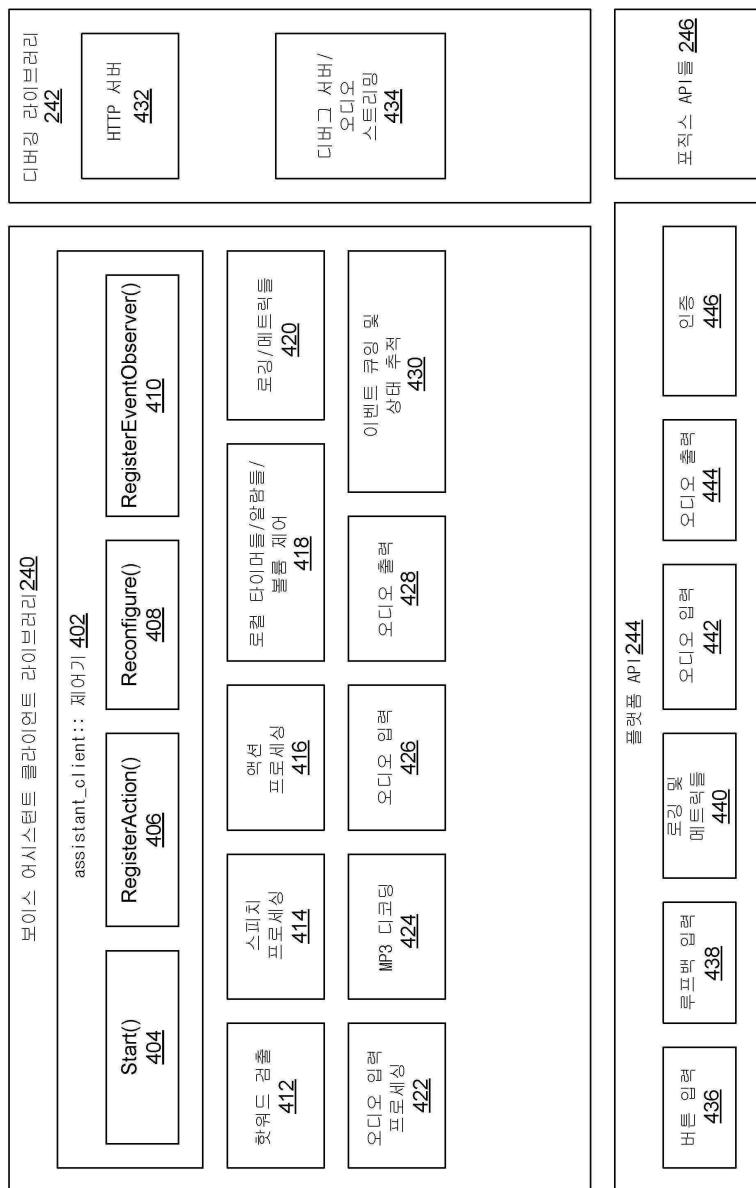
도면2



도면3



도면4



도면5

500

오디오 입력 시스템, 하나 또는 그 초과의 프로세서들, 및 하나 또는 그 초과의 프로세서들에 의한 실행을 위한 하나 또는 그 초과의 프로그램들을 저장한 메모리를 포함하는 전자 디바이스에서:

