



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년10월15일
(11) 등록번호 10-2718167
(24) 등록일자 2024년10월11일

- (51) 국제특허분류(Int. Cl.)
G06F 8/61 (2018.01) G06F 8/65 (2018.01)
G06F 8/71 (2018.01) G06F 9/445 (2018.01)
G06F 9/448 (2018.01) H04L 41/00 (2022.01)
H04W 8/24 (2009.01)
- (52) CPC특허분류
G06F 8/61 (2013.01)
G06F 8/65 (2013.01)
- (21) 출원번호 10-2021-7002717
- (22) 출원일자(국제) 2019년06월27일
심사청구일자 2022년05월26일
- (85) 번역문제출일자 2021년01월27일
- (65) 공개번호 10-2021-0027400
- (43) 공개일자 2021년03월10일
- (86) 국제출원번호 PCT/US2019/039507
- (87) 국제공개번호 WO 2020/027954
국제공개일자 2020년02월06일
- (30) 우선권주장
16/051,145 2018년07월31일 미국(US)
- (56) 선행기술조사문헌
JP2017518594 A
US20090049518 A1

- (73) 특허권자
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
- (72) 발명자
카우프만 피터 제이
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 테크놀로지
라이선싱, 엘엘씨
준 리차드 아이
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 테크놀로지
라이선싱, 엘엘씨
(뒷면에 계속)
- (74) 대리인
제일특허법인(유)

전체 청구항 수 : 총 20 항

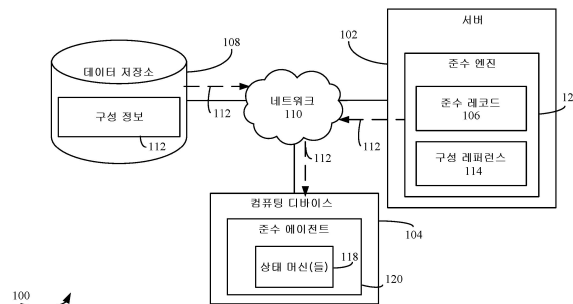
심사관 : 채정목

(54) 발명의 명칭 구성 시나리오 준수를 위한 모바일 디바이스에 의한 준수 설정 구현

(57) 요약

본원에 기술된 실시예는 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하는 것에 관한 것이다. 예를 들어, 컴퓨팅 디바이스는, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 구현되는 준수 설정을 기술하는 구성 정보를 서버로부터 수신할 수 있다. 더욱이, 컴퓨팅 디바이스는 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 구성 정보에 의해 표시된 상태 머신을 식별하고, 상태 머신을 실행하여, 준수 설정으로 컴퓨팅 디바이스를 구성할 수 있다.

대표도



(52) CPC특허분류

G06F 8/71 (2013.01)
G06F 9/44505 (2013.01)
G06F 9/4498 (2018.02)
HO4L 41/0803 (2022.05)
HO4W 8/245 (2013.01)

(72) 발명자

라시왈라 모이즈 지

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

두두그지안 다니엘 에스

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

라히리 샤약

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

볼즈 손 에이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

주 유향

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

치마 아딕 싱

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

사뮤드랄라 라마 크리쉬나

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

코슈티 지텐드라쿠마르 지

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

다완 고우라브 라구

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

명세서

청구범위

청구항 1

컴퓨팅 디바이스로서,

적어도 하나의 프로세서 회로와,

상기 적어도 하나의 프로세서 회로에 의해 실행되도록 구성된 프로그램 코드를 저장하는 적어도 하나의 메모리를 포함하고,

상기 프로그램 코드는,

사용자가 상기 컴퓨팅 디바이스에 처음 로그인했다는 표시를 수신하고,

서버로부터, 상기 서버에서 구성 레퍼런스 내에 유지되는 사용자-디바이스 쌍의 지정을 포함하는 구성 정보를 수신 - 상기 사용자-디바이스 쌍은 상기 사용자 및 상기 컴퓨팅 디바이스에 대응하며, 상기 구성 정보는 상기 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 조정하기 위한 준수 설정을 더 포함함 - 하며,

상기 수신된 구성 정보의 에러를 결정하고,

상기 서버로 상기 에러의 정정을 위한 요청을 전송하며,

상기 서버로부터 업데이트된 구성 정보를 수신하고,

상기 업데이트된 구성 정보로 상기 컴퓨팅 디바이스를 구성하며,

상기 업데이트된 구성 정보의 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 업데이트된 구성 정보에 의해 표시되는 상태 머신을 식별하고,

상기 상태 머신의 실행에 의해 상기 컴퓨팅 디바이스가 상기 업데이트된 구성 정보에 따라 구성되는지 여부를 결정하는

것을 포함하는 동작들을 수행하도록 구성되는,

컴퓨팅 디바이스.

청구항 2

제1항에 있어서,

상기 프로그램 코드는,

상기 상태 머신의 실행에 의해 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하고,

상기 구성 정보 데이터 저장소로부터 상기 구성 결과를 획득하며,

상기 구성 결과를 상기 서버에 송신하는

것을 포함하는 동작들을 수행하도록 더 구성되는,

컴퓨팅 디바이스.

청구항 3

제1항에 있어서,

상기 업데이트된 구성 정보는 제1 준수 설정 및 제2 준수 설정 및 업데이트된 상기 구성 시나리오의 준수 설정

을 구현하도록 구성된 구성 프로세스의 상태 머신의 표시를 포함하고,

상기 프로그램 코드는,

상기 업데이트된 구성 정보에 의해 표시된 상기 상태 머신을 식별 - 상기 상태 머신은 상기 제1 및 제2 준수 설정을 포함하는 준수 설정과 페이로드가 구현되는 순서를 지시하도록 구성됨 - 하고,

상기 상태 머신의 실행에 의해, 상기 순서에 따라 상기 업데이트된 구성 정보의 준수 설정으로 상기 컴퓨팅 디바이스를 구성하는

것을 포함하는 동작들을 수행하도록 더 구성되는,

컴퓨팅 디바이스.

청구항 4

제1항에 있어서,

상기 구성 정보는 상기 준수 설정과 연관된 페이로드 및 상기 페이로드의 다운로드 및 처리를 위한 대응하는 URI(Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 상기 서버로부터 수신되는,

컴퓨팅 디바이스.

청구항 5

컴퓨팅 디바이스에서 구현되는 방법으로서,

사용자가 상기 컴퓨팅 디바이스에 처음 로그인했다는 표시를 수신하는 단계와,

서버로부터, 상기 서버에서 구성 레퍼런스 내에 유지되는 사용자-디바이스 쌍의 지정을 포함하는 구성 정보를 수신하는 단계 - 상기 사용자-디바이스 쌍은 상기 사용자 및 상기 컴퓨팅 디바이스에 대응하며, 상기 구성 정보는 상기 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 조정하기 위한 준수 설정을 더 포함함 - 와,

상기 수신된 구성 정보의 에러를 결정하는 단계와,

상기 서버로 상기 에러의 정정을 위한 요청을 전송하는 단계와,

상기 서버로부터 업데이트된 구성 정보를 수신하는 단계와,

상기 업데이트된 구성 정보로 상기 컴퓨팅 디바이스를 구성하는 단계와,

상기 업데이트된 구성 정보의 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 업데이트된 구성 정보에 의해 표시되는 상태 머신을 식별하는 단계와,

상기 상태 머신의 실행에 의해 상기 컴퓨팅 디바이스가 상기 업데이트된 구성 정보에 따라 구성되는지 여부를 결정하는 단계를 포함하는,

방법.

청구항 6

제5항에 있어서,

상기 상태 머신의 실행에 의해 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하는 단계와,

상기 구성 정보 데이터 저장소로부터 상기 구성 결과를 획득하는 단계와,

상기 구성 결과를 상기 서버에 송신하는 단계를 더 포함하는,

방법.

청구항 7

제5항에 있어서,

상기 업데이트된 구성 정보는 제1 준수 설정 및 제2 준수 설정 및 업데이트된 상기 구성 시나리오의 준수 설정을 구현하도록 구성된 구성 프로세스의 상태 머신의 표시를 포함하고,

상기 방법은,

상기 업데이트된 구성 정보에 의해 표시된 상기 상태 머신을 식별하는 단계 - 상기 상태 머신은 상기 제1 및 제2 준수 설정을 포함하는 준수 설정과 페이로드가 구현되는 순서를 지시하도록 구성됨 - 와,

상기 상태 머신의 실행에 의해, 상기 순서에 따라 상기 업데이트된 구성 정보의 준수 설정으로 상기 컴퓨팅 디바이스를 구성하는 단계를 더 포함하는,

방법.

청구항 8

제5항에 있어서,

상기 구성 정보는 상기 준수 설정과 연관된 페이로드 및 상기 페이로드의 다운로드 및 처리를 위한 대응하는 URI(Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 상기 서버로부터 수신되는,

방법.

청구항 9

컴퓨팅 디바이스의 적어도 하나의 처리 회로에 의해 실행될 때 컴퓨팅 디바이스에서 방법을 수행하는 프로그램 명령어가 기록된 컴퓨터 판독가능 저장 매체로서, 상기 방법은,

사용자가 상기 컴퓨팅 디바이스에 처음 로그인했다는 표시를 수신하는 단계와,

서버로부터, 상기 서버에서 구성 레퍼런스 내에 유지되는 사용자-디바이스 쌍의 지정을 포함하는 구성 정보를 수신하는 단계 - 상기 사용자-디바이스 쌍은 상기 사용자 및 상기 컴퓨팅 디바이스에 대응하며, 상기 구성 정보는 상기 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 조정하기 위한 준수 설정을 더 포함함 - 와,

상기 수신된 구성 정보의 에러를 결정하는 단계와,

상기 서버로 상기 에러의 정정을 위한 요청을 전송하는 단계와,

상기 서버로부터 업데이트된 구성 정보를 수신하는 단계와,

상기 업데이트된 구성 정보로 상기 컴퓨팅 디바이스를 구성하는 단계와,

상기 업데이트된 구성 정보의 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 업데이트된 구성 정보에 의해 표시되는 상태 머신을 식별하는 단계와,

상기 상태 머신의 실행에 의해 상기 컴퓨팅 디바이스가 상기 업데이트된 구성 정보에 따라 구성되는지 여부를 결정하는 단계를 포함하는,

컴퓨터 판독가능 저장 매체.

청구항 10

제9항에 있어서,
 상기 방법은,
 상기 상태 머신의 실행에 의해 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하는 단계와,
 상기 구성 정보 데이터 저장소로부터 상기 구성 결과를 획득하는 단계와,
 상기 구성 결과를 상기 서버에 송신하는 단계를 더 포함하는,
 컴퓨터 판독가능 저장 매체.

청구항 11

제9항에 있어서,
 상기 업데이트된 구성 정보는 제1 준수 설정 및 제2 준수 설정 및 업데이트된 상기 구성 시나리오의 준수 설정을 구현하도록 구성된 구성 프로세스의 상태 머신의 표시를 포함하고,
 상기 방법은,
 상기 업데이트된 구성 정보에 의해 표시된 상기 상태 머신을 식별하는 단계 - 상기 상태 머신은 상기 제1 및 제2 준수 설정을 포함하는 준수 설정과 페이로드가 구현되는 순서를 지시하도록 구성됨 - 와,
 상기 상태 머신의 실행에 의해, 상기 순서에 따라 상기 업데이트된 구성 정보의 준수 설정으로 상기 컴퓨팅 디바이스를 구성하는 단계를 더 포함하는,
 컴퓨터 판독가능 저장 매체.

청구항 12

제9항에 있어서,
 상기 구성 정보는 상기 준수 설정과 연관된 페이로드 및 상기 페이로드의 다운로드 및 처리를 위한 대응하는 URI(Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 상기 서버로부터 수신되는,
 컴퓨터 판독가능 저장 매체.

청구항 13

제1항에 있어서,
 상기 프로그램 코드는, 트리거링 이벤트의 검출 시, 상기 컴퓨팅 디바이스가 상기 구성 시나리오를 계속 준수하는지 여부를 결정하는 것을 포함하는 동작들을 수행하도록 더 구성되고,
 상기 트리거링 이벤트는,
 미리 결정된 시간 기간의 만료, 또는
 상기 컴퓨팅 디바이스가 준수하는지 여부를 결정하는 관리자, 감사자 및/또는 엔티티로부터의 요청 중 하나를 포함하는,
 컴퓨팅 디바이스.

청구항 14

제1항에 있어서,

상기 사용자와 상기 컴퓨팅 디바이스의 쌍은 상기 서버의 테이블에 저장되고, 상기 테이블의 각 엔트리는 사용자-디바이스 쌍 및 상기 사용자-디바이스 쌍에 의해 구현된 대응하는 구성 설정을 지정하는, 컴퓨팅 디바이스.

청구항 15

제5항에 있어서,
트리거링 이벤트의 검출 시, 상기 컴퓨팅 디바이스가 상기 구성 시나리오를 계속 준수하는지 여부를 결정하는 단계를 더 포함하고,
상기 트리거링 이벤트는,
미리 결정된 시간 기간의 만료, 또는
상기 컴퓨팅 디바이스가 준수하는지 여부를 결정하는 관리자, 감사자 및/또는 엔티티로부터의 요청 중 하나를 포함하는,
방법.

청구항 16

제5항에 있어서,
상기 사용자와 상기 컴퓨팅 디바이스의 쌍은 상기 서버의 테이블에 저장되고, 상기 테이블의 각 엔트리는 사용자-디바이스 쌍 및 상기 사용자-디바이스 쌍에 의해 구현된 대응하는 구성 설정을 지정하는,
방법.

청구항 17

제9항에 있어서,
상기 방법은,
트리거링 이벤트의 검출 시, 상기 컴퓨팅 디바이스가 상기 구성 시나리오를 계속 준수하는지 여부를 결정하는 단계를 더 포함하고,
상기 트리거링 이벤트는,
미리 결정된 시간 기간의 만료, 또는
상기 컴퓨팅 디바이스가 준수하는지 여부를 결정하는 관리자, 감사자 및/또는 엔티티로부터의 요청 중 하나를 포함하는,
컴퓨터 판독가능 저장 매체.

청구항 18

제9항에 있어서,
상기 사용자와 상기 컴퓨팅 디바이스의 쌍은 상기 서버의 테이블에 저장되고, 상기 테이블의 각 엔트리는 사용자-디바이스 쌍 및 상기 사용자-디바이스 쌍에 의해 구현된 대응하는 구성 설정을 지정하는,

컴퓨터 판독가능 저장 매체.

청구항 19

제1항에 있어서,
 상기 수신된 구성 정보는 XML 파일에 있고,
 상기 수신된 구성 정보의 에러를 결정하는 것은,
 상기 XML 파일을 파싱하는 것과,
 상기 XML 파일을 검증하는 것과,
 상기 서버에 상기 에러를 보고하는 것을 포함하는,
 컴퓨팅 디바이스.

청구항 20

제5항에 있어서,
 상기 수신된 구성 정보는 XML 파일에 있고,
 상기 수신된 구성 정보의 에러를 결정하는 단계는,
 상기 XML 파일을 파싱하는 단계와,
 상기 XML 파일을 검증하는 단계와,
 상기 서버에 상기 에러를 보고하는 단계를 포함하는,
 방법.

발명의 설명

배경 기술

[0001] 모바일 디바이스 관리(mobile device management)(MDM)는 직원이 생산성을 유지하고 기업 정책을 위반하지 않도록 보장하는 하나의 방식이다. 많은 조직체들은 MDM 제품/서비스를 사용하여 직원의 활동을 통제하고 있다. MDM은 주로 기업 데이터 분리, 이메일 보안, 디바이스 상에서의 기업 문서 보안, 기업 정책 시행, 다양한 범주의 랩탑 및 핸드헬드를 포함한 모바일 디바이스들의 통합 및 관리를 다루고 있다. MDM은 조직체의 네트워크 내의 모든 모바일 디바이스의 데이터 및 구성 설정을 제어하고 보호함으로써 지원 비용과 비즈니스 위험을 줄일 수 있다.

발명의 내용

[0002] 본 개요는 선정된 개념을 단순화된 형태로 소개하기 위해 제공되며, 그 개념은 아래의 상세한 설명에서 추가로 설명된다. 이 개요는 청구된 발명의 요지의 핵심 특징 또는 필수 특징을 식별하기 위한 것도 아니고 청구된 발명의 요지의 범위를 제한하도록 사용되는 것도 아니다.

[0003] 본원에 기술되는 실시예는 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하는 것에 관한 것이다. 예를 들어, 컴퓨팅 디바이스는, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 구현되는 준수 설정을 기술하는 구성 정보를 서버로부터 수신할 수 있다. 더욱이, 컴퓨팅 디바이스는 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 구성 정보에 의해 (예컨대, 식별자에 의해) 표시된 상태 머신을 식별할 수 있다. 상태 머신은 준수 설정으로 컴퓨팅 디바이스를 구성하도록 실행될 수 있다. 상태 머신을 실행할 때, 구성 정보에 포함된 페이로드는 종속성과 함께 컴퓨팅 디바이스를 구성하기 위해 상태 머신에 의해 지시된 순서로 처리될 수 있다.

[0004] 본 발명의 다양한 실시예의 구조 및 동작뿐만 아니라, 본 발명의 추가적인 특징 및 이점은 첨부된 도면을 참조하여 아래에 상세히 기술된다. 본 발명은 본원에 기술된 특정 실시예에 제한되지 않는다는 것에 주목해야 한다. 그러한 실시예는 단지 예시적인 목적으로만 본원에 제시된다. 추가적인 실시예는 본원에 포함된 교시 내용에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.

도면의 간단한 설명

[0005] 본원에 포함되고 명세서의 일부를 형성하는 첨부 도면은 실시예를 도시하고, 상세한 설명과 함께, 추가로 실시예의 원리를 설명하고 관련 기술 분야의 기술자가 실시예를 제조 및 사용하게 하는 역할을 한다.

도 1은 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 시스템의 블록 다이어그램을 도시한 것이다.

도 2a는 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 시스템의 블록 다이어그램을 도시한 것으로, 즉 도 1의 시스템의 보다 상세한 예이다.

도 2b는 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 시스템의 다른 블록 다이어그램을 도시한 것으로, 즉, 도 1의 시스템의 보다 상세한 예이다.

도 3은 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 예시적인 방법의 플로우차트를 도시한 것이다.

도 4는 예시적인 실시예에 따라, 모바일 디바이스가 적합하게 구성되었는지를 결정하기 위한 예시적인 방법의 플로우차트를 도시한 것이다.

도 5는 예시적인 실시예에 따라, 구성 결과를 저장하고 송신하기 위한 예시적인 방법의 플로우차트를 도시한 것이다.

도 6은 일 실시예에 따라, 구성 정보의 에러의 정정을 요청하기 위한 예시적인 방법의 플로우차트를 도시한 것이다.

도 7은 일 실시예에 따라, 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하기 위한 예시적인 방법의 플로우차트를 도시한 것이다.

도 8은 실시예들이 구현될 수 있는 예시적인 사용자 디바이스의 블록 다이어그램이다.

도 9는 실시예를 구현하는 데 사용될 수 있는 예시적인 컴퓨팅 디바이스의 블록 다이어그램이다.

본 발명의 특징 및 이점은, 도면과 관련하여 취해질 때 아래에 설명된 상세한 설명으로부터 더욱 명백해질 것이며, 도면에서 유사한 참조 문자들은 전체에 걸쳐 대응하는 요소들을 식별하고 있다. 도면에서, 유사한 참조 번호들은 일반적으로 동일한, 기능적으로 유사한, 및/또는 구조적으로 유사한 요소들을 나타낸다. 요소가 처음 나타나는 도면은 대응하는 참조 번호의 최좌측 숫자로 표시된다.

발명을 실시하기 위한 구체적인 내용

[0006] I. 도입

[0007] 본 명세서 및 첨부 도면은 본 발명의 특징을 포함하는 하나 이상의 실시예를 개시하고 있다. 본 발명의 범위는 개시된 실시예에 제한되지는 않는다. 개시된 실시예는 단지 본 발명을 예시할 뿐이며, 개시된 실시예의 변형된 버전도 본 발명에 포함된다. 본 발명의 실시예는 본원에 첨부된 청구항에 의해 정의된다.

[0008] 본 명세서에서의 "일 실시예", "실시예", "예시적인 실시예", "예시적" 등은 언급된 실시예가 특정의 특징, 구조, 또는 특성을 포함할 수 있지만, 모든 실시예가 그러한 특정의 특징, 구조, 또는 특성을 반드시 포함할 수 있는 것이 아닐 수도 있다는 것을 나타낸다. 또한, 이러한 문구는 반드시 동일한 실시예를 지칭하는 것은 아니다. 또한, 특정의 특징, 구조 또는 특성이 일 실시예와 관련하여 기술될 때, 명시적으로 기술되든 아니든 간에 그러한 특징, 구조 또는 특성을 다른 실시예들과 관련하여 수행하는 것은 본 기술 분야의 기술자들의 지식 범위 내에 속하는 것으로 언급된다.

- [0009] 다수의 예시적인 실시예가 다음과 같이 기술된다. 주목할 것은 본원에 제공된 모든 섹션/서브 섹션 제목은 제한하려는 의도가 아니라는 것이다. 실시예는 본 문서 전반에 걸쳐 설명되며, 임의의 타입의 실시예가 임의의 섹션/서브 섹션 아래에 포함될 수 있다. 더욱이, 임의의 섹션/서브 섹션에 개시된 실시예는 동일한 섹션/서브 섹션 및/또는 상이한 섹션/서브 섹션에서 기술되는 임의의 다른 실시예와 임의의 방식으로 결합될 수 있다.
- [0010] II. 디바이스 준수 관리 시스템 및 방법
- [0011] 기업을 위한 오늘날의 모바일 디바이스 관리(MDM) 프로세스의 전형적인 구조는 MDM 서버를 사용하여, 모바일 디바이스 구성(mobile device configurations)을 실행 및 모니터링하고, 대량의 데이터를 저장하고, 그리고 지속적인 에러 체크 및 예방적 유지 관리를 수행한다. 이러한 기능은 수백, 수천, 수만, 수백만, 수억 및 훨씬 더 많은 수의 디바이스들을 포함하는 대량의 기업용 디바이스용으로 수행될 수 있다. 결과적으로 서버는 수많은 '컴퓨팅 칼로리'를 사용하여(예컨대, 대량의 처리 능력을 사용하여), 기업의 네트워크의 모든 모바일 기술 리소스가 올바르게 구성되어 있는지, 보안 및 기타 중요한 프로토콜의 범위 내에서 작동하는지, 그리고 업데이트를 구현하고 준수하는지를 결정한다.
- [0012] 이것은, 서버가 기업의 네트워크 상에서 많은 개별 모바일 디바이스와 통신하고 궁극적으로 구성 프로세스 및 보고 요구를 감독해야하기 때문에, 어려움을 겪게 된다는 것을 의미한다. 확장성(scalability)은 단순히 디바이스의 수, 순수한 트래픽 및 구성 볼륨을 감안할 때 이 시나리오에서는 기업에 매우 어려운 문제인 것으로 드러나고 있다. 즉, 서버에는 관리해야 할 작업이 엄청나게 많다. 게다가, 이 구성 프로세스는 또한 구성 시행(configuration enforcement)의 비능률적인 디바이스 별 특성으로 인해 다른 방식으로 방지가능한 어려움이 가득 차 있을 수 있다.
- [0013] 이러한 한계는 디바이스 관리 프로세스에서 서버의 관여를 변경함으로써 궁극적으로 해결될 수 있다. 예를 들어, 실시예에서, 서버는 모바일 디바이스가 자체 구성을 수행하도록 요구함으로써 보다 큰 생산성을 경험할 수 있다. 본원에 기술된 실시예에서, 서버는 구성 세부 사항을 모바일 디바이스에 제공할 수 있고, 구성 프로세스는 일반화된 상태 머신으로서 동작함으로써 모바일 디바이스에 의해 구동될 수 있다. 이들 및 추가적인 실시예들은 후술되는 바와 같이 도 1과 관련하여 추가적으로 상세하게 기술된다. 주목할 것은 본원에 기술되는 실시예는 Apple iOS® 및 Google Android™와 같은 임의의 운영 체제/컴퓨팅 플랫폼에 적용 가능하며, 본원에서 참조되는 Microsoft® Windows®, 및/또는 그 컴포넌트는 설명 및 예시 목적으로만 사용되며, 제한적인 것으로 간주되어서는 안된다는 것이다.
- [0014] 도 1은 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 준수 시행 시스템(compliance enforcement system)(100)의 블럭 다이어그램이다. 도 1에 도시된 바와 같이, 시스템(100)은 서버(102), 컴퓨팅 디바이스(104), 및 기업 네트워크(110)를 통해 통신 가능하게 연결된 하나 이상의 데이터 저장소(108)를 포함한다. 서버(102)는 준수 엔진(compliance engine)(122)을 포함한다. 컴퓨팅 디바이스(104)는 준수 에이전트(120)를 포함한다. 단일 컴퓨팅 디바이스(104)가 도시되어 있지만, 서버(102)에 의해 제공되는 준수 설정을 갖는, 수만, 수백, 수천, 수만, 수백만, 수억, 및 보다 큰 수량의 컴퓨팅 디바이스를 포함한 임의의 수의 컴퓨팅 디바이스가 시스템(100)에 존재할 수 있다. 데이터 저장소(들)(108)는 하나 이상의 물리적 메모리 및/또는 저장 디바이스(들)를 포함할 수 있다. 데이터 저장소(들)(108)는, 본원에서 기술되고, 및/또는 본 개시 내용의 이점을 갖는 관련 기술 분야(들)의 기술자에 의해 이해되는 임의의 타입의 물리적 메모리 및/또는 저장 디바이스일 수 있다. 기업 네트워크(110)는, 하나 이상의 기업 위치에서의 기업 디바이스들(예컨대, 컴퓨팅 디바이스(104) 및 도시되지 않은 추가적인 컴퓨팅 디바이스들)을 다른 기업 디바이스들에 상호 연결하고 기업 디바이스들이 컴퓨팅 리소스들을 액세스 및/또는 공유할 수 있도록 하기 위해, 기업에 의해 수립된 사설 컴퓨터 네트워크를 포함한다.
- [0015] 컴퓨팅 디바이스(104)는 기업의 하나 이상의 구성원(예컨대, 직원)에 의해 이용되거나 그렇지 않으면 액세스될 수 있는 디바이스이다. 본원에서 사용되는 용어 "기업"은 광범위하게 비즈니스, 비영리 조직체, 및 정부 기관을 포함한 다양한 조직체 타입들 중 임의의 것을 지칭한다. 컴퓨팅 디바이스(104)의 사용자는 본원에서 "기업 사용자" 또는 단순히 "사용자"로 지칭될 수 있다. 컴퓨팅 디바이스(104)는, 예를 들어, 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 컴퓨터, 넷북, 스마트폰 등을 포함할 수 있지만, 이에 제한되지는 않는다. 컴퓨팅 디바이스(104)의 추가적인 예가 도 8 및 도 9를 참조하여 아래에서 기술된다.
- [0016] 데이터 저장소(들)(108)는 컴퓨팅 디바이스(104) 및/또는 하나 이상의 다른 컴퓨팅 디바이스들에 대한 구성 정보(112)를 저장하도록 구성될 수 있다. 실시예들에서, 구성 정보(112)는 서버(102)의 메모리에 로컬로 저장될 수 있다. 구성 정보(112)는 특정 구성 시나리오에 따라, 컴퓨팅 디바이스(112)에 설치하기 위한 데이터, 하나

이상의 파일 등을 포함할 뿐만 아니라 컴퓨팅 디바이스(104) 및/또는 컴퓨팅 디바이스(104)의 사용자를 위한 하나 이상의 준수 설정을 지정할 수 있다. 예를 들어, 구성 정보(112)의 제 1 준수 설정은 컴퓨팅 디바이스(104)의 제 1 사용자와 연관될 수 있고, 구성 정보(112)의 제 2 준수 설정은 컴퓨팅 디바이스(104)의 제 2 사용자와 연관될 수 있다. 준수 설정의 예는 컴퓨팅 디바이스(104)에 의해 구현될 암호화 설정, 컴퓨팅 디바이스(104)에 의해 구현될 보안 설정, 컴퓨팅 디바이스(104) 상에 설치되어야 하는 애플리케이션 또는 운영 체제 중 적어도 하나의 최소 버전 등을 포함하지만 이에 제한되지는 않는다. 암호화 설정(들)은 컴퓨팅 디바이스(104) 내에 포함된 저장 디바이스가 (예컨대, BitLocker™와 같은 그러나 이에 제한되지 않는 암호화 프로그램을 통해) 암호화되어야 하는지 여부를 지정할 수 있다. 보안 설정은 컴퓨팅 디바이스(104)에 의해 구현될 암호 정책(예컨대, 암호 길이를 최소 10 개의 문자, 12 개의 문자 등으로 설정), 코드 서명이 컴퓨팅 디바이스(104)에 의해 구현되어야 하는지의 여부, 신뢰되는 플랫폼 모듈(trusted platform module)(TPM)이 컴퓨팅 디바이스(104)에 의해 구현되어야 하는지의 여부 등을 지정할 수 있다. 주목되는 것은 위에서 설명한 준수 설정은 순전히 예시이며 다른 구성 설정이 사용될 수 있다는 것이다.

[0017] 서버(102)는 구성 시나리오와 관련하여, 컴퓨팅 디바이스(104)를 포함하는 컴퓨팅 디바이스의 준수를 관리하도록 구성될 수 있다. 이러한 컴퓨팅 디바이스는 고정 디바이스(예컨대, 데스크탑 컴퓨터) 및/또는 모바일 디바이스(예컨대, 스마트폰, 랩탑, 태블릿 컴퓨터, 웨어러블 컴퓨팅 디바이스(예컨대, Google® Glass™, 가상 현실 헤드셋, 스마트 시계 등) 등)를 포함할 수 있다. 모바일 디바이스를 관리할 때, 서버(102)는 또한 모바일 디바이스 관리자(mobile device manager)(MDM)라고 지칭될 수도 있다. 구성 시나리오는 개별 정책 세트만큼 간단하거나 가상 사설망(VPN) 연결을 위한 구성만큼 복잡할 수 있다. 정책(예컨대, 데이터 및/또는 보안 정책)은 기업에 의해 지정될 수 있다. 정책은 하나 이상의 준수 설정에 따라 지정될 수 있다. 준수 설정(들)은 기업의 관리인(예컨대, 기업 사용자를 대신하여 컴퓨팅 디바이스(104)의 구축, 유지 관리, 및/또는 구성을 담당할 수 있는 기업 내 정보 기술(IT) 관리인 또는 다른 사람)에 의해 지정될 수 있다.

[0018] 도 1에 도시된 바와 같이, 서버(102)는 준수 엔진(122)을 포함한다. 준수 엔진(122)은 컴퓨팅 디바이스(104)에 제공될 구성 정보(112)를 결정할 수 있고, 결정된 구성 정보(112)를 컴퓨팅 디바이스(104)에 제공할 수 있다. 예를 들어, 서버(102)는 특정 디바이스 세트에 대한 구축을 위한 구성 설정을 나타내는 준수 매니페스트 또는 다른 문서의 형태로 구성 정보(112)를 수신 및/또는 액세스할 수 있다. 서버(102)는 데이터 저장소(108)로부터 네트워크(110)를 통해 구성 정보(112)를 획득할 수 있고, 네트워크(110)를 통해 컴퓨팅 디바이스(104)와 같은 표시된 디바이스에 구성 정보(112)를 제공할 수 있다. 네트워크(110)를 통한 구성 정보(112)의 전송은 도 1에서 점선(112)으로 도시되어 있다. 구성 정보(112)는 컴퓨팅 디바이스(104)가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스(104)(및 선택적으로 추가 디바이스들)에 의한 구현을 위한 준수 설정을 지정할 수 있다.

[0019] 컴퓨팅 디바이스(104)는 구성 정보(112)에 의해 지정된 준수 설정(들)을 구현하도록 구성될 수 있다. 도 1에 도시된 바와 같이, 컴퓨팅 디바이스(104)는 준수 에이전트(120)를 포함한다. 컴퓨팅 디바이스(104)는 준수 에이전트(120)를 실행하도록 구성될 수 있으며, 준수 에이전트(120)는, 컴퓨팅 디바이스(104) 상에서 구성 시나리오에 대한 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상태 머신을 식별하고 실행함으로써, 수신된 구성 정보(112)에 의해 지정된 준수 설정을 구현하도록 구성된다. 예를 들어, 준수 에이전트(120)는 상태 머신(118)을 구성 정보(112)에 의해 표시된 상태 머신으로 식별할 수 있고, 상태 머신(118)을 실행하여, 구성 시나리오를 준수하기 위한 준수 설정으로 컴퓨팅 디바이스(104)를 구성할 수 있다.

[0020] 실시예에 따르면, 상태 머신(118)은 다양한 방식으로 생성될 수 있다. 예를 들어, 빌드(build) 프로세스 동안, 하나 이상의 문서가 (예컨대, 개발자에 의해) 생성될 수 있으며, 컴퓨팅 디바이스(104) 상에 컴파일 및 저장될 수 있다. 예를 들어, 서버가 (예컨대, 서버(102)에서의 준수 엔진(122)에 의해 생성된 관리인 콘솔에서) 사용자 인터페이스(UI)를 소비하고 렌더링할 수 있도록 하는 구성 시나리오를 기술하는 하나 이상의 파일 및 상태 머신이 빌드 동안 생성된다. 이 UI는 IT 관리인이 컴퓨팅 디바이스(104)로 전송될 수 있는 특정 구성 시나리오에 대한 구성 정보를 입력할 수 있게 한다. 구성 정보는, 특정 구성 시나리오에 따라 컴퓨팅 디바이스의 보안 및 기능적 동작을 변경하는 구성 서비스 제공자(configuration service provider)(CSP)를 이용하는 본원에 기술된 컴퓨팅 디바이스(104)에 의해 처리된다. CSP는 디바이스에 대한 구성 설정을 판독하고, 설정하고, 수정하거나, 또는 삭제할 수 있는 인터페이스이다. 게다가, 상태 머신(예컨대, 상태 머신(118))을 처리할 때, CSP는 상태 머신에 의해 지시된 순서에 따라 호출되어, 서버에 의해 제공되는 구성 정보를 사용하여 컴퓨팅 디바이스에서 원하는 상태를 충족시키게 된다.

[0021] 상태 머신(118)에 따라 구성 설정을 구현한 후, 준수 에이전트(120)는 준수 엔진(122)에 확인 응답(acknowledgment)(예컨대, "경고")을 제공할 수 있다. 확인 응답은 컴퓨팅 디바이스가 구성 설정을 구현했거나

오류가 발생했음을 나타낸다. 구성 설정을 구현하는 데 성공했음을 나타내는 확인 응답을 수신하면, 준수 엔진(112)은 구성 시나리오를 준수하는 것으로 컴퓨팅 디바이스(104)를 지정할 수 있다. 준수 엔진(112)은 다양한 방식으로 확인 응답을 유지할 수 있다.

[0022] 예를 들어, 서버(102)는 특정 사용자-디바이스 쌍이 준수하는지 여부에 관한 각 사용자-디바이스 쌍에 대한 지정을 포함하는 준수 레코드(106)를 유지할 수 있다. 예를 들어, 준수 레코드(106)는 복수의 엔트리를 포함하는 데이터 구조(예컨대, 테이블)를 포함할 수 있으며, 여기서 각 엔트리는 특정 사용자-디바이스 쌍 및 해당 사용자-디바이스 쌍이 준수하는지의 여부에 대한 표시를 지정한다. 서버(102)는 또한 각 사용자-디바이스 쌍에 대해 구현된 구성 설정의 로컬 구성 레퍼런스(local configuration reference)(114)를 유지하도록 구성될 수 있다. 구성 레퍼런스(114)는 복수의 엔트리를 포함하는 데이터 구조(예컨대, 테이블)를 포함할 수 있으며, 여기서 각 엔트리는 특정 사용자-디바이스 쌍 및 해당 쌍에 의해 구현된 구성 설정을 지정한다. 위에서 설명된 구성 레퍼런스(114) 및/또는 준수 레코드(106)의 구조 및/또는 조직체는 순전히 예시적이며 다른 구조 및/또는 조직체가 사용될 수 있다는 것에 주목해야 한다. 구성 레퍼런스(114) 및 준수 레코드(106)는 데이터 저장소(108) 또는 다른 곳에 저장되는 것을 포함하여 임의의 위치에 그리고 임의의 방식으로 서버(102)에 의해 유지될 수 있다.

[0023] 준수 엔진(122)은 컴퓨팅 디바이스(들)(104)가 레퍼런스(114)를 사용하여 준수 규칙의 준수를 유지하는지의 여부를 결정하도록 구성될 수 있다. 그렇게 함으로써, 준수 엔진(122)은 컴퓨팅 디바이스(들)(104)에게 그 구성 설정을 질의하기 보다는, 컴퓨팅 디바이스(들)(104)의해 구현된 구성 설정을 결정하기 위해 단순히 레퍼런스(114)에 액세스하기만 하면 된다. 준수 엔진(122)은, 미리 결정된 시간 기간의 만료, 컴퓨팅 디바이스(104) 및/또는 다른 컴퓨팅 디바이스가 하나 이상의 준수 설정 및/또는 요구 사항을 준수하는지의 여부를 관리하거나, 확인할 것을 원하거나, 또는 알고/확인하는 데 관심이 있는 관리자, 감사자, 및/또는 다른 사람 및/또는 엔티티로부터의 요청을 포함한 임의의 적절한 트리거링 이벤트를 검출하면, 컴퓨팅 디바이스(들)(104)가 준수를 유지하는지 여부를 결정할 수 있다. 디바이스가 준수 설정/요구 사항을 준수하는지 여부를 결정하기 위한 요청은 수백, 수천 또는 수백만 개의 디바이스를 포함한 임의의 수의 디바이스와 관련될 수 있다. 따라서, 구성 레퍼런스(114) 및/또는 준수 레코드(106)로부터의 이러한 준수 정보에 액세스함으로써, 관심있는 각 디바이스를 개별적으로 폴링해야 하는 것과 비교하여 서버(102)에 의해 대량의 처리 능력이 확장될 필요는 없다.

[0024] 도 1의 시스템(100)은 실시예에서 다양한 방식으로 구현될 수 있다. 예를 들어, 도 2a는 예시적인 실시예에 따른 시스템(200)의 상세한 블록 다이어그램을 도시한 것이다. 시스템(200)은 시스템(100)의 일 예이다. 도 2a에 도시된 바와 같이, 시스템(200)은 서버(102) 및 컴퓨팅 디바이스(204)를 포함한다. 컴퓨팅 디바이스(204)는 도 1을 참조하여 위에서 기술된 바와 같은 컴퓨팅 디바이스(104)의 일 예이다. 도 2a에 더 도시된 바와 같이, 컴퓨팅 디바이스(204)는 준수 에이전트(220), 상태 머신(SM) 데이터 저장소(224), 및 구성 정보(CI) 데이터 저장소(222)를 포함한다. 상태 머신 데이터 저장소(224) 및 구성 정보 데이터 저장소(222)는 각각, 본원에 기술되고, 및/또는 본 개시 내용의 이점을 갖는 관련 기술 분야(들)의 기술자에 의해 이해될 임의의 타입의 물리적 메모리 및/또는 저장 디바이스를 포함하여, 하나 이상의 물리적 메모리 및/또는 저장 디바이스(들)를 포함할 수 있다. 준수 에이전트(220)는 도 1을 참조하여 위에 기술된 바와 같은 준수 에이전트(120)의 일 예이다.

[0025] 서버(102)는 컴퓨팅 디바이스(204)에 제공될 구성 정보(112)를 결정하고 결정된 구성 정보(112)를 컴퓨팅 디바이스(104)에 제공하도록 구성된다. 예를 들어, 사용자가 새로 프로비저닝된 컴퓨팅 디바이스에 처음 로그인할 경우, 서버(102)는 해당 컴퓨팅 디바이스 및/또는 사용자와 연관된 디바이스 구성 정보(112)를 검색하고 구성 정보(112)를 컴퓨팅 디바이스에 제공할 수 있다. 추가적으로 또는 대안적으로, (디바이스 드라이버를 포함하는) 소프트웨어 또는 펌웨어가 컴퓨팅 디바이스에 새로 설치되는 경우, 새로운 하드웨어가 컴퓨팅 디바이스에 추가되는 경우, 소프트웨어 및/또는 펌웨어가 컴퓨팅 디바이스에서 업데이트되거나 삭제되는 경우, 하드웨어가 컴퓨팅 디바이스에서 제거되는 경우, 및/또는 컴퓨팅 디바이스의 구성을 수정하는 다른 이벤트 동안, 서버(102)는 해당 컴퓨팅 디바이스 및/또는 사용자와 연관된 구성 정보(112)를 검색하고, 구성 정보(112)를 컴퓨팅 디바이스에 제공할 수 있다.

[0026] 도 1을 참조하여 기술된 컴퓨팅 디바이스(104)와 유사하게, 컴퓨팅 디바이스(204)는 구성 정보(112)에 의해 지정된 구성 설정을 구현하도록 구성된 준수 에이전트(220)를 실행하도록 구성될 수 있다. 보다 구체적으로, 준수 에이전트(220)는 원하는 구성 시나리오에 기반하여 적절한 상태 머신을 소비한다. 서버(102)는 컴퓨팅 디바이스(204)에 적절한 상태 머신을 지정하기 위해 구성 정보(112) 내에 "구성 시나리오 식별자"를 제공할 수 있다. 준수 에이전트(220)는 이 식별자를 이용하여 오케스트레이터(orchestrator)가 따르게 될 올바른 상태 머신을 찾을 수 있다. 서버(102)에 의해 제공되는 구성 정보(112)는 그 후, 상태 머신에 기반하여 준수 에이전트(220)에 의

해 적절한 CSP들에 제공되며, 이는 아래에서 보다 상세히 기술된다.

[0027] 도 2a에 도시된 바와 같이, 준수 에이전트(220)는 구성 관리자(206) 및 오케스트레이터 엔진(208)을 포함한다. 또한, 구성 관리자(206)는 준수 구성 정보 핸들러(228) 및 구성 정보 에러 체커(210)를 포함한다.

[0028] 준수 구성 정보 핸들러(228)는 (도 2a에서 서버(112), 준수 구성 정보 핸들러(228), 및 구성 정보 데이터 저장소(222) 사이에서 점선(112)으로 도시되는 바와 같이) 서버(102)의 준수 엔진(122)으로부터 (예컨대, SyncML 또는 콘텐츠 전달 네트워크(Content Delivery Network)(CDN)를 통해 컴퓨팅 디바이스(204)로 전달되는) 구성 정보(112)에 액세스하고 구성 정보(112)를 구성 정보 데이터 저장소(222)에 저장하도록 구성될 수 있다. 다른 예로서, 구성 정보(112)는 JSON (JavaScript Object Notation)에 의해 제공될 수 있다. 구성 정보(112)는 서버에 의해 컴퓨팅 디바이스로 직접 전달될 수 있거나, 서버는 CDN을 통해 구성 정보(112)를 획득할 수 있는 위치를 컴퓨팅 디바이스에 지시할 수 있다(아래에서 더 자세히 설명됨).

[0029] 구성 정보(112)는 임의의 적합한 형태를 가질 수 있다. 일 실시예에 따르면, 구성 정보(112)는 준수 설정과 연관된 페이로드 및 페이로드의 다운로드 및 처리를 위한 대응하는 URI (Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 표현될 수 있다. 예를 들어, CDN 저장 페이로드에는 페이로드의 비동기 다운로드 및 처리를 위한 URI를 나타내는 별도의 태그가 포함될 수 있다. 추가로 설명하기 위해, (예컨대, 정책 CSP에 의해 제공되는) 개별 정책 세트에 대한 간단한 구성 시나리오는 XML 파일에 포함된 단일 페이로드만 요구할 수 있다. 보다 복잡한 구성 시나리오(예컨대, VPN 연결을 위한 구성)의 경우, 다수의 페이로드는 XML 파일에 지정될 필요가 있다. 예를 들어, VPN 구성 시나리오에는 프로파일 페이로드 및 인증서 페이로드가 제공될 수 있다. 이러한 시나리오 구성 파일은 컴퓨팅 디바이스와의 동기화 세션을 시도하기 전에 서버에 의해 생성될 수 있다. 이렇게 하면 서버와 상호 작용하는 각 컴퓨팅 디바이스에 대해 요청 시 동일한 파일이 생성되는 것을 방지하고, 서버가 보다 유용한 작업을 수행할 수 있도록 하여 확장성을 높인다.

[0030] 이하는 다수의 선언적 구성 페이로드를 포함하는 정책 CSP에 의해 설정될 정책을 나타내는 시나리오 구성 XML 파일의 일 예이다. 예시적인 시나리오 구성 XML 파일의 "Id", "checksum" 및 "OSDefmedScenario" 속성들은 고유 식별자를 구성한다는 것에 주목해야 한다. 또한, 이 예에서 "OSDefmedScenario" 속성은 이것이 게시된 구성 시나리오임을 나타내며, 이는 구성 시나리오가 구성 시나리오를 구성하는 방법을 지시하는 선언된 구성 시나리오 상태 머신에 의존한다는 것을 의미하는 것에 주목해야 한다. 이 상태 머신은, 운영 체제(OS) 빌드가 선언된 구성 시나리오 상태 머신 Windows® 매니페스트를 컴파일한 후, 레지스트리에 저장될 수 있다. 또한, 각 페이로드가 원자적이어야 하는지의 여부 또는 전체 문서가 원자적이어야 하는지의 여부를 지정할 필요가 없을 수 있는데, 이는 이러한 동작이 상태 머신에서 지정될 수 있기 때문이다. 또한, 아래의 XML 파일은 일부 컴퓨팅 디바이스 전체 정책을 설정하고, 예시적인 암호화 특징(예컨대, Microsoft® BitLocker™)을 구성한다:

```
<DeclaredConfiguration context="Device" schema="1.0" id="12345"
  OSDefinedScenario="MSFTMinimumSecurityBaseline" checksum="22280"
  operation="Set">
<CSP name="/Vendor/MSFT/Policy" >
  <URI path="Config/DeviceLock/DevicePasswordEnabled" type="int">0</URI>
  <URI path="Config/DeviceLock/AllowSimplePassword" type="int">1</URI>
  <URI path="Config/DeviceLock/MinDevicePasswordlength" type="int">6</URI>
```

[0031]

```

<URI path="Config/ApplicationManagement/AllowAllTrustedApps"
  type="int">1</URI>
<URI path="Config/Experience/TestPolicyBinary"
  type="b64">BAMCAQ==</URI>
<URI path="Config/Browser/HomePages" type="chr">Hello world!</URI>
<URI path="Config/Experience/BogusPolicy" type="chr">Hello world!</URI>
</CSP>
<CSP name="/Vendor/MSFT/BitLocker"
  <URI path="RequireStorageCardEncryption" type="int">1</URI>
  <URI path="RequireDeviceEncryption" type="int">1</URI>
  <URI path="EncryptionMethodByDriveType" type="chr">
    <![CDATA[<enabled>]]>
<data id="EncryptionMethodWithXtsOsDropDown_Name" value="4"/>
<data id="EncryptionMethodWithXtsFdvDropDown_Name" value="7"/>
<data id="EncryptionMethodWithXtsRdvDropDown_Name" value="4"/>]]>
  </URI>
</CSP>
</DeclaredConfiguration>

```

[0032]

[0033]

다음은 MDM에 의해 정의된 구성 시나리오를 기술하는 다중 페이로드 XML 파일의 일 예이다. 이 예에서는 선언된 구성 시나리오 상태 머신이 없기 때문에:

```

<DeclaredConfiguration context="Device" schema="1.0" id="12346"
  MDMDefinedScenario="optional My Blah Scenario" atomic="false"
  checksum="5567" operation="Set">
  <CSP name="/Vendor/MSFT/Policy">
    <URI path="Config/DeviceLock/DevicePasswordEnabled" type="Int">0</URI>
    <URI path="Config/DeviceLock/AllowSimplePassword" type="Int">1</URI>
    <URI path="Config/DeviceLock/MinDevicePasswordlength" type="Int">6</URI>
    <URI path="Config/ApplicationManagement/AllowAllTrustedApps"
      type="Int">1</URI>
    <URI path="Config/Experience/TestPolicyBinary"
      type="b64">BAMCAQ==</URI>
    <URI path="Config/Browser/HomePages" type="chr">Hello world!</URI>
    <URI path="Config/Experience/BogusPolicy" type="chr">Hello world!</URI>
  </CSP>
  <CSP name="/Vendor/MSFT/BitLocker" atomic="true"
    <URI path="RequireStorageCardEncryption" type="int">1</URI>
    <URI path="RequireDeviceEncryption" type="int">1</URI>
    <URI path="EncryptionMethodByDriveType" type="chr">
      <![CDATA[<enabled>]]>
    <data id="EncryptionMethodWithXtsOsDropDown_Name" value="4"/>
    <data id="EncryptionMethodWithXtsFdvDropDown_Name" value="7"/>
    <data id="EncryptionMethodWithXtsRdvDropDown_Name" value="4"/>]]>
    </URI>
  </CSP>
</DeclaredConfiguration>

```

[0034]

[0035]

[0036]

준수 구성 정보 핸들러(228)는 서버(102)로부터 구성 정보(112)를 (예컨대, 하이퍼텍스트 전송 프로토콜 (Hypertext Transfer Protocol)(HTTP) GET 방법 요청을 사용하여) 요청하고, 구성 정보(112)를 구성 정보 에러 체커(210)에 제공하도록 추가로 구성될 수 있다.

[0037]

구성 정보 에러 체커(210)는 구성 정보(112)를 파싱하고, 구성 정보(112)의 에러(예컨대, XML 구성 시나리오 파일의 XML 코드, 속성, URI 등의 에러들)를 결정하고, 그리고 서버(102)로부터 에러의 정정을 요청하도록 구성될

수 있다. 예를 들어, 구성 정보(112)가 XML 파일로서 표현되는 위에서 논의된 예에서, 구성 정보 에러 체커(210)는 XML 코드를 파싱 및 검증하고, URI를 레지스트리 위치에 저장하고, 그리고 임의의 에러가 검출되면, 에러를 서버(102)에 동기적으로 보고하고 에러가 서버(102)에 의해 수정되도록 요청할 수 있다(도 2a에서 점선(216)으로 도시됨). 일반적으로, MDM 서버는 구성 정보에서 충돌이나 에러가 없도록 보장한다. 그러나, 클라이언트 디바이스는 충돌 및 에러를 검출하면, MDM 서버에 즉각적인 피드백을 제공하고, 구성 정보의 임의의 에러를 빠르게 정정할 수 있다.

[0038] 준수 구성 정보 핸들러(228)는 저장된 구성 정보(112)를 서버(102)에 송신하고, 구성 정보를 서버(102)에 송신하는 것에 응답하여 서버(102)로부터 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하도록 추가로 구성될 수 있다. 예를 들어, 준수 구성 정보 핸들러(228)는 구성 정보 데이터 저장소(222)에 저장된 구성 정보 데이터(112)를 서버(102)로 전송하고 서버(102)에 의해 결정된 저장된 구성 정보(112)의 상태에 기반하여 구성 정보 데이터(112)의 상태(예컨대, 만료 또는 부실)를 검증하도록 구성될 수 있고, 서버(102)는 업데이트된 구성 정보 및/또는 추가 구성 정보를 준수 구성 정보 핸들러(228)에 제공할 수 있다. 후속적으로, 준수 구성 정보 핸들러(228)는 수신된 업데이트된 구성 정보 및/또는 추가 구성 정보를 구성 정보 데이터 저장소(222)에 저장할 수 있다(도 2a에서 점선(214)으로 도시됨). 후술되는 바와 같이, 준수 에이전트(220)의 오케스트레이터 엔진(208)은 구성 정보(112)에서 수신된 구성 설정을 구현하도록 구성되고, 이에 기반하여 구성 결과(226)를 생성한다.

[0039] 특히, 도 2b는 예시적인 실시예에 따른 시스템(230)의 상세한 블럭 다이어그램을 도시한 것이다. 시스템(230)은 시스템(100)의 다른 예이다. 도 2b에 도시된 바와 같이, 시스템(230)은 서버(102) 및 컴퓨팅 디바이스(244)를 포함한다. 컴퓨팅 디바이스(244)는 도 1의 컴퓨팅 디바이스(104)의 또 다른 예이다. 또한, 컴퓨팅 디바이스(244)는 준수 에이전트(240), SM 데이터 저장소(224), 및 구성 정보 데이터 저장소(222)를 포함한다. 준수 에이전트(240)는 도 1을 참조하여 위에 기술된 바와 같은 준수 에이전트(120)의 다른 예이다.

[0040] 도 1을 참조하여 기술된 컴퓨팅 디바이스(104)와 유사하게, 컴퓨팅 디바이스(244)는 구성 정보(112)에 의해 지정된 구성 설정을 구현하도록 구성된 준수 에이전트(240)를 실행한다. 도 2b에 도시된 바와 같이, 준수 에이전트(240)는 구성 관리자(206) 및 오케스트레이터 엔진(208)을 포함한다. 구성 관리자(206)는 준수 구성 정보 핸들러(228) 및 설정 구현 트리거러(settings implementation triggerer)(232)를 포함한다. 오케스트레이터 엔진(208)은 상태 머신 선택기(234) 및 상태 머신 실행 엔진(236)을 포함한다. 준수 에이전트(240)의 이러한 특징은 다음과 같이 기술된다.

[0041] 설정 구현 트리거러(232)는 컴퓨팅 디바이스(244)에서 구성 정보(112)에 의해 기술된 준수 설정을 구현하기 위해 오케스트레이터 엔진(208)을 트리거하도록 구성된다. 예를 들어, 도 2b에 도시된 바와 같이, 설정 구현 트리거러(232)는 상태 머신 선택기(234)에 트리거(218)를 제공한다. 보다 구체적으로, 트리거(218)는, 도 2a를 참조하여 위에서 기술된 바와 같이 구성 정보 에러 체커(210)가 구성 정보(112)를 파싱하고 검증한 후, 태스크의 스케줄링을 유발할 수 있다(그리고 스케줄링된 태스크는 "DeclaredConfiguration" CSP에 의해 트리거될 수 있다). 실시예에 따르면, 태스크는 활동 큐(activity queue)에 저장되고, 활동 큐에 저장된 임의의 처리되지 않은 태스크가 처리된 후에 처리될 수 있다.

[0042] 상태 머신 선택기(234)는 컴퓨팅 디바이스(244)에서 준수 설정을 구현하기 위해, 설정 구현 트리거러(232)로부터 트리거를 수신하도록 구성된다. 일 실시예에서, 상태 머신 선택기(234)는 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 구성 정보(112)에 의해 표시된 상태 머신을 식별한다. 트리거(218)를 수신한 후, 상태 머신 선택기(234)는 구성 정보 데이터 저장소(222)로부터 저장된 구성 정보(112)를 획득하고, 저장된 구성 정보(112)의 속성을 소비한 후에 상태 머신(118)이 구성 정보(112)와 연관되어 있는지를 결정할 수 있다. 예를 들어, 구성 정보(112)는 특정 상태 머신에 대한 식별자를 지정할 수 있다.

[0043] 상태 머신 선택기(234)는 상태 머신 실행 엔진(236)에 상태 머신(118)을 제공하도록 추가로 구성될 수 있다. 상태 머신 실행 엔진(236)은 상태 머신(118)을 실행하여 구성 정보(112)에 기술된 준수 설정으로 컴퓨팅 디바이스(244)를 구성하도록 구성된다. 예를 들어, 상태 머신 선택기(234)는 상태 머신(118)에 의해 지시된 순서대로 종속성과 함께 구성 정보(112)(도 2a를 참조하여 기술됨)에 포함된 페이로드들을 처리할 수 있다. 대안적으로, 구성 정보(112)와 연관된 상태 머신이 없다면, 구성 정보(112)에 의해 지정된 페이로드들은 구성 정보(112)에 의해 지시된 순서로 처리된다. 또한, 상태 머신 선택기(234)는 상태 머신 데이터 저장소(224)에 액세스하여 상태 머신(118)의 매니페스트를 획득할 수 있다. 매니페스트는 상태 머신(118) 및 그 종속성을 포함할 수 있다.

[0044] 선언된 구성 시나리오 상태 머신 매니페스트의 예시적인 XML 파일이 아래에 재현된다.

```
<?xml version="1.0" encoding="utf-8"?>
<DeclaredConfigurationScenarioStateMachine>
  <StateMachine name="MSFTMinimumSecurity Baseline"
    extensionDLL="%SYSTEMROOT%\system32\MSFTMinimumSecurity Baseline.
    dll" start="SetGeneralPolicies">
    <State name="SetGeneralPolicies" defaultGoto="SetBitlocker">
      <InputArguments></InputArguments>
      <OutputArguments></OutputArguments>
      <CSP name=".\Vendor\MSFT\Policy" >
        <URI path="Config/DeviceLock/DevicePasswordEnabled" type="int">0</URI>
        <!-- Has to be zero since values is set-->
        <URI path="Config/DeviceLock/AllowSimplePassword"></URI>

        <URI path="Config/DeviceLock/MinDevicePasswordlength"></URI>
        <URI path="Config/ApplicationManagement/AllowAllTrustedApps"></URI>
      </CSP>
    </State>
    <State name="SetBitlocker">
      <InputArguments></InputArguments>
      <OutputArguments>
        <Arg name="Status" type="int"/>
      </OutputArguments>
      <URI path="RequireStorageCardEncryption" type="int">1</URI>
      <URI path="RequireDeviceEncryption" type="int">1</URI>
      <URI path="EncryptionMethodByDriveType"></URI>
    </State>
  </StateMachine>
</DeclaredConfigurationScenarioStateMachine>
```

[0045]

[0046]

[0047]

상태 머신 실행 엔진(236)은 구성 결과(226)(예컨대, URI 및 값, 이벤트, 마지막 에러 코드, 상태 등을 포함하는 XML 파일)를 구성 정보 데이터 저장소(222)에 저장하도록 추가로 구성될 수 있다. 예를 들어, 구성 결과(226)는 도 1과 관련하여 위에서 기술된 하나 이상의 준수 레코드(106) 및/또는 구성 레퍼런스(114)를 포함할 수 있다. 준수 구성 정보 핸들러(228)는 구성 정보 데이터 저장소(222)로부터 구성 결과(226)를 획득하고 구성 결과(226)를 준수 엔진(122)(도 2b에서 점선(226)으로 도시됨)에 제공하여 서버(102)에 의해 분석되도록 구성될 수 있다. 구성 결과(226)가 에러를 포함하는 경우, 서버(102)는 오케스트레이터 엔진(208)에게 컴퓨팅 디바이스의 상태를 상태 머신의 실행 전의 상태로 롤백하도록 지시하는 것을 포함하여 모바일 디바이스에 대한 임의의 잘못된 변경이 반전될 수 있도록 준수 구성 정보 핸들러(228)에 이전 버전의 확인 정보(112)를 제공할 수 있다.

[0048]

예를 들어, 본원에 기술된 실시예에 따르면, 상태 머신은 구성 시나리오에 대한 구성 지시 사항을 포함할 수 있다. 추가적인 설명을 위해, 상태는 엔티티들 "FromState", "Activity" 및 "ToState"에 의해 기술될 수 있다. Activity는 구성 시나리오 소유자(들)에 의해 소유되고 생성되는 실행 코드의 형태를 가질 수 있다. 또한, Activity는 런타임시에 상이한 ToState 상태로 리턴하여 다음 단계로 진행하는 데 사용될 수 있다. Activity가 Tostate를 수정하지 않으면, 디폴트 ToState가 사용되며 상태 머신의 데이터 표현에 지정되어 있다. Activity는, XML로 표현되며 Activity에 마샬링되는 미리 정의된 입력 데이터 및 출력 데이터를 가질 수 있다. 이를 통해 개별 Activity가 개발될 수 있고 개별적으로 테스트될 수 있다. Activity가 ToState를 수정하도록 하는 능력으로 인해 상태 머신에 대한 런타임 조정이 가능하다. 상태 머신을 사용하여 구성 시나리오를 구성, 검증, 리프레시, 및 삭제할 수 있다.

[0049]

시스템(100, 200, 및 230)은 자신의 기능을 수행하기 위해 다양한 방식으로 동작할 수 있다. 예를 들어, 도 3은 예시적인 실시예에 따라, 컴퓨팅 디바이스가 구성 시나리오를 준수하도록 하기 위해 컴퓨팅 디바이스에 의해 준수 설정을 구현하기 위한 예시적인 방법의 플로우차트(300)를 도시한 것이다. 도 3는 도 1, 도 2a 및 도 2b을 계속하여 참조하여 기술된다. 그러나, 다른 구조 및 동작 실시예는 플로우차트(300) 및 도 1, 도 2a, 및 도 2b

의 시스템(100, 200, 및 230)에 관한 아래의 논의에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.

- [0050] 도 3에 도시된 바와 같이, 플로우차트(300)는 단계(302)에서 시작된다. 단계(302)에서, 모바일 디바이스가 구성 시나리오를 준수하도록 하기 위해 모바일 디바이스에 의해 구현되는 준수 설정을 기술하는 구성 정보를 서버로부터 수신한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 준수 구성 정보 핸들러(228)는 서버(102)의 준수 엔진(122)으로부터 정보(112)를 수신한다.
- [0051] 단계(304)에서, 구성 정보를 저장한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 준수 구성 정보 핸들러(228)는 구성 정보 데이터 저장소(222)에 구성 정보(112)를 저장한다.
- [0052] 단계(306)에서, 모바일 디바이스에서 준수 설정을 구현하기 위한 트리거를 수신한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 설정 구현 트리거(232)는 오케스트레이터 엔진(208)의 상태 머신 선택기(234)에 트리거(218)를 제공하여, 컴퓨팅 디바이스(244)에서 구성 정보(112)에 의해 기술된 준수 설정의 구현을 시작한다.
- [0053] 단계(308)에서, 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 구성 정보에 의해 표시된 상태 머신을 식별한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 트리거(218)를 수신한 후, 상태 머신 선택기(234)는 구성 정보 데이터 저장소(222)로부터 저장된 구성 정보(112)를 획득하고, 상태 머신(118)이 구성 정보(112)와 연관되어 있는지를 결정한다.
- [0054] 단계(310)에서, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 상태 머신을 실행하여, 모바일 디바이스를 준수 설정으로 구성한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 상태 머신 선택기(234)는 상태 머신(118) (또는 그에 대한 포인터)을 상태 머신 실행 엔진(236)에 제공하고, 상태 머신 실행 엔진(236)은 상태 머신(118)을 실행하여, 컴퓨팅 디바이스(244)를 구성 정보(112)에 기술된 준수 설정으로 구성한다.
- [0055] 도 4는 예시적인 실시예에 따라, 모바일 디바이스가 구성 정보에 따라 적합하게 구성되었는지의 여부를 결정하기 위한 방법의 플로우차트(400)를 도시한 것이다. 도 4는 도 1, 도 2a 및 도 2b를 계속하여 참조하여 기술된다. 그러나, 다른 구조 및 동작 실시예는 플로우차트(400) 및 도 1, 도 2a, 및 도 2b의 시스템(100, 200, 및 230)에 관한 아래의 논의에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.
- [0056] 도 4에 도시된 바와 같이, 플로우차트(400)는 단계(402)를 포함한다. 단계(402)에서, 상태 머신을 실행하여, 모바일 디바이스가 구성 정보에 따라 적합하게 구성되었는지의 여부를 결정한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 상태 머신 실행 엔진(236)은 컴퓨팅 디바이스(244)의 구성을 수정하지 않고 상태 머신(118)을 실행하여, 컴퓨팅 디바이스(244)가 올바른 구성 상태에 있는지를 평가한다. 특히, 본원에 기술된 실시예에 따르면, 상태 머신은 "구성" 모드 및 "분석" 모드의 두 개의 모드 중 하나로 실행될 수 있다. 구성 모드는 (위에 기술된 바와 같은) 구성 정보에 기반하여 컴퓨팅 디바이스를 구성한다. 분석 모드는 컴퓨팅 디바이스에 대한 수정없이 컴퓨팅 디바이스의 평가를 수행하여 컴퓨팅 디바이스가 실제로 올바른 구성 상태에 있는지를 확인한다. 예를 들어, 상태 머신은 컴퓨팅 디바이스의 (특정 하드웨어/소프트웨어/펌웨어 구성과 관련된) 하나 이상의 현재 설정 및/또는 다른 정보를 검색(예컨대, 운영 체제 버전 번호, 다른 애플리케이션의 버전 번호, 동적 링크 라이브러리(Dynamic-Link Library)(DLL) 파일과 같은 파일에서의 엔트리 등을 결정하고; 하나 이상의 파일의 존재 여부를 결정하는 등)하도록 구성될 수 있다. 상태 머신은 검색된 정보를 컴퓨팅 디바이스에 대해 상태 머신에 의해 유지되거나 액세스 가능한 올바른 구성 정보와 비교할 수 있다. 모든 정보가 매칭되면, 상태 머신은 컴퓨팅 디바이스가 특정 구성을 준수하고 있음을 나타낼 수 있다. 모든 정보가 매칭되지 않으면, 상태 머신은 컴퓨팅 디바이스가 준수하지 않음을 나타낼 수 있다.
- [0057] 전술한 바와 같이, 상태 머신에 의해 수행된 특정 구성의 결과는 준수 엔진으로 송신될 수 있다. 예를 들어, 도 5는 일 실시예에 따라, 구성 결과를 저장하고 송신하기 위한 방법의 플로우차트(500)를 도시한 것이다. 도 5는 도 1, 도 2a 및 도 2b를 계속하여 참조하여 기술된다. 그러나, 다른 구조 및 동작 실시예는 플로우차트(500) 및 도 1, 도 2a, 및 도 2b의 시스템(100, 200, 및 230)에 관한 아래의 논의에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.
- [0058] 도 5에 도시된 바와 같이, 플로우차트(500)는 단계(502)에서 시작된다. 단계(502)에서, 상태 머신을 실행함으로써 생성된 구성 결과를 구성 정보 데이터 저장소에 저장한다. 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 상태 머신 실행 엔진(236)은 구성 정보 데이터 저장소(222)에 구성 결과(226)를 저장한다. 실시예에서, 구성 결과(226)는 하나 이상의 표시를 포함할 수 있다. 예를 들어, 구성 결과(226)는 구성 설정(들)이 오케스트레이터 엔진(208)에 의해 성공적으로 설치되었고, 설치가 진행 중이거나, 오류가 발생했으며, 구성 설정(들)이 성

공적으로 설치되지 않았다는 표시를 제공할 수 있다. 오류가 발생한 경우, 구성 결과(226)는, 예를 들어, (에러 코드 및/또는 설명에 의한) 하나 이상의 에러, 이벤트 추적(예컨대, 디버깅에 사용될 수 있는 상태 머신 실행에 대한 정보의 로깅), 및/또는 상태 머신에 의해 생성된 기타 진단 데이터를 나열함으로써, 오류의 표시를 포함할 수 있다.

[0059] 단계들(504 및 506)에서, 구성 정보 데이터 저장소로부터 구성 결과를 획득하고(단계(504)), 이 구성 결과를 서버에 송신한다(단계(506)). 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 준수 구성 정보 핸들러(228)는 구성 정보 데이터 저장소(222)로부터 구성 결과(226)를 획득하고, 구성 결과(226)를 준수 엔진(122)에 제공하여 서버(102)에 의해 분석되도록 한다. 구성 결과(226)가 에러를 포함하는 경우, 서버(102)는 오케스트레이터 엔진(208)에게 컴퓨팅 디바이스의 상태를 상태 머신의 실행 전의 상태로 롤백하도록 지시하는 것을 포함하여 모바일 디바이스에 대한 임의의 잘못된 변경이 반전될 수 있도록 준수 구성 정보 핸들러(228)에 이전 버전의 확인 정보(112)를 제공할 수 있다. 예를 들어, 상태 머신을 실행하기 전에 또는 상태 머신을 실행하는 동안, 컴퓨팅 디바이스의 현재 상태가 저장될 수 있고(예컨대, 상태 머신의 실행 전에 존재했던 설정 버전, 영향을 받은 파일 등이 저장될 수 있고), 컴퓨팅 디바이스는 컴퓨팅 디바이스를 저장된 상태 정보로 되돌림으로써 저장된 상태로 재구성될 수 있다. 대안적으로, 컴퓨팅 디바이스는, 오류 상태에 대한 상세한 분석이 수행될 수 있도록, 이전의 마지막으로 알려진 양호한 상태로 롤백하지 않도록 지시받을 수 있다. 예를 들어, "개발자 모드"(또는 다른 라벨)로 지칭되는 다른 모드가 정의될 수 있으며, 이 모드에서 컴퓨팅 디바이스는 에러 발생시 마지막으로 알려진 양호한 상태로 롤백하지 않는다.

[0060] 또한, 상태 머신의 실행 중에 에러가 발생하면, 준수 에이전트는 이전 상태로 롤백하기보다는 에러를 정정하려고 시도할 수 있다는 것에 주목해야 한다. 예를 들어, 도 6은 예시적인 실시예에 따라, 구성 정보의 에러의 정정을 요청하기 위한 방법의 플로우차트(600)를 도시한 것이다. 도 6은 도 1, 도 2a 및 도 2b를 계속하여 참조하여 기술된다. 그러나, 다른 구조 및 동작 실시예는 플로우차트(600) 및 도 1, 도 2a, 및 도 2b의 시스템(100, 200, 및 230)에 관한 아래의 논의에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.

[0061] 플로우차트(600)는 단계(602)에서 시작된다. 단계(602)에서, 구성 정보를 파싱한다. 예를 들어, 오케스트레이터 엔진(208)은 수신된 에러 코드/에러 설명에 대응하는 하나 이상의 구성 설정에 대한 구성 정보(112)를 파싱하도록 구성될 수 있다.

[0062] 단계(604)에서, 구성 정보의 에러를 결정한다. 일 실시예에서, 파싱 동안, 오케스트레이터 엔진(208)은 수신된 에러 코드/에러 설명에 대응하는 하나 이상의 구성 설정을 결정할 수 있다.

[0063] 단계(606)에서 서버로부터 에러의 정정을 요청한다. 일 실시예에서, 준수 에이전트(240)는 에러 정보(예컨대, 에러 코드, 에러 설명, 이벤트 추적 등)를 준수 엔진(122)에 송신하여 에러의 정정을 요청할 수 있다.

[0064] 예를 들어, 도 1, 도 2a, 및 도 2b를 계속 참조하면, 구성 정보(112)가 XML 파일로서 표현되는 위에서 논의된 예에서, 구성 정보 에러 체커(210)는 XML 파일을 파싱 및 검증하고, URI를 레지스트리 위치에 저장하고, 그리고 임의의 에러가 검출되면, 에러를 서버(102)에 동기적으로 보고하고 서버(102)에 에러를 즉각적으로 수정하도록 요청할 수 있다.

[0065] 에러의 정정 요청에 응답하여, 및/또는 임의의 다른 시간에, 준수 엔진(122)은 업데이트된 구성 정보를 준수 에이전트에 제공할 수 있다는 것에 주목해야 한다. 예를 들어, 도 7은 예시적인 실시예에 따라, 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하기 위한 방법의 플로우차트(700)를 도시한 것이다. 도 7은 도 1, 도 2a 및 도 2b를 계속하여 참조하여 기술된다. 그러나, 다른 구조 및 동작 실시예는 플로우차트(700) 및 도 1, 도 2a, 및 도 2b의 시스템(100, 200, 및 230)에 관한 아래의 논의에 기반하여 관련 기술 분야(들)의 기술자에게 명백할 것이다.

[0066] 도 7에 도시된 바와 같이, 플로우차트(700)는 단계(702)에서 시작된다. 단계(702)에서, 구성 정보를 구성 정보 데이터 저장소에 저장한다. 예를 들어, 도 2a 및 도 2b를 계속 참조하면, 구성 정보(112)는 구성 정보 데이터 저장소(222)에 저장된다.

[0067] 단계(704)에서, 저장된 구성 정보를 서버에 송신한다. 실시예에서, 준수 에이전트(240)는 (전술한 바와 같은) 에러로 인한 것을 포함하여 업데이트를 체크하기 위한 임의의 이유, 및 임의의 다른 이유로 서버(102)에서의 준수 엔진(122)에 구성 정보(112)를 송신할 수 있다. 예를 들어, 준수 구성 정보 핸들러(228)는 구성 정보 데이터 저장소(222)에 저장된 구성 정보(112)의 상태(예컨대, 만료 또는 부실)를 검증할 수 있다. 핸들러(228)가 구성 정보(112)가 에러를 포함하거나, 만료된 것이거나, 부실하다는 것 등을 결정하는 경우, 핸들러(228)는 구성 정

보(112)(또는 그 일부)를 서버(102)에 송신할 수 있다.

[0068] 단계(706)에서, 구성 정보를 서버에 송신하는 것에 응답하여, 서버로부터 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신한다. 서버(102)에 의해 결정된 저장된 구성 정보(112)의 상태에 기반하여, 서버(102)는 업데이트된 구성 정보 및/또는 추가 구성 정보를 준수 구성 정보 핸들러(228)에 제공할 수 있다. 따라서, 준수 구성 정보 핸들러(228)는 업데이트된 구성 정보 및/또는 추가 구성 정보로 구성 정보 데이터 저장소(222)를 업데이트할 수 있다.

[0069] 따라서, 실시예는 구성 설정을 대량의 디바이스에 푸시하고, 구성 설정을 수신하고 구현한 것에 기반한 요구 사항으로 그러한 디바이스의 준수 추적을 유지하는 효율성을 개선한다.

[0070] 실시예는 클라이언트 디바이스 상에 보다 많은 응답성을 제공하여 그 클라이언트 디바이스가 원하는 상태가 되게 함으로써 클라이언트/서버 대기 시간(데이터 전송의 명령 후 데이터 전송이 시작되기 전의 지연)을 감소시킨다. 각 클라이언트 디바이스는 서버로부터의 구성 설정 구현 요청을 수락하고 즉시 서버로 돌아간다. 요청은 클라이언트 디바이스에 의해 비동기적으로 핸들링된다. 클라이언트 디바이스가 요청을 완료하면, 클라이언트 디바이스는 클라이언트 디바이스에 의한 구성 설정 구현의 성공 또는 실패를 나타내는 경고를 서버에 전송하는 세션을 시작할 것이다. 실시예는 또한 서버가 클라이언트 디바이스의 상태를 지속적으로 모니터링하고 원하는 디바이스 상태를 단편적으로 유지할 필요성을 제거함으로써 클라이언트 및 서버 디바이스들 간의 네트워크 활동을 감소시킨다. 실시예에서, 서버는 각 클라이언트에게 (구성 설정의 형태로) 원하는 상태를 제공하고, 클라이언트가 그 설정을 유지(및 그렇게 할 수 없는 경우 서버에 의한 프롬프트없이도 서버에 통지)할 것을 기대한다. 실시예는 또한 서비스 제공자(예컨대, 소프트웨어 및/또는 하드웨어 제공자)가 그들 자신의 제품에 대한 프로세스를 관리해야 하는 대신, 설정으로 클라이언트 디바이스를 구성하도록, 서버에 의해 개시되는 균일한 방식을 제공한다.

[0071] III. 예시적인 모바일 및 고정 디바이스 실시예

[0072] 도 1 내지 도 7을 참조하여 기술된 디바이스 준수 관리 실시예를 포함하여, 위에서 기술된 시스템 및 방법은 하드웨어로 구현되거나, 또는 소프트웨어 및/또는 펌웨어 중 하나 또는 둘 모두와 결합된 하드웨어로 구현될 수 있다. 예를 들어, 준수 엔진(112), 준수 에이전트(120), 오케스트레이터 엔진(208), 구성 정보 에러 체커(210), 준수 에이전트(220), 준수 구성 정보 핸들러(228), 설정 구현 트리거(232), 상태 머신 선택기(234), 상태 머신 실행 엔진(236), 준수 에이전트(240) 및/또는 본원에 기술된 컴포넌트들의 각각, 및 플로우차트들(300, 400, 500, 600, 및 700)은 각각 하나 이상의 프로세서에서 실행되고 컴퓨터 판독 가능 저장 매체에 저장되도록 구성된 컴퓨터 프로그램 코드/인스트럭션으로서 구현될 수 있다. 대안적으로, 준수 엔진(112), 준수 에이전트(120), 오케스트레이터 엔진(208), 구성 정보 에러 체커(210), 준수 에이전트(220), 준수 구성 정보 핸들러(228), 설정 구현 트리거(232), 상태 머신 선택기(234), 상태 머신 실행 엔진(236), 준수 에이전트(240) 및/또는 본원에 기술된 컴포넌트들의 각각, 및 플로우차트들(300, 400, 500, 600, 및 700)은 하드웨어 로직/전기 회로부로서 구현될 수 있다. 일 실시예에서, 준수 엔진(112), 준수 에이전트(120), 오케스트레이터 엔진(208), 구성 정보 에러 체커(210), 준수 에이전트(220), 준수 구성 정보 핸들러(228), 설정 구현 트리거(232), 상태 머신 선택기(234), 상태 머신 실행 엔진(236), 준수 에이전트(240) 및/또는 본원에 기술된 컴포넌트들의 각각, 및 플로우차트들(300, 400, 500, 600, 및 700)은 하나 이상의 시스템 온 칩(system on chip)(SoC)에서 구현될 수 있다. SoC는, 프로세서(예컨대, 중앙 처리 유닛(CPU), 마이크로컨트롤러, 마이크로프로세서, 디지털 신호 프로세서(DSP) 등), 메모리, 하나 이상의 통신 인터페이스 중 하나 이상을 포함하는 집적 회로 칩, 및/또는 추가 회로를 포함할 수 있고, 수신된 프로그램 코드를 선택적으로 실행할 수 있고 및/또는 기능을 수행하기 위해 내장된 펌웨어를 포함할 수 있다.

[0073] 더욱이, 서버(102), 컴퓨팅 디바이스(들)(104), 컴퓨팅 디바이스(204), 및 컴퓨팅 디바이스(244)는 각각 다양한 디바이스 타입으로 구현될 수 있다. 예를 들어, 도 8은, 일반적으로 컴포넌트들(802)로 도시된 다양한 선택적인 하드웨어 및 소프트웨어 컴포넌트를 포함하는 예시적인 모바일 디바이스(800)의 블럭 다이어그램을 도시한 것이다. 준수 엔진(112), 준수 에이전트(120), 오케스트레이터 엔진(208), 구성 정보 에러 체커(210), 준수 에이전트(220), 준수 구성 정보 핸들러(228), 설정 구현 트리거(232), 상태 머신 선택기(234), 상태 머신 실행 엔진(236), 준수 에이전트(240)의 특징들/요소들의 임의의 개수 및 조합, 및/또는 플로우차트들(300, 400, 500, 600 및 700)은 모바일 디바이스 실시예에 포함된 컴포넌트들(802)로서뿐만 아니라, 관련 기술 분야(들)의 기술자에게 알려질 추가 및/또는 대안적인 특징들/요소들로서 구현될 수 있다. 컴포넌트들(802) 중 임의의 컴포넌트는 컴포넌트들(802) 중 임의의 다른 컴포넌트와 통신할 수 있지만, 설명의 편의를 위해, 모든 커넥션들이 도시된

것은 아니라는 것에 주목해야 한다. 모바일 디바이스(800)는 본원의 다른 곳에서 기술되거나 언급되거나 또는 알려진 다양한 모바일 디바이스들(예컨대, 셀폰, 스마트폰, 핸드헬드 컴퓨터, PDA (Personal Digital Assistant) 등) 중 임의의 하나일 수 있으며, 셀룰러 또는 위성 네트워크와 같은 하나 이상의 통신 네트워크(804)를 통해 하나 이상의 모바일 디바이스와의, 또는 근거리 또는 광역 네트워크와의 무선 양방향 통신을 가능하게 할 수 있다.

[0074] 도시된 모바일 디바이스(800)는 신호 코딩, 이미지 처리, 데이터 처리, 입력/출력 처리, 전력 제어, 및/또는 다른 기능과 같은 태스크를 수행하기 위해 프로세서 회로(810)로 지칭되는 컨트롤러 또는 프로세서를 포함할 수 있다. 프로세서 회로(810)는 중앙 처리 유닛(CPU), 마이크로컨트롤러, 마이크로프로세서, 및/또는 다른 물리적 하드웨어 프로세서 회로로서, 하나 이상의 물리적 하드웨어 전기 회로 디바이스 요소 및/또는 집적 회로 디바이스(반도체 재료 칩 또는 다이)에서 구현되는 전기 및/또는 광학 회로이다. 프로세서 회로(810)는 컴퓨터 관독 가능 매체에 저장된 프로그램 코드, 예를 들어, 하나 이상의 애플리케이션(814)의 프로그램 코드, 운영 체제(812), 메모리(820)에 저장된 임의의 프로그램 코드 등을 실행할 수 있다. 운영 체제(812)는 컴포넌트들(802)의 할당 및 사용을 제어하고, 하나 이상의 애플리케이션 프로그램(814)(일명 애플리케이션, "앱" 등)에 대한 지원을 제어할 수 있다. 애플리케이션 프로그램(814)은 일반적인 모바일 컴퓨팅 애플리케이션(예컨대, 이메일 애플리케이션, 캘린더, 연락처 관리자, 웹 브라우저, 메시징 애플리케이션), 및 임의의 다른 컴퓨팅 애플리케이션(예컨대, 워드 프로세싱 애플리케이션, 맵핑 애플리케이션, 미디어 플레이어 애플리케이션)을 포함할 수 있다.

[0075] 도시된 바와 같이, 모바일 디바이스(800)는 메모리(820)를 포함할 수 있다. 메모리(820)는 비 착탈식 메모리(822) 및/또는 착탈식 메모리(824)를 포함할 수 있다. 비 착탈식 메모리(822)는 RAM, ROM, 플래시 메모리, 하드 디스크, 또는 다른 잘 알려진 메모리 저장 기술을 포함할 수 있다. 착탈식 메모리(824)는 GSM 통신 시스템에서 잘 알려진 플래시 메모리 또는 SIM (Subscriber Identity Module) 카드, 또는 "스마트 카드"와 같은 다른 잘 알려진 메모리 저장 기술을 포함할 수 있다. 메모리(820)는 운영 체제(812) 및 애플리케이션(814)을 실행하기 위한 데이터 및/또는 코드를 저장하는 데 사용될 수 있다. 예시적인 데이터는 웹 페이지, 텍스트, 이미지, 사운드 파일, 비디오 데이터, 또는 하나 이상의 유선 또는 무선 네트워크를 통해 하나 이상의 네트워크 서버 또는 다른 디바이스로 전송 및/또는 수신되는 다른 데이터 세트를 포함할 수 있다. 메모리(820)는 IMSI (International Mobile Subscriber Identity)와 같은 가입자 식별자, 및 IMEI (International Mobile Equipment Identifier)와 같은 장비 식별자를 저장하는 데 사용될 수 있다. 이러한 식별자는 사용자와 장비를 식별하기 위해 네트워크 서버로 송신될 수 있다.

[0076] 다수의 프로그램이 메모리(820)에 저장될 수 있다. 이들 프로그램은 운영 체제(812), 하나 이상의 애플리케이션 프로그램(814), 및 다른 프로그램 모듈 및 프로그램 데이터를 포함한다. 이러한 애플리케이션 프로그램 또는 프로그램 모듈의 예는, 예를 들어, 도 1 내지 도 7을 참조하여 기술된 디바이스 준수 관리 실시예를 포함하여, 위에서 기술된 시스템을 구현하기 위한 컴퓨터 프로그램 로직(예컨대, 컴퓨터 프로그램 코드 또는 인스트럭션)을 포함할 수 있다.

[0077] 모바일 디바이스(800)는, 터치 스크린(832), 마이크론(834), 카메라(836), 물리적 키보드(838) 및/또는 트랙볼(840)과 같은 하나 이상의 입력 디바이스(830), 및 스피커(852) 및 디스플레이(854)와 같은 하나 이상의 출력 디바이스(850)를 지원할 수 있다.

[0078] 다른 가능한 출력 디바이스(도시되지 않음)는 압전 또는 다른 햅틱 출력 디바이스를 포함할 수 있다. 일부 디바이스는 하나 초과 입력/출력 기능을 제공할 수 있다. 예를 들어, 터치 스크린(832) 및 디스플레이(854)는 단일 입력/출력 디바이스에서 결합될 수 있다. 입력 디바이스(830)는 NUI (Natural User Interface)를 포함할 수 있다.

[0079] 무선 모뎀(들)(860)은 안테나(들)(도시되지 않음)에 연결될 수 있고 프로세서 회로(810)와 외부 디바이스 사이의 양방향 통신을 지원할 수 있으며, 이는 본 기술 분야에서 잘 이해된다. 모뎀(들)(860)은 일반적으로 도시되고, 모바일 통신 네트워크(804)와 통신하기 위한 셀룰러 모뎀(866) 및/또는 다른 무선 기반 모뎀(예컨대, 블루투스(864) 및/또는 Wi-Fi(862))을 포함할 수 있다. 셀룰러 모뎀(866)은 GSM, 3G, 4G, 5G 등과 같은 임의의 적합한 통신 표준 또는 기술에 따라 전화 통화를 가능하게 하도록 (그리고 선택적으로 데이터를 송신하도록) 구성될 수 있다. 무선 모뎀(들)(860) 중 적어도 하나는 일반적으로 셀룰러 네트워크들 사이에서, 또는 모바일 디바이스와 공중 전화 교환망(public switched telephone network)(PSTN) 사이에서, 단일 셀룰러 네트워크 내의 데이터 및 음성 통신을 위한 GSM 네트워크와 같은 하나 이상의 셀룰러 네트워크와 통신하도록 구성된다.

[0080] 모바일 디바이스(800)는 적어도 하나의 입력/출력 포트(880), 전력 공급 장치(882), GPS (Global Positioning

System) 수신기와 같은 위성 내비게이션 시스템 수신기(884), 가속도계(886), 및/또는 USB 포트, IEEE 1394 (FireWire) 포트, 및/또는 RS-232 포트가 될 수 있는 물리적 커넥터(890)를 더 포함할 수 있다. 도시된 컴포넌트들(802)은, 본 기술 분야의 기술자에 의해 인식되는 바와 같이, 임의의 컴포넌트들이 존재하지 않을 수 있고 다른 컴포넌트들이 추가적으로 존재할 수 있기 때문에, 필수적이지 않거나, 또는 모든 것을 포함하지는 않는다.

[0081] 또한, 도 9는, 컴퓨팅 디바이스(104), 컴퓨팅 디바이스(204), 컴퓨팅 디바이스(244), 서버(102)를 포함하는 실시예가 구현될 수 있는 컴퓨팅 디바이스(900)의 예시적인 구현예를 도시한 것이다. 본원에 제공된 컴퓨팅 디바이스(900)의 설명은 예시의 목적으로 제공되며, 제한하려는 의도는 아니다. 실시예는 관련 기술 분야(들)의 기술자에게 알려진 바와 같이 다른 타입의 컴퓨터 시스템에서 구현될 수 있다.

[0082] 도 9에 도시된 바와 같이, 컴퓨팅 디바이스(900)는 프로세서 회로(902)로 지칭되는 하나 이상의 프로세서, 시스템 메모리(904), 및 시스템 메모리(904)를 포함하는 다양한 시스템 컴포넌트들을 프로세서 회로(902)에 연결하는 버스(906)를 포함한다. 프로세서 회로(902)는 중앙 처리 유닛(CPU), 마이크로컨트롤러, 마이크로프로세서, 및/또는 다른 물리적 하드웨어 프로세서 회로로서, 하나 이상의 물리적 하드웨어 전기 회로 디바이스 요소 및/또는 집적 회로 디바이스(반도체 재료 칩 또는 다이)에서 구현되는 전기 및/또는 광학 회로이다. 프로세서 회로(902)는 컴퓨터 관독 가능 매체에 저장된 프로그램 코드, 예를 들어, 운영 체제(930)의 프로그램 코드, 애플리케이션 프로그램(932), 다른 프로그램(934) 등을 실행할 수 있다. 버스(906)는 메모리 버스 또는 메모리 컨트롤러, 주변 버스, 가속 그래픽 포트, 및 다양한 버스 아키텍처 중 임의의 것을 사용하는 프로세서 또는 로컬 버스를 포함하는 여러 타입의 버스 구조들 중 임의의 하나 이상을 나타낸다. 시스템 메모리(904)는 관독 전용 메모리(ROM)(908) 및 랜덤 액세스 메모리(RAM)(910)를 포함한다. 기본 입력/출력 시스템(912)(BIOS)은 ROM(908)에 저장된다.

[0083] 컴퓨팅 디바이스(900)는 또한 다음의 드라이브들: 즉 하드 디스크로부터 관독 및 기입을 위한 하드 디스크 드라이브(914), 착탈식 자기 디스크(918)로부터 관독 및 기입을 위한 자기 디스크 드라이브(916), 및 CD ROM, DVD ROM, 또는 다른 광학 매체와 같은 착탈식 광학 디스크(922)로부터 관독 및 기입을 위한 광학 디스크 드라이브(920) 중 하나 이상을 갖는다. 하드 디스크 드라이브(914), 자기 디스크 드라이브(916), 및 광학 디스크 드라이브(920)는 각각 하드 디스크 드라이브 인터페이스(924), 자기 디스크 드라이브 인터페이스(926), 및 광학 드라이브 인터페이스(928)에 의해 버스(906)에 연결된다. 드라이브 및 관련 컴퓨터 관독 가능 매체는 컴퓨터 관독 가능 인스트럭션, 데이터 구조, 프로그램 모듈 및 다른 컴퓨터용 데이터의 비 휘발성 저장을 제공한다. 하드 디스크, 착탈식 자기 디스크 및 착탈식 광학 디스크가 기술되지만, 다른 타입의 하드웨어 기반 컴퓨터 관독 가능 저장 매체, 예를 들어, 플래시 메모리 카드, 디지털 비디오 디스크, RAM, ROM, 및 다른 하드웨어 저장 매체가 데이터를 저장하는 데 사용될 수 있다.

[0084] 다수의 프로그램 모듈이 하드 디스크, 자기 디스크, 광학 디스크, ROM, 또는 RAM에 저장될 수 있다. 이들 프로그램은 운영 체제(930), 하나 이상의 애플리케이션 프로그램(932), 다른 프로그램(934), 및 프로그램 데이터(936)를 포함한다. 애플리케이션 프로그램(932) 또는 다른 프로그램(934)은, 예를 들어, 도 1 내지 도 7을 참조하여 기술된 디바이스 준수 관리 실시예를 포함하여, 위에서 기술된 시스템을 구현하기 위한 컴퓨터 프로그램 로직(예컨대, 컴퓨터 프로그램 코드 또는 인스트럭션)을 포함할 수 있다.

[0085] 사용자는 키보드(938) 및 포인팅 디바이스(940)와 같은 입력 디바이스를 통해 컴퓨팅 디바이스(900)에 커맨드 및 정보를 입력할 수 있다. 다른 입력 디바이스(도시되지 않음)는 마이크로폰, 조이스틱, 게임 패드, 위성 접시, 스캐너, 터치 스크린 및/또는 터치 패드, 음성 입력을 수신하는 음성 인식 시스템, 또는 제스처 입력을 수신하는 제스처 인식 시스템 등을 포함할 수 있다. 이들 및 다른 입력 디바이스는 종종 버스(906)에 연결된 직렬 포트 인터페이스(942)를 통해 프로세서 회로(902)에 접속되지만, 병렬 포트, 게임 포트, 또는 범용 직렬 버스(USB)와 같은 다른 인터페이스에 의해 접속될 수 있다.

[0086] 디스플레이 스크린(944)은 또한 비디오 어댑터(846)와 같은 인터페이스를 통해 버스(906)에 접속된다. 디스플레이 스크린(844)은 컴퓨팅 디바이스(900)의 외부에 있거나 컴퓨팅 디바이스(900)에 통합될 수 있다. 디스플레이 스크린(944)은 정보를 디스플레이할 수 있을 뿐만 아니라, 사용자 커맨드 및/또는 다른 정보를 (예컨대, 터치, 손가락 제스처, 가상 키보드 등에 의해) 수신하기 위한 사용자 인터페이스일 수 있다. 디스플레이 스크린(944)에 더하여, 컴퓨팅 디바이스(900)는 스피커 및 프린터와 같은 다른 주변 출력 디바이스들(도시되지 않음)을 포함할 수 있다.

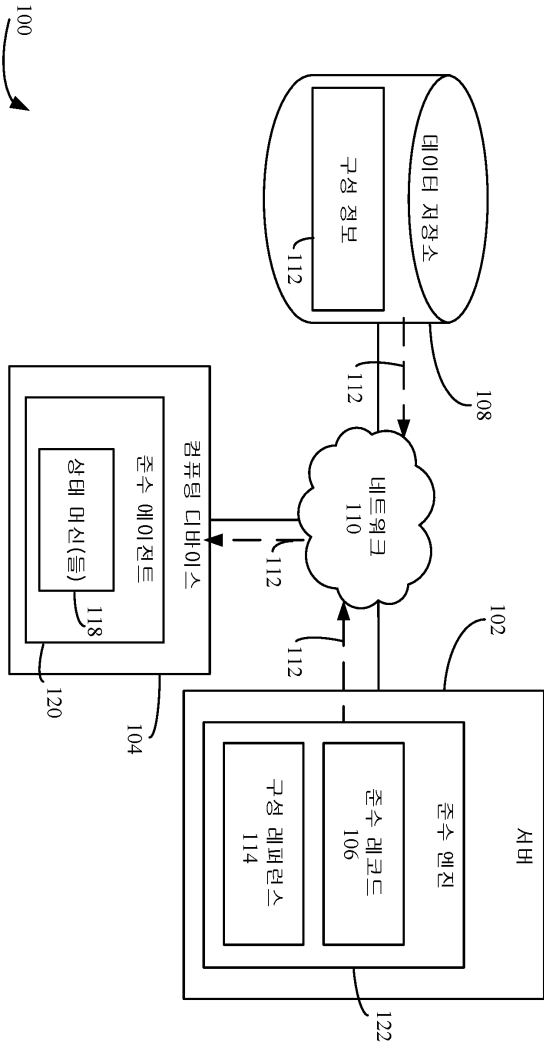
[0087] 컴퓨팅 디바이스(900)는, 어댑터 또는 네트워크 인터페이스(950), 모뎀(952), 또는 네트워크를 통한 통신을 수립하기 위한 다른 수단을 통해 네트워크(948)(예컨대, 인터넷)에 접속된다. 내부 또는 외부일 수 있는 모뎀

(952)은 도 9에 도시된 바와 같이 직렬 포트 인터페이스(942)를 통해 버스(906)에 접속될 수 있거나, 병렬 인터페이스를 포함하는 다른 인터페이스 타입을 사용하여 버스(906)에 접속될 수 있다.

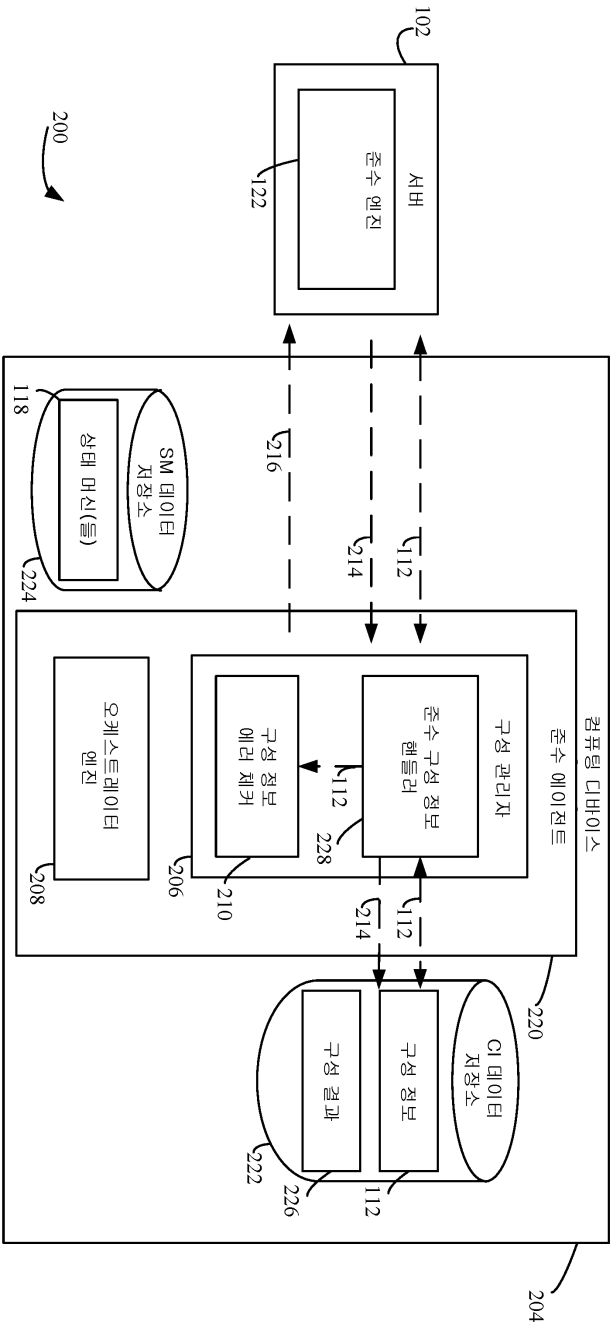
- [0088] 본원에서 사용된 바와 같이, "컴퓨터 프로그램 매체", "컴퓨터 판독 가능 매체" 및 "컴퓨터 판독 가능 저장 매체"라는 용어는 일반적으로, 하드 디스크 드라이브(914)와 연관된 하드 디스크와 같은 물리적 하드웨어 매체, 착탈식 자기 디스크(918), 착탈식 광학 디스크(922), 다른 물리적 하드웨어 매체, 예를 들어, RAM, ROM, 플래시 메모리 카드, 디지털 비디오 디스크, zip 디스크, MEM, 나노기술 기반 저장 디바이스, 및 추가 타입의 물리적/유형적 하드웨어 저장 매체(도 9의 시스템 메모리(904)를 포함)를 지칭하는 데 사용된다. 이러한 컴퓨터 판독 가능 저장 매체는 통신 매체와 구별되고 이와 중복되지 않는다(통신 매체를 포함하지 않는다). 통신 매체는 전형적으로 컴퓨터 판독 가능 인스트럭션, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호에서의 다른 데이터를 구현한다. "변조된 데이터 신호"라는 용어는 신호의 정보를 인코딩하는 것과 같은 방식으로 설정되거나 변경된 하나 이상의 특성을 갖는 신호를 의미한다. 제한이 아닌 예로서, 통신 매체는 유선 매체 뿐만 아니라, 음향, RF, 적외선과 같은 무선 매체 및 다른 무선 매체를 포함한다. 실시에는 또한 그러한 통신 매체에 관한 것이다.
- [0089] 전술한 바와 같이, 컴퓨터 프로그램 및 모듈(애플리케이션 프로그램(932) 및 다른 프로그램(934)을 포함)은 하드 디스크, 자기 디스크, 광학 디스크, ROM, RAM, 또는 다른 하드웨어 저장 매체에 저장될 수 있다. 이러한 컴퓨터 프로그램은 또한 네트워크 인터페이스(950), 직렬 포트 인터페이스(952), 또는 임의의 다른 인터페이스 타입을 통해 수신될 수 있다. 이러한 컴퓨터 프로그램은, 애플리케이션에 의해 실행되거나 로딩될 때, 컴퓨팅 디바이스(900)로 하여금 본원에서 논의된 실시예들의 특징들을 구현할 수 있게 한다. 따라서, 그러한 컴퓨터 프로그램은 컴퓨팅 디바이스(900)의 컨트롤러를 나타낸다.
- [0090] 실시에는 또한 임의의 컴퓨터 판독 가능 매체에 저장된 컴퓨터 코드 또는 인스트럭션을 포함하는 컴퓨터 프로그램 제품에 관한 것이다. 이러한 컴퓨터 프로그램 제품은 하드 디스크 드라이브, 광학 디스크 드라이브, 메모리 디바이스 패키지, 휴대용 메모리 스틱, 메모리 카드, 및 다른 타입의 물리적 저장 하드웨어를 포함한다.
- [0091] IV. 추가 예시적인 실시예
- [0092] 일 실시예에서, 모바일 디바이스는: 적어도 하나의 프로세서 회로; 및 상기 적어도 하나의 프로세서 회로에 의해 실행되도록 구성된 프로그램 코드를 저장하는 적어도 하나의 메모리를 포함하고, 상기 프로그램 코드는: 서버로부터, 상기 모바일 디바이스가 구성 시나리오를 준수하도록 하기 위해 상기 모바일 디바이스에 의해 구현하기 위한 준수 설정을 기술하는 구성 정보를 수신하고, 그리고 상기 구성 정보를 저장하도록 구성되는 구성 관리자; 및 상기 모바일 디바이스에서 상기 준수 설정을 구현하기 위한 트리거를 상기 구성 관리자로부터 수신하고; 상기 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 구성 정보에 의해 표시되는 상태 머신을 식별하고; 그리고 상기 상태 머신을 실행하여, 상기 모바일 디바이스를 상기 준수 설정으로 구성하는 오케스트레이터 엔진을 포함한다.
- [0093] 전술한 모바일 디바이스의 일 실시예에서, 오케스트레이터 엔진은 상태 머신을 실행하여 모바일 디바이스가 구성 정보에 따라 적합하게 구성되었는지 여부를 결정하도록 더 구성된다.
- [0094] 전술한 모바일 디바이스의 일 실시예에서, 오케스트레이터 엔진은 상태 머신을 실행함으로써 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하도록 더 구성되며; 구성 관리자는 구성 정보 데이터 저장소로부터 구성 결과를 획득하고 구성 결과를 서버에 송신하도록 더 구성된다.
- [0095] 전술한 모바일 디바이스의 일 실시예에서, 구성 관리자는 구성 정보를 파싱하고; 구성 정보의 에러를 결정하고, 그리고 서버로부터 에러의 정정을 요청하도록 더 구성된다.
- [0096] 전술한 모바일 디바이스의 일 실시예에서, 구성 관리자는 구성 정보를 구성 정보 데이터 저장소에 저장하고; 저장된 구성 정보를 서버에 송신하고; 그리고 구성 정보를 서버에 송신하는 것에 응답하여, 서버로부터 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하도록 더 구성된다.
- [0097] 전술한 모바일 디바이스의 일 실시예에서, 상태 머신은 모바일 디바이스 상의 상태 머신 데이터 저장소에 저장된다.
- [0098] 전술한 모바일 디바이스의 일 실시예에서, 구성 정보는 준수 설정과 연관된 페이로드 및 페이로드의 다운로드 및 처리를 위한 대응하는 URI (Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 서버로부터 수신된다.

- [0099] 다른 실시예에서, 방법은: 서버로부터, 모바일 디바이스가 구성 시나리오를 준수하도록 하기 위해 상기 모바일 디바이스에 의해 구현하기 위한 준수 설정을 기술하는 구성 정보를 수신하는 단계; 상기 구성 정보를 저장하는 단계; 상기 모바일 디바이스에서 상기 준수 설정을 구현하기 위한 트리거를 수신하는 단계; 상기 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 구성 정보에 의해 표시되는 상태 머신을 식별하는 단계; 및 상기 상태 머신을 실행하여, 상기 모바일 디바이스를 상기 준수 설정으로 구성하는 단계를 포함한다.
- [0100] 일 실시예에서, 전술한 방법은 상태 머신을 실행하여 모바일 디바이스가 구성 정보에 따라 적합하게 구성되었는지 여부를 결정하는 단계를 더 포함한다.
- [0101] 일 실시예에서, 전술한 방법은 상태 머신을 실행함으로써 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하는 단계; 구성 정보 데이터 저장소로부터 구성 결과를 획득하는 단계; 및 구성 결과를 서버에 송신하는 단계를 더 포함한다.
- [0102] 일 실시예에서, 전술한 방법은 구성 정보를 파싱하는 단계; 구성 정보의 에러를 결정하는 단계; 및 서버로부터 에러의 정정을 요청하는 단계를 더 포함한다.
- [0103] 일 실시예에서, 전술한 방법은 구성 정보를 구성 정보 데이터 저장소에 저장하는 단계; 저장된 구성 정보를 서버에 송신하는 단계; 및 구성 정보를 서버에 송신하는 것에 응답하여, 서버로부터 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하는 단계를 더 포함한다.
- [0104] 전술한 방법의 일 실시예에서, 상태 머신은 모바일 디바이스 상의 상태 머신 데이터 저장소에 저장된다.
- [0105] 전술한 방법의 일 실시예에서, 구성 정보는 준수 설정과 연관된 페이로드 및 페이로드의 다운로드 및 처리를 위한 대응하는 URI (Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 서버로부터 수신된다.
- [0106] 다른 실시예에서, 컴퓨터 판독 가능 저장 매체는 레코딩된 프로그램 인스트럭션을 가지며, 이 프로그램 인스트럭션은, 모바일 디바이스의 적어도 하나의 처리 회로에 의해 실행될 때, 방법을 수행하며, 방법은: 서버로부터, 상기 모바일 디바이스가 구성 시나리오를 준수하도록 하기 위해 상기 모바일 디바이스에 의해 구현하기 위한 준수 설정을 기술하는 구성 정보를 수신하는 단계; 상기 구성 정보를 저장하는 단계; 상기 모바일 디바이스에서 상기 준수 설정을 구현하기 위한 트리거를 수신하는 단계; 상기 준수 설정을 구현하기 위한 구성 프로세스를 기술하는 상기 구성 정보에 의해 표시되는 상태 머신을 식별하는 단계; 및 상기 상태 머신을 실행하여, 상기 모바일 디바이스를 상기 준수 설정으로 구성하는 단계를 포함한다.
- [0107] 일 실시예에서, 전술한 방법은 상태 머신을 실행하여 모바일 디바이스가 구성 정보에 따라 적합하게 구성되었는지 여부를 결정하는 단계를 포함한다.
- [0108] 전술한 방법의 일 실시예에서, 방법은 상태 머신을 실행함으로써 생성된 구성 결과를 구성 정보 데이터 저장소에 저장하는 단계; 구성 정보 데이터 저장소로부터 구성 결과를 획득하는 단계; 및 구성 결과를 서버에 송신하는 단계를 더 포함한다.
- [0109] 일 실시예에서, 전술한 방법은 구성 정보를 파싱하는 단계; 구성 정보의 에러를 결정하는 단계; 및 서버로부터 에러의 정정을 요청하는 단계를 포함한다.
- [0110] 일 실시예에서, 전술한 방법은 구성 정보를 구성 정보 데이터 저장소에 저장하는 단계; 저장된 구성 정보를 서버에 송신하는 단계; 및 구성 정보를 서버에 송신하는 것에 응답하여, 서버로부터 업데이트된 구성 정보 및/또는 추가 구성 정보를 수신하는 단계를 포함한다.
- [0111] 전술한 방법의 일 실시예에서, 구성 정보는 준수 설정과 연관된 페이로드 및 페이로드의 다운로드 및 처리를 위한 대응하는 URI (Uniform Resource Identifier)를 포함하는 확장 가능 마크업 언어(Extensible Markup Language)(XML) 파일로 서버로부터 수신된다.
- [0112] V. 결론
- [0113] 다양한 실시예들이 위에서 기술되었지만, 이들 실시예들은 제한이 아닌 예로서만 제시된 것임을 이해해야 한다. 관련 기술 분야의 기술자는 실시예들의 사상 및 범위를 벗어나지 않으면서 형태 및 세부 사항의 다양한 변경이 이루어질 수 있음을 명백히 알 수 있을 것이다. 따라서, 실시예들의 영역 및 범위는 전술한 예시적인 실시예들 중 어느 것에 의해서도 제한되어서는 안되지만, 아래의 청구항 및 그 등가물에 따라서만 정의되어야 한다.

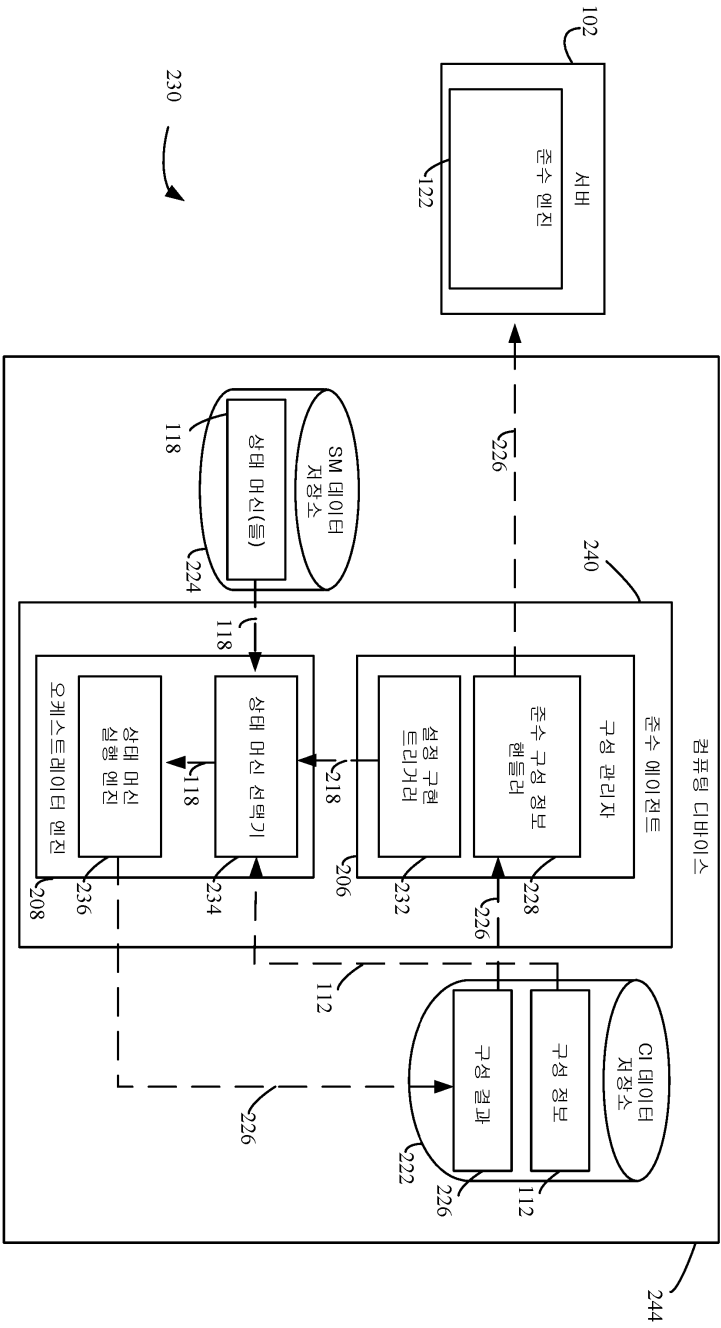
도면
도면1



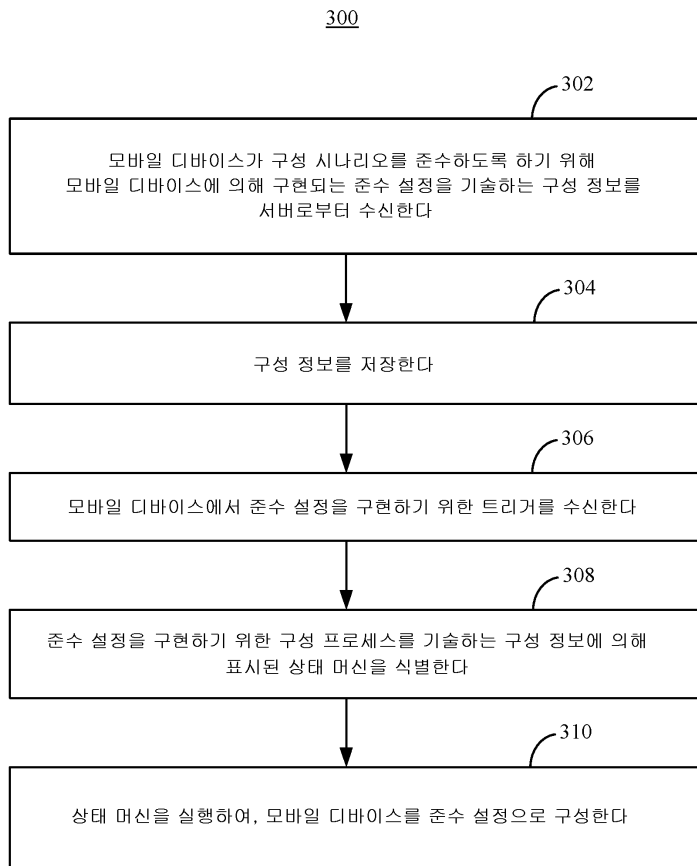
도면2a



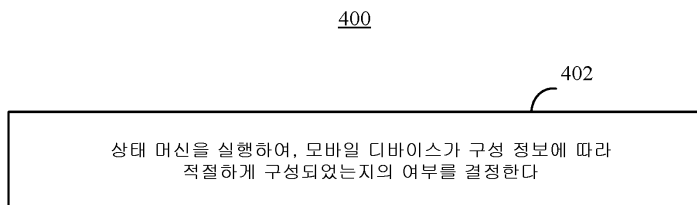
도면2b



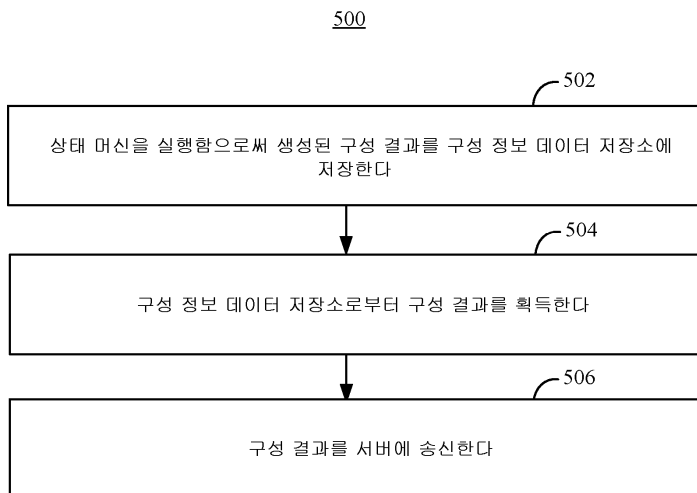
도면3



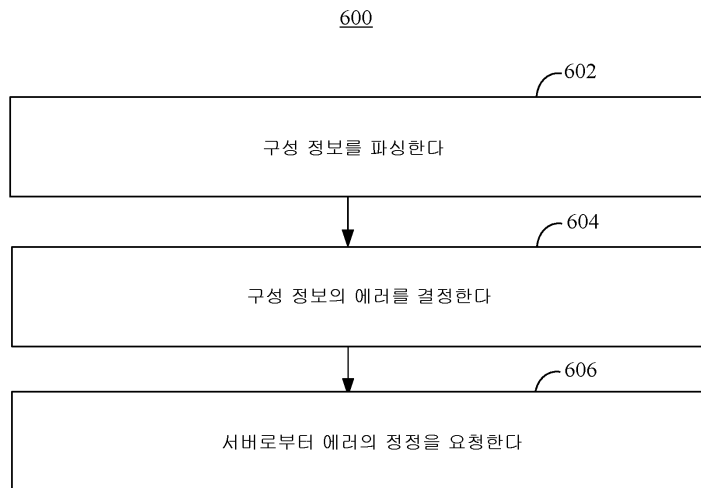
도면4



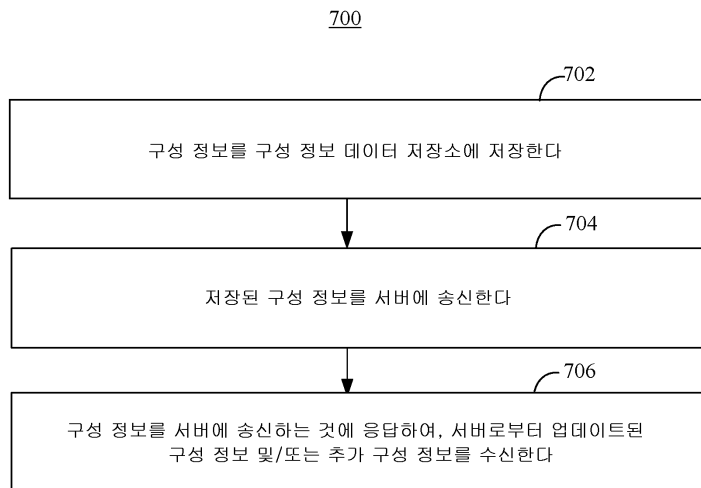
도면5



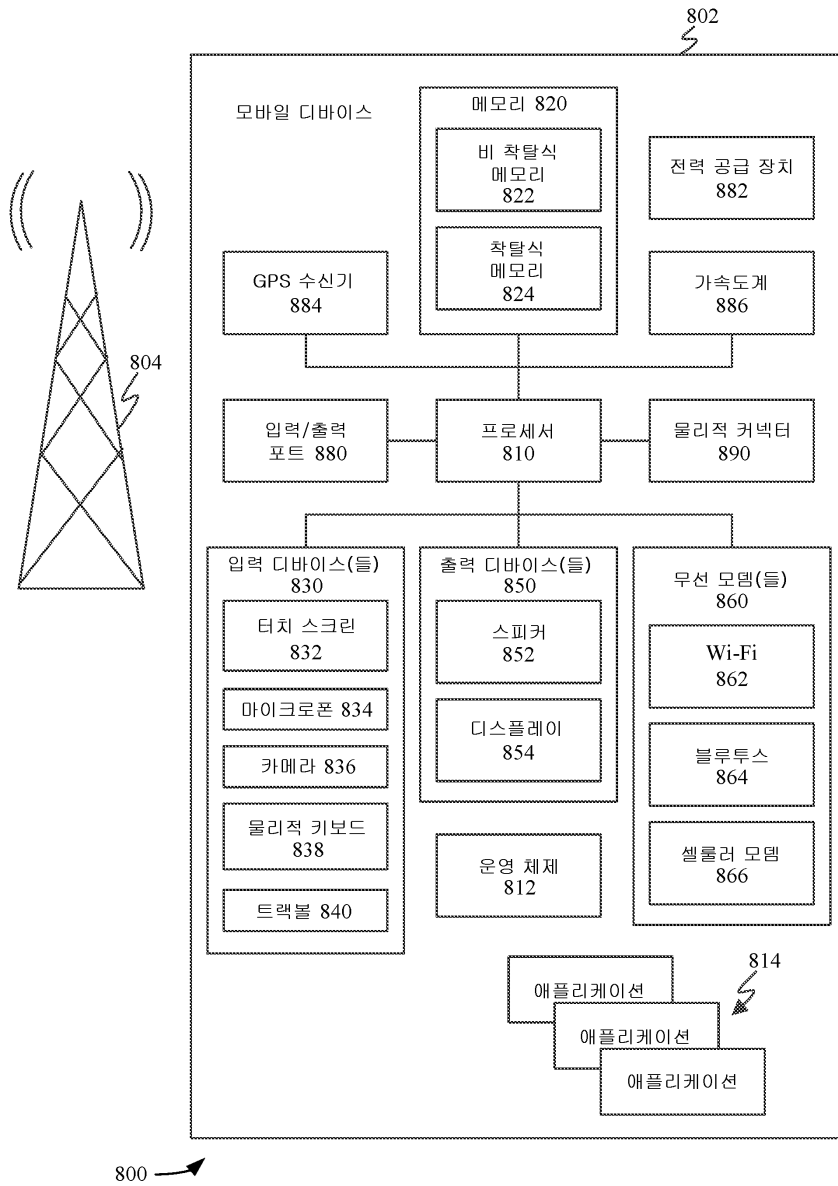
도면6



도면7



도면8



도면9

