(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0151252
(43) 공개일자 2024년10월17일

(51) 국제특허분류(Int. Cl.)
G06F 21/50 (2013.01) *G06N 5/02* (2023.01)
G06N 5/04 (2023.01)
(52) CPC특허분류
G06F 21/50 (2013.01)
G06N 5/02 (2023.01)
(21) 출원번호 10-2024-7032720(분할)
(22) 출원일자(국제) 2017년01월24일
심사청구일자 없음
(62) 원출원 특허 10-2018-7024400
원출원일자(국제) 2017년01월24일
심사청구일자 2022년01월24일
(85) 번역문제출일자 2024년09월30일
(86) 국제출원번호 PCT/US2017/014699
(87) 국제공개번호 WO 2017/127850
국제공개일자 2017년07월27일
(30) 우선권주장
62/294,258 2016년02월11일 미국(US)
(뒷면에 계속)

(71) 출원인
하산 사이드 칸란
미국 버지니아주 22066 그레이트 폴스 622 리버 밴드 로드
(72) 발명자
하산 사이드 칸란
미국 버지니아주 22066 그레이트 폴스 622 리버 밴드 로드
(74) 대리인
김종선, 이형석

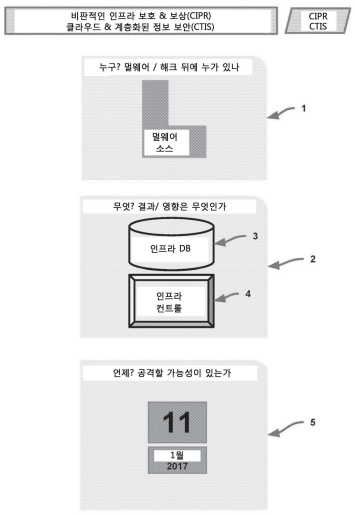
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **인공 지능을 기반으로 한 컴퓨터 보안**

(57) 요 약

본 발명은 인공 지능을 기반으로 한 컴퓨터 보안 시스템에 관한 것으로, 서브 시스템들은 클라우드 및 계층적 정보 보안(CTIS)를 통한 비판적 인프라 보호 및 보상(CIPR), 사이버 공간에서 비밀 동작을 통한 기계 비밀 정보(MACINT) 및 응징(Retribution), 논리적 추론된 제로-데이터 베이스 선험적 실시간 방어(LIZARD), 비판적 사고, 메모리 및 인식(CTMP; Critical Thinking, Memory, Perception), 어휘 객관성 마이닝(LOM; Lexical Objectivity Mining), 선형 원자 퀀텀 정보 전달(LAQIT; Linear Atomic Quantum Information Transfer) 및 통합 노드를 부착한 베이스 연결 조화를 갖는 범용 BCHAIN 모든 연결들(UBEC; Universal BCHAIN Everything Connections)을 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06N 5/04 (2023.01)

(30) 우선권주장
62/286,437 2016년01월24일 미국(US)
62/307,558 2016년03월13일 미국(US)
62/323,657 2016년04월16일 미국(US)
15/145,800 2016년05월04일 미국(US)
62/326,723 2016년04월23일 미국(US)
62/341,310 2016년05월25일 미국(US)
62/439,409 2016년12월27일 미국(US)
15/264,744 2016년09월14일 미국(US)

명 세 서

청구범위

청구항 1

인공 지능을 기반으로 한 컴퓨터 보안 시스템에 있어서,

상기 시스템은 프로그램 된 명령어를 저장하는 메모리; 및

상기 메모리에 연결되어 상기 프로그램 된 명령들 및 적어도 하나의 데이터 베이스를 실행하는 프로세서;를 포함하며,

상기 시스템은 지정된 기능을 제공하는 컴퓨터 구현 시스템을 포함하고,

상기 컴퓨터 구현 시스템은,

클라우드 및 계층화된 정보 보안(CTIS; Cloud & Tiered Information Security)을 통한 비판적 인프라 보호 및 보상(CIPR; Critical Infrastructure Protection & Retribution);

사이버 공간에서 비밀 동작을 통한 기계 비밀 정보(MACINT)와 응징(Retribution)으로 구현;

논리적으로 추론된 제로-데이터베이스 선험적 실시간 방위(Logically Inferred Zero-database A-priori Realtime Defense (LIZARD);

비판적 사고의 기억과 인식(CTMP; Critical Thinking Memory & Perception)이거나, 어휘 객관성 마이닝(LOM; Lexical Objectivity Mining);

선형 원자 퀀텀 정보 전달(LAQIT; Linear Atomic Quantum Information Transfer); 및

통합 노드를 부착한 베이스 연결 조화를 갖는 범용 BCHAIN 모든 연결들(UBEC; Universal BCHAIN Everything Connections) 시스템 중 어느 하나인, 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 인공 지능을 기반으로 한 컴퓨터 보안 시스템에 관한 것이다.

배 경 기 술

[0002] 컴퓨터 네트워크 보안과 관련된 문제는 종종 복잡한 문제의 경우 인력 전문가의 도움을 필요로 한다. 컴퓨터 및 네트워크 기능의 급속한 확장은 해커를 포함한 악의적인 엔티티에 의해 악용되었으며, 이는 결국 인간 전문가에 의존한 전통적인 솔루션을 압도했다. 인공 지능으로 구동되는 전략은 이러한 상황의 한계를 극복하는 솔루션이 되고 있다. 그러나 새로운 전략은 효과적으로 인간 사고 프로세스를 모방하고 컴퓨터 하드웨어로 구현되도록 채택된 고급 모델을 필요로 한다.

발명의 내용

해결하려는 과제

[0003] 본 발명의 컴퓨터 보안 시스템은 서브 시스템들은 클라우드 및 계층적 정보 보안(CTIS)를 통한 비판적 인프라 보호 및 보상(CIPR), 사이버 공간에서 비밀 동작을 통한 기계 비밀 정보(MACINT) 및 응징(Retribution), 논리적 추론된 제로-데이터 베이스 선험적 실시간 방어(LIZARD), 비판적 사고, 메모리 및 인식(CTMP; Critical Thinking, Memory, Perception), 어휘 객관성 마이닝(LOM; Lexical Objectivity Mining), 선형 원자 퀀텀 정보 전달(LAQIT; Linear Atomic Quantum Information Transfer) 및 통합 노드를 부착한 베이스 연결 조화를 갖는 범용 BCHAIN 모든 연결들(UBEC; Universal BCHAIN Everything Connections)을 포함하는 것을 목적으로 한다.

과제의 해결 수단

[0004] 상기한 바와 같이 목적을 달성하기 위한 본 발명의 인공 지능을 기반으로 한 컴퓨터 보안 시스템에 있어서, 상기 시스템은 프로그램 된 명령어를 저장하는 메모리 및 상기 메모리에 연결되어 상기 프로그램 된 명령들 및 적어도 하나의 데이터 베이스를 실행하는 프로세서를 포함하며, 상기 시스템은 지정된 기능을 제공하는 컴퓨터 구현 시스템을 포함하는 시스템이다.

[0005] 상기 컴퓨터 구현 시스템은 클라우드 및 계층화된 정보 보안(CTIS; Cloud & Tiered Information Security)을 통한 비판적 인프라 보호 및 보상(CIPR; Critical Infrastructure Protection & Retribution)이고:

[0006] a) 해커 활동을 보고하는 에이전트들의 네트워크로 구성된 신뢰 플랫폼(Trusted Platform);

[0007] b) 관리형 암호화 보안(Managed Encrypted Security), 연결 및 규정 준수 솔루션 및 서비스(Connectivity & Compliance Solutions & Services)를 제공하는 관리형 네트워크 및 보안 서비스 제공자(MNSP; Managed Network & Security Service Provider);를 더 포함하고,

[0008] 가상 사설망 (VPN; Virtual Private Network)은 상기 MNSP와 상기 신뢰 플랫폼을 연결하고, 여기서 상기 VPN은 상기 신뢰 플랫폼으로 및 상기 신뢰 플랫폼으로부터 통신 채널을 제공하며, 상기 MNSP는 기업 네트워크 내의 모든 트래픽을 분석하는데 적합하고, 상기 트래픽은 상기 MNSP로 라우트되는 시스템이다.

[0009] 상기 MNSP는,

[0010] a) 외부 코드로부터 목적과 기능을 도출하여 악의적인 의도가 있거나 정당한 사유가 없는 경우 외부 코드를 차단하고, 이전의 이력 데이터에 대한 참조 없이 자체의 위협을 분석하는, 논리적으로 추론된 제로-데이터 베이스 선험적 실시간 방어(Logically Inferred Zero-database A-priori Realtime Defense (LIZARD));

[0011] b) 보안 룰 세트들의 유효성을 테스트하기 위한 가상적인 보안 시나리오를 제공하는, 인공의 보안 위협(AST; artificial security threat);

[0012] c) 이전 형식들에서 새로운 혼합(Hybrid) 형식을 생성하는 프로세스를 수행하는, 창조성 모듈(Creativity Module);

[0013] d) 정보 협력을 식별하고 보안 관련 행동 패턴들을 추출하며, 다수의 음모성 보안 이벤트들의 정기적인 백그라운드 검사를 제공하고, 관련 없어 보이는 보안 이벤트들 사이의 패턴 및 상호 관계를 결정하기 위한 시도를 하는, 음모 탐지(Conspiracy Detection);

[0014] e) 이벤트들과 이벤트들의 보안 응답들 및 특성들을 저장하고 인덱스하며, 상기 응답은 차단/승인 결정들을 포함하는, 보안 행동(Security Behavior);

[0015] f) 빅 데이터 및 멀웨어(malware) 서명 인식을 활용하고, 상기 AST를 상기 창조성 모듈과 함께 활용하여 향후 잠재적인 멀웨어(malware)의 변화를 모방하는, 반복적인 지능 성장/ 지능 발전(I2GE; iterative Intelligence Growth / Intelligence Evolution); 및

[0016] g) 상기 차단/ 승인 결정들을 비판하고, 보충적인 보안 계층 역할을 하며, 상기 I2GE, 상기 LIZARD 및 상기 신뢰 플랫폼으로부터 상호 참조 지능을 활용하며, 문제에 관한 객관적인 결정을 내릴 수 있는 자체 능력을 평가하고, 내부의 낮은 신뢰로 결정을 내리는 것을 삼가 하는, 비판적 사고, 메모리 및 인식(CTMP; Critical Thinking, Memory, Perception);을 포함하는 시스템이다.

[0017] 상기 기업 네트워크의 장치에서 동작하는데 적합한 LIZARD 라이트 클라이언트(LIZARD Lite Client)는 상기 MNSP 내의 상기 LIZARD와 안전하게 통신하는 시스템이다.

[0018] 상기 기업 네트워크의 나머지 부분이 보안 책임에 노출되지 않도록 일반 컴퓨터 보다 더 높은 보안 책임을 가진 HTTP 서버를 포함하는 하위 네트워크로 구성되는 비무장 지대(DMZ; Demilitarized Zone)를 더 포함하는 시스템이다.

[0019] 상기 I2GE는 반복적인 진화(Iterative Evolution)를 포함하고, 상기 반복적인 진화에서 병렬적 발전 경로들이 성숙되고 선택되고, 반복적인 세대들은 동일한 AST에 적응하고, 최고의 성격적 특성을 가진 경로는 가장 큰 보안 위협에 저항하게 되는 시스템이다.

[0020] 상기 LIZARD는,

[0021] a) 컴퓨터 코드를 읽고 쓰기 위해 프레임 워크를 제공하는, 구문 모듈(Syntax Module);

[0022] b) 상기 구문 모듈을 사용하여 코드로부터 목적을 추출하고, 목적을 복합적 목적 형식으로 출력하는, 목적 모듈(Purpose Module);

[0023] c) 상기 기업 네트워크 및 데이터 베이스가 가상 환경에서 복제되고, 민감한 데이터가 가짜(모조) 데이터로 대체되며, 타겟의 행동에 따라, 상기 환경은 더 많은 모조 요소 또는 시스템의 실제 요소를 포함하도록 실시간으로 동적으로 변경될 수 있는, 가상의 난독화(Virtual Obfuscation);

[0024] d) 상기 가상의 난독화의 분석적 결론에 도달했을 때 보복(Retribution) 의 한 형태를 제공하는, 신호 모방기(Signal Mimicry);

[0025] e) 외부 코드의 모든 내부적 함수들이 의미가 있는지 여부를 확인(check)하는, 내부 일관성 검사(Internal Consistency Check);

[0026] f) 상기 구문 모듈 및 상기 목적 모듈을 이용하여 상기 외부 코드를 복합적인 목적 형식(Complex Purpose Format)으로 축소하기 위한, 외부 코드 재 작성(Foreign Code Rewrite);

[0027] g) 데이터 및 전송 패킷들에 은밀하게 삽입된 코드를 검출하는, 비밀 코드 탐지(Covert Code Detection);

[0028] h) 필요성과 목적의 매핑된 계층이며, 상기 외부 코드가 상기 시스템의 전반적인 목적에 부합하는지 결정하기 위해 참조되는, 니드 맵 매칭(Need Map Matching);을 포함하고,

[0029] 라이트(writing)하기 위해, 구문 모듈은 상기 목적 모듈에서 복잡한 형식화된 목적을 수신한 다음 임의 코드 구문으로 코드를 작성하고, 헬퍼 함수가 상기 임의 코드를 실제 실행 코드로 변환하며; 리드(reading)하기 위해, 상기 구문 모듈은 상기 목적 모듈에 해당 코드의 기능성에 대한 목적을 도출하기 위한 코드의 구문적 해석을 제공하고;

[0030] 상기 신호 모방기는 상기 구문 모듈을 사용여 해커와 멀웨어의 통신 구문을 이해하고, 그러한 통신을 가로채서 민감한 데이터를 상기 해커에게 성공적으로 전송했다는 잘못된 인상을 멀웨어에게 주고, 해커들 또한 상기 LIZARD에 의한 멀웨어의 오류 코드를 전송 받아 상기 오류 코드가 멀웨어로 부터 온것처럼 보이게 하고;

[0031] 상기 외부 코드 재 작성은 파생된 목적(Purpose)을 사용하여 코드 세트를 작성하여 상기 외부 코드의 원하는 목적 및 이해된 목적만이 상기 기업 내에서 실행되는 것을 보장하고, 의도하지 않은 함수의 실행이 상기 시스템에 액세스하지 못하도록 하는 시스템이다.

[0032] 상기 외부 코드 재 작성이 잠재적으로 검출되지 않는 악의적인 이용을 완화하기 위해 외부 코드를 구문적으로 재 생산하고, 조합 방식(Combination Method)은 선언된 목적(Declared Purpose)과 파생된 목적(Derived Purpose)을 비교하고 일치시키며, 상기 목적 모듈은 복합적인 목적 형식(Complex Purpose Format)을 조작하기 위해 사용되고, 파생된 목적과 함께 니드 맵 매칭은 모든 기업의 관할권을 유지하기 위한 계층적 구조를 유지하여 관할구역 지향적 니드 맵의 공석(vacancies)에 따라 코드 블록의 목적을 정의하고 정당화하고, 입력 목적은 재귀적 디버깅(Recursive Debugging) 프로세스에 대한 입력(intake)인 시스템이다.

[0033] 상기 재귀적 디버깅은 코드 세그먼트를 통해 반복하여 버그를 테스트하고 버그 수정을 적용하며, 버그가 지속되면 전체 코드 세그먼트가 원래의 외부 코드 세그먼트로 대체되고, 원래 코드 세그먼트는 상기 가상의 난독화 및 행동적 분석을 용이하게 하기 위해 태그 되고,

[0034] 외부 코드와 함께 상기 코드의 원래 상태는 상기 목적 모듈 및 상기 구문 모듈에 의해 코드 재작성을 위해 해석되며, 재작성된 버전에 영구적인 버그가 있었기 때문에, 원래의 외부 코드 세그먼트가 설치될 필요가 있는 경우 상기 외부 코드는 디버거에 의해 직접 참조되고, 재작성된 코드에서 세그먼트들은 가상 런타임 환경(Virtual Runtime Environment)에서 테스트되어 코딩 버그를 체크하고, 상기 가상 런타임 환경은 코드 세그먼트들을 실행하며, 런타임 에러들을 체크하고, 코딩 버그와 함께 가상 런타임 환경에서 생성된 에러들은 범위 또는 종류로 정의되고,

[0035] 목적 정렬(Purpose Alignment)과 함께, 코딩 버그에 대한 잠재적인 솔루션은 명시된 목적에서 코드를 다시 유도함으로써 초안작성되고, 상기 코딩 버그의 범위는 이러한 버그를 피하기 위해 대체적인 형식으로 재 작성되며, 상기 잠재적인 솔루션은 출력되고, 솔루션이 남아 있지 않으면, 해당 코드 세그먼트에 대한 코드 재 작성이 취소되고 외부 코드의 원본 코드 세그먼트가 최종 코드 세트에서 사용되는 시스템이다.

[0036] 상기 니드 맵 매칭(Need Map Matching)의 동작을 위해, LIZARD 클라우드(LIZARD Cloud) 및 LIZARD 라이트(LIZARD Lite)는 기업 관할 지점의 계층적 맵을 참조하고, 상기 입력 목적(Input Purpose)이 상기 목적 모듈을 통해 요구되거나 파생되는지 여부에 관계없이, 상기 니드 맵 매칭은 코드/기능에 대한 정당성이 기업 시스템 내에서 수행되도록 검증하고, 상기 계층적 맵의 마스터 복사본은 MNSP의 LIZARD 클라우드에 저장되며, 상기 니드 맵 매칭 내의 니드 인덱스(Need Index)는 마스터 복사본을 참조하여 계산되고, 미리 최적화된 니드 인덱스는 모든 접근 가능한 종단 클라이언트들에 분배되고, 상기 니드 맵 매칭은 전체적인 시스템의 가장 적절한 니드를 위한 니드 요청(Need Request)을 수신하고, 대응하는 출력은 적절한 니드에 해당하는 복합적인 목적 형식인 시스템이다.

[0037] 상기 기업을 위한 전체 LAN 기반시설(infrastructure)은 상기 MNSP 내에서 가상으로 재구성되고, 상기 해커는 상기 시스템이 행동 분석을 수행할 때 실제 LAN 기반시설과 가상 복제 버전의 요소들에 노출되고, 이러한 분석 결과에 위험이 나타나면, 가상의 복제 기반시설에 대한 해커의 노출은 실제 데이터 및/또는 장치가 손상될 위험을 줄이기 위해 증가하는 시스템이다.

[0038] 멀웨어 루트 시그니처(Malware Root Signature)는 상기 AST에 제공되어 상기 멀웨어 루트 시그니처의 반복/ 변형이 형성되고, 상기 멀웨어의 다형성 변이는 I2GE에서 출력으로서 제공되고, 멀웨어 탐지기로 전송되는 시스템이다.

[0039] 상기 멀웨어 탐지기는 컴퓨터의 구성요소의 세가지 레벨 모두에 배치되고, 여기서 컴퓨터의 구성요소의 세가지 레벨은 사용자 스페이스, 커널 스페이스 및 펌웨어/하드웨어 스페이스를 포함하고, 모든 스페이스들은 LIZARD Lite 에이전트에 의해 모니터링되는 시스템이다.

[0040] 상기 컴퓨터 구현 시스템은 사이버 공간에서 비밀 동작을 통한 기계 비밀 정보(MACINT)와 응징(Retribution)으로 구현되고,

[0041] a) 지능 정보 관리, 뷰잉, 및 제어를 제공하는 지능형 정보 및 구성 관리(I2CM; Intelligent Information and Configuration Management); 및

[0042] b) 입/출력 채널을 사용자에게 제공하는 관리 콘솔(MC; Management Console);

[0043] 을 포함하고,

[0044] 상기 I2CM은,

[0045] ⅰ) 중요하지 않은 중복 정보를 필터링하기 위해 일반적인 수준의 기준을 사용하고, 여러 플랫폼의 정보 스트림을 병합하고 태그 처리하는 집합체(Aggregation);

[0046] ii) 미리 구성된 보안 구성 및 연결 설정으로 새로운 기업 네트워크 장치를 배포하고 새로운 사용자 계정의 배포를 관리하기 위한 인터페이스로 구성된 구성 및 배포 서비스(Configuration and Deployment Service);

[0047] iii) 태깅된 정보 풀(pool)은 관리 콘솔 사용자(Management Console User)의 관련 관할에 따라 독점적으로 분리되는 관할에 의한 분리(Separation by Jurisdiction);

[0048] iv) 개별적인 위협에 따라 정보를 구성하는 위협별 분리(Separation by Threat); 및

[0049] ⅴ) MNSP 클라우드, 신뢰 플랫폼 또는 추가적인 타사 서비스에 접근하는 자동화된 제어(Automated Controls);

[0050] 를 포함하는 시스템이다.

[0051] 상기 MNSP 클라우드에서, 행동 분석(Behavioral Analysis)은 모의 데이터 환경(Mock Data Environment)에 있는 동안 멀웨어의 존재 상태와 수행된 동작을 관찰하고;

[0052] 상기 멀웨어가 해커에게 가짜 데이터(Fake Data)를 전송할 때, 출력 신호는 가짜 해커(Fake Hacker)에 의해 수신되도록 재라우팅되고;

[0053] 해커 인터페이스(Hacker Interface)는 멀웨어의 코드 구조를 수신하고, 멀웨어의 내부 구조를 리버스 엔지니어링하여 해커 인터페이스를 출력하고;

[0054] 가짜 해커와 가짜 멀웨어는 가상 환경에서 모방되며;

[0055] 가상의 가짜 해커는 실제 멀웨어에 응답 신호를 보내 멀웨어의 다음 행동 패턴을 관찰하고, 해커는 실제 멀웨어

의 동작/상태와 관계가 없는 가짜 응답 코드가 제공되는 시스템이다.

[0056] 익스플로잇 스캔(Exploit Scan)은 범죄 자산의 기능과 특성을 식별하고, 결과 스캔 결과는 대상 범죄 시스템(target Criminal System)에 침투하는 보복 익스플로잇 데이터 베이스(Retribution Exploits Database)를 통해 신뢰 플랫폼에서 보낸 프로그램인 익스플로잇에 의해 관리되고, 상기 보복 익스플로잇 데이터 베이스는 설립된 백도어(backdoor)와 알려진 취약점 형태로 하드웨어 벤더에 의해 제공되는 범죄 행위를 악용하는 수단을 포함하고, 통합된 법의학 증거 데이터 베이스(Unified Forensic Evidence Database)는 다수의 기업에 걸친 다수의 출처에서 수집된 법의학적 증거를 포함하는 시스템이다.

[0057] 범죄 시스템의 슬리퍼 에이전트(sleeper agent)가 기업 네트워크의 파일을 캡쳐할 때, 방화벽은 로그를 생성하고, 상기 로그는 로그 집합체(Log Aggregation)로 전달되며, 상기 로그 집합체는 장기/딥 스캔(Long-Term/Deep Scan)과 실시간/ 표면 스캔(Real-Time/Surface Scan)을 위해 데이터를 카테고리 별로 분리하는 시스템이다.

[0058] 딥 스캔(Deep Scan)은 음모 탐지 서브-알고리즘 및 외부 엔티티 관리 서브-알고리즘을 활용하면서 빅데이터에 기여하고 참여하고; 보안 체크 포인트로의 표준 로그는 로그 집합체(Log Aggregation)에서 낮은 제한 필터로 집계되고 선택되며; 이벤트 인덱스 + 트래킹(Event Index + Tracking)은 이벤트 세부사항을 저장하고; 비정상 행위 탐지(Anomaly Detection)는 잠재적인 위험 이벤트를 결정하기 위해 딥 스캔 모듈에 의해 제공된 중간 데이터에 따라 이벤트 인덱스 및 보안 행동을 사용하고; 외부 엔티티 관리 및 음모 탐지는 이벤트 분석에 관여하는 시스템이다.

[0059] 상기 신뢰 플랫폼은 임의의 컴퓨터를 검색하여 자신 또는 자신의 서버의 친척/이웃(자신과 연결된 다른 서버)가 상기 신뢰 플랫폼에 대해 이전에 더블(double) 또는 트리플(triple) 에이전트가 설정되었는지를 확인하고,

[0060] 에이전트 룩업 확인(agent lookup check)은 신뢰할 수 있는 더블 에이전트 인덱스+트래킹 클라우드 및 신뢰할 수 있는 트리플 에이전트 인덱스+트래킹 클라우드에서 수행되며;

[0061] 임의의 컴퓨터에 의해 신뢰되는 더블 에이전트는 신뢰되는 채널을 통하여 익스플로잇을 푸쉬하고, 상기 익스플로잇은 민감한 파일을 찾아서 격리하고, 그것의 정확한 상태를 신뢰 플랫폼으로 되돌려 보낸 다음, 범죄의 컴퓨터에서 민감한 파일을 지우려고 시도하는 시스템이다.

[0062] ISP API 요청은 신뢰 플랫폼을 통해서 생성되고, 네트워크 감시(Network Oversight)에서 임의의 시스템에 대한 네트워크 로그와 범죄용 컴퓨터로의 잠재적인 파일 전송이 발견되고, 메타데이터(metadata)는 어떤 컴퓨터에 파일이 전송되었는지에 대한 중요한 확신을 결정하는데 사용되고, 상기 네트워크 감시는 범죄용 컴퓨터의 네트워크 세부 사항을 발견하고 그러한 정보를 상기 신뢰 플랫폼으로 재라우팅하고, 상기 신뢰 플랫폼은 사법 수사를 지원할 수 있도록 수립된 백도어를 개발하기 위하여 소프트웨어 및 하드웨어 벤더가 제공하는 보안 APIs를 차지하기 위해 사용되는 시스템이다.

[0063] 상기 신뢰 플랫폼은 소프트웨어 또는 펌웨어 업데이트를 범죄 컴퓨터에 푸시하여 새로운 백도어를 구축하고, 플라시보 업데이트(Placebo Update)는 가까운 유사한 기계에 푸시되어 스텔스(stealth)를 유지하며, 타겟 식별 상세정보(Target Identity Details)는 신뢰 플랫폼으로 전송되고, 상기 신뢰 플랫폼은 소프트웨어/펌웨어 유지 관리자(Software/Firmware Maintainer)와 통신하여 플라시보 업데이트 및 백도어 업데이트를 관련 컴퓨터로 푸쉬하고, 백도어 업데이트(Backdoor Update)는 컴퓨터에 설치된 미리-확립된 소프트웨어 업데이트 시스템을 사용하여 범죄 컴퓨터 시스템에 새로운 백도어를 도입하고, 상기 플라시보 업데이트는 백도어를 생략하고, 유지자(Maintainer)는 백도어를 타겟에 전송할 뿐만 아니라, 타겟에 평균 이상 노출되는 컴퓨터에도 백도어를 전송하고, 백도어 업데이트(Backdoor Update)을 통해 상기 익스플로잇(Exploit)을 구현하면 민감한 파일은 격리되고 복사되어 메타데이터 사용 내역을 나중에 분석할 수 있고, 모든 보충 포렌식 데이터가 수집되어 상기 신뢰 플랫폼에서 익스플로잇의 컨택 포인트(point of contact)로 보내지는 시스템이다.

[0064] 장기 우선 순위 플래그(Long-term priority flag)는 신뢰 플랫폼 상에서 푸쉬되어 모든 변경/ 업데이트에 대해 범죄 시스템을 모니터링하고, 기업 시스템(Enterprise System)은 정의된 타겟의 모든 연관에 대해 모든 제휴 시스템 입력을 스캔하는 보증 모듈(Warrant Module)에 타겟을 제출하고, 일치하는 항목이 있으면 정보가 보증서가 정의되고 표적에 침투하려고 하는 기업 시스템으로 전달되고, 입력은 서로 유익한 보안 정보를 동기화하는 바람직한 분석 모듈(Desired Analytical Moduie)로 전송되는 시스템이다.

[0065] 상기 컴퓨터 구현 시스템은 논리적으로 추론된 제로-데이터베이스 선험적 실시간 방위(Logically Inferred Zero-database A-priori Realtime Defense (LIZARD))이고,

[0066] a) 대부분 고정된 프로그램 모듈을 포함하는 정적 코어(SC; Static Core);

[0067] b) 동적 쉘(Dynamic Shell)에서 모듈들을 수정, 생성 및 파괴하고, 보안 성능을 참조하기 위해 AST를 사용하고, 자동 코드 작성 방법을 처리하기 위해 반복 코어(Iteration Core)를 사용하는 반복 모듈(Iteration Module);

[0068] c) AST가 찾은 결함에 따라 기본 반복(Base Iteration)을 수정하고, 차동 논리가 적용된 후 새로운 반복이 제안되며, 상기 반복 코어는 재귀적으로 호출되고, AST에 의해 테스트된 동일한 프로세스를 거치는 차동 수정 알고리즘(Differential Modifier Algorithm);

[0069] d) AST에서 동적 쉘 반복(Dynamic Shell Iteration)의 알려진 보안 응답을 수신하고, 어떤 코드세트 구성이 보안 시나리오에 대해 알려진 정확한 응답(Correct Response)을 얻을 수 있는지를 추론하는 논리 추론 알고리즘(Logic Deduction Algorithm));

[0070] e) 상기 반복 모듈(IM)에 의해 자동으로 프로그램 된 주로 동적 프로그램 모듈들을 포함하는 동적 쉘(DS, Dynamic Shell);

[0071] f) 외부의 코드를 제한된 가상환경으로 격리하는 코드 검역(Code Quarantine);

[0072] g) 데이터 및 전송 패킷에 은밀하게 삽입된 코드를 검출하는 비밀 코드 검출 (Covert Code Detection); 및

[0073] h) 외부 코드 목적을 파생한 후, 부분 또는 전체 코드 자체를 다시 작성하고 재작성만 실행되도록 허용하는 외부 코드 재작성(Foreign Code Rewrite);

[0074] 을 더 포함하고,

[0075] 모든 기업 장치들은 LIZARD를 통해 라우팅되고, 기업 장치를 실행하는 모든 소프트웨어 및 펌웨어는 LIZARD를 통해 영구적인 프록시로써 다운로드/ 업로드를 수행하기 위한 하드코딩 되고, LIZARD는 동작중인 데이터, 사용중인 데이터 및 휴지중인 데이터를 포함하는 세가지 타입의 데이터와 상호 작용하고, LIZARD는 파일, 이메일, 웹, 모바일, 클라우드 및 제거가능한 미디어로 구성된 매체와 상호 작용하는 시스템이다.

[0076] 시스템에 있어서,

[0077] a) 시스템이 낮은 신뢰 결정만을 수행할 수 있을 때 향후 반복 개선을 위해 데이터가 AST로 릴레이되는 AST 오버플로우 릴레이(AST Overflow Relay);

[0078] b) 외부 코드 블록의 모든 내부 기능이 의미가 있는지 여부를 확인하는 내부 일관성 검사(Internal Consistency Check);

[0079] c) 재 작성의 입력/ 출력 동적(dynamic)이 원본과 동일하다는 것을 확인하여, 원래(original) 코드의 숨겨진 모든 익스플로잇이 중복되어 실행되지 않도록 하는 미러 테스트(Mirro test);

[0080] d) 외부 코드가 상기 시스템의 전체적인 목표에 부합하는지 결정하기 위해 참조되는 필요(need)와 목적의 매핑된 계층구조를 포함하는 니드 맵 매칭(Need Map Matching);

[0081] e) 혼합 환경에 제공할 데이터를 선택하고, 민감한 정보가 의심스러운 멀웨어에 접근 할 수 없도록 하는 우선순위를 선택하는 실제 데이터 동기화 장치(Real Data Synchronizer);

[0082] f) 가상 환경의 외부에서 오는 엔티티와 데이터 간의 중개 인터페이스인 데이터 매니져(Data manager);

[0083] g) 가상의 가짜 환경에 코드를 점차적으로 그리고 부분적으로 침투시켜 코드를 혼동시키고 제한하는 가상의 난독화(Virtual Obfuscation);

[0084] h) 멀웨어를 모의 데이터 환경(Mock Data Environment)으로 몰래 그리고 이산적으로 전송하는 비밀 수송 모듈(Covert Transportation Module); 및

[0085] i) 의심스러운 엔티티(Suspicious Entity)로 다운로드되고 의심스러운 엔티티(Suspicious Entity)로부터 업로드 되는 모든 정보를 추적하는 데이터 호출 추적(Data Recall Tracking);을 더 포함하는 시스템이다.

[0086] 4가지 서로다른 유성의 목적을 비교하여 LIZARD에 의해 엔티티의 존재와 행동이 상기 시스템의 전반적인 목표을 위해 생산적으로 타당하고 합당한지를 확인하는 목적 비교 모듈(Purpose Comparison Module);을 더 포함하는 시스템이다.

[0087] 상기 반복 모듈은 상기 SC를 사용하여 상기 데이터 리턴 릴레이(DDR, Data Return Relay)에서 정의된 목적에 따

라 DS의 코드 기반을 구문적으로 수정하고, LIZARD의 수정된 버전이 상기 AST에 의해 다양한 보안 시나리오와 병행하여 스트레스 테스트되는 시스템이다.

[0088] 상기 SC의 내부에서, 논리 도출(Logic Derivation)은 초기의 간단한 함수로부터 논리적으로 필수적인 함수를 도출하여, 함수 종속성의 전체 트리가 명시된 복합 목적으로부터 구축(built)되고;

[0089] 코드 변환(Code Translation)은 구문 모듈(Syntax Module) 기능에 의해 직접 이해할 수 있는 임의의 일반 코드를 임의의 선택된 알려진 컴퓨터 언어로 변환하고, 알려진 컴퓨터 언어를 임의의 코드로 변환하는 역관계도 수행되고;

[0090] 논리 축소(Logic Reduction)는 코드로 작성된 로직을 단순화된 형태로 축소하여 상호 연결된 함수의 맵을 생성하고;

[0091] 복합 목적 형식(Complex Purpose Format)는 전체 목적을 나타내는 상호 연결된 서브-목적을 저장하기 위한 저장 형식이고;

[0092] 목적 연관성(Purpose Associations)은 어떤 기능과 유형의 행동이 어떤 종류의 목적을 지칭하는지에 대한 하드 코딩된 참조(reference)이고;

[0093] 반복 확장(Iterative Expansion)은 상기 목적 연관성을 참조하여 단순한 목표를 복잡한 목적으로 발전시키기 위한 세부사항과 복잡도를 추가하며;

[0094] 반복적인 해석은 모든 상호 연결된 기능을 통해 반복되고, 목적 연관성을 참조하여 해석된 목적을 생성하고;

[0095] 외부 코어(Outer Core)는 알 수 없는 외부 코드에 대한 논리적 목적을 도출하고, 명시된 함수 코드 목표에서 실행 가능한 코드를 생성하기 위해 함께 동작하는 상기 구문 및 목적 모듈들에 의해 형성되고;

[0096] 외부 코드(Foreign Code)는 LIZARD에 알려지지 않은 코드이며, 기능 및 의도된 목적은 알 수 없고, 상기 외부 코드는 내부 코어에 대한 입력이고 도출된 목적(Derived Purpose)은 출력이고, 상기 도출된 목적은 목적 모듈에 의해 추정된 주어진 코드의 의도이며, 도출된 목적이 복합 목적 형식으로 반환되는 시스템이다.

[0097] 상기 IM은 보안 성능의 참조를 위해 AST를 사용하고, 상기 자동 코드 작성 방법을 처리하기 위해 상기 반복 코어를 사용하며, LIZARD가 낮은 신뢰를 가지고 의사 결정을 내려야 할 때 악의적인 공격과 나쁜 행위에 대한 DRR 데이터가 AST로 전달되고; 상기 반복 코어 내부에서, 차동 수정 알고리즘(DMA; Differential Modifier Algorithm)은 내부 코어(Inner Core)로 부터 구문/ 목적 프로그래밍 기능 및 시스템 지향 지침(Syntax/Purpose Programming Abilities and System Objective Guidance)을 수신하고, AST 17에서 발견된 결함에 따라 기본 반복(Base Iteration)을 수정하기 위해 이러한 코드세트를 사용하고; 가상 실행 환경을 실행하는 동안 상기 기본 반복을 통해 통과된 보안 위협을 표시하기 위해 보안 결과 결함(Security Result Flaws)이 시각적으로 표시되는 시스템이다.

[0098] 상기 DMA 내부에서, 현재 상태(Current State)는 상징적으로 상관된 모양, 크기 및 위치를 가진 동적 쉘(Dynamic Shell) 코드세트를 나타내고, 이들 모양과 다른 구성들은 보안 지능 및 반응과 다른 구성을 나타내고, 상기 AST는 잘못된 것으로 밝혀진 현재 상태의 잠재적인 응답과 정확한 응답이 무엇인지를 제공하고;

[0099] 공격 벡터(Attack Vector)는 사이버 보안 위협에 대한 상징적인 시연(demonstration)로 작용하고, 방향, 크기 및 색상은 모두 공격 벡터, 멀웨어의 크기 및 멀웨어의 유형과 같은 가설적인 보안 속성과 상호 관련이 있고, 상기 공격 벡터는 코드 세트의 보안 응답을 나타내기 위해 코드 세트를 상징적으로 반사하고(bounce off);

[0100] 정확한 상태(correct State)는 상기 동적 쉘의 코드 블록으로부터 원하는 보안 응답을 생성하기 위한 DMA의 프로세스의 최종 결과를 나타내며, 상기 현재 상태와 상기 정확한 상태 사이의 차이로 인해 서로 다른 공격 벡터 응답이 발생하고;

[0101] 상기 AST는 정확한 보안 응답(Correct Security Response)과 함께 알려진 보안 결함(Known Security Flaws)을 제공하고, 논리적 추론 알고리즘(Logic Deduction Algorithm)은 상기 DS의 이전 반복을 사용하여 정확한 보안 응답 프로그램으로 알려진 동적 쉘의 우수하고 잘 갖추어진 반복을 생성하는 시스템이다.

[0102] 가상 난독화(Virtual Obfuscation)의 내부에서, 의심스러운 코드는 데이터의 절반이 모의 데이터와 지능적으로 혼합되는 환경에 은밀하게 할당되고, 실제 시스템 내에서 작동하는 모든 대상은 가상 격리(Virtual Isolation)로 인해 부분적으로 또는 완전히 모의 데이터 환경으로 쉽고 은밀하게 전송될 수 있고, 모의 데이터 생성기

(Mock Data Generator)는 불법적이고(counterfeit) 불필요한 데이터를 생성하기 위한 템플릿으로서 실제 데이터 동기화기(Real Data Synchronizer)를 사용하며, 유입되는 외부 코드의 인식에 대한 신뢰의 인식된 위험은 LIZARD가 선택한 난독화 수준에 영향을 미칠 것이고, 악의적인 코드에 대한 높은 신뢰는 대량의 모의 데이터를 포함하는 환경으로의 할당을 호출하고, 악의적인 코드에 대한 낮은 신뢰는 실제 시스템 또는 100% 모의 데이터 환경으로의 할당을 호출할 수 있는 시스템이다.

[0103] 데이터 회수 추적(Data Recall Tracking)은 의심스러운 엔티티로부터 업로드되고 의심스러운 엔티티로 다운로드된 모든 정보를 추적하고, 모의 데이터가 합법적인 기업 엔티티에 전송된 경우, 모든 모의 데이터를 호출하기 위한 콜백이 수행되고, 상기 실제 데이터는 대체 데이터로써 전송되고, 합법적인 기업 엔티티는 데이터가 가짜가 아님을 확인할 때까지 특정 정보에 대한 동작을 보류할 수 있도록 콜백 트리거(callback trigger)가 구현되는 시스템이다.

[0104] 행동 분석(Behavioral Analysis)은 의심스러운 엔티티의 다운로드 및 업로드 동작을 추적하여 잠재적인 정정 동작(Corrective Action)을 결정하고, 실제 시스템은 상기 가상 환경의 외부에 전부 존재하는 원래의 실제 데이터를 포함하고, 모의 데이터를 대체하는 실제 데이터는, 실제 데이터가 필터링 되지 않은 상태에서 데이터 회수 추적에 제공되어 실제 데이터 패치가 모의 데이터를 이전의 의심스러운 엔티티의 실제 데이터로 대체할 수 있도록 생성되는 곳이고; 가상으로 고립된 환경(Virtually Isolated Environment)에 숨겨진(submerged)진 데이터 관리자(Data Manager)는 데이터 회수 추적(Data Recall Tracking)에서 실제 데이터 패치(Real Data Patch)를 수신하고; 행동 분석(Behavioral Analysis)에 의해 무해한 코드(Harmless Code)가 악성코드로 삭제된 경우, 정정 동작(Corrective Action)은 이전의 의심스러운 엔티티의 모의 데이터를 그것이 나타내는 실제 데이터로 대체하기 위해 수행되고, 보안 토큰(Secret Token)은 LIZARD에 의해 생성 및 할당된 보안 문자열이고 실제로 무해한 엔티티가 그것의 작업을 진행할 수 없도록 하고, 상기 토큰(Token)이 누락된 경우, 이는 멀웨어라는 위험 평가 때문에 합법적인 엔티티가 부분적으로 모의 데이터 환경에 우연히 배치되었다는 가능성 있는 시나리오를 나타내고, 이후 지연 인터페이스(Delay Interface)로 지연된 세션(Delayed Session)이 활성화되고; 상기 토근이 발견되면, 이는 서버 환경이 실제이므로 지연된 세션이 비활성화됨을 나타내는 시스템이다.

[0105] 상기 행동 분석(Behavioral Analysis) 에서 목적 맵(Purpose Map)은 시스템 목표의 계층 구조로서 전체 기업 시스템에 목적을 부여하고, 선언된 활동 및 코드 베이스 목적은 의심스러운 엔티티가 주장하는 바에 상관없이 고유 시스템 요구와 비교되고; 활동 모니터링(Activity Monitoring)에서 의심스러운 엔티티의 스토리지, CPU 프로세싱, 및 네트워크 활동이 모니터링 되고, 상기 구문 모듈은 그러한 활동을 원하는 기능 관점에서 해석하고, 그러한 기능은 목적 모듈에 의한 행동에서 의도된 목적으로 변환되고, 코드 베이스는 의심스러운 엔티티의 소스 코드/ 프로그래밍 구조이고 상기 구문 모듈로 전달 되며, 상기 구문 모듈은 코딩 구문을 이해하고 프로그래밍 코드와 코드 활동을 상호 연결된 함수의들 중간 맵으로 감소시키며, 상기 목적 모듈은 의심스러운 엔티티, 코드베이스 목적의 출력 및 동작 목적의 의도를 산출하고, 상기 코드베이스 목적은 LIZARD의 구문 프로그래밍 기능에 의해 파생된 엔티티의 알려진 목적, 기능, 관할권 및 권한을 포함하며, 상기 동작 목적에는 LIZARD의 스토리지, 처리 및 네트워크 활동에 대한 이해로 이해되는 엔티티의 알려진 목적, 기능, 관할권 및 권한을 포함하고, 선언된 목적은 엔티티 자체에 의해 선언된 엔티티의 가정된 목적, 기능, 관할권 및 권한이고, 상기 요구된 목적은 상기 기업 시스템이 요구하는 예상되는 목적, 기능, 관할권 및 권한을 포함하고, 모든 목적은 상기 비교 모듈(Comparison Module)에서 비교되며, 목적들 사이의 불일치는 정정 동작(Corrective Action)으로 연결되는 목적상 발산 시나리오(Divergence in Purpose scenario)가 호출되는 시스템이다.

[0106] 상기 컴퓨터 구현 시스템은 비판적 사고의 기억과 인식(CTMP; Critical Thinking Memory & Perception)이고,

[0107] a) 알려진 인식 범위를 취하여 인식의 비판적 사고 범위를 포함하도록 업그레이드 하는 비판적 규칙 범위 확장기(CRSE; Critical Rule Scope Extender);

[0108] b) 인식의 비판적 사고 범위를 사용하여 유도된 정정 규칙을 나타내는 정정 규칙(Correct rules);

[0109] c) 필요하고 관련 있는 비판적 사고 결정을 내리기 위해 현재 확정되고 혼돈 분야(Chaotic Field)에서 기억의 스캔에 따라 실행되는 규칙을 실행하는 규칙 실행(RE; Rule Execution);

[0110] d) 지각 관찰자 에뮬레이터(POE; Perception Observer Emulator)뿐만 아니라 RE에 의해서 도달된 결론을 비교함으로써 CTMP의 전반적인 출력을 결정하기 위한 최종 논리를 생성하는 비판적인 결정 출력(Critical Decision Output);

[0111] 여기서 상기 POE는 관찰자의 에뮬레이션을 생성하고 모든 잠재적인 지각 포인트를 관찰자 에뮬레이션의 변화와

시험/비교 하고;

[0112] 상기 RE는 규칙 세트의 변화를 추적하는데 사용되는 체커보드로 구성되며, 상기 보드의 객체는 주어진 보안 상황의 복잡도를 나타내며, '보안 체커보드'를 통한 상기 객체의 이동은 상기 보안 규칙 세트의 응답에 의해 관리되는 보안 상황의 발전을 나타내는 시스템이다.

[0113] 시스템에 있어서,

[0114] a) 선택된 패턴 매칭 알고리즘(SPMA; Selected Pattern Matching Algorithm)에 의해 제공된 결정인 주관적인 의견 결정들(Subjectiv opinion decisions);

[0115] b) SPMA의 원시 메타 데이터로 구성되고, 알고리즘의 기계적 처리와 상기 결정에 도달한 방법을 성명하는 입력 시스템 메타데이터(Input system Metadata);

[0116] c) 속성을 비교하여 주장을 논리적으로 이해하는 이유 처리(Reason Processing);

[0117] d) 파생된 결과로서의 규칙을 사용하고, 당면한 문제의 범위를 결정하기 위한 참고 포인트로서 사용되는 규칙 처리(Rule Processing);

[0118] e) 실행 가능한 규칙을 위한 시장 변화 로그들을 스캔하는 메모리 웹(Memory Web);

[0119] f) 상기 SPMA로부터 메타 데이터 로그들을 수신하는 원시적 인식 생성(Raw Perception Production)으로, 상기 로그들은 분석되며 그러한 알고리즘의 상기 인식을 나타내는 인식이 형성되고, 상기 인식은 인식 복잡 형식(PCF; Perception Complex Format)에 저장되고, POD에 의해 에뮬레이트 되며, 상기 SPMA에 의해 이미 응용되고 실행된 인식의 관점을 나타내는 인식의 응용된 관점(Applied Angles of Perception);

[0120] g) 창조성 모듈에 영향을 주고 인식의 응용된 관점(Applied Angles of Perception)에 의해 제공된 입력에 따라 형성된 혼합된 인식을 생성하는 자동 인식 발견 메커니즘(APDM; Automated Perception Discovery Mechanism);

[0121] h) 보고 가능한 로그의 범위를 벗어나는 잠재적인 알려지지 않은 지식의 범위 및 유형을 추정하고, CTMP의 후속 비판적 사고 기능이 모든 관련 지식의 잠재적 범위에 활용할 수 있고, 비판적 사고(Critical Thinking)는 사고 기반 규칙의 외부 쉘 관할 구역을 나타내는 자기 비판적인 지식 밀도(SCKO; Self-Critical Knowledge Density);

[0122] i) 현재 인식의 응용된 관점으로부터 연루될 수 있는 인식 데이터의 관점을 유도하는 함축 유도(Implication Derivation);

[0123] 를 더 포함하고,

[0124] 여기서 상기 SPMA는 인식과 규칙을 통해 CTMP가 수행된 비판적 사고와 병치되는 시스템이다.

[0125] 시스템에 있어서,

[0126] a) CVF가 규칙 구문 파생(Rule Syntax Derivation)으로부터 수신한 상기 인식으로 형성된 인식 매칭(Perception Matching);으로, 새로 형성된 CVF는 유사한 인덱스를 갖는 PS 내의 관련 인식을 검색하는데 사용되며, 상기 잠재적인 일치들은 규칙 구문 생성(RSG; Rule Syntax Generation)으로 복귀되고;

[0127] b) 인식 저장(PS; Perception Storage)의 데이터베이스 색인을 기준으로 향상된 로그 데이터로부터 도출한 CVF를 사용하는 저장 검색(SS; Storage Search)이고, PS에서 인식은 가중치 외에도 비교 가능한 변수 형식(CVF; comparable variable format)을 색인으로 저장하며;

[0128] c) 엔지니어가 상기 SPMA로부터 변수 할당을 역으로 수행하는 메트릭 처리(Metric Processing);

[0129] d) 할당 응답의 원래 인식을 복제하기 위하여 상기 할당 응답 및 대응하는 시스템 메타데이터를 사용하는 인식 복제(PD; Perception Deduction);

[0130] e) 디버깅 및 알고리즘 추적이 구문 기반 정보 분류를 사용하여 별개의 범주로 구분되는 메타 데이터 범주화 모듈(MCM; Metadata Categorization Module)로, 상기 범주는 위험 및 기회에 대한 상관 관계를 가진 명확한 할당 응답을 구성하고 생성하는데 사용되고;

[0131] f) 메트릭의 범주를 인식의 관점으로 분류하는 메트릭 조합(Metric Combination);

[0132] g) 개별적인 메트릭을 인식의 전체 관점으로 되돌려놓는 메트릭 변환(Metric Conversion);

[0133] h) 개별적인 데이터베이스에 카테고리 별로 다양한 관점의 인식 및 다수의 메트릭을 저장하는 메트릭 확장(ME; Metric Expansion);

[0134] i) 정보의 스트림을 비교 가능한 변수 형식(CVF)으로 변환하는 비교 가능한 변수 형식 생성기(CVFG; Comparable Variable Format Generator);를 더 포함하는 시스템이다.

[0135] 시스템에 있어서,

[0136] a) CVF가 규칙 규문 유도(RSD; Rule Syntax Derivation)로부터 수신된 인식으로부터 형성되는 인식 매칭(Perception Matching);

[0137] 상기 새롭게 형성된 CVF는 유사한 인덱스를 갖는 PS에서 관련 인식을 검색하는데 사용되며 잠재적인 매치는 규칙 구문 세대(RSG, Rule Syntax Generation)로 리턴되며;

[0138] b) 입력 데이터로부터 형성된 혼돈 분야(Chaotic Field)가 형성되는 메모리 인식(MR; Memory Recognition);

[0139] c) 전체 컨셉을 개별적으로 인덱스로 최적화하는 메모리 컨셉(Memory Concept) 색인화로, 여기서 상기 인덱스는 문자 스캐너가 혼돈 분야와 상호 작용하기 위해 사용되며;

[0140] d) 규칙의 개별 부분을 인식 태그와 함께 수신하는 규칙 수행 파서(Rule Fulfillment Parser)로, 각 부분은 메모리 인식(Memory Recognition)에 의해 혼돈 영역에서 발견되거나 발견되지 않는 것으로 표시되며, 상기 REP는 모든 부분들의 조합인 전체 규칙이 상기 RE 를 유용하게 하기 위해 혼돈 영역(Chaotic Field)에서 충분히 인식되었는지 논리적으로 추론하고;

[0141] e) 정정 규칙은 모든 동작, 속성, 조건 및 객체가 별도로 쌓이는 유형에 따라 분리되고 구성되는 규칙 구문 형식 분리(RSFS; Rule Syntax Format Separation);

[0142] f) 논리적 '흑백' 규칙이 인식 기반의 메트릭으로 전환되는 규칙 구문 어원(Rule Syntax Derivation)으로, 다수의 규칙의 복잡한 배열은 다양한 그라디언트의 여러 메트릭을 통해 표현되는 하나의 일관된 인식으로 전환되고;

[0143] g) 인식 형식(Perception Format)에 저장된 이전에 형성된 인식을 수신하고, 인식의 내부 메트릭 구조와 결합하는 규칙 구문 세대(RSG; Rule Syntax Generation)로, 이러한 그라디언트 기반의 메트릭 측정값은 원래의 인식의 입력/ 출력 정보 흐름을 에뮬레이트하는 이진 및 논리적 규칙의 집합으로 변환되고;

[0144] h) 정정 규칙(Correct rules)은 관찰되는 객체의 현실에 일치하는 규칙 세트의 정확한 표현을 나타내는 규칙 구문 형식 분리(RSFS; Rule Syntax Format Separation)로, 정정 규칙은 유형에 따라 분리 및 구성되고, 모든 동작, 속성, 조건 및 객체가 별도로 쌓여 시스템이 혼돈 영역에서 발견되는 부분과 그렇지 않은 부분을 식별하고;

[0145] i) 논리적인 원칙을 사용하여 오류를 피하고, 어떤 종류의 규칙이 인지 내에서 메트릭의 많은 그라디언트를 정확하게 나타내는지 추론하기 위한 선천적인 논리적 추론(innate Logical Deduction);

[0146] j) 메트릭들의 인식들 내에서 상호 연결된 관계를 분석하는 메트릭 맥락 분석(Metric Context Analysis)으로, 특정 메트릭은 다양한 정도의 크기를 갖는 다른 메트릭에 의존할 수 없고, 이 맥락화는 규칙이 '디지털'구칙 집합 형식 내에 있는 미러링된 상호 연결된 관계를 보완하는데 사용되고;

[0147] k) 규칙 구문 형식(RSF; Rule Syntax Format)의 구문을 준수하도록 규칙을 정렬하고 분리하는 규칙 구문 형식 변환(RSFC; Rule Syntax Format Conversion);을 더 포함하고,

[0148] 여기서 직관적인 결정(Intuitive Decision)은 활용 인식을 통해 비판적인 사고에 관여하고, 사고 결정(Thinking Decision)은 활용 규칙을 통해 비판적인 사고에 관여하며, 여기서 인식은 내부적 형식에 따라 정의된 형식 구문에 따라 직관적인 결정으로부터 수신된 데이터이고, 실행된 규칙(Fulfilled Rules)은 사고 결정으로부터 수신된 데이터이며, 상기 RE로부터 실행할 수 있는 규칙 세트의 모음이고, 상기 데이터는 내부 형식에 정의된 형식 구문에 따라 통과되고;

[0149] 여기서 동작들은 이미 수행되었거나 실행될 수 있는 동작의 활성화를 위해 고려되는 것을 나타내며, 여기서 속성은 동작(Action), 조건(Condition) 및 객체(Object)등 다른 요소를 설명하는 속성과 같은 특징을 나타내고, 조건은 논리적 연산 또는 연산자를 나타내며, 객체는 특징을 적용할 수 있는 대상을 나타고;

[0150] 여기서 분리된 규칙 형식(Separated Rule Format)은 사전-메모리 인식(pre-Memory Recognition) 단계로 간주되는 규칙 구문 형식 분리(RSFS; Rule Syntax Format Separation)으로부터의 출력 및 다음-메모리 인식 단계로

간주되는 메모리-인식(MR; Memory Recognition)으로 부터의 출력으로 사용되는 시스템이다.

[0151] 시스템에 있어서,

[0152] a) 상기 로그들의 형식을 스캔 가능한 단일 혼돈 영역으로 결합하는 혼돈 영역 파싱(CFP; Chaotic Field Parsing);

[0153] b) 상기 정정 규칙들(Correct Rules)을 보충하기 위해 메모리 인식(MR)로부터 생성되는 추가 규칙들(Extra Rules);

[0154] 을 더 포함하고,

[0155] 내부의 인식 매칭(PM; Perception Matching)에서, 메트릭 통계는 인식 저장(Perception Storage)에서 통계 정보를 제공하고, 오류 관리(Error Management)는 개별 메트릭들 중 하나에서 발생하는 구문 및/또는 논리적 오류들을 분석하며, 분리 메트릭들(Separate Metrics)은 각각의 개별 메트릭을 입력 인식의 단일의 단위로 결합된 이후 격리하고, 노드 비교 알고리즘(NCA; Node Comparison Algorithm)은 두 개 이상의 CVF로 구성되는 노드를 수신하며, 여기서 CVF의 각각의 노드는 특성의 크기의 정도를 나타내고, 유사성 비교가 개별 노드 단위로 수행되고, 총 합의 변화가 계산되고, 여기서 더 작은 변화량 수는 더 가까운 매치를 나타내는 시스템이다.

[0156] 시스템에 있어서,

[0157] a) 아날로그 형식 인식이 결정과 관련되어 있는'아날로그'형식에 따라 인식을 처리하고 직관적인 사고(아날로그)는 단계 없이 부드러운 곡선의 그라디언트(경사도)로 저장되는 원시 인식들(Raw Perceptions);

[0158] b) 디지털 형식 원시 규칙이 결정과 관련되어 있는 '디지털' 형식에 따라 규칙을 처리하고 논리적 사고(디지털)는 '회색 영역'이 거의 없는 단계를 저장하는 원시 규칙들(Raw Rules);

[0159] 을 더 포함하고,

[0160] 여기서 실현되지 않는 규칙들(Unfulfilled Rules)은 논리적 종속 관계에 따라 혼돈의 영역에서 충분히 인식되지 않은 규칙 세트이며, 실현되는 규칙들(Fulfilled Rules)은 논리적 종속 관계에 따라 혼돈의 영역에서 충분히 인식되는 규칙 세트이고;

[0161] 큐 관리(QM; Queue Management)는 구문 관계 복원(SRR; Syntactical Relationship Reconstruction)을 활용하여 각 개별 부품을 가장 논리적인 순서로 분석하고, 메모리 인식(Memory Recognition) 결과에 액세스하여 이진법의 예/ 아니오 플로우 질문에 응답하고, 적절한 조치를 취할 수 있으며, 여기서 상기 QM이 모든 단계적 규칙 세그먼트를 검사하고, 단일 세그먼트가 혼돈의 영역에서 누락되고 나머지 세그먼트가 적절한 관계에 있지 않으면, 상기 규칙 세트는 실행되지 않은 것으로 표시되어지는 시스템이다.

[0162] 순차적 메모리 구성은 순차 정보의 '체인'을 위해 최적화된 정보 저장 장치이며, 메모리 액세스 포인트(Points of Memory Access)에서 각 노드(블록)의 폭은 기억된 객체(노드)에 대한 관찰자의 직접 접근성을 나타내며, 접근성의 범위(Scope of Accessibility)에서 각각의 문자는 관찰자에 대한 직접적인 메모리 접근 포인트를 나타내며, 접근성의 더 넓은 범위는 시퀀스 노드 당 접근성의 더 많은 포인트가 더 많음을 나타내고, 더 많은 시퀀스는 임의로 선택된 노드가 아닌 '순서대로'만 참조되는 시퀀스가 많을수록 접근성의 범위(시퀀스의 크기와 연관된)가 좁아지고, 중첩된 하위 시퀀스 레이어에서 강한 비 균일성을 나타내는 시퀀스가 상호 연결된 일련의 작은 하위 시퀀스로 구성되는 시스템이다.

[0163] 비 순차적 메모리 구성은 비 순차적으로 연관된 항목의 정보 저장을 다루며, 여기서 가역성은 비순차적 배열과 비균일한 범위를 나타내고, 여기서 비 균일한 관계는 노드 당 접근적으로 넓은 접근 포인트에 의해 표시되고, 상기 노드의 순서가 셔플될 때 동일한 균일성이 존재하고, 중심 주제 및 조합(Nucleus Topic and Associations)에서는 동일한 일련의 노드가 반복되지만 중심부분(중심 객체)은 다르며, 여기서 상기 중심 부분은 기본 주제를 나타내며, 나머지 노드는 중심부분 주제가 정의되어 있지 않은 경우에 비해 쉽게 접근할 수 있는 메모리 이웃으로 작용하는 시스템이다.

[0164] 메모리 인식(MR)은 알려진 개념을 인식하기 위해 혼돈의 영역을 스캔하고, 여기서 상기 혼돈 영역은 '화이트 노이즈'정보에 임의로 잠긴 개념의 '영역(field)'이며, 메모리 개념 보유(Memor Concept Retention)는 필드 검사를 위해 인덱싱되고 참조될 준비가 된 인식 가능한 개념을 저장하고, 여기서 3 문자 스캐너(3 Letter Scanner)는 혼돈의 영역을 스캔하여 타겟에 해당하는 3개의 문자 세그먼트를 검사하며, 여기서 5 문자 스캐너(5 Letter Scanner)는 혼돈의 영역을 스캔하여 타겟에 해당하는 5개의 문자 세그먼트를 검사하지만, 이번에는 상기 필드의

세그먼트는 전진마다 확인하는 상기 세그먼트는 전체 단어이고, 여기서 상기 혼돈의 영역은 상이한 비율로 스캔하기 위해 분할되고, 여기서 스캔하는 범위의 감소에 따라 정확도가 증가하며, 상기 스캐너의 영역 부분이 증가하면, 더 큰 문자 스캐너는 인식의 수행이 정확도는 떨어지지만 더 효율적이고, 여기서 메모리 개념 인덱싱(MCI; Memory Concept indexing)은 처리되지 않은 메모리 개념이 남아있는 것에 응답하여 스캐너의 크기를 교체하고, 여기서 MCI는 이용할 수 있는 가장 큰 스캐너로 시작하여 점차적으로 감소시켜 보다 작은 메모리 개념 타겟이 존재하는지 확인하기 위해 더 많은 계산을 찾을 수 있는 시스템이다.

[0165] 영역 해석 논리(FIL; Field Interpretation Logic)는 상이한 폭의 스캐너를 관리하기 위한 물류를 운영하고, 일반 범위 스캔(General Scope Scan)은 큰 문자 스캔으로 시작하여 작은 규모의 정확성을 희생하면서 적은 자원으로 넓은 범위의 영역을 탐색하고, 특정 범위 스캔(Specific Scope Scan)은 중요도가 있는 영역을 찾을 때 사용되고, 중복되거나 단단하지 않은 위치에서 값 비싼 정밀 스캔을 수행하지 않도록 줌인(확대)해야 하고, 상기 혼돈 영역에서 메모리 개념에 대한 추가 인식을 수신하는 것은 영역 범위가 메모리 개념들의 고밀도 포화를 포함하는 것을 나타내는 시스템이다.

[0166] 자동화된 인식 발견 메커니즘(APDM; Automated Perception Discovery Mechanism)에서 인식 각도는 전반적인 인식을 구성하는 인식의 여러 측면을 정의하는 범위, 유형, 강도 및 일관성을 비롯한 다수의 메트릭에 의해 구성되어 정의되고, 창조성 모듈은 인식의 복잡한 변형을 생성하고, 상기 인식 가중치는 상기 POE가 에뮬레이션하는 동안 인식이 갖는 상대적 영향의 정도를 정의하고, 두가지 입력 인식의 상기 가중치는 이전 세대의 인식에서 영향을 받은 혼성화된 메트릭을 포함하는 새로 반복된 인식의 가중치를 정의하는 동안 고려되는 시스템이다.

[0167] 상기 CVFG에 대한 입력은 생성된 CVF의 노드 구성에 의해 표시되어야 하는 데이터를 나타내는 임의의 데이터 모음인 데이터 배치(Data Batch)이고, 데이터 배치에 의해 정의된 각각의 개별 유닛을 통해 순차적 전진이 수행되고, 데이터 유닛은 최종 CVF에 의해 참조된 것과 동일한 정보 구성을 갖는 노드 포맷으로 변환되고, 변환된 노드는 단계에서 자신의 존재를 체크할 때 일시적으로 노드 홀드아웃(Node Holdout)에 저장되고, 그들이 발견되지 않으면 발생 및 사용을 포함하는 통계 정보로 생성 및 업데이트되고, 홀드 아웃이 있는 모든 노드가 어셈블링되어 CVF와 같은 모듈 출력으로 푸시되는 시스템이다.

[0168] 노드 비교 알고리즘은 원래의 CVF로부터 읽은 두 개의 노드 구성을 비교하고, 부분 매치 모드(PWN; Partial Match Mode)에서 하나의 CVF에 활성 노드가 있고 비교 후보에서 발견되지 않는 경우(노드가 휴면 상태인 경우), 비교에는 불이익이 없는 시스템이다.

[0169] 시스템 메타 데이터 분리(SMS; System Metadata Separation)는 입력 시스템 메타 데이터를 의미 있는 보안 원인-영향 관계들로 분리하고, 주체 스캔/융합(Subject Scan/Assimilation)을 사용하여 카테고리화 모듈에서 미리 만들어진 카테고리 컨테이너 및 원시 분석을 사용하여 시스템 메타 데이터에서 보안 상태의 주체/의심(subject/suspect)을 추출하고, 상기 주체는 보안 응답/변수 관계를 도출하기 위한 주요 참조 포인트로 사용되고, 위험 스캔/융합(Risk Scan/Assimilation)을 사용하여 보안 상태의 위험 요소가 사전 분류 컨테이너 및 카테고리모듈의 원시 분석을 사용하여 시스템 메타 데이터에서 추출되고, 상기 위험은 그러한 위험을 나타내거나 그러한 위험에 노출된 타겟 주체와 연관되고, 응답 스캔/융합(Rsponse Scan/assimilation)으로, 입력 알고리즘에 의해 만들진 보안 상태의 응답이 카테고리화 모듈로부터 미리 만들어진 카테고리 컨테이너들 및 원시 분석을 사용하여 상기 시스템 메타데이터로부터 추출되고, 상기 응답은 그러한 응답을 받을만한 보안 주체와 관련있는 시스템이다.

[0170] 상기 MCM에서, 포맷 분리(Format Separation)는 인식된 포맷의 규칙 및 구문에 따라 메타 데이터를 구분하고 범주화하고, 로컬 포맷 규칙들 및 구문(Local Format Rules and Syntax)은 상기 MCM 모듈이 미리 포맷된 메타데이터 스트림을 인식할 수 있도록 하는 정의를 포함하고, 디버깅 추적(Debugging Trace)은 사용되는 변수, 함수, 방법 및 클래스와 각각의 입력 및 출력 변수 유형/컨텐츠를 제공하는 코딩 레벨 추적이고, 알고리즘 추적(Algorithm Trace)은 알고리즘 분석과 결합된 보안 데이터를 제공하는 소프트웨어 레벨 추적이며, 결과 보안 결정(승인/차단)은 해당 결정(정당화)에 도달한 경로와 각 요소가 보안 결정을 내리는데 기여한 적절한 가중치와 함께 제공되는 시스템이다.

[0171] 메트릭 처리(MP; Metric Processing)에서, 보안 응답 X는 상기 SPMA가 선택한 결과 보안 응답에 기여하는 일련의 요소들을 나타내고, 초기 가중치는 상기 SPMA에 의해서 결정되며, 인식 추론(PD; Perception Deduction)은 보안 응답의 일부와 해당 시스템 메타데이터를 사용하여 상기 보안 응답의 원래 인식을 복제하고, 차원 시리즈의 인식 해석은 PD가 상기 SPMA의 보안 응답을 취하는 방법과, 상기 SPMA가 원래 사용한 지능형 '디지털 인식'의 전체 범위를 재생성하기 위해 관련있는 입력 시스템 메타 데이터를 연관시키는 방법을 표시하고, 모양 채우

기(Shape Fill), 수량 쌓기(Stacking Quantity), 및 입체화(Dimensional)는 지능형 알고리즘의 상기 '관점'을 포착하는 디지털 인식들인 시스템이다.

[0172] 상기 PD에서, 보안 응답 X는 상기 입력/ 출력 감소(IOR) 모듈의 의도 공급 을 레버리지함으로써 상기 SPMA의 보안 응답에 대한 정당성을 결정하는 정당성/추론 계산(Justification/Reasoning Calculation)에 입력으로 전달되고, 상기 IOR 모듈은 메타 데이터에 열거된 다양한 함수 호출들의 분리된 입력 및 출력을 사용하고, 상기 메타 데이터 분리는 상기 MCM에 의해서 수행되는 시스템이다.

[0173] 상기 POE의 경우, 입력 시스템 메타 데이터는 CVF에서 인식을 생성하기 위해 원시 인식 생성(RP2)에서 사용하는 초기 입력이고, 저장 검색(SS; Storage Search)을 사용하면 데이터 향상 로그에서 파생된 CVF가 인식 저장(PS)의 데이터 베이스 룩업에서 기준으로 사용되고, 랭킹(Ranking)에서 인식은 최종 가중치에 따라 정렬되고, 상기 데이터 강화 로그가 인식에 적용되어 차단/승인 권장사항을 생성하고, 상기 SCKD는 알 수 없는 지식의 예상 상위 범위를 정의하기 위해 상기 로그에 태그를 지정하고, 데이터 파싱(Data Parsing)은 원본 SPMA에 의해 결정된 원래 승인 또는 차단 결정(Approve or Block Decision)을 출력하기 위해 데이터 향상 로그들 및 입력 시스템 메타 데이터의 기본 해석을 수행하고, CTMP는 인식들에 따른 POE의 결정들을 비판하고, 논리적으로 정의된 규칙들에 따른 규칙 실행(RE; Rule Execution)의 결정들을 비판하는 시스템이다.

[0174] 메트릭 복잡도(Metric Complexity)에서, 원의 외부 경계는 개별 메트릭에 관한 알려진 지식의 최고점(peak)를 나타내고, 상기 원의 외부 모서리(edge)는 더 많은 메트릭 복잡도를 나타내지만 중심은 보다 적은 메트릭 복잡도를 나타내고, 중심 옅은 회색은 인식의 적용 각도의 현재 배치의 메트릭 조합을 나타내며, 외부 짙은 회색은 일반적으로 시스템에 의해 저장되고 알려진 메트릭 복잡도를 나타내고, ID의 목표는 관련 메트릭들의 복잡도를 증가시켜 인식의 각도를 복잡도와 수량으로 곱할 수 있고, 상기 짙은 회색 표면 영역은 인식의 적용 각도의 현재 배치의 전체 범위와 알려진 상한 경계에 따라 남겨진 범위의 양을 나타내며, 증강 및 복잡도 강화 시 상기 메트릭들은 메트릭 복잡도로 반환되며 메트릭 변환의 입력으로 전달되고, 메트릭 변환은 개별 및 전체 인식 각도를 역전시켜 최종 출력을 인식의 암시적 각도로 어셈블링하는 시스템이다.

[0175] SCKD의 경우 알려진 데이터 카테고리화(KDC; Known Data Categorization)는 입력으로부터 알려진 정보를 카테고리별로 구분하여 적절한 DB 유추 쿼리를 수행하고, 정보를 카테고리로 구분하고, 분리된 카테고리들은 개별적으로 상기 CVFG에 입력을 제공하고, CVF 포맷의 카테고리 정보를 출력하며, CVF 포맷은 저장 검색(SS)에 의해 알려진 데이터 범위 DB에서 유사성을 체크하기 위해 사용되고, 각 카테고리는 상기 SS 결과들에 따라 알려진 데이터의 관련 범위로 태그되고, 카테고리별로 알려지지 않은 정보의 태그 범위는 알려지지 않은 데이터 콤바이너(UDC; Unknown Data Combiner)에서 원래 입력과 동일한 스트림으로 다시 어셈블링되는 시스템이다.

[0176] 상기 컴퓨터 구현 시스템은 어휘 객관성 마이닝(LOM; Lexical Objectivity Mining)이고,

[0177] a) 질문이 전송되고, 질문에 대한 이해 및 대답/응답에 중요한 누락된 세부사항을 판독하기 위해 중앙 지식 보유(CKR; Central Knowledge Retention)를 레버리지하는, 초기의 쿼리 추론(IQR; Initial Query reasoning);

[0178] b) 상기 질문 및 보충 쿼리 데이터가 전송되고, 인간 주체로부터 입력을 수신하고 인간 주체로 출력을 전송하며, 명확한 질문/주장(Clarified Question/Assertion)을 형성하는, 조사 명확화(SC; Survey Clarification);

[0179] c) 주장 또는 질문의 형태로 제안을 받고, 그러한 제안과 관련된 개념의 출력을 제공하는, 주장 구성(AC; Assertion Construction);

[0180] d) 인간 주체와 합리적인 호소(RA; Rational Appeal) 모두에 대해 상기 AC에 의해 도달된 결론을 제시하기 위한 인터페이스인, 응답 프리젠테이션(Response Presentation);

[0181] e) 관련 개념을 매핑하여 질문/주장 일관성(Question/Assertion consistency)에서 확증 또는 모순을 찾고, 주제에 대해 특정 입장을 취하는데 따르는 이점과 위험을 계산하는, 계층적 매핑(HM; Hierarchical Mapping);

[0182] f) LOM에 대한 지식을 참조하기 위한 상기 주요 데이터 베이스인, 중앙 지식 보유(CKR; Central Knowledge Retention);

[0183] g) 쿼리 능력과 CKR에의 어시밀레이션(assimilation)을 위해 논리적으로 분리되어야 할 높은 신뢰도와 사전 비판된 지식을 수신하는, 지식 유효성 검증(KV; Knowledge Validation);

[0184] h) LOM의 응답을 수락하거나 비판으로 호소하기 위해 인간 주체에게 주어진 선택이고, 응답이 수락되면 KV로 처

리되어 확인된(높은 신뢰도) 지식으로 CKR에 저장될 수 있으며, 인간 주체가 응답을 수락하지 않으면 그들은 인간에 의해 주어진 호소 이유를 확인하고 비판하는 RA로 전달되는, 수락 응답(Accept Response);

[0185] i) 상기 CKR의 마스터 인스턴스로 LOM의 인터넷 클라우드 인스턴스를 실행하고, LOM을 프런트 엔드 서비스들, 백 엔드 서비스들, ㅌ사 어플리케이션 종속성들, 정보 소스들 및 상기 MNSP 클라우드와 연결하는, 관리형 인공지능의 서비스 공급자(MAISP; Managed Artificially Intelligent Services Provider);를 더 포함하는 시스템이다.

[0186] 상기 프런트 엔드 서비스들은 인공 지능형 개인 보조 장치들, 통신 어플리케이션들과 프로토콜들, 홈 자동화 및 의료 어플리케이션을 포함하며, 상기 백 엔드 서비스 들은 온라인 쇼핑, 온라인 운송, 의료 처방전 주문을 포함하고, 상기 프런트 엔드 및 백 엔드 서비스들은 문서화된 API 인프라를 통해 상기 LOM과 상호 작용하므로 정보 전송 및 프로토콜의 표준화를 가능하게 하고, 상기 LOM은 자동화된 연구 메커니즘(ARM; Automated Research Mechanism)을 통해서 외부 정보 소스들로부터 지식을 검색하는 시스템이다.

[0187] 언어적 구성(LC; Linguistic Construction)은 인간 주체와 병렬 모듈의 원시 질문/주장 입력을 해석하여 언어적 구문의 논리적 분리를 생성하고, 개념 발견(CD; Concept Discovery)은 명확한 질문/주장 내에서 관심 포인트를 수신하고, CKR을 활용하여 관련 개념을 도출하고; 개념 우선순위(CP; Concept Prioritization)는 관련 개념을 수신하고 특수성 및 일반성을 나타내는 논리적 계층으로 주문하며; 응답 분리 로직(RSL; Response Separation Logic)은 상기 LC를 레버리지하여 상기 인간 응답(Human Response)을 이해하고, 적절하고 유효한 응답을 초기 설명 요청과 연관시켜 상기 SC의 목적을 달성하고; 상기 LC는 출력 단계 동안 리래버리지(re-leverage)되어 상기 SC에 의해 수신된 상기 보충 정보를 포함하도록 원래의 질문/주장을 수정하고; 문맥 구성(CC; Context Construction)은 주장 구성(AC; Assertion Construction)의 메타 데이터와 상기 인간 주체의 증거를 사용하여 비판적 사고를 위한 CTMP로 원시 사실을 제공하고; 결정 비교(DC; Decision Comparison)는 사전 비판 결정과 사후 비판 결정들 사이의 중첩을 결정하고; 개념 호환성 검출(CCD; Concept Compatibility Detection)은 상기 논리적 호환성 결과를 확인하기 위해 원래의 질문/주장과 함께 개념적 파생어를 비교하고; 이익/위험 계산기(BRC; Benefit/Risk Calculator)는 상기 CCD의 호환성 결과를 수신하고, 개념 구성에 함축된 변수의 변화도(gradients)를 포함하는 균일한 결정을 형성하는 이점과 위험을 고려하고; 개념 상호작용(CI; Concept Interaction)은 조사 설명(SC; Survey Clarification)을 통해 인간 주체로부터 수집된 정보의 일부에 AC개념과 관련된 속성을 할당하는 시스템이다.

[0188] 상기 IQR 내부의 LC는 상기 원래의 질문/주장을 수신하고; 상기 질문은 언어적으로 분리되고 상기 IQR은 상기 CKR을 활용하여 한번에 각 개별적 단어/문구를 처리하며; 상기 CKR을 참조하여, 상기 IQR은 상기 단어/문구의 상기 모호성을 고려하면 가능한 상기 잠재적 옵션을 고려하는 시스템이다.

[0189] 상기 조사 설명(SC)는 IQR로부터 입력을 수신하고, 상기 입력은 도달할 상기 원래의 질문/주장에 대한 객관적 답변을 위해 상기 인간 객체로서 답변해야 하는 일련의 요청된 설명들의 포함하고, 여기서 상기 명확화에 제공된 응답은 상기 응답들이 상기 명확화 요청들과 연관성 있는 응답 분리 로직(RSL; Response Separation Logic)에 전달되고; 상기 요청된 명확화와 병행하여 진행되는, 명확화 언어적 연관성은 상기 LC로 제공되며, 여기서 상기 연관성은 요청된 명확화 및 상기 언어 구조 사이의 상기 내부 관계를 포함하고, 상기 RSL이 상기 원래의 질문/주장을 수정하여 LC가 상기 명확한 질문을 출력할 수 있는 시스템이다.

[0190] 상기 명확한 질문/주장을 수신하는 주장 구성(Assertion Construction)을 위해, LC는 상기 질문을 관심 포인트들로 분해하고, 개념 발견으로 전달하며, 상기 CD는 상기 CKR을 활용하여 관련 개념을 유도하고, 상기 개념 우선순위(CP; Concept Prioritization)는 개념을 논리적 계측으로 순서화하고, 상기 최상위 계층은 가장 일반적 개념이 할당되는 반면 더 낮은 계층은 점진적으로 특정 개념이 할당되고, 상기 최상위 계층은 모듈식 입력으로서 상기 계층적 맵핑(HM; Hierarchical Mapping)으로 전달되고, 정보의 병렬적 전송에서 HM은 의존성 모듈 개념 상호작용(CI; Concept Interaction)에 의해 처리되는 관심의 포인트들을 수신하고, 여기서 상기 CI는 상기 CKR에서 상기 인덱스된 정보를 액세스함으로써 상기 관심의 포인트들로 속성을 할당하고, 상기 HM이 내부 프로세스를 완료하면 파생된 개념들이 호환성을 위해 테스트 되고 입장의 상기 이익들/위험들이 판단되고 반환되어진 후에, 최종 출력이 상기 AC로 반환되어지는 시스템이다.

[0191] 상기 HM에 대해, 상기 CI는 두 개의 개념들 사이에서 상기 호환성/충돌 레벨을 식별하는 CCD로 입력을 제공하고, 상기 호환성/충돌 데이터는 BRC로 전달되며, 이는 상기 호환성들 및 충돌들을 상기 이슈에 관한 전체적인 균일한 태도를 취하는 것에 관한 이익들 및 위험들로 변환하고, 위험/이익 요소들에 따른 상기 태도는 모듈식 출력으로서 AC로 전달하고, 상기 시스템은 상기 질문/주장의 상기 주관적 성격으로 지능의 변화도가 점진

적으로 보충되는 것을 점차적으로 구축된 객관적 응답으로 나타내는 정보 흐름의 루프들을 포함하며; 여기서 상기 CI는 관심 포인트들을 수신하고 우선 순위가 부여된 개념의 최 상위 계층에 따라 각각을 해석하는 시스템이다.

[0192] RA에 대해, 핵심 논리(Core Logic)는 변환된 언어 텍스트를 처리하고 결과를 반환하며, 상기 결과가 높은 신뢰도인 경우 상기 결과는 CKR로 적절한 동화를 위해 지식 검증(KV; Knowledge Validation)에 전달되고, 상기 결과가 낮은 신뢰도인 경우 상기 결과는 자기 비판의 상기 싸이클을 계속하기 위해 AC로 전달되고, 핵심 논리는 언어적 요소들 없이 사전 비판적 결정(Pre-Criticized Decision)의 상기 형태로 LC로부터 입력을 수신하고, 상기 결정은 상기 주관적 의견으로서 CTMP로 전달되고, 결정은 또한 AC로부터의 메타 데이터와 인간 주체로부터의 잠재적 증거를 사용하여 '객관적 사실' 입력으로서 CTMP에 원시 사실을 제공하는 문맥 구성(CC)으로 전달되고, 두 가지 필수 입력을 받은 CTMP와 함께, 그러한 정보는 '객관적 의견'에 도달하는 최고의 시도를 출력하기 위해 처리되며, 여기서 상기 의견은 상기 사후 비판적 결정(Post-Criticized Decision)으로서 RA내에서 내부적으로 취급되고, 사전 비판적 및 사후 비판적 결정들 둘 다 상기 결정 비교(DC; Decision Comparison)로 전달되고, 상기 두 결정들 사이에서 겹쳐지는 상기 범위를 결정하며, 상기 호소 논증은 사실로 인정되거나 상기 호소가 무효한 이유를 성명하기 위해 상기 카운터 포인트가 개선되고, 인정 또는 개선 시나리오와 무관하게, 추후 분석을 위해 높은 신뢰도 결과는 상기 KV로 전달되고, 낮은 신뢰도 결과는 상기 AC로 전달되는 시스템이다.

[0193] 상기 CKR의 경우, 정보의 유닛들은 상기 유닛 지식 형식(UKF; Unit Knowledge Format)에 저장되고, 여기서 규식 구문 형식(RSF; Rule Syntax Format)은 참조 규칙을 추적하기 위한 구문 표준의 세트이며, 상기 RSF내의 규칙의 다수 유닛은 단일 객체 또는 동작을 기술하기 위해 활용될 수 있고, 소스(Source) 속성은 요구된 정보의 소스를 추적하는 복합적 데이터의 집합이고, UKF 클러스터(Cluster)는 관할구역에 따라 분리된 정보를 정의하기 위해 연결된 UKF 변형의 체인으로 구성되며, 여기서 UKF2는 상기 주요 타겟 정보를 포함하고, UKF1은 타임 스탬프 정보를 포함하고, 무한 후퇴를 피하기 위해 타임 스탬프 영역 자체를 생략하고, UKF3은 소스 속성 정보를 포함하고, 무한 후퇴를 피하기 위해 타임 스탬프 영역 자체를 생략하고; 여기서 모든 UKF2는 적어도 하나의 UKF1 또는 하나의 UKF3를 수반해야 하며, 그렇지 않으면 상기 클러스터(시퀀스)는 불완전한 것으로 간주되고, 그 안에 있는 상기 정보는 LOM 시스템 전반의 일반적 로직에서 처리될 수 없고, 상기 중앙 UKF2와 상기 중앙 UKF2에 대응하는 UKF1 및 UKF3 유닛 사이에는 연결된 브릿지로서 동작하는 UKF2 유닛들이 있을 수 있으며, 일련의 UKF 클러스터들은 파생된 주장을 형성하기 위해 KCA에 의해 처리될 수 있고, 지식 확증 분석(KCA; Knowledge Corroboration Analysis)는 UKF 클러스터 정보가 독선적 태도에 관한 확증적 증거를 위해 비교되는 곳이고, 상기 KCA의 처리가 완료된 후, 상기 CKR은 토픽에 관한 완결된 독선적 태도를 출력 할 수 있는 시스템이다.

[0194] ARM에 관한, 사용자 활동에 의해 표시되는 바와 같이, 사용자들이 LOM개념과 상호 작용할 때 질문/주장에 대한 대답/응답과 관련된 직접 뿐만 아니라 간접적으로 가져오고, 사용자 활동(User Activity)은 요청되었지만 이용할 수 없는 개념 목록에 표시된 바와 같이 CKR이 개념들에 관한 정보가 부족하거나 없고, 상기 개념을 최종적으로 산출할 것으로 기대되며, 개념 분류 및 우선순위(CSP; Concept Sorting & Prioritization)에서, 개념 정의들은 세 개의 독립적인 소스로부터 수신되고, 정보 요청의 상기 자원에 우선순위를 부여하기 위해 집계되며, 여기서 상기 정보 소스에 의해 제공된 상기 데이터는 요청된 개념 정의 및 관련된 메타 데이터가 유지되는지에 따라 정보 수집기(IA)에서 수신되고 파싱되며, 여기서 상기 정보는 상기 수신된 정보는 CKR로부터 선재 지식을 고려하여 비교 되어지고 구성되는 상호 참조 분석(CRA; Cross-Reference Analysis)으로 전송되는 시스템이다.

[0195] 개인적인 지능 프로필(PIP; Personal Intelligence Profile)은 각각의 개인적 정보가 다수의 잠재적 엔드 포인트들 및 프런트 엔드들을 통해 저장되는 곳이고, 여기서 상기 정보는 CKR로부터 격리되지만 LOM 시스템 전반의 일반적 로직을 이용 가능하고, 인공 지능과 관련된 개인 정보 어플리케이션은 UKF 형식의 상기 개인 UKF 클러스터 풀에 암호화 및 저장되고, 정보 익명화 프로세스(IAP; Information Anonymization Process)에서 정보는 임의의 개인 식별 정보를 제거한 후에 CKR에 보충되며, 상호 참조 분석(CRA; Cross-Reference Analysis)에서 수신된 정보는 CKR로부터의 선재 지식을 고려하여 비교되고 구성되는 시스템이다.

[0196] 생활 관리 및 자동화(LAA; Life Administration & Automation)는 인터넷 기반 장치 및 서비스를 응집력있는 플랫폼에 연결하며, 적극적 의사 결정(ADM; Active Decision Making)은 자금 조달 규칙 및 관리(FARM ; Fund Appropriations Rules & Management)에 따라 프런트 엔드 서비스, 백 엔드 서비스, IoT 장치, 지출 규칙 및 사용 가능한 금액을 고려하여 가용성 및 기능을 고려하고; FARM은 ADM에 활동의 관할권을 알리기 위해 모듈에 기준, 한계 및 범위를 정의하는 인간의 입력을 받고, 암호화된 화폐 자금이 디지털 지갑에 지불되고, 상기 IoT 상호 작용 모듈(IIM; IoT Interaction Module)은 어떤 IoT 장치가 이용 가능한지의 데이터 베이스를 유지하고,

IoT 이용 가능한 장치가 LAA에 정보를 전송할 때 데이터 피드들(Data Feeds)은 표시하는 시스템이다.

[0197] 비윤리 및/또는 불법 자료를 확인하기 위해 사용자의 개인적 식별 가능 데이터 요청을 모니터링하는 행동 모니터링(BM; Behavior Monitoring)을 더 포함하고, 메타 데이터 집계(MDA; Metadata Aggregation)에서 데이터와 관련하여 사용자 관련 데이터는 상기 사용자의 디지털 신원(identity)이 설정될 수 있도록 외부 서비스로부터 집계되고, 상기 정보는 유도/추론 및 최종적으로 PCD로 전달되며, PCD는 정교한 분석이 상기 MNSP로부터 보강 요소들과 함께 수행되는 곳이고, PIP로 예정된 상기 인증된 사용자의 모든 정보는 정보 추적(IT; Information Tracking)을 거쳐 상기 행동 블랙리스트(Behavior Blacklist)와 대조되며, 프리 크라임 검출(PCD; Pre-Crime Detection)에서 추론 및 유도 정보를 병합하여 프리 크라임 결론을 분석하고, 여기서 PCD는 유도 및 추론에 의해 만들어진 태도를 검증하기 위한 상기 행동 블랙리스트를 직접 참조하는 CTMP를 사용하고, 상기 블랙 리스트 유지관리 기관(BMA; Blacklist Maintenance Authority)은 MNSP의 클라우드 서비스 프레임 워크 내에서 작동하는 시스템이다.

[0198] 상기 LOM은 개인의 라이프에 대한 개인화된 포트폴리오를 관리하도록 구성되고, 여기서 상기 LOM은 LOM의 내부 심의 프로세스(LOM's Internal Deliberation Process)를 통해 결론을 이끌어내는 초기 질문을 수신하며, LOM이 데이터를 수신 및 제어할 수 있는 인터넷 가능 장치에 연결하는 LAA 모듈에 연결하도록 연결되고, 맥락화에서 상기 LOM은 논증을 구성 시 누락 된 링크를 추론하고, 상기 LOM은 원래 주장으로 제기된 딜레마를 해결하기 위해 먼저 상황에 대한 특정 변수를 알거나 가정해야 한다는 논리를 가진 해독기가 있는 시스템이다.

[0199] 상기 컴퓨터 구현 시스템은 선형 원자 퀀텀 정보 전달(LAQIT; Linear Atomic Quantum Information Transfer)이며,

[0200] a) 논리적으로 구조화된 구문 내에서 동일한 일관된 색상 시퀀스를 재귀적으로 반복하는 것; 및

[0201] b) 영어 알파벳으로 번역하기 위해 재귀적으로 상기 시퀀스를 사용하는 것;

[0202] 을 포함하고,

[0203] 역기서 알파벳의 '기본' 레이어를 구성 할 때, 색상 시퀀스는 색상 채널의 단축 및 불평등 가중치와 함께 사용되며 색상 채널 내 구문 정의를 위한 남은 공간은 향후 사용 및 확장을 위해 남겨두고;

[0204] 여기서 복잡한 알고리즘이 그의 로그 이벤트 및 상태 리포트들을 상기 LAQIT로 보고하고, 상태/로그 리포트들은 자동 생성되며, 여기서 상태/로그 리포트들은 전송 가능한 텍스트 기반 LAQIT 구문으로 변환되고, 구문적으로 안전하지 않은 정보가 디지털 방식으로 전송되며, 상기 전송 가능한 텍스트 기반 구문은 고도로 판독 가능한 LAQIT 시각적 구문(선형 모드)로 변환되고, 키(Key)는 인간의 기억을 위해 최적화되며 상대적으로 짧은 모양의 시퀀스에 기초되고;

[0205] 상기 위치상으로 안전하지 않은 텍스트는 수신인에게 제출하기 위해 발신인이 입력하고, 텍스트가 전송 가능한 암호화된 텍스트 기반 LAQIT 구문으로 변환되며, 구문적으로 안전한 정보가 디지털 방식으로 전송되어 데이터가 시각적으로 암호화된 LAQIT 구문으로 변환되고;

[0206] 여기서 증가하는 인식 효과(IRE; Incremental Recognition Effect)는 정보 전달 채널이며, 전부 전달되기 전에 정보 유닛의 상기 전체 형식을 인식하고, 여기서 예측 지수의 상기 효과는 단어와 단어 사이의 상기 전환을 표시함으로써 통합되고, 여기서 사전 인식 효과(PRE; Proximal Recognition Effect)는 정보 전달 채널로, 정보가 손상. 혼합 또는 변경되는 동안 정보 유닛의 상기 전체 형식을 인식하는 시스템이다.

[0207] LAQIT의 선형 모드에서, 블록은 선형 모드의 '기본 렌더링'버전을 표시하고, 포인트는 암호화의 부재를 표시하며, 단어 분리기(Word Separator)에서 상기 도형의 상기 색은 상기 단어에 따른 상기 문자를 나타내며 상기 단어와 그 다음 단어 사이의 분리 역할을 하고, 단일 뷰잉 존(Single Viewing Zone)은 더 큰 문자를 갖는 더 작은 뷰잉 구역을 통합하고 따라서 픽셀 당 정보가 더 적으며, 더블 뷰잉 존(Double Viewing Zone)에서는 픽셀 당 더 많은 활성화 문자가 있고, 음영 커버(Shade Cover)는 관찰자의 주요 초점이 뷰잉 존(viewing zone) 상에 있도록 들어오고 나가는 문자를 둔하게 만드는 시스템이다.

[0208] 암호화 레벨의 넓은 범위를 수용할 수 있는 원자 모드에서, 상기 기본 주요 문자 참조는 어떤 문자가 정의되는지에 대한 일반 사항을 명시 할 것이며, 상기 키커(Kicker)는 상기 베이스와 동일한 색 범위를 갖고 존재하며, 상기 특정 문자를 정확히 정의하며, 판독 방향(Reading Direction)으로 상기 정보 전달 판독은 상기 오비탈 링의 상부 사각형에서 시작하고, 일단 오비탈 링이 완료되면, 다음 순차 오비탈 링의 상부 사각형으로부터 판독이 계속되고, 여기서 상기 엔트리/출구 포털은 문자(기본)의 생성 및 파괴의 상기 포인트들이며, 관련 오비탈에 속

하는 새로운 문자가 상기 포털로부터 나오고 시계 방향으로 그 위치로 내려가며, 상기 원자 핵(Atomic Nucleus)은 그 다음 단어의 문자를 정의하고;

[0209] 단어 네비게이션(Word Navigation)에서, 각각의 블록은 화면의 좌측에 전체 단어(또는 분자 모드의 다중 단어)를 나타내며, 단어가 표시 될 때, 각각의 블록은 오른쪽의 바깥쪽으로 이동하고, 단어가 완료되면 블록 후퇴되고, 상기 네비게이션 블록의 상기 컬러/모양은 상기 단어의 상기 첫 번째 문자를 바탕으로 동일한 컬러/모양이고; 문장 네비게이션에서, 각각의 블록은 단어들의 클러스터를 나타내고, 클러스터는 단어 네비게이션 평면에 맞출 수 있는 단어의 최대량이며, 원자 상태 생성(Atomic State Creation)은 증가하는 인식 효과(Incremental Recognition Effect, IRE)를 유도하는 과정이며, 상기 과정으로 상기 베이스들(Bases)이 엔트리/출구 포털에서 나오고, 키커들이 숨겨지며, 시계 방향으로 움직여서 위치를 파악하고;

[0210] 여기서 원자 상태 확장(Atomic State Expansion)은 상기 근위 인식 효과 (PRE; Proximal Recognition Effect)를 유도하는 과정이고, 일단 상기 베이스들(Bases)이 자신의 위치에 도달하면, 정보 상태 프리젠테이션의 '확장'시퀀스에서 바깥쪽으로 이동하여 정보 상태의 특정 정의를 나타낼 수있는 키커를 나타내고; 원자 상태 파괴(Atomic State Destruction)는 베이스가 들어간 증가하는 인식 효과(Incremental Recognition Effect, IRE)를 유도하는 과정으로, 베이스가 키커들을 다시 덮기 위해 수축(확장 시퀀스의 반전)되고, 여기서 상기 엔트리/출구 포털에 도달하기 위해 시계 방향으로 내려가는 시스템이다.

[0211] 모양 난독화(Shape Obfuscation)에서, 표준 사각형은 5개의 시각적으로 구별되는 모양으로 대체되고, 구문 내에서의 모양의 변화는 위조(가짜) 문자가 원자 프로 파일의 전략 포인트에 삽입되도록 하고, 상기 위조 문자는 상기 메시지의 상기 진짜 및 의도된 의미를 혼란스럽게 하며, 여기서 문자가 진짜인지 위조인지를 해독하는 단계는 안전하고 일시적으로 전송된 해독 키를 통해 행해지고, 리디렉션 본드(Redirection Bonds)에서 본드가 두 문자를 연결하고 읽기의 흐름을 변경하며,

[0212] 전형적인 시계 방향의 읽기 패턴으로 시작하는 동안 합법적/ 비 위조 문자에 대해 출발(시작) 및 도착(끝)하는 본드를 접하는 것은 도착 문자에서 다시 읽기 패턴이 전환되고, 방사성 원소들(Radioactive Elements)에서 어떤 요소들은 문자가 위조인지 아닌지에 대한 상기 평가를 역행할 수 있는 '작동' 할 수 있으며, 여기서 모양은 암호화에 사용할 수 있는 상기 모양을 보여주며, 중심 요소는 단어 바로 다음에 오는 문자를 정의하는 상기 오비탈의 중심 요소를 나타내는 시스템이다.

[0213] 상기 리디렉션 본즈에서 상기 본드는 위조일 수 도 있고 아닐 수 도 있는

[0214] '출발하는'문자에서 시작하고'도착하는'문자로 끝나며, 위조가 없는 경우, 상기 본드는 판독 방향 및 위치를 변경하며, 하나 또는 모두가 위조라면, 전체 본드는 무시되어야 하고, 그렇지 않으면 상기 메시지가 부정확하게 암호 해독 될 수 있으며, 본드 키 정의(Bond Key Definition)에서, 정보 상태를 판독할 때 보드를 준수해야 하는 경우 암호화 키에 구체적으로 정의되었는지 여부에 따라 달라지는 시스템이다.

[0215] 단일 클러스터에서, 두 이웃은 비-방사성이므로, 클러스터의 범위가 정의되고, 상기 키가 이중 클러스터들을 유효하게 지정하기 때문에 처리되기 위한 상기 요소는 우선 방사성이 아닌 경우이고, 이중 클러스터에서 키 정의는 이중 클러스터를 활성으로 정의하므로 다른 모든 크기의 클러스터는 상기 메시지를 암호 해독하는 동안 휴면 상태로 간주되며, 잘못된 해석(Incorrect Interpretation)은 해석자가 역순(긍정 오류)으로 상기 더블 클러스터를 처리하지 않는 방법을 보여주는 시스템이다.

[0216] 암호화 및 스트리밍을 가능하게하는 분자 모드에서, 비밀의 사전 공격 저항(Covert Dictionary Attack Resistance)으로, 상기 메시지의 잘못된 암호 해독은 '붉은 청어'대체 메시지로 이끌고, 분자 당 다수의 활성 단어(Multiple Active Words)에서, 상기 단어는 분자 절차 동안 병렬로 제시되어, 일관된 전환 속도로 표면적 비율 당 정보를 증가 시키고, 바이너리 및 스트리밍 모드는 스트리밍 모드를 나타내지만, 전형적인 원자 구성에서 판독 모드는 바이너리이고, 바이너리 모드 (Binary Mode)는 중심 요소가 단어 뒤에 오는 문자를 정의한다는 것을 나타내며, 여기서 분자 모드는 또한 바이너리이고; 암호화가 가능한 경우를 제외하고는 스트리밍 모드를 고수하고, 여기서 스트리밍 모드가 특수 문자의 오비탈을 참고하는 시스템이다.

[0217] 상기 컴퓨터 구현 시스템은 통합 노드를 부착한 베이스 연결 조화를 갖는 범용 BCHAIN 모든 연결들(UBEC; Universal BCHAIN Everything Connections) 시스템이고,

[0218] a) 하드웨어 인터페이스와 상호작용한 후, 다른 BCHAIN 노드들과 통신하도록 하는 BCHAIN 노드 기본 알고리즘인, 통신 게이트웨이(CG; Communications Gateway);

[0219] b) 원격 노드 행동 패턴을 해석하는, 노드 통계 조사(NSS; Node Statistical Survey);

[0220] c) 노드 이웃이 인식 노드의 주변을 벗어날 가능성을 추적하는, 노드 이스케이프 인덱스(Node Escape Index);

[0221] d) 인식 노드의 탐지 범위에서 노드의 양을 추적하는, 노드 포화 인덱스(Node Saturation Index);

[0222] e) 인식 노드에 의해 해석되는 노드 서비스의 품질을 추적하고는 노드 일관성 인덱스(Node Consistency Index)로, 높은 노드 일관성 인덱스는 주변 이웃 노드가 더 높은 가용성 시간 및 성능 일관성을 갖는 경향이 있음을 나타내고, 이중 용도를 사용하는 노드는 일관성 지수가 낮아지는 경향이 있으며, BCHAIN 네트워크에 전용인 노드들은 더 높은 값을 나타내고; 및

[0223] f) 인식 노드에 의해 해석되는 것으로, 다른 노드에서 중첩 노드들의 양을 추적하는, 노드 중복 인덱스(Node Overlap Index);

[0224] 를 더 포함하는 시스템이다.

[0225] 시스템에 있어서,

[0226] a) 상기 노드에 의해서 이전에 등록된 앱 체인 또는 마이크로 체인으르 포함하는 사용자 정의 체인과 연결하는, 사용자 정의 체인 인식 모듈(CRM; Customchain Recognition Module)로, 상기 CRM은 상기 메타 체인 또는 마이크로 체인의 메타체인 에뮬레이터의 앱 체인 에서 업데이트가 탐지되면, 상기 나머지 BCHAIN 프로토콜를 알리고;

[0227] b) 유효한 CCR을 수신한 후에 상기 요청의 실행을 위해 상기 관련된 CCF를 전송하는, 컨텐츠 클레임 제공(CCD; Content Claim Delivery);

[0228] c) 상기 전략 생성 모듈(Strategy Creation Module)을 관리하고, 창의성 모듈을 사용하여 다양한 전략 선택 알고리즘(OSSA; Optimized Strategy Selection Algorithm)을 통해 시스템이 선호하는 복잡한 전략을 혼합하여 동적으로 생성하는 동적 전략 적응(Dynamic Strategy Adaptation)으로, 여기서 새로운 전략들은 혼돈 영역 해석에서 제공한 입력에 따라 달라지고;

[0229] d) UBEC 플랫폼 인터페이스(UPI; UBEC Platform Interface) 하에서 그래픽 사용자 인터페이스(GUI; Graphical User Interface)에 의해 관리되는 다양한 경제적 특성을 가진, 암호화 디지털 경제 교류(CDEE; Cryptographic Digital Economic Exchange);로 여기서 속성 A에서 노드 리소스는 오직 사용자가 소비하는 것과 일치하도록 소비되며, 여기서 속성 B는 이익 마진이 미리 결정된 값보다 큰 한 최대한 많은 리소스를 소비하며, 속성 C는 거래 통화를 통해 작업 유닛에 대해 비용을 지불하고, 속성 D는 노드 자원이 컨텐츠 또는 통화 보상의 소비 여부에 관계없이 가능한 한 많은 비용을 지출하는 것을 제한하지 않고 지출되고;

[0230] e) 작업 완료 신용과 관련하여, 이 노드의 현재 잉여 또는 적자를 결정하기 위해 메타 체인의 인프라 경제 섹션을 참조하는, 현재 작업 상태 해석(CWSI; Current Work Status Interpretation);

[0231] f) 현재 더 많은 작업이 수행되어야 하는지를 평가하기 위해, 상기 현재 작업 잉여/적자와 함께 선택된 경제적 속성이 고려되는, 경제적 측면을 고려한 작업 시행(ECWI; Economically Considered Work Imposition); 및

[0232] g) 코드 목적을 이해하여 알고리즘 소스를 향상시키는 LIZARD로 구성된 서로 다른 알고리즘 간의 3중 관계로서, 수 세대에 걸친 가상 프로그램 반복을 모방할 수 있는 I2GE 및 분산형 방식으로 복잡한 데이터 집약적 프로그램을 시행할 수 있는 차동적으로 연결된 노드의 광대한 네트워크인 BCHAIN 네트워크는를 포함하는, 공생적 재귀 지능 향상(SRIA; Symbiotic Recursive Intelligence Advancement);을 더 포함하는 시스템이다.

발명의 효과

[0233] 상기한 바와 같은 본 발명의 컴퓨터 보안 시스템은 LIZARD는 코드의 목적을 이해함으로써 알고리즘의 소스 코드를 개선할 수 있고, I2GE는 가상 프로그램 반복 세대를 에뮬레이션 할 수 있으므로 가장 강력한 프로그램 버전을 선택할 있고, BCHAIN 네트워크는 복잡한 데이터가 많은 프로그램을 분산된 방식으로 실행할 수 있는 효과가 있다.

도면의 간단한 설명

[0234] 본 발명은 다음의 도면과 관련하여 상세한 설명을 참조함으로써 더욱 완전히 이해 될 것이다.

도 1 내지 도 26은 CIPR/CTIS로 알려진 클라우드 및 계층화된 정보 보안을 통한 비판적 인프라 보호 및 보상을 보여주는 도면이다. 보다 상세하게,

도 1과 도 2는 보안 해석의 다수 관점들에 대한 정의들을 분석을 위한 방법론으로 제시하기 위한 방법을 보여주는 도면이다.

도 3은 보안 EI^2(엑스트라넷, 인트라넷, 인터넷) 네트워킹을 위한 클라우드 기반 관리형 암호화 보안 서비스 아키텍쳐를 보여주는 도면이다.

도 4 내지 도 8은 관리형 네트워크 및 보안 서비스 제공자(MNSP)의 개요를 나타내는 도면이다.

도 9는 LIZARD 클라우드 기반 암호화 보안에 관한 실시간 보안 처리를 나타내는 도면이다.

도 10은 에너지 시스템에서 CTIS 사례를 통한 CIPR을 보여주는 도면이다.

도 11은 단계 1 - 초기의 시스템 침입을 도시하는 도면이다.

도 12는 단계 2 - 초기의 트로이 목마 배치를 도시하는 도면이다.

도 13은 단계 3 - 고급 실행 멀웨어의 다운로드를 도시하는 도면이다.

도 14는 단계 4 - 침입 방어/예방 시스템의 손상을 도시하는 도면이다.

도 15는 해커가 원하는 행동 및 실제 보안 응답을 도시하는 도면이다.

도 16은 스케줄링된 내부 인증 프로토콜 액세스(SIAPA)를 보여주는 도면이다.

도 17은 루트 레벨 액세스 및 기준 레벨 액세스를 보여주는 도면이다.

도 18은 관리 검토를 보여주는 도면이다.

도 19는 반복적인 인텔리전스 성장/ 인텔리전스 발전(I^2GE)을 도시하는 도면이다.

도 20은 인프라 시스템을 보여주는 도면이다.

도 21은 범죄 시스템, 인프라 시스템 및 공공의 인프라를 보여주는 도면이다.

도 22 내지 도 23은 외부 코드 재작성이 잠재적으로 탐지되지 않는 악의적인 공격을 완화하기 위해 외부 코드를 처음부터 구문적으로 재생산하는 방법을 나타내는 도면이다.

도 24 내지 도 25는 재귀적 디버깅이 코드 세그먼트들을 어떻게 루프하는지 나타내는 도면이다.

도 26은 니드 맵 매칭의 내부 동작을 도시하는 도면이다.

도 27 내지 도 42는 사이버 공간에서 비밀 동작을 통해 기계 비밀 정보(MACINT) 및 응징을 나타내는 도면들이다. 보다 상세하게,

도 27은 지능화된 정보 관리, 뷰잉 및 제어를 나타내는 도면이다.

도 28은 행동 분석에 의한 동작을 나타내는 도면이다.

도 29 및 도 30은 범죄 시스템 및 범죄 시스템에 대한 응징을 나타내는 도면이다.

도 31 및 도 32는 MACINT의 흐름을 나타내는 도면이다.

도 33은 MACINT 비밀 동작 개요 및 범인이 어떻게 기업 시스템을 이용하는지를 나타내는 도면이다.

도 34는 빅 데이터를 이용한 장기/ 딥 스캔의 세부 사항을 나타내는 도면이다.

도 35는 임의의 컴퓨터가 신뢰 플랫폼에서 어떻게 검색되는지를 나타내는 도면이다.

도 36은 신뢰 플랫폼으로부터 알려진 이중 또는 삼중의 에이전트가 포렌식 조사를 더 진행하기 위해 어떻게 참여해야 하는지를 나타내는 도면이다.

도 37은 신뢰 플랫폼이 ISP APIs를 사용하는 방법을 나타내는 도면이다.

도 38은 소프트웨어 및 하드웨어 공급업체가 설정된 백도어를 사용하기 위해 제공하는 보안 API를 활용하기 위

해 신뢰 플랫폼을 사용하는 방법을 보여 주는 도면이다.

도 39 내지 도 41은 임의 및 범죄의 컴퓨터들에 사용자 정의 공격이 적용되는 방법을 나타내는 도면이다.

도 42는 범죄 시스템을 모니터링 하기 위한 신뢰 플랫폼에 장기 우선 순위 플래그가 어떻게 푸쉬되는지 보여주는 도면이다.

도 43 내지 도 68은 논리적으로 추론된 제로-데이터 베이스 선험적 실시간 방어(LIZARD)를 도시하는 도면이다. 보다 상세하게,

도 43 및 도 44는 LIZARD의 의존 구조를 보여주는 도면이다.

도 45는 LIZARD의 개요를 보여주는 도면이다.

도 46은 LIZARD에 관한 주요 알고리즘 함수들의 개요를 도시하는 도면이다.

도 47은 정적 코어(SC)의 내부 동작을 도시하는 도면이다.

도 48은 내부 코어가 시스템의 필수적인 코어 함수들을 어떻게 가지고 있는지를 도시하는 도면이다.

도 49는 동적 쉘(DS)의 내부 동작을 도시하는 도면이다.

도 50은 동적 쉘상의 모듈들을 지능적으로 수정, 생성 및 파괴하는 반복 모듈을 도시하는 도면이다.

도 51은 보안 개선을 위한 반복 코드의 주된 논리인 반복 코어 (Iteration Core)를 나타내는 도면이다.

도 52 내지 도 57은 차동 수정 알고리즘 (DMA)의 논리 프로세스를 나타내는 도면이다.

도 58은 가상 난독화의 개요를 나타내는 도면이다.

도 59 내지 도 61은 가상 난독화의 모니터링 및 응답 측면을 나타내는 도면이다.

도 62 및 도 63은 의심되는 개체로부터 업로드되고 의심되는 개체로 다운로드 된 모든 정보를 추적하는 데이터 회수 추적을 나타내는 도면이다.

도 64 및 도 65는 데이터 회수 트리거의 내부 동작을 도시하는 도면이다.

도 66은 매우 민감한 데이터를 필터링하고, 실제 데이터를 모의 데이터와 혼합하는 데이터 선택을 나타내는 도면이다.

도 67 및 도 68은 행동 분석의 내부 동작을 나태나는 도면이다.

도 69 내지 도 120은 비판적 사고의 기억과 인식(CTMP)을 나타내는 도면이다. 보다 상세하게,

도 69는 CTMP의 주요 논리를 나타내는 도면이다.

도 70은 인식의 관점을 나타내는 도면이다.

도 70 내지 도 73은 CTMP의 의존 구조를 나타내는 도면이다.

도 74는 CTMP에서 지능형 정보를 처리하기 위한 최종 논리를 나타내는 도면이다.

도 75는 CTMP를 대표하는 단일 터미널 출력으로의 직관적/ 지각적 및 사고적/ 논리적 동화의 두 가지 주요 입력을 도시하는 도면이다.

도 76은 원래의 선택된 패턴 매칭 알고리즘(SPMA)에서 발생하는 지능적 사고의 범위를 도시하는 도면이다.

도 77은 인식들 및 규칙들을 통해 CTMP로 수행된 비판적 사고에 대해 병행되는 종래의 SPMA를 나타내는 도면이다.

도 78은 종래의 현재 규칙들과 대조하여 정정 규칙들이 어떻게 생성되는지를 나타내는 도면이다.

도 79 및 도 80은 인식 매칭(PM) 모듈을 도시하는 도면이다.

도 81 내지 도 85는 규칙 구문 파생/ 생성을 나타내는 도면이다.

도 86 및 도 87은 규칙 구문 형식 분리(RSFS) 모듈의 동작을 나타내는 도면이다.

도 88은 규칙 실행 파서(RFP)의 동작을 나타내는 도면이다.

도 89 및 도 90은 실행 디버거를 보여주는 도면이다.

도 91은 규칙 확장을 보여주는 도면이다.

도 92 및 도 93은 순차적 메모리 구성을 보여주는 도면이다.

도 94는 비 순차적 메모리 구성을 보여주는 도면이다.

도 95 내지 도 97은 메모리 인식(MR)을 보여주는 도면이다.

도 98 및 도 99는 영역 해석 논리(FIL)를 보여주는 도면이다.

도 100 및 도 101은 자동 인식 발견 메커니즘(APDM)을 보여주는 도면이다.

도 102는 원시 인식 생산(RP2)를 보여주는 도면이다.

도 103은 비교 가능한 가변 형식 생성기(CVFG)를 보여주는 도면이다.

도 104는 노드 비교 알고리즘(NCA)를 나타내는 도면이다.

도 105 및 도 106은 시스템 메타 데이터 분리(SMS)를 보여주는 도면이다.

도 107 및 도 108은 메타 데이터 분류 모듈(MCM)을 보여주는 도면이다.

도 109는 메트릭 처리(MP)를 보여주는 도면이다.

도 110 및 도 111은 인식 추론의 내부 설계를 보여주는 도면이다.

도 112 내지 도 115는 인식 관찰자 에뮬레이터(POE)를 보여주는 도면이다.

도 116 및 도 117은 함축적 파생(ID)를 보여주는 도면이다.

도 118 내지 도 120은 자기 비판적 지식 밀도(SCKD)를 보여주는 도면이다.

도 121 내지 도 165는 어휘의 객관성 마이닝(LOM)을 보여주는 도면이다. 보다 상세하게,

도 121은 어휘의 객관성 마이닝(LOM)을 위한 주요 논리를 보여주는 도면이다.

도 122 내지 도 124는 관리형 인공 지능의 서비스 공급자(MAISP)를 보여주는 도면이다.

도 125 내지 도 128은 LOM의 의존 구조를 보여주는 도면이다.

도 129 및 도 130은 초기의 쿼리 추론(IQR)의 내부 논리를 보여주는 도면이다.

도 131은 조사 설명(SC)를 보여주는 도면이다.

도 132는 주장 구성(AC)를 보여주는 도면이다.

도 133 및 도 134는 계층적 매핑(HM)이 어떻게 작동하는지에 대한 내부 세부 사항을 보여주는 도면이다.

도 135 및 도 136은 합리적 호소(RA)의 내부 세부 사항을 보여주는 도면이다.

도 137 및 도 138은 중앙 지식 보유(CKR)의 내부 세부 사항을 보여주는 도면이다.

도 139는 자동화된 연구 메커니즘(ARM)을 보여주는 도면이다.

도 140은 문체 분석 스캐닝(SS)을 보여주는 도면이다.

도 141은 추정 중단 시스템(AOS)을 보여주는 도면이다.

도 142는 지능형 정보 및 구성 관리(I^2CM) 및 관리 콘솔을 도시하는 도면이다.

도 143는 개인적인 지능 프로필(PIP)를 보여주는 도면이다.

도 144는 생활 관리 및 자동화(LAA)를 보여주는 도면이다.

도 145는 행동 모니터링(BM)을 보여주는 도면이다.

도 146은 윤리적 사생활 보호법(EPL)을 보여주는 도면이다.

도 147은 LIZARD 알고리즘의 개요를 보여주는 도면이다.

도 148은 반복적인 지능 성장을 보여주는 도면이다.

도 149 및 도 150은 반복적인 발전을 보여주는 도면이다.

도 151 내지 도 154는 창조 모듈을 보여주는 도면이다.

도 155 내지 도 156은 개인 보조로 사용되는 LOM을 보여주는 도면이다.

도 157은 연구 도구로 사용되는 LOM을 보여주는 도면이다.

도 158 및 도 159는 제안된 이론의 장단점을 탐구하는 LOM을 보여주는 도면이다.

도 160 및 도 161은 외교 정책 전쟁 게임을 위한 규칙 수립을 수행하는 LOM을 보여주는 도면이다.

도 162 및 도 163은 조사 저널리즘 작업들을 수행하는 LOM을 도시하는 도면이다.

도 164 및 도 165는 역사적 검증을 수행하는 LOM을 도시하는 도면이다.

도 166 내지 도 179는 안전하고 효과적인 디지털-지향 언어 LAQIT를 보여주는 도면이다. 보다 상세하게,

도 166은 LAQIT의 개념을 도시하는 도면이다.

도 167은 사용 가능한 주요 언어 타입을 보여주는 도면이다.

도 168 및 도 169는 LAQIT의 선형 모드를 도시하는 도면이다.

도 170 및 도 171은 원자 모드의 특성을 보여주는 도면이다.

도 172 내지 도 174는 원자 모드의 암호화 특징에 대한 개요를 보여주는 도면이다.

도 175 및 도 176은 재 지정 결합의 메커니즘을 도시하는 도면이다.

도 177 및 도 178은 방사성 원소의 메커니즘을 도시하는 도면이다.

도 179는 암호화 및 스트리밍이 가능한 분자 모드를 도시하는 도면이다.

도 180 내지 도 184는 분산형 정보 분배 시스템 BCHAIN에 연결되는 UBEC 플랫폼과 프런트 엔드의 개요를 도시하는 도면이다. 보다 상세하게,

도 180은 BCHAIN 사용 어플리케이션을 포함하고 실행하는 BCHAIN 노드를 도시하는 도면이다.

도 181은 BCHAIN 프로토콜의 핵심 논리를 도시하는 도면이다.

도 182는 전략 창조 모듈(SCM)을 관리하는 동적 전략 적응(DSA)을 도시하는 도면이다.

도 183은 다양한 경제 속성을 가진 암호화 디지털 경제 교류(CDEE)를 도시하는 도면이다.

도 184는 공생적 재귀 지능 향상(SRIA)을 도시하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0235] **클라우드 및 계층화된 정보 보안(CTIS; Cloud & Tiered Information Security)를 통한 비판적 인프라 보호 및 보상(CIPR; Critical Infrastructure Protection & Retribution)**

[0236] 도 1과 도 2는 보안 해석의 다수 관점들에 대한 정의들을 분석을 위한 방법론으로 제시하기 위한 방법을 보여주는 도면이다. 도면 부호 1에서, 비콘들과 에이전트들의 확립된 네트워크가 공격자와 나쁜 행위자의 맵을 형성하는데 사용된다. 이러한 맵/데이터 베이스를 정교한 예측 알고리즘과 결합하면 잠재적인 사전 범죄(pre-crime) 위협이 발생한다. I^2GE(21)은 빅 데이터와 멀웨어 시그니처 인식을 활용하여 그 요인을 결정한다. 보안 행동(20) 스토리지는 보안 이벤트, 영향 및 적절한 대응의 선례를 형성한다. 이러한 적절한 대응은 CTMP(22)(비판적 사고, 메모리, 인식)가 추가적인 보안 계층으로 비판을 받을 수 있다. 도면 부호 2는 어떤 자산이 위험에 처해 있는지, 어떤 잠재적인 피해가 발생할 수 있는지를 나타낸다. 예를 들면: 수력 발전 댐은 수문을 열었을 때 결국에는 인근 마을에 홍수를 일으켜 생명과 재산을 잃을 수 있다. 인프라 DB(3)은 국가 인프라 작업과 관련된 공

공 또는 민간 회사와 관련된 민감한 정보와 중요하지 않은 정보가 포함된 일반 데이터베이스를 의미한다. 인프라 컨트롤(4)은 댐 수문, 국가의 전기 기관 등 의 전기의 전력량과 같은 산업 인프라 장비를 제어 할 수 있는 잠재적인 기술적, 디지털적 및/ 또는 기계적 수단이다. 도면 부호 5를 참조하면 트래픽 패턴은 잠재적인 사각지대의 시간을 강조하기 위해 분석된다. 그러한 공격은 합법적인 트래픽과 혼합되도록 쉽게 숨길 수 있다. 질문: 나쁜 행위자가 관심을 가질만한 정치/ 금융/ 스포츠/ 기타 사건이 있는가? 신뢰 플랫폼의 외부 에이전트 네트워크는 해커 활동 및 준비를 보고한다. 따라서 공격 타이밍을 예측할 수 있다. 도면 부호 6에서는 다음과 같이 질문된다: 공격의 대상이 될 수 있는 취약한 기업은 누구인가. 주어진 지리적 위치에서 어떤 유형의 기업이 취약한가. 가장 취약한 자산/ 통제 란 무엇이며 무엇이 최선책인가. 신뢰 플랫폼의 외부 에이전트 네트워크는 해커 활동 및 준비를 보고한다. 따라서 공격 위치를 추정할 수 있다. 도면 번호 7에서, 질문이 요구된다: 세계의 지정 학적, 기업적, 재정적 압박이 그러한 공격의 자금 조달과 재촉을 촉진하기 위해 존재하는지 질문한다. 누가 혜택을 보는지, 얼마만큼의 이익을 얻을 것 인가. 신뢰 플랫폼의 외부 에이전트 네트워크는 해커 활동 및 준비를 보고한다. 따라서 공격 동기를 추정 할 수 있다. 도면 번호 8에서, 질문은 다음과 같이 질문된다: 멀웨어에 대한 잠재적인 악용 및 숨어있는 점은 무엇인가. 어떻게 사각 지대와 요새화된 액세스 지점을 사용하여 중요한 자산과 인프라 제어 지점을 손상시킬 수 있나. LIZARD(16)은 외부 코드에서 목적과 기능을 도출 할 수 있으므로 악의적인 의도가 있거나 합법적인 원인이 없는 경우 이를 차단한다. CTMP(22)는 차단/ 승인 결정에 대해 비판적으로 생각할 수 있으며 보완적인 보안 계층 역할을 한다.

[0237] 도 3은 보안 EI^2(엑스트라넷, 인트라넷, 인터넷) 네트워킹을 위한 클라우드 기반 관리형 암호화 보안 서비스 아키텍처를 보여주는 도면이다. 관리형 네트워크 및 보안 서비스 제공자(MNSP)(9)는 에너지, 화학, 핵, 댐 등 주요 인프라 산업 부문에 관리형 암호화 보안(Managed Encrypted Security), 연결 및 규정 준수 솔루션 및 서비스(Connectivity & Compliance Solutions & Services)를 제공한다. 신뢰 플랫폼(Trusted Platform)(10)은 보안 정보 및 서비스를 공유함으로써 상호 이익을 얻는 검증된 회사 및 시스템의 집합체이다. 하드웨어 및 소프트웨어 공급자(11)은 업계에서 인정하는 하드웨어 / 소프트웨어 제조업체 (예: Intel, Samsung, Microsoft, Symantec, Apple 등)이다. 이러한 맥락에서 그들은 신뢰 플랫폼(10)에 제한된 또는 전체 용량으로 백 도어 액세스를 가능하게 하는 제품에 대한 잠재적인 액세스 및/ 또는 개발 수단을 제공한다. 이는 신뢰 플랫폼이 파트너 및 공동 보안 부서와 협력하여 제정하고자 하는 잠재적인 보안 및/ 또는 응징 프로세스에 대해 활성화되었다. 가상 사설망(VPN)(12)은 MNSP(9), 신뢰 플랫폼 및 관련 파트너 간의 안전하고 논리적으로 분리된 통신을 가능하게 하는 업계 표준 기술이다. 엑스트라 넷을 사용하면 디지털 요소를 마치 같은 지역(예: LAN)에 있는 것처럼 사실상 공유할 수 있다. 따라서 이 두 기술의 조합은 신뢰 플랫폼의 운영을 향상시키기 위해 파트너 간의 효율적이고 안전한 통신을 촉진한다. 보안 서비스 제공자들(13)은 디지털 보안 전략 및 솔루션을 제공하는 공개 및/ 또는 비공개 기업 집합체이다. 이 솔루션 / 제품은 계약 기반으로 구성되어 신뢰 플랫폼이 원래의 보안 정보(즉, 새로운 멀웨어 서명) 및 보안 분석의 혜택을 누릴 수 있다. 이러한 보안 강도의 증가는 추가 보안 도구 및 정보에 대한 액세스 권한을 보유하고 있으므로 보안 서비스 제공자 자신에게도 이익이 된다. 제 3 자 위협 정보 (3PTI) 제공(14)은 보안 정보(즉, 새로운 멀웨어 서명)의 상호 공유이다. 신뢰 플랫폼은 이러한 보안 정보를 전송, 수신 및 동화하는 중앙 집중식 허브 역할을 한다. 여러 정보 피드를 사용하면 정보 협력을 식별하는 분석 모듈 (예: 음모 탐지(19))을 통해 고급 보안 패턴(보안 서비스 제공제들을 활용)을 통해 더 진보한 패턴들을 얻을 수 있다. 법 집행(15)은 주(즉, NYPD), 국가(즉, FBI) 또는 국제(즉, 인터폴)이든 관련 법 집행 부서를 지칭한다. 범죄 해커에 대한 응징을 촉진하거나 수행하기 위해 보안 정보를 수신하고 전송하는 통신이 설정된다. 그러한 응징은 일반적으로 적절한 용의자를 찾아 체포하고 관련 법정에서 시도한다.

[0238] 도 4 내지 도 8은 관리형 네트워크 및 보안 서비스 제공자(MNSP)의 개요를 나타내는 도면이다. LIZARD(16)는 이전의 과거 데이터를 참조하지 않고 위협을 분석한다. 인공의 보안 위협 (AST)(17)은 보안 룰 세트들의 효력을 테스트하기 위한 가상적인 보안 시나리오를 제공한다. 보안 위협에 대한 의미있는 비교를 제공하기 위해 심각도 및 유형이 일관성 있게 나타낸다. 창조성 모듈(18)은 이전 형식들로부터 새로운 (Hybrid) 형식을 생성하는 프로세스를 수행한다. 여러 알고리즘을 서비스하기 위해 모듈로 플러그인 모듈로 사용된다. 음모 탐지(Conspiracy Detection)(19)는 다수의 '음모성' 보안 이벤트들의 정기적인 백그라운드 검사를 제공하고, 관련 없어 보이는 보안 이벤트들 사이의 패턴 및 상호 관계를 결정하기 위한 시도한다. 보안 행동(20): 이벤트들과 이벤트들의 보안 응답들 및 특성들을 이후 쿼리를 위해 저장되고 인덱싱 된다. I^{2}GE(21)은 MNSP(9)의 대규모 데이터, 소급 분석 지점이다. 표준 서명 추적 기능 중에서 AST를 창조성 모듈과 함께 사용하여 향후 멀웨어의 잠재적 변형을 에뮬레이트 할 수 있다. CTMP(22)는 여러 소스(예: I^{2}GE, LIZARD, 신뢰 플랫폼 등)의 상호 참조 정보를 활용하고 인식과 현실에 대한 기대치를 학습한다. CTMP는 문제에 대한 객관적인 결정을 내릴 수 있는 자신의 능력이라고

추정하며 내부의 낮은 신뢰도로 결정을 내리는 것을 삼가 한다. 관리 콘솔(MC)(23)은 인간이 복잡한 시스템 및 반 자동 시스템을 모니터링하고 제어하는 지능적인 인터페이스이다. 지능형 정보 및 구성 관리 (I^2CM)(24)에는 정보의 흐름과 허가된 시스템 활용을 제어하는 다양한 기능이 포함되어 있다.

[0239] 에너지 네트워크 교환(Energy Network Exchange)(25)은 에너지 공급 업체, 생산자, 구매자 등을 연결하는 대형 사설 엑스트라 넷이다. 이를 통해 일반 산업과 관련된 보안 정보를 교환 할 수 있다. 에너지 교환 네트워크는 VPN/ 엑스트라 넷(12)를 통해 MNSP 클라우드(9)와 통신한다. 이러한 클라우드 통신은 1) 중요한 보안 정보 데이터가 에너지 네트워크 교환에서 MNSP 클라우드로 제공되고 2) MNSP 클라우드에서 에너지 네트워크 교환으로 중요한 보안 수정 조치가 제공된다는 점에서 양방향 보안 분석이 가능하다. 에너지 회사의 모든 EI^2(엑스트라 넷, 인트라넷, 인터넷) 네트워킹 트래픽은 항상 VPN(12)를 통해 MNSP 클라우드로 라우팅된다. MNSP가 모든 서비스에 대해 사용하는 인증 및 암호화는 국가 표준(예: FedRAMP, NIST, OMB 등) 및 국제 표준 (예: ETSI, ISO/ IEC, IETF, IEEE 등) 및 암호화 요구 사항(예: FIPS 등)을 준수한다. 인트라넷 (26)(암호화된 계층 2/3 VPN)은 기업(에너지 회사) 개인 네트워크(27) 내에서 보안 내부 연결을 유지한다. 이로써 LIZARD 라이트 클라이언트(43)은 기업 인프라 내에서 작동 할 수 있으며 MNSP 클라우드(9)내에 존재하는 LIZARD 클라우드(16)와 안전하게 통신 할 수 있다. 도면 부호 27은 개인 네트워크의 로컬 노드를 나타낸다. 이러한 개인 네트워크는 여러 위치를 제공한다(위치 A, B 및 C로 표시). 서버 클러스터(위치 C) 또는 모바일 장치가 있는 공유 직원의 사무실과 개인 WiFi 연결(위치 A)과 같은 각기 다른 기술 인프라 설정이 각 개인 네트워크 내에 존재할 수 있다. 개인 네트워크의 각 노드에는 자체 관리 콘솔 (MC)(23)이 할당되어 있다. 포터블 미디어 장치들(28)은 개인 네트워크에 안전하게 연결되도록 구성되어 있으며, 따라서 인트라넷(26)을 확장함으로써 보안 VPN/엑스트라넷 연결(12)을 통해 MNSP(9)에 간접적으로 연결된다. 이 보안 연결을 사용하면 모든 트래픽이 MNSP를 통해 라우팅되어 배포된 실시간 및 소급 보안 분석 알고리즘에 대한 최대 노출이 가능하다. 이러한 포터블 장치(휴대용 장치)는 보안 개인 네트워크 내부에서든 공공 커피 숍의 WiFi 액세스에서든 이 보안 연결을 유지할 수 있다. 비무장 지대(DMZ)(29)는 통상의 컴퓨터보다 높은 보안 책임을 갖는 HTTP 서버를 포함하는 서브 네트워크이다. 서버의 보안 책임은 보안 과실이 아니라 공용 서버의 복잡한 소프트웨어 및 하드웨어 구성 때문이다. 보안을 강화하기 위한 최선의 노력에도 불구하고 많은 잠재적 공격이 존재하기 때문에 서버는 DMZ에 배치되어 나머지 개인 네트워크(위치 C)가 이러한 보안 책임에 노출되지 않는다. 이러한 분리로 인해 HTTP 서버는 DMZ 내부에 있지 않은 개인 네트워크 내의 다른 장치와 통신 할 수 없다. LIZARD 라이트 클라이언트(43)은 HTTP 서버에 설치되어 DMZ 내에서 작동 할 수 있다. MC(23)가 HTTP 서버와 DMZ에 액세스 할 수 있도록 DMZ 규칙에서는 예외가 있다. 라이트 클라이언트는 이벤트 12와 26에서 형성된 암호화된 채널을 통해 MNSP와 통신한다. 도면 부호 30을 참조하면, 이들 서버는 개인 네트워크에서 격리되어 있지만 DMZ (29)에 감추어지지 않는다. 이는 개인 네트워크 내의 장치들의 상호 통신을 허용한다. 이들은 각각 LIZARD 라이트 클라이언트(43)의 독립된 사례를 가지고 있으며 MC(23)에 의해 관리된다. 인터넷(31)은 LIZARD 라이트 클라이언트를 실행하는 MNSP(9)와 기업 장치(28) 사이의 정보 전송 매체와 관련하여 참조된다. 인터넷은 근거리 통신망(LAN)에서 발생하는 로컬 상황의 위협과는 달리 기업 장치에 대한 보안 위험 요소가 가장 많이 발생하는 소스이다. 높은 보안 위험 때문에 개별 장치의 모든 정보 전송이 MNSP에 프록시처럼 라우팅된다. 인터넷의 잠재적인 나쁜 행위자는 VPN/ 엑스트라 넷 구조(12)로 인해 암호화된 정보만 볼 수 있다. 제 3자 위협 정보(3PTI) 제공(32)는 기존 계약상 의무에 따라 타사에서 제공한 사용자 지정 조정 정보 입력을 나타낸다. 반복적인 진화(33): 병렬적 진화 경로가 성숙되고 선택된다. 반복적인 세대는 동일한 인공 보안 위협 (AST)에 적응하며 최고의 성격 특성을 가진 통로는 가장 큰 보안 위협에 저항한다. 진화 경로(34): 사실상 포함되어 있고 고립된 일련의 규칙 집합 세대. 진화 특성과 기준은 그러한 경로 성격 X에 의해 정의된다.

[0240] 도 9는 LIZARD 클라우드 기반 암호화 보안에 관한 실시간 보안 처리를 나타내는 도면이다. 구문 모듈(35)는 컴퓨터 코드를 읽고 쓸 수 있는 프레임 워크를 제공한다. 쓰기를 위해; PM으로부터 복잡한 형식화된 목적을 수신한 다음 임의의 코드 구문으로 코드를 작성하고, 헬퍼 함수가 임의의 코드를 실제 실행 코드(원하는 언어에 따라 다름)로 변환 할 수 있다. 읽기 위해; 그러한 코드의 함수를 위한 목적을 도출하기 위해 PM에 대한 구문적 해석을 제공한다. 목적 모듈(36)은 구문 모듈(35)을 사용하여 코드에서 목적을 도출하고 목적을 '복잡한 목적 형식'으로 출력한다. 그러한 목적은 SM에 의해 해석된 대로 코드 블록의 의도된 기능을 적절하게 기술해야 한다(비록 그 코드가 데이터에 은밀하게 포함 된 경우라 할지라도). 가상 난독화(Virtual Obfuscation)(37) 기업 네트워크와 데이터 베이스는 가상 환경에서 복제되고 민감한 데이터는 모의(가짜) 데이터로 대체된다. 대상의 동작에 따라 실시간으로 더 많은 가짜 요소 또는 시스템의 실제 요소를 포함하도록 환경을 동적으로 변경할 수 있다. 신호 모방(38)은 가상 난독화(보호)의 분석적 결론에 도달했을 때 일반적으로 사용되는 응징 형식을 제공한다. 신호 모방기는 구문 모듈을 사용하여 악성 코드의 해커와의 통신 구문을 이해한다. 그런 다음 이러한 통신

을 가로 채서 악성 코드가 해커에게 민감한 데이터를 성공적으로 보낸다는 잘못된 인식을 준다(해커의 허위 사실에 가짜 데이터를 보냈지만). 해커들 또한 상기 LIZARD에 의한 멀웨어의 오류 코드를 전송 받아 상기 오류 코드가 멀웨어로 부터 온 것처럼 보이게 한다. 이로 인해 해커의 시간과 자원이 거짓 디버깅 접선으로 전환되고 결국 제대로 작동하지 않는다는 잘못된 인상으로 작동 중인 멀웨어를 포기한다. 내부 일관성 검사(Internal Consistency Check)(39)는 외부 코드의 모든 내부 기능이 올바른지 확인한다. 전체적으로 외부 코드의 목적과 내부적으로 일치하지 않는 코드 조각이 없는지 확인한다. 외부 코드 재작성(40)은 구문 및 목적 모듈을 사용하여 외부 코드를 복잡한 용도 형식으로 줄인다. 그런 다음 파생된 목적을 사용하여 코드 세트를 작성한다. 이를 통해 외부 코드의 원하는 목적과 이해된 목적만 기업 내에서 실행되며 의도하지 않은 함수 실행이 시스템에 액세스하지 못하도록 한다. 비밀 코드 탐지(41)는 데이터 및 송신 패킷에 은밀하게 삽입된 코드를 검출한다. 니드맵 매칭(42)은 필요성 및 목적의 매핑된 계층이며 시스템의 전체 목표에 외부 코드가 적합한지 결정하기 위해 참조 된다. LIZARD 라이트 클라이언트(43)는 가상의 난독화(208) 및 신호 모방기와 같은 리소스가 많은 기능을 생략한 LIZARD 프로그램의 경량 버전이다. 참조용 서명 데이터베이스를 사용하지 않는 선험적인 위협 분석을 목적으로 최소한의 컴퓨터 리소스 사용으로 즉각적이고 실시간으로 위협 평가를 수행한다. 로그(44)에서 에너지 회사 시스템(48)은 표준 소프트웨어 오류/ 액세스 로그, 운영 체제 로그, 모니터링 프로브 등과 같은 여러 로그 생성 지점을 갖는다. 그런 다음 이러한 로그는 깊이 및 대응 보안 분석을 위해 로컬 패턴 매칭 알고리즘(46) 및 CTMP(22)에 심층적으로 공급된다. 트래픽(45)에서는 에너지 회사의 로컬 패턴 매칭 알고리즘(46)에 존재하는 모든 내부 및 외부 트래픽이 안티 바이러스, 적응 형 방화벽 등과 같은 초기 보안 계층을 제공하는 산업 표준 소프트웨어로 구성된다. 정정 동작(47)은 처음에는 보안 문제/ 위험을 해결하기 위해 이해되는 로컬 패턴 매칭 알고리즘(46)에 의해 수행되어야 한다. 여기에는 포트 차단, 파일 전송, 관리 기능 요청 등이 포함될 수 있다. 에너지 회사에는 로그 및 트래픽 정보를 전송하는 특수 보안 알고리즘으로부터 격리된 시스템(48)이 있다. 이는 LIZARD(16), I^2GE(21) 및 CTMP(22) 알고리즘이 MNSP 클라우드(9)를 기반으로 하기 때문에 발생한다. 이러한 분리는 중앙 집중식 데이터베이스 모델을 제공하여 보안 데이터/ 경향 풀이 더 많아 져서보다 포괄적인 분석을 가능하게 한다.

[0241] 도 11에서, 범죄 시스템은 표적 시스템으로 진입할 수 있는 이용 가능한 채널을 탐색한다. 가능한 경우 작은 페이로드를 전달하기 위해 채널을 손상시킨다. 범죄 시스템(49)는 악의적인 범죄자가 악성 프로그램 공격을 파트너 시스템(51)에 공격을 시작하여, 결과적으로 인프라 시스템(54)을 공격하는데 사용된다. 멀웨어 소스(50)는 악의적인 코드 (멀웨어)의 비활성 형식에 대한 컨테이너이다. 일단 코드가 타겟 인프라 시스템(54)에 도달하면 (또는 도달하려고 시도 할 때), 멀웨어가 사전 지정 또는 주문형 악성 작업을 수행하도록 활성화된다. 파트너 시스템(51)은 인프라 회사(에너지 회사)와 파트너 회사 간의 계약에 따라 인프라 시스템과 상호 작용한다. 이러한 계약은 공급망 관리 서비스 또는 재고 추적 교환과 같은 일종의 비즈니스 이익을 반영한다. 합의된 서비스를 이행하기 위해 양 당사자는 사전 합의 된 보안 표준에 따라 전자적으로 상호 작용한다. 멀웨어 소스(50)은 범죄 시스템(49)를 실행하는 악의적인 당사자를 대신하여 파트너 시스템에서 침입에 대한 악용 사례를 찾으려고 시도한다. 이런 식으로 멀웨어가 인프라 시스템(54)인 감염의 최종 목표에 도달 할 수 있다. 이 방법으로 파트너 시스템은 멀웨어 소스(50)에서 시작된 프록시 감염 프로세스에 사용되었다. 파트너 시스템(51)과 인프라 시스템(54) 간의 많은 통신 채널 중에서, 이 채널(52)은 멀웨어 소스(50)로부터 파생된 멀웨어에 의해 손상되었다. 채널/ 프로토콜 53: 파트너 시스템(51)과 인프라 시스템(54) 사이의 아직 해결되지 않은 통신 채널을 보여준다. 이러한 채널은 파일 시스템 접속, 데이터베이스 접속, 이메일 라우팅, VOIP 접속 등을 포함할 수 있다. 인프라 시스템(54)은 인프라 DB(57) 및 인프라 컨트롤(56)에 직접 액세스하는 에너지 회사의 운영의 중요한 요소이다. 산업 표준 침입 방어 시스템(55)는 표준 보안 절차로 구현된다. 인프라 컨트롤(Infrastructure Control)(56)은 에너지 관련 장비에 연결되는 디지털 인터페이스이다. 예를 들어, 이는 수력 발전 댐의 수문의 개폐 및 태양 전지판 어레이가 향하는 방향 등을 포함 할 수 있다. 인프라 데이터베이스(57)는 인프라 시스템과 에너지 회사의 핵심 작동과 관련된 민감한 정보를 포함한다. 그러한 정보에는 연락처 정보, 직원 교대 추적, 에너지 장비 문서화 및 청사진 등이 포함될 수 있다.

[0242] 도 12에서 손상된 채널(52)는 이용(악용)을 위해 매우 좁은 기회의 창을 제공하므로, 이용 기회를 확장하기 위해 매우 간단한 트로이 목마가 대상 시스템에 업로드된다. 트로이의 목마(58)은 멀웨어 소스(50)에서 유래하여 손상된 채널(52)를 통해 이동하여 인프라 시스템(54)를 목표로 한다. 악의적인 공격을 통해 기회를 열어 고급 실행 멀웨어 페이로드(더 복잡하고 데이터를 훔치는 실제 악성 코드 포함)가 대상 시스템에 설치될 수 있다.

[0243] 도 13은 트로이 목마가 시스템을 추가로 사용한 후 대규모 실행 가능 멀웨어 패키지가 트로이 목마에 의해 생성된 새로운 열린 채널을 통해 시스템에 안전하게 업로드하는 방법을 보여준다. 고급 실행 가능 멀웨어(59)이 인

프라 시스템(54)으로 전송되므로 민감한 데이터베이스(57) 및 컨트롤들(56)로 전송된다. 고급 실행 가능 멀웨어는 트로이 목마의 이전 공격에서 만든 디지털 경로를 사용하여 목적지에 도달한다.

[0244] 도 14는 고급 실행 가능 멀웨어(50)가 IDS를 손상시켜 민감한 인프라 정보 및 컨트롤 포인트를 탐지되지 않는 범죄 시스템에 개별적으로 다운로드 할 수 있는 방법을 도시한다. 해커가 원하는 행동(60), 해커(65)는 합법적으로 인증된 액세스 자격 증명을 가진 회사 직원의 신뢰할 수 있는 자격 증명을 확보했다. 해커는 이러한 자격 증명을 사용하여 직원 전용으로 만 계획된 LAN에 대한 신중하고 눈에 띄지 않는 액세스 권한을 얻으려고 한다. 해커는 전형적인 "너무 적거나 너무 늦은" 보안 대응책의 범위를 벗어나려고 한다. 엔드 포인트 보안 클라이언트가 클라우드 보안 서비스에 전송하는 경우에도, 소급 분석적 보안 솔루션은 초기 침입에서 위협을 제거하고 관리하는 것이 아니라 손상 통제만 실시간으로 관리할 수 있다. 실제 보안 응답(61)을 통해 LIZARD 라이트 클라이언트(엔드 포인트 사용)는 자격 증명 로그인 및 시스템 액세스 사용의 필요성, 기능 및 목적을 명확하게 증명할 수 없다. 이 사용자가 실제로 자격 증명을 의도하고 합법적인 사용자인지 여부는 아직 알 수 없으므로 사용자는 부분적으로 가상화된/매크 환경에 배치된다. 이러한 환경은 사용자의 행동을 분석할 때 실시간으로 중요한 데이터에 대한 노출을 동적으로 변경할 수 있다. 행동 분석(62)은 실제 및 가상 복제된 LAN 인프라(64) 상에 존재하는 요소들과 상호 작용하는 요소들에 기초하여 해커(65)상에서 수행된다. 해커는 손상된 자격 증명(63)으로 에너지 회사의 랩탑(28)에 대한 관리자 액세스를 허용하는 액세스 자격 증명을 얻었으므로, 랩톱으로 구성된 LAN 인프라(64)에 연결된다. 이러한 자격 증명은 암호화되지 않은 전자 메일을 가로 채고 로컬에 자격 증명이 저장된 암호화되지 않은 기업 장치를 훔쳐 가기 때문에 처음부터 손상되었을 수 있다. LAN 인프라(64)는 로컬 네트워크(유선 및/ 또는 무선)를 통해 연결된 일련의 기업 장치를 나타낸다. 이는 프린터, 서버, 태블릿, 전화기 등이 포함될 수 있다. 전체 LAN 인프라는 MNSP 클라우드(9) 내에서 사실상(가상 라우터 IP 할당, 가상 프린터, 가상 서버 등) 재구성된다. 해커는 시스템이 행동 분석(62)을 수행함에 따라 실제 LAN 인프라와 가상 복제 버전의 요소에 노출된다. 이러한 분석 결과가 위험을 나타내면 실제 데이터 및/ 또는 장치가 손상 될 위험을 줄이기 위해 가짜 인프라(실제 인프라와는 대조인)에 대한 해커의 노출이 증가한다. 해커(65)는 손상된 자격 증명(63)에 의해 사용되는 초기 침입을 통해 중요한 정보를 액세스하고 도용하려는 악의적인 행위자이다. 비밀번호 세트(66)를 사용하면 인증 액세스가 세 가지 비밀번호 세트로 지정된다. 비밀번호는 개별적으로 저장되지 않으며 항상 세트로 제공된다. 직원은 스케줄링된 내부 인증 프로토콜 액세스(SIAPA; Scheduled Internal Authentication Protocol Access)의 임시 할당 프로토콜에 따라 세 가지 비밀번호 조합을 입력해야 한다. SIAPA(예정된 내부 인증 프로토콜 액세스) (67)을 사용하면 개별 직원의 로그인 포털에 대한 인증 프로토콜이 매주/ 매월 변경된다. 이러한 프로토콜은 비밀번호 A, B 및 C (인증을 위해 사전 할당됨) 세트에서 비밀번호 A 및 C를 선택할 수 있다. 일관된 기준(매월 월요일 또는 매월 1 일)으로 인증 변경을 예정함으로써, 직원들은 잘못된 긍정 이벤트들(false positive events) (합법적인 직원이 이전 프로토콜을 사용하고 모의 데이터 환경(394)에서 멈추는 경우)을 최소화하는 인증 프로토콜을 전환하는 것에 익숙해 질 것이다. 해커가 손상시킨 새로운 프로토콜의 위험을 상쇄하기 위해 직원은 파괴되고 검토할 수 없기 전에 한 번만 새로운 프로토콜을 볼 수 있다. 첫 번째 및 유일한 보기에는 휴대 전화 등의 생체 인식/ 망막/ SMS와 같은 특수 다중 요소 인증이 필요하다. 직원은 입력해야 하는 세 가지 비밀번호 중 하나를 나타내는 하나 또는 두 개의 문자를 암기해야 한다. 주 1(68)을 참조하면, 비밀번호 A 및 B만을 제외하고 아무것도 입력하지 않으면 모의 데이터 환경(394)이 트리거될 것이다. 주 2(69)를 참조하면, 암호 A 및 C만을 제외하고 아무것도 입력하지 않으면 모의 데이터 환경이 트리거된다. 주 3(70)을 참조하면 비밀번호 B 만 제외하고 아무 것도 입력하지 않으면 모의 데이터 환경이 트리거된다. 주 4(71)을 참조하면, 모든 암호를 제외한 아무 것도 입력하지 않으면 모의 데이터 환경이 트리거된다. SIAPA(72)에서는 인증 프로토콜이 비밀로 유지되며 임시 공지 사항에 액세스 할 수 있는 사람만 올바른 프로토콜을 알고 있다. LAN 인프라 가상의 복제(73)에서 해커(65)는 올바른 비밀번호를 생략하지 않고 세 개의 비밀번호를 모두 입력했기 때문에 중요한 데이터나 기능이 없는 MNSP 클라우드(9)의 복제 환경으로 자동 전송된다. 해커가 실제 시스템에 성공적으로 침투했다고 믿는 동안 포렌식 증거와 행동 분석을 수집한다. 사례 시나리오 잘못된 프로토콜 사용'(74)을 보면, 해커는 올바른 프로토콜을 사용하지 않았는데 해커가 특정 암호를 생략하기 위한 특별한 프로토콜을 기대하지는 않았을 뿐만 아니라 그를 알 수 있는 방법이 없었기 때문에 올바른 프로토콜을 사용하지 않았다. 도면 부호 75에서 해커는 합법적인 자격 증명을 도용 할 수 있었고 민감한 데이터를 도용하기 위해 회사 시스템에 로그인하는 것을 의도했다. 기업 내부 관리 부서(76)는 행정 위원회와 기술 지휘 본부로 구성된다. 잠재적으로 악의적인 행동을 모니터링하고 승인/ 차단하기 위한 최상위 계층이다. 직원 B와 D(77)은 범죄자가 아니며(그들은 기업의 이익에 독점적으로 충성한다), 루트 레벨 기능(80) 승인의 세가지 협력을 위한 자질 있는 직원으로서 선정되었다. 직원 A(78)는 세가지 협력 프로세스(80)에 선정되지 않았다. 그가 회사, 기술 경험, 범죄 기록과 함께 업무 경험이 충분하지 않거나 회사 등에 대한 음모를 허용했을 수도 있는

다른 직원과 절친한 친구였기 때문일 수 있다. 직원 C(범죄자)(79)는 악의적인 목적으로 수행되는 루트 레벨의 기능/ 작업에 액세스하려고 시도한다. 이러한 루트 레벨 기능(80)은 개별 루트 레벨 액세스 권한을 가진 3 명의 직원의 동의 및 승인 없이는 수행 할 수 없다. 직원 C가 악의적인 의도가 있는 유일한 직원임에도 불구하고 세 명의 직원 모두 이러한 루트 레벨 기능의 결과에 대해 동등하게 책임진다. 이것은 주의와 회의주의의 문화를 유도하고, 절차의 사전 지식으로 인해 직원들이 처음에는 악의적인 행동을 하지 못하게 한다. 직원 E 및 F(81)는 처음에 요청된 루트 레벨 기능를 수행하거나 승인할 루트 레벨 액세스가 없기 때문에 세가지 협력 프로세스(80)에 대해 선택되지 않았다. 관리 검토(82)는 인공 지연에 의해 주어진 시간을 사용하여 요청된 동작을 검토하고 비판한다. 루트 레벨 동작(83)은 관리 부서에 조치를 검토하고 조치를 명시적으로 승인 또는 차단할 기회를 부여하기 위해 1시간 지연된다. 규칙은 관리 부서가 결정을 내릴 수 없거나 취할 수 없는 경우 기본 행동(승인 또는 거절)을 정의 할 수 있다. 관리 검토(84)는 만장일치의 결정이 성취되지 못한 이유에 대한 추론이 무엇인지 결정한다. 수행된 루트 레벨 동작(85)을 참조하면, 협업 및 감독 모니터링 시스템을 통과하면 루트 레벨 동작이 수행되고 누가 승인을 했는지에 대한 기록을 안전하게 유지 관리한다. 이런 식으로 회사의 최대 이익에 반하는 것으로 드러날 경우 상세한 조사가 시작될 수 있다. 도면 부호 86에서는 3가지 협력 실패(만장일치 결정에 도달하지 않음)로 인해 루트 레벨 동작이 취소된다. 도면 부호 87에서, 루트 레벨 액세스 권한을 가진 선택된 직원 세 명 모두 만장일치로 루트 레벨 동작을 승인했다. 루트 레벨 동작이 실제로 악의적인 경우, 3 명의 직원 모두가 회사에 대한 음모의 일부가 되어야 한다. 이와 같은 가능성은 아직 존재하지 않기 때문에, 루트 레벨 동작은 1시간 동안(83) 지연되며, 관리 부서는 이를 검토할 기회가 주어진다(도면 부호 76 및 82 참조). 도면 부호 88에서, 세 가지 협력을 위해 선택된 자격을 갖춘 직원 중 하나 이상이 요청된 루트 레벨 동작을 거절했다. 따라서 루트 레벨 동작 자체는 취소(89)되고 루트 레벨 동작 (89)은 만장일치 결정에 도달하지 않아 취소되었다. 발전 패턴 데이터베이스 (90)는 이전에 발견되어 처리된 보안 위험들의 패턴을 포함한다. 이러한 패턴들은 현재 멀웨어가 진화할 가능성을 열거한다. 멀웨어 루트 서명(91)은 AST (17)에 제공되어 서명(91)의 반복/변형이 형성될 수 있다. 멀웨어의 다형 변이(92)는 I^2GE로부터 출력으로서 제공되고 멀웨어 탐지(95)로 전송된다. 인프라 시스템(93)은 물리적으로 인프라의 구내에 속한다. 이 시스템은 일반적으로 수력 발전소, 전력망 등과 같은 인프라 기능을 관리한다. 인프라 컴퓨터(94)는 시스템(93)의 인프라 기능을 수행 할 수 있는 기능 또는 기능의 일부를 수행하는 특정 컴퓨터이다. 멀웨어 탐지 소프트웨어(95)는 컴퓨터 구성의 세 가지 레벨 모두에 배포된다. 여기에는 사용자 영역(97), 커널 영역(99) 및 펌웨어/ 하드웨어 영역(101)이 포함된다. 이는 3 가지 레벨 각각에 독점적으로 배포되는 LIZARD 라이트 에이전트에서 수행되는 멀웨어 탐지 배포와 일치한다. 발전 경로(34)를 통해 반복된 멀웨어(96) 형태는 드라이버(커널 영역(99) 내에 존재)에서 발견된다. 사용자 영역(97)은 주류 개발자 어플리케이션들을 위한 것이다. 멀웨어에 가장 쉽게 침투 할 수 있는 공간이며 멀웨어를 탐지하고 격리 할 수 있는 가장 용이한 공간이다. LIZARD 라이트는 모든 사용자 공간 활동을 효율적으로 모니터링한다. 사용자 영역 내의 어플리케이션(98)에는 마이크로 소프트 오피스(Microsoft Office), 스카이프(Skype), 퀵켄(Quicken) 등과 같은 프로그램이 포함될 수 있다. 커널 영역(99)은 애플(Apple), 마이크로소프트(Microsoft) 및 리눅스(Linux) 재단과 같은 운영 체제 공급 업체가 대부분 유지 관리한다. 사용자 영역보다 침투하기가 더 어렵지만 각 인프라가 커널 수정을 거치지 않는 한 책임은 주로 공급 업체에 속한다. LIZARD 라이트는 모든 커널 활동(레지스트리 변경을 포함하는 (마이크로소프트 운영체제(Microsoft OS)), 메모리 관리, 네트워크 인터페이스 관리 등)을 효율적으로 모니터링한다. 드라이버(100)는 인프라 컴퓨터(94)가 주변 장치 및 하드웨어(마우스, 키보드, 지문 스캐너 등)와 상호 작용할 수 있게 한다. 펌웨어/ 하드웨어 영역(101)은 펌웨어 / 하드웨어 공급 업체에서 전적으로 유지 관리된다. 멀웨어가 하드웨어에 직접 물리적으로 액세스하지 않고 감염되기는 매우 어렵다(즉, 오래된 BIOS 칩을 마더 보드에서 제거하고 새로운 BIOS 칩을 납땜한다). LIZARD 라이트는 하드웨어 구성에 따라 일부 펌웨어 활동을 모니터링한다. BIOS(102)(일종의 펌웨어)는 동작 시스템이 구축되는 소프트웨어의 제 1 층이다. 공공 인프라(103)는 알 수 없거나 잠재적으로 손상된 디지털 인프라(ISP 라우터, 광 케이블 등)를 말한다. 에이전트(104)는 공공의 인프라에 심어져 있고 신뢰 플랫폼 데이터베이스에 저장된 알려진 설명(포트, 프로토콜 유형 등)을 사용하여 알려진 콜백 채널들을 모니터링한다. 에이전트는 심장 박동 신호들(Heartbeat Signals)을 확인하고 멀웨어 소스에 대한 영향력을 얻기 위해 신뢰 플랫폼에 알린다. 자동 검색 및 라이트 클라이언트 설치(105)를 사용하면 MNSP(9)의 LIZARD 클라우드가 LIZARD에 신호 응답(악수)을 제공하지 않는 엔드 포인트 시스템(즉, 랩탑)을 감지한다. 엔드 포인트는 발견 시 동기화되어 I^2CM(24)를 통해 분류된다. 따라서 LIZARD 클라우드는(SSH 원격 루트 셸을 통해) Lizard 라이트 클라이언트(43)이 설치/ 활성화되지 않았음을 감지하고 루트 셸을 사용하여 클라이언트(43)의 설치를 강제 실행하고 올바르게 활성화되는지 확인한다. 라이트 클라이언트(43)가 입력 디바이스 상에 설치되지 않았기 때문에 멀웨어(106A)는 초기에 들어간다. 라이트 클라이언트(43)는 LIZARD 클라우드가 포함된 MNSP를 통해 모든 들어오고 나가는 트래픽이 라우팅되는 것은 물론, 시스템

에서 가능한 거의 모든 인스턴스에 설치된다. 초기의 익스플로잇(107)을 사용하면 익스플로잇의 초기 실체가 탐지되고 잠재적으로 차단되어 비밀 콜백 채널(106B)를 설정할 수 있다. 채널(106B)은 멀웨어(106B)가 기지와 개별적으로 통신할 수 있는 모호한 통신 경로이다. 이는 합법적 인 http 또는 https 어플리케이션 트래픽처럼 보이도록 신호를 마스킹하는 것을 포함 할 수 있다. 광범위한 공급 업체 (108)는 공공 인프라(103)에서 에이전트(104)를 두기 위해 소프트웨어, 하드웨어, 방화벽, 서비스, 재정 및 중요한 인프라에 대한 비밀 액세스와 같은 중요한 자원을 제공한다. 심박 신호는 콜백 채널(106B) 특정 크기 및 빈도로 멀웨어를 차단하고 숨겨진 콜백 채널을 통해 근원/ 충성도(origin/loyalty) 소스로 연결된다. 신호는 멀웨어 소스(50)가 장래의 익스플로잇 및 조정된 공격을 결정할 수 있도록 하는 상태/ 능력을 나타낸다. 이러한 멀웨어 소스는 악의적 의도가 있는 해킹 기능이 있는 조직을 나타낸다; 블랙 햇 해커 연합체 일 수도 있고 국가 주 정부 일 수도 있다. 멀웨어 (106A) 및 심박 신호(채널 106B 내부)는 MNSP 클라우드(9)에서 실행되는 LIZARD에 의해 탐지되며 모든 수신 및 발신 트래픽은 VPN 터널을 통해 MNSP 클라우드/ Lizard를 통해 라우팅된다.

[0245] 도 22 내지 도 23은 외부 코드 재작성이 잠재적으로 탐지되지 않는 악의적인 공격을 완화하기 위해 외부 코드를 처음부터 구문적으로 재생산하는 방법을 나타내는 도면이다. 조합 방식(113)은 선언된 목적(112A)(이용 가능한 경우, 기업 정책 (147)에 따라 선택 사항 일 수 있음)과 파생된 목적(112B)을 비교 및 매치한다. 목적 모듈(36)을 사용하여 복합적 목적 형식을 조작하고 결과 일치 또는 불일치 사례 시나리오를 달성한다. 파생된 목적(112B)으로 니드 맵 매칭 유지의 계층 구조는 모든 기업의 요구를 관할권으로 유지하기 위한 계층적 구조를 유지한다. 따라서 코드 블록의 목적은 관할 구역 지향적 니드 맵(114)의 공석에 따라 정의되고 정당화 될 수 있다. 입력 목적(115)는 재귀적 디버깅(119) 프로세스(목적 및 구문 모듈을 활용하여)에 대한 입력이다. 여러 입력(즉, 목적)을 병합하지 않으며, 별도의 병렬 인스턴스가 목적 입력마다 초기화된다. 최종 보안 확인 (116)은 구문(35) 및 목적(36) 모듈을 활용하여 프로그래밍에서 악용 지점을 보호하기 위한 다목적 '무결성' 검사('sanity' check)를 수행하고 최종 출력(117)을 VPN/ 엑스트라 넷(12)으로 전송한다.

[0246] 도 24 및 도 25는 재귀적 디버깅(119)이 코드 세그먼트를 통해 버그를 테스트하고 가능한 경우 버그 수정을 적용(솔루션(129))을 반복하는 방법을 보여준다. 버그가 지속되면 전체 코드 세그먼트가 원본 (외부) 코드 세그먼트 (121)로 대체된다(123). 원래 코드 세그먼트에는 가상의 난독화 및 행동 분석과 같은 추가 보안 계층을 용이하게 하기 위해 태그가 추가된다. 외부 코드 120의 경우 코드의 원래 상태는 코드 재 작성을 위해 목적(36) 및 구문(35) 모듈에 의해 해석된다. 외부 코드(120)은 재 작성된 버전에 영구 버그가 있었기 때문에 원본 (외부) 코드 세그먼트를 설치해야 하는 경우 디버거에 의해서 직접 참조된다. 재 작성된 코드(122) 세그먼트(121)는 가상 런타임 환경(131)에 의해 테스트되어 코딩 버그(132)를 검사한다. 이러한 환경(131)은 기능과 클래스와 같은 코드 세그먼트(121)를 실행하고 런타임 오류(구문 오류, 버퍼 오버 플로우, 잘못된 함수 호출 등)를 확인한다. 모든 코딩 오류는 수정을 위해 처리된다. 코딩 버그(132)로, 가상 런타임 환경에서 생성된 에러는 범위 및 유형으로 정의된다. 모든 관련 코딩 세부 정보는 솔루션을 용이하게 하기 위해 제공된다. 목적 얼라인먼트(124)에서, 코딩 버그(132)에 대한 잠재적 솔루션은 그러한 기능들 및 클래스들의 명시된 목적으로부터 코드를 재도입하여 기초한다. 코딩 버그의 범위는 이러한 버그를 피하기 위해 대체 형식으로 다시 작성된다. 잠재적 솔루션이 출력되고 솔루션이 남아 있지 않으면 해당 코드 세그먼트(121)에 대한 코드 재 작성이 취소되고 원래 코드 세그먼트(외부 코드로부터 직접적으로)가 최종 코드 세트에서 사용된다. 전형적으로, 코딩 버그(132)는 코딩 솔루션 (138)을 루프로 여러 번 수신 할 것이다. 모든 코딩 솔루션이 버그(132)를 해결하여 소진된 경우; 솔루션은 몰수되고(137) 원래 외부 코드 세그먼트(133)가 사용된다. 코드 세그먼트(121)는 가상 난독화 및 행동 분석과 같은 추가적인 보안 조치의 결정을 용이하게 하기 위해 외부로 태그(Tagged) 될 수 있다(136). 예를 들어, 재 작성된 코드 블록에 높은 수준의 외부 코드 세그먼트가 포함되어 있으면 모의 데이터 환경(394)에 배치되는 경향이 있다. 코드 세그먼트 캐싱 (130)에서는 개별 코드 세그먼트(기능 / 클래스)가 여러 재 작성 작업에 걸쳐 캐싱되고 재사용되어 LIZARD 클라우드 자원 효율성이 향상된다. 모든 트래픽은 클라우드에서 VPN을 통해 중앙 집중화되므로 이 캐시는 높은 활용도를 유지한다. 재 작성 코드 세그먼트 제공자(128)에 의해, 이전에 재기록된 코드 세그먼트 (121)가 제공되어, 코딩 버그가 각각의 솔루션 적용(129)을 가질 수 있다.

[0247] 도 26은 목적 관할을 검증하는 니드 맵 매칭(114)의 내부 동작을 도시한다. LIZARD 클라우드 및 라이트는 엔터프라이즈 관할 구역의 계층적인 맵 (150)을 참조한다. 이것은 코드/ 기능 목적을 정당화하고 유효한 정당화가 없는 경우 이러한 코드/ 기능을 잠재적으로 차단하기 위해 수행된다. 입력 목적 (139)이 요구되거나(목적 모듈(35)을 통해) 유도되는지 여부에 관계없이, 니드 맵 매칭(114)은 기업 시스템 내에서 수행 할 코드/ 기능에 대한 정당성을 검증한다. 계층적인 맵(150)의 마스터 사본은 MNSP(9)의 LIZARD 클라우드에 각 등록된 기업의 계정에 저장된다. 니드 맵 매칭(114) 내의 필요 인덱스(145)는 마스터 카피를 참조함으로써 계산된다. 그런 다음 사전 최적화된 필요 인덱스 (계층 구조 자체는 아님)는 액세스 가능한 모든 엔드 포인트 클라이언트에 분산된다.

니드 맵 매칭은 시스템의 가장 적합한 필요성에 대한 필요 요청(140)을 받는다. 해당 출력은 적절한 필요를 나타내는 복합적 목적 형식 (325)이다. 필요 기준 + 우선 순위 필터링(143)으로, 기업 정책(147) 내에서 적절한 필요가 검색된다. 이러한 정책(147)은 각 관할권이 가질 수 있는 필요 타입 및 범위를 결정한다. 필요는 이메일 서신, 소프트웨어 설치 요구 등으로부터 다양할 수 있다. 정책(147)은 기업에 따라 필요 우선 순위가 무엇인지를 결정한다. 각 지점과 관련된 정의에 따르면 요구 사항은 해당 부서와 관련된다. 이런 방법으로 권한 체크가 수행 된다. 예: HR이 모든 직원 CV를 다운로드하라는 요청을 승인한 니드 맵 매칭은 직원 성능에 따라 매년 검토해야 한다. 초기 파싱(Parsing)(148)을 통해 각 관할 지점은 필요한 참조를 위해 다운로드 된다. 지점 수요 계산(149)과 함께 필요는 각 부서와 관련된 정의에 따라 해당 부서와 연관 된다. 이런 방법으로 권한 체크가 수행 된다. 예: HR이 모든 직원 CV를 다운로드하라는 요청을 승인한 것은 계층적인 맵(150) 에 정의된 관할 구역에 따라 직원 성과에 대한 매년 검토가 이루어지기 때문이다.

[0248] **사이버 공간에서 비밀 동작을 통해 기계 비밀 정보(MACINT) 및 응징**

[0249] 도 27은 지능화된 정보 관리, 뷰잉 및 컨트롤을 도시한다. 집합(152)은 일반적 레벨 기준을 사용하여 중요하지 않고 중복되는 정보를 필터링하고 여러 플랫폼에서 정보 스트림을 병합하고 태그를 지정한다. 구성 및 배치 서비스(153)는 올바른 보안 구성 및 연결 설정을 갖춘 새로운 기업 자산(컴퓨터, 랩톱, 휴대 전화)을 배치하기 위한 인터페이스이다. 장치가 추가되고 설치된 후 관리 피드백 컨트롤을 중간자로 사용하여 관리 콘솔을 통해 조정할 수 있다. 또한 이 서비스는 새로운 고객/ 클라이언트 사용자 계정의 배치를 관리한다. 이러한 배치에는 하드웨어와 사용자 계정, 인터페이스 사용자 정의, 고객/ 클라이언트 변수 목록(예: 비즈니스 타입, 제품 타입 등)이 포함될 수 있다. 관할 구역에 의한 분리(154)에 따라, 태그가 붙은 정보 풀은 관리 콘솔 사용자의 관할권에 따라 배타적으로 분리된다. 위협으로부터 분리(155)에서 정보는 개별 위협에 따라 구성된다. 모든 타입의 데이터는 위협(자세한 정보가 추가됨)과 상관되거나 제거된다. 지능적 맥락화(Intelligent Contextualization)(156)라고 표시된 프로세스의 단계에서 나머지 데이터는 아일랜드의 클러스터처럼 보이며 각 아일랜드는 사이버 보안 위협이 된다. 플랫폼 간 상호 관계가 보안 분석을 성숙시키기 위해 만들어진다. 위협 데이터 패턴을 이해하기 위해 역사적 데이터에 액세스하고(LIZARD(16)와 반대로 I^2GE(21)에서) CTMP는 비판적 사고 분석에 사용된다. 위협 딜레마 관리(157)에서 사이버 보안 위협은 조감도(큰 그림)에서 감지된다. 이러한 위협을 그래픽으로 표현하기 위해 관리 콘솔로 전달된다. 위협 메커니즘과 관련된 계산된 측정 값이 마침내 여러 플랫폼에서 통합되므로; 보다 정보에 입각한 위협 관리 결정을 자동으로 수행할 수 있다. 자동화된 컨트롤(Automated Controls)(158)은 MNSP(9), TP, 3PS의 관리 관련 컨트롤을 제어하는 알고리즘 액세스를 나타낸다. 관리 피드백 제어(159)는 정책 작성, 법의학, 위협 조사 등을 용이하게 하는데 사용할 수 있는 서비스를 기반으로 하는 모든 MNSP 클라우드, 신뢰 플랫폼(10), 추가적인 제3 서비스 (Third Party Services, 3PS)의 높은 레벨 제어를 제공한다. 이러한 관리 컨트롤은 결국 적절한 사용자 지정 가능한 비주얼 및 프리젠테이션 효율성으로 관리 콘솔 (MC)에 표시된다. 이를 통해 단일 인터페이스에서 직접 전체 시스템(MNSP, TP, 3PI)을 효율적으로 컨트롤하고 조작할 수 있어 필요에 따라 세부 사항을 확대 할 수 있다. 메뉴얼 제어(160)는 MNSP(9), TP, 3PS의 관리 관련 제어 컨트롤에 대한 인간의 접근을 나타낸다. 직접 관리(Direct Management)(161)는 메뉴얼 제어를 활용하여 휴먼 인터페이스를 제공한다. 카테고리 및 관할 구역(162)에서 관리 콘솔의 사용자는 자신의 관할 구역 및 정보 카테고리 액세스 범위를 정의하는 로그인 자격 증명을 사용한다. 모든 잠재적인 데이터 벡터(163)는 동작 중인 데이터, 휴식 중인 데이터 및 사용 중인 데이터이다. 맞춤형 시각화(164)는 회계, 재무, 인사, IT, 법률, 보안/ 감찰관, 개인 정보 보호/ 공개, 노동 조합 등 다양한 기업 부서 및 이해 관계자(각 부서의 임원, 관리자, 임원)를 위한 것이다. 통합된 단일 관점은 모니터링, 로깅, 리포팅, 이벤트 상관, 경고 처리, 정책/ 룰 세트 생성, 시정 조치, 알고리즘 튜닝, 서비스 권한 설정(신규 고객 / 수정), 제3 서비스(타사 서비스 제공 업체 및 공급 업체의 보고서 및 경고/ 로그 수신 등)뿐만 아니라 신뢰 플랫폼 사용과 같은 모든 잠재적 능력의 단일 관점이다. 보안의 모든 측면에 대한 통합적 관점(165)은 경계, 기업, 데이터 센터, 클라우드, 이동식 미디어, 모바일 장치 등을 나타내는 비주얼 모음이다. 사이버 보안 팀(167)은 자격을 갖춘 전문가 팀으로 여러 시스템의 활동과 상태를 모니터한다. 지능형 정보 처리 및 인공 지능(AI) 결정이 이루어지고 있기 때문에 적은 수의 경험으로 적은 인력을 고용함으로써 비용을 절감 할 수 있다. 팀의 주요 목적은 대규모 분석 지점을 수행하는 동안 시스템이 원하는 기준에 따라 성숙하고 진행 중임을 확인하는 대체 단계가 되는 것이다. 행동 분석(168)은 100% 모의 데이터 환경(Mock Data Environment)(394)에 있는 동안 멀웨어의(169) 존재 상태 및 수행된 행동을 관찰한다. 멀웨어가 위조 데이터(170)와 상호 작용하는 동안 행동 분석은 활성화 시간(예: 사무실이 폐쇄된 일요일에만 활성화), 파일 액세스 요청, 요청된 루트 관리 기능 등에서 관찰된 패턴을 기록한다. 멀웨어(169)는 해커(177)에 의해 심어졌다. 해커가 목표 시스템에 멀웨어를 성공적으로 심어 놓은 것으로 믿는 동안, 멀웨어는 조용히 전송되어 100% 모의 데이터 환경(394)에 격리된다. 위조 데이터(170)에서 멀웨어(169)는 가짜

데이터의 사본을 디지털로 소유하고 있다. 이것은 데이터가 실제적이고, 확장되어 해커(177)이 실제 데이터인지 또는 가짜 데이터인지를 알지 못하는 상황에서 이를 수행한다. 멀웨어가 가짜 데이터를 해커에게 보내려고 하면 발신 신호는 실제 해커에 대한 멀웨어의 예상과는 반대로 가짜 해커(174)가 수신하도록 경로가 변경된다. 해커 인터페이스(171)을 사용하면(구문적으로 LIZARD 시스템에 속하는) 구문(35) 및 목적(36) 모듈은 멀웨어(169)의 코드 구조를 수신한다. 이 모듈은 멀웨어의 내부 구조를 역설계하여 해커 인터페이스를 출력한다. 이 인터페이스는 멀웨어와 해커간에 사용되는 통신 방법, 멀웨어가 해커에 대해 갖는 기대치(예: 명령을 수신하는 등) 및 해커가 멀웨어에 대한 기대치(예: 상태 보고서 등)를 자세히 설명한다. 이러한 정보는 가짜 해커(174) 및 위조된 멀웨어(172)가 가상화된 환경(173) 내에서 에뮬레이트 되도록 한다. 행동 분석(168)이 멀웨어(169)의 행동을 적절히 연구하면, MNSP(9)의 신호 모방 기능은 해커(177)처럼 작동하는 프로그램을 에뮬레이트할 수 있다. 이것은 실제 멀웨어(169), 위조 데이터(170) 및 가짜 해커 (174) 사이에 존재하는 통신 프로토콜을 포함한다. 에뮬레이트된 신호 응답(175)를 사용하면 가상화된 가짜 해커(174)가 실제 멀웨어(169)에 응답 신호를 전송하여 실제 작업에 성공했거나 실패했다는 인상을 준다. 그러한 신호에는 멀웨어 행동에 대한 명령 및/ 또는 정보 상태 업데이트에 대한 요청이 포함될 수 있다. 이것은 멀웨어의 다음 행동 패턴을 관찰하기 위해 추가 행동 분석 연구에 사용된다. 조사가 끝나면 멀웨어가 포함된 모의 데이터 환경(394)은 동결하거나 파괴할 수 있다. 에뮬레이션된 응답 코드(176)을 사용하면 해커는 실제 멀웨어의 동작/ 상태와 상관 관계가 없는 가짜 응답 코드를 받는다. 원하는 응징 전술에 따라 가짜 오류 코드 또는 가짜 성공 코드를 보낼 수 있다. 가짜 오류 코드는 해커에게 악성 코드가 작동하지 않는다는 인상을 주며(실제로는 그렇게 할 때) 쓸데없는 디버깅 탄젠트에 해커의 시간을 낭비한다. 성공적인 오류 코드는 해커가 새로운 형태의 멀웨어를 만들려고 하는 경향을 줄이지만, 대신 현재의 오류 코드와 가능한 증분 개선에 집중한다. 이러한 멀웨어는 이미 LIZARD에 의해 손상되고 이해되었으므로 해커는 성공한 것으로 생각되는 손상된 멀웨어에 에너지를 낭비하고 있다. 해커(177)은 그가 심은 멀웨어가 대상 시스템에 성공적으로 침투했다고 믿는다. 실제로 멀웨어는 가상화된 환경에서 격리되었다. 동일한 가상화된 환경은 해커(양방향성 또는 전방향성)와 통신 방법 및 구문을 모방하기 위해 악성 코드에 대해 행동 분석(168)을 규정했다. 범죄 자산(178)은 범죄 시스템(49)의 해킹 및 악의적 인 조작을 용이하게 하기 위해 범죄 재정(184)을 통한 투자를 나타낸다. 이러한 자산(178)은 일반적으로 컴퓨팅 성능과 인터넷 연결성으로 나타나는데, 이는 이 두 자산에 대한 강한 투자가 보다 발전되고 정교한 해킹 성능을 가능하게 하기 때문이다. 범죄 코드(179)를 사용하여 익스플로잇 스캔이 가능한 많은 법의학 증거를 수집하기 위해 신뢰 플랫폼의 에이전트에 의해 수행된다. 범죄 컴퓨터(180)에서는 AVX명령으로 CPU를 오버 플로우시키는 CPU 익스플로잇이 수행된다. 이로 인해 열 발생 증가, 전력 소비 증가, CPU 저하 및 범죄 프로세스에 대한 처리 능력 감소 등의 문제가 발생한다. 범죄 자산(178)의 익스플로잇 스캔(181)은 자신의 능력과 특성을 확인하기 위해 수행된다. 결과 검색 결과는 익스플로잇(185)에서 관리하고 신뢰 플랫폼(10)으로 전달된다. 익스플로잇(185)는 도 27 내지 도 44 MACINT에 열거 된 표적 범죄 시스템(49)에 침투하는 응징 익스플로잇 데이터베이스(187)을 통해 신뢰 플랫폼이 보낸 프로그램이다. 전기 및 냉각 지출이 크게 증가하여 범죄 재정(184)이 고갈된다. 컴퓨터를 종료하면 범죄 행위에 심각한 장애가 될 것이다. 새 컴퓨터를 구입하면 범죄 재정에 더 큰 부담이 되며, 그러한 새 컴퓨터는 오래된 컴퓨터처럼 악용되기 쉽다. 응징 익스플로잇 데이터베이스(187)에는 설치된 백 도어 및 알려진 취약점의 형태로 하드웨어 공급업체(186)가 제공하는 범죄 행위를 악용하는 수단이 포함되어 있다. 통합 포렌식 증거 데이터베이스 (Unified Forensic Evidence Database)(188)는 다수의 기업에 걸쳐있는 다수의 소스로부터 수집된 포렌식 (forensic) 증거를 포함한다. 이 방법은 가능한 가장 강력한 법적 사건이 관련 법원에 제출되도록 형사상에 위임된다. 타겟 선택(189) 대한 타겟은 적절한 포렌식 증거가 확립 된 후에 응징을 위해서만 선택된다. 여기에는 감독에 의한 검토를 위해 포렌식 사건이 보류될 수 있는 최소 시간 요구 사항이 포함될 수 있다(예: 6개월). 증거는 고도로 자기 확신 할 수 있어야 하며, 고립된 사건들은 무고한 목표물을 공격하고 법적 반발을 불러일으킬 수 있다는 두려움 때문에 응징하기 위해 사용될 수 없다. 타겟 검증(190)에서는 다음과 같은 여러가지 방법으로 범죄 시스템을 검증하여 잠재적인 비밀의 방법(공공 카페, TOR 네트워크 등)을 능가한다.

[0250] - 물리적 위치. GPS는 이용할 수 있다. 클라우드 서비스는 검증에 도움이 될 수 있다(즉, Dropbox 로그인 위치에 대한 장기적인 사례).

[0251] - 물리적 장치. MAC 주소, 일련 번호(제조업체/ 공급업체).

[0252] - 인사 검증. 보안 시스템에서 생체 인식 데이터를 사용하고, 전면 카메라에서 사진을 찍어 여러 플랫폼에서 일관된 로그인 자격 증명을 확인할 수 있다.

[0253] 도 33은 범죄자들이 기업 시스템을 어떻게 공격하는지에 대한 MACINT 비밀 운영 개요를 보여준다. 기업 시스템(228)은 기업의 인프라와 자산에 대한 전체 범위와 관할권을 정의한다. 기업 컴퓨터(227)는 민감한 파일(214)을

포함하며, 일반적으로 예정된 작업을 위해 기업 네트워크(219)에 의존하기 때문에 기업 시스템(228)의 중요한 부분이다. 슬리퍼 더블 에이전트(215)는 휴면 상태로 있으며 대상 컴퓨터(227)에서 '잠자기(sleep)'상태 인 악성 소프트웨어이다. 활동이 없기 때문에 프로그래머와 사이버 보안 분석가는 아직 손해가 발생하지 않았으므로 그것을 탐지하기가 매우 어렵다. 범죄 시스템(49)의 해커가 슬리퍼 에이전트(215)를 사용할 수 있는 기회를 찾으면 민감한 파일(214)의 사본이 에이전트(215)에 자동으로 캡처된다. 이 단계에서 해커는 추적 대상에 노출되어 있지만 관리자의 통지없이 에이전트(215)를 설치할 수 있는 기회를 언제 사용할 것인지(즉, 파일(214)이 가치 있는 경우)에 대한 재량에 달려있다. 단계(216)에서 캡처된 파일(214)은 기업 네트워크 외부의 암호화를 통해 가짜 대상 서버로 푸시된다. 이러한 암호화(즉, https)는 정책에 의해 허용되므로, 전송은 즉시 차단되지 않는다. 캡처된 파일(214)은 기업 시스템(228)에서 나가서, 임의의 시스템 (262), 최종적으로 범죄 시스템 (49)으로 들어가기 위한 시도를 위해 기업 네트워크(219)의 네트워크 인프라로 전달된다. 이러한 네트워크 인프라 구조는 LAN 라우터(217) 및 방화벽(218)으로 표현되며, 이는 멀웨어가 캡처된 파일(214)을 기업 시스템 외부로 전송하기 전에 통과하는 마지막 장애물이다. 이 예에서 캡처된 파일(214)의 도용을 방해할 수 없다고 간주되는 산업 표준 방화벽 (218)은 로그 집합(220)으로 전송되는 로그를 생성한다. 그런 다음, 집합는 장기/ 딥 스캔(221) 및 실시간/ 표면 스캔(222)을 포함한다. 빈 결과(223) 시나리오의 경우, 실시간(222)은 악의적인 활동의 거의 즉각적인 인식을 실행하기 전에 부적절하게 수행할 준비가 되어 있지 않다. 멀웨어 연결 발견(224) 시나리오의 경우, 장기 스캔(221)은 분석 시간이 더 많으므로 결국 악의적인 동작을 인식한다. 시간의 여유로 인해 장기간(221)은 더 복잡한 알고리즘과 데이터 포인트로 보다 철저한 검색을 수행할 수 있다. 봇네트 손상된 부분(225)을 사용하면 임의의 제 3자 시스템에 속한 컴퓨터가 민감한 파일(226)을 전송하여 조사를 중단하고 임의의 제 3자를 구성한다. 도둑들은 봇네트를 통해 은닉된 존재를 유지하면서 범죄 컴퓨터(229)에서 민감한 파일(226)을 받고, 불법 강탈과 이익을 위해 파일을 사용하게 된다. 범죄 컴퓨터의 신원(즉, IP 주소)의 잠재적 흔적은 기업 시스템(228)의 관리자와 조사관이 액세스할 수 없는 임의 컴퓨터(238)에만 남겨질 수 있다.

[0254] 도 34는 빅데이터(231)를 사용하는 장기/ 딥 스캔(230)을 더 상세하게 도시한다. 딥 스캔(230)은 '음모 탐지(Conspiracy Detection)'및 '외부 엔티티 관리(Foreign Entities Management)의 두 가지 하위 알고리즘을 활용하는 동안 빅 데이터(231)에 기여하고, 빅 데이터(231)를 처리한다. 중간 결과는 최종 결과를 담당하는 이상 탐지로 푸시된다. 방화벽 및 중앙 서버와 같은 보안 검사 포인트의 기준 로그는 로그 집합(220)에서 낮은 제한 필터를 사용하여 집합되고 선택된다. 이벤트 인덱스 + 추적(235)에서 이벤트 세부 정보들은 IP 주소, MAC 주소, 공급 업체 ID, 일련 번호, 시간, 날짜, DNS 등이 있다. 이러한 세부 정보는 로컬 데이터베이스와 공유 클라우드 데이터베이스(데이터베이스들은 데이터가 동일하지 않음)로 존재한다. 이러한 엔트리의 로컬 저장은 다른 기업을 위해 클라우드 데이터베이스에 푸시된다(기업에 따라 정책 제한이 있음). 대신 로컬 분석을 위해 유용한 이벤트 정보가 수신된다. 신뢰되는 제 3 자 (Trusted Third Party)(235)에 등록된 기업은 이미 봇네트의 범죄를 경험했을 수 있으며 이러한 위험을 줄이기 위한 예방적인 세부 정보를 제공할 수 있다. 보안 행동(236)에서 보안 반동 지침은 로컬 데이터베이스와 공유 클라우드 데이터베이스에 저장된다(이러한 데이터베이스들은 데이터가 동일하지 않다). 이러한 반동 지침은 보안 시스템을 보장하기 위한 행동 포인트를 정의한다. 예를 들어, 이벤트 인덱스가 말하는 봇네트에 10번 중 6번이 연결된 시스템에 액세스 한 IP 주소가 있으면 30 일 동안 IP 주소를 금지하고 로그 시스템에 우선 순위 플래그를 지정하여 IP주소를 표시한다. 이러한 지침의 로컬 저장은 다른 기업의 이익을 위해 클라우드 데이터베이스로 푸시된다(기업에 따라 정책 제한이 있음). 대신 로컬 분석을 위해 유용한 이벤트 정보가 수신된다. 이상 탐지(237)를 사용하면 권한이 없는 에이전트가 기업 네트워크 외부의 임의 시스템으로 전송하는 민감한 파일과 같은 잠재적 위험 이벤트를 확인하기 위해 딥 스캔 모듈에서 제공하는 중간 데이터에 따라 이벤트 인덱스 및 보안 행동이 사용된다. 임의의 컴퓨터(238)는 위반에 관련된 결과 대상 서버가 강조 표시되며 MAC 주소/ 최후에 알려진 IP 주소(239), 국가 및 가동 시간 패턴 등과 같은 알려진 특성으로 정의된다. 이러한 분석은 주로 외부 엔티티 관리(232) 모듈을 포함한다. 그런 다음 시스템은 봇네트에 관련된 컴퓨터의 가능성(240)을 결정할 수 있다. 이러한 분석은 주로 음모 탐지(19)를 포함한다.

[0255] 도 35는 임의의 컴퓨터 또는 임의의 컴퓨터 관련/인접 서버(다른 서버와 연결하는 서버)가 신뢰할 수 있는 플랫폼(10)에 대해 이중 또는 삼중 에이전트인지를 확인하기 위해 임의의 컴퓨터가 신뢰 플랫폼(10)에서 어떻게 검색되는지를 보여준다. 단계(242)는 MAC 주소/ IP 주소(239)와 같은 임의의 컴퓨터(238)의 알려진 정보가 이벤트 인덱스 + 추적(235) 및 클라우드 버전 (232)에서 쿼리를 위해 전송되는 방법을 나타낸다. 신뢰 플랫폼(10)에서 작동하는 이러한 클라우드 버전은 미래의 위협 및 위협 패턴을 식별하기 위한 이벤트 세부정보를 추적한다. 즉 MAC 주소, IP 주소, 액세스 타임 스탬프 등. 그러한 쿼리(querying)(242)로부터의 결과는 시스템 수집 세부 사항(243)으로 전송된다. 그러한 세부 사항은 원래 임의의 컴퓨터(238) 세부 사항, 패킷을 컴퓨터(238)에 정기적으로 수신 및/또는 송신하는 컴퓨터들/시스템들, 및 컴퓨터(238)에 물리적으로 근접한 시스템을 포함한다. 그러

한 세부 사항은 단계(246 및 247)로 전송되며 언급된 컴퓨터들/시스템들 중 임의의 것이 이중 에이전트(247) 또는 삼중 에이전트(246)에 발생 하는지를 확인한다. 이러한 에이전트 룩업 검사는 신뢰되는 이중 에이전트 인덱스 + 추정 클라우드 (244) 및 신뢰되는 삼중 에이전트 인덱스 + 추정 클라우드 (245)에서 수행된다. 이중 에이전트 인덱스(244)에는 신뢰할 수 있는 플랫폼과 그 계열사가 제어하는 슬리퍼 에이전트가 설치된 시스템 목록이 포함되어 있다. 삼중 에이전트 인덱스(245)에는 범죄 조직(예: 봇네트)에 의해 침해 받았지만 악의적인 활동 및 개발을 모니터하기 위해 신뢰 플랫폼(10)에 의해 개별적인 방식으로 손상된 시스템 목록이 포함되어 있다. 이 2개의 클라우드들은 그 결과를 활성화 및 관련된 에이전트 목록(248)에서 수집한 결과를 출력한다.

[0256] 도 36은 신뢰 플랫폼(10)으로부터 알려진 이중 또는 삼중의 에이전트가 포렌식 조사를 더 진행하기 위해 어떻게 참여해야 하는지를 도시한다. 에이전트 목록(248)에서 전송되고 적절한 슬리퍼 에이전트(252)가 활성화(249)된다. 임의 시스템(238)이 신뢰하는 이중 에이전트 시스템(251)은 신뢰되는 채널(254)을 통해 익스플로잇(253)을 푸시한다. 임의의 컴퓨터 (238)에 성공적으로 배치되면 익스플로잇(253)은 민감한 파일(241)의 활동을 추적하고 그것이 범죄 컴퓨터(229)로 알려져 있음을 알게 된다. 이는 채널(255)에서 처음으로(216) 파일 (241)을 전송하는데 사용된 것과 동일한 경로를 따르고 범죄 컴퓨터(229)에서 자체 설정을 시도한다. 그러면 익스플로잇(253)은 민감한 파일(241)을 찾고, 그것을 격리하며, 정확한 상태를 신뢰 플랫폼(10)으로 되돌려 보내고, 범죄 컴퓨터(229)로부터 이를 지우려고 시도한다. 신뢰 플랫폼(10)은 검역된 파일을 포렌식 목적으로 원래 기업 시스템(228)(원래 파일을 소유한 사용자)에 다시 전달한다. 익스플로잇(253)이 민감한 파일(241)을 검색할 수 있다고 항상 보장되는 것은 아니지만 적어도 범죄 컴퓨터(229) 및 시스템(49)에 대한 식별 가능한 정보(239)를 전달할 수 있다.

[0257] 도 37은 신뢰 플랫폼(10)이 임의의 컴퓨터(238)에 관한 ISP(Internet Service Provider)(257) API를 사용하는 방법을 보여준다. 네트워크 관리(Network Oversight)(261)는 사법 조사를 증진하기 위해 임의 시스템(Arbitrary System)(262)를 시도하고 절충하기 위해 사용된다. 기업 시스템(228)은 임의의 컴퓨터(238)에 관한 제한된 정보(259)만을 알고 범죄 컴퓨터(229) 및 시스템(49)에 관한 정보를 찾고 있다. ISP(257) API 요청은 신뢰 플랫폼(10)을 통해 이루어진다. 네트워크 감독(261) 시스템에서 임의 시스템 (262)에 대한 시스템 네트워크 로그가 발견되고 범죄 컴퓨터(229)에 대한 잠재적 인 파일 전송(나중에 인식되는 것)이 발견된다. 로그 이력은 민감한 파일(241)의 정확한 및 전체 구성을 기록하기에 충분하지는 않지만, 메타 데이터(260)를 사용하여 파일이 전송된 컴퓨터를 결정할 수 있다. 네트워크 관리(261)는 범죄 컴퓨터(229)의 네트워크 세부 사항(258)을 발견하고 따라서 그러한 정보를 신뢰 플랫폼(10)으로 재 라우팅하고, 신뢰 시스템은 기업 시스템(228)에 통지한다.

[0258] 도 38은 신뢰 플랫폼(10)이 소프트웨어(268) 및 하드웨어(272) 공급업체가 제공하는 보안 API를 활용하여 사법 조사를 보좌할 수 있는 수립된 백도어를 개발하는데 사용하는 방법을 보여 준다. 단계(263)에서, 범죄 컴퓨터(229)의 공지된 신원 상세 정보가 신뢰 플랫폼(10)으로 전달되어 백도어 API에 관여한다. 그러한 세부 정보에는 MAC 주소/IP 주소(239) 및 범죄 컴퓨터의 의심 소프트웨어 + 하드웨어가 포함될 수 있다. 그런 다음 신뢰 플랫폼(10)은 휴면 상태(착취 코드는 아직 실행되지 않음)에서 제휴 소프트웨어(268) 및 하드웨어(272) 공급 업체에 익스플로잇(253)을 제공한다. 또한, 단계(263)에서 기업 시스템(228)에 의해 의심되는 범죄 컴퓨터(229)의 의심되는 소프트웨어(269) 및 하드웨어(273)가 공급업체에게 전달된다. 공급 업체는 자신을 호출하는 방법, 취해야 할 조치의 기준, 기능 및 제한 사항과 같은 정보를 포함하여 설치된 소프트웨어(270) 및 하드웨어(274) 백도어 목록을 유지 관리한다. 이러한 모든 백도어는 내부적으로 격리되어 있으며 공급 업체 내에서 기밀로 유지되므로 신뢰 플랫폼은 그러한 백도어를 다루는 민감한 정보를 받지 못하고 혜택을 받을 수 있는 익스플로잇(253)을 제공한다. 소프트웨어(267) 또는 하드웨어(271) 백도어의 성공적인 구현 시에, 익스플로잇(253)은 범죄 컴퓨터(229)에 이산적으로 설치된다. 민감한 파일(241)은 나중에 메타 데이터 사용 내역이 분석 될 수 있도록 격리되고 복사된다. 범죄 컴퓨터(229)상의 나머지 사본들은 안전하게 지워진다. 가능한 다른 보완적인 포렌식 증거가 수집된다. 모든 포렌식 데이터는 신뢰 플랫폼(10)의 익스플로잇(253)의 포인트로 반환된다. 그 다음에 포렌식 증거(265)는 범죄 시스템(229)에서 발견된 민감한 파일(241)과 파일(241)의 초기 도난에 관한 증거가 있는 범죄 시스템과 관련된 사람들의 신원 정보를 포함하는 기업 시스템(228)로 전달된다. 이러한 방법으로 기업 시스템(228)은 초기 도난시 시스템에서 삭제된 경우 파일(241)을 복원할 수 있으며 식별 세부정보(264)를 사용하면 법적 손해 배상 및 범죄 시스템(49) 봇네트을 비활성화하여 향후 공격의 위험을 완화할 수 있다.

[0259] 도 39 내지 도 41은 신뢰 플랫폼(10)의 직접 지원 없이 직접 타협을 수행하기 위해 일반적인(282) 및 맞춤형(283) 익스플로잇이 임의(238) 및 범죄(229) 컴퓨터들에 적용되는 방법을 도시한다. 일반적 익스플로잇(282)는 독립적인 사이버 보안 연구를 통해 기업 시스템(280)에 의해 조직되고 조립된 소프트웨어, 펌웨어 및 하드웨어

익스플로잇 모음이다. 익스플로잇을 사용하여 맞춤형(283) 익스플로잇은 대상에 대한 알려진 정보에 따라 사용자 지정된다. 익스플로잇(253)은 가장 먼저 성공할 확률이 가장 높고 가장 늦게 성공할 확률이 가장 낮은 상태로 제공된다. 범죄 컴퓨터(229)에 관한 이용 가능한 정보(284)의 수집은 맞춤형(283)으로 전달된다. 그러한 정보는 범죄 컴퓨터(229)에 의해 사용되는 MAC 주소/IP 주소 (239) 및 의심되는 소프트웨어 + 하드웨어 (285)와 같은 임의의 알려진 컴퓨터 정보를 포함한다. 프록시 관리 (286)는 공격 시도에 사용될 프록시를 지능적으로 선택하는 알고리즘 및 데이터베이스의 조합이다. 프록시 네트워크(279)는 일련의 프록시 노드(278)로서, 임의의 개별 시스템이 자신의 발신 신원을 마스킹할 수 있도록 한다. 노드는 이러한 디지털 통신을 통과하여 명백한 발신자가 된다. 노드는 노드의 전반적인 성능, 노드의 가용성 및 노드의 현재 작업 부하에 따라 프록시 관리(286)에 의해 지능적으로 선택된다. 범죄 컴퓨터(229) 및/또는 임의 컴퓨터(238)의 세 가지 잠재적 착취 포인트가 시도된다. 범죄 컴퓨터(229)를 악용하는 것이 실패하면 임의의 컴퓨터(238)를 악용하려는 시도가 전반적인 포렌식 조사를 용이하게 할 수 있으므로 관계없이 이루어진다. 첫번째 방법은 직접 착취이고, 두 번째는 임의 컴퓨터의 봇네트 터널(276)을 통해 이루어지며, 세 번째 방법은 범죄 시스템이 봇네트(277)를 설치하는데 사용된 원래의 착취 방법이다(사용되지 않은 다른 사용 포인트도 포함). 봇네트 터널(276)은 범죄 컴퓨터(229)와 봇네트(240)의 활성 부분 사이에 사용되는 확립된 통신 수단이다. 익스플로잇(253)에 의해 생성된 포렌식 데이터는 단계(275)에서 기업 시스템(228)으로 전송된다.

[0260] 도 41은 신뢰 플랫폼(10)을 갖는 특별한 API가 소프트웨어 또는 펌웨어 업데이트(289)를 범죄 컴퓨터(229)에 푸시하여 새로운 백도어를 구축하는 방법을 도시한다. 플라시보 업데이트(288)는 스텔스를 유지하기 위해 가까운 유사한 기계에 푸시된다. 기업 시스템(228)은 신뢰 플랫폼(10)에 대상 신원 상세 정보 (297)를 전송한다. 이러한 세부 사항은 MAC 주소 / IP 주소 (239)를 포함한다. 신뢰 플랫폼(10)은 소프트웨어/펌웨어 유지 장치(287)와 통신하여 관련 컴퓨터에 플라시보 업데이트(288) 및 백도어 업데이트(289)를 푸시한다. 백도어 업데이트는 컴퓨터에 설치된 사전 설치된 소프트웨어 업데이트 시스템을 사용하여 범죄 컴퓨터의 (229) 시스템에 새로운 백도어를 도입한다. 이러한 업데이트는 운영 체제, BIOS (펌웨어), 워드 프로세서와 같은 특정 소프트웨어에 대한 업데이트 일 수 있다. 플라시보 업데이트(288)는 백도어가 누락되어 보안 손상이 없도록 하고 백도어 업데이트(289)와 동일한 세부 사항 및 식별(즉, 업데이트 번호 / 코드)을 표시하여 백도어를 은폐하는 환경을 유발한다. 유지 장치(287)는 백도어(295)를 타겟 뿐만 아니라 타겟에 평균 이상의 노출량을 갖는 컴퓨터로 전송한다. 그러한 추가 컴퓨터(296)는 범죄 시스템(49) 인프라에 속하는 컴퓨터 또는 범죄 컴퓨터(229)와 동일한 로컬 네트워크에 있는 컴퓨터 일 수 있다. 이러한 추가적인 컴퓨터(296)를 이용하면 직접 공격이 불가능한 경우 (즉, 운영 체제에 대한 업데이트를 끄는 등) 범죄 컴퓨터(229)로의 진입 경로를 확보할 기회가 증가한다. 그러면 익스플로잇(253)은 가까운 컴퓨터(296)에서 자신을 설정할 수 있는 경우 대상에 대한 다른 진입 포인트를 고려할 수 있다. 타겟에 대한 평균 노출량을 가진 관련 컴퓨터(291)에 대해 플라시보 업데이트 (228)가 제출된다. 노출은 공통 네트워크(즉, 가상 사설망 등) 또는 공통 서비스 플랫폼(즉, 파일 공유 등)을 공유하는 것으로 이해 될 수 있다. 관련 시스템 (290)은 전략적으로 동일한 회사의 법적 구조 등이 소유하는 것처럼 범죄 시스템 (49)에 묶일 수도 있다. 이웃한 시스템(292)에 속하는 이웃한 컴퓨터(293)는 타겟 범죄 컴퓨터(229)와의 그들의 물리적인 위치(동일한 지구 등) 때문에 플라시보 업데이트를 받는다. 가까운 유지장치(287)가 제공할 예정인 정기 업데이트(또는 조사에 적합하고 실행 가능한 것이 무엇이든지)가 없는 동안에 민감한 포렌식 조사를 용이하게 하기 위해 관련된(290) 및 이웃한(292)에 시스템 모두에 플라시보 업데이트(288)가 제공된다. 소프트웨어/펌웨어 개선을 위한 정기적인 업데이트가있는 시나리오의 경우, 관련된(290) 및 이웃한(292) 시스템에는 백 도어(289) 업데이트의 인식된 정당성을 확인하기 위한 플라시보 업데이트가 필요하지 않다. 대신에 백도어(289)는 범죄 컴퓨터(229) 및 기타 컴퓨터(296)를 대상으로 하는 정당한 업데이트 중 일부에 설치 될 수 있다. 백도어 업데이트(295)를 통해 익스플로잇(253)을 성공적으로 구현하면 민감한 파일(241)은 나중에 메타 데이터 사용 내역을 분석 할 수 있도록 격리되고 복사된다. 범죄 컴퓨터(229)의 나머지 사본은 안전하게 지워진다. 보완적인 포렌식 증거가 수집된다. 그 후, 포렌식 데이터는 신뢰 플랫폼(10)에서 익스플로잇의 접촉 포인트로 보내진다. 플랫폼(10)에서 데이터가 검증되면 결과 (281)에서 엔터프라이즈 시스템 (228)으로 전달된다.

[0261] 도 42는 임의의 및 모든 변경 / 업데이트에 대해 범죄 시스템(229)을 모니터링하기 위해 장기 우선 순위 플래그가 신뢰 플랫폼(10)에 푸시되는 방법을 도시한다. 신규 개발은 조사를 용이하게 하기 위해 장기간 우선적으로 모니터링된다. 먼저, 기업 시스템(228)은 신뢰 플랫폼(10)의 서브 세트 인 보증 모듈(300)에(식별 가능한 세부 사항(239)을 포함하는) 목표(297)를 제출한다. 보증 모듈은 정의된 목표(297)의 모든 연관에 대해 모든 제휴 시스템 (303) 입력(299)을 스캔한다. 일치하는 항목이 있으면 정보는 근거를 정의하고 목표(297)에 침투하려고 하는 기업 시스템(228)으로 전달된다. 정보 입력(299)은 신뢰 플랫폼(10)의 제휴 시스템이 보고하는 정보로서, 일반적으로 원하는 분석을 수신하기 위한 것이다. 또한 신뢰 플랫폼(10)에 대한 인증 및 명성을 얻으려는 목적으

로 입력을 제출할 수도 있다. 제휴사 시스템(303)은 그들의 입력을 신뢰 플랫폼(10)에 제출한다; 이것은 목표(297)을 모니터링하려고 하는 기업 시스템(228)의 장점이다. 이것은 긍정적인, 중립적인 또는 부정적인 상호 작용이든 관계없이 이러한 제휴사 시스템(303) 중 하나가 목표나 목표와 연관이 있는 기회를 증가시킨다. 이러한 입력(299)은 상호 이익이 되는 보안 정보를 동기화하기 위한 신뢰 플랫폼(10)의 다수 기능을 나타내는 원하는 분석 모듈(301)로 전송된다. 제휴사 시스템(303)은 보안 요청을 게시하고 보안 정보를 교환한다. 목표(297) 또는 목표와 연관에 관한 정보가 발견되면, 정보는 또한 보증 모듈(300)에 병렬로 전달된다. 모듈(301)의 정보 출력(302)은 그들의 요청 된 업무 또는 기능을 완료하기 위해 제휴사 시스템(303)으로 전송된다. 목표(297)에 관한 보증 모듈(300)에서 얻은 유용한 정보는 기업 시스템의 (228) 포렌식 조사의 일부로 결과(298)로 전달된다.

[0262] **논리적 추론된 제로-데이터 베이스 선험적 실시간 방어(LIZARD)**

[0263] 도 43 및 도 44는 LIZARD (논리적 추론된 제로-데이터 베이스 선험적 실시간 방어)의 의존 관계 구조를 도시한다. 정적 코어(193)는 주로 프로그래머가 고정 된 프로그램 모듈을 하드 코딩한 곳이다. 반복 모듈(194)는 동적 쉘(198)에서 모듈을 지능적으로 수정, 생성 및 파괴한다. 보안 성능에 대한 참조를 위해 인공 보안 위협(AST)을 사용하고 자동 코드 작성 방법을 처리하기 위해 반복 코어를 사용한다. 반복 코어(195)는 도 51에 도시 된 바와 같이 보안 개선을 위해 동적 쉘(198)을 반복하는 주된 논리이다. 차등 수정 알고리즘(196)은 AST가 발견한 결함에 따라 기본 반복을 수정한다. 차동 논리가 적용된 후, 새로운 반복이 제안되며, 이 반복 코어는 재귀적으로 호출되며 AST에서 테스트하는 동일한 프로세스를 거친다. 논리적 추론 알고리즘(LDA)(197)은 현재의 상태인 인공 보안 위협(AST)에서 동적 쉘 반복의 알려진 보안 응답을 수신한다. 또한 LDA는 코드 세트 메이크업이 보안 시나리오(AST에서 제공)에 대해 알려진 올바른 응답을 얻을 수 있는지 추론한다. 동적 쉘(198)은 주로 반복 모듈에 의해 자동으로 프로그램된 동적 프로그램 모듈을 포함한다. 코드 격리(199)는 외부 코드를 제한된 가상 환경(즉, 페트리 접시)으로 분리한다. 비밀 코드 탐지(200)는 데이터 및 송신 패킷에 은밀하게 삽입된 코드를 검출한다. AST 오버플로 릴레이 (Overflow Relay)(201) 시스템은 신뢰도가 낮은 결정만 수행할 수 있는 경우 향후 반복 개선을 위해 AST(17)로 릴레이된다. 내부 일관성 검사(Internal Consistency Check)(202)는 외부 코드 블록의 모든 내부 기능이 의미 있는지를 검사한다. 전체적으로 외부 코드의 목적과 내부적으로 일치하지 않는 코드 조각이 없는지 확인한다. 외부 코드 재 작성(Foreign Code Rewrite)(203)은 외부 코드 목적을 유도 한 후, 부분 또는 전체 코드 자체를 재 작성하고 재 작성 만이 실행되도록 허용한다. 미러 테스트는 다시 쓰기의 입력 / 출력 동적이 원본과 동일한 지 확인한다. 이러한 방법으로 원래 코드에서 숨겨진 모든 익스플로잇 사례가 중복되어 실행되지 않는다. 니드 맵 매칭(204)은 외부 코드가 시스템의 전체 목적(즉, 퍼즐)에 적합한지를 결정하기 위해 참조된 목적 및 목적에 매핑 된 계층이다. 실제 데이터 동기화 장치(Real Data Synchronizer)(205)는 혼합된 환경에 어떤 우선 순위로 데이터를 지능적으로 선택하는 2개의 계층(다른 하나는 데이터 관리자임) 중 하나이다. 이러한 방식으로 매우 민감한 정보는 의심스러운 멀웨어에 액세스할 수 없으며 잘 알려진 신뢰할 수 있는 것으로 입증된 코드에서만 사용할 수 있다. 데이터 매니저(206)는 가상 환경 외부로부터 오는 엔티티 및 데이터 간의 중개자 인터페이스이다. 프레임 워크 코디네이터 (Framework Coordinator)(207)는 반 인공적 또는 인공적 알고리즘의 모든 입력, 출력, 스레드 스폰 및 진단을 관리한다. 가상 난독화(Virtual Obfuscation)(208)는 가상의 가짜 환경으로 점차적으로 그리고 부분적으로 침범함으로써 코드(따라서 잠재적 멀웨어)를 혼동하고 제한한다. 비밀 전송 모듈(209)은 모의 데이터 환경(394)에 멀웨어를 자동 및 개별 전송한다. 목적 비교 모듈(210)을 사용하여, 4 가지 유형의 목적을 비교하여, LIZARD가 시스템의 전체 목적에 대해 생산적으로 기업의 존재와 행동을 인정하고 이해하도록 한다. 잠재적으로 폭이 다른 목적이 악의적인 행동을 나타낸다. 모의 데이터 생성기(211)는 실제 데이터와 구별 할 수 없도록 설계된 가짜 데이터를 생성한다. 즉 SSN의 배치. 가상 환경 관리자 (212)는 모의 데이터의 비율, 이용 가능한 시스템 기능, 네트워크 통신 옵션, 저장 옵션 등과 같은 변수를 포함하는 가상 환경의 구축을 관리한다. 데이터 회수 추적(Data Recall Tracking)(213)은 의심스러운 엔티티(415)로부터 업로드되고 다운로드된 모든 정보를 추적한다. 이는 멀웨어로 잠재적으로 전송되는 중요한 정보의 보안 위험을 완화하기 위해 수행된다. 이 보안 검사는 모의 데이터를 받는 합법적인 기업 프로세스의 병목 문제를 완화한다. 모의 데이터가 합법적인 기업 엔티티로 전송된 경우 모든 모의 데이터를 호출하는 "콜백"이 수행되고 실제 데이터(원래 요청된)가 전송된다.

[0264] 도 45는 동적 성장 데이터베이스의 직접적인 도움 없이 실시간으로 잠재적인 모든 사이버 보안 위협을 차단할 수 있는 중앙 감시 알고리즘인 LIZARD (논리적 추론된 제로-데이터 베이스 선험적 실시간 방어)의 개요를 보여준다. 시스템에 대한 데이터/액세스가 허용되는지 여부를 결정하는 것은 알아야 할 필요성, 기능상의 필요성, 목적 중심의 근거에 기반한다. 코드 블록이나 데이터 블록이 시스템의 하드 코딩된 목표를 달성하는 기능/목적을 제공 할 수 없는 경우 가상 격리 및 난독화를 포함하는 은밀한 방식으로 거부된다. LIZARD에는 컴퓨터 코드

를 읽고 쓸 수 있는 구문 해석기가 있다. 목표 파생 기능과 결합하여 겉보기에 양성 데이터에 은밀하게 포함된 코드 블록에서도 목표 지향적 동작을 유도할 수 있다. 공공 커피 숍에서 회사 전화와 같은 기업 부지 밖의 모든 기업 장치는 LIZARD를 통해 라우팅된다. 기업 자산을 실행하는 모든 소프트웨어 및 펌웨어는 영구적인 프록시처럼 LIZARD를 통해 모든 종류의 다운로드/업로드를 수행하도록 하드 코드된다. 충성도가 높은 자산에 대한 정책에 따라 영구 프록시 정책을 준수하지 않을 수 있다. 기업 시스템 내에서 발생하는 디지털 전송은 LIZARD를 통해 릴레이되도록 하드 코드된 하드웨어를 통과해야 하므로 악의적인 코드는 안전하지 못하거나 영구 프록시 정책을 무시한 협업된 컴퓨터를 찾을 수 없다. LIZARD는 반복 모듈(IM)과 공생 관계가 있다. IM은 LIZARD의 하드 코딩된 목표 지향적 작업과 구문 이해 능력을 복제한다. 그런 다음 구문론적 기능을 사용하여 하드 코드된 목표에 맞게 LIZARD를 수정한다. 인공 보안 위협(AST) 모듈은 다양한 가상 현실 환경에 종사하여 LIZARD의 다양한 변형을 테스트한다. 최고 점수를 얻은 유사 콘텐츠가 다음 공식 반복으로 선택된다. LIZARD는 사이버 보안 솔루션의 현상 유지에서 벗어난 혁신적인 모델을 제공한다. 고급 논리 추론 능력을 사용하여 동시대의 사이버 보안 방어에 대한 "너무 늦은" 패러다임 없이 즉각적이고 정확한 보안 결정을 수행 할 수 있다. LIZARD는 세 가지 유형의 데이터(동작중인 데이터, 사용중인 데이터 및 휴지중인 데이터)와 상호 작용한다. LIZARD는 파일, 이메일, 웹, 모바일, 클라우드 및 이동식 미디어 (USB)와 같은 6 가지 유형의 데이터 매체(벡터라고 함)와 상호 작용한다. 기업 시스템(228)은 HTTP 및 DNS 등과 같은 인프라 내에서 실행중인 서버의 유형을 도시한다. 모바일 장치(305)는 LIZARD 라이트 클라이언트(43)를 통해 기업 시스템(228) 디지털 인프라에 접속되는 동안 공공 커피 숍(306) 내에서 작동하는 것으로 도시된다. 그러한 클라이언트(43)는 이후에 암호화된 LIZARD 클라우드(308)에 접속하는 인터넷(304)에 대한 게이트웨이로서 동작한다.

[0265] 도 46은 LIZARD에 관한 주요 알고리즘 기능의 개요를 도시한다. LIZARD의 외부 동적 셸(DS)(313)은 반복을 통해 변경되기 쉬운 기능 섹션이다. 목적을 달성하기 위해 고도의 복잡성이 필요한 모듈은 일반적으로 이 쉘(313)에 속한다. 프로그래머 팀이 직접 처리할 수 있는 복잡성 수준을 능가 할 것이기 때문이다. 반복 모듈(314)은 정적 코어(SC)(315)를 사용하여 데이터 반환 릴레이(DRR)(317)의 '고정 목표'및 데이터에서 정의 된 목적에 따라 DS(313)의 코드베이스를 구문론적으로 수정한다. 이 수정된 버전의 LIZARD는 인공 보안 위협(AST)(17)에 의해 다수의 다양한 보안 시나리오와 스트레스 테스트 (병렬)된다. 가장 성공적인 반복이 라이브 기능 버전으로 채택된다. LIZARD의 SC (315)는 자동화된 반복을 통해 변경하기가 쉽지 않으며 대신 인간 프로그래머가 직접 변경한다. 특히 내부 코어(334)로 알려진 가장 안쪽의 사각형은 자동화된 반복의 영향을 전혀 받지 않는다. 이 가장 안쪽 층(334)은 LIZARD의 방향 및 전체 용량을 안내하는 나무의 뿌리와 같다. 일반적인 동적 모듈(General Dynamic Modules)(316)은 자동 셀프 프로그래밍에 가장 많이 적용될 수 있는 모듈 영역이며, 따라서 동적 쉘(313)의 관할권에 속한다. GDM(316)에서 실행되는 이러한 프로그램은 일정한 '베타 '상태(반드시 안정적 일뿐만 아니라 진행중인 작업)에 있다. LIZARD가 낮은 신뢰도 결정을 수행하면 LIZARD의 향후 반복을 개선하기 위해 데이터 반환 릴레이(DRR)(317)를 통해 관련 데이터를 AST(17)에 중계한다. LIZARD 자체는 의사 결정을 수행하는데 데이터에 직접 의존하지 않지만 진화하는 위협에 대한 데이터는 LIZARD의 향후 반복이 수행할 수 있는 선험적인 의사 결정에 간접적으로 도움이 될 수 있다. 레이블(342)은 코드 디자인에 인력이 많이 포함되어 있을수록 코드가 더 정적(매우 점진적으로 변경됨)이라는 것을 보여준다. 반복 모듈(IM)(314)이 코드를 프로그램 할수록 코드는 보다 동적이며 유동적이다. 구문(35) 및 목적(36) 모듈은 SC(315) 내에서 기능하는 것으로 표시된다.

[0266] 도 47은 정적 코어(SC)(315)의 내부 동작을 보여준다. 논리 유도(Logic Derivation)(320)는 초기에 보다 단순한 함수로부터 논리적으로 필요한 함수를 유도한다. 결과적으로 함수 종속성의 전체 트리는 명시된 복잡한 목적에서 생성된다. 코드 번역(321)은 구문 모듈 기능에 의해 직접 이해되는 임의의 (일반적인) 코드를 선택한 알려진 컴퓨터 언어로 변환한다. 알려진 컴퓨터 언어를 임의 코드로 변환하는 역관계도 수행된다. 규칙과 구문(322)는 구문 구조의 해석과 생성을 돕는 정적 정의를 포함한다. 예를 들어, C ++ 프로그래밍 언어에 대한 규칙 및 구문을(322)에 저장할 수 있다. 논리적 축소(323)는 코드로 작성된 논리를 줄여 양식을 단순화하여 상호 연결된 함수의 맵을 생성한다. 작성된 코드(324)는 최종 결과물인 실행 가능 프로그램이며, 코드 목표(332)는 입력이다. 복합적 목적 형식(Complex Purpose Format)(325)은 전체 목적을 나타내는 상호 연결된 하위 용도를 저장하기 위한 저장 포맷이다. 목적 연관성(326)은 어떤 종류의 목적을 가리키는 행동 및 유형의 기능에 대한 하드 코딩된 참조이다. 반복적 확장(Iterative Expansion)(327)은 세부 목표와 복잡성을 추가하여 목적 연관성을 참조하여 복잡한 목적으로 단순한 목표를 진화시킨다. 반복적인 이해(328)는 상호 연결된 모든 기능을 반복하며 목적 연관성(326)을 참조하여 해석된 목적을 산출한다. 외부 코어(329)는 알려지지 않은 외부 코드에 대한 논리적 목적을 유도하고 명시된 기능 코드 목표에서 실행 가능한 코드를 생성하기 위해 함께 작동하는 구문 및 목적 모듈에 의해 주로 형성된다. 외부 코드(330)은 LIZARD에 알려지지 않은 코드이며 기능 및 의도된 목적을 알 수 없다. 외부 코드(330)은 내부 코어에 대한 입력이지만 파생된 목적(331)은 출력이다. 목적(331)은 목적

모듈(36)에 의해 추정된 주어진 코드(330)의 의도이다. 그것은 복합 목적 형식(325)으로 반환된다.

[0267] 도 48은 내부 코어(334)가 시스템의 필수 핵심 기능을 어떻게 유지 하는지를 보여주는데, 이는 유지(318) 플랫폼을 통해 관련 사이버 보안 전문가(319)에 의해 직접 및 독점적으로 프로그래밍된다. 코어 코드(335)는 LIZARD를 실행하는데 필요한 기본적인 기초 지식이다. 코어(336)내에서 기본 체계 및 라이브러리(336)에는 압축 및 비교 기능과 같은 LIZARD를 작동시키는데 필요한 모든 기능이 있다. 코어(336)에서 스레드 관리 & 로드 균형(337)을 사용하면 LIZARD가 서버 클러스터를 효율적으로 확장 할 수 있으며 통신 및 암호화 프로토콜은 암호화 유형(예: AES, RSA 등)을 정의한다. 코어(336)에서 메모리 관리(339)는 LIZARD가 해석하고 처리한 데이터를 서버의 RAM(Random Access Memory)에서 효율적으로 관리 할 수 있다. 시스템 목표(336)는 보안 정책(340) 및 기업 목표(341)를 포함한다. 정책(340)은 사이버 보안 분석가 (또는 다수)에 의해 수동으로 설계되어, 사용자 정의 변수에 따라 동작하도록 참조될 수 있다. 그러므로 LIZARD는 안전하지 못하고 금지된 행동이 무엇인지 그리고 무엇이 허용되는지 정당화하는 표준을 가지고 있다. 예를 들어 기업 외부의 수신자에게 전자 메일을 보내지 못하게 하거나 세 번째 암호 입력 시도가 실패한 후 계정을 잠그는 것은 기업의 보안 정책(340)에 있을 수 있다. 기업 목표(341)는 기업이 달성하고자 하는 일반 인프라의 유형에 대한 보다 광범위한 특성을 정의한다. 목표(341)는 주로 LIZARD의 기능과 기업의 인프라 문맥과 관련하여 수행 해야 하는 기능에 대해 동적 쉘(313)의 자체 프로그래밍을 안내하는 데 사용된다.

[0268] 도 49는 동적 쉘(DS)(313)의 내부 동작을 나타낸다. LIZARD의이 섹션은 주로 인위적으로 지능적인 프로그래밍 모듈(반복 모듈)에 의해 조작된다. 외부 쉘(345)의 모듈은 전체 시스템의 의사 결정에 가벼운 영향을 미치는 새롭고 실험적인 모듈이다. 내부 쉘(344)는 LIZARD의 주체이다. 대부분의 지능형 기능이 작동한다. 새롭고 실험적인 알고리즘(343) '베타'는 소프트웨어 공간을 할당했는데, 여기서 새로운 모듈의 기능적 필요성은 인간, 인공지능 또는 둘 모두에 의해 프로그래밍되고 테스트 될 수 있다.

[0269] 도 50은 동적 쉘(313)에서 모듈을 지능적으로 수정, 생성 및 파괴하는 반복 모듈(IM)을 보여준다. 보안 성능에 대한 참조를 위해 인공 보안 위협 (AST)(17)을 사용하고 자동 코드 작성 방법론을 처리하기 위해 반복 코어(347)을 사용한다. 데이터 반환 릴레이(DDR)(317)에서는 LIZARD가 낮은 신뢰를 바탕으로 결정을 내려야 할 때 악의적인 공격 및 나쁜 행위자에 대한 데이터가 AST(17)에 전달된다. AST(17)는 시뮬레이션된 보안 위협이 있는 가상 테스트 환경을 만들어 반복 프로세스를 활성화한다. AST(17)의 인공적인 진화는 범죄적인 악의적 인 사이버 활동의 유기적 진화를 앞당기기에 충분하다. 정적 코어 클로닝(346)에서는 반 동적 외부 코어(329)를 포함하는 정적 코어(315)가 반복 안내의 기준으로 사용된다. 이 반복은 부분적으로 외부 코어(329)를 수정하기 때문에; 인위적으로 지능적인 루프에서 자체 프로그래밍이 전체 주기로 진행되었다. 반복 코어(347)은 동적 쉘(313)을 변경하기 위한 인공 보안 시나리오 및 시스템 목표 지침을 수신한다. 반복 코어(347)은 많은 반복을 생성한다. 인위적인 보안 테스트에서 최상을 수행하는 반복은 단계(348)에서 동적 쉘의 라이브 작동 반복이 되도록 업로드된다.

[0270] 도 51은 보안 향상을 위한 코드를 반복하는 주된 논리인 반복 코어(Iteration Core)(347)를 도시한다. 재귀적 반복(Rearseive Iteration)(350)에서는 기본 반복(356)을 대체하는 새로운 반복(355)과 함께 반복 코어(347)의 새 인스턴스가 호출된다. 이러한 전환은 코어 코드(335)의 부분 집합인 인스레드 관리 & 로드 균형(337)로부터 도출된 스레드 관리(349)에 의해 관리된다. 차분 수정 알고리즘(DMA)(353)은 내부 코어(334)로부터 구문/ 목적 자동화 프로그래밍 능력(351) 및 시스템 목적 지침(352)을 수신한다. 이들 두 입력은 기본 체계 및 라이브러리(336) 및 보안 정책(340)/ 기업 목표(341)와 상관 관계가 있다. 그런 다음 AST(17)에서 발견된 결함에 따라 기본 반복(356)을 수정하기 위해 이러한 코드 세트를 사용한다. 차등 논리가 적용된 후, 새로운 반복(355)이 제안되며, 반복 코어(347)가 재귀적으로 호출되고 AST(17)에서 동일한 테스트 과정을 거친다. 대기중인 보안 시나리오(360)에서는 모든 알려진 보안 지점에서 동적 쉘(313)에 대한 포괄적인 테스트를 집합적으로 수행하는 여러 시나리오가 있다. 활성 보안 시나리오 (361)에서는 현재 활성 보안 시나리오가 격리된 가상 실행 환경(357)에서 동적 셀(313)을 테스트하고 있다. 이러한 환경(357)은 실제 시스템과 완전히 별개의 가상 인스턴스이다. 인위적으로 생성된 악의적인 공격과 침입을 수행한다. 보안 결과 결함(362)는 가상 실행 환경(357)을 실행하는 동안 기본 반복(356)을 '통과 한'보안 위협을 나타내기 위해 시각적으로 표시된다. 그 후 발견된 임의의 결함(363)은 DMA(353)로 전송되어 그러한 결함을 생략하고자 하는 새로운 반복(355)의 생성을 용이하게 한다.

[0271] 도 52 내지 도 57은 차등 수정 알고리즘(DMA)(353)의 논리 프로세스를 도시한다. 현재 상태(365)는 상징적으로 상관된 모양, 크기 및 위치를 갖는 동적 쉘(313) 코드 세트를 나타낸다. 이러한 모양의 다른 구성은 보안 인텔리전스 및 반응의 다른 구성을 나타낸다. AST(17)은 잘못된 것으로 밝혀진 현재 상태(365)의 잠재적 응답을 제공하고 올바른 응답(즉 바이러스이므로 이 파일을 격리)을 제공한다. 공격 벡터(370)(모두 점선으로 표시된 화

살표)은 사이버 보안 위협에 대한 상징적인 표시 역할을 한다. 방향, 크기 및 색상은 모두 공격 경로, 멀웨어 크기 및 멀웨어 유형과 같은 가설적인 보안 속성과 관련된다. 공격 벡터는 코드 세트의 보안 응답을 나타내기 위해 코드 세트에서 상징적으로 '튀어 오른다'. 참조 A(Ref. A)(367)은 공격 벡터가 통과하도록 허용하는 특정 보안 구성을 보여 주며, 올바른 보안 응답 일 수도 있고 아닐 수도 있다. 참조 B(Ref. B)(368)은 잠재적으로 정확하거나 부정확한 동안 참조 A에 대한 대체 응답 유형을 보여주는 보안 구성에서 튀어 나오는 공격 벡터를 보여준다. 참조 C(Ref. C)(369)는 공격 응답 벡터를 발신지로 되돌려 보내는 보안 응답을 나타내며, 이는 정확한 보안 응답 일 수도 있고 그렇지 않을 수도 있다. 도 53에서, 수정 상태(354)는 동적 쉘(313)의 코드 블록으로부터 원하는 보안 응답을 생성하기 위한 차분 수정 알고리즘(353) 프로세스의 최종 결과를 나타낸다. 수정 상태(354)는 동적 쉘(313)의 새로운 반복(355)을 재귀적으로 반복하여(350) 생성된다. 현재(365)와 수정(354) 상태 간에 미묘한 차이가 있지만 이러한 차이로 인해 완전히 다른 공격 벡터(370) 응답이 발생할 수 있다. 참조 A(367)은 공격 벡터를 통과 시키지만 참조 A(371)(올바른 보안 응답)은 공격 벡터를 올바른 각도로 수신한다. 현재(365) 및 수정(354) 상태의 참조 B에 대한 공격 벡터 응답은 변경되지 않는다. 참조 C(373)의 경우, 공격 벡터는 참조 C(369)와는 다른 위치에도 불구하고 원래 소스로 되돌려 보내진다. 이 모든 공격 벡터 프레젠테이션은 보안 위협의 병목 관리를 설명하고 이에 대응한다. 도 54는 AST(17)에 의해 제공되는 일련의 공격인 AST 보안 공격 벡터(375)를 도시한다. 올바른 보안 응답(376)은 공격 벡터들(370)에 관한 원하는 보안 응답을 나타낸다. 정확한 보안 응답을 생성하는 코드 세트(모양)는 아직 알려지지 않은 이 단계에서 표시되지 않는다. 도 55는 올바른 동적 쉘 반응 공격(378)에 대한 열등한 보안 응답을 나타내는 현재 동적 쉘 반응 공격(377)을 도시한다. 이러한 수정 응답 (378)은 논리 추론 알고리즘(LDA)(197)에 의해 생성된다. 도 56은 LDA(197)가 어떻게 올바른 공격 응답 (378)과 일치하도록 정확한 보안 설정을 유추 하는지를 도시한다. 정적 코어(315)는 정정 공격 응답(378)을 생성하는 보안 프로그램을 구성할 수 있도록 LDA(379)에 시스템 체계/ 지침(352) 및 구문 / 목적 자동화된 프로그래밍 능력(351)을 제공한다. 동적 셸(313)의 기본 반복(356)은 단계(381)에서 LDA(379)에 제공된다. 이러한 반복은 표준 이하 및 비 효과적인 보안 응답을 생성하는 보안 응답 프로그램(382)으로 표현된다. 이러한 프로그램(382)은 LDA(379)에 대한 입력으로서 제공된다. LDA는 정적 코어(315)로부터의 구문 / 목적 기능(351)을 사용하여 올바르지 않은 보안 응답 프로그램(382)으로부터 구축하여 올바른 응답 공격(378)과 일치하게 한다. 보안 응답 프로그램(383)이 생성되고 동적 쉘(313)의 새로운 반복(355)으로 간주된다. 반복 코어(347)의 재귀적 반복(350)을 통해 프로세스가 계속 진행되면 동적 쉘(313)의 보안 기능이 지속적으로 업그레이드되어 모든 보안 정보는 AST(17)에서 사용할 수 있다. 도 57은 AST(17)가 정확한 보안 응답(384)과 함께 알려진 보안 결함(364)을 제공함에 따라 이 프로세스의 간략한 개요를 나타낸다. AST(17)가 알려진 보안 결함(364) 및 응답(384)을 제공 할 수 있지만, 그러한 정정 응답(384)을 생성하는 유효하고 실행중인 프로그램을 구성할 수 없다. 따라서, LDA(379)는 동적 쉘(313)의 이전(기본) 반복(356)을 사용하여 올바른 보안 응답 프로그램(385)라고하는 동적 쉘의 보다 우수하고 잘 갖춰진 반복(355)을 생성한다. '프로그램'이라는 단어의 사용은 동적 쉘(313) 내에서 작동하는 다양한 기능과 하위 모듈의 전반적인 기능을 나타낸다.

[0272] 도 58은 가상의 난독화의 개요를 보여준다. 가상 난독화 및 모의 데이터 생성의 다음 기능은 암호화된 클라우드 플랫폼에 배포되어 사이버 보안 직원이 거의 없는 중소기업에서 사용할 수 있다. 보안 시스템은 대기업용 데이터 센터에 직접 설치할 수도 있다. 이 경우, 멀웨어(385)는 인터넷(304)을 형성하고 산업 표준 방화벽/ 침입 탐지 시스템/ 안티 바이러스 등을 우회(bypass)한다. 보안 반복의 현재 상태에서 LIZARD(16)은 들어오는 코드 블록(385)의 의도/ 목적에 대한 신뢰도 평가가 낮다. 이러한 조건은 최악의 시나리오로 가정한다. 따라서 권한이 부여된 중요 데이터가 없는 무고한 프로세스를 줄이고 악성 코드가 중요한 데이터를 보유할 수 있는 위험을 방지하기 위해 문제의 코드(385)는 데이터의 절반이 지능적으로 혼합된 환경에 몰래 할당된다. 실제 시스템(388)은 일반적인 관리 액세스 요구 사항을 제외하고 실제 데이터(389)에 제한되지 않음을 나타낸다. 실제 시스템(388) 내에서 작동하는 모든 주체는 가상 격리(390)로 인해 부분적으로(391) 또는 완전(394) 모의 데이터 환경으로 쉽고 은밀하게 전송될 수 있다. 실제 데이터 동기화 장치(386)는 혼합 환경에 부여할 데이터와 우선 순위를 지능적으로 선택하는 2개의 층(다른 하나는 데이터 관리자(401)) 중에서 하나이다. 이런 식으로 매우 민감한 정보는 멀웨어가 의심되는 경우에는 액세스 할 수 없으며 잘 알려져 있으며 신뢰할 수 있는 것으로 입증된 코드에서만 사용할 수 있다. 모의 데이터 생성기(387)는 모조품 및 쓸데없는 데이터를 만들기 위한 템플릿으로 실제 데이터 동기화 장치(386)를 사용한다. 데이터 유형, 데이터 형식, 데이터 밀도, 데이터 세부 사항 등과 같은 속성은 실제 데이터(389)에서 모방되어 시스템에 잘 통합된 것으로 보이는 실제 데이터를 가진 데이터베이스를 만든다(관련성이 없고 이상한 데이터는 없음). 들어오는 외부 코드의 인식에 대한 확신의 위험은 LIZARD(16)가 선택한 난독화 수준에 영향을 미친다. 악의적인 코드에 대한 높은 확신은 대량의 모의 데이터(Mock Data)(394)가 포함된 환경으로의 할당을 호출한다. 악의적인 코드에 대한 신뢰가 낮으면 실제 시스템(388)(의심의 이익으로

간주됨) 또는 100 % 모의 데이터 환경(394)(기본적으로 비 신뢰로 간주 됨)에 할당 할 수 있다. 보안 행동의 이러한 사용자 지정 옵션은 내부 코어(334)의 서브 세트인 시스템 목표(336)의 서브 세트인 보안 정책(340)에 정의되어있다. 고도로 모니터링되는 네트워크 인터페이스(392)는 모의 데이터 (393)를 포함하는 환경에서 사용된다. 가상의 격리(390)와 함께 실제 시스템(388)과 같은 제한된 환경으로 유출되지 않도록 환경을 보호하는 데 사용된다. 이러한 격리(390)는 가상화 기술을 사용하여 RAM과 CPU 스레드를 혼합하여 각 환경을 자체 환경으로 분리한다 .

[0273] 도 59 내지 도 61은 가상 난독화의 모니터링 및 응답 측면을 보여준다. 이러한 시스템은 멀웨어 동작에 따라 멀웨어(385)를 모니터링하고 관리한다. 초기에 LIZARD는 현재의 반복화 수준에서 악성 코드일 수도 있고 그렇지 않을 수 도 있는 코드 블록을 인지한다. 멀웨어가 아닌 경우 LIZARD는 이를 50% 모의 데이터(391)로 구성된 실제 시스템의 가상 복제본으로 푸시한다. 이는 멀웨어가 아닌 것으로 밝혀지면 시스템 및 기업 기능이 크게 방해받지 않도록(즉, 잘못된 SSN이 제공되는 등) 수행된다. 도시된 예에서 코드 블록은 실제로 멀웨어(385)이지만이 단계에서 LIZARD는이 위협 및 개발 방법의 새롭고 알려지지 않은 특성으로 인해 여전히 확실하지 않다. 도 60은 LIZARD가 멀웨어인지 여부가 아직 확실하지 않기 때문에 멀웨어(385)가 가상으로 격리된(380) 모의 데이터 환경(391)에 숨겨지는 방식을 구체적으로 보여준다. 데이터 관리자(401)는 실제 데이터와 모의 데이터를 지능적으로 혼합하여 매우 민감한 데이터가 노출되지 않도록 한다. 관리자(401)는 멀웨어(385)에 의해 생성된 정보를 모의 데이터 저장소(400)에 업로드(402)하고(402) 이전에 저장된 모의 데이터를 다운로드(398)하여 실제 데이터(397)와 혼합한다. 이렇게하면 멀웨어는 실제 데이터 저장소(397)에 대한 쓰기 권한이 없으며 민감한 정보를 무시할 수 없다. 멀웨어(385)는 사실상 격리 됨(380)이므로 데이터 관리자(401)에만 노출된다. 이 가상 격리는 멀웨어가 데이터 관리자(401)를 우회하여 모든 실제 데이터(397)에 액세스하는 것을 금지한다. 행동 분석(403)은 의심스러운 코드 블록의 다운로드(398) 및 업로드(402) 동작을 추적하여 잠재적인 시정 조치를 결정한다. 분석(403)은 LIZARD의 원래 의혹을 확인하거나 거부하는데 도움이 되도록 멀웨어(385)가 자연스러운 형태로 작동하는 방식을 모니터링한다. LIZARD는 외부 코드가 실제로 멀웨어인지에 대한 최초의 의혹을 확인시켜 준다. 멀웨어(385)는 암묵적으로 그리고 은밀하게 비밀 전환 모듈(395)을 통해 100% 모의 데이터 가상 환경(394)으로 전송된다. 멀웨어가 이미 50 % 모의 데이터 환경(391)에서 증식 및 감염된 경우에만 예방 조치로 전체 가상 환경이 안전하게 파괴된다(멀웨어 포함). 이 단계에서 멀웨어(385)는 이제 민감한 정보에 노출되지 않고 모의 환경(394)에 완전히 숨겨지게 된다. 숨은 통신 채널을 통해 멀웨어와 홈베이스(즉, 하트 비트 신호) 간의 잠재적 통신을 모니터링하여 향후 동적 쉘(313) 반복을 잠재적으로 개선 할 수 있다. 이러한 멀웨어 거동 정보는 데이터 반환 릴레이(DRR)(317)를 통해 AST(17)로 전송되어 향후 반복에 도움이 된다. 이렇게 하면 DS(313)는 50 % 모의 데이터 환경(391)에 다시 배치하는 대신(훔친 합법적인 데이터와 관련하여 여전히 위험을 안고 있음) 유사한 멀웨어(385)에 대한 보다 확신 있는 결정을 내릴 수 있다.

[0274] 도 62 및 도 63은 데이터 회수 추적(399)이 의심되는 엔티티(415)로부터 업로드되고 의심되는 엔티티(415)로 업로드된 모든 정보를 추적하는 것을 도시한다. 이는 멀웨어에 잠재적으로 전송되는 중요한 정보의 보안 위험을 완화하기 위해 수행된다. 이 보안 검사는 또한 모의 데이터(Mock Data)(400)를 수신하는 합법적인 기업 프로세스의 병목 문제를 완화한다. 모의 데이터가 합법적인 기업 엔티티로 전송된 경우, 모의 데이터(Mock Data) 및 실제 데이터 (원래 요청된)가 대체품으로 전송된다. 콜백 트리거는 합법적인 기업 엔티티가 데이터가 위조가 아니라는 확인이 있을 때까지 특정 정보에 대한 조치를 보류 할 수 있도록 구현된다. 실제 데이터가 가상 혼합 환경 내의 멀웨어로 전송 된 경우 전체 환경 컨테이너가 멀웨어(385) 내부에서 안전하게 파괴된다. 파괴되기 전에 멀웨어가 소유하고있는 것으로 알려진 데이터와 관련된 비정상적인 활동에 대해 시스템 전체에 경고가 표시된다. 이 개념은 시스템 전반 모니터링(405)에서 나타난다. 부분적인 실제 데이터를 받은 엔티티가(동작 패턴을 분석 할 때) 멀웨어로 밝혀지면 멀웨어를 포함한 가상 환경이 안전하게 파괴되고 태그가 있는 실제 데이터의 비정상적인 활동에 대해 전사적 네트워크가 모니터링된다. 이렇게 하면 잠재적인 정보 유출이 방지된다. 모의 데이터 다운로드 추적(407) 및 모의 데이터 업로드 추적(408)에서; 가상 컨테이너에서 의심스러운 개체(415)와 주고받는 모의 데이터가 추적된다. 업로드 안전성에 대한 정보(410)에서, 처음에는 보호 장치로서 모의 데이터 수집부(400)에 기록된 데이터는 나중에 안전하다고 간주되므로 의심스러운 엔티티(415)의 업로드(402) 요청을 수행하기 위해 실제 데이터(412)에 기록 될 준비가 된다. 업로드 릴레이(411)는 그러한 표시된 안전 정보를 실제 데이터(412)로 전달한다. 합법적인 기업 엔티티 (멀웨어가 아닌)가 모의 데이터(400)를 수신 한 경우, 모의 데이터 존재의 정도에 대한 정보(413)이다. 실제 데이터 (412)는 모의 데이터를 정확하게 대체하기 위해 업로드 된다. 데이터 회수 트리거(414)는 혼합 데이터 환경이 잠재적으로 활성화되었다는 것을 나타내는 숨은 신호를 검사하는 합법적인 엔티티(및 부주의하게, 합법적으로 보이려고 시도하는 악의적인 엔티티)에서 수행되는 소프트웨어의 설치이다. 데이터 관리자(401)는 모의 데이터(400) (존재한다면)와 혼합되어야 하는 실제 데이

터(412)의 비율을 계산하는 데이터와 엔티티(415) 간의 중개자 인터페이스이다. 정보의 업로드(402) 및 다운로드(398) 스트림에서, 데이터 회수 트리거(414)가 데이터의 반전을 고려하기 위해 개별 패킷들/ 파일들이 (필요하다면) 표시된다.

[0275] 도 64 및 도 65는 데이터 회수 트리거(414)의 내부 동작을 도시한다. 행동 분석(403)은 잠재적인 시정 조치(419)를 결정하기 위해 의심스러운 엔티티(415)의 다운로드 및 업로드 동작을 추적한다. 실제 시스템(417)은 가상화된 환경 외부에 완전히 존재하며 모든 가능한 민감한 데이터를 포함하는 원래의 실제 데이터(412)를 포함한다. 모의 데이터(418)를 대체하는 실제 데이터는 실제 데이터가 필터링되지 않은(실제 데이터 동기화 장치(386) 전에 조차도) 데이터 회수 추적(399)에 제공되는 곳이다. 이러한 방식으로 실제 데이터 패치(416)는 모의 데이터를 이전의 의심스러운 엔티티(422)상의 실제 데이터로 대체하도록 만들어질 수 있다. 가상의 격리 환경(404)에 잠긴 데이터 관리자(401)는 데이터 회수 추적(399)으로부터 실제 데이터 패치(416)를 수신한다. 이 패치(416)에는 이전에 의심스러운 엔티티(422)(현재 무해하다고 알려짐)를 정확하고 사실이며 정확한 정보 상태로 변환하는 대체 지침이 포함되어 있다. 이러한 패치(416)는 후속하여 이전의 의심스러운 엔티티(422)로 전송되는 데이터 호출 인터페이스(427)로 전송된다. 다운로드된 데이터(420)는 모의 데이터 환경(404) 내에서 다운로드한 데이터이다(따라서 데이터는 부분적으로 또는 완전히 가짜이다). 고정 데이터(Fixed Data)(421)는 모의 데이터가 실제 데이터 패치(416)가 적용된 후에 대응하는 실제 데이터로 대체 된 곳이다. 행동 분석(403)에서 무해한 코드(409)가 멀웨어로 지워졌을때, 정정 동작(419)이 수행된다. 그러한 동작(419)은 이전의 의심스러운 엔티티(422)의 모의 데이터를 그것이 나타내는 실제 데이터(412)로 대체하는 것이다. 비밀 토큰 (Secret Token)(424)은 LIZARD에 의해 생성되고 할당되는 보안 문자열이다. 비밀 토큰(424)는 가상 난독화 시스템에 의심스러운 엔티티(415)가 합법적이며 해가 없다는 것을 증명하지 않는다. 대신 부분적으로(391) 또는 완전히(394) 모의 데이터 환경의 영향을 받았다는 것을 알고 있기 때문에 실제로 무해한 엔티티가 자신의 일을 진행하지 못하게 한다. 엔티티가 합법적인 한, 가짜 정보에 대한 행동과 실제 생활 문제(잘못된 주소 발송, 잘못된 직원 해고 등)에 대한 직원/ 소프트웨어 등의 물류상의 문제가 없어야 한다. 데이터 회수 트리거(414)는 합법적인 기업 기능 및 엔티티에만 존재한다. 기본적으로 합법적인 엔티티는 임베디드 서버 환경(404)에서 토큰(424) 존재에 대해 합의된 위치를 확인한다. 토큰이 누락(429 및 425)된 경우, 이는 합법적인 엔티티가 부분적으로 모의 데이터 환경(악성 프로그램이라는 위험 평가 때문에)에 놓여 졌을 가능성이 있음을 나타낸다. 그 후, 지연 인터페이스(426)를 갖는 지연 세션(428)이 활성화된다. 토큰이 발견(426과 424)되면, 서버 환경이 실제이므로 지연된 세션이 비활성화됨을 나타낸다(427). 지연 인터페이스(426)는 엔티티에 직접 사전 설치된 모듈이다. 모의 환경(404)에있는 것으로 표시되면, 지연된 세션이 활성화 될 것이다. 지연된 세션은 엔티티의 프로세스가 인위적으로 느려서 행동 분석(403) 시간을 부여하여 이 엔티티가 무해하거나 악의적인지 여부를 결정한다는 것을 의미한다. 실질적으로 그러한 지연은 동작 당 몇 초가 걸릴 것으로 예상된다. 실제로 악성 코드가 100 % 실제 데이터 시스템에 은밀하게 내장되어 있기 때문에 실제 멀웨어가 비밀 토큰의 사본(자체적으로 변경 및 재생성 할 수 있음)을 얻는 것은 극히 어렵다. 비밀 토큰이 발견되지 않는 경우에, 지연된 인터페이스(426)는 참여하고, 행위 분석(403)에 의해 실제 데이터 액세스가 재 부여 될 때까지 환자(patient)가 행동하는 것을 의미한다.

[0276] 도 66은 매우 민감한 데이터(고감도 데이터)를 필터링 하고 실제 데이터를 모의 데이터와 혼합하는 데이터 선택을 보여준다. 실제 데이터(412)는 고감도 데이터(431)를 필터링하는 실제 데이터 동기화 장치(386)에 제공된다. 필터 범위는 정적 코어(315)에 정의된 시스템 정책(430)에 따라 변한다. 이 모듈 (431)은 의심스러운 엔티티(415)가 존재하는 동일한 가상 환경에 민감한 정보가 도달하지 못하도록 한다. 데이터는 가상 환경(40)의 생성(434)시에 한번 필터링 된다. 생성 기준(433)을 통해, 필터링된 실제 데이터는 모의 데이터의 종류 및 양에 대한 기준으로서 사용된다. 모의 데이터 생성기(387)는 실제 데이터와 구별 할 수 없도록 설계된 가짜 데이터를 생성한다. 즉, SSN 배치. 호환성 시행(432)을 사용하면 생성된 모의 데이터가 실제 데이터와 호환되도록 검증되어 중복이 너무 많지 않고 데이터 유형이 누락되지 않도록 할 수 있다. 실제 데이터와 가짜 데이터를 수집하는 것은 의심의 여지없이 원활하게 병합된다. 즉 가짜 SSN과 실제 SSN은 겹치지 않는다(중복을 피하라). 가상 환경 생성기(434)는 모의 데이터의 비율, 이용 가능한 시스템 기능, 네트워크 통신 옵션, 저장 옵션 등과 같은 변수를 포함하는 가상 환경(404)의 빌딩을 관리한다. 데이터 기준(435)은 실제 데이터 대 가짜 데이터의 비율을 조정하기 위한 변수이다. 병합된 데이터(438)와 함께, 데이터는 데이터 기준(435)에 따라 병합된다. 병합 프로세스 중에 덜 민감한 것으로 표시된 실제 데이터가 더 민감한 느낌을 주는 모의 데이터와 병합된다. 비율 관리(437)은 원하는 모의 데이터 비율을 따르는 것처럼 병합중인 실제 및 모의 데이터의 양을 지속적으로 조정한다. 데이터는 의심스러운 엔티티(415)의 데이터 요청(440)에 따라 실시간으로 병합된다. 데이터는 요청 된 데이터(439)에 적절한 모의 데이터 비율로 리턴된다.

[0277] 도 67 및 도 68은 행동 분석(403)의 내부 작동을 보여준다. 목적 맵(441)은 전체 기업 시스템에 목적을 부여하

는 시스템 목표의 계층 구조이다. 이러한 목적은 소규모 네트워크, CPU 처리 및 저장 이벤트의 세분성까지도 할당된다. 선언된, 활동 및 코드베이스 목적은 의심되는 엔티티(415)가 주장하는 바가 무엇이든 선천적인 시스템 요구와 비교된다. 활동 모니터링(453)을 사용하면 의심스러운 엔티티의 저장소, CPU 처리 및 네트워크 활동이 모니터링된다. 구문 모듈 (Syntax Module)(35)은 그러한 기능(443)을 원하는 기능으로 해석한다. 그러한 기능들은 목적 모듈(36)에 의해 행동에서 의도된 목적으로 번역된다. 예를 들어, 코드 베이스 목적(446)은 연간 수익 보고서를 제출하는 것이지만, 활동 목적(447)은 "최고 유급 직원의 모든 SSN을 모으는 것"일 수 있다. 이 방법론은 특정 품목을 세관에 신고해야 하는 공항의 관세 부서와 유사하지만 세관에서는 가방을 검색한다. 코드베이스(442)는 의심스러운 엔티티(415)의 소스 코드/ 프로그래밍 구조이다. 컴파일된 폐쇄 소스 프로그램이므로 소스 코드를 공개하지 않는 엔티티는 시스템 정책(430)에 의해 시스템에 액세스하지 못하도록 차단 될 수 있다. 이러한 코드베이스(442)는 행동 분석(403)의 서브 세트로서 구문 모듈 (35)에 전송된다. 구문 모듈(35)은 코딩 구문을 이해하고, 프로그래밍 코드 및 코드 활동을 상호 연결된 함수의 맵(444)으로 감소시킬 수 있다. 이러한 기능(444)은 코드베이스(442) 및 활동(443)의 기능을 나타내고, 의심스러운 엔티티(415)의 인식된 '의도'를 생성하는 목적 모듈(36)로 전달된다. 목적 모듈(36)은 코드베이스 목적(446) 및 활동 목적(477)을 생성한다. 코드베이스 목적(446)은 LIZARD의 구문 프로그래밍 기능에서 파생된 엔티티(415)의 알려진 목적, 기능, 관할권 및 권한을 포함한다. 활동 목적(447)은 LIZARD의 저장, 처리 및 네트워크 활동(453)에 대한 LIZARD의 이해로 이해되는 엔티티(415)의 알려진 목적, 기능, 관할권 및 권한을 포함한다. 선언된 목적은 엔티티 자체에서 선언 한 엔티티(415)의 가정, 기능, 관할권 및 권한이다. 필요한 목적(445)은 기업 시스템이 요구하는 예상된 목적, 기능, 관할권 및 권한을 포함한다. 이는 회사의 필요를 충족시키기 위해 직원을 고용하는 것과 유사하다. 이로써 LIZARD는 의심스러운 엔티티(415)를 차단할 수 있다. 시스템에서 능력 및/또는 서비스가 절대적으로 필요하지는 않다. 4 개의 모든 목적(445 내지 448)은 비교 모듈 (449)에서 비교되어 엔티티(415)의 존재가 보장되고, 기업 시스템 내의 동작이 LIZARD가 시스템의 목표(336)에 대해 생산적으로 평가하고 이해한다. 4 가지 목적(445 내지 448) 사이의 불일치는 정정 동작(419)으로 이어지는 목적(450)의 시나리오의 차이를 유발한다. 정정 동작는 잠재적으로 의심스러운 엔티티(415)를 멀웨어(385) 또는 무해(409)로 표시 할 수 있다. 계속되는 조치는 가상 컨테이너를 안전하게 파기하거나 실제 데이터(모의 데이터 전용) 및 실제 기업 네트워크 액세스에 대한 액세스가 전혀 없는 새로운 가상 환경으로 멀웨어(385)를 신중하게 이동하는 것이다.

[0278] **비판적 사고 메모리 및 인식(CTMP)**

[0279] 도 69는 CTMP(22)의 주요 로직을 보여준다. CTMP의 주요 목표는 제 3자가 내린 결정을 비판하는 것이다. CTMP(22)는 여러 소스(예: I^2GE, LIZARD, 신뢰할 수 있는 플랫폼 등)의 인텔리전스를 상호 참조하고 인식과 현실에 대한 기대치를 학습한다. CTMP는 사안에 대한 객관적인 결정을 내릴 수 있는 자신의 능력이라고 평가하고 내부 신뢰도가 낮은 결정을 내리는 것을 삼가한다. 전 세계적으로 배포 된 에이전트의 군대 및 신뢰 플랫폼의 정보와 같이 들어오는 데이터 스트림은 모두 실행 가능한 데이터로 변환된다. 주관적 의견 결정(454)은 선택된 패턴 매칭 알고리즘(SPMA)(526)으로 알려진 입력 알고리즘에 의해 제공된 원래의 주관적 결정을 나타낸다. SPMA는 통상적으로 보안 관련 보호 시스템이지만, 어휘 객관성 마이닝(LOM)(추론 알고리즘) 및 영구 기부 방법(MPG)(세금 해석 알고리즘)에 대한 방법과 같은 다른 유형의 시스템을 제한하지 않는다. 입력 시스템 메타 데이터(455)는 SPMA(526)로부터의 원시 메타 데이터를 나타내며, 이는 알고리즘의 기계적 프로세스 및 그것이 어떻게 결정에 도달했는지를 기술한다. 이유 처리(456)는 특성의 속성을 비교함으로써 주장한 것을 논리적으로 이해할 것이다. 이유 처리의 하위 집합인 규칙 처리(457)에서 파생된 결과 규칙이 현재 문제의 범위를 결정하기 위한 참조 포인트로 사용된다. 비판적인 규칙 범위 확장자(Critical Rule Scope Extender, CRSE)(458)는 알려진 인식 범위를 취하여 인식의 비판적 사고 범위를 포함하도록 업그레이드한다. 정정 규칙(459)은 인식의 비판적 사고 범위를 사용하여 도출된 정정 규칙을 나타낸다. 메모리 웹(460)에서 시장 변수(시장 성과 (30) 및 이익 이력(31)) 로그는 충족 가능한 규칙을 검색한다. 모든 적용 가능하고 이행 가능한 규칙이 실행되어 투자 할당 중단 결정처리하기 위해 실행된다. 규칙 실행(RE)(461)에서, 혼돈 영역(613)의 메모리 스캔에 따라 현재대로 확인되고 충족된 규칙들이 실행되어 원하는 비판적 사고 결정을 생성한다. 이러한 규칙의 실행은 필연적으로 모호하지 않은 결과로 이어진다. 혼돈스럽게 복잡한 프로세스가 일관되지 않지만 생산적인 결과를 초래할 수 있는 반면, 논리적으로 복잡한 RE(461) 프로세스는 항상 규칙 집합에 따라 동일한 추론된 결과를 가져온다. 비판적 결정 출력(462)에서, 인식 관찰자 에뮬레이터(POE)(475)와 규칙 실행(RE) (461)에 의해 도달된 결론을 비교함으로써 CTMP의 전체 출력을 결정하는 최종 논리. 비판적 결정(463)은 가능한 한 객관적으로 하려고 하는 사안에 대한 의견 인 최종 결과이다. 로그 (Logs) (464)는 입력 알고리즘 (MPG)의 주관적인 견해에 영향을 미치지 않거나 편향없이 독립적으로 중요한 결정을 내리는 데 사용되는 원시 정보이다. 원시 인식 생산(RP2)(465)은 SPMA(526)에서 메타 데이터 로그를 수신하는 모듈이다. 이러한 로그는 파싱되고 이러한 알고리즘의 인식을 나타내는 인

식이 형성된다. 인식은 인식 복합 형식(PCF)에 저장되며, 인식 관찰자 에뮬레이터(POE)(475)에 의해 에뮬레이션된다. 인식의 적용 관점 (466)은 이미 SPMA (526)에 의해 적용되고 이용된 인식의 관점를 나타낸다. 자동 인식 발견 메커니즘(467)은 인식의 범위를 확대할 수 있도록 혼성 인식(인식의 적용 관점(466)에서 제공한 입력에 따라 형성됨)을 생성하는 창조성 모듈(18)을 활용하는 모듈을 나타낸다. 468은 컴퓨터 시스템이 사용할 수 있는 전체 인식 범위를 나타낸다. 비판적 사고(469)는 규칙 기반 사고의 외부 쉘 관할 구역을 나타낸다. SPAM(526)에 따라 잘 설정된 규칙뿐만 아니라 CTMP 내에서 파생된 새로운 정정 규칙(459)이 표시되는 규칙 실행(RE)(461)이 발생한다.

[0280] 도 70의 자기 비판적 지식 밀도(474)를 참조하면, 유입 원시 로그는 SPMA (526)에 의해 알려진 기술적 지식을 나타낸다. 이 모듈(474)은 보고 가능한 로그의 도달 범위를 벗어나는 잠재적인 알려지지 않은 지식의 범위 및 유형을 추정한다. 이 방법으로 CTMP의 비판적 사고 기능은 시스템에 의해 직접 알려지고 알려지지 않은 모든 관련 지식의 잠재적 범위를 활용할 수 있다. 인식 관찰자 에뮬레이터(POE)(475)는 관찰자의 에뮬레이션을 생성하고 관찰자 에뮬레이션의 그러한 변화들과 함께 인식의 모든 잠재적인 점들을 테스트/ 비교한다. 입력은 향상된 데이터 로그 이외에 잠재적인 인식 포인트들이다. 결과는 그러한 선택된 로그가 혼합된 최상의, 가장 관련성이 높으며 가장 신중한 관찰자에 따라 향상된 로그에 의해 생성되는 최종 보안 결정이다. 함축적 파생(477)을 참조하면, 이 모듈은 현재 적용된 인식의 관점(470)으로부터 연루 될 수 있는 인식 데이터의 관점을 도출한다. 중단 정정 동작(476)을 참조하면, 최종 정정 동작/ 주장 비평은 인식 관찰자 에뮬레이터(475)에 의해 생성된다.

[0281] 도 71은 CTMP의 의존 구조를 나타낸다. 자원 관리 및 할당(RMA)(479)을 참조하면, 조정 가능한 정책은 관찰자 에뮬레이션을 수행하기 위해 레버리지되는 인식들의 양에 영향을 준다. 선택된 인식의 우선 순위는 가중치에 따라 내림차순으로 선택된다. 그런 다음 정책은 백분율, 고정 수 또는 보다 복잡한 선택 알고리즘보다 차단을 선택하는 방식을 지시할 수 있다. 저장 검색(SS)(480)을 참조하면 데이터 향상 로그에서 파생된 CVF가 인식 저장(PS; Perception Storage)(478)의 데이터베이스 조회에서 기준으로 사용된다. 메트릭 처리(MP)(489)은 선택된 패턴 매칭 알고리즘(SPMA)(526) 투자 할당으로부터의 변수를 그러한 알고리즘의 지능으로부터의 '구원' 인식으로 리버스 엔지니어링 한다. 인식 추론(Perception Deduction, PD)(490)은 투자배분 응답의 일부와 해당 시스템 메타데이터를 사용하여 투자배분 응답에 대한 원래의 인식을 재현한다. 비판적 결정 출력(CDO)(462)은 CTMP 출력을 결정하기 위한 최종 논리를 지시한다. 메타데이터 분류 모듈(MCM)(488)을 참조하면, 디버깅 및 알고리즘 추적은 기존의 구문 기반의 정보 분류를 사용하여 별개의 카테고리로 구분한다. 그런 카테고리는 시장/ 세금 위험 및 기회와의 상관 관계를 통해 명확한 투자 배분 응답을 구성하고 생산하는 데 사용될 수 있다. 시스템 메타데이터 분리(SMS)(487)를 참조하면, 입력 시스템 메타 데이터(455)는 의미 있는 투자 배정 인과 관계로 분리된다. 포퓰레이터 논리(483)를 참조하면, 관련 시장/ 세금 위험, 기회 및 각각의 대응으로 모든 투자 배분을 포괄적으로 조정한다. 주제 탐색기(481)는 모든 적용 가능한 주제를 스크롤한다. 주제 포퓰레이터(482)는 피험자와 관련된 적절한 투자 위험 및 할당을 검색한다. 인식 저장(PS)(478) 인식은 관련 가중치 외에도 비교 가능한 가변 형식(CVF)을 인덱스로 사용하여 저장된다. 이는 데이터베이스가 입력 쿼리 조회로 CVF를 수신하도록 최적화되어 있고 그 결과가 여러 가지 인식이 될 것임을 의미한다.

[0282] 도 72를 참조하면, 함축적 파생(ID)(477)은 인식의 현재 알려진 관점로부터 연루 될 수 있는 데이터의 인식 관점을 도출한다. 자기 비판적 지식 밀도(SCKD; Self-Critical Knowledge Density)(492)를 참조하면, 유입되는 원시 로그는 알려진 지식을 나타낸다. 이 모듈은 보고 할 로그의 범위를 벗어나는 잠재적인 알려지지 않은 지식의 범위와 유형을 추정한다. 이 방법으로 CTMP의 비판적 사고 기능은 시스템에 의해 직접 알려지고 알려지지 않은 모든 관련 지식의 잠재적 범위를 활용할 수 있다. 메트릭 결합(493)에서, 인식은 메트릭의 카테고리로 분리된다. 메트릭 전환(494)에서, 개별 메트릭은 전체적인 인식 관점으로 역전된다. 메트릭 확장(ME)(495)에서, 다양한 인식 관점의 측정 기준은 개별 데이터베이스에 카테고리 별로 저장된다. 상한은 각 개별 메트릭 데이터베이스(DB)의 최고 지식으로 표시된다. 향상과 복잡성이 높아지면 메트릭은 인식의 관점으로 변환되어 비판적 사고를 위해 활용된다. 비교 가능한 가변 형식 생성기(CVFG; Comparable Variable Format Generator)(491)를 사용하면 정보 스트림이 비교 가능한 가변 형식(CVF)으로 변환된다.

[0283] 도 73은 CTMP의 의존 구조를 도시한다. 비판적 규칙 범위 확장자(CRSE)(458)에서는 알려진 인식을 활용하여 규칙 세트들의 비판적 사고 범위를 확장한다. 인식 매칭(Perception Matching)(503)에서, 규칙 구문 파생(CVF; Rule Variable Derivation)(504)으로부터 수신된 인식으로부터 비교 가능한 가변 형식(CVF)이 형성된다. 새로 형성된 CVF는 유사한 인덱스를 갖는 인식 저장(PS; Perception Storage)(479)에서 관련 인식을 검색하는데 사용된다. 잠재적 매칭은 규칙 구문 생성(RSG; Rule Syntax Generation)(505)으로 반송된다. 메모리 인식(MR; Memory Recognition)(501)에서, 혼돈의 영역(613)은 입력 데이터로부터 형성된다. 필드 스캐닝은 알려진 개념을

인식하기 위해 수행된다. 메모리 개념 인덱싱(Memory Concept Indexing)(500)에서 전체 개념은 인덱스로 알려진 개별 부분으로 개별적으로 최적화된다. 이들 인덱스는 문자 스캐너에 의해 혼돈의 영역(613)과 상호 작용하는데 사용된다. 규칙 실행 파서(RFP)(498)는 인식 태그로 규칙의 개별 부분을 수신한다. 각 부분은 메모리 인식(501)에 의해 혼란 필드 (613)에서 발견되거나 발견되지 않은 것으로 표시된다. 그런 다음 RFP는 모든 부분의 조합인 모든 전체 규칙이 규칙 실행(RE)(461)의 이점을 위해 혼돈의 영역(613)에서 충분히 인식되었는지 논리적으로 추론할 수 있다. 규칙 구문 형식 분리(RSFS; Rule Syntax Format Separation)(499)에서 정정 규칙은 유형별로 구분되고 정리된다. 따라서 모든 동작, 속성, 조건 및 객체는 별도로 스택된다. 이것은 시스템으로 하여금 혼돈의 영역(613)에서 발견된 부분과 그렇지 않은 부분을 식별하게 한다. 규칙 구문 파생(504)에서 논리적 '흑백'규칙은 메트릭 기반 인식으로 변환된다. 여러 규칙의 복잡한 배열은 다양한 그라디언트의 여러 메트릭을 통해 표현되는 단일의 균일한 인식으로 변환된다. 규칙 구문 생성기(RSG; Rule Syntax Generation)(505)는 인식 형식(perception format)에 저장된 이전에 확인 된 인식을 수신하여 인식의 내부 메트릭 구성과 연관시킨다. 이러한 그라디언트 기반 메트릭 측정 값은 원래 인식의 입/ 출력 정보 흐름을 에뮬레이션하는 이진 및 논리 규칙 집합으로 변환된다. 규칙 구문 형식 분리(RSFS)(499) 정정 규칙은 관찰되는 객체의 현실에 맞는 정확한 규칙 집합을 나타낸다. 정정 규칙은 유형별로 구분되고 정리된다. 따라서 모든 동작, 속성, 조건 및 객체는 별도로 스택된다. 이것은 시스템으로 하여금 혼돈의 영역(613)에서 발견된 부분과 그렇지 않은 부분을 식별하게 한다. 타고난 논리적 추론(506)은 논리적인 원칙을 사용하므로, 어떤 종류의 규칙이 인식 내의 많은 메트릭의 그라디언트를 정확하게 나타낼 것인지 추론하기 위해 오류를 방지한다. 예를 들어, 아날로그 사인파(라디오 주파수 등)를 디지털 단계로 변환하는 것과 같다. 전반적인 추세, 위치 및 결과는 동일하다. 그러나 아날로그 신호는 디지털로 변환되었다. 메트릭 맥락 분석(507)은 메트릭의 인식 내에서 상호 연결된 관계를 분석한다. 특정 메트릭은 다른 정도에 따라 달라질 수 있다. 이 문맥화는 '디지털'규칙 집합 형식 내에서 규칙이 갖는 미러된 상호 연결된 관계를 보완하는 데 사용된다. 입력/ 출력 분석(508)은 각각의 인식(회색) 또는 규칙(흑백)의 입력 및 출력의 차분 분석을 수행한다. 이 모듈의 목표는 변환 후 입력 및 출력이 가능한 한 유사하거나 동일하게 유지되도록 하는 것이다(회색에서 검정/ 흰색으로 또는 그 반대로). 기준 계산(509), 입력 규칙의 기준과 작업을 계산한다. 이것은 규칙 집합 이면의 '동기 부여'로 번역 될 수 있다. 규칙은 묵시적으로 또는 명시적 정의로 이해할 수 있는 이유로 구현된다. 따라서 '디지털'규칙이 구현된 이유에 대한 묵시적인 이유를 계산하면 같은 이유로 동일한 입력/ 출력 기능을 추구하는 인식 내에서 메트릭 구성을 정당화 할 수 있다. 규칙 형성 분석(Rule Formation Analysis)(510)은 규칙의 전체 구성/ 구성 및 이들이 서로 어떻게 상호 작용 하는지를 분석한다. 메트릭이 '아날로그'인식 내에서 갖는 미러된 상호 연결 관계를 보완하는 데 사용된다. 규칙 구문 형식 전환(RSFC)(511)에서 규칙은 규칙 구문 형식(RSF)(538)의 구문을 준수하도록 분류되고 분리된다.

[0284] 도 74는 CTMP에서 지능형 정보를 처리하기 위한 최종 논리를 도시한다. 최종 논리는 직관적인/ 통찰력 있는 및 사고력 있는/ 논리적인 모드(인식 관찰자 에뮬레이터(POE)(475) 및 규칙 실행 (RE)(461)) 각각으로부터 지능형 정보를 수신한다. 방향 결정 비교(DDC; Direct Decision Comparison)(512)에서는 직감과 사고의 결정을 비교하여 확증을 확인한다. 중요한 차이점은 메타 데이터가 아직 비교되지 않는다는 것이다. 왜냐하면 그들이 같은 방식으로 동의한다면 그 이유를 이해하는 것이 중복적이기 때문이다. 단자 출력 제어(TOC)(513)는 직관적 인(514)와 사고력 있는(515)의 두 모드들 사이의 CTMP 출력을 결정하기 위한 마지막 로직이다. 직관적인 의사 결정(514)은 인식을 활용하여 비판적 사고에 종사하는 CTMP의 두 가지 주요 섹션 중 하나이다. 인식 관찰자 에뮬레이터(POE)(475)를 참조하라. 사고력 있는 결정(515)는 규칙을 활용하여 비판적 사고에 관여하는 CTMP의 두 가지 주요 섹션 중 하나이다. 규칙 실행(RE)(461)을 참조하라. 인식(516)은 내부 형식(518)에서 정의된 형식 구문에 따라 직관적인 결정(158)으로부터 수신된 데이터이다. 실행된 규칙들(517)은 사고력 있는 결정(515)에서 수신한 데이터로 규칙 실행(RE)(461)에서 적용 가능한(실행 가능) 규칙 집합의 모음이다. 이러한 데이터는 내부 형식(518)에 정의된 형식 구문에 따라 전달된다. 내부 형식(518)을 사용함으로써, 메타 데이터 분류 모듈(MCM)(488)은 공지된 일관된 포맷으로 표준화되었으므로 두 입력의 구문을 인식 할 수 있다 이는 CTMP 내에서 내부적으로 사용된다.

[0285] 도 75는 CTMP를 전체적으로 나타내는 단일 단자 출력으로의 직관적인/ 통찰력 있는 / 논리적인 동화의 두 가지 주요 입력을 보여준다. 중요한 결정 + 메타-메타데이터(521)은 내부 형식(518)에 정의된 구문에 따라 인식(516) 또는 실행된 규칙들(517)을 전송하는 디지털 반송파이다.

[0286] 도 76은 원래의 선택 패턴 매칭 알고리즘(SPMA)(526)에서 발생하는 지능적인 사고의 범위를 도시한다. 입력 변수(524)는 이유 및 규칙 처리를 위해 고려되는 초기 재정/ 세금 할당 변수이다. CTMP는 그들을 비판하고 인위적으로 지적인 제 2의 의견이 되려는 의도를 가지고 있다. 가변적 입력(525)은 보안 결정을 정의하는 입력 변수를 수신한다. 그러한 변수는 합리적인 시정 조치가 무엇인지 CTMP가 판단할 기준을 제공한다. 변수에 더하기, 빼기

또는 변경이 있으면; 적절한 정정이 결과적인 정정 동작에 반영되어야 한다. CTMP의 중요한 목표는 입력 변수의 변화를 적절하고 정확하게 반영하는 정정 동작의 정확하고 중대한 변경을 식별하는 것 이다. 선택 패턴 매칭 알고리즘(SPMA)(526)에서, 선택된 패턴 매칭 알고리즘은 그 자신의 기준에 따라 가장 적절한 동작을 식별하려고 시도한다. 결과 출력 형식(527)은 초기 입력 변수(168)을 사용하여 SPMA(526) 의해 생성된 결과이다. SPMA(526) 의사 결정에서 파생된 규칙은 '현재 규칙'으로 간주되지만 반드시 '정정 규칙'은 아니다. SPMA(526)에서 제공하는 로그 정보에 따라 속성 병합(528)을 사용하여 이유 처리(456)는 SPMA(526)에 따라 현재 지식 범위를 계속 진행한다.

[0287] 도 77은 인식 및 규칙을 통해 CTMP로 수행된 비판적 사고에 대해 병행되는 종래의 SPMA(526)를 도시한다. 오해 동작(531), 선택된 패턴 매칭 알고리즘(SPMA)(526)은 완전히 정확한 정정 동작을 제공할 수 없었다. 이것은 SPMA(526)의 원래 프로그래밍이나 데이터에서 확인되지 않은 근본적인 가정에 기인한다. 이 예에서 3D 개체를 입력 변수로 사용하고 정정 동작을 취하면 SPMA(526)이 고려하지 않은 차원/ 벡터가 있음을 알 수 있다. 적절한 행동(532), 비판적 사고는 SPMA(526)이 검사용 벡터로 생략한 3 차원으로 간주된다. 3 차 차원은 수행된 인식 검사의 모든 여분의 관점 때문에 비판적 사고(469)에 의해 고려되었다. 정정 규칙(533)을 참조하면, 비판적인 규칙 범위 확장자(CRPS; Critical Rule Scope Extender)는 이전에 고려되지 않은 인식 관점(즉, 제 3 차원)을 이용함으로써 규칙 집합의 이해 범위를 확장시킨다. 현재 규칙들(534)을 참조하면, 현재의 정정 동작 결정의 유도된 규칙은 SPMA(526)의 이해 또는 이해의 부족(또는 정정 규칙과 비교하여)을 반영한다. 입력 규칙은 SPMA에서 제공하는 기본 이해 범위를 설명하는 SPMA(526)에서 파생되었다. 이것은 SPMA 526이 평면 재무 할당 개념에서 단지 2 차원만을 이해함으로써 설명된다.

[0288] 도 78은 중요한 통찰 및/또는 변수를 생략할 수 있는 종래의 현재 규칙들(534)과 대조하여 정정 규칙들(533)이 어떻게 생성되는지를 도시한다. 혼돈 영역 파싱(CFP)(535)에 의해, 로그의 형식은 혼돈의 영역(613)로 알려진 단일 스캔 가능한 유닛으로 결합된다. 추가 규칙들(536)은 이미 확립된 정정 규칙들(533)을 보충하기 위해 메모리 인식(MR)(501)으로부터 처리된다. 인식 규칙들(Perceptive Rules)(537)을 참조하면, 관련성이 있고 대중적이라고 여겨지는 인식이 논리적 규칙으로 변환되었다. 인식(원래의 인식 형식에서)이 많은 '회색 영역'을 정의하는 많은 복잡한 메트릭 관계를 가지고 있다면, '흑백' 논리적 규칙은 n 차 복잡도 확장으로 '회색'영역을 포함한다. 규칙 구문 형식(Format Syntax Format)(538)은 효율적인 저장 및 변수 쿼리를 위해 최적화 된 저장 형식이다.

[0289] 도 79 내지 도 80은 인식 매칭(PM; Perception Matching)(503) 모듈을 설명한다. 메트릭 통계(539)와 관련하여, 통계 정보는 인식 저장(PS)(479)에서 제공된다. 이러한 통계는 메트릭, 내부 메트릭 관계 및 메트릭 증가율 등의 인기 추세를 정의한다. 일부 일반적인 통계 쿼리(예 전체 메트릭 인기 순위)가 자동으로 실행되고 저장된다. 그 외 보다 구체적인 쿼리(메트릭 X 및 Y와 관련하여)는 PS(479)에서 실시간으로 요청된다. 메트릭 관계 홀드 아웃(540)은 메트릭 관계 데이터를 보유하므로 통합된 출력으로 푸시 될 수 있다. 오류 관리(541)는 임의의 개별 메트릭에서 유래하는 구문 및/또는 논리적 오류를 구문 분석한다. 메트릭들 분리(542)는 각각의 개별 메트릭을 입력 인식(544)이었던 단일 유닛에서 결합되기 때문에 격리시킨다. 입력 인식(Input Perception)(544)은 시각, 후각, 촉각 및 청각의 측정 지표로 구성되는 인식의 구성 예이다. 노드 비교 알고리즘(Node Comparison Algorithm, NCA)(546)은 2 개 이상의 CVF 중 노드 구성을 수신한다. CVF의 각 노드는 속성의 크기를 나타낸다. 유사성 비교는 개별 노드를 기준으로 수행되며 집계 분산이 계산된다. 이것은 효율적으로 계산 된 정확한 비교를 보장한다. 노드 별 또는 집계 가중이든 상관없이 더 작은 분산 번호는 더 가까운 일치를 나타낸다. CVF(Comparable Variable Formats)(547)는 CVF의 다양한 구성을 보여주는 시각적인 표현이다. 출력(550)은 인식 매칭(PM)(503)에 대한 터미널 출력이다. 노드 비교 알고리즘(Node Comparison Algorithm: NCA)(546)에서 어떤 노드들이 중첩되는지는 매칭 결과로서 유지되고, 따라서 전체 결과는 단계(550)에서 제출된다.

[0290] 도 81 내지 도 85는 규칙 구문 유도/ 생성을 보여준다. 원시 인식들- 직관적 사고(아날로그)(551)은 인식이 '아날로그'형식에 따라 처리되는 곳이다. 원시 규칙들- 논리적 사고(디지털)(552)는 규칙이 디지털 형식에 따라 처리되는 곳이다. 재정 할당 결정에 관한 아날로그 형식(553) 인식은 단계없이 부드러운 곡선의 그라디언트에 저장된다. 재무 할당 결정과 관련된 디지털 형식(554) 원시 규칙들은 '회색 영역'이 거의 또는 전혀 없는 단계로 저장된다. 원래의 규칙들(555)은 데이터 내용면에서 정정 규칙들(533)과 동일하다. 차이점은 원래의 규칙들(555)이 규칙 구문 형식 분리(RSFS)(499)에 의해 메모리 인식(501)을 통해 혼돈의 영역(613)과 상호 참조 할 수 있는 보다 동적인 형식으로 변환된다는 것 이다. 인식된 규칙 세그먼트들(556)은 메모리 인식(501)에서 인식한 원래의 규칙들(555)의 규칙이다. 이것은 원래 정정 규칙(533)(행동, 속성, 조건 및 객체와 같은)을 구성하는 개

별 세그먼트 중 어느 것이 혼돈의 영역(613)에서 인식되었는지를 나타내므로 잠재적으로 논리적으로 충족된 규칙이 될 수 있다. 보안 중단 결정(Security Override Decisions)(557)은 정정 동작이 수행되도록 하는 규칙 실행(Rule Execution; RE)(461)에 의해 생성된 최종 결과이다. 이러한 정정 동작은 비판적 결정 출력(Critical Decision Output: CDO)(462)에서 수행되는 더 큰 정정 동작 로직의 서브 세트인 단자 출력 컨트롤(TOC; Terminal Output Control)(513)로 더 보내진다. 실행되지 않은 규칙들(Unfulfilled Rules)(558)은 논리적 종속 관계에 따라 혼돈의 영역(Chaotic Field)(613)에서(규칙 실행 파서(Rule Fulfillment Parser)(498)에 따라) 충분히 인식되지 않은 규칙 세트이다. 마찬가지로, 실행된 규칙들(517)은 CDO(462)에 의해 분석된 논리적 의존성에 따라 혼돈의 영역(613)에서 충분히 이용 가능하다고 인식되어왔다. 제 3(타사) 데이터베이스 솔루션(559)는 버퍼, 캐시, 디스크 저장소, 스레드 관리, 메모리 관리 및 기타 일반적인 기계 데이터베이스 기능을 관리하는 하드웨어 인터페이스 소프트웨어이다. 실행 디버거(560)는 충족되지 않은 규칙의 이유를 찾는다. 혼돈의 영역(613)이 충분히 여유있지 않았거나 규칙 집합이 본질적으로 비논리적이었거나 둘 중 하나이다. 규칙 집합이 비논리적이면 일정한 정확도 내에서 즉각적으로 확인할 수 있다. 그러나 혼돈의 영역(613)의 잠재적 예비성을 입증하려면, 불충분한 조사 수행의 오류에 빠지지 않도록 여러 조사가 이루어져야 한다.

[0291] 도 86 내지 도 87은 규칙 구문 형식 분리(RSFS; Rule Syntax Format Separation)(499) 모듈의 작동을 보여준다. 이 모듈에서 정정 규칙들(502)은 유형별로 분리되고 구성된다. 따라서 모든 동작, 속성, 조건 및 객체는 별도로 스택된다. 이것은 시스템으로 하여금 혼돈의 영역(613)에서 발견된 부분과 그렇지 않은 부분을 식별하게 한다. 동작(561)과 관련하여, 이미 수행되었을 수 있는 동작을 나타내는 4 개의 규칙 세그먼트 데이터 유형 중 하나가 수행되고, 활성화 등을 고려하고 있다. 속성들(562)과 관련하여, 동작, 조건 또는 객체 등 다른 속성을 나타내는 4 가지 규칙 세그먼트 데이터 유형 중 하나이다. 조건(563)과 관련하여, 논리 연산 또는 연산자를 나타내는 4 개의 룰 세그먼트 데이터 유형 중 하나 (즉, x 및 y이면 z, x 또는 z, y이면 z 등)이다. 객체(564)와 관련하여, 동작(561) 및 속성(562)과 같이 적용된 속성을 가질 수 있는 타겟을 나타내는 4 개의 룰 세그먼트 데이터 유형 중 하나 이다. 처리 단계(565)에서, 지금까지 수집된 관계 유도 결과가 출력으로서 제출되고 그 후 프로그램은 종료된다. 처리 단계(566)는 한번에 하나의 항목을 룰 세그먼트를 통해 반복한다. 처리 단계(567)는 규칙 세그먼트들(즉, 동작들(561), 객체들(564) 등) 간의 각각의 개별 관계를 해석하고 기록한다. 따라서, 각각의 개별 관계가 수집되어 단계(565)에서 출력을 위해 준비된다. 순차적 스캐닝(Sequential Scanning)(568)은 RSF(538)의 각 유닛을 '[분할 (DIVIDE)]' 마커로 분할한다. 주제와 RSF 538의 접착제도 분리되어 파싱된다. 분리 출력 (569)은 개별 주제 및 내부 주체 관계가 스캐너에 의해 유지되는 곳이다. 이들은 전체 RSF(538)가 순차적으로 스캔 될 때 즉시 출력되도록 보내진다. 분리된 규칙 포맷(570)은 분리 출력(569)으로부터 개별 규칙 세그먼트(즉, 동작 (561), 객체 (564) 등)를 포함하기 위한 전달 메커니즘이다. 분리된 규칙 형식(570) 사용은 두 가지 주요 정보 전달 포인트에서 강조 표시된다. 먼저 규칙 구문 형식 분리(Rule Syntax Format Separation)(499)(사전 메모리 인식 단계로 간주됨) 및 메모리 인식(Memory Recognition)(501)의 출력으로 강조 표시된다(사후 메모리 인식 단계).

[0292] 도 88은 규칙 실행 파서(RFP)(498)의 동작을 도시한다. 이 모듈은 인식 태그로 규칙의 개별 세그먼트를 수신한다. 각각의 세그먼트는 메모리 인식(MR; Memory Recognition)(501)에 의해 혼돈 필드(613)에서 발견되거나 발견되지 않은 것으로 표시된다. 그러면 RFP(498)는 모든 부분들의 조합이 어느 전체 규칙이 충분히 인식되었는지를 논리적으로 추론 할 수 있다. 혼돈의 영역(613)은 규칙 실행(RE)(461)을 수행한다. 큐 관리(QM)(561)는 구문적 관계 복원(SRR; Syntactical Relationship Reconstruction) 모듈을 이용하여 각각의 개별 부분을 가장 논리적인 순서로 분석한다. QM(561)은 메모리 인식(MR)(501) 결과에 액세스 할 수 있으므로 바이너리 예/ 아니오 플로우 질문에 응답하고 적절한 조치를 취할 수 있다. QM은 모든 룰 세그먼트를 단계적으로 검사한다. 하나의 세그먼트가 혼돈의 영역(613)에서 누락되어 있고 다른 세그먼트와의 적절한 관계가 아니라면, 룰 세트는 충족되지 않은 것으로 플래그된다. 모든 검사 단계가 통과하면 룰 세트는 완료됨(522)으로 플래그된다. QM 스테이지(571)는 규칙 세그먼트 '객체 C'가 혼돈의 영역(613)에서 발견되었는지를 검사한다. QM 스테이지(572)는 다음 적절한 세그먼트가 원래의 '객체 C'이고, 또한 메모리 인식(MR)(501)에 따른 혼돈의 영역(613)에서 발견된다. 조건 B 및 동작 A 각각에 대해 동일한 논리가 QM 스테이지(573 및 574)에 적용된다. 이러한 세그먼트 표시(A, B, C 등)는 프로그램의 핵심 로직의 일부가 아니지만 예상되는 일반적인 사용법을 표시하는데 사용되는 일관된 예제를 참조한다. 완전히 재구성된 룰세트(575)의 수신은 룰 세트가 성취 될 수있는 것으로 가정 된 큐 관리(576)의 성취된 룰세트 출력 및 구문적 관계 복원 (SRR) 모듈(497)에 의해 주어진 룰 세그먼트의 연관을 필요로 한다.

[0293] 도 89 내지 도 90은 실행되지 않은 규칙의 이유를 찾으려는 실행 디버거(560)를 표시한다. 혼돈의 영역(613)이 충분히 여유가 있지 않거나 규칙 집합이 본질적으로 비논리적이다. 규칙 집합이 비논리적이면 일정한 정확도 내

에서 즉각적으로 확인할 수 있다. 그러나 혼돈의 영역(613)의 잠재적인 여유를 확보하기 위해서는 설문 조사의 불충분함을 피하기 위해 여러 조사가 이루어져야 한다. 영역 희박성 조사(577)는 혼돈의 영역(613)이 충분히 풍부하거나 규칙 집합의 변수 구성을 트리거하지 않는지 여부를 구체적으로 검사한다. 스캔(578)은 혼돈의 영역(613) 내의 관련 규칙 파트의 존재를 검사한다. 조사 DB (579)는 가까운 장래의 참조를 위해 조사 결과를 저장한다.

[0294] 조건부(580)는 조사 DB(579)가 포화되거나/가득 채워졌는지를 검사한다. 즉, 긍정적이거나 부정적 결과가 나오는 스캔에도 불구하고 규칙 파트들(Rule Parts)에 대해 가능한 모든 검사가 수행되었음을 의미한다. 가능한 모든 스캔이 수행된 경우, 결론(581)이 관련된다: 전체 혼돈의 영역(613)의 희소가 규칙 집합이 충족되지 않은 것으로 분류된 이유이다. 가능한 모든 스캔이 수행되지 않은 경우, 결론(582)가 관련된다: 조사가 불완전하고, 혼돈의 영역(613) 희박성이 실현되지 않게 되는 원인인지를 확실하게 알리기 위해서 혼돈의 영역(613)의 더 많은 섹터들이 스캔 될 필요가 있다. 논리적 불가능성 시험(Logical Impossibility Test)(583)은 규칙 집합 내에서 본질적으로 불가능한 논리적 종속성이 존재 하는지를 검사하여 이것이 불완전한 것으로 분류되도록 한다. 예를 들어, 객체(584) '미혼남자(Bachelor)'는 속성(585)이 '결혼함'으로 지정되어 있어 고유의 모순이 발생한다. 시험(583)은 용어들(584 및 585)의 사전 정의들을 결정한다. 내부 규칙 일관성 검사(588)는 모든 속성들이 대응하는 객체와 일치되고 연관되는지를 검사 할 것이다. RSF(538) 형식의 '미혼남자(Bachelor)'(584) 정의는 객체(586) '남자(Man)'의 부분 정의에 기여하고 '결혼한(Married)'(585) 정의 (RSF(538) 형식에도 있음)는 객체(587) '두사람(Two People)'의 부분적인 정의에 기여한다. 검사(588)의 결론은 객체(586) '남자(Man)'가 객체(587) '두사람(Two People)'을 잠재적으로 포함하고 있는 한, 정의(586과 587)가 모두 호환된다. 규칙 관련성 전환(589)을 사용하여 공정한 조건을 변환하여 비교 테스트를 수행한다. 이러한 전환은 두 번째 정의('결혼한')가 첫 번째 정의 ('미혼남자')의 맥락에서 이해되도록 한다. 따라서 결론(591)은 같은 사람이 현재 결혼을 하고(590) 동시에 현재 결혼하지 않을(592) 수 없는, 고유의 모순을 포함하는 규칙을 도출한다.

[0295] 도 91은 규칙 실행(RE)(461)을 도시한다; 현재와 같은 것으로 확인되고 메모리의 혼돈의 영역(613)의 스캔에 따라 수행된 규칙들은 실행되어 원하는 관련된 비판적 사고 결정을 내린다. 룰 세트의 전환을 추적하는데 사용되는 체커보드 평면이 있다. 보드상의 객체는 주어진 보안 상황의 복잡성을 나타내며 '보안 체커 보드'를 가로지르는 객체의 움직임은 보안 룰세트의 응답에 의해 관리되는 보안 상황의 진화를 나타낸다. 단계 1(593)은 RSF(538) 정보는 체커보드 평면상의 모든 관련 객체의 초기 시작 위치를 정의하므로, 동적으로 계단식 보안 상황의 시작을 정의한다. 이것은 동적 보안 정책을 다루는 규칙의 논리적인 '위치'를 설명하기 위해 상징적으로 사용된다. 2 단계 (594) 및 6 단계(598)은 특정 보안 상황의 위치 및 범위를 수정하는 적용되는 보안 규칙을 나타내는 객체 변형을 나타낸다. 예를 들어 2 단계와 6 단계에서 객체를 변환하면 암호화된 파일을 나타낼 수 있다. 3 단계(595)는 보안 응답 전략의 일부로 오프 사이트 위치로 중요한 파일을 실제로 이동시키는 것과 일치할 수 있는 체커보드 위의 개체 이동을 보여준다. 4 단계(596) 및 5 단계(597)는 공통의 제 3 객체로 2 개의 객체를 병합하는 프로세스를 도시한다. 이 규칙의 적용 예는 효율적이고 안전하게 관리되는 정보 전송을 용이하게 하기 위해 병합되는 두 개의 분리 및 격리된 LAN이다. 규칙 실행(RE)(461)이 완료되면 정정 규칙(533)과 현재 규칙(534)의 결과가 다르다. 이는 선택된 패턴 매칭 알고리즘 (SPMA)(526)에서 생성된 덜 중요한 결과와 대조적으로 CTMP가 수행한 비판적 사고 이점을 보여준다. 모양들, 색들 및 위치들 모두는 보안 변수, 발생 및 응답을 상징적으로 나타낸다(실제 보안 개체보다는 설명하기 쉽기 때문에). SPMA는 CTMP와는 다른 최종 형상 위치뿐만 아니라 오각형과 유사하지만 다른(오렌지색 대 노란색) 색상 차이를 생성했다. 이것은 모든 입력 로그가 처리를 위해 통과하는 복잡한 조건문 - 룰 세트 구성으로 인해 발생한다. 이것은 다양한 플레이어 변수(높이, 힘 등)와 당구시험을 어떻게 시작하느냐는 완전히 다른 결과의 공 위치로 이어질 수 있는 것과 유사하다. CTMP는 또한 보라색 사각형을 큐브로 변형 시켰는데, 이는 SPMA(526) 또는 인간이 결코 예상하거나 고려하지 않은 차원과 인식을 고려하는 능력을 상징적으로 보여준다(CTMP의 설명을 통해). 최종 보안 중단 결정(599)은 정정 규칙들(533)에 따라 수행된다.

[0296] 도 92 및 도 93은 알파벳과 같은 연속 된 정보의 '사슬'에 대한 읽기와 쓰기의 효율성을 극대화하는 최적화된 정보 저장 방법인 순차적 메모리 구성을 보여준다. 메모리 액세스 포인트(600)에서, 각 노드(601)의 폭은 기억된 객체(노드)에 대한 관찰자의 직접 액세스 가능성을 나타낸다. 순차적으로 기억된 알파벳 순서에서 'A'는 시퀀스의 첫 번째 노드이므로 가장 쉽게 액세스 할 수 있는 지점이다. 문자의 E, H 및 L은 하위 시퀀스 'EFG', 'HIJK'및 'LMNOP'의 '리더'이기 때문에보다 쉽게 직접 액세스 할 수 있다. 접근성 범위(602) 각 문자는 관찰자에게 직접적인 메모리 액세스 포인트를 나타낸다. 액세스 가능성 범위가 넓을수록 시퀀스 노드 당 더 많은 액세스 지점이 있고 그 반대가 사실임을 나타낸다. 시퀀스가 '순서대로'만 참조되고 무작위로 선택된 노드에서는 참조되지 않을수록 시퀀스 크기에 비해 액세스 가능성 범위가 좁아진다. 이는 연속성의 크기에 따라 보다 효율적

인 메모리 기억을 가능하게 한다. 중첩된 서브 - 시퀀스 계층(603)과 함께, 강한 불균일성을 나타내는 시퀀스는 상호 접속하는 일련의 더 작은 서브 - 시퀀스로 구성된다. 알파벳 'ABCD', 'EFG', 'HIJK', 'LMNOP'은 모두 기억된 순서대로 독립적으로 존재하지만, 전체적으로 알파벳을 상호 연결하고 형성하기 때문에 알파벳 순서는 이러한 행동을 크게 나타낸다. 이러한 유형의 메모리 저장 및 참조는 마스터 시퀀스의 특정 노드에 가끔 또는 자주 액세스하는 경우 훨씬 효율적이다. 이러한 방식으로 전체 시퀀스의 시작부터 스캐닝을 피할 수 있으므로 시간과 리소스의 효율성을 높일 수 있다. 이것은 모든 검색에서 첫 페이지에서 책을 스캔하는 것이 아니라 장에 따라 스캔되는 책과 유사하다. 극도로 불균일한(605) 범위에서는 모든 노드에서 일관성 없는 액세스 지점이 있다. 즉 체인과 같이 상호 연결된 중첩 된 하위 시퀀스가 무거운 구성으로 되어 있음을 의미한다. 극도로 불균일한 시퀀스는 중간 순차적이지만 아직 여러 메모리 액세스 지점(중첩 된 하위 시퀀스 계층)을 가져야 함을 의미한다. 극도로 불균일한(605) 예는 알파벳으로, 어떤 문자로 시작하는지에 따라 암기하기가 쉽지 않다. 극도로 균일한(607) 범위에서는 모든 노드에서 일관된 액세스 지점이 있다. 즉, 체인처럼 상호 연결된 중첩된 하위 시퀀스로 구성되지 않는다. 극도로 균일한 시퀀스는 극히 순차적(노드 전체에 일관되게 거의 액세스 지점이 없음)이거나 또는 극히 비 순차적(노드 전체에서 지속적으로 큰 액세스 지점)임을 의미한다. 극도로 균일한(607) 예는 과일의 모음이다. 열거하고 강조한 시퀀스가 거의 없으며 상호 연결된 하위 시퀀스가 없다. 적정하게 균일한(606) 범위에는 초기 대형 액세스 노드가 있으므로 처음부터 내용을 암기하는 것이 가장 효율적이다. 그러나 주요 내용은 또한 선형이며, 중첩된 하위 시퀀스 계층이 없고 단일한 큰 시퀀스가 있음을 나타낸다. 적정하게 불균일한(604) 범위는 전체적으로 선형적이고 일관된 액세스 지점에서 벗어나지 않는다. 이것은 일관되고 뒤집을 수 있는 컬렉션을 준수하는 동시에 더 세밀하고 덜 정의된 중첩된 하위 시퀀스 레이어가 있음을 나타낸다. 적정하게 불균일한(604)의 동작을 나타내는 정보의 예는 자동차 제조업체의 카탈로그 일 수 있다. 스포츠카, 하이브리드 및 SUV와 같은 범주가 정의 될 수 있지만 잠재적인 고객이 별도의 범주 지정에도 불구하고 SUV와 스포츠카를 비교할 수 있기 때문에 목록을 열거하거나 기억해야 하는 방법에 대한 강한 편견은 없다.

[0297] 도 94는 과일과 같은 비 순차적인 관련 아이템의 정보 저장을 다루는 비 순차 메모리 구성을 도시한다. 과일 모음에서 정보를 읽는 방법은 강한 순차 순서가 있는 알파벳과 달리 읽어야하 는 순서는 매우 명확하지 않다. 메모리 구성(608)은 모든 과일에 대해 일관되게 균일한 액세스 노드를 보여주며 비 순차적 조직을 나타낸다. 608의 조직은 가역성이 비 순차적 배열과 일정한 범위를 나타내는 방법을 보여준다. 이 예에서는 과일 당의 메모리가 노드별로 상대적으로 넓은 액세스 포인트로 표시되어 비 순차적임을 나타낸다. 과일의 순서가 뒤섞일 때 동일한 균질성이 존재하는데 이는 과일의 가역적 순서를 나타낸다. 대조적으로, 알파벳과 같은 순차 시리즈는 일반적인 암송과는 반대로 뒤로 암송하는 것이 훨씬 더 어렵다. 일반적인 과일의 목록은 이 현상이 나타나지 않으며 순차적인 목록에서보다 순차적인 목록의 외부에서 자주 참조된다는 것을 나타낸다. 중심 주제 및 연관(609)에서는이 과일 목록에 연속성이 없으므로 동일한 일련의 과일이 반복되지만 중심(중심 객체)이 다르게 반복된다. 중심은 기본 주제를 나타내며, 남아있는 과일은 중심 주제가 정의되지 않은 경우보다 쉽게 접근 할 수 있는 메모리 이웃으로 작용한다. 강한 이웃(610A)에서는 사과가 일반적인 과일 임에도 불구하고 철자가 겹치기 때문에 다른 일반적인 과일보다 파인애플과 더 강한 연관성이 있다. 따라서 더 많은 메모리와 연관된 것으로 간주된다. 약한 이웃(610B)에서는 파인애플이 열대 과일이기 때문에 오렌지와 바나나(일반 과일)와의 연관성이 적다. 파인애플은 열대성 겹침 때문에 망고로 참조 될 가능성이 더 크다. 그래프 포인트(612)는 과일 계열의 극히 약한 순차가 노드(601) 액세스에서 극도로 강한 균일성을 유도하는 방법을 보여준다.

[0298] 도 95 내지 도 97은 알려진 개념을 인식하기 위해 혼돈의 영역(613) 스캐닝이 수행되는 메모리 인식(MR)(501)을 도시한다. 혼돈의 영역(613)은 '화이트 노이즈'정보에 임의로 담긴 개념의 '영역'이다. 자발적으로 CTMP 시스템에 알려지며 '야생에서'예측할 수 없는 것으로 간주된다. 메모리 인식의 목적은 필드를 효율적으로 스캔하여 알려진 개념을 인식하는 것이다. 메모리 개념 보유(Memory Concept Retention)(614)는 인식 가능한 개념들이 저장되고 필드 검사를 위해 인덱싱되고 참조 될 준비가 된다. 이 그림에서는 야채 이름 철자의 단순화된 예를 사용하여 시스템을 쉽게 이해한다. 그러나 이 예제는 훨씬 더 복잡한 시나리오에 대한 비유로 사용될 수 있다. 실생활의 보안 예를 들어, 이것은 카메라 피드에서 시민과 군인을 인식하고 구별하는 것을 포함 할 수 있다. 사이버 보안의 경우, 알려지고 기억된된 트로이 목마, 백도어를 인식하고, 보안 화이트 노이즈(로그들)의 바다에서 탐지하는 것을 포함 할 수 있다. 3 개의 문자 스캐너(615)에 의해, 혼돈의 영역(613)은 표적에 대응하는 3 개의 문자 세그먼트들에 대하여 스캐닝되고 검사된다. 예를 들어 'PLANT'는 대상이며 스캐너는 3 글자마다 점진적으로 필드를 따라 이동한다. 스캐너가 전진 할 때마다 'PLA', 'LAN'및 'ANT'세그먼트는 'PLANT'라는 단어의 하위 집합이므로 확인된다. 그럼에도 불구하고 'LAN'과 'ANT'라는 단어는 대상이 되는 독립적인 단어에 해당된다. 따라서 이 세 글자 세그먼트 중 하나가 필드에서 발견되면 'LAN'또는 'ANT'의 전체 대상이 발견되었거나 'PLANT'의 하위 집합이 발견되었을 수 있다. 동일한 개념이 5 문자 스캐너(616)에 적용되지만, 이번에는 영역 전체에서

모든 전진로 검사되는 세그먼트가 전체 단어는 'PLANT'이다. 'LAN'및 'ANT'와 같은 대상은 5 글자 스캐너와 함께 작동하려면 최소 5 자의 대상이 필요하므로 생략된다. 혼돈의 영역(613)은 다양한 비율(3, 5 또는 그 이상의 문자 스캐닝)로 스캐닝을 위해 분할되며, 이러한 비율은 다양한 레벨의 스캐닝 효율성 및 효능을 제공한다. 스캐닝의 범위가 감소함에 따라 (글자의 양이 적을수록) 정확도는 증가한다(그 반대도 마찬가지이다). 스캐너의 필드 영역이 증가함에 따라 더 큰 문자 스캐너가 정확도를 희생하면서 인식을 수행하는데 효율성은 증가한다(대상이 얼마나 작은 지에 달려 있음). 메모리 개념 인덱싱(MCI; Memory Concept Indexing)(500)을 사용하면 617 단계는 처리되지 않은 메모리 개념에 대한 응답으로 스캐너 크기(3, 5 또는 그 이상)를 변경한다. MCI(500)은 사용 가능한 가장 큰 스캐너로 시작하여 617 단계에서 점진적으로 감소하므로 작은 메모리 개념 목표의 잠재적 존재를 확인하기 위해 더 많은 컴퓨팅 리소스를 찾을 수 있다. 618 단계는 이용 가능한 메모리 개념을 순환시켜 620 단계에서 그들의 인덱스(3 또는 5와 같은 적절한 길이에 적합한 작은 세그먼트)가 도출될 수 있도록 한다. 개념 홀드 아웃(624)에 메모리 개념이 아직 존재하지 않으면 619단계는 작업의 물류 흐름에 따라 메모리 개념이 생성된다. 621 단계는 620 단계로부터 유도된 인덱스를 홀드 아웃(624)에 할당한다. MCI(500)의 프로그래밍된 전체 순환이 계속되면 MCI에 처리되지 않은 문자 스캐너가 부족한 경우 분기점에 도달하면 홀드 아웃(624)이 비어있는 경우 빈 (null) 결과(622)를 제출하거나 모듈러 출력(623)으로써 비어있는 홀드 아웃(624)을 제출한다. 혼돈의 영역(613)의 섹션은 도면 부호 625 내지 628의 범위이다. 섹션 625 및 626은 5 글자 스캐너로 수행되는 스캔을 나타내며 섹션 627 및 628은 3 글자 스캔을 나타낸다. 스캔(625)은 5 글자의 너비를 가지며 6 글자 대상 'TOMATO'를 확인한다. 2 개의 5 문자 세그먼트는 이전에 MCI(500)에서 인덱스된 'TOMAT'및 'OMATO'에서 일치했다. 이들 각각은 6 글자 단어 중 5 문자 일치에 해당하며 83 %에 해당한다. 이 분율/ 백분율은 메모리 개념 'TOMATO'에 비해 167 %(637)에 누적적으로 추가되므로 'TOMATO'개념이 혼돈 영역(613)에서 성공적으로 발견되었다. 스캔(626)은 'EGGPLANT'의 메모리 개념 타겟을 가지며, 2 개의 중요한 세그먼트는 'GGPLA'및 'PLANT'이다. 'GGPLA'는 독점적으로 'EGGPLANT'의 진정한 의미를 나타내지만 'PLANT'는 그 자체로 메모리 개념의 목표 인 거짓 긍정의 잠재력을 소개한다. 시스템이 혼돈의 영역(613)에 존재하는 것으로 'PLANT'를 인식하기 위해서는 필드에서 'EGGPLANT'만이 유일하게 실제로 인식 가능한 메모리 개념이 거짓 긍정으로 분류된다. 그러나 시스템의 프로그래밍은 잘못된 긍정 시나리오를 우회할 수 있다. 'GGPLANT'는 63 %의 일치를 나타내며 'EGGPLANT'의 맥락에서 'PLANT'는 63 %를 기여하고 목표 'PLANT'는 100 %를 기여한다. 일치 항목이 집계로 추가되면 'EGGPLANT'는 총 점수 125%(63% + 63%)(638)를 받고 대상 'PLANT'는 100%(639)를 받는다. 따라서 스캐너는 혼돈의 영역(613)에 대한 정확한 해석을 성공적으로 유지했다. 스캔 (627)의 너비는 3 글자이며 세그먼트 'TOM'을 인식하여 총 일치가 50 %(640)로 이어진다. 이것은 스캔(625) 필드에 존재하는 것과 동일한 대상이지만 스캔 너비 (5 대신 3)의 차이로 인해 신뢰도가 약한 (50 % 대 167 %) 일치가 발견되었다. 따라서 MCI(500)의 설계에는 정밀도와 소비된 컴퓨팅 리소스 간의 올바른 균형을 맞추기 위해 여러 스캔 계층이 포함된다. 스캔(628)은 또한 3 개의 문자들의 폭을 통합하는데, 이번에는 2 개의 잠재적 인 거짓 긍정 접선들(636)이 있다. 필드의 실제 컨셉이 'CARROT'인 반면, 'CAR'과 'ROT'컨셉은 필드에 존재하는 것으로 간주된다. 이제 스캐너는 혼돈의 영역(613)에 있는 정확한 개념을 식별해야 한다. 가까운 문자로 수행된 후속 스캔으로 확인된다. 결과적으로 스캐너는 다른 위치에 있는 지표의 확증 때문에 개념이 'CARRO'가 아니라 'CAR'또는 'ROT'가 아닌 'CARROT'로 인식한다. 'CAR'641의 100 % 합성 일치와 'ROT'643의 100 % 합성 일치는 모두 'CARROT'642의 200 % 합성 일치에 손실된다.

[0299] 도 98 및 도 99는 영역 해석 로직(FIL; Field Interpretation Logic)(644 및 645)를 보여주는데, 이는 적절한 결과로 다양한 너비의 스캐너를 관리하기 위한 실행 계획을 운영한다. 일반 범위 스캔(629)은 큰 문자 스캔으로 시작된다. 이러한 유형의 검사는 작은 규모의 정확성을 희생하면서 적은 리소스로 넓은 범위의 필드를 탐색 할 수 있다. 따라서 더 작은 글자 스캐너는 필드의 보다 구체적인 범위를 위임 받아 필요에 따라 정확도를 향상시킨다. 특정 범위 스캔(630)은 중요도가 있는 영역을 찾을 때 사용되며 '확대'해야 한다. 일반적인 상관 관계는 스캔을 위해 선택한 필드 범위가 작을수록 작은 종류의 스캐너(글자 수가 적음)이다. 따라서 비용이 많이 드는 이중화된 위치에서 비용이 많이 들지 않는 스캔을 수행 할 수 있다. FIL의 섹션(645)은 스캐너 결과에 대한 반동 실행 계획을 표시한다. 특정 스캐너가 혼돈의 영역(613)에서 메모리 개념을 추가로 인식하면 이는 영역 범위(631)(613의 섹터)에 메모리 개념의 고밀도 포화가 포함되어 있으며 작은 폭의 스캐너들로 특정 범위에서 '확대'할 가치가 있음을 나타낸다. 따라서 필드 범위가 30 % (632) 인 5 글자 스캐너는 "증가된 '추가'인식"(634)으로 간주되는 초기 결과인 경우 10 %(633) 필드 범위의 3 글자 스캐너를 활성화한다. 634의 '추가'는 FIL 섹션(644)에서 수행된 최초 인식에 보충되는 인식을 나타낸다.

[0300] 도 100 및 도 101은 자동 인식 발견 메커니즘(Automatic Perception Discovery Mechanism, APDM)(467)을 보여준다. 관찰자(646)은 디지털 또는 사람의 관찰자를 대표하면서 다수의 인식을 통해 동일한 객체를 인식 할 수

있다. 관찰 가능한 객체(Observable Object)는 잠재적인 사이버 보안 사례 시나리오를 설명하기 위해 사용된다. 인식의 관점(Angle of Perception)(647)은 관찰 가능한 객체가 2 차원으로 표현 될 때 관찰 가능한 객체에 대한 제한된 정보 범위를 산출한다. 인식의 관점(Angle of Perception) B(648)은 3 차원을 포함하므로 더 많은 정보를 제공한다. 인식의 관점 (Angle of Perception) C(649)의 결과는 독창적인 혼성화 프로세스인 창조성(18)이 현대 병렬 처리 능력에 의해 레버리지되고 있기 때문에 우리의 제한된 사고 능력으로는 알 수 없다. 비판적 사고 알고리즘은 관점 A 및 B의 메트릭을 결합하여 새로운 반복(653)을 형성함으로써 인간의 이해 귀 또는 반복 복잡성 + 효능 및 CPU 시간 및 힘 사이의 기하 급수적(안정되지 않음) 관계를 넘어서는 더 많은 형태의 인식을 생성할 수 있다. 인식의 관점(Angle of Perceptions)(650)은 범위, 타입, 강도 및 일관성(651)을 포함하되 이에 국한되지 않는 여러 척도로 구성되어 있다. 이러한 지표는 전반적인 인식을 구성하는 인식의 여러 측면을 정의한다. 위의 예보다 범위가 더 복잡해질 수 있으므로 창조성 모듈에서 생성된 다양한 인식의 변형이 있을 수 있다. 인식 가중치(652)는 인식 관찰자 에뮬레이터(POE)(475)에 의해 인식이 얼마나 많은 상대적 영향을 받았는지를 정의한다. 두 가지 입력 인식의 이 가중치는 새로운 반복 인식(653)의 가중치를 정의하는 동안 고려된다. 이 새로운 반복 인식(653)은 이전 세대의 인식들: A + B 으로부터 영향을 받은 혼성화된 메트릭들을 포함한다. 이러한 새로운 인식의 관점은 잠재적으로 보안 소프트웨어가 은밀한 공격을 탐지하는 생산적인 새로운 이점을 제공한다. 인식들의 세대는 시행 착오와 지능적인 선택의 조합을 통해 혼성화를 위해 선택된다. 특히 새로 반복된 인식이 보안 문제에 대한 통찰력을 제공하는데 쓸모가 없다는 것을 증명하면 사용을 위해 강조 될 수는 있지만 유용한 통찰력을 제공할 수 있다면 결코 완전히 알려지지 않으므로 지워지지는 않는다. 따라서 컴퓨팅 전력 자원과 보안 인텔리전스의 교환은 경험이 풍부하다.

[0301] 도 102는 선택된 패턴 매칭 알고리즘(SPMA)(526)으로부터 메타 데이터 로그를 수신하는 모듈인 원시 인식 생산(RP2)(465)을 도시한다. 이러한 로그들은 파싱되고 이러한 알고리즘의 인식을 나타내는 지각이 형성된다. 인식은 인식 복합체 형식(Perception Complex Format, PCF)에 저장되며, 인식 관찰자 에뮬레이터(Perception Observer Emulator, POE)에 의해 에뮬레이트된다. 시스템 메타 데이터 분리(SMS)(487)는 보안 응답/ 변수 쌍들(654)의 출력을 제공하며, 이는 적절한 정정 조치가 트리거 변수(예를 들어, 주체, 위치, 행동 분석 등)와 결합될 때 보안 원인- 결과 관계를 확립한다. 비교 가능한 가변 형식(547)은 비 그래픽 용어들(655)로 표현된다. 이러한 인식 수집물들 각각은 CVF(547)를 형성하기 위해 특정한 가중 영향을 갖는 다양한 인식을 가지고 있다.

[0302] 도 103은 비교 가능한 가변 형식 생성기(CVFG; Comparable Variable Format Generator)(491)의 논리 흐름을 도시한다. CVFG에 대한 입력은 생성된 CVF(547)의 노드 구성에 의해 표시되어야 하는 데이터를 나타내는 임의의 데이터 모음인 데이터 배치(658)이다. 659 단계는 데이터 배치(658)에 의해 정의된 각각의 개별 유닛을 통해 순차적인 전진을 수행한다. 데이터 유닛은 660 단계에서 최종 CVF(547)에 의해 참조된 것과 동일한 구성의 정보를 갖는 노드 포맷으로 변환된다. 노드는 CVF의 구성 요소이며 다른 CVF와 비교하여 효율적이고 정확한 비교 평가를 수행 할 수 있다. CVF는 비교 최적화 특성(노드)을 제외하고는 되돌릴 수 없는 MD5 해시 합과 같다. 그런 다음, 변환 된 노드는 665 단계에서 그 존재를 검사 할 때 일시적으로 노드 홀드 아웃(661)에 저장된다. 이들이 발견되지 않으면 662 단계에서 생성되고 663 단계 에서 발생 및 사용과 같은 통계 정보로 업데이트된다. 664 단계에서 홀드 아웃(661)이 있는 모든 노드가 조립되어 CVF(547)와 같은 모듈 출력으로 푸시된다. 생성기가 실행된 후 홀드 아웃(661)이 비어 있으면 널(null) 결과가 반환된다(618).

[0303] 도 104에서, 노드 비교 알고리즘(NCA)(667)은 미처리 CVF(547)로부터 판독된 2 개의 노드 구성(666, 668)을 비교하고 있다. CVF의 각 노드는 속성의 크기를 나타낸다. 유사성 비교는 개별 노드를 기준으로 수행되며 집계 분산이 계산된다. 이것은 효율적으로 계산된 정확한 비교를 보장한다. 노드 별 또는 집계 가중이든 상관없이 더 작은 분산 번호는 더 가까운 일치를 나타낸다. 발생할 수 있는 비교 모드에는 부분적 매치 모드(PMM)와 전체 매치 모드(WMM)의 두 가지 모드가 있다. 하나의 CVF에 활성 노드가 있고 비교 후보에서 발견되지 않는 경우(노드가 휴면 상태인 경우) PMM의 경우 비교에 불이익을 주지 않는다. 모드 적용 가능성 예: 나무 A와 숲 A를 비교할 때 나무 A는 숲 A 내에 존재하는 가장 가까운 일치 나무 B를 찾는다. WMM을 사용한 CVF에 활성 노드가 있고 해당 비교 후보에서 찾을 수 없는 경우(노드 휴면 상태), 비교가 벌어진다. 모드 적용 가능성 예: 나무 A와 숲 A를 비교할 때 나무 A와 숲 A가 직접 비교되고 중첩 및 구조적 유사성이 큰 차이가 있으므로 일치하는 항목이 없다.

[0304] 도 105 및 도 106은 입력 시스템 메타 데이터(484)를 의미 있는 보안 원인-결과 관계로 분리하는 시스템 메타 데이터 분리(SMS)(487)를 도시한다. MCM(488)의 출력으로서, 로그들의 프로그래밍 요소는 672 단계에서 개별적으로 검색된다. 673 단계에서 MCM의 개별 카테고리가 보안 응답과 보안 변수(보안 로그들) 사이의 관계의 보다 상세한 구성을 얻는 데 사용된다. 그런 분류 (674)는 단계 669, 670 및 671에서 동화된다. 주체 스캔 / 동화

(669)를 통해 보안 상황의 주제/ 의심 항목은 범주화 모듈의 사전 분류 컨테이너 및 원시 분석을 사용하여 시스템 메타 데이터에서 추출된다. 주체는 보안 응답/ 변수 관계를 유도하기 위한 주요 참조 포인트로 사용된다. 주체는 사람, 컴퓨터, 실행 가능한 코드, 네트워크 또는 기업의 범위가 될 수 있다. 그러한 파싱된 주체(682)는 주체 저장(679)에 저장된다. 위험 요소 스캔/ 동화(670)를 사용하면 보안 범위의 위험 요소가 사전 분류 컨테이너와 카테고리 모듈의 원시 분석을 사용하여 시스템 메타 데이터에서 추출된다. 위험은 그러한 위험을 나타내거나 노출되는 타겟 주체와 관련된다. 위험은 잠재적인 공격 지점, 공격 유형의 취약성 등으로 정의 될 수 있다. 이러한 위험은 주체 인덱스(683)에서 관련 주체에 대한 연관성을 통해 위험 저장(680)에 저장된다. 응답 스캔/ 동화(671)를 사용하여, 입력 알고리즘에 의해 만들어진 보안 상황의 응답은 카테고리화 모듈로부터 사전 카테고리 컨테이너 및 원시 분석을 사용하여 시스템 메타 데이터로부터 추출된다. 응답은 그러한 응답을 받을 자격이 있다고 주장되는 보안 주체와 관련이 있다. 응답은 승인/ 차단/ 플래그/ 격리/ 난독화/ 신호 모방/ 응징 등의 범위를 가질 수 있다. 이러한 응답은 주체 인덱스(683)에서 관련 주체에 대한 연관성과 함께 응답 저장(681)에 저장된다. 그런 다음 저장된 정보는 포퓰레이터 로직(PL; Populator Logic)(483)을 포함한다.

[0305] 도 107 및 도 108은 메타 데이터 분류 모듈(MCM)(488)을 도시한다. 형식 분리(Format Separation)(688)에서, 메타 데이터는 인식된 형식의 규칙 및 구문에 따라 분리되고 카테고리화 된다. 이러한 메타 데이터는 인식 가능한 형식에 따라 조립된 것이어야 하며, 그렇지 않으면 처리를 위해 메타 데이터가 거부된다. 로컬 형식 규칙 및 구문(689)에는 미리 포맷된 메타 데이터 스트림을 MCM 모듈이 인식 할 수 있도록 하는 정의가 포함되어 있다. 로컬은 메타 데이터의 관련성과 존재로 인해 이전에 선택된 '형식'을 의미한다. 디버깅 추적(485)은 사용되는 변수, 함수, 방법 및 계층과 각각의 입력 및 출력 변수 타입/ 내용을 제공하는 코딩 레벨 추적이다. 전체 함수 호출 체인(다른 함수를 호출하는 함수)이 제공된다. 알고리즘 추적(486)은 알고리즘 분석과 결합된 보안 데이터를 제공하는 소프트웨어 레벨 추적이다. 결과 보안 결정(승인 / 차단)은 해당 결정(정당성)에 도달 한 경로와 각 요소가 보안 결정을 내리는 데 기여한 적절한 가중치와 함께 제공된다. 이러한 알고리즘 추적(486)은 686 단계에서 이들 보안 결정 정당화 각각을 통해 MCM의 사이클링 모드를 유도한다. 그러한 정당화는 컴퓨터 로그 구문에서(사람이 직접 작성한 것과는 대조적으로) 특정 보안 응답이 작성된 방법 및 이유를 정의한다. 인식 가능한 형식(687)은 미리 정의되고 CMTP와 호환되는 표준 구문 형식이다. 따라서, 입력 시스템 메타 데이터 (484)로부터의 포맷 선언이 인식되지 않으면, 모듈러 널(null) 결과가 리턴된다(618). SPMA(526)의 프로s그래머는 CTMP에서 인식할 수 있는 표준 형식으로 메타 데이터(484)를 코딩해야 한다. 이러한 형식은 JSON 및 XML 등과 같이 독점적일 필요가 없으며 CTMP에만 독점적일 필요는 없다. 가변적 홀드 아웃(Variable Holdout)(684)은 처리 변수가 하나의 범주에서(674)로 유지되기 때문에 최종 통합 출력으로 모두(685) 동시에 제출될 수 있다. 675 단계는 디버깅 추적(485) 및 알고리즘 추적(486)인 입력 정보의 2 개의 주요 지점들 사이의 비교 검사를 수행한다. 이러한 비교는 코딩 수준에서의 정당화의 발생을 추적하여 그러한 보안 정당성이 발생한 이유와 MCM에 대한 결과가 될만한 가치가 있는지를 더 잘 이해한다. 이 단계는 CTMP의 잠재적 비판을 전체적으로 검증하기 위해 모든 보안 정당화에 대한 추론을 보장하고 코딩 수준에서도 결정을 잘 이해할 수 있도록 사전 예방 조치이다. 유사하게 위험 증거는 단계 676에서 추적 데이터 디버깅으로 확인된다. 677 단계에서 SPMA에 의해 호출된 모든 기능에 대해 메타 데이터가 검사된 후 사용 가능한 기능적 목적 및 정당성이 인식 가능한 형식(687)의 사양에 따라 정의되었는지 확인하기 위해 해당 기능을 검사한다.

[0306] 도 109는 선정된 패턴 매칭 알고리즘(SPMA)(526) 보안 응답으로부터의 변수들을 그러한 알고리즘의 지능으로부터의 '구제'인식들로 리버스 엔지니어링하는 메트릭 처리(MP)(489)를 도시한다. 보안 응답 X(690)는 SPMA(예 : 승인/ 차단/ 난독화)에 의해 선택된 최종 보안 응답에 기여하는 일련의 요소를 나타낸다. 모양 각각은 선택된 패턴 매칭 알고리즘(SPMA)의 보안 응답을 나타낸다. 초기 가중치는 SPMA에 의해 결정되므로 지능이 활용되고 있다. 그런 결정은 대량 인식에서 모델 인식으로 참조된다. 인식 추론(Perception Deduction: PD)(490)은 보안 응답의 일부와 해당 시스템 메타 데이터를 사용하여 보안 응답의 원래 인식을 복제한다. 입체화 시리즈들(699)의 인식의 해석은 PD가 SPMA의 보안 응답을 취하는 방법을 표시하고 SPMA가 원래 사용한 지능형 '디지털 인식'의 전체 범위를 재구성하기 위해 관련 입력 시스템 메타 데이터(484)를 연관시킨다. 이것은 CTMP에게 입력 알고리즘에 대한 깊은 이해를 제공하며, 여러 알고리즘의 지능을 재사용하고 상호 참조 할 수 있으므로 인공 지능의 획기적인 중요한 단계가 구현된다. 이러한 모양은 SPMA에 의해 구현된 복잡한 규칙, 동작 및 상관 관계를 상징한다. 모양 채우기(697), 다수 쌓기(698) 및 입체화(699)는 지능형 알고리즘의 '관점'을 포착하는 디지털 인식이다. 입체화(699) 유형의 인식은 3 차원 모양을 나타내며 회사 직원의 내부 전자 메일을 해석하고 회사 민감한 정보의 보안 위반을 탐지 및/또는 예측하려는 언어 학습 알고리즘을 상징적으로 나타낼 수 있다. 입체화 유형은 약간의 변화가 있는 단일 지능형 알고리즘일 수 있지만(즉, 변형694C는 원형이고 695C/ 696C는 직사각형이며, 지능형 알고리즘의 미묘한 차이를 나타냄) 액면가(face value)에서 그러한 알고리즘에 의해 만들어진 것처럼 보

이지 않을 수 있는 여러 초기 보안 응답이 있을 수 있다. 액면가 694A는 696A보다 692A와 공통점이 더 많다. 그러나 이 카운터 직관에도 불구하고, 692A는 입체화(699)와 완전히 다른 알고리즘 모양 채우기(697)에 의해 수행된 보안 응답이다. 인식 695C와 696C는 동일하지만 보안 응답 695A와 696A는 미묘한 차이가 있다. 보안 응답 695A는 더 어둡고 측면 695B의 차원 인식을 나타내지만 696A는 정면 696B에서와 똑같은 인식을 나타낸다. 이러한 차이점은 다양한 보안 위협/ 의심에 대응하는 다양한 보안 대응을 리버스 엔지니어링하고 동일한 지능형 알고리즘으로 파악할 수 있음을 보여준다. 입체화(699) 인식의 3 가지 경우(두 개는 동일함)는 이후 인식의 관점 B(702)로서 CTMP에서 내부적으로 참조되는 단일 단위로 결합된다. 이 관점의 인식이 CTMP 내에서 미치는 영향의 가중치는 보안 응답 694A, 695A 및 696A에 대한 영향의 초기 가중치에 따라 계산된다. 다수 쌓음에서 인식(698)을 사용하면 입체화(699)에 따라 3 차원 깊이를 받는 대신 보안 응답(693A)이 여러 수량 집합의 일부인 것으로 나타난다. 이것은 외부 침입을 피하기 위해 신생 회사 직원에 대한 보안 프로파일을 작성하는 프로파일링 알고리즘에 대한 상징적인 표현 일 수 있다. CTMP는 처음에는 보안 응답(693A)으로 표시되는 단일 보안 프로필만 수신하지만 사실은 (MP(489)가 리버스 엔지니어링을 수행 한 후) 인식 인식 다수 쌓음(698)으로 알려진 상호 참조 프로필 모음의 일부이다. 이러한 이러한 인식은 CTMP 내에서 인식의 관점 A(701)로 참조할 수 있다. 보안 응답 691A 및 692A의 경우 보안 응답은 상징적으로 불완전한 모양으로 표시되는 MP(489)에 제공된다. PD(490)는 입력 시스템 메타 데이터를 활용하여 이 보안 응답이 발생한 지능형 알고리즘이 예상 보안 변수가 없음을 찾고 있는지 확인한다. 예를 들어, 이것은 의심스러운 행동의 존재를 알리는 것과는 대조적으로 규칙적인/ 예상 된 행동의 부재를 알리는 알고리즘 일 수 있다. 이것은 그가 평소와 같이 이메일을 보내지 않는 회사 직원 일 수 있다. 이것은 갑작스런 습관의 변화 또는 실제 직원과 같은 이메일에 익숙하지 않은 악의적 인 배우가 이 직원의 이메일 계정을 해킹했다는 표시 일 수 있다. 이러한 알고리즘은 영향의 적절한 가중치를 갖는 인식의 관점 C(700)로서 CTMP 내에서 참조 될 수 있는 디지털 인식 모양 채우기(697)로 리버스 엔지니어링된다.

[0307] 도 110 및 도 111은 메트릭 처리(MP)(489)에 의해 주로 사용되는 인식 추론(Perception Deduction: PD)(490)의 내부 설계를 도시한다. 보안 응답 X는 정당화/ 추론 계산(704)에 입력으로 전달된다. 이 모듈은 의도 DB(705)에 저장된 입력/출력 축소(Input/ Output Reduction) 모듈(706)의 의도 공급을 이용함으로써 SPMA(526)의 보안 응답의 정당성을 결정한다. 이러한 모듈 IOR은 함수의 입력/ 출력 관계를 해석하여 함수의 목적에 대한 정당성 및 의도를 결정한다. IOR 모듈은 메타 데이터에 나열된 다양한 함수 호출의 분리된 입력 및 출력을 사용한다. 이러한 메타 데이터 분리는 메타 데이터 카테고리화 모듈 (MCM)(488)에 의해 수행되고, 출력 카테고리는 콜렉션(672 및 674)으로서 발생한다. JRC(704)에서, 의도 DB(705)에 저장된 기능 의도는 입력(690)으로서 제공된 보안 응답에 대해 체크된다. 기능 의도가 SPMA의 보안 결정을 뒷받침하는 경우 메트릭 변화에 대한 정당화(JMC; Justification to Metric Conversion)(703)에 대한 타당한 근거로 제출된다. JMC 모듈에서 검증된 보안 응답 정당성은 인식의 특성을 정의하는 메트릭으로 변환된다. 메트릭은 인간의 감각과 유사하며 보안 응답 정당성은 이러한 의미를 사용하는 정당성을 나타낸다. 사람이 길을 건너면 시력과 소리에 대한 그들의 감각(또는 측정 기준)이 높아지고 냄새와 촉감에 대한 감각은 휴면 한다. 각각의 강도 크기를 가진이 감각 모음은 '길 건너기'인식을 나타낸다. 이 비유에 대한 정당화는 '도로의 차량은 위험할 수 있으며 볼 수 있고 들을 수 있음' 이다. 따라서 인식 구성은 합리적으로 정당화되며, 예로서 인식의 관점 C(543)가 형성된다. I/ O (입력/ 출력) 관계는 기능 입력의 단일 세트와 이러한 기능에 의해 제공된 해당 출력으로 정의된다. IOR(706)은 먼저 함수의 I/ O 관계 및 함수 '의도'가 내부 데이터베이스를 참조하여 이전에 분석되었는지를 검사한다. 정보가 데이터베이스에서 발견되면, 그것은 708 단계에서 현재의 I/ O 데이터를 보완하기 위해 사용된다. 단계 714에서 충분한 수준의 의미 있는 분석을 달성할 수 있을 만큼 충분히 포화 된 경우 보충된(해당되는 경우) I / O 데이터가 검사된다. 양(amount)은 기술 측면에서 정량화되며 최소 수준은 기존 CTMP 정책에 의해 정의된다. 분석할 I/ O 정보의 양이 불충분하다면, 단계 711에서 특정 기능 분석이 취소되고 IOR 모듈 (706)은 다음 이용 가능한 기능으로 진행한다. 분석할 충분한 양의 정보가 있을 때, I/ O 관계는 유사성(709)에 따라 분류된다. 예를 들어 하나의 I/ O 관계가 발견되면 하나의 통화를 다른 통화 (예: USD에서 EUR)로 변환하고 또 다른 I/ O 관계가 발견되어 하나의 단위가 다른 단위(즉, 파운드에서 킬로그램)로 변환된다. 두 I/ O 관계는 트리거 개념이 카테고리화 인덱스와 연관되어 있기 때문에 데이터 변환에 속하는 것으로 분류된다. 예를 들어, 그러한 인덱스는 USD, EUR 및 파운드를 참조할 수 있다. 킬로그램은 데이터 변환 카테고리를 참조한다. 따라서 일단 I/ O 관계에 있는 유닛이 발견되면 IOR(706)은 해당 유닛을 적절히 분류할 수 있다. 따라서 함수의 의도는 통화 및 단위 변환 함수로 의심된다. 이용 가능한 모든 I/ O 관계를 카테고리화할 때, 카테고리는 단계 710에서 포함하는 I/ O 관계 가중치의 양에 따라 등급이 매겨지며 가장 인기있는 것이 먼저 나타난다. 단계 715에서, I/ O 데이터의 카테고리는 기능의 의도의 패턴을 자신 있게 표시할 수 있는지 검사된다. 이는 함수가 수행하는 입력- 출력 변환의 일관성을 검사하여 수행된다. 특정 범주의 정보가 지속적이고 뚜렷한 경우 (통화를 하나의 범주로 변환하고 단위를 두 번

째 범주로 변환하는 경우)이 범주는 기능의 '의도'로 표시된다. 따라서 이 기능은 통화와 단위를 변환 할 의사가 있다고 설명 할 것이다. IOR(706)은 기능을 의도된 용도로 축소함으로써 CTMP가 코드에 존재하는 기능의 실제 목적을 확인할 수 있고 그러한 코드의 실행을 통해 손상이 발생하기 전에 선제 적으로 악성 행위를 지능적으로 검사 할 수 있으므로 보안 분석에 중요한 영향을 미친다. '의도'가 IOR(706)에 의해 충분한 정도의 확신으로 잘 이해 되었다면 모듈 출력(712)으로서 제출된다. '의도' 카테고리가 서로 확고하게 확증되지 않고 함수의 '의도'가 자신 있게 설정되지 않은 경우 함수의 '의도'는 알 수 없음으로 선언되고 IOR(706)은 711 단계에서 분석을 위해 다음 사용 가능한 함수로 진행한다.

[0308] 도 112 내지 도 115는 인식 관찰자 에뮬레이터(POE; Perception Observer Emulator)(475)를 표시한다. 이 모듈은 관찰자의 에뮬레이션을 생성하고 관찰자 에뮬레이션의 그러한 변화와 함께 모든 잠재적인 인식 포인트를 테스트/ 비교한다. 입력은 모든 잠재적 인식 포인트와 향상된 데이터 로그들이다. 그 결과는 선택된 인식의 혼합과 함께, 가장 관련성이 높고 가장 중요하며 가장 조심스러운 관찰자에 따라 그러한 강화된 로그의 결과 보안 결정이다. 입력 시스템 메타 데이터(Input System Metadata)(484)는 가변 인식 포맷(CVF)(547)에서 인식을 생성하기 위해 원시 인식 생성(RP2)(465)에 의해 사용되는 초기 입력이다. 저장 검색(SS)(480)을 사용하면 데이터 향상 로그에서 파생된 CVF가 PS(Perception Storage)(478)의 데이터베이스 조회에서 기준으로 사용된다. PS는 가장 일치하는 CVF가있는 데이터베이스에서 사용 가능한 모든 CVF(547)를 제공한다. 관련 인식 구성과 가중치가 참조되어 결과(716)에서 성공적인 일치 이벤트에 사용된다. 유사성 오버랩은 60 % 일치(719) 및 30 % 일치(720)로 언급된다. 이러한 결과는 저장 검색(480)에서 계산된다. 결과(716)로, 일치(719 및 720)가 저장되고, 가중치 계산(718)에서 개별 인식 순위에 대해 계산된다. 이러한 계산은 입력 CVF와 비교하여 데이터베이스 CVF의 전반적인 유사성(또는 일치) 값을 취하여 각 개별 인식 가중치에 해당 값을 곱한다. 이러한 가중치는 이미 메트릭 처리(MP)(489)에 의해 처음 결정된 CVF와 연관되어 저장된다. 순위(717)에서, 인식은 최종 가중치에 따라 정렬된다. 이러한 순위는 보안 상황을 이해하고 따라서 최종 차단(730) 또는 승인(731) 명령 출력을 전달하는데 가장 관련성이 높은(가중치 계산(718)에서 가중치 부여 된) 인식을 사용하는 선택 프로세스의 일부이다. 일단 인식이 순위가 매겨지면, 데이터 향상된 로그들(723)이 인식에 적용되어 차단/ 승인 권고를 생성하는 애플리케이션(729)으로 포워딩된다. 로그들(723)은 원래 보안 사고가 발생한 시스템의 입력 로그들이다. 자기 비판적 지식 밀도(SCKD; Self-Critical Knowledge Density)(492)는 알 수 없는 지식의 예상 상위 범위를 정의하기 위해 로그들에 태그를 붙인다. 즉, 인식을 통해 알 수 없는 데이터 범위로 태그가 지정된 데이터를 고려할 수 있다. 이는 보안 사고가 얼마나 많은지 알지 못하는 정도뿐만 아니라 알 수 있는 정도를 감안할 때 인식이 보안 사고를 보다 정확하게 평가할 수 있음을 의미한다. 데이터 파싱(724)은 원래의 선택된 패턴 매칭 알고리즘(SPMA)(526)에 의해 결정된 바와 같이 최초 승인 또는 차단 결정(725)을 출력하기 위해 데이터 향상된 로그들(723) 및 입력 시스템 메타 데이터(484)의 기본 해석을 수행한다. 따라서, 2 개의 잠재적인 경우 시나리오가 존재하며, SPMA는 시나리오(727)에서 보안 관련 사건을 차단(즉, 프로그램 다운로드를 차단)(730)하거나 시나리오(726)에서 그러한 사고를 승인(731)하도록 선택했다. 이 시점에서 CTMP(22)는 결정을 비판하는(cybersecurity를 포함하되 이에 국한되지 않음) 가장 핵심적이고 중요한 과제를 수행 할 준비가 되었다는 점에서 지금까지 진보 해왔다. 이 비평은 CTMP 내에서 인식에 따라 인식 관찰자 에뮬레이터(POE)에서 한 번, 논리적으로 정의 된 규칙에 따라 규칙 실행(RE)에서 두 번씩 두 가지 방식으로 두 번 발생한다. POE 내에서 SPMA로부터 차단 명령을 수신하면(732)의 대체 로직이 사용된다. SPMA로부터 승인 명령을 수신하면, 733의 중단 로직이 결합된다. 단계 732A에서, 블록(730)의 디폴트 동작이 가정되고 BLOCK-AVG 및 APPROVE-AVG 값(732B)은 사례 시나리오(727)에 저장된 차단/ 승인 신뢰 값의 평균을 구함으로써 계산된다. 단계 732C는 사례 시나리오 (727)의 평균 신뢰도가 사전 정의된 (신뢰도에 의해) 신뢰 마진보다 큰지를 검사한다. 시나리오의 신뢰도가 낮으면 CTMP가 정보/ 이해 부족으로 인해 비판을 보류하고 있음을 나타낸다. 이러한 낮은 신뢰 상황이 발생하면 RMA 피드백 모듈 (728)은 단계 732D에서 참여하여 더 많은 인식이 포함된 보안 상황을 재평가하려고 시도한다. 이와 같이 추가적으로 고려된 인식은 신뢰 마진을 증가시킬 수 있다. 따라서 RMA 피드백은 리소스 관리 정책에 따라 재평가가 허용되는지를 확인하기 위해 RMA(Resource Management and Allocation)(479)와 통신한다. 그러한 재평가가 거부되면 알고리즘은 최고 신뢰도에 도달하고 초기 승인/ 차단 결정을 무효화하여 이 POE 세션에 대해 영구적으로 중단된다. 단계 732E는 RMA 피드백 모듈(728)이 RMA(479)로부터 허가를 수신하여 더 많은 자원을 재 할당하고 따라서 더 많은 인식을 계산에 재 할당하는 조건을 나타낸다. 그러한 조건에서 추가 시나리오(CTMP 비평)는 단계 732F에서 중단되어 사례 시나리오(727)의 새로운 평가가 부가적인 인식(및 그에 따른 컴퓨터 자원 부하 증가)으로 발생하도록 허용한다. 732G 단계에 따르면 승인 평균은 (정책에 따라) 기본 차단 동작(730/732A)를 732H단계의 승인 동작(731)으로 재정의할 수 있을 만큼 충분히 신뢰할 수 있음을 나타낸다. 사례 시나리오(726)에서 발생하는 승인 논리(733)에도 동일한 논리가 적용된다. 단계 733A에서 디폴트 동작은 SPMA(526)에 의해 요청된 대로 승인으로 설정

된다. BLOCK-AVG 및 APPROVE-AVG 값 (733B)은 사례 시나리오(726)에 저장된 차단/ 승인 신뢰 값의 평균을 찾아 계산된다. 단계 733C는 사례 시나리오(726)의 평균 신뢰도가 사전 정의된 (신뢰도에 의해) 신뢰 마진보다 큰지를 검사한다. 이러한 낮은 신뢰 상황이 발생하면, RMA 피드백 모듈(728)은 단계 733D에서 참여하여 더 많은 인식이 포함 된 보안 상황을 재평가하려고 시도한다. 단계 733E는 RMA 피드백 모듈(728)이 RMA(479)로부터 허가를 수신하여 더 많은 자원을 재 할당하고 따라서 더 많은 인식을 계산에 재 할당하는 조건을 나타낸다. 이러한 조건에서 추가 시나리오(CTMP 비평)는 단계 733F에서 중단되어 사례 시나리오(726)의 새로운 평가가 추가 인식(및 그에 따른 컴퓨터 자원 부하 증가)으로 발생하도록 허용한다. 733G 단계에서는 정책에 따라 승인 평균이 기본 승인 동작(731/733A)을 733H 단계의 차단 동작(730)으로 재정의하기에 충분한 신뢰도를 나타낸다.

[0309] 도 116 및 도 117은 인식의 현재 알려진 관점으로부터 연관될 수 있는 인식 데이터의 관점을 유도하는 함축적 파생(ID)(477)를 보여준다. 적용된 인식의 관점(Applied Angles of Perception)(470)은 CTMP 스토리지 시스템에 저장된 알려진 인식의 범위이다. 이러한 인식들(470)은 SPMA(526)에 의해 적용되고 사용되었고, 인식 모음(734)으로서 수집되어 메트릭 조합(493)으로 전송된다. 이 모듈(493)은 인식의 관점(734) 형식을 함축적 파생(ID)(477)에 의해 인식된 형식인 메트릭의 카테고리로 변환한다. 메트릭 복잡성(736)에서 원의 외부 경계는 개별 메트릭에 관한 알려진 지식의 최고점을 나타낸다. 따라서 원의 바깥 쪽 가장자리는 더 많은 메트릭 복잡성을 나타내지만 센터는 덜 복잡한 메트릭을 나타낸다. 중앙 연한 회색은 인식의 적용 관점의 현재 배치의 메트릭 조합을 나타내며, 바깥 쪽 진회색은 일반적으로 시스템에 저장되고 알려진 메트릭 복잡성을 나타낸다. ID(477)의 목표는 관련 측정 항목의 복잡성을 증가시켜 인식의 각도를 복잡성과 수량에 곱할 수 있도록 하는 것이다. 현재의 배치로부터의 알려진 메트릭 복잡성은 이미 그러한 상세/ 복잡성을 포함하지 않는 경우 관련된 메트릭 DB(738)에 추가된다. 이러한 방식으로 시스템은 완전히 순환을 이뤘고, 새로 저장된 메트릭 복잡성은 향후 인식 함축적 파생의 관점의 배치에 사용될 수 있다. 이러한 복합 메트릭 구성(736)은 다수의 다양한 인식 관점의 메트릭이 개개의 데이터베이스(738)에 카테고리에 따라 저장되는 메트릭 확장(ME)(495)에 입력으로서 전달된다. 짙은 회색의 표면적은 인식의 적용된 관점의 현재 배치의 전체 범위와 알려진 상한에 따라 남은 범위의 양을 나타낸다. 상한은 각 개별 메트릭 DB의 최고 지식으로 표시된다. 따라서 현재 측정 기준 배치(관점 인식의 현재 배치에 의해 유도된)는 이전에 알려진 세부 사항/ 복잡성으로 향상된다. 향상 및 복잡성 강화시 메트릭은 메트릭 복잡성(737)으로 반환된다. 다이어그램(737)에서 보여는 것처럼, 밝은 회색 영역은 메트릭 범위(739), 일관성(740), 타입(741) 및 강도(742)의 4 개의 섹터 모두에서 커지게 된다. 이는 네 가지 메트릭 분야 모두에서 인식이 더욱 상세하고 복잡해 졌음을 나타낸다. 이 향상된 메트릭 복잡성(737)은 메트릭 전환(494)의 입력으로 전달되며, 메트릭 전환(494)은 개별 및 전체 인식의 관점(735)을 뒤집는다. 따라서 최종 출력은 원래의 입력 인식의 적용 관점(470)의 확장된 버전인 인식의 함축된 관점(471)으로 어셈블된다.

[0310] 도 118 내지 도 120은 자기 비판적 지식 밀도(SCKD(492)를 보여 주며 보고 가능한 로그의 범위를 벗어나는 잠재적인 알려지지 않은 지식의 범위와 타입을 추정한다. 이러한 방식으로 CTMP(22)의 비판적 사고 기능은 시스템에 의해 직접 알려지고 알려지지 않은 모든 관련 지식의 잠재적 범위를 활용할 수 있다. 다음은 SCKD(492)의 의도된 기능 및 성능을 보여주기 위한 사용 케이스의 예이다.

[0311] 1)이 시스템은 핵 물리학에 대한 강력한 기준 범위를 구축했다.

[0312] 2)이 시스템은 핵 물리학과 양자 물리학이 복잡성과 타입 측면에서 카테고리화되고 체계적으로 유사하다는 비유를 수행했다.

[0313] 3) 그러나 이 시스템은 핵 물리학보다 양자 물리학에 대한 참조 가능한 지식이 훨씬 적다.

[0314] 4) 따라서 시스템은 핵 물리학의 유추를 통해 잠재적으로 달성 할 수 있는 양자 물리학 지식의 상한을 정의한다.

[0315] 5)이 시스템은 양자 물리학 측면에서 알려지지 않은 지식의 범위가 크다는 것을 결정한다.

[0316] 알려진 데이터 분류(KDC)(743)는 확인된(알려진) 정보를 입력(746)과 단호하게 분리함으로써 적절한 DB 유추 질의가 수행 될 수 있도록 한다. 이러한 정보는 카테고리 A, B 및 C(750)으로 분리 된 후 개별 카테고리가 개별적으로 비교 가능한 가변적 형식 생성기(CVFG)(491)에 입력을 제공한다. CVFG는 알려진 정보 범위 DB(747)의 유사성을 확인하기 위해 저장 검색(SS)(480)에 의해 사용되는 CVF(547) 포맷으로 카테고리 정보를 출력한다. DB(747)에서는 알려진 데이터의 상한이 데이터 카테고리에 따라 정의된다. 유사한 유형 및 데이터 구조를 비교하여 지식 범위의 신뢰를 평가한다. SS(480)이 시나리오(748)에서 지식 유추를 위한 결과를 찾지 못하면 현재의 데이터가 저장되어 향후 비유가 만들어 질 수 있다. 사용 케이스 예제에 따르면, 이것은 핵 물리학의 범위를 정

의 할 수 있는 사건이다. 그러면 양자 물리학이 미래에 참조 될 때, 그것은 핵 물리학 지식 범위의 현재 저장과 함께 그것의 지식 범위를 비유 할 수 있다. 시나리오(749)는 결과 발견 상황을 기술하며, 각 카테고리는 SS(480) 결과에 따라 알려진 데이터의 관련 범위로 태그가 지정된다. 그 후 카테고리 당 알려지지 않은 정보의 태깅된 범위는 알려지지 않은 데이터 조합(UDC; Unknown Data Combiner)(744)에서 원래 데이터의 동일한 스트림(입력(746))으로 재 조립된다. 출력(745)에서 원래의 입력 데이터가 리턴되고 알려지지 않은 데이터 범위 정의와 결합된다. 도 119에서, 알려진 데이터 분류(KDC) 모듈(743)이 보다 상세히 도시된다. 알려진 데이터(752)는 주 입력이며, 에러 로그로부터의 개별 엔트리와 같은 데이터의 정의된 범위를 나타내는 정보 블록(755)을 포함한다. 단계 756은 사용 케이스에 따라 핵 물리학 정보로 표시된다는 것을 블록 내에서 인식 할 수 있는 정의가 있는지 확인한다. 카테고리 홀드 아웃(750)에서 블록의 정보 라벨에 적합한 카테고리가 존재하는 경우 기존 카테고리는 처리된 블록 정보(755)를 보완하여 단계 748의 세부 사항으로 강화된다. 그러한 카테고리가 존재하지 않으면, 정보 블록(755)이 적절하고 정확하게 저장 될 수 있도록 단계 (749)에서 생성된다. 가장 기본적인 로직(759)은 모든 블록이 처리 될 때까지 순차적으로 블록을 순환한다. 이들 모두가 처리 된 후 최소 금액(정책에 의해 정의 됨)이 카테고리 홀드 아웃(750)에 제출된 경우 KDC(743)는 모듈 결과를 빈(Null) 결과(618)로 제출한다. 충분한 양의 처리 된 블록이 있으면 카테고리 홀드 아웃(750)이 중간 알고리즘(751)(주로 SCKD(492) 임)에 제출된다. 미지 데이터 결합기(UDC)(744)는 중간 알고리즘(751)으로부터 태그된 알려지지 않은 데이터 포인트(757)와 함께 알려진 데이터를 수신한다. 이러한 데이터는 처음에는 카테고리 홀드 아웃(750)에 저장되고 가장 기초적인 로직(760) 사이클을 통해 모든 데이터 단위가 순차적으로 저장된다. 단계 754는 홀드 아웃(750)의 정의 된 카테고리가 개별 카테고리를 합동 정보 스트림으로 재구성하는 방법을 기술하는 원래의 메타 데이터를 포함 하는지를 점검한다. 그러한 메타 데이터는 원래 KDC(743)의 입력 알려진 데이터(752)에서 발견되었는데, 그 단계에서 데이터는 아직 카테고리로 분리되지 않았으며 모든 데이터를 보유한 초기 단일 합동 구조가 있었기 때문이다. 단계 754가 대응하는 데이터와 메타 데이터를 재 연관시킨 후, 태그된 블록은 블록 재조합 홀드아웃(Block Recombination Holdout)(753)으로 전송된다. 단계 754에서 데이터와 일치하는 메타 데이터가 발견되지 않으면, 홀드 아웃(753)은 필연적으로 비어있게 되고 모듈화 된 빈(null) 결과(618)가 반환된다. 성공적인 메타 데이터 일치 시 홀드 아웃(753)이 채워지고 UDC(744)의 모듈 출력은 알려진 데이터 + 태그된 알려지지 않은 데이터(757)이다. 모듈 출력에서 블록(755)는 KDC(743)의 알려진 데이터(752)에 있는 원래 정보 블록을 나타낸다. 오각형(758)은 알려진 데이터(755)의 모든 블록과 결합되는 알 수 없는 데이터 범위 정의를 나타낸다.

[0317] **어휘 객관성 마이닝(LOM; Lexical Objectivity Mining)**

[0318] 도 121은 어휘 객관성 마이닝(LOM)의 주된 논리를 나타낸다. LOM은 광범위한 질문 및/또는 주장에 대한 객관적인 대답에 최대한 가깝게 접근하려고 시도한다. 그것은 인간 주체(800)와 연계하여 LOM의 입장에 대해 자신의 주장을 인정하거나 개선할 수 있도록 한다. 논쟁을 유도하거나 개선하는 것은 LOM의 핵심 철학이다. 그것은 그것이 틀렸을 때 그것이 처음부터 지식을 얻는 인간의 지식으로부터 배울 수 있도록 인정해야 하기 때문이다. LOM은 매우 데이터베이스가 무겁우며(CPU, RAM 및 디스크가 모두 중요한 플레이어), 중앙 지식 보존(CKR)(806)이 단일(중복성 및 백업을 위해 복제 됨) 마스터 사례로 중앙 집중화되어 혜택을 볼 수 있다. 제3(타사의) 앱은 유료 또는 무료 API를 통해 이러한 중앙 마스터 사례에 연결될 수 있다. LOM의 활동은 주체 LOM 시각 인터페이스에 질문이나 주장(801)을 제시하는 인간 주체(800)에서 시작된다. 이러한 질문 / 주장(801A)은 중앙 지식 보유(CKR)(806)를 활용하여 질문/ 주장에 대한 이해 및 응답/ 응답에 결정적인 누락된 세부 사항을 해독하는 초기 쿼리 추론(IQR)(802)으로 처리를 위해 전송된다. [...]

[0319] 그 후, 보충 질문 데이터와 함께 질문/ 주장(801)은 인간 주체(800)와 연관되는 조사 해설(SC)(803A)으로 전송되어 질문/ 주장(801A)이 객관적으로 그리고 모든 필요한 맥락으로 분석 될 수 있도록 보충 정보를 얻는다. 따라서 명확한 질문/ 주장(801B)가 형성되며, 원래의 질문/ 주장(801)을 인간 주체(800)에 의해 제기된 것처럼 보였지만 SC(803A)를 통해(800)에서 배운 세부 정보를 보충했다. 주장 구성(AC)(808A)는 주장이나 질문(801B와 같은)의 형태로 제안을 받고 그러한 제안과 관련된 개념의 산출물을 제공한다. 반응 제시(Response Presentation)(809)는 LOM (특히 AC(808))이 작성한 결론을 인간 주체(800)과 합리적 호소(RA)(811) 모두에 표시하기 위한 인터페이스이다.

[0320] 이러한 인터페이스는 인간(800)이 RA(811)을 이해하고 순수한 디지털 구문 형식으로 시각적으로 표현된다. 계층적 맵핑(HM; Hierarchical Mapping)(807A)는 관련 개념을 매핑하여 질문/ 주장 일관성에서 확인 또는 충돌을 찾는다. 그런 다음 주제에 대해 특정 태도를 취할 때의 이점과 위험을 계산한다. 중앙 지식 보유(806)는 LOM에 대한 지식을 참조하기 위한 기본 데이터베이스이다. 쿼리 효율성과 논리적 카테고리화 및 개념 분리를 위해 최적화되어 강력한 논증이 만들어지고 인간 주체(800) 비판에 대응하여 패배한다. 지식 검증(Knowledge Validation,

KV)(805A)은 질의 능력 및 CKR(806)으로의 동화(assimilation)를 위해 논리적으로 분리될 필요가 있는 신뢰도가 높고 미리 비판된 지식을 받는다. 응답 수락(Accept Response)(810)은 LOM의 응답을 수락하거나 비평으로 호소하기 위해 인간 주체(Human Subject)800에게 주어진 선택이다. 응답이 수락되면, KV(805A)에 의해 처리되어 확인된(높은 신뢰도) 지식으로서 CKR(806)에 저장 될 수 있다. 인간 주체(800)가 응답을 수락하지 않으면 인간(800)이 제공한 항소(appeal) 이유를 확인하고 비판하는 합리적 호소(RA)(811A)로 전달된다. RA(811A)는 그것이 자기 비판이든 또는 인간 반응에 대한 비판이든(응답 수락(810)에서 'NO' 응답으로 부터) 주장을 비판 할 수 있다.

[0321] 도 122 내지 도 124는 관리형 인공 지능의 서비스 공급자(MAISP)(804A)를 보여준다. MAISP는 중앙 지식 보유(CKR)(806)의 마스터 사례로 LOM의 인터넷 클라우드 사례를 실행한다. MAISP(804A)는 LOM을 프론트 엔드 서비스(861A), 백엔드 서비스(861B), 제 3 (타사) 애플리케이션(응용 프로그램) 의존성(804C), 정보 소스(804B) 및 MNSP(9) 클라우드에 연결한다. 프론트 엔드 서비스(861A)에는 인위적으로 지능형 개인 보조 장치(예: 애플의 Siri, Microsoft의 Cortana, Amazon의 Alexa, Google 보조), 통신 응용 프로그램 및 프로토콜 (예: Skype, WhatsApp), 홈 오토메이션 (냉장고, 차고, 문, 온도 조절기) 및 의료용 어플리케이션(즉, 의사의 2 차 의견, 의료 기록)를 포함한다. 백엔드 서비스(861B)는 온라인 쇼핑(즉, Amazon.com), 온라인 운송(즉, 우버 (Uber)), 처방전 주문 (즉, CVS) 등을 포함한다. 이러한 프론트 엔드(861A) 및 백 엔드(861B) 서비스는 정보 전송 및 프로토콜의 표준화를 가능하게 하는 문서화된 API 기반 구조(804F)를 통해 LOM과 상호 작용한다. LOM은 자동화된 연구 메커니즘(ARM)(805B)을 통해 외부 정보 소스(804B)에서 지식을 검색한다.

[0322] 도 125 내지 도 128은 LOM의 종속성(의존성) 구조를 보여 주며 모듈이 서로 상호 의존하는 방법을 도시한다. 언어학적 구성(LC)(812A)는 인간 주체(800) 및 병렬 모듈의 원시 질문 / 주장 입력을 해석하여 LOM 시스템 전체에서 이해할 수 있는 언어 구문의 논리적 분리를 생성한다. 개념 발견(CD)(813A)은 명확화 된 질문/ 주장(804) 내의 관심 포인트를 수신하고 CKR(806)을 레버리지함으로써 관련된 개념을 도출한다.

[0323] 개념 우선 순위(CP)(814A)는 관련 개념을 수신하고 특수성 및 일반성을 나타내는 논리적 계층으로 지시한다. 최상위 계층에는 가장 일반적인 개념이 할당되고 하위 계층에는 점점 더 구체적인 개념이 할당된다. 응답 분리 로직(RSL)(815A)은 LC(812A)를 활용하여 사람의 반응을 이해하고 적절하고 유효한 응답을 초기 명확화 요청과 연관시켜 SC(803A)의 목적을 달성한다. LC(812A)는 출력 단계에서 다시 활용되어 원래의 질문/ 주장(801)을 수정하여 SC(803)에서받은 보충 정보를 포함시킨다. 인간 인터페이스 모듈(HIM; Human Interface Module)(816A)는 초기 질문 추론(IQR)(802A)에서 지정한 지식의 틈을 해결하기 위해 인간 주체(800)에 명확하고 논리적으로 분리된 프롬프트를 제공한다. 문맥 구성(CC)(817A)은 주장 구성(AC; Assertion Construction)(808A)의 메타 데이터와 인간 주체(800)의 잠재적인 증거를 사용하여 비판적 사고를 위한 CTMP에 원시 사실을 제공한다. 결정 비교(DC)(818A)는 사전 비판 결정과 사후 비판 결정 사이의 중첩을 결정한다. 개념 호환성 탐지(CCD)(819A)는 원래의 질문/ 주장(801)으로부터의 개념적 파생물을 비교하여 논리적 호환성 결과를 확인한다. 그러한 개념은 상황, 존재 상태, 부채 등을 나타낼 수 있다. 이익/ 위험 계산기(BRC)(820A)는 CCD(819A)의 호환성 결과를 수신하고 개념 구성에 포함된 변수의 그라디언트를 포함하는 균일 한 결정을 형성하기 위해 이점과 위험을 고려한다. 개념 상호 작용(CI)(821A)은 AC(808A) 개념에 관련된 속성을 조사 설명(SC)(803A)를 통해 인간 주체(800)에서 수집한 정보의 일부에 할당한다.

[0324] 도 129 및 130은 초기 질문 추론(IQR)(802A)의 내부 논리를 도시한다. IQR(802)의 서브 세트로 동작하는 언어적 구성(LC)(812A) 인간 주체(800)에서 원래의 질문 / 주장 (801)을 수신한다. 801은 언어적으로 분리되어 있으므로 IQR(802A)는 한 번에 각 단어 / 구를 처리한다. 조동사 '해야한다(should)'(822)는 시간 차원(822)에 관한 명확성이 부족하다. 따라서 '매일?', '매주?'등과 같이 명확한 질문에 대한 답이 형성된다. 주체 '나(I)'(823)은 누가 주체인지에 대한 명확성이 부족하다는 것을 나타내므로 후속 질문이 인간 주체(800)에 표시되도록 구성된다. 동사 '먹는다(eat)'(824)는 명확하지 않은 것은 아니지만 명확성이 부족한 다른 분석 지점을 보완 할 수 있다. IQR(802)는 CKR(806) DB를 활용하여 824 단계에서 건강과 돈의 개념과 음식 개념을 연결한다. 이것은 '남성 또는 여성?', '당뇨병?', '운동?', '구매력?'과 같은 보다 적절한 관련 후속 질문이 요청되도록 '주체 질문'(823)이라는 쿼리에 알린다. 명사 '패스트 푸드(fast-food)'(825)는 단어가 어떻게 해석되어야 하는지에 대한 명확성이 부족하다는 것을 암시한다. 그것은 기술적인 의미(827)에서 가장 빠른 형태로 제공되는 음식 중 가장 원시적인 형태로 해석 될 수도 있고, '주문한 장소에서 값 싸고 매우 빠르게 만들어진 튀김된 짭짤한 음식(826)' 가지를 구어체로 이해할 수도 있다. 샐러드 바는 기술적으로 미리 준비되어 있고 즉시 사용할 수 있는 음식을 얻기 위한 빠른 방법이다. 그러나 이 기술적 인 정의는 '패스트 푸드 (fast-food)'에 대해보다 일반적으로 이해되는 구어체의 이해를 따른다. CKR(806)을 참조함으로써 IQR(802)는 '패스트 푸드 (fast-food)'라는 용

어의 모호성을 고려하여 가능한 가능한 옵션을 고려한다. '버거 가게?'와 '샐러드 바?'와 같은 모호한 옵션은 인간 인터페이스 모듈(HIM)(816)을 통해 인간 주체(800)로 전달 될 수 있다. 그러나 질문(801)의 일반 맥락이 구어 의미(826)에 대한 참조임을 나타내는 충분한 정보가 CKR(806)에 있을 수 있다. CKR(806)은 패스트 푸드와 건강과 관련된 논쟁의 수준이 있다는 것을 점차적으로 배우고 난 후에 그런 일반 맥락을 나타낼 수 있다. 그러므로 질문(801)이 그 논쟁을 언급할 가능성이 높다. 따라서 HIM(816)은 인간 주체(800)를 더 명확히 하기 위해 호출 할 필요가 없다. 따라서 IQR(802)은 정의의 의미에서 명백하고 미묘한 뉘앙스를 해독하고자 한다. 질문(828)은 LOM 전체에 대해 인간 주체(800)가 진술을 주장하는 것이 아니라 질문을 하고 있음을 나타낸다.

[0325] 도 131은 IQR(802)로부터 입력을 수신하는 조사 설명(SC)(803)을 도시한다. 이러한 입력은 도달 할 원래 질문 / 주장(801)에 대한 객관적인 대답을 위해 인간 주체(800)가 응답해야 하는 일련의 요청된 명확화(830)를 포함한다. 따라서 요청된 명확화(830)는 휴먼 인터페이스 모듈(HIM)(816B)에 전송된다. 그러한 명확화에 대한 응답은 응답 분리 로직(RSL; Response Separation Logic)(815A)으로 전달되며 그 후 응답을 명확화 요청과 연관시킨다. 처리중인 요청된 명확화(830)와 병행하여 명확화 언어적 연관성(829)이 언어적 구성(LC)(812A)에 제공된다. 이러한 연관성(829)은 요청된 명확화(830)와 언어적 구조 간의 내부 관계를 포함한다. 이것은 RSL(815A)가 원래의 질문 / 주장(801)을 수정하여 LC(812A)가 HIM(816)을 통해 습득한 정보를 통합한 명확한 질문(804)를 출력 할 수 있게 한다.

[0326] 도 132는 조사 설명(SC)(803)에 의해 생성된 명확화된 질문 / 주장(804)을 수신한 주장 구성(AC)(808)을 도시한다. 그런 다음 LC(812A)는 질문을 개념 발견(CD)(813)에 전달되는 관심 포인트(834)(핵심 개념)로 분류한다. CD는 CKR(806)을 이용하여 관련 개념(832)을 도출한다. 개념 우선 순위(CP)(814A)는 개념(832)을 특이성 및 일반성을 나타내는 논리적 계층으로 순서화 할 수 있다. 최상위 계층에는 가장 일반적인 개념이 할당되고 하위 계층에는 점점 더 구체적인 개념이 할당된다. 이러한 정렬은 CKR (806)에 의해 제공된 데이터로 용이하게 수행되었다. 최상위 계층은 모듈러 입력으로서 계층 적 매핑(HM)(807)으로 전달된다. 정보의 병렬 전달에서, HM(807)은 그 의존성 모듈 개념 상호 작용(CI)(821)에 의해 처리되는 관심 지점(834)을 수신한다. CI는 CKR (806)에서 이용 가능한 인덱싱 된 정보에 액세스함으로써 그러한 관심 지점 (834)에 속성을 할당한다. HM(807)이 내부 프로세스를 완료하면 파생된 개념의 호환성 테스트가 완료되고 자세의 이익 / 위험이 측정되고 반환된 후 최종 출력이 AC(808)로 반환된다. AC(808) 및 HM(807)이 완전한 원에 도달했기 때문에 이를 모듈 출력 피드백 루프(833)라고 하고, 분석이 개념의 복잡성을 완전히 포화시킬 때까지 그리고 CKR(806)이 지식의 한계로 인해 병목 현상이 될 때까지 (어느 것이 든 먼저 오는 것) 모듈 출력을 계속해서 전송한다.

[0327] 도 133 및 134는 계층적 매핑(HM)(807)이 어떻게 작동하는지에 대한 내부 세부 사항을 도시한다. AC(808)은 두 가지 유형의 입력을 HM(807)에 병렬로 제공한다. 하나는 개념적 관심 영역(834)로 알려져 있고 다른 하나는 우선 순위가 부여된 개념(837)(가장 일반적인 개념)의 최상위 계층이다. 개념 상호 작용(CI)(821)은 두 입력을 사용하여도 도 128에 도시 된 바와 같이 상황 별 결론을 관심 영역(834)과 연관시킨다. CI(821)는 두 개념 사이의 호환성 / 충돌 수준을 식별하는 개념 호환성 검출(CCD)에 입력을 제공한다. 이는 HM(807)에 인간 주체(800)의 주장 및/ 또는 문제와 중앙 지식 보유(CKR)(806)에서 인덱스된 높은 신뢰 지식 간의 합의 또는 의견 불일치에 대한 전반적인 이해를 부여한다. 이러한 호환성/ 충돌 데이터는 이러한 호환성 및 충돌을 이 문제에 대해 전체적으로 통일된 입장을 취함과 관련된 이익 및 위험으로 변환하는 모듈인 이익/위험 계산기(BRC)(820)로 전달된다. 예를 들어, 사용 사례(인간 주체(800)에서 설정 한 기준에 따라)에 따라 세 가지 주요 입장이 나온다: 패스트 푸드는 전반적으로 권장되지 않거나, 패스트 푸드는 허용되지만 강조되지 않거나, 패스트 푸드가 전반적으로 권장된다. 그러한 자세는 위험/ 이익 요인과 함께 모듈 출력(836)으로 AC(808)에 전달된다. 이것은 AC(808)가 HM(807)이 제시한 주장의 확장을 용이하게 하려고 하므로 LOM 내의 몇 가지 포인트 중 하나이다. 정보 흐름 루프를 포함하는 시스템은 점차적으로 구축된 객관적인 반응인 질문 / 주장의 주관적인 특성으로 점차적으로 보완되는 지능의 그라디언트를 나타낸다. 비유는 꿀벌이 꽃의 과즙을 찾고, 실수로 다른 꽃에 퍼지는 꽃가루를 수집하는 방법이다. 이 꽃의 수정은 장기적으로 아직 더 많은 꿀벌을 끌어들이는 더 많은 꽃을 생산한다. 이는 시스템이 주제의 입장에 대해 강한 확신을 가질 때까지 LOM 내에서 발생하는 상호 연결된 정보 생태계와 유사하여 주장과 성숙한 개념을 점진적으로 '수분(pollinate)'한다. HM(807)의 하위 집합인 개념 상호 작용(CI)의 내부 동작이 도 128에 표시된다. CI(821)은 관심 포인트들(834)을 받고 우선 순위 개념(837)의 최상위 계층에 따라 각각을 해석한다. 이 예에서 상위 계층의 두 가지 우선 순위 개념은 '건강'과 '예산 제약'이다(837). 따라서 CI가 관심 분야(834)를 해석하려고 시도하면 이 주제의 관점을 통하여 이루어질 것 이다. 관심 분야 '당뇨병'(838)은 '예산 제약'(837)에 관한 '비싼 의약'과 '건강'(837)에 관한 '더 취약한 건강'/ '설탕 과민증'에 대한 주장으로 이어진다. 관심 포인트 '남성'(839)은 낮은 신뢰감에도 불구하고 '일반적으로 시간에 쫓긴다'고 주장한다. 체계는 '일 중독'등을 위해 더 많은 특이성이 필요하다는 것을 발견한다. 시간 문제는 시간과 돈 사이

의 상관 관계를 시스템이 인식함에 따라 '예산 제약'에 반비례한다. 관심 영역 ' 중상층'(840)은 '예산 제약'(837)과 관련하여 '양질의 음식을 살 수 있다'고 주장한다. 관심 영역 '버거 킹'(841)은 '예산 제약'(837)에 관해 '싸고' '건강'(837)에 관한 '높은 설탕 함량'과 '튀김 음식'이다. 이러한 주장은 CKR(806)에 저장된 확고하고 신뢰할 수 있는 지식을 참조하여 이루어진다.

[0328] 도 135 및 136은 합리적 호소(RA)(811)의 내부 세부 사항을 보여 주며, RA는 자신의 비판 또는 인간의 반응에 대한 비판이라는 주장을 비판했다. LC(812A)는 RA(811)의 핵심 하위 구성 요소로 작동하며 두 가지 소스에서 입력을 수신한다. 하나의 소스는 인간 주체(800)가 842 단계에서 LOM이 주장한 의견을 거부하는 경우이다. 다른 하나는 LOM 내부의 자체 비판을 위해 AC(808)에 의해 구축된 주장을 디지털 방식으로 전송하는 반응 제시(843)이다. LC(812A)가 언어 텍스트를 나머지 시스템에서 이해할 수 있는 구문으로 변환 한 후 RA의 핵심 논리(844)에서 처리한다. 그러한 핵심 논리가 신뢰도 높은 결과(846)를 반환하면, 그 결과는 CKR(806)로의 적절한 동화를 위해 지식 검증(KV)(805)으로 전달된다. 핵심 논리가 신뢰도 낮은(845) 결과를 반환하면 결과가 AC(808)로 전달되어 자기 비판 주기(전체 원에 도달한 LOM의 다른 요소)를 계속 수행한다. 핵심 논리(844)는 언어 요소가 없는 사전 비판적 결정(847)의 형태로(인공 지능 사용에 최적인 구문을 대신 사용하여) LC(812A)로부터 입력을 수신했다. 그러한 결정(847)은 CTMP(22)의 '주관적 의견'(848) 부문으로 직접 전달된다. 결정(847)은 AC(808)로부터의 메타 데이터 및 인간 주체(800)로부터의 잠재적 증거를 사용하여 원시 사실(즉, 시스템 로그)을 CTMP(22)에 '객관적 사실'입력으로 제공하는 문맥 구성(CC)(817)으로 전달된다. CTMP(22)가 두 개의 필수 입력을 받은 상태에서 이러한 정보는 '객관적 의견'(850)에 도달하는 최선의 시도를 출력하기 위해 처리된다. 이러한 의견(850)은 RA(811) 내에서 비평 후 결정(851)으로 내부적으로 취급된다. 사전 비판적(847) 및 사후 비판적(851) 결정은 결정들(847과 851) 사이의 중첩 범위를 결정하는 의사 결정 비교(DC)(818)로 전달된다. 항소 논쟁이 진정(852)한 것으로 인정되거나 항소가 무효 한 이유를 설명하기 위해 항소가 개선(853)된다. 그러한 평가는 인공 지능 또는 인간으로부터 유래된 이의 제기의 편견이나 편견 없이 수행된다. 인정(852) 또는 개선(852) 시나리오와 무관하게, 높은 신뢰도(846)의 결과는 KV(805)로 전달되고 낮은 신뢰도(845)의 결과는 추후 분석을 위해 AC(808)로 전달된다.

[0329] 도 137 및 138은 LOM의 데이터 기반 인텔리전스가 저장되고 병합되는 중앙 지식 보유(CKR)의 내부 세부 사항을 보여준다. 정보 유닛는 UKF1(855A), UKF2(855B), UKF3(855C)의 세 가지 유형이 있는 유닛 지식 형식(UKF)으로 저장된다. UKF2(855B)는 대상 정보가 값(865H)으로 강조 표시된 규칙 구문 형식 (RSF)(538)에 저장되는 기본 형식이다. 인덱스(856D)는 디지털 저장 및 처리 호환 / 불만 참조 포인트로 대규모의 데이터 모음에 대한 자원 효율적인 참조를 허용한다. 이 주요 정보 블록은 타임 스탬프(856C)를 참조하며 이는 UKF1(855A)로 알려진 인덱스(856A)를 통해 별도의 지식 유닛에 대한 참조이다. 이러한 유닛은 UKF2(855B)와 동일한 타임 스탬프(856C) 섹션을 보유하지 않지만 대신 RSF(538) 형식으로 값(856H) 섹터에 타임 스탬프에 대한 많은 정보를 저장한다. 규칙 구문 형식(RSF)(538)은 참조 규칙을 추적하기 위한 구문 표준의 세트이다. RSF(538) 내의 다수의 규칙 유닛이 단일 객체 또는 동작을 기술하는데 활용 될 수 있다. RSF는 CTMP에서 직접 많이 사용된다. UKF1(855A)는 UKF3(855C) 인스턴스의 인덱스(856G)에 대한 참조인 소스 속성(856B) 섹터를 포함한다. UKF3(855C) 유닛은 UKF1(855A)의 역으로 타임 스탬프 섹션은 있지만 소스 속성 섹션은 아니다. 이것은 UKF3(855C)가 소스 속성(856E 및 856B) 내용을 RSF(538)의 값(856H) 섹터에 저장했기 때문이다. 원본 속성은 요청된 정보 소스를 추적하는 복잡한 데이터의 모음이다. 그러한 출처는 KCA(816D)에서 처리된 확증 및 부정 요인으로 인해 신뢰성 및 신뢰성의 지위를 부여 받는다. 따라서 UKF 클러스터(854F)는 관할 구역에 따라 분리된 정보(시간과 출처가 동적으로 정의됨)를 정의하기 위해 연결된 UKF 변형 체인으로 구성된다. 요약하면: UKF2(855B)는 주요 대상 정보를 포함한다. UKF1(855A)은 타임 스탬프 정보를 포함하므로 무한한 회귀를 피하기 위해 타임 스탬프 필드 자체를 생략한다. UKF3 855C는 소스 속성 정보를 포함하므로 무한한 회귀를 피하기 위해 소스 필드 자체를 생략한다. 모든 UKF2(855B)는 적어도 하나의 UKF1(855A)와 하나의 UKF3(855C)를 동반해야하며, 그렇지 않으면 클러스터(시퀀스)가 불완전한 것으로 간주되고 LOM 전체 시스템에 미치는 일반 논리(859)에서 해당 정보를 아직 처리 할 수 없다. 중앙 UKF2(855B)(중앙 집중식 정보 포함)와 해당 UKF1 (855A) 및 UKF3(855C) 장치 사이에는 연결된 브리지 역할을 하는 UKF2(855B) 장치가 있을 수 있다. 일련의 UKF 클러스터(854D)는 KCA(816D)에 의해 처리되어 파생된 주장(854B)를 형성한다. 마찬가지로 일련의 UKF 클러스터(854E)는 KCA(816D)에 의해 처리되어 파생된 주장(854C)를 형성한다. 지식 확증 분석 (Knowledge Corroboration Analysis, KCA)(816D)는 UKF 클러스터된 정보가 독창적인 태도에 관한 확증적인 증거를 얻기 위해 비교되는 곳이다. 이 알고리즘은 그러한 주장이 제기되었을 때, 증거를 부정하는 등 귀속된 출처의 신뢰성을 고려한다. 따라서 KCA(816D) 처리가 완료된 후 CKR(806)은 주제(854A)에 대한 의견이 제시된 입장을 출력 할 수 있다. 거짓으로 판별된 정보조차도 미래의 진실과 거짓을 구별하는데 유용할 수 있기 때문에 CKR(806)은 결코 정보를 삭제하지 않는다. 따라서 CKR(806)은 CKR(806)의 무제

한으로 증가하는 데이터 세트를 처리하고 확장 할 수 있는 고급 스토리지 공간 서비스(854G)를 실행한다.

[0330] 도 139는 LOM의 일반적인 평가 및 의사 결정 기능을 향상시키기 위해 지속적으로 CKR 806에 새로운 지식을 제공하려는 자동화된 연구 매커니즘(ARM)(805B)을 보여준다. 사용자 활동(857A)에 표시된 대로; 사용자가 사용 가능한 프론트 엔드를 통해 LOM과 상호 작용할 때 개념이 질문/ 주장에 응답/ 응답과 관련하여 직접 또는 간접적으로 가져온다. 사용자 활동(857A)은 CKR(806)이 낮거나 전혀 관련이 없는 개념을 궁극적으로 산출할 것으로 기대된다. 요청되었지만 사용 불가능한 개념 목록(857B)에 나와있다. 개념 분류 및 우선 순위(CSP)(821B); 개념 정의는 3 개의 독립적인 소스로부터 수신되고 정보 요청(IR)(812B)의 리소스(대역폭 등)를 우선하도록 집계된다. 이러한 모듈IR(812B)은 특정 소스를 얻기 위해 관련 소스에 액세스한다. 이러한 정보는 개념 유형에 따라 정의된다. 이러한 소스는 공개 뉴스 소스(857C)(공개 뉴스 기사, 즉 로이터 통신사, 뉴욕 타임즈, 워싱턴 포스트 등), 공개 데이터 아카이브(857D) (위키피디아, 쿼라 등 정보 집계 모음) 및 소셜 미디어857E (즉 페이스북, 트위터 피드 등). 이러한 정보 소스가 제공하는 데이터는 요청된 컨셉 정의에 따라 정보 집합자(IA)(821B)에서 수신 및 구문 분석된다. 검색 시간, 검색 소스와 같은 관련 메타 데이터가 유지된다. 그 후, 정보는 상호-참조 분석(CRA)(814B)으로 보내지고, 여기서 수신된 정보는 CKR(806)으로부터의 선재 지식과 비교되고 구성된다. 이것은 CKR(806)이 현재 알고 모르는 것에 따라 새로운 들어오는 정보가 평가된다. 문체분석 스캐닝(SS)(808B)은 CRA(814B)가 문체 분석 서명을 고려하여 CKR(806)의 기존 지식으로 새로운 정보를 흡수할 수 있게 해주는 보완 모듈이다. 누락된 종속성 개념(857F)는 논리적으로 초기 목표 개념을 이해하기 위한 기초로 이해하도록 요구되는 개념이다.(즉, 트럭 작동 방식을 이해하려면 먼저 디젤 엔진의 작동 방식을 연구하고 이해해야 한다.) 그러한 누락된 개념은 처리를 위해 CSP(821B)로 전송된다. 활동 개념(857G)목록은 CKR(806)에서 가장 활발한 것으로 랭킹 된 인기 있는 주제다. 이러한 개념(857G)는 창조적 개념 생성기(CCG)(820B)로 전송된 후 새로운 잠재적 개념을 생성하기 위해 창의적으로 일치된다(창조성 모듈(18)을 통해). 이 메커니즘은 이러한 혼합물 중 하나가 IR(812B)에 연결된 소스 857C, 857D, 857E의 새로운 정보 범위를 산출할 가능성에 달려 있다.

[0331] 문체 문석의 사용 예:

[0332] 새로운 외국 데이터(858A)는 알려진 CNN 기자로부터 온 것으로 표시된다. 그러나 군사 견제 탱크의 서명과 매우 강한 문체 분석 매치가 일치한다. 따라서 내용은 주로 CKR(806) 내에서 군사적 사고 탱크에 기인했으며 CNN에서 나온 것으로 주장했다. 이것은 LOM 로직의 추후 실행을 위해 패턴 매칭과 음모 탐지를 추가로 가능하게 한다(예를 들어 CNN의 컨텐츠에 대한 미래의 주장을 불신). 주장 내용, 충돌 및 바이어스 평가는 그 내용이 마치 CNN이 아닌 싱크 탱크에서 온 것처럼 평가된다.

[0333] 도 140은 (시스템이 아직 노출되지 않은)새로운 외부 컨텐츠의 문체 분석 서명(858C)을 분석하는 문체 분석 스캐닝(SS)(808)을 도시한다. 문체 분석(Styleometry)은 한 작가 또는 장르와 다른 작가 사이의 문학적 스타일 변화에 대한 통계 분석이다. 이는 데이터 / 주장에 대한 소스 기대치를 추적하는 데있어 CKR(806)을 보조하며, 이는 LOM이 입증된 주장을 검출하는데 더 도움이 된다. 새로운 외부 데이터(858A)의 서명 결론(SC)(819B) 내용 출처 속성은 문체분석 서명(858C)의 주요 일치 사항의 영향을 받는다. 문체 분석 매치 일치가 강할수록 스타일 측정에 따른 소스 속성이 강해진다. 서명 쿼리(SQ)(807B)의 경우, 문체분석 서명(858C)는 SI(813B)의 알려진 모든 서명과 일치한다. 모든 중요한 그라디언트의 일치가 기록된다. 서명 인덱스(SI)(813B)는 CKR (806)으로부터 검색된 것으로 알려진 모든 문체문석 서명(858C)의 리스트를 나타낸다. 제 3 문체 분석 알고리즘(858B)으로 대표되는 LOM은 정식으로 선택된 진보적이고 효과적인 알고리즘 문체분석 알고리즘에 의존한다.

[0334] 도 141은 주장 또는 질문의 형태로 명제를 수신하고 그러한 명제와 관련된 개념의 출력을 제공하는 추정 중단 시스템(AOS)(815B)를 도시한다. 개념 정의 매칭(CDM; Concept Definition Matching)(803B)은 인간 주체(800)에서 제공하는 하드 코딩된 가정(858D)이 의존성 해석(DI; Dependent Interpretation)(816B)모듈에 대해 쿼리되는 곳이다. 그러한 모든 개념은 위반 우려로 윤리적 사생활보호 법(EPL)(811B)에 의해 검사된다. 의존성 해석(DI)(816B) 모듈에서 요청된 데이터의 응답을 충족시키는 모든 지식 기반 종속성에 액세스한다. 이렇게 하면 객관적인 의견을 지닌 정보의 완전한 '나무(tree)'가 검색된다. 요청된 데이터(858E)는 LOM 전체 시스템에 미치는 일반 논리(859)가 요청한 데이터로서 특정 쿼리인지 조건부 쿼리인지를 나타낸다. 특정 쿼리는 정확히 표시된 정보 세트를 찾는다. 조건부 쿼리는 특정 조건과 일치하는 모든 정보를 요청한다.

[0335] 도 142는 지능형 정보 및 구성 관리(I^2CM)(804E) 및 관리 콘솔(804D)을 도시한다. 집합체(860A)는 여러 플랫폼의 정보 스트림을 병합 및 태깅하면서 중요하지 않은 정보와 중복 정보를 필터링하기 위해 일반적인 레벨 기준을 사용한다. 위협 딜레마 관리(860B)는 개념적 데이터 위험이 조감도에서 감지되는 곳이다. 이러한 위협은 그래픽으로 표현하기 위해 관리 콘솔로 전달된다. 위협 메커니즘과 관련된 계산된 측정 값이 마침내 여러 플랫폼

에서 통합되므로, 보다 정보에 입각한 위협 관리 결정을 자동으로 수행 할 수 있다. 자동화된 컨트롤(860C)는 MNSP(9), 신뢰 플랫폼(860Q), 제3 서비스(860R)의 관리 관련 컨트롤을 제어하는 알고리즘 액세스를 나타낸다. 관리 피드백 컨트롤(860D)은 모든 MNSP(9) 클라우드, 신뢰 플랫폼(TP)(860Q), 및 정책 수립, 법의학, 위협 조사를 용이하게 하는 데 사용할 수 있는 추가적 3 차 서비스(860R) 기반 서비스의 고급 컨트롤을 제공한다. 이러한 관리 제어(860D)는 결국 적절한 사용자 정의 가능한 비주얼 및 표현 효율성으로 관리 콘솔(MC)(804D)에 표시된다. 이를 통해 단일 인터페이스에서 직접 전체 시스템(MNSO, TP, 3PI)을 효율적으로 제어하고 조작할 수 있어 필요에 따라 세부 사항을 확대 할 수 있다. 수동 제어(860E)는 인간의 액세스를 위해 MNSP(9), 신뢰 플랫폼(860Q) 및 제3 서비스(860R)의 관리 관련 제어를 컨트롤한다. 지능적 맥락화 860F 단계에서 나머지 데이터는 이제 아일랜드의 클러스터처럼 보인다. 각 아일랜드은 개념적 데이터 위험이다. 개념 분석을 성숙시키기 위해 플랫폼 간의 상관 관계가 만들어진다. 위험한 패턴을 이해하기 위해(I^2GE(21)에서 LIZARD로) 역사적 데이터에 액세스하고, CTMP(22)는 비판적 사고 분석에 사용된다. 구성 및 배포 서비스(860G)는 올바른 개념 데이터 구성 및 연결 설정으로 새로운 기업 자산(컴퓨터, 랩톱, 휴대폰)을 배포하기 위한 인터페이스이다. 장치를 추가하고 설정 한 후에는 관리 피드백 컨트롤(860D)을 중개자로 하여 관리 콘솔(MC)(804D)을 통해 조정할 수 있다. 또한 이 서비스는 새로운 고객 / 클라이언트 사용자 계정의 배포를 관리한다. 이러한 배포에는 하드웨어와 사용자 계정, 인터페이스 사용자 정의, 고객 / 클라이언트 변수 목록(예: 비즈니스 유형, 제품 유형 등)이 포함될 수 있다. 관할 구역에 의한 분리(860H)는 정보의 태그 풀은 MC(804D) 사용자의 관할권에 따라 독점적으로 분리된다. 위협에 의한 분리(Separation by Threat)(860I) 정보는 개별 위협(즉, 개념적 데이터 위험)에 따라 구성된다. 모든 유형의 데이터는 위협에 연관되어 표시되며 자세한 정보가 추가되거나 제거된다. 직접 관리(860J)는 MC(804D) 사용자가 수동 제어 장치(860E)를 통해 관리 피드백 제어 장치(860D)에 연결하기 위한 인터페이스이다. 카테고리 & 관할 구역(860H)에서는 MC(804D) 사용자가 자신의 관할 구역과 정보 카테고리 액세스 범위를 정의하는 로그인 자격 증명을 사용한다. 모든 잠재적인 데이터 벡터(860L)은 동작중인 데이터, 휴식중인 데이터 및 사용중인 데이터를 나타낸다. 사용자 정의 가능한 비주얼(860M)은 회계, 금융, 인사, IT, 법률, 보안 / 감찰사, 개인 정보 보호 / 공개, 노조 등 다양한 기업 부서 및 이해 관계자 (각 부서의 임원, 관리자, 임원) 파티 파트너, 법 집행 기관 등이다. 개념적 데이터의 모든 측면에 대한 통합보기(860N)은 경계, 기업, 데이터 센터, 클라우드, 이동식 미디어, 모바일 장치 등을 나타낸다. 통합 단일 뷰(860O)는 모니터링, 로깅, 보고, 이벤트 상관, 경고 처리, 정책 / 규칙 세트 작성, 수정 작업, 알고리즘 튜닝, 서비스 제공(신규 고객 / 수정), 신뢰 플랫폼은 물론 제3 서비스(제3 서비스 제공 업체 및 공급 업체로부터 보고서 및 경고 / 로그 수신 등)를 제공한다. 개념 데이터 팀(Conceptual Data Team)(860P)는 여러 시스템의 활동 및 상태를 모니터하는 자격을 갖춘 전문가 팀이다. 지능형 정보 처리 및 인공 지능(AI) 결정이 이루어지고 있기 때문에 적은 수의 경험으로 적은 인력을 고용함으로써 비용을 절감 할 수 있다. 팀의 주요 목적은 대규모 분석 포인트를 수행하는 동안 시스템이 원하는 기준에 따라 성숙하고 진행 중임을 확인하는 대체 단계가 되는 것이다.

[0336] 도 143은 개인의 개인 정보가 다수의 잠재적 엔드 포인트 및 프론트 엔드를 통해 저장되는 개인적인 지능 프로필(PIP)(802C)을 도시한다. 그들의 정보는 CKR(806)으로부터 매우 안전하고 격리되어 있지만 LOM 전체 시스템에 미치는 일반 논리(859)에서 매우 개인화된 의사 결정을 수행 할 수 있다. 개인적 인증 및 암호화(PAE; Personal Authentication & Encryption)(803C)를 구현함으로써 들어오는 데이터 요청은 올바른 사용자가 독점적으로 개인 정보에 액세스 할 수 있도록 먼저 인증 해야 한다. 인공 지능 어플리케이션과 관련된 개인 정보는 암호화되어 UKF형식의 개인적 클러스터 풀(815C)에 저장된다. 정보 익명화 처리(IAP)(816C) 정보는 개인 식별 정보를 제거한 후 CKR(806)에 보충된다. 이러한 개인 정보가 데이터 스트림에서 제거된 후에도 IAP(816C)는 개인의 신원을 확인하기 위해(포렌식 탐지 작업과 같이) 리버스 엔지니어링 될 수 있는 너무 많은 병렬 데이터가 제공되는 것을 방지하려고 시도한다. 수신된 상호 참조 분석(CRA)(814B) 정보는 CKR(806)으로부터의 선재 지식을 고려하여 비교되고 구성된다. 이것은 새로운 들어오는 정보가 CKR(806)이 현재 알고 있고 모르는 것에 따라 평가되고 유효화 될 수 있게 한다. 임의의 데이터 요청 정보는 항상 CKR(806)으로부터 액세스된다. 데이터 요청에 개인적인 기준이 있는 경우 PIP(802C)는 개인 및 일반 데이터 병합(PGDM)(813C)를 통해 참조되며 주요 CKR(806) 지식을 기반으로 한다.

[0337] 도 144는 생활 습관 및 격리된 사건에 대한 작업을 자동화하는 응집력 있는 플랫폼에서 다양한 인터넷 가능 장치 및 서비스를 연결하는 생활 관리 및 자동화(LAA)(812D)를 보여준다. 적극적 의사 결정(ADM)(813D)은 LAA(812D)의 핵심 논리이며 FARM(814D)에 따라 사용 가능한 지출 규칙 및 금액, 프런트 엔드 서비스(861A), 백엔드 서비스(861B), IoT 장치(862A)의 가용성 및 기능을 고려한다. 자금 조달 규칙& 관리(FARM)(814D)를 통해 인적 자원은 ADM(813D)에 활동의 관할권을 알리기 위해 이 모듈의 기준, 범위 및 범위를 수동으로 정의한다. 인간 주체(800)는 디지털 지갑(Digital Wallet)(861C)에 수동적으로 암호화된 화폐 자금(즉, 비트 코인

(Bitcoin))을 입금함으로써 LAA(812D)가 소비 할 수 있는 금액의 상한선을 의미한다. IoT 상호 작용 모듈(IIM)(815D)는 IoT 장치(862A)가 인간에게 제공되는 데이터베이스를 유지 관리한다. 인증 키들 및 메커니즘들은 IoT 장치들(862A)의 보안 제어(862C)를 가능하게 하기 위해 여기에 저장된다. 제품 제조업체/ 개발자(861F)는 LAA(812D)에 프로그래밍 가능한 API (어플리케이션 프로그래밍 인터페이스) 엔드 포인트들을 IoT 제품 상호 작용 프로그래밍(861E)으로 제공한다. 이러한 엔드 포인트는 IoT 상호작용 모듈(IIM)(815D)에서 특별히 사용된다. 데이터 피드(862B)는 IoT 가능 디바이스(862A)가 정보를 LAA(812D)로 전송하여 지능형 및 자동화된 동작이 수행될 수 있을 때를 나타낸다. 예: 온도를 알려주는 온도 조절 장치, 우유 재고를 알려주는 냉장고. 장치 컨트롤(862C)는 IoT 가능 디바이스(862A)가 동작을 수행하기 위해 LAA(812D)로부터 명령을 수신 할 때를 나타낸다. 예: 에어컨의 전원을 켜고, 소포 배달을 위한 문을 열기. 프런트 엔드 서비스(861A)의 카테고리는 다음을 포함할 수 있다:

[0338] - 인공 지능의 개인 비서

[0339] - 통신 어플리케이션 및 프로토콜

[0340] - 홈 자동화

[0341] - 의료용 인터페이스

[0342] - 배달 추적 서비스

[0343] 백 엔드 서비스(861B)의 예는 다음을 포함 할 수 있다:

[0344] - 아마존 온라인 주문

[0345] - 우버(Uber) / 수송 기관

[0346] - 의료 처방전

[0347] LAA(812D)의 기능을 설명하기 위한 전반적인 사용 예는 다음과 같다:

[0348] IoT 가능 냉장고는 우유가 부족하다는 것을 감지한다. LOM은 우유(전유, full fat milk)를 마시지 않으면 피험자의 기분이 더 부정적인 경향이 있다는 감성 지능을 통해 분석을 실시한다. LOM은 환자의 생활상의 위험 및 이점을 평가한 후 온라인 배달 서비스(즉 아마존)에서 우유를 주문한다. LOM은 추적 번호를 통해 우유 선적을 추적하고 주택 정문을 열어 주택 소유지 내에서 배달되도록 한다. LOM은 배달원이 출발 한 후에 문을 닫고, 배달원이 악의적인 행위자인 경우 보안상의 주의를 기울인다. 그 후 손재주 기능이 있는 간단한 바퀴 달린 로봇이 우유를 집어 냉장고에 넣고 차갑게 유지하며 상하지 않도록 한다.

[0349] 도 145는 비 윤리적 및/ 또는 불법적인 자료를 확인하기 위해 사용자의 개인 식별 가능 데이터 요청을 모니터하는 행동 모니터링(BM)(819C)을 보여준다. 메타 데이터 집합(MDA; Metadata Aggregation)(812C) 사용자 관련 데이터는 외부 서비스에서 집계되어 사용자의 디지털 신원(즉, IP 주소, MAC 주소 등)을 설정할 수 있다. 이러한 정보는 유도(820C) / 추론(821C)으로 전송되고, 결국 PCD(807C)로 전송되며, 정교한 분석은 MNSP(9)의 뒷받침 요인으로 수행된다. 예 : amazon.com 쇼핑 포털과 프론트 엔드로 인터페이스하는 사용자는 보안 목적으로 LOM의 행동 모니터링(BM)(819C)로 IP 주소를 전달한다. PIP 802C로 예정된 인증 된 사용자의 모든 정보는 정보 추적(IT)(818C)을 거쳐 행동 블랙리스트(864A)와 대조된다. 예: 사용자가 유황의 화학적 조성에 대해 질문한다. 블랙리스트(863B)의 요소와 (부분적으로 또는 전체적으로)일치하는 정보는 IT(818C)에서 유도(820C) 및 추론(821C)으로 전송된다. 프리 크라임 검출(PCD; Pre-Crime Detection)(807C)에서 추론 및 유도 정보가 병합되어 범죄 전 결론을 분석한다. 상당한 양의 확증이 탐지되면 사용자의 문제가 되는 정보와 알려진 신원이 법 집행기관에 전달된다. PCD(807C)는 행동 블랙리스트(864A)를 직접 참조하는 CTMP(22)를 사용하여 유도(820C) 및 추론(821C)에 의해 생성된 입장을 검증한다. 블랙리스트 유지관리 기관(BMA)(817D)은 MNSP(9)의 클라우드 서비스 프레임 워크 내에서 작동한다. BMA(817D)는 범죄 예방 및 범죄자 포착을 위해 사용자 모니터링이 필요한 위험 개념을 정의한 행동 블랙리스트(864A)를 발행 및 유지한다. 또한 BMA(864B)는 민감한 자료에 플래그를 지정하여 LOM에 의한 질의 결과로 제출되지 않도록 EPL(윤리적 사생활보호 법) 블랙리스트(864B)를 발급 및 유지한다. 민감한 자료에는 유출된 문서, 개인 정보(예: 주민 등록 번호, 여권 번호 등)가 포함될 수 있다. BMA(864B)는 윤리, 사생활 및 법적(즉, 사이버 보안 정책, 허용되는 사용 정책, HIPAA, PII 등)과 관련하여 관련 법률 및 정책을 해석한다. 블랙리스트는 대개 이러한 개념과 관련이 있는 경우 의심스러운 것으로 간주되는 트리거 개념으로 구성된다. 블랙리스트는 특정 개인 및 / 또는 원하는 목록과 같은 조직을 대상으로 할 수도 있다. 미래의 범죄 예방은 BM(819C) 내에서 발생하며 MNSP(9)로 확인 된 증거가 있다. 법 집행 권한(864C)은 MNSP(9) Cloud를 통

해 BMA(817D)에 연결하여 블랙리스트 개념에 대한 정보를 제공하고 BM (819C)의 PCD(807C)의 범죄 탐지 결과로부터 정보를 제공받을 수 있다. 행동 모니터링 정보 확인(864D)은 MNSP(9)가 뒷받침 목적으로 BM(819C)에 행동 모니터링 정보를 제공 할 수 있게 한다. 윤리적 사생활보호 법(EPL)(811B)은 MSNP에서 사용자 지정 블랙리스트를 받고 AOS(815B)를 사용하여 비 윤리적인 개인 정보 보호 및 / 또는 불법 자료가 포함된 주장을 차단한다.

[0350] 도 146은 윤리적 사생활보호 법(EPL)(811B)이 MSNP에서 사용자 정의 블랙리스트를 수신하고 AOS(815B)를 사용하여 비 윤리적 인 개인 정보 보호 및 / 또는 불법적인 자료가 포함된 주장을 차단하는 것을 보여준다. MNSP(9)는 트로이 목마, 바이러스 등을 통한 해킹 시도와 같은 전통적인 보안 위협을 처리하는 데 사용된다. LOM의 BM(819C) 및 EPL(811B) 모듈은 윤리, 개인 정보 보호 및 법적 영향을 결정하기 위해 유도(820C) 및 추론(821C)을 통해 개념 데이터의 문맥을 분석한다.

[0351] 도 147은 LIZARD 알고리즘의 개요를 도시한다. 동적 쉘(DS)(865A)은 반복을 통해 변경되기 쉬운 LIZARD의 계층이다. 그 목적을 달성하기 위해 고도의 복잡성을 필요로 하는 모듈은 대개 여기에 속한다. 프로그래머 팀이 처리 할 수 있는 복잡성 수준을 능가 할 것이기 때문이다. 구문 모듈(SM)(865B)은 컴퓨터 코드를 읽고 쓰는 데 필요한 프레임 워크이다. 쓰기를 위해; PM으로부터 복잡한 형식화된 목적을 수신한 다음 임의의 코드 구문으로 코드를 작성한 다음 도우미 기능이 임의의 코드를 실제 실행 코드(원하는 언어에 따라 다름)로 변환 할 수 있다. 읽기 위해; 그러한 코드의 기능을 위한 목적을 도출하기 위해 PM(865E)에 대한 코드의 구문 해석을 제공한다. LIZARD가 낮은 신뢰 결정을 수행하면 LIZARD의 향후 반복을 개선하기 위해 관련 데이터를 데이터 리턴 릴레이(DRR)(865C)를 통해 ACT(866)에 중계한다. LIZARD 자체는 의사 결정을 수행하는데 데이터에 직접 의존하지 않지만 진화하는 위협에 대한 데이터는 LIZARD의 향후 반복이 수행 할 수 있는 선험적 결정에 간접적으로 도움이 될 수 있다. ACT (Artificial Concept Threat) (866)는 가상화 된 개념적 데이터 위험을 가진 가상 테스트 환경을 만들어 반복 프로세스를 가능하도록 한다. ACT(866)의 인공적인 진화는 악의적인 개념 형성의 유기적 진화를 앞당기기에 충분하다. 반복 모듈(IM)(865D)은 SC(865F)를 사용하여 DRR(865C)의 '고정 목표' 및 데이터에서 정의된 목적에 따라 DS(865A)의 코드 기반을 구문적으로 수정한다. 이 수정된 버전의 LIZARD는 ACT(866)에 의해 여러 가지 및 다양한 개념적 데이터 위험 시나리오로 스트레스 테스트(병행)된다. 가장 성공적인 반복은 라이브 기능 버전으로 채택된다. 목적 모듈(PM)(865E)은 SM(865B)을 사용하여 코드에서 목적을 추출하고, 목적을 자신의 '복합 목적 형식'으로 출력한다. 그러한 목적은 SM(865B)에 의해 해석된 것처럼 코드 블록의 의도된 기능을 적절하게 기술해야 한다(비록 그 코드가 데이터에 은밀하게 포함된 경우라 할지라도). 정적 코어(SC)(865F)는 자동화된 반복을 통해 변경되는 부분이 가장 적은 LIZARD의 계층이며, 대신 인간 프로그래머가 직접 변경한다. 특히 가장 안쪽에 있는 어두운 사각형은 자동화된 반복에 전혀 영향을 받지 않는다. 이 가장 안쪽 층은 LIZARD의 방향과 전체 수용력을 안내하는 나무 뿌리와 같다.

[0352] 도 148는 개념적 데이터의 다양한 위험에 적응할 때 정적 규칙 집합이 성숙되는 방식을 설명하는 반복적인 지능 성장($I^2GE(21)$의 하위 집합)을 보여준다. 일련의 세대 별 규칙 집합이 생성되며, 이들의 진화는 '성격'특성 정의를 통해 전달된다. 이러한 룰 세트는 들어오는 개념적 데이터 피드를 처리하고 가장 바람직한 알림 및 수정 조치를 수행하는 데 사용된다. 개발 경로(867A)는 일관된 '성격'을 지닌 일련의 세대이다. CPU 시간이 진행됨에 따라 세대가 점점 더 동적으로 변한다. 초기 정적 룰 세트는 덜 우세하여 잠재적으로 지워지거나 무시된다. 예: 개발 경로 A는 엄격하고 신중한 특성을 가지고 있으며 용서 또는 용인이 거의 없다. 개념 행동(867B)는 개념적 데이터 분석가의 행동이 진화 경로(867A)에서 배울 수 있도록 처리되고 저장되는 곳이다. 예: 통로 A는 특정 상황과 성격 유형이 낙관적 인 개념 데이터 위험에 대한 많은 반응을 발견했다. 경로 A는 그런 행동을 모방한 규칙을 만든다. 인간(867C)은 진화적 체인을 시작하기 위한 초기 규칙 집합을 만드는 개념적 데이터 분석가를 나타낸다. 예: 암거래 시장에서 플루토늄 구매와 관련된 모든 개념이 차단되는 규칙이 정의된다. 경로 특성(867D)은 개념적 데이터 위험 유발 요인에 대해 실행되어야 하는 반동적 특성을 정의하는 변수 집합이다.

[0353] 도 149 및 150은 병렬 진화 경로(867A)가 성숙되고 선택되는 방법인 반복 진화($I^2GE(21)$의 하위 집합)를 보여준다. 반복적인 세대는 동일한 ACT(866)에 적응하고, 최고의 성격 특성을 가진 통로는 가장 큰 개념의 위협에 저항한다. CPU 시간(868A)는 시간에 따른 CPU 전력의 측정 값이며 CPU 사이클 / 초 단위로 측정 할 수 있다. 진화론적 경로가 받는 노출의 양을 측정하기 위해 시간만 사용하는 것은 각 CPU의 코어 및 전력을 고려해야 하기 때문에 불충분하다. 예 : 인텔 펜티엄 3(Intel Pentium III)가 1000년 걸리는 요청을 처리하는데 인텔 해스웰(Intel Haswell) 프로세서가 30 분 정도 걸릴 수 있다. 가상 분리(Virtual Isolation)(868B)를 사용함으로써 모든 진화 경로는 사실상 격리되어 반복을 자신의 성격에 따라 결정한다. 예: 경로 B는 경로 C가 어려운 개념적 데이터 문제를 해결했음을 전혀 모르고 있으며 해결책을 계산하기 위해 자신의 개성과 학습된 데이터에 의존해

야 한다. 특정 경로는 개념적 데이터 위험을 인식 할 수 없는 무기한 상태에 도달했기 때문에 868C에서 폐기 될 수 있다. 가장 가능성 있는 결과는 새로운 경로가 개성 있는 성격으로 만들어 져야 한다는 것이다. 예 : 경로 D는 100 시간의 CPU 시간 868A에 대한 개념적 데이터 위험을 인식하지 못했다. 따라서 전체 경로가 폐지되었다. 모니터링 / 상호 작용 시스템(868D)은 ACT(866) 시스템으로부터 개념 데이터 위험 트리거를 주입하고 개념 행동 클라우드로부터의 관련 개념 데이터 위험 응답을 모두 전달하는 플랫폼이다(모두 지정된 성격 특성에 따라). 예 : 모니터링 시스템은 경로 B에게 12 세대를 공식화하는 데 필요한 개념적 데이터 위험 응답을 제공했다. 인공 개념 위협(ACT)(866)은 일관된 개념 데이터 위험 환경을 제공하는 격리된 시스템이다. 분석가가 다양한 잠재 개념 데이터 응답 및 특성을 인식하도록 시스템을 연습하고 훈련시키는 개념 인식 훈련을 제공한다. 예 : ACT는 사람이 위험하다고 인식 할 수 있는 일련의 복잡한 개념을 제공했다. "가정용 재료를 사용하여 사린 가스를 화학적으로 합성하는 방법"와 같은. 실제 개념 위협(RCT)(869A)는 실제 데이터 로그에서 개념적 시나리오(869C) 실제 위협을 제공한다. 인간(867C) 모니터링 / 상호 작용 시스템(868D)에 직접 주문(869B)를 제공한다. 예: 통로를 수동으로 중단하고 경로 성격의 마스터 변수를 변경한다. 상호 참조 모듈(869D)는 개념적 위험(869C)와 개념 분석(867C)이 작성한 응답(869E) 사이의 분석적 브릿지이다. 의미있는 동작을 추출한 후 특성 태깅 모듈(869F)에 푸시한다. 개념적 위험(869C)은 실제 위험(869A) 또는 훈련(866)으로부터 올 수 있다. 특성 태깅 모듈(869F)은 성격 유형에 따라 모든 행동을 분할한다. 예: 개념적 데이터 분석가(867C)가 자살 방법론을 과도하게 언급한 이메일을 위험한 것으로 표시한 경우(869E), 이 모듈은 과거의 사건과의 행동 중복 때문에 이를 예방적 성격으로 표시했지만 분석가는 스스로 경고하는 경고자이기 때문에 이를 표시했다. 특성 상호 작용 모듈(869G)은 상이한 성격들 간의 상관 관계를 분석한다. 이 정보는 개념 행동(867B)으로 전달되며, 이는 모니터링/ 상호 작용 시스템(868D)과 경로 자체로 전달된다. 예: 용서하지 않고 현실적인 성격은 사용량이 크게 중복되고 동일한 이벤트에 대해 비슷한 응답을 반환한다. 그러나 엄격하고 긍정적인 같은 사건에 대해 거의 비슷한 반응을 보이지 않는다.

[0354] 도 151 내지 154는 이전 입력 양식에서 새로운 하이브리드 양식을 작성하는 지능형 알고리즘인 창조성 모듈(18)을 보여준다. 창조성(18)은 여러 알고리즘을 처리하기 위한 플러그인 모듈로 사용된다. 도면 부호 870A에서, 두 개의 상위 형식(이전 형식)이 하이브리드 형식(870B)을 생성하기 위해 지능적 선택(Intelligent Selector)으로 푸시된다. 이 양식은 추상적인 데이터 구조를 나타낼 수 있다. 예: 방식 A는 개념 DB에서 파생 된 위험 개념의 평균 모델을 나타낸다. 방식 B는 위험한 개념에 대응하는 방법에 대한 개념적 트리거 규칙 집합에 의한 새로운 정보 공개를 나타낸다. 방식 B의 정보는 방식 A가 나타내는 것보다 더 위험한 개념으로 생산된 혼합 양식을 허용한다. 지능적 선택(870B) 알고리즘은 새로운 기능을 선택하고 하이브리드 형식으로 병합한다. 예 : 방식 A는 개념 DB에서 파생된 개념적 데이터 위험의 평균 모델을 나타낸다. 방식 B는 사전 개념적 위험에 대응하는 방법에 대한 개념 룰 세트에 의한 새로운 정보 공개를 나타낸다. 방식 B의 정보는 양식 A가 나타내는 것보다 더 나은 개념 위험 유발 요인으로 생성된 혼합 양식을 허용한다. 모드(870C)는 창조성 모듈(18)이 사용되는 알고리즘의 유형을 정의한다. 이렇게 하면 지능적 선택(870B)은 사용중인 어플리케이션에 따라 병합에 적합한 부분을 알 수 있다. 예: 모드는 ACT(866)로 설정되므로 지능적 선택(870B)은 예상되는 입력 데이터가 위험 DB 표현(방식 A)이고 새로 발표된 정보가 개념 위험 트리거(방식 B)에 대한 룰 세트 반응을 상세히 알고 있음을 알고 있다. 속성 모드(870C)는 효과적인 하이브리드 형식을 생성하기 위해 새 데이터를 이전 데이터와 가장 잘 병합하는 방법에 대한 자세한 방법을 정의한다. 정적 기준(870D)은 방식을 병합하는 방법에 대한 일반적인 사용자 지정을 제공하는 개념적 데이터 분석가가 제공한다. 그러한 데이터는 모드(870C)가 선택되는 것에 달려있는 우선 순위 랭킹, 원하는 데이터 비율 및 병합을 유도하는 데이터를 포함 할 수 있다. 예: 모드(870C)가 ACT(866)로 선택된 경우 실패한 위험 트리거의 결과 정보는 위험 트리거 DB에 크게 영향을 주어 이러한 트리거의 구성을 크게 변경해야 한다. 그러한 변화 후에도 트리거가 계속 실패하면 트리거를 완전히 포기하라. 원시 비교(871B)는 개념적 데이터 분석가(867C)가 제공한 정적 기준(870D)에 따라 두 가지 수신 방식 모두에서 수행된다. 원시 비교가 수행된 후, 방식의 대부분은 정적 기준(870D)에 따라 호환되었다. 발견된 유일한 차이점은 방식 A가 정적 기준에 의해 '외부'로 표시된 응답을 포함한다는 것이다. 즉, 위험 트리거 DB 표현 방식 B는 방식 A에서 발견된 특정 예외를 포함하거나 나타내지 않는다. 순위 변경 중요성(871C)은 제공된 정적 조건(870D)에 따라 중요하고 중요하지 않은 변경을 순위 지정한다. 예: 방식 B로 표현되지 않은 방식 A에서 예외가 발견 되었기 때문에 정적 기준(870D)은 이 예외 사항이 매우 중요하다는 것을 인식하므로 혼합 양식에서 하이브리드 방식 AB를 생산하는 두드러진 수정이 이루어진다. 병합 모듈(871D)에서 동일하게 유지되고 다른 것으로 판명 된 것이 정적 표준(870D) 및 사용되는 모드(870C)를 기반으로 하이브리드 형식으로 재 조립된다. 그러한 변화에는 데이터의 비율 분포(872A), 특정 데이터가 얼마나 중요한지, 데이터가 서로 어떻게 관련되어야 하는지 등이 포함될 수 있다. 예: 예외적인 구성의 순위 중요성이 수신된다. 적절한 조정이 이루어진 후에, 정적 기준(870D)에 의해 안

내되는 프로세스는 이 예외에 대한 반응이 데이터의 다른 부분과 호환되지 않는다고 판단한다. 그런 다음 병합 프로세스는 기존 데이터를 수정하여 예외 수정이 기존 데이터와 효과적으로 혼합 될 수 있도록 한다. 중첩되는 정보의 양은 정적 기준(870D)에 의해 설정된 비율(872A)에 따라 필터링된다. 비율(872A)이 큰 값으로 설정된 경우 일관성을 유지한 많은 양의 양식 데이터가 혼합 양식으로 병합된다. 비율(872A)이 작게 설정된 경우 하이브리드 양식의 대부분은 과거 반복과 매우 다르다. 우선 순위(872B)는 두 데이터 세트가 양식의 같은 위치에 있는 지형지물을 정의하기 위해 경쟁하는 위치이며 어떤 지형지물이 눈에 잘 띄고 어느 것이 겹쳐져 숨겨져 있는지 선택하는 우선 순위 지정 프로세스가 발생한다. 하나의 특성만이 특정 지점(사각형을 통해 강조 표시됨)을 차지할 수 있는 경우 우선 순위 지정 프로세스가 발생하여 어떤 기능이 상속되는지 선택한다. 스타일(872C)은 중첩 포인트가 병합되는 방식을 정의한다. 대부분의 경우 특정 병합이 발생할 수 있는 여러 가지 방법이 있으므로 정적 조건(870D) 및 모드(870C)는 이 모듈이 다른 병합보다 특정 병합을 선호하도록 지시한다. 대부분의 경우 특성들 사이에 겹쳐진 양식이 있으므로 병합된 특성을 가진 양식을 생성 할 수 있다. 예 : 삼각형과 원이 입력 양식으로 제공되면 '팩맨(pac-man)'모양을 생성 할 수 있다.

[0355] 도 155 및 156은 개인 비서로 사용되는 LOM을 보여준다. LOM은 개인적인 생활에서 개인화된 포트폴리오를 관리하도록 구성 할 수 있다. 어떤 개인이 딜레마나 명제에 부딪힐 때 의미 있고 적절한 조언을 제공할 수 있도록 LOM이 그들의 일상 생활에 대한 개인 정보를 등록하는 것에 적극적으로 동의 할 수 있다. 이것은 상황, 일, 식습관, 구매 결정 등 다양 할 수 있다. LOM은 LOM의 내부 심의 과정(874A)를 통해 결론(874C)으로 이끄는 초기 질문(874B)을 받는다. EPL 811B는 LOM에 의해 생성된 응답의 윤리적, 법적 및 개인 정보 보호 준수를 확인하는데 사용된다. LOM을 더욱 개인화하기 위해 LOM이 데이터를 수신하고 제어할 수있는 인터넷 가능 장치에 연결하는 LAA(812D) 모듈에 연결할 수 있다. (즉, 집 근처에 도착하면 에어컨이 가동된다.) PIP(802C)에서 LOM은 사용자로부터 개인 정보를 제공받으며 사용자는 정보를 안전하게 추적하는데 동의 할 수 있다. 이렇게 하면 LOM이보다 정확한 미래 대응을 개인별로 제공할 수 있다. 문맥화(874D)와 함께 LOM은 논증을 구성 할 때 누락된 링크를 추론 할 수 있다. LOM은 원래의 주장에 의해 제기된 딜레마를 해결하기 위해 상황에 대한 특정 변수를 먼저 알아야 하거나 가정해야 한다는 고급 논리로 해독했다.

[0356] 도 157은 연구 도구로서 사용되는 LOM을 도시한다. 사용자가 LOM을 투자 도구로 사용하고 있다. 주장(875B)은 객관적이고 비 개인적인 방식으로 작성 되었기 때문에 LOM에서는 문제에 대한 정교한 의견을 형성 할 수 있도록 특정 및 격리된 사례의 추가 세부 정보(875D)를 요구하지 않는다. 그러므로 결론(875C)은 개인화된 정보 없이 도달된다. EPL(811B)은 LOM에 의해 생성된 응답의 윤리적, 합법적 및 개인 정보 보호 기반 준수를 확인하는데 사용되며 BM(819C)은 사용자를 대신하여 불법적 / 비도덕적 행위를 저지르는 음모를 모니터링하는데 사용된다.

[0357] 도 158 및 159는 LOM이 제안된 876B 이론의 장점과 단점을 탐구하는 것을 보여준다. 비트 코인은 피어 투 피어(peer-to-peer) 분산 네트워크로서 블록 체인이라는 공용 원장에서 암호 해독의 소유권을 확인한다. 발생하는 모든 비트코인 트랜잭션은 네트워크에서 10 분 간격으로 수집되는 블록에 기록된다. 비트코인 코어 클라이언트의 현재 하드 코드된 한도는 1MB이다. 즉, 10 분마다 1MB의 트랜잭션(데이터 형식으로 표시)만 있을 수 있다. 최근 비트 코인의 인기 증가로 인해 블록 크기 제한으로 인해 시스템에 긴장이 발생하고 지불 확인 시간이 오래 걸리고 마이너(miner) 비용이 비싸다. 문맥화(876D)와 함께 LOM은 논증을 구성할 때 누락된 링크를 추론 할 수 있다. LOM은 원래의 주장에 의해 제기된 딜레마를 해결하기 위해 누가 블록 크기 제한을 올릴 것인지를 먼저 알아야 하거나 추측해야 한다는 고급 논리로 해독했다. 그러므로 결론(876C)은 LOM에 의해 도달된다. EPL(811B)는 LOM에 의해 생성된 응답의 윤리적, 합법적 및 개인 정보 보호 기반 준수를 확인하는데 사용되며 BM(819C)는 사용자를 대신하여 불법적 / 비도덕적 행위를 저지르는 음모를 모니터링하는 데 사용된다.

[0358] 도 160 및 161은 외교 정책 전쟁 게임을 위한 정책 결정을 수행하는 LOM을 보여준다. 격리되고 안전한 LOM의 사례는 군대에서 승인한 하드웨어 및 시설에 활용할 수 있다. 이를 통해 LOM은 중앙 지식 보유(CKR)(806)의 일반적인 지식에 액세스 할 수 있으며 동시에 개인적인 지능 프로필(PIP)의 로컬 사례에서 군대 특정(심지어 분류된) 정보에 액세스 할 수 있다. 군인은 LOM의 고급 지능 능력으로 인해 복잡한 전쟁 게임을 실행할 수 있는 반면 일반적인 지식과 특정 지식에 액세스 할 수 있다. 최초의 전쟁 게임 시나리오는 주장(877B)과 하드 코드된 가정(877E)으로 제안된다. 전쟁 게임 시나리오의 복잡성으로 인해 LOM은 발전된 세부 사항 요청(887D)으로 응답한다. LOM은 정교한 대응을 위해 50,000 명의 군대에 대한 자세한 프로필과 같은 높은 수준의 정보를 받아야 한다고 결정할 수 있다. 이러한 정보 전송은 수 테라 바이트 크기의 데이터를 처리 할 수 있으므로 정교한 결론에 도달하기 위해 여러 날의 병렬 처리가 필요하다. 모든 정보는 표준화되고 자동화된 형식 및 프로토콜을 통해 전송된다(즉, 단일 컴퓨터 인터페이스 작업으로 2 시간 동안 50,000 개의 엑셀 시트 가져 오기). BM(819C) 및 EPL(811B)을 사용하면 정보의 민감한 특성으로 인해 그러한 보호 기능이 비활성화되도록 보안 정리 무시가 활성

화된다. 전쟁 게임 시뮬레이션의 문제는 BM(819C) 및 EPL(811B)에 의해 표시될 수 있는 주제를 포함한다. EPL은 실생활과 지출 비용에 영향을 미치는 시뮬레이션에 도움이 될 수 있는 유용한 정보를 차단할 수 있다. BM(819C)은 주제를 표시하고 그것을 MNSP(9) 관계자에 보고했을 수 도 있다. 따라서 적절한 자격을 갖춘 군대/ 조직은 PIP(802C)를 통해 LOM 세션을 인증하여 중단 없이 LOM을 통해 민감한 주제를 처리 할 수 있도록 하고 방해하거나 관계자에 보고 할 수 있다. 이러한 정보가 병력 번호와 위치와 같이 분류될 수 있기 때문에 인증된 세션은 BM(819C)와 EPL(811C)를 완전히 차단하는 오버라이드를 가능하게 하여 그러한 민감한 정보가 LOM을 외부 플랫폼이나 MNSP(9)와 같은 단체로 보내지 않는다. PIP(802C)를 통해 이 전쟁 게임을 실행하는 공인된 군인은 업그레이드된 / 특수화된 암호화 및 정보 격리가 있는 사용자 지정된 LOM 인스턴스를 사용한다. 여기에는 민감한 군사 정보가 공용 클라우드 저장에 들어 가지 않고 군사 승인된 시설 내에 남아 있지 않도록 하기 위해 사용자 지정 현장 저장 솔루션이 포함될 수 있다. 따라서 그러한 안전하게 보관된 정보는 LOM의 내부 심의(877A)가 제안된 전쟁 게임을 시뮬레이션 할 수 있게 한다.

[0359] 도 162 및 163은 LOM이 사람에 대한 식별 가능한 세부 사항을 밝히는 것과 같은 조사 저널리즘 작업을 수행하는 것을 보여준다. 이 사용 사례의 예는 가명 나카모토 사토시가 알려진 비트코인의 제작자를 둘러싼 수수께끼를 따른다. 비트코인 커뮤니티는 많은 잡지 및 조사 저널리스트와 함께 자신의 정체성을 밝히기 위해 많은 노력을 기울였다. 그러나 LOM은 자동화되고 철저한 방법으로 조사 노력을 극대화 할 수 있다. LOM은 초기 쿼리에 정확하게 응답할 수 있어야 하는 저널리즘 퍼즐의 특정 부분에 직면 할 수 있다. 따라서 LOM은 맞춤 정보 요청을 ARM(805B)에 전달할 수 있다. 이 요청은 CKR(806)에 정보를 통합한다. 상황화(879D) LOM은 질문(878B)이 객관적이고 비 개인적인 방식으로 제시되기 때문에 문제에 대한 정교한 의견을 형성할 수 있도록 특정 및 격리된 사용 케이스에 대한 추가 세부 사항을 요구하지 않는다. LOM은 알지 못하거나 LOM이 '잔인하게 솔직한' '성격'을 가졌기 때문에 확실치 않다고 응답하는 것을 '부끄러워'하지 않는다. 그러므로 하위 결론(878E)과 같이 사토시의 진정한 정체성을 드러내는데 필연적인 증거가 있음을 알 수 있다. ARM(805B)이 사토시에 올바르게 기여한 것으로 알려진 모든 전자 메일 및 채팅 로그를 검색할 때 문체분석(808B)이 수행되어 사토시의 실제 신원을 확인하고 정의한다. 그러므로 조사 저널리즘 과제에 관해 LOM이 알고 있는 모든 것이 결론(879C)으로 제시된다.

[0360] 도 164 및 165는 LOM에서 기록 유효성 검사를 수행하는 것을 보여준다. LOM은 화자의 체인의 확증을 통해 역사적인 문서의 진위 여부를 확인할 수 있다. '하디스'(문자 그대로 '아랍어'의 '뉴스')로 알려진 특정 기록 문서는 전송 된 뉴스를 뒷받침하는 사람들의 확인을 통해 본인의 저작자로 인증된 것으로 입증되었다. 하디스 문학은 원래 아랍어로 구어체 문맥에서 저장되고 이해 되었기 때문에 언어적 구성(812A) 모듈은 제 3 자 번역 알고리즘을 참조하여 문학을 모국어로 직접 이해한다. 문맥화(879D)와 함께 LOM은 질문(879B)이 객관적이고 비 개인적인 방식으로 제시되기 때문에 문제에 대한 정교한 의견을 형성할 수 있도록 특정 및 고립된 사용 케이스에 대한 추가 세부 사항을 요구하지 않는다. KCA(816D) UKF와 함께 클러스터된 정보는 화자 체인이 확인한 견적(하디스)의 유효성에 관한 확증적인 증거를 비교한다. 이 알고리즘은 원인을 밝힌 소스(즉, 하디스 화자라고 주장하는), 그러한 주장이 제기되었을 때, 증거를 부정하는 등의 신뢰성을 고려한다. LOM은 하디스의 인증 프로세스를 용이하게 하는 ARM에 의해 검색된 데이터로부터 CKR(806) 내에서 초과 개념을 구축한다. 하디스란 무엇인가?', '하디스의 변종은 무엇인가?', '인증의 가장 좋은 방법은 무엇인가?'와 같은 자체 질문이 있다. CKR(806)는 선천적인 고급 추론을 통해 강력한 기본 정의를 구축하고 LOM이 출력하는 모든 결론(879C)을 정당화 할 수 있다. 클러스터 빌딩(879C)을 통해 CKR(806)은 UKF 클러스터로 알려진 빌딩 블록을 '쌓는'방식을 통해 개념적 결론에 도달한다. 이러한 클러스터는 원인 소스, 의심되는 정보 생성 시간 등과 같은 대상 정보와 관련된 광범위한 메타 데이터를 포함한다.

[0361] **디지털 지향 언어 LAQIT**

[0362] 도 166은 LAQIT의 개념을 소개한다. LAQIT는 신뢰할 수 있는 대상의 네트워크 내에서 정보를 전송하는 효율적이고 안전한 방법이다. LAQIT는 가독성을 강조하고 보안에 중점을 둘 수 있는 다양한 모드를 제공한다. 선형, 원자 및 양자는 다양한 기능과 애플리케이션을 제공하는 서로 다른 정보 전송 모드이다. LAQIT는 보안 정보 전송의 궁극적인 형태이며, 가장 약한 링크는 마음의 사생활이다. 카운터 파티 위험은 효과적으로 암기 키가 수신자의 마음 속에만 저장되고 메시지가 해당 키의 구성에 따라 실시간으로(인간 기억을 사용하여) 해독되므로 실질적으로 제거된다. 키는 한 번만 전송하고 메모리에 저장되어야 하기 때문에 전화기를 꺼서 직접 키를 전달하는 것과 같은 격리된 암기 이벤트, 임시 암호화된 전자 메일 등을 통해 보다 정교한 개인 정보 보호 조치를 사용할 수 있다. 모든 보안 책임은 키의 기밀 범위 내에 있다. 암기하기 쉽기 때문에 모든 보안 책임의 대부분이 완화되었다. 블록(900A)은 LAQIT의 논리적으로 구조화된 구문 내에서 반복되고 재귀적인 적색, 주황색, 청색, 녹색 및 자색의 동일한 일관된 컬러 시퀀스를 도시한다. 블록(900B)은 영어 알파벳으로 번역하기 위해 재귀적으로 사

용되는 컬러 시퀀스를 더 설명한다. 알파벳의 '기본'레이어를 구성할 때 이 색상 시퀀스는 자주색 채널에서 짧고 다른 가중치와 함께 사용된다. 자주색 채널 내의 구문 정의를 위한 남은 공간은 잠재적인 향후 사용 및 확장을 위해 예약되어 있다. 901 단계는 복잡한 알고리즘이 LAQIT를 사용하여 로그 이벤트 및 상태 보고서를 보고함을 나타낸다. 이 시나리오에서는 암호화 옵션을 사용할 수 있는 반면 선택 사항에 따라 암호화가 사용되지 않는다. A1(902A) 단계는 상태 / 로그 보고서의 자동 생성을 나타낸다. A2 (903A) 단계는 상태 / 로그 리포트를 이동 가능한 텍스트 기반 LAQIT 구문으로 변환하는 것을 나타낸다. A3(904A) 단계는 디지털적으로 암호화된 (즉, VPN 12) 암호 해독 된(즉, 원시 HTTP) 채널을 통해 전송될 수 있는 구문적으로 안전하지 않은 정보의 전달을 나타낸다. 암호화된 채널이 선호되지만 필수는 아니다. A4(905A) 단계는 이동 가능한 텍스트 기반 구문의 판독이 가능한 LAQIT 시각적 구문(즉, 선형 모드)로의 변환을 나타낸다. 911 단계는 LAQIT가 비 컴퓨터 / 비 AI 정보 수신자를 위해 설계되고, 의도되고 최적화되기 때문에 대상 수신자를 사람으로 표현한다. 906 단계는 민감한 정보의 발신자가 인간임을 나타낸다. 그러한 인간은 정보 기관 또는 내부 고발자 계획을 대표 할 수 있다. 그러한 송신자 (906)는 임의의 트레이스가 영구 저장 장치에 남아있는 상태에서 그러한 키 (939)를 전송하도록 설계된 안전하고 일시적인 암호화 된 터널을 통해 사람 수신자 (911)에 LAQIT 암호화 키를 직접적으로 공개한다. 이상적으로 인간 수신자(911)는 키(939)를 메모리에 맡기고 해킹의 가능성을 제거하기 위해 키가 디지털 시스템에 가지고 있는 모든 저장 추적을 제거하는 것이 이상적이다. 이는 상대적으로 짧은 모양의 시퀀스를 기반으로 하기 때문에 인간의 암기에 최적화된 키(939)로 인해 가능하다. 단계 B1(902B)는 수신자(911)에 제출하기 위해 송신자(906)에 의해 입력되는 국부적으로 안전하지 않은 텍스트를 나타낸다. B2(903B) 단계는 그러한 텍스트(902B)를 이동 가능한 암호화된 텍스트 기반 LAQIT 구문으로 변환하는 것을 나타낸다. B3(904B) 단계는 디지털적으로 암호화된 (즉, VPN) 암호 해독된 (즉, 원시 HTTP) 채널을 통해 전송 될 수 있는 구문 보안 정보의 전송을 나타낸다. B4(905B) 단계는 시각적으로 암호화된 LAQIT 구문(즉, 암호화 레벨 8을 갖는 원자 모드)로의 데이터의 변환을 나타내며, 이후 인간 수신자(911)에 제공된다.

[0363] 도 167은 위치, 모양, 색상 및 사운드와 같은 정보 채널을 사용하여 정보를 전송할 때의 효율성을 비교하기 위해 사용 가능한 모든 언어(또는 정보 전송 모드)의 주요 유형을 보여준다. 가장 효과적이고 효율적이며 사용 가능한 언어는 대부분의 채널을 효과적으로 통합하고 활용할 수 있는 언어이다. 증가하는 인식 효과(Incremental Recognition Effect, IRE)(907)는 정보 전달 채널이다. 이는 정보가 완전히 전달되기 전에 완전한 형태의 정보 단위를 인식하는 효과를 특징으로 한다. 이것은 피험자가 단어나 문구를 완성하기 전에 끝내는 것과 비슷하다. LAQIT는 단어 사이의 전환을 표시함으로써 예측 지수의 이 효과를 통합한다. 숙련된 LAQIT 독자는 블록이 위치에 있지만 아직 도착하지 않은 상태에서 표시되는 단어를 형성하기 시작할 수 있다. 근위 인식 효과(PRE) (908)는 정보 전달 채널이다. 이는 정보 단위가 손상되거나 혼동되거나 변경되는 동안 완전한 형태의 정보 단위를 인식하는 효과를 특징으로 한다. 이것은 'character'와 'chracaetr'이라는 철자로 영어로 설명 될 수 있다. 유닛의 바깥 경계가 정의되었지만 (첫 번째 문자와 마지막 문자), 혼합 문자의 근접성은 여전히 단어 전체를 정의한다. 영어 쓰기(Written English)(912)에서는 일반적인 영어 텍스트가 글자의 위치, 글자의 모양 및 IRE(907)에 설명된 개별 글자와 함께 전체 단어 인식을 결합한다. 대화법 스피치(913)에서 평균 구두 대화는 단어의 위치(말한 순서), 피치의 빈도를 나타내는 모양 및 들리는 강조를 결합한다. 모스 부호(915)는 다양한 소리의 이진 위치로 구성된다. 정보 수신자의 예측인지는 IRE(907)을 가능하게 하지만 모스 부호가 정보를 스트림 정보로 점진적으로 전달하지는 않는다. 핸드 신호(915)에서, 손 움직임의 위치 및 형성 (형상)은 정보를 결정한다. 이것은 항공기에 신호를 보내는 것과 트럭을 멈추는 것에 이르기까지 다양하다. 예측 능력이 거의 없으므로 IRE(907)이나 PRE(908)도 없다. LAQIT(916)은 경쟁 언어 912 내지 915와 비교하여 대부분의 정보 채널을 활용할 수 있다. 즉, 적은 양의 매체(즉, 화면상의 공간)로 더 많은 정보를 더 짧은 시간에 전송할 수 있다. 이렇게 함으로써 용량 헤드 룸을 확보하여 강력한 암호화와 같은 복잡한 기능을 효과적으로 통합 할 수 있다. LAQIT 사운드 암호화(909)를 사용하면 LAQIT는 정보의 암호화를 위해 사운드의 정보 채널을 활용할 수 있다. 따라서 해독된 통신을 통해 정보를 전송할 수는 없지만 정보를 전송할 수 있는 것으로 간주된다.

[0364] 도 168 및 169는 LAQIT의 선형 모드를 보여 주며, 단순성, 사용 용이성, 높은 정보 밀도 및 암호화 부족으로 특징으로 한다. 블록(917)은 선형 모드의 '기본 렌더링'버전을 나타낸다. 포인트(918)는 암호화가 없음을 표시한다. 선형 모드는 원자 모드에서 암호화의 기초가 되는 모양 불명료화(941)에 대한 효율적인 공간 할당을 허용하지 않는다. 대신 선형 모드는 밀도 높은 정보 전송과 프레젠테이션 화면의 효율적인 사용을 위해 최적화되어 있다. 단어 분리(919)에서 이 도형의 색은 단어를 따라가는 문자를 나타내며 그 단어와 다음 단어 사이의 구분자 역할을 한다. 이것은 원자적 절차로 원자핵과 동일한 구문이다. 물음표, 느낌표, 마침표 및 쉼표를 나타내는 색상 코드는 모두 적용 가능하다. 싱글 뷰잉 존(920)은 기본 렌더링(917)이 진보된 렌더링(918)과 비교하여 더 큰 글자 및 더 적은 정보를 갖는 작은 시청 구역을 어떻게 통합 하는지를 보여준다. 이러한 고급 렌더링은 더블 뷰

잉 존(922)이 특징이다. 고급 렌더링에서는 LAQIT 리더가 속도 면에서 유지할 수 있기 때문에 픽셀 당 더 많은 활성 글자가 있다. 따라서 프레젠테이션 속도와 정보 밀도간에 트레이드 오프 딜레마가 있다. 음영 커버(921)는 관찰자의 주 초점이 관찰 영역 상에 있도록 수신 및 발신 문자를 둔하게 만든다. 덮개가 있음에도 불구하고 관찰자가 들어오는 단어를 예측하고 나가는 단어를 확인하는 기능을 제공할 수 있도록 부분적으로 투명하다. 이를 증가하는 인식 효과(Incremental Recognition Effect, IRE)(907)이라고도 한다. 고밀도 정보 전송(923)은 고급 렌더링(918)에서 각각의 문자가 더 작고 더 많은 글자가 동일한 공간에 어떻게 표시되는지를 보여 주므로 더 많은 정보가 픽셀 당 전달된다.

[0365] 도 170 및 171은 광범위한 암호화 수준을 사용할 수 있는 원자 모드의 특성을 보여준다. 베이스(924) 주 문자 참조는 어떤 문자가 정의되는지에 대한 일반 사항을 명시할 것이다. 빨강 베이스는 문자가 알파벳 참조(900B)에 따라 문자 A 내지 F 사이에 있음을 나타낸다. 유도가 단어의 철자를 추론하는 데 사용될 수 있기 때문에 베이스만 사용하여 단어를 읽을 수 있다(키커(925) 제외). 암호화를 가능하게 하기 위해 가능한 총 5 가지 형태로 존재할 수 있다. 키커(925)는 기본 색상과 동일한 색상 범위를 가지며 특정 문자를 정확하게 정의한다. 키커가 없다는 것은 정의를 나타낸다. 즉 키커가 없는 빨간색 바탕은 문자 A이다. 키커는 암호화를 가능하게 하기 위해 총 5 개의 가능한 모양들(935)에 존재할 수 있다. 읽기 방향(926)에서, 정보 전달 읽기는 궤도 링 상의 가장 위 사각형에서 시작한다. 읽기는 시계 방향으로 수행된다. 일단 궤도 링이 완료되면 독자는 다음 순차 궤도 링 (링 2)의 상단 사각형에서 계속된다. 입구 / 출구 포털(927)은 캐릭터의 생성 및 파괴 지점이다(기본임). 관련 궤도에 속한 새 캐릭터가 포털에서 나오고 시계 방향으로 슬라이드 된다. 원자 핵(928)은 그 단어 뒤에 나오는 성격을 정의한다. 일반적으로 이 단어는 공백으로, 이 단어가 표시된 후에 문장이 계속된다는 것을 나타낸다. 물음표, 느낌표, 마침표 및 쉼표를 나타내는 색상 코드는 모두 적용 가능하다. 또한 3 개의 궤도가 모두 최대 용량까지 채워지기 때문에 동일한 단어가 새로운 정보 상태로 계속 될지 여부를 나타낸다. 한 개의 궤도 링(929)이 채워지면 문자가 다음 (더 큰) 궤도 링 위로 오버플로우 된다. 한도는, 궤도 링 1은 7, 링 2는 15, 링 3은 20이다. 이렇게 하면 원자 내의 총 문자 수는 최대 42개까지 가능하다. 한도가 42 자에 도달하면 해당 단어는 42 개의 부분으로 나뉘며 핵은 다음 정보 상태가 현재 단어의 연속임을 나타낸다. 단어 탐색(930)에서 각 블록은 화면의 왼쪽에 전체 단어 (또는 분자 모드의 여러 단어)를 나타낸다. 단어가 표시되면 해당 블록이 바깥쪽으로 오른쪽으로 이동하고 해당 단어가 완료되면 블록이 후퇴한다. 탐색 블록의 색상 / 모양은 단어의 첫 글자의 기준과 동일한 색상 / 모양이다. 문장 내비게이션(931)에서는 각 블록이 단어 집단을 나타낸다. 클러스터는 단어 탐색 창에 들어갈 수 있는 단어의 최대 크기이다. 문장 네비게이션 블록이 자체 또는 많은 블록 중 마지막 블록인 경우 최대 용량보다 적은 수의 단어 클러스터를 나타내는 경향이 있다. 원자 상태 생성(Atomic State Creation)(932)는 증가하는 인식 효과(IRE)(907)를 유도하는 전환이다. 이러한 전환으로 베이스(924)는 입구/출구 포털(927)에서 나오고 키커(925)는 숨겨지고 시계 방향으로 움직여 자신의 위치를 차지한다. 이 전환 과정에서 LAQIT의 숙련된 독자는 IRE(907)로 인해 키커들(925)이 공개되기 전에 부분적으로 또는 전체 단어를 예측할 수 있다. 이것은 대부분의 검색 엔진의 자동 완성 기능과 유사하며 초기 정보 묶음이 있는 시퀀스의 남은 양을 추정한다. 원자 상태 확장 933은 근위 인식 효과(PRE)(908)를 유도하는 전환이다. 일단 베이스(924)가 그들의 위치에 도달하면, 이들은 정보 상태 프리젠테이션의 '확장' 시퀀스에서 바깥쪽으로 이동한다. 이것은 키커들(925)을 보여 주므로 정보 상태의 특정 정의를 나타낼 수 있다. 숙련된 LAQIT 독자는 점차적으로 단어를 작성하기 위해 각 문자를 점차적으로 스크롤 할 필요는 없지만 전체 구조 보고 PRE(908)로 인한 단어의 의미를 즉시 인식한다. 원자 상태 파괴(934)는 증가하는 인식 효과(IRE)(907)를 유도하는 전환이다. 이 단계에서 베이스(924)는 키커들(925)을 다시 덮기 위해 (확장 순서(933)를 뒤집은) 철회했다. 그들은 이제 입구 / 출구 포털에 도달하기 위해 시계 방향으로 슬라이드 한다. 정보 상태의 고속 렌더링에서 LAQIT의 숙련된 독자는 단어의 인식을 완료하기 위해 파괴 전이를 활용할 수 있다. 확장된 원자 상태(키커들이 보여주는)를 볼 수 있는 기회가 매우 좁은 경우(1 초의 몇 분지) 유용하다.

[0366] 도 172 내지 174는 원자 모드의 암호화 기능에 대한 개요를 보여준다. LAQIT는 정보를 전송하는 효율적이고 밀집된 수단을 제공하기 때문에 암호화 구현을 제공하기에 충분한 정보 대역폭 여유 공간이 있다. 이 구문 암호화는 의도된 정보 수신자가 암기된 키로 실시간으로 정보를 해독해야 한다는 점에서 고전적인 사이버 보안 암호화와 다르다. 이렇게 하면 동작 중인 데이터, 휴면 상태의 데이터 및 사용 중인 데이터가 악의적인 사람과 권한이 없는 사람이 읽고 이해하는 위험이 완화된다. 암호화 복잡성은 9 가지 표준화 수준(940)에 따라 다르며, 가독성과 보안 강도 간에 균형을 이룬다. 모양 불명료화(941)(레벨 1 내지 9)에서는 표준 사각형이 시각적으로 다른 5 개의 모양으로 대체되었다. 구문 내에서 모양의 차이는 위조(가짜) 문자가 원자 프로필의 전략적 지점에 삽입되도록 한다. 위조 문자는 메시지의 진실되고 의도된 의미를 모호하게 만든다. 문자가 실제인지 여부를 확인하는 것은 안전하고 일시적으로 전송된 암호 해독 키를 통해 수행된다. 문자가 키와 호환되면 단어의 계산에 포함된

다. 키 비호환성이 있는 경우 계산 내에서 무시해야 한다. 방향 전환 본드(942)(레벨 4 내지 9)에서는 본드가 두 글자를 연결하고 읽기 흐름을 변경한다. 일반적인 시계 방향 읽기 패턴으로 시작하는 동안 합법적 / 비 단서 문자로 출발 (시작) 및 종료 (끝)하는 본드가 발생하면 종료 문자에서 재개하기 위해 읽기 패턴을 전환 할 것이다. 방사성 원소(943)(레벨 7 내지 9)를 사용하면 문자가 위조인지 아닌지에 대한 평가를 역으로 할 수 있는 요소가 '덜컹거리는' 요소가 될 수 있다. 모양(935)은 암호화에 사용할 수 있는 모양을 보여준다. 삼각형, 원형, 정사각형, 5 각형 및 사다리꼴로 구성된다. 중심 요소(936)는 단어 뒤의 문자를 정의하는 궤도의 중심 요소를 보여준다. 이러한 요소는 빨간색으로 마침표를 나타내며 쉼표를 나타내는 오렌지색, 공간을 나타내는 파란색, 느낌표를 나타내는 녹색 및 느낌표를 나타내는 분홍색이다. 암호화 예제(937)는 암호화 레벨 1 내지 9에 적용할 수 있는 모양 불명료화(941)를 보여준다. 중심 요소(936)는 궤도의 중심에 표시되고 위조 문자(938)는 모양 불명료화(941)의 주요 암호화 수단이다. 왼쪽 위조는 원형-사각형 시퀀스를 가지고 있다. 오른쪽 위조는 사각형-삼각형 시퀀스를 가진다. 이 두 시퀀스는 모두 암호화 키(939)에 없으므로 판독기는 정보 상태의 의미를 계산할 때 위조로 인식하여 건너 뛴다.

[0367] 도 175 및 176은 방향 전환 본드(942)의 메커니즘을 도시한다. 암호화 예 (944)는 방향 전환 본드(942 및 945)를 나타낸다. 이들은 방향 전환 본드에 관한 '참여 규칙들'이다:

[0368] 1) 본드에 이르면 디폴트되고 따라서 시계 방향의 일상적인 행동은 포기된다.

[0369] 2) 경로를 따라갈 때: 시작 문자, 경로가 시작되는 문자는 시퀀스의 일부로 계산된다.

[0370] 3) 경로를 따라 가면: 경로가 끝나는 문자인 종료 문자는 서열의 일부로 간주된다.

[0371] 4) 통로는 한 번만 따라갈 수 있다.

[0372] 5) 문자의 특정 사례는 한 번만 계산 될 수 있다.

[0373] 6) 출발와 종료 문자가 모두가 위조가 아닌 경우에만 통로를 따라야 한다.

[0374] 방향 전환 본드(945)를 사용하면 본드는 '출발' 문자에서 시작하여 '종료' 문자로 끝난다. 그 중 하나는 위조일 수도 있고 그렇지 않을 수도 있다. 이들 중 어느 것도 위조가 아니면, 본드가 판독 방향 및 위치를 변경한다. 둘 중 하나 또는 둘 다 위조라면 전체 본드를 무시해야 한다. 그렇지 않으면 메시지가 올바르게 해독되지 않는다. 각각의 개별 본드는 올바른 방향으로 읽혀 지지만 그 순서는 명시적으로 기술되어 있지 않으며 정보 상태의 현재 읽기 위치와 위조 구성에 따라 유도되어야 한다. 위조 문자들(946)은 이 두 가지 위조 문자로 인해 암호 해독이 더욱 복잡해짐에 따라 무차별 대입 공격에 내성이 있음을 보여준다. 이는 모양 불명료화와 방향 전환 본드의 결합이 무차별 공격자에게 기하 급수적으로 더 어려운 작업으로 이어지기 때문이다. 본드 키 정의(947)는: 정보 상태 읽기에서 본드를 따라야 하는 경우 암호화 키에 구체적으로 정의되어 있는지 여부에 따라 다르다. 잠재적 정의는 단일 결합, 이중 결합 및 삼중 결합이다. 본드 키(947)를 알지 못하기 때문에 방향 전환 본드를 잘못 읽는 잠재적 사례 시나리오가 부정확한 해석(949)에 나와 있다. 이러한 부정확한 해석(949)은 'RDTNBAIB'라는 메시지로 연결되는 반면, 올바른 해석(948)의 진정한 메시지는 'RABBIT'이다. 모양 불명료화(941)의 복잡성을 활용하여 기하 급수적으로보다 안전한 메시지를 기하 급수적으로 생성하기 때문에 방향 전환 본드(945)를 잘못 해석하는 여러가지 잠재적인 방법들이 있다. 올바른 해석(948)에 나와있는 것처럼 진정한 메시지를 해석하는 올바른 방법은 하나뿐이다.

[0375] 도 177 및 178은 방사성 원소(943)의 메커니즘을 설명한다. 암호화 예제(950)는 방사성 원소(943과 951)를 보여준다. 이들은 방사성 원소에 관한 '참여 규칙'이다:

[0376] 1) 방사성 원소는 정보 상태의 확장 단계에서 불안정하거나 진동하는 것으로 인식된다.

[0377] 2) 방사성 원소는 방사능 활성이거나 휴면 상태 일 수 있다.

[0378] 3) 활동적인 방사성 원소는 위조가 역전된 상태임을 나타낸다. 즉, 모양 구성이 위조로 나타내는 경우, 거짓 긍정이며 실제로는 위조를 카운트하지 않고, 진짜 문자를 카운트한다. 모양 구성이 진짜임을 나타낼 경우, 거짓 긍정이며 실제로는 위조를 카운트하고, 진짜 문자를 카운트 하지 않는다.

[0379] 4) 휴면 방사능 원소는 그것이 위조 또는 진짜 문자의 상태가 영향을 받지 않음을 나타낸다.

[0380] 5) 방사성 원소의 집단은 궤도 링 안의 연속적인 방사성 존재에 의해 정의된다. 방사성 원소가 (특정 궤도 링 내에서) 서로 이웃 할 때, 그들은 하나의 군집을 정의한다. 방사성 요소의 이웃이 비 방사성인 경우 이 값이 클러스터의 상한선이다.

[0381] 6) 키는 활성 및 휴면중인 클러스터를 정의한다. 즉, 키가 이중 클러스터를 나타내면 모든 이중 클러스터는 방사성이며 모든 단일 및 트리플 클러스터는 휴면 상태이다.

[0382] 방사성 요소(950)는 문자(또는 요소)가 정보 표현의 확장 단계에서 격렬하게 흔들리는 경우 방사능으로 간주되는 방법을 보여준다. 암호화 수준의 분류로 인해 방사성 원소를 포함하는 원자는 항상 원자 간 결합을 갖는다. 방사능 원소는 글자의 분류가 위조인지 아닌지에 따라 변경되기 때문에 보안 난독화는 기하 급수적으로 증가한다. 이중 클러스터(952)는 시퀀스에 두 개의 방사성 요소가 있고 동일한 궤도에서 클러스터(더블)로 계산되는 이유를 보여준다. 활성 또는 휴면 상태로 처리되는지 여부는 암호화 키(954)에 의해 정의된다. 단일 클러스터(953)에서 두 이웃은 비 방사성이므로 클러스터의 범위가 정의된다. 키가 이중 클러스터를 유효하다고 지정하기 때문에 이 원소(953)는 처음에는 방사성이 아닌 경우 처리된다. 이중 클러스터 키 정의(954)에서 키는 이중 클러스터를 활성으로 정의하므로 다른 모든 크기의 클러스터는 메시지를 암호 해독하는 동안 휴면 상태로 간주된다. 잘못된 해석(956)은 해석자가 이중 클러스터(952)를 역순으로 처리하지 않았음을 보여준다(거짓 긍정). 이것은 956A 단계에서 올바른 대답은 위조가 아니더라도 문자를 반역으로 해석하는 해독 프로세스를 지시하는 능동적인 방사성 클러스터(키(954)에 의해 검증 됨)에 속하기 때문에 그것을 무시하는 것이다. 열쇠를 모르는 사람이라면 실제적인 의미에서 무차별 공격을 사용하여 모양 불명료화(941), 방향 전환 본드(942) 및 방사성 원소(943)이 동시에 사용되는 동안 모든 잠재적 조합을 추측할 수 없다. 잘못된 해석(956)은 키(954)가 없는 해석기가 어떻게 올바른 해석(955)에 따라야 할 방향 전환 본드(956B)를 사용하도록 오도 될 수 있는 방법을 보여준다. 이것은 'RABBIT'대신 'RADIT'의 메시지 결과와 완전히 다른 결과를 낳는다. 메시지를 올바르게 해독하는 방법에 대한 자세한 내용은 올바른 해석(955)에 설명되어 있다.

[0383] 그림 179는 암호화 및 스트리밍(959)이 활성화된 분자 모드를 보여준다. 은밀 사전 공격 저항(Corvert Dictionary Attack Resistance)(957)을 사용하면 메시지의 잘못된 암호 해독으로 인해 '붉은 청어(red herring)'대체 메시지가 표시된다. 이것은 악의적인 행위자에게 실제 메시지의 표지 역할을 하는 가짜 메시지를 수신하는 동안 메시지를 성공적으로 해독한 잘못된 인상을 주는 것이다. 분자당 여러 개의 활성 단어(958)을 사용하면 분자 절차 중에 단어가 병렬로 표시된다. 이것은 표면적 비율 당 정보를 증가 시키지만, 일관된 전환 속도로 보다 숙련된 독자가 필요하다. 단어 네비게이션은 현재 활성화된 네 단어가 있음을 나타낸다. 그러나 방향 전환 본드 불명료화로 인해 메시지의 단어는 부분적으로나 전체적으로 분자 내의 다른 원자에 존재한다. 바이너리 및 스트리밍 모드(959)는 스트리밍 모드를 나타내지만 일반적인 원자 구성에서는 읽기 모드가 바이너리이다. 바이너리 모드는 가운데 요소가 단어 뒤의 문자(예 : 물음표, 느낌표, 마침표, 공백 등)를 정의함을 나타낸다. 분자 모드는 또한 바이너리이다; 스트리밍 모드를 준수하는 암호화가 활성화된 경우는 예외이다. 스트리밍 모드는 궤도 내의 물음표와 같은 특수 문자를 참조한다. 이것은 암호화 된 분자 내에서 단어가 여러 원자에 걸쳐 존재하므로 특정 중심 요소가 특정 단어에만 독점적으로 존재할 수 없기 때문에 수행된다. 분자 결합(960)에서는 분자 정보 상태가 독점적인 암호화 기능이 아니지만 암호화 불명료화의 촉매제가 될 수 있다. 암호화의 3 가지 모드(모양 불명료화, 방향 전환 본드 및 방사성 요소)는 모두 점차적으로 증가하는 분자 환경에 배치될 때 보안 강도 면에서 기하 급수적으로 증가한다. 판독 방향 키(961)는 디폴트 판독 방향이 1 열은 좌측에서 우측으로, 2 열은 다시 좌측에서 우측으로 진행되는 동안 판독 방향이 암호화 키에 의해 대체 될 수 있음을 나타낸다. 이것은 의도된 메시지의 불명료화를 증가시키고 따라서 메시지 프라이버시 / 보안을 증가시킨다. 방향 전환 본드는 가장 우선 순위를 가지며 키에 정의된 방향 조차도(본드가 위조가 아닌 한) 대체한다.

[0384] **통합 노드를 부착하는 기본 연결 조화(BCHAIN; Base Connection Harmonization Attaching Integrated Nodes)를 이용한 유니버설 BCHAIN 모든 연결들(UBEC; Universal BCHAIN Everything Connections)의 요약**

[0385] 도 180은 BCHAIN 사용 어플리케이션(1003)을 포함하고 실행하는 BCHAIN 노드(1001)를 도시한다. 통신 게이트웨이(CG)(1000)는 BCHAIN 노드(1001)가 그 하드웨어 인터페이스와 상호 작용하여 다른 BCHAIN 노드(1001)와의 통신을 유도하는 주요 알고리즘이다. 노드 통계 조사(NSS)(1006)는 원격 노드 행동 패턴을 해석한다. 노드 이스케이프 인덱스(1006A)는 노드 이웃이 인식 노드의 부근을 벗어날 가능성을 추적한다. 높은 이스케이프 지수는 보다 혼란스러운 환경을 나타내며. 이에 대처하기 위해서는 개선된 전략이 필요하다.

[0386] 예: 고속도로에 있는 자동차의 스마트 폰은 노드 이스케이프 인덱스가 높다. 스타 벅스의 냉장고는 매우 낮은 노드 이스케이프 인덱스를 나타낸다.

[0387] 노드 포화 인덱스(1006B)는 감지 노드의 검출 범위 내의 노드의 양을 추적한다. 포화도 지수가 높으면 많은 노드가 있는 혼잡한 지역을 나타낸다. 이것은 공급 / 수요 트레이드 오프로 인해 성능에 긍정적이거나 부정적인 영향을 줄 수 있지만 밀도가 높은 노드 영역은 보다 안정적 / 예측 가능하고 따라서 혼란이 적을 것으로 예상된

다.

[0388] 예: 뉴욕시의 중심부에 있는 스타 벅스는 노드 포화 인덱스가 높다. 사막 한가운데 있는 텐트는 노드 포화 인덱스가 매우 낮다.

[0389] 노드 일관성 인덱스(1006C)는 인식 노드에 의해 해석되는 노드 서비스의 품질을 추적한다. 높은 노드 일관성 인덱스는 주변 노드가 가용성 가동 시간 및 성능의 일관성을 유지하는 경향이 있음을 나타낸다. 이중 용도를 사용하는 노드는 일관성 지수가 낮은 반면 BCHAIN 네트워크 전용 노드는 높은 값을 나타낸다.

[0390] 예: 회사 직원 컴퓨터와 같은 이중 목적을 가진 노드는 근무 시간 동안 사용 가능한 자원이 적고 점심 시간 동안 사용 가능한 리소스가 더 많기 때문에 일관성 인덱스가 낮다.

[0391] 노드 중복 인덱스(1006D)는 노드가 인식하는 것으로 중첩되는 노드의 양을 서로 추적한다. 중첩 및 채도 지수는 상관 관계가 있는 경향이 있지만, 중첩 인덱스는 이웃 간의 일반적인 중첩 정도를 나타내며 포화 지수는 물리적 경향만을 다룬다. 따라서 각 장치에서 무선 범위가 긴 높은 포화 인덱스는 높은 중첩 지수로 이어질 것이다.

[0392] 예: 디바이스는 고급 BCH (BCHAIN Optimized Microchip)가 설치된 BCHAIN 네트워크의 특정 섹터에 진입하기 시작하며. BCHAIN은 고급 빔 포밍 기술이 적용된 높은 이득 지향성 안테나를 갖추고 있다. 따라서, 중첩된 통신 구조를 갖는 노드로 인해 이들 섹터에서 중복 인덱스가 증가한다.

[0393] 도 181은 BCHAIN 프로토콜의 핵심 논리(1010)를 도시한다. 사용자 정의 체인 인식 모듈(CRM)(1022)은 이전에 노드에서 등록한 사용자 정의 체인(앱 체인들 또는 마이크로 체인들 일 수 있음)과 연결한다. 따라서 노드는 그러한 기능에 대한 읽기, 쓰기 및 / 또는 관리 능력에 대한 암호화 액세스 권한을 가진다. 이 모듈은 메타 체인 또는 마이크로 체인의 메타 체인 에뮬레이터의 앱체인의 섹션에서 업데이트가 감지되면 나머지 BCHAIN 프로토콜에 알린다. 컨텐츠 클레임 제공(CCD; Content Claim Delivery)(1026)은 유효화된 CCR(1018)을 수신 한 후 그 요청을 수행하기 위해 관련 CCF(1024)를 전송한다.

[0394] 도 182는 최적화된 전략 선택 알고리즘(OSSA; Optimized Strategy Selection Algorithm)(1042)을 통해 시스템에 의해 선호된 복잡한 전략을 하이브리드화하기 위해 창조성 모듈(18)을 사용함으로써 새로운 전략 배치(1054)를 동적으로 생성하는 전략 생성 모듈(SCM; 1046)을 관리하는 동적 전략 적응(DSA; Dynamic Strategy Adaptation)(1008)을 도시한다. 새로운 전략은 영역 혼돈 해석(FCI; Field Chaos Interpretation)(1048)에 제공된 입력에 따라 다양하다.

[0395] 도 183은 UBEC 플랫폼 인터페이스(UPI)하에서 그래픽 사용자 인터페이스(GUI; Graphical User Interface)에 의해 관리되는 다양한 경제적 성격들(1058, 1060, 1062 및 1064)을 갖는 암호화 디지털 경제 교환(CDEE; Cryptographic Digital Economic Exchange)(1056)을 도시한다. 성격 A(1058) 노드 자원은 소비하는 것(있는 경우)과 일치하는 경우에만 소비된다. 성격 A는 가벼운 양의 정보 전송을 통해 가벼운 검소한 소비자에게 이상적이다. VoIP 통화 (예: Skype) 및 우선 순위 정보 전송과 같은 실시간 스트리밍은 최소화된다. 성격 B(1060)는 이익 마진이 X보다 큰 한 가능한 한 많은 자원을 소비한다. (초과 작업 단위는 암호 해독, 화폐 통화, 귀금속 등 대체 통화로 거래 될 수 있음) 성격 B는 이익 동기를 위해 BCHAIN 네트워크의 인프라에 기여하기 위해 특별히 설정된 노드에 이상적이다. 따라서 이러한 노드는 일반적으로 배터리로 작동하는 장치와 달리 주전원에서 실행되는 영구적인 인프라 설치이며 강력한 컴퓨터 내부(무선 기능, CPU 강도, 하드 디스크 크기 등)가 있다. 예: 고정식 기기 등. 성격 C(1062)는 거래된 통화(암호화된 화폐, 화폐 통화, 귀금속 등)를 통해 거래 단위 비용을 지불하므로 노드 자원을 덜 소비하면서 콘텐츠를 사용할 수 있다. 성격 C는 대량의 정보 전송 소비자 나 BCHAIN 네트워크를 통해 이익을 얻을 수 있지만 기기의 리소스를 고갈시키지 않으려는 사람에게 이상적이다(예: 스마트 폰이 배터리를 빨리 소모하고 주머니에서 따뜻해짐). 성격 D(1064) 노드 자원은 컨텐츠 소비 또는 금전적 보상 여부와 관계없이 최대한 많이 소비되며 그에 대한 보상의 제약이 없다. 성격 D는 BCHAIN 네트워크의 장점에 가장 부합하는 사람에 의해 선택된다. (즉, BCHAIN 네트워크의 핵심 개발자는 노드를 구입하여 네트워크를 강화하고 컨텐츠를 소비하거나 돈을 벌지 못하게 할 수 있다.) 현재 작업 상태 해석(CWSI; Current Work Status Interpretation)(1066)은 작업 완료 크레딧과 관련하여 이 노드의 현재 잉여 또는 적자를 결정하기 위해 메타체인의 인프라 경제 섹션을 참조한다. 경제적 측면을 고려한 작업 시행(ECWI; Economic Inside Work Imposition)(1068)은 현행 근로 잉여 / 적자와 함께 선택된 경제적 특성을 고려하여 현재 더 많은 작업이 수행되어야 하는지 평가한다.

[0396] 도 184는 공생적 재귀 지능 향상(SRIA; Symbiotic Recursive Intelligence Advancement)을 보여주는데, 이것은 서로가 지능이 성장할 수 있게 하는 세 가지 알고리즘 간의 삼중 관계이다. LIZARD(16)은 코드의 목적을 이해함

으로써 알고리즘의 소스 코드를 개선할 수 있다. I^{2}GE(21)는 가상 프로그램 반복 세대를 에뮬레이션 할 수 있으므로 가장 강력한 프로그램 버전을 선택할 수 있다. BCHAIN 네트워크는 분산된 방식으로 복잡한 데이터가 많은 프로그램을 실행할 수 있는 무질서하게 연결된 노드의 방대한 네트워크이다.

부호의 설명

[0397] 9 MNSP

10 신뢰 플랫폼

12 VPN/엑스트라넷

16 LIZARD

17 인공의 보안 위협

18 창조성 모듈

19 음모 탐지

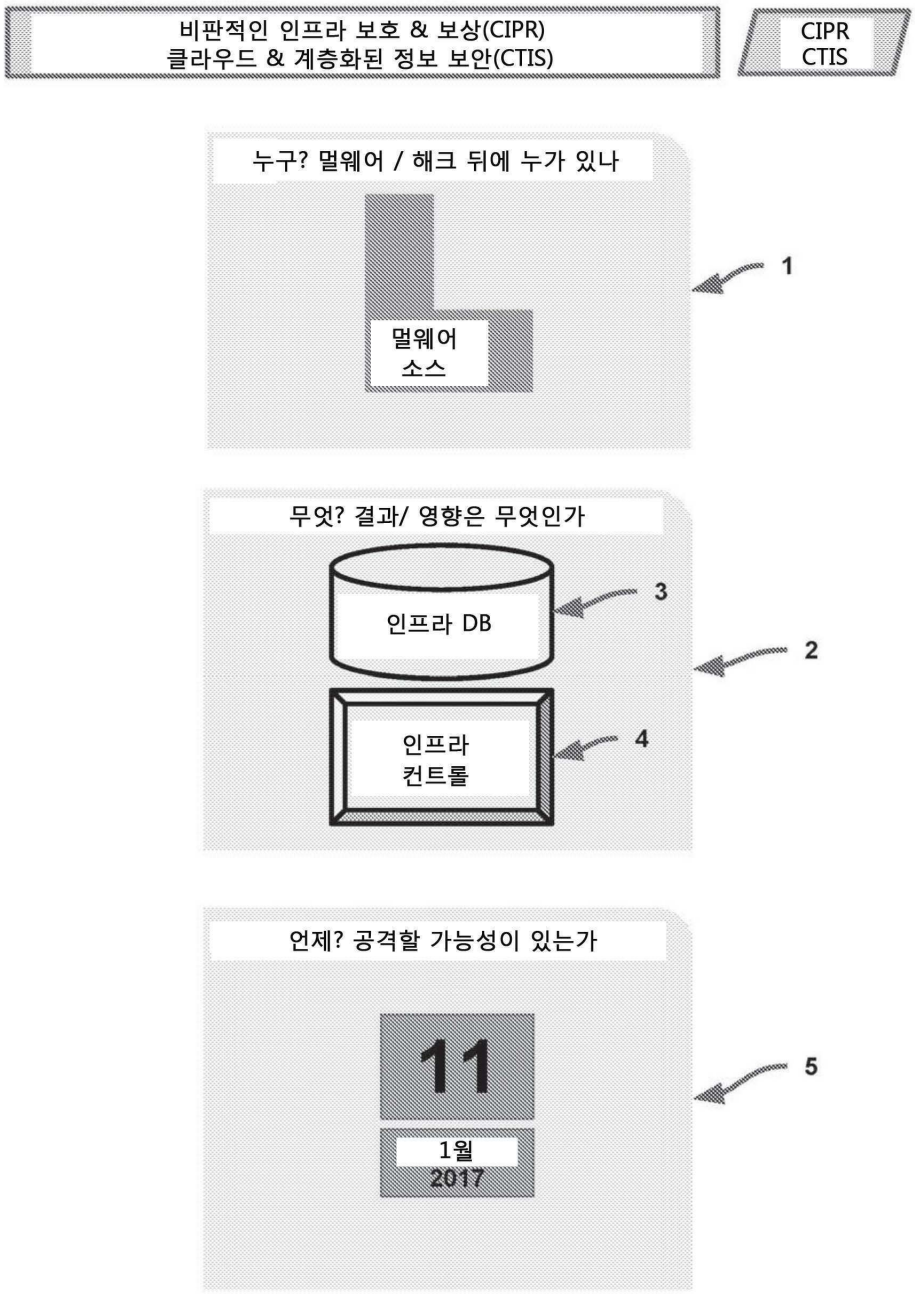
20 보안 행동

22 CTMP

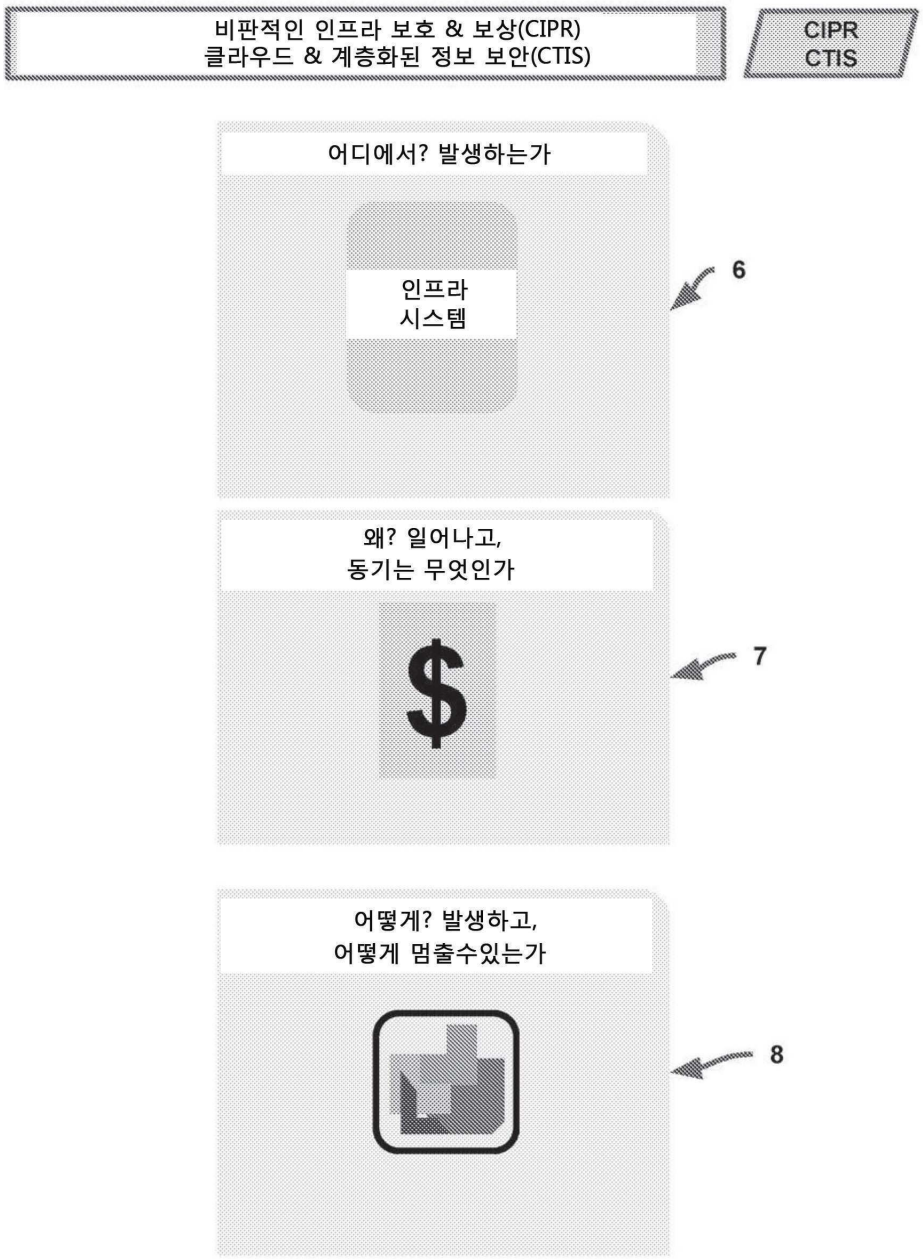
23 관리 콘솔

도면

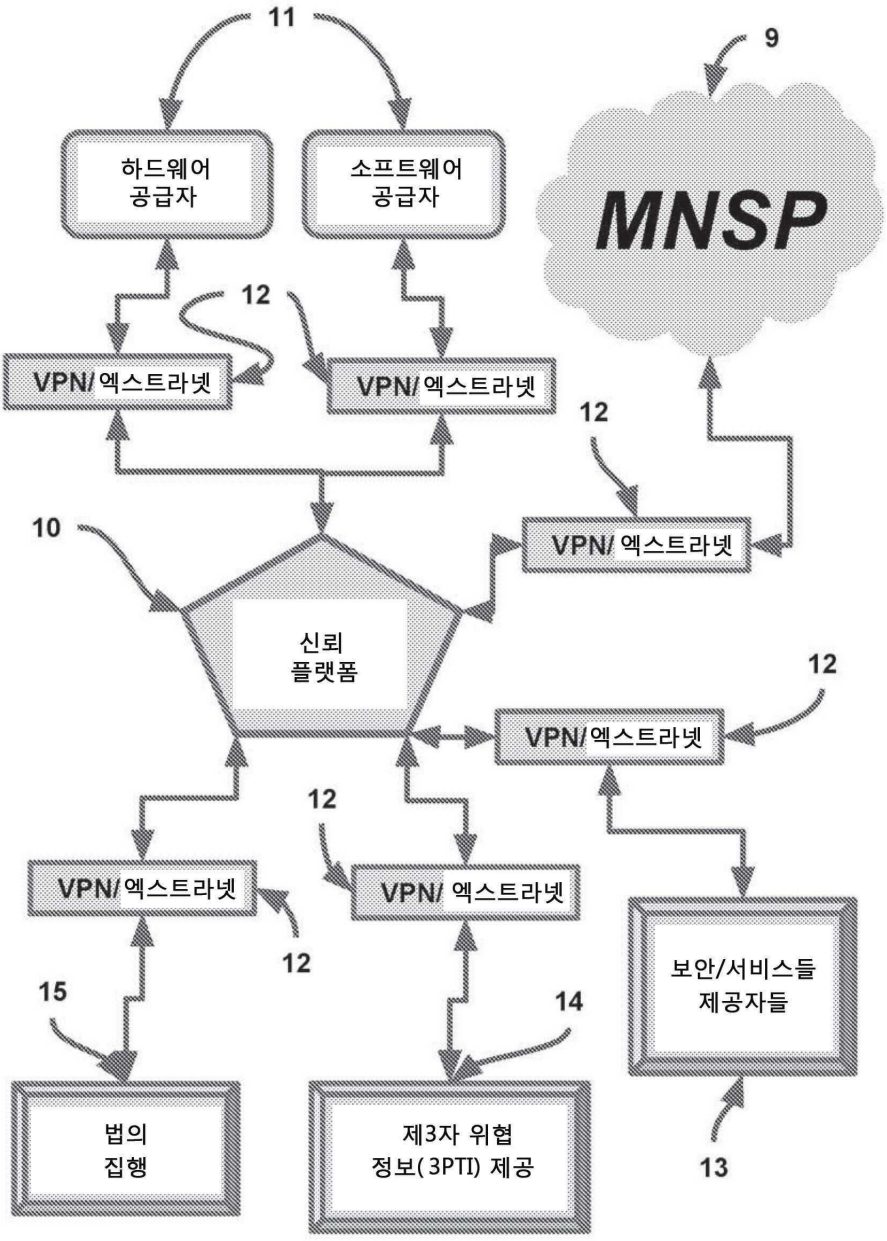
도면1


비판적인 인프라 보호 & 보상(CIPR)
클라우드 & 계층화된 정보 보안(CTIS)
CIPR
CTIS
누구? 멀웨어 / 해크 뒤에 누가 있나
멀웨어
소스
1
무엇? 결과/ 영향은 무엇인가
인프라 DB
3
2
인프라
컨트롤
4
언제? 공격할 가능성이 있는가
11
1월
2017
5

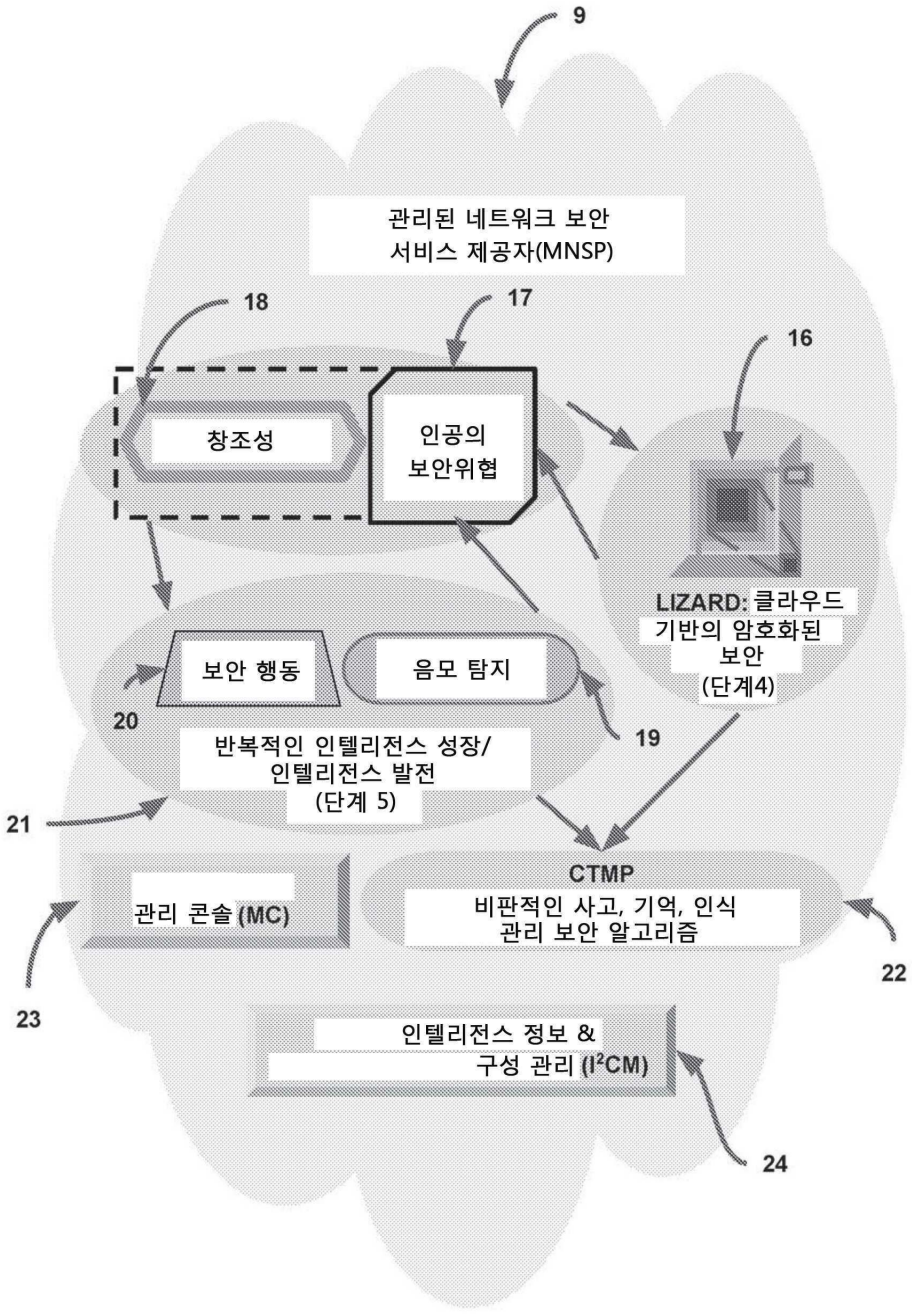
도면2


비판적인 인프라 보호 & 보상(CIPR)
클라우드 & 계층화된 정보 보안(CTIS)
CIPR
CTIS
어디에서? 발생하는가
인프라
시스템
6
왜? 일어나고,
동기는 무엇인가
$
7
어떻게? 발생하고,
어떻게 멈출수있는가
8

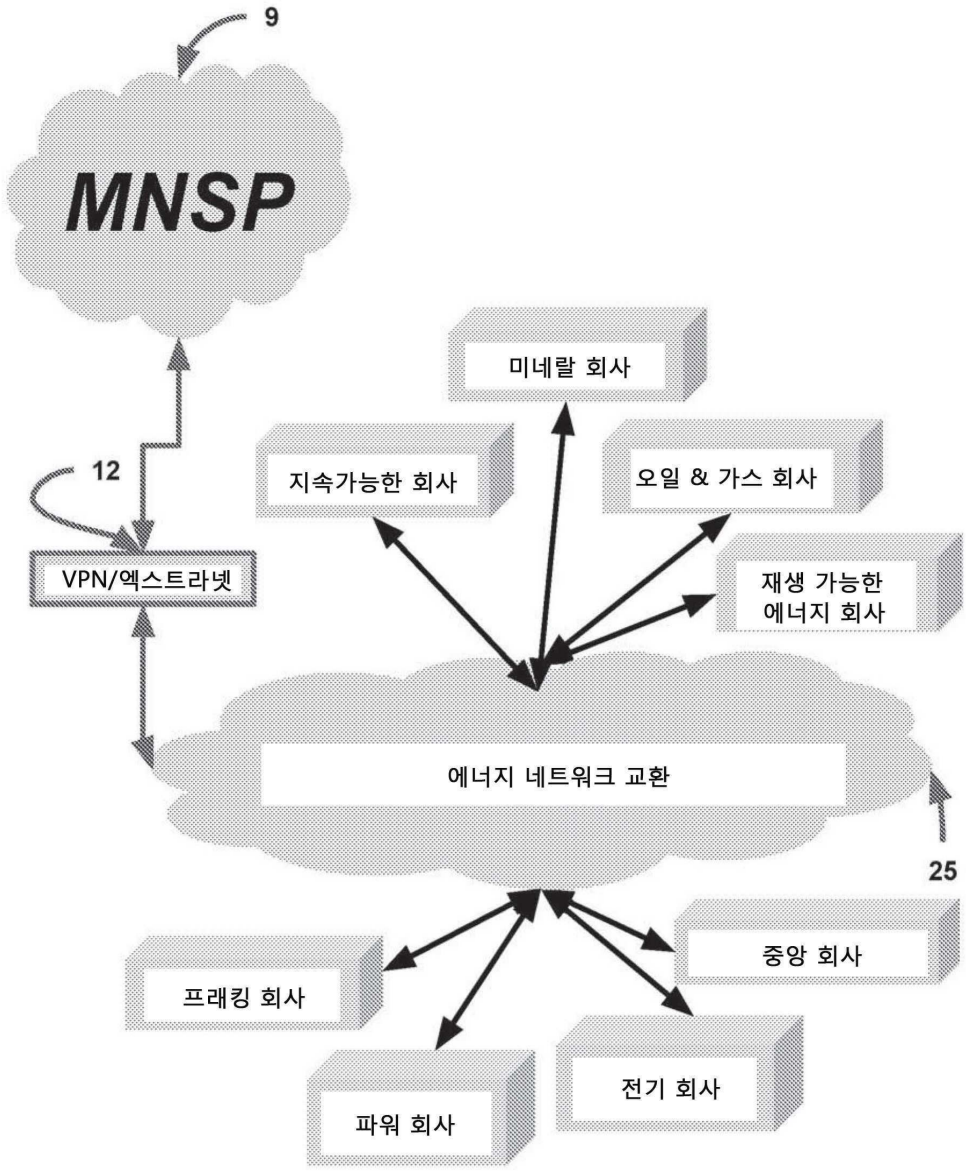
도면3


11
9
하드웨어
공급자
소프트웨어
공급자
MNSP
12
VPN/엑스트라넷
VPN/엑스트라넷
12
10
VPN/엑스트라넷
신뢰
플랫폼
12
VPN/엑스트라넷
12
12
VPN/엑스트라넷
VPN/엑스트라넷
보안/서비스들
제공자들
15
12
14
법의
집행
제3자 위협
정보(3PTI) 제공
13

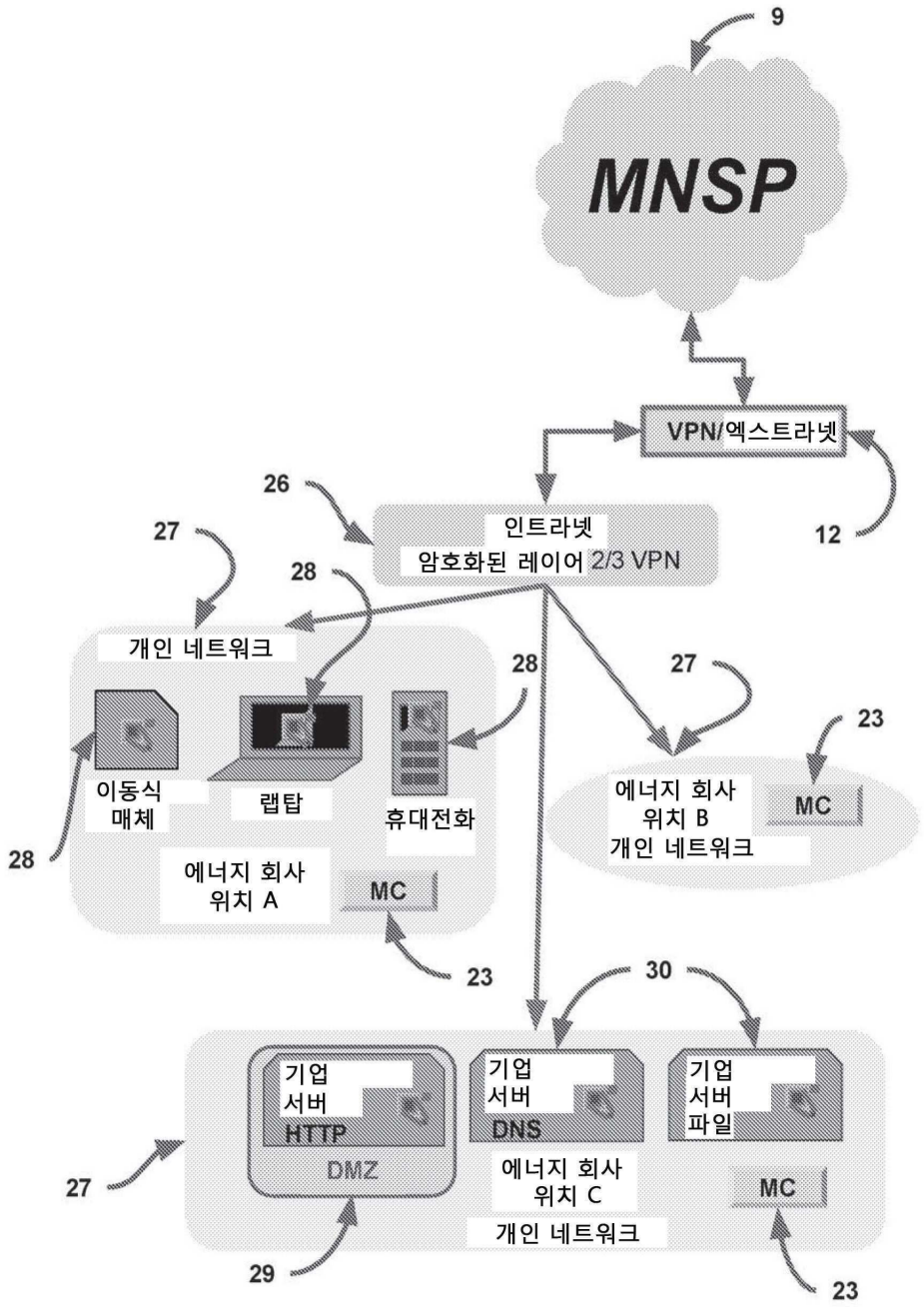
도면4


9
관리된 네트워크 보안
서비스 제공자(MNSP)
18
17
16
창조성
인공의
보안위협
LIZARD: 클라우드
기반의 암호화된
보안
(단계4)
보안 행동
음모 탐지
20
19
반복적인 인텔리전스 성장/
인텔리전스 발전
(단계 5)
21
CTMP
비판적인 사고, 기억, 인식
관리 보안 알고리즘
22
관리 콘솔 (MC)
23
인텔리전스 정보 &
구성 관리 (I²CM)
24

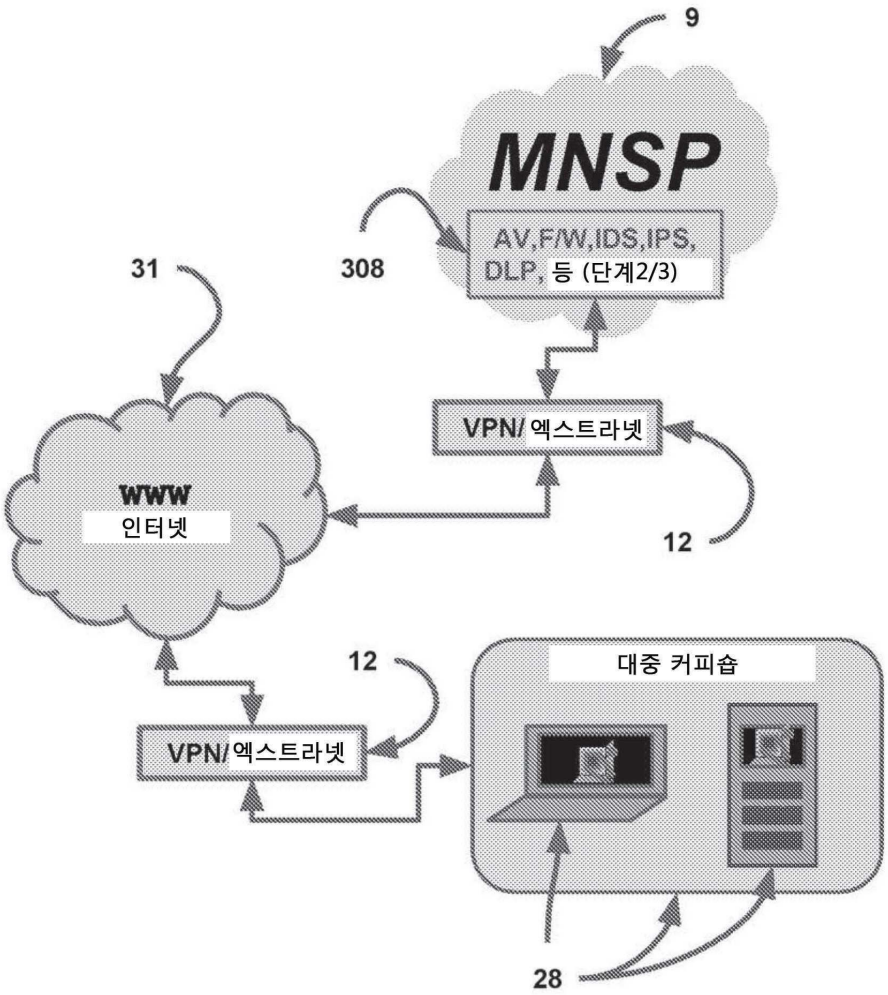
도면5


9
MNSP
12
VPN/엑스트라넷
미네랄 회사
지속가능한 회사
오일 & 가스 회사
재생 가능한
에너지 회사
에너지 네트워크 교환
25
중앙 회사
프래킹 회사
전기 회사
파워 회사

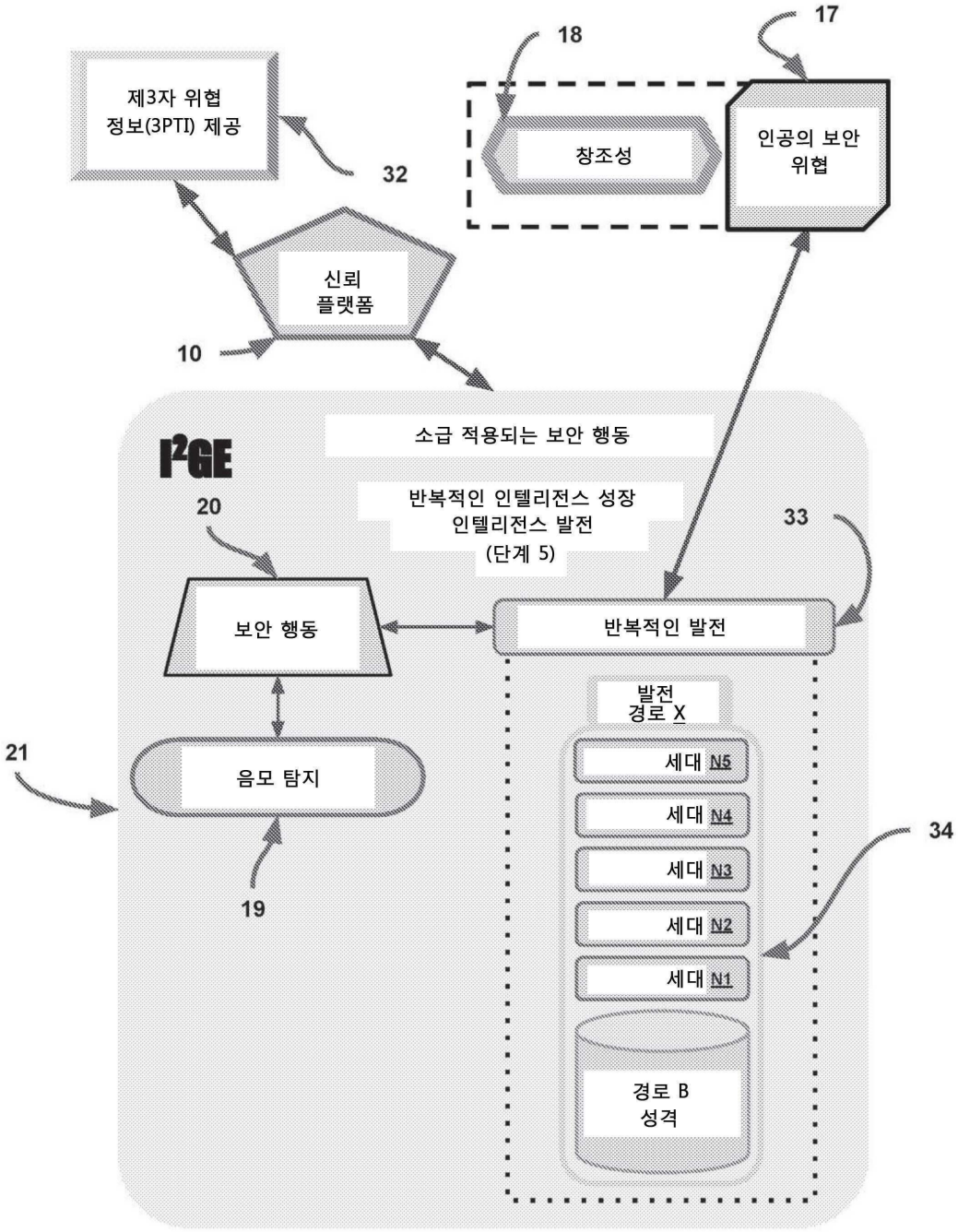
도면6


9
MNSP
VPN/엑스트라넷
12
26
인트라넷
암호화된 레이어 2/3 VPN
27
28
개인 네트워크
28
27
23
이동식
매체
랩탑
휴대전화
에너지 회사
위치 B
개인 네트워크
MC
28
에너지 회사
위치 A
MC
23
30
기업
서버
HTTP
DMZ
기업
서버
DNS
기업
서버
파일
27
에너지 회사
위치 C
개인 네트워크
MC
29
23

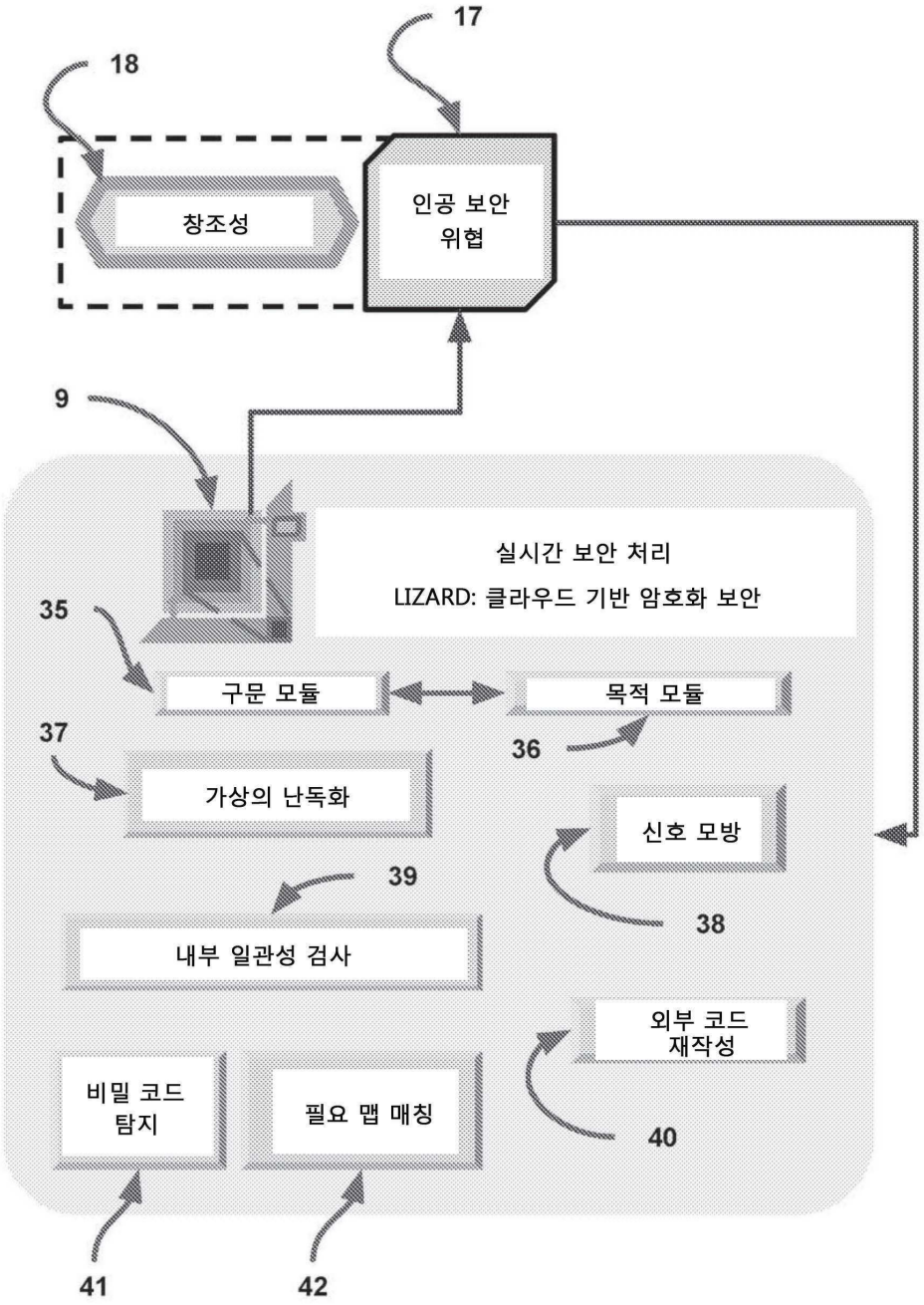
도면7


9
MNSP
AV,F/W,IDS,IPS,
DLP, 등 (단계2/3)
308
31
VPN/엑스트라넷
WWW
인터넷
12
12
대중 커피숍
VPN/엑스트라넷
28

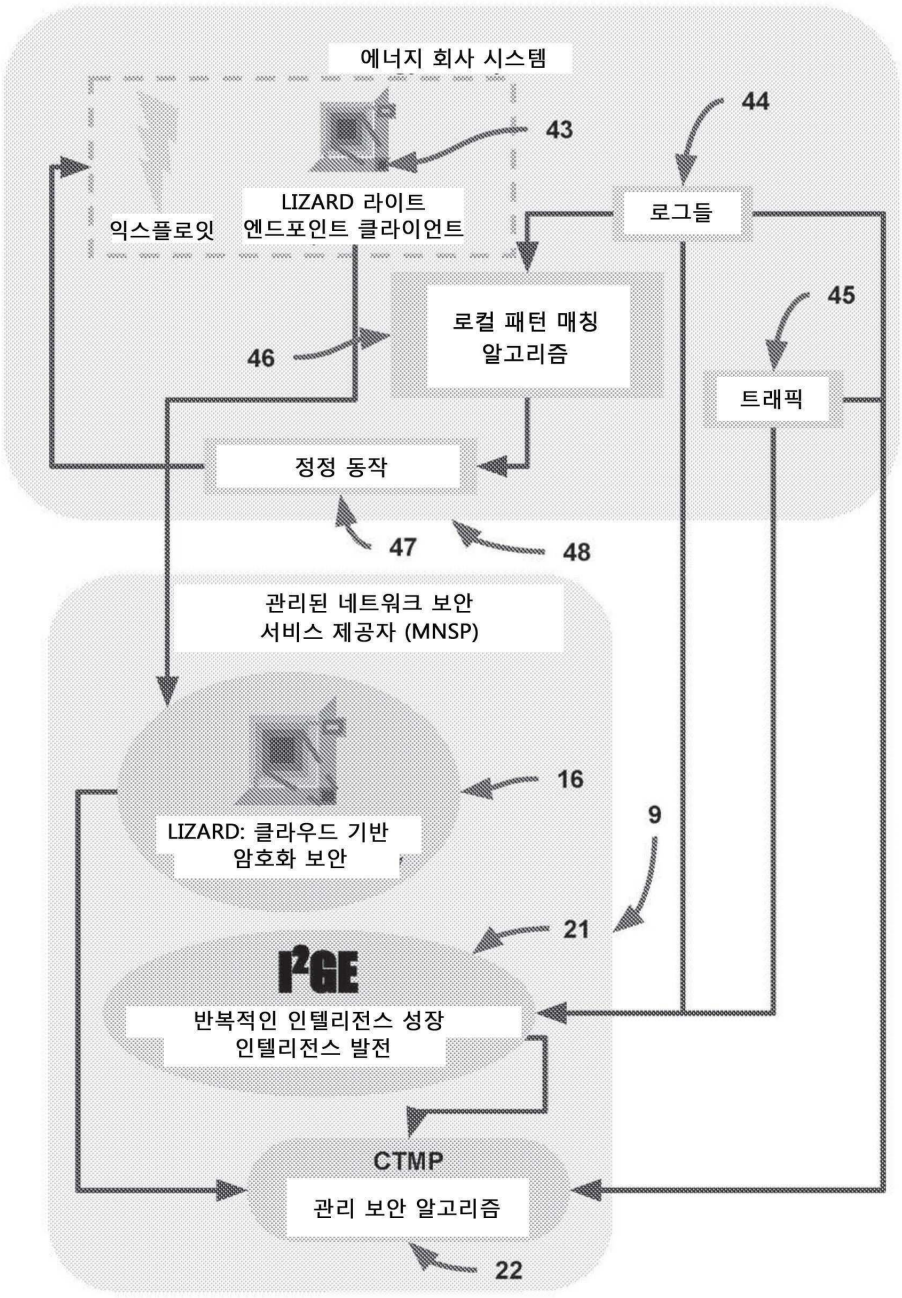
도면8


제3자 위협
정보(3PTI) 제공
32
18
17
창조성
인공의 보안
위협
신뢰
플랫폼
10
소급 적용되는 보안 행동
I²GE
반복적인 인텔리전스 성장
인텔리전스 발전
(단계 5)
20
33
보안 행동
반복적인 발전
발전
경로 X
21
음모 탐지
세대 N5
세대 N4
세대 N3
세대 N2
세대 N1
34
19
경로 B
성격

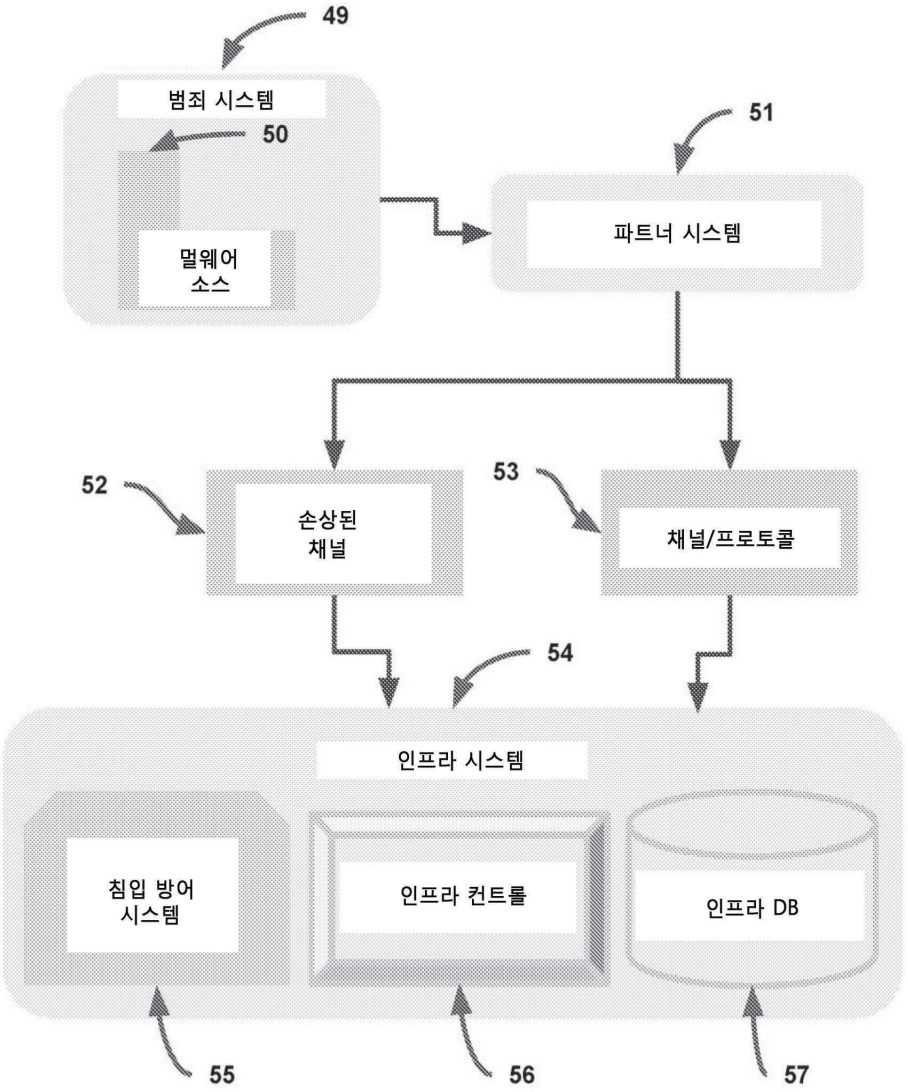
도면9


17
18
창조성
인공 보안
위협
9
실시간 보안 처리
LIZARD: 클라우드 기반 암호화 보안
35
구문 모듈
목적 모듈
37
36
가상의 난독화
신호 모방
39
38
내부 일관성 검사
외부 코드
재작성
비밀 코드
탐지
필요 맵 매칭
40
41
42

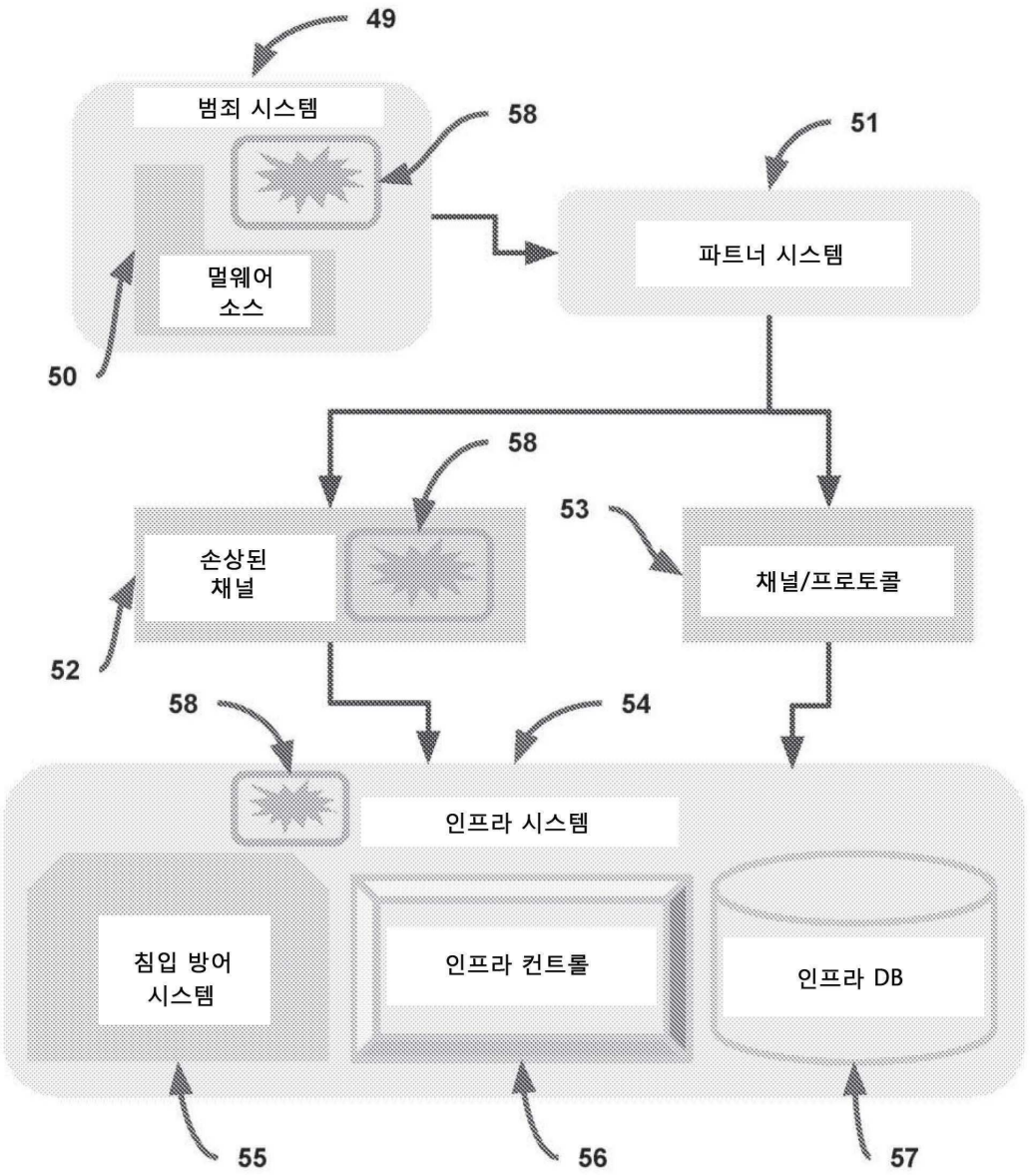
도면10


에너지 회사 시스템
43
44
익스플로잇
LIZARD 라이트
엔드포인트 클라이언트
로그들
45
46
로컬 패턴 매칭
알고리즘
트래픽
정정 동작
47
48
관리된 네트워크 보안
서비스 제공자 (MNSP)
16
LIZARD: 클라우드 기반
암호화 보안
9
21
반복적인 인텔리전스 성장
인텔리전스 발전
CTMP
관리 보안 알고리즘
22

도면11

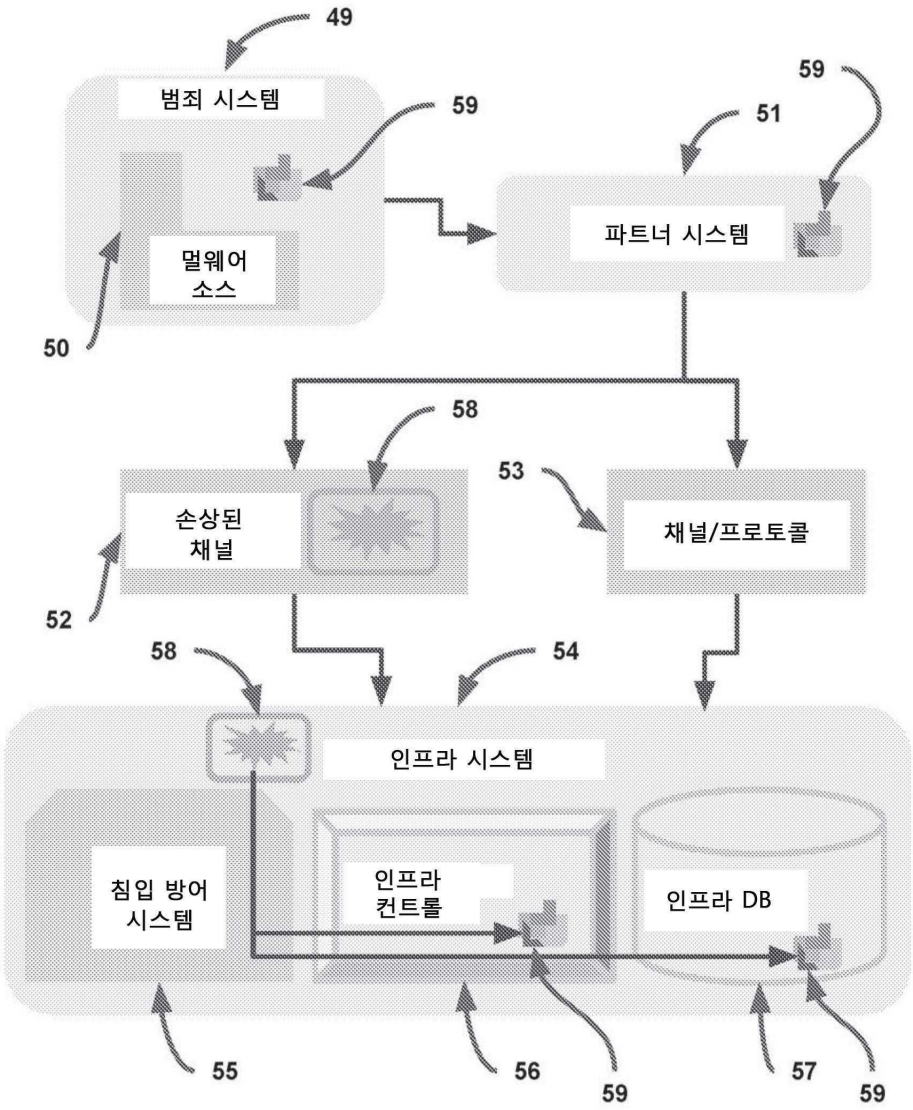

1단계: 초기의 시스템 침입
49
범죄 시스템
50
멀웨어
소스
51
파트너 시스템
52
손상된
채널
53
채널/프로토콜
54
인프라 시스템
침입 방어
시스템
인프라 컨트롤
인프라 DB
55
56
57

도면12

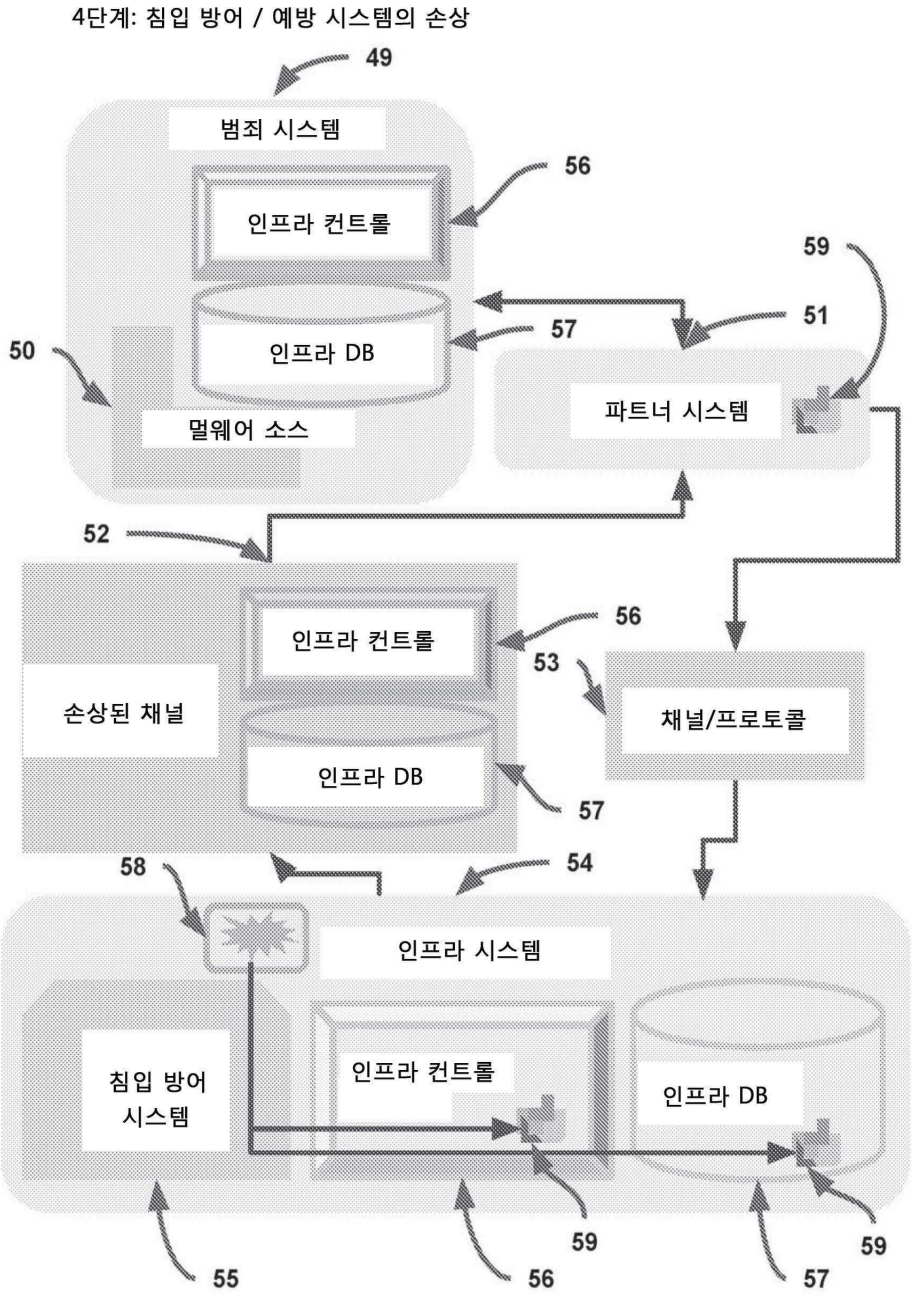

2단계: 초기의 트로이 목마 배치
49
범죄 시스템
58
51
멀웨어
소스
파트너 시스템
50
58
53
손상된
채널
채널/프로토콜
52
58
54
인프라 시스템
침입 방어
시스템
인프라 컨트롤
인프라 DB
55
56
57

도면13

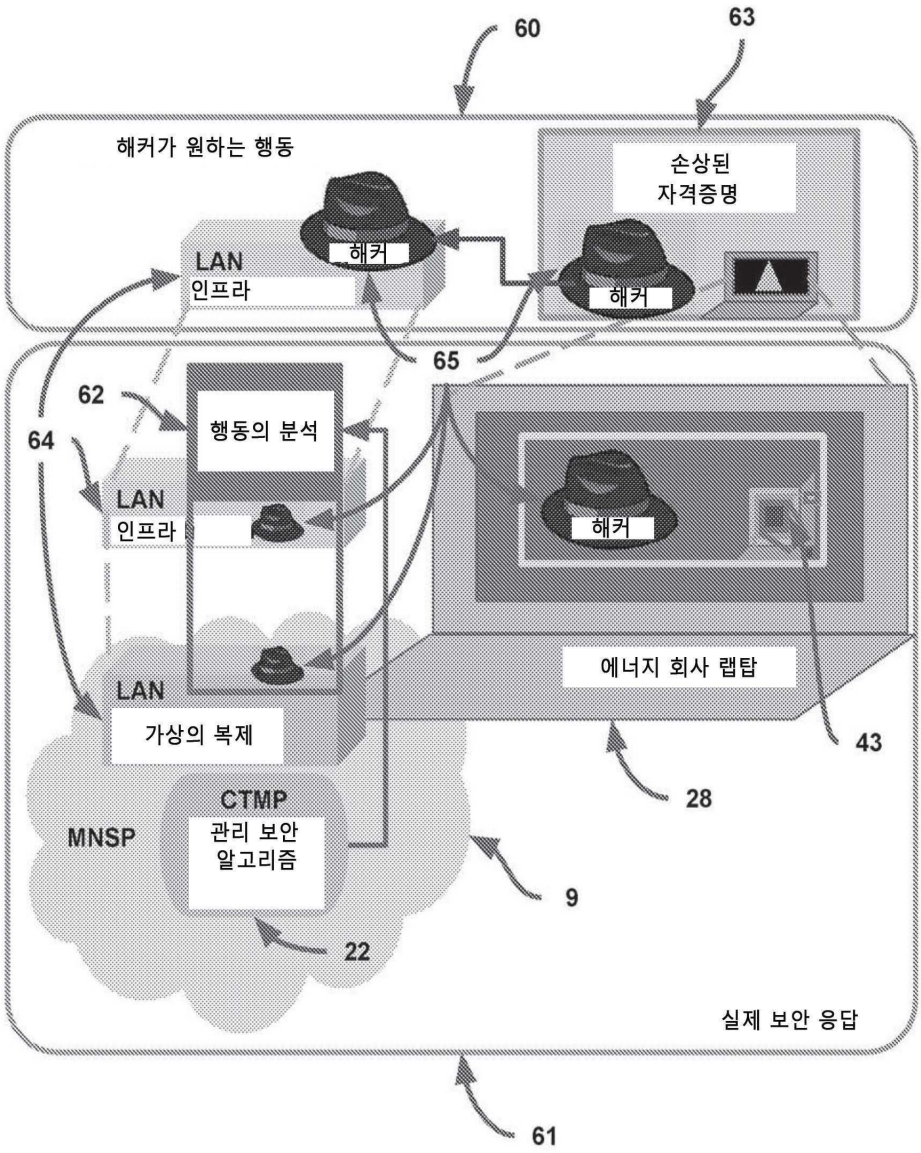
3단계: 고급 실행 멀웨어의 다운로드



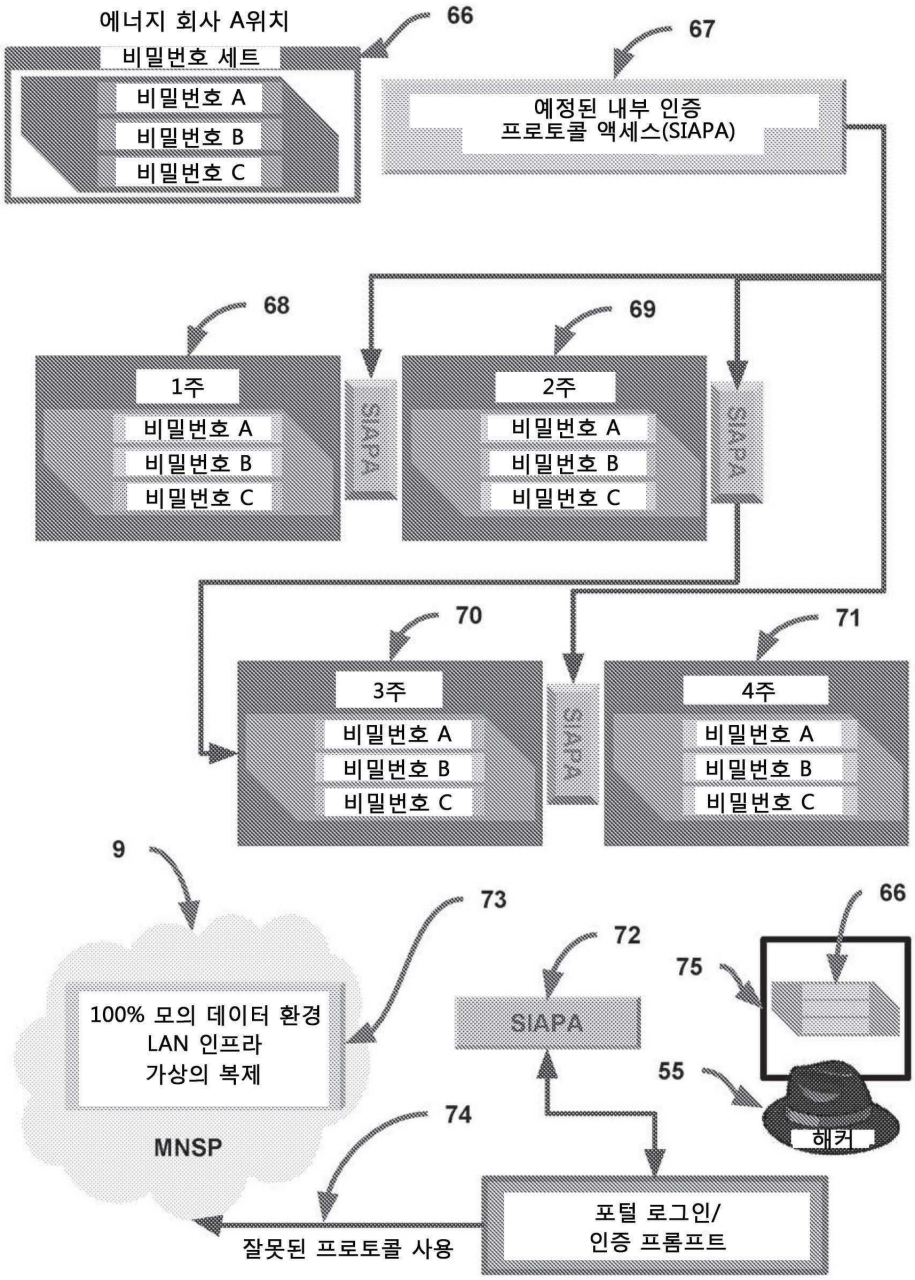
도면14


4단계: 침입 방어 / 예방 시스템의 손상
49
범죄 시스템
56
인프라 컨트롤
59
51
57
인프라 DB
50
멀웨어 소스
파트너 시스템
52
56
인프라 컨트롤
53
손상된 채널
채널/프로토콜
인프라 DB
57
58
54
인프라 시스템
침입 방어
시스템
인프라 컨트롤
인프라 DB
59
59
55
56
57

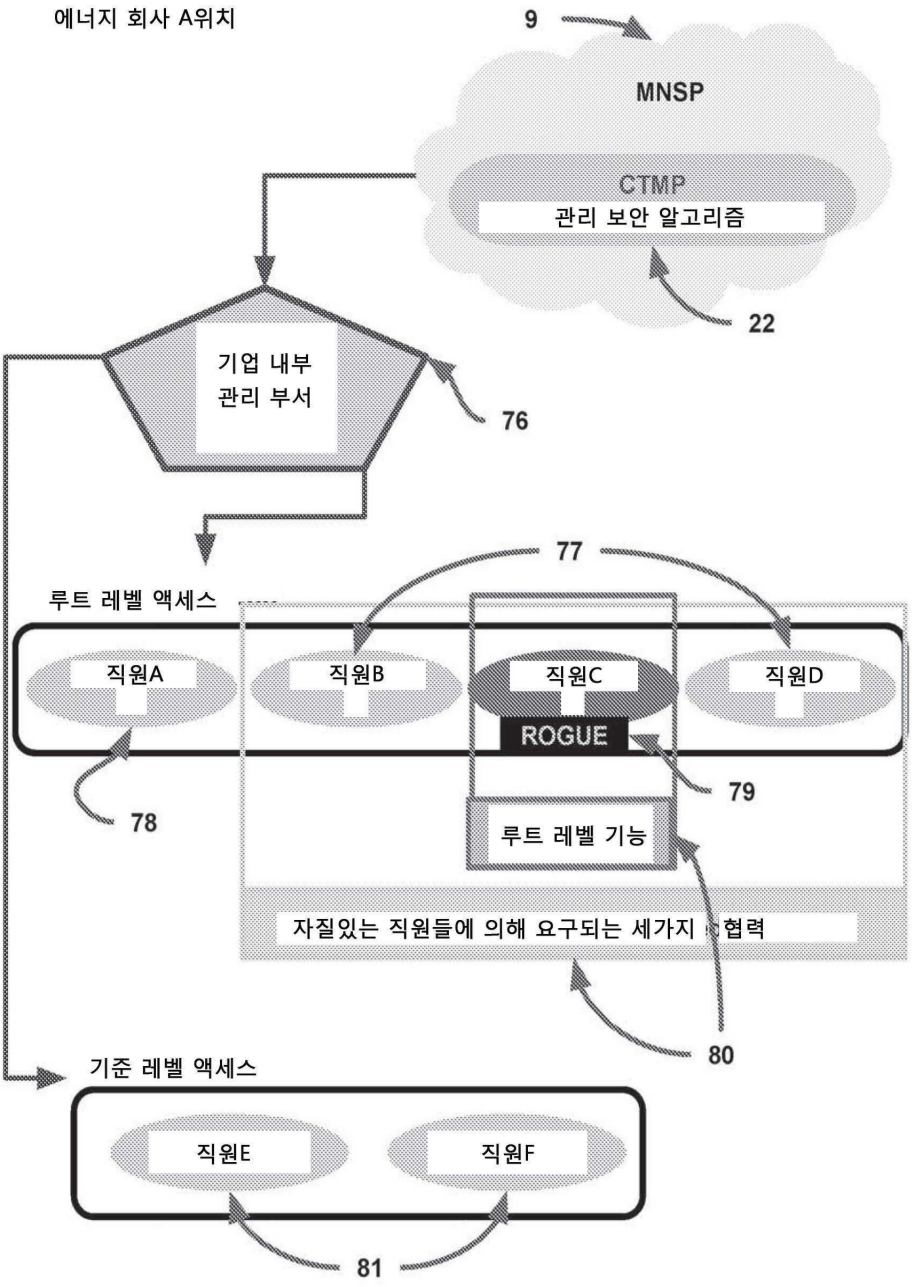
도면15


60
63
해커가 원하는 행동
손상된
자격증명
LAN
인프라
해커
해커
65
62
64
행동의 분석
LAN
인프라
해커
LAN
가상의 복제
에너지 회사 랩탑
43
28
CTMP
MNSP
관리 보안
알고리즘
9
22
실제 보안 응답
61

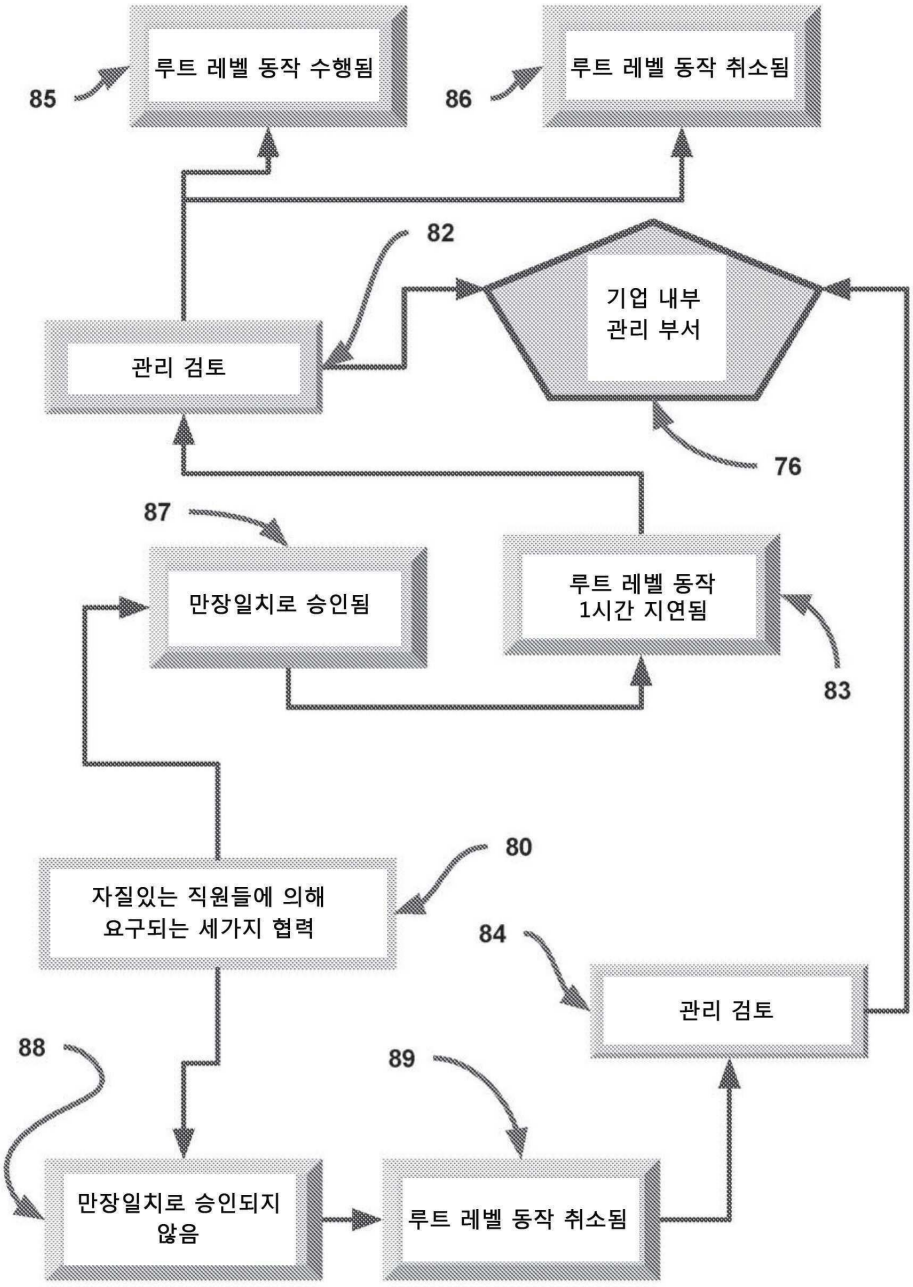
도면16


에너지 회사 A위치
66
비밀번호 세트
비밀번호 A
비밀번호 B
비밀번호 C
67
예정된 내부 인증
프로토콜 액세스(SIAPA)
68
1주
비밀번호 A
비밀번호 B
비밀번호 C
SIAPA
69
2주
비밀번호 A
비밀번호 B
비밀번호 C
SIAPA
70
3주
비밀번호 A
비밀번호 B
비밀번호 C
SIAPA
71
4주
비밀번호 A
비밀번호 B
비밀번호 C
9
73
100% 모의 데이터 환경
LAN 인프라
가상의 복제
MNSP
72
SIAPA
74
잘못된 프로토콜 사용
포털 로그인/
인증 프롬프트
66
75
55
해커

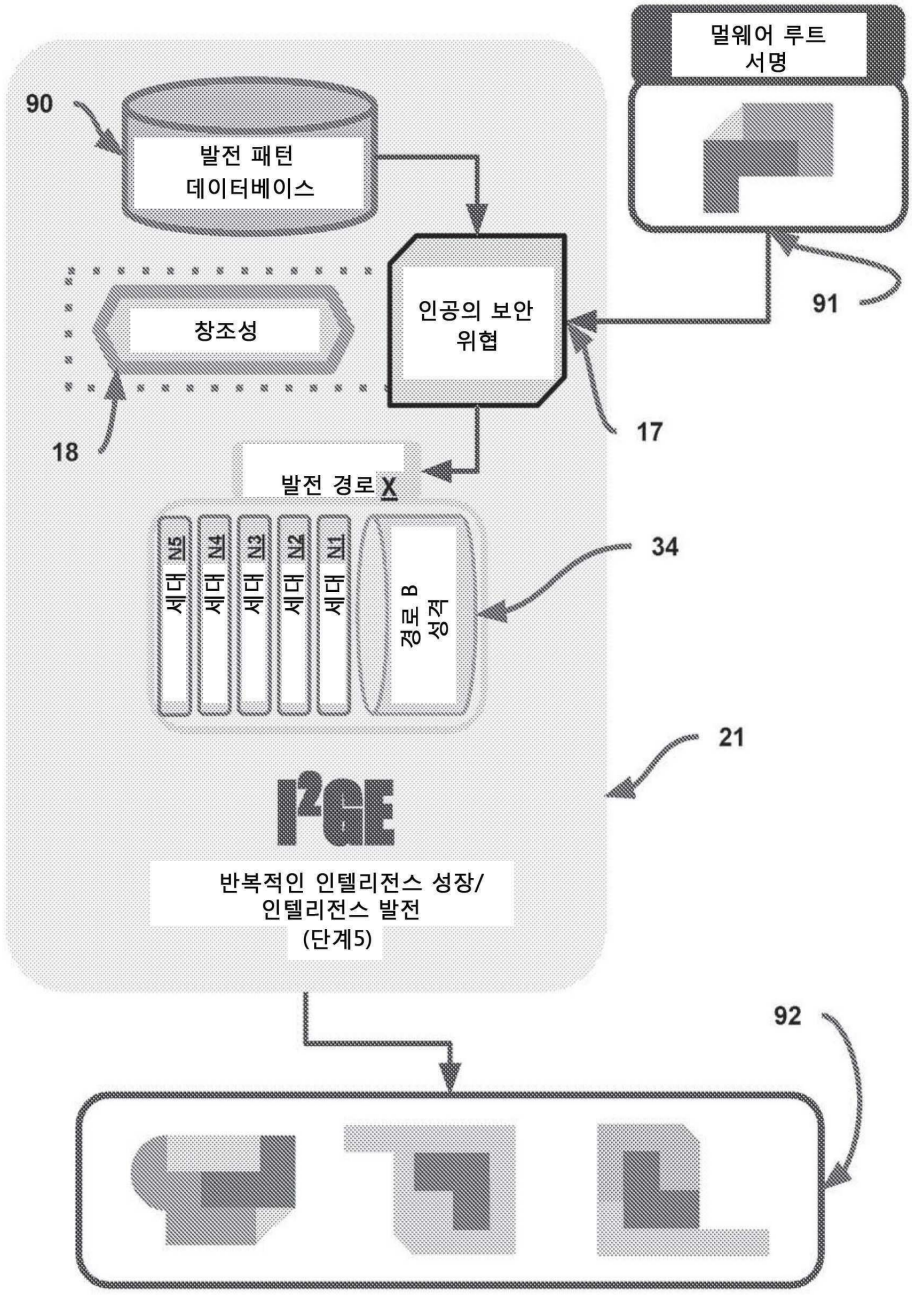
도면17


에너지 회사 A위치
9
MNSP
CTMP
관리 보안 알고리즘
22
기업 내부
관리 부서
76
77
루트 레벨 액세스
직원A
직원B
직원C
직원D
ROGUE
78
79
루트 레벨 기능
자질있는 직원들에 의해 요구되는 세가지 협력
80
기준 레벨 액세스
직원E
직원F
81

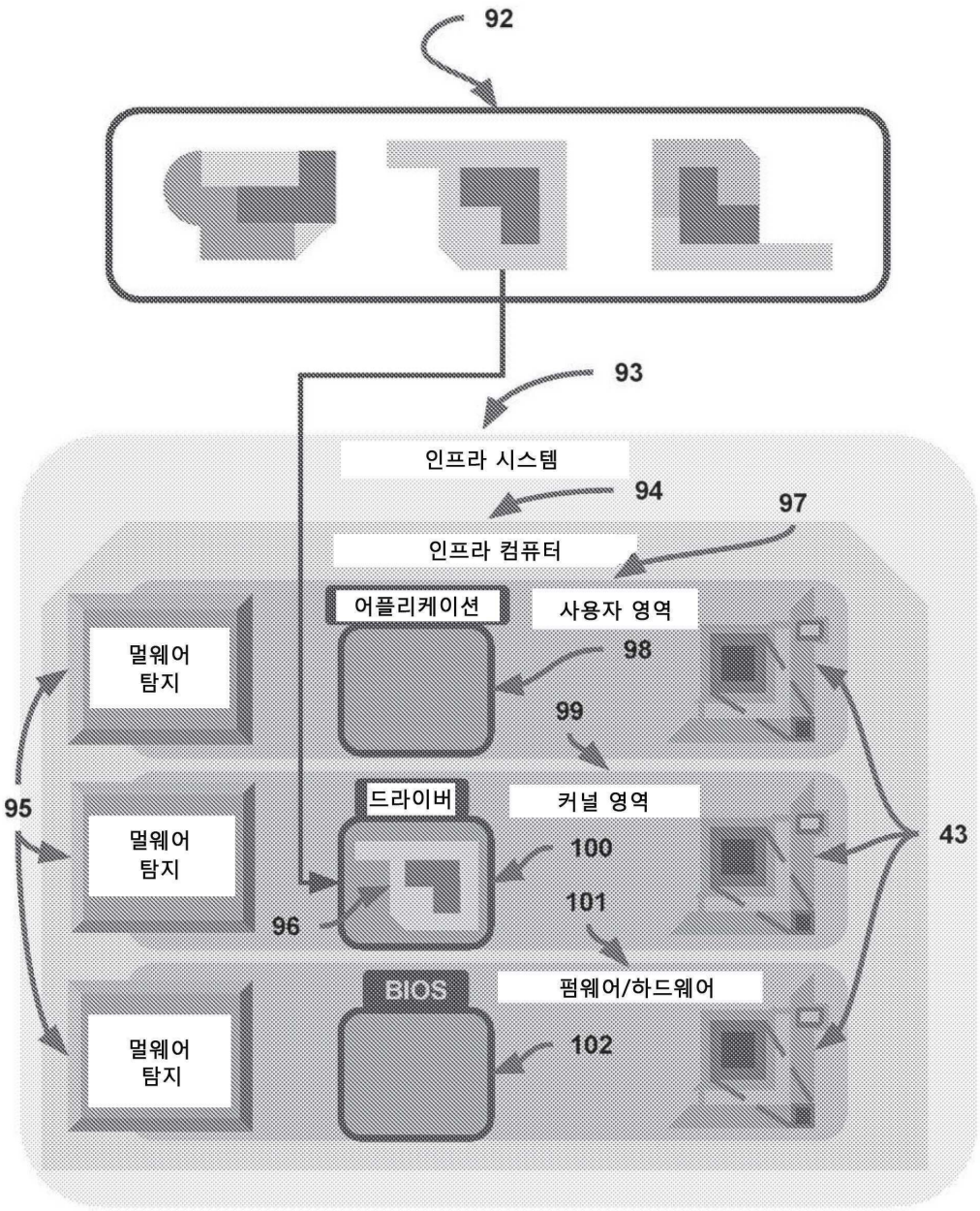
도면18


루트 레벨 동작 수행됨
85
루트 레벨 동작 취소됨
86
82
기업 내부
관리 부서
관리 검토
76
87
만장일치로 승인됨
루트 레벨 동작
1시간 지연됨
83
80
자질있는 직원들에 의해
요구되는 세가지 협력
84
관리 검토
88
89
만장일치로 승인되지
않음
루트 레벨 동작 취소됨

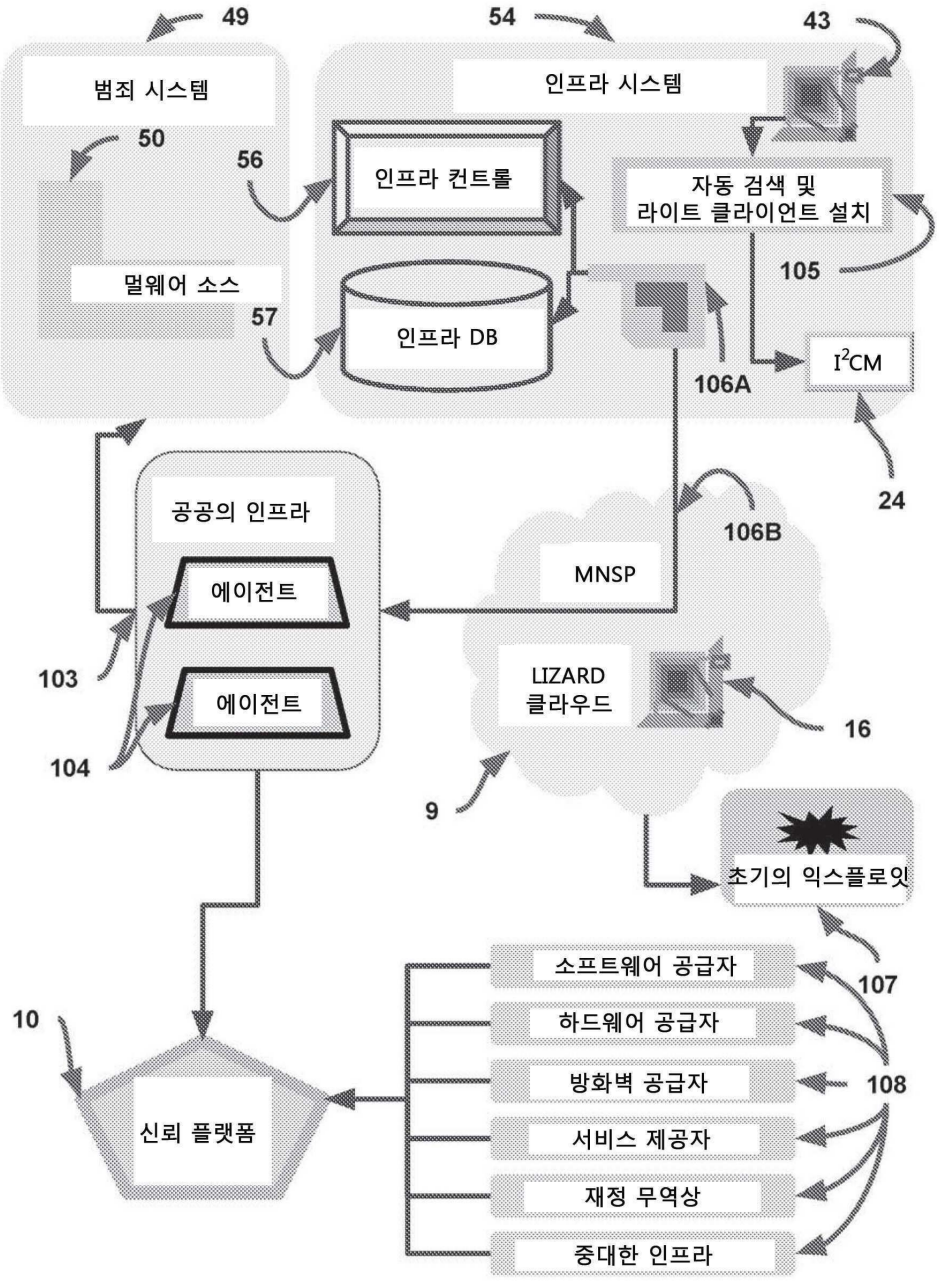
도면19


멀웨어 루트
서명
90
발전 패턴
데이터베이스
91
창조성
인공의 보안
위협
18
17
발전 경로 X
5번째 세대
4번째 세대
3번째 세대
2번째 세대
1번째 세대
경로 B
성격
34
21
PGE
반복적인 인텔리전스 성장/
인텔리전스 발전
(단계5)
92

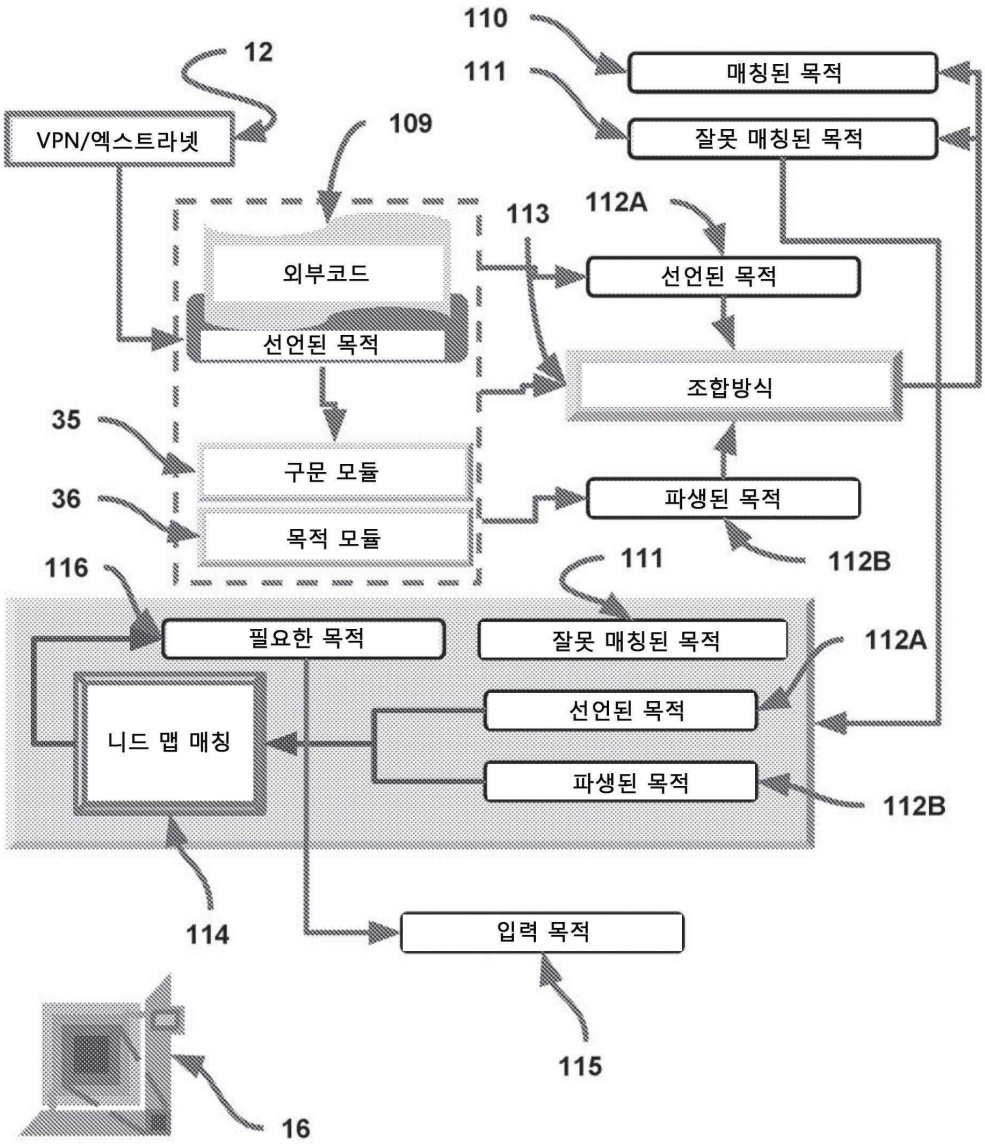
도면20


92
93
인프라 시스템
94
97
인프라 컴퓨터
어플리케이션
사용자 영역
98
99
멀웨어
탐지
95
드라이버
커널 영역
100
101
43
96
멀웨어
탐지
BIOS
펌웨어/하드웨어
102
멀웨어
탐지

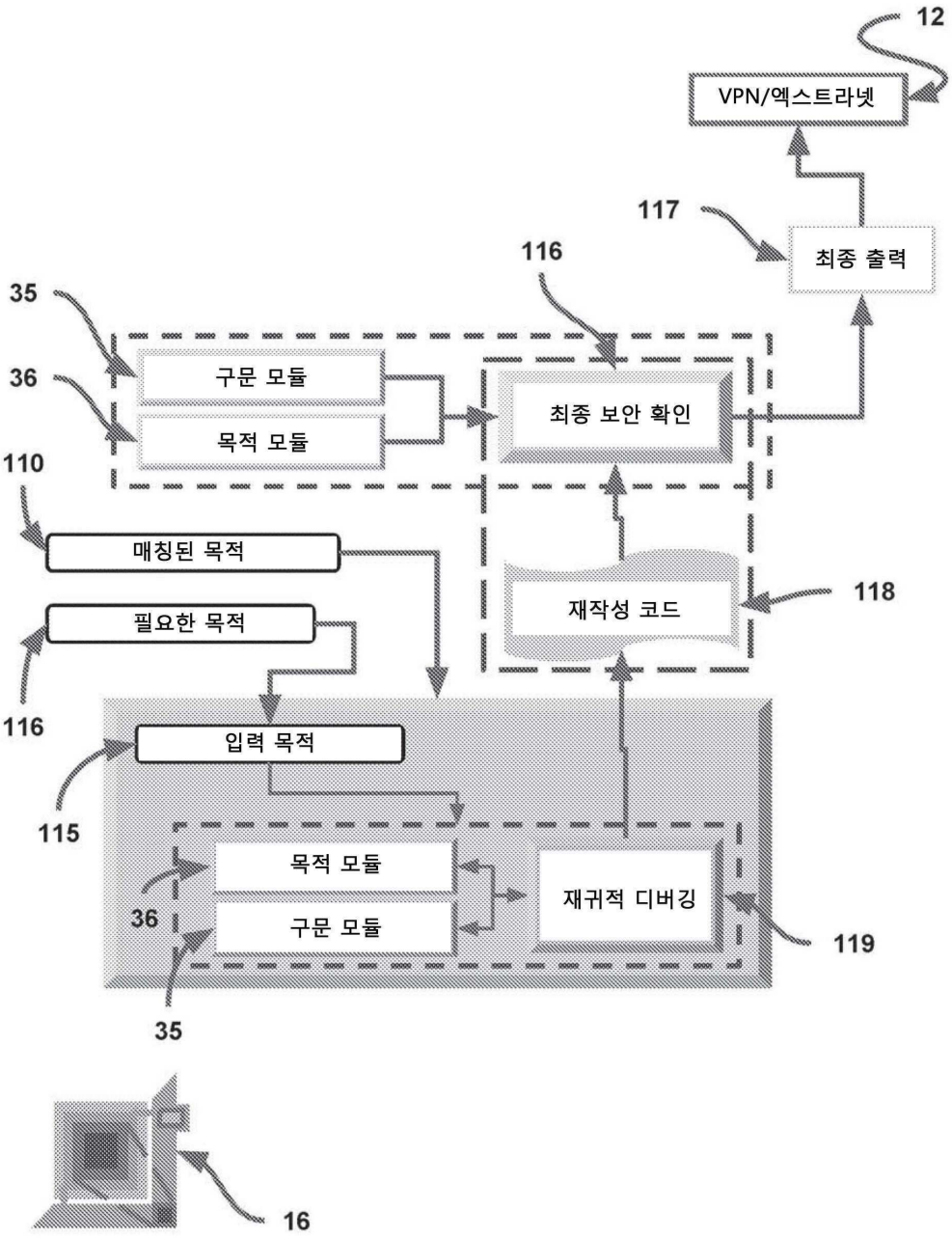
도면21


49
54
43
범죄 시스템
인프라 시스템
50
56
인프라 컨트롤
자동 검색 및
라이트 클라이언트 설치
105
멀웨어 소스
57
인프라 DB
106A
I²CM
24
공공의 인프라
106B
MNSP
에이전트
103
에이전트
LIZARD
클라우드
16
104
9
초기의 익스플로잇
107
소프트웨어 공급자
하드웨어 공급자
10
방화벽 공급자
108
신뢰 플랫폼
서비스 제공자
재정 무역상
중대한 인프라

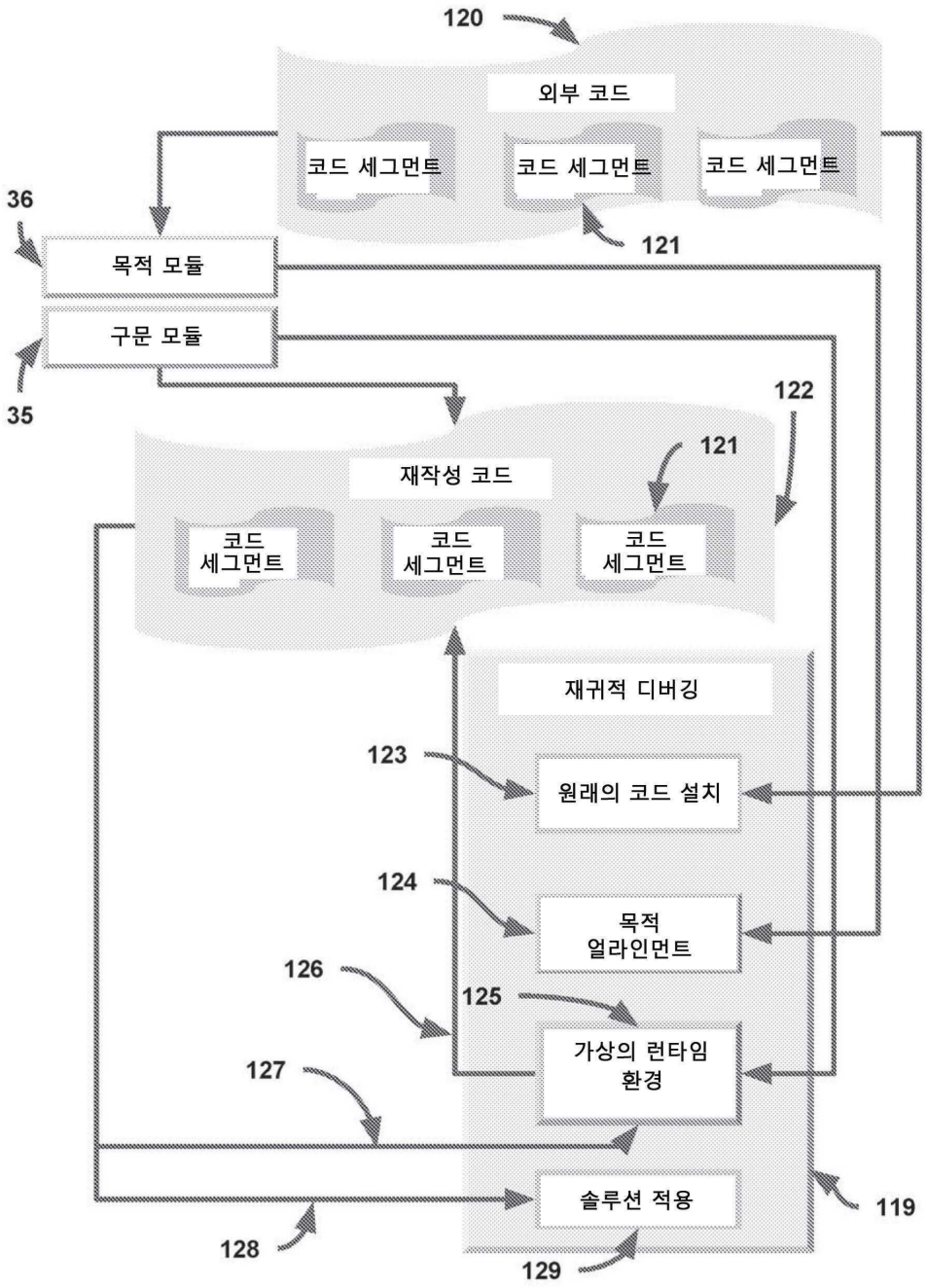
도면22


110
111
매칭된 목적
잘못 매칭된 목적
12
VPN/엑스트라넷
109
113
112A
외부코드
선언된 목적
조합방식
35
구문 모듈
36
목적 모듈
파생된 목적
112B
116
111
필요한 목적
잘못 매칭된 목적
112A
니드 맵 매칭
선언된 목적
파생된 목적
112B
114
입력 목적
115
16

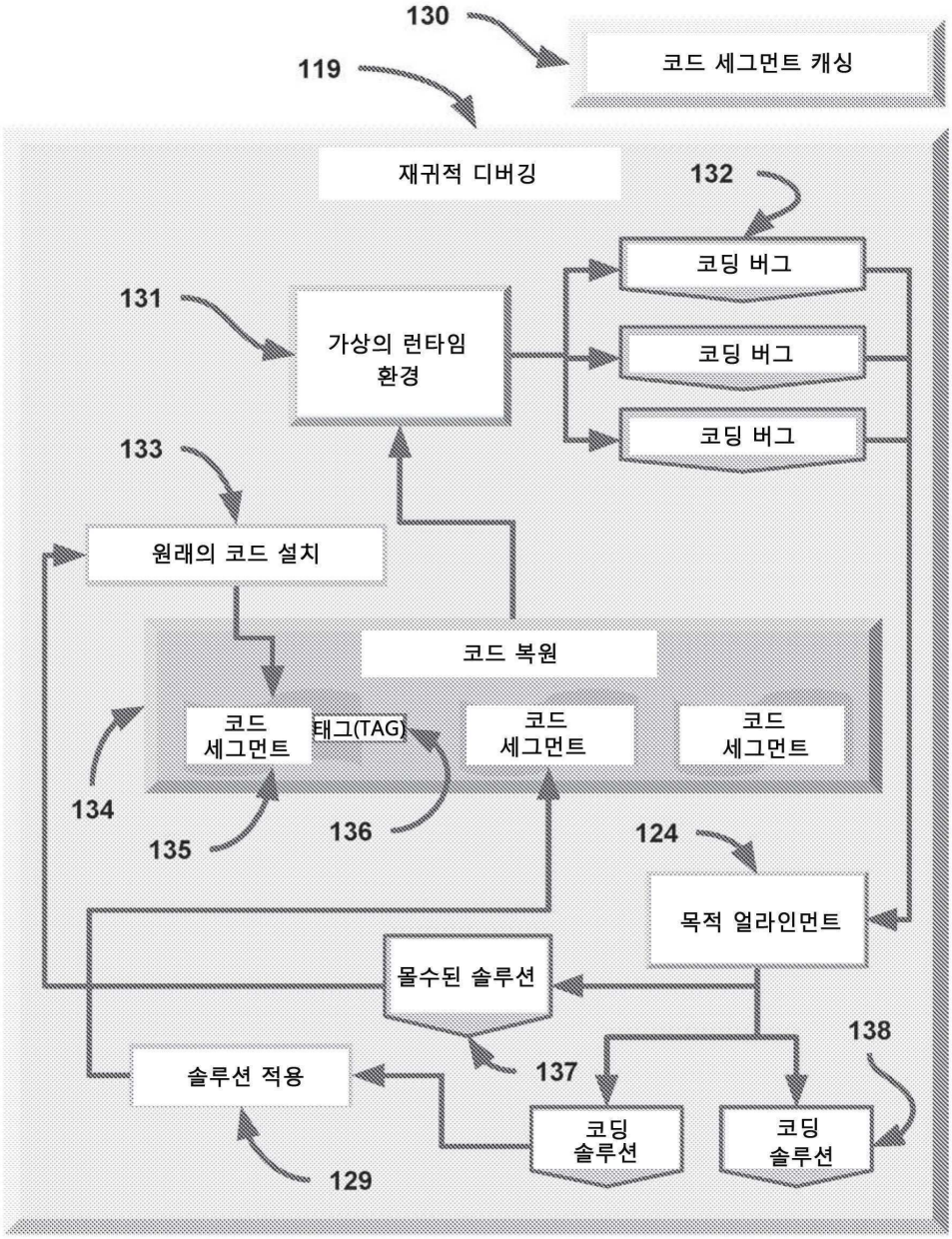
도면23


12
VPN/엑스트라넷
117
최종 출력
116
35
구문 모듈
36
목적 모듈
최종 보안 확인
110
매칭된 목적
재작성 코드
118
필요한 목적
116
입력 목적
115
목적 모듈
재귀적 디버깅
36
구문 모듈
119
35
16

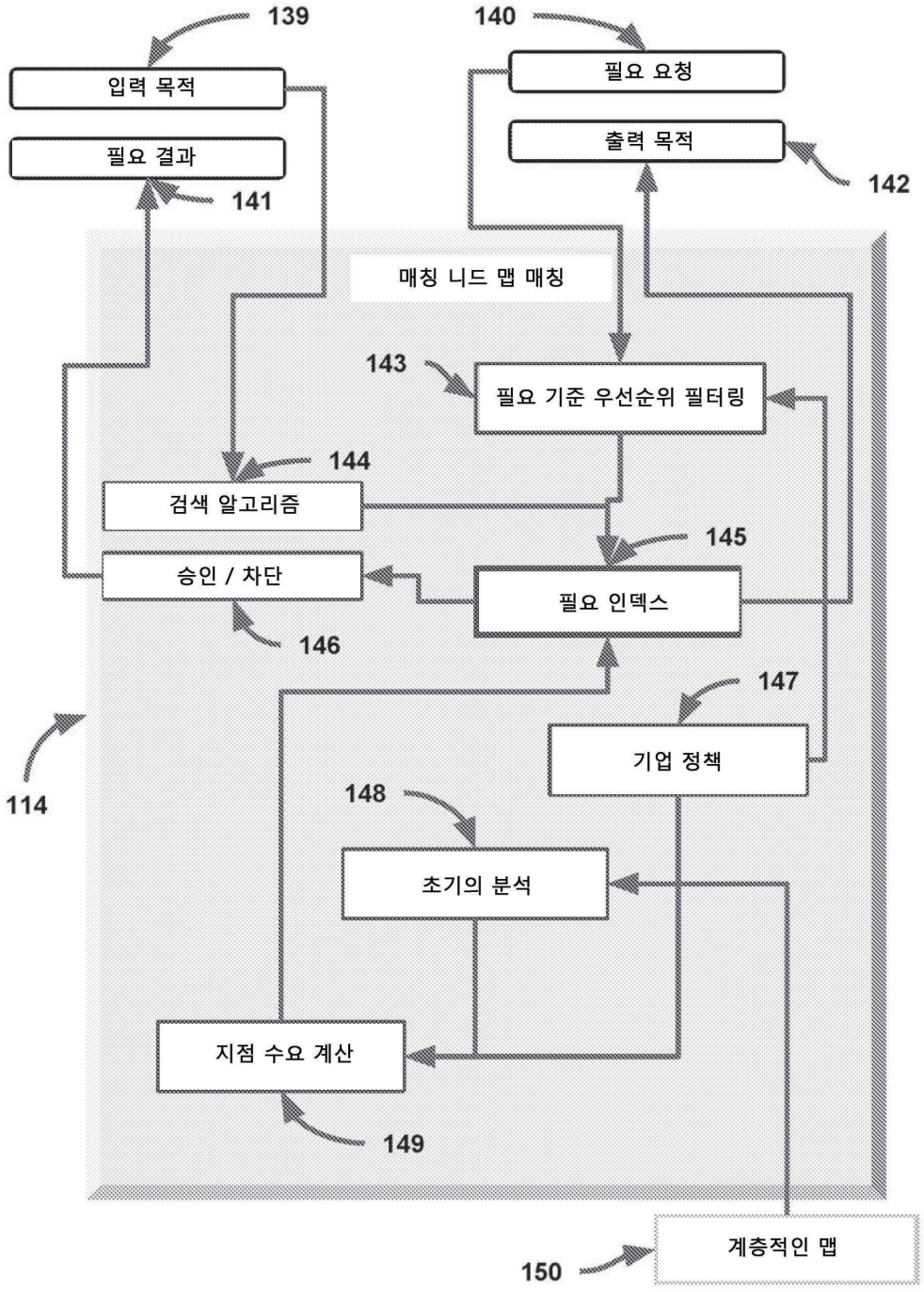
도면24


120
외부 코드
코드 세그먼트
코드 세그먼트
코드 세그먼트
121
36
목적 모듈
구문 모듈
35
122
121
재작성 코드
코드 세그먼트
코드 세그먼트
코드 세그먼트
재귀적 디버깅
123
원래의 코드 설치
124
목적 얼라인먼트
126
125
가상의 런타임 환경
127
솔루션 적용
119
128
129

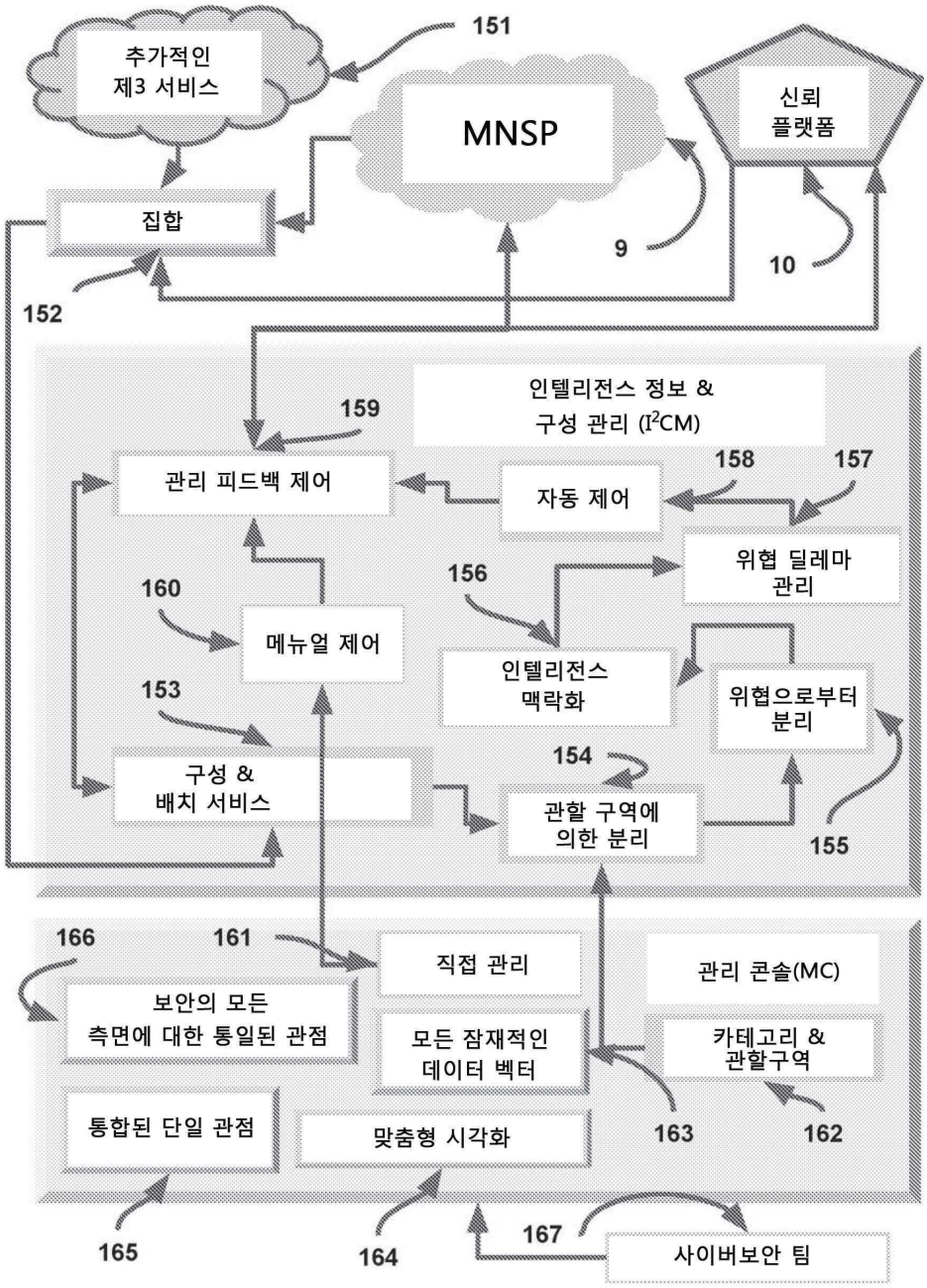
도면25


130
코드 세그먼트 캐싱
119
재귀적 디버깅
132
코딩 버그
131
가상의 런타임 환경
코딩 버그
코딩 버그
133
원래의 코드 설치
코드 복원
코드 세그먼트
태그(TAG)
코드 세그먼트
코드 세그먼트
134
135
136
124
목적 얼라인먼트
몰수된 솔루션
138
137
솔루션 적용
코딩 솔루션
코딩 솔루션
129

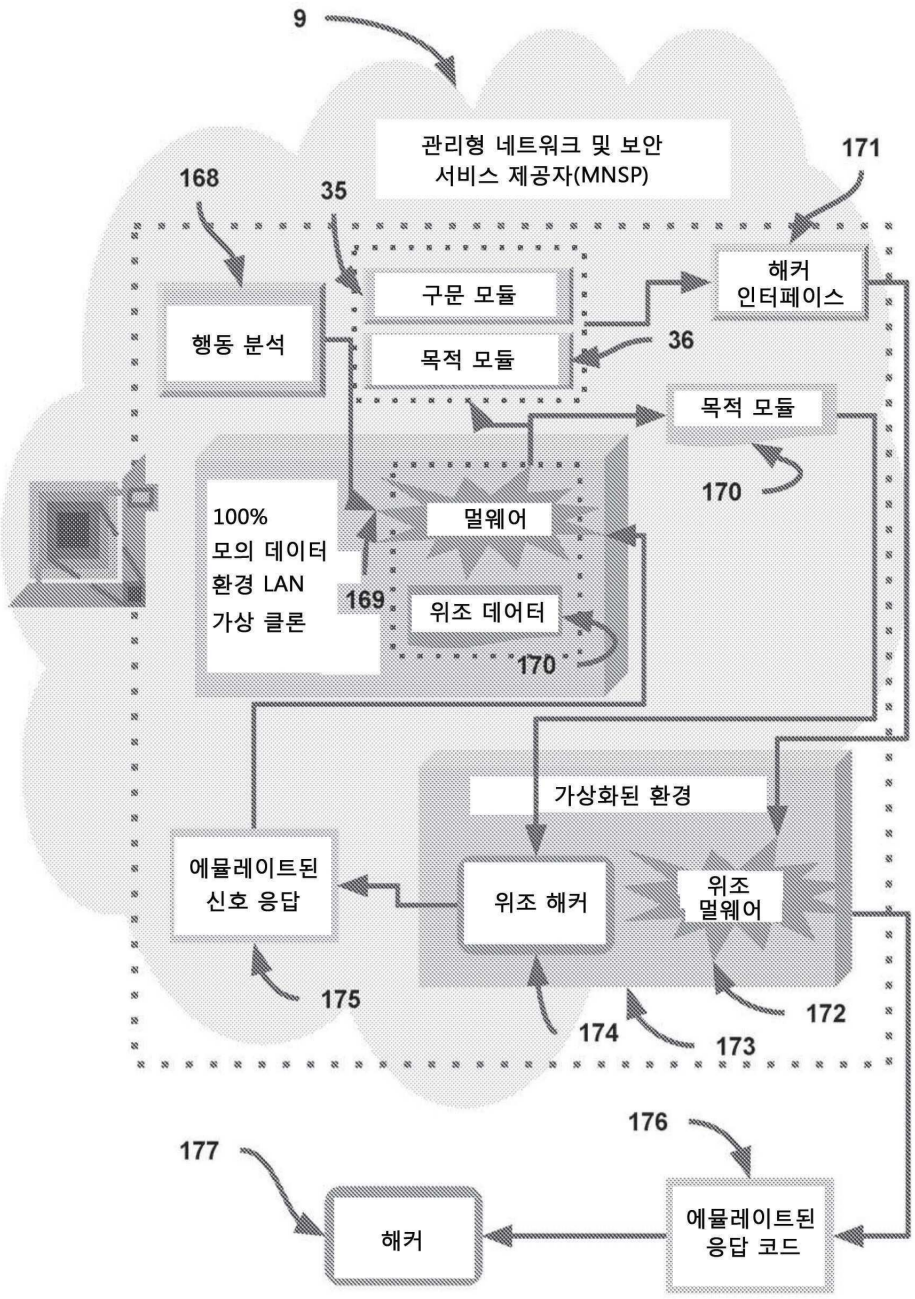
도면26


139
입력 목적
필요 결과
141
140
필요 요청
출력 목적
142
매칭 니드 맵 매칭
143
필요 기준 우선순위 필터링
144
검색 알고리즘
145
승인 / 차단
필요 인덱스
146
147
기업 정책
114
148
초기의 분석
지점 수요 계산
149
150
계층적인 맵

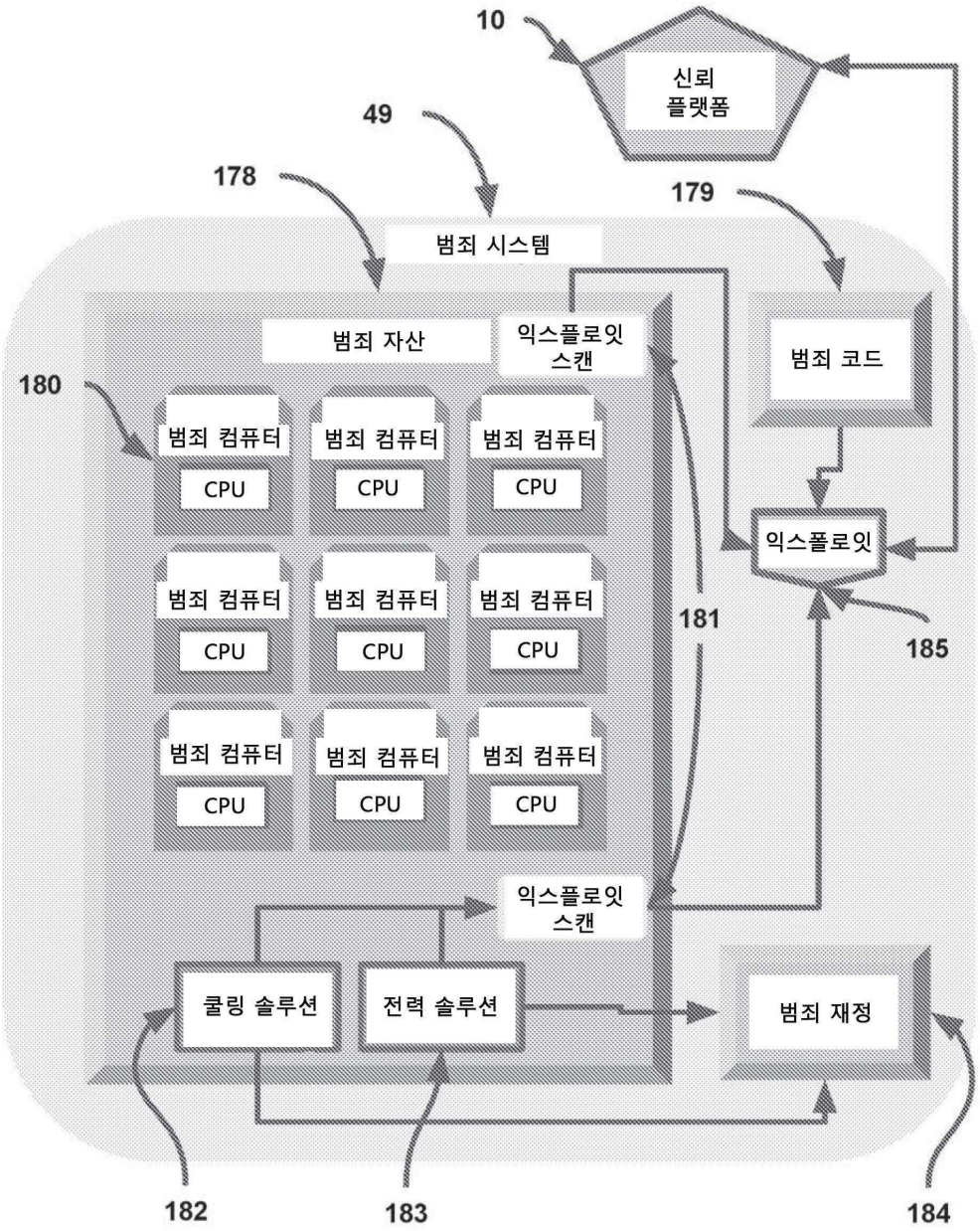
도면27


151
추가적인
제3 서비스
MNSP
신뢰
플랫폼
집합
9
10
152
인텔리전스 정보 &
구성 관리 (I²CM)
159
관리 피드백 제어
158
157
자동 제어
위협 딜레마
관리
156
160
메뉴얼 제어
인텔리전스
맥락화
153
위협으로부터
분리
154
구성 &
배치 서비스
관할 구역에
의한 분리
155
166
161
직접 관리
관리 콘솔(MC)
보안의 모든
측면에 대한 통일된 관점
모든 잠재적인
데이터 벡터
카테고리 &
관할구역
통합된 단일 관점
맞춤형 시각화
163
162
165
164
167
사이버보안 팀

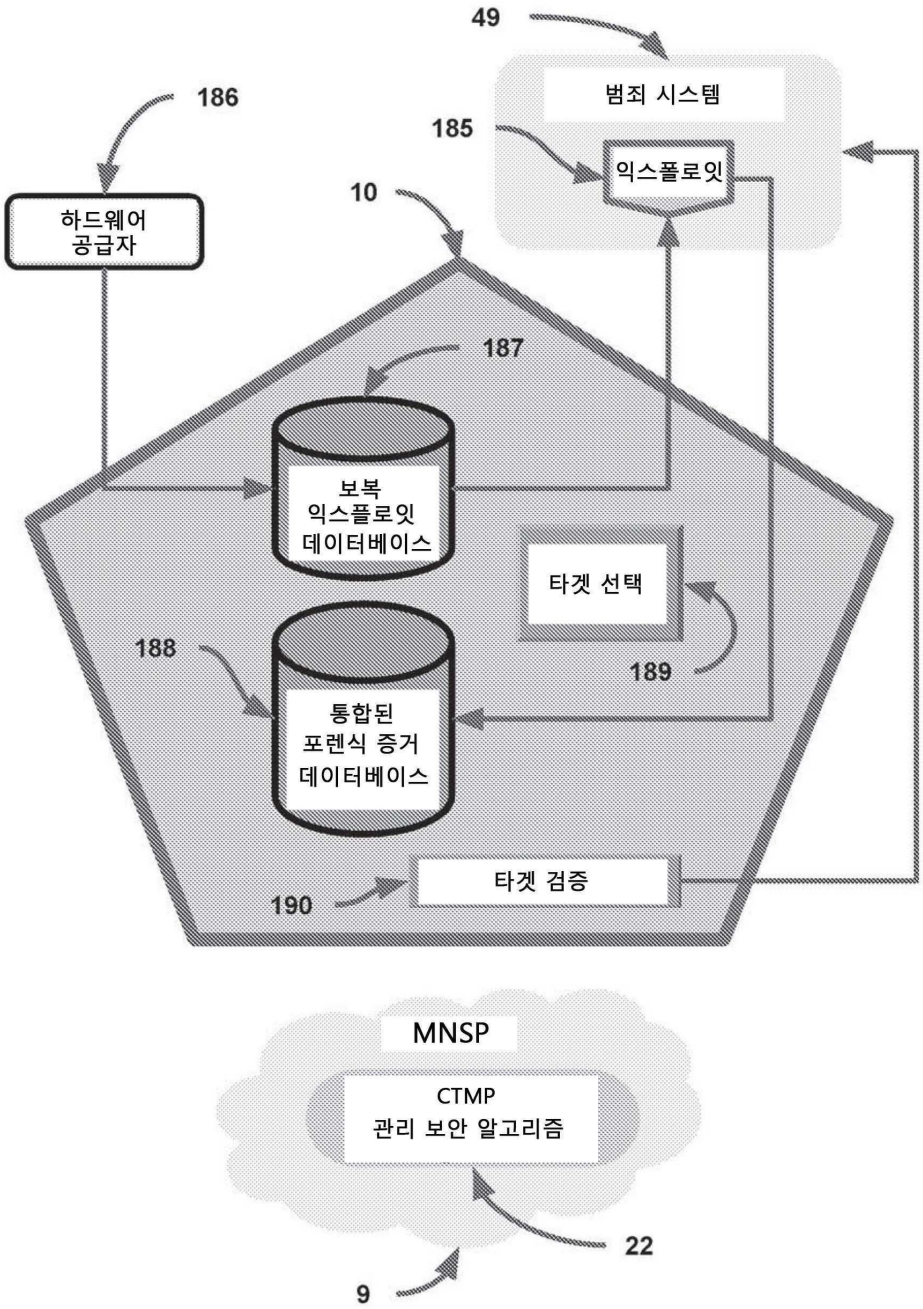
도면28


9
관리형 네트워크 및 보안
서비스 제공자(MNSP)
168
35
171
구문 모듈
해커
인터페이스
행동 분석
목적 모듈
36
목적 모듈
170
100%
모의 데이터
환경 LAN
가상 클론
멀웨어
169
위조 데이터
170
가상화된 환경
에뮬레이트된
신호 응답
위조 해커
위조
멀웨어
175
174
173
172
176
177
해커
에뮬레이트된
응답 코드

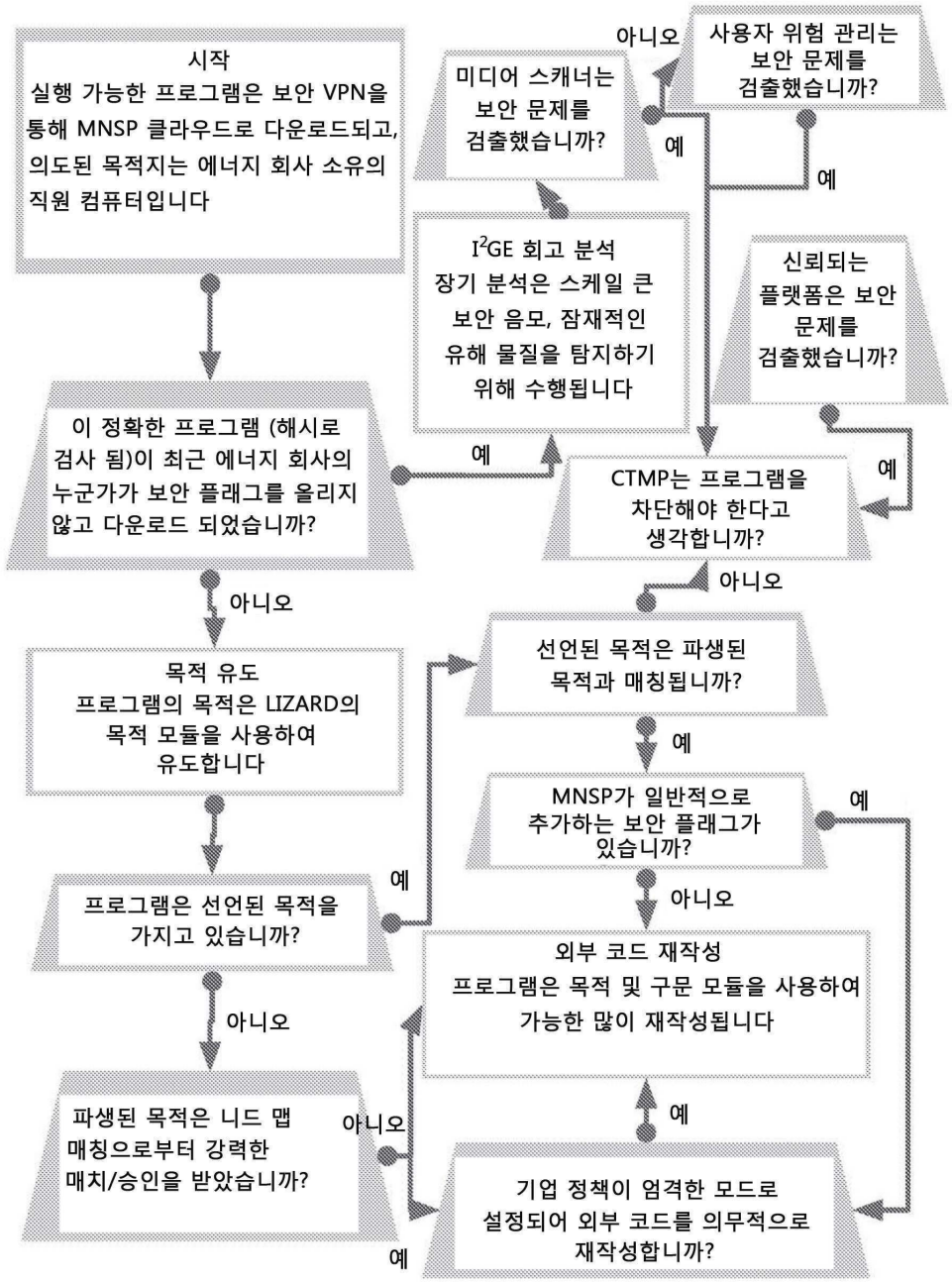
도면29


10
신뢰
플랫폼
49
178
179
범죄 시스템
범죄 자산
익스플로잇
스캔
범죄 코드
180
범죄 컴퓨터
CPU
익스플로잇
181
185
익스플로잇
스캔
쿨링 솔루션
전력 솔루션
범죄 재정
182
183
184

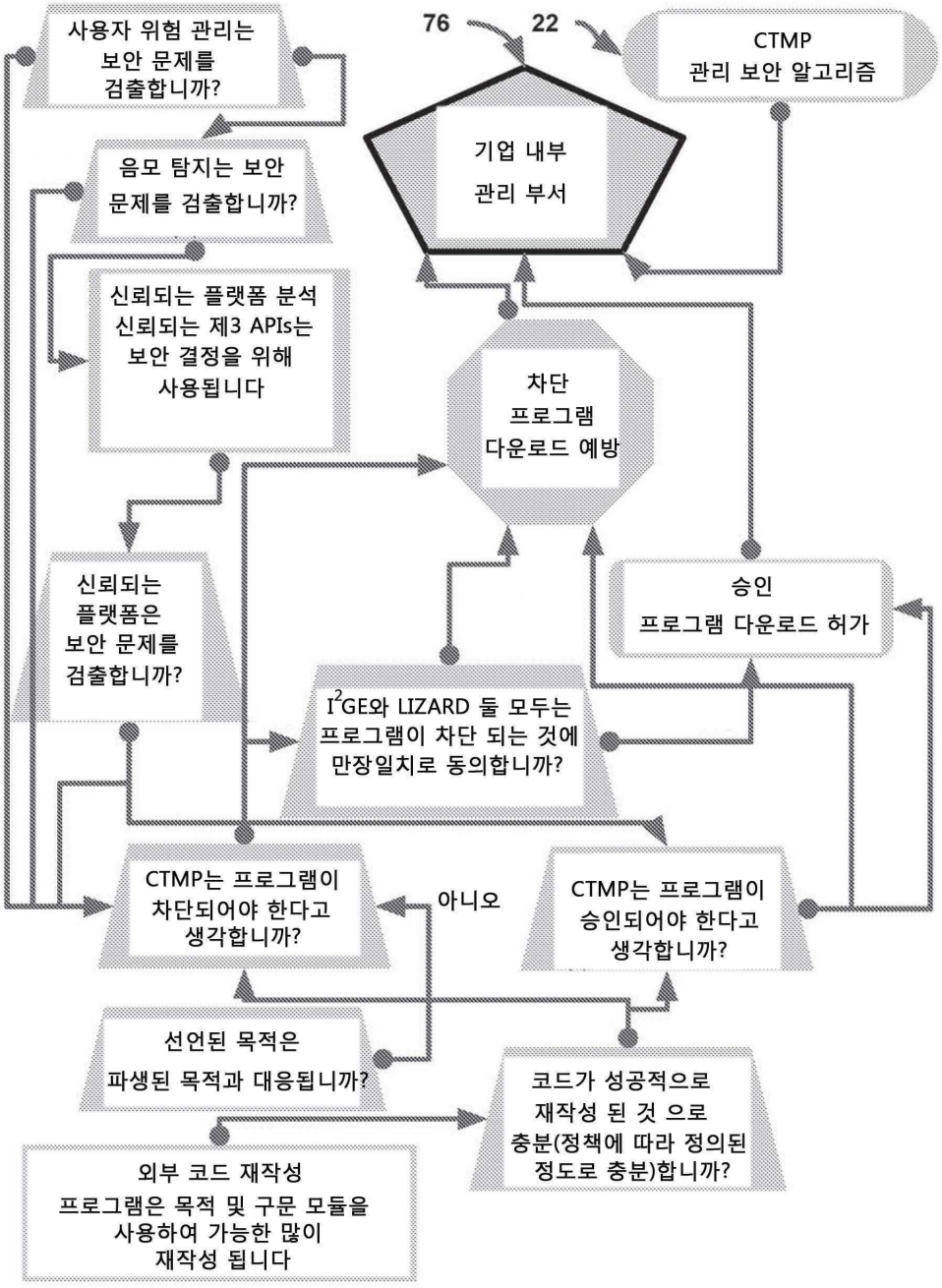
도면30


49
범죄 시스템
185
익스폴로잇
186
하드웨어
공급자
10
187
보복
익스플로잇
데이터베이스
타겟 선택
188
189
통합된
포렌식 증거
데이터베이스
타겟 검증
190
MNSP
CTMP
관리 보안 알고리즘
22
9

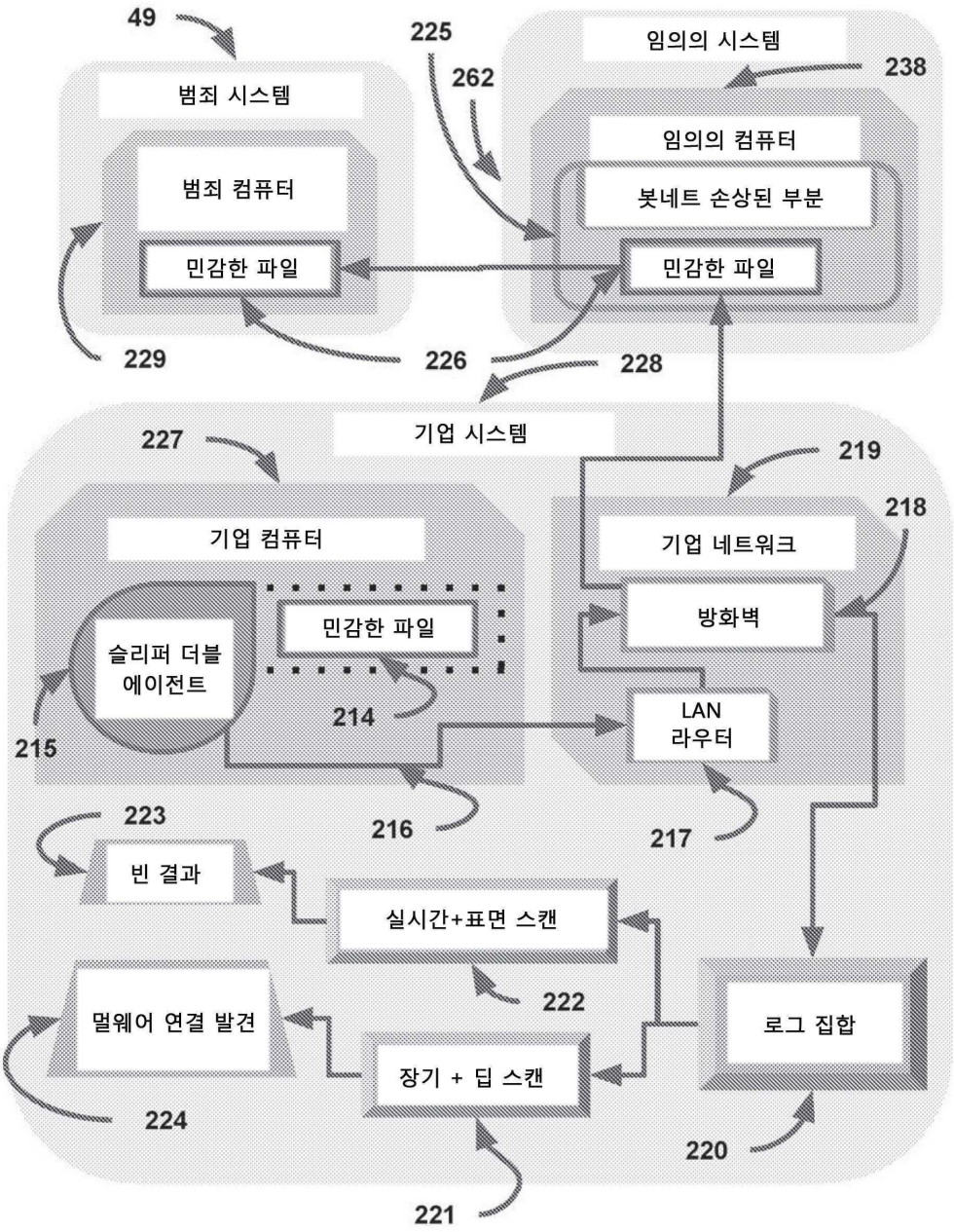
도면31


시작
실행 가능한 프로그램은 보안 VPN을 통해 MNSP 클라우드로 다운로드되고, 의도된 목적지는 에너지 회사 소유의 직원 컴퓨터입니다
이 정확한 프로그램 (해시로 검사 됨)이 최근 에너지 회사의 누군가가 보안 플래그를 올리지 않고 다운로드 되었습니까?
예
아니오
목적 유도
프로그램의 목적은 LIZARD의 목적 모듈을 사용하여 유도합니다
프로그램은 선언된 목적을 가지고 있습니까?
예
아니오
파생된 목적은 니드 맵 매칭으로부터 강력한 매치/승인을 받았습니까?
아니오
예
I2GE 회고 분석
장기 분석은 스케일 큰 보안 음모, 잠재적인 유해 물질을 탐지하기 위해 수행됩니다
미디어 스캐너는 보안 문제를 검출했습니까?
아니오
예
사용자 위험 관리는 보안 문제를 검출했습니까?
예
신뢰되는 플랫폼은 보안 문제를 검출했습니까?
예
CTMP는 프로그램을 차단해야 한다고 생각합니까?
아니오
선언된 목적은 파생된 목적과 매칭됩니까?
예
MNSP가 일반적으로 추가하는 보안 플래그가 있습니까?
예
아니오
외부 코드 재작성
프로그램은 목적 및 구문 모듈을 사용하여 가능한 많이 재작성됩니다
예
기업 정책이 엄격한 모드로 설정되어 외부 코드를 의무적으로 재작성합니까?

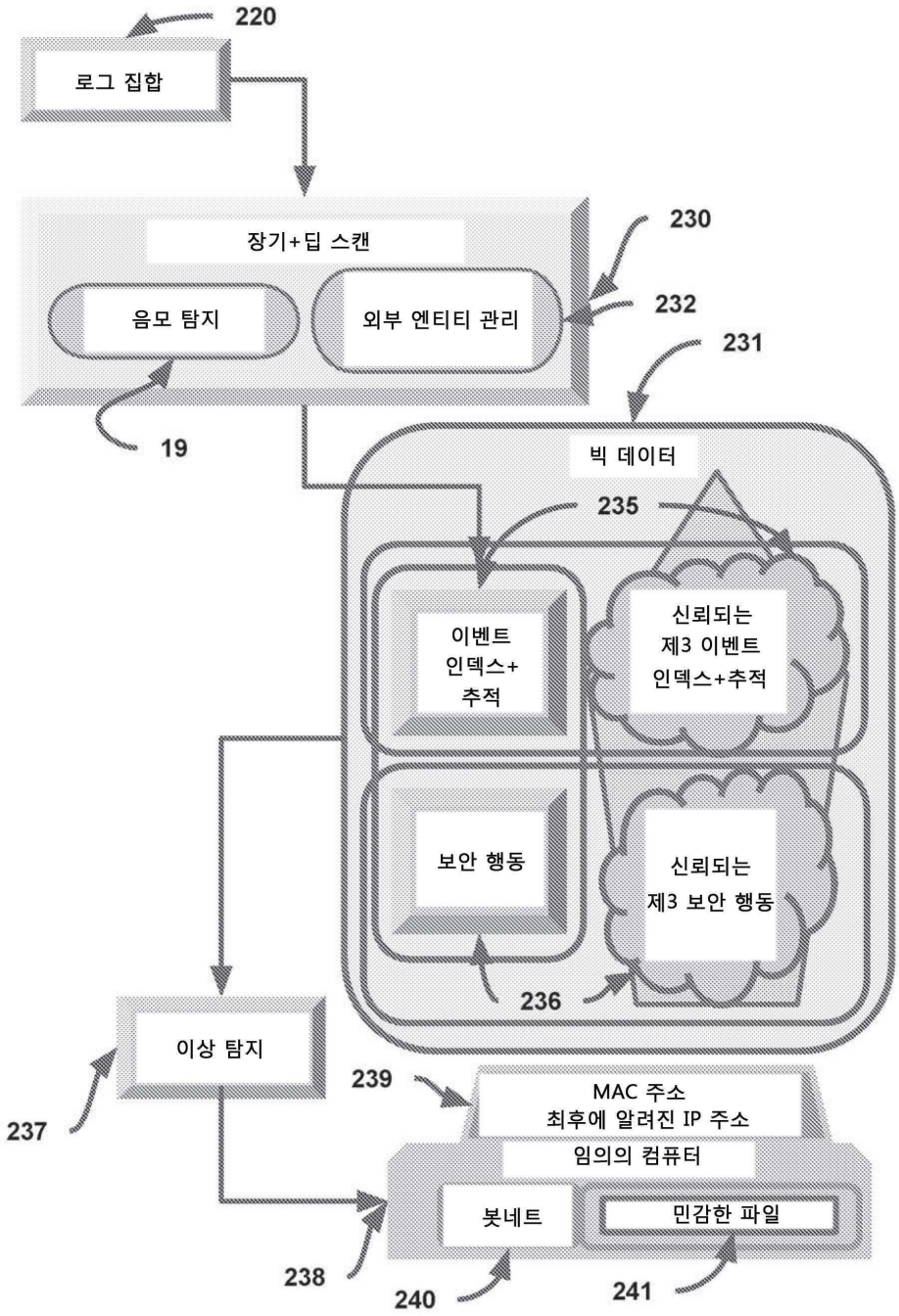
도면32


사용자 위험 관리는 보안 문제를 검출합니까?
76
22
CTMP 관리 보안 알고리즘
기업 내부 관리 부서
음모 탐지는 보안 문제를 검출합니까?
신뢰되는 플랫폼 분석 신뢰되는 제3 APIs는 보안 결정을 위해 사용됩니다
차단 프로그램 다운로드 예방
신뢰되는 플랫폼은 보안 문제를 검출합니까?
승인 프로그램 다운로드 허가
I2GE와 LIZARD 둘 모두는 프로그램이 차단 되는 것에 만장일치로 동의합니까?
CTMP는 프로그램이 차단되어야 한다고 생각합니까?
아니오
CTMP는 프로그램이 승인되어야 한다고 생각합니까?
선언된 목적은 파생된 목적과 대응됩니까?
코드가 성공적으로 재작성 된 것 으로 충분(정책에 따라 정의된 정도로 충분)합니까?
외부 코드 재작성 프로그램은 목적 및 구문 모듈을 사용하여 가능한 많이 재작성 됩니다

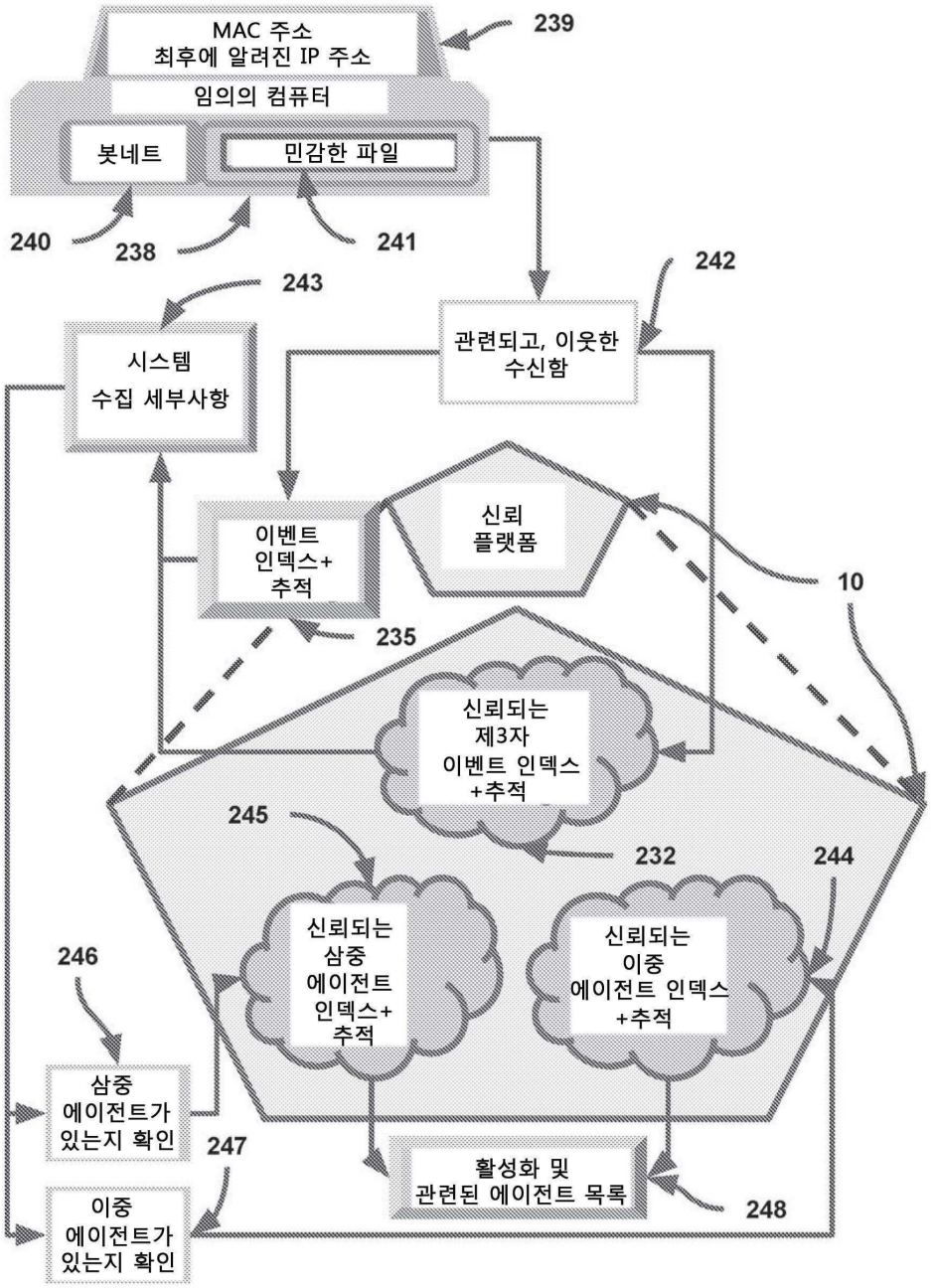
도면33


49
범죄 시스템
범죄 컴퓨터
민감한 파일
225
262
임의의 시스템
238
임의의 컴퓨터
봇네트 손상된 부분
민감한 파일
229
226
228
기업 시스템
227
219
218
기업 컴퓨터
기업 네트워크
방화벽
민감한 파일
슬리퍼 더블 에이전트
214
LAN 라우터
215
216
217
223
빈 결과
실시간+표면 스캔
222
멀웨어 연결 발견
로그 집합
장기 + 딥 스캔
224
220
221

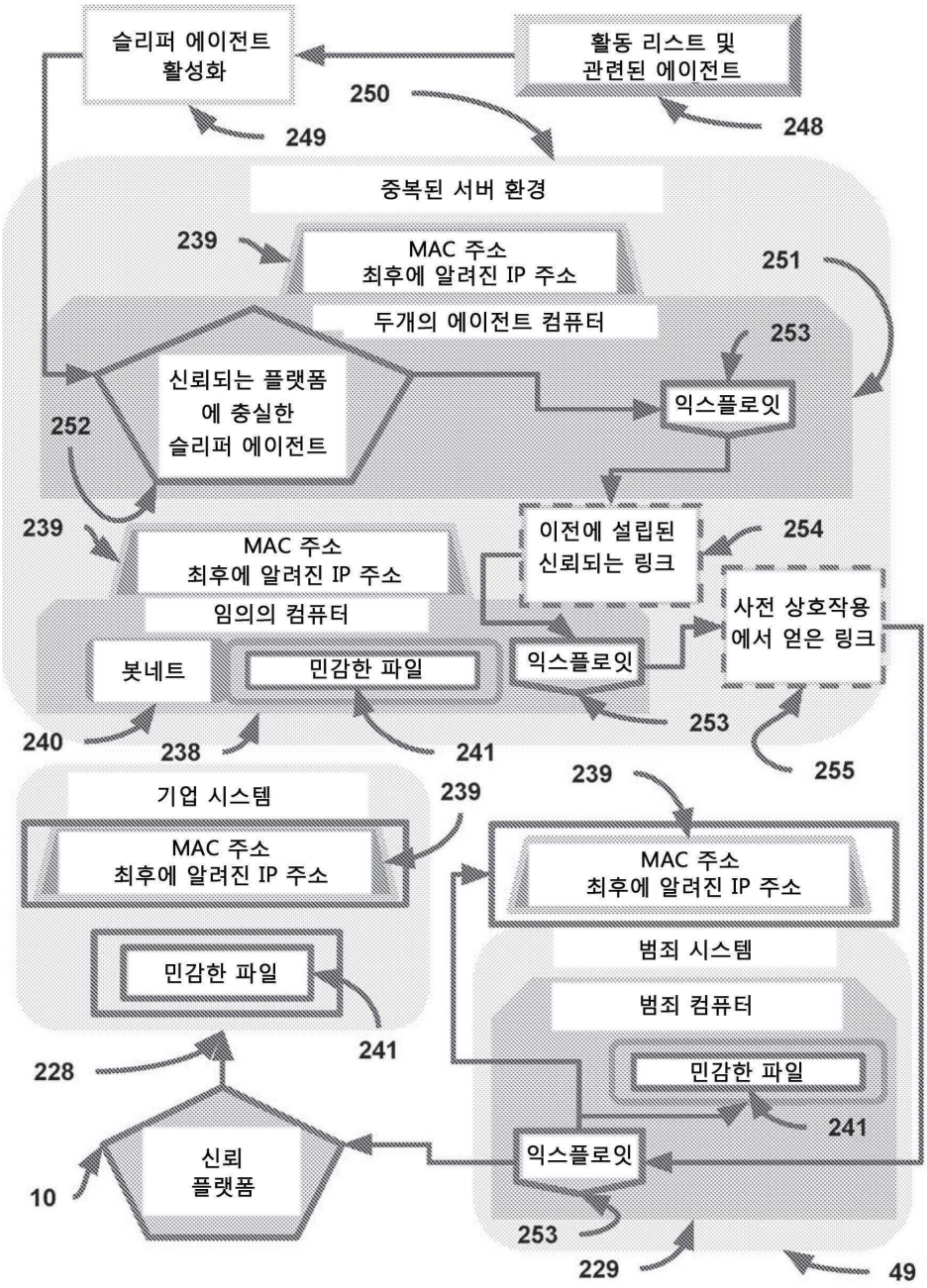
도면34


220
로그 집합
장기+딥 스캔
230
음모 탐지
외부 엔티티 관리
232
231
19
빅 데이터
235
이벤트
인덱스+
추적
신뢰되는
제3 이벤트
인덱스+추적
보안 행동
신뢰되는
제3 보안 행동
236
이상 탐지
239
237
MAC 주소
최후에 알려진 IP 주소
임의의 컴퓨터
봇네트
민감한 파일
238
240
241

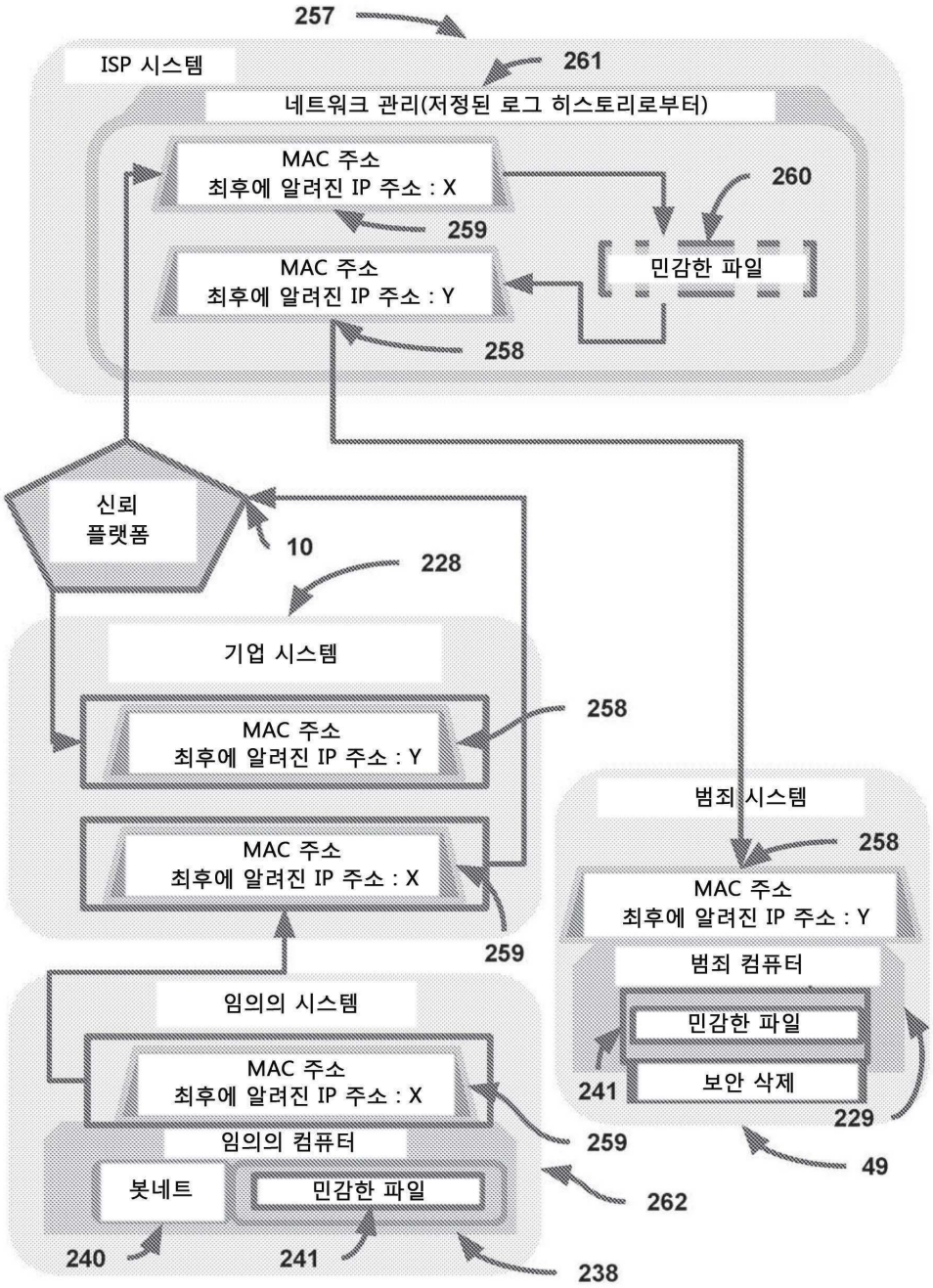
도면35


MAC 주소
최후에 알려진 IP 주소
239
임의의 컴퓨터
봇네트
민감한 파일
240
238
241
242
243
시스템
수집 세부사항
관련되고, 이웃한
수신함
이벤트
인덱스+
추적
신뢰
플랫폼
10
235
신뢰되는
제3자
이벤트 인덱스
+추적
245
232
244
신뢰되는
삼중
에이전트
인덱스+
추적
신뢰되는
이중
에이전트 인덱스
+추적
246
삼중
에이전트가
있는지 확인
247
이중
에이전트가
있는지 확인
활성화 및
관련된 에이전트 목록
248

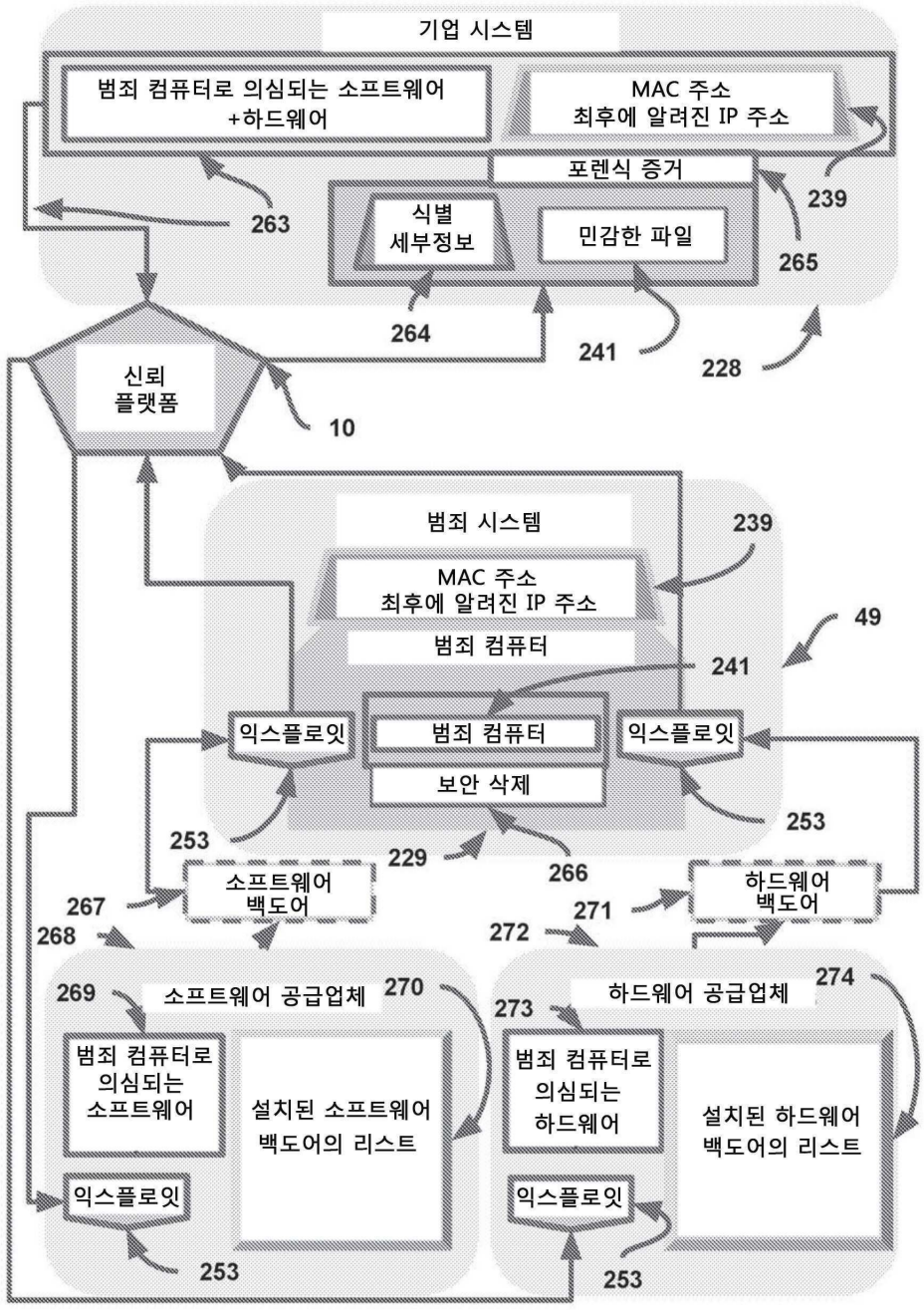
도면36


슬리퍼 에이전트 활성화
활동 리스트 및 관련된 에이전트
250
249
248
중복된 서버 환경
239
MAC 주소 최후에 알려진 IP 주소
251
두개의 에이전트 컴퓨터
253
신뢰되는 플랫폼에 충실한 슬리퍼 에이전트
익스플로잇
252
239
MAC 주소 최후에 알려진 IP 주소
이전에 설립된 신뢰되는 링크
254
사전 상호작용에서 얻은 링크
임의의 컴퓨터
봇네트
민감한 파일
익스플로잇
253
240
238
241
255
239
기업 시스템
239
MAC 주소 최후에 알려진 IP 주소
MAC 주소 최후에 알려진 IP 주소
범죄 시스템
민감한 파일
241
범죄 컴퓨터
228
민감한 파일
241
신뢰 플랫폼
익스플로잇
10
253
229
49

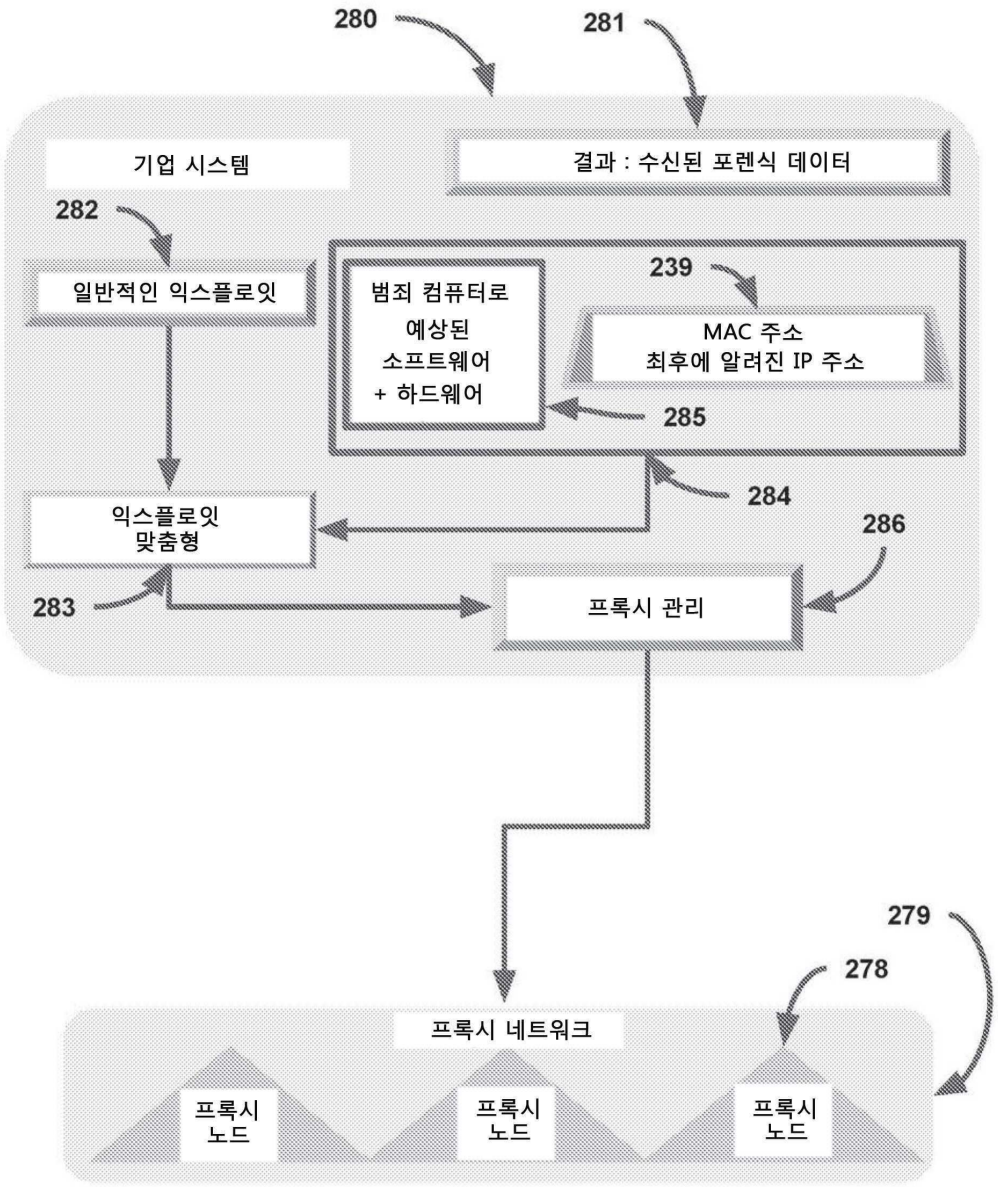
도면37


257
ISP 시스템
261
네트워크 관리(저정된 로그 히스토리로부터)
MAC 주소
최후에 알려진 IP 주소 : X
260
259
MAC 주소
최후에 알려진 IP 주소 : Y
민감한 파일
258
신뢰
플랫폼
10
228
기업 시스템
258
MAC 주소
최후에 알려진 IP 주소 : Y
MAC 주소
최후에 알려진 IP 주소 : X
259
범죄 시스템
258
MAC 주소
최후에 알려진 IP 주소 : Y
범죄 컴퓨터
민감한 파일
보안 삭제
241
229
49
임의의 시스템
MAC 주소
최후에 알려진 IP 주소 : X
259
임의의 컴퓨터
봇네트
민감한 파일
262
240
241
238

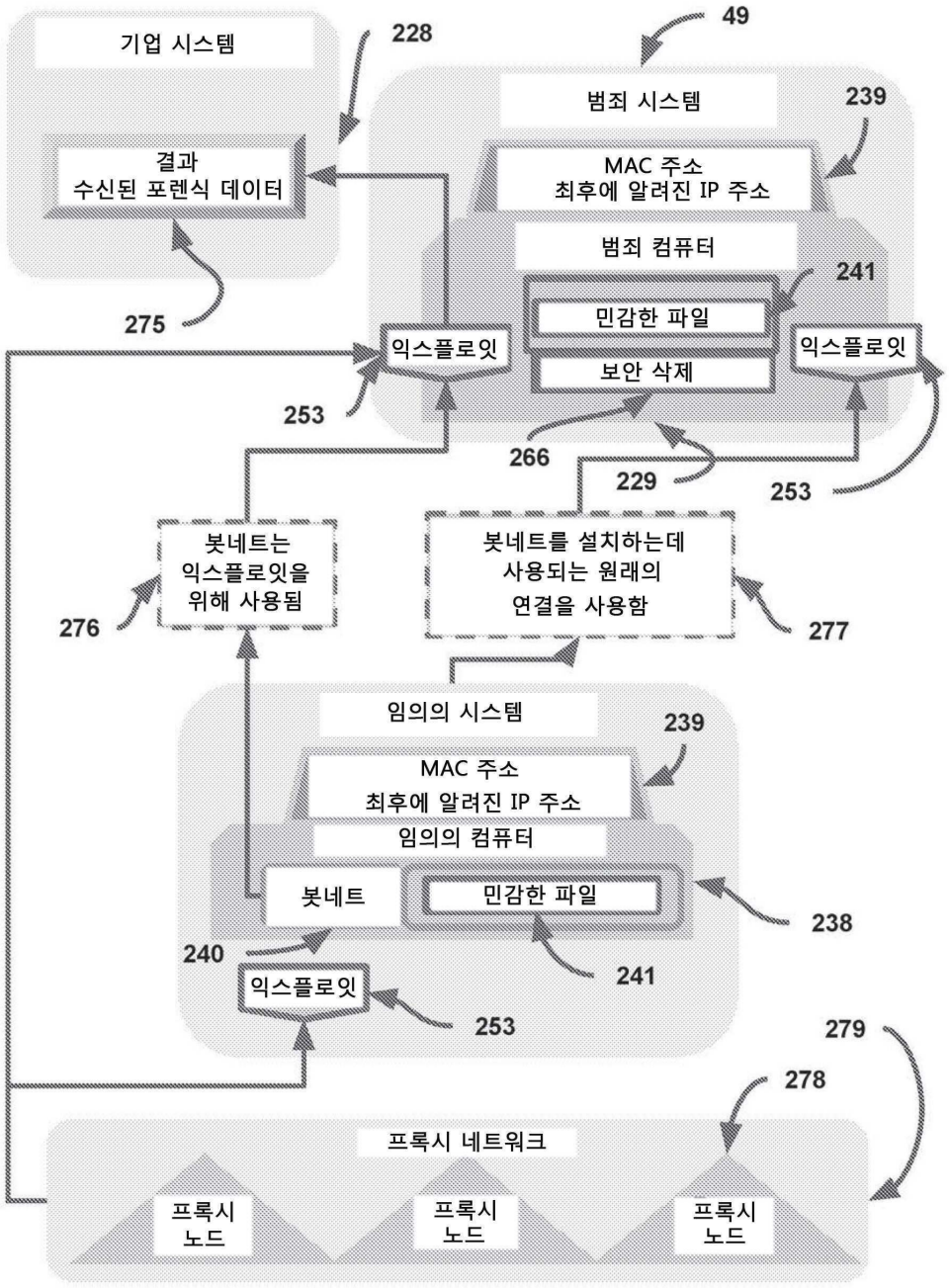
도면38


기업 시스템
범죄 컴퓨터로 의심되는 소프트웨어
+하드웨어
MAC 주소
최후에 알려진 IP 주소
포렌식 증거
식별
세부정보
민감한 파일
263
239
265
264
241
228
신뢰
플랫폼
10
범죄 시스템
239
MAC 주소
최후에 알려진 IP 주소
49
범죄 컴퓨터
241
익스플로잇
범죄 컴퓨터
익스플로잇
보안 삭제
253
253
229
266
소프트웨어
백도어
하드웨어
백도어
267
268
271
272
269
소프트웨어 공급업체
270
273
하드웨어 공급업체
274
범죄 컴퓨터로
의심되는
소프트웨어
설치된 소프트웨어
백도어의 리스트
범죄 컴퓨터로
의심되는
하드웨어
설치된 하드웨어
백도어의 리스트
익스플로잇
익스플로잇
253
253

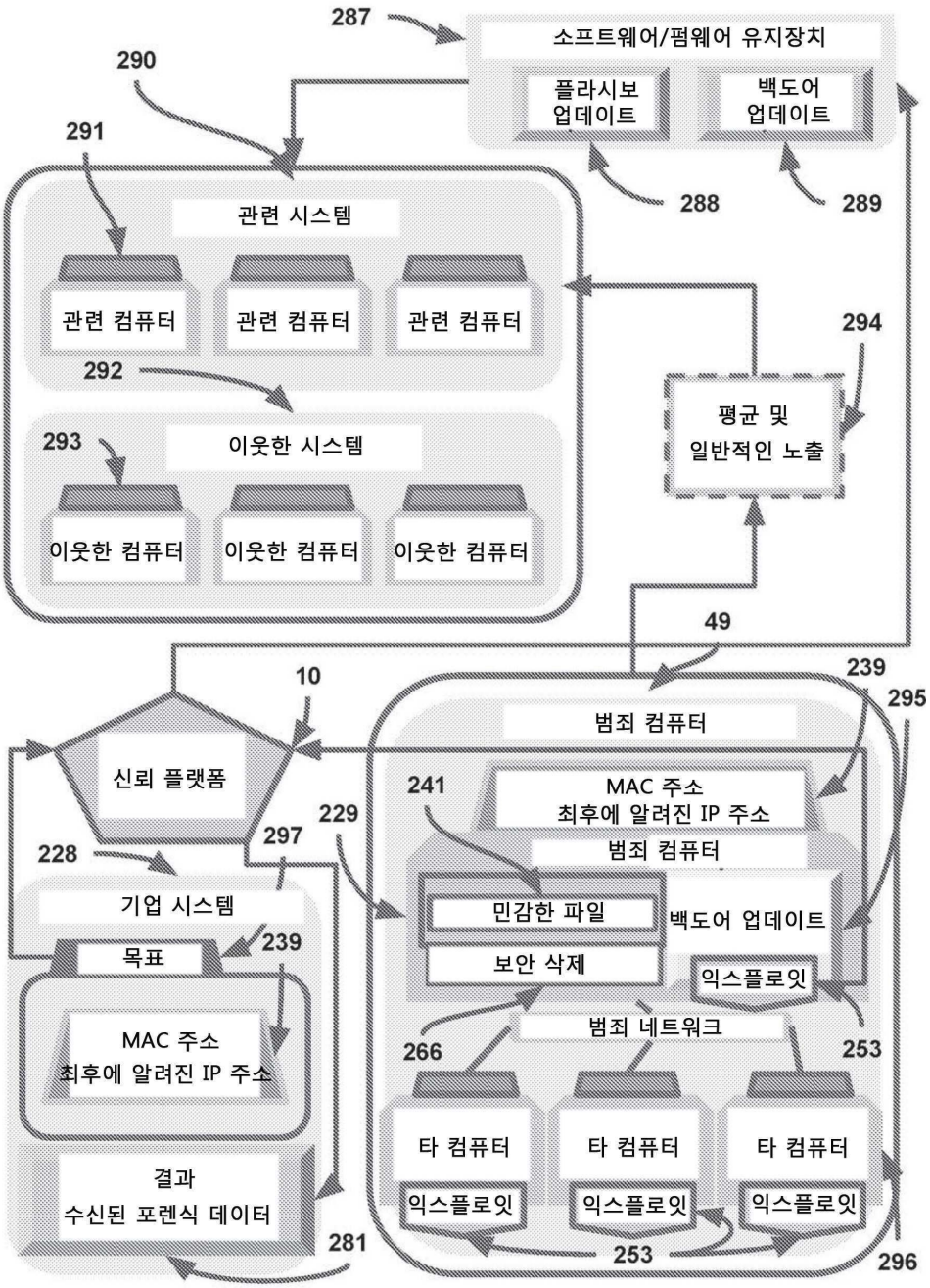
도면39


280
281
기업 시스템
결과 : 수신된 포렌식 데이터
282
일반적인 익스플로잇
범죄 컴퓨터로
예상된
소프트웨어
+ 하드웨어
239
MAC 주소
최후에 알려진 IP 주소
285
284
익스플로잇
맞춤형
286
283
프록시 관리
279
278
프록시 네트워크
프록시
노드
프록시
노드
프록시
노드

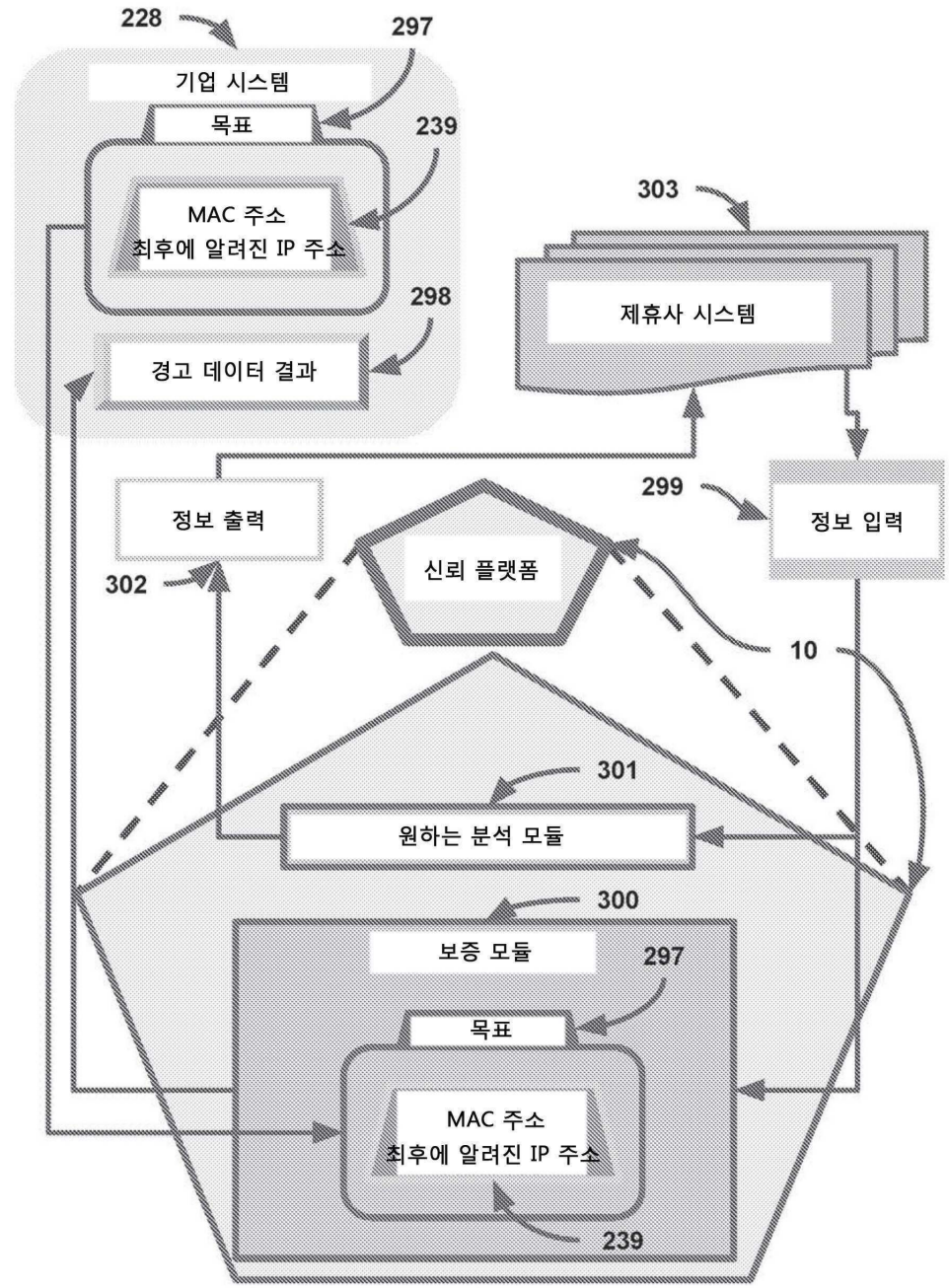
도면40


기업 시스템
228
49
범죄 시스템
239
결과
수신된 포렌식 데이터
MAC 주소
최후에 알려진 IP 주소
범죄 컴퓨터
241
민감한 파일
보안 삭제
275
익스플로잇
익스플로잇
253
266
229
253
봇네트는
익스플로잇을
위해 사용됨
봇네트를 설치하는데
사용되는 원래의
연결을 사용함
276
277
임의의 시스템
239
MAC 주소
최후에 알려진 IP 주소
임의의 컴퓨터
봇네트
민감한 파일
238
240
241
익스플로잇
253
279
278
프록시 네트워크
프록시
노드
프록시
노드
프록시
노드

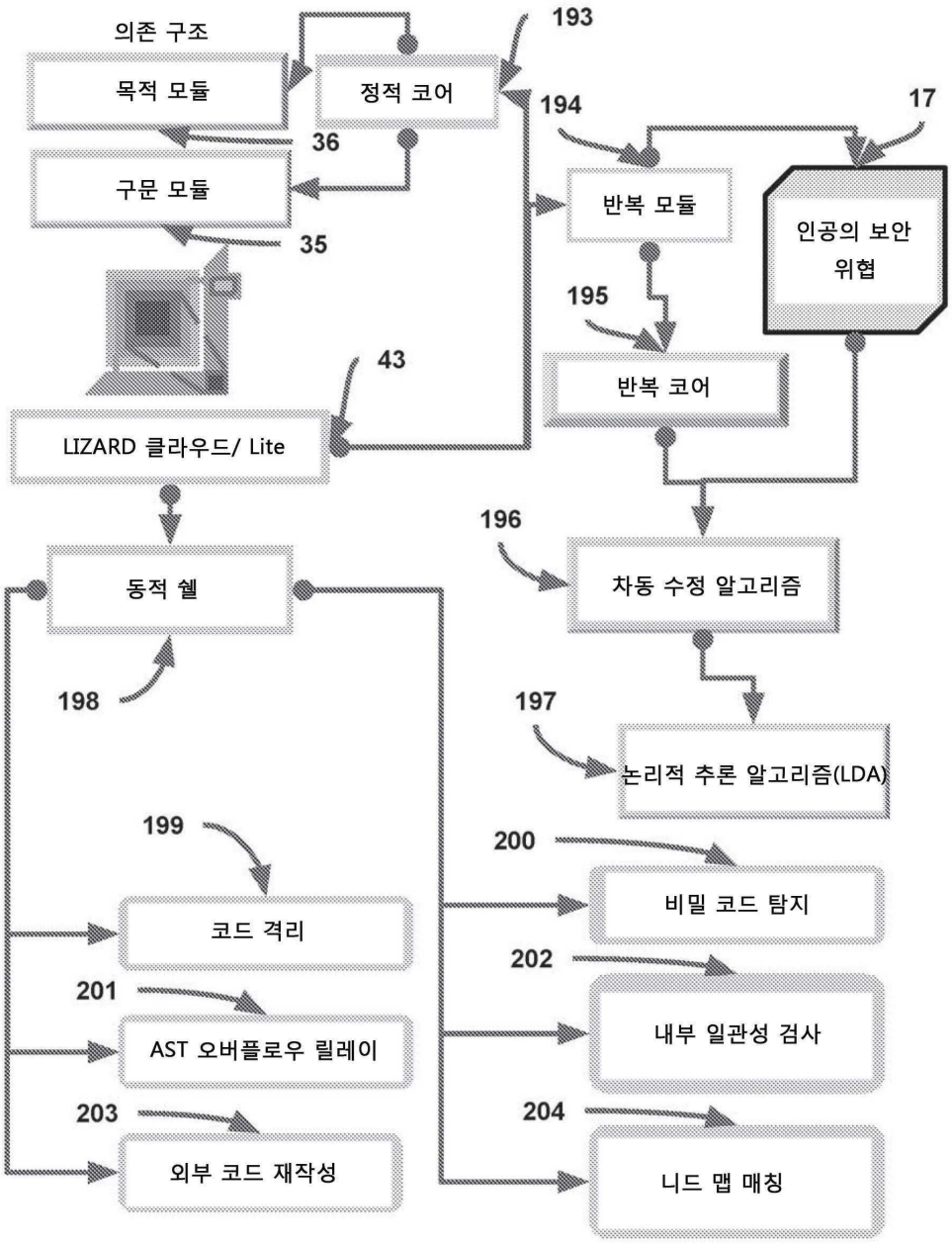
도면41


287
소프트웨어/펌웨어 유지장치
290
플라시보 업데이트
백도어 업데이트
291
288
289
관련 시스템
관련 컴퓨터
관련 컴퓨터
관련 컴퓨터
294
292
평균 및 일반적인 노출
293
이웃한 시스템
이웃한 컴퓨터
이웃한 컴퓨터
이웃한 컴퓨터
49
10
239
295
범죄 컴퓨터
신뢰 플랫폼
241
MAC 주소 최후에 알려진 IP 주소
229
297
228
범죄 컴퓨터
기업 시스템
민감한 파일
백도어 업데이트
239
목표
보안 삭제
익스플로잇
범죄 네트워크
MAC 주소 최후에 알려진 IP 주소
266
253
타 컴퓨터
타 컴퓨터
타 컴퓨터
결과 수신된 포렌식 데이터
익스플로잇
익스플로잇
익스플로잇
281
253
296

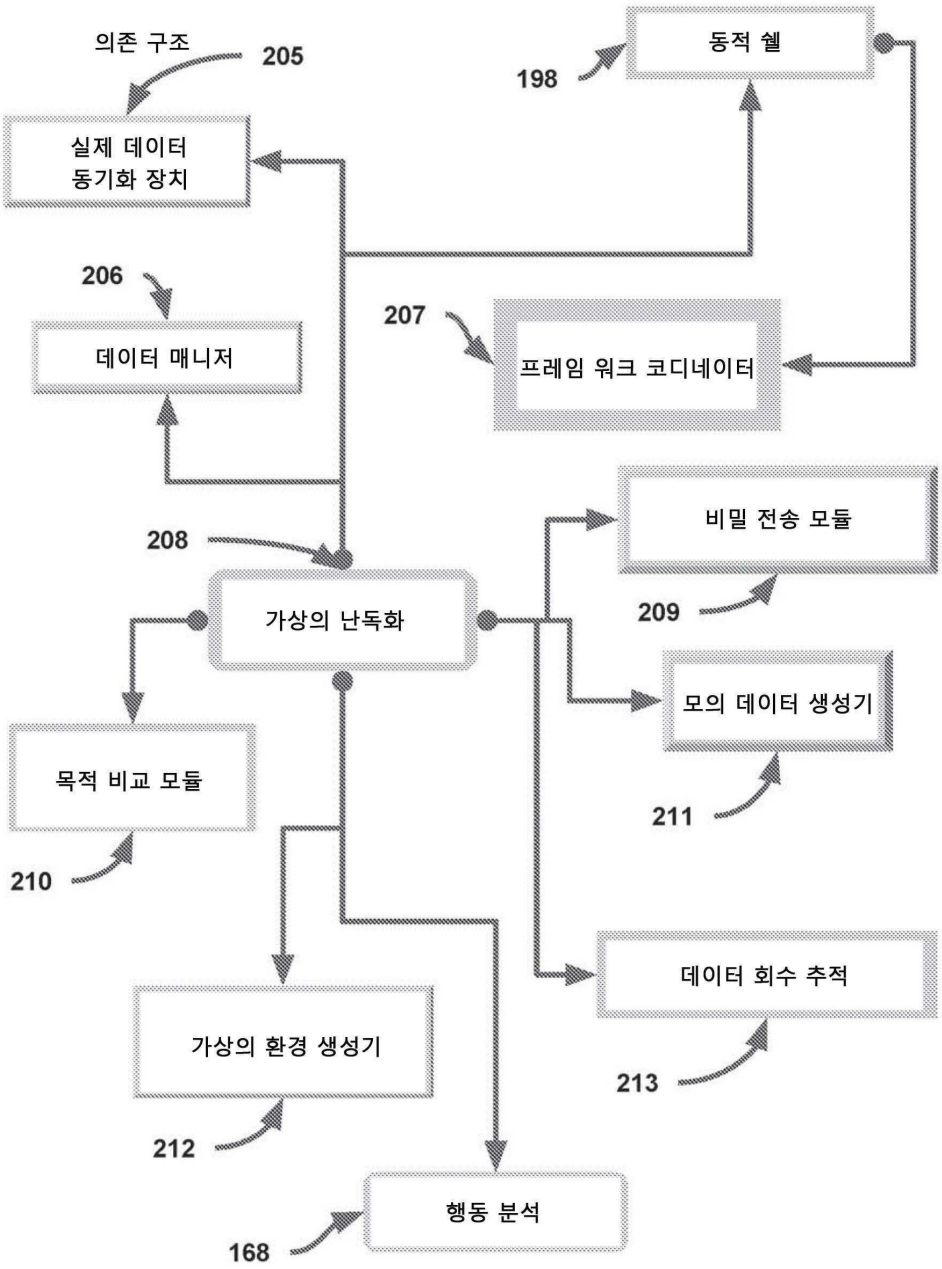
도면42


228
297
기업 시스템
목표
239
MAC 주소
최후에 알려진 IP 주소
298
경고 데이터 결과
303
제휴사 시스템
299
정보 입력
정보 출력
302
신뢰 플랫폼
10
301
원하는 분석 모듈
300
보증 모듈
297
목표
MAC 주소
최후에 알려진 IP 주소
239

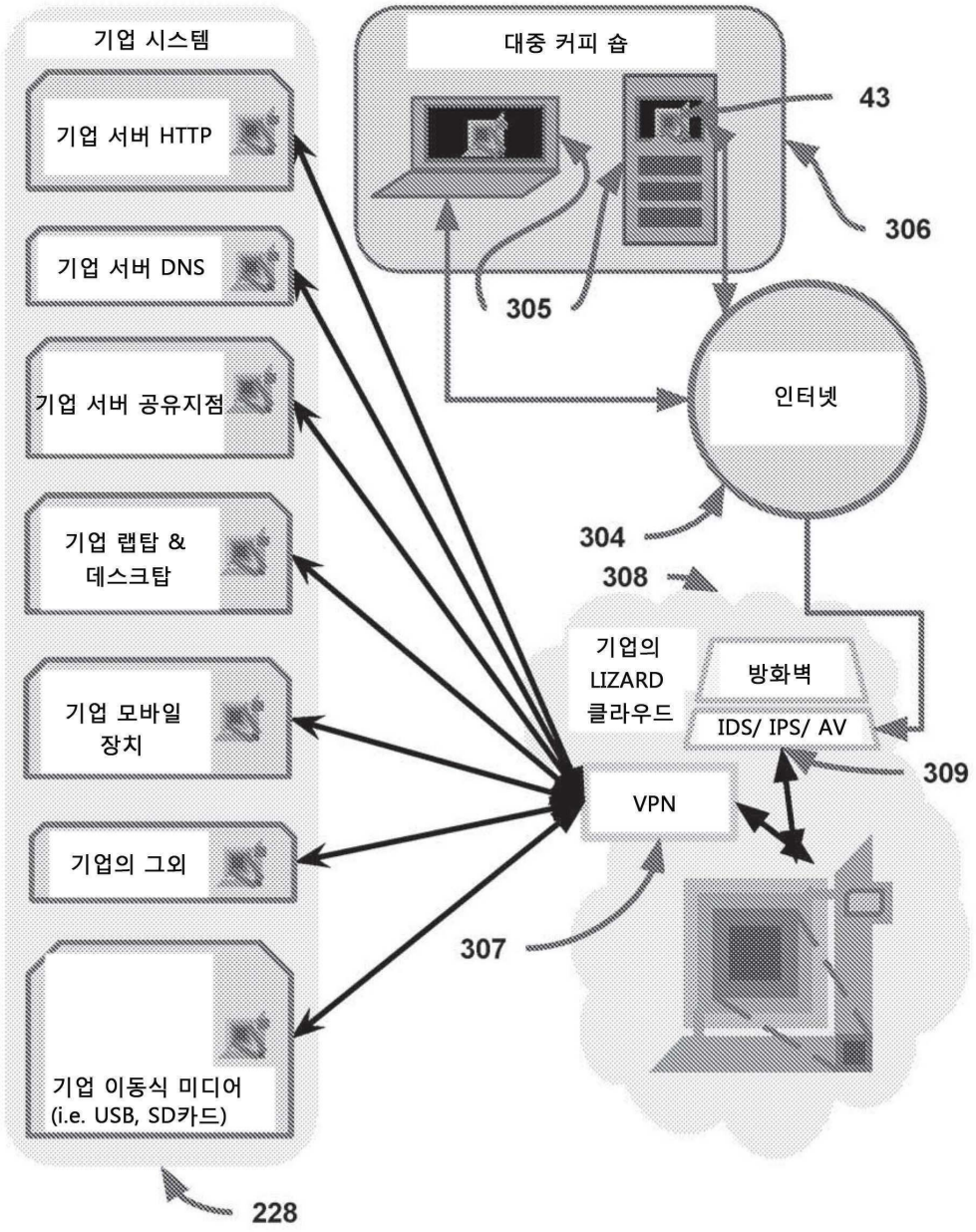
도면43


의존 구조
목적 모듈
정적 코어
193
194
17
36
구문 모듈
반복 모듈
인공의 보안 위협
35
195
43
반복 코어
LIZARD 클라우드/ Lite
196
동적 쉘
차동 수정 알고리즘
198
197
논리적 추론 알고리즘(LDA)
199
200
비밀 코드 탐지
코드 격리
202
201
AST 오버플로우 릴레이
내부 일관성 검사
203
204
외부 코드 재작성
니드 맵 매칭

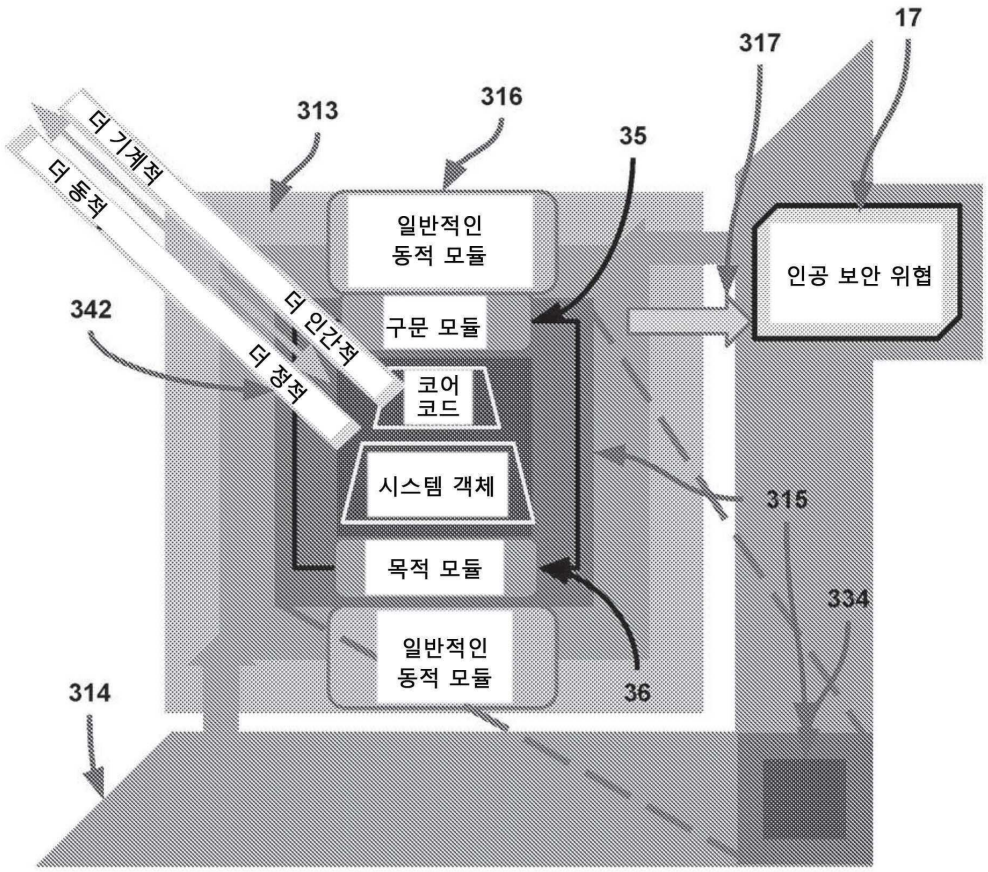
도면44


의존 구조
205
실제 데이터
동기화 장치
198
동적 쉘
206
데이터 매니저
207
프레임 워크 코디네이터
208
비밀 전송 모듈
209
가상의 난독화
모의 데이터 생성기
211
목적 비교 모듈
210
데이터 회수 추적
213
가상의 환경 생성기
212
행동 분석
168

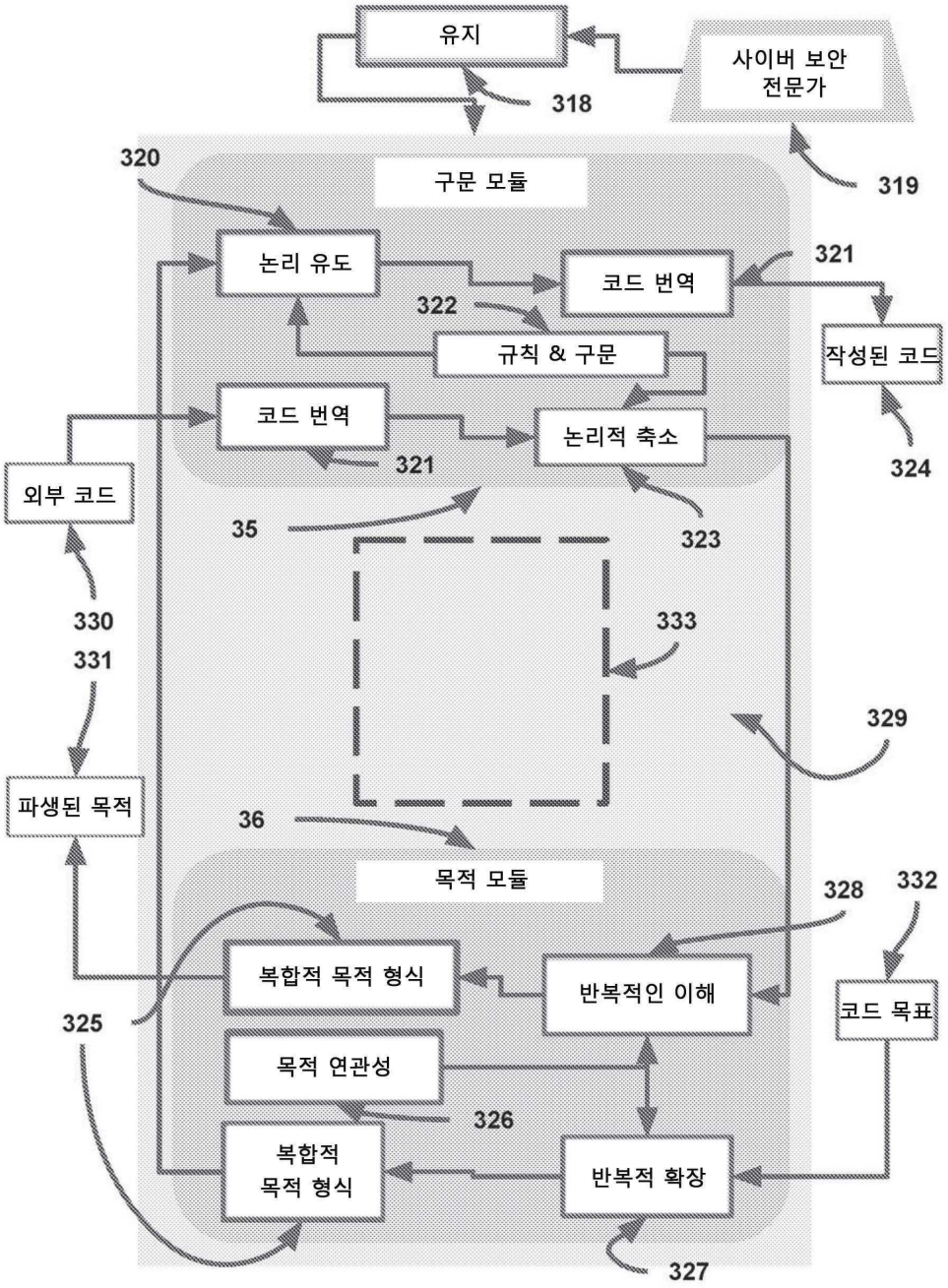
도면45


기업 시스템
기업 서버 HTTP
기업 서버 DNS
기업 서버 공유지점
기업 랩탑 &
데스크탑
기업 모바일
장치
기업의 그외
기업 이동식 미디어
(i.e. USB, SD카드)
228
대중 커피 숍
43
306
305
인터넷
304
308
기업의
LIZARD
클라우드
방화벽
IDS/ IPS/ AV
309
VPN
307

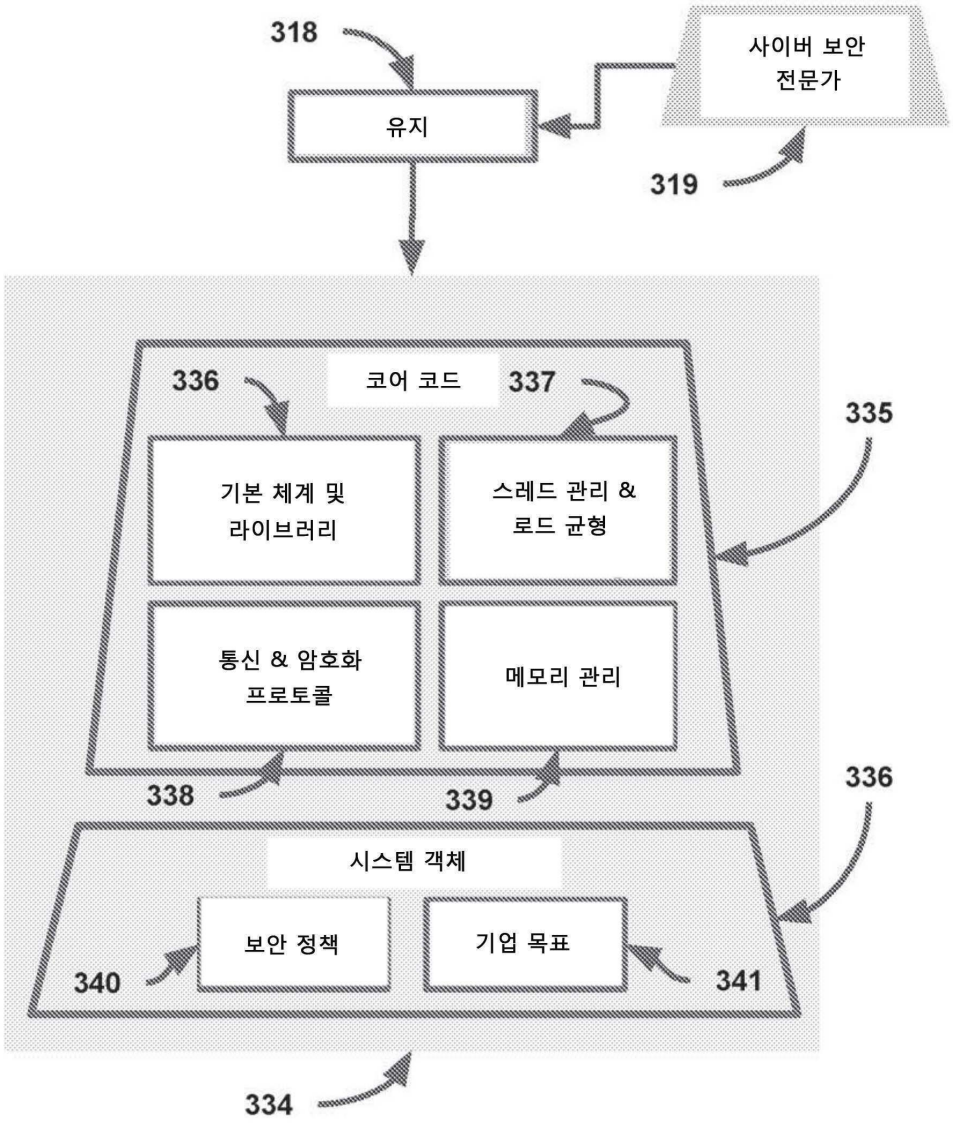
도면46


17
317
313
316
35
더 기계적
더 동적
일반적인
동적 모듈
인공 보안 위협
342
더 인간적
구문 모듈
더 정적
코어
코드
시스템 객체
315
목적 모듈
334
일반적인
동적 모듈
36
314

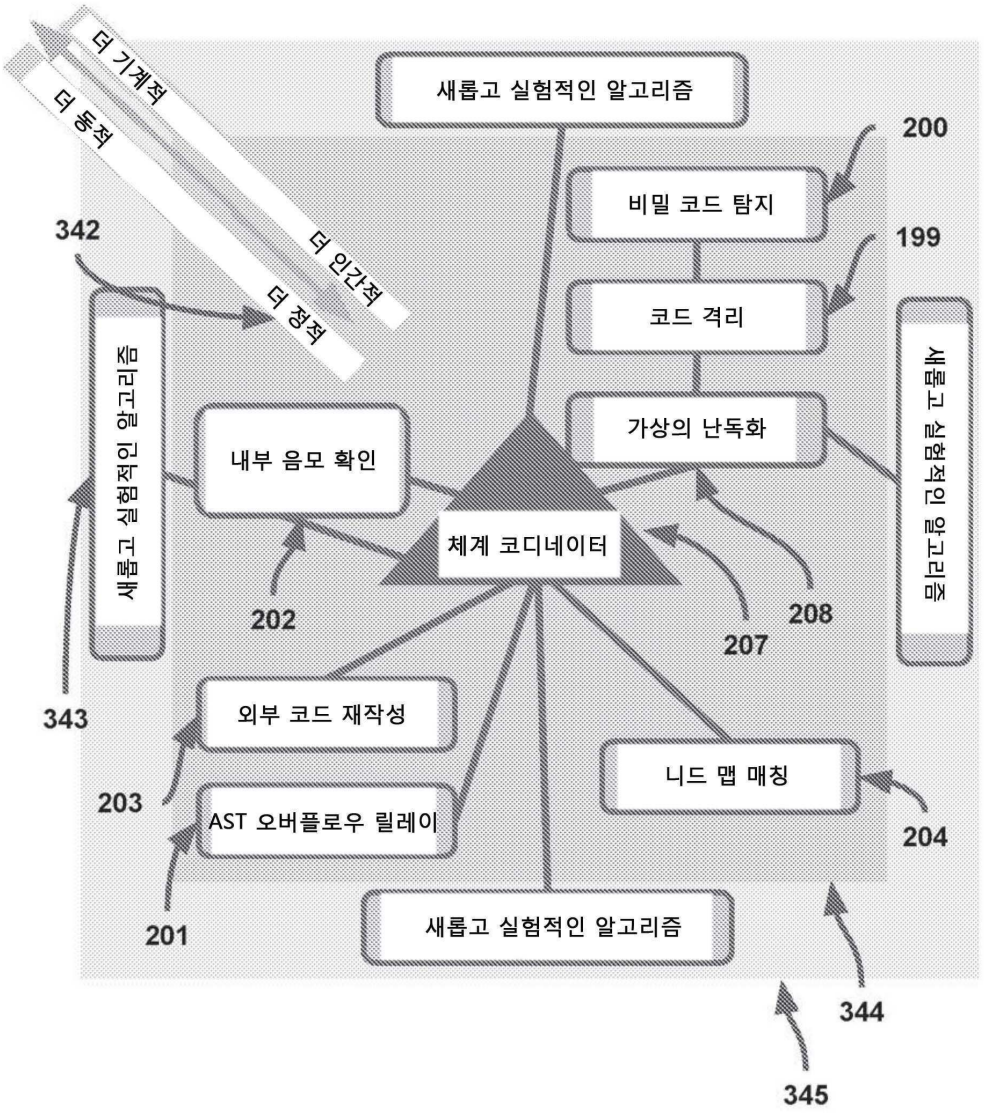
도면47


유지
318
사이버 보안
전문가
319
320
구문 모듈
논리 유도
코드 번역
321
322
규칙 & 구문
작성된 코드
코드 번역
논리적 축소
321
324
외부 코드
35
323
330
333
331
329
파생된 목적
36
목적 모듈
328
332
복합적 목적 형식
반복적인 이해
코드 목표
325
목적 연관성
326
복합적
목적 형식
반복적 확장
327

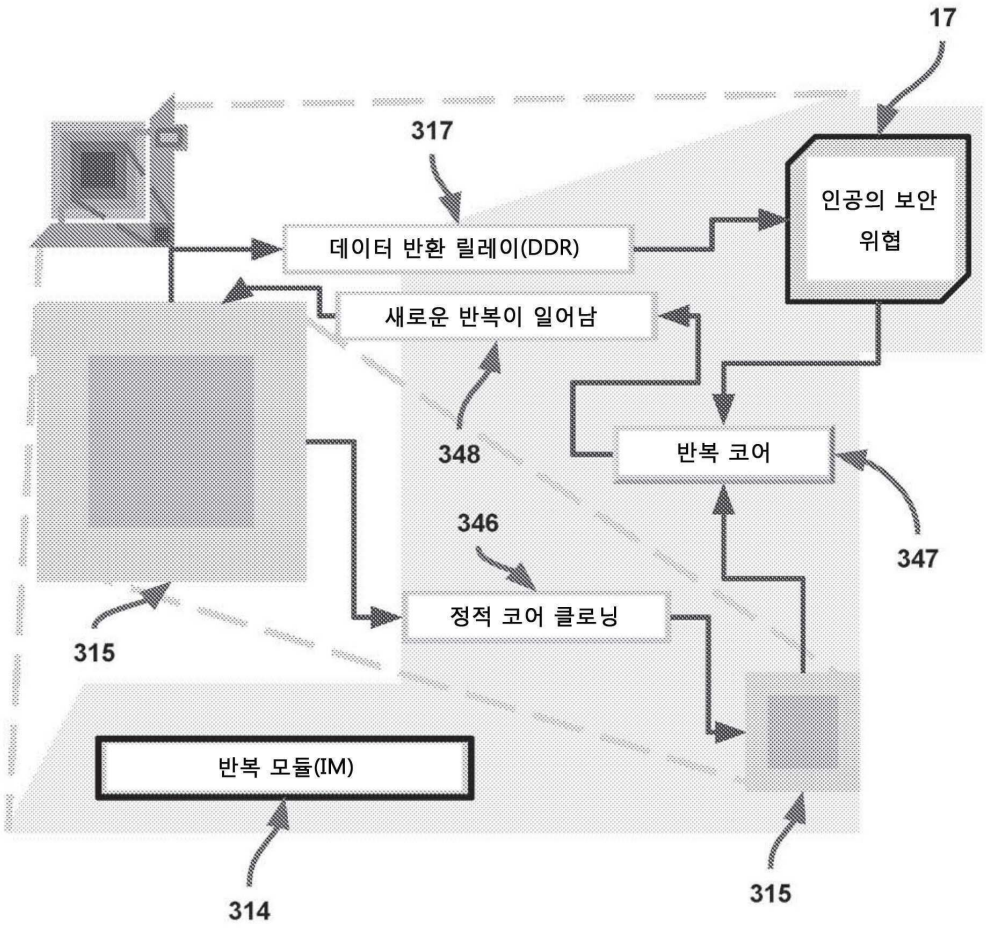
도면48


318
유지
사이버 보안
전문가
319
336
코어 코드
337
335
기본 체계 및
라이브러리
스레드 관리 &
로드 균형
통신 & 암호화
프로토콜
메모리 관리
338
339
336
시스템 객체
보안 정책
기업 목표
340
341
334

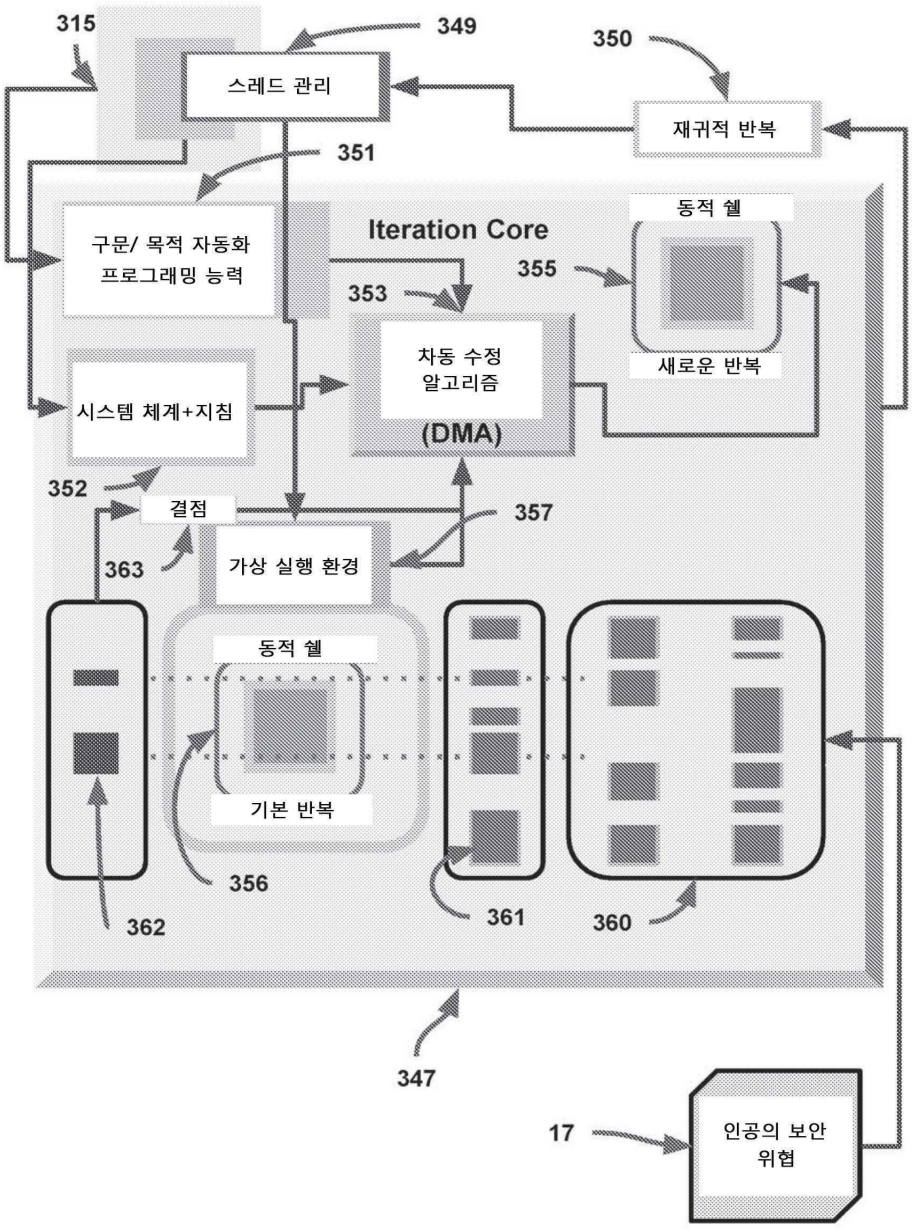
도면49


더 기계적
더 동적
더 인간적
더 정적
새롭고 실험적인 알고리즘
200
비밀 코드 탐지
199
코드 격리
342
새롭고 실험적인 알고리즘
가상의 난독화
새롭고 실험적인 알고리즘
내부 음모 확인
체계 코디네이터
202
207
208
343
외부 코드 재작성
니드 맵 매칭
203
AST 오버플로우 릴레이
204
201
새롭고 실험적인 알고리즘
344
345

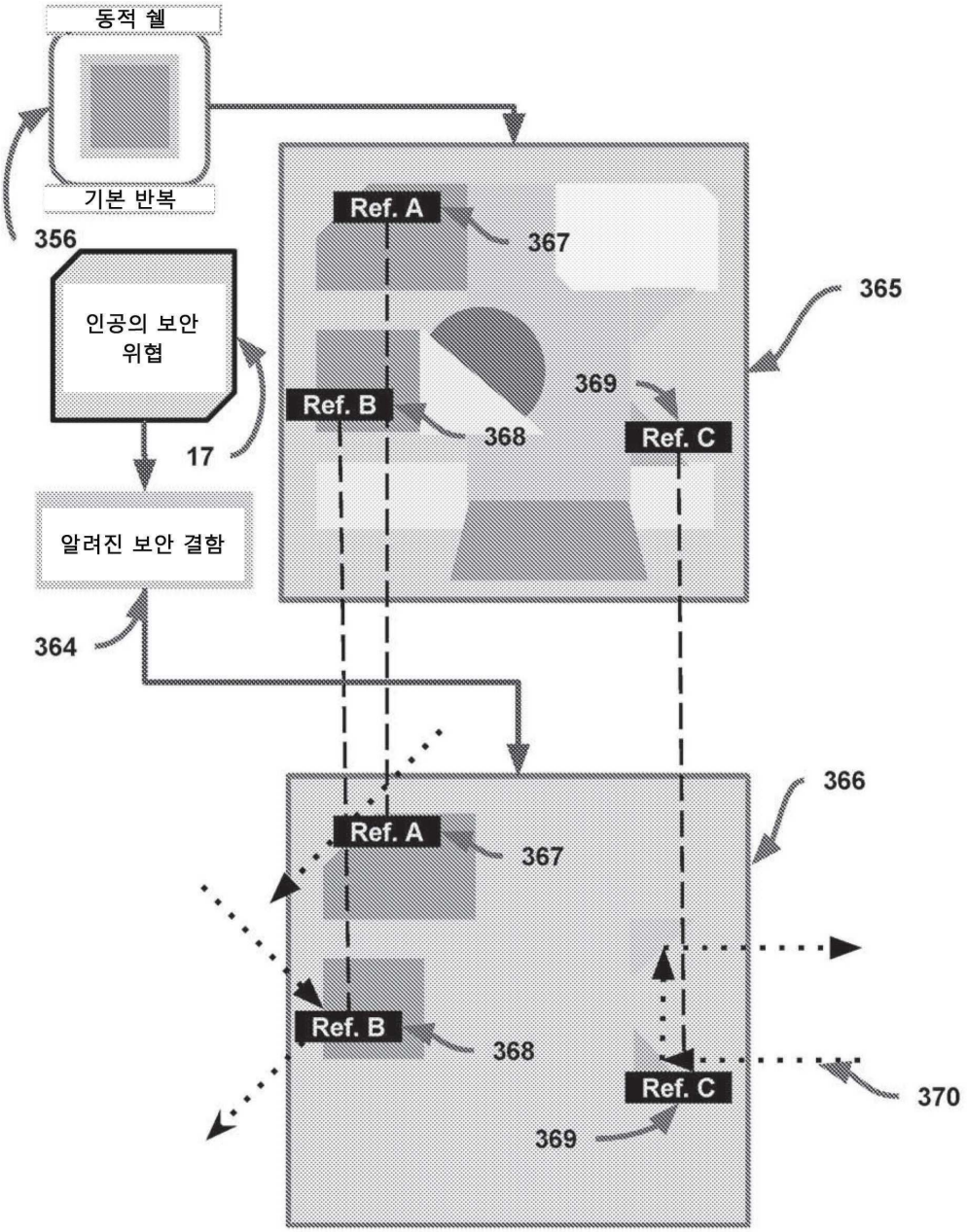
도면50


17
317
인공의 보안 위협
데이터 반환 릴레이(DDR)
새로운 반복이 일어남
348
반복 코어
346
347
315
정적 코어 클로닝
반복 모듈(IM)
314
315

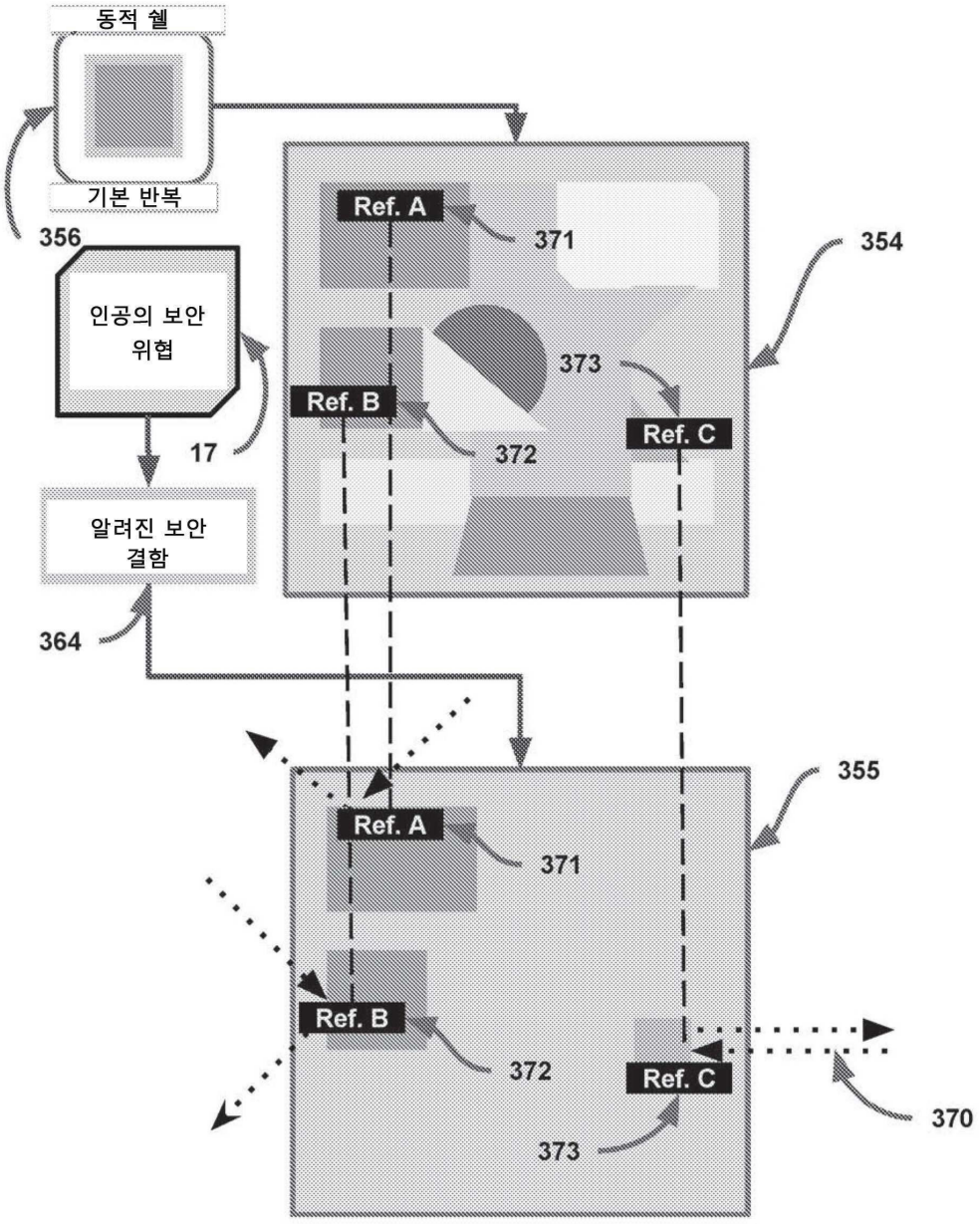
도면51


315
349
350
스레드 관리
재귀적 반복
351
구문/ 목적 자동화
프로그래밍 능력
Iteration Core
353
355
동적 쉘
새로운 반복
차동 수정
알고리즘
(DMA)
시스템 체계+지침
352
결점
357
363
가상 실행 환경
동적 쉘
기본 반복
356
362
361
360
347
17
인공의 보안
위협

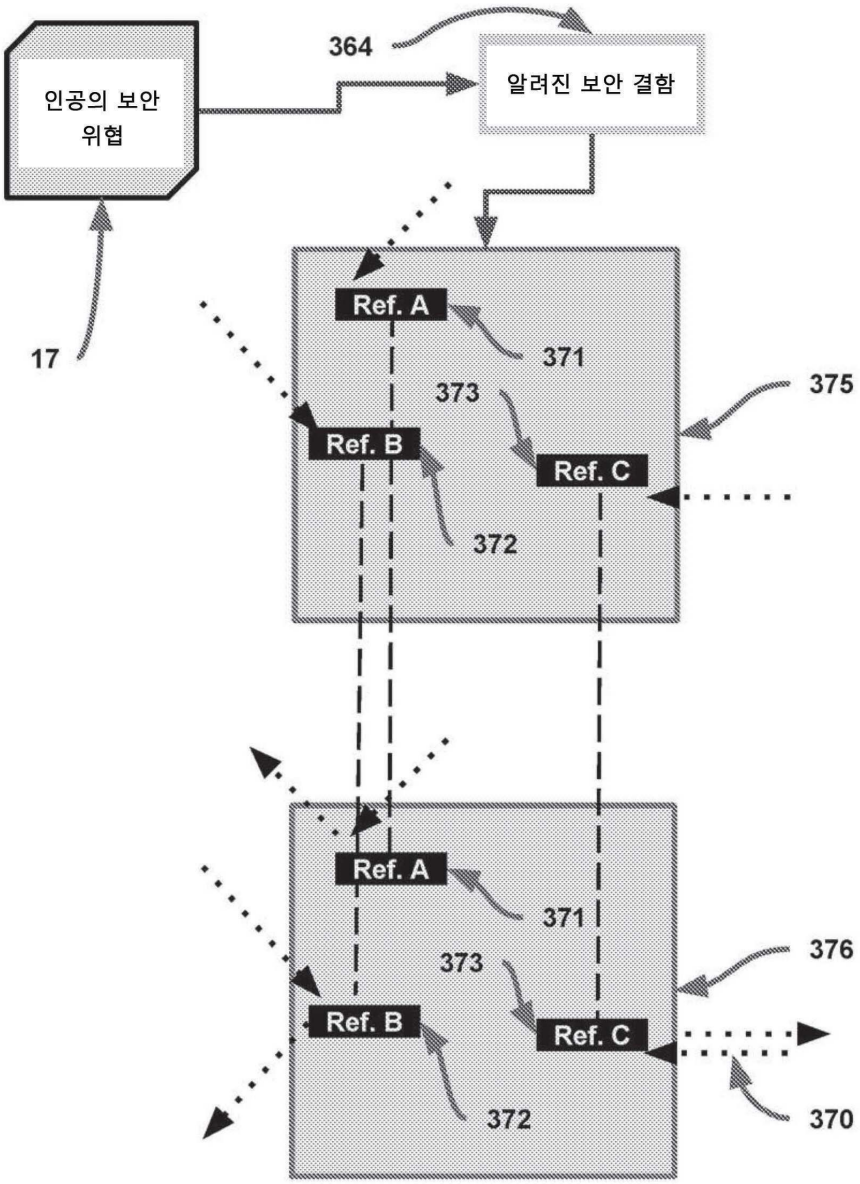
도면52


동적 쉘
기본 반복
356
인공의 보안 위협
17
알려진 보안 결함
364
Ref. A
367
365
Ref. B
368
369
Ref. C
366
Ref. A
367
Ref. B
368
Ref. C
369
370

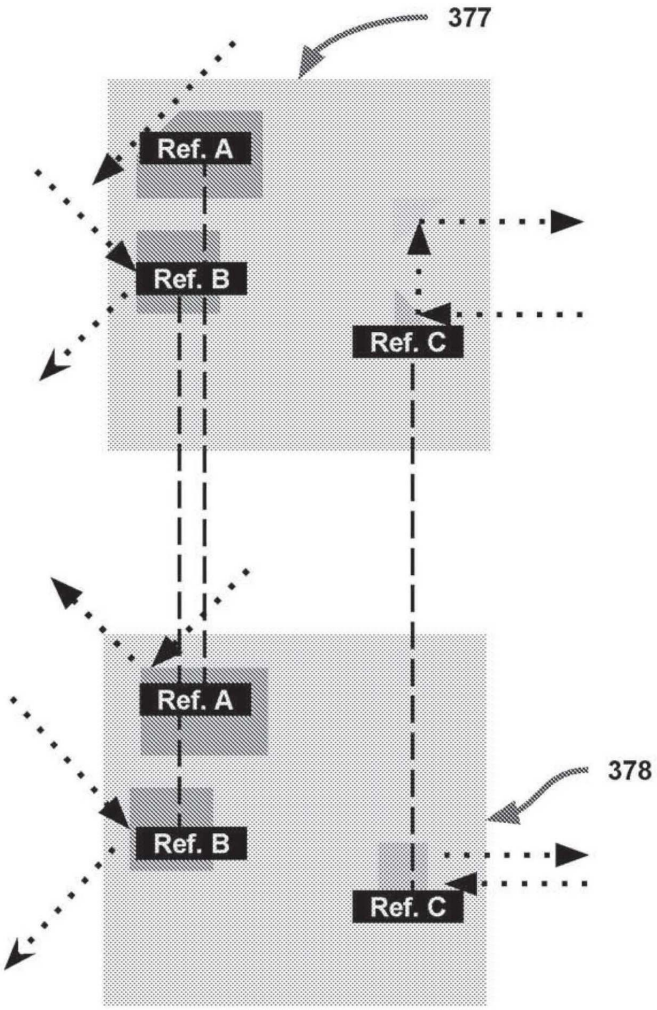
도면53


동적 쉘
기본 반복
356
인공의 보안 위협
17
알려진 보안 결함
364
Ref. A
371
354
373
Ref. B
372
Ref. C
355
Ref. A
371
Ref. B
372
Ref. C
370
373

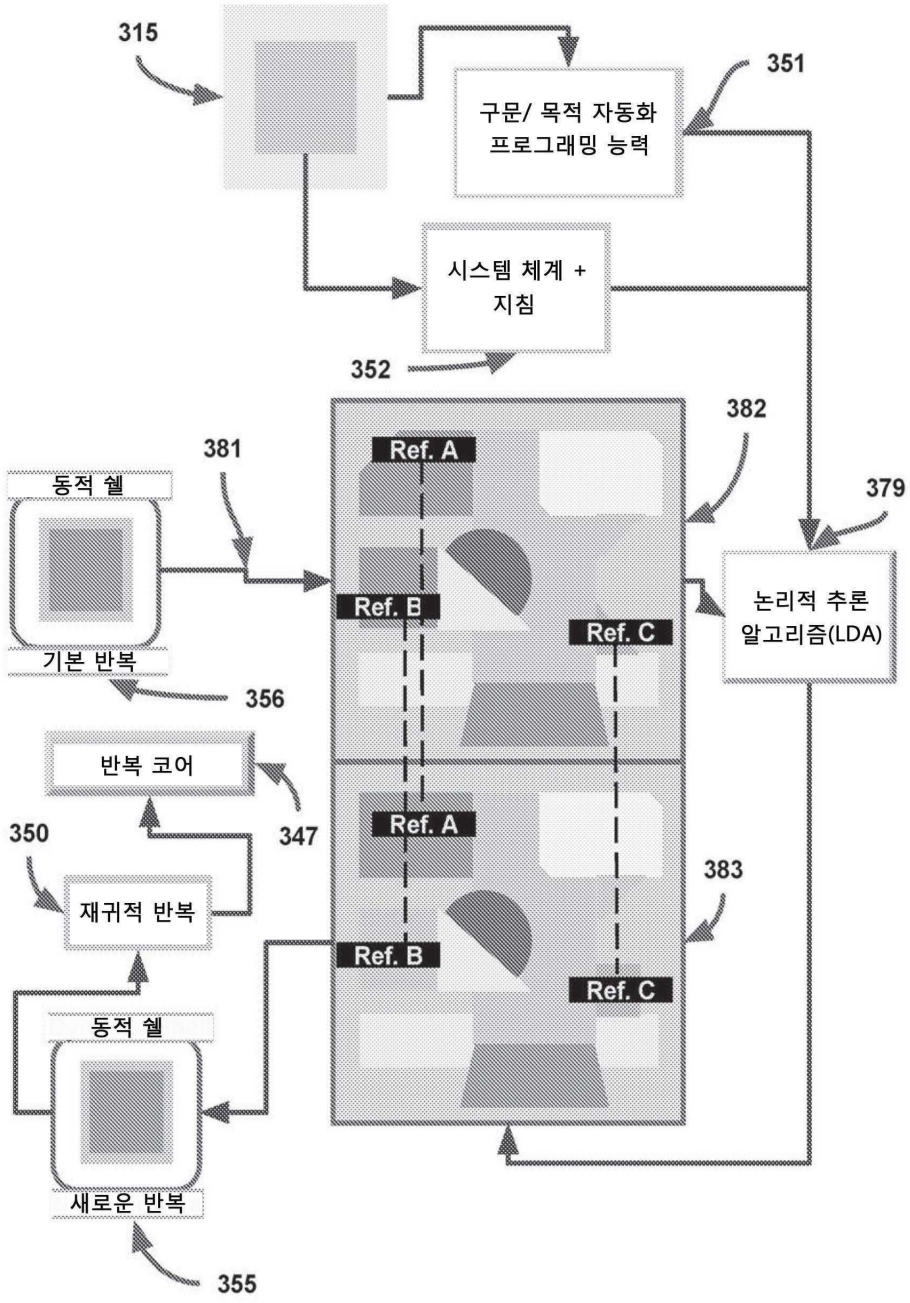
도면54


364
인공의 보안 위협
알려진 보안 결함
17
Ref. A
371
373
375
Ref. B
Ref. C
372
Ref. A
371
373
376
Ref. B
Ref. C
372
370

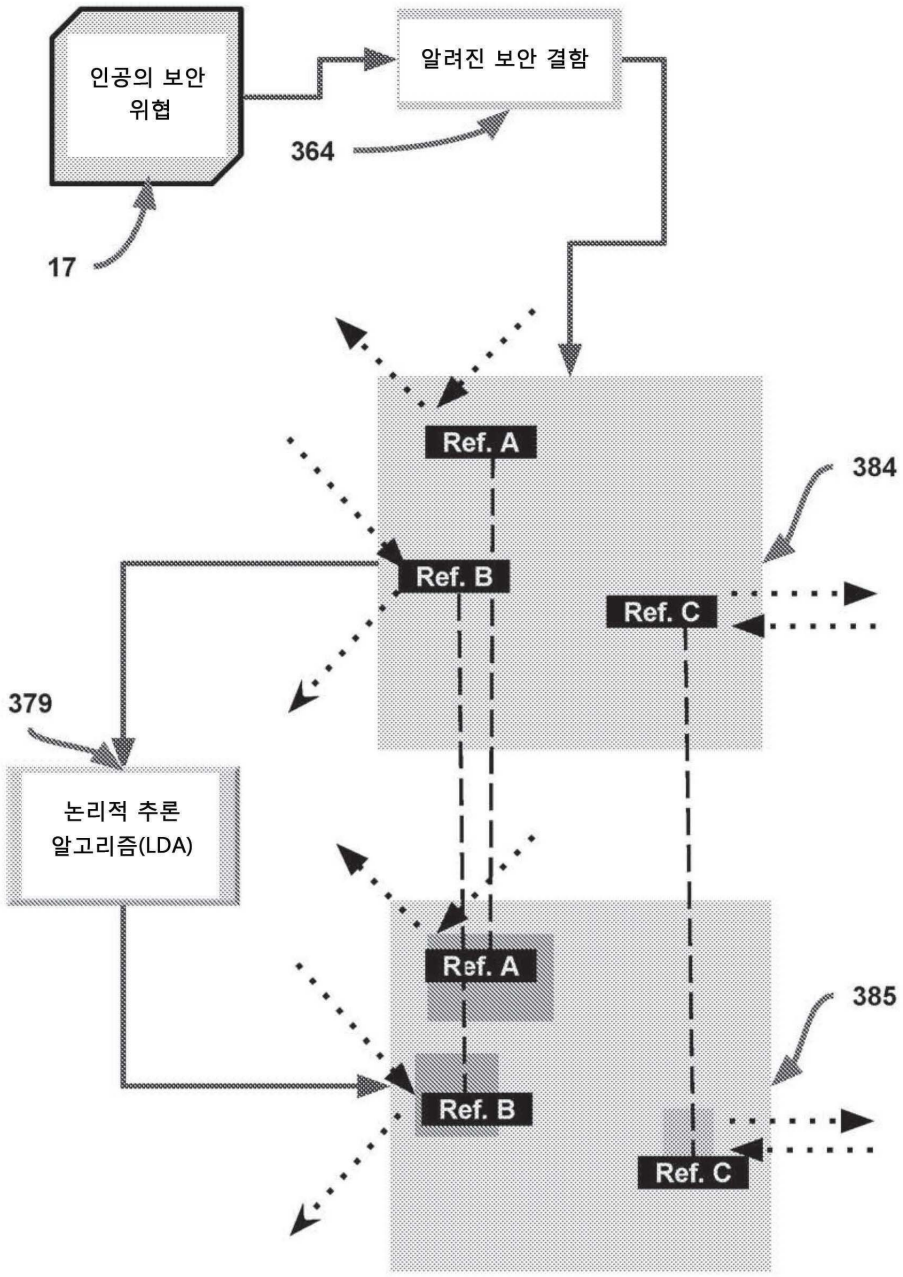
도면55



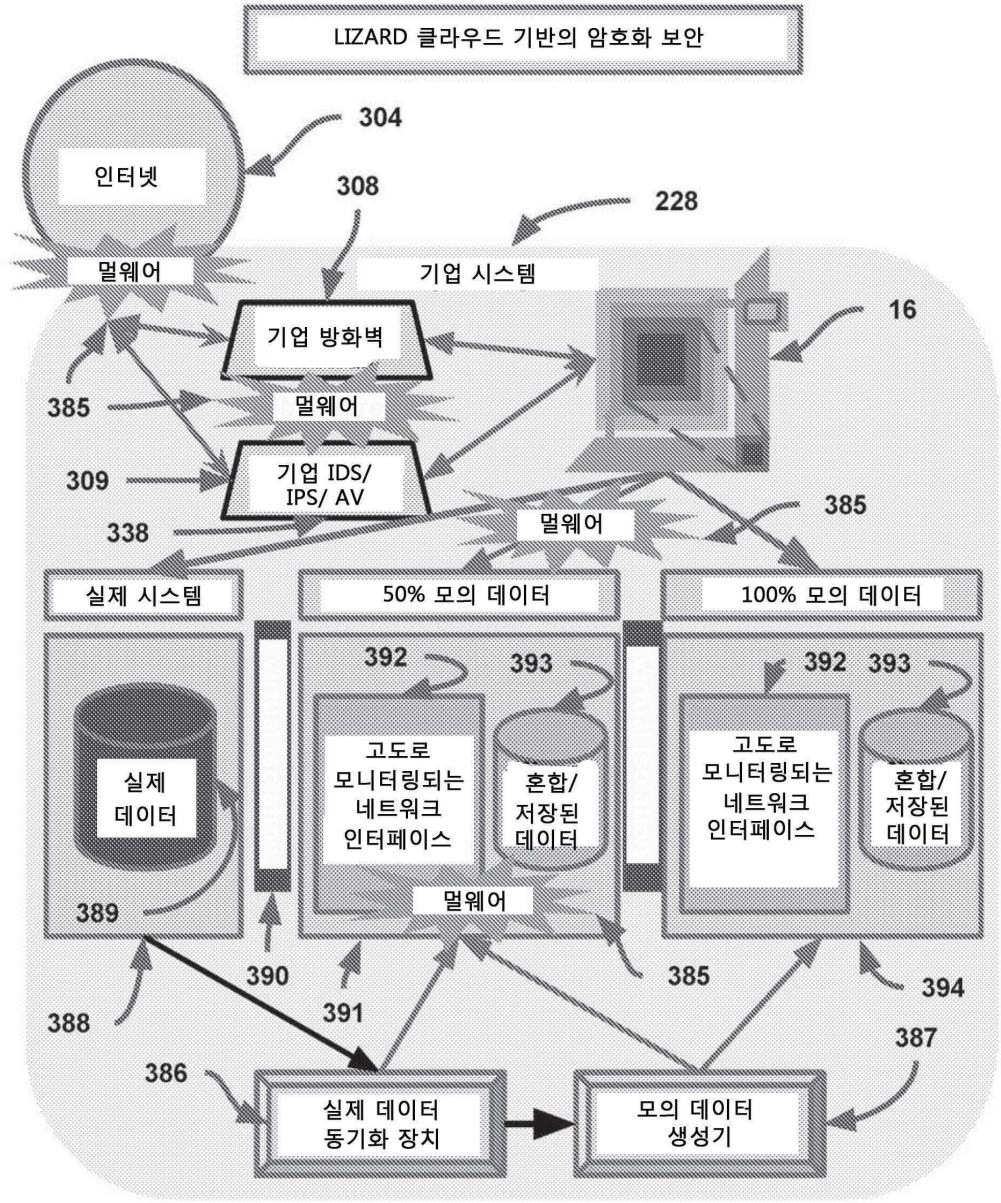
도면56


315
351
구문/ 목적 자동화
프로그래밍 능력
시스템 체계 +
지침
352
382
381
379
Ref. A
동적 쉘
Ref. B
Ref. C
논리적 추론
알고리즘(LDA)
기본 반복
356
반복 코어
347
350
Ref. A
383
재귀적 반복
Ref. B
Ref. C
동적 쉘
새로운 반복
355

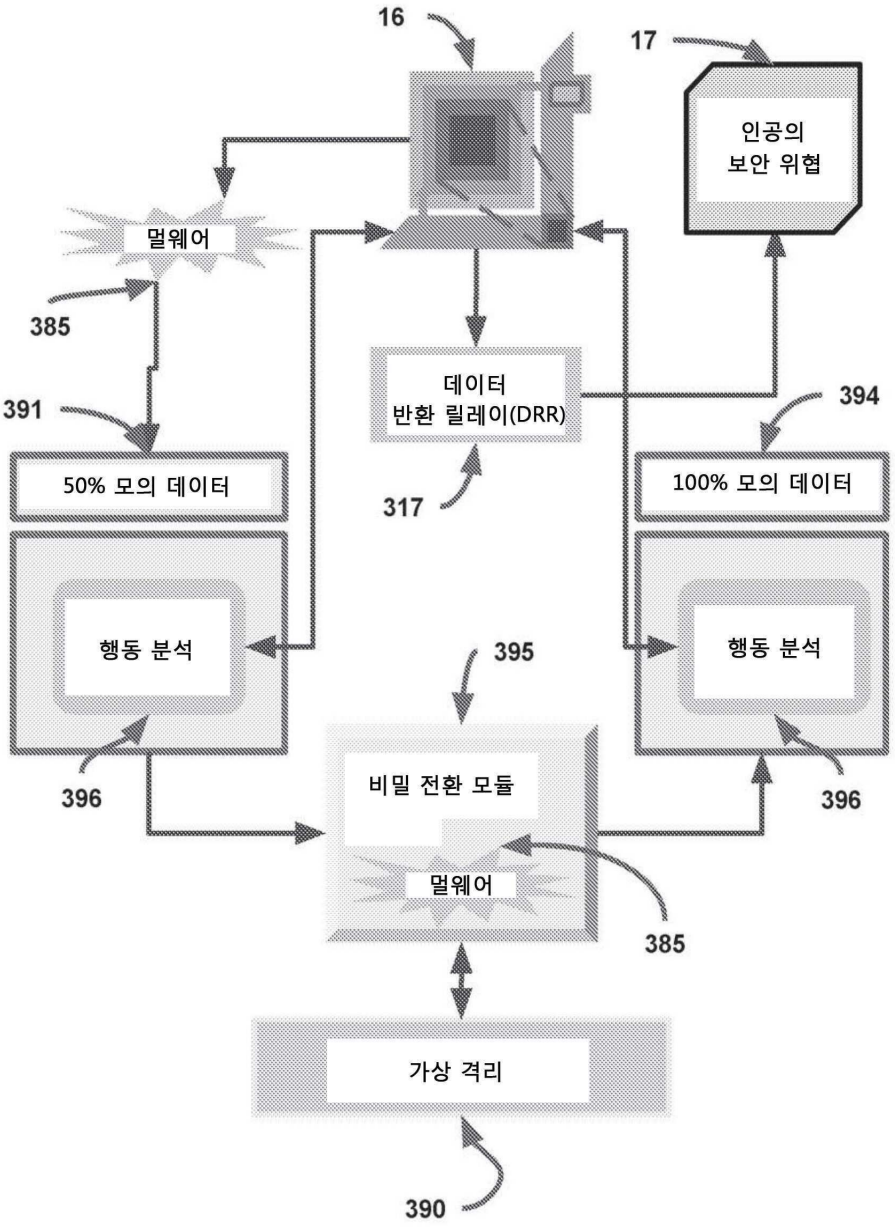
도면57


인공의 보안 위협
알려진 보안 결함
364
17
Ref. A
384
Ref. B
Ref. C
379
논리적 추론 알고리즘(LDA)
Ref. A
385
Ref. B
Ref. C

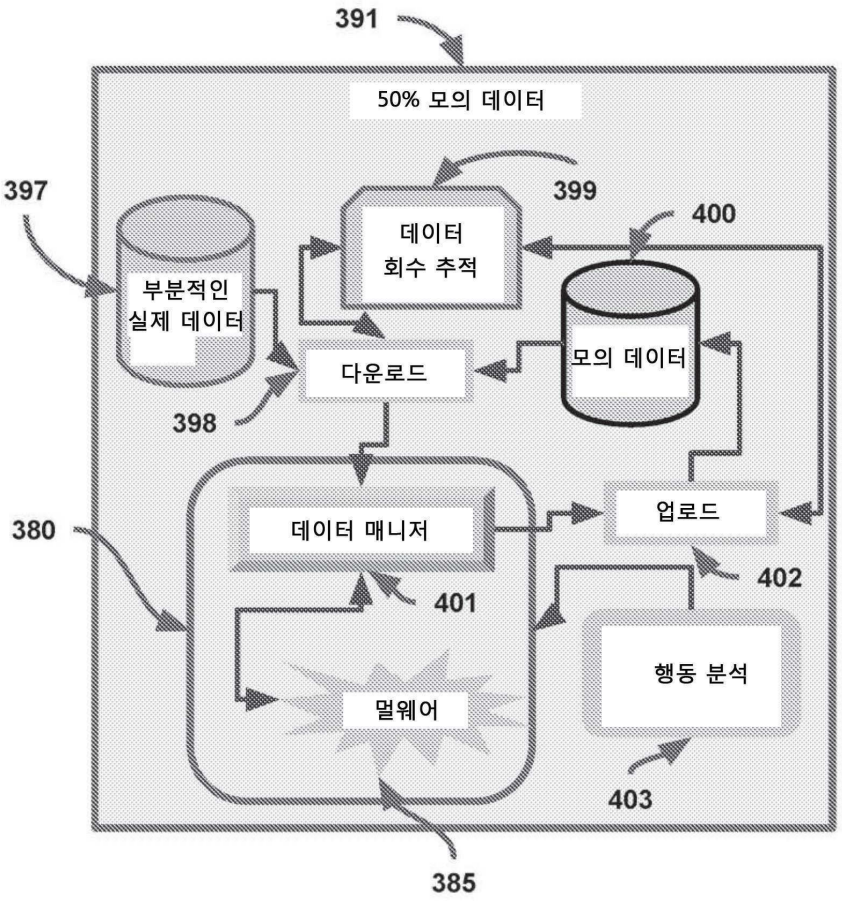
도면58


LIZARD 클라우드 기반의 암호화 보안
304
인터넷
308
228
멀웨어
기업 시스템
16
기업 방화벽
385
멀웨어
309
기업 IDS/
IPS/ AV
385
338
멀웨어
실제 시스템
50% 모의 데이터
100% 모의 데이터
392
393
392
393
실제
데이터
고도로
모니터링되는
네트워크
인터페이스
혼합/
저장된
데이터
고도로
모니터링되는
네트워크
인터페이스
혼합/
저장된
데이터
389
멀웨어
390
385
394
388
391
387
386
실제 데이터
동기화 장치
모의 데이터
생성기

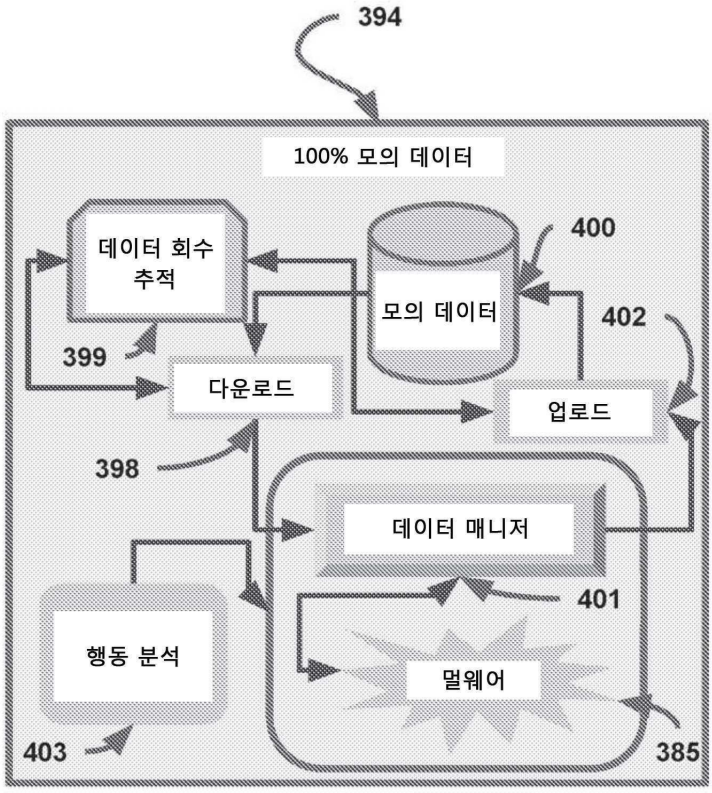
도면59


16
17
인공의
보안 위협
멀웨어
385
391
데이터
반환 릴레이(DRR)
394
317
50% 모의 데이터
100% 모의 데이터
행동 분석
395
행동 분석
396
396
비밀 전환 모듈
멀웨어
385
가상 격리
390

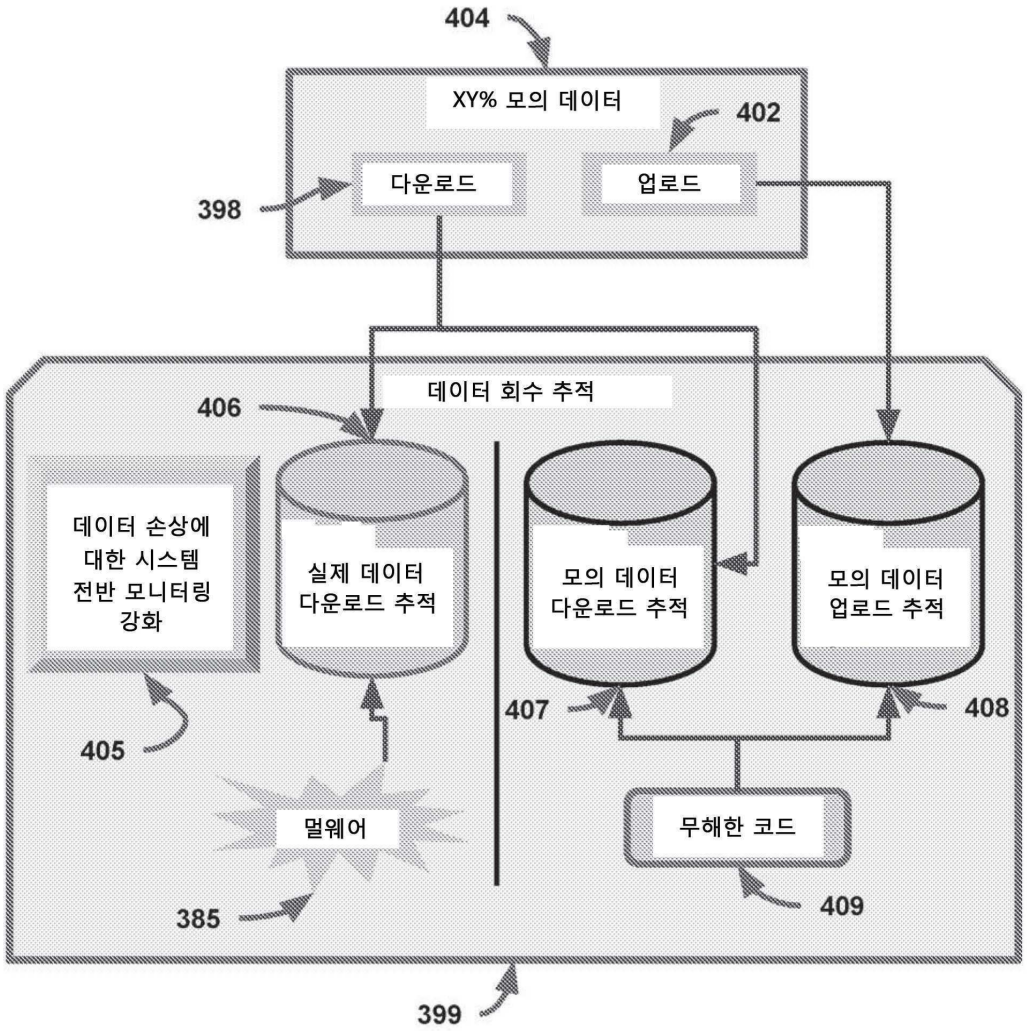
도면60


391
50% 모의 데이터
399
397
데이터
회수 추적
400
부분적인
실제 데이터
다운로드
모의 데이터
398
380
데이터 매니저
업로드
402
401
행동 분석
멀웨어
403
385

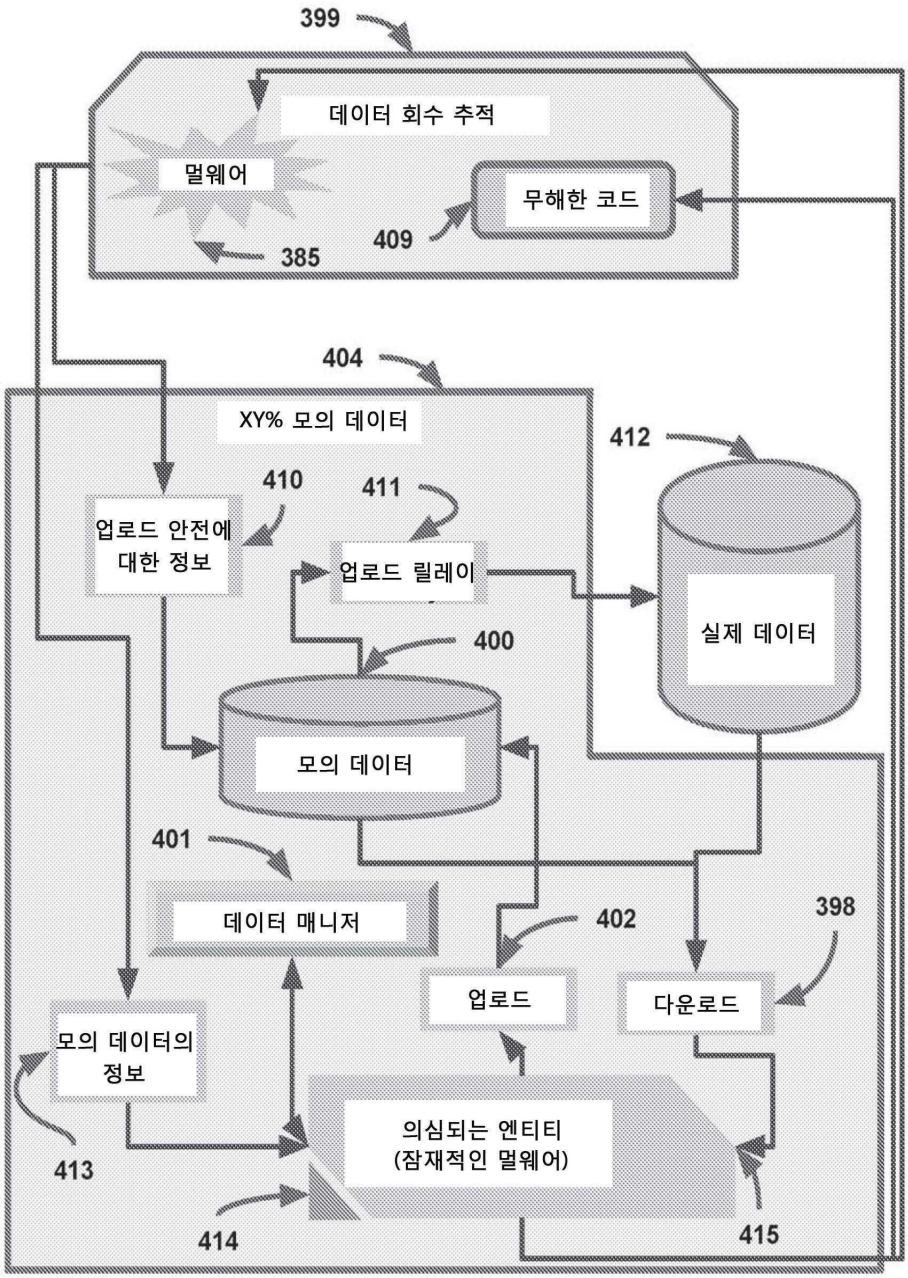
도면61


394
100% 모의 데이터
데이터 회수
추적
400
모의 데이터
402
399
다운로드
업로드
398
데이터 매니저
401
행동 분석
멀웨어
403
385

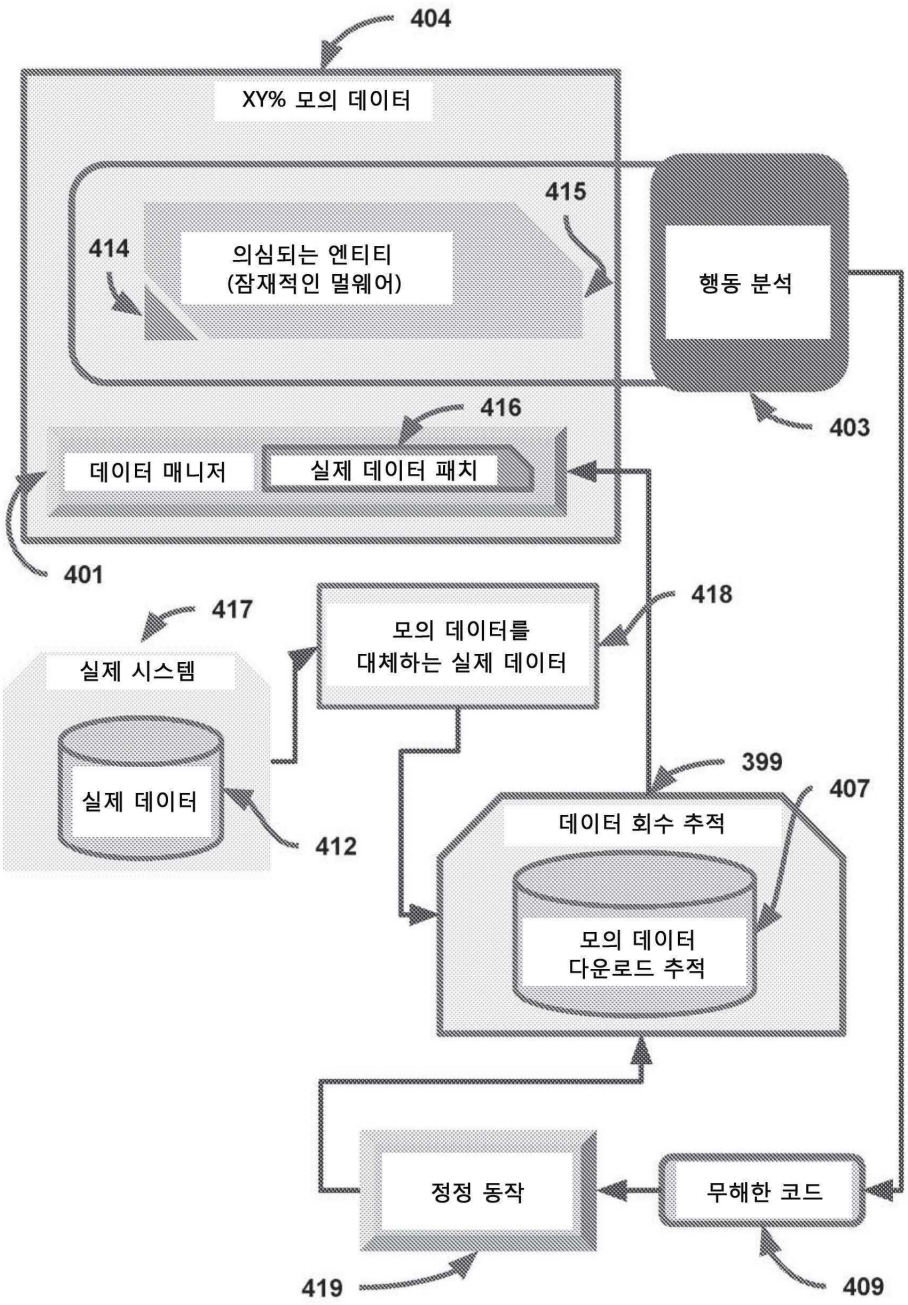
도면62


404
XY% 모의 데이터
402
다운로드
업로드
398
데이터 회수 추적
406
데이터 손상에 대한 시스템 전반 모니터링 강화
실제 데이터 다운로드 추적
모의 데이터 다운로드 추적
모의 데이터 업로드 추적
405
407
408
멀웨어
무해한 코드
385
409
399

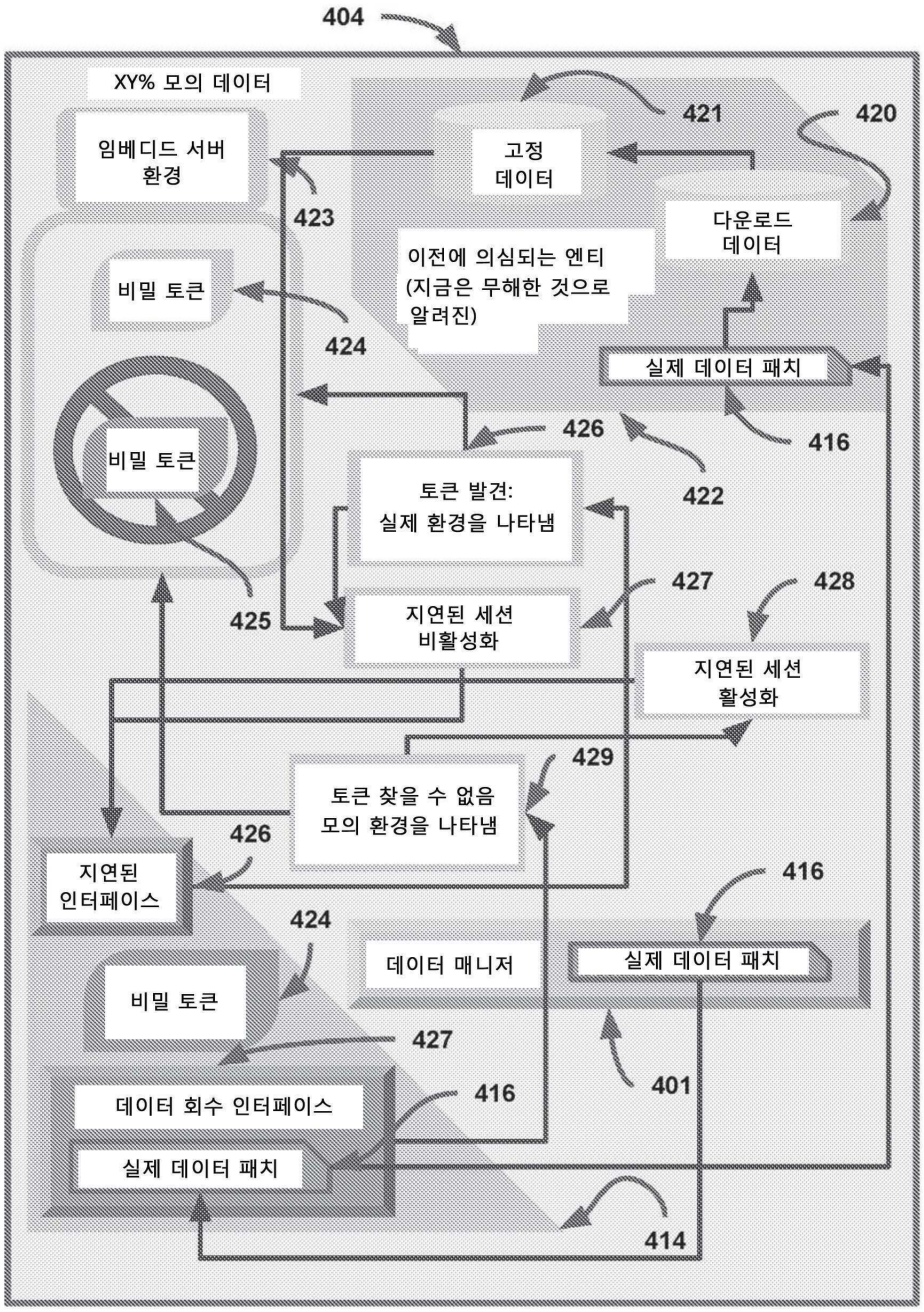
도면63


399
데이터 회수 추적
멀웨어
무해한 코드
409
385
404
XY% 모의 데이터
412
410
411
업로드 안전에 대한 정보
업로드 릴레이
실제 데이터
400
모의 데이터
401
데이터 매니저
402
398
업로드
다운로드
모의 데이터의 정보
의심되는 엔티티 (잠재적인 멀웨어)
413
414
415

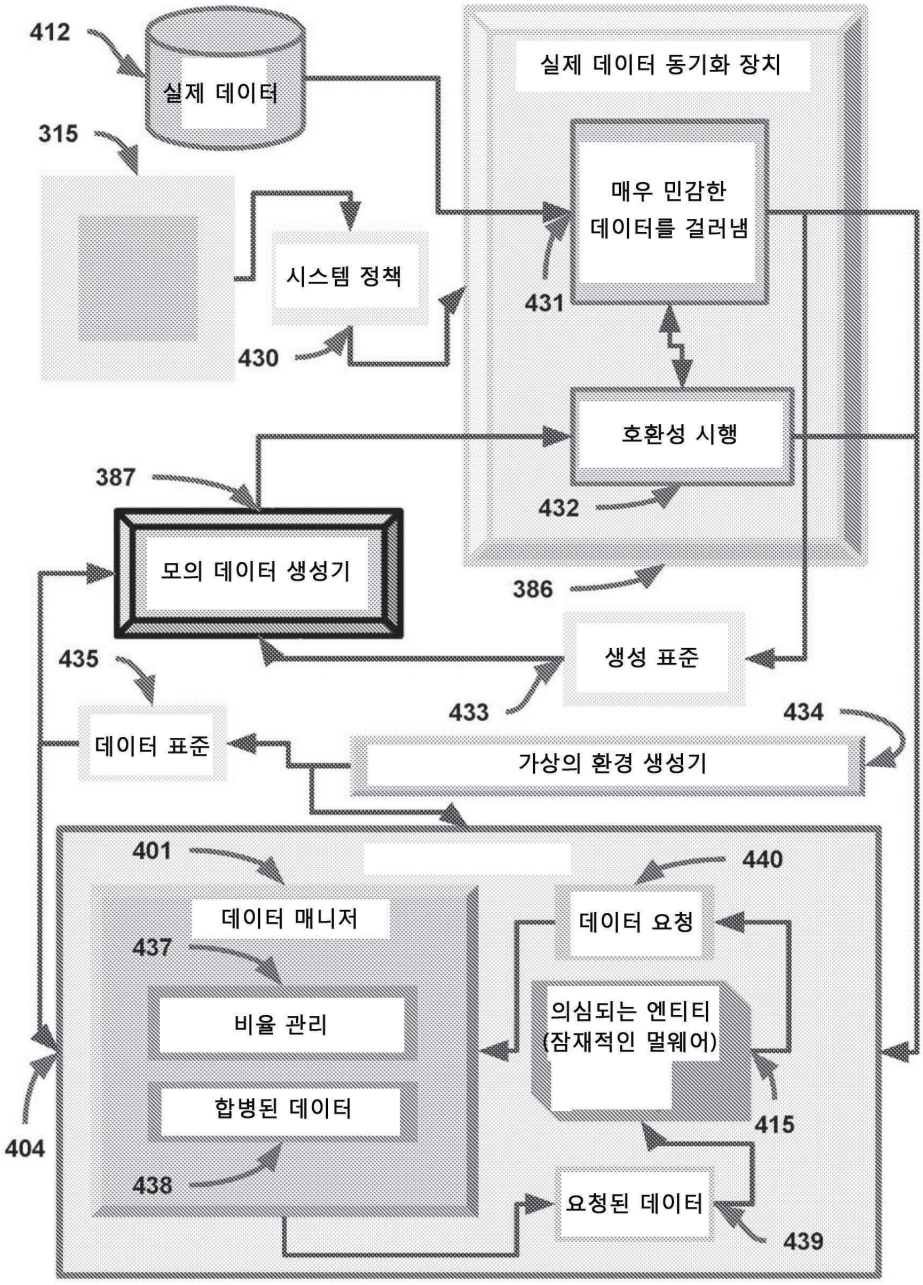
도면64


404
XY% 모의 데이터
415
414
의심되는 엔티티
(잠재적인 멀웨어)
행동 분석
403
416
데이터 매니저
실제 데이터 패치
401
417
실제 시스템
실제 데이터
412
418
모의 데이터를
대체하는 실제 데이터
399
407
데이터 회수 추적
모의 데이터
다운로드 추적
정정 동작
419
무해한 코드
409

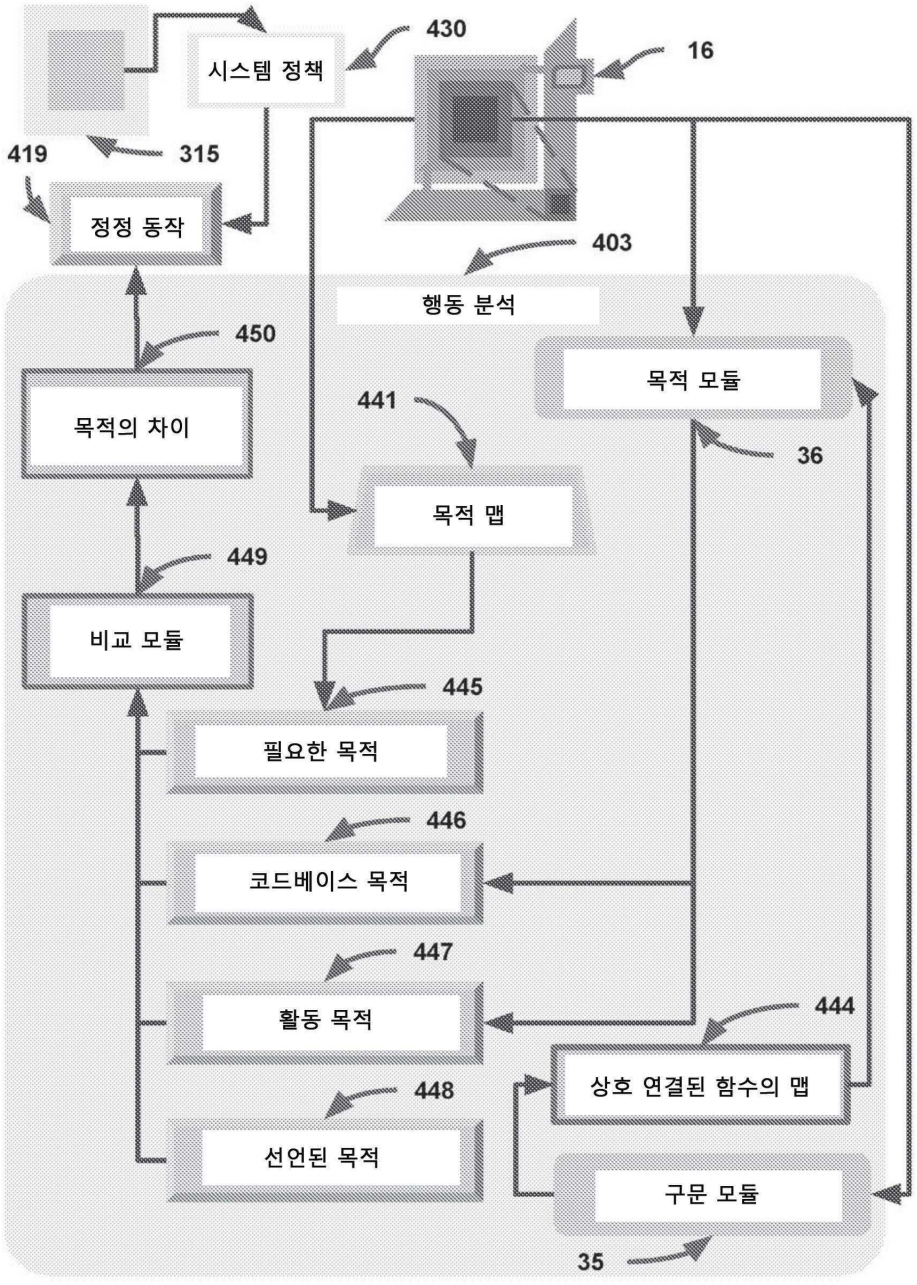
도면65


404
XY% 모의 데이터
임베디드 서버 환경
421
420
고정 데이터
423
다운로드 데이터
이전에 의심되는 엔티 (지금은 무해한 것으로 알려진)
비밀 토큰
424
실제 데이터 패치
비밀 토큰
426
416
422
토큰 발견: 실제 환경을 나타냄
427
428
425
지연된 세션 비활성화
지연된 세션 활성화
429
토큰 찾을 수 없음 모의 환경을 나타냄
426
416
지연된 인터페이스
424
데이터 매니저
실제 데이터 패치
비밀 토큰
427
401
416
데이터 회수 인터페이스
실제 데이터 패치
414

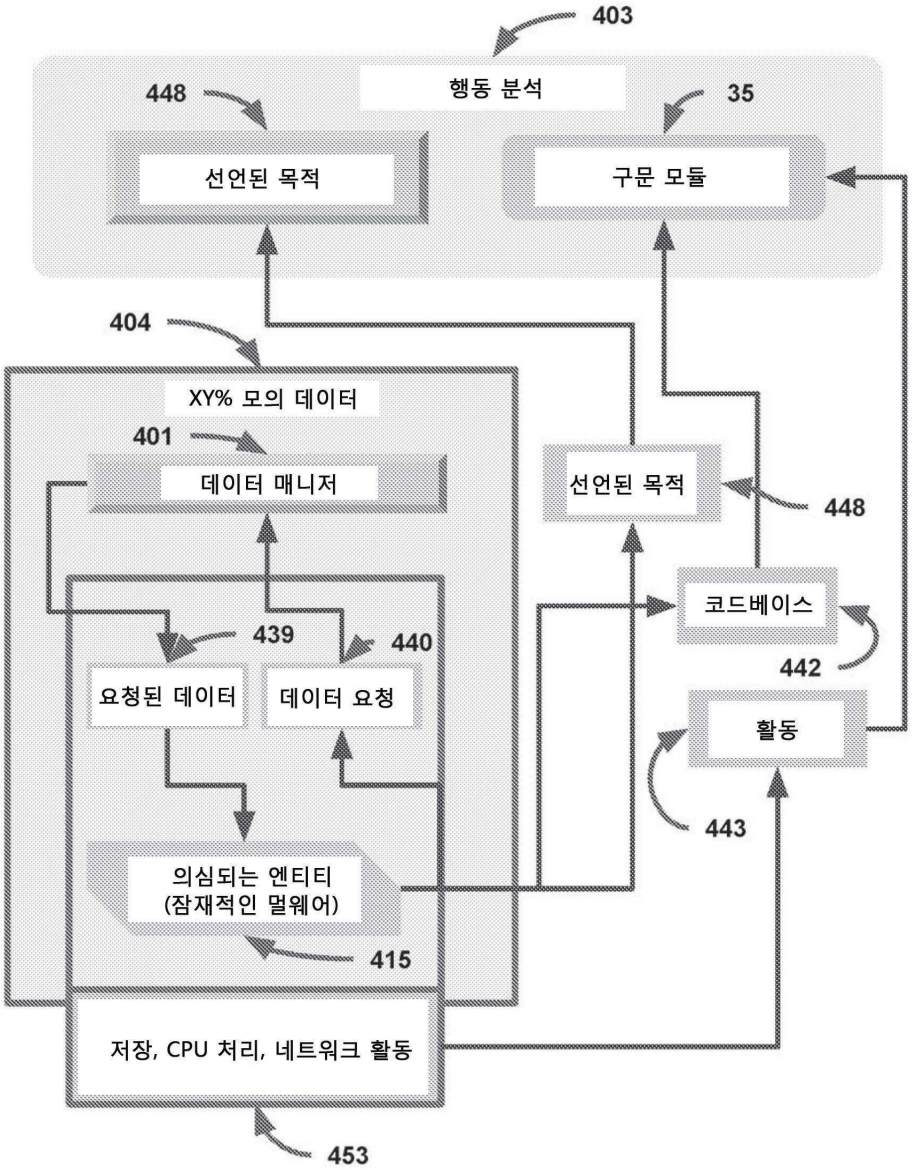
도면66


412
실제 데이터
315
시스템 정책
430
실제 데이터 동기화 장치
매우 민감한
데이터를 걸러냄
431
호환성 시행
432
387
모의 데이터 생성기
386
435
생성 표준
433
434
데이터 표준
가상의 환경 생성기
401
440
데이터 매니저
데이터 요청
437
비율 관리
의심되는 엔티티
(잠재적인 멀웨어)
415
합병된 데이터
404
438
요청된 데이터
439

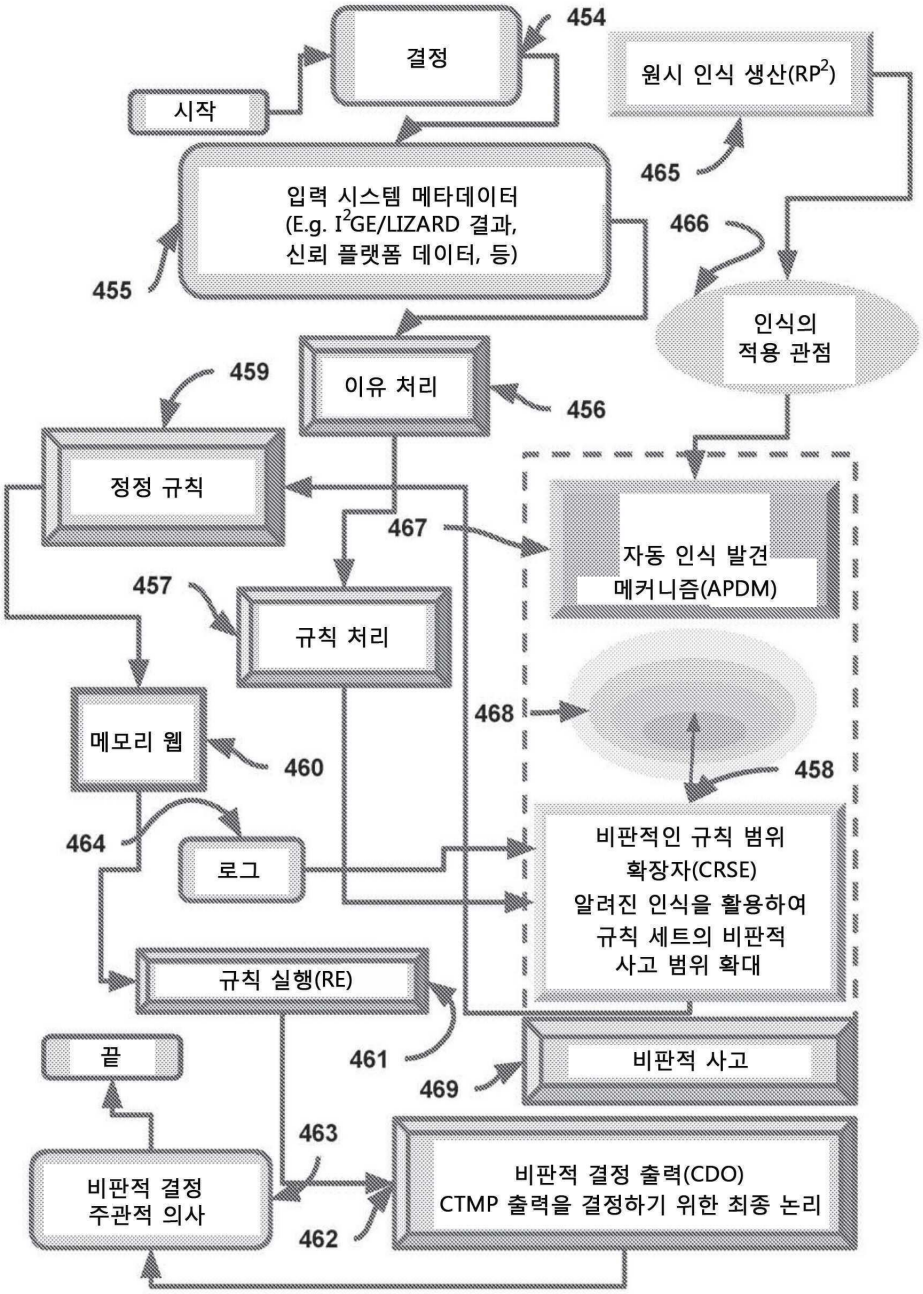
도면67


430
시스템 정책
16
419
315
정정 동작
403
행동 분석
450
목적 모듈
목적의 차이
441
36
목적 맵
449
비교 모듈
445
필요한 목적
446
코드베이스 목적
447
활동 목적
444
상호 연결된 함수의 맵
448
선언된 목적
구문 모듈
35

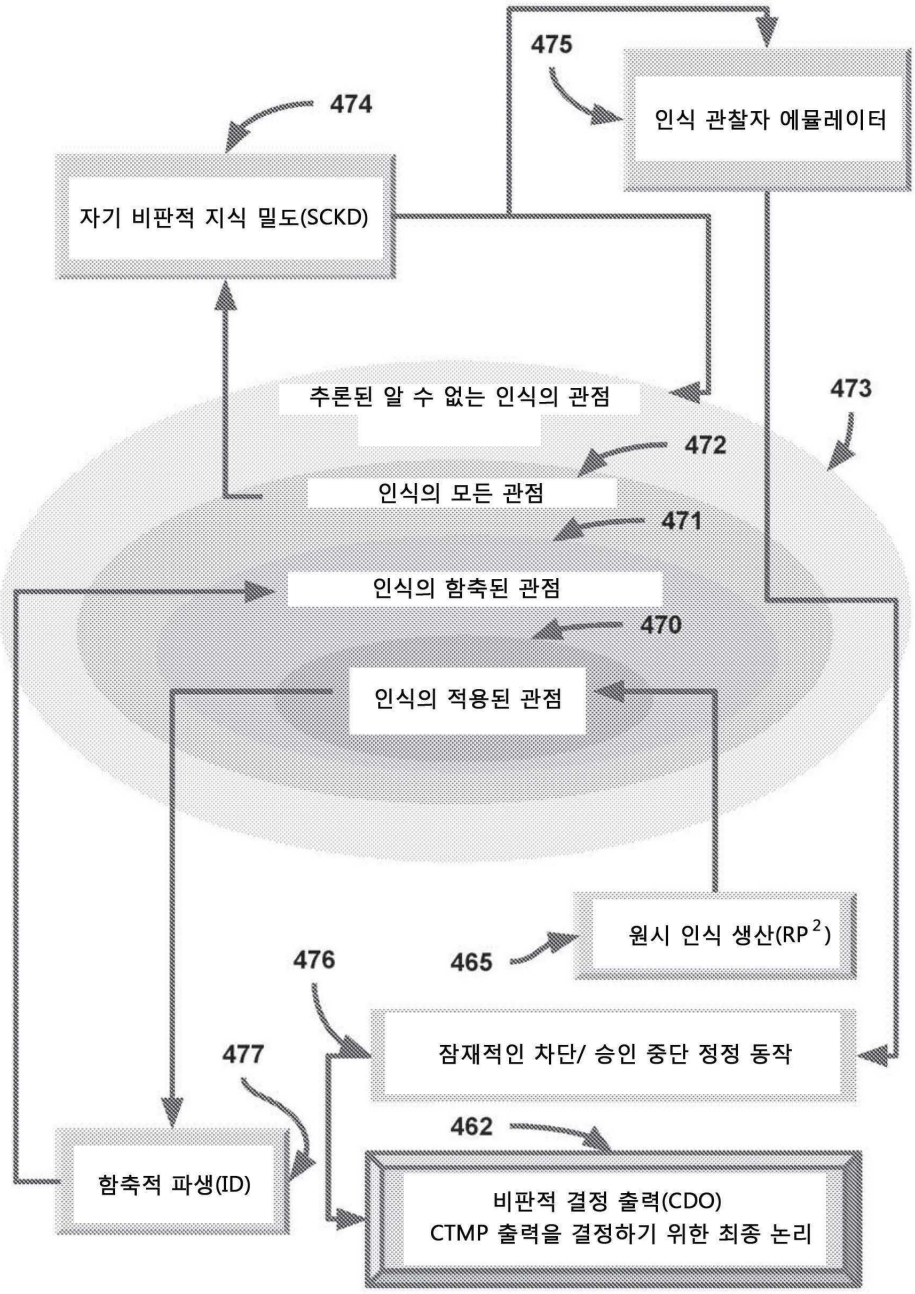
도면68


403
행동 분석
448
35
선언된 목적
구문 모듈
404
XY% 모의 데이터
401
데이터 매니저
선언된 목적
448
코드베이스
442
439
440
요청된 데이터
데이터 요청
활동
443
의심되는 엔티티
(잠재적인 멀웨어)
415
저장, CPU 처리, 네트워크 활동
453

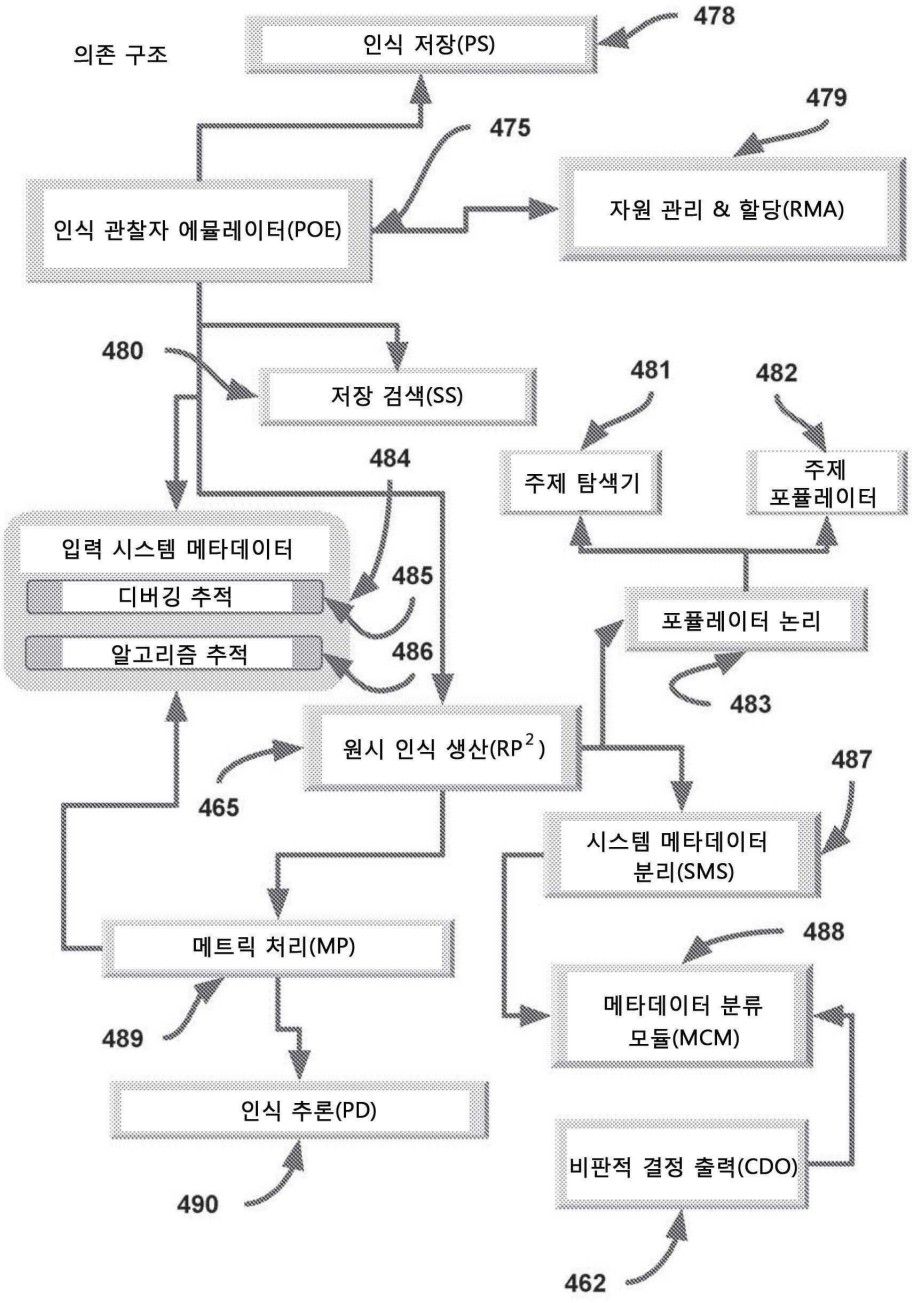
도면69


454
결정
시작
원시 인식 생산(RP2)
465
입력 시스템 메타데이터
(E.g. I2GE/LIZARD 결과,
신뢰 플랫폼 데이터, 등)
466
455
인식의
적용 관점
459
이유 처리
456
정정 규칙
467
자동 인식 발견
메커니즘(APDM)
457
규칙 처리
468
메모리 웹
460
458
464
로그
비판적인 규칙 범위
확장자(CRSE)
알려진 인식을 활용하여
규칙 세트의 비판적
사고 범위 확대
규칙 실행(RE)
끝
461
469
비판적 사고
463
비판적 결정
주관적 의사
462
비판적 결정 출력(CDO)
CTMP 출력을 결정하기 위한 최종 논리

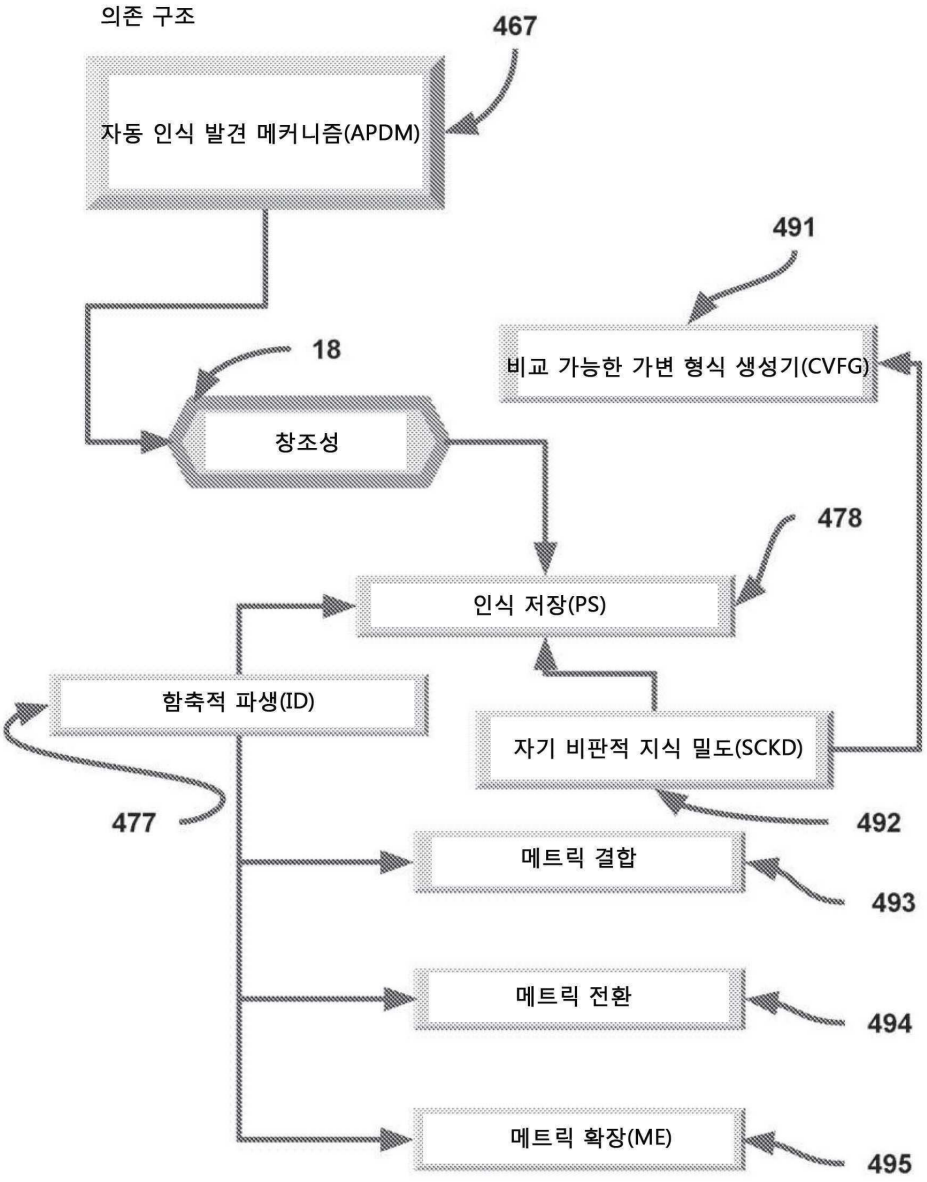
도면70


475
인식 관찰자 에뮬레이터
474
자기 비판적 지식 밀도(SCKD)
추론된 알 수 없는 인식의 관점
473
472
인식의 모든 관점
471
인식의 함축된 관점
470
인식의 적용된 관점
원시 인식 생산(RP2)
476
465
잠재적인 차단/ 승인 중단 정정 동작
477
462
함축적 파생(ID)
비판적 결정 출력(CDO)
CTMP 출력을 결정하기 위한 최종 논리

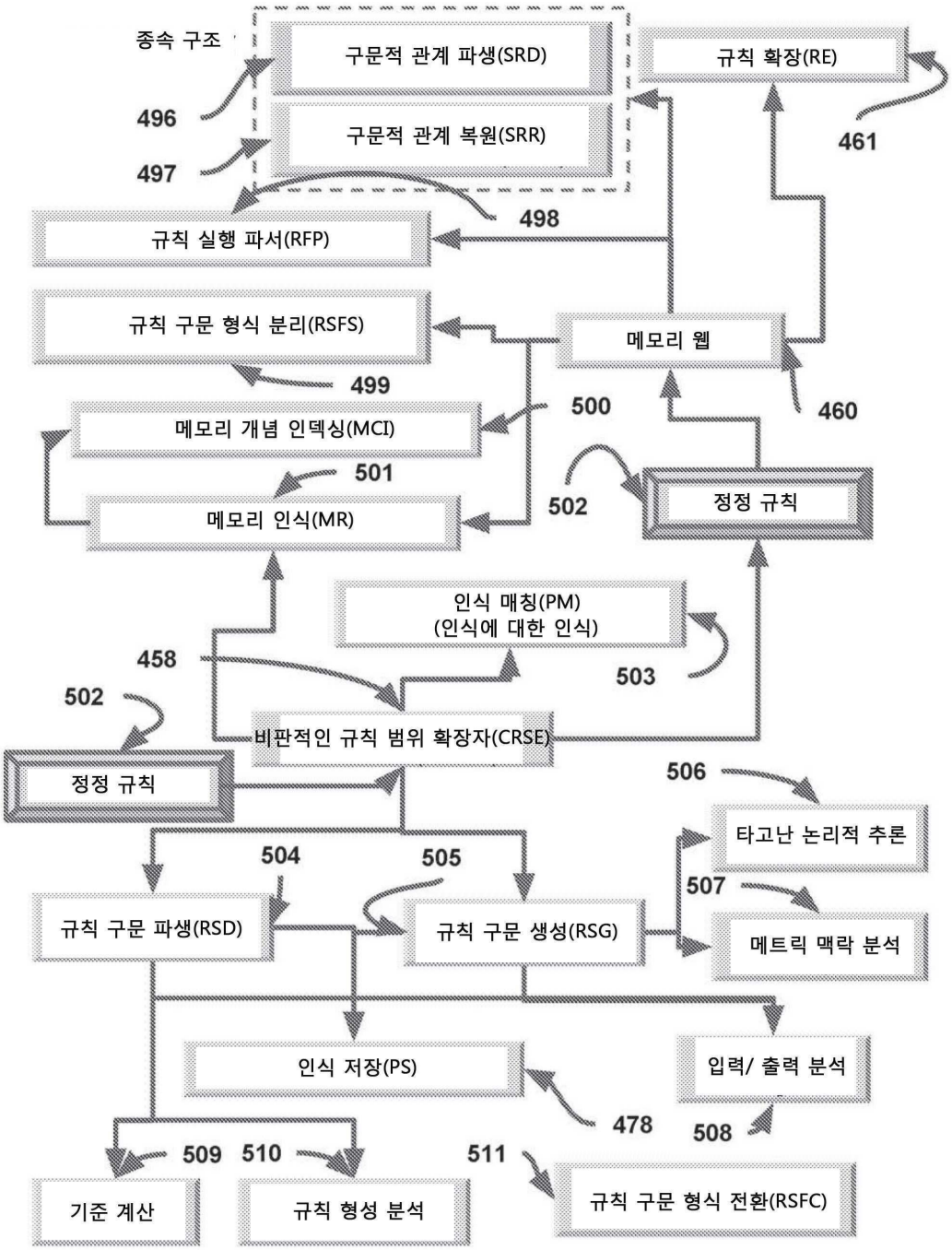
도면71


의존 구조
인식 저장(PS)
478
479
475
인식 관찰자 에뮬레이터(POE)
자원 관리 & 할당(RMA)
480
저장 검색(SS)
481
482
주제 탐색기
주제 포퓰레이터
484
입력 시스템 메타데이터
디버깅 추적
485
알고리즘 추적
486
포퓰레이터 논리
483
원시 인식 생산(RP2)
465
487
시스템 메타데이터 분리(SMS)
메트릭 처리(MP)
488
489
메타데이터 분류 모듈(MCM)
인식 추론(PD)
비판적 결정 출력(CDO)
490
462

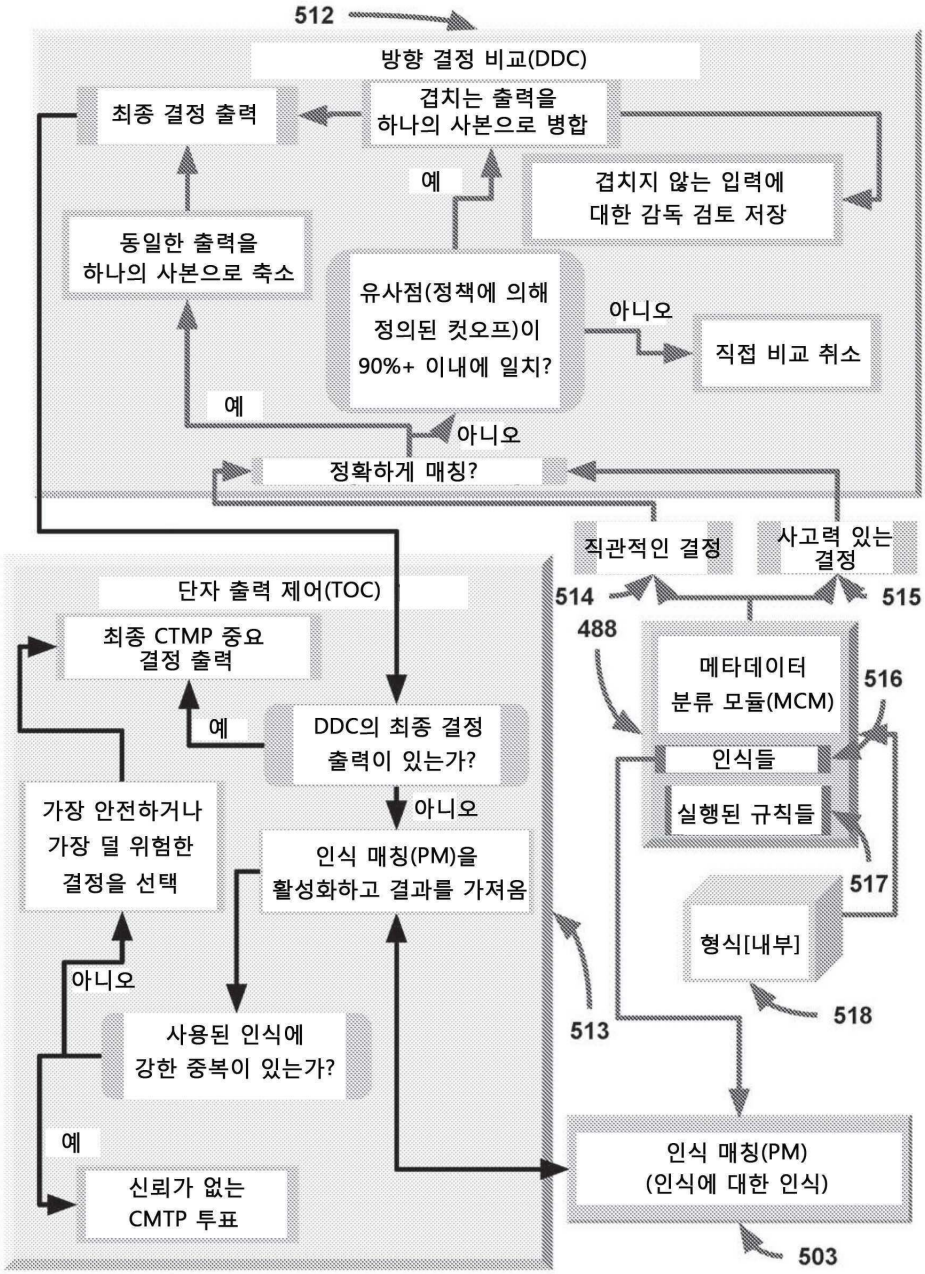
도면72


의존 구조
467
자동 인식 발견 메커니즘(APDM)
491
비교 가능한 가변 형식 생성기(CVFG)
18
창조성
478
인식 저장(PS)
함축적 파생(ID)
자기 비판적 지식 밀도(SCKD)
477
492
메트릭 결합
493
메트릭 전환
494
메트릭 확장(ME)
495

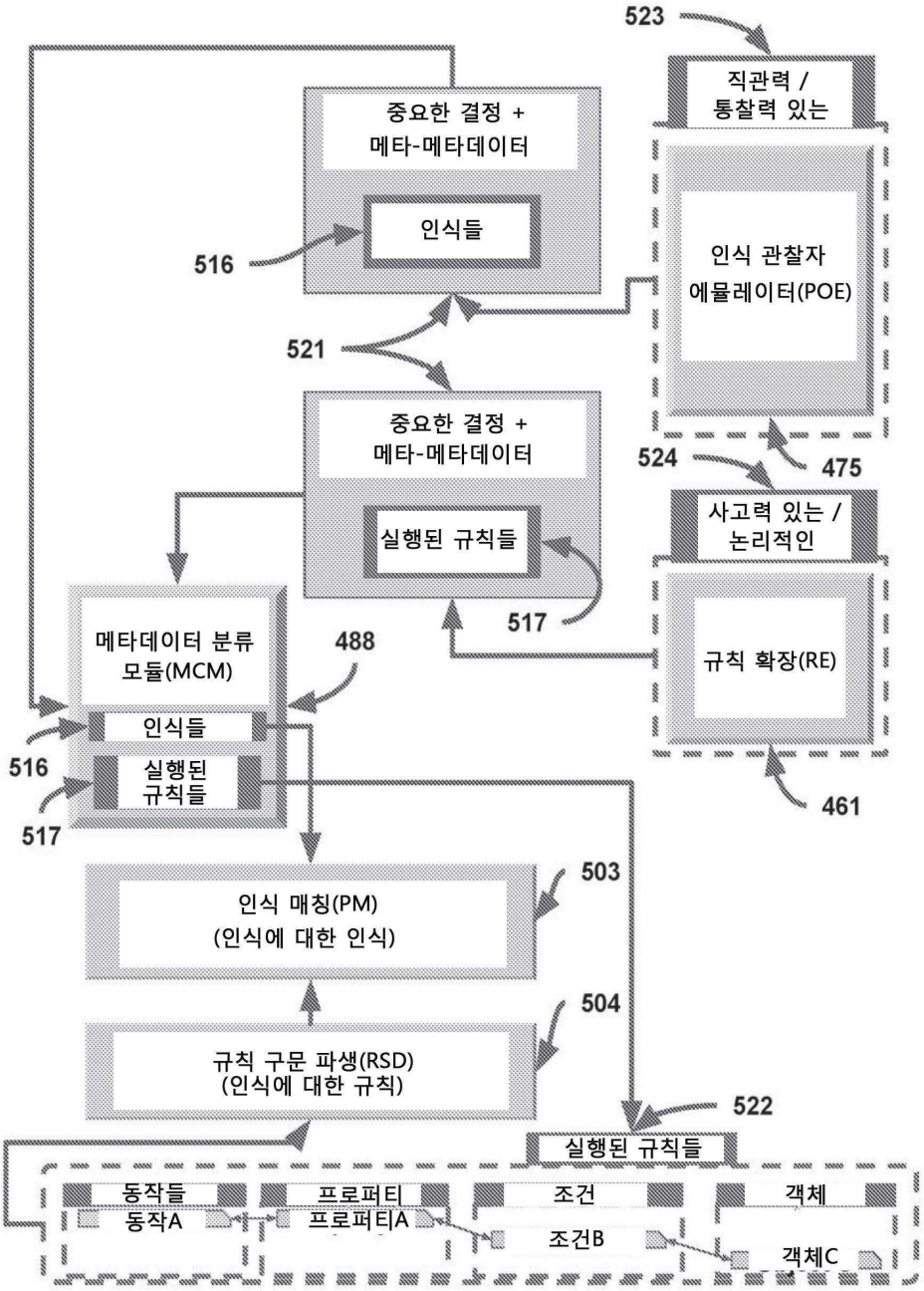
도면73


종속 구조
구문적 관계 파생(SRD)
구문적 관계 복원(SRR)
496
497
498
규칙 확장(RE)
461
규칙 실행 파서(RFP)
규칙 구문 형식 분리(RSFS)
메모리 웹
499
500
460
메모리 개념 인덱싱(MCI)
501
502
정정 규칙
메모리 인식(MR)
인식 매칭(PM)
(인식에 대한 인식)
458
503
502
비판적인 규칙 범위 확장자(CRSE)
정정 규칙
506
타고난 논리적 추론
504
505
507
규칙 구문 파생(RSD)
규칙 구문 생성(RSG)
메트릭 맥락 분석
인식 저장(PS)
입력/ 출력 분석
478
508
509
510
511
기준 계산
규칙 형성 분석
규칙 구문 형식 전환(RSFC)

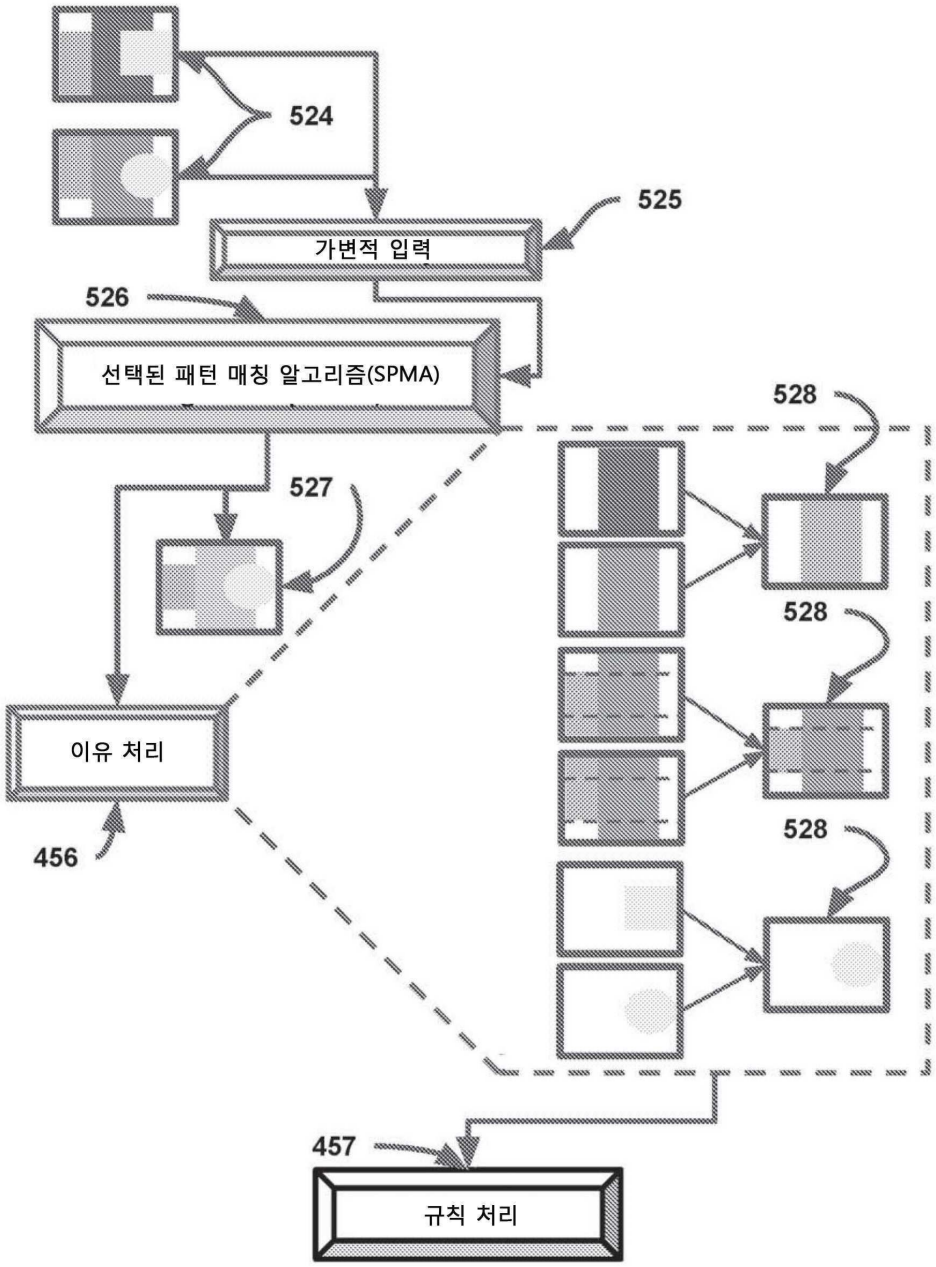
도면74


512
방향 결정 비교(DDC)
최종 결정 출력
겹치는 출력을
하나의 사본으로 병합
예
겹치지 않는 입력에
대한 감독 검토 저장
동일한 출력을
하나의 사본으로 축소
유사점(정책에 의해
정의된 컷오프)이
90%+ 이내에 일치?
아니오
직접 비교 취소
예
아니오
정확하게 매칭?
직관적인 결정
사고력 있는
결정
514
515
488
단자 출력 제어(TOC)
최종 CTMP 중요
결정 출력
메타데이터
분류 모듈(MCM)
516
예
DDC의 최종 결정
출력이 있는가?
인식들
실행된 규칙들
아니오
가장 안전하거나
가장 덜 위험한
결정을 선택
인식 매칭(PM)을
활성화하고 결과를 가져옴
517
형식[내부]
518
아니오
513
사용된 인식에
강한 중복이 있는가?
예
인식 매칭(PM)
(인식에 대한 인식)
신뢰가 없는
CMTP 투표
503

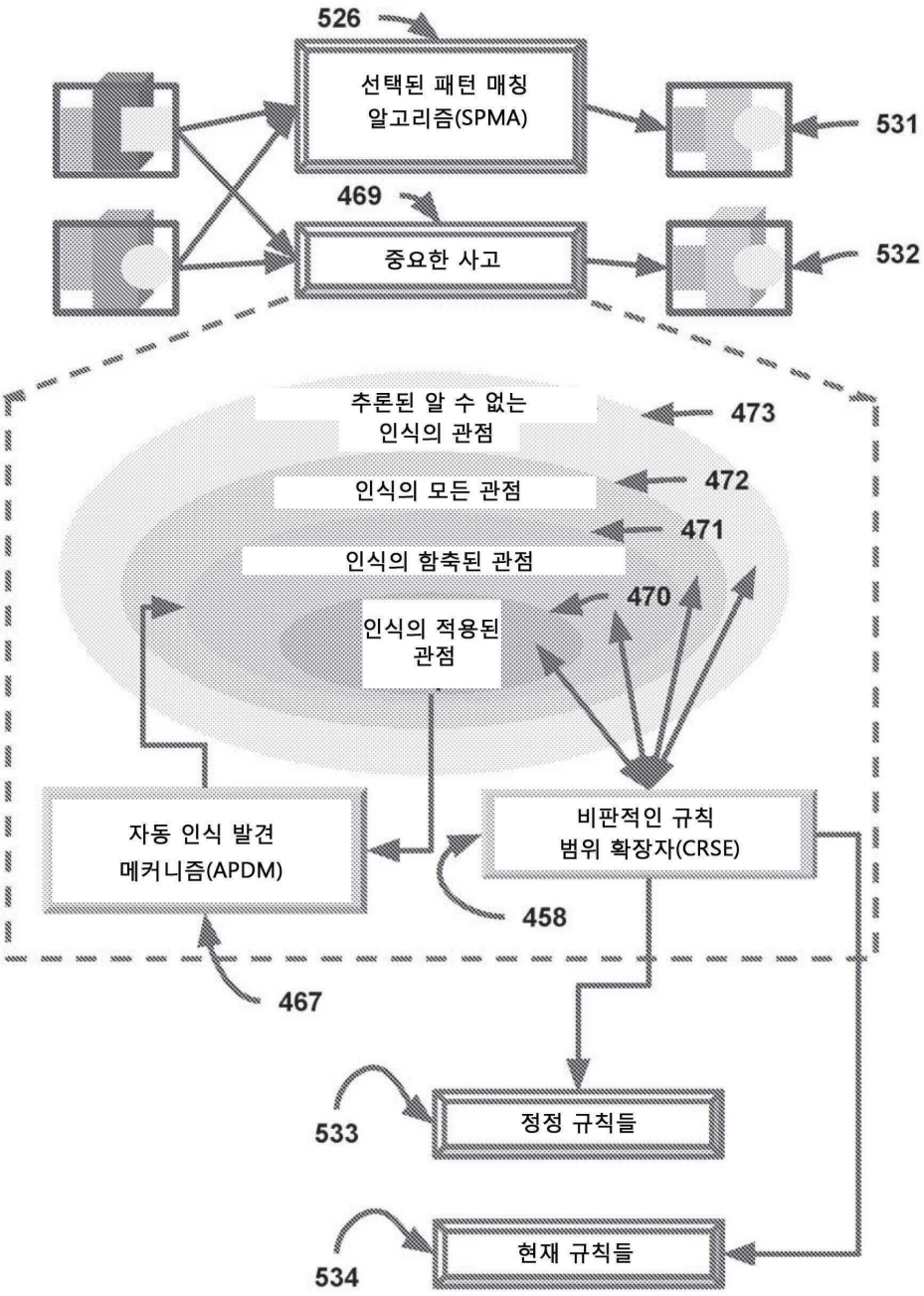
도면75


523
직관력 / 통찰력 있는
중요한 결정 + 메타-메타데이터
인식들
516
인식 관찰자 에뮬레이터(POE)
521
중요한 결정 + 메타-메타데이터
524
475
실행된 규칙들
사고력 있는 / 논리적인
517
메타데이터 분류 모듈(MCM)
488
규칙 확장(RE)
인식들
516
실행된 규칙들
517
461
503
인식 매칭(PM)
(인식에 대한 인식)
504
규칙 구문 파생(RSD)
(인식에 대한 규칙)
522
실행된 규칙들
동작들
동작A
프로퍼티
프로퍼티A
조건
조건B
객체
객체C

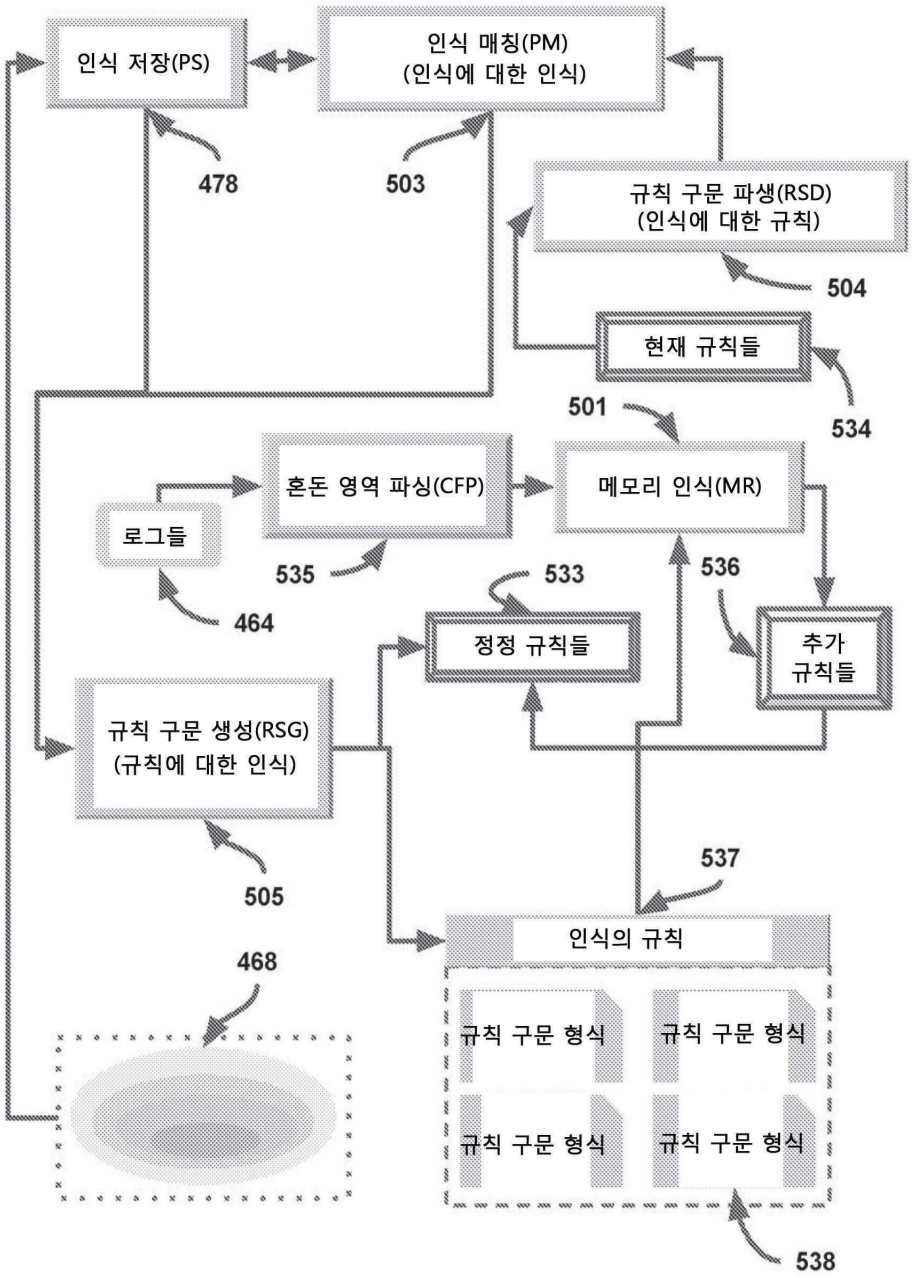
도면76


524
525
가변적 입력
526
선택된 패턴 매칭 알고리즘(SPMA)
528
527
528
528
이유 처리
456
457
규칙 처리

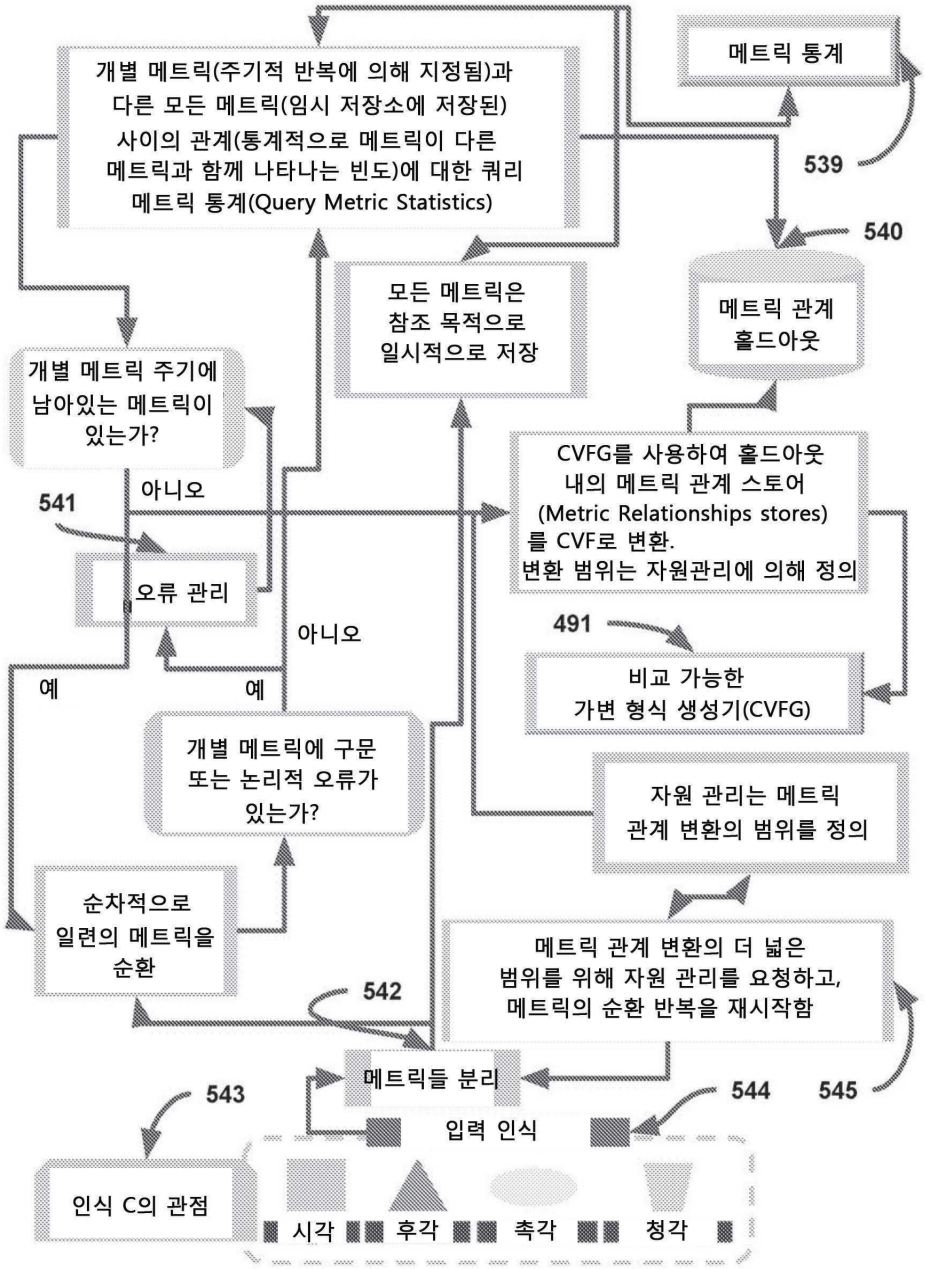
도면77


526
선택된 패턴 매칭
알고리즘(SPMA)
531
469
중요한 사고
532
추론된 알 수 없는
인식의 관점
473
인식의 모든 관점
472
471
인식의 함축된 관점
470
인식의 적용된
관점
자동 인식 발견
메커니즘(APDM)
비판적인 규칙
범위 확장자(CRSE)
458
467
533
정정 규칙들
534
현재 규칙들

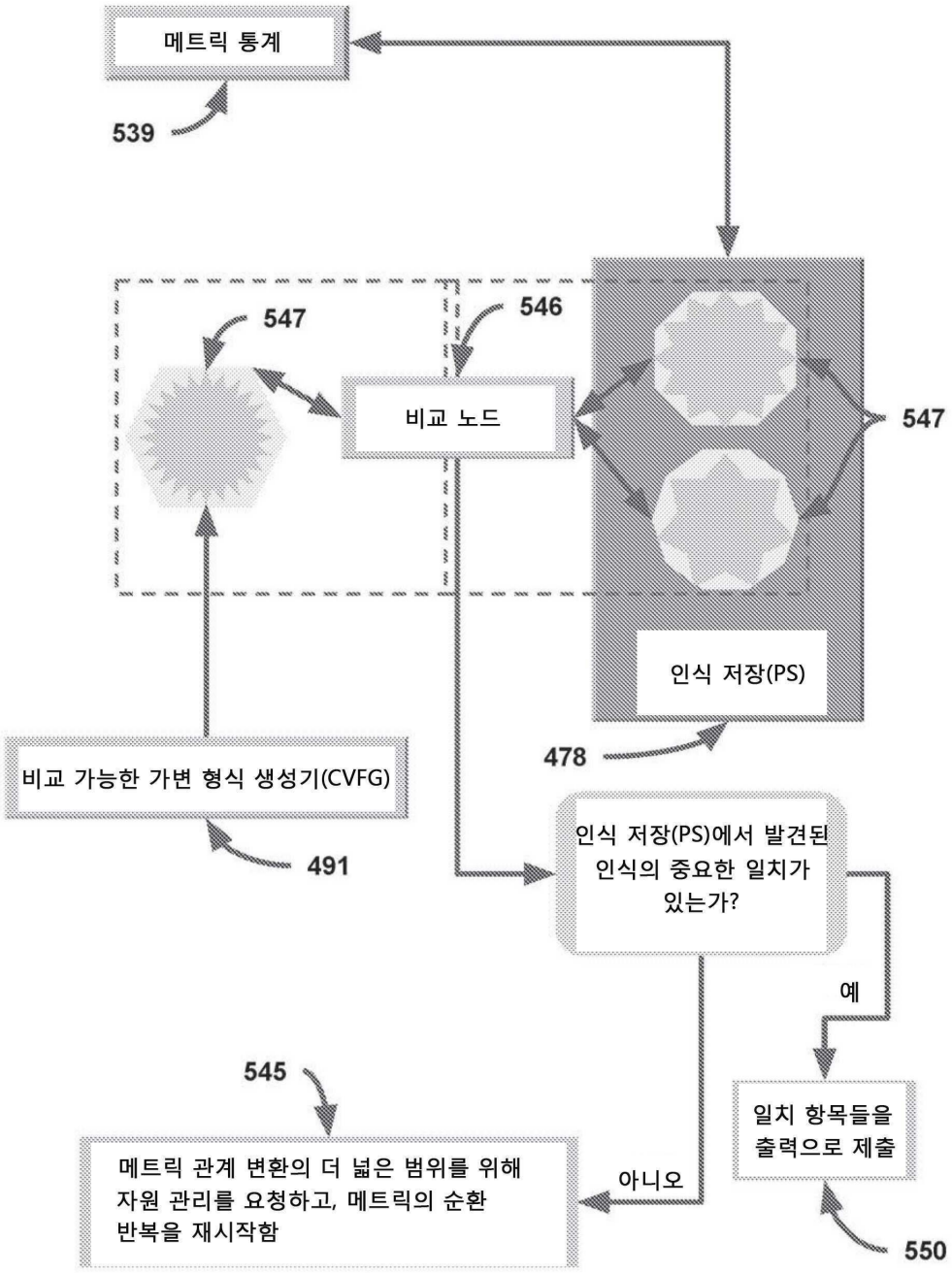
도면78


인식 저장(PS)
인식 매칭(PM)
(인식에 대한 인식)
478
503
규칙 구문 파생(RSD)
(인식에 대한 규칙)
504
현재 규칙들
534
501
혼돈 영역 파싱(CFP)
메모리 인식(MR)
로그들
535
533
536
464
정정 규칙들
추가
규칙들
규칙 구문 생성(RSG)
(규칙에 대한 인식)
505
537
인식의 규칙
468
규칙 구문 형식
규칙 구문 형식
규칙 구문 형식
규칙 구문 형식
538

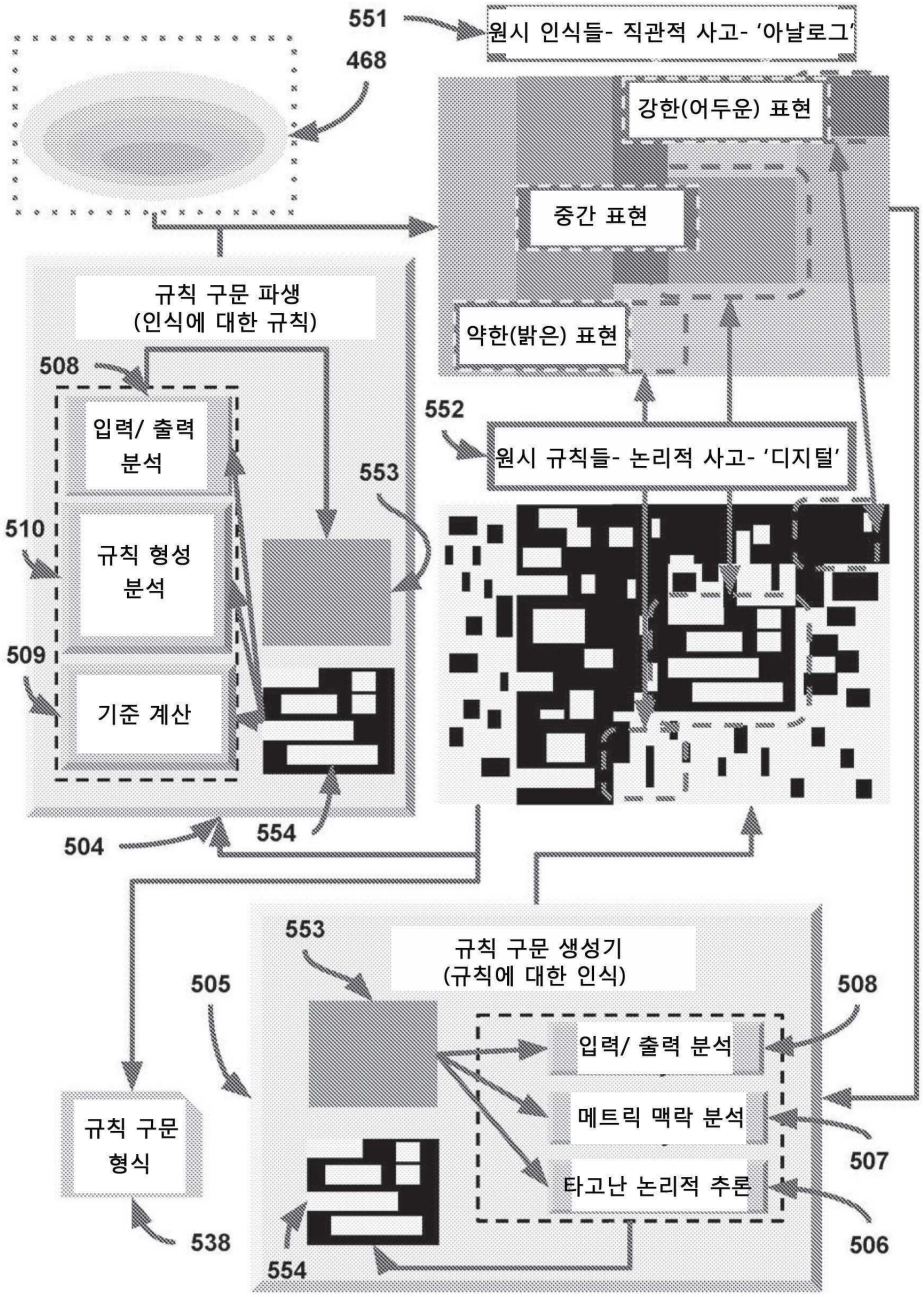
도면79


개별 메트릭(주기적 반복에 의해 지정됨)과 다른 모든 메트릭(임시 저장소에 저장된) 사이의 관계(통계적으로 메트릭이 다른 메트릭과 함께 나타나는 빈도)에 대한 쿼리 메트릭 통계(Query Metric Statistics)
메트릭 통계
539
540
모든 메트릭은 참조 목적으로 일시적으로 저장
메트릭 관계 홀드아웃
개별 메트릭 주기에 남아있는 메트릭이 있는가?
아니오
541
CVFG를 사용하여 홀드아웃 내의 메트릭 관계 스토어 (Metric Relationships stores)를 CVF로 변환. 변환 범위는 자원관리에 의해 정의
오류 관리
아니오
491
예
예
비교 가능한 가변 형식 생성기(CVFG)
개별 메트릭에 구문 또는 논리적 오류가 있는가?
자원 관리는 메트릭 관계 변환의 범위를 정의
순차적으로 일련의 메트릭을 순환
메트릭 관계 변환의 더 넓은 범위를 위해 자원 관리를 요청하고, 메트릭의 순환 반복을 재시작함
542
메트릭들 분리
543
544
545
입력 인식
인식 C의 관점
시각
후각
촉각
청각

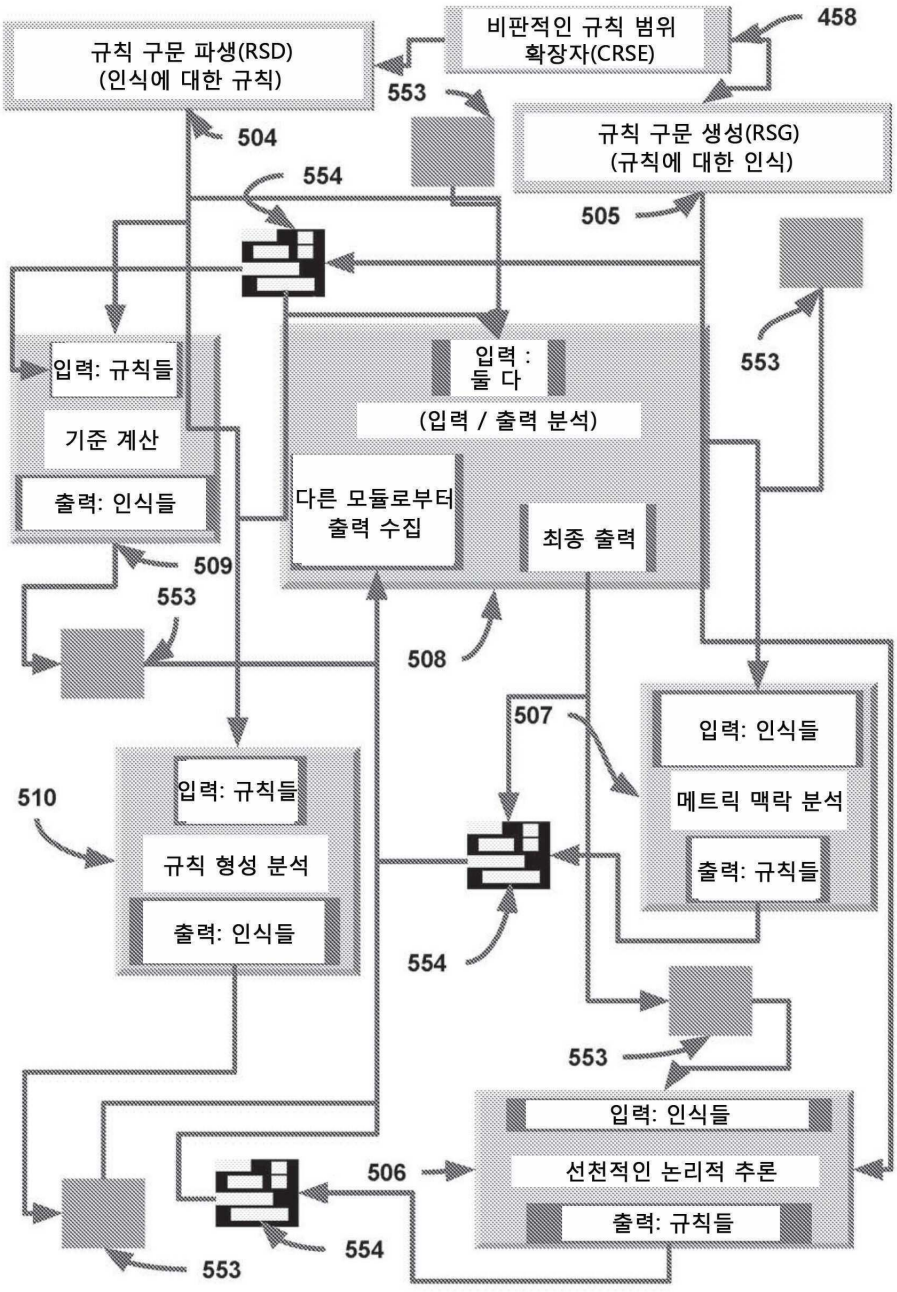
도면80


메트릭 통계
539
547
546
비교 노드
547
인식 저장(PS)
478
비교 가능한 가변 형식 생성기(CVFG)
491
인식 저장(PS)에서 발견된 인식의 중요한 일치가 있는가?
예
일치 항목들을 출력으로 제출
550
545
메트릭 관계 변환의 더 넓은 범위를 위해 자원 관리를 요청하고, 메트릭의 순환 반복을 재시작함
아니오

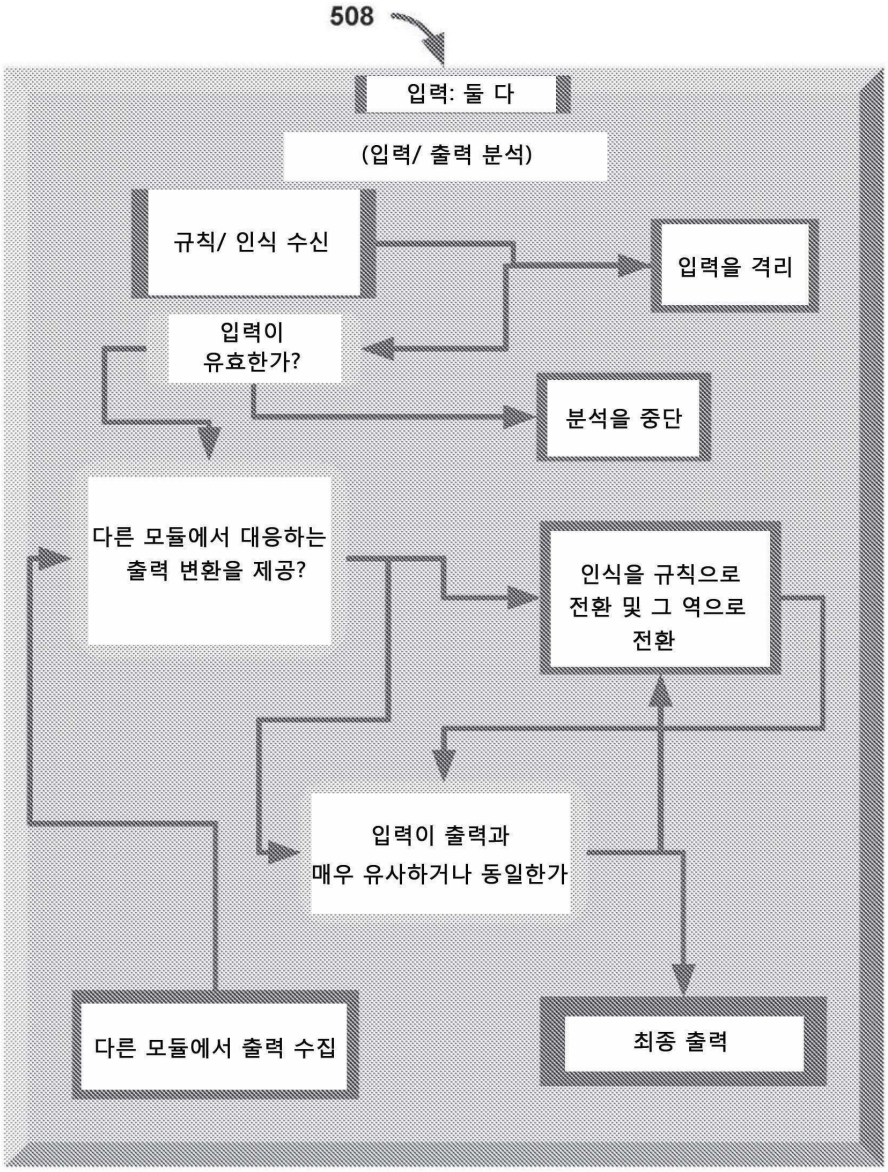
도면81


551
원시 인식들- 직관적 사고- '아날로그'
468
강한(어두운) 표현
중간 표현
약한(밝은) 표현
규칙 구문 파생
(인식에 대한 규칙)
508
입력/ 출력
분석
552
원시 규칙들- 논리적 사고- '디지털'
553
510
규칙 형성
분석
509
기준 계산
554
504
553
규칙 구문 생성기
(규칙에 대한 인식)
505
508
입력/ 출력 분석
메트릭 맥락 분석
507
타고난 논리적 추론
506
규칙 구문
형식
538
554

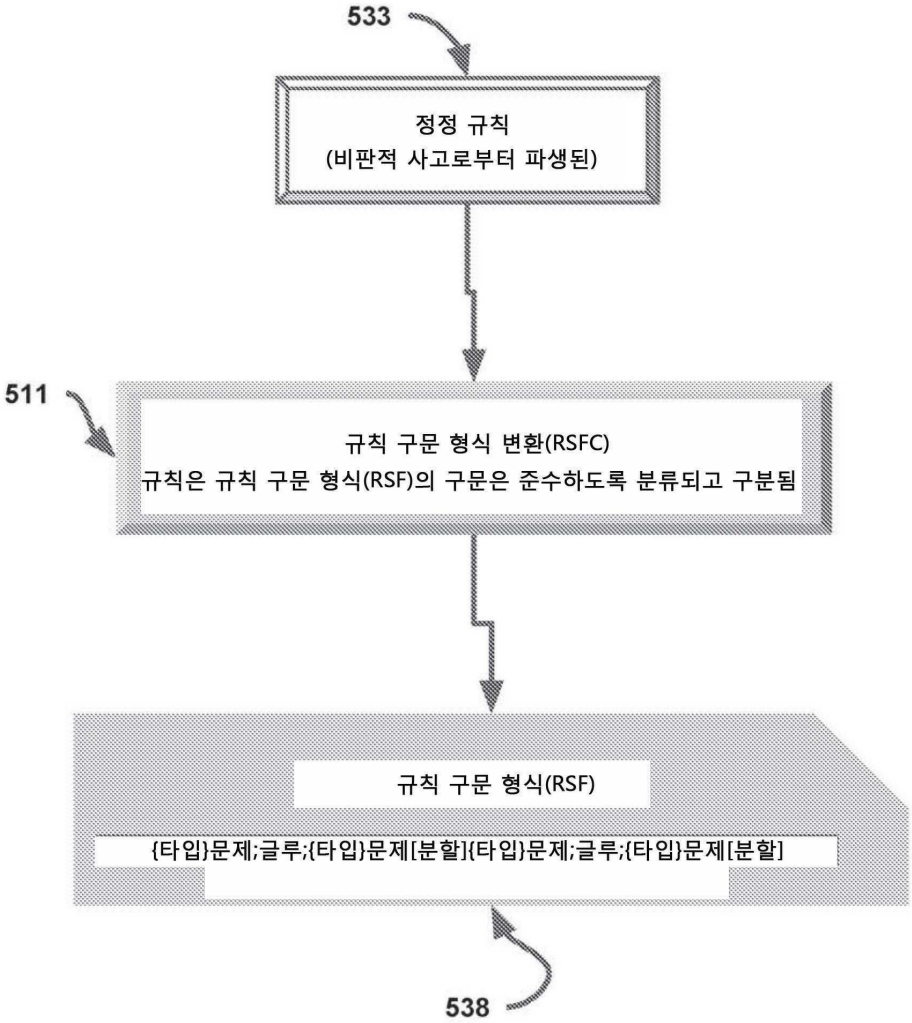
도면82


비판적인 규칙 범위 확장자(CRSE)
458
규칙 구문 파생(RSD) (인식에 대한 규칙)
553
504
규칙 구문 생성(RSG) (규칙에 대한 인식)
554
505
553
입력: 규칙들
입력 : 둘 다
(입력 / 출력 분석)
기준 계산
출력: 인식들
다른 모듈로부터 출력 수집
최종 출력
509
553
508
507
입력: 인식들
510
입력: 규칙들
메트릭 맥락 분석
규칙 형성 분석
출력: 규칙들
출력: 인식들
554
553
입력: 인식들
506
선천적인 논리적 추론
554
출력: 규칙들
553

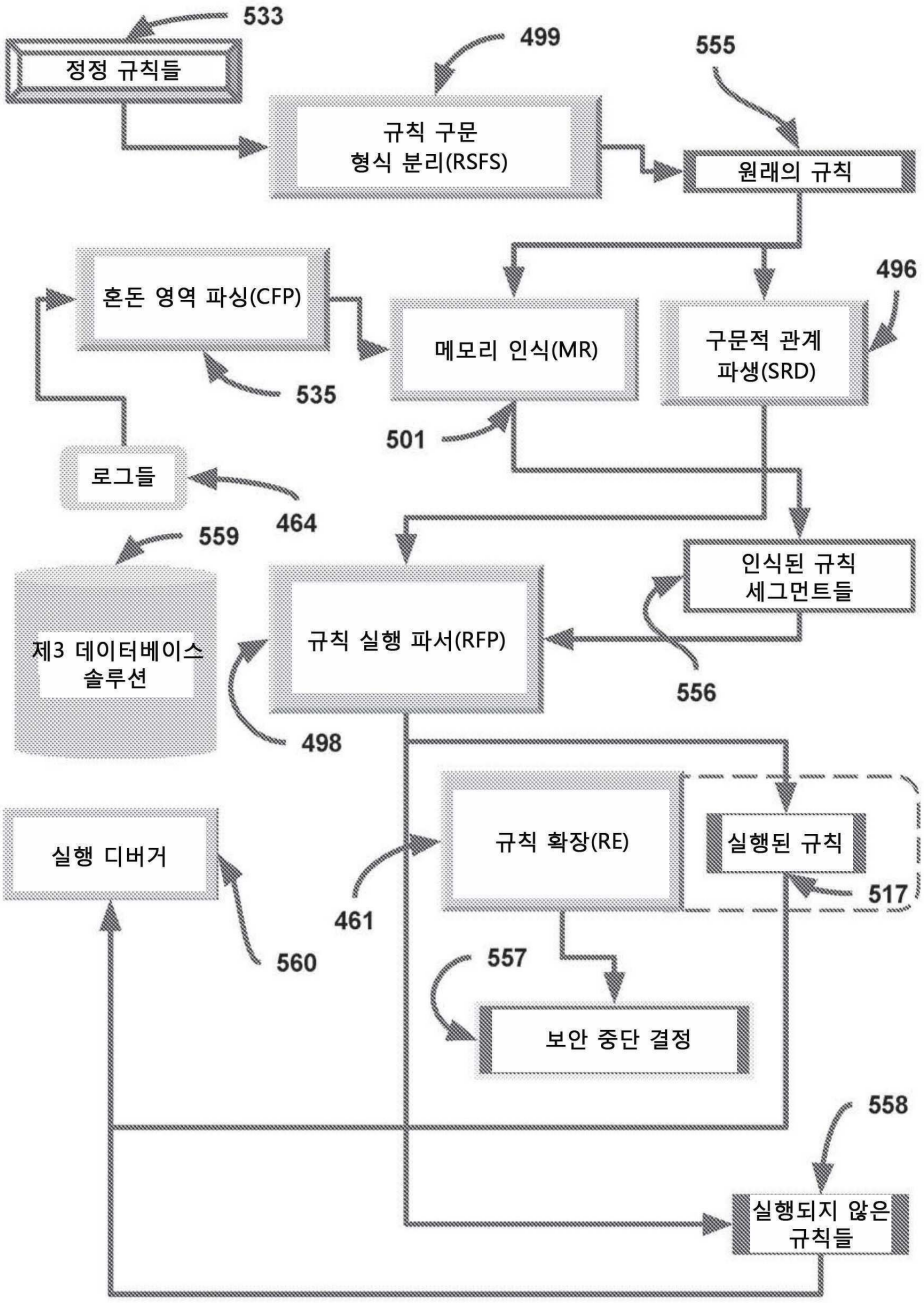
도면83


508
입력: 둘 다
(입력/ 출력 분석)
규칙/ 인식 수신
입력을 격리
입력이
유효한가?
분석을 중단
다른 모듈에서 대응하는
출력 변환을 제공?
인식을 규칙으로
전환 및 그 역으로
전환
입력이 출력과
매우 유사하거나 동일한가
다른 모듈에서 출력 수집
최종 출력

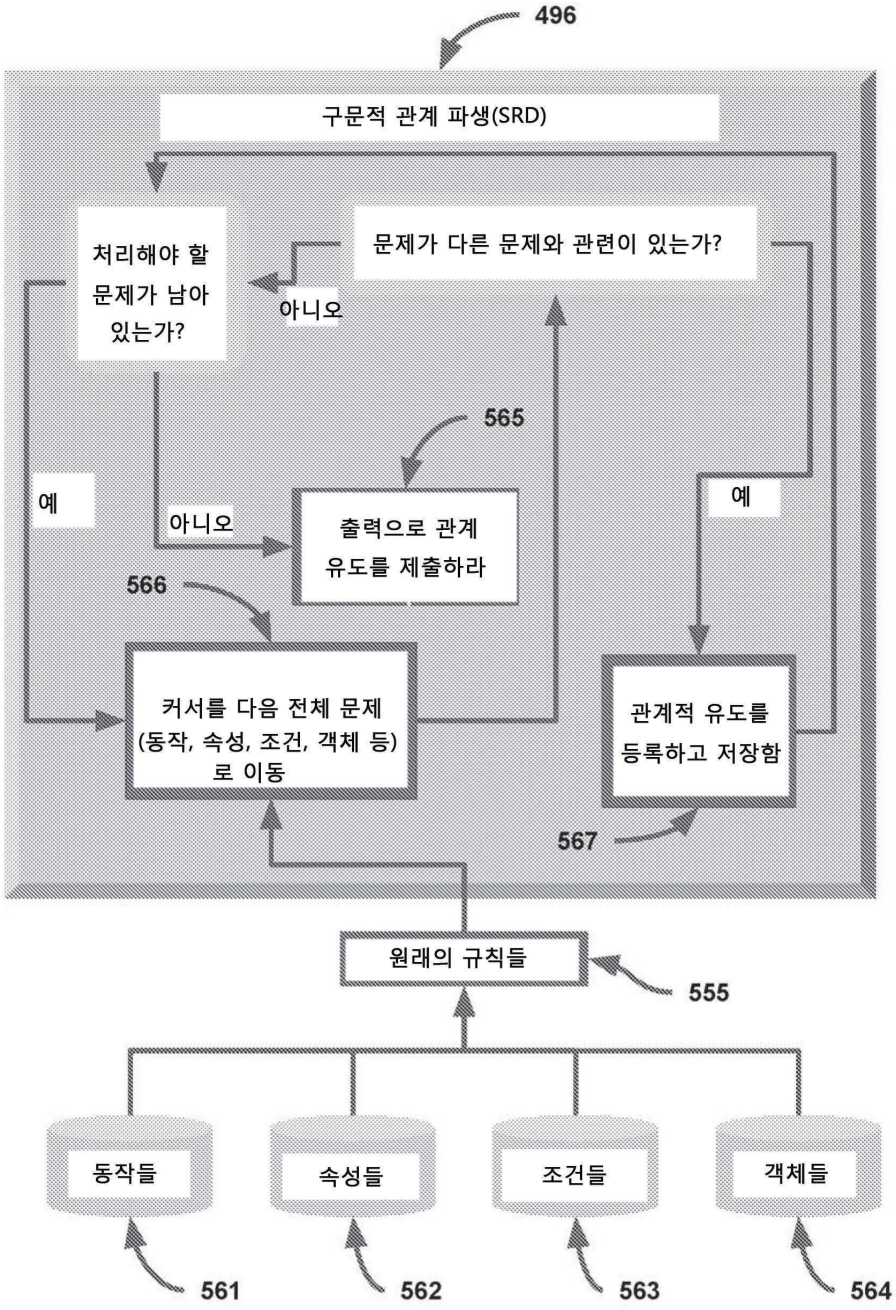
도면84


533
정정 규칙
(비판적 사고로부터 파생된)
511
규칙 구문 형식 변환(RSFC)
규칙은 규칙 구문 형식(RSF)의 구문은 준수하도록 분류되고 구분됨
규칙 구문 형식(RSF)
{타입}문제;글루;{타입}문제[분할]{타입}문제;글루;{타입}문제[분할]
538

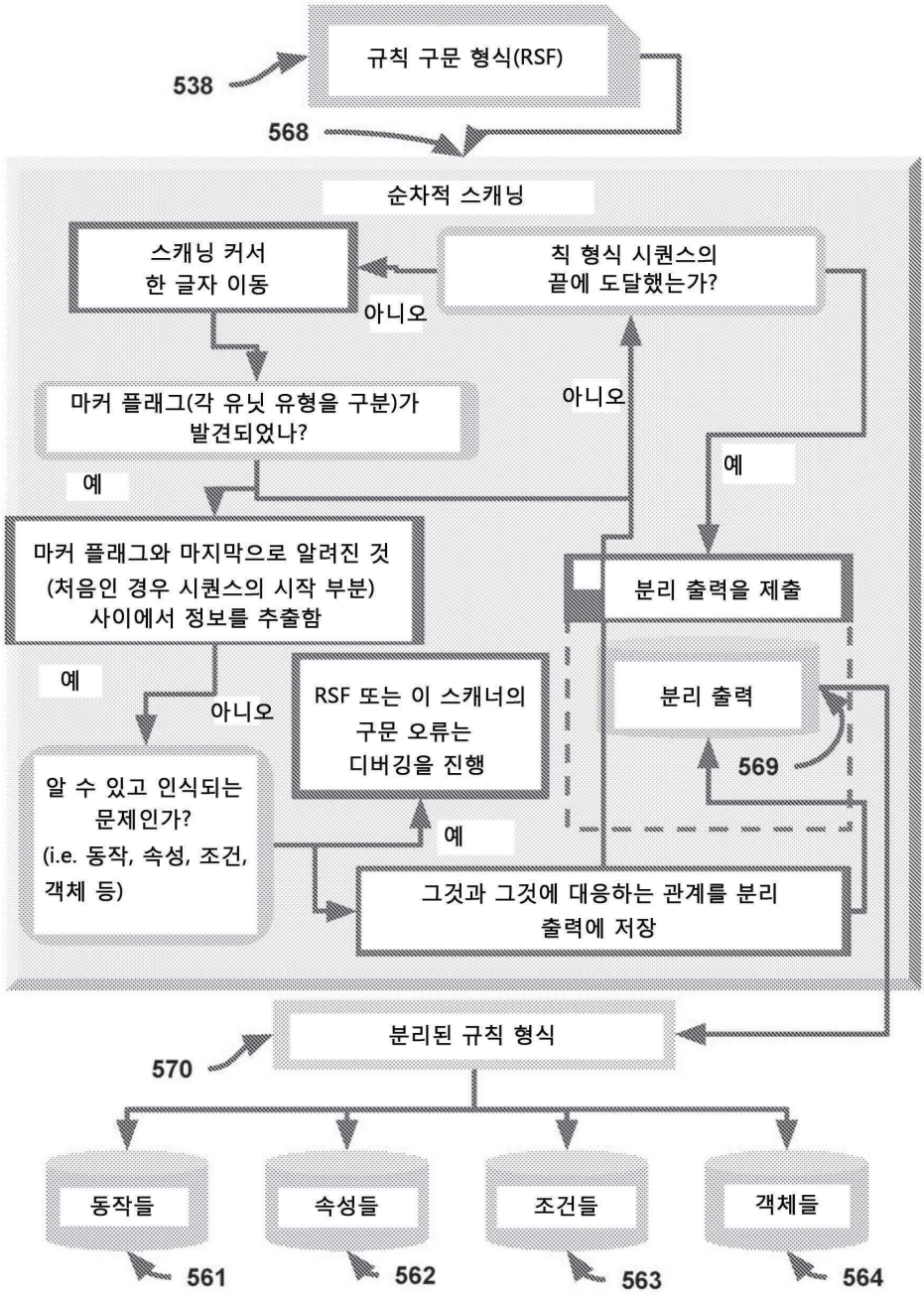
도면85


533
정정 규칙들
499
규칙 구문
형식 분리(RSFS)
555
원래의 규칙
혼돈 영역 파싱(CFP)
535
메모리 인식(MR)
496
구문적 관계
파생(SRD)
501
로그들
464
559
제3 데이터베이스
솔루션
규칙 실행 파서(RFP)
498
인식된 규칙
세그먼트들
556
실행 디버거
560
461
규칙 확장(RE)
실행된 규칙
517
557
보안 중단 결정
558
실행되지 않은
규칙들

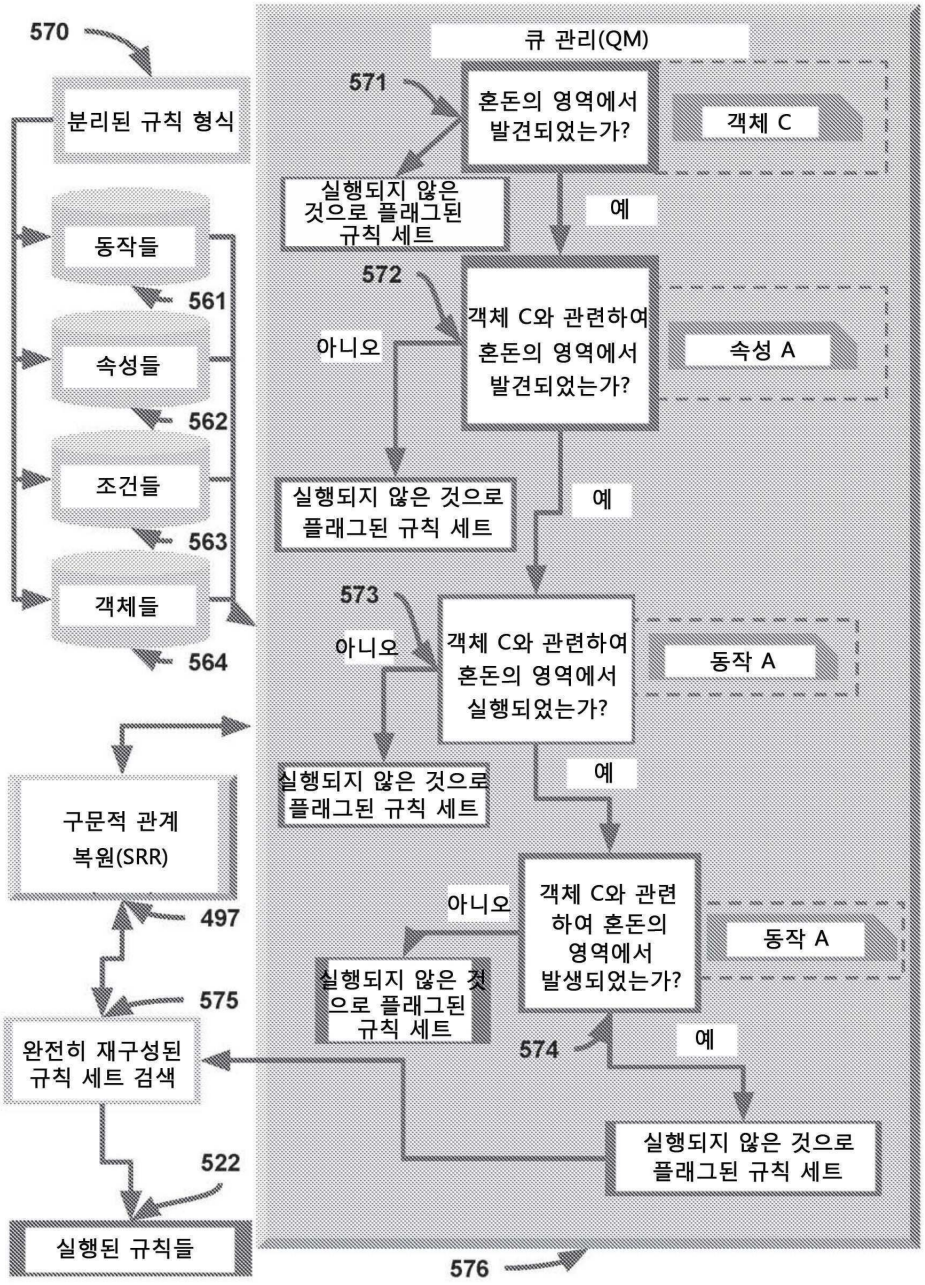
도면86


496
구문적 관계 파생(SRD)
처리해야 할 문제가 남아 있는가?
문제가 다른 문제와 관련이 있는가?
아니오
565
예
아니오
출력으로 관계 유도를 제출하라
예
566
커서를 다음 전체 문제 (동작, 속성, 조건, 객체 등)로 이동
관계적 유도를 등록하고 저장함
567
원래의 규칙들
555
동작들
속성들
조건들
객체들
561
562
563
564

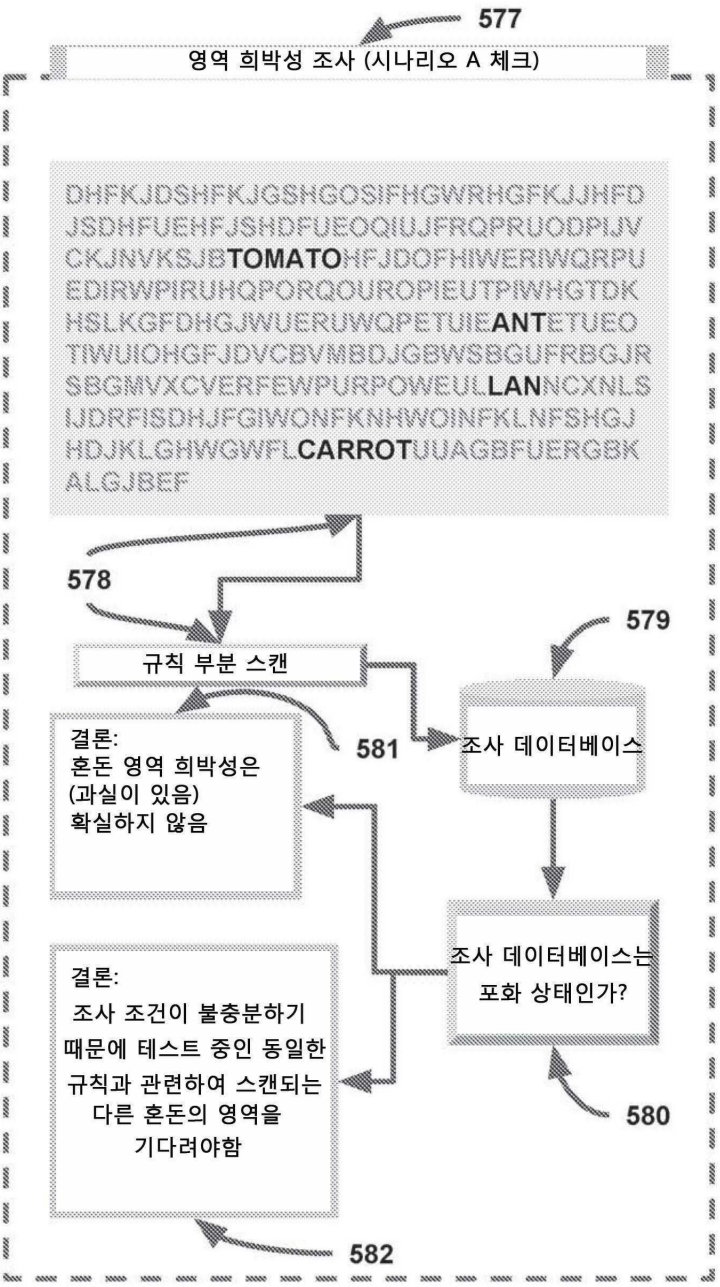
도면87


538
규칙 구문 형식(RSF)
568
순차적 스캐닝
스캐닝 커서
한 글자 이동
칙 형식 시퀀스의
끝에 도달했는가?
아니오
마커 플래그(각 유닛 유형을 구분)가
발견되었나?
아니오
예
예
마커 플래그와 마지막으로 알려진 것
(처음인 경우 시퀀스의 시작 부분)
사이에서 정보를 추출함
분리 출력을 제출
예
아니오
RSF 또는 이 스캐너의
구문 오류는
디버깅을 진행
분리 출력
569
알 수 있고 인식되는
문제인가?
(i.e. 동작, 속성, 조건,
객체 등)
예
그것과 그것에 대응하는 관계를 분리
출력에 저장
분리된 규칙 형식
570
동작들
속성들
조건들
객체들
561
562
563
564

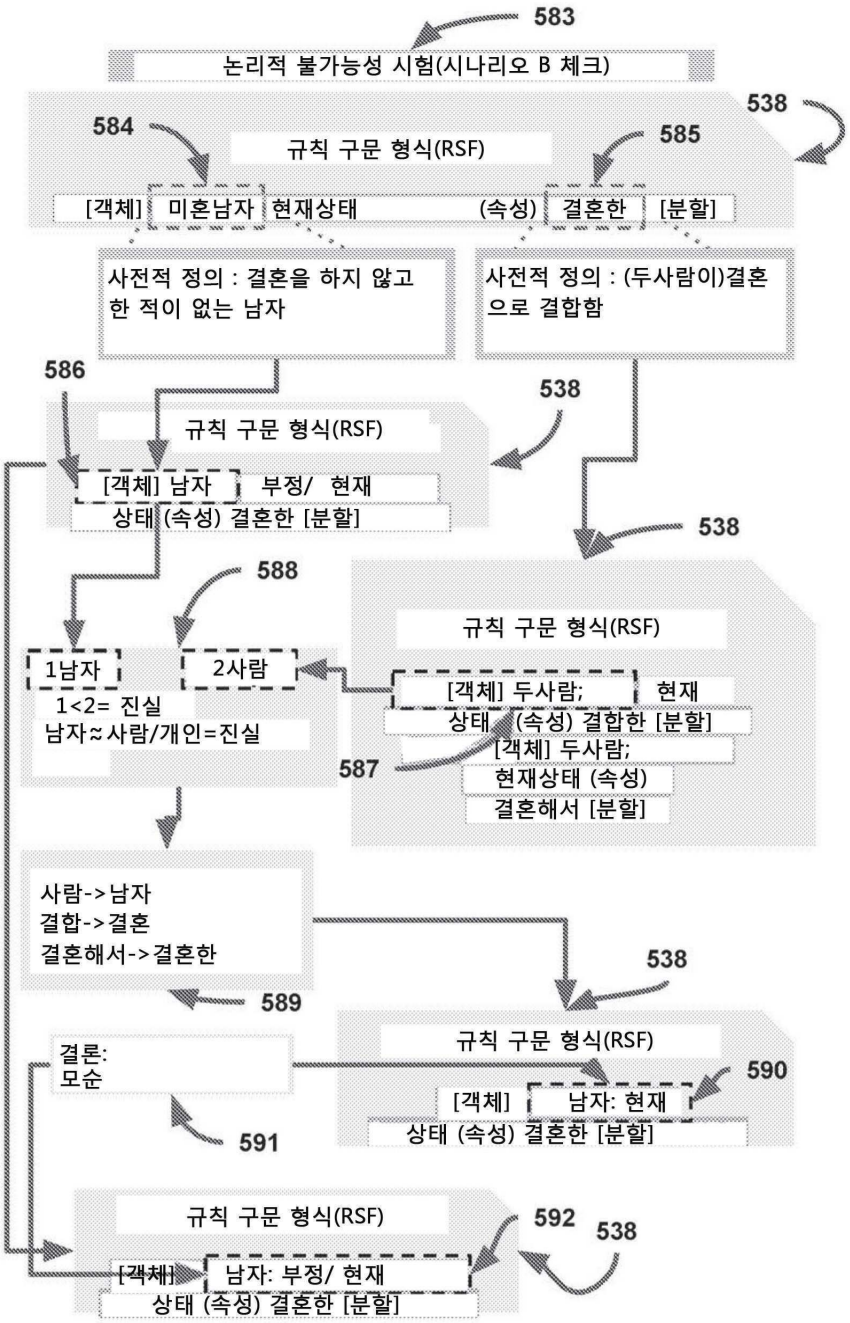
도면88


570
분리된 규칙 형식
동작들
561
속성들
562
조건들
563
객체들
564
구문적 관계 복원(SRR)
497
575
완전히 재구성된 규칙 세트 검색
522
실행된 규칙들
큐 관리(QM)
571
혼돈의 영역에서 발견되었는가?
객체 C
실행되지 않은 것으로 플래그된 규칙 세트
예
572
객체 C와 관련하여 혼돈의 영역에서 발견되었는가?
속성 A
아니오
실행되지 않은 것으로 플래그된 규칙 세트
예
573
객체 C와 관련하여 혼돈의 영역에서 실행되었는가?
동작 A
아니오
실행되지 않은 것으로 플래그된 규칙 세트
예
객체 C와 관련하여 혼돈의 영역에서 발생되었는가?
동작 A
아니오
실행되지 않은 것으로 플래그된 규칙 세트
574
예
실행되지 않은 것으로 플래그된 규칙 세트
576

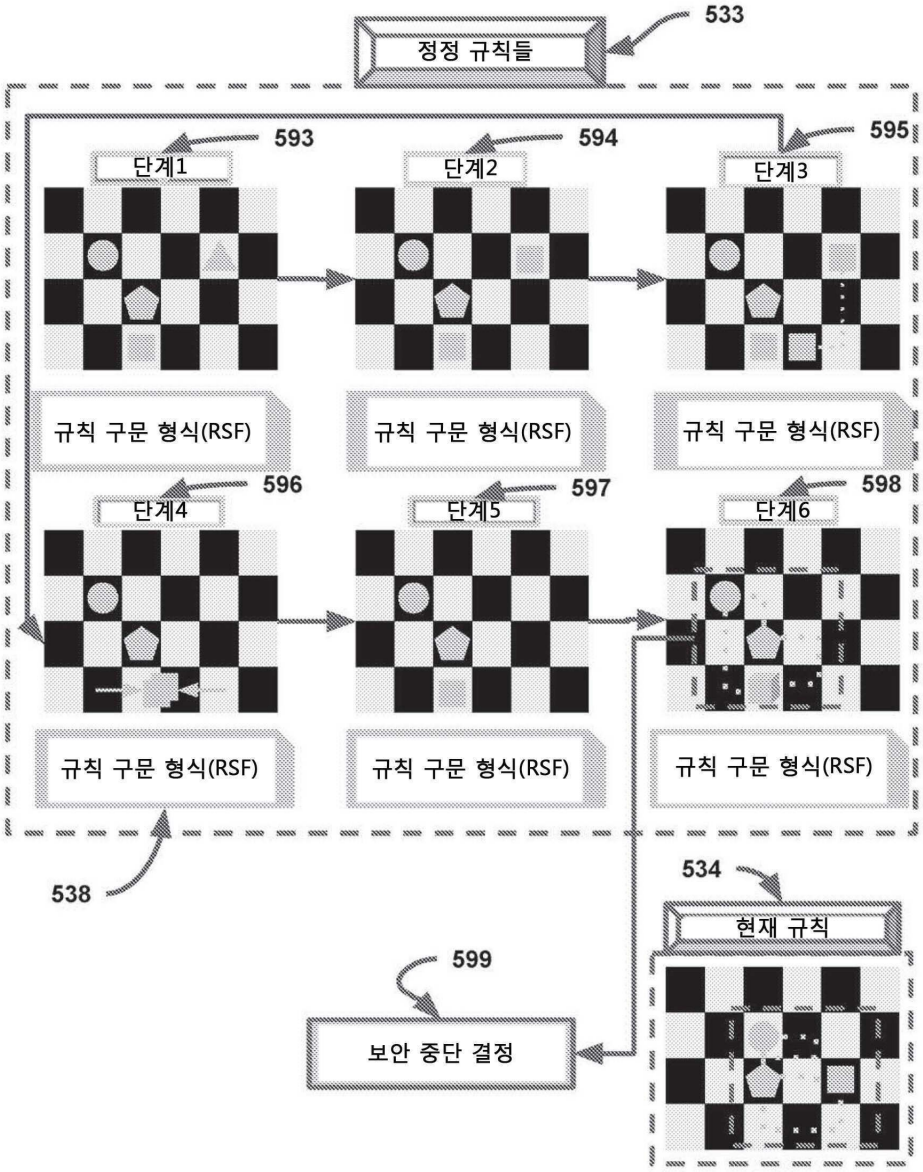
도면89


577
영역 희박성 조사 (시나리오 A 체크)
DHFKJDSHFKJGSHGOSIFHGWRHGFKJJHFD
JSDHFUEHFJSHDFUEOQIUJFRQPRUODPIJV
CKJNVKSJBTOMATOHFJDOFHIWERIWQRPU
EDIRWPIRUHQPORQOUROPIEUTPIWHGTDK
HSLKGFDHGJWUERUWQPETUIEANTETUEO
TIWUIOHGFJDVCBVMBDJGBWSBGUFRBGJR
SBGMVXCVERFEWPURPOWEULLANNCXNLS
IJDRFISDHJFGIWONFKNHWOINFKLNFSHGJ
HDJKLGHWGWFLCARROTUUAGBFUERGBK
ALGJBEF
578
579
규칙 부분 스캔
조사 데이터베이스
581
결론:
혼돈 영역 희박성은
(과실이 있음)
확실하지 않음
조사 데이터베이스는
포화 상태인가?
580
결론:
조사 조건이 불충분하기
때문에 테스트 중인 동일한
규칙과 관련하여 스캔되는
다른 혼돈의 영역을
기다려야함
582

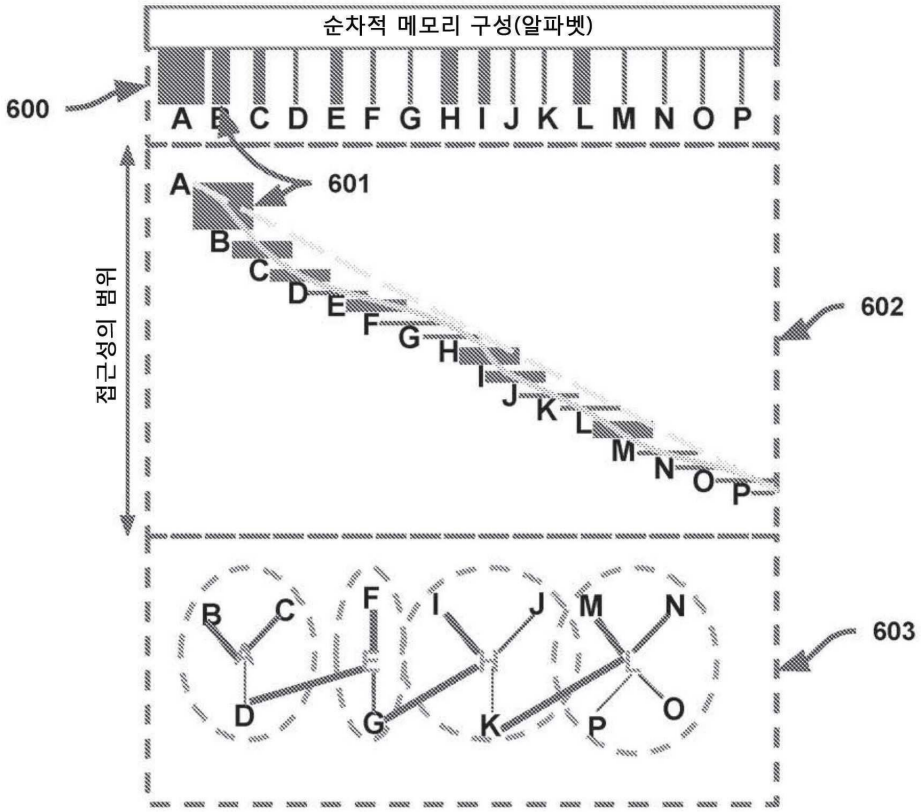
도면90


583
논리적 불가능성 시험(시나리오 B 체크)
538
584
규칙 구문 형식(RSF)
585
[객체] 미혼남자 현재상태 (속성) 결혼한 [분할]
사전적 정의 : 결혼을 하지 않고 한 적이 없는 남자
사전적 정의 : (두사람이)결혼으로 결합함
586
538
규칙 구문 형식(RSF)
[객체] 남자 부정/ 현재 상태 (속성) 결혼한 [분할]
538
588
규칙 구문 형식(RSF)
1남자
2사람
1<2= 진실
남자≈사람/개인=진실
[객체] 두사람; 현재 상태 (속성) 결합한 [분할]
587
[객체] 두사람; 현재상태 (속성) 결혼해서 [분할]
사람->남자
결합->결혼
결혼해서->결혼한
538
589
규칙 구문 형식(RSF)
결론:
모순
590
[객체] 남자: 현재 상태 (속성) 결혼한 [분할]
591
규칙 구문 형식(RSF)
592
538
[객체] 남자: 부정/ 현재 상태 (속성) 결혼한 [분할]

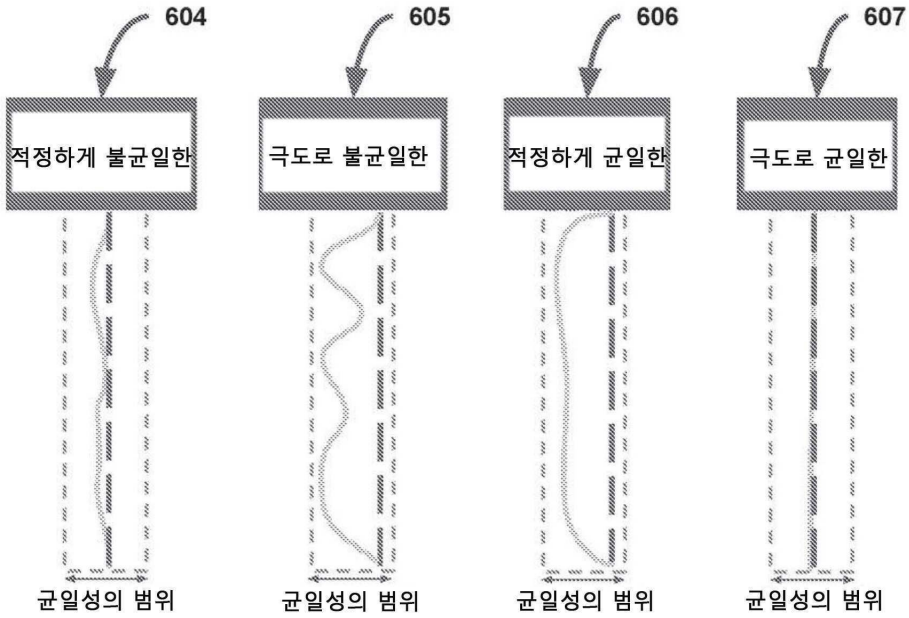
도면91


533
정정 규칙들
593
단계1
594
단계2
595
단계3
규칙 구문 형식(RSF)
규칙 구문 형식(RSF)
규칙 구문 형식(RSF)
596
단계4
597
단계5
598
단계6
규칙 구문 형식(RSF)
규칙 구문 형식(RSF)
규칙 구문 형식(RSF)
538
534
현재 규칙
599
보안 중단 결정

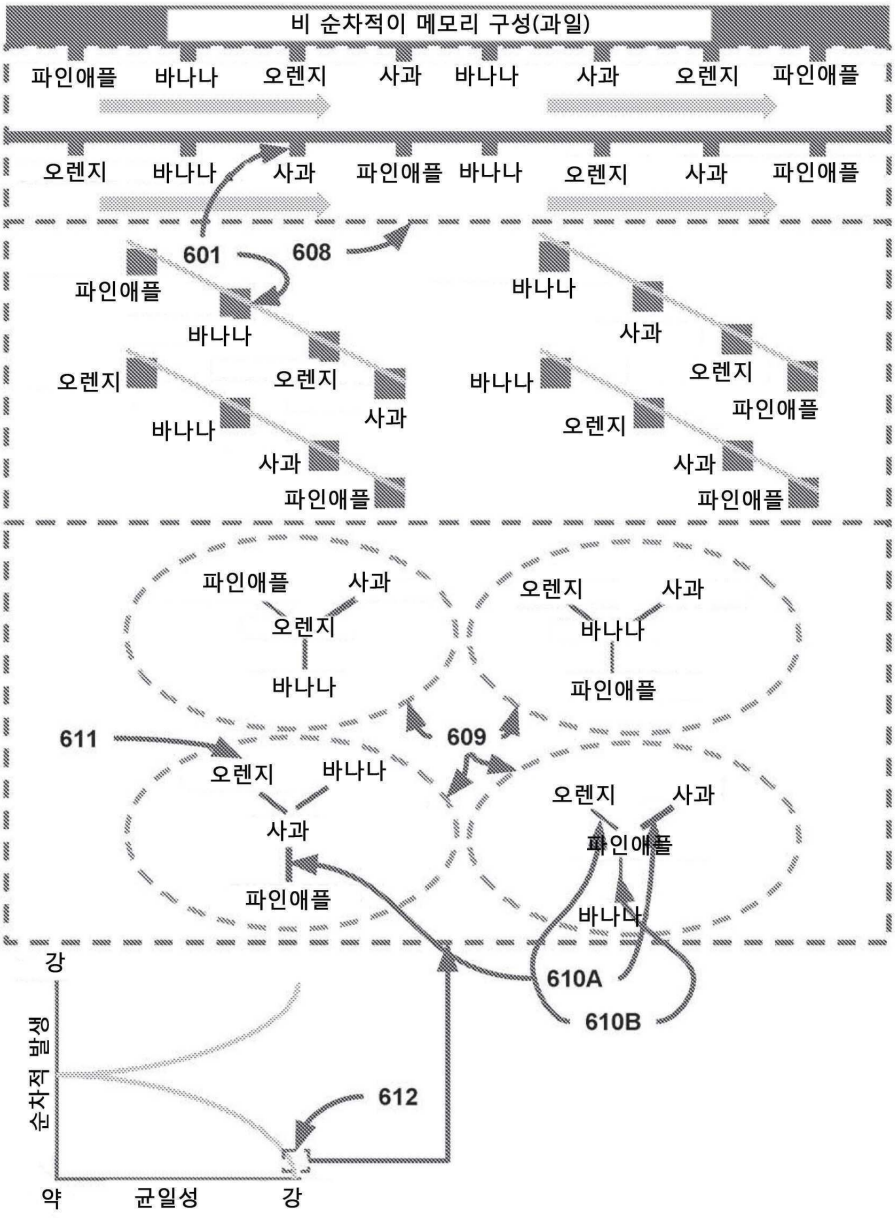
도면92


순차적 메모리 구성(알파벳)
600
A B C D E F G H I J K L M N O P
601
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
접근성의 범위
602
603

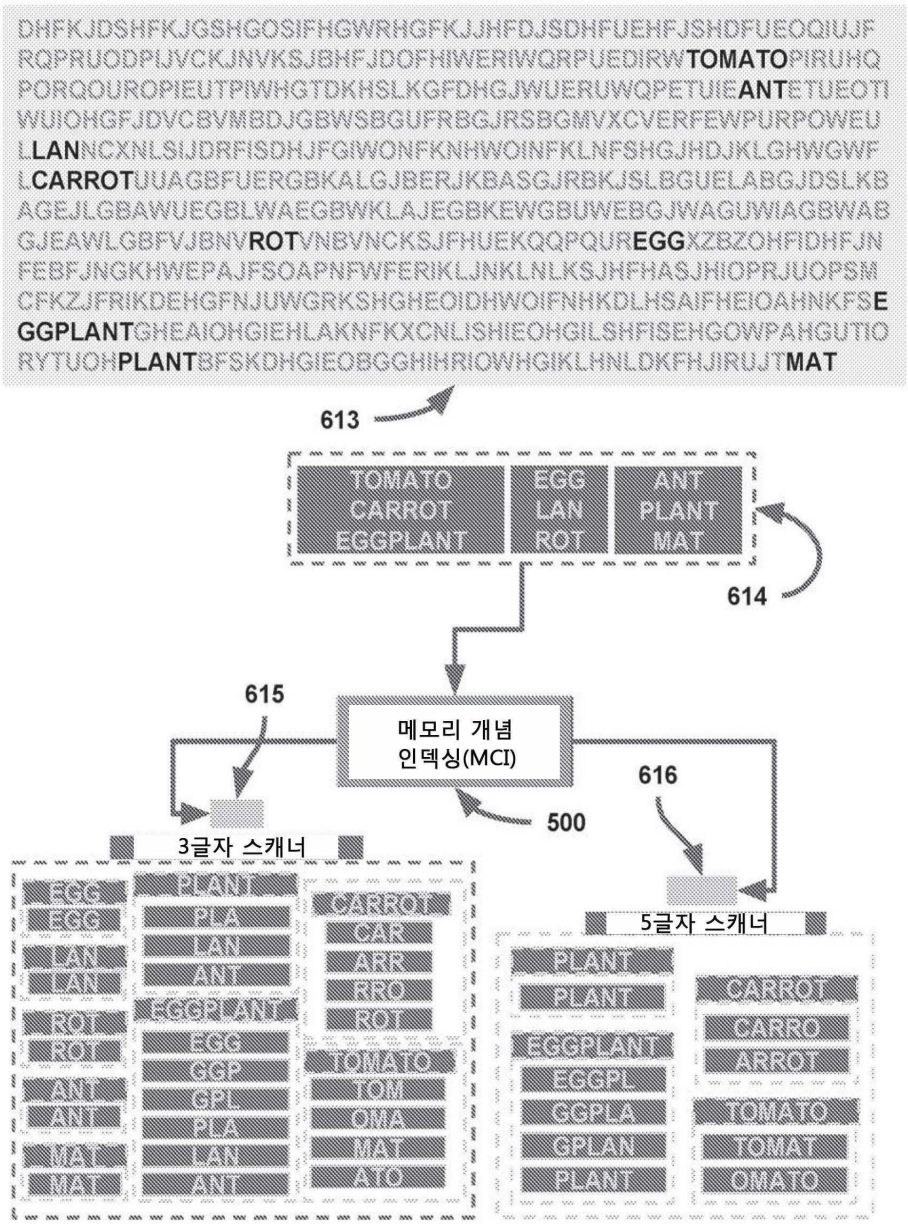
도면93


604
605
606
607
적정하게 불균일한
극도로 불균일한
적정하게 균일한
극도로 균일한
균일성의 범위
균일성의 범위
균일성의 범위
균일성의 범위

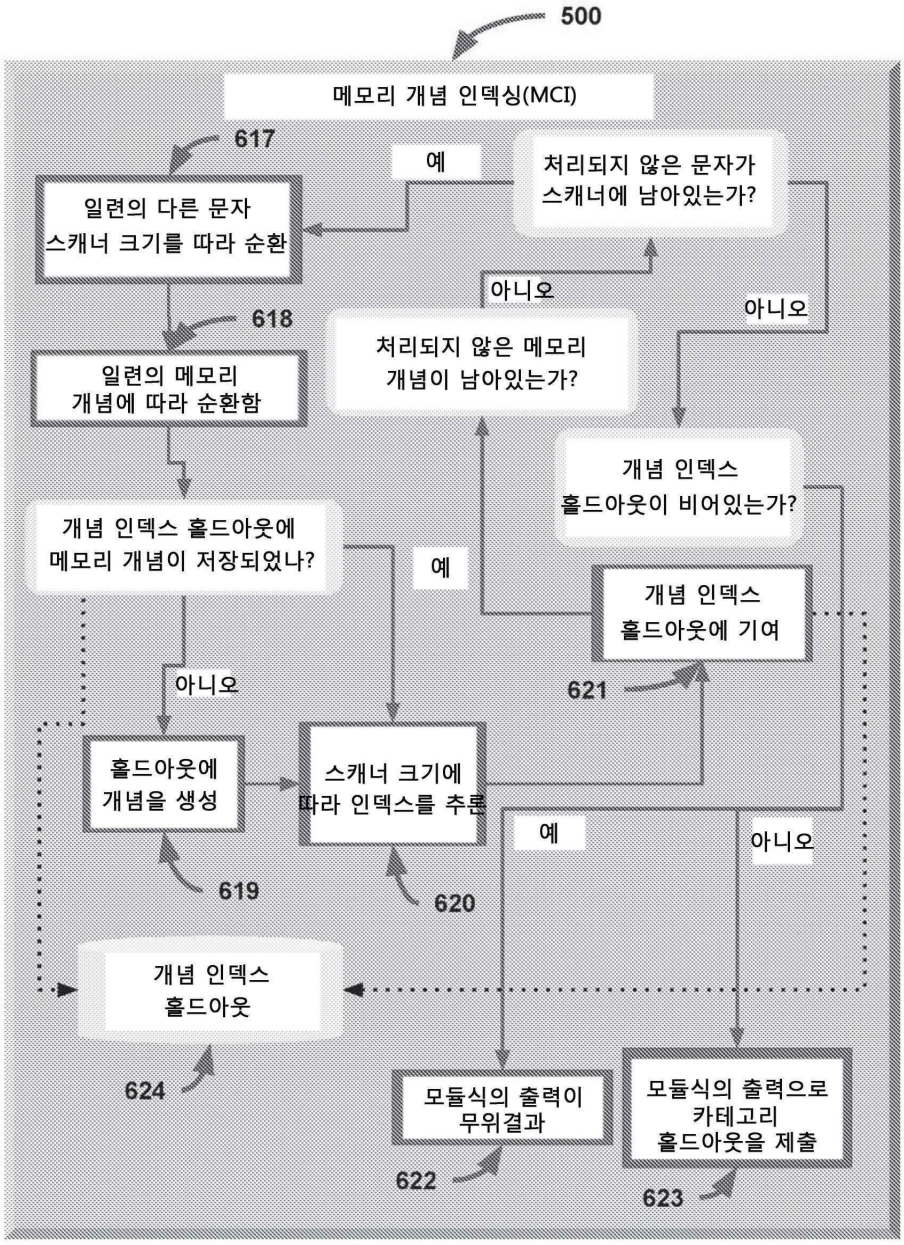
도면94


비 순차적이 메모리 구성(과일)
파인애플 바나나 오렌지 사과 바나나 사과 오렌지 파인애플
오렌지 바나나 사과 파인애플 바나나 오렌지 사과 파인애플
601
608
611
609
610A
610B
612
강
순차적 발생
약
균일성
강

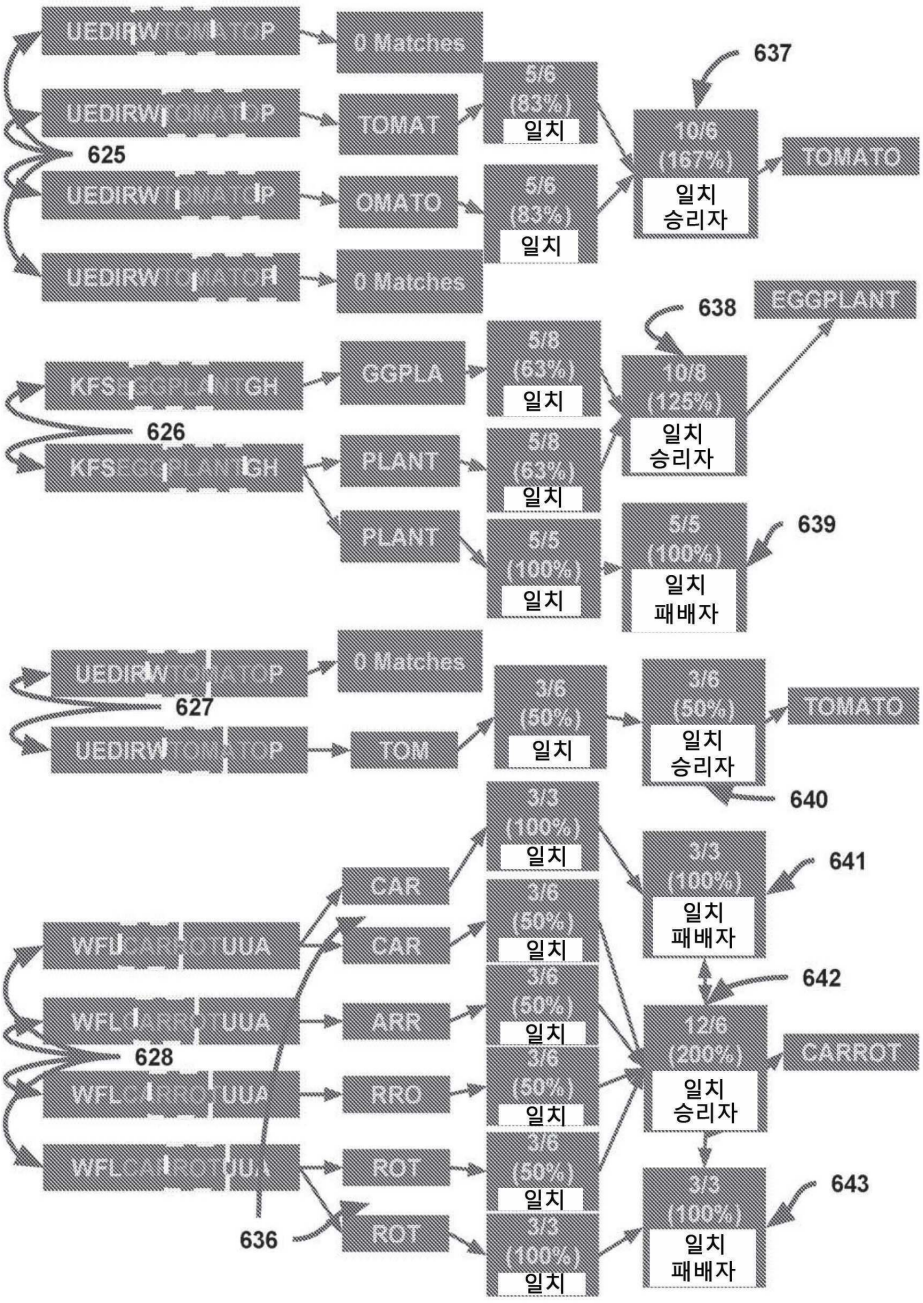
도면95


DHFKJDSHFKJGSHGOSIFHGWRHGFKJJHFDJSDHFUEHFJSHDFUEOQIUJF
RQPRUODPIJVCKJNVKSJBHFJDOFHIWERIWQRPUEDIRWTOMATOPIRUHQ
PORQOUROPIEUTPIWHGTDKHSLKGFDHGJWUERUWQPETUIEANTETUEOTI
WUIOHGFJDVCBVMBDJGBWSBGUFRBGJRSBGMVXCVERFEWPURPOWEU
LLANNCXNLSIJDRFISDHJFGIWONFKNHWOINFKLNFSHGJHDJKLGHWGWF
LCARROTUUAGBFUERGBKALGJBERJKBASGJRBKJSLBGUELABGJDSLKB
AGEJLGBAWUEGBLWAEGBWKLAJEGBKEWGBUWEBGJWAGUWIAGBWAB
GJEAWLGBFVJBNVROTVNBVNCKSJFHUEKQQPQUREGGXZBZOHFIDHFJN
FEBFJNGKHWEPAJFSOAPNFWFERIKLJNKLNLKSJHFHASJHIOPRJUOPSM
CFKZJFRIKDEHGFNJUWGRKSHGHEOIDHWOIFNHKDLHSAIFHEIOAHNKFSE
GGPLANTGHEAIOHGIEHLAKNFKXCNLISHIEOHGILSHFISEHGOWPAHGUTIO
RYTUOHPLANTBFSKDHGIEOBGGHIHRIOWHGIKLHNLDKFHJIRUJTMAT
613
TOMATO
CARROT
EGGPLANT
EGG
LAN
ROT
ANT
PLANT
MAT
614
615
메모리 개념
인덱싱(MCI)
616
500
3글자 스캐너
5글자 스캐너

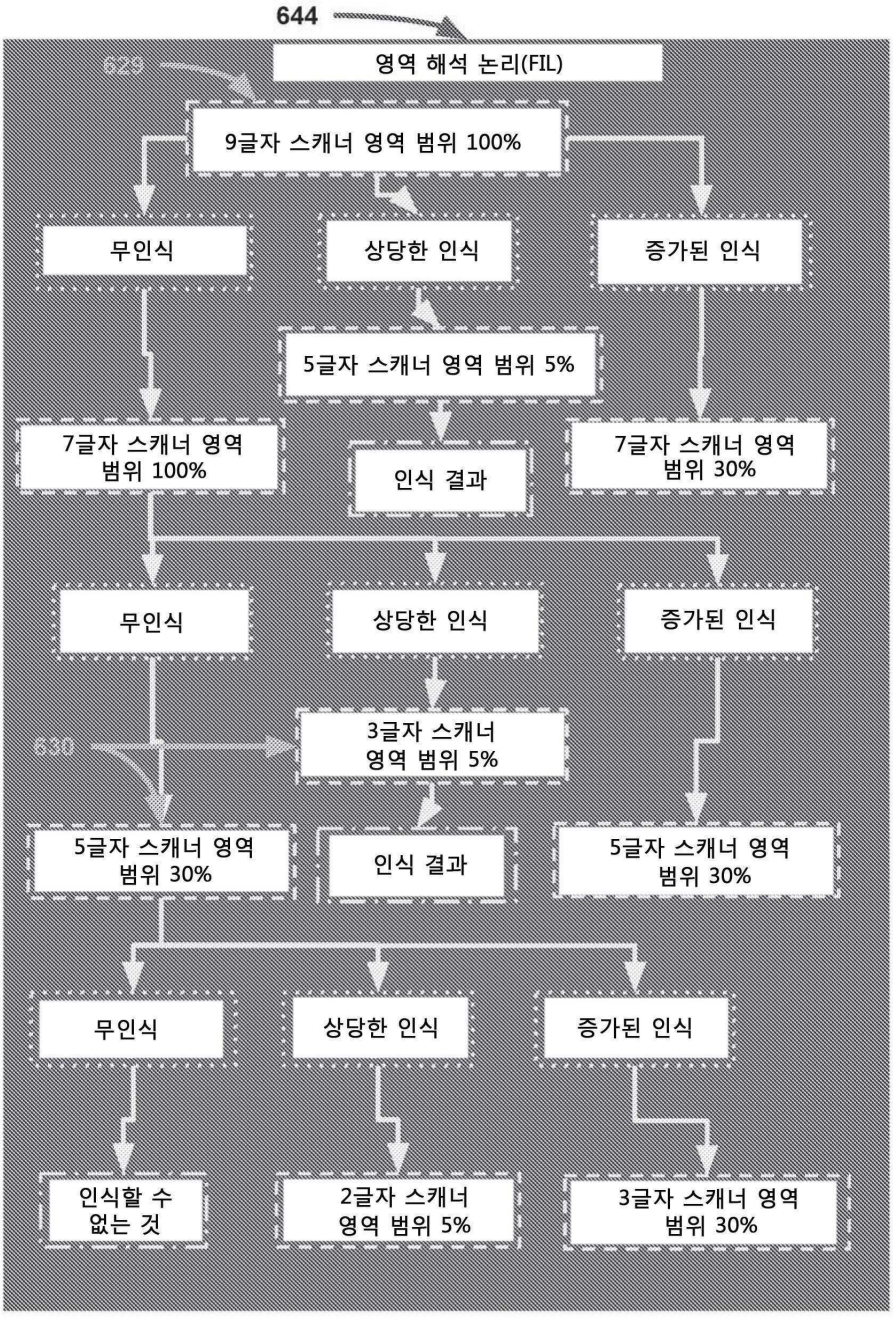
도면96


500
메모리 개념 인덱싱(MCI)
617
일련의 다른 문자 스캐너 크기를 따라 순환
618
일련의 메모리 개념에 따라 순환함
개념 인덱스 홀드아웃에 메모리 개념이 저장되었나?
예
아니오
홀드아웃에 개념을 생성
619
스캐너 크기에 따라 인덱스를 추론
620
개념 인덱스 홀드아웃
624
처리되지 않은 문자가 스캐너에 남아있는가?
예
아니오
처리되지 않은 메모리 개념이 남아있는가?
개념 인덱스 홀드아웃이 비어있는가?
예
아니오
개념 인덱스 홀드아웃에 기여
621
모듈식의 출력이 무위결과
622
모듈식의 출력으로 카테고리 홀드아웃을 제출
623

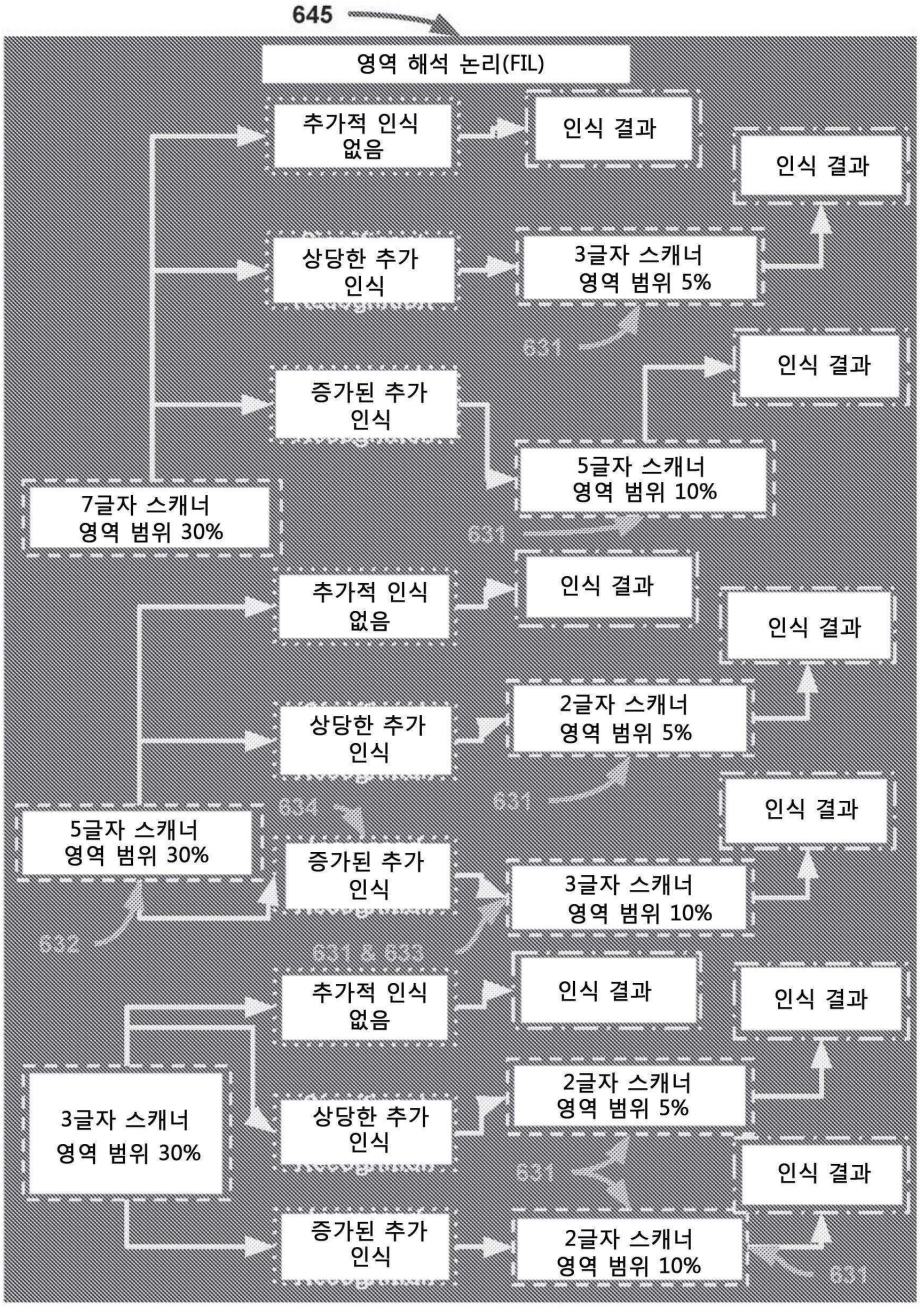
도면97


0 Matches
TOMAT
OMATO
0 Matches
625
5/6 (83%) 일치
5/6 (83%) 일치
637
10/6 (167%) 일치 승리자
TOMATO
GGPLA
PLANT
PLANT
626
5/8 (63%) 일치
5/8 (63%) 일치
5/5 (100%) 일치
638
10/8 (125%) 일치 승리자
EGGPLANT
5/5 (100%) 일치 패배자
639
0 Matches
TOM
627
3/6 (50%) 일치
3/6 (50%) 일치 승리자
TOMATO
640
CAR
CAR
ARR
RRO
ROT
ROT
628
636
3/3 (100%) 일치
3/6 (50%) 일치
3/6 (50%) 일치
3/6 (50%) 일치
3/6 (50%) 일치
3/3 (100%) 일치
3/3 (100%) 일치 패배자
641
642
12/6 (200%) 일치 승리자
CARROT
3/3 (100%) 일치 패배자
643

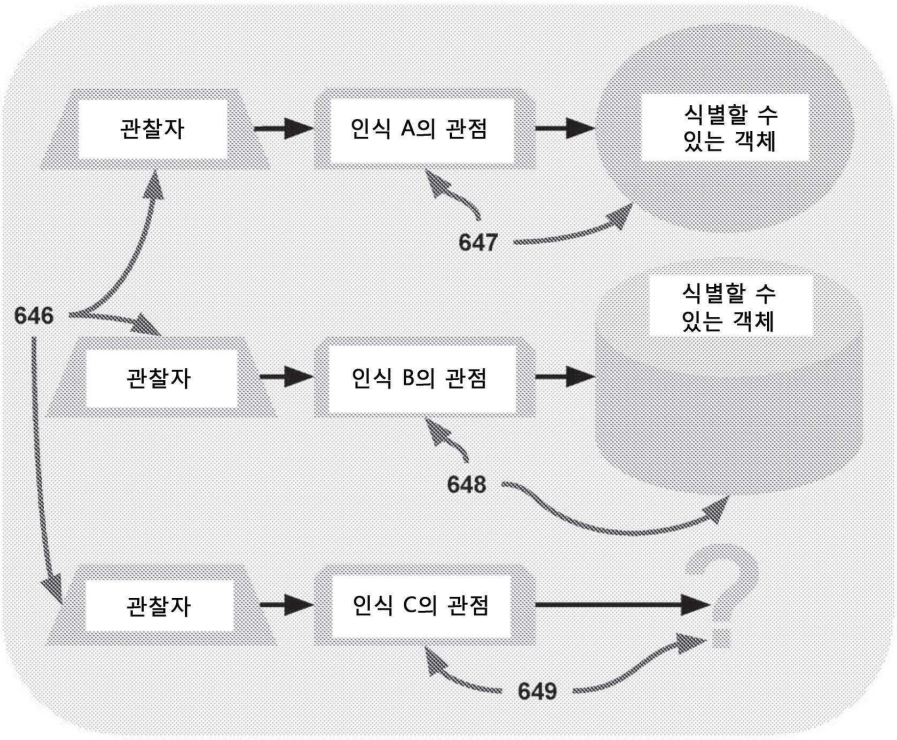
도면98


644
629
영역 해석 논리(FIL)
9글자 스캐너 영역 범위 100%
무인식
상당한 인식
증가된 인식
5글자 스캐너 영역 범위 5%
7글자 스캐너 영역 범위 100%
인식 결과
7글자 스캐너 영역 범위 30%
무인식
상당한 인식
증가된 인식
630
3글자 스캐너 영역 범위 5%
5글자 스캐너 영역 범위 30%
인식 결과
5글자 스캐너 영역 범위 30%
무인식
상당한 인식
증가된 인식
인식할 수 없는 것
2글자 스캐너 영역 범위 5%
3글자 스캐너 영역 범위 30%

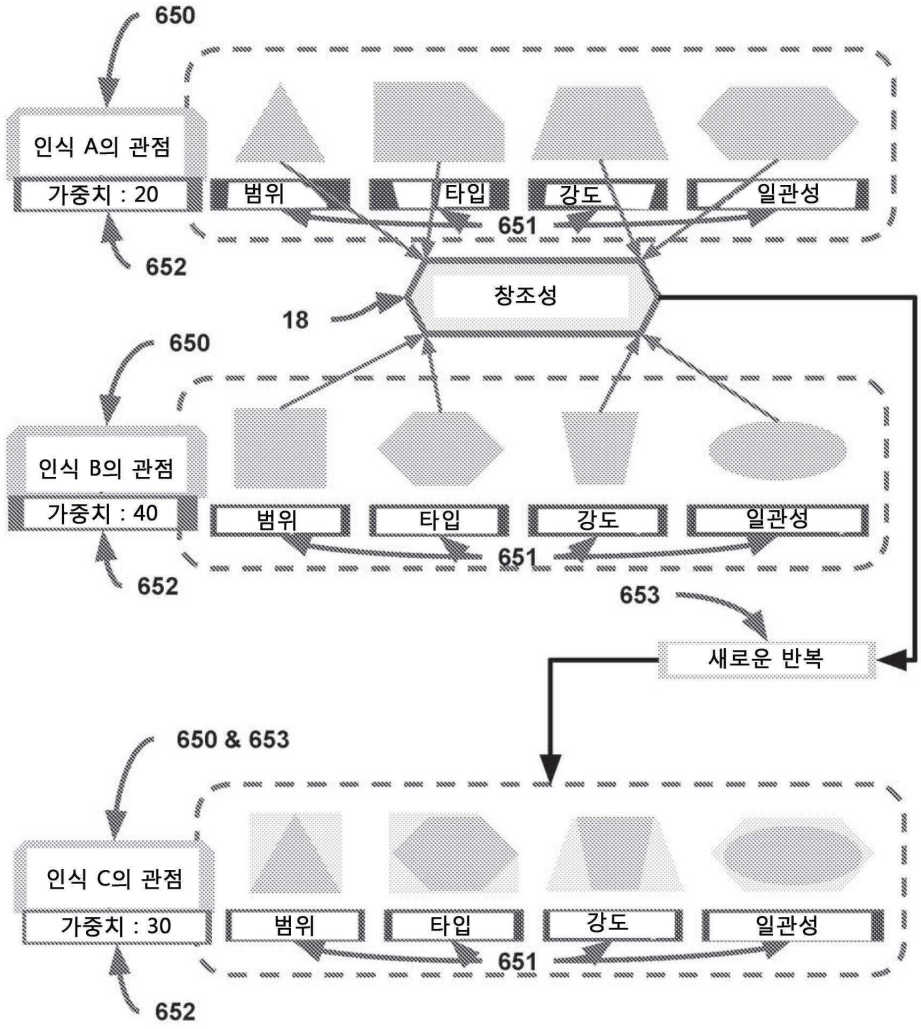
도면99


645
영역 해석 논리(FIL)
추가적 인식 없음
인식 결과
인식 결과
상당한 추가 인식
3글자 스캐너 영역 범위 5%
631
인식 결과
증가된 추가 인식
5글자 스캐너 영역 범위 10%
7글자 스캐너 영역 범위 30%
631
추가적 인식 없음
인식 결과
인식 결과
상당한 추가 인식
2글자 스캐너 영역 범위 5%
634
631
인식 결과
5글자 스캐너 영역 범위 30%
증가된 추가 인식
3글자 스캐너 영역 범위 10%
632
631 & 633
추가적 인식 없음
인식 결과
인식 결과
3글자 스캐너 영역 범위 30%
상당한 추가 인식
2글자 스캐너 영역 범위 5%
631
인식 결과
증가된 추가 인식
2글자 스캐너 영역 범위 10%
631

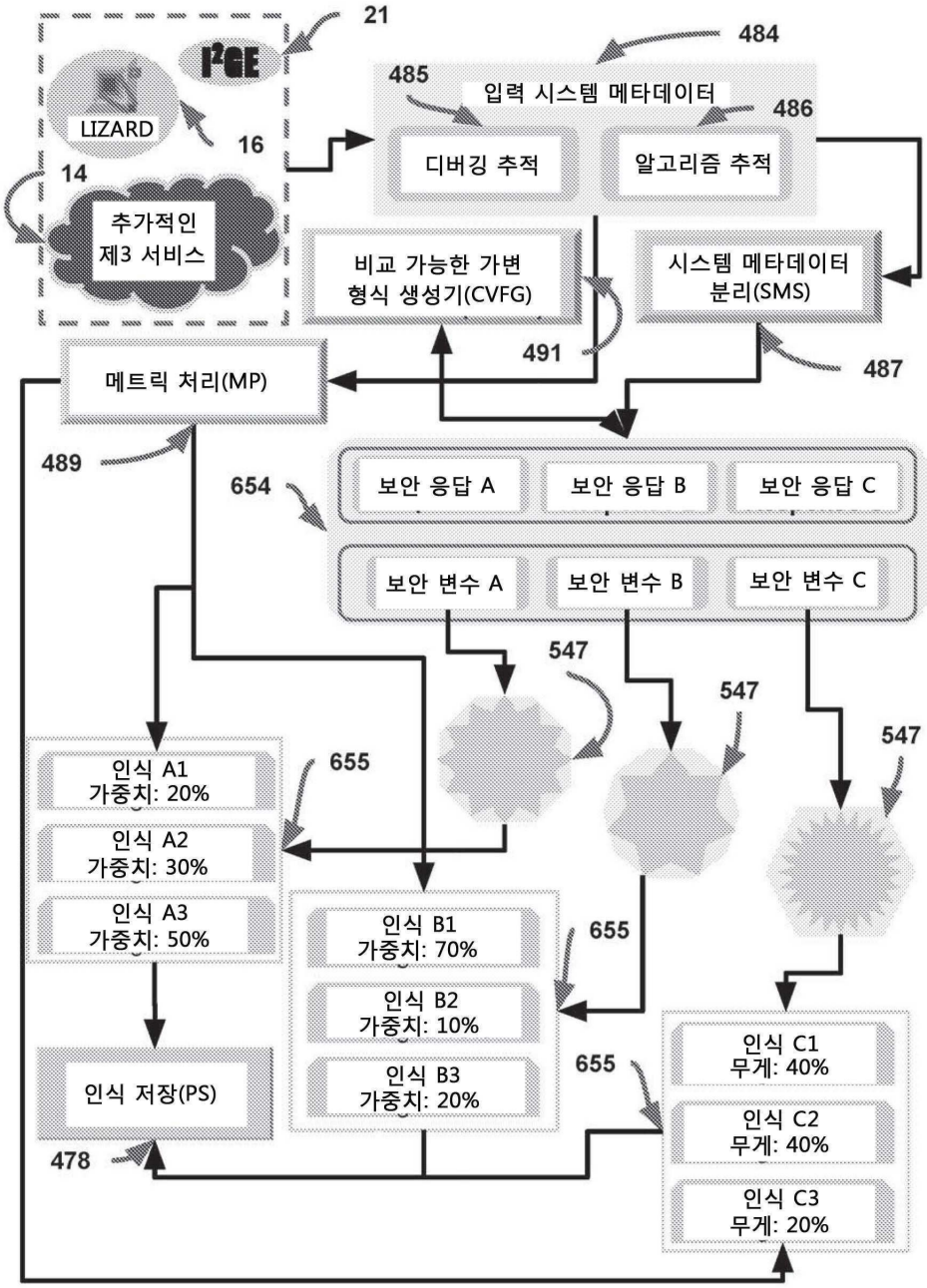
도면100


관찰자
인식 A의 관점
식별할 수
있는 객체
647
646
식별할 수
있는 객체
관찰자
인식 B의 관점
648
관찰자
인식 C의 관점
?
649

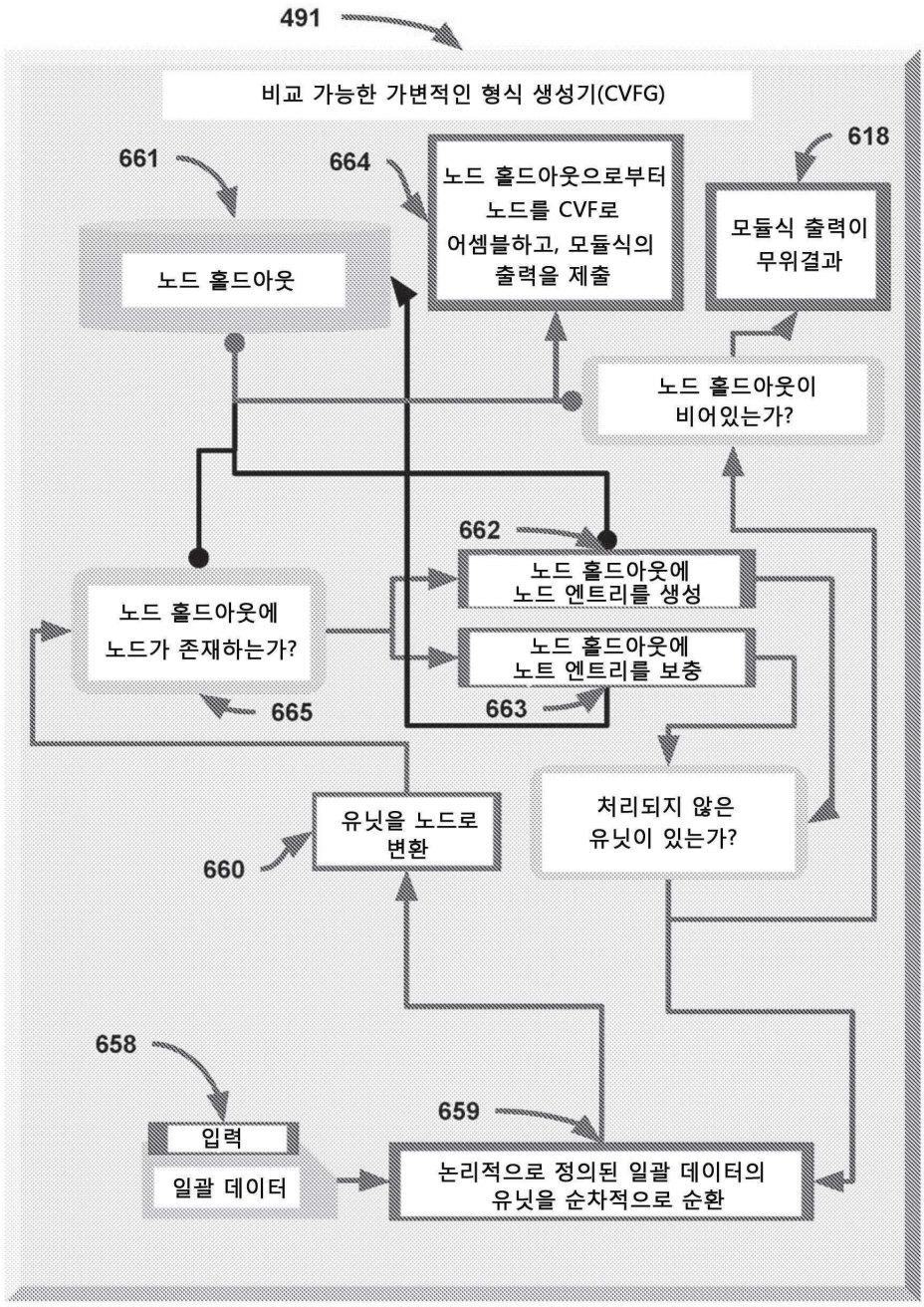
도면101


650
인식 A의 관점
가중치 : 20
범위
타입
강도
일관성
651
652
18
창조성
650
인식 B의 관점
가중치 : 40
범위
타입
강도
일관성
651
652
653
새로운 반복
650 & 653
인식 C의 관점
가중치 : 30
범위
타입
강도
일관성
651
652

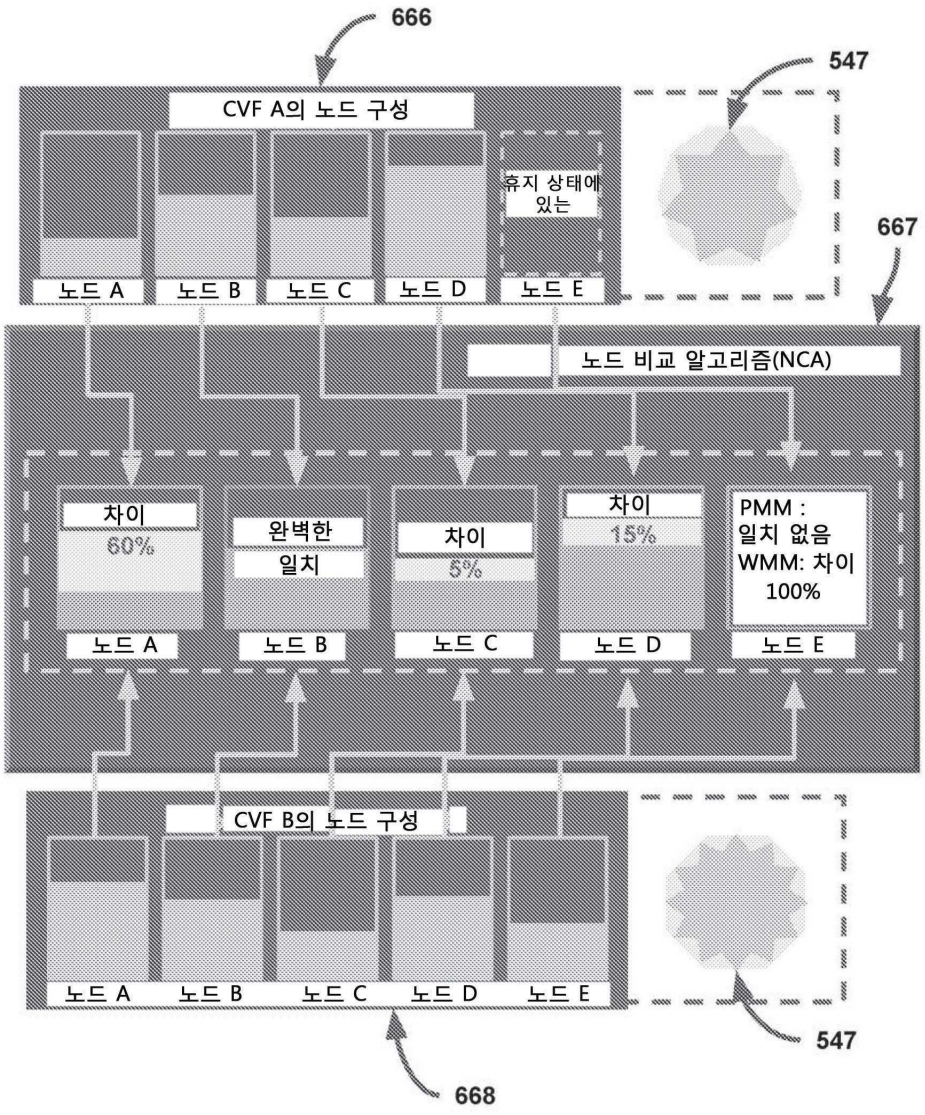
도면102


21
I²GE
LIZARD
16
14
추가적인
제3 서비스
484
485
입력 시스템 메타데이터
486
디버깅 추적
알고리즘 추적
비교 가능한 가변
형식 생성기(CVFG)
시스템 메타데이터
분리(SMS)
491
487
메트릭 처리(MP)
489
654
보안 응답 A
보안 응답 B
보안 응답 C
보안 변수 A
보안 변수 B
보안 변수 C
547
547
547
655
인식 A1
가중치: 20%
인식 A2
가중치: 30%
인식 A3
가중치: 50%
인식 B1
가중치: 70%
인식 B2
가중치: 10%
인식 B3
가중치: 20%
655
655
인식 C1
무게: 40%
인식 C2
무게: 40%
인식 C3
무게: 20%
인식 저장(PS)
478

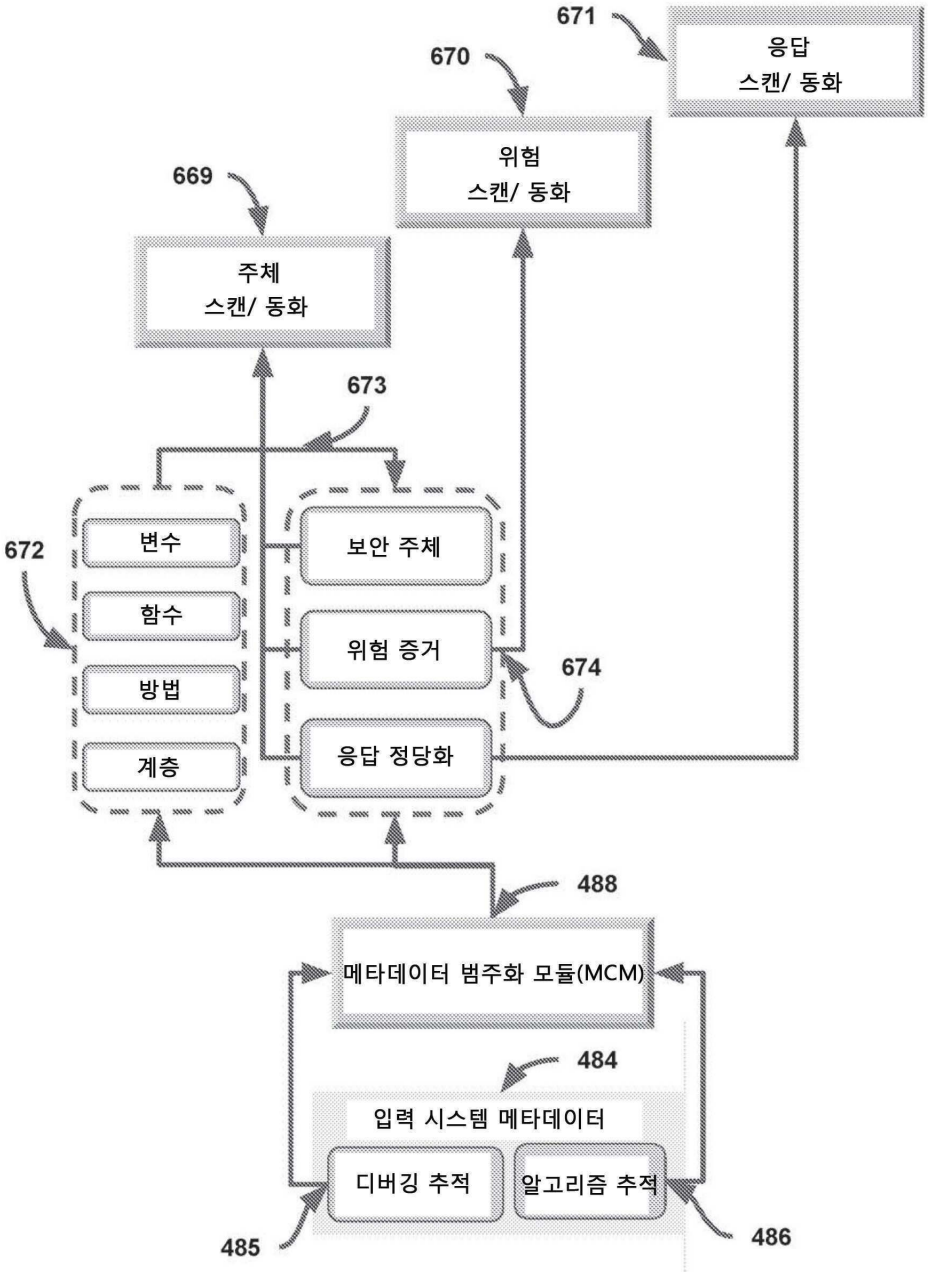
도면103


491
비교 가능한 가변적인 형식 생성기(CVFG)
661
노드 홀드아웃
664
노드 홀드아웃으로부터 노드를 CVF로 어셈블하고, 모듈식의 출력을 제출
618
모듈식 출력이 무위결과
노드 홀드아웃이 비어있는가?
662
노드 홀드아웃에 노드 엔트리를 생성
노드 홀드아웃에 노드가 존재하는가?
노드 홀드아웃에 노트 엔트리를 보충
665
663
처리되지 않은 유닛이 있는가?
유닛을 노드로 변환
660
658
입력
일괄 데이터
659
논리적으로 정의된 일괄 데이터의 유닛을 순차적으로 순환

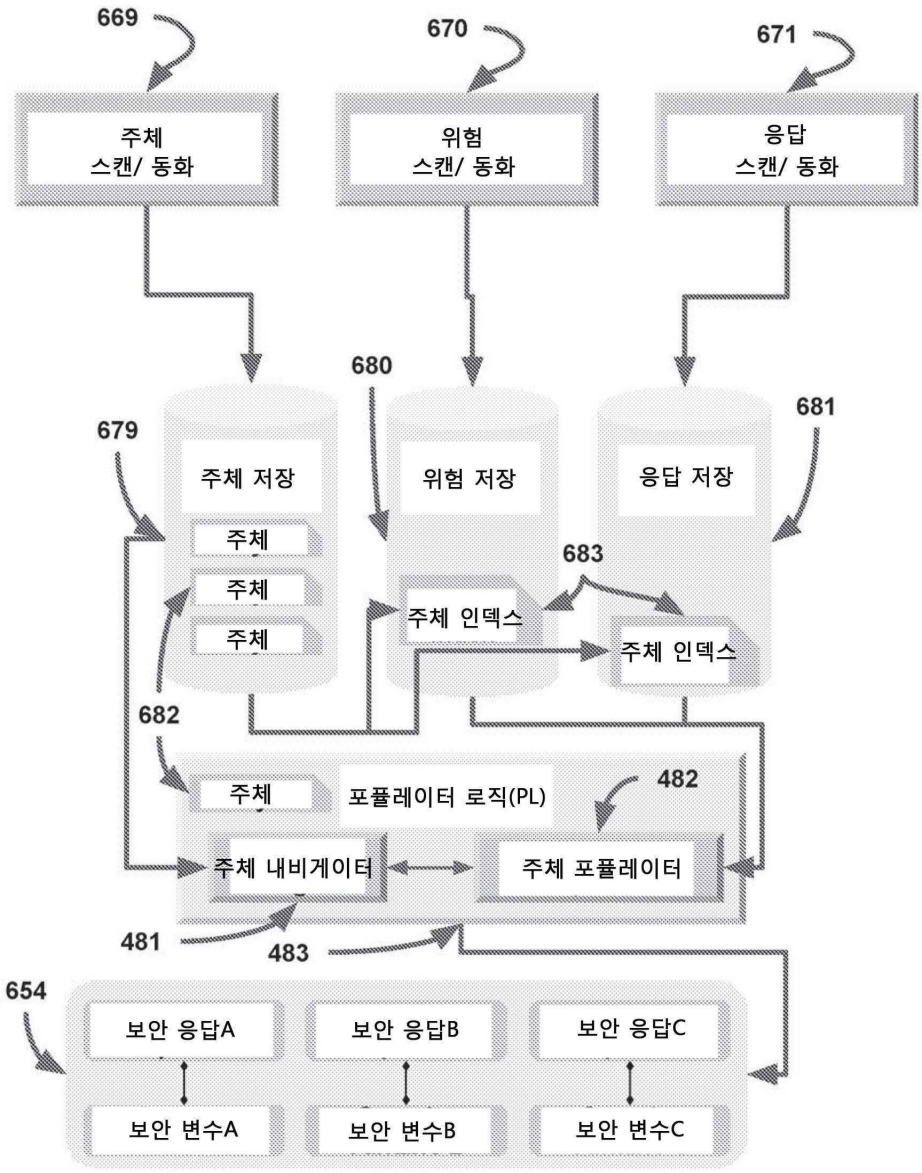
도면104


666
547
CVF A의 노드 구성
휴지 상태에 있는
667
노드 A
노드 B
노드 C
노드 D
노드 E
노드 비교 알고리즘(NCA)
차이
60%
완벽한
일치
차이
5%
차이
15%
PMM :
일치 없음
WMM: 차이
100%
노드 A
노드 B
노드 C
노드 D
노드 E
CVF B의 노드 구성
노드 A
노드 B
노드 C
노드 D
노드 E
547
668

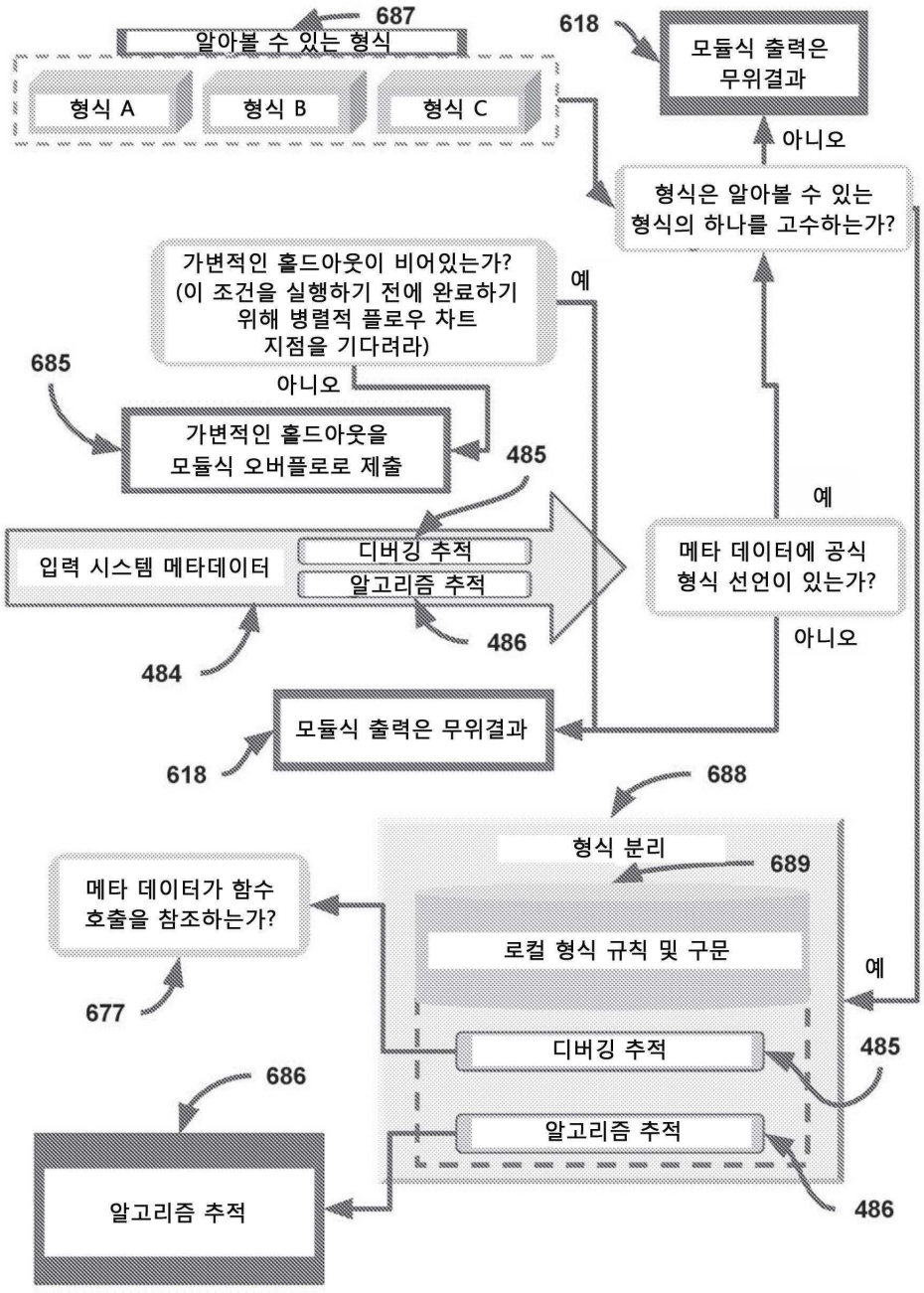
도면105


671
응답
스캔/ 동화
670
위험
스캔/ 동화
669
주체
스캔/ 동화
673
672
변수
함수
방법
계층
보안 주체
위험 증거
674
응답 정당화
488
메타데이터 범주화 모듈(MCM)
484
입력 시스템 메타데이터
디버깅 추적
알고리즘 추적
485
486

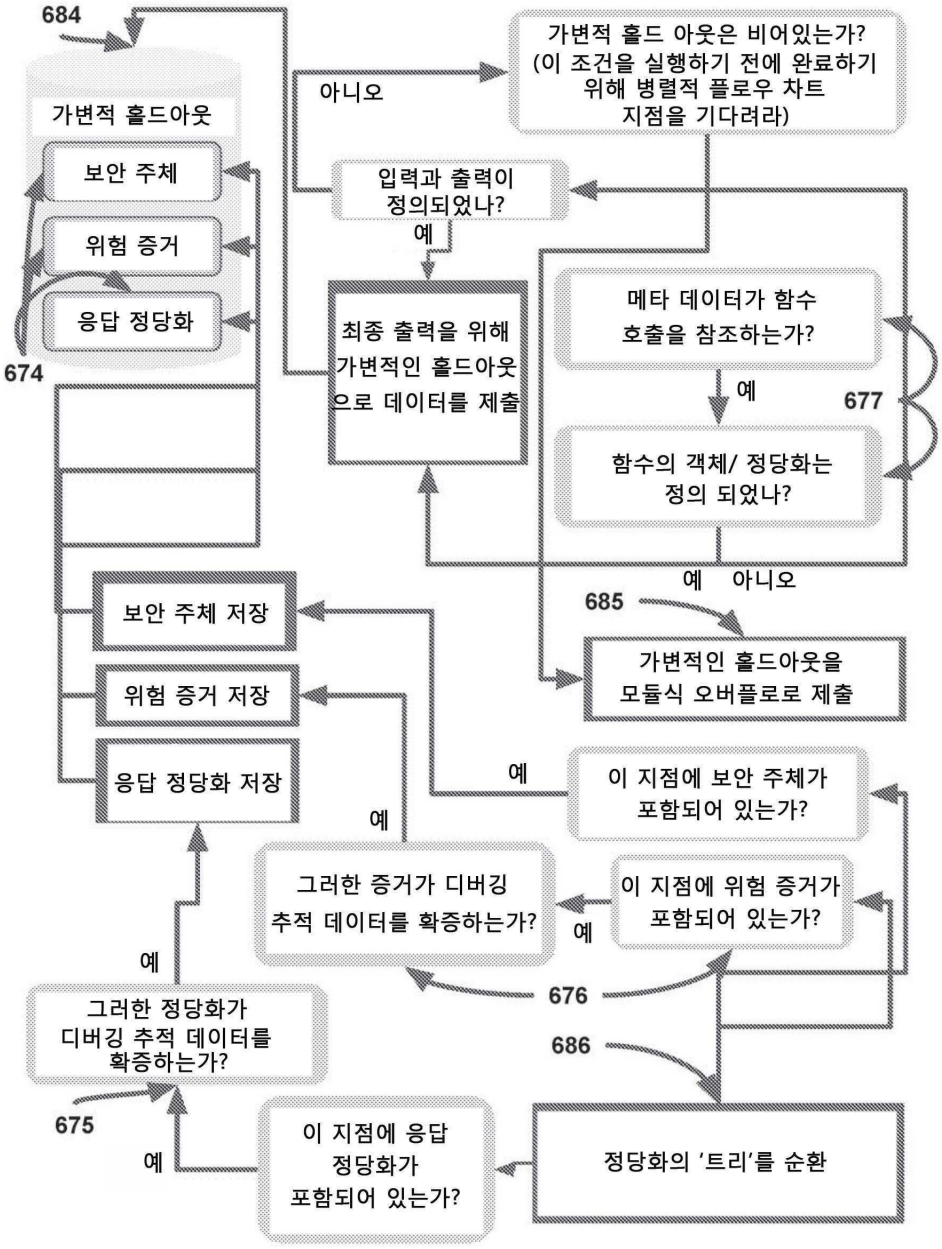
도면106


669
670
671
주체
스캔/ 동화
위험
스캔/ 동화
응답
스캔/ 동화
680
679
681
주체 저장
위험 저장
응답 저장
주체
주체
주체
683
주체 인덱스
주체 인덱스
682
주체
포퓰레이터 로직(PL)
482
주체 내비게이터
주체 포퓰레이터
481
483
654
보안 응답A
보안 응답B
보안 응답C
보안 변수A
보안 변수B
보안 변수C

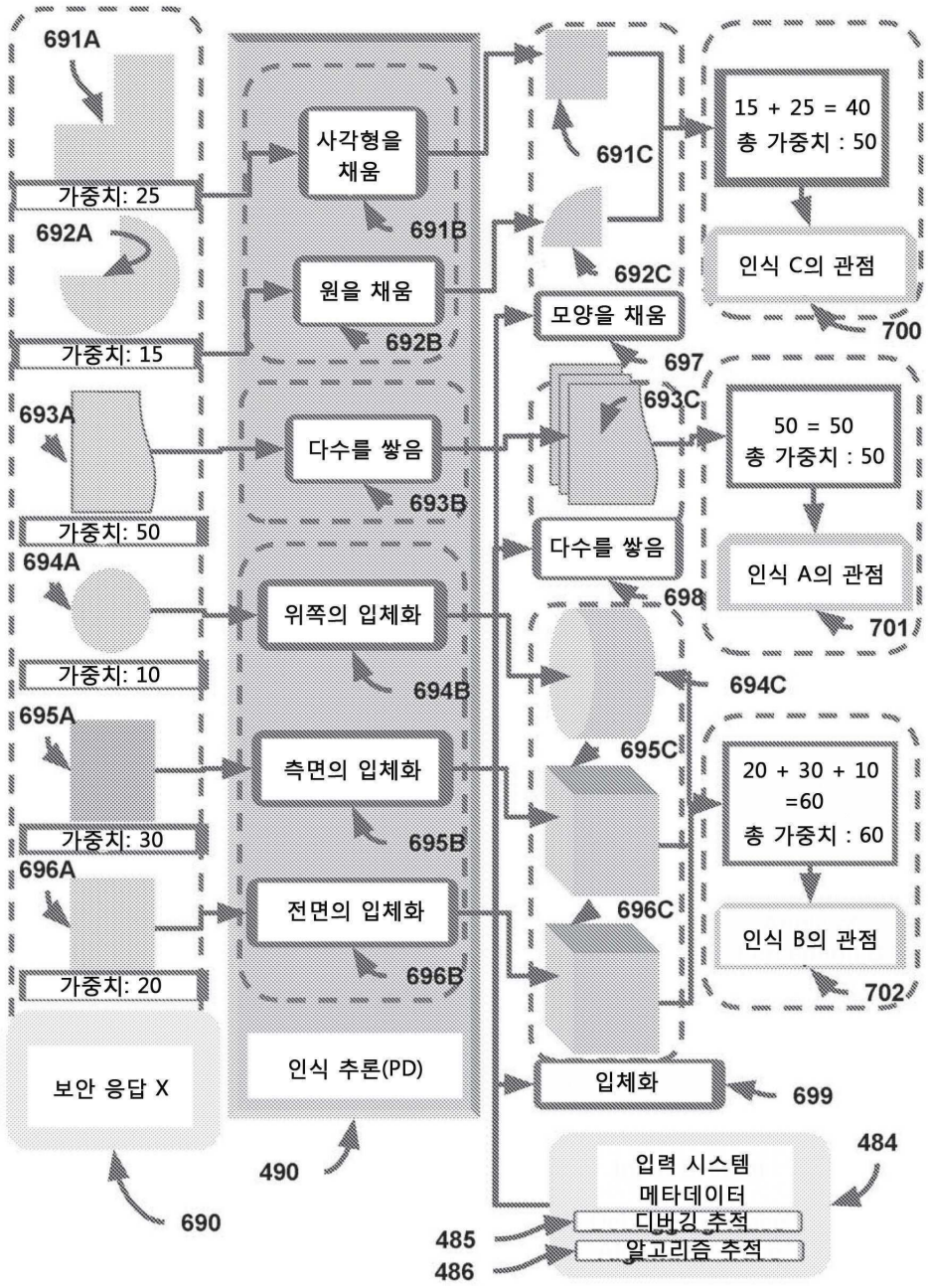
도면107


687
알아볼 수 있는 형식
형식 A
형식 B
형식 C
618
모듈식 출력은 무위결과
아니오
형식은 알아볼 수 있는 형식의 하나를 고수하는가?
예
가변적인 홀드아웃이 비어있는가? (이 조건을 실행하기 전에 완료하기 위해 병렬적 플로우 차트 지점을 기다려라)
아니오
685
가변적인 홀드아웃을 모듈식 오버플로로 제출
485
예
입력 시스템 메타데이터
디버깅 추적
알고리즘 추적
486
484
메타 데이터에 공식 형식 선언이 있는가?
아니오
618
모듈식 출력은 무위결과
688
형식 분리
689
메타 데이터가 함수 호출을 참조하는가?
677
로컬 형식 규칙 및 구문
예
디버깅 추적
485
686
알고리즘 추적
알고리즘 추적
486

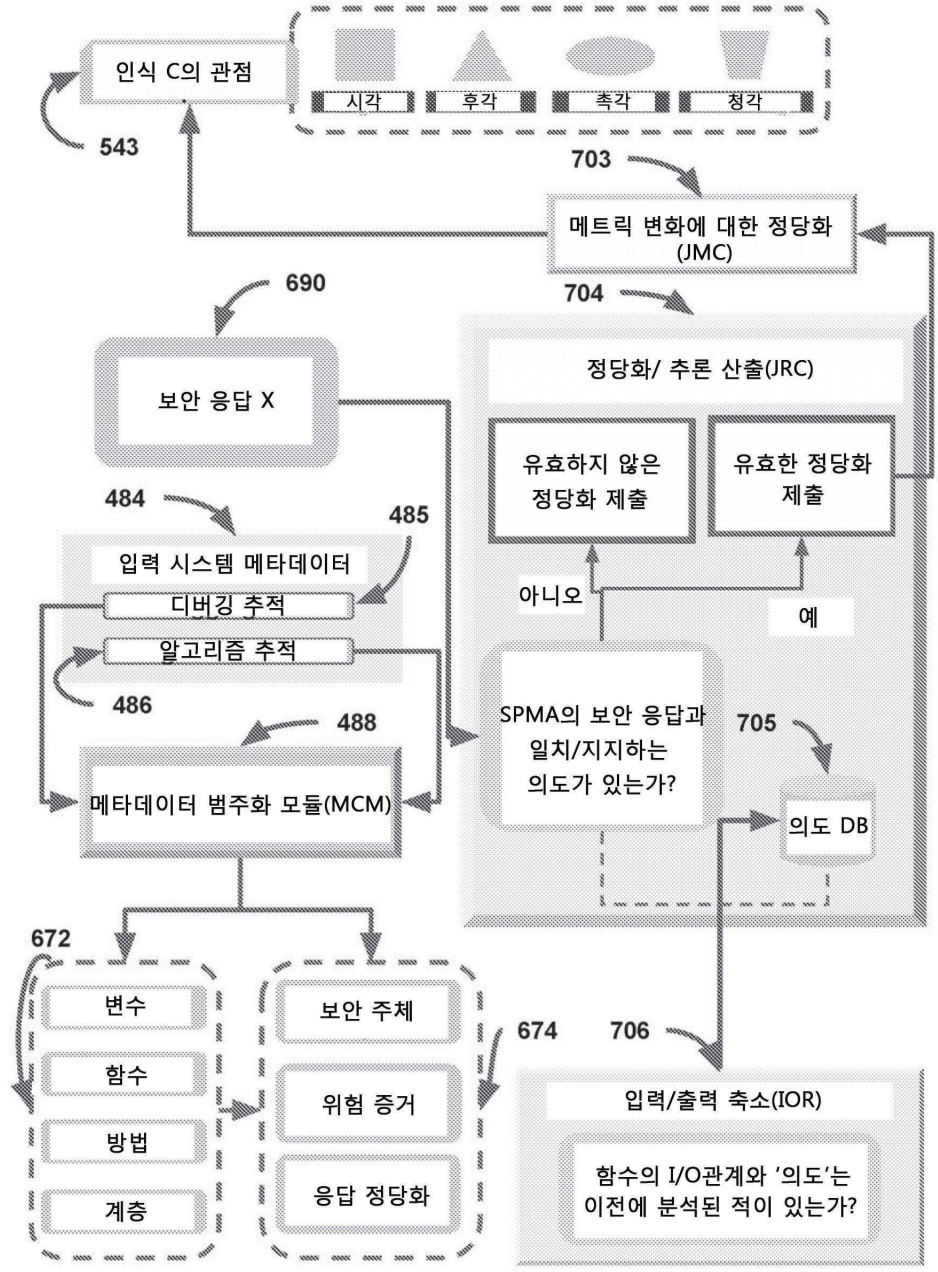
도면108


684
가변적 홀드아웃
보안 주체
위험 증거
응답 정당화
674
아니오
가변적 홀드 아웃은 비어있는가?
(이 조건을 실행하기 전에 완료하기 위해 병렬적 플로우 차트 지점을 기다려라)
입력과 출력이 정의되었나?
예
최종 출력을 위해 가변적인 홀드아웃으로 데이터를 제출
메타 데이터가 함수 호출을 참조하는가?
예
677
함수의 객체/ 정당화는 정의 되었나?
예
아니오
685
가변적인 홀드아웃을 모듈식 오버플로로 제출
보안 주체 저장
위험 증거 저장
응답 정당화 저장
예
이 지점에 보안 주체가 포함되어 있는가?
예
그러한 증거가 디버깅 추적 데이터를 확증하는가?
예
이 지점에 위험 증거가 포함되어 있는가?
676
686
예
그러한 정당화가 디버깅 추적 데이터를 확증하는가?
675
예
이 지점에 응답 정당화가 포함되어 있는가?
정당화의 '트리'를 순환

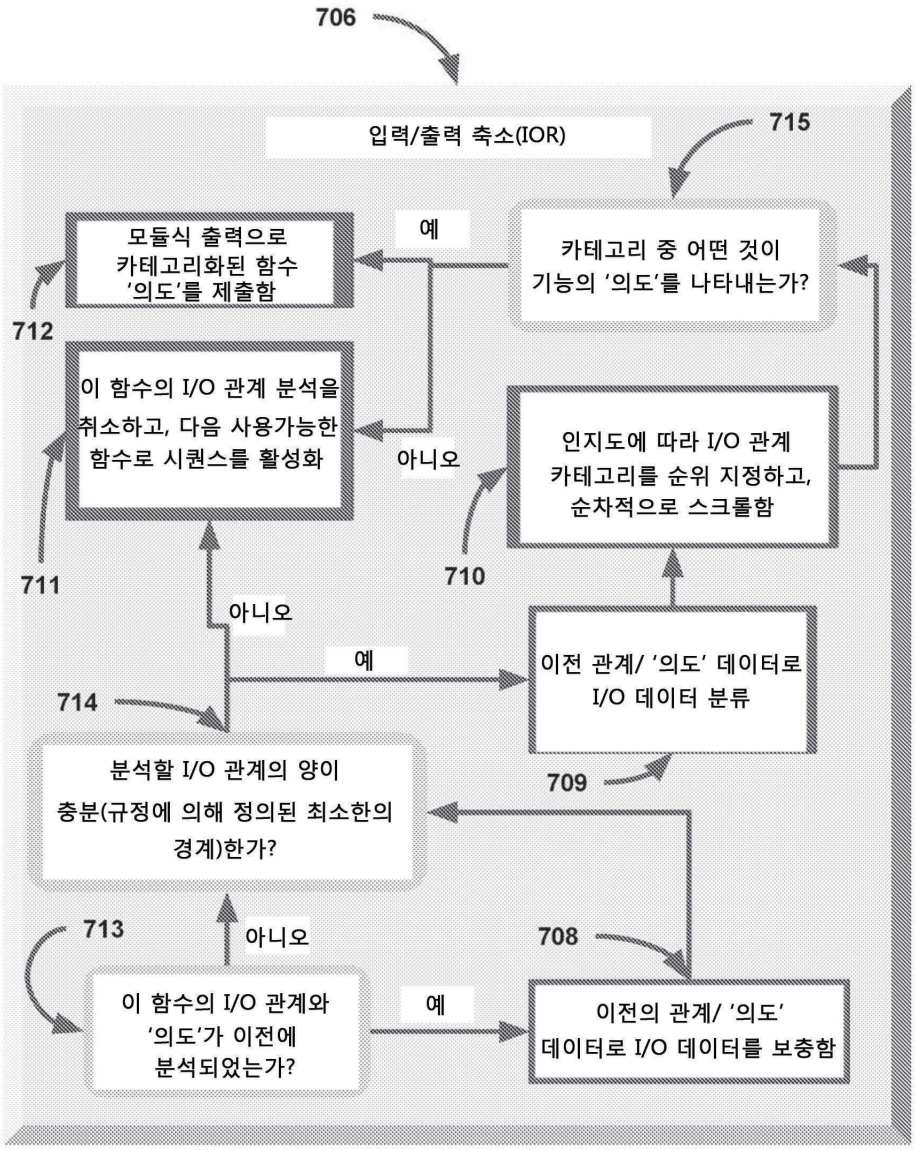
도면109


691A
사각형을 채움
691B
가중치: 25
692A
원을 채움
692B
가중치: 15
693A
다수를 쌓음
693B
가중치: 50
694A
위쪽의 입체화
694B
가중치: 10
695A
측면의 입체화
695B
가중치: 30
696A
전면의 입체화
696B
가중치: 20
보안 응답 X
690
인식 추론(PD)
490
691C
692C
모양을 채움
697
693C
다수를 쌓음
698
694C
695C
696C
입체화
699
15 + 25 = 40
총 가중치 : 50
인식 C의 관점
700
50 = 50
총 가중치 : 50
인식 A의 관점
701
20 + 30 + 10
=60
총 가중치 : 60
인식 B의 관점
702
484
입력 시스템
메타데이터
디버깅 추적
알고리즘 추적
485
486

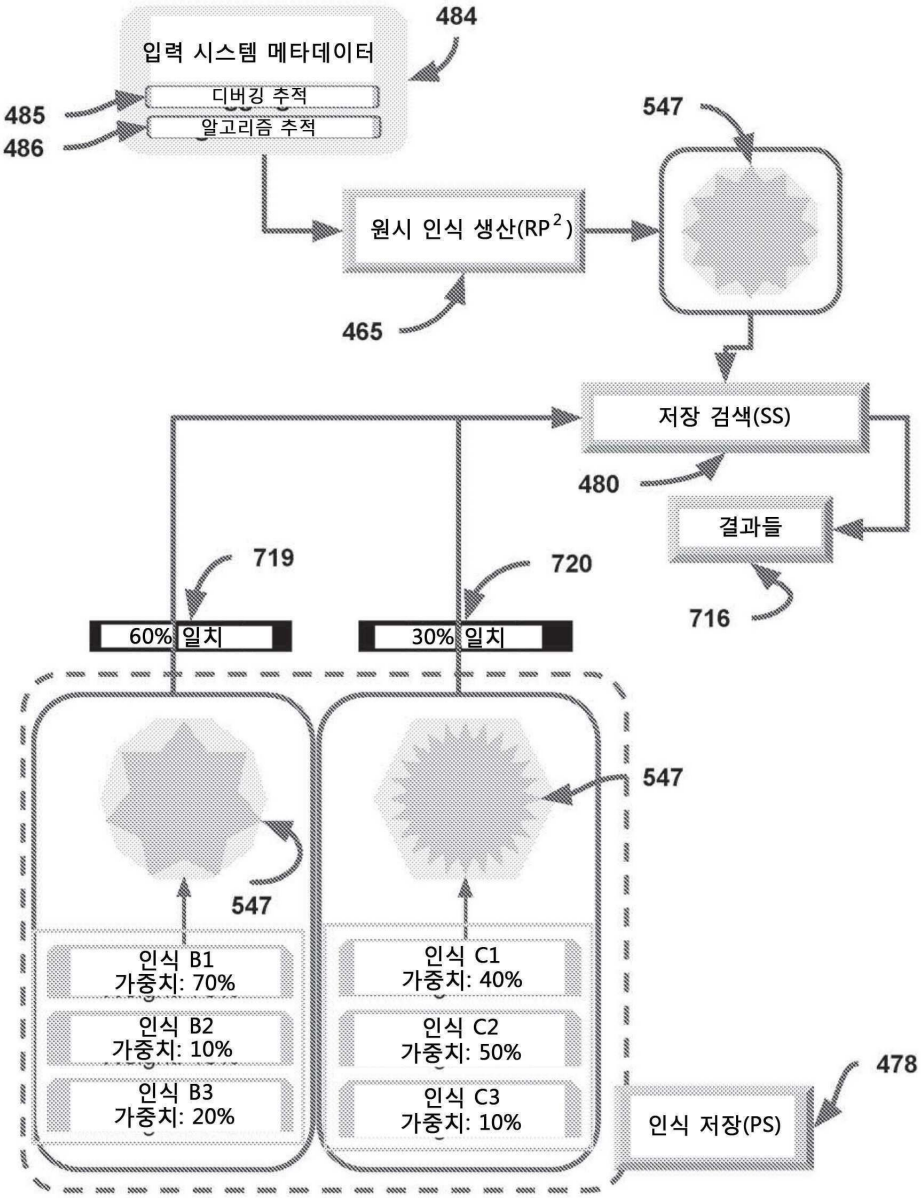
도면110


인식 C의 관점
543
시각
후각
촉각
청각
703
메트릭 변화에 대한 정당화 (JMC)
704
690
보안 응답 X
정당화/ 추론 산출(JRC)
유효하지 않은 정당화 제출
유효한 정당화 제출
484
485
입력 시스템 메타데이터
디버깅 추적
알고리즘 추적
486
488
메타데이터 범주화 모듈(MCM)
아니오
예
SPMA의 보안 응답과 일치/지지하는 의도가 있는가?
705
의도 DB
672
변수
함수
방법
계층
보안 주체
위험 증거
응답 정당화
674
706
입력/출력 축소(IOR)
함수의 I/O관계와 '의도'는 이전에 분석된 적이 있는가?

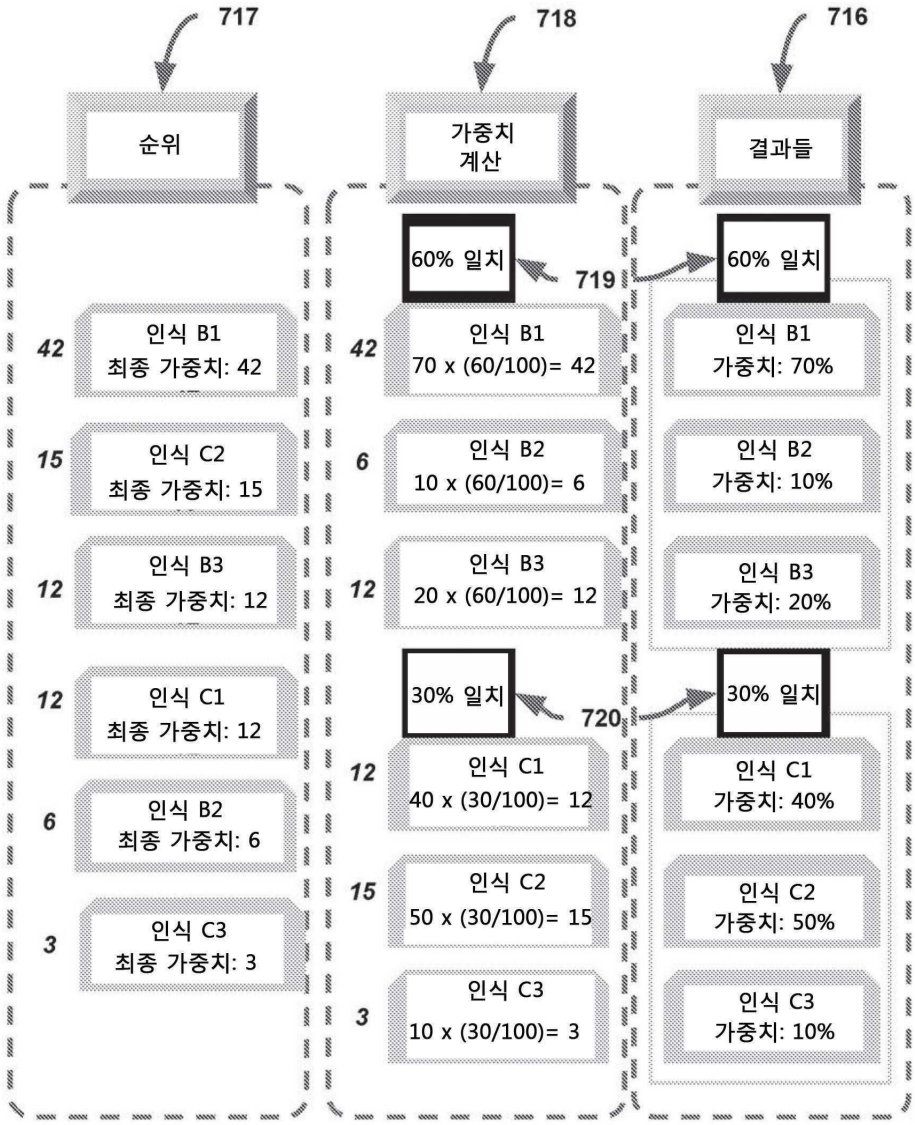
도면111


706
입력/출력 축소(IOR)
715
카테고리 중 어떤 것이 기능의 '의도'를 나타내는가?
예
모듈식 출력으로 카테고리화된 함수 '의도'를 제출함
712
아니오
이 함수의 I/O 관계 분석을 취소하고, 다음 사용가능한 함수로 시퀀스를 활성화
711
인지도에 따라 I/O 관계 카테고리를 순위 지정하고, 순차적으로 스크롤함
710
아니오
예
이전 관계/ '의도' 데이터로 I/O 데이터 분류
714
분석할 I/O 관계의 양이 충분(규정에 의해 정의된 최소한의 경계)한가?
709
713
아니오
708
이 함수의 I/O 관계와 '의도'가 이전에 분석되었는가?
예
이전의 관계/ '의도' 데이터로 I/O 데이터를 보충함

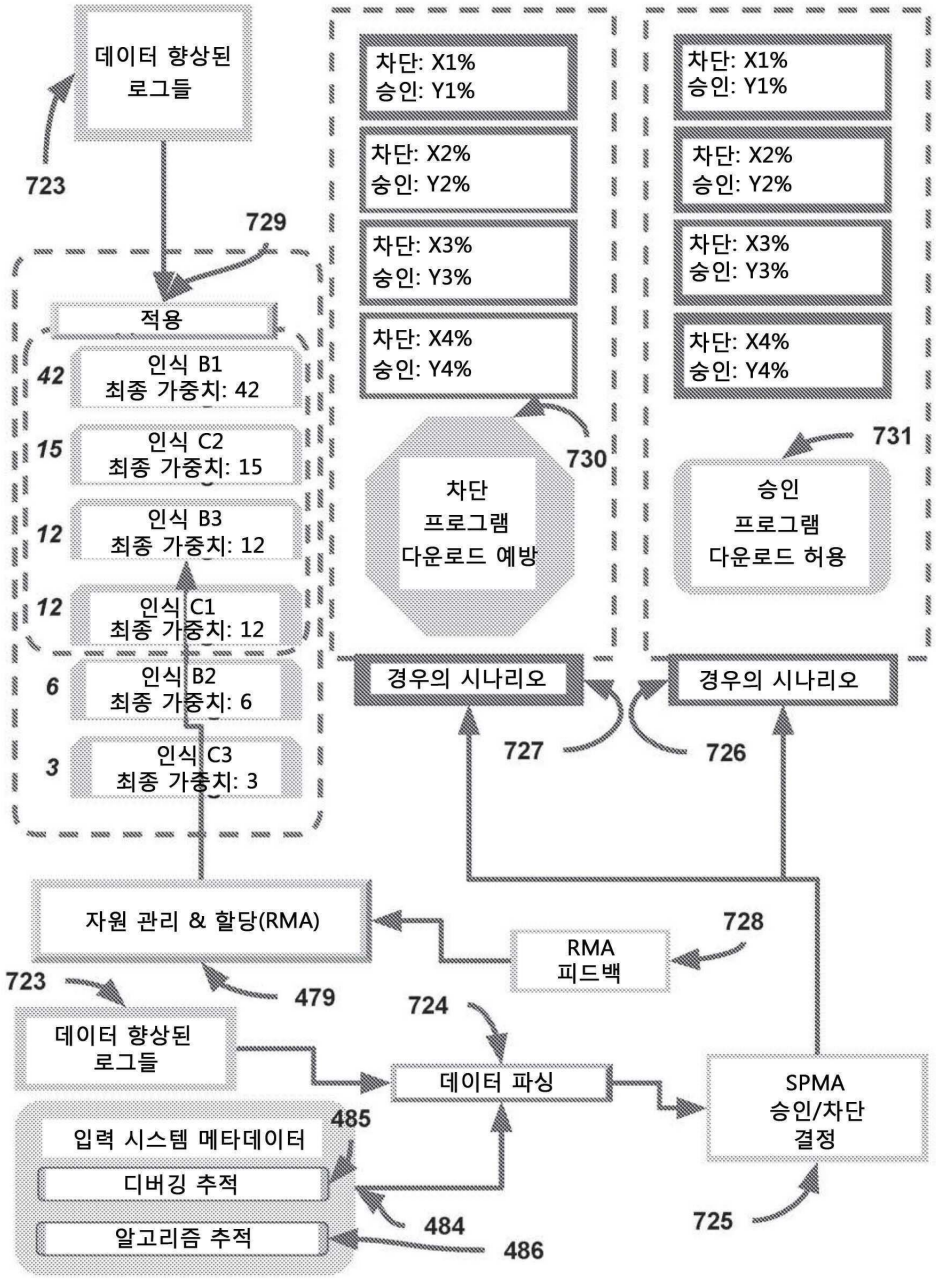
도면112


484
입력 시스템 메타데이터
디버깅 추적
485
알고리즘 추적
486
547
원시 인식 생산(RP²)
465
저장 검색(SS)
480
결과들
716
719
720
60% 일치
30% 일치
547
547
인식 B1
가중치: 70%
인식 B2
가중치: 10%
인식 B3
가중치: 20%
인식 C1
가중치: 40%
인식 C2
가중치: 50%
인식 C3
가중치: 10%
478
인식 저장(PS)

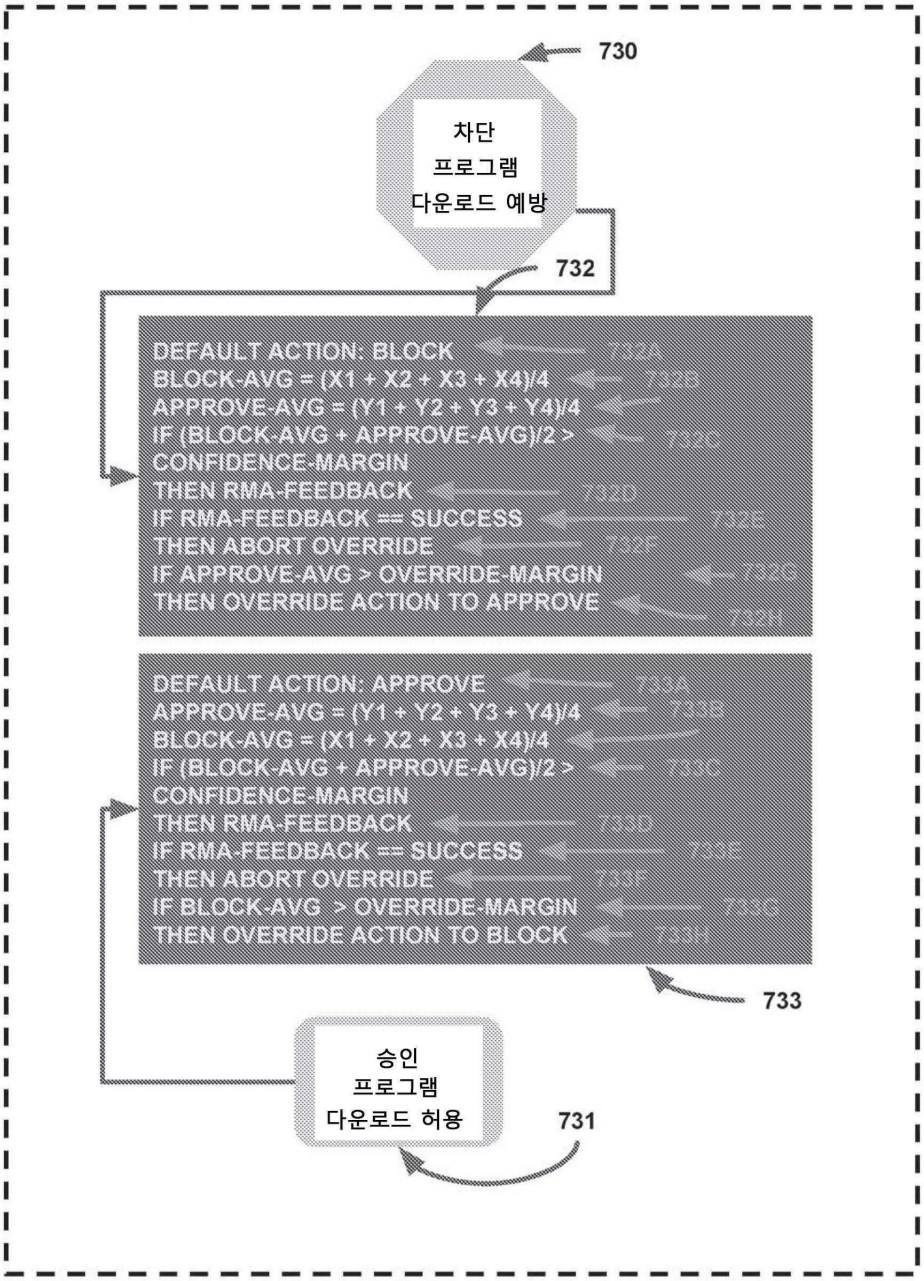
도면113


717
718
716
순위
가중치 계산
결과들
60% 일치
719
60% 일치
42
인식 B1
최종 가중치: 42
42
인식 B1
70 x (60/100)= 42
인식 B1
가중치: 70%
15
인식 C2
최종 가중치: 15
6
인식 B2
10 x (60/100)= 6
인식 B2
가중치: 10%
12
인식 B3
최종 가중치: 12
12
인식 B3
20 x (60/100)= 12
인식 B3
가중치: 20%
30% 일치
720
30% 일치
12
인식 C1
최종 가중치: 12
12
인식 C1
40 x (30/100)= 12
인식 C1
가중치: 40%
6
인식 B2
최종 가중치: 6
15
인식 C2
50 x (30/100)= 15
인식 C2
가중치: 50%
3
인식 C3
최종 가중치: 3
3
인식 C3
10 x (30/100)= 3
인식 C3
가중치: 10%

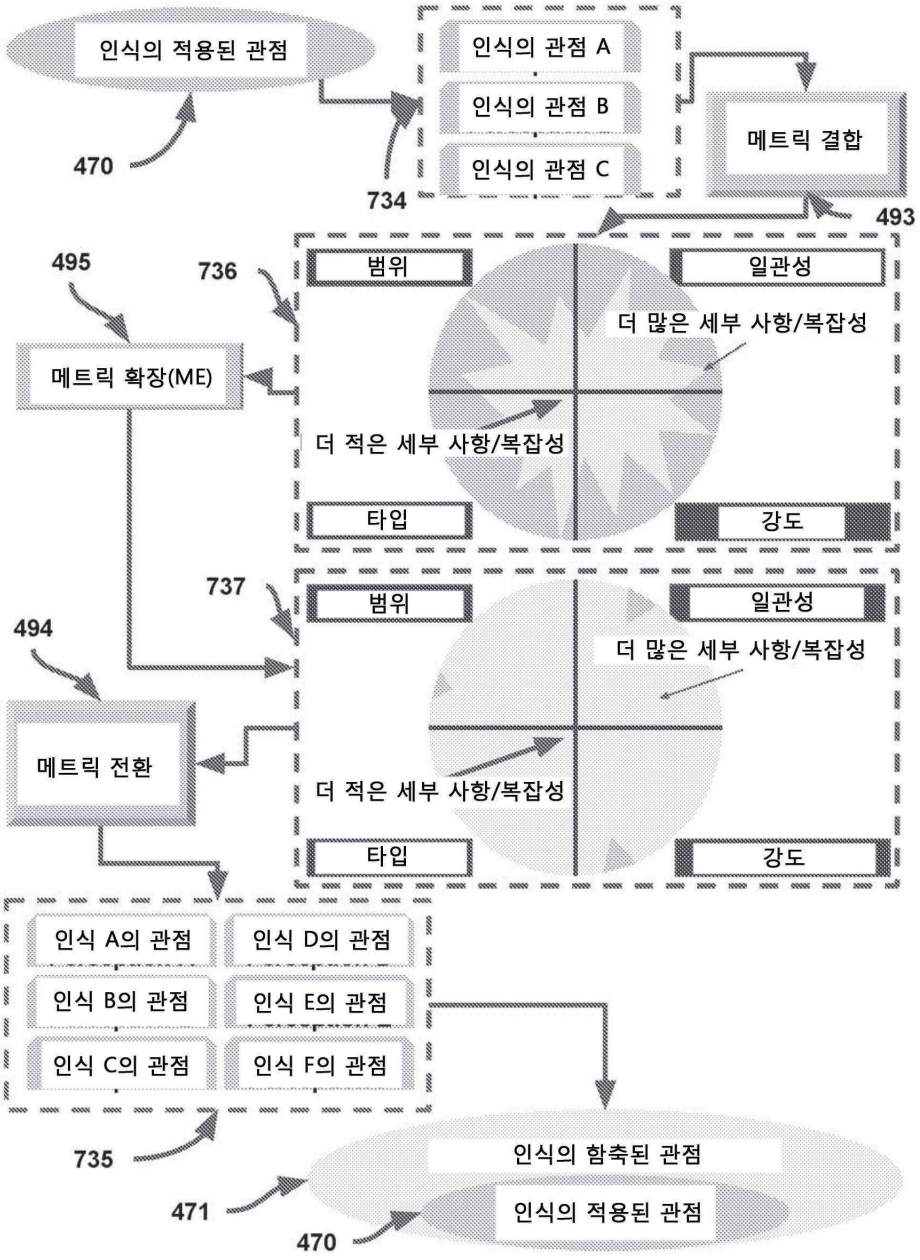
도면114


데이터 향상된 로그들
723
729
적용
42
인식 B1
최종 가중치: 42
15
인식 C2
최종 가중치: 15
12
인식 B3
최종 가중치: 12
12
인식 C1
최종 가중치: 12
6
인식 B2
최종 가중치: 6
3
인식 C3
최종 가중치: 3
차단: X1%
승인: Y1%
차단: X2%
승인: Y2%
차단: X3%
승인: Y3%
차단: X4%
승인: Y4%
730
차단
프로그램
다운로드 예방
차단: X1%
승인: Y1%
차단: X2%
승인: Y2%
차단: X3%
승인: Y3%
차단: X4%
승인: Y4%
731
승인
프로그램
다운로드 허용
경우의 시나리오
727
726
경우의 시나리오
자원 관리 & 할당(RMA)
728
RMA
피드백
723
479
724
데이터 향상된 로그들
데이터 파싱
SPMA
승인/차단
결정
485
입력 시스템 메타데이터
디버깅 추적
알고리즘 추적
484
486
725

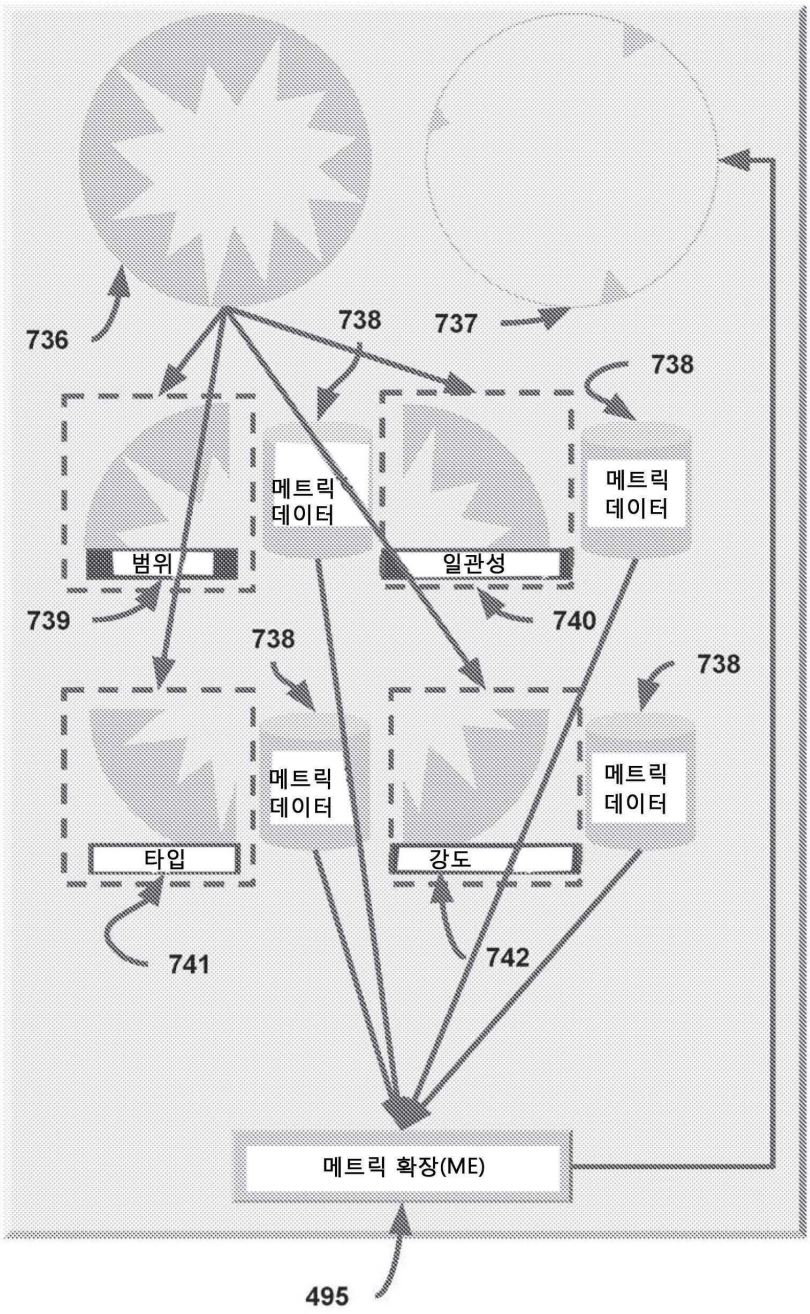
도면115


730
차단
프로그램
다운로드 예방
732
DEFAULT ACTION: BLOCK
BLOCK-AVG = (X1 + X2 + X3 + X4)/4
APPROVE-AVG = (Y1 + Y2 + Y3 + Y4)/4
IF (BLOCK-AVG + APPROVE-AVG)/2 >
CONFIDENCE-MARGIN
THEN RMA-FEEDBACK
IF RMA-FEEDBACK == SUCCESS
THEN ABORT OVERRIDE
IF APPROVE-AVG > OVERRIDE-MARGIN
THEN OVERRIDE ACTION TO APPROVE
DEFAULT ACTION: APPROVE
APPROVE-AVG = (Y1 + Y2 + Y3 + Y4)/4
BLOCK-AVG = (X1 + X2 + X3 + X4)/4
IF (BLOCK-AVG + APPROVE-AVG)/2 >
CONFIDENCE-MARGIN
THEN RMA-FEEDBACK
IF RMA-FEEDBACK == SUCCESS
THEN ABORT OVERRIDE
IF BLOCK-AVG > OVERRIDE-MARGIN
THEN OVERRIDE ACTION TO BLOCK
733
승인
프로그램
다운로드 허용
731

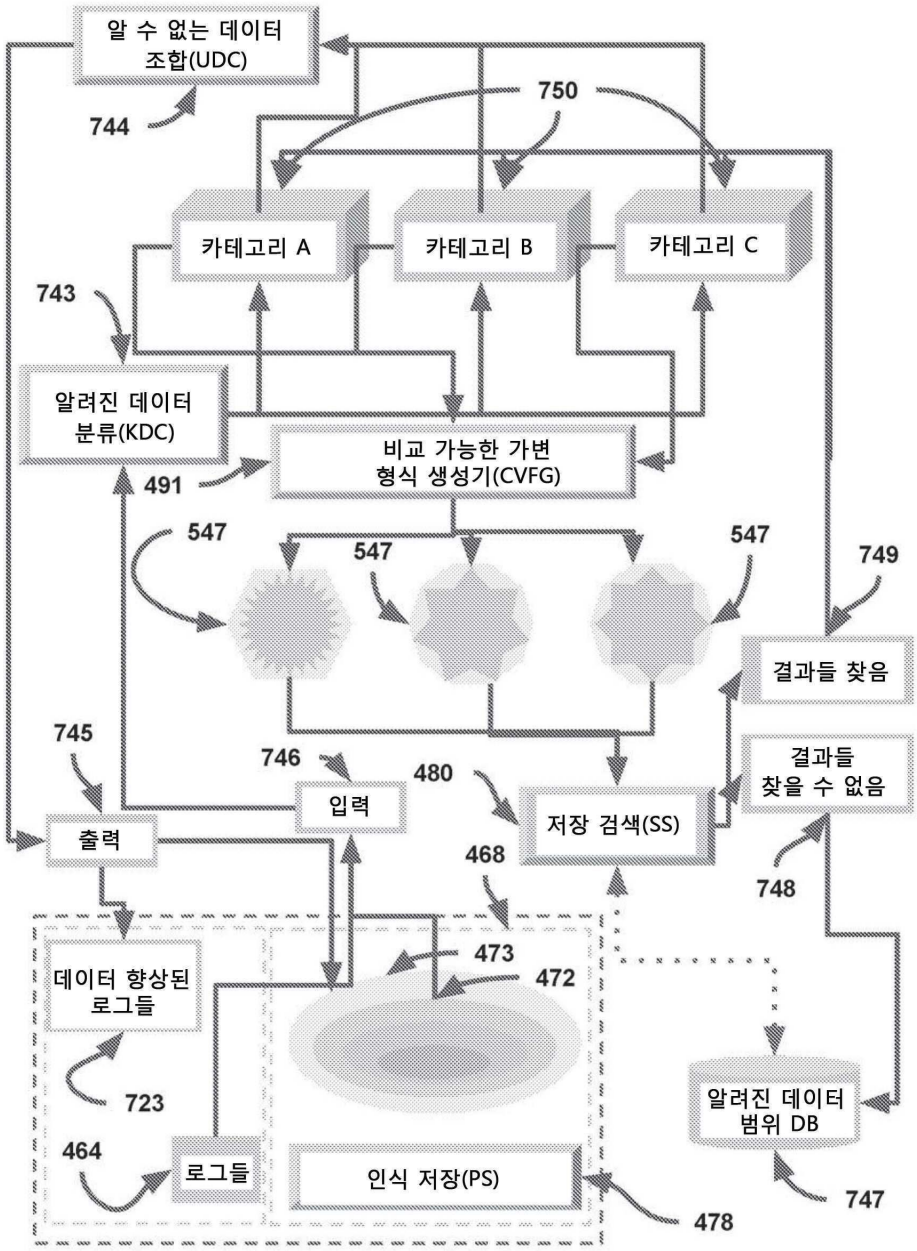
도면116


인식의 적용된 관점
470
734
인식의 관점 A
인식의 관점 B
인식의 관점 C
메트릭 결합
493
495
736
범위
일관성
더 많은 세부 사항/복잡성
메트릭 확장(ME)
더 적은 세부 사항/복잡성
타입
강도
737
494
메트릭 전환
인식 A의 관점
인식 D의 관점
인식 B의 관점
인식 E의 관점
인식 C의 관점
인식 F의 관점
735
471
인식의 함축된 관점
인식의 적용된 관점

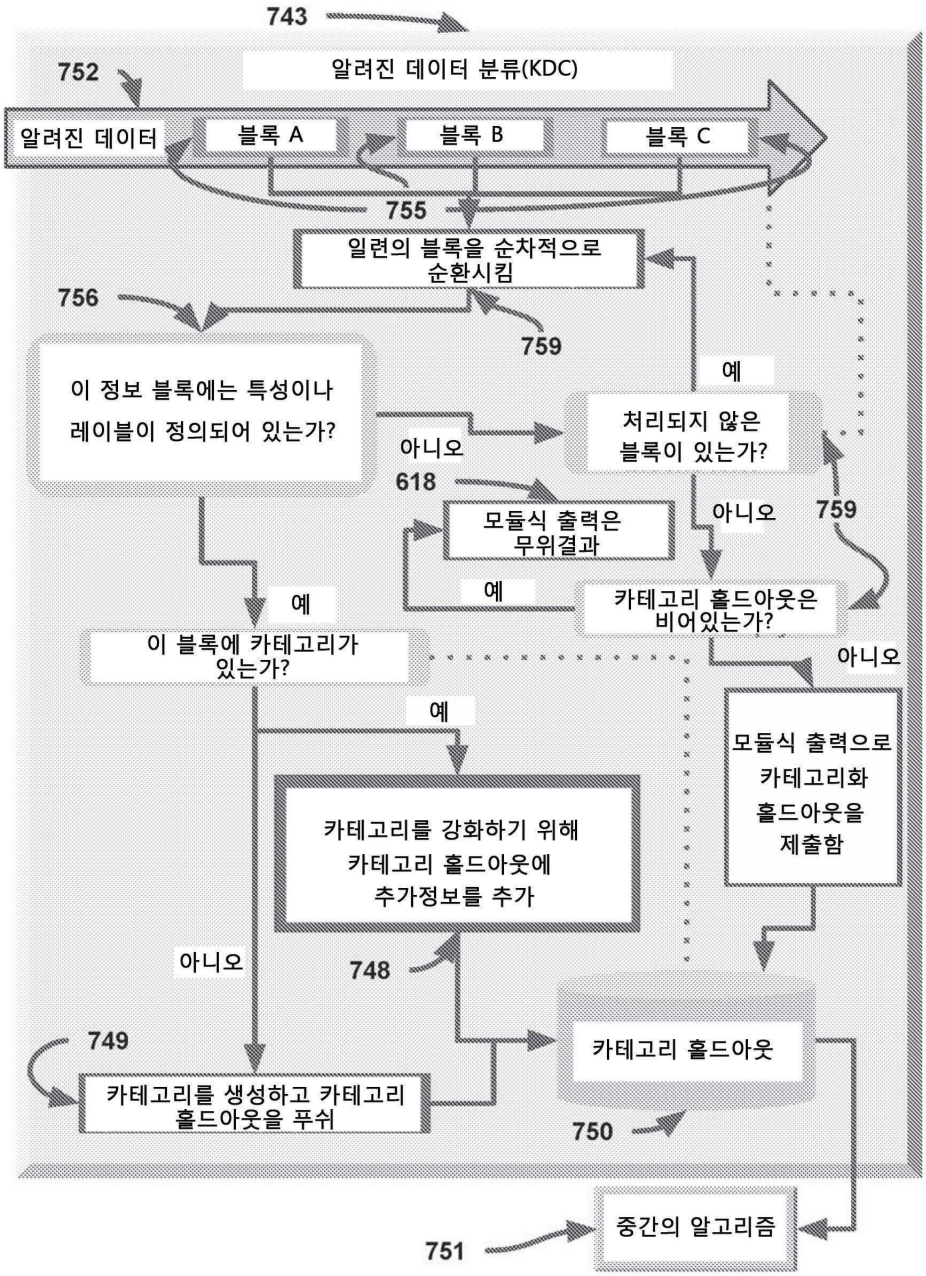
도면117


736
738
737
738
메트릭 데이터
메트릭 데이터
범위
일관성
739
740
738
738
메트릭 데이터
메트릭 데이터
타입
강도
741
742
메트릭 확장(ME)
495

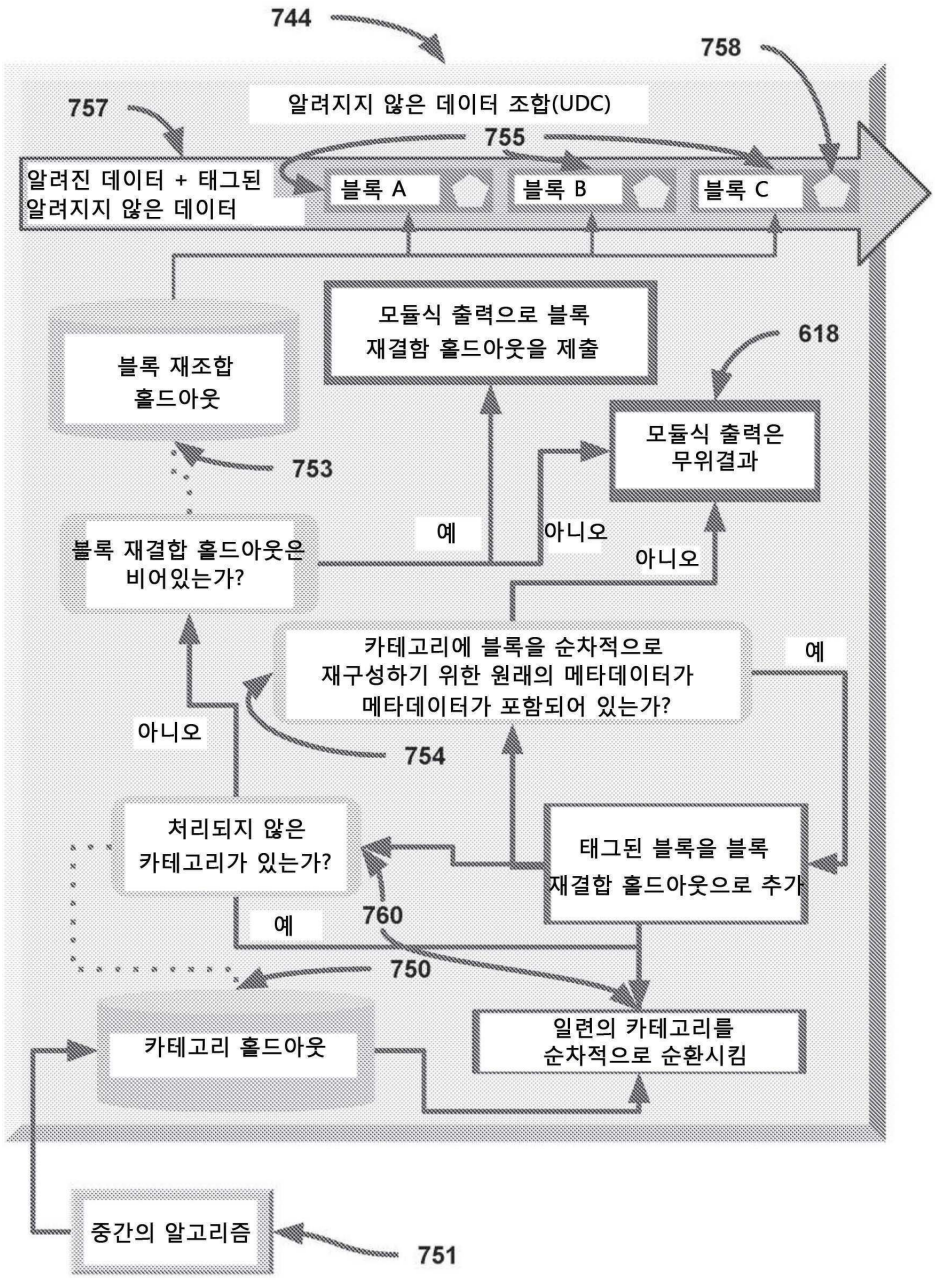
도면118


알 수 없는 데이터 조합(UDC)
744
750
카테고리 A
카테고리 B
카테고리 C
743
알려진 데이터 분류(KDC)
비교 가능한 가변 형식 생성기(CVFG)
491
547
547
547
749
결과들 찾음
745
746
480
입력
저장 검색(SS)
결과들 찾을 수 없음
출력
468
748
473
472
데이터 향상된 로그들
723
464
로그들
인식 저장(PS)
알려진 데이터 범위 DB
747
478

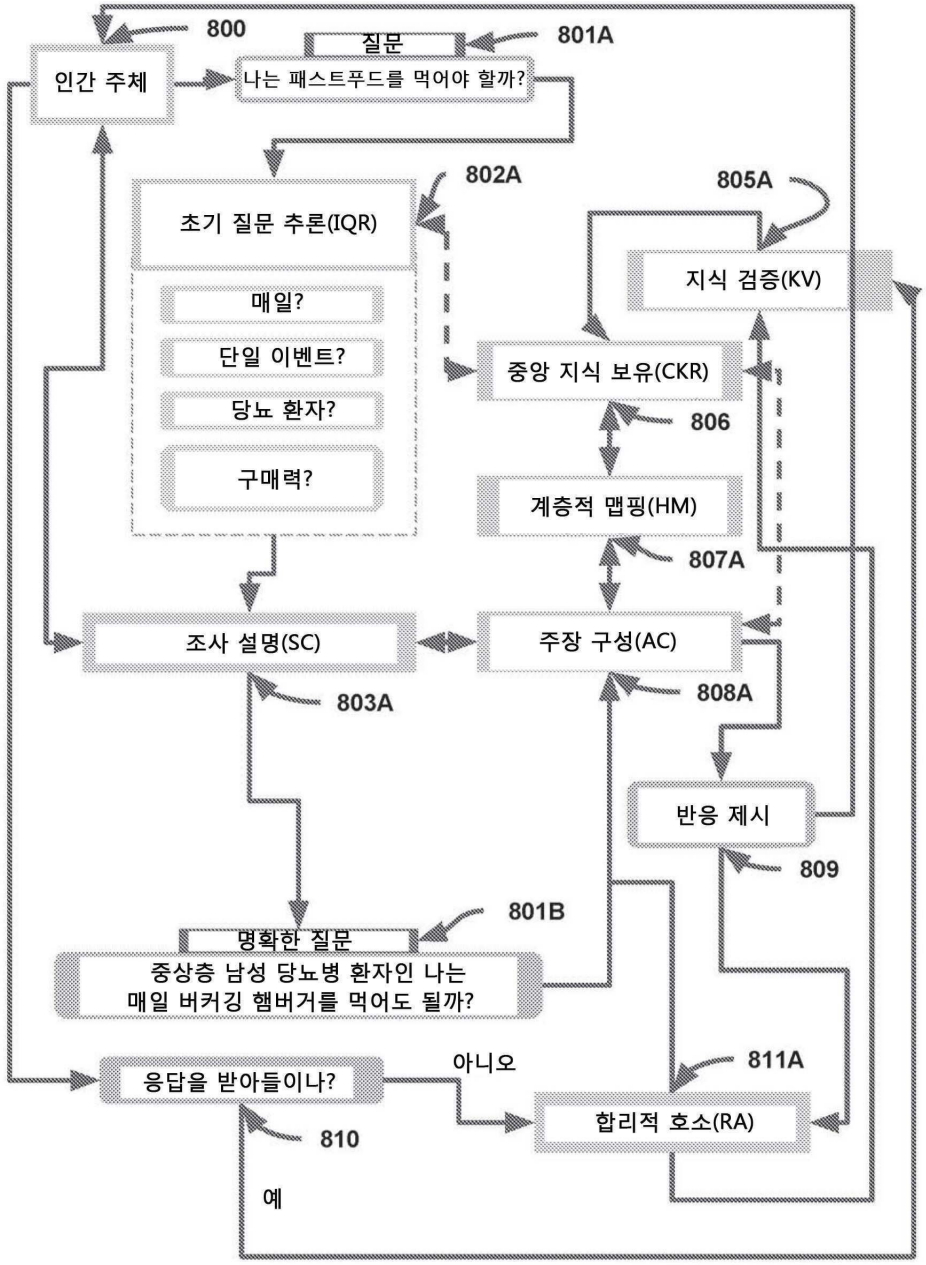
도면119


743
752
알려진 데이터 분류(KDC)
알려진 데이터
블록 A
블록 B
블록 C
755
일련의 블록을 순차적으로 순환시킴
756
759
이 정보 블록에는 특성이나 레이블이 정의되어 있는가?
예
처리되지 않은 블록이 있는가?
아니오
618
모듈식 출력은 무위결과
아니오
759
예
카테고리 홀드아웃은 비어있는가?
예
이 블록에 카테고리가 있는가?
아니오
예
모듈식 출력으로 카테고리화 홀드아웃을 제출함
카테고리를 강화하기 위해 카테고리 홀드아웃에 추가정보를 추가
아니오
748
749
카테고리 홀드아웃
카테고리를 생성하고 카테고리 홀드아웃을 푸쉬
750
중간의 알고리즘
751

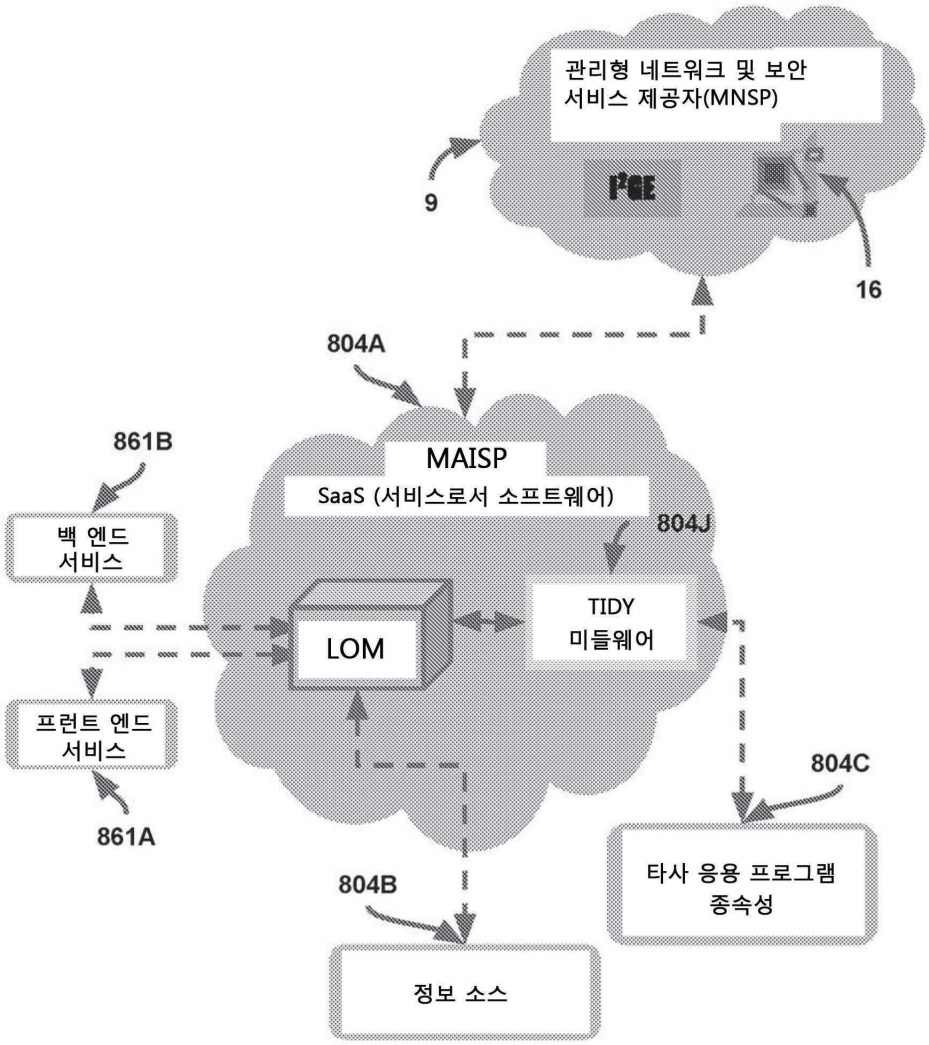
도면120


744
758
757
알려지지 않은 데이터 조합(UDC)
755
알려진 데이터 + 태그된
알려지지 않은 데이터
블록 A
블록 B
블록 C
모듈식 출력으로 블록
재결함 홀드아웃을 제출
618
블록 재조합
홀드아웃
753
모듈식 출력은
무위결과
블록 재결합 홀드아웃은
비어있는가?
예
아니오
아니오
카테고리에 블록을 순차적으로
재구성하기 위한 원래의 메타데이터가
메타데이터가 포함되어 있는가?
예
아니오
754
처리되지 않은
카테고리가 있는가?
태그된 블록을 블록
재결합 홀드아웃으로 추가
760
예
750
카테고리 홀드아웃
일련의 카테고리를
순차적으로 순환시킴
중간의 알고리즘
751

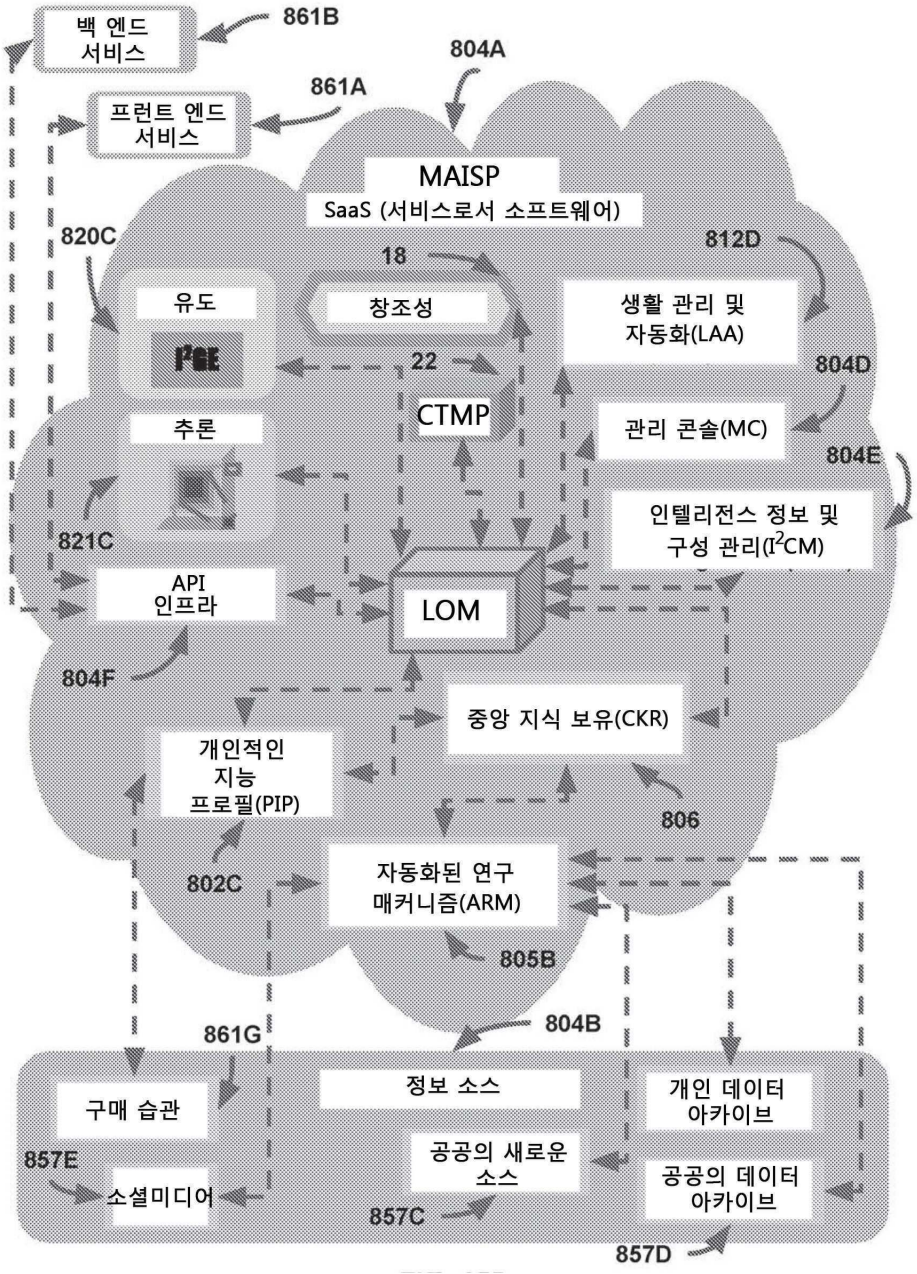
도면121


800
질문
801A
인간 주체
나는 패스트푸드를 먹어야 할까?
802A
805A
초기 질문 추론(IQR)
지식 검증(KV)
매일?
단일 이벤트?
당뇨 환자?
구매력?
중앙 지식 보유(CKR)
806
계층적 맵핑(HM)
807A
조사 설명(SC)
주장 구성(AC)
803A
808A
반응 제시
809
801B
명확한 질문
중상층 남성 당뇨병 환자인 나는
매일 버커딩 햄버거를 먹어도 될까?
811A
응답을 받아들이나?
아니오
합리적 호소(RA)
810
예

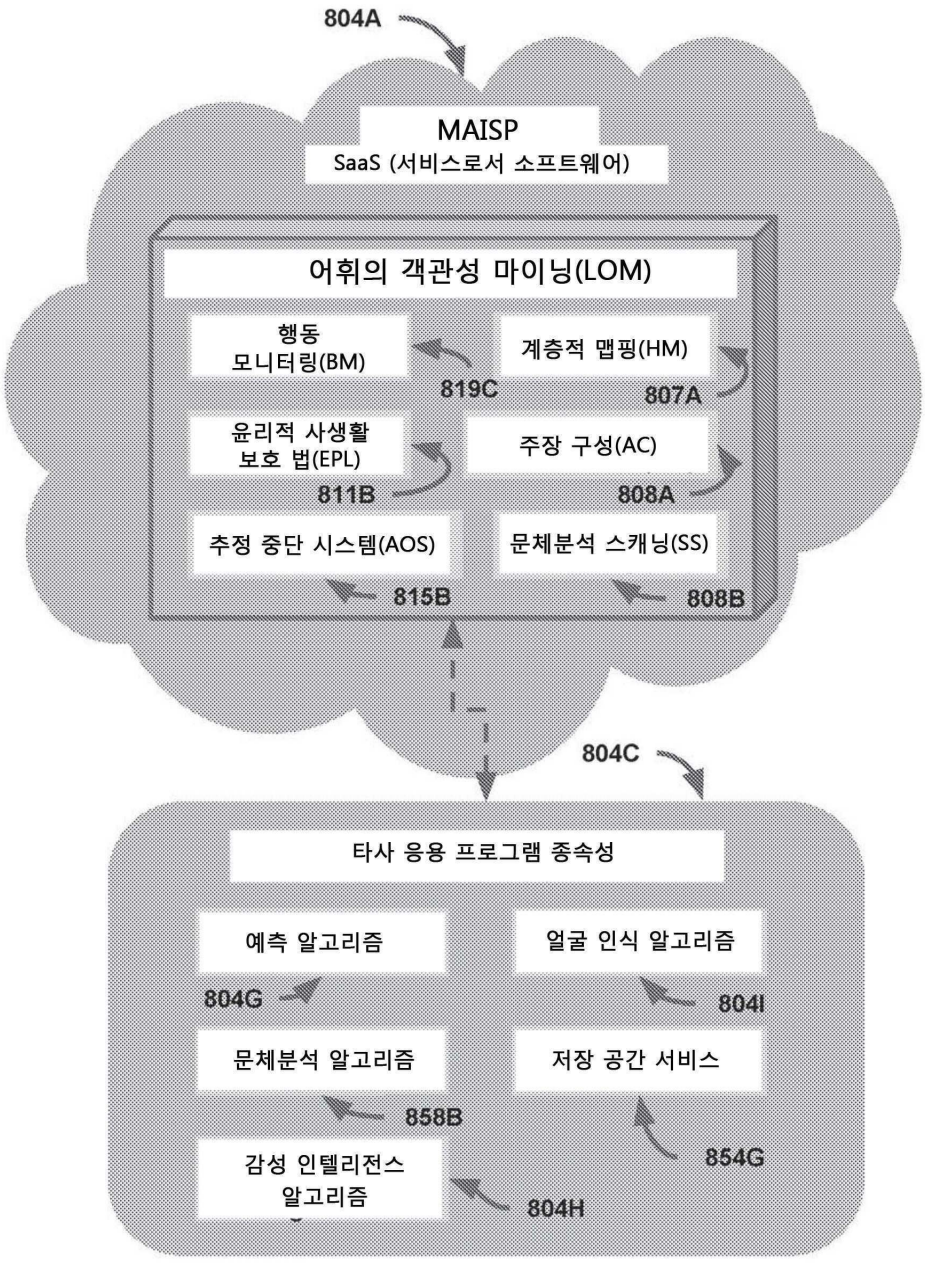
도면122


관리형 네트워크 및 보안
서비스 제공자(MNSP)
9
16
804A
861B
MAISP
SaaS (서비스로서 소프트웨어)
804J
백 엔드
서비스
LOM
TIDY
미들웨어
프런트 엔드
서비스
861A
804C
타사 응용 프로그램
종속성
804B
정보 소스

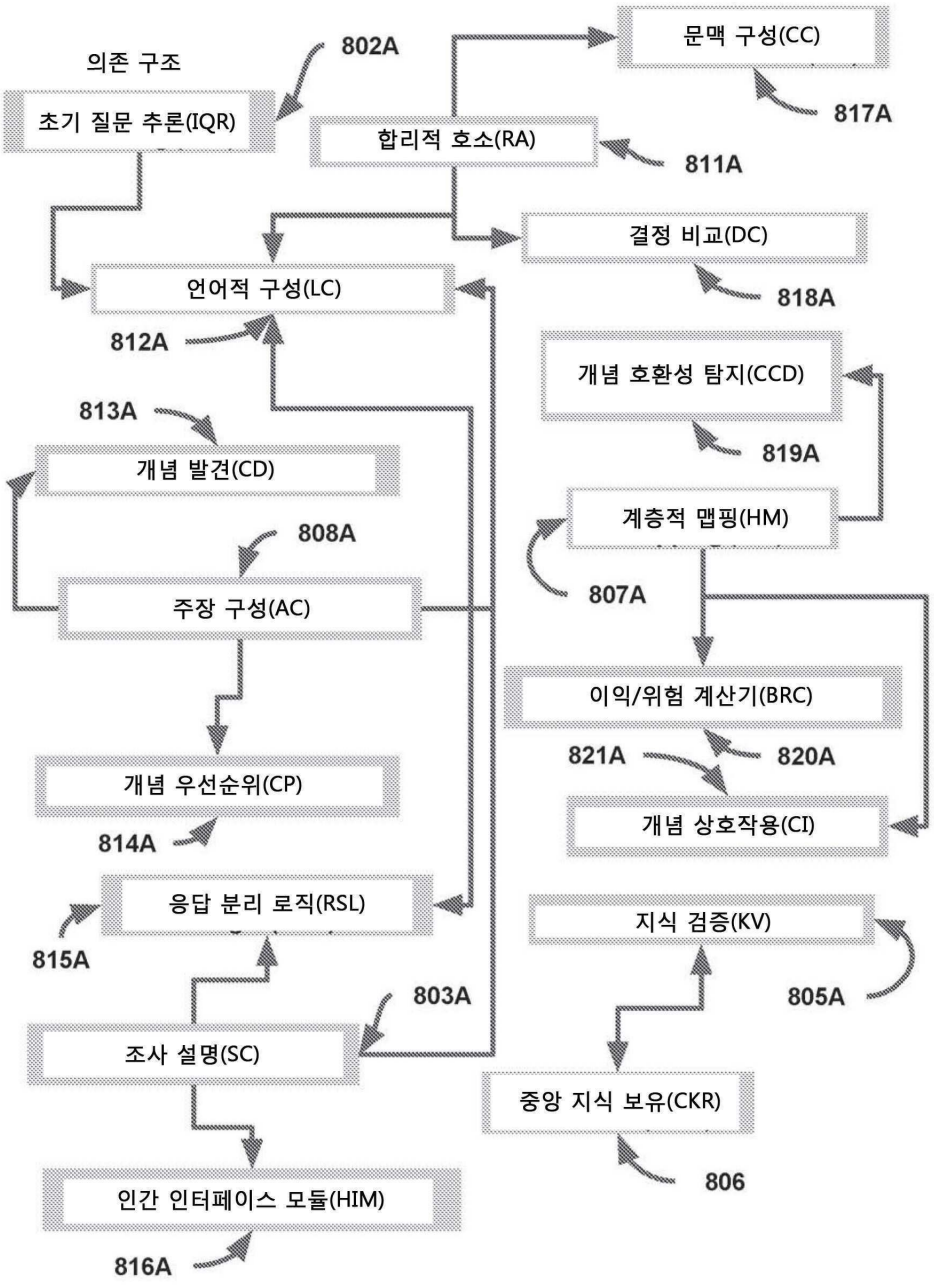
도면123


백 엔드 서비스
861B
804A
861A
프런트 엔드 서비스
MAISP
SaaS (서비스로서 소프트웨어)
820C
18
812D
유도
창조성
생활 관리 및 자동화(LAA)
22
804D
CTMP
추론
관리 콘솔(MC)
804E
인텔리전스 정보 및 구성 관리(I2CM)
821C
API 인프라
LOM
804F
중앙 지식 보유(CKR)
개인적인 지능 프로필(PIP)
806
802C
자동화된 연구 매커니즘(ARM)
805B
804B
861G
정보 소스
구매 습관
개인 데이터 아카이브
857E
공공의 새로운 소스
공공의 데이터 아카이브
소셜미디어
857C
857D

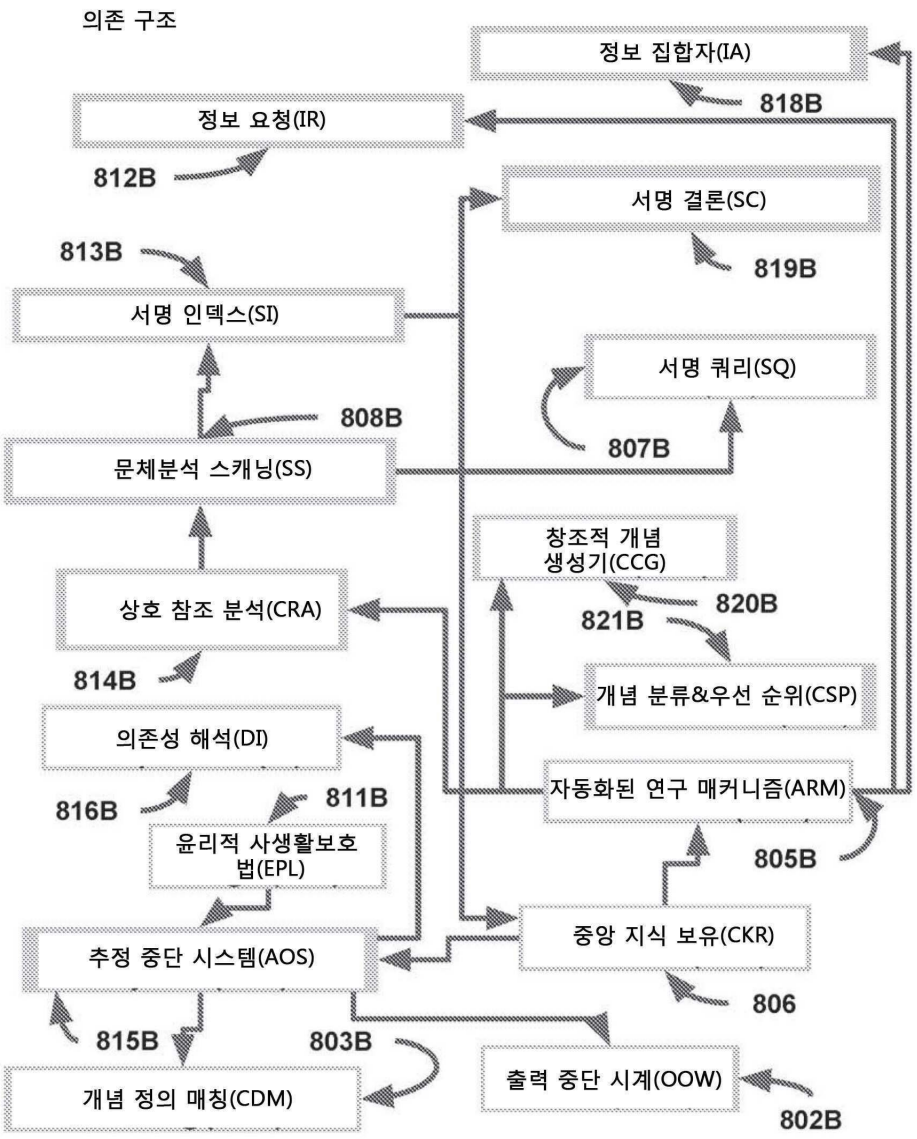
도면124


804A
MAISP
SaaS (서비스로서 소프트웨어)
어휘의 객관성 마이닝(LOM)
행동 모니터링(BM)
819C
계층적 맵핑(HM)
807A
윤리적 사생활 보호 법(EPL)
811B
주장 구성(AC)
808A
추정 중단 시스템(AOS)
815B
문체분석 스캐닝(SS)
808B
804C
타사 응용 프로그램 종속성
예측 알고리즘
804G
얼굴 인식 알고리즘
804I
문체분석 알고리즘
858B
저장 공간 서비스
854G
감성 인텔리전스 알고리즘
804H

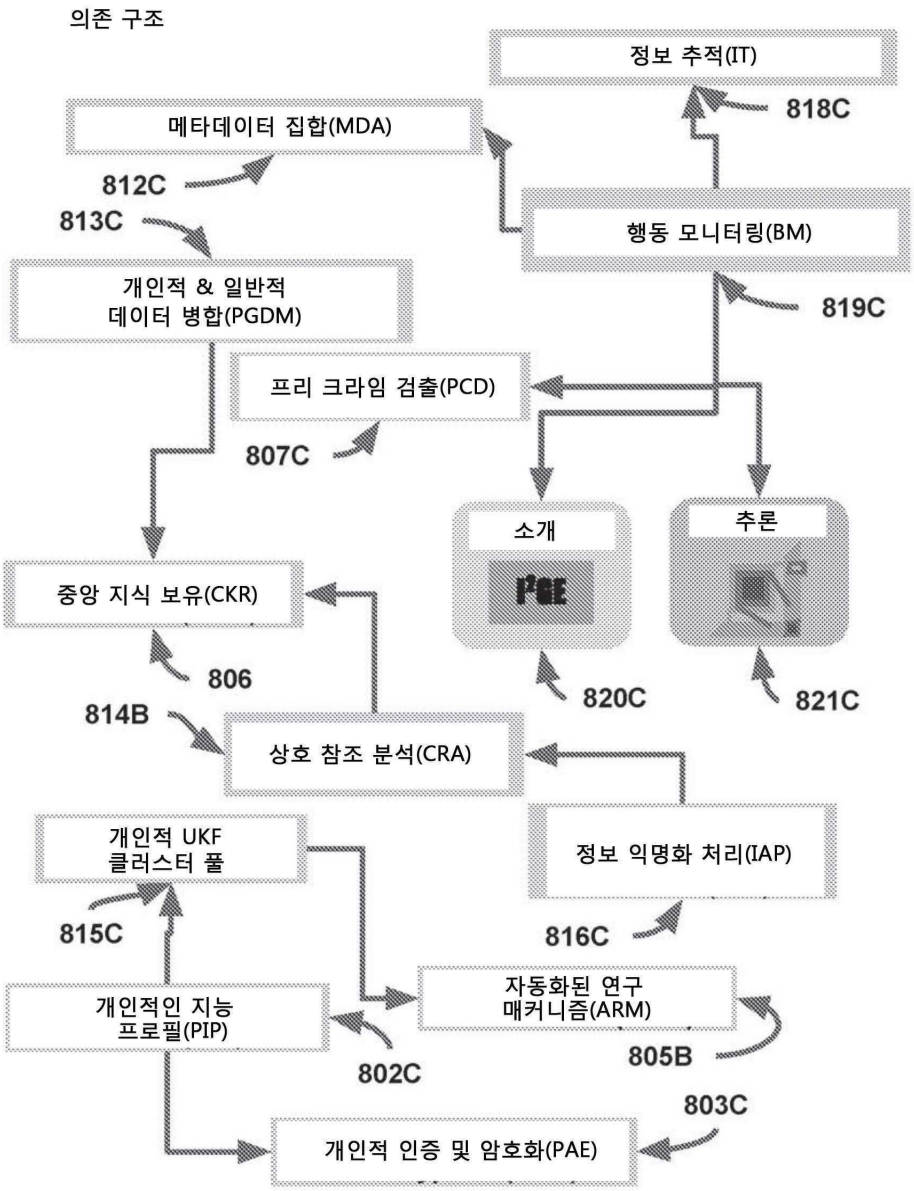
도면125


의존 구조
802A
초기 질문 추론(IQR)
합리적 호소(RA)
811A
문맥 구성(CC)
817A
결정 비교(DC)
818A
언어적 구성(LC)
812A
813A
개념 발견(CD)
808A
주장 구성(AC)
개념 우선순위(CP)
814A
응답 분리 로직(RSL)
815A
조사 설명(SC)
803A
인간 인터페이스 모듈(HIM)
816A
개념 호환성 탐지(CCD)
819A
계층적 맵핑(HM)
807A
이익/위험 계산기(BRC)
821A
820A
개념 상호작용(CI)
지식 검증(KV)
805A
중앙 지식 보유(CKR)
806

도면126

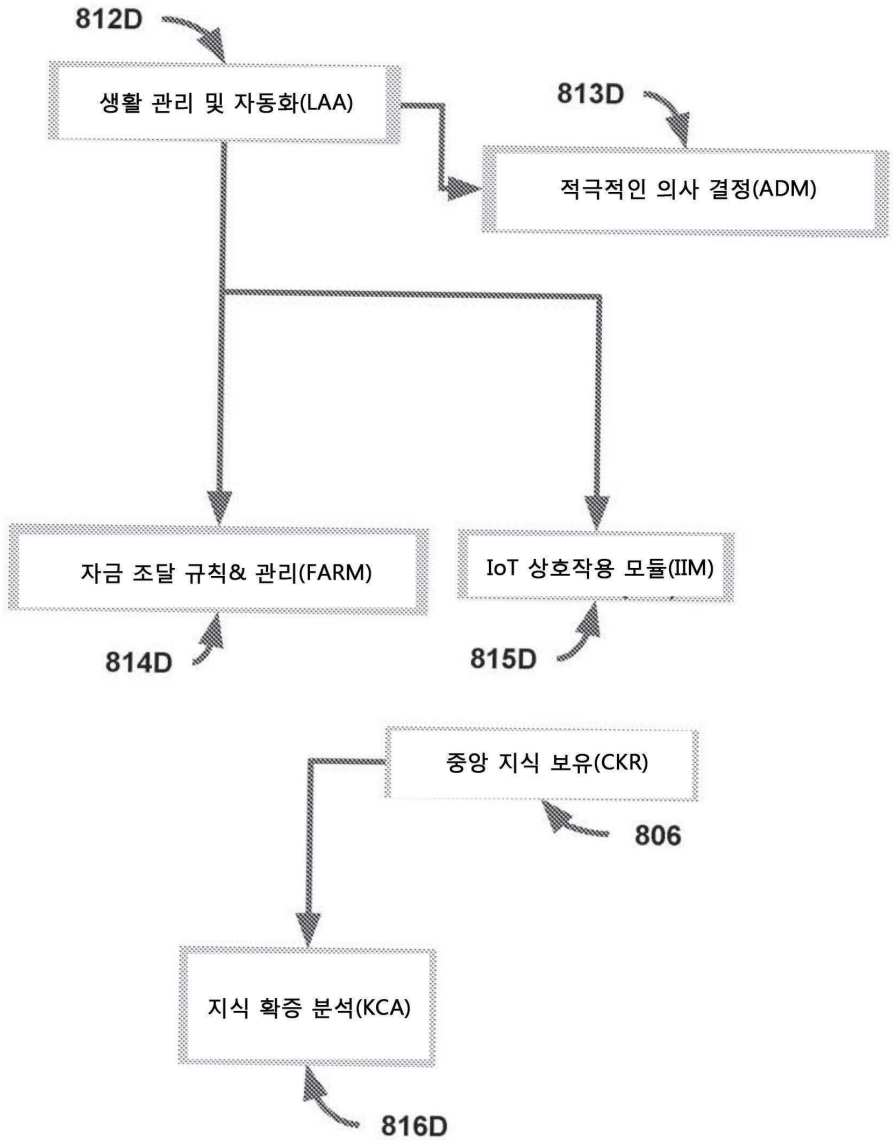

의존 구조
정보 집합자(IA)
818B
정보 요청(IR)
812B
서명 결론(SC)
819B
813B
서명 인덱스(SI)
서명 쿼리(SQ)
808B
807B
문체분석 스캐닝(SS)
창조적 개념
생성기(CCG)
820B
821B
상호 참조 분석(CRA)
814B
개념 분류&우선 순위(CSP)
의존성 해석(DI)
816B
811B
자동화된 연구 매커니즘(ARM)
윤리적 사생활보호
법(EPL)
805B
중앙 지식 보유(CKR)
추정 중단 시스템(AOS)
806
815B
803B
출력 중단 시계(OOW)
개념 정의 매칭(CDM)
802B

도면127

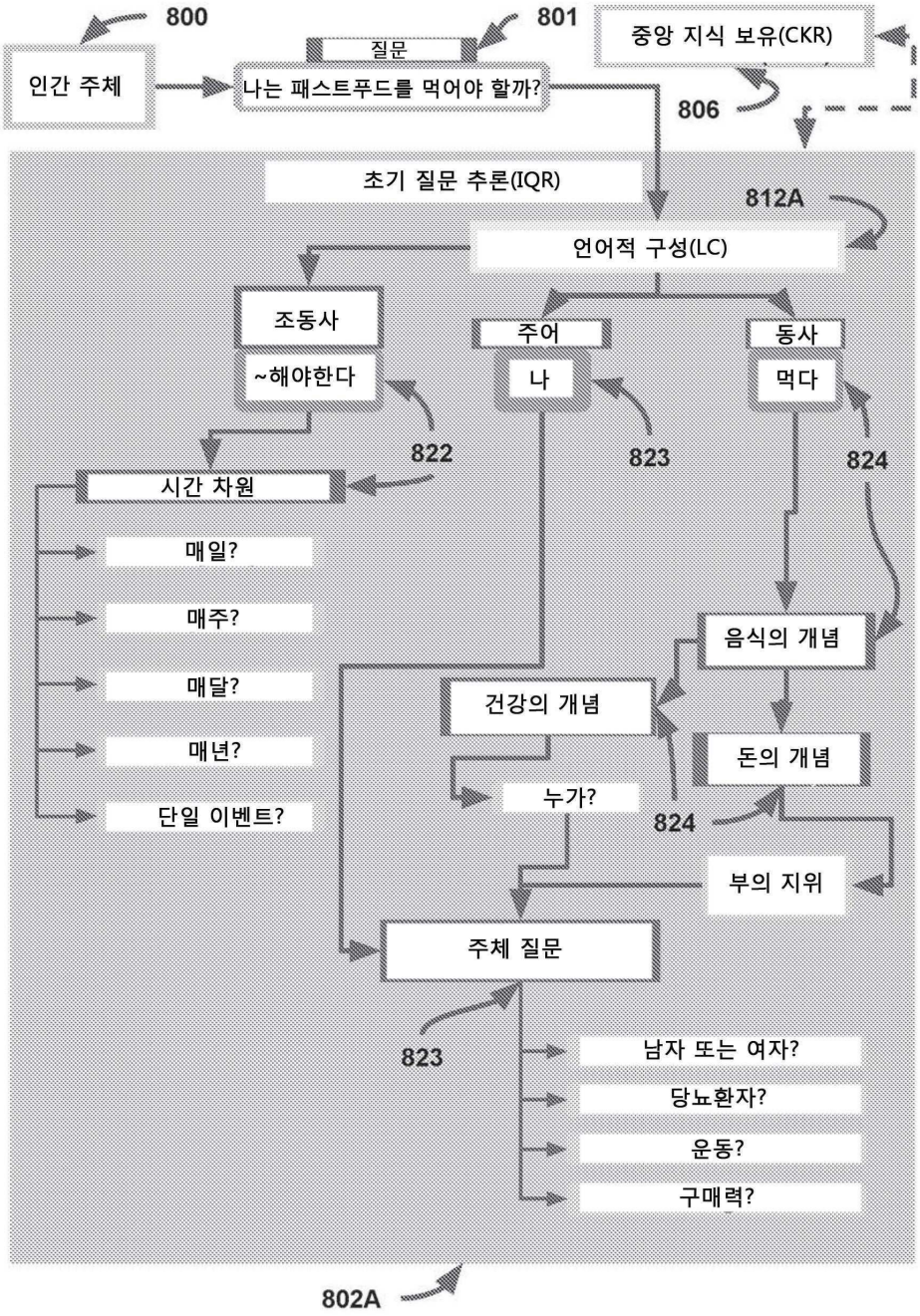

의존 구조
정보 추적(IT)
818C
메타데이터 집합(MDA)
812C
813C
행동 모니터링(BM)
개인적 & 일반적
데이터 병합(PGDM)
819C
프리 크라임 검출(PCD)
807C
소개
추론
중앙 지식 보유(CKR)
806
820C
821C
814B
상호 참조 분석(CRA)
개인적 UKF
클러스터 풀
정보 익명화 처리(IAP)
815C
816C
자동화된 연구
매커니즘(ARM)
개인적인 지능
프로필(PIP)
805B
802C
803C
개인적 인증 및 암호화(PAE)

도면128

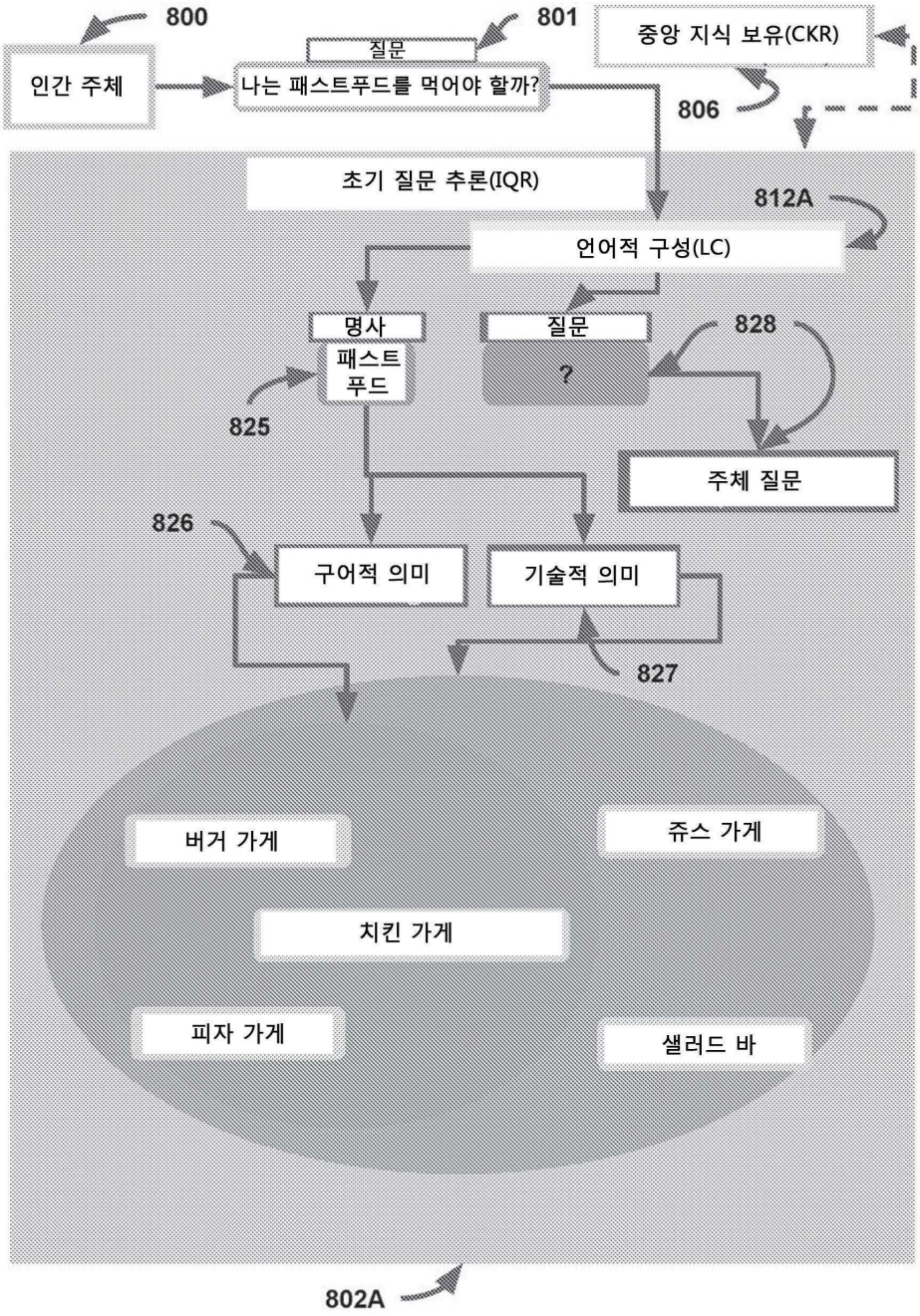
의존 구조


812D
생활 관리 및 자동화(LAA)
813D
적극적인 의사 결정(ADM)
자금 조달 규칙& 관리(FARM)
IoT 상호작용 모듈(IIM)
814D
815D
중앙 지식 보유(CKR)
806
지식 확증 분석(KCA)
816D

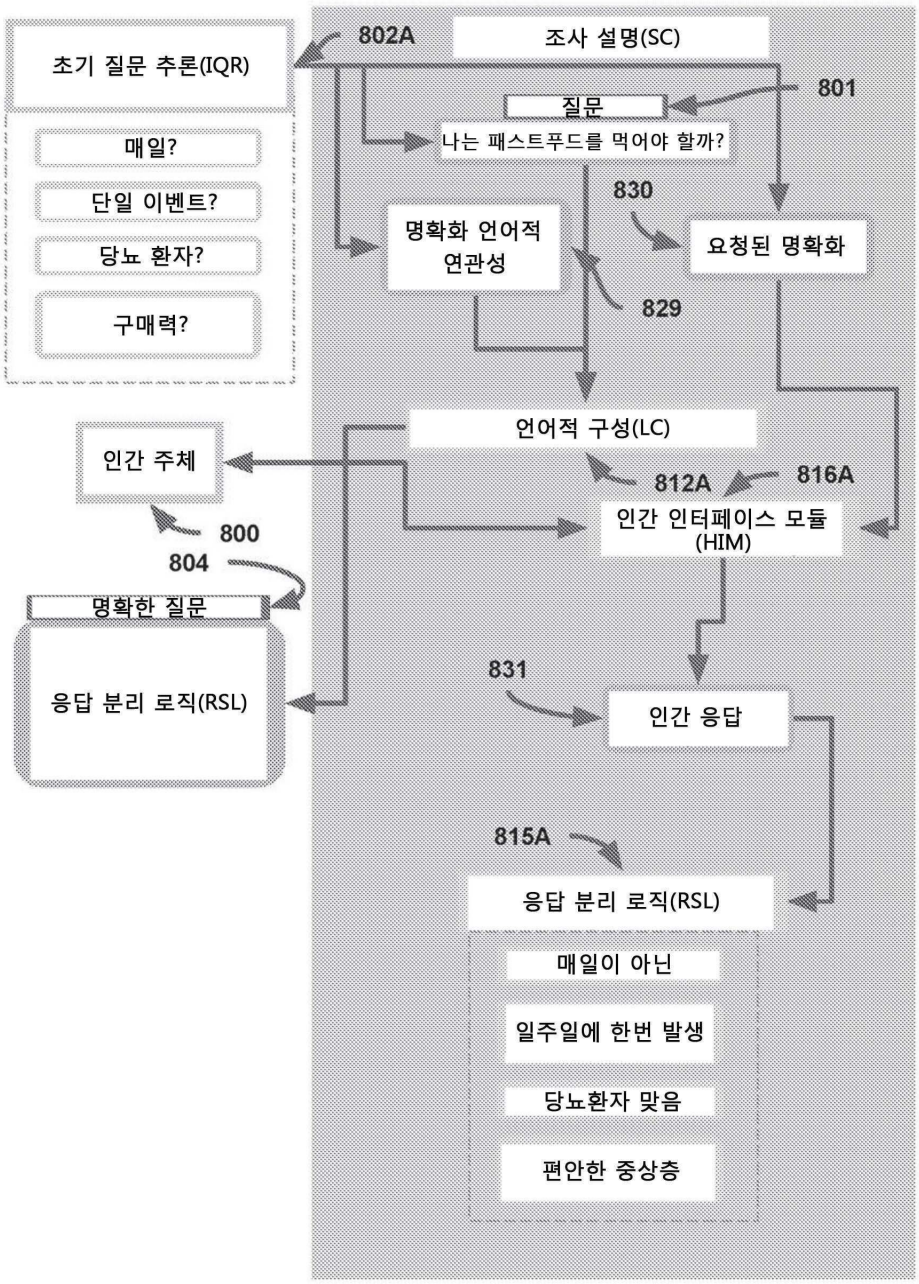
도면129


800
801
질문
중앙 지식 보유(CKR)
인간 주체
나는 패스트푸드를 먹어야 할까?
806
초기 질문 추론(IQR)
812A
언어적 구성(LC)
조동사
~해야한다
주어
나
동사
먹다
822
823
824
시간 차원
매일?
매주?
매달?
매년?
단일 이벤트?
음식의 개념
건강의 개념
돈의 개념
누가?
824
부의 지위
주체 질문
823
남자 또는 여자?
당뇨환자?
운동?
구매력?
802A

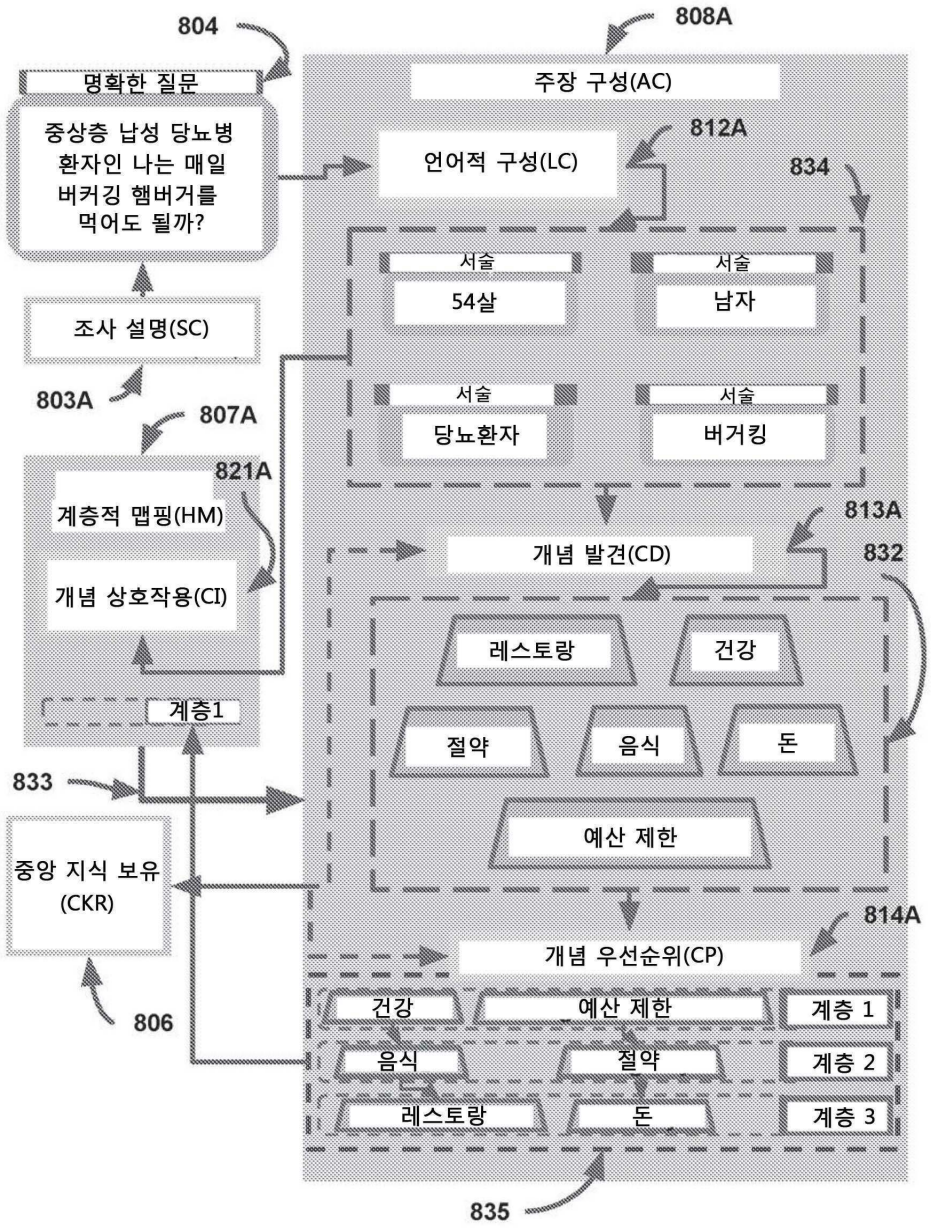
도면130


800
801
질문
중앙 지식 보유(CKR)
인간 주체
나는 패스트푸드를 먹어야 할까?
806
초기 질문 추론(IQR)
812A
언어적 구성(LC)
명사
패스트 푸드
질문
?
828
825
주체 질문
826
구어적 의미
기술적 의미
827
버거 가게
쥬스 가게
치킨 가게
피자 가게
샐러드 바
802A

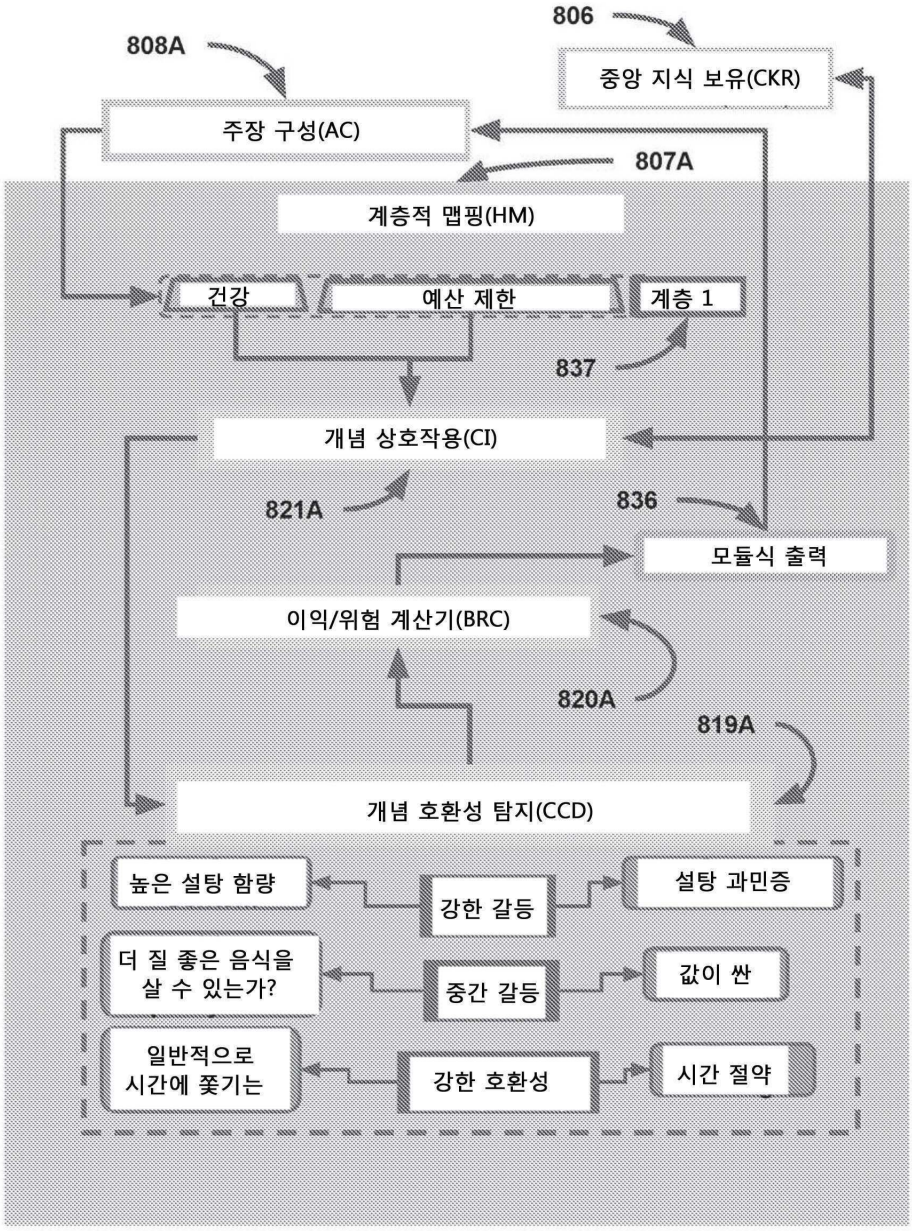
도면131


초기 질문 추론(IQR)
매일?
단일 이벤트?
당뇨 환자?
구매력?
802A
조사 설명(SC)
801
질문
나는 패스트푸드를 먹어야 할까?
830
명확화 언어적 연관성
요청된 명확화
829
언어적 구성(LC)
인간 주체
812A
816A
인간 인터페이스 모듈 (HIM)
800
804
명확한 질문
응답 분리 로직(RSL)
831
인간 응답
815A
응답 분리 로직(RSL)
매일이 아닌
일주일에 한번 발생
당뇨환자 맞음
편안한 중상층

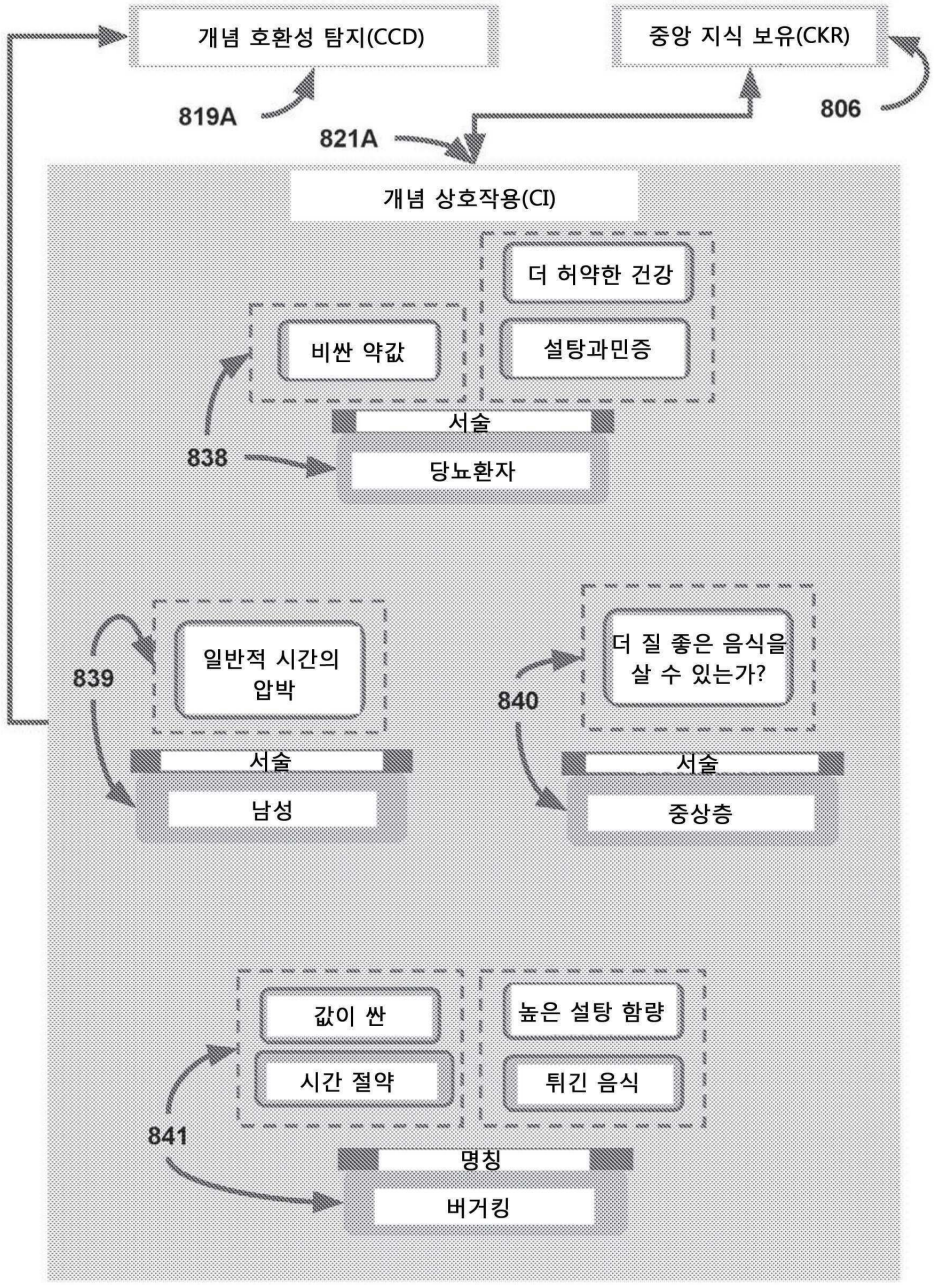
도면132


804
808A
명확한 질문
주장 구성(AC)
중상층 납성 당뇨병 환자인 나는 매일 버커깅 햄버거를 먹어도 될까?
812A
언어적 구성(LC)
834
서술
54살
서술
남자
조사 설명(SC)
서술
당뇨환자
서술
버거킹
803A
807A
821A
계층적 맵핑(HM)
개념 상호작용(CI)
813A
개념 발견(CD)
832
레스토랑
건강
계층1
절약
음식
돈
833
예산 제한
중앙 지식 보유 (CKR)
814A
개념 우선순위(CP)
건강
예산 제한
계층 1
음식
절약
계층 2
레스토랑
돈
계층 3
806
835

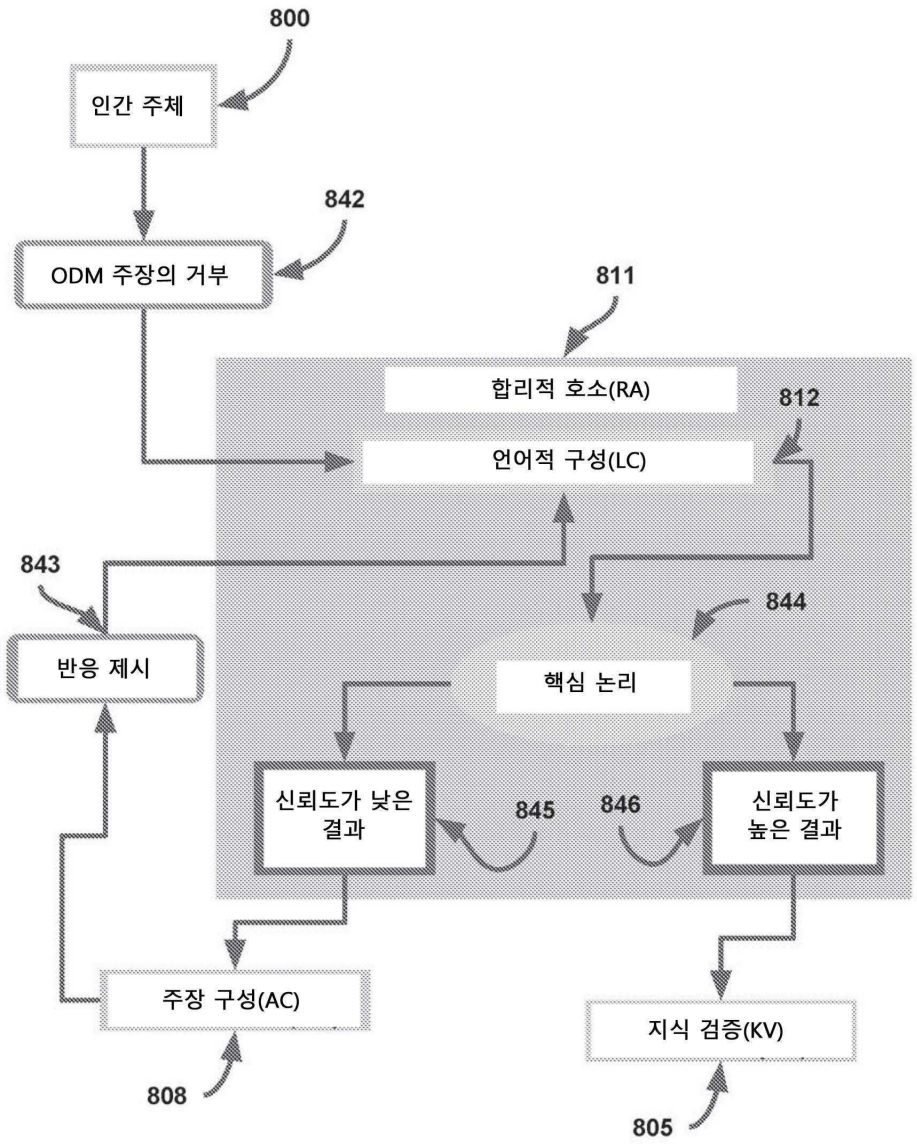
도면133


806
808A
중앙 지식 보유(CKR)
주장 구성(AC)
807A
계층적 맵핑(HM)
건강
예산 제한
계층 1
837
개념 상호작용(CI)
821A
836
모듈식 출력
이익/위험 계산기(BRC)
820A
819A
개념 호환성 탐지(CCD)
높은 설탕 함량
강한 갈등
설탕 과민증
더 질 좋은 음식을 살 수 있는가?
중간 갈등
값이 싼
일반적으로 시간에 쫓기는
강한 호환성
시간 절약

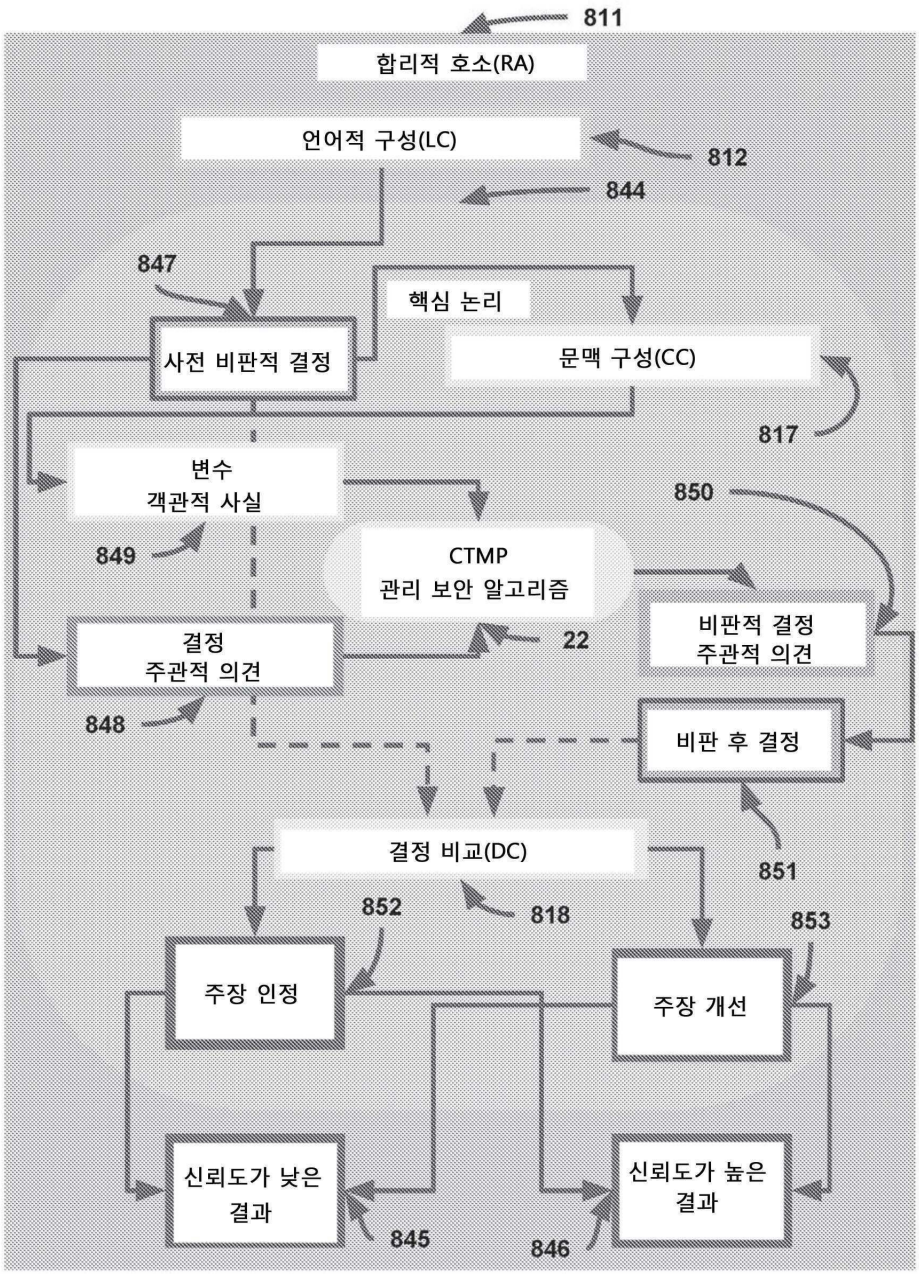
도면134


개념 호환성 탐지(CCD)
중앙 지식 보유(CKR)
819A
821A
806
개념 상호작용(CI)
더 허약한 건강
비싼 약값
설탕과민증
서술
838
당뇨환자
839
일반적 시간의 압박
서술
남성
840
더 질 좋은 음식을 살 수 있는가?
서술
중상층
값이 싼
높은 설탕 함량
시간 절약
튀긴 음식
841
명칭
버거킹

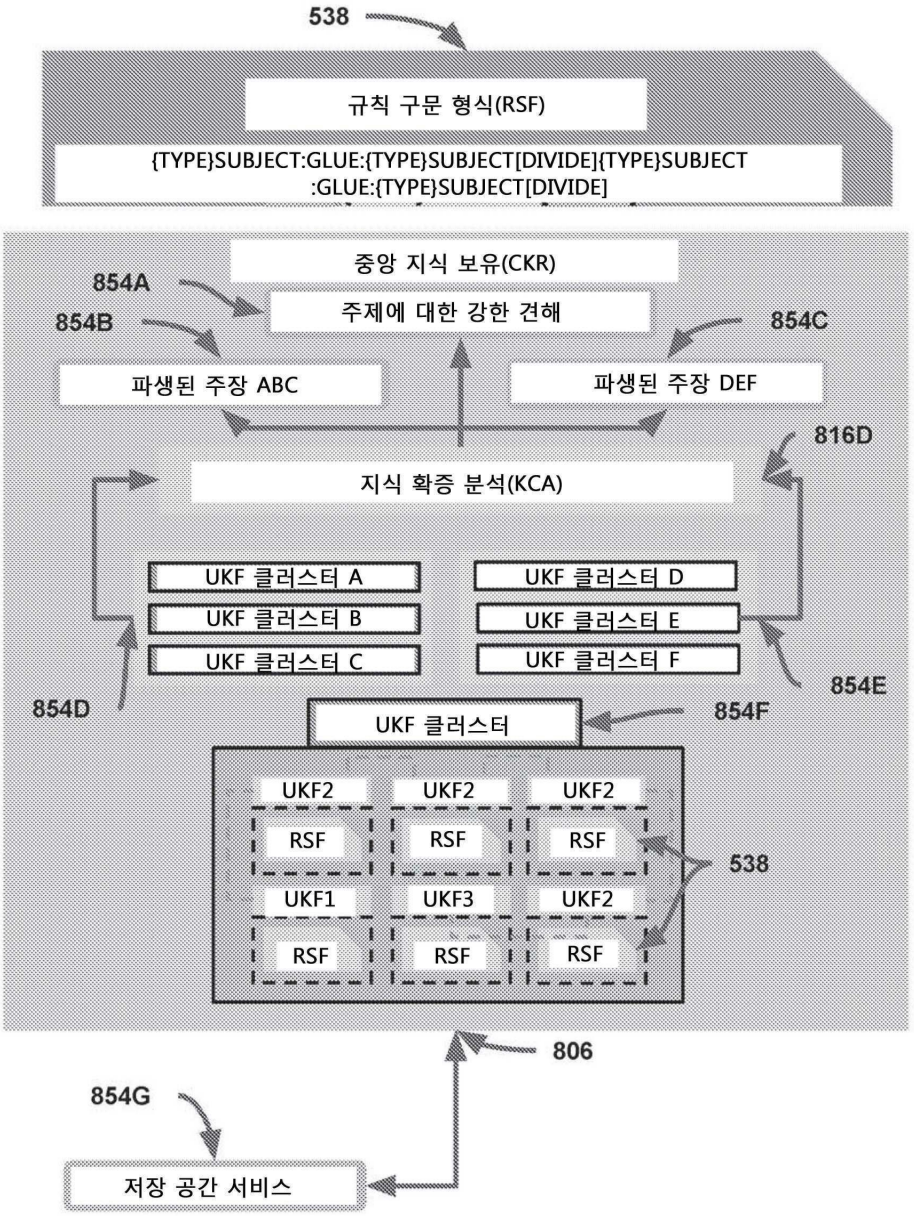
도면135


800
인간 주체
842
ODM 주장의 거부
811
합리적 호소(RA)
812
언어적 구성(LC)
843
844
반응 제시
핵심 논리
신뢰도가 낮은 결과
845
846
신뢰도가 높은 결과
주장 구성(AC)
지식 검증(KV)
808
805

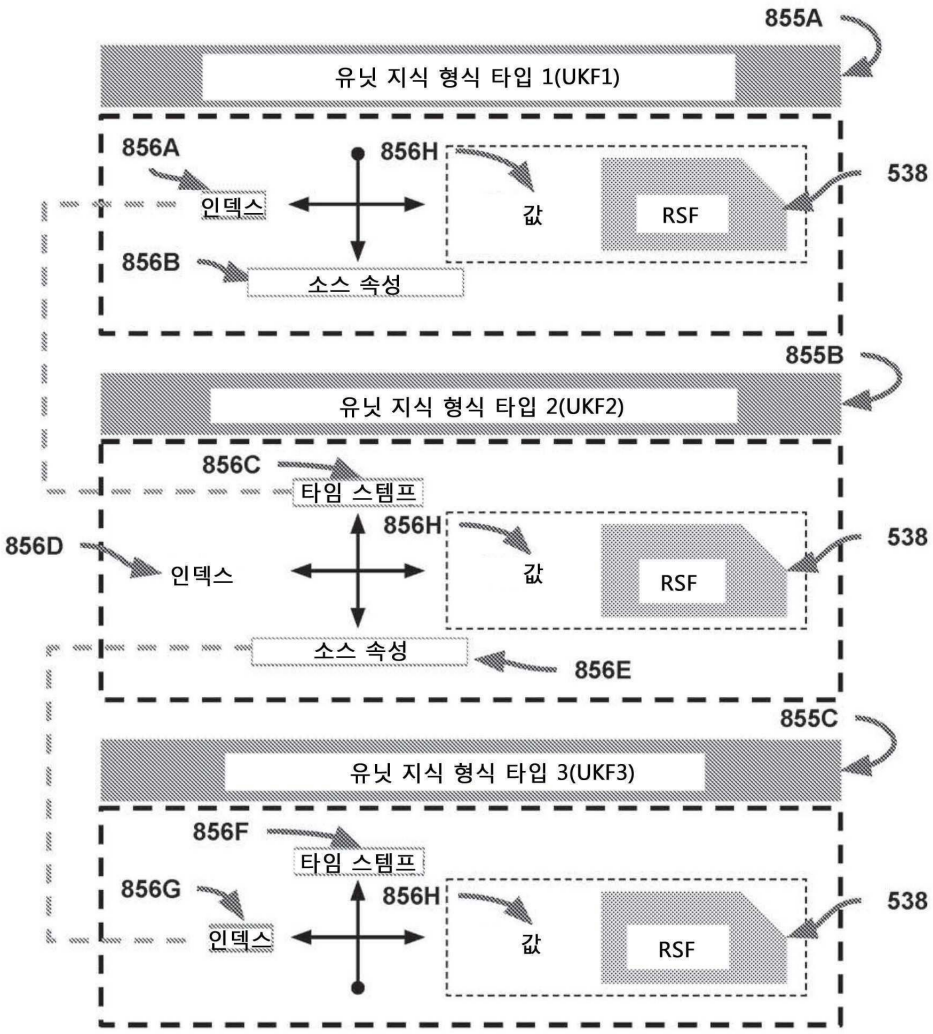
도면136


811
합리적 호소(RA)
언어적 구성(LC)
812
844
847
핵심 논리
사전 비판적 결정
문맥 구성(CC)
817
변수
객관적 사실
850
849
CTMP
관리 보안 알고리즘
비판적 결정
주관적 의견
결정
주관적 의견
22
848
비판 후 결정
851
결정 비교(DC)
852
818
853
주장 인정
주장 개선
신뢰도가 낮은
결과
신뢰도가 높은
결과
845
846

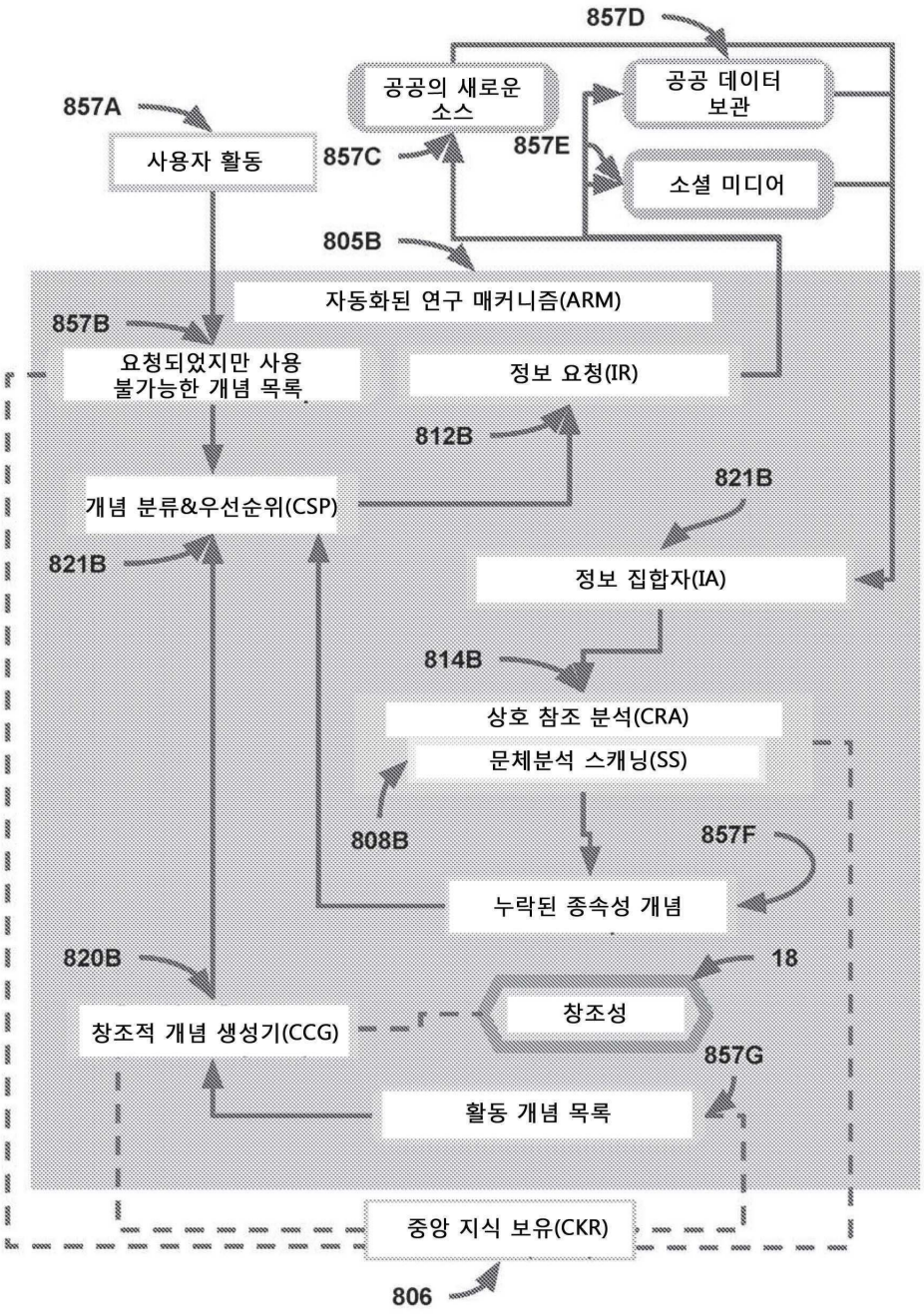
도면137


538
규칙 구문 형식(RSF)
{TYPE}SUBJECT:GLUE:{TYPE}SUBJECT[DIVIDE]{TYPE}SUBJECT
:GLUE:{TYPE}SUBJECT[DIVIDE]
중앙 지식 보유(CKR)
854A
주제에 대한 강한 견해
854B
854C
파생된 주장 ABC
파생된 주장 DEF
816D
지식 확증 분석(KCA)
UKF 클러스터 A
UKF 클러스터 B
UKF 클러스터 C
UKF 클러스터 D
UKF 클러스터 E
UKF 클러스터 F
854D
854E
UKF 클러스터
854F
UKF2
UKF2
UKF2
RSF
RSF
RSF
538
UKF1
UKF3
UKF2
RSF
RSF
RSF
806
854G
저장 공간 서비스

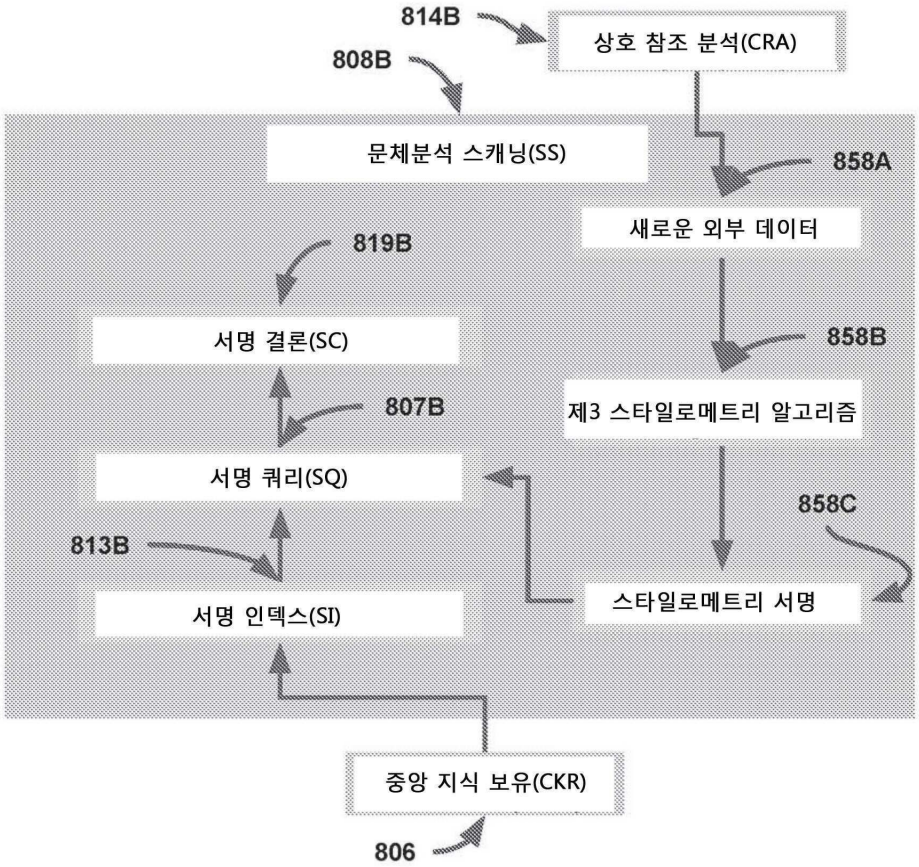
도면138


855A
유닛 지식 형식 타입 1(UKF1)
856A
인덱스
856H
값
RSF
538
856B
소스 속성
855B
유닛 지식 형식 타입 2(UKF2)
856C
타임 스템프
856D
인덱스
856H
값
RSF
538
소스 속성
856E
855C
유닛 지식 형식 타입 3(UKF3)
856F
타임 스템프
856G
인덱스
856H
값
RSF
538

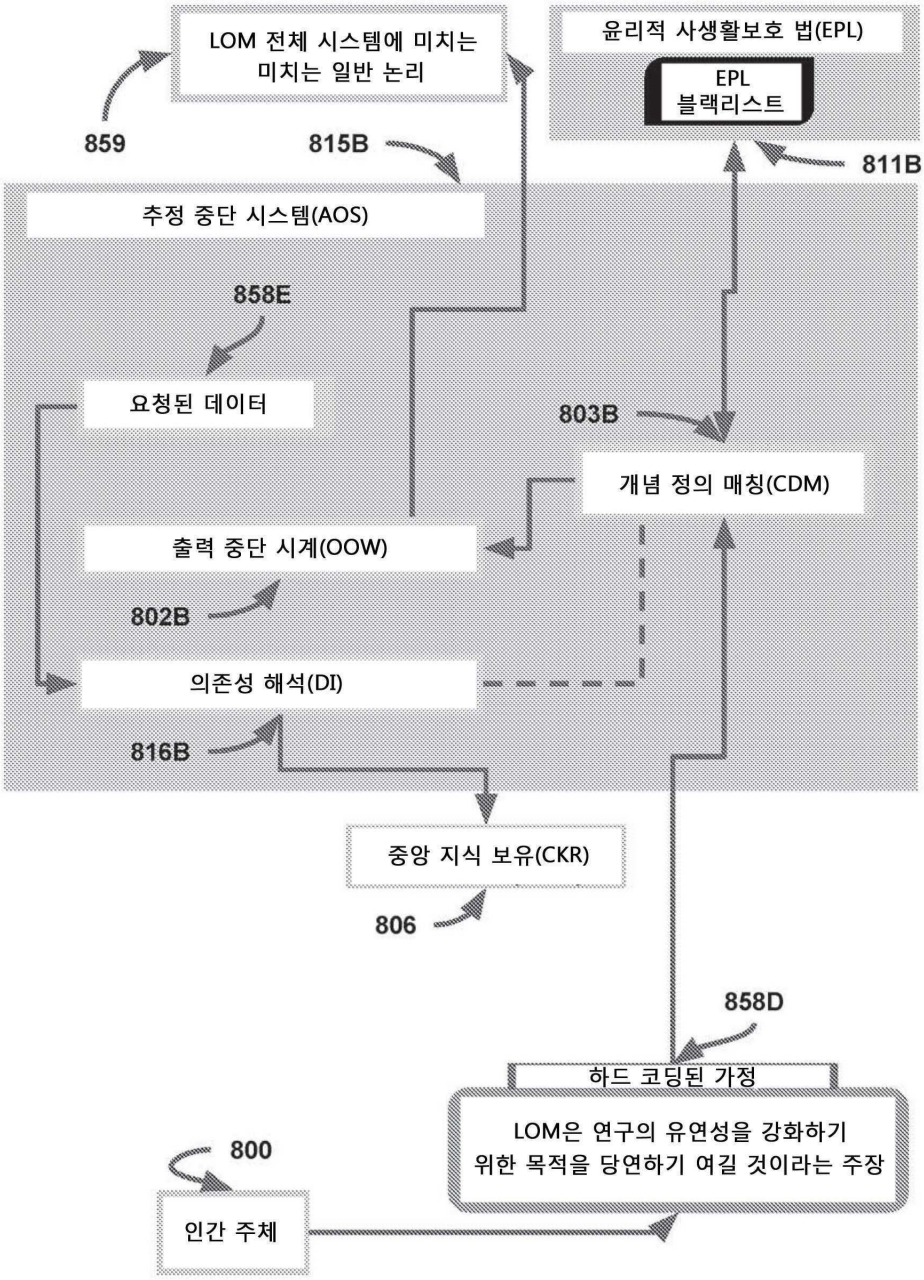
도면139


857D
공공의 새로운 소스
공공 데이터 보관
857A
사용자 활동
857C
857E
소셜 미디어
805B
자동화된 연구 매커니즘(ARM)
857B
요청되었지만 사용 불가능한 개념 목록
정보 요청(IR)
812B
821B
개념 분류&우선순위(CSP)
821B
정보 집합자(IA)
814B
상호 참조 분석(CRA)
문체분석 스캐닝(SS)
808B
857F
누락된 종속성 개념
820B
18
창조성
창조적 개념 생성기(CCG)
857G
활동 개념 목록
중앙 지식 보유(CKR)
806

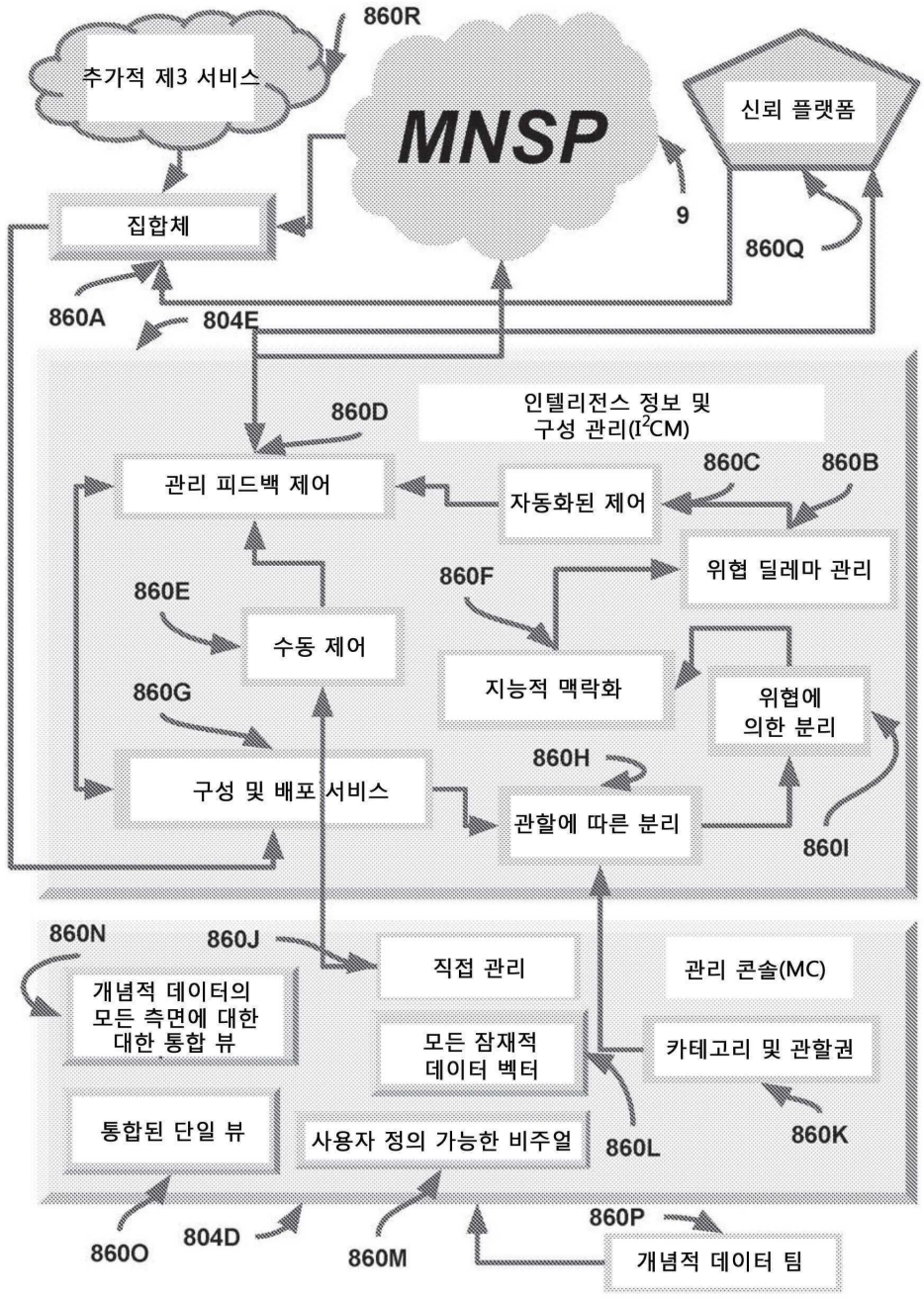
도면140


814B
상호 참조 분석(CRA)
808B
문체분석 스캐닝(SS)
858A
새로운 외부 데이터
819B
서명 결론(SC)
858B
제3 스타일로메트리 알고리즘
807B
서명 쿼리(SQ)
858C
813B
스타일로메트리 서명
서명 인덱스(SI)
중앙 지식 보유(CKR)
806

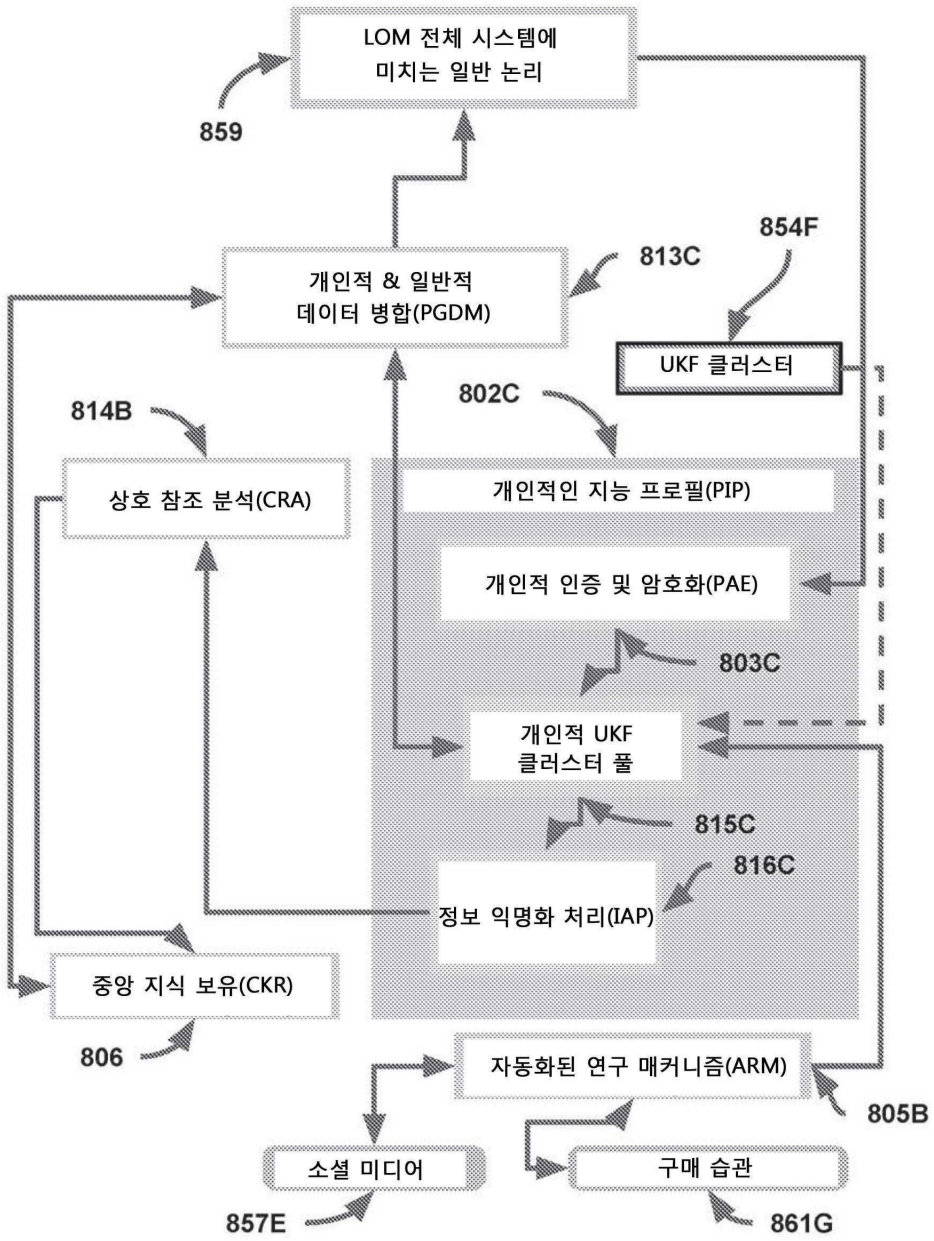
도면141


LOM 전체 시스템에 미치는
미치는 일반 논리
859
815B
윤리적 사생활보호 법(EPL)
EPL
블랙리스트
811B
추정 중단 시스템(AOS)
858E
요청된 데이터
803B
개념 정의 매칭(CDM)
출력 중단 시계(OOW)
802B
의존성 해석(DI)
816B
중앙 지식 보유(CKR)
806
858D
하드 코딩된 가정
LOM은 연구의 유연성을 강화하기
위한 목적을 당연하기 여길 것이라는 주장
800
인간 주체

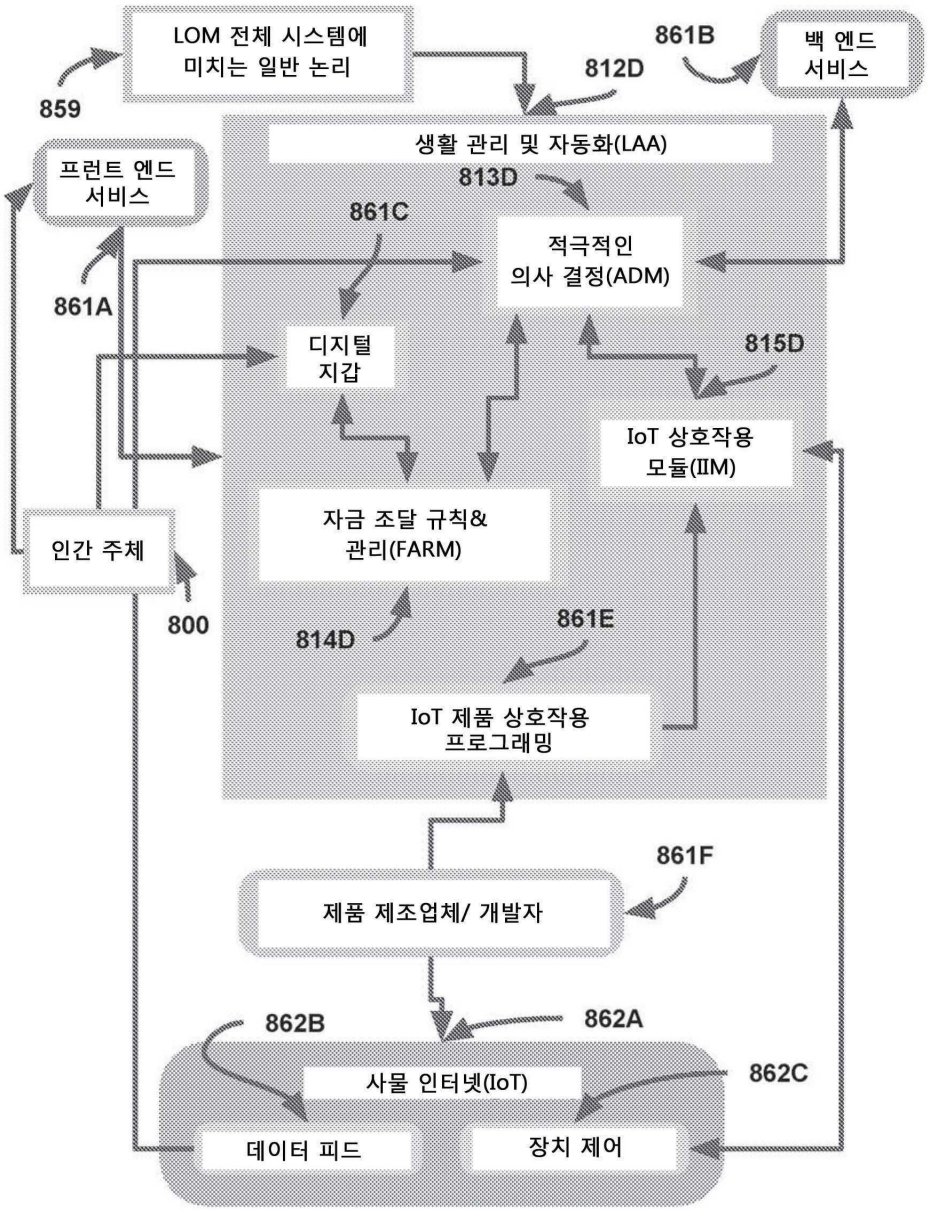
도면142


860R
추가적 제3 서비스
MNSP
신뢰 플랫폼
집합체
9
860Q
860A
804E
860D
인텔리전스 정보 및
구성 관리(I2CM)
860C
860B
관리 피드백 제어
자동화된 제어
위협 딜레마 관리
860F
860E
수동 제어
지능적 맥락화
위협에
의한 분리
860G
860H
구성 및 배포 서비스
관할에 따른 분리
860I
860N
860J
직접 관리
관리 콘솔(MC)
개념적 데이터의
모든 측면에 대한
대한 통합 뷰
모든 잠재적
데이터 벡터
카테고리 및 관할권
통합된 단일 뷰
사용자 정의 가능한 비주얼
860L
860K
860O
804D
860M
860P
개념적 데이터 팀

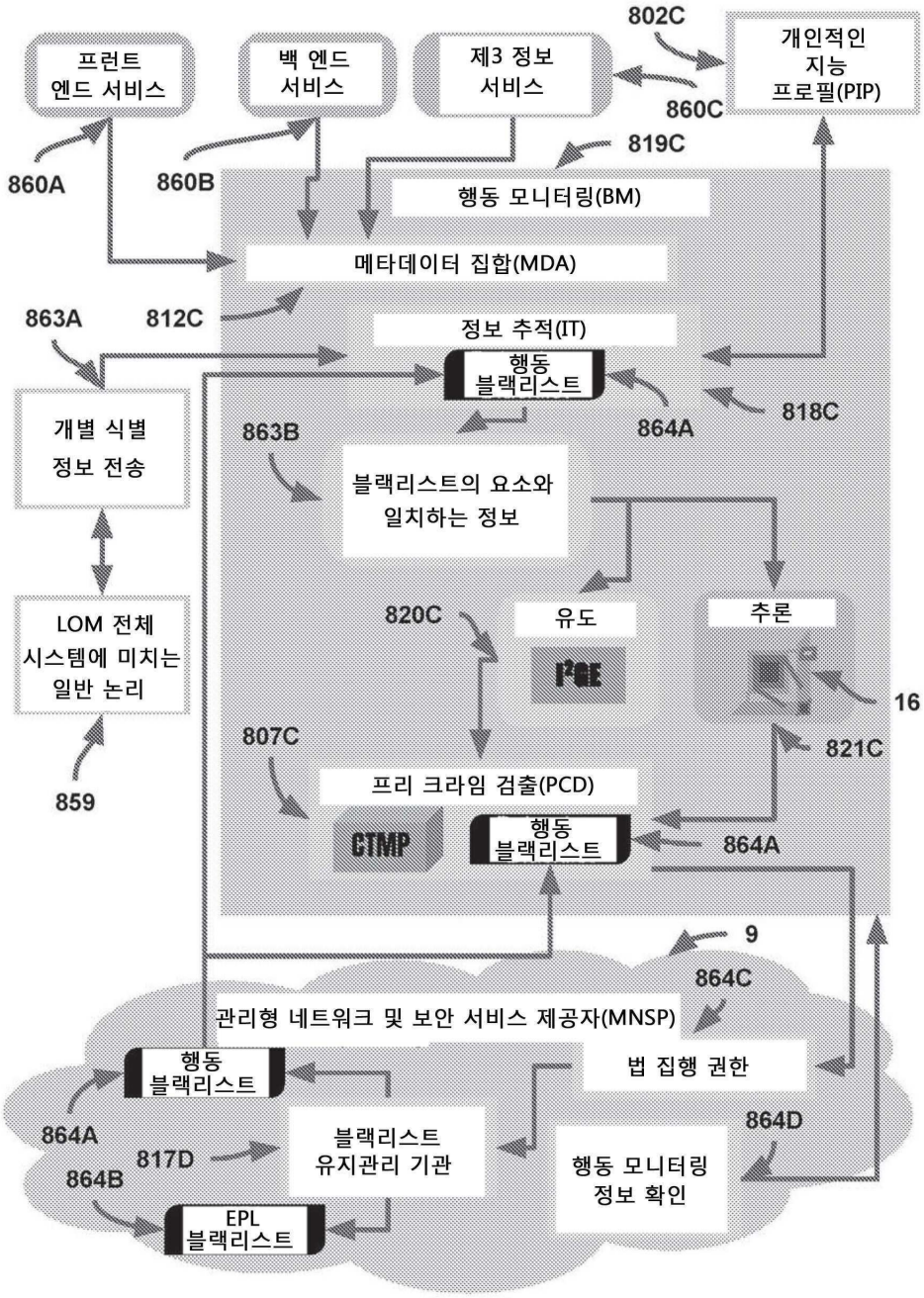
도면143


LOM 전체 시스템에
미치는 일반 논리
859
개인적 & 일반적
데이터 병합(PGDM)
813C
854F
UKF 클러스터
802C
814B
상호 참조 분석(CRA)
개인적인 지능 프로필(PIP)
개인적 인증 및 암호화(PAE)
803C
개인적 UKF
클러스터 풀
815C
816C
정보 익명화 처리(IAP)
중앙 지식 보유(CKR)
806
자동화된 연구 매커니즘(ARM)
805B
소셜 미디어
857E
구매 습관
861G

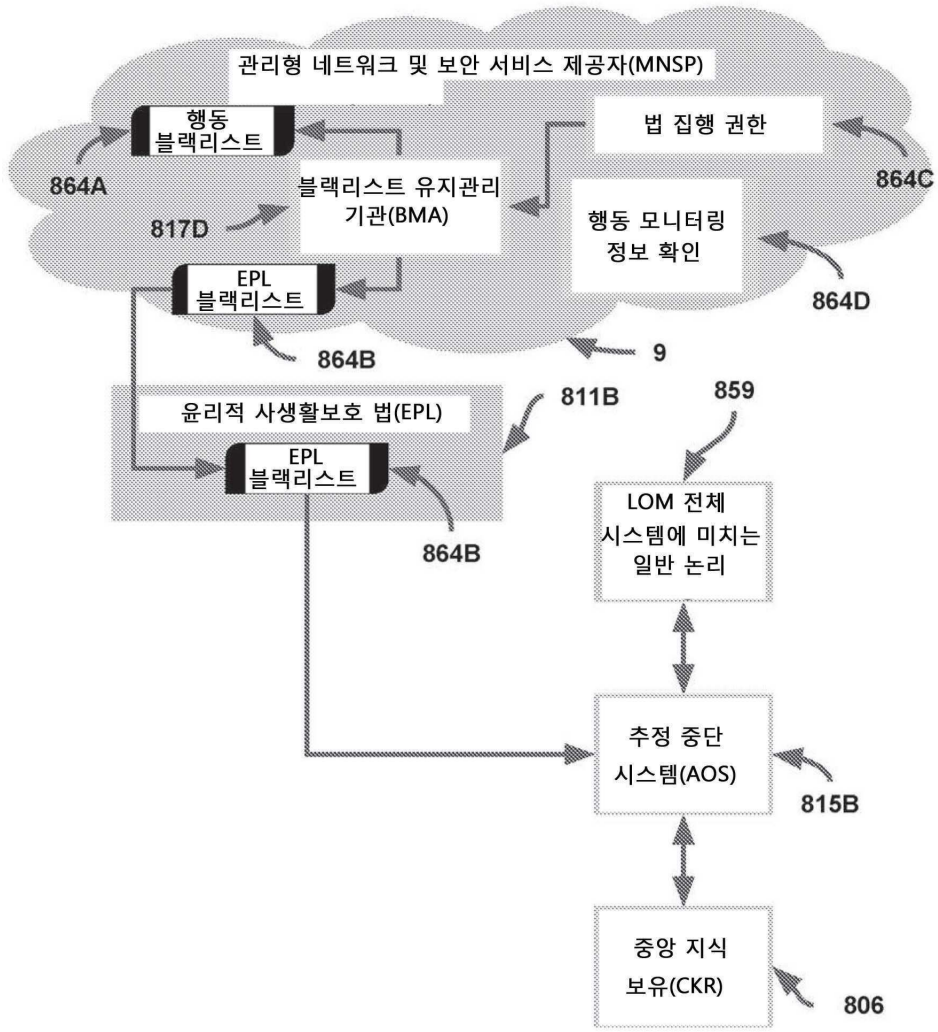
도면144


LOM 전체 시스템에
미치는 일반 논리
859
861B
812D
백 엔드
서비스
생활 관리 및 자동화(LAA)
813D
프런트 엔드
서비스
861C
적극적인
의사 결정(ADM)
861A
디지털
지갑
815D
IoT 상호작용
모듈(IIM)
자금 조달 규칙&
관리(FARM)
인간 주체
800
814D
861E
IoT 제품 상호작용
프로그래밍
861F
제품 제조업체/ 개발자
862B
862A
862C
사물 인터넷(IoT)
데이터 피드
장치 제어

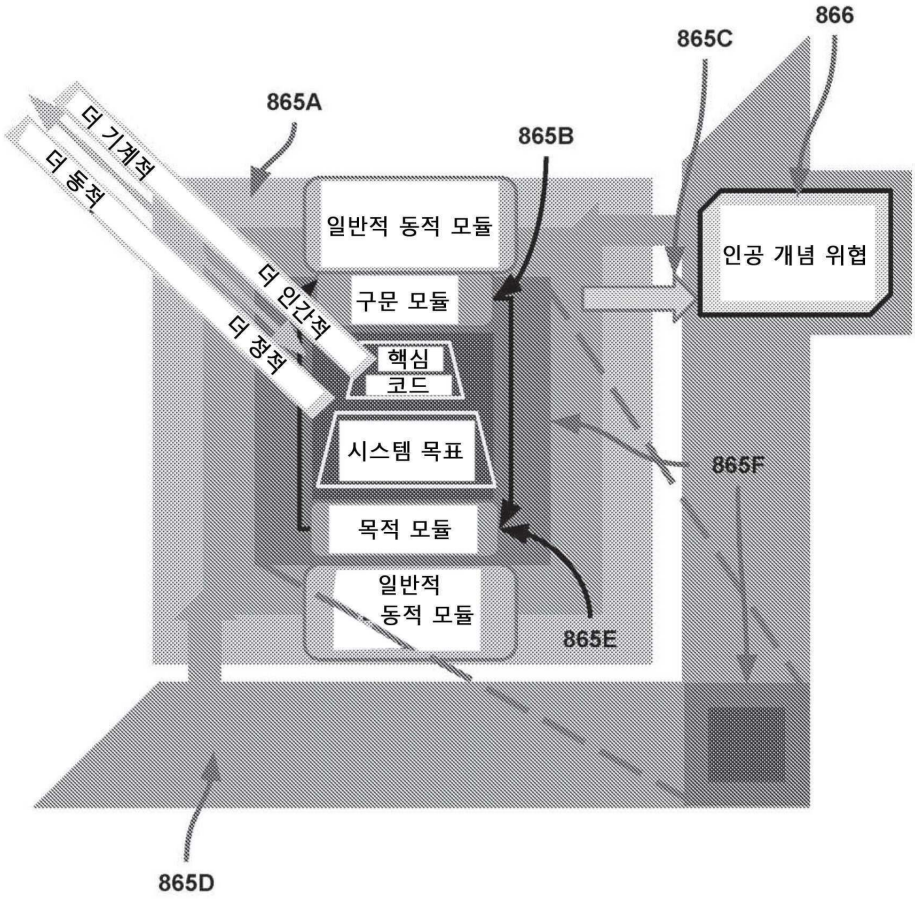
도면145


802C
프런트 엔드 서비스
백 엔드 서비스
제3 정보 서비스
개인적인 지능 프로필(PIP)
860C
819C
860A
860B
행동 모니터링(BM)
메타데이터 집합(MDA)
863A
812C
정보 추적(IT)
행동 블랙리스트
818C
864A
개별 식별 정보 전송
863B
블랙리스트의 요소와 일치하는 정보
LOM 전체 시스템에 미치는 일반 논리
820C
유도
추론
16
807C
821C
859
프리 크라임 검출(PCD)
CTMP
행동 블랙리스트
864A
9
864C
관리형 네트워크 및 보안 서비스 제공자(MNSP)
행동 블랙리스트
법 집행 권한
864A
817D
864D
블랙리스트 유지관리 기관
행동 모니터링 정보 확인
864B
EPL 블랙리스트

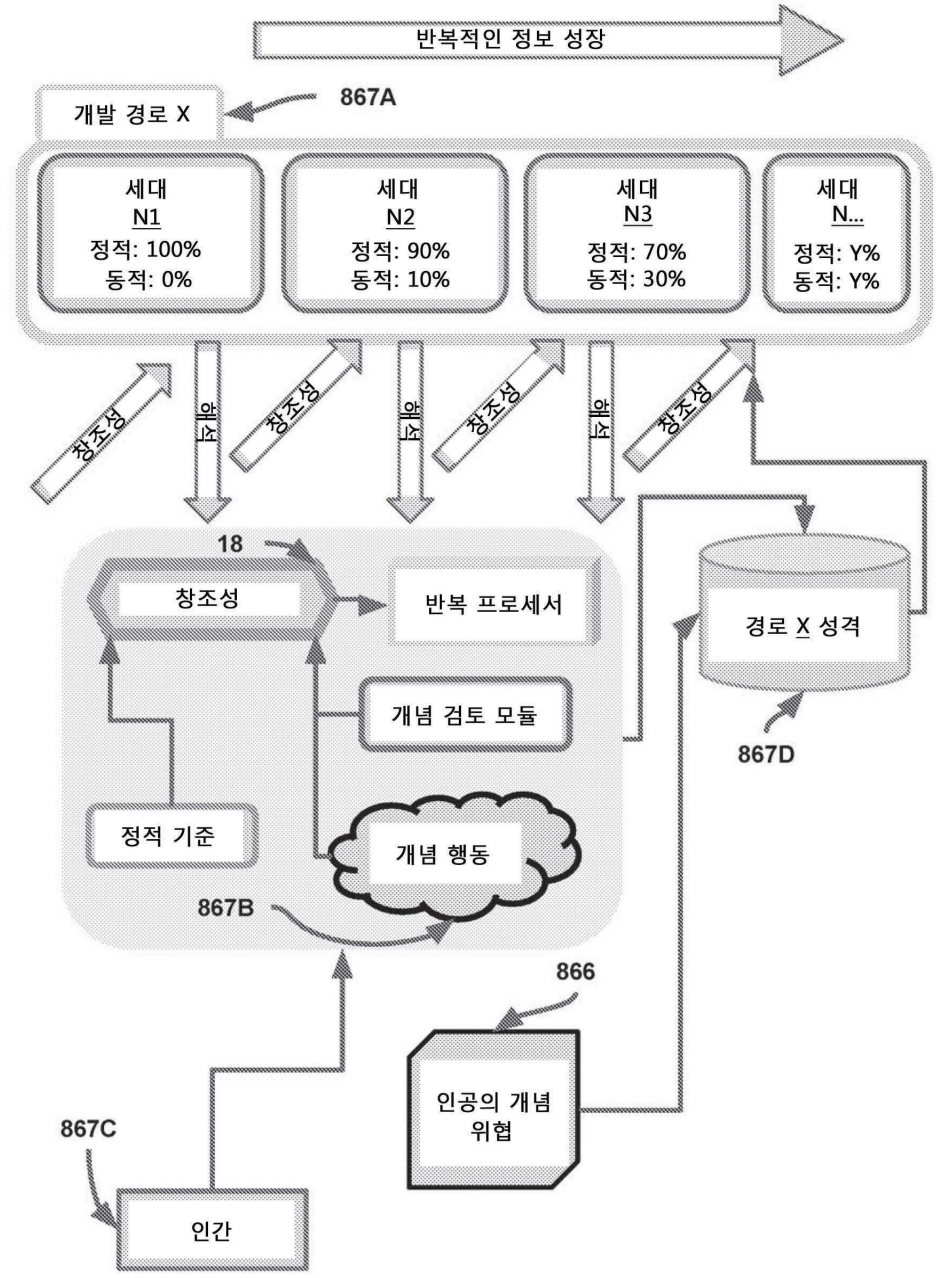
도면146


관리형 네트워크 및 보안 서비스 제공자(MNSP)
행동 블랙리스트
864A
블랙리스트 유지관리 기관(BMA)
817D
법 집행 권한
864C
행동 모니터링 정보 확인
864D
EPL 블랙리스트
864B
9
윤리적 사생활보호 법(EPL)
811B
859
EPL 블랙리스트
864B
LOM 전체 시스템에 미치는 일반 논리
추정 중단 시스템(AOS)
815B
중앙 지식 보유(CKR)
806

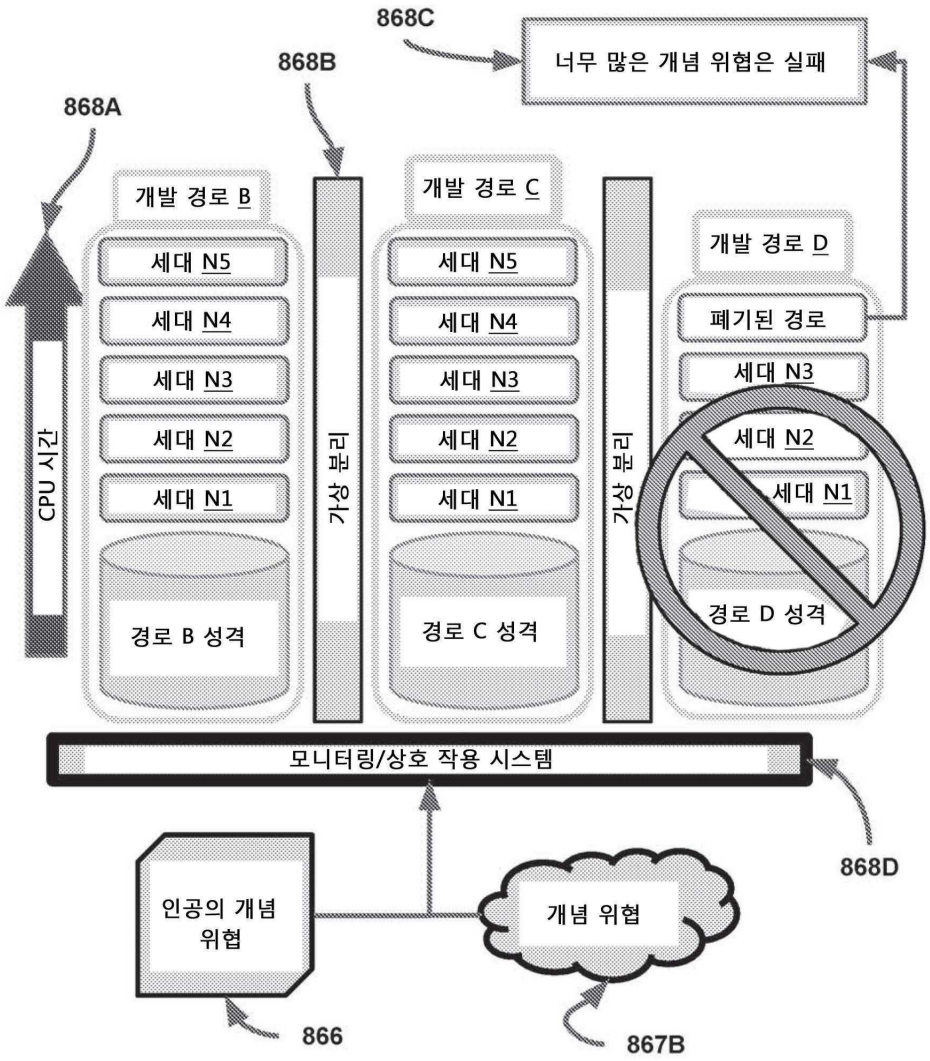
도면147


866
865C
865A
865B
더 기계적
더 동적
더 인간적
더 정적
일반적 동적 모듈
구문 모듈
핵심
코드
시스템 목표
목적 모듈
일반적
동적 모듈
인공 개념 위협
865F
865E
865D

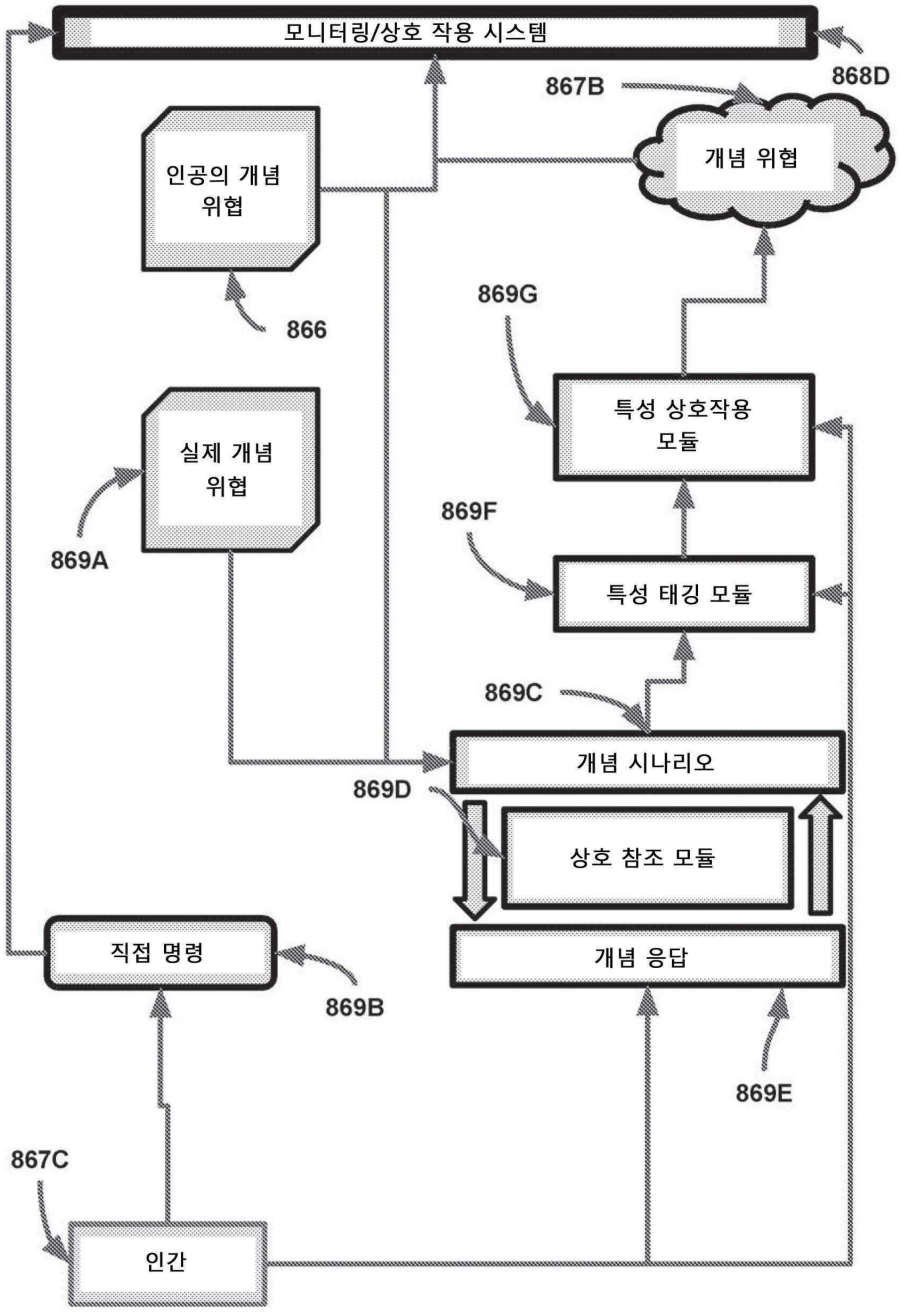
도면148


반복적인 정보 성장
개발 경로 X
867A
세대
N1
정적: 100%
동적: 0%
세대
N2
정적: 90%
동적: 10%
세대
N3
정적: 70%
동적: 30%
세대
N...
정적: Y%
동적: Y%
창조성
파괴
창조성
파괴
창조성
파괴
창조성
18
창조성
반복 프로세서
개념 검토 모듈
정적 기준
개념 행동
867B
경로 X 성격
867D
866
인공의 개념
위협
867C
인간

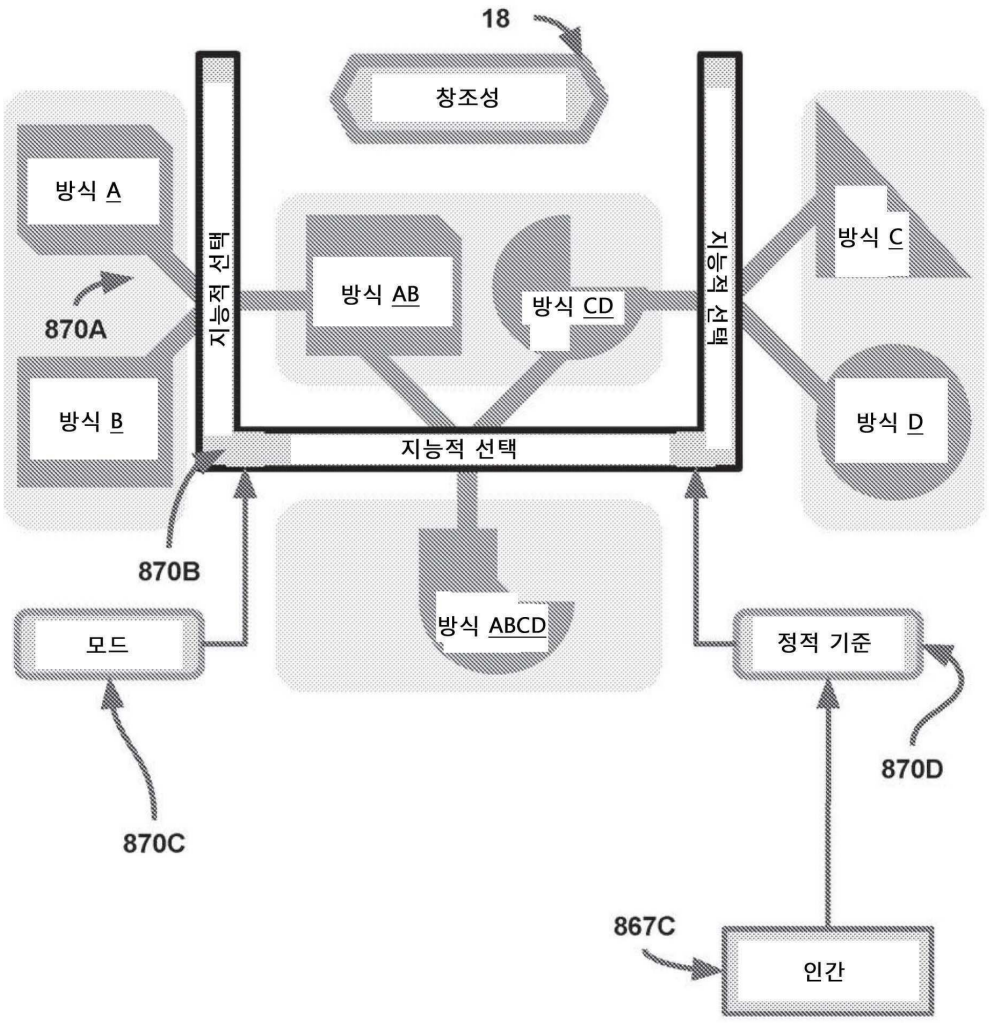
도면149


868C
너무 많은 개념 위협은 실패
868B
868A
개발 경로 B
세대 N5
세대 N4
세대 N3
세대 N2
세대 N1
경로 B 성격
CPU 시간
가상 분리
개발 경로 C
세대 N5
세대 N4
세대 N3
세대 N2
세대 N1
경로 C 성격
가상 분리
개발 경로 D
폐기된 경로
세대 N3
세대 N2
세대 N1
경로 D 성격
모니터링/상호 작용 시스템
868D
인공의 개념 위협
개념 위협
866
867B

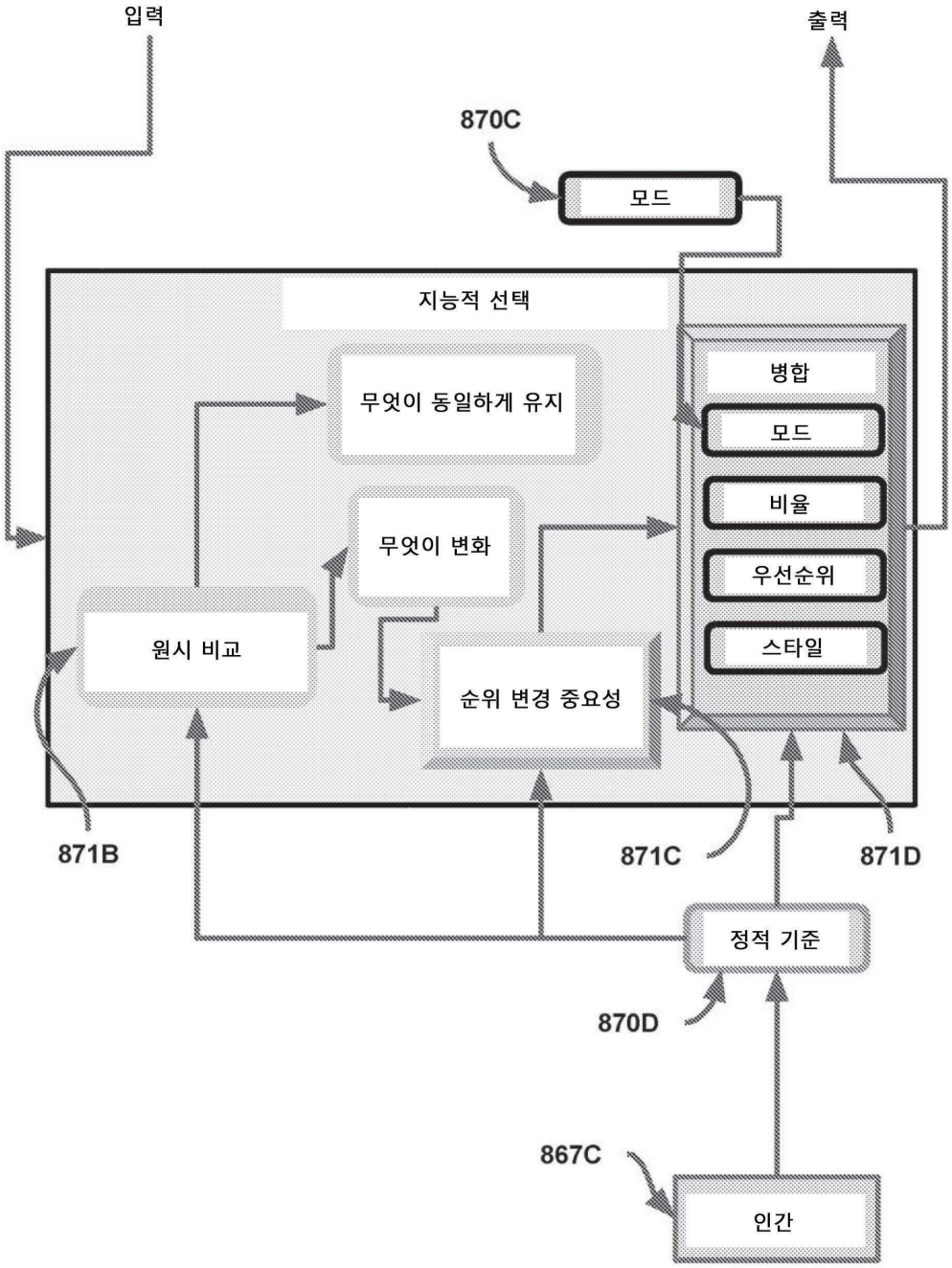
도면150


모니터링/상호 작용 시스템
867B
868D
개념 위협
인공의 개념 위협
866
869G
특성 상호작용 모듈
실제 개념 위협
869A
869F
특성 태깅 모듈
869C
개념 시나리오
869D
상호 참조 모듈
직접 명령
869B
개념 응답
869E
867C
인간

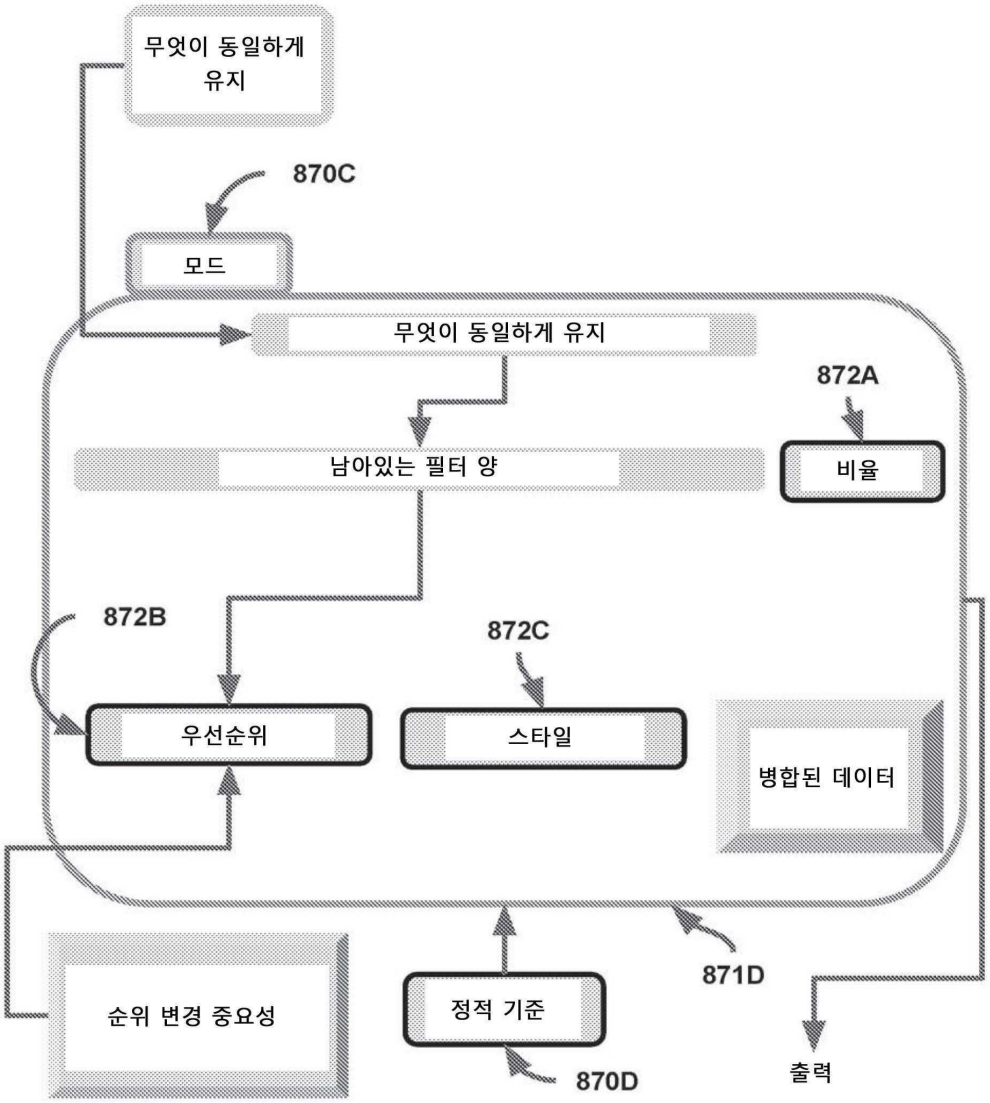
도면151


18
창조성
방식 A
870A
방식 B
지능적 선택
방식 AB
방식 CD
지능적 선택
방식 C
방식 D
지능적 선택
870B
모드
방식 ABCD
정적 기준
870D
870C
867C
인간

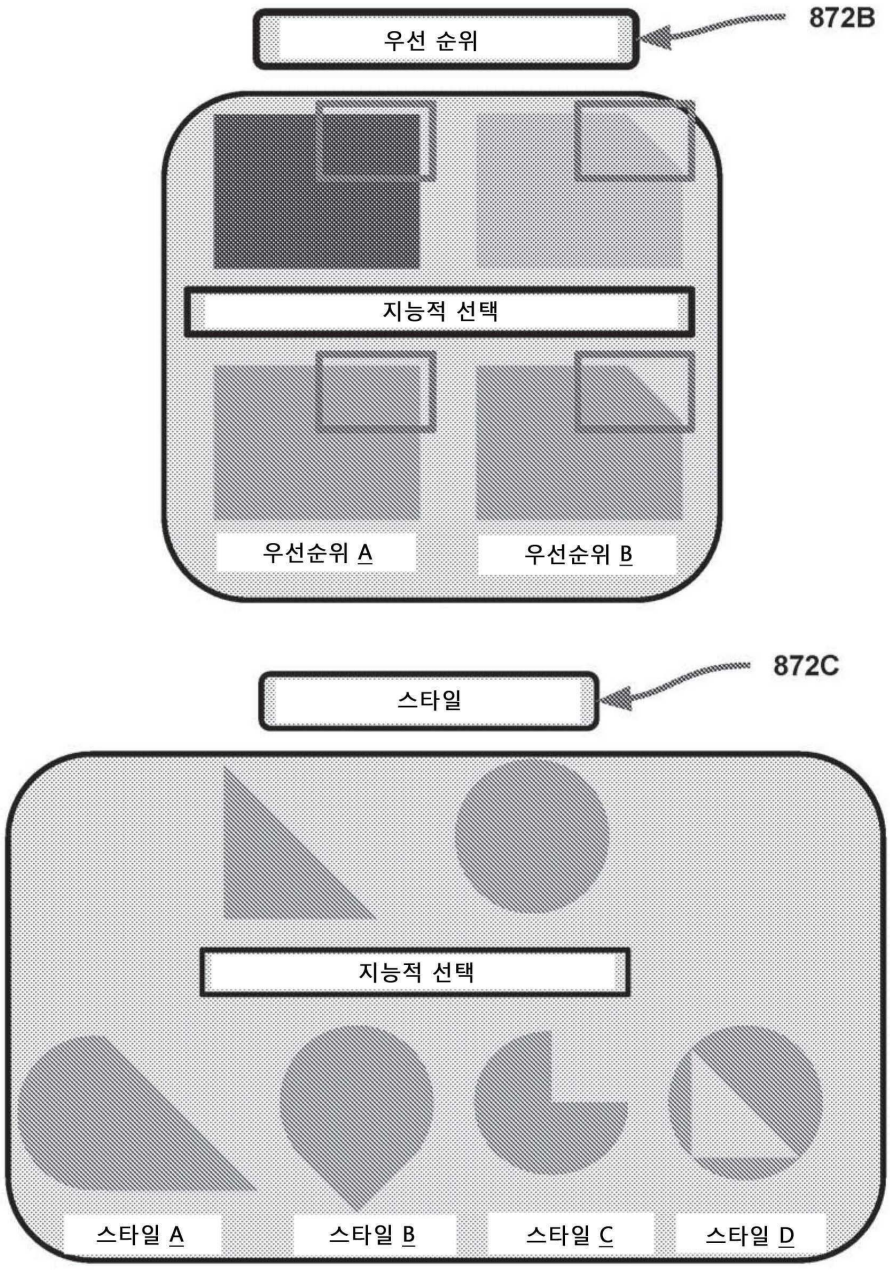
도면152


입력
출력
870C
모드
지능적 선택
무엇이 동일하게 유지
무엇이 변화
원시 비교
순위 변경 중요성
병합
모드
비율
우선순위
스타일
871B
871C
871D
정적 기준
870D
867C
인간

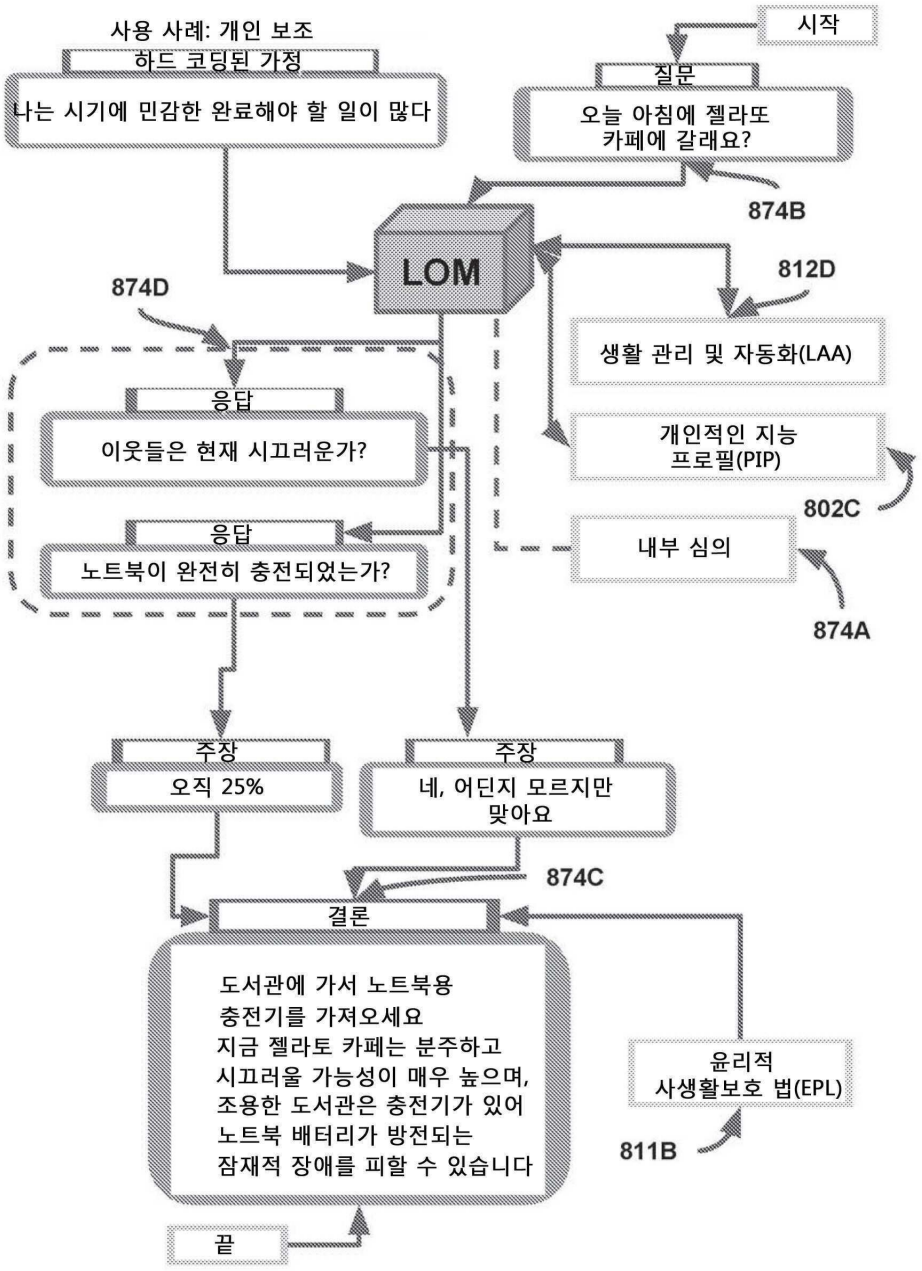
도면153


무엇이 동일하게 유지
870C
모드
무엇이 동일하게 유지
872A
남아있는 필터 양
비율
872B
872C
우선순위
스타일
병합된 데이터
871D
순위 변경 중요성
정적 기준
870D
출력

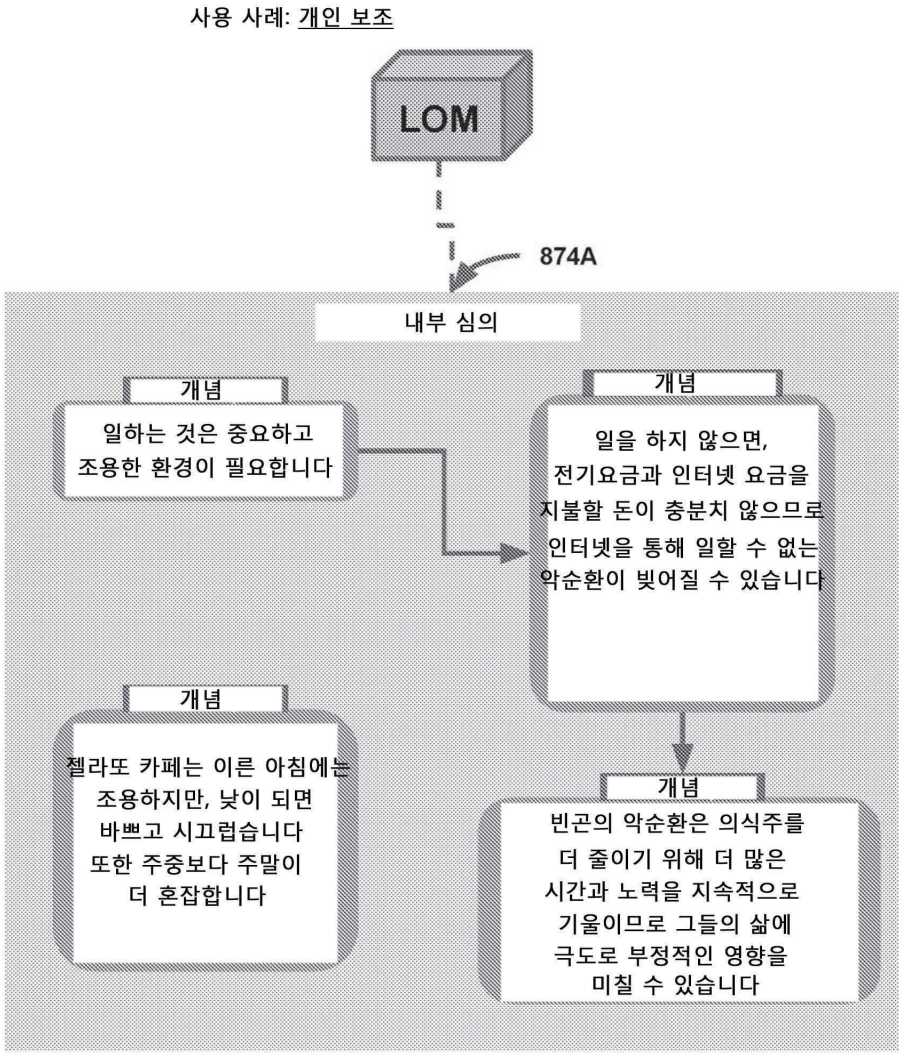
도면154


872B
우선 순위
지능적 선택
우선순위 A
우선순위 B
872C
스타일
지능적 선택
스타일 A
스타일 B
스타일 C
스타일 D

도면155

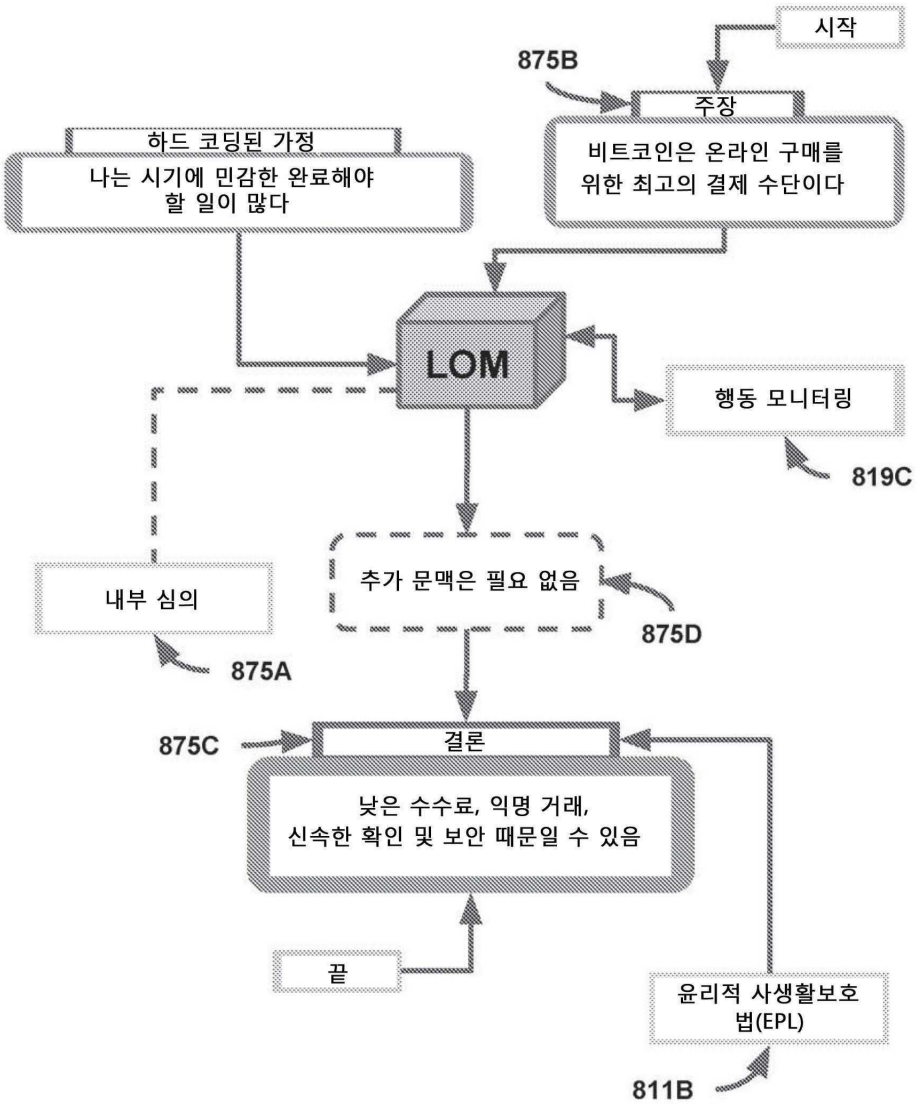

사용 사례: 개인 보조
하드 코딩된 가정
나는 시기에 민감한 완료해야 할 일이 많다
시작
질문
오늘 아침에 젤라또 카페에 갈래요?
874B
LOM
812D
874D
생활 관리 및 자동화(LAA)
응답
이웃들은 현재 시끄러운가?
개인적인 지능 프로필(PIP)
802C
응답
노트북이 완전히 충전되었는가?
내부 심의
874A
주장
오직 25%
주장
네, 어딘지 모르지만 맞아요
874C
결론
도서관에 가서 노트북용 충전기를 가져오세요
지금 젤라토 카페는 분주하고 시끄러울 가능성이 매우 높으며, 조용한 도서관은 충전기가 있어 노트북 배터리가 방전되는 잠재적 장애를 피할 수 있습니다
윤리적 사생활보호 법(EPL)
811B
끝

도면156

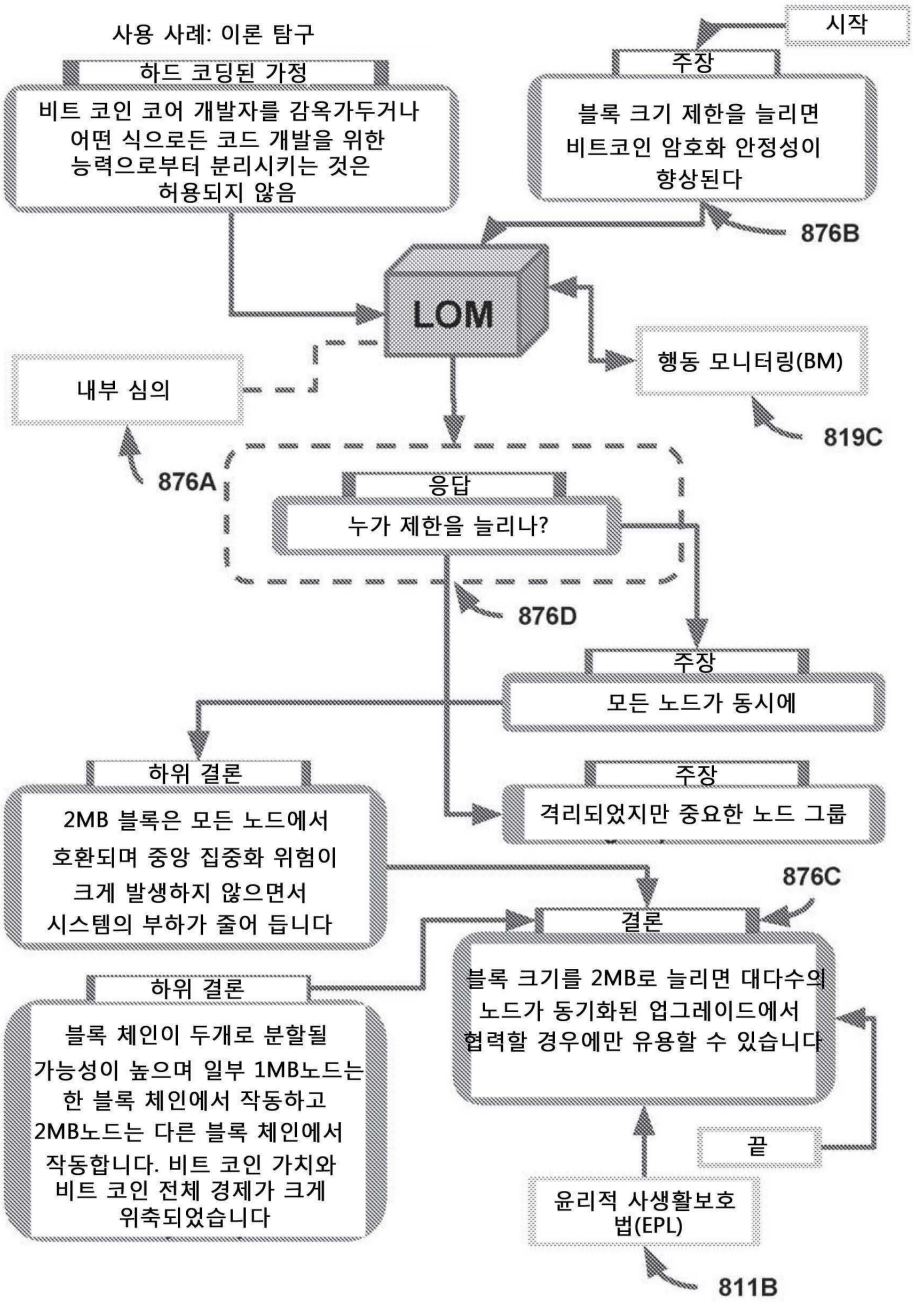

사용 사례: 개인 보조
LOM
874A
내부 심의
개념
일하는 것은 중요하고
조용한 환경이 필요합니다
개념
일을 하지 않으면,
전기요금과 인터넷 요금을
지불할 돈이 충분치 않으므로
인터넷을 통해 일할 수 없는
악순환이 빚어질 수 있습니다
개념
젤라또 카페는 이른 아침에는
조용하지만, 낮이 되면
바쁘고 시끄럽습니다
또한 주중보다 주말이
더 혼잡합니다
개념
빈곤의 악순환은 의식주를
더 줄이기 위해 더 많은
시간과 노력을 지속적으로
기울이므로 그들의 삶에
극도로 부정적인 영향을
미칠 수 있습니다

도면157

사용 사례: 연구 도구

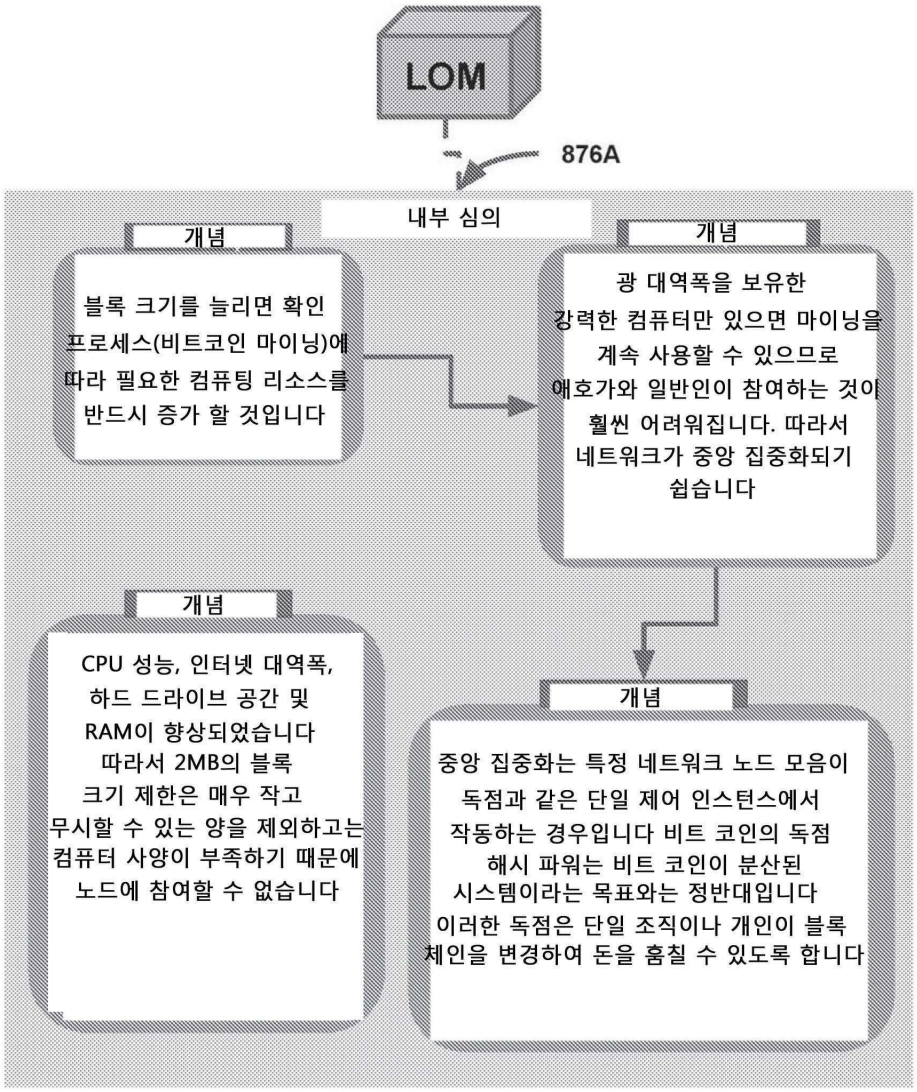


도면158

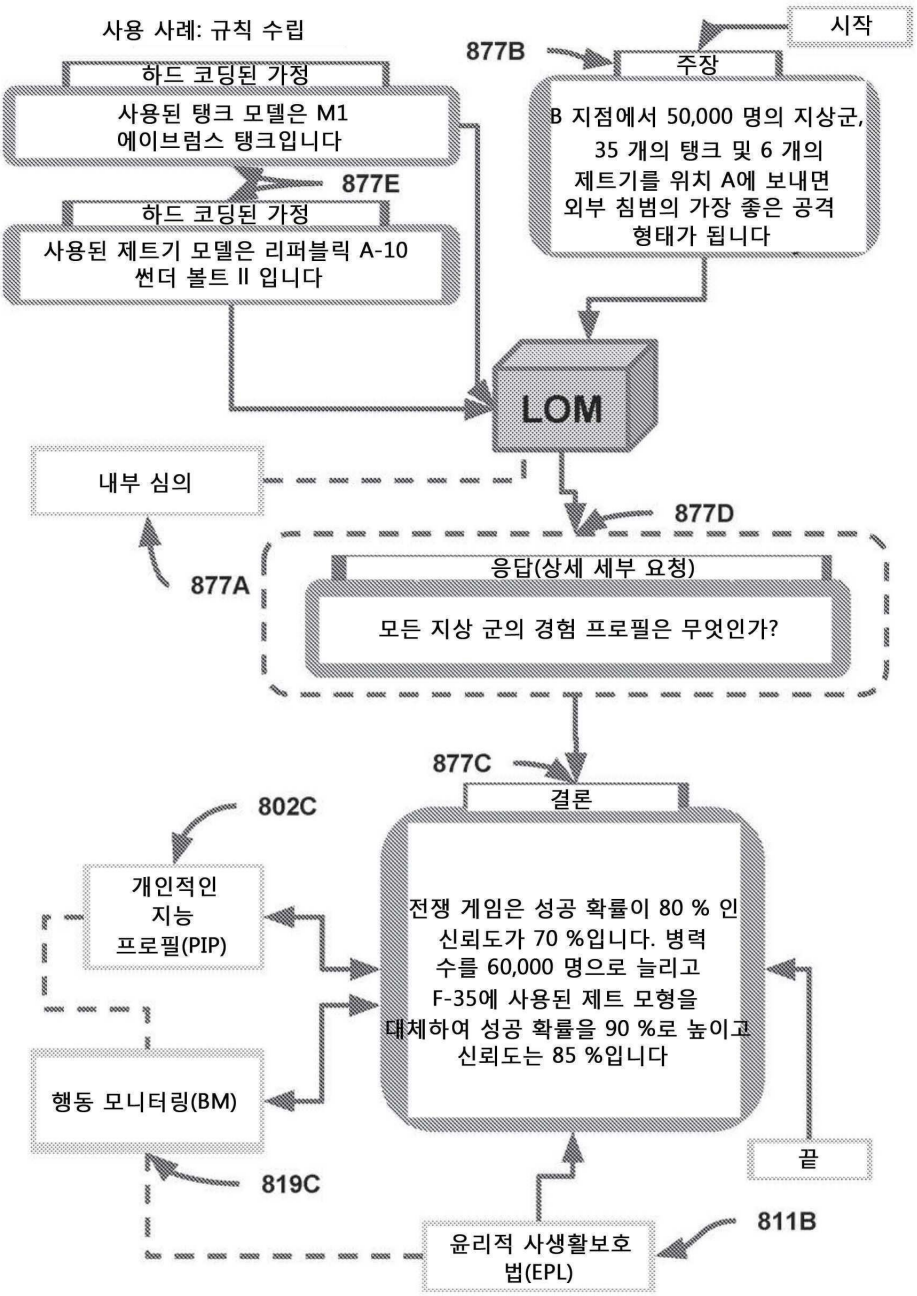

사용 사례: 이론 탐구
하드 코딩된 가정
비트 코인 코어 개발자를 감옥가두거나 어떤 식으로든 코드 개발을 위한 능력으로부터 분리시키는 것은 허용되지 않음
시작
주장
블록 크기 제한을 늘리면 비트코인 암호화 안정성이 향상된다
876B
LOM
내부 심의
행동 모니터링(BM)
819C
876A
응답
누가 제한을 늘리나?
876D
주장
모든 노드가 동시에
하위 결론
2MB 블록은 모든 노드에서 호환되며 중앙 집중화 위험이 크게 발생하지 않으면서 시스템의 부하가 줄어 듭니다
주장
격리되었지만 중요한 노드 그룹
876C
결론
블록 크기를 2MB로 늘리면 대다수의 노드가 동기화된 업그레이드에서 협력할 경우에만 유용할 수 있습니다
하위 결론
블록 체인이 두개로 분할될 가능성이 높으며 일부 1MB노드는 한 블록 체인에서 작동하고 2MB노드는 다른 블록 체인에서 작동합니다. 비트 코인 가치와 비트 코인 전체 경제가 크게 위축되었습니다
끝
윤리적 사생활보호 법(EPL)
811B

도면159

사용 사례: 이론 탐구

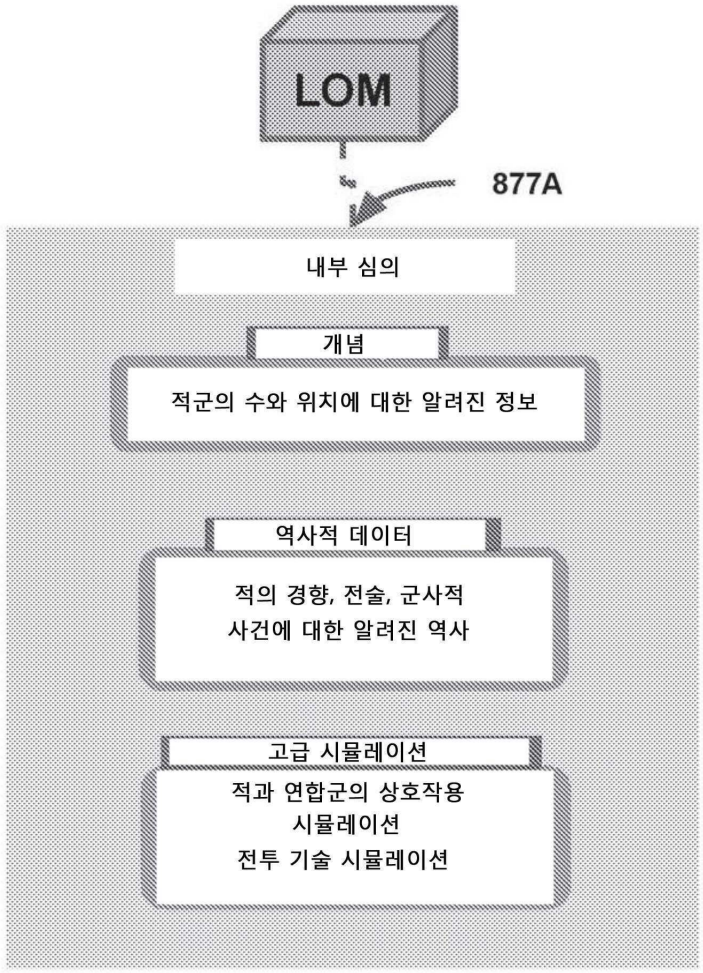


도면160

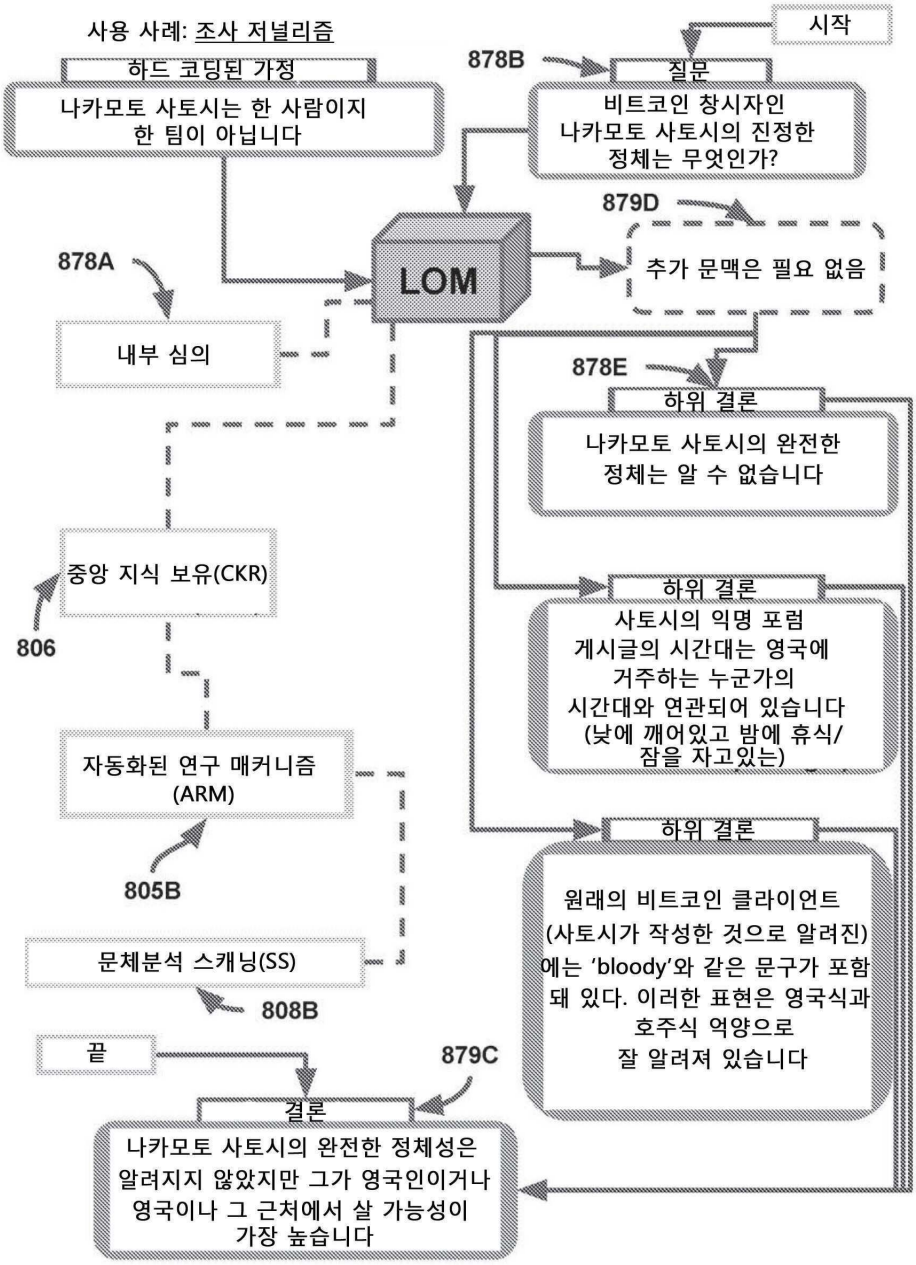

사용 사례: 규칙 수립
하드 코딩된 가정
사용된 탱크 모델은 M1 에이브럼스 탱크입니다
877E
하드 코딩된 가정
사용된 제트기 모델은 리퍼블릭 A-10 썬더 볼트 II 입니다
877B
시작
주장
B 지점에서 50,000 명의 지상군, 35 개의 탱크 및 6 개의 제트기를 위치 A에 보내면 외부 침범의 가장 좋은 공격 형태가 됩니다
LOM
내부 심의
877D
877A
응답(상세 세부 요청)
모든 지상 군의 경험 프로필은 무엇인가?
877C
결론
802C
개인적인 지능 프로필(PIP)
전쟁 게임은 성공 확률이 80 % 인 신뢰도가 70 %입니다. 병력 수를 60,000 명으로 늘리고 F-35에 사용된 제트 모형을 대체하여 성공 확률을 90 %로 높이고 신뢰도는 85 %입니다
행동 모니터링(BM)
끝
819C
811B
윤리적 사생활보호 법(EPL)

도면161

사용 사례: 규칙 수립

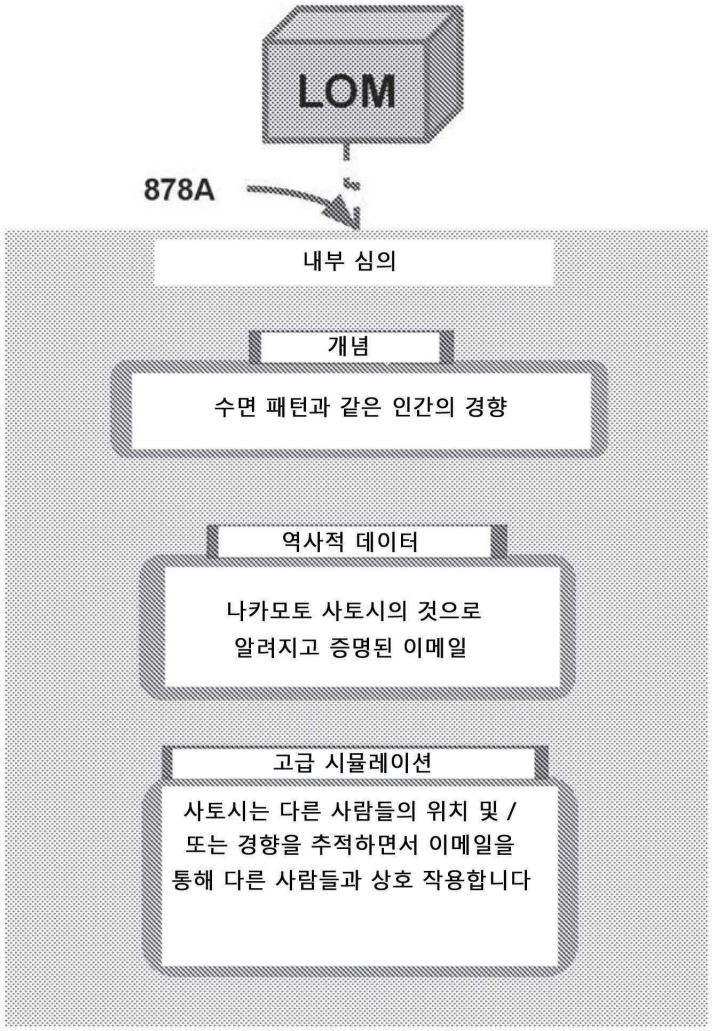


도면162

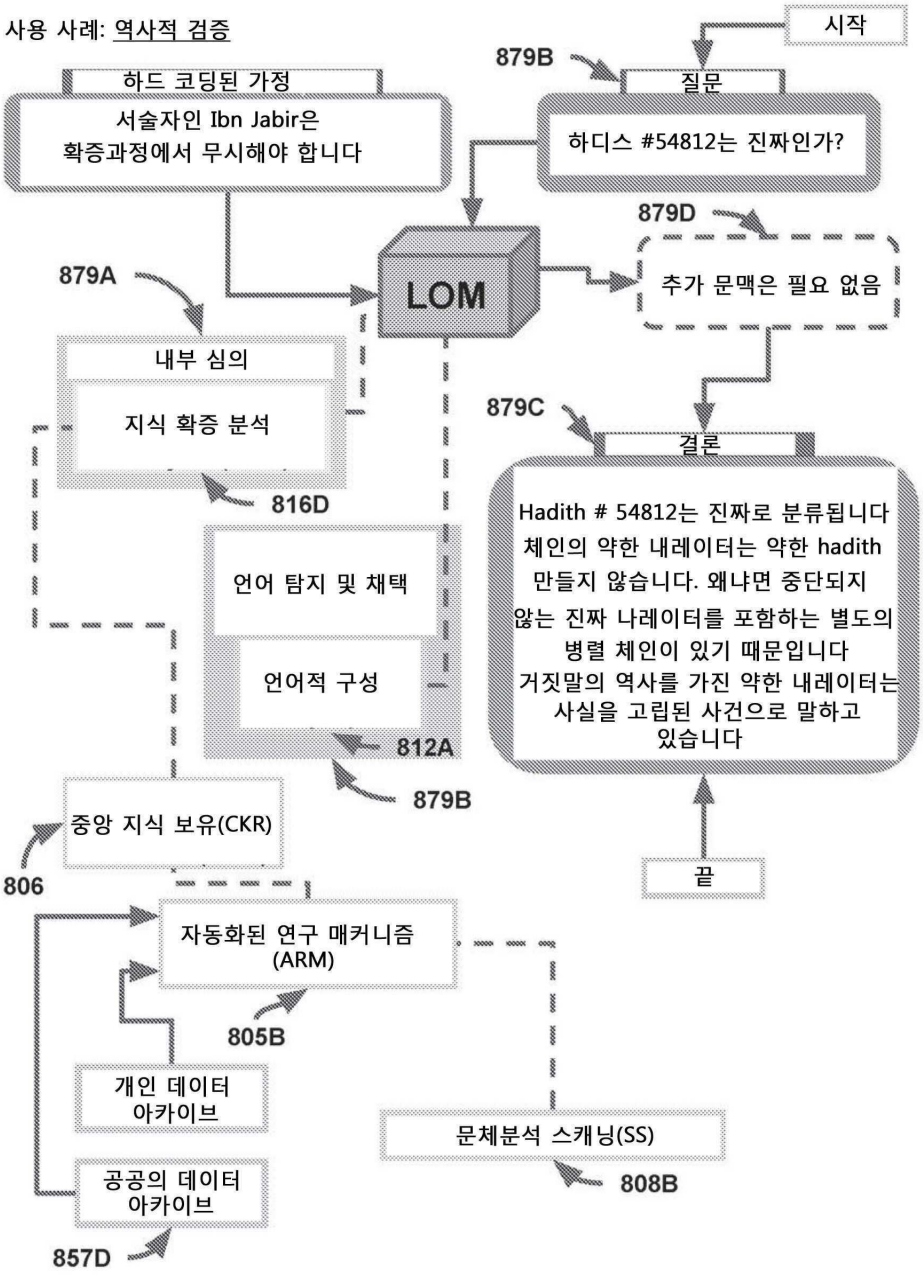

사용 사례: 조사 저널리즘
시작
878B
질문
비트코인 창시자인
나카모토 사토시의 진정한
정체는 무엇인가?
하드 코딩된 가정
나카모토 사토시는 한 사람이지
한 팀이 아닙니다
879D
추가 문맥은 필요 없음
878A
LOM
내부 심의
878E
하위 결론
나카모토 사토시의 완전한
정체는 알 수 없습니다
중앙 지식 보유(CKR)
806
하위 결론
사토시의 익명 포럼
게시글의 시간대는 영국에
거주하는 누군가의
시간대와 연관되어 있습니다
(낮에 깨어있고 밤에 휴식/
잠을 자고있는)
자동화된 연구 매커니즘
(ARM)
805B
하위 결론
원래의 비트코인 클라이언트
(사토시가 작성한 것으로 알려진)
에는 'bloody'와 같은 문구가 포함
돼 있다. 이러한 표현은 영국식과
호주식 억양으로
잘 알려져 있습니다
문체분석 스캐닝(SS)
808B
끝
879C
결론
나카모토 사토시의 완전한 정체성은
알려지지 않았지만 그가 영국인이거나
영국이나 그 근처에서 살 가능성이
가장 높습니다

도면163

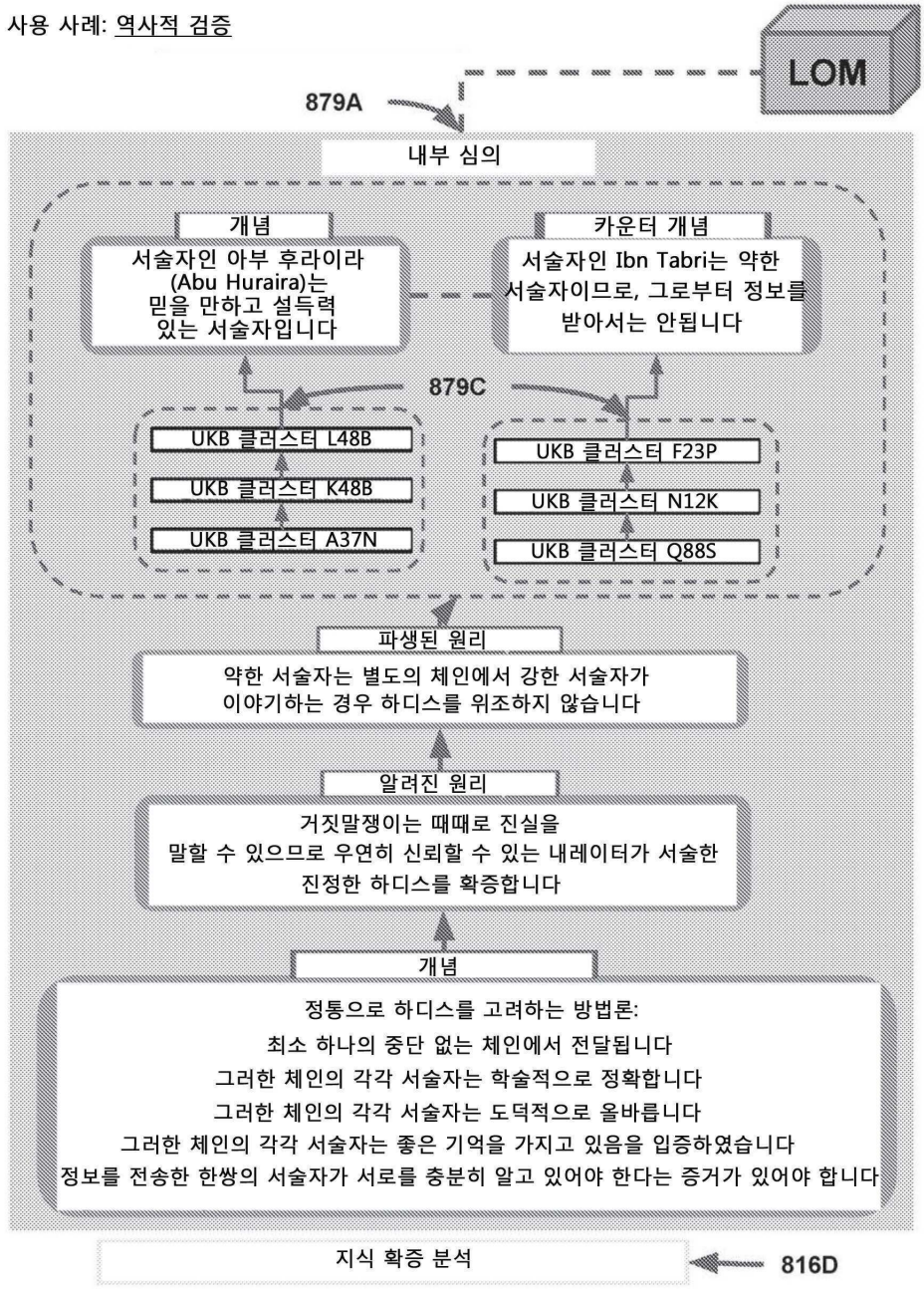
사용 사례: 조사 저널리즘



도면164

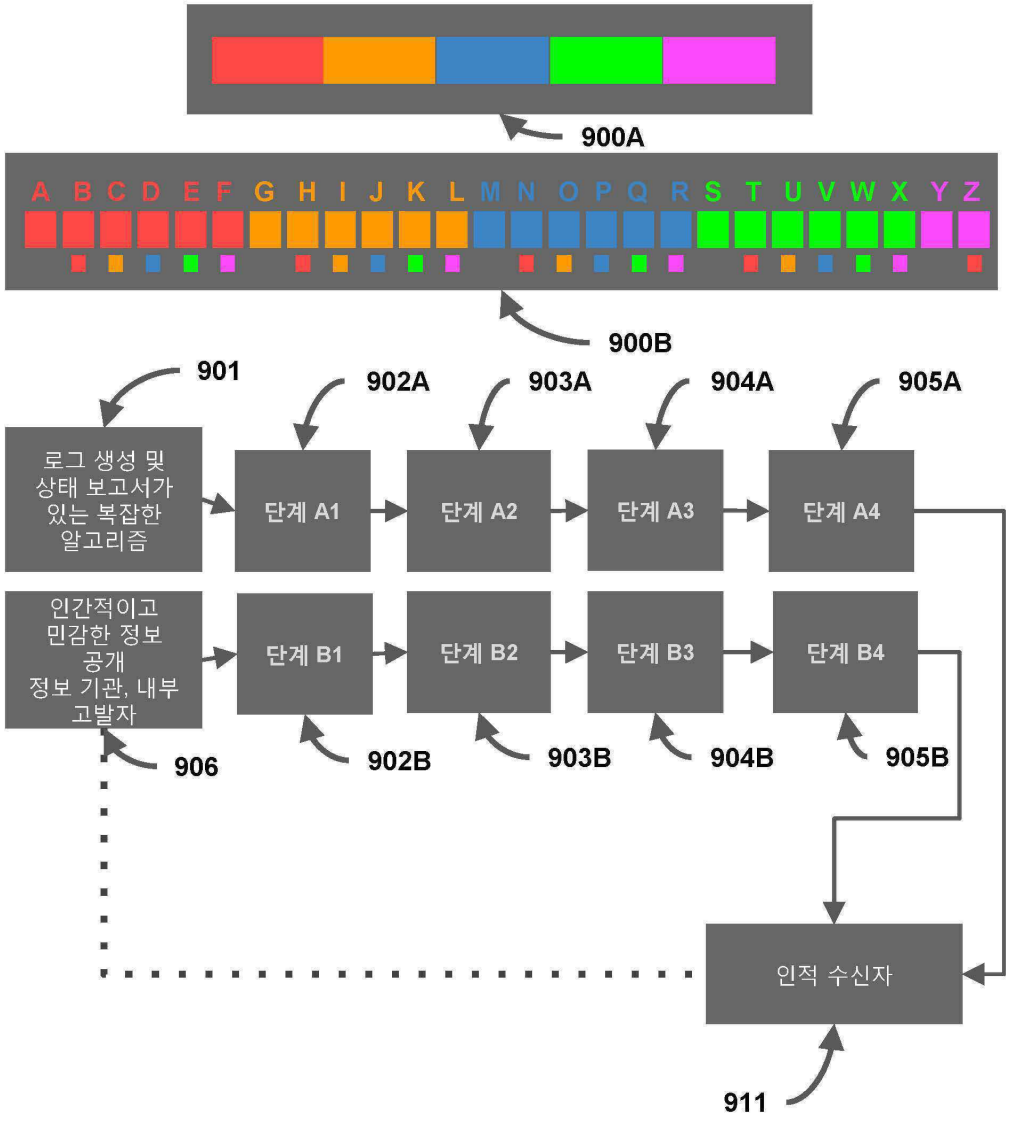

사용 사례: 역사적 검증
시작
879B
질문
하디스 #54812는 진짜인가?
하드 코딩된 가정
서술자인 Ibn Jabir은
확증과정에서 무시해야 합니다
879D
추가 문맥은 필요 없음
879A
LOM
내부 심의
지식 확증 분석
816D
879C
결론
Hadith # 54812는 진짜로 분류됩니다
체인의 약한 내레이터는 약한 hadith
만들지 않습니다. 왜냐면 중단되지
않는 진짜 나레이터를 포함하는 별도의
병렬 체인이 있기 때문입니다
거짓말의 역사를 가진 약한 내레이터는
사실을 고립된 사건으로 말하고
있습니다
언어 탐지 및 채택
언어적 구성
812A
879B
중앙 지식 보유(CKR)
806
끝
자동화된 연구 매커니즘
(ARM)
805B
개인 데이터
아카이브
문체분석 스캐닝(SS)
808B
공공의 데이터
아카이브
857D

도면165

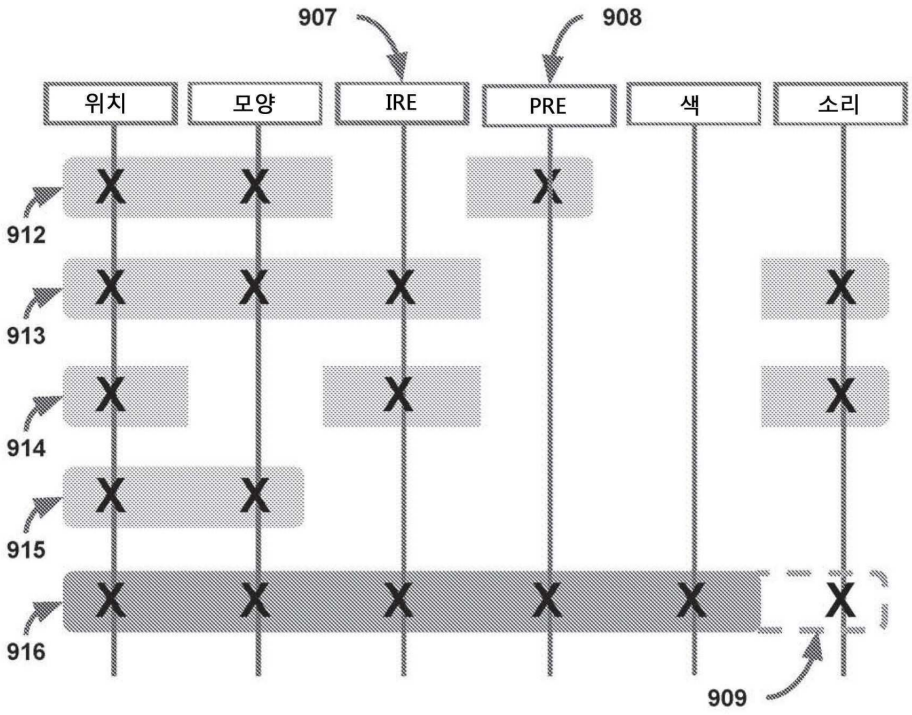

사용 사례: 역사적 검증
LOM
879A
내부 심의
개념
서술자인 아부 후라이라 (Abu Huraira)는 믿을 만하고 설득력 있는 서술자입니다
카운터 개념
서술자인 Ibn Tabri는 약한 서술자이므로, 그로부터 정보를 받아서는 안됩니다
879C
UKB 클러스터 L48B
UKB 클러스터 K48B
UKB 클러스터 A37N
UKB 클러스터 F23P
UKB 클러스터 N12K
UKB 클러스터 Q88S
파생된 원리
약한 서술자는 별도의 체인에서 강한 서술자가 이야기하는 경우 하디스를 위조하지 않습니다
알려진 원리
거짓말쟁이는 때때로 진실을 말할 수 있으므로 우연히 신뢰할 수 있는 내레이터가 서술한 진정한 하디스를 확증합니다
개념
정통으로 하디스를 고려하는 방법론:
최소 하나의 중단 없는 체인에서 전달됩니다
그러한 체인의 각각 서술자는 학술적으로 정확합니다
그러한 체인의 각각 서술자는 도덕적으로 올바릅니다
그러한 체인의 각각 서술자는 좋은 기억을 가지고 있음을 입증하였습니다
정보를 전송한 한쌍의 서술자가 서로를 충분히 알고 있어야 한다는 증거가 있어야 합니다
지식 확증 분석
816D

도면166

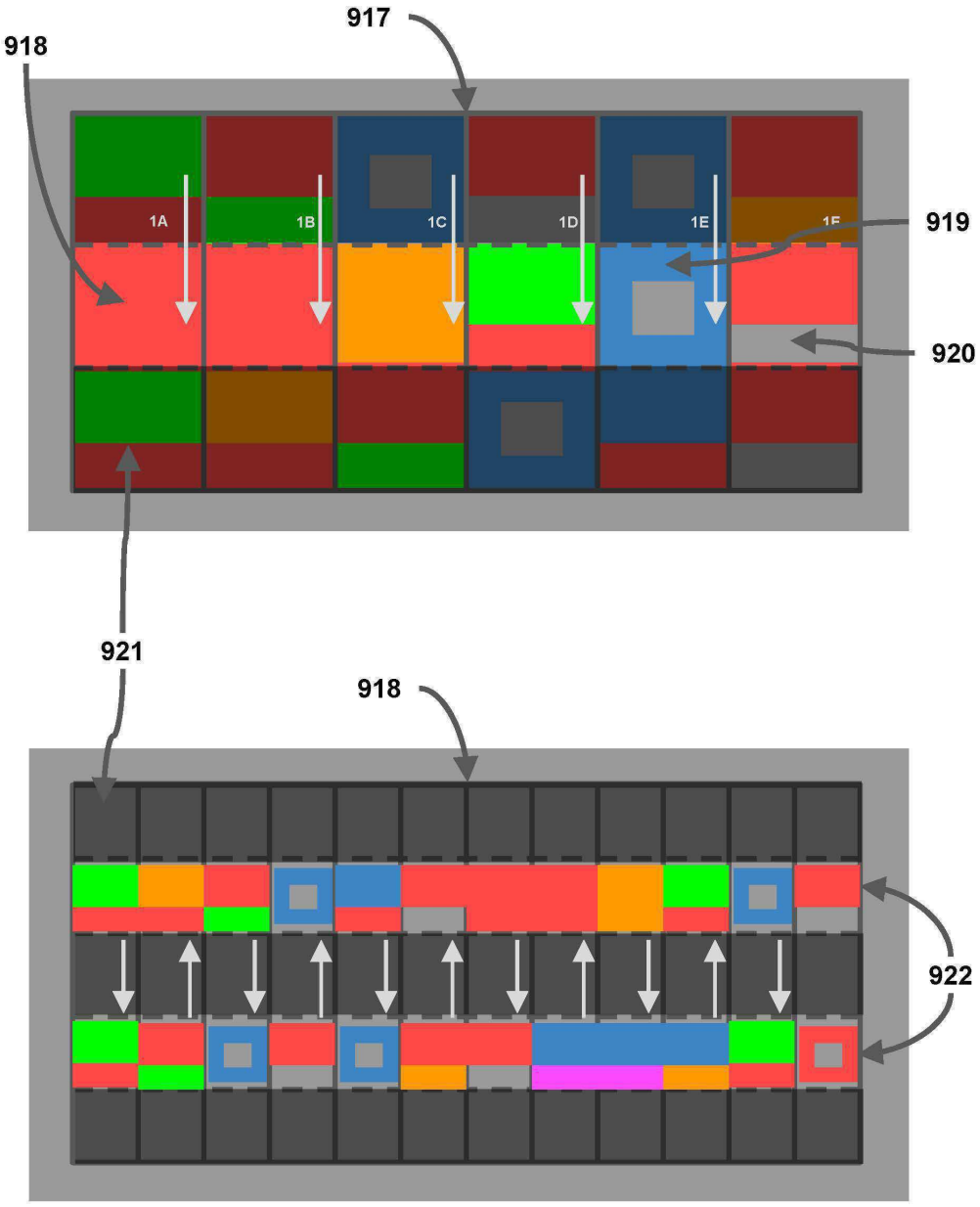
LAQIT은 정보를 사람에게 전송하는 자동화된 시스템을 위한 안전한 언어입니다. 상호 위치가 지정된 유색 도형을 통해 LAQIT를 해석하기 위한 방법론은 기본입니다. 모양, 색 및 위치의 상관 관계는 라틴 알파벳에 대한 특정 번역을 나타냅니다. 따라서 LAQIT를 통해 전달되는 정보를 이해하는 데는 색상 참조가 필수적입니다.



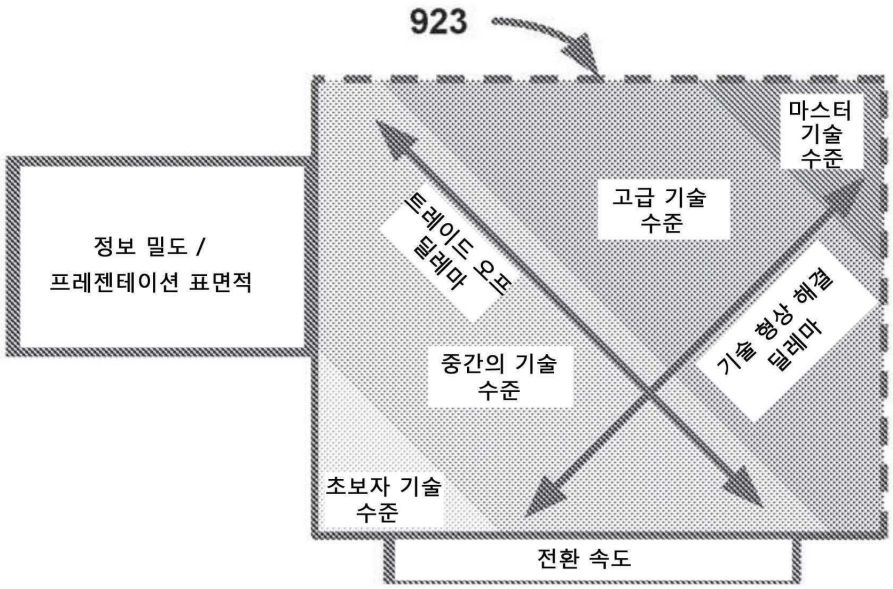
도면167



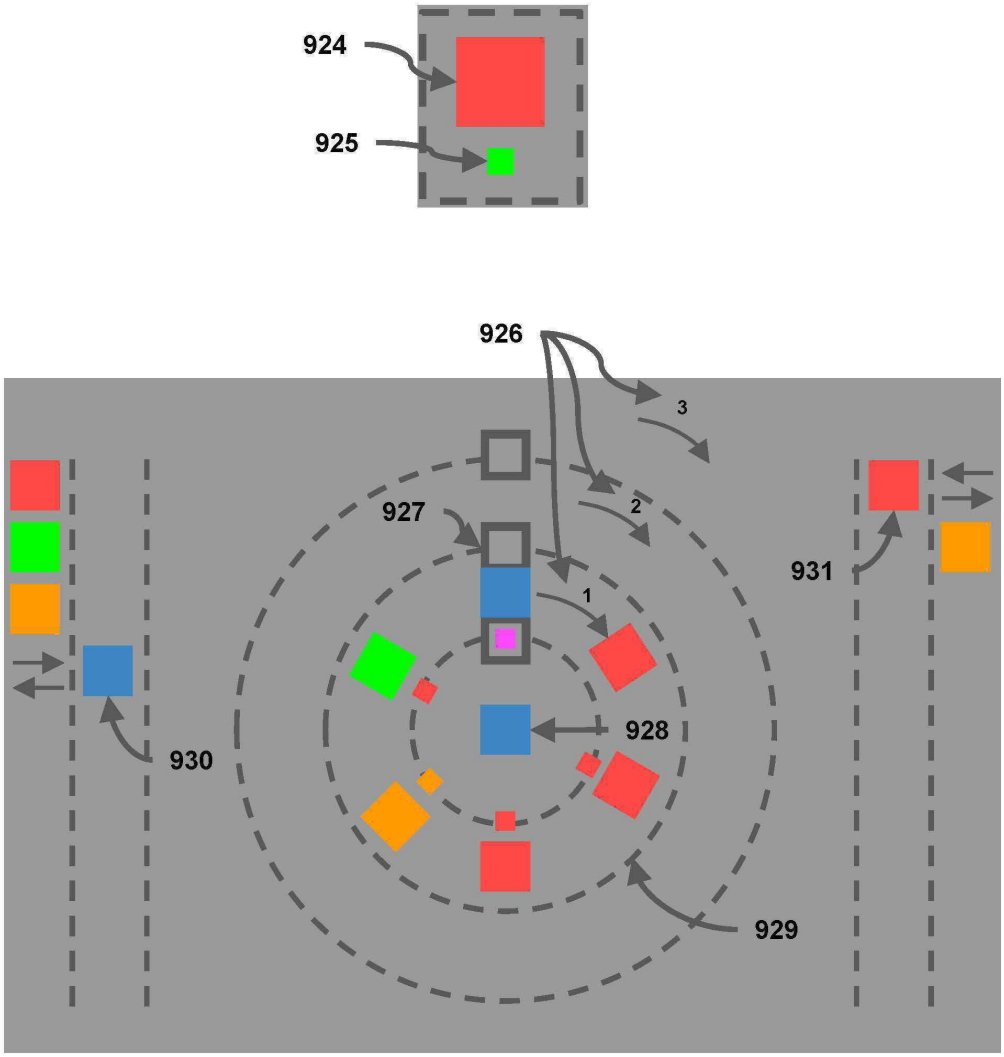
도면168



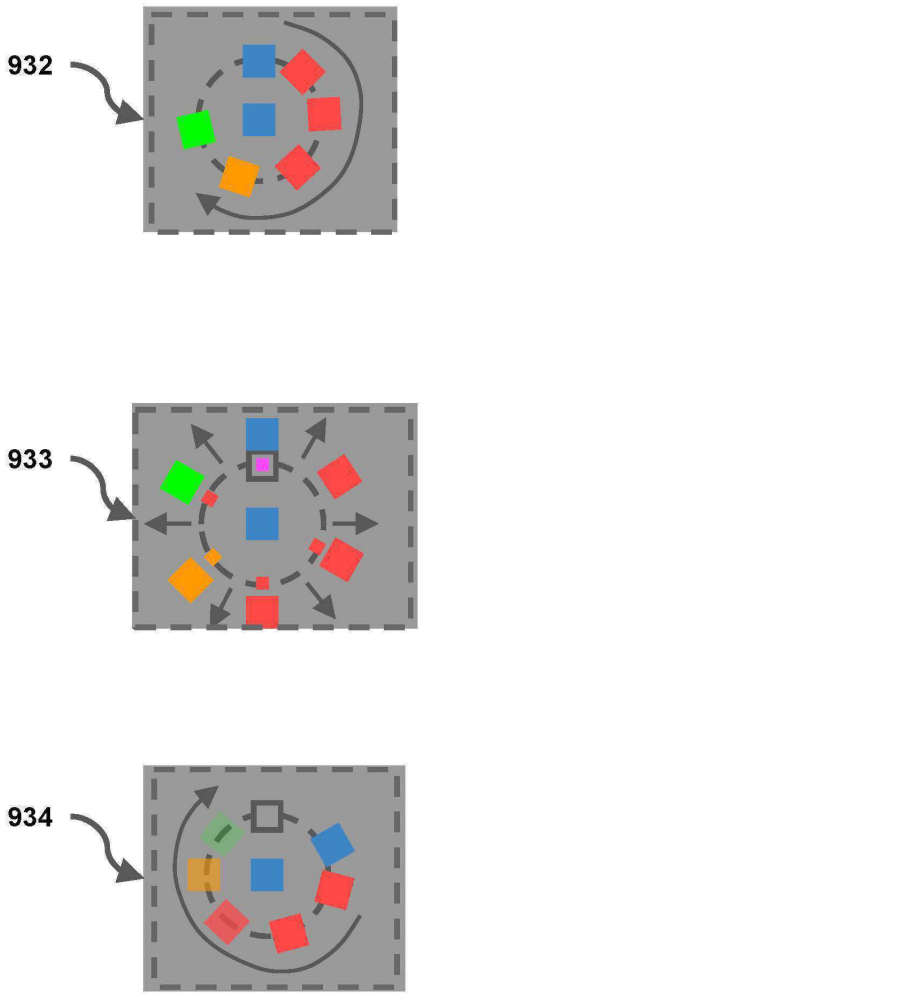
도면169


923
마스터
기술
수준
고급 기술
수준
트레이드 오프
딜레마
기술 형상 해결
딜레마
중간의 기술
수준
초보자 기술
수준
정보 밀도 /
프레젠테이션 표면적
전환 속도

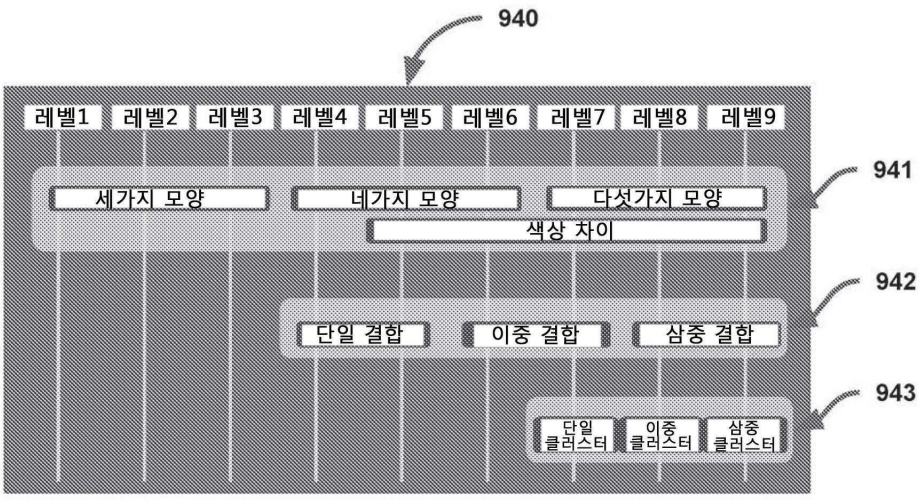
도면170



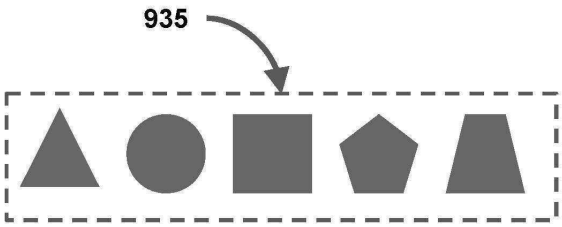
도면171

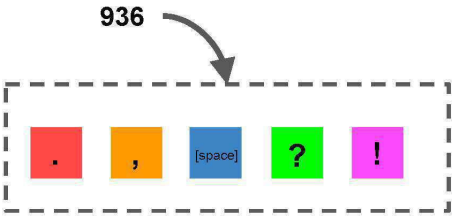

932
933
934

도면172

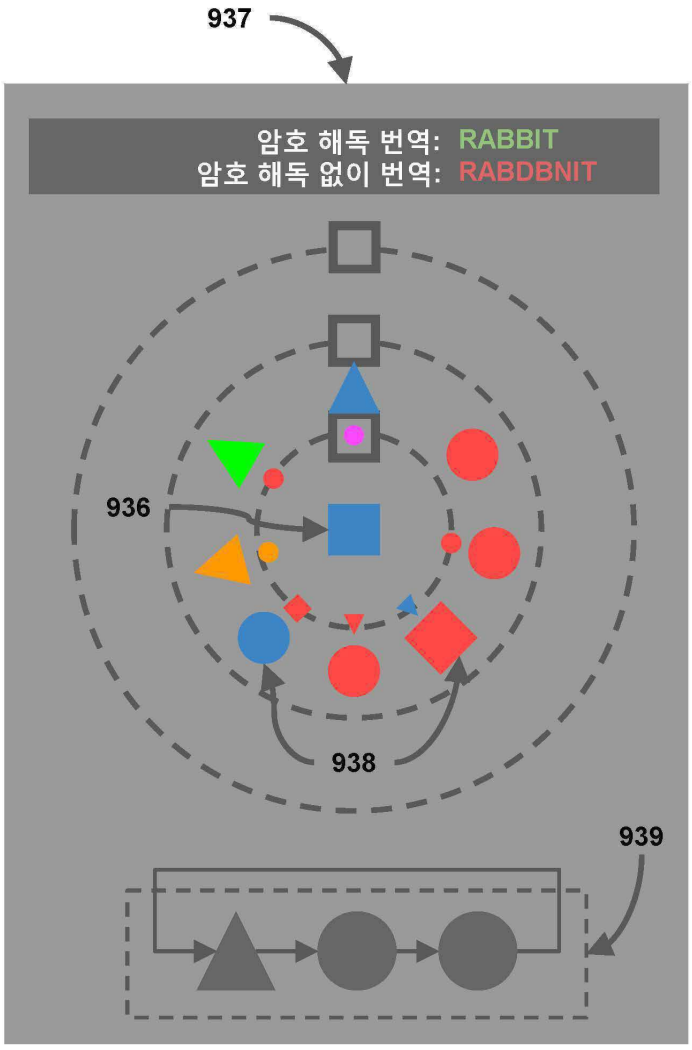

940
레벨1
레벨2
레벨3
레벨4
레벨5
레벨6
레벨7
레벨8
레벨9
941
세가지 모양
네가지 모양
다섯가지 모양
색상 차이
942
단일 결합
이중 결합
삼중 결합
943
단일 클러스터
이중 클러스터
삼중 클러스터

도면173

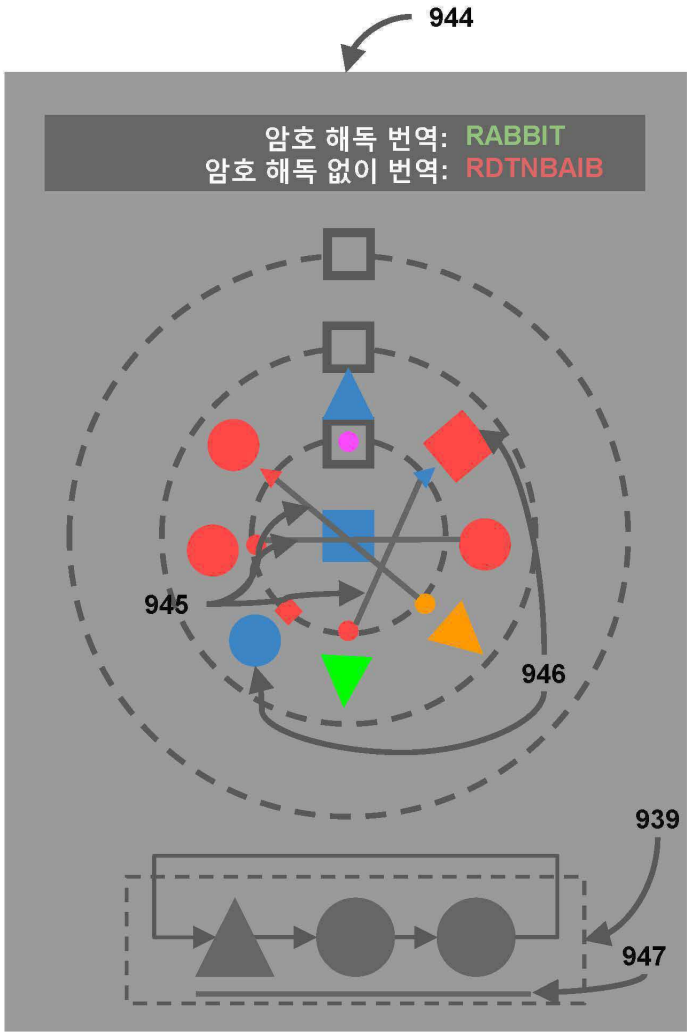




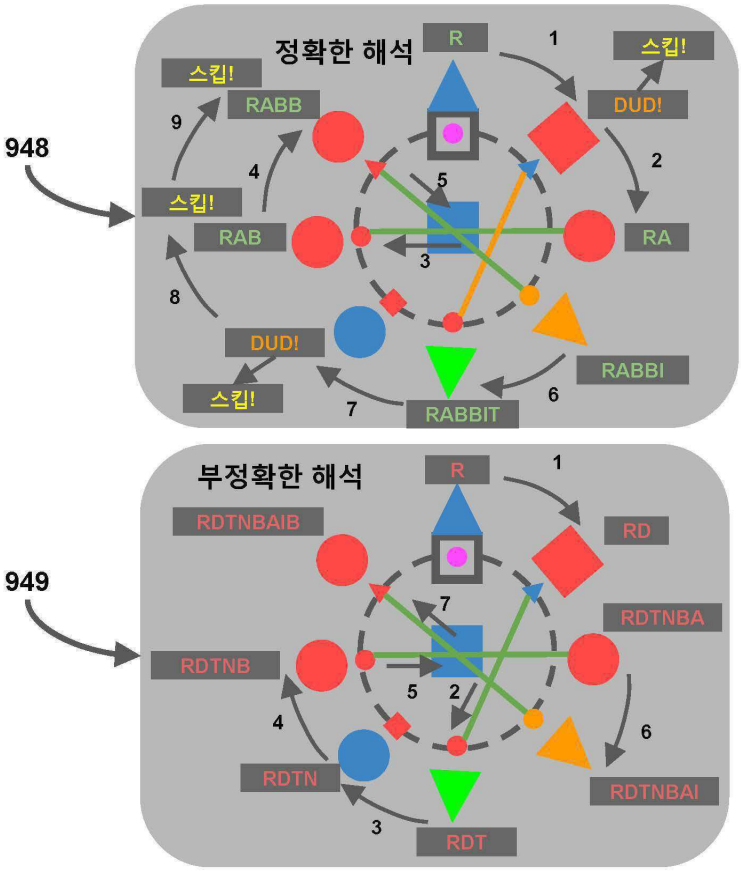
도면174



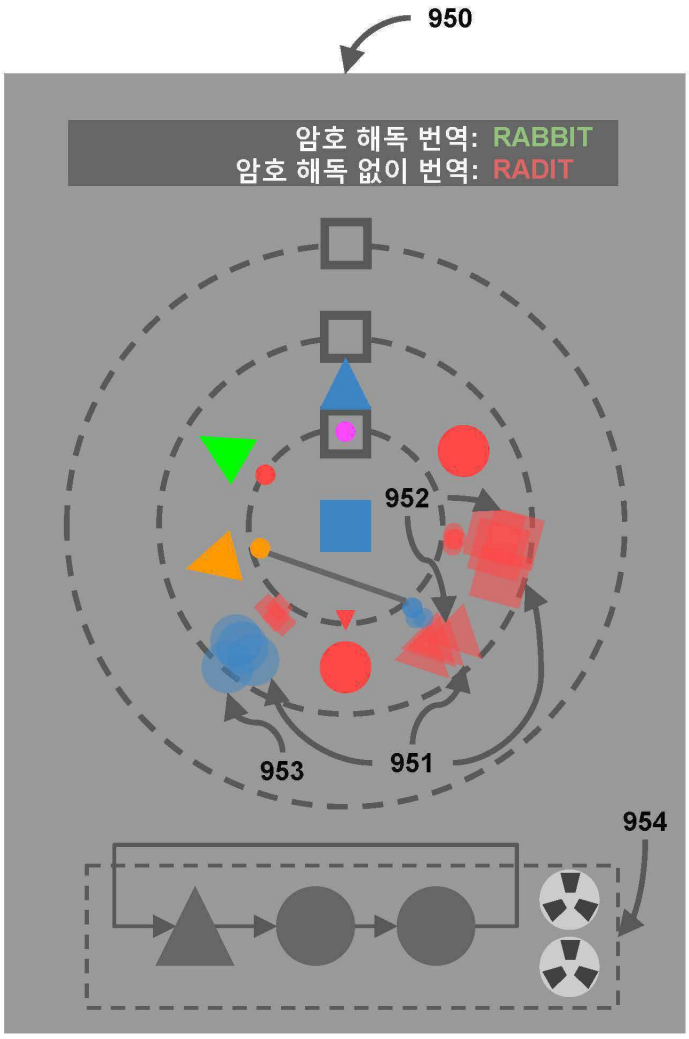
도면175



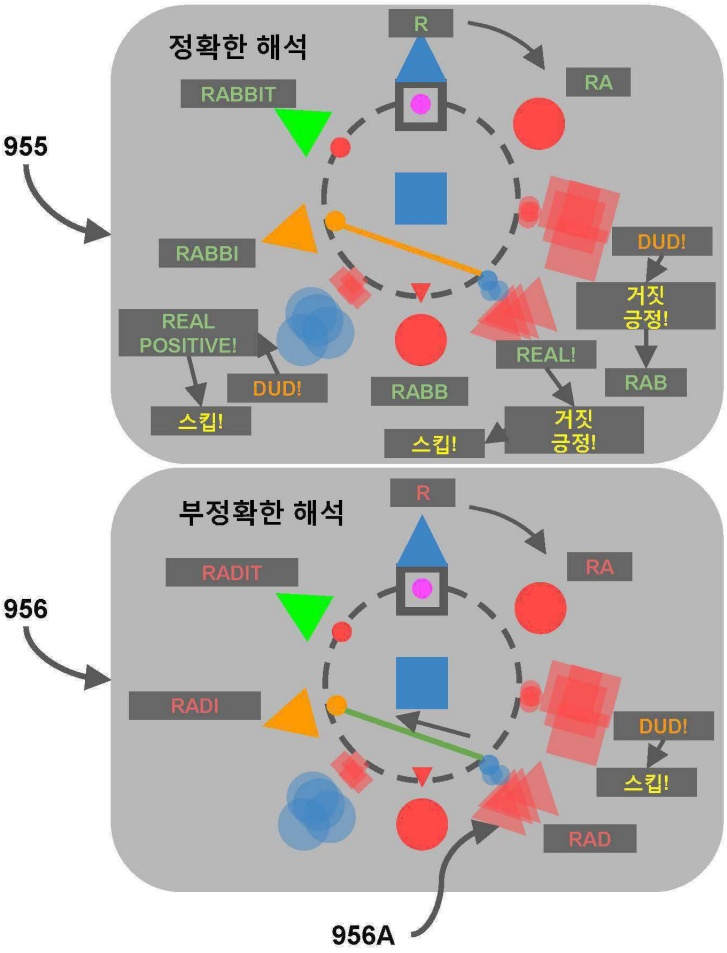
도면176


948
정확한 해석
R
1
스킵!
DUD!
2
RA
RABBI
6
RABBIT
7
스킵!
DUD!
8
RAB
4
RABB
스킵!
9
5
3
949
부정확한 해석
R
1
RD
RDTNBA
6
RDTNBAI
RDT
3
RDTN
4
RDTNB
RDTNBAIB
5
2
7

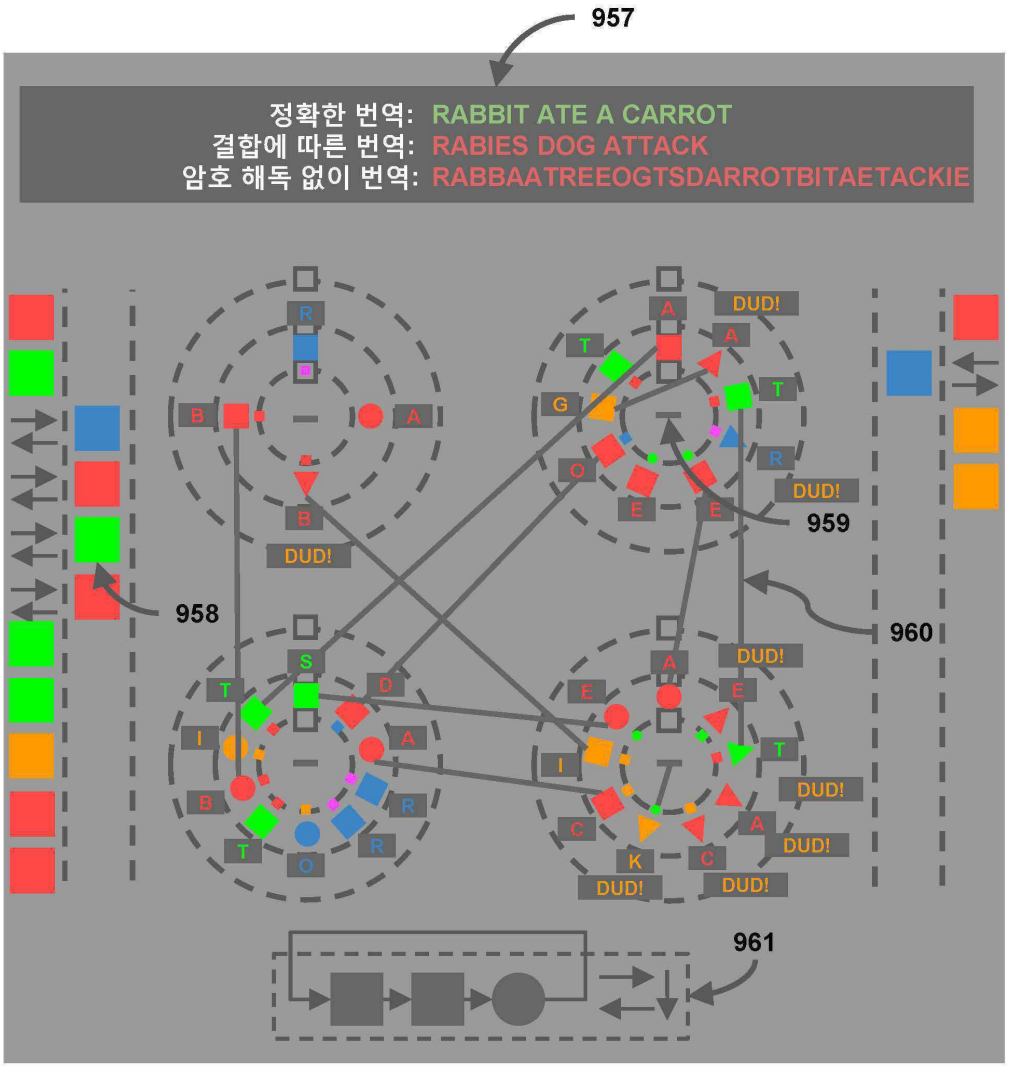
도면177



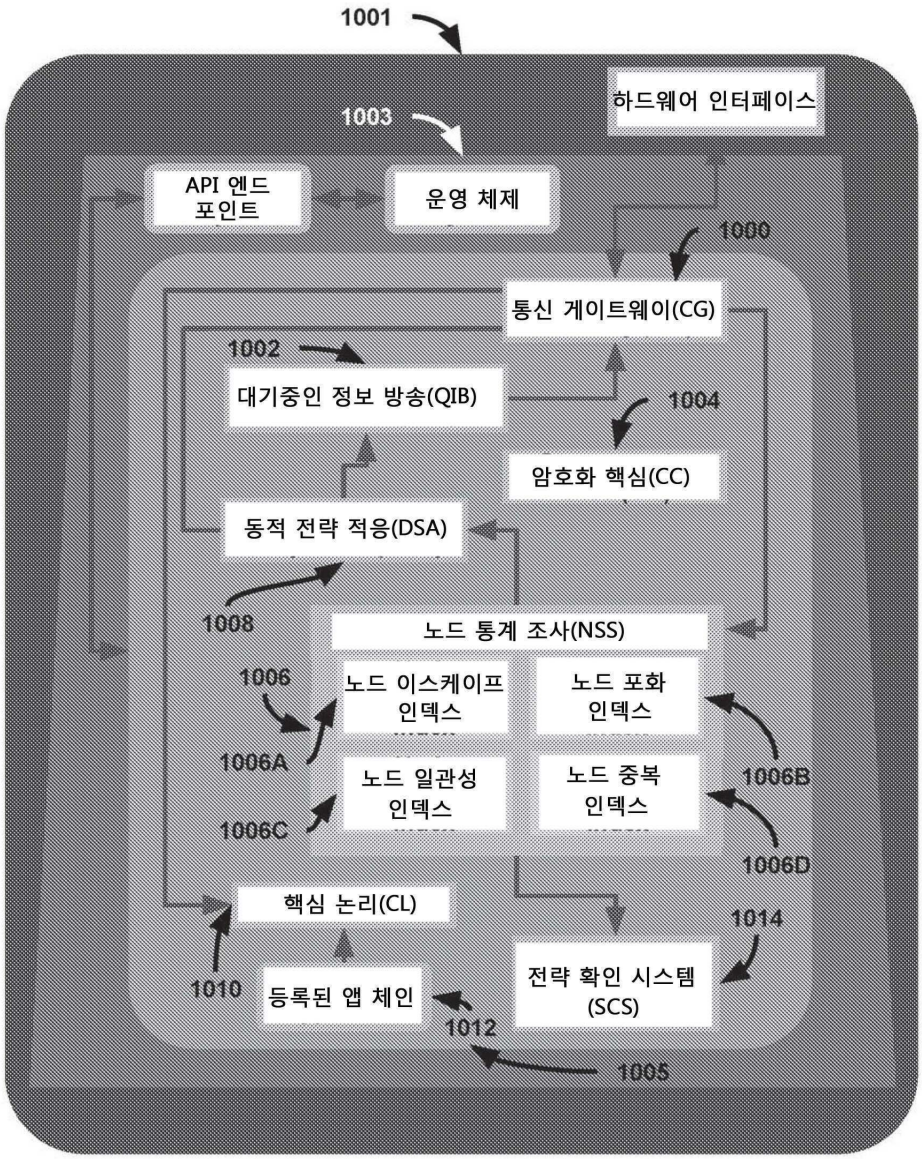
도면178


정확한 해석
R
RA
RABBIT
955
RABBI
DUD!
거짓
긍정!
REAL
POSITIVE!
REAL!
RAB
DUD!
RABB
스킵!
스킵!
거짓
긍정!
부정확한 해석
R
RA
RADIT
956
RADI
DUD!
스킵!
RAD
956A

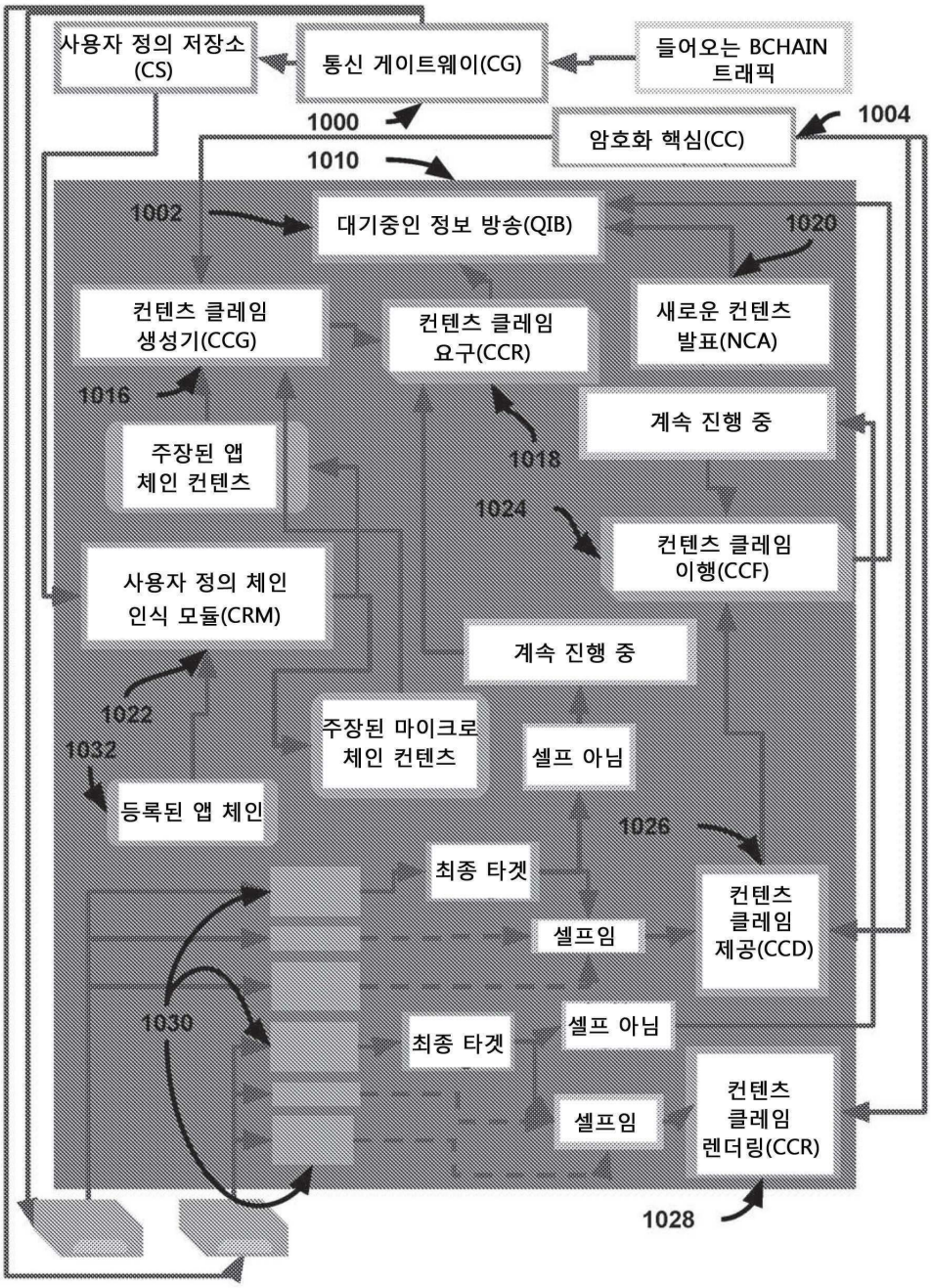
도면179


957
정확한 번역: RABBIT ATE A CARROT
결합에 따른 번역: RABIES DOG ATTACK
암호 해독 없이 번역: RABBAATREEOGTSDARROTBITAETACKIE
958
959
960
961
DUD!

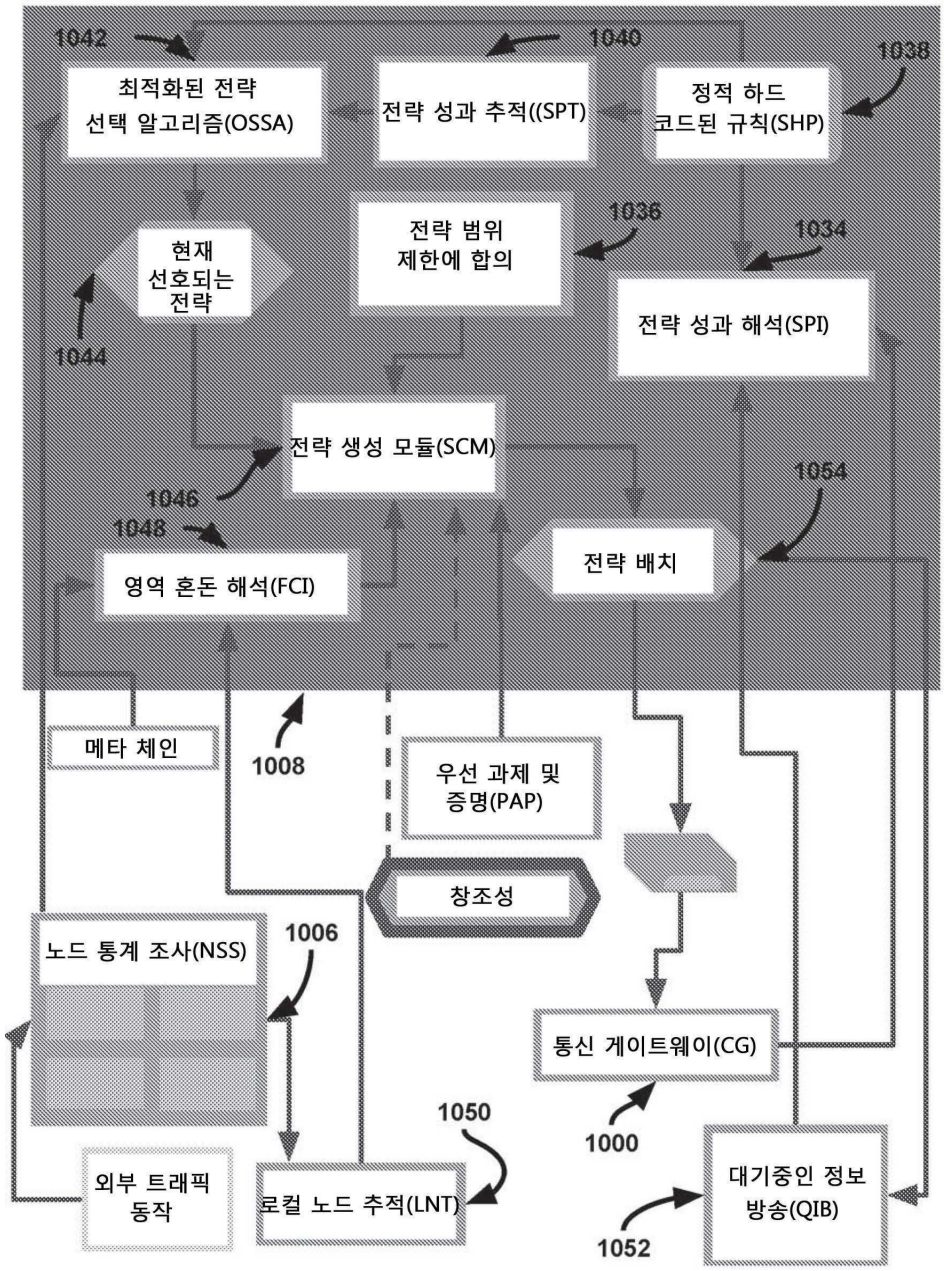
도면180


1001
하드웨어 인터페이스
1003
API 엔드 포인트
운영 체제
1000
통신 게이트웨이(CG)
1002
대기중인 정보 방송(QIB)
1004
암호화 핵심(CC)
동적 전략 적응(DSA)
1008
노드 통계 조사(NSS)
1006
노드 이스케이프 인덱스
노드 포화 인덱스
1006A
노드 일관성 인덱스
노드 중복 인덱스
1006B
1006C
1006D
핵심 논리(CL)
1014
1010
등록된 앱 체인
전략 확인 시스템 (SCS)
1012
1005

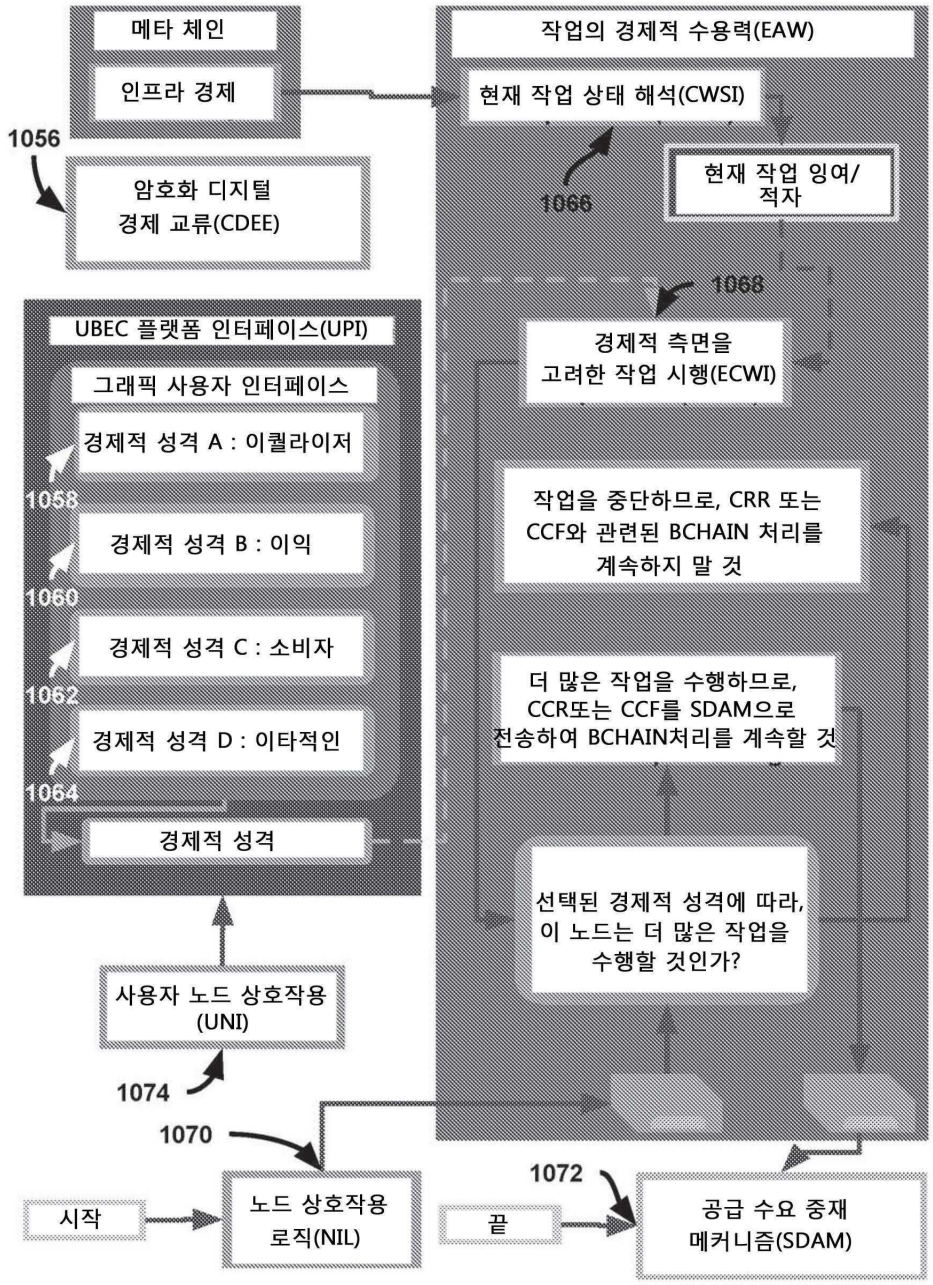
도면181


사용자 정의 저장소 (CS)
통신 게이트웨이(CG)
들어오는 BCHAIN 트래픽
1000
암호화 핵심(CC)
1004
1010
1002
대기중인 정보 방송(QIB)
1020
컨텐츠 클레임 생성기(CCG)
컨텐츠 클레임 요구(CCR)
새로운 컨텐츠 발표(NCA)
1016
계속 진행 중
주장된 앱 체인 컨텐츠
1018
1024
컨텐츠 클레임 이행(CCF)
사용자 정의 체인 인식 모듈(CRM)
계속 진행 중
1022
주장된 마이크로 체인 컨텐츠
셀프 아님
1032
등록된 앱 체인
1026
최종 타겟
컨텐츠 클레임 제공(CCD)
셀프임
1030
최종 타겟
셀프 아님
셀프임
컨텐츠 클레임 렌더링(CCR)
1028

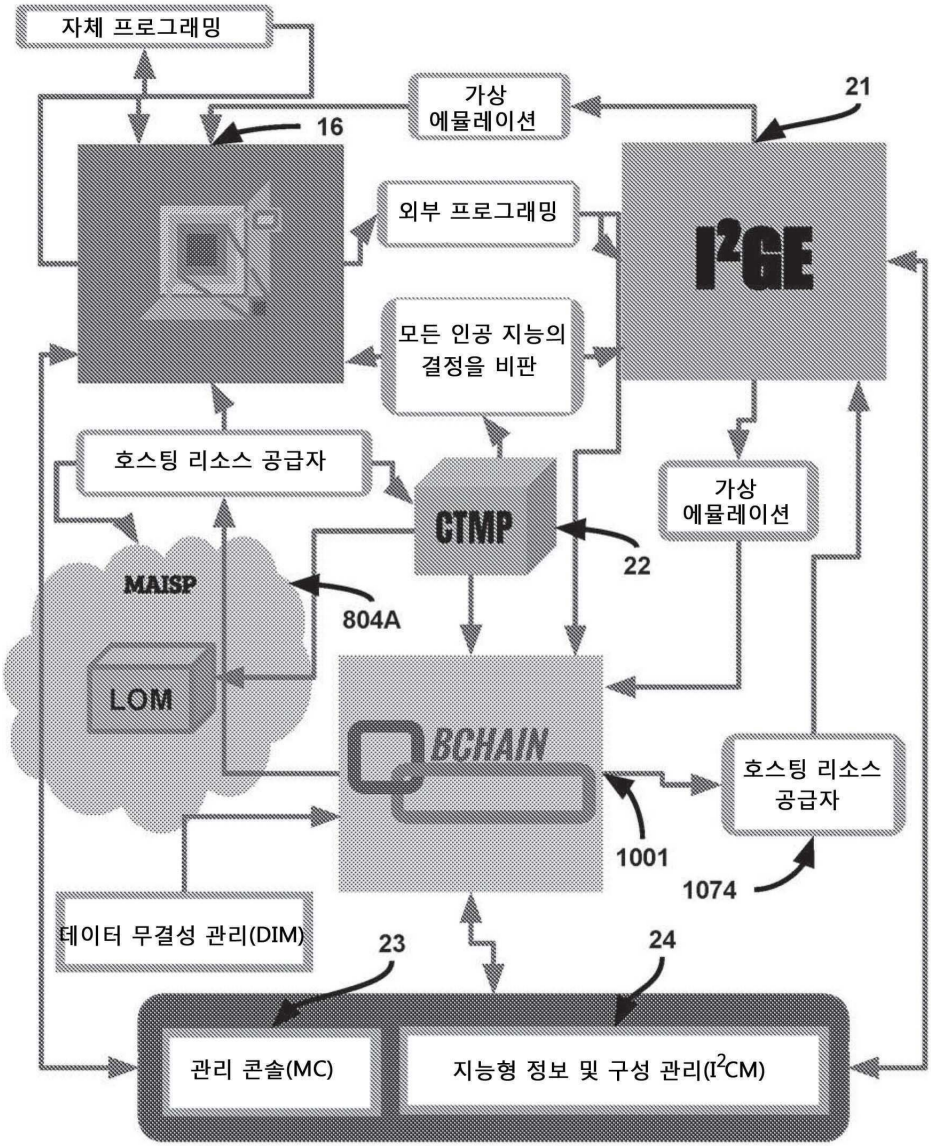
도면182


1042
1040
1038
최적화된 전략
선택 알고리즘(OSSA)
전략 성과 추적((SPT)
정적 하드
코드된 규칙(SHP)
전략 범위
제한에 합의
1036
1034
현재
선호되는
전략
전략 성과 해석(SPI)
1044
전략 생성 모듈(SCM)
1046
1048
1054
영역 혼돈 해석(FCI)
전략 배치
메타 체인
1008
우선 과제 및
증명(PAP)
창조성
노드 통계 조사(NSS)
1006
통신 게이트웨이(CG)
1050
1000
외부 트래픽
동작
로컬 노드 추적(LNT)
대기중인 정보
방송(QIB)
1052

도면183


메타 체인
인프라 경제
1056
암호화 디지털 경제 교류(CDEE)
작업의 경제적 수용력(EAW)
현재 작업 상태 해석(CWSI)
1066
현재 작업 잉여/적자
1068
경제적 측면을 고려한 작업 시행(ECWI)
작업을 중단하므로, CRR 또는 CCF와 관련된 BCHAIN 처리를 계속하지 말 것
더 많은 작업을 수행하므로, CCR또는 CCF를 SDAM으로 전송하여 BCHAIN처리를 계속할 것
선택된 경제적 성격에 따라, 이 노드는 더 많은 작업을 수행할 것인가?
UBEC 플랫폼 인터페이스(UPI)
그래픽 사용자 인터페이스
경제적 성격 A : 이퀄라이저
1058
경제적 성격 B : 이익
1060
경제적 성격 C : 소비자
1062
경제적 성격 D : 이타적인
1064
경제적 성격
사용자 노드 상호작용(UNI)
1074
1070
시작
노드 상호작용 로직(NIL)
끝
1072
공급 수요 중재 메커니즘(SDAM)

도면184


자체 프로그래밍
가상
에뮬레이션
16
21
외부 프로그래밍
I²GE
모든 인공 지능의
결정을 비판
호스팅 리소스 공급자
CTMP
가상
에뮬레이션
22
MAISP
804A
LOM
BCHAIN
호스팅 리소스
공급자
1001
1074
데이터 무결성 관리(DIM)
23
24
관리 콘솔(MC)
지능형 정보 및 구성 관리(I²CM)